SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D
Notions complémentaires

Manuel de programmation

Valable pour

Commande
SINUMERIK 840D sl / 840DE sl / 828D

CNC Software Version 4.8 SP3

08/2018
6FC5398-2BP40-6DA2

Avant-propos
Consignes de sécurité élémentaires 1
Programmation CN flexible 2
Gestion des fichiers et programmes 3
Zones de protection 4
Instructions de déplacement spéciales 5
Transformations des coordonnées (frames) 6
Transformations 7
Chaînes cinématiques 8
Prévention des collisions avec chaînes cinématiques 9
Transformations avec chaînes cinématiques 10
Corrections d'outils 11
Modes de déplacement 12
couplages d'axes 13
Actions synchrones 14
Oscillation 15
Poinçonnage et grignotage 16
Rectification 17
Autres fonctions 18
Programmes de chariotage personnalisés 19
Programmation externe de cycles 20
Tableaux 21
Annexe A

Промэлектроника, sales@prom-elec.com

Mentions légales

Signalétique d'avertissement

Ce manuel donne des consignes que vous devez respecter pour votre propre sécurité et pour éviter des dommages matériels. Les avertissements servant à votre sécurité personnelle sont accompagnés d'un triangle de danger, les avertissements concernant uniquement des dommages matériels sont dépourvus de ce triangle. Les avertissements sont représentés ci-après par ordre décroissant de niveau de risque.

⚠ DANGER
signifie que la non-application des mesures de sécurité appropriées **entraîne** la mort ou des blessures graves.

⚠ ATTENTION
signifie que la non-application des mesures de sécurité appropriées **peut entraîner** la mort ou des blessures graves.

⚠ PRUDENCE
signifie que la non-application des mesures de sécurité appropriées peut entraîner des blessures légères.

IMPORTANT
signifie que la non-application des mesures de sécurité appropriées peut entraîner un dommage matériel.

En présence de plusieurs niveaux de risque, c'est toujours l'avertissement correspondant au niveau le plus élevé qui est reproduit. Si un avertissement avec triangle de danger prévient des risques de dommages corporels, le même avertissement peut aussi contenir un avis de mise en garde contre des dommages matériels.

Personnes qualifiées

L'appareil/le système décrit dans cette documentation ne doit être manipulé que par du **personnel qualifié** pour chaque tâche spécifique. La documentation relative à cette tâche doit être observée, en particulier les consignes de sécurité et avertissements. Les personnes qualifiées sont, en raison de leur formation et de leur expérience, en mesure de reconnaître les risques liés au maniement de ce produit / système et de les éviter.

Utilisation des produits Siemens conforme à leur destination

Tenez compte des points suivants:

⚠ ATTENTION
Les produits Siemens ne doivent être utilisés que pour les cas d'application prévus dans le catalogue et dans la documentation technique correspondante. S'ils sont utilisés en liaison avec des produits et composants d'autres marques, ceux-ci doivent être recommandés ou agréés par Siemens. Le fonctionnement correct et sûr des produits suppose un transport, un entreposage, une mise en place, un montage, une mise en service, une utilisation et une maintenance dans les règles de l'art. Il faut respecter les conditions d'environnement admissibles ainsi que les indications dans les documentations afférentes.

Marques de fabrique

Toutes les désignations repérées par ® sont des marques déposées de Siemens AG. Les autres désignations dans ce document peuvent être des marques dont l'utilisation par des tiers à leurs propres fins peut enfreindre les droits de leurs propriétaires respectifs.

Exclusion de responsabilité

Nous avons vérifié la conformité du contenu du présent document avec le matériel et le logiciel qui y sont décrits. Ne pouvant toutefois exclure toute divergence, nous ne pouvons pas nous porter garants de la conformité intégrale. Si l'usage de ce manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
ALLEMAGNE

Numéro de référence du document: 6FC5398-2BP40-6DA2
℗ 08/2018 Sous réserve de modifications

Copyright © Siemens AG 1995 - 2018.
Tous droits réservés

Avant-propos

Documentation SINUMERIK

La documentation SINUMERIK comporte les catégories suivantes :

- Documentation générale/catalogues
- Documentation utilisateur
- Documentation constructeur/S.A.V.

Informations complémentaires

Sous ce lien (https://support.industry.siemens.com/cs/de/en/view/108464614), vous trouverez différentes informations pour :

- Commander la documentation/aperçu des documents
- Télécharger des documents à partir de liens supplémentaires
- Utiliser la documentation en ligne (trouver des manuels/informations et y effectuer des recherches)

Pour toute question concernant la documentation technique (par ex. suggestion, correction), envoyez un courriel à cette adresse (mailto:docu.motioncontrol@siemens.com).

mySupport/Documentation

Plus d'informations sur la documentation Siemens disponible et sur les possibilités de l'utiliser pour créer votre propre documentation machine, voir : lien (https://support.industry.siemens.com/My/ww/fr/documentation).

Formation

Ce lien (http://www.siemens.com/sitrain) livre des informations sur SITRAIN, le programme de formations de Siemens pour les produits, systèmes et solutions d'entraînements et d'automatisation.

FAQ

La Foire Aux Questions se trouve sur les pages Service&Support sous Support produit (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

Des informations relatives à SINUMERIK figurent à cette adresse (http://www.siemens.com/sinumerik).

Groupe cible

Le présent manuel s'adresse aux :

- programmeurs
- ingénieurs de projet

Utilité

Le Manuel de programmation permet au groupe cible de créer, d'écrire, de tester des programmes et des interfaces logicielles et de supprimer des erreurs.

Version standard

Le présent manuel de programmation décrit les fonctionnalités de la version standard. Les options complémentaires ou modifications apportées par le constructeur de la machine-outil sont documentées par celui-ci.

La commande numérique peut posséder des fonctions qui dépassent le cadre de la présente description. Le client ne peut toutefois pas faire valoir de droit en liaison avec ces fonctions, que ce soit dans le cas de matériels neufs ou dans le cadre d'interventions du service après-vente.

De même, pour des raisons de clarté, la présente documentation ne contient pas toutes les informations de détail relatives à toutes les variantes du produit ; elle ne peut pas non plus tenir compte de tous les cas d'installation, d'exploitation et de maintenance.

Technical Support

Si vous avez besoin de conseils techniques, vous trouverez sur Internet les coordonnées téléphoniques appropriées à l'adresse (https://support.industry.siemens.com/sc/ww/fr/sc/support-technique/oid2090) suivante, dans la zone "Contact".

Informations sur la structure et le contenu

Manuel de programmation Notions de base / Notions complémentaires

Les descriptions relatives à la programmation CN sont réparties sur deux manuels :

1. **Notions de base**
   Le manuel de programmation "Notions de base" s'adresse aux techniciens utilisant la machine-outil et suppose d'avoir la connaissance préalable des opérations de perçage, de fraisage et de tournage. Les instructions et commandes, conformes à la norme DIN 66025, sont illustrées par des exemples de programmation simples.
2. **Notions complémentaires**
   Le manuel de programmation "Notions complémentaires" s'adresse aux technologues connaissant toutes les possibilités de programmation. Grâce à un langage de programmation spécifique, la commande SINUMERIK permet d'élaborer un programme pièce complexe (par exemple : surfaces gauches, coordination entre canaux...) et dispense le technologue d'une programmation fastidieuse.

Disponibilité des éléments de langage

Tous les éléments de langage CN décrits dans le présent manuel sont disponibles pour SINUMERIK 840D sl. La disponibilité pour SINUMERIK 828D est indiquée dans le tableau "Instructions : Disponibilité pour SINUMERIK 828D (Page 912)".

Sommaire

Avant-propos....3

1 Consignes de sécurité élémentaires....19

1.1 Consignes de sécurité générales....19

1.2 Garantie et responsabilité pour les exemples d'application....20

1.3 Sécurité industrielle....21

2 Programmation CN flexible....23

2.1 Variables....23
2.1.1 Données système....23
2.1.2 Variables utilisateur prédéfinies : Paramètre de calcul....26
2.1.2.1 Paramètres de calcul spécifiques à un canal (R)....26
2.1.2.2 Paramètres de calcul globaux (RG)....27
2.1.3 Variables utilisateur prédéfinies : variables Link....29
2.1.4 Définition de variables utilisateur (DEF)....31
2.1.5 Redéfinition de données système, données utilisateur et instructions de langage CN (REDEF)....37
2.1.6 Attribut : Valeur d'initialisation....41
2.1.7 Attribut : valeurs limites (LLI, ULI)....44
2.1.8 Attribut : unité physique (PHU)....46
2.1.9 Attribut : droits d'accès (APR, APW, APRP, APWP, APRB, APWB)....48
2.1.10 Attribut : classe de données (DCM, DCI, DCU) – SINUMERIK 828D uniquement....54
2.1.11 Vue d'ensemble des attributs définissables et redéfinissables....55
2.1.12 Définition et initialisation de variables de tableau (DEF, SET, REP)....56
2.1.13 Définition et initialisation de variables de tableau (DEF, SET, REP) : Informations complémentaires....61
2.1.14 Types de données....62
2.1.15 Vérification de l'existence d'une variable (ISVAR)....63
2.1.16 Lecture des valeurs d'attribut / du type de données (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)....65

2.2 Programmation indirecte....72
2.2.1 Programmation indirecte d'adresses....72
2.2.2 Programmation indirecte d'instructions G....74
2.2.3 Programmation indirecte d'attributs de position (GP)....75
2.2.4 Programmation indirecte de lignes de programme pièce (EXECSTRING)....78

2.3 Fonctions de calcul....80

2.4 Opérations relationnelles et opérations logiques....83

2.5 Correction de la précision pour les erreurs relationnelles (TRUNC)....85

2.6 Minimum, maximum et plage de variables (MINVAL, MAXVAL, BOUND)....87

2.7 Priorité des opérations....89

2.8 Conversions de types possibles....90

2.9 Opérations sur les chaînes de caractères....91
2.9.1 Conversion de type en type STRING (AXSTRING) 91
2.9.2 Conversion de type à partir de STRING (NUMBER, ISNUMBER, AXNAME) 92
2.9.3 Concaténation de chaînes de caractères (<<) 93
2.9.4 Conversion en minuscules/majuscules (TOLOWER, TOUPPER) 94
2.9.5 Déterminer la longueur d'une chaîne de caractères (STRLEN) 95
2.9.6 Recherche de caractères/chaînes de caractères dans une chaîne (INDEX, RINDEX, MINDEX, MATCH) 96
2.9.7 Sélection d'une chaîne de caractères partielle (SUBSTR) 97
2.9.8 Lecture et écriture de caractères individuels....98
2.9.9 Formatage d'une chaîne (SPRINT)....99

2.10 Sauts de programme....109
2.10.1 Retour au début du programme (GOTOS)....109
2.10.2 Sauts de programme sur repères de saut (IF, GOTOB, GOTOF, GOTO, GOTOC)....110
2.10.3 Saut de programme (CASE ... OF ... DEFAULT ...)....113

2.11 Répétition de sections de programme (REPEAT, REPEATB, ENDLABEL, P)....115

2.12 Structures de contrôle....121
2.12.1 Instruction conditionnelle et branchement (IF, ELSE, ENDIF)....123
2.12.2 Boucle de programme sans fin (LOOP, ENDLOOP)....124
2.12.3 Boucle de comptage (FOR ... TO ..., ENDFOR)....125
2.12.4 Boucle de programme avec condition en début de boucle (WHILE, ENDWHILE)....126
2.12.5 Boucle de programme avec condition en fin de boucle (REPEAT, UNTIL)....127
2.12.6 Exemple de programme avec des structures de contrôle imbriquées....127

2.13 Ordres de coordination (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)....129

2.14 Routine d'interruption (ASUP)....134
2.14.1 Fonction d'une routine d'interruption....134
2.14.2 Création d'une routine d'interruption....135
2.14.3 Affectation et démarrage d'une routine d'interruption (SETINT, PRIO, BLSYNC)....136
2.14.4 Désactivation/activation de l'affectation d'une routine d'interruption (DISABLE, ENABLE)....138
2.14.5 Suppression d'une affectation de routine d'interruption (CLRINT)....139
2.14.6 Relèvement rapide du contour (SETINT LIFTFAST, ALF)....140
2.14.7 Sens de retrait rapide du contour 142
2.14.8 Séquence de déplacement avec des routines d'interruption....145

2.15 Permutation d'axe, permutation de broche (RELEASE, GET, GETD)....146

2.16 Transmettre un axe à un autre canal (AXTOCHAN)....151

2.17 Activation des paramètres machine (NEWCONF)....153

2.18 Ecriture d'un fichier (WRITE)....154

2.19 Suppression d'un fichier (DELETE)....158

2.20 Lecture des lignes d'un fichier (READ)....159

2.21 Vérification de l'existence d'un fichier (ISFILE)....161

2.22 Lecture des informations fichier (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)....162

2.23 Arrondissement (ROUNDUP) 164

2.24 Technique des sous-programmes....165

2.24.1 Généralités....165
2.24.1.1 Sous-programme....165
2.24.1.2 Nom de sous-programme....166
2.24.1.3 Imbrication de sous-programmes....167
2.24.1.4 Chemin de recherche....168
2.24.1.5 Paramètres formels et paramètres effectifs....169
2.24.1.6 Transfert de paramètres....169
2.24.2 Définition d'un sous-programme....171
2.24.2.1 Sous-programme sans transfert de paramètres....171
2.24.2.2 Sous-programme avec transfert de paramètres Call-by-Value (PROC)....172
2.24.2.3 Sous-programme avec transfert de paramètres Call-by-Reference (PROC, VAR)....174
2.24.2.4 Sauvegarde des fonctions G modales (SAVE)....176
2.24.2.5 Inhibition de l'exécution bloc par bloc (SBLOF, SBLON)....177
2.24.2.6 Inhibition de l'affichage du bloc courant (DISPLOF, DISPLON, ACTBLOCNO)....183
2.24.2.7 Identifier les sous-programmes avec prétraitement (PREPRO)....186
2.24.2.8 Retour de sous-programme M17....187
2.24.2.9 Retour de sous-programme RET....188
2.24.2.10 Retour paramétrable dans les sous-programmes (RET ...)....189
2.24.2.11 Retour paramétrable dans les sous-programmes (RETB...)....195
2.24.3 Appel d'un sous-programme....199
2.24.3.1 Appel de sous-programme sans transfert de paramètres....199
2.24.3.2 Appel d'un sous-programme avec transfert de paramètres (EXTERN)....201
2.24.3.3 Nombre de répétitions de programme (P)....203
2.24.3.4 Appel modal d'un sous-programme (MCALL)....204
2.24.3.5 Appel indirect de sous-programme (CALL)....206
2.24.3.6 Appel indirect d'un sous-programme avec indication de la section de programme à exécuter (CALL BLOCK ... TO ...)....207
2.24.3.7 Appel indirect d'un programme programmé en langage ISO (ISOCALL)....209
2.24.3.8 Appel de sous-programme avec chemin d'accès et paramètres (PCALL)....209
2.24.3.9 Extension du chemin de recherche pour l'appel de sous-programmes (CALLPATH)....210
2.24.3.10 Exécution de sous-programme externe (840D sl) (EXTCALL)....211
2.24.3.11 Exécution d'un sous-programme externe (828D) (EXTCALL)....215
2.25 Macroprogrammation (DEFINE ... AS)....220
3 Gestion des fichiers et programmes....223
3.1 Mémoire de programmes....223
3.1.1 Mémoire programme CN....223
3.1.2 Mémoires de programmes externes....225
3.1.3 Adressage des fichiers de la mémoire de programmes....227
3.1.4 Chemin de recherche pour l'appel de sous-programmes....232
3.1.5 Interrogation du chemin et du nom de fichier....233
3.2 Mémoire de travail (CHANDATA, COMPLETE, INITIAL)....235
4 Zones de protection....239
4.1 Définition des zones de protection (CPROTDEF, NPROTDEF)....239
4.2 Activation/désactivation des zones de protection (CPROT, NPROT)....244
4.3 Contrôle des dépassements de zone de protection, des limitations de la zone de travail et des fins de course logicielles (CALCPOSI)....248

5 Instructions de déplacement spéciales....259

5.1 Accostage de positions codées (CAC, CIC, CDC, CACP, CACN)....259

5.2 Interpolation de type spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)....260

5.3 Groupe spline (SPLINEPATH)....271

5.4 Activer/désactiver le compactage de bloc CN (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)....273

5.5 Interpolation polynomiale (POLY, POLYPATH, PO, PL)....274

5.6 Référence de trajectoire réglable (SPATH, UPATH)....280

5.7 Mesure avec palpeur à déclenchement (MEAS, MEAW)....282

5.8 Mesure spécifique à l'axe (MEASA, MEAWA, MEAC) (option)....285

5.9 Fonctions spéciales pour l'utilisateur OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)....296

5.10 Réduction de l'avance avec décélération aux angles (FENDNORM, G62, G621) 297

5.11 Critère programmable de fin de déplacement (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)....298

6 Transformations des coordonnées (frames)....301

6.1 Transformation de coordonnées par variables frames....301

6.1.1 Variable frame prédéfinie ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)....303

6.2 Affectations de valeurs aux frames....307

6.2.1 Affectation de valeurs directes (valeur d'axe, angle, échelle)....307

6.2.2 Lecture et modification de composantes de frame (TR, FI, RT, SC, MI)....309

6.2.3 Calcul avec des frames....310

6.2.4 Définition des variables frame (DEF FRAME)....312

6.3 Décalage grossier et fin (CTRANS, CFINE)....313

6.4 Décalage d'origine externe ($AA_ETRANS)....315

6.5 Forçage de valeurs réelles avec perte de l'état de référencement (PRESETON)....317

6.6 Forçage de valeurs réelles sans perte de l'état de référencement (PRESETONS)....319

6.7 Calcul du frame à partir de 3 points mesurés dans l'espace (MEAFRAME)....321

6.8 Frames à définition globale pour NCU....325

6.8.1 Frames spécifiques à un canal ($P_CHBFR, $P_UBFR)....326

6.8.2 Frames actifs dans un canal....326

7 Transformations....333

7.1 Programmation générale des types de transformation....333

7.1.1 Mouvements d'orientation lors des transformations....335

7.1.2 Aperçu de la transformation d'orientation TRAORI....339

7.2 Transformation trois, quatre et cinq axes (TRAORI)....341

7.2.1 Corrélations générales de tête de fraisage de type cardan....341

7.2.2 Transformation trois, quatre et cinq axes (TRAORI)....344

7.2.3 Variantes de programmation d'orientation et position initiale (ORIRESET)....345
7.2.4 Programmation de l'orientation de l'outil (A..., B..., C..., LEAD, TILT)....347
7.2.5 Fraisage en bout (A4, B4, C4, A5, B5, C5)....353
7.2.6 Référence des axes d'orientation (ORIWKS, ORIMKS)....354
7.2.7 Programmation des axes d'orientation (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)....356
7.2.8 Programmation de l'orientation le long d'une enveloppe de cône (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)....358
7.2.9 Définition de l'orientation de deux points de contact (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)....362
7.3 Polynômes d'orientation (PO[angles], PO[coordonnée])....364
7.4 Rotations de l'orientation de l'outil (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)....366
7.5 Orientations par rapport à la trajectoire....369
7.5.1 Types d'orientation par rapport à la trajectoire....369
7.5.2 Rotation de l'orientation de l'outil relative à la trajectoire (ORIPATH, ORIPATHS, angle de rotation)....370
7.5.3 Interpolation relative à la trajectoire de la rotation de l'outil (ORIROTC, THETA)....372
7.5.4 Lissage du tracé d'orientation (ORIPATHS A8=, B8=, C8=)....374
7.6 Compactage de l'orientation (COMPON, COMPCURV, COMPCAD, COMPSURF)....376
7.7 Activation/désactivation du lissage du tracé d'orientation (ORISON, ORISOF)....379
7.8 Transformation cinématique....381
7.8.1 Activer la transformation de surfaces d'attaque (TRANSMIT)....381
7.8.2 Activer la transformation de surfaces cylindriques (TRACYL)....381
7.8.3 Activation de la transformation à angle oblique avec angle programmable (TRAANG). 384
7.8.4 Plongée oblique sur les rectifieuses (G5, G7)....385
7.9 Activer la transformation concaténée (TRACON)....387
7.10 Déplacement PTP cartésien....389
7.10.1 Activer/désactiver le déplacement PTP cartésien (PTP, PTPG0, PTPWOC, CP)....389
7.10.2 Indiquer la position des articulations (STAT)....390
7.10.3 Indiquer les signes des angles d'axe (TU)....394
7.10.4 Exemple 1 : Déplacement PTP d'un robot à 6 axes avec la transformation ROBX....397
7.10.5 Exemple 2 : Déplacement PTP avec une transformation générique 5 axes....398
7.10.6 Exemple 3 : PTPG0 et TRANSMIT....398
7.11 Conditions marginales pour l'activation d'une transformation....400
7.12 Désactivation de la transformation (TRAFOOF)....401
8 Chaînes cinématiques....403
8.1 Suppression de composants (DELOBJ)....403
8.2 Détermination de l'indice par le nom (NAMETOINT)....407
9 Prévention des collisions avec chaînes cinématiques....409
9.1 Contrôle de la présence de paire de collision (COLLPAIR)....410
9.2 Demander un nouveau calcul du modèle de machine pour la prévention des collisions (PROTA)....412
9.3 Réglage de l'état des zone de protection (PROTS)....413

9.4 Détermination de la distance entre deux zones de protection (PROTD)....414

10 Transformations avec chaînes cinématiques....417

10.1 Activation de la transformation (TRAFOON)....417

10.2 Modification de la transformation d'orientation après mesurage de la machine (CORRTRAFO)....419

11 Corrections d'outils....427

11.1 Mémoire de correcteurs....427

11.2 Corrections additives....430
11.2.1 Sélection des corrections additives (DL)....430
11.2.2 Définition des valeurs d'usure et de réglage ($TC_SCPxy[t,d], $TC_ECPxy[t,d])....431
11.2.3 Effacer les corrections additives (DELDL)....432

11.3 Correcteur d'outil : Interventions spéciales....434
11.3.1 Application de la fonction miroir aux longueurs d'outil....436
11.3.2 Exploitation du signe de l'usure....436
11.3.3 Système de coordonnées de l'usinage actif (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)....437
11.3.4 Longueurs d'outil et changement de plan....440

11.4 Correction d'outil en ligne....442
11.4.1 Définir une fonction polynôme (FCTDEF)....442
11.4.2 Ecriture en continu de la correction d'outil en ligne (PUTFTOCF)....443
11.4.3 Ecriture discrète de la correction d'outil en ligne (PUTFTOC)....444
11.4.4 Activer/désactiver la correction d'outil en ligne (FTOCON/FTOCOF)....445

11.5 Correction du rayon d'outil 3D....447
11.5.1 Sélection de la correction 3D du rayon d'outil pour le fraisage périphérique 3D (CUT3DC, CUT3DCD, ISD)....447
11.5.2 Sélection de la correction 3D du rayon d'outil pour le fraisage en bout 3D (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)....451
11.5.3 Fraisage périphérique 3D avec prise en compte d'une surface de délimitation (CUT3DCC, CUT3DCCD)....457

11.6 Orientation de l'outil (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)....462

11.7 Numéros D libres, numéro de tranchant....468
11.7.1 Numéros D libres, numéro de tranchant (adresse CE)....468
11.7.2 Libre affectation des numéros D : contrôle des numéros D (CHKDNO)....468
11.7.3 Libre affectation des numéros D : modification des numéros D (GETDNO, SETDNO)....469
11.7.4 Libre affectation des numéros D : détermination du numéro T correspondant au numéro D prescrit (GETACTTD)....470
11.7.5 Libre affectation des numéros D : numéros D déclarés non valides (DZERO)....470

11.8 Cinématique de l'organe porte-outil....472

11.9 Correction de longueur d'outil pour organes porte-outils orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)....477

11.10 Correction en ligne de la longueur d'outil (TOFFON, TOFFOF)....480

11.11 Modification des données de correction pour outils orientables....483
11.11.1 Calcul de l'orientation (ORISOLH)....483

11.11.2 Activation de la modification des données de correction pour outils orientables (CUTMOD, CUTMODK)....493
11.12 Utilisation des environnements d'outils....501
11.12.1 Enregistrement de l'environnement d'outil (TOOLENV)....501
11.12.2 Suppression de l'environnement d'outil (DELTOOLENV)....504
11.12.3 Lecture des numéros T, D et DL (GETTENV)....505
11.12.4 Lecture des informations relatives aux environnements d'outil archivés ($P_TOOLENVN, ($P_TOOLENV)....506
11.12.5 Lecture de longueurs d'outil ou de composantes de longueur d'outil (GETTCOR)....507
11.12.6 Modification des composantes d'outil (SETTCOR)....514
11.13 Lecture de l'affectation des longueurs d'outil L1, L2, L3 par rapport aux axes de coordonnées (LENTOAX)....528
12 Modes de déplacement....531
12.1 Positionnement tangentiel....531
12.1.1 Définir le couplage (TANG)....531
12.1.2 Activer la génération de bloc intermédiaire (TLIFT)....533
12.1.3 Activer le couplage (TANGON)....534
12.1.4 Désactiver le couplage (TANGOF)....535
12.1.5 Supprimer un couplage (TANGDEL)....536
12.2 Variation de l'avance (FNORM, FLIN, FCUB, FPO)....538
12.3 Comportement à l'accélération....543
12.3.1 Mode d'accélération (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)....543
12.3.2 Influence de l'accélération dans le cas des axes asservis (VELOLIMA, ACCLIMA, JERKLIMA)....545
12.3.3 Activation de valeurs dynamiques spécifiques à la technologie (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)....547
12.4 Déplacement avec commande anticipatrice (FFWON, FFWOF)....549
12.5 Précision de contour programmable (CPRECON, CPRECOF)....550
12.6 Exécution du programme avec une mémoire tampon (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)....552
12.7 Définition de sections d'arrêt temporisé (DELAYFSTON, DELAYFSTOF)....555
12.8 Inhibition d'une partie de programme pour SERUPRO (IPTRLOCK, IPTRUNLOCK)....558
12.9 Réaccostage du contour (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)....561
12.10 Correction du pilotage des déplacements....570
12.10.1 Correction de l'à-coup en pourcentage (JERKLIM)....570
12.10.2 Correction de la vitesse en pourcentage (VELOLIM)....571
12.10.3 Exemple de programme pour JERKLIM et VELOLIM....573
12.11 Programmation d'une tolérance de contour/orientation (CTOL, OTOL, ATOL)....574
12.12 Mode de changement de bloc pour couplage actif (CPBC)....579
13 couplages d'axes....581
13.1 Déplacements conjugués (TRAILON, TRAILOF)....581
13.2 Tables de courbes (CTAB)....586

13.2.1 Définition de tables de courbes (CTABDEF, CATBEND)....586
13.2.2 Vérification de l'existence d'une table de courbes (CTABEXISTS)....593
13.2.3 Suppression d'une table de courbes (CTABDEL)....593
13.2.4 Verrouillage de tables de courbes contre la suppression et l'écrasement (CTABLOCK, CTABUNLOCK)....594
13.2.5 Tables de courbes : détermination des propriétés de la table (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)....596
13.2.6 Lecture des valeurs de la table de courbes (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)....597
13.2.7 Tables de courbes : Vérification de l'utilisation des ressources (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)....602

13.3 Couplage de deux axes par valeur pilote (LEADON, LEADOF)....604

13.4 Réducteur électronique (EG)....610
13.4.1 Définition d'un réducteur électronique (EGDEF)....610
13.4.2 Activation du réducteur électronique (EGON, EGONSYN, EGONSYNE)....611
13.4.3 Désactivation du réducteur électronique (EGOFS, EGOFC)....614
13.4.4 Suppression de la définition d'un réducteur électronique (EGDEL)....615
13.4.5 Avance par tour (G95) / réducteur électronique (FPR)....616

13.5 Broche synchrone....617
13.5.1 Broche synchrone : Programmation (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)....617

13.6 Couplage générique (CP...)....628

13.7 Couplage maître/esclave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)....636

14 Actions synchrones....641

14.1 Définition d'une action synchrone....641

15 Oscillation....643

15.1 Oscillation asynchrone (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)....643

15.2 Oscillation commandée par des actions synchrones (OSCILL)....648

16 Poinçonnage et grignotage....657

16.1 Activation/désactivation....657
16.1.1 Activation/désactivation du poinçonnage et du grignotage (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)....657

16.2 Segmentation automatique des déplacements....662
16.2.1 Segmentation des déplacements pour axes à interpolation....665
16.2.2 Segmentation du déplacement dans le cas d'axes individuels....666

17 Rectification....669

17.1 Activer/désactiver la surveillance d'outil spécifique à la rectification (TMON, TMOF)....669

18 Autres fonctions....671

18.1 Fonctions d'axe (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)....671

18.2 Axes géométriques permutables (GEOAX)....674

18.3 Conteneur d'axes (AXCTSWE, AXCTSWED, AXCTSWEC)....679

18.4 Attente de la position d'axe définitive (WAITENC)....681
18.5 Commutation de jeu de paramètres programmable (SCPARA)....683
18.6 Contrôler le langage CN existant (STRINGIS)....685
18.7 Appel interactif des fenêtres depuis le programme pièce (MMC)....689
18.8 Temps d'exécution du programme / Compteur de pièces....694
18.8.1 Temps d'exécution de programme....694
18.8.2 Compteurs de pièces....698
18.9 Process DataShare - Sortie sur un appareil/fichier externe (EXTOPEN, WRITE, EXTCLOSE)....700
18.10 Alarmes (SETAL)....705
18.11 Arrêt étendu et retrait (ESR)....707
18.11.1 ESR assisté par CN....708
18.11.1.1 Retrait assisté par CN (POLF, POLFA, POLFMASK, POLFMLIN)....708
18.11.1.2 Arrêt assisté par CN....712
18.11.2 ESR contrôlé par le variateur....713
18.11.2.1 Configuration d'un arrêt contrôlé par le variateur (ESRS)....713
18.11.2.2 Configuration d'un retrait contrôlé par le variateur (ESRR)....713
18.12 Définition d'une pièce brute (WORKPIECE)....715
18.13 Basculement du mode de langage (G290, G291)....719
19 Programmes de chariotage personnalisés....721
19.1 Fonctions additionnelles pour le chariotage....721
19.2 Création d'une table de contour (CONTPRON)....722
19.3 Création d'une table de contour codée (CONTDCON)....728
19.4 Détermination de l'intersection entre deux éléments de contour (INTERSEC)....732
19.5 Exécution des éléments de contour d'une table bloc par bloc (EXECTAB)....734
19.6 Calcul de données de cercles (CALCDAT)....735
19.7 Désactivation de la préparation du contour (EXECUTE)....737
20 Programmation externe de cycles....739
20.1 Cycles technologiques....739
20.1.1 Introduction....739
20.1.2 Vue d'ensemble spécifique à la technologie....740
20.1.3 HOLES1 - Rangée de trous....742
20.1.4 HOLES2 - Cercle de trous....742
20.1.5 POCKET3 - Fraisage d'une poche rectangulaire....744
20.1.6 POCKET4 - Fraisage d'une poche circulaire....747
20.1.7 SLOT1 - Rainure rectiligne....750
20.1.8 SLOT2 - Rainure sur cercle....752
20.1.9 LONGHOLE - Trou oblong....755
20.1.10 CYCLE60 - Cycle de gravure....757
20.1.11 CYCLE61 - Surfaçage....759
20.1.12 CYCLE62 - Appel de contour....761
20.1.13 CYCLE63 - Fraisage de poche de contour....762

20.1.14 CYCLE64 - Perçage ébauche d'une poche de contour....765
20.1.15 CYCLE70 - Fraisage de filetage....767
20.1.16 CYCLE72 - Fraisage en contournage....768
20.1.17 CYCLE76 - Fraisage d'un tourillon rectangulaire....773
20.1.18 CYCLE77 - Fraisage d'un tourillon circulaire....775
20.1.19 CYCLE78 - Fraisage de filetages....777
20.1.20 CYCLE79 - Polyèdre....780
20.1.21 CYCLE81 - Perçage, centrage....782
20.1.22 CYCLE82 - Perçage, lamage....783
20.1.23 CYCLE83 - Perçage profond....785
20.1.24 CYCLE84 - Taraudage sans porte-taraud compensateur....788
20.1.25 CYCLE85 - Alésage à l'alésoir....791
20.1.26 CYCLE86 - Alésage au tour....792
20.1.27 CYCLE92 - Tronçonnage....794
20.1.28 CYCLE95 - Chariotage au contour....795
20.1.29 CYCLE98 - Concaténation de filetages....797
20.1.30 CYCLE99 - Filetage au tour....801
20.1.31 CYCLE435 - Définir le système de coordonnées du dresseur....806
20.1.32 CYCLE495 - Profilage....806
20.1.33 CYCLE800 - Pivotement....808
20.1.34 CYCLE801 - Réseau ou cadre....812
20.1.35 CYCLE802 - Positions quelconques....813
20.1.36 CYCLE830 - Perçage profond 2....816
20.1.37 CYCLE832 - Réglages High Speed....822
20.1.38 CYCLE840 - Taraudage avec porte-taraud compensateur....826
20.1.39 CYCLE899 - Fraisage de rainure borgne....828
20.1.40 CYCLE930 - Gorge....831
20.1.41 CYCLE940 - Formes de dégagement....834
20.1.42 CYCLE951 - Chariotage....837
20.1.43 CYCLE952 - Plongée de contour....840
20.1.44 CYCLE4071 - Rectification longitudinale avec pénétration au point d'inversion....846
20.1.45 CYCLE4072 - Rectification longitudinale avec pénétration au point d'inversion et signal d'abandon....847
20.1.46 CYCLE4073 - Rectification longitudinale avec pénétration continue....851
20.1.47 CYCLE4074 - Rectification longitudinale avec pénétration continue et signal d'abandon....852
20.1.48 CYCLE4075 - Rectification plane avec pénétration au point d'inversion....855
20.1.49 CYCLE4077 - Rectification plane avec pénétration au point d'inversion et signal d'abandon....858
20.1.50 CYCLE4078 - Rectification plane avec pénétration continue....861
20.1.51 CYCLE4079 - Rectification plane avec pénétration intermittente....863
20.1.52 GROUP_BEGIN - Début du bloc de programme....865
20.1.53 GROUP_END - Fin du bloc de programme....866
20.1.54 GROUP_ADDEND - Fin du supplément de mise au point....866
20.1.55 Conditions marginales....867
20.1.55.1 Mise à l'échelle de la technologie dans les masques de cycle....867
20.2 Cycles de mesure....869
21 Tableaux....871
21.1 Instructions....871
21.2 Instructions : Disponibilité pour SINUMERIK 828D....912
21.2.1 Variantes de commande fraisage/tournage....912

21.2.2 Variantes de commande rectification....940
21.3 Langue actuelle dans l'IHM....968
A Annexe....969
A.1 Liste des abréviations....969
A.2 Vue d'ensemble de la documentation....978
Glossaire....979
Index....1003

Consignes de sécurité élémentaires

1

1.1 Consignes de sécurité générales

⚠ ATTENTION

Le non respect des consignes de sécurité et le manque de prise en compte des risques résiduels peuvent entraîner la mort

Le non respect des consignes de sécurité et des remarques relatives aux risques résiduels dans la documentation du matériel peut conduire à des accidents susceptibles d'entraîner la mort ou de causer des blessures graves.

- Respecter les consignes de sécurité figurant dans la documentation du matériel.
- Tenir compte des risques résiduels pour l'évaluation des risques.

⚠ ATTENTION

Danger de mort lié à des dysfonctionnements de la machine suite à un paramétrage incorrect ou modifié

Un paramétrage incorrect ou modifié peut entraîner des dysfonctionnements sur les machines, susceptibles de provoquer des blessures, voire la mort.

- Protéger les paramètres contre l'accès non autorisé.
- Prendre les mesures appropriées pour palier aux défauts éventuels (p. ex. un arrêt ou une coupure d'urgence).

1.2 Garantie et responsabilité pour les exemples d'application

Les exemples d'application sont sans engagement et n'ont aucune prétention d'exhaustivité concernant la configuration, les équipements et les éventualités de toutes sortes. Les exemples d'application ne constituent pas des solutions client spécifiques, mais ont uniquement pour objet d'apporter une aide dans la résolution de problèmes typiques.

L'utilisateur est seul responsable de la mise en œuvre des produits selon les règles de l'art. Les exemples d'application ne vous dispensent pas des obligations de précaution lors de l'utilisation, de l'installation, de l'exploitation et de la maintenance.

1.3 Sécurité industrielle

Remarque

Sécurité industrielle

Siemens commercialise des produits et solutions comprenant des fonctions de sécurité industrielle (Industrial Security) qui contribuent à une exploitation sûre des installations, systèmes, machines et réseaux.

Pour garantir la sécurité des installations, systèmes, machines et réseaux contre les cybermenaces, il est nécessaire d'implémenter (et de préserver) un concept de sécurité industrielle global et moderne. Les produits et solutions de Siemens ne constituent qu'une partie d'un tel concept.

Il incombe au client d'empêcher tout accès non autorisé à ses installations, systèmes, machines et réseaux. Les systèmes, machines et composants doivent uniquement être connectés au réseau d'entreprise ou à Internet si et dans la mesure où c'est nécessaire et si des mesures de protection correspondantes (p. ex. utilisation de pare-feu et segmentation du réseau) ont été prises.

En outre, les recommandations de Siemens sur les mesures de protection correspondantes doivent être respectées. Plus d'informations sur la sécurité industrielle, voir :

Sécurité industrielle (http://www.siemens.com/industrialsecurity)

Les produits et solutions Siemens font l'objet de développements continus pour être encore plus sûrs. Siemens vous recommande donc vivement d'effectuer des actualisations dès que les mises à jour correspondantes sont disponibles et de ne toujours utiliser que les versions de produit actuelles. L'utilisation de versions obsolètes ou qui ne sont plus prises en charge peut augmenter le risque de cybermenaces.

Pour être informé sur les mises à jour produit dès leur sortie, s'abonner au flux RSS Siemens Industrial Security sur :

Sécurité industrielle (http://www.siemens.com/industrialsecurity)

Plus d'informations, voir sur Internet :

Manuel de configuration Industrial Security (https://support.industry.siemens.com/cs/ww/fr/view/108862708/en)

⚠ ATTENTION

États de fonctionnement non sûrs suite à une manipulation du logiciel

Les manipulations des logiciels (p. ex. les virus, chevaux de Troie, logiciels malveillants, vers) peuvent provoquer des états de fonctionnement non sûrs de l'installation, susceptibles d'e causer la mort, des blessures graves et des dommages matériels.

- Les logiciels doivent être maintenus à jour.
- Intégrer les composants d'entraînement et d'automatisation dans un concept global de sécurité industrielle (Industrial Security) de l'installation ou de la machine selon l'état actuel de la technique.
- Tenir compte de tous les produits utilisés dans le système global de sécurité industrielle (Industrial Security).
- Il convient de protéger les données stockées sur les supports de mémoire amovibles contre les logiciels nuisibles avec les mesures de protection appropriées, par exemple avec un antivirus.
- Protéger le mécanisme d'entraînement contre toute modification non autorisée en activant la fonction variateur "Protection de savoir-faire".

Programmation CN flexible 2

2.1 Variables

L'utilisation de variables des domaines Données système et Données utilisateur, en particulier en association avec des fonctions de calcul et des structures de contrôle, permet de réaliser des programmes CN et des cycles de façon flexible.

- Données système
  Les données système contiennent les variables prédéfinies dans le système. Ces variables ont une signification définie. Elles sont utilisées avant tout par le logiciel système.
  L'utilisateur peut lire et écrire ces variables dans les programmes CN et les cycles.
  Exemple : paramètres machine, données de réglage, variables système.
  Bien que la signification d'une donnée système soit prédéfinie et fixe, les propriétés peuvent être modifiées dans une certaine mesure par l'utilisateur via une redéfinition.
  Voir "Redéfinition de données système, données utilisateur et instructions de langage CN (REDEF) (Page 37)"
- Données utilisateur
  Les données utilisateur contiennent les variables définies par l'utilisateur, dont la signification est établie exclusivement par l'utilisateur. Elles ne sont pas exploitées par le système.
  Les données utilisateur sont réparties en :
  - Variables utilisateur prédéfinies
    Il s'agit de variables déjà définies dans le système, dont le nombre est paramétré au moyen des paramètres machine. L'utilisateur a la possibilité d'adapter les propriétés de ces variables.
    Voir "Redéfinition de données système, données utilisateur et instructions de langage CN (REDEF) (Page 37)".
  - Variables définies par l'utilisateur
    Il s'agit de variables définies par l'utilisateur et que le système crée lors de l'exécution. Leur nombre, type de données, visibilité ainsi que toutes les autres propriétés sont définies exclusivement par l'utilisateur.
    Voir "Définition de variables utilisateur (DEF) (Page 31)"

2.1.1 Données système

Les données système contiennent les variables prédéfinies dans le système, qui permettent d'accéder, dans les programmes CN et les cycles, au paramétrage actuel de la commande, ainsi qu'aux états de la machine, de la commande et du processus.

Variables de prétraitement

Il s'agit de données système qui sont lues et écrites dans le contexte du prétraitement, autrement dit au moment de l'interprétation du bloc dans lequel la variable est programmée. Les variables de prétraitement ne déclenchent pas d'arrêt de prétraitement.

Variables d'exécution

Il s'agit de données système qui sont lues ou écrites dans le contexte de l'exécution, autrement dit au moment de l'exécution du bloc dans lequel la variable est programmée. Les variables d'exécution sont des :

- Variables pouvant être programmées dans des actions synchrones (lecture/écriture)
- Variables pouvant être programmées dans le programme CN et capables de déclencher un arrêt de prétraitement (lecture/écriture)
- Variables pouvant être programmées dans le programme CN, et dont la valeur est déterminée durant le prétraitement mais n'est écrite que lors de l'exécution (synchrones à l'exécution : écriture seule).

Systématique de préfixe

Pour l'identification spécifique des données système, le nom est normalement précédé d'un préfixe formé par le caractère $, suivi d'une ou de deux lettres et d'un caractère de soulignement :

$ + 1ère Lettre	**Signification : type de donnée**
Données de prétraitement (Données système qui sont lues / écrites durant le prétraitement des blocs)	
$M	Paramètres machine[1)]
$S	Données de réglage, zones de protection [1)]
$T	Paramètres de gestion des outils
$P	Valeurs programmées
$C	Variables des cycles enveloppes ISO
$O	Données optionnelles
R	Paramètres R (paramètres de calcul)[2)]
Données d'exécution (données système qui sont lues / écrites durant l'exécution des blocs)	
$$M	Paramètres machine[1)]
$$S	Données de réglage [1)]
$A	Données d'exécution courantes
$V	Données du régulateur de position
$R	Paramètres R (paramètres de calcul)[2)]
[1)] Selon que les paramètres machine et les données de réglage sont précédés d'un ou de deux caractères $, ils sont traités comme des variables de prétraitement ou d'exécution. Le choix de la notation est libre et spécifique à l'application. [2)] Si un paramètre R est utilisé en tant que variable d'exécution dans le programme pièce / le cycle, aucun préfixe n'est spécifié, p. ex. R10. Lors de l'utilisation en tant que variable d'exécution dans une action synchrone, le préfixe est formé d'un caractère $, p. ex. $R10.	

2ème caractère	**Signification : Visibilité**
N	Variable globale CN (**C**N)
C	Variable spécifique au canal (**C**hannel)
A	Variable spécifique à l'axe (**A**xis)

Autres conditions à prendre en compte

Exceptions dans la systématique de préfixe

Les variables système ci-après s'écartent de la systématique de préfixe indiquée plus haut :

- $TC_... : Le deuxième caractère C ne désigne pas une variable spécifique au canal, mais une variable système spécifique au porte-outil (TC = Tool Carrier)
- $P_ ... : Variables système spécifiques à un canal

Utilisation de paramètres machine et de données de réglage dans les actions synchrones

Lors de l'utilisation de paramètres machine et de données de réglage dans les actions synchrones, il est possible de déterminer, au moyen du préfixe, si le paramètre machine ou la donnée de réglage sont écrits/lus de façon synchrone au prétraitement ou à l'exécution.

Si le paramètre reste inchangé pendant l'exécution, la lecture peut être effectuée de façon synchrone au prétraitement. Le préfixe du paramètre machine ou de la donnée de réglage s'écrit de ce fait avec un caractère $:

```
ID=1 WHENEVER $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Si le paramètre est modifié pendant l'exécution, la lecture /écriture doit être effectuée de façon synchrone à l'exécution. Le préfixe du paramètre machine ou de la donnée de réglage s'écrit de ce fait avec deux caractères $:

```
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Remarque

Ecriture des paramètres machine et des données de réglage

Lors de l'écriture d'un paramètre machine ou d'une donnée de réglage, il faut s'assurer que le niveau d'accès actif pour l'exécution du programme pièce / cycle autorise la protection en écriture et que la prise d'effet du paramètre soit "IMMEDIATE".

Bibliographie

Pour obtenir une vue d'ensemble complète de toutes les variables système, référez-vous aux :

Tables de paramètres Variables système

Voir aussi

Variables (Page 23)

2.1.2 Variables utilisateur prédéfinies : Paramètre de calcul

2.1.2.1 Paramètres de calcul spécifiques à un canal (R)

Les paramètres de calcul ou paramètres R spécifiques de canal sont des variables utilisateur prédéfinies désignées par R et définies comme tableau de type de données REAL. Pour des raisons historiques, outre la notation des paramètres R avec un indice de tableau, p. ex. `R[10]`, la notation sans indice de tableau, par ex. `R10` est également autorisée.

En cas d'utilisation dans des actions synchrones, la lettre $ doit figurer en tête, p. ex. `$R10`.

Syntaxe

En cas d'utilisation comme variable de prétraitement :
```
R<n>
R[<expression>]
```

En cas d'utilisation comme variable d'exécution :
```
$R<n>
$R[<expression>]
```

Signification

`R` :	Descripteur en cas d'utilisation comme variable de prétraitement, p. ex. dans le programme pièce	
`$R` :	Descripteur en cas d'utilisation comme variable d'exécution, p. ex. dans des actions synchrones	
	Type :	REAL
	Plage de valeurs :	En notation non exponentielle : ± (0,000 0001 ... 9999 9999) **Remarque :** Le nombre maximum de décimales autorisées est de 8
		En notation exponentielle : ± ($1*10^{-300}$ ... $1*10^{+300}$) **Remarque :** • Notation : <Mantisse>EX<exponent> par ex. 8.2EX-3 • Le nombre maximum de caractères autorisés, y compris le signe et le point décimal, est de 10.
`<n>` :	Numéro du paramètre R	
	Type :	INT
	Plage de valeurs :	0 - MAX_INDEX **Remarque** MAX_INDEX résulte du nombre de paramètres R paramétrés : MAX_INDEX = (PM28050 $MN_MM_NUM_R_PARAM) - 1
`<expression>` :	Indice de tableau L'indice de tableau peut être une expression quelconque, à condition que le résultat de cette expression puisse être converti dans le type de données INT (INT, REAL, BOOL, CHAR)	

Exemple

Affectations de paramètres R et utilisation de paramètres R dans des fonctions mathématiques :

Code de programme	Commentaire
R0=3.5678	; Affectation dans le prétraitement des blocs
R[1]=-37.3	; Affectation dans le prétraitement des blocs
R3=-7	; Affectation dans le prétraitement des blocs
$R4=-0.1EX-5	; Affectation dans l'exécution des blocs : R4 = -0.1 * 10^-5
$R[6]=1.874EX8	; Affectation dans l'exécution des blocs : R6 = 1.874 * 10^8
R7=SIN(25.3)	; Affectation dans le prétraitement des blocs
R[R2]=R10	; Adressage indirect au moyen d'un paramètre R
R[(R1+R2)*R3]=5	; Adressage indirect au moyen d'une expression mathématique
X=(R1+R2)	; Déplacer l'axe X à la position résultant de la somme de R1 et de R2
Z=SQRT(R1*R1+R2*R2)	; Déplacer l'axe Z à la position racine carrée (R1^2 + R2^2)

Voir aussi

Variables (Page 23)

2.1.2.2 Paramètres de calcul globaux (RG)

Fonction

En plus des paramètres R spécifiques de canal, l'utilisateur disposent également de paramètres R globaux. Ils existent une fois au sein de la commande et peuvent être lus/écrits depuis tous les canaux.

Les paramètres globaux R sont, par exemple, utilisés pour transférer des informations d'un canal dans un autre. Les réglages globaux qui doivent être exploités pour tous les canaux en sont un autre exemple, comme la saillie de la pièce brute hors de la broche.

La lecture et l'écriture des paramètres R globaux s'effectuent via l'interface utilisateur ou dans le programme CN lors du prétraitement. Une utilisation dans des actions synchrones ou des cycles technologiques n'est pas possible.

Remarque

Lors de la lecture et de l'écriture des paramètres R globaux, **aucune** synchronisation n'intervient entre les canaux.

La lecture et l'écriture intervenant lors du prétraitement, l'instant auquel une valeur écrite issue d'un canal prend effet dans un autre canal n'est pas défini.

Exemple :

dans le canal 1, une boucle tourne avec un paramètre R global comme compteur de cycles. Le canal 2 écrit, dans ce paramètre R global, une valeur qui entraîne une interruption de la boucle dans le canal 1. Toutes les boucles interprétées jusqu'alors dans le canal 1 lors du prétraitement sont cependant toujours exécutées. Le nombre de boucles que cela représente n'est pas défini et dépend, entre autres, de l'utilisation du canal.

La synchronisation entre les canaux doit être réalisée de manière applicative par l'utilisateur lui-même, par exemple à l'aide de repères WAIT !

Syntaxe

Ecriture dans le programme CN

```
RG[<n>]=<valeur>
RG[<expression>]=<valeur>
```

Lecture dans le programme CN

```
R...=RG[<n>]
R...=RG[<expression>]
```

Signification

`RG` :	Nom par défaut de l'adresse CN pour paramètres R **globaux** **Remarque :** Le nom de l'adresse CN peut être réglé par le biais de PM15800 $MN_R_PARAM_NCK_NAME	
`<n>` :	Numéro du paramètre R global	
	Type :	INT
	Plage de valeurs :	0 ... MAX_INDEX **Remarque** MAX_INDEX résulte du nombre de paramètres R globaux paramétrés : MAX_INDEX = (PM18156 $MN_MM_NUM_R_PARAM_NCK) - 1
`<expression>` :	L'indice de tableau peut être une expression quelconque, à condition que le résultat de cette expression puisse être converti dans le type de données INT (INT, REAL, BOOL, CHAR)	

<valeur> :	Valeur du paramètre R global	
	Type :	REAL
	Plage de valeurs :	En notation non exponentielle : ± (0,000 0001 ... 9999 9999) **Remarque :** Le nombre maximum de décimales autorisées est de 8
		En notation exponentielle : ± ($1*10^{-300}$ ... $1*10^{+300}$) **Remarque :** • Notation : <Mantisse>EX<exponent> par ex. 8.2EX-3 • Le nombre maximum de caractères autorisés, y compris le signe et le point décimal, est de 10.

2.1.3 Variables utilisateur prédéfinies : variables Link

Dans le cadre de la fonction "Link NCU", les variables Link permettent de réaliser l'échange cyclique de données entre des NCU mises en réseau. Elle autorisent un accès spécifique au format de données à la mémoire de variables Link. La taille et la structure de données de la mémoire de variables Link sont définies de manière spécifique à l'installation par l'utilisateur / le constructeur de machine.

Les variables Link sont des variables utilisateur globales du système, dont l'écriture et la lecture peuvent être effectuées dans des programmes pièce et des cycles à partir de toutes les NCU du groupe Link, à condition que la communication Link soit configurée. Contrairement aux variables utilisateur globales (GUD), l'utilisation des variables Link est également possible dans des actions synchrones.

Pour des installations ne disposant pas de Link NCU actif, les variables Link peuvent être utilisées localement dans la commande en plus des variables utilisateur globales (GUD), comme variables utilisateur globales supplémentaires.

Syntaxe

```
$A_DLB[<indice>]
$A_DLW[<indice>]
$A_DLD[<indice>]
$A_DLR[<indice>]
```

Signification

$A_DLB :	Variable Link pour le format de données BYTE (1 octet)	
	Type de données :	UINT
	Plage de valeurs :	0 ... 255

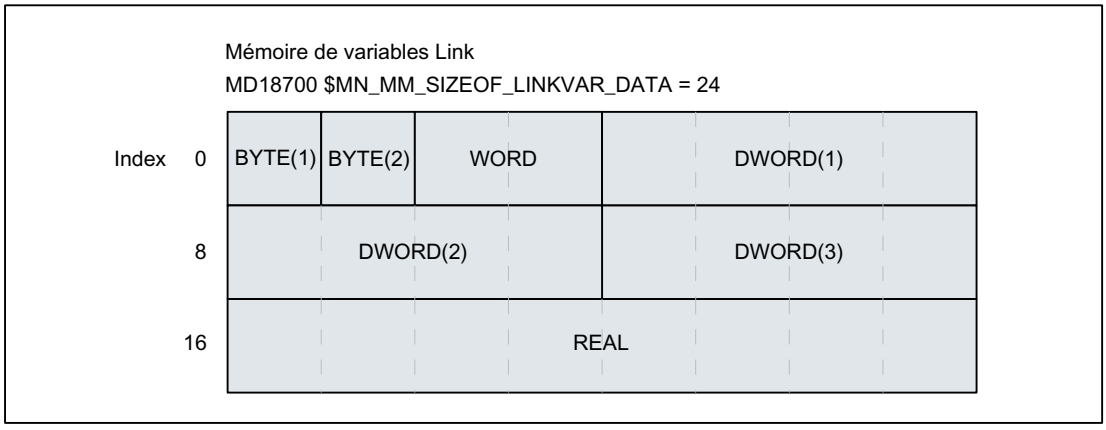
`$A_DLW` :	Variable Link pour le format de données WORD (2 octets)	
	Type de données :	INT
	Plage de valeurs :	-32768 ... 32767
`$A_DLD` :	Variable Link pour le format de données DWORD (4 octets)	
	Type de données :	INT
	Plage de valeurs :	-2147483648 ... 2147483647
`$A_DLR` :	Variable Link pour le format de données REAL (8 octets)	
	Type de données :	REAL
	Plage de valeurs :	$\pm(2{,}2*10^{-308} \ldots 1{,}8*10^{+308})$
`<indice>` :	Indice d'adresse en octets, calculé à partir du début de la mémoire de variables Link	
	Type de données :	INT
	Plage de valeurs :	0 - MAX_INDEX **Remarque** • MAX_INDEX résulte de la taille paramétrée pour la mémoire de variables Link : MAX_INDEX = (PM18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1 • Seule est autorisée la programmation d'indices pour lesquels les octets adressés dans la mémoire de variables Link correspondent à une limite de format de données ⇒ Indice = n * octets, avec n = 0, 1, 2, ... – `$A_DLB[i]` : i = 0, 1, 2, ... – `$A_DLW[i]` : i = 0, 2, 4, ... – `$A_DLD[i]` : i = 0, 4, 8, ... – `$A_DLR[i]` : i = 0, 8, 16, ...

Exemple

L'installation d'automatisation ne contient que 2 NCU (NCU1 et NCU2). L'axe machine AX2 déplacé comme axe Link par NCU2 est raccordé à NCU1.

NCU1 inscrit de façon cyclique la valeur réelle du courant ($VA_CURR) de l'axe AX2 dans la mémoire de variables Link. NCU2 lit de façon cyclique la valeur réelle du courant transmise par la communication Link et affiche l'alarme 61000 en cas de dépassement de la valeur limite.

La structure de données dans la mémoire des variables Link est représentée dans la figure suivante. La valeur réelle du courant est transmise au moyen de la valeur REAL.



NCU1

Durant la période d'appel de l'interpolateur pendant une action synchrone statique, NCU1 écrit de façon cyclique la valeur réelle du courant de l'axe AX2 dans la mémoire des variables Link, au moyen de la variable Link $A_DLR[16].

Code de programme

```
N111 IDS=1 WHENEVER TRUE DO $A_DLR[16]=$VA_CURR[AX2]
```

NCU2

Durant la période d'appel de l'interpolateur pendant une action synchrone statique, NCU2 lit de façon cyclique la valeur réelle du courant de l'axe AX2 à partir de la mémoire des variables Link, au moyen de la variable Link $A_DLR[16]. Si la valeur réelle du courant est supérieure à 23,0 A, l'alarme 61000 s'affiche.

Code de programme

```
N222 IDS=1 WHEN $A_DLR[16] > 23.0 DO SETAL(61000)
```

Voir aussi

Variables (Page 23)

2.1.4 Définition de variables utilisateur (DEF)

L'instruction `DEF` permet de définir des variables spécifiques à l'utilisateur ou des variables utilisateur (User Data) et de leur affecter des valeurs.

Les catégories suivantes de variables utilisateur sont disponibles en fonction du domaine de validité, autrement dit du domaine dans lequel la variable est visible :

- Variables utilisateur locales (LUD)
  Les variables utilisateur locales (LUD) sont des variables définies dans un programme CN qui n'est pas le programme principal au moment de l'exécution. Elles sont déclarées à l'appel du programme CN et supprimées au reset de fin de programme ou au démarrage suivant de la commande. L'accès aux variables LUD peut uniquement être réalisé dans le programme CN dans lequel ces variables sont définies.
- Variables utilisateur globales d'un programme (PUD)
  Les variables utilisateur globales d'un programme (PUD) sont des variables définies dans un programme CN utilisé comme programme principal. Elles sont déclarées à l'appel du programme CN et supprimées au reset de fin de programme ou au démarrage suivant de la commande. L'accès aux variables PUD peut être réalisé dans le programme principal et dans tous ses sous-programmes.

  Remarque

  Disponibilité des variables utilisateur globales d'un programme (PUD)

  Les variables utilisateur globales définies dans le programme principal (PUD) ne sont également disponibles dans les sous-programmes que si le paramètre machine suivant est mis à 1 :

  PM11120 $MN_LUD_EXTENDED_SCOPE = 1

  Lorsque PM11120 = 0, les variables utilisateur globales définies dans le programme principal sont uniquement disponibles dans le programme principal.

- Variables utilisateur globales (GUD)
  Les variables utilisateur globales (GUD) sont des variables globales d'une CN ou d'un canal, définies dans un bloc de données (SGUD, MGUD, UGUD, GUD4 ... GUD9) et conservées même après un reset de fin de programme ou au démarrage suivant de la commande. Les variables GUD sont accessibles dans tous les programmes CN.

Les variables utilisateur doivent avoir été définies avant leur utilisation (lecture / écriture). Il convient de respecter les règles suivantes :

- Les variables GUD doivent être définies dans un fichier de définition, p. ex. _N_DEF_DIR/ _N_UGUD_DEF.
- Les variables PUD et LUD doivent être définies dans la section de définition d'un programme CN.
- La définition des données doit être effectuée dans un bloc spécifique.
- Un seul type de données peut être utilisé par définition de données.
- Plusieurs variables du même type de données peuvent être définies dans chaque définition de données.

Syntaxe

LUD et PUD

```
DEF <type> <unité physique> <valeurs limites> <nom>[<valeur 1>,
<valeur 2>, <valeur 3>]=<valeur d'initialisation>
```

GUD

```
DEF <domaine> <arrêt prétraitement blocs> <droits d'accès> <classe
de données> <type> <unité physique> <valeurs limites>
<nom>[<valeur_1>, <valeur_2>, <valeur_3>]=<valeur d'initialisation>
```

Signification

DEF :	Instruction de définition de variables utilisateur GUD, PUD, LUD	
<domaine> :	Domaine de validité, uniquement significatif pour GUD :	
	CN :	Variable utilisateur globale dans la CN
	CHAN :	Variable utilisateur globale dans le canal
<arrêt prétraitement blocs> :	Arrêt du prétraitement des blocs, uniquement significatif pour GUD (facultatif)	
	SYNR :	arrêt du prétraitement lors de la lecture
	SYNW :	arrêt du prétraitement lors de l'écriture
	SYNRW :	arrêt du prétraitement lors de la lecture / l'écriture
<droits d'accès> :	Niveau de protection pour la lecture / l'écriture de variables GUD par programme CN ou OPI (facultatif)	
	APRP <niveau de protection> :	Lecture : Programme CN
	APWP <niveau de protection> :	Ecriture : Programme CN
	APRB <niveau de protection> :	Lecture : OPI
	APWB <niveau de protection> :	Ecriture : OPI
	<niveau de protection> :	Plage de valeurs : 0 ... 7
	Voir "Attribut : droits d'accès (APR, APW, APRP, APWP, APRB, APWB) (Page 48)"	
<classe de données> :	Attribution d'une classe de données (SINUMERIK 828D uniquement !)	
	DCM :	Classe de données M (= Manufacturer)
	DCI :	Classe de données I (= Individual)
	DCU :	Classe de données U (= User)
	Voir "Attribut : classe de données (DCM, DCI, DCU) – SINUMERIK 828D uniquement (Page 54)".	
<type> :	Type de données :	
	INT :	valeur entière avec signe
	REAL :	Nombre réel (LONG REAL selon IEEE)
	BOOL :	Valeur booléenne TRUE (1) / FALSE (0)
	CHAR :	Caractère ASCII
	STRING[<Longueur maximale>] :	Chaîne de caractères de longueur définie
	AXIS :	Descripteur d'axe/de broche
	FRAME :	Indications géométriques pour une transformation statique de coordonnées
	Voir "Types de données (Page 62)"	

<unité physique> :	Unité physique (facultative)	
	PHU <unité> :	Unité physique
	Voir "Attribut : unité physique (PHU) (Page 46)"	
<valeurs limites> :	Valeur limite inférieure/supérieure (facultative)	
	LLI <valeur limite> :	Valeur limite inférieure (lower limit)
	ULI <valeur limite> :	Valeur limite supérieure (upper limit)
	Voir "Attribut : valeurs limites (LLI, ULI) (Page 44)"	
<nom> :	Nom de la variable **Remarque** • 31 caractères au maximum • Les deux premiers caractères doivent être une lettre et/ou un caractère de soulignement. • Le caractère "$" est réservé aux variables système et ne doit pas être utilisé.	
[<valeur_1>, <valeur_2>, <valeur_3>] :	Indication de la dimension des tableaux pour des variables de tableau comportant 1 à 3 dimensions au maximum (facultative) Pour l'initialisation de variables de tableau, voir "Définition et initialisation de variables de tableau (DEF, SET, REP) (Page 56)"	
<valeur d'initialisation> :	Valeur d'initialisation (facultative) voir "Attribut : Valeur d'initialisation (Page 41)" Pour l'initialisation de variables de tableau, voir "Définition et initialisation de variables de tableau (DEF, SET, REP) (Page 56)"	

Exemples

Exemple 1 : définitions de variables utilisateur dans le bloc de données pour constructeur de machines

```
Code de programme                                                    Commentaire
%_N_MGUD_DEF                                                         ; Bloc GUD : constructeur de machines
$PATH=/_N_DEF_DIR
DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2
; Description
; Définition de deux variables GUD : STROM_1, STROM_2
; Domaine de validité : pour l'ensemble du canal
; Type de données : REAL
; Arrêt du prétraitement des blocs : non programmé => valeur par défaut = pas d'arrêt du prétraitement
des blocs
; Unité phys. : 24 = [A]
; Valeurs limites : inférieure = 0,0, supérieure = 10,0
; Droits d'accès : non programmés => valeur par défaut = 7 = position 0 du commutateur à clé
; Valeur d'initialisation : non programmée => valeur par défaut = 0,0

DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45
```

Code de programme	Commentaire

```
; Description
; Définition de deux variables GUD : ZEIT_1, ZEIT_2
; Domaine de validité : global dans la CN
; Type de données : REAL
; Arrêt du prétraitement des blocs : non programmé => valeur par défaut = pas d'arrêt du prétraitement
des blocs
; Unité phys. : 13 = [s]
; Valeurs limites : inférieure = 10,0, supérieure = non programmée => limite du domaine de définition
; Droits d'accès :
; Programme CN : écriture/lecture = 3 = utilisateur final
; OPI : écriture = 0 = Siemens, lecture = 3 = utilisateur final
; Valeur d'initialisation : ZEIT_1 = 12,0, ZEIT_2 = 45,0

DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER"
; Description
; Définition d'une variable GUD : GUD5_NAME
; Domaine de validité : global dans la CN
; Type de données : STRING, 5 caractères au maximum
; Arrêt du prétraitement des blocs : non programmé => valeur par défaut = pas d'arrêt du prétraitement
des blocs
; Unité phys. : non programmée => valeur par défaut = 0 = pas d'unité physique
; Valeurs limites : non programmées => limites du domaine de définition : inférieure = 0, supérieure
= 255
; Droits d'accès :
; Programme CN : écriture/lecture = 3 = utilisateur final
; OPI : écriture = 0 = Siemens, lecture = 3 = utilisateur final
; Valeur d'initialisation : "COUNTER"
M30
```

Exemple 2 : variables utilisateur globales et locales du programme (PUD/LUD)

Code de programme	Commentaire
PROC MAIN	; Programme principal
DEF INT VAR1	; Définition PUD
...	
SUB2	; Appel de sous-programme
...	
M30	

Code de programme	Commentaire
PROC SUB2	; Sous-programme SUB2
DEF INT VAR2	; Définition LUD
...	
IF (VAR1==1)	; Lecture PUD
VAR1=VAR1+1	; Lecture et écriture PUD

Code de programme	Commentaire
VAR2=1	; Ecriture LUD
ENDIF	
SUB3	; Appel de sous-programme
...	
M17	

Code de programme	Commentaire
PROC SUB3	; Sous-programme SUB3
...	
IF (VAR1==1)	; Lecture PUD
VAR1=VAR1+1	; Lecture et écriture PUD
VAR2=1	; Erreur : LUD de SUB2 non connu
ENDIF	
...	
M17	

Exemple 3 : définition et utilisation de variables utilisateur du type de données AXIS

Code de programme	Commentaire
DEF AXIS ABSCISSE	; 1er axe géométrique
DEF AXIS BROCHE	; Broche
...	
IF ISAXIS(1) == FALSE GOTOF SUITE	
ABSCISSE = $P_AXN1	
SUITE :	
...	
BROCHE=(S1)	; 1ère Broche
OVRA[BROCHE]=80	; Correction de vitesse de rotation de broche = 80 %
BROCHE=(S3)	; 3ème Broche

Autres conditions à prendre en compte

Variables utilisateur globales (GUD)

Les paramètres machines suivants doivent être pris en compte dans le cadre de la définition de variables utilisateur globales (GUD) :

Nº	Descripteur : $MN_	Signification
11140	GUD_AREA_ SAVE_TAB	Sauvegarde supplémentaire pour blocs GUD
18118 [1)]	MM_NUM_GUD_MODULES	Nombre de fichiers GUD dans le système de fichiers actif
18120 [1)]	MM_NUM_GUD_NAMES_NCK	Nombre de noms GUD globaux
18130 [1)]	MM_NUM_GUD_NAMES_CHAN	Nombre de noms GUD spécifiques à un canal
18140 [1)]	MM_NUM_GUD_NAMES_AXIS	Nombre de noms GUD spécifiques à un axe

N°	Descripteur : $MN_	Signification
18150 [1]	MM_GUD_VALUES_MEM	Espace mémoire pour valeurs GUD globales
18660 [1]	MM_NUM_SYNACT_GUD_REAL	Nombre de variables GUD configurables du type de données REAL
18661 [1]	MM_NUM_SYNACT_GUD_INT	Nombre de variables GUD configurables du type de données INT
18662 [1]	MM_NUM_SYNACT_GUD_BOOL	Nombre de variables GUD configurables du type de données BOOL
18663 [1]	MM_NUM_SYNACT_GUD_AXIS	Nombre de variables GUD configurables du type de données AXIS
18664 [1]	MM_NUM_SYNACT_GUD_CHAR	Nombre de variables GUD configurables du type de données CHAR
18665 [1]	MM_NUM_SYNACT_GUD_STRING	Nombre de variables GUD configurables du type de données STRING

[1] PM lisible pour SINUMERIK 828D uniquement !

Utilisation dans tous les canaux d'une variable utilisateur globale dans la CN du type de données AXIS

Une variable utilisateur globale de la NC du type de données `AXIS`, initialisée au moyen d'un descripteur lors de la définition dans le bloc de données, peut uniquement être utilisée dans différents canaux de la CN, si l'axe possède le même numéro de canal dans l'ensemble des canaux.

Si tel n'est pas le cas, la variable doit être chargée au début du programme CN, ou la fonction AXNAME(...) (voir "Fonctions d'axe (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Page 671)") doit être utilisée, comme dans l'exemple suivant.

```
Code de programme                          Commentaire
DEF NCK STRING[5] ACHSE="X"                ; Définition dans le bloc de données
...
N100 AX[AXNAME(AXE)]=111 G00               ; Utilisation dans le programme CN
```

2.1.5 Redéfinition de données système, données utilisateur et instructions de langage CN (REDEF)

L'instruction `REDEF` permet de modifier les attributs de données système, données utilisateur et instructions de langage CN. La condition de base nécessaire pour une redéfinition est qu'elle soit exécutée après la définition correspondante.

Lors d'une redéfinition, il n'est pas possible de modifier simultanément plusieurs attributs. Une instruction `REDEF` spécifique doit être programmée pour chaque attribut à modifier.

Dans le cas où plusieurs modifications d'attributs concurrentes sont programmées, la modification active est toujours la dernière.

Réinitialisation des valeurs d'attribut

Les attributs pour les droits d'accès et l'instant d'initialisation modifiés avec `REDEF` peuvent être réinitialisés à leur valeur par défaut par une nouvelle programmation de `REDEF`, suivie du nom de la variable ou de l'instruction de langage CN :

- Droits d'accès : Niveau de protection 7
- Instant d'initialisation : pas d'initialisation ou conservation de la valeur actuelle

Attributs redéfinissables

Voir "Vue d'ensemble des attributs définissables et redéfinissables (Page 55)".

Variables utilisateur locales (PUD / LUD)

Aucune redéfinition n'est autorisée pour des variables utilisateur locales (PUD / LUD).

Syntaxe

```
REDEF <nom> <arrêt prétraitement blocs>
REDEF <nom> <unité physique>
REDEF <nom> <valeurs limites>
REDEF <nom> <droits d'accès>
REDEF <nom> <instant d'initialisation>
REDEF <nom> <instant d'initialisation> <valeur d'initialisation>
REDEF <nom> <classe de données>
REDEF <nom>
```

Signification

`REDEF` :	Instruction pour la redéfinition d'un attribut précis ou pour la réinitialisation des attributs "Droits d'accès" et/ou "Instant d'initialisation" de variables système, de variables utilisateur et d'instructions de langage CN	
`<Nom>` :	Nom d'une variable déjà définie ou d'une instruction de langage CN	
`<arrêt prétraitement blocs>` :	Arrêt du prétraitement des blocs	
	`SYNR` :	arrêt du prétraitement lors de la lecture
	`SYNW` :	arrêt du prétraitement lors de l'écriture
	`SYNRW` :	arrêt du prétraitement lors de la lecture / l'écriture

<unité physique> :	Unité physique	
	PHU <unité> :	Unité physique
	Voir "Attribut : unité physique (PHU) (Page 46)". **Remarque** Non redéfinissable pour : • Variables système • Données utilisateur globales (GUD) des types de données : BOOL, AXIS, STRING, FRAME	
<valeurs limites> :	Valeur limite inférieure/supérieure	
	LLI <valeur limite> :	Valeur limite inférieure (lower limit)
	ULI <valeur limite> :	Valeur limite supérieure (upper limit)
	Voir "Attribut : valeurs limites (LLI, ULI) (Page 44)". **Remarque** Non redéfinissable pour : • Variables système • Données utilisateur globales (GUD) des types de données : BOOL, AXIS, STRING, FRAME	
<droits d'accès> :	Droits d'accès en lecture / écriture au moyen du programme pièce ou de BTSS	
	APX <niveau de protection> :	Exécuter : Elément de langage CN
	APRP <niveau de protection> :	Lecture : programme pièce
	APWP <niveau de protection> :	Ecriture : programme pièce
	APRB <niveau de protection> :	Lecture : OPI
	APWB <niveau de protection> :	Ecriture : OPI
	<niveau de protection> :	Plage de valeurs : 0 ... 7
	Voir "Attribut : droits d'accès (APR, APW, APRP, APWP, APRB, APWB) (Page 48)".	
<instant d'initialisation> :	Instant de réinitialisation de la variable	
	INIPO :	Power On
	INIRE :	Fin du programme principal, Reset CN ou Power On
	INICF :	NEWCONF ou fin du programme principal, Reset CN ou Power On
	PRLOC :	Fin du programme principal, Reset CN après une modification locale ou Power On
	Voir "Attribut : Valeur d'initialisation (Page 41)".	

<valeur d'initialisation> :	Valeur d'initialisation Pour redéfinir la valeur d'initialisation, il convient également de toujours spécifier un instant d'initialisation (voir <instant d'initialisation>). Voir "Attribut : Valeur d'initialisation (Page 41)". Pour l'initialisation de variables de tableau, voir "Définition et initialisation de variables de tableau (DEF, SET, REP) (Page 56)". **Remarque** Non redéfinissable pour les variables système, sauf les données de réglage.	
<classe de données> :	Attribution d'une classe de données (SINUMERIK 828D uniquement !)	
	DCM :	Classe de données M (= Manufacturer)
	DCI :	Classe de données I (= Individual)
	DCU :	Classe de données U (= User)
	Voir "Attribut : classe de données (DCM, DCI, DCU) – SINUMERIK 828D uniquement (Page 54)".	

Exemple

Redéfinitions de la variable système $TC_DPCx dans le bloc de données pour le constructeur de machines

Code de programme

```
%_N_MGUD_DEF                                ; Bloc GUD : constructeur de machines
N100 REDEF $TC_DPC1 APWB 2 APWP 3
N200 REDEF $TC_DPC2 PHU 21
N300 REDEF $TC_DPC3 LLI 0 ULI 200
N400 REDEF $TC_DPC4 INIPO (100, 101, 102, 103)
N800 REDEF $TC_DPC1
N900 REDEF $TC_DPC4
M30
```

Concernant N100 : Droit d'accès en écriture : BTSS = niveau de protection 2, programme pièce = niveau de protection 3

Concernant N200 : Unité physique [%]

Concernant N300 : inférieure = 0, supérieure = 200

Concer- La variable de tableau est initialisée avec les quatre valeurs lors d'un PowerOn.
nant
N400 :

Concer- Réinitialisation des valeurs d'attribut "Droits d'accès" et/ou "Instant d'initialisation"
nant
N800 /
N900

Remarque

Utilisation des fichiers ACCESS

Lorsque des fichiers ACCESS sont utilisés, la redéfinition des droits d'accès doit être transposée de _N_MGUD_DEF en _N_MACCESS_DEF.

Autres conditions à prendre en compte

Granularité

Une redéfinition concerne toujours la variable complète, caractérisée de façon univoque par son nom. Il n'est pas possible, p. ex. pour des variables de tableau, d'attribuer différentes valeurs d'attribut à des éléments individuels de tableau.

2.1.6 Attribut : Valeur d'initialisation

Définition (DEF) de variables utilisateur

Lors de la définition, une valeur d'initialisation peut être prédéfinie pour les variables utilisateur suivantes :

- Variables utilisateur globales (GUD)
- Variables utilisateur globales d'un programme (PUD)
- Variables utilisateur locales (LUD)

Redéfinition (REDEF) de variables système et de variables utilisateur

Lors de la redéfinition, une valeur d'initialisation peut être prédéfinie pour les variables suivantes :

- Données système
  - Données de réglage
- Données utilisateur
  - Paramètres R
  - Variable à action synchrone ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - Variable GUD à action synchrone (SYG_xy[ ], avec x=R, I, B, A, C, S et y=S, M, U, 4, ..., 9)
  - Paramètre EPS
  - Données d'outil OEM
  - Données de magasin OEM
  - Variables utilisateur globales (GUD)

Instant de réinitialisation

Lors de la redéfinition, il est possible de spécifier l'instant auquel la variable est réinitialisée, c'est-à-dire doit reprendre sa valeur d'initialisation :

- `INIPO` (Power On)
  La variable est réinitialisée lors d'un Power On.
- `INIRE` (Reset)
  La variable est réinitialisée lors d'un Reset CN, d'un Reset GMFC, à la fin du programme pièce (`M02`/`M30`) ou lors d'un Power On.
- `INICF` (NEWCONF)
  La variable est réinitialisée lors de la fonction "Activation des paramètres machine" via IHM, via une instruction du programme pièce `NEWCONF` ou via un Reset CN, un Reset GMFC, ou à la fin du programme pièce (M02/M30) ou lors d'un Power On.
- `PRLOC` (modification locale du programme)
  La variable est uniquement réinitialisée en cas de Reset CN, Reset GMFC, à la fin du programme pièce (M02 / M30) si elle a été modifiée dans le cadre du programme pièce actuel.
  L'attribut `PRLOC` ne doit être utilisé que dans le contexte de données de réglage programmables (voir le tableau suivant).

Tableau 2-1 Données de réglage programmables

Numéro	Descripteur	Fonction G [1)]
42000	$SC_THREAD_START_ANGLE	`SF`
42010	$SC_THREAD_RAMP_DISP	`DITS` / `DITE`
42400	$SA_PUNCH_DWELLTIME	`PDELAYON`
42800	$SA_SPIND_ASSIGN_TAB	`SETMS`
43210	$SA_SPIND_MIN_VELO_G25	`G25`
43220	$SA_SPIND_MAX_VELO_G26	`G26`

Numéro	Descripteur	Fonction G [1)]
43230	$SA_SPIND_MAX_VELO_LIMS	LIMS
43300	$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
43420	$SA_WORKAREA_LIMIT_PLUS	G26
43430	$SA_WORKAREA_LIMIT_MINUS	G25
43510	$SA_FIXED_STOP_TORQUE	FXST
43520	$SA_FIXED_STOP_WINDOW	FXSW
43700	$SA_OSCILL_REVERSE_POS1	OSP1
43710	$SA_OSCILL_REVERSE_POS2	OSP2
43720	$SA_OSCILL_DWELL_TIME1	OST1
43730	$SA_OSCILL_DWELL_TIME2	OST2
43740	$SA_OSCILL_VELO	FA
43750	$SA_OSCILL_NUM_SPARK_CYCLES	OSNSC
43760	$SA_OSCILL_END_POS	OSE
43770	$SA_OSCILL_CTRL_MASK	OSCTRL
43780	$SA_OSCILL_IS_ACTIVE	OS
43790	$SA_OSCILL_START_POS	OSB
1) Cette fonction G sollicite la donnée de réglage		

Autres conditions à prendre en compte

Valeur d'initialisation : Variables utilisateur globales (GUD)

- Dans le cas de la variable utilisateur globale (GUD) avec le domaine de validité CN, seul l'instant d'initialisation INIPO (Power On) peut être spécifié.
- Pour les variables utilisateur globales (GUD) avec le domaine de validité CHAN, il est possible de spécifier comme instant d'initialisation, outre INIPO (Power On), également INIRE (Reset) ou INICF (NEWCONF).
- Pour les variables utilisateur globales (GUD) avec le domaine de validité CHAN et l'instant d'initialisation INIRE (Reset) ou INICF (NEWCONF), les variables sont seulement réinitialisées pour Reset CN, Reset GMFC et "Activation des paramètres machine" dans les canaux dans lesquels les événements spécifiés ont été déclenchés.

Valeur d'initialisation : type de données FRAME

Pour les variables du type de données FRAME, la spécification d'une valeur d'initialisation n'est pas autorisée. Les variables du type de données FRAME sont implicitement toujours initialisées avec le frame par défaut.

Valeur d'initialisation : type de données CHAR

Pour les variables du type de données CHAR, il est possible de programmer à la place du code ASCII (0...255), le caractère ASCII correspondant représenté entre guillemets, p. ex. "A"

Valeur d'initialisation : type de données STRING

Pour les variables du type de données STRING, la chaîne de caractères doit figurer entre guillemets, p. ex. : ...= "MACHINE_1"

Valeur d'initialisation : type de données AXIS

Pour les variables du type de données `AXIS`, le descripteur d'axe doit figurer entre guillemets pour la notation étendue d'adresse, p. ex. : ...=(X3)

Valeur d'initialisation : Variable système

Pour les variables système, il n'est pas possible de prédéfinir par redéfinition des valeurs d'initialisation spécifiques à l'utilisateur. Les valeurs d'initialisation des variables système sont prédéfinies par le système. Il est par contre possible de modifier par redéfinition l'instant (`INIRE`, `INICF`) auquel la variable système est réinitialisée.

Valeur d'initialisation implicite : type de données AXIS

Pour les variables du type de données `AXIS`, la valeur d'initialisation implicite suivante est utilisée :

- Données système : "premier axe géométrique"
- GUD d'action synchrone (désignation : SYG_A*), PUD, LUD :
  descripteur d'axe à partir du paramètre machine : PM20082
  $MC_AXCONF_CHANAX_DEFAULT_NAME

Valeur d'initialisation implicite : données d'outil et de magasin

Pour les données d'outil et de magasin, il est possible de spécifier des valeurs d'initialisation à partir du paramètre machine suivant : PM17520 $MN_TOOL_DEFAULT_DATA_MASK

Remarque

Synchronisation

La synchronisation d'événements déclenchant la réinitialisation d'une variable globale avec lecture de cette variable à un autre endroit relève totalement de la responsabilité de l'utilisateur / du constructeur de machine.

Voir aussi

Variables (Page 23)

2.1.7 Attribut : valeurs limites (LLI, ULI)

Une valeur limite supérieure et inférieure de la plage de définition peut uniquement être prédéfinie pour les types de données suivants :

- INT
- REAL
- CHAR

Définition (DEF) de variables utilisateur : valeurs limites et valeurs d'initialisation implicites

Si aucune valeur d'initialisation explicite n'est définie lors de la déclaration d'une variable utilisateur de l'un des types de données précités, la valeur d'initialisation implicite du type de données est affectée à la variable :

- INT : 0
- REAL : 0.0
- CHAR : 0

Si la valeur d'initialisation implicite se situe en dehors de la plage de définition déterminée par les valeurs limites programmées, la variable est initialisée avec la valeur limite la plus proche de la valeur d'initialisation implicite :

- Valeur d'initialisation implicite **<** valeur limite inférieure (LLI) ⇒
  valeur d'initialisation **=** valeur limite inférieure
- Valeur d'initialisation implicite **>** valeur limite supérieure (ULI) ⇒
  valeur d'initialisation **=** valeur limite supérieure

Exemples :

Code de programme	**Commentaire**

`DEF REAL GUD1`	`; Valeur limite inférieure = limite de la plage de définition`
	`; Valeur limite supérieure = limite de la plage de définition`
	`; Aucune valeur d'initialisation programmée`
	`; => valeur d'initialisation implicite = 0.0`

`DEF REAL LLI 5.0 GUD2`	`; Valeur limite inférieure = 5.0`
	`; Valeur limite supérieure = limite de la plage de définition`
	`; => Valeur d'initialisation = 5.0`

`DEF REAL ULI -5 GUD3`	`; Valeur limite inférieure = limite de la plage de définition`
	`; Valeur limite supérieure = -5.0`
	`; => Valeur d'initialisation = -5,0`

Redéfinition (REDEF) de variables utilisateur : valeurs limites et valeurs réelles actuelles

Si lors de la redéfinition des valeurs limites d'une variable utilisateur, celles-ci sont modifiées de telle sorte que la valeur réelle actuelle se situe en dehors de la nouvelle plage de définition, une alarme est générée et les valeurs limites ne sont pas appliquées.

Remarque

Redéfinition (REDEF) de variables utilisateur

Lors de la redéfinition des valeurs limites d'une variable utilisateur, il faut veiller à la modification cohérente des valeurs suivantes :

- Valeurs limites
- Valeur réelle
- Valeur d'initialisation lors de la redéfinition et de la réinitialisation automatique sur la base de INIPO, INIRE ou INICF

Voir aussi

Variables (Page 23)

2.1.8 Attribut : unité physique (PHU)

Une unité physique peut uniquement être prédéfinie pour des variables possédant l'un des types de données suivants :

- INT
- REAL

Unités physiques programmables (PHU)

L'unité physique est spécifiée en tant que nombre à virgule fixe : `PHU <unité>`

Les unités physiques suivantes peuvent être programmées :

<Unité>	Signification	Unité physique
0	Pas d'unité physique	-
1	Position linéaire ou angulaire[1)2)]	[mm], [inch], [degré]
2	Position linéaire[2)]	[mm], [inch]
3	Position angulaire	[degré]
4	Vitesse linéaire ou angulaire[1)2)]	[mm/min], [inch/min], [tr/min]
5	Vitesse linéaire[2)]	[mm/min]
6	Vitesse angulaire	[tr/min]
7	Accélération linéaire ou angulaire[1)2)]	[m/s^2], [$inch/s^2$], [tr/s^2]
8	Accélération linéaire[2)]	[m/s^2], [$inch/s^2$]
9	Accélération angulaire	[tr/s^2]
10	A-coup linéaire ou angulaire[1)2)]	[m/s^3], [$inch/s^3$], [tr/s^3]

<Unité>	Signification	Unité physique
11	A-coup linéaire[2)]	[m/s^3], [inch/s^3]
12	A-coup angulaire	[tr/s^3]
13	Temporisation	[s]
14	Gain du régulateur de position	[16.667/s]
15	Avance par tour [2)]	[mm/tr], [inch/tr]
16	Compensation de température [1)2)]	[mm], [inch]
18	Force	[N]
19	Masse	[kg]
20	Moment d'inertie[3)]	[kgm^2]
21	Pour cent	[%]
22	Fréquence	[Hz]
23	Tension	[V]
24	Courant	[A]
25	Température	[°C]
26	Angle	[degré]
27	KV	[1000/min]
28	Position linéaire ou angulaire[3)]	[mm], [inch], [degré]
29	Vitesse de coupe [2)]	[m/min], [pied/min]
30	Vitesse circonférentielle[2)]	[m/s], [pied/s]
31	Résistance	[Ohm]
32	Inductance	[mH]
33	Couple [3)]	[Nm]
34	Constante du couple [3)]	[Nm/A]
35	Gain du régulateur de courant	[V/A]
36	Gain de vitesse de rotation [3)]	[Nm/(rad*s)]
37	Vitesse de rotation	[tr/min]
42	Puissance	[kW]
43	Courant faible	[µA]
46	Couple faible [3)]	[µNm]
48	Pour mille	-
49	-	[Hz/s]
65	Débit	[l/min]
66	pression	[bar]
67	Volumes [3)]	[cm^3]
68	Gain du système[3)]	[mm/(V*min)]
69	Gain du système pour régulateur de couple	[N/V]
155	Pas de vis [3)]	[mm/tr], [inch/tr]
156	Modification du pas de vis [3)]	[mm/tr / tr], [inch/tr / tr]
1) L'unité physique dépend du type d'axe : axe linéaire ou rotatif		

<Unité>	Signification	Unité physique
2) Commutation du système de coordonnées G70/G71(inch/métrique) Après une commutation du système de base (PM10240 $MN_SCALING_SYSTEM_IS_METRIC) avec G70/G71, **aucune** conversion des valeurs (valeur réelle, valeur par défaut et valeurs limites) n'est effectuée lors d'accès d'écriture/de lecture à des variables système et utilisateur relatives à des longueurs G700/G710(inch/métrique) Après une commutation du système de base (PM10240 $MN_SCALING_SYSTEM_IS_METRIC) avec G700/G710, **une** conversion des valeurs (valeur réelle, valeur par défaut et valeurs limites) est effectuée lors d'accès d'écriture/de lecture à des variables système et utilisateur relatives à des longueurs		
3) La variable n'est **pas** convertie automatiquement dans le système de coordonnées actuel de la CN (en pouces / métrique). La conversion relève de l'entière responsabilité de l'utilisateur / constructeur de machine.		

Remarque

Débordement de niveau dû aux conversions de format

Le format de stockage interne de l'ensemble des variables utilisateur (GUD / PUD / LUD) exprimées en unités physiques de longueur est métrique. Une utilisation excessive de telles variables lors de l'exécution de la NCK, par ex. durant des actions synchrones, risquerait de provoquer un dépassement de la durée de calcul du niveau de l'interpolateur lors d'un changement de système de coordonnées, conduisant ainsi à la génération de l'alarme 4240.

Remarque

Compatibilité d'unités

La compatibilité des unités concernées n'est pas vérifiée lors de l'utilisation de variables (affectation, comparaison, calcul, etc.). Une éventuelle conversion requise relève exclusivement de la responsabilité de l'utilisateur / constructeur de machine.

Voir aussi

Variables (Page 23)

2.1.9 Attribut : droits d'accès (APR, APW, APRP, APWP, APRB, APWB)

Désignation

La désignation des attributs d'accès `AP`... se compose comme suit :

1. A : **A**ccess
2. P : **P**rotection
3. R / W : **R**ead / **W**rite
4. P / B : **P**rogram / **B**TSS

Droits d'accès / niveaux de protection

Les niveaux de protection suivants, à indiquer lors de la programmation, correspondent aux différents droits d'accès :

Droit d'accès	Niveau de protection
Mot de passe système	0
Mot de passe constructeur de machines	1
Mot de passe maintenance	2
Mot de passe utilisateur final	3
Commutateur à clé, position 3	4
Commutateur à clé, position 2	5
Commutateur à clé, position 1	6
Commutateur à clé, position 0	7

Définition (DEF) des données utilisateur

Il est possible de définir des droits d'accès (APR... / APW...) pour les données suivantes :

- Données utilisateur globales (GUD)

Redéfinition (REDEF) des données système et utilisateur

Il est possible de redéfinir des droits d'accès (APR... / APW...) pour les données suivantes :

- Données système
  - Paramètres machine

    Remarque

    Redéfinition des droits de lecture des paramètres machine

    Le niveau de protection pour la lecture des paramètres machine ne peut être réglé que simultanément pour le programme pièce et BTSS à l'aide du mot clé APR.

    Les mots clé APRP et APRB ne sont pas pris en charge pour la redéfinition des droits de lecture et génèrent l'alarme 12490 "Le droit d'accès APRP/APRB <niveau de protection> n'a pas été paramétré".

  - Données de réglage
  - Variable système
  - Données process
  - Données de magasin
  - Données d'outil
- Données utilisateur
  - Paramètres R
  - Variable à action synchrone ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - Variable GUD à action synchrone (SYG_xy[ ], avec x=R, I, B, A, C, S et y=S, M, U, 4, ..., 9)
  - Paramètre EPS
  - Données d'outil OEM
  - Données de magasin OEM
  - Variables utilisateur globales (GUD)

    Remarque

    Lors de la redéfinition, il est possible d'attribuer le droit d'accès à une variable quelconque comprise entre le niveau de protection le plus faible, à savoir 7, et son propre niveau de protection, par ex. 1 (constructeur de machines).

Redéfinition (REDEF) d'instructions de langage CN

Il est possible de redéfinir le droit d'accès ou d'exécution (APX) pour les instructions de langage CN suivantes :

- Fonctions G / fonctions préparatoires
  Bibliographie
  Manuel de programmation Notions de base ; chapitre : Fonctions G / fonctions préparatoires
- Fonctions prédéfinies
  Bibliographie
  Manuel de programmation Notions de base ; chapitre : Fonctions prédéfinies
- Appels de sous-programmes prédéfinis
  Bibliographie
  Manuel de programmation Notions de base ; chapitre : Appels de sous-programmes prédéfinis
- Instruction DO pour les actions synchrones
- Descripteurs de cycles dans un programme
  Le cycle doit être enregistré dans un répertoire de cycles et contenir une instruction PROC.

Droits d'accès concernant les programmes CN et les cycles (APRP, APWP)

Les différents droits d'accès à partir d'un programme CN ou d'un cycle ont les incidences suivantes :

- APRP 0 / APWP 0
  - L'exécution du programme CN requiert le mot de passe système.
  - Le cycle doit être enregistré dans le répertoire _N_CST_DIR (système).
  - Pour le répertoire _N_CST_DIR, le droit d'exécution doit être défini sur Système dans le paramètre machine PM11160 $MN_ACCESS_EXEC_CST.
- APRP 1 / APWP 1 ou APRP 2 / APWP 2
  - L'exécution du programme CN requiert le mot de passe du constructeur de machines ou du service de maintenance.
  - Le cycle doit être enregistré dans le répertoire _N_CMA_DIR (constructeur de machines) ou _N_CST_DIR.
  - Pour les répertoires _N_CMA_DIR ou _N_CST_DIR, les droits d'exécution doivent être définis au moins pour le constructeur de machines dans les paramètres machine PM11161 $MN_ACCESS_EXEC_CMA ou PM11160 $MN_ACCESS_EXEC_CST.

- APRP 3 / APWP 3
  - L'exécution du programme CN requiert le mot de passe de l'utilisateur final.
  - Le cycle doit être enregistré dans le répertoire _N_CUS_DIR (utilisateur), _N_CMA_DIR ou _N_CST_DIR.
  - Pour les répertoires _N_CUS_DIR, _N_CMA_DIR ou _N_CST_DIR, les droits d'exécution doivent être définis au moins pour l'utilisateur final dans les paramètres machine PM11162 $MN_ACCESS_EXEC_CUS, PM11161 $MN_ACCESS_EXEC_CMA ou PM11160 $MN_ACCESS_EXEC_CST.
- APRP 4...7 / APWP 4...7
  - L'exécution du programme CN requiert la position 3 ... 0 du commutateur à clé.
  - Le cycle doit être enregistré dans le répertoire _N_CUS_DIR, _N_CMA_DIR ou _N_CST_DIR.
  - Pour les répertoires _N_CUS_DIR, _N_CMA_DIR ou _N_CST_DIR, les droits d'exécution doivent être définis au moins pour la position correspondante du commutateur à clé dans les paramètres machine PM11162 $MN_ACCESS_EXEC_CUS, PM11161 $MN_ACCESS_EXEC_CMA ou PM11160 $MN_ACCESS_EXEC_CST.

Droits d'accès concernant BTSS (APRB, APWB)

Les droits d'accès (APRB, APWB) restreignent l'accès via BTSS aux variables système et variables utilisateur de la même manière pour tous les composants système (IHM, AP, ordinateurs externes, services EPS, etc.).

Remarque

Droits d'accès IHM locaux

Lors de la modification des droits d'accès de données système, il est nécessaire de veiller à la cohérence avec les droits d'accès définis par les mécanismes de l'IHM.

Attributs d'accès APR / APW

Pour des raisons de compatibilité, les attributs APR et APW sont représentés de manière implicite sur les attributs APRP / APRB et APWP / APWB :

- APR x ⇒ APRP x APRB x
- APW y ⇒ APWP y APWB y

Droits d'accès au moyen de fichiers ACCESS

Lorsque vous utilisez des fichiers ACCESS pour l'attribution de droits d'accès, les redéfinitions de ces droits pour les données système, données utilisateur et instructions de langage CN doivent être effectuées exclusivement dans les fichiers ACCESS. Les données utilisateur globales (GUD) constituent une exception. Pour celles-ci, la redéfinition doit toujours être effectuée dans les fichiers de définition *_DEF correspondants.

Pour une protection d'accès continue, les paramètres machine doivent être adaptés de façon cohérente pour les droits d'exécution et le droit d'accès des répertoires correspondants.

Il en résulte la marche à suivre de principe suivante :

1. Création des fichiers de définition requis :
   - _N_DEF_DIR/_N_SACCESS_DEF
   - _N_DEF_DIR/_N_MACCESS_DEF
   - _N_DEF_DIR/_N_UACCESS_DEF
2. Paramétrage du droit d'accès pour les fichiers de définition à la valeur requise pour la redéfinition :
   - PM11170 $MN_ACCESS_WRITE_SACCESS = <niveau de protection>
   - PM11171 $MN_ACCESS_WRITE_MACCESS = <niveau de protection>
   - PM11172 $MN_ACCESS_WRITE_UACCESS = <niveau de protection>
3. L'accès depuis des cycles à des éléments protégés nécessite l'adaptation des droits d'exécution et des droits d'écriture aux répertoires _N_CST_DIR, _N_CMA_DIR et _N_CST_DIR de ces cycles :
   Droits d'exécution
   - PM11160 $MN_ACCESS_EXEC_CST = <niveau de protection>
   - PM11161 $MN_ACCESS_EXEC_CMA = <niveau de protection>
   - PM11162 $MN_ACCESS_EXEC_CUS = <niveau de protection>

   Droits d'écriture
   - PM11165 $MN_ACCESS_WRITE_CST = <niveau de protection>
   - PM11166 $MN_ACCESS_WRITE_CMA = <niveau de protection>
   - PM11167 MN_ACCESS_WRITE_CUS = <niveau de protection>

   Le droit d'exécution doit être défini au moins au même niveau de protection que le niveau de protection le plus élevé de l'élément utilisé.
   Le droit d'écriture doit être défini au moins au même niveau de protection que le droit d'exécution.
4. Les droits d'écriture des répertoires des cycles pour l'IHM locale doivent être définis au même niveau de protection que les répertoires des cycles de la CN locale.
   Bibliographie
   Manuel d'utilisation

Appels de sous-programmes dans des fichiers ACCESS

Pour une meilleure structuration de la protection d'accès, il est possible d'appeler également des sous-programmes (extension SPF ou MPF) dans les fichiers ACCESS. Les droits d'accès du fichier ACCESS appelant sont alors attribués à ces sous-programmes.

Remarque

Seuls les droits d'accès peuvent être redéfinis dans les fichiers ACCESS. La programmation ou la redéfinition de tous les autres attributs doit toujours encore s'effectuer dans les fichiers de définition correspondants.

Voir aussi

Variables (Page 23)

2.1.10 Attribut : classe de données (DCM, DCI, DCU) – SINUMERIK 828D uniquement

Pour simplifier le traitement des données dans le cadre de la mise en service, de la mise en service de série et de la mise à niveau de machines et séries de machines, toutes les données système et utilisateur de la CN sont réparties en classes de données.

Classe de données	Données
S = System (système)	Données système mises à disposition par Siemens, comme les paramètres machine et données de réglage, les cycles standard et de mesure, les définitions (SGUD) et macros (SMAC), etc.
M = Manufacturer (constructeur de machine)	Données de mise en service spécifiques à la série de machines, comme les cycles constructeur, les définitions (MGUD) et macros (MMAC), ainsi que les paramètres machine, qui déterminent l'étendue des fonctions de la machine.
I = Individual (spécifiques à la machine)	Données de mise en service spécifiques à la machine, comme les données de compensation et les décalages de point de référence.
U = User (utilisateur)	Données spécifiques à la machine, générées lors du fonctionnement de la machine, comme les données d'outil, les données de réglage, les programmes pièce, les cycles utilisateur, les définitions (UGUD) et les macros (UMAC).

Bibliographie :
SINUMERIK 828D Manuel de mise en service Tournage et Fraisage ; chap. : "Introduction et utilisation des classes de données"

Définition (DEF) des données utilisateur

La classe d'une donnée est déterminée implicitement par la classe de données du fichier ou du répertoire dans lequel une donnée utilisateur est définie. La classe de la donnée ne peut pas être modifiée.

Lors de la définition (`DEF`) de la donnée utilisateur, il est toutefois possible d'attribuer à la **valeur de donnée** une classe de données propre, différente de la classe de la donnée en elle-même.

Pour la classe de données de la valeur de donnée, on doit avoir :

Priorité de la classe de données de la valeur de donnée **≤** priorité de la classe de la donnée en elle-même

Exemple :

La définition de la variable GUD qui décrit un palpeur doit se trouver dans la classe de données M (= Manufacturer) car elle est nécessaire pour l'exécution des cycles constructeur.

Cependant, la valeur de la donnée doit appartenir à la classe de données I (= Individual) car le type de palpeur peut être différent selon les machines.

MGUD.DEF (classe de données M)

```
...
DEF CHAN DCI INT CALIPER
...
```

Redéfinition (REDEF) des données système

La classe des données système peut être modifiée au moyen d'une redéfinition (REDEF). La redéfinition doit être effectuée dans un fichier de définition de la classe de données S ou M.

Lors de l'utilisation de fichiers ACCESS, les redéfinitions doivent être effectuées uniquement dans les fichiers ACCESS.

La classe de données respective des paramètres machine, des données de réglage et des données optionnelles, ainsi que des variables système, se trouve dans :

- Tables de paramètres Description détaillée des paramètres machine, paramètre : "Classe"
- Tables de paramètres Variables système

2.1.11 Vue d'ensemble des attributs définissables et redéfinissables

Les tableaux suivants indiquent pour quels types de données il est possible de définir (DEF) et/ou redéfinir (REDEF) les différents attributs.

Données système

Type de donnée	Valeur initiale	Valeurs limites	Unité physique	Droits d'accès	Classe de données (828D uniquement)
Paramètres machine	---	---	---	REDEF	REDEF
Données de réglage	REDEF	---	---	REDEF	---
Données FRAME	---	---	---	REDEF	---
Données process	---	---	---	REDEF	---
Compensation d'erreur de pas de vis de transmission (EEC)	---	---	---	REDEF	---
Compensation de la flexion (CEC)	---	---	---	REDEF	---
Compensation des défauts aux transitions entre quadrants (QEC)	---	---	---	REDEF	---
Données de magasin	---	---	---	REDEF	---
Données d'outil	---	---	---	REDEF	---
Zones de protection	---	---	---	REDEF	---
Organes porte-outil orientables	---	---	---	REDEF	---
Chaînes cinématiques	---	---	---	REDEF	---

Type de donnée	Valeur initiale	Valeurs limites	Unité physique	Droits d'accès	Classe de données (828D uniquement)
Zones de protection 3D	---	---	---	REDEF	---
Limitation de la zone de travail	---	---	---	REDEF	---

Données utilisateur

Type de donnée	Valeur initiale	Valeurs limites	Unité physique	Droits d'accès	Classe de données
Paramètre R	REDEF	REDEF	REDEF	REDEF	---
Variable à action synchrone ($AC_...)	REDEF	REDEF	REDEF	REDEF	---
GUD d'action synchrone (SYG_...)	REDEF	REDEF	REDEF	REDEF	---
Paramètre EPS	REDEF	REDEF	REDEF	REDEF	---
Données outil OEM	REDEF	REDEF	REDEF	REDEF	---
Données magasin OEM	REDEF	REDEF	REDEF	REDEF	---
Variables utilisateur globales (GUD)	DEF / REDEF	DEF	DEF	DEF / REDEF	DEF / REDEF
Variables utilisateur locales (PUD / LUD)	DEF	DEF	DEF	---	---

Voir aussi

Variables (Page 23)

2.1.12 Définition et initialisation de variables de tableau (DEF, SET, REP)

Une variable utilisateur peut être définie comme tableau (Array) de 1 à 3 dimensions au maximum :

- 1 dimension : `DEF <type de données> <nom de variable>[<n>]`
- 2 dimensions : `DEF <type de données> <nom de variable>[<n>,<m>]`
- 3 dimensions : `DEF <type de données> <nom de variable>[<n>,<m>,<o>]`

Remarque

Les variables utilisateur du type de données STRING peuvent être définies comme tableau à 2 dimensions au maximum.

Types de données

Les variables utilisateur peuvent être définie comme tableaux pour les types de données suivants : BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Affectation de valeur à des éléments de tableau

Les affectations de valeurs à des éléments de tableau peuvent être réalisées aux instants suivants :

- Lors de la définition du tableau (valeurs d'initialisation)
- Pendant l'exécution du programme

Les affectations de valeurs peuvent être réalisées par :

- Indication explicite d'un élément de tableau
- Indication explicite d'un élément de tableau comme élément de départ et indication d'une liste de valeurs (`SET`)
- Indication explicite d'un élément de tableau comme élément de départ et indication d'une valeur ainsi que de sa fréquence d'itération (`REP`)

Remarque

Il n'est pas possible d'affecter des valeurs d'initialisation à des variables utilisateur du type de données FRAME.

Syntaxe (`DEF`)

```
DEF <type de données> <nom de variable>[<n>,<m>,<o>]
DEF  STRING[<longueur de chaîne>] <nom de variable>[<n>,<m>]
```

Syntaxe (`DEF...=SET...`)

Utilisation d'une liste de valeurs :

- Lors de la définition :
  ```
  DEF <type de données> <nom de variable>[<n>,<m>,<o>] =
  SET(<valeur1>,<valeur2>,...)
  ```
  Equivaut à :
  ```
  DEF <type de données> <nom de variable>[<n>,<m>,<o>]=
  (<valeur1>,<valeur2>,...)
  ```

  Remarque

  Lors de l'initialisation au moyen d'une liste de valeurs, l'indication de `SET` est facultative.

- Lors d'une affectation de valeur :
  ```
  <Nom de variable>[<n>,<m>,<o>]=SET(<VALEUR1>,<valeur2>,...)
  ```

Syntaxe (DEF...=REP...)

Utilisation d'une valeur avec itération

- Lors de la définition :
  ```
  DEF <type de données> <nom de variable>[<n>,<m>,<o>]=REP(<Valeur>)
  DEF <type de données> <nom de variable>[<n>,<m>,<o>] =
  REP(<valeur>,<nombre d'éléments de tableau>)
  ```
- Lors d'une affectation de valeur :
  ```
  <Nom de variable>[<n>,<m>,<o>]=REP(<Valeur>)
  <Nom de variable>[<n>,<m>,<o>] = REP(<valeur>,<nombre d'éléments
  de tableau>)
  ```

Signification

DEF :	Instruction de définition des variables	
<type de données> :	Type de données de la variable	
	Plage de valeurs : • Pour les variables système : BOOL, CHAR, INT, REAL, STRING, AXIS • Pour les variables GUD et LUD : BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME	
<longueur de chaîne> :	Nombre maximal de caractères du type de données STRING	
<nom de variable> :	Nom de variable	
[<n>,<m>,<o>] :	Dimensions de tableaux ou indices de tableaux	
<n> :	Dimension ou indice pour un tableau à 1 dimension	
	Type :	INT (également AXIS pour les variables système)
	Plage de valeurs :	Dimension de tableau max. : 65535 Indice de tableau : 0 ≤ n ≤ 65534
<m> :	Dimension ou indice pour un tableau à 2 dimension	
	Type :	INT (également AXIS pour les variables système)
	Plage de valeurs :	Dimension de tableau max. : 65535 Indice de tableau : 0 ≤ m ≤ 65534
<o> :	Dimension ou indice pour un tableau à 3 dimension	
	Type :	INT (également AXIS pour les variables système)
	Plage de valeurs :	Dimension de tableau max. : 65535 Indice de tableau : 0 ≤ o ≤ 65534
SET :	Affectation de valeur au moyen de la liste de valeurs indiquée	
(<valeur1>,<valeur2>,...) :	Liste de valeurs	
REP :	Affectation de valeur au moyen de la <valeur> indiquée	

<valeur> :	Valeur à écrire dans les éléments de tableau lors de l'initialisation avec REP
<nombre d'éléments de tableau> :	Nombre d'éléments de tableau à décrire avec la <valeur> indiquée. Pour les autres éléments de tableau, s'applique selon l'instant : • Initialisation lors d la définition du tableau : → Les éléments de tableau restant sont décrits avec des zéros • Affectation pendant l'exécution du programme : → Les valeurs courantes des éléments de tableau demeurent inchangées. Si le paramètre n'est pas programmé, tous les éléments de tableau sont décrits avec <valeur>. Si le paramètre est égal à zéro, s'applique selon l'instant : • Initialisation lors d la définition du tableau : → Tous les éléments sont mis à zéro par le système. • Affectation pendant l'exécution du programme : → Les valeurs courantes des éléments de tableau demeurent inchangées.

Indice de tableau

L'ordre implicite des éléments de tableau, p. ex. lors d'une affectation de valeur via SET ou REP, est déterminé par itération de l'indice de tableau de droite à gauche.

Exemple : Initialisation d'un tableau à 3 dimensions comportant 24 éléments de tableau :

```
DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1      1. élément de tableau
  FELD[0,0,1] = 1      2. élément de tableau
  FELD[0,0,2] = 1      3. élément de tableau
  FELD[0,0,3] = 1      4. élément de tableau
  ...
  FELD[0,1,0] = 1      5. élément de tableau
  FELD[0,1,1] = 1      6. élément de tableau
  ...
  FELD[0,2,3] = 1      12. élément de tableau
  FELD[1,0,0] = 1      13. élément de tableau
  FELD[1,0,1] = 1      14. élément de tableau
  ...
  FELD[1,2,3] = 1      24. élément de tableau
```

Correspond à :

```
FOR n=0 TO 1
  FOR m=0 TO 2
     FOR o=0 TO 3
```

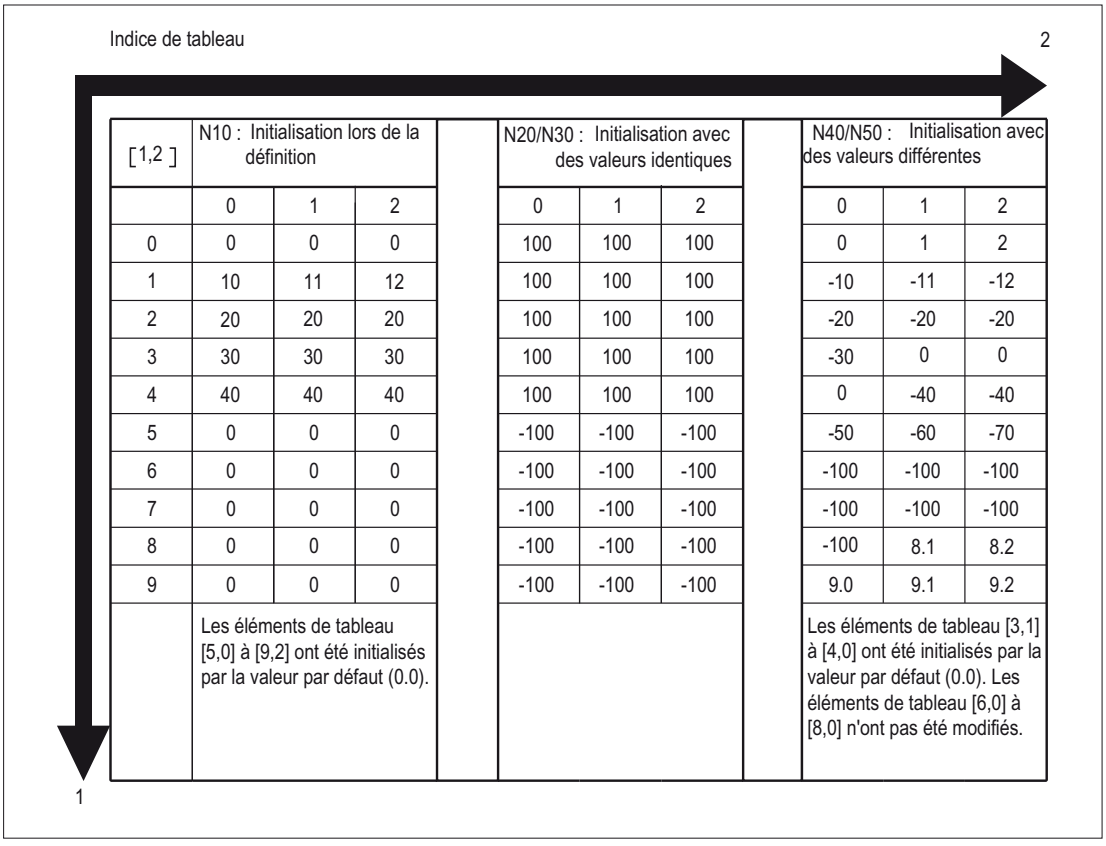
```
        FELD[n,m,o] = 1
      ENDFOR
    ENDFOR
ENDFOR
```

Exemple : Initialisation de tableaux de variables complets

Pour l'affectation courante, voir la figure.

Code de programme

```
N10 DEF REAL TABLEAU1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)
N20 TABLEAU1[0,0]=REP(100)
N30 TABLEAU1[5,0]=REP(-100)
N40 TABLEAU1[0,0]=SET(0,1,2,-10,-11,-12,-20,-20,-20,-30, , , ,-40,-40,-50,-60,-70)
N50 TABLEAU1[8,1]=SET(8.1,8.2,9.0,9.1,9.2)
```



[1,2]	N10 : Initialisation lors de la définition		
	0	1	2
0	0	0	0
1	10	11	12
2	20	20	20
3	30	30	30
4	40	40	40
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
	Les éléments de tableau [5,0] à [9,2] ont été initialisés par la valeur par défaut (0.0).		

N20/N30 : Initialisation avec des valeurs identiques		
0	1	2
100	100	100
100	100	100
100	100	100
100	100	100
100	100	100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100

N40/N50 : Initialisation avec des valeurs différentes		
0	1	2
0	1	2
-10	-11	-12
-20	-20	-20
-30	0	0
0	-40	-40
-50	-60	-70
-100	-100	-100
-100	-100	-100
-100	8.1	8.2
9.0	9.1	9.2
Les éléments de tableau [3,1] à [4,0] ont été initialisés par la valeur par défaut (0.0). Les éléments de tableau [6,0] à [8,0] n'ont pas été modifiés.		

Voir aussi

Définition et initialisation de variables de tableau (DEF, SET, REP) : Informations complémentaires (Page 61)

Variables (Page 23)

2.1.13 Définition et initialisation de variables de tableau (DEF, SET, REP) : Informations complémentaires

Informations complémentaires (SET)

Initialisation lors de la définition

- Le nombre d'éléments de tableau correspondant aux éléments programmés dans la liste de valeurs sont initialisés avec les valeurs de la liste, à commencer par le 1er élément de tableau.
- La valeur 0 est affectée aux éléments de tableau sans valeur explicite indiquée dans la liste de valeurs (lacunes dans la liste de valeurs).
- Des lacunes dans la liste de valeurs ne sont pas admises pour les variables du type de données AXIS.
- Si la liste contient davantage de valeurs que d'éléments de tableau ne sont définis, une alarme est émise.

Affectation de valeur pendant l'exécution du programme

Les règles décrites ci-dessus dans le cadre de la définition s'appliquent à l'affectation de valeur pendant l'exécution du programme. Il existe en outre les possibilités suivantes :

- Il est également possible d'utiliser des expressions comme éléments dans la liste de valeurs.
- L'affectation de valeur débute par l'indice de tableau programmé. De cette façon, on peut affecter des valeurs à des tableaux partiels.

Exemple :

Code de programme	Commentaire
`DEF INT TABLEAU[5,5]`	`; Définition de tableau`
`TABLEAU[0,0]=SET(1,2,3,4,5)`	`; Affectation de valeur aux 5 premiers éléments de tableau [0,0] - [0,4]`
`FELD[0,0]=SET(1,2, , ,5)`	`; Affectation de valeur avec lacune aux 5 premiers éléments de tableau [0,0] - [0,4], éléments de tableau [0,2] et [0,3] = 0`

`TABLEAU[2,3]=SET(VARIABLE,4*5.6)`	`; Affectation de valeur avec une variable et une expression à partir de l'indice de tableau [2,3] :`
	`[2,3] = VARIABLE`
	`[2,4] = 4 * 5.6 = 22.4`

Informations complémentaires (REP)

Initialisation lors de la définition

- L'ensemble des éléments de tableau ou le nombre d'entre eux spécifié de manière facultative sont initialisés avec la valeur indiquée (constante).
- Les variables du type de données FRAME ne peuvent pas être initialisées.

Exemple :

Code de programme	Commentaire
`DEF REAL varName[10]=REP(3.5,4)`	`; Initialiser la définition de tableau et les éléments de tableau [0] à [3] avec la valeur 3,5`

Affectation de valeur pendant l'exécution du programme

Les règles décrites ci-dessus dans le cadre de la définition s'appliquent à l'affectation de valeur pendant l'exécution du programme. Il existe en outre les possibilités suivantes :

- Il est également possible d'utiliser des expressions comme éléments dans la liste de valeurs.
- L'affectation de valeur débute par l'indice de tableau programmé. De cette façon, on peut affecter des valeurs à des tableaux partiels.

Exemples :

Code de programme	Commentaire
`DEF REAL varName[10]`	`; Définition de tableau`
`varName[5]=REP(4.5,3)`	`; Eléments de tableau [5] à [7] = 4,5`
`R10=REP(2.4,3)`	`; Paramètres R R10 à R12 = 2,4`
`DEF FRAME FRM[10]`	`; Définition de tableau`
`FRM[5] = REP(CTRANS (X,5))`	`; Eléments de tableau [5] à [9] = CTRANS(X,5)`

Voir aussi

Définition et initialisation de variables de tableau (DEF, SET, REP) (Page 56)

2.1.14 Types de données

Les types de données suivants sont disponibles dans la CN :

type de données	Signification	Plage de valeurs
INT	Valeur entière avec signe	-2147483648 ... +2147483647
REAL	Nombre réel (LONG REAL selon IEEE)	±(~2,2*10^{-308} … ~1,8*10^{+308})
BOOL	Valeur booléenne TRUE (1) et FALSE (0)	1, 0
CHAR	Caractère ASCII	Code ASCII 0 ... 255
STRING	Chaîne de caractères de longueur définie	au maximum 200 caractères (caractères spéciaux exclus)
AXIS	Descripteur d'axe/de broche	Descripteur d'axe de canal
FRAME	Indications géométriques pour une transformation statique de coordonnées (décalage, rotation, mise à l'échelle, fonction miroir)	---

Conversion implicite du type de données

Les conversions suivantes de types de données sont possibles et réalisées de façon implicite pour les affectations et transmissions de paramètres :

de ↓/ vers →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x
x : Réalisable sans restrictions o : Risque de perte de données par écrasement de la plage de valeurs ⇒ alarme ; arrondissement : valeur après la virgule ≥ 0,5 ⇒ arrondissement à la valeur supérieure, valeur après la virgule < 0,5 ⇒ arrondissement à la valeur inférieure &: valeur ≠ 0 ⇒ TRUE, valeur == 0 ⇒ FALSE			

Voir aussi

Variables (Page 23)

2.1.15 Vérification de l'existence d'une variable (ISVAR)

La fonction prédéfinie ISVAR permet de vérifier si une variable système ou une variable utilisateur (par ex. paramètre machine, donnée de réglage, variable système, variable générale telle que variable GUD) est connue dans la CN.

Variables

La variable à interroger a la structure suivante :

Variable sans dimension :	<Variable>
Variable à une dimension sans indice de tableau :	<Variable>[ ]
Variable à une dimension avec indice de tableau n :	<Variable>[<n>]
Variable à deux dimensions sans indice de tableau :	<Variable>[,]
Variable à deux dimensions avec indices de tableau n et m :	<Variable>[<n>,<m>]

Syntaxe

```
<Ergebnis>=ISVAR(<Variable>[<n>,<m>])
```

Signification

<Ergebnis> :	Valeur de retour		
	Type de données :	BOOL	
	Plage de valeurs :	1	Variable présente
		0	Variable inconnue
ISVAR :	Vérifie que la variable système/utilisateur est connue dans la CN.		
<Variable> :	Nom de la variable système/utilisateur		
	Type de données :	STRING	
<n> :	Indice de tableau de la première dimension (**facultatif**)		
	Type de données :	INT	
<m> :	Indice de tableau de la deuxième dimension (**facultatif**)		
	Type de données :	INT	

Conformément au paramètre de transfert, les points suivants sont vérifiés :

- Le nom est-il connu ?
- La variable est-elle un tableau ?
- S'agit-il d'un tableau à une ou deux dimensions ?
- L'indice de tableau se trouve-t-il dans la plage autorisée ?

Si tous ces points sont vérifiés, l'état TRUE (1) est signalé en retour.

Si l'un des point n'est pas vérifié ou en présence d'une erreur de syntaxe, FALSE (0) est signalé.

Exemples

Code de programme	**Commentaire**
DEF INT VAR1	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VAR1")	; Dans ce cas, IS_VAR est TRUE.

Code de programme	**Commentaire**
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VARARRAY[,]")	; Dans ce cas, IS_VAR est TRUE ; c'est un tableau à deux dimensions.
N20 IS_VAR=ISVAR("VARARRAY")	; IS_VAR est TRUE ; la variable existe.
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	; IS_VAR est FALSE ; l'indice de tableau est illicite.
N40 IS_VAR=ISVAR("VARARRAY[8,8")	; IS_VAR est FALSE ; il manque "]" (erreur de syntaxe).
N50 IS_VAR=ISVAR("VARARRAY[,8]")	; IS_VAR est TRUE ; l'indice de tableau est autorisé.
N60 IS_VAR=ISVAR("VARARRAY[8,]")	; IS_VAR est TRUE ; l'indice de tableau est autorisé.

Code de programme	Commentaire
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"	; Le paramètre de transfert est un paramètre machine ; IS_VAR est TRUE.

Code de programme	Commentaire
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("$P_EP")	; Dans ce cas, IS_VAR est TRUE.
N20 IS_VAR=ISVAR("$P_EP[X]")	; Dans ce cas, IS_VAR est TRUE.

2.1.16 Lecture des valeurs d'attribut / du type de données (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)

Les fonctions GETVARPHU, GETVARAP, GETVARLIM et GETVARDFT permettent de lire les valeurs d'attribut des variables système ou utilisateur, et GETVARTYP permet d'en lire le type de données.

Lecture de l'unité physique

Syntaxe :
<Ergebnis>=GETVARPHU(<Name>)

Signification :

<Ergebnis> :	Valeur numérique de l'unité physique		
	Type de données :	INT	
	Plage de valeurs :	voir le tableau dans "Attribut : unité physique (PHU) (Page 46)"	
		En cas d'erreur :	
		- 2	Le <nom> spécifié n'est pas associé à un paramètre système ni à une variable utilisateur.
GETVARPHU :	Lecture de l'unité physique d'une variable système ou utilisateur		
<Name> :	Nom de la variable système ou utilisateur		
	Type de données :	STRING	

Exemple :

Soit la variable GUD suivante définie dans la NC :

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

Code de programme	Commentaire
DEF INT result=0	
result=GETVARPHU("electric")	; Détermine l'unité physique correspondant à la variable GUD

Code de programme	Commentaire
`IF (result < 0) GOTOF error`	

Le résultat signalé en retour est la valeur 42 qui correspond à l'unité physique [kW].

Remarque

GETVARPHU permet par exemple de vérifier si, pour l'affectation de variables a = b, les deux variables possèdent l'unité physique attendue.

Lecture du droit d'accès :

Syntaxe :

`<Ergebnis>=GETVARAP(<Name>,<Zugriff>)`

Signification :

`<Ergebnis>` :	Niveau de protection pour l'<accès> spécifié		
	Type de données :	INT	
	Plage de valeurs :	0 ... 7	Voir "Attribut : droits d'accès (APR, APW, APRP, APWP, APRB, APWB) (Page 48)".
		En cas d'erreur :	
		- 1	pas d'accès en écriture (pertinent uniquement pour <accès> "`WP`" et "`WB`")
		- 2	Le `<nom>` spécifié n'est pas associé à un paramètre système ni à une variable utilisateur.
		- 3	valeur incorrecte pour `<accès>`
`GETVARAP` :	Lecture du droit d'accès à une variable système ou utilisateur		
`<Name>` :	Nom de la variable système ou utilisateur		
	Type de données :	STRING	
`<Zugriff>` :	Type d'accès		
	Type de données :	STRING	
	Plage de valeurs :	`"RP"`	Lecture par le programme pièce
		`"WP"`	Ecriture par le programme pièce
		`"RB"`	Lecture par OPI
		`"WB"`	Ecriture par OPI

Exemple :

Code de programme	Commentaire
`DEF INT result=0`	
`result=GETVARAP("$TC_MAP8","WB")`	`; Détermine la protection de l'accès pour le paramètre système "position de magasin" pour la lecture par OPI.`
`IF (result < 0) GOTOF error`	

Le résultat signalé en retour est la valeur 7 qui correspond à la position 0 du commutateur à clé (= accès non protégé).

Remarque

GETVARAP permet par exemple de réaliser un programme de vérification qui contrôle les droits d'accès attendus par l'application.

Lecture des valeurs limites

Syntaxe :
`<Status>=GETVARLIM(<Name>,<Grenzwert>,<Ergebnis>)`

Signification :

`<Status>` :	Etat de la fonction		
	Type de données :	INT	
	Plage de valeurs :	1	OK
		-1	pas de valeur limite définie (pour les variables de type AXIS, STRING, FRAME)
		-2	Le `<nom>` spécifié n'est pas associé à un paramètre système ni à une variable utilisateur.
		-3	valeur incorrecte pour `<valeur limite>`
`GETVARLIM` :	Lecture de la valeur limite inférieure ou supérieure d'une variable système ou utilisateur		
`<Name>` :	Nom de la variable système ou utilisateur		
	Type de données :	STRING	
`<Grenzwert>` :	Indique la valeur limite à lire		
	Type de données :	CHAR	
	Plage de valeurs :	`"L"` :	= valeur limite inférieure
		`"U"` :	= valeur limite supérieure
`<Ergebnis>` :	Valeur limite retournée		
	Type de données :	VAR REAL	

Exemple :

Code de programme	**Commentaire**
`DEF INT state=0`	
`DEF REAL result=0`	
`state=GETVARLIM("$MA_MAX_AX_VELO","L",result)`	`; Détermine la valeur limite inférieure du MD32000 $MA_MAX_AX_VELO.`

Code de programme	Commentaire
IF (result < 0) GOTOF error	

Lecture d'attributs / du type de données

Syntaxe :

```
<Ergebnis>=GETVARDIM( <Name>, Index)
```

Signification :

<Résultat> :	Dimension/nombre du tableau <indice>	
	Type de données :	INT
GETVARDIM :	Lecture de la valeur limite inférieure ou supérieure d'une variable système ou utilisateur	
<Name> :	Lecture du nombre d'éléments du tableau	
	Type de données :	STRING
<Index> :	Numéro du tableau, max. 3.	
	Type de données :	INT

Exemple :

Code de programme	Commentaire
N5 DEF REAL myReal[5,4]	
N10 R1 = GETVATDIM("myReal",1)	R1 = 5
N15 R2 = GETVATDIM(„myReal“,2)	R2 = 4

Lecture de la valeur par défaut

Syntaxe :

```
<Status>=GETVARDFT(<Name>,<Ergebnis>[,<Index_1>,<Index_2>,<Index_3>]
)
```

Signification :

<Status> :	Etat de la fonction		
	Type de données :	INT	
	Plage de valeurs :	1	OK
		-1	pas de valeur par défaut disponible (par ex. parce que <résultat> n'a pas le type correspondant à <nom>)
		-2	Le <nom> spécifié n'est pas associé à un paramètre système ni à une variable utilisateur.
		-3	valeur incorrecte pour <indice_1>, dimension inférieure à un (= pas de tableau = scalaire)
		-4	valeur incorrecte pour <indice_2>
		-5	valeur incorrecte pour <indice_3>
GETVARDFT :	Lecture de la valeur par défaut d'une variable système ou utilisateur		
<Name> :	Nom de la variable système ou utilisateur		
	Type de données :	STRING	
<Ergebnis> :	Valeur par défaut retournée		
	Type de données :	VAR REAL (pour la lecture de la valeur par défaut de variables de type INT, REAL, BOOL, AXIS)	
		VAR STRING (pour la lecture de la valeur par défaut de variables de type STRING et CHAR)	
		VAR FRAME (pour la lecture de la valeur par défaut de variables de type FRAME)	
<Index_1> :	Indice de la première dimension (facultatif)		
	Type de données :	INT	
	Non programmé = 0		
<Index_2> :	Indice de la deuxième dimension (facultatif)		
	Type de données :	INT	
	Non programmé = 0		
<Index_3> :	Indice de la troisième dimension (facultatif)		
	Type de données :	INT	
	Non programmé = 0		

Exemple :

Code de programme	**Commentaire**
DEF INT state=0	
DEF REAL resultR=0	; Variable pour l'acquisition des valeurs par défaut de type INT, REAL, BOOL, AXIS.

Code de programme	Commentaire
DEF FRAME resultF=0	; Variable pour l'acquisition des valeurs par défaut de type FRAME
IF (GETVARTYP("$MA_MAX_AX_VELO") <> 4) GOTOF error	
state=GETVARDFT("$MA_MAX_AX_VELO", resultR, AXTOINT(X))	; Détermine la valeur par défaut de l'axe "X".
IF (resultR < 0) GOTOF error	
IF (GETVARTYP("$TC_TP8") <> 3) GOTOF error	
state=GETVARDFT("$TC_TP8", resultR)	
IF (GETVARTYP("$P_UBFR") <> 7) GOTOF error	
state=GETVARDFT("$P_UBFR", resultF)	

Lecture du type de données

Syntaxe :

<Ergebnis>=GETVARTYP(<Name>)

Signification :

<Ergebnis> :	Type de données des variables système ou utilisateur spécifiées		
	Type de données :	INT	
	Plage de valeurs :	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		En cas d'erreur :	
		< 0	Le <nom> spécifié n'est pas associé à un paramètre système ni à une variable utilisateur.
GETVARTYP :	Lecture du type de données d'une variable système ou utilisateur		
<Name> :	Nom de la variable système ou utilisateur		
	Type de données :	STRING	

Exemple :

Code de programme	Commentaire
DEF INT result=0	
DEF STRING name="R"	
result=GETVARTYP(name)	; Détermine le type du paramètre R.

Code de programme	**Commentaire**
`IF (result < 0) GOTOF error`	

Le résultat signalé en retour est la valeur 4 qui correspond au type de données REAL.

2.2 Programmation indirecte

2.2.1 Programmation indirecte d'adresses

Dans le cas de la programmation indirecte d'adresses, l'adresse étendue (<indice >) est remplacé par une variable de type approprié.

Remarque

La programmation indirecte d'adresses est impossible pour :

- N (numéro de bloc)
- L (sous-programme)
- Adresses réglables
  (par exemple, X[1] au lieu de X1 n'est pas autorisé)

Syntaxe

```
<ADRESSE>[<indice>]
```

Signification

`<ADRESSE>[...]` :	Adresse fixe avec extension (indice)
`<indice>` :	Variable telle que n° de broche, axe, etc.

Exemples

Exemple 1 : Programmation indirecte d'un numéro de broche

Programmation directe :

Code de programme	Commentaire
`S1=300`	`; Vitesse de rotation de 300 tr/min pour la broche avec le numéro 1.`

Programmation indirecte :

Code de programme	Commentaire
`DEF INT NUMBRO=1`	`; Définition des variables de type INT et affectation de valeur.`
`S[SPINU]=300`	`; Vitesse de rotation de 300 tr/min pour la broche dont le numéro est consigné dans la variable SPINU (dans cet exemple, la broche avec le numéro 1).`

Exemple 2 : Programmation indirecte d'un axe

Programmation directe :

Code de programme	Commentaire
FA[U]=300	; Avance de 300 pour l'axe "U".

Programmation indirecte :

Code de programme	Commentaire
DEF AXIS AXVAR2=U	; Définition d'une variable de type AXIS et affectation de valeur.
FA[AXVAR2]=300	; Avance de 300 pour l'axe dont le nom d'adresse est consigné dans la variable AXVAR2.

Exemple 3 : Programmation indirecte d'un axe

Programmation directe :

Code de programme	Commentaire
$AA_MM[X]	; Lecture de la valeur de mesure du palpeur (SCM) de l'axe "X".

Programmation indirecte :

Code de programme	Commentaire
DEF AXIS AXVAR3=X	; Définition d'une variable de type AXIS et affectation de valeur.
$AA_MM[AXVAR3]	; Lecture de la valeur de mesure du palpeur (SCM) pour l'axe dont le nom est consigné dans la variable AXVAR3.

Exemple 4 : Programmation indirecte d'un axe

Programmation directe :

Code de programme
X1=100 X2=200

Programmation indirecte :

Code de programme	Commentaire
DEF AXIS AXVAR1 AXVAR2	; Définition de deux variables de type AXIS.
AXVAR1=(X1) AXVAR2=(X2)	; Affectation des noms d'axes.
AX[AXVAR1]=100 AX[AXVAR2]=200	; Déplacement des axes dont les noms d'adresse sont consignés dans les variables AXVAR1 et AXVAR2.

Exemple 5 : Programmation indirecte d'un axe

Programmation directe :

Code de programme
G2 X100 I20

Programmation indirecte :

Code de programme	Commentaire
`DEF AXIS AXVAR1=X`	`; Définition d'une variable de type AXIS et affectation de valeur.`
`G2 X100 IP[AXVAR1]=20`	`; Programmation indirecte du centre de l'axe dont le nom d'adresse est consigné dans la variable AXVAR1.`

Exemple 6 : Programmation indirecte d'éléments de tableau

Programmation directe :

Code de programme	Commentaire
`DEF INT TABLEAU1[4,5]`	`; Définition du tableau 1.`

Programmation indirecte :

Code de programme	Commentaire
`DEFINE DIM1 AS 4`	`; Les dimensions de tableau sont à indiquer comme valeurs fixes.`
`DEFINE DIM2 AS 5`	
`DEF INT TA-` `BLEAU[DIM1,DIM2]`	
`TABLEAU[DIM1-1,DIM2-1]=5`	

Exemple 7 : Appel indirect de sous-programme

Code de programme	Commentaire
`CALL "L" << R10`	`; Appel du programme dont le numéro figure dans R10 (concaténation de chaînes de caractères).`

2.2.2 Programmation indirecte d'instructions G

La programmation indirecte des instructions G permet une programmation efficace des cycles.

Syntaxe

`G[<groupe>]=<numéro>`

Signification

`G[...]` :	Instruction G avec extension (indice)	
`<groupe>` :	Paramètre d'indice : Groupe G	
	Type :	INT
`<numéro>` :	Variable du numéro d'instruction G	
	Type :	INT ou REAL

Remarque

En règle générale, seules les instructions G non déterminantes pour la syntaxe peuvent être programmées indirectement.

Parmi les instructions G déterminantes pour la syntaxe, seuls celles du groupe de fonctions G 1 sont possibles.
Les instructions G déterminantes pour la syntaxe des groupes de fonction 2, 3 et 4 ne sont pas possibles.

Remarque

La programmation indirecte des instructions G n'admet pas les fonctions arithmétiques. Le calcul requis du numéro d'instruction G doit s'effectuer dans une ligne distincte du programme pièce, avant la programmation indirecte de l'instruction G.

Exemples

Exemple 1 : Décalage d'origine réglable (groupe G 8)

Code de programme	Commentaire
N1010 DEF INT INT_VAR	
N1020 INT_VAR=2	
...	
N1090 G[8]=INT_VAR G1 X0 Y0	;G54
N1100 INT_VAR=INT_VAR+1	; Calcul d'instruction G
N1110 G[8]=INT_VAR G1 X0 Y0	;G55

Exemple 2 : Sélection de niveau (groupe G 6)

Code de programme	Commentaire
N2010 R10=$P_GG[6]	; Lire l'instruction G active du Groupe G 6
...	
N2090 G[6]=R10	

Bibliographie

Pour des informations sur les groupes G, voir :
Manuel de programmation Notions de base, chapitre "Groupes G"

2.2.3 Programmation indirecte d'attributs de position (GP)

Les attributs de position tels que la programmation relative ou absolue de la position d'axe peuvent être programmés indirectement en tant que variables, en liaison avec le mot clé `GP`.

Application

La programmation indirecte d'attributs de position s'utilise dans les **cycles de substitution**, car elle présente alors les avantages suivants par rapport à la programmation des attributs de position comme mot clé (IC, AC, etc.)

Avec la programmation indirecte comme variables, **aucune** instruction CASE n'est nécessaire pour accéder à tous les attributs de position possibles.

Syntaxe

```
<INSTRUCTION DE POSITIONNEMENT>[<axe/broche>]=
GP(<position>,<attribut de position>)
<Axe/broche>=GP(<position>,<attribut de position>)
```

Signification

<INSTRUCTION DE POSITIONNEMENT>[] :	Les instructions de positionnement suivantes peuvent être programmées avec le mot clé GP : POS, POSA, SPOS, SPOSA Autres possibilités : • Tous les descripteurs d'axe/broche présents dans le canal : <Axe/broche> • Descripteur d'axe/broche variable AX
<axe/broche> :	Axe/broche à positionner
GP() :	Mot clé pour le positionnement
<position> :	Paramètre 1 Position d'axe/broche comme constante ou variable
<attribut de position> :	Paramètre 2 Attribut de position (par exemple le mode d'accostage de la position) comme variable (par exemple $P_SUB_SPOSMODE) ou comme mot clé (IC, AC, etc.)

Les valeurs fournies par les variables ont la signification suivante :

Valeur	Signification	Admis pour :
0	Aucune modification de l'attribut de position	
1	AC	POS, POSA, SPOS, SPOSA, AX, adresse d'axe
2	IC	POS, POSA, SPOS, SPOSA, AX, adresse d'axe
3	DC	POS, POSA, SPOS, SPOSA, AX, adresse d'axe
4	ACP	POS, POSA, SPOS, SPOSA, AX, adresse d'axe
5	ACN	POS, POSA, SPOS, SPOSA, AX, adresse d'axe
6	OC	-
7	PC	-
8	DAC	POS, POSA, AX, adresse d'axe
9	DIC	POS, POSA, AX, adresse d'axe
10	RAC	POS, POSA, AX, adresse d'axe

Valeur	Signification	Admis pour :
11	RIC	POS, POSA, AX, adresse d'axe
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

Exemple

En cas de couplage actif de broches synchrones entre la broche pilote S1 et la broche asservie S2, l'instruction SPOS appelle le cycle de substitution suivant pour le positionnement des broches dans le programme principal.

Le positionnement est effectué par l'instruction de N2230 :
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE) SPOS[2]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)

La position à accoster est lue dans la variable système $P_SUB_SPOSIT et le mode d'accostage de la position dans la variable système $P_SUB_SPOSMODE.

```
N1000 PROC LANG_SUB DISPLOF SBLOF
...
N2100 IF($P_SUB_AXFCT==2)
N2110                                   ; Substitution de l'instruction
                                        SPOS / SPOSA / M19 en cas de couplage ac-
                                        tif de broches synchrones
N2185 DELAYFSTON                        ; Début de la plage d'arrêt temporisé
N2190 COUPOF(S2,S1)                     ; Désactivation du couplage de broches
                                        synchrones
N2200                                   ; Positionnement de la broche pilote et
                                        de la broche asservie
N2210 IF($P_SUB_SPOS==TRUE) OR
($P_SUB_SPOSA==TRUE)
N2220                                   ; Positionnement de broche avec SPOS :
N2230    SPOS[1]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
         SPOS[2]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
N2250 ELSE
N2260                                   ; Positionnement de broche avec M19 :
N2270 M1=19 M2=19                       ; Positionnement de la broche pilote et
                                        de la broche asservie
N2280 ENDIF
N2285 DELAYFSTOF                        ; Fin de la plage d'arrêt temporisé
N2290 COUPON(S2,S1)                     ; Activation du couplage de broches syn-
                                        chrones
N2410 ELSE
```

Code de programme	Commentaire
N2420	; Interrogation de l'existence d'autres substitutions
...	
N3300 ENDIF	
...	
N9999 RET	

Autres conditions à prendre en compte

- Dans les actions synchrones, la programmation indirecte des attributs de position est impossible.

Bibliographie

Description fonctionnelle Fonctions de base ; GMFC, Canal, Mode de programme, Comportement au reset (K1),
chapitre : Substitution de fonctions CN par des sous-programmes

2.2.4 Programmation indirecte de lignes de programme pièce (EXECSTRING)

Une variable chaîne de caractères créée au préalable peut être exécutée en tant que ligne de programme pièce au moyen de l'instruction `EXECSTRING`.

Syntaxe

`EXECSTRING` est programmée dans une ligne spécifique du programme pièce :
`EXECSTRING (<variable de type STRING>)`

Signification

`EXECSTRING` :	Instruction pour l'exécution d'une variable chaîne de caractères en tant que ligne de programme pièce
`<variable de type STRING>` :	Variable du type STRING dans laquelle figure la ligne de programme pièce à exécuter proprement dite

Remarque

Tous les éléments susceptibles d'être programmés **dans la partie exécutable d'un programme pièce** peuvent être transmis avec `EXECSTRING`, à l'exception des structures de contrôle (Page 121). Les instructions `PROC` et `DEF` et, d'une manière générale, l'utilisation dans des fichiers INI et DEF en sont exclues.

Exemple

Code de programme	**Commentaire**
N100 DEF STRING[100] MY_BLOCK	; Définition des variables chaînes de caractères pour la prise en compte des lignes de programme pièce à exécuter.
N110 DEF STRING[10] MFCT1="M7"	
...	
N200 EXECSTRING(MFCT1 << "M4711")	; Exécution de la ligne "M7 M4711" du programme pièce.
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF(R10)	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(MY_BLOCK)	; Exécution de la ligne "M3 M7" du programme pièce.

2.3 Fonctions de calcul

Opérateur arithmétique / fonction de calcul	Signification
+	Addition
–	Soustraction
*	Multiplication
/ [1)]	Division[1)]
DIV[1)]	Division entière[1)]
MOD[1)]	Division modulo (fournit le reste de la division entière)[1)]
:	Opérateur de concaténation pour variables de type FRAME
SIN()	Sinus
COS()	Cosinus
TAN()	Tangente
ASIN()	Arc sinus
ACOS()	Arc cosinus
ATAN2(,)[1)]	Arc tangente 2[1)]
SQRT( )	Racine carrée
ABS( )	Valeur absolue
POT( )	Puissance 2 (carré)
TRUNC( )	Partie entière Précision réglable avec TRUNC dans les instructions relationnelles (voir "Correction de la précision pour les erreurs relationnelles (TRUNC) (Page 85)")
ROUND()	arrondir à un nombre entier
LN( )	Logarithme naturel
EXP( )	Fonction exponentielle
MINVAL()	valeur la plus petite de deux variables (voir "Minimum, maximum et plage de variables (MINVAL, MAXVAL, BOUND) (Page 87)")
MAXVAL()	valeur la plus grande de deux variables (voir "Minimum, maximum et plage de variables (MINVAL, MAXVAL, BOUND) (Page 87)")
BOUND()	valeur de variable se trouvant dans la plage de valeurs définie (voir "Minimum, maximum et plage de variables (MINVAL, MAXVAL, BOUND) (Page 87)")
CTRANS()	Décalage
CROT ()	Rotation
CSCALE ()	Changement d'échelle
CMIRROR ()	Fonction miroir
1) Voir le paragraphe "Exemples"	

Programmation

La notation mathématique usuelle est valable pour les fonctions de calcul. Les priorités de traitement sont fixées par des parenthèses. Les fonctions trigonométriques et leurs inverses emploient la notation en degrés (angle droit = 90°).

Exemples

Division : /

(Type REAL) = (Type INT ou type REAL) / (Type INT ou type REAL) ;

Exemple : 3 / 4 = 0.75

Division entière : DIV

(Type INT) = (Type INT ou REAL) / (Type INT ou REAL) ;

Exemple : 7 DIV 4,1 = 1

Division modulo (fournit le reste de la division entière) : MOD

(Type REAL) = (Type INT ou REAL) MOD (Type INT ou REAL) ;

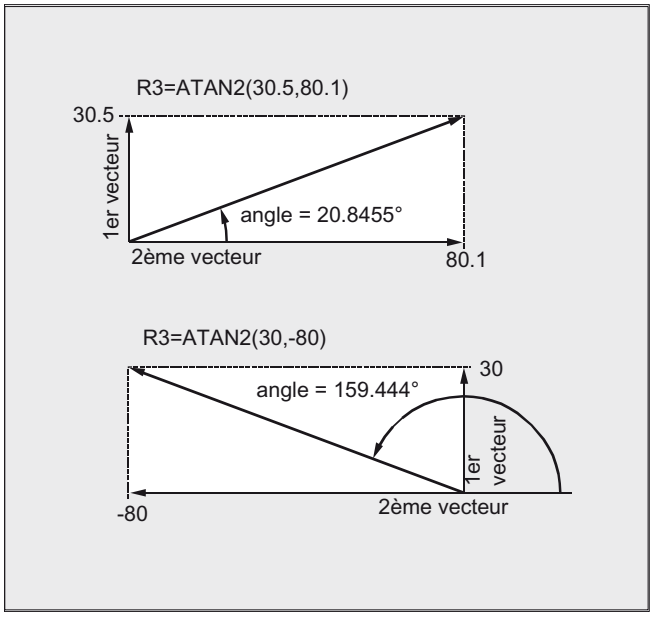
Exemple : 7 MOD 4,1 = 2,9

Arc tangente 2 : ATAN2

A partir de deux vecteurs orthogonaux, la fonction de calcul ATAN2 détermine l'angle du vecteur somme.

Le résultat s'inscrit dans la plage des quatre quadrants (-180° < 0 < +180°).

La base de la référence angulaire est toujours représentée par la 2e valeur dans le sens positif.



Exemples de programmation

Code de programme	**Commentaire**
R1=R1+1	; Nouveau R1 = ancien R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; La multiplication est prioritaire.
R14=(R1+R2)*R3	; Les expressions entre parenthèses sont prioritaires.
R15=SQRT(POT(R1)+POT(R2))	; Les parenthèses internes sont levées en premier : R15 = racine carrée ((R1^2 + R2^2))
RESFRAME=FRAME1:FRAME2 FRAME3=CTRANS(…):CROT(…)	; Combinaison FRAME avec opérateur de concaténation Affectation de valeurs à une composante FRAME

2.4 Opérations relationnelles et opérations logiques

Les opérations relationnelles peuvent servir par exemple à formuler une condition de saut dans le programme. Elles s'appliquent également aux expressions complexes.

Les opérations relationnelles sont applicables aux variables de type `CHAR`, `INT`, `REAL` et `BOOL`. Avec le type `CHAR`, la valeur de code est comparée.
Pour les types `STRING`, `AXIS` et `FRAME`, sont possibles : == et <> qui peuvent aussi être employés pour les opérations de type `STRING`, y compris dans des actions synchrones.

Le résultat d'opérations relationnelles est toujours du type `BOOL`.

Les opérateurs logiques servent à la combinaison de valeurs booléennes.

Les opérations logiques s'appliquent exclusivement aux variables du type `BOOL`. Avec la conversion de type interne, ils sont également applicables aux types de données `CHAR`, `INT` et `REAL`.

Dans le cas des opérations logiques (booléennes), la convention suivante s'applique aux données du type `BOOL`, `CHAR`, `INT` et `REAL` :

- 0 correspond à : FALSE
- valeur différente de 0 correspond à : TRUE

Opérateurs logiques sur bits

Les variables des types `CHAR` et `INT` permettent également d'effectuer des opérations logiques sur bits. Le cas échéant, il est procédé à une conversion automatique de type.

Programmation

Opérateur relationnel	Signification
==	égal à
<>	différent de
>	supérieur à
<	inférieur à
>=	supérieur ou égal à
<=	inférieur ou égal à

Opérateur logique	Signification
AND	ET
OR	OU
NOT	Négation
XOR	OU exclusif

Opérateur logique sur bits	Signification
B_AND	ET sur bits
B_OR	OU sur bits
B_NOT	Négation sur bits
B_XOR	OU exclusif sur bits

Remarque

Des parenthèses placées dans les expressions arithmétiques permettent de fixer un ordre des opérations différent des règles normales de priorité.

Remarque

Entre les opérandes booléens et les opérateurs doivent figurer des espacements.

Remarque

L'opérateur `B_NOT` ne se rapporte qu'à un seul opérande qui suit l'opérateur.

Exemples

Exemple 1 : opérateurs relationnels

```
IF R10>=100 GOTOF DESTINATION
```

ou

```
R11=R10>=100
IF R11 GOTOF DESTINATION
```

Le résultat de la comparaison `R10>=100` est d'abord mémorisé dans `R11`.

Exemple 2 : opérateurs logiques

```
IF (R10<50) AND ($AA_IM[X]>=17.5) GOTOF DESTINATION
```

ou

```
IF NOT R10 GOTOB DEPART
```

L'opérateur `NOT` ne se rapporte qu'à un seul opérande.

Exemple 3 : opérateurs logiques sur bits

```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTOF ACT_PLANE
```

2.5 Correction de la précision pour les erreurs relationnelles (TRUNC)

L'instruction TRUNC ampute l'opérande de la multiplication par un facteur de précision.

Précision réglable dans les instructions relationnelles

En interne, les données de type REAL du programme pièce sont représentées par 64 bits en format IEEE. De ce fait, les nombres décimaux ne peuvent pas être représentés de manière précise et leur comparaison avec des valeurs calculées de façon idéale peut conduire à des résultats inattendus.

Egalité relative

Afin que les imprécisions causées par le format de représentation ne faussent pas le flux du programme, les instructions relationnelles ne vérifient pas l'égalité absolue, mais l'égalité relative.

Syntaxe

Correction de la précision pour les erreurs relationnelles

```
TRUNC (R1*1000)
```

Signification

`TRUNC` :	troncature des décimales

Egalité relative de 10^{-12} pour

- Egalité : (==)
- Différence : (<>)
- Supérieur ou égal : (>=)
- Inférieur ou égal : (<=)
- Supérieur/inférieur : (><) avec égalité absolue
- Supérieur : (>)
- Inférieur : (<)

Compatibilité

Pour des raisons de compatibilité, le contrôle d'égalité relative de (>) et (<) peut être désactivé par définition du paramètre machine MD10280 $MN_ PROG_FUNCTION_MASK Bit0 = 1.

Remarque

Pour les raisons qui ont été citées, les comparaisons avec des données du type REAL connaissent d'une manière générale une certaine imprécision. Lorsque les écarts ne sont pas acceptables, vous devez effectuer des calculs de type INTEGER en multipliant les opérandes par un facteur de précision, puis en les amputant avec TRUNC.

Actions synchrones

Ce qui a été décrit pour les instructions relationnelles est aussi valable pour les actions synchrones.

Exemples

Exemple 1 : considérations relatives à la précision

Code de programme	Commentaire
N40 R1=61.01 R2=61.02 R3=0.01	; affectation des valeurs initiales
N41 IF ABS(R2-R1) > R3 GOTOF ERREUR	; Le saut serait exécuté jusqu'ici.
N42 M30	; Fin du programme
N43 ERREUR : SETAL(66000)	
R1=61.01 R2=61.02 R3=0.01	; affectation des valeurs initiales
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000) R13=TRUNC(R3*1000)	; Correction de la précision
IF ABS(R12-R11) > R13 GOTOF ERREUR	; Le saut n'est plus exécuté.
M30	; Fin du programme
ERREUR : SETAL(66000)	

Exemple 2 : formation et analyse du quotient des deux opérandes

Code de programme	Commentaire
R1=61.01 R2=61.02 R3=0.01	; affectation des valeurs initiales
IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF ERREUR	; Le saut n'est pas exécuté.
M30	; Fin du programme
ERREUR : SETAL(66000)	

2.6 Minimum, maximum et plage de variables (MINVAL, MAXVAL, BOUND)

Les instructions MINVAL et MAXVAL permettent de comparer entre elles les valeurs de deux variables. Le résultat fourni en retour la valeur la plus petite (pour MINVAL) ou la valeur la plus grande (pour MAXVAL).

L'instruction BOUND permet de vérifier si la valeur d'une variable à tester se trouve dans une plage de valeurs définie.

Syntaxe

```
<valeur la plus petite>=MINVAL(<variable1>,<variable2>)
<valeur la plus grande>=MAXVAL(<variable1>,<variable2>)
<valeur de retour>=<BOUND>(<minimum>,<maximum>,<variable à tester>)
```

Signification

MINVAL :	Détermine la **plus petite** parmi deux variables (<variable1>, <variable2>)
<valeur la plus petite> :	Variable résultante pour l'instruction MINVAL Est réglée à la valeur la plus petite de la variable.
MAXVAL :	Détermine la **plus grande** parmi deux variables (<variable1>, <variable2>)
<valeur la plus grande> :	Variable résultante pour l'instruction MAXVAL Est réglée à la valeur la plus grande de la variable.
BOUND :	Vérifie si une variable (<variable à tester>) se trouve dans la plage de valeurs définie.
<minimum> :	Variable définissant la valeur minimale de la plage de valeurs
<maximum> :	Variable définissant la valeur maximale de la plage de valeurs
<valeur de retour> :	Variable résultante pour l'instruction BOUND Si la valeur de la variable à tester se trouve dans la plage de valeurs définie, la variable résultante est réglée à la valeur de la variable à tester. Si la valeur de la variable à tester est supérieure à la valeur maximale, la variable résultante est réglée à la valeur maximale de la plage de définition. Si la valeur de la variable à tester est inférieure à la valeur minimale, la variable résultante est réglée à la valeur minimale de la plage de définition.

Remarque

MINVAL, MAXVAL et BOUND peuvent également être programmées dans actions synchrones.

Remarque

Comportement en cas d'égalité

En cas d'égalité de MINVAL/MAXVAL, cette valeur identique est fournie. Avec BOUND, la valeur de la variable à tester est de nouveau fournie.

Exemple

Code de programme	**Commentaire**
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4,	rVar5, rValMin, rValMax, rRetVar
rValMin=MINVAL(rVar1,rVar2)	; rValMin est réglée à la valeur 10.5.
rValMax=MAXVAL(rVar1,rVar2)	; rValMax est réglée à la valeur 33.7.
rVar3=19.7	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 se trouve dans les limites, rRetVar est réglée à 19.7.
rVar3=1.8	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 se situe en-deçà de la limite minimale, rRetVar est réglée à 10.5.
rVar3=45.2	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 se situe au-dessus de la limite maximale, rRet-Var est réglée à 33.7.

2.7 Priorité des opérations

A chaque opérateur est affectée une priorité. Lors du traitement d'une expression, les opérateurs au degré de priorité le plus élevé sont appliqués en premier lieu. En cas d'opérateurs d'égale priorité, le traitement s'opère de gauche à droite.

Des parenthèses placées dans les expressions arithmétiques permettent de fixer un ordre des opérations différent des règles normales de priorité.

Ordre des opérateurs

De la priorité la plus élevée à la moins élevée

1.	NOT, B_NOT	Négation, négation sur bits
2.	*, /, DIV, MOD	Multiplication, division
3.	+, –	Addition, soustraction
4.	B_AND	ET sur bits
5.	B_XOR	OU exclusif sur bits
6.	B_OR	OU sur bits
7.	AND	ET
8.	XOR	OU exclusif
9.	OR	OU
10.	<<	Concaténation de chaînes de caractères, type de résultat STRING
11.	==, <>, >, <, >=, <=	Opérateurs relationnels

Remarque

L'opérateur de concaténation ":" pour frames ne doit pas figurer avec d'autres opérateurs dans une même expression. Cet opérateur ne nécessite donc pas d'ordre de priorité.

Exemple d'instruction If

```
If (otto==10) and (anna==20) gotof end
```

2.8 Conversions de types possibles

Fonction

Conversion de type lors d'une affectation

La valeur numérique constante, la variable ou l'expression assignée à une variable doit être compatible avec le type de cette variable. Si cette condition n'est pas remplie, le type est automatiquement converti au moment de l'affectation.

Conversions de type possibles

en	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
de							
REAL	oui	oui*	oui[1])	oui*	–	–	–
INT	oui	oui	oui[1])	oui[2])	–	–	–
BOOL	oui	oui	oui	oui	oui	–	–
CHAR	oui	oui	oui[1])	oui	oui	–	–
STRING	–	–	oui[4])	oui[3])	oui	–	–
AXIS	–	–	–	–	–	oui	–
FRAME	–	–	–	–	–	–	oui

Observations

* En cas de conversion du type REAL en type INT, une partie décimale >= 0,5 est arrondie par excès ; dans le cas contraire, elle est arrondie par défaut (voir fonction ROUND)

[1]) La valeur <> 0 équivaut à TRUE, la valeur == 0 équivaut à FALSE

[2]) Si la valeur se situe dans la plage numérique admise

[3]) Dans le cas d'un seul caractère

[4]) Longueur de chaîne de caractères 0 = >FALSE, sinon TRUE

Remarque

Un message d'erreur est émis lorsqu'une valeur de conversion est supérieure à la plage cible.

Lorsque des types différents figurent dans une même expression, leur adaptation s'effectue automatiquement. Des conversions de type sont aussi possibles dans des actions synchrones, voir le chapitre Actions synchrones au déplacement, conversion de type implicite".

2.9 Opérations sur les chaînes de caractères

Opérations sur les chaînes de caractères

Outre les opérations classiques "Affectation" et "Comparaison", les opérations suivantes sont possibles sur les chaînes de caractères :

- Conversion de type en type STRING (AXSTRING) (Page 91)
- Conversion de type à partir de STRING (NUMBER, ISNUMBER, AXNAME) (Page 92)
- Concaténation de chaînes de caractères (<<) (Page 93)
- Conversion en minuscules/majuscules (TOLOWER, TOUPPER) (Page 94)
- Déterminer la longueur d'une chaîne de caractères (STRLEN) (Page 95)
- Recherche de caractères/chaînes de caractères dans une chaîne (INDEX, RINDEX, MINDEX, MATCH) (Page 96)
- Sélection d'une chaîne de caractères partielle (SUBSTR) (Page 97)
- Lecture et écriture de caractères individuels (Page 98)
- Formatage d'une chaîne (SPRINT) (Page 99)

Signification particulière du caractère 0

En interne, le caractère 0 est interprété comme identification de fin d'une chaîne de caractères. Le remplacement d'un caractère par 0 raccourcit donc la chaîne de caractères.

Exemple :

Code de programme	**Commentaire**
`DEF STRING[20] STRG="axe . à l'arrêt"`	
`STRG[6]="X"`	
`MSG(STRG)`	`; Fournit le message "Axe X à l'arrêt".`
`STRG[6]=0`	
`MSG(STRG)`	`; Fournit le message "Axe".`

2.9.1 Conversion de type en type STRING (AXSTRING)

La fonction "Conversion de type en type STRING" permet d'utiliser des variables de types différents comme éléments d'un message (MSG).

Lors de l'utilisation de l'opérateur <<, la fonction est exécutée implicitement pour les données de type INT, REAL, CHAR et BOOL (voir " Concaténation de chaînes de caractères (<<) (Page 93) ").

Une valeur INT est convertie sous la forme normale lisible. Les valeurs REAL sont données avec un nombre de décimales pouvant aller jusqu'à 10.

L'instruction `AXSTRING` convertit des variables de type AXIS en type STRING.

Syntaxe

```
<STRING_ERG> = << <type_quelconque>
<STRING_ERG> = AXSTRING(<descripteur d'axe>)
```

Signification

<STRING_ERG> :	Variable du résultat de la conversion de type	
	Type :	STRING
<Type_quelconque> :	Types de variables INT, REAL, CHAR, STRING et BOOL	
AXSTRING :	L'instruction AXSTRING fournit le descripteur d'axe indiqué comme chaîne de caractères.	
<descripteur d'axe> :	Variable du descripteur d'axe	
	Type :	AXIS

Remarque

Les variables FRAME ne peuvent pas être converties.

2.9.2 Conversion de type à partir de STRING (NUMBER, ISNUMBER, AXNAME)

Avec l'instruction NUMBER, la conversion se fait du type STRING en type REAL. L'instruction ISNUMBER s'utilise pour vérifier si la conversion est possible.

L'instruction AXNAME convertit une chaîne de caractères de type STRING en type AXIS.

Syntaxe

```
<RES_REAL>=NUMBER("<chaîne de caractères>")
<RES_BOOL>=ISNUMBER("<chaîne de caractères>")
<RES_AXIS>=AXNAME("<chaîne de caractères>")
```

Signification

NUMBER :	L'instruction NUMBER donne en retour le nombre représenté par la <chaîne de caractères> comme valeur REAL.	
<chaîne> :	Variable de type STRING à convertir	
<REAL_ERG> :	Variable du résultat de la conversion de type avec NUMBER	
	Type :	REAL
ISNUMBER :	L'instruction ISNUMBER permet de contrôler si la <chaîne de caractères> est convertible en un nombre valable.	

<BOOL_ERG> :	Variable du résultat de l'interrogation avec ISNUMBER		
	Type :	BOOL	
	Valeur :	TRUE	ISNUMBER fournit la valeur TRUE quand la <chaîne de caractères> représente un nombre REAL valide selon les règles du langage.
		FALSE	Si ISNUMBER indique la valeur FALSE, une alarme est émise lors de l'appel de NUMBER avec la même <chaîne de caractères>.
AXNAME :	L'instruction AXNAME transforme la <chaîne de caractères> indiquée en un descripteur d'axe. **Remarque :** Une alarme est émise si la <chaîne de caractères> ne peut pas être affectée à un descripteur d'axe existant.		
<AXIS_ERG> :	Variable du résultat de la conversion de type avec AXNAME		
	Type :	AXIS	

Exemple

Code de programme	Commentaire
DEF BOOL BOOL_ERG	
DEF REAL REAL_ERG	
DEF AXIS AXIS_ERG	
RES_BOOL=ISNUMBER("1234.9876Ex-7")	; BOOL_ERG == TRUE
RES_BOOL=ISNUMBER("1234XYZ")	; BOOL_ERG == FALSE
RES_REAL=NUMBER("1234.9876Ex-7")	; REAL_ERG == 1234.9876Ex-7
RES_AXIS=AXNAME("X")	; AXIS_ERG == X

2.9.3 Concaténation de chaînes de caractères (<<)

La fonction "Concaténation de chaînes de caractères" permet de composer une chaîne de caractères à partir d'éléments individuels.

La concaténation s'effectue avec l'opérateur "**<<**". Cet opérateur possède, pour toutes les combinaisons des types de base, CHAR, BOOL, INT, REAL et STRING, comme type de destination STRING. Une conversion éventuellement indispensable est entreprise selon les règles établies.

Syntaxe

```
<Type_quelconque> << <type_quelconque>
```

Signification

<Type_quelconque> :	Variable de type CHAR, BOOL, INT, REAL ou STRING
<< :	Opérateur pour la concaténation de variables (<type_quelconque>) en chaîne de caractères composée (type STRING). Cet opérateur est également disponible seul en variante dite "unaire". permettant d'effectuer une conversion de type explicite en type STRING (à l'exception des types FRAME et AXIS) : << <type_quelconque>

Il est ainsi possible, par exemple, de composer un message ou un ordre à partir de listes de textes et d'insérer des paramètres (nom de bloc, etc.) :
`MSG(STRG_TAB[LOAD_IDX]<<NOM_BLOC)`

Remarque

Les résultats intermédiaires de la concaténation de la chaîne de caractères ne doivent pas dépasser la longueur de chaîne maximale.

Remarque

L'utilisation des types FRAME et AXIS est impossible avec l'opérateur "**<<**".

Exemples

Exemple 1 : Chaînage de caractères

```
Code de programme                                   Commentaire
DEF INT IDX=2
DEF REAL VALUE=9.654
DEF STRING[20] STRG="INDEX:2"
IF STRG=="Index:"<<IDX GOTOF NO_MSG
MSG("Index:"<<IDX<<"/valeur:"<<VALUE)               ; Affichage :
                                                    "Indice:2/valeur:9,654"
NO_MSG:
```

Exemple 2 : conversion explicite de type avec <<

```
Code de programme          Commentaire
DEF REAL VALUE=3.5
<<VALUE                    ; La variable spécifiée de type REAL est convertie en ty-
                           pe STRING.
```

2.9.4 Conversion en minuscules/majuscules (TOLOWER, TOUPPER)

La fonction "Conversion en minuscules/majuscules" permet d'harmoniser l'écriture au sein d'une chaîne de caractères.

Syntaxe

```
<RES_STRING>=TOUPPER("<chaîne de caractères>")
<RES_STRING>=TOLOWER("<chaîne de caractères>")
```

Signification

TOUPPER :	L'instruction TOUPPER convertit toutes les lettres d'une chaîne de caractères en **majuscules**.	
TOLOWER :	L'instruction TOLOWER convertit toutes les lettres d'une chaîne de caractères en **minuscules**.	
<chaîne> :	Chaîne de caractères à convertir	
	Type :	STRING
<STRING_ERG> :	Variable du résultat de la conversion	
	Type :	STRING

Exemple

Etant donné qu'il est possible de rattacher à l'interface utilisateur des indications introduites par l'utilisateur, cette fonctionnalité permet d'obtenir une présentation uniformisée en lettres minuscules ou majuscules:

Code de programme

```
DEF STRING [29] STRG
...
IF "LEARN.CNC"==TOUPPER(STRG) GOTOF LOAD_LEARN
```

2.9.5 Déterminer la longueur d'une chaîne de caractères (STRLEN)

L'instruction STRLEN permet de déterminer la longueur d'une chaîne de caractères.

Syntaxe

```
<RES_INT>=STRLEN("<chaîne de caractères>")
```

Signification

STRLEN :	L'instruction STRLEN détermine la longueur de la chaîne de caractères indiquée. Le résultat est le nombre de caractères comptés à partir du début de la chaîne et qui ne sont pas des caractères 0.	
<chaîne> :	Chaîne de caractères dont la longueur doit être déterminée	
	Type :	STRING
<INT_ERG> :	Variable du résultat de la détermination	
	Type :	INT

Exemple

En liaison avec l'accès à des caractères individuels, la fonction permet de déterminer la fin d'une chaîne de caractères :

Code de programme

```
IF (STRLEN(NOM_BLOC)>10) GOTOF ERREUR
```

2.9.6 Recherche de caractères/chaînes de caractères dans une chaîne (INDEX, RINDEX, MINDEX, MATCH)

Cette fonctionnalité permet de rechercher des caractères isolés ou une suite de caractères dans une chaîne de caractères. Les résultats signalent à quel endroit de la chaîne de caractères se trouve le caractère / la suite de caractères qui fait l'objet de la recherche.

Syntaxe

`INT_ERG=INDEX(STRING,CHAR)` ; **Type de résultat : INT**

`INT_ERG=RINDEX(STRING,CHAR)` ; **Type de résultat : INT**

`INT_ERG=MINDEX(STRING,STRING)` ; **Type de résultat : INT**

`INT_ERG=MATCH(STRING,STRING)` ; **Type de résultat : INT**

Sémantique

Fonctions de recherche : elles fournissent la position trouvée dans la chaîne de caractères (premier paramètre). Si le caractère ou la chaîne de caractères est introuvable, le résultat est -1, la position du premier caractère étant 0.

Signification

`INDEX` :	Recherche du caractère (depuis le début) indiqué comme deuxième paramètre dans le premier paramètre.
`RINDEX` :	Recherche du caractère (depuis la fin) indiqué comme deuxième paramètre dans le premier paramètre.
`MINDEX` :	Correspond à la fonction INDEX, sauf qu'elle fournit une liste de caractères (sous forme de chaîne) avec l'indice du premier caractère trouvé.
`MATCH` :	Recherche d'une chaîne de caractères dans une chaîne de caractères.

Il est ainsi possible de décomposer les chaînes de caractères suivant des critères définis tels que les espaces et les caractères de séparation des chemins ("/").

Exemple

Décomposition d'une entrée en noms de chemin et de bloc

Code de programme	**Commentaire**
`DEF INT CHEMIDX, PROGIDX`	

Code de programme	Commentaire
DEF STRING[26] ENTREE	
DEF INT LISTIDX	
ENTREE = "/_N_MPF_DIR/_N_EXECU- TE_MPF"	
LISTIDX = MINDEX (ENTREE, "M,N,O,P") + 1	; Le résultat de LISTIDX est la valeur 3, "N" étant le premier caractère du paramètre ENTREE dans la liste, depuis le début.
CHEMIDX = INDEX (ENTREE, "/") +1	; On a donc : CHEMIDX = 1
PROGIDX = RINDEX (ENTREE, "/") +1	; On a donc : PROGIDX = 12
	; Avec la fonction SUBSTR présentée au chapitre suivant, il est possible de décomposer la variable ENTREE en "chemin" et en "bloc" :
VARIABLE = SUBSTR (ENTREE, CHEMIDX, PROGIDX-CHEMIDX-1)	; Fournit alors "_N_MPF_DIR".
VARIABLE = SUBSTR (ENTREE, PROGIDX)	; Fournit alors "_N_EXECUTE_MPF".

2.9.7 Sélection d'une chaîne de caractères partielle (SUBSTR)

La fonction SUBSTRING permet de lire des sous-parties quelconques d'une chaîne de caractères.

Syntaxe

```
<STRING_ERG>=SUBSTR(<chaîne>,<indice>,<longueur>)
<STRING_ERG>=SUBSTR(<chaîne>,<indice>)
```

Signification

SUBSTR :	La fonction fournit une chaîne partielle d'une <chaîne> de caractères de la <longueur> spécifiée, à partir de <indice>. Si le paramètre <longueur> n'est pas spécifié, la fonction fournit une chaîne partielle qui va de l'<indice> à la fin de la chaîne de caractères.
<indice> :	Position de début de la chaîne partielle dans la chaîne de caractères. Si la position de début se situe après la fin de la chaîne de caractères, le résultat est une chaîne vide (""). Premier caractère de la chaîne : Indice = 0 Plage de valeurs : 0 ... (longueur de chaîne de caractères - 1)
<longueur> :	Longueur de la chaîne partielle. Si la longueur spécifiée est trop grande, la chaîne partielle retournée s'arrête à la fin de la chaîne de caractères. Plage de valeurs : 1 ... (longueur de chaîne de caractères - 1)

Exemple

Code de programme	Commentaire
DEF STRING[29] RES	

Code de programme	Commentaire
; 1	
; 0123456789012345678	
ERG = SUBSTR("ACQUITTEMENT : 10 à 99", 10, 2)	; ERG == "10"
ERG = SUBSTR("ACQUITTEMENT : 10 à 99", 10)	; ERG == "10 à 99"

2.9.8 Lecture et écriture de caractères individuels

Il est possible de lire ou d'écrire des caractères individuellement dans une chaîne.

Les conditions marginales suivantes sont à prendre en considération :

- Cette fonction ne concerne que les variables utilisateur ; elle est inapplicable aux variables système
- Les caractères individuels d'une chaîne ne sont transmis que par paramètres "call by value" lors d'appels de sous-programmes

Syntaxe

```
<caractère>=<chaîne>[<indice>]
<caractère>=<tableau chaîne>[<indice de tableau>,<indice>]
<chaîne>[<indice>]=<caractère>
<tableau chaîne>[<indice de tableau>,<indice>]=<caractère>
```

Signification

<chaîne> :	Chaîne de caractères quelconque
<caractère> :	Variable de type CHAR
<indice> :	Position du caractère dans la chaîne Premier caractère de la chaîne : Indice = 0 Plage de valeurs : 0 ... (longueur de chaîne de caractères - 1)

Exemples

Exemple 1 : variable Message

Code de programme	Commentaire
; 0123456789	
DEF STRING [50] MESSAGE = "Axe n a atteint sa position"	
MESSAGE[6] = "X"	
MSG(MESSAGE)	; "Axe X a atteint sa position"

Exemple 2 : évaluation d'une variable système

Code de programme	Commentaire
DEF STRING[50] STRG	; Mémoire tampon pour variable système
...	

Code de programme	Commentaire
STRG = $P_MMCA	; Chargement de la variable système
IF STRG[0] == "E" GOTO ...	; Evaluation de la variable système

Exemple 3 : transfert des paramètres "call by value" et "call by reference"

Code de programme	Commentaire
; 0123456	
DEF STRING[50] STRG = "axe X"	
DEF CHAR CHR	
...	
EXTERN UP_VAL(AXE)	; Définition du sous-programme avec le paramètre "call by value"
EXTERN UP_REF(VAR AXE)	; Définition du sous-programme avec le paramètre "call by ref."
...	
UP_VAL(STRG[6])	; Transfert du paramètre "by value"
...	
CHR = STRG[6]	; Mise en mémoire tampon
UP_REF(CHR)	; Transfert du paramètre "by reference"

2.9.9 Formatage d'une chaîne (SPRINT)

Avec la fonction prédéfinie SPRINT, des chaînes de caractères peuvent être formatées et préparées, par exemple, pour la sortie vers des périphériques externes (voir aussi "Process DataShare - Sortie sur un appareil/fichier externe (EXTOPEN, WRITE, EXTCLOSE) (Page 700)").

Syntaxe

```
"<Chaîne_résultat>"=SPRINT("<Chaîne_format>",<valeur_1>,<valeur_2>,...,
<valeur_n>)
```

Signification

SPRINT :	Descripteur d'une fonction prédéfinie qui fournit une valeur de type STRING.
"<Chaîne_format>" :	Chaîne de caractères contenant une partie fixe et une partie variable. Les parties variables sont définies à l'aide du caractère de commande de format % et d'une description de format qui suit.
< valeur_1>,< valeur_2>,…,< valeur_n> :	Valeur se présentant sous la forme d'une constante ou d'une variable CN qui est insérée à l'emplacement du n-ième caractère de commande %, en fonction de la description du format, dans la <Chaîne_format>.
"<Chaîne_résultat>" :	Chaîne de caractères formatée (maximum 400 octets)

Descriptions de format disponibles

%B :	Conversion en chaîne de caractères "TRUE" lorsque la valeur à convertir :
	• est différente de 0,
	• n'est pas une chaîne de caractères vide (dans le cas de valeurs de type String).
	Conversion en chaîne de caractères "FALSE" lorsque la valeur à convertir :
	• est égale à 0,
	• est une chaîne de caractères vide.
	Exemple :
	`N10 DEF BOOL BOOL_VAR=1`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF BOOL_VAR:TRUE".

%C :	Conversion en caractères ASCII.
	Exemple :
	`N10 DEF CHAR CHAR_VAR="X"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C",CHAR_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF CHAR_VAR:X".

%D :	Conversion en une chaîne de caractères comportant un nombre entier (INTEGER).
	Exemple :
	`N10 DEF INT INT_VAR=123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D",INT_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF INT_VAR:123".

%<m>D :	Conversion en une chaîne de caractères comportant un nombre entier (INTEGER). La longueur minimale de la chaîne est de <m> caractères. Les chiffres manquants sont remplis avec des caractères non significatifs alignés à gauche.
	Exemple :
	`N10 DEF INT INT_VAR=-123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D",INT_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF INT_VAR:xx-123" (dans l'exemple, "x" remplace un caractère non significatif).

%F :	Conversion en une chaîne de caractères comportant un nombre à 6 décimales. Les décimales peuvent être arrondies ou complétées par des 0 non significatifs.
	Exemple :
	`N10 DEF REAL REAL_VAR=-1.2341234EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR: -1234.123400".

%<m>F :	Conversion en une chaîne de caractères comportant un nombre à 6 décimales et ayant une longueur totale d'au moins <m> caractères. Les décimales peuvent être arrondies ou complétées par des 0 non significatifs. Les caractères manquants pour atteindre la longueur totale <m> sont remplis avec des caractères non significatifs alignés à gauche.
	Exemple :
	`N10 DEF REAL REAL_VAR=-1.23412345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR: xxx-1234.123457" (dans l'exemple, "x" remplace un caractère non significatif).

%.<n>F :	Conversion en une chaîne de caractères comportant un nombre à <n> décimales. Les décimales peuvent être arrondies ou complétées par des 0 non significatifs.
	Exemple :
	`N10 DEF REAL REAL_VAR=-1.2345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:-1234.568".

%<m>.<n>F :	Conversion en une chaîne de caractères comportant un nombre à <n> décimales et ayant une longueur totale d'au moins <m> caractères. Les décimales peuvent être arrondies ou complétées par des 0 non significatifs. Les caractères manquants pour atteindre la longueur totale <m> sont remplis avec des caractères non significatifs alignés à gauche.
	Exemple :
	`N10 DEF REAL REAL_VAR=-1.2341234567890EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:xx-1234.12" (dans l'exemple, "x" remplace un caractère non significatif).

%E :	Conversion en une chaîne de caractères comportant un nombre au format exponentiel. La mantisse est enregistrée de façon normalisée avec un chiffre avant le séparateur de décimales et 6 chiffres après. Les décimales peuvent être arrondies ou complétées par des 0 non significatifs. L'exposant commence par le mot clé "EX". Il est suivi du signe ("+" ou "-") et d'un nombre à deux ou trois chiffres.
	Exemple :
	`N10 DEF REAL REAL_VAR=-1234.567890`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:-1.234568EX+03".

%<m>E :	Conversion en une chaîne de caractères comportant un nombre décimal au format exponentiel avec une longueur totale d'au moins <m> caractères. Les caractères manquants sont remplis par des caractères non significatifs alignés à gauche. La mantisse est enregistrée de façon normalisée avec un chiffre avant le séparateur de décimales et 6 chiffres après. Les décimales peuvent être arrondies ou complétées par des 0 non significatifs. L'exposant commence par le mot clé "EX". Il est suivi du signe ("+" ou "-") et d'un nombre à deux ou trois chiffres.
	Exemple :
	`N10 DEF REAL REAL_VAR=-1234.5`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03" (dans l'exemple, "x" remplace un caractère non significatif).

%.<n>E :	Conversion en une chaîne de caractères comportant un nombre au format exponentiel. La mantisse est enregistrée de façon normalisée avec un chiffre avant le séparateur de décimales et <n> chiffres après. Les décimales peuvent être arrondies ou complétées par des 0 non significatifs. L'exposant commence par le mot clé "EX". Il est suivi du signe ("+" ou "-") et d'un nombre à deux ou trois chiffres.
	Exemple :
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:-1.23EX+03".

%<m>.<n>E :	Conversion en une chaîne de caractères comportant un nombre décimal au format exponentiel avec une longueur totale d'au moins <m> caractères. Les caractères manquants sont remplis par des caractères non significatifs alignés à gauche. La mantisse est enregistrée de façon normalisée avec un chiffre avant le séparateur de décimales et <n> chiffres après. Les décimales peuvent être arrondies ou complétées par des 0 non significatifs. L'exposant commence par le mot clé "EX". Il est suivi du signe ("+" ou "-") et d'un nombre à deux ou trois chiffres.
	Exemple :
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:xx-1.23EX+03" (dans l'exemple, "x" remplace un caractère non significatif).

%G :	Conversion en une chaîne de caractères comportant un nombre décimal en fonction de la plage de valeurs, au format décimal ou exponentiel. Si la valeur absolue à représenter est inférieure à 1.0EX-04 ou supérieure ou égale à 1.0EX+06, le format exponentiel est choisi, sinon c'est le format décimal. Etant donné qu'il n'est pas possible d'afficher plus de 6 caractères significatifs, le nombre est arrondi si nécessaire.
	Exemple au format décimal :
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:0.000123457".
	Exemple au format exponentiel :
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:1.23457EX+06".

%<m>G :	Conversion en une chaîne de caractères comportant un nombre décimal au format décimal ou exponentiel en fonction de la plage de valeurs (tel que %G). La longueur totale minimale de la chaîne de caractères est d'au moins <m> caractères. Les caractères manquants sont remplis par des caractères non significatifs alignés à gauche.
	Exemple au format décimal :
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:xxxx0.000123457" (dans l'exemple, "x" remplace un caractère non significatif).
	Exemple au format exponentiel :
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:xxx1.23457EX+06" (dans l'exemple, "x" remplace un caractère non significatif).

%.<n>G :	Conversion en une chaîne de caractères comportant un nombre décimal au format décimal ou exponentiel en fonction de la plage de valeurs. Etant donné qu'il n'est pas possible d'afficher plus de <n> caractères significatifs, le nombre est arrondi si nécessaire. Si la valeur absolue à représenter est inférieure à 1.0EX-04 ou supérieure ou égale à 1.0EX(+<n>), le format exponentiel est choisi, sinon c'est le format décimal.
	Exemple au format décimal :
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:0.000123".
	Exemple au format exponentiel :
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:1.23EX+03".

%<m>.<n>G :	Conversion en une chaîne de caractères comportant un nombre décimal au format décimal ou exponentiel en fonction de la plage de valeurs (tel que %<n>G). La longueur totale minimale de la chaîne de caractères est d'au moins <m> caractères. Les caractères manquants sont remplis par des caractères non significatifs alignés à gauche.
	Exemple au format décimal :
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:xxx0.0001235" (dans l'exemple, "x" remplace un caractère non significatif).
	Exemple au format exponentiel :
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF REAL_VAR:xx1.235EX+06" (dans l'exemple, "x" remplace un caractère non significatif).

%.<n>P :	Conversion d'une valeur REAL en valeur INTEGER en tenant compte du nombre de décimales <n>. La valeur entière INTEGER est transmise en sortie sous forme de nombre binaire sur 32 bits. Si la valeur à convertir ne peut pas être représentée sur 32 bits, le traitement est interrompu et une alarme est générée.
	Etant donné qu'une série d'octets produite avec l'instruction de formatage %.<n>P peut également contenir des zéros binaires, la chaîne de caractères complète obtenue ne correspond plus aux conventions du type de données CN STRING. Il ne peut donc ni être stocké dans une variable de type STRING ni être traité à l'aide des instructions de type String du langage CN. La seule utilisation possible est le transfert de paramètres à l'instruction WRITE avec sortie sur un périphérique externe correspondant (voir exemple ci-après).
	Dès que la <chaîne_format> contient une description de format de type %P, la chaîne de caractères complète est transmise en sortie, à l'exception du nombre binaire généré avec %.<n>P conformément au PM10750 $MN_SPRINT_FORMAT_P_CODE dans le code de caractères ASCII, ISO (DIN6024) ou EIA (RS244). Si un caractère non convertible est programmé, le traitement est interrompu et une alarme est générée.
	Exemple :
	`N10 DEF REAL REAL_VAR=123.45`
	`N20 DEF INT ERROR`
	`N30 DEF STRING[20] EXT_DEVICE="/ext/dev/1"`
	`...`
	`N100 EXTOPEN(ERROR,EXT_DEVICE)`
	`N110 IF ERROR <> 0`
	`... ; Traitement des erreurs`
	`N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)`
	`N210 IF ERROR <> 0`
	`… ; Traitement des erreurs`
	Résultat : La chaîne de caractères "INTEGER BINARY CODED: 'H0001E23A'" est transmise au périphérique de sortie /ext/dev/1. La valeur hexadécimale 0x0001E23A correspond à la valeur décimale 123450.

%<m>.<n>P :	Conversion d'une valeur REAL en fonction du réglage contenu dans le paramètre machine PM10751 $MN_SPRINT_FORMAT_P_DECIMA, dans une chaîne de caractères comportant :
	• un nombre entier de <m> + <n> chiffres **ou**
	• un nombre décimal comportant au maximum <m> chiffres avant le séparateur de décimales et exactement <n> après.
	Comme pour la description de format %.<n>P la chaîne de caractères entière est stockée dans le code de caractères défini par PM10750 $MN_SPRINT_FORMAT_P_CODE.
	Conversion pour PM10751 = 0 :
	La valeur réelle est convertie en une chaîne de caractères comportant un nombre entier de <m> + <n> chiffres. Le cas échéant, les chiffres après le séparateur de décimales sont arrondis à <n> ou complétés avec des 0. Les chiffres manquants avant la virgule sont remplis avec des caractères non significatifs. Le signe moins est ajouté à gauche et un espace est inséré à la place du signe plus.
	Exemple :
	`N10 DEF REAL REAL_VAR=-123.45`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "PUNCHED TAPE FORMAT:-xx123450" (dans l'exemple, "x" remplace un caractère non significatif).
	Conversion pour PM10751 = 1 :
	La valeur REAL est convertie en une chaîne de caractères comportant un nombre décimal avec au maximum <m> chiffres avant le séparateur de décimales et exactement <n> chiffres après . Le cas échéant, les chiffres avant le séparateur de décimales sont tronqués et les chiffres après le séparateur de décimales sont arrondis ou complétés avec des 0. Si <n> est égal à 0, le signe décimal est omis.
	Exemple :
	`N10 DEF REAL REAL_VAR1=-123.45`
	`N20 DEF REAL REAL_VAR2=123.45`
	`N30 DEF STRING[80] RESULT`
	`N40 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P VAR2:%2.0P", REAL_VAR1,REAL_VAR2)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "PUNCHED TAPE FORMAT:-123.450 VAR2:23".

%S :	Insertion d'une chaîne de caractères.
	Exemple :
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF STRING_VAR:ABCDEFG".

%<m>S :	Insertion d'une chaîne de caractères comportant au moins <m> caractères. Les caractères manquants sont remplis avec des caractères non significatifs.
	Exemple :
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)`
	Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF STRING_VAR:xxxABCDEFG" (dans l'exemple, "x" remplace un caractère non significatif).

%.<n>S :	Insertion de <n> caractères d'une chaîne (à partir du premier caractère). **Exemple :** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR) Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF STRING_VAR:ABC".
%<m>.<n>S :	Insertion de <n> caractères d'une chaîne (à partir du premier caractère). La longueur totale minimale de la chaîne obtenue créée est d'au moins <m> caractères. Les caractères manquants sont remplis avec des caractères non significatifs. **Exemple :** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR) Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "CONTENT OF STRING_VAR:xxxxxABCDE" (dans l'exemple, "x" remplace un caractère non significatif).
%X :	Conversion d'une valeur INTEGER en un chaîne de caractères au format hexadécimal. **Exemple :** N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR) Résultat : La variable de type String RESULT est décrite par la chaîne de caractères "INTEGER HEXADECIMAL:A5B8".

Remarque

La caractéristique du langage CN, en ce qui concerne les descripteurs et les mots-clés, qui consiste à ne pas différencier les minuscules des majuscules, s'applique également aux descriptions de format. Lors de la programmation, il n'existe pas de différence fonctionnelle entre les minuscules et les majuscules.

Possibilités de combinaison

Le tableau ci-après fournit des indications sur les types de données CN pouvant être combinés aux différentes descriptions de format. Les règles relatives à la conversion implicite de types de données s'appliquent ici (voir "Types de données (Page 62)").

	Types de données CN						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-

	Types de données CN						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

Remarque

Le tableau montre que les types de données CN AXIS et FRAME ne peuvent pas être utilisés directement dans la fonction SPRINT. Il est toutefois possible de :

- convertir le type de données AXIS avec la fonction AXSTRING en une chaîne de caractères, qui peut ensuite être traitée avec SPRINT ;
- lire les différentes valeurs du type de données FRAME via un accès aux composants Frame. On obtient ainsi un paramètre de type REAL qui peut être traité avec SPRINT.

2.10 Sauts de programme

2.10.1 Retour au début du programme (GOTOS)

Pour une répétition de programme, l'instruction GOTOS permet de retourner au début d'un programme principal ou d'un sous-programme.

Pour le retour au début du programme, les paramètres machine permettent de régler :

- la mise à "0" du temps d'exécution du programme,
- l'incrémentation du comptage de pièces de la valeur "1".

Syntaxe

```
GOTOS
```

Signification

GOTOS :	Instruction de saut avec destination au début du programme.	
	L'exécution est commandée par le signal d'interface CN/AP : DB21, ... DBX384.0 (commander un saut de programme)	
	Valeur :	Signification :
	0	Sans retour au début du programme. Après GOTOS, l'exécution du programme reprend au bloc suivant du programme pièce.
	1	Retour au début du programme et répétition du programme pièce.

Autres conditions à prendre en compte

- GOTOS déclenche un STOPRE interne (arrêt du prétraitement).
- Dans le cas d'un programme pièce avec des définitions de données (variables LUD), GOTOS exécute un retour au premier bloc de programme qui suit la section de définition, autrement dit l'exécution des définitions de données n'est pas répétée. Les variables définies gardent donc la valeur atteinte dans le bloc GOTOS et ne sont pas remises aux valeurs par défaut programmées dans la section de définition.
- L'instruction GOTOS n'est pas disponible dans les actions synchrones et les cycles technologiques.

Exemple

```
Code de programme          Commentaire
N10 ...                    ; Début du programme.
...
N90 GOTOS                  ; Saut au début du programme.
...
```

2.10.2 Sauts de programme sur repères de saut (IF, GOTOB, GOTOF, GOTO, GOTOC)

Des repères de saut (étiquettes) à rejoindre au sein d'un programme avec les instructions `GOTOF`, `GOTOB`, `GOTO` et `GOTOC` peuvent être définis dans ce programme. L'exécution du programme reprend alors à l'instruction qui suit immédiatement le repère de saut. Il est donc possible de réaliser des sauts à l'intérieur du programme.

Outre les repères de saut, les destinations de saut peuvent être également des numéros de bloc principal ou auxiliaire.

Si une condition de saut (`IF ...`) est formulée avant l'instruction de saut, le saut de programme n'est exécuté que si cette condition est remplie.

Syntaxe

```
GOTOB <destination de saut>
IF <condition de saut> == TRUE GOTOB <destination de saut>

GOTOF <destination de saut>
IF <condition de saut> == TRUE GOTOF <destination de saut>

GOTO <destination de saut>
IF <condition de saut> == TRUE GOTO <destination de saut>

GOTOC <destination de saut>
IF <condition de saut> == TRUE GOTOC <destination de saut>
```

Signification

`GOTOB` :	Instruction de saut avec destination de saut en direction du début du programme.
`GOTOF` :	Instruction de saut avec destination de saut en direction de la fin du programme.
`GOTO` :	Instruction de saut avec recherche de destination de saut. La destination est d'abord recherchée en direction de la fin du programme, puis en direction du début du programme.
`GOTOC` :	Même effet que GOTO, sauf que l'alarme 14080 "Destination de saut pas trouvée" est inhibée. Cela signifie que l'exécution du programme n'est pas annulée en cas de recherche de destination de saut sans résultat, mais qu'elle reprend à la ligne de programme qui suit l'instruction GOTOC.

<destination de saut> :	Paramètre de la destination de saut Les indications possibles sont :	
	<repère de saut> :	La destination de saut correspond au repère de saut défini dans le programme avec un nom défini par l'utilisateur : <repère de saut> :
	<numéro de bloc> :	La destination de saut est un numéro de bloc principal ou auxiliaire (par exemple : 200, N300)
	Variable de type STRING :	Destination de saut variable. La variable correspond à un repère de saut ou à un numéro de bloc.
IF :	Mot clé pour formuler la condition de saut. Pour la condition de saut, vous pouvez utiliser toutes les opérations relationnelles et logiques (résultat : TRUE ou FALSE). Le saut est exécuté si le résultat de l'opération est TRUE.	

Remarque

Repères de saut (étiquettes)

Les repères de saut figurent toujours au début d'un bloc. S'il existe un numéro de programme, le repère de saut figure immédiatement après le numéro de bloc.

Les règles suivantes s'appliquent à la définition des repères de saut :

- Nombre de caractères :
  - Minimum 2
  - Maximum 32
- Caractères autorisés :
  - Caractères alphabétiques
  - Chiffres
  - Traits de soulignement
- Les deux premiers caractères doivent être obligatoirement des lettres ou des traits de soulignement.
- Le nom d'un repère de saut est suivi de deux points (":").

Autres conditions à prendre en compte

- La destination du saut doit être un bloc avec un repère de saut ou un numéro de bloc figurant **dans** le programme.
- Une instruction de saut sans condition de saut doit être programmée dans un bloc spécifique. Cette restriction ne s'applique pas aux instructions de saut avec des conditions de saut. Dans ce cas, plusieurs instructions de saut peuvent être formulées dans un bloc.
- Dans le cas de programmes comportant des instructions de saut sans condition de saut, l'instruction de fin de programme M2/M30 peut figurer avant la fin du programme.

Exemples

Exemple 1 : sauts sur repères de saut

Code de programme	Commentaire
N10 …	
N20 GOTOF Label_1	; Saut en direction de la fin du programme sur le
	; repère de saut "Label_1".
N30 …	
N40 Label_0: R1=R2+R3	; Définition du repère de saut "Label_0".
N50 …	
N60 Label_1:	; Définition du repère de saut "Label_1".
N70 …	
N80 GOTOB Label_0	; Saut en direction du début du programme sur le
	; repère de saut "Label_0".
N90 …	

Exemple 2 : saut indirect sur numéro de bloc

Code de programme	Commentaire
IF <condition> == TRUE	
R10=100	; Attribuer la destination de saut
ELSE	
R10=110	; Attribuer la destination de saut
ENDIF	
; Saut en direction de la fin du programme sur le bloc dont le numéro est indiqué dans R10	
N10 GOTOF "N"<<R10	
...	
N90 ...	
N100 ...	; Destination de saut
N110 ...	
...	

Exemple 3 : saut sur destination de saut variable

Code de programme	Commentaire
DEF STRING[20] DESTINATION	
IF <condition> == TRUE	
DESTINATION = "repère1"	; Attribuer la destination de saut
ELSE	
DESTINATION = "repère2"	; Attribuer la destination de saut
ENDIF	
; Saut en direction de la fin du programme sur la destination de saut variable "Contenu de DESTINATION".	
GOTOF DESTINATION	
Repère1: T="foret1"	; Destination de saut 1
...	

Code de programme	Commentaire
Repère2: T="foret2"	; Destination de saut 2
...	

Exemple 4 : saut avec condition de saut

```
Code de programme                                          Commentaire
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20                    ; Affectation des valeurs initiales
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6                ; Repère de saut LA1
N42 R1=R1+R3 R4=R4-1
; IF condition de saut == TRUE
; THEN saut en direction du début du programme sur le repère de saut LA1
N43 IF R4>0 GOTOB LA1
N44 M30                                                    ; Fin du programme
```

2.10.3 Saut de programme (CASE ... OF ... DEFAULT ...)

La fonction CASE permet de vérifier la valeur actuelle (type : INT) d'une variable ou d'une fonction de calcul et de sauter à différents endroits dans le programme en fonction du résultat obtenu.

Syntaxe

```
CASE(<expression>) OF <constante_1> GOTOF <destination de saut_1>
<constante_2> GOTOF <destination de saut_2> ... DEFAULT GOTOF
<destination de saut_n>
```

Signification

CASE :	Instruction de saut	
<expression> :	Variable ou fonction de calcul	
OF :	Mot clé pour formuler les sauts de programme conditionnels.	
<constante_1> :	Première valeur constante indiquée pour la variable ou la fonction de calcul	
	Type :	INT
<constante_2> :	Deuxième valeur constante indiquée pour la variable ou la fonction de calcul	
	Type :	INT
DEFAULT :	Au cas où la variable ou la fonction de calcul ne prend aucune des valeurs constantes indiquées, la destination du saut peut être définie par l'instruction DEFAULT. **Remarque :** Au cas où l'instruction DEFAULT n'est pas programmée, le bloc qui suit l'instruction CASE devient destination du saut.	
GOTOF :	Instruction de saut avec destination de saut en direction de la fin du programme. Au lieu de GOTOF, il est également possible de programmer toutes les autres instructions GOTO (voir chapitre "Sauts de programme sur repères de saut").	

<destination de saut_1> :	Destination de saut si la valeur de la variable ou de la fonction de calcul correspond à la première constante indiquée. La destination de saut peut être indiquée de la manière suivante :	
	<repère de saut> :	La destination de saut correspond au repère de saut défini dans le programme avec un nom défini par l'utilisateur : <repère de saut> :
	<numéro de bloc> :	La destination de saut est un numéro de bloc principal ou auxiliaire (par exemple : 200, N300)
	Variable de type STRING :	Destination de saut variable. La variable correspond à un repère de saut ou à un numéro de bloc.
<destination de saut_2> :	Destination de saut si la valeur de la variable ou de la fonction de calcul correspond à la deuxième constante indiquée.	
<destination de saut_n> :	Destination de saut si la valeur de la variable ou de la fonction de calcul ne correspond à aucune des valeurs constantes indiquées.	

Exemple

Code de programme

```
...
N20 DEF INT VAR1 VAR2 VAR3
N30 CASE(VAR1+VAR2-VAR3) OF 7 GOTOF Label_1 9 GOTOF La-
bel_2 DEFAULT GOTOF Label_3
N40 Label_1: G0 X1 Y1
N50 Label_2: G0 X2 Y2
N60 Label_3: G0 X3 Y3
...
```

L'instruction CASE de N30 définit les possibilités de saut de programme suivantes :

1. Si la valeur de la fonction de calcul VAR1+VAR2-VAR3 = 7, la destination de saut correspond au bloc avec la définition du repère de saut "Label_1" (→ N40).
2. Si la valeur de la fonction de calcul VAR1+VAR2-VAR3 = 9, la destination de saut correspond au bloc avec la définition du repère de saut "Label_2" (→ N50).
3. Si la valeur de la fonction de calcul VAR1+VAR2-VAR3 n'est ni égale à 7 ni à 9, la destination de saut correspond au bloc avec la définition du repère de saut "Label_3" (→ N60).

2.11 Répétition de sections de programme (REPEAT, REPEATB, ENDLABEL, P)

Cette fonction permet la répétition de sections de programme dans un ordre quelconque.

Les lignes ou sections de programme à répéter sont repérées par des repères de saut (étiquettes).

Remarque

Repères de saut (étiquettes)

Les repères de saut figurent toujours au début d'un bloc. S'il existe un numéro de programme, le repère de saut figure immédiatement après le numéro de bloc.

Les règles suivantes s'appliquent à la définition des repères de saut :

- Nombre de caractères :
  - Minimum 2
  - Maximum 32
- Caractères autorisés :
  - Caractères alphabétiques
  - Chiffres
  - Traits de soulignement
- Les deux premiers caractères doivent être obligatoirement des lettres ou des traits de soulignement.
- Le nom d'un repère de saut est suivi de deux points (":").

Syntaxe

1. répéter des lignes de programme individuelles :

```
<repère de saut> : ...
...
REPEATB <repère de saut> P=<n>
...
```

2. répéter la section de programme entre le repère de saut et l'instruction REPEAT :

```
<repère de saut> : ...
...
REPEAT <repère de saut> P=<n>
...
```

3. répéter la section entre deux repères de saut :

```
<repère de saut de début> : ...
...
<repère de saut de fin> : ...
...
```

```
REPEAT <repère de saut de début> <repère de saut de fin>
P=<n>
...
```

Remarque

Il n'est pas possible de placer l'instruction `REPEAT` entre les deux repères de saut. Si le `<repère de saut de début>` est placé devant l'instruction `REPEAT` et si le `<repère de saut de fin>` n'est pas atteint avant l'instruction `REPEAT`, la répétition a lieu entre le `<repère de saut de début>` et l'instruction `REPEAT`.

4. répéter la section entre le repère de saut et ENDLABEL :

```
<repère de saut> : ...
...
ENDLABEL : ...
...
REPEAT <repère de saut> P=<n>
...
```

Remarque

Il n'est pas possible de placer l'instruction `REPEAT` entre le `<repère de saut>` et `ENDLABEL`. Si le `<repère de saut>` est placé devant l'instruction `REPEAT` et si le `ENDLABEL` n'est pas atteint avant l'instruction `REPEAT`, la répétition a lieu entre le `<repère de saut>` et l'instruction `REPEAT`.

Signification

`REPEATB` :	Instruction de répétition d'une ligne de programme
`REPEAT` :	Instruction de répétition d'une section de programme
`<repère de saut>` :	Le `<repère de saut>` repère : • la ligne de programme à répéter (avec `REPEATB`) ou • le début de la section de programme à répéter (avec `REPEAT`)
	La ligne de programme repérée par le `<repère de saut>` peut se trouver avant ou après l'instruction `REPEAT`/`REPEATB`. La recherche commence en direction du début du programme. Si le repère de saut n'est pas trouvée dans cette direction, la recherche est s'effectue en direction de la fin du programme. **Exception :** si la section de programme doit être répétée entre le repère de saut et l'instruction `REPEAT` (voir le point 2 sous Syntaxe), la ligne de programme repérée par le `<repère de saut>` doit se trouver **devant** l'instruction `REPEAT`, car dans ce cas, la rechercher s'effectue **uniquement** en direction du début du programme. Si la ligne contenant le `<repère de saut>` contient d'autres instructions, celles-ci sont exécutées une nouvelle fois à chaque répétition.

ENDLABEL :	Mot-clé qui repère la fin d'une section de programme à répéter Si la ligne contenant ENDLABEL contient d'autres instructions, celles-ci sont exécutées une nouvelle fois à chaque répétition. ENDLABEL peut être utilisée plusieurs fois dans le programme.
P :	Adresse pour l'indication du nombre de répétitions
<n> :	Nombre de répétitions de la section de programme
	Type : INT
	La section de programme à répéter est répétée <n> fois. Après la dernière répétition, l'exécution du programme reprend à la ligne suivant celle qui contient REPEAT/REPEATB. **Remarque :** si P=<n> n'est pas indiqué, la section de programme a répéter est répétée exactement une fois.

Exemples

Exemple 1 : répéter des lignes de programme individuelles

```
Code de programme                    Commentaire
N10 POSITION1: X10 Y20
N20 POSITION2: CYCLE(0,,9,8)         ; Cycle de positions
N30 ...
N40 REPEATB POSITION1 P=5            ; Exécuter le bloc N10 cinq fois.
N50 REPEATB POSITION2                ; Exécuter le bloc N20 une fois.
N60 ...
N70 M30
```

Exemple 2 : répéter la section de programme entre le repère de saut et l'instruction REPEAT

```
Code de programme                Commentaire
N5 R10=15
N10 Begin: R10=R10+1             ; Côté
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 Z=10+R10
N80 REPEAT BEGIN P=4             ; Exécuter la section N10 à N70 quatre fois.
N90 Z10
N100 M30
```

Exemple 3 : répéter la section entre deux repères de saut

```
Code de programme                    Commentaire
N5 R10=15
N10 Begin: R10=R10+1                 ; Côté
```

```
Code de programme                Commentaire
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 END: Z=10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3        ; Exécuter la section N10 à N70 trois fois.
N110 Z10
N120 M30
```

Exemple 4 : répéter la section entre le repère de saut et ENDLABEL

```
Code de programme            Commentaire
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X10
N40 Y10
N50 BEGIN2:
N60 X20
N70 Y30
N80 ENDLABEL: Z10
N90 X0 Y0 Z0
N100 Z-10
N110 BEGIN3: X20
N120 Y30
N130 REPEAT BEGIN3 P=3       ; Exécuter la section N110 à N120 trois fois.
N140 REPEAT BEGIN2 P=2       ; Exécuter la section N50 à N80 deux fois.
N150 M100
N160 REPEAT BEGIN1 P=2       ; Exécuter la section N20 à N80 deux fois.
N170 Z10
N180 X0 Y0
N190 M30
```

Exemple 5 : réaliser un fraisage, des trous taraudés avec différentes technologies

```
Code de programme          Commentaire
N10 FORET_CENTRER()        ; Mettre en place le foret à centrer.
N20 POS_1:                 ; Réseau de positions 1
N30 X1 Y1
N40 X2
N50 Y2
N60 X3 Y3
N70 ENDLABEL :
```

```
Code de programme          Commentaire
N80 POS_2 :                ; Réseau de positions 2
N90 X10 Y5
N100 X9 Y-5
N110 X3 Y3
N120 ENDLABEL :
N130 FORET()               ; Mettre en place le foret et le cycle de perçage.
N140 TARAUD(6)             ; Mettre en place le taraud M6 et le cycle de taraudage.
N150 REPEAT POS_1          ; Répéter la section de programme une fois à partir de
                           POS_1 jusqu'à ENDLABEL.
N160 FORET()               ; Mettre en place le foret et le cycle de perçage.
N170 TARAUD(8)             ; Mettre en place le taraud M8 et le cycle de taraudage.
N180 REPEAT POS_2          ; Répéter la section de programme une fois à partir de
                           POS_2 jusqu'à ENDLABEL.
N190 M30
```

Informations complémentaires

- Des imbrications sont autorisées dans une section de programme à répéter. Chaque appel correspond à un niveau de sous-programme.
- Si `M17` ou `RET` sont programmés pendant l'exécution de la répétition d'une section de programme, celle-ci est annulée. L'exécution du programme reprend au bloc suivant la ligne `REPEAT`.
- Dans l'affichage du programme en cours, la répétition de la section de programme est visualisée en tant que niveau spécifique de sous-programme.
- Si un abandon est déclenché pendant l'exécution de la section de programme, l'exécution du programme reprend après le bloc comportant l'instruction REPEAT.
  Exemple :

```
Code de programme              Commentaire
N5 R10=15
N10 BEGIN: R10=R10+1           ; Côté
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10                      ; Abandon
N50 X=-R10
N60 Y=-R10
N70 END: Z10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3
N120 Z10                       ; Poursuivre l'exécution du programme.
N130 M30
```

- Structures de contrôle et répétition de sections de programme peuvent être combinées. Il y a cependant lieu d'éviter les chevauchements. Il est recommandé que la répétition d'une section de programme se trouve à l'intérieur d'une branche de structure de contrôle ou d'une structure de contrôle se trouvant à l'intérieur d'une répétition de section de programme.
- En cas de programmation mixte de sauts et d'une répétition de section de programme, les blocs sont exécutés de façon purement séquentielle. En cas de saut à partir d'une boucle de répétition de section de programme par exemple, l'exécution se poursuit jusqu'à ce que la fin de la section de programme à répéter soit trouvée.
  Exemple :

Code de programme

```
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X=10
N40 Y=10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL: Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30
```

Remarque

Il est recommandé que l'instruction `REPEAT` se trouve après les blocs de déplacement.

2.12 Structures de contrôle

En version standard, la commande traite les blocs CN dans l'ordre de leur programmation.

Il est possible de varier cet ordre en programmant des blocs de programme et des boucles de programme alternatifs. La programmation de ces structures de contrôle est effectuée avec les mots-clés `IF`, `ELSE`, `ENDIF`, `LOOP`, `FOR`, `WHILE` et `REPEAT`.

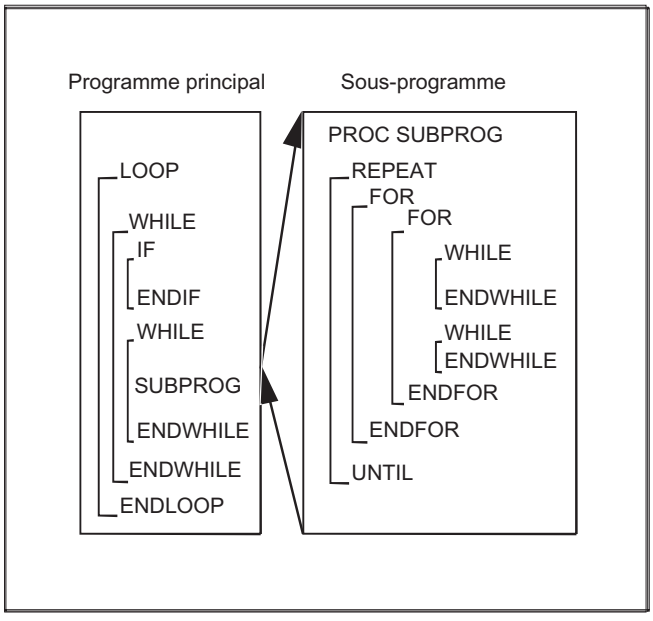
IMPORTANT
Erreur de programmation Les structures de contrôle ne sont possibles qu'à l'intérieur de la partie instructions d'un programme. Les définitions dans l'en-tête de programme ne peuvent pas être exécutées de façon conditionnelle ou répétitive. Les mots-clés pour les structures de contrôle, tout comme les destinations de sauts, ne doivent pas être utilisés dans des macros. Ceci ne fait pas l'objet d'une vérification lors de la définition des macros.

Prise d'effet

Une structure de contrôle ne peut pas être utilisée de façon globale par tous les programmes.

Niveaux d'imbrication

Une imbrication de 16 structures de contrôle au maximum est possible à l'intérieur de chaque niveau de sous-programme.



Durée d'exécution des programmes

En mode interpréteur standard, l'utilisation de sauts permet d'obtenir une exécution du programme plus rapide qu'avec les structures de contrôle.

Dans des cycles précompilés, il n'y a pas de différence entre les sauts dans le programme et les structures de contrôle.

Affichage de blocs actuel pour les boucles de programme

Si dans une boucle de programme seuls des blocs de prétraitement sont exécutés, l'affichage de blocs actuel représente le dernier bloc principal **avant** la boucle de programme.

Afin que, par exemple à des fins de diagnostic, les blocs de prétraitement exécutés deviennent également visibles dans l'affichage de blocs actuel, le décodage bloc par bloc SBL2 doit être activé.

Bibliographie

Description fonctionnelle Fonctions de base, chapitre : GMFC, Canal, Mode de programme, Comportement au Reset (K1) > Bloc par bloc > Décodage bloc par bloc SBL2 avec arrêt du prétraitement des blocs implicite

Rectification sans bloc principal

Si dans une boucle de programme aucun bloc principal n'est programmé, la boucle est exécutée en prétraitement tant que la condition de la boucle n'est pas satisfaite.

Ceci engendre une charge de traitement élevée pouvant ralentir l'affichage.

Un remède possible est l'insertion de l'instruction STOPRE ou d'un arrêt temporisé G04 de 0 seconde **dans** la boucle.

Autres conditions à prendre en compte

- Les blocs comportant des éléments de structures de contrôle ne peuvent pas être déclarés optionnels.
- Les repères de saut (étiquettes) ne sont pas autorisés dans les blocs contenant des éléments de structures de contrôle.
- Les structures de contrôle sont exécutées de façon interprétative. En cas d'identification d'une fin de boucle, le début de la boucle est recherché compte tenu des structures de contrôle ainsi trouvées. Par conséquent, en mode interpréteur, la structure d'un programme n'est pas complètement testée.
- En principe, il est recommandé de ne pas utiliser en commun des structures de contrôle et des sauts dans le programme.
- Si des cycles subissent un prétraitement, l'imbrication correcte des structures de contrôle est vérifiée.

2.12.1 Instruction conditionnelle et branchement (IF, ELSE, ENDIF)

Instruction conditionnelle : IF - Bloc de programme - ENDIF

En présence d'une instruction conditionnelle, le bloc de programme entre `IF` et `ENDIF` n'est exécuté que lorsque la condition est remplie.

Branchement : IF - Bloc de programme_1 - ELSE - Bloc de programme_2 - ENDIF

En présence d'un branchement, l'un de deux blocs de programme est toujours exécuté.

Si la condition est remplie, le bloc de programme_1 entre `IF` et `ELSE` est exécuté.

Si la condition n'est **pas** remplie, le bloc de programme_2 entre `ELSE` et `ENDIF` est exécuté.

Syntaxe

Instruction conditionnelle

```
IF <condition>
   Bloc de programme                    ; Exécution lorsque : <condition> == TRUE
ENDIF
```

Branchement

```
IF <condition>
   Bloc de programme_1                  ; Exécution lorsque : <condition> == TRUE
ELSE
   Bloc de programme_2                  ; Exécution lorsque : <condition> == FALSE
ENDIF
```

Signification

`IF` :	Début de l'instruction conditionnelle ou du branchement.
`ELSE` :	Début du bloc de programme alternatif.
`ENDIF` :	Fin de l'instruction conditionnelle ou du branchement.
`<condition >` :	Expression logique, dont l'évaluation a pour résultat TRUE ou FALSE.

Exemple : sous-programme de changement d'outil

Code de programme	**Commentaire**
`PROC L6`	`Routine de changement d'outil`
`N500 DEF INT TNR_COURANT`	`Variable du numéro T actif`
`N510 DEF INT TNR_PRESELECTION`	`Variable du numéro T présélec-tionné`
	`Détermination de l'outil courant`
`N520 STOPRE`	
`N530 IF $P_ISTEST`	`En mode de test de programme ...`

Code de programme	Commentaire
N540 TNR_COURANT = $P_TOOLNO	... lecture de l'outil "courant" du contexte du programme.
N550 ELSE	Sinon ...
N560 TNR_COURANT = $TC_MPP6[9998,1]	... lecture de l'outil de la broche.
N570 ENDIF	
N580 GETSELT(TNR_PRESELECTION)	Lecture du numéro T de l'outil présélectionné sur la broche.
N590 IF TNR_COURANT <> TNR_PRESELECTION	Si l'outil présélectionné n'est pas encore l'outil courant, ...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... accostage de la position de changement d'outil ...
N610 M206	... et exécution du changement d'outil.
N620 ENDIF	
N630 M17	

2.12.2 Boucle de programme sans fin (LOOP, ENDLOOP)

La boucle infinie trouve son application dans des programmes infinis. Un retour au début de boucle a toujours lieu en fin de boucle.

Syntaxe

```
LOOP
...
ENDLOOP
```

Signification

LOOP :	Début de la boucle sans fin.
ENDLOOP :	Fin de la boucle provoquant un retour au début de la boucle.

Exemple

Code de programme
...
LOOP
MSG ("Pas de changement d'outil actif")
M0
STOPRE
ENDLOOP
...

2.12.3 Boucle de comptage (FOR ... TO ..., ENDFOR)

La boucle de comptage est utilisée lorsqu'une section de programme doit être répétée un certain nombre de fois.

Syntaxe

```
FOR <variable> = <valeur initiale> TO <valeur finale>
...
ENDFOR
```

Signification

FOR :	Début de la boucle de comptage.	
ENDFOR :	Fin de la boucle provoquant un retour au début de la boucle tant que la valeur finale du comptage n'est pas atteinte.	
<variable> :	Variable de comptage incrémentée de la valeur "1" à chaque exécution, de la valeur initiale à la valeur finale.	
	type	INT ou REAL **Remarque :** Le type REAL est pris en compte par exemple lorsque le paramètre R est programmé pour une boucle de comptage. Si la variable de comptage est de type REAL, sa valeur est arrondie à un nombre entier.
<valeur initiale> :	Valeur initiale du comptage Condition : La valeur initiale doit être inférieure à la valeur finale.	
<valeur finale> :	Valeur finale du comptage	

Exemples

Exemple 1 : variable INT ou paramètre R en tant que variable de comptage

Variable INT en tant que variable de comptage :

```
Code de programme                    Commentaire
DEF INT iVARIABLE1
R10=R12-R20*R1 R11=6
FOR iVARIABLE1 = R10 TO R11          ; Variable de comptage = variable INT
 R20=R21*R22+R33
ENDFOR
M30
```

Paramètre R en tant que variable de comptage :

```
Code de programme                    Commentaire
R11=6
```

Code de programme	Commentaire
FOR R10=R12-R20*R1 TO R11 R20=R21*R22+R33 ENDFOR M30	; Variable comptage = paramètre R (variable REAL)

Exemple 2 : fabrication d'un nombre de pièces défini

Code de programme	Commentaire
DEF INT SERIE	; Définition de la variable INT avec le nom "SERIE".
FOR SERIE = 0 TO 100	; Début de la boucle de comptage. La variable "SERIE" est incrémentée de la valeur initiale "0" à la valeur finale "100".
G01 …	
ENDFOR	; Fin de la boucle de comptage.
M30	

2.12.4 Boucle de programme avec condition en début de boucle (WHILE, ENDWHILE)

Dans une boucle WHILE, la condition figure au début de la boucle. La boucle WHILE est exécutée tant que la condition est remplie.

Syntaxe

```
WHILE <condition>
...
ENDWHILE
```

Signification

WHILE :	Début de la boucle de programme.
ENDWHILE :	Fin de la boucle provoquant un retour au début de la boucle.
<condition > :	Condition à remplir pour l'exécution de la boucle WHILE.

Exemple

Code de programme	Commentaire
...	
WHILE $AA_IW[AXE DE PERÇAGE] > -10	; Appel de la boucle WHILE sous la condition suivante : la consigne SCP courante de l'axe de perçage doit être supérieure à -10.
G1 G91 F250 AX[AXE DE PERÇAGE] = -1	

Code de programme	Commentaire
ENDWHILE	
...	

2.12.5 Boucle de programme avec condition en fin de boucle (REPEAT, UNTIL)

Dans une boucle REPEAT, la condition figure à la fin de la boucle. La boucle REPEAT est exécutée une fois, puis répétée jusqu'à ce que la condition soit remplie.

Syntaxe

```
REPEAT
...
UNTIL <condition>
```

Signification

REPEAT :	Début de la boucle de programme.
UNTIL :	Fin de la boucle provoquant un retour au début de la boucle.
<condition > :	Condition à remplir pour que la boucle REPEAT ne soit plus exécutée.

Exemple

Code de programme	Commentaire
...	
REPEAT	; Appel de la boucle REPEAT.
...	
UNTIL ...	; Vérifie si la condition est remplie.
...	

2.12.6 Exemple de programme avec des structures de contrôle imbriquées

Code de programme	Commentaire
LOOP	
IF NOT $P_SEARCH	; IF pas de recherche de bloc
G1 G90 X0 Z10 F1000	
WHILE $AA_IM[X] <= 100	; WHILE (Consigne axe X <= 100)
G1 G91 X10 F500	; Réseau de trous
Z-5 F100	
Z5	
ENDWHILE	

Code de programme	**Commentaire**
ELSE	; ELSE recherche de bloc
MSG("Pas de perçage pendant la recherche de bloc")	
ENDIF	; ENDIF
$A_OUT[1] = 1	; plaque à percer suivante
G4 F2	
ENDLOOP	
M30	

2.13 Ordres de coordination (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

Un canal de la CN peut en principe exécuter son propre programme, indépendamment des autres canaux de son groupe de mode de fonctionnement (GMFC). Si plusieurs programmes dans plusieurs canaux du GMFC sont au même moment partie prenante dans la réalisation d'une pièce, alors le déroulement des programmes dans les différents canaux doit être coordonné à l'aide des ordres de coordination suivants.

Condition

Tous les canaux concernés par la coordination de programmes doivent appartenir au **même groupe à mode de fonctionnement commun (GMFC)**.

MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<Kanal>] = <BAG-Nummer>

Nom de canal au lieu de numéro de canal

Il est également possible d'utiliser, au lieu des numéros de canal, les noms de paramètre renseignés dans PM20000 $MC_CHAN_NAME[<indice de canal>] en tant que paramètres des procédures prédéfinies de coordination de programmes. À cet effet, l'utilisation des noms de canal doit d'abord être débloquée dans les programmes CN.

MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE

Syntaxe

```
INIT(<ChanNr>, <Prog>, <AckMode>)
START(<ChanNr>, <ChanNr>, ...)
WAITM(<MarkNr>, <ChanNr>, <ChanNr>, ...)
WAITE(<ChanNr>, <ChanNr>, ...)
WAITMC(<MarkNr>, <ChanNr>, <ChanNr>, ...)
SETM(<MarkNr>, <MarkNr>, ...)
CLEARM(<MarkNr>, <MarkNr>, ...)
```

Signification

`INIT()` :	Procédure prédéfinie de sélection du programme CN à exécuter dans le canal spécifié
`START()` :	Procédure prédéfinie pour le démarrage du programme sélectionné dans chaque canal
`WAITM()` :	Procédure prédéfinie pour attendre l'atteinte d'un repère d'attente dans les canaux spécifiés Dans le canal propre à l'instruction, le repère d'attente spécifié est défini par `WAITM`. Le bloc précédent est terminé par un arrêt précis. Le repère d'attente est effacé après la synchronisation. Jusqu'à 10 repères peuvent être définis dans un même canal.
`WAITE()` :	Procédure prédéfinie pour l'attente de la fin du programme dans un ou plusieurs canaux différents
`WAITMC()` :[1)]	Procédure prédéfinie pour attendre l'atteinte d'un repère d'attente dans les canaux spécifiés A la différence de `WAITM`, le freinage de l'axe pour l'arrêt précis n'est déclenché que si les autres canaux n'ont pas encore atteint le repère d'attente.

SETM() :[1]	Procédure prédéfinie pour la définition d'un ou de plusieurs repères d'attente pour la coordination de canaux L'exécution dans le canal propre n'en est pas affectée. SETM conserve sa validité au-delà de Reset canal et de Départ programme.		
CLEARM() :[1]	Procédure prédéfinie pour la suppression d'un ou de plusieurs repères d'attente de la coordination de canaux L'exécution dans le canal propre n'en est pas affectée. CLEARM() supprime tous les repères d'attente du canal. CLEARM(0) ne supprime que le repère d'attente "0". CLEARM conserve sa validité au-delà de Reset canal et de Départ programme.		
<ChanNr> :	Numéro de canal Il n'est pas nécessaire de spécifier le numéro du canal propre.		
	Type :	INT	
<Prog> :	Chemin d'accès absolu ou relatif (**facultatif**) + nom de programme		
	Type :	STRING	
	Concernant le chemin d'accès, voir : **Bibliographie** Manuel de programmation Préparation du travail, chapitre "Gestion des fichiers et des programmes" > "Mémoire de programme" > "Adressage des fichiers de la mémoire de programmes" (Page 227)		
<AckMode> :	Mode d'acquittement (**facultatif**)		
	Type :	CHAR	
	Valeurs :	"N"	Sans acquittement L'exécution du programme se poursuit après l'envoi de l'ordre. Il n'y a pas de signalisation en retour de l'exécution de l'ordre.
		"S"	Acquittement synchrone L'exécution du programme est suspendue tant que l'élément récepteur n'a pas acquitté l'ordre. Lors d'un acquittement positif, l'instruction suivante est exécutée. En cas d'acquittement négatif, un message d'erreur s'affiche.
<MarkNr> :	Numéro de la marque d'attente **Remarque** Dans un système multicanal, 100 repères d'attente sont disponibles au maximum (repère d'attente 0 ... 99). Dans un système monocanal, seul le repère d'attente 0 est disponible.		

1) Pour la communication personnalisée et/ou la coordination de canaux, les repères d'attente peuvent également être utilisés via SETM / CLEARM sans l'instruction d'attente WAITMC. Dans ce cas, les repères d'attente conservent leurs valeurs au-delà de Reset canal et de Départ programme.

Exemples

START à l'aide du nom de canal dans PM20000

- Paramétrage
```
MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "USINAGE" ; nom du canal 1
$MC_CHAN_NAME[ 1 ] = "ALIMENTATION" ; nom du canal 2
```
- Programmation

Code de programme	Commentaire
START(USINAGE)	; Départ de canal 1
START(ALIMENTATION)	; Départ de canal 2

START par "noms de canaux" locaux et variable utilisateur

Code de programme	Commentaire
DEF INT MACHINE = 1	; Définition variable utilisateur pour canal 1
DEF INT CHARGEUR = 2	; Définition variable utilisateur pour canal 2
...	
START(MACHINE)	; Départ de canal 1
START(CHARGEUR)	; Départ de canal 2

START par "noms de canaux" locaux, variable utilisateur et noms de canaux paramétrés

Code de programme	Commentaire
DEF INT chanNo1	; Définition variable utilisateur pour canal 1
DEF INT chanNo2	; Définition variable utilisateur pour canal 2
chanNo1 = CHAN_1	; Affectation de noms de canaux paramétrés canal 1
chanNo2 = CHAN_2	; Affectation de noms de canaux paramétrés canal 2
...	
START(chanNo1)	; Départ de canal 1
START(chanNo2)	; Départ de canal 2

instruction INIT avec chemin d'accès absolu

Sélection de programme /_N_MPF_DIR/_N_CHARIOTAGE1_MPF dans canal 2.

Code de programme
```
INIT(2,"/_N_WKS_DIR/_N_ARBRE1_WPD/_N_CHARIOTAGE1_MPF")
```

Instruction INIT avec noms de programmes

Sélection du programme avec le nom "MYPROG". La commande recherche le programme à partir du chemin de recherche.

Code de programme
```
INIT(2,"MYPROG")
```

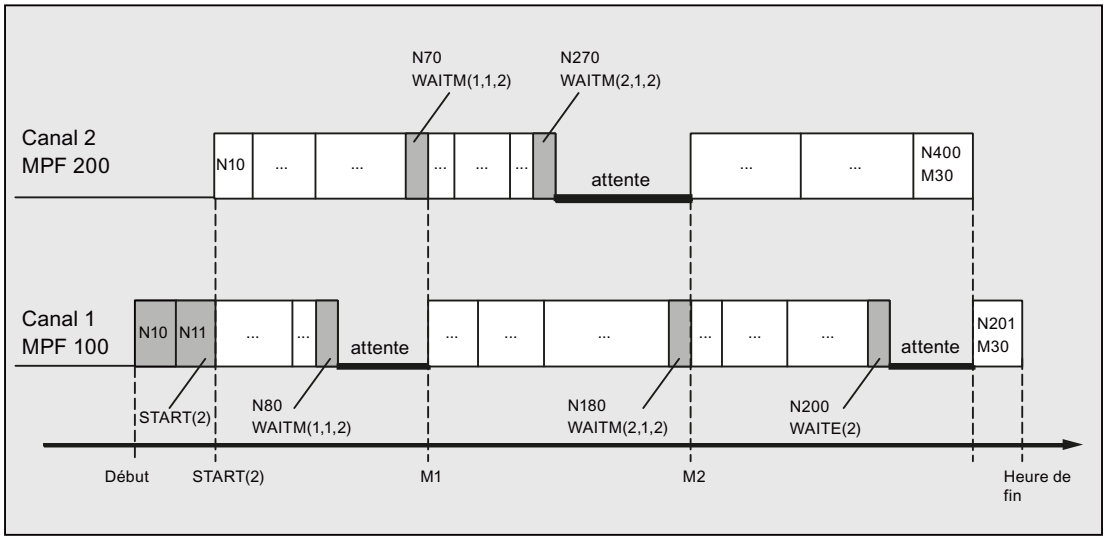
coordination de programmes avec WAITM

- Canal 1 : Le programme /_N_MPF_DIR/_N_MPF100_MPF est déjà sélectionné et démarré.

Code de programme	**Commentaire**
	; Programme MPF100
N10 INIT(2,"MPF200","N")	; Sélection programme MPF200, canal 2
N11 START (2)	; Départ de canal 2
...	
N80 WAITM(1,1,2)	; Attente du repère WAIT 1 dans le canal 1 et 2
N81 ...	; Canal 1, N81 et canal 2, N71 sont ; démarrés de manière synchrone
...	
N180 WAITM(2,1,2)	; Attente du repère WAIT 2 dans le canal 1 et 2
N181 ...	; Canal 1, N181 et canal 2, N271 sont ; démarrés de manière synchrone
...	
N200 WAITE(2)	; Attente de la fin de programme dans le canal 2
N201 ...	; N201 n'est démarré qu'après la fin de programme de ; MPF200 dans le canal 2
N201 M30	; Fin de programme dans le canal 1

- Canal 2 : Dans le canal 1, le programme MPF200_MPF est sélectionné et démarré au moyen des blocs N10 et N20 pour le canal 2.

Code de programme	**Commentaire**
;$PATH=/_N_MPF_DIR	; Programme MPF200
...	
N70 WAITM(1,1,2)	Attente du repère WAIT 1 dans le canal 1 et 2
N71 ...	; Canal 1, N81 et canal 2, N71 sont ; démarrés de manière synchrone
...	
N270 WAITM(2,1,2)	Attente du repère WAIT 2 dans le canal 1 et 2
N271 ...	; Canal 1, N181 et canal 2, N271 sont ; démarrés de manière synchrone
...	
N400 M30	Fin de programme dans le canal 2



Autres conditions à prendre en compte

Démarrage non synchrone de l'exécution de blocs asservis selon des repères WAIT

Lors d'une coordination de canaux au moyen de repères WAIT, un démarrage non synchrone de l'exécution des blocs asservis peut se produire. Ce comportement survient lorsqu'une action est déclenchée immédiatement avant l'atteinte du repère d'attente commun dans un des canaux en synchronisation, ayant pour conséquence l'effacement de la distance restant à parcourir avec repositionnement implicite (REPOSA).

Hypothèse : Affectation d'axe actuelle dans les canaux 1 et 2

- Canal 1 : Axes X1 et U
- Canal 2 : Axe X2

Tableau 2-2 Déroulement chronologique dans les canaux 1 et 2

Canal 1	Canal 2	Description
...	...	Exécution quelconque dans le canal 1 et 2
N100 WAITM(20,1,2)		Canal 1 : atteint le repère WAIT et attend la synchronisation avec le canal 2
Début de l'exécution GETD(U) : • *Permutation d'axe* • *Effacement de la distance restant à parcourir* • *REPOSA* *Fin*	N200 GETD(U)	Canal 2 : exige l'axe U à partir du canal 1 Canal 1 : Exécution de GET(U) à l'arrière plan
	N210 WAITM(20,1,2)	Canal 2 : Atteint le repère WAIT. ⇒ La synchronisation des canaux 1 et 2 est terminée
	N220 G0 X2=100	Canal 2 : Début de l'exécution de N220
N110 G0 X1=100		Canal 1 : Début différé de l'exécution de N110

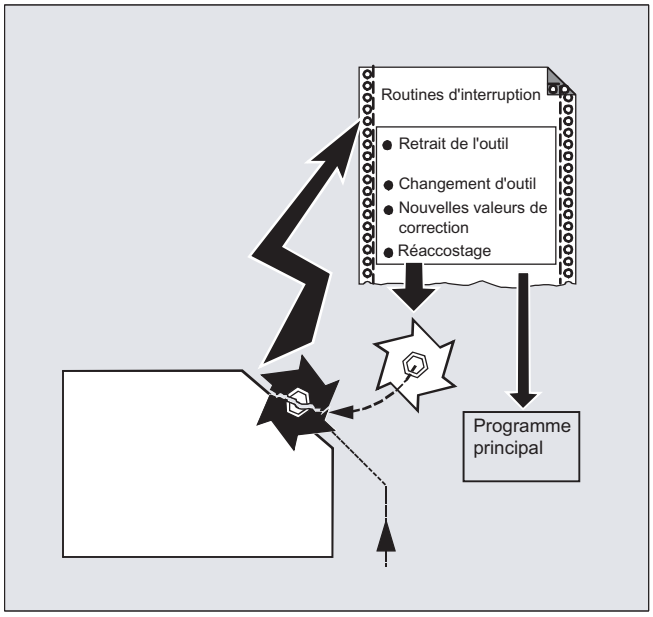
2.14 Routine d'interruption (ASUP)

2.14.1 Fonction d'une routine d'interruption

Remarque

Les termes "sous-programme asynchrone (ASUP)" et "routine d'interruption" utilisés en alternance dans la description suivante désignent la même fonctionnalité.

La fonction d'une routine d'interruption est explicitée à l'aide d'un exemple typique :



L'outil se brise pendant l'usinage. Ceci déclenche un signal qui arrête la phase d'usinage en cours et lance simultanément un sous-programme appelé "routine d'interruption". Ce sous-programme contient toutes les instructions qui doivent être exécutées dans ce cas.

Lorsque l'exécution du sous-programme est terminée (et la machine à nouveau prête), la commande retourne dans le programme principal et poursuit l'usinage, selon l'instruction `REPOS`, à l'endroit où il avait été interrompu (voir "Réaccostage du contour (Page 561)").

⚠ PRUDENCE

Risque de collision

Si vous ne programmez pas d'instruction `REPOS` dans le sous-programme, il y a positionnement au point final du bloc qui suit le bloc interrompu.

Bibliographie

Description fonctionnelle Fonctions de base ; GMFC, Canal, Mode de programme, Comportement au reset (K1), chapitre : "Sous-programmes asynchrones (ASUP), routines d'interruption"

2.14.2 Création d'une routine d'interruption

Créer une routine d'interruption en tant que sous-programme

La routine d'interruption est définie comme un sous-programme.

Exemple :

Code de programme	Commentaire
`PROC RETR_Z`	`; Nom de programme "RELEV_Z"`
`N10 ...`	`; Puis suivent les blocs CN.`
`...`	
`N50 M17`	`; Pour terminer, fin du programme et retour au programme principal.`

Sauvegarder les instructions M modales (SAVE)

Lors de la définition, la routine d'interruption peut être accompagnée de l'attribut `SAVE`.

L'attribut `SAVE` entraîne la sauvegarde des instructions G modales qui sont actives avant l'appel de la routine d'interruption, et leur réactivation à la fin de la routine d'interruption (voir " Sous-programmes avec mécanisme de sauvegarde (SAVE) (Page 176) ").

Il est ainsi possible de reprendre l'usinage au point d'interruption après l'exécution de la routine d'interruption.

Exemple :

Code de programme
`PROC RETR_Z SAVE`
`N10 ...`
`...`
`N50 M17`

Affectation d'autres routines d'interruption (SETINT)

A l'intérieur de la routine d'interruption, il est possible de programmer des instructions `SETINT` (voir "Affectation et démarrage d'une routine d'interruption (SETINT) (Page 136)") pour activer d'autres routines d'interruption. Le démarrage de ces dernières n'a lieu que lorsque l'entrée correspondante commute.

Bibliographie

Pour plus d'informations sur la création de sous-programmes, consultez le chapitre "Sous-programmes, macro-instructions".

2.14.3 Affectation et démarrage d'une routine d'interruption (SETINT, PRIO, BLSYNC)

La commande dispose de plusieurs entrées rapides (entrée 1 à 8), qui déclenchent chacune une interruption (1 à 8). Une priorité et une routine d'interruption peuvent être affectées à chaque interruption par l'instruction `SETINT`. Si l'interruption est déclenchée par l'activation de l'entrée rapide, l'exécution actuelle dans le canal est interrompue et la routine d'interruption est lancée.

Priorité d'interruption

Si des interruptions sont affectées à plusieurs entrées dans un programme pièce, différentes priorités doivent être affectées aux interruptions.

Une valeur de priorité comprise entre 1 et 128 peut être affectée à une interruption. La valeur de priorité 1 correspond à la priorité la plus élevée, la valeur 128 à la priorité la plus faible.

Syntaxe

```
SETINT(<n>) <NOM>
SETINT(<n>) PRIO=<valeur> <NOM>
SETINT(<n>) PRIO=<valeur> <NOM> BLSYNC
SETINT(<n>) PRIO=<valeur> <NOM> LIFTFAST
```

Signification

`SETINT(<n>)` :	Le programme CN (ASUP) <nom> est affecté au signal d'interruption <n>. La routine d'interruption affectée est lancée dès que le signal d'interruption <n> == 1 est détecté. **Remarque :** Si une autre routine d'interruption est affectée à un signal d'interruption <n>, l'affectation précédente est désactivée.	
`<n>` :	Numéro du signal d'interruption	
	Type :	INT
	Plage de valeurs :	1 ... 32
`PRIO=` :	Priorité de l'interruption (facultatif)	
`<valeur>` :	(**facultatif**) Valeur de priorité	
	Type :	INT
	Plage de valeurs :	1 ... 128 (1 ⇒ priorité la plus élevée)
`<NOM>` :	Nom du programme CN (ASUP)	

BLSYNC :	(**facultatif**) BLSYNC entraîne, dans un premier temps après le déclenchement de l'interruption, l'attente de la fin de l'exécution du bloc actuel. La routine d'interruption n'est exécutée qu'après.
LIFTFAST :	(**facultatif**) LIFTFAST entraîne, dans un premier temps après le déclenchement de l'interruption, un relèvement rapide (voir chapitre "Relèvement rapide du contour (SETINT LIFTFAST, ALF) (Page 140)"). La routine d'interruption n'est exécutée qu'après.

Autres conditions à prendre en compte

Règles d'interruption

1. Pour chaque interruption ne pouvant pas être exécutée immédiatement, ou déjà en cours d'exécution, une demande d'interruption supplémentaire est enregistrée. Au-delà de cela, toute demande d'interruption supplémentaire pour cette interruption est perdue.
2. Si une interruption est en cours d'exécution quand une autre interruption de priorité supérieure est déclenchée, cette dernière interrompt l'interruption de priorité inférieure. Une fois l'interruption de priorité supérieure terminée, l'interruption de priorité inférieure se poursuit. Si, pendant l'exécution de l'interruption de priorité supérieure, d'autres demandes arrivent pour l'interruption de priorité inférieure, une demande est enregistrée. Toute demande supplémentaire est perdue.
3. Si une interruption est en cours d'exécution quand une autre interruption de priorité supérieure est déclenchée, cette dernière interrompt l'interruption de priorité inférieure. L'interruption de priorité supérieure est exécutée. Si une interruption de priorité supérieure est une nouvelle fois déclenchée, l'interruption actuelle est interrompue et l'interruption de priorité supérieure est exécutée. Jusqu'à six niveaux d'interruption actifs sont possibles : un niveau d'interruption en cours d'exécution et cinq niveaux d'interruption en attente. Pour chaque niveau d'interruption actif, une seule demande d'interruption supplémentaire est enregistrée. Toutes les autres demandes d'interruption sont perdues. De même, toute demande d'interruption est perdue lorsqu'elle concerne des niveaux d'interruption supplémentaires (niveau d'interruption ≥ 7).

Exemples

Exemple 1 : affectation des routines d'interruption et définition de priorité

```
Code de programme                    Commentaire
N20 SETINT(3) PRIO=1 RELEV_Z         ; IF entrée 3 == 1
                                     ; THEN lancer la routine d'interruption "RELEV_Z"
N30 SETINT(2) PRIO=2 RELEV_X         ; IF entrée 2 == 1
                                     ; THEN lancer la routine d'interruption "RELEV_X"
```

Si les entrées commutent simultanément, les routines d'interruption sont exécutées successivement, dans l'ordre des valeurs de priorité : d'abord "RELEV_Z", puis "RELEV_X".

Exemple 2 : affectation d'une nouvelle routine d'interruption

Code de programme	Commentaire
`N20 SETINT(3) PRIO=2 RELEV_Z`	`; IF entrée 3 == 1`
	`; THEN lancer la routine d'interruption "RELEV_Z"`
`...`	
`N80 SETINT(3) PRIO=1 RELEV_X`	`; IF entrée 3 == 1`
	`; THEN lancer la routine d'interruption "RELEV_X"`

2.14.4 Désactivation/activation de l'affectation d'une routine d'interruption (DISABLE, ENABLE)

Une instruction `SETINT` se désactive avec `DISABLE` et se réactive avec `ENABLE` sans perdre l'affectation entrée → routine d'interruption.

Syntaxe

```
DISABLE(<n>)
ENABLE(<n>)
```

Signification

`DISABLE(<n>)` :	Instruction : **Désactivation** de l'affectation de la routine d'interruption à l'entrée <n>	
`ENABLE(<n>)` :	Instruction : **Réactivation** de l'affectation de la routine d'interruption à l'entrée <n>	
`<n>` :	Paramètre : Numéro du signal d'interruption	
	Type :	INT
	Plage de valeurs :	1 ... 32

Exemple

Code de programme	Commentaire
`N20 SETINT(3) PRIO=1 RELEV_Z`	`; Sur commutation de l'entrée 3, la`
	`; routine d'interruption "RELEV_Z" doit démarrer.`
`...`	
`N90 DISABLE(3)`	`; Désactivation de l'instruction SETINT de N20.`
`...`	
`N130 ENABLE(3)`	`; Réactivation de l'instruction SETINT de N20.`
`...`	

2.14.5 Suppression d'une affectation de routine d'interruption (CLRINT)

L'affectation d'un signal d'interruption à un programme CN (ASUP) définie avec `SETINT` peut être supprimée avec `CLRINT`.

Syntaxe

```
CLRINT(<n>)
```

Signification

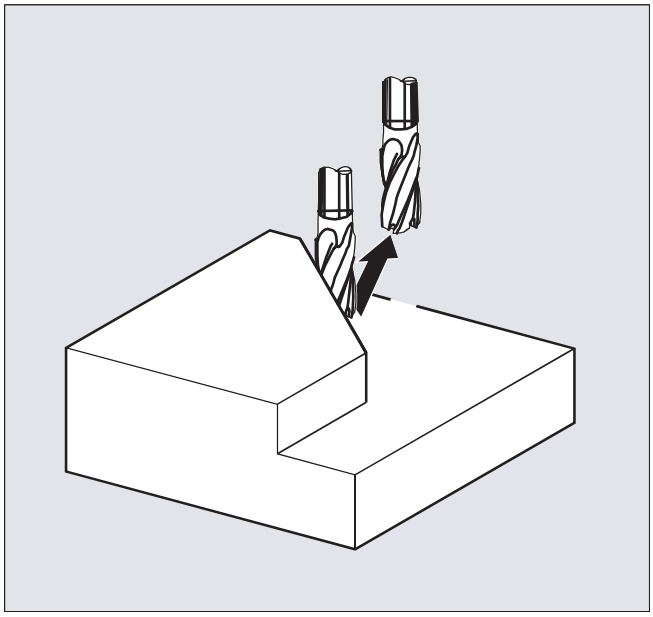
`CLRINT(<n>)` :	Instruction : Suppression de l'affectation du signal d'interruption <n> au programme CN (ASUP) <n> défini avec SETINT	
<n> :	Paramètre : Numéro du signal d'interruption	
	Type :	INT
	Plage de valeurs :	1 ... 32

Exemple

Code de programme	**Commentaire**
`N20 SETINT(3) PRIO=2 RELEV_Z`	
`...`	
`N50 CLRINT(3)`	`; L'affectation de la routine d'interruption "RELEV_Z" à l'entrée "3" est supprimée.`

2.14.6 Relèvement rapide du contour (SETINT LIFTFAST, ALF)

Dans le cas d'une instruction `SETINT` avec `LIFTFAST`, un relèvement rapide éloigne l'outil du contour de pièce lorsque l'entrée commute.



La suite dépend de l'instruction `SETINT`, c'est-à-dire si celle-ci contient ou non une routine d'interruption en plus de `LIFTFAST` :

Avec une routine d'interrup-tion :La routine d'interruption est exécutée **après** le relèvement rapide.

Sans routine d'interruption :Après le relèvement rapide, l'usinage s'arrête avec une alarme.

Syntaxe

```
SETINT(<n>) PRIO=1 LIFTFAST
SETINT(<n>) PRIO=1 <NOM> LIFTFAST
```

Signification

`SETINT(<n>)` :	Instruction : affectation de l'entrée <n> à une routine d'interruption. La routine d'interruption affectée démarre lorsque l'entrée <n> commute.	
`<n>` :	Paramètre : Numéro de l'entrée	
	Type :	INT
	Plage de valeurs :	1 ... 8
`PRIO=` :	Définition de la priorité	
`<valeur>` :	Valeur de priorité	
	Plage de valeurs :	1 ... 128
	La priorité 1 correspond à la priorité maximale.	

<nom> :	Nom du sous-programme (routine d'interruption) qui doit être exécuté.
LIFTFAST :	Instruction : Relèvement rapide du contour
ALF=… :	Instruction : Sens de déplacement programmable (figure dans le bloc de déplacement) Pour les options de programmation avec ALF, voir chapitre "Sens de relèvement rapide du contour (Page 142)".

Autres conditions à prendre en compte

Comportement en cas de frame actif avec fonction miroir

Lors de la détermination du sens de relèvement, il est vérifié si un frame avec fonction miroir est actif. Dans ce cas, la gauche et la droite sont inversées pour le sens de relèvement par rapport à la direction tangentielle. Les parts directionnelles dans le sens de l'outil ne seront pas prises en compte par la fonction miroir. Ce comportement est activé par le réglage du paramètre machine :

PM21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

Exemple

Un outil brisé doit être remplacé automatiquement par un outil de rechange. L'usinage se poursuit alors avec le nouvel outil.

Programme principal :

Programme principal	**Commentaire**
N10 SETINT(1) PRIO=1 CHANG_O LIFTFAST	; Si l'entrée 1 commute, un relèvement rapide éloigne immédiatement l'outil du contour (code n° 7 pour la correction du rayon d'outil G41). Il est suivi de l'exécution de la routine d'interruption "CHANG_O".
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

Sous-programme :

Sous-programme	**Commentaire**
PROC CHANG_O SAVE	; Sous-programme avec mise en mémoire de l'état courant

Sous-programme	Commentaire
N10 G0 Z100 M5	;position de changement de l'outil, arrêt de la broche
N20 T11 M6 D1 G41	; Changement d'outil
N30 REPOSL RMBBL M3	; Réaccostage du contour et retour au programme principal (programmation dans un bloc)

2.14.7 Sens de retrait rapide du contour

Déplacement de retrait

Le plan du déplacement de retrait est déterminé par les instructions G suivants :

- LFTXT
  Le plan du déplacement de retrait est déterminé par la tangente de la trajectoire et la direction de l'outil (réglage par défaut).
- LFWP
  Le plan du déplacement de retrait est le plan de travail actif sélectionné avec les instructions G17, G18 ou G19. Le sens du déplacement de retrait est indépendant de la tangente à la trajectoire. Ceci permet de programmer un relèvement rapide paraxial.
- LFPOS
  Retrait de l'axe introduit avec POLFMASK / POLFMLIN à la position d'axe absolue programmée avec POLF.
  ALF n'a aucune influence sur la direction du relèvement pour plusieurs axes ainsi que pour plusieurs axes dans un contexte linéaire.
  Bibliographie :
  Manuel de programmation Notions de base ; chapitre : "Retrait rapide en cours de filetage"

sens de déplacement programmable (ALF=…)

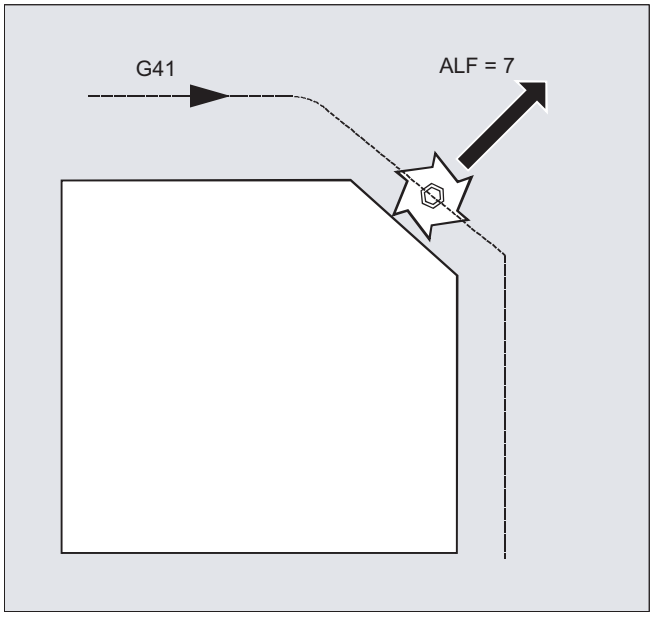
Dans le plan du mouvement de retrait, la direction est programmée avec ALF par pas discrets de 45 degrés.

Les sens de déplacement possibles sont enregistrés dans la commande sous des nº de code spéciaux et peuvent être appelés par le biais de ces numéros.

Exemple :

Code de programme
N10 SETINT(2) PRIO=1 RETR_Z LIFTFAST
ALF=7

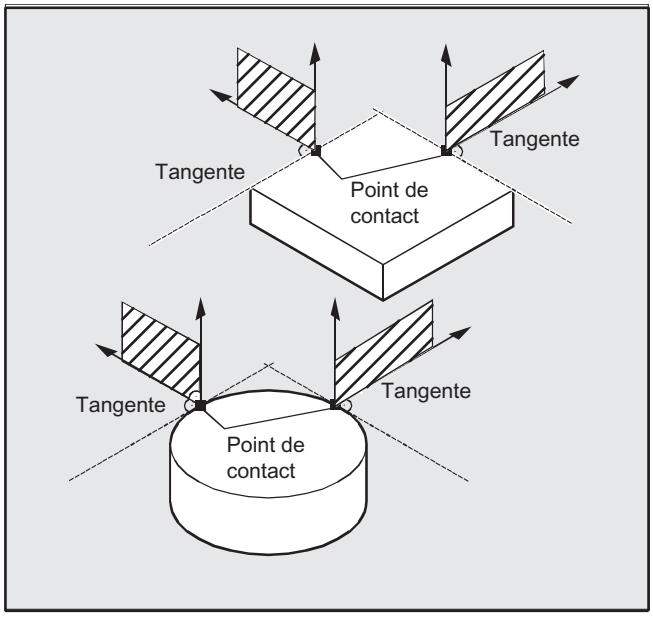
Lorsque G41 est activé (sens d'usinage à gauche du contour), le retrait de l'outil est perpendiculaire au contour.



Plan de référence pour la description du sens de déplacement avec LFTXT

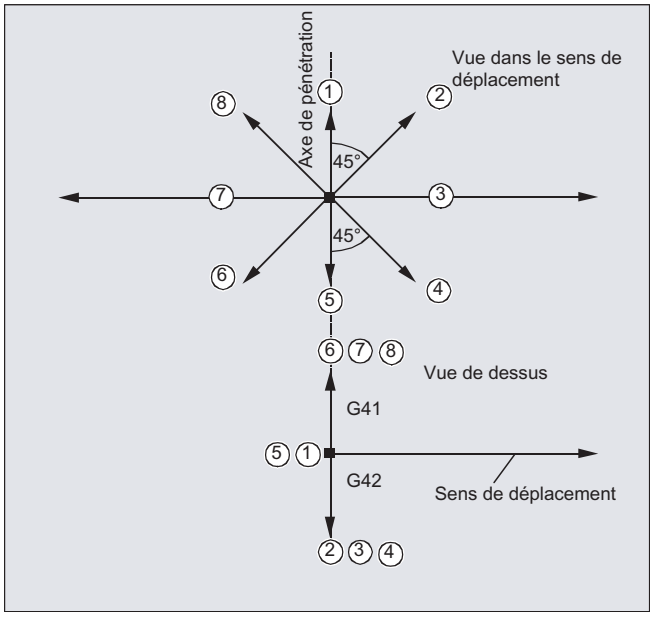
Un plan est défini au point de contact de l'outil avec le contour programmé ; ce plan sert de référence pour préciser le mouvement de relèvement avec le numéro de code correspondant.

Le plan de référence est défini à partir de l'axe longitudinal de l'outil (direction de pénétration) et d'un vecteur perpendiculaire à ce dernier et à la tangente au point de contact de l'outil avec le contour.



Numéros de code pour le sens de déplacement avec LFTXT

La figure ci-dessous contient les numéros de code avec les sens de déplacement en partant du plan de référence.



Avec `ALF=1`, le retrait a lieu dans la direction de l'outil.

Avec `ALF=0`, la fonction de relèvement rapide est désactivée.

⚠ PRUDENCE

Risque de collision

Lorsque la correction du rayon d'outil est activée, il est recommandé de ne pas utiliser :

- les codes 2, 3, 4 pour `G41`
- les codes 6, 7, 8 pour `G42`

car, dans ces cas, l'outil se déplace vers le contour et entre en collision avec la pièce.

Numéros de code pour le sens de déplacement avec LFWP

Avec `LFWP`, la direction dans le plan de travail est tributaire des configurations suivantes :

- `G17` : Plan X/Y (X/Y plane)
  `ALF=1`: retrait en direction X
  `ALF=3` : retrait en direction Y
- `G18` : Plan Z / X
  `ALF=1`: retrait en direction Z
  `ALF=3` : retrait en direction X
- `G19` : Plan Y/Z (Y/Z plane)
  `ALF=1`: retrait en direction Y
  `ALF=3` : retrait en direction Z

2.14.8 Séquence de déplacement avec des routines d'interruption

Routine d'interruption sans LIFTFAST

Les déplacements d'axe sont freinés sur la trajectoire jusqu'à l'arrêt. Puis la routine d'interruption démarre.

La position d'arrêt est enregistrée comme position d'interruption et est accostée à la fin de la routine d'interruption en présence de `REPOS` avec `RMIBL`.

Routine d'interruption avec LIFTFAST

Les déplacements d'axe sont freinés sur la trajectoire. Simultanément, le déplacement `LIFTFAST` est exécuté comme déplacement superposé. Lorsque le déplacement avec interpolation et le déplacement `LIFTFAST` se sont immobilisés, la routine d'interruption démarre.

La position où le déplacement `LIFTFAST` a démarré sur le contour et où l'outil a quitté la trajectoire est enregistrée comme position d'interruption.

Avec `LIFTFAST` et `ALF=0`, le comportement de la routine d'interruption est identique à celui de la routine sans `LIFTFAST`.

Remarque

La distance de relèvement rapide des axes géométriques par rapport au contour est réglable dans un paramètre machine.

2.15 Permutation d'axe, permutation de broche (RELEASE, GET, GETD)

Un ou plusieurs axes ou broches ne peuvent être interpolés que dans un canal. Si l'on prévoit d'utiliser un axe alternativement dans deux canaux différents (par exemple dans le cas d'un changeur de palettes), il faut d'abord le libérer dans le canal actuel, puis le prendre en charge dans l'autre canal. L'axe est permuté entre les canaux.

Extensions de permutation d'axe

Un axe/broche peut être remplacé avec un arrêt du prétraitement et une synchronisation entre le prétraitement et l'exécution des blocs ou encore sans arrêt du prétraitement. Par ailleurs, une permutation d'axe est également possible via

- une rotation conteneur d'axes AXCTSWE ou AXCTWED via un `GET/GETD` implicite.
- Frame avec rotation lorsque cet axe est combiné avec d'autres axes.
- Actions synchrones, voir Actions synchrones au déplacement, "Permutation d'axe `RELEASE`, `GET`".

Constructeur de machine

Veuillez tenir compte des indications du constructeur de machine. Des paramètres machines configurables doivent définir de manière univoque un axe pour la permutation d'axe dans tous les canaux et des paramètres machines peuvent aussi modifier le comportement de la permutation d'axe.

Syntaxe

`RELEASE (nom d'axe, nom d'axe, ...)` ou `RELEASE (S1)`

`GET (nom d'axe, nom d'axe, ...)` ou `GET (S2)`

`GETD (nom d'axe, nom d'axe, ...)` ou `GETD (S3)`

Avec GETD (GET Directly), un axe est extrait directement d'un autre canal. Cela signifie que, pour ce GETD, il n'est pas nécessaire de programmer un RELEASE approprié dans un autre canal. Cela signifie aussi que désormais, une autre communication de canal doit être mise en place (par exemple marques wait).

Signification

`RELEASE (nom d'axe, nom d'axe, …)` :	Déblocage de/des axe(s)
`GET (nom d'axe, nom d'axe, …)` :	Prise en charge de/des axe(s)
`GETD (nom d'axe, nom d'axe, …)` :	Prise en charge directe de/des axe(s)
`nom d'axe` :	Affectation des axes dans le système : AX1, AX2, … ou indication du nom des axes de la machine
`RELEASE(S1)` :	Libération des broches S1, S2, …
`GET(S2)` :	Prise en charge des broches S1, S2, …
`GETD(S3)` :	Prise en charge directe des broches S1, S2, …

Demande GET sans arrêt du prétraitement

Lorsque, après une demande GET **sans** arrêt du prétraitement, l'axe est de nouveau débloqué avec `RELEASE(Axe)` ou `WAITP(Axe)`, le `GET` qui suit entraîne un `GET` **avec** arrêt du prétraitement.

⚠ PRUDENCE

Affectation des axes modifiée

Un axe ou une broche pris(e) en charge avec GET reste assigné à ce canal, même après un RESET de touche ou de programme.

En cas de redémarrage du programme, l'affectation des axes permutés ou des broches permutées doit être programmée si les axes ou les broches doivent être utilisés dans leurs canaux d'origine.

A la mise sous tension de la CN, les axes et les broches sont affectés aux canaux spécifiés dans les paramètres machine.

Exemples

Exemple 1 : permuter des axes entre deux canaux

Sur 6 axes sont utilisés pour le traitement dans le canal 1 : 1., 2., 3. et le 4e axe.
Les axes 5 et 6 sont employés dans le canal 2 pour le changement de pièce.

L'axe 2 doit pouvoir être permuté entre les deux canaux et, lors de la mise sous tension de la CN, affecté au canal 1.

Programme "MAIN" dans le canal 1 :

Code de programme	**Commentaire**
`INIT (2,"PERMUT2")`	`; Sélection du programme PERMUT2 dans le canal 2.`
`N… START (2)`	`; Démarrage du programme dans le canal 2.`
`N… GET (AX2)`	`; Prise en charge de l'axe AX2.`
`...`	
`N… RELEASE (AX2)`	`; Libération de l'axe AX2.`
`N… WAITM (1,1,2)`	`; Attente du repère WAIT dans les canaux 1 et 2 pour la synchronisation dans les deux canaux.`
`...`	`; Poursuite de l'exécution après la permutation d'axe.`
`N… M30`	

Programme "PERMUT2" dans le canal 2 :

Programmation	**Commentaire**
`N… RELEASE (AX2)`	
`N160 WAITM (1,1,2)`	`; Attente du repère WAIT dans les canaux 1 et 2 pour la synchronisation dans les deux canaux.`
`N150 GET (AX2)`	`; Prise en charge de l'axe AX2.`
`...`	`; Poursuite de l'exécution après la permutation d'axe.`
`N… M30`	

Exemple 2 : permuter des axes sans synchronisation

Lorsque l'axe ne doit pas être synchronisé, GET ne génère pas d'arrêt du prétraitement.

Programmation	Commentaire
N01 G0 X0	
N02 RELEASE(AX5)	
N03 G64 X10	
N04 X20	
N05 GET(AX5)	; Si aucune synchronisation n'est requise, ce bloc ne sera pas exécutable.
N06 G01 F5000	; Bloc non exécutable.
N07 X20	; Bloc non exécutable, car position X comme dans N04.
N08 X30	; Premier bloc exécutable après N05.
...	

Exemple 3 : activer une permutation d'axes sans arrêt du prétraitement des blocs

Prérequis : la permutation d'axe sans arrêt du prétraitement doit être configurée via un paramètre machine.

Programmation	Commentaire
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP(B)	; L'axe B devient un axe banalisé.
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; L'axe ne déclenche pas d'arrêt du prétraitement / de REORG.
N041 G4 F2	
N050 M5	
N099 M30	

Lorsque la broche ou l'axe B sont déplacés directement après le bloc N023 comme **axe AP** par exemple sur 180 degrés puis à nouveau sur 1 degré, alors cet axe redevient un axe banalisé et ne déclenche pas d'arrêt du prétraitement dans le bloc `N40`.

plus d'informations

Conditions nécessaires à la permutation d'axe

- L'axe doit être défini avec les paramètres machine dans tous les canaux qui utilisent cet axe.
- Le canal auquel l'axe est affecté lors de la mise sous tension doit être défini à l'aide du paramètre machine spécifique à un **axe**.

Description

Libération de l'axe : RELEASE

Lors de la libération de l'axe, respecter les règles suivantes :

1. L'axe ne doit participer à aucune transformation.
2. En cas de couplage d'axes (commande tangentielle), tous les axes du groupe doivent être libérés.
3. Il est impossible de permuter un axe de positionnement concurrent dans cet état.
4. Dans le cas d'un axe maître Gantry (axe de portique), tous les axes asservis sont aussi permutés.
5. Dans le cas d'un couplage d'axes (déplacement conjugués, couplage pilote, réducteur électronique), seul l'axe pilote du groupe peut être libéré.

Prise en charge de l'axe : GET

La permutation d'axe devient effective avec cette instruction. Le canal dans lequel l'instruction a été programmée prend entièrement en charge l'axe.

Effets de GET :

Permutation d'axe avec synchronisation :

Un axe doit toujours être synchronisé s'il a été affecté entre temps à un canal différent ou à l'AP, et qu'il n'y a pas eu synchronisation avant GET par "WAITP", G74 ou effacement de la distance restant à parcourir.

- Un arrêt du prétraitement des blocs a lieu (comme pour STOPRE).
- L'exécution du programme est interrompue jusqu'à ce que la permutation d'axe soit terminée.

"GET" automatique

Si un axe est disponible par principe dans le canal, mais rendu momentanément indisponible comme axe de canal, alors un GET est automatiquement exécuté. Si le ou les axes sont déjà synchronisés, aucun arrêt du prétraitement des blocs n'a lieu.

Paramétrage du comportement de permutation d'axe

Le moment de fin de prise en compte des axes peut être paramétré de différentes façons :

- La permutation d'axe entre deux canaux s'effectue automatiquement même si l'axe a banalisé avec WAITP (comportement tel que jusqu'à présent)
- Lors de la demande d'une rotation du conteneur d'axes, tous les axes assignables au canal à traiter du conteneur d'axes sont prélevés dans le canal à l'aide de GET ou de GETD implicites. Une permutation d'axe n'est ensuite à nouveau autorisée qu'une fois la rotation du conteneur d'axe terminée.
- Lorsqu'un bloc intermédiaire a été intercalé, une vérification est effectuée lors exécution du bloc pour déterminer si une réorganisation est nécessaire ou non. Une réorganisation est nécessaire uniquement si les états de l'axe figurant dans ce bloc ne correspondent **pas** aux états actuels de l'axe.
- À la place d'un bloc GET avec arrêt du prétraitement des blocs et synchronisation entre le prétraitement et l'exécution des blocs, une permutation d'axe peut aussi avoir lieu sans arrêt du prétraitement. Un seul bloc intermédiaire est alors généré avec la demande GET. Pendant l'exécution de ce bloc durant l'exécution du programme, ce dernier vérifie si les états de l'axe dans le bloc correspondent bien aux états d'axe actuels.

Plus d'informations sur la fonction d'une permutation d'axe ou de broche, voir Description fonctionnelle Fonctions d'extension ; GMFC, canaux, permutation d'axe (K5).

2.16 Transmettre un axe à un autre canal (AXTOCHAN)

Avec l'instruction de langage AXTOCHAN, un axe peut être demandé pour transmettre cet axe à un autre canal. L'axe peut aussi bien être amené au canal correspondant par le programme pièce CN que par une action synchrone.

Syntaxe

```
AXTOCHAN(nom d'axe,numéro de canal[,nom d'axe,numéro de canal[,...]])
```

Signification

Elément	Description
AXTOCHAN :	Demande d'axe pour un canal donné
nom d'axe :	Affectation des axes dans le système : X, Y, … ou indication des noms d'axe machine concernés. Le canal à exécuter ne doit pas être son canal propre et ce ne doit pas être non plus le canal qui possède actuellement le droit d'interpolation pour l'axe
numéro de canal :	Numéro du canal devant être assigné à l'axe

Remarque

Axe de positionnement concurrent et axe exclusivement contrôlé par l'AP

Un axe AP ne peut pas changer de canal en tant qu'axe de positionnement concurrent. Un axe exclusivement contrôlé par l'AP ne peut pas être assigné au programme CN.

Bibliographie :
Description fonctionnelle Fonctions d'extension ; axes de positionnement (P2)

Exemple

AXTOCHAN dans le programme CN

Les axes X et Y sont connus dans le 1er et dans le 2e canal. Actuellement, c'est le canal 1 qui détient le droit d'interpolation et le programme suivant démarre dans le canal 1:

```
Code de programme                Commentaire
N110 AXTOCHAN(Y,2)               ; Translater l'axe Y dans le 2e canal.
N111 M0
N120 AXTOCHAN(Y,1)               ; Aller reprendre l'axe Y (banalisé).
N121 M0
N130 AXTOCHAN(Y,2,X,2)           ; Translater l'axe Y et l'axe X dans le 2e canal (axes
                                   banalisés).
N131 M0
N140 AXTOCHAN(Y,2)               ; Translater l'axe Y dans le 2e canal (programme CN).
N141 M0
```

Informations complémentaires

AXTOCHAN dans le programme CN

Un `GET` est alors exécuté uniquement en cas de demande de l'axe pour le programme CN dans un canal séparé et la modification d'état réelle est attendue. Lorsque l'axe est demandé pour un autre canal ou s'il doit devenir un axe banalisé dans un canal séparé, alors seulement la demande est générée.

AXTOCHAN à partir d'une action synchrone

Lorsqu'un axe est demandé pour un canal séparé alors `AXTOCHAN` est représenté sur un `GET` à partir d'une action synchrone. Dans ce cas, l'axe devient axe banalisé à la première demande pour le canal propre. À la deuxième demande, l'axe est affecté au programme CN d'une manière analogue à la demande GET dans le programme CN. En ce qui concerne la demande GET à partir d'une action synchrone, se reporter au chapitre "Actions synchrones au déplacement".

2.17 Activation des paramètres machine (NEWCONF)

L'instruction NEWCONF permet d'activer tous les paramètres machine. La fonction peut également être activée dans l'interface utilisateur IHM en actionnant la touche logicielle "Activer PM".

Lors de l'exécution de la fonction "NEWCONF", un arrêt implicite du prétraitement des blocs a lieu, c'est-à-dire que le déplacement avec interpolation est interrompu.

Syntaxe

```
NEWCONF
```

Signification

NEWCONF :	Instruction de prise d'effet de l'ensemble des paramètres machine du niveau de prise d'effet "NEW_CONFIG"

Exécution de NEWCONF pour tous les canaux depuis le programme pièce

Si des paramètres machine axiaux du programme pièce sont modifiés puis activés avec NEWCONF, alors l'instruction NEWCONF active uniquement les paramètres machine entraînant des modifications du canal du programme pièce.

Remarque

Afin d'activer toutes les modifications de façon sûre, l'instruction NEWCONF doit être exécutée dans chaque canal dans lequel doivent être calculés les axes ou fonctions concernés par les modifications des paramètres machine.

Avec NEWCONF, aucun paramètre machine axial n'est activé.

Un RESET axial doit être exécuté pour les axes contrôlés par l'AP.

Exemple

Réseau de trous : réaliser des trous taraudés en différentes positions

```
Code de programme                          Commentaire
N10 $MA_CONTOUR_TOL[AX]=1.0                ; Modifier un paramètre machine.
N20 NEWCONF                                ; Activer des paramètres machine.
...
```

2.18 Ecriture d'un fichier (WRITE)

L'instruction `WRITE` permet d'écrire des blocs/données du programme CN à la fin d'un fichier se trouvant dans le système de fichiers passif ou sur une mémoire de programmes externe (fichier journal). Cette écriture peut également concerner le programme en cours d'exécution.

Remarque

Si le fichier à décrire avec l'instruction `WRITE` n'existe pas dans la mémoire de programmes, il est créé.

Condition

Le niveau de protection réglé doit être supérieur ou égal au droit WRITE du fichier. Si ce n'est pas le cas, l'accès est refusé avec émission du message d'erreur (valeur de retour de la variable d'erreur = 13).

Syntaxe

```
DEF INT <erreur>
...
WRITE(<erreur>,"<nom de fichier>"/"PériphExt","<bloc/données>")
```

Signification

`WRITE` :	Instruction d'insertion d'un bloc ou de données à la fin du fichier spécifié		
`<erreur>` :	**Paramètre 1 :** variable pour signalisation en retour de la valeur d'erreur		
	Type :	INT	
	Valeur :	0	Aucune erreur
		1	Chemin d'accès non autorisé
		2	Chemin d'accès introuvable
		3	Fichier introuvable
		4	Type de fichier incorrect
		10	Fichier plein
		11	Fichier en cours d'utilisation
		12	Pas de ressources disponibles
		13	Pas de droits d'accès
		14	Absence ou échec d'EXTOPEN pour le périphérique de sortie
		15	Erreur lors de l'écriture sur le périphérique de sortie
		16	Chemin externe programmé non valide

<nom de fichier> :	**Paramètre 2 :** nom du fichier dans lequel le bloc ou les données indiqués doivent être ajoutés	
	Type :	STRING
	Le chemin absolu peut être indiqué avant le nom de fichier proprement dit. En l'absence de chemin d'accès, le fichier est recherché dans le répertoire courant (= répertoire du programme sélectionné). Pour les règles concernant le chemin d'accès, voir "Adressage des fichiers de la mémoire de programmes (Page 227)".	
<PériphExt> :	Si des données doivent être transmises à un périphérique/fichier externe via la fonction "Process DataShare", le descripteur symbolique du périphérique/fichier externe à ouvrir doit être spécifié à la place du nom de fichier.	
	Type :	STRING
	Pour plus d'informations, voir "Process DataShare - Sortie sur un appareil/fichier externe (EXTOPEN, WRITE, EXTCLOSE) (Page 700)". **Remarque :** Le descripteur doit être identique à celui spécifié dans l'instruction EXTOPEN.	
<bloc/ données> :	**Paramètre 3 :** bloc ou données à ajouter dans le fichier spécifié.	
	Type :	STRING

Remarque

Lors de l'écriture dans le système de fichiers passif ou sur une mémoire de programmes externe, l'instruction WRITE insère implicitement un caractère "LF" (LINE FEED = nouvelle ligne) à la fin de la chaîne de sortie.

Ce comportement ne s'applique pas à la sortie sur un périphérique/fichier externe via la fonction "Process DataShare". Si un "LF" doit être transmis en même temps, ceci doit être indiqué explicitement dans la chaîne de sortie.

→ Sur ce point, voir exemple 3 : "LF" implicite/explicite.

Autres conditions à prendre en compte

- **Taille maximale du fichier (→ constructeur de machine !)**
  La taille maximale possible des fichiers journaux dans le système de fichiers passif est réglée avec le paramètre machine :
  PM11420 $MN_LEN_PROTOCOL_FILE
  La taille maximale s'applique à tous les fichiers créés dans le système de fichiers passif avec l'instruction WRITE. En cas de dépassement, un message d'erreur est généré et l'enregistrement du bloc ou des données ne s'effectue pas. Si la capacité mémoire est suffisante, un nouveau fichier peut être créé.

Exemples

Exemple 1 : instruction WRITE dans le système de fichiers passif sans chemin d'accès absolu

Code de programme	Commentaire
N10 DEF INT ERROR	; Définition des variables d'erreur.

Code de programme	Commentaire
N20 WRITE(ERROR,"PROT","JOURNAL DU 07/02/97")	; Ecrire le texte "JOURNAL DU 07/02/97" dans le fichier _N_PROT_MPF.
N30 IF ERROR	; Analyse des erreurs.
N40 MSG ("Erreur lors instruction WRITE:" << ERROR)	
N50 M0	
N60 ENDIF	
...	

Exemple 2 : Instruction WRITE dans le système de fichiers passif avec chemin d'accès absolu

```
Code de programme
...
WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","JOURNAL DU
07/02/97")
...
```

Exemple 3 : "LF" implicite/explicite

a, écriture dans le système de fichiers passif avec le "LF" généré de façon implicite

```
Code de programme
...
N110 DEF INT ERROR
N120 WRITE(ERROR,"/_N_MPF_DIR/_N_MYPROTFILE_MPF","MY_STRING")
N130 WRITE(ERROR,"/_N_MPF_DIR/_N_MYPROTFILE_MPF","MY_STRING")
N140 M30
```

Résultat en sortie :

MY_STRING

MY_STRING

b, écriture dans un fichier externe sans "LF" généré de façon implicite

```
Code de programme
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING")
N250 WRITE(ERROR,DEV_1,"MY_STRING")
N260 EXTCLOSE(ERROR,DEV_1)
```

Code de programme

```
N270 M30
```

Résultat en sortie :

MY_STRINGMY_STRING

c, écriture dans un fichier externe avec un "LF" programmé de façon explicite

Pour obtenir le même résultat que pour a, la programmation suivante doit être effectuée :

Code de programme

```
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")
N250 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30
```

Résultat en sortie :

MY_STRING

MY_STRING

2.19 Suppression d'un fichier (DELETE)

L'instruction `DELETE` permet de supprimer tous les fichiers, qu'ils aient été créés avec l'instruction `WRITE` ou non. `DELETE` permet également de supprimer les fichiers ayant été créés avec des niveaux d'accès supérieurs.

Syntaxe

```
DEF INT <erreur>
DELETE(<erreur>,"<nom de fichier>")
```

Signification

`DELETE` :	Instruction de suppression du fichier spécifié		
`<erreur>` :	Variable pour la valeur d'erreur en retour		
	type.	INT	
	Valeur :	0	Aucune erreur
		1	Chemin d'accès non autorisé
		2	Chemin d'accès introuvable
		3	Fichier introuvable
		4	Type de fichier incorrect
		11	Fichier en cours d'utilisation
		12	Pas de ressources disponibles
		20	Autre erreur
`<nom de fichier>` :	Nom du fichier à supprimer		
	Type :	STRING	
	Le chemin absolu peut être indiqué avant le nom de fichier proprement dit. En l'absence de chemin d'accès, le fichier est recherché dans le répertoire courant (= répertoire du programme sélectionné). Pour les règles concernant le chemin d'accès, voir "Adressage des fichiers de la mémoire de programmes (Page 227)".		

Exemple

Code de programme	**Commentaire**
`N10 DEF INT ERROR`	`; Définition des variables d'erreur.`
`N15 STOPRE`	`; Arrêt du prétraitement des blocs.`
`N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")`	`; Supprimer le fichier TEST1 du répertoire du sous-programme.`
`N30 IF ERROR`	`; Analyse des erreurs.`
`N40 MSG ("Erreur lors instruction DELETE :" <<ERROR)`	
`N50 M0`	
`N60 ENDIF`	

2.20 Lecture des lignes d'un fichier (READ)

L'instruction `READ` permet de lire une ou plusieurs lignes dans le fichier spécifié et d'enregistrer les informations lues dans un tableau de type STRING. Chaque ligne lue occupe un élément de ce tableau.

Condition

Le niveau de protection réglé doit être supérieur ou égal au droit READ du fichier. Si ce n'est pas le cas, l'accès est refusé avec émission du message d'erreur (valeur de retour de la variable d'erreur = 13).

Syntaxe

```
DEF INT <erreur>
DEF STRING[<longueur de chaîne>] <résultat>[<n>,<m>]
READ(<erreur>,"<nom de fichier>",<ligne de début>,<nombre de
lignes>,<résultat>)
```

Signification

`READ` :	Instruction de lecture des lignes du fichier spécifié et d'enregistrement de ces lignes dans un tableau de variables.		
`<erreur>` :	Variable pour la valeur d'erreur en retour (paramètre Call-By-Reference)		
	type.	INT	
	Valeur :	0	Aucune erreur
		1	Chemin d'accès non autorisé
		2	Chemin d'accès introuvable
		3	Fichier introuvable
		4	Type de fichier incorrect
		11	Fichier en cours d'utilisation
		13	Droits d'accès insuffisants
		21	Ligne inexistante (paramètres `<ligne de début>` ou `<nombre de lignes>` supérieurs au nombre de lignes du fichier spécifié).
		22	Longueur de tableau de la variable de résultat (`<résultat>`) trop petite.
		23	Zone de lignes trop grande (paramètre `<nombre de lignes>` trop grand, si bien que la lecture se poursuit au-delà de la fin du fichier).
`<nom de fichier>` :	Nom du fichier à lire (paramètre Call-By-Value)		
	Type :	STRING	
	Le chemin absolu peut être indiqué avant le nom de fichier proprement dit. En l'absence de chemin d'accès, le fichier est recherché dans le répertoire courant (= répertoire du programme sélectionné). Pour les règles concernant le chemin d'accès, voir "Adressage des fichiers de la mémoire de programmes (Page 227)".		

<ligne de début> :	Ligne de début de la zone de fichier à lire (paramètre Call-By-Value)		
	Type :	INT	
	Valeur :	0	Le nombre de lignes, avant la fin du fichier, indiqué par le paramètre <nombre de lignes> est lu.
		1 … n	Numéro de la première ligne à lire.
<nombre de lignes> :	Nombre de lignes à lire (paramètre Call-By-Value)		
	Type :	INT	
<résultat> :	Variable de résultat (paramètre Call-By-Reference) Tableau de variables dans lequel le texte lu est enregistré.		
	Type :	STRING (longueur max. : 255)	
	Si le nombre de lignes indiqué dans le paramètre <nombre de lignes> est inférieur à la dimension du tableau [<n>,<m>] de la variable de résultat, les autres éléments du tableau ne sont pas modifiés. Les caractères de commande de fin de ligne "LF" (Line Feed) ou "CR LF" (Carrige Return Line Feed) ne sont **pas** enregistrés dans la variable de résultat. Si la longueur de la ligne est supérieure à la longueur de chaîne définie, les lignes lues sont tronquées. Il n'y a pas de signalisation d'erreur.		

Remarque

La lecture dans un fichier binaire est impossible. L'erreur "Type de fichier incorrect" (valeur de retour de la variable d'erreur = 4) est signalée. Les types de fichiers suivants ne sont pas lisibles : _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Exemple

Code de programme	Commentaire
N10 DEF INT ERROR	; Définition des variables d'erreur.
N20 DEF STRING[255] RESULT[5]	; Définition des variables de résultat.
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF", 1,5,RESULT)	; Nom du fichier avec identificateur de domaine, extension de fichier et chemin d'accès.
N40 IF ERROR <>0	; Analyse des erreurs.
N50 MSG("ERREUR"<<ERROR<<"POUR INSTRUCTION READ")	
N60 M0	
N70 ENDIF	
...	

2.21 Vérification de l'existence d'un fichier (ISFILE)

L'instruction `ISFILE` permet de vérifier si un fichier existe dans la mémoire de programmes.

Syntaxe

```
<Résultat>=ISFILE("<nom de fichier>")
```

Signification

`ISFILE`:	Instruction pour vérifier l'existence d'un fichier		
`<nom de fichier>`:	Nom du fichier dont l'existence doit être vérifiée.		
	Type :	STRING	
	Le chemin absolu peut être indiqué avant le nom de fichier proprement dit. En l'absence de chemin d'accès, le fichier est recherché dans le répertoire courant (= répertoire du programme sélectionné). Pour les règles concernant le chemin d'accès, voir "Adressage des fichiers de la mémoire de programmes (Page 227)".		
`<résultat>`:	Variable de résultat pour l'acquisition du résultat de la vérification		
	type.	BOOL	
	Valeur :	TRUE	Fichier existant
		FALSE	Fichier inexistant

Exemples

Exemple 1

Code de programme	**Commentaire**
`N10 DEF BOOL RESULT`	`; Définition des variables de résultat.`
`N20 RESULT=ISFILE("TESTFILE")`	
`N30 IF(RESULT==FALSE)`	
`N40  MSG("FICHIER INEXISTANT")`	
`N50  M0`	
`N60 ENDIF`	
`...`	

Exemple 2

Code de programme	**Commentaire**
`N10 DEF BOOL RESULT`	`; Définition des variables de résultat.`
`N20 RESULT=ISFILE("TESTFILE")`	
`N30 IF(NOT ISFILE("TESTFILE"))`	
`N40  MSG("FICHIER INEXISTANT")`	
`N50  M0`	
`N60 ENDIF`	
`...`	

2.22 Lecture des informations fichier (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Les instructions FILEDATE, FILETIME, FILESIZE, FILESTAT et FILEINFO permettent de lire des informations fichier telles que date/heure du dernier accès en écriture, taille actuelle du fichier, état du fichier ou l'ensemble de ces informations.

Condition

Le niveau de protection actuellement paramétré doit être égal ou supérieur au droit show du répertoire principal. Si ce n'est pas le cas, l'accès est refusé avec émission du message d'erreur (valeur de retour de la variable d'erreur = 13).

Syntaxe

```
FILE....(<erreur>,"<nom de fichier>",<résultat>)
```

Signification

FILEDATE:	Fournit la **date** du dernier accès en écriture à un fichier		
FILETIME:	Fournit l'**heure** du dernier accès en écriture à un fichier		
FILESIZE:	Fournit la **taille actuelle** d'un fichier		
FILESTAT:	Fournit l'**état** d'un fichier se rapportant aux **droits** suivants : • Lecture (r : read) • Ecriture (w : write) • Exécution (x : execute) • Affichage (s : show) • Suppression (d : delete) **Remarque :** Ces niveaux de protection sont des propriétés spéciales du système de fichiers passif. Ainsi, lors de l'accès à des mémoires de programmes externes, FILESTAT ne fournit que les droits d'accès par défaut (77777).		
FILEINFO:	Fournit, pour un fichier, l'**ensemble des informations** lisibles via FILEDATE, FILETIME, FILESIZE et FILESTAT.		
<erreur> :	Variable pour la valeur d'erreur en retour (paramètre Call-By-Reference)		
	type.	VAR INT	
	Valeur :	0	Aucune erreur
		1	Chemin d'accès non autorisé
		2	Chemin d'accès introuvable
		3	Fichier introuvable
		4	Type de fichier incorrect
		13	Droits d'accès insuffisants
		22	Longueur de chaîne de la variable de résultat (<résultat>) trop petite.

<nom de fichier> :	Nom du fichier dont les informations de fichier doivent être lues			
	Type :	CHAR[160]		
	Le chemin absolu peut être indiqué avant le nom de fichier proprement dit. En l'absence de chemin d'accès, le fichier est recherché dans le répertoire courant (= répertoire du programme sélectionné). Pour les règles concernant le chemin d'accès, voir "Adressage des fichiers de la mémoire de programmes (Page 227)".			
<résultat> :	Variable de résultat (paramètre Call-By-Reference) Variable dans laquelle l'information de fichier interrogée est enregistrée.			
	Type :	VAR CHAR[8]	pour	FILEDATE Format : "dd.mm.yy"
		VAR CHAR[8]	pour	FILETIME Format : "hh:mm.ss"
		VAR INT	pour	FILESIZE La taille du fichier est indiquée en octets.
		VAR CHAR[5]	pour	FILESTAT Format : "rwxsd" (r : **r**ead, w : **w**rite, x : e**x**ecute, s : **s**how, d : **d**elete)
		VAR CHAR[32]	pour	FILEINFO Format : "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

Exemple

Code de programme	**Commentaire**
N10 DEF INT ERROR	; Définition des variables d'erreur.
N20 STRING[32] RESULT	; Définition des variables de résultat.
N30 FILEINFO(ERROR,"/_N_MPF_DIR/_N_TESTFILE_MPF",RESULT)	; Nom du fichier avec identificateur de domaine, extension de fichier et chemin d'accès.
N40 IF ERROR <> 0	; Analyse des erreurs
N50 MSG("ERREUR"<<ERROR<<"SUR L'INSTRUCTION FILEINFO")	
N60 M0	
N70 ENDIF	
...	

L'exemple pourrait p. ex. fournir le résultat suivant dans la variable de résultat RESULT :

"77777 12345678 26.05.00 13:51:30"

2.23 Arrondissement (ROUNDUP)

La fonction "ROUNDUP" arrondit les valeurs d'entrée de type REAL (nombres rationnels avec point décimal) à l'entier supérieur le plus proche.

Syntaxe

```
ROUNDUP(<valeur>)
```

Signification

`ROUNDUP` :	Instruction d'arrondissement d'une valeur d'entrée
`<valeur>` :	Valeur d'entrée du type REAL

Remarque

Les valeurs d'entrée de type INT (nombres entiers) restent inchangées.

Exemples

Exemple 1 : arrondis de différentes valeurs d'entrée

Exemple	Arrondi
`ROUNDUP(3.1)`	4.0
`ROUNDUP(3.6)`	4.0
`ROUNDUP(-3.1)`	-3.0
`ROUNDUP(-3.6)`	-3.0
`ROUNDUP(3.0)`	3.0
`ROUNDUP(3)`	3.0

Exemple 2 : ROUNDUP dans le programme CN

Code de programme

```
N10 X=ROUNDUP(3.5) Y=ROUNDUP(R2+2)
N15 R2=ROUNDUP($AA_IM[Y])
N20 WHEN X=100 DO Y=ROUNDUP($AA_IM[X])
...
```

2.24 Technique des sous-programmes

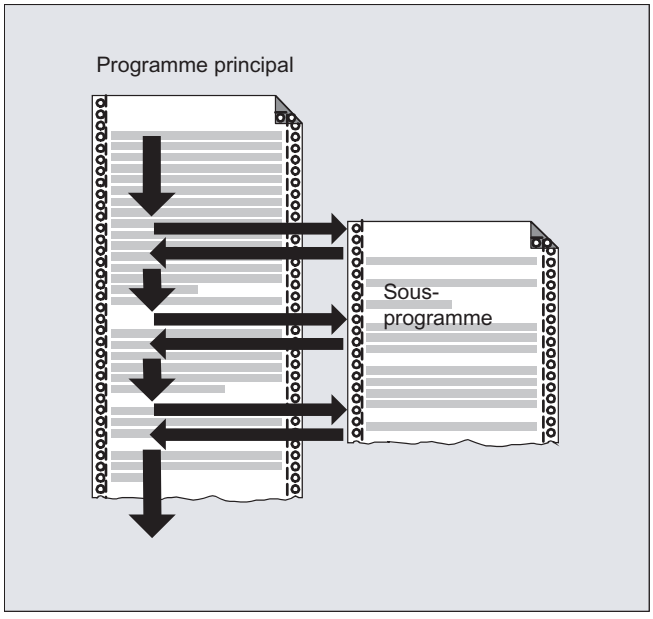
2.24.1 Généralités

2.24.1.1 Sous-programme

La désignation "Sous-programme" date encore de l'époque où les programmes pièce étaient répartis de façon déterminée en programmes principaux et sous-programmes. Les programmes principaux correspondaient aux programmes pièce qui, en vue de leur exécution, étaient sélectionnés, puis démarrés sur la commande. Les sous-programmes correspondaient aux programmes pièce qui étaient appelés par le programme principal.

Cette répartition déterminée n'a plus lieu d'être avec la langage CN SINUMERIK actuel. Chaque sous-programme peut par principe être sélectionné et démarré comme programme principal ou être appelé comme sous-programme par un autre programme pièce.

Ainsi, nous désignerons par la suite par sous-programme, un programme pièce appelé par un autre programme pièce.



Application

Comme dans tous les langages de programmation évolués, les sous-programmes dans le langage CN servent également à transférer des sections de programme, qui sont exécutées plusieurs fois, en tant que programmes complets, autonomes.

Les sous-programmes offrent les avantages suivants :

- Augmentation de la clarté et de la lisibilité des programmes
- Augmentation de la qualité grâce à la réutilisation de sections de programme testées

- Possibilité de créer des bibliothèques d'usinage spécifiques
- Economie de capacité mémoire

2.24.1.2 Nom de sous-programme

Règles de dénomination

Le nom du sous-programme peut être choisi librement dans la mesure où les règles suivantes sont respectées :

- Caractères autorisés :
  - Caractères alphabétiques : A ... Z, a ... z
  - Chiffres : 0 ... 9
  - Tiret bas : (_)
- Les deux premiers caractères doivent être deux lettres, ou bien un tiret bas suivi d'une lettre.

  Remarque

  Si cette condition est remplie, un programme CN peut être appelé, par simple indication du nom du programme, comme sous-programme depuis un autre programme. En revanche, si le nom du programme commence par des chiffres, l'appel en tant que sous-programme est possible uniquement avec l'instruction CALL.

- Longueur maximale : 24 caractères

Remarque

Majuscules/minuscules

Dans le langage CN SINUMERIK, **aucune** distinction n'est faite entre majuscules et minuscules.

Remarque

Noms de sous-programme interdits

Pour éviter tout problème avec les applications Windows, les noms de programme suivants ne doivent **pas** être utilisés :

- CON, PRN, AUX, NUL
- COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9
- LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

Extensions internes à la commande

Un préfixe et un postfixe sont ajoutés en interne à la commande comme extensions au nom de sous-programme, attribué lors la création de celui-ci :

- Préfixe : `_N_`
- Postfixe : `_SPF`

Utilisation du nom de programme

Pour l'utilisation du nom de programme, p. ex. lors d'un appel de sous-programme, toutes les combinaisons entre préfixe, nom de programme et postfixe sont possibles.

Exemple :

Le sous-programme avec le nom `SUB_PROG` peut être démarré par les descripteurs suivants :

1. `SUB_PROG`
2. `_N_SUB_PROG`
3. `SUB_PROG_SPF`
4. `_N_SUB_PROG_SPF`

Noms identiques pour programmes principaux et sous-programmes

S'il existe un programme principal (.MPF) et un sous-programme (.SPF) portant le même nom de programme, l'extension de fichier correspondante doit être indiquée à des fins d'identification univoque lors de l'utilisation du nom du programme dans le programme CN. Sinon, le programme utilisé sera le premier trouvé avec le nom indiqué dans le chemin de recherche.

2.24.1.3 Imbrication de sous-programmes

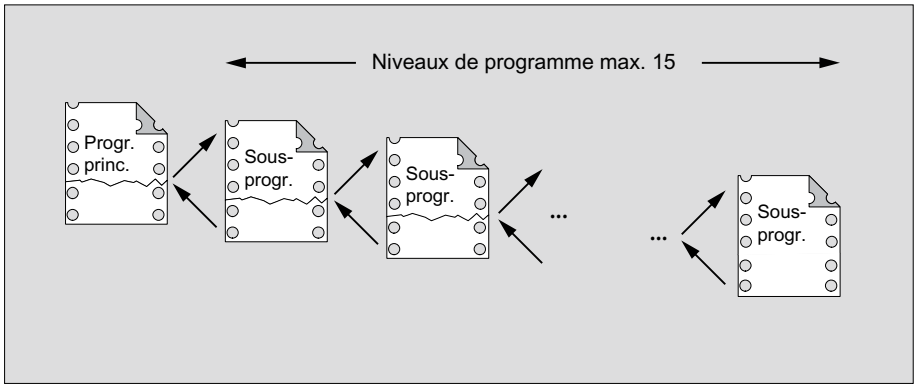
Un programme principal peut appeler des sous-programmes qui, à leur tour, appellent des sous-programmes. L'exécution des programmes est ainsi imbriquée. Chaque programme s'exécute dans un niveau de programme spécifique.

Niveaux d'imbrication

Le langage CN met actuellement à disposition 16 niveaux de programme. Le programme principal s'exécute toujours dans le niveau de programme supérieur, le niveau 0. Un sous-programme s'exécute toujours dans le niveau de programme directement inférieur consécutif à l'appel. Le niveau de programme 1 correspond ainsi au premier niveau de sous-programme.

Cloisonnement des niveaux de programme :

- Niveau de programme 0 : Niveau de programme principal
- Niveau de programme 1 - 15 : Niveau de sous-programme 1 - 15



Routines d'interruption (ASUP)

Si un sous-programme est appelé dans le cadre d'une routine d'interruption, il n'est pas traité dans le niveau de programme actuel (n) dans le canal actif, mais également dans le niveau de programme immédiatement consécutif (n+1). Afin que ceci soit également possible dans le niveau de programme le plus bas, 2 niveaux de programme supplémentaires (16 et 17) sont disponibles en ce qui concerne les routines d'interruption.

Si plus de 2 niveaux de programme sont requis, il faut en prendre compte de manière explicite dans la structuration du programme pièce exécuté dans le canal. Autrement dit, le nombre maximum de niveaux de programme utilisé doit ne doit pas faire en sorte qu'il n'en existe pas suffisamment pour l'exécution de l'interruption.

Si l'exécution de l'interruption requiert p. ex. 4 niveaux de programme, le programme pièce doit être structuré de sorte à ce qu'il occupe au maximum le niveau de programme 13. Si une interruption a ensuite lieu, les 4 niveaux de programme (14 à 17) qu'elle requiert sont disponibles.

Cycles Siemens

Les cycles Siemens requièrent 3 niveaux de programme. L'appel d'un cycle Siemens doit donc être réalisé au plus tard lors de :

- l'exécution du programme pièce : niveau de programme 12
- la routine d'interruption : niveau de programme 14

2.24.1.4 Chemin de recherche

Lors de l'appel d'un sous-programme sans indication de chemin, la commande effectue une recherche, selon un ordre prédéfini (voir "Chemin de recherche pour l'appel de sous-programmes (Page 232)"), dans la mémoire de programme existante.

2.24.1.5 Paramètres formels et paramètres effectifs

On parle de paramètres formels et de paramètres effectifs dans le contexte de la définition et de l'appel de sous-programmes avec transfert de paramètres.

Paramètres formels

Lors de la définition d'un sous-programme, les paramètres à transférer au sous programme, appelés paramètres formels, doivent être définis avec leur type et nom de paramètre.

Les paramètres formels définissent ainsi l'interface du sous-programme.

Exemple :

Code de programme	**Commentaire**
`PROC CONTOUR (REAL X, REAL Y)`	`; Paramètres formels : X et Y, tous deux du type REAL`
`N20 X1=X Y1=Y`	`; Déplacement de l'axe X1 à la position X et de l'axe Y1 à la position Y`
`...`	
`N100 RET`	

Paramètres effectifs

A l'appel d'un sous-programme, des valeurs absolues ou variables, appelés paramètres formels, doivent être transférées au sous-programme.

Les paramètres effectifs fournissent ainsi des valeurs effectives à l'interface du sous-programme.

Exemple :

Code de programme	**Commentaire**
`N10 DEF REAL LARGEUR`	`; Définition de variable`
`N20 LARGEUR=20.0`	`; Affectation de variable`
`N30 CONTOUR (5.5, LARGEUR)`	`; Appel de sous-programme avec paramètres effectifs : 5.5 et LARGEUR`
`...`	
`N100 M30`	

2.24.1.6 Transfert de paramètres

Définition d'un sous-programme avec transfert de paramètres

La définition d'un sous-programme avec transfert de paramètres s'effectue avec le mot-clé `PROC` et une liste complète de tous les paramètres attendus par le sous-programme.

Transfert incomplet de paramètres

A l'appel du sous-programme, il n'est pas toujours nécessaire de transférer de manière explicite l'ensemble des paramètres définis dans l'interface du sous-programme. Si un paramètre est omis, la valeur par défaut "0" est transférée pour ce paramètre.

Pour une identification univoque de l'ordre des paramètres, les virgules doivent cependant toujours être indiquées comme caractères de séparation des paramètres. Le dernier paramètre constitue une exception. Si celui-ci est omis à l'appel, la dernière virgule peut également l'être.

Exemple :

Sous-programme :

Code de programme	**Commentaire**
`PROC SUB_PROG (REAL X, REAL Y, REAL Z)`	`; Paramètres formels : X, Y et Z.`
`...`	
`N100 RET`	

Programme principal :

Code de programme	**Commentaire**
`PROC MAIN_PROG`	
`...`	
`N30 SUB_PROG(1.0,2.0,3.0)`	`; Appel de sous-programme avec transfert complet de paramètres :` `X=1.0, Y=2.0, Z=3.0`
…	
`N100 M30`	

Exemples d'appel de sous-programme dans `N30`, avec indication complète des paramètres :

```
N30 SUB_PROG( ,2.0,3.0)          ; X=0.0, Y=2.0, Z=3.0
N30 SUB_PROG(1.0, ,3.0)          ; X=1.0, Y=0.0, Z=3.0
N30 SUB_PROG(1.0,2.0)            ; X=1.0, Y=2.0, Z=0.0
N30 SUB_PROG( , ,3.0)            ; X=0.0, Y=0.0, Z=3.0
N30 SUB_PROG( , , )              ; X=0.0, Y=0.0, Z=0.0
```

IMPORTANT

Transfert de paramètres de type Call-by-Reference

Les paramètres transférés via Call-by-Reference ne doivent pas être omis à l'appel du sous-programme.

IMPORTANT

Type de données AXIS

Les paramètres du type de données AXIS ne doivent pas être omis à l'appel du sous-programme.

Vérification des paramètres de transfert :

La variable système $P_SUBPAR [n] avec n = 1, 2, ... permet de vérifier dans le sous-programme, si un paramètre a été transféré de manière explicite ou omis. L'indice n se rapporte à l'ordre des paramètres formels. L'indice n = 1 se rapporte au 1er paramètre formel, l'indice n = 2 au 2e paramètre formel, etc.

L'extrait de programme suivant illustre à titre d'exemple comment réaliser une vérification pour le 1er paramètre formel :

Programmation	**Commentaire**
`PROC SUB_PROG (REAL X, REAL Y, REAL Z)`	`; Paramètres formels : X, Y et Z.`
`N20 IF $P_SUBPAR[1]==TRUE`	`; Vérification du 1er paramètre formel X.`
`...`	`; Ces actions sont réalisées si le paramètre formel X a été transféré de manière explicite.`
`N40 ELSE`	
`...`	`; Ces actions sont réalisées si le paramètre formel X n'a pas été transféré.`
`N60 ENDIF`	
`...`	`; Actions générales`
`N100 RET`	

2.24.2 Définition d'un sous-programme

2.24.2.1 Sous-programme sans transfert de paramètres

Dans le cas de la définition de sous-programmes sans transfert de paramètres, la ligne de définition au début du programme n'est pas nécessaire.

Syntaxe

```
[PROC <nom de programme>]
...
```

Signification

`PROC` :	Instruction de définition au début d'un programme
`<nom du programme>` :	Nom du programme

Exemple

Exemple 1 : Sous-programme avec l'instruction `PROC`

Code de programme	Commentaire
`PROC SUB_PROG`	`; Ligne de définition`
`N10 G01 G90 G64 F1000`	
`N20 X10 Y20`	
`...`	
`N100 RET`	`; Retour en sautant des sous-programmes`

Exemple 2 : Sous-programme sans instruction `PROC`

Code de programme	Commentaire
`N10 G01 G90 G64 F1000`	
`N20 X10 Y20`	
`...`	
`N100 RET`	`; Retour en sautant des sous-programmes`

Voir aussi

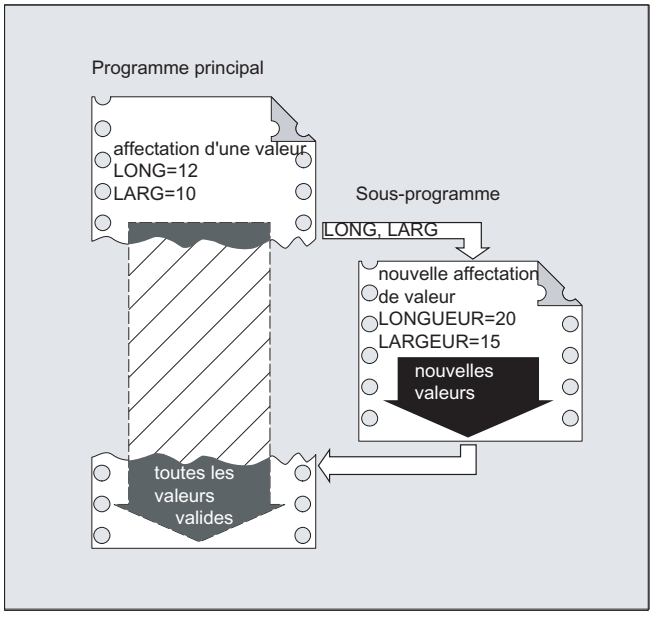
Appel de sous-programme sans transfert de paramètres (Page 199)

2.24.2.2 Sous-programme avec transfert de paramètres Call-by-Value (PROC)

La définition d'un sous-programme avec transfert de paramètres Call-by-Value s'effectue avec le mot-clé `PROC`, suivi du nom de programme et d'une liste complète de tous les paramètres, y compris le type et le nom. L'instruction de définition doit figurer dans la première ligne du programme.

Call-by-Value

Avec un transfert de paramètres Call-by-Value, le programme appelant transfère au sous-programme uniquement la valeur d'une variable. Ainsi, le sous-programme n'obtient pas d'accès direct à la variable. Ainsi, en cas de modification de la valeur de paramètre, seule la valeur visible dans le sous-programme est modifiée. La valeur des variables définies dans le programme appelant reste inchangée. Le transfert de paramètres Call-by-Value n'a donc aucun effet sur le programme appelant.



Syntaxe

```
PROC <nom de programme> (<type de paramètre> <nom de
paramètre>=<valeur d'initialisation>, ...)
```

Remarque

Il est possible de transférer jusqu'à 127 paramètres.

Signification

`PROC` :	Instruction de définition au début d'un programme
`<nom du programme>` :	Nom du programme
`<type de paramètre>` :	Type de données du paramètre (p. ex. REAL, INT, BOOL)
`<nom de paramètre>` :	Nom du paramètre
`<valeur d'initialisation>` :	Valeur facultative pour l'initialisation du paramètre (**facultatif**) Si aucun paramètre n'est spécifié lors de l'appel du sous-programme, la valeur d'initialisation est affectée au paramètre.

Exemples

Exemple 1

Définition d'un sous-programme SUB_PROG avec trois paramètres de type REAL et leurs valeurs par défaut :

Code de programme

```
PROC SUB_PROG(REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HEIGHT=30.0)
```

Exemple 2

Différentes variantes d'appel

Code de programme

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; variantes d'appel
  SUB_PROG
  SUB_PROG(PAR_1, PAR_2, PAR_3)
  SUB_PROG(PAR_1)
  SUB_PROG(PAR_1, , PAR_3)
  SUB_PROG( , , PAR_3)
N100 RET
```

Voir aussi

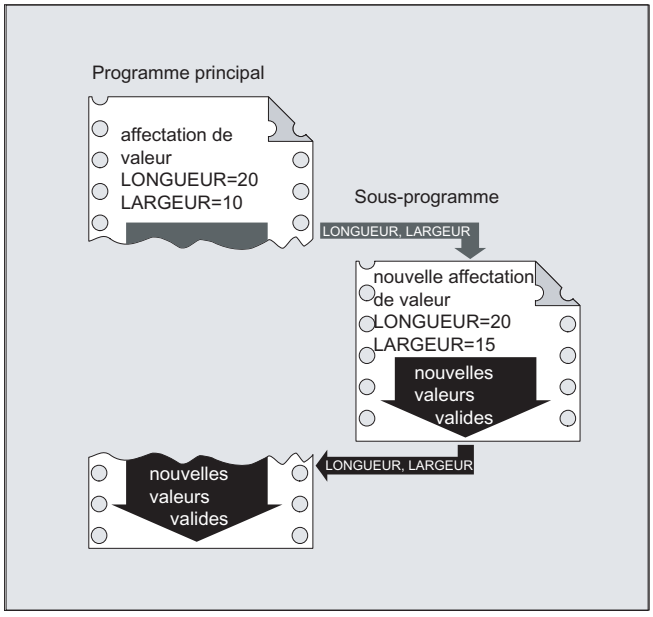
Appel d'un sous-programme avec transfert de paramètres (EXTERN) (Page 201)

2.24.2.3 Sous-programme avec transfert de paramètres Call-by-Reference (PROC, VAR)

La définition d'un sous-programme avec transfert de paramètres Call-by-Reference s'effectue avec le mot-clé `PROC`, suivi du nom de programme et d'une liste complète de tous les paramètres avec le mot-clé `VAR`, y compris le type et le nom. L'instruction de définition doit figurer dans la première ligne du programme. Des références à des tableaux peuvent également être transférées en tant que paramètres.

Call-by-Reference

Avec un transfert de paramètres Call-by-Reference, le programme appelant ne transfère pas au sous-programme la valeur d'une variable, mais une référence (pointeur) à la variable. Ainsi, le sous-programme obtient un accès direct à la variable. Ainsi, en cas de modification de la valeur de paramètre, la valeur visible dans le sous-programme est modifiée, mais aussi la valeur des variables définies dans le programme appelant. Le transfert de paramètres Call-by-Reference a donc un effet sur le programme appelant, même après la fin du sous-programme.



Remarque

Le transfert de paramètres Call-by-Reference est uniquement nécessaire si la variable transférée est définie localement dans un programme appelant (LUD). Il n'est pas nécessaire de transférer les variables globales du canal ou à la CN, car il est possible d'y accéder directement à partir du sous-programme.

Syntaxe

```
PROC <nom de programme> (VAR <type de paramètre> <nom de
paramètre>, ...)
PROC <nom de programme> (VAR <type de tableau> <nom de tableau>
[<m>,<n>,<o>], ...)
```

Remarque

Il est possible de transférer jusqu'à 127 paramètres.

Signification

`PROC` :	Instruction de définition au début d'un programme
`VAR` :	Mot-clé pour le transfert de paramètres par référence
`<nom du programme>` :	Nom du programme
`<type de paramètre>` :	Type de données du paramètre (p. ex. REAL, INT, BOOL)
`<nom de paramètre>` :	Nom du paramètre
`<type de tableau>` :	Type de données des éléments de tableau (p. ex. REAL, INT, BOOL)
`<nom de tableau>` :	Nom du tableau

[<m>,<n>,<o>] :	Dimension du tableau A l'heure actuelle, l'utilisation de tableaux à 3 dimensions est possible	
	<m> :	Taille du tableau dans la 1ère dimension
	<n> :	Taille du tableau dans la 2ème dimension
	<o> :	Taille du tableau dans la 3ème dimension

Remarque

- Le nom de programme indiqué après le mot-clé PROC doit correspondre avec le nom de programme attribué dans l'interface utilisateur.
- L'utilisation de tableaux de longueur indéterminée en tant que paramètres formels permet aux sous-programmes de traiter des tableaux de longueur variable. A cet effet, la longueur de la première dimension n'est p. ex. pas indiquée dans la définition d'un tableau bidimensionnel utilisé en tant que paramètre formel. Par contre, la virgule doit y figurer.
  Exemple : PROC <nom de programme> (VAR REAL FELD[,5])

Exemple

Définition d'un sous-programme avec deux paramètres comme référence au type REAL :

Code de programme

```
; Paramètre 1 : référence au type : REAL, nom : LONGUEUR
; Paramètre 2 : référence au type : REAL, nom : LARGEUR
PROC SUB_PROG(VAR REAL LONGUEUR, VAR REAL LARGEUR)
```

Voir aussi

Appel d'un sous-programme avec transfert de paramètres (EXTERN) (Page 201)

2.24.2.4 Sauvegarde des fonctions G modales (SAVE)

Avec l'attribut SAVE, les instructions G modales actives avant l'appel du sous-programme sont sauvegardées, puis réactivées à la fin du sous-programme.

IMPORTANT
Interruption du contournage Si un sous-programme avec l'attribut SAVE est appelé alors que le contournage est actif, le contournage est interrompu à la fin du sous-programme (saut de retour).

Syntaxe

```
PROC <non de sous-programme> SAVE
```

Signification

SAVE :	Sauvegarde des instructions G modales avant l'appel du sous-programme et restauration à la fin du sous-programme

Exemple

Dans le sous-programme CONTOUR, c'est l'instruction modale G91 (cote relative), qui s'applique. L'instruction G modale G90 (cote absolue) est active dans le programme principal. En raison de la définition de SAVE dans le sous-programme, G90 agit de nouveau dans la programme principal après la fin du sous-programme.

Définition de sous-programme :

Code de programme	Commentaire
PROC CONTOUR (VALEUR 1 REELLE) SAVE	; Définition de sous-programme avec paramètre SAVE
N10 G91 ...	; Instruction modale G G91 : Cote relative
N100 M17	; Fin de sous-programme

Programme principal :

Code de programme	Commentaire
N10 G0 X... Y... G90	; Instruction modale G G90 : Cote absolue
N20 ...	
...	
N50 CONTOUR (12.4)	; Appel de sous-programme
N60 X... Y...	; Instruction G modale G90 réactivée par SAVE

Autres conditions à prendre en compte

Frames

Dans le cas de sous-programmes avec l'attribut SAVE, le comportement des frames dépend du type de frame et se règle dans les paramètres machine.

Bibliographie

Description fonctionnelle Fonctions de base ; Axes, Systèmes de coordonnées, Frames (K2)
Chapitre : "Retour de sous-programme avec SAVE"

2.24.2.5 Inhibition de l'exécution bloc par bloc (SBLOF, SBLON)

Inhibition du bloc par bloc pour l'ensemble du programme

Les programmes identifiés par SBLOF sont exécutés en entier comme un bloc lorsque l'exécution bloc par bloc est activée, autrement dit l'exécution bloc par bloc est inhibée pour l'ensemble du programme.

SBLOF figure dans la ligne PROC et est actif jusqu'à la fin ou l'abandon du sous-programme. L'instruction de retour permet de déterminer si l'exécution doit ou non être interrompue à la fin du sous-programme :

Retour avec M17 :	Arrêt en fin de sous-programme
Retour avec RET :	Pas d'arrêt en fin de sous-programme

Inhibition du bloc par bloc à l'intérieur du programme

SBLOF doit être le seul élément du bloc. A partir de ce bloc, le mode bloc par bloc est désactivé :

- jusqu'au prochain SBLON
  ou
- jusqu'à la fin du niveau de sous-programme actif.

Syntaxe

Inhibition du bloc par bloc pour l'ensemble du programme :
```
PROC ... SBLOF
```

Inhibition du bloc par bloc à l'intérieur du programme :

```
SBLOF
...
SBLON
```

Signification

PROC :	Première instruction d'un programme
SBLOF :	Instruction de désactivation de l'exécution bloc par bloc SBLOF peut figurer dans un bloc PROC ou dans un bloc spécifique.
SBLON :	Instruction d'activation de l'exécution bloc par bloc SBLON doit figurer dans un bloc spécifique.

Autres conditions à prendre en compte

- **Inhibition du bloc par bloc et affichage de bloc**
  DISPLOF permet d'inhiber l'affichage du bloc courant dans les cycles/sous-programmes. Si DISPLOF est programmé en liaison avec SBLOF, l'appel de cycle/sous-programme est encore affiché en cas d'arrêts bloc par bloc dans le cycle/sous-programme.
- **Inhibition du bloc par bloc dans l'ASUP système ou l'ASUP utilisateur**
  Si l'arrêt bloc par bloc est inhibé dans l'ASUP système ou utilisateur par les réglages du paramètre machine PM10702 $MN_IGNORE_SINGLEBLOCK_MASK (bit0 = 1 et bit1 = 1), la programmation de SBLON permet de réactiver l'arrêt bloc par bloc dans l'ASUP.
  Si l'arrêt bloc par bloc est inhibé dans l'ASUP utilisateur par le réglage du paramètre machine PM20117 $MC_IGNORE_SINGLEBLOCK_ASUP, il n'est **pas** possible de réactiver l'arrêt bloc par bloc dans l'ASUP en programmant SBLON.
- **Particularités de l'inhibition du bloc par bloc pour les différents types d'exécution bloc par bloc**
  Lorsque l'exécution bloc par bloc SBL2 (arrêt après chaque bloc du programme pièce) est activée, il n'y a **pas** d'arrêt dans le bloc SBLON si le bit 12 est mis à "1" dans le paramètre machine PM10702 $MN_IGNORE_SINGLEBLOCK_MASK (inhibition de l'arrêt bloc par bloc).
  Lorsque l'exécution bloc par bloc SBL3 (arrêt après chaque bloc du programme pièce même dans un cycle), l'instruction SBLOF est inhibée.

Exemples

Exemple 1 : inhibition du bloc par bloc à l'intérieur d'un programme

```
Code de programme            Commentaire
N10 G1 X100 F1000
N20 SBLOF                    ; Désactivation du bloc par bloc
N30 Y20
N40 M100
N50 R10=90
N60 SBLON                    ; Réactivation du bloc par bloc
N70 M110
N80 ...
```

En mode bloc par bloc, la section de programme entre N20 et N60 est exécutée comme un pas.

Exemple 2 : cycle devant apparaître comme une instruction pour l'utilisateur

Programme principal :

```
Code de programme
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
```

```
Code de programme
N50 M30
```

CYCLE1 :

```
Code de programme                                   Commentaire
N100 PROC CYCLE1 DISPLOF SBLOF                      ; Inhibition du bloc par bloc
N110 R10=3*SIN(R20)+5
N120 IF (R11 <= 0)
N130 SETAL(61000)
N140 ENDIF
N150 G1 G91 Z=R10 F=R11
N160 M17
```

Le CYCLE1 est exécuté lorsque l'exécution bloc par bloc est active, c'est-à-dire que la touche Start doit être activée une fois pour le traitement du CYCLE1.

Exemple 3 : Un ASUP lancé par l'AP pour l'activation de décalages d'origine et de corrections d'outil modifiés ne doit pas être visible.

```
Code de programme
N100 PROC NV SBLOF DISPLOF
N110 CASE $P_UIFRNUM OF        0 GOTOF _G500
                               1 GOTOF _G54
                               2 GOTOF _G55
                               3 GOTOF _G56
                               4 GOTOF _G57
                               DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET
```

Exemple 4 : lorsque le bit 12 = 1 dans le paramètre machine PM10702, il n'y a pas d'arrêt.

Situation de départ :

- L'exécution bloc par bloc est activée.
- PM10702 $MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

Programme principal :

Code de programme	Commentaire
N10 G0 X0	; Arrêt dans cette ligne du programme pièce
N20 X10	; Arrêt dans cette ligne du programme pièce
N30 CYCLE	; Bloc de déplacement généré par le cycle.
N50 G90 X20	; Arrêt dans cette ligne du programme pièce
M30	

CYCLE :

Code de programme	Commentaire
PROC CYCLE SBLOF	; Inhibition de l'arrêt du bloc par bloc
N100 R0 = 1	
N110 SBLON	; Comme le bit 12 = 1 dans le paramètre machine PM10702, il n'y a pas d'arrêt dans cette ligne du programme pièce.
N120 X1	; Dans cette ligne du programme pièce, il y a arrêt.
N140 SBLOF	
N150 R0 = 2	
RET	

Exemple 5 : Inhibition du bloc par bloc dans le cas d'un programme imbriqué

Situation de départ :

L'exécution bloc par bloc est activée.

Imbrication de programmes :

Code de programme	Commentaire
N10 X0 F1000	; Dans ce bloc, il y a arrêt.
N20 UP1(0)	
PROC UP1(INT _NR) SBLOF	; Inhibition de l'arrêt du bloc par bloc.
N100 X10	
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	
N210 SBLON	; Activation de l'arrêt bloc par bloc.
N220 X22	; Dans ce bloc, il y a arrêt.
N230 UP3(0)	
PROC UP3(INT _NR)	
N300 SBLOF	; Inhibition de l'arrêt du bloc par bloc.
N305 X30	
N310 SBLON	; Activation de l'arrêt bloc par bloc.

Code de programme	Commentaire
N320 X32	; Dans ce bloc, il y a arrêt.
N330 SBLOF	; Inhibition de l'arrêt du bloc par bloc.
N340 X34	
N350 M17	; SBLOF est actif.
N240 X24	; Dans ce bloc, il y a arrêt. SBLON est actif.
N250 M17	; Dans ce bloc, il y a arrêt. SBLON est actif.
N120 X12	
N130 M17	; Dans ce bloc de retour, il y a arrêt. SBLOF de l'instruction PROC est actif.
N30 X0	; Dans ce bloc, il y a arrêt.
N40 M30	; Dans ce bloc, il y a arrêt.

Informations complémentaires

Blocage du bloc par bloc pour des sous-programmes asynchrones

Pour exécuter un sous-programme asynchrone bloc par bloc en une seule étape, il est nécessaire de programmer une instruction PROC avec SBLOF dans ce sous-programme asynchrone. Ceci est également valable pour la fonction "ASUP système modifiable" (MD11610 $MN_ASUP_EDITABLE).

Exemple d'ASUP système modifiable :

Code de programme	Commentaire
N10 PROC ASUP1 SBLOF DISPLOF	
N20 IF $AC_ASUP==’H200’	
N30 RET	; Pas de REPOS si changement de mode de fonctionnement.
N40 ELSE	
N50 REPOSA	; REPOS dans tous les autres cas.
N60 ENDIF	

Influence sur le programme dans le bloc par bloc

En exécution bloc par bloc, l'utilisateur peut exécuter le programme pièce bloc après bloc. Il existe les types de réglage suivants :

- SBL1 : Bloc par bloc IPO avec arrêt après chaque bloc de fonction machine.
- SBL2 : Bloc par bloc avec arrêt après chaque bloc.
- SBL3 : Interruption du cycle (l’instruction SBLOF est inhibée par la sélection de SBL3).

Inhibition du bloc par bloc dans le cas d'un programme imbriqué

Si, dans un sous-programme, SBLOF a été programmé à l’intérieur de l’instruction PROC, l’arrêt intervient lors du retour au sous-programme avec M17. L'exécution du bloc suivant dans le programme appelant est ainsi inhibée. Si une inhibition du bloc par bloc a été activée avec SBLOF, sans SBLOF dans un sous-programme dans l'instruction PROC, le programme s'arrête

après l'exécution du bloc machine suivant dans le programme appelant. Si vous ne le désirez pas, vous devez reprogrammer SBLON dans le sous-programme juste avant le retour (M17). Dans le cas d'un retour dans un programme de niveau supérieur avec RET l'exécution du programme n'est pas arrêtée.

2.24.2.6 Inhibition de l'affichage du bloc courant (DISPLOF, DISPLON, ACTBLOCNO)

Le bloc courant du programme s'affiche par défaut. Dans les cycles ou sous-programmes, l'affichage du bloc courant peut être inhibé par l'instruction DISPLOF. C'est l'appel du cycle ou du sous-programme qui s'affiche alors à la place du bloc courant. L'instruction DISPLON permet d'annuler l'inhibition de l'affichage du bloc courant.

DISPLOF ou DISPLON sont programmées dans la ligne de programme avec l'instruction PROC et s'appliquent à l'ensemble du sous-programme et, implicitement, à tous les sous-programmes appelés par celui-ci et ne contenant pas d'instruction DISPLON ou DISPLOF. Ce comportement s'applique également aux ASUP.

Syntaxe

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

Signification

DISPLOF :	Instruction d'inhibition de l'affichage du bloc courant.	
	Positionnement :	A la fin de la ligne de programme avec l'instruction PROC
	Prise d'effet :	Jusqu'au retour depuis le sous-programme ou la fin du programme.
	Remarque : si, avec l'instruction DISPLOF, vous appelez d'autres sous-programmes depuis le sous-programme, l'affichage du bloc courant y sera également inhibé, à condition que DISPLON n'y soit pas programmée de manière explicite.	
DISPLON :	Instruction d'annulation de l'inhibition de l'affichage du bloc courant	
	Positionnement :	A la fin de la ligne de programme avec l'instruction PROC
	Prise d'effet :	Jusqu'au retour depuis le sous-programme ou la fin du programme.
	Remarque : si, avec l'instruction DISPLON, vous appelez d'autres sous-programmes depuis le sous-programme, le bloc de programme courant s'y affiche également, à condition que DISPLOF n'y soit pas programmée de manière explicite.	
ACTBLOCNO :	Utilisé avec l'attribut ACTBLOCNO, DISPLOF a pour effet d'afficher le numéro du bloc courant dans lequel une éventuelle alarme est survenue. Ceci est également valable lorsque DISPLOF est programmé seul dans un niveau de programme inférieur. Lorsque DISPLOF est programmé sans ACTBLOCNO, c'est par contre le numéro de bloc de l'appel de cycle ou de sous-programme du dernier niveau de programme sans DISPLOF qui s'affiche.	

Exemples

Exemple 1 : inhibition de l'affichage du bloc courant dans le cycle

Code de programme	Commentaire
PROC CYCLE (AXIS TOMOV, REAL POSITION) SAVE DISPLOF	; Inhibition de l'affichage du bloc courant. L'appel de cycle doit s'afficher à sa place, p. ex. : CYCLE(X,100.0)
DEF REAL DIFF	; Contenu du cycle
G01 ...	
...	
RET	; Retour de sous-programme. Le bloc qui suit l'appel du cycle s'affiche.

Exemple 2 : affichage de bloc en cas d'émission d'une alarme

Sous-programme SUBPROG1 (avec ACTBLOCNO) :

Code de programme	Commentaire
PROC SUBPROG1 DISPLOF ACTBLOC- NO	
N8000 R10 = R33 + R44	
...	
N9040 R10 = 66 X100	; Déclenchement de l'alarme 12080
...	
N10000 M17	

Sous-programme SUBPROG2 (sans ACTBLOCNO) :

Code de programme	Commentaire
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; Déclenchement de l'alarme 12080
...	
N7000 M17	

Programme principal :

Code de programme	Commentaire
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; Message d'alarme = "12080 canal K1 bloc N9040 erreur de syntaxe dans texte R10="
N2060 ...	

Code de programme	Commentaire
N2350 SUBPROG2	; Message d'alarme = "12080 canal K1 bloc N2350 erreur de syntaxe dans texte R10="
...	
N3000 M30	

Exemple 3 : Annulation de l'inhibition d'affichage du bloc courant

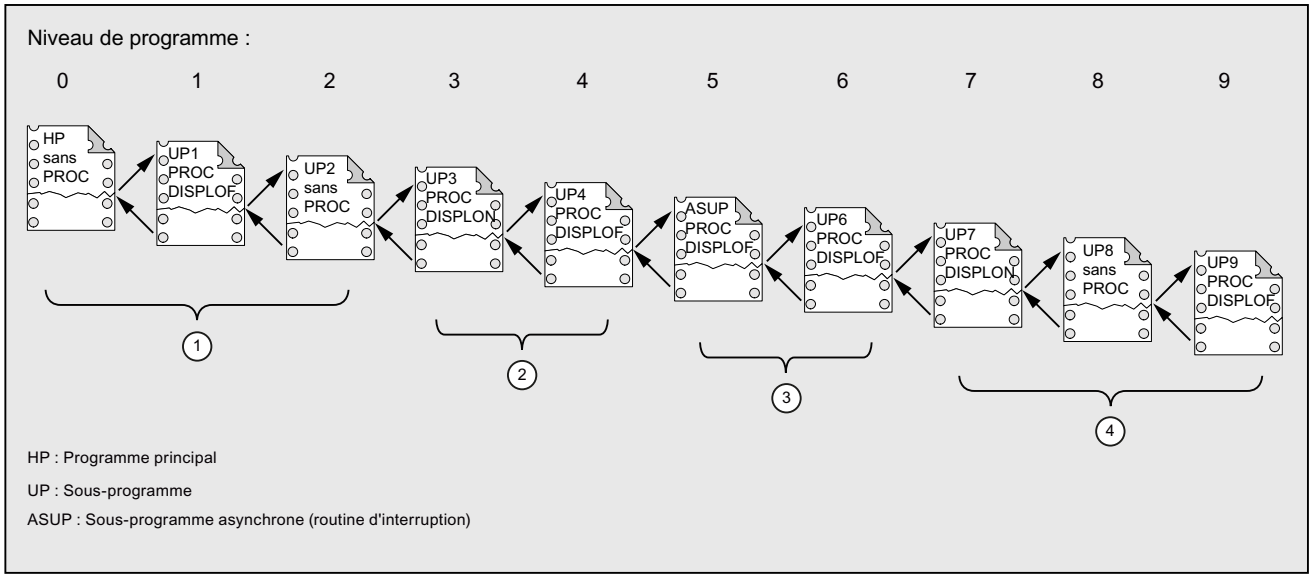
Sous-programme SUB1 avec inhibition :

Code de programme	Commentaire
PROC SUB1 DISPLOF	; Inhiber l'affichage du bloc courant dans le sous-programme SUB1. Afficher à la place le bloc avec l'appel SUB1.
...	
N300 SUB2	; Appeler le sous-programme SUB2.
...	
N500 M17	

Sous-programme SUB2 sans inhibition :

Code de programme	Commentaire
PROC SUB2 DISPLON	; Annuler l'inhibition d'affichage du bloc courant dans le sous-programme SUB2.
...	
N200 M17	; Retour au sous-programme SUB1. L'affichage du bloc courant est de nouveau inhibé dans SUB1.

Exemple 4 : comportement d'affichage pour diverses combinaisons DISPLON/DISPLOF



① L'affichage du bloc courant affiche les lignes du programme pièce du niveau 0.
② L'affichage du bloc courant affiche les lignes du programme pièce du niveau 3.
③ L'affichage du bloc courant affiche les lignes du programme pièce du niveau 3.
④ L'affichage du bloc courant affiche les lignes du programme pièce du niveau 7/8.

2.24.2.7 Identifier les sous-programmes avec prétraitement (PREPRO)

Le mot-clé `PREPRO` permet d'identifier tous les fichiers pendant le démarrage réalisé à la fin de la ligne d'instruction `PROC`.

Remarque

Ce prétraitement de programme dépend des paramètres machine configurés. Veuillez tenir compte des indications du constructeur de machine.

Bibliographie :
Description fonctionnelle Fonctions spéciales ; Conditionnement (V2)

Syntaxe

```
PROC … PREPRO
```

Signification

`PREPRO` :	Mot clé pour l'identification de tous les fichiers prétraités pendant le démarrage relatifs aux programmes se trouvant dans les répertoires cycliques

Lecture de sous-programmes avec préparation et appel de sous-programme

Les répertoires de cycle sont traités dans le même ordre, aussi bien à la mise en route de sous-programmes préparés avec des paramètres, qu'à l'appel du sous-programme :

1. _N_CUS_DIR cycles utilisateurs
2. _N_CMA_DIR cycles constructeurs
3. _N_CST_DIR cycles standard

S'il existe des programmes CN de même nom avec un ciblage différent, l'instruction `PROC` trouvée en premier est activée et l'autre instruction `PROC` est ignorée sans qu'il y ait de message d'alarme.

2.24.2.8 Retour de sous-programme M17

A la fin d'un sous-programme, on trouve l'instruction de retour `M17` (ou l'instruction de fin du programme pièce `M30`). Elle entraîne le retour dans le programme appelant sur le bloc du programme pièce suivant l'appel du sous-programme.

Remarque

`M17` et `M30` sont traités de la même manière dans le langage CN.

Syntaxe

```
PROC <nom de programme>
...
M17/M30
```

Autres conditions à prendre en compte

effet du retour du sous-programme sur le mode contournage

Si `M17` (ou `M30`) figure seul dans le bloc du programme pièce, un contournage actif dans le canal est interrompu.

Pour éviter l'interruption du contournage, `M17` (ou `M30`) doit également être inscrit dans le dernier bloc de déplacement. De plus, le paramètre machine suivant doit être mis sur "0" :

PM20800 $MC_SPF_END_TO_VDI = 0 (pas de sortie M30/M17 dans l'interface CN/AP)

Exemple

1. Sous-programme avec `M17` dans un bloc spécifique

Code de programme	Commentaire
`N10 G64 F2000 G91 X10 Y10`	
`N20 X10 Z10`	
`N30 M17`	`; Retour avec interruption du contournage.`

2. Sous-programme avec M17 dans le dernier bloc

Code de programme	Commentaire
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	; Retour sans interruption du contournage.

2.24.2.9 Retour de sous-programme RET

A la place de l'instruction de M17, il est également possible d'utiliser l'instruction RET dans le sous-programme. RET doit être programmée dans un bloc spécifique du programme pièce. Comme M17, RET entraîne le retour dans le programme appelant sur le bloc du programme pièce suivant l'appel du sous-programme.

Remarque

La programmation de paramètres permet de modifier le comportement de retour de RET (voir "Retour paramétrable dans les sous-programmes (RET ...) (Page 189)").

Application

L'instruction RET doit être utilisée lorsqu'un contournage G64 (G641 ... G645) ne doit pas être interrompu par le saut de retour.

Condition

L'instruction RET peut uniquement être utilisée dans des sous-programmes n'ayant pas été définis avec l'attribut SAVE.

Syntaxe

```
PROC <nom de programme>
...
RET
```

Exemple

Programme principal :

Code de programme	Commentaire
PROC MAIN_PROGRAM	; Début du programme
...	
N50 SUB_PROG	; Appel de sous-programme : SUB_PROG
N60 ...	
...	
N100 M30	; Fin du programme

Sous-programme :

Code de programme	Commentaire
PROC SUB_PROG	
...	
N100 RET	; Le retour s'effectue sur le bloc N60 dans le programme principal.

2.24.2.10 Retour paramétrable dans les sous-programmes (RET ...)

En général, l'instruction `RET` permet de retourner dans le programme appelant depuis un sous-programme. L'exécution reprend alors à la ligne de programme qui suit l'appel du sous-programme. Si l'exécution du programme doit reprendre à un autre endroit, les possibilités suivantes sont disponibles :

- Poursuite de l'exécution du programme après l'appel des cycles de chariotage en mode dialecte ISO (après la description du contour).
- Retour dans le programme principal à partir d'un niveau de sous-programmes quelconque (après un sous-programme asynchrone ASUP aussi) pour traiter une erreur.
- Retour sur plusieurs niveaux de programme pour des applications spéciales dans des cycles de compilation, en mode dialecte ISO.

Pour cela, l'instruction `RET` doit être programmée avec d'autres paramètres.

Sens de recherche

En spécifiant le paramètre `<bloc de destination>`, le retour s'effectue d'abord au bloc suivant le bloc d'appel. Ensuite, la recherche de la destination s'effectue en direction de la **fin** du programme auquel on retourne. Si la recherche est infructueuse, elle s'effectue alors en direction du début du programme.

Syntaxe

```
RET("<bloc de destination>")
RET("<bloc de destination>",<bloc suivant le bloc de destination>)
RET("<bloc de destination>",<bloc suivant le bloc de destination>,<n
ombre de niveaux de retour>)
RET("<bloc de destination>", ,<bloc suivant le bloc de destination>)
RET("<bloc de destination>",<bloc suivant le bloc de destination>,<n
ombre de niveaux de retour>,
<retour au début du programme>)
RET(, ,<nombre de niveaux de retour>,<retour au début du programme>)
```

Signification

`RET` :	Fin de sous-programme		
`<Bloc de destination>` :	Le bloc où l'exécution du programme doit reprendre est indiqué dans le paramètre comme destination de saut. Si le paramètre <nombre de niveaux de retour> n'est pas programmé, la destination de saut se trouve dans le programme depuis lequel le sous-programme courant a été appelé. Les indications possibles sont :		
	<Numéro de bloc>	Numéro du bloc de destination. La recherche du numéro de bloc s'effectue d'abord en direction de la fin du programme dans le programme auquel on retourne.	
	<Repère de saut>	Repère de saut qui doit être présent dans le programme auquel on retourne. La recherche du repère de saut s'effectue d'abord en direction de la fin du programme dans le programme auquel on retourne.	
	<Chaîne de caractères>	Chaîne de caractères qui doit être présente dans le programme auquel on retourne (par exemple nom de programme ou de variable). La recherche de la chaîne de caractères s'effectue d'abord en direction de la fin du programme dans le programme auquel on retourne. Les règles suivantes sont valables pour la programmation de la chaîne de caractères dans le bloc de destination : • **Espace à la fin** (contrairement au repère de saut identifié par un ":" à la fin). • Seuls un numéro de bloc et/ou un repère de saut peuvent **précéder la chaîne de caractères** (**pas d'instructions de programme**).	
`<Bloc suivant le bloc de destination>` :	Le paramètre permet d'indiquer si l'exécution du programme doit être poursuivie dans le bloc indiqué par le paramètre <bloc de destination> ou dans le bloc qui suit ce dernier.		
	Type :	INT	
	Valeur :	0	Le retour s'effectue dans le bloc indiqué par le paramètre <bloc de destination>.
		> 0	Le retour s'effectue dans le bloc qui suit celui indiqué par le paramètre <bloc de destination>.

`<Nombre de niveaux de retour>:`	Ce paramètre permet d'indiquer le nombre de niveaux de programme à remonter pour chercher le bloc de destination et reprendre l'exécution du programme.		
	Type :	INT	
	Valeur :	1	L'exécution du programme se poursuit au "niveau de programme courant - 1" (comme `RET` sans paramètre).
		2	L'exécution du programme se poursuit au "niveau de programme courant - 2", un niveau étant sauté.
		3	L'exécution du programme se poursuit au "niveau de programme courant - 3", deux niveaux étant sautés.
		...	
	Plage de valeurs :	1 ... 15	
`<Retour au début du programme>:`	Ce paramètre permet d'indiquer si le programme doit reprendre au début lors du retour dans le programme principal lorsque le **mode dialecte ISO** y est activé.		
	Type :	BOOL	
	Valeur :	1	Si le retour se fait dans le programme principal et si le **mode dialecte ISO** y est activé, le saut se fera au début du programme.

Remarque

Lorsqu'une chaîne de caractères est indiquée pour la recherche du bloc de destination d'un saut de retour de sous-programme, le système recherche toujours d'abord un repère de saut dans le programme appelant.

Pour qu'une chaîne de caractères définisse une destination de saut univoque, la chaîne de caractères ne doit pas être identique à un nom de repère de saut, le saut de retour de sous-programme étant sinon toujours effectué sur le repère de saut et non pas sur la chaîne de caractères (voir l'exemple 2).

Autres conditions à prendre en compte

Lors d'un retour sur plusieurs niveaux de programme, les instructions `SAVE` des différents niveaux sont évaluées.

Si un sous-programme modal est actif lors d'un retour sur plusieurs niveaux de programme et que l'instruction de désélection `MCALL` est programmée pour le sous-programme modal dans l'un des sous-programmes sautés, le sous-programme modal reste actif.

IMPORTANT
Erreur de programmation
Lors d'un retour sur plusieurs niveaux de programme, il relève de la seule responsabilité de l'utilisateur de veiller à ce que l'exécution reprenne avec les réglages modaux requis. Ceci peut être obtenu, par exemple, par la programmation d'un bloc principal adéquat.

Exemples

Exemple 1 : retour dans le programme principal après l'exécution d'un sous-programme asynchrone ASUP

Programmation	**Commentaire**
N10010 CALL "UP1"	; Niveau de programme 0 (programme principal)
N11000 PROC UP1	; Niveau de programme 1
N11010 CALL "UP2"	
N12000 PROC UP2	; Niveau de programme 2
...	
N19000 PROC ASUP	; Niveau de programme 3 (exécution d'un sous-programme asynchrone ASUP)
...	
N19100 RET("N10900", ,$P_STACK)	; Retour du sous-programme au programme principal
	; $P_STACK : niveau de programme actuel
N10900	; Bloc de destination dans le programme principal
N10910 MCALL	; Désactivation de l'appel modal de sous-programme
N10920 G0 G60 G40 M5	; Initialisation des autres réglages modaux

Exemple 2 : chaîne de caractères (<chaîne de caractères>) en tant qu'indication pour la recherche du bloc de destination

Programme principal :

Code de programme	**Commentaire**
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; Appel du sous-programme "subProg1"
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; Appel du sous-programme "subProg2"
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1: G1 X30	; "subProg1" est défini ici comme repère de saut.
N1560 ...	
N1600 subProg3	; Appel du sous-programme "subProg3"
N1610 ...	
N1900 M30	

Sous-programme subProg1 :

Code de programme	**Commentaire**
PROC subProg1	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; Retour au bloc N1400 du programme principal

Sous-programme subProg2 :

Code de programme	**Commentaire**
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; Retour au bloc N1500 du programme principal

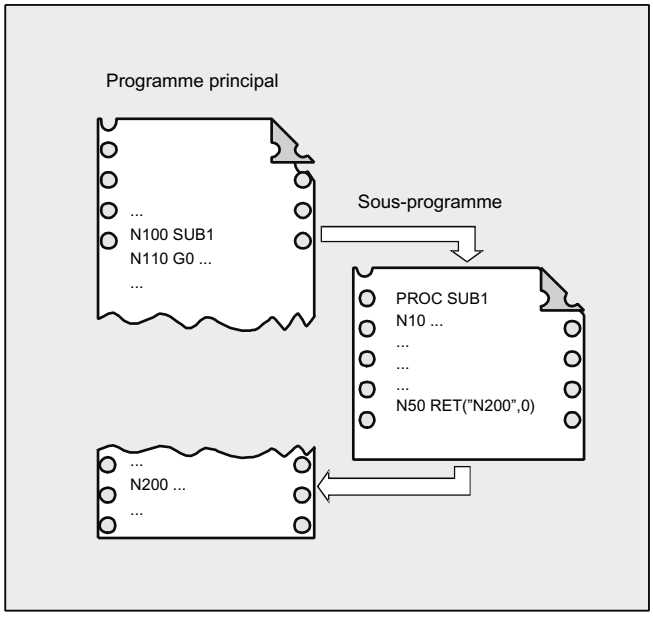
Sous-programme subProg3 :

Code de programme	**Commentaire**
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; Retour au bloc N1550 du programme principal

Informations complémentaires

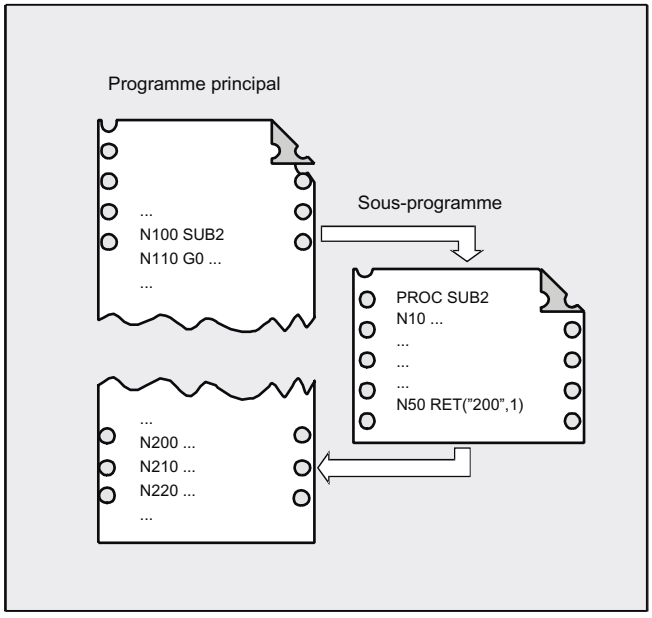
Les graphiques suivants montrent les différents effets des paramètres de renvoi.

1. <bloc de destination> = "N200", <bloc suivant le bloc de destination> = 0



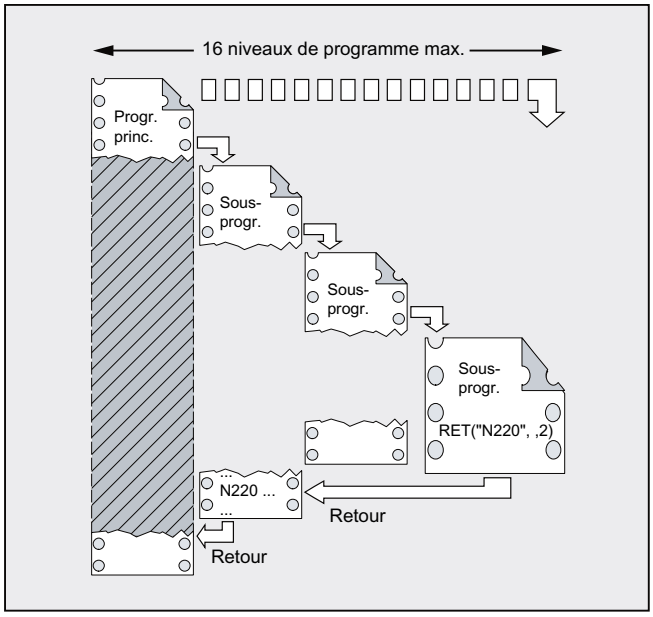
Après l'instruction `RET`, l'exécution du programme reprend au bloc `N200` du programme principal.

2. <bloc de destination> = "N200", <bloc suivant le bloc de destination> = 1



Après l'instruction `RET`, l'exécution du programme reprend au bloc (`N210`) qui suit le bloc `N200` dans programme principal.

3. <bloc de destination> = "N220", <nombre de niveaux de retour> = 2



Après l'instruction `RET`, l'exécution du programme retourne de deux niveaux de programme et reprend au bloc `N220`.

2.24.2.11 Retour paramétrable dans les sous-programmes (RETB...)

En général, l'instruction `RETB` permet de retourner dans le programme appelant depuis un sous-programme. L'exécution reprend alors à la ligne de programme qui suit l'appel du sous-programme. Si l'exécution du programme doit reprendre à un autre endroit, les possibilités suivantes sont disponibles :

- Poursuite de l'exécution du programme après l'appel des cycles de chariotage en mode dialecte ISO (après la description du contour).
- Retour dans le programme principal à partir d'un niveau de sous-programmes quelconque (après un sous-programme asynchrone ASUP aussi) pour traiter une erreur.
- Retour sur plusieurs niveaux de programme pour des applications spéciales dans des cycles de compilation, en mode dialecte ISO.

Pour cela, l'instruction `RETB` doit être programmée avec d'autres paramètres.

Sens de recherche

En spécifiant le paramètre `<bloc de destination>`, le retour s'effectue d'abord au bloc suivant le bloc d'appel. Ensuite, la recherche de la destination s'effectue en direction du **début** du programme auquel on retourne. Si la recherche est infructueuse, elle s'effectue alors en direction de la fin du programme.

Syntaxe

```
RETB("<bloc de destination>")
RETB("<bloc de destination>",<bloc suivant le bloc de destination>)
```

```
RETB("<bloc de destination>",<bloc suivant le bloc de destination>,<
nombre de niveaux de retour>)
RETB("<bloc de destination>", ,<nombre de niveaux de retour>)
RETB("<bloc de destination>",<bloc suivant le bloc de destination>,<
nombre de niveaux de retour>,
<retour au début du programme>)
RETB( , ,<nombre de niveaux de retour>,<retour au début du programme>)
```

Signification

RETB :	Fin de sous-programme		
<Bloc de destination> :	Le bloc où l'exécution du programme doit reprendre est indiqué dans le paramètre comme destination de saut. Si le paramètre <nombre de niveaux de retour> n'est pas programmé, la destination de saut se trouve dans le programme depuis lequel le sous-programme courant a été appelé. Les indications possibles sont :		
	<Numéro de bloc>	Numéro du bloc de destination. La recherche du numéro de bloc s'effectue d'abord en direction du début du programme dans le programme auquel on retourne.	
	<Repère de saut>	Repère de saut qui doit être présent dans le programme auquel on retourne. La recherche du repère de saut s'effectue d'abord en direction du début du programme dans le programme auquel on retourne.	
	<Chaîne de caractères>	Chaîne de caractères qui doit être présente dans le programme auquel on retourne (par exemple nom de programme ou de variable). La recherche de la chaîne de caractères s'effectue d'abord en direction du début du programme dans le programme auquel on retourne. Les règles suivantes sont valables pour la programmation de la chaîne de caractères dans le bloc de destination : • **Espace à la fin** (contrairement au repère de saut identifié par un ":" à la fin). • Seuls un numéro de bloc et/ou un repère de saut peuvent **précéder la chaîne de caractères** (**pas d'instructions de programme**).	
<Bloc suivant le bloc de destination> :	Le paramètre permet d'indiquer si l'exécution du programme doit être poursuivie dans le bloc indiqué par le paramètre <bloc de destination> ou dans le bloc qui suit ce dernier.		
	Type :	INT	
	Valeur :	0	Le retour s'effectue dans le bloc indiqué par le paramètre <bloc de destination>.
		> 0	Le retour s'effectue dans le bloc qui suit celui indiqué par le paramètre <bloc de destination>.

`<Nombre de niveaux de retour>:`	Ce paramètre permet d'indiquer le nombre de niveaux de programme à remonter pour chercher le bloc de destination et reprendre l'exécution du programme.		
	Type :	INT	
	Valeur :	1	L'exécution du programme se poursuit au "niveau de programme courant - 1" (comme `RET` sans paramètre).
		2	L'exécution du programme se poursuit au "niveau de programme courant - 2", un niveau étant sauté.
		3	L'exécution du programme se poursuit au "niveau de programme courant - 3", deux niveaux étant sautés.
		...	
	Plage de valeurs :	1 ... 15	
`<Retour au début du programme>:`	Ce paramètre permet d'indiquer si le programme doit reprendre au début lors du retour dans le programme principal lorsque le **mode dialecte ISO** y est activé.		
	Type :	BOOL	
	Valeur :	1	Si le retour se fait dans le programme principal et si le **mode dialecte ISO** y est activé, le saut se fera au début du programme.

Remarque

Lorsqu'une chaîne de caractères est indiquée pour la recherche du bloc de destination d'un saut de retour de sous-programme, le système recherche toujours d'abord un repère de saut dans le programme appelant.

Pour qu'une chaîne de caractères définisse une destination de saut univoque, la chaîne de caractères ne doit pas être identique à un nom de repère de saut, le saut de retour de sous-programme étant sinon toujours effectué sur le repère de saut et non pas sur la chaîne de caractères (voir l'exemple 2).

Autres conditions à prendre en compte

Lors d'un retour sur plusieurs niveaux de programme, les instructions `SAVE` des différents niveaux sont évaluées.

Si un sous-programme modal est actif lors d'un retour sur plusieurs niveaux de programme et que l'instruction de désélection `MCALL` est programmée pour le sous-programme modal dans l'un des sous-programmes sautés, le sous-programme modal reste actif.

IMPORTANT
Erreur de programmation
Lors d'un retour sur plusieurs niveaux de programme, il relève de la seule responsabilité de l'utilisateur de veiller à ce que l'exécution reprenne avec les réglages modaux requis. Ceci peut être obtenu, par exemple, par la programmation d'un bloc principal adéquat.

Exemple

Code de programme	Commentaire
EXEMPLE.MPF	
…	
N3000 START_CYC(parm1, param2, …)	
N3010 TECH_CYC1(param1, param2, …)	
N3020 TECH_CYC2(param1, param2, …)	
N3030 TECH_CYC3(param1, param2, …)	
N3040 END_CYC(param1, param2, …)	
N3040 **END_CYC**(param1, param2, …)	
N3050 …	
N4500 START_CYC(param11, param12, …)	
N4510 …	
N4590 END_CYC(param11, param12, ..)	
N5000 …	
…	
N6000 M30	

Code de programme	Commentaire
PROC END_CYC(…)	; Appel dans la ligne N3040 du programme principal
N10000 …	
N15000 if status == 1	
N15010 RETB("START_CYC")	; Retour au programme appelant EXEMPLE.MPF
	; Recherche de la chaîne de caractères "START_CYC"
	; Sens de recherche : en arrière en direction du début du programme
	; Poursuite de l'exécution du programme à la ligne N3000
N15020 endif	
N15030 if status == 0	
N15040 RET	; Retour au programme appelant EXEMPLE.MPF
	; Poursuite de l'exécution du programme à la ligne N3050
N15050 endif	
N16000 RET("START_CYC")	; Retour au programme appelant EXEMPLE.MPF
	; Recherche de la chaîne de caractères "START_CYC"
	; Sens de recherche : en avant, en direction de la fin du programme
	; Poursuite de l'exécution du programme à la ligne N4500
N17060 RETB	; Retour au programme appelant EXEMPLE.MPF
	; Poursuite de l'exécution du programme à la ligne N3050
	; RETB sans paramètre est identique à RET

2.24.3 Appel d'un sous-programme

2.24.3.1 Appel de sous-programme sans transfert de paramètres

L'appel d'un sous-programme s'effectue soit avec l'adresse L et le numéro de sous-programme, soit par indication du nom du programme.

Un programme principal peut être appelé aussi en tant que sous-programme. Dans ce cas, la fin du programme `M2` ou `M30` spécifiée dans le programme principal est traitée comme `M17` (fin du programme avec retour au programme appelant).

Remarque

Inversement, un sous-programme peut également être lancé comme programme principal.

Stratégie de recherche de la commande :

*_MPF existe-t-il ?

*_SPF existe-t-il ?

Il en résulte : si le nom du sous-programme appelant est identique au nom du programme principal, c'est le programme principal qui sera rappelé. Pour éviter cet effet qu'on ne désire généralement pas, il convient de nommer sans aucune ambiguïté les programmes principaux et les sous-programmes.

Remarque

Des sous-programmes ne nécessitant pas de transfert de paramètres peuvent également être appelés à partir d'un fichier d'initialisation.

Syntaxe

```
L<numéro>/<nom de programme>
```

Remarque

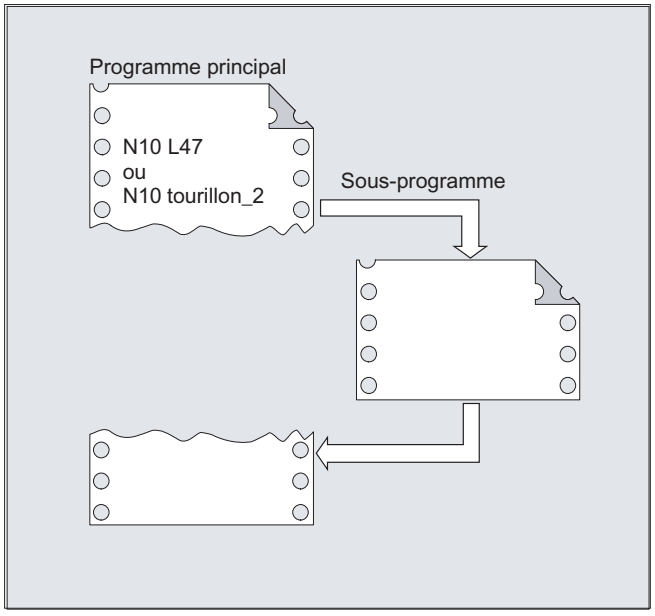
L'appel d'un sous-programme doit toujours être programmé dans le bloc CN spécifique.

Signification

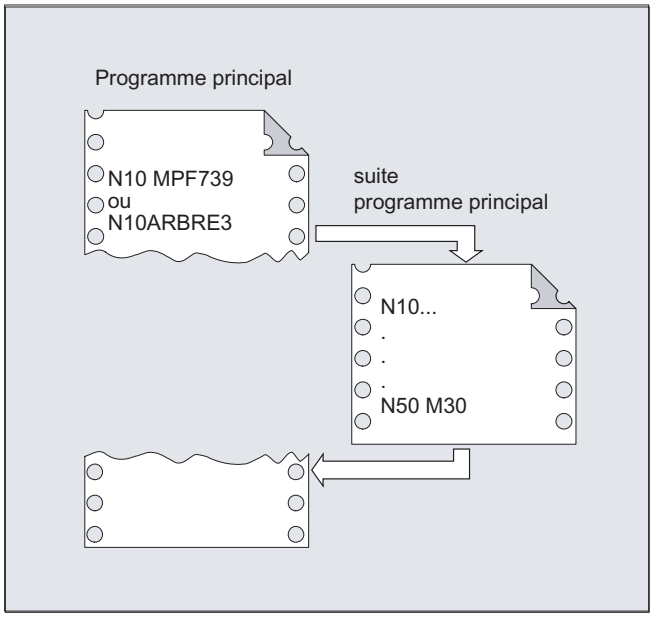
`L` :	Adresse pour l'appel du sous-programme	
`<numéro>` :	Numéro du sous-programme	
	Type :	INT
	Valeur :	Maximum 7 chiffres en notation décimale **Attention :** Les zéros en début de nom ont une signification (⇒ L123, L0123 et L00123 sont trois sous-programmes différents).
`<nom du programme>` :	Nom du sous-programme (ou programme principal)	

Exemples

Exemple 1 : appel d'un sous-programme sans transfert de paramètres



Exemple 2 : appel d'un programme principal comme sous-programme



Voir aussi

Sous-programme sans transfert de paramètres (Page 171)

2.24.3.2 Appel d'un sous-programme avec transfert de paramètres (EXTERN)

A l'appel d'un sous-programme avec transfert de paramètres, des variables ou valeurs peuvent être transférées directement (sauf pour les paramètres VAR).

Les sous-programmes avec transfert de paramètres doivent être déclarés dans le programme principal avec l'instruction EXTERN avant leur appel (par exemple en début de programme). Il faut alors indiquer le nom du sous-programme et les types de variables dans l'ordre du transfert.

IMPORTANT
Risque de confusion Les types de variables de même que l'ordre de transfert doivent correspondre aux définitions déclarées sous PROC dans le sous-programme. Les noms des paramètres peuvent être différents dans le programme principal et le sous-programme.

Syntaxe

```
EXTERN <nom de programme>(<type_Par1>,<type_Par2>,<type_Par3>)
...
<Nom de programme>(<valeur_Par1>,<valeur_Par2>,<valeur_Par3>)
```

Remarque

Un appel de sous-programme doit toujours être programmé dans un bloc CN spécifique.

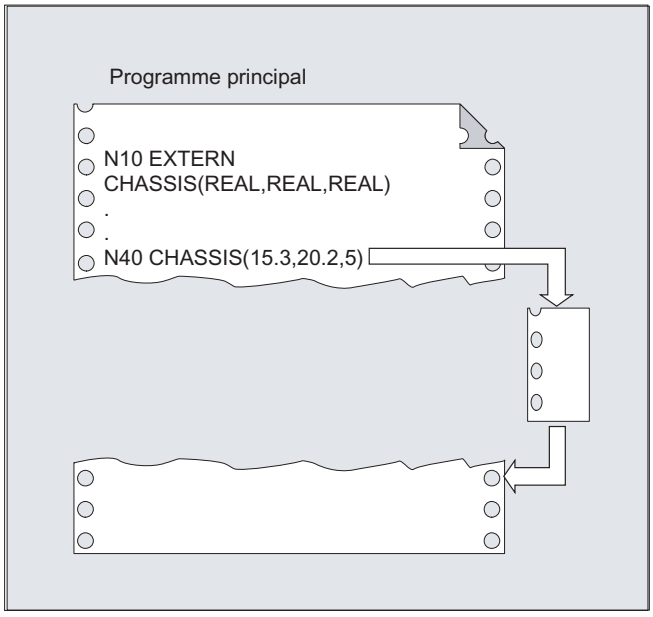
Signification

<nom du programme> :	Nom du sous-programme
EXTERN :	Mot clé pour la déclaration d'un sous-programme avec transfert de paramètres **Remarque :** L'instruction EXTERN ne doit être indiquée que si le sous-programme se trouve dans le répertoire pièce ou dans le répertoire des sous-programmes globaux. Il n'est pas nécessaire de déclarer les cycles avec l'instruction EXTERN .
<type_par1>,<type_par2>,<type_par3> :	Types de variable des paramètres à transférer, dans l'ordre du transfert
<valeur_par1>,<valeur_par2>,<valeur_par3> :	Valeurs de variable des paramètres à transférer

Exemples

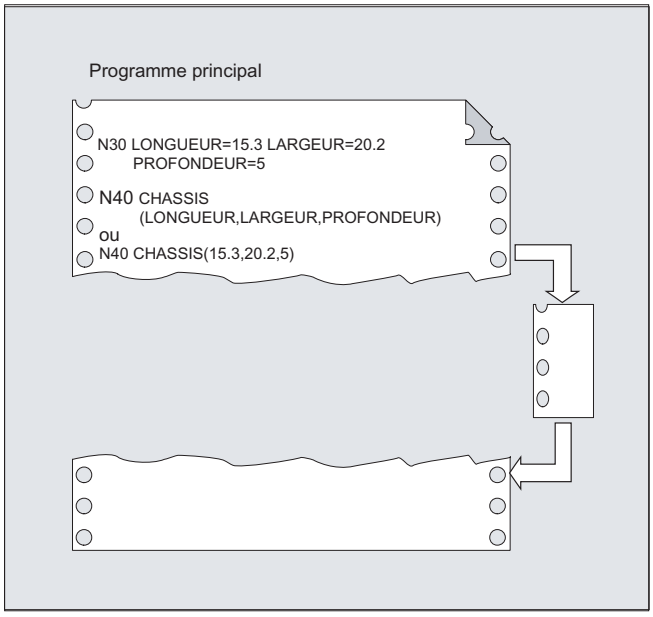
Exemple 1 : appel d'un sous-programme avec déclaration préalable

Code de programme	Commentaire
N10 EXTERN CADRE(REAL,REAL,REAL)	; Indication du sous-programme.
...	
N40 CADRE(15.3,20.2,5)	; Appel du sous-programme avec transfert de paramètres.



Exemple 2 : appel d'un sous-programme sans déclaration préalable

Code de programme	Commentaire
N10 DEF REAL LONGUEUR, LARGEUR, PROFONDEUR	
N20 …	
N30 LONGUEUR=15.3 LARGEUR=20.2 PROFONDEUR=5	
N40 CADRE(LONGUEUR,LARGEUR,PROFONDEUR)	; ou : N40 CADRE(15.3,20.2,5)



Voir aussi

Sous-programme avec transfert de paramètres Call-by-Value (PROC) (Page 172)

Sous-programme avec transfert de paramètres Call-by-Reference (PROC, VAR) (Page 174)

2.24.3.3 Nombre de répétitions de programme (P)

Si un sous-programme doit être exécuté plusieurs fois d'affilée, le nombre souhaité de répétitions peut être programmé sous l'adresse `P`, dans le bloc qui contient l'appel du sous-programme.

⚠ PRUDENCE

Appel de sous-programme avec répétition et transfert de paramètres

Les paramètres sont transférés uniquement à l'appel du sous-programme, c'est-à-dire lors de la première exécution. Ils restent inchangés lors des répétitions. Si vous désirez que les paramètres soient modifiés lors de répétitions, vous devez définir des conventions correspondantes dans le sous-programme.

Syntaxe

```
<nom du programme> P<valeur>
```

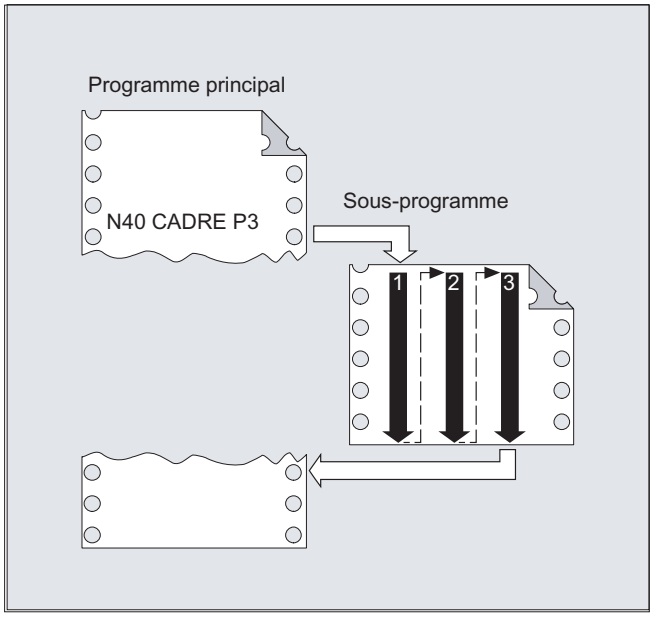
Signification

`<nom du programme>` :	appel de sous-programme
`P` :	Adresse pour la programmation des répétitions de programme

<valeur> :	Nombre de répétitions de programme	
	Type :	INT
	Plage de valeurs :	1 … 9999 (sans signe)

Exemple

Code de programme	Commentaire
...	
N40 CADRE P3	; Le sous-programme CADRE doit être exécuté trois fois d'affilée.
...	



2.24.3.4 Appel modal d'un sous-programme (MCALL)

Avec l'appel modal de sous-programme `MCALL(<nom de programme>)`, le sous-programme spécifié n'est pas immédiatement appelé. Au lieu de cela, à partir de cet instant, l'appel a lieu automatiquement dans le programme pièce après chaque bloc de déplacement comportant des déplacements avec interpolation, également sur plusieurs niveaux de programme.

Remarque

Dans une exécution de programme, c'est toujours le dernier appel modal de sous-programme `MCALL(<nom de programme>)` qui est actif, et uniquement celui-là. L'appel modal de sous-programme actuel remplace l'appel actif jusqu'à présent.

Si les paramètres sont transmis au sous-programme, le transfert de paramètres s'effectue uniquement lors de l'appel `MCALL(<nom de programme>(Par1, Par2...))`.

IMPORTANT
Appels modaux de sous-programme sans déplacement avec interpolation Dans les conditions suivantes, le sous-programme modal est appelé même sans programmation d'un déplacement avec interpolation : • Programmation des adresses `S` ou `F` lorsque `G0` ou `G1` est actif. • Lorsque `G0` ou `G1` est seul dans le bloc ou a été programmé en même temps que d'autres instructions G.

Syntaxe

```
MCALL <nom de programme>
...
MCALL
```

Signification

`MCALL <nom de programme>` :	Activation de la fonction "Appel modal de sous-programme"
`<nom du programme>` :	Nom du sous-programme
`MCALL` :	Avec `MCALL`, si aucun nom de programme n'est spécifié, la fonction "Appel modal de sous-programme" est désactivée.

Autres conditions à prendre en compte

ASUP

Si l'exécution d'un programme pièce est interrompue par un ASUP (voir le chapitre "Routine d'interruption (ASUP) (Page 134)"), aucun appel modal de sous-programme n'est exécuté dans cet ASUP.

Si un ASUP est lancé à l'état de canal "Reset", il se comporte comme un programme pièce normal en ce qui concerne les appels modaux de sous-programme.

Cycle de changement d'outil

Si la fonction "Appel modal de sous-programme" est désélectionnée dans le cycle de changement d'outil, il convient de noter que le cycle de changement d'outil est appelé implicitement par l'ASUP de recherche, éventuellement aussi après une recherche de bloc, ou manuellement par écrasement en mémoire. Dans cette situation, la fonction "Appel modal de sous-programme" ne doit pas être désélectionnée, car cela fausserait le résultat de la recherche. Par conséquent, il est recommandé de programmer la désélection de la fonction "Appel modal de sous-programme" dans le cycle de changement d'outil comme suit :

```
Code de programme        Commentaire
...
```

Code de programme	Commentaire
IF $AC_ASUP == 0	; L'appel ne s'effectue pas par l'ASUP de recherche ou un écrasement en mémoire.
MCALL	; Désactivation de la fonction "Appel modal de sous-programme".
ENDIF	
...	

Exemples

Exemple 1

Code de programme	Commentaire
N10 G0 X0 Y0	
N20 MCALL L70	; Activation de l'appel modal de sous-programme pour L70.
N30 X10 Y10	; X10 Y10 est accosté, puis L70 est appelé.
N40 X20 Y20	; X20 Y20 est accosté, puis L70 est appelé.
...	
N100 MCALL	; Désactivation de la fonction "Appel modal de sous-programme".
N110 X0 Y0	; X0 Y0 est accosté, L70 n'est **pas** appelé.

Exemple 2

Code de programme
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

Dans cet exemple, les blocs CN ci-après figurent avec des axes à interpolation programmés dans le sous-programme L80. L70 est appelé par L80.

2.24.3.5 Appel indirect de sous-programme (CALL)

Dans certaines conditions, il peut être nécessaire de faire appel à différents sous-programmes à un endroit donné du programme. Pour ce faire, indiquer le nom du sous-programme dans une variable du type STRING. L'appel du sous-programme s'effectue avec `CALL` et le nom de la variable.

Remarque

L'appel indirect de sous-programme ne s'adresse qu'à des sous-programmes sans transfert de paramètres. Pour l'appel direct d'un sous-programme, rangez le nom de ce dernier dans une constante de type STRING.

Syntaxe

```
CALL <nom_de_programme>
```

Signification

CALL :	Instruction pour l'appel indirect d'un sous-programme	
<nom du programme> :	Nom du sous-programme (variable ou constante)	
	Type :	STRING

Exemple

Appel direct avec constante de type STRING :

Code de programme	**Commentaire**
…	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PIECE1_SPF"	; Appeler directement le sous-programme PIECE1 avec CALL.
…	

Appel indirect par le biais d'une variable :

Code de programme	**Commentaire**
…	
DEF STRING[100] NOM_PROG	; Définition d'une variable.
NOM_PROG="/_N_SCP_DIR/_N_SUBPROG_WPD/_N_PIECE1_SPF"	; Affectation du sous-programme PIECE1 à la variable NOM_PROG.
CALL NOM_PROG	; Appel indirect du sous-programme PIECE1 via CALL et variable NOM_PROG.
…	

2.24.3.6 Appel indirect d'un sous-programme avec indication de la section de programme à exécuter (CALL BLOCK ... TO ...)

Avec CALL et la combinaison de mots-clés BLOCK ... TO, un sous-programme est appelé indirectement et la section de programme repérée par repères de début et de fin est exécutée.

Syntaxe

```
CALL <nom de programme> BLOCK <repère de début> TO <repère de fin>
CALL BLOCK <repère de début> TO <repère de fin>
```

Signification

CALL :	Instruction pour l'appel indirect d'un sous-programme	
<nom du programme> :	Nom du sous-programme (variable ou constante) contenant la section de programme à exécuter (**indication optionnelle**).	
	Type :	STRING
	Remarque : Si aucun <nom de programme> n'est programmé, le système recherche et exécute la section de programme repérée par le <repère de début> et le <repère de fin> dans le programme actuel.	
BLOCK ... TO ... :	Combinaison de mots-clés pour l'exécution indirecte d'une section de programme	
<repère de début> :	Variable indiquant le début de la section de programme à exécuter.	
	Type :	STRING
<repère de fin> :	Variable indiquant la fin de la section de programme à exécuter.	
	Type :	STRING

Exemple

Programme principal :

Code de programme	Commentaire
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; Définition des variables pour le repère de début et le repère de fin.
ETIQUETTE_DEBUT="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	; Appel indirect d'un sous-programme et identification de la section de programme à exécuter.
...	

Sous-programme :

Code de programme	Commentaire
PROC CONTOUR_1 ...	
LABEL_1	; Repère de début : début de l'exécution de la section de programme
N1000 G1 ...	
...	
LABEL_2	; Repère de fin : fin de l'exécution de la section de programme
...	

2.24.3.7 Appel indirect d'un programme programmé en langage ISO (ISOCALL)

ISOCALL permet d'appeler de façon indirecte un programme programmé en langage ISO. L'appel active le mode ISO réglé dans les paramètres machine. Le mode d'usinage initial est rétabli à la fin du programme. En l'absence du mode ISO dans les paramètres machine, l'appel du sous-programme s'effectuera en mode Siemens.

Pour de plus amples informations sur le mode ISO, voir :
Bibliographie :
Description fonctionnelle Dialectes ISO

Syntaxe

```
ISOCALL <nom_programme>
```

Signification

ISOCALL :	Mot clé pour l'appel indirect du sous-programme activant le mode ISO réglé dans les paramètres machine
<nom du programme> :	Nom du programme programmé dans un langage ISO (variable ou constante de type STRING)

Exemple : Appel d'un contour avec programmation de cycles depuis le mode ISO

Code de programme	**Commentaire**
0122_SPF	; Description d'un contour en mode ISO
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] NOM_PROG = "0122"	; Programme pièce (cycle) Siemens
...	
N2000 R11 = $AA_IW[X]	
N2010 ISOCALL NOM_PROG	
N2020 R10 = R10+1	; Exécution du programme 0122.spf en mode ISO
...	
N2400 M30	

2.24.3.8 Appel de sous-programme avec chemin d'accès et paramètres (PCALL)

PCALL permet d'appeler des sous-programmes avec chemin d'accès absolu et transfert de paramètres.

Syntaxe

```
PCALL <chemin/nom de programme>(<paramètre 1>, …,<paramètre n>)
```

Signification

`PCALL` :	Mot-clé pour l'appel d'un sous-programme avec chemin d'accès absolu.
`<chemin/nom du programme>` :	Chemin d'accès absolu, y compris nom du sous-programme. Pour les règles concernant le chemin d'accès, voir "Adressage des fichiers de la mémoire de programmes (Page 227)". Si aucune indication de chemin absolu n'est faite, `PCALL` se comporte comme un appel standard de sous-programme avec descripteur de programme. Le nom du programme est spécifié sans préfixe et sans extension de fichier. Si le nom du programme doit être programmé avec préfixe et extension de fichier, il convient de le déclarer de manière explicite avec préfixe et extension de fichier comme instruction `EXTERN`.
`<paramètre 1>, ...` :	Paramètres courants selon l'instruction `PROC` du sous-programme.

Exemple

Code de programme
`PCALL/_N_WKS_DIR/_N_ARBRE_WPD/ARBRE(paramètre1,paramètre2,` `…)`

2.24.3.9 Extension du chemin de recherche pour l'appel de sous-programmes (CALLPATH)

L'instruction `CALLPATH` permet d'étendre le chemin de recherche pour l'appel de sous-programmes. Ainsi il est également possible d'appeler des sous-programmes d'un répertoire pièce qui n'est pas sélectionné, sans indication complète du chemin d'accès absolu du sous-programme.

Une autre application est possible en mode de fonctionnement EES "EES sans GDIR", lorsqu'un répertoire sur une mémoire de programmes externe est utilisé pour stocker des sous-programmes globaux. Dans ce cas, l'instruction `CALLPATH` permet d'étendre le chemin de recherche à ce répertoire de sous-programmes.

L'extension du chemin de recherche s'effectue avant l'entrée des cycles utilisateur (_N_CUS_DIR).

Les événements suivants annulent de nouveau l'extension du chemin de recherche :

- `CALLPATH` avec espace
- `CALLPATH` sans paramètre
- Fin de programme pièce
- Reset

Syntaxe

```
CALLPATH("<chemin d'accès>")
```

Signification

`CALLPATH` :	Mot clé pour l'extension programmable du chemin de recherche. Programmation dans une ligne spécifique du programme pièce.
`<chemin d'accès>` :	Constante ou variable de type STRING Contient le chemin d'accès absolu du répertoire pour l'extension du chemin de recherche. Pour les règles concernant le chemin d'accès, voir "Adressage des fichiers de la mémoire de programmes (Page 227)".

Exemple

Le chemin de recherche doit être étendu à un répertoire pièce donné :

Code de programme

```
...
CALLPATH("/_N_WKS_DIR/_N_MYWPD_WPD")
...
```

Le réglage du chemin de recherche est alors le suivant (le point 5 est nouveau) :

1. Répertoire actuel/*nom*
2. Répertoire actuel/*nom_SPF*
3. Répertoire actuel/*nom_MPF*
4. //NC:/_N_SPF_DIR / *nom_SPF*
5. **/_N_WKS_DIR/_N_MYWPD_WPD/nom_SPF**
6. /N_CUS_DIR/*nom_SPF*
7. /_N_CMA_DIR/*nom_SPF*
8. /_N_CST_DIR/*nom_SPF*

Autres conditions à prendre en compte

- `CALLPATH` vérifie l'existence réelle du chemin d'accès programmé. En cas d'erreur, l'exécution du programme pièce est annulée avec l'alarme de correcteur 14009.
- `CALLPATH` peut également être programmé dans les fichiers INI. L'instruction est alors active pendant la durée d'exécution du fichier INI (fichier WPD-INI ou programme d'initialisation de données actives CN, par exemple frames dans le 1er canal _N_CH1_UFR_INI). Puis le chemin de recherche est de nouveau annulé.

2.24.3.10 Exécution de sous-programme externe (840D sl) (EXTCALL)

L'instruction `EXTCALL` permet de recharger un programme pièce à partir d'une mémoire externe et de l'exécuter.

Les mémoires externes disponibles sont les suivantes :

- Lecteur local
- Lecteur réseau
- Lecteur USB

Remarque

Seules les interfaces USB sur le pupitre opérateur ou TCU doivent être utilisées pour l'exécution d'un programme externe depuis un lecteur USB.

IMPORTANT
Endommagement de l'outil / de la pièce par une clé USB Pour exécuter un sous-programme externe, il est recommandé de ne pas utiliser de clé USB. En effet, toute coupure de communication avec la clé USB pendant l'exécution du programme pièce, résultant de mauvais contacts, d'une déconnexion ou d'un retrait accidentel pendant le fonctionnement, entraîne l'arrêt immédiat de l'usinage. Ceci pourrait endommager l'outil et/ou la pièce.

Préréglage du chemin vers le programme externe

Le chemin d'accès au répertoire de programmes externe peut être préréglé par la donnée de réglage :

SD42700 $SC_EXT_PROG_PATH

Avec le chemin et le descripteur du programme spécifié lors de l'appel `EXTCALL`, on obtient le chemin global du programme pièce à appeler.

Remarque

Paramètres

Lors de l'appel d'un programme externe, aucun paramètre ne peut être transmis à celui-ci.

Syntaxe

```
EXTCALL("<chemin/><nom du programme>")
```

Signification

EXTCALL :	Instruction d'appel d'un sous-programme externe	
''<chemin/><nom du programme>'' :	Constante/variable de type STRING	
	<chemin/> :	Chemin d'accès absolu ou relatif (**facultatif**)
	<nom du programme> :	le nom du programme est spécifié sans le préfixe "_N_". L'extension de fichier ("MPF", "SPF") peut être ajoutée au nom du programme avec le caractère "_" ou "." (**facultatif**). Exemple : "ARBRE" "ARBRE_SPF" "ARBRE.SPF"

Chemin d'accès : désignations abrégées

Les désignations abrégées ci-après sont utilisables pour le chemin d'accès :

- Lecteur local : "**LOCAL_DRIVE:**"
- Carte CF : "**CF_CARD:**"
- Lecteur USB (pupitre opérateur) : "**USB:**"

L'une ou l'autre des désignations abrégées "CF_CARD:" et "LOCAL_DRIVE:" peut être utilisée.

Exemple

Exécution depuis le lecteur local

Le programme principal "MAIN.MPF" se trouve dans la mémoire CN et est sélectionné pour être exécuté.

Sous-programme "SP_1"

Le sous-programme externe "SP_1.SPF" ou "SP_1.MPF" se trouve sur le lecteur local dans le répertoire "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

Le chemin d'accès au répertoire du programme externe doit être réglé comme suit :

SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

Remarque

Chemin d'accès pour l'appel d'un sous-programme externe :

- Sans utiliser le préréglage : "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- En utilisant le préréglage : "SP_1"

Sous-programme "SP_2"

Le sous-programme externe "SP_2.SPF" ou "SP_2.MPF" se trouve dans le répertoire WKS.DIR/WST1.WPD du lecteur USB. Le préréglage du chemin d'accès au répertoire de

programmes externe est utilisé pour le chemin vers le sous-programme "SP_1" et n'est pas repris dans le programme principal. Le chemin complet doit donc être indiqué lors de l'appel du sous-programme "SP_2".

Programme principal "MAIN"

```
Code de programme
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

Informations complémentaires

Appel EXTCALL avec chemin d'accès absolu

Si le sous-programme existe sous le chemin spécifié, il est exécuté par l'appel `EXTCALL`. Si le sous-programme n'existe pas sous le chemin spécifié, l'exécution du programme par l'appel `EXTCALL` est annulée.

Appel EXTCALL avec chemin d'accès relatif / sans chemin d'accès

Lors d'un appel `EXTCALL` avec chemin d'accès relatif ou sans chemin d'accès, les mémoires de programme présentes sont parcourues selon le schéma suivant :

1. Lorsqu'un chemin d'accès est préréglé dans SD42700 $SC_EXT_PROG_PATH, la recherche s'effectue, dans un premier temps, à partir de cet endroit à l'aide de l'indication fournie dans l'appel `EXTCALL` (nom du programme avec chemin d'accès relatif, le cas échéant). Le chemin absolu est déterminé alors par concaténation de chaînes :
   - Chemin d'accès préréglé dans SD42700 $SC_EXT_PROG_PATH
   - Séparateur "/"
   - Chemin d'accès et nom de sous-programme dans l'instruction `EXTCALL`
2. Si le sous-programme indiqué au point 1. n'a pas été trouvé, tous les répertoires de la mémoire utilisateur sont parcourus.

La recherche est terminée dès que la première occurrence du sous-programme est trouvée. Si le sous-programme reste introuvable, l'exécution du programme par l'appel `EXTCALL` est annulée.

Mémoire de chargement réglable (tampon FIFO)

L'exécution d'un sous-programme externe requiert une mémoire de chargement. La taille de la mémoire de chargement est préréglée sur 30 Ko et ne peut être modifiée que par le constructeur de machines (via PM18360 MM_EXT_PROG_BUFFER_SIZE).

Remarque

Sous-programmes avec instructions de saut

Pour les sous-programmes externes qui contiennent des instructions de saut (`GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, `REPEAT`, `IF`, `ELSE`, `ENDIF`, etc.), les destinations doivent figurer dans la mémoire de chargement.

Remarque

Programmes ShopMill/ShopTurn

Les programmes ShopMill et ShopTurn doivent être stockés dans la mémoire de chargement dans leur intégralité du fait des descriptions de contour ajoutées à la fin du fichier.

Si plusieurs sous-programmes externes sont exécutés en parallèle, une mémoire de chargement propre à chacun d'eux est nécessaire.

Reset / Fin de programme / POWER ON

Avec Reset et POWER ON, les appels de sous-programme externes sont interrompus et les mémoires de chargement correspondantes supprimées.

Un programme sélectionné pour "exécution d'un programme externe" conserve cette sélection après un Reset, une fin de programme ou un POWER ON. Le comportement ne diffère pas par rapport aux programmes sélectionnés en interne tant que la mémoire de programmes externe reste disponible.

Bibliographie

Pour plus d'informations sur l'"exécution d'un programme externe", voir :

Description fonctionnelle Fonctions de base ; GMFC, Canal, Mode de programme, Comportement au reset (K1)

2.24.3.11 Exécution d'un sous-programme externe (828D) (EXTCALL)

L'instruction `EXTCALL` permet de recharger un programme pièce à partir d'une mémoire externe et de l'exécuter.

Les mémoires externes disponibles sont les suivantes :

- Carte CF utilisateur
- Lecteur réseau
- Lecteur USB

Remarque

La seule interface utilisable pour l'exécution depuis un lecteur USB est l'interface USB du pupitre opérateur (PPU).

IMPORTANT
Endommagement de l'outil / de la pièce par une clé USB Pour exécuter un sous-programme externe, il est recommandé de ne pas utiliser de clé USB. En effet, toute coupure de communication avec la clé USB pendant l'exécution du programme pièce, résultant de mauvais contacts, d'une déconnexion ou d'un retrait accidentel pendant le fonctionnement, entraîne l'arrêt immédiat de l'usinage. Ceci pourrait endommager l'outil et/ou la pièce.

Préréglage du chemin vers le programme externe

Le chemin d'accès au répertoire de programmes externe peut être préréglé par la donnée de réglage :

SD42700 $SC_EXT_PROG_PATH

Avec le chemin et le descripteur du programme spécifié lors de l'appel `EXTCALL`, on obtient le chemin global du programme pièce à appeler.

Remarque

Paramètres

Lors de l'appel d'un programme externe, aucun paramètre ne peut être transmis à celui-ci.

Syntaxe

```
EXTCALL("<chemin/><nom du programme>")
```

Signification

EXTCALL :	Instruction d'appel d'un sous-programme externe	
''<chemin/><nom du programme>'' :	Constante/variable de type STRING	
	<chemin/> :	Chemin d'accès absolu ou relatif (**facultatif**)
	<nom du programme> :	le nom du programme est spécifié sans le préfixe "_N_". L'extension de fichier ("MPF", "SPF") peut être ajoutée au nom du programme avec le caractère "_" ou "." (**facultatif**). Exemple : "ARBRE" "ARBRE_SPF" "ARBRE.SPF"

Chemin d'accès : désignations abrégées

Les désignations abrégées ci-après sont utilisables pour le chemin d'accès :

- Carte CF utilisateur : "**CF_CARD:**"
- Lecteur USB (pupitre opérateur) : "**USB:**"

Exemple

Le programme principal "MAIN.MPF" se trouve dans la mémoire CN et est sélectionné pour être exécuté.

Sous-programme "SP_1"

Le sous-programme externe "SP_1.SPF" ou "SP_1.MPF" se trouve sur la carte CF utilisateur dans le répertoire "/WKS.DIR/WST1.WPD".

Le chemin d'accès au répertoire du programme externe doit être réglé comme suit :

SD42700 $SC_EXT_PROG_PATH = CF_CARD:WKS.DIR/WST1.WPD

Remarque

Chemin d'accès pour l'appel d'un sous-programme externe :

- Sans utiliser le préréglage : "CF_CARD:WKS.DIR/WST1.WPD/SP_1"
- En utilisant le préréglage : "SP_1"

Sous-programme "SP_2"

Le sous-programme externe "SP_2.SPF" ou "SP_2.MPF" se trouve dans le répertoire WKS.DIR/WST1.WPD du lecteur USB. Le préréglage du chemin d'accès au répertoire de programmes externe est utilisé pour le chemin vers le sous-programme "SP_1" et n'est pas repris dans le programme principal. Le chemin complet doit donc être indiqué lors de l'appel du sous-programme "SP_2".

Programme principal "MAIN"

Code de programme
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30

Informations complémentaires

Appel EXTCALL avec chemin d'accès absolu

Si le sous-programme existe sous le chemin spécifié, il est exécuté par l'appel EXTCALL. Si le sous-programme n'existe pas sous le chemin spécifié, l'exécution du programme par l'appel EXTCALL est annulée.

Appel EXTCALL avec chemin d'accès relatif / sans chemin d'accès

Lors d'un appel EXTCALL avec chemin d'accès relatif ou sans chemin d'accès, les mémoires de programme présentes sont parcourues selon le schéma suivant :

1. Lorsqu'un chemin d'accès est préréglé dans SD42700 $SC_EXT_PROG_PATH, la recherche s'effectue, dans un premier temps, à partir de cet endroit à l'aide de l'indication fournie dans l'appel EXTCALL (nom du programme avec chemin d'accès relatif, le cas échéant). Le chemin absolu est déterminé alors par concaténation de chaînes :
   - Chemin d'accès préréglé dans SD42700 $SC_EXT_PROG_PATH
   - Séparateur "/"
   - Chemin d'accès et nom de sous-programme dans l'instruction EXTCALL
2. Si le sous-programme indiqué au point 1. n'a pas été trouvé, tous les répertoires de la mémoire utilisateur sont parcourus.

La recherche est terminée dès que la première occurrence du sous-programme est trouvée. Si le sous-programme reste introuvable, l'exécution du programme par l'appel EXTCALL est annulée.

Mémoire de chargement réglable (tampon FIFO)

L'exécution d'un sous-programme externe requiert une mémoire de chargement. La taille de la mémoire de chargement est préréglée (voir PM18360 MM_EXT_PROG_BUFFER_SIZE).

Remarque

Sous-programmes avec instructions de saut

Pour les sous-programmes externes qui contiennent des instructions de saut (GOTOF, GOTOB, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDIF, etc.), les destinations doivent figurer dans la mémoire de chargement.

Remarque

Programmes ShopMill/ShopTurn

Les programmes ShopMill et ShopTurn doivent être stockés dans la mémoire de chargement dans leur intégralité du fait des descriptions de contour ajoutées à la fin du fichier.

Si plusieurs sous-programmes externes sont exécutés en parallèle, une mémoire de chargement propre à chacun d'eux est nécessaire.

Reset / Fin de programme / POWER ON

Avec Reset et POWER ON, les appels de sous-programme externes sont interrompus et les mémoires de chargement correspondantes supprimées.

Un programme sélectionné pour "exécution d'un programme externe" conserve cette sélection après un Reset, une fin de programme ou un POWER ON. Le comportement ne diffère pas par rapport aux programmes sélectionnés en interne tant que la mémoire de programmes externe reste disponible.

Bibliographie

Pour plus d'informations sur l'"exécution d'un programme externe", voir :

Description fonctionnelle Fonctions de base ; GMFC, Canal, Mode de programme, Comportement au reset (K1)

2.25 Macroprogrammation (DEFINE ... AS)

IMPORTANT
Les macro-instructions accroissent la complexité de la programmation
Les macro-instructions peuvent entraîner une modification importante du langage de programmation de la CN. L'emploi des macro-instructions exige une grande précaution.

Une macro-instruction réunit différentes instructions en une nouvelle instruction globale qui a son propre nom. En appelant la macro-instruction pendant l'exécution du programme, les instructions programmées dans la macro-instruction sont exécutées l'une après l'autre.

Les catégories suivantes de macros sont disponibles en fonction du domaine de validité, autrement dit du domaine dans lequel la définition d'une macro-instruction est active :

- Macros locales
  Les macros locales sont des macros définies au début d'un programme CN qui n'est pas le programme principal au moment de l'exécution. Elles sont déclarées à l'appel du programme CN et supprimées au reset de fin de programme ou au démarrage suivant de la commande. L'accès aux macros locales peut uniquement se faire à l'intérieur du programme CN dans lequel elles sont définies.
- Macros globales de programme
  Les macros globales de programme sont des macros définies au début d'un programme CN utilisé comme programme principal. Elles sont déclarées à l'appel du programme CN et supprimées au reset de fin de programme ou au démarrage suivant de la commande. L'accès aux macros globales de programme peut se faire dans le programme principal et dans tous les sous-programmes.

  Remarque

  Disponibilité des macros globales de programme

  Les macros globales de programme définies dans le programme principal ne sont également disponibles dans les sous-programmes que si le paramètre machine suivant est mis à 1 :

  PM11120 $MN_LUD_EXTENDED_SCOPE = 1

  Lorsque PM11120 = 0, les macros globales de programme définies dans le programme principal sont uniquement disponibles dans le programme principal.

- Macros globales
  Les macros globales sont des macros globales d'une CN ou d'un canal, définies dans un fichier de définition (fichier macro) et conservées même après un reset de fin de programme ou au démarrage suivant de la commande. Les macros globales peuvent être appelées et exécutées dans tout programme principal ou sous-programme.

  Remarque

  Pour pouvoir utiliser les macros d'un fichier de macros **externe** dans le programme CN, il faut charger ce fichier dans la CN.

Les macros doivent avoir été définies avant leur utilisation. Il convient de respecter les règles suivantes :

- Dans une macro, vous pouvez définir des descripteurs, des fonctions G, M, H et des noms de sous-programme L quelconques.
- La définition d'une macro peut se faire en début de programme ou dans un fichier de définition spécifique (fichier macro).
- Les macros locales et globales de programme sont définies en début de programme.
- Les macros globales doivent être définies dans un fichier macro, par ex. _N_DEF_DIR/ _N_UMAC_DEF.
- Les macros d'instruction G ne peuvent être définies qu'en tant que macros globales.
- Les fonctions H et L sont programmables à deux chiffres.
- Les instructions M et G sont programmables à trois chiffres.

Remarque

Les mots-clés et les noms réservés ne doivent pas être écrasés par des macros.

Syntaxe

Définition d'une macro-instruction :
```
DEFINE <Macro_Name> AS <Operation_1> <Operation_2> ...
```

Appel dans le programme CN :
```
<Macro_Name>
```

Signification

`DEFINE ... AS` :	Combinaison de mots clés pour la définition d'une macro-instruction
`<Macro_Name>`:	Nom de la macro-instruction Seuls des descripteurs sont autorisés comme noms de macro-instructions. Le nom de la macro-instruction permet de l'appeler depuis le programme CN.
`<Operation_1>`:	Première instruction de programmation dans la macro-instruction
`<Operation_2>`:	Deuxième instruction de programmation dans la macro-instruction

Autres conditions à prendre en compte

Imbrication

L'imbrication de macro-instructions n'est pas possible.

Exemples

Exemple 1 : macro-instruction définie en début de programme

Code de programme	Commentaire
DEFINE LIGNE AS G1 G94 F300	; Définition d'une macro-instruction
...	
N70 LINIE X10 Y20	; Appel de la macro-instruction
...	

Exemple 2 : définitions de macro-instruction dans un fichier de macro-instructions

Code de programme	Commentaire
DEFINE M6 AS L6	; Lors du changement d'outil, un sous-programme qui se charge du transfert de données nécessaire est appelé. La fonction M de changement d'outil (par exemple M106) proprement dite est sortie dans le sous-programme.
DEFINE G81 AS DRILL(81)	; Simulation de l'instruction DIN G.
DEFINE G33 AS M333 G333	; La synchronisation avec l'AP est requise pour le filetage. L'instruction G33 initiale a été renommée en G333 par un PM, si bien que la programmation ne change pas pour l'utilisateur.

Exemple 3 : fichier externe de macro-instructions

Après le chargement du fichier externe de macro-instructions dans la commande, il est nécessaire de charger ce fichier dans la CN. Les macro-instructions ne peuvent être utilisées dans le programme CN qu'ensuite.

Code de programme	Commentaire
%_N_UMAC_DEF	
;$PATH=/_N_DEF_DIR	; Macro-instructions personnalisées
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; Broche sens horaire, arrosage en marche
DEFINE M14 AS M4 M7	; Broche sens antihoraire, arrosage en marche
DEFINE M15 AS M5 M9	; Arrêt broche, arrêt arrosage
DEFINE M6 AS L6	; Appel du programme de changement d'outil
DEFINE G80 AS MCALL	; Désactivation du cycle de perçage
M30	

Gestion des fichiers et programmes 3

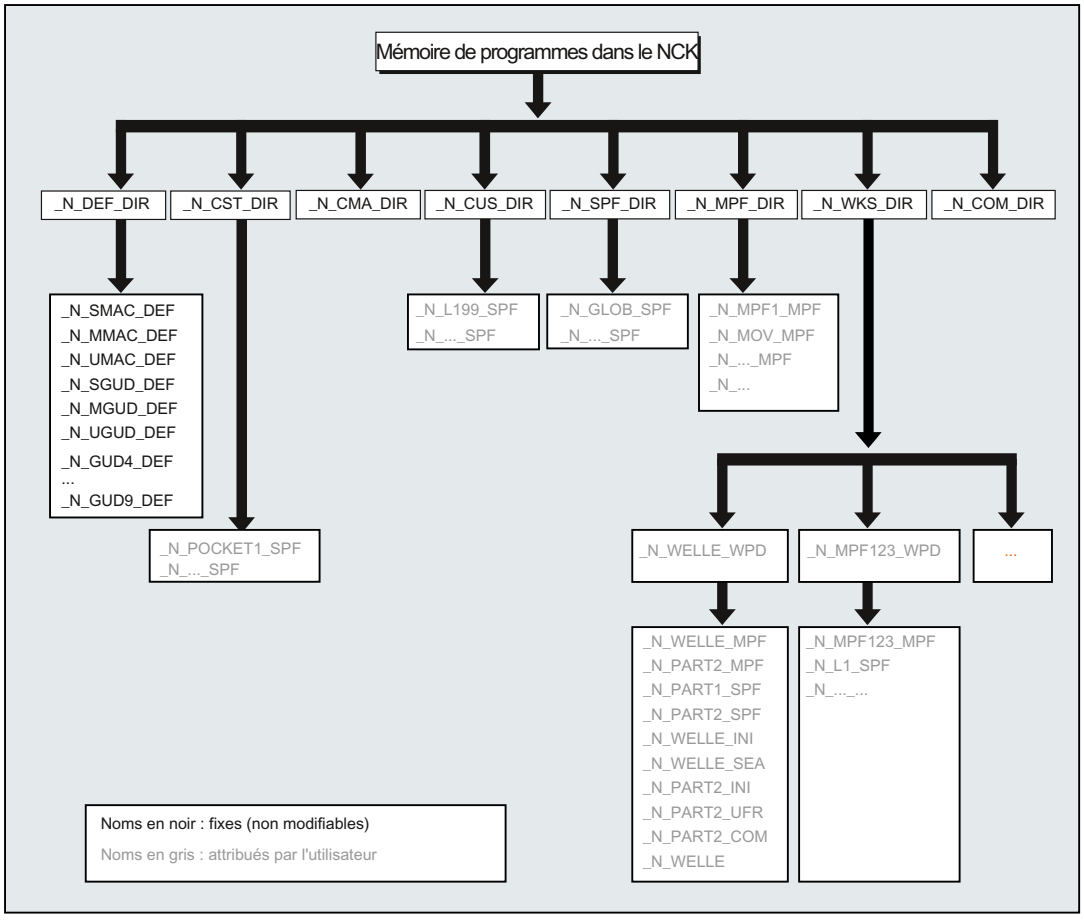
3.1 Mémoire de programmes

3.1.1 Mémoire programme CN

Les fichiers et programmes (p. ex. programmes principaux et sous-programmes, définitions de macros) sont stockés de manière permanente dans la mémoire de programme (→ système de fichiers passif).

Bibliographie :
Description fonctionnelle Fonctions d'extension ; configuration de la mémoire (S7)

Il existe en outre un certain nombre de types de fichiers qui peuvent y être stockés de façon provisoire pour être transférés sur demande (par exemple lors de la réalisation d'une certaine pièce) dans la mémoire de travail (par exemple pour des raisons d'initialisation).



Répertoires standard

Les répertoires suivants sont créés en version standard :

Répertoire	Contenu
_N_DEF_DIR	Blocs de données et blocs de macros
_N_CST_DIR	Cycles standard
_N_CMA_DIR	Cycles constructeur
_N_CUS_DIR	Cycles utilisateur
_N_WKS_DIR	Pièces
_N_SPF_DIR	Sous-programmes globaux
_N_MPF_DIR	Programmes principaux
_N_COM_DIR	Commentaires

Types de fichiers

Les types de fichiers suivants peuvent être introduits dans la mémoire de programmes :

Type de fichier	Description
<Nom>_MPF	Programme principal
<Nom>_SPF	Sous-programme
<Nom>_TEA	Paramètres machine
<Nom>_SEA	Données de réglage
<Nom>_TOA	Corrections d'outil
<Nom>_UFR	Décalages d'origine / frames
<Nom>_INI	Fichier d'initialisation
<Nom>_GUD	Données utilisateur globales
<Nom>_RPA	Paramètres R
<Nom>_COM	Commentaire
<Nom>_DEF	Définitions pour données utilisateur globales et macro-instructions

Catalogue des pièces (_N_WKS_DIR)

Le catalogue des pièces est configuré par défaut dans la mémoire de programmes sous la désignation _N_WKS_DIR. Il contient les répertoires pièce pour toutes les pièces que vous avez programmées.

Catalogues des pièces (..._WPD)

Un répertoire pièce contient tous les fichiers nécessaires pour l'usinage d'une pièce. Ceux-ci peuvent être des programmes principaux, des sous-programmes, tous programmes d'initialisation et fichiers de commentaires.

Les programmes d'initialisation sont exécutés une seule fois après la sélection du programme avec le premier lancement du programme pièce (selon le paramètre machine PM11280 $MN_WPD_INI_MODE).

Exemple :

Le répertoire pièce _N_ARBRE_WPD, qui a été créé pour la pièce ARBRE, contient les fichiers suivants :

Fichier	Description
_N_ARBRE_MPF	Programme principal
_N_PART2_MPF	Programme principal
_N_PART1_SPF	Sous-programme
_N_PART2_SPF	Sous-programme
_N_ARBRE_INI	Programme général d'initialisation des données pour la pièce
_N_ARBRE_SEA	Programme d'initialisation des données de réglage
_N_PART2_INI	Programme général d'initialisation des données pour le programme partie 2
_N_PART2_UFR	Programme d'initialisation des données frame pour le programme partie 2
_N_ARBRE_COM	Fichier de commentaires

En outre, des données qui ne sont pas directement nécessaires pour l'usinage par la CN peuvent également être stockées dans un répertoire pièce. Il peut s'agir de fichiers ASCII mais aussi de fichiers binaires comme, par exemple, des images au format JPG ou des descriptions au format PDF. Pour que la CN puisse les interpréter en tant que fichiers binaires, les extensions de fichier doivent être connues dans la CN (réglage lors de la mise en service via PM17000 $MN_EXTENSIONS_OF_BIN_FILES ; dans le réglage de base, les extensions de fichier suivantes sont prédéfinies : JPG, GIF, PNG, BMP, PDF, ICO, HTM).

Sélection d'une pièce pour exécution

Un répertoire pièce peut être sélectionné dans un canal en vue de son exécution. S'il existe dans ce répertoire un programme principal **de même nom** ou un seul programme principal (_MPF), il sera sélectionné automatiquement pour l'exécution.

Bibliographie :
Manuel d'utilisation

3.1.2 Mémoires de programmes externes

Outre le système de fichiers passif dans la CN, des mémoires de programmes externes peuvent également être présentes sur une machine (par ex. sur le lecteur local ou sur un lecteur réseau).

Les fonctions "Exécution d'un programme externe" et "EES (Execution from External Storage)" permettent d'exécuter des programmes pièce **directement** depuis des mémoires de programmes externes.

Bibliographie :
Description fonctionnelle Fonctions de base ; K1 : GMFC, Canal, Mode de programme, Comportement au reset

Mémoire globale de programmes pièce (GDIR)

Lors de la déclaration des lecteurs, l'un des lecteurs peut être configuré en tant que mémoire globale de programmes pièce (GDIR).

Bibliographie :
Manuel d'utilisation, chapitre : "Gestion des programmes" > "Configuration des lecteurs"

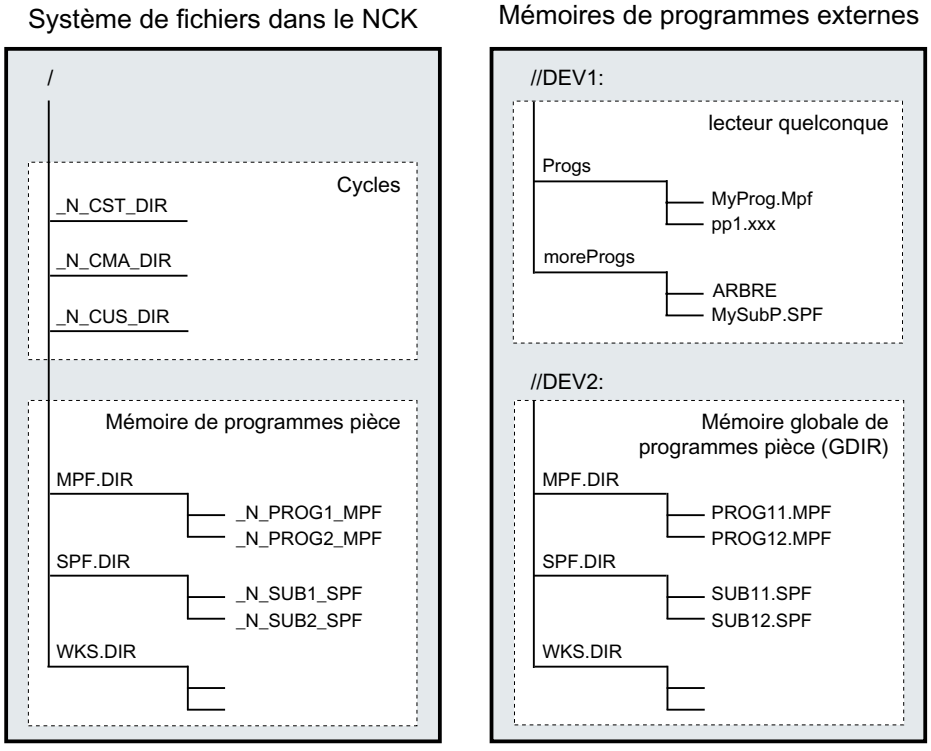
Le système crée alors automatiquement les répertoires MPF.DIR, SPF.DIR et WKS.DIR sur le lecteur. Ces trois répertoires constituent la GDIR.

La GDIR ne joue un rôle que pour la fonction EES. Selon la configuration du lecteur, la GDIR remplace ou étend la mémoire de programmes pièce CN. La configuration d'une GDIR n'est toutefois pas nécessaire pour le fonctionnement de l'EES.

Les répertoires et fichiers de la GDIR peuvent être adressés dans le programme pièce comme dans le système de fichiers passif. Il est ainsi possible d'effectuer un transfert compatible un programme CN avec chemins d'accès à partir du système de fichiers passif dans la GDIR. Le répertoire SPF.DIR de la GDIR se trouve dans le chemin de recherche des sous-programmes.

Organisation de programmes

La figure suivante met en évidence l'organisation des programmes dans les mémoires de programmes externes :



Systèmes de fichiers insensibles à la casse

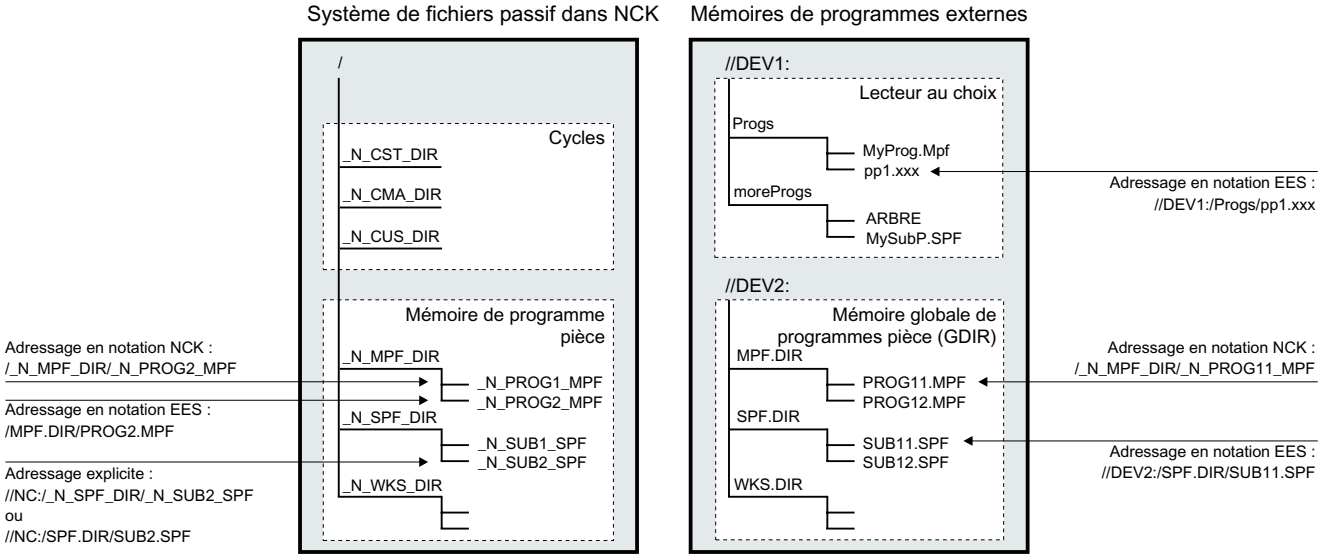
Remarque

Pour éviter les problèmes liés aux majuscules/minuscules lors de l'adressage des fichiers (voir "Adressage des fichiers de la mémoire de programmes (Page 227)"), des systèmes de fichiers **insensibles à la casse** doivent être utilisés comme mémoires de programmes externes.

3.1.3 Adressage des fichiers de la mémoire de programmes

Un fichier de la mémoire de programmes, adressé par une instruction de gestion de fichier (p. ex. WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO), est référencé soit par un chemin absolu et le nom de fichier, soit par le nom de fichier uniquement. Dans le deuxième cas, le chemin du programme sélectionné est utilisé comme chemin d'accès au fichier.

Adressage en notation CN/EES



Adressage des fichiers du système de fichiers passif

L'adressage des fichiers du système de fichiers passif s'effectue généralement en **notation CN** (les noms des répertoires et des fichiers commencent par l'identificateur de domaine "_N_", le séparateur pour l'extension de répertoire/fichier est "_") sans spécification du nom de lecteur. Un adressage en **notation EES** (sans l'identificateur de domaine "_N_", le séparateur pour l'extension de répertoire/fichier est ".") est également possible.

Exemple :

- Notation CN : "/_N_SPF_DIR/_N_SUB1_SPF"
- Notation EES : "/SPF.DIR/SUB1.SPF"

Remarque

L'adressage des fichiers du système de fichiers passif en notation EES est converti en interne en notation CN selon les règles suivantes :

- les noms de répertoires et de fichiers sont complétés par l'identificateur de domaine "_N_".
- Si le quatrième caractère en partant de la fin dans les noms de répertoires ou de fichiers est un point ("."), il est converti en tiret bas ("_").

Le nom de lecteur prédéfini "//NC:" permet également un adressage ciblé du système de fichiers passif.

Exemple :

- Notation CN : "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- Notation EES : "//NC:/SPF.DIR/SUB1.SPF"

Adressage des fichiers d'une mémoire de programmes externe

L'adressage des fichiers d'une mémoire de programmes externe non configurée en tant que GDIR doit être effectué en notation EES. Le nom de lecteur (p. ex. "//DEV1:") doit être spécifié au début du chemin d'adressage. Tous les noms symboliques de périphériques configurés dans /user/sinumerik/hmi/cfg/logdrive.ini sont autorisés.

Exemple :

- Notation EES : "//DEV1:/MyProgDir/pp1.xxx"
- Notation CN : non admissible

Adressage des fichiers de la mémoire globale de programmes pièce (GDIR)

Lors de l'adressage des fichiers de la GDIR, un chemin d'accès en notation CN est également autorisé en plus du chemin d'accès en notation EES.

Exemple :

- Notation EES : "//DEV2:/MPF.DIR/PROG11.MPF"
- Notation CN : "/_N_MPF_DIR/_N_PROG11_MPF"

Remarque

L'adressage des fichiers de la GDIR en notation CN est converti en interne en notation EES selon les règles suivantes :

- L'identificateur de domaine "_N_" est supprimé dans les noms de répertoires et de domaines.
- Si le quatrième caractère en partant de la fin dans les noms de répertoires ou de fichiers est un tiret bas ("_"), il est converti en point (".").

Règles concernant le chemin d'accès

L'indication complète d'un chemin comprend le nom du lecteur, le chemin du répertoire et le nom du fichier.

Nom du lecteur

Les règles suivantes s'appliquent pour l'indication du nom du lecteur :

- Tous les noms symboliques de périphériques configurés dans /user/sinumerik/hmi/cfg/logdrive.ini sont autorisés.
- Il commence par les caractères "//", suivi d'au moins une lettre ou un chiffre.
- Quant aux caractères qui suivent, il peut s'agir d'une combinaison quelconque de lettres, de chiffres, de traits de soulignement ("_") et d'espaces.
- Le nom se termine par une lettre ou un chiffre, suivi d'un ":".
- Aucun autre caractère spécial n'est autorisé.

Remarque

Pour le système de fichiers passif, le nom du lecteur prédéfini est "//NC:".

Exemples :

- Mémoire de programmes externe :
  - //Drive1:
  - //Drive_1:
  - //Drive 1:
  - //A B:
  - //1 B C 2:

Chemin de répertoire

Les règles suivantes s'appliquent pour l'indication du chemin de répertoire :

- Le caractère "/" se trouve au début et à la fin du chemin de répertoire, et il sert de séparateur pour les différentes parties du chemin.

  Remarque

  Un double trait oblique ("//") au sein du chemin de répertoire n'est **pas** autorisé !

- Noms de répertoires :
  - les noms de répertoires doivent commencer par une lettre ou par un chiffre. Les noms de répertoires ne commencent par l'identificateur de domaine "_N_" que dans le cas de l'adressage en notation CN.
  - Quant aux caractères qui suivent, il peut s'agir d'une combinaison quelconque de lettres, de chiffres et de "_".

    Remarque

    Les espaces sont également autorisés dans les noms de répertoires pour les mémoires de programmes externes. Cela n'est toutefois pas le cas si la mémoire de programmes externe est configurée en tant que mémoire globale de programmes pièce (GDIR).

  - Aucun autre caractère spécial n'est autorisé.
- Extensions de répertoires :
  - les extensions de répertoires doivent comprendre exactement trois lettres/chiffres.
  - Elles sont séparées du nom du répertoire par "_" (notation CN) ou "." (notation EES).

  Remarque

  Dans le système de fichiers passif, les seules extensions de répertoires sont _DIR et _WPD.

Exemples :

- Système de fichiers passif ou GDIR :
  - Notation CN : _N_WKS_DIR/_N_MYNCPROGS_WPD/...
  - Notation EES : WKS.DIR/MYPROGS.WPD/...
- Mémoire de programmes externe :
  - /abc
  - /ab_c.def
  - /ab c1.def
  - /a b c .d11
  - /abc.def/ghi.klm

Nom de fichier

Les règles suivantes s'appliquent pour le nom de fichier :

- Les noms de fichiers ne commencent par l'identificateur de domaine "_N_" que dans le cas de l'adressage en notation CN.
- Les deux caractères suivants doivent être deux lettres, ou bien un tiret bas suivi d'une lettre.

  Remarque

  Si cette condition est remplie, un programme CN peut être appelé, par simple indication du nom du programme, comme sous-programme depuis un autre programme. En revanche, si le nom du programme commence par des chiffres, l'appel en tant que sous-programme est possible uniquement avec l'instruction CALL.

- Quant aux caractères qui suivent, il peut s'agir d'une combinaison quelconque de lettres, de chiffres et de "_".
- Extension du nom de fichier :
  - l'extension du nom de fichier doit comprendre exactement trois lettres/chiffres.

    Remarque

    Pour les extensions du nom de fichier autorisées dans le système de fichiers passif, voir "Mémoire programme CN (Page 223)".

  - L'extension est séparée du nom du fichier par "_" (notation CN) ou "." (notation EES).

Exemples :

- Système de fichiers passif ou GDIR :
  - Notation CN : _N_SUB1_SPF
  - Notation EES : SUB1.SPF
- Mémoire de programmes externe :
  - Partie 1
  - _pièce1
  - Pièce_1.spf
  - Pièce1.mpf

Nom de sous-programme DIN

Les règles suivantes s'appliquent pour le nom de sous-programme DIN :

- Le premier caractère est la lettre "L".
- Les caractères suivants sont des chiffres (au moins un).
- Extension du nom de fichier :
  - L'extension du nom de fichier doit comprendre exactement trois lettres.
  - L'extension est séparée du nom du fichier par "_" (notation CN) ou "." (notation EES).

Exemples :

- L123
- L1_SPF (notation CN) ou L1.SPF (notation EES)

Longueur de chemin maximale

Pour indiquer le nom du lecteur et le chemin du répertoire, vous disposez au maximum de 128 octets, et la longueur du nom du fichier ne doit pas dépasser 31 octets. Pour l'ensemble du chemin, cela donne une longueur maximale de 159 octets.

3.1.4 Chemin de recherche pour l'appel de sous-programmes

Lors d'appels de sous-programmes sans chemin d'accès, le chemin absolu est déterminé par l'exécution d'un chemin de recherche fixe.

La recherche dans les mémoires de programmes s'effectue dans l'ordre suivant :

		Répertoire	Description
1		répertoire actuel / *nom*	Le répertoire actuel est le répertoire à partir duquel le programme principal est appelé. A savoir : • un répertoire pièce ou le répertoire standard _N_MPF_DIR dans la mémoire de programmes pièce du CN ou la mémoire globale de programmes pièce ou • un répertoire quelconque d'une mémoire de programmes externe
2		répertoire actuel / *nom_SPF*	
3		répertoire actuel / *nom_MPF*	
4	a	//NC:/_N_SPF_DIR / *nom_SPF*	Répertoire de sous-programmes dans la mémoire de programmes pièce de la CN
	b	//DEV2:/_N_SPF_DIR / name_SPF [1]	Répertoire de sous-programmes dans la mémoire globale de programmes pièce **Remarque :** Cette étape de recherche est omise lorsque aucune mémoire globale de programmes pièce n'est configurée ou que l'appel de programme a eu lieu dans la mémoire de programme pièce de la CN.
5		Extension du chemin de recherche programmée avec `CALLPATH` (voir "Extension du chemin de recherche pour l'appel de sous-programmes (CALLPATH) (Page 210)"). **Remarque :** Cette étape de recherche est omise lorsque aucun `CALLPATH` n'est programmé.	
6		/_N_CUS_DIR / *nom_SPF*	Répertoire de cycles utilisateur
7		/_N_CMA_DIR / *nom_SPF*	Répertoire de cycles constructeur
8		/_N_CST_DIR / *nom_SPF*	Répertoire de cycles standard

[1] //DEV2:" représente par exemple le lecteur sur lequel la mémoire de programme pièce globale se trouve.

Les règles suivantes s'appliquent à la recherche :

- Le chemin de recherche est parcouru pour chaque appel de sous-programme individuel, c'est-à-dire que l'emplacement du programme de niveau supérieur est sans importance.
- Les types de fichiers pris en compte varient en fonction du répertoire.
- En règle générale, aucune recherche n'est effectuée dans les répertoires de niveau inférieur, c.-à-d. imbriqués, d'un répertoire.

3.1.5 Interrogation du chemin et du nom de fichier

Les variables système suivantes, pouvant être lues dans le programme pièce, sont disponibles pour l'interrogation du chemin et du nom de fichier d'un programme CN :

Variable système	Type	Signification
$P_STACK	INT	Fournit le niveau de programme dans lequel le programme CN actuel est exécuté.
$P_PATH[<n>]	STRING	Fournit le chemin du programme CN qui est exécuté dans le niveau de programme sélectionné par l'indice de tableau <n>. Exemples : $P_PATH[0] fournit le chemin pour le programme principal, p. ex. "/_N_WKS_DIR/_N_ARBRE_WPD/". $P_PATH[$P_STACK - 1] fournit le chemin du programme appelant. Si le chemin se rapporte à un programme CN stocké dans le système de fichiers passif de la CN ou dans la mémoire globale de programmes pièce (GDIR), le chemin est fourni en notation CN. Si le chemin se rapporte à un programme CN exécuté depuis une mémoire de programmes externe autre que la mémoire globale de programmes pièce, $P_PATH fournit le chemin en notation EES.
$P_PROG[<n>]	STRING	Fournit le nom du programme CN qui est exécuté dans le niveau de programme sélectionné par l'indice de tableau <n>. Si le programme CN est stocké dans le système de fichiers passif de la CN ou dans la mémoire globale de programmes pièce, le nom du programme est fourni en notation CN. Si le programme CN est exécuté depuis un lecteur externe autre que la mémoire globale de programmes pièce, $P_PROG fournit le nom en notation EES.
$P_PROGPATH	STRING	Fournit le chemin du programme CN en cours d'exécution. L'appel de $P_PROGPATH est identique à $P_PATH[$P_STACK].

Variable système	Type	Signification	
$P_IS_EES_PATH[<n>]	BOOL	Interrogation pour savoir si le chemin fourni par $P_PATH[<n>] ou le nom de programme fourni par $P_PROG[<n>] correspond à la notation CN ou à la notation EES.	
		= FALSE	$P_PATH[<n>] et $P_PROG[<n>] fournissent la notation CN, c.-à-d. qu'un préfixe "_N_" précède chaque descripteur. Le caractère de séparation pour l'extension de fichier est "_". Exemples : • Chemin en notation CN : "/_N_WKS_DIR/_N_MYWPD_WPD/" • Nom du programme en notation CN : "_N_MYPROG_MPF" Un chemin en notation CN peut se rapporter aussi bien au système de fichiers passif dans la CN qu'à la mémoire globale de programmes pièce.
		= TRUE	$P_PATH[<n>] et $P_PROG[<n>] fournissent la notation EES, c.-à-d. qu'aucun préfixe "_N_" ne précède le descripteur. Le caractère de séparation pour l'extension de fichier est ".". Exemples : • Chemin en notation EES : "//DEV1:/WKS.DIR/MYWPD.WPD/" • Nom du programme en notation EES : "MYPROG.MPF"

<n> : l'indice <n> définit le niveau de programme depuis lequel les informations de chemin doivent être lues (plage de valeurs : 0 ... 17)

Remarque

En mode de fonctionnement EES, en dehors de la mémoire globale de programmes pièce (GDIR), les chemins d'accès sont fournis en notation EES par les variables système $P_PROG, $P_PATH et $P_PROGPATH Les programmes utilisateur qui exploitent et traitent ces chemins d'accès doivent par conséquent, pour le fonctionnement EES, être étendus de telle sorte qu'ils puissent traiter également les chemins d'accès en notation EES.

3.2 Mémoire de travail (CHANDATA, COMPLETE, INITIAL)

Fonction

La mémoire de travail contient les données système et les données utilisateur courantes qui permettent à la commande de fonctionner (système actif de fichiers). Par exemple :

- Paramètres machine actifs
- Données de correction d'outil
- Décalages d'origine
- ...

programmes d'initialisation

Les programmes d'initialisation servent à l'initialisation des données de la mémoire de travail. Les types de fichier suivants peuvent être utilisés :

Type de fichier	Description
Nom_TEA	Paramètres machine
Nom_SEA	Données de réglage
Nom_TOA	Corrections d'outils
Nom_UFR	Décalages d'origine / frames
Nom_INI	Fichier d'initialisation
Nom_GUD	Données utilisateur globales
Nom_RPA	Paramètres R

Zones de données

Vous pouvez intégrer les données dans différentes zones au sein desquelles elles doivent être valides. Ainsi, une commande peut disposer de plusieurs canaux ou, d'une manière générale, également de plusieurs axes.

La répartition est la suivante :

identificateur	Zones de données
NC	Données spécifiques à la CN
CH<n>	Données spécifiques à un canal (<n> indique le numéro de canal)
AX<n>	Données spécifiques à un axe (<n> indique le numéro de l'axe machine)
TO	Données d'outil
COMPLETE	Toutes les données

Création d'un programme d'initialisation sur PC externe

A l'aide des identificateurs des zones de données et des types de données, vous pouvez déterminer les zones à considérer comme un ensemble lors de la sauvegarde des données :

_N_AX5_TEA_INI	Paramètres machine pour l'axe 5
_N_CH2_UFR_INI	Frames du canal 2
_N_COMPLETE_TEA_INI	Tous les paramètres machine

Après la mise en service de la commande, un bloc de données garantissant un fonctionnement correct de cette dernière est présent dans la mémoire de travail.

Procédure dans le cas des commandes multicanaux (CHANDATA)

`CHANDATA(<numéro de canal>)` pour plusieurs canaux est uniquement autorisé dans le fichier _N_INITIAL_INI. C'est le fichier de mise en service avec lequel toutes les données de la commande sont initialisées.

Code de programme	**Commentaire**
`%_N_INITIAL_INI`	
`CHANDATA(1)`	
	`; Affectation des axes machine au canal 1 :`
`$MC_AXCONF_MACHAX_USED(0) = 1`	
`$MC_AXCONF_MACHAX_USED[1]=2`	
`$MC_AXCONF_MACHAX_USED[2]=3`	
`CHANDATA(2)`	
	`; Affectation des axes machine au canal 2 :`
`$MC_AXCONF_MACHAX_USED[0]=4`	
`$MC_AXCONF_MACHAX_USED[1]=5`	
`CHANDATA(1)`	
	`; Paramètres machine axiaux :`
	`; Fenêtre d'arrêt précis grossier :`
`$MA_STOP_LIMIT_COARSE[AX1]=0.2`	`; Axe 1`
`$MA_STOP_LIMIT_COARSE[AX2]=0.2`	`; Axe 2`
	`; Fenêtre d'arrêt précis fin :`
`$MA_STOP_LIMIT_FINE[AX1]=0.01`	`; Axe 1`
`$MA_STOP_LIMIT_FINE[AX1]=0.01`	`; Axe 2`

IMPORTANT

Instruction CHANDATA

Dans le programme pièce, l'instruction `CHANDATA` ne peut être utilisée que pour le canal dans lequel le programme CN est exécuté ; c'est-à-dire que l'instruction peut être utilisée pour protéger le programme CN contre l'exécution dans un canal non prévu à cet effet.

En cas d'erreur, l'exécution du programme est abandonnée.

Remarque

Les fichiers INI dans des listes de tâches ne contiennent pas d'instruction `CHANDATA`.

Sauvegarde des programmes d'initialisation (COMPLETE, INITIAL)

Les fichiers de la mémoire de travail peuvent être sauvegardés sur un PC externe, puis être rechargés depuis celui-ci.

- La sauvegarde des fichiers s'effectue avec `COMPLETE`.
- `INITIAL` génère un fichier INI (_N_INITIAL_INI) pour toutes les zones.

Chargement de programmes d'initialisation

IMPORTANT
Perte de données Après le chargement du fichier "INITIAL_INI", tous les paramètres qui ne sont pas définis dans le fichier sont initialisés avec des paramètres par défaut, excepté les paramètres machine. Les **données de réglage, données d'outil, décalages d'origine, valeurs GUD, ...** sont donc réglés à des valeurs par défaut (normalement "ZERO").

Le fichier COMPLETE_TEA_INI convient par exemple au chargement de paramètres machine individuels. Dans ce fichier, la commande n'attend que des paramètres machine. Les autres zones de données restent donc inchangées dans ce cas.

Chargement de programmes d'initialisation

Les programmes INI peuvent également être sélectionnés et appelés comme programme pièce s'ils utilisent uniquement des données d'un canal. Il est ainsi également possible d'initialiser des données commandées par programme.

Zones de protection 4

4.1 Définition des zones de protection (CPROTDEF, NPROTDEF)

Les zones de protection qui doivent protéger des éléments de machine des collisions sont définies en blocs dans le programme pièce. Ceux-ci contiennent les éléments suivants :

1. Définition du plan de travail
   Avant de définir la zone de protection à proprement parler, il convient de sélectionner le plan de travail auquel doit se rapporter la description de contour de la zone de protection.
2. Début de la définition
   Selon l'instruction CN, c'est soit une zone de protection spécifique au canal, soit une zone de protection spécifique à la machine qui est créée.
3. Description du contour de la zone de protection
   Le contour d'une zone de protection est décrit à l'aide de mouvements de déplacement. Ceux-ci ne sont pas exécutés et n'ont aucun lien avec les descriptions géométriques précédentes ou suivantes. Ils définissent exclusivement la zone de protection.
4. Fin de la définition

Syntaxe

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/... X/Y/Z...
...
EXECUTE(<Var>)
```

Signification

`DEF INT <Var>` :	Définition d'une variable auxiliaire locale du type de données INTEGER
`<Var>` :	Nom de la variable auxiliaire
`G17/G18/G19` :	Plan de travail **Remarque :** Le plan de travail ne doit pas être modifié avant la fin de la définition. La programmation de la cote entre le début et la fin de la définition n'est pas admissible.
`CPROTDEF()` :	Procédure prédéfinie pour la définition d'une zone de protection spécifique au **canal**
`NPROTDEF()` :	Procédure prédéfinie pour la définition d'une zone de protection spécifique à la **machine**

<n> :	Numéro de la zone de protection définie		
	Type de données :	INT	
<t> :	Type de la zone de protection		
	Type de données :	BOOL	
	Valeur :	TRUE	zone de protection orientée **outil**
		FALSE	zone de protection orientée **pièce**
<AppLim> :	Type de limitation dans la 3e dimension		
	Type de données :	INT	
	Valeur :	0	aucune limitation
		1	limitation dans le sens plus
		2	limitation dans le sens moins
		3	limitation dans le sens positif et négatif
<AppPlus> :	Valeur de la limitation dans le sens positif de la 3e dimension		
	Type de données :	REAL	
<AppMinus> :	Valeur de la limitation dans le sens négatif de la 3e dimension		
	Type de données :	REAL	

`G0/G1/... X/Y/Z... ...:`	On peut décrire le contour d'une zone de protection avec 11 déplacements au maximum dans le plan de travail sélectionné. Le premier déplacement est constitué par l'accostage du contour. Le dernier point et le premier point de la description du contour doivent toujours coïncider. La zone se trouvant à gauche du contour correspond à la zone de protection. • Zone de protection interne Le contour pour une zone de protection interne doit être décrit dans le sens antihoraire. • Zone de protection externe (seulement admissible pour les zones de protection orientées pièce !) Le contour pour une zone de protection externe doit être décrit dans le sens horaire. Les éléments de contour suivants sont admissibles : • `G0` et `G1` pour les parties rectilignes du contour • `G2` pour les arcs de cercle dans le sens horaire Seulement admissible pour les zones de protection orientées pièce ! Non admissible pour les zones de protection orientées pièce, compte tenu qu'elles doivent être que convexes. • `G3` pour les arcs de cercle dans le sens antihoraire **Remarque :** Une zone de protection ne peut pas être décrite par un cercle complet. Un cercle complet doit être segmenté en deux cercles partiels. **Remarque :** La séquence `G2` → `G3` ou `G3` → `G2` n'est pas admissible ! Un bloc `G1` court doit être inséré entre les deux blocs à interpolation circulaire.
`EXECUTE(<Var>)` :	Procédure prédéfinie qui marque la fin de la définition. `EXECUTE` permet de revenir à l'exécution normale du programme.

Exemple

Voir exemple sous "Activation/désactivation des zones de protection (CPROT, NPROT) (Page 244)".

informations complémentaires

Zones de protection spécifiques à la machine

Une zone de protection spécifique à la machine (soit son contour) est définie au moyen des axes géométriques, c'est-à-dire par rapport au système de coordonnées de base (SCB) d'un canal. Pour assurer une surveillance correcte de la zone de protection dans tous les canaux

où la zone de protection spécifique à la machine est active, le système de coordonnées de base (SCB) de tous les canaux concernés doit être identique :

- Position de l'origine système de coordonnées par rapport à l'origine machine
- Orientation de l'axe de coordonnées

Point de référence de la description du contour

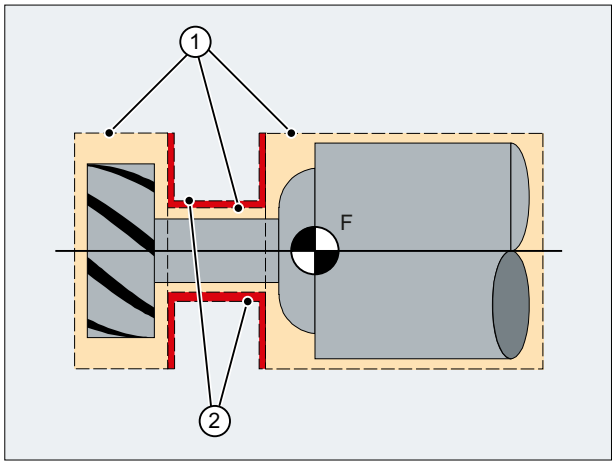
- Zones de protection orientées outil
  Les coordonnées pour les zones de protection **orientées outil** doivent être spécifiées sous forme de coordonnées absolues, rapportées au **point de référence F du porte-outil**.
- Zones de protection orientées pièce
  Les coordonnées pour les zones de protection **orientées pièce** doivent être spécifiées sous forme de coordonnées absolues, rapportées à l'origine du **système de coordonnées de base (SCB)**.

Zones de protection à symétrie de révolution

Dans le cas de zones de protection à symétrie de révolution (par ex. pour un mandrin), l'ensemble du contour doit être décrit, et pas seulement le contour jusqu'au centre de rotation.

Zones de protection orientées outil

Les zones de protection orientées outil doivent toujours être convexes. S'il faut une zone de protection concave, il faut la fractionner en plusieurs zones de protection convexes.



① Zones de protection convexes
② Zones de protection concaves (**non admissibles !**)
F Point de référence du porte-outil

Autres conditions

Pendant la définition d'une zone de protection, les fonctions suivantes ne doivent pas être activés ou utilisées :

- Correction du rayon d'outil (correction du rayon de fraise, correction du rayon du tranchant)
- Transformation
- Accostage de point de référence (`G74`)

- Accostage de point fixe (G75)
- Arrêt temporisé (G4)
- Arrêt de recherche de blocs (STOPRE)
- Fin du programme (M17, M30)
- Fonctions M : M0, M1, M2

4.2 Activation/désactivation des zones de protection (CPROT, NPROT)

Les zones de protection préalablement définies dans le programme pièce peuvent être activées à tout moment ou préactivées en vue d'une activation ultérieure par le programme utilisateur de l'AP. Les zones de protection actives peuvent être désactivées à tout moment.

Lors de l'activation ou de la pré-activation, il est possible en outre de décaler le point de référence de la zone de protection de manière relative.

Remarque

Une zone de protection n'est prise en compte qu'après le référencement de tous les axes géométriques du canal dans lequel elle a été activée.

Remarque

Surveillance des zones de protection

Si aucune zone de protection orientée outil n'est active, la trajectoire de l'outil fera l'objet d'une surveillance rapportée aux zones de protection orientées pièce.

Si aucune zone de protection orientée pièce n'est active, il n'y aura aucune surveillance de la zone de protection.

Syntaxe

```
CPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
NPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
```

Signification

`CPROT` :	Procédure prédéfinie pour l'activation d'une zone de protection spécifique à un **canal**		
`NPROT` :	Procédure prédéfinie pour l'activation d'une zone de protection spécifique à une **machine**		
`<n>` :	Numéro de la zone de protection		
	Type de données :	INT	
`<Status>` :	Ce paramètre permet de définir l'état d'activation spécifique au canal		
	Type de données :	INT	
	Valeur :	`0`	Désactivation de la zone de protection
		`1`	Préactivation de la zone de protection
		`2`	Activation de la zone de protection
		`3`	Préactivation de la zone de protection avec arrêt conditionnel

<XMov>,<YMov>,<ZMov> :	Valeurs de décalage additives dans le sens X/Y/Z Le décalage peut être réalisé dans 1, 2 ou 3 dimensions. Les valeurs de décalage se rapportent : • à l'origine machine dans le cas d'une zone de protection orientée **pièce** • au point de référence F du porte-outil dans le cas d'une zone de protection orientée **outil**.	
	Type de données :	REAL

Exemple

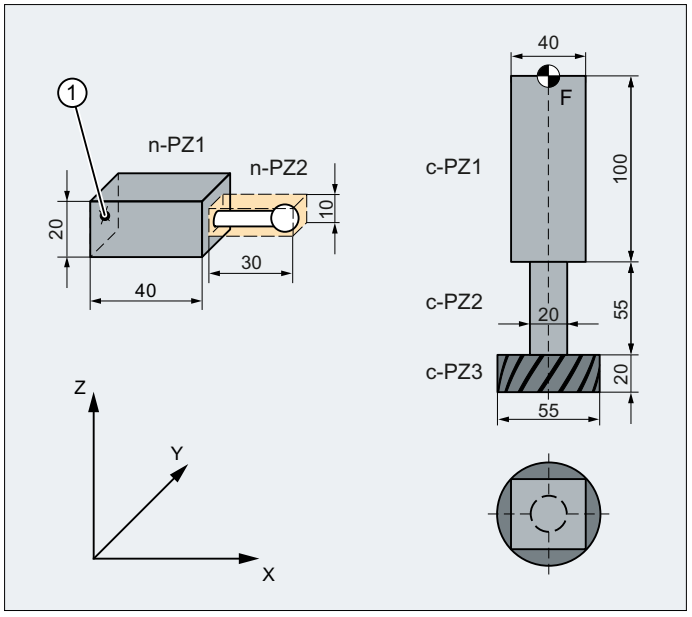
Sur une fraiseuse, il convient d'éviter une collision entre la fraise et le palpeur de mesure. La position du palpeur doit être indiquée par un décalage lors de l'activation.

A cet effet, on définit les zones de protection suivantes :

- Une zone de protection spécifique à la machine et orientée pièce pour le support du palpeur (n-PZ1) et pour le palpeur lui-même (n-PZ2).
- Une zone de protection spécifique au canal et orientée outil pour l'organe porte-fraise (c-PZ1), la queue de la fraise (c-PZ2) et la fraise elle-même (c-PZ3).

Toutes les zones de protection sont orientées en Z.

Le point de référence du palpeur de mesure doit se trouver en X = -120, Y = 60 et Z = 80 lors de l'activation.



① Point de référence pour la zone de protection du palpeur

F Point de référence du porte-outil

Code de programme	Commentaire
DEF INT PROTZONE	; Définition d'une variable auxiliaire

Code de programme	**Commentaire**
G17	; Plan de travail XY
; Définition des zones de protection :	
NPROTDEF(1,FALSE,3,10,-10)	; Zone de protection n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE(PROTZONE)	
NPROTDEF(2,FALSE,3,5,-5)	; Zone de protection n-PZ2
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE(PROTZONE)	
CPROTDEF(1,TRUE,3,0,-100)	; Zone de protection c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	
Y-20	
EXECUTE(PROTZONE)	
CPROTDEF(2,TRUE,3,-100,-150)	; Zone de protection c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE(PROTZONE)	
CPROTDEF(3,TRUE,3,-150,-170)	; Zone de protection c-PZ3
G01 X0 Y-27,5	
G03 X0 Y27,5 J27,5	
X0 Y27,5 J-27,5	
EXECUTE(PROTZONE)	
; Activation des zones de protection :	
NPROT(1,2,-120,60,80)	; Activer la zone de protection n-PZ1 avec décalage
NPROT(2,2,-120,60,80)	; Activer la zone de protection n-PZ2 avec décalage
CPROT(1,2,0,0,0)	; Activer la zone de protection c-PZ1
CPROT(2,2,0,0,0)	; Activer la zone de protection c-PZ2
CPROT(3,2,0,0,0)	; Activer la zone de protection c-PZ3

Informations complémentaires

État d'activation après démarrage de la commande

Une zone de protection peut être active tout de suite après le démarrage de la commande et le référencement des axes. C'est le cas lorsque la variable système suivante est définie sur TRUE pour la zone de protection :

- $SN_PA_ACTIV_IMMED[<n>] (pour une zone de protection spécifique à la machine) ou
- $SC_PA_ACTIV_IMMED[<n>] (pour une zone de protection spécifique au canal)
  L'indice "<n>" correspond au numéro de la zone de protection : 0 = 1re zone de protection

La zone de protection est activée avec l'état = 2 et sans décalage.

Activation multiple d'une zone de protection

Une zone de protection spécifique à la machine peut être activée dans plusieurs canaux à la fois (par ex. : la zone de protection d'un fourreau de contre-poupée quand deux chariots se font face). La surveillance des zones de protection se fait uniquement si la prise de référence a été effectuée dans tous les axes géométriques.

Une zone de protection ne peut pas être activée simultanément avec différents décalages dans un même canal.

Surveillance de zone de protection avec correction du rayon d'outil active

Lorsque la correction du rayon d'outil est active, une surveillance de zone de protection opérationnelle n'est possible que si le plan de la correction du rayon d'outil est identique au plan des définitions de zones de protection.

4.3 Contrôle des dépassements de zone de protection, des limitations de la zone de travail et des fins de course logicielles (CALCPOSI)

Fonction

La fonction CALCPOSI vérifie dans le système de coordonnées pièce (SCP) si, en partant de la position de départ, les **axes géométriques** peuvent parcourir une trajectoire donnée sans dépasser les limitations actives. Dans le cas où le déplacement ne pourrait pas être exécuté sur l'ensemble de la distance à parcourir en raison de limitations, une valeur d'état positive codée en décimal et la distance maximale à parcourir possible sont retournées.

Définition

```
INT CALCPOSI(VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

Syntaxe

```
<Status> = CALCPOSI(VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

Signification

`CALCPOSI(...)` :	Fonction prédéfinie pour le test portant sur les dépassements de limitations par rapport aux axes géométriques	
	Arrêt du prétraitement des blocs :	non
	Uniquement dans le bloc :	oui

<Status> : (Partie 1)	Valeur de retour de la fonction. Les valeurs négatives indiquent des états d'erreur.		
	Type de données :		INT
	Plage de valeurs :		-8 ≤ x ≤ 100 000
	Valeur :	0	Le déplacement peut être exécuté sur l'ensemble de la distance à parcourir
		-1	Au moins une composante est négative dans <Limit>.
		-2	Erreur lors d'un calcul de transformation. Exemple : La distance à parcourir comprend un point singulier sur lequel les positions d'axe ne sont pas définies.
		-3	La distance à parcourir spécifiée <Dist> et la distance à parcourir maximale possible <MaxDist> sont linéairement dépendantes. **Remarque** Ne peut se produire qu'en relation avec <TestLim>, bit 4 == 1.
		-4	La projection du sens de déplacement contenu dans <Dist> sur la surface de délimitation est un vecteur nul, autrement dit le sens de déplacement est perpendiculaire à la surface de limitation dépassée. **Remarque** Ne peut se produire qu'en relation avec <TestLim>, bit 5 == 1.
		-5	Dans <TestLim>, bit 4 == 1 ET bit 5 == 1
		-6	Au moins un axe machine, qui doit être pris en compte pour la vérification des limites de déplacement, n'est pas référencé.
		-7	Fonction de prévention des collisions : Définition non valide de la chaîne cinématique ou des zones de protection.
		-8	Fonction de prévention des collisions : La fonction ne peut pas être exécutée à cause d'une capacité de mémoire insuffisante.

<Status> : (Partie 2)	Position des unités **Remarque** Si plusieurs limites sont dépassées simultanément, celle entraînant la plus forte restriction de la distance à parcourir spécifiée est signalée.		
	Valeur :	1	Distance à parcourir restreinte par des fins de course logicielles
		2	Distance à parcourir restreinte par une limitation de la zone de travail
		3	Distance à parcourir restreinte par des zones de protection
		4	Fonction de prévention des collisions : Distance à parcourir restreinte par des zones de protection
	Position des dizaines		
	Valeur :	1x	Limite dépassée par la valeur initiale
		2x	Limite dépassée par la droite spécifiée La valeur de retour correspond également à cette valeur lorsque le point final en soi ne dépasse aucune limite, mais que le déplacement du point de départ au point final provoquerait le dépassement d'une limite (par exemple la traversée d'une zone de protection ou des fins de course logicielles courbés dans le SCP dans le cas de transformations non linéaires telles que Transmit).
<Status> : (Partie 3)	Position des centaines		
	Valeur :	1xx	ET position des unités == 1 ou 2 : Dépassement de la valeur limite positive.
			ET position des unités == 3[1] : Dépassement d'une zone de protection spécifique au canal.
		2xx	ET position des unités == 1 ou 2 : Dépassement de la valeur limite négative.
			ET position des unités == 3[1] : Violation d'une zone de protection spécifique au canal.
<Status> : (Partie 4)	Position des milliers		
	Valeur :	1xxx	ET position des unités == 1 ou 2 : Facteur multipliant le numéro de l'axe qui dépasse la limite. Le comptage des axes commence par 1. Référence : • Fin de course logiciel : axes machine • Limitation de la zone de travail : axes géométriques ET position des unités == 3[1] : Facteur multipliant le numéro de la zone de protection dépassée.

<Status> : (Partie 5)	Position des centaines de milliers		
	Valeur :	0xxxxx	Position des centaines de milliers == 0 : <Dist> reste inchangé
		1xxxxx	Un vecteur de direction définissant le sens de la poursuite du déplacement sur la surface de limitation est retourné dans <Dist>. Ne peut se produire qu'avec les autres conditions suivantes : • Dépassement de fin de course logiciel ou de limitation de la zone de travail (pas au point de départ) • Une transformation n'est **pas** active • <TestID>, bit 4 ou bit 5 == 1
<Start> :	Référence à un vecteur avec les positions de départ : • <Start> [0] : 1. axe géométrique • <Start> [1] : 2. axe géométrique • <Start> [2] : 3. axe géométrique		
	Type de paramètre :		Entrée
	Type de données :		VAR REAL[3]
	Plage de valeurs :		Valeur Valeur REAL ≤ x[<n>] ≤ +max. REAL +max.
<Dist> :	Référence à un vecteur.		
	Entrée : distance à parcourir relative • <Dist> [0] : 1. axe géométrique • <Dist> [1] : 2. axe géométrique • <Dist> [2] : 3. axe géométrique		
	Sortie (uniquement avec position des centaines de milliers activée dans <Status>) :		
		<Dist> contient un vecteur unité **v** en tant que valeur de sortie, qui définit la poursuite du sens de déplacement dans le SCP.	
		Cas 1 : formation du vecteur **v** pour <TestID>, bit 4 == 1 Les vecteurs d'entrée <Dist> et <MaxDist> définissent le plan de déplacement. Ce plan est coupé par la surface de délimitation dépassée. La droite d'intersection des deux plans définit la direction du vecteur **v**. L'orientation (le signe) est alors choisie de telle sorte que l'angle entre le vecteur d'entrée <MaxDist> et **v** ne dépasse pas 90 degrés.	
		Cas 2 : formation du vecteur **v** pour <TestID>, bit 5 == 1 Le vecteur **v** est le vecteur unité en direction de la projection du vecteur de déplacement contenu dans <Dist> sur la surface de délimitation. Une erreur est retournée si la projection du vecteur de déplacement sur la surface de délimitation est un vecteur nul.	
	Type de paramètre :		Entrée/sortie
	Type de données :		VAR REAL[3]
	Plage de valeurs :		Valeur Valeur REAL ≤ x[<n>] ≤ +max. REAL +max.

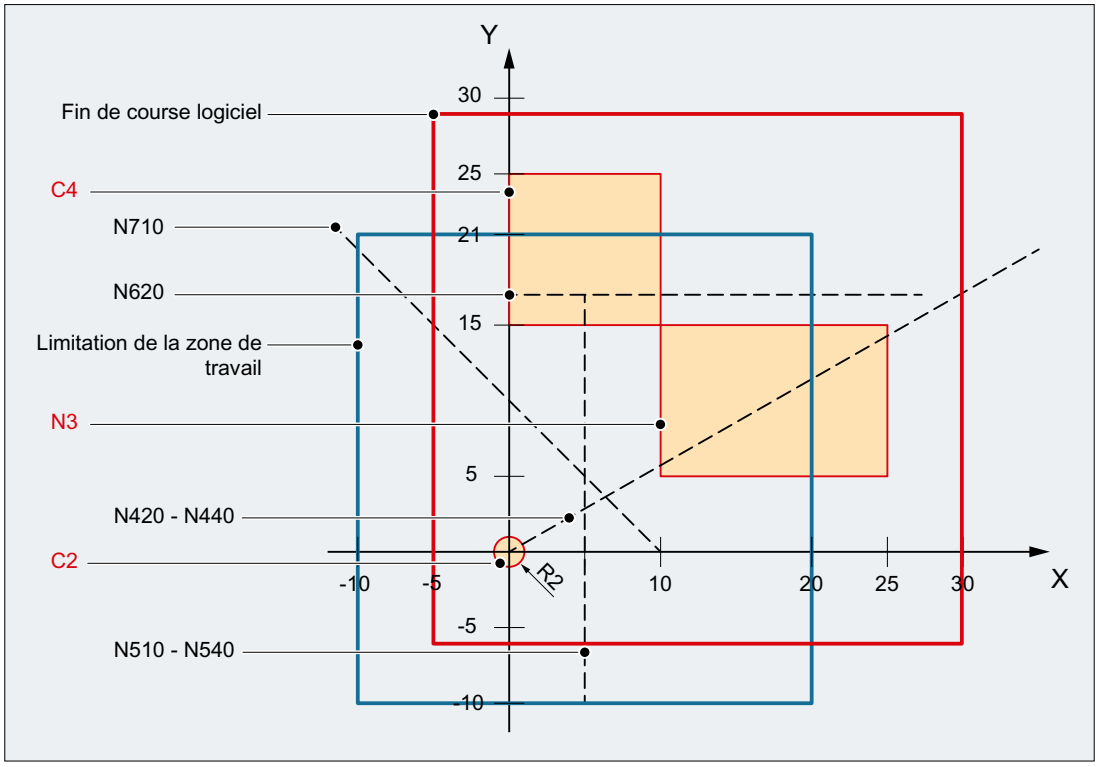
<Limit> :	Référence à une zone de longueur 5. • <Limit> [0 - 2] : distances minimales des axes géométriques aux limitations : – <Limit> [0] : 1. axe géométrique – <Limit> [1] : 2. axe géométrique – <Limit> [2] : 3. axe géométrique Les distances minimales sont respectées pour : – Limitation de la zone de travail : Sans restriction – Fin de course logiciel : Si aucune transformation n'est active, ou si une transformation est active pour laquelle une affectation univoque des axes géométriques à des axes machine linéaires est possible, par ex. transformations à 5 axes. • <Limit> [3] : Contient la distance minimale pour les axes machine linéaires qui ne peuvent être affectés à aucun axe géométrique à cause d'une transformation non linéaire, par exemple. Cette valeur est en outre utilisée en tant que valeur limite pour la surveillance des zones de protection conventionnelles et des zones de protection de la prévention des collisions. • <Limit> [4] : Contient la distance minimale pour les axes machine rotatifs qui ne peuvent être affectés à aucun axe géométrique en raison d'une transformation non linéaire, par exemple. **Remarque** Cette valeur prend effet uniquement lors de la surveillance des fins de course logicielles de transformations spécifiques.	
	Type de paramètre :	Entrée
	Type de données :	VAR REAL[5]
	Plage de valeurs :	Valeur REAL -max. ≤ x[n] ≤ valeur REAL +max.
<MaxDist> :	Référence à un vecteur avec la distance à parcourir relative pour lequel la distance minimale spécifiée d'une limite d'axe n'est dépassée par aucun de l'ensemble des axes machine concernés : • <Dist> [0] : 1. axe géométrique • <Dist> [1] : 2. axe géométrique • <Dist> [2] : 3. axe géométrique Si la distance à parcourir n'est pas limitée, le contenu de ce paramètre de retour est identique à celui de <Dist>.	
	Pour <TestID>, bit 4 == 1 : <Dist> et <MaxDist> <MaxDist> et <Dist> doivent contenir en tant que valeurs d'entrée des vecteurs qui définissent un plan de déplacement. Les deux vecteurs doivent être linéairement dépendants l'un de l'autre. La valeur absolue de <MaxDist> est quelconque. Pour le calcul du sens de déplacement, voir la description de <Dist>.	
	Type de paramètre :	Sortie
	Type de données :	VAR REAL[3]
	Plage de valeurs :	Valeur Valeur REAL ≤ x[<n>] ≤ +max. REAL +max.

<MeasSys> :	Système de mesure (anglo-saxon / métrique) pour les indications de position et de longueur (**facultatif**)		
	Type de données :		BOOL
	Valeur :	FALSE (Par défaut)	Système de mesure correspondant à l'instruction G actuellement active du groupe G 13 (G70, G71, G700, G710). **Remarque** Lorsque G70 est active et que le système de base est métrique ou que G71 est active et que le système de base est anglo-saxon, les variables système $AA_IW et $AA_MW sont fournies dans le système de base et doivent être converties le cas échéant lorsqu'elles sont utilisées pour CALCPOSI.
		TRUE	Système de mesure selon le système de base réglé : PM52806 $MN_ISO_SCALING_SYSTEM
<TestLim> :	Sélection codée sur bits des limitations à surveiller (**facultatif**)		
	Type de données :		INT
	Valeur par défaut :		Bit 0, 1, 2, 3, 6, 7 == 1 (207)
	Bit	**Décimale**	**Signification**
	0	1	Fin de course logiciel
	1	2	Limitation de la zone de travail
	2	4	Zones de protection conventionnelles activées
	3	8	Zones de protection conventionnelles pré-activées
	4	16	En cas de dépassement de fins de course logicielles ou de limitations de zones de travail dans <Dist>, fournir en retour le sens de déplacement selon le **cas 1** (voir ci-dessus).
	5	32	En cas de dépassement de fins de course logicielles ou de limitations de zones de travail dans <Dist>, fournir en retour le sens de déplacement selon le **cas 2** (voir ci-dessus).
	6	64	Zones de protection de la prévention des collisions activées
	7	128	Zones de protection de la prévention des collisions pré-activées
	8	256	Paires de zones de protection activées et pré-activées de la prévention des collisions

[1] En cas de dépassement de plusieurs zones de protection, la zone de protection signalée est celle entraînant la plus forte restriction de la distance à parcourir spécifiée.

Exemple

Limitations



L'exemple représente les fins de course logicielles et les limitations de zone de travail actives dans le plan X-Y ainsi que les trois zones de protection suivantes :

- C2 : Zone de protection spécifique au canal orientée outil, active, circulaire, rayon = 2 mm
- C4 : Zone de protection orientée pièce spécifique au canal, pré-activée, carrée, longueur des côtés = 10 mm
- N3 : Zone de protection spécifique à la machine, active, rectangulaire, longueur des côtés = 10 mm x 15 mm

Programme CN

Dans un premier temps, les zones de protection et les limitations de zone de travail sont définies dans le programme CN. La fonction `CALCPOSI()` est ensuite appelée avec différents paramétrages.

Code de programme

```
N10 DEF REAL _START[3]
N20 DEF REAL _DIST[3]
N30 DEF REAL _LIMIT[5]
N40 DEF REAL _MAXDIST[3]
N50 DEF INT _PA
N60 DEF INT _STATUS
```

Code de programme

```
; Zone de protection C2 relative à l'outil
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
; Zone de protection C4 relative à la pièce
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
; Zone de protection N3 spécifique à la machine
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
; Activation ou pré-activation des zones de protection
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
; Définition des limitations de la zone de travail
N290 G25 XX=-10 YY=-10
N300 G26 XX=20 YY=21
N310 _START[0] = 0.
N320 _START[1] = 0.
N330 _START[2] = 0.
N340 _DIST[0] = 35.
N350 _DIST[1] = 20.
N360 _DIST[2] =  0.
N370 _LIMIT[0] = 0.
N380 _LIMIT[1] = 0.
N390 _LIMIT[2] = 0.
N400 _LIMIT[3] = 0.
N410 _LIMIT[4] = 0.
N420 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)
N430 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,3)
N440 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,1)
N450 _START[0] =  5.
N460 _START[1] = 17.
N470 _START[2] =  0.
N480 _DIST[0] =  0.
N490 _DIST[1] =-27.
N500 _DIST[2] =  0.
N510 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,14)
N520 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N530 _LIMIT[1] = 2.
```

Code de programme

```
N540 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N550 _START[0] = 27.
N560 _START[1] = 17,1
N570 _START[2] =  0.
N580 _DIST[0] =-27.
N590 _DIST[1] =  0.
N600 _DIST[2] =  0.
N610 _LIMIT[3] = 2.
N620 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,12)
N630 _START[0] = 0.
N640 _START[1] = 0.
N650 _START[2] = 0.
N660 _DIST[0] =  0.
N670 _DIST[1] = 30.
N680 _DIST[2] =  0.
N690 TRANS X10
N700 AROT Z45
N710 _STATUS = CALCPOSI(_START,_DIST, _LIMIT, _MAXDIST)
; Suppression des frames de N690 et N700
N720 TRANS
N730 _START[0] =  0.
N740 _START[1] = 10.
N750 _START[2] =  0.
; Les vecteurs _DIST et _MAXDIST définissent le plan de déplacement
N760 _DIST[0] = 30.
N770 _DIST[1] = 30.
N780 _DIST[2] =  0.
N790 _MAXDIST[0] = 1.
N800 _MAXDIST[1] = 0.
N810 _MAXDIST[2] = 1.
N820 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,17)
N830 M30
```

Résultats de CALCPOSI()

N...	<Status>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ Y	Observations
420	3123	8,040	4,594	Dépassement de N3
430	1122	20,000	11,429	Aucune surveillance des zones de protection, la limitation de la zone de travail est dépassée.
440	1121	30,000	17,143	Seule la surveillance des fins de course logicielles est encore active
510	4213	0,000	0,000	Dépassement du point de départ C4
520	0000	0,000	-27,000	La zone de protection pré-activée C4 n'est pas surveillée. Le déplacement peut être exécuté sur l'ensemble de la distance spécifiée.

N...	<Status>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ Y	Observations
540	2222	0,000	-25,000	Comme _LIMIT[1] = 2, la distance à parcourir est restreinte par la limitation de la zone de travail.
620	4223	-13,000	0,000	Du fait de C2 et de _LIMIT[3], la distance totale par rapport à C4 est de 4 mm. La distance C2 → N3 de 0,1 mm ne restreint pas la distance à parcourir.
710	1221	0,000	21,213	Frame actif avec translation et rotation. La distance à parcourir admissible dans _DIST s'applique au SCP décalé et tourné.
820	102121	18,000	18,000	Dépassement du fin de course logiciel de l'axe Y. Le réglage <_TESTLIM> = 17 permet de demander le calcul d'un sens de poursuite du déplacement. Ce sens de déplacement figure dans <_DIST> (0,707 ; 0,0 ; 0,707). Il est valide car la position des centaines de milliers est activée dans <_STATUS>.

Informations complémentaires

État de l'axe "référencé"

Tous les axes machine pris en compte par `CALCPOSI()` doivent être référencés.

Indications de distance relatives à un cercle

Toutes les indications de distance se rapportant à un cercle sont **toujours** interprétées en tant qu'indication de rayon. Ceci doit particulièrement être pris en compte pour les axes transversaux pour lesquels la programmation du diamètre est activée (`DIAMON` / `DIAM90`).

Réduction de la distance à parcourir

Si la distance à parcourir spécifiée d'un axe est limitée, la distance à parcourir des autres axes est également réduite proportionnellement dans la valeur de retour `<MaxDist>`. Le point final obtenu se trouve par conséquent toujours sur la trajectoire spécifiée.

Axes rotatifs

Les axes rotatifs sont surveillés uniquement si ce ne sont pas des axes rotatifs modulo.

L'absence de définition de fins de course logicielles, de limitations de la zone de travail ou de zones de protection pour un ou plusieurs des axes impliqués est admissible.

État des fins de course logicielles et des limitations de la zone de travail

Les fins de course logicielles et les limitations de la zone de travail sont pris en compte uniquement s'ils sont actifs lors de l'exécution de `CALCPOSI()`. L'état peut être influencé par exemple par :

- Paramètres machine : PM21020 $MC_WORKAREA_WITH_TOOL_RADIUS
- Données de réglage : $AC_WORKAREA_CS_...

- Signaux d'interface CN/AP : DB31, ... DBX12.2 / 3
- Instructions : WALIMON / WALIMOF

Fins de course logicielles et transformations

Pour CALCPOSI() en présence de différentes transformations cinématiques (par ex. TRANSMIT), les positions des axes machine (SCM) ne peuvent pas toujours être déterminées de manière univoque à partir des positions des axes géométriques (SCP) en raison d'ambiguïtés à certains endroits de la plage de déplacement. En mode de déplacement normal, l'univocité résulte en règle générale des précédents et de la condition selon laquelle un mouvement continu des axes machine dans le SCP doit correspondre à un mouvement continu dans le SCM. C'est pourquoi, lors de la surveillance des fins de course logicielles, la position de la machine présente au moment de l'exécution de CALCPOSI() est utilisée dans ce type de cas pour résoudre l'ambiguïté.

Remarque

Arrêt du prétraitement des blocs

Lorsque CALCPOSI() est utilisée en relation avec des transformations, l'utilisateur assume l'entière responsabilité quant à la programmation d'un arrêt du prétraitement des blocs (STOPRE) pour la synchronisation des positions d'axes machine avec le prétraitement des blocs avant CALCPOSI().

Distance de la zone de protection et zones de protection conventionnelles

Pour les zones de protection conventionnelles, il n'est **pas** garanti que lors d'un mouvement de déplacement sur la distance à parcourir spécifiée, la distance de sécurité renseignée dans <Limit>[3] est respectée par rapport à toutes les zones de protection. Il est seulement garanti que lors d'un prolongement du point final fourni en retour dans <Dist> correspondant à la distance de sécurité dans le sens de déplacement, aucune zone de protection n'est dépassée. Le parcours de la droite peut cependant passer plus ou moins près d'une zone de protection.

Distance de la zone de protection et zones de protection de la prévention des collisions

Pour les zones de protection de la prévention des collisions, il est garanti que lors d'un mouvement de déplacement sur la distance à parcourir spécifiée, la distance de sécurité renseignée dans le paramètre <Limit>[3] est respectée par rapport à toutes les zones de protection.

La distance de sécurité renseignée dans le paramètre <Limit>[3] prend uniquement effet lorsque :

<Limit>[3] > (PM10619 $MN_COLLISION_TOLERANCE)

Si dans le paramètre <TestLim> le bit 4 est mis à 1 (calcul du sens de la poursuite du déplacement), le vecteur de direction contenu dans <DIST> n'est valide que lorsque la position des centaines de milliers est activée dans la valeur de retour de la fonction (<Status>). Si un tel sens ne peut pas être déterminé, soit en raison d'un dépassement des zones de protection, soit parce qu'une transformation est active, la valeur d'entrée dans <DIST> reste inchangée. Aucun autre message d'erreur n'est généré.

Instructions de déplacement spéciales 5

5.1 Accostage de positions codées (CAC, CIC, CDC, CACP, CACN)

Les instructions suivantes permettent, au moyen de numéros de position, de déplacer des axes linéaires et rotatifs jusqu'à des positions d'axe fixes définies dans des tableaux de paramètres machine. Ce type de programmation est appelé "accostage de positions codées".

Syntaxe

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

Signification

`CAC(<n>)` :	Accostage de la position codée du numéro de position n
`CIC(<n>)` :	Accostage de la position codée n positions en avant (+n) ou en arrière (–n), à partir du numéro de position courant
`CDC(<n>)` :	Accostage de la position codée sur le chemin le plus court depuis le numéro de position n (uniquement pour les axes rotatifs)
`CACP(<n>)` :	Accostage de la position codée dans le sens positif depuis le numéro de position n (uniquement pour les axes rotatifs)
`CACN(<n>)` :	Accostage de la position codée dans le sens négatif depuis le numéro de position n (uniquement pour les axes rotatifs)
`<n>` :	Numéro de position dans le tableau de paramètres machine Plage de valeurs : 0, 1, … (nombre maximal de positions dans le tableau - 1)

Exemple : Accostage des positions codées d'un axe de positionnement

Code de programme	**Commentaire**
`N10 FA[B]=300`	`; Avance de l'axe de positionnement B`
`N20 POS[B]=CAC(10)`	`; Accostage de la position codée du numéro de position 10`
`N30 POS[B]=CIC(-4)`	`; Accostage de la position codée du "numéro de position courant" - 4`

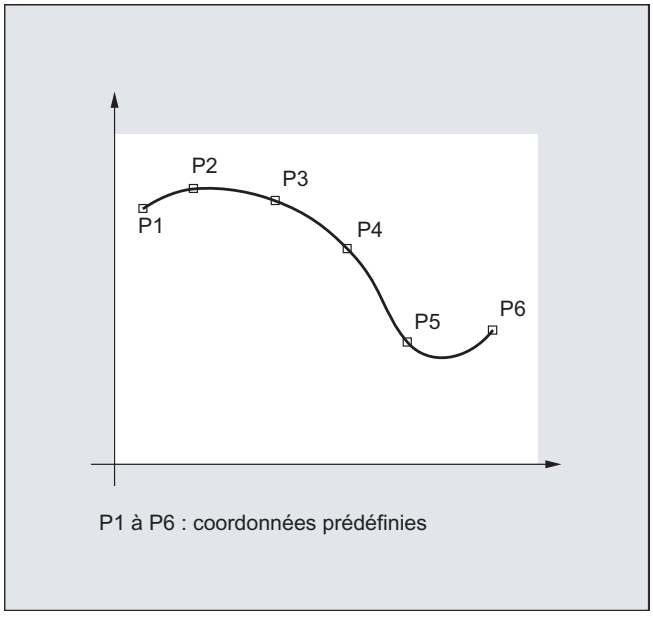
Bibliographie

- Description fonctionnelle Fonctions d'extension ; Axes indexés (T1)
- Description fonctionnelle Actions synchrones

5.2 Interpolation de type spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Pour les contours de pièce de forme courbe quelconque, une description analytique exacte est impossible. C'est pourquoi de tels contours sont décrits approximativement par un nombre limité de points intermédiaires, notamment lors de la numérisation des surfaces. Pour générer une surface de pièce numérisée, il est nécessaire de relier les points intermédiaires et d'en faire une description du contour. C'est ce que permet l'interpolation de type spline.

Un spline définit une courbe composée de polynômes du 2e ou 3e degré. Les propriétés des points intermédiaires d'un spline peuvent être définis **selon le type de spline utilisé**.



SINUMERIK solution line met à disposition des types de spline suivants :

- Spline A
- Spline B
- Spline C

Syntaxe

Généralités :

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

Le spline B permet en outre de programmer :

```
PW=<n>
SD=2
PL=<valeur>
```

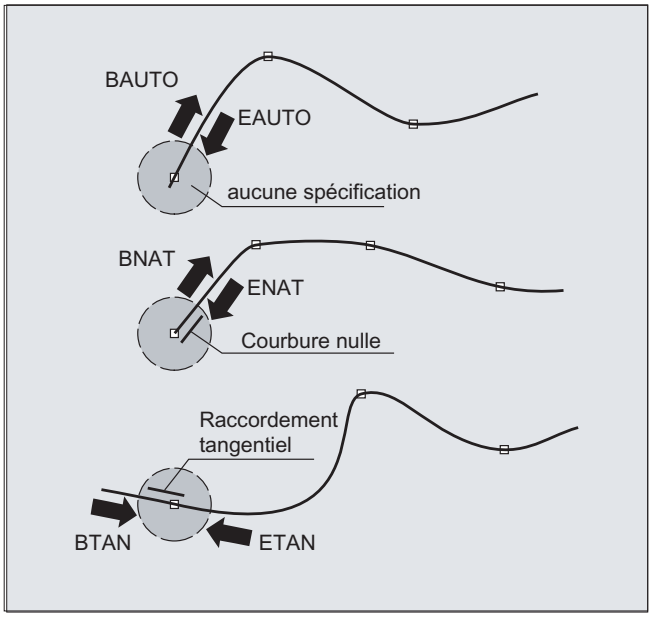
Les splines A et C permettent en outre de programmer :

`BAUTO` / `BNAT` / `BTAN`

EAUTO / ENAT / ETAN

Signification

Type d'interpolation spline :			
ASPLINE :	Instruction d'activation de l'interpolation de type spline A		
BSPLINE :	Instruction d'activation de l'interpolation de type spline B		
CSPLINE :	Instruction d'activation de l'interpolation de type spline C		
	Les instructions ASPLINE, BSPLINE et CSPLINE ont un effet modal et font partie du groupe des instructions de déplacement.		
Points intermédiaires et points de contrôle :			
X... Y... Z... A... B... C...	Positions en coordonnées cartésiennes		
Poids de point (uniquement spline B) :			
PW :	L'instruction PW permet de programmer un "poids de point" pour chaque point intermédiaire.		
<n> :	"Poids de point"		
	Plage de valeurs :	0 ≤ n ≤ 3	
	Pas :	0.0001	
	Effet :	n > 1	La courbe tend plus fortement.
		n < 1	La courbe tend moins fortement.
Degré spline (uniquement spline B) :			
SD :	Un polygone du troisième degré est appliqué de façon standard. Mais la programmation de SD=2 permet également d'utiliser un polygone du deuxième degré.		
Distance entre les pôles (uniquement spline B) :			
PL :	Les distances entre les pôles sont calculées par le système, Mais la commande peut également traiter les distances prescrites en tant que "Paramètre - Intervalle - Longueur" avec l'instruction PL.		
<valeur> :	Paramètre-Intervalle-Longueur		
	Plage de valeurs :	comme pour le positionnement non indexé	
Comportement de transition au début de la courbe spline (splines A et C uniquement) :			
BAUTO :	Aucune prescription pour le comportement de transition. Le début résulte de la position du premier point.		
BNAT :	Courbure nulle		
BTAN :	Raccordement tangentiel avec bloc précédent (fonction initialisée)		
Comportement de transition à la fin de la courbe spline (splines A et C uniquement) :			
EAUTO :	Aucune prescription pour le comportement de transition. La fin résulte de la position du dernier point.		
ENAT :	Courbure nulle		

ETAN :	Raccordement tangentiel avec bloc précédent (fonction initialisée)
	

Remarque

Le comportement de transition programmable n'influence pas le spline B. Au point de départ et au point final, le spline B est toujours tangent au polygone de contrôle.

Autres conditions à prendre en compte

- La correction du rayon d'outil est utilisable.
- La surveillance anticollision s'effectue par projection dans le plan.

Exemples

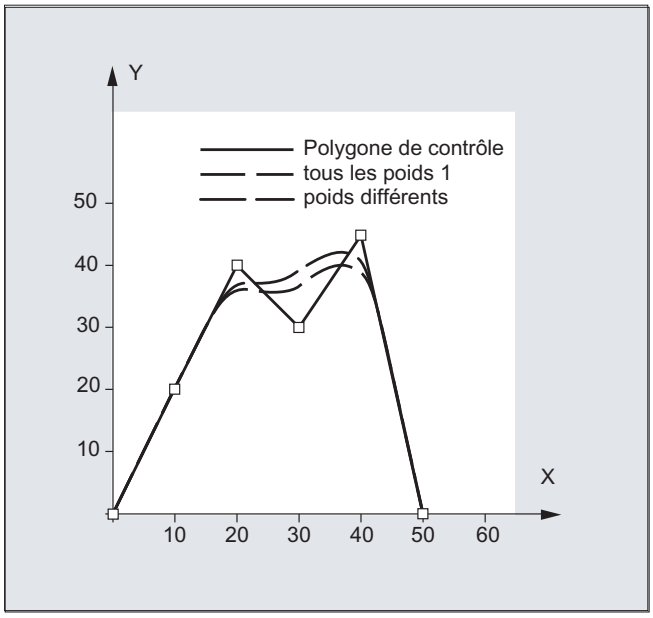
Exemple 1 : Spline B

```
Code de programme 1 (tous les poids 1)
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

Code de programme 2 (différents poids)

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20 PW=2
N40 X20 Y40
N50 X30 Y30 PW=0.5
N60 X40 Y45
N70 X50 Y0
```

Code de programme 3 (polygone de contrôle)	**Commentaire**
N10 G1 X0 Y0 F300 G64	
N20	; ne s'applique pas
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	

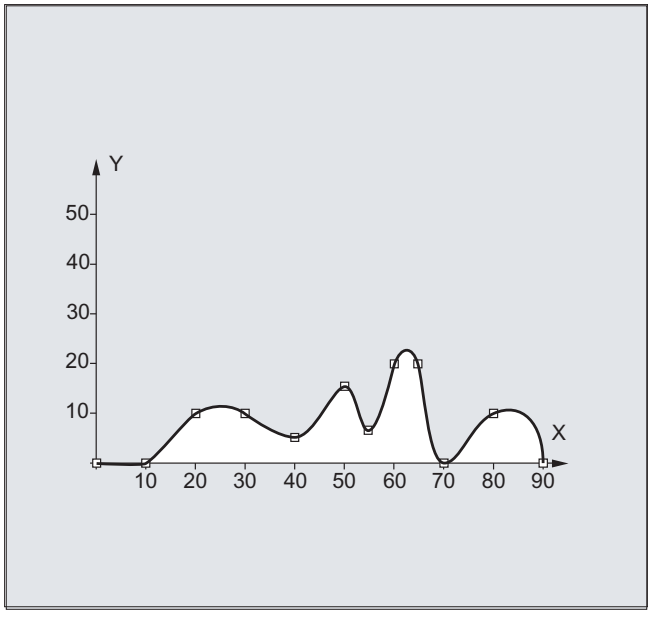


Exemple 2 : Spline C, courbure nulle au début et à la fin

Code de programme

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
```

```
Code de programme
N40 X30
N50 X40 Y5
N60 X50 Y15
N70 X55 Y7
N80 X60 Y20
N90 X65 Y20
N100 X70 Y0
N110 X80 Y10
N120 X90 Y0
N130 M30
```



Exemple 3 : Interpolation spline (spline A) et transformation de coordonnées (ROT)

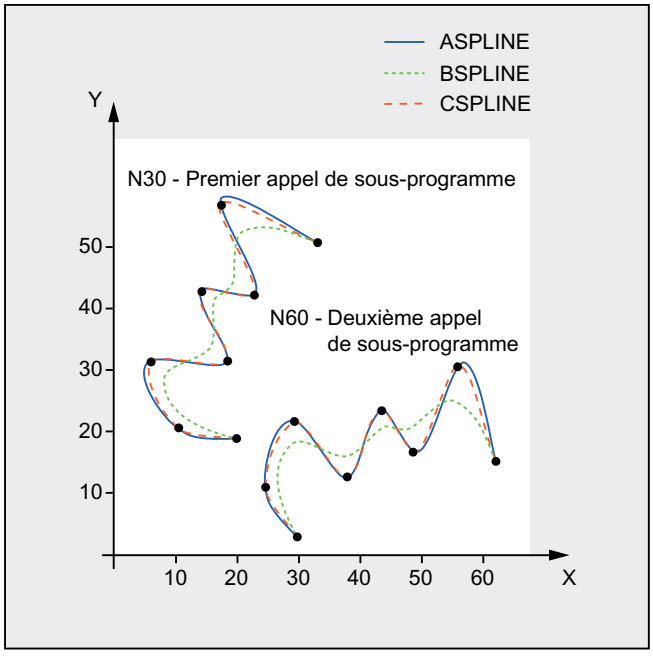
Programme principal :

Code de programme	Commentaire
N10 G00 X20 Y18 F300 G64	; Accostage du point de départ
N20 ASPLINE	; Activation de l'interpolation de type spline A.
N30 CONTOUR	; Premier appel du sous-programme.
N40 ROT Z-45	; Transformation de coordonnées : rotation du SCP de -45° autour de l'axe Z.
N50 G00 X20 Y18	; Accostage du point de départ du contour.
N60 CONTOUR	; Deuxième appel du sous-programme.
N70 M30	; Fin du programme

Sous-programme "Contour" (contenant les coordonnées des points intermédiaires) :

```
Code de programme
N10 X20 Y18
N20 X10 Y21
N30 X6 Y31
N40 X18 Y31
N50 X13 Y43
N60 X22 Y42
N70 X16 Y58
N80 X33 Y51
N90 M1
```

La figure suivante contient la courbe spline résultante de l'exemple de programme (ASPLINE), ainsi que les courbes splines qui auraient résulté d'une interpolation de type spline B ou C (BSPLINE, CSPLINE) :



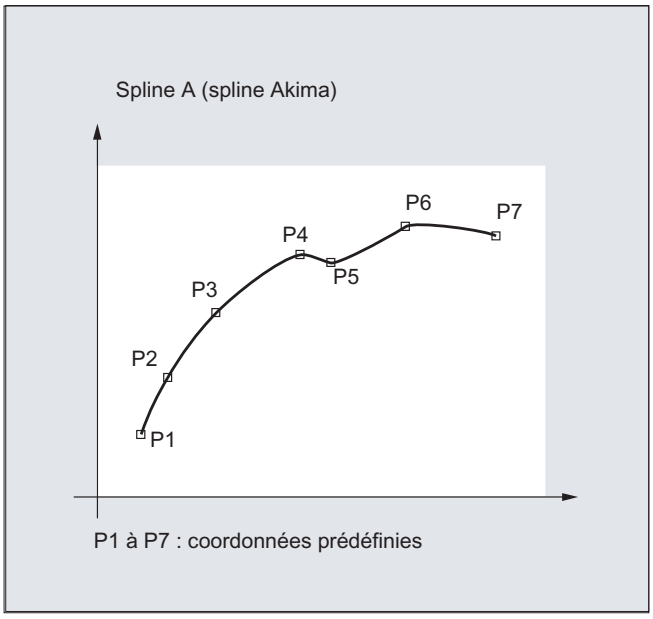
plus d'informations

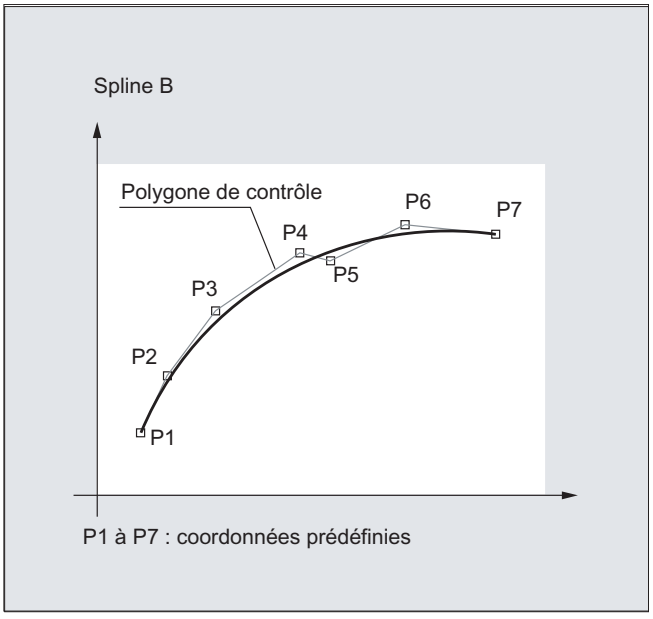
Avantages de l'interpolation de type spline

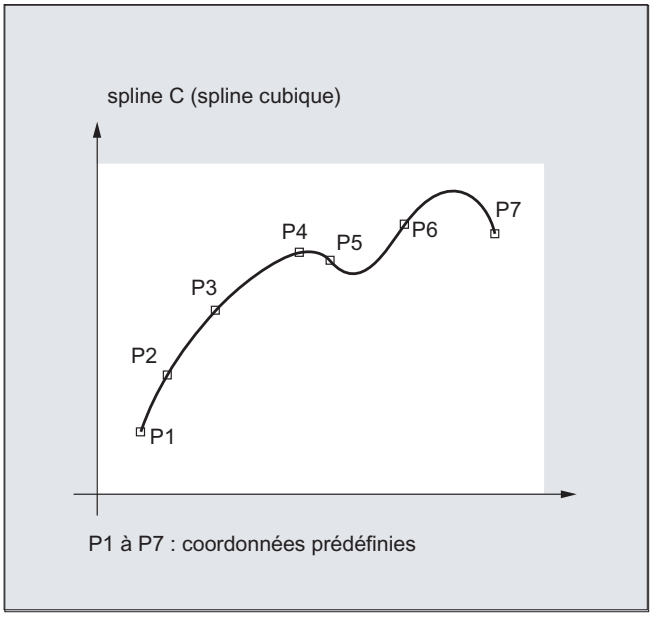
Contrairement à l'utilisation de blocs linéaires `G01`, l'interpolation de type spline présente les avantages suivants :

- Réduction de nombre de blocs requis dans le programme pièce pour la description du contour
- Courbes douces ménageant la mécanique lors des transitions entre les blocs du programme pièce

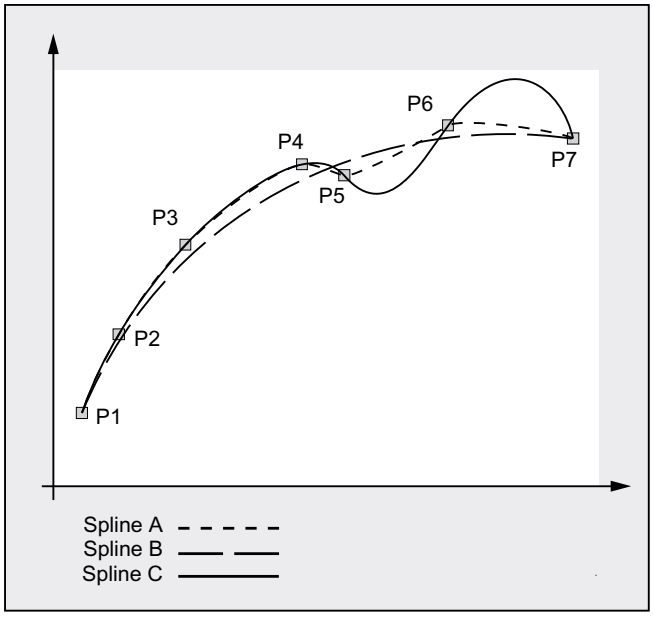
Propriétés et applications des différents types de spline

Type spline	Propriétés et applications
Spline A	 **Propriétés :** • Courbe passant exactement par les points intermédiaires indiqués. • La courbe est caractérisée par la continuité de la tangente, mais pas du rayon de courbure. • Elle ne crée presque pas de vibrations indésirables. • La modification d'un point intermédiaire a un effet local, ce qui signifie qu'elle n'influence pas plus de 6 points intermédiaires voisins. **Application :** Le spline A convient avant tout à l'interpolation de courbes avec de grandes variations de pente (par exemple les courbes en forme d'escalier).

Type spline	Propriétés et applications
Spline B	 **Propriétés :** • Courbe ne passant pas par les points intermédiaires indiqués, mais à proximité de ceux-ci. La courbe tend vers les points intermédiaires. L'utilisation d'un facteur de pondération des points intermédiaires permet en outre d'influencer le tracé de la courbe. • La courbe est caractérisée par la continuité de la tangente et du rayon de courbure. • Elle ne crée aucune vibration indésirable. • La modification d'un point intermédiaire a un effet local, ce qui signifie qu'elle n'influence pas plus de 6 points intermédiaires voisins. **Application :** Le spline B est destiné en premier lieu à servir d'interface avec les systèmes de CAO.

Type spline	Propriétés et applications
Spline C	 **Propriétés :** • Courbe passant exactement par les points intermédiaires indiqués. • La courbe est caractérisée par la continuité de la tangente et du rayon de courbure. • Elle crée fréquemment des vibrations indésirables, en particulier aux endroits avec de grandes variations de pente. • La modification d'un point intermédiaire a un effet global, ce qui signifie qu'elle influence l'ensemble du tracé de la courbe. **Application :** Le spline C convient particulièrement bien lorsque les points se situent sur une courbe analytique connue (cercle, parabole, hyperbole).

Comparaison des trois types de splines avec des points intermédiaires identiques



Nombre minimum de blocs spline

Les instructions G `ASPLINE`, `BSPLINE` et `CSPLINE` relient les points finaux de blocs par des courbes de type spline. Pour ce faire, la commande doit effectuer le décodage simultané d'une série de blocs (points finaux) pendant le prétraitement. En version standard, la taille du tampon pour le décodage vaut 10 blocs. Toutes les informations de bloc ne sont pas forcément des fins de bloc spline. La commande nécessite néanmoins un nombre précis de fins de bloc spline provenant de 10 blocs :

Type spline	Nombre minimum de blocs spline
spline A :	Sur 10 blocs, **4** au moins doivent être des blocs spline. Les blocs de commentaire et les calculs de paramètres ne sont pas pris en compte.
spline B :	Sur 10 blocs, **6** au moins doivent être des blocs spline. Les blocs de commentaire et les calculs de paramètres ne sont pas pris en compte.
spline C :	Le nombre minimum de blocs spline nécessaire se calcule de la manière suivante : valeur de PM20160 $MC_CUBIC_SPLINE_BLOCKS + 1 Le nombre de points à l'aide desquels la section de spline est calculée est renseigné dans PM20160. Le paramétrage par défaut est 8. Sur 10 blocs, **9** au moins doivent être des blocs spline dans le cas standard.

Remarque

Si la valeur est inférieure à la valeur tolérée, une alarme est émise, de même que si un axe participant à l'interpolation de type spline est programmé comme axe de positionnement.

Groupement de blocs spline courts

Lors de l'interpolation de type spline des blocs spline courts peuvent être créés, qui contribue inutilement à la réduction de la vitesse tangentielle. La fonction "groupement de blocs spline courts" permet de grouper ces blocs de manière que la longueur de bloc en résultant soit assez longue et à ne pas réduire la vitesse tangentielle.

La fonction est activée via le paramètre machine spécifique au canal :

PM20488 $MC_SPLINE_MODE (réglage pour interpolation de type spline)

Bibliographie :
Description fonctionnelle Fonctions de base ; contournage, arrêt précis, look-ahead (B1), chapitre : Groupement de blocs spline courts

5.3 Groupe spline (SPLINEPATH)

Les axes à interpoler dans le groupe spline sont sélectionnés avec l'instruction `SPLINEPATH`. L'interpolation de type spline admet jusqu'à 8 axes à interpolation.

Remarque

Si l'instruction `SPLINEPATH` n'est pas programmée de façon explicite, ce sont les trois premiers axes du canal qui seront déplacés en tant que groupe spline.

Syntaxe

La définition du groupe spline se fait dans un bloc séparé :

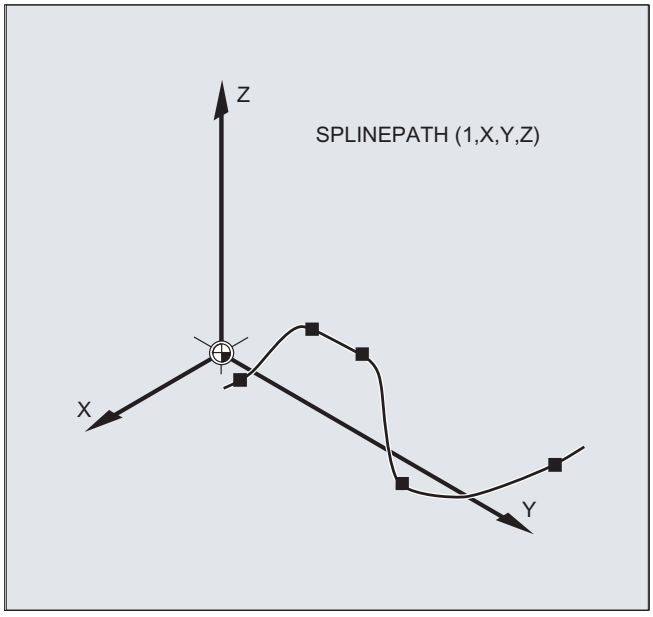
```
SPLINEPATH(n,X,Y,Z,…)
```

Signification

`SPLINEPATH` :	Instruction de définition du groupe spline
`n` :	=1 (valeur fixe)
`X,Y,Z,…` :	Descripteurs des axes à interpoler dans le groupe spline

Exemple : Groupe spline avec trois axes à interpolation

```
Code de programme                                   Commentaire
N10 G1 X10 Y20 Z30 A40 B50 F350
N11 SPLINEPATH(1,X,Y,Z)                             ; Groupe spline
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60         ; Spline C
N14 X30 Y40 Z50 A60 B70                             ; Points intermédiaires
…
N100 G1 X… Y…                                       ; Désactivation de l'interpo-
                                                    lation de type spline
```


Z
SPLINEPATH (1,X,Y,Z)
X
Y

5.4 Activer/désactiver le compactage de bloc CN (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)

Les fonctions de compactage de blocs linéaires (et aussi de blocs circulaires et/ou à vitesse rapide selon le paramétrage) sont activées/désactivées avec les instructions G du groupe G 30. Les instructions sont des fonctions à effet modal.

Syntaxe

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

Signification

COMPON :	Activation de la fonction de compresseur COMPON
COMPCURV :	Activation de la fonction de compresseur COMPCURV
COMPCAD :	Activation de la fonction de compresseur COMPCAD
COMPSURF:	Activation de la fonction de compresseur COMPSURF
COMPOF :	Désactivation de la fonction de compresseur actuellement active

Remarque

Pour une amélioration supplémentaire de l'état de surface, il est possible d'utiliser la fonction d'arrondissement G642 et la limitation des à-coups SOFT. Ces instructions doivent figurer en début de programme.

Exemple : COMPCAD

```
Code de programme          Commentaire
N10 G00 X30 Y6 Z40
N20 G1 F10000 G642         ; Activation : fonction d'arrondissement G642
N30 SOFT                   ; Activation : limitation des à-coups SOFT
N40 COMPCAD                ; Activation : fonction de compresseur COMPCAD
N50 STOPFIFO
N24050 Z32.499             ; 1er bloc de déplacement
N24051 X41.365 Z32.500     ; 2ème bloc de déplacement
...
N99999 X... Z...           ; Dernier bloc de déplacement
COMPOF                     ; Fonction de compresseur désactivée.
...
```

5.5 Interpolation polynomiale (POLY, POLYPATH, PO, PL)

L'interpolation polynomiale (`POLY`) n'est en fait pas une interpolation de type spline. Elle est destinée avant tout à servir d'interface pour la programmation de courbes spline générées de manière externe. Les sections spline peuvent être programmées directement.

Ce mode d'interpolation évite à la CN de calculer les coefficients de polynôme. Elle peut être utilisée de manière optimale lorsque les coefficients sont issus directement d'un système de CAO ou d'un postprocesseur.

Syntaxe

Polynôme du 3e degré :
```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

Polynôme du 5e degré et nouvelle syntaxe polynomiale :
```
POLY X=PO(xe,a2,a3,a4,a5) Y=PO(ye,b2,b3,b4,b5) Z=PO(ze,c2,c3,c4,c5)
PL=n
POLYPATH("AXES","VECT")
```

Remarque

La somme des coefficients de polynôme et des axes programmés dans un bloc CN ne doit pas dépasser le nombre maximal d'axes autorisés par bloc.

Signification

`POLY` :	Activation de l'interpolation polynomiale par un bloc avec POLY.
`POLYPATH` :	Interpolation polynomiale sélectionnable pour les deux groupes d'axes AXIS ou VECT
`PO[descripteur d'axe/variable]` :	Points finals et coefficients de polynôme
`X, Y, Z` :	Descripteur d'axe
`xe, ye, ze` :	Indication de la position finale de l'axe correspondant ; plage de valeurs comme pour le positionnement non indexé
`a2, a3, a4, a5` :	Les coefficients a_2, a_3, a_4, et a_5 sont écrits avec leur valeur ; plage de valeurs comme pour le positionnement non indexé. Le dernier coefficient peut être omis si sa valeur est égale à zéro.
`PL` :	Longueur de l'intervalle de paramètre sur lequel le polynôme est défini (domaine de définition de la fonction f(p)). L'intervalle commence toujours à 0 et la valeur de p peut aller de 0 à PL. Plage de valeurs théorique de PL : 0,0001 … 99 999,9999 **Remarque :** La valeur de PL est valable pour le bloc dans lequel elle figure. Si PL n'est pas programmé, PL=1.

Activation/désactivation de l'interpolation polynomiale

L'interpolation polynomiale est activée dans le programme pièce au moyen de l'instruction G `POLY`.

L'instruction G `POLY` ainsi que `G0`, `G1`, `G2`, `G3`, `ASPLINE`, `BSPLINE` et `CSPLINE` font partie du 1er groupe G.

Etant uniquement programmés avec un nom et un point final (par ex. `X10`) les axes sont déplacés linéairement. Après avoir programmé tous les axes d'un bloc CN de la sorte, la commande se comporte comme pour G1.

L'interpolation polynomiale est de nouveau désactivée de manière implicite en programmant une autre instruction du 1er groupe G (par ex. `G0`, `G1`).

Coefficient du polynôme

La valeur `(PO[]=)` ou `...=PO(...)` indique tous les coefficients de polynôme d'un axe. Selon le degré du polynôme, plusieurs valeurs séparées par des virgules sont spécifiées. Dans un bloc, différents degrés de polynôme de différents axes sont possibles.

Sous-programme POLYPATH

`POLYPATH(...)` permet de débloquer l'interpolation polynomiale de manière sélective pour certains groupes d'axes :

Uniquement axes à interpolation et axes supplémentaires :	`POLYPATH("AXES")`
Uniquement axes d'orientation : (lors du déplacement avec transformation d'orientation)	`POLYPATH("VECT")`

Les axes respectifs non débloqués sont déplacés linéairement.

Par défaut, l'interpolation polynomiale est débloquée pour les deux groupes d'axes.

Une programmation sans paramètre `POLYPATH( )` permet de désactiver l'interpolation polynomiale pour tous les axes.

Exemple

Code de programme	**Commentaire**
`N10 G1 X… Y… Z… F600`	
`N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2) PL=1.5`	`; Interpolation polynomiale activée`
`N12 PO[X]=(0,2.5,1.7) PO[Y]=(2.3,1.7) PL=3`	
`...`	
`N20 M8 H126 …`	
`N25 X70 PO[Y]=(9.3,1,7.67) PL=5`	`; Indications mixtes pour les axes`
`N27 PO[X]=(10,2.5) PO[Y]=(2.3)`	`; PL n'est pas programmé, donc PL=1.`
`N30 G1 X… Y… Z.`	`; Interpolation polynomiale désactivée`

Code de programme	Commentaire
…	

Exemple : Nouvelle syntaxe polynomiale

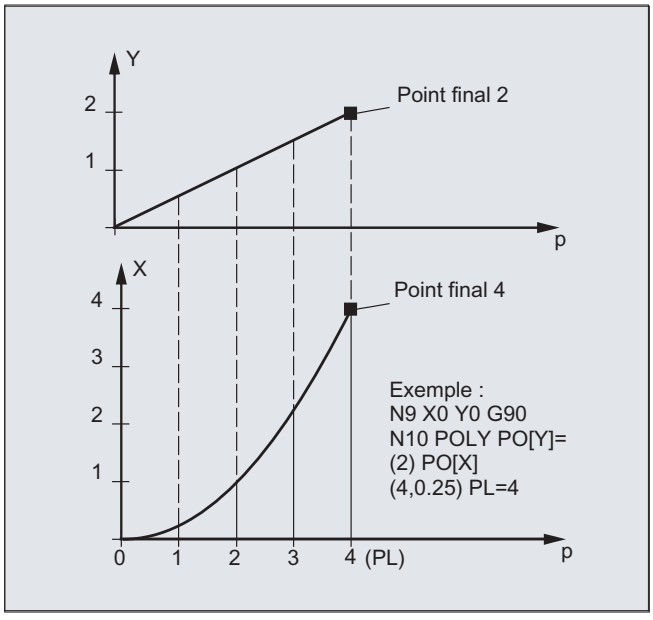
Syntaxe polynomiale encore valable	Nouvelle syntaxe polynomiale
PO[descripteur d'axe]=(.. , ..)	Descripteur d'axe=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[variable]=IC(.. , ..)	Variable=PO IC(.. , ..)

Exemple : Cercle dans le plan X/Y

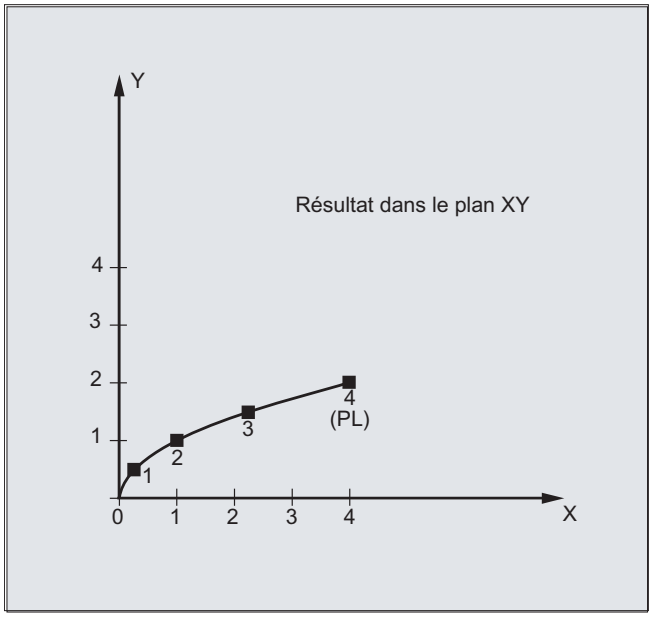
Programmation

Code de programme
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

Evolution des courbes X(p) et Y(p)



Evolution de la courbe dans le plan XY



Description

La forme générale de la fonction polynomiale est la suivante :

$f(p)= a_0 + a_1p + a_2p^2 + . . . + a_np^n$

où : a_i : coefficients constants (i = 0, 1, ..., n)
p : paramètres

Le nombre maximum de polynômes du 5e degré pouvant être programmés dans la commande est le suivant :

$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$

En affectant des valeurs concrètes aux coefficients, il est possible de générer différentes courbes : droites, paraboles et fonctions puissance.

Une courbe droite est créée par $a_2 = a_3 = a_4 = a_5 = 0$:

$f(p)= a_0 + a_1p$

En outre :

a_0 : position de l'axe à la fin du bloc précédent
p = PL
$a_1 = (x_E - a_0 - a_2*p^2 - a_3*p^3) / p$

Des polynômes peuvent être programmés
sans avoir que l'interpolation polynomiale n'ait été activée au moyen de l'instruction G `POLY`. Dans ce cas, les polynômes programmés ne seront pas interpolés, mais les points finaux programmés pour les axes seront accostés linéairement (`G1`). Les polynômes programmés ne sont déplacés en tant que tels qu'après avoir explicitement activé l'interpolation polynomiale dans le programme pièce (`POLY`).

Particularité : polynôme dénominateur

Pour les axes géométriques, `PO[]=(…)` permet aussi de programmer un polynôme dénominateur commun sans indiquer le nom d'un axe, c'est-à-dire que le déplacement des axes géométriques sera interpolé en tant que quotient de deux polynômes.

Ceci permet, par exemple, de représenter avec exactitude des coniques (cercle, ellipse, parabole, hyperbole).

Exemple :

Code de programme	**Commentaire**
`POLY G90 X10 Y0 F100`	`; Déplacement linéaire des axes géométriques à la position X10 Y0.`
`PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)`	`; Déplacement des axes géométriques en quart de cercle sur position X0 Y10.`

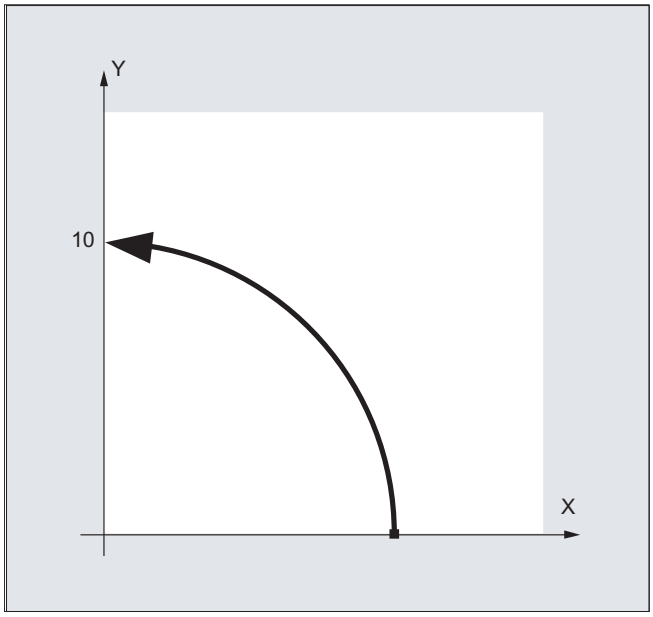
Le coefficient constant (a_0) du polynôme dénominateur est toujours considéré comme étant égal à 1. Le point final programmé est indépendant de `G90` / `G91`.

X(p) et Y(p) sont calculés comme suit à partir des valeurs programmées :

$X(p) = (10 - 10 * p^2) / (1 + p^2)$
$Y(p) = 20 * p / (1 + p^2)$
avec $0 \leq p \leq 1$

Compte tenu de la programmation des points de départ, des points finaux, du coefficient a_2 et de PL=1, les résultats intermédiaires suivants sont obtenus :

Numérateur (X) = $10 + 0 * p - 10 * p^2$
Numérateur (Y) = $0 + 20 * p + 0 * p^2$
Dénominateur = $1 + p^2$



Lorsque l'interpolation polynomiale est activée, la programmation d'un polynôme dénominateur possédant des racines à l'intérieur de l'intervalle `[0,PL]` est rejetée avec une alarme. Le polynôme dénominateur n'a aucune influence sur les déplacements des axes supplémentaires.

Remarque

Dans le cas de l'interpolation polynomiale, on peut activer une correction de rayon d'outil avec `G41`, `G42` et l'utiliser comme pour l'interpolation linéaire ou circulaire.

5.6 Référence de trajectoire réglable (SPATH, UPATH)

Pour l'interpolation polynomiale (POLY, ASPLINE, BSPLINE, CSPLINE, COMPON, COMPCURV), les positions des axes à interpolation i sont spécifiées par des polynômes $p_i(U)$. Dans un bloc CN, le paramètre de courbe U varie de 0 à 1.

FGROUP permet de sélectionner les axes (axes FGROUP) auxquels se rapporte l'avance tangentielle F. Toutefois, une interpolation à vitesse constante sur la trajectoire S des axes FGROUP signifie généralement une variation non constante du paramètre de courbe U pendant l'interpolation polynomiale. Pour les axes non contenus dans FGROUP, il est ainsi possible de choisir entre deux possibilités de poursuite en fonction des axes FGROUP :

- de manière synchrone avec la trajectoire S (SPATH)
- de manière synchrone avec le paramètre de courbe U (UPATH)

Syntaxe

```
SPATH
UPATH
```

Signification

SPATH :	Les axes non contenus dans FGROUP sont déplacés en fonction de la trajectoire S
UPATH :	Les axes non contenus dans FGROUP sont déplacés en fonction du paramètre de courbe U

Remarque

UPATH et SPATH déterminent également la relation entre le polynôme décrivant le mot F (FPOLY, FCUB, FLIN) et le déplacement le long de la trajectoire.

Autres conditions à prendre en compte

SPATH ou UPATH n'ont aucune signification pour :

- interpolation linéaire (G1)
- interpolation circulaire (G2, G3)
- blocs de filetage (G33, G34, G35, G33x, G63)
- Tous les axes à interpolation sont contenus dans FGROUP

Exemple

L'exemple suivant illustre la différence entre les deux modes de pilotage du déplacement.

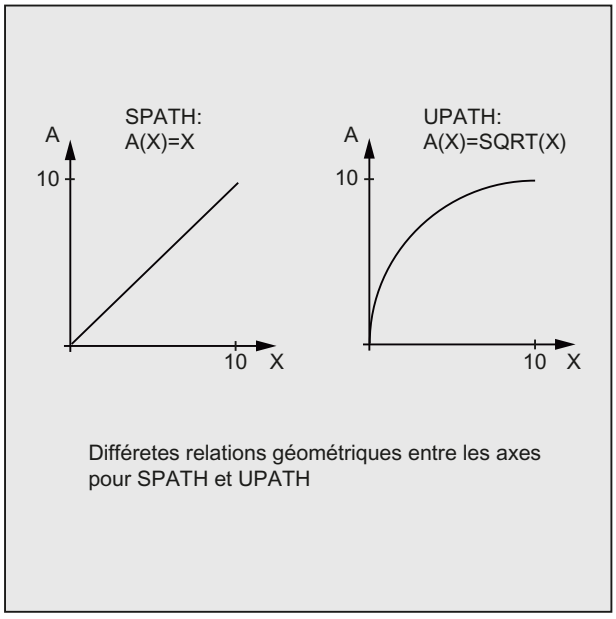
Code de programme

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 SPATH                ; SPATH
N20 POLY PO[X]=(10,10) A10
```

Code de programme

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH                ; UPATH
N20 POLY PO[X]=(10,10) A10
```

Dans les deux sections de programme, la trajectoire S dans `N20` des axes FGROUP dépend du carré du paramètre de courbe U. C'est pourquoi on obtient des positions différentes de l'axe synchrone A en fonction du déplacement de X, selon que `SPATH` ou `UPATH` est actif :



Différetes relations géométriques entre les axes pour SPATH et UPATH

Informations complémentaires

Comportement de la commande en cas de Reset et paramètres machine/d'option

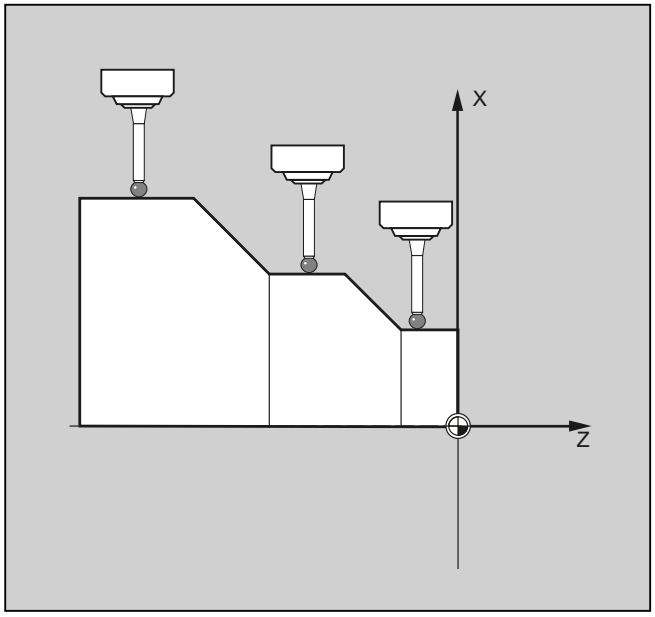
Après un Reset, l'instruction G définie par PM20150 $MC_GCODE_RESET_VALUES[44] est active (45e groupe G).

La valeur de position initiale pour le type d'arrondissement est définie par PM20150 $MC_GCODE_RESET_VALUES[9] (10e groupe G).

Le paramètre machine spécifique à l'axe PM33100 $MA_COMPRESS_POS_TOL[<n>] a une signification étendue : il contient les tolérances pour la fonction compresseur et l'arrondissement avec G642.

5.7 Mesure avec palpeur à déclenchement (MEAS, MEAW)

La fonction "Mesure avec palpeur à déclenchement" permet d'accoster des positions réelles sur la pièce, de mesurer les positions de chaque axe programmé dans le bloc de mesure lors du front de déclenchement du palpeur et de les écrire dans la cellule mémoire correspondante.



Les deux adresses fixes suivantes sont disponibles pour la programmation de la fonction :

- `MEAS`
  L'instruction `MEAS` permet d'effacer la distance restant à parcourir entre la position réelle et la position de consigne.
- `MEAW`
  `MEAW` s'utilise pour les tâches de mesure où la position programmée doit impérativement être accostée.

`MEAS` et `MEAW` sont à effet non modal et peuvent être programmées avec des instructions de déplacement. L'avance et le type d'interpolation (G0, G1...), de même que le nombre d'axes, doivent être adaptés au problème de mesure concerné.

Syntaxe

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

Signification

`MEAS` :	Instruction : Mesure **avec** effacement de la distance restant à parcourir	
	Prise d'effet :	bloc par bloc (non modale)
`MEAW` :	Instruction : Mesure **sans** effacement de la distance restant à parcourir	
	Prise d'effet :	bloc par bloc (non modale)

<TE> :	Evénement déclencheur de la mesure		
	Type :	INT	
	Plage de valeurs :	-2, -1, 1, 2	
	Signification :		
	(+)1	Front montant du palpeur de mesure 1 (à l'entrée de mesure 1)	
	-1	Front descendant du palpeur de mesure 1 (à l'entrée de mesure 1)	
	(+)2	Front montant du palpeur de mesure 2 (à l'entrée de mesure 2)	
	-2	Front descendant du palpeur de mesure 2 (à l'entrée de mesure 2)	
	Remarque : il existe au maximum 2 palpeurs de mesure (selon le niveau d'extension).		
G... :	Mode d'interpolation, p. ex. G0, G1, G2 ou G3		
X... Y... Z... :	Points finals en coordonnées cartésiennes		

Exemple

Code de programme	Commentaire
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; Bloc de mesure avec palpeur de mesure de la première entrée de mesure et interpolation linéaire. Un arrêt du prétraitement est automatiquement créé.
...	

Informations complémentaires

Etat de la tâche de mesure

Lorsqu'il est nécessaire de savoir, dans le programme, si le déclenchement du palpeur a eu lieu ou non, il est possible d'interroger la variable d'état $AC_MEA[<n>] (<n> = numéro du palpeur) :

Valeur	Signification
0	Tâche de mesure non exécutée
1	Tâche de mesure exécutée correctement (le déclenchement du palpeur a eu lieu)

Remarque

Si le palpeur est dévié dans le programme, la variable est mise à 1. Au début d'un bloc de mesure, la variable est mise automatiquement à l'état initial du palpeur.

Enregistrement des valeurs de mesure

Les positions sont mesurées pour tous les axes à interpolation et axes de positionnement d'un bloc, qui sont déplacés (le nombre maximal d'axes dépend de la configuration de la

commande). Avec `MEAS`, le déclenchement du palpeur entraîne un freinage défini du déplacement.

Remarque

Si un axe géométrique est programmé dans un bloc de mesure, les valeurs de mesure sont enregistrées pour tous les axes géométriques courants.

Si un axe concerné par une transformation est programmé dans un bloc de mesure, les valeurs de mesure sont enregistrées pour tous les axes concernés par cette transformation.

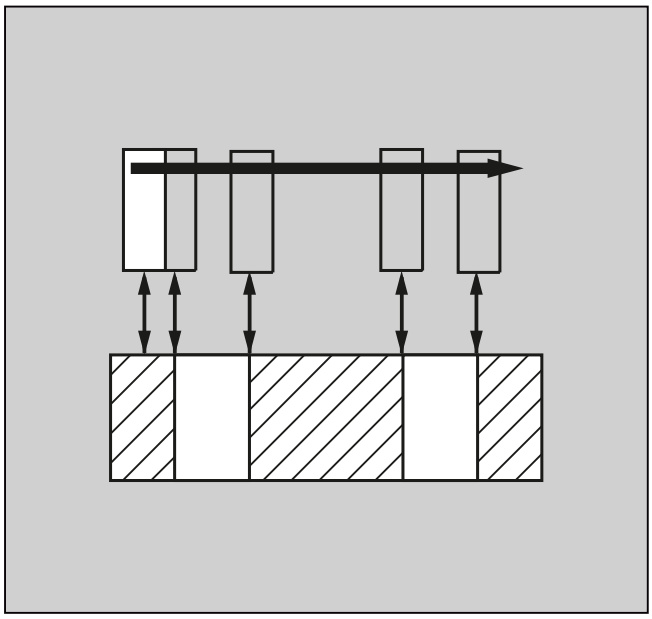
Lecture des résultats de mesure

Les résultats de mesure pour les axes enregistrés avec le palpeur de mesure peuvent être lus via les variables système suivantes :

- $AA_MM[<axe>]
  Résultats de mesure dans le système de coordonnées machine
- $AA_MW[<axe>]
  Résultats de mesure dans le système de coordonnées pièce

5.8 Mesure spécifique à l'axe (MEASA, MEAWA, MEAC) (option)

Plusieurs types de palpeur et plusieurs systèmes de mesure peuvent être utilisés pour la mesure spécifique à l'axe.



Les mots-clés `MEASA`, `MEAWE` et `MEAC` sont disponibles pour la programmation de la fonction.

`MEASA` et `MEAWA` permettent de saisir à chaque mesure jusqu'à quatre valeurs de mesure pour l'axe programmé et de les stocker dans des variables système en fonction de l'événement déclencheur.

`MEAC` permet d'effectuer des tâches de mesure continues. Les résultats des mesures sont stockés dans des variables FIFO.

Syntaxe

```
MEASA[<axe>]=(<mode>,<TE1>,...,<TE4>)
MEAWA[<axe>]=(<mode>,<TE1>,...,<TE4>)
MEAC[<axe>]=(<mode>,<mémoire de mesure>,<TE1>,...,<TE4>)
```

Remarque

`MEASA` et `MEAWA` ont un effet non modal et peuvent être programmées dans un bloc. Par contre, si vous programmez `MEASA`/`MEAWA` avec `MEAS`/`MEAW` dans un même bloc, un message d'erreur sera émis.

Signification

`MEASA` :	Mot-clé : Mesure spécifique à l'axe avec effacement de la distance restant à parcourir	
	Prise d'effet :	bloc par bloc (non modale)

MEAWA :	Mot-clé : Mesure spécifique à l'axe sans effacement de la distance restant à parcourir	
	Prise d'effet :	bloc par bloc (non modale)
MEAC :	Mot-clé : Mesure continue spécifique à l'axe sans effacement de la distance restant à parcourir	
	Prise d'effet :	bloc par bloc (non modale)
<axe> :	Nom de l'axe de canal utilisé pour la mesure	
<mode> :	Nombre à deux chiffres pour l'indication du mode de fonctionnement (mode de mesure et système de mesure)	
	Unité (mode de mesure) :	
	0	annulation de la tâche de mesure.
	1	Jusqu'à 4 événements déclencheurs différents, activables simultanément.
	2	Jusqu'à 4 événements déclencheurs différents, activables successivement
	3	Jusqu'à 4 événements déclencheurs activables successivement, cependant sans surveillance de l'événement déclencheur 1 au démarrage (les alarmes 21700/21703 sont inhibées). **Remarque :** ce mode n'est pas possible avec MEAC.
	Dizaine (système de mesure) :	
	0 (ou aucune indication)	système de mesure actif
	1	système de mesure 1
	2	système de mesure 2
	3	les deux systèmes de mesure
<TE> :	Evénement déclencheur de la mesure	
	Type :	INT
	Plage de valeurs :	-2, -1, 1, 2
	Signification :	
	(+)1	Front montant du palpeur 1
	-1	Front descendant du palpeur 1
	(+)2	Front montant du palpeur 2
	-2	Front descendant du palpeur 2
<mémoire de mesure> :	Numéro de la FIFO (mémoire à file d'attente)	

Exemples

Exemple 1 : Mesure spécifique à l'axe avec effacement de la distance restant à parcourir en mode 1 (évaluation dans l'ordre chronologique)

a) avec 1 système de mesure

Code de programme	Commentaire
...	

Code de programme	Commentaire
N100 MEASA[X] = (1,1,-1) G01 X100 F100	; Mesure en mode 1 avec système de mesure actif. Attente du signal de mesure avec front montant/descendant du palpeur 1 sur le trajet vers X=100.
N110 IF $AC_MEA[1]==FALSE GOTOF ENDE	; Vérification du succès de la mesure.
N120 R10=$AA_MM1[X]	; Mémorisation de la valeur de mesure correspondant au premier événement déclencheur programmé (front montant).
N130 R11=$AA_MM2[X]	; Mémorisation de la valeur de mesure correspondant au deuxième événement déclencheur programmé (front descendant).
N140 FIN :	

b) avec 2 systèmes de mesure

Code de programme	Commentaire
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; Mesure en mode 1 avec les deux systèmes de mesure. Attente du signal de mesure avec front montant/descendant du palpeur 1 sur le trajet vers X=100.
N210 IF $AC_MEA[1]==FALSE GOTOF ENDE	; Vérification du succès de la mesure.
N220 R10=$AA_MM1[X]	; Mémorisation de la valeur de mesure du système de mesure 1 sur front montant.
N230 R11=$AA_MM2[X]	; Mémorisation de la valeur de mesure du système de mesure 2 sur front montant.
N240 R12=$AA_MM3[X]	; Mémorisation de la valeur de mesure du système de mesure 1 sur front descendant.
N250 R13=$AA_MM4[X]	; Mémorisation de la valeur de mesure du système de mesure 2 sur front descendant.
N260 FIN :	

Exemple 2 : Mesure spécifique à l'axe avec effacement de la distance restant à parcourir en mode 2 (évaluation dans l'ordre de programmation)

Code de programme	Commentaire
...	
N100 MEASA[X] = (2,1,-1,2,-2) G01 X100 F100	; Mesure en mode 2 avec système de mesure actif. Attente du signal de mesure dans l'ordre suivant : front montant du palpeur 1, front descendant du palpeur 1, front montant du palpeur 2, front descendant du palpeur 2 sur le trajet vers X=100.
N110 IF $AC_MEA[1]==FALSE GOTOF MESSTASTER2	; Vérification du succès de la mesure avec palpeur 1.

Code de programme	Commentaire
N120 R10=$AA_MM1[X]	; Mémorisation de la valeur de mesure correspondant au premier événement déclencheur programmé (front montant palpeur de mesure 1).
N130 R11=$AA_MM2[X]	; Mémorisation de la valeur de mesure correspondant au deuxième événement déclencheur programmé (front montant palpeur de mesure 1).
N140 DETECTEUR2 :	
N150 IF $AC_MEA[2]==FALSE GOTOF ENDE	; Vérification du succès de la mesure avec palpeur 2.
N160 R12=$AA_MM3[X]	; Mémorisation de la valeur de mesure correspondant au troisième événement déclencheur programmé (front montant palpeur de mesure 2).
N170 R13=$AA_MM4[X]	; Mémorisation de la valeur de mesure correspondant au quatrième événement déclencheur programmé (front montant palpeur de mesure 2).
N180 FIN :	

Exemple 3 : Mesure continue spécifique à l'axe en mode 1 (évaluation dans l'ordre chronologique)

a) Mesure de 100 valeurs ou moins

Code de programme	Commentaire
...	
N110 DEF REAL VAL_MESURE[100]	
N120 DEF INT boucle=0	
N130 MEAC [X] = (1,1,-1) G01 X1000 F100	; Mesure en mode 1 avec système de mesure actif, mémorisation des valeurs de mesure dans $AC_FIFO1, attente du signal de mesure avec front descendant du palpeur 1 sur le trajet vers X=1000.
N135 STOPRE	
N140 MEAC[X]=(0)	; Abandon de la mesure dès l'accostage de la position d'axe.
N150 R1=$AC_FIFO1[4]	; Mémorisation du nombre de valeurs de mesure accumulées dans le paramètre R1.
N160 FOR boucle=0 TO R1-1	
N170 VAL_MESURE[boucle] = $AC_FIFO1[0]	; Lecture et mémorisation des valeurs de mesure à partir de $AC_FIFO1.
N180 ENDFOR	

b) Mesure avec effacement de la distance restant à parcourir après 10 valeurs de mesure

Code de programme	**Commentaire**
...	
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; Effacement de la distance restant à parcourir.
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC [X]=(0)	
N40 R1=$AC_FIFO1[4]	; Nombre de valeurs de mesure.
...	

c) Mesure d'un flanc de dent montant/descendant à l'aide de 2 palpeurs de mesure

Code de programme	**Commentaire**
...	
N110 DEF REAL VAL_MESURE[16]	
N120 DEF INT boucle=0	
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	; Mesure en mode 1 avec système de mesure actif, mémorisation des valeurs de mesure dans $AC_FIFO1, attente du signal de mesure dans l'ordre front descendant du palpeur 1, front montant du palpeur 2, sur le trajet vers X=100.
N140 STOPRE	; Arrêt du prétraitement
N150 MEAC[X]=(0)	; Abandon de la mesure dès l'accostage de la position d'axe.
N160 R1=$AC_FIFO1[4]	; Mémorisation du nombre de valeurs de mesure accumulées dans le paramètre R1.
N170 FOR boucle=0 TO R1-1	
N180 VAL_MESURE[boucle]=$AC_FIFO1[0]	; Lecture et mémorisation des valeurs de mesure à partir de $AC_FIFO1.
N190 ENDFOR	

Informations complémentaires

Tâche de mesure

La programmation d'une tâche de mesure peut avoir lieu dans le programme pièce ou à partir d'une action synchrone (voir chapitre "Actions synchrones (Page 641)"). On ne peut activer qu'une seule tâche de mesure par axe à un moment donné.

Remarque

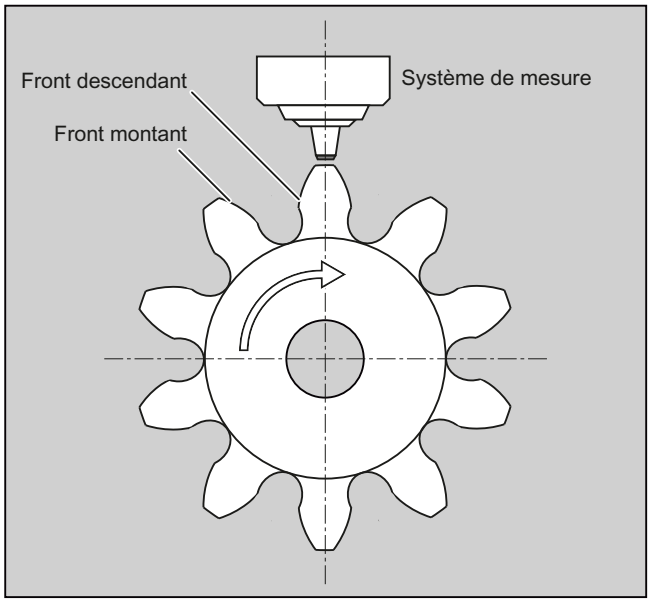
L'avance est à adapter au problème de mesure posé.

Pour `MEASA` et `MEAWA`, des résultats corrects sont garantis uniquement avec des avances pour lesquelles il ne survient pas plus d'un même événement déclencheur et pas plus de quatre événements déclencheurs différents par cycle de régulateur de position.

Pour la mesure continue avec `MEAC`, le rapport entre le cycle d'interpolation et le cycle du régulateur de position ne doit pas être supérieur à 1:8.

Evénement déclencheur

Un événement déclencheur se compose du numéro du palpeur de mesure et du critère de déclenchement du signal de mesure (front montant ou descendant).



Pour chaque mesure, le programme peut traiter respectivement jusqu'à 4 événements déclencheurs des palpeurs, autrement dit jusqu'à deux palpeurs avec deux fronts de mesure chacun. L'ordre du traitement et le nombre maximal de facteurs de déclenchement dépendent du mode sélectionné.

Remarque

Règle valable pour le mode de mesure 1 : un même événement de déclenchement ne doit être programmé qu'une seule fois dans chaque tâche de mesure !

Avec MEAC, le nombre de valeurs de mesure par événement déclencheur peut être augmenté à l'aide du télégramme PROFIBUS 395 à un total de 8 valeurs de mesure pour front montant et de 8 valeurs pour front descendant par événement déclencheur et cycle de régulateur de position.

- Un palpeur : 8 valeurs de mesure pour front montant et 8 pour front descendant
- Deux palpeurs : 4 valeurs de mesure pour front montant et 4 pour front descendant par palpeur

Le télégramme PROFIBUS 395 permet ainsi de réaliser des avances ou des vitesses plus élevées.

Bibliographie :
Description fonctionnelle Fonctions d'extension, Mesure (M5), chapitre : Mesure axiale

Mode de fonctionnement

Le premier chiffre (dizaines) du mode de fonctionnement permet de sélectionner le système de mesure souhaité. S'il n'existe qu'un seul système de mesure, alors que le second a été programmé, c'est automatiquement le système de mesure existant qui sera pris en compte.

Le second chiffre (unités) permet de sélectionner le mode de mesure souhaité. La procédé de mesure s'adapte ainsi aux possibilités de la commande respective :

- **Mode 1**
  Les événements déclencheurs sont traités dans l'ordre chronologique de leur apparition. Dans ce mode, quand on met en œuvre des cartes 6 axes, on ne peut programmer qu'un seul événement déclencheur ou alors, si on en programme plusieurs, il y a commutation automatique dans le mode 2 (sans avertissement préalable).
- **Mode 2**
  Les événements déclencheurs sont traités dans l'ordre de leur programmation.
- **Mode 3**
  Les événements déclencheurs sont traités dans l'ordre de leur programmation, cependant aucune surveillance de l'événement déclencheur 1 au démarrage n'est réalisée.

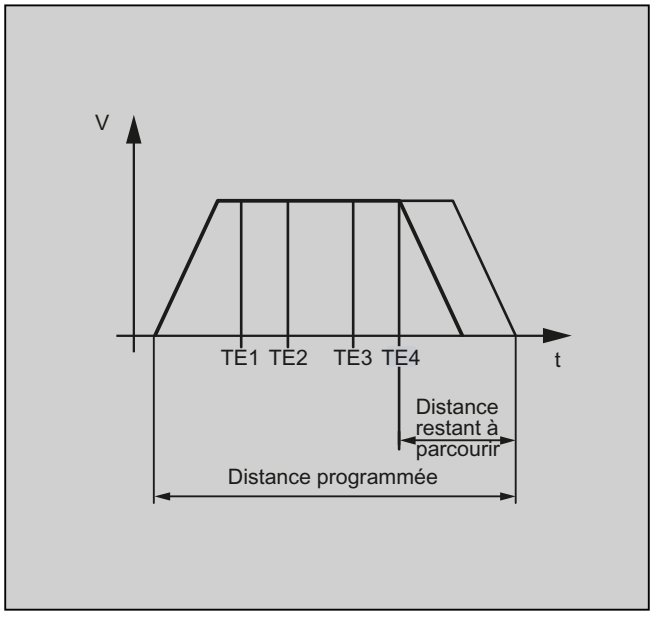
Remarque

Si vous travaillez avec 2 systèmes de mesure, seulement deux événements déclencheurs sont programmables.

Mesure avec et sans effacement de la distance restant à parcourir

Quand vous programmez MEASA, l'effacement de la distance restant à parcourir est effectué seulement après la saisie de toutes les valeurs de mesure requises.

Pour des tâches de mesure spéciales où la position programmée doit être accostée impérativement, utilisez la fonction MEAWA.



Remarque

`MEASA` n'est pas programmable dans des actions synchrones. Si on le souhaite, on peut programmer `MEAWA` plus effacement de la distance restant à parcourir en tant qu'action synchrone.

Quand la tâche de mesure est lancée avec `MEAWA` à partir d'une action synchrone, les valeurs de mesure sont disponibles uniquement dans le système de coordonnées machine.

Résultats de mesure pour MEASA, MEAWA

Les résultats de mesure sont disponibles sous les variables suivantes :

- dans le système de coordonnées machine :

`$AA_MM1[<axe>]`	Valeur de mesure du système de mesure programmé pour l'événement déclencheur 1
`...`	...
`$AA_MM4[<axe>]`	Valeur de mesure du système de mesure programmé pour l'événement déclencheur 4

- dans le système de coordonnées pièce :

`$AA_MW1[<axe>]`	Valeur de mesure du système de mesure programmé pour l'événement déclencheur 1
`...`	...
`$AA_MW4[<axe>]`	Valeur de mesure du système de mesure programmé pour l'événement déclencheur 4

Axes géométriques / transformations

Si la mesure axiale doit être lancée pour un axe géométrique, il convient de programmer de façon explicite la même tâche de mesure pour tous les autres axes géométriques. Il en va de même pour les axes qui participent à une transformation.

Exemples :
```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

ou
```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

Tâche de mesure avec 2 systèmes de mesure

Lorsqu'une tâche de mesure est exécutée avec deux systèmes de mesure, chacun des deux événements déclencheurs possibles est saisi par les deux systèmes de mesure de l'axe en question. Les variables réservées sont ainsi affectées :

`$AA_MM1[<axe>]`	ou	`$AA_MW1[<axe>]`	Valeur de mesure du système de mesure 1 pour l'événement déclencheur 1
`$AA_MM2[<axe>]`	ou	`$AA_MW2[<axe>]`	Valeur de mesure du système de mesure 2 pour l'événement déclencheur 1
`$AA_MM3[<axe>]`	ou	`$AA_MW3[<axe>]`	Valeur de mesure du système de mesure 1 pour l'événement déclencheur 2
`$AA_MM4[<axe>]`	ou	`$AA_MW4[<axe>]`	Valeur de mesure du système de mesure 2 pour l'événement déclencheur 2

Variables système

L'état du palpeur est disponible dans la variable système suivante :

$A_PROBE[<n>]

Valeur	Signification
1	Palpeur actionné
0	Palpeur non actionné

La limitation du palpeur est fournie par la variable système suivante :

$A_PROBE_LIMITED[<n>]

Valeur	Signification
1	Limitation du palpeur active
0	Limitation du palpeur inactive

<n> = palpeur de mesure

Bibliographie :
Tables de paramètres Variables système

Etat de la tâche de mesure MEASA, MEAWA

S'il est nécessaire de connaître, dans le programme, l'état de la tâche de mesure, il peut être interrogé avec $AC_MEA[<n>], où <n> = numéro du palpeur de mesure. Aussitôt que tous les événements déclencheurs du palpeur <n> programmés dans un bloc ont eu lieu, cette variable fournit la valeur 1. Sinon, elle fournit la valeur 0.

Remarque

Si la mesure est lancée à partir d'actions synchrones, $AC_MEA n'est plus actualisé. Dans ce cas, il convient d'interroger le nouveau signal d'interface de l'AP, DB31, … DBX62.3, ou la variable équivalente $AA_MEAACT[<axe>].

Signification :

$AA_MEAACT==1 : mesure activée

$AA_MEAACT==0 : mesure non activée

Mesure continue (MEAC)

Avec `MEAC` , les valeurs de mesure sont fournies dans le système de coordonnées machine et rangées dans la mémoire FIFO[n] indiquée (mémoire à file d'attente). Quand deux palpeurs ont été configurés pour la mesure, les valeurs de mesure du second palpeur seront rangées séparément dans la mémoire FIFO[n+1] configurée à cet effet (par le biais des PM).

La mémoire FIFO est une mémoire à file d'attente dans laquelle on enregistre les valeurs de mesure dans des variables $AC_FIFO suivant le principe "premier entré, premier sorti" (voir chapitre "Actions synchrones (Page 641)").

Remarque

Le contenu de la mémoire FIFO ne peut être lu qu'une seule fois. Si vous souhaitez pouvoir réutiliser les valeurs mesurées, mémorisez-les temporairement dans les données utilisateur.

Dès que le nombre de valeurs de mesure destinées à la mémoire FIFO dépasse le nombre maximum prévu dans les paramètres machine, il est mis fin automatiquement à la mesure.

Pour réaliser une mesure sans fin, programmez la lecture cyclique des valeurs de mesure. La lecture doit être effectuée au moins à la même fréquence que l'entrée de nouvelles valeurs de mesure.

Bibliographie :

- Description fonctionnelle Actions synchrones ; Description détaillée, chapitre : Paramètre ($AC_FIFO)
- Description fonctionnelle Fonctions d'extension ; Mesure (M5), chapitre : Mesure axiale

Protection contre les erreurs de programmation

Les erreurs de programmation suivantes sont détectées et signalées par un message d'erreur :

- MEASA/MEAWA programmées avec MEAS/MEAW dans un même bloc
  Exemple :
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA avec un nombre de paramètres <2 ou >5
  Exemple :
  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA avec événement déclencheur différent de 1/ -1/ 2/ -2
  Exemple :
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```
- MEASA/MEAWA avec mode incorrect
  Exemple :
  ```
  N01 MEAWA[B]=(4,1) B100
  ```
- MEASA/MEAWA avec événement déclencheur programmé deux fois
  Exemple :
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```
- MEASA/MEAWA et axe géométrique manquant
  Exemple :
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;axe GEOX/Y/
  Z
  ```
- Tâche de mesure hétérogène pour axes géométriques
  Exemple :
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50
  F100
  ```

5.9 Fonctions spéciales pour l'utilisateur OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

Adresses OEM

L'utilisateur OEM définit la signification des adresses OEM. La fonctionnalité est introduite par le biais de cycles de compilation. Cinq adresses OEM sont réservées (`OMA1` ... `OMA5`). Les descripteurs d'adresse peuvent être spécifiés. Les adresses OEM sont admises dans chaque bloc.

Appels d'instruction G réservés

Les appels d'instruction G suivantes sont réservés pour l'utilisateur OEM :

- `OEMIPO1`, `OEMIPO2` (à partir du groupe G 1)
- `G810` ... `G819` (groupe G 31)
- `G820` ... `G829` (groupe G 32)

La fonctionnalité est introduite par le biais de cycles de compilation.

Fonctions et sous-programmes

En outre, l'utilisateur OEM peut créer des fonctions et des sous-programmes prédéfinis avec transfert de paramètres.

Remarque

Simulation de pièce

Jusqu'à la version 4.4. du logiciel aucun cycle de compilation (CC) n'est pris en charge par la simulation de pièce. A partir de la version 4.4, seule une sélection de cycles de compilation est prise en charge.

Les instructions de langage du programme pièce provenant de cycles de compilation non pris en charge (`OMA1` ... `OMA5`, `OEMIPO1/2`, `G810` ... `G829`, procédures et fonctions utilisateur) entraînent donc, sans traitement individuel, la signalisation d'une alarme et l'interruption de la simulation.

Solution : Traiter individuellement les éléments de langage spécifiques CC manquants du programme pièce (requête $P_SIM).
Exemple :

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ; inactif lors de la simulation CC
ELSE
N5 X300 OMA1=10
ENDIF
```

5.10 Réduction de l'avance avec décélération aux angles (FENDNORM, G62, G621)

Lors de la décélération automatique aux angles, l'avance est réduite juste avant l'angle concerné selon une courbe en forme de cloche. En outre, l'étendue du comportement d'outil pertinent pour l'usinage peut être paramétrée au moyen des données de réglage. Ce sont :

- Début et fin de la réduction de l'avance
- Correction de vitesse avec laquelle l'avance est réduite
- Identification de l'angle pertinent

En tant qu'angles pertinents, on prend en considération les angles dont l'angle interne est inférieur à l'angle paramétré par l'intermédiaire de la donnée de réglage .

La valeur par défaut `FENDNORM` désactive la fonction de correction automatique aux angles.

Bibliographie :
/FBA/ Description fonctionnelle Dialectes ISO

Syntaxe

```
FENDNORM
G62 G41
G621
```

Signification

`FENDNORM` :	Décélération automatique aux angles désactivée
`G62` :	Décélération aux angles intérieurs avec correction du rayon d'outil activée
`G621` :	Décélération à tous les angles avec correction du rayon d'outil activée

G62 intervient seulement au niveau des angles internes avec

- correction active du rayon d'outil `G41`, `G42` et
- contournage actif `G64`, `G641`

L'angle correspondant est accosté avec l'avance réduite qui résulte de :

F * (correction de vitesse pour la réduction de l'avance) * correction de vitesse de l'avance

La réduction maximale de l'avance peut être atteinte lorsque l'outil doit assurer, par rapport à la trajectoire du centre de l'outil, le changement de direction vers l'angle concerné.

G621 intervient comme G62 au niveau de chaque angle formé par les axes définis avec `FGROUP`.

5.11 Critère programmable de fin de déplacement (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

De façon analogue aux critères de changement de bloc en interpolation de trajectoire (G601, G602 et G603), vous pouvez programmer un critère de fin de déplacement en interpolation d'axes individuels dans un programme pièce ou une action synchrone pour axes de commande/AP.

Selon le réglage du critère de fin de déplacement, les blocs de programme pièce ou de cycle technologique comportant des déplacements d'axes individuels se terminent plus ou moins rapidement. Cela vaut également pour l'AP via FC15/16/18.

Syntaxe

```
FINEA[<axe>]
COARSEA[<axe>]
IPOENDA[<axe>]
IPOBRKA(<axe>[,<instant>])
ADISPOSA(<axe>[,<mode>,<taille de fenêtre>])
```

Signification

`FINEA` :	Critère de fin de déplacement : "Arrêt précis fin"		
	Prise d'effet :	modale	
`COARSEA` :	Critère de fin de déplacement : "Arrêt précis grossier"		
	Prise d'effet :	modale	
`IPOENDA` :	Critère de fin de déplacement : "Arrêt interpolateur"		
	Prise d'effet :	modale	
`IPOBRKA` :	Critère de changement de bloc : Rampe de freinage		
	Prise d'effet :	modale	
`ADISPOSA` :	Fenêtre de tolérance pour le critère de fin de déplacement		
	Prise d'effet :	modale	
`<axe>` :	Nom d'axe de canal (X, Y....)		
`<instant>` :	Instant du changement de bloc en % de la rampe de freinage : • 100 % = début de la rampe de freinage • 0 % = fin de la rampe de freinage, même signification que `IPOENDA`		
	Type :	REAL	
`<mode>` :	Référence de la fenêtre de tolérance		
	Plage de valeurs :	0	Fenêtre de tolérance pas active
		1	Fenêtre de tolérance rapportée à la consigne de position
		2	Fenêtre de tolérance rapportée à la position réelle
	Type :	INT	
`<taille de fenêtre>` :	Taille de la fenêtre de tolérance		
	Type :	REAL	

Exemples

Exemple 1 : critère de fin de déplacement : "Arrêt interpolateur"

```
Code de programme
; déplacement axe de positionnement X sur 100, vitesse
200 m/min, accélération 90 %,
; critère de fin de déplacement : Arrêt interpolateur
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]


; Action synchrone :
; TOUJOURS LORSQUE : l'entrée 1 est à 1
; PUIS déplacement axe de positionnement X sur 50, vi-
tesse 200 m/min, accélération 140 %,
;      critère de fin de déplacement : Arrêt interpola-
teur
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]
```

Exemple 2 : Critère de changement de bloc : "Rampe de freinage"

Code de programme	Commentaire
	; Prise d'effet des paramètres par défaut
N40 POS[X]=100	; Mouvement de positionnement de X à la position 100
	Critère de changement de bloc : Arrêt précis fin
N20 IPOBRKA(X,100)	; Critère de changement de bloc : "Rampe de freinage",
	100 % = début de la rampe de freinage
N30 POS[X]=200	: Le changement de bloc a lieu dès que l'axe X commence à freiner
N40 POS[X]=250	; L'axe X ne continue pas à freiner à la position 200 mais continue à se déplacer jusqu'à la position 250.
	Dès que l'axe commence à freiner, le changement de bloc intervient.
N50 POS[X]=0	; L'axe X freine et retourne à la position 0.
	Le changement de bloc intervient à la position 0 avec "Arrêt précis fin"
N60 X10 F100	; L'axe X se déplace, en tant qu'axe à interpolation, vers la position 10

Informations complémentaires

Variable système pour le critère de fin de déplacement

Le critère de fin de déplacement actif peut être lu via la variable système $AA_MOTEND.

Bibliographie : /LIS2sl/ Tables de paramètres, volume 2

Critère de changement de bloc : "Rampe de freinage" (IPOBRKA)

Lors de l'activation du critère de changement de bloc "Rampe de freinage", si une valeur a été programmée pour l'instant de changement de bloc optionnel, cette valeur prend effet pour le mouvement de positionnement suivant et elle est écrite dans la donnée de réglage, de façon synchrone à l'exécution. Si aucune valeur n'a été indiquée pour l'instant de changement de bloc, la valeur actuelle de la donnée de réglage prend effet.

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Lors de la prochaine programmation d'un critère de fin de déplacement axial (`FINEA`, `COARSEA`, `IPOENDA`), `IPOBRKA` est désactivé pour l'axe correspondant.

Critère de changement de bloc supplémentaire : "Fenêtre de tolérance" (ADISPOSA)

La fenêtre de tolérance `ADISPOSA` permet de définir, en tant que critère de changement de bloc supplémentaire, une fenêtre de tolérance autour du point de fin de bloc (au choix, position réelle ou position de consigne). Les deux conditions doivent alors être remplies pour le changement de bloc :

- Critère de changement de bloc : "Rampe de freinage"
- Critère de changement de bloc : "Fenêtre de tolérance"

Bibliographie

Pour plus d'informations sur le critère de changement de bloc des axes de positionnement, référez-vous à la :

- Description fonctionnelle Fonctions d'extension ; axes de positionnement (P2)
- Manuel de programmation Notions de base, chapitre "Régulation par avance"

Transformations des coordonnées (frames) 6

6.1 Transformation de coordonnées par variables frames

Outre les instructions décrites dans le manuel de programmation Notions de base, chapitre "Transformations des coordonnées (Frames)", comme `ROT`, `AROT`, `SCALE`, etc., le système de coordonnées pièce (SCP) peut également être transformé via les variables frame $P_...FR (frames de gestion des données) et $P_...FRAME (frames actifs).

La figure suivante donne un aperçu de la structuration des variables frame :

- Frames de gestion des données
- Frames actifs
- Frame global actif : concaténation de tous les frames actifs
- Frames à définition globale pour NCU
- Frames spécifiques à un canal

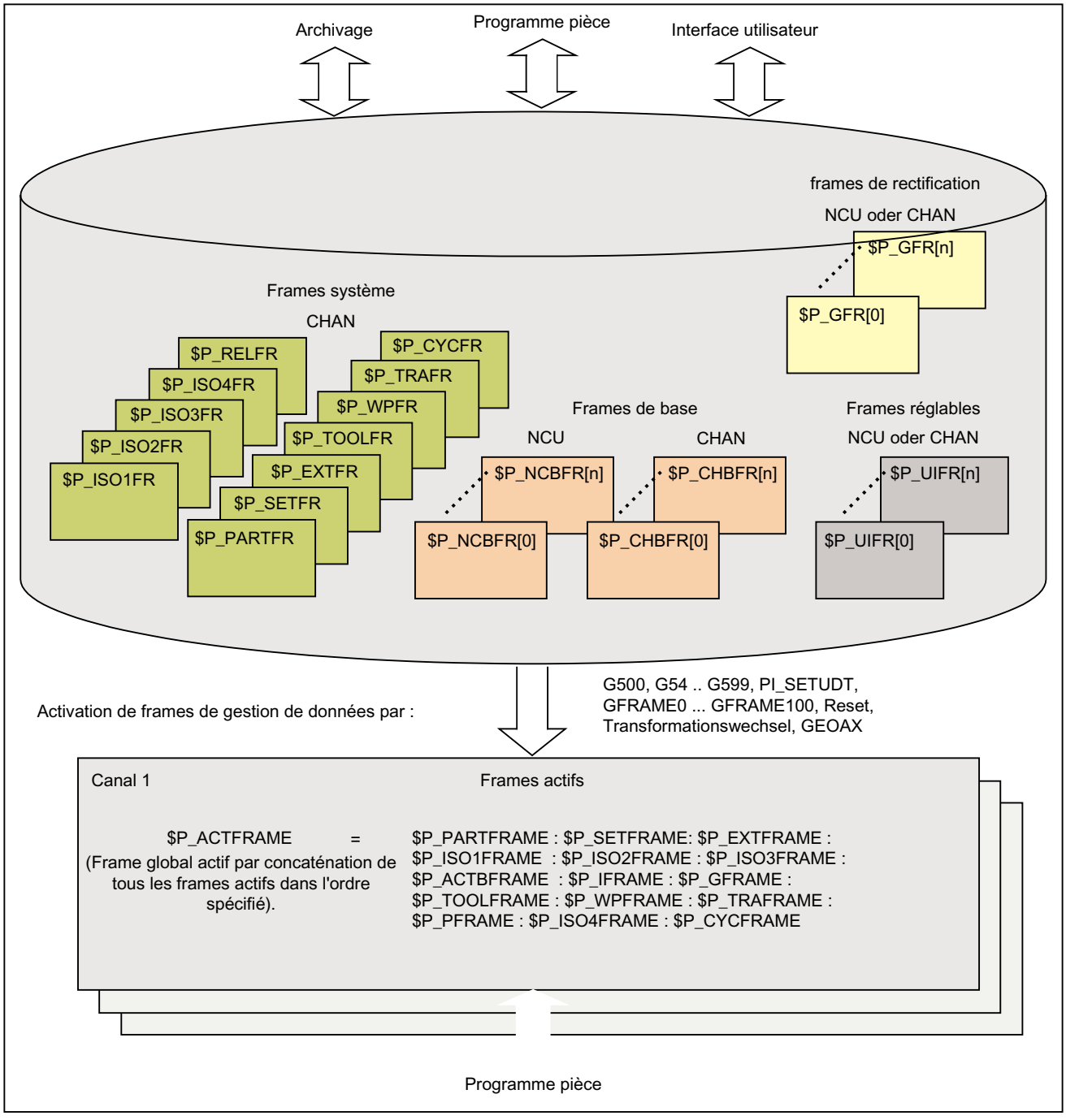


Figure 6-1 Vue d'ensemble des variables frame

6.1.1 Variable frame prédéfinie ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)

Coché : frames de base spécifiques à un canal $P_CHBFRAME[<n>] ($P_BFRAME)

Remarque

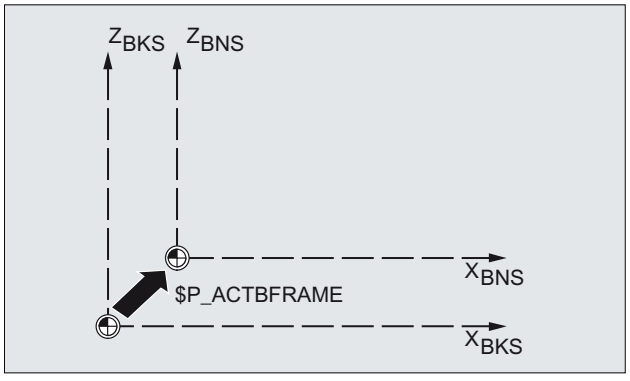
Le frame de base actuel `$P_BFRAME` et le frame de base de gestion des données `$P_UBFR` sont conservés pour des raisons de compatibilité.

- `$P_BFRAME` □ `$P_CHBFRAME[0]`
- `$P_UBFR` □ `$P_CHBFR[0]`.

Les variables frame $P_CHBFRAME[<n>] définissent la corrélation entre le système de coordonnées de base (SCB) et le système de coordonnées de base réglable (SBR).

Si le frame de base spécifique à un canal actuel $P_CHBFRAME[`<n>`] doit être immédiatement effectif dans le programme CN, les possibilités suivantes sont disponibles

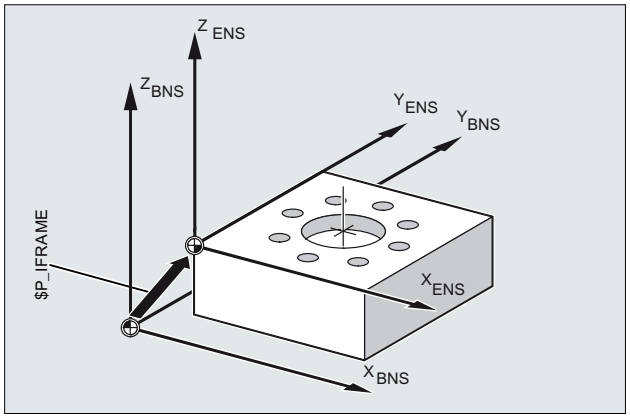
- Instructions :
  - `G500` (désactivation de tous les frames réglables, les frames de base restent actifs)
  - `G54` ... `G599` (décalages d'origine réglables)
- Affectation d'un frame de base spécifique à un canal de la gestion des données à un frame de base spécifique à un canal actuel :
  `$P_CHBFRAME[<n>] = $P_CHBFR[<m>]`



Coché : frame réglable spécifique à un canal $P_IFRAME

La variable frame $P_IFRAME définit la corrélation entre le système de coordonnées de base réglable (SBR) et le système de coordonnées pièce réglable (SPR).

- `$P_IFRAME` correspond à `$P_UIFR[$P_IFRNUM]`
- `$P_IFRAME` contient, après la programmation de `G54` par exemple, la translation, la rotation, l'homothétie et la fonction miroir définies par `G54`.

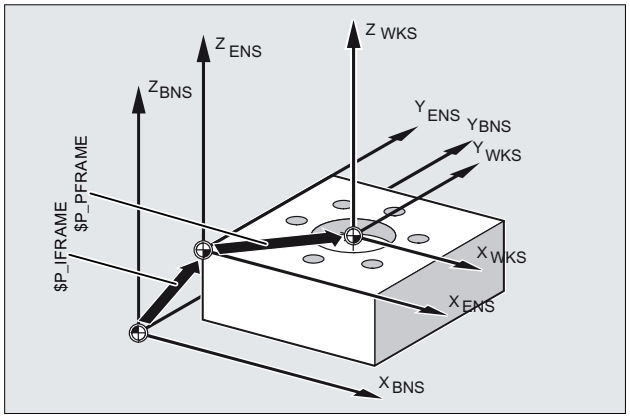


Coché : frame programmable spécifique à un canal $P_PFRAME

La variable frame $P_PFRAME définit la relation entre le système de coordonnés réglable (SCR) et le système de coordonnées-pièce (SCP).

`$P_PFRAME` contient le frame résultant, obtenu

- **à partir de la programmation** de `TRANS/ATRANS`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR` ou
- **à partir de l'affectation** de `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` au frame programmable.



Coché : frame global $P_ACTFRAME

Le frame global effectif dans un canal résulte de la concaténation de tous les frames actifs dans un canal.

```
$P_ACTFRAME =    $P_PARTFRAME : $P_SETFRAME  : $P_EXTFRAME :
                 $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :
                 $P_ACTBFRAME : $P_IFRAME    : $P_GFRAME :
                 $P_TOOLFRAME : $P_WPFRAME   : $P_TRAFRAME :
                 $P_PFRAME    : $P_ISO4FRAME : $P_CYCFRAME
```

`$P_ACTFRAME` décrit l'origine du système de coordonnées pièce courant.

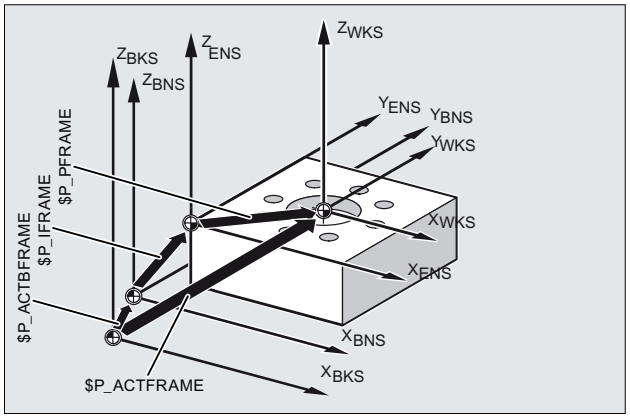
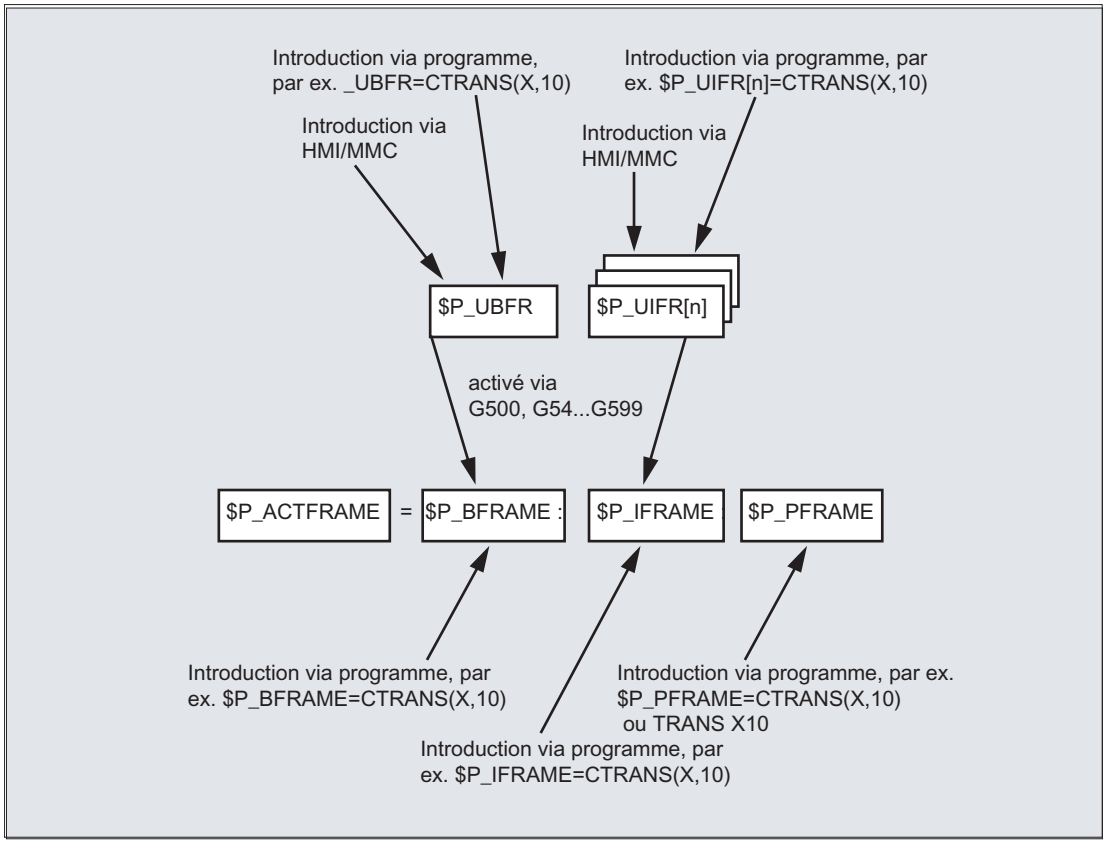


Figure 6-2 Variable frame $P_ACTFRAME

Si l'un des frames suivants `$P_BFRAME` / `$P_CHBFRAME[<n>]`, `$P_IFRAME` ou `$P_PFRAME` est modifié, le frame global courant `$P_ACTFRAME` est recalculé.



L'activation du frame de base ou du frame réglable après un reset dépend du réglage du PM 20110 RESET_MODE_MASK :

bit0=1, bit14=1 --> `$P_UBFR` (frame de base) activé

bit0=1, bit5=1 --> `$P_UIFR[$P_UIFRNUM]` (frame réglable) activé

Gestion des données : frames de base spécifiques à un canal $P_CHBFR[<n>]

Les frames de base de gestion des données sont lus/écrits via les variables frame $P_CHBFR[<n>]. La procédure d'écriture n'active pas immédiatement le frame de gestion des données dans le canal. L'activation du frame écrit intervient comme suit :

- Reset de canal et PM20110 $MC_RESET_MODE_MASK, Bit0 == 1 et Bit14 == 1
- Instruction `G500`, `G54` ... `G57`, `G505` ... `G599` (activation/désactivation des frames de base puis nouveau calcul du frame global courant)

Gestion des données : Frames réglables spécifiques à un canal $P_UIFR[<n>]

Les frames réglables de gestion des données sont lus/écrits via les variables frame $P_UIFR[<n>]. La procédure d'écriture n'active pas immédiatement le frame dans le canal. La prise en compte du frame écrit dans le canal intervient comme suit :

- Instruction `G500` (désactivation de tous les frames réglables ou décalages d'origine)
- Instruction `G54` ... `G57`, `G505` ... `G599` (activation d'un frame réglable ou d'un décalage d'origine)

Frame réglable actif	Frame de gestion des données	(correspond à l'instruction)
`$P_IFRAME` =	`$P_UIFR[0]`	`G500`
	`$P_UIFR[1]`	`G54`
	`$P_UIFR[2]`	`G55`
	`$P_UIFR[3]`	`G56`
	`$P_UIFR[4]`	`G57`
	`$P_UIFR[5]`	`G505`
	`$P_UIFR[6]`	`G506`
	...	...
	`$P_UIFR[99]`	`G599`

6.2 Affectations de valeurs aux frames

6.2.1 Affectation de valeurs directes (valeur d'axe, angle, échelle)

Vous pouvez affecter directement des valeurs à des frames ou à des variables frame dans le programme CN.

Syntaxe

Syntaxe

```
$P_PFRAME = CTRANS(X, <valeur du décalage>, Y, <valeur du décalage>,
Z, <valeur du décalage>, ...)
$P_PFRAME = ROT(X, <angle>, Y, <angle>, Z, <angle>, ...)
$P_UIFR[..] = CROT(X, <angle>, Y, <angle>, Z, <angle>,  ...
$P_PFRAME = CSCALE(X, <échelle>, Y, <échelle>, Z, <échelle>, ...)
$P_PFRAME = CMIRROR(X, Y, Z)
```

La syntaxe pour `$P_CHBFRAME[<n>]` est identique à `$P_PFRAME`.

Signification

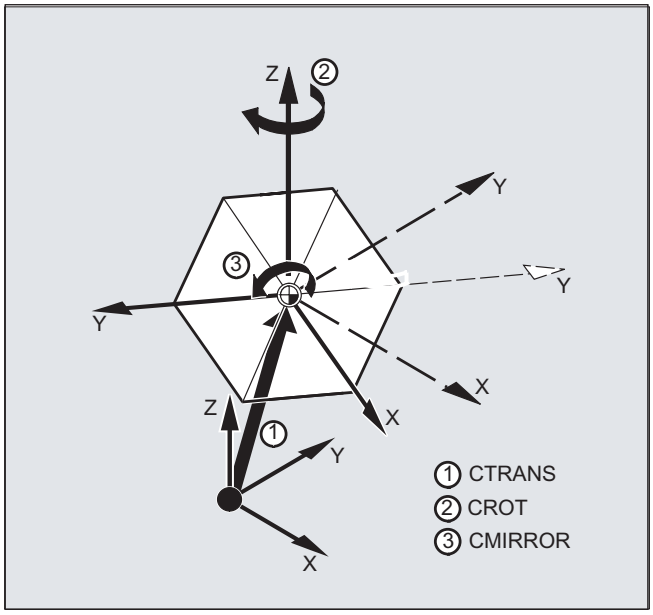
`CTRANS` :	Décalage (translation) dans les axes indiqués
`CROT` :	Rotation autour des axes indiqués
`CSCALE` :	Changement d'échelle dans les axes indiqués
`CMIRROR` :	Inversion du sens de l'axe indiqué
`X, Y, Z` :	Valeur du décalage en direction de l'axe géométrique indiqué
`<valeur du décalage>` :	Valeur du décalage
`<angle>` :	angle selon lequel s'effectue la rotation
`<échelle>` :	indication de l'échelle

Exemples

Affectations de valeurs à des composantes du frame programmable courant

Affectation de valeur aux composantes Translation, Rotation et Fonction miroir du frame programmable courant :

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) : CROT(Z,45) : CMIRROR(Y)
```



Ecrire les composantes de rotation d'un frame

Affectation de valeurs aux trois axes de la composante Rotation du frame de gestion des données réglable `$P_UIFR` avec `CROT` :

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

Affectation directe des valeurs individuelles à chaque axe de la composante Rotation du frame de gestion des données comme alternative :

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

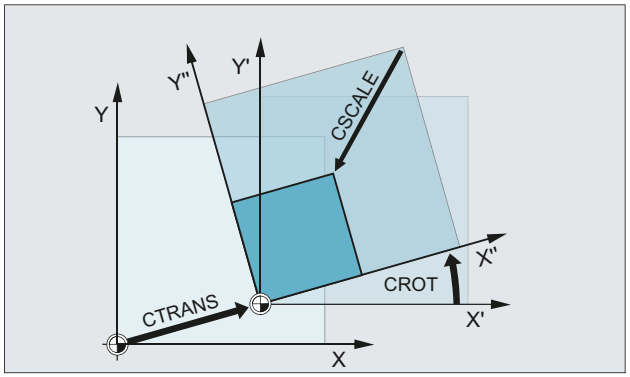
Description

Plusieurs opérations sur un frame peuvent être liées entre elles par l'opérateur de concaténation `:`. Les opérations sont alors exécutées l'une après l'autre de gauche à droite.

Exemple

Opérations concaténées sur $P_PFRAME avec décalage, rotation et homothétie :

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```



6.2.2 Lecture et modification de composantes de frame (TR, FI, RT, SC, MI)

Vous avez la possibilité d'accéder à des données **individuelles** d'un frame, par exemple à une valeur de décalage ou à un angle de rotation bien défini. Vous pouvez modifier ces valeurs ou les affecter à une autre variable.

Syntaxe

`R10=$P_UIFR[$P_UIFNUM,X,RT]`	L'angle de rotation RT autour de l'axe X issu du décalage d'origine réglable actuellement valide $P_UIFRNUM doit être affecté à la variable R10.
`R12=$P_UIFR[25,Z,TR]`	La valeur de décalage TR dans Z, issue du bloc de données du frame nº 25 réglé, doit être affectée à la variable R12.
`R15=$P_PFRAME[Y,TR]`	La valeur de décalage TR en Y du frame programmable courant doit être affectée à la variable R15.
`$P_PFRAME[X,TR] = 25`	La valeur de décalage TR en X du frame programmable courant doit être modifiée. X25 s'applique dès à présent.

Signification

`$P_UIFRNUM:`	`Cette variable établit automatiquement le lien avec le décalage d'origine réglable actuellement valide.`
`P_UIFR[n,…,…]:`	`En indiquant le n° de frame n, vous accédez au frame réglable n° n`
	`Indication de la composante qui doit être lue ou modifiée :`
`TR:`	`TR Translation`
`FI:`	`FI Translation Fine`
`RT:`	`RT Rotation`
`SC:`	`SC Scale Changement d'échelle`

`MI:`	`MI Fonction miroir`
`X, Y, Z:`	`Est indiqué en outre (voir exemples) l'axe X, Y, Z correspondant.`

Plage de valeurs pour la rotation RT :

Rotation autour du 1er axe géométri- -180° à +180°
que :
Rotation autour du 2e axe géométri- -90° à +90°
que :
Rotation autour du 3e axe géométri- -180° à +180°
que :

Description

Appeler le frame

En indiquant la variable système $P_UIFRNUM, vous pouvez accéder directement au décalage d'origine courant, réglé avec $P_UIFR ou G54, G55, …
($P_UIFRNUM contient le n° du frame courant réglé).

Vous appelez tous les autres frames réglables mémorisés $P_UIFR en indiquant le n° correspondant $P_UIFR[n].

Pour les variables frame prédéfinies et les frames que vous avez définis vous-même, indiquez le nom, par exemple $P_IFRAME.

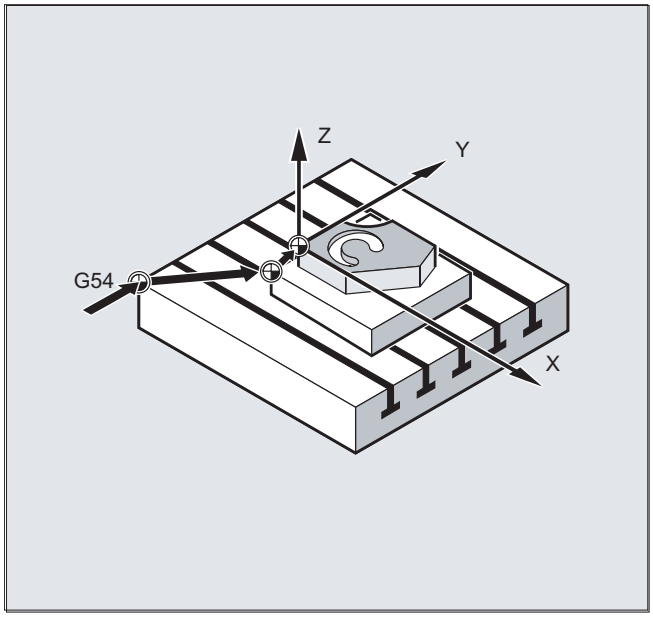
Appeler les données

Le nom de l'axe et la composante de frame à laquelle vous voulez accéder ou que vous voulez modifier figurent entre crochets droits, par ex. [X, RT] ou [Z, MI].

6.2.3 Calcul avec des frames

Dans le programme CN, un frame peut être affecté à un autre frame ou bien des frames peuvent être concaténés.

On utilise la concaténation de frames pour décrire, par exemple, plusieurs pièces disposées sur une palette et qui doivent être usinées en un seul cycle.



Pour la description des tâches d'usinage sur palette, les composantes de frame pourraient, par exemple, ne contenir que certaines valeurs partielles dont la concaténation pourraient générer différentes origines pièce.

Exemples

Affectations

Code de programme	**Commentaire**
`DEF FRAME REGLAGE_1`	`; Définition d'une variable frame locale`
`REGLAGE_1 = CTRANS(X,10)`	`; Affectation du résultat d'une fonction à la variable frame`
`$P_PFRAME = REGLAGE_1`	`; Affectation des variables frame au frame courant`
`DEF FRAME REGLAGE_4`	`; Définition d'une variable frame locale`
`REGLAGE_4 = $P_PFRAME`	`; Mise en mémoire tampon du frame courant dans les variables frame`
`...`	
`$P_PFRAME = REGLAGE_4`	`; Lecture du frame courant depuis les variables frame`

Concaténations

Les frames sont liés les uns aux autres dans l'ordre programmé par l'opérateur `:`. Les composantes de frame, comme les décalages, les rotations, etc., sont exécutées de manière additive les unes après les autres.

Code de programme	Commentaire

`$P_IFRAME = $P_UIFR[15] : $P_UIFR[16]`	`; Affectation du frame de résultat depuis la concaténation des`
	`; deux frames réglables de gestion de données au`
	`; frame global réglable actif.`
	`; Exemple d'application :`
	`; $P_UIFR[15] : Décalage`
	`; $P_UIFR[16] : Rotation`

`$P_UIFR[3] = $P_UIFR[4] : $P_UIFR[5]`	`; Affectation du frame de résultat depuis la concaténation des`
	`; deux frames réglables de gestion de données à un`
	`; autre frame réglable de gestion de données`

6.2.4 Définition des variables frame (DEF FRAME)

Outre les variables frame prédéfinies, il est également possible de définir des variables frame propres. Les variables frame propres sont des variables utilisateur de type FRAME. Le nom du frame peut être attribue librement dans le cadre des règles pour les variables utilisateur.

Avec les fonctions CTRANS, CROT, CSCALE, CMIRROR, les variables frame propres peuvent être affectées à des valeurs.

Syntaxe

```
DEF FRAME <nom>
```

Signification

`DEF FRAME` :	définition d'une variable utilisateur de type FRAME.
<Nom> :	nom de la variable frame

Exemple

Définition d'une variable frame "PALETTE" et affectation de valeurs de décalage et de rotation :

Code de programme	Commentaire
`DEF FRAME PALETTE`	`; Définir la variable frame PALETTE`
`PALETTE = CTRANS(...) : CROT(...)`	`; Affectation du frame de résultat de la concaténation du` `; décalage et de la rotation à la variable frame PALETTE`

6.3 Décalage grossier et fin (CTRANS, CFINE)

Décalage fin

Un décalage fin `CFINE(...)` peut être appliqué aux frames suivants :

- Frames réglables : $P_UIFR ou $P_IFRAME
- Cadres de base : $P_NCBFR[<n>], $P_CHBFR[<n>] / $P_CHBFRAMES[<n>] ou $P_ACTBFRAME
- Frame programmable : $P_PFRAME

Le décalage fin d'un frame se programme à l'aide de l'instruction `CFINE(...)`.

Décalage grossier

Le décalage grossier `CTRANS(...)` peut être appliqué à tous les frames.

Décalage total

Le décalage total découle de l'addition des décalages grossier et fin.

Paramètres machine

Déblocage du décalage fin

Le décalage fin est débloqué avec le paramètre machine :

PM18600 $MN_MM_FRAME_FINE_TRANS = 1

Syntaxe

Décalage fin

- Frame global
  - `<frame> = CFINE(<K_1>,<valeur>)`
  - `<frame> = CFINE(<K_1>,<valeur>, <K_2>, <valeur>)`
  - `<frame> = CFINE(<K_1>,<valeur>, <K_2>, <valeur>, <K_3>, <valeur>)`
- Composante de frame
  - `<frame>[<n>, <K_1>, FI] = <valeur>`

Décalage grossier

- Frame global
  - `<frame> = CTRANS(<K_1>,<valeur>)`
  - `<frame> = CTRANS(<K_1>,<valeur>, <K_2,<valeur>)`
  - `<frame> = CTRANS(<K_1>,<valeur>, <K_2,<valeur>, <K_3,<valeur>)`
- Composante de frame
  - `<frame>[<n>,<K_1>,TR] = <valeur>`

Spécialement pour le frame programmable `$P_PFRAME` :

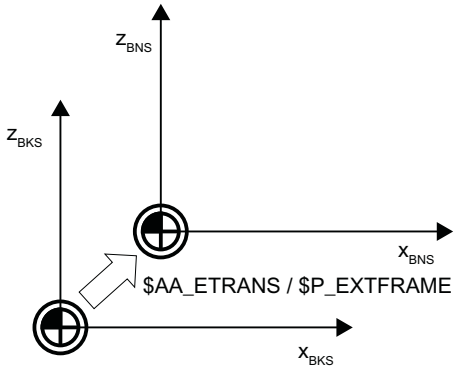
- `TRANS <K_1> <valeur>`
- `TRANS <K_1> <valeur> <K_2> <valeur>`
- `TRANS <K_1> <valeur> <K_2> <valeur> <K_3> <valeur>`

Signification

`<frame>` :	Frame, par exemple frame réglable de la gestion de données `$P_UIFR[<n>]`
`CFINE` :	Décalage fin, décalage additif.
`CTRANS` :	Décalage grossier, décalage absolu.
`TRANS` :	Uniquement frame programmable : Décalage grossier, décalage absolu.
`<K_n>` :	Axes de coordonnées X, Y, Z
`<valeur>` :	Valeur du décalage

6.4 Décalage d'origine externe ($AA_ETRANS)

Le décalage d'origine externe est un décalage linéaire entre le système de coordonnées de base (SCB) et le système de coordonnées de base (SBR).



Le décalage d'origine externe via $AA_ETRANS agit de deux manières, selon les paramètres machine :

1. La variable système $AA_ETRANS agit, après activation par le signal d'interface CN/AP, directement en tant que valeur de décalage
2. La valeur de la variable système $AA_ETRANS est reprise, après activation par le signal d'interface CN/AP, dans les frames système actifs $P:EXTFRAME et dans le frame de gestion de données $P_EXTFR. Enfin, le frame global actif $P_ACTFRAME est recalculé.

Paramètres machine

Concernant la variable système $AA_ETRANS, il convient de distinguer deux procédures sélectionnées via le paramètre machine suivant :

PM28082 $MC_MM_SYSTEM_FRAME_MASK,Bit1 = <valeur>

<Valeur>	Signification
0	Fonction : Ecriture directe de $AA_ETRANS[<axe>] par l'AP, l'IHM ou le programme CN. Déblocage de l'exécution du décalage d'origine de $AA_ETRANS[<axe>] dans le prochain bloc de déplacement possible : DB31, ... DBX3.0
1	Fonction : Activation du frame système actif $P:EXTFRAME et du frame de gestion de données $P_EXTFR Déblocage de l'exécution du décalage d'origine de $AA_ETRANS[<axe>] via : DB31, ... DBX3.0. Il se produit ensuite dans le canal : • arrêt de tous les déplacements dans le canal (hors axes d'ordre et d'AP) • Arrêt du prétraitement des blocs avec réorganisation (STOPRE) • Décalage grossier du frame actif $P_EXTFRAME[<axe>] = $AA_ETRANS[<axe>] • Décalage grossier du frame de gestion des données $P_EXTFR[<axe>] = $AA_ETRANS[<axe>] • Nouveau calcul du frame global actif $P_ACTFRAME • Exécution du décalage dans les axes programmés. • Poursuite du déplacement interrompu ou du programme CN

Programmation

- Syntaxe
  `$AA_ETRANS[<axe>] = <valeur>`
- Signification

`$AA_ETRANS` :	variable système pour mise en mémoire tampon du décalage d'origine externe
`<axe>` :	Axe de canal
`<valeur>` :	Valeur du décalage

Signal d'interface CN/AP

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<axe>] = $P_EXTFR[<axe>] = $AA_ETRANS[<axe>]

6.5 Forçage de valeurs réelles avec perte de l'état de référencement (PRESETON)

La procédure `PRESETON()` définit, pour un ou plusieurs axes, une nouvelle valeur réelle dans le système de coordonnées machine (SCM). Cela correspond à un décalage d'origine du SCM de l'axe. Ainsi, l'axe ne se déplace pas.

`PRESETON` déclenche un arrêt du prétraitement des blocs avec synchronisation. La position réelle est affectée à l'axe à l'arrêt seulement.

Si, pour `PRESETON`, l'axe n'est pas affecté au canal, la procédure suivante dépend de la configuration spécifique à l'axe de la procédure de permutation d'axe :

PM30552 $MA_AUTO_GET_TYPE

Etat de référencement

La définition d'une nouvelle valeur réelle dans le système de coordonnées machine remet à zéro l'état de référencement de l'axe machine :

DB31, ... DBX60.4 / .5 = 0 (référencé / synchronisé système de mesure 1 / 2)

Il est recommandé de n'utiliser `PRESETON` qu'avec des axes sans point de référence obligatoire.

Pour rétablir le système de coordonnées machine d'origine, le système de mesure de l'axe machine doit être de nouveau référencé, par exemple par accostage du point de référence depuis le programme pièce (`G74`).

⚠ PRUDENCE

Perte de l'état de référencement

La définition d'une nouvelle valeur réelle dans le système de coordonnées machine avec `PRESETON` remet l'état de référencement de l'axe machine sur "non référence / synchronisé".

Programmation

Syntaxe

```
PRESETON(<axe_1>, <valeur_1> [, <axe_2>, <valeur_2>, ... <axe_8>,
<valeur_8>])
```

Signification

`PRESETON` :	Forçage de valeurs réelles avec perte de l'état de référencement	
	Arrêt du prétraitement des blocs :	oui
	Uniquement dans le bloc :	oui
`<axe_x>` :	Nom d'axe machine	
	Type :	AXIS
	Plage de valeurs :	noms de machine défini dans le canal

<valeur_x> :	Nouvelle valeur réelle de l'axe machine dans le système de coordonnées machine (SCM) La saisie s'effectue dans le système de mesure actuellement en vigueur (inch/ métrique) Une programmation active de diamètre (DIAMON) est prise en compte	
	Type :	REAL

Bibliographie

PRESETONS dans les programmes CN

Une description détaillée de PRESETON dans les programmes CN se trouve dans :

Description fonctionnelle Fonctions de base ; chapitre "K2 : Axes, systèmes de coordonnées, frames" > "Systèmes de coordonnées" > "Système de coordonnées machine (SCM)" > "Forçage des valeurs réelles avec perte de l'état de référencement (PRESETON)"

PRESETONS dans des actions synchrones

Une description détaillée de PRESETON dans les actions synchrones se trouve dans :

Description fonctionnelle Actions synchrones, chapitre : "Description détaillée" > "Actions dans des actions synchrones" > "Forçage des valeurs réelles avec perte de l'état de référencement (PRESETON)"

6.6 Forçage de valeurs réelles sans perte de l'état de référencement (PRESETONS)

La procédure `PRESETONS()` définit, pour un ou plusieurs axes, une nouvelle valeur réelle dans le système de coordonnées machine (SCM). Cela correspond à un décalage d'origine du SCM de l'axe. Ainsi, l'axe ne se déplace pas.

`PRESETONS` déclenche un arrêt du prétraitement des blocs avec synchronisation. La position réelle est affectée à l'axe à l'arrêt seulement.

Si, pour `PRESETONS`, l'axe n'est pas affecté au canal, la procédure suivante dépend de la configuration spécifique à l'axe de la procédure de permutation d'axe :

PM30552 $MA_AUTO_GET_TYPE

Etat de référencement

La définition d'une nouvelle valeur réelle dans le système de coordonnées machine (SCM) avec `PRESETONS` ne modifie **pas** l'état de référencement de l'axe machine.

Conditions

- **Type de capteur**
  `PRESETONS` n'est possible que pour les types de codeurs suivants du système de mesure actif :
  - PM30240 $MA_ENC_TYPE[<système de mesure>] = 0 (codeur simulé)
  - PM30240 $MA_ENC_TYPE[<système de mesure>] = 1 (codeur de signal brut)
- **Mode de référencement**
  `PRESETONS` n'est possible que pour les modes de référencement suivants du système de mesure actif :
  - PM34200 $MA_ENC_REFP_MODE[<système de mesure>] = 0 (aucun accostage du point de référence possible)
  - PM34200 $MA_ENC_REFP_MODE[<système de mesure>] = 1 (référencement de systèmes de mesure relatifs, rotatifs ou linéaires : impulsion top zéro sur la voie du codeur)

Programmation

Syntaxe

```
PRESETONS(<axe_1>, <valeur_1> [, <axe_2>, <valeur_2>, ... <axe_8>,
<valeur_8>])
```

Signification

`PRESETONS` :	Forçage de valeurs réelles sans perte de l'état de référencement	
	Arrêt du prétraitement des blocs :	oui
	Uniquement dans le bloc :	oui

<axe_x> :	Nom d'axe machine	
	Type :	AXIS
	Plage de valeurs :	noms de machine défini dans le canal
<valeur_x> :	Nouvelle valeur réelle courante de l'axe machine dans le système de coordonnées machine (SCM) La saisie s'effectue dans le système de mesure actif (inch/métrique) Une programmation active de diamètre (DIAMON) est prise en compte	
	Type :	REAL

Bibliographie

PRESETONS dans les programmes CN

Une description détaillée de PRESETONS dans les programmes CN se trouve dans :

Description fonctionnelle Fonctions de base ; chapitre "K2 : Axes, systèmes de coordonnées, frames" > "Systèmes de coordonnées" > "Système de coordonnées machine (SCM)" > "Forçage des valeurs réelles sans perte de l'état de référencement (PRESETONS)"

PRESETONS dans des actions synchrones

Une description détaillée de PRESETONS dans les actions synchrones se trouve dans :

Description fonctionnelle Actions synchrones, chapitre : "Description détaillée" > "Actions dans des actions synchrones" > "Forçage des valeurs réelles sans perte de l'état de référencement (PRESETONS)"

6.7 Calcul du frame à partir de 3 points mesurés dans l'espace (MEAFRAME)

La fonction MEAFRAME s'utilise en assistance des cycles de mesure Elle calcule le frame à partir de trois points idéaux et des points mesurés correspondants.

Lorsqu'une pièce est positionnée pour l'usinage, sa position dans le système de coordonnées cartésiennes machine est en général décalée et pivotée par rapport à sa position idéale. Pour un usinage ou une mesure précise, un réglage physique onéreux ou une modification des déplacements dans le programme pièce est nécessaire.

Un frame peut être déterminé par palpage de trois points dans l'espace, dont les positions idéales sont connues. Le palpage s'effectue avec un capteur optique ou à contact qui accoste des trous ou des billes de mesure qui ont été positionnés avec une grande précision sur la plaque support.

Syntaxe

```
MEAFRAME(<points idéaux>,<points de mesure>,<qualité>)
```

Signification

`MEAFRAME` :	Appel de fonction		
`<points idéaux>` :	Tableau REAL à 2 dim. qui contient les 3 coordonnées des points idéaux		
`<points de mesure>` :	Tableau REAL à 2 dim. qui contient les 3 coordonnées des points mesurés		
`<qualité>` :	Variable permettant de retourner les informations de qualité du calcul de FRAME		
	Type :	VAR REAL	
	Valeur :	-1	Les points idéaux se trouvent approximativement sur une droite : le frame n'a pas pu être calculé. La variable FRAME en retour comprend un frame neutre.
		-2	Les points mesurés se trouvent approximativement sur une droite : le frame n'a pas pu être calculé. La variable FRAME en retour comprend un frame neutre.
		-4	Le calcul de la matrice de rotation est impossible pour une autre raison.
		≥ 0.0	Somme des distorsions (écarts entre les points) nécessaire pour transférer passer le triangle mesuré dans un triangle congruent au triangle idéal.

Remarque

Qualité de la mesure

Pour que les coordonnées mesurées puissent être mises en relation avec les coordonnées idéales par une rotation/translation combinées, il doit y avoir congruence entre le triangle formé par les points mesurés et le triangle idéal. Ceci est obtenu par un algorithme de compensation qui minimise la somme des carrés des écarts qui permettent de passer du triangle mesuré au triangle idéal.

La distorsion effectivement nécessaire des points mesurés étant le reflet de la qualité de la mesure, elle est fournie par MEAFRAME en tant que variable supplémentaire.

Remarque

Le frame créé par MEAFRAME peut être transformé en un autre frame de la chaîne de frames au moyen de la fonction ADDFRAME (voir l'exemple "Concaténation avec ADDFRAME").

Exemples

Exemple 1 :

Programme pièce 1 :

```
Code de programme
...
DEF FRAME CORR_FRAME
```

Définition de points de mesure :

```
Code de programme                                   Commentaire
DEF REAL IDEAL_POINT[3,3]=
SET(10.0,0.0,0.0,0.0,10.0,0.0,0.0,0.0,10.0)
DEF REAL MEAS_POINT[3,3]=                           ; Pour le test.
SET(10.1,0.2,-0.2,-0.2,10.2,0.1,-0.2,0.2,9.8)


DEF REAL FIT_QUALITY=0
DEF REAL ROT_FRAME_LIMIT=5                          ; Permet une rotation de la posi-
                                                    tion de la pièce de 5 degrés au
                                                    maximum.
DEF REAL FIT_QUALITY_LIMIT=3                        ; Permet au maximum un décalage de
                                                    3 mm
                                                    entre le triangle idéal et le tri-
                                                    angle mesuré.
DEF REAL SHOW_MCS_POS1[3]
DEF REAL SHOW_MCS_POS2[3]
DEF REAL SHOW_MCS_POS3[3]
```

Code de programme	**Commentaire**
N100 G01 G90 F5000	
N110 X0 Y0 Z0	
N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS_POINT,FIT_QUALITY)	
N230 IF FIT_QUALITY < 0	
SETAL(65000)	
GOTOF NO_FRAME	
ENDIF	
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT	
SETAL(65010)	
GOTOF NO_FRAME	
ENDIF	
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT	; Limitation du 1er angle RPY.
SETAL(65020)	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT	; Limitation du 2e angle RPY.
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; Limitation du 3e angle RPY.
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 $P_IFRAME=CORR_FRAME	; Activation du frame de balayage par un frame réglable. ; Vérification du frame en positionnant les axes géométriques sur les points idéaux.
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=$AA_IM[X]	
N420 SHOW_MCS_POS1[1]=$AA_IM[Y]	
N430 SHOW_MCS_POS1[2]=$AA_IM[Z]	
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2]	
N510 SHOW_MCS_POS2[0]=$AA_IM[X]	
N520 SHOW_MCS_POS2[1]=$AA_IM[Y]	
N530 SHOW_MCS_POS2[2]=$AA_IM[Z]	
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2]	
N610 SHOW_MCS_POS3[0]=$AA_IM[X]	
N620 SHOW_MCS_POS3[1]=$AA_IM[Y]	
N630 SHOW_MCS_POS3[2]=$AA_IM[Z]	
N700 G500	; Désactiver le frame réglable, car préréglage avec le frame nul (sans indication de valeur).

Code de programme	Commentaire
No_FRAME	; Désactivation du frame réglable, car préaffecté avec le frame nul (sans indication de valeur).
M0	
M30	

Exemple 2 : concaténation de frames

Concaténation de MEAFRAME à des fins de correction

La fonction MEAFRAME fournit un frame de correction. Si ce frame de correction est concaténé avec le frame réglable $P_UIFR[1] qui était actif lors de l'appel de la fonction (par ex. G54), on obtient un frame réglable pour d'autres calculs de déplacement ou d'usinage.

Concaténation avec ADDFRAME

Si ce frame de correction est actif à un autre endroit de la chaîne de frames ou si d'autres frames sont actifs avant le frame réglable, la fonction ADDFRAME peut être utilisée pour la concaténation à l'un des frames de base spécifiques à un canal ou à un frame système.

Ne doivent pas être actifs dans les frames :

- la fonction miroir avec MIRROR
- la mise à l'échelle avec SCALE

Les paramètres d'entrée pour les valeurs réelles et les valeurs de consigne sont les coordonnées de la pièce. Dans le système de base de la commande, ces coordonnées doivent toujours être indiquées en mm ou en inch (G71/G70) en tant que mesure se rapportant au rayon (DIAMOF).

Bibliographie :
Pour d'autres informations sur ADDFRAME, voir :
Description fonctionnelle Fonctions de base, K2 : Axes, systèmes de coordonnées, frames

6.8 Frames à définition globale pour NCU

Les frames à définition globale pour NCU sont disponibles une seule fois par NCU, pour tous les canaux. Ils peuvent être écrasés et lus par tous les canaux. L'activation des frames à définition globale pour NCU a lieu dans le canal correspondant.

Les frames à définition globale permettent d'appliquer des décalages, des changements d'échelle et des fonctions miroir à des **axes de canal ou à des axes machine**.

Liens géométriques et concaténations de frames

Dans le cas de ces frames, il n'existe pas de lien géométrique entre les axes. C'est pourquoi les rotations et la programmation de descripteurs d'axes géométriques sont impossibles.

- Les frames à définition globale ne peuvent contenir aucune composante de rotation. La programmation d'une rotation est refusée avec émission de l'alarme : "18310 Canal %1 Bloc %2 Frame : rotation non autorisée".
- La concaténation de frames à définition globale et de frames spécifiques à un canal est possible. Le frame résultant contient toutes les composantes des frames, y compris les rotations pour tous les axes. L'affectation d'un frame comportant des composantes de rotation à un frame à définition globale est refusée avec émission de l'alarme "Frame : rotation non autorisée".

Frames à définition globale pour NCU

Frames de base à définition globale pour NCU $P_NCBFR[n]

jusqu'à 8 frames de base NCU globaux peuvent être configurés :

Des frames de base spécifiques à un canal peuvent être actifs simultanément.

Les frames à définition globale peuvent être écrasés et lus par tous les canaux d'une NCU. Lors de l'écrasement d'un frame à définition globale, l'utilisateur doit assurer la coordination entre les canaux. Cela est par exemple réalisable avec des marques wait (WAITMC).

Constructeur de machine

Le nombre de frames de base globaux est configuré par des paramètres machine.

Bibliographie :
Description fonctionnelle Fonctions de base ; axes, systèmes de coordonnées, frames (K2)

Frames réglables à définition globale pour NCU $P_UIFR[n]

Tous les frames réglables G500, G54...G599 peuvent être configurés comme étant soit à définition globale pour NCU, soit spécifiques à un canal.

Constructeur de machine

Tous les frames paramétrables peuvent être reconfigurés en frames globaux à l'aide du PM18601 $MN_MM_NUM_GLOBAL_USER_FRAMES.

En tant que descripteurs d'axes pour les instructions Frame, vous pouvez utiliser des descripteurs d'axes de canaux et des descripteurs d'axes machine. La programmation de descripteurs d'axes géométriques est refusée avec émission d'une alarme.

6.8.1 Frames spécifiques à un canal ($P_CHBFR, $P_UBFR)

Les frames réglables ou les frames de base peuvent être lus via le programme pièce et via OPI par la commande et par l'AP.

Le décalage fin est également possible pour les frames à définition globale. L'inhibition de frames à définition globale a lieu comme pour les frames spécifiques à un canal, avec G53, G153, SUPA et G500.

Constructeur de machine

Le paramètre machine PM28081 $MC_MM_NUM_BASE_FRAMES permet de définir le nombre de frames de base dans un canal. En configuration standard, il existe au moins un frame de base par canal. 8 frames de base sont possibles au maximum par canal. En plus de ces 8 frames de base, 8 frames de base à définition globale pour NCU peuvent également exister dans le canal.

Frames spécifiques à un canal

$P_CHBFR[n]

La variable système $P_CHBFR[n] permet de lire et d'écraser les frames de base. Lors de l'écrasement d'un frame de base, le frame de base concaténé résultant n'est pas activé immédiatement, mais uniquement lors de l'exécution d'une instruction G500, G54 ... G599. Cette variable sert principalement à enregistrer les opérations d'écriture dans le frame de base effectuées par l'IHM ou l'AP. Ces variables frame sont sauvegardées.

Premier frame de base du canal

Un écrasement de la variable prédéfinie $P_UBFR n'active pas simultanément le frame de base d'indice 0 ; l'activation n'a lieu que lors de l'exécution d'une instruction G500, G54 ... G599. Cette variable peut également être lue et écrasée dans le programme.

$P_UBFR

$P_UBFR est identique à $P_CHBFR[0]. En version standard, il y a toujours un frame de base dans le canal, de sorte que la variable système est compatible avec les versions de logiciel précédentes. S'il n'y a pas de frame de base spécifique au canal, l'alarme "Frame : instruction non autorisée" est émise lors de l'écriture ou de la lecture.

6.8.2 Frames actifs dans un canal

Les frames actifs dans un canal sont introduits par le programme pièce via les variables système respectives de ces frames. Les frames système en font également partie. Ces variables système permettent de lire et d'écraser le frame système courant dans le programme pièce.

Frames actifs courants dans un canal

Vue d'ensemble

Frames système courants	pour :
$P_PARTFRAME	TCARR et PAROT
$P_SETFRAME	Préréglage des mémoires de valeurs réelles et effleurement
$P_EXTFRAME	Décalage d'origine externe
$P_NCBFRAME[n]	Frames de base courants à définition globale pour NCU
$P_CHBFRAME[n]	Frames de base courants spécifiques à un canal
$P_BFRAME	Premier frame de base courant du canal
$P_ACTBFRAME	Frame de base résultant
$P_CHBFRMASK et $P_NCBFRMASK	Frame de base résultant
$P_IFRAME	Frame réglable courant
Frames système courants	pour :
$P_TOOLFRAME	TOROT et TOFRAME
$P_WPFRAME	Points de référence de la pièce
$P_TRAFRAME	Transformations
$P_PFRAME	Frame programmable courant
Frame système courant	pour :
$P_CYCFRAME	Cycles
P_ACTFRAME	Frame résultant courant
Concaténation de frames	Le frame courant contient le frame de base résultant.

$P_NCBFRAME[n] Frames de base courants à définition globale pour NCU

La variable système $P_NCBFRAME[n] permet de lire et d'écraser les éléments de tableau des frames de base courants à définition globale. Le frame de base résultant est pris en considération dans le canal lors de l'opération d'écriture.

Le frame modifié n'est actif que dans le canal dans lequel le frame a été programmé. Si le frame doit être modifié pour tous les canaux d'une NCU, vous devez écraser simultanément $P_NCBFR[n] et $P_NCBFRAME[n]. Ensuite, le frame doit encore être activé dans les autres canaux à l'aide de G54, par ex. En cas d'écrasement d'un frame de base, le frame de base résultant est recalculé.

$P_CHBFRAME[n] Frames de base courants spécifiques à un canal

La variable système $P_CHBFRAME[n] permet de lire et d'écraser les éléments de tableau des frames de base courants spécifiques à un canal. Le frame de base résultant est pris en considération dans le canal lors de l'opération d'écriture. En cas d'écrasement d'un frame de base, le frame de base résultant est recalculé.

$P_BFRAME 1er frame de base courant du canal

La variable frame prédéfinie $P_BFRAME permet de lire et d'écraser, dans le programme pièce, le frame de base courant d'indice 0, qui est valide dans le canal. Ce frame est pris immédiatement en considération.

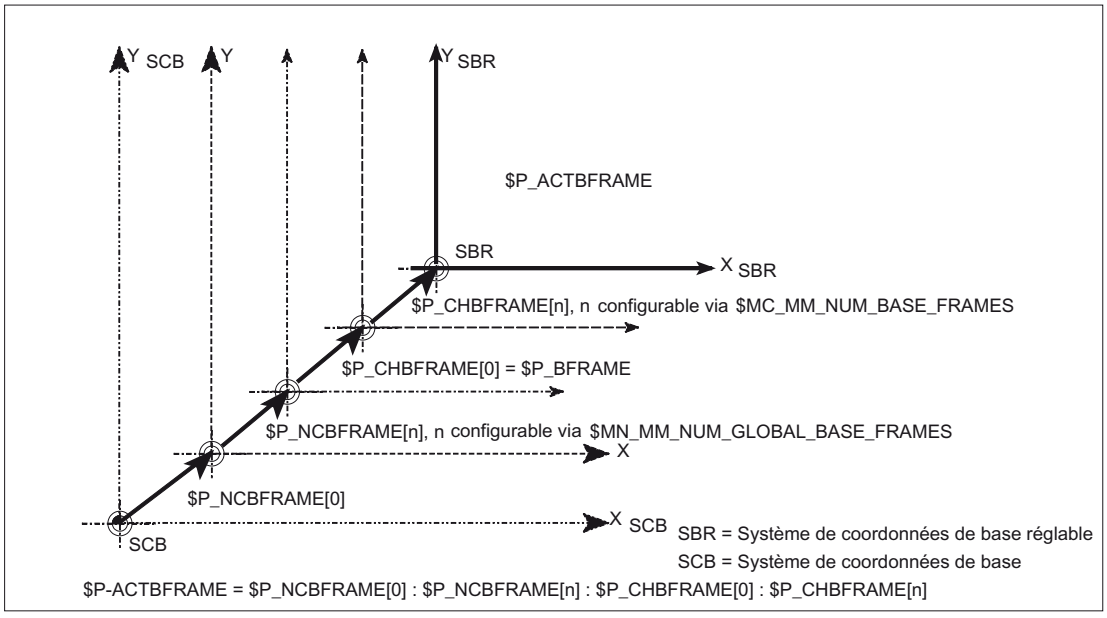
$P_BFRAME est identique à $P_CHBFRAME[0]. Par défaut, la variable système a toujours une valeur valide. S'il n'y a pas de frame de base spécifique au canal, l'alarme "Frame : instruction non autorisée" est émise lors de l'écriture ou de la lecture.

$P_ACTBFRAME Frame de base résultant

La variable $P_ACTFRAME détermine le frame de base résultant concaténé. Elle n'est accessible qu'en lecture.

$P_ACTFRAME correspond à :

$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n].



$P_CHBFRMASK et $P_NCBFRMASK Frame de base résultant

Avec les variables système $P_CHBFRMASK et $P_NCBFRMASK, l'utilisateur peut sélectionner les frames de base pour le calcul du frame de base "résultant". Ces variables ne peuvent être programmées que dans le programme et lues via l'OPI. Les valeurs de ces variables sont interprétées en tant que masques binaires et indiquent les éléments de $P_ACTFRAME à prendre en considération.

$P_CHBFRMASK permet d'indiquer les frames de base spécifiques au canal à prendre en considération et $P_NCBFRMASK, les frames de base à définition globale pour NCU à prendre en considération.

La programmation de ces variables entraîne un nouveau calcul du frame de base résultant et du frame résultant. Les valeurs de $P_CHBFRMASK et $P_NCBFRMASK après Reset et les préréglages sont les suivantes :

$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK

$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK

Exemple :

`$P_NCBFRMASK = 'H81'` ; $P_NCBFRAME[0] : $P_NCBFRAME[7]

`$P_CHBFRMASK = 'H11'` ; $P_CHBFRAME[0] : $P_CHBFRAME[4]

$P_IFRAME Frame réglable courant

La variable frame prédéfinie $P_IFRAME permet de lire et d'écraser, dans le programme pièce, le frame réglable courant qui est valide dans le canal. Le frame réglable écrasé est pris immédiatement en considération.

Dans le cas des frames réglables à définition globale pour NCU, le frame modifié n'est actif que dans le canal dans lequel le frame a été programmé. Si le frame doit être modifié pour tous les canaux d'une NCU, il faut écraser simultanément $P_UIFR[n] et $P_IFRAME. Ensuite, le frame correspondant doit encore être activé dans les autres canaux à l'aide de G54, par ex.

$P_PFRAME Frame programmable courant

$P_PFRAME est le frame programmable obtenu à partir de la programmation de TRANS/ ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR ou de l'affectation de CTRANS, CROT, CMIRROR, CSCALE au FRAME programmable.

Variable de frame programmable courante qui établit la relation entre le système de coordonnés-pièce réglable (SPR) et le système de coordonnées-pièce (SCP).

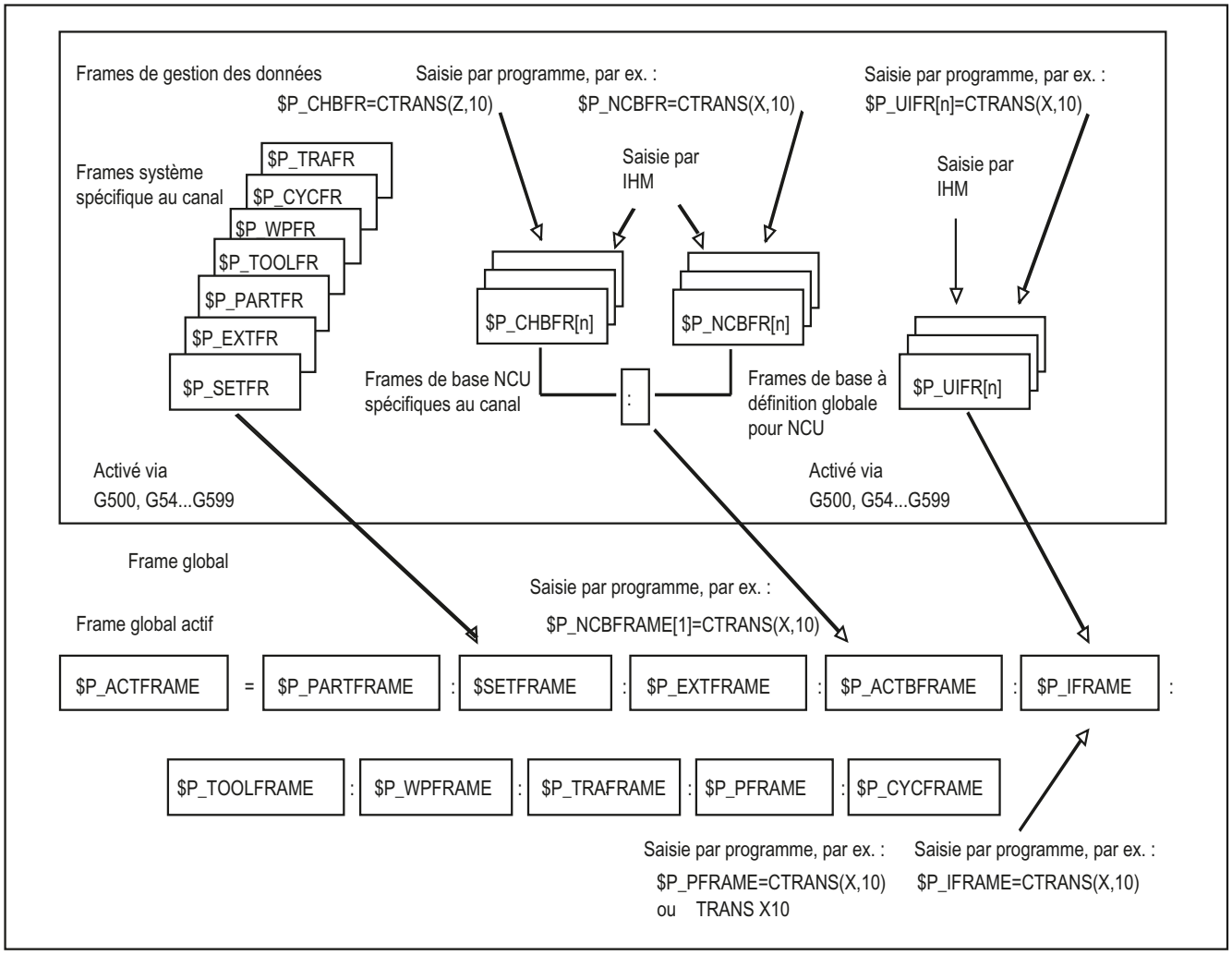
P_ACTFRAME Frame résultant courant

Le frame résultant courant $P_ACTFRAME résulte de la concaténation de tous les frames de base, du frame réglable courant et du frame programmable. Le frame courant est toujours actualisé si une composante de frame subit une modification.

`$P_ACTFRAME` correspond à :

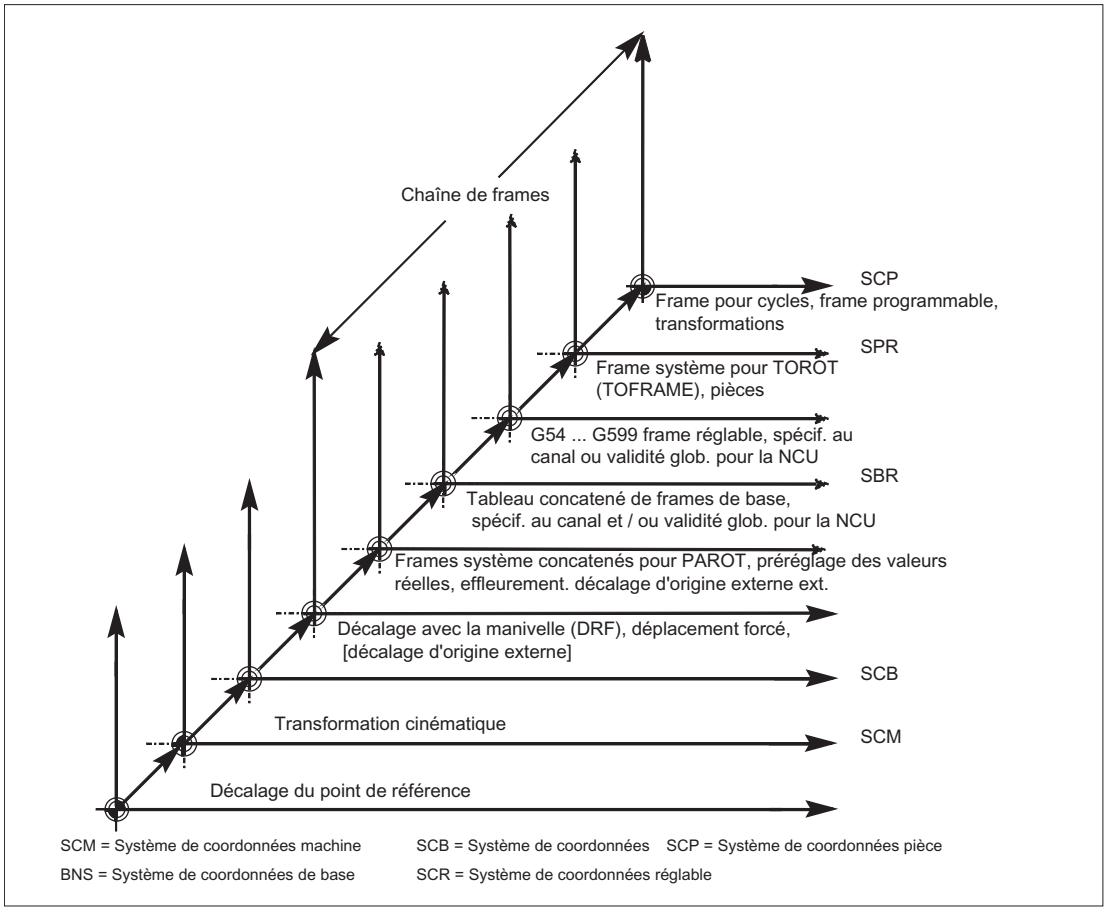
`$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :`

`$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME`


Frames de gestion des données
Saisie par programme, par ex. :
$P_CHBFR=CTRANS(Z,10)
$P_NCBFR=CTRANS(X,10)
Saisie par programme, par ex. :
$P_UIFR[n]=CTRANS(X,10)
Frames système spécifique au canal
$P_TRAFR
$P_CYCFR
$P_WPFR
$P_TOOLFR
$P_PARTFR
$P_EXTFR
$P_SETFR
Saisie par IHM
Saisie par IHM
$P_CHBFR[n]
$P_NCBFR[n]
$P_UIFR[n]
Frames de base NCU spécifiques au canal
:
Frames de base à définition globale pour NCU
Activé via G500, G54...G599
Activé via G500, G54...G599
Frame global
Saisie par programme, par ex. :
$P_NCBFRAME[1]=CTRANS(X,10)
Frame global actif
$P_ACTFRAME
=
$P_PARTFRAME
:
$SETFRAME
:
$P_EXTFRAME
:
$P_ACTBFRAME
:
$P_IFRAME
:
$P_TOOLFRAME
:
$P_WPFRAME
:
$P_TRAFRAME
:
$P_PFRAME
:
$P_CYCFRAME
Saisie par programme, par ex. :
$P_PFRAME=CTRANS(X,10)
ou TRANS X10
Saisie par programme, par ex. :
$P_IFRAME=CTRANS(X,10)

Concaténation de frames

Le frame courant résulte de la concaténation du frame de base résultant, du frame réglable, du frame système et du frame programmable (voir frame résultant courant, ci-dessus).



Transformations 7

7.1 Programmation générale des types de transformation

Fonctionnement général

Pour adapter la commande à différentes cinématiques de machine, il est possible de programmer une sélection de types de transformation avec des paramètres appropriés. Par le biais de ces paramètres, il est possible, pour la transformation choisie, de convenir de l'orientation appropriée de l'outil dans l'espace, mais aussi des mouvements d'orientation des axes rotatifs.

Avec les transformations trois, quatre ou cinq axes, les indications de position programmées se réfèrent toujours à la pointe de l'outil qui est poursuivie perpendiculairement à la surface d'usinage dans l'espace. Les coordonnées cartésiennes sont converties du système de coordonnées de base au système de coordonnées machine et se rapportent aux axes géométriques. Ceux-ci décrivent le point de fonctionnement. Les axes rotatifs virtuels décrivent les orientations de l'outil dans l'espace et se programment avec TRAORI.

Lors de la transformation cinématique, il est possible de programmer des positions dans le système de coordonnées cartésiennes. La commande transforme les déplacements du système de coordonnées cartésiennes programmés avec TRANSMIT, TRACYL et TRAANG en déplacements des axes machine réels.

Programmation

Transformations trois, quatre et cinq axes TRAORI

La transformation d'orientation convenue est activée avec l'instruction TRAORI et les trois paramètres possibles pour numéro transfo, vecteur d'orientation et décalages d'axes rotatifs.

```
TRAORI(numéro transfo, vecteur d'orientation, décalages d'axes
rotatifs)
```

Transformations cinématiques

Parmi les transformations cinématique, on compte les transformations convenues
`TRANSMIT(numéro transfo)`

`TRACYL(diamètre d'usinage, numéro transfo)`

`TRAANG(angle de l'axe en oblique, numéro transfo)`

Désactiver la transformation active

`TRAFOOF` permet de désactiver la transformation active à ce moment.

Transformation d'orientation

Transformations trois, quatre et cinq axes TRAORI

Pour usiner de manière optimale des surfaces à relief irrégulier dans la zone de travail de la machine, les machines-outils ont besoin d'axes supplémentaires en plus des trois axes linéaires X, Y et Z. Les axes supplémentaires décrivent l'orientation dans l'espace et sont nommés ci-après "axes d'orientation". On en dispose sous forme d'axes de rotation sur quatre types de machines dotées d'une cinématique différente.

1. Tête pivotante sur deux axes, par exemple tête d'outil à Cardan avec un axe rotatif parallèle à un axe linéaire, le plateau porte-outil étant fixe.
2. Plateau tournant sur deux axes, par exemple tête pivotante fixe avec plateau porte-outil pivotant sur deux axes.
3. Tête pivotante sur un axe et plateau tournant sur un axe, par exemple tête pivotante tournante avec outil orientable et plateau porte-outil pivotant sur un axe.
4. Tête pivotante sur deux axes et plateau tournant sur un axe, par exemple plateau porte-outil pivotant sur un axe et tête pivotante tournante avec outil tournant autour de son propre axe.

Les **transformations 3 et 4 axes** sont des formes particulières des transformations 5 axes et se programment de manière analogue aux transformations 5 axes.

La "**transformation générique 3, 4, 5, 6 axes**" couvre grâce à l'étendue de ses fonctions pour les axes rotatifs disposés à angle droit ainsi que les transformations pour la tête de fraisage à Cardan et peut être activée avec TRAORI comme toute autre transformation d'orientation pour ces quatre types de machine. En cas de transformation générique 5/6 axes, l'orientation de l'outil dispose d'un troisième degré de liberté pour lequel l'outil peut tourner sur son axe propre dans l'espace pour la direction de l'outil.

Bibliographie : /FB3/ Description fonctionnelle Fonctions spéciales ; Transformation 3 à 5 axes (F2).

Position initiale d'orientation de l'outil indépendante de la cinématique

ORIRESET

Lorsque une transformation d'orientation est active avec TRAORI, alors ORIRESET peut indiquer les trois positions initiales de trois axes d'orientation maxi avec des paramètres A, B, C en option. L'affectation de l'ordre des paramètres programmés aux axes rotatifs s'effectue dans l'ordre des axes d'orientation défini par la transformation. Avec la programmation de ORIRESET(A, B, C), les axes d'orientation sont parcourus de façon linéaire et synchronisée de leur position momentanée à la position initiale indiquée.

Transformations cinématiques

TRANSMIT et TRACYL

Pour les fraisages à effectuer sur des tours, on peut, pour la transformation convenue, programmer soit

1. un usinage frontal avec TRANSMIT dans le dispositif de serrage du tour ou
2. un usinage de rainures quelconques sur des corps cylindriques avec TRACYL

.

TRAANG

Si l'axe de pénétration doit aussi pouvoir pénétrer en oblique (pour une utilisation en rectification par exemple), on peut, avec TRAANG, programmer un angle paramétrable pour la transformation convenue.

Déplacement PTP cartésien

Ce que l'on appelle le "déplacement PTP cartésien", dans lequel on peut programmer jusqu'à 8 positions d'articulations STAT= différentes, fait également partie de la transformation cinématique. Les positions sont programmées dans le système de coordonnées cartésiennes, le déplacement de la machine s'effectuant en coordonnées machine.

Bibliographie :
/FB2/ Description fonctionnelle Fonctions d'extension ; transformation cinématique (M1)

Transformations concaténées

Il est possible d'effectuer deux transformations successives. Pour la deuxième transformation ainsi concaténée, les trajets des axes sont repris à partir de la première transformation.

La première transformation peut être :

- Transformation de l'orientation TRAORI
- Transformation polaire TRANSMIT
- Transformation de cylindre TRACYL
- Transformation d'axe oblique TRAANG

La deuxième transformation doit être de type "axe oblique TRAANG"

7.1.1 Mouvements d'orientation lors des transformations

Déplacements d'outils et déplacements d'orientation

Les déplacements des orientations programmables dépendent en premier lieu du type de machine. pour une transformation trois, quatre ou cinq axes avec TRAORI, les axes de rotation ou les axes linéaires orientables décrivent les déplacements d'orientation de l'outil.

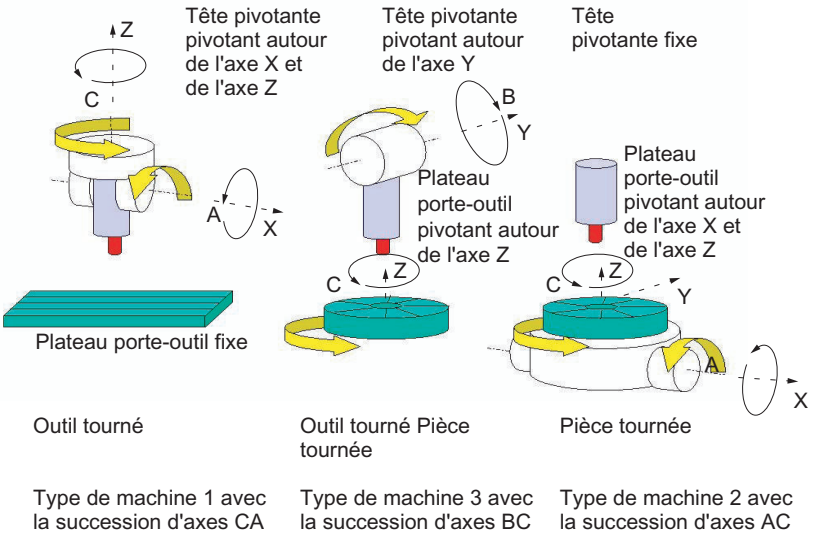
Toute modification de position des axes rotatifs participant à la transformation d'orientation entraîne des déplacements compensateurs des autres axes machine. Ce faisant, la position de la pointe de l'outil reste inchangée.

Les mouvements d'orientation de l'outil peuvent être programmés via les descripteurs d'axe rotatif A…, B…, C… des axes virtuels par l'indication d'angles d'Euler ou d'angles RPY, de vecteurs normaux à la direction ou à la surface, de vecteurs normés pour l'axe de rotation d'un cône ou pour l'orientation intermédiaire sur l'enveloppe d'un cône, selon l'application.

Lors de la transformation cinématique avec TRANSMIT, TRACYL et TRAANG, la commande transforme les déplacements programmés du système de coordonnées cartésiennes en déplacements des axes machine réels.

Cinématique de machine avec transformation trois, quatre et cinq axes TRAORI

Soit l'outil soit le plateau tournant est orientable avec deux axes rotatifs au plus. La combinaison d'une tête pivotante sur un axe et d'un plateau tournant lui aussi sur un axe est également possible.



Type de machine	Programmation de l'orientation
Transformation trois axes Types de machines 1 et 2	Programmation de l'orientation de l'outil uniquement dans le plan perpendiculaire à l'axe rotatif. Il existe **deux** axes de translation (axes linéaires) et **un** axe de rotation (axe rotatif).
Transformation quatre axes Types de machines 1 et 2	Programmation de l'orientation de l'outil uniquement dans le plan perpendiculaire à l'axe rotatif. Il existe **trois** axes de translation (axes linéaires) et **un** axe de rotation (axe rotatif).
Transformation cinq axes Types de machine 3 Tête pivotante sur un axe et plateau tournant sur un axe	Programmation de la transformation d'orientation. Cinématique avec **trois** axes linéaires et **deux** axes rotatifs orthogonaux. Les axes rotatifs sont parallèles à deux des trois axes linéaires. Le premier axe rotatif est déplacé par deux axes linéaires cartésiens. Il fait tourner le troisième axe linéaire avec l'outil. Le second axe rotatif fait pivoter la pièce.

Transformations 5/6 axes génériques

Type de machine	Programmation de la transformation d'orientation
Transformation cinq/six axes générique Types de machine 4 tête orientable à 2 axes avec outil indexable sur lui-même et plateau tournant à un axe	Programmation de la transformation d'orientation. Cinématique avec **trois** axes linéaires et **trois** axes rotatifs orthogonaux. Les axes rotatifs sont parallèles à deux des trois axes linéaires. Le premier axe rotatif est déplacé par deux axes linéaires cartésiens. Il fait tourner le troisième axe linéaire avec l'outil. Le second axe rotatif fait pivoter la pièce. L'orientation de base de l'outil peut être programmée par une rotation supplémentaire sur lui-même avec l'angle de rotation THETA.

A l'appel de la "transformation générique trois, quatre et cinq/six axes", l'orientation de base de l'outil peut de plus être transmise. Les restrictions concernant les directions des axes rotatifs ne s'appliquent plus. Lorsque les axes rotatifs ne sont pas exactement perpendiculaires entre eux ou que les axes rotatifs existants ne sont pas parfaitement parallèles aux axes linéaires, la "transformation générique cinq/six axes" peut fournir de meilleurs résultats pour l'orientation de l'outil.

Transformations cinématiques TRANSMIT, TRACYL et TRAANG

Pour les fraisages à effectuer sur des tours ou sur un axe à pénétration oblique en rectification, ce sont les dispositions d'axes suivantes qui sont valables en standard, en fonction de la transformation:

TRANSMIT	Activation de la transformation polaire
usinage frontal sans démontage de la pièce	un axe rotatif un axe de pénétration dans le plan perpendiculaire à l'axe rotatif un axe longitudinal parallèle à l'axe rotatif

TRACYL	Activation de la transformation de surface latérale de cylindre
Usinage de rainures de forme quelconque sur des corps cylindriques	un axe rotatif un axe de pénétration dans le plan perpendiculaire à l'axe rotatif un axe longitudinal parallèle à l'axe rotatif

TRAANG	Activation de la transformation axe oblique
Usinage avec axe de pénétration incliné	un axe rotatif un axe de pénétration avec angle paramétrable un axe longitudinal parallèle à l'axe rotatif

Déplacement PTP cartésien

Le déplacement de la machine s'effectue en coordonnées machine et se programme avec:

TRAORI	Activation de la transformation
Déplacement point à point PTP	Déplacement à une position dans le système de coordonnées cartésiennes (SCM)
CP	Déplacement avec interpolation des axes cartésiens dans (SCB)
STAT	La position des articulations dépend de la transformation
TU	Selon quel angle faut-il déplacer les axes au plus court?

Déplacement PTP avec une transformation générique cinq/six axes

Le déplacement de la machine s'effectue dans les coordonnées machine et l'orientation de l'outil peut aussi bien être programmé avec des positions d'axes rotatifs qu'avec des vecteurs Euler ou des angles RPY indépendants de la cinématique ou encore des vecteurs de direction.

Une interpolation d'axe rotatif, une interpolation vectorielle avec une interpolation circulaire de grand rayon ou une interpolation du vecteur d'orientation sur une enveloppe de cône sont possibles.

Exemple Transformation trois à cinq axes pour une tête de fraisage à Cardan

La machine-outil comporte au moins 5 axes, dont

- Trois par translation pour des mouvements rectilignes déplaçant le point de fonctionnement à n'importe quelle position dans l'espace de travail.
- Deux axes de pivotement rotatifs disposés selon un angle configurable (généralement 45 degrés) permettant à l'outil d'adopter des orientations dans l'espace qui se limitent à une demi-sphère pour une disposition à 45 degrés.

7.1.2 Aperçu de la transformation d'orientation TRAORI

Types de programmation possibles en relation avec TRAORI

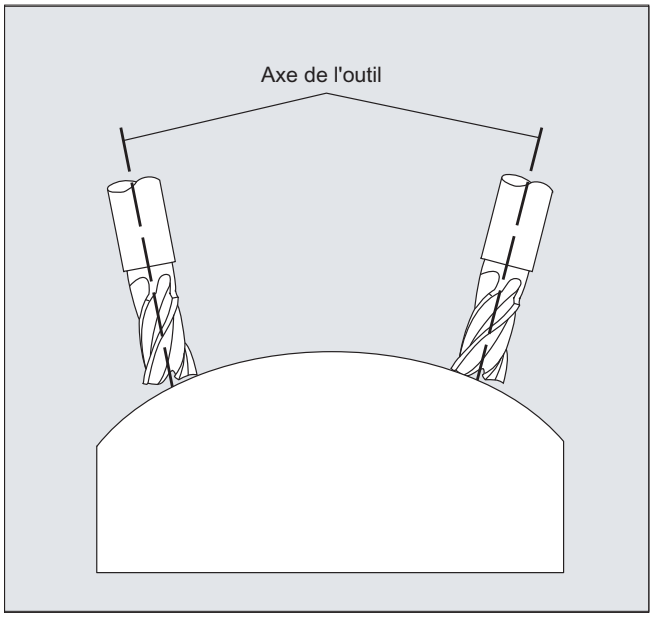
Type de machine	Programmation en cas de transformation TRAORI active
Types de machine 1, 2 ou 3 tête orientable à deux axes ou plateau tournant à deux axes ou une combinaison d'une tête orientable et d'un plateau tournant sur axe chacun.	La succession d'axes des axes d'orientation et la direction d'orientation de l'outil est configurable soit **par rapport à la machine** par le biais des paramètres machine en fonction de la cinématique de machine soit **par rapport à la pièce** avec orientation programmable indépendamment de la cinématique de machine. Les sens de rotation des axes d'orientation dans le système de référence se programment avec: - Système de référence ORIMKS = Système de coordonnées machine - Système de référence ORIWKS = Système de coordonnées pièce La configuration de base est ORIWKS Programmation des axes d'orientation avec: A, B, C positions d'axe machine directes A2, B2, C2 programmation d'angles d'axes virtuels avec - ORIEULER par angle d'Euler (par défaut) - ORIRPY par angle RPY - ORIVIRT1 par axes virtuels d'orientation 1e définition - ORIVIRT2 par axes virtuels d'orientation 2e définition Avec distinction du type d'interpolation : **Interpolation linéaire** - ORIAXES d'axes d'orientation ou d'axes machine **Interpolation circulaire de grand rayon** (interpolation du vecteur d'orientation) - ORIVECT d'axes d'orientation Programmation des axes d'orientation par indication A3, B3, C3 des composantes de vecteur (normale à la direction/surface) Programmation de l'orientation résultante de l'outil A4, B4, C4 du vecteur de la normale à la surface au début du bloc A5, B5, C5 du vecteur de la normale à la surface à la fin du bloc LEAD angle d'avance pour l'orientation de l'outil TILT angle latéral pour l'orientation de l'outil

Type de machine	Programmation en cas de transformation TRAORI active
	Interpolation du vecteur d'orientation sur une enveloppe de cône Changements d'orientation sur une enveloppe de cône placée **librement dans l'espace** Enveloppe de cône par interpolation : - ORIPLANE dans le plan (interpolation circulaire de grand rayon) - ORICONCW sur une enveloppe de cône dans le sens horaire - ORICONCCW sur une enveloppe de cône dans le sens antihoraire A6, B6, C6 vecteur de direction (axe de rotation du cône) -OICONIO interpolation sur une enveloppe de cône avec : A7, B7, C7 Vecteurs intermédiaires (orientation de départ et orientation finale) ou - ORICONTO sur une enveloppe de cône transition tangentielle Changements d'orientation par rapport à **une trajectoire** avec - ORICURVE Indication du déplacement de deux points de contact par le biais de PO[XH]=(xe, x2, x3, x4, x5) Polynômes d'orientation jusqu'au 5e degré PO[YH]=(ye, y2, y3, y4, y5) Polynômes d'orientation jusqu'au 5e degré PO[ZH]=(ze, z2, z3, z4, z5) Polynômes d'orientation jusqu'au 5e degré - ORIPATHS lissage du tracé d'orientation avec A8, B8, C8 La phase de réorientation de l'outil correspond: à la direction et à la longueur de parcours de l'outil lors du mouvement de retrait
Types de machine 1 et 3 D'autres types de machine avec rotation supplémentaire de l'outil sur lui-même exigent un troisième axe rotatif Transformation d'orientation, par exemple une transformation générique six axes. Rotations du vecteur d'orientation.	Programmation **des rotations** de l'orientation de l'outil avec LEAD Angle d'avance Angles par rapport au vecteur normal à la surface PO[PHI]=(.. , ..)] Programmation d'un polynôme jusqu'au 5e degré TILT Angle latéral Rotation par rapport à la tangente à la trajectoire (direction Z) PO[PSI] Programmation d'un polynôme jusqu'au 5e degré THETA Angle de rotation (rotation par rapport à la direction en Z de l'outil) THETA= Valeur atteinte en fin de bloc THETA=AC(...) Passage bloc par bloc à l'introduction des cotes en valeurs absolues THETA=IC(...) Passage bloc par bloc à l'introduction des cotes relatives THETA=Θ_e Interpoler l'angle programmé G90/G91 PO[THT]=(..) Programmation d'un polynôme jusqu'au 5e degré Programmation du vecteur de rotation - ORIROTA Rotation absolue - ORIROTR Vecteur de rotation relatif - ORIROTT Vecteur de rotation tangentiel
Orientation relative à la trajectoire pour des modifications d'orientation par rapport à la trajectoire ou rotation du vecteur de rotation tangentiellement à la trajectoire	Changements d'orientation **par rapport à la trajectoire** avec - ORIPATH Orientation de l'outil par rapport à la trajectoire - ORIPATHS additionnellement en cas de coude dans le tracé d'orientation Programmation du vecteur de rotation - ORIROTC Vecteur de rotation tangentiel, Rotation par rapport à la tangente à la trajectoire

7.2 Transformation trois, quatre et cinq axes (TRAORI)

7.2.1 Corrélations générales de tête de fraisage de type cardan

Pour réunir des conditions de coupe optimales lors de l'usinage de surfaces courbes, il est nécessaire de pouvoir modifier l'angle d'inclinaison de l'outil.

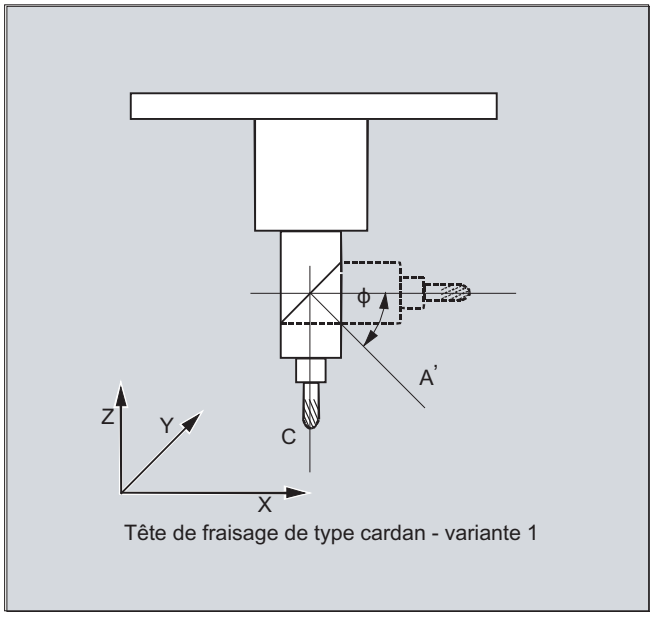


La configuration correspondante de la machine est rangée dans les paramètres machine relatifs aux axes.

Transformation 5 axes

Tête de fraisage de type cardan

Trois axes linéaires (X, Y, Z) et deux axes d'orientation (C, A) déterminent l'angle d'inclinaison et le point d'attaque de l'outil. L'un des deux axes d'orientation - dans l'exemple A' - est incliné, souvent à 45°.

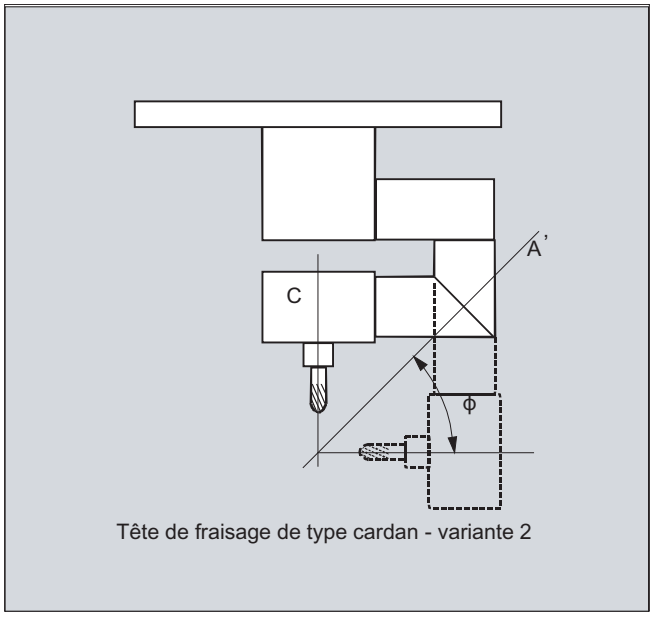


Tête de fraisage de type cardan - variante 1

Dans les exemples montrés ici, vous voyez les dispositions illustrées avec la tête d'outil à Cardan de la cinématique de machine CA!

Constructeur de machine

La succession d'axes des axes d'orientation et la direction d'orientation de l'outil peuvent se régler en fonction de la cinématique de machine par le biais de paramètres machine.



Tête de fraisage de type cardan - variante 2

Sur cet exemple, A' se trouve sous l'angle φ par rapport à l'axe X

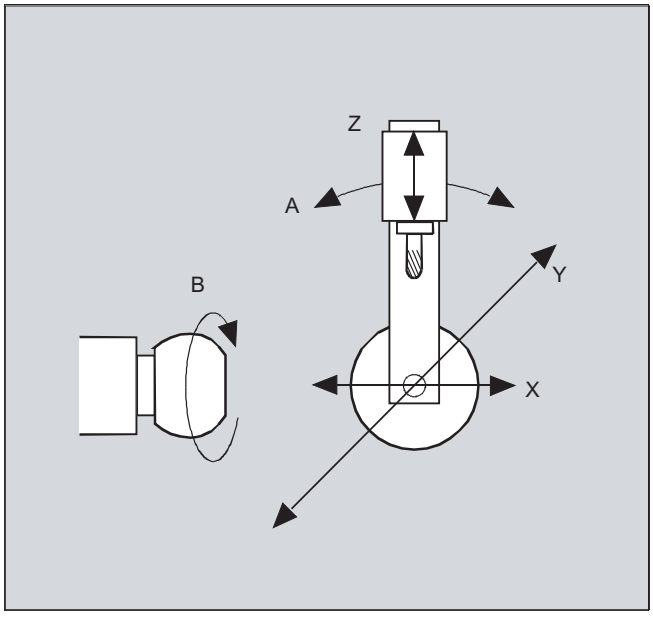
D'une manière générale, les corrélations possibles sont les suivantes:

A' se trouve sous l'angle φ par rapport à	l'axe X
B' se trouve sous l'angle φ par rapport à	l'axe Y
C' se trouve sous l'angle φ par rapport à	l'axe Z

L'angle φ peut être configuré dans la plage 0° à +89° par le biais des paramètres machine.

Avec axe linéaire pivotant

Il s'agit d'une configuration avec une pièce mobile et un outil mobile. La cinématique se compose de trois axes linéaires (X, Y, Z) et de deux axes rotatifs orthogonaux. Le premier axe rotatif se déplace sur un chariot à deux axes linéaires croisés. L'outil est parallèle au troisième axe linéaire. Le second axe rotatif fait pivoter la pièce. Le troisième axe linéaire (axe de pivotement) se situe dans le plan du chariot croisé.



La succession d'axes des axes rotatifs et la direction d'orientation de l'outil sont peuvent se régler en fonction de la cinématique de machine par le biais de paramètres machine.

Les corrélations possibles sont les suivantes :

Axes :	Configuration axiale :
1. Axe rotatif	A A B B C C
2. Axe rotatif	B C A C A B
Axe linéaire pivoté	Z Y Z X Y X

Pour plus d'explications en ce qui concerne les successions d'axes pour la direction d'orientation de l'outil, se reporter à

Bibliographie : /FB3/ Description fonctionnelle Fonctions spéciales ; Transformation 3 à 5 axes (F2), chapitre tête de fraisage de type cardan, "Paramétrage".

7.2.2 Transformation trois, quatre et cinq axes (TRAORI)

L'utilisateur peut configurer deux ou trois axes linéaires et un axe rotatif. On suppose que l'axe rotatif est perpendiculaire au plan d'orientation.

L'orientation de l'outil est possible uniquement dans le plan perpendiculaire à l'axe rotatif. Les transformations sont plus particulièrement destinées aux machines avec outil et pièce mobiles.

La configuration et la programmation des transformations 3 axes et 4 axes sont analogues aux transformations 5 axes.

Bibliographie :
Description fonctionnelle Fonctions spéciales ; Transformations multi-axes (F2)

Syntaxe

```
TRAORI(<n>)
TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)
TRAFOOF
```

Signification

`TRAORI` :	Activation de la première transformation d'orientation convenue	
`TRAORI(<n>)` :	Activation de la n-ième transformation d'orientation convenue	
`<n>` :	Numéro de la transformation	
	Valeur :	1 ou 2
	Exemple : TRAORI(1) active la transformation d'orientation 1	
`<X>,<Y>,<Z>` :	Composante du vecteur d'orientation dans laquelle est dirigé l'outil.	
`<A>,<B>` :	Offset programmable pour les axes rotatifs	
`TRAFOOF` :	Désactiver la transformation	

Orientation de l'outil

En fonction de l'orientation de l'outil sélectionnée, il convient de régler le plan de travail actif (G17, G18, G19) défini dans le programme CN, de façon que la correction de longueur d'outil agisse dans la direction de l'orientation de l'outil.

Remarque

Dès que la transformation est activée, les indications de position (X, Y, Z) se réfèrent toujours à la pointe de l'outil. Toute modification des positions des axes rotatifs impliqués dans la transformation entraîne des déplacements compensatoires des autres axes machine, ce qui permet que la position de la pointe de l'outil reste inchangée.

La transformation d'orientation est toujours dirigée de la pointe de l'outil vers le porte-outil.

Offset pour les axes d'orientation

L'activation de la transformation d'orientation permet de programmer directement un offset supplémentaire pour les axes d'orientation.

Les paramètres peuvent être omis si l'ordre correct est respecté lors de la programmation.

Exemple :

`TRAORI(, , , ,A,B)` ; lorsqu'un seul offset doit être renseigné

Comme alternative à la programmation directe, l'offset supplémentaire pour les axes d'orientation peut aussi être repris automatiquement à partir du décalage d'origine présentement actif. La prise en charge est définie à l'aide des paramètres machine.

Exemples

```
TRAORI(1,0,0,1)          ; L'orientation de base de l'outil pointe dans la direction
                         Z
TRAORI(1,0,1,0)          ; L'orientation de base de l'outil pointe dans la direction
                         Z
TRAORI(1,0,1,1)          ; L'orientation de base de l'outil pointe dans le direction
                         Y/Z (correspond à la position -45°)
```

7.2.3 Variantes de programmation d'orientation et position initiale (ORIRESET)

Programmation d'orientation de l'outil avec TRAORI

En association avec une transformation d'orientation programmable TRAORI, des positions d'axe ou des axes virtuels avec angles ou composants vectoriels peuvent être également programmés par des descripteurs d'axes rotatifs A.., B..., C... en plus des axes linéaires X, Y, Z. Pour les axes d'orientation et les axes machine, différents types d'interpolations sont possibles. Indépendamment des polynômes d'orientation PO[angle] et des polynômes d'axe PO[axe] activés actuellement, on peut programmer plusieurs types de polynômes différents tels que G1, G2, G3, CIP ou POLY.

La modification d'orientation de l'outil peut se programmer également par le biais de vecteurs d'orientation. Ce faisant, l'orientation finale de chaque bloc peut s'effectuer soit par programmation directe du vecteur, soir par programmation des positions des axes rotatifs.

Variantes de programmation d'orientation pour transformations trois à cinq axes

Les variantes de programmation de l'orientation suivantes s'excluent mutuellement.

`A, B, C`	Indication directe de positions d'axe rotatif
`A2, B2, C2`	Programmation des angles des axes virtuels par angle d'Euler ou angle RPY
`A3, B3, C3`	Indication des composantes du vecteur
`LEAD, TILT`	Indication de l'angle d'avance et de l'angle latéral par rapport à la trajectoire et à la surface
`A4, B4, C4` `A5, B5, C5`	Vecteur normal à la surface en début et en fin de bloc

A6, B6, C6 A7, B7, C7	Interpolation du vecteur d'orientation sur une enveloppe de cône
A8, B8, C8	Réorientation de l'outil, direction et longueur de parcours du mouvement de retrait

Accostage de la position de base de l'orientation de l'outil (ORIRESET)

ORIRESET(...) permet de déplacer les axes d'orientation de la cinématique de machine active, de manière linéaire et synchrone, à partir de la position actuelle vers la position initiale programmée. Si, pour un axe, aucune position initiale n'est programmée, alors la position du paramètre machine correspondant $MC_TRAFO5_ROT_AX_OFFSET_1/2 est utilisée.

Des frames des axes rotatifs actifs ne sont pas pris en compte.

Exemples de cinématique de machine CA (nom d'axe de canal C, A)

Instructions	Description
ORIRESET(90, 45)	Axe C : 90° Axe A : 45°
ORIRESET(, 30)	Axe C : $MC_TRAFO5_ROT_AX_OFFSET_1/2[0], Axe A : 30°
ORIRESET( )	Axe C : $MC_TRAFO5_ROT_AX_OFFSET_1/2[0], Axe A : $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

Exemples de cinématique de machine CAC (nom d'axe de canal C, A, B)

Instructions	Description
ORIRESET(90, 45, 90)	Axe C : 90° Axe A : 45° Axe B : 90°
ORIRESET( )	Axe C : $MC_TRAFO5_ROT_AX_OFFSET_1/2[0], Axe A : $MC_TRAFO5_ROT_AX_OFFSET_1/2[1] Achse B : $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]

Remarque

La position initiale de l'orientation de l'outil ne doit être accostée avec ORIRESET(...) que si la transformation d'orientationTRAORI(...) est activée.

Programmation des rotations LEAD, TILT et THETA

Angle d'avance LEAD et angle latéral TILT

Dans le cas de la transformation trois à cinq axes, les rotations de l'orientation de l'outil se programment avec l'angle d'avance LEAD et l'angle latéral TILT.

Angle de rotation THETA

Lors d'une transformation **avec un troisième axe rotatif**, la rotation de l'outil sur lui-même peut être programmée avec l'angle de rotation THETA, aussi bien pour l'orientation avec des composantes de vecteur qu'avec la programmation des angles LEAD, TILT.

7.2.4 Programmation de l'orientation de l'outil (A..., B..., C..., LEAD, TILT)

On dispose des possibilités suivantes pour programmer l'orientation de l'outil:

1. Programmation directe du déplacement des axes rotatifs. Le changement d'orientation s'effectue toujours dans le système de coordonnées de base ou dans le système de coordonnées machine. Les axes d'orientation sont déplacés sous forme d'axes synchrones.
2. Programmation dans les angles Euler ou RPY selon la définition d'angle via `A2, B2, C2`.
3. Programmation du vecteur de direction par le biais de `A3, B3, C3`. Le vecteur de direction est dirigé de la pointe de l'outil vers le porte-outil.
4. Programmation du vecteur normal à la surface en début de bloc avec `A4, B4, C4` et en fin de bloc avec `A5, B5, C5` (fraisage avant).
5. Programmation par le biais de l'angle d'avance `LEAD` et de l'angle latéral `TILT`
6. Programmation de l'axe rotatif du cône comme vecteur normé par le biais de `A6, B6, C6` ou de l'orientation intermédiaire sur l'enveloppe du cône par le biais de `A7, B7, C7`, voir chapitre "Programmation de l'orientation le long d'une enveloppe de cône (ORIPLANE, ORICONxx)".
7. Programmation de réorientation, de direction et longueur de parcours de l'outil pendant le mouvement de relèvement via `A8, B8, C8`,
   voir chapitre "Lissage du tracé d'orientation (ORIPATHS A8=, B8=, C8=)"

Remarque

Dans tous les cas, la programmation de l'orientation est autorisée uniquement si une transformation d'orientation est activée.

Avantage : ces programmes sont transposables sur n'importe quelle cinématique de machine.

Définition de l'orientation de l'outil via une instruction G

Remarque

Constructeur de machine

Le paramètre machine permet de commuter entre angle Euler ou RPY. Avec des paramètres machine adaptés, une commutation aussi bien dépendante qu'indépendante de l'instruction G active du groupe 50 est possible. Les paramétrages suivants sont possibles :

1. Lorsque les deux paramètres machine pour la définition des axes d'orientation et la définition de l'angle d'orientation sont remis à zéro par l'instruction G :
   Les angles programmés avec `A2, B2, C2` sont interprétés **en fonction du paramètre machine** Définition de l'angle de la programmation d'orientation soit comme des angles Euler soit comme des angles RPY.
2. Lorsque le paramètre machine pour la définition des axes d'orientation est mis à un par l'instruction G, la commutation a lieu
   en fonction de l'instruction G active du groupe 50 :
   Les angles programmés avec `A2, B2, C2` sont interprétés selon l'une des instructions G actives `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS` et `ORIPY2`. Les valeurs programmées avec les axes d'orientation sont également interprétées comme angles d'orientation selon l'instruction G active du groupe 50.
3. Lorsque le paramètre machine pour la définition des angles d'orientation au moyen de l'instruction G est mis à un et que le paramètre machine pour la définition des axes d'orientation au moyen de l'instruction G est mis à zéro, la commutation s'effectue **indépendamment** de l'instruction G active dans le groupe 50 :
   les angles programmés avec `A2, B2, C2` sont interprétés en fonction de l'une des instructions G `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2 ORIAXPOS` et `ORIPY2` actives. Les valeurs programmées avec les axes d'orientation sont toujours interprétées comme positions d'axe rotatif, indépendamment de l'instruction G active du groupe 50.

Syntaxe

Positions des axes rotatifs

```
G1 X<valeur> Y<valeur> Z<valeur> A<valeur> B<valeur> C<valeur>
```

Angles d'Euler

```
G1 X<valeur> Y<valeur> Z<valeur> A2<valeur> B2<valeur> C2<valeur>
```

Vecteur de direction

```
G1 X<valeur> Y<valeur> Z<valeur> A3<valeur> B3<valeur> C3<valeur>
```

Vecteur normal à la surface en début de bloc

```
G1 X<valeur> Y<valeur> Z<valeur> A4<valeur> B4<valeur> C4<valeur>
```

Vecteur normal à la surface en fin de bloc

```
G1 X<valeur> Y<valeur> Z<valeur> A5<valeur> B5<valeur> C5<valeur>
```

Angle d'avance

`LEAD=<valeur>`

Angle latéral

`TILT=<valeur>`

Signification

`G1` :	interpolation linéaire
`X, Y, Z` :	Positions des axes linéaires
`A, B, C` :	Positions des axes rotatifs
`A2=, B2=, C2=`:	Programmation d'angles (angles d'Euler ou RPY)
`A3=, B3=, C3=`:	Vecteur de direction en coordonnées X, Y et Z du SCP
`A4=, B4=, C4=`:	Vecteur normal à la surface en début de bloc, en coordonnées X, Y et Z du SCP
`A5=, B5=, C5=`:	Vecteur normal à la surface en fin de bloc, en coordonnées X, Y et Z du SCP
`LEAD=` :	Angle d'avance [1)]
`TILT=` :	Angle latéral [1)]
1) L'interprétation des indications d'angle dépend des réglages dans PM21094 $MC_ORIPATH_MODE	

Informations complémentaires

En règle générale, les programmes 5 axes sont générés par des systèmes de CAO/FAO et ne sont pas entrés au pied de la machine. C'est pourquoi les explications suivantes s'adressent principalement aux programmeurs de postprocesseurs.

Les instructions suivantes sont disponibles pour la programmation de l'orientation :

Instruction	Signification
`ORIEULER` :	Angle d'Euler avec ordre de rotation ZX'Z"
`ORIRPY` :	Angle RPY avec ordre de rotation XY'Z"
`ORIRPY2` :	Angle RPY avec ordre de rotation ZY'X"
`ORIVIRT1` :	Axes virtuels d'orientation avec positions ordre de rotation librement définissables par : PM21120 $MC_ORIAX_TURN_TAB_1
`ORIVIRT2` :	Axes virtuels d'orientation avec positions ordre de rotation librement définissables par : PM21130 $MC_ORIAX_TURN_TAB_2
`ORIAXPOS` :	Axes virtuels d'orientation avec positions d'axe rotatif

Remarque

Le constructeur de machine peut définir différentes variantes via des paramètres machine. Veuillez tenir compte des indications du constructeur de machine.

Programmation en angles d'Euler ORIEULER, ordre de rotation Z X' Z"

Les valeurs programmées avec A2, B2, C2 lors de la programmation de l'orientation ORIEULER sont interprétées comme angles d'Euler (en degrés).

Le nouveau vecteur d'orientation est obtenu à partir des trois rotations suivantes du vecteur d'orientation d'origine

1. avec l'axe rotatif A2 autour de l'axe de coordonnées Z
2. avec l'axe rotatif B2 autour du nouvel axe de coordonnées X'
3. avec l'axe rotatif C2 autour du nouvel axe de coordonnées Z"

Dans ce cas, la valeur de C2 (rotation autour du nouvel axe Z) est sans signification et n'est pas à programmer.

Programmation en angles RPY ORIRPY, ordre de rotation X Y' Z"

Les valeurs programmées avec A2, B2, C2 lors de la programmation de l'orientation ORIRPY sont interprétées comme angles RPY (en degrés) avec l'ordre de rotation X Y' Z".

Remarque

Contrairement à la programmation d'angles avec ORIEULER, les trois valeurs agissent avec ORIRPY sur le vecteur d'orientation.

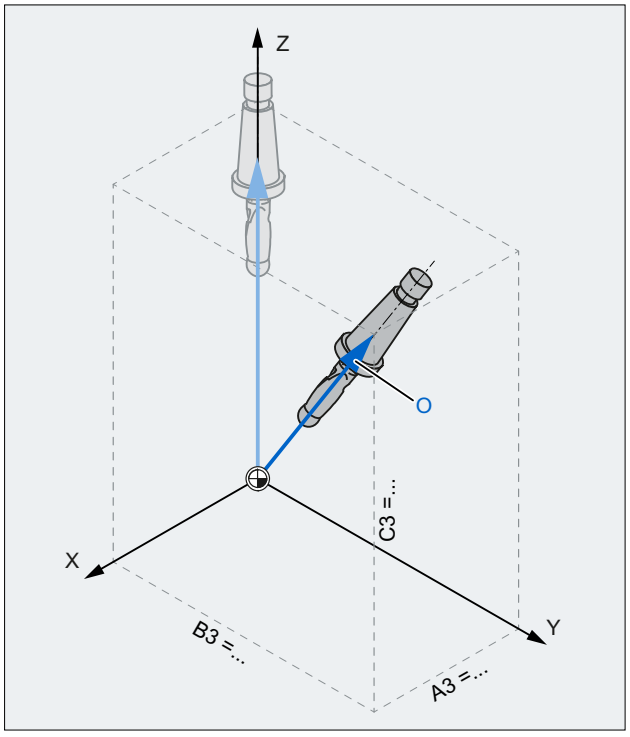
Le nouveau vecteur d'orientation est obtenu à partir des trois rotations suivantes du vecteur d'orientation d'origine

1. avec l'axe rotatif A2 autour de l'axe de coordonnées X
2. avec l'axe rotatif B2 autour du nouvel axe de coordonnées Y'
3. avec l'axe rotatif C2 autour du nouvel axe de coordonnées Z"

Programmation du vecteur de direction

Les composantes du vecteur de direction sont programmées avec A3, B3, C3 . Le vecteur est dirigé vers le porte-outil ; la longueur du vecteur est sans importance.

La commande attribue la valeur zéro aux composantes non programmées du vecteur.



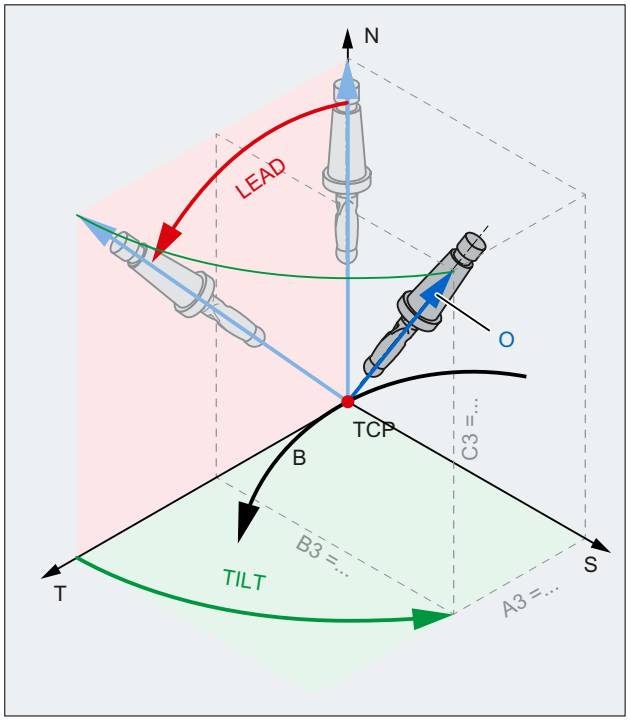
X, Y, Z	Axes de coordonnées du SCP
A3, B3, C3	Composantes du vecteur de direction
O	Vecteur d'orientation

Figure 7-1 Programmation du vecteur de direction

Programmation de l'orientation de l'outil avec LEAD et TILT

L'orientation résultante de l'outil est déterminée à partir de :

- de la tangente à la trajectoire
- Vecteur normal à la surface
  En début de bloc `A4, B4, C4` et en fin de bloc `A5, B5, C5`
- Angle d'avance `LEAD`
  Angle dans le plan défini par la tangente à la trajectoire et le vecteur normal à la surface
- Angle latéral `TILT` en fin de bloc
  Angle par rapport au vecteur normal à la surface, dans le plan perpendiculaire à la tangente à la trajectoire



T	Tangente à la trajectoire
S	Perpendiculaire à la tangente à la trajectoire
N	Normale à la surface
B	Trajectoire
TCP	Tool Center Point :
O	Vecteur d'orientation

Figure 7-2 Programmation de LEAD TILT

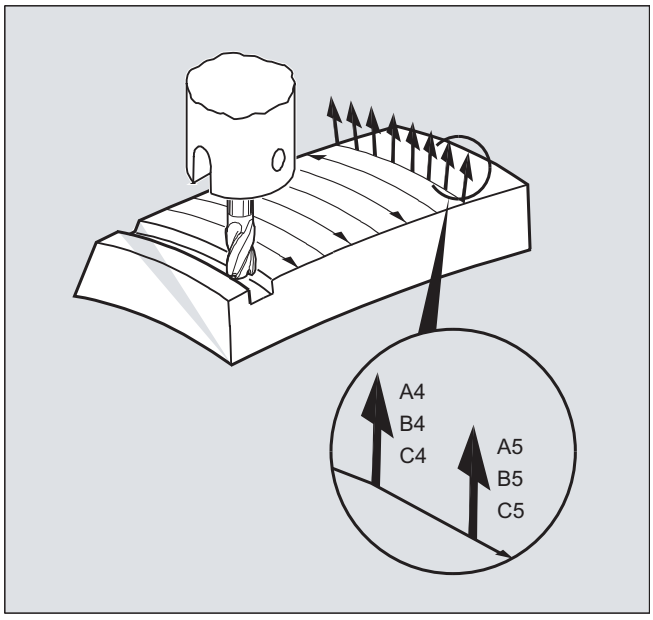
Remarque

Comportement aux angles rentrants pour CO 3D

L'orientation programmée de l'outil est toujours effectuée en fin de bloc, même si le bloc est tronqué au niveau d'un angle rentrant.

7.2.5 Fraisage en bout (A4, B4, C4, A5, B5, C5)

Le fraisage en bout sert à l'usinage des surfaces à courbure quelconque.



Pour ce type de fraisage 3D, vous avez besoin de la description des différentes trajectoires 3D qui balayent la surface de la pièce, ligne après ligne.

Les calculs sont généralement réalisés à l'aide d'un système de FAO, en tenant compte de la forme et des dimensions de l'outil. Les blocs CN générés sont ensuite lus dans la commande par l'intermédiaire de postprocesseurs.

Programmation de la courbure de la trajectoire

Description des surfaces

La description de la courbure de la trajectoire s'effectue par le biais de vecteurs normaux à la surface ayant les composantes suivantes :

`A4, B4, C4` : vecteur de début en début de bloc

`A5, B5, C5` : vecteur de fin en fin de bloc

Si un bloc ne contient que le vecteur de début, le vecteur normal à la surface restera constant pendant tout le bloc. Si un bloc ne contient que le vecteur de fin, une interpolation sur le grand cercle est effectuée de la valeur de fin du bloc précédent jusqu'à la valeur de fin programmée.

Si le vecteur de début et le vecteur de fin sont programmés, une interpolation sur le grand cercle est également effectuée entre les deux directions. Ceci permet d'obtenir des trajectoires lisses.

Dans la configuration de base, les vecteurs normaux à la surface sont orientés dans la direction Z, indépendamment du plan G17 à G19 activé.

La longueur des vecteurs est sans importance.

La valeur zéro est attribuée par défaut aux composantes de vecteur non programmées.

Lorsque ORIWKS est activé (voir "Référence des axes d'orientation (ORIWKS, ORIMKS) (Page 354)"), les vecteurs normaux à la surface se réfèrent au frame actif et subissent également une rotation en cas de rotation du frame.

Constructeur de machine

Le vecteur normal à la surface doit être perpendiculaire à la tangente à la trajectoire avec une zone de tolérance qui est réglable par le biais d'un paramètre machine, sinon une alarme est émise.

7.2.6 Référence des axes d'orientation (ORIWKS, ORIMKS)

Lors de la programmation de l'orientation dans le système de coordonnées pièces par le biais

- des angles d'Euler ou RPY ou
- du vecteur d'orientation,

`ORIMKS`/`ORIWKS` permettent de régler la trajectoire du mouvement de rotation.

Remarque

Constructeur de machine

Le mode d'interpolation pour l'orientation est défini par le paramètre machine :

PM21104 $MC_ORI_IPO_WITH_G_CODE

= FALSE : Les instructions G ORIWKS et ORIMKS servent de référence

= TRUE : Les instructions G du 51e groupe (ORIAXES, ORIVECT, ORIPLANE, ...) servent de référence

Syntaxe

```
ORIMKS=...
ORIWKS=...
```

Signification

`ORIMKS` :	Rotation dans le système de coordonnées machine
`ORIWKS` :	Rotation dans le système de coordonnées pièce

Remarque

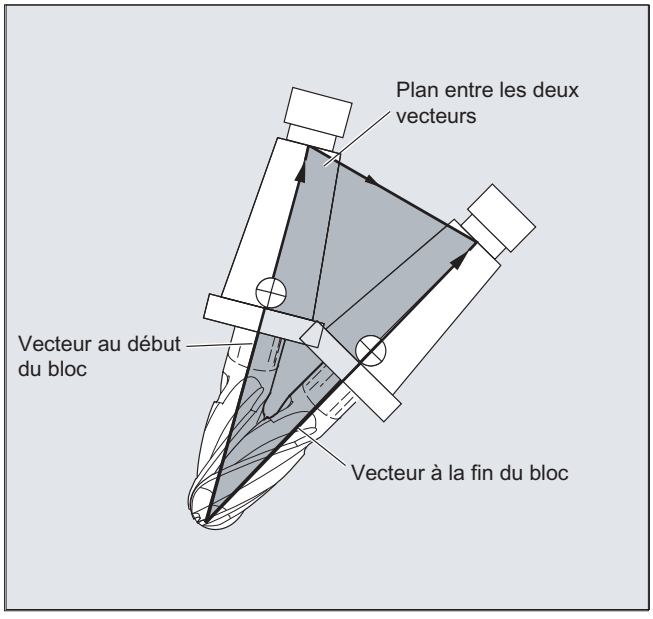
ORIWKS correspond au réglage de base. Si on ne connaît pas d'avance la machine sur laquelle doit fonctionner un programme à 5 axes, il faut par principe choisir ORIWKS. Les déplacements effectivement exécutés par la machine dépendent de sa cinématique.

`ORIMKS` permet de programmer les mouvements réels de la machine, p. ex. pour éviter des collisions avec des équipements, etc.

plus d'informations

Avec ORIMKS, le déplacement effectué par l'outil **dépend** de la cinématique de la machine. Dès qu'il y a un changement d'orientation avec pointe de l'outil fixe, il y a exécution d'une interpolation linéaire entre les positions des axes rotatifs.

Avec ORIWKS, le déplacement de l'outil est **indépendant** de la cinématique de la machine. Dès qu'il y a un changement d'orientation avec pointe de l'outil fixe, l'outil se déplace dans le plan défini par le vecteur de début et le vecteur de fin.



Positions singulières

Remarque

ORIWKS

Les mouvements d'orientation au voisinage de la position singulière d'une machine à 5 axes nécessitent des déplacements importants de la part des axes machine. (C'est le cas par exemple sur une tête pivotante et tournante, avec C comme axe de rotation et A comme axe de pivotement, toutes les positions avec A = 0 étant des positions singulières.)

Constructeur de machine

Pour ne pas surcharger les axes machine, le pilotage de la vitesse réduit fortement la vitesse tangentielle au voisinage des points singuliers.

Les paramètres machine

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

permettent de paramétrer la transformation de façon que les déplacements d'orientation au voisinage du pôle passent effectivement par le pôle, ce qui permet un usinage plus rapide.

Les positions singulières sont traitées uniquement avec le PM $MC_TRAFO5_POLE_LIMIT .

Bibliographie :
/FB3/ Description fonctionnelle Fonctions spéciales ; Transformation 3 à 5 axes (F2), chapitre "Positions singulières et leur traitement".

7.2.7 Programmation des axes d'orientation (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

La fonction "Axes d'orientation" décrit l'orientation de l'outil dans l'espace et s'obtient en programmant les offsets pour les axes rotatifs. On peut obtenir un troisième degré de liberté par une rotation supplémentaire de l'outil autour de son propre axe. Cette orientation de l'outil s'effectue au choix dans l'espace par le biais d'un troisième axe rotatif et requiert la transformation six axes. La rotation de l'outil autour de son propre axe se fixe en fonction du type d'interpolation des vecteurs de rotation avec l'angle de rotation THETA (voir "Rotations de l'orientation de l'outil (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Page 366)".

Les axes d'orientation se programment par le biais des descripteurs d'axes A2, B2, C2.

Syntaxe

```
N... ORIAXES/ORIVECT            ; Interpolation linéaire ou circulaire de
                                  grand rayon
N... G1 X Y Z A B C

N... ORIPLANE                   ; Interpolation d'orientation du plan

N... ORIEULER/ORIRPY/ORIRPY2    ; Angle d'orientation d'Euler / RPY
N... G1 X Y Z A2= B2= C2=       ; Programmation des angles des axes virtuels

N... ORIVIRT1/ORIVIRT2          ; Axes virtuels d'orientation, déf. 1/2
N... G1 X Y Z A3= B3= C3=       ; Programmation du vecteur de direction
```

Remarque

Pour les modifications d'orientation le long d'une enveloppe de cône située dans l'espace, on peut programmer d'autres offsets d'axes rotatifs des axes d'orientation (voir "Programmation de l'orientation le long d'une enveloppe de cône (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Page 358)".

Signification

ORIAXES :	Interpolation linéaire des axes machine ou d'orientation
ORIVECT :	Interpolation circulaire de grand rayon (identique avec ORIPLANE)
ORIMKS : ORIWKS :	Rotation dans le système de coordonnées machine Rotation dans le système de coordonnées pièce Pour une description, voir "Référence des axes d'orientation (ORIWKS, ORIMKS) (Page 354)".

`A= B= C=` :	Programmation de la position des axes machine
`ORIEULER` :	Programmation d'orientation par le biais d'angles d'Euler
`ORIRPY` :	Programmation de l'orientation par le biais d'angles RPY L'ordre de rotation est XYZ, avec : • A2 est l'angle de rotation autour de X • B2 est l'angle de rotation autour de Y • C2 est l'angle de rotation autour de Z
`ORIRPY2` :	Programmation de l'orientation par le biais d'angles RPY L'ordre de rotation est ZYX, avec : • A2 est l'angle de rotation autour de Z • B2 est l'angle de rotation autour de Y • C2 est l'angle de rotation autour de X
`A2= B2= C2=` :	Programmation des angles des axes virtuels
`ORIVIRT1/ORIVIRT2` :	Programmation de l'orientation par le biais d'axes d'orientation virtuels Définition 1 : Selon MD21120 $MC_ORIAX_TURN_TAB_1 Définition 2 : Selon MD21130 $MC_ORIAX_TURN_TAB_2
`A3= B3= C3=` :	Programmation du vecteur de direction de l'axe d'orientation

plus d'informations

Constructeur de machine

Le paramètre machine PM21102 $MC_ORI_DEF_WITH_G_CODE permet de fixer le mode de définition des angles A2, B2, C2 programmés :

La définition s'effectue d'après le paramètre machine PM21100 $MC_ORIENTATION_IS_EULER (par défaut) ou d'après le groupe de codes G 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Le paramètre machine PM21104 $MC_ORI_IPO_WITH_G_CODE permet de définir le type d'interpolation : ORIWKS/ORIMKS ou ORIAXES/ORIVECT.

Mode de fonctionnement Jog

Dans ce mode de fonctionnement, les angles d'orientation sont toujours interpolés de façon linéaire. Lors d'un positionnement relatif et continu à l'aide des touches de déplacement, un seul axe d'orientation peut être déplacé. Avec les manivelles, le déplacement simultané des axes d'orientation est possible.

En déplacement manuel d'axes d'orientation, le commutateur de correction de l'avance spécifique au canal est actif ou, le cas échéant, le commutateur de correction du rapide en déplacement en rapide.

Les paramètres machine suivants permettent de définir séparément les vitesses :

PM21160 $MC_JOG_VELO_RAPID_GEO

PM21165 $MC_JOG_VELO_GEO

PM21150 $MC_JOG_VELO_RAPID_ORI

PM21155 $MC_JOG_VELO_ORI

Remarque

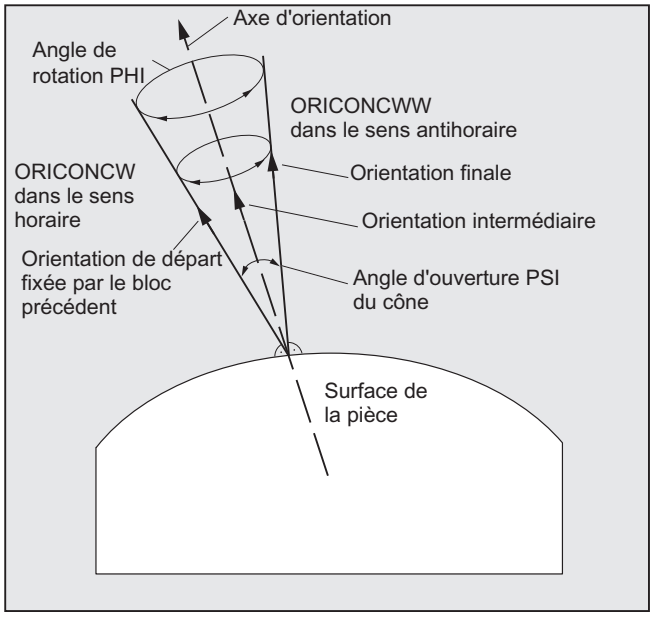
SINUMERIK 840D sl avec "Paquet de transformation pour manipulation"

En mode JOG, la fonction "Déplacement manuel cartésien" permet de régler la translation d'axes géométriques dans les systèmes de référence SCM, SCP et SCO, séparément les uns des autres.

Bibliographie :
Description fonctionnelle Fonctions d'extension ; transformation cinématique (M1)

7.2.8 Programmation de l'orientation le long d'une enveloppe de cône (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

L'orientation étendue permet d'effectuer des changements d'orientation le long d'une enveloppe de cône située dans l'espace. L'interpolation du vecteur d'orientation sur une enveloppe de cône s'effectue avec les instructions modales ORICONxx. Pour l'interpolation dans un plan, on peut programmer l'orientation finale avec ORIPLANE. D'une manière générale, l'orientation de départ est déterminée par les blocs précédents.



Programmation

L'orientation finale se détermine soit par l'indication de la programmation d'angles en angles d'Euler ou RPY avec A2, B2, C2 ou par la programmation des positions des axes rotatifs avec a, B, C. Pour les axes d'orientation le long de l'enveloppe de cône, d'autres indications de programmation sont nécessaires:

- Axe rotatif du cône sous forme de vecteur avec `A6, B6, C6`
- Angle au centre `PSI` avec les descripteurs `NUT`
- Orientation intermédiaire dans l'enveloppe de cône avec `A7, B7, C7`

Remarque

Programmation du vecteur de direction A6, B6, C6 pour l'axe rotatif du cône

La programmation d'une orientation finale n'est pas absolument nécessaire. Si aucune orientation finale n'est indiquée, une enveloppe de cône complète est interpolée avec 360 degrés.

Programmation de l'angle d'ouverture du cône avec NUT=angle

L'indication d'une orientation finale est impérativement nécessaire.

De cette manière, on ne peut pas interpoler une enveloppe de cône complète avec 360 degrés.

Programmation de l'orientation intermédiaire A7, B7, C7 dans l'enveloppe de cône

L'indication d'une orientation finale est impérativement nécessaire. Le changement d'orientation et le sens de rotation sont déterminés de manière univoque par les trois vecteurs Orientation de départ, Orientation finale et Orientation intermédiaire. Ce faisant, les trois vecteurs doivent être différents. Si l'orientation intermédiaire programmée est parallèle à l'orientation de départ ou à l'orientation finale, une interpolation circulaire de grand rayon de l'orientation est effectuée dans le plan défini entre le vecteur de départ et le vecteur final.

Interpolation d'orientation étendue sur une enveloppe de cône

```
N... ORICONCW ou ORICONCCW
N... A6= B6= C6= A3= B3= C3=
ou
N... ORICONTO
N... G1 X Y Z A6= B6= C6=
ou
N... ORICONIO
N... G1 X Y Z A7= B7= C7=
N... PO[PHI]=(a2, a3, A4, A5)
N... PO[PSI]=(b2, b3, b4, b5)
```

Interpolation sur une enveloppe de cône avec

vecteur de direction dans le sens horaire/ antihoraire du cône et orientation finale ou

transition tangentielle et
indication de l'orientation finale

ou
indication de l'orientation finale et d'une orientation intermédiaire dans l'enveloppe de cône avec

polynômes pour l'angle de rotation et
polynômes pour l'angle d'ouverture

Paramètres

`ORIPLANE` :	Interpolation dans le plan (interpolation circulaire de grand rayon)
`ORICONCW` :	Interpolation sur une enveloppe de cône dans le sens horaire
`ORICONCCW` :	Interpolation sur une enveloppe de cône dans le sens anti-horaire
`ORICONTO` :	Interpolation sur une enveloppe de cône avec transition tangentielle
`A6= B6= C6=` :	Programmation de l'axe rotatif du cône (vecteur normé)
`NUT=angle` :	Angle au centre du cône en degrés
`NUT=+179` :	Angle de déplacement inférieur ou égal à 180 degrés
`NUT=-181` :	Angle de déplacement supérieur ou égal à 180 degrés
`ORICONIO` :	Interpolation sur une enveloppe de cône
`A7= B7= C7=` :	Orientation intermédiaire (programmation comme vecteur normé)
`PHI` :	Angle de rotation de l'orientation autour de l'axe d'orientation du cône
`PSI` :	Angle d'ouverture du cône
Polynômes possibles PO[PHI]=(a2, a3, A4, A5) PO[PSI]=(b2, b3, b4, b5)	En dehors des différents angles, on peut aussi programmer des polynômes au maximum du 5e degré

Exemple : Différents changements d'orientation

Code de programme	**Commentaire**
…	
`N10 G1 X0 Y0 F5000`	
`N20 TRAORI(1)`	`; Transformation d'orientation activée.`
`N30 ORIVECT`	`; Interpoler l'orientation de l'outil comme vecteur.`
…	`; Orientation de l'outil dans le plan.`
`N40 ORIPLANE`	`; Sélectionner l'interpolation circulaire de grand rayon.`
`N50 A3=0 B3=0 C3=1`	
`N60 A3=0 B3=1 C3=1`	`; Orientation dans le plan Y/Z tournée de 45 degrés, l'orientation (0,1/√2,1/√2) est atteinte en fin de bloc.`
…	
`N70 ORICONCW`	`; Programmation de l'orientation sur l'enveloppe de cône :`
`N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1`	`; Le vecteur d'orientation est interpolé dans le sens horaire sur une enveloppe de cône avec la direction (0,0,1) jusqu'à l'orientation (1/√2,0,1/√2), l'angle de rotation étant de 270 degrés.`
`N90 A6=0 B6=0 C6=1`	`; L'orientation de l'outil effectue un tour complet sur la même enveloppe de cône.`

Informations complémentaires

Si l'on veut décrire des changements d'orientation sur une enveloppe de cône située à un endroit quelconque dans l'espace, il faut que le vecteur autour duquel la rotation de l'orientation de l'outil doit s'effectuer, soit connu. Il faut en outre que l'orientation de départ et l'orientation finale soient prédéfinies. L'orientation de départ découle du bloc précédent et l'orientation finale doit être soit programmée, soit déterminée par d'autres conditions.

La programmation dans le plan ORIPLANE correspond à ORIVECT

La programmation de l'interpolation circulaire de grand rayon conjugée aux polynômes trigonométriques correspond à l'interpolation linéaire et polynômiale de contours. L'orientation de l'outil est interpolée dans un plan défini par l'orientation de départ et l'orientation finale. Si l'on programme aussi des polynômes, il se peut que le vecteur d'orientation bascule également hors du plan.

Programmation de cercles dans un plan G2/G3, CIP et CT

L'orientation étendue correspond à l'interpolation de cercles dans un plan. En ce qui concerne les possibilités de programmation de cercles avec indication du centre ou du rayon comme G2/G3, cercle via point intermédiaire CIP et cercles tangentiels CT, voir

Bibliographie : Manuel de programmation Bases, "Programmation des instructions de déplacement".

Programmation de l'orientation

Interpolation du vecteur d'orientation sur une enveloppe de cône ORICONxx

Pour l'interpolation d'orientations sur une enveloppe de cône, on peut sélectionner quatre types d'interpolations différents dans le groupe G 51 :

1. Interpolation sur une enveloppe de cône dans le sens horaire `ORICONCW` avec indication de l'orientation finale et de la direction du cône ou de son angle d'ouverture. Le vecteur de direction se programme avec les descripteurs `A6, B6, C6` et l'angle d'ouverture du cône avec le descripteur NUT= plage de valeurs dans l'intervalle 0 à 180 degrés.
2. Interpolation sur une enveloppe de cône dans le sens antihoraire `ORICONCWW` avec indication de l'orientation finale et de la direction du cône ou de son angle d'ouverture. Le vecteur de direction se programme avec les descripteurs `A6, B6, C6` et l'angle d'ouverture du cône avec le descripteur NUT= plage de valeurs dans l'intervalle 0 à 180 degrés.
3. Interpolation sur une enveloppe de cône `ORICONIO` avec indication de l'orientation finale et d'une orientation intermédiaire, programmée avec les descripteurs `A7, B7, C7`.
4. Interpolation sur une enveloppe de cône `ORICONTO` avec transition tangentielle et indication de l'orientation finale. Le vecteur de direction se programme avec les descripteurs `A6, B6, C6`.

7.2.9 Définition de l'orientation de deux points de contact (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Programmation du changement d'orientation par la deuxième courbe spatiale ORICURVE

Une autre possibilité pour programmer les changements d'orientation est de programmer avec `ORICURVE`, outre la pointe de l'outil le long d'une courbe spatiale, également le mouvement d'un deuxième point de contact de l'outil. On peut ainsi, comme lors de la programmation du vecteur de l'outil lui-même, définir de manière univoque les changements d'orientation de l'outil.

Constructeur de machine

Veuillez respecter les indications du constructeur de la machine-outil en ce qui concerne les descripteurs d'axes réglables par le biais de paramètres machine pour la programmation de la deuxième trajectoire d'orientation de l'outil.

Programmation

Dans le cas de ce type d'interpolation, on peut, pour les deux courbes spatiales, programmer des points avec `G1` ou des polynômes avec `POLY`. Les cercles et les développantes ne sont pas admis. De plus, une interpolation de type spline peut être activée avec `BSPLINE` et avec une fonction "groupement de blocs spline courts".

Bibliographie :
Description fonctionnelle Fonctions de base ; contournage, arrêt précis, look-ahead (B1), chapitre : Groupement de blocs spline courts

Les autres types de spline `ASPLINE` et `CSPLINE` tout comme l'activation d'un compacteur avec `COMPON`, `COMPCURV` ou `COMPCAD` ne sont pas admissibles.

Le déplacement des deux points de contact de l'outil peut être prédéfini lors de la programmation des polynômes d'orientation pour des coordonnées du 5e degré maximum.

Interpolation d'orientation étendue avec courbe spatiale supplémentaire et polynômes pour coordonnées

```
N... ORICURVE
N... PO[XH]=(xe, x2, x3, x4, x5)
N... PO[YH]=(ye, y2, y3, y4, y5)
N... PO[ZH]=(ze, z2, z3, z4, z5)
```

Indication du déplacement du deuxième point de contact de l'outil et polynômes supplémentaires des coordonnées correspondantes

Paramètres

`ORICURVE`	`Interpolation de l'orientation avec indication du déplacement de deux points de contact de l'outil.`
`XH YH ZH`	`Descripteur des coordonnées du deuxième point de contact de l'outil du contour supplémentaire en tant que courbe spatiale`

Polynômes possibles PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)	Les courbes spatiales peuvent se programmer avec les points de fin correspondants, mais en plus aussi avec des polynômes.
xe, ye, ze	Points finals de la courbe spatiale
xi, yi, zi	Coefficients des polynômes du 5e degré maximum

Remarque

Descripteurs XH YH ZH pour la programmation d'une 2e trajectoire d'orientation

On doit choisir les descripteurs de manière à ce qu'il ne se produise pas de conflits avec les autres descripteurs des axes linéaires

, les axes X Y Z

et les axes rotatifs tels que

A2 B2 C2, les angles d'Euler ou les angles RPY

A3 B3 C3, les vecteurs de direction

A4 B4 C4 ou A5 B5 C5, les vecteurs normaux à la surface

A6 B6 C6, les vecteurs de rotation ou les coordonnées de points intermédiaires A7 B7 C7

ou d'autres paramètres d'interpolation.

7.3 Polynômes d'orientation (PO[angles], PO[coordonnée])

Indépendamment de l'interpolation polynomiale du groupe G 1 actuellement activée, on peut programmer deux types de polynômes d'orientation jusqu'au 5e degré maximum dans le cas d'une transformation trois à cinq axes.

1. Polynômes pour **angles:** Angle d'avance LEAD, angle latéral TILT par rapport au plan défini par l'orientation de départ et l'orientation finale.
2. Polynômes pour **coordonnées:** XH, YH, ZH de la deuxième courbe spatiale pour l'orientation de l'outil d'un point de référence sur l'outil.

Dans le cas d'une transformation six axes, on peut, en plus de l'orientation de l'outil, programmer la rotation du vecteur de rotation THT avec des polynômes du 5e degré maximum pour les rotations de l'outil lui-même.

Syntaxe

Polynômes d'orientation du type 1 pour **angles**

```
N... PO[PHI]=(a2, a3, A4, A5)
N... PO[PSI]=(b2, b3, b4, b5)
```

Transformation trois à cinq axes

Polynômes d'orientation du type 2 pour **coordonnées**

```
N… PO[XH]=(xe, x2, x3, x4, x5)
N… PO[YH]=(ye, y2, y3, y4, y5)
N… PO[ZH]=(ze, z2, z3, z4, z5)
```

Descripteur pour les coordonnées de la deuxième trajectoire d'orientation pour l'orientation de l'outil

On peut en outre, dans les deux cas, programmer un polynôme pour la **rotation** dans le cas d'une transformation six axes avec

```
N… PO[THT]=(c2, c3, c4, c5)
```

ou

```
N… PO[THT]=(d2, d3, d4, d5)
```

Interpolation par rapport à la trajectoire de la rotation

Interpolation absolue, relative et tangentielle du vecteur d'orientation par rapport au changement d'orientation

. Ceci est possible si la transformation supporte un vecteur de rotation avec un décalage interpolable et programmable par le biais de l'angle de rotation THETA.

Signification

`PO[PHI]`	Angle dans le plan entre l'orientation de départ et l'orientation finale
`PO[PSI]`	Angle décrivant le basculement de l'orientation depuis le plan entre l'orientation de départ et l'orientation finale.
`PO[THT]`	Angle de rotation décrit par la rotation du vecteur de rotation d'une des instructions G du groupe 54 programmés avec THETA

PHI	Angle d'avance LEAD
PSI	Angle latéral TILT
THETA	Rotation par rapport à la direction de l'outil en Z
PO[XH]	Coordonnée X du point de référence sur l'outil
PO[YH]	Coordonnée Y du point de référence sur l'outil
PO[ZH]	Coordonnée Z du point de référence sur l'outil

Informations complémentaires

On ne peut pas programmer les polynômes d'orientation

- si les interpolations spline ASPLINE, BSPLINE, CSPLINE sont activées.
  Les polynômes du type 1 pour les angles d'orientation sont possibles pour chaque type d'interpolation sauf Spline c.-à-d. pour l'interpolation linéaire avec vitesse rapide G00 ou avec avance G01
  pour l'interpolation polynômiale avec POLY et
  pour l'interpolation circulaire ou à développante avec G02, G03, CIP, CT, INVCW et INCCCW

  Les polynômes du type 2 pour les coordonnées d'orientation ne sont en revanche possibles que si
  l'interpolation linéaire avec vitesse rapide G00 ou avec avance G01 ou
  l'interpolation polynômiale avec POLY est activée.
- si l'orientation est interpolée au moyen de l'interpolation d'axes ORIAXES. Dans ce cas, on peut programmer directement des polynômes avec PO[A] et PO[B] pour les axes d'orientation A et B.

Polynômes d'orientation du type 1 avec ORIVECT, ORIPLANE et ORICONxx

En cas d'interpolation circulaire de grand rayon et d'interpolation sur enveloppe de cône avec ORIVECT, ORIPLANE et ORICONxx, seuls les polynômes d'orientation du type 1 sont possibles.

Polynômes d'orientation du type 2 avec ORICURVE

Si l'interpolation est activée avec la courbe spatiale supplémentaire ORICURVE, les composantes cartésiennes du vecteur d'orientation sont interpolées et seuls les polynômes d'orientation du type 2 sont possibles.

7.4 Rotations de l'orientation de l'outil (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)

Si l'orientation de l'outil doit pouvoir être modifiée sur les machines avec outil mobile, chaque bloc doit être programmé avec une orientation finale. En fonction de la cinématique de la machine, le sens d'orientation des axes d'orientation ou le sens de rotation du vecteur d'orientation THETA peuvent être programmés. Différents types d'interpolation sont programmables pour ces vecteurs de rotation :

- ORIROTA : Angle de rotation rapporté à un sens de rotation en indication absolue.
- ORIROTR : Angle de rotation relatif par rapport au plan entre orientation de départ et orientation finale.
- ORIROTT : Angle de rotation relatif par rapport à la modification du vecteur d'orientation.
- ORIROTC : Angle de rotation tangentiel à la tangente à la trajectoire.

Syntaxe

A la condition expresse que le type d'interpolation ORIROTA soit actif, l'angle de rotation ou le vecteur de rotation peut être programmé selon l'un des quatre modes suivants :

1. Directement dans les positions d'axes rotatifs `A, B, C`
2. En angles d'Euler (en degrés) par le biais de `A2, B2, C2`
3. En angles RPY (en degrés) par le biais de `A2, B2, C2`
4. En vecteur de direction par le biais de `A3, B3, C3` (angle de rotation à l'aide de `THETA=<valeur>`)

Si ORIOTR ou ORIOTT est actif, l'angle de rotation ne peut être programmé que directement à l'aide de `THETA`.

Une rotation peut également être programmée seule dans un bloc, sans qu'un changement d'orientation ne se produise. ORIROTR et ORIROTT n'ont alors aucune signification. Dans ce cas, l'angle de rotation est toujours interprété par rapport à la direction absolue (ORIROTA).

`N... ORIROTA`	Définition de l'interpolation du vecteur de rotation
`N... ORIROTR`	
`N... ORIROTT`	
`N... ORIROTC`	
`N... A3= B3= C3= THETA=<valeur>`	Définir la rotation du vecteur d'orientation
`N... PO[THT]=(`d_2`, `d_3`, `d_4`, `d_5`)`	Interpolation de l'angle de rotation à l'aide d'un polynôme du cinquième degré

Signification

`ORIROTA` :	Angle de rotation rapporté à un sens de rotation en indication absolue
`ORIROTR` :	Angle de rotation relatif par rapport au plan entre orientation de départ et orientation finale

ORIROTT :	Angle de rotation comme vecteur de rotation tangentiel au changement d'orientation
ORIROTC :	Angle de rotation comme vecteur de rotation tangentiel à la tangente à la trajectoire
THETA :	Rotation du vecteur d'orientation
THETA=<valeur> :	Angle de rotation en degrés atteint en fin de bloc
THETA=Θe :	Angle de rotation avec angle final Θ_e du vecteur de rotation
THETA=AC(…) :	Passage bloc par bloc à l'introduction des cotes en valeurs absolues
THETA=AC(…) :	Passage bloc par bloc à l'introduction des cotes relatives
Θe :	L'angle final du vecteur de rotation est actif aussi bien en valeurs absolues avec G90 qu'en valeurs relatives avec G91 (introduction de cotes relatives).
PO[THT]=(....) :	Polynôme pour l'angle de rotation

Exemple : Rotations des orientations

Code de programme	Commentaire
N10 TRAORI	; Activer la transformation d'orientation
N20 G1 X0 Y0 Z0 F5000	; Orientation de l'outil
N30 A3=0 B3=0 C3=1 THETA=0	; en direction Z avec angle de rotation 0
N40 A3=1 B3=0 C3=0 THETA=90	; en direction X et rotation de 90 degrés
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	; Orientation
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; en direction Y et rotation sur 180 degrés
N70 ORIROTT	; demeure constant et rotation sur 90 degrés
N80 A3=1 B3=0 C3=0 THETA=30	; Angle de rotation par rapport au changement d'orientation
	; Vecteur de rotation formant un angle de 30 degrés par rapport au plan X-Y

Lors de l'interpolation du bloc N40, l'angle de rotation est interpolé de manière linéaire de la valeur de départ 0 degré à la valeur finale 90 degrés. Dans le bloc N50, l'angle de rotation passe de 90 à 180 degrés conformément à la parabole θ(u) = +90u2. En N60, une rotation peut également être effectuée sans qu'il se produise de changement d'orientation.

En N80, l'orientation de l'outil tourne, passant de la direction Y à la direction X. Ce faisant, le changement d'orientation s'opère dans le plan X-Y et le vecteur de rotation forme un angle de 30 degrés par rapport à ce plan.

Informations complémentaires

ORIROTA

L'angle de rotation THETA est interpolé dans l'espace par rapport à une direction déterminée de manière absolue. Le sens de rotation par défaut est fonction des paramètres machine.

ORIROTR

L'angle de rotation THETA est interprété de manière relative par rapport au plan défini par l'orientation initiale et l'orientation finale.

ORIROTT

L'angle de rotation `THETA` est interprété de manière relative par rapport au changement d'orientation. Pour`THETA=0`, le vecteur de rotation est interpolé de manière tangentielle par rapport au changement d'orientation et ne se distingue d'`ORIROTR` que si au moins un polynôme pour l'"angle de basculement PSI" a été programmé pour l'orientation. Il en résulte un changement d'orientation qui ne se produit pas dans le plan. La programmation d'un angle de rotation `THETA` supplémentaire permet d'interpoler par exemple le vecteur de rotation de manière à ce qu'il forme toujours une valeur définie par rapport au changement d'orientation.

ORIROTC

Le vecteur de rotation est interpolé par rapport à la tangente à la trajectoire avec un décalage programmable par le biais de l'angle `THETA` . Pour l'angle de décalage, on peut aussi programmer un polynôme `PO[THT]=(c2, c3, c4, c5)` du 5e degré maximum.

7.5 Orientations par rapport à la trajectoire

7.5.1 Types d'orientation par rapport à la trajectoire

Cette fonction étendue permet d'atteindre l'orientation relative non seulement en fin de bloc, mais aussi sur tout le tracé de la trajectoire. L'orientation atteinte au cours du bloc précédent est transférée au moyen de l'interpolation circulaire de grand rayon dans l'orientation finale programmée. Par principe, il existe deux possibilités pour programmer l'orientation souhaitée par rapport à la trajectoire:

1. L'orientation de l'outil et la rotation de l'outil sont interpolées avec ORIPATH, ORPATHTS par rapport à la trajectoire.
2. Le vecteur d'orientation est programmé et interpolé de la manière habituelle. La rotation du vecteur d'orientation par rapport à la tangente à la trajectoire s'opère avec ORIROTC.

Syntaxe

Le type d'interpolation de l'orientation et de la rotation de l'outil se programme avec:

`N... ORIPATH`	Orientation par rapport à la trajectoire
`N... ORIPATHS`	Orientation par rapport à la trajectoire avec lissage du tracé d'orientation
`N... ORIROTC`	Interpolation par rapport à la trajectoire du vecteur de rotation

Avec `ORIPATHS`, on peut lisser un coude de l'orientation provoqué par un angle dans le tracé de la trajectoire. La direction et la longueur de parcours du mouvement de retrait se programme par le biais du vecteur avec les composants `A8=X, B8=Y C8=Z .`

Avec `ORIPATH/ORIPATHS` , on peut programmer différentes références par rapport à la tangente à la trajectoire par le biais des trois angles

- `LEAD=` indication de l'angle d'avance par rapport à la trajectoire et à la surface
- `TILT=` indication de l'angle latéral par rapport à la trajectoire et à la surface
- `THETA=` angle de rotation

pour l'ensemble du tracé de la trajectoire. En plus de l'angle de rotation `THETA`, on peut aussi programmer des polynômes du 5e degré maximum avec `PO[THT]=(...)`.

Remarque

Constructeur de machine

Veuillez tenir compte des indications du constructeur de machine. Par le biais de paramètres machine et de données de réglage configurables, on peut procéder à d'autres réglages en plus du type d'orientation par rapport à la trajectoire. Pour plus d'explications, se reporter à

Bibliographie :
/FB3/ Description fonctionnelle Fonctions spéciales ; Transformation 3 à 5 axes (F2), Chapitre "Orientation"

Signification

L'interpolation des angles `LEAD` et `TILT` peut se régler différemment par le biais de paramètres machine:

- La référence de l'orientation de l'outil programmée avec `LEAD` et `TILT` est respectée pendant toute la durée du bloc.
- Angle d'avance `LEAD`: Rotation par rapport à la direction perpendiculaire à la tangente et vecteur normal `TILT`: Rotation de l'orientation par rapport au vecteur normal.
- Angle d'avance `LEAD`: Rotation par rapport à la direction perpendiculaire à la tangente et vecteur normal `Angle latéral TILT`: Rotation de l'orientation par rapport à la direction de la tangente à la trajectoire.
- Angle de rotation `THETA`: Rotation de l'outil autour de son propre axe avec un troisième axe rotatif supplémentaire en tant qu'axe d'orientation dans le cas de transformation six axes.

Remarque

Orientation relative à la trajectoire non autorisée avec OSC, OSS, OSSE, OSD, OST

L'interpolation d'orientation par rapport à la trajectoire `ORIPATH` ou `ORIPATHS` et `ORIOTC` ne peut pas être programmée avec le lissage du tracé d'orientation avec l'une des instructions G du groupe 34. Pour ce faire, il faut qu'OSOF soit activé.

7.5.2 Rotation de l'orientation de l'outil relative à la trajectoire (ORIPATH, ORIPATHS, angle de rotation)

Dans le cas d'une transformation six axes, en plus de l'orientation de l'outil d'une manière quelconque dans l'espace, ce dernier peut également tourner autour de son propre axe avec un troisième axe supplémentaire. Dans le cas d'une rotation de l'orientation de l'outil par rapport à la trajectoire avec ORIPATH ou ORIPATHS, la rotation supplémentaire peut se programmer par le biais de l'angle de rotation THETA. En guise d'alternative, les angles LEAD et TILT peuvent être programmés par le biais d'un vecteur perpendiculaire à la direction de l'outil dans le plan.

Constructeur de machine

Veuillez tenir compte des indications du constructeur de machine. L'interpolation des angles LEAD et TILT peut se régler différemment par le biais de paramètres machine.

Syntaxe

Rotation de l'orientation de l'outil et de l'outil

On active le type d'orientation de l'outil par rapport à la trajectoire avec ORIPATH ou ORIPATHS.

`N... ORIPATH`	Activer le type d'orientation par rapport à la trajectoire
`N... ORIPATHS`	Activer le type d'orientation par rapport à la trajectoire avec lissage du tracé d'orientation

Activation des trois angles possibles avec effet de rotation:

`N... LEAD=`	Angle pour l'orientation programmée par rapport au vecteur normal à la surface
`N... TILT=`	Angle pour l'orientation programmée dans le plan perpendiculaire à la tangente à la trajectoire par rapport au vecteur normal à la surface
`N... THETA=`	; Angle de rotation par rapport au changement d'orientation par rapport à la direction d'outil du troisième axe rotatif

Les valeurs des angles en fin de bloc se programment avec `LEAD=valeur`, `TILT=valeur` ou `THETA=valeur`. En plus des angles constants, on peut programmer des polynômes du 5e degré maximum pour les trois angles.

`N... PO[PHI]=(a2, a3, A4, A5)`	Polynôme de l'angle d'avance LEAD
`N... PO[PSI]=(b2, b3, b4, b5)`	Polynôme de l'angle latéral TILT
`N... PO[THT]=(d2, d3, d4, d5)`	Polynôme de l'angle de rotation THETA

Pendant la programmation, on peut laisser de côté les coefficients de polynôme plus élevés, égaux à zéro. Exemple `PO[PHI]=a2` donne une parabole pour l'angle d'avance `LEAD`.

Signification

Orientation de l'outil par rapport à la trajectoire

`ORIPATH` :	Orientation de l'outil par rapport à la trajectoire
`ORIPATHS` :	Orientation de l'outil rapportée à la trajectoire, une cassure dans le tracé de l'orientation est lissée
`LEAD` :	Angle par rapport au vecteur normal à la surface, dans le plan défini par la tangente à la trajectoire et le vecteur normal à la surface
`TILT` :	Rotation de l'orientation par rapport à la direction Z ou rotation par rapport à la tangente à la trajectoire
`THETA` :	Rotation par rapport à la direction de l'outil vers Z
`PO[PHI]` :	Polynôme d'orientation pour l'angle d'avance LEAD
`PO[PSI]` :	Polynôme d'orientation pour l'angle latéral TILT
`PO[THT]` :	Polynôme d'orientation pour l'angle de rotation THETA

Remarque

Angle de rotation THETA

Pour la rotation de l'outil avec un troisième axe rotatif en tant qu'axe d'orientation par rapport à lui-même, une transformation six axes est nécessaire.

7.5.3 Interpolation relative à la trajectoire de la rotation de l'outil (ORIROTC, THETA)

Type d'interpolation avec vecteurs de rotation

En plus de la rotation de l'outil (programmée avec ORIROTC) par rapport à la tangente à la trajectoire, on peut aussi interpoler le vecteur de rotation avec un décalage programmable par le biais de l'angle de rotation THETA. À cette occasion, on peut, avec PO[THT], programmer un polynôme du 5e degré maximum pour l'angle de décalage.

Syntaxe

`N... ORIROTC`	Appliquer la rotation de l'outil par rapport à la tangente à la trajectoire
`N... A3= B3= C3= THETA=valeur`	Définir la rotation du vecteur d'orientation
`N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)`	Interpolation de l'angle de décalage avec polynôme du 5e degré au maximum

Une rotation peut également être programmée seule dans un bloc, sans qu'un changement d'orientation ne se produise.

Signification

Interpolation (par rapport à la trajectoire) de la rotation de l'outil dans le cas d'une transformation six axes

`ORIROTC` :	Appliquer le vecteur de rotation tangentiel à la tangente à la trajectoire
`THETA=valeur` :	Angle de rotation en degrés atteint en fin de bloc
`THETA=`θe :	Angle de rotation avec angle final Θ_e du vecteur de rotation
`THETA=AC(…)` :	Passage bloc par bloc à l'introduction des cotes en valeurs absolues
`THETA=IC(…)` :	Passage bloc par bloc à l'introduction des cotes relatives
`PO[THT]=(c2, c3, c4, c5)` :	Interpolation de l'angle de décalage à l'aide d'un polynôme du 5e degré

Remarque

Interpolation du vecteur de rotation ORIROTC

Si l'on veut appliquer également la rotation de l'outil par rapport à la tangente à la trajectoire dans le sens opposé à la direction d'orientation de l'outil, ce n'est possible que dans le cas d'une transformation six axes.

Dans le cas où ORIROTC est activé

Le vecteur de rotation ORIROTA ne peut pas être programmé. Dans le cas d'une programmation, le message d'alarme 14128 "Programmation absolue de la rotation de l'outil dans le cas où ORIROTC est activé" est émis.

Direction d'orientation de l'outil dans le cas d'une transformation trois à cinq axes

La direction d'orientation de l'outil peut se programmer comme on en a l'habitude avec une transformation trois à cinq axes par le biais d'angles d'Euler ou d'angles RPY ou de vecteurs de direction. Également possibles: changements d'orientation de l'outil dans l'espace par programmation de l'interpolation circulaire de grand rayon ORIVECT, de l'interpolation linéaire des axes d'orientation ORIAXES, de toutes les interpolations sur une enveloppe de cône ORICONxx ainsi que de l'interpolation en plus de la courbe spatiale avec deux points de contact de l'outil ORICURVE.

`G....:`	Indication du type de déplacement des axes rotatifs
`X, Y, Z:`	Indication des axes linéaires
`ORIAXES:`	Interpolation linéaire des axes machine ou d'orientation
`ORIVECT:`	Interpolation circulaire de grand rayon (identique avec ORIPLANE)
`ORIMKS:` `ORIWKS:`	Rotation dans le système de coordonnées machine Rotation dans le système de coordonnées pièce Pour une description, voir le chapitre "Rotations de l'orientation de l'outil"
`A= B= C=:`	Programmation de la position des axes machine
`ORIEULER:`	Programmation d'orientation par le biais d'angles d'Euler
`ORIRPY:`	Programmation de l'orientation par le biais d'angles RPY
`A2= B2= C2=:`	Programmation des angles des axes virtuels
`ORIVIRT1:` `ORIVIRT2:`	Programmation de l'orientation par le biais d'axes d'orientation virtuels (définition 1), selon PM $MC_ORIAX_TURN_TAB_1 (définition 2), selon PM $MC_ORIAX_TURN_TAB_2
`A3= B3= C3=:`	Programmation du vecteur de direction de l'axe d'orientation
`ORIPLANE:`	Interpolation dans le plan (interpolation circulaire de grand rayon)
`ORICONCW:`	Interpolation sur une enveloppe de cône dans le sens horaire
`ORICONCCW:`	Interpolation sur une enveloppe de cône dans le sens anti-horaire
`ORICONTO:`	Interpolation sur une enveloppe de cône avec transition tangentielle
`A6= B6= C6=:`	Programmation de l'axe rotatif du cône (vecteur normé)
`NUT=angle`	Angle au centre du cône en degrés
`NUT=+179`	Angle de déplacement inférieur ou égal à 180 degrés
`NUT=-181`	Angle de déplacement supérieur ou égal à 180 degrés
`ORICONIO:`	Interpolation sur une enveloppe de cône
`A7= B7= C7=:`	Orientation intermédiaire (programmation comme vecteur normé)
`ORICURVE` `XH YH ZH par exemple avec polynôme PO[XH]=(xe, x2, x3, x4, x5)`	Interpolation de l'orientation avec indication du déplacement de deux points de contact de l'outil. Outre le(s) point(s) final(aux) en question, des courbes spatiales supplémentaires sont programmables sous forme de polynômes.

Remarque

Si l'orientation de l'outil est interpolée par le biais des axes d'orientation alors qu'ORIAXES est activé, l'application de l'angle de rotation par rapport à la trajectoire n'est réalisée qu'en fin de bloc.

7.5.4 Lissage du tracé d'orientation (ORIPATHS A8=, B8=, C8=)

Dans le cas de changements d'orientation à accélération continue sur le contour, toute interruption des déplacements avec interpolation, pouvant notamment survenir à un angle du contour, est inopportune. Le coude qui en résulte dans le tracé d'orientation peut être lissé en ajoutant un bloc intermédiaire qui lui est propre. Le changement d'orientation s'effectue alors à accélération continue même si ORIPATHS est actif pendant la réorientation. Pendant cette phase, un mouvement de relèvement de l'outil peut être exécuté.

Constructeur de machine

Veuillez respecter les indications du constructeur de machine en ce qui concerne les paramètres machine et les données de réglage prédéfinis le cas échéant et avec lesquels cette fonction est activée.

Par le biais des paramètres machine, on peut définir comment interpréter le vecteur de relèvement.

1. Dans le système de coordonnées de l'outil, on définit la coordonnée Z par la direction de l'outil.
2. Dans le système de coordonnées pièce, on définit la coordonnée Z par le plan actif.

Pour plus d'explications sur la fonction "Orientation par rapport à la trajectoire", se reporter à **Bibliographie :**
Description fonctionnelle Fonctions spéciales ; Transformations multi-axes (F2)

Syntaxe

Pour des orientations d'outil continues par rapport à l'ensemble de la trajectoire, d'autres indications de programmation sont nécessaires à un angle du contour. La direction et la longueur de parcours de ce mouvement se programme par le biais du vecteur avec les composants A8=X, B8=Y C8=Z:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Signification

`ORIPATHS` :	Orientation de l'outil par rapport à la trajectoire, lissage de coude dans le tracé d'orientation.
`A8= B8= C8=` :	Composants de vecteur pour la direction et la longueur de parcours
`X, Y, Z` :	Mouvement de relèvement dans la direction de l'outil

Remarque

Programmation du vecteur de direction A8, B8, C8

Si la longueur de ce vecteur est égale à zéro, il ne se produit pas de mouvement de relèvement.

ORIPATHS

L'orientation de l'outil par rapport à la trajectoire est activée avec ORIPATHS. Dans le cas contraire, l'orientation est transférée par le biais d'une interpolation circulaire de grand rayon de l'orientation de départ vers l'orientation finale.

7.6 Compactage de l'orientation (COMPON, COMPCURV, COMPCAD, COMPSURF)

Les programmes CN dans lesquels une transformation d'orientation (TRAORI) est active et des orientations d'outil sont programmées (de quelque type que ce soit), peuvent être compactés sous réserve de respecter les tolérances spécifiées.

Programmation

Orientation de l'outil

Si une transformation d'orientation (TRAORI) est active, l'orientation de l'outil peut être programmée de la manière suivante (indépendante de la cinématique) sur les machines à 5 axes :

- Programmation du **vecteur** de direction avec :
  `A3=<...> B3=<...> C3=<...>`
- Programmation des **angles** d'Euler ou des **angles** RPY avec :
  `A2=<...> B2=<...> C2=<...>`

Rotation de l'outil

Sur les machines à **6 axes**, on peut programmer la rotation de l'outil en plus de son orientation.

La programmation de l'angle de rotation s'effectue avec :

`THETA=<...>`

Voir "Rotations de l'orientation de l'outil (Page 366)".

Remarque

Les blocs CN dans lesquels une rotation est programmée en plus ne sont compactables que si l'angle de rotation varie **linéairement**. Donc il ne doit pas y avoir de polynôme programmé avec PO[THT]=(...) pour l'angle de rotation.

Forme générale d'un bloc CN compactable

La forme générale d'un bloc CN compactable peut donc être la suivante :

`N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>`

ou

`N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>`

Remarque

Les valeurs de position peuvent être indiquées directement (par exemple X90) ou indirectement par le biais d'affectations de paramètres (par exemple X=R1*(R2+R3)).

Programmation de l'orientation de l'outil par des positions d'axes rotatifs

L'orientation de l'outil peut également être indiquée par des positions d'axes rotatifs, par exemple sous la forme suivante :

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

Dans ce cas, le compactage est effectué de deux manières différentes selon qu'une interpolation circulaire de grand rayon est exécutée ou non. Si une interpolation circulaire de grand rayon a lieu, le changement d'orientation compacté est représenté de manière habituelle par des polynômes axiaux pour les axes rotatifs.

Précision de contour

Selon le mode de compactage réglé (PM20482 $MC_COMPRESSOR_MODE), les tolérances qui prennent effet pour les axes géométriques et les axes d'orientation lors du compactage sont soit les tolérances spécifiques aux axes (configurées dans le paramètre machine MD33100 $MA_COMPRESS_POS_TOL) ou les tolérances spécifiques aux canaux (réglables dans les données de réglage) suivantes :

SD42475 $SC_COMPRESS_CONTUR_TOL (écart maximal par rapport au contour)

SD42476 $SC_COMPRESS_ORI_TOL (écart maximal par rapport à l'angle d'orientation de l'outil)

SD42477 $SC_COMPRESS_ORI_ROT_TOL (écart maximal par rapport à l'angle de rotation de l'outil) (disponible uniquement pour les machines à 6 axes)

Bibliographie :
Description fonctionnelle Fonctions de base ; Transformation 3 à 5 axes (F2),
chapitre : "Compactage de l'orientation"

Activation/désactivation

Les instructions G modales `COMPON`, `COMPCURV`, `COMPCAD` et `COMPSURF` activent les fonctions compacteur.

La fonction compacteur se termine avec `COMPOF`.

Voir "Compactage de bloc CN (COMPON, COMPCURV, COMPCAD) (Page 273)".

Remarque

Le déplacement d'orientation est uniquement compacté lorsque l'interpolation circulaire de grand diamètre est activée (l'orientation de l'outil se modifie dans le plan défini par l'orientation initiale et l'orientation finale).

Une interpolation circulaire de grand diamètre est exécutée dans les conditions suivantes :

- PM21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  ORIWKS est actif et
  l'orientation est programmée au moyen de vecteurs (avec A3, B3, C3 et A2, B2, C2).
- PM21104 $MC_ORI_IPO_WITH_G_CODE = 1 et
  ORIVECT ou ORIPLANE est actif.
  L'orientation de l'outil peut être programmée sous forme de vecteur de direction ou de positions d'axes rotatifs. Si l'une des instructions G `ORICONxx` et `ORICURVE` est active ou si des polynômes sont programmés pour les angles d'orientation (`PO[PHI]` et `PO[PSI]`), il n'y a pas d'interpolation circulaire de grand rayon.

Exemple

Dans cet exemple de programmation, un cercle, qui est obtenu par une approximation polygonale, est compacté. L'orientation de l'outil se déplace de façon synchrone sur une surface latérale de cône. Bien que les changements successifs d'orientation programmés soient discontinus, la fonction compacteur génère un tracé lisse de l'orientation.

Programmation	**Commentaire**
DEF INT NOMBRE=60	
DEF REAL RAYON=20	
DEF INT COMPTEUR	
DEF REAL ANGLE	
N10 G1 X0 Y0 F5000 G64	
$SC_COMPRESS_CONTUR_TOL=0.05	; Ecart maximal par rapport au contour = 0,05 mm
$SC_COMPRESS_ORI_TOL=5	; Ecart maximal par rapport à l'orientation = 5 degrés
TRAORI	
COMPCURV	; Le déplacement a lieu sur un cercle obtenu par une approximation polygonale. L'orientation se déplace sur un cône autour de l'axe Z avec un angle au centre de 45 degrés.
N100 X0 Y0 A3=0 B3=-1 C3=1	
N110 FOR COUNTER=0 TO NOMBRE	
N120 ANGLE=360*COUNTER/NOMBRE	
N130 X=RAYON*cos(ANGLE) Y=RAYON*sin(ANGLE)	
A3=sin(ANGLE) B3=-cos(ANGLE) C3=1	
N140 ENDFOR	

7.7 Activation/désactivation du lissage du tracé d'orientation (ORISON, ORISOF)

Le "Lissage du tracé d'orientation" est activé/désactivé dans le programme pièce par les instructions du groupe G 61. Les instructions sont des fonctions à effet modal.

Conditions requises

- Système avec transformation 5/6 axes.
- La fonction de compresseur COMPCAD est active.

Syntaxe

```
ORISON
...
ORISOF
```

Signification

ORISON :	Activation du lissage du tracé d'orientation
ORISOF :	Désactivation du lissage du tracé d'orientation

Exemple

Code de programme	Commentaire
...	
TRAORI()	; Activation de la transformation d'orientation.
COMPCAD	; Activation de la fonction de compresseur COMPCAD.
ORISON	; Activation du lissage d'orientation.
$SC_ORISON_TOL=1.0	; Écart maximum par rapport à l'angle d'orientation de l'outil = 1,0 degré.
G91	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
...	
ORISOF	; Désactivation du lissage d'orientation.
...	

Un pivotement de l'orientation de 90 degrés est réalisé dans le plan XZ, de -45 à +45 degrés. En raison du lissage de son tracé, l'orientation n'atteint plus les valeurs angulaires maximales de -45 ou +45 degrés.

7.8 Transformation cinématique

7.8.1 Activer la transformation de surfaces d'attaque (TRANSMIT)

La transformation de face frontale (TRANSMIT) est activée dans le programme pièce ou dans une action synchrone par l'instruction `TRANSMIT`.

Syntaxe

```
TRANSMIT
TRANSMIT(<n>)
```

Signification

`TRANSMIT` :	Activer TRANSMIT avec le premier bloc de données TRANSMIT
`TRANSMIT(n)` :	Activation de TRANSMIT avec le <n>-ième bloc de données TRANSMIT

Remarque

Une transformation TRANSMIT active dans le canal est désactivée par :

- Désactivation de la transformation : `TRAFOOF`
- Activation d'une autre transformation : p. ex. `TRACYL`, `TRAANG`, `TRAORI`

7.8.2 Activer la transformation de surfaces cylindriques (TRACYL)

La transformation de surface cylindrique (TRACYL) est activée dans le programme pièce ou dans une action synchrone par l'instruction `TRACYL`.

Syntaxe

```
TRACYL(<d>)
TRACYL(<d>,<n>)
TRACYL(<d>,<n>,<k>)
```

Signification

`TRACYL(<d>)` :	Activation de TRACYL avec le premier bloc de données TRACYL et le diamètre d'usinage <d>
`TRACYL (<d>,<n>)` :	Activation de TRACYL avec le <n>-ième bloc de données TRACYL et le diamètre d'usinage <d>
`<d>` :	Diamètre de référence ou d'usinage La valeur doit être supérieure à 1.

<n> :	Numéro de bloc de données TRACYL (facultatif)	
	Plage de valeurs :	1, 2
<k> :	Le paramètre <k> est seulement pertinent pour le type de transformation 514	
	k = 0 :	sans correction des flancs de rainure
	k = 1 :	avec correction des flancs de rainure
	Si le paramètre n'est pas spécifié, la position de base paramétrée s'applique : $MC_TRACYL_DEFAULT_MODE_<n> avec <n> = numéro de bloc de données TRACYL	

Remarque

Une transformation TRACYL active dans le canal est désactivée par :

- Désactivation de la transformation : `TRAFOOF`
- Activation d'une autre transformation : p. ex. `TRAANG`, `TRANSMIT`, `TRAORI`

Exemple

Code de programme	Commentaire
...	
N40 TRACYL(40.)	; Activation de TRACYL avec le premier bloc de données TRACYL et le diamètre d'usinage 40 mm.
...	

Informations complémentaires

Structure du programme

Un programme pièce pour le fraisage d'une rainure avec la transformation TRACYL 513 (TRACYL avec correction du flanc de rainure) comprend généralement les étapes suivantes :

1. Sélection de l'outil.
2. Sélection de TRACYL.
3. Sélection du décalage du système de coordonnées approprié (FRAME).
4. Positionnement.
5. Programmation d'OFFN.
6. Sélection de la CRO (correction du rayon d'outil).
7. Bloc d'accostage (accostage de la CRO et accostage du flanc de la rainure).
8. Contour correspondant à l'axe de symétrie de la rainure.
9. Désactivation de la CRO.
10. Bloc de retrait (retrait de la CRO et éloignement du flanc de rainure).

11. Positionnement.

12. TRAFOOF.

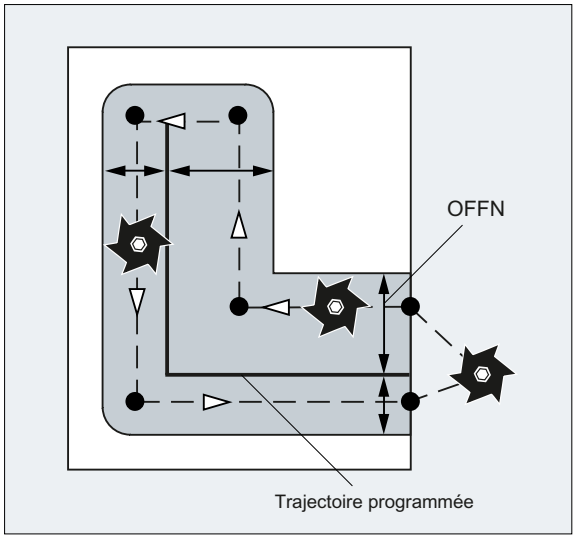
13. Reprise du décalage du système de coordonnées initial (FRAME).

Offset sur contour (OFFN)

Pour fraiser des rainures avec la transformation TRACYL 513, programmer dans le programme pièce l'axe de symétrie de la rainure et **la demi-largeur** à l'aide de l'adresse OFFN.

Afin d'éviter toute détérioration du flanc de rainure, OFFN n'est activé que lorsque la correction du rayon d'outil a été sélectionnée.

Il est possible de modifier OFFN à l'intérieur du programme pièce. Ainsi, vous pouvez décentrer la ligne médiane de la rainure :



Remarque

OFFN doit être au moins égal au rayon d'outil, afin d'empêcher toute détérioration du flanc de rainure opposé.

Remarque

OFFN avec TRACYL a un comportement différent de OFFN sans TRACYL. Il est recommandé de remettre à zéro OFFN après TRAFOOF, du fait que OFFN est également pris en compte sans TRACYL lorsque la correction de rayon d'outil CRO est activée.

IMPORTANT
L'effet d'OFFN dépend du type de transformation Avec la transformation TRACYL 513 (TRACYL avec correction du flanc de rainure), programmer la demi-largeur de la rainure pour OFFN. Par contre, avec la transformation TRACYL 512 (TRACYL sans correction du flanc de rainure), la valeur d'OFFN a pour effet de produire une surépaisseur qui s'ajoute à la correction de rayon d'outil (CRO).

Correction du rayon d'outil (CRO)

Avec la transformation TRACYL 513, la correction du rayon d'outil n'est pas calculée par rapport au flanc de rainure mais par rapport à l'axe de symétrie programmé de la rainure. Pour que l'outil se déplace à gauche du flanc de rainure, il convient de programmer l'instruction G42 à la place de G41 ou d'indiquer la valeur d'OFFN avec un signe négatif.

Diamètre de l'outil

Avec TRACYL et un outil dont le diamètre est inférieur à la largeur de la rainure, la géométrie de flanc de rainure générée n'est pas la même qu'avec un outil dont le diamètre est égal à la largeur de la rainure. Pour améliorer la précision, il est conseillé de choisir un diamètre d'outil à peine inférieur à la largeur de la rainure.

Utilisation des axes

Remarque

Les axes suivants ne peuvent pas être utilisés comme axes de positionnement ou d'oscillation :

- l'axe géométrique dans le sens de la circonférence de l'enveloppe du cylindre (axe Y)
- l'axe linéaire supplémentaire en cas de correction des flancs de rainure (axe Z)

7.8.3 Activation de la transformation à angle oblique avec angle programmable (TRAANG)

La transformation à angle oblique avec angle programmable est activée dans le programme pièce ou dans une action synchrone par l'instruction `TRAANG`.

Syntaxe

```
TRAANG
TRAANG()
TRAANG(, <n>)
TRAANG(<α>)
TRAANG(<α>,<n>)
```

Signification

`TRAANG` : `TRAANG()` :	Activation de TRAANG avec le premier bloc de données TRAANG et le dernier angle <α> valide	
`TRAANG(,<n>)` :	Activation de TRAANG avec le <n>-ième bloc de données TRAANG et le dernier angle <α> valide	
`TRAANG(<α>)` :	Activation de TRAANG avec le premier bloc de données TRAANG et l'angle <α>	
`TRAANG(<α>,<n>)` :	Activation de TRAANG avec le <n>-ième bloc de données TRAANG et l'angle <α>	
<α> :	Angle de l'axe incliné (facultatif)	
	Plage de valeurs :	-90° < α < + 90°
	Si aucun angle n'est spécifié, la position de base paramétrée dans le paramètre machine s'applique : PM2xxxx $MC_TRAANG_ANGLE_<n>	
`<n>` :	Numéro de bloc de données TRAANG (facultatif)	
	Plage de valeurs :	1, 2

Remarque

Une transformation à angle oblique `TRAANG` active dans le canal est désactivée par :

- Désactivation de la transformation : `TRAFOOF`
- Activation d'une autre transformation : p. ex. `TRACYL`, `TRANSMIT`, `TRAORI`

Exemple

Code de programme	Commentaire
`N20 TRAANG(45)`	`; Activation de TRAANG avec le premier bloc de données TRAANG et l'angle 45°`

7.8.4 Plongée oblique sur les rectifieuses (G5, G7)

Les instructions G G7 et G5 permettent de simplifier la programmation pour la plongée oblique sur les rectifieuses avec transformation "axe incliné (TRAANG)", de manière que seul l'axe incliné se déplace lors de la plongée.

Pour la programmation, seule la position finale souhaitée du mouvement de plongée se trouve dans X et Z. La position de départ associée est calculée et accostée par la CN pour G7, en partant de la position actuelle de l'axe X, de la position finale programmée et de l'angle α de l'axe incliné.

La position de départ résulte du point d'intersection des deux droites :

- Droite parallèle à l'axe Z à la distance de la position actuelle de l'axe X
- Droite parallèle à l'axe incliné passant par la position finale programmée

L'axe incliné se déplacera à la position finale programmée avec la fonction G5 suivante.

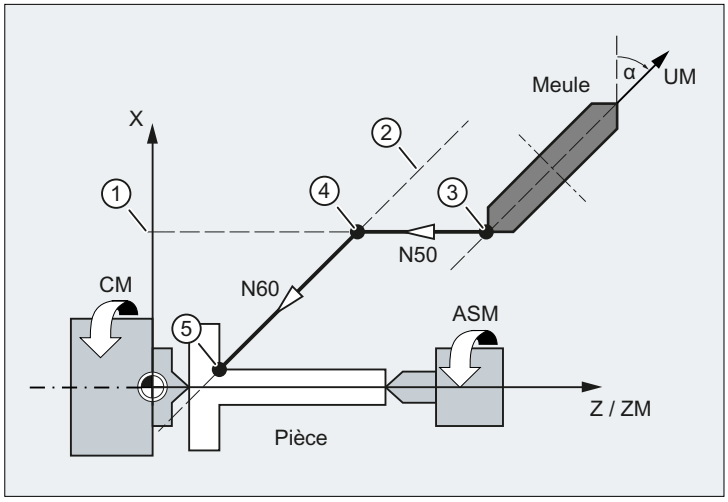
Syntaxe

```
G7 <PosFinale_X> <PosFinale_Z>
G5 <PosFinale_X>
```

Signification

G7 :	Calcul et accostage de la position de départ pour la plongée oblique
G5 :	Déplacement de l'axe incliné à la position finale programmée
<PosFinale_X> :	Position finale de l'axe X
<PosFinale_Z> :	Position finale de l'axe Z

Exemple



① Parallèle à l'axe Z à la distance de la position actuelle de l'axe X
② Parallèle à l'axe incliné passant par la position finale programmée
③ Position de départ
④ Plongée : position de départ
⑤ Plongée : position finale
X Axe géométrique
Z Axe géométrique
ZM Axe machine
UM Axe machine

Code de programme	**Commentaire**
N... G18	; Sélection du plan XZ
N40 TRAANG(45.0)	; Activation de la transformation TRAANG, angle = 45°.
N50 G7 X40 Z70 F4000	; Calcul et accostage de la position de départ.
N60 G5 X40 F100	; Déplacement de l'axe incliné à la position finale.
N70 ...	

7.9 Activer la transformation concaténée (TRACON)

Une transformation concaténée est activée dans le programme pièce ou dans une action synchrone par l'instruction TRACON.

Syntaxe

```
TRACON(<N° de transf>,<Par_1>,<Par_2...)
...
TRAFOOF
```

Signification

TRACON :	Activation de la transformation concaténée TRACON() désactive implicitement toute transformation active auparavant.		
<N° de transf> :	Numéro de la transformation concaténée		
	Type :	INT	
	Plage de valeurs :	0 ... 2	
	Valeur :	0, 1	Première/unique transformation concaténée
		2	Deuxième transformation concaténée
		Pas d'indication	Même signification que 0 ou 1
		Remarque : Les valeurs différentes de 0, 1, 2 génèrent une alarme.	
<Par_1>,<Par_2...> :	Paramètres pour les transformations concaténées (p. ex. angle de l'axe incliné) Si des paramètres ne sont pas programmés, les préréglages ou les valeurs utilisées en derniers lieu sont actifs. Les virgules sont nécessaires pour que les paramètres soient traités dans l'ordre dans lequel ils sont escomptés, si des préréglages doivent être pris en considération pour certains paramètres antéposés. En particulier, si un seul paramètre est indiqué, il convient de placer une virgule devant celui-ci, même si l'indication du <N° de transf> est inutile, soit par exemple TRACON(, 3.7). Dans le cas de TRACON avec Transmit ou TRAORI (usinage 5 axes), le deuxième paramètre Par_2 n'agit pas comme angle pour l'axe oblique. Dans le cas de TRACON avec TRACYL (usinage de surfaces latérales), Par est utilisé pour le rayon unitaire.		
TRAFOOF :	Désactivation de la dernière transformation (concaténée) activée		

Exemple

```
Code de programme            Commentaire
...
```

Code de programme	**Commentaire**
N230 TRACON(1,45.)	; Activation de la première transformation concaténée. ; La transformation active précédemment est automatiquement désélectionnée. ; Le paramètre pour l'axe incliné est 45°.
...	
N330 TRACON(2,40.)	; Activation de la deuxième transformation concaténée. ; Le paramètre pour l'axe incliné est 40°.
...	
N380 TRAFOOF	; Désactivation de la deuxième transformation concaténée.
...	

7.10 Déplacement PTP cartésien

7.10.1 Activer/désactiver le déplacement PTP cartésien (PTP, PTPG0, PTPWOC, CP)

Le déplacement point à point ou PTP cartésien est activé/désactivé dans le programme CN avec les instructions du groupe G 49.

Les instructions sont des fonctions à effet modal. Le réglage par défaut est le déplacement cartésien avec interpolation (CP).

Contrairement à la méthode CP, lorsque le déplacement PTP est actif, seule la transformation du point de destination cartésien est exécutée et les axes machine se déplacent de manière synchrone.

Pour que le point de destination cartésien puisse être converti de façon univoque en valeurs d'axes machine, d'autres informations caractérisant les positions des axes sont nécessaires en plus des indications de position et d'angle. Ces indications s'effectuent via les adresses réglables STAT (Page 390) et TU (Page 394).

Condition

La transformation TRAORI, TRANSMIT, RCTRA ou ROBX est active.

Syntaxe

```
PTP / PTPG0 / PTPWOC
...
CP
```

Signification

`PTP` :	Activer le déplacement point à point PTP La position cartésienne programmée dans les blocs G0 et G1 est accostée avec un déplacement d'axe synchrone.
`PTPG0` :	Activer le déplacement point à point PTPG0 La position cartésienne programmée est accostée avec un déplacement d'axe synchrone uniquement dans les blocs G0. Dans les blocs G1, le déplacement avec interpolation CP est utilisé.
`PTPWOC` :	Activation du déplacement point à point PTPWOC (possible uniquement lorsque la transformation d'orientation est active) Comme PTP, mais sans les mouvements compensatoires provoqués par les déplacements d'axes rotatifs et d'axes d'orientation.
`CP` :	Désactiver le déplacement point à point et activer le déplacement avec interpolation CP Avec CP, un déplacement cartésien avec interpolation est exécuté.

Remarque

PTPWOC

En association avec une transformation RCTRA ou ROBX, l'utilisation de PTPWOC n'a pas de sens !

Exemples

Voir :

- Exemple 1 : Déplacement PTP d'un robot à 6 axes avec la transformation ROBX (Page 397)
- Exemple 2 : Déplacement PTP avec une transformation générique 5 axes (Page 398)
- Exemple 3 : PTPG0 et TRANSMIT (Page 398)

7.10.2 Indiquer la position des articulations (STAT)

L'indication de position avec des coordonnées cartésiennes et l'indication de l'orientation de l'outil ne suffisent pas pour déterminer de manière univoque la position de la machine, étant donné que plusieurs positions d'articulation sont possibles avec la même orientation d'outil. Selon le type de cinématique, 8 positions d'articulation différentes sont possibles. Ces différentes positions d'articulation sont spécifiques à la transformation.

Pour éviter les ambiguïtés, la position des articulations est indiquée sous l'adresse STAT.

Remarque

La commande prend en compte les valeurs STAT programmées uniquement lors des déplacements PTP. Lors des déplacements CP, elles sont ignorées car un changement de position n'est normalement pas possible lors du déplacement avec la transformation active. Lors du déplacement avec CP actif, la position du point de destination est reprise du point de départ.

Syntaxe

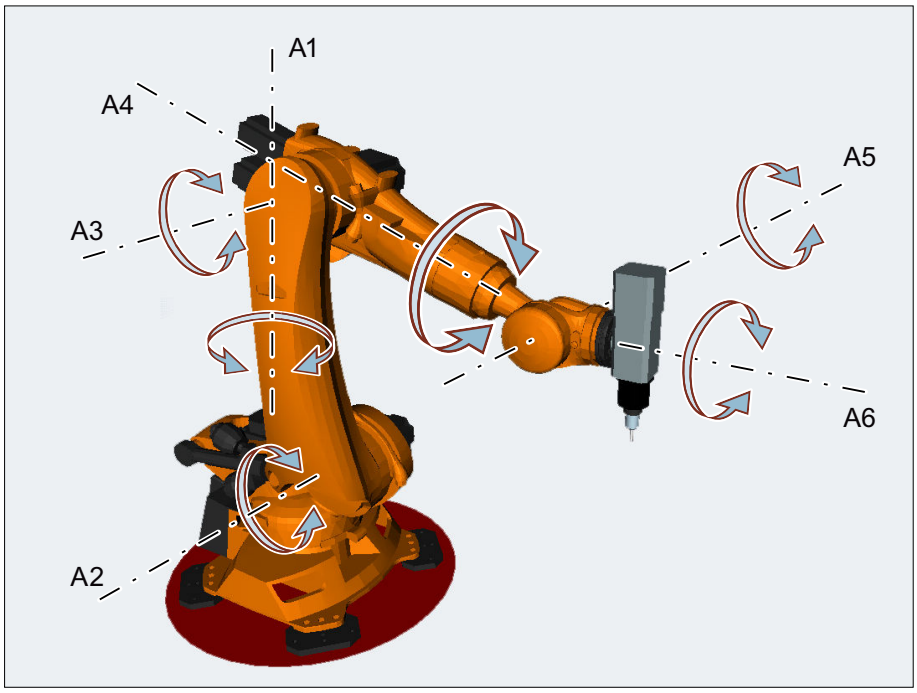
```
STAT=<valeur>
```

Signification

`STAT` :	Adresse réglable pour l'indication des positions d'articulation
`<valeur>` :	Valeur binaire ou décimale Contient un bit pour chacune des positions possibles. La signification des bits est définie par la transformation en question.

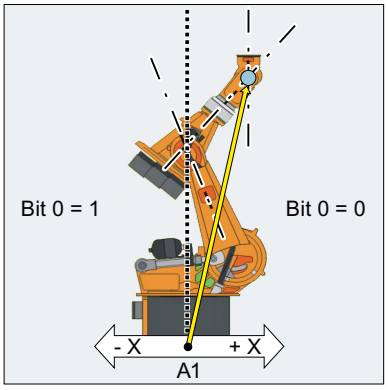
L'utilisation de STAT doit être illustrée en prenant l'exemple d'un robot anthropomorphe à 6 axes avec broche porte-fraise. La transformation cinématique doit s'effectuer via la

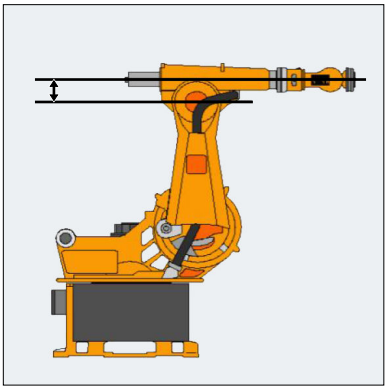
transformation robotique ROBX (condition : le cycle de compilation "RMCC/ROBX Transformation robotique étendue" est chargé et actif).



Les axes A1, A2 et A3 sont les axes principaux du robot anthropomorphe. Les axes A4, A5 et A6, qui sont appelés axes de tête ou de poignet, sont positionnés dans la zone de travail avec les axes principaux. Grâce aux possibilités de déplacement supplémentaires des axes de poignet, la broche porte-fraise peut être orientée dans la zone de la manière requise pour la tâche d'usinage. Ainsi, avec la même orientation de l'outil, différentes positions d'articulation sont possibles.

La sélection des positions d'articulation à utiliser pour l'usinage s'effectue en programmant le bit 0 ... 2 de l'adresse STAT réglable :

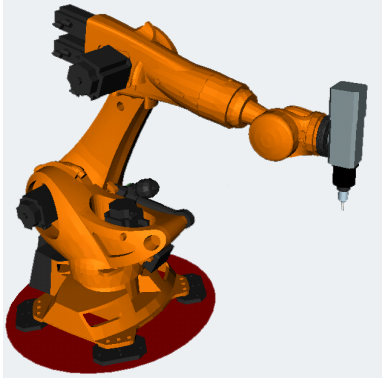
Bit 0	Position du point d'intersection des axes de poignet (A4, A5, A6)		
	= 0	Zone de base (**Shoulder Right**) Le robot se trouve dans la zone de base lorsque la valeur X du point d'intersection des axes de poignet, rapportée au système de coordonnées A1, est positive.	 Exemple : Le point d'intersection des axes de poignet se trouve dans la zone de base
	= 1	Zone de plafond (**Shoulder Left**) Le robot se trouve dans la zone de plafond lorsque la valeur X du point d'intersection des axes de poignet, rapportée au système de coordonnées A1, est négative.	

Bit 1	Position de l'axe 3 L'angle auquel la valeur du bit 1 change dépend du type de robot. Pour les robots dont les axes 3 et 4 se coupent, on a :	
	= 0	A3 < 0° (**Elbow Down**)
	= 1	A3 ≥ 0° (**Elbow Up**)
	Remarque : Pour les robots avec un décalage entre les axes 3 et 4, l'angle auquel la valeur du bit 1 change dépend de la valeur de ce décalage.	 Offset entre A3 et A4
Bit 2	Position de l'axe 5	
	= 0	A5 ≤ 0° (**No Handflip**)
	= 1	A5 > 0° (**Handflip**)

Exemple de programme :

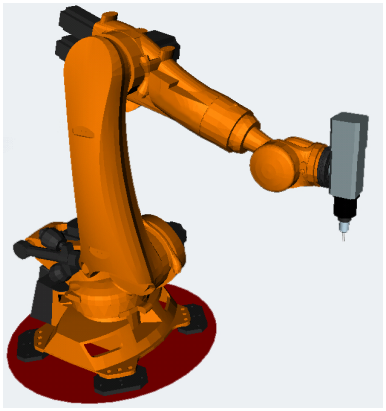
Code de programme	**Commentaire**
...	
N14 T="T8MILLD20" D1	; $TC_DP3[1,1]=132.95
N16 ORIMKS	
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 **STAT=...** F2000	; La valeur STAT détermine les positions d'articulation (voir ci-dessous).
...	

STAT=1 ('B001') → Shoulder Left
→ Elbow Down
→ No Handflip



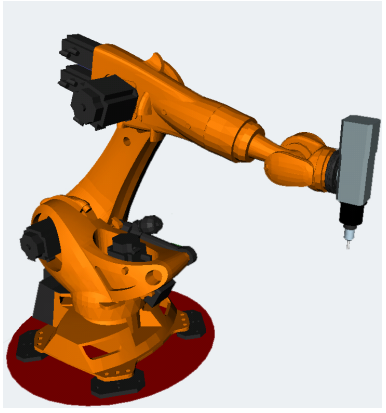
`STAT=2 ('B010')`

→ Shoulder Right
→ Elbow Up
→ No Handflip



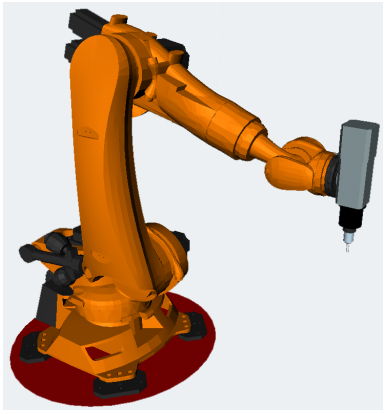
`STAT=5 ('B101')`

→ Shoulder Left
→ Elbow Down
→ Handflip



`STAT=6 ('B110')`

→ Shoulder Right
→ Elbow Up
→ Handflip



TRANSMIT

Avec TRANSMIT, l'adresse STAT est utilisée pour éviter les ambiguïtés en ce qui concerne le pôle.

Lorsque l'axe rotatif doit tourner de 180º ou lorsque le contour passerait par le pôle avec CP, on a :

Bit 0	Pertinent uniquement quand $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 ou 2 :	
	= 0	L'axe rotatif se déplace de +180º ou tourne dans le sens des aiguilles d'une montre.
	= 1	L'axe rotatif se déplace de -180º ou tourne dans le sens inverse des aiguilles d'une montre.
Bit 1	Pertinent uniquement quand $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0 :	
	= 0	Il y a un passage par le pôle. L'axe rotatif ne tourne pas.
	= 1	Il y a une rotation autour du pôle. Le bit 0 de STAT est alors pertinent.

7.10.3 Indiquer les signes des angles d'axe (TU)

Avec les axes rotatifs, pour pouvoir accoster également les angles d'axe supérieurs à +180° ou inférieurs à -180° sans stratégie de déplacement particulière (par ex. points intermédiaires), les signes des angles d'axe doivent être indiqués sous l'adresse réglable TU.

Remarque

La commande prend en compte les valeurs TU programmées uniquement lors des déplacements PTP. Lors des déplacements CP, elles sont ignorées.

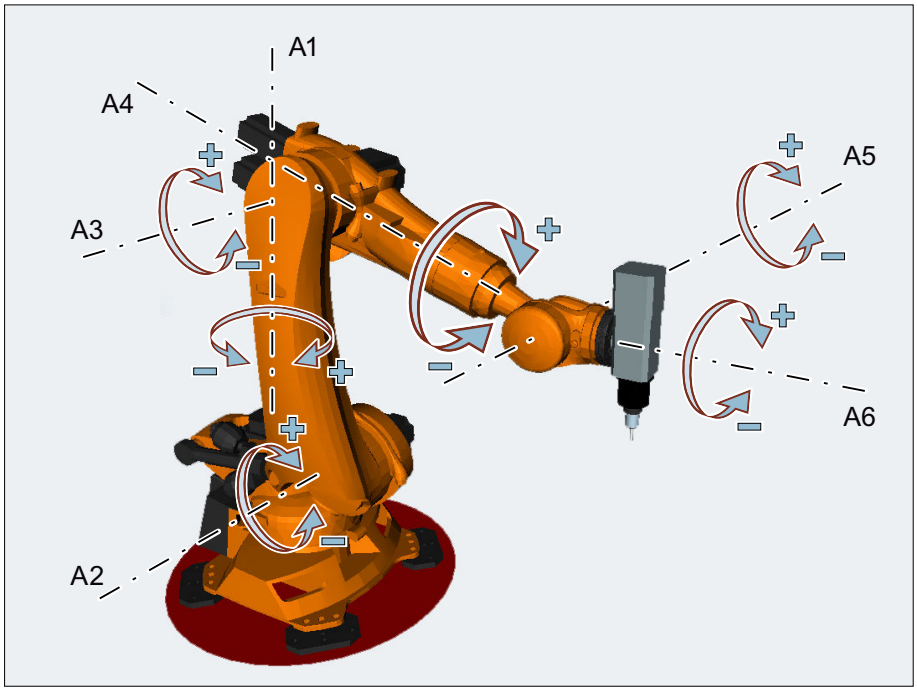
Syntaxe

```
TU=<valeur>
```

Signification

`TU` :	Adresse réglable pour l'indication des signes des angles d'axe			
`<valeur>` :	Valeur binaire ou décimale Pour chaque axe qui entre dans la transformation, un bit indique le signe de l'angle d'axe (θ) et par conséquent le sens de déplacement.			
	Bit	= 0	Signe de l'angle d'axe : +	Plage de l'angle d'axe : 0° ≤ θ < 360°
		= 1	Signe de l'angle d'axe : -	Plage de l'angle d'axe : - 360° < θ < 0°

Exemple : Robot anthropomorphe à 6 axes



Bit	Signification	Valeur	Signe de l'angle d'axe	Angle d'axe
Bit 0 [1]	Signe pour l'angle d'axe d'A1	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 1 [1]	Signe pour l'angle d'axe d'A2	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 2 [1]	Signe pour l'angle d'axe d'A3	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 3 [1]	Signe pour l'angle d'axe d'A4	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 4 [1]	Signe pour l'angle d'axe d'A5	= 0	+	≥ 0°
		= 1	-	< 0°
Bit 5 [1]	Signe pour l'angle d'axe d'A6	= 0	+	≥ 0°
		= 1	-	< 0°

[1] Les numéros de bit TU réels sont obtenus à partir des numéros d'axe de canal des axes du robot ! Dans l'exemple, les axes du robot (A1 à A6) sont les six premiers axes du canal, par conséquent les bits TU 0 ... 5 sont utilisés. En cas d'affectation différente des axes de canal aux axes du robot, les numéros de bit TU des axes du robot changeraient en conséquence (par ex. : les axes du robot sont les axes 3 à 8 du canal, c'est-à-dire que les bits TU 2 ... 7 sont utilisés pour les axes du robot).

TU=19 (correspond à TU='B010011) signifierait également :

Bit	Valeur		Angle d'axe
0	= 1	⇒	$\theta_{A1} < 0°$
1	= 1	⇒	$\theta_{A2} < 0°$

2	= 0	⇒	$\theta_{A3} \geq 0°$
3	= 0	⇒	$\theta_{A4} \geq 0°$
4	= 1	⇒	$\theta_{A5} < 0°$
5	= 0	⇒	$\theta_{A6} \geq 0°$

Remarque

Pour les axes avec une plage de déplacement > ±360°, le déplacement se fait toujours par le trajet le plus court, car la position des axes ne peut pas être déterminée de manière univoque au moyen des informations TU.

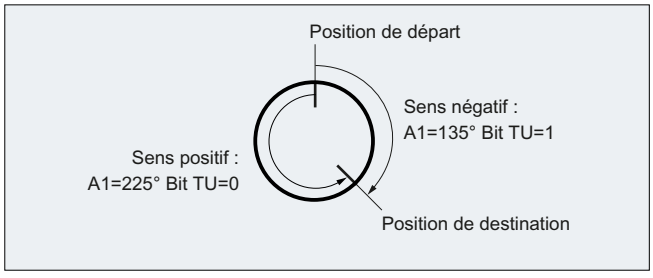
Si TU n'est pas programmé pour une position, le déplacement se fait par le trajet le plus court ou le plus long en fonction du PM30455 $MA_MISC_FUNCTION_MASK (voir chapitre "Prise en compte des limites logicielles lors du déplacement PTP" dans la Description fonctionnelle Fonctions d'extension).

TRANSMIT

Lors du déplacement PTP avec TRANSMIT actif, l'adresse TU est sans importance.

Exemple

La position d'axes rotatifs indiquée dans l'illustration suivante peut être accostée en sens négatif ou positif. La position d'angle est programmée sous l'adresse A1. Le sens de déplacement ne devient univoque qu'avec l'indication TU.



7.10.4 Exemple 1 : Déplacement PTP d'un robot à 6 axes avec la transformation ROBX

Dans l'exemple d'application suivant, le déplacement PTP cartésien et les instructions CN s'y rapportant sont illustrés de manière exemplaire.

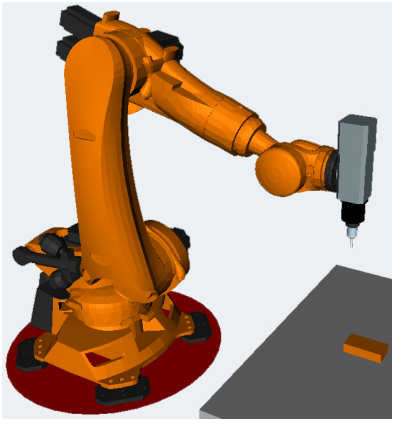


Figure 7-3 Robot anthropomorphe à 6 axes avec broche porte-fraise

```
N1 G90
N2 T="T8MILLD20" D1 M6
N3 TRAORI
;$P_UIFR[1]=CTRANS(X,1500,Y,0,Z,400):CROT(X,0,Y,0,Z,-90)
N4 G54
N5 M3 S20000
N6 ORIWKS
N7 ORIVIRT1
N8 CYCLE832(0.01,_FINISH,1)
;HOME
N9 TRAFOOF
N10 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N11 TRAORI
N12 G54
N13 G0 PTP X1369.2426 Y956.7528 Z502.5517 A=135.5761 B=-33.2223 C=161.1435
STAT='B010' TU='B001011'
N14 G0 X1355.1242 Y1014.9394 Z424.9695 A=135.8491 B=-33.1439 C=160.9941
STAT='B010' TU='B001011'
N15 G1 CP X1354.8361 Y1016.1269 Z423.3862 A=136.0635 B=-33.0819 C=160.8770
F1000
N16 G1 X1336.4283 Y1016.1269 Z426.6311 A=136.0484 B=-32.2151 C=160.9643
F2000
N17 G1 X1317.9831 Y1016.1269 Z429.6730 A=136.0175 B=-31.3394 C=161.0655
;HOME
N18 TRAFOOF
N19 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N20 M30
```

7.10.5 Exemple 2 : Déplacement PTP avec une transformation générique 5 axes

Hypothèse : Une cinématique orthogonale CA sert de base.

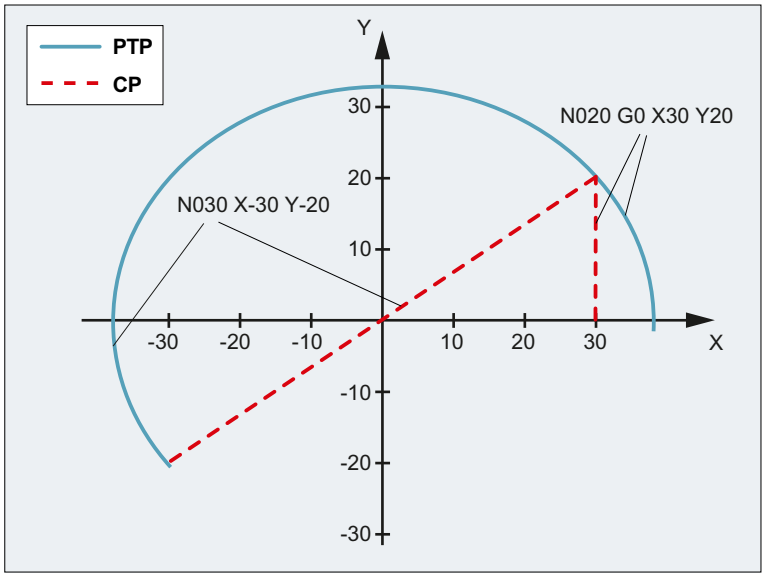
Code de programme	Commentaire
TRAORI	; Transformation de cinématique CA activée
PTP	; Activer le déplacement PTP
N10 A3=0 B3=0 C3=1	; Positions des axes rotatifs C=0 A=0
N20 A3=1 B3=0 C3=1	; Positions des axes rotatifs C=90 A=45
N30 A3=1 B3=0 C3=0	; Positions des axes rotatifs C=90 A=90
N40 A3=1 B3=0 C3=1 **STAT=1**	; Positions des axes rotatifs C=270 A=–45

Sélectionner la position d'accostage univoque de l'axe rotatif :

Dans le bloc N40, les axes rotatifs parcourent, du fait de la programmation **STAT=1**, le trajet le plus long de leur point de départ (C=90, A=90) à leur point final (C=270, A=–45). Au lieu de cela, avec **STAT=0**, les axes rotatifs parcourraient le trajet le plus court jusqu'au point final (C=90, A=45).

7.10.6 Exemple 3 : PTPG0 et TRANSMIT

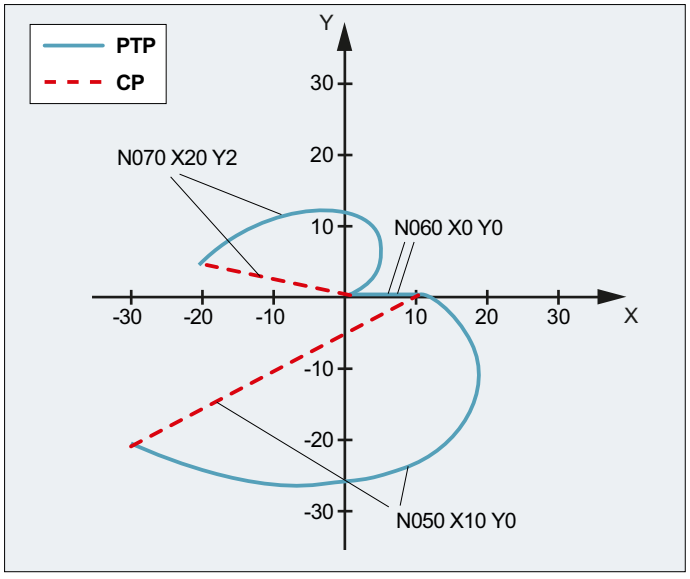
Contournement du pôle avec PTPG0 et TRANSMIT



Code de programme	Commentaire
N001 G0 X30 Z0 F10000 T1 D1 G90	; Position de départ Cote absolue
N002 SPOS=0	
N003 **TRANSMIT**	; Transformation TRANSMIT
N010 **PTPG0**	; PTP automatiquement pour chaque bloc G0 puis à nouveau CP

Code de programme	Commentaire
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Dégagement du pôle avec PTPG0 et TRANSMIT



Programmation	Commentaire
N001 G0 X90 Z0 F10000 T1 D1 G90	; Position de départ
N002 SPOS=0	
N003 **TRANSMIT**	; Transformation TRANSMIT
N010 **PTPG0**	; PTP automatiquement pour chaque bloc G0 puis à nouveau CP
N020 G0 X90 Y60	
N030 X-90 Y-60	
N040 X-30 Y-20	
N050 X10 Y0	
N060 X0 Y0	
N070 X-20 Y2	
N170 G1 X0 Y0	
N160 X10 Y0	
N150 X-30 Y-20	
M30	

7.11 Conditions marginales pour l'activation d'une transformation

Fonction

La sélection de transformations est possible par le biais du programme pièce ou en mode MDA. Attention :

- Aucun bloc intermédiaire de déplacement n'est inséré (chanfreins/rayons).
- Une séquence de blocs de type spline doit être terminée, sinon un message s'affiche.
- La correction fine d'outil doit être désactivée (FTOCOF), sinon un message s'affiche.
- La correction de rayon d'outil doit être désactivée (G40), sinon un message s'affiche.
- Une correction de longueur d'outil activée est reprise dans la transformation par la commande.
- Le frame courant activé avant la transformation est désactivé par la commande.
- Une limitation active de la zone de travail est désactivée par la commande pour les axes impliqués dans la transformation (correspond à WALIMOF).
- La surveillance de la zone de protection est désactivée.
- Le contournage et l'arrondissement sont interrompus.
- Tous les axes indiqués dans les paramètres machine doivent être synchronisés au niveau des blocs.
- Les axes permutés sont rétablis, sinon un message apparaît.
- Un message apparaît en présence d'axes dépendants.

Changement d'outil

Un changement d'outil n'est autorisé que si la correction de rayon d'outil est désactivée.

Un changement de la correction de longueur d'outil et une activation/ désactivation de la correction de rayon d'outil ne doivent pas être programmés dans le même bloc.

Changement de frame

Toutes les instructions qui se rapportent uniquement au système de coordonnées de base sont autorisées (FRAME, correction de rayon d'outil). Un changement de frame avec G91 (cotes relatives) ne sera cependant pas traité à part, contrairement à ce qui se passe lorsque la transformation est inactive. L'incrément à parcourir est traité dans le système de coordonnées pièce du nouveau frame, quel que soit le frame qui était actif dans le bloc précédent.

Exclusions

Les axes impliqués dans une transformation ne peuvent pas être utilisés :

- comme axes Preset (alarme),
- pour l'accostage de point fixe (alarme),
- pour la prise de référence (alarme).

7.12 Désactivation de la transformation (TRAFOOF)

La procédure prédéfinie TRAFOOF permet de désactiver tous les frames et transformations actifs.

Remarque

Les conditions marginales (Page 400) pour l'activation d'une transformation s'appliquent également à la désactivation d'une transformation.

Les frames nécessaires doivent ensuite être activés par une nouvelle programmation.

Syntaxe

```
...
TRAFOOF
```

Signification

`TRAFOOF` :	Désactiver l'ensemble des transformations/frames actifs

Chaînes cinématiques 8

8.1 Suppression de composants (DELOBJ)

La fonction `DELOBJ()` "supprime" des composants en réinitialisant les variables système affectées sur leur valeur par défaut :

- Eléments de chaînes cinématiques
- Zones de protection, éléments de zone de protection et paires de collision
- Données de transformation

Syntaxe

```
[<RetVal>=] DELOBJ(<CompType>[,,,<NoAlarm>)])
[<RetVal>=] DELOBJ(<CompType>,<Index1>[,,<NoAlarm>])
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][,<Index2>][,<NoAlarm>])
```

Signification

`DELOBJ:`	Suppression d'éléments de chaînes cinématiques, de zones de protection, d'éléments de zones de protection, de paires de collision et de données de transformation

<CompType> :	Type du composant à supprimer	
	Type de données :	STRING
	Valeur : "KIN_CHAIN_ELEM" **Signification** : Variables système de tous les éléments cinématiques : $NK_...	
	Valeur : "KIN_CHAIN_SWITCH" **Signification** : Variable système $NK_SWITCH[<i>]	
	Valeur : "KIN_CHAIN_ALL" **Signification** : Tous les éléments et commutateurs cinématiques. Même signification que l'appel successif de DELOBJ avec "KIN_CHAIN_ELEM" et "KIN_CHAIN_SWITCH"	
	Valeur : "PROT_AREA" **Signification** : Variables système des zones de protection : • $NP_PROT_NAME • $NP_CHAIN_NAME • $NP_CHAIN_ELEM • $NP_1ST_PROT	
	Valeur : "PROT_AREA_ELEM" **Signification** : Variables système des éléments de zone de protection des zones de protection pour machine et/ou des zones de protection automatiques pour outils : • $NP_NAME • $NP_NEXT • $NP_NEXTP • $NP_COLOR • $NP_D_LEVEL • $NP_USAGE • $NP_TYPE • $NP_FILENAME • $NP_PARA • $NP_OFF • $NP_DIR • $NP_ANG	
	Valeur : "PROT_AREA_COLL_PAIRS" **Signification** : Variables système des paires de collision : • $NP_COLL_PAIR • $NP_SAFETY_DIST	
	Valeur : "PROT_AREA_ALL" **Signification** : L'ensemble des zones de protection, des éléments de zone de protection et des paires de collision (variables système $NP_...) Même signification que l'appel successif de DELOBJ avec "PROT_AREA", "PROT_AREA_ELEM" et "PROT_AREA_COLL_PAIRS"	
	Valeur : "TRAFO_DATA" **Signification** : Variables système de toutes les transformations $NT_...	

<Index1> :	Indice du premier composant à supprimer (**facultatif**)	
	Type de données :	INT
	Valeur par défaut :	-1
	Plage de valeurs :	-1 ≤ x ≤ (nombre maximal de composants configurés -1)
	Valeur	**Signification**
	0, 1, 2, ...	Indice du composant à supprimer.
	-1	Tous les composants du type indiqué sont supprimés. <Index2> n'est pas exploité.
<Index2> :	Indice des derniers composants à supprimer (**facultatif**) Si <Index2> n'est pas programmé, seules les variables système du composant référencé dans <Index1> sont supprimées.	
	Type de données :	INT
	Valeur par défaut :	Seules les variables système du composant référencé dans <Index1> sont supprimées.
	Plage de valeurs :	<Index1> < x ≤ (nombre max. de composants configurés -1)
<NoAlarm> :	Inhibition des alarmes (**facultatif**)	
	Type de données :	BOOL
	Valeur par défaut :	FALSE
	Valeur	**Signification**
	FALSE	En cas d'erreur (<RetVal> < 0), l'exécution du programme est arrêtée et une alarme est affichée.
	TRUE	En cas d'erreur, l'exécution du programme n'est pas arrêtée et aucune alarme n'est affichée. Cas d'application : réaction personnalisée en fonction de la valeur de retour
<RetVal> :	Valeur de retour de la fonction	
	Type de données :	INT
	Plage de valeurs :	0, -1, -2, ... -7
	Valeur	**Signification**
	0	Aucune erreur n'est survenue.
	-1	Appel de la fonction sans paramètre. Au moins le paramètre <CompType> doit être spécifié.
	-2	<CompType> désigne un composant inconnu
	-3	<Index1> est inférieur à -1
	-4	<Index1> est supérieur au nombre de composants configuré
	-5	<Index1> a une valeur différente de -1 lors de la suppression d'un **groupe** de composants
	-6	<Index2> est inférieur à <Index1>
	-7	<Index2> est supérieur au nombre de composants configuré

8.2 Détermination de l'indice par le nom (NAMETOINT)

Des noms personnalisés sont saisis dans des tableaux de variables système du type STRING. Au moyen du descripteur des variables système et du nom, la fonction `NAMETOINT()` détermine la valeur de l'indice correspondant au nom, sous lequel il est enregistré dans le tableau de variables système.

Syntaxe

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

Signification

`NAMETOINT` :	Détermination de l'indice de variable système	
`<SysVar>` :	Nom du tableau de variables système du type STRING	
	Type de données :	STRING
	Plage de valeurs :	Noms de tous les tableaux de variables système de la CN du type STRING
`<Name>` :	Chaîne de caractères ou nom pour lequel/laquelle l'indice de variable système doit être déterminé.	
	Type de données :	STRING
`<NoAlarm>` :	Inhibition des alarmes (**facultatif**)	
	Type de données :	BOOL
	Valeur par défaut :	FALSE
	Valeur	**Signification**
	TRUE	En cas d'erreur, l'exécution du programme n'est pas arrêtée et aucune alarme n'est affichée. Cas d'application : réaction personnalisée en fonction de la valeur de retour
	FALSE	En cas d'erreur (<RetVal> < 0), l'exécution du programme est arrêtée et une alarme est affichée.
`<RetVal>` :	Indice de variables système ou message d'erreur	
	Type de données :	INT
	Plage de valeurs :	-1 ≤ x ≤ (nombre max. de composants configurés -1)
	Valeur	**Signification**
	≥ 0	Le nom recherché a été trouvé sous l'indice de variables système indiqué.
	-1	Le nom recherché n'a pas été trouvé ou une erreur est survenue.

Exemple

Code de programme	**Commentaire**
`DEF INT INDEX`	
`$NP_PROT_NAME[27] = "Recouvrement"`	
`...`	
`INDEX = NAMETOINT("$NP_PROT_NAME","Recouvrement")`	`; INDEX == 27`

Prévention des collisions avec chaînes cinématiques 9

Remarque

Zones de protection

Les zones de protection évoquées dans les chapitres suivants se rapportent à la fonction "Modélisation géométrique de la machine"

Bibliographie :
Description fonctionnelle Fonctions spéciales, chapitre "Modélisation géométrique de la machine"

9.1 Contrôle de la présence de paire de collision (COLLPAIR)

La fonction `COLLPAIR(...)` détermine si deux zones de protection forment une paire de collision.

Syntaxe

```
[<RetVal> =] COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>)])
```

Signification

`COLLPAIR`:	Vérification de l'appartenance à une paire de collision		
`<RetVal>`:	Valeur de retour de la fonction		
	Type de données :	INT	
	Valeur :	≥ 0	Les deux zones de protection forment une paire de collision. Valeur de retour == Indice de paire de collision m (voir $NP_COLL_PAIR)
		-1	Soit moins de deux chaînes ont été spécifiées, soit l'une des deux est une chaîne vide.
		-2	La zone de protection spécifiée dans le premier paramètre n'a pas été trouvée.
		-3	La zone de protection spécifiée dans le second paramètre n'a pas été trouvée.
		-4	Aucune des deux zones de protection spécifiées n'a été trouvée.
		-5	Les deux zones de protection spécifiées ont été trouvées, mais pas ensemble dans une paire de collision.
`<Name_1>`:	Nom de la première zone de protection		
	Type de données :	STRING	
	Plage de valeurs :	Noms de zone de protection paramétrés	
`<Name_2>`:	Nom de la seconde zone de protection		
	Type de données :	STRING	
	Plage de valeurs :	Noms de zone de protection paramétrés	

<NoAlarm>:	Inhibition des alarmes **(facultatif)**		
	Type de données :	BOOL	
	Valeur :	FALSE (Par défaut)	En cas d'erreur (<RetVal> < 0), l'exécution du programme est arrêtée et une alarme est affichée.
		TRUE	En cas d'erreur, l'exécution du programme n'est pas arrêtée et aucune alarme n'est affichée. Cas d'application : réaction personnalisée en fonction de la valeur de retour

9.2 Demander un nouveau calcul du modèle de machine pour la prévention des collisions (PROTA)

Lorsque des variables système $NK_..., de la modélisation géométrique de la machine ou de la prévention de collisions $NP_... sont écrites dans le programme pièce, la procédure `PROTA` doit ensuite être appelée afin de rendre effectives les modifications du modèle de machine interne à la CN pour la prévention de collisions.

Syntaxe

```
PROTA[(<Par>)]
```

Signification

`PROTA`:	Demander un nouveau calcul du modèle de machine pour la prévention des collisions		
	• Déclenche un arrêt du prétraitement des blocs. • Doit figurer seule dans le bloc.		
`<Par>`:	Paramètres (**option**)		
	Type de données :	STRING	
	Valeur :	---	Sans paramètre. Un nouveau calcul du modèle de machine est effectué. Les états des zones de protection restent inchangés.
		"R"	Un nouveau calcul du modèle de machine est effectué. Les zones de protection sont mises dans leur état d'initialisation en fonction de $NP_INIT_STAT.

Autres conditions à prendre en compte

Simulation

La procédure `PROTA` ne doit pas être utilisée dans des programmes pièce en association avec la simulation (simNC).

Exemple : Empêchement de l'appel de `PROTA` lorsque la simulation est active.

```
Code de programme                Commentaire
...
IF $P_SIM == FALSE               ; IF simulation inactive
   PROTA                         ; THEN recalculer le modèle de collision
ENDIF                            ; ENDIF
...
```

Voir aussi

Réglage de l'état des zone de protection (PROTS) (Page 413)

9.3 Réglage de l'état des zone de protection (PROTS)

La procédure `PROTS(...)` règle l'état des zones de protection sur la valeur indiquée.

Syntaxe

```
PROTS(<State>[,<Name_1>,...,<Name_n>])
```

Signification

`PROTS` :	Réglage de l'état des zones de protection		
	• Doit figurer seule dans le bloc.		
`<State>` :	Etat sur lequel les zones de protection spécifiées doivent être réglées		
	Type de données :	CHAR	
	Valeur :	"A" ou "a"	Etat : active
		"I" ou "i"	Etat : inactive
		"P" ou "p"	Etat : préactivé ou commandé par AP [1]
		"R" ou "r"	État : Valeur de l'état d'initialisation interne à la CN [2)]
`<Name_1> ... <Name_n>` :	Nom d'une ou de plusieurs zones de protection à régler sur l'état spécifié (**option**) Si aucun nom n'est spécifié, l'état spécifié est réglé pour l'ensemble des zones de protection définies.		
	Type de données :	STRING	
	Plage de valeurs :	Noms de zone de protection paramétrés	
	Remarque : Le nombre maximal de zones de protection pouvant être spécifiées en paramètre dépend uniquement du nombre maximal possible de caractères par ligne de programme.		

[1)] L'activation/désactivation est effectuée via : DB10.DBX234.0 - DBX241.7

[2)] L'état est réglé sur la valeur de l'état d'initialisation interne à la CN, c.-à-d. sur la valeur de la variable système $NP_INIT_STAT à l'instant du dernier appel de PROTA(...) (Page 412).

9.4 Détermination de la distance entre deux zones de protection (PROTD)

La fonction `PROTD(...)` calcule la distance entre deux zones de protection.

Caractéristiques de la fonction :

- Le calcul de la distance est effectué indépendamment de l'état des zones de protection (activée, désactivée, pré-activée).
- Pour le calcul des distances de deux zones de protection, seuls les éléments de zone de protection marqués par $NP_USAGE = "C" ou "A" peuvent être utilisés. Les éléments de la zone de protection marqués par $NP_USAGE = "V" seront ignorés.
- Les zones de protection dont tous les éléments sont marqués par $NP_USAGE = "V" ne peuvent pas être utilisées pour le calcul des distances.
- Le calcul de la distance est effectué avec les positions valides à la fin du bloc précédent.
- Les chevauchements qui sont pris en compte dans l'exécution des blocs, p. ex. décalage DRF ou décalages d'origine externes, entrent dans le calcul de la distance avec les valeurs valides au **moment de l'interprétation** de la fonction.

Remarque

Synchronisation

Lorsque la fonction `PROTD(...)` est utilisée, l'utilisateur assume l'entière responsabilité quant à une synchronisation de l'exécution et du prétraitement des blocs le cas échéant au moyen d'un arrêt du prétraitement des blocs `STOPRE`.

Collision

En présence d'une collision entre les zones de protection spécifiées, la fonction présente une distance de 0,0. Une collision existe lorsque les deux zones de protection se touchent ou interfèrent.

La distance de sécurité pour le contrôle de collision (PM10622 $MN_COLLISION_SAFETY_DIST) n'est pas prise en compte dans le calcul de la distance.

Syntaxe

```
[<RetVal> =] PROTD([<Name_1>],[<Name_2>],VAR <Vector>[,<System>])
```

Signification

`PROTD:`	Calcul de la distance entre les deux zones de protection spécifiées	
	• Doit figurer seule dans le bloc.	
`<RetVal>:`	Valeur en retour de la fonction : Valeur absolue de la distance entre les deux zones de protection ou 0,0 en cas de collision (voir ci-dessus : paragraphe Collision)	
	Type de données :	REAL
	Plage de valeurs :	0,0 ≤ x ≤ +max. Valeur REAL

<Name_1>, <Name_2>:	Noms des deux zones de protection dont la distance les séparant doit être calculée (**option**)		
	Type de données :	STRING	
	Plage de valeurs :	Noms de zone de protection paramétrés	
	Valeur par défaut :	"" (chaîne vide) Si aucune zone de protection n'est spécifiée, la fonction calcule la distance actuellement la plus faible pour toutes les zones de protection activées et pré-activées contenues dans le modèle de collision.	
<Vector>:	Valeur de retour : vecteur de distance tridimensionnel de la zone de protection <Name_2> à la zone de protection <Name_1> avec : • <Vector>[0]: Cordonnée X dans le système de coordonnées mondial • <Vector>[1]: Cordonnée Y dans le système de coordonnées mondial • <Vector>[2]: Cordonnée Z dans le système de coordonnées mondial En cas de collision : <Vector> == vecteur nul		
	Type de données :	VAR REAL [3]	
	Plage de valeurs :	<Vector> [n]: 0,0 ≤ x ≤ ±valeur REAL max.	
<System>:	Système de mesure (anglo-saxon / métrique) pour la distance et le vecteur de distance (**facultatif**)		
	Type de données :	BOOL	
	Valeur :	FALSE (Par défaut)	Système de mesure correspondant à l'instruction G actuellement active du groupe G 13 (G70, G71, G700, G710).
		TRUE	Système de mesure correspondant au système de base réglé : PM52806 $MN_ISO_SCALING_SYSTEM

Transformations avec chaînes cinématiques 10

10.1 Activation de la transformation (TRAFOON)

Une transformation avec chaînes cinématiques est activée par la procédure prédéfinie TRAFOON. L'appel doit être isolé dans un bloc.

Remarque

Variante : une transformation définie avec chaînes cinématiques peut également être activée à l'aide des instructions conventionnelles, par ex. TRAORI ou TRANSMIT. Une valeur différente de zéro doit être renseignée à cet effet dans la variable système $NT_TRAFO_INDEX.

Pour plus d'informations sur $NT_TRAFO_INDEX, voir "Tables de paramètres Variables système".

Syntaxe

```
TRAFOON(<Trafoname>,<Diameter>,<k>)
```

Signification

TRAFOON:	Procédure d'activation d'une transformation avec chaînes cinématiques		
<Trafoname> :	Nom du jeu de paramètres de transformation		
	Type de données :	STRING	
	Plage de valeurs :	Tous les noms de jeu de paramètres de transformation définis dans $NT_NAME	
	Remarque : Le nom du jeu de paramètres de transformation doit être univoque Il ne doit être présent dans $NT_NAME qu'une seule fois.		
<Diameter> :	Diamètre de référence ou d'usinage (uniquement TRACYL)		
	Type de données :	REAL	
	La valeur doit être > 1.		
<k> :	Définit l'utilisation de la correction des flancs de rainure (uniquement TRACYL).		
	Type de données :	BOOL	
	Valeur :	FALSE	sans correction des flancs de rainure
		TRUE	avec correction des flancs de rainure
	Correspond au type de transformation TRACYL 514 (correction des flancs de rainure programmable). Si <k> n'est pas renseigné, le réglage paramétré de bit 10 dans $NT_CNTRL[<n>] est effectif.		

Exemple

Code de programme	Commentaire
TRAFOON["Trans_1"]	Active la transformation du nom Trans_1.

10.2 Modification de la transformation d'orientation après mesurage de la machine (CORRTRAFO)

Dans le cas des machines avec transformations d'orientation définies à l'aide de chaînes cinématiques, l'utilisateur peut utiliser la fonction prédéfinie CORRTRAFO, permettant, après un mesurage de la machine, de modifier les vecteurs d'offset ou les vecteurs directionnels des axes d'orientation dans le modèle cinématique de la machine.

Syntaxe

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode> [,
<No_Alarm>])
```

Signification

CORRTRAFO:	Appel de fonction

<Corr_Status> :	Valeur de retour de la fonction		
	Type de données :	INT	
	Valeurs :	0	La fonction a été exécutée sans erreur.
		1	Aucune transformation n'est active.
		2	La transformation actuellement active n'est pas une transformation d'orientation.
		3	La transformation d'orientation active n'a pas été définie à l'aide de chaînes cinématiques.
		10	Le paramètre d'appel <Corr_Index> est négatif.
		11	Le paramètre d'appel <Corr_Mode> est négatif.
		12	Renvoi non valide sur une section d'une chaîne partielle (position des unités dans <Corr_Index>). La valeur ne doit pas être supérieure au nombre d'axes d'orientation dans la chaîne partielle.
		13	Renvoi non valide sur l'axe d'orientation d'une chaîne partielle (position des unités dans <Corr_Index>). La valeur doit être inférieure au nombre d'axes d'orientation dans la chaîne partielle.
		14	Renvoi non valide sur une chaîne partielle (position des dizaines dans <Corr_Index>). Seules les valeurs 0 et 1 sont autorisées (renvoi sur une chaîne Part ou une chaîne Tool). Ce numéro d'erreur est également signalé lorsque la chaîne partielle figurant dans le renvoi <Corr_Index> n'existe pas.
		15	Aucun élément de correction n'est défini dans la section référencée par le paramètre <Corr_Index> ($NT_CORR_ELEM_P ou $NT_CORR_ELEM_T).
		20	Mode de correction non valide (position des unités de <Corr_Mode>). Seules les valeurs 0 et 1 sont autorisées.
		21	Mode de correction non valide (position des dizaines et/ou des centaines de <Corr_Mode>). Lors de l'écriture d'une direction d'axe, seule la position des unités peut être égale à zéro.
		30	La position des centaines de <Corr_Mode> n'est pas valide. Seules les valeurs 0 et 1 sont autorisées.
		31	La position des milliers de <Corr_Mode> n'est pas valide. Seules les valeurs 0 et 1 sont autorisées.
		40	Le vecteur directionnel devant être appliqué comme direction de l'axe est un vecteur nul. Cette erreur ne peut être signalée que si la position des milliers de <Corr_Mode> est égale à 0. Si la position des milliers de ce paramètre est égale à 1 (surveillance de la correction maximale désactivée), un vecteur nul peut également être écrit.
		41	Lors de la correction d'un vecteur d'offset, l'écart dans au moins une coordonnée par rapport à la valeur actuelle est supérieur à la valeur maximale spécifiée par la donnée de réglage SD41610 $SN_CORR_TRAFO_LIN_MAX. Le paramètre <Corr_Vect> est alors écrasé par un vecteur d'erreur. Ceci s'applique également lorsque l'exécution est annulée avec une alarme (voir paramètre <No_Alarm>). Les composantes du vecteur d'erreur, dont la valeur de correction a dépassé la limite autorisée, contiennent la différence

			(avec le signe correct) entre la valeur de correction déterminée et la limite. Le contenu des composantes n'ayant pas dépassé leur limite est égal à zéro.
		42	Lors de la correction d'un vecteur directionnel, l'écart angulaire par rapport à la direction actuelle est supérieur à la valeur maximale spécifiée par la donnée de réglage SD41611 $SN_CORR_TRAFO_DIR_MAX.
		43	La tentative d'écrire une variable système a été refusée pour cause de droits d'écriture insuffisants.
<Corr_Vect> :	Vecteur de correction Le contenu du vecteur de correction est défini par les paramètres <Corr_Index> et <Corr_Mode> suivants. Si <Corr_Status> = 41, le contenu du vecteur est écrasé (voir ci-dessus).		
	Type de données :	REAL	
<Corr_Index> :	Section dont l'élément de correction doit être modifié / indice de l'axe d'orientation dont le vecteur directionnel doit être modifié		
	Type de données :	INT	
	Le paramètre <Corr_Index> est codé en décimal (position des unités et dizaines) :		
	Position des unités :	Contient l'indice de la section ou de l'axe d'orientation dans la chaîne partielle.	
	Position des dizaines :	Identifie la chaîne partielle.	
		0x	Chaîne pièce
		1x	Chaîne outil

`<Corr_Mode>` :	Mode de correction		
	Type de données :	INT	
	Le paramètre `<Corr_Mode>` est codé en décimal (position des unités jusqu'à la position des milliers) :		
	Position des unités :	Définit l'élément à corriger.	
		xxx**0**	Correction d'un vecteur d'offset linéaire
		xxx**1**	Correction d'un vecteur directionnel d'un axe d'orientation
	Position des dizaines :	Définit comment l'élément de correction référencé par le contenu de `<Corr_Index>` doit être modifié.	
		xx**0**x	Le vecteur de correction est directement écrit dans l'élément de correction. Cette variante peut être utilisée pour écrire directement l'élément de correction sans que l'indice <n> de la donnée système ($NK_OFF_DIR[<n>, ...]) soit nécessairement connu.
		xx**1**x	Comme 0, avec la différence que la valeur de correction transmise est interprétée en coordonnées mondiales. Une différence entre les variantes 0 et 1 peut apparaître lorsque la chaîne cinématique contient d'autres rotations en position de base (position de tous les axes d'orientation égale à 0).
		xx**2**x	Comme 1, avec la différence que la valeur de correction s'applique sur la totalité de la section. Autrement dit, la valeur renseignée dans l'élément de correction est telle que la totalité de la section atteint la longueur définie par la valeur de correction.
		Remarque : Les valeurs 1 et 2 ne sont pas autorisées pour l'écriture du vecteur directionnel d'un axe d'orientation.	
	Position des centaines :	Définit comment le contenu du paramètre `<Corr_Vect>` doit être interprété.	
		x**0**xx	Le vecteur de correction transmis `<Corr_Vect>` contient la totalité de la nouvelle longueur de l'élément de correction ou de la section référencée par `<Corr_Index>` en association avec la position des dizaines de `<Corr_Mode>` (correction absolue).
		x**1**xx	Le vecteur de correction transmis `<Corr_Vect>` ne contient que la différence par rapport à la longueur actuelle de l'élément de correction ou de la section référencée par `<Corr_Index>` en association avec la position des dizaines de `<Corr_Mode>` (correction incrémentale).
		Remarque : Pour la correction du vecteur directionnel d'un axe d'orientation, le contenu de la position des centaines doit être égal à 0.	
	Position des milles :	Définit si la correction doit être limitée par la valeur maximale suivante : • SD41610 $SN_CORR_TRAFO_LIN_MAX ou • SD41611 $SN_CORR_TRAFO_DIR_MAX	
		0xxx	La surveillance de la correction maximale est active.

		1xxx	La surveillance de la correction maximale n'est pas active.
<No_Alarm> :	Comportement en cas d'erreur (valeur retournée > 0) (**facultatif**)		
	Type de données :	BOOL	
	Valeur :	FALSE (Default)	En cas d'erreur, l'exécution du programme est abandonnée et l'alarme 14103 est émise.
		TRUE	En cas d'erreur, l'exécution du programme n'est pas arrêtée et aucune alarme n'est affichée. Cas d'application : réaction personnalisée en fonction de la valeur de retour

Remarque

Si une erreur se produit lors de l'appel de la fonction, une alarme est émise ou un numéro d'erreur est retourné (voir paramètre <No_Alarm>), permettant à l'utilisateur de réagir de manière appropriée à l'état d'erreur. La cause de l'erreur est définie plus en détail par un paramètre d'alarme. Un numéro d'erreur retourné au lieu d'une alarme est identique au paramètre d'alarme.

Plus d'informations sur CORRTRAFO

La structure cinématique d'une machine avec transformation d'orientation est décrite par une ou deux chaînes cinématiques (chaînes partielles) qui partent de l'origine du système de coordonnées mondiales. L'une des deux chaînes, la **chaîne outil**, se termine au point de référence de l'outil, l'autre, la **chaîne pièce** dans l'origine du système de coordonnées de base.

La fonction CORRTRAFO mesure les longueurs de bras de levier et les directions d'axe des machines comportant une transformation d'orientation et écrit ces valeurs dans des éléments de correction spéciaux. Une chaîne cinématique est décrite entre autres par des éléments de type OFFSET définis avec $NK_TYPE.

CORRTRAFO utilise des sections

Les deux chaînes partielles peuvent être divisées chacune en quatre sections au maximum :

- La section 1 commence au point de départ de la chaîne et se termine au premier axe d'orientation.
- La section 2 est la section comprise entre l'axe d'orientation 1 et l'axe d'orientation 2.
- La section 3 est la section comprise entre l'axe d'orientation 2 et l'axe d'orientation 3.
- La section 4 est la section comprise entre l'axe d'orientation 3 et la fin de la chaîne outil ou de la chaîne pièce.

Les sections peuvent contenir chacune des éléments de chaîne constants de type OFFSET ou ROT_CONST.

Le graphique suivant présente une transformation d'orientation avec 2 axes d'orientation.

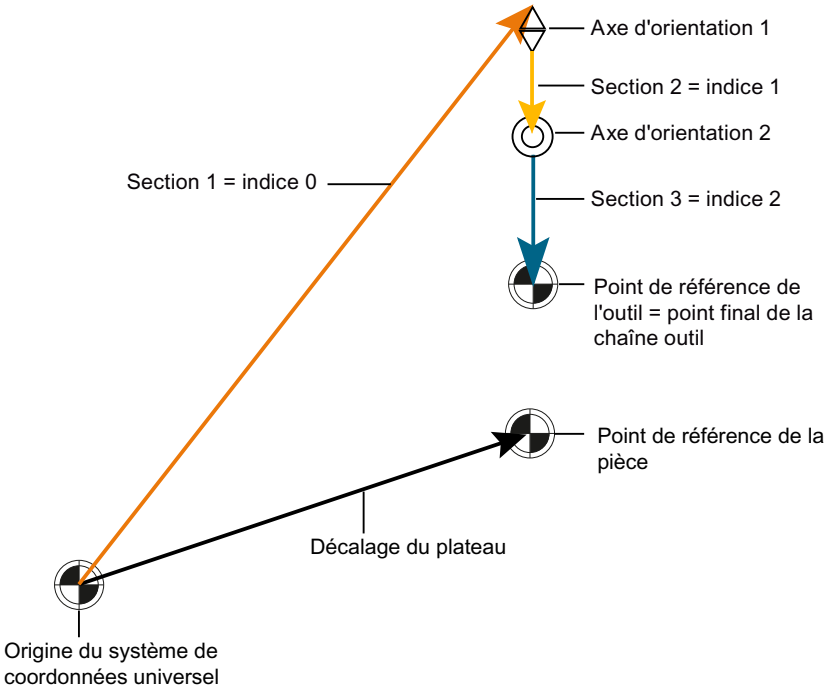


Figure 10-1 Exemple CORRTRAFO

Les sections sont définies de manière univoque : Lorsqu'on parcourt la chaîne cinématique partielle depuis son point de départ jusqu'à la fin, la première section comporte l'indice 0, la suivante l'indice 1, etc. L'indice de la dernière section est par conséquent toujours égal au nombre d'axes d'orientation.

Éléments de correction

Dans chacune de ces sections, la variable système $NT_CORR_ELEM_T[<n>, 0 ... 3] ou $NT_CORR_ELEM_P[<n>, 0 ... 3] permet de référencer un élément constant de chaîne cinématique du type $NK_TYPE[<n>] = "OFFSET". Les valeurs de correction déterminées lors du mesurage de la machine sont écrites avec la fonction CORRTRAFO dans ces éléments ainsi désignés.

Exemple avec indice de transformation = 1 :

- $NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET" ; Le décalage de l'axe C (axe d'orientation 1), dans la section 1, est défini comme élément de correction.
- $NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET" ; Le décalage de l'axe C (axe d'orientation 2), dans la section 2, est défini comme élément de correction.
- $NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"; Le décalage de l'axe B par rapport au point de référence d'outil, dans la section 3, est défini comme élément de correction.

L'ordre des références dans $NT_CORR_ELEM_T/P[<n>, 0 ... 3] doit correspondre aux sections susmentionnées. Autrement dit, $NT_CORR_ELEM_T/P [<n>, 0] ne peut contenir qu'un élément de chaîne se trouvant avant le premier axe d'orientation, etc.

Dans les éléments de correction ainsi définis, la fonction CORRTRAFO écrit les valeurs déterminées par le mesurage de la machine. La modification des valeurs de correction est définie dans CORRTRAFO par le paramètre <Corr_Mode>.

Fermeture de la chaîne

Lorsque le bit 7 ou le bit 8 sont mis à 1 dans la variable système $NT_CNTRL[<n>], des éléments de chaîne constants sont insérés automatiquement en interne à la fin de la chaîne pièce (bit 7) ou avant le point de départ de la chaîne outil (bit 8). Ces éléments forment une connexion entre la fin de la chaîne et l'origine machine ("Fermeture de la chaîne").

Ces éléments insérés automatiquement ne sont pas accessibles en écriture externe mais uniquement en lecture (voir également les variables système $AC_TRAFO_CORR_ELEM_P/T).

Point de fermeture de la chaîne outil

Si la variable système $NT_CLOSE_CHAIN_T n'est pas vide, la chaîne outil ne sera pas fermée à la fin de la chaîne, mais plutôt à la fin de l'élément de chaîne spécifié. D'autres éléments de chaîne qui se trouvent après ce point provoquent un décalage d'origine correspondant lorsque la transformation est activée.

Indice d'un axe d'orientation

Outre les offsets constants entre les axes d'orientation, la fonction CORRTRAFO permet également d'écrire les vecteurs directionnels des axes d'orientation. L'indice d'un axe d'orientation désigne dans ce cas l'indice résultant si l'on parcourt la chaîne cinématique partielle depuis l'origine jusqu'à la fin, en commençant le comptage à zéro. L'indice d'un axe d'orientation est alors toujours égal à l'indice de la section précédente.

L'indice d'un axe d'orientation peut également être déterminé à l'aide de la variable système $AC_TRAFO_ORIAX_LOC.

Modification maximale admissible d'un élément de chaîne

La modification maximale admissible d'un élément de chaîne peut être limitée à l'aide des deux données de réglage SD41610 $SN_CORR_TRAFO_LIN_MAX pour les vecteurs d'offset et SD41611 $SN_CORR_TRAFO_DIR_MAX pour les vecteurs directionnels des axes d'orientation. SD41610 $SN_CORR_TRAFO_LIN_MAX indique la valeur de modification maximale autorisée de chaque composante de vecteur par rapport à la valeur de référence. SD41611 $SN_CORR_TRAFO_DIR_MAX indique l'angle maximal autorisé de modification de la direction de l'axe du vecteur par rapport à la valeur de référence. La valeur de référence est toujours la valeur correspondante qui est effective lors de l'appel des transformations actives de CORRTRAFO. En d'autres termes, les contenus des données de cinématique, éventuellement modifiées dans la gestion des données après l'activation de la transformation, n'ont aucune incidence sur les effets de la fonction CORRTRAFO.

Corrections d'outils 11

11.1 Mémoire de correcteurs

Structure de la mémoire de correcteurs

Chaque tableau de données peut être appelé à l'aide d'un numéro T et d'un numéro D (sauf "Numéros D absolus") et contient, outre les données géométriques de l'outil, d'autres indications telles que le type d'outil.

Structure des numéros D absolus

La "structure horizontale des numéros D" est utilisée lorsque la gestion d'outils a lieu à l'extérieur de la CN. Dans ce cas, les numéros D, c'est-à-dire les correcteurs d'outil, sont créés sans relation avec les outils.

Dans le programme pièce, vous pouvez continuer à programmer T. Ce numéro T est cependant indépendant du numéro D programmé.

Données de tranchant utilisateur

Des données utilisateur de tranchant peuvent être configurées à l'aide de PM. Veuillez tenir compte des indications du constructeur de machine.

Paramètres d'outil

Remarque

Les différentes valeurs de la mémoire de correcteurs

Les différentes valeurs de la mémoires de correcteurs P1 à P25 peuvent être lues et écrites par le programme grâce à des variables système. Tous les autres paramètres sont réservés.

Les paramètres d'outil $TC_DP6 à $TC_DP8, $TC_DP10 et $TC_DP11 ainsi que $TC_DP15 à $TC_DP17, $TC_DP19 et $TC_DP20 possèdent, en fonction du type d'outil, une autre signification.

Paramètre d'outil numéro (DP)	Signification des variables système	Remarque
$TC_DP1	Type d'outil	Voir liste ci-dessous
$TC_DP2	Position du tranchant	Uniquement outils de tournage
Géométrie	**Correction de longueur**	
$TC_DP3	Longueur 1	Prise en compte selon
$TC_DP4	Longueur 2	Type et niveau
$TC_DP5	Longueur 3	
Géométrie	**Rayon**	

Paramètre d'outil numéro (DP)	Signification des variables système	Remarque
$TC_DP6[1] $TC_DP6[2]	Rayon 1 / Longueur 1 Diamètre d	Fraise/outil indexable/meule Scie à rainurer
$TC_DP7[1] $TC_DP7[2]	Longueur 2 / Rayon de congé de fraise conique Largeur de rainure b rayon de congé	Fraises Scie à rainurer
$TC_DP8[1] $TC_DP8[2]	Rayon d'arrondi 1 pour fraises Projection k	Fraises Scie à rainurer
$TC_DP9[1] [3]	Rayon d'arrondi 2	Réservé
$TC_DP10[1]	Angle 1 face frontale de l'outil	Fraises coniques
$TC_DP11[1]	Angle 2 axe longitudinal de l'outil	Fraises coniques
Usure	**Correction de longueur et de rayon**	
$TC_DP12	Longueur 1	
$TC_DP13	Longueur 2	
$TC_DP14	Longueur 3	
$TC_DP15[1] $TC_DP15[2]	Rayon 1 / Longueur 1 Diamètre d	Fraise/outil indexable/meule Scie à rainurer
$TC_DP16[1] $TC_DP16[3]	Longueur 2 / Rayon de congé de fraise conique Largeur de rainure b rayon de congé	Fraises Scie à rainurer
$TC_DP17[1] $TC_DP17[2]	Rayon d'arrondi 1 pour fraises Projection k	Fraises / fraises avant 3D Scie à rainurer
$TC_DP18[1] [3]	Rayon d'arrondi 2	Réservé
$TC_DP19[1]	Angle 1 face frontale de l'outil	Fraises coniques
$TC_DP20[1]	Angle 2 axe longitudinal de l'outil	Fraises coniques
Cote de base/adaptateur	**Corrections de longueur**	
$TC_DP21	Longueur 1	
$TC_DP22	Longueur 2	
$TC_DP23	Longueur 3	
Technologie		
$TC_DP24	Angle de dépouille	Uniquement outils de tournage
$TC_DP25		Réservé

[1] Egalement valable pour les fraises destinées au fraisage en bout 3D

[2] Pour le type d'outil scie à rainurer

[3] Réservé (n'est pas utilisé sur SINUMERIK 840D sl)

Commentaires

Les grandeurs géométriques (par exemple longueur 1 ou rayon) sont constituées de plusieurs composantes. Ces dernières sont additionnées pour donner une grandeur résultante (par exemple la longueur totale 1, le rayon total) qui est alors prise en compte pour la correction.

La valeur zéro est affectée aux corrections inutiles.

Paramètres d'outil $TC-DP1 à $TC-DP23 avec outils de contour

Remarque

Les paramètres d'outil qui ne figurent pas dans le tableau, par exemple $TC_DP7, ne sont pas exploités ; autrement dit, leur contenu n'est pas significatif.

Paramètre d'outil numéro (DP)	Signification	Découper Dn		Remarque
$TC_DP1	Type d'outil			400 à 599
$TC_DP2	Position du tranchant			
Géométrie	**Correction de longueur**			
$TC_DP3	Longueur 1			
$TC_DP4	Longueur 2			
$TC_DP5	Longueur 3			
Géométrie	**Rayon**			
$TC_DP6	Rayon			
Géométrie	**Angle limite**			
$TC_DP10	Angle limite minimal			
$TC_DP11	Angle limite maximal			
Usure	**Correction de longueur et de rayon**			
$TC_DP12	Usure Longueur 1			
$TC_DP13	Usure Longueur 2			
$TC_DP14	Usure Longueur 3			
$TC_DP15	Usure rayon			
Usure	**Angle limite**			
$TC_DP19	Usure de l'angle limite min.			
$TC_DP20	Usure de l'angle limite max.			
Cote de base/adaptateur	**Corrections de longueur**			
$TC_DP21	Longueur 1			
$TC_DP22	Longueur 2			
$TC_DP23	Longueur 3			

Valeur de base et valeur d'usure

Les grandeurs résultantes sont la somme de la valeur de base et de la valeur d'usure (par exemple $TC_DP6 + $TC_DP15 pour le rayon). Pour la longueur d'outil du premier tranchant, la cote de base ($TC_DP21 – $TC_DP23) est par ailleurs également additionnée. De plus, toutes les autres grandeurs pouvant influencer la longueur d'outil même pour un outil conventionnel agissent sur cette longueur d'outil (adaptateur, organe porte-outil orientable, données de réglage).

Angle limite 1 et 2

Les angles limites 1 ou 2 se rapportent chacun au vecteur du centre du bec de l'outil au point de référence du tranchant et sont comptés en sens anti-horaire.

11.2 Corrections additives

11.2.1 Sélection des corrections additives (DL)

Les corrections additives permettent de programmer des compensations qui dépendent du processus d'usinage. Elles se rapportent aux données géométriques d'un tranchant et font donc partie des données de tranchant d'outil.

L'accès aux corrections additives a lieu avec un numéro DL (DL : location dependent ; corrections en fonction du lieu d'utilisation) et leur saisie s'effectue sur l'interface utilisateur.

Application

Les corrections additives permettent de compenser les erreurs de cotes dues au lieu d'utilisation.

Syntaxe

```
DL=<numéro>
```

Signification

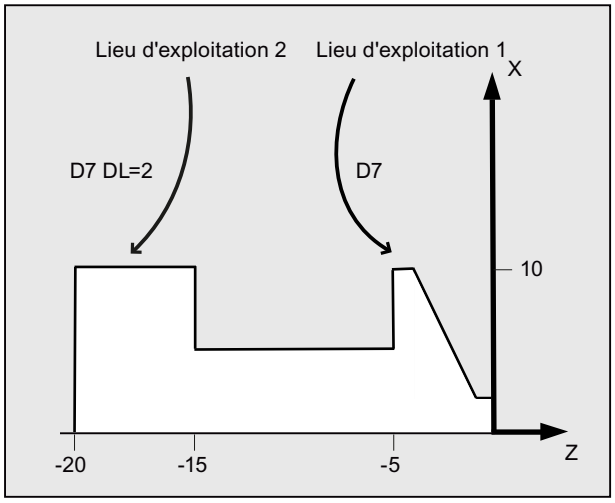
`DL` :	Instruction d'activation d'une correction additive
`<numéro>` :	Le bloc de données de correction additive à activer est indiqué dans le paramètre `<numéro>`.

Remarque

La détermination du nombre de corrections additives et leur activation ont lieu par le biais de paramètres machine (→ voir les indications du constructeur de machine).

Exemple

Le même tranchant est utilisé pour usiner 2 portées de roulement :



Code de programme	Commentaire
N110 T7 D7	; La tourelle revolver est positionnée à l'emplacement 7. D7 et DL=1 sont activés et les corrections correspondantes sont exécutées dans le bloc suivant.
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; DL=2 est activé en supplément de D7 et la correction correspondante est exécutée dans le bloc suivant.
N150 G1 Z-21	
N160 G0 X200 Z200	; Accostage de la position de changement d'outil.
...	

11.2.2 Définition des valeurs d'usure et de réglage ($TC_SCPxy[t,d], $TC_ECPxy[t,d])

La lecture et l'écriture des valeurs d'usure et de réglage s'effectuent à l'aide de variables système. La structure des nouveaux paramètres système est analogue à celle des variables système correspondants pour les outils et les tranchants.

Variables système

$TC_SCPxy[<t>,<d>] :	Valeur d'usure associée au paramètre géométrique correspondant par le biais de xy, x étant le numéro de la valeur d'usure et y le numéro du paramètre géométrique.
$TC_ECPxy[<t>,<d>] :	Valeur de réglage associée au paramètre géométrique correspondant par le biais de xy, x étant le numéro de la valeur de réglage et y le numéro du paramètre géométrique.
<t> : numéro T de l'outil <d> : numéro D du tranchant de l'outil	

Remarque

Les valeurs d'usure et de réglage sont ajoutées aux valeurs des paramètres géométriques et des autres paramètres de correction (numéro D).

Exemple

La valeur d'usure associée à la longueur 1 est fixée à 1.0 pour le tranchant <d> de l'outil <t>.

Paramètre : $TC_DP**3** (longueur 1, pour outils de tournage)

Valeurs d'usure : $TC_SCP1**3** à $TC_SCP6**3**

Valeurs de réglage : $TC_ECP1**3** à $TC_ECP6**3**

$TC_SCP4**3** [<t>,<d>] = 1.0

11.2.3 Effacer les corrections additives (DELDL)

L'instruction `DELDL` permet d'effacer les corrections additives associées à des tranchants d'outil (libération d'espace mémoire). L'effacement concerne les valeurs d'usure aussi bien que les valeurs de réglage.

Syntaxe

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<Etat>=DELDL[<t>,<d>]
```

Signification

`DELDL` :	Instruction de suppression des corrections additives	
`<t>` :	Numéro T de l'outil	
`<d>` :	Numéro D du tranchant de l'outil	
`DELDL[<t>,<d>]` :	Suppression toutes les corrections additives du tranchant `<d>` de l'outil `<t>`.	
`DELDL[<t>]` :	Suppression de toutes les corrections additives de tous les tranchants de l'outil `<t>`.	
`DELDL` :	Suppression de toutes les corrections additives de tous les tranchants de tous les outils de l'unité TO (pour le canal dans lequel l'instruction est programmée).	
`<état>` :	Etat de suppression	
	Valeur :	Signification :
	0	Effacement exécuté correctement.
	-	L'effacement n'a pas eu lieu (lorsque le paramétrage désigne un tranchant précis) ou l'effacement n'a pas été entièrement effectué (lorsque le paramétrage désigne plusieurs tranchants).

Remarque

Les valeurs d'usure et de réglage d'outils actifs ne peuvent pas être effacées (par analogie avec l'effacement de correcteurs `D` et de données d'outil).

11.3 Correcteur d'outil : Interventions spéciales

Les données de réglage SD42900 à SD42960 permettent de gérer l'exploitation du signe pour la longueur d'outil et l'usure.

Ceci concerne également le comportement des composantes d'usure lors de l'application de la fonction miroir à des axes géométriques ou en cas de changement de plan de travail, mais aussi pour la compensation de température dans la direction d'outil.

Valeurs d'usure

Lorsque nous faisons référence à des valeurs d'usure dans ce qui suit, il s'agit de la somme des valeurs d'usure proprement dites ($TC_DP12 à $TC_DP20) et des corrections totales qui comprennent les valeurs d'usure ($SCPX3 à $SCPX11) et les valeurs de réglage ($ECPX3 à $ECPX11).

Pour de plus amples informations sur les corrections totales, voir :
Bibliographie :
Description fonctionnelle Gestion des outils

Données de réglage

SD42900 $SC_MIRROR_TOOL_LENGTH	Fonction miroir appliquée aux composantes de longueur d'outil et de cote de base.
SD42910 $SC_MIRROR_TOOL_WEAR	Fonction miroir appliquée aux valeurs d'usure pour les composantes de longueur d'outil.
SD42920 $SC_WEAR_SIGN_CUTPOS	Exploitation du signe des composantes d'usure en fonction de la position du tranchant.
SD42930 $SC_WEAR_SIGN	Inverse le signe des cotes d'usure.
SD42935 $SC_WEAR_TRANSFORM	Transformation des valeurs d'usure.
SD42940 $SC_TOOL_LENGTH_CONST	Affectation des composantes de longueur d'outil aux axes géométriques.
SD42950 $SC_TOOL_LENGTH_TYPE	Affectation des composantes de longueur d'outil indépendamment du type d'outil-
SD42960 $SC_TOOL_TEMP_COMP	Valeur de compensation de la température dans la direction de l'outil. Est également active pour l'orientation de l'outil actuelle.

Bibliographie

Description fonctionnelle Fonctions de base, Correction d'outil (W1)

Informations complémentaires

Prise d'effet des données de réglage modifiées

En cas de modification des données de réglage décrites, les composantes du correcteur d'outil ne sont actualisées que lors de la prochaine sélection du tranchant d'outil. Dans le cas où un outil est déjà actif et où les données de correction modifiées de cet outil doivent prendre effet, il faut appeler à nouveau cet outil.

Ceci est également valable si la longueur résultante de l'outil est modifiée du fait de l'application de la fonction miroir à un axe. L'outil doit faire l'objet d'une nouvelle sélection après la fonction miroir afin que les composantes modifiées de longueur d'outil prennent effet.

Organes porte-outil orientables et nouvelles données de réglage

Les données de réglage SD42900 à SD42940 n'ont pas d'effet sur les composantes d'un organe porte-outil orientable éventuellement actif. En présence d'un organe porte-outil orientable, l'outil est toujours pris en considération avec sa longueur totale résultante (longueur d'outil + usure + cote de base). Lors du calcul de cette longueur totale résultante, la commande tient compte de toutes les modifications provoquées par des données de réglage c'est-à-dire les vecteurs d'un organe porte-outil orientable sont donc indépendants du plan d'usinage.

Remarque

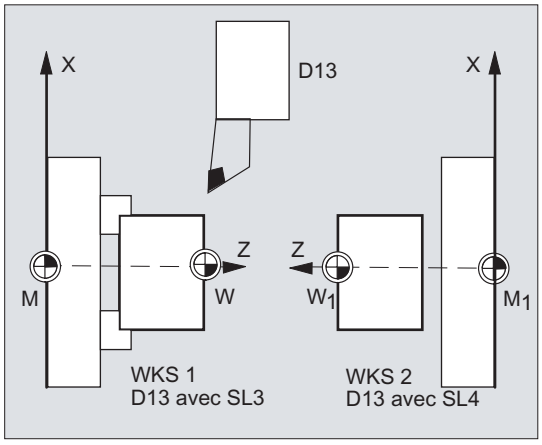
En cas de mise en œuvre d'organes porte-outil orientables, il sera souvent préférable de définir tous les outils pour un système de base non inversé par fonction miroir, même ceux utilisés uniquement pour un usinage symétrique. Pour l'usinage avec des axes inversés, l'organe porte-outil sera pivoté de telle sorte que la position effective de l'outil soit décrite correctement. Toutes les composantes de longueur d'outil agissent alors automatiquement dans le sens correct, de sorte que l'interprétation de certaines composantes par les données de réglage en fonction de l'état d'inversion d'axes est superflue.

Autres possibilités d'applications

L'utilisation de la fonctionnalité "organe porte-outil orientable" peut aussi s'avérer intéressante si la machine ne permet pas d'orienter physiquement des outils, mais dispose d'outils ayant des orientations diverses et installés à demeure. La cotation des outils peut alors être effectuée uniformément avec une orientation de base, les cotes valables pour l'usinage s'obtenant par rotation d'un organe porte-outil virtuel.

11.3.1 Application de la fonction miroir aux longueurs d'outil

Avec des données de réglage SD42900 $SC_MIRROR_TOOL_LENGTH et SD42910 $SC_MIRROR_TOOL_WEAR différentes de zéro, il est possible d'appliquer la fonction miroir aux composantes de longueur d'outil et de cote de base avec les valeurs d'usure par rapport à leurs axes respectifs.



SD42900 $SC_MIRROR_TOOL_LENGTH

Donnée de réglage **différente** de zéro :

Les composantes de longueur d'outil ($TC_DP3, $TC_DP4 et $TC_DP5) et les composantes de cote de base ($TC_DP21, $TC_DP22 et $TC_DP23), dont les axes associés font l'objet de la fonction miroir, sont aussi soumises à la fonction miroir - par inversion de leur signe.

La fonction miroir n'est **pas** appliquée aux valeurs d'usure. Si cela doit être le cas, vous devez régler de façon adéquate la donnée de réglage SD42910 $SC_MIRROR_TOOL_WEAR.

SD42910 $SC_MIRROR_TOOL_WEAR

Donnée de réglage **différente** de zéro :

Les valeurs d'usure pour les composantes de longueur d'outil, dont les axes associés font l'objet de la fonction miroir, sont aussi soumises à la fonction miroir - par inversion de leur signe.

11.3.2 Exploitation du signe de l'usure

Avec les données de réglage SD42920 $SC_WEAR_SIGN_CUTPOS et SD42930 $SC_WEAR_SIGN différentes de zéro, il est possible d'inverser l'exploitation du signe des composantes d'usure.

SD42920 $SC_WEAR_SIGN_CUTPOS

Donnée de réglage **différente** de zéro :

Dans le cas des outils à position de tranchant définie (outils de tournage et de rectification – type d'outil 400), l'exploitation du signe des composantes d'usure dans le plan de travail

dépend de la position du tranchant. Dans le cas des outils sans position de tranchant définie, cette donnée de réglage n'est pas prise en considération.

Dans le tableau suivant, les cotes dont le signe est inversé par le SD42920 (différent de 0) sont repérées par X :

Position du tranchant	Longueur 1	Longueur 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Remarque

Les modifications de signe en fonction des SD42920 et SD42910 sont indépendantes. Si le signe d'une cote est modifié par les deux données de réglage par exemple, il restera inchangé.

SD42930 $SC_WEAR_SIGN

Donnée de réglage **différente** de zéro :

Le signe de toutes les cotes d'usure est inversé. Cette donnée agit aussi bien sur la longueur d'outil que sur les autres dimensions, par exemple le rayon de l'outil, le rayon d'arrondi, etc.

S une cote d'usure positive est entrée, l'outil devient "plus court" et "plus fin". A ce sujet, voir chapitre "Correcteur d'outil : Interventions spéciales, Prise d'effet des données de réglage modifiées".

11.3.3 Système de coordonnées de l'usinage actif (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

En fonction de la cinématique de la machine ou de la présence d'un organe porte-outil orientable, les valeurs d'usure mesurées dans l'un de ces systèmes de coordonnées sont transposées voire transformées dans un système de coordonnées adapté.

Systèmes de coordonnées de l'usinage actif

Des offsets de longueur d'outil peuvent résulter des systèmes de coordonnées suivants et prendre en compte les composantes d'usure de longueur d'outil avec l'instruction G correspondante du groupe 56 dans un outil actif :

- Système de coordonnées machine (SCM)
- Système de coordonnées de base (SCB)
- Système de coordonnées pièce (SCP)

- Système de coordonnées de l'outil (SCO)
- Système de coordonnées de l'outil de transformation cinématique (SCT)

Syntaxe

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

Signification

`TOWSTD` :	Valeur de position de base pour corrections sur la longueur d'outil de la valeur d'usure
`TOWMCS` :	Corrections dans la longueur d'outil dans SCM
`TOWWCS` :	Corrections dans la longueur d'outil dans SCP
`TOWBCS` :	Corrections dans la longueur d'outil dans SCB
`TOWTCS` :	Corrections de la longueur d'outil sur le point de référence de l'organe porte-outil (organe porte-outil orientable)
`TOWKCS` :	Corrections de la longueur d'outil de la tête d'outil (transformation cinétique)

plus d'informations

Différences

Le tableau suivant représente les principales différences :

Instruction G	Valeur d'usure	Organe porte-outil orientable actif
`TOWSTD` :	Valeur de position initiale, longueur d'outil	Les valeurs d'usure sont assujetties à la rotation.
`TOWMCS` :	Valeur d'usure dans SCM. `TOWMCS` est identique à `TOWSTD` lorsque aucun organe porte-outil orientable n'est actif.	Seul le vecteur de la longueur d'outil résultante tourne, sans prise en compte de l'usure.
`TOWWCS` :	La valeur d'usure est convertie du SCP au SCM.	Le vecteur d'outil est calculé sans prise en compte de l'usure comme pour `TOWMCS`.
`TOWBCS` :	La valeur d'usure est convertie dans SCB au SCM.	Le vecteur d'outil est calculé sans prise en compte de l'usure comme pour `TOWMCS`.
`TOWTCS` :	La valeur d'usure est convertie dans le système de coordonnées de l'outil au SCM.	Le vecteur d'outil est calculé sans prise en compte de l'usure comme pour `TOWMCS`.

`TOWWCS`, `TOWBCS`, `TOWTCS` : Le vecteur d'usure est additionné au vecteur d'outil.

Transformation linéaire

La longueur d'outil est uniquement judicieusement définissable dans SCM lorsque le SCM est issu de SCB par une transformation linéaire.

Transformation non linéaire

Si par exemple une transformation non linéaire est active avec TRANSMIT, alors le SCB est automatiquement utilisé lorsque le SCM est indiqué comme système de coordonnées souhaité.

Pas de transformation cinématique et pas d'organe porte-outil orientable

Lorsque ni une transformation cinématique ni un organe porte-outil orientable n'est actif, alors, mis à part le SCP, les quatre autres systèmes de coordonnées coïncident. Le SCP se différencie alors de tous les autres. Du fait que seules les longueurs d'outil doivent être interprétées, les translations entre les systèmes de coordonnées n'ont plus de signification.

Bibliographie :
Plus d'informations sur la correction d'outil, voir :
Description fonctionnelle Fonctions de base, Correction d'outil (W1)

Prise en compte des valeurs d'usure

La donnée de réglage **SD42935 $SC_WEAR_TRANSFORM** détermine quelle composante d'usure parmi les trois composantes :

- Usure
- Corrections totales fines
- Corrections totales grossières

d'une rotation doit être assujettie par une transformation d'adaptateur ou par un organe porte-outil orientable lorsque l'une des instructions G suivantes est active :

- `TOWSTD`
  Position d'initialisation : Pour corrections sur la longueur d'outil
- `TOWMCS`
  Valeurs d'usure dans le système de coordonnées machine (SCM).
- `TOWWCS`
  Valeurs d'usure dans le système de coordonnées pièce (SCP)
- `TOWBCS`
  Valeurs d'usure dans le système de coordonnées de base (SCB).

- TOWTCS
  Valeurs d'usure dans le système de coordonnées d'outil sur le mandrin du porte-outil (référence de l'organe porte-outil T)
- TOWKCS
  Valeurs d'usure dans le système de coordonnées de la tête d'outil avec transformation cinétique.

Remarque

L'évaluation de chaque composante d'usure (affectation aux axes géométriques, exploitation du signe) est influencée par les facteurs suivants :

- Niveau actif
- Transformation d'adaptateur
- Données de réglage :
  - SD42910 $SC_MIRROR_TOOL_WEAR
  - SD42920 $SC_WEAR_SIGN_CUTPOS
  - SD42930 $SC_WEAR_SIGN
  - SD42940 $SC_TOOL_LENGTH_CONST
  - SD42950 $SC_TOOL_LENGTH_TYPE

11.3.4 Longueurs d'outil et changement de plan

Avec les données de réglage SD42940 $SC_TOOL_LENGTH_CONST différentes de zéro, il est possible d'affecter les composantes de longueur d'outil telles que la longueur, l'usure et la cote de base aux axes géométriques pour les outils de tournage et de rectification lors d'un changement de plan.

SD42940 $SC_TOOL_LENGTH_CONST

Donnée de réglage **différente** de zéro :

L'affectation des composantes de longueur d'outil (longueur, usure et cote de base) aux axes géométriques n'est pas modifiée en cas de changement de plan de travail (G17 à G19).

Le tableau suivant indique l'affectation des composantes de longueur d'outil aux axes géométriques pour des outils de tournage et de rectification (types 400 à 599) :

Contenu	Longueur 1	Longueur 2	Longueur 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Toute valeur différente de 0, qui n'est pas égale à une des six valeurs mentionnées, est interprétée comme la valeur 18.

Le tableau suivant indique l'affectation des composantes de longueur d'outil aux axes géométriques pour tous les autres outils (types < 400 ou > 599) :

Plan de travail	Longueur 1	Longueur 2	Longueur 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Toute valeur différente de 0, qui n'est pas égale à une des six valeurs mentionnées, est interprétée comme la valeur 17.

Remarque

Dans ces tableaux, on suppose que les axes géométriques 1 à 3 sont désignés par X, Y, Z. Pour l'affectation d'une correction à un axe, ce n'est pas le descripteur d'axe mais l'ordre des axes qui est déterminant.

11.4 Correction d'outil en ligne

11.4.1 Définir une fonction polynôme (FCTDEF)

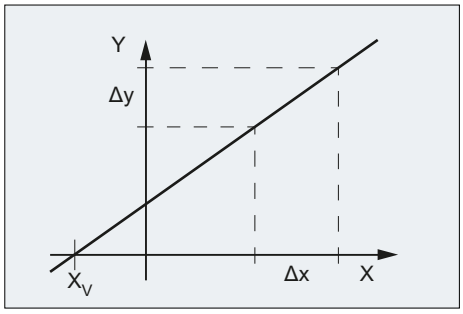
Certaines stratégies de dressage (par ex. molette de dressage) se distinguent par le fait que la meule décroît en continu (de manière linéaire) au niveau du rayon pendant la pénétration de la molette de dressage. A cet effet, une fonction linéaire est nécessaire entre la pénétration de la molette de dressage et l'écriture de la valeur d'usure de la longueur correspondante. La définition de la fonction linéaire s'effectue via la procédure FCTDEF(...) prédéfinie pour les fonctions polynôme jusqu'au 3ème degré maximum.

Equation de droite

$y = f(x) = a_0 + a_1 * x_1$

a_1 : Pente de la droite, avec $a_1 = \Delta x / \Delta y$

a_0 : Décalage de la droite le long de l'axe X, avec $a_0 = -a1 * X_V$



Syntaxe

```
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
```

Signification

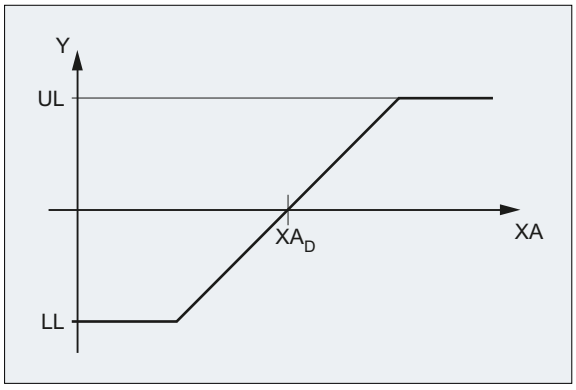
`FCTDEF(...)` :	Définition d'une fonction polynôme pour PUTFTOCF(...) : $y = f(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3$	
<`Func`**>** :	numéro de fonction	
	Type de données :	INT
	Plage de valeurs :	1, 2, 3
`<LLimit>` :	Valeur limite inférieure	
	Type de données :	REAL
`<ULimit>` :	Valeur limite supérieure	
	Type de données :	REAL
`<a0>,<a1>,<a2>,<a3>` :	Coefficients de la fonction polynomiale	
	Type de données :	REAL

Exemple

Règles

- Numéro de fonction : 1
- Valeurs limites inférieure et supérieure : -100, 100
- Pente de la caractéristique : $a_1 = 1$
- Le point de fonctionnement doit se situer au milieu de la caractéristique. La caractéristique doit pour cela être décalée dans le sens Y négatif à partir de la position de consigne de l'axe XA dans le SCP au moment de la définition de la fonction dans le programme CN. $a_0 = -a_1 * XA_D = -1 * \$AA\_IW$
- $a_2 = a_3 = 0$

Caractéristique



UL Valeur limite supérieure

LL Valeur limite inférieure

XA_D Consigne de l'axe XA au moment de la définition de la fonction dans le programme CN

Programmation

Code de programme	Commentaire
`FCTDEF(1,-100,100,-$AA_IW[XA],1)`	`; Définition de la fonction`

11.4.2 Ecriture en continu de la correction d'outil en ligne (PUTFTOCF)

La procédure PUTFTOCF(...) prédéfinie permet d'effectuer une correction d'outil en ligne via une fonction polynomiale définie au préalable avec FCTDEF(...) (Page 442).

Remarque

La correction d'outil en ligne peut aussi être réalisée par le biais d'une action synchrone.

Pour plus d'informations, voir Description fonctionnelle Actions synchrones.

Syntaxe

```
PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)
```

Signification

`PUTFTOCF(...)` :	Ecriture continue modale de la correction d'outil en ligne à l'aide de la fonction polynomiale définie avec FCTDEF(...)	
`<Func>` :	Numéro de fonction, spécifié par la définition de fonction avec FCTDEF(...)	
	Type de données :	INT
	Plage de valeurs :	1, 2, 3
`<RefVal>` :	Valeur de référence à partir de laquelle la correction doit être déduite (p. ex. consigne d'un axe)	
	Type de données :	VAR REAL
`<ToolPar>` :	Numéro du paramètre d'usure (longueur 1, 2 ou 3) dans lequel la valeur de correction doit être prise en compte	
	Type de données :	INT
`<Chan>` :	Numéro du canal dans lequel la correction d'outil en ligne doit être effective **Remarque** : Uniquement requis lorsque la correction ne doit pas être effective dans le canal actif.	
	Type de données :	INT
`<Sp>` :	Numéro de la broche pour laquelle la correction d'outil en ligne doit être effective **Remarque :** Uniquement requis lorsqu'il s'agit de corriger une meule non active à la place de l'outil actif mis en œuvre.	
	Type de données :	INT

11.4.3 Ecriture discrète de la correction d'outil en ligne (PUTFTOC)

Fonction

La procédure PUTFTOC(...) prédéfinie permet d'effectuer une correction d'outil en ligne avec une valeur de correction fixe.

Syntaxe

```
PUTFTOC(<CorrVal>,<ToolPar>,<Chan>,<Sp>)
```

Signification

`PUTFTOC(...)` :	Ecriture de la correction d'outil en ligne	
`<CorrVal>` :	Valeur de correction ajoutée au paramètre d'usure	
	Type de données :	REAL
`<ToolPar>` :	Numéro du paramètre d'usure (longueur 1, 2 ou 3) dans lequel la valeur de correction doit être prise en compte	
	Type de données :	INT
`<Chan>` :	Numéro du canal dans lequel la correction d'outil en ligne doit être effective **Remarque :** Uniquement requis lorsque la correction ne doit pas être effective dans le canal actif.	
	Type de données :	INT
`<Sp>` :	Numéro de la broche pour laquelle la correction d'outil en ligne doit être effective **Remarque :** Uniquement requis lorsqu'il s'agit de corriger une meule non active à la place de l'outil actif mis en œuvre.	
	Type de données :	INT

11.4.4 Activer/désactiver la correction d'outil en ligne (FTOCON/FTOCOF)

Les instructions G FTOCON et FTOCOF permettent d'activer ou de désactiver la correction d'outil en ligne.

Syntaxe

```
FTOCON
...
FTOCOF
```

Signification

FTOCON :	Activer la correction d'outil en ligne L'instruction doit être programmée dans le canal dans lequel la correction d'outil en ligne doit être activée.
FTOCOF :	Désactiver la correction d'outil en ligne L'instruction doit être programmée dans le canal dans lequel la correction d'outil doit être désactivée. **Remarque :** Avec FTOCOF, la correction d'outil n'est plus appliquée. Cependant, la valeur prise en considération avec PUTFTOC/PUTFTOCF est conservée dans les données de correction spécifiques au tranchant. Pour la désactivation définitive de la correction d'outil en ligne, FTOCOF doit être suivie d'une sélection/désélection de l'outil (T...).

11.5 Correction du rayon d'outil 3D

11.5.1 Sélection de la correction 3D du rayon d'outil pour le fraisage périphérique 3D (CUT3DC, CUT3DCD, ISD)

La correction 3D du rayon d'outil (CRO 3D) pour le fraisage périphérique 3D, sans prise en compte de surfaces de délimitation, est sélectionnée avec l'instruction G à effet modal CUT3DC ou CUT3DCD.

L'activation proprement dite s'effectue avec G41 ou G42. G40 désactive la correction du rayon d'outil.

Syntaxe

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

Signification

CUT3DC :	CRO 3D pour le fraisage périphérique (uniquement si la transformation 5 axes est activée).	
CUT3DCD :	CRO 3D rapportée à un outil différentiel pour le fraisage périphérique (uniquement si la transformation 5 axes est activée). La différence de rayon est spécifiée par le paramètre outil $TC_DP15.	
G41/G42 X... Y... Z... :	Activer la correction de rayon d'outil	
	G41 :	Correction du rayon d'outil à gauche du contour
	G42 :	Correction du rayon d'outil à droite du contour
	Remarque : L'activation doit intervenir dans un bloc linéaire (G0/G1).	
CDOF2 :	Désactiver la détection des violations du contour pour le fraisage périphérique 3D	
ORIC/ORID :	Les instructions G ORIC et ORID déterminent le comportement lors de changements d'orientation aux angles saillants.	
	ORIC :	Les changements d'orientation aux angles saillants sont superposés au bloc circulaire à insérer.
	ORID :	Les changements d'orientation aux angles saillants sont exécutés avant le bloc circulaire devant être inséré.

`ISD=<Value>:`	Lors du fraisage périphérique avec la correction du rayon d'outil 3D **active**, l'adresse `ISD` (InSertion Depth) permet de modifier la profondeur de pénétration de l'outil.
	`<Value>:` Longueur de la profondeur de pénétration
`G40 X... Y... Z...:`	Désactiver la correction de rayon d'outil **Remarque :** La désactivation doit intervenir dans un bloc linéaire (G0/G1) avec déplacements d'axes géométriques.

Remarque

Les instructions G pour la sélection de CRO 3D sont évaluées dans le bloc d'accostage, donc typiquement dans le bloc contenant G41 ou G42.

G41 ou G42 peut également être programmée dans des blocs sans déplacement pour les axes géométriques concernés par la correction. Dans ce cas, le bloc d'accostage est le premier bloc de déplacement après ce type de bloc.

Tout changement de la variante CRO 3D, alors que la correction du rayon d'outil est active, est ignoré sans alarme.

Exemple

```
Code de programme                    Commentaire
                                     ; Définition de l'outil D1 :
$TC_DP1[1,1]=120                     ; Type (fraise à queue)
$TC_DP3[1,1]=20                      ; Vecteur de correction de longueur
$TC_DP6[1,1]=8                       ; Rayon

N10 X0 Y0 Z0 T1 D1 F12000            ; Sélection de l'outil.
N20 TRAORI(1)                        ; Activation de la transformation.
N30 G42 ORIC ISD=10 CUT3DC G64 X30   ; Activation du fraisage périphérique 3D,
                                     ; changement d'orientation en continu au niveau des angles saillants,
                                     ; profondeur de pénétration : 10 mm
N40 ORIWKS A30 B15                   ; Changement d'orientation au niveau d'un angle en spécifiant les positions d'axe.
N50 Y20 A3=1 C3=1                    ; Bloc de déplacement avec changement d'orientation,
                                     ; spécification de l'orientation avec vecteur directionnel.
N60 X50 Y30                          ; Bloc de déplacement avec orientation constante.
N70 Y50 A3=0.5 B3=1 C3=5             ; Bloc de déplacement avec changement d'orientation.
N80 M63                              ; Bloc sans information de déplacement.
N90 X0 ISD=20                        ; Bloc de déplacement avec changement de la profondeur de pénétration.
N100 G40 Y0                          ; Désactivation de la correction du rayon d'outil.
N110 M30
```

plus d'informations

Trajectoire et orientation

La variante de fraisage périphérique utilisée ici est réalisée par spécification d'un contour (ligne directrice) et de l'orientation correspondante. Dans ce type d'usinage, la forme de l'outil n'a aucune importance pour la trajectoire. Seul le rayon est déterminant au point d'attaque de l'outil.

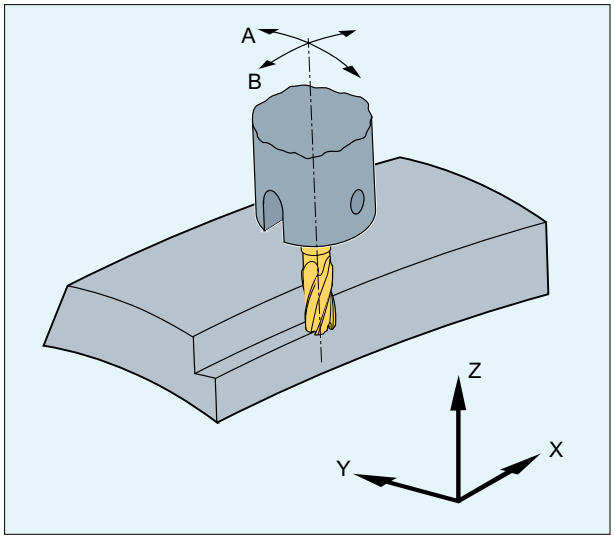


Figure 11-1 Fraisage périphérique

Comportement à l'accostage

Le comportement à l'accostage est toujours NORM pour les variantes 3D de la correction du rayon d'outil.

Comportement aux angles saillants

Lors du fraisage périphérique avec CRO 3D au niveau des angles saillants, les instructions G du groupe 18 (correction d'outil pour comportement aux angles) sont évaluées de manière analogique aux conditions de la CRO 2½D.

- G450 : Arc de raccordement (l'outil contourne les angles de la pièce sur une trajectoire circulaire)
  Contrairement aux solutions de la CRO 2½D, l'élément de contour inséré au niveau d'un angle saillant est toujours un cercle de rayon 0 sur lequel la correction du rayon d'outil agit de la même manière que sur toute autre trajectoire programmée. Il n'est pas possible d'insérer des coniques à la place des cercles. Dans ce cas, l'adresse DISC est sans signification et n'est pas évaluée.
- G451 : Point d'intersection des équidistantes (l'outil s'arrête aux angles de la pièce)
  Le point d'intersection est déterminé en prolongeant les trajectoires décalées des deux blocs concernés dans le plan perpendiculaire à la direction de l'outil au niveau de l'angle.

La méthode du point d'intersection (G451) n'est pas utilisée lorsqu'au moins un bloc contenant un changement de l'orientation de l'outil est inséré entre les blocs de déplacement concernés. Dans un tel cas, c'est toujours un cercle qui est inséré au niveau de l'angle.

Comportement lors de changements d'orientation au niveau des angles saillants

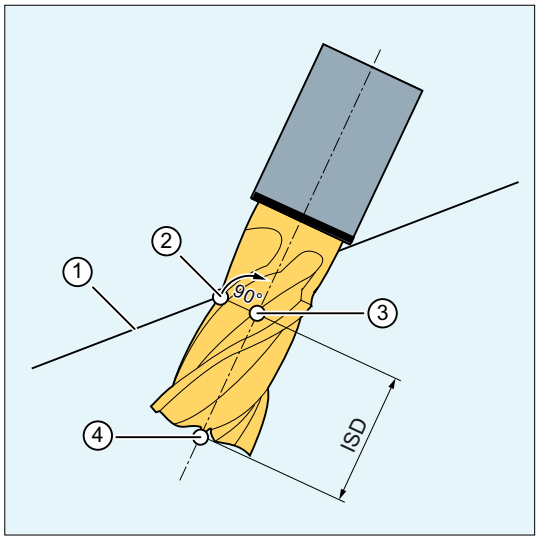
Les instructions G ORIC et ORID permettent de définir si les changements d'orientation qui ont été programmés entre les deux blocs de déplacement formant l'angle seront effectués avant le début du bloc à interpolation circulaire inséré (ORID) ou en même temps que celui-ci (ORIC).

Profondeur de pénétration

La profondeur de pénétration de la fraise correspond à la distance entre le point auxiliaire de la fraise et la pointe de l'outil.

Le point auxiliaire de la fraise est la projection orthogonale du point d'usinage de la fraise sur la trajectoire programmée sur l'axe longitudinal de l'outil.

Ainsi, la profondeur de pénétration règle la position du point d'usinage sur la surface latérale de l'outil.



① Trajectoire programmée
② Point d'usinage de la fraise
③ Point auxiliaire de la fraise
④ Pointe de la fraise
ISD Profondeur de pénétration (InSertion Depth)

Figure 11-2 Profondeur de pénétration

Correction du rayon d'outil par rapport à un outil différentiel

La CRO 3D rapportée à un outil différentiel pour le fraisage périphérique est activée au moyen de l'instruction CUT3DCD. Elle doit être utilisée lorsque le contour programmé se rapporte à la trajectoire du centre d'un outil standard et que l'usinage s'effectue avec un outil différent. Ainsi, lors du calcul de la correction du rayon d'outil 3D, seuls la valeur d'usure du rayon de l'outil actif ($TC_DP15) et les offsets de correction d'outil OFFN et TOFFR éventuellement programmés sont pris en compte. Le rayon de base ($TC_DP6) de l'outil actif n'est **pas** pris en compte.

Fraisage de poche avec faces latérales obliques pour fraisage périphérique avec CUT3DC

Dans le cas de cette correction de rayon d'outil 3D, un écart du rayon de fraise est compensé par un déplacement perpendiculaire à la surface à usiner. Le plan dans lequel se trouve la surface d'attaque de la fraise reste inchangé lorsque la profondeur de pénétration ISD reste inchangée. Une fraise dotée par exemple d'un rayon plus petit par rapport à un outil standard n'atteindrait pas le fond de la poche, qui représente également la surface de délimitation. Pour permettre une pénétration automatique de l'outil, la commande doit connaître cette surface de délimitation, voir chapitre "Fraisage périphérique 3D avec prise en compte d'une surface de délimitation (CUT3DCC, CUT3DCCD) (Page 457)".

Advanced Surface / Top Surface

Remarque

Pour l'utilisation de la correction du rayon d'outil CUT3DCD en association avec l'option soumise à la licence "Advanced Surface" ou "Top Surface", tenir compte des recommandations de réglages relatives à "Advanced Surface" / "Top Surface".

Pour vérifier les paramètres réglés, utiliser les programmes de vérification spéciaux, disponibles sur le portail SIOS.

- Programmes de vérification pour Advanced Surface (https://support.industry.siemens.com/cs/ww/fr/view/78956392/en)
- Programmes de vérification pour Top Surface (https://support.industry.siemens.com/cs/ww/fr/view/109738423/en)

11.5.2 Sélection de la correction 3D du rayon d'outil pour le fraisage en bout 3D (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

La correction 3D du rayon d'outil (CRO 3D) pour le fraisage en bout 3D est sélectionnée avec l'instruction G à effet modal CUT3DF, CUT3DFS, CUT3DFF ou CUT3DFD.

L'activation proprement dite s'effectue avec G41 ou G42.

Pour pouvoir calculer la correction du rayon d'outil en mode fraisage en bout 3D, la définition du vecteur normal à la surface du plan à usiner est nécessaire. Cette définition doit être effectuée avec G41 ou G42 et les adresses A4, B4, C4 et A5, B5, C5.

G40 désactive la correction du rayon d'outil.

Syntaxe

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

Signification

`CUT3DFS`:	CRO 3D pour le fraisage en bout avec orientation d'outil constante. L'orientation de l'outil est définie par G17 - G19 et n'est pas influencée par les frames.	
`CUT3DFF`:	CRO 3D pour le fraisage en bout avec orientation d'outil constante. L'orientation de l'outil est la direction définie par G17 - G19 et, le cas échéant, par la composante rotationnelle d'un frame.	
`CUT3DF`:	CRO 3D pour le fraisage en bout avec modification de l'orientation d'outil (uniquement si la transformation 5 axes est activée).	
`CUT3DFD`:	La CRO 3D rapportée à un outil différentiel pour le fraisage en bout avec modification de l'orientation d'outil (uniquement si la transformation 5 axes est activée). La différence de rayon est spécifiée par le paramètre outil $TC_DP15. **Remarque** : CUT3DFD n'est possible qu'en association avec le "lissage des normales à la surface en fraisage en bout 3D". Celui-ci est activé par l'appel de la fonction "Top Surface" (soumise à licence) avec CYCLE832(...) (Page 822).	
`G41/G42 X... Y... Z...`:	Activer la correction d'outil Le comportement pour G41 et pour G42 est identique en mode fraisage en bout 3D. **Remarque :** L'activation doit intervenir dans un bloc linéaire (G0/G1).	
`A4/5=... B4/5=... C4/5=...`:	Définition du vecteur normal à la surface du plan à usiner	
	`A4=... B4=... C4=...`:	Définition en début de bloc
	`A5=... B5=... C5=...`:	Définition en fin de bloc
`ORIC/ORID`:	Les instructions G `ORIC` et `ORID` déterminent le comportement lors de changements d'orientation aux angles saillants.	
	`ORIC`:	Les changements d'orientation aux angles saillants sont superposés au bloc circulaire à insérer.
	`ORID`:	Les changements d'orientation se font avant l'exécution du bloc circulaire inséré.
`G40 X... Y... Z...`:	Désactiver la correction de rayon d'outil **Remarque :** La désactivation doit intervenir dans un bloc linéaire (G0/G1) avec déplacements d'axes géométriques.	

Remarque

G41 ou G42 peut également être programmée dans des blocs sans déplacement pour les axes géométriques concernés par la correction. Dans ce cas, le bloc d'accostage est le premier bloc de déplacement après ce type de bloc.

Tout changement de la variante CRO 3D, alors que la correction du rayon d'outil est active, est ignoré sans alarme.

Exemples

Exemple 1 : Fraisage en bout 3D avec CUT3DF

Code de programme	**Commentaire**
N10	; Définition de l'outil D1 :
N20 $TC_DP1[1,1]=121	; Type d'outil (fraise convexe demi-cercle)
N30 $TC_DP3[1,1]=20	; Correction de longueur
N40 $TC_DP6[1,1]=5	; Rayon
N50 $TC_DP7[1,1]=3	; Rayon d'arrondi
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; Sélection de l'outil.
N90 TRAORI(1)	; Sélectionner la transformation d'orientation.
N100 B4=-1 C4=1	; Définition du plan.
N110 **G41** ORID **CUT3DF** G64 X10 Y0 Z0	; Activer la correction d'outil.
N120 X30	
N130 Y20 A4=1 C4=1	; Angle saillant, nouvelle définition du plan.
N140 B3=1 C3=5	; Changement d'orientation avec ORID.
N150 B3=1 C3=1	; Changement d'orientation avec ORID.
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; Changement d'orientation avec ORIC
N180 A3=-1 C3=1	; Changement d'orientation avec ORIC
N190 Y-10 A4=-1 C4=3	; Nouvelle définition du plan.
N200 X-20 Y-20 Z10	; Angle rentrant avec le bloc précédent.
N210 X-30 Y10 A4=1 C4=1	; Angle rentrant, nouvelle définition du plan.
N220 A3=1 B3=0.5 C3=1.7	; Changement d'orientation avec ORIC.
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; Changement d'orientation.
N250 X0 Y30 C4=1	; Déplacement avec interpolation, nouveau plan,
	; orientation avec programmation relative.
N260 BSPLINE X20 Z15	; Début de la courbe de type spline, programmation relative de l'orientation
N270 X30 Y25 Z18	; Reste active pendant la fonction spline.
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	

Code de programme	Commentaire
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; Hélice, orientation avec programmation d'axe.
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; Déplacement avec interpolation, orientation, plan non constant.
N330 X-20 G40	; Désactivation de la correction du rayon d'outil
N340 M30	

Exemple 2 : Programme CN (extrait) avec CUT3DFD, généré depuis un système CAO

Code de programme	Commentaire
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
; Accostage de la position de départ dans le système de coordonnées machine en fonction de la longueur d'outil.	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; Fin du positionnement dans le système de coordonnées machine.	
;	
TRAORI(1)	; Sélectionner la transformation d'orientation.
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; Appel de CYCLE832 avec : ; tolérance de contour = 0,01 mm, ; type d'usinage : Top Surface avec lissage, ; finition avec saisie d'une tolérance d'orientation, ; tolérance d'orientation = 1 degré
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; Les blocs N09 à N10 constituent la part rapide du mouvement d'accostage de la pièce à orientation constante.
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; La part lente du mouvement d'accostage est réalisée dans les blocs N11 à N21. L'outil est déjà près de la pièce et les réglages d'usinage de moule (par exemple COMPSURF) de CYCLE832 sont alors actifs (en raison de G01 actif). La longueur de trajectoire de cette phase transitoire du comportement d'usinage de moule doit être d'environ 1000 fois la tolérance de contour (10 mm dans cet exemple).
N12 X-247.69126 Y33.82182 Z24.78219 F1061.00000	

Code de programme	**Commentaire**
N13 X-247.76560 Y33.13299 Z21.73022	
N14 X-247.82755 Y32.55897 Z19.18691	
N15 X-247.87918 Y32.08062 Z17.06748	
N16 X-247.92220 Y31.68200 Z15.30129	
N17 X-247.95805 Y31.34981 Z13.82947 A3=-.216361686 B3=.934284391 C3=-.283373071	
N18 X-247.98792 Y31.07299 Z12.60295 A3=-.216360662 B3=.934280801 C3=-.283385691	
N19 X-248.01282 Y30.84230 Z11.58085 A3=-.216336015 B3=.934194446 C3=-.283689030	
N20 X-248.03357 Y30.65006 Z10.72910 A3=-.216233089 B3=.933833626 C3=-.284952647	
N21 X-248.05086 Y30.48986 Z10.01931 A5=-.060687572 B5=.974940255 C5=-.214029243 A3=-.215712821 B3=.932005189 C3=-.291263295	
N22 **G41** X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	; À partir de N22, la normale à la surface est définie pour la première fois sur l'ensemble du bloc (autrement dit, la normale à la surface à la fin du bloc précédent N21 est disponible et, de ce fait, la normale à la surface au début du bloc N22 et la normale à la surface à la fin du bloc N22). La condition d'activation de la correction d'outil avec G41/G42 est donc remplie.
N23 X-248.07130 Y30.15119 Z8.71082 A5=-.060165696 B5=.971048883 C5=-.231179920 A3=-.214177198 B3=.926684940 C3=-.308841625	
N24 X-248.07829 Y29.97126 Z8.05094 A5=-.059884286 B5=.968941717 C5=-.239928784 A3=-.213318480 B3=.923853466 C3=-.317789237	
N25 X-248.08317 Y29.78487 Z7.38844 A5=-.059584206 B5=.966718449 C5=-.248807482 A3=-.212397895 B3=.920898045 C3=-.326854594	
N26 X-248.08578 Y29.59254 Z6.72679 A5=-.059263963 B5=.964380907 C5=-.257793037 A3=-.211418355 B3=.917822366 C3=-.336012474	
...	

Remarque

Pour le fraisage en bout 3D avec CUT3DFD, la définition des normales à la surface est nécessaire pour l'activation de la correction d'outil avec G41/G42. La programmation de G41/G42 sans définition des normales à la surface provoque une alarme.

plus d'informations

Fraisage en bout 3D

Ce type de fraisage 3D nécessite la description des différentes trajectoires 3D qui balayent la surface de la pièce, ligne après ligne. Les calculs sont généralement réalisés à l'aide d'un

système de FAO, en tenant compte de la forme et des dimensions de l'outil. Dans le programme pièce, le postprocesseur écrit non seulement les blocs CN, mais également les orientations d'outil (lorsque la transformation 5 axes est active) et l'instruction G de la correction d'outil 3D souhaitée. Ainsi l'opérateur de la machine a la possibilité d'utiliser des outils légèrement plus petits par rapport à l'outil utilisé pour le calcul des trajectoires CN.

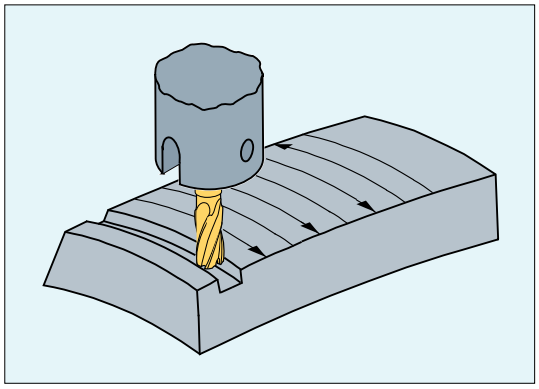


Figure 11-3 Fraisage en bout

Comportement à l'accostage

Le comportement à l'accostage est toujours NORM pour les variantes 3D de la correction du rayon d'outil.

Comportement aux angles saillants

En mode fraisage en bout, les angles saillants sont traités en tant que cercles de rayon 0, sachant que le plan du cercle est déterminé par la tangente au point final du premier bloc et la tangente du point de départ du deuxième bloc. Ainsi un changement d'orientation est possible à la transition de blocs. Par conséquent, un cercle est toujours inséré en tant qu'élément de contour au niveau d'un angle saillant. La méthode du point d'intersection n'est pas disponible en mode fraisage en bout.

Comportement lors de changements d'orientation au niveau des angles saillants

Les instructions G ORIC et ORID permettent de définir si les changements d'orientation qui ont été programmés entre les deux blocs de déplacement formant l'angle seront effectués avant le début du bloc à interpolation circulaire inséré (ORID) ou en même temps que celui-ci (ORIC).

Correction du rayon d'outil par rapport à un outil différentiel

La correction du rayon d'outil 3D rapportée à un outil différentiel est activée avec l'instruction CUT3DFD. Elle doit être utilisée lorsque le contour programmé se rapporte à la trajectoire du centre d'un outil standard et que l'usinage s'effectue avec un outil différent. Ainsi, lors du calcul de la correction du rayon d'outil 3D, seuls la valeur d'usure du rayon de l'outil actif ($TC_DP15) et les offsets de correction d'outil OFFN et TOFFR éventuellement programmés sont pris en compte. Le rayon de base ($TC_DP6) de l'outil actif n'est **pas** pris en compte.

Le fraisage en bout 3D avec CUT3DFD n'est possible qu'en association avec le "lissage de la normale à la surface en fraisage en bout 3D". Celui-ci est activé par l'appel de la fonction "Top Surface" (soumise à licence) avec CYCLE832(...) (Page 822). L'activation doit avoir lieu **avant** l'activation de la correction d'outil avec G41/G42, et cela non pas juste avant l'attaque de l'outil, mais une longueur de trajectoire plut tôt, cette longueur étant d'environ 1000 fois la

tolérance de contour (par exemple 1000 x 0,01 mm = 10 mm). Pour la désactivation, il faut suivre la procédure inverse : d'abord désactiver la correction d'outil avec G40, puis, après une longueur de trajectoire correspondant environ à 1000 fois la tolérance de contour, procéder à la désactivation avec p. ex. CUT2D (ou autre).

Pour pouvoir appliquer le "lissage des normales à la surface en fraisage en bout 3D", le déblocage de la fonction "Interpolation des normales à la surface par polynômes" est nécessaire :

MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

Remarque

Pour le fraisage en bout 3D avec CUT3DFD en association avec "Top Surface", tenir compte des recommandations de réglages relatives à "Top Surface".

Pour vérifier les paramètres réglés, utiliser les programmes de vérification spéciaux, disponibles sur le portail SIOS (https://support.industry.siemens.com/cs/ww/fr/view/109738423/en).

11.5.3 Fraisage périphérique 3D avec prise en compte d'une surface de délimitation (CUT3DCC, CUT3DCCD)

Dans le cas du fraisage périphérique 3D avec modification continue ou constante de l'orientation de l'outil, la trajectoire du centre de l'outil est souvent programmée pour un outil standard défini. Compte tenu que dans bien des cas réels, on ne dispose pas des outils standard adéquats, on peut utiliser un outil ne différant pas trop sensiblement (≤ 5 %) d'un outil standard.

Avec CUT3DCCD, une surface de délimitation, qui décrirait l'outil standard programmé, est prise en compte pour un outil différentiel réel. Le programme CN décrit la trajectoire centrale de l'outil standard.

Avec CUT3DCC, une surface de délimitation, qui aurait atteint l'outil standard programmé, est prise en compte dans le cas de l'utilisation d'outils cylindriques. Le programme CN décrit le contour de la surface d'usinage.

Le vecteur normal à la surface de délimitation est spécifié comme en mode fraisage en bout 3D par A4, B4, C4 et A5, B5, C5.

Syntaxe

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

Signification

`CUT3DCCD:`	CRO 3D pour le fraisage périphérique en tenant compte d'une surface de délimitation avec outil différentiel sur la trajectoire du centre de l'outil : déplacement perpendiculaire jusqu'à la surface de délimitation	
`CUT3DCC:`	CRO 3D pour le fraisage périphérique en tenant compte d'une surface de délimitation avec correction de rayon 3D : contour de la surface d'usinage	
`G41/G42 X... Y... Z...:`	Activer la correction de rayon d'outil	
	`G41:`	Correction du rayon d'outil à gauche du contour
	`G42:`	Correction du rayon d'outil à droite du contour
	Remarque : L'activation doit intervenir dans un bloc linéaire (G0/G1).	
`CDOF2:`	Désactiver la détection des violations du contour pour le fraisage périphérique 3D	
`A4/5=... B4/5=... C4/5=...:`	Définition du vecteur normal à la surface de délimitation	
	`A4=... B4=... C4=...:`	Définition en début de bloc
	`A5=... B5=... C5=...:`	Définition en fin de bloc
`G40 X... Y... Z...:`	Désactiver la correction de rayon d'outil **Remarque :** La désactivation doit intervenir dans un bloc linéaire (G0/G1) avec déplacements d'axes géométriques.	

Remarque

Les instructions G pour la sélection de CRO 3D sont évaluées dans le bloc d'accostage, donc typiquement dans le bloc contenant G41 ou G42.

G41 ou G42 peut également être programmée dans des blocs sans déplacement pour les axes géométriques concernés par la correction. Dans ce cas, le bloc d'accostage est le premier bloc de déplacement après ce type de bloc.

Tout changement de la variante CRO 3D, alors que la correction du rayon d'outil est active, est ignoré sans alarme.

Exemple

```
Code de programme                                  Commentaire
N10 $TC_DP1[1,1]=120                               ; Fraise cylindrique
N20 $TC_DP6[1,1]=10
N30 $TC_DP15[1,1]=-3
...
; Usinage avec fraise cylindrique et CUT3DCCD
N110 TRAORI                                        ; Activation de la transformation.
N120 A4=0 B4=0 C4=1                                ; Définition de la normale à la surface
                                                   de délimitation en début de bloc.
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000
```

Code de programme	**Commentaire**
N140 X10 Y0 Z0 G41 **CUT3DCCD** CDOF2 G64	; Activation du fraisage périphérique 3D en fonction de la surface de délimitation + désactivation de la détection des violations du contour.
N150 X20	
N160 X30 A45	; Angle obtus ==> pas de pénétration
N170 X40 A-45	; Angle aigu ==> pénétration
N180 X55	
N190 Y10 Z10	; Mouvement dans la direction de l'outil.
N200 Y20	
N210 C45	; Simple changement d'orientation.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; Changement de la surface.
N240 Y50 G40	; Désactivation de la correction du rayon d'outil.
...	

plus d'informations

Type d'outil

Le type d'outil (paramètre d'outil $TC_DP1) est évalué. Seules les fraises à corps cylindrique (fraises cylindriques, convexes ou demi-cercle et, cas limite, la fraise à bout hémisphérique) sont autorisées. Cela correspond aux types d'outil 1 à 399 à l'exception des numéros 111 et 155 à 157.

Outils standard avec arrondi

L'arrondi de l'outil standard est défini par le paramètre d'outil $TC_DP7. L'écart de l'arrondi de l'outil réel par rapport à l'outil standard est défini par le paramètre d'outil $TC_DP16.

Exemple : Fraise convexe demi-cercle avec un rayon plus petit par rapport à l'outil standard.

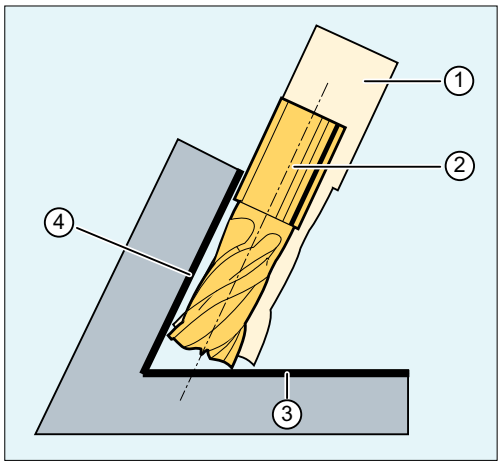
Type d'outil	Rayon de tige (R)	Rayon de congé (r)
Outil standard avec arrondi	R = $TC_DP6	r = $TC_DP7
Outil réel avec arrondi Types d'outil 121 et 131 fraises convexes demi-cercle (fraises deux tailles à queue avec arrondi)	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16

Dans cet exemple, $TC_DP15 + OFFN ainsi que $TC_DP16 sont négatifs.

CRO 3D avec CUT3DCCD : Trajectoire du centre de l'outil avec déplacement jusqu'à la surface de délimitation

En cas d'utilisation d'un outil avec un rayon plus petit que celui de l'outil standard correspondant, une fraise pénétrant dans le sens longitudinal avance jusqu'à ce qu'elle touche de nouveau le fond de la poche. Ainsi l'angle formé par la surface d'usinage et la surface de délimitation est évidé autant que l'outil le permet. Il s'agit d'un mode d'usinage mixte par fraisage périphérique et par fraisage en bout. De manière analogue à l'outil avec un rayon

réduit, le déplacement s'effectue dans la direction inverse dans le cas d'un outil avec un rayon plus grand.



① Outil standard
② Outil avec un rayon plus petit avec pénétration jusqu'à la surface de délimitation
③ Surface de délimitation
④ Surface d'usinage

Par rapport à toutes les autres corrections d'outil du groupe d'instructions G 22, un paramètre d'outil $TC_DP6 spécifié pour CUT3DCCD est sans signification pour le rayon d'outil et n'influence pas la correction résultante. Le décalage de correction résulte de la somme de la valeur d'usure du rayon de l'outil (paramètre d'outil $TC_DP15) et d'un décalage d'outil OFFN programmé pour le calcul du décalage perpendiculaire à la surface de délimitation.

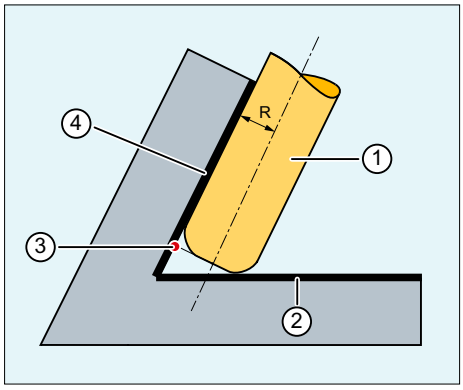
Le programme pièce généré ne permet pas de savoir si la surface usinée se situe à gauche ou à droite de la trajectoire. On part donc d'un rayon positif et d'une valeur d'usure négative de l'outil original. Une valeur d'usure négative décrit toujours un outil de diamètre réduit.

Utilisation d'outils cylindriques

En cas d'utilisation d'outils cylindriques, un déplacement est uniquement nécessaire si la surface d'usinage et la surface de délimitation forment un angle aigu (moins de 90 degrés). L'utilisation de fraises convexes demi-cercle (fraises deux tailles à queue avec arrondi) exige un déplacement de l'outil dans le sens longitudinal aussi bien en présence d'un angle aigu que d'un angle obtus.

CRO 3D avec CUT3DCC : contour de la surface d'usinage

Si CUT3DCC est actif avec une fraise convexe demi-cercle, la trajectoire programmée se rapporte à une fraise cylindrique fictive de diamètre identique. La référence de trajectoire qui en résulte est représentée sur la figure suivante pour l'utilisation d'une fraise convexe demi-cercle.



① Fraise convexe demi-cercle
② Surface de délimitation
③ Référence de trajectoire
④ Surface d'usinage
R Rayon de tige (rayon d'outil)

L'angle formé par la surface d'usinage et la surface de délimitation peut également passer d'un angle aigu à un angle obtus, et vice versa, à l'intérieur d'un bloc.

L'outil utilisé réellement peut être aussi bien plus grand que plus petit que l'outil standard. Cependant le rayon d'arrondi résultant ne doit pas devenir négatif et le signe du rayon d'outil résultant doit rester inchangé.

Avec CUT3DCC, le programme pièce CN se rapporte au contour de la surface d'usinage. Comme pour la correction de rayon d'outil conventionnelle, on tient compte du rayon total constitué des composantes suivantes :

- Rayon d'outil (paramètre d'outil $TC_DP6)
- Valeur d'usure (paramètre d'outil $TC_DP15)
- Décalage d'outil OFFN programmé pour le calcul du décalage perpendiculaire à la surface de délimitation.

La position de la surface de délimitation est déterminée à partir de la différence suivante :

Dimensions de l'outil standard - rayon d'outil (paramètre d'outil $TC_DP6)

Advanced Surface / Top Surface

Remarque

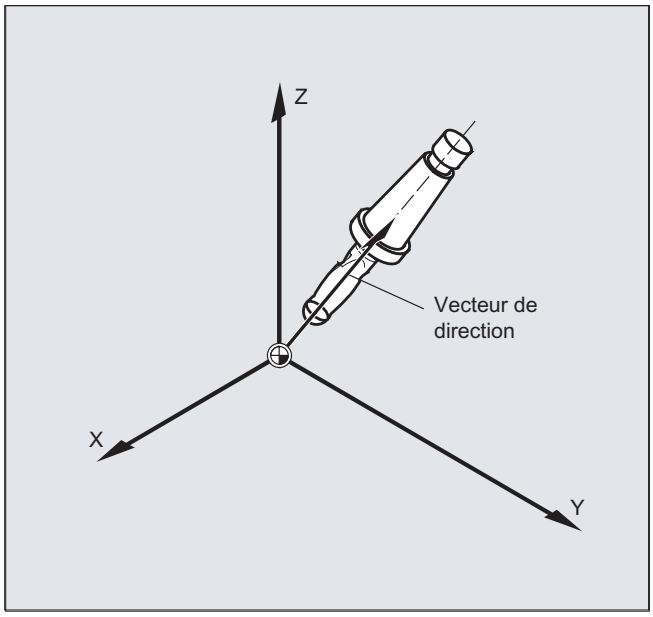
Pour l'utilisation de la correction du rayon d'outil CUT3DCC / CUT3DCCD en association avec la fonction soumise à la licence "Advanced Surface" ou "Top Surface", tenir compte des recommandations de réglages relatives à "Advanced Surface" / "Top Surface".

Pour vérifier les paramètres réglés, utiliser les programmes de vérification spéciaux, disponibles sur le portail SIOS.

- Programmes de vérification pour Advanced Surface (https://support.industry.siemens.com/cs/ww/fr/view/78956392/en)
- Programmes de vérification pour Top Surface (https://support.industry.siemens.com/cs/ww/fr/view/109738423/en)

11.6 Orientation de l'outil (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

On entend par orientation de l'outil l'orientation géométrique de l'outil dans l'espace. Sur une machine-outil à cinq axes, l'orientation de l'outil peut être spécifiée par des instructions du programme.



Des déplacements d'arrondissement de l'orientation activés par `OSD` et `OST` sont formés différemment selon le type d'interpolation pour l'orientation de l'outil.

En cas d'interpolation vectorielle active, le tracé lissé de l'orientation est également interpolé par l'interpolation vectorielle. Par contre, en cas d'interpolation d'axe rotatif active, l'orientation est directement lissée par des déplacements d'axe rotatif.

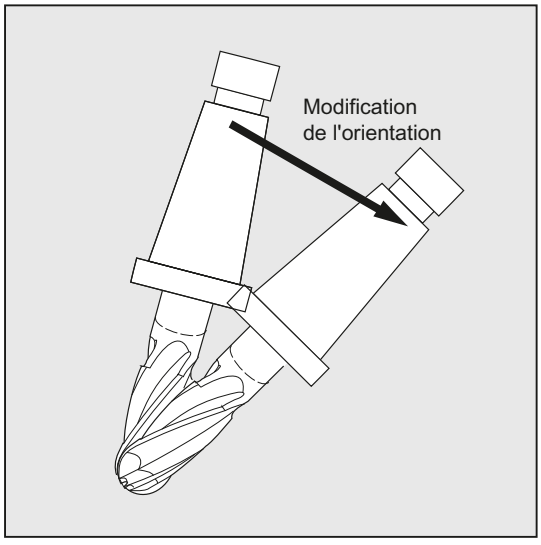
Programmation

Programmation du changement d'orientation :

On peut programmer un changement d'orientation de l'outil par :

- programmation directe des axes rotatifs `A`, `B`, `C` (interpolation d'axe rotatif)
- Angles d'Euler ou RPY
- Vecteur de direction (interpolation vectorielle par indication de `A3` ou de `B3` ou de `C3`)
- `LEAD/TILT` (fraisage en bout)

Le système de coordonnées de référence est soit le système de coordonnées machine (`ORIMKS`), soit le système de coordonnées pièce courant (`ORIWKS`).



Programmation de l'orientation de l'outil :

ORIC :	orientation et déplacement le long de la trajectoire en parallèle
ORID :	orientation et déplacement le long de la trajectoire successifs
OSOF :	aucun lissage de l'orientation
OSC :	orientation constante
OSS :	lissage de l'orientation seulement en début du bloc
OSSE :	lissage de l'orientation en début et en fin de bloc
ORIS :	Vitesse, en degrés par mm, du changement d'orientation lors de l'activation du lissage de l'orientation (s'applique à OSS et OSSE)
OSD :	Arrondissement de l'orientation par prescription de la longueur d'arrondissement avec la donnée de réglage : SD42674 $SC_ORI_SMOOTH_DIST
OST :	Arrondissement de l'orientation par prescription de la tolérance d'angle en degrés en cas d'interpolation vectorielle avec la donnée de réglage : SD42676 $SC_ORI_SMOOTH_TOL En cas d'interpolation d'axe rotatif, la tolérance prescrite est prise comme étant l'écart maximal des axes d'orientation.

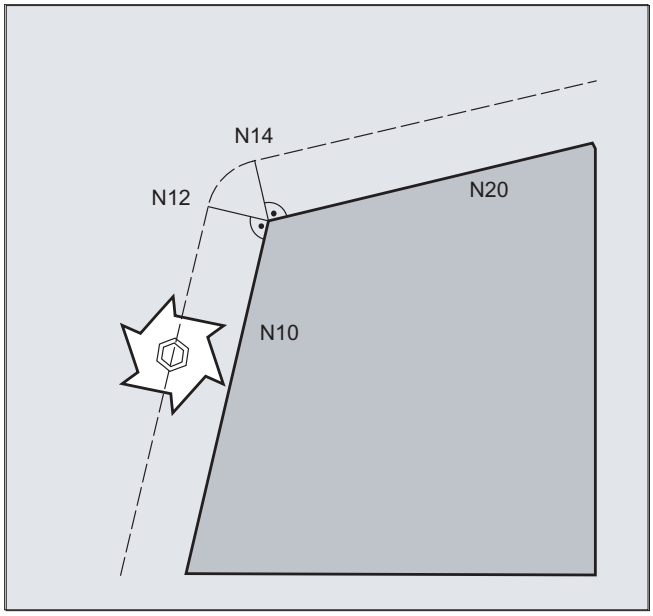
Remarque

Toutes les instructions réalisant un arrondissement de l'orientation de l'outil (OSOF, OSC, OSS, OSSE, OSD et OST) sont regroupées dans le groupe G 34. Elles sont à effet modal, autrement dit, une seule de ces instructions peut être exécutée à la fois.

Exemples

Exemple 1 : ORIC

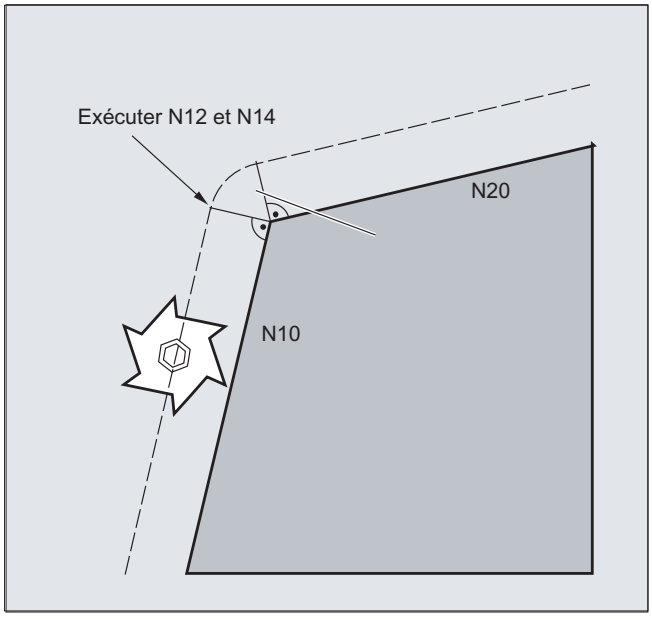
Si, entre les blocs de déplacement `N10` et `N20`, deux blocs ou plus avec des changements d'orientation (par ex. `A2=... B2=... C2=`) sont programmés et que `ORIC` est activée, le bloc à interpolation circulaire inséré est réparti sur ces blocs intermédiaires conformément à la valeur des variations angulaires.



Code de programme	Commentaire
ORIC	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 C2=… B2=…	; Le bloc à interpolation circulaire qui est inséré à l'angle saillant est réparti sur N12 et N14 en fonction du changement d'orientation. Le mouvement circulaire et le changement d'orientation sont exécutés en parallèle.
N14 C2=… B2=…	
N20 X =…Y=… Z=… G1 F200	

Exemple 2 : ORID

Si `ORID` est activée, tous les blocs situés entre les deux blocs de déplacement sont exécutés à la fin du premier bloc de déplacement. Le bloc à interpolation circulaire avec orientation constante est exécuté directement avant le deuxième bloc de déplacement.



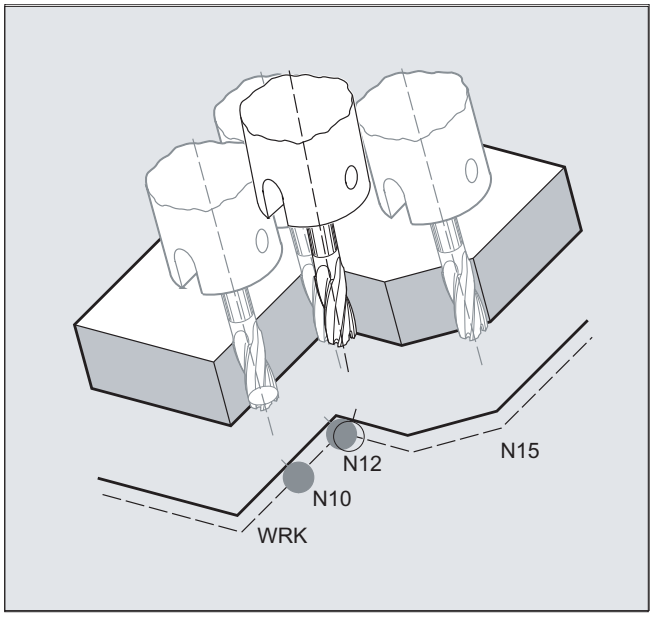
Code de programme	Commentaire
ORID	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 A2=… B2=… C2=…	; Le bloc N12 et N14 est exécuté à la fin de N10. Puis le bloc à interpolation circulaire se retire avec l'orientation courante.
N14 M20	; Fonctions auxiliaires, etc.
N20 X… Y… Z…	

Remarque

L'instruction de programme qui est active dans le premier bloc de déplacement d'un angle saillant est déterminante pour le type de changement d'orientation à cet angle saillant.

Sans changement d'orientation : Si l'orientation n'est pas modifiée à la fin du bloc, la section de l'outil est un cercle en contact avec les deux contours.

Exemple 3 : changement d'orientation à un angle rentrant



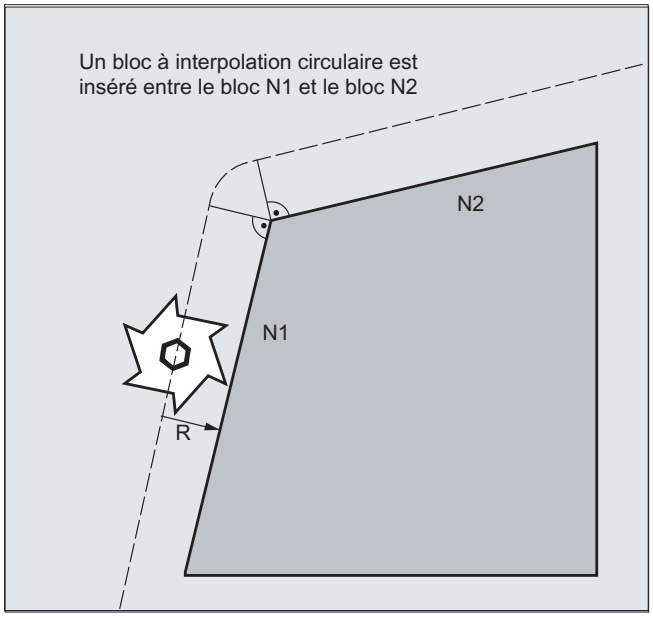
Code de programme
ORIC
N10 X …Y… Z… G1 F500
N12 X …Y… Z… A2=… B2=… C2=…
N15 X …Y… Z… A2=… B2=… C2=…

Informations complémentaires

Comportement aux angles saillants

Un bloc à interpolation circulaire avec un rayon égal à celui de la fraise est toujours inséré aux angles saillants.

Les instructions de programme `ORIC` et `ORID` permettent de définir si les changements d'orientation qui ont été programmés entre le bloc `N1` et le bloc `N2` seront effectués avant le début du bloc à interpolation circulaire inséré ou en même temps que celui-ci.



Si un changement d'orientation est nécessaire aux angles saillants, celui-ci peut s'effectuer parallèlement à l'interpolation ou bien séparément, en même temps que le déplacement le long de la trajectoire.

Avec `ORID`, les blocs insérés sont exécutés d'abord sans déplacement le long de la trajectoire. Le bloc de raccordement par interpolation circulaire est inséré directement avant le second des deux blocs de déplacement qui forment l'angle.

Si plusieurs blocs d'orientation sont insérés à un angle saillant et si `ORIC` est actif, le mouvement circulaire est réparti sur les différents blocs insérés, conformément aux valeurs des changements d'orientation de ces blocs.

Arrondissement de l'orientation avec OSD ou OST

Lors de l'arrondissement avec `G642`, l'écart maximal pour les axes de contour et les axes d'orientation ne peut pas varier beaucoup. La tolérance, la plus petite des deux, détermine la forme du déplacement d'arrondissement et de la tolérance d'angle et permet de lisser assez fortement le tracé de l'orientation sans écarts de contour importants.

L'activation de `OSD` ou `OST` permet, avec une longueur d'arrondissement ou une tolérance d'angle prescrites, de lisser "généreusement" de très faibles écarts du tracé de l'orientation sans écarts de contour importants.

Remarque

A la différence de l'arrondissement du contour (et du tracé de l'orientation) avec `G642`, aucun bloc séparé n'est formé lors de l'arrondissement de l'orientation avec `OSD` ou `OST`, mais le déplacement d'arrondissement est directement introduit dans les blocs d'origine programmés.

Avec `OSD` ou `OST`, aucune transition entre blocs ne peut être arrondie pour lesquels un changement du type d'interpolation pour l'orientation de l'outil (vecteur → axe rotatif, axe rotatif → vecteur) a lieu. Ces transitions entre blocs peuvent éventuellement être arrondies avec les fonctions d'arrondissement conventionnelles `G641`, `G642` ou `G643`.

11.7 Numéros D libres, numéro de tranchant

11.7.1 Numéros D libres, numéro de tranchant (adresse CE)

Numéro D

Les numéros D peuvent être utilisés en tant que numéros de correcteur. En outre, CE permet l'adressage du numéro de tranchant. L'écriture du numéro de tranchant est possible à l'aide de la variable système $TC_DPCE.

Préréglage : n° correction == n° tranchant

Le nombre maximal de numéros D (numéros de tranchant) et le nombre maximal de tranchants par outil sont définis dans les paramètres machine (→ constructeur de la machine-outil). Les instructions suivantes sont recommandées uniquement lorsque le nombre maximal de numéros de tranchant (paramètre machine MD18105) est plus grand que le nombre de tranchants par outil (MD18106). Veuillez observer les indications du constructeur de machines.

Remarque

Outre les numéros D relatifs, vous pouvez également attribuer des numéros D "absolus" (1-32000) sans référence à un numéro T (avec la fonction "structure horizontale des numéros D").

Bibliographie

Description fonctionnelle Fonctions de base, Correction d'outil (W1)

11.7.2 Libre affectation des numéros D : contrôle des numéros D (CHKDNO)

L'instruction `CKKDNO` permet un contrôle d'univocité de l'attribution des numéros D. Les numéros D de tous les outils définis dans une unité TO ne peuvent être attribués qu'une fois. La vérification ne porte pas sur les outils de rechange.

Syntaxe

```
state=CHKDNO(Tno1,Tno2,Dno)
```

Signification

`state` :	=TRUE :	Les numéros D ont été attribués de façon univoque dans le domaine contrôlé.
	=FALSE :	Une collision entre des numéros D a été constatée ou le paramétrage est incorrect. Tno1, Tno2 et Dno sont les paramètres ayant conduit à la collision. Ces données peuvent être exploitées dans le programme pièce.
`CHKDNO(Tno1,Tno2)` :	Tous les numéros D des outils indiqués sont comparés.	
`CHKDNO(Tno1)` :	Tous les numéros D de l'outil Tno1 sont comparés à tous les numéros D des autres outils.	
`CHKDNO` :	Tous les numéros D de tous les outils sont comparés à tous les numéros D de tous les autres outils.	

11.7.3 Libre affectation des numéros D : modification des numéros D (GETDNO, SETDNO)

L'attribution des numéros D doit être univoque. Deux tranchants distincts d'un outil ne peuvent pas posséder le même numéro D.

GETDNO

Cette instruction fournit le numéro D d'un certain tranchant (ce) d'un outil avec le numéro T t. S'il n'existe pas de numéro D correspondant aux paramètres indiqués, la valeur d=0 est attribuée. Si le numéro D n'est pas valable, d contient une valeur supérieure à 32000.

SETDNO

Avec cette instruction, vous attribuez la valeur d au numéro D du tranchant ce de l'outil t. ▯Le paramètre state fournit le résultat de l'instruction (TRUE ou FALSE). S'il n'existe pas d'enregistrement correspondant aux paramètres indiqués, le résultat de l'instruction est FALSE. Les erreurs de syntaxe génèrent une alarme. La valeur 0 ne peut pas être attribuée explicitement à un numéro D.

Syntaxe

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

Signification

`d` :	Numéro D du tranchant de l'outil
`t` :	Numéro T de l'outil
`ce` :	Numéro du tranchant (numéro CE) de l'outil
`state` :	Indique si l'instruction a pu être exécutée sans erreur (TRUE ou FALSE).

Exemple de changement de nom d'un numéro D

```
Programmation                                          Commentaire
$TC_DP2[1,2] = 120
$TC_DP3[1,2] = 5.5
$TC_DPCE[1,2] = 3                                      ; Numéro de tranchant CE
...
N10 def int N°DAncien, N°DNouveau = 17
N20 N°DAncien = GETDNO(1,3)
N30 SETDNO(1,3,N°DNouveau)
```

Le nouveau correcteur D17 est affecté au tranchant CE=3. L'accès aux données de ce tranchant a dorénavant lieu avec le numéro D17 ; ceci concerne les paramètres système aussi bien que la programmation avec l'adresse CN.

11.7.4 Libre affectation des numéros D : détermination du numéro T correspondant au numéro D prescrit (GETACTTD)

Avec l'instruction `GETACTTD`, vous déterminez le numéro T correspondant à un numéro D absolu. Il n'y a pas de contrôle d'univocité. En présence de plusieurs numéros D identiques dans une unité TO, l'instruction fournit le numéro T du premier outil trouvé. En cas d'utilisation de la "structure horizontale des numéros D", l'instruction ne présente pas d'intérêt, car son résultat est toujours "1" (pas de numéro T dans la gestion des données).

Syntaxe

```
Etat=GETACTTD(Tnr,Dnr)
```

Signification

`Dnr` :	Numéro D pour lequel le numéro T doit être recherché.	
`Tnr` :	Numéro T trouvé	
`status` :	Valeur :	Signification :
	0	Le numéro T a été trouvé. Le paramètre nº T contient sa valeur.
	-1	Il n'existe pas de numéro T pour le numéro D indiqué ; nº T=0.
	-2	Le numéro D n'est pas univoque. Le paramètre nº T contient la valeur du premier outil trouvé comportant le numéro D dont la valeur est nº D.
	-5	La fonction n'a pas pu être exécutée pour une autre raison.

11.7.5 Libre affectation des numéros D : numéros D déclarés non valides (DZERO)

L'instruction `DZERO` sert à faciliter le réoutillage. Les correcteurs déclarés non valides ne font plus l'objet d'un contrôle par l'instruction `CHKDNO`. Pour avoir de nouveau accès aux correcteurs, il faut procéder à une nouvelle affectation des numéros D avec `SETDNO`.

Syntaxe

```
DZERO
```

Signification

`DZERO` :	Déclare tous les numéros D de l'unité TO non valides.

11.8 Cinématique de l'organe porte-outil

Conditions

Un organe porte-outil ne peut orienter un outil dans toutes les directions de l'espace que

- si deux axes rotatifs V_1 et V_2 sont présents.
- si les axes rotatifs sont perpendiculaires les uns par rapport aux autres.
- si l'axe longitudinal de l'outil est perpendiculaire au deuxième axe rotatif V_2.

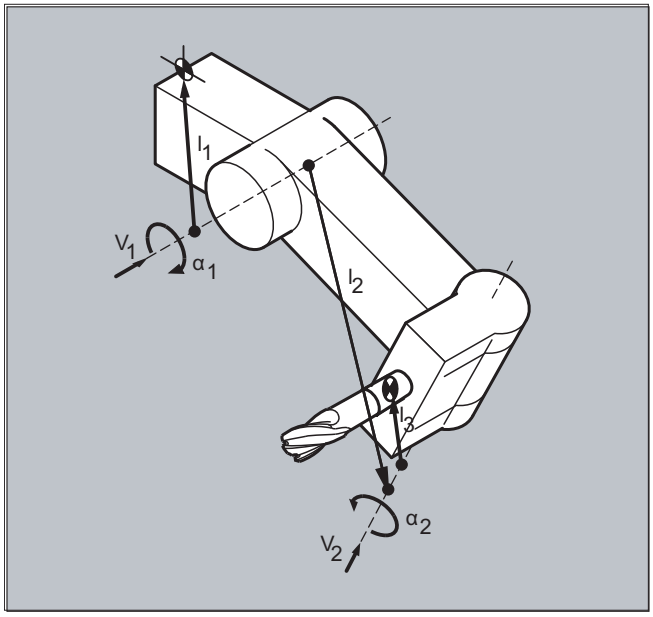
En outre, la condition ci-dessous doit être satisfaite dans le cas des machines sur lesquelles il doit être possible de régler toutes les orientations :

- - l'orientation de l'outil doit être perpendiculaire au premier axe rotatif V_1.

Fonction

La cinématique d'un organe porte-outil ayant deux axes rotatifs au maximum v_1 ou v_2 est décrite à l'aide des 17 variables système allant de $TC_CARR1[m] à $TC_CARR17[m]. Cette description comprend :

- le vecteur distance entre le premier axe rotatif et le point de référence de l'organe porte-outil I_1, le vecteur distance entre le premier et le second axe rotatif I_2, le vecteur distance entre le second axe rotatif et le point de référence de l'outil I_3.
- le vecteur de direction des deux axes rotatifs V_1, V_2.
- les angles rotatifs α1, α2 autour des deux axes. Les angles rotatifs sont positifs s'ils provoquent une rotation en sens horaire pour un observateur regardant dans la direction des axes rotatifs.



Pour les machines à **cinématique décomposée** (l'outil et la pièce sont orientables), les variables système ont été complétées par $TC_CARR18[m] bis $TC_CARR23[m].

Paramètre

Fonction des variables système pour organes porte-outil orientables			
Désignation	**Composante x**	**Composante y**	**Composante z**
Vecteur offset l_1	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
Vecteur offset l_2	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
Axe rotatif v_1	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
Axe rotatif v_2	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
Angle rotatif α_1 Angle rotatif α_2	$TC_CARR13[m] $TC_CARR14[m]		
Vecteur offset l_3	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]

Extension des variables système pour organes porte-outil orientables			
Désignation	**Composante x**	**Composante y**	**Composante z**
Vecteur offset l_4	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
Descripteur d'axe Axe rotatif v_1 Axe rotatif v_2	Descripteur d'axe pour les axes rotatifs $v_1$et v_2 (préréglage avec zéro) $TC_CARR21[m] $TC_CARR22[m]		
Type de cinématique **T**ool **P**art **M**ixed mode	$TC_CARR23[m]		
	Type de cinématique T ->	Type de cinématique P ->	Type de cinématique M
	seul l'outil est orientable (préréglage).	seule la pièce est orientable.	pièce et outil sont orientables.
Offset des axes rotatifs v_1 et v_2	Angle en degrés des axes rotatifs $v_1$et v_2 en position initiale $TC_CARR24[m] $TC_CARR25[m]		
Décalage angulaire des axes rotatifs v_1 et v_2	Offset en degrés de la denture Hirth des axes rotatifs $v_1$et v_2 $TC_CARR26[m] $TC_CARR27[m]		
Incrément angulaire v_1 Axe rotatif v_2 Axe rotatif	Incrément en degrés de la denture Hirth des axes rotatifs $v_1$et v_2 $TC_CARR28[m] $TC_CARR29[m]		
Position min. Axe rotatif v_1 Axe de rotatif v_2	Limite logicielle pour position minimale des axes rotatifs $v_1$et v_2 $TC_CARR30[m] $TC_CARR31[m]		
Position max. Axe rotatif v_1 Axe rotatif v_2	Limite logicielle pour position maximale des axes rotatifs $v_1$et v_2 $TC_CARR32[m] $TC_CARR33[m]		
Nom de l'organe porte-outil	À la place d'un nombre, un organe porte-outil peut se voir affecter un nom. $TC_CARR34[m]		
Utilisateur : Nom d'axe 1 Nom d'axe 2 Identificateur	Utilisation prévue par l'utilisateur à l'intérieur des cycles de mesure. $TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
Position	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
Décalage fin	Paramètres pouvant être ajoutés **aux valeurs contenues dans les paramètres de base**.		

Extension des variables système pour organes porte-outil orientables			
Vecteur offset l_1	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
Vecteur offset l_2	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
Vecteur offset l_3	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
Vecteur offset l_4	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
Axe rotatif v_1	$TC_CARR64[m]		
Axe rotatif v_2	$TC_CARR65[m]		

Remarque

Explication des paramètres

"m" est le numéro de l'organe porte-outil à décrire.

Les paramètres $TC_CARR47 jusqu'à $TC_CARR54 ainsi que $TC_CARR61 jusqu'à $TC_CARR63 ne sont pas définis et génèrent une alarme en cas de tentative d'accès en lecture ou en écriture.

Vous pouvez choisir librement les points initiaux et finals des vecteurs distances sur les axes. Les angles rotatifs α_1, $\alpha_2$autour des deux axes sont égaux à 0° lorsque l'organe porte-outil est en position initiale. Il existe donc un grand nombre de possibilités pour décrire la cinématique d'un organe porte-outil.

Vous pouvez décrire des organes porte-outil n'ayant qu'un axe rotatif ou aucun axe rotatif en annulant les composantes du vecteur de direction d'un ou des deux axes rotatifs.
Dans le cas d'un organe porte-outil sans axe rotatif, les vecteurs distances ont le même effet que des corrections d'outil supplémentaires, dont les composantes ne sont pas influencées par le basculement entre les plans d'usinage (G17 à G19).

Extension des paramètres

Paramètres des axes rotatifs

Les variables système $TC_CARR24[m] à $TC_CARR33[m], dont les significations sont indiquées ci-dessous, sont introduites :

Offset des axes rotatifs v_1 et v_2	Changement de position de l'axe rotatif v_1 ou v_2 lorsque l'organe porte-outil orientable est en position initiale.
Décalage **angulaire/ incrément** angulaire des axes rotatifs $v_1$et v_2	Offset ou incrément angulaire de la denture Hirth des axes rotatifs v_1 et v_2. L'angle programmé ou calculé est arrondi à la valeur la plus proche qui s'obtient à l'aide de la formule phi = s + n * d, n étant un nombre entier.
Position minimale et maximale Axes rotatifs v_1, v_2	Position minimale/maximale des axes rotatifs Angles limites (limite logicielle) des axes rotatifs v1 et v2.

Paramètres pour l'utilisateur

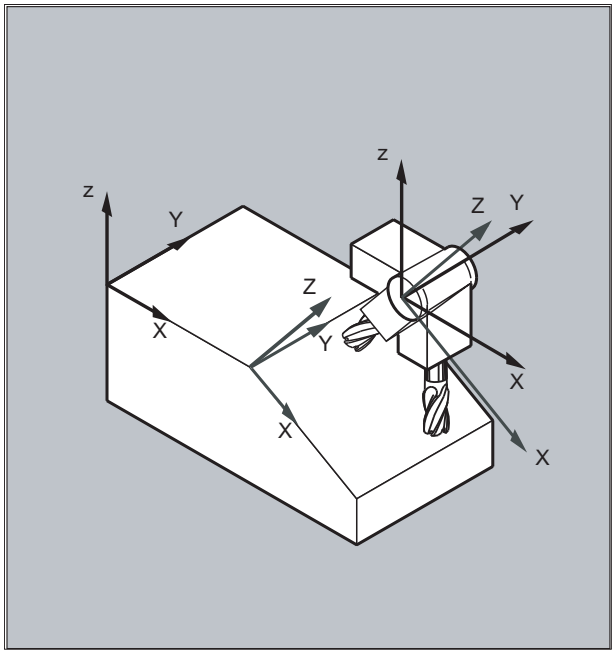
$TC_CARR34 à $TC_CARR40 contiennent des paramètres qui sont mis à la libre disposition de l'utilisateur et qui, jusqu'à la version de logiciel 6.4, ne sont pas exploités en standard à l'intérieur du NCK ou qui sont sans signification.

Paramètres de décalage fin

$TC_CARR41 à $TC_CARR65 contiennent des paramètres de décalage fin pouvant être ajoutés aux valeurs contenues dans les paramètres de base. On obtient une valeur de décalage fin affectée à un paramètre de base lorsque la valeur 40 est ajoutée au numéro de paramètre.

Exemple

L'organe porte-outil utilisé dans l'exemple suivant peut être décrit entièrement par une rotation autour de l'axe Y.



Code de programme	**Commentaire**
N10 $TC_CARR8[1]=1	; Définition de la composante Y du premier axe rotatif de l'organe porte-outil 1.
N20 $TC_DP1[1,1]=120	; Définition d'une fraise cylindrique.
N30 $TC_DP3[1,1]=20	; Définition d'une fraise cylindrique d'une longueur de 20 mm.
N40 $TC_DP6[1,1]=5	; Définition d'une fraise cylindrique d'un rayon de 5 mm.
N50 ROT Y37	; Définition d'un frame avec rotation de 37° autour de l'axe Y.
N60 X0 Y0 Z0 F10000	; Accostage du point de départ.
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; Réglage correction de rayon, correction de longueur d'outil dans le frame pivoté, sélection organe porte-outil 1, outil 1.

Code de programme	Commentaire
N80 X40	; Effectuer l'usinage sous un angle rotatif de 37°.
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Informations complémentaires

Cinématique décomposée

Pour les machines à cinématique décomposée (l'outil et la pièce sont orientables), les variables système supplémentaires `$TC_CARR18[m]` à `$TC_CARR23[m]` permettent de définir :

l'organe porte-pièce orientable à l'aide :

- du vecteur distance entre le second axe rotatif V_2 et le point de référence d'une table porte-pièce pivotante I_4 du troisième axe rotatif.

les axes rotatifs à l'aide :

- des deux descripteurs d'axes de canal associés aux axes rotatifs V_1et V_2, et dont les positions peuvent être interrogées pour la détermination de l'orientation.

le type de cinématique à l'aide d'une des valeurs T, P ou M :

- type de cinématique T : seul l'outil est orientable.
- type de cinématique P : seule la pièce est orientable.
- type de cinématique M : l'outil et la pièce sont orientables.

Effacement des données d'organe porte-outil

Avec `$TC_CARR1[0] = 0`, vous pouvez effacer les données de tous les organes porte-outil.

La valeur du paramètre pour le type de cinématique `$TC_CARR23[T] = T` doit être une des trois lettres majuscules ou minuscules (T, P, M) ; ce paramètre ne doit donc pas être effacé.

Modification des données d'organe porte-outil

Vous pouvez modifier chacune des valeurs en lui affectant une nouvelle valeur dans le programme pièce. Tout autre caractère que T, P ou M conduit à une alarme lors de la tentative d'activer l'organe porte-outil orientable.

Lecture des données d'organe porte-outil

Vous pouvez lire chacune des valeurs en l'affectant à une variable dans le programme pièce.

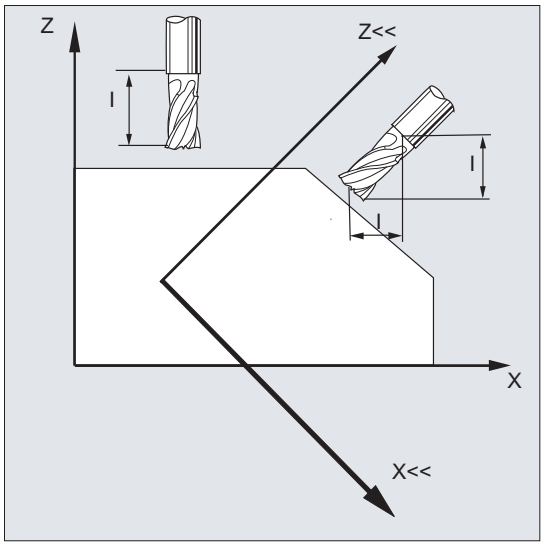
Décalage fin

Une valeur de décalage fin non autorisée n'est détectée que lorsqu'un organe porte-outil orientable contenant une telle valeur est activé et que, dans le même temps, la donnée de réglage SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE.

La valeur absolue du décalage fin autorisée est limitée par les paramètres machine à une valeur maximale admissible.

11.9 Correction de longueur d'outil pour organes porte-outils orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Les composantes de longueur d'outil varient lorsque l'orientation dans l'espace de l'outil en question change.



Par conséquent, après un changement, par réglage manuel ou remplacement de l'organe porte-outil à orientation spatiale fixe par exemple, les composantes de longueur d'outil sont à redéfinir. Ceci se fait avec les instructions `TCOABS` et `TCOFR`.

Avec un organe porte-outil orientable d'un frame actif, il est possible de déterminer, lors de la sélection de l'outil avec `TCOFRZ`, `TCOFRY` et `TCOFRX`, la direction vers laquelle l'outil doit pointer.

Syntaxe

```
TCARR=[<m>]
TCOABS
TCOFR
TCOFRZ
avec TCOFRY
avec TCOFRX
```

Signification

`TCARR=[<m>]` :	Appel de l'organe porte-outil ayant le numéro "m"
`TCOABS` :	Calculer les composantes de longueur d'outil à partir de l'orientation de l'organe porte-outil.
`TCOFR` :	Détermination des composantes de longueur d'outil à partir de l'orientation du frame actif.
`TCOFRZ` :	Organe porte-outil orientable du frame actif dont l'outil est pointé en direction Z
`TCOFRY` :	Organe porte-outil orientable du frame actif dont l'outil est pointé en direction Y
`TCOFRX` :	Organe porte-outil orientable du frame actif dont l'outil est pointé en direction X

Informations complémentaires

Correction de longueur d'outil à partir de l'organe porte-outil (TCOABS)

`TCOABS` calcule la correction de longueur d'outil sur la base des angles d'orientation de l'organe porte-outil, qui sont rangés dans les variables système $TC_CARR13 et $TC_CARR14.

Pour la définition de la cinématique de l'organe porte-outil avec des variables système, voir "Cinématique de l'organe porte-outil (Page 472)".

Pour que la correction de longueur d'outil soit recalculée en cas de changement de frame, il convient de rappeler l'outil.

direction d'outil du frame actif

L'organe porte-outil orientable peut être réglé de telle façon que l'outil soit orienté dans les directions suivantes :

- dans la direction Z avec `TCOFR` ou `TCOFRZ`
- dans la direction Y avec `TCOFRY`
- dans la direction X avec `TCOFRX`

Toute commutation entre `TCOFR` et `TCOABS` provoque un recalcul de la correction de longueur d'outil.

Appel de l'organe porte-outil (TCARR)

Avec `TCARR`, vous appelez les données géométriques de l'organe porte-outil de numéro m (mémoire de correcteurs).

Avec m=0, vous désactivez l'organe porte-outil actif.

Les données géométriques de l'organe porte-outil ne deviennent opérantes qu'après l'appel de l'outil. L'outil sélectionné reste activé même après un changement d'organe porte-outil.

Les données géométriques de l'organe porte-outil peuvent aussi être définies dans le programme pièce, par le biais des variables système correspondantes.

Nouveau calcul de la correction de longueur d'outil (TCOABS) en cas de changement de frame

Pour que la correction de longueur d'outil soit recalculée en cas de changement de frame, il convient de rappeler l'outil.

Remarque

L'orientation de l'outil doit être adaptée manuellement au frame actif.

Lors du calcul de la correction de longueur d'outil, les angles de rotation de l'organe porte-outil font également l'objet d'un calcul. Etant donné que les organes porte-outil à deux axes de rotation ont généralement deux paires d'angles de rotation, avec lesquels on peut adapter l'orientation d'outil au frame actif, les valeurs des angles de rotation rangées dans les variables

système doivent correspondre au moins approximativement aux angles de rotation réglés mécaniquement.

Remarque

Orientation de l'outil

La commande n'est pas en mesure de contrôler la possibilité de réglage sur la machine des angles de rotation calculés sur la base de l'orientation du frame.

Si les axes de rotation de l'organe porte-outil sont situés de telle façon qu'il soit impossible à l'outil de rallier l'orientation calculée sur la base de l'orientation du frame, une alarme est émise.

Combiner la correction fine d'outil et les fonctionnalités de la correction de longueur d'outil n'est pas autorisé pour les organes porte-outils orientables. Si vous tentez d'appeler les deux fonctions en même temps, un message d'erreur vous sera donné.

Avec `TOFRAME` il est possible de définir un frame en fonction de la direction dans laquelle est orienté l'organe porte-outil. Pour de plus amples informations, se reporter au chapitre "Frames".

Quand une transformation d'orientation est active (transformation 3, 4, 5 axes), vous pouvez appeler un organe porte-outil ayant une orientation s'écartant de la position zéro, sans déclencher une alarme.

Paramètres de transfert des cycles standard et cycles de mesure

Pour les paramètres de transfert des cycles standard et cycles de mesure, on applique les plages de valeurs bien définies.

Dans le cas des valeurs angulaires, la plage de valeurs est définie ainsi :

- Rotation autour du 1er axe géométrique : -180 degrés jusqu'à +180 degrés
- Rotation autour du 2e axe géométrique : -90 degrés jusqu'à +90 degrés
- Rotation autour du 3e axe géométrique : -180 degrés jusqu'à +180 degrés

Voir chapitre Frames, "Rotation programmable (ROT, AROT, RPL)".

Remarque

Lors du transfert de valeurs angulaires à un cycle standard ou de mesure, il convient

d'arrondir à zéro toute valeur inférieure à l'unité de calcul de la CN !

L'unité de calcul de la CN pour les positions angulaires est définie dans le paramètre machine :

PM10210 $MN_INT_INCR_PER_DEG

11.10 Correction en ligne de la longueur d'outil (TOFFON, TOFFOF)

La variable système $AA_TOFF[<n>] permet de corriger en temps réel et en trois dimensions les longueurs d'outil effectives en fonction des trois directions de l'outil.

Les trois descripteurs d'axes géométriques servent d'indices <n>. De cette façon, le nombre des directions de correction activées est déterminé par les axes géométriques qui sont actifs au même moment.

Toutes les corrections peuvent être actives simultanément.

La fonction de correction des longueurs d'outil en ligne est applicable dans :

- Transformation de l'orientation TRAORI
- Organes porte-outil orientables TCARR

Remarque

La correction en ligne de longueur d'outil est une **option** qui doit être validée auparavant. Cette fonction n'a de sens qu'en liaison avec la transformation active de l'orientation ou avec un organe porte-outil orientable à l'état activé.

Syntaxe

```
TRAORI
TOFFON(<direction de correction>[,<valeur de l'off-
set>])
WHEN TRUE DO $AA_TOFF[<sens de correction>]            ; Dans des actions synchro-
                                                       nes.
...
TOFFOF(<sens de correction>)
```

Pour plus d'explications sur la programmation de la correction en ligne de la longueur d'outil dans des actions synchrones au déplacement, voir "Actions synchrones (Page 641)".

Signification

TOFFON :	**Activation** de la correction en ligne de la longueur d'outil	
	<sens de correction> :	Direction de l'outil (X, Y, Z) dans laquelle la correction en ligne de la longueur d'outil doit prendre effet.
	<valeur de l'offset> :	Dès que la fonction est activée, une valeur offset peut être indiquée pour la direction dans laquelle doit s'effectuer la correction et cette valeur fait immédiatement l'objet d'un mouvement de correction.
TOFFOF :	**Remise à zéro** de la correction en ligne de la longueur d'outil Les valeurs de correction dans le sens de correction spécifié sont remises à zéro et un arrêt du prétraitement des blocs est déclenché.	

Exemples

Exemple 1 : activation de la correction de longueur d'outil

Code de programme	Commentaire
PM21190 $MC_TOFF_MODE =1	; Accostage des valeurs absolues.
PM21194 $MC_TOFF_VELO[0] =1000	
PM21196 $MC_TOFF_VELO[1] =1000	
PM21194 $MC_TOFF_VELO[2] =1000	
PM21196 $MC_TOFF_ACCEL[0] =1	
PM21196 $MC_TOFF_ACCEL[1] =1	
PM21196 $MC_TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	; Activation de la transformation.
N20 TOFFON(Z)	; Activation de la correction en ligne de la longueur d'outil pour la direction Z de l'outil.
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; Interpolation d'une valeur de correction de longueur d'outil égale à 10 pour la direction Z de l'outil.
...	
N100 XOFFSET=$AA_TOFF_VAL[X]	; Affectation de la correction actuelle dans la direction X.
N120 TOFFON(X,-XOFFSET) G4 F5	; Remise à 0 de la correction de longueur d'outil dans la direction X de l'outil.

Exemple 2 : Désactivation de la correction de longueur d'outil

Code de programme	Commentaire
N10 TRAORI(1)	; Activation de la transformation.
N20 TOFFON(X)	; Activation de la correction en ligne de la longueur d'outil pour la direction X de l'outil.
N30 WHEN TRUE DO $AA_TOFF[X]=10 G4 F5	; Interpolation d'une valeur de correction de longueur d'outil égale à 10 pour la direction X de l'outil.
...	
N80 TOFFOF(X)	; L'offset de position de la direction d'outil X est supprimé : ...$AA_TOFF[X]=0 Aucun axe n'est déplacé. L'offset de position est ajouté à la position actuelle dans le SCP en fonction de l'orientation actuelle.

Informations complémentaires

Préparation des blocs

Lors de la préparation des blocs au cours de leur prétraitement, l'offset de longueur d'outil qui agit dans l'exécution de blocs est pris en considération. Afin de pouvoir exploiter au mieux les vitesses d'axes maximales autorisées, il est nécessaire d'arrêter la préparation des blocs avec un arrêt du prétraitement des blocs `STOPRE` pendant la réalisation d'un offset d'outil.

Cet offset d'outil est connu au moment du prétraitement des blocs si les corrections de longueur d'outil n'ont pas été modifiées après le lancement du programme ou - quand il y a eu modification des corrections de longueur d'outil - si le nombre de blocs exécutés est supérieur au nombre de blocs que peut recevoir le tampon IPO entre le prétraitement des blocs et leur exécution.

Variable $AA_TOFF_PREP_DIFF

Le degré de différence entre la correction actuellement effective dans l'interpolateur et la correction effective au moment de la préparation des blocs peut être interrogée dans la variable $AA_TOFF_PREP_DIFF[<n>].

Réglage des paramètres machine et des données de réglage

Les données système suivantes sont disponibles pour la correction en ligne de la longueur d'outil :

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE (réserve d'accélération pour déplacement forcé)
- MD21190 $MC_TOFF_MODE
  Le contenu de la variable système $AA_TOFF[<n>] fait l'objet d'un déplacement en absolu ou d'une intégration forcée.
- MD21194 $MC_TOFF_VELO (vitesse de la correction en ligne de la longueur d'outil)
- MD21196 $MC_TOFF_ACCEL (accélération de la correction en ligne de la longueur d'outil)
- Donnée de réglage pour la spécification de valeurs limites :
  SD42970 $SC_TOFF_LIMIT (limite supérieure de la valeur de correction de la longueur d'outil)

Bibliographie :
Description fonctionnelle Fonctions spéciales ; F2 : transformations multi-axes

11.11 Modification des données de correction pour outils orientables

11.11.1 Calcul de l'orientation (ORISOLH)

La fonction ORISOLH prédéfinie assiste l'utilisateur au réglage des positions d'axe rotatif d'une machine, de telle sorte qu'un outil de tournage soit orienté dans une position définie, relative à la pièce et indépendante de la cinématique. Une transformation 6 axes, paramétrée avec des chaînes cinématiques, doit être préalablement active.

Deux fonctions de base sont disponibles :

- Orienter l'outil
  Les deux angles β et γ sont spécifiés. La fonction calcule alors les angles des trois axes d'orientation nécessaires.
- Orienter directement l'outil
  Les angles du deuxième et troisième axe d'orientation sont spécifiés. La fonction calcule alors les angles correspondants β et γ ainsi que l'angle du premier axe d'orientation encore manquant.

Remarque

Ordre des axes d'orientation

En parcourant la chaîne cinématique qui décrit la structure de la machine en allant de la pièce jusqu'à l'outil, les règles suivantes s'appliquent pour l'ordre des trois axes d'orientation d'une transformation 6 axes :

- L'axe d'orientation le plus proche de la **pièce** est le **premier** axe d'orientation.
- L'axe d'orientation le plus proche de l'**outil** est le **troisième** axe d'orientation.

Le premier axe d'orientation est habituellement une broche, et la rotation correspondante est réalisée dans ces cas par un frame rotatif.

Syntaxe

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

Signification

`ORISOLH` :	Appel de fonction		
`<RetVal>` :	Valeur de retour de la fonction		
	Type de données :	INT	
	Plage de valeurs :	0, -2, -3, ..., -17	
	Valeurs :	0	La fonction a été terminée sans erreur.
		-2	Aucune transformation valide (transformation d'orientation 6 axes) n'est active.
		-3	Le premier paramètre (<Cntrl>) est négatif.
		-4	La position des unités du premier paramètre (<Cntrl>) n'est pas valide. Seules les valeurs 0 et 1 sont autorisées.
		-5	La position des dizaines du premier paramètre (<Cntrl>) n'est pas valide. Seules les valeurs 0 et 3 sont autorisées.
		-6	La position des centaines du premier paramètre (<Cntrl>) n'est pas valide. Seules les valeurs 0 et 1 sont autorisées.
		-7	La position des milliers du premier paramètre (<Cntrl>) n'est pas valide. Seules les valeurs 0 et 3 sont autorisées.
		-8	Dans la fonction "Orienter directement l'outil", l'angle γ est trop grand.
		-9	Au moins une des positions d'axe spécifiées dépasse une limite d'axe dans la fonction "Orienter directement l'outil".
		-10	Aucun outil n'est actif.
		-11	L'orientation demandée ne peut être réglée.
		-12	L'adaptation des angles d'axe libres pour denture Hirth est impossible pour la première et seule solution.
		-13	L'adaptation des angles d'axe libres pour denture Hirth est impossible pour la deuxième solution.
		-14	L'adaptation des angles d'axe libres pour denture Hirth est impossible pour aucune des deux solutions.
		-15	Le premier axe d'orientation est paramétré comme axe Hirth.
		-16	Le deuxième et le troisième axe rotatif sont paramétrés comme axes Hirth. Un des deux axes au maximum doit être un axe Hirth.
		-17	Pour la fonction "Pivotement direct", au moins une des positions d'axes spécifiées n'est pas compatible avec la denture Hirth correspondante.

<Cntrl> :	Commande le comportement de la fonction		
	Type de données :	INT	
	Le paramètre <Cntrl> est codé en décimal (position des unités jusqu'à la position des milliers) :		
	Position des unités :	La position des unités commande le comportement en cas d'erreur.	
		xxx**0**	En cas d'erreur (valeur retournée < 0), l'exécution du programme est annulée et l'alarme 14106 est émise. **Remarque :** Indépendamment de la valeur de la position des unités, l'alarme est également émise lorsque le paramètre <Cntrl> est négatif.
		xxx**1**	En cas d'erreur (valeur retournée < 0), **aucune** alarme n'est émise. L'utilisateur a la possibilité de programmer sa propre réaction.
	Position des dizaines :	Commande de comportement lors de la présence d'un axe d'orientation avec denture Hirth. **Remarque :** Ce paramètre n'est évalué que dans le contexte de la fonction "Orienter l'outil" (c.-à-d. lorsque la valeur de la position des centaines est "0").	
		xx**0**x	La position de l'axe est arrondie à la position la plus proche.
		xx**1**x	Les positions d'axe sont arrondies de telle sorte que l'écart de l'angle β par rapport à sa valeur programmée devienne minimal.
		xx**2**x	Les positions d'axe sont arrondies de telle sorte que l'angle β prenne la valeur maximale inférieure à la valeur programmée (β arrondi vers le bas).
		xx**3**x	Les positions d'axe sont arrondies de telle sorte que l'angle β prenne la valeur minimale supérieure à la valeur programmée (β arrondi vers le haut).
	Position des centaines :	Spécifie la fonction qui doit être exécutée ou la signification des deux paramètres suivants <W1> et <W2>.	
		x**0**xx	Fonction "Orienter l'outil" Les paramètres <W1> et <W2> ont la signification suivante : • <W1> = β • <W2> = γ Les angles correspondants des axes d'orientation sont calculés.
		x**1**xx	Fonction "Orienter directement l'outil" <W1> est la position spécifiée pour le deuxième axe d'orientation, <W2> est la position spécifiée pour le troisième axe d'orientation d'une transformation 6 axes. La fonction détermine la position du premier axe d'orientation ainsi que les angles β et γ, compatibles avec les deux spécifications de position. En l'absence d'erreur, la variable système $P_ORI_POS[<n>, <m>] fournit toujours deux solutions. Le premier indice <n> (0 ou 1) renvoie à la solution et le deuxième indice <m> (0 ... 2) à l'axe d'orientation :

			• $P_ORI_POS[0/1, **0**] : Position du premier axe d'orientation • $P_ORI_POS[0/1, **1**] : Angle β • $P_ORI_POS[0/1, **2**] : Angle γ La fonction vérifie si les positions spécifiées <W1> et <W2> sont compatibles avec des dentures Hirth ou limites logicielles éventuellement actives. Si ce n'est pas le cas, un numéro d'erreur correspondant est retourné (voir paramètre <RetVal>). Lors d'une sélection quelconque des deux angles <W1> et <W2>, le tranchant de l'outil ne se trouve généralement pas dans le plan d'usinage. L'angle γ formant l'écart entre le tranchant et le plan d'usinage ne doit pas dépasser la valeur limite définie par la donnée de réglage SD42999 $SC_ORISOLH_INCLINE_TOL.
	Position des milliers :	Spécifie comment les positions des solutions sont éventuellement modifiées lorsque la valeur de la position des centaines est "0", c.-à-d. pour la fonction "Orienter l'outil".	
		0xxx	Les positions d'axe calculées doivent être les plus proches possibles des positions d'axe machine.
		1xxx	Dans le cas des axes à valeur modulo, les positions d'axe calculées doivent être les plus proches possibles du milieu de la plage modulo, dans le cas des autres axes les plus proches possibles de 0. Cela signifie pour les axes non modulo que les positions d'axe sont limitées à la plage -180° … +180°.
		2xxx	Les positions d'axe calculées doivent être réduites à la plage -180° … +180° indépendamment du type d'axe.
<W1> :	Premier angle La signification est déterminée par la position des centaines du paramètre <Cntrl>.		
	Type de données :	REAL	
<W1> :	Deuxième angle La signification est déterminée par la position des centaines du paramètre <Cntrl>.		
	Type de données :	REAL	

Remarque

Les paramètres non programmés ont par défaut la valeur "0".

Informations complémentaires

Les variables système suivantes permettent de lire le nombre de solutions trouvées ainsi que les informations d'état relatives à l'exécution de la fonction ORISOLH :

Variable système	Signification		
$P_ORI_POS [<n>, <m>]	Fournit l'angle des axes d'orientation résultant de la programmation de l'orientation.		
	<n> :	Indice de la solution	
		Plage de valeurs :	0, 1
	<m> :	Indice de l'axe d'orientation	
		Plage de valeurs :	0 ... 2
			L'ordre des axes d'orientation (1 ... 3) se rapporte à la définition des axes dans $NT_ROT_AX_NAME.
	Lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil", les variables $P_ORI_POS[0/1, 1] et P_ORI_POS[0/1, 2] contiennent les valeurs des deux angles β et γ des deux solutions. La première solution enregistrée dans $P_ORI_POS[<n>, <m>], portant l'indice <n> = 0, est toujours la solution sélectionnée par la commande lorsque l'orientation demandée est approchée directement. Le deuxième indice <m> se rapporte à l'axe d'orientation, c.-à-d. $NT_ROT_AX_NAME. Les positions d'axe enregistrées dans $P_ORI_POS[<n>, <m>] tiennent compte des offsets renseignés dans $NK_OFF et $NK_OFF_FINE, c.-à-d. ces angles d'axe peuvent être utilisés sans aucune modification dans les blocs suivants pour régler l'orientation souhaitée. Si un axe rotatif est un axe Hirth, les positions de la solution sont arrondies à la position indexée la plus proche de la denture Hirth. Dans le cas des axes rotatifs avec denture Hirth, la variable système $P_ORI_DIFF permet de lire les différences des positions d'axe par rapport à la solution exacte ainsi que les solutions adaptées à l'indexage Hirth.		
$P_ORI_DIFF [<n>, <m>]	Fournit la différence entre les positions exactes des axes d'orientation et celles fournies par $P_ORI_POS résultant de la programmation de l'orientation.		
	<n> :	Indice de la solution	
		Plage de valeurs :	0, 1
	<m> :	Indice de l'axe d'orientation	
		Plage de valeurs :	0 ... 2
			L'ordre des axes d'orientation (1 ... 3) se rapporte à la définition des axes dans $NT_ROT_AX_NAME.
	Le contenu ne peut être différent de zéro que si les positions sont indexées (denture Hirth), c.-à-d. lorsque la donnée système $NT_HIRTH_INCR de l'axe concerné est différente de zéro et qu'il s'agit d'un axe rotatif manuel.		

Variable système	Signification		
$P_ORI_SOL	Lorsque, pour une transformation d'orientation avec plusieurs axes d'orientation, les angles d'axe sont calculés pour obtenir une orientation spécifiée, il existe généralement plusieurs solutions. La variable système $P_ORI_SOL fournit le nombre de solutions valides ainsi que des informations d'état supplémentaires. Le contenu de $P_ORI_SOL est codé de la manière suivante :		
	Valeurs < 0	États d'erreur généraux	
		-1	Aucune solution n'a encore été calculée pour la transformation active (appel d'ORISOLH manquant).
		-2	Aucune transformation n'est active ou la transformation active n'est pas une transformation d'orientation (transformation 6 axes) pouvant fournir des positions pour une programmation spécifiée de l'orientation.
		-4	L'orientation souhaitée n'est pas réalisable avec la cinématique donnée.
		-5	Aucune solution n'a été trouvée lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil".
		-6	Lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil", l'angle γ est trop grand.
		-7	Lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil", un angle a été spécifié qui ne peut pas être réglé en raison de la denture Hirth.
		-8	Le premier axe d'orientation (axe frame) ne doit pas être paramétré comme axe Hirth.
		-9	Le deuxième et le troisième axe rotatif sont paramétrés comme axes Hirth. Un des deux axes au maximum doit être un axe Hirth.
		-10	Aucune adaptation de la / des solution(s) à l'axe Hirth n'a été trouvée.
	Valeurs > 0 Position des unités	Nombre de solutions mathématiquement possibles sans tenir compte des limites d'axe et d'éventuelles conditions d'erreur.	
		0	Il n'existe aucune solution, c.-à-d. l'orientation demandée ne peut être réglée. Ce cas de figure peut avoir trois causes différentes : • En raison de la cinématique de la machine (positionnement non orthogonal des axes d'orientation), l'orientation n'est par principe pas atteignable indépendamment de la plage de déplacement des axes d'orientation. Dans ce cas, les positions des dizaines et des centaines de $P_ORI_SOL sont égales à zéro et les variables d'état $P_ORI_STAT affectées aux axes d'orientation contiennent la valeur "-4". • Les solutions calculées ne peuvent pas être atteintes car elles dépasseraient les limitations d'axe. Les positions des axes d'orientation qui résulteraient sans limitation d'axe sont fournies dans $P_ORI_POS. • Lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil", les positions d'axe ont été spécifiées de telle sorte que le vecteur d'orientation ou le vecteur normal d'orientation de l'outil est parallèle au premier axe

Variable système	Signification		
			d'orientation dont la position doit être calculée. La position de cet axe n'est pas définie dans ces cas.
		1	Il n'existe pas de solution. Ce cas de figure peut avoir trois causes différentes : • En raison de l'orientation spécifiée et de la cinématique de la machine, il n'existe qu'une seule solution, même sans tenir compte des limitations d'axe (du point de vue mathématique, il s'agit de deux solutions identiques). Cette situation se produit, dans le cas de cinématiques non orthogonales, à la limite de la plage d'orientation. $P_ORI_POS contient les deux solutions (identiques). • Il n'existe qu'une seule solution, car la deuxième solution n'est pas valide en raison d'un dépassement des limites d'axe. La solution valide est alors toujours la première solution dans $P_ORI_POS. La deuxième solution, calculée sans prise en compte des limites d'axe, peut également être lue dans $P_ORI_POS. • C'est le cas normal lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil". Lorsque les positions d'axe sont spécifiées pour deux axes d'orientation, il n'existe généralement qu'une seule position valide pour les axes d'orientation manquants à calculer.
		2	Il existe deux solutions.
		8	Il existe un nombre infini de solutions ; autrement dit, la position d'un axe d'orientation (axe polaire) est indifférente. Concernant le ou les autre(s) axe(s), l'une des positions possibles est exclue en raison d'un dépassement des limites d'axe.
		9	Il existe un nombre infini de solutions ; autrement dit, la position d'un axe d'orientation (axe polaire) est indéfinie. La position des centaines de la variable système $P_ORI_STAT identifie l'axe indéfini.
	Valeurs > 0 Position des dizaines	Indication codée sur bits des limites d'axe dépassées. La variable système $P_ORI_STAT permet de déterminer la cause exacte de l'erreur.	
		Bit 0 (valeur 10) :	Au moins une limite d'axe du 1er axe d'orientation a été dépassée dans l'une des solutions.
		Bit 1 (valeur 20) :	Au moins une limite d'axe du 2e axe d'orientation a été dépassée dans l'une des solutions.
		Bit 2 (valeur 40) :	Au moins une limite d'axe du 3e axe d'orientation a été dépassée dans l'une des solutions.
	Valeurs > 0 Position des centaines	Indication codée sur bits des positions d'axe non définies (uniquement possible s'il existe un nombre infini de solutions, c.-à-d. si la position des unités est égale à "9").	
		Bit 0 (valeur 100) :	La position du 1er axe d'orientation n'est pas définie.
		Bit 1 (valeur 200) :	La position du 2e axe d'orientation n'est pas définie.

Variable système	Signification		
		Bit 2 (valeur 400) :	La position du 3e axe d'orientation n'est pas définie.
	Les désignations "1er, 2e et 3e axe d'orientation" se rapportent aux définitions des axes dans $NT_ROT_AX_NAME.		

Variable système	Signification			
$P_ORI_STAT [<n>]	Fournit l'état pour chacun des 3 axes d'orientation (max.) après l'appel d'ORISOLH.			
	<n> :	Indice de l'axe d'orientation (correspond à l'indice de l'axe d'orientation concerné dans $NT_ROT_AX_NAME)		
		Plage de valeurs :	0 ... 2	
			L'ordre des axes d'orientation (1 ... 3) se rapporte à la définition des axes dans $NT_ROT_AX_NAME.	
	Le contenu de $P_ORI_STAT est codé de la manière suivante :			
	Valeurs < 0	États d'erreur généraux		
		-1	L'état n'est pas défini (appel manquant d'ORISOLH).	
		-2	Aucune transformation n'est active ou la transformation active n'est pas une transformation d'orientation (transformation 6 axes) pouvant fournir des positions pour une programmation spécifiée de l'orientation.	
		-3	L'axe ne fait pas partie de la transformation active.	
		-4	Impossible de calculer la position de l'axe, car l'orientation souhaitée n'est pas réalisable avec la cinématique donnée, même en supposant une plage de déplacement quelconque.	
		-5	Lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil", les positions d'axe ont été spécifiées de telle sorte que le vecteur d'orientation ou le vecteur normal d'orientation de l'outil est parallèle au premier axe d'orientation dont la position doit être calculée. La position de cet axe n'est pas définie dans ces cas.	
		-6	Lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil", l'angle γ est trop grand.	
		-7	Lors de l'appel de la fonction ORISOLH en mode "Orienter directement l'outil", un angle a été spécifié qui ne peut pas être réglé en raison de la denture Hirth.	
		-8	Le premier axe d'orientation (axe frame) ne doit pas être paramétré comme axe Hirth.	
		-9	Le deuxième et le troisième axe rotatif sont paramétrés comme axes Hirth. Un des deux axes au maximum doit être un axe Hirth.	
		-10	Aucune adaptation de la / des solution(s) à l'axe Hirth n'a été trouvée.	
	Valeurs > 0 Position des unités	Indication codée sur bits des limites d'axe dépassées de la première solution.		
		Bit 0 (valeur 1) :	La première solution dépasse la limite d'axe inférieure.	
		Bit 1 (valeur 2) :	La première solution dépasse la limite d'axe supérieure.	
	Valeurs > 0 Position des dizaines	Indication codée sur bits des limites d'axe dépassées de la deuxième solution.		
		Bit 0 (valeur 10) :	La deuxième solution dépasse la limite d'axe inférieure.	
		Bit 1 (valeur 20) :	La deuxième solution dépasse la limite d'axe supérieure.	

Variable système	Signification		
	Valeurs > 0	Indication d'une position d'axe non définie.	
	Position des centaines	Bit 0 (valeur 100) :	La position de l'axe d'orientation n'est pas définie, c.-à-d. l'orientation demandée peut être réalisée avec tous les réglages possibles de l'axe rotatif (position polaire). Cette information est également présente dans la variable système $P_ORI_SOL.
	Plusieurs des numéros d'erreur indiquant un dépassement des limites d'axe peuvent apparaître simultanément. Lors d'un dépassement d'une limite d'axe, la fonction essaie d'atteindre une position à l'intérieur des limites d'axe autorisées en additionnant ou soustrayant des multiples de 360°. Si cela est impossible, le résultat ne permet pas de déterminer de manière univoque si la limite supérieure ou inférieure a été dépassée. Si aucune solution existe pour l'orientation demandée ($P_ORI_SOL = 0), l'état des axes d'orientation contenus dans la transformation est égal à "0".		

Remarque

$NT_ROT_AX_NAME

Cette variable système identifie un maximum de 3 axes utilisés pour le réglage de l'orientation. Elle contient les noms des éléments de chaîne ($NK_NAME) définissant les axes machine (axes rotatifs) qui doivent exécuter les déplacements d'orientation résultant d'une transformation cinématique. L'ordre des trois axes rotatifs (maximum) contenus dans la variable système est sans importance pour la cinématique de la machine, car il est dérivé de la structure des chaînes cinématiques. Toutefois, comme cet ordre définit l'ordre d'accès des autres variables aux axes rotatifs, l'ordre des axes d'orientation dans $NT_ROT_AX_NAME doit correspondre à la description de la cinématique.

Remarque

Informations d'état

Les informations d'état, signalant par ex. qu'une orientation n'est pas réalisable ou que sa réalisation entraînerait un dépassement des limites d'axe en question, ne génèrent pas d'alarme CN. Il relève de la responsabilité de l'utilisateur de réagir de manière appropriée aux conditions signalées.

11.11.2 Activation de la modification des données de correction pour outils orientables (CUTMOD, CUTMODK)

La modification des données de correction pour outils orientables est activée dans le programme CN par l'instruction CUTMOD (en association avec les organes porte-outil orientables) ou CUTMODK (pour les transformations d'orientation définies à l'aide de chaînes cinématiques).

Remarque

Comme les organes porte-outil orientables et les transformations d'orientation définies à l'aide de chaînes cinématiques ne peuvent pas être actifs simultanément, aucun conflit n'apparaît entre les deux variantes.

Syntaxe

```
CUTMOD = <Value>
```

ou

```
CUTMODK = <Command>
```

Signification

CUTMOD :	Appel de fonction dans le contexte des organes porte-outil orientables		
<Value> :	Valeur affectée		
	Type de données :	INT	
	Valeur :	0	La fonction est désactivée. Les valeurs fournies par les variables système $P_AD... sont égales aux paramètres d'outil correspondants.
		> 0	La fonction est activée si un organe porte-outil orientable portant le numéro indiqué est actif, ce qui signifie que l'activation est liée à un organe porte-outil orientable défini. Par rapport aux paramètres d'outil correspondants, les valeurs fournies par les variables système $P_AD... sont modifiées le cas échéant en fonction de la rotation active. La désactivation de l'organe porte-outil orientable désigné désactive la fonction temporairement et l'activation d'un autre organe porte-outil orientable la désactive durablement. Dans le premier cas, la fonction est donc de nouveau active si le même organe porte-outil orientable est réactivé. Dans le deuxième cas, il est nécessaire de l'activer à nouveau même si l'organe porte-outil orientable portant le numéro indiqué est réactivé ultérieurement. La fonction n'est pas influencée par un reset.
		-1	La fonction est toujours activée lorsqu'un organe porte-outil orientable est actif. Lors d'un changement de l'organe porte-outil ou lorsque celui-ci est désactivé, puis réactivé ultérieurement, il n'est pas nécessaire de réactiver CUTMOD.
		-2	La fonction est toujours activée lorsqu'un organe porte-outil orientable dont le numéro est identique à l'organe porte-outil orientable actif à l'instant est actif. Si aucun organe porte-outil orientable n'est actif, cela équivaut à CUTMOD=0. Si un organe porte-outil orientable est actif, cela équivaut à l'indication directe du numéro de l'organe porte-outil courant.
		< -2	Les valeurs inférieures à 2 sont ignorées, ce qui signifie que ce cas est traité comme si CUTMOD n'était pas programmé. **Remarque :** L'utilisation de cette plage de valeurs est déconseillée, car elle est réservée pour d'éventuelles extensions ultérieures.
CUTMODK :	Appel de fonction dans le contexte des transformations d'orientation définies à l'aide de chaînes cinématiques		

<Command> :	Ordre affecté		
	Type de données :	STRING	
	Valeur :	"NEW"	Seuls les états pertinents pour la "Modification des données de correction" d'une transformation active définie à l'aide de chaînes cinématiques, de même que le nom de la transformation et le frame de contour sont enregistrés. **Remarque :** Cet ordre n'est autorisé que si une transformation appropriée (TRAORI_DYN, TRAORI_STAT ou TRAANG_K) est active.
		"OFF"	Désactive la "Modification des données de correction" active. Les données précédemment enregistrées avec "NEW" sont conservées. **Remarque :** Cet ordre n'est autorisé que si CUTMODK n'est pas active. Il reste alors sans effet. Un jeu de paramètres éventuellement existant pour la "Modification des données de correction" est conservé.
		"ON"	Avec cet ordre, la "Modification des données de correction" est de nouveau activée avec un jeu de paramètres enregistré avec l'ordre "NEW". Lors de l'exécution de cet ordre, si une transformation du même nom que le jeu de paramètre enregistré est active, la "Modification des données de correction" est immédiatement effective. Sinon l'activation est retardée jusqu'à ce qu'une transformation active soit activée.
		"CLEAR"	Désactive, tout comme l'ordre "OFF", la "Modification des données de correction" et supprime en outre le jeu de paramètres enregistré. **Remarque :** Cet ordre n'est autorisé que si CUTMODK n'est pas active.

Remarque

SD42984 $SC_CUTDIRMOD

La fonction qui est activable avec l'instruction CUTMOD ou CUTMODK remplace la fonction activable au moyen de la donnée de réglage SD42984 $SC_CUTDIRMOD. Cependant la disponibilité de cette fonction reste inchangée. Toutefois, l'utilisation parallèle des deux fonctions étant déconseillée, elle ne peut être activée que lorsque CUTMOD est égal à zéro et CUTMODK égal à la chaîne vide.

Informations complémentaires

Lecture des données de correction modifiées

Les données de correction modifiées sont disponibles dans les variables système et variables OPI suivantes :

Signification	Variable système	Variable BTSS
Position du tranchant	$P_AD[2]	cuttEdgeParam2
Angle du porte-outil	$P_AD[10]	cuttEdgeParam10
Sens de coupe	$P_AD[11]	cuttEdgeParam11
Angle de dépouille	$P_AD[24]	cuttEdgeParam24

Les données sont toujours modifiées par rapport aux paramètres d'outil correspondants ($TC_DP2[..., ...], etc.) lorsque la fonction "Modification des données de correction des outils orientables" a été activée avec l'instruction `CUTMOD` ou `CUTMODK` et qu'un outil a été orienté par un organe porte-outil orientable ou une transformation d'orientation.

Autres variables système pertinentes pour la fonction

Variable système	Signification
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	Fournit l'angle de rotation d'un outil dans le plan d'usinage actif, utilisé par les fonctions CUTMOD ou CUTMODK pour la détermination des données de tranchant modifiées.
$P_CUTMOD / $AC_CUTMOD	Lecture de la valeur courante valide, qui a été programmée en dernier avec l'instruction CUTMOD (numéro de l'organe porte-outil pour lequel la modification des données de correction doit être activée). Si la dernière valeur programmée était `CUTMOD = -2` (activation avec l'organe porte-outil orientable actif à l'instant), la valeur de retour de la variable système n'est pas "-2", mais le numéro de l'organe porte-outil orientable qui était actif au moment de la programmation.
$P_CUTMODK / $AC_CUTMODK	Lit le nom de la transformation utilisé pour créer le jeu de paramètres "Modification des données de correction" actuellement valide.

Variable système	Signification	
$P_CUT_INV / $AC_CUT_INV	Livre la valeur TRUE lorsque l'outil est tourné de sorte que le sens de rotation de la broche doit être inversé. A cet effet, les quatre conditions suivantes doivent être remplies dans le bloc auquel se rapporte l'opération de lecture correspondante : 1. Un outil de tournage ou de rectification est actif (types d'outil 400 à 599 et/ou SD42950 $SC_TOOL_LENGTH_TYPE = 2). 2. La modification des données de correction a été activée à l'aide de l'instruction `CUTMOD` ou `CUTMODK`. 3. Un organe porte-outil orientable ou une transformation d'orientation définie à l'aide de chaînes cinématiques est actif/active, sélectionné(e) par l'instruction CUTMOD ou CUTMODK. 4. L'outil est orienté par l'organe porte-outil orientable ou par la transformation d'orientation cinématique de sorte que la normale résultante du tranchant de l'outil est tournée de plus de 90° (normalement 180°) par rapport à la position initiale. Si au moins une des quatre conditions mentionnées n'est pas remplie, la variable contient la valeur FALSE. Pour les outils dont la position du tranchant n'est pas définie, la valeur des variables est toujours FALSE.	
$P_CUTMOD_ERR	État d'erreur après le dernier appel de la fonction CUTMOD La fonction CUTMOD peut également être appelée de manière implicite lors d'un changement d'outil. La variable est réinitialisée à zéro lors d'un Reset. À chaque changement d'outil, elle est d'abord réinitialisée puis, le cas échéant, réécrite. La variable est codée par bits. Les bits ont les significations suivantes :	
	Bit 0 :	Aucun sens de coupe valide n'est défini pour l'outil actif.
	Bit 1 :	Les angles de coupe (angle de dépouille et angle de porte-outil) de l'outil actif ont la valeur zéro.
	Bit 2 :	La valeur de l'angle de dépouille de l'outil actif n'est pas autorisée (< 0° ou > 180°).
	Bit 3 :	La valeur de l'angle de porte-outil de l'outil actif n'est pas autorisée (< 0° ou > 90°).
	Bit 4 :	La valeur de l'angle de la plaquette de l'outil actif n'est pas autorisée (< 0° ou > 90°).
	Bit 5 :	La combinaison position du tranchant - angle du porte-outil de l'outil actif n'est pas autorisée (pour les positions du tranchant 1 à 4, l'angle du porte-outil doit être ≤ 90°, pour les positions du tranchant 5 bis 8 ≥ 90°).
	Bit 6 :	Rotation non autorisée de l'outil actif. L'outil a été tourné de ± 90° (avec une tolérance d'environ 1°) par rapport au plan d'usinage actif. Par conséquent, la position du tranchant dans le plan d'usinage n'est plus définie.
	Bit 7 :	La plaquette ne se trouve pas dans le plan d'usinage et l'angle entre plaquette et plan d'usinage dépasse la limite supérieure spécifiée dans la donnée de réglage SD42998 $SC_CUTMOD_PLANE_TOL.
	Bit 8 :	La plaquette ne se trouve pas dans le plan d'usinage. La valeur absolue de l'angle α est supérieure à 1°. L'angle α est l'angle de rotation autour de l'axe de coordonnée perpendiculaire à la fois à l'axe de rotation de l'angle β et à l'axe de rotation de l'angle γ (pour G18, c'est l'axe X).

$P_... : Variables de prétraitement

$AC_... : Variables d'exécution

Toutes les variables d'exécution peuvent être lues dans des actions synchrones. Un accès en lecture depuis le prétraitement entraîne un arrêt du prétraitement.

Changement de plan

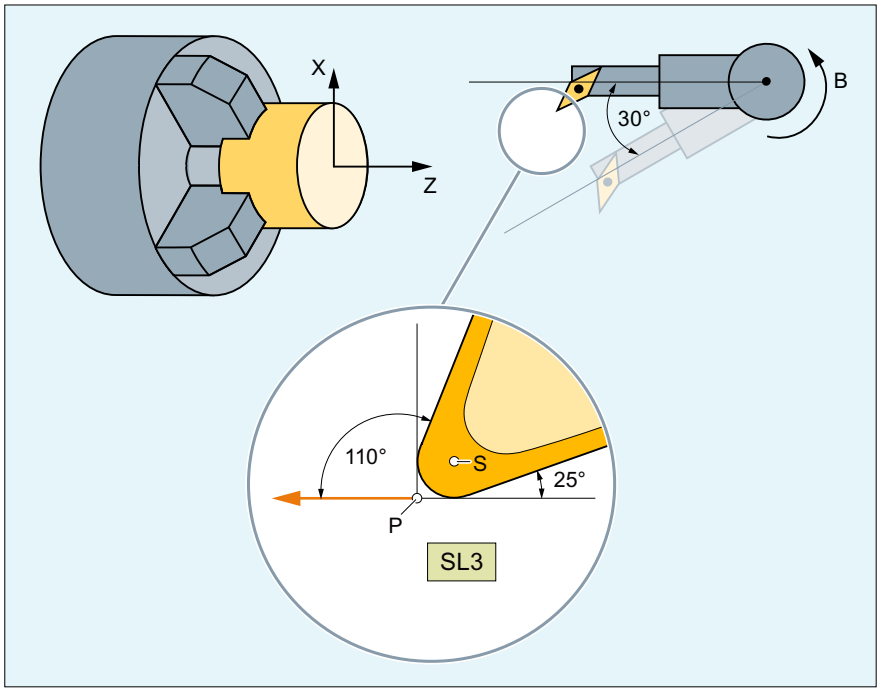
Pour la détermination de la position modifiée du tranchant, du sens de coupe et de l'angle du porte-outil ou de l'angle de dépouille, le tranchant doit être considéré dans le plan actif correspondant (G17 - G19).

Si la donnée de réglage SD42940 $SC_TOOL_LENGTH_CONST (modification des composantes de longueur d'outil en cas de changement de plan) contient cependant une valeur valide différente de zéro (plus ou moins 17, 18 ou 19), son contenu détermine le plan dans lequel les grandeurs déterminantes sont considérées.

Cette règle de priorité des données de réglage par rapport au code G peut être invalidée en activant le bit 18 du paramètre machine $MC_TOOL_PARAMETER_DEF_MASK. Autrement dit, lorsque ce bit est à 1, c'est toujours le plan défini avec l'instruction G du groupe 6 qui est valide.

Prise d'effet des données de tranchant modifiées

La position modifiée du tranchant et le point de référence modifié du tranchant prennent effet immédiatement même s'ils sont programmés pour un outil déjà actif. A cet effet, il n'est pas nécessaire de réactiver l'outil.



S : Centre du bec de l'outil

P : Point de référence du tranchant

SL : Position du tranchant

Figure 11-4 Outil avec la position de tranchant 3 et un organe porte-outil orientable pouvant tourner l'outil autour de l'axe B.

Code de programme	**Commentaire**
N10 $TC_DP1[1,1]=500	
N20 $TC_DP2[1,1]=3	; Position du tranchant
N30 $TC_DP3[1,1]=12	
N40 $TC_DP4[1,1]=1	
N50 $TC_DP6[1,1]=6	
N60 $TC_DP10[1,1]=110	; Angle du porte-outil
N70 $TC_DP11[1,1]=3	; Sens de coupe
N80 $TC_DP24[1,1]=25	; Angle de dépouille
N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0	; Axe B
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0 $TC_CARR12[2]=1	; Axe C
N110 $TC_CARR13[2]=0	
N120 $TC_CARR14[2]=0	
N130 $TC_CARR21[2]=X	
N140 $TC_CARR22[2]=X	
N150 $TC_CARR23[2]="M"	
N160 TCOABS CUTMOD=0	
N170 G18 T1 D1 TCARR=2	; X Y Z
N180 X0 Y0 Z0 F10000	; 12.000 0.000 1.000
N190 $TC_CARR13[2]=30	
N200 TCARR=2	
N210 X0 Y0 Z0	; 10.892 0.000 -5.134
N220 G42 Z-10	; 8.696 0.000 -17.330
N230 Z-20	; 8.696 0.000 -21.330
N240 X10	; 12.696 0.000 -21.330
N250 G40 X20 Z0	; 30.892 0.000 -5.134
N260 CUTMOD=2 X0 Y0 Z0	; 8.696 0.000 -7.330
N270 G42 Z-10	; 8.696 0.000 -17.330
N280 Z-20	; 8.696 0.000 -21.330
N290 X10	; 12.696 0.000 -21.330
N300 G40 X20 Z0	; 28.696 0.000 -7.330
N310 M30	

Les valeurs numériques des commentaires indiquent les positions de fin de bloc en coordonnées machine (SCM), dans l'ordre X → Y → Z.

Explications

Dans le bloc `N180`, l'outil est d'abord activé avec `CUTMOD=0` et organe porte-outil orientable non pivoté. Tous les vecteurs d'offset de l'organe porte-outil étant 0, la position accostée correspond aux longueurs d'outil indiquées dans `$TC_DP3[1,1]` et `$TC_DP4[1,1]`.

Dans le bloc `N200`, l'organe porte-outil orientable est activé avec une rotation de 30° autour de l'axe B. La position du tranchant n'étant pas modifiée compte tenu de `CUTMOD=0`, l'ancien point de référence du tranchant reste déterminant. C'est pourquoi la position qui garde l'ancien point de référence du tranchant à l'origine (ce qui signifie que le vecteur (1, 12) tourne de 30° dans le plan Z/X) est accostée dans le bloc `N210`.

Contrairement au bloc `N200`, CUTMOD=2 est actif dans le bloc `N260`. En raison de la rotation de l'organe porte-outil orientable, la position modifiée du tranchant est 8. Il en résulte également des positions d'axe différentes.

Dans les blocs `N220` et `N270`, la correction du rayon d'outil (CRO) est activée. La position du tranchant, qui diffère dans les deux sections de programme, n'influence pas les positions finales des blocs dans lesquels la correction du rayon d'outil (CRO) est active. Les positions correspondantes sont donc identiques. Ce n'est que dans les blocs de désactivation `N260` et `N300` que les différentes positions du tranchant ont de nouveau un effet.

11.12 Utilisation des environnements d'outils

Vue d'ensemble des fonctions

- Enregistrement de l'environnement d'outil (TOOLENV) (Page 501)
- Suppression de l'environnement d'outil (DELTOOLENV) (Page 504)
- Lecture des numéros T, D et DL (GETTENV) (Page 505)
- Lecture de longueurs d'outil ou de composantes de longueur d'outil (GETTCOR) (Page 507)
- Modification des composantes d'outil (SETTCOR) (Page 514)

Vue d'ensemble des variables système

- Lecture des informations relatives aux environnements d'outil archivés ($P_TOOLENVN, ($P_TOOLENV) (Page 506)

11.12.1 Enregistrement de l'environnement d'outil (TOOLENV)

La fonction TOOLENV sert à enregistrer tous les états actuels significatifs pour l'exploitation des données d'outil stockées dans la mémoire.

Dans le détail, il s'agit des données suivantes :

- L'instruction G active du groupe :
  - 6 (G17, G18, G19)
  - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- L'axe transversal actif :
- Les paramètres machine :
  - PM18112 $MN_MM_KIND_OF_SUMCORR (caractéristiques des corrections totales dans la zone TO)
  - PM20360 $MC_TOOL_PARAMETER_DEF_MASK (définition des paramètres d'outil)

- Les données de réglage :
  - SD42900 $SC_MIRROR_TOOL_LENGTH (inversion du signe de la longueur d'outil pour la fonction miroir)
  - SD42910 $SC_MIRROR_TOOL_WEAR (inversion du signe de l'usure d'outil pour la fonction miroir)
  - SD42920 $SC_WEAR_SIGN_CUTPOS (signe de l'usure pour les outils avec dispositifs de coupe)
  - SD42930 $SC_WEAR_SIGN (signe de l'usure)
  - SD42935 $SC_WEAR_TRANSFORM (transformations pour composantes d'outil)
  - SD42940 $SC_LENGTH_CONST (changement des composantes de longueur d'outil lors d'un changement de plan)
  - SD42942 $SC_TOOL_LENGTH_CONST_T (Changement des composantes de longueur d'outil pour outils de tournage lors d'un changement de plan)
  - SD42950 $SC_TOOL_LENGTH_TYPE (affectation des composantes de longueur d'outil indépendamment du type d'outil)
  - SD42954 $SC_TOOL_ORI_CONST_M (Changement des composantes d'orientation de l'outil pour outils de fraisage lors d'un changement de plan)
  - SD42956 $SC_TOOL_ORI_CONST_T (Changement des composantes d'orientation de l'outil pour outils de tournage lors d'un changement de plan)
- La composante d'orientation du frame global actuel (rotations et fonctions miroir, pas de décalages d'origine ni d'homothéties)
- La composante d'orientation et la longueur résultante de l'organe porte-outil orientable actif
- La composante d'orientation et la longueur résultante d'une transformation active

Outre les données citées qui décrivent l'environnement de l'outil, les numéros T, D et DL de l'outil actif sont également enregistrés afin de pouvoir accéder ultérieurement à cet outil dans le même environnement comme lors de l'appel de TOOLENV, sans qu'il soit nécessaire de désigner à nouveau l'outil.

Syntaxe

```
<Status> = TOOLENV(<Name>)
```

Signification

`TOOLENV(...)` :	Fonction prédéfinie pour l'enregistrement d'un environnement d'outil	
	Seule dans le bloc :	oui

<Status> :	Valeur de retour de la fonction. Les valeurs négatives indiquent des états d'erreur.		
	Type de données :		INT
	Valeur :	0	Fonction OK
		-1	Pas d'espace mémoire réservé pour les environnements d'outil : PM18116 $MN_MM_NUM_TOOL_ENV = 0 C'est-à-dire que la fonctionnalité "Environnements d'outil" n'est pas disponible.
		-2	Plus aucun espace mémoire libre disponible pour les environnements d'outil.
		-3	Une chaîne vide en tant que nom d'environnement d'outil est non admissible.
		-4	Aucun paramètre (<Name>) spécifié.
Paramètres			
1 <Name> :	Nom sous lequel le jeu de paramètres actuel doit être enregistré Si un jeu de paramètres du même nom existe déjà, celui-ci est écrasé. Dans ce cas, le statut est "0".		
	Type de données :		STRING

Informations complémentaires

Cote de base / cote d'adaptation - Correction de longueur d'outil

Si la gestion de magasin d'outils est active (seulement disponible avec l'option "Gestion des outils"), c'est la valeur du paramètre machine qui décide si oui ou non la longueur de l'adaptateur ou la cote de base (paramètre spécifique au tranchant $TC_DP21, $TC_DP22 et $TC_DP23) doit entrer dans le calcul de la longueur d'outil :

PM18104 $MN_MM_NUM_TOOL_ADAPTER (adaptateur d'outils dans la zone TO)

Compte tenu qu'une modification de ce paramètre machine peut seulement devenir effective au démarrage de la commande, il n'est pas enregistré dans l'environnement d'outil.

Longueur résultante des porte-outils orientables et des transformations

Remarque

Tant pour les porte-outils orientables que pour les transformations, il existe des variables système ou des paramètres machine qui agissent comme des composantes supplémentaires de longueur d'outil et qui sont assujettis en tout ou partie aux rotations exécutables par celles-ci. Les composantes supplémentaires de longueur d'outil qui en résultent doivent être enregistrées lors de l'appel de TOOLENV compte tenu qu'elles constituent une partie de l'environnement dans lequel l'outil est utilisé.

Transformation d'adaptateur

La transformation d'adaptateur est une caractéristique de l'adaptateur d'outil et de ce fait de l'outil complet. C'est pourquoi elle ne fait pas partie intégrante d'un environnement d'outil qui peut être appliqué à un autre outil.

L'enregistrement de l'ensemble des données requises pour la détermination de la longueur totale de l'outil permet de calculer à un moment ultérieur la longueur effective d'un outil, même si celui-ci n'est plus actif à ce moment-là ou si les conditions ambiantes (par ex. instructions G ou données de réglage) ont changé. De même, la longueur effective d'un autre outil peut être calculée en supposant que celui-ci a été utilisé dans les mêmes conditions que l'outil pour lequel l'état a été enregistré.

Nombre maximal de jeux de paramètres pour les environnements d'outil

Le paramètre machine PM18116 $MN_MM_NUM_TOOL_ENV permet de déterminer le nombre maximal de jeux de paramètres pouvant être stockés pour la description d'environnements d'outil. Les données se trouvent dans la zone TOA. Elles sont conservées même après la mise hors tension de la commande.

La sauvegarde des données est impossible. En d'autres termes, ces données ne peuvent pas être transmises entre différentes commandes.

11.12.2 Suppression de l'environnement d'outil (DELTOOLENV)

La fonction DELTOOLENV permet de supprimer des jeux de paramètres pour la description d'environnements d'outil La suppression signifie que le jeu de paramètres enregistré sous un nom défini n'est plus accessible (toute tentative d'accès génère une alarme).

Remarque

Les jeux de paramètres peuvent seulement être supprimés avec la fonction DELTOOLENV, via un download INITIAL.INI ou un démarrage à froid (démarrage de la CN avec paramètres machine standard). Il n'existe aucune autre procédure de suppression automatique.

Syntaxe

```
<Status> = DELTOOLENV(<Name>)
<Status> = DELTOOLENV()
```

Signification

DELTOOLENV(...) :	Fonction prédéfinie pour la suppression d'un environnement d'outil	
	Seule dans le bloc :	oui

`<Status>` :	Valeur de retour de la fonction. Les valeurs négatives indiquent des états d'erreur.		
	Type de données :		INT
	Valeur :	0	Fonction OK
		-1	Pas d'espace mémoire réservé pour les environnements d'outil : PM18116 $MN_MM_NUM_TOOL_ENV = 0 C'est-à-dire que la fonctionnalité "Environnements d'outil" n'est pas disponible.
		-2	Il n'existe pas d'environnement d'outil avec le nom spécifié.
Paramètres			
1 `<Name>` :	Nom du jeu de paramètres à supprimer		
	Type de données :		STRING
`DELTOOLENV()` :	`DELTOOLENV()` sans spécification de nom supprime tous les jeux de paramètres relatifs à la description des environnements d'outil		

11.12.3 Lecture des numéros T, D et DL (GETTENV)

La fonction GETTENV sert à lire les numéros T, D et DL enregistrés dans un environnement d'outil

Syntaxe

```
<Status> = GETTENV(<Name>, <TDDL>)
```

Signification

`GETTENV(...)` :	Fonction prédéfinie pour la lecture de numéros T, D et DL dans un jeu de paramètres relatif à la description d'un environnement d'outil		
	Seule dans le bloc :		oui
`<Status>` :	Valeur de retour de la fonction. Les valeurs négatives indiquent des états d'erreur.		
	Type de données :		INT
	Valeur :	0	Fonction OK
		-1	Pas d'espace mémoire réservé pour les environnements d'outil : PM18116 $MN_MM_NUM_TOOL_ENV = 0 C'est-à-dire que la fonctionnalité "Environnements d'outil" n'est pas disponible.
		-2	Il n'existe pas d'environnement d'outil avec le nom spécifié.
Paramètres			

1	<Name> :	Nom du jeu de paramètres à partir duquel doivent être lus les numéros T, D et DL	
		Type de données :	STRING
2	<TDDL> :	Le champ de ce paramètre de résultat contient les numéros T, D et DL de l'outil dont l'environnement est enregistré dans le jeu de paramètres spécifié : • <TDDL> [0] : Numéro T • <TDDL> [1] : Numéro D • <TDDL> [2] : numéro DL	
		Type de données :	INT[3]
GETTENV(,<TDDL>), GETTENV("",<TDDL>) :		Lors de l'appel de la fonction GETTENV, l'omission du premier paramètre ou le transfert de la chaîne vide en tant que premier paramètre est autorisé(e). Dans ces deux cas particuliers, les numéros T, D et DL de l'outil **actif** sont retournés dans <TDDL>.	

11.12.4 Lecture des informations relatives aux environnements d'outil archivés ($P_TOOLENVN, ($P_TOOLENV)

Les informations relatives aux environnements d'outil archivés peuvent être lues dans les variables système suivantes :

$P_TOOLENVN :	Fournit le nombre de jeux de paramètres définis (et pas encore supprimés) à l'aide de TOOLENV pour la description d'environnements d'outil			
	Syntaxe :	<n> = $P_TOOLENVN		
	Signification :	<n> :	Nombre de jeux de paramètres définis.	
			Type de données :	INT
			Plage de valeurs :	0 ... PM18116 $MN_MM_NUM_TOOL_ENV
	L'accès à cette variable système est également autorisé lorsqu'aucun environnement d'outil n'est possible (PM18116 = 0). Dans ce cas, la valeur de retour est "0".			

`$P_TOOLENV` :	Fournit le nom du <i>-ième jeu de paramètres pour la description d'un environnement d'outil			
	Syntaxe :	`<Name> = $P_TOOLENV[<i>]`		
	Signification :	`<Name>` :	Nom du jeu de paramètres avec le numéro <i>	
			Type de données :	STRING
		`<i>` :	Numéro du jeu de paramètres	
			Type de données :	INT
			Plage de valeurs :	1 ... $P_TOOLENVN
	L'affectation des numéros aux jeux de paramètres n'est pas fixe mais peut au contraire être modifiée en supprimant ou en créant de nouveaux jeux de paramètres. La numérotation des jeux de paramètres s'effectue en interne. Si <i> renvoie à un jeu de paramètres non défini, la chaîne vide est retournée. Si l'indice <i> est non valide, c'est-à-dire que si <i> est inférieur à 1 ou supérieur au nombre maximal de jeux de paramètres pour les environnements d'outil (PM18116 $MN_MM_NUM_TOOLENV), l'alarme suivante est émise : Alarme 17020 "Indice1 de tableau non autorisé"			

11.12.5 Lecture de longueurs d'outil ou de composantes de longueur d'outil (GETTCOR)

La fonction GETTCOR sert à lire les longueurs d'outil ou les composantes de longueur d'outil.

Le paramétrage permet de spécifier les composantes à prendre en compte et les conditions d'utilisation dans lesquelles l'outil doit être pris en considération.

Syntaxe

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

Signification

`GETTCOR(...)` :	Fonction prédéfinie pour la lecture de longueurs d'outil ou de composantes de longueur d'outil	
	Seule dans le bloc :	oui

<Status> :	Valeur de retour de la fonction. Les valeurs négatives indiquent des états d'erreur.		
	Type de données :		INT
	Valeur :	0	Fonction OK
		-1	Pas d'espace mémoire réservé pour les environnements d'outil : PM18116 $MN_MM_NUM_TOOL_ENV = 0 C'est-à-dire que la fonctionnalité "Environnements d'outil" n'est pas disponible.
		-2	Il n'existe pas d'environnement d'outil avec le nom indiqué sous <Stat>.
		-3	Chaîne non valide dans le paramètre <Comp>. Les causes de cette erreur peuvent être des caractères non valides ou programmés en double.
		-4	Numéro T non valide
		-5	Numéro D non valide
		-6	Numéro DL non valide
		-7	Tentative d'accès à un module mémoire non disponible.
		-8	Tentative d'accès à une option non disponible (par ex. orientation d'outil programmable, gestion d'outil).
		-9	La chaîne <Comp> contient un deux-points (identificateur pour les spécifications d'un système de coordonnées), toutefois aucun caractère valide ne suit pour la désignation du système de coordonnées requis.
Paramètres			

1	<Len> :	Vecteur résultat	
		Type de données :	REAL[11]
		Les composantes du vecteur sont classées dans l'ordre suivant : • <Len> [0] : Type d'outil • <Len> [1] : Position du tranchant • <Len> [2] : Abscisse • <Len> [3] : Ordonnée • <Len> [4] : Cote • <Len> [5] : Rayon de l'outil Le système de coordonnées défini dans <Comp> et <Stat> est considéré comme système de coordonnées de référence pour les composantes de longueur. Si aucun système de coordonnées n'est défini dans <Comp>, les longueurs d'outil sont représentées dans le système de coordonnées machine. L'affectation de l'abscisse, de l'ordonnée et de la cote relatives aux axes géométriques dépend du niveau actif dans l'environnement d'outil utilisé. C'est-à-dire que pour G17 l'abscisse est parallèle à X, pour G18 elle est parallèle à Z, etc. Les composantes <Len>[6] à <Len>[10] contiennent les paramètres supplémentaires pouvant être spécifiés pour la description géométrique d'un outil (par ex. $TC_DP7 à $TC_DP11 pour la géométrie ou les composantes correspondantes pour usure ou corrections de somme et de paramétrage). Ces 5 éléments supplémentaires et le rayon d'outil sont seulement définis pour les composantes E, G, S et W. Leur évaluation ne dépend pas de <Stat>. C'est pourquoi les valeurs correspondantes dans <Len>[6] à <Len>[10] ne peuvent être différentes de zéro que si au moins une des quatre composantes citées entre dans le calcul de la longueur d'outil. Les autres composantes n'ont aucune incidence sur le résultat. Les indications de cotes se rapportent au système de base de la commande (anglo-saxon ou métrique).	

2	<Comp> :	Composantes de longueur d'outil (facultatif)			
		Type de données :	STRING		
		La chaîne de caractères est constituée de deux chaînes partielles séparées l'une de l'autre par un deux-points. Forme gén. : "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1> :	La **première chaîne partielle** désigne les composantes de longueur d'outil qui doivent être prises en compte lors du calcul de longueur d'outil. L'ordre des caractères dans la chaîne partielle ainsi que leur casse (majuscules ou minuscules) est quelconque. Autant d'espaces ou tabulations (white spaces) que nécessaire peuvent être insérés entre les caractères. **Remarque :** Les caractères de la chaîne partielle **ne doivent pas** être programmés en double !		
			Symbole :	–	Signe moins (seulement admissible en tant que premier caractère !) La longueur totale d'outil est calculée, après soustraction des composantes spécifiées dans la chaîne suivante.
				C	Cote d'adaptation ou cote de base (celle des deux variantes de composantes qui est active pour l'outil utilisé)
				E	Corrections de réglage
				G	Géométrie
				K	Transformation cinématique (seulement exploitée en cas de transformation générique à 3, 4 et 5 axes !)
				S	Corrections totales
				T	Porte-outil orientable
				W	Usure
			Si la première chaîne partielle (à l'exception des espaces) est vide, cela signifie que la longueur totale d'outil doit être calculée en tenant compte de toutes les composantes. Il en va de même lorsque le paramètre <Comp> n'est pas spécifié.		
		<Substr_2> :	La **deuxième chaîne partielle** facultative désigne le système de coordonnées dans lequel la longueur d'outil doit être délivrée. La deuxième chaîne partielle est constituée d'un seul et unique caractère pertinent.		
			Symbole :	A	Système de coordonnées réglable (SCR)
				B	Système de coordonnées de base (SCB)
				K	Système de coordonnées de l'outil de la transformation cinématique (SCT)
				M	Système de coordonnées machine (SCM)
				T	Système de coordonnées de l'outil (SCO)
				W	Système de coordonnées pièce (SCP)
			Si aucun système de coordonnées n'est spécifié, l'évaluation a lieu dans le système de coordonnées machine (SCM). Les ro-		

			tations à prendre en compte le cas échéant sont déterminées par l'environnement d'outil défini dans <Stat>.
3	<_Stat> :	Nom du jeu de paramètres pour la description d'un environnement d'outil (facultatif)	
		Type de données :	STRING
		Si la valeur de ce paramètre est la chaîne vide ("") ou si elle n'est pas spécifiée, l'état actuel est utilisé. Si aucun outil n'est spécifié, l'outil actuel est utilisé.	
4	<T> :	Numéro T interne de l'outil (facultatif)	
		Type de données :	INT
		Si ce paramètre n'est pas spécifié ou si sa valeur est "0", l'outil mémorisé dans <Stat> est utilisé. Si la valeur de ce paramètre est "-1", le numéro T de l'outil actif est utilisé. La spécification explicite du numéro de l'outil actif est également admise. **Remarque :** Si <Stat> n'est pas spécifié, l'état actuel est utilisé comme environnement d'outil. Compte tenu qu'avec <T> = 0, le renvoi a lieu au numéro T enregistré dans l'environnement d'outil, c'est l'outil actif qui est utilisé dans celui-ci, c'est-à-dire que les indications <T> = 0 et <T> = -1 ont la même signification dans ce cas particulier.	
5	<D> :	Numéro de tranchant de l'outil (facultatif)	
		Type de données :	INT
		Si ce paramètre n'est pas spécifié ou si sa valeur est "0", le numéro D utilisé dépend de la source du numéro T. Si le numéro T est utilisé à partir de l'environnement d'outil, le numéro D de l'environnement d'outil est lu également, autrement c'est celui de l'outil actuellement actif.	
6	<DL> :	Numéro de la correction en fonction du lieu (facultatif)	
		Type de données :	INT
		Si ce paramètre n'est pas spécifié, le numéro DL utilisé dépend de la source du numéro T. Si le numéro T est utilisé à partir de l'environnement d'outil, le numéro D de l'environnement d'outil est lu également, autrement c'est celui de l'outil actuellement actif.	

Exemples

`GETTCOR(_LEN)`	Calcule la longueur de l'outil actuellement actif dans le système de coordonnées machine en tenant compte de toutes les composantes.
`GETTCOR(_LEN,"CGW:W")`	La longueur d'outil constituée de la cote d'adaptation ou de base, de la géométrie et de l'usure est calculée pour l'outil actif. D'autres constituantes telles que par ex. le porte-outil orientable ou les transformations cinématiques ne sont pas prises en compte. La valeur est toujours donnée dans le système de coordonnées pièce.

`GETTCOR (_LEN,"-K:B")`	La longueur totale de l'outil actif sans prise en compte des composantes de longueur d'une transformation cinématique éventuellement active est calculée. La valeur est donnée dans le système de coordonnées de base.
`GETTCOR (_LEN,":M","Testenv1",,3)`	La longueur totale de l'outil archivé pour l'environnement d'outil sous le nom "Testenv1" est calculée dans le système de coordonnées machine. Le calcul s'effectue toutefois indépendamment du numéro de tranchant mémorisé pour le numéro de tranchant D3.

plus d'informations

Transformation d'adaptateur / porte-outil orientable / transformation cinématique

Les rotations et inversions de composantes qui ont été réalisées le cas échéant par la transformation d'adaptateur, le porte-outil orientable et une transformation cinématique, font partie intégrante de l'environnement d'outil. C'est pourquoi elles sont toujours effectuées, même si la composante de longueur correspondante ne doit pas être prise en compte. Si celles-ci ne sont pas souhaitées, des environnements d'outil pour lesquels les transformations correspondantes ne sont pas actives, doivent être définis. Dans de nombreux cas (notamment lorsqu'aucune transformation ni porte-outil orientable n'est utilisé sur une machine), les jeux de paramètres archivés pour les environnements d'outil remplissent ces conditions automatiquement. Ainsi l'utilisateur n'a pas besoin de les prendre spécialement en compte.

Outils de tournage et de rectification : Calcul de la longueur d'outil en fonction de PM20360 $MC_TOOL_PARAMETER_DEF_MASK

La façon dont l'usure ou la longueur d'outil sur les outils de tournage et de rectification doit être évaluée dans un axe diamétral éventuellement présent est déterminée dans le paramètre machine :

PM20360 $MC_TOOL_PARAMETER_DEF_MASK (définition des paramètres d'outil)

Bit	Valeur	
0	Sur les outils de tournage et de rectification, le **paramètre d'usure** de l'axe transversal est pris en compte en tant que valeur de diamètre :	
	= 0 (par défaut)	non
	= 1	oui
1	Sur les outils de tournage et de rectification, la **composante de longueur d'outil** de l'axe transversal est prise en compte en tant que valeur de diamètre :	
	= 0 (par défaut)	non
	= 1	oui

Si les bits concernés sont mis à 1, l'entrée associée est évaluée avec le facteur 0,5. Cette évaluation influence également le calcul de la longueur d'outil fournie par GETTCOR.

Exemple :

PM20360 $MC_TOOL_PARAMETER_DEF_MASK = 3

PM20100 $MC_DIAMETER_AX_DEF (axe géométrique avec fonction d'axe transversal) = "X"

X est l'axe diamétral (configuration de tour standard)

```
Code de programme                    Commentaire
N30 $TC_DP1[1,1]=500
N40 $TC_DP2[1,1]=2
N50 $TC_DP3[1,1]=3,0                 ; Géométrie L1
N60 $TC_DP4[1,1]=4,0
N70 $TC_DP5[1,1]=5,0
N80 $TC_DP12[1,1]=12,0               ; Usure L1
N90 $TC_DP13[1,1]=13,0
N100 $TC_DP14[1,1]=14,0
N110 T1 D1 G18
N120 R1=GETTCOR(_LEN,"GW")
N130 R3=_LEN[2]                      ; 17,0 (= 4,0 + 13,0)
N140 R4=_LEN[3]                      ; 7,5 (= 0,5 * 3,0 + 0,5 * 12,0)
N150 R5=_LEN[4]                      ; 19,0 (= 5,0 + 14,0)
N160 M30
```

Composantes de longueur de la transformation cinématique et du porte-outil orientable

Si un **porte-outil orientable** est pris en compte pour le calcul de la longueur d'outil, les vecteurs suivants entrent dans le calcul de la longueur d'outil :

Type	Vecteurs
M	l1 et l2
T	l1, l2 et l3
P	La longueur d'outil n'est pas influencée par le porte-outil orientable.

Dans le cas de la **transformation 5 axes** générique, les paramètres machine suivants entrent dans le calcul de la longueur d'outil pour les types de transformation 24 et 56 :

Type de transformation	Paramètres machine
24	PM24550/24650 $MC_TRAFO5_BASE_TOOL_1/2 PM24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 PM24558/24658 $MC_TRAFO5_PART_OFFSET_1/2
56	PM24550/24650 $MC_TRAFO5_BASE_TOOL_1/2 PM24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2

Le type de transformation 56 (outil en mouvement et pièce en mouvement) correspond au type M pour le porte-outil orientable.

Pour cette transformation 5 axes, le vecteur PM24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 (vecteur du décalage cinématique de la 1e/2e transformation 5 axes dans le canal) correspond, dans les versions logicielles antérieures, à la somme des deux vecteurs l_1 et l_3 pour le porte-outil orientable de type M.

Dans les deux cas, seules la somme est pertinente pour la transformation. Le type de composition issu des deux composantes individuelles est sans importance. Lors du calcul de la longueur d'outil il importe toutefois de connaître la part à affecter à l'outil et la part à affecter

à la table de travail. C'est pourquoi le paramètre machine PM24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 (vecteur du décalage cinématique dans la table) a été introduit. Il correspond au vecteur l3. Le paramètre machine PM24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 ne correspond alors plus à la somme de l1 et l3, mais seulement au vecteur l1. Si le paramètre machine PM24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 est égal à zéro, le comportement est identique au comportement précédent.

Compatibilité

Avec les fonctions TOOLENV et SETTCOR entre autres, la fonction GETTCOR sert à remplacer des parties de la fonctionnalité qui étaient réalisées jusqu'ici de manière externe dans les cycles de mesure.

Dans ces derniers, seule une partie des paramètres déterminant en fin de compte la longueur effective de l'outil a été analysée. Les fonctions citées peuvent être paramétrées de manière que le comportement des cycles de mesure concernant le calcul de la longueur d'outil soit reproductible.

11.12.6 Modification des composantes d'outil (SETTCOR)

La fonction SETTCOR sert à modifier les composantes d'outil en tenant compte de toutes les autres conditions influant sur l'évaluation des différentes composantes.

Remarque

Concernant la terminologie : Dans les chapitres suivants, lorsqu'il est question de composantes d'outil en rapport avec la longueur d'outil, il s'agit des composantes prises en compte sous la forme vectorielle, qui composent la longueur totale de l'outil (par ex. géométrie ou usure). C'est pourquoi untel telle composante est constituée de trois valeurs individuelles (L1, L2, L3) désignées ci-après comme valeurs de coordonnées.

La composante d'outil "Géométrie" est constituée par ex. par conséquent de trois valeurs de coordonnées $TC_DP3 à $TC_DP5.

Syntaxe

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>, <GeoAx>,
<Stat>, <T>, <D>, <DL>])
```

Signification

`SETTCOR(...)` :	Fonction prédéfinie pour la modification de composantes d'outil	
	Seule dans le bloc :	oui

<Status> :	Valeur de retour de la fonction. Les valeurs négatives indiquent des états d'erreur.		
	Type de données :		INT
	Valeur :	0	Fonction OK
		-1	Pas d'espace mémoire réservé pour les environnements d'outil : PM18116 $MN_MM_NUM_TOOL_ENV = 0 C'est-à-dire que la fonctionnalité "Environnements d'outil" n'est pas disponible.
		-2	Il n'existe pas d'environnement d'outil avec le nom indiqué sous <Stat>.
		-3	Chaîne non valide dans le paramètre <Comp>. Les causes de cette erreur peuvent être des caractères non valides ou programmés en double.
		-4	Numéro T non valide.
		-5	Numéro D non valide.
		-6	Numéro DL non valide.
		-7	Tentative d'accès à un module mémoire non disponible.
		-8	Tentative d'accès à une option non disponible (par ex. orientation d'outil programmable, gestion d'outil).
		-9	Valeur chiffrée non admissible pour le paramètre <CorComp>.
		-10	Valeur chiffrée non admissible pour le paramètre <CorMode>.
		-11	Les contenus des paramètres <Comp> et <CorComp> sont contradictoires.
		-12	Les contenus des paramètres <Comp> et <CorMode> sont contradictoires.
		-13	Le contenu du paramètre <GeoAx> ne désigne aucun axe géométrique.
		-14	Tentative d'écriture sur une correction de réglage non disponible.
Paramètres			
1	<CorVal> :	Vecteur de correction L'affectation suivante est valable dans le système de coordonnées pièce (SCP) défini par <Stat> : • <CorVal> [0] : abscisse • <CorVal> [1] : ordonnée • <CorVal> [2] : cote Si une seule composante d'outil doit être corrigée (c'est-à-dire aucune correction vectorielle, voir paramètre <CorMode>), la valeur de correction se trouve toujours dans <CorVal>[0], quel que soit l'axe sur lequel elle agit. Le contenu des deux autres composantes n'est alors pas évalué. Si <CorVal> ou une composante de <CorVal> se rapporte à l'axe transversal, l'indication est évaluée en tant que **rayon**. Cela signifie qu'un outil par ex. "s'allonge" de la cote indiquée, ce qui entraîne par conséquent une modification deux fois plus grande du diamètre de la pièce. Les indications de cotes se rapportent au **système de base** (anglo-saxon ou métrique) de la commande.		
		Type de données :	REAL[3]

2	<Comp> :	Composante(s) d'outil			
		Type de données :	STRING		
		La chaîne de caractères est constituée de deux chaînes partielles séparées l'une de l'autre par un deux-points. Forme gén. : "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1> :	La **première chaîne partielle** doit toujours être disponible et peut être constituée d'un ou de deux caractères. À cet effet, le premier ou l'unique caractère est présent pour la 1re composante (Val_1) et le deuxième caractère pour la 2e composante (Val_2), qui sont éditées en fonction des paramètres <CorComp> et <CorMode> suivants.		
			Symbole :	C	Cote d'adaptation ou cote de base (celle des deux variantes de composantes qui est active pour l'outil utilisé)
				E	Corrections de réglage
				G	Géométrie
				S	Corrections totales
				W	Usure
		<Substr_2> :	La **deuxième chaîne partielle** est facultative. Elle peut être constituée alternativement de la lettre (unique) "W" ou "T".		
			Symbole :	W	Si la deuxième chaîne partielle est vide ou contient la lettre "W", les valeurs de correction sont calculées comme si elles étaient mesurées dans le système de coordonnées **pièce** (SCP).
				T	Si la deuxième chaîne partielle contient la lettre "T", les valeurs de correction sont calculées comme si elles étaient mesurées dans le systèmes de coordonnées **outil** (**T**ool Coordinate System, TCS).
		La casse des caractères dans la chaîne (majuscules ou minuscules) est sans importance. Autant d'espaces ou tabulations (white spaces) que nécessaire peuvent être insérés entre les caractères.			

3	<CorComp> :	Spécifie la ou les composant(e)s du jeu de paramètres d'outil qui doivent être décrites (facultatif)		
		Type de données :		INT
		Valeur :	0	La valeur de correction <CorVal>[0] se rapporte à l'axe géométrique transmis dans le paramètre <GeoAx> dans le système de coordonnées pièce ou dans le système de coordonnées outil (voir à ce sujet la description du paramètre <Comp>). C'est-à-dire que la valeur de correction doit entrer dans le calcul des composantes d'outil désignées de manière que, en tenant compte de tous les paramètres qui peuvent avoir une influence sur le calcul de la longueur d'outil, il en résulte une modification de la longueur totale de l'outil dans le sens de l'axe donné et avec la valeur prescrite. Cette modification doit être obtenue en corrigeant la composante indiquée sous <Comp> et la règle opératoire symbolique spécifiée sous <CorMode> (voir paramètre suivant). C'est pourquoi la correction qui en résulte peut avoir un effet sur les trois composantes d'axe.
			1	Comme "0" mais vectorielle. Le contenu du vecteur <CorVal> se rapporte aux abscisses, aux ordonnées et aux cotes dans le système de coordonnées pièce ou outil (voir à ce sujet la description du paramètre <Comp>). Le paramètre suivant <GeoAx> n'est pas évalué.
			2	Les corrections vectorielles, c'est-à-dire L1, L2 et L3 peuvent être modifiées simultanément. Contrairement aux variantes "0" et "1", les valeurs de correction contenues dans <CorVal> se rapportent toutefois aux coordonnées de la composante Val_1 (voir le paramètre suivant <CorMode>) de l'outil. Une inclinaison éventuellement disponible de l'outil par rapport au système de coordonnées pièce n'a aucune influence sur la correction.
			3 - 5	Correction de la longueur d'outil L1 à L3 ($TC_DP3 à $TC_DP5) ou des valeurs correspondantes en cas d'usure, de corrections de réglage ou de somme. La valeur de correction est contenue dans <CorVal>[0]. Elle est mesurée dans les coordonnées de la composante Val_1 (voir le paramètre suivant <CorMode>) de l'outil. Une inclinaison éventuellement disponible de l'outil par rapport au système de coordonnées pièce n'a aucune influence sur la correction.
			6	Correction du rayon d'outil ($TC_DP6) ou des valeurs correspondantes en cas d'usure, de corrections de réglage ou de somme. Les bits 10 et 11 (évaluation de la spécification de diamètre ou d'usure de diamètre en tant qu'indication de rayon ou de diamètre) du paramètre machine PM20360 $MC_TOOL_PARAMETER_DEF_MASK sont pris en compte.
			7 – 11	Correction de $TC_DP7 à $TC_DP11 ou des valeurs correspondantes en cas d'usure, de corrections de réglage ou de somme. Ces paramètres sont traités comme le rayon d'outil.
		Si ce paramètre n'est pas spécifié, sa valeur est "0".		

<table>
<tr><td>4</td><td><CorMode> :</td><td colspan="3">Spécifie le type d'opération d'écriture à exécuter (facultatif)</td></tr>
<tr><td></td><td></td><td colspan="2">Type de données :</td><td>INT</td></tr>
<tr><td></td><td></td><td rowspan="4">Valeur :</td><td>0</td><td>Val_{1nouv} = <CorVal></td></tr>
<tr><td></td><td></td><td>1</td><td>Val_{1nouv} = Val_{1anc} + <CorVal></td></tr>
<tr><td></td><td></td><td>2</td><td>Val_{1nouv} = <CorVal>
Val_{2nouv} = 0</td></tr>
<tr><td></td><td></td><td>3</td><td>Val_{1nouv} = Val_{1anc} + Val_{2anc} + <CorVal>
Val_{2nouv} = 0</td></tr>
<tr><td></td><td></td><td colspan="3">La syntaxe Val_{1anc} + Val_{2anc} est à interpréter de manière symbolique. Si les deux composantes sont évaluées avec des valeurs différentes (en raison de l'état de <_Stat>), c'est-à-dire qu'une rotation est effective entre les deux composantes, Val_{2anc} est transformée avant l'addition de sorte que la longueur d'outil qui en résulte après la suppression de Val_{2nouv} et avant l'addition de <CorVal> reste inchangée.
<CorVal> se rapporte toujours à Val_1. <CorVal> est une valeur mesurée en fonction de la deuxième partie du paramètre <Comp> dans le système de coordonnées pièce (SCP) ou dans le système de coordonnées outil (SCO). C'est pourquoi il est déjà transformé le cas échéant par rapport à la composante d'outil dans laquelle il doit être pris en compte. Par conséquent, il ne peut pas être immédiatement calculé avec la valeur enregistrée mais doit être retransformé avant d'être additionné à Val_1 ou Val_2. Il peut en résulter une action de la correction dans un axe autre que celui défini par <CorComp> ou une action sur plusieurs axes.
Dans le cas où <CorComp> = 0, c'est-à-dire si <CorVal> ne contient pas de vecteur mais seulement une valeur individuelle, les opérations décrites sont exécutées dans les coordonnées dans lesquelles <CorVal> a été mesuré (SCP/SCO). Cela vaut notamment aussi pour la mise à zéro de Val_{2nouv} dans les variantes 2 et 3. Ce résultat est alors retransformé dans les coordonnées de l'outil. Cela peut avoir pour conséquence qu'aucune des valeurs de coordonnées à remettre à zéro (L1, L2, L3) n'est nulle ou que les valeurs de coordonnées qui étaient nulles auparavant sont à présent différentes de zéro. Toutefois, l'exécution successive des opérations correspondantes pour les trois axes géométriques garantit la nullité pour les trois valeurs de coordonnées de la composante à supprimer. Si l'outil n'est pas pivoté par rapport au système de coordonnées pièce, ou bien pivoté de telle sorte que toutes les composantes d'outil restent parallèles aux axes de coordonnées (permutations d'axes), la modification d'une seule coordonnée d'outil est garantie.
L'exécution successive de la même opération (<CorMode>) avec <CorComp> = 0 pour la totalité des trois axes de coordonnées dans un ordre quelconque est identique à l'exécution unique de la même opération avec <CorComp>=1.
Pour les valeurs de paramètre "0" et "1", le paramètre <Comp> doit contenir un caractère et deux caractères pour les valeurs de paramètre "2" et "3".
Exemple :
<Comp> contient la chaîne "ES", <CorMode> la valeur "2"
⇒ Correction de réglage$_{nouv}$ = <CorVal>, correction de somme$_{nouv}$ = 0
Si le paramètre <CorMode> n'est pas spécifié, sa valeur est "0".</td></tr>
</table>

5	<GeoAx> :	Spécifie l'indice de l'axe géométrique dans lequel la valeur de correction <CorVal>[0] a été mesurée (facultatif)	
		Type de données :	INT
		Plage de valeurs :	0 ... 2
		Les indices 0 à 2 se rapportent aux abscisses, aux ordonnées et aux cotes dans le niveau actif (G17/G18/G19) de l'environnement d'outil actuel. Le contenu de ce paramètre n'est évalué que lorsque le paramètre <CorComp> a la valeur "0".	
6	<Stat> :	Nom du jeu de paramètres pour la description d'un environnement d'outil (facultatif)	
		Type de données :	STRING
		Si la valeur de ce paramètre est la chaîne vide ("") ou si elle n'est pas spécifiée, l'état actuel est utilisé. Si aucun outil n'est spécifié, l'outil actuel est utilisé.	
7	<T> :	Numéro T interne de l'outil (facultatif)	
		Type de données :	INT
		Si ce paramètre n'est pas spécifié ou si sa valeur est "0", l'outil mémorisé dans <Stat> est utilisé. Si la valeur de ce paramètre est "-1", le numéro T de l'outil actif est utilisé. La spécification explicite du numéro de l'outil actif est également admise. **Remarque :** Si <Stat> n'est pas spécifié, l'état actuel est utilisé comme environnement d'outil. Compte tenu qu'avec <T> = 0, le renvoi a lieu au numéro T enregistré dans l'environnement d'outil, c'est l'outil actif qui est utilisé dans celui-ci, c'est-à-dire que les indications <T> = 0 et <T> = -1 ont la même signification dans ce cas particulier.	
8	<D> :	Numéro de tranchant de l'outil (facultatif)	
		Type de données :	INT
		Si ce paramètre n'est pas spécifié ou si sa valeur est "0", le numéro D utilisé dépend de la source du numéro T. Si le numéro T est utilisé à partir de l'environnement d'outil, le numéro D de l'environnement d'outil est lu également, autrement c'est celui de l'outil actuellement actif.	
9	<TL> :	Numéro de la correction en fonction du lieu (facultatif)	
		Type de données :	INT
		Si ce paramètre n'est pas spécifié, le numéro DL utilisé dépend de la source du numéro T. Si le numéro T est utilisé à partir de l'environnement d'outil, le numéro D de l'environnement d'outil est lu également, autrement c'est celui de l'outil actuellement actif. Si T, D et DL spécifient un outil sans corrections en fonction du lieu, aucune correction de somme ou de réglage ne doit être spécifiée dans le paramètre <Comp> (code d'erreur dans <Status>).	

Remarque

Les combinaisons possibles des trois paramètres <Comp>, <CorComp> et <CorMode> ne sont pas toutes pertinentes. Par exemple, la règle opératoire 3 dans <CorComp> exige l'indication de deux caractères dans <Comp>. Si une combinaison non admissible de paramètres est spécifiée, un code d'erreur correspondant est retourné dans <Status>.

Exemples

Exemple 1

Code de programme	Commentaire
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Fraise
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP12[1,1]=1,0	; Usure L1
N50 _CORVAL[0]=0,333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"G",0,0,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> Position SCM X0.000 Y0.000 **Z1.333**
N90 M30	

<CorComp> est "0", c'est pourquoi la valeur de coordonnées effective dans le sens Z de la composante géométrique est remplacée par la valeur de correction 0,333.

La longueur totale d'outil qui en résulte est par conséquent : L1 = 0,333 + 1,000 = 1,333

Exemple 2

Code de programme	Commentaire
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Fraise
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP12[1,1]=1,0	; Usure L1
N50 _CORVAL[0]=0,333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"W",0,1,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> Position SCM X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> est "1", c'est pourquoi la valeur de correction de 0,333 effective dans le sens Z est additionnée à la valeur d'usure de 1,0.

La longueur totale d'outil qui en résulte est par conséquent : L1 = 10,0 + 1,333 = 11,333

Exemple 3

Code de programme	Commentaire
N10 DEF REAL _CORVAL[3]	

Code de programme	Commentaire
N20 $TC_DP1[1,1]=120	; Fraise
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP12[1,1]=1,0	; Usure L1
N50 _CORVAL[0]=0,333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"GW",0,2,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> Position SCM X0.000 Y0.000 Z0.333
N90 M30	

<CorComp> est "2", c'est pourquoi la correction effective dans le sens Z est enregistrée dans la composante géométrique (l'ancienne valeur est écrasée), et la valeur d'usure est supprimée.

La longueur totale d'outil qui en résulte est par conséquent : L1 = 0,333 + 0,0 = 0,333

Exemple 4

Code de programme	Commentaire
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Fraise
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP12[1,1]=1,0	; Usure L1
N50 _CORVAL[0]=0,333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"GW",0,3,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> Position SCM X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> est "3", c'est pourquoi la valeur d'usure et la valeur de correction sont additionnées à la composante géométrique et la composante d'usure est supprimée.

La longueur totale d'outil qui en résulte est par conséquent : L1 = 11,333 + 0,0 = 11,333.

Exemple 5

Code de programme	Commentaire
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Fraise
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP12[1,1]=1,0	; Usure L1
N50 _CORVAL[0]=0,333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"GW",0,3,0)**	
N80 T1 D1 X0 Y0 Z0	; ==> Position SCM X0.333 Y0.000 Z11.000
N90 M30	

<CorComp> est "3" comme dans l'exemple précédent ; la correction agit toutefois à présent sur l'axe géométrique avec l'indice "0" (axe X) qui est affecté dans le cas d'une fraise à la composante d'outil L3 du fait de G17. C'est pourquoi l'appel de SETTCOR n'a aucune influence sur les paramètres d'outil $TC_DP3 et $TC_DP12. À la place, la valeur de correction est enregistrée dans $TC_DP5.

Exemple 6

Code de programme	**Commentaire**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; Outil de tournage
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP4[1,1]=15,0	; Géométrie L2
N50 $TC_DP12[1,1]=10,0	; Usure L1
N60 $TC_DP13[1,1]=0,0	; Usure L2
N70 _CORVAL[0]=5,0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",0,3,1)**	
N110 T1 D1 X0 Y0 Z0	; ==> Position SCM X24.330 Y0.000 Z17.500
N120 M30	

L'outil est un outil de tournage. Une rotation du frame est activée dans le bloc N80 de manière que le système de coordonnées de base (SCB) soit pivoté par rapport au système de coordonnées pièce (SCP). La valeur de correction (N70) agit dans le SCP sur l'axe géométrique avec l'indice "1", c'est-à-dire, vu que G18 est actif, sur l'axe X. Compte tenu que <CorMode> = 3 s'applique, l'usure de l'outil dans le sens de l'axe X du SCP doit être nulle après l'exécution de N100.

C'est pourquoi le contenu des paramètres d'outil pertinents est le suivant à la fin du programme :

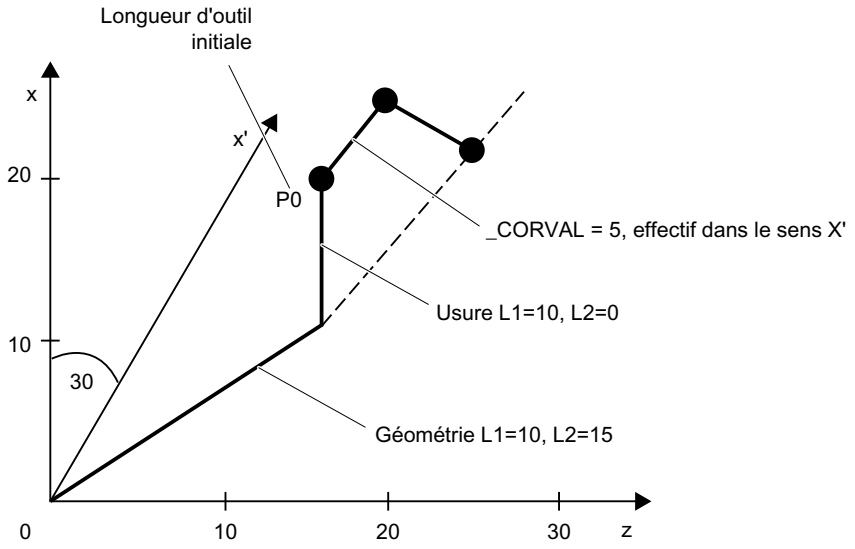
$TC_DP3[1,1] : 21,830 ; Géométrie L1

$TC_DP4[1,1] : 21,830 ; Géométrie L2

$TC_DP12[1,1] : 2,500 ; Usure L1

$TC_DP13[1,1] : -4,330 ; Usure L2

La figure ci-dessous illustre les conditions géométriques. L'usure totale incluant _CORVAL est projetée sur le sens X' dans le SCP. Il en résulte le point P2. Les coordonnées de ce point (mesurées en coordonnées X-Y) sont enregistrées dans la composante géométrique de l'outil. Le vecteur de différence P_2 - P_1 subsiste dans l'usure. De ce fait, l'usure n'a plus aucune composante dans le sens de _CORVAL.



Si l'on poursuit l'exemple de programme après N110 avec les instructions suivantes, l'usure résiduelle est totalement prise en compte dans la géométrie, compte tenu que la correction agit à présent dans l'axe Z' (paramètre <GeoAx> = 0) :

```
N120 _CORVAL[0]=0,0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> Position SCM X24.330 Y0.000 Z17.500
```

Compte tenu que la nouvelle valeur de correction est "0", la longueur totale de l'outil et, de ce fait également, la position accostée dans N140 ne doit pas être modifiée. Si _CORVAL était différent de "0" dans N120, il en résulterait une nouvelle longueur totale de l'outil et, par voie de conséquence, une position modifiée dans N140. La composante d'usure de la longueur d'outil serait toutefois nulle dans ce cas, c'est-à-dire que la longueur totale de l'outil est ensuite toujours contenue dans la composante géométrique de l'outil.

Le même résultat qu'avec l'appel répété de la fonction SETTCOR avec le paramètre <CorComp> = 0 est obtenu également par un appel unique avec <CorComp> = 1 (correction vectorielle) :

Code de programme	**Commentaire**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; Outil de tournage
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP4[1,1]=15,0	; Géométrie L2
N50 $TC_DP12[1,1]=10,0	; Usure L1
N60 $TC_DP13[1,1]=0,0	; Usure L2
N70 _CORVAL[0]=0,0	
N71 _CORVAL[1]=5,0	
N72 _CORVAL[2]=0,0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",1,3,1)**	
N110 T1 D1 X0 Y0 Z0	; ==> Position SCM X24.330 Y0.000 Z17.500

Code de programme	Commentaire
N120 M30	

Dans ce cas, toutes les composantes d'usure de l'outil sont égales à zéro immédiatement après le premier appel de SETTCOR dans N100.

Exemple 7

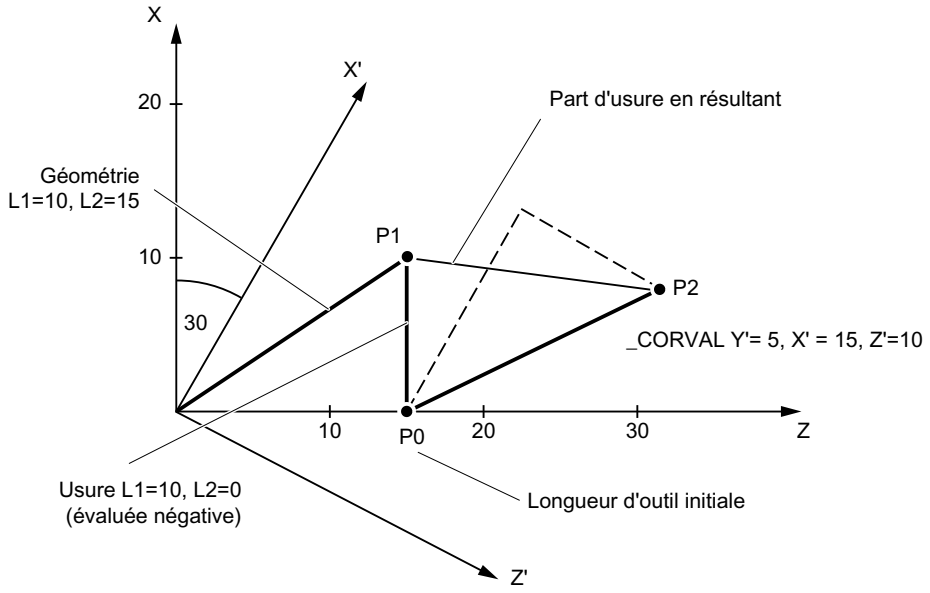
Code de programme	Commentaire
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; Outil de tournage
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP4[1,1]=15,0	; Géométrie L2
N50 $TC_DP12[1,1]=10,0	; Usure L1
N60 $TC_DP13[1,1]=0,0	; Usure L2
N70 _CORVAL[0]=5,0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",3,3)**	
N110 T1 D1 X0 Y0 Z0	; ==> Position SCM X25.000 Y0.000 Z15.000
N120 M30	

Par rapport à l'exemple 6, le paramètre est ici <CorComp> = 3, c'est pourquoi le paramètre <GeoAx> peut être omis. La valeur contenue dans _CORVAL[0] agit à présent immédiatement sur la composante d'outil L1, la rotation dans N80 n'a aucune influence sur le résultat, les composantes d'usure dans $TC_DP12 sont prises en compte avec _CORVAL[0] dans la composante géométrique. Ainsi la longueur totale de l'outil se trouve, du fait de $TC_DP13, dans la composante géométrique de l'outil dès le premier appel de SETTCOR dans N100.

Exemple 8

Code de programme	Commentaire
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; Outil de tournage
N30 $TC_DP3[1,1]=10,0	; Géométrie L1
N40 $TC_DP4[1,1]=15,0	; Géométrie L2
N50 $TC_DP5[1,1]=20,0	; Géométrie L3
N60 $TC_DP12[1,1]=10,0	; Usure L1
N70 $TC_DP13[1,1]=0,0	; Usure L2
N80 $TC_DP14[1,1]=0,0	; Usure L3
N90 $SC_WEAR_SIGN=TRUE	
N100 _CORVAL[0]=10,0	
N110 _CORVAL[1]=15,0	
N120 _CORVAL[2]=5,0	
N130 ROT Y-30	
N140 T1 D1 G18 G0	
N150 R1=**SETTCOR(_CORVAL,"W",1,1)**	
N160 T1 D1 X0 Y0 Z0	; ==> Position SCM X7.990 Y25.000 Z31.160
N170 M30	

La donnée de réglage SD42930 $SC_WEAR_SIGN est mise à 1 dans N90, c'est-à-dire que l'usure doit être évaluée avec un signe négatif. La correction est vectorielle (<CorComp> = 1) et le vecteur de correction doit être ajouté à l'usure (<CorMode> = 1). La figure ci-dessous illustre les conditions géométriques dans le plan Z-X :



Étant donné que <CorMode> = 1, la composante géométrique de l'outil reste inchangée. Le vecteur de correction défini dans le SCP (rotation autour de l'axe Y) doit être pris en compte dans la composante d'usure, de sorte que la longueur totale de l'outil renvoie au point P_2 dans la figure 3. C'est pourquoi la composante d'usure de l'outil qui en résulte est donnée par la distance des deux points P_1 et P_2.
Mais compte tenu de l'évaluation négative de l'usure en raison de la donnée de réglage SD42930 $SC_WEAR_SIGN, la correction ainsi calculée doit être saisie dans la mémoire de correcteurs avec un signe négatif. Ainsi, le contenu des paramètres d'outil pertinents est le suivant à la fin du programme :

$TC_DP3[1,1] : 10,000 ; Géométrie L1 (inchangée)

$TC_DP4[1,1] : 15,000 ; Géométrie L2 (inchangée)

$TC_DP5[1,1] : 10,000 ; Géométrie L3 (inchangée)

$TC_DP12[1,1] : 2,010 ; Usure L1 (= 10 - 15 * cos(30) + 10 * sin(30))

$TC_DP13[1,1] : -16,160 ; Usure L2 (= -15 * sin(30) - 10 * cos(30))

$TC_DP14[1,1] : -5,000 ; Usure L3

Au niveau de la composante L3 dans le sens Y, on constate l'effet de la donnée de réglage SD42930 $SC_WEAR_SIGN sans la complication supplémentaire due à la rotation du frame.

Informations complémentaires

Outils de tournage et de rectification : Calcul de la longueur d'outil en fonction de PM20360 $MC_TOOL_PARAMETER_DEF_MASK

La façon dont l'usure ou la longueur d'outil des outils de tournage et de rectification doit être évaluée dans un axe diamétral éventuellement présent est déterminée dans le paramètre machine :

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <Valeur>

<Bit>	<Valeur>	Signification
0	0	Sur les outils de tournage et de rectification, le **paramètre d'usure** de l'axe transversal est pris en compte dans la **valeur de rayon** :
	1	Sur les outils de tournage et de rectification, le **paramètre d'usure** de l'axe transversal est pris en compte en tant que **valeur de diamètre** :
1	0	Sur les outils de tournage et de rectification, la **composante de longueur d'outil** de l'axe transversal est prise en compte en tant que **valeur de rayon** :
	1	Sur les outils de tournage et de rectification, la **composante de longueur d'outil** de l'axe transversal est prise en compte en tant que **valeur de diamètre** :

Si les bits concernés sont mis à 1, l'entrée associée est évaluée avec le facteur 0,5. La correction au moyen de SETTCOR est réalisée de sorte que la modification effective de la longueur totale de l'outil soit identique à celle transmise dans <CorVal>. Si une longueur est évaluée avec le facteur 0,5 sur la base du paramètre machine PM20360 $MC_TOOL_PARAMETER_DEF_MASK pour le calcul de longueur, la correction de cette composante doit donc s'effectuer avec la valeur double transférée.

Exemple

PM20360 $MC_TOOL_PARAMETER_DEF_MASK = 2 (la longueur de l'outil doit être évaluée dans l'axe diamétral avec le facteur 0,5)

L'axe X et l'axe de diamétral.

```
Code de programme                          Commentaire
N10 DEF REAL _LEN[11]
N20 DEF REAL _CORVAL[3]
N30 $TC_DP1[1,1]=500                       ; Type d'outil
N40 $TC_DP2[1,1]=2                         ; Position du tranchant
N50 $TC_DP3[1,1]=3.                        ; Géométrie - longueur 1
N60 $TC_DP4[1,1]=4.                        ; Géométrie - longueur 2
N70 $TC_DP5[1,1]=5.                        ; Géométrie - longueur 3
N80 _CORVAL[0]=1.
N90 _CORVAL[1]=1.
N100 _CORVAL[2]=1.
N110 T1 D1 G18 G0 X0 Y0 Z0                 ; ==> Position SCM X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)
N130 T1 D1 X0 Y0 Z0                        ; ==> Position SCM X2.5 Y6 Z5
N140 R3=$TC_DP3[1,1]                       ; = 5. = (3,000 + 2.*1,000)
N150 R4=$TC_DP4[1,1]                       ; = 5. = (4,000 + 1,000)
```

Code de programme	**Commentaire**
N160 R5=$TC_DP5[1,1]	; = 6. = (5,000 + 1,000)
N170 M30	

La correction de la longueur d'outil doit être de 1 mm dans chaque axe (N80 à N100). C'est pourquoi 1 mm est ajouté à la longueur initiale dans les longueurs L2 et L3. En revanche, la valeur de correction double (2 mm) est ajoutée à la longueur d'outil initiale dans L1, afin de modifier de 1 mm la longueur totale de l'outil comme exigé. Par rapport aux positions accostées dans les blocs N110 et N130, on constate que chaque position d'axe a changé de 1 mm.

11.13 Lecture de l'affectation des longueurs d'outil L1, L2, L3 par rapport aux axes de coordonnées (LENTOAX)

La fonction LENTOAX fournit des informations sur l'affectation des longueurs L1, L2 et L3 de l'outil **actif** par rapport aux abscisses, aux ordonnées et aux cotes. L'affectation des abscisses, ordonnées et cotes aux axes géométriques est influencée par les frames et le plan actif (G17 - G19).

Seule la composante géométrique d'un outil ($TC_DP3[<t>,<d>] à $TC_DP5[<t>,<d>]) est prise en compte, c'est-à-dire qu'une affectation d'axe d'autres composantes qui diffèrerait de celle-ci (par ex. usure) n'a aucune incidence sur le résultat.

Syntaxe

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

Principe

`LENTOAX(...)` :	Fonction prédéfinie pour la lecture de l'affectation des longueurs L1, L2 et L3 de l'outil actif aux axes de coordonnées.		
	Seule dans le bloc :		oui
`<Status>` :	Valeur de retour de la fonction. Les valeurs négatives indiquent des états d'erreur.		
	Type de données :		INT
	Valeur :	0	Fonction OK L'information qui figure dans <AxInd> est suffisante pour la description (toutes les composantes de longueur d'outil sont parallèles aux axes géométriques).
		1	La fonction est OK mais pour une description correcte le contenu de <Matrix> doit être analysé (les composantes de longueur d'outil ne sont pas parallèles aux axes géométriques).
		-1	Chaîne non valide dans le paramètre <Coord>.
		-2	Aucun outil actif.
Paramètres			

1	<AxInd> :	Si les composantes de longueur d'outil sont parallèles aux axes géométriques, les indices d'axe assignés aux composantes de longueur L1 à L3 sont retournés dans le champ **<**AxInd**>** : • **<**AxInd**>** [0] : abscisse • **<**AxInd**>** [1] : ordonnée • **<**AxInd**>** [2] : cote		
		Type de données :		INT[3]
		Valeur :	0	Aucune affectation disponible (l'axe n'existe pas)
			1 ... 3 ou -1 ... -3	Numéro de la longueur qui agit dans l'axe de coordonnées correspondant. Le signe est négatif lorsque la composante de longueur d'outil est orientée dans le sens négatif des coordonnées.
		Si les composantes de longueur ne sont pas toutes parallèles ou antiparallèles aux axes géométriques, l'indice est retourné dans **<**AxInd**>** à l'axe qui contient la plus grande part d'une composante de longueur d'outil. Dans ce cas (si la fonction ne génère pas une erreur pour une autre raison), la valeur retournée est **<**Status**>** = 1. La représentation des composantes de longueur d'outil L1 à L3 sur les axes géométriques 1 à 3 est alors entièrement décrite par le contenu du 2e paramètre **<**Matrix**>**.		
2	<Matrix> :	Matrice qui constitue le vecteur des longueurs d'outil (L1=1, L2=1, L3=1) dans le vecteur des axes de coordonnées (abscisse, ordonnée, cote), c'est-à-dire que les composantes de longueur d'outil sont assignés aux colonnes dans l'ordre L1, L2, L3, et les axes sont affectés aux lignes dans l'ordre abscisse, ordonnée, cote.		
		Type de données :		REAL
		Dans la matrice, tous les éléments sont toujours valides, même si l'axe géométrique faisant partie d'un axe de coordonnées n'est pas disponible, c'est-à-dire si l'entrée correspondante dans **<**AxInd**>** est égale à zéro.		
3	<Coord> :	Système de coordonnées pour lequel l'affectation s'applique (facultatif)		
		Type de données :		STRING
		Symbole :	MCS M	Représentation de la longueur d'outil dans le système de coordonnées machine
			BCS B	Représentation de la longueur d'outil dans le système de coordonnées de base
			WCS W	Représentation de la longueur d'outil dans le système de coordonnées pièce (par défaut)
			KCS K	Représentation de la longueur d'outil dans le système de coordonnées outil de la transformation cinématique
			TCS T	Représentation de la longueur d'outil dans le système de coordonnées outil
		La casse des caractères dans la chaîne (majuscules ou minuscules) est sans importance. Si le paramètre **<**Coord**>** est non spécifié, le SCP est utilisé (par défaut).		

Remarque

Dans le TCS, toutes les composantes de longueur d'outil sont toujours parallèles ou antiparallèles aux axes.

Les composantes ne peuvent être antiparallèles que si la fonction miroir est active et que la donnée de réglage suivante est définie :

SD42900 $SC_MIRROR_TOOL_LENGTH (inversion du signe de la longueur d'outil pour la fonction miroir)

Exemple

Cas standard d'outil de fraisage avec G17.

L1 agit dans Z (cote), L2 agit dans Y (ordonnée), L3 agit dans X (abscisse).

Appel de la fonction sous la forme :
`<Status>=LENTOAX(<AxInd>,<Matrix>,"WCS")`

Le paramètre de résultat <AxInd> contient alors les valeurs :

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

ou en abrégé : (3, 2, 1)

La matrice associée (<Matrix>) est dans ce cas :

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Le passage de G17 vers G18 ou G19 ne modifie en rien le résultat car l'affectation des composantes de longueur aux axes géométriques varie de la même façon que l'affectation des abscisses, ordonnées et cotes.

Si G17 est actif, une rotation du frame de 60 degrés autour de Z est programmée, par ex. :
`ROT Z60`

Le sens de la cote (sens Z) reste inchangé, la part principale de L2 se situe à présent dans le sens du nouvel axe X, la part principale de L1 dans le sens de l'axe Y négatif. La valeur de retour (<Status>) est donc "1", <AxInd> contient les valeurs (2, -3, 1).

La matrice associée (<Matrix>) est dans ce cas :

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

Modes de déplacement 12

12.1 Positionnement tangentiel

12.1.1 Définir le couplage (TANG)

La procédure TANG(...) prédéfinie permet de définir un couplage tangentiel entre un axe rotatif en tant qu'axe asservi et deux axes géométriques en tant qu'axes pilotes. L'axe asservi est alors aligné en continu sur la tangente à la trajectoire des axes pilotes.

Remarque

facteur de couplage

Un facteur de couplage de 1 ne doit pas obligatoirement être programmé de manière explicite. Un facteur de couplage de -1 provoque la rotation du sens de l'axe tangentiel.

Syntaxe

```
TANG(<Axe asservi>, <Axe pilote_1>, <Axe pilote_2>, <Facteur de
couplage>, <Système de coordonnées>, <Optimisation>)
```

Signification

TANG(...) :	Définition d'un couplage tangentiel	
<Axe asservi> :	Nom d'axe de l'axe asservi (axe rotatif)	
	Type de données :	AXIS
	Plage de valeurs :	Noms d'axes de canal
<Axe pilote_1> <Axe pilote_2> :	Noms d'axes des axes pilotes (axes géométriques) [1]	
	Type de données :	AXIS
	Plage de valeurs :	Noms des axes géométriques du canal
<Facteur de couplage> :	Facteur n de variation angulaire de l'axe asservi pour modifier la tangente à la trajectoire des axes pilotes : Variation angulaire$_{\text{Axe asservi}}$ = Variation angulaire$_{\text{Tangente à la trajectoire}}$ * n	
	Type de données :	REAL
	Valeur par défaut :	1,0

<Système de coordonnées> :	Système de coordonnées actif [2]		
	Type de données :	CHAR	
	Valeur :	"B" :	Système de coordonnées de base (valeur par défaut)
		"W" :	Système de coordonnées pièces (non disponible)
<Optimisation> :	Type d'optimisation		
	Type de données :	CHAR	
	Valeur :	"S" :	Standard (par défaut) La dynamique de l'axe rotatif n'a aucun effet sur les axes pilotes. Si la dynamique de l'axe rotatif est plus importante que nécessaire pour l'asservissement, cette méthode est suffisamment précise. Si la dynamique de l'axe rotatif n'est pas suffisamment importante pour suivre la modification des tangentes à la trajectoire, l'orientation de l'axe rotatif s'écarte de l'orientation de consigne le long d'une trajectoire d'arrondissement non définie.
		"P" :	La dynamique de l'axe rotatif est prise en compte lors de la planification de trajectoire des axes pilotes. A cet effet, deux paramètres supplémentaires doivent être indiqués lors de l'activation du couplage tangentiel avec TANGON() : • Trajectoire d'arrondissement • Tolérance angulaire Voir chapitre "Activer le couplage (TANGON) (Page 534)" **Remarque** En relation avec les transformations cinématiques, il est recommandé d'utiliser le type d'optimisation "P".

Remarque

Les valeurs par défaut ne doivent pas obligatoirement être programmées de manière explicite.

[1] **Remarque**

Comme axes pilotes du couplage tangentiel, il est nécessaire d'utiliser les axes géométriques qui effectuent un retrait selon la trajectoire programmée dans le système de coordonnées machine (SCM) par rapport à la position de base de la machine. Par exemple, sur une fraiseuse avec tête orientable, si le cycle d'orientation CYCLE800 est utilisé, l'interpolation a lieu, selon la configuration du cycle, dans le SCP par ex. avec les axes géométriques X et Y. Cependant, le couplage tangentiel doit être défini avec comme axes pilotes les axes géométriques qui effectuent un retrait selon la trajectoire programmée dans le SCM. De plus, il convient d'utiliser comme axes pilotes les axes géométriques à l'état **non orienté** de la machine.

[2] **Remarque**

Le système de coordonnées de base (SCB) ne doit pas être pivoté par rapport au SCM. Si le SCB est pivoté par ex. via l'instruction ROT ou via le cycle d'orientation CYCLE800, le positionnement tangentiel ne fonctionne plus correctement.

12.1.2 Activer la génération de bloc intermédiaire (TLIFT)

Si la modification de la tangente de l'axe asservi dépasse à un point donné de la trajectoire programmée des axes pilotes la valeur limite configurée dans le paramètre machine PM37400 $MA_EPS_TLIFT_TANG_STEP, la suite de la planification de la trajectoire s'effectue en fonction du comportement aux angles réglé. Sans l'utilisation de la procédure TLIFT(...) prédéfinie, le déplacement s'effectue en fonction du comportement d'arrondissement programmé en relation avec TANG(...) (Page 531) et TANGON(...) (Page 534).

Activer la génération de bloc intermédiaire

Avec la programmation de TLIFT(...) après TANG(...), le prétraitement insère un bloc intermédiaire généré automatiquement par la commande en cas de détection d'un angle à cet endroit de la trajectoire.

Lors de l'exécution du programme, les axes pilotes sont alors arrêtés lorsque le bloc intermédiaire est atteint. Dans le bloc intermédiaire, l'axe asservi est pivoté avec la dynamique d'axe maximale en direction de la tangente à la trajectoire du bloc suivant. Ensuite, le déplacement des axes pilotes selon la trajectoire programmée se poursuit.

Désactiver la génération de bloc intermédiaire

Pour désactiver la génération de bloc intermédiaire, le couplage tangentiel doit être de nouveau défini au moyen de TANG(...), mais sans activation consécutive de la génération de bloc intermédiaire au moyen de TLIFT(...).

Syntaxe

```
TLIFT(<Axe asservi>)
```

Signification

`TLIFT(...)` :	Activation de la détection d'angle avec calcul de bloc intermédiaire	
`<Axe asservi>` :	Nom d'axe de l'axe asservi (axe rotatif)	
	Type de données :	AXIS
	Plage de valeurs :	Noms d'axes de canal

Vitesse de rotation de l'axe asservi

Axe d'interpolation

Si l'axe asservi se déplaçait déjà en tant qu'axe d'interpolation avant l'activation du couplage tangentiel, le mouvement de rotation s'effectue dans le bloc intermédiaire en tant qu'axe d'interpolation.

En indiquant le rayon de référence avec `FGREF[<Axe>]=0.001`, le mouvement de rotation s'effectue à la vitesse d'axe maximale paramétrée :

PM32000 $MA_MAX_AX_VELO[<Axe asservi>]

Axe de positionnement

Si l'axe asservi ne se déplaçait pas encore en tant qu'axe d'interpolation avant l'activation du couplage tangentiel, la rotation s'effectue dans le bloc intermédiaire en tant qu'axe de positionnement.

Le mouvement de rotation s'effectue alors à la vitesse d'axe de positionnement paramétrée :

PM32060 $MA_POS_AX_VELO[<Axe asservi>]

12.1.3 Activer le couplage (TANGON)

La procédure TANGON(...) prédéfinie permet d'activer un couplage tangentiel défini au préalable avec TANG(...) (Page 531). L'axe asservi est ensuite aligné en continu sur la tangente à la trajectoire lors du déplacement suivant des axes pilotes.

Angle de l'axe asservi

L'angle que prend l'axe asservi par rapport à la tangente à la trajectoire dépend du rapport de transmission indiqué dans TANG(...), de l'angle de décalage configuré dans le paramètre machine PM37402 $MA_TANG_OFFSET et de l'angle de décalage indiqué avec TANGON(...), qui s'y ajoute.

Optimisation "P"

Si la valeur "P" a été indiquée comme paramètre d'optimisation lors de la définition du couplage tangentiel (TANG(...)), le paramètre "Trajectoire d'arrondissement" et, de manière facultative, le paramètre "Tolérance angulaire" doivent être indiqués lors de l'activation du couplage.

Si la valeur 0 est spécifiée comme tolérance angulaire, seul le paramètre "Trajectoire d'arrondissement" est effectif.

Si une valeur supérieure à 0 est spécifiée comme tolérance angulaire, la trajectoire d'arrondissement effective est la valeur la plus faible entre la trajectoire d'arrondissement paramétrée et la trajectoire d'arrondissement due à la tolérance angulaire paramétrée.

Si la dynamique de l'axe asservi n'est pas suffisante pour respecter les conditions paramétrées, la vitesse tangentielle des axes pilotes est réduite en conséquence.

Syntaxe

```
TANGON(<Axe asservi>, <Angle de décalage>, <Trajectoire
d'arrondissement>, <Tolérance angulaire>)
```

Signification

`TANGON(...)` :	Activation d'un couplage tangentiel	
`<Axe asservi>` :	Nom d'axe de l'axe asservi (axe rotatif)	
	Type de données :	AXIS
	Plage de valeurs :	Noms d'axes de canal

<Angle de décalage> :	Angle de décalage de l'axe asservi par rapport à la tangente à la trajectoire Le point de référence est l'origine de l'axe rotatif.	
	Type de données :	REAL
<Trajectoire d'arrondissement> :	Trajectoire d'arrondissement maximale admissible Si la trajectoire d'arrondissement a augmenté en raison des conditions de dynamique, la vitesse tangentielle des axes pilotes est réduite.	
	Type de données :	REAL
<Tolérance angulaire> :	Tolérance maximale admissible par rapport à l'angle indiqué entre l'origine de l'axe asservi et la tangente à la trajectoire	
	Type de données :	REAL

12.1.4 Désactiver le couplage (TANGOF)

La procédure TANGOF(...) prédéfinie permet de désactiver un couplage tangentiel défini avec TANG(...) (Page 531) et activé avec TANGON(...) (Page 534). L'axe asservi n'est ensuite plus aligné sur la tangente à la trajectoire des axes pilotes. Cependant, le couplage de l'axe asservi avec les axes pilotes est maintenu même après la désactivation, ce qui empêche par ex. les fonctions suivantes :

- Changement de plan
- Permutation d'axes géométriques
- Définition d'un nouveau couplage tangentiel pour l'axe asservi

L'annulation complète du lien entre l'axe asservi et les axes pilotes a lieu uniquement après la suppression du couplage via TANGDEL(...) (Page 536).

Programmation

```
TANGOF(<Axe asservi>)
```

Signification

TANGOF(...) :	Désactivation d'un couplage tangentiel	
<Axe asservi> :	Nom d'axe de l'axe asservi (axe rotatif)	
	Type de données :	AXIS
	Plage de valeurs :	Noms d'axes de canal

12.1.5 Supprimer un couplage (TANGDEL)

Un couplage tangentiel défini avec TANG(...) (Page 531) est maintenu même après la désactivation du couplage tangentiel avec TANGOF(...) (Page 535). Le couplage tangentiel existant continue alors à empêcher par ex. les fonctions suivantes :

- Changement de plan
- Permutation d'axes géométriques
- Définition d'un nouveau couplage tangentiel pour l'axe asservi

La procédure TANGDEL(...) prédéfinie permet de supprimer le couplage tangentiel existant après la désactivation du couplage tangentiel avec TANGOF(...)

Syntaxe

```
TANGDEL(<Axe asservi>)
```

Signification

TANGDEL(...) :	Suppression d'un couplage tangentiel défini avec TANG()	
	Prise d'effet :	Non modal
<Axe asservi> :	Nom de l'axe asservi dont le couplage tangentiel doit être supprimé	
	Type de données :	AXIS
	Plage de valeurs :	Noms d'axes de canal

Exemples

Changement d'axe pilote

Avant qu'un nouveau couplage tangentiel avec un autre axe pilote ne puisse être défini pour l'axe asservi, le couplage tangentiel existant doit d'abord être supprimé.

```
Code de programme                Commentaire
N10 TANG(A, X, Y, 1)             ; Définir un couplage tangentiel pour l'axe asservi A :
                                   A avec X et Y
N20 TANGON(A)                    ; Activer le couplage tangentiel pour l'axe asservi A
N30 X10 Y20
...
N80 TANGOF(A)                    ; Désactiver le couplage tangentiel pour l'axe asservi
                                   A
N90 TANGDEL(A)                   ; Supprimer le couplage tangentiel pour l'axe asservi A
...
N120 TANG(A, X, Z)               ; Définir un nouveau couplage tangentiel pour l'axe as-
                                   servi A
N130 TANGON(A)                   ; Activer le nouveau couplage tangentiel pour l'axe as-
                                   servi A
...
```

Permutation d'axes géométriques

Avant qu'une permutation d'axes géométriques ne puisse être effectuée pour un couplage existant, le couplage doit d'abord être supprimé.

Code de programme	**Commentaire**
N10 GEOAX(2, Y1)	; 2ème axe géométrique = Axe machine Y1
N20 TANG(A, X, Y)	; Définir un couplage tangentiel pour l'axe asservi A
N30 TANGON(A, 90)	; Activer le couplage tangentiel pour l'axe asservi A
N40 G2 F8000 X0 Y0 I0 J50	; Bloc de déplacement
N50 TANGOF(A)	; Désactiver le couplage tangentiel pour l'axe asservi A
N60 **TANGDEL**(A)	; Supprimer le couplage tangentiel pour l'axe asservi A
N70 **GEOAX**(2, Y2)	; 2ème axe géométrique = Axe machine Y2
N80 TANG(A, X, Y)	; Définir un nouveau couplage tangentiel pour l'axe asservi A
N90 TANGON(A, 90)	; Activer le nouveau couplage tangentiel pour l'axe asservi A
...	

12.2 Variation de l'avance (FNORM, FLIN, FCUB, FPO)

En vue d'une définition plus flexible de l'avance, la programmation de l'avance selon DIN 66025 est étendue aux variations linéaires et cubiques.

Les variations cubiques peuvent être programmées directement ou sous la forme de splines interpolés. Par ce moyen, il est possible de programmer des variations continues et lissées, dépendant de la courbure de la pièce à usiner.

Ces variations de la vitesse permettent des variations d'accélération sans à-coup et, de ce fait, la réalisation d'états de surface plus réguliers.

Syntaxe

```
F… FNORM
F… FLIN
F… FCUB
F=FPO(…,…,…)
```

Signification

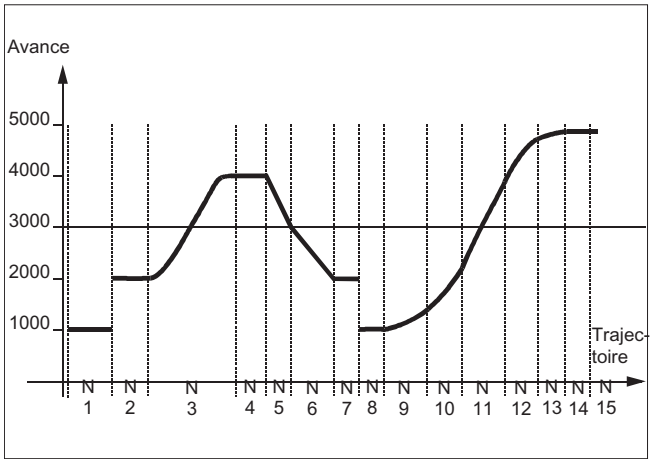
`FNORM` :	Préréglage. La valeur de l'avance est définie sur la longueur de la trajectoire du bloc et vaut alors comme valeur modale.
`FLIN` :	Variation de l'avance tangentielle **linéaire :** La valeur de l'avance varie linéairement de la valeur en début de bloc à la valeur en fin de bloc et vaut alors comme valeur modale. Ce comportement peut être combiné avec G93 et G94.
`FCUB` :	Variation de l'avance tangentielle **cubique :** Les valeurs F programmées dans les blocs - rapportées au point final de bloc - sont raccordées par un spline. Le spline débute et se termine de façon tangentielle aux valeurs d'avance programmées dans le bloc précédent, voire suivant, et agit avec G93 et G94. Si l'adresse F manque dans un bloc, c'est la dernière valeur F programmée qui sera utilisée.
`F=FPO…` :	Variation de l'avance tangentielle **définie par un polynôme :** l'adresse F décrit la variation de l'avance selon un polynôme, de la valeur courante jusqu'à la valeur en fin de bloc. La valeur finale vaut alors comme valeur modale.

Optimisation de l'avance sur des sections de trajectoire courbes

Le polynôme d'avance `F=FPO` et le spline d'avance `FCUB` doivent toujours être exécutés à vitesse de coupe constante `CFC` . De ce fait, une variation de consigne d'avance à accélération continue est générée.

Exemple : Différentes courbes d'avance

Vous trouverez, dans cet exemple, la programmation et la représentation graphique de différentes courbes d'avance.



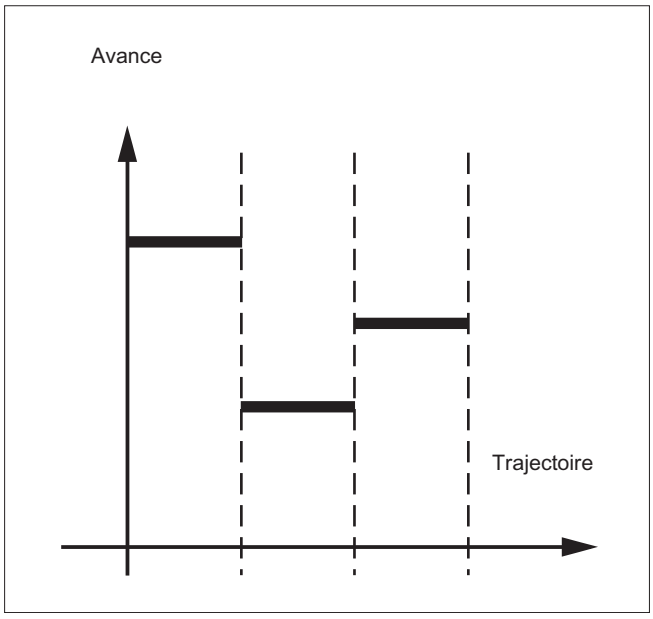
Code de programme	**Commentaire**
N1 F1000 FNORM G1 X8 G91 G64	; Courbe d'avance constante, indication en cotes relatives
N2 F2000 X7	; Variation brusque de la vitesse de consigne
N3 F=FPO(4000, 6000, -4000)	; Courbe d'avance par polynôme avec avance 4000 en fin de bloc
N4 X6	; Avance polynomiale 4000 prise comme valeur modale.
N5 F3000 FLIN X5	; Courbe d'avance linéaire
N6 F2000 X8	; Courbe d'avance linéaire
N7 X5	; Avance linéaire prise comme valeur modale
N8 F1000 FNORM X5	; Courbe d'avance constante avec variation d'accélération en échelon.
N9 F1400 FCUB X8	; Toutes les valeurs de F programmées dans les blocs suivants sont reliées par des splines.
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; Désactiver la courbe spline.
N14 FNORM X5	
N15 X20	

Informations complémentaires

FNORM

L'adresse d'avance F désigne l'avance tangentielle en tant que valeur constante selon DIN 66025.

Vous trouverez plus d'informations sur ce sujet dans "Manuel de programmation - Notions de base".

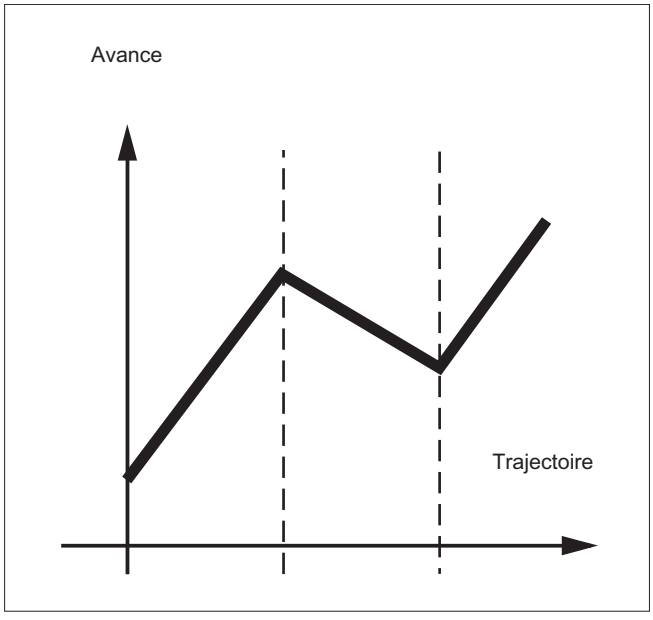


FLIN

L'avance varie de façon linéaire jusqu'à la fin du bloc, depuis la valeur d'avance courante jusqu'à la valeur F programmée.

Exemple :

```
N30 F1400 FLIN X50
```

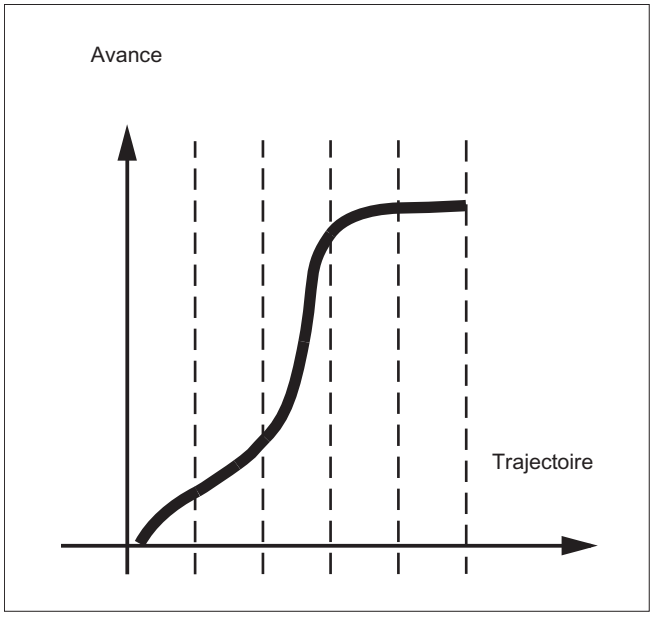


FCUB

L'avance varie de façon cubique jusqu'à la fin du bloc, depuis la valeur courante jusqu'à la valeur F programmée. La commande relie toutes les valeurs programmées dans les blocs avec un FCUB actif par des splines. Les valeurs d'avance servent dans ce cas de points intermédiaires pour le calcul de l'interpolation spline.

Exemple :

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```



F=FPO(...,...,...)

La variation de l'avance est directement programmée par le biais d'un polynôme. L'introduction des coefficients du polynôme est la même que pour l'interpolation polynomiale.

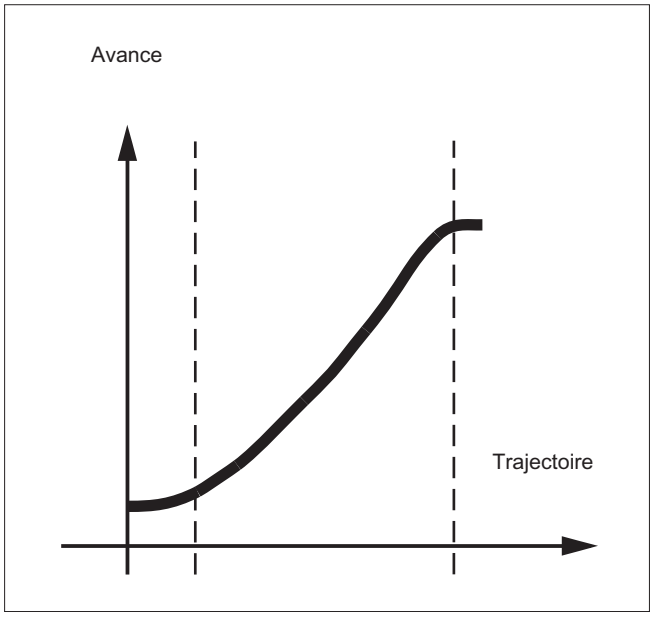
Exemple :

```
F=FPO(endfeed, quadf, cubf)
```

endfeed, quadf et cubf sont des variables précédemment définies.

endfeed :	Avance en fin de bloc
quadf :	Coefficient de polynôme de degré 2
cubf :	Coefficient de polynôme de degré 3

Quand `FCUB` est activée, le spline se raccorde tangentiellement en début et en fin de bloc à l'avance définie par le biais de `FPO` .



Autres conditions à prendre en compte

- Les fonctions programmées pour le mode de déplacement restent valables, indépendamment de la variation de l'avance.
- La variation programmable de l'avance est toujours valable de façon absolue, indépendamment de `G90` ou `G91`.
- La variation de l'avance `FLIN` et `FCUB` fonctionne avec `G93` et `G94`, **sauf** avec `G95`, `G96/G961` et `G97/G971`.
- Lorsque le compacteur `COMPON` est activé, les règles ci-après sont appliquées quand plusieurs blocs sont regroupés en un segment de spline :

`FNORM` :	Pour le segment de spline, le mot F du dernier bloc qui en fait partie est valide.
`FLIN` :	Pour le segment de spline, le mot F du dernier bloc qui en fait partie est valide. La valeur F programmée est valide en fin du segment et est alors accostée de façon linéaire.
`FCUB` :	Le spline d'avance généré s'écarte au maximum des points finaux programmés d'une valeur égale à la valeur définie dans le paramètre machine PM20172 $MC_COMPRESS_VELO_TOL.
`F=FPO(…,…,…)` :	Ces blocs ne sont pas compactés.

12.3 Comportement à l'accélération

12.3.1 Mode d'accélération (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

Pour programmer le mode d'accélération, on dispose des instructions de programme pièce suivantes :

- "BRISK, BRISKA"
  Le déplacement des axes individuels et des axes à interpolation s'opère avec une accélération maximale jusqu'à ce que la vitesse d'avance programmée soit atteinte (**accélération sans limitation des à-coups**).
- "SOFT, SOFTA"
  Le déplacement des axes individuels et des axes à interpolation s'opère avec une accélération constante jusqu'à ce que la vitesse d'avance programmée soit atteinte (**accélération avec limitation des à-coups**).
- "DRIVE, DRIVEA"
  Le déplacement des axes individuels et des axes à interpolation s'opère avec une accélération maximale jusqu'à une limite de vitesse programmée (réglage du PM). L'accélération est ensuite réduite (réglage du PM) jusqu'à ce que la vitesse d'avance programmée soit atteinte.

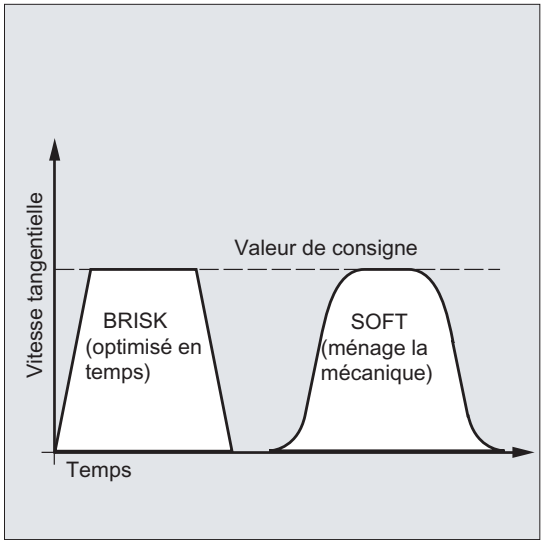


Figure 12-1 Courbes de vitesse tangentielle avec `BRISK` et `SOFT`

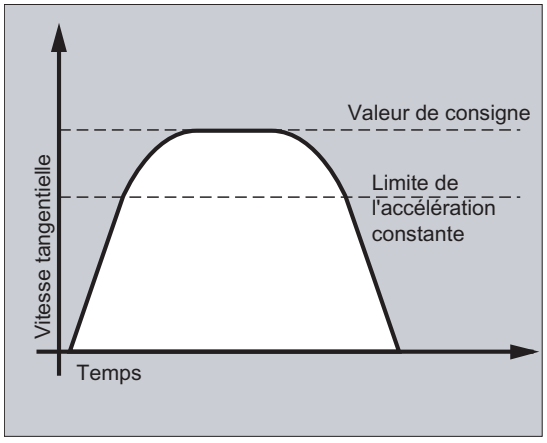


Figure 12-2 Courbe de vitesse tangentielle avec DRIVE

Syntaxe

```
BRISK
BRISKA(<axe1>,<axe2>,…)
SOFT
SOFTA(<axe1>,<axe2>,…)
DRIVE
DRIVEA(<axe1>,<axe2>,…)
```

Signification

BRISK :	Instruction d'activation de l'accélération sans limitation des à-coups pour les axes à interpolation
BRISKA :	Instruction d'activation de l'accélération sans limitation des à-coups pour les déplacements d'axes individuels (JOG, JOG/INC, axe de positionnement, axe d'oscillation, etc.)
SOFT :	Instruction d'activation de l'accélération avec limitation des à-coups pour les axes à interpolation
SOFTA :	Instruction d'activation de l'accélération avec limitation des à-coups pour les déplacements d'axes individuels (JOG, JOG/INC, axe de positionnement, axe d'oscillation, etc.)
DRIVE :	Instruction d'activation de l'accélération réduite au-dessus d'une limite de vitesse configurée (PM35220 $MA_ACCEL_REDUCTION_SPEED_POINT) pour les axes à interpolation
DRIVEA :	Instruction d'activation de l'accélération réduite au-dessus d'une limite de vitesse configurée (PM35220 $MA_ACCEL_REDUCTION_SPEED_POINT) pour les déplacements d'axes individuels (JOG, JOG/INC, axe de positionnement, axe d'oscillation, etc.)
(<axe1>,<axe2>,…) :	Axes individuels auxquels le mode d'accélération activé soit s'appliquer

Autres conditions à prendre en compte

Changement de mode d'accélération en cours d'usinage

Un changement de mode d'accélération en cours d'usinage dans un programme pièce (BRISK ↔ SOFT) entraîne un changement de bloc au niveau de la transition avec arrêt précis en fin de bloc même en contournage.

Exemples

Exemple 1 : SOFT et BRISKA

```
Code de programme
N10 G1 X… Y… F900 SOFT
N20 BRISKA(AX5,AX6)
...
```

Exemple 2 : DRIVE et DRIVEA

```
Code de programme
N05 DRIVE
N10 G1 X… Y… F1000
N20 DRIVEA (AX4, AX6)
...
```

Bibliographie

Description fonctionnelle Fonctions de base ; accélération (B2)

12.3.2 Influence de l'accélération dans le cas des axes asservis (VELOLIMA, ACCLIMA, JERKLIMA)

Pour les couplages d'axes (asservissement tangentiel, déplacements conjugués, couplage par valeur pilote, réducteur électronique ; voir "couplages d'axes (Page 581)"), le déplacement des axes/broches asservis s'effectue en fonction d'un ou de plusieurs axes/broches pilotes.

Il est possible d'influencer les limitations dynamiques des axes/broches asservis avec les fonctions VELOLIMA, ACCLIMA et JERKLIMA du programme pièce ou au moyen d'actions synchrones, même lorsque le couplage d'axes est actif.

Remarque

La fonction JERKLIMA n'est pas disponible pour tous les types de couplage.

Bibliographie :

- Description fonctionnelle Fonctions spéciales ; couplages d'axes (M3)
- Description fonctionnelle Fonctions d'extension ; broches synchrones (S3)

Remarque

Disponibilité pour SINUMERIK 828D

Avec SINUMERIK 828D, les fonctions VELOLIMA, ACCLIMA et JERKLIMA peuvent uniquement être utilisées en relation avec la fonction "Déplacements conjugués".

Syntaxe

```
VELOLIMA(<axe>)=<valeur>
ACCLIMA(<axe>)=<valeur>
JERKLIMA(<axe>)=<valeur>
```

Signification

VELOLIMA :	Instruction pour la correction de la **vitesse** maximale programmée
ACCLIMA :	Instruction pour la correction de l'**accélération** maximale programmée
JERKLIMA :	Instruction pour la correction de l'**à-coup** maximum programmé
<axe> :	Axe asservi dont il fait corriger les limitations dynamiques
<valeur> :	Valeur de correction en pourcentage

Exemples

Exemple 1 : correction des limitations dynamiques pour un axe asservi (AX4)

Code de programme	Commentaire
...	
VELOLIMA[AX4]=75	; Correction de limitation à 75 % de la vitesse axiale maximale inscrite dans le paramètre machine.
ACCLIMA[AX4]=50	; Correction de limitation à 50 % de l'accélération axiale maximale inscrite dans le paramètre machine.
JERKLIMA[AX4]=50	; Correction de limitation à 50 % de l'à-coup axial maximum inscrit dans le paramètre machine pour un déplacement avec interpolation.
...	

Exemple 2 : réducteur électronique

L'axe 4 est couplé à l'axe X par un couplage "Réducteur électronique". La capacité d'accélération de l'axe asservi est limitée à 70 % de l'accélération maximale. La vitesse maximale admise est limitée à 50 % de la vitesse maximale. Lorsque le couplage a été réalisé, la vitesse maximale admise est ramenée à 100 %.

Code de programme	Commentaire
...	
N120 ACCLIMA[AX4]=70	; Accélération maximale après réduction.
N130 VELOLIMA[AX4]=50	; Vitesse maximale après réduction.
...	

Code de programme	Commentaire
N150 EGON(AX4,"FINE",X,1,2)	; Réalisation du couplage par réducteur électronique.
...	
N200 VELOLIMA[AX4]=100	; Vitesse maximale restaurée.
...	

Exemple 3 : influencer le couplage par valeur pilote via une action synchrone statique

L'axe 4 est couplé à l'axe X par valeur pilote. L'action synchrone statique 2 limite l'accélération à 80 % à partir de la position 100.

Code de programme	Commentaire
...	
N120 IDS=2 WHENEVER $AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80	; Action synchrone
N130 LEADON(AX4, X, 2)	; Couplage par valeur pilote activé
...	

12.3.3 Activation de valeurs dynamiques spécifiques à la technologie (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)

Au moyen du groupe G "Technologie", il est possible d'activer une dynamique adaptée pour 5 pas d'usinage technologique distincts.

Les valeurs dynamiques et les instructions G sont configurables et dépendent donc des réglages des paramètres machine (→ Constructeur de machine !).

Bibliographie :
Description fonctionnelle Fonctions de base ; contournage, arrêt précis, anticipation (LookAhead) (B1)

Syntaxe

Activation de valeurs dynamiques :
```
DYNNORM
DYNPOS
DYNROUGH
DYNSEMIFIN
DYNFINISH
```

Remarque

Les valeurs dynamiques sont déjà actives dans le bloc dans lequel l'instruction G correspondante est programmée. Il n'y a pas d'arrêt d'usinage.

Lecture ou écriture d'un élément de tableau déterminé :
```
R<m>=$MA...[n,X]
```

```
$MA...[n,X]=<valeur>
```

Signification

DYNNORM :	Fonction G pour l'activation de la **dynamique normale**	
DYNPOS :	Fonction G pour l'activation de la dynamique pour le **mode de positionnement**, **taraudage**	
DYNROUGH :	Fonction G pour l'activation de la dynamique pour l'**ébauche**	
DYNSEMIFIN :	Fonction G pour l'activation de la dynamique pour la **finition**	
DYNFINISH :	Fonction G pour l'activation de la dynamique pour la **superfinition**	
R<m> :	Paramètre de calcul de numéro <m>	
$MA...[n,X] :	Paramètre machine avec élément de tableau déterminant la dynamique	
<n> :	Indice de tableau	
	Plage de valeurs :	0 ... 4
	0	Dynamique normale (DYNNORM)
	1	Dynamique pour le mode positionnement (DYNPOS)
	2	Dynamique pour l'ébauche (DYNROUGH)
	3	Dynamique pour la finition (DYNSEMIFIN)
	4	Dynamique pour la superfinition (DYNFINISH)
<X> :	Adresse d'axe	
<valeur> :	Valeur dynamique	

Exemples

Exemple 1 : activation de valeurs dynamiques

Code de programme	Commentaire
DYNNORM G1 X10	; Initialisation
DYNPOS G1 X10 Y20 Z30 F...	; Mode de positionnement, taraudage
DYNROUGH G1 X10 Y20 Z30 F10000	; Ébauche
DYNSEMIFIN G1 X10 Y20 Z30 F2000	; Finition
DYNFINISH G1 X10 Y20 Z30 F1000	; Superfinition

Exemple 2 : lecture ou écriture d'un élément de tableau déterminé

Accélération maximale pour l'ébauche, axe X.

Code de programme	Commentaire
R1=$MA_MAX_AX_ACCEL[2, X]	; Lecture
$MA_MAX_AX_ACCEL[2, X]=5	; Écriture

12.4 Déplacement avec commande anticipatrice (FFWON, FFWOF)

La commande anticipatrice ramène à zéro la distance de poursuite qui dépend de la vitesse dans un déplacement avec interpolation. La commande anticipatrice permet d'accroître la précision et donc, d'améliorer la qualité de l'usinage.

Syntaxe

```
FFWON
FFWOF
```

Signification

FFWON :	Instruction d'**activation** de la commande anticipatrice
FFWOF	Instruction de **désactivation** de la commande anticipatrice

Remarque

Dans les paramètres machine, on définit le type de commande anticipatrice et les axes à interpolation dont le déplacement devra être effectué avec la commande anticipatrice.

Par défaut : Commande anticipatrice dépendant de la vitesse

Option : Commande anticipatrice dépendant de l'accélération

Exemple

Code de programme

```
N10 FFWON
N20 G1 X… Y… F900 SOFT
```

12.5 Précision de contour programmable (CPRECON, CPRECOF)

La fonction "Précision de contour programmable" permet de réduire les défauts de trajectoire sur les contours incurvés grâce à l'adaptation automatique de la vitesse.

La précision de contour à respecter est spécifiée soit en fonction de la configuration de la machine (PM20470 $MC_MC_CPREC_WITH_FFW ; voir les spécifications du constructeur de machine), soit par la donnée de réglage $SC_CONTPREC, soit par la tolérance de contour programmée CTOL. Plus la valeur et le facteur K_V des axes géométriques sont petits, plus l'avance tangentielle est réduite au niveau des contours incurvés.

Les instructions de programme CN `CPRECON` et `CPRECOF` permettent d'activer et de désactiver la fonction "Précision de contour programmable".

Syntaxe

```
CPRECON
...
CPRECOF
```

Signification

`CPRECON` :	Appel d'instruction G : **activer** la fonction "Précision de contour programmable".	
	Prise d'effet :	modale
`CPRECOF` :	Appel d'instruction G : **désactiver** la fonction "Précision de contour programmable".	
	Prise d'effet :	modale

CPRECON et CPRECOF forment toutes deux le groupe de fonctions G 39 (Précision de contour programmable).

Remarque

Le paramètre de réglage $SC_MINFEED (avance tangentielle minimale pour CPRECON) permet à l'utilisateur de spécifier une vitesse minimale pour l'avance tangentielle.

L'avance ne peut descendre en deçà de cette valeur que si une valeur F plus faible a été programmée ou que les limitations dynamiques des axes nécessitent une vitesse tangentielle inférieure.

Remarque

La fonction "Précision de contour programmable" ne prend en considération que les axes géométriques de la trajectoire. Elle n'a aucune incidence sur les vitesses des axes de positionnement.

Exemple

Code de programme	Commentaire
N10 G0 X0 Y0	
N20 CPRECON	; Activation de la "précision de contour programmable".
N30 G1 G64 X100 F10000	; Usinage à 10 m/min en contournage.
N40 G3 Y20 J10	; Limitation automatique de l'avance dans le bloc à interpolation circulaire.
N50 G1 X0	; Avance de nouveau sans limitation (10 m/min).
...	
N100 CPRECOF	; Désactivation de la "précision de contour programmable".
N110 G0 ...	

Bibliographie

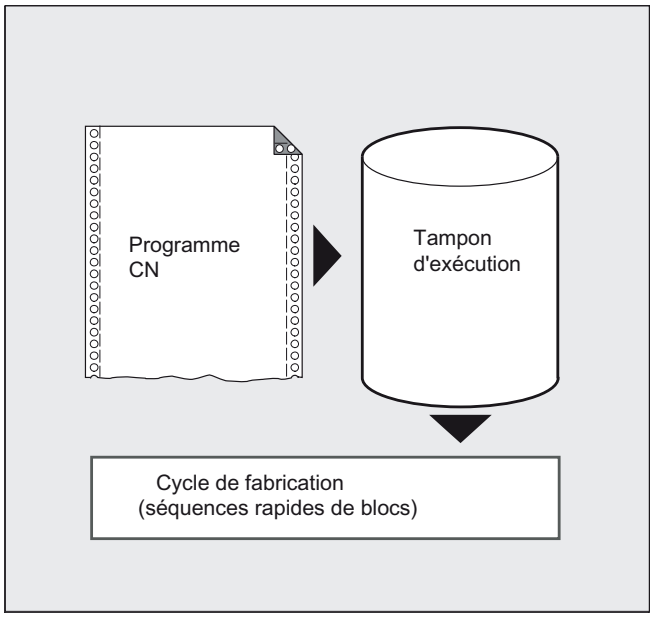
Pour la programmation de CTOL, voir "Programmation d'une tolérance de contour/orientation (CTOL, OTOL, ATOL) (Page 574)"

Pour obtenir des informations détaillées sur la fonction "Précision de contour programmable", voir :

Description fonctionnelle Fonctions spéciales ; Surveillance du contour par fonction tunnel (K6), chapitre : "Précision de contour programmable"

12.6 Exécution du programme avec une mémoire tampon (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

Selon la configuration, la commande dispose d'une mémoire tampon d'une certaine capacité, capable de stocker les blocs prétraités afin de pouvoir ensuite les transmettre à un rythme rapide lors du déroulement de l'usinage. Ainsi, on peut effectuer des trajets courts à grande vitesse. La mémoire tampon est remplie au fur et à mesure que le permet la commande.



Repérage de la phase d'exécution

La section d'exécution à mémoriser temporairement dans la mémoire tampon est repérée au début par "STOPFIFO" et à la fin par "STARTFIFO". L'exécution des blocs prétraités et mémorisés dans la mémoire tampon débute seulement après l'instruction "STARTFIFO" ou quand la mémoire tampon est pleine.

Contrôle automatique de la mémoire tampon

L'instruction "FIFOCTRL" permet d'appeler le contrôle automatique de la mémoire tampon. "FIFOCTRL" agit d'abord comme "STOPFIFO". À chaque programmation, le traitement de la mémoire tampon a lieu seulement lorsqu'elle celle-ci est pleine. Il n'en est pas de même dans le cas de la marche à vide de la mémoire tampon : à partir d'un niveau de remplissage égal à 2/3, la vitesse tangentielle est progressivement réduite avec "FIFOCTRL", afin d'éviter une marche à vide complète et une décélération jusqu'à l'arrêt.

Arrêt du prétraitement des blocs

Le traitement et la mise en mémoire tampon des blocs s'arrêtent lorsque l'instruction "STOPRE" est programmée dans le bloc. Le bloc suivant n'est exécuté que lorsque tous les

blocs prétraités et mémorisés auparavant ont été exécutés en entier. Un arrêt précis (comme G9) a lieu dans le bloc précédent.

IMPORTANT
Interruption de programme
Lorsque la correction d'outil et activée et dans le cas d'interpolations de type spline, il est recommandé de ne pas programmer "STOPRE" afin d'éviter d'interrompre des séquences de blocs contiguës.

Syntaxe

Tableau 12-1 Repérage de la section d'exécution :

```
STOPFIFO
...
STARTFIFO
```

Tableau 12-2 Contrôle automatique de la mémoire tampon :

```
...
FIFOCTRL
...
```

Tableau 12-3 Arrêt du prétraitement des blocs :

```
...
STOPRE
...
```

Remarque

Les instructions "STOPFIFO", "STARTFIFO", "FIFOCTRL" et "STOPRE" doivent être programmées dans un bloc propre.

Signification

STOPFIFO :	"STOPFIFO" repère le début de la section d'exécution à stocker temporairement dans la mémoire tampon. Avec "STOPFIFO", l'exécution est arrêtée et la mémoire tampon remplie jusqu'à ce que : • "STARTFIFO" ou "STOPRE" soit détecté ou • la mémoire tampon soit pleine ou • la fin de programme soit atteinte.
STARTFIFO :	"STARTFIFO" démarre l'exécution rapide de la section de traitement parallèlement au remplissage de la mémoire tampon
FIFOCTRL :	Activation du contrôle automatique de la mémoire tampon
STOPRE :	Arrêter le prétraitement des blocs

Remarque

Le remplissage de la mémoire tampon n'est ni effectué, ni interrompu lorsque la section d'exécution contient des instructions qui nécessitent un fonctionnement sans tampon (accostage du point de référence, fonctions de mesure, ...).

Remarque

Lors de l'accès à des données d'état de la machine ($SA...), la commande déclenche un arrêt du prétraitement des blocs.

Exemple : arrêter le prétraitement des blocs

```
Code de programme                          Commentaire
...
N30 MEAW=1 G1 F1000 X100 Y100 Z50          ; Bloc de mesure avec palpeur de mesure
                                           de la première entrée de mesure et in-
                                           terpolation linéaire.
N40 STOPRE                                 ; Arrêt du prétraitement des blocs.
...
```

12.7 Définition de sections d'arrêt temporisé (DELAYFSTON, DELAYFSTOF)

Les procédures DELAYFSTON et DELAYFSTOF prédéfinies servent à définir une section interruptible sous conditions dans le programme pièce (section d'arrêt temporisé).

Remarque

DELAYFSTON et DELAYFSTOF ne sont **pas** autorisés dans les actions synchrones !

Syntaxe

```
DELAYFSTON
...
DELAYFSTOF
```

Signification

DELAYFSTON :	Définition du début d'une section d'arrêt temporisé	
	Uniquement dans le bloc :	oui
DELAYFSTOF :	Définition de la fin d'une section d'arrêt temporisé	
	Uniquement dans le bloc :	oui

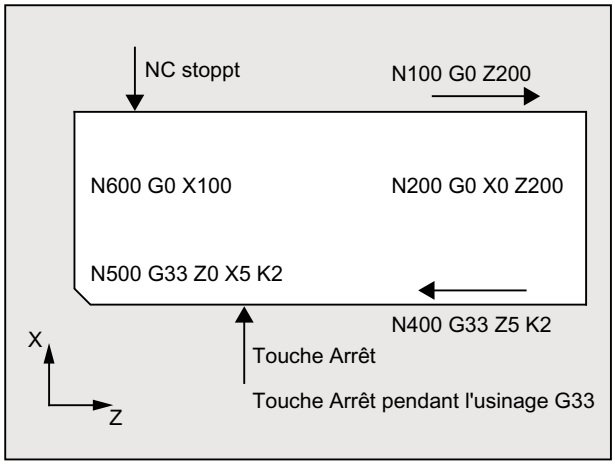
Exemple de programmation

Dans une boucle, le bloc de programme suivant est répété :

Code de programme

```
...
N99 MY_LOOP:
N100 G0 Z200
N200 G0 X0 Z200
N300 DELAYFSTON
N400 G33 Z5 K2 M3 S1000
N500 G33 Z0 X5 K3
N600 G0 X100
N700 DELAYFSTOF
N800 GOTOB MY_LOOP
...
```

On peut voir sur la figure suivante que l'utilisateur appuie sur "Arrêt" dans la section d'arrêt temporisé et que la CN commence l'opération de freinage à l'extérieur de la section d'arrêt temporisé, c'est-à-dire dans le bloc N100. Ainsi, la CN s'arrête au début de N100.



informations complémentaires

Fin de sous-programme

A la fin du sous-programme dans lequel DELAYFSTON a été appelée, l'instruction DELAYFSTOF est activée de manière implicite.

Imbrication

Si le sous-programme 1 appelle le sous-programme 2 dans une section d'arrêt temporisé, la totalité du sous-programme 2 forme une section d'arrêt temporisé. Notamment, l'instruction DELAYFSTOF est sans effet dans le sous-programme 2.

Exemple :

```
Code de programme              Commentaire
N10010 DELAYFSTON              ; Blocs avec N10xxx comme niveau de programme 1.
N10020 R1 = R1 + 1
N10030 G4 F1                   ; Début de la section d'arrêt temporisé.
...
N10040 sous-programme2
...
...                            ; Interprétation du sous-programme 2.
N20010 DELAYFSTON              ; Sans effet, nouveau départ, 2ème niveau.
...
N20020 DELAYFSTOF              ; Sans effet, fin dans un autre niveau.
N20030 RET
N10050 DELAYFSTOF              ; Fin de la section d'arrêt temporisé dans le même ni-
                               veau.
...
N10060 R2 = R2 + 2
N10070 G4 F1                   ; Fin de la section d'arrêt temporisé. Les arrêts s'ac-
                               tivent directement à partir de maintenant.
```

Variables système

Les variables système suivantes permettent d'interroger le système pour savoir si l'exécution du programme pièce se trouve à l'instant donné dans une section d'arrêt temporisé.

- dans le programme pièce avec $P_DELAYFST
- dans les actions synchrones avec $AC_DELAYFST

Valeur	Signification
0	Section d'arrêt temporisé inactive
1	Section d'arrêt temporisé active

12.8 Inhibition d'une partie de programme pour SERUPRO (IPTRLOCK, IPTRUNLOCK)

Pour certaines situations mécaniques complexes, il est nécessaire d'interdire l'opération de recherche de bloc SERUPRO sur la machine.

Avec un pointeur d'interruption programmable, il est possible via la "recherche de points d'interruption", d'intervenir avant la position interdite à la recherche.

Il est également possible de définir des plages interdites à la recherche dans les plages de programme pièce dans lesquelles la CN ne peut pas reprendre le traitement. Avec l'abandon de programme, la CN note le bloc traité en dernier sur lequel peut être effectuée la recherche via l'interface utilisateur IHM.

Syntaxe

```
IPTRLOCK
IPTRUNLOCK
```

Les instructions se trouvent seules sur une ligne du programme pièce et permettent l'utilisation d'un pointeur d'interruption programmable

Signification

`IPTRLOCK` :	Début de la section de programme interdite à la recherche
`IPTRUNLOCK` :	Fin de la section de programme interdite à la recherche

Ces deux instructions ne sont autorisées que dans les programmes pièce et **non pas** dans les actions synchrones.

Exemple

Imbrication de sections de programme interdites à la recherche dans deux niveaux de programmes avec l'instruction implicite "IPTRUNLOCK". L'instruction implicite "IPTRUNLOCK" du sous-programme 1 termine la plage interdite à la recherche.

```
Code de programme                Commentaire
N10010 IPTRLOCK()
N10020 R1 = R1 + 1
N10030 G4 F1                     ; Bloc d'arrêt qui commence la section de programme
                                 interdite à la recherche.
...
N10040 sous-programme2
...                              ; Interprétation du sous-programme 2.
N20010 IPTRLOCK ()               ; Sans effet, nouveau départ.
...
N20020 IPTRUNLOCK ()             ; Sans effet, fin dans un autre niveau.
N20030 RET
...
N10060 R2 = R2 + 2
```

Code de programme	Commentaire
N10070 RET	; Fin de la section de programme interdite à la recherche.
N100 G4 F2	; Le programme principal reprend.

Le pointeur d'interruption fournit alors à nouveau une interruption sur 100.

Informations complémentaires

Saisie et recherche de plages interdites à la recherche

Les sections de programme interdites à la recherche sont identifiées à l'aide des instructions "IPTRLOCK" et "IPTRUNLOCK".

L'instruction "IPTRLOCK" gèle le pointeur d'interruption sur un mode bloc par bloc exécutable (SBL1). Ce bloc est désigné dans ce qui suit par "bloc d'arrêt". Si une interruption de programme se produit après "IPTRLOCK", la recherche est possible sur l'interface utilisateur IHM après ce bloc d'arrêt.

Reprise sur le bloc courant

Avec "IPTRUNLOCK", le pointeur d'interruption est placé au point d'interruption sur le bloc courant pour la section de programme suivante.

Après qu'une destination de recherche a été trouvée, il est possible de répéter une nouvelle destination de recherche avec le même bloc d'arrêt.

Un pointeur d'interruption édité par l'utilisateur doit toujours être supprimé via HMI.

Règles relatives aux imbrications

Les points suivants régissent l'interaction des instructions "IPTRLOCK" et "IPTRUNLOCK" avec les imbrications et la fin de sous-programme :

1. A la fin du sous-programme dans lequel "IPTRLOCK" a été appelée, l'instruction "IPTRUNLOCK" est activée de manière implicite.
2. Dans une plage interdite à la recherche, "IPTRLOCK" reste sans effet.
3. Si le sous-programme1 appelle le sous-programme2 dans une plage interdite à la recherche, le sous-programme2 reste complètement interdit à la recherche. Notamment, l'instruction "IPTRUNLOCK" est sans effet dans le sous-programme2.

Pour de plus amples informations voir
/FB1/ Manuel de fonctions de base ; GMF, Canal, Mode de programme (K1).

Variable système

La présence d'une plage interdite à la recherche peut être déterminée à l'aide de l'instruction "$P_IPTRLOCK" dans le programme pièce.

Pointeur d'interruption automatique

La fonction de pointeur d'interruption automatique définit automatiquement comme interdit à la recherche un type de couplage précédemment défini. Le paramètre machine active le pointeur d'interruption automatique pour

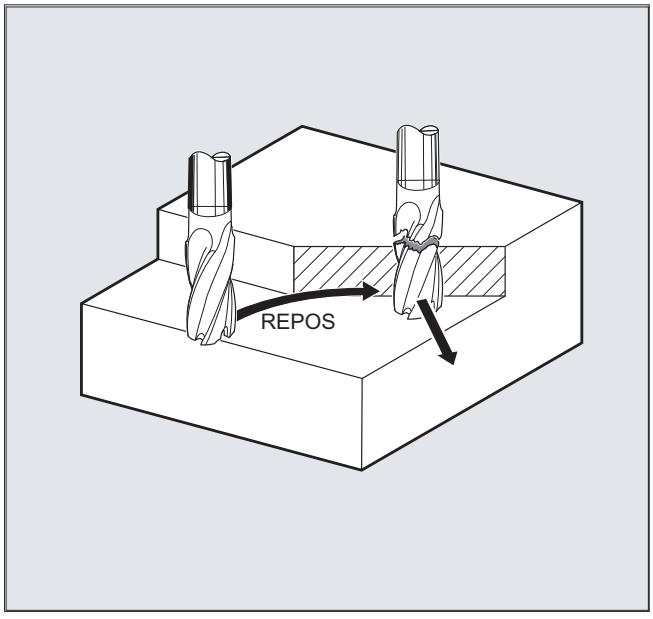
- Réducteur électronique pour "EGON"
- Couplage des axes par valeur pilote pour "LEADON".

Si le pointeur d'interruption programmé et le pointeur d'interruption automatique se chevauchent, la plage interdite à la recherche la plus grande possible est créée.

12.9 Réaccostage du contour (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Si vous stoppez le programme en cours d'usinage et que vous dégagez l'outil, par exemple à cause d'un bris d'outil ou parce que vous voulez effectuer une mesure, vous pouvez par la suite réaccoster, à l'aide du programme, le contour à un point de votre choix.

La commande `REPOS` a le même effet dans un ASUP qu'un retour de sous-programme (par exemple M17). Les blocs suivants ne sont plus exécutés. Pour de plus amples informations sur l'interruption de l'exécution d'un programme, voir aussi "Routine d'interruption (ASUP) (Page 134)".



Syntaxe

```
REPOSA RMIBL DISPR=…
REPOSA RMBBL
REPOSA RMEBL
REPOSA RMNBL
REPOSL RMIBL DISPR=…
REPOSL RMBBL
REPOSL RMEBL
REPOSL RMNBL
REPOSQ RMIBL DISPR=… DISR=…
REPOSQ RMBBL DISR=…
REPOSQ RMEBL DISR=…
REPOSQA DISR=…
REPOSH RMIBL DISPR=… DISR=…
REPOSH RMBBL DISR=…
REPOSH RMEBL DISR=…
REPOSHA DISR=…
```

Signification

Sélection du trajet d'accostage

`REPOSA` :	Réaccostage du contour avec les axes géométriques sur une droite. Tous les autres axes de canal sont également repositionnés.
`REPOSL` :	Réaccostage du contour avec les axes géométriques sur une droite. Tous les autres axes de canal doivent être programmés explicitement.
`REPOSQ DISR=…` :	Réaccostage du contour avec les axes géométriques sur un quart de cercle avec rayon DISR. Tous les autres axes de canal doivent être programmés explicitement.
`REPOSQA DISR=…` :	Réaccostage du contour avec les axes géométriques sur un quart de cercle avec rayon DISR. Tous les autres axes de canal sont également repositionnés.
`REPOSH DISR=…` :	Réaccostage du contour avec les axes géométriques sur un demi-cercle avec diamètre DISR. Tous les autres axes de canal doivent être programmés explicitement.
`REPOSHA DISR=…` :	Réaccostage du contour avec les axes géométriques sur un demi-cercle avec rayon DISR. Tous les autres axes de canal sont également repositionnés.

Sélection du point de réaccostage

`RMIBL` :	Accostage du point d'interruption
`RMIBL DISPR=…` :	Point d'entrée à une distance DISPR en mm/inch avant le point d'interruption
`RMBBL` :	Accostage du point de début de bloc
`RMEBL` :	Accostage du point de fin de bloc
`RMEBL DISPR=…` :	Accostage du point de fin de bloc à une distance DISPR avant le point final
`RMNBL` :	Accostage au point de contour suivant
`A0 B0 C0` :	Axes dans lesquels le déplacement doit avoir lieu

Remarque

Compatibilité

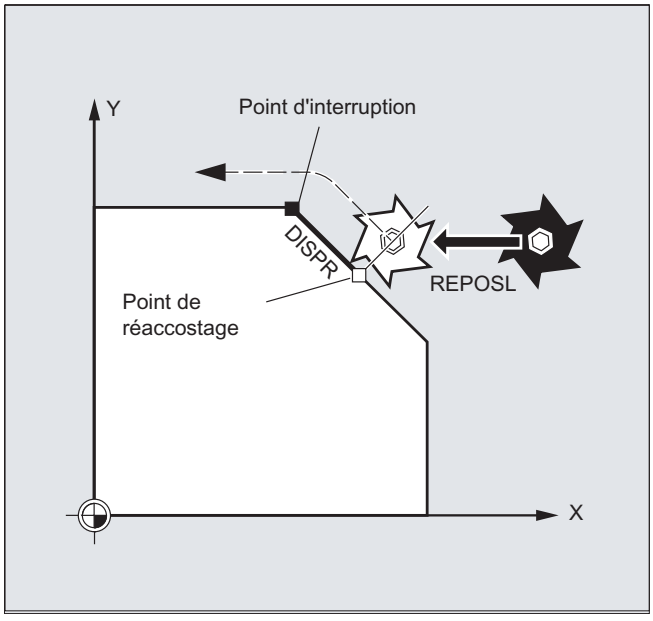
Pour préserver la compatibilité avec des versions logicielles antérieures, le mode d'accostage `REPOS` peut toujours être programmé au moyen des instructions G modales `RMI`, `RMB`, `RME` et `RMN`. Lors d'une utilisation au sein d'un ASUP, celui-ci doit être doté d'un attribut `SAVE` dans l'instruction `PROC`. Dans le cas contraire, le mode d'accostage `REPOS` modal, s'il diffère du préréglage de `RMI`, aurait une incidence sur des actions `REPOS` ultérieures.

Réaccostage du contour sur une droite, REPOSA, REPOSL

L'outil rallie le point de réaccostage sur une ligne droite.

Exemple

```
REPOSL RMIBL DISPR=6 F400
```

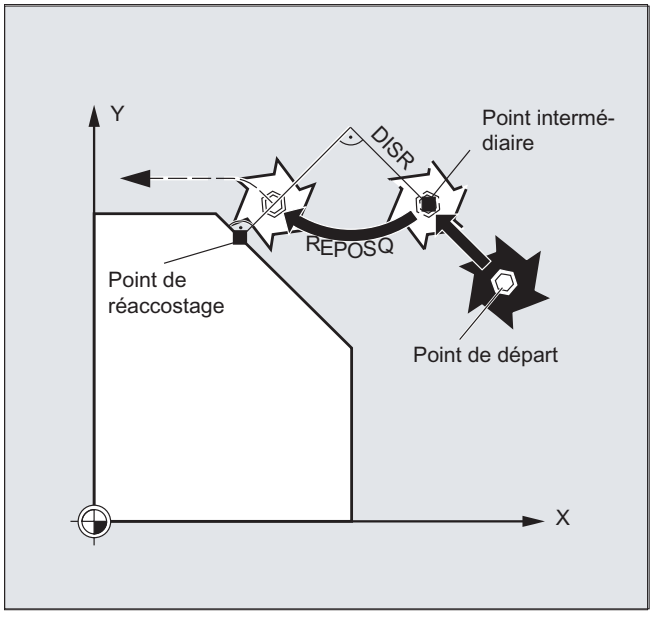


Réaccostage du contour avec quart de cercle, REPOSQ, REPOSQA

L'outil rallie le point de réaccostage sur un quart de cercle de rayon`DISR=...`. La commande calcule automatiquement le point intermédiaire nécessaire entre le point de départ et le point de réaccostage.

Exemple

```
REPOSQ RMIBL DISR=10 F400
```

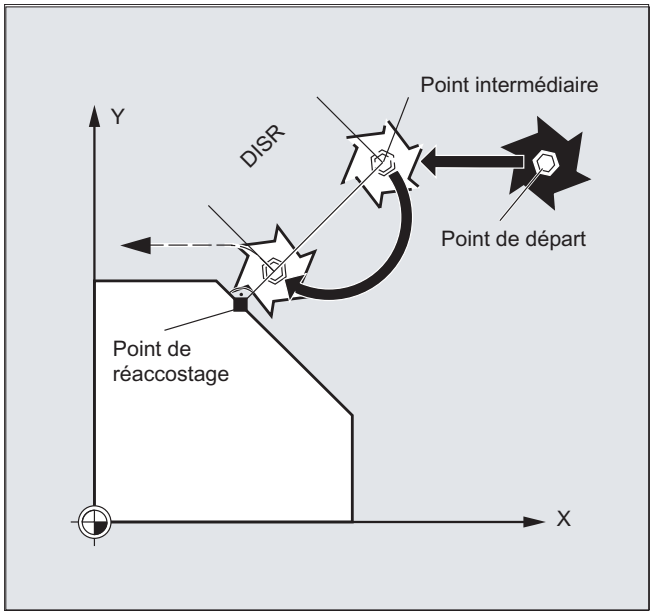


Réaccostage du contour avec demi-cercle, REPOSH, REPOSHA

L'outil rallie le point de réaccostage sur un demi-cercle de diamètre `DISR=...`. La commande calcule automatiquement le point intermédiaire nécessaire entre le point de départ et le point de réaccostage.

Exemple

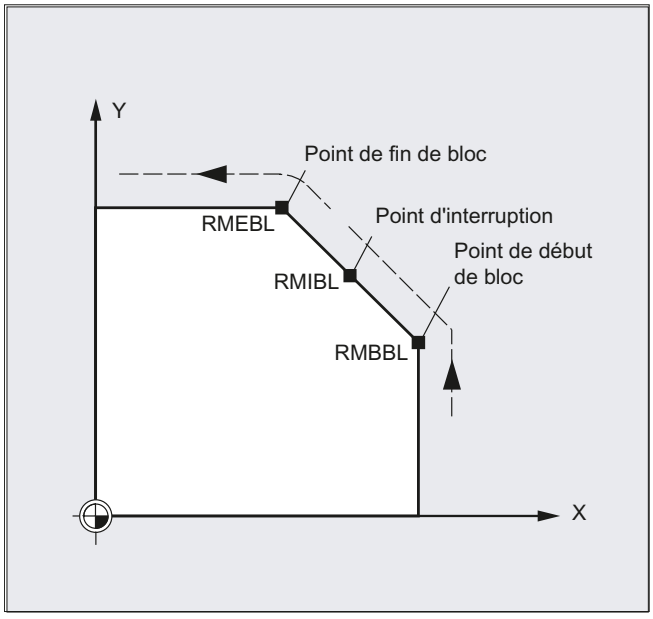
```
REPOSH RMIBL DISR=20 F400
```



Définition d'un point de réaccostage (sauf pour l'accostage SERUPRO sous RMNBL)

En fonction du bloc CN dans lequel l'interruption a eu lieu, vous avez le choix entre trois points de réaccostage :

- RMIBL, point d'interruption
- RMBBL, point de début de bloc ou dernier point final
- RMEBL, point de fin de bloc



Avec RMIBL DISPR=... ou RMEBL DISPR=..., vous pouvez définir un point de réaccostage se trouvant avant le point d'interruption ou le point de fin de bloc.

Avec DISPR=..., vous programmez, en mm/inch, la distance sur le contour entre le point de réaccostage et le point d'interruption ou le point de fin de bloc. Ce point ne peut pas figurer au-delà du point de début de bloc – même pour les plus grandes valeurs.

Si DISPR=... n'est pas programmé, DISPR=0 s'applique, c'est-à-dire que le réaccostage a lieu au point d'interruption (pour RMIBL) ou au point de fin de bloc (pour RMEBL).

Signe de DISPR

Le signe de DISPR est exploité. Si le signe est positif, le comportement est le même que précédemment.

Si le signe est négatif, le réaccostage a lieu derrière le point d'interruption ou, pour RMBBL, derrière le point de départ.

La distance point d'interruption-point de réaccostage est la valeur absolue de DISPR. Ce point ne peut pas se trouver au-delà du point de fin de bloc, même pour les plus grandes valeurs absolues.

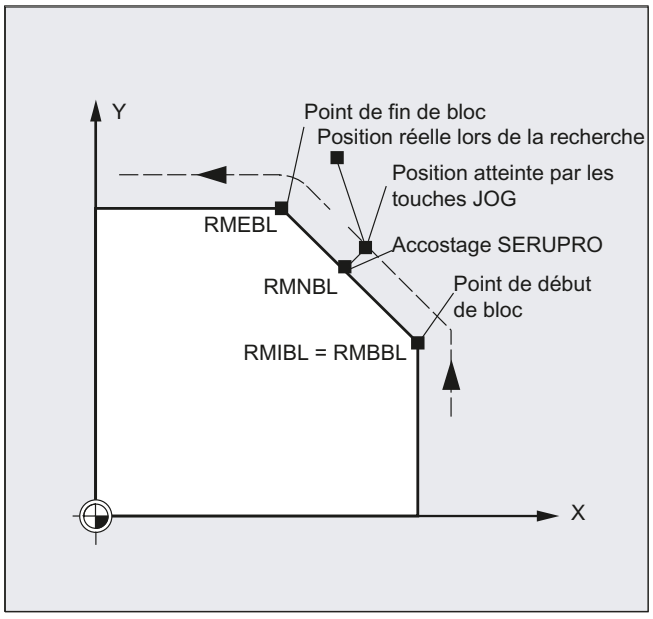
Exemple d'application :

Un capteur surveille les abords d'une griffe de serrage. Une ASUP permettant de contourner la griffe de serrage est déclenchée.

Ensuite, un repositionnement à un point situé derrière la griffe de serrage avec DISPR négatif a lieu et le programme se poursuit.

Accostage SERUPRO sous RMNBL

Si, lors de l'usinage, une interruption a été forcée sur une position quelconque, la distance la plus courte depuis la position d'interruption est parcourue avec l'accostage SERUPRO sous RMNBL afin de n'exécuter ensuite que la distance restant à parcourir. L'utilisateur lance une opération SERUPRO sur le bloc d'interruption et se positionne, à l'aide des touches JOG, avant la position défectueuse du bloc recherché.



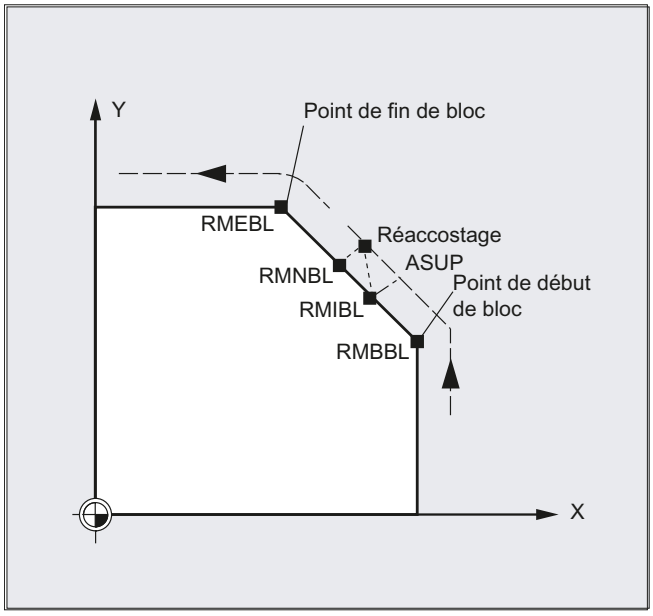
Remarque

SERUPRO

Pour SERUPRO, RMIBL et RMBBL sont identiques. RMNBL n'est pas seulement limité à SERUPRO, il est valide de manière générale.

Accostage au point de contour RMNBL suivant

Au moment de l'interprétation de REPOSA après une interruption, le réaccostage avec RMNBL n'a pas encore complètement commencé, seule la distance restant à parcourir a été traitée. C'est le point de contour suivant du bloc interrompu qui est réaccosté.



Etat du mode REPOS valide

Le mode REPOS valide du bloc interrompu peut être lu via des actions synchrones avec la $AC_REPOS_PATH_MODE :

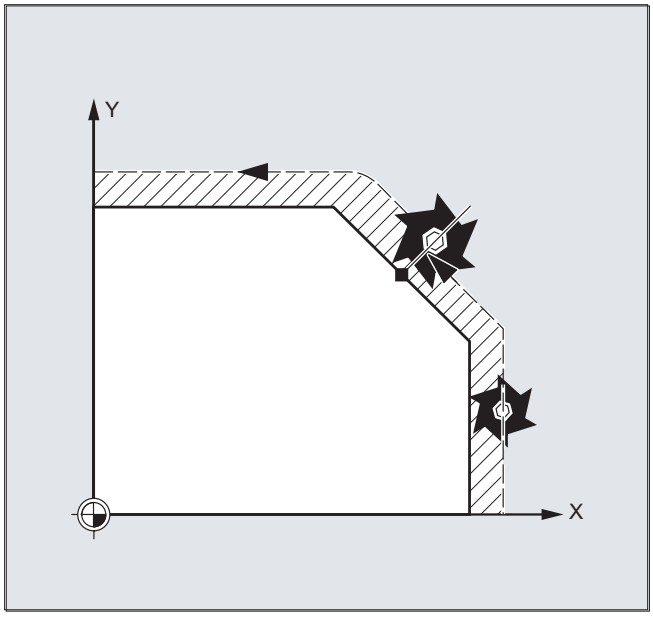
0 Accostage non défini
1 RMBBL : accostage au début
2 RMIBL : accostage au point d'interruption
3 RMEBL : accostage au point de fin de bloc
4 RMNBL : accostage au point de contour suivant du bloc interrompu

accostage avec un nouvel outil

Si vous avez stoppé l'exécution du programme suite à un bris d'outil :

En programmant le nouveau numéro D, vous poursuivez l'exécution du programme à partir du point de réaccostage et avec des valeurs de correction d'outil modifiées.

Suite aux différentes valeurs de correction d'outil, le point d'interruption risque de ne pas pouvoir être réaccosté. Dans ce cas, c'est le point le plus proche du point d'interruption qui devient point de réaccostage du contour (le cas échéant avec une modification DISPR).



Accostage du contour

Le déplacement à effectuer pour réaccoster le contour est programmable. Indiquez zéro pour les adresses des axes à déplacer.

Avec les instructions REPOSA, REPOSQA et REPOSHA, tous les axes sont repositionnés automatiquement. Il n'est pas nécessaire de préciser les axes.

Quand on programme REPOSL, REPOSQ et REPOSH, tous les axes géométriques se déplacent automatiquement, autrement dit sans qu'il soit nécessaire de le préciser dans l'instruction. Tous les autres axes à repositionner sont à préciser dans l'instruction.

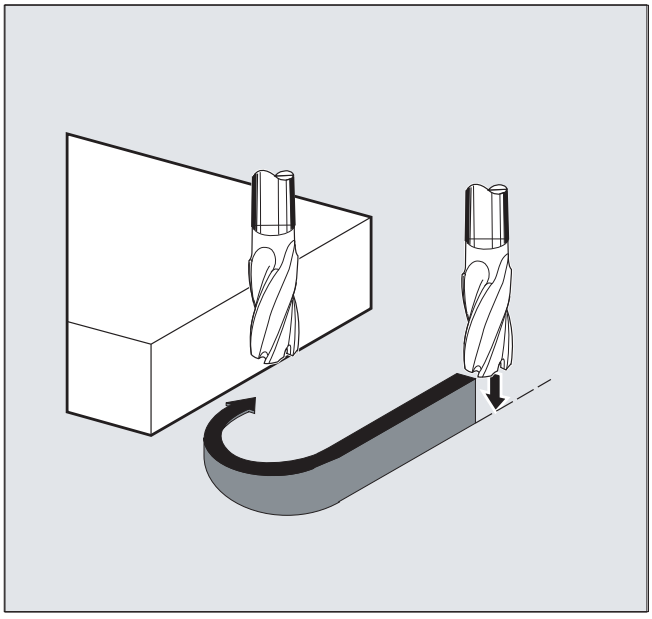
Les points suivants s'appliquent pour les mouvements circulaires REPOSH et REPOSQ :

Le mouvement circulaire est décrit dans le plan de travail G17 à G19 indiqué.

Si vous indiquez le troisième axe géométrique (pénétration) dans le bloc d'accostage et si la position de l'outil et la position programmée dans la direction de pénétration ne concordent pas, le réaccostage du contour se fera suivant un mouvement hélicoïdal.

Dans les cas suivants, il y a commutation automatique sur accostage linéaire REPOSL :

- Vous n'avez indiqué aucune valeur pour DISR.
- Il n'existe aucun sens d'accostage défini (interruption de l'exécution du programme dans un bloc ne contenant aucune information de déplacement).
- Le sens d'accostage est perpendiculaire au plan de travail courant.



12.10 Correction du pilotage des déplacements

12.10.1 Correction de l'à-coup en pourcentage (JERKLIM)

L'instruction CN "JERKLIM" permet de réduire ou d'augmenter l'à-coup maximal possible paramétré par un paramètre machine lors de déplacements avec interpolation dans des sections critiques d'un programme.

Condition

Il est nécessaire que le mode d'accélération SOFT soit activé.

Prise d'effet

La fonction prend effet :

- dans les modes de fonctionnement AUTOMATIQUE,
- uniquement sur des axes à interpolation.

Syntaxe

```
JERKLIM[<axe>]=<valeur>
```

Signification

`JERKLIM` :	Instruction de correction d'à-coup	
`<axe>` :	Axe machine dont la valeur limite d'à-coup doit être adaptée.	
`<valeur>` :	Valeur de correction en pourcentage, en rapport à l'à-coup maximal configuré de l'axe lors d'un déplacement avec interpolation (PM32431 $MA_MAX_AX_JERK).	
	Plage de valeurs :	1 ... 200
	La valeur 100 n'entraîne aucune correction de l'à-coup.	

Remarque

Le comportement de JERKLIM à la fin du programme pièce ou lors d'un Reset de canal est configuré avec le bit 0 dans le paramètre machine
PM32320 $MA_DYN_LIMIT_RESET_MASK :

- Bit 0 = 0 :
  La valeur programmée pour JERKLIM est remise à 100 % avec un Reset de canal/M30.
- Bit 0 = 1 :
  La valeur programmée pour JERKLIM reste inchangée même après un Reset de canal/M30.

Exemple

Code de programme	Commentaire
...	
N60 JERKLIM[X]=75	; Dans la direction X, le chariot de déplacement axial ne doit accélérer/décélérer qu'avec 75 % au maximum de l'à-coup autorisé pour l'axe.
...	

12.10.2 Correction de la vitesse en pourcentage (VELOLIM)

L'instruction VELOLIM permet, dans le programme pièce ou une action synchrone, de réduire la vitesse maximale possible d'un axe ou la vitesse de rotation maximale possible d'une broche en fonction du rapport de transmission réglée au moyen d'un paramètre machine.

Prise d'effet

La fonction prend effet :

- dans les modes de fonctionnement AUTOMATIQUE,
- sur des axes à interpolation et de positionnement,
- sur des broches en mode broche / mode axe.

Syntaxe

```
VELOLIM[<Axe/Broche>]=<Valeur>
```

Signification

VELOLIM :	Instruction de correction de vitesse
<axe/broche> :	Axe ou broche dont la valeur limite de vitesse doit être adaptée. **VELOLIM pour broches** Le paramètre machine (PM30455 $MA_MISC_FUNCTION_MASK, Bit 6) permet de configurer dans le programme pièce si "VELOLIM" prend effet indépendamment de son utilisation actuelle en tant que broche ou axe (bit 6 = 1) ou bien s'il doit pouvoir être programmé individuellement pour chaque mode de fonctionnement (bit 6 = 0). Si un effet distinct est configuré, la sélection a lieu lors de la programmation au moyen du descripteur : • Descripteur de broche S<n> pour modes broche • Descripteur d'axe, par ex. "C", pour mode axe

<valeur> :	Valeur de correction en pourcentage La valeur de correction se rapporte : • Pour des axes/broches en mode axe (PM30455 Bit 6 = 0) : à la vitesse d'axe maximale configurée (PM32000 $MA_MAX_AX_VELO). • Pour des broches en mode broche ou axe (PM30455 Bit 6 = 1) : à la vitesse de rotation maximale du rapport de transmission actif (PM35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])	
	Plage de valeurs :	1 ... 100
	La valeur 100 n'entraîne aucune correction de la vitesse ni de la vitesse de rotation.	

Remarque

Comportement à la fin du programme pièce et lors d'un Reset de canal

Le comportement de "VELOLIM" à la fin du programme pièce ou lors d'un Reset de canal peut être réglé au moyen du paramètre machine : PM32320 $MA_DYN_LIMIT_RESET_MASK, bit 0

Détection d'une limitation de vitesse active en mode broche

Une limitation de vitesse activée par "VELOLIM" (moins de 100 %) peut être détectée en mode broche au moyen de la variable système suivante :

- $AC_SMAXVELO (vitesse de rotation maximale possible de la broche)
- $AC_SMAXVELO_INFO (identificateur de la cause de la limitation de vitesse)

Exemples

Exemple 1 : limitation de vitesse d'un axe machine

```
Code de programme                Commentaire
...
N70 VELOLIM[X]=80                ; Dans la direction X, le chariot de déplacement axial
                                 ne doit se déplacer qu'au maximum à 80 % de la vitesse
                                 autorisée pour l'axe.
...
```

Exemple 2 : limitation de vitesse de rotation d'une broche

```
Code de programme                Commentaire
N05 VELOLIM[S1]=90               ; Limitation de la vitesse de rotation maximale de la
                                 broche 1 à 90 % de 1000 tr/min.
...
N50 VELOLIM[C]=45                ; Limitation de la vitesse de rotation à 45 % de 1000 tr/
                                 min, C étant le descripteur d'axe de S1.
...
```

Réglages des paramètres machine pour broche 1 (AX5)

- Vitesse de rotation maximale du rapport de transmission 1 = 1000 tr/min :
  PM35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
- La programmation de "VELOLIM" prend effet en même temps pour le mode broche et le mode axe, indépendamment du descripteur programmé :
  PM30455 $MA_MISC_FUNCTION_MASK[AX5], Bit 6 = 1

12.10.3 Exemple de programme pour JERKLIM et VELOLIM

Le programme suivant représente un exemple d'application pour une limitation proportionnelle de l'à-coup et de la vitesse :

Code de programme	**Commentaire**
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20	
ACC[Y]=30	
N1200 G1 Y20 VELOLIM[X]=5	; Dans la direction X, le chariot doit se déplacer au maximum à 5 % de la vitesse autorisée pour l'axe.
JERKLIM[Y]=200	; Dans la direction X, le chariot peut accélérer/décélérer au maximum avec 200 % de l'à-coup autorisé pour l'axe.
N1300 G1 X0 JERKLIM[X]=2	; Dans la direction X, le chariot doit accélérer/décélérer au maximum avec 2 % de l'à-coup autorisé pour l'axe.
N1400 G1 Y0	
M30	

12.11 Programmation d'une tolérance de contour/orientation (CTOL, OTOL, ATOL)

Les adresses CTOL, OTOL et ATOL permettent d'adapter les tolérances d'usinage paramétrées dans les paramètres machine et les données de réglage pour les fonctions de compresseur, l'arrondissement et le lissage d'orientation dans le programme pièce.

Les valeurs de tolérance programmées s'appliquent jusqu'à ce qu'elles soient reprogrammées ou supprimées par l'affectation d'une valeur négative. Elles sont en outre supprimées à la fin du programme ou lors d'un reset. Après la suppression, les valeurs de tolérance paramétrées redeviennent effectives.

Syntaxe

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

Signification

`CTOL` :	Adresse pour la programmation de la **tolérance de contour**			
	Domaine d'application :	• toutes les fonctions de compresseur • tous les modes d'arrondissement, à l'exception de G641 et G644		
	Arrêt du prétraitement des blocs :	Non		
	Prise d'effet :	Modale		
	`<Value>` :	La valeur de la tolérance de contour est exprimée par une cote.		
		Type :	REAL	
		Unité :	inch/mm (en fonction de la cotation actuellement réglée)	
		Plage de valeurs :	≥ 0 :	Valeur de tolérance
			< 0 :	Suppression de la valeur de tolérance programmée ⇒ La valeur de tolérance paramétrée dans le paramètre machine ou les données de réglage reprend effet.

OTOL :	Adresse pour la programmation de la **tolérance d'orientation**			
	Domaine d'application :	• toutes les fonctions de compresseur • au lissage de l'orientation ORISON • tous les modes d'arrondissement, à l'exception de G641, G644 et OSD		
	Arrêt du prétraitement des blocs :	Non		
	Prise d'effet :	Modale		
	<Value> :	La valeur de la tolérance d'orientation est exprimée par une cote angulaire.		
		Type :	REAL	
		Unité :	degré	
		Plage de valeurs :	≥ 0 :	Valeur de tolérance
			< 0 :	Suppression de la valeur de tolérance programmée ⇒ La valeur de tolérance paramétrée dans le paramètre machine ou les données de réglage reprend effet.
ATOL :	Adresse pour la programmation d'une **tolérance spécifique à l'axe**			
	Domaine d'application :	• toutes les fonctions de compresseur • au lissage de l'orientation ORISON • tous les modes d'arrondissement, à l'exception de G641, G644 et OSD		
	Arrêt du prétraitement des blocs :	Non		
	Prise d'effet :	Modale		
	<Axis> :	Nom de l'axe de canal pour lequel la tolérance programmée doit s'appliquer		
	<Value> :	Selon le type d'axe (linéaire ou rotatif), la valeur de la tolérance de l'axe est exprimée par une cote ou une cote angulaire.		
		Type :	REAL	
		Unité :	pour axes linéaires :	inch/mm (en fonction de la cotation actuellement réglée)
			pour axes rotatifs :	degré
		Plage de valeurs :	≥ 0 :	Valeur de tolérance
			< 0 :	Suppression de la valeur de tolérance programmée ⇒ La valeur de tolérance paramétrée dans le paramètre machine ou les données de réglage reprend effet.

Remarque

Les valeurs de tolérance CTOL et OTOL programmées spécifiques au canal ont une priorité plus élevée que les valeurs de tolérance ATOL programmées spécifiques à l'axe.

Remarque

Frames mis à l'échelle

Les frames mis à l'échelle agissent sur la tolérance programmée de la même manière que sur les positions des axes, c'est-à-dire que la tolérance relative reste identique.

Exemple

Code de programme	Commentaire
COMPCAD G645 G1 F10000	; Activer la fonction de compresseur COMPCAD.
X... Y... Z...	; Ici, les paramètres machine et données de réglage sont actifs.
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; A partir d'ici, une tolérance de contour de 0,02 mm est active.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
ASCALE X0.25 Y0.25 Z0.25	; A partir d'ici, une tolérance de contour de 0,005 mm est active.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=-1	; A partir d'ici, les paramètres machine et données de réglage sont de nouveau actifs.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Variables système

Lecture avec arrêt du prétraitement des blocs

Les tolérances actuellement actives peuvent être lues dans le programme pièce et l'action synchrone par le biais des variables système suivantes :

- $AC_CTOL
  Tolérance de contour spécifique au canal, qui était effective lors du traitement du bloc principal actuel.
  Lorsqu'aucune tolérance de contour n'est effective, $AC_CTOL fournit la racine de la somme des carrés des tolérances appliquées aux axes géométriques.
- $AC_OTOL
  Tolérance d'orientation spécifique au canal, qui était effective lors du traitement du bloc principal actuel.
  Lorsqu'aucune tolérance d'orientation n'est effective, $AC_OTOL fournit la racine de la somme des carrés des tolérances appliquées aux axes d'orientation lorsqu'une transformation d'orientation est active, sinon la valeur "-1".
- $AA_ATOL[<axe>]
  Tolérance de contour spécifique à l'axe, qui était effective lors du traitement du bloc principal actuel.
  Lorsqu'une tolérance de contour est effective, $AA_ATOL[<axe géométrique>] fournit la tolérance de contour divisée par la racine du nombre d'axes géométriques.
  Lorsqu'une tolérance et une transformation d'orientation sont effectives, $AA_ATOL[<axe d'orientation>] fournit la tolérance d'orientation divisée par la racine du nombre d'axes d'orientation.

Remarque

Lorsqu'aucune valeur de tolérance n'a été programmée, les variables $A ne sont pas suffisamment différenciées pour permettre une distinction entre les tolérances de chacune des fonctions.

De tels cas sont susceptibles de se produire lorsque les paramètres machine et les données de réglage définissent des tolérances différentes pour les fonctions de compresseur, l'arrondissement et le lissage de l'orientation. Les variables système fournissent alors la valeur la plus grande des fonctions en cours. Lorsqu'une fonction de compresseur est p. ex. active avec une tolérance d'orientation égale à 0,1° et un lissage de l'orientation ORISON égal à 1°, $AC_OTOL fournit la valeur "1". Si l'on désactive le lissage de l'orientation, $AC_OTOL fournit la valeur "0,1".

Lecture sans arrêt du prétraitement des blocs

Les tolérances actuellement effectives peuvent être lues dans le programme pièce par le biais des variables système suivantes :

- $P_CTOL
  Tolérance de contour spécifique au canal actuellement effective.
- $P_OTOL
  Tolérance d'orientation spécifique au canal actuellement effective.
- $PA_ATOL
  Tolérance de contour spécifique à l'axe actuellement effective.

Autres conditions

Les tolérances programmées avec CTOL, OTOL et ATOL agissent également sur des fonctions qui dépendent indirectement de ces tolérances :

- La limitation de l'erreur cordale lors du calcul de la consigne
- Les fonctions de base du mode formes libres

Les fonctions d'arrondissement suivantes ne sont **pas** influencées par la programmation de CTOL, OTOL et ATOL :

- Arrondissement de l'orientation avec OSD
  OSD n'utilise pas de tolérance mais une distance par rapport à la transition entre blocs.
- Arrondissement avec G644
  G644 ne sert pas pour l'usinage mais pour l'optimisation des changements d'outil et d'autres déplacements sans usinage.
- Arrondissement avec G645
  G645 se comporte presque toujours comme G642 et utilise ainsi les tolérances programmées. La valeur de tolérance issue du paramètre machine PM33120 $MA_PATH_TRANS_POS_TOL n'est utilisée que pour les transitions tangentielles entre blocs avec un saut de courbure, par ex. une transition tangentielle cercle-droite. À ces emplacements, la trajectoire d'arrondissement peut aussi se trouver sur l'extérieur du contour programmé, où de nombreuses applications sont moins tolérantes. De plus, une petite tolérance réglée de manière fixe est en général suffisante pour compenser les variations de courbure, dont le programmateur CN n'a pas à se soucier.

12.12 Mode de changement de bloc pour couplage actif (CPBC)

L'instruction CPBC détermine que le critère de changement de bloc doit être rempli pour qu'un changement de bloc puisse avoir lieu dans le programme pièce lorsque le couplage est actif.

Syntaxe

```
CPBC[<axe asservi>] = <critère>
```

Signification

CPBC :	Critère de changement de bloc pour couplage actif	
<axe asservi> :	Descripteur de l'axe asservi	
<critère> :	Critère de changement de bloc	
	Type :	STRING
	Valeur	**Signification :** Le changement de bloc s'effectue
	"NOC"	indépendamment de l'état du couplage
	"IPOSTOP"	à la valeur de consigne du synchronisme
	"COARSE"	lors du synchronisme "grossier" côté valeur réelle
	"FINE"	lors du synchronisme "fin" côté valeur réelle

Exemple

Code de programme

```
; le changement de bloc s'effectue pour :
;  - couplage avec l'axe asservi X2 == actif
;  - synchronisme côté consigne == actif
CPBC[X2]="IPOSTOP"
```

couplages d'axes

13

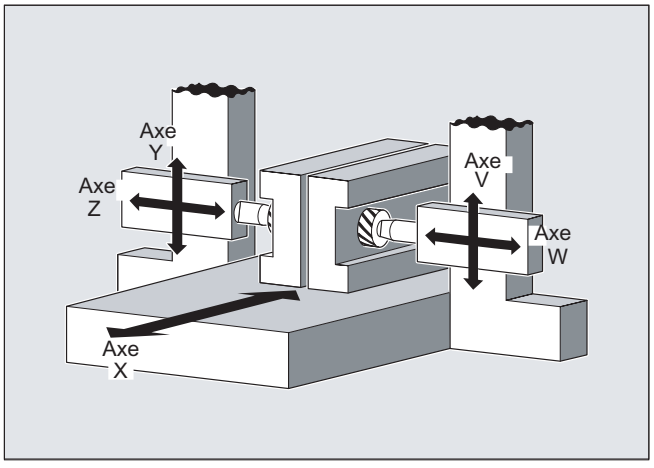
13.1 Déplacements conjugués (TRAILON, TRAILOF)

Lors du déplacement d'un axe pilote défini, les axes conjugués qui lui sont affectés (= axes asservis) décrivent les courses qui découlent de ce déplacement, compte tenu d'un facteur de couplage.

L'axe pilote et les axes asservis constituent un groupe d'axes à déplacements conjugués.

Domaines d'application

- Déplacement d'un axe par le biais d'un autre axe simulé. L'axe pilote est un axe simulé et l'axe conjugué un axe réel. Ainsi, l'axe réel peut être déplacé en tenant compte du facteur de couplage.
- Usinage bilatéral avec 2 groupe d'axes à déplacements conjugués :
  1. Axe pilote Y, axe conjugué V
  2. Axe pilote Z, axe conjugué W



Syntaxe

```
TRAILON(<axe asservi>,<axe pilote>,<facteur de couplage>)
TRAILOF(<axe asservi>,<axe pilote>,<axe pilote 2>)
TRAILOF(<axe asservi>)
```

Signification

TRAILON :	Instruction pour l'activation et la définition d'un groupe d'axes à déplacements conjugués	
	Prise d'effet :	modale

`<Axe asservi>` :	Paramètre 1 : désignation de l'axe conjugué (asservi) **Remarque :** un axe conjugué peut également être l'axe pilote d'autres axes conjugués. Ceci permet de réaliser des groupes d'axes à déplacements conjugués interdépendants.
`<Axe pilote>` :	Paramètre 2 : désignation de l'axe pilote
`<Facteur de couplage>` :	Paramètre 3 : facteur de couplage Le facteur de couplage indique le rapport souhaité entre la course de l'axe conjugué et celle de l'axe pilote : <facteur de couplage> = course de l'axe conjugué/course de l'axe pilote
	Type : REAL
	Réglage par défaut : 1
	En entrant une valeur négative, les déplacements de l'axe pilote et de l'axe conjugué s'effectuent en sens opposé. Si le facteur de couplage n'est pas indiqué lors de la programmation, le facteur de couplage 1 est pris en compte automatiquement.
`TRAILOF` :	Instruction de désactivation d'un groupe d'axes à déplacements conjugués
	Prise d'effet : modale

	`TRAILOF` avec 2 paramètres désactive uniquement le couplage à l'axe pilote indiqué :
	`TRAILOF(<axe asservi>,<axe pilote>)`
	Si un axe conjugué possède 2 axes pilotes, `TRAILOF` peut être appelé avec 3 paramètres pour désactiver les deux couplages :
	`TRAILOF(<axe asservi>,<axe pilote>,<axe pilote 2>)`
	La programmation de `TRAILOF` sans indication d'axe pilote fournit le même résultat :
	`TRAILOF(<axe asservi>)`

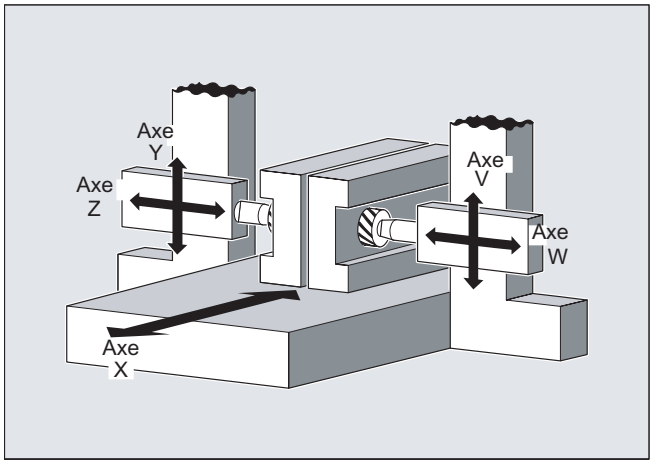
Remarque

Les déplacements conjugués s'effectuent toujours dans le système de coordonnées de base (SCB).

Le nombre de groupes d'axes à déplacements conjugués que vous pouvez activer simultanément dépend uniquement des combinaisons d'axes possibles sur la machine.

Exemple

La pièce doit être usinée simultanément sur deux faces avec la configuration d'axes représentée ci-contre. Pour ce faire, former deux groupes d'axes à déplacements conjugués.



Code de programme	Commentaire
…	
N100 TRAILON(V,Y)	; Activation du 1er groupe d'axes à déplacements conjugués
N110 TRAILON(W,Z,-1)	; Activation du 2e groupe d'axes à déplacements conjugués. Facteur de couplage négatif : l'axe conjugué se déplace en sens opposé par rapport à l'axe pilote.
N120 G0 Z10	; Déplacement des axes Z et W en sens opposé.
N130 G0 Y20	; Déplacement des axes Y et V dans le même sens.
…	
N200 G1 Y22 V25 F200	; Superposition d'un déplacement dépendant et d'un déplacement indépendant de l'axe conjugué V.
…	
TRAILOF(V,Y)	; Désactivation du 1er groupe d'axes à déplacements conjugués.
TRAILOF(W,Z)	; Désactivation du 2e groupe d'axes à déplacements conjugués.

Informations complémentaires

Types d'axes

Un groupe d'axes à déplacements conjugués peut combiner à volonté axes linéaires et axes rotatifs. Un axe simulé peut aussi être défini comme axe pilote.

Axes conjugués

A un axe conjugué, vous pouvez affecter deux axes pilotes à la fois. L'affectation s'effectue dans des groupes d'axes à déplacements conjugués différents.

Un axe conjugué peut être programmé avec toutes les instructions de déplacement disponibles (G0, G1, G2, G3, …). En plus des déplacements définis indépendamment, l'axe conjugué décrit les courses déduites de ses axes pilotes avec les facteurs de couplage.

Limitation dynamique

La limitation dynamique dépend du mode d'activation du groupe d'axes à déplacements conjugués :

- Activation dans le programme pièce
  Lorsque l'activation s'effectue dans le programme pièce et que tous les axes pilotes sont des axes du programme dans le canal d'activation, la dynamique de chaque axe conjugué est prise en compte lors du déplacement des axes pilotes, de sorte à ce qu'aucun axe conjugué ne soit surchargé.
  Lorsque l'activation s'effectue dans le programme pièce avec des axes pilotes qui ne sont pas des axes de programme dans le canal d'activation ($AA_TYP ≠ 1), la dynamique de l'axe conjugué n'est pas prise en compte lors du déplacement des axes pilotes. Dans le cas d'axes conjugués, dont la dynamique est inférieure à celle requise pour le couplage, il peut ainsi en résulter une surcharge.
- Activation dans une action synchrone
  Lorsque l'activation s'effectue dans une action synchrone, la dynamique des axes conjugués n'est pas prise en compte lors du déplacement des axes pilotes. Dans le cas d'axes conjugués, dont la dynamique est inférieure à celle requise pour le couplage, il peut ainsi en résulter une surcharge.

⚠ PRUDENCE

Surcharge de l'axe

Lorsqu'un groupe d'axes à déplacements conjugués est activé dans

- des actions synchrones,
- le programme pièce avec des axes pilotes qui ne sont pas des axes de programme dans le canal de l'axe conjugué,

il relève de la responsabilité expresse de l'utilisateur / du constructeur de machines de prévoir des mesures adaptées, afin que les déplacements de l'axe pilote n'entraînent pas de surcharge des axes conjugués.

Etat du couplage

L'interrogation de l'état du couplage d'un axe peut être réalisée dans le programme pièce au moyen de la variable système :

$AA_COUP_ACT[<axe>]

Valeur	Signification
0	aucun couplage activé
8	déplacements conjugués actifs

Affichage de la distance restant à parcourir par les axes conjugués pour des axes rotatifs modulo

Si les axes pilote et conjugués sont des axes rotatifs modulo, les déplacements de l'axe pilote de n * 360°, où n = 1, 2, 3... , s'additionnent dans l'affichage de la distance restant à parcourir des axes conjugués jusqu'à la désactivation du couplage.

Exemple : section de programme avec TRAILON, axe pilote B et axe asservi C

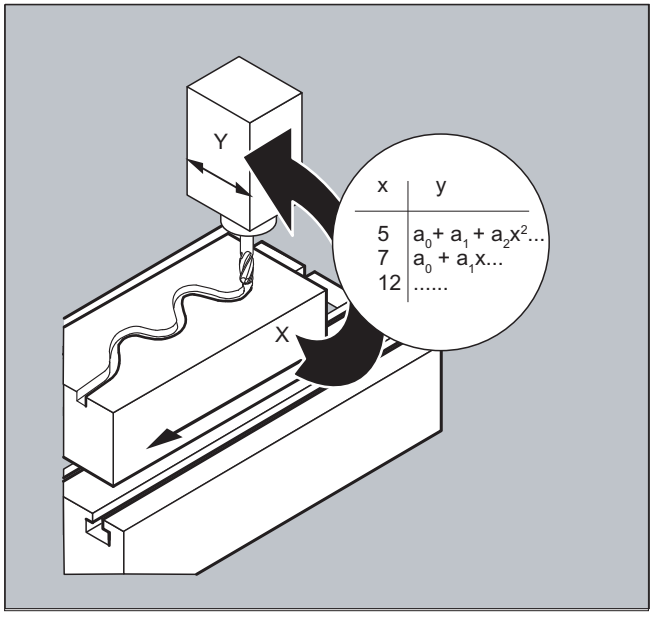
Code de programme	Commentaire
TRAILON(C,B,1)	; Activation du couplage
G0 B0	; Position de départ
	; Affichage de la distance restant à parcourir au début du bloc :
G91 B360	; B=360, C=360
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

13.2 Tables de courbes (CTAB)

Les tables de courbes permettent de programmer des rapports de position et de vitesse entre deux axes (axe pilote et axe asservi). La définition d'une table de courbes s'effectue dans le programme pièce.

Application

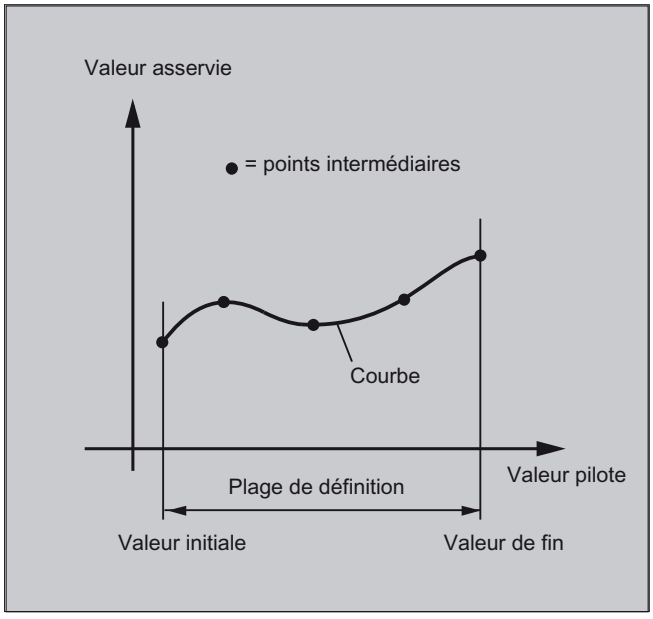
Les tables de courbes remplacent des cames mécaniques. la table de courbes est à la base du couplage de deux axes par valeur pilote, puisqu'elle crée le lien fonctionnel entre valeur pilote et valeur asservie : A partir des positions en corrélation réciproque de l'axe asservi et de l'axe pilote, la commande calcule conformément à la programmation, un polynôme équivalent à la came qu'il est appelé à remplacer.



13.2.1 Définition de tables de courbes (CTABDEF, CATBEND)

Une table de courbes représente un programme pièce ou une partie d'un programme pièce qui est identifié(e) par l'instruction CTABDEF en début et l'instruction CTABEND en fin de programme.

Dans cette section du programme pièce, on met en correspondance, par le biais d'instructions de déplacement, des positions isolées de l'axe pilote avec des positions univoques de l'axe asservi ; ces positions servent de points intermédiaires pour le calcul d'une courbe sous forme d'un polynôme du 5e degré au maximum.



Condition

La définition de tables de courbes requiert la réservation d'un espace mémoire au moyen d'une configuration correspondante des paramètres machine (→ constructeur de machine !).

Syntaxe

```
CTABDEF(<axe asservi>,<axe pilote>,<n>,<périodicité>[,<lieu
d'enregistrement>])
...
CTABEND
```

Signification

`CTABDEF( ) :`	Début de la définition de tables de courbes	
`CTABEND :`	Fin de la définition de tables de courbes	
`<Axe asservi> :`	Axe dont il s'agit de calculer le déplacement au moyen de la table de courbes	
`<Axe pilote> :`	Axe fournissant les valeurs pilotes pour le calcul du déplacement de l'axe asservi	
`<n> :`	Numéro (**ID**) de la table de courbes Le numéro d'une table de courbes est univoque et indépendant du lieu d'enregistrement. Des tables portant le même numéro ne peuvent pas se trouver à la fois dans la mémoire statique et dans la mémoire dynamique de la CN.	
`<périodicité> :`	Périodicité de la table	
	`0`	la table n'est pas périodique (elle n'est traitée qu'une seule fois, même dans le cas d'axes rotatifs)
	`1`	la table est périodique par rapport à l'axe pilote
	`2`	la table est périodique par rapport à l'axe pilote et l'axe asservi

<lieu de stockage> :	Indication du lieu d'enregistrement (facultative)	
	"SRAM"	La table de courbes est créée dans la mémoire **statique** de la CN.
	"DRAM"	La table de courbes est créée dans la mémoire **dynamique** de la CN.
	Remarque : si aucune valeur n'est programmée pour ce paramètre, le lieu d'enregistrement par défaut paramétré avec PM20905 $MC_CTAB_DEFAULT_MEMORY_TYPE est utilisé.	

Remarque

Ecraser

Une table de courbes est écrasée lorsque son numéro (<n>) est utilisé pour la nouvelle définition d'une table de courbes (exception : la table de courbes est active dans un couplage d'axes ou bloquée avec CTABLOCK). **L'écrasement n'est pas accompagné d'un avertissement correspondant !**

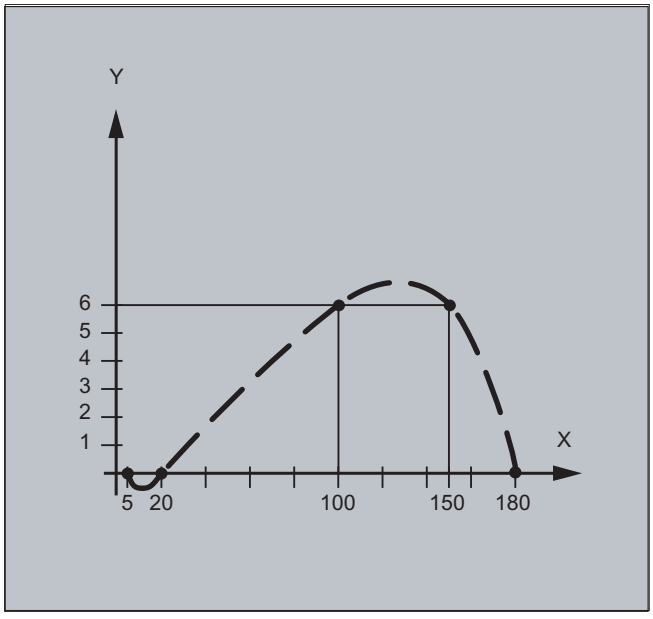
Exemples

Exemple 1 : section de programme comme définition d'une table de courbes

Une partie d'un programme doit servir, sans être modifiée, à la définition d'une table de courbes. L'instruction d'arrêt du prétraitement des blocs STOPRE qui y figure peut y être conservée et sera réactivée aussitôt que cette section de programme ne sera plus utilisée pour la définition d'une table de courbes, c'est-à-dire, dès que CTABDEF et CTABEND auront été supprimés.

```
Code de programme                Commentaire
…
CTABDEF(Y,X,1,1)                 ; Définition d'une table de courbes.
…
IF NOT ($P_CTABDEF)
STOPRE
ENDIF
…
CTABEND
```

Exemple 2 : définition d'une table de courbes non périodique



Code de programme	Commentaire
N100 CTABDEF(Y,X,3,0)	; Début de la définition d'une table de courbes non périodique ayant le numéro 3.
N110 X0 Y0	; 1re instruction de déplacement, définit les valeurs de départ et le 1er point intermédiaire : valeur pilote : 0, valeur asservie : 0
N120 X20 Y0	; 2e point intermédiaire : valeur pilote : 0…20, valeur asservie : valeur de départ...0
N130 X100 Y6	; 3e point intermédiaire : valeur pilote : 20…100, valeur asservie : 0…6
N140 X150 Y6	; 4e point intermédiaire : valeur pilote : 100…150, valeur asservie : 6…6
N150 X180 Y0	; 5e point intermédiaire : valeur pilote : 150…180, valeur asservie : 6…0
N200 CTABEND	; Fin de la définition. La table de courbes est générée dans sa représentation interne, en tant que polynôme du 5e degré au maximum. Le calcul de la courbe avec les points intermédiaires indiqués dépend du mode d'interpolation à effet modal sélectionné (interpolation circulaire, linéaire, de type spline). L'état du programme pièce avant le début de la définition est restauré.

Exemple 3 : Définition d'une table de courbes périodique

Définition d'une table de courbes périodique ayant le numéro 2, une plage de valeurs pilotes de 0 à 360, un déplacement de l'axe asservi de 0 à 45 avec retour à 0 :

```
Code de programme                                   Commentaire
N10 DEF REAL DEPPOS
N20 DEF REAL GRADIENT
N30 CTABDEF(Y,X,2,1)                                ; Début de la définition.
N40 G1 X=0 Y=0
N50 POLY
N60 PO[X]=(45.0)
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-90)
N80 PO[X]=(270.0)
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,90)
N100 PO[X]=(360.0)
N110 CTABEND                                        ; Fin de la définition.
; Test de la courbe par couplage de Y à X :
N120 G1 F1000 X0
N130 LEADON(Y,X,2)
N140 X360
N150 X0
N160 LEADOF(Y,X)

N170 DEPPOS=CTAB(75.0,2,GRADIENT)                   ; Lecture de la fonction de la ta-
                                                    ble à la valeur pilote 75.0.
N180 G0 X75 Y=DEPPOS                                ; Positionnement de l'axe pilote
                                                    et de l'axe asservi.
Quand le couplage est activé, il n'est pas nécessaire de synchroniser l'axe asservi.
N190 LEADON(Y,X,2)
N200 G1 X110 F1000
N210 LEADOF(Y,X)
N220 M30
```

plus d'informations

Valeur de départ et valeur de fin de la table de courbes

Le couple de positions axiales corrélées que l'on indique en premier (il s'agit de la première instruction de déplacement) dans la définition de la table de courbes est considéré comme valeur de départ de la plage de définition de la table de courbes. La valeur finale de la plage de définition de la table de courbes est constituée par la dernière instruction de déplacement.

Langage disponible

Pour définir la table de courbes, vous disposez du langage CN dans son intégralité.

Remarque

Les indications suivantes ne sont pas autorisées dans les définitions de tables de courbes :

- Arrêt du prétraitement des blocs
- Variations brusques dans le déplacement de l'axe pilote (en cas de changement de transformation par exemple),
- Instruction de déplacement seule pour l'axe asservi,
- Inversion du déplacement de l'axe pilote, autrement dit la position de l'axe pilote ne doit jamais être ambiguë,
- Programmation des instructions CTABDEF et CTABEND dans des niveaux de programme différents.

Effet d'instructions modales

Toutes les instructions à effet modal qui interviennent dans la définition de la table de courbes deviennent caduques dès que la définition est terminée. En d'autres termes, le programme pièce dans lequel a eu lieu la définition, présente le même état avant et après la définition de la table de courbes.

Affectations à des paramètres R

Les affectations à des paramètres R dans la définition d'une table de courbes sont réinitialisées après CTABEND.

Exemple :

Code de programme	Commentaire
...	
R10=5 R11=20	; R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	; R10=25
X=R10	
CTABEND	
...	; R10=5

Activation de ASPLINE, BSPLINE, CSPLINE

Si une spline ASPLINE, BSPLINE ou CSPLINE est activée à l'intérieur d'une définition de table de courbes CTABDEF ... CTABEND , au moins un point de départ doit être programmé avant cette activation de spline. Cela devrait éviter une activation immédiate après CTABDEF, compte tenu que la spline dépend de la position d'axe courante avant la définition de table de courbes.

Exemple :

Code de programme
...
CTABDEF(Y,X,1,0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND

Réutilisation de tables de courbes

Le lien fonctionnel créé entre l'axe pilote et l'axe asservi par la table de courbes est mémorisé sous le numéro de table qui a été choisi ; il est conservé au-delà de la fin du programme et d'un POWER OFF si la table a été enregistrée dans la mémoire statique de la CN (SRAM).

Une table créée dans la mémoire dynamique (DRAM) est supprimée lors d'un POWER ON et doit éventuellement être recréée.

Une fois créée, une table de courbes est utilisable pour des combinaisons quelconques entre axe pilote et axe asservi, indépendamment des axes qui ont servi à la créer.

Écrasement de tables de courbes

Dès que vous redéfinissez une table de courbe sous un numéro de table déjà utilisé, vous écrasez la table existante.

Exception : dans un couplage d'axes, une table de courbes est active ou bloquée avec CTABLOCK.

Remarque

L'écrasement d'une table de courbes n'est pas accompagné d'un avertissement particulier !

Définition de table de courbe active ?

La variable système `$P_CTABDEF` permet à tout moment d'interroger le système à partir du programme pièce, afin de savoir si une définition de table de courbes est active.

Annulation de la définition d'une table de courbes

Après la mise entre parenthèses des instructions de définition de la table de courbes, la section du programme pièce peut à nouveau être utilisée comme un programme pièce réel.

Chargement de tables de courbes avec "Exécution d'un programme externe"

Pour l'exécution externe de tables de courbes, la taille du tampon de rechargement (DRAM) doit être sélectionnée à l'aide de PM18360 $MN_MM_EXT_PROG_BUFFER_SIZE de sorte que celui-ci puisse contenir la définition complète des tables de courbes. Si ce n'est pas le cas, l'exécution du programme pièce est annulée et une alarme est émise.

Sauts de l'axe asservi

En l'absence de déplacement,
des sauts de l'axe asservi sont tolérables en fonction du paramètre machine :
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION

13.2.2 Vérification de l'existence d'une table de courbes (CTABEXISTS)

L'instruction CTABEXISTS permet de vérifier la présence d'un numéro donné de table de courbes dans la mémoire de la CN.

Syntaxe

```
CTABEXISTS(<n>)
```

Signification

CTABEXISTS :	Vérification de la présence de la table de courbes de numéro <n> dans la mémoire statique ou dynamique de la CN	
	0	La table n'existe pas.
	1	La table existe
<n> :	Numéro (**ID**) de la table de courbes	

13.2.3 Suppression d'une table de courbes (CTABDEL)

CTABDEL permet de supprimer des tables de courbes.

Remarque

Vous ne pouvez pas effacer des tables de courbes quand elles sont activées dans un couplage d'axes.

Syntaxe

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<lieu d'enregistrement>)
CTABDEL()
CTABDEL(,,<lieu d'enregistrement>)
```

Signification

`CTABDEL` :	Instruction de suppression de tables de courbes	
`<n>` :	Numéro (**ID**) de la table de courbes à supprimer Lors de la suppression d'une plage de tables de courbes `CTABDEL(<n>,<m>)`, `<n>` indique le numéro de la première table de courbes de la plage.	
`<m>` :	Lors de la suppression d'une plage de tables de courbes `CTABDEL(<n>,<m>)`, `<m>` indique le numéro de la dernière table de courbes de la plage. `<m>` doit être supérieur à `<n>` !	
`<lieu de stockage>` :	Indication du lieu d'enregistrement (facultative) Lors d'une suppression **sans** indication de lieu d'enregistrement, les tables de courbes indiquées sont supprimées dans les mémoires statique et dynamique de la CN. Lors d'une suppression **avec** indication de lieu d'enregistrement, seules les tables de courbes indiquées se trouvant dans la mémoire spécifiée sont supprimées. Les autres tables de courbes sont conservées.	
	`"SRAM"`	Suppression dans la mémoire **statique** de la CN
	`"DRAM"`	Suppression dans la mémoire **dynamique** de la CN

Si `CTABDEL` est programmé sans indication de la table de courbes à supprimer, **toutes** les tables de courbes ou toutes celles qui se trouvent dans la mémoire spécifiée sont supprimées :

`CTABDEL()` :	Supprime toutes les tables de courbes dans les mémoires statique et dynamique de la CN
`CTABDEL(,,"SRAM")` :	Supprime toutes les tables de courbes dans la mémoire statique de la CN
`CTABDEL(,,"DRAM")` :	Supprime toutes les tables de courbes dans la mémoire dynamique de la CN

Remarque

Si, en cas de suppression multiple `CTABDEL(<n>,<m>)` ou `CTABDEL()`, l'une au moins des tables de courbes à supprimer est active dans un couplage, l'instruction de suppression n'est pas exécutée, autrement dit **aucune** des tables de courbes adressées n'est supprimée.

13.2.4 Verrouillage de tables de courbes contre la suppression et l'écrasement (CTABLOCK, CTABUNLOCK)

Les tables de courbes peuvent être protégées contre la suppression et l'écrasement par inadvertance au moyen d'un verrouillage. Ce dernier peut à tout moment de nouveau être annulé.

Syntaxe

Verrouillage :
`CTABLOCK(<n>)`

```
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<lieu d'enregistrement>)
CTABLOCK()
CTABLOCK(,,<lieu d'enregistrement>)
```

Annulation du verrouillage :

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<lieu d'enregistrement>)
CTABUNLOCK()
CTABUNLOCK(,,<lieu d'enregistrement>)
```

Signification

`CTABLOCK` :	Instruction de **verrouillage** contre la suppression/l'écrasement	
`CTABUNLOCK` :	Instruction d'**annulation** du verrouillage contre la suppression/l'écrasement `CTABUNLOCK` libère les tables de courbes verrouillées avec `CTABLOCK`. Les tables de courbes qui agissent dans un couplage actif restent verrouillées et ne peuvent pas être supprimées. Le verrouillage avec `CTABLOCK` est annulé aussitôt que le couplage actif est désactivé. Cette table peut donc être effacée. Un nouvel appel de `CTABUNLOCK` n'est pas nécessaire.	
`<n>` :	Numéro (**ID**) de la table de courbes à verrouiller/déverrouiller Lors du verrouillage/déverrouillage d'une plage de tables de courbes `CTABLOCK(<n>,<m>)`/`CTABUNLOCK(<n>,<m>)`, `<n>` indique le numéro de la première table de courbes de la plage.	
`<m>` :	Lors du verrouillage/déverrouillage d'une plage de tables de courbes `CTABLOCK(<n>,<m>)`/`CTABUNLOCK(<n>,<m>)`, `<m>` indique le numéro de la dernière table de courbes de la plage. `<m>` doit être supérieur à `<n>` !	
`<lieu de stockage>` :	Indication du lieu d'enregistrement (facultative) Lors du verrouillage/déverrouillage **sans** indication de lieu d'enregistrement, les tables de courbes indiquées sont verrouillées/déverrouillées dans les mémoires statique et dynamique de la CN. Lors du verrouillage/déverrouillage **avec** indication de lieu d'enregistrement, seules les tables de courbes indiquées se trouvant dans la mémoire spécifiée sont verrouillées/déverrouillées. Les tables de courbes restantes ne sont pas verrouillées/déverrouillées.	
	`"SRAM"`	Verrouillage/déverrouillage dans la mémoire **statique** de la CN
	`"DRAM"`	Verrouillage/déverrouillage dans la mémoire **dynamique** de la CN

Si CTABLOCK/CTABUNLOCK est programmé sans indication de la table de courbes à verrouiller/déverrouiller, **toutes** les tables de courbes ou toutes celles qui se trouvent dans la mémoire spécifiée sont verrouillées/déverrouillées.

`CTABLOCK()` :	Verrouille toutes les tables de courbes dans les mémoires statique et dynamique de la CN
`CTABLOCK(,,"SRAM")` :	Verrouille toutes les tables de courbes dans la mémoire statique de la CN
`CTABLOCK(,,"DRAM")` :	Verrouille toutes les tables de courbes dans la mémoire dynamique de la CN

CTABUNLOCK() :	Déverrouille toutes les tables de courbes dans les mémoires statique et dynamique de la CN
CTABUNLOCK(,,"SRAM") :	Déverrouille toutes les tables de courbes dans la mémoire statique de la CN
CTABUNLOCK(,,"DRAM") :	Déverrouille toutes les tables de courbes dans la mémoire dynamique de la CN

13.2.5 Tables de courbes : détermination des propriétés de la table (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Ces instructions permettent d'interroger les principales propriétés d'une table de courbes (numéro de la table, état de verrouillage, lieu d'enregistrement, périodicité).

Syntaxe

```
CTABID(<p>)
CTABID(<p>,<lieu d'enregistrement>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

Signification

CTABID :	Indique le **numéro de la table** inscrit comme <p>ème table de courbes dans la mémoire spécifiée. Exemple : CTABID(1,"SRAM") indique le numéro de la première table de courbes dans la mémoire statique de la CN. La première table de courbes correspond à celle dont le numéro est le plus élevé. **Remarque :** si l'ordre des tables de courbes est modifié dans la mémoire entre des appels consécutifs de CTABID, p. ex. pour la suppression de tables de courbes avec CTABDEL, CTABID(<p>,...) peut indiquer une autre table de courbes que précédemment avec le même numéro <p>.	
CTABISLOCK :	Indique l'**état de verrouillage** de la table de courbes portant le numéro <n> :	
	0	La table n'est pas verrouillée
	1	La table est verrouillée par CTABLOCK
	2	La table est verrouillée par le couplage actif
	3	La table est verrouillée par CTABLOCK et par le couplage actif
	-1	La table n'existe pas.
CTABMEMTYP :	Indique le **lieu d'enregistrement** de la table de courbes portant le numéro <n> :	
	0	table dans la mémoire statique de la CN
	1	table dans la mémoire dynamique de la CN
	-1	La table n'existe pas.

CTABPERIOD :	Indique la **périodicité** de la table de courbes portant le numéro <n> :	
	0	la table n'est pas périodique
	1	la table est périodique dans l'axe pilote
	2	la table est périodique dans l'axe pilote et dans l'axe asservi
	-1	La table n'existe pas.
<p> :	Numéro d'entrée en mémoire	
<n> :	Numéro (**ID**) de la table de courbes	
<lieu de stockage> :	Indication du lieu d'enregistrement (facultative)	
	"SRAM"	Mémoire **statique** de la CN
	"DRAM"	Mémoire **dynamique** de la CN
	Remarque : si aucune valeur n'est programmée pour ce paramètre, le lieu d'enregistrement par défaut paramétré avec PM20905 $MC_CTAB_DEFAULT_MEMORY_TYPE est utilisé.	

13.2.6 Lecture des valeurs de la table de courbes (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Les valeurs suivantes de la table de courbes peuvent être lues dans le programme pièce :

- Valeurs de l'axe asservi et de l'axe pilote au début et à la fin d'une table de courbes
- Valeurs de l'axe asservi au début et à la fin d'un segment de courbe
- Valeur de l'axe asservi correspondant à une valeur de l'axe pilote
- Valeur de l'axe pilote correspondant à une valeur de l'axe asservi
- Valeur minimale et valeur maximale de l'axe asservi
  - dans la plage de définition entière de la table de courbes
    ou
  - dans un intervalle défini de la table de courbes

Syntaxe

```
CTABTSV(<n>,<gradient>[,<axe asservi>])
CTABTEV(<n>,<gradient>[,<axe asservi>])
CTABTSP(<n>,<gradient>[,<axe pilote>])
CTABTEP(<n>,<gradient>[,<axe pilote>])
CTABSSV(<valeur pilote>,<n>,<gradient>[,<axe asservi>])
CTABSEV(<valeur pilote>,<n>,<gradient>[,<axe asservi>])
CTAB(<valeur pilote>,<n>,<gradient>[,<axe asservi>,<axe pilote>]
CTABINV(<valeur asservie>,<valeur approchée>,<n>,<gradient>[,<axe
asservi>,<axe pilote>]
CTABTMIN(<n>[,<axe asservi>])
CTABTMAX(<n>[,<axe asservi>])
CTABTMIN(<n>,<a>,<b>[,<axe asservi>,<axe pilote>])
CTABTMAX(<n>,<a>,<b>[,<axe asservi>,<axe pilote>])
```

Signification

CTABTSV :	Lecture de la valeur de l'axe asservi au **début** de la table de courbes portant le numéro <n>
CTABTEV :	Lecture de la valeur de l'axe asservi à la **fin** de la table de courbes portant le numéro <n>
CTABTSP :	Lecture de la valeur de l'axe pilote au **début** de la table de courbes portant le numéro <n>
CTABTEP :	Lecture de la valeur de l'axe pilote à la **fin** de la table de courbes portant le numéro <n>
CTABSSV :	Lecture de la valeur de l'axe asservi au **début** du segment de courbe appartenant à la valeur de l'axe pilote spécifié (<valeur pilote>)
CTABSEV :	Lecture de la valeur de l'axe asservi à la **fin** du segment de courbe appartenant à la valeur de l'axe pilote spécifié (<valeur pilote>)
CTAB :	Lecture de la valeur de l'axe asservi correspondant à la valeur spécifiée de l'axe pilote (<valeur pilote>)
CTABINV :	Lecture de la valeur de l'axe pilote correspondant à la valeur spécifiée de l'axe asservi (<valeur asservie>)
CTABTMIN :	Définition de la **valeur minimale** de l'axe asservi : • dans la plage de définition entière de la table de courbes ou • dans un intervalle défini <a> ... <b>
CTABTMAX :	Définition de la **valeur maximale** de l'axe asservi : • dans la plage de définition entière de la table de courbes ou • dans un intervalle défini <a> ... <b>
<n> :	Numéro (**ID**) de la table de courbes
<Gradient> :	Le paramètre <gradient> indique la **pente** de la fonction représentant la table de courbes à la position déterminée
<Axe asservi> :	Axe dont il s'agit de calculer le déplacement au moyen de la table de courbes (facultatif)
<Axe pilote> :	Axe fournissant les valeurs pilotes pour le calcul du déplacement de l'axe asservi (facultatif)
<Valeur asservie> :	Valeur de l'axe asservi pour la lecture de la valeur correspondante de l'axe pilote pour CTABINV
<Valeur pilote> :	Valeur de l'axe pilote : • pour la lecture de la valeur correspondante de l'axe asservi pour CTAB ou • pour la sélection du segment de courbe pour CTABSSV/CTABSEV
<Valeur approchée> :	L'affectation d'une valeur de l'axe pilote à une valeur de l'axe asservi pour CTABINV ne doit pas toujours être univoque. C'est la raison pour laquelle CTABINV requiert, comme paramètre, une valeur approchée de la valeur recherchée de l'axe pilote.
<a> :	Limite **inférieure** de l'intervalle des valeurs pilotes pour CTABTMIN/ CTABTMAX

<b> :	Limite **supérieure** de l'intervalle des valeurs pilotes pour CTABTMIN/ CTABTMAX
	Remarque : L'intervalle des valeurs pilotes <a> ... <b> doit se situer dans la plage de définition de la table de courbes.

Exemples

Exemple 1 :

Déterminer les valeurs de l'axe asservi et de l'axe pilote au début et à la fin de la table de courbes ainsi que les valeurs minimale et maximale de l'axe asservi dans la plage de définition entière de la table de courbes.

```
Code de programme                          Commentaire
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL STARTPARA
N40 DEF REAL ENDPARA
N50 DEF REAL MINVAL
N60 DEF REAL MAXVAL
N70 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                      ; Début de la définition de la table
N110 X0 Y10                                ; Position de départ du 1er segment de table
N120 X30 Y40                               ; Position finale du 1er segment de table =
                                           position de départ du 2e segment de table
N130 X60 Y5                                ; Position finale du 2e segment de table = ...
N140 X70 Y30
N150 X80 Y20
N160 CTABEND                               ; Fin de la définition de la table.
...
N200 STARTPOS=CTABTSV(1,GRADIENT)          ; Valeur de l'axe asservi au début de la table
                                           de courbes = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)            ; Valeur de l'axe asservi à la fin de la table
                                           de courbes = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)         ; Valeur de l'axe pilote au début de la table
                                           de courbes = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)           ; Valeur de l'axe pilote à la fin de la table
                                           de courbes = 80
N240 MINVAL=CTABTMIN(1)                    ; Valeur minimale de l'axe asservi pour Y=5
N250 MAXVAL=CTABTMAX(1)                    ; Valeur maximale de l'axe asservi pour Y=40
```

Exemple 2 :

Déterminer les valeurs de l'axe asservi au début et à la fin du segment de courbe correspondant à la valeur de l'axe pilote X=30.

Code de programme	Commentaire
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	; Début de la définition de la table.
N110 X0 Y0	; Position de départ du 1er segment de table
N120 X20 Y10	; Position finale du 1er segment de table = position de départ du 2e segment de table
N130 X40 Y40	; Position finale du 2e segment de table = ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	; Fin de la définition de la table.
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; Position de départ Y dans le 2e segment = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	; Position finale Y dans le 2e segment = 40

Informations complémentaires

Utilisation dans des actions synchrones

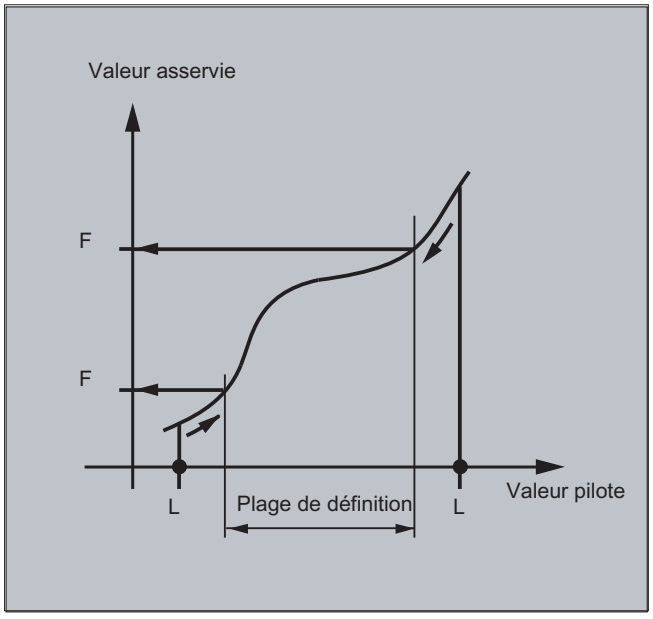
Il est également possible de lire dans des actions synchrones l'ensemble des instructions de lecture des valeurs d'une table de courbes (voir aussi le chapitre "Actions synchrones au déplacement").

Pour l'utilisation des instructions CTABINV, CTABTMIN et CTABTMAX, il faut s'assurer que :

- la puissance de la CN soit suffisante au moment de l'exécution
  ou
- le nombre de segments de la table de courbes soit interrogé avant l'appel afin de pouvoir subdiviser, le cas échéant, la table concernée

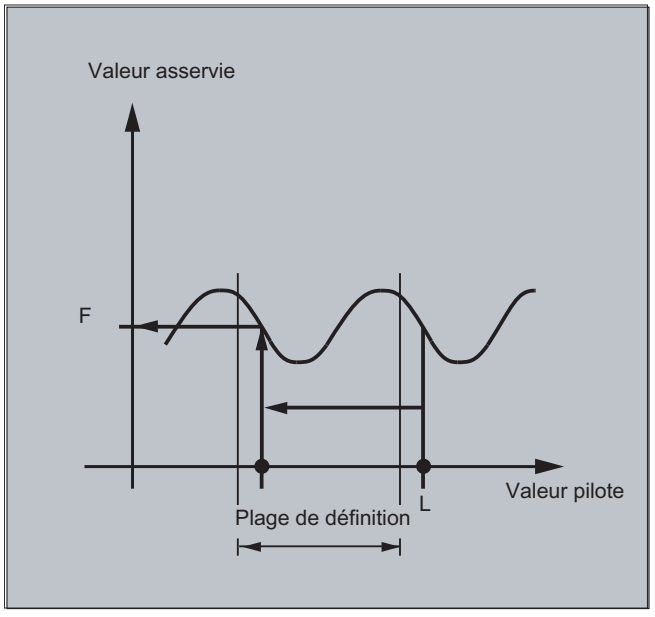
CTAB dans le cas de tables de courbes non périodiques

Lorsque la <valeur pilote> se situe en-dehors de la plage de définition, la valeur asservie fournie est la limite supérieure ou inférieure :



CTAB dans le cas de tables de courbes périodiques

Lorsque la `<valeur pilote>` indiquée se situe en-dehors de la plage de définition, la valeur pilote modulo de la plage de définition est évaluée et la valeur asservie correspondante est fournie :



Valeur approchée pour CTABINV

Pour fournir la valeur pilote recherchée, l'instruction CTABINV requiert une valeur approchée. CTABINV fournit la valeur pilote la plus proche de la valeur approchée. Il peut p. ex. s'agir de la valeur pilote du dernier cycle d'interpolation.

Pente de la fonction représentant la table de courbes

L'indication de la pente (<gradient>) permet de calculer la vitesse de l'axe pilote ou de l'axe asservi à la position correspondante.

Indication de l'axe pilote ou de l'axe asservi

L'indication facultative de l'axe pilote ou de l'axe asservi est nécessaire lorsque l'axe pilote et l'axe asservi sont configurés avec des unités de longueur différentes.

CTABSSV, CTABSEV

Dans les cas suivants, les instructions CTABSSV et CTABSEV ne conviennent **pas** pour l'interrogation des segments programmés :

- des cercles ou des développantes sont programmés,
- des chanfreins ou des arrondis sont activés avec CHF/RND,
- l'arrondissement avec G643 est activé,
- la compression de blocs de la CN avec COMPON/COMPCURV/COMPCAD est activée.

13.2.7 Tables de courbes : Vérification de l'utilisation des ressources (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Ces instructions permettent au programmeur de s'informer sur l'occupation actuelle des ressources requises par les tables de courbes, segments de tables et polynômes.

Syntaxe

```
CTABNO
CTABNOMEM(<lieu d'enregistrement>)
CTABFNO(<lieu d'enregistrement>)
CTABSEGID(<n>,<lieu d'enregistrement>)
CTABSEG(<lieu d'enregistrement>,<type de segment>)
CTABFSEG(<lieu d'enregistrement>,<type de segment>)
CTABMSEG(<type de segment>,<type de segment>)
CTABPOLID(<n>)
CTABPOL(<lieu d'enregistrement>)
CTABFPOL(<lieu d'enregistrement>)
CTABMPOL(<lieu d'enregistrement>)
```

Signification

CTABNO :	Détermination du nombre total de tables de courbes **définies** (dans les mémoires statique et dynamique de la CN)
CTABNOMEM :	Détermination du nombre de tables de courbes **définies** dans le <lieu d'enregistrement> indiqué
CTABFNO :	Détermination du nombre de tables de courbes qu'il est **encore possible** d'enregistrer dans le <lieu d'enregistrement> indiqué

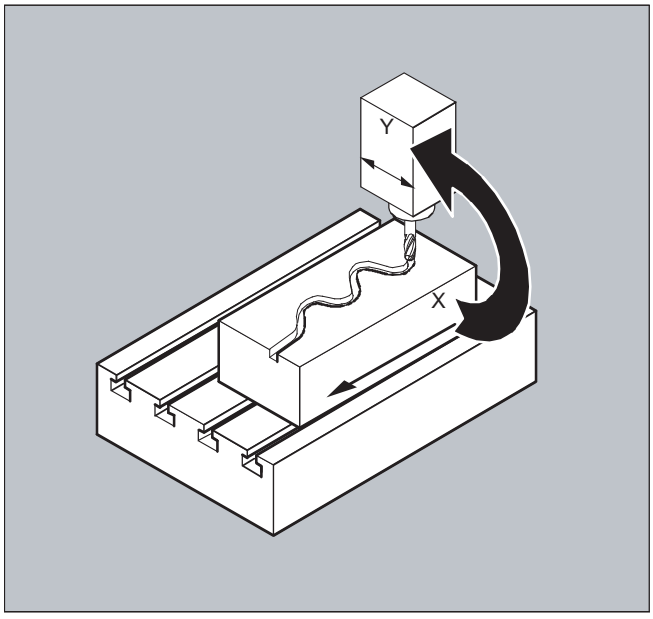
CTABSEGID :	Détermination du nombre de segments de courbe avec le <type de segment> spécifié qu'il est possible d'utiliser dans la table de courbes de numéro <n>	
CTABSEG :	Détermination du nombre de segments de courbe **utilisés** avec le <type de segment> spécifié dans le <lieu d'enregistrement> indiqué	
CTABFSEG :	Détermination du nombre de segments de courbe avec le <type de segment> spécifié qu'il est **encore possible** d'enregistrer dans le <lieu d'enregistrement> indiqué	
CTABMSEG :	Détermination du nombre **maximum possible** de segments de courbe avec le <type de segment> spécifié qu'il est possible d'enregistrer dans le <lieu d'enregistrement> indiqué	
CTABPOLID :	Détermination du nombre de polynômes de courbe utilisés par la table de courbes de numéro <n>	
CTABPOL :	Détermination du nombre de polynômes de courbes **utilisés** dans le <lieu d'enregistrement> indiqué	
CTABFPOL :	Détermination du nombre de polynômes de courbes qu'il est **encore possible** d'enregistrer dans le <lieu d'enregistrement> indiqué	
CTABMPOL :	Détermination du nombre **maximum possible** de polynômes de courbes qu'il est possible d'enregistrer dans le <lieu d'enregistrement> indiqué	
<n> :	Numéro (**ID**) de la table de courbes	
<lieu de stockage> :	Indication du lieu d'enregistrement (facultative)	
	"SRAM"	Mémoire **statique** de la CN
	"DRAM"	Mémoire **dynamique** de la CN
	Remarque : si aucune valeur n'est programmée pour ce paramètre, le lieu d'enregistrement par défaut paramétré avec PM20905 $MC_CTAB_DEFAULT_MEMORY_TYPE est utilisé.	
<type de segment> :	Indication du type de segment (facultative)	
	"L"	Segments linéaires
	"P"	Segments polynômiaux
	Remarque : si aucune valeur n'est programmée pour ce paramètre, la somme des segments linéaires et polynômiaux est fournie.	

13.3 Couplage de deux axes par valeur pilote (LEADON, LEADOF)

Remarque

Cette fonction n'est pas disponible pour SINUMERIK 828D.

Quand il y a couplage de deux axes par valeur pilote, l'axe pilote et l'axe asservi se déplacent de façon synchrone. Une position donnée de l'axe asservi correspond sans équivoque à une position de l'axe pilote (simulée le cas échéant) sur la base d'une table de courbe ou d'un polynôme en découlant.



Axe pilote est le nom donné à l'axe qui fournit les valeurs d'entrée pour la table de courbe.
Axe asservi est le nom donné à l'axe qui prend les positions calculées sur la base de la table de courbe.

Couplage par valeur réelle et couplage par valeur de consigne

Comme valeurs pilotes, autrement dit comme valeurs de départ pour déterminer les positions de l'axe asservi, on peut utiliser :

- les valeurs réelles de l'axe pilote : Couplage par valeur réelle
- Valeurs de consigne de la position de l'axe pilote : Couplage par valeur de consigne

Le couplage par valeur pilote se réfère toujours au système de coordonnées de base.

Pour créer des tables de courbe, voir chapitre "Tables de courbe".

Syntaxe

```
LEADON(<axe asservi>,<axe pilote>,<n>)
LEADOR(<axe asservi>,<axe pilote>)
```

ou désactivation sans indication de l'axe pilote :

```
LEADOF(<axe asservi>)
```

Le couplage par valeur pilote peut être activé et désactivé aussi bien depuis le programme pièce que pendant les déplacements depuis une action synchrone.

Signification

`LEADON` :	Activer le couplage par valeur pilote
`LEADOF` :	Désactiver le couplage par valeur pilote
`<Axe asservi>` :	Axe asservi
`<Axe pilote>` :	Axe pilote
`<n>` :	Numéro de la table de courbe
`$SA_LEAD_TYPE` :	Permutation entre le couplage par valeur réelle et le couplage par valeur de consigne

Désactiver le couplage par valeur pilote, LEADOF

Après la désactivation du couplage par valeur pilote, l'axe asservi redevient axe de commande normal.

Couplage de deux axes par valeur pilote et différents états de fonctionnement, RESET

Selon le réglage qui a été fait dans les paramètres machine, les couplages par valeur pilote sont désactivés avec RESET.

Exemple de couplage par valeur pilote à partir d'une action synchrone

Sur une presse, un couplage mécanique conventionnel entre un axe pilote (axe d'emboutissage) et les axes d'un système de transfert est à remplacer par un couplage électronique.

Il démontre comment on peut remplacer sur une presse un système de transfert mécanique par un système de transfert électronique. Les phases de couplage et de découplage sont réalisées sous forme **d'actions synchrones statiques**.

Les axes de transfert et les axes auxiliaires sont pilotés par l'axe pilote VP (axe d'emboutissage) en tant qu'axes asservis, par le biais des tables de courbe.

Axes asservis

X - Axe d'avance ou axe longitudinal
YL - Axe de fermeture ou axe transversal
ZL - Axe de course
U - Avance par rouleaux, axe auxiliaire
V - Tête de dressage, axe auxiliaire
W - Graissage, axe auxiliaire

Actions

Parmi les actions synchrones, on trouve :

- couplage, `LEADON(<axe asservi>,<axe pilote>,<numéro de la table de courbe>)`
- Découplage, `LEADOF(<axe asservi>,<axe pilote>)`
- Préréglage des mémoires de valeurs réelles, `PRESETON(<axe>,<valeur>)`

- Définition du mémento, `$AC_MARKER[i]=<valeur>`
- Type de couplage : valeur pilote réelle/virtuelle
- Accostage de positions d'axes, `POS[<axe>]=<valeur>`

Conditions

Des entrées TOR rapides, des variables temps réel `$AC_MARKER` et des comparaisons de positions, combinées avec l'opérateur logique AND, sont exploitées comme conditions.

Remarque

Dans l'exemple qui suit, nous avons utilisé des retours de lignes, des retraits et des caractères **gras** pour une meilleure lisibilité du programme. Pour la commande, tout ce qui figure sous un numéro de ligne représente effectivement une ligne de programme.

Commentaire

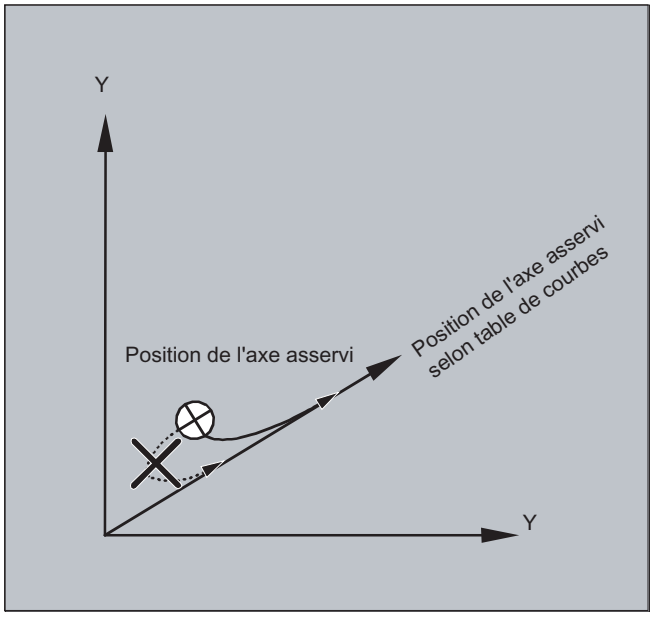
Code de programme	**Commentaire**
	; Définit toutes les actions synchrones statiques.
	; **** Remise à zéro des mémentos
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
	; **** E1 0=>1 Couplage transfert ACTIF
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_MARKER[0]==0) DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) $AC_MARKER[0]=1	
	; **** E1 0=>1 Couplage dispositif d'avance à rouleaux ACTIF
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0) DO LEADON(U,LW,4) PRESETON(U,0) $AC_MARKER[5]=1	
	; **** E1 0->1 Couplage tête de dressage ACTIF
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0) DO LEADON(V,LW,4) PRESETON(V,0) $AC_MARKER[6]=1	
	; **** E1 0->1 Couplage graissage ACTIF
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[7]==0) DO LEADON(W,LW,4) PRESETON(W,0) $AC_MARKER[7]=1	
	; **** E2 0=>1 Couplage NON ACTIF
N30 IDS=3 EVERY ($A_IN[2]==1) DO LEADOF(X,LW) LEADOF(YL,LW) LEADOF(ZL,LW) LEADOF(U,LW) LEADOF(V,LW) LEADOF(W,LW) $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
....	
N110 G04 F01	
N120 M30	

Description

Le couplage par valeur pilote exige la synchronisation de l'axe pilote et de l'axe asservi. Cette synchronisation est obtenue uniquement si l'axe asservi se trouve, au moment où le couplage par valeur pilote est activé, dans la plage de tolérance de la courbe calculée avec la table.

La plage de tolérance pour la position de l'axe asservi est définie par le paramètre machine
PM 37200 : COUPLE_POS_POL_COARSE A_LEAD_TYPE .

Si, lors de l'activation du couplage par valeur pilote, l'axe asservi n'est pas encore positionné correctement, le synchronisme s'enclenchera automatiquement dès que la position de consigne calculée pour l'axe asservi approchera de sa position effective. Pendant la synchronisation, l'axe asservi est déplacé dans la direction définie par sa vitesse de consigne (calculée à partir de la vitesse de l'axe pilote et CTAB).

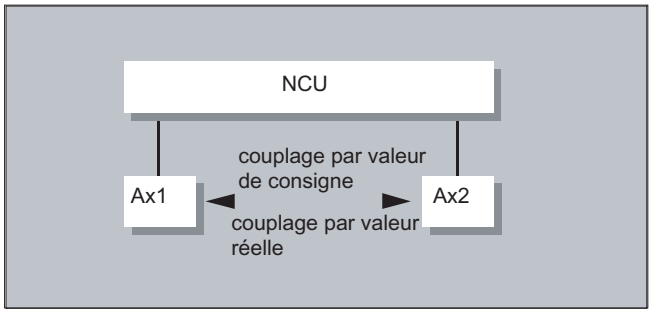


Pas de synchronisation

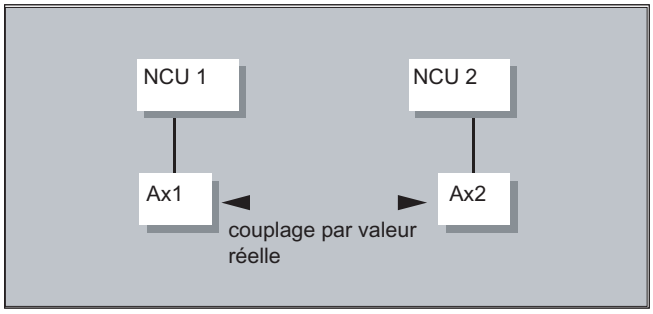
Si la position calculée de l'axe asservi s'éloigne de sa position effective lors de l'activation du couplage par valeur pilote, il n'y a pas de synchronisation.

Couplage par valeur réelle et couplage par valeur de consigne

Comparé au couplage par la valeur réelle, le couplage par la valeur de consigne fournit un meilleur synchronisme entre l'axe pilote et l'axe asservi ; c'est la raison pour laquelle il est préréglé en version standard.



Le couplage par la valeur de consigne est possible uniquement si l'axe pilote et l'axe asservi sont interpolés par la même NCU. Quand l'axe pilote est un axe externe, le couplage de l'axe asservi avec cet axe pilote ne peut se faire que par des valeurs réelles.



Le **basculement** est possible par le biais de la donnée de réglage $SA_LEAD_TYPE.

Le basculement entre couplage par valeur réelle et couplage par valeur de consigne doit toujours se faire quand l'axe asservi est en état d'immobilisation. En effet, une resynchronisation après le basculement n'a lieu que dans cet état.

Exemple d'application

Quand la machine est soumise à de fortes vibrations, la lecture des valeurs réelles ne peut pas se faire correctement. Pour mettre en œuvre le couplage par valeur pilote dans un transfert sur presse, il peut être nécessaire de basculer du couplage par valeur réelle au couplage par valeur de consigne, dans les phases d'usinage où les vibrations sont les plus fortes.

Simulation de valeurs pilotes dans un couplage par valeur de consigne

Par le biais d'un paramètre machine, on peut séparer l'interpolateur de l'axe pilote et le servo. On peut alors, dans un couplage par valeur de consigne, créer des valeurs de consigne sans qu'il y ait un déplacement réel de l'axe pilote.

Les valeurs pilotes créées par un couplage par la valeur de consigne peuvent être lues dans les variables suivantes, afin d'être utilisées dans des actions synchrones par exemple :

- $AA_LEAD_P	Valeur pilote de position
- $AA_LEAD_V	Valeur pilote de vitesse

Création de valeurs pilotes

Vous pouvez aussi créer des valeurs pilotes avec d'autres procédés que vous avez vous-même programmés. Les valeurs pilotes ainsi générées sont écrites dans les variables

- $AA_LEAD_SP	Valeur pilote de position
- $AA_LEAD_SV	Valeur pilote de vitesse

et lues à partir de celles-ci. Pour utiliser ces variables, il convient de définir la donnée de réglage comme suit : $SA_LEAD_TYPE = 2.

Etat du couplage

Vous pouvez interroger l'état du couplage dans le programme pièce CN avec la variable système suivante :

```
$AA_COUP_ACT[axe]
```
0 : aucun couplage activé
16 : couplage par valeur pilote activé

Gestion des états pour les actions synchrones

Les phases de commutation et de couplage sont gérées par des variables temps réel :

```
$AC_MARKER[i] = n
```

avec :
numéro de mémento i
valeur d'état n

13.4 Réducteur électronique (EG)

La fonction "Réducteur Electronique" permet, après un bloc de déplacement linéaire, de faire piloter le déplacement d'un **axe asservi** en fonction de 5 **axes pilotes** au maximum. Les relations entre les axes pilotes et l'axe asservi sont définis, pour chaque axe pilote, par un facteur de couplage.

Le trajet de l'axe asservi est calculé en additionnant les différents trajets des axes pilotes multipliés par les facteur de couplage respectifs. Lors de l'activation d'un groupe d'axes EG, une synchronisation de l'axe asservi à une position définie peut avoir lieu. Un groupe de réducteurs électroniques peut être :

- défini,
- activé,
- désactivé,
- effacé

.

Le déplacement de l'axe asservi peut être déduit des :

- valeurs de consignes des axes pilotes,
- valeurs réelles des axes pilotes.

Dans le cadre de l'extension, des couplages non linéaires entre les axes pilotes et les axes asservis peuvent être réalisés par des **tables de courbes** (voir chapitre Mode de déplacement). Les réducteurs électroniques peuvent être montés en cascade, c'est-à-dire que l'axe asservi d'un réducteur électronique peut être l'axe pilote d'un autre réducteur électronique.

13.4.1 Définition d'un réducteur électronique (EGDEF)

Un groupe d'axes EG est déterminé par l'indication de l'axe asservi et d'au moins un (au plus cinq) axes pilotes avec le type de couplage respectif.

Condition

Condition pour la définition d'un groupe d'axes EG :

un couplage d'axe ne doit pas être défini pour l'axe asservi (le cas échéant, un couplage existant doit être effacé avec EGDEL).

Syntaxe

```
EGDEF(axe asservi,axe pilote1,type de couplage1,axe pilote2,type de
couplage2,...)
```

Signification

`EGDEF` :	Définition d'un réducteur électronique
`Axe asservi` :	Axe influencé par les axes pilotes

Axe pilote1 ,..., axe pilote5	Axes qui influencent l'axe asservi	
Type de couplage1 ,..., type de couplage5	Type de couplage Les types de couplage pouvant être différents, ceux-ci doivent être indiqués pour tous les axes pilotes.	
	Valeur :	**Signification :**
	0	L'axe asservi est influencé par la **valeur réelle** de l'axe pilote correspondant.
	1	L'axe asservi est influencé par la **consigne** de l'axe pilote correspondant.

Remarque

Les facteurs de couplage sont préréglés avec la valeur zéro lors de la définition du groupe d'axes EG.

Remarque

EGDEF déclenche un arrêt du prétraitement des blocs. La définition de transmission avec EGDEF est alors à utiliser de manière inchangée si, sur les systèmes, un ou plusieurs axes pilotes influencent l'axe asservi par des **tables de courbes**.

Exemple

Code de programme	Commentaire
EGDEF(C,B,1,Z,1,Y,1)	; Définition d'un groupe d'axes EG. Les axes pilotes B, Z, Y influencent l'axe asservi C par la valeur de consigne.

13.4.2 Activation du réducteur électronique (EGON, EGONSYN, EGONSYNE)

Il existe 3 variantes pour activer un groupe d'axes EG.

Syntaxe

Variante 1 :

Le groupe d'axes EG est activé de façon sélective sans synchronisation par :

```
EGON(FA,"mode de changement de
bloc",AP1,Z1,N1,AP2,Z2,N2,...,AP5,Z5,N5)
```

Variante 2 :

Le groupe d'axes EG est activé de façon sélective avec synchronisation par :

```
EGONSYN(FA,"mode de changement de bloc",SynPosAA,
[,APi,SynPosAPi,Zi,Ni])
```

Variante 3 :

Le groupe d'axes EG est activé de façon sélective avec synchronisation et le mode d'accostage est défini par :
`EGONSYNE(FA,"mode de changement de bloc",SynPosAA,mode d'accostage[,APi,SynPosAPi,Zi,Ni])`

Signification

Variante 1 :

FA	Axe asservi	
`mode de changement de bloc` :	Les modes suivants peuvent être utilisés :	
	`"NOC"`	Le changement de bloc s'effectue immédiatement.
	`"FINE"`	Le changement de bloc s'effectue au "synchronisme fin".
	`"COARSE"`	Le changement de bloc s'effectue au "synchronisme grossier".
	`"IPOSTOP"`	Le changement de bloc s'effectue à la valeur de consigne du synchronisme.
AP1, ... AP5	Axes pilotes	
Z1, ... Z5	Numérateur pour le facteur de couplage i	
N1, ... N5	Dénominateur pour le facteur de couplage i Facteur de couplage i = numérateur i / dénominateur i	

Seuls les axes pilotes spécifiés auparavant dans l'instruction EGDEF peuvent être programmés. Un axe pilote au moins doit être programmé.

Variante 2 :

`FA`	Axe asservi	
`mode de changement de bloc` :	Les modes suivants peuvent être utilisés :	
	`"NOC"`	Le changement de bloc s'effectue immédiatement.
	`"FINE"`	Le changement de bloc s'effectue au "synchronisme fin".
	`"COARSE"`	Le changement de bloc s'effectue au "synchronisme grossier".
	`"IPOSTOP"`	Le changement de bloc s'effectue à la valeur de consigne du synchronisme.
`[,APi,SynPosAPi,Zi,Ni]`	(ne pas écrire les crochets droits) Au minimum 1, au maximum 5 suites de :	
`AP1, ... AP5`	Axes pilotes	
`PosSynAPi`	Position de synchronisation pour l'axe pilote i	
`Z1, ... Z5`	Numérateur pour le facteur de couplage i	
`N1, ... N5`	Dénominateur pour le facteur de couplage i Facteur de couplage i = numérateur i / dénominateur i	

Seuls les axes pilotes spécifiés auparavant dans l'instruction EGDEF peuvent être programmés. Les "positions de synchronisme" programmés pour l'axe asservi (PosSynAA) et pour les axes pilotes (PosSynAP) sont des positions dans lesquelles le couplage d'axes est considéré *synchronisé*. Si le réducteur électronique n'est pas synchronisé à l'activation, l'axe asservi se déplace à la position de synchronisation définie.

Variante 3 :

Les paramètres sont identiques à ceux de la variante 2 avec en plus :

`mode d'accostage` :	Les modes suivants peuvent être utilisés :	
	`"NTGT"`	accoster l'entredent suivant avec une optimisation du temps
	`"NTGP"`	accoster l'entredent suivant avec une optimisation du parcours
	`"ACN"`	déplacer l'axe rotatif en sens de rotation négatif, en absolu
	`"ACP"`	déplacer l'axe rotatif en sens de rotation positif, en absolu
	`"DCT"`	Position de synchronisme programmée avec une optimisation du temps
	`"DCP"`	Position de synchronisme programmée avec une optimisation du parcours

La variante 3 influe uniquement sur les axes asservis à modulos couplés aux axes pilotes à modulos. L'optimisation du temps tient compte des limites de vitesse de l'axe asservi.

plus d'informations

Description des variantes d'activation

Variante 1 :

Les positions des axes pilotes et des axes asservis sont enregistrées, lors de l'activation, en tant que "positions de synchronisme". Les "positions de synchronisme" peuvent être lues à l'aide de la variable `$AA_EG_SYN`.

Variante 2 :

Si le groupe d'axes comprend des axes à valeur modulo, leurs positions sont ramenés à la plage définie par la valeur modulo. Ceci garantit que la position de synchronisation la plus proche est accostée (*synchronisation relative* : par exemple l'entredent suivant). Si le signal d'interface DB (30 +numéro d'axe), DBX 26 bit 4 "Déblocage forçage déplacement axe asservi" n'est pas à 1 pour l'axe asservi, la position de synchronisation n'est pas accostée. Dans ce cas, le programme est arrêté au bloc EGONSYN et l'alarme à effacement automatique 16771 est signalée jusqu'à ce que le signal mentionné ci-dessus soit mis à 1.

Variante 3 :

L'entredent (degrés) est calculé de la manière suivante : 360 * Zi/Ni. Si l'axe asservi est à l'arrêt au moment de l'activation, l'optimisation du parcours génère le même comportement que l'optimisation du temps.

Si l'axe asservi est déjà en déplacement, NTGP est synchronisé avec l'entredent suivant indépendamment de sa vitesse actuelle. Si l'axe asservi est déjà en déplacement, NTGP est synchronisé avec l'entredent suivant en fonction de sa vitesse actuelle. Le cas échéant, la vitesse est réduite.

Tables de courbes

Si une **table de courbes** est utilisée pour un axe pilote :

Ni Le dénominateur du facteur de couplage linéaire (Ni) doit être mis à 0. (Le dénominateur 0 serait illicite pour des couplages linéaires). Pour la commande, le dénominateur 0 est l'identificateur Zi.

Zi Zi doit être interprété comme le numéro de table de courbes à utiliser. La table de courbes avec le numéro indiqué doit être déjà définie lors de l'activation.

APi L'indication de l'axe pilote (APi) correspond à l'indication de l'axe pilote lors du couplage par facteur (couplage linéaire).

Autres remarques concernant l'utilisation de tables de courbes, la cascade de réducteurs électroniques et leur synchronisation, voir :
Bibliographie :
Description fonctionnelle Fonctions spéciales ; couplages d'axes et ESR (M3), chapitre "Déplacements conjugués et couplage par valeur pilote".

Comportement d'un réducteur électronique en cas de mise sous tension, RESET, changement de mode de fonctionnement, recherche de bloc

- Après la mise sous tension, **aucun** couplage n'est actif.
- Les couplages actifs sont conservés après RESET et changement de mode de fonctionnement.
- À recherche de bloc, les instructions d'activation, de désactivation, d'effacement et de définition du réducteur électronique ne sont ni exécutées ni accumulées, mais sautées.

Variables système du réducteur électronique

À l'aide des variables système du réducteur électronique, vous pouvez déterminer, dans le programme pièce, les états courants d'un groupe d'axes EG et, le cas échéant, déclencher des réactions adéquates.

Les variables système du réducteur électronique sont repérées de la manière suivante :

$AA_EG_ ...

ou

$VA_EG_ ...

Bibliographie :
Manuel des variables système

13.4.3 Désactivation du réducteur électronique (EGOFS, EGOFC)

Il existe 3 variantes pour désactiver un groupe d'axes EG actif.

Programmation

Variante 1 :

Syntaxe	Signification
`EGOFS(axe asservi)`	Le réducteur électronique est désactivé. L'axe asservi est freiné jusqu'à l'arrêt. Cette instruction déclenche un arrêt du prétraitement des blocs.

Variante 2 :

Syntaxe	Signification
`EGOFS(axe asservi,axe pilote 1, …,axe pilote 5)`	Ce paramétrage permet de supprimer **de façon sélective** l'influence de certains axes pilotes sur le déplacement de l'axe asservi.

Un axe pilote au moins doit être spécifié. L'influence des axes pilotes indiqués sur l'axe asservi est supprimée. Cette instruction déclenche un arrêt du prétraitement des blocs. S'il reste des axes pilotes actifs, l'axe asservi continue de se déplacer sous leur influence. Lorsque toutes les influences d'axes pilotes sont désactivées de cette manière, l'axe asservi est freiné jusqu'à l'arrêt.

Variante 3 :

Syntaxe	Signification
`EGOFC(broche asservie1)`	Le réducteur électronique est désactivé. La broche asservie continue de tourner à la vitesse qu'elle a à l'instant de la désactivation. Cette instruction déclenche un arrêt du prétraitement des blocs.

Remarque

Cette variante n'est autorisée que pour des broches.

13.4.4 Suppression de la définition d'un réducteur électronique (EGDEL)

La définition d'un groupe d'axes EG ne peut être effacée que si celui-ci a été désactivé auparavant.

Programmation

Syntaxe	Signification
`EGDEL(axe asservi)`	La définition du couplage du groupe d'axes est effacée. Vous pouvez à nouveau définir des groupes d'axes avec `EGDEF` jusqu'à ce que le nombre maximal de groupes d'axes actif simultanément soit atteint. Cette instruction déclenche un arrêt du prétraitement des blocs.

13.4.5 Avance par tour (G95) / réducteur électronique (FPR)

L'instruction FPR permet également d'indiquer l'axe asservi d'un réducteur électronique en tant qu'axe déterminant l'avance par tours. Dans ce cas, les règles suivantes s'appliquent :

- L'avance dépend de la vitesse de consigne de l'axe asservi du réducteur électronique.
- La vitesse de consigne est calculée à partir des vitesses des broches pilotes et des axes pilotes à valeur modulo (qui ne sont pas des axes à interpolation) et des facteurs de couplage correspondants.
- Les vitesses des axes pilotes linéaires et des axes pilotes sans valeur modulo ainsi que les déplacements forcé de l'axe asservi ne sont pas pris en considération.

13.5 Broche synchrone

En mode de "broche synchrone", il y a une broche pilote (BP) et une broche asservie (BA), qui forment la **paire de broches synchrones**. Quand le couplage est activé (mode "broche synchrone"), la broche asservie suit les déplacements de la broche pilote, en fonction du lien fonctionnel qui a été défini auparavant.

Les paires de broches synchrones peuvent être aussi bien configurées de manière fixe pour chaque machine à l'aide de paramètres machine spécifiques à un canal ou définies en fonction de l'application par le programme pièce de la CNC. Deux paires de broches synchrones sont utilisables simultanément par canal CN.

A partir du programme pièce, le couplage peut être

- défini ou modifié,
- activé,
- désactivé,
- effacé

.

De plus, en fonction de la version du logiciel,

- les conditions de synchronisme peuvent être contrôlées,
- le mode de changement de bloc peut être modifié,
- le type de couplage, soit couplage par valeur de consigne soit couplage par valeur réelle, peut être sélectionné ou le décalage angulaire entre broche pilote et broche asservi peut être prescrit
- à l'activation du couplage, une programmation précédente de la broche asservie peut être reprise
- un écart de synchronisme mesuré ou encore déjà connu peut être corrigé

.

13.5.1 Broche synchrone : Programmation (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

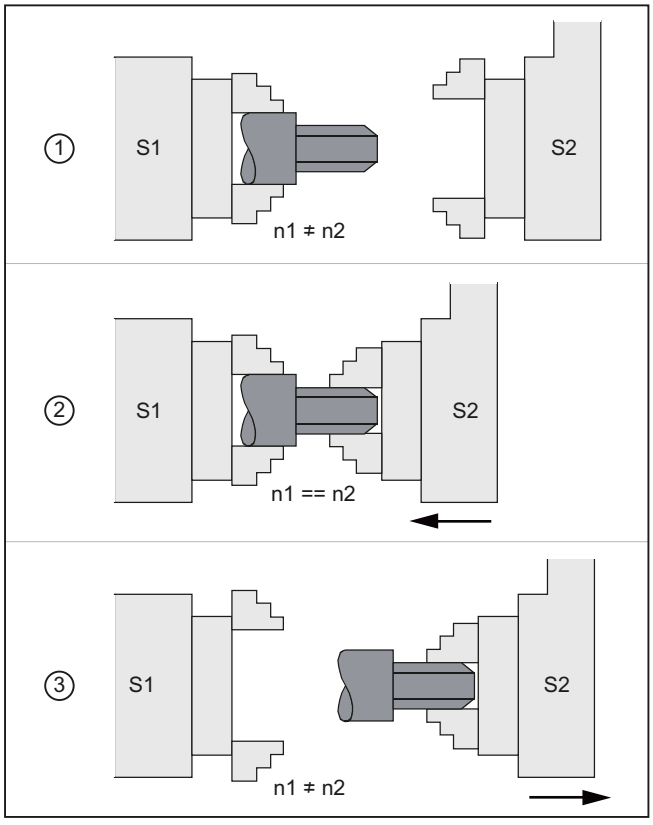
La fonction "Broche synchrone" permet un déplacement à vitesse de rotation synchrone de la broche asservie (BA) et de la broche pilote (BP) avec un rapport de transmission programmable.

Cette fonction dispose des modes suivants :

- Synchronisme de vitesse de rotation ($n_{BA} = n_{BP}$)
- Synchronisme de position ($\phi_{BA} = \phi_{BP}$)
- Synchronisme de position avec décalage angulaire ($\phi_{BA} = \phi_{BP}$)

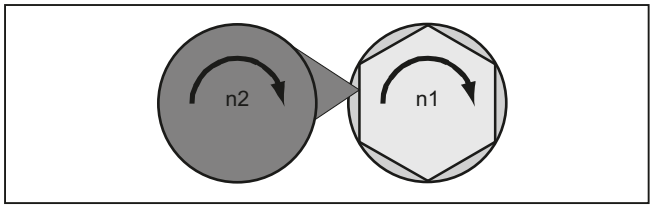
Exemples d'application :

- Transfert de pièce à la volée, par exemple, pour l'usinage de la face arrière, rapport de transmission : 1:1



① Synchronisation de la vitesse de rotation
② Transfert de la pièce
③ Usinage de la face arrière

- Usinage de plusieurs faces (polygonage), synchronisme de vitesse de rotation, rapport de transmission : n_1:n_2



Syntaxe

```
COUPDEF(<BA>,<BP>,<NBA>,<DBP>,<Changement de bloc>,<Mode de
couplage>)
COUPON(<BA>,<BP>,<POSBA>)
COUPONC(<BA>,<BP>)
COUPOF(<BA>,<BP>,<POSBA>,<POSBP>)
COUPOFS(<BA>,<BP>)
COUPOFS(<BA>,<BP>,<POSBA>)
```

```
COUPRES(<BA>,<BP>)
COUPDEL(<BA>,<BP>)
WAITC(<BA>,<Changement de bloc>,<BP>,<Changement de bloc>)
```

Remarque

Notation abrégée

Les instructions COUPOF, COUPOFS, COUPRES et COUPDEL permettent l'utilisation d'une notation abrégée sans indication de la broche pilote.

Signification

COUPDEF :	Définition/modification spécifique à l'utilisateur du couplage
COUPON :	activer le couplage. La broche asservie se synchronise sur la broche pilote à partir de la vitesse de rotation actuelle
COUPONC :	Prise en charge du couplage à l'activation par la programmation précédente de M3 S... ou M4 S.... Une vitesse de rotation différentielle de la broche asservie est aussitôt prise en charge.
COUPOF :	Désactiver le couplage. • avec changement de bloc immédiat : COUPOF(<S2>,<S1>) • Changement de bloc seulement après dépassement de la ou des positions de coupure <POSBA> ou <POSBP> : COUPOF(<S2>,<S1>,<POSBA>) COUPOF(<S2>,<S1>,<POSBA>,<POSBP>)
COUPOFS :	Désactivation d'un couplage avec arrêt de la broche asservie Changement de bloc aussi rapide que possible avec changement de bloc immédiat : COUPOFS(<S2>,<S1>) Changement de bloc uniquement après le passage de la position d'arrêt : COUPOFS(<S2>,<S1>,<POSBA>)
COUPRES :	Remettre à zéro les paramètres de couplage sur les PM et SD configurés
COUPDEL :	effacer le couplage défini par l'utilisateur
WAITC :	Attente de la condition de synchronisme (NOC sont supprimés sur l'interpolateur au changement de bloc)
<BA> :	Désignation de la broche asservie
Paramètres optionnels :	
<BP> :	Désignation de la broche pilote Indication avec numéro de broche : par ex. S2, S1
<NBA>, <DBP> :	Rapport de transmission entre BA et BP. <NBA> / <DBP> = numérateur / dénominateur Réglage par défaut : <NBA> / <DBP> = 1.0 ; indication facultative du dénominateur

<Changement de bloc> :	Mode de changement de bloc Le changement de bloc s'effectue :	
	"NOC"	sur le champ,
	"FINE"	lorsque le "Synchronisme fin" est atteint
	"COARSE"	lorsque le "Synchronisme grossier" est atteint
	"IPOSTOP"	lorsque IPOSTOP est atteint, c.-à-d. quand la valeur de consigne du synchronisme est atteinte (paramétrage par défaut)
	Le mode de changement de bloc est modal.	
<Mode de couplage> :	Type de couplage : couplage entre BA et BP	
	"DV"	Couplage par la valeur de consigne (préréglage)
	"AV"	Couplage par valeur réelle
	"VV"	Couplage de vitesse
	Le type de couplage est modal.	
<POSBA> :	décalage angulaire entre broche pilote et broche asservie	
	Plage de valeurs :	0°… 359,999°
<POSBA>,<POSBP> :	Positions d'arrêt de broche asservie et de broche pilote "Le changement de bloc est débloqué après le passage par POS_{BA}, POS_{BP}."	
	Plage de valeurs :	0°… 359,999°

Exemples

Utilisation de la broche pilote et de la broche asservie

Code de programme	**Commentaire**
	Broche pilote = broche maître = broche 1
	Broche asservie = broche 2
N05 M3 S3000 M2=4 S2=500	La broche pilote tourne à 3000 tr/min, la broche asservie tourne à 500 tr/min.
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	Définition du couplage (peut également être configuré).
...	
N70 SPCON	Placement de la broche pilote en asservissement de position (contrôle de consigne)
N75 SPCON(2)	Broche asservie intégrée dans la régulation de position.
N80 COUPON(S2,S1,45)	Couplage au vol sur position d'offset = 45 degrés.
...	
N200 FA[S2]=100	Vitesse de positionnement = 100 degrés/min
N205 SPOS[2]=IC(-90)	Déplacement forcé de 90 degrés dans le sens négatif.
N210 WAITC(S2,"Fine")	Attente du synchronisme "fin".
N212 G1 X... Y... F...	Usinage
...	

Code de programme	Commentaire
N215 SPOS[2]=IC(180)	Déplacement forcé de 180 degrés dans le sens positif.
N220 G4 S50	Durée d'attente = 50 tours de la broche maître
N225 FA[S2]=0	Activation de la vitesse configurée (PM).
N230 SPOS[2]=IC(-7200)	20 tours. Déplacement dans le sens négatif à la vitesse configurée.
...	
N350 COUPOF(S2,S1)	Découplage au vol, S=S2=3000
N355 SPOSA[2]=0	Arrêt de la broche asservie à 0 degrés.
N360 G0 X0 Y0	
N365 WAITS(2)	Attendre la broche 2.
N370 M5	Arrêter la broche asservie.
N375 M30	

Programmation d'une vitesse de rotation différentielle

Code de programme	Commentaire
	Broche pilote = broche maître = broche 1
	Broche asservie = broche 2
N01 M3 S500	Broche pilote tourne à 500 tr/min.
N02 M2=3 S2=300	Broche asservie tourne à 300 tr/min.
...	
N10 G4 F1	Temporisation de la broche maître.
N15 COUPDEF (S2,S1,-1)	Facteur de couplage avec un rapport de transmission de -1:1
N20 COUPON(S2,S1)	Activer le couplage. La vitesse de rotation de la broche asservie est définie à partir de la vitesse de rotation de la broche pilote et du facteur de couplage.
...	
N26 M2=3 S2=100	Programmation d'une vitesse de rotation différentielle.

Exemples de prise en charge d'un déplacement à la vitesse de rotation différentielle

Activer le 1er couplage avec COUPON, alors que la broche asservie a déjà été programmée

Code de programme	Commentaire
	Broche pilote = broche maître = broche 1
	Broche asservie = broche 2
N05 M3 S100 M2=3 S2=200	La broche pilote tourne à 100 tr/min, la broche asservie tourne à 200 tr/min.
N10 G4 F5	Temporisation = 5 secondes de la broche maître
N15 COUPDEF(S2,S1,1)	Le rapport de transmission BA par rapport à BP est de 1,0 (réglage par défaut).
N20 COUPON(S2,S1)	Couplage au vol sur la broche pilote.
N10 G4 F5	Broche asservie tourne à 100 tr/min.

Activer le 2e couplage avec COUPON, alors que la broche asservie a déjà été programmée

Code de programme	Commentaire
	Broche pilote = broche maître = broche 1
	Broche asservie = broche 2
N05 M3 S100 M2=3 S2=200	La broche pilote tourne à 100 tr/min, la broche asservie tourne à 200 tr/min.
N10 G4 F5	Temporisation = 5 secondes de la broche maître
N15 COUPDEF(S2,S1,1)	Le rapport de transmission BA par rapport à BP est de 1,0 (réglage par défaut).
N20 COUPONC(S2,S1)	Couplage au vol sur la broche pilote et application de la vitesse de rotation précédente à S2.
N10 G4 F5	S2 tourne à 100 tr/min + 200 tr/min = 300 tr/min

Activer le 3e couplage avec COUPON, la broche asservie étant à l'arrêt

Code de programme	Commentaire
	Broche pilote = broche maître = broche 1
	Broche asservie = broche 2
N05 SPOS=10 SPOS[2]=20	Broche asservie S2 en mode de positionnement.
N15 COUPDEF(S2,S1,1)	Le rapport de transmission BA par rapport à BP est de 1,0 (réglage par défaut).
N20 COUPON(S2,S1)	Couplage au vol sur la broche pilote.
N10 G4 F1	Le couplage est fermé, S2 s'arrête à 20 degrés.

Activer le 4e couplage avec COUPONC, la broche asservie étant à l'arrêt

Remarque

Mode de positionnement ou mode axe

Si la broche asservie se trouve en amont du couplage en mode de positionnement ou en mode axe, la broche asservie se comporte de la même façon pour COUPON(<BA>,<BP>) et pour COUPONC(<BA>,<BP>).

Remarque

Broche pilote et mode d'axe

Si la broche pilote se trouve en mode axe avant que le couplage ne soit défini, la valeur limite de vitesse transmise par le paramètre machine suivant agit également après activation du couplage :

PM32000 $MA_MAX_AX_VELO (vitesse d'axe maximale)

Afin d'éviter ce comportement, l'axe doit être commuté en mode broche (M3 S... ou M4 S...) avant la définition du couplage.

plus d'informations

Couplage configuré

La BP et la BA sont définies au moyen de paramètres machine pour le couplage configuré. Les broches configurées ne peuvent pas être modifiées dans le programme pièce. Le paramétrage du couplage peut être réalisé dans le programme pièce avec `COUPDEF` (condition : qu'aucune protection en écriture ne soit définie).

Couplage personnalisé

`COUPDEF` permet de définir un nouveau couplage ou de modifier un couplage existant dans le programme pièce. Si un couplage est déjà actif, il doit d'abord être supprimé avec `COUPDEL` avant la définition d'un nouveau couplage.

Un couplage est de façon complète avec :

```
COUPDEF(<BA>,<BP>,<RBA>,<RBP>, comportement de changement de bloc,
mode de couplage)
```

Broche asservie (BA) et broche pilote (BP)

Les noms d'axe de la BA et de la BP définissent le couplage sans ambiguïté. Ils doivent être programmés dans chaque instruction `COUPDEF`. Les autres paramètres de couplage sont modaux et doivent uniquement être programmés en cas de modification.

Exemple :

```
COUPDEF(S2,S1)
```

Rapport de transmission

Le rapport de transmission est indiqué comme rapport de vitesse de rotation entre BA et BP :

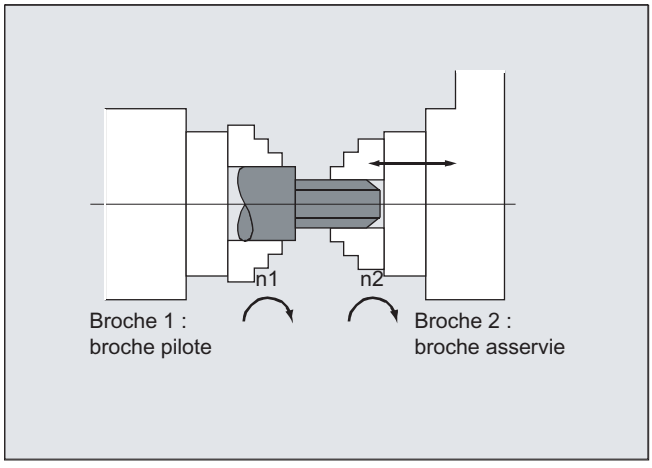
Broche asservie / broche pilote = numérateur / dénominateur

Le numérateur doit être programmé. Le dénominateur ne doit pas obligatoirement être programmé. La valeur par défaut 1.0 est alors définie pour le dénominateur.

Exemple :

Broche asservie S2 et broche pilote S1, rapport de transmission = 1 / 1

```
COUPDEF(S2, S1, 1.0)
```



Remarque

Le rapport de transmission peut également être modifié pendant que le couplage est activé et que les broches sont en rotation.

Mode de changement de bloc NOC, FINE, COARSE, IPOSTOP

Pour la programmation du mode de changement de bloc, la notation abrégée suivante est autorisée :

- `"NO"` : immédiat (préréglage)
- `"FI"` : lorsque le "Synchronisme fin" est atteint
- `"CO"` : lorsque le "Synchronisme grossier" est atteint
- `"IP"` : lorsque IPOSTOP est atteint, c.-à-d. quand la valeur de consigne du synchronisme est atteinte

Type de couplage

Remarque

La modification du type de couplage ne peut se faire que lorsque le couplage est désactivé.

Activation du mode synchrone COUPON, <POSBA>

- Activation du couplage avec un décalage angulaire quelconque entre BP et BA :
  - `COUPON(S2,S1)`
  - `COUPON(S2)`
- Activation du couplage avec un décalage angulaire <`POSBA`>
  `<POSBA>` se rapporte à la position 0° de la broche pilote dans le sens de rotation positif
  Plage de valeurs `<POSBA>` : 0°… 359,999°
  - `COUPON(S2,S1,30)`

Remarque

Le décalage angulaire peut aussi être modifié alors que le couplage est actif.

Positionnement de la broche asservie

Des BA peuvent aussi être positionnées dans la plage ±180° indépendamment de la BP quand le couplage de broches synchrones est activé.

- Positionnement de la BA avec `SPOS`
  Exemple : `SPOS[2]=IC(-90)`
  Plus d'informations sur `SPOS`, voir :
  Bibliographie :
  Manuel de programmation Notions de base

Vitesse de rotation différentielle

En mode de régulation de vitesse de rotation et lorsque le couplage de broches synchrones est activé, une vitesse de rotation différentielle résulte d'une correction avec signe d'une vitesse de rotation de BA suite à un déplacement de BP et d'une vitesse de rotation de BA suite à la programmation de broche :

- Couplage de broches synchrones avec `COUPONC`
- `S<BA>=<Vitesse de rotation>` [`M<BA>=<Sens de rotation>]`

Remarque

Autres conditions à prendre en compte

- Avec le sens de rotation `M3` / `M4`, la vitesse de rotation `S...` doit également être reprogrammée.
- La correction d'une vitesse de rotation de broche (`M<Sens de rotation> S<BA>`) par le déplacement de BP en présence d'un couplage de broches synchrones `COUPONC` ne prend effet que lorsque la correction est débloquée.
- La dynamique de la broche pilote doit être restreinte de sorte que les valeurs limites dynamiques de la broche asservie superposée ne soient pas dépassées.

Plus d'informations sur la vitesse de rotation différentielle, voir :
Bibliographie :
Description fonctionnelle Fonctions d'extension ; broche synchrone (S3)

Vitesse, accélération : AA, ACC, OVRA, VELOLIMA

La programmation de la vitesse et de l'accélération axiales d'une broche asservie s'effectue avec :

- `FA[SPI(S<n>)]` ou `FA[S<n>]` (vitesse axiale)
- `ACC[SPI(S<n>)]` ou `ACC[S<n>]` (accélération axiale)
- `OVRA[SPI(S<n>)]` ou `OVRA[S<n>]` (correction axiale)
- `VELOLIMA[SPI(S<n>)]` ou `VELOLIMA[S<n>]` (augmentation ou réduction de la vitesse axiale)

Avec `<n>` = 1, 2, 3, ... (Numéro des broches asservies)

Bibliographie :
Manuel de programmation Notions de base

Remarque

Une réduction ou augmentation de l'à-coup axial maximal n'a pas d'effet sur les broches.

Plus d'informations sur la dynamique axiale, voir :
Bibliographie :
Description fonctionnelle Fonctions d'extension ; axes rotatifs (R2)

Comportement programmable de changement de bloc WAITC

`WAITC` permet de prédéfinir le comportement de changement de bloc en fonction de différentes conditions de synchronisme (grossier, fin, `IPOSTOP`), p. ex. après une modification des paramètres de couplage ou des opérations de positionnement. Si aucune condition de

synchronisme n'est spécifiée, le comportement de changement de bloc indiqué dans la définition de COUPDEF est appliqué.

Exemples

- Attendre que la condition de synchronisme définie dans FINE soit atteinte pour la broche asservie S2 et celle définie dans COARSE pour la broche asservie S4 :
  WAITC(S2,"FINE",S4,"COARSE")
- Attente de l'atteinte de la condition de synchronisme définie dans COUPDEF : WAITC( )

Désactivation du couplage COUPOF

COUPOF permet de prédéfinir le comportement de désactivation du couplage :

- Désactivation du couplage avec changement de bloc immédiat :
  - COUPOF(S2,S1) (avec indication de la broche pilote)
  - COUPOF(S2) (sans indication de la broche pilote)
- Désactivation du couplage après le passage de la position d'arrêt. Le changement de bloc a lieu après le passage de la position d'arrêt.
  - COUPOF(S2,S1,150) (position d'arrêt BA : 150°)
  - COUPOF(S2,S1,150,30) (position d'arrêt BA : 150°, BP : 30°)

Désactivation du couplage avec arrêt de la broche asservie COUPOFS

COUPOFS permet de prédéfinir le comportement de désactivation du couplage avec arrêt de la broche asservie :

- Désactivation du couplage avec arrêt de la broche asservie et changement de bloc immédiat :
  - COUPOFS(S2,S1) (avec indication de la broche pilote)
  - COUPOFS(S2) (sans indication de la broche pilote)
- Désactivation du couplage après le passage de la position d'arrêt avec arrêt de la broche asservie. Le changement de bloc a lieu après le passage de la position d'arrêt.
  - COUPOFS(S2,S1,150) (position d'arrêt BA : 150°)

Effacement de couplages COUPDEL

COUPDEL permet de supprimer le couplage :

- COUPDEL(S2,S1) (avec indication de la broche pilote)
- COUPDEL(S2) (sans indication de la broche pilote)

Remise à zéro des paramètres de couplage COUPRES

COUPRES active les valeurs du couplage définies dans les paramètres machine et des données de réglage :

- COUPRES(S2,S1) (avec indication de la broche pilote)
- COUPRES(S2) (sans indication de la broche pilote)

Variables système

- État de couplage courant de la broche asservie
  La lecture de l'état de couplage actuel d'une broche asservie peut s'effectuer codée sur bits via :
  `<Wert> = $AA_COUP_ACT[<FS>]`

Bit	<Valeur>	Signification
-	0	Aucun couplage activé
2	4	le couplage de broches synchrones est activé

Remarque

- Toutes les autres valeurs se rapportent au mode axe
- Si la broche est une broche asservie de plusieurs couplages, la valeur fournie en retour est la somme des états de couplage de tous les couplages.

- Décalage angulaire courant
  La lecture du décalage angulaire actuel entre la broche asservie et la broche pilote peut être effectuée via :
  - `$AA_COUP_OFFS[<FS>]` (valeur de consigne du décalage angulaire)
  - `$VA_COUP_OFFS[<FS>]` (valeur effective du décalage angulaire)

  Exemple d'application
  Correction de la différence de décalage angulaire dans le programme CN après l'annulation du mode poursuite :
  Différence de décalage angulaire = décalage angulaire programmé - variable système

Bibliographie

Des informations détaillées sur les variables système figurent dans :

Tables de paramètres Variables système

13.6 Couplage générique (CP...)

Le "couplage générique" est une fonction de couplage générale dans laquelle sont regroupées toutes les caractéristiques de couplage des types de couplage existants (déplacement conjugué, couplage par valeur pilote, réducteur électronique et broche synchrone).

La fonction permet un type de programmation flexible :

- L'utilisateur peut sélectionner librement les caractéristiques de couplage requises pour son application (principe modulaire).
- Chaque caractéristique de couplage est programmable individuellement.
- Les caractéristiques de couplage d'un couplage défini (par ex. facteur de couplage) sont modifiables.
- L'utilisation ultérieure d'autres caractéristiques de couplage est possible.
- Le système de référence des coordonnées de l'axe asservi (système de coordonnées de base ou système de coordonnées machine) est programmable.
- Certaines caractéristiques de couplage peuvent aussi être programmées dans des actions synchrones.
  Bibliographie : Description fonctionnelle Actions synchrones

Remarque

Les appels de couplage antérieurs pour le déplacement conjugué (TRAIL*), le couplage par valeur pilote (LEAD*), les réducteurs électroniques (EG*) et les broches synchrones (COUP*) sont toujours pris en charge au moyen de cycles d'adaptation.

Vue d'ensemble de tous les mots-clés et des caractéristiques de couplage

Le tableau suivant présente une vue d'ensemble de tous les mots-clés du couplage générique et des caractéristiques de couplage ainsi programmables :

Mot-clé	Caractéristique de couplage / signification	Syntaxe
CPDEF	Création d'un module de couplage	CPDEF=(<FAx>)
CPDEL	Suppression d'un module de couplage	CPDEL=(<FAx>)
CPLA	Définition d'un axe pilote	CPLA[<FAx>]=(<LAx>)
CPLDEF	Définition d'un axe pilote et création d'un module de couplage (également possible avec CPDEF + CPLA)	CPLDEF[<FAx>]=(<LAx>) ou CPDEF=(<FAx>) CPLA[<FAx>]=(<LAx>)
CPLDEL	Suppression d'un axe pilote d'un module de couplage (également possible avec CPDEL + CPLA)	CPLDEL[<FAx>]=(<LAx>) ou CPDEL=(<FAx>) CPLA[<FAx>]=(<LAx>)

Mot-clé	Caractéristique de couplage / signification	Syntaxe		
CPON	Activation d'un module de couplage	CPON=(<FAx>)		
CPOF	Désactivation d'un module de couplage	CPOF=(<FAx>)		
CPLON	Activation d'un axe pilote d'un module de couplage	CPLON[<FAx>]=<LAx>		
CPLOF	Désactivation d'un axe pilote d'un module de couplage	CPLOF[<FAx>]=<LAx>		
CPLNUM	Numérateur du facteur de couplage	CPLNUM[FAx,LAx]=<valeur>		
CPLDEN	Dénominateur du facteur de couplage	CPLDEN[FAx,LAx]=<valeur>		
CPLCTID	Numéro de la table de courbes	CPLCTID[FAx,LAx]=<valeur>		
CPLSETVAL	Référence du couplage	CPLSETVAL[FAx,LAx]="<référence du couplage>"		
		"<référence du couplage>":	"CMDPOS"	Couplage par valeur de consigne
			"CMDVEL"	Couplage de vitesse
			"ACTPOS"	Couplage par valeur réelle
CPFRS	Système de référence des coordonnées	CPFRS[FAx]="<référence des coordonnées>"		
		"<référence des coordonnées>":	"BCS"	Système de coordonnées de base
			"MCS"	Système de coordonnées machine
CPBC	Critère de changement de bloc	CPBC[FAx]="<critère de changement de bloc>"		
		"<critère de changement de bloc>":	"NOC"	Le changement de bloc s'effectue indépendamment de l'état du couplage.
			"IPOSTOP"	Le changement de bloc s'effectue lors de la synchronisation de la valeur de consigne.
			"COARSE"	Le changement de bloc s'effectue lors de la synchronisation "grossière" de la valeur réelle.
			"FINE"	Le changement de bloc s'effectue lors de la synchronisation "fine" de la valeur réelle.
CPFPOS + CPON	Position synchrone de l'axe asservi lors de l'activation	CPON=FAx CPFPOS[FAx]=<valeur>		
CPLPOS + CPON	Position synchrone de l'axe pilote lors de l'activation	CPLPOS[FAx,LAx]=<valeur>		

Mot-clé	Caractéristique de couplage / signification	Syntaxe		
CPFMSON	Mode de synchronisation	CPFMSON[FAx]="<mode de synchronisation>"		
		"<mode de synchronisation>":	"CFAST"	Le couplage est fermé avec optimisation du temps.
			"CCOARSE"	Le couplage n'est activé que lorsque la position de l'axe asservi requise conformément à la loi de couplage se trouve dans la zone de la position actuelle de l'axe asservi.
			"NTGT"	L'entredent suivant est accosté avec optimisation du temps.
			"NTGP"	L'entredent suivant est accosté avec optimisation du parcours.
			"NRGT"	Le segment suivant est accosté avec optimisation du temps en fonction du rapport entre le nombre de spires et le nombre de dents.
			"NRGP"	Le segment suivant est accosté avec optimisation du parcours en fonction du rapport entre le nombre de spires et le nombre de dents.
			"ACN"	Seulement pour les axes rotatifs ! L'axe rotatif accoste la position synchrone dans le sens négatif de l'axe. La synchronisation s'effectue aussitôt.
			"ACP"	Seulement pour les axes rotatifs ! L'axe rotatif accoste la position synchrone dans le sens positif de l'axe. La synchronisation s'effectue aussitôt.
			"DCT"	Seulement pour les axes rotatifs ! L'axe rotatif accoste la position synchrone programmée avec optimisation du temps. La synchronisation s'effectue aussitôt.
			"DCP"	Seulement pour les axes rotatifs !

Mot-clé	Caractéristique de couplage / signification	Syntaxe		
				L'axe rotatif accoste la position synchrone programmée avec optimisation du parcours. La synchronisation s'effectue aussitôt.
CPFMON	Comportement de l'axe asservi lors de l'activation	CPFMON[FAx]="<comportement à l'enclenchement>"		
		"<comportement à l'enclenchement>":	"STOP"	Seulement pour les broches ! Un mouvement actif de la broche asservie est arrêté avant l'enclenchement.
			"CONT"	Seulement pour les broches et les axes de l'exécution principale ! Le mouvement actuel de l'axe / la broche asservi(e) est pris en compte dans le couplage en tant que mouvement de départ.
			"ADD"	Seulement pour les broches ! Les composantes de déplacement du couplage agissent en plus du déplacement forcé actuel, c'est-à-dire que le mouvement actuel de l'axe / la broche asservi(e) est conservé en tant que déplacement forcé.
CPFMOF	Comportement de l'axe asservi lors d'une désactivation complète	CPFMOF[FAx]="<comportement à la désactivation>"		
		"<comportement à la désactivation>":	"STOP"	Arrêt de l'axe / la broche asservi(e). Un déplacement forcé actif est freiné jusqu'à l'immobilisation. Le couplage est ouvert ensuite.
			"CONT"	Seulement pour les broches et les axes de l'exécution principale ! La broche asservie continue de tourner à la vitesse qu'elle a à l'instant de la désactivation.
CPFPOS + CPOF	Position de coupure de l'axe asservi lors de la désactivation	CPOF=(FAx) CPFPOS[FAx]=<valeur>		

Mot-clé	Caractéristique de couplage / signification	Syntaxe		
CPMRESET	Comportement du couplage lors d'un RESET	CPMRESET[FAx]="<comportement reset>"		
		"<comportement reset>" :	"NONE"	L'état actuel du couplage est conservé.
			"ON"	Si le module de couplage correspondant est créé, le couplage est activé. Toutes les relations définies pour l'axe pilote sont activées. Cela se produit également si toutes ou des parties de ces relations de l'axe pilote sont déjà actives, c'est-à-dire qu'une resynchronisation a lieu même en présence d'un couplage entièrement activé.
			"OF"	Un déplacement forcé actif est freiné jusqu'à l'immobilisation. Le couplage est désactivé ensuite. Si le module de couplage correspondant a été créé sans définition explicite (CPDEF), le module de couplage est supprimé. Dans le cas contraire, il est tout de même créé, c'est-à-dire qu'il peut tout de même être utilisé.
			"OFC"	Seulement possible pour les broches ! La broche asservie continue de tourner à la vitesse qu'elle a à l'instant de la désactivation. Le couplage est désactivé. Si le module de couplage correspondant a été créé sans définition explicite (CPDEF), le module de couplage est supprimé. Dans le cas contraire, il est tout de même créé, c'est-à-dire qu'il peut tout de même être utilisé.
			"DEL"	Un déplacement forcé actif est freiné jusqu'à l'immobilisation. Le couplage est ensuite désactivé puis supprimé.
			"DELC"	Seulement possible pour les broches ! La broche asservie continue de tourner à la vitesse

Mot-clé	Caractéristique de couplage / signification	Syntaxe		
				qu'elle a à l'instant de la désactivation. Le couplage est désactivé puis supprimé.
CPMSTART	Comportement du couplage au lancement du programme pièce	CPMSTART[FAx]="<comportement au lancement>"		
		"<comportement au lancement>" :	"NONE"	L'état actuel du couplage est conservé.
			"ON"	Couplage activé. Toutes les relations définies pour l'axe pilote sont activées. Cela se produit également si toutes ou des parties de ces relations de l'axe pilote sont déjà actives, c'est-à-dire qu'une resynchronisation a lieu même en présence d'un couplage entièrement activé.
			"OF"	Le couplage est désactivé. Si le module de couplage correspondant a été créé sans définition explicite (CPDEF), le module de couplage est supprimé. Dans le cas contraire, il est tout de même créé, c'est-à-dire qu'il peut tout de même être utilisé.
			"DEL"	Le couplage est désactivé puis supprimé.
CPMPRT	Comportement du couplage au lancement du programme pièce lors d'une recherche en mode test du programme	CPMPRT[FAx]="<comportement au lancement>"		
		"<comportement au lancement>" :	voir CPMSTART	
CPLINTR	Valeur de décalage pour la valeur d'entrée d'un axe pilote	CPLINTR[FAx,LAx]=<valeur>		
CPLINSC	Facteur d'échelle pour la valeur d'entrée d'un axe pilote	CPLINSC[FAx,LAx]=<valeur>		
CPLOUTTR	Valeur de décalage pour la valeur de sortie d'un couplage	CPLOUTTR[FAx,LAx]=<valeur>		
CPLOUTSC	Facteur d'échelle pour la valeur de sortie d'un couplage	CPLOUTSC[FAx,LAx]=<valeur>		
CPSYNCOP	Valeur de seuil pour la synchronisme "grossière" de position	CPSYNCOP[FAx]=<valeur>		
CPSYNFIP	Valeur de seuil pour la synchronisation "fine" de position	CPSYNFIP[FAx]=<valeur>		

Mot-clé	Caractéristique de couplage / signification	Syntaxe		
CPSYNCOP2	Deuxième valeur de seuil pour la synchronisation "grossière" de position	CPSYNCOP2[FAx]=<valeur>		
CPSYNFIP2	Deuxième valeur de seuil pour la synchronisation "fine" de position	CPSYNFIP2[FAx]=<valeur>		
CPSYNCOV	Valeur de seuil pour la synchronisation "grossière" de la vitesse	CPSYNCOV[FAx]=<valeur>		
CPSYNFIV	Valeur de seuil pour la synchronisation "fine" de la vitesse	CPSYNFIV[FAx]=<valeur>		
CPMBRAKE	Comportement de l'axe asservi lors de certains signaux et ordres d'arrêt	CPMBRAKE[FAx]=<valeur codée sur bits>		
CPMVDI	Comportement de l'axe asservi en réaction à certains signaux d'interface CN/AP	CPMVDI[FAx]=<valeur codée sur bits>		
CPMALARM	Inhibition des générations d'alarme spéciales relatives au couplage	CPMALARM[FAx]=<valeur codée sur bits>		
CPSETTYPE	Type de couplage	CPSETTYPE[FAx]="<type de couplage>"		
		"<type de couplage>":	"CP"	Libre programmabilité
			"TRAIL"	Type de couplage "déplacement conjugué"
			"LEAD"	Type de couplage "couplage par valeur pilote"
			"EG"	Type de couplage "réducteur électronique"
			"COUP"	Type de couplage "broche synchrone"

FAx: Axe/broche asservi(e)

LAx: Axe/broche pilote

Remarque

Les caractéristiques de couplage qui ne sont pas programmées de manière explicite (dans le programme pièce ou dans les actions synchrones) prennent effet avec leur réglage par défaut.

En fonction du paramétrage du mot-clé CPSETTYPE, des caractéristiques de couplage préréglées peuvent aussi prendre effet à la place des réglages par défaut (CPSETTYPE="CP").

Bibliographie

Pour obtenir des informations détaillées sur le couplage générique, voir :

- Description fonctionnelle Fonctions spéciales ; M3 : Couplages d'axes, chapitre : "Couplage générique"

13.7 Couplage maître/esclave (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

Le "couplage maître/esclave" permet d'effectuer :

- le couplage des axes esclaves avec l'axe maître lorsque les axes concernés sont à l'arrêt,
- le couplage et le découplage de broches en mode asservissement de vitesse qui sont **en rotation**,
- la configuration dynamique.

Remarque

Mode Positionnement

Pour les axes et les broches en mode de positionnement, la fermeture et l'ouverture du couplage ne s'opèrent qu'à l'arrêt.

Syntaxe

```
MASLON(<esclave_1>,<esclave_2>...)
MASLOF(<esclave_1>,<esclave_2>...)
MASLOFS(<esclave_1>,<esclave_2>...)
```

Configuration dynamique :

```
MASLDEF(<esclave_1>,<esclave_2>,...,<maître>)
MASLDEL(<esclave_1>,<esclave_2>...)
```

Signification

MASLON :	Activation d'un couplage maître/esclave temporaire	
	<esclave_x>,... :	axe esclave 1 ... n
MASLOF :	Ouverture d'un couplage maître/esclave actif	
	<esclave_1>,... :	axe esclave 1 ... n
MASLOFS :	Ouverture du couplage maître/esclave et freinage automatique de la broche asservie (voir Remarque "Comportement au couplage de broches en mode asservissement de vitesse")	
	<esclave_1>,... :	axe esclave 1 ... n
MASLDEF :	Création/modification du couplage maître/esclave à partir du programme pièce	
	<esclave_1>,... :	axe esclave 1 ... n
	<maître> :	Axe maître
MASLDEL :	Ouverture du couplage maître/esclave et suppression de la définition du groupe	
	<esclave_1>,... :	axe esclave 1 ... n
	Remarque : Les définitions maître/esclave configurées dans les paramètres machine sont conservées.	

Remarque

Comportement au couplage de broches en mode asservissement de vitesse

Pour les broches en mode asservissement de vitesse, le comportement au couplage est défini explicitement par MASLON, MASLOF, MASLOFS et MASLDEL dans le paramètre machine suivant :

PM37263 $MA_MS_SPIND_COUPLING_MODE

Avec le réglage par défaut PM37263 = 0, la fermeture et l'ouverture du couplage des axes esclaves s'opèrent uniquement lorsque les axes concernés sont immobilisés. MASLOFS correspond à MASLOF.

Lorsque PM37263 = 1, l'instruction de couplage est exécutée immédiatement, et donc même en mouvement. MASLON ferme immédiatement le couplage et MASLOFS ou MASLOF l'ouvre immédiatement. Les broches esclaves en rotation à cet instant conservent leurs vitesses de rotation pour MASLOF jusqu'à la reprogrammation des vitesses. Au contraire, avec MASLOFS, elles sont automatiquement freinées.

Remarque

L'arrêt implicite du prétraitement des blocs n'est pas exécuté avec MASLOF/MASLOFS . Du fait de l'absence de cet arrêt du prétraitement des blocs, les variables système $P ne fournissent pas de valeurs actualisées pour les axes esclaves jusqu'au moment de la reprogrammation.

Remarque

Pour l'axe esclave, PRESETON permet de synchroniser la valeur réelle sur la même valeur que l'axe maître. Cela exige une brève désactivation du couplage permanent maître/esclave pour que la valeur réelle de l'axe esclave non référencé puisse être définie à la valeur de l'axe maître avec un POWER ON. Le couplage permanent est ensuite rétabli.

Le couplage permanent maître/esclave est activé par le réglage suivant du paramètre machine :
PM37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1
Ce réglage n'a pas d'effet sur les instructions de couplage temporaire.

Exemples

Exemple 1 : Forçage des valeurs réelles pour l'axe esclave d'un couplage maître/esclave

Pour un couplage maître/esclave permanent, PRESETON permet de forcer la valeur réelle de l'axe esclave sur la valeur de l'axe maître.

Code de programme	Commentaire
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; Désactiver le couplage permanent de l'axe esclave
NEWCONF	; Activer la modification des paramètres machine
STOPRE	

Code de programme	Commentaire
MASLOF(Y1)	; désactiver le couplage temporaire
PRESETON(AX2,$VA_IM(M_AX))	; Valeur réelle de l'axe esclave = valeur réelle de l'axe maître
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; Activer le couplage permanent de l'axe esclave
NEWCONF	; Activer la modification des paramètres machine

Exemple 2 : configuration dynamique d'un couplage maître/esclave

Pour que le couplage avec une autre broche puisse se réaliser après une rotation du conteneur d'axes, il convient de supprimer le couplage précédent et sa configuration et de reconfigurer le nouveau couplage.

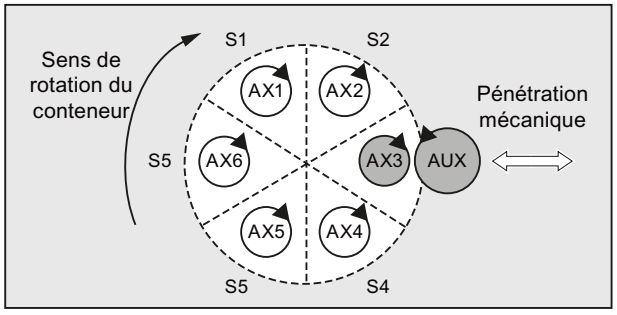


Figure 13-1 Avant la rotation du conteneur d'axes

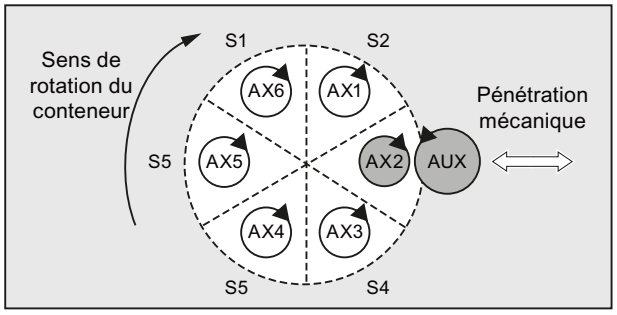


Figure 13-2 Après rotation du conteneur d'axes de la valeur d'un emplacement :

Code de programme	Commentaire
MASLDEF(AUX,S3)	; AUX : esclave, S3 : maître = AX3
MASLON(AUX)	; Activer le couplage
M3=3 S3=4000	; Rotation maître
MASLDEL(AUX)	; Séparer et supprimer couplage
AXCTSWE(CT1)	; Validation de la rotation du conteneur d'axes
MASLDEF(AUX,S3)	; AUX : esclave, S3 : maître = AX2

Bibliographie

- Description fonctionnelle Fonctions spéciales ; Chapitre TE3 : Couplage vitesse de rotation / couple, maître/esclave"
- Description fonctionnelle Fonctions d'extension ; chapitre "B3 : Système décentralisé - uniquement 840D sl" > "Link NCU" > "Conteneur d'axes"

Actions synchrones

14

14.1 Définition d'une action synchrone

Une action synchrone se définit dans un bloc de programme pièce. Aucune autre instruction que les constituants de l'action synchrone ne doit être programmée dans ce bloc.

Une action synchrone se compose des éléments suivants :

Validité, nº d'identification (en option)	Partie condition (en option)			Partie action		
	Fréquence	Instruction G (en option)	Condition	Mot-clé	Instruction G (en option)	Actions
--- [1] ID=<nº> IDS=<nº>	--- [1] WHENEVER FROM WHEN EVERY	G...	Expression logique	DO	G...	Action 1 ... Action n

[1] non programmée

Syntaxe

```
DO <action 1> ... <action n>
<fréquence> [<fonction G>] <condition> DO <action 1> ... <action n>
ID=<n°> <fréquence> [<fonction G>] <condition> DO <action 1> ...
<action n>
IDS=<n°> <fréquence> [<fonction G>] <condition> DO <action 1> ...
<action n>
```

Bibliographie

Pour une description détaillée de la fonctionnalité des actions synchrones, voir :

Description fonctionnelle Actions synchrones

Oscillation 15

15.1 Oscillation asynchrone (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Un axe d'oscillation va et vient, avec une avance donnée, entre deux points d'inversion de sens 1 et 2 jusqu'à désactivation de l'oscillation.

Pendant l'oscillation, d'autres axes peuvent être interpolés librement. Une pénétration continue peut être obtenue par un déplacement de contournage ou avec un axe de positionnement. Il n'existe toutefois **aucun lien** entre l'oscillation et le mouvement de pénétration.

Caractéristiques de l'oscillation asynchrone

- L'oscillation asynchrone agit au-delà des limites du bloc et est spécifique à un axe.
- Une activation de l'oscillation, synchronisée avec le bloc, est assurée par le programme pièce.
- Une interpolation commune de plusieurs axes et une correction de la course d'oscillation ne sont pas possibles.

Programmation

Les instructions suivantes permettent d'activer et d'agir sur l'oscillation asynchrone à partir du programme pièce, en fonction de l'exécution du programme CN.

Les valeurs programmées sont inscrites dans les données de réglage correspondantes pendant l'exécution du programme et restent actives jusqu'à la prochaine modification.

Syntaxe

```
OSP1[<axe>]=<valeur> OSP2[<axe>]=<valeur>
OST1[<axe>]=<valeur> OST2[<axe>]=<valeur>
FA[<axe>]=<valeur>
OSCTRL[<axe>]=(<options de réglage>,<option de désactivation>)
OSNSC[<axe>]=<valeur>
OSE[<axe>]=<valeur>
OSB[<axe>]=<valeur>
OS[<axe>]=1
OS[<axe>]=0
```

Signification

`<axe>` :	Nom de l'axe d'oscillation		
`OS` :	Activer/désactiver l'oscillation		
	Valeur :	1	**Activer** l'oscillation
		0	**Désactiver** l'oscillation
`OSP1` :	Définir la position à partir du point d'inversion de sens 1		

OSP2 :	Définir la position à partir du point d'inversion de sens 2 **Remarque :** si un déplacement relatif est actif, la position est calculée de manière relative par rapport à la dernière position d'inversion correspondante programmée dans le programme CN.		
OST1 :	Définir le temps d'arrêt au point d'inversion de sens 1 en [s]		
OST2 :	Définir le temps d'arrêt au point d'inversion de sens 2 en [s]		
	<valeur> :	-2	l'interpolation est poursuivie sans attente de l'arrêt précis
		-1	attente de l'arrêt précis grossier
		0	attente de l'arrêt précis fin
		>0	attente de l'arrêt précis fin puis écoulement du temps d'arrêt indiqué **Remarque :** l'unité du temps d'arrêt est identique à celle programmée avec G4.
FA :	Définir la vitesse d'avance La vitesse d'avance appliquée ici est celle définie pour l'axe de positionnement. Si aucune vitesse d'avance n'a été définie, c'est la valeur qui figure dans le paramètre machine qui s'applique.		
OSCTRL :	Indique les options d'activation et de désactivation Les valeurs d'option 0 - 3 verrouillent le comportement au point d'inversion à la désactivation. L'une des variantes 0 - 3 peut être sélectionnée. Une combinaison quelconque des paramétrages restant avec la variante sélectionnée est ensuite possible. La combinaison de plusieurs options s'effecue par insertion du signe plus (+).		
	<valeur> :	0	A la désactivation de l'oscillation, arrêt au point d'inversion suivant (paramétrage par défaut) **Remarque :** ceci n'est possible que par réinitialisation des valeurs 1 et 2.
		1	À la désactivation de l'oscillation, arrêt au point d'inversion de sens 1
		2	À la désactivation de l'oscillation, arrêt au point d'inversion de sens 2
		3	À la désactivation de l'oscillation, ne pas accoster de point d'inversion de sens, si aucune passe à lécher n'est programmée
		4	Après les passes à lécher, accoster la position de fin
		8	Si l'oscillation est interrompue par effacement de la distance restant à parcourir, le traitement des passes à lécher doit ensuite être terminé et, le cas échéant, la position finale accostée.
		16	Si l'oscillation est interrompue par effacement de la distance restant à parcourir, la position d'inversion correspondante doit être accostée comme dans le cas de la désactivation.
		32	L'avance modifiée ne prend effet qu'à partir du prochain point d'inversion de sens
		64	FA est égal à 0, FA = 0 : le forçage du déplacement est activé FA est différent de 0, FA <> 0 : la correction de vitesse est activée
		128	Dans le cas d'un axe rotatif DC (course minimale)
		256	Passe à lécher exécutée sous forme de course double (standard). 1=passe à lécher exécutée sous forme de course simple.
		512	Accoster d'abord la position de départ
OSNSC :	Définir le nombre de passes à lécher		

OSE :	Définir la position finale à accoster (dans le SCP) après désactivation de l'oscillation **Remarque :** la programmation de "OSE" active de façon implicite l'option 4 pour "OSCTRL".
OSB :	Définir la position de départ à accoster (dans le SCP) avant activation de l'oscillation La position de départ est accostée avant le point d'inversion de sens 1. Si la position de départ coïncide avec la position d'inversion 1, la position d'inversion 2 est ensuite accostée. Lorsque la position de départ est atteinte, aucun temps d'arrêt n'a d'effet, même lorsque la position de départ coïncide avec la position d'inversion 1, par contre un arrêt précis fin est attendu. La condition de précision réglée est respectée. **Remarque :** afin que la position de départ puisse être accostée, le bit 9 doit être mis à 1 dans la donnée de réglage SD43770 $SA_OSCILL_CTRL_MASK.

Exemples

Exemple 1 : l'axe d'oscillation doit osciller entre deux points d'inversion

L'axe d'oscillation Z doit osciller entre la position 10 et la position 100. Le point d'inversion de sens 1 doit être accosté avec un arrêt précis fin, le point d'inversion de sens 2 avec un arrêt précis grossier. L'avance de l'axe d'oscillation doit être égale à 250. A la fin de l'usinage, 3 passes à lécher doivent être exécutées et l'axe d'oscillation doit atteindre la position finale 200. L'avance pour l'axe de pénétration doit être égale à 1, la fin de la pénétration en direction X doit être atteinte à la position 15.

Code de programme	Commentaire
WAITP(X,Y,Z)	; Position de départ.
G0 X100 Y100 Z100	; Commutation en mode d'axe de positionnement.
WAITP(X,Z)	
OSP1[Z]=10 OSP2[Z]=100	; Point d'inversion de sens 1, point d'inversion de sens 2.
OSE[Z]=200	; Position finale.
OST1[Z]=0 OST2[Z]=-1	; Temps d'arrêt sur U1 : Arrêt précis fin
	; Temps d'arrêt sur U2 : arrêt précis grossier
FA[Z]=250 FA[X]=1	; Avance axe d'oscillation, avance axe de pénétration.
OSCTRL[Z]=(4,0)	; Options de réglage.
OSNSC[Z]=3	; 3 passes à lécher.
OS[Z]=1	; Démarrer l'oscillation.
WHEN $A_IN[3]==TRUE DO DELDTG(X)	; Effacement de la distance restant à parcourir.
POS[X]=15	; Position de départ axe X
POS[X]=50	; Position finale axe X.
OS[Z]=0	; Arrêter l'oscillation.
M30	

Remarque

La séquence d'instructions "OSP1[Z]=..." à "OSNCS[Z]=..." peut également être programmée dans un bloc.

Exemple 2 : oscillation avec modification en ligne de la position d'inversion

Les données de réglage nécessaires pour l'oscillation asynchrone peuvent être réglées dans le programme pièce.

Si les données de réglage sont écrites directement dans le programme pièce, la modification prend effet dès le prétraitement. Le comportement synchrone peut être obtenu par un arrêt du prétraitement (STOPRE).

Code de programme	Commentaire
$SA_OSCILL_REVERSE_POS1[Z]=-10	
$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0	; Arrêt de l'axe de pénétration lorsque la valeur réelle de l'axe d'oscillation a dépassé le point d'inversion.
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; Activer l'oscillation.
OS[Z]=0	; Désactiver l'oscillation.
M30	

Informations complémentaires

Axe d'oscillation

Pour les axes d'oscillation, il est convenu que :

- chaque axe peut être utilisé comme axe d'oscillation
- Plusieurs axes d'oscillation peuvent être actifs simultanément (au maximum : le nombre des axes de positionnement).
- Quelle que soit la fonction G momentanément active dans le programme, l'interpolation linéaire G1 est toujours active pour l'axe d'oscillation.

L'axe d'oscillation peut être :

- un axe d'entrée pour la transformation dynamique
- un axe directeur pour les axes Gantry et conjugués
- déplacé :
  - Sans limitation des à-coups "BRISK"
    ou
  - Avec limitation des à-coups "SOFT"
    ou
  - avec une ligne caractéristique d'accélération coudée (comme les axes de positionnement)

Points d'inversion de sens

Pour déterminer les points d'inversion de l'oscillation, il convient de tenir compte des décalages actuels ci-après :

- Déclaration en absolu
  "OSP1[Z]=<valeur>"
  Position du point d'inversion = Somme des décalages + valeur programmée
- Déclaration en relatif
  "OSP1[Z]=IC(<valeur>)"
  Position du point d'inversion = point d'inversion 1 + valeur programmée

Exemple :

Code de programme

```
N10 OSP1[Z]=100 OSP2[Z]=110
...
N40 OSP1[Z]= IC(3)
```

WAITP

Pour faire osciller un axe géométrique, il faut débloquer ce dernier pour l'oscillation avec "WAITP".

Lorsque l'oscillation est terminée, "WAITP" l'axe d'oscillation est à nouveau défini comme axe de positionnement et peut à nouveau être utilisé normalement.

Oscillation avec actions synchrones au déplacement et temps d'arrêt

Après écoulement des temps d'arrêt réglés, le changement de bloc interne a lieu pendant l'oscillation (visible aux nouvelles distances restant à parcourir des axes). La fonction de désactivation est vérifiée lors du changement de bloc. Elle est déterminée après le réglage de la séquence de déplacements (`OSCTRL`). *Ce comportement temporel peut être modifié par la correction de l'avance.*

Il est possible qu'une course d'oscillation soit encore effectuée ensuite avant le démarrage des passes à lécher ou l'accostage de la position finale. *On a l'impression que le comportement à la désactivation est modifié. Ceci n'est toutefois pas le cas.*

15.2 Oscillation commandée par des actions synchrones (OSCILL)

Avec ce type d'oscillation, le mouvement de pénétration n'est autorisé qu'aux points d'inversion de sens ou dans des zones d'inversion de sens bien définies.

Selon les besoins, l'oscillation peut

- soit être poursuivie
- soit être arrêtée jusqu'à ce que la pénétration soit entièrement effectuée.

Syntaxe

1. **Fixer les paramètres pour l'oscillation**
2. **Définir les actions synchrones au déplacement**
3. **Affecter les axes, fixer la profondeur de pénétration**

Signification

`OSP1[<Axe d'oscillation>]=`	Position du point d'inversion de sens 1
`OSP2[<Axe d'oscillation>]=`	Position du point d'inversion de sens 2
`OST1[<Axe d'oscillation>]=`	Temps d'arrêt au point d'inversion de sens 1 en secondes
`OST2[<Axe d'oscillation>]=`	Temps d'arrêt au point d'inversion de sens 2 en secondes
`FA[<Axe d'oscillation>]=`	Avance de l'axe d'oscillation
`OSCTRL[<Axe d'oscillation>]=`	Options d'activation ou de désactivation
`OSNSC[<Axe d'oscillation>]=`	Nombre de passes à lécher
`OSE[<Axe d'oscillation>]=`	Position finale
`WAITP(<Axe d'oscillation>)`	Libération de l'axe pour l'oscillation

Affectation des axes, profondeur de pénétration

```
OSCILL[<Axe d'oscillation>]=(<Axe de pénétration 1>,<Axe de
pénétration 2>,<Axe de pénétration 3>)
POSP[<Axe de pénétration>]=(<Position de fin>,<Longueur
partielle>,<Mode>)
```

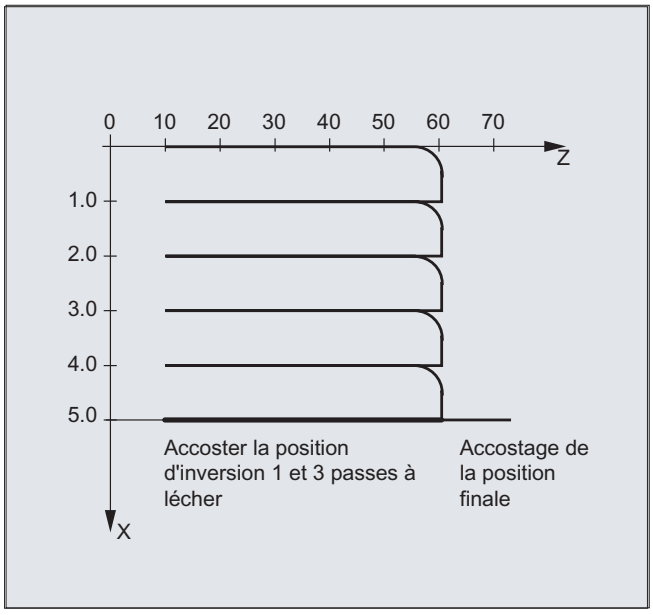
`OSCILL` :	Affecter l'axe ou les axes de pénétration à l'axe d'oscillation
`POSP` :	Définir la pénétration totale et partielle (voir chapitre Gestion des fichiers et programmes)
`Position de fin` :	Position de fin pour l'axe de pénétration, quand toutes les pénétrations partielles ont été effectuées.
`Longueur partielle` :	Valeur de la pénétration partielle au point d'inversion de sens/dans la zone d'inversion de sens
`Mode` :	Division de la pénétration totale en pénétrations partielles = deux dernières pénétrations partielles identiques (préréglage) ; = toutes les pénétrations partielles identiques

Actions synchrones au déplacement

WHEN… … DO	Quand..., alors...
WHENEVER … DO	À chaque fois que..., alors...

Exemple

Il ne doit pas y avoir de pénétration au point d'inversion de sens 1. Au point d'inversion de sens 2, la pénétration doit déjà commencer à la distance ii2 du point d'inversion de sens 2 et l'axe d'oscillation ne doit pas attendre la fin de la pénétration partielle. L'axe Z est axe d'oscillation et l'axe X axe de pénétration.



1. Paramètres pour l'oscillation

Code de programme	Commentaire
DEF INT ii2	; Définir la variable pour la zone d'inversion de sens 2
OSP1[Z]=10 OSP2[Z]=60	; Définir les points d'inversion de sens 1 et 2
OST1[Z]=0 OST2[Z]=0	; Point d'inversion 1 : Arrêt précis fin Point d'inversion 2 : Arrêt précis fin
FA[Z]=150 FA[X]=0.5	; Avance de l'axe d'oscillation Z, avance de l'axe de pénétration X
OSCTRL[Z]=(2+8+16,1)	; Désactiver l'oscillation au point d'inversion de sens 2 ; après effacement de la distance restant à parcourir, passes à lécher et accostage de la position de fin ; après effacement de la distance restant à parcourir, accostage de la position d'inversion de sens correspondante
OSNC[Z]=3	; Passes à lécher
OSE[Z]=70	; Position de fin = 70
ii2=2	; Régler la zone d'inversion de sens
WAITP(Z)	; Autorise l'oscillation pour l'axe Z

2. Action synchrone au déplacement

Code de programme	Commentaire
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z] DO -> $AA_OVR[X]=0 $AC_MARKER[0]=0	; À chaque fois que la position courante de l'axe d'oscillation Z est inférieure au début de la zone d'inversion de sens 2 dans le SCM, mettre la correction axiale de l'axe de pénétration X à 0 % et le mémento avec l'indice 0 à la valeur 0.
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0	; À chaque fois que la position courante de l'axe d'oscillation Z est supérieure à la position d'inversion de sens 2 dans le SCM, mettre la correction axiale de l'axe d'oscillation Z à 0 %.
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[0]=1	; À chaque fois que la distance restant à parcourir pour l'approche partielle est égale à 0, mettre le mémento avec l'indice 0 à la valeur 1.
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100	; À chaque fois que le mémento avec l'indice 0 est égal à 1, mettre la correction axiale de l'axe d'approche X à 0 %. Ceci permet d'empêcher une approche prématurée (axe d'oscillation Z n'a pas encore quitté la zone d'inversion 2, l'axe d'approche X est toutefois prêt pour une nouvelle approche). Modifier la correction axiale de l'axe d'oscillation Z de 0 % (action de la 2e action synchrone) de nouveau à la valeur 100 % pour le déplacement.

-> doit être programmé dans un bloc

3. Démarrage de l'oscillation

Code de programme	Commentaire
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; Démarrage des axes L'axe X est affecté comme axe de pénétration à l'axe d'oscillation Z. L'axe X doit rejoindre la position de fin 5 par incréments de 1.
M30	; Fin du programme

plus d'informations

1. **Fixer les paramètres pour l'oscillation**
   Les paramètres pour l'oscillation sont à fixer dans le programme avant le bloc de déplacement qui contient d'une part la correspondance entre l'axe de pénétration et l'axe d'oscillation et d'autre part l'instruction avec les paramètres de pénétration (voir "Oscillation asynchrone").
2. **Définir les actions synchrones au déplacement**
   Par le biais des actions synchrones, il est possible de :
   Inhiber le mouvement de pénétration, jusqu'à ce que l'axe d'oscillation se trouve à l'intérieur d'une zone d'inversion
   (ii1, ii2) ou à un point d'inversion (U1, U2).
   Arrêter l'oscillation pendant la pénétration au point d'inversion de sens.
   Redémarrer l'oscillation lorsque la pénétration partielle est **terminée**.
   Définir le lancement de la pénétration partielle suivante .
3. **Affecter les axes d'oscillation et de pénétration** et définir **la pénétration totale et partielle**.

Fixer les paramètres pour l'oscillation

Affectation des axes d'oscillation et de pénétration : OSCILL

```
OSCILL[<Axe d'oscillation>]=(<Axe de pénétration1>,<Axe de
pénétration2>,<Axe de pénétration3>)
```

L'instruction "OSCILL" permet d'effectuer les affectations d'axe et le démarrage de l'oscillation.

Trois axes de pénétration peuvent être affectés au maximum à un axe d'oscillation.

Remarque

Avant le démarrage de l'oscillation, les conditions synchrones pour le comportement des axes doivent être définies.

Définition des pénétrations : POSP

```
POSP[<Axe de pénétration>]=(<Position de fin>,<Longueur
partielle>,<Mode>)
```

Avec l'instruction "POSP", on fournit les indications suivantes à la commande :

- la pénétration totale (en se basant sur la position de fin)
- la valeur de la pénétration partielle au point d'inversion de sens ou dans la zone d'inversion de sens
- le comportement lorsque la position de fin est atteinte (avec le mode)

Mode = 0	Pour les deux dernières pénétrations partielles, la distance restant à parcourir jusqu'à la destination est divisée en deux incréments identiques (préréglage).
Mode = 1	Toutes les pénétrations partielles sont identiques. Elles sont calculées à partir de la pénétration totale.

Définir les actions synchrones au déplacement

De façon tout à fait générale, on utilise pour l'oscillation les actions synchrones au déplacement énumérées ci-après.

Des exemples sont donnés qui pourront vous servir de base pour réaliser vos propres programmes d'oscillation.

Remarque

Dans certains cas particuliers, les conditions synchrones peuvent également être programmées différemment.

Mots-clés

`WHEN … DO …`	Quand..., alors...
`WHENEVER … DO`	À chaque fois que..., alors...

Fonctions

Avec les instructions décrites en détail ci-après, vous pouvez réaliser les fonctions suivantes :

1. Pénétration au point d'inversion de sens.
2. Pénétration dans la zone d'inversion de sens.
3. Pénétration aux deux points d'inversion de sens.
4. Arrêt de l'oscillation au point d'inversion de sens.
5. Redémarrage de l'oscillation.
6. Inhibition puis démarrage de la pénétration partielle suivante.

Toutes les actions synchrones mentionnées ici à titre d'exemple sont basées sur les hypothèses suivantes :

- point d'inversion de sens 1 < point d'inversion de sens 2
- Z = axe d'oscillation
- X = axe de pénétration

Remarque

Pour plus d'explications, voir le chapitre Actions synchrones au déplacement.

Affecter les axes d'oscillation et de pénétration et définir la pénétration totale et partielle.

Pénétration dans la zone d'inversion de sens

Le mouvement de pénétration doit débuter à l'intérieur d'une zone d'inversion de sens, avant que le point d'inversion de sens ne soit atteint.

Ces actions synchrones inhibent le mouvement de pénétration jusqu'à ce que l'axe d'oscillation se trouve dans une zone d'inversion de sens.

Avec les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes :

Zone d'inversion de sens 1 :

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X] = 0
```

À chaque fois que la position courante de l'axe d'oscillation est supérieure au début de la zone d'inversion de sens 1 dans le SCM, mettre la correction axiale de l'axe de pénétration à 0 %.

Zone d'inversion de sens 2 :

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X] = 0
```

À chaque fois que la position courante de l'axe d'oscillation est inférieure au début de la zone d'inversion de sens 2 dans le SC, mettre la correction axiale de l'axe de pénétration à 0 %.

Pénétration au point d'inversion de sens

Tant que l'axe d'oscillation n'a pas atteint le point d'inversion, l'axe de pénétration n'effectue aucun déplacement.

Avec les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes :

Zone d'inversion de sens 1 :

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

À chaque fois que la position courante de l'axe d'oscillation Z est supérieure ou inférieure à la position du point d'inversion de sens 1 dans le SCM, mettre la correction axiale de l'axe de pénétration X à 0 % et la correction axiale de l'axe d'oscillation Z à 100 %.

Zone d'inversion de sens 2 :

Pour le point d'inversion 2 :

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

À chaque fois que la position courante de l'axe d'oscillation Zu est supérieure ou inférieure à la position du point d'inversion de sens 2 dans le SCM, mettre la correction axiale de l'axe de pénétration X à 0 % et la correction axiale de l'axe d'oscillation Z à 100 %.

Arrêt de l'oscillation au point d'inversion de sens

L'axe d'oscillation s'arrête au point d'inversion de sens, le mouvement de pénétration commence au même moment. L'oscillation est poursuivie lorsque la pénétration est entièrement effectuée.

Cette action synchrone peut également être utilisée pour redémarrer le mouvement de pénétration si celui-ci a été arrêté par une action synchrone précédente encore active.

Avec les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes :

Zone d'inversion de sens 1 :

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

À chaque fois que la position courante de l'axe d'oscillation est égale à la position d'inversion de sens 1 dans le SCM, mettre la correction axiale de l'axe d'oscillation à 0 % et la correction axiale de l'axe de pénétration à 100 %.

Zone d'inversion de sens 2 :

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

À chaque fois que la position courante de l'axe d'oscillation Zu est égale à la position d'inversion de sens 2 dans le SCM, mettre la correction axiale de l'axe d'oscillation X à 0 % et la correction axiale de l'axe de pénétration à 100 %.

Exploitation en ligne du point d'inversion de sens

Si une variable d'exécution marquée par `$$` figure du côté droit de la comparaison, les deux variables seront exploitées en continu à la période d'appel de l'interpolateur et comparées entre elles.

Remarque

Plus d'informations à ce sujet, voir chapitre "Actions synchrones au déplacement".

Redémarrage de l'oscillation

Cette action synchrone sert à reprendre le mouvement d'oscillation, quand la pénétration partielle est terminée.

Avec les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes :

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]= 100
```

À chaque fois que la distance restant à parcourir pour la pénétration partielle de l'axe de pénétration X est égale à zéro dans le SCM, mettre la correction axiale de l'axe d'oscillation à 100 %.

Pénétration partielle suivante

Lorsque la pénétration a été effectuée, il convient d'éviter que la pénétration partielle suivante ne démarre trop tôt.

On utilise pour ce faire un mémento spécifique au canal (`$AC_MARKER[Index]`) qui est mis à 1 à la fin de la pénétration partielle (distance partielle restant à parcourir 0) et qui est effacé à la sortie de la zone d'inversion de sens. Une action synchrone empêche ensuite le mouvement de pénétration suivant.

Selon les hypothèses formulées (voir ci-dessus), les instructions sont les suivantes pour le point d'inversion de sens 1 par exemple :

1. Définir les marques :

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1] = 1
```

À chaque fois que la distance restant à parcourir pour la pénétration partielle de l'axe de pénétration X est égale à zéro dans le SCM, mettre le mémento avec l'indice 1 à la valeur 1.

2. Effacer les marques

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1] = 0
```

Chaque fois que la position actuelle de l'axe d'oscillation Z est supérieure ou inférieure à la position du point d'inversion de sens 1 dans le SCM, mettre le mémento 1 à 0.

3. Inhiber la pénétration

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X] = 0
```

À chaque fois que le mémento 1 est égal, mettre la correction axiale de l'axe de pénétration X à 0 %.

Poinçonnage et grignotage 16

16.1 Activation/désactivation

16.1.1 Activation/désactivation du poinçonnage et du grignotage (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Activation/désactivation du poinçonnage ou du grignotage

PON et SON permettent d'activer la fonction de poinçonnage ou de grignotage. SPOF met fin à toutes les fonctions spécifiques au poinçonnage et au grignotage. Les instructions modales PON et SON s'excluent mutuellement, c'est-à-dire que PON désactive SON et vice versa.

Poinçonnage/grignotage à la période d'appel de l'interpolateur

Avec SONS et PONS, vous activez aussi respectivement la fonction poinçonnage et la fonction grignotage.

Contrairement au pilotage du poinçon au niveau de l'interpolation actif avec SON/PON, le pilotage du poinçon s'effectue ici par le biais de signaux au niveau servo. Ceci permet de travailler avec une cadence de frappe plus rapide et donc une meilleure productivité.

Pendant le traitement des signaux à la période d'appel de l'interpolateur, toutes les fonctions susceptibles de modifier la position des axes de grignotage ou de poinçonnage sont verrouillées (par ex. déplacement par manivelle électronique, modifications de frames via AP, fonctions de mesure).

Poinçonnage avec temporisation

PDELAYON temporise l'application du coup de poinçon. Cette instruction modale comporte une fonctionnalité préparatoire et se place de ce fait, en règle générale, avant PON. Après PDELAYOF, le poinçonnage se poursuit de manière normale.

Remarque

La définition du temps de retard s'effectue dans la donnée de réglage SD42400 $SC_PUNCH_DWELLTIME.

Accélération asservie à la course

PUNCHACC permet de paramétrer une caractéristique d'accélération, définissant différentes accélérations selon la distance entre les trous.

Deuxième interface de poinçonnage

Pour les machines utilisant en alternance une deuxième interface de poinçonnage (deuxième unité de poinçonnage ou support similaire), il est possible de commuter sur une deuxième paire d'entrées et de sorties TOR rapides de la commande (paire d'E/S). La fonctionnalité de poinçonnage/grignotage complète est applicable avec chacune des deux interfaces de

poinçonnage. La commutation entre la première et la deuxième interface de poinçonnage s'effectue avec les instructions `SPIF1` et `SPIF2`.

Remarque

Condition : une deuxième paire d'E/S doit être définie dans un paramètre machine pour la fonctionnalité de poinçonnage (→ voir les indications du constructeur de machine !).

Syntaxe

```
PON G... X... Y... Z...
SON G... X... Y... Z...
SONS G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYON
PDELAYOF
PUNCHACC(<Smini>,<Amini>,<Smaxi>,<Amaxi>)
SPIF1/SPIF2
SPOF
```

Signification

`PON` :	Activer le poinçonnage	
`SON` :	Activer le grignotage	
`PONS` :	Activer le poinçonnage à la période d'appel de l'interpolateur	
`SONS` :	Activer le grignotage à la période d'appel de l'interpolateur	
`SPOF` :	Désactiver le poinçonnage/grignotage	
`PDELAYON` :	Activer le poinçonnage avec temporisation	
`PDELAYOF` :	Désactiver le poinçonnage avec temporisation	
`PUNCHACC` :	Activation de l'accélération en fonction de la distance Paramètres :	
	`<Smini>`	Distance minimale entre les trous
	`<Amini>`	Accélération initiale `<Amini>` peut être supérieur à `<Amaxi>`.
	`<Smaxi>`	Distance maximale entre les trous
	`<Amaxi>`	Accélération finale `<Amax>` peut être supérieur à `<Amin>`.
`SPIF1` :	Activer la **première** interface de poinçonnage Le pilotage du poinçon s'effectue avec la première paire d'E/S rapides.	
`SPIF2` :	Activer la **deuxième** interface de poinçonnage Le pilotage du poinçon s'effectue avec la deuxième paire d'E/S rapides.	
	Remarque : après un RESET ou un démarrage de la commande, la première interface de poinçonnage est toujours active. Si une seule interface de poinçonnage est utilisée, il n'est pas nécessaire de la programmer.	

Exemples

Exemple 1 : activer le grignotage

Code de programme	Commentaire
...	
N70 X50 SPOF	; Positionnement sans déclenchement de poinçonnage.
N80 X100 SON	; Activer le grignotage, déclenchement d'un coup avant le déplacement (X=50) et à la fin du déplacement programmé (X=100).
...	

Exemple 2 : poinçonnage avec temporisation

Code de programme	Commentaire
...	
N170 PDELAYON X100 SPOF	; Positionnement sans déclenchement de poinçonnage, activation du déclenchement de poinçonnage temporisé.
N180 X800 PON	; Activer le poinçonnage. Lorsque la position finale est atteinte, un coup de poinçon est déclenché avec une temporisation.
N190 PDELAYOF X700	; Désactiver le poinçonnage avec une temporisation, déclenchement normal du poinçonnage à la fin du déplacement programmé.
...	

Exemple 3 : poinçonnage avec deux interfaces de poinçonnage

Code de programme	Commentaire
...	
N170 SPIF1 X100 PON	; A la fin du bloc, le déclenchement d'un coup est réalisé sur la première sortie rapide. Une surveillance du signal "Coup actif" a lieu sur la première entrée.
N180 X800 SPIF2	; Le deuxième déclenchement d'un coup est réalisé sur la deuxième sortie rapide. Une surveillance du signal "Coup actif" a lieu sur la deuxième entrée.
N190 SPIF1 X700	; La commande du coup s'effectue avec la première interface pour tous les autres coups.
...	

Informations complémentaires

Poinçonnage et grignotage à la période d'appel de l'interpolateur (PONS/SONS)

Le poinçonnage et le grignotage à la période d'appel de l'interpolateur ne sont pas possibles simultanément dans plusieurs canaux. L'activation de PONS ou SONS n'est possible que dans un seul canal à la fois.

Accélération en fonction de la course (PUNCHACC)

Exemple :

```
PUNCHACC(2,50,10,100)
```

Distances entre les trous inférieures à 2mm :

Le déplacement a lieu avec une accélération égale à 50% de l'accélération maximale.

Distances entre les trous comprises entre 2mm et 10mm :

L'accélération augmente proportionnellement à la distance, jusqu'à 100%.

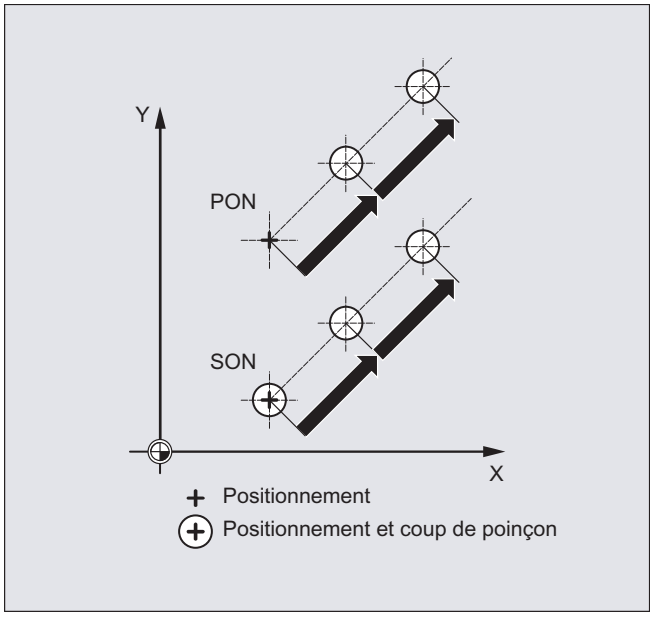
Distances entre les trous supérieures à 10mm :

Déplacement avec une accélération égale à 100%.

Déclenchement du premier coup de poinçon

Selon qu'il s'agit du poinçonnage ou du grignotage, le déclenchement du premier coup de poinçon, après activation de la fonction, intervient à des instants différents :

- `PON/PONS` :
  - Tous les coups de poinçon - y compris celui du premier bloc après l'activation - sont exécutés en fin de bloc.
- `SON/SONS` :
  - Le premier coup de poinçon après l'activation du grignotage, est toujours exécuté en début de bloc.
  - Les autres coups de poinçon sont toujours libérés en fin de bloc.



Poinçonnage et grignotage sur place

Un coup de poinçon ne peut être déclenché que si le bloc contient une instruction de déplacement pour les axes de poinçonnage ou de grignotage (axes du plan actif).

Afin de pouvoir néanmoins déclencher un coup de poinçon, tout en restant au même endroit, l'un des axes de poinçonnage/grignotage est programmé avec un déplacement égal à 0.

Utilisation d'outils indexables

Remarque

Pour positionner des outils indexables de façon tangentielle à la trajectoire programmée, utilisez la fonction Positionnement tangentiel.

Utilisation d'instructions M

Grâce à la macroprogrammation, il est toujours possible d'utiliser des fonctions M spéciales au lieu des instructions en langage de programmation (compatibilité). Les correspondances avec les systèmes plus anciens sont les suivantes :

```
M20, M23   ≙  SPOF
M22        ≙  SON
M25        ≙  PON
M26        ≙  PDELAYON
```

Exemple de fichier macro :

Code de programme	**Commentaire**
`DEFINE M25 AS PON`	`; Activation du poinçonnage`
`DEFINE M125 AS PONS`	`; Activation du poinçonnage à la période d'appel de l'interpolateur`
`DEFINE M22 AS SON`	`; Activation du grignotage`
`DEFINE M122 AS SONS`	`; Activation du grignotage à la période d'appel de l'interpolateur`
`DEFINE M26 AS PDELAYON`	`; Activation du poinçonnage avec temporisation`
`DEFINE M20 AS SPOF`	`; Désactivation du poinçonnage, du grignotage`
`DEFINE M23 AS SPOF`	`; Désactivation du poinçonnage, du grignotage`

Exemple de programmation :

Code de programme	**Commentaire**
`...`	
`N100 X100 M20`	`; Positionnement sans déclenchement de poinçonnage.`
`N110 X120 M22`	`; Activer le grignotage, déclenchement d'un coup avant et après le déplacement.`
`N120 X150 Y150 M25`	`; Activer le poinçonnage, déclenchement d'un coup à la fin du déplacement.`
`...`	

16.2 Segmentation automatique des déplacements

Subdivision en distances partielles

Quand le poinçonnage ou le grignotage est activé, SPP et SPN provoquent la segmentation en un nombre donné de distances partielles de même longueur (segmentation équidistante des déplacements) de l'ensemble de la distance de déplacement programmée pour les axes à interpolation. En interne, à chaque distance partielle correspond un bloc.

Nombre de coups de poinçon

En poinçonnage, le premier coup de poinçon intervient au point final de la première distance partielle ; par contre, en grignotage, il intervient au point de départ de la première distance partielle. Sur la distance globale parcourue, les valeurs sont les suivantes :

Poinçonnage : nombre de coups = nombre de distances partielles

Grignotage : nombre de coups = nombre de distances partielles + 1

Fonctions auxiliaires

Les fonctions auxiliaires sont exécutées dans le premier des blocs générés.

Syntaxe

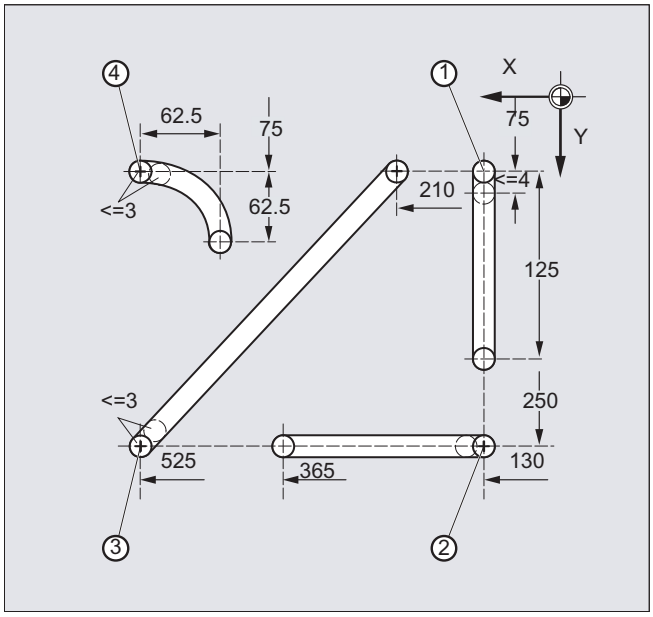
```
SPP=
SPN=
```

Signification

`SPP` :	Taille de la distance partielle (distance maximale entre coups de poinçon consécutifs) ; fonction à effet modal
`SPN` :	Nombre de distances partielles par bloc ; fonction à effet modal

Exemple 1

Les parcours de grignotage doivent être décomposés automatiquement en distances partielles identiques.

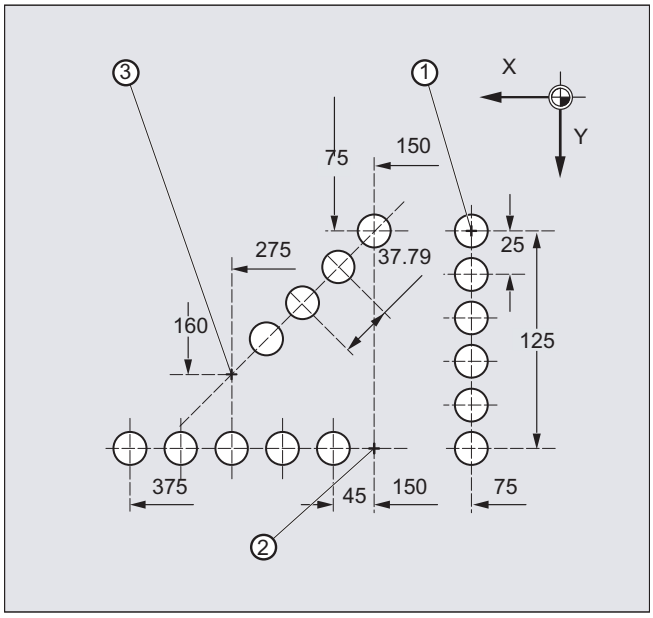


Code de programme	**Commentaire**
N100 G90 X130 Y75 F60 SPOF	; Positionnement sur le point de départ 1
N110 G91 Y125 SPP=4 SON	; Activation du grignotage ; longueur maximale de distance partielle pour segmentation automatique des déplacements : 4 mm
N120 G90 Y250 SPOF	; Désactiver le grignotage ; positionnement sur le point de départ 2
N130 X365 SON	; Activation du grignotage ; longueur maximale de distance partielle pour segmentation automatique des déplacements : 4 mm
N140 X525 SPOF	; Désactiver le grignotage ; positionnement sur le point de départ 3
N150 X210 Y75 SPP=3 SON	; Activation du grignotage ; longueur maximale de distance partielle pour segmentation automatique des déplacements : 3 mm
N160 X525 SPOF	; Désactiver le grignotage ; positionnement sur le point de départ 4

Code de programme	Commentaire
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; Activation du grignotage ; longueur maximale de distance partielle pour segmentation automatique des déplacements : 3 mm
N180 G00 G90 Y300 SPOF	; Désactiver le grignotage

Exemple 2

On souhaite une segmentation automatique des déplacements pour chaque rangée de trous. Pour la segmentation, la longueur maximale de distance partielle (valeur de SPP) sera programmée.



Code de programme	Commentaire
N100 G90 X75 Y75 F60 PON	; Positionnement sur le point de départ 1 ; activation du poinçonnage, poinçonner un trou unique
N110 G91 Y125 SPP=25	; Longueur maximale de distance partielle pour segmentation automatique des déplacements : 25 mm
N120 G90 X150 SPOF	; Désactiver le poinçonnage ; positionnement sur le point de départ 2
N130 X375 SPP=45 PON	; Activation du poinçonnage ; longueur maximale de distance partielle pour segmentation automatique des déplacements : 45 mm
N140 X275 Y160 SPOF	; Désactiver le poinçonnage ; positionnement sur le point de départ 3
N150 X150 Y75 SPP=40 PON	; Activer le poinçonnage ; au lieu de la longueur de distance partielle programmée de 40 mm, la longueur de distance partielle calculée de 37,79 mm est utilisée.
N160 G00 Y300 SPOF	; Désactiver le poinçonnage ; positionnement

16.2.1 Segmentation des déplacements pour axes à interpolation

Longueur de la distance partielle SPP

Avec `SPP`, vous définissez la distance maximale entre coups de poinçon consécutifs et également la longueur maximale des distances partielles, sur lesquelles la distance totale de déplacement doit être répartie. La désactivation de l'instruction se fait avec `SPOF` ou `SPP=0`.

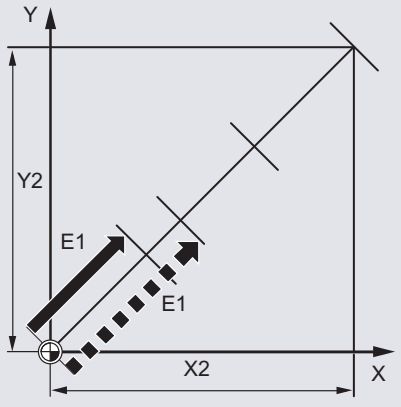
Exemple :

```
N10 SON X0 Y0
N20 SPP=2 X10
```

La distance totale de déplacement de 10 mm est divisée en 5 distances partielles de 2 mm chacune (`SPP=2`).

Remarque

La segmentation des déplacements avec `SPP` donne toujours lieu à des déplacements équidistants : toutes les distances partielles ont la même longueur. Autrement dit, la grandeur de la distance partielle programmée (valeur de `SPP`) n'est valable que si, et seulement si, le quotient de la distance totale de déplacement par la valeur de SPP est un entier. Si ce n'est pas le cas, la taille de la distance partielle est réduite, de façon interne, de manière à que le quotient soit un entier.



X2/Y2 Distance à parcourir programmée (bloc de grignotage / poinçonnage)
E1 Distance partielle programmée
E1' Distance partielle automatiquement arrondie

Exemple :

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

Avec une distance totale de déplacement de 15 mm et une distance partielle de 3,5 mm, on obtient un quotient qui n'est pas un nombre entier (4,28). Une réduction de la valeur de SPP a donc lieu jusqu'à l'obtention d'un quotient entier. Il en résulte une taille de distance partielle de 3 mm.

Nombre de distances partielles SPN

Avec `SPN` , vous précisez le nombre de distances partielles à générer sur l'ensemble de la distance de déplacement. La longueur des distances partielles est calculée automatiquement. `SPN` ayant un effet non modal, il convient d'activer auparavant le poinçonnage ou le grignotage avec `PON` ou `SON`.

16.2.2 Segmentation du déplacement dans le cas d'axes individuels

Si, en plus des axes d'interpolation, des axes individuels sont aussi configurés comme axes de poinçonnage/grignotage, ceux-ci sont aussi soumis à la segmentation automatique des déplacements.

Comportement d'un axe individuel avec SPP

La longueur programmée de la distance partielle (`SPP`) se rapporte en premier lieu aux axes d'interpolation. C'est la raison pour laquelle la valeur SPP est ignorée dans un bloc où sont programmés le déplacement d'axe individuel et la valeur SPP, mais aucun axe d'interpolation.

Si, dans un bloc, des axes individuels aussi bien que des axes d'interpolation sont programmés, le comportement de l'axe individuel dépend du réglage du paramètre machine concerné.

1. Réglage par défaut
   Le déplacement de l'axe individuel est réparti de façon égale sur les blocs intermédiaires générés par `SPP`.

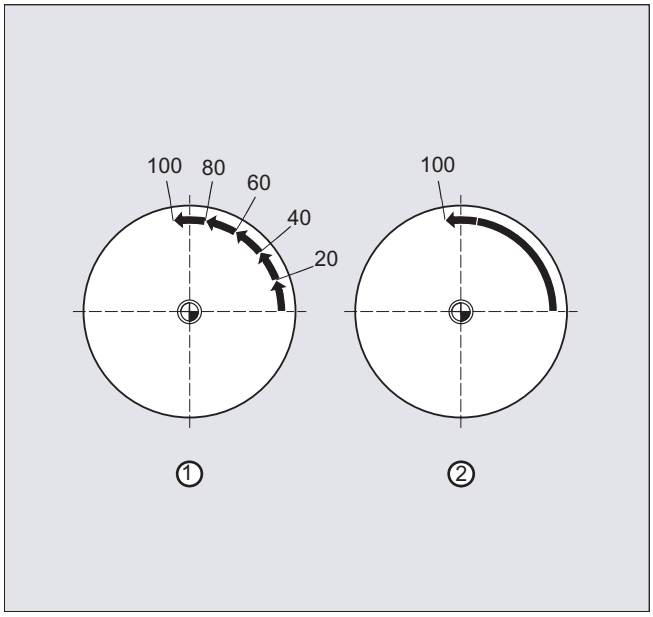
Exemple :

```
N10 G1 SON X10 A0

N20 SPP=3 X25 A100
```

En fonction de la distance entre coups de poinçon de 3 mm, le déplacement global de l'axe X (axe d'interpolation) de 15 mm donne lieu à 5 blocs.

L'axe A pivote de ce fait de 20° par bloc.



1. Axe individuel sans segmentation des déplacements
   L'axe individuel parcourt son déplacement global dans le premier des blocs générés.
2. Segmentation variable des déplacements
   Le comportement de l'axe individuel dépend de l'interpolation des axes d'interpolation :

- Interpolation circulaire : Segmentation des déplacements
- Interpolation linéaire : pas de segmentation des déplacements

Comportement avec SPN

Le nombre de distances partielles programmé reste valable, même si un axe d'interpolation n'a pas été programmé en même temps.

Condition : L'axe individuel est défini comme axe de poinçonnage/grignotage.

Rectification 17

17.1 Activer/désactiver la surveillance d'outil spécifique à la rectification (TMON, TMOF)

Les procédures TMON(...) et TMOF(...) prédéfinies permettent d'activer ou de désactiver la surveillance d'outil spécifique à la rectification (surveillance de la géométrie et de la vitesse).

Condition

Les paramètres $TC_TPG1 à $TC_TPG9 spécifiques à un outil doivent être définis.

Syntaxe

```
TMON(<TNr>)
...
TMOF(<TNr>)
```

Signification

`TMON(...)` :	Activer la surveillance d'outil spécifique à la rectification L'instruction doit être programmée dans le canal dans lequel la surveillance d'outil spécifique à la rectification doit être activée.
`TMOF(...)` :	Désactiver la surveillance d'outil spécifique à la rectification L'instruction doit être programmée dans le canal dans lequel la surveillance d'outil spécifique à la rectification doit être désactivée.
`<TNr>` :	Numéro T **Remarque :** Requis uniquement lorsque la surveillance doit être activée ou désactivée pour une meule non active à la place de l'outil actif mis en œuvre.
`TMOF(0)` :	désactiver la surveillance pour tous les outils

Autres fonctions 18

18.1 Fonctions d'axe (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

"AXNAME" est utilisé par exemple pour la création de cycles à caractère général lorsque les noms des axes ne sont pas connus.

"AX" est utilisé pour la programmation indirecte d'axes géométriques et d'axes synchrones. Le descripteur d'axe est alors enregistré dans une variable de type AXIS ou est fourni par une instruction telle que "AXNAME" ou "SPI".

"SPI" est utilisé lorsque des fonctions d'axe sont programmées pour une broche (par exemple une broche synchrone).

"AXTOSPI" est utilisé pour convertir un descripteur d'axe en indice de broche (fonction inverse de "SPI").

"AXSTRING" est utilisé pour convertir un descripteur d'axe (type de données AXIS) en chaîne de caractères (fonction inverse de "AXNAME").

"ISAXIS" est utilisé dans des cycles à caractère général, pour vérifier l'existence d'un axe géométrique bien défini et pour éviter qu'un appel consécutif de $P_AXNX soit interrompu du fait d'une erreur.

"MODAXVAL" est utilisé pour déterminer la position modulo d'axes rotatifs modulo.

Syntaxe

```
AXNAME("chaîne de caractères")
AX[AXNAME("chaîne de caractères")]
SPI(n)

AXTOSPI(A) ou AXTOSPI(B) ou AXTOSPI(C)
AXSTRING( SPI(n) )
ISAXIS(<numéro d'axe géométrique>)
<Position modulo>=MODAXVAL(<axe>,<position d'axe>)
```

Signification

AXNAME :	Convertit la chaîne de caractères d'entrée en un descripteur d'axe ; la chaîne de caractères d'entrée doit comporter un nom d'axe valide.
AX :	Descripteur d'axe variable
SPI :	Conversion du numéro de broche indiqué en descripteur d'axe ; le paramètre de transfert doit être un numéro de broche valide.
n :	numéro de broche
AXTOSPI :	Convertit un descripteur d'axe en un indice de broche de type Integer. "AXTOSPI" correspond à la fonction d'inversion pour "SPI".
X, Y, Z :	Descripteur d'axe de type AXIS comme variable ou comme constante
AXSTRING :	La chaîne de caractères avec numéro de broche attribué.

`ISAXIS:`	vérifie si l'axe géométrique indiqué existe
`MODAXVAL:`	Détermination de la position modulo des axes rotatifs modulo, qui correspond au reste modulo par rapport à la zone modulo paramétrée (0 à 360 degrés en réglage par défaut ; les paramètres machine PM30340 MODULO_RANGE_START et PM30330 $MA_MODULO_RANGE permettent de modifier le début et la taille de la zone modulo).

Remarque

Extensions SPI

La fonction d'axe SPI(n) est également utilisable pour la lecture et l'écriture de composantes de frame. Il est ainsi possible, par exemple, d'écrire des frames avec la syntaxe `$P_PFRAME[SPI(1),TR]=2.22`.

La programmation supplémentaire de positions d'axe avec l'adresse `AX[SPI(1)]=<position d'axe>` permet de déplacer un axe, à condition que la broche soit en mode de positionnement ou en mode axe.

Exemples

Exemple 1 : AXNAME, AX, ISAXIS

Code de programme	**Commentaire**
`OVRA[AXNAME("axe transversal")]=10`	`; Correction d'axe transversal`
`AX[AXNAME("axe transversal")]=50.2`	`; Position de fin de l'axe transversal`
`OVRA[S1]=70`	`; Correction de la broche 1`
`AX[SPI(1)]=180`	`; Position de fin de la broche 1`
`IF ISAXIS(1)== FALSE GOTOF SUITE`	`; Abscisse existante ?`
`AX[$P_AXN1]=100`	`; Déplacement de l'abscisse`
`SUITE :`	

Exemple 2 : AXSTRING

En cas de programmation avec AXSTRING[SPI(n)], ce n'est plus l'indice de l'axe auquel la broche est affectée qui est sorti en tant que numéro de broche, mais la chaîne de caractères "Sn".

Code de programme	**Commentaire**
`AXSTRING[SPI(2)]`	`; Sortie de la chaîne de caractères "S2".`

Exemple 3 : MODAXVAL

La position modulo de l'axe rotatif modulo A doit être déterminée.

Le calcul part de la position d'axe 372.55.

La zone modulo paramétrée s'étend de 0 à 360 degrés :

PM30340 MODULO_RANGE_START = 0

PM30330 $MA_MODULO_RANGE = 360

Code de programme	Commentaire
R10=MODAXVAL(A,372.55)	; Position modulo calculée R10 = 12.55.

Exemple 4 : MODAXVAL

Si le descripteur d'axe programmé ne se rapporte pas à un axe rotatif modulo, la valeur à convertir (<position d'axe>) est retournée telle quelle.

Code de programme	Commentaire
R11=MODAXVAL(X,372.55)	; X est l'axe linéaire ; R11 = 372.55.

18.2 Axes géométriques permutables (GEOAX)

La fonction "Axes géométriques permutables" permet de remplacer les axes géométriques configurés dans les paramètres machine par d'autres axes de canal.

Syntaxe

```
GEOAX(<n>,<axe de canal>,<n>,<axe de canal>,<n>,<axe de canal>)
GEOAX()
```

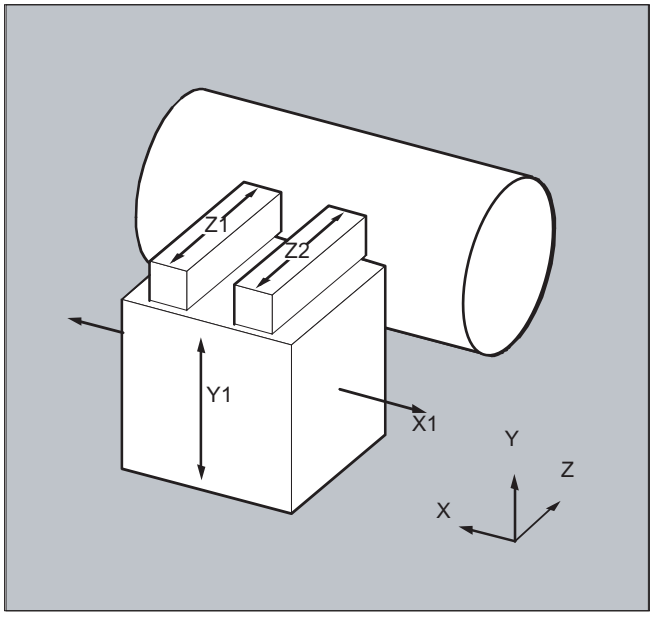
Signification

`GEOAX(...)`	Fonction de permutation des axes géométriques. **Remarque :** `GEOAX()` sans indication de paramètre réactive la configuration de base des axes géométriques définie dans les paramètres machine.
`<n>`	Numéro de l'axe géométrique qui sera remplacé par l'axe de canal indiqué. Plage de valeurs : 0, 1, 2, 3 **Remarque :** 0: l'axe de canal indiqué est supprimé du groupe d'axes géométriques sans remplacement 1: 1. axe géométrique ≙ axe de coordonnées X (abscisse) du SCP 2: 2. axe géométrique ≙ axe de coordonnées Y (ordonnée) du SCP 3: 3. axe géométrique ≙ axe de coordonnées Z (cote) du SCP
`<Axe de canal>`	Nom de l'axe de canal qui doit être intégré dans le groupe d'axes géométriques.

Exemples

Exemple 1 : deux axes permutés de façon alternée comme axe géométrique

Un chariot porte-outil peut être déplacé par le biais des axes de canal X1, Y1, Z1, Z2 :



Les axes géométriques sont configurés de sorte que d'abord, après la mise sous tension, Z1 soit actif en tant que 3e axe géométrique avec le nom d'axe géométrique "Z" et qu'il forme le groupe d'axes géométriques avec X1 et Y1.

Dans le programme pièce, les axes Z1 et Z2 doivent ensuite intervenir de façon alternée comme axe géométrique Z :

Code de programme	Commentaire
...	
N100 GEOAX (3,Z2)	; L'axe de canal Z2 intervient en tant que 3e axe géométrique (Z).
N110 G1 ...	
N120 GEOAX(3,Z1)	; L'axe de canal Z1 intervient en tant que 3e axe géométrique (Z).
...	

Exemple 2 : permutation des axes géométriques en présence de 6 axes de canal

Une machine comporte 6 axes de canal désignés par XX, YY, ZZ, U ,V ,W.

Le préréglage de la configuration des axes géométriques effectué dans les paramètres machine est le suivant :

axe de canal XX = 1er axe géométrique (axe X)

axe de canal YY = 2e axe géométrique (axe Y)

axe de canal ZZ = 3e axe géométrique (axe Z)

Code de programme	Commentaire
N10 GEOAX()	; La configuration de base des axes géométriques est activée.
N20 G0 X0 Y0 Z0 U0 V0 W0	; Tous les axes accostent la position 0 en vitesse rapide.

Code de programme	Commentaire
N30 GEOAX(1,U,2,V,3,W)	; L'axe de canal U devient le premier (X), V devient le deuxième (Y)
	; et W devient le troisième axe géométrique (Z).
N40 GEOAX(1,XX,3,ZZ)	; L'axe de canal XX devient le premier (X), ZZ devient le troisième
	; axe géométrique (Z). L'axe de canal V reste le deuxième
	: axe géométrique (Y).
N50 G17 G2 X20 I10 F1000	; Cercle complet dans le plan X/Y. Déplacement des
	; axes de canal XX et V.
N60 GEOAX(2,W)	; L'axe de canal W devient le deuxième axe géométrique (Y).
N80 G17 G2 X20 I10 F1000	; Cercle complet dans le plan X/Y. Déplacement des
	; axes de canal XX et W.
N90 GEOAX()	; Réinitialisation à l'état de base.
N100 GEOAX(1,U,2,V,3,W)	; L'axe de canal U devient le premier (X), V devient le deuxième
	; (Y) et W devient le troisième axe géométrique (Z).
N110 G1 X10 Y10 Z10 XX=25	; Les axes de canal U, V, W se déplacent respectivement à la
	; position 10. L'axe XX, en tant qu'axe supplémentaire, se déplace à la position 25.
N120 GEOAX(0,V)	; V est supprimé du groupe d'axes géométriques.
	; U et W restent premier (X) et troisième
	; axe géométrique (Z).
	; Le deuxième axe géométrique (Y) reste non affecté.
N130 GEOAX(1,U,2,V,3,W)	; L'axe de canal U reste premier (X), V devient
	; le deuxième (Y) et W reste le troisième axe géométrique (Z).
N140 GEOAX(3,V)	; V devient le troisième axe géométrique (Z), écrasant W
	; qui est ainsi supprimé du groupe d'axes
	; géométriques. Le deuxième axe géométrique (Y)
	; n'est toujours pas affecté.

Paramètres machine

Configuration d'axes

Affectation des axes géométriques, des axes supplémentaires et des axes machine aux axes de canal :

- PM10000 $MN_AXCONF_MACHAX_NAME_TAB
- PM20050 $MC_AXCONF_GEOAX_ASIGN_TAB
- PM20060 $MC_AXCONF_GEOAX_NAME_TAB
- PM20070 $MC_AXCONF_MACHAX_USED

- PM20080 $MC_AXCONF_CHANAX_NAME_TAB
- PM35000 $MA_SPIND_ASSIGN_TO_MACHAX

Comportement au reset

Comportement au reset des affectations d'axes géométriques modifiées :

- PM20110 $MC_RESET_MODE_MASK, bit 12
- PM20118 $MC_GEOAX_CHANGE_RESET

Comportement de départ CN

- PM20112 $MC_START_MODE_MASK, bit 12

Notification au programme utilisateur AP

Option de paramétrage de l'instruction M qui sera transmise à l'interface CN/AP en présence d'une permutation d'axes géométriques :

- PM22532 $MC_GEOAX_CHANGE_M_CODE

Autres conditions à prendre en compte

Aucune permutation d'axes géométriques

- Quand l'une des fonctions suivantes est active, aucune permutation d'axes géométriques n'est possible :
  - Transformation
  - Interpolation de type spline
  - Correction du rayon d'outil
  - Correction fine d'outil
- L'axe géométrique et un autre axe de canal portent le même nom.
- L'un des axes concernés par la permutation d'axes géométriques est impliqué dans une action qui dure au-delà des limites de bloc. Par exemple : axe de positionnement pour l'ensemble des blocs ou axe asservi d'un couplage d'axes.

Axes rotatifs

Les axes rotatifs ne peuvent pas être définis comme axes géométriques.

Etat d'un axe après le remplacement

Un axe remplacé par commutation dans le groupe d'axes géométriques est programmable en tant qu'axe supplémentaire au moyen de son nom d'axe de canal après la commutation.

Frames, zones de protection, limitations de la zone de travail

La permutation des axes géométriques supprime tous les frames, les zones de protection et les limitations de la zone de travail.

Coordonnées polaires

Une permutation des axes géométriques avec `GEOAX` met les coordonnées polaires modales à la valeur 0 de manière analogue à un changement de plan avec `G17-G19`.

DRF, DO

Un éventuel décalage par manivelle (DRF) ou un décalage d'origine externe (DO) reste actif après la permutation.

Configuration de base des axes géométriques

L'instruction `GEOAX()` appelle la configuration de base du groupe d'axes géométriques.

Après un POWER ON et en cas de commutation en mode de fonctionnement "Prise de référence", on retourne automatiquement à la configuration de base.

Correction de longueur d'outil

Une correction de longueur d'outil active reste active après la permutation. Pour les axes géométriques qui viennent d'être intégrés ou dont la position a été permutée, on considère cependant que le déplacement correspondant n'a pas encore été exécuté. Pour la première instruction de déplacement de ces axes géométriques, le déplacement résultant est donc égal à la somme de la correction de longueur d'outil et du déplacement programmé.

Les axes géométriques qui gardent leur position dans le groupe d'axes après la permutation conservent également leur état relatif à la correction de longueur d'outil.

Configuration d'axe géométrique en cas de transformation active

- La configuration d'axe géométrique définie dans les paramètres machine pour une transformation active ne peut pas être modifiée avec la fonction "Axes géométriques permutables".
- En présence d'une configuration d'axe géométrique différente pour une transformation, différents blocs de données doivent être définis dans les paramètres machine de transformation.
- L'activation d'une transformation supprime une configuration d'axe géométrique modifiée avec `GEOAX`.
- En ce qui concerne les axes géométriques, les paramétrages d'axes géométriques spécifiques aux transformations pour les transformations actives ont la priorité sur les paramétrages concernant la permutation d'axes géométriques.
  Exemple : une transformation est active. D'après les paramètres machine, la transformation doit être conservée après un reset de canal. Toutefois, après un reset de canal, la configuration de base des axes géométriques doit également être rétablie. La configuration d'axe géométrique qui a été définie pour la transformation est conservée.
- Le réglage de base de la configuration d'axe géométrique reprend effet lors de la désactivation d'une transformation.

Mode JOG, fonction machine REF

La configuration d'axe géométrique définie dans les paramètres machine prend effet lors du passage en mode JOG, fonction machine REF (accostage du point de référence).

18.3 Conteneur d'axes (AXCTSWE, AXCTSWED, AXCTSWEC)

Les instructions "AXCTSWE" ou "AXCTSWED" permettent de débloquer la rotation des conteneurs d'axes spécifiés.

L'instruction "AXCTSWEC" permet d'annuler un déblocage de rotation du conteneur d'axes ayant déjà été spécifié.

Syntaxe

```
AXCTSWE(<ID>)
AXCTSWED(<ID>)
AXCTSWEC(<ID>)
```

Signification

AXCTSWE :	Déblocage de la rotation du conteneur d'axes L'exécution du programme ne s'arrête pas avec "AXCTSWE". La rotation se produit dès que le déblocage a été ordonné à tous les canaux impliqués dans le conteneur d'axes.	
AXCTSWED :	Déblocage de la rotation du conteneur d'axes sans prise en compte des autres canaux impliqués. **Remarque** • Variante d'instruction destinée à la simplification de la mise en service du programme pièce ou de l'action synchrone. • Le comportement par rapport aux autres canaux impliqués dans le conteneur d'axes peut être spécifié à l'aide du paramètre machine suivant : PM12760 $MN_ AXCT_FUNCTION_MASK, bit 0	
AXCTSWEC :	Annulation du déblocage de la rotation du conteneur d'axes **Remarque** Le déblocage de la rotation du conteneur d'axes ne peut être annulé que si la rotation n'a pas encore commencé : $AN_AXCTSWA[<conteneur d'axes>] == 0 Au sujet de la variable système, voir "Conteneur d'axes (AXCTSWE, AXCTSWED, AXCTSWEC) (Page 679)"	
<ID> :	Descripteur du conteneur d'axes ou d'un axe de conteneur :	
	CT<numéro> :	Descripteur par défaut d'un axe de conteneur : PM12750 $MN_AXCT_NAME_TAB Exemple : "CT1"
	<conteneur> :	Descripteur spécifique à l'utilisateur d'un conteneur d'axes : PM12750 $MN_AXCT_NAME_TAB Exemple : "CONTENEUR_1
	<axe> :	Descripteur d'un axe de conteneur connu dans le canal

Remarque

Pas

Le pas de rotation d'un conteneur d'axes est réglé par la donnée de réglage suivante :

SD41700 $SN_AXCT_SWWIDTH

Informations complémentaires

Diagnostic

La lecture de l'état actuel d'un conteneur d'axe peut s'effectuer avec la variable système suivante :

Variable système	Type	Description
$AC_AXCTSWA[<nom>]	BOOL	Etat spécifique au canal du conteneur d'axes
$AN_AXCTSWA[<conteneur d'axes>]	BOOL	Etat spécifique à la NCU du conteneur d'axes
$AN_AXCTSWE[<conteneur d'axes>]	INT	Etat de la rotation du conteneur d'axes spécifique à l'emplacement La variable système indique **bit par bit** l'état des emplacements du conteneur d'axes. Chaque bit correspond à un emplacement.
$AN_AXCTAS[<conteneur d'axes>]	INT	Nombre d'emplacements (slots) autour desquels le conteneur d'axes a été permuté.

Rotation du conteneur d'axes avec GET/GETD implicite

Le paramètre machine ci-après permet de définir que, avec l'instruction AXCTSWE, tous les axes de conteneur du canal peuvent être récupérées dans le canal au moyen d'un "GET/GETD" implicite. Une permutation d'axes ne redevient possible qu'après une rotation de conteneur réussie.

PM10722 $MN_AXCHANGE_MASK, bit 1 = 1

Remarque

La rotation d'un conteneur d'axes avec "GET / GETD" implicite n'est **pas** exécutée pour un axe à l'état "axe d'exécution de bloc" (par ex. axe AP), car l'axe devrait quitter cet état pour permettre la rotation.

18.4 Attente de la position d'axe définitive (WAITENC)

L'instruction de langage "WAITENC" permet d'attendre dans le programme CN que les positions d'axes synchronisées ou restaurées pour les axes configurés avec PM34800 $MA_WAIT_ENC_VALID = 1 soient disponibles.

Une interruption peut avoir lieu en mode d'attente, par ex. par le démarrage d'un ASUP ou par le changement de mode de fonctionnement en JOG. Le mode d'attente est éventuellement repris avec la poursuite du programme.

Remarque

Le mode d'attente est affiché dans l'interface utilisateur au moyen de l'état d'arrêt "Attendre système de mesure".

Syntaxe

"WAITENC" peut être programmé dans la section d'un programme CN quelconque.

La programmation doit être réalisée dans un bloc spécifique :

```
...
WAITENC
...
```

Exemple

"WAITENC" est par ex. utilisé dans le programme d'application commandé par événement .../_N_CMA_DIR/_N_PROG_EVENT_SPF ainsi que le montre l'exemple d'application suivant.

Exemple d'application : retrait d'outil après POWER OFF avec transformation d'orientation

Un usinage avec orientation de l'outil a été interrompu par une coupure de courant.
Le programme d'application commandé par événement .../_N_CMA_DIR/_N_PROG_EVENT_SPF est appelé lors du démarrage s'en suivant.

Dans le programme d'application commandé par événement, "WAITENC" permet d'attendre des positions d'axe synchronisées ou restaurées afin de pouvoir calculer par la suite une grille pour orienter le SCP dans la direction de l'outil.

Code de programme	Commentaire
...	
IF $P_PROG_EVENT == 4	; Démarrage.
IF $P_TRAFO <> 0	; Transformation a été activée.
WAITENC	; Attente des positions valables des axes d'orientation.
TOROTZ	; Rotation de l'axe Z du SCP en direction de l'axe d'outil.
ENDIF	
M17	
ENDIF	

Code de programme	Commentaire
...	

En mode JOG, l'outil peut par la suite, par un déplacement de retrait, être dégagé dans la direction de l'axe d'outil.

18.5 Commutation de jeu de paramètres programmable (SCPARA)

L'instruction "SCPARA" permet de demander la commutation à un certain jeu de paramètres pour un axe.

Remarque

Pas de commutation de jeu de paramètres pendant l'usinage du filetage

Lors d'une opération de filetage G33 et de taraudage G331 / G332, le jeu de paramètres est sélectionné par la commande et ne peut être modifié.

Commutation de jeu de paramètres bloquée

Une commutation de jeu de paramètres peut également être demandée via l'interface CN/AP. Pour éviter tout conflit, il est possible de bloquer la commutation de jeu de paramètres via l'interface CN/AP (`SCPARA`) :

DB31, ... DBX9.3 (spécification de jeu de paramètres bloquée par CN)

Remarque

Si une commutation de jeu de paramètres est demandée par "SCPARA" alors que la commutation de jeu de paramètres est bloquée par l'interface CN/AP, la commutation sera refusée sans génération de message d'erreur.

Syntaxe

```
SCPARA[<axe>]=<valeur>
```

Signification

`SCPARA` :	Instruction : commuter le jeu de paramètres	
`<axe>` :	Descripteur d'axe (axe de canal)	
	Type :	AXIS
`<valeur>` :	Numéro de jeu de paramètres : 1, 2, 3, ... numéro maximum de jeu de paramètres	

Exemple

Code de programme	**Commentaire**
...	
N110 SCPARA[X] = 3	; Sélection : axe X, jeu de paramètres 3
...	

Informations complémentaires

Déblocage de la commutation de jeu de paramètres

La commutation de jeu de paramètres doit être débloquée de manière explicite :

PM35590 $MA_PARAMSET_CHANGE_ENABLE[<axe>]

Lecture du numéro de jeu de paramètres

La variable système $AA_SCPAR permet de lire le numéro du jeu de paramètres sélectionné (jeu de paramètres prescrit).

Bibliographie

Pour obtenir des informations détaillées sur les jeux de paramètres, voir :

Description fonctionnelle Fonctions de base ; chapitre "Vitesses, système de consignes/ mesures, régulation (G2)" > "Régulation" > "Jeux de paramètres du régulateur de position"

18.6 Contrôler le langage CN existant (STRINGIS)

La fonction "STRINGIS(...)" permet de contrôler si la chaîne de caractères indiquée est disponible comme élément de langage de programmation CN dans l'étendue actuelle des langages.

Définition

```
INT STRINGIS(STRING <nom>)
```

Syntaxe

```
STRINGIS(<nom>)
```

Signification

`STRINGIS` :	Fonction avec valeur de retour
`<nom>` :	Nom de l'élément à contrôler du langage de programmation CN
Valeur de retour :	Le format de la valeur de retour est yxx (décimal).

Eléments du langage de programmation CN

Les éléments suivants du langage de programmation CN peuvent être contrôlés :

- instructions G de tous les groupes de fonctions G existants, par exemple "G0", "INVCW", "POLY", "ROT", "KONT", "SOFT", "CUT2D", "CDON", "RMBBL", "SPATH"
- Adresses DIN ou CN, comme "ADIS", "RNDM", "SPN", "SR", "MEAS"
- Fonctions, par exemple "TANG(...)" ou "GETMDACT"
- Procédures par ex. "SBLOF".
- Mots-clés, par exemple "ACN", "DEFINE" ou "SETMS"
- Données système par ex. paramètres machine $M... , données de réglage $S... ou données optionnelles $O...
- Variables système $A... , $V... , $P...
- Paramètres de calcul R...
- Noms de cycles activés
- Variables GUD et LUD
- Noms de macros
- Noms d'étiquettes

Valeur de retour

La valeur de retour est uniquement significative dans les 3 premiers chiffres en notation décimale. Le format de la valeur de retour est yxx, y étant alors = information de base et xx = information détaillée.

Valeur de retour	Signification	
000	Le 'nom' de la chaîne de caractères n'est pas connu dans le présent système [1)]	
100	Le 'nom' de la chaîne de caractères est un élément du langage de programmation CN, il n'est toutefois actuellement pas programmable (option/fonction est inactive)	
2xx	Le 'nom' de la chaîne de caractères est un élément programmable du langage de programmation CN (option/fonction est active). L'information détaillée xx contient d'autres informations sur le type de l'élément :	
	xx	**Signification**
	01	Adresse DIN ou CN [2)]
	02	Instruction G (par ex. G04, INVCW)
	03	Fonction avec valeur de retour
	04	Fonction sans valeur de retour
	05	Mot-clé (par ex. DEFINE)
	06	Paramètre machine ($M...) , donnée de réglage ($S...) ou donnée optionnelle ($O...)
	07	Paramètre système, par ex. variable système ($...) ou paramètre de calcul (R...)
	08	Cycle (le cycle doit être chargé dans la CN et les programmes de cycle doivent être actifs [3)])
	09	Variable GUD (la variable GUD doit correspondre à celle définie dans les fichiers de définition GUD et les variables GUD doivent être activées)
	10	Nom de macro (le macro doit correspondre à celui défini dans les fichiers de définition Macro et les macros doivent être activés) [4)]
	11	Variable LUD du programme pièce actuel
	12	Instruction G ISO (mode de langage ISO doit être actif)
400	Le 'nom' de la chaîne de caractères est une adresse CN n'ayant pas été reconnue comme xx == 01 ou xx == 10 et n'étant pas G ni R [2)]	
y00	Pas d'affectation spécifique possible	

1) En fonction de la commande, seul un sous-ensemble des instructions de langage CN Siemens est éventuellement connu, par ex. SINUMERIK 802D sl. Sur ces commandes, la valeur 0 est retournée pour des chaînes de caractères n'étant pas principalement des instructions de langage CN Siemens. Ce comportement peut être modifié au moyen de PM10711 $MN_NC_LANGUAGE_CONFIGURATION. Dans le cas de PM10711 = 1, la valeur 100 est toujours retournée pour des instructions de langage CN Siemens.

2) Les lettre suivantes sont des adresses CN : A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Ces adresses CN peuvent également être programmées avec une extension d'adresse. L'extension d'adresse peut être indiquée lors du contrôle avec STRINGIS.
Exemple : 201 == STRINGIS("A1").
Les lettres : D, F, H, L, M, N, O, P, S, T sont des adresses CN ou des fonctions auxiliaires utilisées de manière définie par l'utilisateur. La valeur 400 est toujours retournée pour ces adresses. Exemple : 400 == STRINGIS("D"). Ces adresses CN ne peuvent pas être indiquées avec extension d'adresse lors du contrôle avec STRINGIS.
Exemple : 000 == STRINGIS("M02"), mais 400 == STRINGIS("M").

3) Les noms de paramètres de cycles ne peuvent pas être contrôlés avec STRINGIS.
4) Des adresses définies comme macro, par ex. G, H, M, L, sont identifiées comme macro

Exemples

Dans les exemples ci-dessous, il est supposé que les éléments de langage CN indiqués comme chaîne de caractères, sauf indication contraire, sont principalement programmables dans la commande.

1. La chaîne de caractères "T" est définie comme fonction auxiliaire :
   ```
   400 == STRINGIS("T")
   000 == STRINGIS ("T3")
   ```
2. La chaîne de caractères "X" est définie comme axe :
   ```
   201 == STRINGIS("X")
   201 == STRINGIS("X1")
   ```
3. La chaîne de caractères "A2" est définie comme adresse CN avec extension :
   ```
   201 == STRINGIS("A")
   201 == STRINGIS("A2")
   ```
4. La chaîne de caractères "INVCW" est définie comme instruction G nommée :
   ```
   202 == STRINGIS("INVCW")
   ```
5. La chaîne de caractères "$MC_GCODE_RESET_VALUES" est définie comme paramètre machine :
   ```
   206 == STRINGIS("$MC_GCODE_RESET_VALUES")
   ```
6. La chaîne de caractères "GETMDACT" est une fonction de langage CN :
   ```
   203 == STRINGIS("GETMDACT ")
   ```
7. La chaîne de caractères "DEFINE" est un mot-clé :
   ```
   205 == STRINGIS("DEFINE")
   ```
8. La chaîne de caractères "$TC_DP3" est un paramètre système (composantes de longueur d'outil) :
   ```
   207 == STRINGIS("$TC_DP3")
   ```
9. La chaîne de caractères "$TC_TP4" est un paramètre système (taille d'outil) :
   ```
   207 == STRINGIS("$TC_TP4")
   ```
10. La chaîne de caractères "$TC_MPP4" est un paramètre système (état de l'emplacement d'outil) :
    - La gestion de magasin d'outils est active : `207 == STRINGIS("$TC_MPP4") ;`
    - La gestion de magasin d'outils n'est pas active : `000 == STRINGIS("$TC_MPP4")`

    Voir également ci-dessous le paragraphe : Gestion de magasin d'outils.
11. La chaîne de caractères "MACHINERY_NAME" est définie comme variable GUD :
    ```
    209 == STRINGIS("MACHINERY_NAME")
    ```
12. La chaîne de caractères "LONGMACRO" est définie comme macro :
    ```
    210 == STRINGIS("LONGMACRO")
    ```
13. La chaîne de caractères "MYVAR" est définie comme variable LUD :
    ```
    211 == STRINGIS("MYVAR")
    ```
14. La chaîne de caractères "XYZ" n'est pas une instruction, variable GUD, nom de macro ni nom de cycle connu dans la CN :
    ```
    000 == STRINGIS("XYZ")
    ```

Gestion de magasin d'outils

Si la fonction de gestion d'outils n'est pas active, STRINGIS fournit pour les paramètres système de la gestion de magasin d'outils, indépendamment du paramètre machine

- PM10711 $MN_NC_LANGUAGE_CONFIGURATION

toujours la valeur 000.

Mode ISO

Si la fonction "Mode ISO" est active :

- PM18800 $MN_MM_EXTERN_LANGUAGE (activation de langages CN externes)
- PM10880 $MN_ MM_EXTERN_CNC_SYSTEM (système de commande devant être adapté)

TRINGIS contrôle la chaîne de caractères indiquée tout d'abord en tant qu'instruction G SINUMERIK. Si la chaîne de caractères n'est pas une instruction G SINUMERIK, elle sera par la suite contrôlée en tant qu'instruction G ISO.

Des commutations programmées (`G290` (mode SINUMERIK), `G291` (mode ISO)) n'ont aucun effet sur STRINGIS.

Exemple

Les importants paramètres machine pour la fonction STRINGIS(...) ont les valeurs suivantes :

- PM10711 $MN_NC_LANGUAGE_CONFIGURATION = 2 (seules les instructions de langage CN dont les options sont configurées sont considérées comme étant reconnues)
- PM19410 $ON_TRAFO_TYPE_MASK = 'H0' (option : transformations)
- PM10700 $MN_PREPROCESSING_LEVEL='H43' (conditionnement pour cycles actif)

Le programme exemplaire ci-dessous est exécuté sans message d'erreur :

Code de programme	**Commentaire**
`N1 R1=STRINGIS("TRACYL")`	`; R1 == 0, étant donné que TRACYL est identifiée comme ''étant non connue" en raison de l'absence de l'option de transformation`
`N2 IF STRINGIS("TRACYL") == 204`	
`N3  TRACYL(1,2,3)`	`; N3 est ignoré`
`N4 ELSE`	
`N5  G00`	`; et N5 est exécuté à la place`
`N6 ENDIF`	
`N7 M30`	

18.7 Appel interactif des fenêtres depuis le programme pièce (MMC)

Le sous-programme MMC(...) prédéfini permet d'afficher à partir d'un programme CN des boîtes de dialogue spécifiques à l'utilisateur sur l'interface utilisateur de SINUMERIK Operate par exemple.

La configuration des boîtes de dialogue est possible pour les types de boîte de dialogue suivants :

- Run MyScreens
- Easy XML
- XML utilisateur

Bibliographie

- Manuel de programmation Run MyScreens
- Manuel de programmation Easy XML

Syntaxe

```
MMC("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>","<QUIT>")
```

Signification

<table>
<tr><td>MMC(...) :</td><td colspan="2">Descripteur de sous-programme
Les paramètres sont indiqués avec un code d'emplacement et séparés par des virgules au sein de deux chaînes de caractères, la chaîne de commande et la chaîne d'acquittement.</td></tr>
<tr><td colspan="3">Paramètres au sein de la chaîne de commande :</td></tr>
<tr><td rowspan="6"><ADDRESS> :</td><td colspan="2">Groupe fonctionnel dans lequel sont exécutées les boîtes de dialogue utilisateur qui ont été configurées</td></tr>
<tr><td>Fonction</td><td>Groupes fonctionnels</td></tr>
<tr><td>Boîte de dialogue utilisateur "Run MyScreens"</td><td>CYCLES</td></tr>
<tr><td>Boîte de dialogue utilisateur "Easy XML"</td><td>CYCLES</td></tr>
<tr><td>XML utilisateur</td><td>XML</td></tr>
<tr><td>Fenêtre contextuelle "Run MyScreens"</td><td>POPUPDLG</td></tr>
<tr><td>Fenêtre contextuelle "Easy XML"</td><td>POPUPDLG</td></tr>
<tr><td rowspan="6"><COMMAND> :</td><td colspan="2">Commande à exécuter</td></tr>
<tr><td>Fonction</td><td>Commandes</td></tr>
<tr><td>Boîte de dialogue utilisateur "Run MyScreens"</td><td>PICTURE_ON, PICTURE_OFF</td></tr>
<tr><td>Boîte de dialogue utilisateur "Easy XML"</td><td>PICTURE_ON, PICTURE_OFF</td></tr>
<tr><td>XML utilisateur</td><td>XML_ON, XML_OFF</td></tr>
<tr><td>Fenêtre contextuelle "Run MyScreens"</td><td>PICTURE_ON, PICTURE_OFF</td></tr>
<tr><td>Fenêtre contextuelle "Easy XML"</td><td>PICTURE_ON, PICTURE_OFF</td></tr>
</table>

<FILE> :	Nom du fichier dans lequel la boîte de dialogue à afficher est programmée	
	Fonction	**Fichiers**
	Boîte de dialogue utilisateur "Run MyScreens"	<nom>.com
	Boîte de dialogue utilisateur "Easy XML"	<nom>.xml
	XML utilisateur	<nom>.xml
	Fenêtre contextuelle "Run MyScreens"	<nom>.com
	Fenêtre contextuelle "Easy XML"	<nom>.xml
	Fenêtre contextuelle "Easy XML" avec configuration directe dans le programme CN (voir exemple 2)	xmldial_emb.xml
<DIALOG> :	Nom de la boîte de dialogue à afficher	
	Fonction	**Nom de la boîte de dialogue**
	Toutes les fonctions à l'exception de Fenêtre contextuelle "Easy XML" avec configuration directe dans le programme CN	Nom configuré de la boîte de dialogue dans le fichier <FILE>
	Fenêtre contextuelle "Easy XML" avec configuration directe dans le programme CN (voir exemple 3)	main
Paramètres au sein de la chaîne d'acquittement :		
<QUIT> :	Type d'acquittement	
	N :	Pas d'acquittement. L'exécution du programme se poursuit après l'envoi de l'ordre. Il n'y a aucune signalisation en retour lorsque la commande n'a pas pu être exécutée avec succès. **Remarque** Le type d'acquittement "N" doit être utilisé lorsqu'une durée d'affichage (arrêt temporisé) est programmée dans le programme CN (voir Exemple 2 ci-dessous)
	A:	Acquittement asynchrone L'exécution du programme se poursuit après l'envoi de l'ordre. La valeur de retour est enregistrée dans une variable d'acquittement (variable GUD) spécifique à l'utilisateur définie dans le cadre de la configuration de la boîte de dialogue et peut être lue dans le programme CN.

Exemple

Exemple 1

Affichage d'une boîte de dialogue et réaction à l'action de l'utilisateur dans un programme CN.

```
; La variable d'acquittement QUIT a déjà été créée en tant que variable utilisateur globale (GUD)
; du type STRING lors de la configuration de la boîte de dialogue :
; DEF NCK STRING[20] QUIT
QUIT = "XXX"                                ; Initialiser la variable d'acquittement
G4 F5
```

Code de programme	**Commentaire**
MMC("CYCLES,PICTURE_ON,test.com,test1","A")	; Afficher la boîte de dialogue ; - Groupe fonctionnel : CYCLES ; - État de l'image : PICTURE_ON (afficher) ; - Fichier de boîte de dialogue : test.com ; - Boîte de dialogue : test1
INPUT :	; Attente de l'action de l'utilisateur
STOPRE	; Arrêt du prétraitement des blocs
IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK	; Touche logicielle "RUN"
IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK	; Touche logicielle "CHK"
GOTOB INPUT	; => Attente
WORK :	; Touche logicielle "RUN" actionnée
MSG("Poursuivre l'usinage -> Départ CN")	; Générer signalisation
MMC("CYCLES,PICTURE_OFF","N")	; Fermer la boîte de dialogue
M0	; Attente du départ CN
GOTOF END	; => Fin du programme
CHECK :	; Touche logicielle "CHK" actionnée
MSG("Accoster la position -> Départ CN")	; Générer signalisation
MMC("CYCLES,PICTURE_OFF","N")	; Fermer la boîte de dialogue
M0	; Attente du départ CN
GOTOF END	; => Fin du programme
END :	
...	

Exemple 2

La durée d'affichage d'une boîte de dialogue est définie dans le programme CN, par exemple par le biais d'un arrêt temporisé.

Code de programme	**Commentaire**
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_**ON**,xmldial_emb.xml,main","N")	; Afficher la boîte de dialogue
X200	
Z40	
MMC("POPUPDLG,PICTURE_**OFF**","N")	; Fermer la boîte de dialogue

Exemple 3

Intégration d'un script de fenêtre contextuelle dans un programme CN et de son utilisation.

Code de programme

```
PROC POPUP_TEST
```

Code de programme

```
; ----------------------------- Script -----------------------------
; <main_dialog entry="rpara_main">
;    <let name="xpos" />
;    <let name="ypos" />
;    <let name="field_name" type="string" />
;    <let name="num" />
;    <menu name="rpara_main">
;       <open_form name="rpara_form"/>
;       <softkey_back>
;          <close_form />
;       </softkey_back>
;    </menu>
;    <form name="rpara_form">
;       <init>
;          <caption>mask from NC part program</caption>
;          <let name="count" >0</let>
;          <op>
;             xpos = 120;
;             ypos = 34;
;             "nck/Channel/Parameter/R[10]" = 10;
;          </op>
;          <!-- load the number of controls -->
;          <op>
;             num = "nck/Channel/Parameter/R[10]";
;          </op>
;          <while>
;             <condition> count < num</condition>
;             <print name="field_name" text="edit%d">count</print>
;             <op>
;                ypos = ypos + 24;
;                count = count + 1;
;             </op>
;          </while>
;       </init>
;       <paint>
;          <op>
;             xpos =  8;
;             ypos = 36;
;             count = 0;
;          </op>
;          <while>
;             <condition>count < num</condition>
;             <print name="field_name" text="R-Parameter%d">count</print>
;             <text xpos = "$xpos" ypos = "$ypos" >$$$field_name</text>
```

Code de programme

```
;              <op>
;                 ypos = ypos + 24;
;                 count = count + 1;
;              </op>
;           </while>
;       </paint>
;    </form>
; </main_dialog>
; ========================= Partie du programme ==========================
...
G94 F100
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")
G4 F4
X200
MMC("POPUPDLG,PICTURE_OFF","N")
G4 F2
X0
...
```

Autres conditions

- Les fichiers de définition *.com des boîtes de dialogue doivent être enregistrés dans le dossier "proj".
- Les fichiers de définition Easy XML *.xml des boîtes de dialogue doivent être enregistrés dans le dossier "appl".
  Si les fichiers de définition sont enregistrés dans un autre répertoire, le chemin d'accès doit être indiqué de façon indirecte en partant du répertoire "appl".
- Les boîtes de dialogue définies par l'utilisateur ne peuvent pas être affichées simultanément à partir de différents canaux.

18.8 Temps d'exécution du programme / Compteur de pièces

Des informations relatives au temps d'exécution du programme et au nombre de pièces sont disponibles pour aider l'opérateur de la machine-outil.

Ces informations peuvent être traitées comme variables système dans le programme CN et/ou AP. En même temps, elles sont disponibles pour l'affichage sur l'interface utilisateur.

18.8.1 Temps d'exécution de programme

Pour la surveillance de processus technologiques, la fonction "Temps d'exécution" met à disposition des compteurs chronométriques internes de la CN, qui peuvent être lus dans le programme pièce et dans les actions synchrones au moyen de variables système spécifiques à la CN et au canal.

Le déclencheur de la mesure de temps d'exécution ($AC_PROG_NET_TIME_TRIGGER) est la seule variable système de la fonction, qui est accessible en écriture. Il sert à opérer une mesure sélective de sections de programme. En d'autres termes, l'écriture du déclencheur dans le programme CN permet d'activer et de désactiver la mesure de temps.

Variable système	Signification	Activité
Spécifiques à la CN		
$AN_SETUP_TIME	Temps (en minutes) depuis le dernier démarrage de la commande avec les valeurs par défaut ("démarrage à froid") Remise à "0" automatique à chaque démarrage de la commande avec les valeurs par défaut.	• Activation permanente
$AN_POWERON_TIME	Temps en minutes depuis le dernier démarrage normal de la commande ("démarrage à chaud") Remise à "0" automatique à chaque démarrage normal de la commande.	
Spécifiques au canal		

Variable système	Signification	Activité
$AC_OPERATING_TIME	Temps global en secondes pour l'exécution de programmes CN en mode automatique Remise à "0" automatique de la valeur à chaque démarrage de la commande.	• Activation par le paramètre machine MD27860 • Uniquement en mode AUTO
$AC_CYCLE_TIME	Temps d'exécution du programme CN sélectionné en secondes Remise à "0" automatique de la valeur au démarrage d'un nouveau programme CN.	
$AC_CUTTING_TIME	Temps d'usinage en secondes Le temps d'exécution des axes à interpolation (au moins l'un d'entre eux est actif) est totalisé sans l'activation du rapide dans tous les programmes CN, entre le départ CN et la fin du programme / RESET de la CN. La mesure est suspendue pendant tout arrêt temporisé. Remise à "0" automatique de la valeur à chaque démarrage de la commande avec les valeurs par défaut.	
$AC_ACT_PROG_NET_TIME	Temps d'exécution actuel net en secondes du programme CN courant Remise à "0" automatique au démarrage d'un programme CN.	• Activation permanente • Uniquement en mode AUTO
$AC_OLD_PROG_NET_TIME	Temps d'exécution net du programme terminé correctement à l'instant avec M30, en secondes.	
$AC_OLD_PROG_NET_TIME_COUNT	Modifications sur $AC_OLD_PROG_NET_TIME Après un POWER ON, $AC_OLD_PROG_NET_TIME_COUNT est à "0". $AC_OLD_PROG_NET_TIME_COUNT est toujours incrémenté lorsque la commande a récrit $AC_OLD_PROG_NET_TIME.	

Variable système	Signification		Activité
$AC_PROG_NET_TIME_TRIGGER	Déclencheur de la mesure du temps d'exécution :		• Uniquement en mode AUTO
	0	Etat neutre Le déclencheur n'est pas actif.	
	1	Arrêter Arrête la mesure et copie la valeur de $AC_ACT_PROG_NET_TIME dans $AC_OLD_PROG_NET_TIME. $AC_ACT_PROG_NET_TIME est mis à "0" puis continue.	
	2	Départ Lance la mesure et met à "0" $AC_ACT_PROG_NET_TIME. $AC_OLD_PROG_NET_TIME n'est pas modifié.	
	3	Arrêt Arrêt de la mesure. $AC_OLD_PROG_NET_TIME n'est pas modifié et $AC_ACT_PROG_NET_TIME reste constant jusqu'à la reprise.	
	4	Reprise Reprise de la mesure, c'est-à-dire qu'une mesure arrêtée précédemment reprend. $AC_ACT_PROG_NET_TIME se poursuit. $AC_OLD_PROG_NET_TIME n'est pas modifié.	
Un POWER ON remet toutes les variables système à "0".			

Remarque

Constructeur de machine

Les temporisations activables sont activées au moyen du paramètre machine MD27860 $MC_PROCESSTIMER_MODE.

Le comportement des mesures de temps actives pour certaines fonctions (par ex. GOTOS, correction = 0%, test d'avance actif, test de programme, ASUP, PROG_EVENT, …) est configuré au moyen des paramètres machine MD27850 $MC_PROG_NET_TIMER_MODE et MD27860 $MC_PROCESSTIMER_MODE.

Bibliographie :
Description fonctionnelle Fonctions de base ; GMF, Canal, Mode de programme, Comportement au reset (K1), chapitre : Temps d'exécution de programme

Remarque

Temps restant pour une pièce

Lorsqu'il s'agit de produire des pièces identiques dans la foulée, les valeurs de temporisation suivantes permettent de déterminer le temps restant requis pour une pièce :

- temps d'usinage de la dernière pièce produite (voir $AC_OLD_PROG_NET_TIME)
- temps d'usinage actuel (voir $AC_ACT_PROG_NET_TIME)

Le temps restant s'affiche sur l'interface utilisateur, en plus du temps d'usinage actuel.

Remarque

Utilisation de STOPRE

Les variables système $AC_OLD_PROG_NET_TIME et $AC_OLD_PROG_NET_TIME_COUNT n'entraînent pas d'arrêt implicite du prétraitement. En cas d'utilisation dans le programme pièce, ceci n'est pas critique lorsque la valeur des variables système provient de l'exécution précédente du programme. Par contre, si le déclencheur de la mesure du temps d'exécution ($AC_PROG_NET_TIME_TRIGGER) est écrit avec une fréquence élevée, modifiant ainsi fréquemment $AC_OLD_PROG_NET_TIME, il est recommandé d'utiliser un `STOPRE` explicite dans le programme pièce.

Autres conditions à prendre en compte

- **Recherche de bloc**
  Aucun temps d'exécution de programme n'est déterminé pendant une recherche de bloc.
- **REPOS**
  La durée d'un processus REPOS est ajoutée au temps d'usinage actuel ($AC_ACT_PROG_NET_TIME).

Exemples

Exemple 1 : Mesure du temps de "mySubProgrammA"

Code de programme

```
...
N50 DO $AC_PROG_NET_TIME_TRIGGER=2
N60 FOR ii= 0 TO 300
N70 mySubProgrammA
N80 DO $AC_PROG_NET_TIME_TRIGGER=1
N95 ENDFOR
N97 mySubProgrammB
N98 M30
```

Après le traitement de la ligne N80 par le programme, le temps d'exécution net de "mySubProgrammA" figure dans $AC_OLD_PROG_NET_TIME.

La valeur de $AC_OLD_PROG_NET_TIME :

- est conservée au-delà de M30,
- est actualisée après chaque exécution de boucle.

Exemple 2 : Mesure du temps de "mySubProgrammA" et de "mySubProgrammC"

Code de programme

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
```

```
Code de programme
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
```

18.8.2 Compteurs de pièces

La fonction "Compteurs de pièces" met à disposition différents compteurs, qu'il est en particulier possible d'utiliser en interne à la commande pour le comptage de pièces.

Les compteurs sont disponibles sous forme de variables système spécifiques à un canal avec accès en écriture et lecture dans la plage de valeurs de 0 à 999 999 999.

Variable système	Signification
$AC_REQUIRED_PARTS	Nombre de pièces à fabriquer (nombre de pièces requis) Ce compteur permet de définir le nombre de pièces qui, lorsqu'il est atteint, entraîne la remise à "0" du nombre de pièces réel ($AC_ACTUAL_PARTS).
$AC_TOTAL_PARTS	Nombre total de pièces fabriquées (nombre total réel de pièces) Ce compteur indique le nombre total de pièces fabriquées à partir du début du comptage. Remise à "0" automatique de la valeur uniquement en cas de démarrage de la commande avec les valeurs par défaut.
$AC_ACTUAL_PARTS	Nombre de pièces fabriquées (nombre réel de pièces) Ce compteur enregistre le nombre total de pièces fabriquées depuis le début du comptage. Le compteur est automatiquement remis à "0" (à condition que $AC_REQUIRED_PARTS > 0), lorsque le nombre de pièces requis ($AC_REQUIRED_PARTS) est atteint.
$AC_SPECIAL_PARTS	Nombre de pièces comptées par l'utilisateur Ce compteur permet à l'utilisateur de définir son propre mode de comptage des pièces. Il est possible de définir l'émission d'une alarme lorsque le nombre de pièce requis est atteint ($AC_REQUIRED_PARTS). L'utilisateur doit assurer lui-même la remise à zéro du compteur.

Remarque

Chaque compteur de pièces est préréglé à sa valeur par défaut "0" au démarrage de la commande et son contenu peut être lu/écrit indépendamment de son mode d'activation.

Remarque

Des paramètres machine spécifiques au canal permettent de régler le mode d'activation des compteurs, l'instant de leur remise à zéro et l'algorithme de comptage.

Remarque

Comptage de pièces avec fonction M définie par l'utilisateur

Des paramètres machine permettent de définir le déclenchement des impulsions de comptage pour les différents compteurs de pièces au moyen de fonctions M définies par l'utilisateur à la place du déclenchement à la fin du programme `M2`/`M30`.

18.9 Process DataShare - Sortie sur un appareil/fichier externe (EXTOPEN, WRITE, EXTCLOSE)

L'écriture de données depuis un programme pièce sur un périphérique/fichier externe s'effectue en trois étapes :

1. Ouverture du périphérique/fichier externe
   L'instruction EXTOPEN permet d'ouvrir le périphérique/fichier externe pour le canal en vue de l'écriture.
2. Écriture de données
   La donnée en sortie peut être préparée avec les fonctions String du langage CN, par exemple SPRINT. L'écriture elle-même s'effectue à l'aide de l'instruction WRITE.
3. Fermeture du périphérique/fichier externe
   L'instruction EXTCLOSE, le fait d'atteindre la fin du programme (M30) ou la réinitialisation du canal permet de libérer de nouveau le périphérique/fichier externe occupé dans le canal.

Syntaxe

```
DEF INT <Result>
DEF STRING[<n>] <Output>
…
EXTOPEN(<Result>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
…
<Output>="Sortie données"
WRITE(<Result>,<ExtDev>,<Output>)
…
EXTCLOSE(<Result>,<ExtDev>)
```

Signification

`EXTOPEN` :	Procédure prédéfinie pour l'ouverture d'un périphérique/fichier externe		
`<Result>` :	**Paramètre 1 :** variable de résultat La réussite de l'opération peut être contrôlée dans le programme au moyen de la valeur de la variable de résultat, afin de poursuivre le traitement en conséquence.		
	Type :	INT	
	Valeurs :	0	Aucune erreur
		1	Impossible d'ouvrir le périphérique externe
		2	Le périphérique externe n'est pas configuré
		3	Périphérique externe configuré avec un chemin d'accès non valide
		4	Pas de droits d'accès pour le périphérique externe
		5	Mode d'utilisation : périphérique externe déjà affecté de façon "exclusive"
		6	Mode d'utilisation : périphérique externe déjà affecté en mode "shared"
		7	Longueur de fichier supérieure à LOCAL_DRIVE_MAX_FILESIZE
		8	Nombre maximum de périphériques externes dépassé
		9	Option pour LOCAL_DRIVE non activée
		11	L'interface V.24 est déjà occupée par la fonction Easy-Message (828D uniquement)
		12	Mode d'écriture : Indication en contradiction avec extdev.ini
		16	Chemin externe programmé non valide
		22	Périphérique externe non monté

<table>
<tr><td rowspan="11"><ExtDev> :</td><td colspan="2">Paramètre 2 : descripteur symbolique pour le périphérique/fichier externe à ouvrir</td></tr>
<tr><td>Type :</td><td>STRING</td></tr>
<tr><td colspan="2">Le descripteur symbolique se compose des éléments suivants :
1. nom logique du périphérique
2. éventuellement suivi d'un chemin d'accès au fichier (suffixé avec "/").
Les noms logiques de périphériques suivants sont définis :</td></tr>
<tr><td>"LOCAL_DRIVE" :</td><td>Carte CompactFlash locale (prédéfinie)</td></tr>
<tr><td>"CYC_DRIVE" :</td><td>Indication de lecteur réservée pour l'utilisation dans les cycles SIEMENS (prédéfini)</td></tr>
<tr><td>"/dev/ext/1", ...
"/dev/ext/9" :</td><td>Lecteurs réseau disponibles
Remarque :
Configuration requise dans le fichier extdev.ini !</td></tr>
<tr><td>"/dev/cyc/1",
"/dev/cyc/2" :</td><td>Indications de lecteurs réservées pour l'utilisation dans les cycles SIEMENS
Remarque :
Configuration requise dans le fichier extdev.ini !</td></tr>
<tr><td>"/dev/v24" :</td><td>Interface V.24
Remarque :
Configuration requise dans le fichier extdev.ini !</td></tr>
<tr><td colspan="2">Chemin d'accès :
• Pour "LOCAL_DRIVE" et "CYC_DRIVE", un chemin d'accès doit être indiqué, par exemple :
"LOCAL_DRIVE/my_dir/my_file.txt"
• Les noms logiques de périphériques "/dev/ext/1...9" et "/dev/cyc/1...2" peuvent être configurés pour :
– faire référence à un fichier, auquel cas il suffit d'indiquer le nom logique du périphérique, par exemple :
"/dev/ext/4"
– ou faire référence à un répertoire, auquel cas un chemin d'accès est nécessaire, par exemple :
"/dev/ext/5/my_dir/my_file.txt"
• Pour "/dev/v24", aucun chemin d'accès ne doit être suffixé.</td></tr>
<tr><td colspan="2">Remarque :
Pour les noms logiques de périphériques "/dev/ext/1...9", "/dev/v24" et "/dev/cyc/1...2", aucune distinction n'est faite entre minuscules et majuscules, mais pour l'indication d'un chemin d'accès à un fichier, la distinction est significative. Pour "LOCAL_DRIVE" et "CYC_DRIVE", seules les majuscules sont autorisées.</td></tr>
</table>

<SyncMode> :	**Paramètre 3 :** Mode de traitement des instructions WRITE à destination de ce périphérique/fichier		
	Type :	STRING	
	Valeurs :	"SYN" :	Écriture synchrone L'exécution du programme est interrompue jusqu'à ce que le processus d'écriture soit terminé. Le succès de l'écriture synchrone peut être contrôlé par l'évaluation des variables d'erreur de l'instruction WRITE.
		"ASYN" :	Écriture asynchrone L'exécution du programme n'est pas interrompue par l'instruction WRITE. **Remarque :** La variable de résultat de l'instruction WRITE n'est pas décisive dans ce mode et a toujours la valeur 0 (aucune erreur). Dans ce mode, il n'y a pas de certitude que l'instruction WRITE ait réussi.
<AccessMode> :	**Paramètre 4 :** Mode d'utilisation de ce périphérique/fichier		
	Type :	STRING	
	Valeurs :	"SHARED" :	Le périphérique/fichier est requis dans le mode "Shared". D'autres canaux peuvent également utiliser le périphérique, c'est à dire s'ouvrir également dans ce mode.
		"EXCL" :	Le périphérique/fichier est utilisé exclusivement dans le canal. Aucun autre canal ne peut également utiliser le périphérique.
<WriteMode> :	**Paramètre 5 :** Mode d'écriture pour les instructions WRITE à destination de ce périphérique/fichier (facultatif)		
	Type :	STRING	
	Valeurs :	"APP" :	Ajout Le fichier reste intact quant à son contenu. Les requêtes d'écriture s'ajoutent à la fin.
		"OVR" :	Écraser Le contenu du fichier est effacé et recréé par les appels d'écriture suivants.
	Remarque : Ce paramètre ne permet pas d'écraser le mode d'écriture configuré dans le fichier extdev.ini. En cas de conflit, l'appel EXTOPEN est acquitté avec génération d'une erreur.		

WRITE :	Procédure prédéfinie pour l'écriture des données de sortie

EXTCLOSE :	Procédure prédéfinie pour la fermeture d'un périphérique/fichier externe ouvert		
<Result> :	**Paramètre 1 :** variable de résultat		
	Type :	INT	
	Va-leurs :	0	Aucune erreur
		16	Chemin externe programmé non valide
		21	Erreur lors de la fermeture d'un périphérique externe
<ExtDev> :	**Paramètre 2 :** descripteur symbolique pour le périphérique/fichier externe à fermer : description, voir sous EXTOPEN. **Remarque :** Le descripteur doit être identique à celui indiqué dans l'appel EXTOPEN.		

Exemple

Code de programme

```
N10       DEF INT RESULT
N20       DEF BOOL EXTDEVICE
N30       DEF STRING[80] OUTPUT
N40       DEF INT PHASE
N50       EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60       IF RESULT > 0
N70          MSG("Erreur avec EXTOPEN :" << RESULT)
N80       ELSE
N90          EXTDEVICE=TRUE
N100      ENDIF
…
N200      PHASE=4
N210      IF EXTDEVICE
N220         OUTPUT=SPRINT("Fin de la phase : %D",PHASE)
N230         WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240      ENDIF
…
```

Voir aussi

Opérations sur les chaînes de caractères (Page 91)

Ecriture d'un fichier (WRITE) (Page 154)

18.10 Alarmes (SETAL)

Les alarmes peuvent être définies dans un programme CN. Elles sont affichées dans une fenêtre particulière de l'interface utilisateur. A une alarme correspond obligatoirement une réaction de la commande qui dépend de la catégorie à laquelle appartient l'alarme.

Bibliographie :
Pour de plus amples informations sur les réactions d'alarme, voir Manuel de mise en service.

Syntaxe

```
SETAL(<numéro d'alarme>[,<chaîne de caractères>])
```

Signification

`SETAL` :	Mot clé pour la programmation d'une alarme. `SETAL` doit être programmé dans un bloc CN spécifique.	
`<numéro d'alarme>` :	Variable de type INT contenant le numéro d'alarme. La plage admise pour les numéros d'alarmes est comprise entre 60000 et 69999, les numéros 60000 à 64999 étant réservés pour les cycles SIEMENS et les numéros 65000 à 69999 à la disposition de l'utilisateur.	
`<chaîne de caractères>` :	Lors de la programmation d'alarmes de cyclique utilisateur, il est possible d'ajouter une chaîne de caractères comportant jusqu'à 4 paramètres. Dans ces paramètres, il est possible de définir des textes utilisateur variables. Les paramètres prédéfinis suivants sont également disponibles :	
	Paramètre	**Signification**
	%1	numéro de canal
	%2	numéro de bloc, étiquette
	%3	Indice de texte pour alarmes de cycles
	%4	Paramètre d'alarme supplémentaire

Remarque

Les textes d'alarme doivent être configurés dans l'interface utilisateur.

Remarque

Si les alarmes générées doivent s'afficher dans la langue activée pour l'interface utilisateur, l'utilisateur doit disposer d'informations sur la langue actuellement réglée sur l'IHM. Cette information peut être interrogée dans le programme pièce et dans les actions synchrones, via la variable système $AN_LANGUAGE_ON_HMI (voir "Langue actuelle dans l'IHM (Page 968)").

Exemple

Code de programme	**Commentaire**
...	
N100 SETAL (65000)	; Activer l'alarme n° 65000

Code de programme	Commentaire
...	

18.11 Arrêt étendu et retrait (ESR)

La fonction "Arrêt étendu et retrait", désignée par le sigle ESR dans la suite de ce document, permet de réagir avec souplesse aux situations d'erreur en fonction du process :

- **Arrêt étendu**
  Dans la mesure où le cas d'erreur spécifique le permet, tous les axes débloqués pour l'arrêt étendu sont immobilisés de manière ordonnée.
- **Retrait**
  L'outil se trouvant en cours d'intervention est retiré de la pièce le plus rapidement possible.
- **Fonctionnement en génératrice (fonction SINAMICS "Régulation Vdc")**
  En cas de dépassement par la bas d'une valeur paramétrée de la tension de circuit intermédiaire, par ex. en cas de panne de réseau, l'énergie électrique nécessaire au retrait est générée par réinjection de l'énergie de freinage d'un entraînement prévu à cet effet (fonctionnement en génératrice).

Causes de déclenchement

Causes générales (externes à la CN / globales ou spécifiques à un GMFC/canal)

- Entrées TOR (par ex. d'un module NCU) ou mémoire image lisible, interne à la commande, des sorties TOR ($A_IN, $A_OUT)
- Etat du canal ($AC_STAT)
- Signaux VDI ($A_DBB)
- Signalisations groupées d'un certain nombre d'alarmes ($AC_ALARM_STAT)

Causes axiales

- Seuil de retrait d'urgence de l'axe asservi (synchronisme du couplage électronique, $VC_EG_SYNCDIFF[<axe asservi>])
- Entraînement : seuil d'alarme du circuit intermédiaire (menace de sous-tension imminente), $AA_ESR_STAT[<axe>]
- Entraînement : seuil de vitesse minimale en génératrice (plus d'énergie de rotation réinjectable disponible), $AA_ESR_STAT[<axe>]

Logique combinatoire des actions synchrones statiques : Combinaison de causes/réactions

Les possibilités flexibles de combinaison offertes par les actions synchrones statiques sont utilisées pour déclencher assez rapidement certaines réactions en fonction des différentes causes.

La combinaison de toutes les causes possibles à l'aide d'actions synchrones statiques incombe à l'utilisateur. Il peut évaluer les variables système des causes soit dans leur ensemble, soit de manière sélective à l'aide de masques de bits, puis leur associer les réactions de son choix. Les actions synchrones statiques sont effectives dans tous les modes de fonctionnement.

Bibliographie :
Description fonctionnelle Actions synchrones

Activation

Déblocage des fonctions

Les fonctions de mode génératrice, d'arrêt et de retrait sont débloquées par mise à un du signal de commande correspondant $AA_ESR_ENABLE. Ce signal de commande peut être modifié par des actions synchrones.

Déclenchement des fonctions

La fonction AER est déclenchée simultanément pour l'ensemble des axes concernés par mise à un de la variable système $AC_ESR_TRIGGER.

Le fonctionnement en génératrice est "automatiquement" activé dans l'entraînement dès la détection d'une menace de sous-tension imminente du circuit intermédiaire.

L'arrêt et/ou le retrait contrôlés par le variateur sont activés dès la détection d'une panne de communication (entre la CN et l'entraînement) et d'une sous-tension du circuit intermédiaire dans l'entraînement (sous réserve que la configuration et les paramètres de déblocage le permettent).

L'arrêt et/ou le retrait contrôlés par le variateur peuvent également être déclenchés depuis la CN par mise à un du signal de commande correspondant $AN_ESR_TRIGGER (ordre de diffusion à tous les entraînements).

Bibliographie

Pour obtenir des informations détaillées sur la fonction ESR, voir :

Description fonctionnelle Fonctions spéciales ; Arrêt étendu et retrait (R3)

18.11.1 ESR assisté par CN

18.11.1.1 Retrait assisté par CN (POLF, POLFA, POLFMASK, POLFMLIN)

Pour un retrait assisté par CN, certaines conditions initiales sont requises (voir "Retrait assisté par CN (POLF, POLFA, POLFMASK, POLFMLIN) (Page 708)"). Lorsque ces conditions sont remplies, le relèvement rapide (LIFTFAST) est activé pour le ou les axes de retrait configurés dans le canal par mise à un de la variable système $AC_ESR_TRIGGER (ou $AA_ESR_TRIGGER pour les axes individuels).

Syntaxe

```
POLF(<axe>)=<position>
POLFA(<axe>,<type>,<position>)
POLFMASK(<axe_1>,<axe_2>...)
POLFMLIN(<axe_1>,<axe_2>...)
```

Pour `POLFA`, les formes abrégées suivantes sont autorisées :

```
POLFA(<axe>,<type>) ; forme abrégée pour retrait d'axe individuel
POLFA(axe,0/1/2) ; désactivation ou activation rapide
```

```
POLFA(axe,0,$AA_POLFA[axe]) ; provoque un arrêt du prétraitement des
blocs
POLFA(axe,0) ; ne provoque pas d'arrêt du prétraitement des blocs
```

Signification

POLF :	Adresse d'indication de la position de destination de l'axe de retrait POLF est une fonction modale.				
	<axe> :	Nom de l'axe géométrique ou de l'axe canal/machine qui se retire.			
	<position> :	Position de retrait			
		Type :	REAL		
		Pour les axes géométriques, le système applicable est SCP, sinon SCM. Lorsque les descripteurs d'axe géométrique et d'axe canal/machine sont identiques, le retrait a lieu dans le SCP.			
POLFA :	Appel de sous-programme prédéfini pour l'indication de la position de retrait des axes individuels				
	<axe> :	Descripteur d'axe de canal			
	<type> :	Mode d'indication de position			
		Type :	INT		
		Valeur :	0 :	Identification de la valeur de position comme non valable	
			1 :	La valeur de position est absolue	
			2 :	La valeur de position est relative (distance)	
		Remarque : Si un axe n'est pas un axe individuel ou bien si le type est absent ou nul, une alarme correspondante est générée.			
	<position> :	Position de retrait (cf. supra)			
		Remarque : La valeur de position est également appliquée avec type=0. Toutefois, cette valeur est alors identifiée comme étant non valable et doit être reprogrammée pour le retrait.			
POLFMASK :	Appel de sous-programme prédéfini pour la sélection des axes qui doivent être retirés **indépendamment les uns des autres** après le déclenchement du relèvement rapide.				
	<axe_1>… :	Noms des axes à déplacer sur leurs positions définies avec POLF lors du relèvement rapide. Tous les axes indiqués doivent se trouver dans le même système de coordonnées.			
	L'instruction POLFMASK() sans indication d'un axe désactive le relèvement rapide pour tous les axes qui ont été retirés indépendamment les uns des autres.				
POLFMLIN :	Appel de sous-programme prédéfini pour la sélection des axes qui doivent être retirés **avec corrélation linéaire** après le déclenchement du relèvement rapide.				
	<axe_1>,... :	cf. supra			
	L'instruction POLFMLIN() sans indication d'un axe désactive le relèvement rapide pour tous les axes qui ont été retirés avec corrélation linéaire.				

Remarque

Avant que le relèvement rapide sur une position fixe puisse être débloqué avec `POLFMASK` ou `POLFMLIN`, une position doit avoir été programmée avec `POLF` pour les axes sélectionnés.

Remarque

Si des axes sont débloqués successivement avec `POLFMASK`, `POLFMLIN` ou `POLFMLIN`, `POLFMASK`, la dernière définition est toujours valable pour l'axe correspondant.

Remarque

Les positions programmées avec `POLF` et l'activation avec `POLFMASK` ou `POLFMLIN` sont supprimées au démarrage du programme pièce. Cela signifie que l'utilisateur doit reprogrammer les valeurs de `POLF` et les axes sélectionnés dans `POLFMASK` ou `POLFMLIN` pour chaque programme pièce.

Remarque

Si seul le type est modifié lorsque la forme abrégée `POLFA` est utilisée, l'utilisateur doit veiller à ce que soit la position de retrait soit la distance de retrait contienne une valeur judicieuse. Plus particulièrement, la position et la distance de retrait doivent être redéfinies après un POWER ON.

Exemple

Retrait d'un axe individuel :

Code de programme	Commentaire
`PM37500 $MA_ESR_REACTION[AX1]=21`	`; Retrait assisté par CN.`
`...`	
`$AA_ESR_ENABLE[AX1]=1`	
`POLFA(AX1,1,20.0)`	`; La position de retrait axiale 20.0 (absolue) est affectée à AX1.`
`$AA_ESR_TRIGGER[AX1]=1`	`; Le retrait commence ici.`

Informations complémentaires

Conditions requises pour le retrait assisté par CN

- Un axe de retrait est configuré dans le canal pour le retrait assisté par CN :
  PM37500 $MA_ESR_REACTION = 21
- L'ESR doit être débloqué pour cet axe :
  $AA_ESR_ENABLE = 1
- Des temporisations sont définies :
  PM21380 $MC_ESR_DELAY_TIME1
  PM21381 $MC_ESR_DELAY_TIME2

- Les positions de retrait spécifiques à l'axe sont programmées au moyen de `POLF` dans le programme pièce.
- Les axes sont sélectionnés pour le retrait assisté par CN au moyen de `POLFMASK`/ `POLFMLIN`.
- Les signaux de déblocage pour le mouvement de retrait doivent être mis à 1 et y rester.

Démarrage et libération du retrait assisté par CN

Lorsque la variable système $AC_ESR_TRIGGER est mise à 1 et qu'un axe de retrait est configuré dans ce canal (à savoir PM37500 $MA_ESR_REACTION = 21), $AA_ESR_ENABLE = 1 étant défini pour cet axe, le relèvement rapide (LIFTFAST) est activé dans ce canal.

Le mouvement de relèvement configuré au moyen de `POLF` (ou `LFPOS`) pour les axes sélectionnés au moyen de `POLFMASK` ou `POLFMLIN` remplace le déplacement défini dans le programme pièce pour ces axes.

La temporisation maximale disponible pour le retrait est la somme des temps PM21380 $MC_ESR_DELAY_TIME1 et PM21381 $MC_ESR_DELAY_TIME2. Après l'écoulement de ce laps de temps, un freinage rapide est déclenché également pour l'axe de retrait puis la fonction poursuite est déclenchée.

Remarque

L'extension du mouvement de retrait (c'est-à-dire `LIFTFAST/LFPOS` déclenché par $AC_ESR_TRIGGER) **ne peut pas être interrompue** et ne peut être terminée de manière anticipée que par un arrêt d'urgence.

Remarque

Le retrait déclenché par $AC_ESR_TRIGGER est verrouillé pour empêcher tout retrait multiple.

Retrait d'axe individuel

Pour un retrait d'axe individuel au moyen de `POLFA`, la position de retrait de l'axe doit être programmée et les conditions suivantes respectées :

- $AA_ESR_ENABLE = 1
- L'<axe> doit être un axe individuel à l'instant du déclenchement ($AA_ESR_TRIGGER = 1).
- Le <type> doit être 1 ou 2.

Sens du retrait pour le relèvement rapide

Le frame valable au moment de l'activation du relèvement rapide est pris en compte.

Remarque

Les frames avec rotation influencent également, via `POLF`, le sens dans lequel le relèvement a lieu.

Permutation d'axe

Les axes de retrait doivent toujours être affectés à un canal CN et ne doivent pas être échangés entre les canaux. Si une tentative de faire passer un axe de retrait dans un autre canal a lieu,

une alarme est générée. Ce n'est qu'une fois que l'axe a été à nouveau désactivé avec $AA_ESR_ENABLE[AX] = 0 que l'axe peut être permuté dans un nouveau canal. Une fois la permutation effectuée, les axes peuvent de nouveau être assignés avec $AA_ESR_ENABLE[AX] = 1.

Axes neutres

Des axes neutres ne peuvent pas exécuter d'ESR assisté par CN.

18.11.1.2 Arrêt assisté par CN

La mise à un de la variable système $AC_ESR_TRIGGER (ou $AA_ESR_TRIGGER pour les axes individuels) active l'arrêt assisté par CN pour le ou les axes d'immobilisation configurés dans le canal.

Conditions

- Un axe d'immobilisation est configuré dans le canal pour l'arrêt assisté par CN :
  PM37500 $MA_ESR_REACTION = 22
- L'ESR doit être débloqué pour cet axe :
  $AA_ESR_ENABLE = 1
- Des temporisations sont définies :
  PM21380 $MC_ESR_DELAY_TIME1 (temporisation axes AER)
  PM21381 $MC_ESR_DELAY_TIME2 (temps AER pour freinage avec interpolation)

Déroulement

Pendant le laps de temps du paramètre machine PM21380, l'interpolation de l'axe continue tranquillement suivant la programmation. Après écoulement du laps de temps du paramètre machine PM21380, le freinage piloté par interpolation (rampe d'arrêt) est amorcé. Pour le freinage piloté avec interpolation, le délai maximal disponible est celui spécifié dans PM21381. Après écoulement de ce délai, le freinage rapide est amorcé suivi de l'asservissement.

Exemple

Immobilisation d'un axe individuel :

```
Code de programme                                   Commentaire
PM37500 $MC_ESR_REACTION[AX1]=22                    ; Arrêt assisté par CN.
PM21380 $MC_ESR_DELAY_TIME1[AX1]=0.3
PM21381 $MC_ESR_DELAY_TIME2[AX1]=0.06
...
$AA_ESR_ENABLE[AX1]=1
$AA_ESR_TRIGGER[AX1]=1                              ; L'immobilisation commence ici.
```

18.11.2 ESR contrôlé par le variateur

18.11.2.1 Configuration d'un arrêt contrôlé par le variateur (ESRS)

La fonction `ESRS(...)` permet de configurer les paramètres d'entraînement pour l'"arrêt" de la fonction AER contrôlée par le variateur.

Syntaxe

```
ESRS(<axe_1>,<temps_arrêt_1>[,...,<axe_n>,<temps_arrêt_n>])
```

Signification

`ESRS(...)` :	Fonction d'écriture des paramètres d'entraînement pour la fonction ESR "Arrêt" La fonction : • doit figurer seule dans le bloc, • déclenche un arrêt du prétraitement des blocs, • ne peut pas être utilisée dans des actions synchrones.	
`<axe_1>`, `...`, `<axe_n>` :	Axe pour lequel l'arrêt contrôlé par le variateur doit être configuré. Le paramètre d'entraînement p0888 (configuration) est écrit dans l'entraînement pour cet axe : p0888 = 1	
	Type :	AXIS
	Plage de valeurs :	Descripteur d'axe de canal
`<temps_arrêt_1>`, `...`, `<temps_arrêt_n>` :	Durée pendant laquelle l'entraînement poursuit un fonctionnement constant à la vitesse de consigne actuelle après l'apparition d'un défaut. Le paramètre d'entraînement p0892 (temporisation) est écrit dans l'entraînement pour l'axe spécifié: p0892 = <temps d'arrêt>	
	Unité :	s
	Type :	REAL
	Plage de valeurs :	0.00 - 20.00
5 axes au maximum peuvent être configurés dans un appel de fonction ; n = 5		

18.11.2.2 Configuration d'un retrait contrôlé par le variateur (ESRR)

La fonction `ESRR(...)` permet de configurer les paramètres d'entraînement pour le "retrait" de la fonction AER contrôlée par le variateur.

Syntaxe

```
ESRR(<axe_1>,<distance_retrait_1>,<vitesse_retrait_1>[,...,<axe_n>,<
distance_retrait_n>,<vitesse_retrait_n>])
```

Signification

`ESRR(...)` :	Fonction d'écriture des paramètres d'entraînement pour la fonction ESR "Retrait" La fonction : • doit figurer seule dans le bloc, • déclenche un arrêt du prétraitement des blocs, • ne peut pas être utilisée dans des actions synchrones.	

`<axe_1>,`
`...,`
`<axe_n>` :
	axe pour lequel le retrait contrôlé par le variateur doit être configuré.
	Le paramètre d'entraînement p0888 (configuration) est écrit dans l'entraînement pour cet axe :
	p0888 = 2	

	Type :	AXIS
	Plage de valeurs :	Descripteur d'axe de canal

`<distance_retrait_1>,`
`...,`
`<distance_retrait_n>` :
	La distance de retrait est convertie en vitesse de retrait pour l'entraînement. La valeur est écrite dans le paramètre d'entraînement p0893 (vitesse) pour l'axe spécifié :
	p0893 = (<distance_retrait_n> convertie en vitesse de retrait)	

	Unité :	mm/min, inch/min, degré/min (en fonction de l'unité de l'axe)
	Type :	REAL
	Plage de valeurs :	MIN - MAX

`<vitesse_`
`retrait_1>,`
`...,`
`<vitesse_`
`retrait_n>` :
	La vitesse de retrait est convertie en durée pour l'entraînement. La valeur est écrite dans le paramètre d'entraînement p0892 [s] pour l'axe spécifié :
	p0892 = <distance_retrait_n> / <vitesse_retrait _n>	

	Unité :	mm/min, inch/min, degré/min (en fonction de l'unité de l'axe)
	Type :	REAL
	Plage de valeurs :	0.00 - MAX
5 axes au maximum peuvent être configurés dans un appel de fonction ; n = 5		

18.12 Définition d'une pièce brute (WORKPIECE)

La commande doit connaître la forme et la taille d'une pièce brute pour pouvoir l'afficher dans la simulation graphique. L'utilisateur a donc la possibilité de définir des pièces brutes via l'interface utilisateur ou directement dans le programme CN. Les définitions de pièces brutes sont maintenues après un reset (fin de programme, canal, GMFC). Elles sont supprimées automatiquement au démarrage suivant de la commande.

Syntaxe

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

Signification

`WORKPIECE(...)` :		Procédure prédéfinie pour la définition d'une pièce brute		
		Arrêt du prétraitement des blocs :	oui	
		Uniquement dans le bloc :	oui	
Paramètres :				
1	`"<WP>"` :	Nom de la pièce (facultatif)		
		Type de données :	STRING	
		L'indication est nécessaire uniquement lorsqu'il peut y avoir plusieurs pièces dans un canal. Sans indication, "WORKP<n>" est automatiquement pris en compte, où <n> correspond au numéro du canal déclaratif.		
2	`"<RefP>"` :	Ablocage (facultatif ; uniquement pour les fraiseuses)		
		Type de données :	STRING	
		Plage de valeurs :	"Plateau"	Ablocage sur le plateau fixe
			"A"	Ablocage sur l'axe rotatif A
			"B"	Ablocage sur l'axe rotatif B
			"C"	Ablocage sur l'axe rotatif C
		Condition : Le plateau ou l'axe rotatif doit être débloqué pour l'ablocage de pièce brute via le paramètre machine correspondant (voir Manuel de mise en service SINUMERIK Operate).		
3	`"<ZeroOffset>"` :	Décalage d'origine réglable pour le placement de la pièce brute (non programmable) La sélection d'un décalage d'origine réglable pour le placement de la pièce brute est proposée uniquement lors de l'introduction de pièce brute via l'interface utilisateur. Au lieu de cela, en cas de définition directe de la pièce brute dans le programme pièce, la pièce brute se rapporte toujours au décalage d'origine momentanément en vigueur.		

4	"<Type>" :	Forme de la pièce brute		
		Type de données :	STRING	
		Plage de valeurs :	"CYLINDER" :	Cylindre
			"PIPE" :	Tube
			"RECTANGLE" :	Parallélépipède centré
			"BOX" :	Parallélépipède rectangulaire
			"N_CORNER" :	Polygone
5 ... 12	<Par5> ... <Par12> :	Paramètre de description de la forme de la pièce brute		
		Type de données :	REAL	
		Le nombre de paramètres nécessaires et leur signification dépendent de la forme de la pièce brute et de la valeur du paramètre de bit. Voir : • Tableau "Paramètre de description de la forme de la pièce brute" • Tableau "Paramètre de bit"		
WORPIECE() :		Un appel de WORKPIECE sans paramètre supprime toutes les définitions de pièces brutes.		
WORPIECE(<WP>) :		Un appel de WORKPIECE avec nom de pièce supprime uniquement cette définition de pièce brute.		

Tableau 18-1 Paramètre de description de la forme de la pièce brute

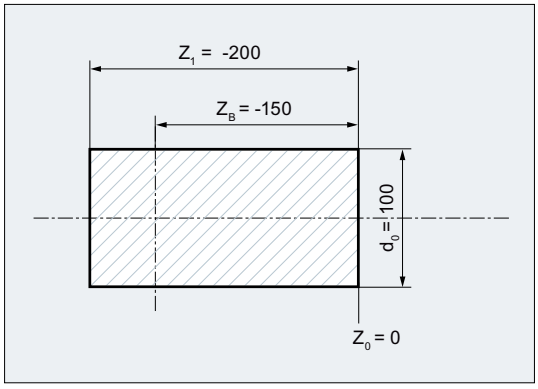
Forme de la pièce brute	Paramètres							
	<Par5>	<Par6>	<Par7>	<Par8>	<Par9>	<Par10>	<Par11>	<Par12>
Cylindre	Paramètre de bit Valeur réelle interprétée comme nombre entier codé sur bits. Les bits définissent la signification des paramètres suivants (voir tableau "Paramètre de bit").	Point de référence Z_0	Longueur Z_1	Cote d'usinage Z_B	Diamètre extérieur d_0	-	Rotation autour de l'axe rotatif	-
Tube		Point de référence Z_0	Longueur Z_1	Cote d'usinage Z_B	Diamètre extérieur d_0	Épaisseur de paroi (rel) / Diamètre intérieur d_1 (abs)	Rotation autour de l'axe rotatif	-
Parallélépipède centré		Point de référence Z_0	Longueur Z_1	Cote d'usinage Z_B	Largeur W	Longueur L	Rotation autour de l'axe rotatif	-
Parallélépipède rectangulaire		Point de référence Z_0	Longueur Z_1	Cote d'usinage Z_B	X_0	Y_0	X_1	Y_1
Polygone		Point de référence Z_0	Longueur Z_1	Cote d'usinage Z_B	Nombre de coins	Cote sur plats	Rotation autour de l'axe rotatif	-

Tableau 18-2 Paramètre de bit

Bit	Signification	
4 (0x0010)	Parallélépipède rectangulaire : X_1	
	= 0	rel
	= 1	abs
5 (0x0020)	Parallélépipède rectangulaire : Y_1	
	= 0	rel
	= 1	abs
6 (0x0040)	Longueur Z_1 (cote finale)	
	= 0	rel
	= 1	abs
Bit 7 (0x0080)	Cote d'usinage Z_B	
	= 0	rel
	= 1	abs
Bit 8 (0x0100)	Tube : Épaisseur de paroi / Diamètre intérieur	
	= 0	rel
	= 1	abs
9 (0x0200)	Polygone	
	= 0	Cote sur plats
	= 1	Longueur d'arête
12 (0x1000)	Ablocage pour tours	
	= 0	Broche principale
	= 1	Contre-broche
13 (0x2000)	Contre-broche	
	= 0	Avec fonction miroir
	= 1	Sans fonction miroir

Exemples

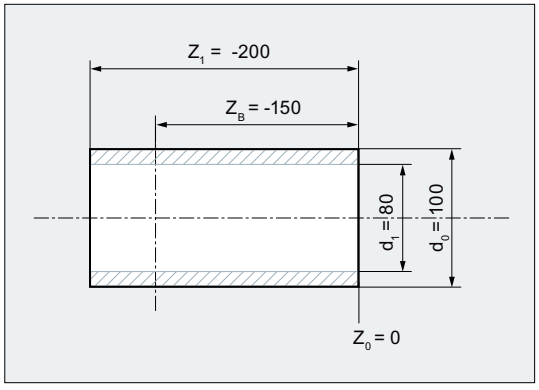
Exemple 1 : Pièce brute cylindrique sur un tour



Code de programme	Commentaire
...	

Code de programme	Commentaire
WORKPIECE(,,,"CYLINDER",0,0,-200,-150,100)	; **Définition de la pièce brute :** ; Forme de la pièce brute : Cylindre ; Paramètre de bit=0(aucun bit activé) → Les valeurs de longueur et de cote d'usinage sont relatives, pièce brute sur la broche principale ; Point de référence(Z0)=0 ; Longueur(Z1)=-200 ; Cote d'usinage(ZB)=-150 ; Diamètre extérieur(d0)=100
...	

Exemple 2 : Pièce brute tubulaire sur un tour



Code de programme	Commentaire
...	
WORKPIECE(,,,"PIPE",256,0,-200,-150,100,80)	; **Définition de la pièce brute :** ; Forme de la pièce brute : Tube ; Paramètre de bit=256(bit8=1) → Le diamètre intérieur est absolu, la longueur et la cote d'usinage sont relatives, pièce brute sur la broche principale ; Point de référence(Z0)=0 ; Longueur(Z1)=-200 ; Cote d'usinage(ZB)=-150 ; Diamètre extérieur(d0)=100 ; Diamètre intérieur(d1)=80
...	

18.13 Basculement du mode de langage (G290, G291)

La commande offre la possibilité de lire et d'exécuter des programmes pièce à partir de systèmes CNC externes. La condition est que le mode de langage CN (dialecte ISO) ait été défini lors de la mise en service.

Bibliographie :
Description fonctionnelle Dialecte ISO

Le mode en dialecte ISO peut être activé séparément pour chaque canal. Par exemple, le canal 1 peut fonctionner dans le mode en dialecte ISO tandis que le canal 2 est en mode SINUMERIK.

Le basculement entre le mode SINUMERIK et le mode en dialecte ISO s'effectue dans le programme CN via les instructions du groupe G 47. Le basculement n'influence pas l'outil actif, les corrections d'outil et les décalages d'origine.

Syntaxe

```
G291
...
G290
```

Signification

G290:	Activer mode de langage SINUMERIK	
	Uniquement dans le bloc :	oui
	Prise d'effet :	modale
G291:	Activer mode de langage ISO	
	Uniquement dans le bloc :	oui
	Prise d'effet :	modale

Conditions

Mode SINUMERIK

- Le préréglage des instructions G peut être défini pour chaque canal via un paramètre machine.
- Le mode SINUMERIK ne permet pas de programmation d'instructions en dialectes ISO.

Mode en dialecte ISO

- En tant que réglage de base de la commande, le mode en dialecte ISO peut être réglé par le biais de paramètres machine. Par défaut, la commande démarre ensuite dans le mode en dialecte ISO.
- Il est uniquement possible de programmer des instructions G du dialecte ISO. La programmation de fonctions SINUMERIK est impossible dans le mode en dialecte ISO.

- Une combinaison de dialecte ISO et de langage SINUMERIK est impossible dans un même bloc CN.
- Un basculement entre le dialecte ISO M (fraisage) et le dialecte ISO T (tournage) par le biais d'instructions G est impossible.
- L'appel de sous-programmes programmés en mode SINUMERIK est possible.
- Lorsque des fonctions SINUMERIK doivent être utilisées, il faut d'abord basculer en mode SINUMERIK (voir exemple).

Exemple

Compactage de blocs linéaires dans le mode en dialecte ISO

Code de programme	**Commentaire**
N5 G290	; Activer le mode de langage SINUMERIK.
N10 COMPON	; COMPON est une instruction du langage Siemens activant une fonction de compactage qui remplace des blocs linéaires consécutifs par des blocs de polynôme avec des longueurs de trajectoire aussi grandes que possible.
N15 G291	; Activer le mode de langage ISO.
N20 G01 X100 Y100 F1000	; Étant donné que COMPON a été activé en mode SINUMERIK, cette fonction peut également comprimer des blocs linéaires dans le mode en dialecte ISO.
...	

Programmes de chariotage personnalisés 19

19.1 Fonctions additionnelles pour le chariotage

Pour le chariotage, on vous propose des cycles d'usinage prêts à l'usage. En plus, vous avez la possibilité de créer vos propres cycles de chariotage grâce aux fonctions élaborées suivantes :

- Création d'une table de contour (CONTPRON)
- Création d'une table de contour codée (CONTDCON)
- Désactivation de la préparation du contour (EXECUTE)
- Détermination de l'intersection entre deux éléments de contour (INTERSEC) (uniquement pour des tables créées avec CONTPRON)
- Exécution des éléments de contour d'une table bloc par bloc (EXECTAB) (uniquement pour des tables créées avec CONTPRON)
- Calcul de données de cercles (CALCDAT)

Remarque

Ces fonctions sont utilisables de façon universelle et pas seulement pour le chariotage.

Conditions

Avant tout appel des fonctions CONTPRON ou CONTDCON, il faut :

- accoster un point de départ permettant un usinage sans collision,
- désactiver la correction du rayon du tranchant avec `G40`.

19.2 Création d'une table de contour (CONTPRON)

La fonction `CONTPRON` permet d'activer la préparation du contour. Les blocs CN appelés par la suite ne sont pas exécutés, mais décomposés en déplacements élémentaires qui sont rangés dans la table de contour. A chaque élément de contour correspond une ligne de la table de contour à deux dimensions. Le nombre de détalonnages qui a été déterminé est indiqué en retour.

Syntaxe

Activer la préparation du contour :
```
CONTPRON(<table de contour>,<type d'usinage>,<détalonnages>,
<sens d'usinage>)
```

Désactiver la préparation du contour et revenir au mode de traitement normal :
```
EXECUTE(<ERREUR>)
```

Voir "Désactivation de la préparation du contour (EXECUTE) (Page 737)"

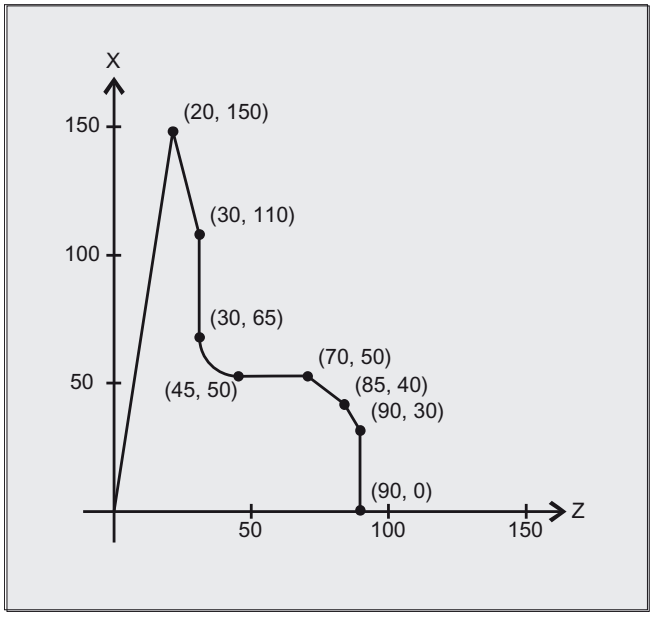
Signification

`CONTPRON` :	Procédure prédéfinie pour activer la préparation du contour pour la création d'une table de contour		
`<Table de contour>` :	Nom de la table de contour		
`<Type d'usinage>` :	Paramètre pour le type d'usinage		
	Type :	CHAR	
	Valeur :	`"G"` :	longitudinale : usinage intérieur
		`"L"` :	longitudinale : usinage extérieur
		`"N"` :	transversale : usinage intérieur
		`"P"` :	transversale : usinage extérieur
`<Détalonnages>` :	Variable de résultat pour le nombre d'éléments de détalonnage générés		
	Type :	INT	
`<Sens d'usinage>` :	Paramètre pour le sens d'usinage		
	Type :	INT	
	Valeur :	0	Préparation du contour en avant (valeur par défaut)
		1	Préparation du contour dans les deux sens

Exemple 1

Création d'une table de contour avec :

- le nom "TABC"
- 30 éléments de contour au maximum (arcs de cercle, segments de droite)
- une variable pour le nombre d'éléments de détalonnage générés
- une variable pour les messages d'erreur



Programme CN :

Code de programme	**Commentaire**
N10 DEF REAL TABC[30,11]	; Table de contour nommée TABC avec 30 éléments de contour au maximum. La valeur de paramètre 11 (nombre de colonnes de la table) est une grandeur fixe.
N20 DEF INT NBDETAL	; Variable pour le nombre d'éléments de détalonnage, nommée NBDETAL.
N30 DEF INT ERREUR	; Variable pour la signalisation d'erreur (0=aucune erreur, 1=erreur).
N40 G18	
N50 CONTPRON(TABC,"G",NBDETAL)	; Activer la préparation du contour.
N60 G1 X150 Z20	; N60 à N120 : Description du contour
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(ERREUR)	; Fin du remplissage de la table de contour et retour au mode de programme normal.
N140 …	; Suite du traitement de la table.

Table de contour TABC :

Indice Ligne	Colonne									
(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)

7	7	11	0	0	20	150	0	82.40535663	0	0
0	2	11	20	150	30	110	-1111	104.0362435	0	0
1	3	11	30	110	30	65	0	90	0	0
2	4	13	30	65	45	50	0	180	45	65
3	5	11	45	50	70	50	0	0	0	0
4	6	11	70	50	85	40	0	146.3099325	0	0
5	7	11	85	40	90	30	0	116.5650512	0	0
6	0	11	90	30	90	0	0	90	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Signification du contenu des colonnes :

(0) Index pointant l'élément de contour suivant (le numéro de ligne de la même table)

(1) Index pointant l'élément de contour précédent

(2) Codage du mode de contour pour le déplacement

Valeurs possibles pour X = abc

a = 10^2 G90 = 0 G91 = 1

b = 10^1 G70 = 0 G71 = 1

c = 10^0 G0 = 0 G1 = 1 G2 = 2 G3 = 3

(3), (4) Point de départ des éléments de contour

(3) = abscisse, (4) = ordonnée dans le plan courant

(5), (6) Point final des éléments de contour

(5) = abscisse, (6) = ordonnée dans le plan courant

(7) Indicateur maxi/mini : indique les maxima et minima locaux du contour

(8) Valeur maximale entre élément de contour et abscisse (dans le cas d'un usinage longitudinal) ou ordonnée (en surfaçage). L'angle dépend du type d'usinage programmé.

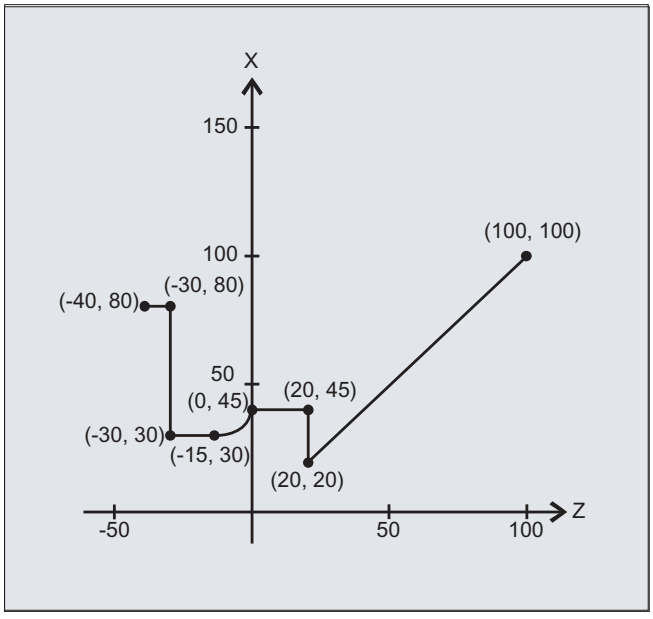
(9), (10) Coordonnées du centre de l'élément de contour, si c'est un arc de cercle..

(9) = abscisse, (10) = ordonnée

Exemple 2

Création d'une table de contour avec

- le nom TABC
- 92 éléments de contour au maximum (arcs de cercle, segments de droite)
- Mode d'usinage : cylindrage, usinage extérieur
- Préparation dans les deux sens



Programme CN :

Code de programme	Commentaire
N10 DEF REAL TABC[92,11]	; Table de contour nommée TABC avec 92 éléments de contour au maximum. La valeur de paramètre 11 est une grandeur fixe.
N20 DEF CHAR MU="L"	; Mode de fonctionnement pour CONTPRON : cylindrage, usinage extérieur
N30 DEF INT ED=0	; Nombre d'éléments de détalonnage=0
N40 DEF INT MODE=1	; Préparation dans les deux sens
N50 DEF INT ERR=0	; Signalisation d'erreur
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(TABC,BT,HE,MODE)	; Activer la préparation du contour.
N110 G1 G90 Z20 X20	
N120 X45	
N130 Z0	
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)	
N150 G1 Z-30	
N160 X80	
N170 Z-40	
N180 EXECUTE(ERR)	; Fin du remplissage de la table de contour et retour au mode de programme normal.
...	

Table de contour TABC :

Après la préparation du contour, le contour est disponible dans les deux sens.

Indice	Colonne										
Ligne	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
0	6[1)]	7[2)]	11	100	100	20	20	0	45	0	0
1	0[3)]	2	11	20	20	20	45	-3	90	0	0
2	1	3	11	20	45	0	45	0	0	0	0
3	2	4	12	0	45	-15	30	5	90	-15	45
4	3	5	11	-15	30	-30	30	0	0	0	0
5	4	7	11	-30	30	-30	45	-1111	90	0	0
6	7	0[4)]	11	-30	80	-40	80	0	0	0	0
7	5	6	11	-30	45	-30	80	0	90	0	0
8	1[5)]	2[6)]	0	0	0	0	0	0	0	0	0
	...										
83	84	0[7)]	11	20	45	20	80	0	90	0	0
84	90	83	11	20	20	20	45	-1111	90	0	0
85	0[8)]	86	11	-40	80	-30	80	0	0	0	0
86	85	87	11	-30	80	-30	30	88	90	0	0
87	86	88	11	-30	30	-15	30	0	0	0	0
88	87	89	13	-15	30	0	45	-90	90	-15	45
89	88	90	11	0	45	20	45	0	0	0	0
90	89	84	11	20	45	20	20	84	90	0	0
91	83[9)]	85[10)]	11	20	20	100	100	0	45	0	0

Signification du contenu des colonnes et des remarques concernant les lignes 0, 1, 6, 8, 83, 85 et 91

Les significations du contenu des colonnes données dans l'exemple 1 sont valables.

Toujours dans ligne 0 :

1) Précédent : ligne n contient la fin du contour en avant

2) Suivant : ligne n est la fin de la table de contour en avant

Une fois pour les éléments de contour en avant :

3) Précédent : début du contour (en avant)

4) Suivant : fin du contour (en avant)

Toujours dans ligne fin de table de contour (en avant) +1 :

5) Précédent : nombre de détalonnages en avant

6) Suivant : nombre de détalonnages en arrière

Une fois pour les éléments de contour en arrière :

7) Suivant : fin du contour (en arrière)

8) Précédent : début du contour (en arrière)

Toujours dans la dernière ligne de la table :

9) Précédent : ligne n est le début de la table de contour (en arrière)

10) Suivant : ligne n contient le début du contour (en arrière)

Informations complémentaires

Instructions de déplacements autorisées, système de coordonnées

Pour la programmation du contour, les instructions G suivantes sont autorisés :

- Groupe G 1 : `G0`, `G1`, `G2`, `G3`

Possibles aussi :

- Arrondi et chanfrein
- Programmation de cercle avec `CIP` et `CT`

Les fonctions spline, polynôme et filetage provoquent des erreurs.

Entre `CONTPRON` et `EXECUTE`, la modification du système de coordonnées par activation d'un frame est illicite. Ceci est également valable pour la permutation entre `G70` et `G71` ou entre `G700` et `G710`.

Pendant la préparation de la table de contour , une permutation d'axe avec `GEOAX` entraîne une alarme.

Eléments de détalonnage

La description des différents éléments de détalonnage peut être réalisée sous la forme d'un sous-programme ou de blocs isolés, au choix.

Chariotage indépendamment du sens de contour programmé

La préparation du contour avec `CONTPRON` a été étendue de sorte qu'après son appel, la table de contour est disponible quel que soit le sens programmé.

19.3 Création d'une table de contour codée (CONTDCON)

En cas d'activation de la préparation du contour avec CONTDCON, les blocs CN appelés par la suite sont enregistrés sous forme codée dans une table de contour de 6 colonnes pour simplifier la mémorisation. A chaque élément de contour correspond une ligne de la table de contour. Compte tenu des règles de codage indiquées ci-dessous, vous pouvez créer des programmes en code DIN à partir des lignes de la table, par exemple pour des cycles. Dans la ligne numéro 0 sont enregistrées les données du point de départ.

Syntaxe

Activer la préparation du contour :
```
CONTDCON(<table de contour>,<sens d'usinage>)
```

Désactiver la préparation du contour et revenir au mode de traitement normal :
```
EXECUTE(<ERREUR>)
```

Voir "Désactivation de la préparation du contour (EXECUTE) (Page 737)"

Signification

CONTDCON :	Procédure prédéfinie pour activer la préparation du contour pour la création d'une table de contour codée		
<Table de contour> :	Nom de la table de contour		
<Sens d'usinage> :	Paramètre pour le sens d'usinage		
	Type :	INT	
	Valeur :	0	Préparation du contour suivant la séquence de blocs de contour (valeur par défaut)
		1	non admis

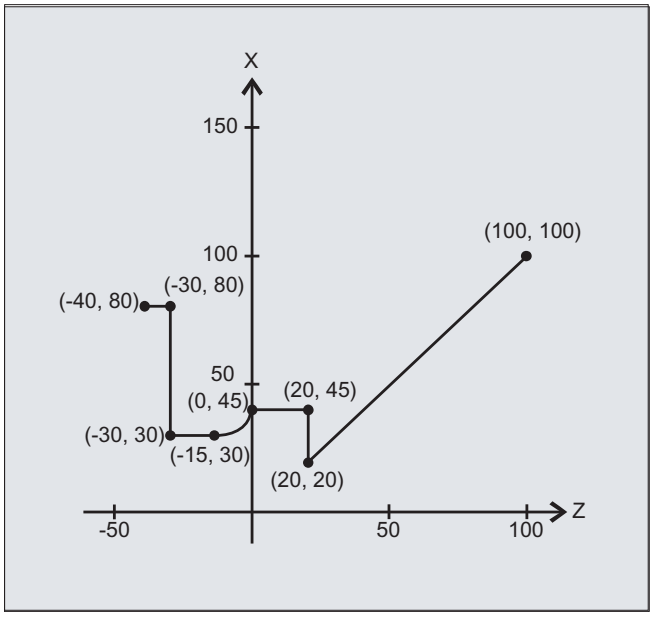
Remarque

Pour CONTDCON, les instructions G admises dans la section de programme à décoder sont plus nombreuses que pour la fonction CONTPRON. En outre, l'avance et le type d'avance sont enregistrés pour chaque élément de contour.

Exemple

Création d'une table de contour avec :

- le nom "TABC"
- les éléments de contour (arcs de cercle, segments de droite)
- Mode d'usinage : tournage
- Sens d'usinage : vers l'avant



Programme CN :

Code de programme	Commentaire
N10 DEF REAL TABC[9,6]	; Table de contour nommée TABC à 9 lignes. 8 blocs de contour sont donc possibles. La valeur 6 (nombre de colonnes de la table) est imposée.
N20 DEF INT MODE = 0	; Variable pour le sens d'usinage. Valeur par défaut 0 : uniquement dans le sens de contour programmé.
N30 DEF INT ERROR = 0	; Variable pour la signalisation d'erreur.
...	
N100 G18 G64 G90 G94 G710	
N101 G1 X100 Z100 F1000	
N105 CONTDCON (TABC, MODE)	; Appel de la préparation du contour (MODE peut être omis).
N110 G1 Z20 X20 F200	; Description du contour.
N120 G9 X45 F300	
N130 Z0 F400	
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)F100	
N150 G64 Z-30 F600	
N160 X80 F700	
N170 Z-40 F800	
N180 EXECUTE(ERROR)	; Fin du remplissage de la table de contour et retour au mode de programme normal.
...	

Table de contour TABC :

	Indice de colonne					
	0	1	2	3	4	5
Indice ligne	Mode de contour	Point final abscisse	Point final ordonnée	Centre abscisse	Centre ordonnée	Avance
0	30	100	100	0	0	7
1	11031	20	20	0	0	200
2	111031	20	45	0	0	300
3	11031	0	45	0	0	400
4	11032	-15	30	-15	45	100
5	11031	-30	30	0	0	600
6	11031	-30	80	0	0	700
7	11031	-40	80	0	0	800
8	0	0	0	0	0	0

Signification du contenu des colonnes :

Ligne 0 : Codes du **point de départ :**

Colonne 0 : 10^0 (position des unités) : G0 = 0
10^1 (position des dizaines) : G70 = 0, G71 = 1, G700 = 2, G710 = 3
Colonne 1 : point de départ abscisse
Colonne 2 : point de départ ordonnée
Colonne 3-4 : 0
Colonne 5 : indice de ligne du dernier élément de contour dans la table

Lignes 1-n : Valeurs des **éléments de contour**

Colonne 0 : 10^0 (position des unités) : G0 = 0, G1 = 1, G2 = 2, G3 = 3
10^1 (position des dizaines) : G70 = 0, G71 = 1, G700 = 2, G710 = 3
10^2 (position des centaines) : G90 = 0, G91 = 1
10^3 (position des milliers) : G93 = 0, G94 = 1, G95 = 2, G96 = 3
10^4 (position des dizaines de milliers) : G60 = 0, G44 = 1, G641 = 2, G642 = 3
10^5 (position des centaines de milliers) : G9 = 1
Colonne 1 : Point final abscisse
Colonne 2 : Point final ordonnée
Colonne 3 : centre abscisse en interpolation circulaire
Colonne 4 : centre ordonnée en interpolation circulaire
Colonne 5 : Avance

Informations complémentaires

Instructions de déplacements autorisées, système de coordonnées

Pour la programmation du contour, les groupes G et les instructions G suivants sont autorisés :

Groupe G 1 : `G0`, `G1`, `G2`, `G3`
groupe G 10 : `G60`, `G64`, `G641`, `G642`
groupe G 11 : `G9`
groupe G 13 : `G70`, `G71`, `G700`, `G710`
groupe G 14 : `G90`, `G91`
groupe G 15 : `G93`, `G94`, `G95`, `G96`, `G961`

Possibles aussi :

- Arrondi et chanfrein
- Programmation de cercle avec `CIP` et `CT`

Les fonctions spline, polynôme et filetage provoquent des erreurs.

Entre `CONTDCON` et `EXECUTE`, la modification du système de coordonnées par activation d'un frame est illicite. Ceci est également valable pour la permutation entre `G70` et `G71` ou entre `G700` et `G710`.

Pendant la préparation de la table de contour , une permutation d'axe avec `GEOAX` entraîne une alarme.

Sens d'usinage

La table de contour générée avec `CONTDCON` est prévue pour le chariotage dans le sens programmé du contour.

19.4 Détermination de l'intersection entre deux éléments de contour (INTERSEC)

`INTERSEC` détermine le point d'intersection de deux éléments de contour normés de la table de contour créée avec `CONTPRON`.

Syntaxe

```
<Etat>=INTERSEC(<table de contour_1>[<élément de contour_1>],
<table de contour_2>[<élément de contour_2>],<intersection>,<type
d'usinage>)
```

Signification

`INTERSEC` :	Fonction prédéfinie pour déterminer l'intersection de deux éléments de contour issus de tables de contour générées avec `CONTPRON`		
`<état>` :	Variable de l'état de l'intersection		
	Type :	BOOL	
	Valeur :	TRUE	existence d'une intersection
		FALSE	pas d'intersection
`<table de contour_1>` :	Nom de la première table de contour		
`<élément de contour_1>` :	Numéro de l'élément de contour de la première table de contour		
`<table de contour_2>` :	Nom de la deuxième table de contour		
`<élément de contour_2>` :	Numéro de l'élément de contour de la deuxième table de contour		
`<intersection>` :	Coordonnées du point d'intersection dans le plan actif (`G17` / `G18` / `G19`)		
	Type :	REAL	
`<Type d'usinage>` :	Paramètre pour le type d'usinage		
	Type :	INT	
	Valeur :	0	Calcul du point d'intersection dans le plan activé avec le paramètre 2 (valeur par défaut)
		1	Calcul du point d'intersection indépendamment du plan transmis

Remarque

Les variables doivent avoir été définies avant leur utilisation.

La transmission des contours exige le respect des valeurs définies avec `CONTPRON` :

Paramètre	Signification
2	Codage du mode de contour pour le déplacement
3	Abscisse du point de départ du contour
4	Ordonnée du point de départ du contour
5	Abscisse du point final du contour
6	Ordonnée du point final du contour

Paramètre	Signification
9	Coordonnées du centre pour l'abscisse (uniquement pour le contour du cercle)
10	Coordonnées du centre pour l'ordonnée (uniquement pour le contour du cercle)

Exemple

Il s'agit de déterminer l'intersection de l'élément de contour 3 de la table TABNAME1 et de l'élément de contour 7 de la table TABNAME2. Les coordonnées de l'intersection, dans le plan actif, sont rangées dans la variable ISCOORD (1er élément = abscisse, 2e élément = ordonnée). S'il n'existe pas d'intersection, il y a saut vers PASINTER (aucune intersection trouvée).

Code de programme	**Commentaire**
`DEF REAL TABNAME1[12,11]`	`; Table de contour 1`
`DEF REAL TABNAME2[10,11]`	`; Table de contour 2`
`DEF REAL ISCOORD [2]`	`; Variable pour les coordonnées du point d'intersection.`
`DEF BOOL ISPOINT`	`; Variable pour l'état du point d'intersection.`
`DEF INT MODE`	`; Variable pour le type d'usinage.`
…	
`MODE=1`	`; Calcul indépendant du plan actif.`
`N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE)`	`; Appel de la fonction Intersection des éléments de contour.`
`N20 IF ISPOINT==FALSE GOTOF PASINTER`	`; Saut vers PASINTER.`
…	

19.5 Exécution des éléments de contour d'une table bloc par bloc (EXECTAB)

La fonction EXECTAB permet d'exécuter bloc par bloc les éléments de contour d'une table qui a été créée avec CONTPRON, par exemple.

Syntaxe

```
EXECTAB(<table de contour>[<élément de contour>])
```

Signification

EXECTAB :	Procédure prédéfinie pour exécuter un élément de contour
<Table de contour> :	Nom de la table de contour
<Elément de contour> :	Numéro de l'élément de contour

Exemple

Les éléments de contour 0 à 2 de la table TABC doivent être exécutés bloc par bloc.

Code de programme	**Commentaire**
N10 EXECTAB (TABC[0])	; Exécuter l'élément 0 de la table TABC.
N20 EXECTAB(TABC[1])	; Exécuter l'élément 1 de la table TABC.
N30 EXECTAB(TABC[2])	; Exécuter l'élément 2 de la table TABC.

19.6 Calcul de données de cercles (CALCDAT)

La fonction `CALCDAT` permet de calculer le rayon et les coordonnées du centre d'un cercle à partir de trois ou quatre points connus du cercle. Les points indiqués doivent être distincts.

Lorsque 4 points qui ne se trouvent pas exactement sur un cercle sont indiqués, une moyenne est formée pour le centre du cercle et le rayon.

Remarque

Règle de calcul pour la formation de moyenne

Le calcul de l'arc de cercle est effectué 4 fois :

1. avec les points 1, 2, 3 du cercle
2. avec les points 1, 2, 4 du cercle
3. avec les points 1, 3, 4 du cercle
4. avec les points 2, 3, 4 du cercle

La valeur d'abscisse et la valeur d'ordonnée des coordonnées du centre du cercle sont calculées en additionnant les valeurs d'abscisse ou les valeurs d'ordonnée des quatre calculs d'arc de cercle, puis en divisant le résultat par 4.

Le rayon est calculé en déterminant la racine carrée de la somme des quatre rayons des calculs d'arc de cercle, puis en multipliant le résultat par 0,5.

Syntaxe

```
<Etat>=CALCDAT(<points du
cercle>[<nombre>,<type>],<nombre>,<résultat>)
```

Signification

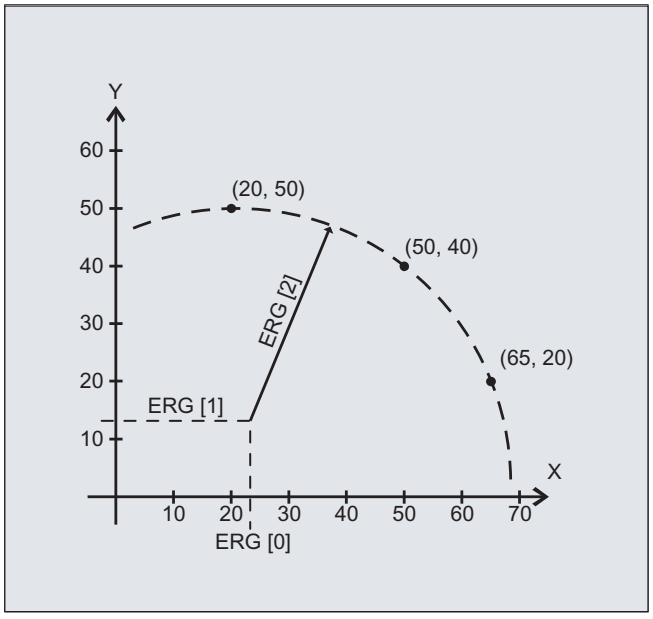
`CALCDAT` :	Fonction prédéfinie pour calculer le rayon et les coordonnées du centre d'un cercle à partir de 3 ou 4 points		
`<Etat>` :	Variable de l'état du calcul du cercle		
	Type :	BOOL	
	Valeur :	TRUE	Les points indiqués se situent sur un cercle.
		FALSE	Les points indiqués ne se situent **pas** sur un cercle.
`<Points du cercle>[]` :	Variable pour l'indication des points du cercle avec les paramètres :		
	`<Nombre>` :	Nombre de points du cercle (3 ou 4)	
	`<Type>` :	Type de coordonnées, par exemple 2 pour coordonnées à deux points	
`<Nombre>` :	Paramètre du nombre de points utilisés pour le calcul (3 ou 4)		
`<Résultat>[3]` :	Variables du résultat : Coordonnées du centre du cercle et rayon		
	0	Coordonnées du centre du cercle : abscisse	
	1	Coordonnées du centre du cercle : ordonnée	
	2	Rayon	

Remarque

Les variables doivent avoir été définies avant leur utilisation.

Exemple

On veut savoir si trois points sont situés sur un arc de cercle.



Code de programme	**Commentaire**
`N10 DEF REAL PT[3,2]=(20,50,50,40,65,20)`	`; Variable pour l'indication des points du cercle.`
`N20 DEF REAL RES[3]`	`; Variable pour le résultat.`
`N30 DEF BOOL ETAT`	`; Variable pour l'état.`
`N40 ETAT=CALCDAT(POINT,3,RES)`	`; Appel des données de cercle calculées.`
`N50 IF ETAT == FALSE GOTOF ERROR`	`; Saut vers l'erreur.`

19.7 Désactivation de la préparation du contour (EXECUTE)

La fonction `EXECUTE` permet d'arrêter la préparation du contour et de retourner au mode d'exécution normal.

Syntaxe

```
EXECUTE(<ERREUR>)
```

Signification

`EXECUTE` :	Procédure prédéfinie pour quitter la préparation du contour	
`<ERREUR>` :	Variable pour la signalisation d'erreur	
	Type :	INT
	La valeur de la variable indique si le contour a pu être préparé correctement :	
	0	Erreur
	1	Aucune erreur

Exemple

Code de programme

```
...
N30 CONTPRON(…)
N40 G1 X… Z…
...
N100 EXECUTE(…)
...
```

Programmation externe de cycles 20

20.1 Cycles technologiques

20.1.1 Introduction

Contenu

Le présent chapitre contient la description des cycles pour les technologies de tournage, fraisage et rectification.

Structure

La description d'un cycle est structurée comme suit :

- Syntaxe
  Nom du cycle et séquence d'appel des paramètres de transfert
- Paramètres
  Tableau explicatif des différents paramètres

Description des paramètres

Les données suivantes sont indiquées dans le tableau pour un paramètre : nom, description, plage de valeurs et dépendances par rapport à d'autres paramètres.

La colonne qui fait référence au paramètre du masque est destinée à faciliter la recherche de valeurs programmées sur la commande lors de la décompilation d'appels de cycle générés en externe.

Paramètres "réservés à l'interface"

Dans le tableau des paramètres sont identifiés par "réservé à l'interface". Ils sont sans signification pour le fonctionnement du cycle. Ils sont uniquement nécessaires pour pouvoir décompiler entièrement des appels de cycle. S'ils ne sont pas programmés, le cycle peut tout de même être décompilé. Les champs sont repérés par une couleur particulière et doivent être renseignés dans le masque.

Paramètre "réservé"

Les paramètres décrits comme "réservés" doivent être programmés avec la valeur 0 ou avec un caractère d'omission pour que l'affectation des paramètres d'appel suivants corresponde aux paramètres de cycle internes. Exception : pour les paramètres de type String, valeur "" ou caractère d'omission.

Répétition de cycles dans les modèles de position

Les cycles de perçage et de fraisage peuvent être répétés dans les modèles de position (appels modaux). Dans ce cas, `MCALL` doit être écrit avant le cycle sur la même ligne, p. ex. `MCALL CYCLE83(...)`.

Remarque

Si certains paramètres de transfert (p. ex. `<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>`) sont indirectement programmés en tant que paramètres, le masque de saisie est ouvert lors de la décompilation mais il ne peut pas être enregistré, car il n'existe aucune affectation univoque pour certains champs de sélection.

20.1.2 Vue d'ensemble spécifique à la technologie

Le tableau synoptique suivant présente une liste de tous les cycles technologiques programmables en externe disponibles, avec la technologie à laquelle ils sont affectés :

Technologie	Cycle technologique
Perçage	• CYCLE81 - Perçage, centrage (Page 782) • CYCLE82 - Perçage, lamage (Page 783) • CYCLE85 - Alésage à l'alésoir (Page 791) • CYCLE86 - Alésage au tour (Page 792) • CYCLE83 - Perçage profond (Page 785) • CYCLE830 - Perçage profond 2 (Page 816) • CYCLE84 - Taraudage sans porte-taraud compensateur (Page 788) • CYCLE840 - Taraudage avec porte-taraud compensateur (Page 826) • CYCLE78 - Fraisage de filetages (Page 777) • CYCLE802 - Positions quelconques (Page 813) • HOLES1 - Rangée de trous (Page 742) • CYCLE801 - Réseau ou cadre (Page 812) • HOLES2 - Cercle de trous (Page 742)
Tournage	• CYCLE951 - Chariotage (Page 837) • CYCLE930 - Gorge (Page 831) • CYCLE940 - Formes de dégagement (Page 834) • CYCLE99 - Filetage au tour (Page 801) • CYCLE98 - Concaténation de filetages (Page 797) • CYCLE92 - Tronçonnage (Page 794)
Tournage de contours	• CYCLE62 - Appel de contour (Page 761) • CYCLE952 - Plongée de contour (Page 840)

Technologie	Cycle technologique
Fraisage	• CYCLE61 - Surfaçage (Page 759) • POCKET3 - Fraisage d'une poche rectangulaire (Page 744) • POCKET4 - Fraisage d'une poche circulaire (Page 747) • CYCLE76 - Fraisage d'un tourillon rectangulaire (Page 773) • CYCLE77 - Fraisage d'un tourillon circulaire (Page 775) • CYCLE79 - Polyèdre (Page 780) • SLOT1 - Rainure rectiligne (Page 750) • SLOT2 - Rainure sur cercle (Page 752) • CYCLE899 - Fraisage de rainure borgne (Page 828) • LONGHOLE - Trou oblong (Page 755) • CYCLE70 - Fraisage de filetage (Page 767) • CYCLE60 - Cycle de gravure (Page 757)
Fraisage de contours	• CYCLE62 - Appel de contour (Page 761) • CYCLE72 - Fraisage en contournage (Page 768) • CYCLE63 - Fraisage de poche de contour (Page 762) • CYCLE64 - Perçage ébauche d'une poche de contour (Page 765)
Rectification	• CYCLE495 - Profilage (Page 806) • CYCLE435 - Définir le système de coordonnées du dresseur (Page 806) • CYCLE4071 - Rectification longitudinale avec pénétration au point d'inversion (Page 846) • CYCLE4072 - Rectification longitudinale avec pénétration au point d'inversion et signal d'abandon (Page 847) • CYCLE4073 - Rectification longitudinale avec pénétration continue (Page 851) • CYCLE4074 - Rectification longitudinale avec pénétration continue et signal d'abandon (Page 852) • CYCLE4075 - Rectification plane avec pénétration au point d'inversion (Page 855) • CYCLE4077 - Rectification plane avec pénétration au point d'inversion et signal d'abandon (Page 858) • CYCLE4078 - Rectification plane avec pénétration continue (Page 861) • CYCLE4079 - Rectification plane avec pénétration intermittente (Page 863)
Divers	• CYCLE800 - Pivotement (Page 808) • CYCLE832 - Réglages High Speed (Page 822)
Tous	• GROUP_BEGIN - Début du bloc de programme (Page 865) • GROUP_END - Fin du bloc de programme (Page 866) • GROUP_ADDEND - Fin du supplément de mise au point (Page 866)

20.1.3 HOLES1 - Rangée de trous

Syntaxe

```
HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	X0	<SPCA>	REAL	Point de référence pour la rangée de trous du 1er axe (abs)		
2	Y0	<SPCO>	REAL	Point de référence pour la rangée de trous du 2ème axe (abs)		
3	α0	<STA1>	REAL	Angle de rotation de base (par rapport au 1er axe)		
4	L0	<FDIS>	REAL	Distance du 1er trou au point de référence		
5	L	<DBH>	REAL	Distance entre les trous		
6	N	<NUM>	INT	Nombre de trous		
7		<_VARI>	INT	Réservé		
8		<_UMODE>	INT	Réservé		
9		<_HIDE>	STRING [200]	Positions masquées • 198 caractères max. • Indication du numéro de position consécutif, par exemple "1,3" (les positions 1 et 3 ne sont pas exécutées)		
10		<_NSP>	INT	Réservé		
11		<_DMODE>	INT	Mode affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)

20.1.4 HOLES2 - Cercle de trous

Syntaxe

```
HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification			
1	X0	<CPA>	REAL	Centre du cercle de trous du 1er axe (abs) Point de référence dans le 1er axe			(pour XY) (pour XA, YB, ZC)
2	Y0	<CPO>	REAL	Centre du cercle de trous du 2e axe (abs) Point de référence dans le 2e axe			(pour XY) (pour XA, YB, ZC)
3	R	<RAD>	REAL	Rayon du cercle de trous			(pour XY)
4	α0	<STA1>	REAL	Angle de départ ou 1ère position d'axe rotatif			(pour XY) (pour XA, YB, ZC)
5	α1	<INDA>	REAL	Incrément angulaire (seulement pour cercle partiel)			(pour XY, XA, YB, ZC)
					< 0 =	Sens horaire	
					> 0 =	Sens antihoraire	
6	N	<NUM>	INT	Nombre de positions			
7		<_VARI>	INT	Type d'usinage			
				UNITES :	Réservé		
				DIZAINES :	Type de positionnement		
					0 =	Accostage de la position - linéaire	
					1 =	Accostage de la position - sur trajectoire circulaire	
				CENTAINES :	Réservé		
				MILLIERS :	Modèle de cercle		
					0 =	Mode compatibilité, si INDA = 0 alors cercle complet, si INDA <> 0 alors cercle partiel	
					1 =	Cercle complet	
					2 =	Cercle partiel	
				DIZAINES DE MILLIERS :	Modèle de positions avec axe rotatif		
					0 =	XY (sans axe rotatif)	(pour XY)
					1 =	XA (axe X et axe rotatif autour de X)	(uniquement pour XA)
					2 =	YB (axe Y et axe rotatif autour de Y)	(uniquement pour YB)
					3 =	ZC (axe Z et axe rotatif autour de C)	(uniquement pour ZC)
				MILLIONS + CENTAINES DE MILLIERS :	Offset (pour plusieurs axes rotatifs autour du même axe, quand l'indice est trop élevé, alors le 1er axe)		
					00 =	1. Axe A, B ou C	
					01 =	2. Axe A, B ou C	
					...		
					10 =	20. Axe A, B ou C	
8		<_UMODE>	INT	Réservé			

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
9		<_HIDE>	STRING [200]	Réservé		
10		<_NSP>	INT	Réservé		
11		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)

20.1.5 POCKET3 - Fraisage d'une poche rectangulaire

Syntaxe

```
POCKET3(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_CRAD>,
<_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>,
<_CDIR>, <_VARI>, <_MIDA>, <_AP1>, <_AP2>, <_AD>, <_RAD1>, <_DP1>,
<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<_RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Profondeur de poche (abs/rel), voir <_AMODE>		
5	L	<_LENG>	REAL	Longueur de poche (rel, à saisir avec le signe)		
6	W	<_WID>	REAL	Largeur de poche (rel, à saisir avec le signe)		
7	R	<_CRAD>	REAL	Rayon d'angle de la poche		
8	X0	<_PA>	REAL	Point de référence, 1er axe (abs)		
9	YO	<_PO>	REAL	Point de référence, 2e axe (abs)		
10	α0	<_STA>	REAL	Angle de rotation, angle entre axe longitudinal (L) et 1er axe		
11	DZ	<_MID>	REAL	Profondeur de passe maximale		
12	UXY	<_FAL>	REAL	Surépaisseur de finition dans le plan		
13	UZ	<_FALD>	REAL	Surépaisseur de finition au fond		
14	F	<_FFP1>	REAL	Avance dans le plan		
15	FZ	<_FFD>	REAL	Avance de pénétration		
16		<_CDIR>	INT	Sens de fraisage :	0 =	En avalant
					1 =	En opposition

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
17		<_VARI>	INT	Type d'usinage		
				UNITES :		
					1 =	Ebauchage
					2 =	Finition
					4 =	Finition du bord
					5 =	Chanfreinage
				DIZAINES :		
					0 =	Avant-trou, passe avec G0
					1 =	Perpendiculaire, passe avec G1
					2 =	Hélicoïdal
					3 =	Avec oscillation dans l'axe longitudinal de la poche
				CENTAINES :	Réservé	
18	DXY	<_MIDA>	REAL	Profondeur de passe maximale dans le plan (unité, voir <_AMODE>)		
19	L1	<_AP1>	REAL	Longueur de l'ébauche (rel)		
20	W1	<_AP2>	REAL	Largeur de l'ébauche (rel)		
21	AZ	<_AD>	REAL	Profondeur de l'ébauchage (rel)		
22	ER	<_RAD1>	REAL	Rayon de la trajectoire hélicoïdale de pénétration hélicoïdale		
	EW			Angle de pénétration maximal pour oscillatoire		
23	EP	<_DP1>	REAL	Pas d'hélice pour pénétration hélicoïdale		
24		<_UMODE>	INT	Réservé		
25	FS	<_FS>	REAL	Largeur de chanfrein (rel)		
26	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs/rel), voir <_AMODE>		

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
27		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération / calcul du point de départ uniquement	
					0 =	Mode de compatibilité
					1 =	Usinage normal
				MILLIERS :	Cotation à partir du milieu/coin	
					0 =	Mode de compatibilité
					1 =	Cotation à partir du milieu
					2 =	Cotation à partir du point d'angle, position de poche +LENG/+WID
					3 =	Cotation à partir du point d'angle, position de poche -LENG/+WID
					4 =	Cotation à partir du point d'angle, position de poche +LENG/-WID
					5 =	Cotation à partir du point d'angle, position de poche -LENG/-WID
				DIZAINES DE MILLIERS :	Usinage complet / reprise	
					0 =	Mode de compatibilité (<_AP1>, <_AP2> et <_AD> à traiter comme précédemment)
					1 =	Usinage complet
					2 =	Reprise

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
28		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Mode d'avance : groupe G (G94/G95) pour l'avance de surface et l'avance de profondeur	
					0 =	Mode de compatibilité
					1 =	instruction G comme avant l'appel de cycle. G94/G95 identique pour avance de surface et avance de profondeur
				CENTAINES :	---	Réservé
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée
29		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de poche (Z1)	
					0 =	Absolu (mode de compatibilité)
					1 =	Relatif
				DIZAINES :	Unité pour la profondeur de passe dans le plan (DXY)	
					0 =	mm
					1 =	% du diamètre de l'outil
				CENTAINES :	Profondeur de pénétration lors du chanfreinage (ZFS)	
					0 =	Absolu
					1 =	Relatif

20.1.6 POCKET4 - Fraisage d'une poche circulaire

Syntaxe

```
POCKET4(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_CDIAM>, <_PA>, <_PO>,
<_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_MIDA>,
```

```
<_AP1>, <_AD>, <_RAD1>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<_RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Profondeur de poche (abs/rel), voir <_AMODE>		
5	∅	<_CDIAM>	REAL	Diamètre de poche ou rayon de poche, voir <_DMODE>		
6	X0	<_PA>	REAL	Point de référence 1er axe (abs)		
7	Y0	<_PO>	REAL	Point de référence 2e axe (abs)		
8	DZ	<_MID>	REAL	Profondeur de passe maximale, voir <_VARI> = dans le plan Pas d'hélice maximal, voir <_VARI> = hélicoïdal		
9	UXY	<_FAL>	REAL	Surépaisseur de finition dans le plan		
10	UZ	<_FALD>	REAL	Surépaisseur de finition au fond		
11	F	<_FFP1>	REAL	Avance pour l'usinage de la surface		
12	FZ	<_FFD>	REAL	Avance de pénétration		
13		<_CDIR>	INT	Sens de fraisage	0 =	En avalant
					1 =	En opposition
14		<_VARI>	INT	Type d'usinage		
				UNITES :	Usinage	
					1 =	Ebauchage
					2 =	Finition
					4 =	Finition du bord
					5 =	Chanfreinage
				DIZAINES :	Type de passe (ébauchage et finition)	
					0 =	Avant-trou, passe avec G0 (la poche est ébauchée)
					1 =	Perpendiculaire, passe avec G1
					2 =	Hélicoïdal
				CENTAINES :	Réservé	
				MILLIERS :		
					0 =	Dans le plan
					1 =	Hélicoïdal
15	DXY	<_MIDA>	REAL	Profondeur de passe maximale dans le plan, voir <_AMODE>, 0 = 0,8 x diamètre de l'outil		
16	∅	<_AP1>	REAL	Diamètre/rayon de l'ébauche (rel)		
17	AZ	<_AD>	REAL	Profondeur de l'ébauchage (rel)		
18	ER	<_RAD1>	REAL	Rayon de la trajectoire hélicoïdale de pénétration hélicoïdale		
19	EP	<_DP1>	REAL	Pas d'hélice pour pénétration sur trajectoire hélicoïdale		
20		<_UMODE>	INT	Réservé		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
21	FS	<_FS>	REAL	Largeur de chanfrein (rel)		
22	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs/rel), voir <_AMODE>		
23		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Usinage / calcul du point de départ	
					0 =	Mode de compatibilité
					1 =	Usinage normal
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Usinage complet / reprise	
					0 =	Mode de compatibilité (<_AP1> et <_AD> à traiter comme précédemment)
					1 =	Usinage complet
					2 =	Reprise
24		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Mode d'avance : groupe G (G94/G95) pour l'avance de surface et l'avance de profondeur	
					0 =	Mode de compatibilité
					1 =	instruction G comme avant l'appel de cycle. G94/G95 identique pour avance de surface et avance de profondeur
				CENTAINES :		
					0 =	Mode de compatibilité (<_CDIAM>/<_AP1> saisi en tant que rayon)
					1 =	<_CDIAM>/<_AP1> saisi en tant que diamètre
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
25		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de poche (Z1)	
					0 =	Absolu (mode de compatibilité)
					1 =	Relatif
				DIZAINES :	Unité pour la largeur de passe (DXY)	
					0 =	mm
					1 =	% du diamètre de l'outil
				CENTAINES :	Profondeur de pénétration lors du chanfreinage (ZFS)	
					0 =	Absolu
					1 =	Relatif

20.1.7 SLOT1 - Rainure rectiligne

Syntaxe

```
SLOT1 (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <WID>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>,
<CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FALD>, <_STA2>,
<_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

N°	Masque de paramètre	Paramètre interne	Type de données	Signification
1	RP	<RTP>	REAL	Plan de retrait (abs)
2	Z0	<RFP>	REAL	Point de référence de l'axe outil (abs)
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)
4	Z1	<_DP>	REAL	Profondeur de rainure (abs)
5		<_DPR>	REAL	Profondeur de rainure (rel), par rapport à Z0 (à saisir sans signe)
6		<NUM>	INT	Nombre de rainures = 1
7	L	<LENG>	REAL	Longueur de la rainure
8	W	<WID>	REAL	Largeur de la rainure
9	X0	<_CPA>	REAL	Point de référence, 1er axe du plan
10	Y0	<_CPO>	REAL	Point de référence, 2e axe du plan
11		<RAD>	REAL	Réservé
12	α	<STA1>	REAL	Angle de rotation
13		<INDA>	REAL	Réservé
14	FZ	<FFD>	REAL	Avance de pénétration
15	F	<FFP1>	REAL	Avance
16	DZ	<_MID>	REAL	Profondeur de passe maximale

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
17		<CDIR>	INT	Sens de fraisage	0 =	En avalant
					1 =	En opposition
18	UXY	<_FAL>	REAL	Surépaisseur de finition Plan ou Bord de rainure		
19		<VARI>	INT	Type d'usinage		
				UNITES :		
					0 =	Réservé
					1 =	Ebauchage
					2 =	Finition
					4 =	Finition du bord (usiner bord uniquement)
					5 =	Chanfreinage
				DIZAINES :	Accostage	
					0 =	Avant-trou, passe avec G0 (la rainure est ébauchée)
					1 =	Perpendiculaire, passe avec G1
					2 =	Hélicoïdal
					3 =	Pendulaire
				CENTAINES :	Réservé	
20	DZF	<_MIDF>	REAL	Réservé		
21	FF	<FFP2>	REAL	Réservé		
22	SF	<SSF>	REAL	Réservé		
23	UZ	<_FALD>	REAL	Surépaisseur de finition au fond		
24	ER	<_STA2>	REAL	Rayon de la trajectoire hélicoïdale de pénétration hélicoïdale		
	EW			Angle de pénétration maximal pour oscillatoire		
25	EP	<_DP1>	REAL	Profondeur de pénétration par tour pour l'hélice		
26		<_UMODE>	INT	Réservé		
27	FS	<_FS>	REAL	Largeur de chanfrein (rel) pour le chanfreinage		
28	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs/rel), voir <_AMODE>		
29		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération ou calcul du point de départ uniquement	
					1 =	Usinage normal
				MILLIERS :	Cotation à partir du point de référence, de la position de la rainure	
					0 =	Centre
					1 =	Gauche intérieur +L
					2 =	Droite intérieur -L
					3 =	Bord gauche +L
					4 =	Bord droit -L

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
30		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Identificateur de version de logiciel	
					1 =	Extension fonctionnelle SLOT1
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée
31		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale Z1 (abs/rel)	
					0 =	Compatibilité
					1 =	Z1 (rel)
					2 =	Z1 (abs)
				DIZAINES :	Réservé	
				CENTAINES :	Profondeur de pénétration lors du chanfreinage ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (rel)

Remarque

Le cycle comporte de nouvelles fonctions par rapport aux précédentes versions du logiciel. Ceci a pour conséquence que certains paramètres ne sont plus affichés dans le masque de saisie (<NUM>, <RAD>, <INDA>). Plusieurs rainures peuvent être programmées sur un modèle de position au moyen de "MCALL" et de l'appel du modèle de position souhaité, p. ex. HOLES2.

20.1.8 SLOT2 - Rainure sur cercle

Syntaxe

```
SLOT2(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <AFSL>, <WID>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>,
```

```
<CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FFCP>, <_UMODE>,
<_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<RTP>	REAL	Plan de retrait (abs)		
2	Z0	<RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Profondeur de rainure (abs)		
5		<_DPR>	REAL	Profondeur de rainure (rel), par rapport à Z0 (à saisir sans signe)		
6	N	<NUM>	INT	Nombre de rainures		
7	α1	<AFSL>	REAL	Angle au centre de la rainure		
8	W	<WID>	REAL	Largeur de la rainure		
9	X0	<_CPA>	REAL	Point de référence = centre du cercle, 1er axe du plan		
10	Y0	<_CPO>	REAL	Point de référence = centre du cercle, 2e axe du plan		
11	R	<RAD>	REAL	Rayon du cercle		
12	α0	<STA1>	REAL	Angle de départ		
13	α2	<INDA>	REAL	Incrément angulaire		
14	FZ	<FFD>	REAL	Avance de pénétration		
15	F	<FFP1>	REAL	Avance		
16	DZ	<_MID>	REAL	Profondeur de passe maximale		
17		<CDIR>	INT	Sens de fraisage	0 =	En avalant
					1 =	En opposition
18	UXY	<_FAL>	REAL	Surépaisseur de finition dans le plan ou bord de rainure		
19		<VARI>	INT	Type d'usinage		
				UNITES :		
					0 =	Usinage complet
					1 =	Ebauchage
					2 =	Finition
					3 =	Finition du bord
					5 =	Chanfreinage
				DIZAINES :		
					0 =	Positionnement intermédiaire avec droite G0
					1 =	Positionnement intermédiaire sur la trajectoire circulaire
				CENTAINES :	Réservé	
				MILLIERS :		
					0 =	Mode compatibilité, si <INDA> = 0 alors cercle complet, si <INDA> <> 0 alors cercle partiel)
					1 =	Cercle complet
					2 =	Cercle partiel

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
20	DZF	<_MIDF>	REAL	Réservé		
21		<FFP2>	REAL	Réservé		
22		<SSF>	REAL	Réservé		
23	FF	<_FFCP>	REAL	Réservé		
24		<_UMODE>	INT	Réservé		
25	FS	<_FS>	REAL	Largeur de chanfrein (rel)		
26	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs/rel), voir <_AMODE>		
27		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération ou calcul du point de départ uniquement	
					0 =	Mode de compatibilité
					1 =	Usinage normal
28		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Identificateur de version de logiciel	
					1 =	Fonctions SLOT2 à partir de la version 2.5
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée
29		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale Z1 (abs/rel)	
					0 =	Compatibilité
					1 =	Z1 (rel)
					2 =	Z1 (abs)
				DIZAINES :	Réservé	
				CENTAINES :	Profondeur de pénétration lors du chanfreinage ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (rel)

20.1.9 LONGHOLE - Trou oblong

Syntaxe

```
LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>,
<_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>,
<_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Profondeur de trou oblong (abs)		
5		<_DPR>	REAL	Profondeur de trou oblong (rel), par rapport à Z0 (à saisir sans signe)		
6		<NUM>	INT	Nombre de trous oblongs = 1		
7	L	<LENG>	REAL	Longueur du trou oblong		
8	X0	<_CPA>	REAL	Point de référence, 1er axe du plan		
9	Y0	<_CPO>	REAL	Point de référence, 2e axe du plan		
10		<RAD>	REAL	Réservé		
11	α0	<STA1>	REAL	Angle de rotation		
12		<INDA>	REAL	Réservé		
13	FZ	<FFD>	REAL	Avance de pénétration		
14	F	<FFP1>	REAL	Avance		
15	DZ	<MID>	REAL	Profondeur de passe maximale		
16		<_VARI>	INT	Type d'usinage		
				UNITES :	Mode de pénétration	
					1 =	Perpendiculairement avec G1
					3 =	Pendulaire
				CENTAINES :	Réservé	
17		<_UMODE>	INT	Réservé		

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
18		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération ou calcul du point de départ uniquement	
					0 =	Mode compatibilité
					1 =	Usinage normal
				MILLIERS :	Cotation à partir du point de référence, de la position de la rainure	
					0 =	Centre
					1 =	Gauche intérieur +L
					2 =	Droite intérieur -L
					3 =	Bord gauche +L
					4 =	Bord droit -L
19		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Mode d'avance : groupe G (G94/G95) pour l'avance de surface et l'avance de profondeur	
					0 =	Mode compatibilité
					1 =	instruction G comme avant l'appel de cycle. G94/G95 identique pour avance de surface et avance de profondeur
				CENTAINES :	Réservé	
				MILLIERS :	Identificateur de version de logiciel	
					1 =	Extension fonctionnelle LONGHOLE (cotation à partir du point de référence)
20		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale Z1 (abs/rel)	
					0 =	Compatibilité
					1 =	Z1 (rel)
					2 =	Z1 (abs)

Remarque

Le cycle comporte de nouvelles fonctions par rapport aux précédentes versions du logiciel. Ceci a pour conséquence que certains paramètres ne sont plus affichés dans le masque de saisie (<NUM>, <RAD>, <INDA>). Plusieurs rainures peuvent être programmées sur un modèle de position au moyen de "MCALL" et de l'appel du modèle de position souhaité, p. ex. HOLES2.

20.1.10 CYCLE60 - Cycle de gravure

Syntaxe

```
CYCLE60(<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
1		<_TEXT>	STRING [200]	Texte à graver (100 caractères max.)
2	RP	<_RTP>	REAL	Plan de retrait (abs)
3	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)
4	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au plan de référence, à saisir sans signe)
5	Z1	<_DP>	REAL	Profondeur (abs), voir <_AMODE>
6	Z1	<_DPR>	REAL	Profondeur (rel), voir <_AMODE>
7	X0	<_PA>	REAL	Point de référence, 1er axe du plan (abs) - cartésien, voir <_VARI>
	R			Point de référence, longueur (rayon) - polaire, voir <_VARI>
8	Y0	<_PO>	REAL	Point de référence, 2e axe du plan (abs) - cartésien, voir <_VARI>
	α0			Point de référence, angle par rapport au 1er axe - polaire, voir <_VARI>
9	α1	<_STA>	REAL	Orientation du texte, angle de la ligne de texte par rapport au 1er axe, voir <_VARI>
10	XM	<_CP1>	REAL	Centre du cercle de texte, 1er axe du plan (abs) - cartésien, voir <_VARI>
	LM			Centre du cercle de texte, longueur (rayon) rapportée à WNP - polaire, voir <_VARI>
11	YM	<_CP2>	REAL	Centre du cercle de texte, 2e axe du plan (abs) - cartésien, voir <_VARI>
	αM			Centre du cercle de texte, angle rapporté au 1er axe - polaire, voir <_VARI>
12	W	<_WID>	REAL	Hauteur des caractères (à saisir sans signe)

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
13	DX1 DX2	<_DF>	REAL	Espacement des caractères / largeur totale, voir <_VARI>		
	α2			Angle au centre, voir <_VARI>		
14	FZ	<_FFD>	REAL	Avance de pénétration, voir <_DMODE>		
15	F	<_FFP1>	REAL	Avance pour l'usinage de la surface		
16		<_VARI>	INT	Usinage (orientation et point de référence du texte à graver)		
				UNITES :	Point de référence	
					0 =	cartésien
					1 =	Polaire
				DIZAINES :	Orientation du texte	
					0 =	texte sur une ligne
					1 =	texte sur un arc de cercle en haut
					2 =	texte sur un arc de cercle en bas
				CENTAINES :	Réservé	
				MILLIERS :	Point de référence horizontal du texte	
					0 =	Gauche
					1 =	au centre
					2 =	Droite
				DIZAINES DE MILLIERS :	Point de référence vertical du texte	
					0 =	en bas
					1 =	au centre
					2 =	en haut
				CENTAINES DE MILLIERS :	Longueur du texte	
					0 =	espacement des caractères
					1 =	largeur totale du texte (uniquement pour texte linéaire)
					2 =	angle d'attaque (disposition du texte en arc de cercle uniquement)
				MILLIONS :	Centre du cercle	
					0 =	Cartésien
					1 =	Polaire
				DIZAINES DE MILLIONS :	Ecriture renversée	
					0 =	Compatibilité
					1 =	Ecriture renversée activée
					2 =	Ecriture renversée désactivée
17		<_CODEP>	INT	Numéro de la page de code pour la police (actuellement uniquement 1252)		
18		<_UMODE>	INT	Réservé		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
19		_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération / calcul du point de départ uniquement	
					0 =	Mode de compatibilité
					1 =	Usinage normal
20		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Mode d'avance : groupe G (G94/G95) pour l'avance de surface et l'avance de profondeur	
					0 =	Mode de compatibilité
					1 =	instruction G comme avant l'appel de cycle. G94/G95 identique pour avance de surface et avance de profondeur
21		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale (<_DP>, <_DPR>)	
					0 =	Compatibilité
					1 =	Relatif (<_DPR>)
					2 =	Absolu (<_DP>)

20.1.11 CYCLE61 - Surfaçage

Syntaxe

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<_RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil, hauteur pièce brute (abs)		
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Hauteur pièce finie (abs/rel), voir <_AMODE>		
5	X0	<_PA>	REAL	Point d'angle 1 sur le 1er axe (abs)		
6	Y0	<_PO>	REAL	Point d'angle 1 sur le 2e axe (abs)		
7	X1	<_LENG>	REAL	Point d'angle 2 sur le 1er axe (abs/rel), voir <_AMODE>		
8	Y1	<_WID>	REAL	Point d'angle 2 sur le 2e axe (abs/rel), voir <_AMODE>		
9	DZ	<_MID>	REAL	Profondeur de passe maximale		
10	DXY	<_MIDA>	REAL	Profondeur de passe maximale dans le plan (unité, voir <_AMODE>)		
11	UZ	<_FALD>	REAL	Surépaisseur de finition au fond		
12	F	<_FFP1>	REAL	Avance d'usinage		
13		<_VARI>	INT	Type d'usinage		
				UNITES :	Usinage	
					1 =	Ebauchage
					2 =	Finition
				DIZAINES :	Sens d'usinage	
					1 =	Parallèle au 1er axe, un seul sens
					2 =	Parallèle au 2e axe, un seul sens
					3 =	Parallèle au 1er axe, sens alternés
					4 =	Parallèle au 2e axe, sens alternés
14		<_LIM>	INT	Limites		
				UNITES :	Limite 1er axe moins	
					0 =	Non
					1 =	Oui
				DIZAINES :	Limite 1er axe plus	
					0 =	Non
					1 =	Oui
				CENTAINES :	Limite 2e axe moins	
					0 =	Non
					1 =	Oui
				MILLIERS :	Limite 2e axe plus	
					0 =	Non
					1 =	Oui

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
15		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
16		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale (<_DP>)	
					0 =	Absolu
					1 =	Relatif
				DIZAINES :	Unité pour la profondeur de passe dans le plan (<_MIDA>)	
					0 =	mm
					1 =	% du diamètre de l'outil
				CENTAINES :	Réservé	
				MILLIERS :	Longueur de la surface	
					0 =	Relatif
					1 =	Absolu
				DIZAINES DE MILLIERS :	Largeur de la surface	
					0 =	Relatif
					1 =	Absolu

20.1.12 CYCLE62 - Appel de contour

Syntaxe

```
CYCLE62(<_KNAME>,<_TYPE>, <_LAB1>, <_LAB2>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	PRG/CON	<_KNAME>	STRING [140]	Le nom du contour ou le nom du sous-programme ne doit pas être programmé lorsque _TYPE = 2		
2		<_TYPE>	INT	Détermination de la saisie du contour		
					0 =	Sous-programme
					1 =	Nom du contour
					2 =	Etiquettes
					3 =	Etiquettes dans le sous-programme
3	LAB1	<_LAB1>	STRING[32]	Etiquette 1, début du contour		
4	LAB2	<_LAB2>	STRING[32]	Etiquette 2, fin du contour		

20.1.13 CYCLE63 - Fraisage de poche de contour

Syntaxe

```
CYCLE63(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_FZ>,
<_DXY>, <_DZ>, <_UXY>, <_UZ>, <_CDIR>, <_XS>, <_YS>, <_ER>, <_EP>,
<_EW>, <_FS>, <_ZFS>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	PRG	<_PRG>	STRING [100]	Nom du programme d'évidement		
2		<_VARI>	INT	Type d'usinage		
				UNITES :	Usinage technologique	
					1 =	Ebauchage
					3 =	Finition du fond
					4 =	Finition du bord
					5 =	Chanfreinage
				DIZAINES :	Type de passe	
					0 =	Plongée centrée
					1 =	Plongée hélicoïdale
					2 =	Plongée pendulaire
				CENTAINES :	Réservé	
				MILLIERS :	Mode de relèvement	
					0 =	Relèvement vers le plan de retrait
					1 =	Relèvement vers le point de référence + distance de sécurité
				DIZAINES DE MILLIERS :	Point de départ pour l'ébauchage et la finition du fond	
					0 =	Auto
					1 =	Manuel
3	RP	<_RP>	REAL	Plan de retrait (abs)		
4	Z0	<_Z0>	REAL	Point de référence de l'axe outil (abs)		
5	SC	<_SC>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
6	Z1	<_Z1>	REAL	Profondeur finale, (voir <_AMODE> UNITES)		
7	F	<_F>	REAL	Avance dans le plan ébauchage/finition		
8	FZ	<_FZ>	REAL	Avance de pénétration		
9	DXY	<_DXY>	REAL	Prise de passe dans le plan - unité (voir <_AMODE> DIZAINES)		
10	DZ	<_DZ>	REAL	Pénétration Profondeur		
11	UXY	<_UXY>	REAL	Surépaisseur de finition dans le plan		
12	UZ	<_UZ>	REAL	Surépaisseur de finition au fond		
13		<_CDIR>	INT	Sens de fraisage	0 =	En avalant
					1 =	En opposition
14	XS	<_XS>	REAL	Point de départ X, absolu		
15	YS	<_YS>	REAL	Point de départ Y, absolu		
16	ER	<_ER>	REAL	Pénétration hélicoïdale : rayon		
17	EP	<_EP>	REAL	Pénétration hélicoïdale : pas		
18	EW	<_EW>	REAL	Pénétration oscillatoire : angle de pénétration maximal		
19	FS	<_FS>	REAL	Largeur de chanfrein (rel) pour le chanfreinage		
20	ZFS	<_ZFS>	REAL	Profondeur de pénétration de la pointe d'outil lors du chanfreinage (voir <_AMODE> CENTAINES)		

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
21	TR	<_TR>	STRING[32]	Nom de l'outil de référence pour l'enlèvement de matière restante		
22	DR	<_DR>	INT	Outil de référence, numéro D, pour enlèvement de matière restante		
23		<_UMODE>	INT	Réservé		
24		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération / calcul du point de départ uniquement	
					0 =	Usinage normal (aucun mode de compatibilité n'est requis)
					1 =	Usinage normal
					2 =	Réservé
25		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Réservé	
				CENTAINES :	Mode technologie	
					1 =	Poche
					2 =	Tourillon
				MILLIERS :	Semi-finition	
					0 =	Non
					1 =	Oui
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
26		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale (Z1)	
					0 =	Absolu (mode de compatibilité)
					1 =	Relatif
				DIZAINES :	Unité pour la profondeur de passe dans le plan (DXY)	
					0 =	mm
					1 =	% du diamètre de l'outil
				CENTAINES :	Profondeur de pénétration lors du chanfreinage (ZFS)	
					0 =	Absolu
					1 =	Relatif
				MILLIERS :	---	Réservé

20.1.14 CYCLE64 - Perçage ébauche d'une poche de contour

Syntaxe

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_DXY>,
<_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	PRG	<_PRG>	STRING [100]	Nom du programme de perçage/centrage		
2		<_VARI>	INT	Type d'usinage		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Mode de relèvement	
					0 =	Relèvement vers le plan de retrait
					1 =	Relèvement vers le point de référence + distance de sécurité
3	RP	<_RP>	REAL	Plan de retrait (abs)		
4	Z0	<_Z0>	REAL	Point de référence (abs)		
5	SC	<_SC>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
6	Z1	<_Z1>	REAL	Profondeur de perçage/centrage, (voir <_AMODE> UNITES)		
7	F	<_F>	REAL	Avance Perçage/Centrage		
8	DXY	<_DXY>	REAL	Prise de passe dans le plan - unité (voir <_AMODE> DIZAINES)		

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
9	UXY	<_UXY>	REAL	Surépaisseur de finition dans le plan		
10	UZ	<_UZ>	REAL	Surépaisseur de finition au fond		
11		<_CDIR>	INT	Sens de fraisage	0 =	En avalant
					1 =	En opposition
12	TR	<_TR>	STRING[20]	Nom de l'outil de référence		
13	DR	<_DR>	INT	Outil de référence, numéro D		
14		<_UMODE>	INT	Réservé		
15		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération / calcul du point de départ uniquement	
					0 =	Usinage normal (aucun mode de compatibilité n'est requis)
					1 =	Usinage normal
					2 =	Réservé
25		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Mode technologie	
					1 =	Perçage ébauche
					2 =	Centrage
26		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de perçage/centrage Z1	
					0 =	Absolu (mode de compatibilité)
					1 =	Relatif
				DIZAINES :	Unité pour la profondeur de passe dans le plan (DXY)	
					0 =	mm
					1 =	% du diamètre de l'outil

20.1.15 CYCLE70 - Fraisage de filetage

Syntaxe

```
CYCLE70(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<_RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Longueur de filet (abs/rel), voir <_AMODE> Tenir compte du dégagement au fond du perçage (un demi pas minimum)		
5	∅	<_DIATH>	REAL	Diamètre nominal du filetage		
6	H1	<_H1>	REAL	Profondeur de filet		
7	U	<_FAL>	REAL	Surépaisseur de finition		
8	P	<_PIT>	REAL	Pas de vis (sélection <_PITA> : mm, inch, MODULE, filets/inch)		
9	NT	<_NT>	INT	Nombre de dents sur la plaque de découpe Longueur d'outil toujours rapportée à la dent inférieure.		
10	DXY	<_MID>	REAL	Pénétration maximale par passe <_MID> > <_H1> : tout en une seule passe		
11	F	<_FFR>	REAL	Avance de fraisage		
12		<_TYPTH>	INT	Type de filetage	0 =	Filetage intérieur
					1 =	Filetage extérieur
13	X0	<_PA>	REAL	Centre du cercle, 1er axe (abs)		
14	Y0	<_PO>	REAL	Centre du cercle, 2e axe (abs)		
15	αS	<_NSP>	REAL	Angle d'attaque (filetage multifilet)		
16		<_VARI>	INT	Type d'usinage		
				UNITES :		
					1 =	Ebauchage
					2 =	Finition
				DIZAINES :		
					1 =	De haut en bas
					2 =	De bas en haut
				CENTAINES :		
					0 =	Filetage à droite
					1 =	Filetage à gauche

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
17		<_PITA>	INT	Evaluation du pas de vis		
					0 =	Mode de compatibilité
					1 =	Pas en mm
					2 =	Pas en filets par inch (TPI)
					3 =	Pas en inch
					4 =	Pas en tant que MODULE
18		<_PITM>	STRING[15]	Chaîne de caractères servant de mémento pour la saisie du pas de vis (réservé à l'interface)		
19		<_PTAB>	STRING[20]	Chaîne de caractères pour table de filetages ("", "ISO", "BSW", "BSP", "UNC") (réservé à l'interface)		
20		<_PTABA>	STRING[20]	Chaîne de caractères pour la sélection dans la table de filetages (p. ex. "M 10", "M 12"...) (réservé à l'interface)		
21		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Usinage / calcul du point de départ	
					0 =	Mode de compatibilité
					1 =	Usinage normal
22		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
23		<_AMODE>	INT	Mode alternatif		
				UNITES :	Longueur de filet (<_DP>)	
					0 =	Absolu
					1 =	Relatif

20.1.16 CYCLE72 - Fraisage en contournage

Syntaxe

```
CYCLE72(<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>,<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
1		<_KNAME>	STRING [141]	Nom du sous-programme de contour
2	RP	<_RTP>	REAL	Plan de retrait (abs)
3	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)
4	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)
5	Z1	<_DP>	REAL	Point final, profondeur finale (abs/rel), voir <_AMODE>
6	DZ	<_MID>	REAL	Profondeur de passe maximale (rel, à saisir sans signe)
7	UXY	<_FAL>	REAL	Surépaisseur de finition dans le plan (rel), surépaisseur sur le contour du bord
8	UZ	<_FALD>	REAL	Surépaisseur de finition au fond (rel), surépaisseur au fond (à saisir sans signe)
9	FX	<_FFP1>	REAL	Avance sur contour
10	FZ	<_FFD>	REAL	Avance pour profondeur de passe (ou pénétration spatiale)

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
11		<_VARI>	INT	Type d'usinage		
				UNITES :	Usinage	
					1 =	Ebauchage
					2 =	Finition
					5 =	Chanfreinage
				DIZAINES :		
					0 =	Trajets intermédiaires avec G0
					1 =	Trajets intermédiaires avec G1
				CENTAINES :	Retrait en fin de contour	
					0 =	Retrait en fin de contour jusqu'au point de référence
					1 =	Retrait en fin de contour jusqu'au point de référence + <_SDIS>
					2 =	Retrait en fin de contour de <_SDIS>
					3 =	Pas de retrait en fin de contour, le point de départ suivant est accosté avec l'avance sur contour
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Usiner le contour	
					0 =	Usiner le contour en marche avant
					1 =	Usiner le contour en marche arrière Restrictions pour la marche arrière : • 170 éléments de contour max. (y compris chanfreins ou arrondis) • Seules les valeurs du plan (X/Y) et F sont évaluées
12		<_RL>	INT	Sens d'usinage		
					40 =	Au centre du contour (G40, accostage et retrait : droit ou perpendiculaire)
					41 =	A gauche du contour (G41, accostage et retrait : droit ou circulaire)
					42 =	A droite du contour (G42, accostage et retrait : droit ou circulaire)

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
13		<_AS1>	INT	Mouvement d'accostage du contour		
				UNITES :		
					1 =	Droite
					2 =	Quart de cercle
					3 =	Demi-cercle
					4 =	Accostage et retrait perpendiculaires
				DIZAINES :		
					0 =	Dernier mouvement, dans le plan
					1 =	Dernier mouvement, spatial
14	L1	<_LP1>	REAL	Distance ou rayon d'accostage (rel, à saisir sans signe)		
15	FZ	<_FF3>	REAL	Avance pour trajets intermédiaires (G94/G95 comme sur contour)		
16		<_AS2>	INT	Mouvement de retrait du contour (sauf pour accostage/retrait perpendiculaires)		
				UNITES :		
					1 =	Droite
					2 =	Quart de cercle
					3 =	Demi-cercle
				DIZAINES :		
					0 =	Dernier mouvement, dans le plan
					1 =	Dernier mouvement, spatial
17	L2	<_LP2>	REAL	Distance ou rayon de retrait (rel, à saisir sans signe)		
18		<_UMODE>	INT	Réservé		
19	FS	<_FS>	REAL	Largeur de chanfrein (rel)		
20	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs/rel), voir <_AMODE>		
21		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération / calcul du point de départ uniquement	
					0 =	Mode de compatibilité
					1 =	Usinage normal

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
22		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Mode d'avance : groupe G (G94/G95) pour l'avance de surface et l'avance de profondeur	
					0 =	Mode de compatibilité
					1 =	instruction G comme avant l'appel de cycle. G94/G95 identique pour avance de surface et avance de profondeur
				MILLIERS :		
					0 =	Mode compatibilité : le nom du contour figure dans <_KNAME>
					1 =	Le nom du contour est programmé dans le CYCLE62 et transféré dans _SC_CONT_NAME
23		<_AMODE>	INT	Mode alternatif		
				UNITES :	Point final Z1 (<_DP>)	
					0 =	Absolu (mode de compatibilité)
					1 =	Relatif
				DIZAINES :	Unité pour la profondeur de passe dans le plan	
					0 =	mm, inch
					1 =	Réservé
				CENTAINES :	Profondeur de pénétration lors du chanfreinage (ZFS) (<_ZFS>)	
					0 =	Absolu
					1 =	Relatif

Remarque

Si les paramètres de transfert ci-après sont programmés indirectement (en tant que paramètres), le masque de saisie n'est pas décompilé :

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

20.1.17 CYCLE76 - Fraisage d'un tourillon rectangulaire

Syntaxe

```
CYCLE76(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_LENG>, <_WID>,
<_CRAD>, <_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>,
<_FFD>, <_CDIR>, <_VARI>, <_AP1>, <_AP2>, <_FS>, <_ZFS>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<_RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Profondeur de tourillon (abs)		
5		<_DPR>	REAL	Profondeur de tourillon (rel), par rapport à Z0 (à saisir sans signe)		
6	L	<_LENG>	REAL	Longueur de tourillon, voir <_GMODE> (à saisir sans signe)		
7	W	<_WID>	REAL	Largeur de tourillon, voir <_GMODE> (à saisir sans signe)		
8	R	<_CRAD>	REAL	Rayon d'angle du tourillon (à saisir sans signe)		
9	X0	<_PA>	REAL	Point de référence du tourillon par rapport au 1er axe du plan (abs)		
10	Y0	<_PO>	REAL	Point de référence du tourillon par rapport au 2e axe du plan (abs)		
11	α0	<_STA>	REAL	Angle de rotation, angle entre axe longitudinal (L) et 1er axe du plan		
12	DZ	<_MID>	REAL	Profondeur de passe maximale (rel, à saisir sans signe)		
13	UXY	<_FAL>	REAL	Surépaisseur de finition dans le plan (rel), surépaisseur sur le contour du bord		
14	UZ	<_FALD>	REAL	Surépaisseur de finition au fond (rel), surépaisseur au fond (à saisir sans signe)		
15	FX	<_FFP1>	REAL	Avance sur contour		
16	FZ	<_FFD>	REAL	Avance de pénétration		
17		<_CDIR>	INT	Sens de fraisage (à saisir sans signe)		
				UNITES :		
					0 =	En avalant
					1 =	En opposition
18		<_VARI>	INT	Usinage		
				UNITES :		
					1 =	Ebauchage
					2 =	Finition
					5 =	Chanfreinage
19	L1	<_AP1>	REAL	Longueur du tourillon brut		
20	W1	<_AP2>	REAL	Largeur du tourillon brut		
21	FS	<_FS>	REAL	Largeur de chanfrein (rel)		
22	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs, rel), voir <_AMODE>		

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
23		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération ou calcul du point de départ uniquement	
					0 =	Mode de compatibilité
					1 =	Usinage normal
				MILLIERS :	Cotation du tourillon à partir du milieu ou du coin	
					0 =	Mode de compatibilité
					1 =	Cotation à partir du milieu
					2 =	Cotation à partir du point d'angle, tourillon +L +W
					3 =	Cotation à partir du point d'angle, tourillon -L +W
					4 =	Cotation à partir du point d'angle, tourillon +L -W
					5 =	Cotation à partir du point d'angle, tourillon -L -W
				DIZAINES DE MILLIERS :	Usinage complet ou reprise	
					0 =	Mode de compatibilité
					1 =	Usinage complet
					2 =	Reprise
24		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	---	Réservé
				CENTAINES :	---	Réservé
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
25		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale Z1 (DP)	
					0 =	Compatibilité
					1 =	Relatif
					2 =	Absolu
				DIZAINES :	Réservé	
				CENTAINES :	Profondeur de pénétration lors du chanfreinage (ZFS)	
					0 =	Absolu
					1 =	Relatif

20.1.18 CYCLE77 - Fraisage d'un tourillon circulaire

Syntaxe

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>,
<_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<_RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Profondeur de tourillon (abs)		
5		<_DPR>	REAL	Profondeur de tourillon (rel), par rapport à Z0 (à saisir sans signe)		
6	∅	<_CDIAM>	REAL	Diamètre du tourillon (à saisir sans signe)		
7	X0	<_PA>	REAL	Point de référence du tourillon par rapport au 1er axe du plan (abs)		
8	Y0	<_PO>	REAL	Point de référence du tourillon par rapport au 2e axe du plan (abs)		
9	DZ	<_MID>	REAL	Profondeur de passe maximale (rel, à saisir sans signe)		
10	UXY	<_FAL>	REAL	Surépaisseur de finition dans le plan (rel), surépaisseur sur le contour du bord		
11	UZ	<_FALD>	REAL	Surépaisseur de finition au fond (rel), surépaisseur au fond (à saisir sans signe)		
12	FX	<_FFP1>	REAL	Avance sur contour		
13	FZ	<_FFD>	REAL	Avance de pénétration		
14		<_CDIR>	INT	Sens de fraisage (à saisir sans signe)		
				UNITES :		
					0 =	En avalant
					1 =	En opposition

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
15		<_VARI>	INT	Type d'usinage		
				UNITES :	Usinage	
					1 =	Ebauchage jusqu'à surépaisseur de finition
					2 =	Finition (surépaisseur X/Y/Z = 0)
					5 =	Chanfreinage
16	∅1	<_AP1>	REAL	Diamètre du tourillon brut		
17	FS	<_FS>	REAL	Largeur de chanfrein (rel)		
18	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs/rel), voir <_AMODE>		
19		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération / calcul du point de départ uniquement	
					0 =	Mode de compatibilité
					1 =	Usinage normal
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Usinage complet / reprise	
					0 =	Mode de compatibilité (<_AP1> à traiter comme précédemment)
					1 =	Usinage complet
					2 =	Reprise
20		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	---	Réservé
				CENTAINES :	---	Réservé
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
21		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale Z1 (DP)	
					0 =	Absolu (mode de compatibilité)
					1 =	Relatif
					2 =	Absolu
				DIZAINES :	Réservé	
				CENTAINES :	Profondeur de pénétration lors du chanfreinage (ZFS)	
					0 =	Absolu
					1 =	Relatif

20.1.19 CYCLE78 - Fraisage de filetages

Syntaxe

```
CYCLE78(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_ADPR>, <_FDPR>, <_LDPR>,
<_DIAM>, <_PIT>, <_PITA>, <_DAM>, <_MDEP>, <_VARI>, <_CDIR>, <_GE>,
<_FFD>, <_FRDP>, <_FFR>, <_FFP2>, <_FFA>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
1	RP	<_RTP>	REAL	Plan de retrait (abs)
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)
4	Z1	<_DP>	REAL	Profondeur de perçage finale (abs/rel), voir <_AMODE>
5		<_ADPR>	REAL	Profondeur de pointage avec avance de perçage réduite (rel), avec <_VARI> DIZAINES DE MILLIERS actif
6	D	<_FDPR>	REAL	Profondeur de passe maximale (rel) D ≥ Z1 ⇒ une passe à la profondeur de perçage finale D < Z1 ⇒ cycle de perçage de trous profonds avec plusieurs passes et débourrage
7	ZR	<_LDPR>	REAL	Profondeur de perçage restante lors du perçage débouchant (rel), avec avance FR
8	∅	<_DIAM>	REAL	Diamètre nominal du filetage
9	P	<_PIT>	REAL	Pas de vis en tant que valeur numérique

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
10		<_PITA>	INT	Unité du pas de vis P		
					1 =	Pas en mm/tr
					2 =	Pas en filets/inch
					3 =	Pas en inch/tr
					4 =	Pas en tant que MODULE
11	DF	<_DAM>	REAL	Valeur/pourcentage pour chaque passe suivante (dégression), voir <_AMODE>		
12	V1	<_MDEP>	REAL	Profondeur de passe minimale (rel), ne prend effet que pour la dégression		
13		<_VARI>	INT	Type d'usinage		
				UNITES :	Réservé	
				DIZAINES :	Débourrage avant fraisage de filetage	
					0 =	Aucun débourrage avant le fraisage de filetages (ne prend effet qu'à la profondeur de perçage finale)
					1 =	Débourrage avant le fraisage de filetages (ne prend effet qu'à la profondeur de perçage finale)
				CENTAINES :	Filetage à droite / à gauche	
					0 =	Filetage à droite
					1 =	Filetage à gauche
				MILLIERS :	Profondeur de perçage restante avec avance de perçage	
					0 =	Aucune profondeur de perçage restante avec avance de perçage FR
					1 =	Profondeur de perçage restante avec avance de perçage FR
				DIZAINES DE MILLIERS :	Pointage avec avance réduite	
					0 =	Aucun pointage avec avance réduite
					1 =	Pointage avec avance réduite Avance de pointage = 0,3 F1, lorsque F1 < 0,15 mm/tour Avance de pointage = 0,1 mm/tour, lorsque F1 ≥ 0,15 mm/tour
14		<_CDIR>	INT	Sens de fraisage	0 =	En avalant
					1 =	En opposition
					4 =	En opposition + en avalant (combinaison ébauchage + finition)
15	Z2	<_GE>	REAL	Valeur de retrait avant fraisage de filetages (rel)		
16	F1	<_FFD>	REAL	Avance de perçage (mm/min ou in/min ou mm/tour)		
17	FR	<_FRDP>	REAL	Avance de perçage pour profondeur de perçage restante (mm/min ou mm/tour)		
18	F2	<_FFR>	REAL	Avance pour fraisage de filetages (mm/min ou mm/dent)		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
19	FS	<_FFP2>	REAL	Avance de finition pour <_CDIR> = 4 (mm/min ou mm/dent)		
20		<_FFA>	INT	Pondération des avances		
				UNITES :	Avance de perçage F1	
				DIZAINES :	Avance de perçage pour profondeur de perçage restante FR	
				CENTAINES :	Avance pour fraisage de filetages F2	
				MILLIERS :	Avance de finition FS	
21		<_PITM>	STRING[15]	Chaîne de caractères servant de mémento pour la saisie du pas de vis (réservé à l'interface)[1)]		
22		<_PTAB>	STRING[20]	Chaîne de caractères pour table de filetages ("", "ISO", "BSW", "BSP", "UNC") (réservé à l'interface)[1)]		
23		<_PTABA>	STRING[20]	Chaîne de caractères pour la sélection dans la table de filetages (p. ex. "M 10", "M 12"...) (réservé à l'interface)[1)]		
24		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées), réservé		
25		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
26		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de perçage = profondeur de perçage finale Z1 abs/rel	
					0 =	Absolu
					1 =	Relatif
				DIZAINES :	Valeur/pourcentage DF pour chaque passe suivante (dégression)	
					0 =	Valeur absolue
					1 =	Pourcentage (0,001 à 100 %)

Remarque

[1)] Les paramètres 21, 22 et 23 sont utilisés uniquement lors de la sélection du filetage dans les tables de filetages du masque de saisie. L'accès aux tables de filetages par la définition de cycle n'est pas possible pendant l'exécution du cycle.

20.1.20 CYCLE79 - Polyèdre

Syntaxe

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<_RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Profondeur du polyèdre (abs/rel), voir <_AMODE>		
5	N	<_NUM>	INT	Nombre d'arêtes (1...n)		
6	SW/L	<_SWL>	REAL	Cote sur plats ou longueur d'arête (en fonction de <_VARI>) ("SW" pour cote sur plats, "L" pour longueur d'arête) Cote sur plats uniquement dans le cas d'un nombre pair d'arêtes ou d'une arête unique		
7	X0	<_PA>	REAL	Point de référence du tourillon, 1er axe (abs)		
8	Y0	<_PO>	REAL	Point de référence du tourillon, 2e axe (abs)		
9	α0	<_STA>	REAL	Angle de rotation centre d'arête par rapport au 1er axe (axe X)		
10	R1/FS1	<_RC>	REAL	Arrondi lorsque <_NUM> > 2 (rayon/chanfrein, voir <_AMODE>) (rel, à saisir sans signe) ("R1" pour rayon, "FS1" pour chanfrein)		
11	∅	<_AP1>	REAL	Diamètre brut du tourillon		
12	DXY	<_MIDA>	REAL	Largeur de passe maximale (unité, voir <_AMODE>)		
13	DZ	<_MID>	REAL	Profondeur de passe maximale		
14	UXY	<_FAL>	REAL	Surépaisseur de finition dans le plan		
15	UZ	<_FALD>	REAL	Surépaisseur de finition au fond		
16	F	<_FFP1>	REAL	Avance d'usinage		
17		<_CDIR>	INT	Sens de fraisage	0 =	En avalant
					1 =	En opposition
18		<_VARI>	INT	Type d'usinage		
				UNITES :	Usinage	
					1 =	Ebauchage
					2 =	Finition
					3 =	Finition du bord
					5 =	Chanfreinage
				DIZAINES :	Cote sur plats ou longueur d'arête	
					0 =	Cote sur plats
					1 =	Longueur d'arête

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
19	FS	<_FS>	REAL	Largeur de chanfrein (rel)		
20	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs/rel), voir <_AMODE>		
21		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération ou calcul du point de départ uniquement	
					1 =	Usinage normal
22		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	---	Réservé
				CENTAINES :	---	Réservé
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée
23		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur finale (<_DP>)	
					0 =	Absolu
					1 =	Relatif
				DIZAINES :	Unité pour la profondeur de passe dans le plan (<_MIDA>)	
					0 =	mm
					1 =	% du diamètre de l'outil
				CENTAINES :	Profondeur de pénétration lors du chanfreinage (ZFS) (<_ZFS>)	
					0 =	Absolu
					1 =	Relatif
				MILLIERS :	Arrondi (<_RC>)	
					0 =	Rayon
					1 =	Chanfrein

20.1.21 CYCLE81 - Perçage, centrage

Syntaxe

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<RTP>	REAL	Plan de retrait (abs)		
2	Z0	<RFP>	REAL	Point de référence (abs)		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1/∅	<DP>	REAL	Profondeur de perçage (abs) / diamètre de centrage (abs), voir <_GMODE>		
5	Z1	<DPR>	REAL	Profondeur de perçage (rel)		
6	DT	<DTB>	REAL	Arrêt temporisé à la profondeur de perçage finale, voir <_AMODE>		
7		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Centrage par rapport à la profondeur / au diamètre	
					0 =	Compatibilité, profondeur
					1 =	Diamètre
8		<_DMODE>	INT	Mode affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
9		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de perçage Z1 (abs/rel)	
					0 =	Compatibilité issue de la programmation DP/DPR
					1 =	Relatif
					2 =	Absolu
				DIZAINES :	Arrêt temporisé à la profondeur de perçage finale DT en secondes/tours	
					0 =	Compatibilité à partir du signe de DTB (> 0 seconde ou < 0 tour)
					1 =	En secondes
					2 =	En tours

20.1.22 CYCLE82 - Perçage, lamage

Syntaxe

```
CYCLE82(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<RTP>	REAL	Plan de retrait (abs)		
2	Z0	<RFP>	REAL	Point de référence (abs)		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<DP>	REAL	Profondeur de perçage (abs), voir <_AMODE>		
5	Z1	<DPR>	REAL	Profondeur de perçage (rel), voir <_AMODE>		
6	DT	<DTB>	REAL	Arrêt temporisé à la profondeur de perçage finale, voir <_AMODE>		
7		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Profondeur de perçage par rapport à pointe/corps	
					0 =	Compatibilité, pointe
					1 =	Corps
8		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Entrée : Complète
					1 =	Entrée : Simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
9		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de perçage Z1 (abs/rel)	
					0 =	Compatibilité issue de la programmation DP/DPR
					1 =	Relatif
					2 =	Absolu
				DIZAINES :	Arrêt temporisé DT à la profondeur de perçage finale en secondes/tours	
					0 =	Compatibilité à partir du signe de DT (> 0 seconde / < 0 tour)
					1 =	En secondes
					2 =	En tours
				CENTAINES :	Profondeur de pointage ZA abs/rel	
					0 =	Relatif
					1 =	Absolu
				MILLIERS :	Evaluation avance de pointage	
					0 =	En % de l'avance de perçage
					1 =	F/min
					2 =	F/tr
				DIZAINES DE MILLIERS :	Profondeur de perçage restante ZD abs/rel	
					0 =	Relatif
					1 =	Absolu
				CENTAINES DE MILLIERS :	Evaluation avance de perçage restante	
					0 =	En % de l'avance de perçage
					1 =	F/min
					2 =	F/tr
10		<_VARI>	INT	Type d'usinage pointage/perçage débouchant		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Perçage débouchant	
					0 =	Perçage débouchant "Non"
					1 =	Perçage débouchant "Oui"
				DIZAINES DE MILLIERS :	Pointage	
					0 =	Pointage "Non"
					1 =	Pointage "Oui"
11	ZA	<S_ZA>	REAL	Profondeur de pointage relative par rapport au point de référence ou absolue (voir <_AMODE> CENTAINES)		
12	FA	<S_FA>	REAL	Avance de pointage comme valeur ou en % (en association avec <_AMODE> MILLIERS)		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
13	ZD	<S_ZD>	REAL	Profondeur de perçage restante relative par rapport à la profondeur de perçage finale ou absolue (en association avec <_AMODE> DIZAINES DE MILLIERS)
14	FD	<S_FD>	REAL	Avance de perçage restante comme valeur ou en % (en association avec <_AMODE> CENTAINES DE MILLIERS)

20.1.23 CYCLE83 - Perçage profond

Syntaxe

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>, <_DAM>,
<DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>, <_DTD>,
<_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<RTP>	REAL	Plan de retrait (abs)		
2	Z0	<RFP>	REAL	Point de référence (abs)		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<DP>	REAL	Profondeur de perçage finale (abs), voir <_AMODE>		
5	Z1	<DPR>	REAL	Profondeur de perçage finale (rel), voir <_AMODE>		
6	D	<FDEP>	REAL	1ère profondeur de perçage (abs), voir <_AMODE>		
7	D	<FDPR>	REAL	1ère profondeur de perçage (rel), voir <_AMODE>		
8	DF	<_DAM>	REAL	Valeur/pourcentage de dégression pour chaque passe suivante, voir <_AMODE>		
9	DTB	<DTB>	REAL	Arrêt temporisé à la profondeur de perçage, voir <_AMODE>		
10	DTS	<DTS>	REAL	Arrêt temporisé au point de départ (pour le débourrage uniquement), voir <_AMODE>		
11	FD1	<FRF>	REAL	Valeur en pourcentage de l'avance pour la première passe, voir <_AMODE>		
12		<VARI>	INT	Type d'usinage		
				UNITES :	Bris de copeaux / débourrage	
					0 =	Bris de copeaux
					1 =	Débourrage
13		<_AXN>	INT	Axe d'outil		
					0 =	3ème axe géométrique
					1 =	1er axe géométrique
					2 =	2ème axe géométrique
					> 2	3ème axe géométrique
14	V1	<_MDEP>	REAL	Profondeur de passe minimale (uniquement avec pourcentage pour dégression)		

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
15	V2	<_VRT>	REAL	Valeur de retrait après chaque usinage (uniquement pour bris de copeaux)		
					> 0	Valeur de retrait variable
					0 =	Valeur par défaut 1 mm
16	DT	<_DTD>	REAL	Arrêt temporisé à la profondeur de perçage finale, voir <_AMODE>		
17	V3	<_DIS1>	REAL	Précote d'arrêt (uniquement pour débourrage), voir <_AMODE>		
18		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Profondeur de perçage par rapport à pointe/corps	
					0 =	Pointe
					1 =	Corps
19		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	---	Réservé
				CENTAINES :	---	Réservé
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
20		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de perçage = profondeur de perçage finale Z1 (abs/rel)	
					0 =	Compatibilité issue de la programmation <DP>/<DPR>
					1 =	Relatif
					2 =	Absolu
				DIZAINES :	Arrêt temporisé à la profondeur de perçage DTB en secondes/tours	
					0 =	Compatibilité à partir du signe de DTB (> 0 seconde ou < 0 tour)
					1 =	En secondes
					2 =	En tours
				CENTAINES :	Arrêt temporisé au point de départ de DTS en secondes/tours	
					0 =	Compatibilité à partir du signe de DTS (> 0 seconde ou < 0 tour)
					1 =	En secondes
					2 =	En tours
				MILLIERS :	Arrêt temporisé à la profondeur de perçage finale DTD en secondes/tours	
					0 =	Compatibilité à partir du signe de DTD (> 0 seconde ou < 0 tour)
					1 =	En secondes
					2 =	En tours
				DIZAINES DE MILLIERS :	1ère profondeur de perçage D (abs/rel)	
					0 =	Compatibilité issue de la programmation<FDEPF>/<DPR>
					1 =	Relatif
					2 =	Absolu
				CENTAINES DE MILLIERS :	Valeur/pourcentage de dégression <_DAM> pour chaque passe suivante	
					0 =	Compatibilité à partir du signe de <_DAM> (> 0 valeur de dégression ou < 0 facteur de 0,001 à 1,0)
					1 =	Valeur de la dégression
					2 =	Pourcentage (0,001 à 100 %)
				MILLIONS :	Précote d'arrêt V3 automatique/manuel	
					0 =	Compatibilité à partir du signe de <_DIS1> (= 0 automatique ou > 0 manuel)
					1 =	Automatique (calculée dans le cycle)
					2 =	Manuel (valeur programmée)
				DIZAINES DE MILLIONS :	Facteur d'avance pour la première passe <FRF> en tant que facteur/pourcentage	

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
					0 =	Compatibilité en tant que facteur (0,001 à 1,0, FRF = 0 signifie 100 %)
					1 =	Pourcentage (0,001 à 999,999 %)

20.1.24 CYCLE84 - Taraudage sans porte-taraud compensateur

Syntaxe

```
CYCLE84(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDAC>, <MPIT>,
<PIT>, <POSS>, <SST>, <SST1>, <_AXN>, <_PITA>, <_TECHNO>, <_VARI>,
<_DAM>, <_VRT>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<RTP>	REAL	Plan de retrait (abs)		
2	Z0	<RFP>	REAL	Point de référence (abs)		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<DP>	REAL	Profondeur de perçage = profondeur de perçage finale (abs), voir <_AMODE>		
5	Z1	<DPR>	REAL	Profondeur de perçage = profondeur de perçage finale (rel), voir <_AMODE>		
6	DT	<DTB>	REAL	Arrêt temporisé à la profondeur de perçage en secondes		
7	SDE	<SDAC>	INT	Sens de rotation après fin du cycle		
8		<MPIT>	REAL	Taille de filetage uniquement pour "ISO métrique" (le pas est calculé en interne pendant l'exécution)		
9	P	<PIT>	REAL	Pas de vis en tant que valeur, unité de mesure voir <_PITA>		
10	αS[1)]	<POSS>	REAL	Position de broche pour un arrêt orienté de la broche		
11	S	<SST>	REAL	Vitesse de rotation de la broche pour taraudage		
12	SR	<SST1>	REAL	Vitesse de rotation de la broche pour le retrait		
13		<_AXN>	INT	Axe de perçage	0 =	3. axe géométrique
					1 =	1. axe géométrique
					2 =	2. axe géométrique
					≥ 3 =	3. axe géométrique

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification			
14		<_PITA>	INT	Unité de mesure du pas de vis (évaluation <PIT> et <MPIT>)			
					0 =	Pas en mm	- Evaluation <MPIT>/<PIT>
					1 =	Pas en mm	- Evaluation <PIT>
					2 =	Pas en TPI	- Evaluation <PIT> (nombre de filets par inch)
					3 =	Pas en inch	- Evaluation <PIT>
					4 =	MODULE	- Evaluation <PIT>
15		<_TECHNO>	INT	Technologie[1)]			
				UNITES :	Type d'arrêt précis		
					0 =	Type d'arrêt précis comme avant appel de cycle actif	
					1 =	Arrêt précis G601	
					2 =	Arrêt précis G602	
					3 =	Arrêt précis G603	
				DIZAINES :	Commande anticipatrice		
					0 =	Avec/sans commande anticipatrice actif comme avant l'appel de cycle	
					1 =	Avec commande anticipatrice FFWON	
					2 =	Sans commande anticipatrice FFWOF	
				CENTAINES :	Accélération		
					0 =	SOFT/BRISK/DRIVE actif comme avant l'appel de cycle	
					1 =	Avec limitation des à-coups SOFT	
					2 =	Sans limitation des à-coups BRISK	
					3 =	Accélération réduite DRIVE	
				MILLIERS :	Mode broche MCALL		
					0 =	Réactivation de la broche avec MCALL	
					1 =	Maintien de l'asservissement de position avec MCALL	
16		<_VARI>	INT	Type d'usinage			
				UNITES :	0 =	1 passe	
					1 =	Bris de copeaux (taraudage profond)	
					2 =	Débourrage (taraudage profond)	
				MILLIERS :	Mode ISO/SIEMENS non significatif pour masque de saisie		
					0 =	Appel à partir de compatibilité ISO	
					1 =	Appel à partir du contexte SIEMENS	
17	D	<_DAM>	REAL	Profondeur de passe maximale (uniquement pour débourrage / bris de copeaux)			
18	V2	<_VRT>	REAL	Valeur de retrait après chaque usinage (uniquement pour bris de copeaux), voir <_AMODE>			

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
19		<_PITM>	STRING[15]	Chaîne de caractères servant de mémento pour la saisie du pas de vis[2]		
20		<_PTAB>	STRING[5]	Chaîne de caractères pour table de filetages ("", "ISO", "BSW", "BSP", "UNC")[2)]		
21		<_PTABA>	STRING[20]	Chaîne de caractères pour la sélection dans la table de filetages (p. ex. "M 10", "M 12"...)[2)]		
22		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
23		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Mode de compatibilité (uniquement pour décompilation du masque de saisie), lorsque PM 52216 bit 0 = 1[1)]	
					0 =	Les paramètres technologiques sont affichés (compatibilité) : les paramètres TECHNO prennent effet
					1 =	Les paramètres technologiques ne sont pas affichés : la technologie "programmée avant l'appel de cycle" reste effective
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
24		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de perçage = profondeur de perçage finale Z1 (abs/rel)	
					0 =	Compatibilité issue de la programmation <DP>/<DPR>
					1 =	Relatif
					2 =	Absolu
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Sens de rotation du filet droite/gauche	
					0 =	Compatibilité à partir du signe PIT/MPIT
					1 =	Droite
					2 =	Gauche
				DIZAINES DE MILLIERS :	Réservé	
				CENTAINES DE MILLIERS :	Réservé	
				MILLIONS :	Valeur de retrait, en fonction de chaque usinage V2 manuel/automatique	
					0 =	Compatibilité issue de la programmation <_VRT> (> 0 valeur variable ou ≤ 0 valeur par défaut 1 mm/0,0394 inch)
					1 =	Automatique (valeur par défaut 1 mm/ 0,0394 inch)
					2 =	Manuel (comme programmé sous V2)

[1] Les champs Technologie peuvent être masqués, en fonction de la donnée de réglage SD52216 $MCS_FUNCTION_MASK_DRILL

[2] Les paramètres 19, 20 et 21 sont utilisés uniquement lors de la sélection du filetage dans les tables de filetages du masque de saisie. L'accès aux tables de filetages par la définition de cycle n'est pas possible pendant l'exécution du cycle.

20.1.25 CYCLE85 - Alésage à l'alésoir

Syntaxe

```
CYCLE85(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
1	RP	<RTP>	REAL	Plan de retrait (abs)
2	Z0	<RFP>	REAL	Point de référence (abs)

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<DP>	REAL	Profondeur de perçage (abs), voir <_AMODE>		
5	Z1	<DPR>	REAL	Profondeur de perçage (rel), voir <_AMODE>		
6	DT	<DTB>	REAL	Arrêt temporisé à la profondeur de perçage finale, voir <_AMODE>		
7	F	<FFR>	REAL	Avance		
8	FR	<RFF>	REAL	Avance pour le retrait		
9		<_GMODE>	INT	Réservé		
10		<_DMODE>	INT	Mode affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
11		<_AMODE>	INT	Mode alternatif (perçage)		
				UNITES :	Profondeur de perçage Z1 (abs/rel)	
					0 =	Compatibilité issue de la programmation DP/DPR
					1 =	Relatif
					2 =	Absolu
				DIZAINES :	Arrêt temporisé DT à la profondeur de perçage finale en secondes/tours	
					0 =	Compatibilité à partir du signe de DT (> 0 seconde ou < 0 tour)
					1 =	En secondes
					2 =	En tours

20.1.26 CYCLE86 - Alésage au tour

Syntaxe

```
CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>,
<RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

N°	Masque de paramètre	Paramètre interne	Type de données	Signification
1	RP	<RTP>	REAL	Plan de retrait (abs)
2	Z0	<RFP>	REAL	Point de référence (abs)

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<DP>	REAL	Profondeur de perçage (abs), voir <_AMODE>		
5	Z1	<DPR>	REAL	Profondeur de perçage (rel), voir <_AMODE>		
6	DT	<DTB>	REAL	Arrêt temporisé à la profondeur de perçage finale, voir <_AMODE>		
7	DIR	<SDIR>	INT	Sens de rotation de la broche	3 =	M3
					4 =	M4
8	DX	<RPA>	REAL	Valeur de relèvement dans la direction X		
9	DY	<RPO>	REAL	Valeur de relèvement dans la direction Y		
10	DZ	<RPAP>	REAL	Valeur de relèvement dans le sens Z		
11	SPOS	<POSS>	REAL	Position de la broche pour le relèvement (pour arrêt orienté de la broche, en degrés)		
12		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Mode de relèvement	
					0 =	Relèvement, compatibilité
					1 =	Pas de relèvement
13		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
14		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de perçage Z1 (abs/rel)	
					0 =	Compatibilité issue de la programmation<DP>/<DPR>
					1 =	Relatif
					2 =	Absolu
				DIZAINES :	Arrêt temporisé à la profondeur de perçage finale DT en secondes/tours	
					0 =	Compatibilité à partir du signe de DT (> 0 seconde ou < 0 tour)
					1 =	En secondes
					2 =	En tours

20.1.27 CYCLE92 - Tronçonnage

Syntaxe

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>, <_SV1>,
<_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>, <_VARI>, <_DN>,
<_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	X0	<_SPD>	REAL	Point de référence (abs, toujours diamètre)		
2	Y0	<_SPL>	REAL	Point de référence (abs)		
3	X1	<_DIAG1>	REAL	Profondeur pour réduction de vitesse de rotation, voir <_AMODE> (UNITES)		
4	X2	<_DIAG2>	REAL	Profondeur finale, voir <_AMODE> (DIZAINES)		
5	R/FS	<_RC>	REAL	Rayon d'arrondi ou largeur de chanfrein, voir <_AMODE> (MILLIERS)		
6	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
7	S	<_SV1>	REAL	Vitesse de rotation de broche constante, voir <_AMODE> (DIZAINES DE MILLIERS)		
	V			Vitesse de coupe constante		
8	SV	<_SV2>	REAL	Vitesse de rotation maximale pour une vitesse de coupe constante		
9	DIR	<_SDAC>	INT	Sens de rotation de la broche	3 =	Pour M3
					4 =	Pour M4
10	F	<_FF1>	REAL	Avance jusqu'à la profondeur de réduction de vitesse de rotation		
11	FR	<_FF2>	REAL	Avance réduite jusqu'à la profondeur finale		
12	SR	<_SS2>	REAL	Vitesse de rotation réduite jusqu'à la profondeur finale		
13	XM	<_DIAGM>	REAL	Profondeur de sortie du récepteur de pièce (abs, toujours diamètre)		
14		<_VARI>	INT	Type d'usinage		
				UNITES :	Retrait	
					0 =	Retrait à <_SPD> + <_SDIS>
					1 =	Pas de retrait à la fin
				DIZAINES :	Récepteur de pièces	
					0 =	Non, n'exécuter aucune instruction M
					1 =	Oui, appel de CUST_TECHCYC(101)- ouverture du tiroir, CUST_TECHCYC(102)- fermeture du tiroir
15		<_DN>	INT	Numéro D pour 2e tranchant, si non programmé ⇒ D+1		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
20		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
21		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur pour réduction de la vitesse de rotation (<_DIAG1>)	
					0 =	En absolu, valeur de l'axe transversal en diamètre
					1 =	En relatif, valeur de l'axe transversal en rayon
				DIZAINES :	Profondeur finale (<_DIAG2>)	
					0 =	En absolu, valeur de l'axe transversal en diamètre
					1 =	En relatif, valeur de l'axe transversal en rayon
				CENTAINES :	Réservé	
				MILLIERS :	Rayon/chanfrein (<_RC>)	
					0 =	Rayon
					1 =	Chanfrein
				DIZAINES DE MILLIERS :	Vitesse de rotation de la broche / vitesse de coupe (<_SV1>)	
					0 =	Vitesse de rotation de broche constante
					1 =	Vitesse de coupe constante

20.1.28 CYCLE95 - Chariotage au contour

Syntaxe

```
CYCLE95(<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>,
<_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	CON	<NPP>	STRING [140]	Nom du contour		
2	D	<MID>	REAL	Profondeur de passe maximale en ébauchage, voir <_GMODE>		
3	UZ	<FALZ>	REAL	Surépaisseur de finition en Z		
4	UX	<FALX>	REAL	Surépaisseur de finition en X		
5	U	<FAL>	REAL	Surépaisseur de finition parallèle au contour (prend effet dans les deux axes)		
6	F	<FF1>	REAL	Avance pour ébauchage		
7	FY	<FF2>	REAL	Avance de plongée Détalonnages		
8	FS	<FF3>	REAL	Avance de finition		
9		<_VARI>	INT	Type d'usinage		
				UNITES et DIZAINES :		
					1 =	Ebauchage, longitudinal, extérieur
					2 =	Ebauchage, transversal, extérieur
					3 =	Ebauchage, longitudinal, intérieur
					4 =	Ebauchage, transversal, intérieur
					5 =	Finition, longitudinale, extérieure
					6 =	Finition, transversale, extérieure
					7 =	Finition, longitudinale, intérieure
					8 =	Finition, transversale, intérieure
					9 =	Usinage complet, longitudinal, extérieur
					10 =	Usinage complet, transversal, extérieur
					11 =	Usinage complet, longitudinal, intérieur
					12 =	Usinage complet, transversal, intérieur
				CENTAINES :		
					0 =	Avec lissage sur le contour, sans matière restante dans les angles
					1 =	Sans lissage sur le contour
					2 =	Lissage uniquement jusqu'au point d'intersection précédent, de la matière peut rester dans les angles
10	DT	<DT>	REAL	Temporisation lors de l'interruption de l'avance		
11	DI	<DAM>	REAL	Ecart entre les interruptions de l'avance		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
12	VRT	<_VRT>	REAL	Relèvement du contour		
					0 =	Un relèvement de 1 mm est utilisé en interne, indépendamment du système de mesure anglo-saxon / métrique
					> 0 =	Relèvement
13		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Evaluation de la profondeur de passe	
					0 =	Profondeur de passe calculée en fonction du groupe G DIAMON/DIAMOF
					1 =	La profondeur de passe prend effet en tant que valeur de rayon (indépendamment de DIAMON/ DIAMOF)
14		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				MILLIERS :		
					0 =	Mode de compatibilité : le nom du contour figure dans NPP
					1 =	Le nom du contour est programmé dans le CYCLE62 et transféré dans _SC_CONT_NAME

20.1.29 CYCLE98 - Concaténation de filetages

Syntaxe

```
CYCLE98(<_PO1>, <_DM1>, <_PO2>, <_DM2>, <_PO3>, <_DM3>, <_PO4>,
<_DM4>, <APP>, <ROP>, <TDEP>, <FAL>, <_IANG>, <NSP>, <NRC>, <NID>,
<_PP1>, <_PP2>, <_PP3>, <_VARI>, <_NUMTH>, <_VRT>, <_MID>, <_GDEP>,
<_IFLANK>, <_PITA>, <_PITM1>, <_PITM2>, <_PITM3>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
1	Z0	<_PO1>	REAL	Point de référence en Z (abs)
2	X0	<_DM1>	REAL	Point de référence en X (abs), dans le diamètre
3	Z1	<_PO2>	REAL	Point intermédiaire 1 en Z (abs/rel), voir <_AMODE> (UNITES)
4	X1	<_DM2>	REAL	Point intermédiaire 1 en X (abs/rel), voir <_AMODE> (DIZAINES) ou
	X1α			Inclinaison du filetage 1 (-90° à 90°) abs toujours diamètre, rel toujours rayon
5	Z2	<_PO3>	REAL	Point intermédiaire 2 en Z (abs/rel), voir <_AMODE> (CENTAINES)
6	X2	<_DM3>	REAL	Point intermédiaire 2 en X (abs/rel), voir <_AMODE> (MILLIERS) ou
	X2α			Inclinaison du filetage 2 (-90° à 90°) abs toujours diamètre, rel toujours rayon
7	Z3	<_PO4>	REAL	Point final en Z (abs/rel), voir <_AMODE> (DIZAINES DE MILLIERS)
8	X3	<_DM4>	REAL	Point final en X (abs/rel), voir <_AMODE> (CENTAINES DE MILLIERS) ou
	X3α			Inclinaison du filetage 3 (-90° à 90°) abs toujours diamètre, rel toujours rayon
9	LW	<APP>	REAL	Course d'approche du filetage (rel, à saisir sans signe)
10	LR	<ROP>	REAL	Sortie de filetage (rel, à saisir sans signe)
11	H1	<TDEP>	REAL	Profondeur de filet (rel, à saisir sans signe)
12	U	<FAL>	REAL	Surépaisseur de finition en X et Z
13	DP	<_IANG>	REAL	Pente de pénétration en tant que distance ou angle, voir <_AMODE> (MILLIONS)
	αP			La pente de pénétration prend effet en fonction du réglage du paramètre <_VARI> (CENTAINES).
				Définition pour <_VARI_CENTAINES = 0 - mode compatibilité :
				> 0 = Pénétration oblique le long d'un flanc
				0 = Pénétration perpendiculaire dans le filet
				< 0 = Pénétration oblique avec flancs en alternance
				Définition pour _VARI_CENTAINES<>0 :
				> 0 = Pénétration sur front montant
				0 = Pénétration centrale
				< 0 = Pénétration sur front descendant
14	α0	<NSP>	REAL	Décalage de l'angle d'attaque pour le 1er filet
15		<NRC>	INT	Nombre de passes d'ébauchage, voir <_VARI> (DIZAINES DE MILLIERS)
16	NN	<NID>	INT	Nombre de passes à vide
17	P0	<_PP1>	REAL	Pas de vis 1ère section du filet, voir <_PITA>
18	P1	<_PP2>	REAL	Pas de vis 2e section du filet, voir <_PITA>
19	P2	<_PP3>	REAL	Pas de vis 3e section du filet, voir <_PITA>

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
20		<_VARI>	INT	Usinage		
				UNITES :	Technologie	
					1 =	Filetage extérieur avec passe linéaire
					2 =	Filetage intérieur avec passe linéaire
					3 =	Filetage extérieur avec passe dégressive, la section du copeau reste constante
					4 =	Filetage intérieur avec passe dégressive, la section du copeau reste constante
				DIZAINES :	Réservé	
				CENTAINES :	Type de passe	
					0 =	Mode compatibilité pour <_IANG>
					1 =	Passe d'un seul côté
					2 =	Passe en alternance
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Profondeur de passe alternative	
					0 =	Compatibilité, indication du nombre de passes d'ébauchage (<_NRC>)
					1 =	Indication d'une valeur pour la 1ère passe (<_MID>)
				CENTAINES DE MILLIERS :	Type d'usinage	
					0 =	Compatibilité (ébauchage et finition)
					1 =	Ebauchage
					2 =	Finition
					3 =	Ebauchage et finition
				MILLIONS :	Ordre d'exécution dans le cas d'un filetage multifilet	
					0 =	Ordre des filets croissant
					1 =	Ordre des filets par rapport à
21	N	<_NUMTH>	INT	Nombre de filets		
22		<_VRT>	REAL	Distance de retrait (rel)		
					0 =	Un relèvement de 1 mm est utilisé en interne, indépendamment du système de mesure anglo-saxon / métrique
					> 0 =	Relèvement
23	D1	<_MID>	REAL	Première passe, voir <_VARI> (DIZAINES DE MILLIERS)		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
24	DA	<_GDEP>	REAL	Profondeur de changement de filet (ne prend effet que pour "multifilet")		
					0 =	Ne prendre en compte aucune profondeur de changement de filet
					> 0 =	Prendre en compte la profondeur de changement de filet
25		<_IFLANK>	REAL	Pente de pénétration en tant que largeur (uniquement pour la surface)		
26		<_PITA>	INT	Evaluation du pas de vis		
					0 =	Mode compatibilité pour le pas de vis, évaluation <_PP1> à <_PP3> comme précédemment, en fonction du système de mesure métrique / anglo-saxon
					1 =	Pas en mm
					2 =	Pas en TPI (nombre de filets par inch)
					3 =	Pas en inch
					4 =	MODULE
27		<_PITM1>	STRING[15]	Chaîne de caractères servant de mémento pour la saisie du pas de vis (réservé à l'interface)		
28		<_PITM2>	STRING[15]	Chaîne de caractères servant de mémento pour la saisie du pas de vis (réservé à l'interface)		
29		<_PITM3>	STRING[15]	Chaîne de caractères servant de mémento pour la saisie du pas de vis (réservé à l'interface)		
30		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	---	Réservé
				CENTAINES :	---	Réservé
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
31		<_AMODE>	INT	Mode alternatif		
				UNITES :	1. point intermédiaire en Z (Z1)	
					0 =	Absolu
					1 =	Relatif
				DIZAINES :	1. point intermédiaire en X (X1)	
					0 =	Absolu
					1 =	Relatif
					2 =	α
				CENTAINES :	2. point intermédiaire en Z (Z2)	
					0 =	Absolu
					1 =	Relatif
				MILLIERS :	2. point intermédiaire en X (X2)	
					0 =	Absolu
					1 =	Relatif
					2 =	α
				DIZAINES DE MILLIERS :	Point final en Z (Z3)	
					0 =	Absolu
					1 =	Relatif
				CENTAINES DE MILLIERS :	Point final en X (X3)	
					0 =	Absolu
					1 =	Relatif
					2 =	α
				MILLIONS :	Sélection de la pente de pénétration en tant qu'angle ou largeur	
					0 =	Angle de pénétration <_IANG>
					1 =	Pente de pénétration <_IFLANK>
				DIZAINES DE MILLIONS :	Passage unique / plusieurs passages	
					0 =	Mode compatibilité (l'angle d'attaque <_NSP> est évalué)
					1 =	Passage unique (avec décalage de l'angle d'attaque <_NSP>)
					2 =	Multifilet

20.1.30 CYCLE99 - Filetage au tour

Syntaxe

```
CYCLE99(<_SPL>, <_SPD>, <_FPL>, <_FPD>, <_APP>, <_ROP>, <_TDEP>,
<_FAL>, <_IANG>, <_NSP>, <_NRC>, <_NID>, <_PIT>, <_VARI>, <_NUMTH>,
<_SDIS>, <_MID>, <_GDEP>, <_PIT1>, <_FDEP>, <_GST>, <_GUD>,
```

```
<_IFLANK>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>,
<_S_XRS>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	Z0	<_SPL>	REAL	Point de référence (abs)		
2	X0	<_SPD>	REAL	Point de référence (abs, toujours diamètre)		
3	Z1	<_FPL>	REAL	Point final en association avec <_AMODE> (UNITES)		
4	X1	<_FPD>	REAL	Point final, en association avec <_AMODE> (DIZAINES)		
5	LW/LW2	<_APP>	REAL	Course d'approche du filetage en association avec <_AMODE> (CENTAINES) ou Entrée de filetage = sortie de filetage en association avec <_AMODE> (CENTAINES)		
6	LR	<_ROP>	REAL	Sortie de filetage		
7	H1	<_TDEP>	REAL	Profondeur de filet		
8	U	<_FAL>	REAL	Surépaisseur de finition en X et Z		
9	DP	<_IANG>	REAL	Pente de pénétration en tant que distance ou angle, en association avec <_AMODE> (MILLIERS)		
	αP				> 0 =	Pénétration sur front montant
					< 0 =	Pénétration sur front descendant
					0 =	Pénétration centrale
10	α0	<_NSP>	REAL	Décalage de l'angle d'attaque (effectif pour "passage unique" seulement)		
11	ND	<_NRC>	INT	Nombre de passes d'ébauchage, en association avec <_VARI> (DIZAINES DE MILLIERS)		
12	NN	<_NID>	INT	Nombre de passes à vide		
13	P	<_PIT>	REAL	Pas de vis en tant que valeur, en association avec <_PITA>		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
14		<_VARI>	INT	Type d'usinage		
				UNITES :	Technologie	
					1 =	Filetage extérieur avec passe linéaire
					2 =	Filetage intérieur avec passe linéaire
					3 =	Filetage extérieur avec passe dégressive, la section du copeau reste constante
					4 =	Filetage intérieur avec passe dégressive, la section du copeau reste constante
				DIZAINES :	Réservé	
				CENTAINES :	Type de passe	
					1 =	Passe d'un seul côté
					2 =	Passe en alternance
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Profondeur de passe alternative	
					0 =	Indication du nombre de passes d'ébauchage (<_NRC>)
					1 =	Indication d'une valeur pour la 1ère passe (<_MID>)
				CENTAINES DE MILLIERS :	Type d'usinage	
					1 =	Ebauchage
					2 =	Finition
					3 =	Ebauchage et finition
				MILLIONS :	Ordre d'exécution dans le cas d'un filetage multifilet	
					0 =	Ordre des filets croissant
					1 =	Ordre des filets par rapport à
15	N	<_NUMTH>	INT	Nombre de filets		
16	VR	<_SDIS>	REAL	Distance de retrait, rel		
17	D1	<_MID>	REAL	Première profondeur de passe, en association avec <_VARI> (DIZAINES DE MILLIERS)		
18	DA	<_GDEP>	REAL	Profondeur de changement de filet (ne prend effet que pour "multifilet")		
					0 =	Ne prendre en compte aucune profondeur de changement de filet
					> 0 =	Prendre en compte la profondeur de changement de filet
19	G	<_PIT1>	REAL	Modification du pas par tour		
					0 =	Le pas de vis est constant (G33)
					> 0 =	Le pas de vis augmente (G34)
					< 0 =	Le pas de vis diminue (G35)
20		<_FDEP>	REAL	Profondeur de plongée (à saisir sans signe)		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
21	N1	<_GST>	INT	Premier filet N1 = 1...N, en association avec <_AMODE> (CENTAINES DE MILLIERS)		
22		<_GUD>	INT	Réservé		
23		<_IFLANK>	REAL	Pente de pénétration en tant que largeur (uniquement pour la surface)		
24		<_PITA>	INT	Unité de mesure du pas de vis (évaluation PIT et/ou MPIT)		
					0 =	Pas en mm - évaluation MPIT/PIT
					1 =	Pas en mm - évaluation PIT
					2 =	Pas en TPI - évaluation PIT (nombre de filets par inch)
					3 =	Pas en inch - évaluation PIT
					4 =	MODULE - évaluation PIT
25		<_PITM>	STRING[15]	Chaîne de caractères servant de mémento pour la saisie du pas de vis (réservé à l'interface)[1)]		
26		<_PTAB>	STRING[20]	Chaîne de caractères pour table de filetages (réservé à l'interface)[1)]		
27		<_PTABA>	STRING[20]	Chaîne de caractères pour sélection dans la table de filetages (réservé à l'interface)[1)]		
28		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Mode de filetage	
					0 =	Filetage longitudinal
					1 =	Filetage transversal
					2 =	Filetage sur corps conique
				CENTAINES :	---	Réservé
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
29		<_AMODE>	INT	Mode alternatif		
				UNITES :	Longueur de filetage en Z	
					0 =	Absolu
					1 =	Relatif
				DIZAINES :	Longueur de filetage en X	
					0 =	En absolu, valeur de l'axe transversal en diamètre
					1 =	En relatif, valeur de l'axe transversal en rayon
					2 =	α
				CENTAINES :	Evaluation course/trajet d'approche <_APP>	
					0 =	Course d'approche du filetage <_APP>
					1 =	Entrée de filetage = sortie de filetage <_APP> = -<_ROP>
					2 =	Indiquer la distance de l'entrée de filetage <_APP> = -<_APP>
				MILLIERS :	Sélection de la pente de pénétration en tant qu'angle ou largeur	
					0 =	Angle de pénétration <_IANG>
					1 =	Pente de pénétration <_IFLANK>
				DIZAINES DE MILLIERS :	Passage unique / plusieurs passages	
					0 =	Passage unique (avec décalage de l'angle d'attaque <_NSP>)
					1 =	Multifilet
				CENTAINES DE MILLIERS :	Premier filet <_GST>	
					0 =	Usinage complet
					1 =	Commencer l'usinage à partir de ce filet
					2 =	Usiner uniquement ce filet
				MILLIONS :	Compensation de flèche pour filetage longitudinal	
					0 =	Hauteur de segment du filetage bombé XS
					1 =	Rayon du filetage bombé RS
30	XS / RS	<_S_XRS>	REAL	Compensation de flèche pour filetage longitudinal en association avec <_AMODE> : MILLIONS		

Remarque

[1] Les paramètres <_PITM>, <_PTAB> et <_PTABA> sont utilisés uniquement lors de la sélection du filetage dans les tables de filetages du masque de saisie.
L'accès aux tables de filetages par la définition de cycle n'est pas possible pendant l'exécution du cycle.

20.1.31 CYCLE435 - Définir le système de coordonnées du dresseur

Syntaxe

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1		<_T>	STRING[32]	Nom d'outil de la meule		
2		<_DD>	INT	Numéro de tranchant de la meule		
3		<S_TA>	STRING[32]	Point de référence outil de dressage - Nom de l'outil de dressage		
4		<S_DA>	INT	Numéro de tranchant de l'outil de dressage		
5		<S_AD>	REAL	Profondeur de passe de dressage - diamètre		
6		<S_AL>	REAL	Profondeur de passe de dressage - plan		
7		<S_PVD>	REAL	Décalage de profilage - diamètre		
8		<S_PVL>	REAL	Décalage de profilage - plan		
9		<S_PD>	REAL	Surépaisseur de profilage - diamètre		
10		<S_PL>	REAL	Surépaisseur de profilage - plan		
11		<_AMODE>	INT	Mode alternatif		
				UNITES :	outil actif en fin de cycle	
					0 =	outil de dressage actif
					1 =	meule active

20.1.32 CYCLE495 - Profilage

Syntaxe

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1		<_T>	STRING[20]	Nom d'outil de la meule		
2		<_DD>	INT	Numéro de tranchant de la meule		
3		<_SC>	REAL	Valeur de relèvement pour le contournement d'obstacles, en relatif		
4		<_F>	REAL	Avance profilage		
5		<_VARI>	INT	Type d'usinage		
				UNITES :	Type de profilage	
					1 =	Parallèle à l'axe
					2 =	Parallèle au contour
				DIZAINES :	Sens d'usinage	
					0 =	En tirant possible avec les positions de tranchant 1 à 4
					1 =	En poussant possible avec les positions de tranchant 1 à 4
					2 =	En alternance possible avec les positions de tranchant 1 à 8
					3 =	Début → fin possible avec les positions de tranchant 1 à 8
					4 =	Fin → début possible avec les positions de tranchant 1 à 8
				CENTAINES :	Direction de pénétration	
					1 =	Pénétration X- pour G18 ou Y- pour G19
					2 =	Pénétration X+ pour G18 ou Y+ pour G19
					3 =	Pénétration Z- pour G18 et pour G19
					4 =	Pénétration Z+ pour G18 et pour G19
6		<_D>	REAL	Profondeur de passe de dressage pour le type de profilage parallèle à l'axe		
7		<_DX>	REAL	Profondeur de passe de dressage X pour G18 ou Y pour G19 pour le type de profilage parallèle au contour		
8		<_DZ>	REAL	Profondeur de passe de dressage Z pour G18 et G19 pour le type de profilage parallèle au contour		
9		<S_PA>	REAL	Surépaisseur de profilage		
10		<S_N>	INT	Nombre de courses dans le programme de profilage		

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
11		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
12		<_AMODE>	INT	Mode alternatif		
				UNITES :	Sélection profilage nouveau/poursuivre	
					1 =	Nouveau
					2 =	Poursuivre
				DIZAINES :	Sélection surépaisseur de profilage	
					0 =	Du contour brut au point le plus bas du contour
					1 =	Du contour brut au point le plus haut du contour
13		<S_FW>	REAL	Angle de débattement de l'outil de dressage		
14		<S_HW>	REAL	Equerre-support de l'outil de dressage		

20.1.33 CYCLE800 - Pivotement

Syntaxe

```
CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>,
<_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)
```

Paramètres

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
1		<_FR>	INT	Mode de dégagement :	0 =	Pas de dégagement
					1 =	Dégagement axe machine Z
					2 =	Dégagement axe machine Z puis XY
					3 =	Réservé
					4 =	Dégagement maximal dans le sens de l'outil
					5 =	Dégagement relatif dans le sens de l'outil

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
2		<_TC>	STRING[32]	Nom du bloc de données de pivotement :	""	(aucun nom) en présence d'un seul bloc de données de pivotement
					"0"	Désélection du bloc de données de pivotement (suppression des frames de pivotement)
3		<_ST>	INT	Etat des transformations		
				UNITES :		
					0 =	Nouveau, suppression du plan orienté et recalcul avec les paramètres actuels
					1 =	Additif, le plan orienté est ajouté au plan orienté actif
				DIZAINES :	Asservir la pointe de l'outil oui/non (actif uniquement si la fonction est configurée lors de la MeS PIVOTEMENT)	
					0 =	Ne pas asservir la pointe de l'outil
					1 =	Asservir la pointe de l'outil (TRAORI)
				CENTAINES :	Approcher/pivoter l'outil (la fonction est affichée dans le masque de saisie PIVOTER Outil)	
					0 =	Ne pas approcher l'outil
					1 =	Approcher l'outil (de préférence fraise radiale)
					2 =	Pivoter l'outil de **tournage** (si la cinématique de l'axe B pour la technologie de tournage est configurée dans MeS Pivotement)
					3 =	Pivoter l'outil de **fraisage** (si la cinématique de l'axe B pour la technologie de tournage est configurée dans MeS Pivotement)
				MILLIERS :	Paramètre interne Pivotement en mode JOG	
				DIZAINES DE MILLIERS :	Voir paramètre direction <_DIR>	
					0 =	Pivotement "oui"
					1 =	Pivotement "non", sens "moins"[3)]
					2 =	Pivotement "non", sens "plus"[3)]
				CENTAINES DE MILLIERS :	Voir paramètre direction <_DIR>	
					0 =	Compatibilité
					1 =	Sélection du sens "moins" optimisé (uniquement pour l'interface utilisateur) [4)]
					2 =	Sélection du sens "plus" optimisé (uniquement pour l'interface utilisateur) [4)]

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
4		<_MODE> 5)	INT	Mode de pivotement : Evaluation de l'angle de pivotement et de la séquence de pivotements (codée sur bits)		
				Bit : 7 6	0 0 :	Angle de pivotement axe par axe -> voir paramètres <_A>, <_B>, <_C>
					0 1 :	Angle solide -> voir paramètres <_A>, <_B>1)
					1 0 :	Angle projeté -> voir Paramètres <_A>, <_B>, <_C> 1)
					1 1 :	Mode de pivotement axes rotatifs en direct -> voir paramètres <_A>, <_B>1)
				Bit : 5 4 3 2 1 0 (sans signification pour les angles solides)	x x x x 0 1	1re rotation _A autour de X
					x x x x 1 0	1re rotation _A autour de Y
					x x x x 1 1	1re rotation _A autour de Z
					x x 0 1 x x	2e rotation _B autour de X
					x x 1 0 x x	2e rotation _B autour de Y
					x x 1 1 x x	2e rotation _B autour de Z
					0 1 x x x x	3e rotation _C autour de X
					1 0 x x x x	3e rotation _C autour de Y
					1 1 x x x x	3e rotation _C autour de Z
5	X0	<_X0>	REAL	Point de référence X avant la rotation		
6	Y0	<_Y0>	REAL	Point de référence Y avant la rotation		
7	Z0	<_Z0>	REAL	Point de référence Z avant la rotation		
8	X(A)	<_A>	REAL	1ère rotation conforme au réglage du paramètre <_MODE>		
9	Y(B)	<_B>	REAL	2ème rotation conforme au réglage du paramètre <_MODE>		
10	Z(C)	<_C>	REAL	3ème rotation conforme au réglage du paramètre <_MODE>		
11	X1	<_X1>	REAL	Point de référence X après la rotation		
12	Y1	<_Y1>	REAL	Point de référence Y après la rotation		
13	Z1	<_Z1>	REAL	Point de référence Z après la rotation		
14	- ou +	<_DIR>	INT	Déclencher le déplacement des axes rotatifs (par défaut = -1)		
					-1 =	Positionner sur la plus petite valeur de l'axe rotatif 1 ou 2 2)
					+1 =	Positionner sur la plus grande valeur de l'axe rotatif 1 ou 2 2)
					0 =	Pivotement non (calcul du frame de pivotement uniquement) 1) 3)
15	FR	<_FR_I>	REAL	Valeur (rel) du dégagement relatif dans l'axe de l'outil		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
16		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Représentation de la valeur bêta pour orientation de l'outil	
					0 =	Valeur
					1 =	Flèche

Remarque

Si les paramètres de transfert ci-après sont programmés indirectement (en tant que paramètres), le masque de saisie n'est pas décompilé : <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

[1)] Sélection possible, si configuré dans la fonction MeS PIVOTEMENT.

[2)] Sélection possible si la référence de sens est réglée sur l'axe rotatif 1 ou 2 dans MeS PIVOTEMENT.

Aucun champ de sélection en l'absence de référence de sens

[3)]La sélection Pivotement "non" peut être masquée SD 55221 bit 0

Pivotement "non" sens "moins" correspond à <_DIR> = 0 et _ST DIZAINES DE MILLIERS = 1

Pivotement "non" sens "plus" correspond à <_DIR> = 0 et _ST DIZAINES DE MILLIERS = 2

[4)] La sélection de sens de l'axe rotatif 1 ou 2 s'effectue également lorsque l'axe rotatif se trouve dans la position polaire avec la référence de sens (valeur de position égale à zéro).

[5)] Exemple de codage : rotation axe par axe, séquence de rotation ZYX

Binaire : 00011011 ; décimal : 27

Les descripteurs d'axe XYZ correspondent aux axes géométriques du canal CN. Les rotations autour des axes XYZ peuvent être exécutées individuellement. P. ex., l'ordre de rotation autour de ZXZ n'est pas autorisé dans un appel du CYCLE800.

20.1.34 CYCLE801 - Réseau ou cadre

Syntaxe

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	X0	<_SPCA>	REAL	Point de référence pour modèles de position (réseau/cadre) du 1er axe (abs)		
2	Y0	<_SPCO>	REAL	Point de référence pour modèles de position (réseau/cadre) dans le 2e axe (abs)		
3	α0	<_STA>	REAL	Angle de rotation de base (par rapport au 1er axe)	< 0 =	Rotation dans le sens horaire
					> 0 =	Rotation dans le sens antihoraire
4	L1	<_DIS1>	REAL	Distance entre les colonnes (écart de position du 1er axe, à saisir sans signe)		
5	L2	<_DIS2>	REAL	Distance entre les lignes (écart de position du 2e axe, à saisir sans signe)		
6	N1	<_NUM1>	INT	Nombre de colonnes		
7	N2	<_NUM2>	INT	Nombre de lignes		
8		<_VARI>	INT	Type d'usinage		
				UNITES :	Modèle de position	
					0 =	Réseau
					1 =	Cadre
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
9		<_UMODE>	INT	Réservé		
10	αX	<_ANG1>	REAL	Angle de déformation par rapport au 1er axe (inclinaison des lignes par rapport au 1er axe)		
					< 0 =	Mesure dans le sens horaire (0 à -90 degrés)
					> 0 =	Mesure dans le sens antihoraire (0 à 90 degrés)
11	αY	<_ANG2>	REAL	Angle de déformation par rapport au 2e axe (inclinaison des colonnes par rapport au 2e axe)		
					< 0 =	Mesure dans le sens horaire (0 à -90 degrés)
					> 0 =	Mesure dans le sens antihoraire (0 à 90 degrés)
12		<_HIDE>	STRING [200]	Positions masquées • 198 caractères max. • Indication du numéro de position consécutif, par exemple "1,3" (les positions 1 et 3 ne sont pas exécutées)		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
13		<_NSP>	INT	Réservé		
14		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)

20.1.35 CYCLE802 - Positions quelconques

Syntaxe

```
CYCLE802(<_XA>, <_YA>, <_X0>, <_Y0>, <_X1>, <_Y1>, <_X2>, <_Y2>,
<_X3>, <_Y3>, <_X4>, <_Y4>, <_X5>, <_Y5>, <_X6>, <_Y6>, <_X7>, <_Y7>,
<_X8>, <_Y8>, <_VARI>, <_UMODE>, <_DMODE>, <S_ABA>, <S_AB0>,
<S_AB1>, <S_AB2>, <S_AB3>, <S_AB4>, <S_AB5>, <S_AB6>, <S_AB7>,
<S_AB8>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1		<_XA>	INT	Alternative pour toutes les positions X (valeur décimale à 9 caractères) Nombre de caractères : 876543210 (le caractère correspond à la position de perçage Xn)		
				Valeur du caractère :	1 =	Absolu (1ère position programmée, toujours absolue)
					2 =	Relatif
2		<_YA>	INT	Alternative pour toutes les positions Y (valeur décimale à 9 caractères) Nombre de caractères : 876543210 (le caractère correspond à la position de perçage Yn)		
				Valeur du caractère :	1 =	Absolu (1ère position programmée, toujours absolue)
					2 =	Relatif
3	X0	<_X0>	REAL	1. position X		
4	Y0	<_Y0>	REAL	1. position Y		
5	X1	<_X1>	REAL	2. position X		
6	Y1	<_Y1>	REAL	2. position Y		
7	X2	<_X2>	REAL	3. position X		

N°	Masque de paramètre	Paramètre interne	Type de données	Signification
8	Y2	<_Y2>	REAL	3. position Y
9	X3	<_X3>	REAL	4. position X
10	Y3	<_Y3>	REAL	4. position Y
11	X4	<_X4>	REAL	5. position X
12	Y4	<_Y4>	REAL	5. position Y
13	X5	<_X5>	REAL	6. position X
14	Y5	<_Y5>	REAL	6. position Y
15	X6	<_X6>	REAL	7. position X
16	Y6	<_Y6>	REAL	7. position Y
17	X7	<_X7>	REAL	8. position X
18	Y7	<_Y7>	REAL	8. position Y
19	X8	<_X8>	REAL	9. position X
20	Y8	<_Y8>	REAL	9. position Y

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
21		<_VARI>	INT	Usinage		
				CENTAINES :	(uniquement pour appel à partir de Jobshop) (pour l'instant seuls 0 et 2 sont exploités)	
					0 =	Pas de blocage de broche
					1 =	Blocage de broche uniquement pour pénétration perpendiculaire avec G00 ou G01
					2 =	Blocage de broche pendant tout l'usinage
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Modèle de position avec/sans axe rotatif – combinaison d'axes (avec <_VARI> CENTAINES DE MILLIERS)	
					0 =	XY (uniquement XY sans axe rotatif, compatibilité)
					1 =	X, Y ou Z et axe rotatif : XA, YB, ZC (1 axe rotatif avec axe géométrique autour duquel l'axe rotatif tourne)
					2 =	XY et axe rotatif : XYA, XYB, XYC (1 axe rotatif avec 1er et 2e axe géométrique, sans TRACYL)
				CENTAINES DE MILLIERS :	Axe rotatif	
					0 =	Sans axe rotatif (uniquement XY, compatibilité)
					1 =	Axe A (axe rotatif autour de X)
					2 =	Axe B (axe rotatif autour de Y)
					3 =	Axe C (axe rotatif autour de Z)
				DIZAINES DE MILLIONS + MILLIONS :	Modèle de positions avec axe rotatif – offset (pour plusieurs axes rotatifs autour du même axe ; quand l'indice est trop élevé, alors 1er axe)	
					00 =	1. Axe A, B ou C ou pour compatibilité
					01 =	2. Axe A, B ou C
					...	
					19 =	20. Axe A, B ou C
22		<_UMODE>	INT	Sélection de la broche à bloquer : (uniquement pour appel à partir de Jobshop) (appel du cycle utilisateur CUST_TECHCYC)		
					3 =	Bloquer/débloquer broche principale
					23 =	Bloquer/débloquer contre-broche

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
23		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
24		<S_ABA>	INT	Alternative pour toutes les positions AB (valeur décimale à 9 caractères) Nombre de caractères : 876543210 (le caractère correspond à la position ABn)		
				Valeur du caractère :	1 =	Absolu (1ère position programmée, toujours absolue)
					2 =	Relatif
25	A0	<S_AB0>	REAL	1. position d'axe rotatif pour modèle de position avec axe rotatif (en association avec <_VARI>)		
26	A1	<S_AB1>	REAL	2. position d'axe rotatif pour modèle de positions avec axe rotatif		
27	A2	<S_AB2>	REAL	3. position d'axe rotatif pour modèle de positions avec axe rotatif		
28	A3	<S_AB3>	REAL	4. position d'axe rotatif pour modèle de positions avec axe rotatif		
29	A4	<S_AB4>	REAL	5. position d'axe rotatif pour modèle de positions avec axe rotatif		
30	A5	<S_AB5>	REAL	6. position d'axe rotatif pour modèle de positions avec axe rotatif		
31	A6	<S_AB6>	REAL	7. position d'axe rotatif pour modèle de positions avec axe rotatif		
32	A7	<S_AB7>	REAL	8. position d'axe rotatif pour modèle de positions avec axe rotatif		
33	A8	<S_AB8>	REAL	9. position d'axe rotatif pour modèle de positions avec axe rotatif		

Remarque

Les positions inutiles des paramètres X1/Y1/A1 à X8/Y8/A8 peuvent être ignorées. Toutefois, les valeurs alternatives pour <_XA>, <_YA> et <S_ABA> doivent toujours être indiquées pour l'ensemble des 9 positions.

Pour le modèle de positions XA, YB ou ZC (un axe géométrique et un axe rotatif), l'axe du plan d'usinage dans lequel il n'y a pas de déplacement via le modèle de positions (Y pour G17 et XA) doit être positionné avant l'appel du cycle.

20.1.36 CYCLE830 - Perçage profond 2

Syntaxe

```
CYCLE830(<RTP>, <RFP>, <SDIS>, <_DP>, <FDEP>, <_DAM>, <DTB>, <DTS>,
<FRF>, <VARI>, <_MDEP>, <_VRT>, <_DTD>, <_DIS1>, <S_FP>, <S_SDAC2>,
<S_SV2>, <S_FB>, <_SDAC>, <_SV1>, <S_SPOS>, <S_ZA>, <S_FA>, <S_ZP>,
```

```
<S_FS>, <S_ZS1>, <S_ZS2>, <S_N>, <S_ZD>, <S_FD>, <S_FR>, <S_SDAC3>,
<S_SV3>, <S_CON>, <S_COFF>, <_GMODE>, <_DMODE>, <_AMODE>,
<S_AMODE2>, <S_AMODE3>, <S_ZPV>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<RTP>	REAL	Plan de retrait (abs)		
2	Z0	<RFP>	REAL	Point de référence (abs)		
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, sans signe)		
4	Z1	<_DP>	REAL	Profondeur de perçage finale abs/rel (voir <_AMODE> UNITES)		
5	D	<FDEP>	REAL	1re profondeur de perçage pour bris de copeaux / débourrage absolue ou relative par rapport au point de référence avec pointage / sans pointage ou par rapport à la profondeur de perçage pilote (voir <_AMODE> DIZAINES DE MILLIERS)		
6	DF	<_DAM>	REAL	Valeur/pourcentage pour chaque passe suivante valeur ou pourcentage de dégression (voir <_AMODE> CENTAINES DE MILLIERS)		
7	DTB	<DTB>	REAL	Arrêt temporisé à chaque profondeur de perçage (voir <_AMODE> DIZAINES)		
8	DTS	<DTS>	REAL	Arrêt temporisé pour le débourrage au point de départ (voir <_AMODE> CENTAINES)		
9	FD1	<FRF>	REAL	Valeur en pourcentage de l'avance pour la première passe (voir <_AMODE> DIZAINES DE MILLIONS)		
10		<VARI>	INT	Usinage		
				UNITES :	Bris de copeaux / débourrage	
					0 =	En une passe
					1 =	Bris de copeaux
					2 =	Débourrage
					3 =	Bris de copeaux et débourrage
				DIZAINES :	Retrait lors du débourrage	
					0 =	A la profondeur de perçage pilote
					1 =	A la distance de sécurité
				CENTAINES :	Attaque souple	
					0 =	non
					1 =	Oui
				MILLIERS :	Perçage débouchant	
					0 =	non
					1 =	Oui
				DIZAINES DE MILLIERS :	Pointage / perçage pilote	
					0 =	Sans pointage
					1 =	Avec pointage
					2 =	Avec trou pilote
				CENTAINES DE MILLIERS :	Retrait	
					0 =	A la profondeur de perçage pilote
					1 =	Jusqu'au plan de retrait

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
11	V1	<_MDEP>	REAL	Profondeur de passe minimale relative (uniquement avec pourcentage pour dégression)		
12	V2	<_VRT>	REAL	Valeur de retrait relative après chaque usinage (uniquement pour bris de copeaux)		
					0 =	Valeur par défaut 1 mm
					> 0 =	Valeur de retrait variable
13	DT	<_DTD>	REAL	Arrêt temporisé à la profondeur de perçage finale (voir <_AMODE> MILLIERS)		
14	V3	<_DIS1>	REAL	Précote d'arrêt relative uniquement pour débourrage (voir <_AMODE> MILLIONS)		
15	FP	<S_FP>	REAL	Avance pour l'accostage du perçage pilote comme valeur ou en % (en association avec <S_AMODE2> CENTAINES)		
16		<S_SDAC2>	INT	Sens de rotation de la broche lors de l'accostage		
					3 =	M3
					4 =	M4
					5 =	M5 (par défaut)
17	SP	<S_SV2>	REAL	Accostage avec	Vitesse de rotation de broche constante (voir <S_AMODE2> DIZAINES DE MILLIONS)	
	V4				Vitesse de coupe constante	
				Vitesse de rotation de broche en % de la vitesse de rotation de perçage		
18	F	<S_FB>	REAL	Avance de perçage (voir <S_AMODE2> UNITES)		
19		<_SDAC>	REAL	Sens de rotation de broche lors du perçage		
					3 =	M3
					4 =	M4
20	S	<_SV1>	REAL	Perçage avec	Vitesse de rotation de broche constante (voir <S_AMODE2> MILLIONS)	
	V5				Vitesse de coupe constante	
21	SPOS	<S_SPOS>	REAL	Position de la broche, uniquement pour accostage avec M5		
22	ZA	<S_ZA>	REAL	Profondeur de pointage relative par rapport au point de référence ou absolue (voir <S_AMODE3> UNITES)		
23	FA	<S_FA>	REAL	Avance de pointage comme valeur ou en % (en association avec <S_AMODE2> DIZAINES)		
24	ZP	<S_ZP>	REAL	Perçage pilote relatif par rapport au point de référence, absolu ou facteur du diamètre de perçage (voir <S_AMODE3> DIZAINES)		
25	FS	<S_FS>	REAL	Avance de l'attaque comme valeur ou en % (en association avec <S_AMODE2> MILLIERS)		
26	ZS1	<S_ZS1>	REAL	Profondeur de chaque attaque avec avance constante (rel)		
27	ZS2	<S_ZS2>	REAL	Profondeur de chaque attaque pour augmentation d'avance (rel)		
28	N	<S_N>	INT	Nombre des coups de bris de copeaux avant chaque débourrage		
29	ZD	<S_ZD>	REAL	Profondeur de perçage restante relative par rapport à la profondeur de perçage finale ou absolue (en association avec <S_AMODE3> CENTAINES)		
30	FD	<S_FD>	REAL	Avance de perçage restante comme valeur ou en % (en association avec <S_AMODE2> DIZAINES DE MILLIERS)		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
31	FR	<S_FR>	REAL	Avance de retrait (en association avec <S_AMODE2> CENTAINES DE MILLIERS)		
32		<S_SDAC3>	INT	Sens de rotation de broche lors du retrait		
					3 =	M3
					4 =	M4
					5 =	M5
33	SR	<S_SV3>	REAL	Retrait avec	Vitesse de rotation de broche constante (voir <S_AMODE2> CENTAINES DE MILLIONS)	
	V6				Vitesse de coupe constante	
				Vitesse de rotation de broche en % de la vitesse de rotation de perçage		
34	Liquide d'arrosage en marche	<S_CON>	STRING[10]	Arrosage en marche, instruction M ou appel de sous-programme		
35	Arrêt de l'arrosage	<S_COFF>	STRING[10]	Arrêt de l'arrosage, instruction M ou appel de sous-programme		
36		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Profondeur de perçage par rapport à pointe/corps	
					0 =	Pointe
					1 =	Corps
37		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Entrée : Complète
					1 =	Entrée : Simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
38		<_AMODE>	INT	Mode alternatif 1		
				UNITES :	Profondeur de perçage = profondeur de perçage finale Z1 abs/rel	
					0 =	Relatif
					1 =	Absolu
				DIZAINES :	Temporisation à chaque profondeur de perçage DTB en secondes/tours	
					0 =	En secondes
					1 =	En tours
				CENTAINES :	Temporisation pour le débourrage DTS en secondes/tours	
					0 =	En secondes
					1 =	En tours
				MILLIERS :	Arrêt temporisé à la profondeur de perçage finale DT en secondes/tours	
					0 =	En secondes
					1 =	En tours
				DIZAINES DE MILLIERS :	1re profondeur de perçage D abs/rel	
					0 =	Relatif
					1 =	Absolu
				CENTAINES DE MILLIERS :	Valeur/pourcentage DF pour chaque passe suivante (dégression)	
					0 =	Valeur absolue
					1 =	Pourcentage (0,001 à 100 %)
				MILLIONS :	Précote d'arrêt V3 automatique/manuel	
					0 =	Automatique (calculée dans le cycle)
					1 =	Manuel (valeur programmée)

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
39		<S_AMODE2>	INT	Mode alternatif 2		
				UNITES :	UNITES : Avance de perçage F	
					0 =	F/min
					1 =	F/tr
				DIZAINES :	Evaluation de l'avance de pointage FA	
					0 =	En % de l'avance de perçage
					1 =	F/min
					2 =	F/tr
				CENTAINES :	Evaluation de l'avance pour l'accostage dans le perçage pilote FP	
					0 =	En % de l'avance de perçage
					1 =	F/min
					2 =	F/tr
				MILLIERS :	Evaluation de l'avance d'attaque FS	
					0 =	En % de l'avance de perçage
					1 =	F/min
					2 =	F/tr
				DIZAINES DE MILLIERS :	Evaluation de l'avance de perçage débouchant FD	
					0 =	En % de l'avance de perçage
					1 =	F/min
					2 =	F/tr
				CENTAINES DE MILLIERS :	Avance de retrait FR	
					0 =	F/min
					1 =	Marche rapide
				MILLIONS :	Perçage - vitesse de rotation de broche / vitesse de coupe (S/V5)	
					0 =	Vitesse de rotation de broche constante
					1 =	Vitesse de coupe constante
				DIZAINES DE MILLIONS :	Accostage avec vitesse de rotation de broche / vitesse de coupe (SP/V4)	
					0 =	Vitesse de rotation de broche constante
					1 =	Vitesse de coupe constante
					2 =	Vitesse de rotation de broche en % de la vitesse de rotation de perçage
				CENTAINES DE MILLIONS :	Retrait - vitesse de rotation de broche / vitesse de coupe (SR/V6)	
					0 =	Vitesse de rotation de broche constante
					1 =	Vitesse de coupe constante

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
					2 =	Vitesse de rotation de broche en % de la vitesse de rotation de perçage
40		<S_AMODE3>	INT	Mode alternatif 3		
				UNITES :	Profondeur de pointage ZA abs/rel	
					0 =	Relatif
					1 =	Absolu
				DIZAINES :	Profondeur du perçage pilote ZP	
					0 =	Relatif
					1 =	Absolu
					2 =	Facteur du diamètre de perçage
				CENTAINES :	Profondeur de perçage restante ZD rel/ abs	
					0 =	Relatif
					1 =	Absolu
41	ZPV	<S_ZPV>	REAL	Précote d'arrêt relative sur profondeur de perçage pilote		

20.1.37 CYCLE832 - Réglages High Speed

Syntaxe

```
CYCLE832(<S_TOL>, <S_TOLM>, <S_OTOL>)
```

Remarque

Le CYCLE832 ne décharge pas le constructeur de machines des tâches d'optimisation nécessaires lors de la mise en service de la machine. Ceci concerne l'optimisation des axes intervenant dans l'usinage et les réglages de la NCU (commande anticipatrice, limitation des à-coups, etc.).

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	Tolérance	<S_TOL>	REAL	Tolérance de contour La tolérance de contour correspond à la tolérance des axes géométriques.		
2		<S_TOLM>	INT	Type d'usinage (technologique)		
				UNITES :		
					0 =	Désactivation
					1 =	Finition (Finish)
					2 =	Préfinition (Semifinish)
					3 =	Ebauchage (Rough)
				DIZAINES :		
					0 =	Compatibilité[1] ou aucune tolérance d'orientation
					1 =	Tolérance d'orientation dans le paramètre <S_OTOL>
				CENTAINES ... CENTAINES DE MILLIERS	occupé pour cause de Compatibilité	
				MILLIONS :		
					0 =	Compatibilité. La meilleure fonction de moulage disponible est automatiquement utilisée : • Option Top Surface inactive : ⇒ Advanced Surface • Option Top Surface active : ⇒ Top Surface avec lissage
					1 =	Top Surface sans lissage
					2 =	Top Surface avec lissage
3	Tolérance ORI	<S_OTOL>	REAL	Tolérance d'orientation ou identificateur de version CYCLE832 Paramètre de tolérance pour l'orientation de l'outil. Nécessaire pour l'exécution d'un programme d'usinage utilisant des vitesses élevées sur des machines avec transformation d'orientation dynamique (p. ex. usinage 5 axes). La programmation du paramètre <S_OTOL> est **obligatoire**. Cette remarque s'applique également aux applications exécutées sur des machines 3 axes en présence de programmes sans orientation de l'outil (<S_OTOL> = 1).		

[1] Tolérance d'orientation dérivée de la tolérance de contour multipliée par le facteur provenant des données de réglage de cycle SD55441 à SD55443.
Bibliographie :
Manuel de mise en service Logiciel de base et logiciel d'exploitation ; SINUMERIK Operate (IM9), chapitre "Configuration de la fonction Réglages High Speed (CYCLE832)"

Saisie de texte en clair

Pour améliorer la lisibilité de l'appel de cycle, le paramètre <S_TOLM> (type d'usinage) peut également être saisi en clair. Les textes en clair sont indépendants de la langue. Les saisies suivantes sont autorisées :

_OFF	pour	0	Désactivation
_FINISH	pour	1	Finition
_SEMIFIN	pour	2	Préfinition
_ROUGH	pour	3	Ebauche
_ORI_FINISH	pour	11	Finition avec saisie d'une tolérance d'orientation
_ORI_SEMIFIN	pour	12	Préfinition avec saisie d'une tolérance d'orientation
_ORI_ROUGH	pour	13	Ebauchage avec saisie d'une tolérance d'orientation
_TOP_SURFACE_SMOOTH_OFF	pour	1000000	Top Surface sans lissage
_TOP_SURFACE_SMOOTH_ON	pour	2000000	Top Surface avec lissage

Pour la saisie de texte en clair avec Top Surface, les textes en clair sont combinés comme dans l'exemple suivant :

```
CYCLE832(0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

Remarque

Les textes en clair s'appuient sur les noms de fonction du groupe G 59 (mode dynamique avec interpolation de trajectoire). Grâce à ces noms explicites, les machines 3 axes et les machines avec transformation d'orientation multi-axe (TRAORI) sont clairement séparées dans l'application.

Désélection de CYCLE832

Lorsque CYCLE832 est désélectionné, le paramètre <S_TOL> doit être transmis avec la valeur zéro.

Exemple : `CYCLE832(0,0,1)`

La syntaxe `CYCLE832()` est également autorisée pour la désélection de CYCLE832.

Exemples

Exemple 1 : CYCLE832 sur machine 3 axes sans transformation d'orientation

a) Appel de cycle avec saisie de texte en clair

Code de programme	Commentaire
G710	; Le système de mesure est le système métrique

Code de programme	Commentaire
CYCLE832(0.004,_FINISH,1)	; Appel du CYCLE832 avec : tolérance de contour = 0,004 mm, type d'usinage : finition
...	; Exécution d'un programme d'usinage utilisant des vitesses élevées

b) Appel du cycle sans saisie de texte en clair

Code de programme	Commentaire
G710	; cf. supra
CYCLE832(0.004,1,1)	; cf. supra
...	; cf. supra

Exemple 2 : CYCLE832 sur machine 5 axes avec transformation d'orientation

a) Appel de cycle et désélection avec saisie de texte en clair

Code de programme	Commentaire
G710	; Le système de mesure est le système métrique
TRAORI	; Activation de la transformation d'orientation
CYCLE832(0.3,_ORI_ROUGH,0.8)	; Appel du CYCLE832 avec : tolérance de contour = 0,3 mm, type d'usinage : ébauchage avec saisie d'une tolérance d'orientation, tolérance d'orientation = 0,8 degré
...	; Exécution d'un programme d'usinage utilisant des vitesses élevées
CYCLE832(0,_OFF,1)	; Tolérance de contour = 0, type d'usinage : Désélection de CYCLE832, Tolérance d'orientation = 0 degré

b) Appel du cycle et désélection sans saisie de texte en clair

Code de programme	Commentaire
G710	; cf. supra
TRAORI	; cf. supra
CYCLE832(0.3,13,0.8)	; cf. supra
...	; cf. supra
CYCLE832(0,0,1)	; cf. supra

20.1.38 CYCLE840 - Taraudage avec porte-taraud compensateur

Syntaxe

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification			
1	RP	<RTP>	REAL	Plan de retrait (abs)			
2	Z0	<RFP>	REAL	Point de référence (abs)			
3	SC	<SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)			
4	Z1	<DP>	REAL	Profondeur de perçage (abs), voir <_AMODE>			
5	Z1	<DPR>	REAL	Profondeur de perçage (rel), voir <_AMODE>			
6	DT	<DTB>	REAL	Arrêt temporisé à la profondeur de perçage / à la distance de sécurité après le retrait, en secondes, voir <ENC>			
7		<SDR>	INT	Sens de rotation pour le retrait			
8	SDE	<SDAC>	INT	Sens de rotation après fin du cycle			
9		<ENC>	INT	Taraudage avec capteur de broche (G33) / taraudage sans capteur de broche (G63)			
					0 =	Avec capteur de broche	- Pas à partir de <MPIT>/<PIT> - sans DT
					20 =	Avec capteur de broche	- Pas à partir de <MPIT>/<PIT> - avec DT après retrait à la distance de sécurité
					11 =	Sans capteur de broche	- Pas à partir de <MPIT>/<PIT> - avec DT à la profondeur de perçage
					1 =	Sans capteur de broche	- Pas provenant de l'avance programmée - avec DT à la profondeur de perçage (avance = vitesse de rotation pas)
10		<MPIT>	REAL	Taille de filetage uniquement pour "ISO métrique" (le pas est calculé en interne pendant l'exécution) Plage de valeurs : 3 à 48 (pour M3 à M48), alternativement à <PIT>			
11		<PIT>	REAL	Pas de vis en tant que valeur, unité de mesure voir<_PITA> Plage de valeurs : > 0, en alternative à MPIT			

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification			
12		<_AXN>	INT	Axe de perçage	0 =	3. axe géométrique	
					1 =	1. axe géométrique	
					2 =	2. axe géométrique	
					≥ 3 =	3. axe géométrique	
13		<_PITA>	INT	Unité de mesure du pas de vis (évaluation <PIT> et <MPIT>)			
					0 =	Pas en mm	- Evaluation <MPIT>/<PIT>
					1 =	Pas en mm	- Evaluation <PIT>
					2 =	Pas en TPI	- Evaluation <PIT> (nombre de filets par inch)
					3 =	Pas en inch	- Evaluation <PIT>
					4 =	MODULE	- Evaluation <PIT>
14		<_TECHNO>	INT	Technologie[1)]			
				UNITES :	Type d'arrêt précis		
					0 =	Type d'arrêt précis comme avant appel de cycle actif	
					1 =	Arrêt précis G601	
					2 =	Arrêt précis G602	
					3 =	Arrêt précis G603	
				DIZAINES :	Commande anticipatrice		
					0 =	Avec/sans commande anticipatrice actif comme avant l'appel de cycle	
					1 =	Avec commande anticipatrice FFWON	
					2 =	Sans commande anticipatrice FFWOF	
15		<_PITM>	STRING[15]	Chaîne de caractères servant de mémento pour la saisie du pas de vis[2]			
16		<_PTAB>	STRING[5]	Chaîne de caractères pour table de filetages ("", "ISO", "BSW", "BSP", "UNC")[2)]			
17		<_PTABA>	STRING[20]	Chaîne de caractères pour la sélection dans la table de filetages (p. ex. "M 10", "M 12"...)[2)]			
18		<_GMODE>	INT	Réservé			

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
19		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :	Mode de compatibilité (uniquement pour décompilation du masque de saisie), lorsque PM 52216 bit 0 = 1[1)]	
					0 =	Les paramètres technologiques sont affichés (compatibilité) : les paramètres TECHNO prennent effet
					1 =	Les paramètres technologiques ne sont pas affichés : la technologie "programmée avant l'appel de cycle" reste effective
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée
20		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de perçage Z1 (abs/rel)	
					0 =	Compatibilité issue de la programmation <DP>/<DPR>
					1 =	Relatif
					2 =	Absolu

[1)] Les champs Technologie peuvent être masqués, en fonction de la donnée de réglage SD52216 MCS_FUNCTION_MASK_DRILL

[2)] Les paramètres 15, 16 et 17 sont utilisés uniquement lors de la sélection du filetage dans les tables de filetages du masque de saisie. L'accès aux tables de filetages par la définition de cycle n'est pas possible pendant l'exécution du cycle.

20.1.39 CYCLE899 - Fraisage de rainure borgne

Syntaxe

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>,
<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>, <_ZFS>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1	RP	<_RTP>	REAL	Plan de retrait (abs)		
2	Z0	<_RFP>	REAL	Point de référence de l'axe outil (abs)		
3	SC	<_SDIS>	REAL	Distance de sécurité (ajoutée au point de référence, à saisir sans signe)		
4	Z1	<_DP>	REAL	Profondeur de rainure (abs/rel), voir <_AMODE>		
5	L	<_LENG>	REAL	Longueur de la rainure (rel)		
6	W	<_WID>	REAL	Largeur de la rainure (rel)		
7	X0	<_PA>	REAL	Point de référence / position de départ 1er axe (abs)		
8	Y0	<_PO>	REAL	Point de référence / position de départ 2e axe (abs)		
9	α0	<_STA>	REAL	Angle de rotation par rapport au 1er axe		
10	DZ	<_MID>	REAL	Profondeur de passe maximale (rel), uniquement pour le fraisage en hélice		
11	DXY	<_MIDA>	REAL	Profondeur de passe maximale dans le plan, voir <_AMODE>		
12	UXY	<_FAL>	REAL	Surépaisseur de finition dans le plan		
13	UZ	<_FALD>	REAL	Surépaisseur de finition au fond		
14	F	<_FFP1>	REAL	Avance		
15		<_CDIR>	INT	Sens de fraisage		
				UNITES :		
					0 =	En avalant
					1 =	En opposition
					4 =	En alternance
16		<_VARI>	INT	Usinage		
				UNITES :		
					1 =	Ebauchage
					2 =	Finition
					3 =	Finition du fond
					4 =	Finition du bord
					5 =	Préfinition
					6 =	Chanfreinage
				DIZAINES :	Réservé	
				CENTAINES :	Réservé	
				MILLIERS :		
					1 =	Fraisage en hélice
					2 =	Fraisage en plongée

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
17		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération / calcul du point de départ uniquement	
					1 =	Usinage normal
				MILLIERS :	Cotation à partir du milieu / de l'arête	
					0 =	Cotation à partir du milieu
					1 =	Cotation à partir de l'arête "gauche" (sens "-" du 1er axe)
					2 =	Cotation à partir de l'arête "droite" (sens "+" du 1er axe)
18		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	---	Réservé
				CENTAINES :	---	Réservé
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée
19		<_AMODE>	INT	Mode alternatif		
				UNITES :	Profondeur de rainure Z1	
					0 =	Absolu
					1 =	Relatif
				DIZAINES :	Unité pour la profondeur de passe dans le plan (<_MIDA>) DXY	
					0 =	mm
					1 =	% du diamètre de l'outil
				CENTAINES :	Profondeur de pénétration lors du chanfreinage ZFS	
					0 =	Absolu
					1 =	Relatif
20		<_UMODE>	INT	Réservé		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
21	FS	<_FS>	REAL	Largeur de chanfrein (rel)
22	ZFS	<_ZFS>	REAL	Profondeur de pénétration (pointe de l'outil) lors du chanfreinage (abs/rel), voir <_AMODE>

20.1.40 CYCLE930 - Gorge

Syntaxe

```
CYCLE930(<_SPD>, <_SPL>, <_WIDG>, <_WIDG2>, <_DIAG>, <_DIAG2>,
<_STA>, <_ANG1>, <_ANG2>, <_RCO1>, <_RCI1>, <_RCI2>, <_RCO2>,
<_FAL>, <_IDEP1>, <_SDIS>, <_VARI>, <_DN>, <_NUM>, <_DBH>, <_FF1>,
<_NR>, <_FALX>, <_FALZ>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
1	X0	<_SPD>	REAL	Point de référence dans l'axe transversal (toujours diamètre)
2	Z0	<_SPL>	REAL	Point de référence dans l'axe longitudinal
3	B1	<_WIDG>	REAL	Largeur de gorge en bas
4	B2	<_WIDG2>	REAL	Largeur de gorge en haut (uniquement pour la surface)
5	T1	<_DIAG>	REAL	Profondeur de gorge au point de référence, pour abs et usinage longitudinal = diamètre, sinon rel
6	T2	<_DIAG2>	REAL	Profondeur de gorge par rapport au point de référence (uniquement pour la surface), pour abs et usinage longitudinal = diamètre, sinon rel
7	α0	<_STA>	REAL	Angle des obliques (-180 ≤ <_STA> ≤ 180)
8	α1	<_ANG1>	REAL	Angle de flanc 1 (0 ≤ <_ANG1> < 90) sur le côté de la gorge déterminé par le point de référence
9	α2	<_ANG2>	REAL	Angle de flanc 2 (0 ≤ <_ANG2> < 90) par rapport au point de référence
10	R1/FS1	<_RCO1>	REAL	Rayon d'arrondi ou largeur de chanfrein 1, à l'extérieur du point de référence
11	R2/FS2	<_RCI1>	REAL	Rayon d'arrondi ou largeur de chanfrein 2, à l'intérieur du point de référence
12	R3/FS3	<_RCI2>	REAL	Rayon d'arrondi ou largeur de chanfrein 3, à l'intérieur par rapport au point de référence
13	R4/FS4	<_RCO2>	REAL	Rayon d'arrondi ou largeur de chanfrein 4, à l'extérieur par rapport au point de référence
14	U	<_FAL>	REAL	Surépaisseur de finition en X et Z, voir <_VARI> (DIZAINES DE MILLIERS) (à saisir sans signe)

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
15	D	<_IDEP1>	REAL	Profondeur de passe maximale lors de la plongée (à saisir sans signe)		
					0 =	1. passe directement sur toute la profondeur
					> 0 =	1. passe <_IDEP1>, 2e passe 2 · <_IDEP1>, etc.
16	SC	<_SDIS>	REAL	Distance de sécurité (à saisir sans signe)		
17		<_VARI>	INT	Type d'usinage		
				UNITES :	Réservé	
				DIZAINES :	Usinage technologique	
					1 =	Ebauchage
					2 =	Finition
					3 =	Ebauchage et finition
				CENTAINES :	Position longitudinale/transversale extérieure/intérieure +Z/+Z ou +X/-X	
					1 =	Longitudinal/extérieur +Z
					2 =	Transversal/intérieur -X
					3 =	Longitudinal/intérieur +Z
					4 =	Transversal/intérieur +X
					5 =	Longitudinal/extérieur -Z
					6 =	Transversal/extérieur -X
					7 =	Longitudinal/intérieur -Z
					8 =	Transversal/extérieur +X
				MILLIERS :	Position du point de référence	
					0 =	Point de référence supérieur
					1 =	Point de référence inférieur
				DIZAINES DE MILLIERS :	Définition de la prise d'effet des surépaisseurs de finition	
					0 =	Surépaisseur de finition U parallèle au contour
					1 =	Surépaisseurs de finition UX et UZ séparées
18		<_DN>	INT	Numéro D pour 2e tranchant de l'outil		
					> 0 =	Numéro D pour la correction d'outil du 2e tranchant de l'outil à plonger
					0 =	Pas de 2e tranchant programmé
19	N	<_NUM>	INT	Nombre de gorges (0 = 1 gorge)		
20	DP	<_DBH>	REAL	Distance entre gorges (requis uniquement lorsque <_NUM> > 1)		
21	F	<_FF1>	REAL	Avance		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
22		<_NR>	INT	Identificateur de la forme de la gorge (correspond à la touche logicielle verticale de sélection de la forme)		
					0 =	Flancs de 90° sans chanfreins/arrondis
					1 =	Flancs obliques avec chanfreins/arrondis (sans α0)
					2 =	Comme 1, mais sur cône (avec α0)
23	UX	<_FALX>	REAL	Surépaisseur de finition dans l'axe X, voir <_VARI> (DIZAINES DE MILLIERS) (à saisir sans signe)		
24	UZ	<_FALZ>	REAL	Surépaisseur de finition dans l'axe Z, voir <_VARI> (DIZAINES DE MILLIERS) (à saisir sans signe)		
25		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
26		<_AMODE>	INT	Mode alternatif		
				UNITES :	Cotation de la profondeur (uniquement pour la surface)	
					0 =	Au point de référence
					1 =	Par rapport au point de référence
				DIZAINES :	Profondeur	
					0 =	Absolu
					1 =	Relatif
				CENTAINES :	Cotation de la largeur (uniquement pour la surface)	
					0 =	Sur diamètre externe (en haut)
					1 =	Sur diamètre interne (en bas)
				MILLIERS :	Rayon/chanfrein 1 (<_RCO1>)	
					0 =	Rayon
					1 =	Chanfrein
				DIZAINES DE MILLIERS :	Rayon/chanfrein 2 (<_RCI1>)	
					0 =	Rayon
					1 =	Chanfrein
				CENTAINES DE MILLIERS :	Rayon/chanfrein 3 (<_RCI2>)	
					0 =	Rayon
					1 =	Chanfrein
				MILLIONS :	Rayon/chanfrein 4 (<_RCO2>)	
					0 =	Rayon
					1 =	Chanfrein

20.1.41 CYCLE940 - Formes de dégagement

Le cycle CYCLE940 permet de programmer différents dégagements. Ceux-ci peuvent différer considérablement les uns des autres en fonction du paramétrage.

Les colonnes supplémentaires du tableau indiquent quel paramètre est requis pour chaque forme de dégagement. Elles correspondent aux touches logicielles de sélection verticales du masque de cycle :

- E : dégagement forme E
- F : dégagement forme F
- A-D : dégagement filet DIN (formes A-D)
- T : dégagement filet (définition libre de la forme)

Syntaxe

```
CYCLE940(<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>, <_VARI>,
<_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>, <_FAL>,
<_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètres	Paramètre interne	Type de données	Progr. pour la forme				Signification		
				E	F	A-D	T			
1	X0	<_SPD>	REAL	x	x	x	x	Point de référence dans l'axe transversal (toujours diamètre)		
2	Z0	<_SPL>	REAL	x	x	x	x	Point de référence de l'axe longitudinal (abs)		
3	FORM	<_FORM>	CHAR	x	x	x	x	Forme du dégagement (en majuscules, p. ex. "T") Sélection, table à partir de laquelle les valeurs de dégagement doivent être prises		
									A =	Extérieur, référence DIN76, A = normal
									B =	Extérieur, référence DIN76, B = court
									C =	Intérieur, référence DIN76, C = normal
									D =	Intérieur, référence DIN76, D = court
									E =	Référence DIN509
									F =	Référence DIN509
									T =	Forme libre
4	LAGE	<_LAGE>	INT	x	x	x	x	Position du dégagement (parallèle Z)	0 =	Extérieur +Z : ____|
									1 =	Extérieur -Z : |____/
									2 =	Intérieur +Z : /-----|
									3 =	Intérieur -Z : |-----\
5	SC	<_SDIS>		x	x	x	x	Distance de sécurité (rel)		
6	F	<_FFP>		x	x	x	x	Avance d'usinage (mm/tr)		
7		<_VARI>	INT	-	-	x	x	Type d'usinage		
								UNITES :	Usinage	
									1 =	Ebauchage
									2 =	Finition
									3 =	Ebauchage + finition
								DIZAINES :	Stratégie d'usinage	
									0 =	Parallèle au contour
									1 =	Longitudinal
				Les dégagements de forme E et F sont toujours usinés en une seule passe, comme pour la finition.						
8	X1	<_EPD>		x	x	-	-	Surépaisseur X (abs/rel), voir <_AMODE>		
				-	-	-	x	Profondeur de dégagement (abs/rel), voir <_AMODE>		
9	Z1	<_EPL>		-	x	-	-	Surépaisseur Z		
				-	-	-	x	Largeur de dégagement (abs/rel), voir <_AMODE>		
10	R1	<_R1>		-	-	-	x	Rayon d'arrondi sur les obliques		
11	R2	<_R2>		-	-	-	x	Rayon d'arrondi dans le coin		
12	α	<_STA>		-	-	x	x	Angle de pénétration		

N°	Masque de paramètres	Paramètre interne	Type de données	Progr. pour la forme				Signification			
13	VX	<_VRT>		x	x	-	-	Chariotage transversal X (abs/rel), voir <_AMODE>			
				-	-	x	x	Chariotage transversal X pour finition (abs/rel), voir <_AMODE>			
14	D	<_MID>		-	-	x	x	Profondeur de passe			
15	U	<_FAL>		-	-	x	x	Surépaisseur de finition parallèle au contour, voir <_AMODE>			
16	UX	<_FALX>		-	-	x	x	Surépaisseur de finition X			
17	UZ	<_FALZ>		-	-	x	x	Surépaisseur de finition Z			
18	P	<_PITI>	INT	-	-	x	-	Sélection du pas, forme A-D, correspond à M1 ... M68			
								0 = 0,20 1 = 0,25 2 = 0,30 3 = 0,35 4 = 0,40 5 = 0,45	6 = 0,50 7 = 0,60 8 = 0,70 9 = 0,75 10 = 0,80 11 = 1,00	12 = 1,25 13 = 1,50 14 = 1,75 15 = 2,00 16 = 2,50 17 = 3,00	18 = 3,50 19 = 4,00 20 = 4,50 21 = 5,00 22 = 5,50 23 = 6,00
				x	x	-	-	Sélection rayon/profondeur, forme E, F			
								0 = 0,6 · 0,3 1 = 1,0 · 0,4 2 = 1,0 · 0,2 3 = 1,6 ··0,3	4 = 2,5 · 0,4 5 = 4,0 · 0,5 6 = 0,4 · 0,2 7 = 0,6 · 0,2	8 = 0,1 · 0,1 9 = 0,2 ··0,1	
19		<_PTAB>	STRING [5]					Chaîne de caractères pour table de filetages ("", "ISO", "BSW", "BSP", "UNC") (réservé à l'interface)			
20		<_PTABA>	STRING [20]					Chaîne de caractères pour la sélection dans la table de filetages (p. ex. "M 10", "M 12"...) (réservé à l'interface)			
21		<_DMODE>	INT					Mode d'affichage			
				x	x	x	x	UNITES :	Plan d'usinage G17/G18/G19		
									0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif	
									1 =	G17 (actif uniquement dans le cycle)	
									2 =	G18 (actif uniquement dans le cycle)	
									3 =	G19 (actif uniquement dans le cycle)	

Nº	Masque de paramètres	Paramètre interne	Type de données	Progr. pour la forme				Signification		
22		<_AMODE>	INT					Mode alternatif		
				x	x	-	x	UNITES :	Paramètre <_EPD> surépaisseur X ou profondeur de dégagement	
									0 =	Absolu (toujours diamètre)
									1 =	Relatif
				x	x	-	x	DIZAINES :	Paramètre <_EPL> surépaisseur Z ou largeur de dégagement	
									0 =	Absolu
									1 =	Relatif
				x	x	x	x	CENTAINES :	Paramètre <_VRT> chariotage transversal X	
									0 =	Absolu (toujours diamètre)
									1 =	Relatif
				-	-	x	x	MILLIERS :	Surépaisseur de finition	
									0 =	Surépaisseur de finition parallèle au contour (<_FAL>)
									1 =	Surépaisseur de finition distincte (<_FALX>/<_FALZ>)

20.1.42 CYCLE951 - Chariotage

Syntaxe

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>,
<_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification
1	X0	<_SPD>	REAL	Point de référence (abs, toujours diamètre)
2	Z0	<_SPL>	REAL	Point de référence (abs)
3	X1	<_EPD>	REAL	Point final
4	Z1	<_EPL>	REAL	Point final
5	XM α1 α2	<_ZPD>	REAL	Point intermédiaire, voir <_DMODE> (DIZAINES)
6	ZM α1 α2	<_ZPL>	REAL	Point intermédiaire, voir <_DMODE> (DIZAINES)

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
7	Position	<_LAGE>	INT	Position du coin de chariotage	0 =	Extérieur/arrière
					1 =	Extérieur/avant
					2 =	Intérieur/arrière
					3 =	Intérieur/avant
8	D	<_MID>	REAL	Profondeur de passe maximale lors de la pénétration		
9	UX	<_FALX>	REAL	Surépaisseur de finition en X		
10	UZ	<_FALZ>	REAL	Surépaisseur de finition en Z		
11		<_VARI>	INT	Type d'usinage		
				UNITES :	Direction de chariotage (longitudinale ou transversale) dans le système de coordonnées	
					1 =	Longitudinal
					2 =	Transversal
				DIZAINES :		
					1 =	Ebauchage jusqu'à surépaisseur de finition
					2 =	Finition
				CENTAINES :	Réservé	
				MILLIERS :	Réservé	
				DIZAINES DE MILLIERS :	Réservé	
12	R1/FS1	<_RF1>	REAL	Rayon d'arrondi ou largeur de chanfrein 1, voir <_AMODE> (DIZAINES DE MILLIERS)		
13	R2/FS2	<_RF2>	REAL	Rayon d'arrondi ou largeur de chanfrein 2, voir <_AMODE> (CENTAINES DE MILLIERS)		
14	R3/FS3	<_RF3>	REAL	Rayon d'arrondi ou largeur de chanfrein 3, voir <_AMODE> (MILLIONS)		
15	SC	<_SDIS>	REAL	Distance de sécurité		
16	F	<_FF1>	REAL	Avance pour ébauchage/finition		
17		<_NR>	INT	Identificateur du mode de chariotage (correspond à la touche logicielle verticale de sélection de la forme) :		
					0 =	Chariotage 1, 90 degrés, coin sans chanfreins/arrondis
					1 =	Chariotage 2, 90 degrés, coin avec chanfreins/arrondis
					2 =	Chariotage 3, coin quelconque avec chanfreins/arrondis

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
18		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Forme de saisie <_ZPD>/<_ZPL>	
					0 =	Xm/Zm
					1 =	Xm/α1
					2 =	Xm/α2
					3 =	α1/Zm
					4 =	α2/Zm
					5 =	α1/α2
21		<_AMODE>	INT	Mode alternatif		
				UNITES :	Point intermédiaire en X	
					0 =	En absolu, valeur de l'axe transversal en diamètre
					1 =	En relatif, valeur de l'axe transversal en rayon
				DIZAINES :	Point intermédiaire en Z	
					0 =	Absolu
					1 =	Relatif
				CENTAINES :	Point final en X	
					0 =	En absolu, valeur de l'axe transversal en diamètre
					1 =	En relatif, valeur de l'axe transversal en rayon
				MILLIERS :	Point final en Z	
					0 =	Absolu
					1 =	Relatif
				DIZAINES DE MILLIERS :	Rayon/Chanfrein 1	
					0 =	Rayon
					1 =	Chanfrein
				CENTAINES DE MILLIERS :	Rayon/Chanfrein 2	
					0 =	Rayon
					1 =	Chanfrein
				MILLIONS :	Rayon/Chanfrein 3	
					0 =	Rayon
					1 =	Chanfrein

20.1.43 CYCLE952 - Plongée de contour

Syntaxe

```
CYCLE952(<_PRG>, <_CON>, <_CONR>, <_VARI>, <_F>, <_FR>, <_RP>, <_D>,
<_DX>, <_DZ>, <_UX>, <_UZ>, <_U>, <_U1>, <_BL>, <_XD>, <_ZD>, <_XA>,
<_ZA>, <_XB>, <_ZB>, <_XDA>, <_XDB>, <_N>, <_DP>, <_DI>, <_SC>,
<_DN>, <_GMODE>, <_DMODE>, <_AMODE>, <_PK>, <_DCH>, <_FS>)
```

Paramètres

N°	Masque de paramètre	Paramètre interne	Type de données	Signification
1	PRG	<_PRG>	STRING[100]	Nom du programme de chariotage
2	CON	<_CON>	STRING[100]	Nom du programme à partir duquel est lu le contour de pièce brute actualisé (pour usinage de la matière restante)
3	CONR	<_CONR>	STRING[100]	Nom du programme dans lequel est écrit le contour de pièce brute actualisé (voir <_AMODE> DIZAINES DE MILLIERS)

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
4		<_VARI>	INT	Type d'usinage		
				UNITES :	Mode de chariotage	
					1 =	Longitudinal
					2 =	Transversal
					3 =	Parallèle au contour
				DIZAINES :	Usinage technologique (voir <_GMODE> CENTAINES)	
					1 =	Ebauchage
					2 =	Finition
					3 =	Réservé
					4 =	Ebauchage sur deux canaux
					5 =	Finition sur deux canaux
				CENTAINES :	Sens d'usinage	
					1 =	Sens d'usinage X -
					2 =	Sens d'usinage X +
					3 =	Sens d'usinage Z -
					4 =	Sens d'usinage Z +
				MILLIERS :	Direction de pénétration	
					1 =	Extérieur X -
					2 =	Intérieur X +
					3 =	Face frontale Z -
					4 =	Face arrière Z +
				DIZAINES DE MILLIERS :	Définition de la prise d'effet des surépaisseurs de finition	
					0 =	Surépaisseurs de finition UX et UZ séparées
					1 =	Surépaisseur de finition U parallèle au contour
				CENTAINES DE MILLIERS :	Lissage	
					0 =	Compatibilité, lissage automatique
					1 =	Avec lissage sur le contour
					2 =	Sans lissage
					3 =	Lissage automatique
				MILLIONS :	Détalonnages	
					0 =	La position n'est pas évaluée pour plongée, - matière restante et plongée G+D, - matière restante
					1 =	Usinage des détalonnages
					2 =	Ne pas usiner les détalonnages
				DIZAINES DE MILLIONS :	Derrière/devant le centre de rotation	
					0 =	Usinage devant le centre de rotation

N°	Masque de paramètre	Paramètre interne	Type de données	Signification		
					1 =	Réservé
5	F	<_F>	REAL	Avance pour ébauchage/finition		
	FZ			Avance abscisse de plongée G+D		
6	FR	<_FR>	REAL	Avance pour la pénétration dans les détalonnages d'ébauchage		
	FX			Avance de l'ordonnée de plongée G+D		
7	RP	<_RP>	REAL	Plan de retrait pour usinage intérieur (abs, toujours diamètre)		
8	D	<_D>	REAL	Passe d'ébauchage (voir <_AMODE> UNITES)		
9	DX	<_DX>	REAL	Passe X (voir <_AMODE> UNITES)		
10	DZ	<_DZ>	REAL	Passe Z (voir <_AMODE> UNITES)		
11	UX	<_UX>	REAL	Surépaisseur de finition X (voir <_VARI> DIZAINES DE MILLIERS)		
12	UZ	<_UZ>	REAL	Surépaisseur de finition Z (voir <_VARI> DIZAINES DE MILLIERS)		
13	U	<_U>	REAL	Surépaisseur de finition parallèle au contour (voir <_VARI> DIZAINES DE MILLIERS)		
14	U1	<_U1>	REAL	Surépaisseur supplémentaire lors de la finition (voir <_AMODE> MILLIERS)		
15	BL	<_BL>	INT	Définition de la pièce brute	1 =	Cylindre avec surépaisseur
					2 =	Surépaisseur sur le contour de la pièce finie
					3 =	Le contour de la pièce brute est indiqué
16	XD	<_XD>	REAL	Définition de la pièce brute X (voir <_AMODE> CENTAINES DE MILLIERS)		
17	ZD	<_ZD>	REAL	Définition de la pièce brute Z (voir <_AMODE> MILLIONS)		
18	XA	<_XA>	REAL	Limite 1 X (abs, toujours diamètre)		
19	ZA	<_ZA>	REAL	Limite 1 Z (abs)		
20	XB	<_XB>	REAL	Limite 2 X (voir <_AMODE> DIZAINES DE MILLIONS)		
21	ZB	<_ZB>	REAL	Limite 2 Z (voir <_AMODE> CENTAINES DE MILLIONS)		
22	XDA	<_XDA>	REAL	Limite de plongée 1 pour la 1re Position de gorge sur la face frontale (abs, toujours diamètre)		
23	XDB	<_XDB>	REAL	Limite de plongée 2 pour la 1re Position de gorge sur la face frontale (abs, toujours diamètre)		
24	N	<_N>	INT	Nombre de gorges		
25	DP	<_DP>	REAL	Distance entre gorges	Gorge longitudinale : parallèle à l'axe Z	
					Gorge transversale : parallèle à l'axe X	
26	DI	<_DI>	REAL	Distance pour interruption de l'avance	0 =	Sans interruption
					> 0 =	Avec interruption
27	SC	<_SC>	REAL	Distance de sécurité pour le contournement d'obstacles, en relatif		
28	D2	<_DN>	INT	Numéro D pour 2e tranchant, si non programmé ⇒ D+1		

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
29		<_GMODE>	INT	Mode géométrie (évaluation des valeurs géométriques programmées)		
				UNITES :	Réservé	
				DIZAINES :	Réservé	
				CENTAINES :	Sélection de l'opération / calcul du point de départ uniquement	
					0 =	Usinage normal (aucun mode de compatibilité n'est requis)
					1 =	Usinage normal
					2 =	Calculer la position de départ - aucun usinage (uniquement pour l'appel à partir de ShopMill/ShopTurn)
				MILLIERS :	Limitation	
					0 =	Non
					1 =	Oui
				DIZAINES DE MILLIERS :	Saisir la limite 1 X	
					0 =	Non
					1 =	Oui
				CENTAINES DE MILLIERS :	Saisir la limite 2 X	
					0 =	Non
					1 =	Oui
				MILLIONS :	Saisir la limite 1 Z	
					0 =	Non
					1 =	Oui
				DIZAINES DE MILLIONS :	Saisir la limite 2 Z	
					0 =	Non
					1 =	Oui

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
30		<_DMODE>	INT	Mode d'affichage		
				UNITES :	Plan d'usinage G17/G18/G19	
					0 =	Compatibilité, le plan actif avant l'appel de cycle reste actif
					1 =	G17 (actif uniquement dans le cycle)
					2 =	G18 (actif uniquement dans le cycle)
					3 =	G19 (actif uniquement dans le cycle)
				DIZAINES :	Mode technologie	
					1 =	Chariotage du contour
					2 =	Plongée de contour
					3 =	Plongée G+D
				CENTAINES :	Semi-finition	
					0 =	Non
					1 =	Oui
				MILLIERS :	---	Réservé
				DIZAINES DE MILLIERS :	Mise à l'échelle de la technologie dans les masques de cycle (Page 867)	
					0 =	Saisie : complète
					1 =	Saisie : simplifiée

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
31		<_AMODE>	INT	Mode alternatif		
				UNITES :	Sélection de passe	
					0 =	Passes DX et DZ pour mode de chariotage parallèle au contour
					1 =	Passe D
				DIZAINES :	Stratégie de passe	
					0 =	Profondeur de passe variable (90 à 100 %)
					1 =	Profondeur de passe constante
				CENTAINES :	Plan de passes	
					0 =	Uniforme
					1 =	Aligner sur les arêtes
				MILLIERS :	Sélection de surépaisseur sur contour U1, double finition	
					0 =	Non
					1 =	Oui
				DIZAINES DE MILLIERS :	Actualiser la sélection de la pièce brute	
					0 =	Non
					1 =	Oui
				CENTAINES DE MILLIERS :	Sélection de la surépaisseur sur la pièce brute XD	
					0 =	En absolu, valeur de l'axe transversal en diamètre
					1 =	En relatif, valeur de l'axe transversal en rayon
				MILLIONS :	Sélection de la surépaisseur sur la pièce brute ZD	
					0 =	Absolu
					1 =	Relatif
				DIZAINES DE MILLIONS :	Sélection de la limite 2 XB	
					0 =	En absolu, valeur de l'axe transversal en diamètre
					1 =	En relatif, valeur de l'axe transversal en rayon
				CENTAINES DE MILLIONS :	Sélection de la limite 2 ZB	
					0 =	Absolu
					1 =	Relatif
				MILLIARDS :		
					0 =	Canal pilote
					1 =	Canal asservi
32		<_PK>	INT	Numéro du canal partenaire lorsque plus de 2 canaux sont disponibles sur la machine.		
33	DCH	<_DCH>	REAL	Décalage de canal		
34	FS	<_FS>	REAL	Avance finition pour usinage complet		

20.1.44 CYCLE4071 - Rectification longitudinale avec pénétration au point d'inversion

Syntaxe

```
CYCLE4071(<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

Paramètres

Nº	Paramètre	Type de données	Signification
1	<S_A>	REAL	Profondeur de passe au départ
2	<S_B>	REAL	Profondeur de passe à la fin
3	<S_W>	REAL	Largeur de rectification
4	<S_U>	REAL	Durée des passes à lécher
5	<S_I>	REAL	Avance pour passe
6	<S_K>	REAL	Avance pour passe transversale
7	<S_H>	INT	Nombre de répétitions
8	<S_A1>	AXIS	Axe de pénétration (facultatif) ou 1er axe géométrique
9	<S_A2>	AXIS	Axe d'oscillation (facultatif) ou 2e axe géométrique

Fonction

Le cycle sert à exécuter des passes <S_H>. Les profondeurs de passe au départ et à la fin peuvent être différentes. Un mouvement tangentiel a lieu entre chaque passe.

Déroulement

1. Démarrage du cycle dans la position actuelle de l'axe d'oscillation.
2. Déplacement de l'axe de pénétration sur la profondeur de passe au départ P1 **<**S_A**>** avec l'avance pour passe P5 <S_I>.
3. Etincelage avec durée des passes à lécher P4 **<**S_U**>**.
4. Déplacement de l'axe d'oscillation avec la largeur de rectification P3 **<**S_W**>** en tant que course et avec l'avance pour passe transversale P6 <S_K>.
5. Déplacement de l'axe de pénétration sur la profondeur de passe à la fin P2 **<**S_B**>** avec l'avance pour passe P5 <S_I>.
6. Etincelage avec durée des passes à lécher P4 **<**S_U**>**.
7. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 **<**S_A**>** en tant que course sur le point de départ et avec l'avance pour passe transversale P6 <S_K>.

Le déroulement ne peut pas être interrompu en bloc par bloc.

Le déroulement est répété en fonction du nombre de répétitions programmé P7 (<S_H>).

Exemple

Exécuter deux mouvements pendulaires avec les paramètres de cycle suivants :

- Profondeur de passe au départ : 0,02 mm
- Profondeur de passe à la fin : 0,01 mm
- Course : 100 mm
- Durée des passes à lécher : 1 s
- Avance pour passe : 1 mm/min
- Avance transversale : 1000 mm/min
- Répétitions : 2
- Axe d'oscillation et de pénétration : Axes géométriques standard

Code de programme

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.45 CYCLE4072 - Rectification longitudinale avec pénétration au point d'inversion et signal d'abandon

Syntaxe

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

Paramètres

Nº	Paramètre	Type de données	Signification
1	<S_GAUGE>	STRING	Conditions d'abandon pour la passe : 1. N° d'une entrée rapide 2. Expression logique
2	<S_A>	REAL	Profondeur de passe au départ
3	<S_B>	REAL	Profondeur de passe à la fin
4	<S_W>	REAL	Largeur de rectification
5	<S_U>	REAL	Durée des passes à lécher
6	<S_I>	REAL	Avance pour passe
7	<S_K>	REAL	Avance pour passe transversale
8	<S_H>	INT	Nombre de répétitions
9	<S_A1>	AXIS	Axe de pénétration (facultatif) ou 1er axe géométrique
10	<S_A2>	AXIS	Axe d'oscillation (facultatif) ou 2e axe géométrique

Fonction

Le cycle sert à exécuter des passes <S_H> en tenant compte d'un signal d'abandon externe. Les profondeurs de passe au départ et à la fin peuvent être différentes. Un mouvement tangentiel a lieu entre chaque passe. La pénétration est annulée lorsque la condition d'abandon est remplie. Une course complète est toujours effectuée après l'abandon de la pénétration.

Déroulement

1. Démarrage du cycle dans la position actuelle de l'axe d'oscillation.
2. Déplacement de l'axe de pénétration sur la profondeur de passe au départ P2 **<**S_A**>** avec l'avance pour passe P6 <S_I>.
3. Étincelage avec durée des passes à lécher P5 **<**S_U**>**.
4. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 **<**S_W**>** en tant que course et avec l'avance pour passe transversale P7 <S_K>.
5. Déplacement de l'axe de pénétration sur la profondeur de passe à la fin P3 **<**S_B**>** avec l'avance pour passe P6 <S_I>.
6. Étincelage avec durée des passes à lécher P5 **<**S_U**>**.
7. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 **<**S_W**>** en tant que course sur le point de départ et avec l'avance pour passe transversale P7 <S_K>.
8. Sans abandon : le déroulement décrit ci-dessus est répété jusqu'à ce que le nombre de répétitions programmé P7 (<S_H>) soit atteint.
   Avec abandon : l'usinage se termine au prochain point de départ.

Le déroulement ne peut pas être interrompu en bloc par bloc.

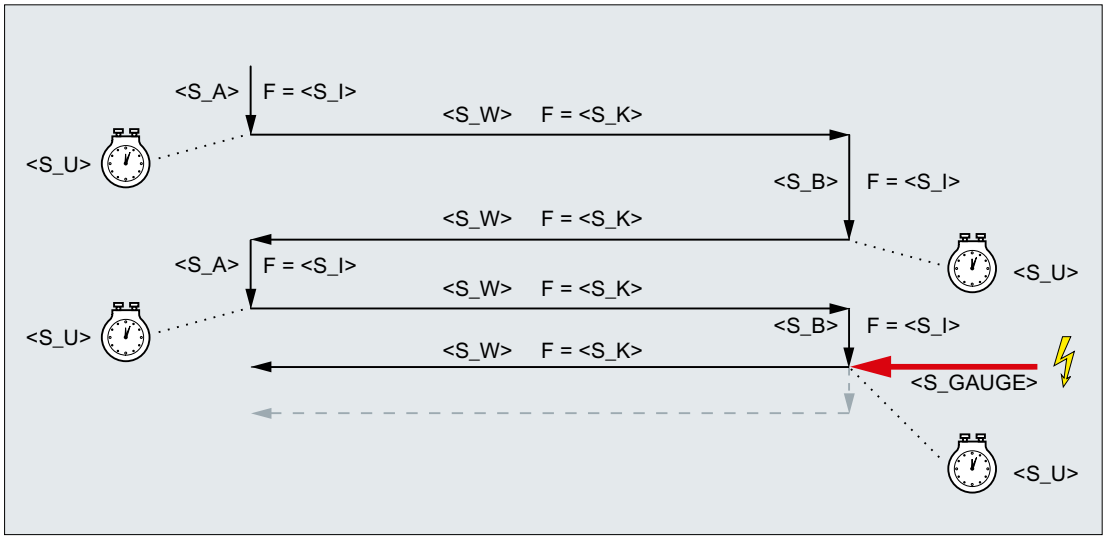


Figure 20-1 Abandon de la passe à la fin

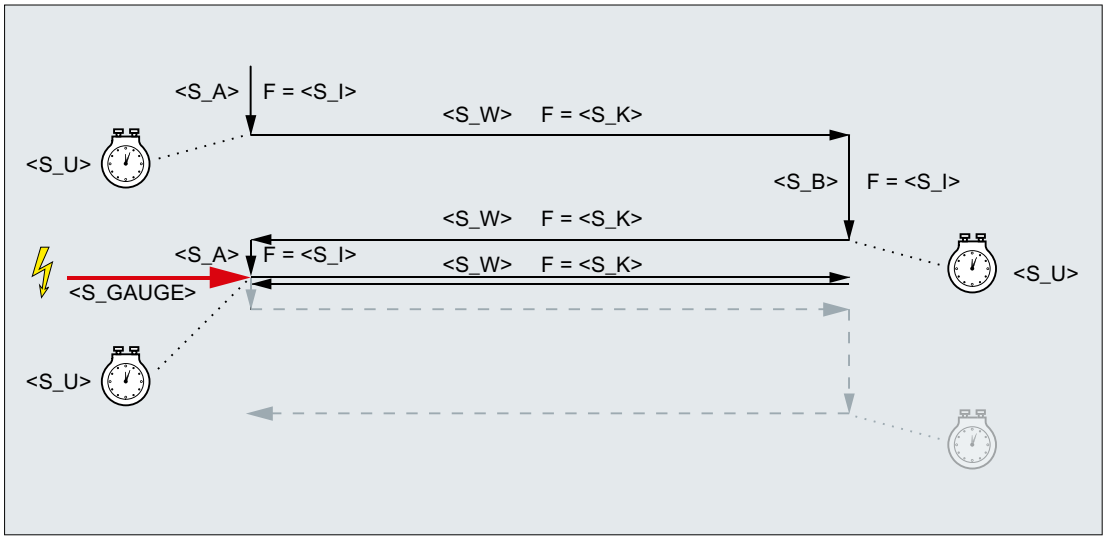


Figure 20-2 Abandon de la passe au départ

Ressources

Le cycle utilise comme ressources une action synchrone pour l'ensemble des blocs et une variable à action synchrone. L'action synchrone est déterminée de manière dynamique à partir de la plage libre de la bande d'actions synchrones (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). SYG_IS[1] est utilisé comme variable à action synchrone.

Exemples

Exemple 1 : Oscillation avec deux courses

Paramètres de cycle :

- Profondeur de passe au départ : 0,02 mm
- Profondeur de passe à la fin : 0,01 mm
- Course : 100 mm
- Durée des passes à lécher : 1 s
- Avance pour passe : 1 mm/min
- Avance transversale : 1000 mm/min
- Répétitions : 2
- Axe d'oscillation et de pénétration : axes géométriques standard

Signal d'abandon : entrée rapide 1 ($A_IN[1])

Code de programme

```
N10 T1 D1
N20 CYCLE4072("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

Exemple 2 : Oscillation avec deux courses

Paramètres de cycle :

- Profondeur de passe au départ : 0,02 mm
- Profondeur de passe à la fin : 0,01 mm
- Course : 100 mm
- Durée des passes à lécher : 1 s
- Avance pour passe : 1 mm/min
- Avance transversale : 1000 mm/min
- Répétitions : 2
- Axe d'oscillation et de pénétration : axes géométriques standard

Signal d'abandon : variable $A_DBR[20] < 0,01

Code de programme

```
N10 T1 D1
N20 CYCLE4072("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.46 CYCLE4073 - Rectification longitudinale avec pénétration continue

Syntaxe

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>, <S_A2>)
```

Paramètres

N°	Paramètre	Type de données	Signification
1	<S_A>	REAL	Profondeur de passe au départ
2	<S_B>	REAL	Profondeur de passe à la fin
3	<S_W>	REAL	Largeur de rectification
4	<S_U>	REAL	Durée des passes à lécher
5	<S_K>	REAL	Avance pour passe transversale
6	<S_H>	INT	Nombre de répétitions
7	<S_A1>	AXIS	Axe de pénétration (facultatif) ou 1er axe géométrique
8	<S_A2>	AXIS	Axe d'oscillation (facultatif) ou 2e axe géométrique

Fonction

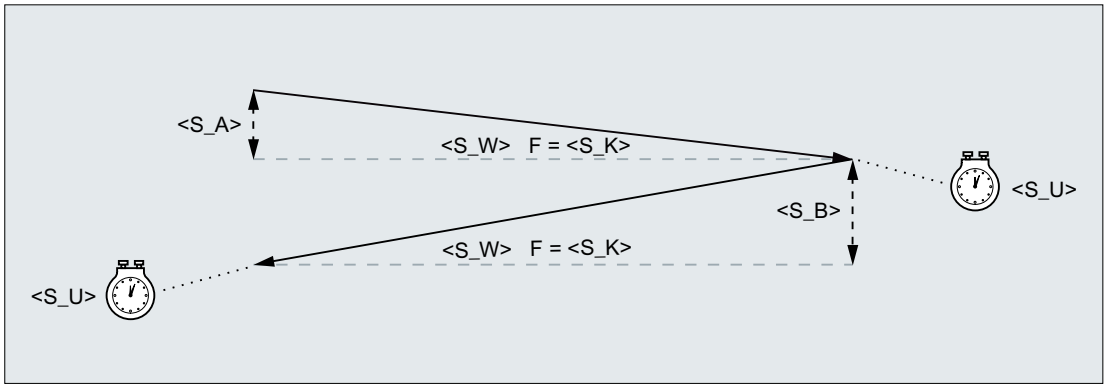
Le cycle sert à exécuter de passes <S_H>. La passe du départ vers la fin peut être différente de la passe de la fin vers le départ.

Déroulement

1. Démarrage du cycle dans la position actuelle de l'axe d'oscillation avec la profondeur de passe 0.
2. Déplacement de l'axe d'oscillation avec la largeur de rectification P3 <S_W> en tant que course et avec l'avance pour passe transversale P5 <S_K> avec augmentation continue de la profondeur de passe jusqu'à la profondeur de passe au départ P1 <S_A>.
3. Etincelage avec durée des passes à lécher P4 <S_U>.
4. Déplacement de l'axe d'oscillation avec la largeur de rectification P3 <S_W> en tant que course sur le point de départ et avec l'avance pour passe transversale P5 <S_K> avec augmentation continue de la profondeur de passe jusqu'à la profondeur de passe à la fin P2 <S_B>.
5. Etincelage avec durée des passes à lécher P4 <S_U>.

Le déroulement ne peut pas être interrompu en bloc par bloc.

Le déroulement est répété en fonction du nombre de répétitions programmé P7 (<S_H>).



Exemple

Oscillation avec deux courses

Paramètres de cycle :

- Profondeur de passe au départ : 0,02 mm
- Profondeur de passe à la fin : 0,01 mm
- Course : 100 mm
- Durée des passes à lécher : 1 s
- Avance transversale : 1000 mm/min
- Répétitions : 2
- Axe d'oscillation et de pénétration : Axes géométriques standard

Code de programme

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.47 CYCLE4074 - Rectification longitudinale avec pénétration continue et signal d'abandon

Syntaxe

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

Paramètres

Nº	Paramètre	Type de données	Signification
1	<S_GAUGE>	STRING	Conditions d'abandon pour la passe : 1. N° d'une entrée rapide 2. Expression logique
2	<S_A>	REAL	Profondeur de passe au départ
3	<S_B>	REAL	Profondeur de passe à la fin
4	<S_W>	REAL	Largeur de rectification
5	<S_U>	REAL	Durée des passes à lécher
6	<S_K>	REAL	Avance pour passe transversale
7	<S_H>	INT	Nombre de répétitions
8	<S_A1>	AXIS	Axe de pénétration (facultatif) ou 1er axe géométrique
9	<S_A2>	AXIS	Axe d'oscillation (facultatif) ou 2e axe géométrique

Fonction

Le cycle sert à exécuter des passe <S_H> en prenant en considération, par exemple, un signal d'abandon externe. Les profondeurs de passe au départ et à la fin peuvent être différentes. La pénétration est annulée lorsque la condition d'abandon est remplie. Une course complète est toujours effectuée après l'abandon de la pénétration.

Déroulement

1. Démarrage du cycle dans la position actuelle de l'axe d'oscillation avec la profondeur de passe 0.
2. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 **<**S_W**>** en tant que course et avec l'avance pour passe transversale P6 **<**S_K**>** avec augmentation continue de la profondeur de passe jusqu'à la profondeur de passe au départ P2 **<**S_A**>**.
3. Étincelage avec durée des passes à lécher P5 **<**S_U**>**.
4. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 **<**S_W**>** en tant que course sur le point de départ et avec l'avance pour passe transversale P6 **<**S_K**>** avec augmentation continue de la profondeur de passe jusqu'à la profondeur de passe à la fin P3 **<**S_B**>**.
5. Étincelage avec durée des passes à lécher P5 **<**S_U**>**.
6. Sans abandon : le déroulement décrit ci-dessus est répété jusqu'à ce que le nombre de répétitions programmé P7 (<S_H>) soit atteint.
   Avec abandon : la pénétration est annulée. L'usinage se termine au prochain point de départ.

Le déroulement ne peut pas être interrompu en bloc par bloc.

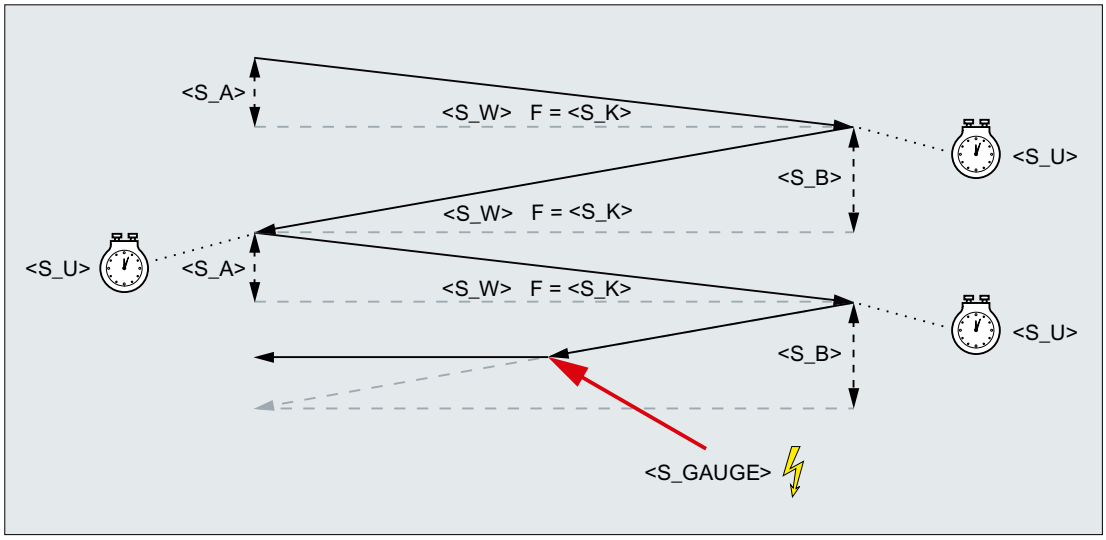


Figure 20-3 Abandon de la passe de la fin vers le départ

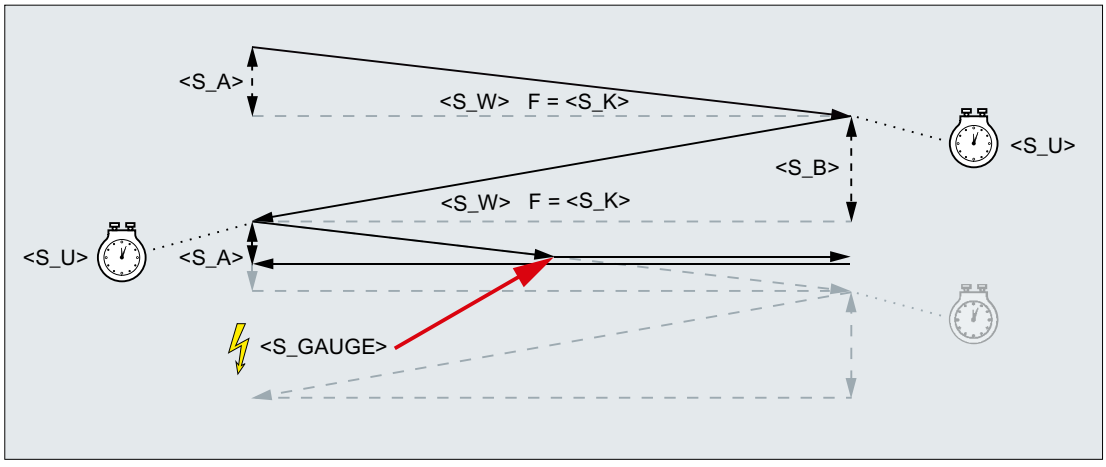


Figure 20-4 Abandon de la passe du départ vers la fin

Ressources

Le cycle utilise comme ressources une action synchrone pour l'ensemble des blocs et une variable à action synchrone. L'action synchrone est déterminée de manière dynamique à partir de la plage libre de la bande d'actions synchrones (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). SYG_IS[1] est utilisé comme variable à action synchrone.

Exemples

Exemple 1 : Oscillation avec deux courses

Paramètres de cycle :

- Profondeur de passe au départ : 0,02 mm
- Profondeur de passe à la fin : 0,01 mm

- Course : 100 mm
- Durée des passes à lécher : 1 s
- Avance transversale : 1000 mm/min
- Répétitions : 2
- Axe d'oscillation et de pénétration : axes géométriques standard

Signal d'abandon : entrée rapide 1 ($A_IN[1])

Code de programme

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

Exemple 2 : Oscillation avec deux courses

Paramètres de cycle :

- Profondeur de passe au départ : 0,02 mm
- Profondeur de passe à la fin : 0,01 mm
- Course : 100 mm
- Durée des passes à lécher : 1 s
- Avance transversale : 1000 mm/min
- Répétitions : 2
- Axe d'oscillation et de pénétration : axes géométriques standard

Signal d'abandon : Variable $A_DBR[20] < 0,01

Code de programme

```
N10 T1 D1
N20 CYCLE4074("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.48 CYCLE4075 - Rectification plane avec pénétration au point d'inversion

Syntaxe

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Paramètres

Nº	Paramètre	Type de données	Signification
1	<S_I>	REAL	Profondeur de passe au départ
2	<S_J>	REAL	Profondeur de passe à la fin

Nº	Paramètre	Type de données	Signification
3	<S_K>	REAL	Profondeur de passe totale
4	<S_A>	REAL	Largeur de rectification
5	<S_R>	REAL	Avance pour passe
6	<S_F>	REAL	Avance pour passe transversale
7	<S_P>	REAL	Durée des passes à lécher
8	<S_A1>	AXIS	Axe de pénétration (facultatif)
9	<S_A2>	AXIS	Axe d'oscillation (facultatif)

Fonction

Le cycle sert à usiner avec une profondeur de passe totale P3 <S_K> par étapes de pénétration. Les profondeurs de passe au départ P1 <S_I> et à la fin P2 <S_J> peuvent être différentes. Un mouvement tangentiel a lieu entre chaque passe.

Les indications de course P1 à P4 peuvent être négatives ou positives.

L'indication de l'axe de pénétration P8 <S_A1> et/ou de l'axe d'oscillation P9 <S_A2> est facultative. Si l'un ou aucun des paramètres n'est indiqué, le cycle utilise les deux premiers axes géométriques du canal.

Lorsque la somme de la profondeur de passe au départ P1 <S_I> et à la fin P2 <S_J> est égale à 0 ou que la profondeur de passe totale P3 <S_K> est égale à 0, une seule course d'étincelage est exécutée.

Déroulement

1. Démarrage du cycle dans la position actuelle de l'axe d'oscillation.
2. Déplacement de l'axe de pénétration sur la profondeur de passe au départ P1 <S_I> avec l'avance pour passe P5 <S_R>.
3. Étincelage avec durée des passes à lécher P7 <S_P>.
4. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 <S_A> en tant que course et avec l'avance pour passe transversale P6 <S_F>.
5. Déplacement de l'axe de pénétration sur la profondeur de passe à la fin P2 <S_J> avec l'avance pour passe P5 <S_R>.
6. Étincelage avec durée des passes à lécher P7 <S_P>.
7. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 <S_A> en tant que course sur le point de départ et avec l'avance pour passe transversale P6 <S_F>.

Le déroulement ne peut pas être interrompu en bloc par bloc.

Le déroulement est répété jusqu'à ce que la profondeur de passe totale P3 <S_K> soit atteinte. La dernière course est donc décomposée de manière irrégulière.

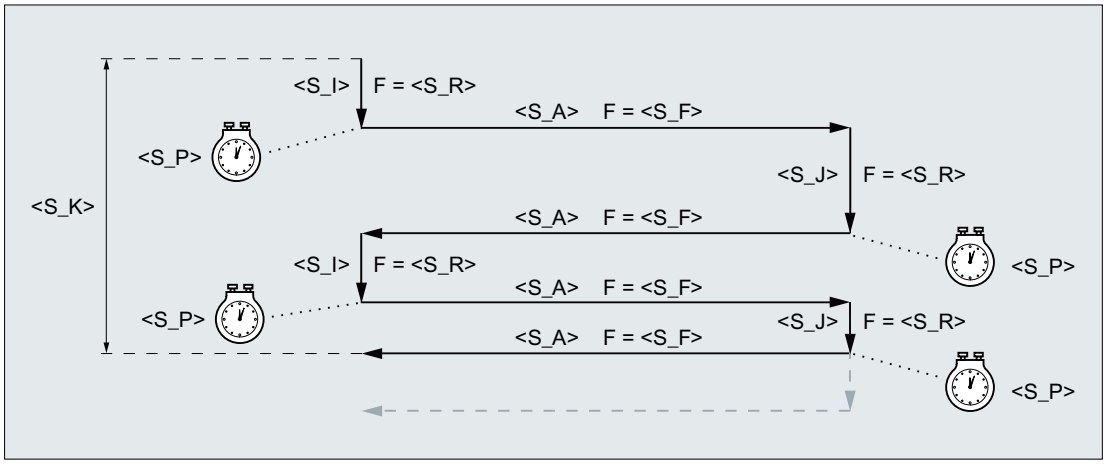


Figure 20-5 Profondeur de passe totale atteinte lors de la pénétration au deuxième point d'inversion

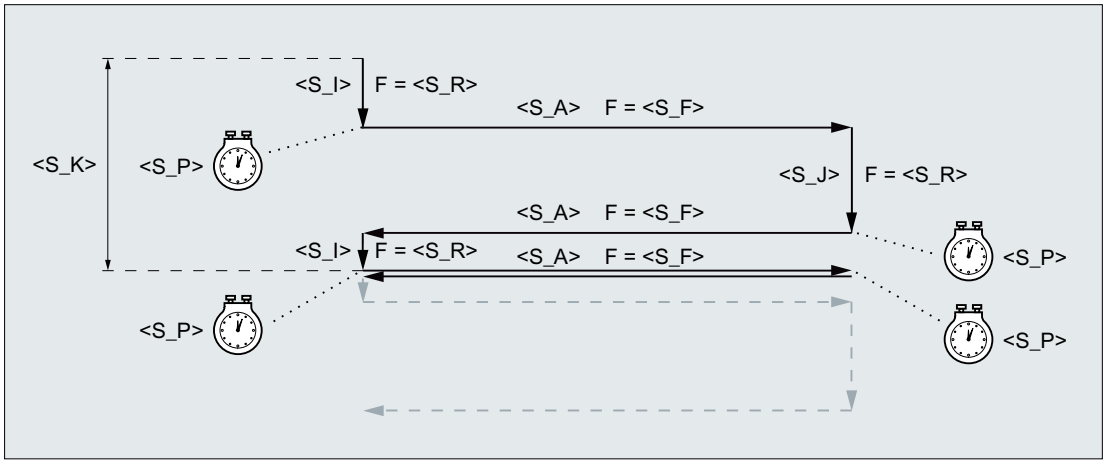


Figure 20-6 Profondeur de passe totale atteinte lors de la pénétration au premier point d'inversion

Exemple

Oscillation avec :

- Profondeur de passe de 0,02 mm au départ
- Profondeur de passe de 0,01 mm à la fin
- Profondeur de passe totale d'1 mm
- Course de 100 mm
- Avance de passe d'1 mm/min
- Avance transversale de 1000 mm/min
- Durée des passes à lécher d'1 seconde
- Axes géométriques standard

Code de programme
N10 T1 D1

Code de programme

```
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.49 CYCLE4077 - Rectification plane avec pénétration au point d'inversion et signal d'abandon

Syntaxe

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

Paramètres

Nº	Paramètre	Type de données	Signification
1	<S_GAUGE>	STRING	Condition d'abandon pour la passe : • N° d'une entrée rapide • Expression logique
2	<S_I>	REAL	Profondeur de passe au départ
3	<S_J>	REAL	Profondeur de passe à la fin
4	<S_K>	REAL	Profondeur de passe totale
5	<S_A>	REAL	Largeur de rectification
6	<S_R>	REAL	Avance pour passe
7	<S_F>	REAL	Avance pour passe transversale
8	<S_P>	REAL	Durée des passes à lécher
9	<S_A1>	AXIS	Axe de pénétration (facultatif)
10	<S_A2>	AXIS	Axe d'oscillation (facultatif)

Fonction

Le cycle sert à usiner avec une profondeur de passe totale P4 <S_K> par étapes de pénétration. Les profondeurs de passe au départ P2 <S_I> et à la fin P3 <S_J> peuvent être différentes. Un mouvement tangentiel a lieu entre chaque passe. La pénétration est annulée lorsque le signal d'abandon de l'entrée rapide est égal à 1 ou que la condition d'abandon est remplie. Une course complète est effectuée après l'abandon.

Les indications de course P2 à P5 peuvent être négatives ou positives.

L'indication de l'axe de pénétration P9 <S_A1> et/ou de l'axe d'oscillation P10 <S_A2> est facultative. Si l'un ou aucun des paramètres n'est indiqué, le cycle utilise les deux premiers axes géométriques du canal.

Lorsque la somme de la profondeur de passe au départ P2 <S_I> et à la fin P3 <S_J> est égale à 0 ou que la profondeur de passe totale P4 <S_K> est égale à 0, une seule course d'étincelage est exécutée.

Déroulement

1. Démarrage du cycle dans la position actuelle de l'axe d'oscillation.
2. Déplacement de l'axe de pénétration sur la profondeur de passe au départ P2 **<**S_I**>** avec l'avance pour passe P6 <S_R>.
3. Etincelage avec durée des passes à lécher P8 **<**S_P**>**.
4. Déplacement de l'axe d'oscillation avec la largeur de rectification P5 **<**S_A**>** en tant que course et avec l'avance pour passe transversale P7 <S_F>.
5. Déplacement de l'axe de pénétration sur la profondeur de passe à la fin P3 **<**S_J**>** avec l'avance pour passe P6 <S_R>.
6. Etincelage avec durée des passes à lécher P8 **<**S_P**>**.
7. Déplacement de l'axe d'oscillation avec la largeur de rectification P5 **<**S_A**>** en tant que course sur le point de départ et avec l'avance pour passe transversale P7 <S_F>.
8. Sans abondon : Le déroulement décrit ci-dessus est répété jusqu'à ce que la profondeur de passe totale P4 **<**S_K**>** soit atteinte. La dernière course est donc décomposée de manière irrégulière.
   Avec abondon : L'usinage se termine au point de départ.

Le déroulement ne peut pas être interrompu en bloc par bloc.

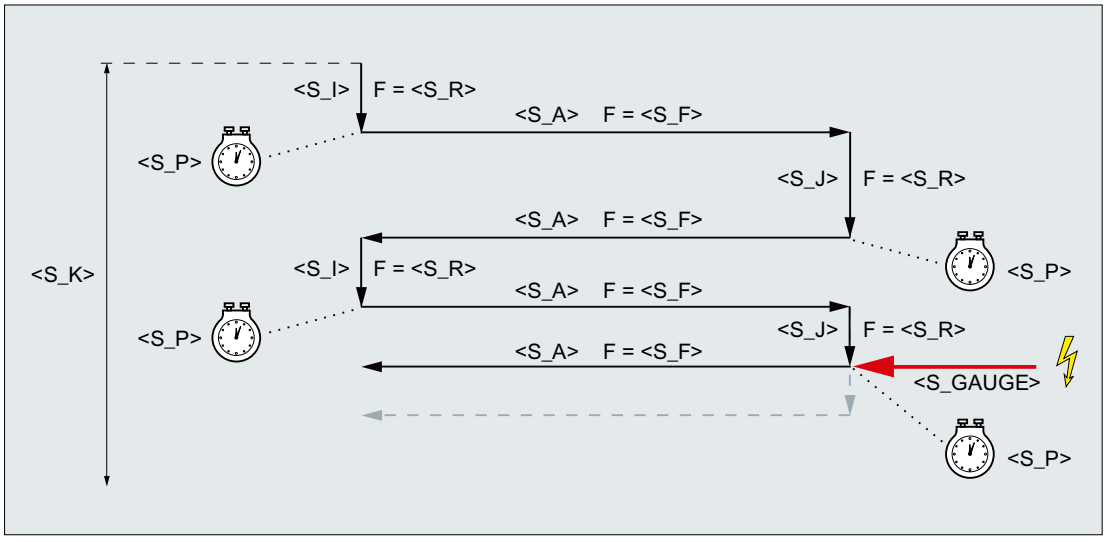


Figure 20-7 Abandon de la passe à la fin

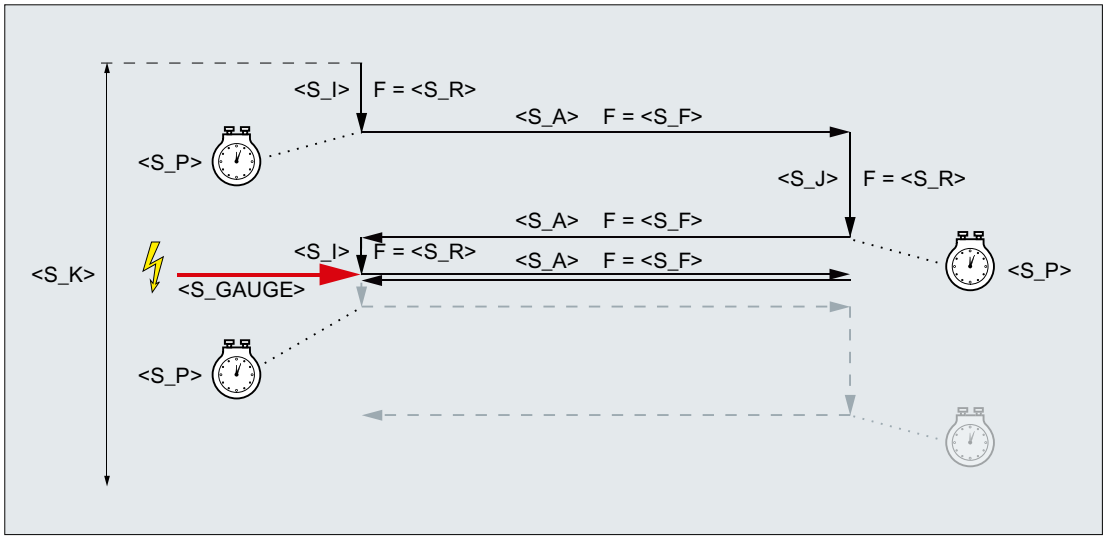


Figure 20-8 Abandon de la passe au départ

Ressources

Le cycle utilise comme ressources une action synchrone pour l'ensemble des blocs et une variable à action synchrone. L'action synchrone est déterminée de manière dynamique à partir de la plage libre de la bande d'actions synchrones (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). SYG_IS[1] est utilisé comme variable à action synchrone.

Exemples

Exemple 1

Oscillation avec :

- Profondeur de passe de 0,02 mm au départ
- Profondeur de passe de 0,01 mm à la fin
- Profondeur de passe totale d'1 mm
- Course de 100 mm
- Avance de passe d'1 mm/min
- Avance transversale de 1000 mm/min
- Durée des passes à lécher d'1 seconde
- Axes géométriques standard

Signal d'abandon : entrée rapide 1 ($A_IN[1])

Code de programme

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

Exemple 2

Oscillation avec :

- Profondeur de passe de 0,02 mm au départ
- Profondeur de passe de 0,01 mm à la fin
- Profondeur de passe totale d'1 mm
- Course de 100 mm
- Avance de passe d'1 mm/min
- Avance transversale de 1000 mm/min
- Durée des passes à lécher d'1 seconde
- Axes géométriques standard

Signal d'abandon : variable RAM 20 à double port inférieure à 0,01 ($A_DBR[20] < 0,01)

Code de programme

```
N10 T1 D1
N20 CYCLE4077("($A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.50 CYCLE4078 - Rectification plane avec pénétration continue

Syntaxe

```
CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)
```

Paramètres

Nº	Paramètre	Type de données	Signification
1	<S_I>	REAL	Profondeur de passe du départ vers la fin
2	<S_J>	REAL	Profondeur de passe de la fin vers le départ
3	<S_K>	REAL	Profondeur de passe totale
4	<S_A>	REAL	Largeur de rectification
5	<S_F>	REAL	Avance
6	<S_P>	REAL	Durée des passes à lécher
7	<S_A1>	AXIS	Axe de pénétration (facultatif)
8	<S_A2>	AXIS	Axe d'oscillation (facultatif)

Fonction

Le cycle sert à usiner avec une profondeur de passe totale P3 <S_K> par passes continues. Les profondeurs de passe du départ vers la fin P1 <S_I> et de la fin vers le départ P2 <S_J> peuvent être différentes.

Les indications de course P1 à P4 peuvent être négatives ou positives.

L'indication de l'axe de pénétration P8 <S_A1> et/ou de l'axe d'oscillation P9 <S_A2> est facultative. Si l'un ou aucun des paramètres n'est indiqué, le cycle utilise les deux premiers axes géométriques du canal.

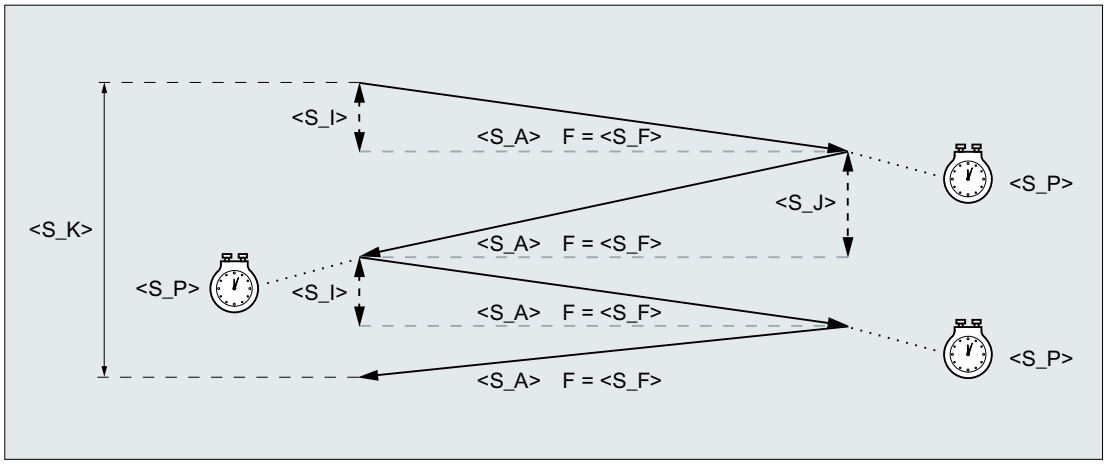
Lorsque la somme des profondeurs de passe P1 <S_I> et P2 <S_J> est égale à 0 ou que la profondeur de passe totale P3 <S_K> est égale à 0, une seule course d'étincelage est exécutée.

Déroulement

1. Démarrage du cycle dans la position actuelle de l'axe d'oscillation avec la profondeur de passe 0.
2. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 <S_A> en tant que course et l'avance P5 <S_F> avec augmentation continue de la profondeur de passe jusqu'à la profondeur de passe au départ P1 <S_I>.
3. Étincelage avec durée des passes à lécher P7 <S_P>.
4. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 <S_A> en tant que course sur le point de départ et avec l'avance P5 <S_F> avec augmentation continue de la profondeur de passe jusqu'à la profondeur de passe à la fin P2 <S_J>.
5. Étincelage avec durée des passes à lécher P7 <S_P>.
6. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 <S_A> en tant que course sur le point de départ et avec l'avance P5 <S_F>.

Le déroulement ne peut pas être interrompu en bloc par bloc.

Le déroulement est répété jusqu'à ce que la profondeur de passe totale P3 <S_K> soit atteinte. La dernière course est donc décomposée de manière irrégulière.



Exemple

Oscillation avec :

- Profondeur de passe de 20 mm au départ
- Profondeur de passe de 10 mm à la fin

- Profondeur de passe totale de 100 mm
- Course de 100 mm
- Avance 1000 mm/min
- Durée des passes à lécher d'1 seconde
- Axes géométriques standard

Code de programme

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

20.1.51 CYCLE4079 - Rectification plane avec pénétration intermittente

Syntaxe

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Paramètres

N°	Paramètre	Type de données	Signification
1	<S_I>	REAL	Profondeur de passe au départ
2	<S_J>	REAL	Profondeur de passe à la fin
3	<S_K>	REAL	Profondeur de passe totale
4	<S_A>	REAL	Largeur de rectification
5	<S_R>	REAL	Avance pour passe
6	<S_F>	REAL	Avance pour passe transversale
7	<S_P>	REAL	Durée des passes à lécher
8	<S_A1>	AXIS	Axe de pénétration (facultatif)
9	<S_A2>	AXIS	Axe d'oscillation (facultatif)

Fonction

Le cycle sert à usiner avec une profondeur de passe totale P3 <S_K> par étapes de pénétration. Les profondeurs de passe au départ P1 <S_I> et à la fin P2 <S_J> peuvent être différentes. Un mouvement tangentiel a lieu entre chaque passe.

Les indications de course P1 à P4 peuvent être négatives ou positives.

L'indication de l'axe de pénétration P8 <S_A1> et/ou de l'axe d'oscillation P9 <S_A2> est facultative. Si l'un ou aucun des paramètres n'est indiqué, le cycle utilise les deux premiers axes géométriques du canal.

Lorsque la somme de la profondeur de passe au départ P1 <S_I> et à la fin P2 <S_J> est égale à 0 ou que la profondeur de passe totale P3 <S_K> est égale à 0, une seule course d'étincelage est exécutée.

Déroulement

1. Démarrage du cycle dans la position actuelle de l'axe d'oscillation.
2. Déplacement de l'axe de pénétration sur la profondeur de passe au départ P1 <S_I> avec l'avance pour passe P5 <S_R>.
3. Étincelage avec durée des passes à lécher P7 <S_P>.
4. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 <S_A> en tant que course et avec l'avance pour passe transversale P6 <S_F>.
5. Déplacement de l'axe de pénétration sur la profondeur de passe à la fin P2 <S_J> avec l'avance pour passe P5 <S_R>.
6. Étincelage avec durée des passes à lécher P7 <S_P>.
7. Déplacement de l'axe d'oscillation avec la largeur de rectification P4 <S_A> en tant que course sur le point de départ et avec l'avance pour passe transversale P6 <S_F>.

Le déroulement ne peut pas être interrompu en bloc par bloc.

Le déroulement est répété jusqu'à ce que la profondeur de passe totale P3 <S_K> soit atteinte. La dernière course est donc décomposée de manière irrégulière.

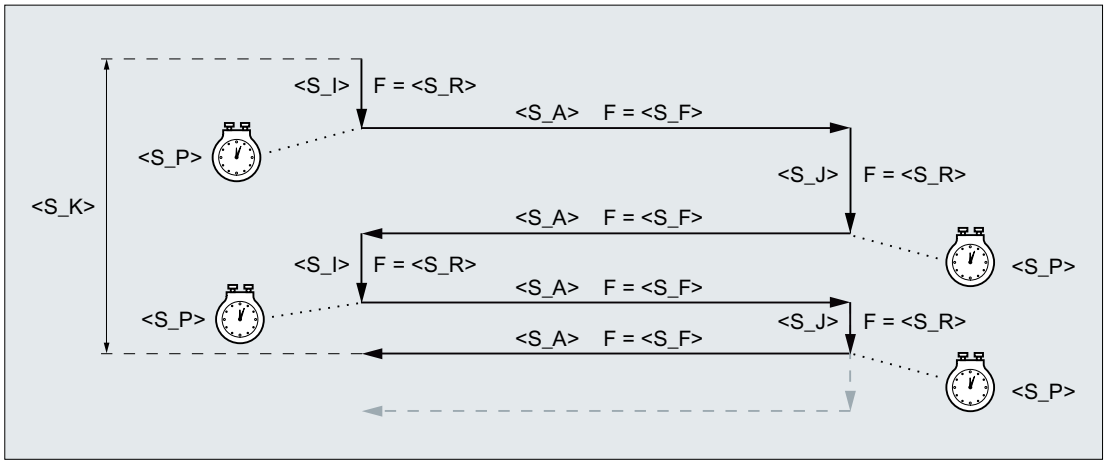


Figure 20-9 Profondeur de passe totale atteinte lors de la pénétration au deuxième point d'inversion

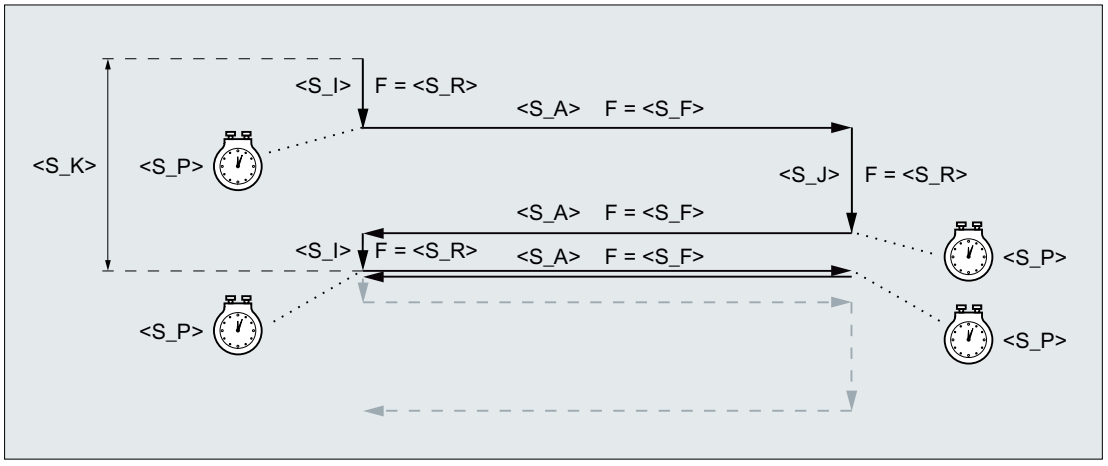


Figure 20-10 Profondeur de passe totale atteinte lors de la pénétration au premier point d'inversion

Exemple

Oscillation avec :

- Profondeur de passe de 0,02 mm au départ
- Profondeur de passe de 0,01 mm à la fin
- Profondeur de passe totale d'1 mm
- Course de 100 mm
- Avance de passe d'1 mm/min
- Avance transversale de 1000 mm/min
- Durée des passes à lécher d'1 seconde
- Axes géométriques standard

Code de programme

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.52 GROUP_BEGIN - Début du bloc de programme

Syntaxe

```
GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification		
1		<_LEVEL>	INT	Plan		
				0 =	Plan principal	
				1 =	1er niveau inférieur	
2		<_NAME>	STRING[128]	Nom de bloc		
3		<_SP>	INT	Broche		
				0 =	Pas de broche	
				1 =	Broche principale	
				2 =	Contre-broche	
4		<_MODE>	INT	Mode		
				Bit 0	= 1	GROUP_ADDEND existe
				Bit 1	= 1	ShopTurn : retrait automatique (accostage de la position de changement d'outil)
				Bit 12	réservé	
				Bit 13	réservé	
5		<S_ICON>	STRING[32]	Nom de l'icône (uniquement pour l'interface utilisateur)		

20.1.53 GROUP_END - Fin du bloc de programme

Syntaxe

```
GROUP_END(<_LEVEL>, <_SP>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification	
1		<_LEVEL>	INT	Plan	
				0 =	Plan principal
				1 =	1er niveau inférieur
2		<_SP>	INT	Broche	
				0 =	Pas de broche
				1 =	Broche principale
				2 =	Contre-broche

20.1.54 GROUP_ADDEND - Fin du supplément de mise au point

Syntaxe

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

Paramètres

Nº	Masque de paramètre	Paramètre interne	Type de données	Signification	
1		<_LEVEL>	INT	Plan	
				0 =	Plan principal
				1 =	1er niveau inférieur
2		<_SP>	INT	Broche	
				0 =	Pas de broche
				1 =	Broche principale
				2 =	Contre-broche

20.1.55 Conditions marginales

20.1.55.1 Mise à l'échelle de la technologie dans les masques de cycle

Si la mise à l'échelle de la technologie est activée, il est possible de sélectionner pour différents masques de cycle la saisie simplifiée, pour laquelle seuls les paramètres de cycle les plus importants sont affichés.

La saisie simplifiée peut être sélectionnée par exemple pour les masques de cycle suivants :

Technologie	Masque de cycle
Perçage	Perçage profond
	Taraudage
Fraisage	Poche rectangulaire
	Fraisage de contour : poche
Tournage	Filetage au tour : longitudinale
	Tournage de contour : chariotage
	Tournage de contour : plongée
	Tournage de contour : plongée G+D

Pour les masques de cycle concernés, l'interface utilisateur propose les options "Saisie : **simplifiée**" et "Saisie : **complète**".

Paramètres de cycle non affichés

Les paramètres de cycle non affichés pour la saisie simplifiée sont soit préréglés avec des valeurs fixes, technologiquement pertinentes, mais non modifiables, soit ils se voient affecter des valeurs paramétrables par les données de réglage de cycle spécifiques au canal. Voir section "Mise en service" > "Données de réglage de cycle spécifiques au canal" ci-dessous

Basculement "Saisie : complète" > "Saisie : simplifiée"

Si un masque de cycle est renseigné avec le réglage "Saisie complète" et ensuite basculé sur "Saisie simplifiée", lors de la génération de l'appel de cycle, les valeurs par défaut ou les valeurs des données de réglage seront utilisées pour les paramètres qui ne sont plus affichés.

Mise en service

Paramètres machine de configuration spécifiques à un canal

Le paramètre machine permet d'activer l'option de mise à l'échelle de la technologie dans les masques de cycle :

PM52210 $MCS_FUNCTION_MASK_DISP, bit 9 = 1 (afficher sélection "Saisie simplifiée")

Données de réglage de cycle spécifiques à un canal

Si la saisie simplifiée est activée dans les masques de cycle, les valeurs pour des paramètres de cycle déterminés peuvent être spécifiées par les données de réglage suivantes :

Numéro	Descripteur	Signification
SD55300	$SCS_EASY_SAFETY_CLEARANCE	Distance de sécurité
SD55301	$SCS_EASY_DWELL_TIME	Arrêt temporisé
SD55305	$SCS_EASY_DRILL_DEEP_FD1	Perçage profond : Pourcentage : 1. Avance
SD55306	$SCS_EASY_DRILL_DEEP_DF	Perçage profond : pourcentage : pénétration
SD55307	$SCS_EASY_DRILL_DEEP_V1	Perçage profond : profondeur de passe minimale
SD55308	$SCS_EASY_DRILL_DEEP_V2	Perçage profond : valeur de retrait
SD55309	$SCS_EASY_THREAD_RETURN_DIST	Filetage au tour : distance de retrait

20.2 Cycles de mesure

Les cycles de mesure sont des sous-programmes spéciaux fournis par Siemens et dédiés à des tâches de mesure. Comme cela est généralement le cas pour les cycles, les cycles de mesure peuvent aussi être adaptés au cas de figure concret par l'intermédiaire de paramètres.

Des cycles de mesure sont disponibles pour les mesures dans les domaines et technologies suivants :

- Mesures d'outil tournage/fraisage
- Mesures de pièce tournage/fraisage

Bibliographie

Vous trouverez la description détaillée des cycles de mesure dans :

Manuel de programmation Cycles de mesure

Tableaux

21

21.1 Instructions

Remarque

Cycles

La liste des instructions contient tous les cycles pouvant exister dans un programme CN (code G), c.-à-d. pouvant être programmés dans l'éditeur de programmes à l'aide de masques ou qui doivent être programmés pour la rectification sans aide à la programmation. Il n'est pas tenu compte des cycles pouvant être présents dans la commande pour des raisons de compatibilité, mais qui ne peuvent plus être modifiés à l'aide de l'éditeur de programmes de SINUMERIK Operate ("cycles de compatibilité").

Instructions A ... C

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
:	O	Numéro du bloc principal de la CN, fin des repères de saut, opérateur de concaténation		+		PGAsl
*	O	Opérateur de multiplication		+		PGAsl
+	O	Opérateur d'addition		+		PGAsl
-	O	Opérateur de soustraction		+		PGAsl
<	O	Opérateur relationnel, inférieur à		+		PGAsl
<<	O	Opérateur de concaténation pour chaînes		+		PGAsl
<=	O	Opérateur relationnel, inférieur ou égal à		+		PGAsl
=	O	Opérateur d'affectation		+		PGAsl
>=	O	Opérateur relationnel, supérieur ou égal à		+		PGAsl
/	O	Opérateur de division		+		PGAsl
/0 … … /7		Le bloc sera sauté (1er niveau)° ... ... Le bloc sera sauté (8e niveau)		+		PGsl
A	A	Nom d'axe	m/b	+		PGAsl
A2	A	Orientation de l'outil : angle RPY ou angle d'Euler	b	+		PGAsl
A3	A	Orientation de l'outil : 1re composante du vecteur de direction	b	+		PGAsl
A4	A	Orientation de l'outil : 1re composante du vecteur normal à la surface en début de bloc	b	+		PGAsl
A5	A	Orientation de l'outil : 1re composante du vecteur normal à la surface en fin de bloc	b	+		PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
A6	A	Orientation de l'outil : 1re composante du vecteur de direction pour l'axe rotatif du cône	b	+		PGAsl
A7	A	Orientation de l'outil : 1re composante du vecteur pour l'orientation intermédiaire sur l'enveloppe du cône	b	+		PGAsl
ABS	F	Valeur absolue (montant)		+	+	PGAsl
AC	K	Cotation absolue de coordonnées/positions	b	+		PGsl
ACC	K	Influence de l'accélération axiale actuelle	m	+	+	PGsl
ACCLIMA	K	Influence de l'accélération axiale maximale actuelle	m	+	+	PGAsl
ACN	K	Cotation absolue pour axes rotatifs, accostage de position dans le sens négatif	b	+		PGsl
ACOS	F	Arc cosinus (fonction trigonométrique)		+	+	PGAsl
ACP	K	Cotation absolue pour axes rotatifs, accostage de position dans le sens positif	b	+		PGsl
ACTBLOCNO	P	Affichage du numéro courant d'un bloc d'alarme, même si "Inhibition de l'affichage du bloc courant" (DISPLOF) est actif !		+		PGAsl
ADDFRAME	F	Prise en compte et éventuellement activation d'un frame mesuré		+	-	PGAsl, FB1sl (K2)
ADIS	A	Distance de transition entre blocs pour fonctions d'interpolation G1, G2, G3, ...	m	+		PGsl
ADISPOS	A	Distance de transition entre blocs pour le rapide G0	m	+		PGsl
ADISPOSA	P	Dimension de la fenêtre de tolérance pour IPOBRKA	m	+	+	PGAsl
ALF	A	Angle de relèvement rapide	m	+		PGAsl
AMIRROR	G	Fonction miroir programmable	b	+		PGsl
AND	K	ET logique		+		PGAsl
ANG	A	Angle d'élément de contour	b	+		PGsl
AP	A	Angle polaire	m/b	+		PGsl
APR	K	Droit d'accès en lecture / à l'affichage		+		PGAsl
APRB	K	Droit d'accès en lecture, OPI		+		PGAsl
APRP	K	Droit d'accès en lecture, programme pièce		+		PGAsl
APW	K	Droit d'accès en écriture		+		PGAsl
APWB	K	Droit d'accès en écriture, OPI		+		PGAsl
APWP	K	Droit d'accès en écriture, programme pièce		+		PGAsl
APX	K	Définition de la protection d'accès pour l'exécution de l'élément de langage indiqué		+		PGAsl
AR	A	Angle au centre	m/b	+		PGsl
AROT	G	Rotation programmable	b	+		PGsl
AROTS	G	Programmation de rotations de frames avec des angles solides	b	+		PGsl

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
AS	K	Définition d'une macro-instruction		+		PGAsl
ASCALE	G	Facteur d'échelle programmable	b	+		PGsl
ASIN	F	Fonction de calcul, arc sinus		+	+	PGAsl
ASPLINE	G	Akima Spline	m	+		PGAsl
ATAN2	F	Arc tangente 2		+	+	PGAsl
ATOL	A	Tolérance spécifique à l'axe pour fonctions de compresseur, lissage de l'orientation et modes d'arrondissement	m	+		PGAsl
ATRANS	G	Décalage d'origine additif programmable	b	+		PGsl
AUXFUDEL	P	Suppression pour un canal d'une fonction auxiliaire de la liste globale		+	-	FB1sl (H2)
AUXFUDELG	P	Suppression pour un canal de toutes les fonctions auxiliaires d'un groupe de fonctions auxiliaires de la liste globale		+	-	FB1sl (H2)
AUXFUMSEQ	P	Détermination de l'ordre de génération des fonctions auxiliaires M		+	-	FB1sl (H2)
AUXFUSYNC	P	Génération, sous forme de chaîne de caractères, d'un bloc de programme pièce complet pour le SERUPRO-Fin-ASUP spécifique à un canal à partir de la liste globale des fonctions auxiliaires		+	-	FB1sl (H2)
AX	K	Descripteur d'axe variable	m/b	+		PGAsl
AXCTSWE	P	Rotation d'un conteneur d'axes		+	-	PGAsl
AXCTSWEC	P	Suppression du déblocage de la rotation du conteneur d'axes		+	+	PGAsl
AXCTSWED	P	Rotation d'un conteneur d'axes (variante d'instruction pour la mise en service !)		+	-	PGAsl
AXIS	K	Descripteur d'axe, adresse d'axe		+		PGAsl
AXNAME	F	Conversion de la chaîne d'entrée en descripteur d'axe		+	-	PGAsl
AXSTRING	F	Conversion de la chaîne en numéro de broche		+	-	PGAsl
AXTOCHAN	P	Demande d'axe pour un canal donné Est possible depuis le programme CN ou à partir d'une action synchrone.		+	+	PGAsl
AXTOSPI	F	Conversion du descripteur d'axe en indice de broche		+	-	PGAsl
B	A	Nom d'axe	m/b	+		PGAsl
B2	A	Orientation de l'outil : angle RPY ou angle d'Euler	b	+		PGAsl
B3	A	Orientation de l'outil : composante du vecteur de direction	b	+		PGAsl
B4	A	Orientation de l'outil : 2e composante du vecteur normal à la surface en début de bloc	b	+		PGAsl
B5	A	Orientation de l'outil : 2e composante du vecteur normal à la surface en fin de bloc	b	+		PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
B6	A	Orientation de l'outil : 2e composante du vecteur de direction pour l'axe rotatif du cône	b	+		PGAsl
B7	A	Orientation de l'outil : 2e composante du vecteur pour l'orientation intermédiaire sur l'enveloppe du cône	b	+		PGAsl
B_AND	O	ET sur bits		+		PGAsl
B_OR	O	OU sur bits		+		PGAsl
B_NOT	O	Négation sur bits		+		PGAsl
B_XOR	O	OU exclusif sur bits		+		PGAsl
BAUTO	G	Définition de la première section spline par les 3 points suivants	m	+		PGAsl
BLOC	K	Définition, en association avec le mot-clé TO, de la section de programme à exécuter dans un sous-programme indirect		+		PGAsl
BLSYNC	K	L'exécution de la routine d'interruption ne doit commencer qu'au changement de bloc suivant		+		PGAsl
BNAT[6)]	G	Raccordement naturel avec le premier bloc spline	m	+		PGAsl
BOOL	K	Type de données : valeurs booléennes TRUE / FALSE ou 1/0		+		PGAsl
BOUND	F	Vérifie si la valeur se trouve la plage de valeurs définie. Égalité retourne la valeur de test.		+	+	PGAsl
BRISK[6)]	G	Accélération résultante sous forme d'échelon	m	+		PGAsl
BRISKA	P	Activer l'accélération résultante par échelon pour les axes programmés		+	-	PGAsl
BSPLINE	G	Spline B	m	+		PGAsl
BTAN	G	Raccordement tangentiel avec le premier bloc spline	m	+		PGAsl
C	A	Nom d'axe	m/b	+		PGAsl
C2	A	Orientation de l'outil : angle RPY ou angle d'Euler	b	+		PGAsl
C3	A	Orientation de l'outil : 3e composante du vecteur de direction	b	+		PGAsl
C4	A	Orientation de l'outil : 3e composante du vecteur normal à la surface en début de bloc	b	+		PGAsl
C5	A	Orientation de l'outil : 3e composante du vecteur normal à la surface en fin de bloc	b	+		PGAsl
C6	A	Orientation de l'outil : 3e composante du vecteur de direction pour l'axe rotatif du cône	b	+		PGAsl
C7	A	Orientation de l'outil : 3e composante du vecteur pour l'orientation intermédiaire sur l'enveloppe du cône	b	+		PGAsl
CAC	K	Accostage absolu d'une position		+		PGAsl
CACN	K	Valeur rangée dans table est accostée de façon absolue en sens négatif		+		PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
CACP	K	Valeur rangée dans table est accostée de façon absolue en sens positif		+		PGAsl
CALCDAT	F	Calcul du rayon et du centre d'un cercle passant par 3 ou 4 points		+	-	PGAsl
CALCPOSI	F	Contrôle du dépassement de la zone de protection, de la limitation de la zone de travail et des limites logicielles		+	-	PGAsl
CALL	K	Appel indirect de sous-programme		+		PGAsl
CALLPATH	P	Chemin de recherche programmable pour l'appel de sous-programmes		+	-	PGAsl
CANCEL	P	Annuler une action synchrone modale		+	-	FBSYsl
CASE	K	Branchement conditionnel		+		PGAsl
CDC	K	Accostage direct d'une position		+		PGAsl
CDOF[6)]	G	Désactiver la surveillance de collision	m	+		PGsl
CDOF2	G	Désactiver la surveillance de collision, fraisage périphérique 3D	m	+		PGsl
CDON	G	Activer la surveillance de collision	m	+		PGsl
CFC[6)]	G	Avance constante au niveau du contour	m	+		PGsl
CFIN	G	Avance constante uniquement pour courbure concave, pas pour courbure convexe	m	+		PGsl
CFINE	F	Affectation d'un décalage fin à une variable FRAME		+	-	PGAsl
CFTCP	G	Avance constante au point de référence de l'arête tranchante de l'outil, trajectoire centrale	m	+		PGsl
CHAN	K	Spécification du domaine de validité de données		+		PGAsl
CHANDATA	P	Réglage du numéro de canal pour des accès à des données de canal		+	-	PGAsl
CHAR	K	Type de données : caractère ASCII		+		PGAsl
CHF	A	Chanfrein ; valeur = longueur du chanfrein	b	+		PGsl
CHKDM	F	Contrôle d'univocité dans un magasin		+	-	FBWsl
CHKDNO	F	Contrôle d'univocité des numéros D		+	-	PGAsl
CHR	A	Chanfrein ; valeur = longueur du chanfrein dans la direction du déplacement		+		PGsl
CIC	K	Accostage relatif d'une position		+		PGAsl
CIP	G	Interpolation circulaire avec point intermédiaire	m	+		PGsl
CLEARM	P	Effacement d'un / de plusieurs repères de coordination entre canaux		+	+	PGAsl
CLRINT	P	Annuler une interruption		+	-	PGAsl
CMIRROR	F	fonction miroir par rapport à un axe de coordonnées		+	-	PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
COARSEA	K	Fin du déplacement lors de l'atteinte de "Arrêt précis grossier"	m	+		PGAsl
COLLPAIR	F	Vérification de l'appartenance à une paire de collision		+		PGAsl
COMPCAD	G	Activation de la fonction compacteur COMPCAD	m	+		PGAsl
COMPCURV	G	Activation de la fonction compacteur COMPCURV	m	+		PGAsl
COMPLETE		Instruction de commande pour la lecture et l'écriture de données		+		PGAsl
COMPOF[6)]	G	Activation du compactage de bloc CN	m	+		PGAsl
COMPON	G	Activation de la fonction compacteur COMPON	m	+		PGAsl
COMPSURF	G	Activation de la fonction compacteur COMSURF	m	+		PGAsl
CONTDCON	P	Activation du décodage du contour, sous forme tabulaire		+	-	PGAsl
CONTPRON	P	Activation de la préparation des références		+	-	PGAsl
CORROF	P	Désélection de tous les déplacements forcés actifs.		+	-	PGsl
CORRTRAFO	F	Modification des vecteurs d'offset ou des vecteurs de direction des axes d'orientation dans le modèle cinématique de la machine		+	-	PGAsl
COS	F	Cosinus (fonction trigonométrique)		+	+	PGAsl
COUPDEF	P	Définition groupe réducteurs électroniques / groupe broches synchrones		+	-	PGAsl
COUPDEL	P	Suppression groupe réducteurs électroniques		+	-	PGAsl
COUPOF	P	Désactivation du groupe réducteurs électroniques / de la paire de broches synchrones		+	-	PGAsl
COUPOFS	P	Désactivation du groupe réducteurs électroniques / de la paire de broches synchrones avec arrêt de la broche asservie		+	-	PGAsl
COUPON	P	Activation du groupe réducteurs électroniques / de la paire de broches synchrones		+	-	PGAsl
COUPONC	P	Activation du groupe réducteurs électroniques / de la paire de broches synchrones avec prise en charge de la programmation précédente		+	-	PGAsl
COUPRES	P	Réinitialisation groupe réducteurs électroniques		+	-	PGAsl
CP[6)]	G	Déplacement avec interpolation	m	+		PGAsl
CPBC	K	Couplage générique : critère de changement de bloc		+	+	FB3sl (M3)
CPDEF	K	Couplage générique : création d'un module de couplage		+	+	FB3sl (M3)

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
CPDEL	K	Couplage générique : suppression d'un module de couplage		+	+	FB3sl (M3)
CPFMOF	K	Couplage générique : comportement de l'axe asservi lors d'une désactivation complète		+	+	FB3sl (M3)
CPFMON	K	Couplage générique : comportement de l'axe asservi lors de l'activation		+	+	FB3sl (M3)
CPFMSON	K	Couplage générique : mode de synchronisation		+	+	FB3sl (M3)
CPFPOS	K	Couplage générique : position synchrone de l'axe asservi		+	+	FB3sl (M3)
CPFRS	K	Couplage générique : système de référence des coordonnées		+	+	FB3sl (M3)
CPLA	K	Couplage générique : définition d'un axe pilote		+	-	FB3sl (M3)
CPLCTID	K	Couplage générique : numéro de la table de courbes		+	+	FB3sl (M3)
CPLDEF	K	Couplage générique : définition d'un axe pilote et création d'un module de couplage		+	+	FB3sl (M3)
CPLDEL	K	Couplage générique : suppression d'un axe pilote d'un module de couplage		+	+	FB3sl (M3)
CPLDEN	K	Couplage générique : dénominateur du facteur de couplage		+	+	FB3sl (M3)
CPLINSC	K	Couplage générique : facteur d'échelle pour la valeur d'entrée d'un axe pilote		+	+	FB3sl (M3)
CPLINTR	K	Couplage générique : valeur de décalage pour la valeur d'entrée d'un axe pilote		+	+	FB3sl (M3)
CPLNUM	K	Couplage générique : numérateur du facteur de couplage		+	+	FB3sl (M3)
CPLOF	K	Couplage générique : désactivation d'un axe pilote d'un module de couplage		+	+	FB3sl (M3)
CPLON	K	Couplage générique : activation d'un axe pilote d'un module de couplage		+	+	FB3sl (M3)
CPLOUTSC	K	Couplage générique : facteur d'échelle pour la valeur de sortie d'un couplage		+	+	FB3sl (M3)
CPLOUTTR	K	Couplage générique : valeur de décalage pour la valeur de sortie d'un couplage		+	+	FB3sl (M3)
CPLPOS	K	Couplage générique : position synchrone de l'axe pilote		+	+	FB3sl (M3)
CPLSETVAL	K	Couplage générique : référence du couplage		+	+	FB3sl (M3)
CPMALARM	K	Couplage générique : inhibition des générations d'alarme spéciales relatives au couplage		+	+	FB3sl (M3)
CPMBRAKE	K	Couplage générique : comportement de l'axe asservi lors de certains signaux et ordres d'arrêt		+	-	FB3sl (M3)
CPMPRT	K	Couplage générique : Comportement du couplage au lancement du programme pièce lors d'une recherche en mode test du programme		+	+	FB3sl (M3)

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
CPMRESET	K	Couplage générique : comportement du couplage lors d'un RESET		+	+	FB3sl (M3)
CPMSTART	K	Couplage générique : comportement du couplage au lancement du programme pièce		+	+	FB3sl (M3)
CPMVDI	K	Couplage générique : comportement de l'axe asservi en réaction à certains signaux d'interface CN/AP		+	+	FB3sl (M3)
CPOF	K	Couplage générique : désactivation d'un module de couplage		+	+	FB3sl (M3)
CPON	K	Couplage générique : activation d'un module de couplage		+	+	FB3sl (M3)
CPRECOF[6)]	G	Désactiver la précision de contour programmable	m	+		PGAsl
CPRECON	G	Activer la précision de contour programmable	m	+		PGAsl
CPRES	K	Couplage générique : activation des données configurées du couplage de broches synchrones		+	-	FB3sl (M3)
CPROT	P	Activation/désactivation de zones de protection spécifiques au canal		+	-	PGAsl
CPROTDEF	P	Définition d'une zone de protection spécifique au canal		+	-	PGAsl
CPSETTYPE	K	Couplage générique : type de couplage		+	+	FB3sl (M3)
CPSYNCOP	K	Couplage générique : valeur de seuil pour la synchronisme "grossière" de position		+	+	FB3sl (M3)
CPSYNCOP2	K	Couplage générique : valeur de seuil pour le synchronisme de position "Grossier" 2		+	+	FB3sl (M3)
CPSYNCOV	K	Couplage générique : valeur de seuil pour la synchronisation "grossière" de la vitesse		+	+	FB3sl (M3)
CPSYNFIP	K	Couplage générique : valeur de seuil pour la synchronisation "fine" de position		+	+	FB3sl (M3)
CPSYNFIP2	K	Couplage générique : valeur de seuil pour le synchronisme de position "Fin" 2		+	+	FB3sl (M3)
CPSYNFIV	K	Couplage générique : valeur de seuil pour la synchronisation "fine" de la vitesse		+	+	FB3sl (M3)
CR	A	Rayon du cercle	b	+		PGsl
CROT	F	rotation du système de coordonnées actuel		+	-	PGAsl
CROTS	F	Rotations de frames programmables avec angles solides (rotation dans les axes indiqués)	b	+	-	PGsl
CRPL	F	Rotation de frame dans un plan quelconque		+	-	FB1sl (K2)
CSCALE	F	Facteur d'échelle pour plusieurs axes		+	-	PGAsl
CSPLINE	F	Spline cubique	m	+		PGAsl
CT	G	Cercle avec transition tangentielle	m	+		PGsl
CTAB	F	Déterminer position axe asservi correspondant à position axe pilote d'après table de courbes		+	+	PGAsl

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
CTABDEF	P	Activer la définition d'une table		+	-	PGAsl
CTABDEL	P	Suppression d'une table de courbes		+	-	PGAsl
CTABEND	P	Désactiver la définition d'une table		+	-	PGAsl
CTABEXISTS	F	Vérifie la table de courbes ayant le numéro n		+	+	PGAsl
CTABFNO	F	Nombre des tables de courbes encore possibles dans la mémoire		+	+	PGAsl
CTABFPOL	F	Nombre des polynômes encore possibles dans la mémoire		+	+	PGAsl
CTABFSEG	F	Nombre des segments de courbes encore possibles dans la mémoire		+	+	PGAsl
CTABID	F	Indication du numéro de table de la n-ième table de courbes		+	+	PGAsl
CTABINV	F	Déterminer position axe pilote correspondant à position axe asservi d'après table de courbes		+	+	PGAsl
CTABISLOCK	F	Retourne l'état de blocage de la table de courbes ayant le numéro n		+	+	PGAsl
CTABLOCK	P	Verrouillage de la suppression et de l'écrasement		+	+	PGAsl
CTABMEMTYP	F	Retourne la mémoire dans laquelle se trouve la table de courbes ayant le numéro n.		+	+	PGAsl
CTABMPOL	F	Nombre maximum des polynômes possibles dans la mémoire		+	+	PGAsl
CTABMSEG	F	Nombre maximum des segments de courbes possibles dans la mémoire		+	+	PGAsl
CTABNO	F	Nombre de tables de courbes définies dans SRAM ou DRAM		+	+	FB3sl (M3)
CTABNOMEM	F	Nombre de tables de courbes définies dans SRAM ou DRAM		+	+	PGAsl
CTABPERIOD	F	Restitue la périodicité de la table de courbes de numéro n		+	+	PGAsl
CTABPOL	F	Nombre des polynômes déjà utilisés dans la mémoire		+	+	PGAsl
CTABPOLID	F	Nombre des polynômes de courbes de la table de courbes ayant le numéro n		+	+	PGAsl
CTABSEG	F	Nombre des segments de courbes déjà utilisés dans la mémoire		+	+	PGAsl
CTABSEGID	F	Nombre des segments de courbes de la table de courbes ayant le numéro n		+	+	PGAsl
CTABSEV	F	Indique la valeur de fin de l'axe asservi du segment de la table de courbes		+	+	PGAsl
CTABSSV	F	Indique la valeur de départ de l'axe asservi du segment de la table de courbes		+	+	PGAsl
CTABTEP	F	Indique la valeur de l'axe pilote à la fin de la table de courbes		+	+	PGAsl
CTABTEV	F	Indique la valeur de l'axe asservi à la fin de la table de courbes		+	+	PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
CTABTMAX	F	Indique la valeur maximale de l'axe asservi de la table de courbes		+	+	PGAsl
CTABTMIN	F	Indique la valeur minimale de l'axe asservi de la table de courbes		+	+	PGAsl
CTABTSP	F	Indique la valeur de l'axe pilote au début de la table de courbes.		+	+	PGAsl
CTABTSV	F	Indique la valeur de l'axe asservi au début de la table de courbes		+	+	PGAsl
CTABUNLOCK	P	Annulation du verrouillage de la suppression et de l'écrasement		+	+	PGAsl
CTOL	A	Tolérance de contour pour fonctions de compresseur, lissage de l'orientation et modes d'arrondissement	m	+		PGAsl
CTRANS	F	Décalage d'origine pour plusieurs axes		+	-	PGAsl
CUT2D[6)]	G	2D-WRK	m	+		PGsl
CUT2DD	G	CRO 2½D rapportée à un outil différentiel	m	+		PGsl
CUT2DF	G	CRO 2D, relative au frame actuel (plan incliné)	m	+		PGsl
CUT2DFD	G	CRO 2½D rapportée à un outil différentiel, relative au frame actuel (plan incliné)	m	+		PGsl
CUT3DC	G	CRO 3D pour le fraisage périphérique	m	+		PGAsl
CUT3DCC	G	CRO 3D pour le fraisage périphérique en tenant compte d'une surface de délimitation avec correction de rayon 3D : contour de la surface d'usinage	m	+		PGAsl
CUT3DCCD	G	CRO 3D pour le fraisage périphérique en tenant compte d'une surface de délimitation avec outil différentiel sur la trajectoire du centre de l'outil : déplacement perpendiculaire jusqu'à la surface de délimitation	m	+		PGAsl
CUT3DCD	G	CRO 3D rapportée à un outil différentiel pour le fraisage périphérique	m	+		PGAsl
CUT3DF	G	CRO 3D pour le fraisage en bout avec changement d'orientation	m	+		PGAsl
CUT3DFD	G	CRO 3D rapportée à un outil différentiel pour le fraisage en bout avec changement d'orientation	m	+		PGAsl
CUT3DFF	G	CRO 3D pour le fraisage en bout avec orientation constante. L'orientation de l'outil est définie par G17 - G19 et, le cas échéant, par la composante rotationnelle d'un frame.	m	+		PGAsl
CUT3DFS	G	CRO 3D pour le fraisage en bout avec orientation constante. L'orientation de l'outil est définie par G17 - G19 et n'est pas influencée par les frames.	m	+		PGAsl
CUTCONOF[6)]	G	Désactiver la correction de rayon constante	m	+		PGsl
CUTCONON	G	Activer la correction de rayon constante	m	+		PGsl

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
CUTMOD	A	Activer la modification des données de correction pour outils indexables (en liaison avec des porte-outils orientables)	m	+		PGAsl
CUTMODK	A	Activer la modification des données de correction pour outils indexables (en liaison avec des transformations d'orientation ayant été définies au moyen de chaînes cinématiques)	m	+		PGAsl
CYCLE60	C (T)	Cycle de gravure		+		PGAsl
CYCLE61	C (T)	Surfaçage		+		PGAsl
CYCLE62	C (T)	Appel de contour		+		PGAsl
CYCLE63	C (T)	Fraisage de poche de contour		+		PGAsl
CYCLE64	C (T)	Perçage des avant-trous d'une poche de contour		+		PGAsl
CYCLE70	C (T)	Fraisage de filetages		+		PGAsl
CYCLE72	C (T)	Fraisage en contournage		+		PGAsl
CYCLE76	C (T)	Fraisage d'un tourillon rectangulaire		+		PGAsl
CYCLE77	C (T)	Fraisage d'un tourillon circulaire		+		PGAsl
CYCLE78	C (T)	Fraisage de filetages		+		PGAsl
CYCLE79	C (T)	Polyèdre		+		PGAsl
CYCLE81	C (T)	Perçage, centrage		+		PGAsl
CYCLE82	C (T)	Perçage, lamage		+		PGAsl
CYCLE83	C (T)	Perçage de trous profonds		+		PGAsl
CYCLE84	C (T)	Taraudage sans porte-taraud compensateur		+		PGAsl
CYCLE85	C (T)	Alésage à l'alésoir		+		PGAsl
CYCLE86	C (T)	Alésage au tour		+		PGAsl
CYCLE92	C (T)	Tronçonnage		+		PGAsl
CYCLE95	C (T)	Chariotage du contour		+		PGAsl
CYCLE98	C (T)	Concaténation de filetages		+		PGAsl
CYCLE99	C (T)	Filetage au tour		+		PGAsl
CYCLE150	C (M)	Affichage/journalisation des résultats de mesure		+		BNMsl
CYCLE435	C (T)	Calculer la position de dressage		+		PGAsl
CYCLE495	C (T)	Profilage		+		PGAsl
CYCLE750	C (A)	Cycle de travail interne pour CYCLE751 ... CYCLE759 (contient l'instruction MMC d'appel de la fonction)		-		FB3sl (T4)
CYCLE751	C (A)	Ouverture / exécution / fermeture d'une session d'optimisation		M		FB3sl (T4)
CYCLE752	C (A)	Ajout d'un axe à une session d'optimisation		M		FB3sl(T4)
CYCLE753	C (A)	Sélection du mode d'optimisation		M		FB3sl (T4)
CYCLE754	C (A)	Ajout / suppression d'un bloc de données		M		FB3sl (T4)
CYCLE755	C (A)	Sauvegarde / restauration d'un bloc de données		M		FB3sl (T4)

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
CYCLE756	C (A)	Activation des résultats d'optimisation		M		FB3sl (T4)
CYCLE757	C (A)	Sauvegarde des données d'optimisation		M		FB3sl (T4)
CYCLE758	C (A)	Modification de la valeur d'un paramètre		M		FB3sl (T4)
CYCLE759	C (A)	Lecture de la valeur d'un paramètre		M		FB3sl (T4)
CYCLE800	C (T)	Orientation		+		PGAsl
CYCLE801	C (T)	Réseau de trous ou cadre		+		PGAsl
CYCLE802	C (T)	Positions quelconques		+		PGAsl
CYCLE830	C (T)	Perçage de trous profonds 2		+		PGAsl
CYCLE832	C (T)	High Speed Settings		+		PGAsl
CYCLE840	C (T)	Taraudage avec porte-taraud compensateur		+		PGAsl
CYCLE899	C (T)	Fraisage de rainure borgne		+		PGAsl
CYCLE930	C (T)	Gorge		+		PGAsl
CYCLE940	C (T)	Dégagement de formes		+		PGAsl
CYCLE951	C (T)	Chariotage		+		PGAsl
CYCLE952	C (T)	Plongée de contour		+		PGAsl
CYCLE961	C (M)	Permet de déterminer la position d'un coin (intérieur ou extérieur) sur une pièce et de l'utiliser pour le décalage de l'origine		+		BNMsl
CYCLE971	C (M)	Permet d'étalonner un palpeur, de mesurer la longueur et/ou du rayon d'un outil (uniquement pour la technologie fraisage)		+		BNMsl
CYCLE973	C (M)	Permet d'étalonner un palpeur de pièce sur une surface de la pièce ou dans une rainure (uniquement pour la technologie tournage)		+		BNMsl
CYCLE974	C (M)	Permet de déterminer l'origine pièce dans l'axe de mesure sélectionné ou une correction d'outil avec une mesure en 1 point (uniquement pour la technologie tournage)		+		BNMsl
CYCLE976	C (M)	Permet d'étalonner entièrement un palpeur d'outil pour un axe et une direction donnés dans le plan de travail ou sur une arête (dans une bague étalon ou sur une sphère de calibrage)		+		BNMsl
CYCLE977	C (M)	Permet de déterminer le centre dans le plan, ainsi que la largeur ou le diamètre		+		BNMsl
CYCLE978	C (M)	Permet de mesurer la position d'une arête dans le système de coordonnées pièce		+		BNMsl
CYCLE979	C (M)	Permet de déterminer le centre dans le plan, ainsi que de mesurer le rayon de segments circulaires		+		BNMsl
CYCLE982	C (M)	Permet d'étalonner un palpeur d'outil, de mesurer les outils de tournage, de perçage et de fraisage (uniquement pour la technologie tournage)		+		BNMsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
CYCLE994	C (M)	Permet de déterminer l'origine pièce dans l'axe de mesure sélectionné avec une mesure en 2 points (uniquement pour la technologie tournage)		+		BNMsl
CYCLE995	C (M)	Permet de mesurer la déviation angulaire de la broche sur une machine-outil.		+		BNMsl
CYCLE996	C (M)	Permet de déterminer des données de transformation pertinentes pour les transformations cinématiques contenant des axes rotatifs.		+		BNMsl
CYCLE997	C (M)	Permet de déterminer le centre et le diamètre d'une sphère ainsi que les centres de trois sphères réparties		+		BNMsl
CYCLE998	C (M)	Permet de déterminer la position angulaire d'une surface (plan) par rapport au plan de travail, ainsi que l'angle des arêtes dans le système de coordonnées pièce.		+		BNMsl
CYCLE4071	C (T)	Rectification longitudinale avec pénétration au point d'inversion		+		PGAsl
CYCLE4072	C (T)	Rectification longitudinale avec pénétration au point d'inversion et signal d'abandon		+		PGAsl
CYCLE4073	C (T)	Rectification longitudinale avec pénétration continue		+		PGAsl
CYCLE4074	C (T)	Rectification longitudinale avec pénétration continue et signal d'abandon		+		PGAsl
CYCLE4075	C (T)	Rectification plane avec pénétration au point d'inversion		+		PGAsl
CYCLE4077	C (T)	Rectification plane avec pénétration au point d'inversion et signal d'abandon		+		PGAsl
CYCLE4078	C (T)	Rectification plane avec pénétration continue		+		PGAsl
CYCLE4079	C (T)	Rectification plane avec pénétration intermittente		+		PGAsl

Instructions D ... F

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1)2)3)4)5)] Explications voir Légende (Page 910).						
D	A	Numéro de correction d'outil		+		PGsl
D0	A	Pour D0, les corrections seront inopérantes pour l'outil en question		+		PGsl
DAC	K	Programmation au diamètre spécifique à l'axe non modale absolue	b	+		PGsl
DC	K	Cotation absolue pour axes rotatifs, accoster directement la position	b	+		PGsl

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1)2)3)4)5)] Explications voir Légende (Page 910).						
DCI	K	Affecter la classe de données I (= Individual) (SINUMERIK 828D uniquement !)		+		PGAsl
DCM	K	Affecter la classe de données M (= Manufacturer) (SINUMERIK 828D uniquement !)		+		PGAsl
DCU	K	Affecter la classe de données U (= User) (SINUMERIK 828D uniquement !)		+		PGAsl
DEF	K	Définition de variables		+		PGAsl
DEFAULT	K	Branche d'un branchement CASE		+		PGAsl
DEFINE	K	Mot-clé pour les définitions de macros		+		PGAsl
DELAYFSTOF	P	Définition de la fin d'une plage d'arrêt temporisé	m	+	-	PGAsl
DELAYFSTON	P	Définition du début d'une plage d'arrêt temporisé	m	+	-	PGAsl
DELDL	F	Suppression des corrections additives		+	-	PGAsl
DELDTG	P	Effacement de la distance restant à parcourir		-	+	FBSYsl
DELETE	P	Effacement du fichier indiqué. Le nom du fichier peut être indiqué avec chemin d'accès et extension de fichier.		+	-	PGAsl
DELMLOWNER	F	Suppression de l'emplacement de magasin propriétaire de l'outil		+	-	FBWsl
DELMLRES	F	Suppression de la réservation d'emplacement de magasin		+	-	FBWsl
DELMT	P	Suppression d'un multitool		+	-	FBWsl
DELOBJ	F	Suppression d'éléments de chaînes cinématiques, de zones de protection, d'éléments de zones de protection, de paires de collision et de données de transformation		+		PGAsl
DELT	P	Suppression d'un outil		+	-	FBWsl
DELTC	P	Suppression d'un bloc de données de l'organe porte-outil		+	-	FBWsl
DELTOOLENV	F	Suppression d'enregistrements de description de contournage d'outil		+	-	PGAsl
DIACYCOFA	K	Programmation au diamètre modale spécifique à l'axe : désactivée dans les cycles	m	+		FB1sl (P1)
DIAM90	G	Programmation au diamètre pour G90 ; programmation au rayon pour G91	m	+		PGAsl
DIAM90A	K	Programmation au diamètre modale spécifique à l'axe pour G90 et AC, programmation au rayon pour G91 et IC	m	+		PGsl
DIAMCHAN	K	Prise en compte de tous les axes des fonctions d'axe PM dans l'état du canal de la programmation au diamètre.		+		PGsl
DIAMCHANA	K	prise en compte de l'état de canal de la programmation de diamètre		+		PGsl
DIAMCYCOF	G	Programmation du diamètre spécifique au canal : désactivée dans les cycles	m	+		FB1sl (P1)

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1)2)3)4)5)] Explications voir Légende (Page 910).						
DIAMOF[6)]	G	Programmation du diamètre : désactivée état initial, voir le constructeur de machines	m	+		PGsl
DIAMOFA	K	Programmation modale du diamètre spécifique à l'axe : désactivée état initial, voir le constructeur de machines	m	+		PGsl
DIAMON	G	Programmation du diamètre : activé	m	+		PGsl
DIAMONA	K	Programmation modale du diamètre spécifique à l'axe : activée validation, voir le constructeur de machines	m	+		PGsl
DIC	K	Programmation au diamètre spécifique à l'axe non modale relative	b	+		PGsl
DILF	A	Distance de retrait (longueur)	m	+		PGsl
DISABLE	P	Désactivation d'une interruption		+	-	PGAsl
DISC	A	Surhaussement arc de raccordement correction du rayon d'outil	m	+		PGsl
DISCL	A	Distance entre le point final du mouvement de pénétration rapide et le plan d'usinage		+		PGsl
DISPLOF	PA	Inhibition de l'affichage du bloc courant		+		PGAsl
DISPLON	PA	Annulation de l'inhibition d'affichage du bloc courant		+		PGAsl
DISPR	A	Différence sur le contour pour repositionnement	b	+		PGAsl
DISR	A	Distance de repositionnement	b	+		PGAsl
DISRP	A	Distance entre le plan de retrait et le plan d'usinage lors d'un accostage et retrait en douceur		+		PGsl
DITE	A	Course en sortie de filetage	m	+		PGsl
DITS	A	Course en entrée de filetage	m	+		PGsl
DIV	K	Division avec résultat entier		+		PGAsl
DL	A	Sélection de la correction d'outil additive fonction de l'endroit (DL, correction de la somme, mise en place)	m	+		PGAsl
DO	A	Mot-clé pour action synchrone, déclenche une action lorsque la condition est remplie		-	+	FBSYsl
DRFOF	P	Désactivation (effacement) des décalages par manivelle (DRF)	m	+	-	PGsl
DRIVE	G	Accélération tangentielle fonction de la vitesse	m	+		PGAsl
DRIVEA	P	Activation de la caractéristique d'accélération avec coude pour les axes programmés		+	-	PGAsl
DYNFINISH	G	Dynamique pour superfinition	m	+		PGAsl
DYNNORM[6)]	G	Dynamique normale	m	+		PGAsl
DYNPOS	G	Dynamique pour mode positionnement, taraudage	m	+		PGAsl
DYNROUGH	G	Dynamique pour ébauche	m	+		PGAsl
DYNSEMIFIN	G	Dynamique pour finition	m	+		PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1)2)3)4)5)] Explications voir Légende (Page 910).						
DZERO	P	Déclare tous les numéros D de l'unité TO non valides		+	-	PGAsl
EAUTO	G	Détermination de la dernière section spline avec les 3 derniers points	m	+		PGAsl
EGDEF	P	Définition d'un réducteur électronique		+	-	PGAsl
EGDEL	P	Suppression de la définition de couplage pour l'axe asservi		+	-	PGAsl
EGOFC	P	Désactivation continue de l'entraînement électronique		+	-	PGAsl
EGOFS	P	Désactivation sélective de l'entraînement électronique		+	-	PGAsl
EGON	P	Activation du réducteur électronique		+	-	PGAsl
EGONSYN	P	Activation du réducteur électronique		+	-	PGAsl
EGONSYNE	P	Activation de l'entraînement électronique, avec prédéfinition du mode d'accostage		+	-	PGAsl
ELSE	K	Branchement si condition IF pas remplie		+		PGAsl
ENABLE	P	Activation d'une interruption		+	-	PGAsl
ENAT[6)]	G	Raccordement naturel avec le bloc de déplacement suivant	m	+		PGAsl
ENDFOR	K	Dernière ligne d'une boucle de comptage FOR		+		PGAsl
ENDIF	K	Dernière ligne d'un branchement IF		+		PGAsl
ENDLABEL	K	Repère de fin pour répétitions du programme pièce via REPEAT		+		PGAsl, FB1sl (K1)
ENDLOOP	K	Dernière ligne d'une boucle sans fin LOOP		+		PGAsl
ENDPROC	K	Dernière ligne d'un programme commençant par PROC		+		
ENDWHILE	K	Dernière ligne d'une boucle WHILE		+		PGAsl
ESRR	P	Paramétrage d'un retrait ESR contrôlé par le variateur dans le variateur		+		PGAsl
ESRS	P	Paramétrage d'un arrêt ESR contrôlé par le variateur dans le variateur		+		PGAsl
ETAN	G	Raccordement tangentiel avec le bloc de déplacement suivant en début de courbe de type spline	m	+		PGAsl
EVERY	K	Exécution d'une action synchrone lors de la transition de la condition de FALSE à TRUE		-	+	FBSYsl
EX	K	Mot-clé pour l'affectation en notation exponentielle		+		PGAsl
EXECSTRING	P	Transfert d'une variable STRING avec la ligne du programme pièce à exécuter		+	-	PGAsl
EXECTAB	P	Exécution d'un élément d'une table de déplacements		+	-	PGAsl
EXECUTE	P	Activation de l'exécution du programme		+	-	PGAsl
EXP	F	Fonction exponentielle ex		+	+	PGAsl
EXTCALL	A	Exécution de sous-programme externe		+	+	PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1)2)3)4)5)] Explications voir Légende (Page 910).						
EXTCLOSE	P	Fermeture de l'appareil / du fichier externe ouvert pour écriture		+	-	PGAsl
EXTERN	K	Déclaration d'un sous-programme avec transfert de paramètres		+		PGAsl
EXTOPEN	P	Ouverture pour écriture de l'appareil / du fichier externe pour le canal		+	-	PGAsl
F	A	Valeur d'avance (l'arrêt temporisé est programmé aussi avec F en liaison avec G4)		+	+	PGsl
FA	K	Avance axiale	m	+	+	PGsl
FAD	A	Avance d'approche pour accostage et retrait en douceur		+		PGsl
FALSE	K	Constante logique : faux		+	+	PGAsl
FB	A	Avance non modale		+		PGsl
FCTDEF	P	Définition d'une fonction polynôme		+	-	PGAsl
FCUB	G	Avance variable selon une courbe de type spline cubique	m	+		PGAsl
FD	A	Avance tangentielle pour correction par manivelle	b	+		PGsl
FDA	K	Avance axiale pour correction par manivelle	b	+		PGsl
FENDNORM[6)]	G	Décélération aux angles OFF	m	+		PGAsl
FFWOF[6)]	G	Désactivation de la commande anticipatrice	m	+		PGAsl
FFWON	G	Activation de la commande anticipatrice	m	+		PGAsl
FGREF	K	Rayon de référence pour les axes rotatifs ou facteurs de référence de trajectoire pour les axes d'orientation (interpolation vectorielle)	m	+		PGsl
FGROUP	P	Sélection de l'axe ou des axes avec avance tangentielle		+	-	PGsl
FI	K	Paramètre d'accès aux données frame : décalage fin		+		PGAsl
FIFOCTRL	G	Commande de la mémoire tampon d'exécution	m	+		PGAsl
FILEDATE	P	Fournit la date du dernier accès en écriture au fichier		+	-	PGAsl
FILEINFO	P	Fournit la somme de FILEDATE, FILESIZE, FILESTAT et FILETIME		+	-	PGAsl
FILESIZE	P	Fournit la taille actuelle du fichier		+	-	PGAsl
FILESTAT	P	Fournit l'état du fichier des droits lecture, écriture, exécution, affichage, suppression (rwxsd)		+	-	PGAsl
FILETIME	P	Fournit l'heure du dernier accès en écriture au fichier		+	-	PGAsl
FINEA	K	Fin du déplacement lors de l'atteinte de "Arrêt précis fin"	m	+		PGAsl
FL	K	Vitesse limite pour axes synchrones	m	+		PGsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1)2)3)4)5)] Explications voir Légende (Page 910).						
FLIN	G	Avance variable linéairement	m	+		PGAsl
FMA	K	Plusieurs avances axiales	m	+		PGsl
FNORM[6)]	G	Avance normale selon DIN66025	m	+		PGAsl
FOC	K	Limitation de couple/force à effet non modal	b	-	+	FBSYsl
FOCOF	K	Désactivation de la limitation de couple/force à effet modal	m	-	+	FBSYsl
FOCON	K	Activation de la limitation de couple/force à effet modal	m	-	+	FBSYsl
FOR	K	Boucle de comptage avec nombre fixe d'itérations		+		PGAsl
FP	A	Point fixe : numéro du point fixe à accoster	b	+		PGsl
FPO	K	Variation de l'avance programmée avec un polynôme		+		PGAsl
FPR	P	Identification axe rotatif		+	-	PGsl
FPRAOF	P	Désactivation de l'avance par tour		+	-	PGsl
FPRAON	P	Activation de l'avance par tour		+	-	PGsl
FRAME	K	Type de données pour la définition des systèmes de coordonnées		+		PGAsl
FRC	A	Avance pour rayon et chanfrein	b	+		PGsl
FRCM	A	Avance modale pour rayon et chanfrein	m	+		PGsl
FROM	K	L'action est exécutée lorsque la condition est remplie une fois et tant que l'action synchrone est active		-	+	FBSYsl
FTOC	P	Modifier la correction fine d'outil		-	+	FBSYsl
FTOCOF[6)]	G	Désactivation de la correction fine d'outil active en ligne	m	+		PGAsl
FTOCON	G	Activation de la correction fine d'outil active en ligne	m	+		PGAsl
FXS	K	Déplacement en butée activé	m	+	+	PGsl
FXST	K	Limite de couple pour l'accostage de butée	m	+	+	PGsl
FXSW	K	Fenêtre de surveillance pour l'accostage de butée		+	+	PGsl
FZ	K	Avance par dent	m	+		PGsl

Instructions G ... L

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
G0	G	Interpolation linéaire en rapide (déplacement à vitesse rapide)	m	+		PGsl
G1[6)]	G	Interpolation linéaire avec avance	m	+		PGsl
G2	G	Interpolation circulaire sens horaire	m	+		PGsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
G3	G	Interpolation circulaire sens antihoraire	m	+		PGsl
G4	G	Arrêt temporisé, prédéterminé dans le temps	b	+		PGsl
G5	G	Rectification en plongée oblique	b	+		PGAsl
G7	G	Mouvement compensatoire pour la rectification en plongée oblique	b	+		PGAsl
G9	G	Arrêt précis - réduction de vitesse	b	+		PGsl
G17[6)]	G	Sélection du plan de travail X/Y	m	+		PGsl
G18	G	Sélection du plan de travail Z/X	m	+		PGsl
G19	G	Sélection du plan de travail Y/Z	m	+		PGsl
G25	G	Limitation inférieure de la zone de travail	b	+		PGsl
G26	G	Limitation supérieure de la zone de travail	b	+		PGsl
G33	G	Filetage à pas constant	m	+		PGsl
G34	G	Filetage avec pas en progression linéaire	m	+		PGsl
G35	G	Filetage avec pas en régression linéaire	m	+		PGsl
G40[6)]	G	Désactivation de la correction du rayon d'outil	m	+		PGsl
G41	G	Correction du rayon d'outil à gauche du contour	m	+		PGsl
G42	G	Correction du rayon d'outil à droite du contour	m	+		PGsl
G53	G	Inhibition du décalage d'origine courant (non modal)	b	+		PGsl
G54	G	1er décalage d'origine réglable	m	+		PGsl
G55	G	2e décalage d'origine réglable	m	+		PGsl
G56	G	3e décalage d'origine réglable	m	+		PGsl
G57	G	4e décalage d'origine réglable	m	+		PGsl
G58 (840D sl)	G	Décalage d'origine absolu programmable (décalage grossier)	b	+		PGsl
G58 (828D)	G	5e décalage d'origine réglable	m	+		PGsl
G59 (840D sl)	G	Décalage d'origine additif programmable (décalage fin)	b	+		PGsl
G59 (828D)	G	6e décalage d'origine réglable	m	+		PGsl
G60[6)]	G	Arrêt précis - réduction de vitesse	m	+		PGsl
G62	G	Décélération aux angles intérieurs avec correction du rayon d'outil activée (G41, G42)	m	+		PGAsl
G63	G	Taraudage avec porte-taraud compensateur	b	+		PGsl
G64	G	Contournage	m	+		PGsl
G70	G	Cotation en inch pour indications géométriques (longueurs)	m	+	+	PGsl
G71[6)]	G	Cotation métrique pour indications géométriques (longueurs)	m	+	+	PGsl
G74	G	Accostage point de référence	b	+		PGsl
G75	G	Accostage d'un point fixe	b	+		PGsl
G90[6)]	G	Introduction des cotes en valeurs absolues	m/b	+		PGsl
G91	G	Indication de cotes relatives	m/b	+		PGsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
G93	G	Avance en sens inverse du temps tr/min	m	+		PGsl
G94[6)]	G	Avance linéaire F en mm/min ou inch/min et degré/min	m	+		PGsl
G95	G	Avance par tour F en mm/tr, inch/tr	m	+		PGsl
G96	G	Avance par tour (comme pour G95) et vitesse de coupe constante	m	+		PGsl
G97	G	Avance par tour et vitesse de rotation de broche constante (vitesse de coupe constante désactivée)	m	+		PGsl
G110	G	Programmation du pôle par rapport à la dernière position de consigne programmée	b	+		PGsl
G111	G	Programmation du pôle par rapport à l'origine pièce courante	b	+		PGsl
G112	G	Programmation du pôle par rapport au dernier pôle valable	b	+		PGsl
G140[6)]	G	Sens d'accostage ARD fixé par G41/G42	m	+		PGsl
G141	G	Sens d'accostage ARD à gauche du contour	m	+		PGsl
G142	G	Sens d'accostage ARD à droite du contour	m	+		PGsl
G143	G	Sens d'accostage ARD en fonction tangente	m	+		PGsl
G147	G	Accostage en douceur en ligne droite	b	+		PGsl
G148	G	Retrait en douceur en ligne droite	b	+		PGsl
G153	G	Inhibition des frames courants, y compris le frame de base	b	+		PGsl
G247	G	Accostage en douceur en quart de cercle	b	+		PGsl
G248	G	Retrait en douceur en quart de cercle	b	+		PGsl
G290[6)]	G	Basculement vers mode SINUMERIK en MARCHE	m	+		FBWsl
G291	G	Basculement vers mode ISO2/3 en MARCHE	m	+		FBWsl
G331	G	Taraudage sans porte-taraud compensateur, pas positif, filetage à droite	m	+		PGsl
G332	G	Taraudage sans porte-taraud compensateur, pas négatif, filetage à gauche	m	+		PGsl
G335	G	Tournage d'un filetage bombé dans le sens horaire	m	+		PGsl
G336	G	Tournage d'un filetage bombé dans le sens antihoraire	m	+		PGsl
G340[6)]	G	Accostage dans l'espace (simultanément en profondeur et dans le plan (hélice))	m	+		PGsl
G341	G	D'abord approche dans l'axe perpendiculaire (z), puis accostage dans le plan	m	+		PGsl
G347	G	Accostage en douceur en demi-cercle	b	+		PGsl
G348	G	Retrait en douceur en demi-cercle	b	+		PGsl
G450[6)]	G	Arc de raccordement	m	+		PGsl
G451	G	Point d'intersection des équidistantes	m	+		PGsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
G460[6)]	G	Activation de la détection des violations du contour pour le bloc d'accostage et de retrait	m	+		PGsl
G461	G	Insertion d'un cercle dans le bloc CRO	m	+		PGsl
G462	G	Insertion d'une droite dans le bloc CRO	m	+		PGsl
G500[6)]	G	Désactivation de tous les frames réglables, les frames de base sont actifs	m	+		PGsl
G505 ... G599	G	5. ... 99. décalage d'origine réglable	m	+		PGsl
G601[6)]	G	Changement de bloc lors arrêt précis fin	m	+		PGsl
G602	G	Changement de bloc lors arrêt précis grossier	m	+		PGsl
G603	G	Changement de bloc en fin de bloc IPO	m	+		PGsl
G621	G	Décélération à tous les angles	m	+		PGAsl
G641	G	Contournage avec arrondissement selon le critère de trajet (= distance de transition programmable)	m	+		PGsl
G642	G	Contournage avec arrondissement avec prise en compte des tolérances définies	m	+		PGsl
G643	G	Contournage avec arrondissement avec prise en compte des tolérances définies (internes au bloc)	m	+		PGsl
G644	G	Contournage avec arrondissement utilisant la dynamique maximale possible	m	+		PGsl
G645	G	Contournage avec arrondissement d'angles et changements de blocs tangentiels avec prise en compte de tolérances définies	m	+		PGsl
G700	G	Cotation en inch pour indications géométriques et technologiques (longueurs, avance)	m	+	+	PGsl
G710[6)]	G	Cotation métrique pour indications géométriques et technologiques (longueurs, avance)	m	+	+	PGsl
G810[6)], ..., G819	G	Groupe G réservé pour l'utilisateur OEM		+		PGAsl
G820[6)], ..., G829	G	Groupe G réservé pour l'utilisateur OEM		+		PGAsl
G931	G	Indication de l'avance par durée du déplacement, désactiver la vitesse tangentielle constante	m	+		
G942	G	Gel de l'avance linéaire et de la vitesse de coupe constante ou de la vitesse de rotation de broche	m	+		
G952	G	Gel de l'avance par tour et de la vitesse de coupe constante ou de la vitesse de rotation de broche	m	+		
G961	G	Avance linéaire (comme pour G94) et vitesse de coupe constante	m	+		PGsl
G962	G	Avance linéaire ou avance par tour et vitesse de coupe constante	m	+		PGsl
G971	G	Avance linéaire et vitesse de rotation de broche constante (vitesse de coupe constante désactivée)	m	+		PGsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
G972	G	Avance linéaire ou avance par tour et vitesse de rotation de broche constante (vitesse de coupe constante désactivée)	m	+		PGsl
G973	G	Avance par tour sans limitation de vitesse de la broche et vitesse de rotation de broche constante (G97 sans LIMS pour mode ISO)	m	+		PGsl
GEOAX	P	Affecter de nouveaux axes de canal aux axes géométriques 1 à 3		+	-	PGAsl
GET	P	Échange de l'axe libéré entre des canaux		+	+	PGAsl
GETACTT	F	Détermination de l'outil actif parmi un groupe d'outils de même nom		+	-	FBWsl
GETACTTD	F	Détermination du numéro T correspondant à un numéro D absolu		+	-	PGAsl
GETD	P	Échange direct de l'axe entre des canaux		+	-	PGAsl
GETDNO	F	Fournit le numéro D d'un tranchant (CE) d'un outil (T)		+	-	PGAsl
GETEXET	P	Lecture du numéro T changé		+	-	FBWsl
GETFREELOC	P	Recherche d'un emplacement vide dans les magasins pour un outil donné		+	-	FBWsl
GETSELT	P	Délivrer le numéro T présélectionné		+	-	FBWsl
GETT	F	Déterminer le numéro T correspondant à un nom d'outil		+	-	FBWsl
GETTCOR	F	Lecture de longueurs d'outil ou de composantes de longueurs d'outil		+	-	PGAsl
GETTENV	F	Lecture des numéros T, D et DL		+	-	PGAsl
GETVARAP	F	Lecture du droit d'accès à une variable système/utilisateur		+	-	PGAsl
GETVARDFT	F	Lecture de la valeur par défaut d'une variable système/utilisateur		+	-	PGAsl
GETVARLIM	F	Lecture de la valeur limite d'une variable système/utilisateur		+	-	PGAsl
GETVARPHU	F	Lecture de l'unité physique d'une variable système/utilisateur		+	-	PGAsl
GETVARTYP	F	Lecture du type de données d'une variable système/utilisateur		+	-	PGAsl
GFRAME0 ... GFRAME100	G	Activation du frame de rectification <n> de la gestion de données dans le canal	m	+		PGsl
GOTO	K	Instruction de saut avec destination en aval puis en amont (vers fin puis vers début de programme)		+		PGAsl
GOTOB	K	Saut avec destination en amont (vers début de programme)		+		PGAsl
GOTOC	K	Comme GOTO, mais avec suppression de l'alarme 14080 "Destination de saut pas trouvée"		+		PGAsl

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
GOTOF	K	Saut avec destination en aval (vers fin programme)		+		PGAsl
GOTOS	K	Retour au début du programme		+		PGAsl
GP	K	Mot clé pour la programmation indirecte d'attributs de positions		+		PGAsl
GROUP_ ADDEND	C (T)	Fin du supplément de mise au point		+		PGAsl
GROUP_BEGIN	C (T)	Début du bloc de programme		+		PGAsl
GROUP_END	C (T)	Fin du bloc de programme		+		PGAsl
GWPSOF	P	Désactiver vitesse périphérique de meule constante (VPM)	b	+	-	PGsl
GWPSON	P	Activer vitesse périphérique de meule constante (VPM)	b	+	-	PGsl
H...	A	Sortie de fonction auxiliaire à l'AP		+	+	PGsl/FB1sl (H2)
HOLES1	C (T)	Rangée de trous		+		PGAsl
HOLES2	C (T)	Cercle de trous		+		PGAsl
I	A	Paramètres d'interpolation	b	+		PGsl
I1	A	Coordonnée de point intermédiaire	b	+		PGsl
IC	K	Saisie de cotes relatives	b	+		PGsl
ICYCOF	P	Exécution de tous les blocs d'un cycle technologique dans une période d'appel de l'interpolateur après ICYCOF		+	+	FBSYsl
ICYCON	P	Exécution de chaque bloc d'un cycle technologique dans une période d'appel de l'interpolateur distincte après ICYCON		+	+	FBSYsl
ID	K	Identification pour actions synchrones modales	m	-	+	FBSYsl
IDS	K	Identification pour actions synchrones statiques modales		-	+	FBSYsl
IF	K	Début d'un saut conditionnel dans le programme pièce / cycle technologique		+	+	PGAsl
INDEX	F	Détermination de l'indice d'un caractère dans une chaîne de caractères		+	-	PGAsl
INICF	K	Initialisation des variables pour NEWCONF		+		PGAsl
INIPO	K	Initialisation des variables pour PowerOn		+		PGAsl
INIRE	K	Initialisation des variables pour Reset		+		PGAsl
INIT	P	Sélection d'un programme CN spécifié pour exécution dans un canal déterminé		+	-	PGAsl
INITIAL		Génération d'un fichier INI pour tous les domaines		+		PGAsl
INT	K	Type de données : valeur entière avec signe		+		PGAsl
INTERSEC	F	Calculer le point d'intersection entre deux éléments de contour		+	-	PGAsl
INVCCW	G	Déplacer développante dans le sens antihoraire	m	+		PGsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
INVCW	G	Déplacer développante dans le sens horaire	m	+		PGsl
INVFRAME	F	Calculer le frame inverse à partir d'un frame		+	-	FB1sl (K2)
IP	K	Paramètre d'interpolation variable		+		PGAsl
IPOBRKA	P	Critère de déplacement à partir du point de déclenchement de la rampe de freinage	m	+	+	
IPOENDA	K	Fin du déplacement lors de "Arrêt interpolateur"	m	+		PGAsl
IPTRLOCK	P	Gel du début de la section de programme interdit à la recherche sur le bloc de fonction machine suivant.	m	+	-	PGAsl
IPTRUNLOCK	P	Définition de la fin de la section de programme interdit à la recherche sur le bloc courant du moment d'interruption.	m	+	-	PGAsl
IR	A	Coordonnée du centre d'un cercle (en direction X) lors du tournage d'un filetage bombé		+		PGsl
ISAXIS	F	Vérifier si l'axe géométrique 1 indiqué comme paramètre existe		+	-	PGAsl
ISD	A	Profondeur de pénétration	m	+		PGAsl
ISFILE	F	Vérifier s'il existe un fichier dans la mémoire d'application CN		+	-	PGAsl
ISNUMBER	F	Vérifier si une chaîne de caractères peut être convertie en un nombre		+	-	PGAsl
ISOCALL	K	Appel indirect d'un programme programmé en langage ISO		+		PGAsl
ISVAR	F	Vérifier si le paramètre de transfert contient une variable connue de la CN		+	-	PGAsl
J	A	Paramètres d'interpolation	b	+		PGsl
J1	A	Coordonnée de point intermédiaire	b	+		PGsl
JERKA	P	Activer, pour les axes programmés, le comportement à l'accélération réglé à l'aide du PM		+	-	
JERKLIM	K	Réduction ou surhaussement de l'à-coup axial maximal	m	+		PGAsl
JERKLIMA	K	Réduction ou surhaussement de l'à-coup axial maximal	m	+	+	PGAsl
JR	A	Coordonnée du centre d'un cercle (en direction Y) lors du tournage d'un filetage bombé		+		PGsl
K	A	Paramètres d'interpolation	b	+		PGsl
K1	A	Coordonnée de point intermédiaire	b	+		PGsl
KONT	G	Contournement du contour lors de la correction d'outil	m	+		PGsl
KONTC	G	Accostage / retrait avec polynôme à courbure continue	m	+		PGsl
KONTT	G	Accostage / retrait avec polynôme tangentiel	m	+		PGsl
KR	A	Coordonnée du centre d'un cercle (en direction Z) lors du tournage d'un filetage bombé		+		PGsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
L	A	Numéro de sous-programme	b	+	+	PGAsl
LEAD	A	Angle d'avance 1re orientation d'outil 2e polynôme d'orientation	m	+		PGAsl
LEADOF	P	Couplage organe pilote d'axe désactivé		+	+	PGAsl
LEADON	P	Couplage organe pilote d'axe activé		+	+	PGAsl
LENTOAX	F	Fournit des informations sur l'affectation des longueurs L1, L2 et L3 de l'outil actif à l'abscisse, à l'ordonnée et à la valeur		+	-	PGAsl
LFOF[6)]	G	Désactivation du retrait rapide pour le filetage	m	+		PGsl
LFON	G	Activation du retrait rapide pour le filetage	m	+		PGsl
LFPOS	G	Retrait de l'axe introduit avec POLFMASK ou POLFMLIN sur la position d'axe absolue programmée avec POLF	m	+		PGsl
LFTXT[6)]	G	Plan du déplacement de retrait rapide lors du relèvement rapide déterminé par la tangente de la trajectoire et la direction actuelle de l'outil	m	+		PGsl
LFWP	G	Plan du déplacement de retrait rapide lors du relèvement rapide déterminé par le plan de travail actuel (G17/G18/G19)	m	+		PGsl
LIFTFAST	K	Relèvement rapide		+		PGsl
LIMS	K	Limitation de la vitesse de rotation pour G96/G961 et G97	m	+		PGsl
LLI	K	Valeur limite inférieure de variables		+		PGAsl
LN	F	Logarithme naturel		+	+	PGAsl
LOCK	P	Bloquer une action synchrone avec ID (bloquer un cycle technologique)		-	+	FBSYsl
LONGHOLE	C (T)	Trou oblong		+		PGAsl
LOOP	K	Début d'une boucle sans fin		+		PGAsl

Instructions M ... R

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
M0		Arrêt programmé		+	+	PGsl
M1		Arrêt facultatif		+	+	PGsl
M2		Fin du programme principal (comme M30)		+	+	PGsl
M3		Sens de rotation de broche à droite		+	+	PGsl
M4		Sens de rotation de broche à gauche		+	+	PGsl
M5		Arrêt de la broche		+	+	PGsl
M6		Changement d'outil		+	+	PGsl
M17		Fin du sous-programme		+	+	PGsl

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
M19		Positionnement de la broche sur la position inscrite dans le SD43240		+	+	PGsl
M30		Fin du programme principal (comme M2)		+	+	PGsl
M40		Changement automatique de rapport de boîte de vitesses		+	+	PGsl
M41 ... M45		Rapport de boîte de vitesses 1 ... 5		+	+	PGsl
M70		Basculement dans mode axe		+	+	PGsl
MASLDEF	P	Définition du couplage d'axes maître/esclave		+	+	PGAsl
MASLDEL	P	Découplage maître/esclave et suppression de la définition du couplage		+	+	PGAsl
MASLOF	P	Arrêter un couplage temporaire		+	+	PGAsl
MASLOFS	P	Désactivation d'un couplage temporaire avec arrêt automatique de l'axe esclave		+	+	PGAsl
MASLON	P	Enclencher un couplage temporaire		+	+	PGAsl
MATCH	F	Rechercher une chaîne dans une chaîne		+	-	PGAsl
MAXVAL	F	Valeur supérieure de deux variables (fonction trigonométrique)		+	+	PGAsl
MCALL	K	Appel modal d'un sous-programme		+		PGAsl
MEAC	K	Mesure axiale continue sans effacement de la distance restant à parcourir	b	+	+	PGAsl
MEAFRAME	F	Calcul d'un frame à partir de points de mesure		+	-	PGAsl
MEAS	A	Mesure avec effacement de la distance restant à parcourir	b	+		PGAsl
MEASA	K	Mesure axiale avec effacement de la distance restant à parcourir	b	+	+	PGAsl
MEASURE	F	Méthode de calcul pour la mesure de pièce et la mesure d'outil		+	-	FB1sl (M5)
MEAW	A	Mesure sans effacement de la distance restant à parcourir	b	+		PGAsl
MEAWA	K	Mesure axiale sans effacement de la distance restant à parcourir	b	+	+	PGAsl
MI	K	Accès aux données frame : fonction miroir		+		PGAsl
MINDEX	F	Détermination de l'indice d'un caractère dans une chaîne de caractères		+	-	PGAsl
MINVAL	F	Valeur inférieure de deux variables (fonction trigonométrique)		+	+	PGAsl
MIRROR	G	Fonction miroir programmable	b	+		PGAsl
MMC	P	Appel interactif de boîtes de dialogue à partir du programme pièce pour affichage sur l'IHM		+	-	PGAsl
MOD	K	Division modulo		+		PGAsl
MODAXVAL	F	Détermination de la position modulo d'un axe rotatif modulo		+	-	PGAsl
MOV	K	Démarrage de l'axe de positionnement		-	+	FBSYsl
MOVT	A	Indication du point final d'un déplacement dans le sens de l'outil				FB1(K2)

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
MSG	P	Messages programmables	m	+	-	PGsl
MVTOOL	P	Instruction de langage pour le déplacement d'un outil		+	-	FBWsl
N	A	Numéro de bloc CN complémentaire		+		PGsl
NAMETOINT	F	Détermination de l'indice de variables système		+		PGAsl
NC	K	Spécification du domaine de validité de données		+		PGAsl
NEWCONF	P	Application des paramètres machine modifiés (correspond à "Activation des paramètres machine")		+	-	PGAsl
NEWMT	F	Création d'un nouveau multitool		+	-	FBWsl
NEWT	F	Création d'un nouvel outil		+	-	FBWsl
NORM[6)]	G	Positionnement à la normale au point de départ / final en correction d'outil	m	+		PGsl
NOT	K	NON logique (Négation)		+		PGAsl
NPROT	P	Activation/désactivation d'une zone de protection spécifique à la machine		+	-	PGAsl
NPROTDEF	P	Définition d'une zone de protection spécifique à la machine		+	-	PGAsl
NUMBER	F	Convertir une chaîne de caractères en un nombre		+	-	PGAsl
OEMIPO1	G	Interpolation OEM 1	m	+		PGAsl
OEMIPO2	G	Interpolation OEM 2	m	+		PGAsl
OF	K	Mot-clé d'un branchement CASE		+		PGAsl
OFFN	A	Surépaisseur d'usinage par rapport au contour programmé	m	+		PGsl
OMA1	A	Adresse OEM 1	m	+		PGAsl
OMA2	A	Adresse OEM 2	m	+		PGAsl
OMA3	A	Adresse OEM 3	m	+		PGAsl
OMA4	A	Adresse OEM 4	m	+		PGAsl
OMA5	A	Adresse OEM 5	m	+		PGAsl
OR	K	Opérateur logique, combinaison OU		+		PGAsl
ORIAXES	G	Interpolation linéaire des axes machine ou des axes d'orientation	m	+		PGAsl
ORIAXPOS	G	Angles d'orientation définis par axes virtuels d'orientation avec positions d'axe rotatif	m	+		PGAsl
ORIC[6)]	G	Les changements d'orientation aux angles saillants sont superposés au bloc circulaire à insérer	m	+		PGAsl
ORICONCCW	G	Interpolation sur une enveloppe de cercle dans le sens antihoraire	m	+		PGAsl/FB3sl (F3)
ORICONCW	G	Interpolation sur une enveloppe de cercle dans le sens horaire	m	+		PGAsl/FB3sl (F4)
ORICONIO	G	Interpolation sur une enveloppe de cercle avec indication d'une orientation intermédiaire	m	+		PGAsl/FB3sl (F4)

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
ORICONTO	G	Interpolation sur une surface circulaire avec transition tangentielle (indication de l'orientation finale)	m	+		PGAsl/FB3sl (F5)
ORICURVE	G	Interpolation de l'orientation avec indication du mouvement de deux points de contact de l'outil	m	+		PGAsl/FB3sl (F6)
ORID	G	Les changements d'orientation se font avant l'exécution du bloc circulaire inséré	m	+		PGAsl
ORIEULER[6)]	G	Orientation définie par angles d'Euler	m	+		PGAsl
ORIMKS	G	Orientation de l'outil dans le système de coordonnées machine	m	+		PGAsl
ORIPATH	G	Orientation de l'outil par rapport à la trajectoire	m	+		PGAsl
ORIPATHS	G	Orientation de l'outil par rapport à la trajectoire, lissage de coude dans le tracé d'orientation	m	+		PGAsl
ORIPLANE	G	Interpolation dans un plan (correspond à ORIVECT) Interpolation circulaire de grand rayon	m	+		PGAsl
ORIRESET	P	Position initiale de l'orientation de l'outil avec jusqu'à 3 axes d'orientation		+	-	PGAsl
ORIROTA[6)]	G	Angle de rotation rapporté à un sens de rotation en indication absolue	m	+		PGAsl
ORIROTC	G	Vecteur de rotation tangentiel à la tangente à la trajectoire	m	+		PGAsl
ORIROTR	G	Angle de rotation relatif par rapport au plan entre orientation de départ et orientation finale	m	+		PGAsl
ORIROTT	G	Angle de rotation relatif par rapport à la modification du vecteur d'orientation	m	+		PGAsl
ORIRPY	G	Orientation définie par angles RPY (XYZ)	m	+		PGAsl
ORIRPY2	G	Orientation définie par angles RPY (ZYX)	m	+		PGAsl
ORIS	A	Modification d'orientation	m	+		PGAsl
ORISOF[6)]	G	Lissage du tracé d'orientation désactivé	m	+		PGAsl
ORISOLH	F	Calcul des orientations		+		PGAsl
ORISON	G	Lissage du tracé d'orientation activé	m	+		PGAsl
ORIVECT[6)]	G	Interpolation circulaire de grand rayon (identique avec ORIPLANE)	m	+		PGAsl
ORIVIRT1	G	Angles d'orientation définis par axes virtuels d'orientation (définition 1)	m	+		PGAsl
ORIVIRT2	G	Angles d'orientation définis par axes virtuels d'orientation (définition 1)	m	+		PGAsl
ORIWKS[6)]	G	Orientation de l'outil dans le système de coordonnées pièce	m	+		PGAsl
OS	K	Activation/désactivation de l'oscillation		+		PGAsl
OSB	K	Oscillation : point de départ	m	+		FB1sl (P5)
OSC	G	Lissage constant de l'orientation de l'outil	m	+		PGAsl
OSCILL	K	Axis : 1 à 3 axes de pénétration	m	+		PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
OSCTRL	K	Options d'oscillation	m	+		PGAsl
OSD	G	Arrondissement de l'orientation de l'outil via la spécification de la longueur d'arrondissement avec SD	m	+		PGAsl
OSE	K	Point d'arrêt de l'oscillation	m	+		PGAsl
OSNSC	K	Oscillation : nombre d'extractions par étincelage	m	+		PGAsl
OSOF[6)]	G	Désactivation du lissage de l'orientation de l'outil	m	+		PGAsl
OSP1	K	Oscillation : point d'inversion gauche	m	+		PGAsl
OSP2	K	Oscillation : point d'inversion droit	m	+		PGAsl
OSS	G	Lissage de l'orientation de l'outil en fin de bloc	m	+		PGAsl
OSSE	G	Lissage de l'orientation de l'outil en début et en fin de bloc	m	+		PGAsl
OST	G	Arrondissement de l'orientation de l'outil via la spécification de la tolérance angulaire en degré avec SD (écart maximal de l'allure de variation de l'orientation programmée)	m	+		PGAsl
OST1	K	Oscillation : point d'arrêt au point d'inversion gauche	m	+		PGAsl
OST2	K	Oscillation : point d'arrêt au point d'inversion droit	m	+		PGAsl
OTOL	A	Tolérance d'orientation pour fonctions de compression, lissage de l'orientation et modes d'arrondissement	m	+		PGAsl
OVR	K	Correction de vitesse de rotation	m	+		PGAsl
OVRA	K	Correction de vitesse de rotation axiale	m	+	+	PGAsl
OVRRAP	K	Correction du rapide	m	+		PGAsl
P	A	Nombre d'exécutions du sous-programme		+		PGAsl
PAROT	G	Orientation du système de coordonnées pièce sur la pièce	m	+		PGsl
PAROTOF[6)]	G	Désactiver la rotation de frame par rapport à l'outil	m	+		PGsl
PCALL	K	Appel d'un sous-programme avec chemin d'accès absolu et transfert de paramètres.		+		PGAsl
PDELAYOF	G	Poinçonnage avec temporisation désactivé	m	+		PGAsl
PDELAYON[6)]	G	Poinçonnage avec temporisation activé	m	+		PGAsl
PHI	K	Angle de rotation de l'orientation autour de l'axe d'orientation du cône		+		PGAsl
PHU	K	Unité physique d'une variable		+		PGAsl
PL	A	1er spline B distance entre les pôles 2e interpolation polynomiale longueur de l'intervalle des paramètres pour l'interpolation polynomiale	b	+		PGAsl
PM	K	Par minute		+		PGsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
PO	K	Coefficient polynomial pour l'interpolation polynomiale	b	+		PGAsl
POCKET3	C (T)	Fraisage d'une poche rectangulaire		+		PGAsl
POCKET4	C (T)	Fraisage d'une poche circulaire		+		PGAsl
POLF	K	Position de retrait LIFTFAST	m	+		PGsl/PGAsl
POLFA	P	Lancement de la position de retrait des axes individuels avec $AA_ESR_TRIGGER	m	+	+	PGsl
POLFMASK	P	Libération des axes pour le retrait sans corrélation entre les axes	m	+	-	PGsl
POLFMLIN	P	Libération des axes pour le retrait avec corrélation linéaire entre les axes	m	+	-	PGsl
POLY	G	Interpolation polynomiale	m	+		PGAsl
POLYPATH	P	Interpolation polynomiale sélectionnable pour les deux groupes d'axes AXIS ou VECT	m	+	-	PGAsl
PON	G	Poinçonnage ACTIVE	m	+		PGAsl
PONS	G	Poinçonnage activé à la période d'appel de l'interpolateur	m	+		PGAsl
POS	K	Positionnement d'un axe		+	+	PGsl
POSA	K	Positionnement d'un axe sur plusieurs blocs		+	+	PGsl
POSM	P	Positionnement d'un magasin		+	-	FBWsl
POSMT	P	Positionnement du multitool sur le porte-outil au numéro d'emplacement		+	-	FBWsl
POSP	K	Positionnement en trajets partiels (oscillation)		+		PGsl
POSRANGE	F	Déterminer si la position de consigne courante interpolée d'un axe se trouve dans une fenêtre autour d'une position de référence prescrite		+	+	FBSYsl
POT	F	Puissance 2 (fonction arithmétique)		+	+	PGAsl
PR	K	Par rotation		+		PGsl
PREPRO	PA	Identifier les sous-programmes avec prétraitement		+		PGAsl
PRESETON	P	Forçage de valeur réelle avec perte de l'état de référencement		+	+	PGAsl
PRESETONS	P	Forçage de valeur réelle sans perte de l'état de référencement		+	+	PGAsl
PRIO	K	Mot-clé pour la définition de priorités lors du traitement d'interruptions		+		PGAsl
PRLOC	K	Initialisation des variables lors d'un Reset uniquement après modification locale		+		PGAsl
PROC	K	Première instruction d'un programme		+		PGAsl
PROTA	P	Demander un recalcul du modèle de collision		+		PGAsl
PROTD	F	Calcul de la distance entre deux zones de protection		+		PGAsl
PROTS	P	Réglage de l'état des zones de protection		+		PGAsl

Instruction	Type[1]	Signification	E[2]	PP[3]	AS[4]	Description, voir[5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
PSI	K	Angle d'ouverture du cône		+		PGAsl
PTP	G	Déplacement point à point (PTP)	m	+		PGAsl
PTPG0	G	Déplacement point à point uniquement pour G0, sinon déplacement avec interpolation CP	m	+		PGAsl
PTPWOC	G	Déplacement point à point sans déplacement de compensation causés par un changement d'orientation	m	+		PGAsl
PUNCHACC	P	Accélération asservie au grignotage		+	-	PGAsl
PUTFTOC	P	Correction fine d'outil pour dressage simultané		+	-	PGAsl
PUTFTOCF	P	Correction fine d'outil selon fonction définie avec FCTDEF pour dressage simultané		+	-	PGAsl
PW	A	Spline B (poids ponctuel)	b	+		PGAsl
QU	K	Sortie rapide de fonction supplémentaire (auxiliaire)		+		PGsl
R...	A	Paramètre de calcul également comme descripteur d'adresse paramétrable et avec extension numérique		+		PGAsl
RAC	K	Programmation au rayon spécifique à l'axe non modale absolue	b	+		PGsl
RDISABLE	P	Blocage de l'introduction via l'interface.		-	+	FBSYsl
READ	P	Permet de lire une ou plusieurs lignes dans le fichier indiqué, les informations lues étant rangées dans un tableau		+	-	PGAsl
REAL	K	Type de données : variable à virgule flottante avec signe (nombres réels)		+		PGAsl
REDEF	K	Redéfinition de variables système, variables utilisateur et instruction de langage CN		+		PGAsl
RELEASE	P	Libérer axes machine pour l'échange		+	+	PGAsl
REP	K	Mot-clé pour initialisation de tous les éléments d'un tableau avec la même valeur		+		PGAsl
REPEAT	K	Itération d'une boucle		+		PGAsl
REPEATB	K	Itération d'une ligne de programme		+		PGAsl
REPOSA	G	Réaccostage linéaire du contour avec tous les axes	b	+		PGAsl
REPOSH	G	Réaccostage du contour avec demi-cercle	b	+		PGAsl
REPOSHA	G	Réaccostage du contour avec tous les axes ; axes géométriques en demi-cercle	b	+		PGAsl
REPOSL	G	Réaccostage linéaire du contour	b	+		PGAsl
REPOSQ	G	Réaccostage du contour en quart de cercle	b	+		PGAsl
REPOSQA	G	Réaccostage linéaire du contour avec tous les axes ; axes géométriques en quart de cercle	b	+		PGAsl
RESETMON	P	Instruction de langage pour activation de la consigne		+	-	FBWsl
RET	P	Fin de sous-programme		+	+	PGAsl
RETB	P	Fin de sous-programme		+	+	PGAsl

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
RIC	K	Programmation au rayon spécifique à l'axe non modale relative	b	+		PGsl
RINDEX	F	Détermination de l'indice d'un caractère dans une chaîne de caractères		+	-	PGAsl
RMB	G	Réaccostage du point de début de bloc	m	+		PGAsl
RMBBL	G	Réaccostage du point de début de bloc	b	+		PGAsl
RME	G	Réaccostage du point final du bloc	m	+		PGAsl
RMEBL	G	Réaccostage du point final du bloc	b	+		PGAsl
RMI[6)]	G	Réaccostage du point d'interruption	m	+		PGAsl
RMIBL [6)]	G	Réaccostage du point d'interruption	b	+		PGAsl
RMN	G	Réaccostage du point de contour le plus proche	m	+		PGAsl
RMNBL	G	Réaccostage du point de contour le plus proche	b	+		PGAsl
RND	A	Arrondir un angle	b	+		PGsl
RNDM	A	Arrondissement modal	m	+		PGsl
ROT	G	Rotation programmable	b	+		PGsl
ROTS	G	Programmation de rotations de frames avec des angles solides	b	+		PGsl
ROUND	F	Arrondissement des décimales		+	+	PGAsl
ROUNDUP	F	Arrondissement d'une valeur d'entrée		+	+	PGAsl
RP	A	Rayon polaire	m/b	+		PGsl
RPL	A	Rotation dans le plan	b	+		PGsl
RT	K	Paramètre d'accès aux données frame : rotation		+		PGAsl
RTLIOF	G	G0 sans interpolation linéaire (interpolation axiale individuelle)	m	+		PGsl
RTLION[6)]	G	G0 avec interpolation linéaire	m	+		PGsl

Instructions S ... Z

Instruction	Type[1)]	Signification	E[2)]	PP[3)]	AS[4)]	Description, voir[5)]
[1) 2) 3) 4) 5)] Explications voir Légende (Page 910).						
S	A	Vitesse de rotation de broche (pour G4, G96/G961, autre signification)	m/b	+	+	PGsl
SAVE	PA	Attribut pour sauvegarde d'informations en cas d'appel de sous-programme		+		PGAsl
SBLOF	P	Inhibition du bloc par bloc		+	-	PGAsl
SBLON	P	Annulation de l'inhibition du bloc par bloc		+	-	PGAsl
SC	K	Paramètre d'accès aux données frame : facteur d'échelle		+		PGAsl
SCALE	G	Facteur d'échelle programmable	b	+		PGsl

Instruction	Type [1]	Signification	E [2]	PP [3]	AS [4]	Description, voir [5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
SCC	K	Affectation sélective d'un axe transversal pour G96/G961/G962. Les descripteurs d'axes peuvent être des axes géométriques, des axes de canal ou des axes machine.		+		PGsl
SCPARA	K	Programmation du jeu de paramètres servo		+	+	PGAsl
SD	A	Degré de spline	b	+		PGAsl
SET	K	Mot-clé pour initialisation de tous les éléments d'un tableau avec les valeurs mentionnées		+		PGAsl
SETAL	P	Activation d'une alarme		+	+	PGAsl
SETDNO	F	Affectation du numéro D du tranchant (CE) d'un outil (T)		+	-	PGAsl
SETINT	K	Définir la routine d'interruption qui doit être activée lorsqu'une entrée CN devient active		+		PGAsl
SETM	P	Définition de repères dans le canal spécifique		+	+	PGAsl
SETMS	P	Retour à la broche maître définie dans le paramètre machine		+	-	PGsl
SETMS(n)	P	La broche n doit devenir broche maître		+		PGsl
SETMTH	P	Définir le numéro du porte-outil maître		+	-	FBWsl
SETPIECE	P	Tenir compte du nombre de pièces pour tous les outils assignés à la broche		+	-	FBWsl
SETTA	P	Activation d'un outil du groupe d'usure		+	-	FBWsl
SETTCOR	F	Modification de composantes d'outils avec prise en compte de l'ensemble des conditions marginales		+	-	PGAsl
SETTIA	P	Désactivation d'un outil du groupe d'usure		+	-	FBWsl
SF	A	Décalage du point de départ pour filetage	m	+		PGsl
SIN	F	Sinus (fonction trigonométrique)		+	+	PGAsl
SIRELAY	F	Activer les fonctions de sécurité paramétrées avec SIRELIN, SIRELOUT et SIRELTIME		-	+	FBSIsl
SIRELIN	P	Initialiser les grandeurs d'entrée du bloc fonctionnel		+	-	FBSIsl
SIRELOUT	P	Initialiser les grandeurs de sortie du bloc fonctionnel		+	-	FBSIsl
SIRELTIME	P	Initialiser les temporisations du bloc fonctionnel		+	-	FBSIsl
SLOT1	C (T)	Rainure rectiligne		+		PGAsl
SLOT2	C (T)	Rainure circulaire		+		PGAsl
SOFT	G	Accélération sur trajectoire avec limitation des à-coups	m	+		PGAsl
SOFTA	P	Activation de l'accélération d'axe avec limitation des à-coups pour les axes programmés		+	-	PGAsl
SON	G	Grignotage ACTIVÉ	m	+		PGAsl
SONS	G	Grignotage activé à la période d'appel de l'interpolateur	m	+		PGAsl

Instruction	Type [1]	Signification	E [2]	PP [3]	AS [4]	Description, voir [5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
SPATH [6]	G	La référence trajectoire pour axes FGROUP est la longueur d'arc	m	+		PGAsl
SPCOF	P	Commutation la broche maître ou broche avec numéro n de l'asservissement de position sur la régulation de la vitesse de rotation	m	+	-	PGsl
SPCON	P	Commutation la broche maître ou broche avec numéro n de la régulation de la vitesse de rotation sur l'asservissement de position	m	+	-	PGAsl
SPI	F	Conversion du numéro de broche en descripteur d'axe		+	-	PGAsl
SPIF1 [6]	G	E/S CN rapides pour poinçonnage/grignotage, octet 1	m	+		FB2sl (N4)
SPIF2	G	E/S CN rapides pour poinçonnage/grignotage, octet 2	m	+		FB2sl (N4)
SPLINEPATH	P	Détermination du groupe Spline		+	-	PGAsl
SPN	A	Nombre de distances partielles par bloc	b	+		PGAsl
SPOF [6]	G	Coup désactivé, poinçonnage, grignotage désactivés	m	+		PGAsl
SPOS	K	Position de broche	m	+	+	PGsl
SPOSA	K	Position de broche au-delà des limites de bloc	m	+		PGsl
SPP	A	Longueur d'une distance partielle	m	+		PGAsl
SPRINT	F	Retourne une chaîne de caractères d'entrée formatée		+		PGAsl
SQRT	F	Racine carrée (fonction arithmétique) (square root)		+	+	PGAsl
SR	A	Distance de retrait de l'oscillation pour une action synchrone	b	+		PGsl
SRA	K	Distance de retrait axiale de l'oscillation suite à une action externe pour une action synchrone	m	+		PGsl
ST	A	Durée des passes à lécher oscillatoires pour une action synchrone	b	+		PGsl
STA	K	Durée des passes à lécher oscillatoires axiales pour une action synchrone	m	+		PGsl
START	P	Lancement des programmes sélectionnés simultanément dans plusieurs canaux, à partir du programme en cours		+	-	PGAsl
STARTFIFO [6]	G	Exécution ; remplissage en parallèle du tampon d'exécution	m	+		PGAsl
STAT		Position des articulations	b	+		PGAsl
STOLF	K	Facteur de tolérance G0	m	+		PGAsl
STOPFIFO	G	Arrêt de l'exécution remplissage du tampon d'exécution jusqu'à la reconnaissance de STARTFIFO, la saturation du tampon d'exécution ou la fin du programme	m	+		PGAsl

Instruction	Type [1]	Signification	E [2]	PP [3]	AS [4]	Description, voir [5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
STOPRE	P	Arrêt du prétraitement des blocs jusqu'à ce que tous les blocs préparés aient été exécutés		+	-	PGAsl
STOPREOF	P	Annulation de l'arrêt du prétraitement des blocs		-	+	FBSYsl
STRING	K	Type de données : chaîne de caractères		+		PGAsl
STRINGIS	F	Contrôle le langage CN existant et si les noms de cycles CN, les variables utilisateurs, les macros et les noms d'étiquettes appartenant spécifiquement à cette instruction existent, sont valables, définis ou encore actifs.		+	-	PGAsl
STRLEN	F	Déterminer la longueur d'une chaîne de caractères		+	-	PGAsl
SUBSTR	F	Détermination de l'indice d'un caractère dans une chaîne de caractères		+	-	PGAsl
SUPA	G	Inhibition du décalage d'origine actuel, y compris les décalages programmés, les frames système, les décalages par manivelle (DRF), le décalage externe d'origine et le déplacement forcé	b	+		PGsl
SVC	K	Vitesse de coupe d'outil	m	+		PGsl
SYNFCT	P	Évaluation d'un polynôme en fonction d'une condition dans une action synchrone au déplacement		-	+	FBSYsl
SYNR	K	Lecture synchrone de la variable, c'est-à-dire au moment de l'exécution		+		PGAsl
SYNRW	K	Lecture et écriture synchrones de la variable, c'est-à-dire au moment de l'exécution		+		PGAsl
SYNW	K	Écriture synchrone de la variable, c'est-à-dire au moment de l'exécution		+		PGAsl
T	A	Appel d'outil (changement uniquement s'il est fixé dans les paramètres machine ; sinon instruction M6 nécessaire)		+		PGsl
TAN	F	Tangente (fonction trigonométrique)		+	+	PGAsl
TANG	P	Positionnement tangentiel : définition du couplage		+	-	PGAsl
TANGDEL	P	Positionnement tangentiel : suppression du couplage		+	-	PGAsl
TANGOF	P	Positionnement tangentiel : désactivation du couplage		+	-	PGAsl
TANGON	P	Positionnement tangentiel : activation du couplage		+	-	PGAsl
TCA (828D: _TCA)	P	Sélection /changement d'outil indépendant de l'état de l'outil		+	-	FBWsl
TCARR	A	Appeler un organe porte-outil (numéro "m")		+		PGAsl
TCI	P	Mettre en place l'outil se trouvant à l'emplacement tampon dans le magasin		+	-	FBWsl

Instruction	Type [1]	Signification	E [2]	PP [3]	AS [4]	Description, voir [5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
TCOABS [6]	G	Détermination des composantes de longueur d'outil à partir de l'orientation actuelle de l'outil	m	+		PGAsl
TCOFR	G	Détermination des composantes de longueur d'outil à partir de l'orientation du frame actif.	m	+		PGAsl
TCOFRX	G	Lors de la sélection d'outil, détermination de l'orientation de l'outil d'un frame activé, outil pointé en direction X	m	+		PGAsl
TCOFRY	G	Lors de la sélection d'outil, détermination de l'orientation de l'outil d'un frame activé, outil pointé en direction Y	m	+		PGAsl
TCOFRZ	G	Lors de la sélection d'outil, détermination de l'orientation de l'outil d'un frame activé, outil pointé en direction Z	m	+		PGAsl
THETA	A	Angle de rotation	b	+		PGAsl
TILT	A	Angle latéral	m	+		PGAsl
TLIFT	P	Positionnement tangentiel : activation de la création de blocs intermédiaires		+	-	PGAsl
TML	P	Sélection d'un outil à l'aide du numéro d'emplacement de magasin		+	-	FBWsl
TMOF	P	Désactivation de la surveillance d'outil		+	-	PGAsl
TMON	P	Activation de la surveillance d'outil		+	-	PGAsl
TO	K	Désigne la valeur finale d'une boucle de comptage FOR		+		PGAsl
TOFF	K	Offset de longueur d'outil dans la direction de la composante de longueur d'outil agissant parallèlement à l'axe géométrique indiqué dans l'indice	m	+		PGsl
TOFFL	K	Offset de longueur d'outil dans la direction de la composante de longueur d'outil L1, L2 ou L3	m	+		PGsl
TOFFOF	P	Initialisation de la correction de la longueur d'outil en ligne		+	-	PGAsl
TOFFON	P	Activation de la correction en ligne de la longueur d'outil		+	-	PGAsl
TOFFR	A	Offset de rayon d'outil	m	+		PGsl
TOFRAME	G	Orienter l'axe Z du SCP parallèlement à l'orientation de l'outil par rotation du frame	m	+		PGsl
TOFRAMEX	G	Orienter l'axe X du SCP parallèlement à l'orientation de l'outil par rotation du frame	m	+		PGsl
TOFRAMEY	G	Orienter l'axe Y du SCP parallèlement à l'orientation de l'outil par rotation du frame	m	+		PGsl
TOFRAMEZ	G	Identique à TOFRAME	m	+		PGsl
TOLOWER	F	Transformer les caractères d'une chaîne de caractères en minuscules		+	-	PGAsl

Instruction	Type [1]	Signification	E [2]	PP [3]	AS [4]	Description, voir [5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
TOOLENV	F	Enregistrer tous les états actuels significatifs pour l'exploitation des données d'outil mémorisées dans la mémoire		+	-	PGAsl
TOOLGNT	F	Détermination du nombre d'outils d'un groupe d'outils		+	-	FBWsl
TOOLGT	F	Détermination du numéro T d'un outil issu d'un groupe d'outils		+	-	FBWsl
TOROT	G	Orienter l'axe Z du SCP parallèlement à l'orientation de l'outil par rotation du frame	m	+		PGsl
TOROTOF [6]	G	Rotations de frames en direction de l'outil désactivées	m	+		PGsl
TOROTX	G	Orienter l'axe X du SCP parallèlement à l'orientation de l'outil par rotation du frame	m	+		PGsl
TOROTY	G	Orienter l'axe Y du SCP parallèlement à l'orientation de l'outil par rotation du frame	m	+		PGsl
TOROTZ	G	Identique à TOROT	m	+		PGsl
TOUPPER	F	Transformer les caractères d'une chaîne de caractères en majuscules		+	-	PGAsl
TOWBCS	G	Valeurs d'usure dans le système de coordonnées de base (SCB)	m	+		PGAsl
TOWKCS	G	Valeurs d'usure dans le système de coordonnées de la tête d'outil avec transformation cinétique (diffère du SCM avec rotation d'outil)	m	+		PGAsl
TOWMCS	G	Valeurs d'usure dans le système de coordonnées machine (SCM)	m	+		PGAsl
TOWSTD [6]	G	Valeur de position de base pour corrections sur la longueur d'outil	m	+		PGAsl
TOWTCS	G	Valeurs d'usure dans le système de coordonnées outil (point de référence du porte-outil T sur le mandrin du porte-outil)	m	+		PGAsl
TOWWCS	G	Valeurs d'usure dans le système de coordonnées pièce (SCP)	m	+		PGAsl
TR	K	Composante de décalage d'une variable de frame		+		PGAsl
TRAANG	P	Transformation axe incliné		+	-	PGAsl
TRACON	P	Transformation en cascade		+	-	PGAsl
TRACYL	P	Cylindre : transformation sur surfaces latérales		+	-	PGAsl
TRAFOOF	P	Désactiver les transformations actives dans le canal		+	-	PGAsl
TRAFOON	P	Activation de la transformation définie avec des chaînes cinématiques		+	-	PGAsl
TRAILOF	P	Déplacement conjugué synchrone à l'axe désactivé		+	+	PGAsl
TRAILON	P	Activation du déplacement conjugué synchrone à l'axe		+	+	PGAsl

Instruction	Type [1]	Signification	E [2]	PP [3]	AS [4]	Description, voir [5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
TRANS	G	Décalage d'origine absolu programmable	b	+		PGsl
TRANSMIT	P	Transformation polaire (usinage sur face frontale)		+	-	PGAsl
TRAORI	P	Transformation 4, 5 axes, transformation générique		+	-	PGAsl
TRUE	K	Constante logique : vrai		+		PGAsl
TRUNC	F	Troncature des décimales		+	+	PGAsl
TU		Angle d'axe	b	+		PGAsl
TURN	A	Nombre de tours pour une hélice	b	+		PGsl
ULI	K	Valeur limite supérieure de variables		+		PGAsl
UNLOCK	P	Libérer une action synchrone avec ID (poursuivre un cycle technologique)		-	+	FBSYsl
UNTIL	K	Condition de sortie d'une boucle REPEAT		+		PGAsl
UPATH	G	La référence trajectoire pour axes FGROUP est paramètre de courbe	m	+		PGAsl
VAR	K	Mot-clé : type du transfert de paramètres		+		PGAsl
VELOLIM	K	Réduction de la vitesse axiale maximale	m	+		PGAsl
VELOLIMA	K	Réduction ou surhaussement de la vitesse axiale maximale de l'axe asservi	m	+	+	PGAsl
WAITC	P	Attendre que le critère de changement de bloc de couplage soit rempli pour les axes/broches		+	-	PGAsl
WAITE	P	Attente de la fin du programme dans un autre canal.		+	-	PGAsl
WAITENC	P	Attente des positions d'axe synchronisées ou restaurées		+	-	PGAsl
WAITM	P	Attente de la marque dans canal indiqué ; terminer le bloc précédent par un arrêt précis.		+	-	PGAsl
WAITMC	P	Attente de la marque dans canal indiqué ; arrêt précis uniquement si les autres canaux n'ont pas encore atteint la marque.		+	-	PGAsl
WAITP	P	Attente de la fin du déplacement de l'axe de positionnement		+	-	PGsl
WAITS	P	Attente jusqu'à ce que la position de la broche soit atteinte		+	-	PGsl
WALCS0 [6]	G	Limitation de la zone de travail SCP désélectionnée	m	+	-	PGsl
WALCS1	G	Groupe de limitations de la zone de travail SCP 1 actif	m	+	-	PGsl
WALCS2	G	Groupe de limitations de la zone de travail SCP 2 actif	m	+	-	PGsl
WALCS3	G	Groupe de limitations de la zone de travail SCP 3 actif	m	+	-	PGsl
WALCS4	G	Groupe de limitations de la zone de travail SCP 4 actif	m	+	-	PGsl

Instruction	Type [1]	Signification	E [2]	PP [3]	AS [4]	Description, voir [5]
[1] [2] [3] [4] [5] Explications voir Légende (Page 910).						
WALCS5	G	Groupe de limitations de la zone de travail SCP 5 actif	m	+	-	PGsl
WALCS6	G	Groupe de limitations de la zone de travail SCP 6 actif	m	+	-	PGsl
WALCS7	G	Groupe de limitations de la zone de travail SCP 7 actif	m	+	-	PGsl
WALCS8	G	Groupe de limitations de la zone de travail SCP 8 actif	m	+	-	PGsl
WALCS9	G	Groupe de limitations de la zone de travail SCP 9 actif	m	+	-	PGsl
WALCS10	G	Groupe de limitations de la zone de travail SCP 10 actif	m	+	-	PGsl
WALIMOF	G	Désactivation de la limitation de la zone de travail SCB	m	+	-	PGsl
WALIMON [6]	G	Activation de la limitation de la zone de travail SCB	m	+	-	PGsl
WHEN	K	L'action est exécutée de manière unique lorsque la condition est remplie une fois.		-	+	FBSYsl
WHENEVER	K	L'action est exécutée de manière cyclique dans chaque cycle d'interpolation lorsque la condition est remplie.		-	+	FBSYsl
WHILE	K	Début d'une boucle WHILE		+		PGAsl
WRITE	P	Écrire un texte dans le système de fichiers. Ajoute un bloc à la fin du fichier indiqué.		+	-	PGAsl
WRTPR	P	Écrire la chaîne de caractères dans la variable OPI		+	-	PGsl
X	A	Nom d'axe	m/b	+	+	PGsl
XOR	O	OU exclusif logique		+		PGAsl
Y	A	Nom d'axe	m/b	+	+	PGsl
Z	A	Nom d'axe	m/b	+	+	PGsl

Légende

1) Type d'instruction :

A Adresse

Descripteur au quel une valeur a été attribuée (par ex. OVR=10). Certaines adresses activent ou désactivent également des fonctions sans attribution de valeur (par ex. CPLON et CPLOF).

C (A) Cycle AST

Programme CN défini par défaut pour la réoptimisation automatique avec AST(= Optimisation Servo automatique). L'adaptation à la situation d'optimisation concrète s'effectue à l'aide des paramètres transmis au cycle lors de l'appel.

C (M) Cycle de mesure

Programme CN prédéfini, contenant un processus de mesure, par ex. la détermination du diamètre intérieur d'une pièce cylindrique, pour être applicable de manière générale. L'adaptation à la situation de mesure concrète s'effectue à l'aide des paramètres transmis au cycle lors de l'appel.

C (T) Cycle technologique

Programme CN prédéfini dans lequel un certain cycle (procédé d'usinage), par ex. le taraudage ou le fraisage d'une poche, est programmé pour être applicable de manière générale. L'adaptation à la situation d'usinage concrète s'effectue à l'aide des paramètres transmis au cycle lors de l'appel.

F Fonction prédéfinie (fournit une valeur de retour)

L'appel de la fonction prédéfinie peut figurer sous forme d'opérande dans l'expression.

G Instruction G

Les instructions G sont réparties en groupes. Dans un bloc, on ne peut écrire qu'une seule instruction G d'un groupe. Une instruction G peut avoir un effet modal (elle s'applique jusqu'à ce qu'elle soit remplacée par une autre instruction du même groupe), ou un effet non modal, auquel cas elle s'applique uniquement dans le bloc dans lequel elle figure.

K Mot-clé

Descripteur qui détermine la syntaxe d'un bloc. Aucune valeur n'est attribuée à un mot clé et un mot clé ne permet pas d'activer ou de désactiver des fonctions CN.

Exemples : conditions (IF, ELSE, ENDIF, WHEN, ...), sauts (GOTOB, GOTO, RET …)

O Opérateur

Opérateur d'une opération mathématique, logique ou relationnelle

P Procédure prédéfinie (ne fournit pas de valeur de retour)

PA Attribut de programme

Les attributs de programme figurent à la fin de la ligne de définition d'un sous-programme :
`PROC <nom de programme>(...) <attribut de programme>`

Ils déterminent le comportement lors de l'exécution du sous-programme.

2) Prise d'effet de l'instruction :

m Modale

b Bloc par bloc (non modale)

3) Possibilité de programmation dans le programme pièce :

\+ Programmable

\- Non programmable

M Programmable uniquement par le constructeur de la machine

[4] Possibilité de programmation dans des actions synchrones :

+	Programmable
-	Non programmable
T	Programmable uniquement dans des cycles technologiques

[5] Référence au document contenant la description complète de l'instruction :

PGsl	Manuel de programmation Notions de base
PGAsl	Manuel de programmation Notions complémentaires
BNMsl	Manuel de programmation Cycles de mesure
BHDsl	Manuel d'utilisation Tournage
BHFsl	Manuel d'utilisation Fraisage
FB1sl ( )	Description fonctionnelle Fonctions de base (avec l'indication entre parenthèses de l'abréviation alphanumérique de la description fonctionnelle correspondante)
FB2sl ( )	Description fonctionnelle Fonctions d'extension (avec l'indication entre parenthèses de l'abréviation alphanumérique de la description fonctionnelle correspondante)
FB3sl ( )	Description fonctionnelle Fonctions spéciales (avec l'indication entre parenthèses de l'abréviation alphanumérique de la description fonctionnelle correspondante)
FBSIsl	Description fonctionnelle Safety Integrated
FBSYsl	Description fonctionnelle Actions synchrones
FBWsl	Description fonctionnelle Gestion d'outils

[6] Réglage par défaut en début du programme (dans la version de base de la commande, si elle n'a pas été programmée autrement).

Figure 21-1 Signification des notes de bas de page dans les tableaux d'instructions.

21.2 Instructions : Disponibilité pour SINUMERIK 828D

Remarque

Cycles

Les cycles sont qualifiés "en option" s'ils dépendent des options suivantes soumises à une licence :

- Fonctions technologiques étendues (numéro d'article : 6FC5800-0AP58-0YB0)
- Cycles de mesure (numéro d'article : 6FC5800-0AP28-0YB0)
- Mesure cinématique (numéro d'article : 6FC5800-0AP18-0YB0)
- SINUMERIK Rectification Advanced (numéro d'article : 6FC5800-0AS35-0YB0)

Les cycles n'acquérant par l'option "Fonctions technologiques étendues" que des fonctionnalités partielles sur certains systèmes ne sont pas spécialement identifiés.

21.2.1 Variantes de commande fraisage/tournage

Instructions A ... C

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
:	●	●	●	●	●	●	●	●
*	●	●	●	●	●	●	●	●
+	●	●	●	●	●	●	●	●
-	●	●	●	●	●	●	●	●
<	●	●	●	●	●	●	●	●
<<	●	●	●	●	●	●	●	●
<=	●	●	●	●	●	●	●	●
=	●	●	●	●	●	●	●	●
>=	●	●	●	●	●	●	●	●
/	●	●	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●	●	●
A	●	●	●	●	●	●	●	●
A2	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-
A4	-	-	-	-	-	-	-	-
A5	-	-	-	-	-	-	-	-

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
A6	-	-	-	-	-	-	-	-
A7	-	-	-	-	-	-	-	-
ABS	●	●	●	●	●	●	●	●
AC	●	●	●	●	●	●	●	●
ACC	●	●	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●	●	●
ACN	●	●	●	●	●	●	●	●
ACOS	●	●	●	●	●	●	●	●
ACP	●	●	●	●	●	●	●	●
ACTBLOCNO	●	●	●	●	●	●	●	●
ADDFRAME	●	●	●	●	●	●	●	●
ADIS	●	●	●	●	●	●	●	●
ADISPOS	●	●	●	●	●	●	●	●
ADISPOSA	●	●	●	●	●	●	●	●
ALF	●	●	●	●	●	●	●	●
AMIRROR	●	●	●	●	●	●	●	●
AND	●	●	●	●	●	●	●	●
ANG	●	●	●	●	●	●	●	●
AP	●	●	●	●	●	●	●	●
APR	●	●	●	●	●	●	●	●
APRB	●	●	●	●	●	●	●	●
APRP	●	●	●	●	●	●	●	●
APW	●	●	●	●	●	●	●	●
APWB	●	●	●	●	●	●	●	●
APWP	●	●	●	●	●	●	●	●
APX	●	●	●	●	●	●	●	●
AR	●	●	●	●	●	●	●	●
AROT	●	●	●	●	●	●	●	●
AROTS	●	●	●	●	●	●	●	●
AS	●	●	●	●	●	●	●	●
ASCALE	●	●	●	●	●	●	●	●
ASIN	●	●	●	●	●	●	●	●
ASPLINE	○	○	○	○	○	○	○	○
ATAN2	●	●	●	●	●	●	●	●
ATOL	●	●	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
AUXFUDELG	●	●	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●	●	●
AX	●	●	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-	-	-
AXIS	●	●	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●	●	●
B	●	●	●	●	●	●	●	●
B2	-	-	-	-	-	-	-	-
B3	-	-	-	-	-	-	-	-
B4	-	-	-	-	-	-	-	-
B5	-	-	-	-	-	-	-	-
B6	-	-	-	-	-	-	-	-
B7	-	-	-	-	-	-	-	-
B_AND	●	●	●	●	●	●	●	●
B_OR	●	●	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○	○	○
BLOC	●	●	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●	●	●
BNAT	○	○	○	○	○	○	○	○
BOOL	●	●	●	●	●	●	●	●
BOUND	●	●	●	●	●	●	●	●
BRISK	●	●	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○	○	○
BTAN	○	○	○	○	○	○	○	○
C	●	●	●	●	●	●	●	●
C2	-	-	-	-	-	-	Nom d'axe de canal	Nom d'axe de canal

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
C3	-	-	-	-	-	-	-	-
C4	-	-	-	-	-	-	-	-
C5	-	-	-	-	-	-	-	-
C6	-	-	-	-	-	-	-	-
C7	-	-	-	-	-	-	-	-
CAC	●	●	●	●	●	●	●	●
CACN	●	●	●	●	●	●	●	●
CACP	●	●	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●	●	●
CALCPOSI	●	●	●	●	●	●	●	●
CALL	●	●	●	●	●	●	●	●
CALLPATH	●	●	●	●	●	●	●	●
CANCEL	●	●	●	●	●	●	●	●
CASE	●	●	●	●	●	●	●	●
CDC	●	●	●	●	●	●	●	●
CDOF	-	-	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-	-	-
CDON	-	-	-	-	-	-	-	-
CFC	●	●	●	●	●	●	●	●
CFIN	●	●	●	●	●	●	●	●
CFINE	●	●	●	●	●	●	●	●
CFTCP	●	●	●	●	●	●	●	●
CHAN	●	●	●	●	●	●	●	●
CHANDATA	●	●	●	●	●	●	●	●
CHAR	●	●	●	●	●	●	●	●
CHF	●	●	●	●	●	●	●	●
CHKDM	●	●	●	●	●	●	●	●
CHKDNO	●	●	●	●	●	●	●	●
CHR	●	●	●	●	●	●	●	●
CIC	●	●	●	●	●	●	●	●
CIP	●	●	●	●	●	●	●	●
CLEARM	-	-	-	-	-	○	-	●
CLRINT	●	●	●	●	●	●	●	●
CMIRROR	●	●	●	●	●	●	●	●
COARSEA	●	●	●	●	●	●	●	●
COLLPAIR	-	-	-	-	-	-	-	-
COMPCAD	●	-	●	-	●	●	-	-

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
COMPCURV	●	-	●	-	●	●	-	-
COMPLETE	●	●	●	●	●	●	●	●
COMPOF	●	-	●	-	●	●	-	-
COMPON	●	-	●	-	●	●	-	-
COMPSURF	-	-	○	-	○	○	-	-
CONTDCON	●	●	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●	●	●
CORROF	●	●	●	●	●	●	●	●
CORRTRAFO	-	-	-	-	-	-	-	-
COS	●	●	●	●	●	●	●	●
COUPDEF	-	○	-	○	-	-	○	○
COUPDEL	-	○	-	○	-	-	○	○
COUPOF	-	○	-	○	-	-	○	○
COUPOFS	-	○	-	○	-	-	○	○
COUPON	-	○	-	○	-	-	○	○
COUPONC	-	○	-	○	-	-	○	○
COUPRES	-	○	-	○	-	-	○	○
CP	●	●	●	●	●	●	●	●
CPBC	○	○	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○	○	○
CPLA	○	○	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○	○	○
CPLON	○	○	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○	○	○

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
CPLOUTTR	○	○	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○	○	○
CPOF	○	○	○	○	○	○	○	○
CPON	○	○	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●	●	●
CPRES	○	○	○	○	○	○	○	○
CPROT	●	●	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○	○	○
CR	●	●	●	●	●	●	●	●
CROT	●	●	●	●	●	●	●	●
CROTS	●	●	●	●	●	●	●	●
CRPL	●	●	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○	○	○
CT	●	●	●	●	●	●	●	●
CTAB	-	-	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-	-	-

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
CTABFSEG	-	-	-	-	-	-	-	-
CTABID	-	-	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-	-	-
CTABPOLID	-	-	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-	-	-
CTOL	●	●	●	●	●	●	●	●
CTRANS	●	●	●	●	●	●	●	●
CUT2D	●	●	●	●	●	●	●	●
CUT2DD	●	●	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●	●	●
CUT2DFD	●	●	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-	-	-

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
CUT3DFS	-	-	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-	-	-
CYCLE60	○	○	●	●	●	●	●	●
CYCLE61	●	●	●	●	●	●	●	●
CYCLE62	●	●	●	●	●	●	●	●
CYCLE63	○	○	●	●	●	●	●	●
CYCLE64	○	○	●	●	●	●	●	●
CYCLE70	○	○	●	●	●	●	●	●
CYCLE72	●	●	●	●	●	●	●	●
CYCLE76	●	●	●	●	●	●	●	●
CYCLE77	●	●	●	●	●	●	●	●
CYCLE78	○	○	●	●	●	●	●	●
CYCLE79	○	○	●	●	●	●	●	●
CYCLE81	●	●	●	●	●	●	●	●
CYCLE82	●	●	●	●	●	●	●	●
CYCLE83	●	●	●	●	●	●	●	●
CYCLE84	●	●	●	●	●	●	●	●
CYCLE85	●	●	●	●	●	●	●	●
CYCLE86	●	●	●	●	●	●	●	●
CYCLE92	●	●	●	●	●	●	●	●
CYCLE95	●	●	●	●	●	●	●	●
CYCLE98	●	●	●	●	●	●	●	●
CYCLE99	●	●	●	●	●	●	●	●
CYCLE150	○	○	○	○	○	○	○	○
CYCLE435	-	-	-	-	-	-	-	-
CYCLE495	-	-	-	-	-	-	-	-
CYCLE750	●	●	●	●	●	●	●	●
CYCLE751	●	●	●	●	●	●	●	●
CYCLE752	●	●	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
CYCLE758	●	●	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●	●	●
CYCLE800	-	-	●	●	●	●	●	●
CYCLE801	●	●	●	●	●	●	●	●
CYCLE802	●	●	●	●	●	●	●	●
CYCLE830	○	○	●	●	●	●	●	●
CYCLE832	●	●	●	●	●	●	●	●
CYCLE840	●	●	●	●	●	●	●	●
CYCLE899	○	○	●	●	●	●	●	●
CYCLE930	●	●	●	●	●	●	●	●
CYCLE940	●	●	●	●	●	●	●	●
CYCLE951	●	●	●	●	●	●	●	●
CYCLE952	○	○	●	●	●	●	●	●
CYCLE961	○	○	○	○	○	○	○	○
CYCLE971	○	○	○	○	○	○	○	○
CYCLE973	○	○	○	○	○	○	○	○
CYCLE974	○	○	○	○	○	○	○	○
CYCLE976	○	○	○	○	○	○	○	○
CYCLE977	○	○	○	○	○	○	○	○
CYCLE978	○	○	○	○	○	○	○	○
CYCLE979	○	○	○	○	○	○	○	○
CYCLE982	○	○	○	○	○	○	○	○
CYCLE994	○	○	○	○	○	○	○	○
CYCLE995	○	○	○	○	○	○	○	○
CYCLE996	○	-	○	-	○	○	-	-
CYCLE997	○	○	○	○	○	○	○	○
CYCLE998	○	○	○	○	○	○	○	○
CYCLE4071	-	-	-	-	-	-	-	-
CYCLE4072	-	-	-	-	-	-	-	-
CYCLE4073	-	-	-	-	-	-	-	-
CYCLE4074	-	-	-	-	-	-	-	-
CYCLE4075	-	-	-	-	-	-	-	-
CYCLE4077	-	-	-	-	-	-	-	-
CYCLE4078	-	-	-	-	-	-	-	-
CYCLE4079	-	-	-	-	-	-	-	-

Instructions D ... F

Instruction	SINUMERIK 828D							
• Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exporta-tion (me42)	SW24x(5) Logiciel CN Tournage Exporta-tion (te42)	SW26x(3) Logiciel CN Fraisage Exporta-tion (me62)	SW26x(3) Logiciel CN Tournage Exporta-tion (te62)	SW28x(2) Logiciel CN Fraisage Exporta-tion (me821)	SW28x(1) Logiciel CN Fraisage Adv. Expor-tation (me822)	SW28x(2) Logiciel CN Tournage Exporta-tion (te821)	SW28x(1) Logiciel CN Tournage Adv. Expor-tation (te822)
D	•	•	•	•	•	•	•	•
D0	•	•	•	•	•	•	•	•
DAC	•	•	•	•	•	•	•	•
DC	•	•	•	•	•	•	•	•
DCI	•	•	•	•	•	•	•	•
DCM	•	•	•	•	•	•	•	•
DCU	•	•	•	•	•	•	•	•
DEF	•	•	•	•	•	•	•	•
DEFINE	•	•	•	•	•	•	•	•
DEFAULT	•	•	•	•	•	•	•	•
DELAYFSTON	•	•	•	•	•	•	•	•
DELAYFSTOF	•	•	•	•	•	•	•	•
DELDL	•	•	•	•	•	•	•	•
DELDTG	•	•	•	•	•	•	•	•
DELETE	•	•	•	•	•	•	•	•
DELMLOWNER	•	•	•	•	•	•	•	•
DELMLRES	•	•	•	•	•	•	•	•
DELMT	•	•	•	•	•	•	•	•
DELOBJ	-	-	-	-	-	-	-	-
DELT	•	•	•	•	•	•	•	•
DELTC	•	•	•	•	•	•	•	•
DELTOOLENV	•	•	•	•	•	•	•	•
DIACYCOFA	•	•	•	•	•	•	•	•
DIAM90	•	•	•	•	•	•	•	•
DIAM90A	•	•	•	•	•	•	•	•
DIAMCHAN	•	•	•	•	•	•	•	•
DIAMCHANA	•	•	•	•	•	•	•	•
DIAMCYCOF	•	•	•	•	•	•	•	•
DIAMOF	•	•	•	•	•	•	•	•
DIAMOFA	•	•	•	•	•	•	•	•
DIAMON	•	•	•	•	•	•	•	•
DIAMONA	•	•	•	•	•	•	•	•
DIC	•	•	•	•	•	•	•	•
DILF	•	•	•	•	•	•	•	•
DISABLE	•	•	•	•	•	•	•	•

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
DISC	●	●	●	●	●	●	●	●
DISCL	●	●	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●	●	●
DISPR	●	●	●	●	●	●	●	●
DISR	●	●	●	●	●	●	●	●
DISRP	●	●	●	●	●	●	●	●
DITE	●	●	●	●	●	●	●	●
DITS	●	●	●	●	●	●	●	●
DIV	●	●	●	●	●	●	●	●
DL	-	-	-	-	-	-	-	-
DO	●	●	●	●	●	●	●	●
DRFOF	●	●	●	●	●	●	●	●
DRIVE	●	●	●	●	●	●	●	●
DRIVEA	●	●	●	●	●	●	●	●
DYNFINISH	●	●	●	●	●	●	●	●
DYNNORM	●	●	●	●	●	●	●	●
DYNPOS	●	●	●	●	●	●	●	●
DYNROUGH	●	●	●	●	●	●	●	●
DYNSEMIFIN	●	●	●	●	●	●	●	●
DZERO	●	●	●	●	●	●	●	●
EAUTO	○	○	○	○	○	○	○	○
EGDEF	-	○	-	○	-	-	○	○
EGDEL	-	○	-	○	-	-	○	○
EGOFC	-	○	-	○	-	-	○	○
EGOFS	-	○	-	○	-	-	○	○
EGON	-	○	-	○	-	-	○	○
EGONSYN	-	○	-	○	-	-	○	○
EGONSYNE	-	○	-	○	-	-	○	○
ELSE	●	●	●	●	●	●	●	●
ENABLE	●	●	●	●	●	●	●	●
ENAT	○	○	○	○	○	○	○	○
ENDFOR	●	●	●	●	●	●	●	●
ENDIF	●	●	●	●	●	●	●	●
ENDLABEL	●	●	●	●	●	●	●	●
ENDLOOP	●	●	●	●	●	●	●	●
ENDPROC	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
ENDWHILE	●	●	●	●	●	●	●	●
ESRR	○	○	○	○	○	○	○	○
ESRS	○	○	○	○	○	○	○	○
ETAN	○	○	○	○	○	○	○	○
EVERY	●	●	●	●	●	●	●	●
EX	●	●	●	●	●	●	●	●
EXECSTRING	●	●	●	●	●	●	●	●
EXECTAB	●	●	●	●	●	●	●	●
EXECUTE	●	●	●	●	●	●	●	●
EXP	●	●	●	●	●	●	●	●
EXTCALL	●	●	●	●	●	●	●	●
EXTCLOSE	●	●	●	●	●	●	●	●
EXTERN	●	●	●	●	●	●	●	●
EXTOPEN	●	●	●	●	●	●	●	●
F	●	●	●	●	●	●	●	●
FA	●	●	●	●	●	●	●	●
FAD	●	●	●	●	●	●	●	●
FALSE	●	●	●	●	●	●	●	●
FB	●	●	●	●	●	●	●	●
FCTDEF	●	●	●	●	●	●	●	●
FCUB	●	●	●	●	●	●	●	●
FD	●	●	●	●	●	●	●	●
FDA	●	●	●	●	●	●	●	●
FENDNORM	●	●	●	●	●	●	●	●
FFWOF	●	●	●	●	●	●	●	●
FFWON	●	●	●	●	●	●	●	●
FGREF	●	●	●	●	●	●	●	●
FGROUP	●	●	●	●	●	●	●	●
FI	●	●	●	●	●	●	●	●
FIFOCTRL	●	●	●	●	●	●	●	●
FILEDATE	●	●	●	●	●	●	●	●
FILEINFO	●	●	●	●	●	●	●	●
FILESIZE	●	●	●	●	●	●	●	●
FILESTAT	●	●	●	●	●	●	●	●
FILETIME	●	●	●	●	●	●	●	●
FINEA	●	●	●	●	●	●	●	●
FL	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
• Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
FLIN	•	•	•	•	•	•	•	•
FMA	•	•	•	•	•	•	•	•
FNORM	•	•	•	•	•	•	•	•
FOCOF	○	○	○	○	○	○	○	○
FOCON	○	○	○	○	○	○	○	○
FOR	•	•	•	•	•	•	•	•
FP	•	•	•	•	•	•	•	•
FPO	-	-	-	-	-	-	-	-
FPR	•	•	•	•	•	•	•	•
FPRAOF	•	•	•	•	•	•	•	•
FPRAON	•	•	•	•	•	•	•	•
FRAME	•	•	•	•	•	•	•	•
FRC	•	•	•	•	•	•	•	•
FRCM	•	•	•	•	•	•	•	•
FROM	•	•	•	•	•	•	•	•
FTOC	•	•	•	•	•	•	•	•
FTOCOF	•	•	•	•	•	•	•	•
FTOCON	•	•	•	•	•	•	•	•
FXS	•	•	•	•	•	•	•	•
FXST	•	•	•	•	•	•	•	•
FXSW	•	•	•	•	•	•	•	•
FZ	•	•	•	•	•	•	•	•

Instructions G ... L

Instruction	SINUMERIK 828D							
• Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
G0	•	•	•	•	•	•	•	•
G1	•	•	•	•	•	•	•	•
G2	•	•	•	•	•	•	•	•
G3	•	•	•	•	•	•	•	•
G4	•	•	•	•	•	•	•	•

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
G5	●	●	●	●	●	●	●	●
G7	●	●	●	●	●	●	●	●
G9	●	●	●	●	●	●	●	●
G17	●	●	●	●	●	●	●	●
G18	●	●	●	●	●	●	●	●
G19	●	●	●	●	●	●	●	●
G25	●	●	●	●	●	●	●	●
G26	●	●	●	●	●	●	●	●
G33	●	●	●	●	●	●	●	●
G34	●	●	●	●	●	●	●	●
G35	●	●	●	●	●	●	●	●
G40	●	●	●	●	●	●	●	●
G41	●	●	●	●	●	●	●	●
G42	●	●	●	●	●	●	●	●
G53	●	●	●	●	●	●	●	●
G54	●	●	●	●	●	●	●	●
G55	●	●	●	●	●	●	●	●
G56	●	●	●	●	●	●	●	●
G57	●	●	●	●	●	●	●	●
G58	→ G505							
G59	→ G506							
G60	●	●	●	●	●	●	●	●
G62	●	●	●	●	●	●	●	●
G63	●	●	●	●	●	●	●	●
G64	●	●	●	●	●	●	●	●
G70	●	●	●	●	●	●	●	●
G71	●	●	●	●	●	●	●	●
G74	●	●	●	●	●	●	●	●
G75	●	●	●	●	●	●	●	●
G90	●	●	●	●	●	●	●	●
G91	●	●	●	●	●	●	●	●
G93	●	●	●	●	●	●	●	●
G94	●	●	●	●	●	●	●	●
G95	●	●	●	●	●	●	●	●
G96	●	●	●	●	●	●	●	●
G97	●	●	●	●	●	●	●	●
G110	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
G111	●	●	●	●	●	●	●	●
G112	●	●	●	●	●	●	●	●
G140	●	●	●	●	●	●	●	●
G141	●	●	●	●	●	●	●	●
G142	●	●	●	●	●	●	●	●
G143	●	●	●	●	●	●	●	●
G147	●	●	●	●	●	●	●	●
G148	●	●	●	●	●	●	●	●
G153	●	●	●	●	●	●	●	●
G247	●	●	●	●	●	●	●	●
G248	●	●	●	●	●	●	●	●
G290	●	●	●	●	●	●	●	●
G291	●	●	●	●	●	●	●	●
G331	●	●	●	●	●	●	●	●
G332	●	●	●	●	●	●	●	●
G335	●	●	●	●	●	●	●	●
G336	●	●	●	●	●	●	●	●
G340	●	●	●	●	●	●	●	●
G341	●	●	●	●	●	●	●	●
G347	●	●	●	●	●	●	●	●
G348	●	●	●	●	●	●	●	●
G450	●	●	●	●	●	●	●	●
G451	●	●	●	●	●	●	●	●
G460	●	●	●	●	●	●	●	●
G461	●	●	●	●	●	●	●	●
G462	●	●	●	●	●	●	●	●
G500	●	●	●	●	●	●	●	●
G505 ... G599	●	●	●	●	●	●	●	●
G601	●	●	●	●	●	●	●	●
G602	●	●	●	●	●	●	●	●
G603	●	●	●	●	●	●	●	●
G621	●	●	●	●	●	●	●	●
G641	●	●	●	●	●	●	●	●
G642	●	●	●	●	●	●	●	●
G643	●	●	●	●	●	●	●	●
G644	●	●	●	●	●	●	●	●
G645	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
G700	●	●	●	●	●	●	●	●
G710	●	●	●	●	●	●	●	●
G810 ... G819	-	-	-	-	-	-	-	-
G820 ... G829	-	-	-	-	-	-	-	-
G931	●	●	●	●	●	●	●	●
G942	●	●	●	●	●	●	●	●
G952	●	●	●	●	●	●	●	●
G961	●	●	●	●	●	●	●	●
G962	●	●	●	●	●	●	●	●
G971	●	●	●	●	●	●	●	●
G972	●	●	●	●	●	●	●	●
G973	●	●	●	●	●	●	●	●
GEOAX	●	●	●	●	●	●	●	●
GET	●	●	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●	●	●
GETD	-	-	-	-	-	○	-	●
GETDNO	●	●	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●	●	●
GETSELT	●	●	●	●	●	●	●	●
GETT	●	●	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●	●	●
GETVARAP	●	●	●	●	●	●	●	●
GETVARDFT	●	●	●	●	●	●	●	●
GETVARLIM	●	●	●	●	●	●	●	●
GETVARPHU	●	●	●	●	●	●	●	●
GETVARTYP	●	●	●	●	●	●	●	●
GFRAME0 ... GFRAME100	-	-	-	-	-	-	-	-
GOTO	●	●	●	●	●	●	●	●
GOTOB	●	●	●	●	●	●	●	●
GOTOC	●	●	●	●	●	●	●	●
GOTOF	●	●	●	●	●	●	●	●
GOTOS	●	●	●	●	●	●	●	●
GP	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
GWPSOF	●	●	●	●	●	●	●	●
GROUP_ ADDEND	●	●	●	●	●	●	●	●
GROUP_BEGIN	●	●	●	●	●	●	●	●
GROUP_END	●	●	●	●	●	●	●	●
GWPSON	●	●	●	●	●	●	●	●
H...	●	●	●	●	●	●	●	●
HOLES1	●	●	●	●	●	●	●	●
HOLES2	●	●	●	●	●	●	●	●
I	●	●	●	●	●	●	●	●
I1	●	●	●	●	●	●	●	●
IC	●	●	●	●	●	●	●	●
ICYCOF	●	●	●	●	●	●	●	●
ICYCON	●	●	●	●	●	●	●	●
ID	●	●	●	●	●	●	●	●
IDS	●	●	●	●	●	●	●	●
IF	●	●	●	●	●	●	●	●
INDEX	●	●	●	●	●	●	●	●
INIPO	●	●	●	●	●	●	●	●
INIRE	●	●	●	●	●	●	●	●
INICF	●	●	●	●	●	●	●	●
INIT	-	-	-	-	-	○	-	●
INITIAL								
INT	●	●	●	●	●	●	●	●
INTERSEC	●	●	●	●	●	●	●	●
INVCCW	-	-	-	-	-	-	-	-
INVCW	-	-	-	-	-	-	-	-
INVFRAME	●	●	●	●	●	●	●	●
IP	●	●	●	●	●	●	●	●
IPOBRKA	●	●	●	●	●	●	●	●
IPOENDA	●	●	●	●	●	●	●	●
IPTRLOCK	●	●	●	●	●	●	●	●
IPTRUNLOCK	●	●	●	●	●	●	●	●
IR	●	●	●	●	●	●	●	●
ISAXIS	●	●	●	●	●	●	●	●
ISD	-	-	-	-	-	-	-	-
ISFILE	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
ISNUMBER	●	●	●	●	●	●	●	●
ISOCALL	●	●	●	●	●	●	●	●
ISVAR	●	●	●	●	●	●	●	●
J	●	●	●	●	●	●	●	●
J1	●	●	●	●	●	●	●	●
JERKA	●	●	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●	●	●
JR	●	●	●	●	●	●	●	●
K	●	●	●	●	●	●	●	●
K1	●	●	●	●	●	●	●	●
KONT	●	●	●	●	●	●	●	●
KONTC	●	●	●	●	●	●	●	●
KONTT	●	●	●	●	●	●	●	●
KR	●	●	●	●	●	●	●	●
L	●	●	●	●	●	●	●	●
LEAD Orientation initiale de l'outil Polyn. d'orientation	- -	- -	- -	- -	- -	- -	- -	- -
LEADOF	-	-	-	●	-	-	●	●
LEADON	-	-	-	●	-	-	●	●
LENTOAX	●	●	●	●	●	●	●	●
LFOF	●	●	●	●	●	●	●	●
LFON	●	●	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●	●	●
LFWP	●	●	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●	●	●
LIMS	●	●	●	●	●	●	●	●
LLI	●	●	●	●	●	●	●	●
LN	●	●	●	●	●	●	●	●
LOCK	●	●	●	●	●	●	●	●
LONGHOLE	●	●	●	●	●	●	●	●
LOOP	●	●	●	●	●	●	●	●

Instructions M ... R

Instruction	SINUMERIK 828D							
• Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
M0	•	•	•	•	•	•	•	•
M1	•	•	•	•	•	•	•	•
M2	•	•	•	•	•	•	•	•
M3	•	•	•	•	•	•	•	•
M4	•	•	•	•	•	•	•	•
M5	•	•	•	•	•	•	•	•
M6	•	•	•	•	•	•	•	•
M17	•	•	•	•	•	•	•	•
M19	•	•	•	•	•	•	•	•
M30	•	•	•	•	•	•	•	•
M40	•	•	•	•	•	•	•	•
M41 ... M45	•	•	•	•	•	•	•	•
M70	•	•	•	•	•	•	•	•
MASLDEF	-	-	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-	-	-
MASLON	-	-	-	-	-	-	-	-
MATCH	•	•	•	•	•	•	•	•
MAXVAL	•	•	•	•	•	•	•	•
MCALL	•	•	•	•	•	•	•	•
MEAC	-	-	-	-	-	-	-	-
MEAFRAME	•	•	•	•	•	•	•	•
MEAS	•	•	•	•	•	•	•	•
MEASA	-	-	-	-	-	-	-	-
MEASURE	•	•	•	•	•	•	•	•
MEAW	•	•	•	•	•	•	•	•
MEAWA	-	-	-	-	-	-	-	-
MI	•	•	•	•	•	•	•	•
MINDEX	•	•	•	•	•	•	•	•
MINVAL	•	•	•	•	•	•	•	•
MIRROR	•	•	•	•	•	•	•	•
MMC	•	•	•	•	•	•	•	•
MOD	•	•	•	•	•	•	•	•
MODAXVAL	•	•	•	•	•	•	•	•

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
MOV	●	●	●	●	●	●	●	●
MOVT	●	●	●	●	●	●	●	●
MSG	●	●	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●	●	●
N	●	●	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●	●	●
NC	●	●	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●	●	●
NEWT	●	●	●	●	●	●	●	●
NORM	●	●	●	●	●	●	●	●
NOT	●	●	●	●	●	●	●	●
NPROT	●	●	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-	-	-
OF	●	●	●	●	●	●	●	●
OFFN	●	●	●	●	●	●	●	●
OMA1	-	-	-	-	-	-	-	-
OMA2	-	-	-	-	-	-	-	-
OMA3	-	-	-	-	-	-	-	-
OMA4	-	-	-	-	-	-	-	-
OMA5	-	-	-	-	-	-	-	-
OR	●	●	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-	-	-
ORIC	-	-	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-	-	-
ORID	-	-	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-	-	-

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
ORIPATHS	-	-	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-	-	-
ORIS	-	-	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-	-	-
ORISON	-	-	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-	-	-
OS	-	-	-	-	-	-	-	-
OSB	-	-	-	-	-	-	-	-
OSC	-	-	-	-	-	-	-	-
OSCILL	-	-	-	-	-	-	-	-
OSCTRL	-	-	-	-	-	-	-	-
OSD	-	-	-	-	-	-	-	-
OSE	-	-	-	-	-	-	-	-
OSNSC	●	●	●	●	●	●	●	●
OSOF	-	-	-	-	-	-	-	-
OSP1	●	●	●	●	●	●	●	●
OSP2	●	●	●	●	●	●	●	●
OSS	-	-	-	-	-	-	-	-
OSSE	-	-	-	-	-	-	-	-
OST	-	-	-	-	-	-	-	-
OST1	●	●	●	●	●	●	●	●
OST2	●	●	●	●	●	●	●	●
OTOL	●	●	●	●	●	●	●	●
OVR	●	●	●	●	●	●	●	●
OVRA	●	●	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
P	●	●	●	●	●	●	●	●
PAROT	●	●	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●	●	●
PCALL	●	●	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-	-	-
PHI	-	-	-	-	-	-	-	-
PHU	●	●	●	●	●	●	●	●
PL	-	-	-	-	-	-	-	-
PM	●	●	●	●	●	●	●	●
PO	-	-	-	-	-	-	-	-
POCKET3	●	●	●	●	●	●	●	●
POCKET4	●	●	●	●	●	●	●	●
POLF	●	●	●	●	●	●	●	●
POLFA	●	●	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●	●	●
POLY	-	-	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-	-	-
PON	-	-	-	-	-	-	-	-
PONS	-	-	-	-	-	-	-	-
POS	●	●	●	●	●	●	●	●
POSA	●	●	●	●	●	●	●	●
POSM	●	●	●	●	●	●	●	●
POSMT	●	●	●	●	●	●	●	●
POSP	●	●	●	●	●	●	●	●
POSRANGE	●	●	●	●	●	●	●	●
POT	●	●	●	●	●	●	●	●
PR	●	●	●	●	●	●	●	●
PREPRO	●	●	●	●	●	●	●	●
PRESETON	●	●	●	●	●	●	●	●
PRESETONS	●	●	●	●	●	●	●	●
PRIO	●	●	●	●	●	●	●	●
PRLOC	●	●	●	●	●	●	●	●
PROC	●	●	●	●	●	●	●	●
PROTA	●	●	●	●	●	●	●	●
PROTD	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
PROTS	●	●	●	●	●	●	●	●
PSI	-	-	-	-	-	-	-	-
PTP	●	●	●	●	●	●	●	●
PTPG0	●	●	●	●	●	●	●	●
PTPWOC	●	●	●	●	●	●	●	●
PUNCHACC	-	-	-	-	-	-	-	-
PUTFTOC	●	●	●	●	●	●	●	●
PUTFTOCF	●	●	●	●	●	●	●	●
PW	○	○	○	○	○	○	○	○
QU	●	●	●	●	●	●	●	●
R...	●	●	●	●	●	●	●	●
RAC	●	●	●	●	●	●	●	●
RDISABLE	●	●	●	●	●	●	●	●
READ	●	●	●	●	●	●	●	●
REAL	●	●	●	●	●	●	●	●
RELEASE	●	●	●	●	●	●	●	●
REP	●	●	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●	●	●
REPOSH	●	●	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●	●	●
RET	●	●	●	●	●	●	●	●
RETB	●	●	●	●	●	●	●	●
RIC	●	●	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●	●	●
RMB	●	●	●	●	●	●	●	●
RME	●	●	●	●	●	●	●	●
RMI	●	●	●	●	●	●	●	●
RMN	●	●	●	●	●	●	●	●
RND	●	●	●	●	●	●	●	●
RNDM	●	●	●	●	●	●	●	●
ROT	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
ROTS	●	●	●	●	●	●	●	●
ROUND	●	●	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●	●	●
RP	●	●	●	●	●	●	●	●
RPL	●	●	●	●	●	●	●	●
RT	●	●	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●	●	●
RTLION	●	●	●	●	●	●	●	●

Instructions S ... Z

Instruction	SINUMERIK 828D							
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
S	●	●	●	●	●	●	●	●
SAVE	●	●	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●	●	●
SBLON	●	●	●	●	●	●	●	●
SC	●	●	●	●	●	●	●	●
SCALE	●	●	●	●	●	●	●	●
SCC	●	●	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●	●	●
SD	○	○	○	○	○	○	○	○
SET	●	●	●	●	●	●	●	●
SETAL	●	●	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●	●	●
SETINT	●	●	●	●	●	●	●	●
SETM	-	-	-	-	-	○	-	●
SETMS	●	●	●	●	●	●	●	●
SETMS(n)	●	●	●	●	●	●	●	●
SETMTH	●	●	●	●	●	●	●	●
SETPIECE	●	●	●	●	●	●	●	●
SETTA	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN Fraisage Exporta-tion (me42)	SW24x(5) Logiciel CN Tournage Exporta-tion (te42)	SW26x(3) Logiciel CN Fraisage Exporta-tion (me62)	SW26x(3) Logiciel CN Tournage Exporta-tion (te62)	SW28x(2) Logiciel CN Fraisage Exporta-tion (me821)	SW28x(1) Logiciel CN Fraisage Adv. Expor-tation (me822)	SW28x(2) Logiciel CN Tournage Exporta-tion (te821)	SW28x(1) Logiciel CN Tournage Adv. Expor-tation (te822)
SETTCOR	●	●	●	●	●	●	●	●
SETTIA	●	●	●	●	●	●	●	●
SF	●	●	●	●	●	●	●	●
SIN	●	●	●	●	●	●	●	●
SIRELAY	-	-	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-	-	-
SLOT1	●	●	●	●	●	●	●	●
SLOT2	●	●	●	●	●	●	●	●
SOFT	●	●	●	●	●	●	●	●
SOFTA	●	●	●	●	●	●	●	●
SON	-	-	-	-	-	-	-	-
SONS	-	-	-	-	-	-	-	-
SPATH	●	●	●	●	●	●	●	●
SPCOF	●	●	●	●	●	●	●	●
SPCON	●	●	●	●	●	●	●	●
SPI	●	●	●	●	●	●	●	●
SPIF1	-	-	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○	○	○
SPN	-	-	-	-	-	-	-	-
SPOF	-	-	-	-	-	-	-	-
SPOS	●	●	●	●	●	●	●	●
SPOSA	●	●	●	●	●	●	●	●
SPP	-	-	-	-	-	-	-	-
SPRINT	●	●	●	●	●	●	●	●
SQRT	●	●	●	●	●	●	●	●
SR	●	●	●	●	●	●	●	●
SRA	●	●	●	●	●	●	●	●
ST	●	●	●	●	●	●	●	●
STA	●	●	●	●	●	●	●	●
START	-	-	-	-	-	○	-	●
STARTFIFO	●	●	●	●	●	●	●	●
STAT	●	●	●	●	●	●	●	●
STOLF	●	●	●	●	●	●	●	●
STOPFIFO	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
STOPRE	●	●	●	●	●	●	●	●
STOPREOF	●	●	●	●	●	●	●	●
STRING	●	●	●	●	●	●	●	●
STRINGFELD	●	●	●	●	●	●	●	●
STRINGIS	●	●	●	●	●	●	●	●
STRLEN	●	●	●	●	●	●	●	●
SUBSTR	●	●	●	●	●	●	●	●
SUPA	●	●	●	●	●	●	●	●
SVC	●	●	●	●	●	●	●	●
SYNFCT	●	●	●	●	●	●	●	●
SYNR	●	●	●	●	●	●	●	●
SYNRW	●	●	●	●	●	●	●	●
SYNW	●	●	●	●	●	●	●	●
T	●	●	●	●	●	●	●	●
TAN	●	●	●	●	●	●	●	●
TANG	-	-	-	-	-	-	-	-
TANGDEL	-	-	-	-	-	-	-	-
TANGOF	-	-	-	-	-	-	-	-
TANGON	-	-	-	-	-	-	-	-
TCA (828D : _TCA)	●	●	●	●	●	●	●	●
TCARR	●	-	●	-	●	●	-	-
TCI	●	●	●	●	●	●	●	●
TCOABS	●	-	●	-	●	●	-	-
TCOFR	●	-	●	-	●	●	-	-
TCOFRX	●	-	●	-	●	●	-	-
TCOFRY	●	-	●	-	●	●	-	-
TCOFRZ	●	-	●	-	●	●	-	-
THETA	-	-	-	-	-	-	-	-
TILT	-	-	-	-	-	-	-	-
TLIFT	-	-	-	-	-	-	-	-
TML	●	●	●	●	●	●	●	●
TMOF	●	●	●	●	●	●	●	●
TMON	●	●	●	●	●	●	●	●
TO	●	●	●	●	●	●	●	●
TOFF	●	●	●	●	●	●	●	●
TOFFL	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
TOFFOF	●	●	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●	●	●
TOFFR	●	●	●	●	●	●	●	●
TOFRAME	●	●	●	●	●	●	●	●
TOFRAMEX	●	●	●	●	●	●	●	●
TOFRAMEY	●	●	●	●	●	●	●	●
TOFRAMEZ	●	●	●	●	●	●	●	●
TOLOWER	●	●	●	●	●	●	●	●
TOOLENV	●	●	●	●	●	●	●	●
TOOLGNT	●	●	●	●	●	●	●	●
TOOLGT	●	●	●	●	●	●	●	●
TOROT	●	●	●	●	●	●	●	●
TOROTOF	●	●	●	●	●	●	●	●
TOROTX	●	●	●	●	●	●	●	●
TOROTY	●	●	●	●	●	●	●	●
TOROTZ	●	●	●	●	●	●	●	●
TOUPPER	●	●	●	●	●	●	●	●
TOWBCS	●	-	●	-	●	●	-	-
TOWKCS	●	-	●	-	●	●	-	-
TOWMCS	●	-	●	-	●	●	-	-
TOWSTD	●	-	●	-	●	●	-	-
TOWTCS	●	-	●	-	●	●	-	-
TOWWCS	●	-	●	-	●	●	-	-
TR	●	●	●	●	●	●	●	●
TRAANG	-	-	-	-	-	-	○	○
TRACON	-	-	-	-	-	-	○	○
TRACYL	○	○	○	○	○	○	○	○
TRAFOOF	●	●	●	●	●	●	●	●
TRAFOON	-	-	-	-	-	-	-	-
TRAILOF	●	●	●	●	●	●	●	●
TRAILON	●	●	●	●	●	●	●	●
TRANS	●	●	●	●	●	●	●	●
TRANSMIT	○	○	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-	-	-
TRUE	●	●	●	●	●	●	●	●
TRUNC	●	●	●	●	●	●	●	●
TU	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN Fraisage Exportation (me42)	SW24x(5) Logiciel CN Tournage Exportation (te42)	SW26x(3) Logiciel CN Fraisage Exportation (me62)	SW26x(3) Logiciel CN Tournage Exportation (te62)	SW28x(2) Logiciel CN Fraisage Exportation (me821)	SW28x(1) Logiciel CN Fraisage Adv. Exportation (me822)	SW28x(2) Logiciel CN Tournage Exportation (te821)	SW28x(1) Logiciel CN Tournage Adv. Exportation (te822)
TURN	●	●	●	●	●	●	●	●
ULI	●	●	●	●	●	●	●	●
UNLOCK	●	●	●	●	●	●	●	●
UNTIL	●	●	●	●	●	●	●	●
UPATH	●	●	●	●	●	●	●	●
VAR	●	●	●	●	●	●	●	●
VELOLIM	●	●	●	●	●	●	●	●
VELOLIMA	●	●	●	●	●	●	●	●
WAITC	●	●	●	●	●	●	●	●
WAITE	-	-	-	-	-	○	-	●
WAITENC	●	●	●	●	●	●	●	●
WAITM	-	-	-	-	-	○	-	●
WAITMC	-	-	-	-	-	○	-	●
WAITP	●	●	●	●	●	●	●	●
WAITS	●	●	●	●	●	●	●	●
WALCS0	●	●	●	●	●	●	●	●
WALCS1	●	●	●	●	●	●	●	●
WALCS2	●	●	●	●	●	●	●	●
WALCS3	●	●	●	●	●	●	●	●
WALCS4	●	●	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●	●	●
WALCS8	●	●	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●	●	●
WHEN	●	●	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●	●	●
WHILE	●	●	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●	●	●
WRITE	●	●	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●	●	●
X	●	●	●	●	●	●	●	●
XOR	●	●	●	●	●	●	●	●

Instruction	SINUMERIK 828D							
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN Fraisage Exporta-tion (me42)	SW24x(5) Logiciel CN Tournage Exporta-tion (te42)	SW26x(3) Logiciel CN Fraisage Exporta-tion (me62)	SW26x(3) Logiciel CN Tournage Exporta-tion (te62)	SW28x(2) Logiciel CN Fraisage Exporta-tion (me821)	SW28x(1) Logiciel CN Fraisage Adv. Expor-tation (me822)	SW28x(2) Logiciel CN Tournage Exporta-tion (te821)	SW28x(1) Logiciel CN Tournage Adv. Expor-tation (te822)
Y	●	●	●	●	●	●	●	●
Z	●	●	●	●	●	●	●	●

21.2.2 Variantes de commande rectification

Instructions A ... C

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
:	●	●	●	●	●	●
*	●	●	●	●	●	●
+	●	●	●	●	●	●
-	●	●	●	●	●	●
<	●	●	●	●	●	●
<<	●	●	●	●	●	●
<=	●	●	●	●	●	●
=	●	●	●	●	●	●
>=	●	●	●	●	●	●
/	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●
A	●	●	●	●	●	●
A2	-	-	-	-	-	-
A3	-	-	-	-	-	-
A4	-	-	-	-	-	-
A5	-	-	-	-	-	-
A6	-	-	-	-	-	-
A7	-	-	-	-	-	-
ABS	●	●	●	●	●	●
AC	●	●	●	●	●	●
ACC	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
ACN	●	●	●	●	●	●
ACOS	●	●	●	●	●	●
ACP	●	●	●	●	●	●
ACTBLOCNO	●	●	●	●	●	●
ADDFRAME	●	●	●	●	●	●
ADIS	●	●	●	●	●	●
ADISPOS	●	●	●	●	●	●
ADISPOSA	●	●	●	●	●	●
ALF	●	●	●	●	●	●
AMIRROR	●	●	●	●	●	●
AND	●	●	●	●	●	●
ANG	●	●	●	●	●	●
AP	●	●	●	●	●	●
APR	●	●	●	●	●	●
APRB	●	●	●	●	●	●
APRP	●	●	●	●	●	●
APW	●	●	●	●	●	●
APWB	●	●	●	●	●	●
APWP	●	●	●	●	●	●
APX	●	●	●	●	●	●
AR	●	●	●	●	●	●
AROT	●	●	●	●	●	●
AROTS	●	●	●	●	●	●
AS	●	●	●	●	●	●
ASCALE	●	●	●	●	●	●
ASIN	●	●	●	●	●	●
ASPLINE	○	○	○	○	○	○
ATAN2	●	●	●	●	●	●
ATOL	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●
AX	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
AXIS	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●
B	●	●	●	●	●	●
B2	-	-	-	-	-	-
B3	-	-	-	-	-	-
B4	-	-	-	-	-	-
B5	-	-	-	-	-	-
B6	-	-	-	-	-	-
B7	-	-	-	-	-	-
B_AND	●	●	●	●	●	●
B_OR	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○
BLOC	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●
BNAT	○	○	○	○	○	○
BOOL	●	●	●	●	●	●
BOUND	●	●	●	●	●	●
BRISK	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○
BTAN	○	○	○	○	○	○
C	●	●	●	●	●	●
C2	-	-	Nom d'axe de canal	-	-	-
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
C6	-	-	-	-	-	-
C7	-	-	-	-	-	-
CAC	●	●	●	●	●	●
CACN	●	●	●	●	●	●
CACP	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
CALCPOSI	●	●	●	●	●	●
CALL	●	●	●	●	●	●
CALLPATH	●	●	●	●	●	●
CANCEL	●	●	●	●	●	●
CASE	●	●	●	●	●	●
CDC	●	●	●	●	●	●
CDOF	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-
CDON	-	-	-	-	-	-
CFC	●	●	●	●	●	●
CFIN	●	●	●	●	●	●
CFINE	●	●	●	●	●	●
CFTCP	●	●	●	●	●	●
CHAN	●	●	●	●	●	●
CHANDATA	●	●	●	●	●	●
CHAR	●	●	●	●	●	●
CHF	●	●	●	●	●	●
CHKDM	●	●	●	●	●	●
CHKDNO	●	●	●	●	●	●
CHR	●	●	●	●	●	●
CIC	●	●	●	●	●	●
CIP	●	●	●	●	●	●
CLEARM	-	-	●	-	-	●
CLRINT	●	●	●	●	●	●
CMIRROR	●	●	●	●	●	●
COARSEA	●	●	●	●	●	●
COLLPAIR	-	-	-	-	-	-
COMPCAD	●	●	●	●	●	●
COMPCURV	●	●	●	●	●	●
COMPLETE	●	●	●	●	●	●
COMPOF	●	●	●	●	●	●
COMPON	●	●	●	●	●	●
COMPSURF	-	-	-	-	-	-
CONTDCON	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●
CORROF	●	●	●	●	●	●
CORRTRAFO	-	-	-	-	-	-
COS	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
COUPDEF	○	○	○	○	○	○
COUPDEL	○	○	○	○	○	○
COUPOF	○	○	○	○	○	○
COUPOFS	○	○	○	○	○	○
COUPON	○	○	○	○	○	○
COUPONC	○	○	○	○	○	○
COUPRES	○	○	○	○	○	○
CP	●	●	●	●	●	●
CPBC	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○
CPLA	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○
CPLON	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○
CPOF	○	○	○	○	○	○
CPON	○	○	○	○	○	○

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
CPRECOF	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●
CPRES	○	○	○	○	○	○
CPROT	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○
CR	●	●	●	●	●	●
CROT	●	●	●	●	●	●
CROTS	●	●	●	●	●	●
CRPL	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○
CT	●	●	●	●	●	●
CTAB	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-
CTABID	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
CTABPOLID	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-
CTOL	●	●	●	●	●	●
CTRANS	●	●	●	●	●	●
CUT2D	●	●	●	●	●	●
CUT2DD	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●
CUT2DFD	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-
CYCLE60	-	-	-	-	-	-
CYCLE61	-	-	-	-	-	-
CYCLE62	●	●	●	●	●	●
CYCLE63	-	-	-	-	-	-
CYCLE64	-	-	-	-	-	-
CYCLE70	-	-	-	-	-	-
CYCLE72	-	-	-	-	-	-
CYCLE76	-	-	-	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
CYCLE77	-	-	-	-	-	-
CYCLE78	-	-	-	-	-	-
CYCLE79	-	-	-	-	-	-
CYCLE81	-	-	-	-	-	-
CYCLE82	-	-	-	-	-	-
CYCLE83	-	-	-	-	-	-
CYCLE84	-	-	-	-	-	-
CYCLE85	-	-	-	-	-	-
CYCLE86	-	-	-	-	-	-
CYCLE92	-	-	-	-	-	-
CYCLE95	-	-	-	-	-	-
CYCLE98	-	-	-	-	-	-
CYCLE99	-	-	-	-	-	-
CYCLE150	-	-	-	-	-	-
CYCLE435	○	○	○	○	○	○
CYCLE495	○	○	○	○	○	○
CYCLE750	●	●	●	●	●	●
CYCLE751	●	●	●	●	●	●
CYCLE752	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●
CYCLE758	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●
CYCLE800	○	○	○	○	○	○
CYCLE801	-	-	-	-	-	-
CYCLE802	-	-	-	-	-	-
CYCLE830	-	-	-	-	-	-
CYCLE832	●	●	●	●	●	●
CYCLE840	-	-	-	-	-	-
CYCLE899	-	-	-	-	-	-
CYCLE930	-	-	-	-	-	-
CYCLE940	-	-	-	-	-	-
CYCLE951	-	-	-	-	-	-
CYCLE952	-	-	-	-	-	-
CYCLE961	-	-	-	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
CYCLE971	-	-	-	-	-	-
CYCLE973	-	-	-	-	-	-
CYCLE974	-	-	-	-	-	-
CYCLE976	-	-	-	-	-	-
CYCLE977	-	-	-	-	-	-
CYCLE978	-	-	-	-	-	-
CYCLE979	-	-	-	-	-	-
CYCLE982	-	-	-	-	-	-
CYCLE994	-	-	-	-	-	-
CYCLE995	-	-	-	-	-	-
CYCLE996	-	-	-	-	-	-
CYCLE997	-	-	-	-	-	-
CYCLE998	-	-	-	-	-	-
CYCLE4071	●	●	●	●	●	●
CYCLE4072	●	●	●	●	●	●
CYCLE4073	●	●	●	●	●	●
CYCLE4074	●	●	●	●	●	●
CYCLE4075	●	●	●	●	●	●
CYCLE4077	●	●	●	●	●	●
CYCLE4078	●	●	●	●	●	●
CYCLE4079	●	●	●	●	●	●

Instructions D ... F

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
D	●	●	●	●	●	●
D0	●	●	●	●	●	●
DAC	●	●	●	●	●	●
DC	●	●	●	●	●	●
DCI	●	●	●	●	●	●
DCM	●	●	●	●	●	●
DCU	●	●	●	●	●	●
DEF	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
DEFINE	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●
DELDL	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●
DELETE	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●
DELMLRES	●	●	●	●	●	●
DELMT	-	-	-	-	-	-
DELOBJ	-	-	-	-	-	-
DELT	●	●	●	●	●	●
DELTC	●	●	●	●	●	●
DELTOOLENV	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●
DIAMOF	●	●	●	●	●	●
DIAMOFA	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●
DIC	●	●	●	●	●	●
DILF	●	●	●	●	●	●
DISABLE	●	●	●	●	●	●
DISC	●	●	●	●	●	●
DISCL	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●
DISPR	●	●	●	●	●	●
DISR	●	●	●	●	●	●
DISRP	●	●	●	●	●	●
DITE	●	●	●	●	●	●
DITS	●	●	●	●	●	●
DIV	●	●	●	●	●	●
DL	-	-	-	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
DO	●	●	●	●	●	●
DRFOF	●	●	●	●	●	●
DRIVE	●	●	●	●	●	●
DRIVEA	●	●	●	●	●	●
DYNFINISH	●	●	●	●	●	●
DYNNORM	●	●	●	●	●	●
DYNPOS	●	●	●	●	●	●
DYNROUGH	●	●	●	●	●	●
DYNSEMIFIN	●	●	●	●	●	●
DZERO	●	●	●	●	●	●
EAUTO	○	○	○	○	○	○
EGDEF	○	○	○	○	○	○
EGDEL	○	○	○	○	○	○
EGOFC	○	○	○	○	○	○
EGOFS	○	○	○	○	○	○
EGON	○	○	○	○	○	○
EGONSYN	○	○	○	○	○	○
EGONSYNE	○	○	○	○	○	○
ELSE	●	●	●	●	●	●
ENABLE	●	●	●	●	●	●
ENAT	○	○	○	○	○	○
ENDFOR	●	●	●	●	●	●
ENDIF	●	●	●	●	●	●
ENDLABEL	●	●	●	●	●	●
ENDLOOP	●	●	●	●	●	●
ENDPROC	●	●	●	●	●	●
ENDWHILE	●	●	●	●	●	●
ESRR	○	○	○	○	○	○
ESRS	○	○	○	○	○	○
ETAN	○	○	○	○	○	○
EVERY	●	●	●	●	●	●
EX	●	●	●	●	●	●
EXECSTRING	●	●	●	●	●	●
EXECTAB	●	●	●	●	●	●
EXECUTE	●	●	●	●	●	●
EXP	●	●	●	●	●	●
EXTCALL	●	●	●	●	●	●
EXTCLOSE	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
EXTERN	●	●	●	●	●	●
EXTOPEN	●	●	●	●	●	●
F	●	●	●	●	●	●
FA	●	●	●	●	●	●
FAD	●	●	●	●	●	●
FALSE	●	●	●	●	●	●
FB	●	●	●	●	●	●
FCTDEF	●	●	●	●	●	●
FCUB	●	●	●	●	●	●
FD	●	●	●	●	●	●
FDA	●	●	●	●	●	●
FENDNORM	●	●	●	●	●	●
FFWOF	●	●	●	●	●	●
FFWON	●	●	●	●	●	●
FGREF	●	●	●	●	●	●
FGROUP	●	●	●	●	●	●
FI	●	●	●	●	●	●
FIFOCTRL	●	●	●	●	●	●
FILEDATE	●	●	●	●	●	●
FILEINFO	●	●	●	●	●	●
FILESIZE	●	●	●	●	●	●
FILESTAT	●	●	●	●	●	●
FILETIME	●	●	●	●	●	●
FINEA	●	●	●	●	●	●
FL	●	●	●	●	●	●
FLIN	●	●	●	●	●	●
FMA	●	●	●	●	●	●
FNORM	●	●	●	●	●	●
FOCOF	○	○	○	○	○	○
FOCON	○	○	○	○	○	○
FOR	●	●	●	●	●	●
FP	●	●	●	●	●	●
FPO	-	-	-	-	-	-
FPR	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●
FRAME	●	●	●	●	●	●
FRC	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
FRCM	●	●	●	●	●	●
FROM	●	●	●	●	●	●
FTOC	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●
FXS	●	●	●	●	●	●
FXST	●	●	●	●	●	●
FXSW	●	●	●	●	●	●
FZ	●	●	●	●	●	●

Instructions G ... L

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
G0	●	●	●	●	●	●
G1	●	●	●	●	●	●
G2	●	●	●	●	●	●
G3	●	●	●	●	●	●
G4	●	●	●	●	●	●
G5	●	●	●	●	●	●
G7	●	●	●	●	●	●
G9	●	●	●	●	●	●
G17	●	●	●	●	●	●
G18	●	●	●	●	●	●
G19	●	●	●	●	●	●
G25	●	●	●	●	●	●
G26	●	●	●	●	●	●
G33	●	●	●	●	●	●
G34	●	●	●	●	●	●
G35	●	●	●	●	●	●
G40	●	●	●	●	●	●
G41	●	●	●	●	●	●
G42	●	●	●	●	●	●
G53	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
G54	●	●	●	●	●	●
G55	●	●	●	●	●	●
G56	●	●	●	●	●	●
G57	●	●	●	●	●	●
G58	→ G505					
G59	→ G506					
G60	●	●	●	●	●	●
G62	●	●	●	●	●	●
G63	●	●	●	●	●	●
G64	●	●	●	●	●	●
G70	●	●	●	●	●	●
G71	●	●	●	●	●	●
G74	●	●	●	●	●	●
G75	●	●	●	●	●	●
G90	●	●	●	●	●	●
G91	●	●	●	●	●	●
G93	●	●	●	●	●	●
G94	●	●	●	●	●	●
G95	●	●	●	●	●	●
G96	●	●	●	●	●	●
G97	●	●	●	●	●	●
G110	●	●	●	●	●	●
G111	●	●	●	●	●	●
G112	●	●	●	●	●	●
G140	●	●	●	●	●	●
G141	●	●	●	●	●	●
G142	●	●	●	●	●	●
G143	●	●	●	●	●	●
G147	●	●	●	●	●	●
G148	●	●	●	●	●	●
G153	●	●	●	●	●	●
G247	●	●	●	●	●	●
G248	●	●	●	●	●	●
G290	●	●	●	●	●	●
G291	-	-	-	-	-	-
G331	●	●	●	●	●	●
G332	●	●	●	●	●	●
G335	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
G336	●	●	●	●	●	●
G340	●	●	●	●	●	●
G341	●	●	●	●	●	●
G347	●	●	●	●	●	●
G348	●	●	●	●	●	●
G450	●	●	●	●	●	●
G451	●	●	●	●	●	●
G460	●	●	●	●	●	●
G461	●	●	●	●	●	●
G462	●	●	●	●	●	●
G500	●	●	●	●	●	●
G505 ... G599	●	●	●	●	●	●
G601	●	●	●	●	●	●
G602	●	●	●	●	●	●
G603	●	●	●	●	●	●
G621	●	●	●	●	●	●
G641	●	●	●	●	●	●
G642	●	●	●	●	●	●
G643	●	●	●	●	●	●
G644	●	●	●	●	●	●
G645	●	●	●	●	●	●
G700	●	●	●	●	●	●
G710	●	●	●	●	●	●
G810 ... G819	-	-	-	-	-	-
G820 ... G829	-	-	-	-	-	-
G931	●	●	●	●	●	●
G942	●	●	●	●	●	●
G952	●	●	●	●	●	●
G961	●	●	●	●	●	●
G962	●	●	●	●	●	●
G971	●	●	●	●	●	●
G972	●	●	●	●	●	●
G973	●	●	●	●	●	●
GEOAX	●	●	●	●	●	●
GET	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●
GETD	-	-	●	-	-	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exportation (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exportation (gse82)
GETDNO	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●
GETSELT	●	●	●	●	●	●
GETT	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●
GETVARAP	●	●	●	●	●	●
GETVARDFT	●	●	●	●	●	●
GETVARLIM	●	●	●	●	●	●
GETVARPHU	●	●	●	●	●	●
GETVARTYP	●	●	●	●	●	●
GFRAME0 ... GFRAME100	< 50	< 100	< 100	< 50	< 100	< 100
GOTO	●	●	●	●	●	●
GOTOB	●	●	●	●	●	●
GOTOC	●	●	●	●	●	●
GOTOF	●	●	●	●	●	●
GOTOS	●	●	●	●	●	●
GP	●	●	●	●	●	●
GWPSOF	●	●	●	●	●	●
GROUP_ADDEND	●	●	●	●	●	●
GROUP_BEGIN	●	●	●	●	●	●
GROUP_END	●	●	●	●	●	●
GWPSON	●	●	●	●	●	●
H...	●	●	●	●	●	●
HOLES1	-	-	-	-	-	-
HOLES2	-	-	-	-	-	-
I	●	●	●	●	●	●
I1	●	●	●	●	●	●
IC	●	●	●	●	●	●
ICYCOF	●	●	●	●	●	●
ICYCON	●	●	●	●	●	●
ID	●	●	●	●	●	●
IDS	●	●	●	●	●	●
IF	●	●	●	●	●	●
INDEX	●	●	●	●	●	●
INIPO	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
INIRE	●	●	●	●	●	●
INICF	●	●	●	●	●	●
INIT	-	-	●	-	-	●
INITIAL						
INT	●	●	●	●	●	●
INTERSEC	●	●	●	●	●	●
INVCCW	-	-	-	-	-	-
INVCW	-	-	-	-	-	-
INVFRAME	●	●	●	●	●	●
IP	●	●	●	●	●	●
IPOBRKA	●	●	●	●	●	●
IPOENDA	●	●	●	●	●	●
IPTRLOCK	●	●	●	●	●	●
IPTRUNLOCK	●	●	●	●	●	●
IR	●	●	●	●	●	●
ISAXIS	●	●	●	●	●	●
ISD	-	-	-	-	-	-
ISFILE	●	●	●	●	●	●
ISNUMBER	●	●	●	●	●	●
ISOCALL	-	-	-	-	-	-
ISVAR	●	●	●	●	●	●
J	●	●	●	●	●	●
J1	●	●	●	●	●	●
JERKA	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●
JR	●	●	●	●	●	●
K	●	●	●	●	●	●
K1	●	●	●	●	●	●
KONT	●	●	●	●	●	●
KONTC	●	●	●	●	●	●
KONTT	●	●	●	●	●	●
KR	●	●	●	●	●	●
L	●	●	●	●	●	●
LEAD Orientation de l'outil Polyn. d'orientation	- -	- -	- -	- -	- -	- -
LEADOF	-	-	-	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
LEADON	-	-	-	-	-	-
LENTOAX	●	●	●	●	●	●
LFOF	●	●	●	●	●	●
LFON	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●
LFWP	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●
LIMS	●	●	●	●	●	●
LLI	●	●	●	●	●	●
LN	●	●	●	●	●	●
LOCK	●	●	●	●	●	●
LONGHOLE	-	-	-	-	-	-
LOOP	●	●	●	●	●	●

Instructions M ... R

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
M0	●	●	●	●	●	●
M1	●	●	●	●	●	●
M2	●	●	●	●	●	●
M3	●	●	●	●	●	●
M4	●	●	●	●	●	●
M5	●	●	●	●	●	●
M6	●	●	●	●	●	●
M17	●	●	●	●	●	●
M19	●	●	●	●	●	●
M30	●	●	●	●	●	●
M40	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●
M70	●	●	●	●	●	●
MASLDEF	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
MASLOF	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-
MASLON	-	-	-	-	-	-
MATCH	●	●	●	●	●	●
MAXVAL	●	●	●	●	●	●
MCALL	●	●	●	●	●	●
MEAC	-	○	○	-	○	○
MEAFRAME	●	●	●	●	●	●
MEAS	●	●	●	●	●	●
MEASA	-	○	○	-	○	○
MEASURE	●	●	●	●	●	●
MEAW	●	●	●	●	●	●
MEAWA	-	○	○	-	○	○
MI	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●
MMC	●	●	●	●	●	●
MOD	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●
MOV	●	●	●	●	●	●
MOVT	●	●	●	●	●	●
MSG	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●
N	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●
NCK	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●
NEWT	-	-	-	-	-	-
NORM	●	●	●	●	●	●
NOT	●	●	●	●	●	●
NPROT	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-
OF	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
OFFN	●	●	●	●	●	●
OMA1	-	-	-	-	-	-
OMA2	-	-	-	-	-	-
OMA3	-	-	-	-	-	-
OMA4	-	-	-	-	-	-
OMA5	-	-	-	-	-	-
OR	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-
ORIC	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-
ORID	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-
ORIS	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-
ORISON	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-
OS	○	○	○	○	○	○
OSB	○	○	○	○	○	○

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
OSC	-	-	-	-	-	-
OSCILL	○	○	○	○	○	○
OSCTRL	○	○	○	○	○	○
OSD	-	-	-	-	-	-
OSE	○	○	○	○	○	○
OSNSC	●	●	●	●	●	●
OSOF	-	-	-	-	-	-
OSP1	●	●	●	●	●	●
OSP2	●	●	●	●	●	●
OSS	-	-	-	-	-	-
OSSE	-	-	-	-	-	-
OST	-	-	-	-	-	-
OST1	●	●	●	●	●	●
OST2	●	●	●	●	●	●
OTOL	●	●	●	●	●	●
OVR	●	●	●	●	●	●
OVRA	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●
P	●	●	●	●	●	●
PAROT	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●
PCALL	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-
PHI	-	-	-	-	-	-
PHU	●	●	●	●	●	●
PL	-	-	-	-	-	-
PM	●	●	●	●	●	●
PO	-	-	-	-	-	-
POCKET3	-	-	-	-	-	-
POCKET4	-	-	-	-	-	-
POLF	●	●	●	●	●	●
POLFA	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●
POLY	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-
PON	-	-	-	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
PONS	-	-	-	-	-	-
POS	●	●	●	●	●	●
POSA	●	●	●	●	●	●
POSM	●	●	●	●	●	●
POSMT	-	-	-	-	-	-
POSP	●	●	●	●	●	●
POSRANGE	●	●	●	●	●	●
POT	●	●	●	●	●	●
PR	●	●	●	●	●	●
PREPRO	●	●	●	●	●	●
PRESETON	●	●	●	●	●	●
PRESETONS	●	●	●	●	●	●
PRIO	●	●	●	●	●	●
PRLOC	●	●	●	●	●	●
PROC	●	●	●	●	●	●
PROTA	●	●	●	●	●	●
PROTD	●	●	●	●	●	●
PROTS	●	●	●	●	●	●
PSI	-	-	-	-	-	-
PTP	●	●	●	●	●	●
PTPG0	●	●	●	●	●	●
PTPWOC	●	●	●	●	●	●
PUNCHACC	-	-	-	-	-	-
PUTFTOC	●	●	●	●	●	●
PUTFTOCF	●	●	●	●	●	●
PW	○	○	○	○	○	○
QU	●	●	●	●	●	●
R...	●	●	●	●	●	●
RAC	●	●	●	●	●	●
RDISABLE	●	●	●	●	●	●
READ	●	●	●	●	●	●
REAL	●	●	●	●	●	●
RELEASE	●	●	●	●	●	●
REP	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●
REPOSH	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
REPOSHA	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●
RET	●	●	●	●	●	●
RETB	●	●	●	●	●	●
RIC	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●
RMB	●	●	●	●	●	●
RME	●	●	●	●	●	●
RMI	●	●	●	●	●	●
RMN	●	●	●	●	●	●
RND	●	●	●	●	●	●
RNDM	●	●	●	●	●	●
ROT	●	●	●	●	●	●
ROTS	●	●	●	●	●	●
ROUND	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●
RP	●	●	●	●	●	●
RPL	●	●	●	●	●	●
RT	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●
RTLION	●	●	●	●	●	●

Instructions S ... Z

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
S	●	●	●	●	●	●
SAVE	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●
SBLON	●	●	●	●	●	●
SC	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
SCALE	●	●	●	●	●	●
SCC	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●
SD	○	○	○	○	○	○
SET	●	●	●	●	●	●
SETAL	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●
SETINT	●	●	●	●	●	●
SETM	-	-	●	-	-	●
SETMS	●	●	●	●	●	●
SETMS(n)	●	●	●	●	●	●
SETMTH	●	●	●	●	●	●
SETPIECE	●	●	●	●	●	●
SETTA	●	●	●	●	●	●
SETTCOR	●	●	●	●	●	●
SETTIA	●	●	●	●	●	●
SF	●	●	●	●	●	●
SIN	●	●	●	●	●	●
SIRELAY	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-
SLOT1	-	-	-	-	-	-
SLOT2	-	-	-	-	-	-
SOFT	●	●	●	●	●	●
SOFTA	●	●	●	●	●	●
SON	-	-	-	-	-	-
SONS	-	-	-	-	-	-
SPATH	●	●	●	●	●	●
SPCOF	●	●	●	●	●	●
SPCON	●	●	●	●	●	●
SPI	●	●	●	●	●	●
SPIF1	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○
SPN	-	-	-	-	-	-
SPOF	-	-	-	-	-	-
SPOS	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
SPOSA	●	●	●	●	●	●
SPP	-	-	-	-	-	-
SPRINT	●	●	●	●	●	●
SQRT	●	●	●	●	●	●
SR	●	●	●	●	●	●
SRA	●	●	●	●	●	●
ST	●	●	●	●	●	●
STA	●	●	●	●	●	●
START	-	-	●	-	-	●
STARTFIFO	●	●	●	●	●	●
STAT	●	●	●	●	●	●
STOLF	●	●	●	●	●	●
STOPFIFO	●	●	●	●	●	●
STOPRE	●	●	●	●	●	●
STOPREOF	●	●	●	●	●	●
STRING	●	●	●	●	●	●
STRINGFELD	●	●	●	●	●	●
STRINGIS	●	●	●	●	●	●
STRLEN	●	●	●	●	●	●
SUBSTR	●	●	●	●	●	●
SUPA	●	●	●	●	●	●
SVC	●	●	●	●	●	●
SYNFCT	●	●	●	●	●	●
SYNR	●	●	●	●	●	●
SYNRW	●	●	●	●	●	●
SYNW	●	●	●	●	●	●
T	●	●	●	●	●	●
TAN	●	●	●	●	●	●
TANG	○	○	○	○	○	○
TANGDEL	○	○	○	○	○	○
TANGOF	○	○	○	○	○	○
TANGON	○	○	○	○	○	○
TCA (828D : _TCA)	●	●	●	●	●	●
TCARR	●	●	●	●	●	●
TCI	●	●	●	●	●	●
TCOABS	●	●	●	●	●	●
TCOFR	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
avec TCOFRX	●	●	●	●	●	●
avec TCOFRY	●	●	●	●	●	●
TCOFRZ	●	●	●	●	●	●
THETA	-	-	-	-	-	-
TILT	-	-	-	-	-	-
TLIFT	○	○	○	○	○	○
TML	●	●	●	●	●	●
TMOF	●	●	●	●	●	●
TMON	●	●	●	●	●	●
TO	●	●	●	●	●	●
TOFF	●	●	●	●	●	●
TOFFL	●	●	●	●	●	●
TOFFOF	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●
TOFFR	●	●	●	●	●	●
TOFRAME	●	●	●	●	●	●
TOFRAMEX	●	●	●	●	●	●
TOFRAMEY	●	●	●	●	●	●
TOFRAMEZ	●	●	●	●	●	●
TOLOWER	●	●	●	●	●	●
TOOLENV	●	●	●	●	●	●
TOOLGNT	●	●	●	●	●	●
TOOLGT	●	●	●	●	●	●
TOROT	●	●	●	●	●	●
TOROTOF	●	●	●	●	●	●
TOROTX	●	●	●	●	●	●
TOROTY	●	●	●	●	●	●
TOROTZ	●	●	●	●	●	●
TOUPPER	●	●	●	●	●	●
TOWBCS	●	●	●	●	●	●
TOWKCS	●	●	●	●	●	●
TOWMCS	●	●	●	●	●	●
TOWSTD	●	●	●	●	●	●
TOWTCS	●	●	●	●	●	●
TOWWCS	●	●	●	●	●	●
TR	●	●	●	●	●	●
TRAANG	○	○	○	-	-	-
TRACON	○	○	○	-	-	-

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
TRACYL	○	○	○	○	○	○
TRAFOOF	●	●	●	●	●	●
TRAFOON	-	-	-	-	-	-
TRAILOF	●	●	●	●	●	●
TRAILON	●	●	●	●	●	●
TRANS	●	●	●	●	●	●
TRANSMIT	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-
TRUE	●	●	●	●	●	●
TRUNC	●	●	●	●	●	●
TU	●	●	●	●	●	●
TURN	●	●	●	●	●	●
ULI	●	●	●	●	●	●
UNLOCK	●	●	●	●	●	●
UNTIL	●	●	●	●	●	●
UPATH	●	●	●	●	●	●
VAR	●	●	●	●	●	●
VELOLIM	●	●	●	●	●	●
VELOLIMA	●	●	●	●	●	●
WAITC	●	●	●	●	●	●
WAITE	-	-	●	-	-	●
WAITENC	●	●	●	●	●	●
WAITM	-	-	●	-	-	●
WAITMC	-	-	●	-	-	●
WAITP	●	●	●	●	●	●
WAITS	●	●	●	●	●	●
WALCS0	●	●	●	●	●	●
WALCS1	●	●	●	●	●	●
WALCS2	●	●	●	●	●	●
WALCS3	●	●	●	●	●	●
WALCS4	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●
WALCS8	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●

Instruction	SINUMERIK 828D					
● Standard ○ Option - Non disponible	SW24x(5) Logiciel CN G-Tech Exportation (gce42)	SW26x(3) Logiciel CN G-Tech Exportation (gce62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gce82)	SW24x(5) Logiciel CN G-Tech Exportation (gse42)	SW26x(3) Logiciel CN G-Tech Exportation (gse62)	SW28x(1) Logiciel CN G-Tech Adv. Exporta-tion (gse82)
WALIMON	●	●	●	●	●	●
WHEN	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●
WHILE	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●
WRITE	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●
X	●	●	●	●	●	●
XOR	●	●	●	●	●	●
Y	●	●	●	●	●	●
Z	●	●	●	●	●	●

21.3 Langue actuelle dans l'IHM

Le tableau ci-après contient toutes les langues disponibles dans l'interface utilisateur.

La langue actuellement réglée peut être interrogée dans le programme pièces et dans les actions synchrones, au moyen de la variable système suivante :

$AN_LANGUAGE_ON_HMI = <valeur>

<valeur>	Langue	Code langue
1	Allemand (Allemagne)	DEU
2	Français	FRA
3	Anglais (Royaume-Uni)	ENG
4	Espagnol	ESP
6	Italien	ITA
7	Néerlandais	NLD
8	Chinois (simplifié)	CHS
9	Suédois	SVE
18	Hongrois	HUN
19	Finnois	FIN
28	Tchèque	CSY
50	Portugais (Brésil)	PTB
53	Polonais	PLK
55	Danois	DAN
57	Russe	RUS
68	Slovaque	SKY
72	Roumain	ROM
80	Chinois (traditionnel)	CHT
85	Coréen	KOR
87	Japonais	JPN
89	Turc	TRK

Remarque

L'actualisation de $AN_LANGUAGE_ON_HMI s'effectue :

- après le démarrage du système,
- après un Reset de la CN et/ou de l'AP
- après la commutation sur une autre CN dans le cadre de M2N,
- après le basculement de la langue sur l'IHM.

Annexe

A

A.1 Liste des abréviations

A	
A	Sortie
ADI4	Interface de commande analogique pour 4 axes
AC	Adaptive Control : commande adaptative
ALM	Active Line Module
ARM	Asynchroner rotatorischer Motor : moteur rotatif asynchrone
SA	Système d'automatisation
ASCII	American Standard Code for Information Interchange : code standard américain pour l'échange d'information
ASIC	Application Specific Integrated Circuit : circuit utilisateur
ASUP	Asynchrones Unterprogramm : sous-programme asynchrone
AUXFU	Auxiliary Function : fonction auxiliaire
LIST	Liste d'instructions
AWP	Anwenderprogramm : programme utilisateur

B	
MF	Mode de fonctionnement
GMFC	Groupe à mode de fonctionnement commun
BCD	Binary Coded Decimals : décimales codées en binaire
BERO	Détecteur de proximité sans contact
BI	Binector Input : entrée binecteur
FCOM	Combinaison de fonctions par interconnexion logicielle
BIN	Binary Files : fichiers binaires
BIOS	Basic Input Output System : système d'entrée-sortie de base
SCB	Système de coordonnées de base
BO	Binector Output : sortie binecteur
OPI	Operator Panel Interface : interface du pupitre opérateur

C	
CAO	Conception assistée par ordinateur
FAO	Fabrication assistée par ordinateur
CC	Compile Cycle : cycles de compilation
CEC	Cross Error Compensation
CI	Connector Input : entrée connecteur
Carte CF	carte Compact Flash

C	
CNC	Computerized Numerical Control : commande numérique par ordinateur
CO	Connector Output : sortie connecteur
CoL	Certificate of License : certificat de licence
COM	Communication
CPA	Compiler Projecting Data : données de configuration du compilateur
CRT	Cathode Ray Tube : tube cathodique
CSB	Central Service Board : module AP
CU	Control Unit
CP	Communication Processor : processeur de communication
CPU	Central Processing Unit : unité centrale de traitement
CR	Carriage Return : retour chariot
CTS	Clear To Send : signalisation de l'état Prêt à émettre (PAE) des interfaces série
CUTCOM	Cutter radius compensation : correction du rayon d'outil

D	
CNA	Convertisseur numérique-analogique
DB	Bloc de données (AP)
DBB	Octet de bloc de données (AP)
DBD	Double mot de bloc de données (AP)
DBW	Mot de bloc de données (AP)
DBX	Bit de bloc de données (AP)
DDE	Dynamic Data Exchange : échange dynamique de données
DDS	Drive Data Set : jeu de paramètres d'entraînement
DIN	Deutsche Industrie Norm : norme industrielle allemande
DIO	Data Input/Output : affichage de la transmission de données
DIR	Directory : répertoire
DLL	Dynamic Link Library : bibliothèque de liens dynamiques
DO	Drive Object : objet entraînement
DPM	Dual Port Memory : mémoire à double accès
DPR	Dual Port RAM : RAM à double accès
DRAM	Mémoire vive dynamique (non secourue)
DRF	Differential resolver function : fonction de résolveur différentiel (manivelle électronique)
DRIVE-CLiQ	Drive Component Link with IQ
DRY	Dry Run : avance de marche d'essai
DSB	Decoding Single Block : décodage bloc par bloc
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Mot de données
DWORD	Double mot (soit 32 bits)

E	
E	Entrée
EES	Execution from External Storage
E/S	Entrée/sortie
ENC	Encoder : codeur de valeurs réelles
EFP	Module de périphérie simple (module E/S de l'AP)
ESD	Electrostatic sensitive devices : composants sensibles aux décharges électrostatiques
CEM	Compatibilité électromagnétique
EN	Norme européenne
ENC	Encoder : codeur de valeurs réelles
EnDat	Interface de capteur
EPROM	Erasable Programmable Read Only Memory : mémoire morte reprogrammable
ePS Network Services	Services de télémaintenance de machines via Internet
EQN	Désignation de type d'un codeur absolu avec 2048 signaux sinusoïdaux par tour
ES	Système d'ingénierie
ESR	Arrêt étendu et retrait
ETC	Touche ETC ">" ; permet d'étendre la barre de touches logicielles dans le même menu

F	
FB	Bloc fonctionnel (AP)
FC	Function Call : Bloc fonctionnel (AP)
FEPROM	Flash EPROM : mémoire EPROM flash accessible en lecture/écriture
FIFO	First in First Out : premier entré, premier sorti ; mémoire fonctionnant sans adressage et dont les données sont lues dans l'ordre de leur stockage
IPOF	Interpolateur fin
FPU	Floating Point Unit : module à virgule flottante
CRF	Correction rayon de fraise
FST	Feed Stop : arrêt avance
LOG	Logigramme (méthode de programmation pour AP)
FW	Firmware

G	
GC	Global Control (PROFIBUS : télégramme de diffusion PROFIBUS
GDIR	Mémoire globale de programmes pièce
GEO	Géométrique, par exemple axe géométrique
GIA	Gear Interpolation Data : données d'interpolation de réducteur
GND	Signal Ground : retour commun du signal
GP	Programme de base (AP)
GS	Rapport de transmission
GSD	Désigne le fichier de données de base de l'appareil décrivant un esclave PROFIBUS

G	
GSDML	Generic Station Description Markup Language : langage de description basé sur XML permettant de créer un fichier GSD
GUD	Global User Data : données utilisateur globales

H	
HEX	Abréviation pour nombre hexadécimal
F-aux	Fonction auxiliaire
HLA	Hydraulischer Linearantrieb : entraînement linéaire hydraulique
IHM	Interface homme-machine : interface utilisateur SINUMERIK
EBR	Entraînement de la broche principale
HW	Hardware : matériel

I	
MeS	Mise en service
IKA	Interpolatorische Kompensation : compensation avec interpolation
IM	Interface module : module de couplage
IMR	Interface Module Receive : carte de couplage pour réception
IMS	Interface Module Send : carte de couplage pour émission
INC	Increment : déplacement en manuel incrémental
INI	Initializing Data : données d'initialisation
IPO	Interpolateur
ISA	International Standard Architecture : architecture standard internationale
ISO	International Organization for Standardization : Organisation internationale de normalisation

J	
JOG	Jogging : mode réglage

K	
K_v	Facteur de gain de la régulation
K_p	Gain proportionnel
R_t	Rapport de transmission
CONT	Schéma à contacts (méthode de programmation pour AP)

L	
LAI	Logic Machine Axis Image : représentation logique des axes machine
LAN	Local Area Network : réseau local
LCD	Liquid Crystal Display : afficheur à cristaux liquides
LED	Light Emitting Diode : diode électroluminescente
LF	Line Feed : présentation de ligne

L	
LMS	Lagemesssystem : système de mesure de position
LR	Lageregler : régulateur de position
LSB	Least Significant Bit : bit de poids faible
LUD	Local User Data : données utilisateur (locales)

M	
MAC	Media Access Control : contrôle d'accès au support
MAIN	Main program : programme principal (OB1, AP)
Mo	Mégaoctet
MCI	Motion Control Interface : interface de pilotage des déplacements
MCIS	Motion Control Information System : système d'information de pilotage des déplacements
MCP	Machine Control Panel : tableau de commande machine
PM	Paramètre(s) machine
MDA	Manual Data Automatic : saisie manuelle des données
MDS	Motor Data Set : jeu de paramètres moteur
MELDW	Mot de signalisation
SCM	Système de coordonnées machine
MM	Motor Module
MPF	Main Program File : programme principal (CN)
TCM	Tableau de commande machine

N	
CN	Commande commande numérique avec traitement de blocs, plage de déplacement, etc.
NCU	Numerical Control Unit : unité matérielle de la CN
NRK	Désignation du système d'exploitation de la CN
SI	Signal d'interface
NURBS	Non Uniform Rational B-Spline : courbes NURBS
DO	Décalage d'origine
NX	Numerical Extension : module d'extension d'axe

O	
OB	Bloc d'organisation dans l'AP
OEM	Original Equipment Manufacturer : fabricant d'équipement d'origine
OP	Operator Panel : pupitre opérateur
OPI	Operator Panel Interface : interface du tableau de commande
OPT	Options : Options
OLP	Optical Link Plug : connecteur optique de raccordement au bus
OSI	Open Systems Interconnection : normalisation de la communication entre ordinateurs

P	
MIS	Mémoire image des sorties
MIE	Mémoire image des entrées
PC	Personal Computer : ordinateur personnel
PCIN	Nom du logiciel pour l'échange de données avec la commande
PCMCIA	Personal Computer Memory Card International Association : normalisation des cartes mémoire enfichables
PCU	PC Unit : boîtier PC (unité de traitement)
PG	Console de programmation
PKE	Identificateur de paramètre : zone de PKW
PKW	Identificateur de paramètre : zone valeur de paramètre d'un PPO
AP	Automate programmable
PN	PROFINET
PNO	PROFIBUS Nutzerorganisation : Association des utilisateurs PROFIBUS
PO	POWER ON
POE	Programmorganisationseinheit : unité d'organisation de programme
POS	Position/Positionnement
POSMO A	Positioning Motor Actuator : moteur de positionnement
POSMO CA	Positioning Motor Compact AC : unité d'entraînement complète avec module de puissance et de régulation intégrés, plus unité de positionnement et mémoire de programmes ; alimentation en courant alternatif
POSMO CD	Positioning Motor Compact DC : comme POSMO CA, sauf l'alimentation qui est en courant continu
POSMO SI	Positioning Motor Servo Integrated : moteur de positionnement ; alimentation en courant continu
PPO	Parameter Prozessdaten Objekt : télégramme de données cyclique lors de la transmission avec PROFIBUS-DP et le profil "Entraînements à vitesse variable"
PPU	Panel Processing Unit : matériel central d'une commande CNC à pupitre opérateur, p. ex. SINUMERIK 828D
PROFIBUS	Process Field Bus : bus de données série
PRT	Test du programme
PSW	Programmsteuerwort : mot de commande de programme
PTP	Point to Point : point à point
PUD	Program Global User Data : variable utilisateur globale d'un programme
PZD	Prozessdaten : partie données process d'un PPO

Q	
QFK	Quadrantenfehler Kompensation : compensation des défauts aux transitions entre quadrants

R	
RAM	Random Access Memory : mémoire vive accessible en lecture/écriture
REF	Fonction accostage de point de référence

R	
REPOS	Fonction repositionnement
RISC	Reduced Instruction Set Computer : type de processeur à jeu d'instructions réduit et exécution rapide des instructions
ROV	Rapid Override : correction du rapide
RP	Paramètre R : paramètre de calcul, variable utilisateur prédéfinie
RPA	R-Parameter Active : Zone de mémoire de la CN pour numéros de paramètres R
RPY	Roll Pitch Yaw : type de rotation d'un système de coordonnées
RTLI	Rapid Traverse Linear Interpolation : interpolation linéaire en mode de déplacement à vitesse rapide
RTS	Request To Send : signalisation de commande d'interfaces de données série, demande d'émission
RTCP	Real Time Control Protocol : protocole de commande en temps réel

S	
AS	Action synchrone
SBC	Safe Brake Control : commande sûre de frein
SBL	Single Block : bloc par bloc
SBR	Subroutine : sous-programme (AP)
SD	Settingdatum ou Settingdaten : donnée(s) de réglage
SDB	Bloc de données système
SEA	Setting Data Active : identificateur (type de fichier) pour données de réglage
SERUPRO	Search Run by Program Test : Recherche en mode test du programme
SFB	System Funktionsbaustein : bloc fonctionnel système
SFC	System Function Call : appel de fonction système
F-DI	Failsafe Digital Input : entrée TOR de sécurité
F-DO	Failsafe Digital Output : sortie TOR de sécurité
STO	Safe Torque Off : suppression sûre du couple
SIM	Single in Line Module
TL	Touche logicielle
SKP	Skip : fonction de masquage d'un bloc de programme pièce
SLM	Synchroner Linearmotor : moteur linéaire synchrone
SM	Schrittmotor : moteur pas à pas
SMC	Sensor Module Cabinet Mounted
SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SPF	Sub Program File : fichier de sous-programme (CN)
AP	Automate programmable
SRAM	Mémoire statique (secourue)
CRP	Correction de rayon de plaquette
SRM	Synchron rotatorischer Motor : moteur rotatif synchrone
CEPV	Compensation d'erreur de pas de vis de transmission
SSI	Serial Synchron Interface : interface série synchrone
SSL	Satzsuchlauf : recherche de bloc

S	
STW	Steuerwort : mot de commande
VCM	Vitesse circonférentielle de meule
SW	Logiciel
SYF	System Files : fichiers système
SYNACT	Synchronized Action : action synchrone

T	
TB	Terminal Board (SINAMICS)
TCP	Tool Center Point : pointe d'outil
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active : identificateur de paramètres machine
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset : correction d'outil
TOA	Tool Offset Active : identificateur (type de fichier) pour corrections d'outil
TRANSMIT	Transform Milling into Turning : transformation des coordonnées pour des opérations de fraisage sur un tour
TTL	Transistor-Transistor Logic (type d'interface)
TZ	Technologiezyklus : cycle technologique

U	
UFR	User Frame : décalage d'origine
SP	Sous-programme
USB	Universal Serial Bus : bus série universel
ASI	Alimentation sans interruption

V	
VDI	Interface de communication interne entre CN et AP
VDI	Verein Deutscher Ingenieure : Association des ingénieurs allemands
VDE	Verband Deutscher Elektrotechniker : Association des électrotechniciens allemands
VI	Voltage Input : entrée de tension
VO	Voltage Output : sortie de tension
EAV	Entraînement d'avance

W	
ARD	Fonction d'accostage et de retrait en douceur
SCP	Système de coordonnées pièce
Out.	Outil
CLO	Correction de longueur d'outil

W	
WOP	Werkstatt-orientierte Programmierung : programmation orientée atelier
WPD	Work Piece Directory : répertoire pièce
CRO	Correction du rayon d'outil
Out.	Outil
CO	Correction d'outil
GO	Gestion des outils
WZW	Changement d'outil

X	
XML	Extensible Markup Language : langage de balisage extensible

Z	
ZOA	Zero Offset Active : identificateur de décalage d'origine
ZSW	Zustandswort : mot d'état (de l'entraînement)

A.2 Vue d'ensemble de la documentation

Documentation générale

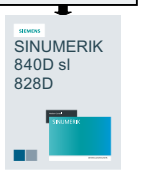
Publication publicitaire
– SINUMERIK 840D sl
– SINUMERIK 828D
– SINUMERIK 828D BASIC

Catalogue NC 62
SINUMERIK 840D sl

Catalogue NC 82
SINUMERIK 828D

Catalogue PM 21
SIMOTION,
SINAMICS S120

Manuel système
Commande intelligente
de la machine-outil



Manuel système
Ctrl-Energy

Documentation utilisateur

Manuel d'utilisation
– Universel
- Tournage
– Fraisage
– Rectification

Manuel de programmation
– Notions de base
– Notions complémentaires
– Cycles de mesure

Manuel de programmation
– Tournage ISO
– Fraisage ISO

Manuel de diagnostic
Alarmes

Manuel de
diagnostic
Alarmes

Documentation constructeur / S.A.V.

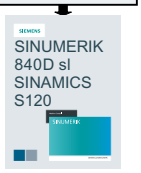
Manuel du produit
– NCU
– Éléments de commande et mise en réseau
– ADI4

Manuel du produit
Manuel de mise en service
Manuel de maintenance

Manuel de mise en service
– CNC : NCK, AP, entraînement
– Logiciel de base et logiciel de commande

Tables de paramètres
– Paramètres machine
– Signaux d'interface
– Variables

Tables de paramètres
– Paramètres machine
– Signaux d'interface
– Paramètres
– Variables



Manuel système
Guide de configuration
des machines

Documentation constructeur / S.A.V.

Description fonctionnelle
– Fonctions de base
– Fonctions d'extension
– Fonctions spéciales
– Actions synchrones
– Dialectes ISO

Description fonction-
nelle
Gestion des outils

Description fonctionnelle
Safety Integrated

Description fonction-
nelle
Safety Integrated

Manuel de configuration
– Directives d'installation CEM
– Industrial Security

Info / Formation

Documentation de formation
– Fraisage plus simple avec ShopMill
– Tournage plus simple avec ShopTurn

Manuel
Fabrication d'outils et de moules

Documentation électronique

DOConCD

Industry Online Support (SIOS)

Industry Mall

Glossaire

Accélération avec limitation des à-coups

Pour obtenir une accélération optimale tout en ménageant les organes mécaniques de la machine, il est possible de basculer, dans le programme d'usinage, entre accélération sous forme d'échelon et accélération continue (sans à-coup).

Accostage d'un point fixe

Les machines-outils peuvent accoster des points fixes qui ont été définis, tels que point de changement d'outil, point de chargement, point de changement de palette, etc. Les coordonnées de ces points sont mémorisées dans la commande. La commande déplace les axes concernés dans la mesure du possible, en → mode rapide.

Actions synchrones

1. Sortie de fonction auxiliaire

   Pendant l'usinage de la pièce, il est possible de sortir du programme CNC des fonctions technologiques (→ fonctions auxiliaires) pour les transmettre à l'AP. A l'aide de ces fonctions auxiliaires, il est possible, par exemple, de piloter des équipements supplémentaires de la machine-outil tels que fourreaux, préhenseurs, mandrins, etc.

2. Sortie de fonctions auxiliaires rapide

   Dans le cas de fonctions de commutation critiques en temps, il est possible de minimiser les délais d'acquittement pour les → fonctions auxiliaires et d'éviter les points d'arrêt inutiles dans le processus d'usinage.

Adresse

L'adresse permet d'identifier un opérande ou une plage d'opérandes déterminée, par ex. une entrée, une sortie etc.

Adresse d'axe

Voir → Nom d'axe

Alarmes

Tous les → messages et alarmes sont affichés à l'écran du tableau de commande, en texte clair, avec leur horodatage et le symbole correspondant au critère d'effacement. Les alarmes sont affichées séparément des messages.

1. Alarmes et messages dans le programme pièce
   L'affichage en texte clair des alarmes et messages peut être déclenché directement dans le programme pièce.
2. Alarmes et messages de l'AP
   Les alarmes et messages de la machine peuvent être affichés en texte clair directement à partir du programme AP. Des kits de blocs fonctionnels supplémentaires ne sont pas nécessaires.

Anticipation

La fonction **Look Ahead** est une fonction d'anticipation qui permet le pilotage optimal de la vitesse d'usinage, un nombre paramétrable de blocs de déplacement à l'avance.

AP

Automate **P**rogrammable : → Automate programmable. Composant de la → CN : Commande d'adaptation pour le traitement de la logique de la machine-outil.

Archivage

Sortie de fichiers et/ou répertoires sur une unité de mémorisation **externe**.

Arrêt orienté de broche

Arrêt de la broche de pièce dans une position angulaire prédéfinie, par exemple pour entreprendre un usinage complémentaire à un emplacement déterminé.

Arrêt précis

Dans le cas d'une instruction d'arrêt précis programmée, la position indiquée dans un bloc peut être accostée de façon précise et éventuellement très lentement. Pour limiter la durée d'approche, des → limites d'arrêt précis sont définies pour le mode rapide et l'avance.

Auto

Mode de fonctionnement d'un système à CN Mode de fonctionnement des systèmes de commandes numériques dans lequel un → programme pièce est sélectionné et exécuté de façon continue.

Automate programmable

Les automates programmables (AP) sont des commandes électroniques dont la fonction est mémorisée sous la forme d'un programme dans le mécanisme de commande. La structure et

le câblage de l'appareil ne dépendent donc pas de la fonction de l'automate. L'automate programmable possède la structure d'un calculateur. Elle se compose d'une CPU (unité centrale) avec mémoire, cartes d'entrée/sortie et système de bus interne. La périphérie et le langage de programmation sont orientés vers les besoins de la technique de commande.

Avance en inverse du temps

Pour le déplacement des axes, à la place de la vitesse d'avance, il est également possible de programmer le temps nécessaire au trajet d'un bloc (G93).

Avance tangentielle

L'avance tangentielle agit sur les si possible → axes à interpolation. Elle correspond à la somme géométrique des avances des → axes géométriques.

Axe à arrondissage

Les axes à arrondissage provoquent une rotation de la pièce ou de l'outil dans une position angulaire correspondant à une grille d'indexation. L'axe à arrondissage est considéré comme étant "en position" lorsque l'incrément d'indexation est atteint.

Axe à interpolation

Les axes à interpolation sont tous les axes d'usinage du → canal
qui sont pilotés par l' → interpolateur de telle manière qu'ils démarrent, accélèrent, s'arrêtent et atteignent le point final simultanément.

Axe C

Axe de la broche porte-pièce autour duquel des mouvements de rotation et de positionnement pilotés ont lieu.

Axe de base

Axe dont la valeur réelle ou la valeur de consigne est prise en compte pour le calcul d'une valeur de compensation.

Axe de compensation

Axe dont la valeur de consigne et la valeur réelle sont modifiées par la valeur de compensation.

Axe de positionnement

Axe exécutant un mouvement auxiliaire sur une machine-outil, (par exemple, magasin d'outils, transport de palettes). Les axes de positionnement sont des axes qui ne sont pas en interpolation avec les → axes d'interpolation.

Axe de synchronisme

L'axe synchrone est → l'axe Gantry dont la position de consigne est toujours dévirée du déplacement de → l'axe directeur, et dont le déplacement est donc synchrone. Du point de vue de l'utilisateur et du programmeur, l'axe synchrone "n'existe pas".

Axe directeur

L'axe directeur est → l'axe Gantry qui existe du point de vue de l'utilisateur et du programmeur, et qui peut donc être influencé comme un axe CN normal.

axe géométrique

Les axes géométriques forment le → système de coordonnées pièce bi- ou tridimensionnel dans lequel, dans les → programmes pièces, la géométrie de la pièce est programmée.

Axe linéaire

L'axe linéaire est un axe qui, contrairement à l'axe circulaire, décrit une droite.

Axe rotatif

Les axes rotatifs provoquent une rotation de la pièce ou de l'outil dans une position angulaire prédéfinie.

Axe synchrone

Pour effectuer leur course, les axes synchrones ont besoin du même temps que les axes géométriques pour leur trajectoire.

Axes

Selon leurs fonctionnalités, les axes CNC sont classés en deux catégories :

- Axes : à interpolation
- axes auxiliaires : axes de déplacement et de positionnement sans interpolation, mais avec une avance spécifique. Les axes auxiliaires ne sont généralement pas concernés par l'usinage proprement dit, par exemple les axes de dispositifs de chargement d'outils, de magasins d'outils.

Axes machine

Dans le cas d'une machine-outil, axes ayant une existence matérielle.

Bloc

On désigne par "blocs" tous les fichiers nécessaires à l’élaboration et à l’exécution du programme.

Bloc de données

1. Entité de données de → l'AP, accessible pour les programmes → HIGHSTEP.
2. Entité de données de la → CN : Les blocs de données contiennent des définitions de données utilisateur globales. Ces données peuvent être initialisées directement dans la définition.

Bloc de programme

Les blocs de programme contiennent les programmes principaux et sous-programmes des → programmes pièce.

Bloc de programme pièce

Partie d'un → programme pièce, par un LF (Line Feed). On distingue les → blocs principaux et les → blocs auxiliaires.

Bloc principal

Bloc introduit par ":" et contenant toutes les indications pour pouvoir exécuter une tâche dans un → programme pièce.

Bloc secondaire

Bloc introduit par "N", comportant des informations pour une opération, par exemple une indication de position.

Blocs intermédiaires

Les déplacements avec → correction d'outil sélectionnée (`G41`/`G42`) peuvent être interrompus par un nombre limité de blocs intermédiaires (blocs sans mouvement axial dans le plan de correction). Dans ce cas, la correction d'outil peut encore être prise en compte. Le nombre autorisé de blocs intermédiaires que la commande peut lire à l'avance peut être réglé par l'intermédiaire d'un paramètre système.

Canal

Un canal se distingue par le fait qu'il est en mesure d'exécuter un → programme pièce indépendamment des autres canaux. Le canal pilote exclusivement les axes et broches qui lui sont assignés. Les séquences d'un programme pièce issues de canaux différents peuvent être coordonnées par une → synchronisation.

Canal d'usinage

La structure multicanal permet de réduire les temps morts grâce à des déplacements en parallèle (par exemple le déplacement d'un portique de chargement simultanément à l'usinage). Un canal CNC est une commande CNC à part entière avec décodage, prétraitement des blocs et interpolation.

CN

Constituants de commande numérique de la → CNC qui exécutent les → programmes pièces et coordonnent les séquences de mouvements pour la machine-outil.

CNC

Voir → CN

Computerized Numerical Control : comprend les constituants → CN, → APR, IHM, → COM.

CNC

Voir → CN

Computerized Numerical Control : comprend les constituants → CN, → APR, IHM, → COM.

Code de programmation

Caractères et chaînes de caractères ayant une signification définie dans le langage de programmation pour les → programmes pièce.

COM

Module de la commande numérique pour l’exécution et la coordination de la communication.

Commande anticipatrice, dynamique

Les imprécisions de → contour dues aux écarts de traînage, peuvent être pratiquement éliminées à l'aide d'une commande anticipatrice dynamique dépendante de l'accélération. Il en résulte une précision d'usinage remarquable, même pour des → vitesses tangentielles élevées. La commande anticipatrice peut être sélectionnée et désélectionnée pour chaque axe, par l'intermédiaire du → programme pièce.

Commutateur à clé

Le commutateur à clé situé sur le → tableau de commande machine possède 4 positions, qui sont affectées à des fonctions par le système d'exploitation de la commande. Le commutateur à clé peut être actionné à l’aide de trois clés de différentes couleurs qui peuvent être retirées dans les positions indiquées.

Compensation avec interpolation

Les compensations en interpolation telles que → Compensation d'erreur de pas de vis de transmission, Compensation de la flèche, Compensation de l'angularité et Compensation de la température permettent de compenser les erreurs mécaniques de la machine.

Compensation d'erreur de pas de vis de transmission

Compensation, par la commande, des imprécisions mécaniques d'une vis à bille participant à l'avance, au moyen de valeurs mémorisées de mesure des écarts.

Compensation des défauts aux transitions entre quadrants

Les erreurs de contour survenant aux jonctions entre les quadrants, qui résultent de rapports de frottement variables au niveau des glissières, peuvent en grande partie être éliminées au moyen d'une compensation des défauts aux transitions entre quadrants. Le paramétrage de cette compensation passe par un test de circularité.

Compensation du jeu

Compensation du jeu mécanique d'une machine, par exemple dans le cas de vis à bille. Pour chaque axe, il est possible d'indiquer une compensation de jeu distincte.

Contour

Contour de la → pièce

Contour de la pièce finie

Contour de la pièce qui a été usinée. Voir → Pièce brute.

Contour de pièce

Contour programmé de la → pièce à fabriquer/usiner.

Contournage

Le but du contournage est d'éviter une décélération importante des → axes à interpolation aux limites de bloc du programme pièce et de passer au bloc suivant si possible avec la même vitesse tangentielle.

Coordonnées polaires

Système de coordonnées permettant de définir la position d’un point dans un plan par la distance qui sépare ce point de l’origine et par l’angle formé par le rayon vecteur avec un axe défini.

correction

Possibilité d'intervention manuelle ou programmable, qui permet à l'utilisateur de corriger des avances ou des vitesses de rotation programmées, pour les adapter à une pièce ou à un matériau spécifique.

Correction d'outil

Prise en compte des dimensions de l'outil lors du calcul de la trajectoire.

Correction de l'avance par commutateur

La correction de vitesse d'avance réglée par commutateur au → tableau de commande machine ou prescrite par l'→ AP (0-200 %) se superpose à la vitesse d'avance programmée.). La vitesse d'avance programmée peut faire l'objet d'une correction supplémentaire dans le programme d'usinage, par le biais d'un facteur programmable (de 1 à 200 %).

Correction de rayon de plaquette

La programmation d'un contour est basée sur le concept d'un outil parfaitement pointu ou tranchant. Ceci n'étant pas réalisable en pratique, le rayon du bec de la plaquette ou du tranchant de l'outil doit être indiqué à la commande numérique. Celle-ci le prend en compte et pilote le centre de la courbure selon une trajectoire décalée de la valeur du rayon et équidistante au contour.

Correction du rayon d'outil

Pour pouvoir programmer directement un → contour de pièce souhaité, la commande doit piloter, en tenant compte du rayon de l'outil utilisé, une trajectoire équidistante pour le contour programmé (`G41`/`G42`).

Cote absolue

Indication de la destination du déplacement d'un axe par une cote qui se rapporte à l'origine du système de coordonnées sélectionné. Voir → Cote relative.

Cote relative

Aussi appelée cote incrémentale : Programmation du point final de déplacement d'un axe au moyen d'une distance à parcourir et d'un sens, par référence à un point déjà atteint. Voir → Cote absolue.

Courbure

La courbure d'un contour est l'inverse du rayon r du cercle osculateur (cercle de courbure) : k = 1/r.

CPU

Central Processing Unit, voir → Automate programmable

Cycles

Sous-programmes protégés pour l'exécution d'usinages récurrents sur la → pièce.

Cycles standard

Des cycles standard sont disponibles pour les tâches d'usinage répétitives :

- pour le perçage/fraisage
- pour la technologie tournage

Dans le groupe fonctionnel "Programme", les cycles disponibles sont affichés dans le menu "Assistance aux cycles". Après sélection du cycle d'usinage souhaité, les paramètres sont affichés en clair pour l'affectation de valeurs.

Débit de transfert

Vitesse (bits/s) à laquelle une transmission de données a lieu.

Décalage d'origine

Spécification d'un nouveau point de référence pour un système de coordonnées, réglable en fonction d'une origine existante et d'un → frame.

1. réglable
   Un nombre configurable de décalages d'origine réglables est disponible pour chaque axe CNC. Les décalages sélectionnables par l'intermédiaire des fonctions G peuvent également être activés.
2. Externe
   En plus de tous les décalages qui définissent la position de l'origine pièce, on peut superposer un décalage d'origine externe par manivelle (décalage DRF) ou par le biais de l'AP.
3. Programmable

   Avec l'instruction `TRANS`, des décalages d'origine sont programmables pour tous les axes d'interpolation et de positionnement.

Décalage d'origine externe

Décalage d'origine imposé par → l'AP.

Définition de variables

Une définition de variable comprend la définition d'un type de données et un nom de variable. Avec le nom de la variable, la valeur des variables peut être activée.

déplacement en manuel incrémental

Indication de la longueur du trajet au moyen d'un nombre d'incréments (incrément). Le nombre d'incréments peut être mémorisé sous la forme de données de réglage ou sélectionné au moyen des touches marquées 10, 100, 1000, 10000.

des mots-clés

Mots ayant une notation fixe et qui, dans le langage de programmation, ont une signification particulière pour des → programmes pièces.

Descripteur

Les mots de la norme DIN 66025 sont complétés par des descripteurs (noms) de variables (variables de calcul, variables système, variables utilisateur), de sous-programmes, de mots-clés et de mots à plusieurs lettres adresse. Ces adjonctions ont la même signification que les mots dans la syntaxe des blocs. Les descripteurs doivent être uniques. Un même descripteur ne peut être utilisé pour désigner plusieurs objets.

Diagnostic

1. Groupe fonctionnel de la commande
2. La commande est dotée d'un programme d'autodiagnostic et d'aides aux tests pour la maintenance : Affichage d'états, d'alarmes et de données pour la maintenance

Données de réglage

Données qui communiquent des caractéristiques de la machine-outil à la commande numérique selon une procédure définie par le logiciel système.

DRF

Differential resolver function : Fonction de résolveur différentiel. Cette fonction entraîne un décalage d'origine incrémental en relation avec une manivelle électronique en mode automatique.

Editeur

L'éditeur permet de créer, de modifier, de compléter, de fusionner et d'insérer des programmes/textes/blocs de programme.

Editeur de texte

Voir → Editeur

Effacement général

Lors de l'effacement général, les mémoires suivantes de la → CPU sont effacées :

- la → mémoire de travail,
- la zone de lecture/d'écriture de la → mémoire de chargement,
- la → mémoire système,
- la → mémoire de sauvegarde.

Entraînement

L'entraînement est l'unité de la CN, qui exécute la régulation de vitesse de rotation et de couple en fonction des consignes de la CN.

Entrées/sorties TOR rapides

Via les entrées TOR, il est possible de lancer par exemple des routines de programme CNC rapides (routines d'interruption). Les sorties TOR de la CNC permettent de déclencher des fonctions de commutation rapides pilotées par le programme.

Fin de course logiciel

Les fins de course logiciels limitent la plage de déplacement d'un axe et empêchent l'arrêt brutal du chariot sur l'interrupteur de fin de course matériel. Pour chaque axe, 2 paires de valeurs peuvent être spécifiées et activées individuellement via l'→ AP.

Fonction miroir

La fonction miroir inverse les signes des coordonnées d'un contour pour un axe. La fonction miroir peut être appliquée simultanément à plusieurs axes.

Fonctions auxiliaires

Les fonctions auxiliaires permettent de transférer, dans les → programmes pièces, des → paramètres à un → AP, qui, à ce niveau, déclenchent des réactions définies par le constructeur de machines.

Fonctions de sécurité

La commande dispose de fonctions de surveillance actives en permanence, capables de détecter les défaillances dans la → CNC, la commande d'adaptation (→ AP) et la machine suffisamment tôt, de telle sorte que les dommages peuvent être en grande partie évités sur la pièce, l'outil ou la machine. En cas de défaillance, le déroulement de l'usinage est interrompu et les entraînements sont arrêtés, la cause de la défaillance sauvegardée en mémoire et affichée sous forme d'alarme. Dans le même temps, la présence d'une alarme CNC est signalée à l'AP.

Frame

Un frame est une règle opératoire qui transpose un système de coordonnées cartésiennes en un autre système de coordonnées cartésiennes. Le frame contient les composants → décalage d'origine, → rotation, → mise à l'échelle, → fonction miroir.

Frames programmables

Avec les → frames programmables, il est possible de définir en dynamique, au cours du traitement du programme pièce, de nouveaux points d'origine du système de coordonnées.

Une distinction est opérée entre la définition absolue, au moyen d'un nouveau frame et la définition additive, par référence à un point d'origine existant.

Géométrie

Description d'une → pièce dans le → système de coordonnées pièce.

Gestion du programme pièce

La gestion des programmes pièce peut être organisée d'après les → pièces. La taille de la mémoire de travail détermine le nombre des programmes et données à gérer. Chaque fichier (programmes et données) peut se voir attribuer un nom ne comportant pas plus de 24 caractères alphanumériques.

Groupe à mode de fonctionnement commun

Des axes et des broches de même appartenance technologique peuvent être regroupés en un groupe à mode de fonctionnement commun (GMFC). Les axes et les broches regroupés dans un GMFC peuvent être commandés par un ou plusieurs → canaux. Les canaux du GMFC se trouvent toujours dans le même → mode de fonctionnement.

HIGHSTEP

Synthèse des possibilités de programmation pour → l'AP du système AS300/AS400.

Homothétie

Elément d'un → frame réalisant des changements d’échelle spécifiques aux axes.

HW Config

Logiciel SIMATIC S7 pour la configuration et le paramétrage de composants matériels dans un projet S7.

Interface utilisateur

L'interface utilisateur (IU) permet la communication entre l'utilisateur et la commande CNC. Elle est constituée d'un écran qui comporte des touches logicielles horizontales et verticales.

Interpolateur

Unité logique de la → CN qui détermine, en fonction de la position de destination dans le programme pièce, des valeurs intermédiaires pour les déplacements à effectuer sur les différents axes.

Interpolation circulaire

→ L'outil doit se déplacer sur un cercle, entre des points définis du contour et avec une avance donnée, pour l'usinage de la pièce.

Interpolation de type spline

Avec l'interpolation de type spline, la commande peut créer une forme de courbe lisse à partir d'un petit nombre de points intermédiaires d'un contour de consigne.

Interpolation hélicoïdale

L'interpolation hélicoïdale est particulièrement adaptée à la fabrication simple de filetages intérieurs ou extérieurs avec des fraises de forme et pour le fraisage de rainures de graissage.

Dans ce cas, l'hélice se compose de deux déplacements :

- Déplacement circulaire dans un plan
- Déplacement linéaire perpendiculaire à ce plan

Interpolation linéaire

L'outil est déplacé sur une droite jusqu'au point cible et la pièce est alors usinée.

interpolation polynomiale

L'interpolation polynomiale permet de générer les formes de courbe les plus diverses, notamment **fonctions droite, parabole, puissance** (SINUMERIK 840D sl).

JOG

Mode de fonctionnement de la commande (mode de réglage) : Le mode de fonctionnement JOG permet de régler la machine. Des axes ou broches spécifiques peuvent être déplacés au moyen des touches de sens en mode manuel. Autres fonctionnalités du mode Jog : → Accostage du point de référence, → Repos et → Preset (Forcer la valeur réelle).

KV

Gain de boucle, grandeur de régulation d'une boucle de régulation.

Langage évolué CNC

Le language évolué sert à écrire des programmes CE, → Actions synchrones et → Cycles. Vous pouvez : → Structures de contrôle, → Variable définie par l'utilisateur, → Variable système, → Macroprogrammation.

Limitation de la zone de travail

Cette fonctionnalité permet de limiter la plage de déplacement des axes, en plus des fins de course. Pour chaque axe, il est possible de définir une paire de valeurs décrivant la zone de travail protégée.

Limitation programmable de la zone de travail

Restriction de la zone de déplacement de l'outil sur un espace défini par des limitations programmées.

Limite d'arrêt précis

Si tous les axes d'interpolation atteignent leur limite d'arrêt précis, la commande se comporte comme si elle avait atteint précisément son objectif. Cet état est suivi d'un changement de bloc du → programme pièce.

Macroprogrammation

Regroupement d'une certaine quantité d'instructions sous un descripteur. Le descripteur représente, au sein du programme, la quantité d'instructions regroupées.

Masse

La masse correspond à la totalité des parties inactives reliées entre elles sur un moyen d'exploitation et ne pouvant pas adopter une tension dangereuse par contact, même en cas d'anomalie.

MDA

Mode de fonctionnement de la commande : Manual Data Automatic. Dans le mode de fonctionnement MDA, des blocs de programme et des séquences de blocs peuvent être introduits et exécutés immédiatement avec la touche Départ programme (sans exécution d'un programme principal ou sous-programme).

Mémoire de chargement

Pour la CPU 314, la mémoire de chargement de → l'AP est équivalente à la → mémoire de travail.

Mémoire de correcteurs

Zone de données de la commande, dans laquelle des données de correction d'outil sont mémorisées.

Mémoire de programmes AP

SINUMERIK 840D sl : Dans la mémoire de travail AP, le programme AP utilisateur et les données utilisateur sont mémorisés ensemble avec le programme de base de l'AP.

mémoire de travail

La mémoire vive est une mémoire RAM se trouvant dans la → CPU et dans laquelle le processeur accède au programme utilisateur durant l'exécution du programme.

Mémoire système

La mémoire système est une mémoire de la CPU, dans laquelle les données ci-après sont sauvegardées :

- Données nécessaires au système d'exploitation
- Opérandes de temps, compteurs, mémentos

mémoire utilisateur

Tous les programmes et toutes les données, tels que les programmes pièce, les sous-programmes, les commentaires, les corrections d'outil, les décalages d'origine / frames ainsi que les données utilisateur de programme et de canal, peuvent être rangés dans une mémoire CNC utilisateur commune.

Messages

Tous les messages et → alarmes programmés dans le programme pièce et reconnus par le système sont affichés à l'écran du tableau de commande, en texte clair, avec leur horodatage et le symbole correspondant au critère d'effacement. Les alarmes sont affichées séparément des messages.

Mode de fonctionnement

Type de fonctionnement d'une commande SINUMERIK. Les modes de fonctionnement → Jog, → MDA, → Automatique sont définis.

Module de périphérie

Les modules de périphérie établissent la relation entre unité centrale et processus.

Les modules de périphérie sont :

- → les modules d'entrées/sorties TOR,
- → les modules d'entrées/sorties analogiques,
- → les modules de simulation.

Mot de données

Entité de données de deux octets faisant partie d'un → bloc de données.

Niveau de programme

Un programme pièce lancé dans le canal s'exécute en tant que → programme principal dans le niveau de programme 0 (niveau du programme principal). Chaque programme pièce appelé dans le programme principal s'exécute en tant que → sous-programme son propre niveau de programme 1 à n.

Nom d'axe

Pour une identification univoque, tous les axes de canal et → axes de machine de la commande doivent être désignés avec des noms univoques au niveau de canal ou de commande. Les → axes géométriques sont nommés X, Y, Z. Les → axes rotatifs tournant autour des axes géométriques sont nommés A, B, C.

NRK

Numeric Robotic Kernel (système d'exploitation de la → CN)

NURBS

Le pilotage des déplacements et l'interpolation de trajectoire dans la commande sont exécutés sur la base de NURBS (**N**on **U**niform **R**ational **B**-**S**plines). Ainsi, pour toutes les interpolations, il existe un déplacement unifié qui est interne à la commande.

OEM

Pour les constructeurs de machines qui veulent créer leur propre interface utilisateur ou introduire des fonctions spécifiques à une technologie dans la commande, des aménagements sont prévus pour des solutions individuelles (applications OEM).

Origine machine

Point fixe de la machine-outil, auquel se rapportent tous les systèmes de mesure (déduits).

Origine pièce

L'origine pièce est l'origine du → système de coordonnées pièce. Elle est définie par des distances par rapport à l'→ origine machine.

Outil

Objet monté sur la machine-outil dont la pointe ou le tranchant génère la forme à obtenir (par ex. outil de tournage, fraise, foret, faisceau LASER ...)

Paramètres R

Paramètres de calcul ; peuvent être renseignés ou interrogés dans le → programme pièce aux fins jugées utiles par le programmeur.

Pièce

Pièce à fabriquer/usiner sur la machine-outil.

Pièce brute

Pièce à l'état brut, avant le commencement de l'usinage.

Pile de sauvegarde

La pile de sauvegarde permet la protection du → programme utilisateur de la → CPU contre les coupures du réseau et la sauvegarde des zones de données, mémentos, horodatages et compteurs de façon rémanente.

Pilotage de la vitesse

Dans le cas de mouvements de déplacement de très faible amplitude par bloc, il est possible, pour pouvoir atteindre une vitesse acceptable, de faire appel à une évaluation anticipée sur plusieurs blocs (→ Look Ahead).

Plage de déplacement

La plage de déplacement maximale autorisée pour les axes linéaires recouvre ± 9 décades. La valeur absolue dépend de la définition et de la résolution sélectionnées du régulateur de position et du système d'unités (inch ou métrique).

Point de référence

Point de la machine-outil, auquel le système de mesure des → axes machine fait référence.

Point machine fixe

Point défini de façon univoque par la machine-outil, par exemple point de référence machine.

Pré-coïncidence

Changement de bloc déjà effectué si le trajet s'est approché de la position finale d'un delta prédéterminé.

Programmation AP

L'AP est programmé à l'aide du logiciel STEP 7. Le logiciel de programmation STEP 7 est basé sur le système d'exploitation WINDOWS et contient les fonctions de la programmation STEP 5, avec des perfectionnements innovateurs.

Programme de transfert des données PCIN

PCIN est un utilitaire pour l'émission et la réception de données utilisateur CNC via l'interface série, telles que programmes pièce, corrections d'outils, etc. Le programme PCIN est exécutable sous MSDOS sur PC industriel standard. Le programme PCIN tourne sur PC industriel standard sous MS-DOS.

Programme pièce

Suite d'instructions transmises à la commande numérique pour la fabrication d'une → pièce déterminée. Réalisation d'un usinage défini sur une → pièce brute.

Programme principal

La désignation programme principale date encore de l'époque où les programmes pièce étaient subdivisés de manière fixe en programmes principaux et → sous-programmes. Cette subdivision fixe n'a plus lieu d'être avec le langage CN SINUMERIK actuel. En principe, chaque programme pièce peut être sélectionné et lancé dans le canal. Il est ensuite exécuté dans le → niveau de programme 0 (niveau du programme principal). Dans le programme principal il est possible d'appeler d'autres programmes pièce ou → cycles en tant que sous-programmes.

Programme utilisateur

Les programmes utilisateur destinés aux systèmes d'automatisation S7-300 sont créés avec le logiciel de programmation STEP 7. De conception modulaire, ils sont constitués de différents blocs.

Les principaux types de blocs sont les suivants :

- Blocs de codes
  Ces blocs contiennent les instructions STEP 7.
- Blocs de données
  Ces blocs contiennent des constantes et des variables pour le programme STEP 7.

Recherche de bloc

La fonction "Recherche de blocs" permet de rechercher une partie quelconque du programme pièce où l'usinage doit être démarré ou repris (pour le test de programmes pièce ou après une interruption de l'usinage).

Redémarrage

Chargement du programme après une mise sous tension.

Relèvement rapide du contour

Lorsqu'une interruption est rencontrée, il est possible d'engager un mouvement au moyen du programme d'usinage CNC, qui rend possible un relèvement rapide de l'outil hors du contour de la pièce qui vient d'être usinée. En outre, l'angle de retrait et la distance de retrait peuvent

être programmés. Un relèvement rapide peut être suivi de l'exécution d'une routine d'interruption supplémentaire.

Réseau

Un réseau est une association de plusieurs S7-300 et d'autres terminaux, par exemple une console de programmation, reliés entre eux au moyen d'un → câble de liaison. Par l'intermédiaire du réseau, il y a un échange de données entre les appareils connectés.

Rotation

Elément d'un → frame définissant la rotation d'un système de coordonnées d'un angle défini.

Routine d'interruption

Les routines d'interruption sont des → sous-programmes spécifiques qui peuvent être lancées par les événements (signaux externes) du processus de traitement. Lorsqu'un bloc de programme pièce est interrompu en cours de traitement, la position d'interruption des axes est automatiquement enregistrée.

RT

Rapport de transmission

Sous-programme

La désignation sous-programme date encore de l'époque où les programmes pièce étaient subdivisés de manière fixe en → programmes principaux et sous-programmes. Cette subdivision fixe n'a plus lieu d'être avec le langage CN SINUMERIK actuel. En principe, tout programme pièce ou tout → cycle peut être appelé en tant que sous-programme au sein d'un autre programme pièce. Il est alors exécuté dans le → niveau de programme (x+1) (niveau de sous-programme (x + 1)).

Sous-programme asynchrone

Programme pièce qui peut être démarré de façon asynchrone (indépendamment de l'état du programme actuel) par un signal d'interruption (par ex. signal "Entrée CN rapide").

Spline C

Le spline C est le spline le plus connu et le plus utilisé. Les raccordements au niveau des points intermédiaires sont caractérisés par la continuité de la tangente et du rayon de courbure. Des polynômes de 3me degré sont utilisés.

Surveillance du contour

Pour évaluer la précision du contour, on utilise l'écart de traînage à l'intérieur d'une plage de tolérance définie. Un écart de traînage excessivement élevé peut s'expliquer, par exemple,

par une surcharge du système d'entraînement. Dans ce cas, une alarme se déclenche et les axes sont immobilisés.

Synchronisation

Instructions figurant dans les → programmes pièce pour la coordination des séquences dans les différents → canaux, à des emplacements d'usinage définis.

Système anglo-saxon

Système de mesure dans lequel les distances sont exprimées en "inch" et en fractions d'inch.

Système de coordonnées

Voir → Système de coordonnées machine, → Système de coordonnées pièce

Système de coordonnées de base

Système de coordonnées cartésiennes qui découle du système de coordonnées machine par une transformation.

Dans le → programme pièce, le programmeur utilise les noms des axes du système de coordonnées de base. Si aucune → transformation n'est active, le système de coordonnées de base est parallèle au → système de coordonnées machine. Les deux systèmes ne diffèrent que par les → noms des axes.

Système de coordonnées machine

Système de coordonnées référencé aux axes de la machine-outil.

Système de coordonnées pièce

L'origine du système de coordonnées pièce est l'→ origine pièce. Dans les programmes écrits dans un système de coordonnées pièce, les cotes et les sens sont définis par rapport à ce système.

Système de mesure en métrique et en inch

Dans le programme d'usinage, les valeurs de position et valeurs de pas peuvent être programmées en inch. Indépendamment du système de mesure programmable (`G70`/`G71`) la commande est réglée sur un système de base.

Système de mesure métrique

Système normalisé d'unités : pour les longueurs, par exemple mm (millimètre), m (mètre).

Table de compensation

Table des points intermédiaires. Elle indique, pour des positions sélectionnées de l'axe de base, les valeurs de compensation de l'axe de compensation.

Tableau de commande machine

Tableau de commande de la machine-outil comportant des organes de commande tels que touches, commutateurs rotatifs, etc. et des éléments d'affichage simples tels que les LED. Il permet d'exercer une influence indirecte sur la machine-outil par l'intermédiaire de l'AP.

Taraudage sans porte-taraud compensateur

Cette fonction permet de percer des filetages sans porte-taraud compensateur. A travers le processus d'interpolation de la broche comme un axe circulaire et axe de perçage, les filetages sont découpés précisément à la profondeur finale, par exemple filetages à trou borgne (condition préalable : utilisation de l'axe de la broche).

Touche logicielle

Touche dont le libellé est représenté par un champ de l'écran, qui s'adapte dynamiquement à la situation de commande courante. Les touches de fonction librement affectables (touches logicielles) sont assignées par voie logicielle à des fonctions définies.

Transformation

Décalage d'origine additif ou absolu d'un axe.

Unité TOA

Chaque → zone TOA peut contenir plusieurs unités TOA. Le nombre des unités TOA possibles est limité via le nombre maximal de → canaux actifs. Une unité TOA comprend un seul bloc de données d'outils et un seul bloc de données de magazine. Un bloc de données de porte-outil peut également être contenu (facultatif).

Usinage de surfaces obliques

Les opérations d'alésage et de fraisage portant sur des surfaces de pièce qui ne se trouvent pas dans les plans de coordonnées de la machine peuvent être exécutées confortablement avec l'aide de la fonction Usinage de surfaces obliques.

V.24

Interface série pour la saisie/sortie de données. Cette interface permet de charger et de sauvegarder des programmes d'usinage ainsi que des données constructeur et utilisateur.

Valeur de compensation

Différence entre la position d'axe mesuré par l'indicateur de mesure et la position d'axe programmée et souhaitée.

Variable définie par l'utilisateur

L'utilisateur peut, à des fins quelconques, déclarer des variables personnalisées dans le → programme pièce ou dans un bloc de données (données utilisateur globales). Une déclaration de variable contient une indication de type de données et le nom de la variable. Voir → Variable système.

Variable système

Variable existant indépendamment de l'intervention du programmeur d'un → programme pièce. Elle est définie par un type de données et par un nom de variable commençant par **$**. Voir → Variable définie par l'utilisateur.

Vitesse limite mécanique

Vitesse de rotation maximale/minimale (de la broche) : A travers la détermination des données machine de → l'AP ou des → données de réglage, il est possible de limiter la vitesse de rotation maximale d'une broche.

Vitesse rapide

Vitesse de déplacement la plus rapide d’un axe. Le rapide est utilisé par exemple pour approcher l’outil du → contour de la pièce à partir d’une position de repos ou pour dégager l’outil du contour de la pièce. La vitesse rapide se règle dans un paramètre machine. Le réglage est spécifique à la machine.

Vitesse tangentielle

La vitesse tangentielle maximale programmable dépend de la résolution d'introduction (définition). Dans le cas d'une résolution d'introduction de 0,1 mm par exemple, la vitesse tangentielle maximale programmable vaut 1000 m/min.

WinSCP

WinSCP est un programme Open Source pour Windows, qui est disponible gratuitement et qui est utilisé pour le transfert de fichiers.

Zone de protection

Espace tridimensionnel à l'intérieur de l'→ espace de travail dans lequel la pointe de l'outil ne doit pas pénétrer.

Zone de travail

Zone tridimensionnelle dans laquelle la pointe d'outil peut se trouver, compte tenu de la structure de la machine-outil. Voir → Volume de protection.

Zone TOA

La zone TOA comprend toutes les données d'outils et de magasins. Par défaut, la zone relative à la portée des données coïncide avec la zone du → canal. Par contre, des paramètres machine permettent de déterminer que plusieurs canaux partagent une → unité TOA de sorte que des données de gestion d'outil communes soient ensuite disponibles pour ces canaux.

Index

$

$AA_ATOL, 577
$AA_COUP_ACT
  Couplage de deux axes par valeur pilote, 608
  Lors de déplacements conjugués, 584
$AA_ESR_ENABLE, 708
$AA_LEAD_SP, 608
$AA_LEAD_SV, 608
$AC_ACT_PROG_NET_TIME, 695
$AC_ACTUAL_PARTS, 698
$AC_AXCTSWA, 680
$AC_AXCTSWE, 680
$AC_CTOL, 577
$AC_CUT_INV, 497
$AC_CUTMOD, 496
$AC_CUTMOD_ANG, 496
$AC_CUTMODK, 496
$AC_CUTTING_TIME, 695
$AC_CYCLE_TIME, 695
$AC_DELAYFST, 557
$AC_ESR_TRIGGER, 708
$AC_OLD_PROG_NET_TIME, 695
$AC_OLD_PROG_NET_TIME_COUNT, 695
$AC_OPERATING_TIME, 695
$AC_OTOL, 577
$AC_PROG_NET_TIME_TRIGGER, 696
$AC_REPOS_PATH_MODE, 567
$AC_REQUIRED_PARTS, 698
$AC_SMAXVELO, 572
$AC_SMAXVELO_INFO, 572
$AC_SPECIAL_PARTS, 698
$AC_TOTAL_PARTS, 698
$AC_TRAFO_CORR_ELEM_P, 426
$AC_TRAFO_CORR_ELEM_T, 426
$AC_TRAFO_ORIAX_LOC, 426
$AN_AXCTAS, 680
$AN_AXCTSWA, 680
$AN_ESR_TRIGGER, 708
$AN_LANGUAGE_ON_HMI, 968
$AN_POWERON_TIME, 694
$AN_SETUP_TIME, 694
$NT_CLOSE_CHAIN_T, 426
$NT_CNTRL, 426
$NT_CORR_ELEM_P, 425
$NT_CORR_ELEM_T, 425
$NT_NAME, 417
$NT_ROT_AX_NAME, 492
$NT_TRAFO_INDEX, 417
$P_ACTBFRAME, 328
$P_AD, 496
$P_BFRAME, 328
$P_CHBFRAME, 327
$P_CHBFRMASK, 328
$P_CTOL, 577
$P_CUT_INV, 497
$P_CUTMOD, 496
$P_CUTMOD_ANG, 496
$P_CUTMOD_ERR, 497
$P_CUTMODK, 496
$P_DELAYFST, 557
$P_IFRAME, 329
$P_IS_EES_PATH, 234
$P_NCBFRAME, 327
$P_NCBFRMASK, 328
$P_ORI_DIFF, 487
$P_ORI_POS, 487
$P_ORI_SOL, 488
$P_ORI_STAT, 491
$P_OTOL, 577
$P_PATH, 233
$P_PFRAME, 329
$P_PROG, 233
$P_PROGPATH, 233
$P_SIM, 296
$P_STACK, 233
$P_SUBPAR, 171
$P_TOOLENV, 507
$P_TOOLENVN, 506
$PA_ATOL, 577
$SA_LEAD_TYPE, 608
$SC_CONTPREC, 550
$SC_MINFEED, 550
$SC_PA_ACTIV_IMMED, 247
$SN_PA_ACTIV_IMMED, 247
$TC_CARR1...14, 472
$TC_CARR18...23, 472
$TC_CARR18[m], 476
$TC_DP1 ... 25, 427
$TC_ECPxy, 431
$TC_SCPxy, 431

*

* (fonction de calcul), 81

/

/ (fonction de calcul), 81

+

+ (fonction de calcul), 81

<

< (opérateur relationnel), 83
<< (opérateur de concaténation), 93
<= (opérateur relationnel), 83
<> (opérateur relationnel), 83

=

== (opérateur relationnel), 83

>

> (opérateur relationnel), 83
>= (opérateur relationnel), 83

A

ABS, 81
ACCLIMA, 545
ACOS, 81
A-coup
    Correction, 570
À-coup
    Limitation, 543
ACTBLOCNO, 183
ACTFRAME, 303
ADISPOSA, 298
Adressage, 227
Adresses OEM, 296
Affichage de bloc
    Inhibition, 183
Alarmes
    Définir dans le programme CN, 705
Alarmes de cycle, 705
Alésage à l'alésoir - CYCLE85
    Programmation externe, 791
Alésage au tour - CYCLE86
    Programmation externe, 792
ALF
    Pour relèvement rapide du contour, 142
AND, 83
Angles d'Euler, 350
Angles RPY, 350
Appel de contour - CYCLE62
    Programmation externe, 761
Appel de sous-programme avec chemin d'accès et paramètres, 209
APR, 49
APRB, 49
APRP, 49
APW, 49
APWB, 49
APWP, 49
Arrêt
    Assisté par CN, 712
    Contrôlé par le variateur, 713
Arrondir au chiffre supérieur, 164
AS, 220
ASIN, 81
ASPLINE, 260
ATAN2, 81
ATOL, 574
Attributs de position
    Programmation indirecte, 75
Autres conditions à prendre en compte lors des transformations, 400
AV, 617
avec TCOFRX, 477
avec TCOFRY, 477
AX, 671
AXCTSWE, 679
AXCTSWEC, 679
AXCTSWED, 679
Axe
    Changement, 146
Axe asservi
    Couplage de deux axes par valeur pilote, 604
Axe géométrique
    Basculer, 674
Axe pilote
    Couplage de deux axes par valeur pilote, 604
Axes
    Conjugué, 583
Axes conjugués, 583
Axes d'orientation, 356
Axes géométriques permutables, 674
Axes rotatifs
    Angle rotatif, 472
    Vecteurs de direction, 472
    Vecteurs distances, 472

AXIS, 31
AXNAME, 92
AXSTRING, 671
AXTOCHAN, 151
AXTOSPI, 671

B

B_AND, 83
B_NOT, 83
B_OR, 83
B_XOR, 83
BAUTO, 260
BFRAME, 303
BLOC, 207
Bloc d'arrêt, 559
Bloc par bloc
    Inhibition, 178
BLSYNC, 136
BNAT, 260
BOOL, 31
Boucle de comptage, 125
Boucle de programme
    Boucle, 124
    Boucle de comptage, 125
    Boucle IF, 123
    Boucle REPEAT, 127
    Boucle WHILE, 126
Boucle sans fin, 124
BOUND, 87
BRISK, 543
BRISKA, 543
Broche
    Changement, 146
Broche synchrone
    Couplage, 617
    Définir la paire, 623
BSPLINE, 260
BTAN, 260

C

CAC, 259
CACN, 259
CACP, 259
CALL, 206
CALLPATH, 210
Caractère 0, 91
CASE, 113
CDC, 259
Centrage - CYCLE81
    Programmation externe, 782
CFINE, 313
Chaîne de caractères
    Concaténation, 93
    Formater, 99
    Longueur, 95
    Opérations, 91
Chaîne outil, 424
Chaîne pièce, 424
CHAN, 31
CHANDATA, 235
CHAR, 31
Chariotage
    Fonctions auxiliaires, 721
Chariotage - CYCLE951
    Programmation externe, 837
Chariotage au contour - CYCLE95
    Programmation externe, 795
Chemin d'accès, 228
Chemin de recherche
    Appel de sous-programme, 232
    Chemin de recherche programmable, 210
Chemin de répertoire, 230
CHKDNO, 468
CIC, 259
Cinématique
    Décomposée, 476
Classe de données, 54
CLEARM, 129
CLRINT, 139
COARSE, 617
COARSEA, 298
Code G
    Programmation indirecte, 74
Coefficient du polynôme, 275
COLLPAIR, 410
COMPCAD, 273
COMPCURV, 273
COMPLETE, 235
COMPOF, 273
COMPON, 273
Composante de frame
    FI, 309
    MI, 309
    RT, 309
    SC, 309
    TR, 309
COMPSURF, 273
Concaténation
    de chaînes de caractères, 93

Concaténation de filetages - CYCLE98
Programmation externe, 797
CONTDCON, 728
Contour
Code, 728
Préparation, 722
Réaccostage, 561
Table de, 722
CONTPRON, 722
Contrôle
Structures, 121
Correction d'outil
Mémoire de correcteurs, 427
Système de coordonnées pour valeurs d'usure, 437
Correction du rayon d'outil
décélération aux angles, 297
Correction en ligne de la longueur d'outil, 480
Corrections d'outil
Additives, 430
CORRTRAFO, 419
COS, 81
COUPDEF, 617
COUPDEL, 617
Couplage
Générique, 628
Couplage de deux axes par valeur pilote, 604
Couplage de vitesse, 620
Couplage par valeur de consigne, 620
Couplage par valeur pilote
Couplage par valeur réelle et couplage par valeur de consigne, 607
Synchronisation de l'axe pilote et de l'axe asservi, 606
Couplage par valeur réelle, 620
COUPOF, 617
COUPOFS, 617
COUPON, 617
COUPONC, 617
COUPRES, 617
CP, 389
CPBC, 629
CPDEF, 628
CPDEL, 628
CPFMOF, 631
CPFMON, 631
CPFMSON, 630
CPFPOS + CPOF, 631
CPFPOS + CPON, 629
CPFRS, 629
CPLA, 628
CPLCTID, 629
CPLDEF, 628
CPLDEL, 628
CPLDEN, 629
CPLINSC, 633
CPLINTR, 633
CPLNUM, 629
CPLOF, 629
CPLON, 629
CPLOUTSC, 633
CPLOUTTR, 633
CPLPOS, 629
CPLSETVAL, 629
CPMALARM, 634
CPMBRAKE, 634
CPMPRT, 633
CPMRESET, 632
CPMSTART, 633
CPMVDI, 634
CPOF, 629
CPON, 629
CPRECOF, 550
CPRECON, 550
CPROT, 244
CPROTDEF, 239
CPSETTYPE, 634
CPSYNCOP, 633
CPSYNCOP2, 634
CPSYNCOV, 634
CPSYNFIP, 633
CPSYNFIP2, 634
CPSYNFIV, 634
Critère de fin de déplacement
Programmable, 298
CSPLINE, 260
CTAB, 597
CTABDEF, 586
CTABDEL, 593
CTABEND, 586
CTABEXISTS, 593
CTABFNO, 602
CTABFPOL, 602
CTABFSEG, 602
CTABID, 596
CTABINV, 597
CTABISLOCK, 596
CTABLOCK, 594
CTABMEMTYP, 596
CTABMPOL, 602
CTABMSEG, 602
CTABNO, 602
CTABNOMEM, 602
CTABPERIOD, 596

CTABPOL, 602
CTABPOLID, 602
CTABSEG, 602
CTABSEGID, 602
CTABSEV, 597
CTABSSV, 597
CTABTEP, 597
CTABTEV, 597
CTABTMAX, 597
CTABTMIN, 597
CTABTSP, 597
CTABTSV, 597
CTABUNLOCK, 594
CTOL, 574
CTRANS, 313
CUT3DC, 447
CUT3DCC, 457
CUT3DCCD, 457
CUT3DCD, 447
CUT3DF, 451
CUT3DFD, 451
CUT3DFF, 451
CUT3DFS, 451
CUTMOD, 493
CUTMODK, 493
CYCLE4071
    Programmation externe, 846
CYCLE4072
    Programmation externe, 847
CYCLE4073
    Programmation externe, 851
CYCLE4074
    Programmation externe, 852
CYCLE4075
    Programmation externe, 855
CYCLE4077
    Programmation externe, 858
CYCLE4078
    Programmation externe, 861
CYCLE4079
    Programmation externe, 863
CYCLE435 - Définir le système de coordonnées du dresseur
    Programmation externe, 806
CYCLE495 - Profilage
    Programmation externe, 806
CYCLE60 - Gravure
    Programmation externe, 757
CYCLE61 - Surfaçage
    Programmation externe, 759
CYCLE62 - Appel de contour
    Programmation externe, 761
CYCLE63 - Fraisage de poche de contour
    Programmation externe, 762
CYCLE64 - Perçage ébauche d'une poche de contour
    Programmation externe, 765
CYCLE70 - Fraisage de filetage
    Programmation externe, 767
CYCLE72 - Fraisage en contournage
    Programmation externe, 768
CYCLE76 - Tourillon rectangulaire
    Programmation externe, 773
CYCLE77 - Tourillon circulaire
    Programmation externe, 775
CYCLE78 - Fraisage de filetages
    Programmation externe, 777
CYCLE79 - Polyèdre
    Programmation externe, 780
CYCLE800 - Pivotement
    Programmation externe, 808
CYCLE801 - Modèle de positions Réseau/Cadre
    Programmation externe, 812
CYCLE802 - Positions quelconques
    Programmation externe, 813
CYCLE81 - Centrage
    Programmation externe, 782
CYCLE82 - Perçage
    Programmation externe, 783
CYCLE83 - Perçage profond
    Programmation externe, 785
CYCLE830 - Perçage profond 2
    Programmation externe, 816
CYCLE832 - Réglages High Speed
    Programmation externe, 822
CYCLE84 - Taraudage sans porte-taraud compensateur
    Programmation externe, 788
CYCLE840 - Taraudage avec porte-taraud compensateur
    Programmation externe, 826
CYCLE85 - Alésage à l'alésoir
    Programmation externe, 791
CYCLE86 - Alésage au tour
    Programmation externe, 792
CYCLE899 - Fraisage de rainure borgne
    Programmation externe, 828
CYCLE92 - Tronçonnage
    Programmation externe, 794
CYCLE930 - Gorge
    Programmation externe, 831
CYCLE940 - Dégagement
    Programmation externe, 834
CYCLE95 - Chariotage au contour
    Programmation externe, 795

CYCLE951 - Chariotage
Programmation externe, 837
CYCLE952 - Plongée de contour
Programmation externe, 840
CYCLE98 - Concaténation de filetages
Programmation externe, 797
CYCLE99 - Filetage au tour
Programmation externe, 801

D

DCI, 54
DCM, 54
DCU, 54
Début du bloc de programme - GROUP_BEGIN, 865
Décalage d'origine
Externe, 315
Décalage d'origine externe, 315
Décélération à tous les angles, 297
Décélération aux angles intérieurs, 297
DEF, 31
DEFAULT, 113
DEFINE ... AS, 220
Définir le système de coordonnées du dresseur - CYCLE435
Programmation externe, 806
Définition de la pièce brute, 715
Dégagement - CYCLE940
Programmation externe, 834
DELAYFSTOF, 555
DELAYFSTON, 555
DELDL, 432
DELETE, 158
DELOBJ, 403
DELTOOLENV, 504
Déplacement point à point, 389
Déplacement PTP cartésien, 389
Déplacements conjugués, 581
DISABLE, 138
DISPLOF, 183
DISPLON, 183
Disponibilité
en fonction du système, 5
DISPR, 561
DIV, 81
DL, 430
DO, 641
Données de cercles
Calcul, 735
DRIVE, 543
DRIVEA, 543
DV, 617
DYNFINISH, 547
DYNNORM, 547
DYNPOS, 547
DYNROUGH, 547
DYNSEMIFIN, 547

E

Easy XML, 689
EAUTO, 260
EES, 225
Effacement de la distance restant à parcourir, 291
EG
réducteur électronique, 610
EGDEF, 610
EGDEL, 615
EGOFC, 614
EGOFS, 614
EGON, 611
EGONSYN, 611
EGONSYNE, 611
Elément de contour
Exécuter, 734
ELSE, 123
ENABLE, 138
ENAT, 260
ENDFOR, 125
ENDIF, 123
ENDLABEL, 115
ENDLOOP, 124
ENDWHILE, 126
ESR, 707
ESRR, 713
ESRS, 713
ETAN, 260
Etat de la tâche de mesure, 294
Etat du couplage
Couplage de deux axes par valeur pilote, 608
Lors de déplacements conjugués, 584
Etiquette, 115
Evénement déclencheur
Pour la mesure, 290
EVERY, 641
EXECSTRING, 78
EXECTAB, 734
EXECUTE, 737
Exécution
Comportement de structures de contrôle, 122
EXP, 81
EXTCALL
pour SINUMERIK 828D, 215
pour SINUMERIK 840D sl, 211

EXTCLOSE, 700
EXTERN, 201
EXTOPEN, 700

F

facteur de couplage, 581
FALSE, 31
FCTDEF, 442
FCUB, 538
FENDNORM, 297
FFWOF, 549
FFWON, 549
Fichier
    Informations, 162
FIFOCTRL, 552
FILEDATE, 162
FILEINFO, 162
FILESIZE, 162
FILESTAT, 162
Filetage au tour - CYCLE99
    Programmation externe, 801
FILETIME, 162
Fin du bloc de programme - GROUP_END
    Programmation externe, 866
Fin du supplément de mise au point - GROUP_ADDEND
    Programmation externe, 866
FINE, 617
FINEA, 298
FLIN, 538
FNORM, 538
Fonctions OEM, 296
FOR, 125
FPO, 538
Fraisage de filetage - CYCLE70
    Programmation externe, 767
Fraisage de filetages - CYCLE78
    Programmation externe, 777
Fraisage de poche de contour – CYCLE63
    Programmation externe, 762
Fraisage de rainure borgne - CYCLE899
    Programmation externe, 828
Fraisage en bout, 353
Fraisage en contournage - CYCLE72
    Programmation externe, 768
FRAME, 31
    appel,
    Concaténation,
Frames
    Attribution, 311
    Concaténation de frames, 312
    Globales pour NCU, 325
    Spécifiques à un canal, 326
    Système, 327
Frames système, 327
FROM, 641
FTOCOF, 445
FTOCON, 445

G

G290, 719
G291, 719
G5, 385
G62, 297
G621, 297
G7, 385
G810 ... G819, 296
G820 ... G829, 296
GEOAX, 674
GET, 146
GETACTTD, 470
GETD, 146
GETDNO, 469
GETTCOR, 507
GETTENV, 505
GETVARAP, 66
GETVARDFT, 68
GETVARDIM, 68
GETVARLIM, 67
GETVARPHU, 65
GETVARTYP, 70
Gorge - CYCLE930
    Programmation externe, 831
GOTO, 110
GOTOB, 110
GOTOC, 110
GOTOF, 110
GOTOS, 109
GP, 75
Gravure - CYCLE60
    Programmation externe, 757
Grignotage, 657
GROUP_ADDEND - Fin du supplément de mise au point
    Programmation externe, 866
GROUP_BEGIN - Début du bloc de programme
    Programmation externe, 865
GROUP_END - Fin du bloc de programme
    Programmation externe, 866
Groupe d'axes à déplacements conjugués, 581
Groupe G
    Technologie, 547

Groupe spline, 271
GUD, 32

H

High Speed Settings - CYCLE832
    Programmation externe, 822
HOLES1 - Modèle de positions Ligne
    Programmation externe, 742
HOLES2 - Modèle de positions Cercle
    Programmation externe, 742

I

ID, 641
IDS, 641
IF, 123
IFRAME, 303
INDEX, 96
Indice de tableau, 59
INICF, 31
INIPO, 31
INIRE, 31
INIT, 129
INITIAL, 235
INITIAL_INI, 235
Initialisation
    de tableaux, 57
insensible à la casse, 227
INT, 31
Interpolation du vecteur de rotation, 366
Interpolation polynomiale, 274
INTERSEC, 732
IPOBRKA, 298
IPOENDA, 298
IPOSTOP, 617
IPTRLOCK, 558
IPTRUNLOCK, 558
ISAXIS, 671
ISFILE, 161
ISNUMBER, 92
ISOCALL, 209
ISVAR, 63

J

JERKLIM, 570
JERKLIMA, 545

L

L..., 199
LEAD, 347
LEADOF, 604
LEADON, 604
LENTOAX, 528
LIFTFAST, 140
Link
    Variables, 29
Lissage
    Du tracé d'orientation, 379
LLI, 45
LN, 81
LONGHOLE - Trou oblong
    Programmation externe, 755
LOOP, 124
LUD, 32

M

M17, 187
M30, 187
Macro, 220
Marque de saut
    Répétitions de sections de programme, 115
    Sauts de programme, 111
MASLDEF, 636
MASLDEL, 636
MASLOF, 636
MASLOFS, 636
MASLON, 636
MATCH, 96
MAXVAL, 87
MCALL, 204
MD10010, 129
MD10280, 129
MD24658, 514
MEAC, 285
MEAFRAME, 321
MEAS, 282
MEASA, 285
MEAW, 282
MEAWA, 285
Mémoire
    Programme, 223
    Tampon, 552
    Travail, 235
Mémoire de correcteurs, 427

Mémoire de programmes
Répertoires standard, 224
Types de fichiers, 224
Mémoire de travail, 235
Mémoire globale de programmes pièce (GDIR), 226
MINDEX, 96
MINVAL, 87
MMC, 689
MOD, 81
MODAXVAL, 671
Mode d'accélération, 543
Mode de fonctionnement
Pour la mesure, 291
Mode de langage, 719
Modèle de position Cercle - HOLES2
Programmation externe, 742
Modèle de position Ligne - HOLES1
Programmation externe, 742
Modèle de position Réseau/Cadre - CYCLE801
Programmation externe, 812
MPF, 224

N

NAMETOINT, 407
NCK, 31
NEWCONF, 153
Niveaux d'imbrication
de structures de contrôle, 121
NOC, 617
Nom de fichier, 231
Nom de sous-programme DIN, 231
Nom du lecteur, 229
NOT, 83
Notation EES, 227
Notation NCK, 227
NPROT, 244
NPROTDEF, 239
NUMBER, 92
Numéro D
attribuer librement, 468
Numéro de tranchant, 468
Numéros D
contrôler, 468
renommer, 469
NUT, 358

O

OEMIPO1/2, 296
OMA1 ... OMA5, 296
Opérateurs logiques, 83
Opérateurs relationnels, 83
OR, 83
Organe porte-outil
cinématique, 472
Orientable, 477
Organes porte-outil orientables, 472
ORIAXES, 356
ORIC, 462
ORICONCCW, 358
ORICONCW, 358
ORICONIO, 358
ORICONTO, 358
ORICURVE, 362
ORID, 462
Orientation de l'outil
Par rapport à la trajectoire, 369
Orientation de l'outil par rapport à la trajectoire, 369
ORIEULER, 356
ORIMKS, 354
ORIPATH, 370
ORIPATHS, 370
ORIPLANE, 358
ORIRESET(A, B, C), 345
ORIROTA, 366
ORIROTC
Interpolation de la rotation de l'outil, 372
Rotation de l'orientation de l'outil, 366
ORIROTR, 366
ORIROTT, 366
ORIRPY, 356
ORIRPY2, 356
ORIS, 462
ORISOF, 379
ORISOLH, 483
ORISON, 379
ORIVECT, 356
ORIVIRT1, 356
ORIVIRT2, 356
ORIWKS, 354
OS, 643
OSB, 643
OSC, 462
OSCILL, 648
Oscillation
Asynchrone, 643
Oscillation asynchrone, 643
Oscillation synchrone, 648
Pénétration au point d'inversion de sens, 653
Pénétration partielle, 651
pilotée par des actions synchrones, 648

Point d'inversion, 651
Zone d'inversion, 651
Oscillation asynchrone, 643
Oscillation synchrone
Actions synchrones, 652
affectation des axes d'oscillation et de pénétration, 651
définir les pénétrations, 651
Exploitation Période d'appel de l'interpolateur, 654
Mouvement de pénétration, 652
Pénétration dans la zone d'inversion de sens, 652
Pénétration partielle suivante, 654
OSCTRL, 643
OSD, 462
OSE, 643
OSNSC, 643
OSOF, 462
OSP1, 643
OSP2, 643
OSS, 462
OSSE, 462
OST, 462
OST1, 643
OST2, 643
OTOL, 574
Outil
Orientation, 462
Orientation en cas de changement de frame, 479
Paramètre, 427

P

P..., 203
P_ACTFRAME, 329
Paramètre
Changement, 427
Paramètre de calcul
Spécifiques à un canal, 26
Paramètres
Effectifs, 169
Formels, 169
Transfert à l'appel d'un sous-programme, 169, 201
Paramètres de calcul
Globaux, 27
PCALL, 209
PDELAYOF, 657
PDELAYON, 657
Perçage - CYCLE82
Programmation externe, 783
Perçage ébauche d'une poche de contour - CYCLE64
Programmation externe, 765
Perçage profond - CYCLE83
Programmation externe, 785
Perçage profond 2 - CYCLE830
Programmation externe, 816
PFRAME, 303
PHI
Orientation le long d'une enveloppe de cône, 358
Polynômes d'orientation, 365
PHU, 46
Pièce
Catalogue, 224
Compteurs, 698
Répertoires, 224
Pivotement - CYCLE800
Programmation externe, 808
PL
Interpolation de type spline, 260
Interpolation polynomiale, 274
Plongée de contour - CYCLE952
Programmation externe, 840
PM15800, 28
PM18104, 503
PM18116, 504
PM18156, 28
PM20360, 512
PM24558, 514
PO, 274
PO[PHI]
Orientation le long d'une enveloppe de cône, 358
Polynômes d'orientation, 364
Rotation de l'orientation de l'outil, 370
PO[PSI]
Orientation le long d'une enveloppe de cône, 358
Polynômes d'orientation, 364
Rotation de l'orientation de l'outil, 370
PO[THT]
Polynômes d'orientation, 364
Rotation de l'orientation de l'outil, 370
PO[XH]
Orientation avec indication de deux points de contact, 362
Polynômes d'orientation, 365
PO[YH]
Orientation avec indication de deux points de contact, 362
Polynômes d'orientation, 365
PO[ZH]
Orientation avec indication de deux points de contact, 362
Polynômes d'orientation, 365

Poche circulaire - POCKET4
Programmation externe, 747
Poche rectangulaire - POCKET3
Programmation externe, 744
POCKET3 - Poche rectangulaire
Programmation externe, 744
POCKET4 - Poche circulaire
Programmation externe, 747
Poinçonnage, 657
Point auxiliaire de la fraise, 450
Point d'usinage de la fraise, 450
Pointe de la fraise, 450
Pointeur d'interruption automatique, 560
POLF
Retrait assisté par CN, 708
POLFA, 708
POLFMASK
Retrait assisté par CN, 708
POLFMLIN
Retrait assisté par CN, 708
POLY, 274
Polyèdre - CYCLE79
Programmation externe, 780
polynôme dénominateur, 278
POLYPATH, 274
PON, 666
PONS, 657
POSBA, 617
Position initiale de l'orientation de l'outil
ORIRESET, 346
Positions quelconques - CYCLE802
Programmation externe, 813
Positions singulières, 355
POT, 81
Précision de contour
Programmable, 550
Préparation du contour
Signalisation d'erreur, 737
PREPRO, 186
PRESETON, 317
PRESETONS, 319
PRIO, 136
PRLOC, 31
Process DataShare, 700
Profilage - CYCLE495
Programmation externe, 806
Profondeur de pénétration, 450
Programmation de l'orientation, 356
Programmation externe, 865
Programmation indirecte
Attributs de position, 75
d'adresses, 72
des codes G, 74
Lignes de programme pièce, 78
Programme
Adressage, 227
Initialisation, 235
Mémoire, 224
Répétition, 203
Saut, 113
Sauts, 110
Temps d'exécution, 694
Programme d'initialisation, 235
PROTA, 412
PROTD, 414
PROTS, 413
PSI
Orientation le long d'une enveloppe de cône, 358
Polynômes d'orientation, 365
PTP, 389
PTPG0, 389
PTPWOC, 389
PUD, 32
PUNCHACC, 657
PUTFTOC, 444
PUTFTOCF, 443
PW, 260

R

Rainure rectiligne - SLOT1
Programmation externe, 750
Rainure sur cercle - SLOT2
Programmation externe, 752
READ, 159
REAL, 31
Rectification en plongée oblique, 385
REDEF, 37
réducteur électronique, 610
RELEASE, 146
Relèvement rapide du contour, 140
REP, 56
REPEAT, 115
REPEATB, 115
Répétition de sections de programme
avec programmation indirecte CALL, 207
REPOSA, 561
REPOSH, 561
REPOSHA, 561
REPOSL, 561
REPOSQ, 561
REPOSQA, 561
RET, 188
RET (paramétrable), 189

RETB (paramétrable), 195
Retrait
  Assisté par CN, 708
  Contrôlé par le variateur, 713
RG, 27
RINDEX, 96
RMBBL, 561
RMEBL, 561
RMIBL, 561
RMNBL, 561
Rotation
  Vecteur d'orientation, 366
ROUND, 81
ROUNDUP, 164
Routine d'interruption
  Déplacement de retrait, 142
  mettre hors tension/sous tension, 138
  Nouvelle affectation, 138
  Relèvement rapide du contour, 140
  sens de déplacement programmable, 142
  Suppression, 139
Run MyScreens, 689

S

Saisie et recherche de plages interdites à la recherche, 559
Saut
  au début du programme, 109
  Repères de saut, 110
SAVE, 176
SBLOF, 178
SBLON, 178
SCPARA, 683
SD, 260
SD41610, 426
SD41611, 426
SD42475, 377
SD42476, 377
SD42477, 377
SD42900, 436
SD42910, 436
SD42920, 436
SD42930, 437
SD42935, 439
SD42940, 440
SD42984, 495
Section de programme
  Répétition, 115
Segmentation automatique des déplacements, 662
SET, 56
SETAL, 705
SETDNO, 469
SETINT, 136
SETM, 129
SETTCOR, 514
Simulation de valeurs pilotes, 608
SIN, 81
SLOT1 - Rainure rectiligne
  Programmation externe, 750
SLOT2 - Rainure sur cercle
  Programmation externe, 752
SOFT, 543
SOFTA, 543
SON, 657
SONS, 657
Sortie
  Vers un périphérique/fichier externe, 700
Sous-programme
  Appel avec transfert de paramètres, 201
  Appel indirect, 206
  Appel modal, 204
  Appel sans transfert de paramètres, 199
  Application, 165
  Chemin de recherche programmable, 210
  Nom, 166
  Répétition, 203
  Retour paramétrable (RET ...), 189
  Retour paramétrable (RETB ...), 195
SPF, 224
SPI, 671
SPIF1, 657
SPIF2, 657
Spline
  interpolation, 260
  Types, 266
Spline A, 266
Spline B, 267
Spline C, 268
SPLINEPATH, 271
SPN, 662
SPOF, 657
SPP, 662
SPRINT, 99
SQRT, 81
START, 129
STARTFIFO, 552
STAT, 390
STOPFIFO, 552
STOPRE, 552
STRING, 31
STRINGIS, 685
STRLEN, 95
SUBSTR, 97

Surfaçage - CYCLE61
Programmation externe, 759
Synchronisme
fin, 620
grossier, 620
Synchronisme de position, 617
Synchronisme de position avec décalage angulaire, 617
Synchronisme de vitesse de rotation, 617
SYNR, 31
SYNRW, 31
SYNW, 31
Système
disponibilité en fonction du, 5

T

Tableau, 56
Définition, 56
Elément, 57
Tampon
Mémoire, 552
TAN, 81
TANG, 531
TANGDEL, 536
TANGOF, 535
TANGON, 534
Taraudage avec porte-taraud compensateur - CYCLE840
Programmation externe, 826
Taraudage sans porte-taraud compensateur - CYCLE84
Programmation externe, 788
TCARR, 477
TCOABS, 477
TCOFR, 477
TCOFRZ, 477
Temps d'usinage, 695
Temps restant
pour une pièce, 696
THETA
Interpolation de la rotation de l'outil, 372
Rotation de l'orientation de l'outil, 366
TILT, 347
TLIFT, 533
TMOF, 669
TMON, 669
TOFFOF, 480
TOFFON, 480
TOLOWER, 94
TOOLENV, 501
TOUPPER, 94
Tourillon circulaire - CYCLE77
Programmation externe, 775
Tourillon rectangulaire - CYCLE76
Programmation externe, 773
TOWBCS, 437
TOWKCS, 437
TOWMCS, 437
TOWSTD, 437
TOWTCS, 437
TOWWCS, 437
TRAANG
avec angle programmable, 384
TRACON, 387
TRACYL, 381
TRAFOOF, 401
TRAFOON, 417
TRAILOF, 581
TRAILON, 581
Transformation à angle oblique (TRAANG)
avec angle programmable, 384
Transformation avec axe linéaire pivotant, 343
Transformation de l'orientation TRAORI
Cinématique de machine, 336
Déplacements d'orientation, 335
Programmation d'orientation, 345
Transformation 5/6 axes générique, 336
Variantes de programmation d'orientation, 345
Transformation de surfaces cylindriques, 337
Transformation polaire, 337
Transformations
concaténées, 387
Position initiale d'orientation de l'outil indépendante de la cinématique, 334
Transformation d'orientation, 333
Transformation trois, quatre ou cinq axes, 344
Transformations cinématiques, 334
Transformations concaténées, 335
TRANSMIT, 381
TRAORI, 344
Tronçonnage - CYCLE92
Programmation externe, 794
Trou oblong - LONGHOLE
Programmation externe, 755
TRUE, 31
TRUNC, 81
TU, 394
Type de cinématique, 476
Type de couplage, 620
Types de transformation
Fonctionnement général, 333

U

ULI, 45
UNTIL, 127

V

Valeur de réglage, 431
Valeur d'usure, 431
Variable
    conversion de type, 90
Variable frame
    affecter des valeurs, 307
    Appel de transformations de coordonnées, 301
    Variable frame prédéfinie, 303
Variables
    Conversion de type, 91
    Définies par l'utilisateur, 31
Variables système
    Etat du palpeur de mesure, 293
    Limitation du palpeur, 293
Vecteur de direction, 350
Vecteur d'orientation THETA, 366
VELOLIM, 571
VELOLIMA, 545

W

WAITC, 617
WAITE, 129
WAITENC, 681
WAITM, 129
WAITMC, 129
WHEN, 641
WHEN-DO, 652
WHENEVER, 641
WHENEVER-DO, 652
WHILE, 126
WORKPIECE, 715
WRITE, 154

X

XML utilisateur, 689
XOR, 83

Z

Zones de protection, 239