SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D
Preparação do trabalho

Manual de programação

Válido para

CONTROLE
SINUMERIK 840D sl / 840DE sl / 828D

Software CNC Versão 4.8 SP3

08/2018
6FC5398-2BP40-6KA2

Prefácio
Indicações básicas de segurança 1
Programação flexível de NC 2
Gerenciamento de arquivos e programas 3
Áreas de proteção 4
Comandos especiais de curso 5
Transformações de coordenadas (Frames) 6
Transformações 7
Cadeias cinemáticas 8
Prevenção de colisão com cadeias cinemáticas 9
Transformações com cadeias cinemáticas 10
Corretores de ferramentas 11
Comportamento de percurso 12
Acoplamentos de eixo 13
Ações síncronas 14
Oscilação 15
Estampagem e puncionamento 16
Retificação 17
Outras funções 18
Programas de desbaste próprios 19
Programação externa de ciclos 20
Tabelas 21
Apêndice A

Промэлектроника, sales@prom-elec.com

Informações jurídicas

Conceito de aviso

Este manual contém instruções que devem ser observadas para sua própria segurança e também para evitar danos materiais. As instruções que servem para sua própria segurança são sinalizadas por um símbolo de alerta, as instruções que se referem apenas à danos materiais não são acompanhadas deste símbolo de alerta. Dependendo do nível de perigo, as advertências são apresentadas como segue, em ordem decrescente de gravidade.

⚠ PERIGO
significa que **haverá** caso de morte ou lesões graves, caso as medidas de segurança correspondentes não forem tomadas.

⚠ AVISO
significa que **poderá haver** caso de morte ou lesões graves, caso as medidas de segurança correspondentes não forem tomadas.

⚠ CUIDADO
indica um perigo iminente que pode resultar em lesões leves, caso as medidas de segurança correspondentes não forem tomadas.

ATENÇÃO
significa que podem ocorrer danos materiais, caso as medidas de segurança correspondentes não forem tomadas.

Ao aparecerem vários níveis de perigo, sempre será utilizada a advertência de nível mais alto de gravidade. Quando é apresentada uma advertência acompanhada de um símbolo de alerta relativamente a danos pessoais, esta mesma também pode vir adicionada de uma advertência relativa a danos materiais.

Pessoal qualificado

O produto/sistema, ao qual esta documentação se refere, só pode ser manuseado por **pessoal qualificado** para a respectiva definição de tarefas e respeitando a documentação correspondente a esta definição de tarefas, em especial as indicações de segurança e avisos apresentados. Graças à sua formação e experiência, o pessoal qualificado é capaz de reconhecer os riscos do manuseamento destes produtos/sistemas e de evitar possíveis perigos.

Utilização dos produtos Siemens em conformidade com as especificações

Tenha atenção ao seguinte:

⚠ AVISO
Os produtos da Siemens só podem ser utilizados para as aplicações especificadas no catálogo e na respetiva documentação técnica. Se forem utilizados produtos e componentes de outros fornecedores, estes têm de ser recomendados ou autorizados pela Siemens. Para garantir um funcionamento em segurança e correto dos produtos é essencial proceder corretamente ao transporte, armazenamento, posicionamento, instalação, montagem, colocação em funcionamento, operação e manutenção. Devem-se respeitar as condições ambiente autorizadas e observar as indicações nas respetivas documentações.

Marcas

Todas denominações marcadas pelo símbolo de propriedade autoral ® são marcas registradas da Siemens AG. As demais denominações nesta publicação podem ser marcas em que os direitos de proprietário podem ser violados, quando usadas em próprio benefício, por terceiros.

Exclusão de responsabilidade

Nós revisamos o conteúdo desta documentação quanto a sua coerência com o hardware e o software descritos. Mesmo assim ainda podem existir diferenças e nós não podemos garantir a total conformidade. As informações contidas neste documento são revisadas regularmente e as correções necessárias estarão presentes na próxima edição.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
ALEMANHA

N.º de encomenda de documento: 6FC5398-2BP40-6KA2
℗ 08/2018 Sujeito a alterações

Copyright © Siemens AG 1995 - 2018.
Todos os direitos reservados

Prefácio

Documentação SINUMERIK

A documentação SINUMERIK é dividida nas seguintes categorias:

- Documentação geral/Catálogos
- Documentação do usuário
- Documentação do fabricante e assistência técnica

Mais informações

No seguinte endereço (https://support.industry.siemens.com/cs/de/en/view/108464614) poderão ser encontradas informações sobre os temas:

- Pedido de documentação / Visão geral das publicações
- Outros links para o download de documentos
- Uso da documentação online (localização e pesquisa de manuais e informações)

Caso haja dúvidas referentes à documentação técnica (por exemplo, sugestões, correções), envie um e-mail para o seguinte endereço (mailto:docu.motioncontrol@siemens.com).

mySupport/Documentação

No endereço (https://support.industry.siemens.com/My/ww/en/documentation) a seguir você encontra informações sobre como agrupar a documentação de modo personalizado com base nos conteúdos Siemens e como adequar a documentação da própria máquina.

Training

No seguinte endereço (http://www.siemens.com/sitrain) poderá encontrar informações sobre o SITRAIN - o training da Siemens para produtos, sistemas e soluções da técnica de acionamento e de automação.

FAQs

As perguntas mais frequentes podem ser encontradas nas páginas de Service&Support em Suporte ao produto (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

Informações sobre o SINUMERIK se encontram no endereço (http://www.siemens.com/sinumerik) abaixo.

Grupo destino

Esta publicação é dirigida a:

- Programadores
- Projetistas

Aplicação

O manual de programação possibilita a criação de progamas e interface de software para editar, testar e para corrigir erros.

Escopo padrão

No presente manual de programação está descrita a funcionalidade do escopo padrão. As complementações e alterações realizadas pelo fabricante da máquina são documentadas pelo fabricante da máquina.

No comando podem existir outras funções que não foram explicadas nesta documentação. Isso, no entanto, não implica nenhuma obrigação destas funções serem fornecidas com um novo controle ou em caso de serviço.

Da mesma forma, devido à grande variedade de itens, esta documentação não compreende todas as informações detalhadas de todos os tipos de produto, e também não podem ser considerados todos os casos possíveis de instalação, operação e manutenção.

Assistência técnica

Os números de telefone específicos do país para a consultoria técnica podem ser encontrados na internet, no seguinte endereço (https://support.industry.siemens.com/sc/ww/en/sc/2090) na área de "contato".

Informações sobre estrutura e conteúdo

Manual de programação fundamentos / preparação do trabalho

A descrição da programação de NC é dividida em 2 manuais:

1. **Fundamentos**
   O manual de programação básico é voltado para o operador de máquinas com conhecimentos específicos em fresamento, furação e torneamento. Exemplos simples de programação são usados para explicar as instruções, que também são definidas pela DIN 66025.
2. **Preparação do trabalho**
   O manual de programação "Preparação de trabalho" oferece ao técnico, conhecimentos sobre todas as possibilidades de programação. O ComandoSINUMERIK permite que com uma linguagem de programação especial sejam feitos complexos programas de peça (por exemplo, superfícies de formas livres, sincronismo de canais, ...) e facilita a programação de operações de alta complexidade.

Disponibilidade dos elementos da linguagem de NC descritos

Todos o elementos de linguagem de NC descritos no seguinte manual são disponíveis para SINUMERIK 840D sl. A disponibilidade com relação aoSINUMERIK 828D está indicada na tabela "Instruções Disponibilidade no SINUMERIK 828D (Página 904)".

Índice remissivo

Prefácio....3

1 Indicações básicas de segurança....19

1.1 Indicações gerais de segurança....19

1.2 Garantia e responsabilidade para exemplos de aplicativos....20

1.3 Industrial Security....21

2 Programação flexível de NC....23

2.1 Variáveis....23
2.1.1 Dados de sistema....23
2.1.2 Variáveis de usuário pré-definidas: Parâmetros de cálculo....26
2.1.2.1 Parâmetro de cálculo específico do canal (R)....26
2.1.2.2 Parâmetro de cálculo global (RG)....27
2.1.3 Variáveis de usuário pré-definidas: Variáveis de ligação....29
2.1.4 Definição de variáveis de usuário (DEF)....31
2.1.5 Redefinição de dados de sistema, dados de usuário e comandos de linguagem NC (REDEF)....37
2.1.6 Atributo: Valor de inicialização....40
2.1.7 Atributo: Valores de limite (LLI, ULI)....43
2.1.8 Atributo: Unidade física (PHU)....45
2.1.9 Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB)....47
2.1.10 Atributo: Classe de dados (DCM, DCI, DCU) - apenas SINUMERIK 828D....53
2.1.11 Visão geral dos atributos definíveis e redefiníveis....54
2.1.12 Definição e inicialização de variáveis de campo (DEF, SET, REP)....55
2.1.13 Definição e inicialização de variáveis de campo (DEF, SET, REP): Outras informações....60
2.1.14 Tipos de dados....61
2.1.15 examinar a existência de uma variável (ISVAR)....62
2.1.16 Ler valores de atributo/tipo de dado (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)....64

2.2 Programação indireta....70
2.2.1 Programação indireta de endereços....70
2.2.2 Programação indireta de comandos G....72
2.2.3 Programação indireta de atributos de posição (GP)....73
2.2.4 Programação indireta de linhas de programa de peça (EXECSTRING)....76

2.3 Funções de cálculo....78

2.4 Operadores de comparação e operadores lógicos....81

2.5 Correção da precisão em erros de comparação (TRUNC)....83

2.6 Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND)....85

2.7 Prioridade das operações....87

2.8 Possíveis conversões de tipo....88

2.9 Operações de String....89

2.9.1 Conversão de tipos para STRING (AXSTRING)....89
2.9.2 Conversão de tipos de STRING (NUMBER, ISNUMBER, AXNAME)....90
2.9.3 Encadeamento de Strings (<<)....91
2.9.4 Mudança para letras minúsculas / letras maiúsculas (TOLOWER, TOUPPER)....92
2.9.5 Definir o tamanho de uma Strings (STRLEN)....93
2.9.6 Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH)....94
2.9.7 Seleção de uma String parcial (SUBSTR)....95
2.9.8 Leitura e gravação dos caracteres individuais....96
2.9.9 Formatação de String (SPRINT)....97

2.10 Saltos e ramificações de programa....106
2.10.1 Salto de retorno ao início do programa (GOTOS)....106
2.10.2 Saltos de programa até marcadores de salto (GOTOB, GOTOF, GOTO, GOTOC)....107
2.10.3 Bifurcação do programa (CASE ... OF ... DEFAULT ...)....110

2.11 Repetição de partes do programa (REPEAT, REPEATB, ENDLABEL, P)....112

2.12 Estruturas de controle....119
2.12.1 Instrução condicional e bifurcação (IF, ELSE, ENDIF)....121
2.12.2 Loop de programa infinito (LOOP, ENDLOOP)....122
2.12.3 Loop contador (FOR ... TO ..., ENDFOR)....123
2.12.4 Loop de programa com condição no início do loop (WHILE, ENDWHILE)....124
2.12.5 Loop de programa com condição no fim do loop (REPEAT, UNTIL)....125
2.12.6 Exemplo de programa com estruturas de controle aninhadas....125

2.13 Comandos de coordenação (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)....127

2.14 Rotina de interrupção (ASUP)....132
2.14.1 Funcionamento de uma rotina de interrupção....132
2.14.2 Criar rotina de interrupção....133
2.14.3 Atribuição e partida de rotinas de interrupção (SETINT, PRIO, BLSYNC)....134
2.14.4 Desativar / ativar a atribuição de uma rotina de interrupção (DISABLE, ENABLE)....136
2.14.5 Deletar a atribuição de uma rotina de interrupção (CLRINT)....136
2.14.6 Retração rápida do contorno (SETINT LIFTFAST, ALF)....137
2.14.7 Sentido de deslocamento na retração rápida do contorno 139
2.14.8 Execução de movimentos com rotinas de interrupção....142

2.15 Troca de eixos, troca de fusos (RELEASE, GET, GETD)....144

2.16 Transferir eixo de outro canal (AXTOCHAN)....149

2.17 Ativar dados de máquina (NEWCONF)....151

2.18 Gravar arquivo (WRITE)....152

2.19 Deletar arquivo (DELETE)....156

2.20 Ler linhas no arquivo (READ)....157

2.21 Controle de presença de um arquivo (ISFILE)....159

2.22 Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)....160

2.23 Arredondamento (ROUNDUP)....162

2.24 Técnica de subrotinas....163
2.24.1 Informações gerais....163
2.24.1.1 Subrotina....163

2.24.1.2 Nomes de subrotina....164
2.24.1.3 Aninhamento de subrotinas....165
2.24.1.4 Caminho de localização....166
2.24.1.5 Parâmetros formais e parâmetros atuais....167
2.24.1.6 Transferência de parâmetros....167
2.24.2 Definição de uma subrotina....169
2.24.2.1 Subrotina sem transferência de parâmetros....169
2.24.2.2 Subrotina com transferência de parâmetros Call-by-Value (PROC)....170
2.24.2.3 Subrotina com transferência de parâmetros Call-by-Reference (PROC, VAR)....172
2.24.2.4 Salvar funções G modais (SAVE)....174
2.24.2.5 Supressão de processamento bloco a bloco (SBLOF, SBLON)....175
2.24.2.6 Supressão da atual exibição de bloco (DISPLOF, DISPLON, ACTBLOCNO)....181
2.24.2.7 Identificar subrotinas com preparação (PREPRO)....184
2.24.2.8 Salto de retorno para a subrotina M17....185
2.24.2.9 Salto de retorno para subrotina RET....186
2.24.2.10 Salto de retorno parametrizável da subrotina (RET ...)....187
2.24.2.11 Salto de retorno parametrizável da subrotina (RET ...)....193
2.24.3 Chamada de uma subrotina....197
2.24.3.1 Chamada de subrotina sem transferência de parâmetros....197
2.24.3.2 Chamada de subrotina com transferência de parâmetros (EXTERN)....199
2.24.3.3 Número de repetições de programa (P)....202
2.24.3.4 Chamada modal de subrotina (MCALL)....203
2.24.3.5 Chamada de subrotinas indireta (CALL)....205
2.24.3.6 Chamada de subrotina indireta com indicação da parte de programa que deve ser executada (CALL BLOCK ... TO ...)....206
2.24.3.7 Chamada indireta de um programa programado em linguagem ISO (ISOCALL)....207
2.24.3.8 Chamada de subrotina com indicação de caminho e parâmetros (PCALL)....208
2.24.3.9 Ampliar o caminho de localização na chamada de subrotinas (CALLPATH)....209
2.24.3.10 Executar subrotina externa (840D sl) (EXTCALL)....210
2.24.3.11 Executar subrotina externa (828D) (EXTCALL)....214
2.25 Técnica de macros (DEFINE ... AS)....219

3 Gerenciamento de arquivos e programas....223

3.1 Memória de programa....223
3.1.1 Memória de programa em NC....223
3.1.2 Memória externa de programas....225
3.1.3 Endereçamento dos arquivos da memória do programa....227
3.1.4 Atalho de busca na chamada de subrotinas....232
3.1.5 Consulta do atalho e do nome do arquivo....233
3.2 Memória de trabalho (CHANDATA, COMPLETE, INITIAL)....235

4 Áreas de proteção....239

4.1 Definir áreas de proteção (CPROTDEF, NPROTDEF)....239
4.2 Ativar/desativar áreas de proteção (CPROT, NPROT)....244
4.3 Verificação quanto à violação da área de proteção, limite da área de trabalho e chave fim de curso de software (CALCPOSI)....248

5 Comandos especiais de curso....259

5.1 Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN)....259

5.2 Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)....260
5.3 Agrupamento de Spline (SPLINEPATH)....271
5.4 Ligar/desligar a compressão de blocos NC (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)....273
5.5 Interpolação de polinômios (POLY, POLYPATH, PO, PL)....274
5.6 Referência ajustável do percurso (SPATH, UPATH)....280
5.7 Medição com apalpador comutável (MEAS, MEAW)....282
5.8 Medição específica de eixos (MEASA, MEAWA, MEAC) (Opção)....285
5.9 Funções especiais para o usuário OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)....295
5.10 Redução de avanço com desaceleração nos cantos (FENDNORM, G62, G621) 296
5.11 Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)....297

6 Transformações de coordenadas (Frames)....301

6.1 Transformação de coordenadas através de variável Frame....301
6.1.1 Variável Frame pré-definida ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)....303
6.2 Atribuições de valores nos frames....307
6.2.1 Atribuir valores diretos (valor de eixo, ângulo, escala)....307
6.2.2 Ler e alterar componentes de Frame (TR, FI, RT, SC, MI)....309
6.2.3 Cálculo com Fr ames....310
6.2.4 Definição das variáveis Frame (DEF FRAME)....312
6.3 Deslocamento aproximado e preciso (CTRANS, CFINE)....314
6.4 Deslocamento externo de ponto zero ($AA_ETRANS)....316
6.5 Definição do valor real com perda do status de referenciamento (PRESETON)....318
6.6 Definição do valor real sem a perda do status de referenciamento (PRESETONS)....320
6.7 Cálculo de Frame a partir de 3 pontos de medição no espaço (MEAFRAME)....322
6.8 Frames globais da NCU....326
6.8.1 Frames específicos de canal ($P_CHBFR, $P_UBFR)....327
6.8.2 Frames ativos no canal....327

7 Transformações....333

7.1 Programação geral de tipos de transformação....333
7.1.1 Movimentos de orientação nas transformações....335
7.1.2 Vista geral da transformação de orientação TRAORI....339
7.2 Transformação de três, quatro e cinco eixos (TRAORI)....341
7.2.1 Relações gerais do cabeçote de ferramenta cardânico....341
7.2.2 Transformação de três, quatro e cinco eixos (TRAORI)....344
7.2.3 Variantes da programação da orientação e posição básica (ORIRESET)....345
7.2.4 Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT)....347
7.2.5 Fresamento de topo (A4, B4, C4, A5, B5, C5)....353

7.2.6 Relação dos eixos de orientação (ORIWKS, ORIMKS)....354
7.2.7 Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)....356
7.2.8 Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)....358
7.2.9 Especificação de orientação de dois pontos de contato (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)....362

7.3 Polinômios de orientação (PO[ângulo], PO[coordenadas])....364

7.4 Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)....366

7.5 Orientações relativas à trajetória....369
7.5.1 Tipos de orientação relativos à trajetória....369
7.5.2 Rotação da orientação da ferramenta relativa à trajetória (ORIPATH, ORIPATHS, ângulo de rotação)....370
7.5.3 Interpolação relativa à trajetória da rotação da ferramenta (ORIROTC, THETA)....372
7.5.4 Suavização do decurso de orientação (ORIPATHS A8=, B8=, C8=)....374

7.6 Compressão da orientação (COMPON, COMPCURV, COMPCAD, COMPSURF)....375

7.7 Ligar/desligar a suavização do decurso de orientação (ORISON, ORISOF)....378

7.8 Transformação cinemática....380
7.8.1 Ligar a transformação dos lados frontais (TRANSMIT)....380
7.8.2 Ligar a transformação de superfície cilíndrica (TRACYL)....380
7.8.3 Ligar a transformação de ângulo oblíquo com ângulo programável (TRAANG)....383
7.8.4 Execução inclinada em máquinas de acabamento (G5, G7)....384

7.9 Ligar a transformação encadeada (TRACON)....386

7.10 Deslocamento PTP cartesiano....388
7.10.1 Alternar o curso cartesiano PTP (PTP, PTPG0, PTPWOC, CP)....388
7.10.2 Indicar a posição das articulações (STAT)....389
7.10.3 Especificar o ângulo do eixo (TU)....393
7.10.4 Exemplo 1: Deslocamento PTP de um robô de 6 eixos com transformação ROBX....396
7.10.5 Exemplo 2: Deslocamento PTP com transformação genérica de 5 eixos....397
7.10.6 Exemplo 3: PTPG0 e TRANSMIT....397

7.11 Condições gerais na seleção de uma transformação....399

7.12 Desselecionar a transformação (TRAFOOF)....400

8 Cadeias cinemáticas....401

8.1 Supressão de componentes (DELOBJ)....401

8.2 Averiguação do índice por nome (NAMETOINT)....404

9 Prevenção de colisão com cadeias cinemáticas....407

9.1 Teste sobre o par de colisão(COLLPAIR)....408

9.2 Solicitar novo cálculo do modelo de máquina da prevenção de colisão (PROTA)....409

9.3 Estabelecer a condição da área de proteção (PROTS) 410

9.4 Determinação da distância de duas áreas de proteção (PROTD)....411

10 Transformações com cadeias cinemáticas....413

10.1 Ligar transformação (TRAFOON)....413

10.2 Modificação da transformação de orientação após a medição da máquina (CORRTRAFO)....415

11 Corretores de ferramentas....423

11.1 Memória de correções....423

11.2 Corretores aditivos....426
11.2.1 Ativar correções aditivas (DL)....426
11.2.2 Definir valores de desgaste e de ajuste ($TC_SCPxy[t,d], $TC_ECPxy[t,d])....427
11.2.3 Deletar correções aditivas (DELDL)....428

11.3 Tratamentos especiais de corretor de ferramenta....430
11.3.1 Espelhar comprimentos de ferramentas....432
11.3.2 Avaliação de sinais de desgaste....432
11.3.3 Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)....433
11.3.4 Comprimento de ferramenta e mudança de planos....436

11.4 Compensação de ferramenta Online....438
11.4.1 Definir função polinomial (FCTDEF)....438
11.4.2 Gravar continuamente a correção de ferramenta Online (PUTFTOCF)....439
11.4.3 Gravar discretamente a correção de ferramenta on-line (PUTFTOC)....440
11.4.4 Ligar/desligar a correção de ferramenta on-line (FTOCON/FTOCOF)....441

11.5 Correção do raio de ferramenta 3D....442
11.5.1 Selecionar correção do raio da ferramenta em 3D para fresamento periférico em 3D (CUT3DC, CUT3DCD, ISD)....442
11.5.2 Selecionar correção do raio da ferramenta 3D para fresamento de topo 3D (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)....446
11.5.3 Fresamento periférico 3D levando em conta uma superfície de limitação (CUT3DCC, CUT3DCCD)....452

11.6 Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)....457

11.7 Seleção livre de número D, número de gume....463
11.7.1 Atribuição livre de números D, número de corte (endereço CE)....463
11.7.2 Atribuição livre de números D: Verificar números D (CHKDNO)....463
11.7.3 Atribuição livre de números D: Renomear números D (GETDNO, SETDNO)....464
11.7.4 Atribuição livre de números D: Determinar o número T para o número D especificado (GETACTTD)....465
11.7.5 Atribuição livre de números D: Invalidar números D (DZERO)....465

11.8 Cinemática do porta-ferramenta....466

11.9 Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)....472

11.10 Correção Online de comprimento de ferramenta (TOFFON, TOFFOF)....475

11.11 Modificação dos dados de correção em ferramentas giratórias....478
11.11.1 Cálculo das orientações (ORISOLH)....478
11.11.2 Ativação da alteração dos dados de correção para ferramentas orientáveis (CUTMOD, CUTMODK)....488

11.12 Trabalhar com ambientes de ferramenta....496
11.12.1 Armazenar o ambiente de ferramenta (TOOLENV)....496
11.12.2 Eliminar um ambiente de ferramenta (DELTOOLENV)....499
11.12.3 Leitura de números T, D e DL (GETTENV)....500
11.12.4 Leitura de informações sobre ambientes de ferramenta armazenadas ($P_TOOLENVN, ($P_TOOLENV)....501
11.12.5 Leitura de comprimentos de ferramenta ou componentes de comprimento de ferramenta (GETTCOR)....502
11.12.6 Modificar componentes de ferramenta (SETTCOR)....508
11.13 Leitura da atribuição de comprimentos de eixo L1, L2, L3 aos eixos de coordenada (LENTOAX)....521

12 Comportamento de percurso....525

12.1 Controle tangencial....525
12.1.1 Definição de acoplamento (TANG)....525
12.1.2 Ligar a geração de ponte (TLIFT)....527
12.1.3 Ligar acoplamento (TANGON)....528
12.1.4 Desligar acoplamento (TANGOF)....529
12.1.5 Deletar acoplamento (TANGDEL)....529
12.2 Característica de avanço (FNORM, FLIN, FCUB, FPO)....531
12.3 Comportamento da aceleração....536
12.3.1 Modo de aceleração (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)....536
12.3.2 Influência da aceleração em eixos escravos (VELOLIMA, ACCLIMA, JERKLIMA)....538
12.3.3 Ativação de valores de dinâmica específicos de tecnologia (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)....540
12.4 Deslocamento com controle antecipado (FFWON, FFWOF)....542
12.5 Precisão de contorno programável (CPRECON, CPRECOF)....543
12.6 Execução do programa com memória de pré-processamento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)....545
12.7 Definir áreas de stop-delay (DELAYFSTON, DELAYFSTOF)....548
12.8 Evitar posição de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK)....551
12.9 Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)....554
12.10 Controle na condução do movimento....563
12.10.1 Correção percentual de solavancos (JERKLIM)....563
12.10.2 Correção percentual da velocidade (VELOLIM)....564
12.10.3 Exemplo de programa para JERKLIM e VELOLIM....566
12.11 Programar tolerância de contorno/orientação (CTOL, OTOL, ATOL)....567
12.12 Ação de mudança de bloco no acoplamento ativo (CPBC)....572

13 Acoplamentos de eixo....573

13.1 Movimento acoplado (TRAILON, TRAILOF)....573
13.2 Tabelas de curvas (CTAB)....578
13.2.1 Definição de tabelas de curvas (CTABDEF, CATBEND)....578
13.2.2 Controle de presença de uma tabela de curvas (CTABEXISTS)....585

13.2.3 Eliminação de tabelas de curvas (CTABDEL)....585
13.2.4 Bloqueio de tabelas de curvas contra eliminação e sobregravação (CTABLOCK, CTABUNLOCK)....586
13.2.5 Tabelas de curvas: Determinação de propriedades de tabela (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)....588
13.2.6 Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)....589
13.2.7 Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)....594

13.3 Acoplamento axial de valor mestre (LEADON, LEADOF)....596

13.4 Caixa de transmissão eletrônica (EG)....602
13.4.1 Definir caixa de transmissão eletrônica (EGDEF)....602
13.4.2 Ativar caixa de transmissão eletrônica (EGON, EGONSYN, EGONSYNE)....603
13.4.3 Desativar transmissão eletrônica (EGOFS, EGOFC)....607
13.4.4 Deletar a definição de uma caixa de transmissão eletrônica (EGDEL)....607
13.4.5 Avanço por rotação (G95) / caixa de transmissão eletrônica (FPR)....608

13.5 Fuso sincronizado....609
13.5.1 Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)....609

13.6 Acoplamento genérico (CP...)....620

13.7 Acoplamento-mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)....628

14 Ações síncronas....631

14.1 Definição de uma ação síncrona....631

15 Oscilação....633

15.1 Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)....633

15.2 Oscilação controlada através de ações síncronas (OSCILL)....639

16 Estampagem e puncionamento....647

16.1 Ativação/desativação....647
16.1.1 Ativar/desativar estampagem e puncionamento (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)....647

16.2 Segmentação automática de curso....652
16.2.1 Segmentação de curso para eixos de percurso....655
16.2.2 Segmentação de curso para eixos individuais....656

17 Retificação....659

17.1 Ligar/desligar monitoramento da ferramenta específica de desbaste (TMON, TMOF)....659

18 Outras funções....661

18.1 Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)....661

18.2 Eixos geométricos comutáveis (GEOAX)....664

18.3 Contentor de eixos (AXCTSWE, AXCTSWED, AXCTSWEC)....669

18.4 Espera pela posição de eixo válida (WAITENC)....671

18.5 Bloco programável de alteração de parâmetros (SCPARA)....673

18.6 Controlar o escopo disponível de linguagem NC (STRINGIS)....675
18.7 Chamada interativa a partir do programa de peça (MMC)....679
18.8 Tempo de processamento do programa / contador de peças....684
18.8.1 Tempo de processamento do programa....684
18.8.2 Contador de peças....687
18.9 Process DataShare - Saída para um dispositivo/arquivo externo (EXTOPEN, WRITE, EXTCLOSE)....689
18.10 Alarmes (SETAL)....694
18.11 Parada e retrocesso ampliados (ESR)....696
18.11.1 ESR controlado por NC....697
18.11.1.1 Retrocessos controladas por NC (POLF, POLFA, POLFMASK, POLFMLIN)....697
18.11.1.2 Parada controlada por NC....701
18.11.2 Comando autárquico ESR....702
18.11.2.1 Configuração da parada independente de acionamento (ESRS)....702
18.11.2.2 Configuração do retrocesso independente de acionamento (ESRR)....702
18.12 Definir a peça bruta (WORKPIECE)....704
18.13 Comutar modo de linguagem (G290, G291)....708

19 Programas de desbaste próprios....711

19.1 Funções de apoio para o desbaste....711
19.2 Criar tabela de contorno (CONTPRON)....712
19.3 Criar tabela de contorno codificada (CONTDCON)....718
19.4 Determinar a intersecção entre dois elementos de contorno (INTERSEC)....722
19.5 Executar por blocos os elementos de contorno de uma tabela (EXECTAB)....724
19.6 Calcular dados de círculos (CALCDAT)....725
19.7 Desativar a preparação de contorno (EXECUTE)....727

20 Programação externa de ciclos....729

20.1 Ciclos tecnológicos....729
20.1.1 Introdução....729
20.1.2 Visão geral da tecnologia específica....730
20.1.3 HOLES1 - Fileira de furos....732
20.1.4 HOLES2 - Fileira de furos....732
20.1.5 POCKET3 - Fresamento de bolsão retangular....735
20.1.6 POCKET4 - Fresamento de bolsão circular....737
20.1.7 SLOT1 - Ranhura longitudinal....740
20.1.8 SLOT2 - Ranhura circular....743
20.1.9 LONGHOLE - Furo oblongo....745
20.1.10 CYCLE60 - Ciclo de gravação....747
20.1.11 CYCLE61- Fresamento de facear....750
20.1.12 CYCLE62 - Chamada de contorno....752
20.1.13 CYCLE63 - Fresar bolsão de contorno....753
20.1.14 CYCLE64 -Pré-furação do bolsão de contorno....755
20.1.15 CYCLE70 - Fresamento de roscas....757
20.1.16 CYCLE72 - Fresamento de percurso....758

20.1.17 CYCLE76 - Fresar a saliência retangular....763
20.1.18 CYCLE77 - Fresar a saliência circular....765
20.1.19 CYCLE78 - Fresar rosca....767
20.1.20 CYCLE79 - Poliedro....771
20.1.21 CYCLE81 - Furação, centragem....773
20.1.22 CYCLE82 - Furação, escareamento plano....774
20.1.23 CYCLE83 - Furação profunda....777
20.1.24 CYCLE84 - Rosqueamento com macho sem mandril de compensação....780
20.1.25 CYCLE85 - Alargamento....783
20.1.26 CYCLE86 - Mandrilamento....784
20.1.27 CYCLE92 - Separação....786
20.1.28 CYCLE95 - Desbaste do contorno....787
20.1.29 CYCLE98 - encadeamento de roscas....789
20.1.30 CYCLE99 - Torneamento de roscas....793
20.1.31 CYCLE435 - definir o sistema de coordenadas de ajuste....798
20.1.32 CYCLE495 - Perfilar....798
20.1.33 CYCLE800 - Rotação....800
20.1.34 CYCLE801 - Grade ou quadro....805
20.1.35 CYCLE802 - Posição definida pelo usuário....806
20.1.36 CYCLE830 - Furação de furo profundo 2....809
20.1.37 CYCLE832 - High Speed Settings....815
20.1.38 CYCLE840 - Rosqueamento com macho com mandril de compensação....818
20.1.39 CYCLE899 - Fresar calha....821
20.1.40 CYCLE930 - Canal....824
20.1.41 CYCLE940 - Formas de alívio....827
20.1.42 CYCLE951- Desbaste....829
20.1.43 CYCLE952 - Usinagem de canal de contorno....832
20.1.44 CYCLE4071 - Retificação longitudinal com penetração no ponto de inversão....839
20.1.45 CYCLE4072 - Retificação longitudinal com penetração no ponto de inversão e sinal de interrupção....840
20.1.46 CYCLE4073 - Retificação longitudinal com penetração contínua....844
20.1.47 CYCLE4074 - Retificação longitudinal com penetração contínua e sinal de interrupção....845
20.1.48 CYCLE4075 - Retificação plana com penetração no ponto de inversão....848
20.1.49 CYCLE4077 - Retificação plana com penetração no ponto de inversão e sinal de interrupção....851
20.1.50 CYCLE4078 - Retificação plana com penetração contínua....854
20.1.51 CYCLE4079 - Retificação plana com penetração intermitente....856
20.1.52 GROUP_BEGIN - Início bloco de programa....858
20.1.53 GROUP_END - Final bloco de programa....859
20.1.54 GROUP_ADDEND - Final adendo de entrada....859
20.1.55 Condições gerais....860
20.1.55.1 Escala tecnológica no interior das telas de ciclo....860

20.2 Ciclos de medição....862

21 Tabelas....863

21.1 Instruções....863

21.2 Instruções Disponibilidade no SINUMERIK 828D....904
21.2.1 Variantes de comando, fresamento / torneamento....904
21.2.2 Variantes de comando, retificar....932

21.3 Atual idioma na HMI....960

A Apêndice....961
A.1 Lista de abreviações....961
A.2 Vista Geral da documentação....970
Glossário....971
Índice....993

Indicações básicas de segurança

1

1.1 Indicações gerais de segurança

⚠ AVISO

Risco de vida em caso de inobservância das orientações de segurança e dos riscos residuais

Em caso de inobservância das orientações de segurança e dos riscos residuais na documentação de hardware pertinente, podem ocorrer acidentes com graves lesões ou morte.

- Respeite as indicações de segurança da documentação de hardware.
- Na avaliação de riscos, considere os riscos residuais.

⚠ AVISO

Funções com falha da máquina em consequência da parametrização incorreta ou alterada

Através da parametrização incorreta ou alterada podem se originar funções com falhas nas máquinas, as quais podem provocar graves lesões ou morte.

- Proteja os parâmetros contra um acesso não autorizado.
- Domine as possíveis funções com falhas através de medidas apropriadas, p. ex., PARADA DE EMERGÊNCIA ou DESLIGAMENTO DE EMERGÊNCIA.

1.2 Garantia e responsabilidade para exemplos de aplicativos

Os exemplos de aplicativos não são vinculativos e não pretendem ser exaustivos relativamente à configuração e equipamento, bem como a quaisquer eventualidades. Os exemplos de aplicativos não representam soluções personalizadas, devendo apenas servir de suporte de ajuda para tarefas típicas.

Como usuário, você é responsável pela operação apropriada dos produtos descritos. Os exemplos de aplicativos não o isentam da responsabilidade pelo manuseamento seguro durante a aplicação, instalação, operação e manutenção.

1.3 Industrial Security

Indicação

Industrial Security

A Siemens oferece produtos e soluções com funções Industrial Security, que suportam o funcionamento seguro de instalações, sistemas, máquinas e redes.

Para proteger instalações, sistemas, máquinas e redes contra ameaças cibernéticas, é necessário implementar um conceito de segurança industrial completo (e manter continuamente) que corresponda ao estado atual da tecnologia. Os produtos e soluções da Siemens formam apenas uma parte de um conceito.

O cliente torna-se responsável por evitar acesso não autorizado às suas instalações, sistemas, máquinas e redes. Sistemas, máquinas e componentes só devem ser conectados com a rede corporativa se isto for necessário, conforme o possível e se medidas de proteção adequadas forem tomadas (p.ex., uso de firewalls e segmentação de rede).

Além disso, as recomendações da Siemens relativas às medidas de proteção correspondentes devem ser seguidas. Outras informações sobre o Industrial Security podem ser encontradas em:

Industrial Security (http://www.siemens.com/industrialsecurity)

Os produtos e soluções da Siemens são continuamente aperfeiçoados, para torná-los ainda mais seguros. A Siemens recomenda veementemente executar as atualizações a partir do momento que os updates relevantes estejam disponíveis e sempre usar as versões de produtos mais recentes. O uso de versões antigas ou já não suportadas pode aumentar o risco de ameaças cibernéticas.

Para estar sempre informado sobre atualizações de produtos, assine o feed de RSS Siemens Industrial Security:

Industrial Security (http://www.siemens.com/industrialsecurity)

Você encontra mais informações na internet:

Manual do projeto de Industrial Security (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ AVISO

Estados operacionais inseguros através da manipulação do software

As manipulações do software, por ex., vírus, cavalos de troia, software malicioso ou vermes, podem provocar estados operacionais inseguros em sua instalação, o que pode provocar morte, graves lesões corporais e danos materiais.

- Mantenha o software atualizado.
- Integre os componentes de automação e de propulsão em um conceito de segurança industrial global ou na máquina de acordo com o nível atual da técnica.
- Considere em seu conceito de segurança industrial global todos os produtos utilizados.
- Proteja os arquivos nas mídias de armazenamento removível contra softwares danosos com medidas de proteção adequadas, como um scanner de vírus.
- Proteja o acionamento contra alterações não autorizadas ativando a função de conversor "Know-Schutz".

Programação flexível de NC

2

2.1 Variáveis

Através do uso de variáveis das áreas de dados do sistema e dados do usuário, especialmente em conjunto com funções de cálculo e estruturas de controle, é possível que os programas NC e ciclos possam ser ajustados de forma flexível.

- Dados de sistema
  Nos dados do sistema ficam as variáveis pré-definidas no sistema. Estas variáveis têm significado definido. Elas são usadas principalmente pelo software do sistema. Estas variáveis podem ser lidas e escritas pelos usuários em programas NC e ciclos. Exemplo: Dados da máquina, dados de ajuste, variáveis de sistema.
  Embora o significado de uma data de sistema seja pré-definido, as propriedades podem ser alteradas até certo ponto pelo usuário através da Redefinição.
  Veja "Redefinição de dados de sistema, dados de usuário e comandos de linguagem NC (REDEF) (Página 37)"
- Dados de usuário
  Nos dados do usuário ficam as variáveis definidas pelo usuário, tendo seus significados definidos exclusivamente pelo usuário. Estes não serão avaliados pelo sistema.
  Os dados do usuário são subdivididos em:
  - Variáveis de usuário pré-definidas
    As variáveis de usuário pré-definidas são variáveis já definidas no sistema, cuja grandeza será parametrizada através dos dados da máquina. As propriedades destas variáveis podem ser adaptadas pelo usuário.
    Consulte "Redefinição de dados de sistema, dados de usuário e comandos de linguagem NC (REDEF) (Página 37) ".
  - Variáveis definidas pelo usuário
    As variáveis definidas pelo usuário são variáveis que são definidas pelo usuário e são criadas pelo sistema durante o processamento. Sua grandeza, tipo de dado, visibilidade e todas demais propriedades são definidas exclusivamente pelo usuário.
    Veja "Definição de variáveis de usuário (DEF) (Página 31)"

2.1.1 Dados de sistema

Os dados do sistema contém as variáveis pré-definidas no sistema, que nos programas de NC e nos ciclos oferecem o acesso à atual parametrização do comando, assim como aos estados da máquina, do comando e do processo.

Variáveis de pré-processamento

As variáveis de pré-processamento são dados do sistema, que no contexto do pré--processamento, ou seja, no momento da interpretação do bloco, em qual a variável está programada, podem ser lidas e escritas. Variável de processamento não aciona a parada de pré-processamento.

Variáveis de processamento principal

As variáveis de processamento principal são dados do sistema, que no contexto de processamento principal, ou seja, no momento da execução do bloco, em qual a variável está programada, podem ser lidas ou escritas. As variáveis de processamento principal são:

- Variáveis, que podem ser programadas em ações síncronas (leitura/gravação)
- Variáveis, que podem ser programadas no programa NC e acionar parada de pré-processamento (leitura/gravação)
- Variáveis, que podem ser programadas no programa NC e que determinam o valor no pré-processamento, mas a gravação somente ocorre no processamento principal (sincronizado com o processamento principal: apenas gravação)

Sistemática de prefixos

Para uma identificação especial dos dados de sistema normalmente se coloca um prefixo no nome, que é composto pelo caractere $, seguido de uma ou duas letras e um sublinhado:

$ + 1. Letra	**Significado: Tipo de dados**
Dados do pré-processamento (dados do sistema que são lidos / gravados no pré-processamento)	
$M	Dados de máquina [1)]
$S	Dados de ajuste, áreas de proteção [1)]
$T	Dados de gerenciamento de ferramentas
$P	Valores programados
$C	Variáveis de ciclo dos ciclos envoltórios ISO
$O	Dados opcionais
R	Parâmetro R (parâmetro de cálculo) [2)]
Dados do processamento principal (dados do sistema que são lidos / gravados no processamento principal)	
$$M	Dados de máquina [1)]
$$S	Dados de ajuste [1)]
$A	Atuais dados de processamento principal
$V	Dados do controlador de posição
$R	Parâmetro R (parâmetro de cálculo) [2)]
[1)] Se os dados da máquina ou da configuração serão tratados como variáveis de pré-processamento ou principal, depende de eles estarem gravados com um ou dois sinais $. o modo de gravação é de livre escolha de acordo com a utilização específica. [2)] Para o uso de um parâmetro R no programa de peça / ciclo como uma variável de pré-processamento não escrito nenhum prefixo, p. ex. R10. Para o uso em uma ação síncrona como uma variável de processamento principal o prefixo é escrito com um caractere $, p. ex. $R10.	

2ª letra	**Significado: Visibilidade**
N	Variável global NC (**N**C)
C	Variável específica de canal (**C**hannel)
A	Variável específica de eixo (**A**xis)

Condições gerais

Exceções na sistemática de prefixos

As seguintes variáveis de sistema diferem da sistema de prefixos mencionada acima:

- $TC_...: A 2ª letra C neste caso não é uma variável específica de canal, mas variável de sistema específica de porta-ferramenta (TC = Tool Carrier)
- $P_ ...: Variáveis de sistema específicas de canal

Emprego de dados de máquina e dados de ajuste em ações síncronas

No emprego de dados de máquina e dados de ajuste em ações síncronas é possível determinar através do prefixo se os dados de máquina ou de ajuste são lidos e gravados de maneira síncrona no pré-processamento ou no processamento principal.

Se o dado permanece inalterado durante o processamento, a leitura pode ser sincronizada com o pré-processamento. Para isso o prefixo do dado de máquina ou do dado de ajuste é escrito com um caractere $:

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

Se o dado variar durante o processamento, a leitura / gravação deve ser sincronizada com o processamento principal. Para isso o prefixo do dado de máquina ou do dado de ajuste é escrito com dois caracteres $:

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

Indicação

Gravação dos dados de máquina e de ajuste

Na gravação de um dado de máquina ou de ajuste deve-se prestar atenção para que o nível de acesso ativo durante a execução do programa de peça / ciclo permita o acesso de gravação e que o efeito do dado seja "IMMEDIATE".

Literatura

Uma vista geral completa de todas as variáveis do sistema se encontra no:

Manual de listas Variáveis de sistema

Veja também

Variáveis (Página 23)

2.1.2 Variáveis de usuário pré-definidas: Parâmetros de cálculo

2.1.2.1 Parâmetro de cálculo específico do canal (R)

Os parâmetros de cálculo ou parâmetros R específicos do canal são variáveis de usuário pré-definidas com a identificação R, definida como campo do tipo de dado REAL. Por motivos históricos é permitido para o parâmetro R na gravação com campo de índice ex. `R[10]`, também na gravação sem campo de índice ex. `R10`.

Para o uso em ações síncronas deve ser colocada a letra $ na frente, ex `$R10`.

Sintaxe

Para uso como variável de pré-processamento:
`R<n>`
`R[<expressão>]`

Para uso como variável de processamento principal:
`$R<n>`
`$R[<expressão>]`

Significado

`R`:	Identificador para uso como variável fluxo, ex. Programa de peça	
`$R`:	Identificador para uso como variável de processamento principal, ex. em ações síncronas	
	Tipo:	REAL
	Faixa de valores:	Para forma escrita não exponencial: ± (0.000 0001 ... 9999 9999) **Nota:** São permitidas no máximo 8 casas decimais
		Para forma escrita exponencial: ± ($1*10^{-300}$ ... $1*10^{+300}$) **Nota:** • Modo de gravação: <Mantisse>EX<exponent> p. ex., 8.2EX-3 • São permitidos até 10 caracteres, inclusive o sinal precedente e o ponto decimal.

<n>:	Número do parâmetro R	
	Tipo:	INT
	Faixa de valores:	0 - MAX_INDEX **Nota** O MAX_INDEX resulta do número parametrizado de parâmetros R: MAX_INDEX = (MD28050 $MN_MM_NUM_R_PARAM) - 1
<expressão>:	Índice de campo Como índice de campo pode ser especificada uma expressão qualquer, enquanto o resultado da expressão permitir sua conversão para o tipo de dado INT (INT, REAL, BOOL, CHAR)	

Exemplo

Atribuições em parâmetros R e uso de parâmetros R em funções matemáticas:

```
Código de programa          Comentário
R0=3.5678                   ; Atribuição no pré-processamento
R[1]=-37.3                  ; Atribuição no pré-processamento
R3=-7                       ; Atribuição no pré-processamento
$R4=-0.1EX-5                ; Atribuição no processamento principal: R4 = -0.1 * 10^-5
$R[6]=1.874EX8              ; Atribuição no processamento principal: R6 = 1.874 * 10^8
R7=SIN(25.3)                ; Atribuição no pré-processamento

R[R2]=R10                   ; Endereçamento indireto através de parâmetro R
R[(R1+R2)*R3]=5             ; Endereçamento indireto através de  expressão matemática

X=(R1+R2)                   ; Desloca o eixo X até a posição resultante da soma de R1
                              e R2
Z=SQRT(R1*R1+R2*R2)         ; Desloca o eixo Z até a posição da raiz quadrada de (R1^2
                              + R2^2)
```

Veja também

Variáveis (Página 23)

2.1.2.2 Parâmetro de cálculo global (RG)

Função

Além dos parâmetros R específicos do canal, também são disponibilizados ao usuário os parâmetros R específicos de eixo. Eles existem uma vez no interior do comando e podem ser lidos/gravados por todos os canais.

Os parâmetros R globais são utilizados, por ex., para receber informações de um canal para o próximo. Um outro exemplo são as configurações globais, que devem ser avalizadas para todos os canais, como por ex., a projeção da peça bruta a partir do fuso.

A leitura e a gravação dos parâmetros R globais são realizadas por meio da interface de usuário ou no programa NC em avanço. Uma utilização nas ações síncronas ou ciclos tecnológicos não é possível.

Indicação

Ao ler e gravar os parâmetros R globais não ocorre **nenhuma** sincronização entre os canais.

Já que a leitura e a gravação ocorrem em avanço, o momento, no qual um valor gravado tem efeito de um canal para o outro canal, não é definido.

Exemplo:

No canal 1 é realizado um loop com um parâmetro R global como contador de loops. O canal 2 grava um valor neste parâmetro R global, que provoca uma interrupção do loop no canal 1. Entretanto, todos os loops interpretados até este momento no canal 1 em avanço são executados. Quantos loops são estes não é definido e depende, entre outros, do grau de aproveitamento do canal.

Uma ação síncrona entre os canais, o próprio usuário deve realizar de modo aplicativo, por ex., com as marcas WAIT!

Sintaxe

Gravação no programa NC

```
RG[<n>]=<valor>
RG[<expressão>]=<valor>
```

Leitura no programa NC

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

Significado

RG:	Nome default do endereço NC para o parâmetro R **global** **Nota:** o nome do endereço NC é ajustável através de MD15800 $MN_R_PARAM_NCK_NAME	
<n>:	Número do parâmetro R global	
	Tipo:	INT
	Faixa de valores:	0 ... MAX_INDEX **Nota** MAX_INDEX resulta do número parametrizado de parâmetros R globais: MAX_INDEX = (MD18156 $MN_MM_NUM_R_PARAM_NCK) - 1
<expressão>:	Como índice de campo pode ser especificada uma expressão qualquer, enquanto o resultado da expressão permitir sua conversão para o tipo de dado INT (INT, REAL, BOOL, CHAR)	

<valor>:	Valor do parâmetro R global	
	Tipo:	REAL
	Faixa de valores:	Para forma escrita não exponencial: ± (0.000 0001 ... 9999 9999) **Nota:** São permitidas no máximo 8 casas decimais
		Para forma escrita exponencial: ± ($1*10^{-300}$ ... $1*10^{+300}$) **Nota:** • Modo de gravação: <Mantisse>EX<exponent> p. ex., 8.2EX-3 • São permitidos até 10 caracteres, inclusive o sinal precedente e o ponto decimal.

2.1.3 Variáveis de usuário pré-definidas: Variáveis de ligação

Através das variáveis de ligação durante a função "NCU-Link" podem ser trocados dados cíclicos entre NCUs, que estão ligadas uma a outra através de uma rede. Aqui elas permitem o acesso específico de formato de dado à memória de variáveis de ligação. A memória de variáveis de ligação é definida de acordo com a instalação, tanto pelo tamanho como pela estrutura de dados do usuário / fabricante da máquina.

Variáveis de ligação são variáveis de usuários globais de sistema, que podem ser lidas e gravadas em programas de peça e em ciclos mediante uma comunicação de ligação (Link) configurada de todas NCUs do grupo de ligação. Em contrapartida às variáveis de usuários globais (GUD) as variáveis de ligação também podem ser utilizadas em ações síncronas.

Para instalações sem NCU-Link ativo, as variáveis de ligação locais do comando podem ser utilizadas paralelamente às variáveis de usuários globais (GUD) como variáveis de usuários globais extras.

Sintaxe

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

Significado

$A_DLB:	Variável de ligação para formato de dado BYTE (1 Byte)	
	Tipo de dado:	UINT
	Faixa de valores:	0 ... 255
$A_DLW:	Variável de ligação para formato de dado WORD (2 Bytes)	
	Tipo de dado:	INT
	Faixa de valores:	-32768 ... 32767

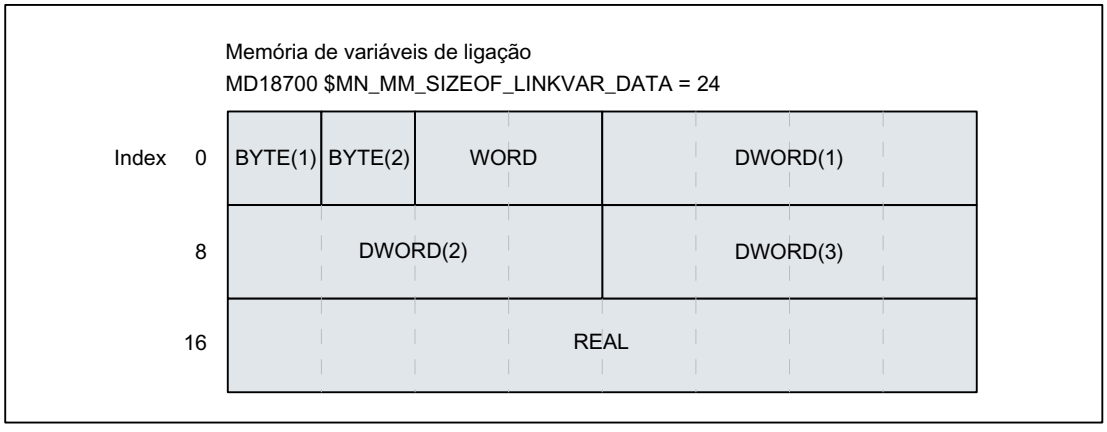
$A_DLD:	Variável de ligação para formato de dado DWORD (4 Bytes)	
	Tipo de dado:	INT
	Faixa de valores:	-2147483648 ... 2147483647
$A_DLR:	Variável de ligação para formato de dado REAL (8 Bytes)	
	Tipo de dado:	REAL
	Faixa de valores:	±(2,2*10^{-308} … 1,8*10^{+308})
<índice>:	Índice de endereço em Byte, calculado do início da memória de variáveis de ligação	
	Tipo de dado:	INT
	Faixa de valores:	0 - MAX_INDEX **Nota** • O MAX_INDEX resulta do tamanho parametrizado da memória de variáveis de ligação: MAX_INDEX = (MD18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1 • Somente podem ser programados índices, de modo que os Bytes endereçados na memória de variáveis de ligação estejam dentro de um limite de formato de dados ⇒ Índice = n * Bytes, com 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, ...

Exemplo

No sistema de automação estão disponíveis 2 NCUs (NCU1 e NCU2). Na NCU1 está conectado o eixo de máquina AX2, que deve ser deslocado como eixo de lincado da NCU2.

A NCU1 grava ciclicamente o valor real de corrente ($VA_CURR) do eixo AX2 na memória de variáveis de ligação. A NCU2 lê ciclicamente o valor real de corrente transmitido pela comunicação de ligação (Link) e mostra o alarme 61000 no caso de ser ultrapassado o valor limite.

A estrutura de dados na memória de variáveis de ligação está representada na figura a seguir. O valor real de corrente é transferido através do valor REAL.



NCU1

NCU1 grava ciclicamente uma ação síncrona estática durante o ciclo IPO o valor real de corrente do eixo AX2 através da variável de Link $A_DLR[16] na memória de variáveis de Link.

Código de programa

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

NCU2

NCU2 efetua uma leitura cíclica em uma ação síncrona estática durante o ciclo IPO do valor real de corrente do eixo AX2 através da variável de Link $A_DLR[16] a partir da memória de variáveis de Link. Se o valor real de corrente for maior que 23.0 A, será exibido o alarme 61000.

Código de programa

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

Veja também

Variáveis (Página 23)

2.1.4 Definição de variáveis de usuário (DEF)

Com o comando `DEF`, as variáveis de usuário específicos ou variáveis de usuário podem ser definidas (User Data) e dispostas com valores.

De acordo com a área de validade, ou seja, a área em que a variável é visível, existem as seguintes categorias de variáveis de usuários:

- Variáveis de usuários locais (LUD)
  As variáveis de usuários locais (LUD) são variáveis que estão definidas em um programa NC, que não é o programa principal no momento da execução. São criadas no momento da chamada do programa NC e apagadas com o reset do fim do programa ou com a próxima inicialização do sistema de comando. A LUD somente pode ser acessada no contexto do programa NC em que foi definida.
- Variáveis de usuários globais de programa (PUD)
  As variáveis de usuários globais de programa (PUD) são variáveis que são definidas em um programa NC utilizado como programa principal. São criadas no momento da chamada do programa NC e apagadas com o reset do fim do programa ou com a próxima inicialização do sistema de comando. Os dados PUD podem ser acessados no programa principal e em todas sub-rotinas.

  Indicação

  Disponibilidade de variáveis de usuário globais de programa (PUD)

  As variáveis de usuário globais de programa (PUD), definidas no programa principal, somente estão disponíveis nos subprogramas quando o seguinte dado de máquina estiver ajustado:

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  Com MD11120 = 0, as variáveis de usuário globais de programa, definidas no programa principal, estão disponíveis somente no programa principal.

- Variáveis de usuários globais (GUD)
  Variáveis de usuário globais (GUD) são variáveis globais de NC ou de canal, que estão definidas em um módulo de dados (SGUD, MGUD, UGUD, GUD4 ... GUD9) e que são conservados também após o reset do fim do programa ou após a próxima inicialização do sistema de comando. A GUD pode ser acessada em todos os programas NC.

As variáveis de usuários devem ser definidas antes de serem utilizadas (leitura/gravação). Aqui devem ser observadas as seguintes regras:

- As GUD devem ser definidas em um arquivo de definições, por ex. _N_DEF_DIR/ _N_MGUD_DEF.
- A PUD e a LUD devem ser definidas na parte de definições de um programa NC.
- A definição de dados deve ser realizada em um bloco próprio.
- Por definição de dados somente pode ser utilizado um tipo de dado.
- Por definição de dados podem ser definidas várias variáveis do mesmo tipo de dado.

Sintaxe

LUD e PUD

```
DEF <tipo> <unid.fís.> <val.lim.> <nome>[<valor_1>, <valor_2>,
<valor_3>]=<val. inic.>
```

GUD

```
DEF <área> <VL_Stop> <direitos de acesso> <classe de dados> <Tipo>
<unidade_física> <valores limite> <Nome>[<valor_1>, <valor_2>,
<valor_3>]=<valor_inicial>
```

Significado

DEF:	Comando para definição de variáveis de usuários GUD, PUD, LUD	
<área>:	Área de validade, apenas relevante para GUD:	
	NC:	Variáveis de usuários globais de NC
	CHAN:	Variáveis de usuários globais de canal
<VL_Stop>:	Parada de pré-processamento, apenas relevante para GUD (opcional)	
	SYNR:	Parada de fluxo na leitura
	SYNW:	Parada de fluxo na gravação
	SYNRW:	Parada de fluxo na leitura e gravação
<direitos_acesso>:	níveis de segurança para leitura / gravação do GUD através do programa NC ou BTSS (opcional)	
	APRP <nível_proteção>:	Leitura: Programa NC
	APWP <nível_proteção>:	Gravação: Programa NC
	APRB <nível_proteção>:	Leitura: BTSS
	APWB <nível_proteção>:	Gravação: BTSS
	<nível de proteção>:	Faixa de valores: 0 ... 7
	Veja "Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 47)"	
<Classe de dados>:	Atribuição da classe de dados (apenas SINUMERIK 828D!)	
	DCM:	Classe de dados M (= Manufacturer)
	DCI:	Classe de dados I (= Individual)
	DCU:	Classe de dados U (=User)
	Consulte "Atributo: Classe de dados (DCM, DCI, DCU) - apenas SINUMERIK 828D (Página 53) ".	
<tipo>:	Tipo de dado:	
	INT:	Valor inteiro com sinal
	REAL:	Número real (LONG REAL conforme IEEE)
	BOOL:	Valor lógico TRUE (1) / FALSE (0)
	CHAR:	Caracteres ASCII
	STRING[<comprimento_máx>]:	Sequência de caracteres de tamanho definido
	AXIS:	Identificador de eixo/fuso
	FRAME:	Especificações geométricas para uma transformação de coordenadas estática
	Veja "Tipos de dados (Página 61)"	
<unidade_física>:	Unidade física (opcional)	
	PHU <unidade>:	Unidade física
	Veja "Atributo: Unidade física (PHU) (Página 45)"	

<valores_limite>:	Valor limite inferior e superior (opcional)	
	LLI <valor_limite>:	Valor limite inferior (lower limit)
	ULI <valor_limite>:	Valor limite superior (upper limit)
	Veja "Atributo: Valores de limite (LLI, ULI) (Página 43)"	
<Nome>:	Nome das variáveis **Nota** • Máximo 31 caracteres • Os primeiros dois caracteres devem ser uma letra e/ou um sublinhado. • O caractere "$" está reservado para variáveis de sistema e não pode ser utilizado.	
[<valor_1>, <valor_2>, <valor_3>]:	Indicação dos tamanhos de campo para variáveis de campo de 1 até 3 dimensões (opcional) Para inicialização das variáveis de campo veja" Definição e inicialização de variáveis de campo (DEF, SET, REP) (Página 55)"	
<valor_inicialização>:	Valor de inicialização (opcional) veja "Atributo: Valor de inicialização (Página 40)" Para inicialização das variáveis de campo veja" Definição e inicialização de variáveis de campo (DEF, SET, REP) (Página 55)"	

Exemplos

Exemplo 1: Definições de variáveis de usuários no módulo de dados para fabricante da máquina

```
Código de programa                                          Comentário
%_N_MGUD_DEF                                                Módulo GUD: Fabricante da máquina
$PATH=/_N_DEF_DIR
DEF CHAN REAL PHU 24 LLI 0 ULI 10 CORRENTE_1, CORRENTE_2
Descrição
Definição de duas GUDs: CORRENTE_1, CORRENTE_2
Área de validade: Amplo canal
Tipo de dado: REAL
Parada de fluxo: não programada => valor Default = sem parada de pré-processamento
; (unidade) fis. Unidade: 24 = [A]
Valores limite: Low = 0.0, High = 10.0
Direitos de acesso: não programado => valor Default = 7 = posição do seletor com chave 0
Valor de inicialização: não programado => valor Default = 0.0

DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 TEMPO_1=12, TEMPO_2=45
Descrição
Definição de duas GUDs: TEMPO_1, TEMPO_2
Área de validade: NC amplo
Tipo de dado: REAL
```

Código de programa	Comentário
Parada de fluxo: não programada => valor Default = sem parada de pré-processamento	
; (unidade) fis. Unidade: 13 = [s]	
Valores limite: Low = 10.0, High = não programado => limite superior da faixa de definição	
Direitos de acesso:	
; Programa NC: Gravação/leitura = 3 = usuário final	
;BTSS: Gravação = 0 = Siemens, leitura = 3 = usuário final	
Valor de inicialização: TEMPO_1 = 12.0, TEMPO_2 = 45.0	
DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NOME = "COUNTER"	
Descrição	
Definição de uma GUD: GUD5_NOME	
Área de validade: NC amplo	
Tipo de dado: STRING, máx. 5 caracteres	
Parada de fluxo: não programada => valor Default = sem parada de pré-processamento	
; (unidade) fis. Unidade: não programada => valor Default = 0 = nenhuma unidade física	
Valores limite: não programado => limites de faixa de definição: Low = 0, High = 255	
Direitos de acesso:	
; Programa NC: Gravação/leitura = 3 = usuário final	
;BTSS: Gravação = 0 = Siemens, leitura = 3 = usuário final	
Valor de inicialização: "COUNTER"	
M30	

Exemplo 2: Variáveis de usuários globais de programa e locais (PUD / LUD)

Código de programa	Comentário
PROC MAIN	Programa principal
DEF INT VAR1	Definição PUD
...	
SUB2	;chamada de sub-rotina
...	
M30	

Código de programa	Comentário
PROC SUB2	Sub-rotina SUB2
DEF INT VAR2	; DEFINIÇÃO LUD
...	
IF (VAR1==1)	Leitura da PUD
VAR1=VAR1+1	;PUD leitura e gravação
VAR2=1	;LUD gravação
ENDIF	
SUB3	;chamada de sub-rotina
...	
M17	

Código de programa	Comentário
PROC SUB3	Sub-rotina SUB3
...	
IF (VAR1==1)	Leitura da PUD
VAR1=VAR1+1	;PUD leitura e gravação
VAR2=1	Erro: LUD não reconhecida pela SUB2
ENDIF	
...	
M17	

Exemplo 3: Definição e uso de variáveis de usuários do tipo de dado AXIS

Código de programa	Comentário
DEF AXIS ABSCISSA	; 1. Eixo geométrico
DEF AXIS SPINDLE	Fuso
...	
IF ISAXIS(1)==FALSE GOTOF CONTINUA	
ABSCISSA = $P_AXN1	
CONTINUA:	
...	
SPINDLE=(S1)	; 1. Fuso
OVRA[SPINDLE]=80	Override de fuso = 80%
SPINDLE=(S3)	; 3. Fuso

Condições gerais

Variáveis de usuários globais (GUD)

Durante a definição de variáveis de usuários globais (GUD) devem ser observados os seguintes dados de máquina:

Nº	Identificador: $MN_	Significado
11140	GUD_AREA_ SAVE_TAB	Segurança adicional para módulos GUD
18118 [1)]	MM_NUM_GUD_MODULES	Número de arquivos GUD no sistema de arquivos ativo
18120 [1)]	MM_NUM_GUD_NAMES_NCK	Número de nomes de GUDs globais
18130 [1)]	MM_NUM_GUD_NAMES_CHAN	Quantidade de GUDs específicas de eixos
18140 [1)]	MM_NUM_GUD_NAMES_AXIS	Quantidade de GUD específicas de eixos
18150 [1)]	MM_GUD_VALUES_MEM	Capacidade de memória para valores GUD globais
18660 [1)]	MM_NUM_SYNACT_GUD_REAL	Número de GUDs configuráveis de tipo de dado REAL
18661 [1)]	MM_NUM_SYNACT_GUD_INT	Número de GUDs configuráveis de tipo de dado INT
18662 [1)]	MM_NUM_SYNACT_GUD_BOOL	Número de GUDs configuráveis de tipo de dado BOOL

Nº	Identificador: $MN_	Significado
18663 [1)]	MM_NUM_SYNACT_GUD_AXIS	Número de GUDs configuráveis de tipo de dado AXIS
18664 [1)]	MM_NUM_SYNACT_GUD_CHAR	Número de GUDs configuráveis de tipo de dado CHAR
18665 [1)]	MM_NUM_SYNACT_GUD_STRING	Número de GUDs configuráveis de tipo de dado STRING

[1)] O MD no SINUMERIK 828D somente pode ser lido!

Utilização por outros canais de uma variável de usuário global de NC do tipo de dado AXIS

Uma variável de usuário global de NC do tipo de dado `AXIS`, que foi inicializada com um identificador de eixo pela definição no módulo de dados, somente pode ser utilizada em diferentes canais do NC se o eixo tiver o mesmo número de eixo de canal nestes canais.

Se este não for o caso, a variável deve ser carregada no início do programa NC, como mostra o próximo exemplo, deve ser utilizada a função AXNAME (...) (veja "Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 661)").

Código de programa	Comentário
`DEF NCK STRING[5] EIXO="X"`	`Definição em módulo de dados`
`...`	
`N100 AX[AXNAME(EIXO)]=111 G00`	`Utilização no programa NC`

2.1.5 Redefinição de dados de sistema, dados de usuário e comandos de linguagem NC (REDEF)

Com o comando `REDEF` podem ser alterados os atributos dos dados do sistema, dados do usuário e comandos de linguagem NC. A condição básica para uma redefinição é que ela seja executada a tempo após a definição correspondente.

Em uma redefinição não podem ser alterados vários atributos simultaneamente. Para cada atributo que deve ser alterado deve ser programada uma instrução `REDEF` própria.

Se forem programadas várias alterações concorrentes de atributos, então a última é que sempre estará ativa.

redefinir valores de atributos

O atributo modificado com `REDEF` para a permissão de acesso e ponto de inicialização podem ser redefinidos para o valor default através de nova reprogramação do `REDEF`, seguido do nome da variável ou do comando de voz do NC:

- Direitos de acesso: Nível de proteção 7
- Ponto de inicialização: sem inicialização ou manter o valor atual

Atributos redefiníveis

Consulte "Visão geral dos atributos definíveis e redefiníveis (Página 54) ".

Variáveis de usuários locais (PUD / LUD)

Para variáveis de usuários locais (PUD / LUD) não pode ser realizada nenhuma redefinição.

Sintaxe

```
REDEF <nome> <parada_pré_processamento>
REDEF <nome> <unidade_física>
REDEF <nome> <valores_limite>
REDEF <nome> <direitos_acesso>
REDEF <nome> <momento_inicialização>
REDEF <nome> <momento> <valor_inicialização>
REDEF <nome> <classe de dados>
REDEF <nome>
```

Significado

REDEF:	Comando de redefinição de um certo atributo ou redefinição de um atributo "direito de acesso" e/ou "ponto de inicialização do sistema de variáveis, variáveis do usuário e comandos de voz do NC	
<Nome>:	Nome de uma variável ou de um comando de linguagem NC que já foi definido	
<Parada de fluxo:	Parada de pré-processamento	
	SYNR:	Parada de fluxo na leitura
	SYNW:	Parada de fluxo na gravação
	SYNRW:	Parada de fluxo na leitura e gravação
<unidade_física>:	Unidade física	
	PHU <unidade>:	Unidade física
	Consulte "Atributo: Unidade física (PHU) (Página 45) ". **Nota** Não redefinível para: • Variáveis de sistema • Dados de utilização globais (GUD) dos tipos de dados: BOOL, AXIS, STRING, FRAME	
<valores_limite>:	Valor limite inferior / superior	
	LLI <valor_limite>:	Valor limite inferior (lower limit)
	ULI <valor_limite>:	Valor limite superior (upper limit)
	Consulte "Atributo: Valores de limite (LLI, ULI) (Página 43) ". **Nota** Não redefinível para: • Variáveis de sistema • Dados de utilização globais (GUD) dos tipos de dados: BOOL, AXIS, STRING, FRAME	

<direitos_acesso>:	Direitos de acesso para leitura / gravação através de programa de peça ou BTSS	
	APX <nível_proteção>:	Execução: Elemento de linguagem NC
	APRP <nível_proteção>:	Leitura: Programa de peça
	APWP <nível_proteção>:	Gravação: Programa de peça
	APRB <nível_proteção>:	Leitura: BTSS
	APWB <nível_proteção>:	Gravação: BTSS
	<nível de proteção>:	Faixa de valores: 0 ... 7
	Consulte "Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 47) ".	
<momento_inicialização>:	Momento em que a variável deve ser reinicializada	
	INIPO:	Power On
	INIRE:	Fim do programa principal, NC-Reset ou Power On
	INICF:	NEWCONF ou fim do programa principal, NC-Reset ou Power On
	PRLOC:	Fim do programa principal, NC-Reset após alteração local ou Power On
	Consulte "Atributo: Valor de inicialização (Página 40) ".	
<valor_inicialização>:	Valor de inicialização Com a redefinição do valor de inicialização sempre deve ser especificado junto um momento de inicialização (veja <momento_inicialização>). Consulte "Atributo: Valor de inicialização (Página 40) ". Para inicialização das variáveis de campo veja" Definição e inicialização de variáveis de campo (DEF, SET, REP) (Página 55)". **Nota** não é possível redefinir para variáveis de sistema, a não ser os dados de ajuste.	
<Classe de dados>:	Atribuição da classe de dados (apenas SINUMERIK 828D!)	
	DCM:	Classe de dados M (= Manufacturer)
	DCI:	Classe de dados I (= Individual)
	DCU:	Classe de dados U (=User)
	Consulte "Atributo: Classe de dados (DCM, DCI, DCU) - apenas SINUMERIK 828D (Página 53) ".	

Exemplo

Redefinições da variável de sistema $TC_DPCx no módulo de dados para fabricante da máquina

Código de programa

```
%_N_MGUD_DEF                                Módulo GUD: Fabricante da máquina
N100 REDEF $TC_DPC1 APWB 2 APWP 3
N200 REDEF $TC_DPC2 PHU 21
```

```
Código de programa
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
```

para N100: Direito de acesso para gravação: BTSS = nível de proteção 2, programa de peça = nível de proteção 3

para N200: Unidade física [%]

para N300: valor limite inferior = 0, valor limite superior = 200

para N400: A variável de campo é inicializada com o PowerOn com os quatro valores.

para N800 / N900 Redefinição dos valores de atributos "direitos de acesso" e/ou momento de inicialização

Indicação

Utilização dos arquivos ACCESS

Na utilização de arquivos ACCESS a Redefinição dos direitos de acesso de _N_MGUD_DEF deve ser deslocada para _N_MACCESS_DEF.

Condições gerais

Granularidade

Uma redefinição sempre estará relacionada à variável inteira, identificada de forma única através de seu nome. Não é possível, p.ex. em variáveis de campo atribuir diferentes valores de atributo para elementos de campo individuais.

2.1.6 Atributo: Valor de inicialização

Definição (DEF) de variáveis de usuários

Com a definição pode ser definido um valor de inicialização para as seguintes variáveis de usuários:

- Variáveis de usuários globais (GUD)
- Variáveis de usuários globais de programa (PUD)
- Variáveis de usuários locais (LUD)

Redefinição (REDEF) de variáveis de sistema e de usuário

Com a redefinição pode ser definido um valor de inicialização para as seguintes variáveis:

- Dados de sistema
  - Dados de ajuste
- Dados de usuário
  - Parâmetros R
  - Variável de ação síncrona ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD de ação síncrona (SYG_xy[ ], com x=R, I, B, A, C, S e y=S, M, U, 4, ..., 9)
  - Parâmetros EPS
  - Dados de ferramenta – OEM
  - Dados de magazine – OEM
  - Variáveis de usuários globais (GUD)

Momento da reinicialização

No caso da redefinição pode ser especificado o momento, no qual a variável é reinicializada, isto é, definida novamente para o valor de inicialização:

- `INIPO` (Power On)
  A variável é reinicializada com PowerOn.
- `INIRE` (Reset)
  A variável é reinicializada com NC-Reset, BAG-Reset, fim do programa de usinagem (`M02` / `M30`) ou com Power O.
- `INICF` (NEWCONF)
  A variável é reinicializada com a função "Ativar dados de máquina" através da HMI, comando de programa de peça `NEWCONF` ou NC-Reset, BAG-Reset, fim de programa de peça (M02 / M30) ou Power On.
- `PRLOC` (alteração local de programa)
  A variável somente é reinicializada com NC-Reset, BAG-Reset ou fim de programa (M02 / M30), se ela for alterada no contexto do atual programa de peça.
  O atributo `PRLOC` somente pode ser utilizado no contexto com dados de ajuste programáveis (veja a tabela a seguir).

Tabelas 2-1 Dados de ajuste programáveis

Número	Identificador	Comando 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`
43230	$SA_SPIND_MAX_VELO_LIMS	`LIMS`
43300	$SA_ASSIGN_FEED_PER_REV_SOURCE	`FPRAON`

Número	Identificador	Comando G [1)]
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) com este comando G é acionado o dado de ajuste		

Condições gerais

Valor de inicialização: Variáveis de usuários globais (GUD)

- Para variáveis de usuários globais (GUD) com a área de validade NC somente pode ser definido o INIPO (Power On) como momento de inicialização.
- Para variáveis de usuários globais (GUD) com a área de validade CHAN, além do INIPO (Power On) também o INIRE (Reset) ou o INICF (NEWCONF) pode ser definido como momento de inicialização.
- Para variáveis de usuários globais (GUD) com a área de validade CHAN e momento de inicialização INIRE (Reset) ou INICF (NEWCONF), com NC-Reset, BAG-Reset e "Ativar dados de máquina" somente serão reinicializadas nos canais as variáveis, nas quais foram disparados os eventos mencionados.

Valor de inicialização: Tipo de dado FRAME

Para variáveis do tipo de dado FRAME não pode ser especificado nenhum valor de inicialização. As variáveis do tipo de dado FRAME sempre são inicializadas de modo implícito com o Frame Default.

Valor de inicialização: Tipo de dado CHAR

Para variáveis do tipo de dado CHAR ao invés do código ASCII (0...255), também pode ser programado o respectivo caractere ASCII entre aspas, p. ex. "A"

Valor de inicialização: Tipo de dado STRING

Para variáveis do tipo de dado STRING devem ser colocadas aspas na sequência de caracteres, p. ex.: ...= "MASCHINE_1"

Valor de inicialização: Tipo de dados AXIS

Para variáveis do tipo de dado `AXIS` com a forma escrita de endereços ampliada, o identificador de eixo deve ser colocado entre parênteses, p. ex.: ...=(X3)

Valor de inicialização: Variável de sistema

Para variáveis de sistema não pode ser definido nenhum valor de inicialização específico de usuário através de redefinição. Os valores de inicialização das variáveis de sistema são estabelecidas pelo sistema. Entretanto, através da redefinição pode ser alterado o momento (`INIRE`, `INICF`) em que a variável é reinicializada.

Valor de inicialização implícito: Tipo de dados AXIS

Para variáveis do tipo de dado `AXIS` são utilizados os seguintes valores de inicialização implícitos:

- Dados do sistema: "primeiro eixo geométrico"
- GUD de ação síncrona (Denominação: SYG_A*), PUD, LUD:
  Identificador de eixo do dado de máquina: MD20082
  $MC_AXCONF_CHANAX_DEFAULT_NAME

Valor de inicialização implícito: Dados de ferramenta e de carregador

Para dados de ferramenta e de magazine podem ser definidos valores de inicialização através do seguinte dado de máquina: MD17520 $MN_TOOL_DEFAULT_DATA_MASK

Indicação

Sincronização

A sincronização de eventos que disparam uma reinicialização de uma variável global com a leitura desta variável em outro ponto, é de responsabilidade exclusiva do usuário / fabricante da máquina.

Veja também

Variáveis (Página 23)

2.1.7 Atributo: Valores de limite (LLI, ULI)

Um valor limite superior e inferior da faixa de definição somente pode ser especificado para os seguintes tipos de dados:

- INT
- REAL
- CHAR

Definição (DEF) de variáveis de usuários: Valores limite e valores de inicialização implícitos

Se na definição de uma variável de usuário de um dos tipos de dados mencionados acima não for definido nenhum valor de inicialização explícito, a variável é passada para o valor de inicialização implícito do tipo de dado:

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

Se o valor de inicialização implícito estiver fora da faixa de definição definida pelos valores limite programados, a variável é inicializada com o valor de limite que estiver mais próximo do valor de inicialização implícito:

- Valor de inicialização implícito < valor limite inferior (LLI) ⇒
  Valor de inicialização = valor limite inferior
- Valor de inicialização implícito > valor limite superior (ULI) ⇒
  Valor de inicialização = valor limite superior

Exemplos:

Código de programa	**Comentário**

`DEF REAL GUD1`	`; Valor limite inferior = limite da faixa de definição`
	`; Valor limite superior = limite da faixa de definição`
	`; Nenhum valor de inicialização programado`
	`; => valor de inicialização implícito = 0.0`

`DEF REAL LLI 5.0 GUD2`	`; Valor limite inferior = 5.0`
	`; Valor limite superior = limite da faixa de definição`
	`; => Valor de inicialização = 5.0`

`DEF REAL ULI -5 GUD3`	`; Valor limite inferior = limite da faixa de definição`
	`; Valor limite superior = -5.0`
	`; => Valor de inicialização = -5.0`

Redefinição (REDEF) de variáveis de usuários: Valores limite e atuais valores reais

Se com a redefinição dos valores limite uma das variáveis de usuários for alterada de modo que o atual valor real esteja fora da nova faixa de definição, então será emitido um alarme e os valores limite não serão assumidos.

Indicação

Redefinição (REDEF) de variáveis de usuários

Na redefinição dos valores limite de uma variável de usuário deve-se observar para uma modificação consistente dos seguintes valores:

- Valores limite
- Valor real
- Valor de inicialização na redefinição e na reinicialização automática em função do INIPO, INIRE ou INICF

Veja também

Variáveis (Página 23)

2.1.8 Atributo: Unidade física (PHU)

Uma unidade física somente pode ser especificada para variáveis dos seguintes tipos de dados:

- INT
- REAL

Unidades físicas programáveis (PHU)

A especificação da unidade física é dada como número vírgula fixa: `PHU <unidade>`

Podem ser programadas as seguintes unidades físicas:

<unidade>	Significado	Unidade física
0	Nenhuma unidade física	-
1	Posição linear ou angular 1)2)	[mm], [pol.], [graus]
2	Posição linear 2)	[mm], [pol.]
3	Posição angular	[graus]
4	Posição linear ou angular 1)2)	[mm/min], [pol./min], [rpm]
5	Posição linear 2)	[mm/min]
6	Velocidade angular	[rpm]
7	Posição linear ou angular 1)2)	[m/s^2], [$pol./s^2$], [$rot./s^2$]
8	Posição linear 2)	[m/s^2], [$pol./s^2$]
9	Aceleração angular	[$rot./s^2$]
10	Posição linear ou angular 1)2)	[m/s^3], [$pol./s^3$], [$rot./s^3$]

<unidade>	Significado	Unidade física
11	Posição linear 2)	[m/s³], [pol./s³]
12	Solavanco angular	[rot./s³]
13	Tempo	[s]
14	Amplificação do circuito do controlador de posição	[16.667/s]
15	Avanço por rotação 2)	[mm/rot.], [pol./rot.]
16	Compensação de temperatura 1)2)	[mm], [pol.]
18	Força	[N]
19	Massa	[kg]
20	Momento de inercia 3)	[kgm²]
21	Por cento	[%]
22	Frequência	[Hz]
23	Tensão	[V]
24	Corrente	[A]
25	Temperatura	[°C]
26	Ângulo	[graus]
27	KV	[1000/min]
28	Posição linear ou angular 3)	[mm], [pol.], [graus]
29	Velocidade de corte 2)	[m/min], [feet/min]
30	Velocidade periférica 2)	[m/s], [feet/s]
31	Resistência	[Ohm]
32	Indutância	[mH]
33	Torque 3)	[Nm]
34	Constante de torque 3)	[Nm/A]
35	Amplificação do controlador de corrente	[V/A]
36	Amplificação do controlador de rotação 3)	[Nm/(rad*s)]
37	Número de rotações	[rpm]
42	Potência	[kW]
43	Corrente, pequena	[µA]
46	Torque, pequeno 3)	[µNm]
48	Por milha	-
49	-	[Hz/s]
65	Fluxo (vazão)	[l/min]
66	Pressão	[bar]
67	Volume 3)	[cm³]
68	Amplificação de percurso 3)	[mm/(V*min)]
69	Amplificação de percurso do controlador de força	[N/V]
155	Passo de rosca 3)	[mm/rot.], [pol./rot.]
156	Variação de passo de rosca 3)	[mm/rot. / rot.], [pol./rot. / rot.]
1) A unidade física depende do tipo de eixo: Eixo linear ou eixo rotativo		

<unidade>	Significado	Unidade física
2) Comutação do sistema de medidas G70/G71(polegadas/métrico) Após uma comutação do sistema básico (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) com G70/G71, nos acessos de gravação e leitura das variáveis de sistema e de usuário referentes aos comprimentos (distâncias) não ocorre **nenhuma** conversão dos valores (valor real, valor nominal e valores de limite) G700/G710(polegadas/métrico) Após uma comutação do sistema básico (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) com G700/G710, nos acessos de gravação e leitura das variáveis de sistema e de usuário referentes aos comprimentos (distâncias) ocorre **uma** conversão dos valores (valor real, valor nominal e valores de limite)		
3) A variável **não** é convertida automaticamente para o atual sistema de medidas do NC (polegadas/métrico). A conversão é de responsabilidade exclusiva do usuário / fabricante da máquina.		

Indicação

Excesso de níveis através da conversão de formato

O formato interno de armazenamento para todas variáveis de usuários (GUD / PUD / LUD) com unidades físicas de distância é métrico. Uma utilização excessiva deste tipo de variável no processamento principal do NCK, p.ex. em ações síncronas, pode levar na mudança do sistema de medidas a um estouro do tempo (timeout) de processamento do plano interpolar, alarme 4240.

Indicação

Compatibilidade de unidades

Ao serem utilizadas variáveis (atribuição, comparação, cálculos, etc.) não é realizado nenhum controle de compatibilidade das unidades envolvidas. Uma eventual necessidade de conversão é de responsabilidade exclusiva do usuário / fabricante da máquina.

Veja também

Variáveis (Página 23)

2.1.9 Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB)

Designação

A designação do atributo de acesso `AP`... é desligada junto:

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

Direitos de acesso / Graus de proteção

Os direitos de acesso correspondem aos seguintes níveis de proteção que devem ser informados na programação:

Direito de acesso	Nível de proteção
Senha do sistema	0
Senha do fabricante da máquina	1
Senha da Assistência Técnica	2
Senha do usuário final	3
Seletor com chave na posição 3	4
Seletor com chave na posição 2	5
Seletor com chave na posição 1	6
Seletor com chave na posição 0	7

Definição (`DEF`) de dados do usuário

Os direitos de acesso (`APR`... / `APW`...) podem ser definidos para os seguintes dados:

- Dados de usuário globais (GUD)

Redefinição(REDEF) de dados de sistema e de usuário

Os direitos de acesso (APR... / APW...) podem ser redefinidos para os seguintes dados:

- Dados de sistema
  - Dados de máquina

    Indicação

    Redefinição dos direitos de leitura de dados de máquinas

    O nível de proteção para a leitura de dados de máquinas somente pode ser definido com a palavra-chave APR em conjunto para o programa de peças e BTSS.

    As palavras-chave APRP e APRB não são válidas para a redefinição dos direitos de leitura e causam um aviso de alarme 12490 "Direito de acesso APRP/APRB <Nível de proteção> não foi ajustado".

  - Dados de ajuste
  - Variável de sistema
  - Dados de processo
  - Dados de magazine
  - Dados de ferramenta
- Dados de usuário
  - Parâmetros R
  - Variável de ação síncrona ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD de ação síncrona (SYG_xy[ ], com x=R, I, B, A, C, S e y=S, M, U, 4, ..., 9)
  - Parâmetros EPS
  - Dados de ferramenta – OEM
  - Dados de magazine – OEM
  - Variáveis de usuários globais (GUD)

    Indicação

    Na redefinição o direito de acesso pode ser atribuído livremente a uma variável entre o nível de proteção mais baixo 7 e o próprio nível de proteção, por exemplo 1 (fabricante da máquina).

Redefinição (**REDEF**) de comandos de linguagem NC

O direito de acesso e de execução (APX) pode ser redefinido para os seguintes comandos de linguagem NC:

- Comandos G / condições de curso
  Literatura
  Manual de programação Fundamentos; capítulo: Comandos G / condições de curso
- Funções pré-definidas
  Literatura
  Manual de programação Fundamentos; capítulo: Funções pré-definidas
- Chamadas de sub-rotina pré-definidas
  Literatura
  Manual de programação Fundamentos; capítulo: Chamadas de sub-rotina pré-definidas
- Instrução DO para ações síncronas
- Identificador de programa de ciclos
  O ciclo deve ser armazenado em um diretório de ciclos e conter uma instrução PROC.

Os direitos de acesso relacionados aos programas NC e ciclos (**APRP**, **APWP**)

Os diferentes direitos de acesso possuem os seguintes efeitos para o acesso de um programa NC ou ciclo:

- APRP 0 / APWP 0
  - a senha de sistema deve ser informada para execução do programa NC
  - o ciclo deve estar armazenado no diretório _N_CST_DIR (sistema)
  - para o diretório _N_CST_DIR o direito de execução deve ser ajustado para sistema no MD11160 $MN_ACCESS_EXEC_CST
- APRP 1 / APWP 1 ou APRP 2 / APWP 2
  - a senha do fabricante de máquina ou de assistência técnica deve ser informada para execução do programa NC
  - o ciclo deve estar armazenado no diretório _N_CMA_DIR (fabricante da máquina) ou _N_CST_DIR
  - para os diretórios _N_CMA_DIR e _N_CST_DIR os direitos de execução devem estar ajustados pelo menos para o fabricante de máquina nos dados de máquina MD11161 $MN_ACCESS_EXEC_CMA e MD11160 $MN_ACCESS_EXEC_CST, respectivamente

- `APRP 3` / `APWP 3`
  - a senha do usuário final deve ser informada para execução do programa NC
  - o ciclo deve estar armazenado no diretório _N_CUS_DIR (usuário), _N_CMA_DIR ou _N_CST_DIR
  - para os diretórios _N_CUS_DIR, _N_CMA_DIR e _N_CST_DIR os direitos de execução devem estar ajustados pelo menos para usuário final nos dados de máquina MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA e MD11160 $MN_ACCESS_EXEC_CST, respectivamente
- `APRP 4...7` / `APWP 4...7`
  - para execução do programa NC o seletor com chave deve estar ajustado na posição 3 ... 0
  - o ciclo deve estar armazenado no diretório _N_CUS_DIR, _N_CMA_DIR ou _N_CST_DIR
  - para os diretórios _N_CUS_DIR, _N_CMA_DIR e _N_CST_DIR os direitos de execução devem estar ajustados pelo menos de acordo com a posição do seletor com chave nos dados de máquina MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA e MD11160 $MN_ACCESS_EXEC_CST, respectivamente

Direitos de acesso relacionados ao BTSS (**APRB, APWB**)

Os direitos de acesso (`APRB`, `APWB`) restringem de modo uniforme o acesso às variáveis de sistema e de usuário através do BTSS para todos componentes do sistema (HMI, PLC, processadores externos, serviços EPS, etc.).

Indicação

Direitos de acesso locais da HMI

Para alterações dos direitos de acesso aos dados de sistema deve-se atentar para que estas sejam realizadas de modo consistente em relação aos direitos de acesso definidos através dos mecanismos da HMI.

Atributos de acesso **APR / APW**

Por motivos de incompatibilidade os atributos `APR` e `APW` são refletidos implicitamente nos atributos `APRP` / `APRB` e `APWP` / `APWB`:

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

direitos de acesso através de arquivos ACCESS

No caso de utilização de arquivos ACCESS para atribuição de direitos de acesso, as redefinições de direitos de acesso para dados de sistema, dados de usuário e comandos de linguagem NC somente poderão ocorrer nestes arquivos ACCESS. Uma exceção são os dados de usuário globais (GUD). Para estes, uma redefinição dos direitos de acesso deverá ser mantida nos arquivos de definição correspondentes * _DEF.

Para uma plena proteção de acesso os dados de máquina devem ser adaptados de modo consistente nos respectivos diretórios para os direitos de execução e para a proteção de acesso.

Adota-se o seguinte procedimento básico:

1. Criação dos arquivos de definição necessários:
   - _N_DEF_DIR/_N_SACCESS_DEF
   - _N_DEF_DIR/_N_MACCESS_DEF
   - _N_DEF_DIR/_N_UACCESS_DEF
2. Parametrização do direito de gravação para os arquivos de definição no valor necessário para a redefinição:
   - MD11170 $MN_ACCESS_WRITE_SACCESS = <nível de proteção>
   - MD11171 $MN_ACCESS_WRITE_MACCESS = <nível de proteção>
   - MD11172 $MN_ACCESS_WRITE_UACCESS = <nível de proteção>
3. Para acessar os elementos de ciclos protegidos deve ser realizada a adaptação dos direitos de execução e de gravação dos diretórios de ciclos _N_CST_DIR, _N_CMA_DIR e _N_CST_DIR:
   Direitos de execução
   - MD11160 $MN_ACCESS_EXEC_CST = <nível de proteção>
   - MD11161 $MN_ACCESS_EXEC_CMA = <nível de proteção>
   - MD11162 $MN_ACCESS_EXEC_CUS = <nível de proteção>

   Direitos de gravação
   - MD11165 $MN_ACCESS_WRITE_CST = <nível de proteção>
   - MD11166 $MN_ACCESS_WRITE_CMA = <nível de proteção>
   - MD11167 MN_ACCESS_WRITE_CUS = <nível de proteção>

   O direito de execução deve ser passado pelo menos para o mesmo nível de proteção como o nível de proteção mais alto do elemento utilizado.
   O direito de gravação deve ser passado pelo menos para o mesmo nível de proteção como o direito de execução.
4. Os direitos de gravação dos diretórios de ciclos locais da HMI devem ser passados para o mesmo nível de proteção como o dos diretórios de ciclos locais do NC.
   Literatura
   Manual de operação

Chamadas de sub-rotinas em arquivos ACCESS

Para uma expansão da estruturação da proteção de acesso nos arquivos ACCESS também podem ser chamadas sub-rotinas (extensão SPF ou MPF). Neste caso as sub-rotinas herdam os direitos de execução do arquivo ACCESS chamado.

Indicação

Nos arquivos ACCESS somente podem ser redefinidos os direitos de acesso. Todos demais atributos ainda devem programados e redefinidos nos respectivos arquivos de definição.

Veja também

Variáveis (Página 23)

2.1.10 Atributo: Classe de dados (DCM, DCI, DCU) - apenas SINUMERIK 828D

Para simplificar o manuseio de dados no âmbito da colocação em operação, da colocação em operação em série e da atualização das máquinas e séries de máquinas, todos os dados de sistema e de usuário do NC são divididos em classes de dados.

Classe de dados	Dados
S = Sistema	Dados de sistema disponibilizados pela Siemens, tais como dados de máquinas e de configurações, ciclos padrão e de medição, definições (SGUD) e Macros (SMAC), etc.
M = Manufacturer (fabricante da máquina)	Os dados de colocação em operação específicos da série de máquinas, tais como ciclos de fabricação, definições (MGUD) e macros (MMAC) e dados de máquinas, que definem o âmbito de funções da máquina.
I = Individual (específico da máquina)	Dados de colocação em operação específicos da máquina, tais como dados de compensação e deslocamentos do ponto de referência.
U = User (Usuário)	Os dados específicos de máquinas, que são criados durante a operação da máquina, tais como dados de ferramenta, dados de configuração, programas de peças, ciclos do usuário, definições (MGUD) e macros (UMAC).

Literatura:
SINUMERIK 828D Manual de colocação em funcionamento de torno e fresa; cap.: "Introdução e aplicação das classes de dados"

Definição (DEF) de dados do usuário

Através da classe de dados do arquivo ou do diretório, no qual é definido um dado de usuário, é determinada de modo implícito a classe de dados do dado. A classe de dados do dado não pode ser alterada.

No caso da definição (`DEF`) do dado do usuário porém, pode ser predefinida para o **valor do dado** uma classe de dados própria, diferente da classe de dados do dado.

Para a classe de dados do valor do dado deve aplicar-se:

Prioridade da classe de dados do valor do dado ≤ prioridade da classe de dados do dado

Exemplo:

A definição de GUD, que descreve um sensor de medição, deve situar-se na classe de dados M (= Manufacturer), visto ser necessário para o decurso dos ciclos de fabricação. O valor do dado porém deve pertencer à classe de dados I (Individual), já que o tipo de sensor de medição pode ser diferente de máquina para máquina.

MGUD.DEF (classe de dados M)

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

Redefinição (REDEF) dos dados de sistema

A classe de dados dos dados de sistema pode ser alterada através da redefinição (REDEF). A redefinição deve ocorrer em um arquivo de definição com a classe de dados S ou M.

No caso de utilização de arquivos ACCESS, as redefinições devem ocorrer apenas no interior dos arquivos ACCESS.

A respectiva classe de dados dos dados de máquinas, de configuração e de opções, assim como as variáveis do sistema pode ser encontrada em:

- Lista de manuais da descrição detalhada dos dados de máquinas, parâmetros: "Classe"
- Lista de manuais Variáveis de sistema

2.1.11 Visão geral dos atributos definíveis e redefiníveis

As seguintes tabelas mostram quais atributos podem ser definidos (DEF) e/ou redefinidos (REDEF) em quais tipos de dados.

Dados do sistema

Tipo de dados	Valor inic.	Valores limite	Unidade física	Direitos de acesso	Classe de dados (apenas 828D)
Dados de máquina	---	---	---	REDEF	REDEF
Dados de ajuste	REDEF	---	---	REDEF	---
Dados de FRAME	---	---	---	REDEF	---
Dados de processo	---	---	---	REDEF	---
Compensação de erro de passo de fuso. (EEC)	---	---	---	REDEF	---
Compensação cruzada (CEC)	---	---	---	REDEF	---
Compensação de erro de quadrante (QEC)	---	---	---	REDEF	---
Dados de magazine	---	---	---	REDEF	---
Dados de ferramenta	---	---	---	REDEF	---

Tipo de dados	Valor inic.	Valores limite	Unidade física	Direitos de acesso	Classe de dados (apenas 828D)
Áreas de proteção	---	---	---	REDEF	---
Porta-ferramenta orientável	---	---	---	REDEF	---
Cadeias cinemáticas	---	---	---	REDEF	---
Áreas de proteção 3D	---	---	---	REDEF	---
Limite de área de trabalho	---	---	---	REDEF	---

Dados de usuário

Tipo de dados	Valor inic.	Valores limite	Unidade física	Direitos de acesso	Classe de dados
Parâmetros R	REDEF	REDEF	REDEF	REDEF	---
Variável de ação síncrona ($AC_...)	REDEF	REDEF	REDEF	REDEF	---
GUD de ação síncrona (SYG_...)	REDEF	REDEF	REDEF	REDEF	---
Parâmetros EPS	REDEF	REDEF	REDEF	REDEF	---
Dados de ferramenta – OEM	REDEF	REDEF	REDEF	REDEF	---
Dados de magazine – OEM	REDEF	REDEF	REDEF	REDEF	---
Variáveis de usuário globais (GUD)	DEF / REDEF	DEF	DEF	DEF / REDEF	DEF / REDEF
Variáveis de usuários locais (PUD / LUD)	DEF	DEF	DEF	---	---

Veja também

Variáveis (Página 23)

2.1.12 Definição e inicialização de variáveis de campo (DEF, SET, REP)

Uma variável de usuário pode ser definida como campo (Array) de 1 até 3 dimensões:

- 1 dimensão: `DEF <tipo de dado> <nome da variável>[<n>]`
- 2 dimensões: `DEF <tipo de dado> <nome da variável>[<n>,<m>]`
- 3 dimensões: `DEF <tipo de dado> <nome da variável>[<n>,<m>,<o>]`

Indicação

As variáveis de usuários do tipo de dados STRING podem ser definidas no máximo como campo de 2 dimensões.

Tipos de dados

As variáveis de usuários podem ser definidas como campos para os seguintes tipos de dados: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Atribuição de valores nos elementos de campo

As atribuições de valores nos elementos de campo podem ser realizadas nas seguintes ocasiões:

- Para a definição de campo (valores de inicialização)
- Durante a execução do programa

Aqui a atribuição de valores pode ser realizada através de:

- Especificação explícita de um elemento de campo
- Especificação explícita de um elemento de campo como elemento de partida e especificação de uma lista de valores (`SET`)
- Especificação explícita de um elemento de campo como elemento de partida e especificação de um valor e a frequência de sua repetição (`REP`)

Indicação

As variáveis de usuários do tipo de dados FRAME não podem ser atribuídas com valores de inicialização.

Sintaxe (`DEF`)

```
DEF <tipo de dado> <nome da variável>[<n>,<m>,<o>]
DEF STRING[<tamanho de String>] <nome de variável>[<n>,<m>]
```

Sintaxe (`DEF...=SET...`)

Utilização de uma lista de valores:

- Para a definição:
  ```
  DEF <tipo de dado> <nome de variável>[<n>,<m>,<o>] =
  SET(<valor1>,<valor2>,...)
  ```
  Equivalente com:
  ```
  DEF <tipo de dado> <nome de variável>[<n>,<m>,<o>] =
  (<valor1>,<valor2>,...)
  ```

Indicação

A especificação do `SET` é opcional para a inicialização através de uma lista de valores .

- Para uma atribuição de valores:
  ```
  <nome da variável>[<n>,<m>,<o>]=SET(<VALOR1>,<valor2>,...)
  ```

Sintaxe (`DEF...=REP...`)

Utilização de um valor com repetição

- Para a definição:
  ```
  DEF <tipo de dado> <nome da variável>[<n>,<m>,<o>]=REP(<valor>)
  DEF <tipo de dado> <nome de variável>[<n>,<m>,<o>] =
  REP(<valor>,<número_elementos_de_campo>)
  ```
- Para uma atribuição de valores:
  ```
  <nome da variável>[<n>,<m>,<o>]=REP(<valor>)
  <nome da
  variável>[<n>,<m>,<o>]=REP(<valor>,<número_de_elementos_de_campo>)
  ```

Significado

`DEF`:	Comando para definição de variáveis	
`<tipo de dado>`:	Tipo de dados das variáveis	
	Faixa de valores: • Para variáveis de sistema: BOOL, CHAR, INT, REAL, STRING, AXIS • Para variáveis GUD ou LUD: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME	
`<tamanho de String>`:	Número máximo de caracteres para o tipo de dado STRING	
`<nome de variável>`:	Nome de variável	
`[<n>,<m>,<o>]`:	Tamanhos de campo e índices de campo	
`<n>`:	Tamanho e índice de campo para 1ª dimensão	
	Tipo:	INT (também AXIS em variáveis de sistema)
	Faixa de valores:	Tamanho de campo máx.: 65535 Índice de campo: 0 ≤ n ≤ 65534
`<m>`:	Tamanho e índice de campo para 2ª dimensão	
	Tipo:	INT (também AXIS em variáveis de sistema)
	Faixa de valores:	Tamanho de campo máx.: 65535 Índice de campo: 0 ≤ m ≤ 65534
`<o>`:	Tamanho e índice de campo para 3ª dimensão	
	Tipo:	INT (também AXIS em variáveis de sistema)
	Faixa de valores:	Tamanho de campo máx.: 65535 Índice de campo: 0 ≤ o ≤ 65534
`SET`:	Atribuição de valores através da lista de valores indicada	
`(<valor1>,<valor2>,...)`:	Lista de valores	
`REP`:	Atribuição de valores através do `<valor>` indicado	

<valor>:	Valor com que os elementos de campo devem ser escritos na inicialização com REP.
<número_de_elementos_de_campo>:	Número de elementos de campo que devem ser preenchidos com o <valor> indicado. Para os elementos de campo restantes a aplicação depende do momento: • Inicialização na definição de campo: → Os elementos de campo restantes são preenchidos com zero • Atribuição durante a execução do programa: → Os atuais valores dos elementos de campo permanecem inalterados. Se o parâmetro não estiver programado, todos elementos de campo são preenchidos com o <valor>. Se o parâmetro for igual a zero, a aplicação depende do momento: • Inicialização na definição de campo: → Todos elementos são ocupados previamente com zero • Atribuição durante a execução do programa: → Os atuais valores dos elementos de campo permanecem inalterados.

Índice de campo

A ordem implícita dos elementos de campo, por exemplo no caso de uma atribuição de valores através de SET ou REP, é realizada através da iteração do índice de campo da direita para a esquerda.

Exemplo: Inicialização de um campo de 3 dimensões com 24 elementos de campo:

```
DEF INT CAMPO[2,3,4] = REP(1,24)
  CAMPO[0,0,0] = 1     1. Elemento de campo
  CAMPO[0,0,1] = 1     2. Elemento de campo
  CAMPO[0,0,2] = 1     3. Elemento de campo
  CAMPO[0,0,3] = 1     4. Elemento de campo
  ...
  CAMPO[0,1,0] = 1     5. Elemento de campo
  CAMPO[0,1,1] = 1     6. Elemento de campo
  ...
  CAMPO[0,2,3] = 1     12. Elemento de campo
  CAMPO[1,0,0] = 1     13. Elemento de campo
  CAMPO[1,0,1] = 1     14. Elemento de campo
  ...
  CAMPO[1,2,3] = 1     24. Elemento de campo
```

de acordo com:

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

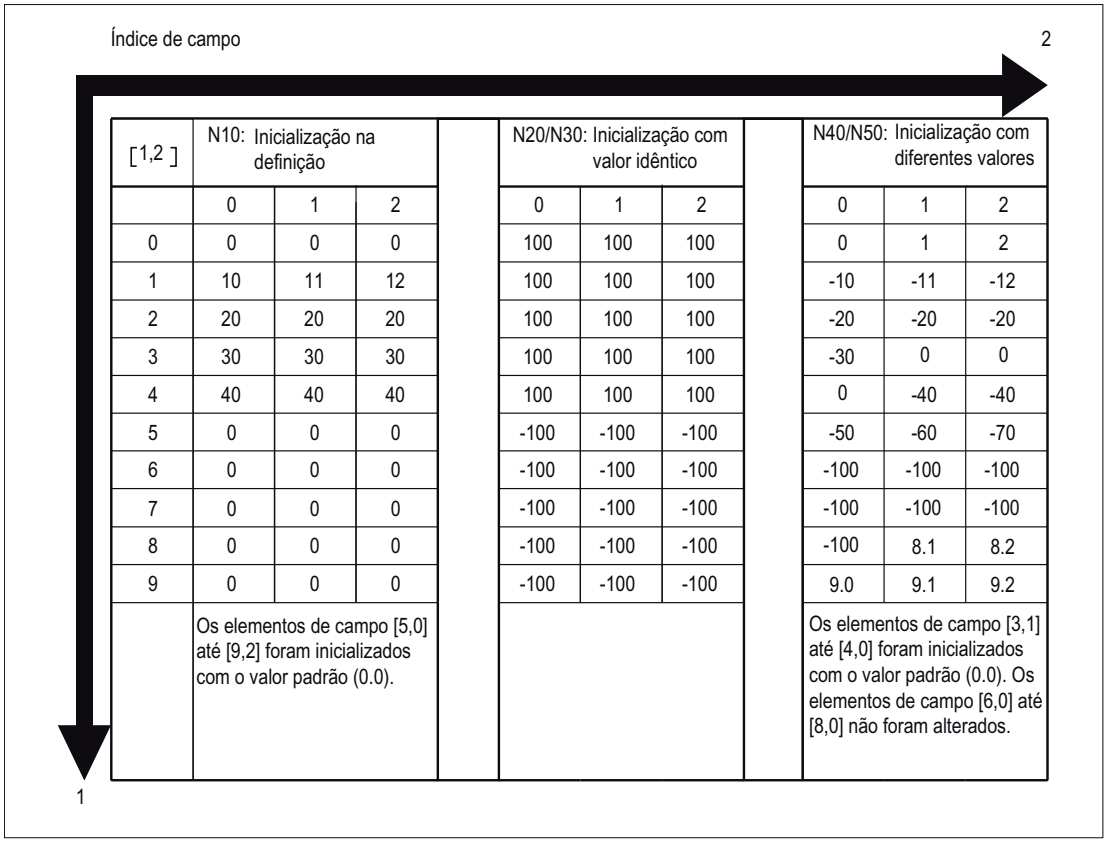
```
    FOR o=0 TO 3
      CAMPO[n,m,o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

Exemplo: Inicialização de campos de variáveis completos

Veja a atual ocupação na figura.

Código de programa

```
N10 DEF REAL CAMPO1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)
N20 CAMPO1[0,0]=REP(100)
N30 CAMPO1[5,0]=REP(-100)
N40 CAMPO1[0,0]=SET(0,1,2,-10,-11,-12,-20,-20,-20,-30, , , ,-40,-40,-50,-60,-70)
N50 CAMPO1[8,1]=SET(8.1,8.2,9.0,9.1,9.2)
```



[1,2]	N10: Inicialização na definição			N20/N30: Inicialização com valor idêntico			N40/N50: Inicialização com diferentes valores		
	0	1	2	0	1	2	0	1	2
0	0	0	0	100	100	100	0	1	2
1	10	11	12	100	100	100	-10	-11	-12
2	20	20	20	100	100	100	-20	-20	-20
3	30	30	30	100	100	100	-30	0	0
4	40	40	40	100	100	100	0	-40	-40
5	0	0	0	-100	-100	-100	-50	-60	-70
6	0	0	0	-100	-100	-100	-100	-100	-100
7	0	0	0	-100	-100	-100	-100	-100	-100
8	0	0	0	-100	-100	-100	-100	8.1	8.2
9	0	0	0	-100	-100	-100	9.0	9.1	9.2
	Os elementos de campo [5,0] até [9,2] foram inicializados com o valor padrão (0.0).						Os elementos de campo [3,1] até [4,0] foram inicializados com o valor padrão (0.0). Os elementos de campo [6,0] até [8,0] não foram alterados.		

Veja também

Definição e inicialização de variáveis de campo (DEF, SET, REP): Outras informações (Página 60)

Variáveis (Página 23)

2.1.13 Definição e inicialização de variáveis de campo (DEF, SET, REP): Outras informações

Outras informações (SET)

Inicialização na definição

- Iniciando-se pelo 1º elemento de campo, são inicializados tantos elementos de campo com os valores da lista de valores como foram programados elementos na lista de valores.
- Os elementos de campo sem valores indicados de modo explícito na lista de peças (brechas na lista de valores) são preenchidos com 0.
- Para variáveis do tipo de dado AXIS não são permitidas brechas na lista de valores.
- Se a lista de valores contém mais valores como do que elementos de campo definidos, aparecerá um alarme.

Atribuição de valores na execução do programa

Para a atribuição de valores na execução do programa são aplicadas as regras descritas acima para a definição. Além disso existem as seguintes possibilidades:

- Como elementos na lista de valores também é permitido o uso de expressões.
- A atribuição de valores inicia com o índice de campo programado. Dessa forma os campos parciais são ocupados de modo controlado.

Exemplo:

Código de programa	**Comentário**
`DEF INT CAMPO[5,5]`	`; Definição de campo`
`CAMPO[0,0]=SET(1,2,3,4,5)`	`; Atribuição de valores nos primeiros 5 elementos de campo [0,0] - [0,4]`
`CAMPO[0,0]=SET(1,2, , ,5)`	`; Atribuição de valores com lacuna nos primeiros 5 elementos de campo [0,0] - [0,4], elementos de campo [0,2] e [0,3] = 0`

`CAMPO[2,3]=SET(VARIÁVEL,4*5.6)`	`; Atribuição de valores com variável e expressão a partir do índice de campo [2,3]:`
	`[2,3] = VARIÁVEL`
	`[2,4] = 4 * 5.6 = 22.4`

Outras informações (REP)

Inicialização na definição

- Todos os elementos ou a quantidade opcional indicada de elementos de campo é iniciada com o valor (constante) indicado.
- As variáveis do tipo de dado FRAME não podem ser inicializadas.

Exemplo:

Código de programa	Comentário
`DEF REAL varName[10]=REP(3.5,4)`	`; Inicializar a definição de campo e os elementos de campo [0] a [3] com o valor 3,5`

Atribuição de valores na execução do programa

Para a atribuição de valores na execução do programa são aplicadas as regras descritas acima para a definição. Além disso existem as seguintes possibilidades:

- Como elementos na lista de valores também é permitido o uso de expressões.
- A atribuição de valores inicia com o índice de campo programado. Dessa forma os campos parciais são ocupados de modo controlado.

Exemplos:

Código de programa	Comentário
`DEF REAL varName[10]`	`; Definição de campo`
`varName[5]=REP(4.5,3)`	`; Elementos de campo [5] a [7] = 4,5`
`R10=REP(2.4,3)`	`; Parâmetros R10 até R12 = 2,4`
`DEF FRAME FRM[10]`	`; Definição de campo`
`FRM[5]=REP(CTRANS(X,5))`	`; Elementos de campo [5] a [9] = CTRANS(X,5)`

Veja também

Definição e inicialização de variáveis de campo (DEF, SET, REP) (Página 55)

2.1.14 Tipos de dados

No NC estão disponíveis os seguintes tipos de dados:

Tipo de dado	Significado	Faixa de valores
INT	Valor inteiro com sinal	-2147483648 ... +2147483647
REAL	Número real (LONG REAL conforme IEEE)	±(~2,2*10^{-308} … ~1,8*10^{+308})
BOOL	Valor lógico TRUE (1) e FALSE (0)	1, 0
CHAR	Caracteres ASCII	Código ASCII 0 ... 255
STRING	Sequência de caracteres de tamanho definido	máximo 200 caracteres (sem caractere especial)

AXIS	Identificador de eixo/fuso	Identificador de canal
FRAME	Indicações geométricas para uma transformação estática de coordenadas (deslocamento, rotação, escalonamento, espelhamento)	---

Mudanças de tipo de dado implícitas

As seguintes mudanças de tipo de dado são possíveis e são realizadas de forma implícita nas atribuições e transferências de parâmetros:

de ↓/ para →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x
x: Possível sem restrições o: Possível perda de dados ao exceder a faixa de valores ⇒ Alarme; Arredondamento: Valor após a vírgula ≥ 0,5 ⇒ arredonda para cima, valor após a vírgula < 0,5 ⇒ arredonda para baixo &: Valor ≠ 0 ⇒ TRUE, valor == 0 ⇒ FALSE			

Veja também

Variáveis (Página 23)

2.1.15 examinar a existência de uma variável (ISVAR)

Com a função pré-definida ISVAR é possível examinar se uma variável de sistema / de usuário (ex. dado da máquina, dado de configuração, variável de sistema, variáveis gerais, como GUD) são conhecidas no NC.

Variável

A variável em questão é constituída do seguinte modo:

Variável adimensional: <Variável>
Variável unidimensional sem índice de campo: <Variável>[ ]
Variável unidimensional com índice de campo n: <Variável>[<n>]
Variável bidimensional sem índice de campo: <Variável>[,]
Variável bidimensional com índices de campo n e m: <Variável>[<n>,<m>]

Sintaxe

```
<Resultado>=ISVAR(<Variável>[<n>,<m>])
```

Significado

<Resultado>:	Valor de retorno		
	Tipo de dado:	BOOL	
	Faixa de valores:	1	variável presente
		0	variável desconhecida
ISVAR:	Verificar se a variável de sistema/ usuário informada é conhecida no NC.		
<variável>:	Nome da variável de sistema / de usuário		
	Tipo de dado:	STRING	
<n>:	Índice de campo da primeira dimensão (**opcional**)		
	Tipo de dado:	INT	
<m>:	Índice de campo da segunda dimensão (**opcional**)		
	Tipo de dado:	INT	

De acordo com o parâmetro de transferência são realizadas as seguintes verificações:

- Se o nome é conhecido
- Se a variável é um campo
- Se se tratar de um campo mono- ou bidimensional
- Se o respectivo índice de campo está no intervalo permitido

TRUE (1) somente é emitido se todas as verificações resultarem positivas.

Se uma verificação resultar negativa, ou ocorrer um erro de sintaxe, FALSE (0) é emitido.

Exemplos

Código de programa	**Comentário**
DEF INT VAR1	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VAR1")	; IS_VAR neste caso é TRUE.

Código de programa	**Comentário**
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VARARRAY[,]")	; IS_VAR neste caso é TRUE, é um campo bi-dimensional.
N20 IS_VAR=ISVAR("VARARRAY")	; IS_VAR é TRUE, a variável existe.
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	; IS_VAR é FALSE, índice de campo não é permitido.
N40 IS_VAR=ISVAR("VARARRAY[8,8")	; IS_VAR é FALSE, falta "]" (erro de sintaxe).
N50 IS_VAR=ISVAR("VARARRAY[,8]")	; IS_VAR é TRUE, o índice de campo é permitido.
N60 IS_VAR=ISVAR("VARARRAY[8,]")	; IS_VAR é TRUE, o índice de campo é permitido.

Código de programa	Comentário
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"	; O parâmetro de transferência é um dado da máquina, IS_VAR é TRUE.

Código de programa	Comentário
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("$P_EP")	; IS_VAR neste caso é TRUE.
N20 IS_VAR=ISVAR("$P_EP[X]")	; IS_VAR neste caso é TRUE.

2.1.16 Ler valores de atributo/tipo de dado (GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)

Com as funções pré-definidas GETVARPHU, GETVARAP, GETVARLIM e GETVARDFT é possível ler os atributos de valor das variáveis de sistema / de usuário, com GETVARTYP o tipo de dados de uma variável de sistema / de usuário.

ler unidade física

Sintaxe:
<Resultado>=GETVARPHU(<Nome>)

Significado:

<Resultado>:	Valor numérico da unidade física		
	Tipo de dado:	INT	
	Faixa de valores:	Consulte a tabela em "Atributo: Unidade física (PHU) (Página 45)"	
		Em caso de erro:	
		- 2	O <Name> especificado não se encontra em nenhum parâmetro de sistema nem em nenhuma variável de usuário.
GETVARPHU:	Lê a unidade física de uma variável de sistema/de usuário		
<Nome>:	Nome da variável de sistema / de usuário		
	Tipo de dado:	STRING	

Exemplo:

O NC possui a seguinte variável GUD:

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

Código de programa	Comentário
DEF INT result=0	
result=GETVARPHU("electric")	; Defina a unidade física das variáveis GUD.

Código de programa	Comentário
`IF (result < 0) GOTOF error`	

O resultado será 42. Este corresponde à unidade física [kW].

Indicação

Com GETVARPHU é possível examinar p.ex., se em um acesso à variável a = b ambas variáveis possuem o valor físico esperado.

ler direito de acesso:

Sintaxe:
`<resultado>=GETVARAP(<Nome>,<acesso>)`

Significado:

`<Resultado>`:	Nível de segurança para o <acesso> especificada		
	Tipo de dado:	INT	
	Faixa de valores:	0 ... 7	Consulte "Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 47) ".
		Em caso de erro:	
		- 1	não gravável (somente relevante em <Acesso> "`WP`" e "`WB`")
		- 2	O `<Name>` especificado não se encontra em nenhum parâmetro de sistema nem em nenhuma variável de usuário.
		- 3	valor incorreto para `<acesso>`
`GETVARAP`:	Leitura do direito do acesso em uma variável de sistema/de usuário		
`<Nome>`:	Nome da variável de sistema / de usuário		
	Tipo de dado:	STRING	
`<acesso>`:	tipo de acesso		
	Tipo de dado:	STRING	
	Faixa de valores:	`"RP"`	Leitura através do programa de usinagem
		`"WP"`	Gravação através do programa de usinagem
		`"RB"`	Leitura através do BTSS
		`"WB"`	Gravação através do BTSS

Exemplo:

Código de programa	Comentário
`DEF INT result=0`	
`result=GETVARAP("$TC_MAP8","WB")`	`; Defina a proteção de acesso para o parâmetro de sistema "posição do magazine" referente à gravação através do BTSS.`
`IF (result < 0) GOTOF error`	

O resultado será 7. Este corresponde à posição do interruptor de chave 0 (= sem proteção de acesso).

Indicação

Com GETVARAP é possível, p.ex, realizar um programa de teste, que examinará o direito de acesso da aplicação esperada.

ler valores limite

Sintaxe:
`<Status>=GETVARLIM(<nome>,<valor limite>,<resultado>)`

Significado:

`<Status>`:	Status da função		
	Tipo de dado:	INT	
	Faixa de valores:	1	O. K.
		-1	nenhum valor limite definido (nas variáveis do tipo AXIS, STRING, FRAME)
		-2	O `<Name>` especificado não se encontra em nenhum parâmetro de sistema nem em nenhuma variável de usuário.
		-3	valor incorreto para `<valor limite>`
`GETVARLIM`:	Leitura do valor limite inferior e superior de uma variável de sistema/de usuário		
`<Nome>`:	Nome da variável de sistema / de usuário		
	Tipo de dado:	STRING	
`<valor limite>`:	Informa qual é o valor limite a ser lido		
	Tipo de dado:	CHAR	
	Faixa de valores:	`"L"`:	= Valor limite inferior
		`"U"`:	= Valor limite superior
`<Resultado>`:	Devolução do valor limite		
	Tipo de dado:	VAR REAL	

Exemplo:

Código de programa	**Comentário**
`DEF INT state=0`	
`DEF REAL result=0`	
`state=GETVARLIM("$MA_MAX_AX_VELO","L",result)`	`; Defina o valor limite inferior para MD32000 $MA_MAX_AX_VELO.`
`IF (result < 0) GOTOF error`	

Ler atributo/ tipo de dado

Sintaxe:
```
<resultado>=GETVARDIM( <nome>, Index)
```

Significado:

<Resultado>:	Dimensão/quantidade de campos <Index>	
	Tipo de dado:	INT
GETVARDIM:	Leitura do valor limite inferior e superior de uma variável de sistema/de usuário	
<nome>:	Ler a quantidade de elementos do campo	
	Tipo de dado:	STRING
<Índice>:	Número de campos, máx. 3.	
	Tipo de dado:	INT

Exemplo:

Código de programa	**Comentário**
N5 DEF REAL myReal[5,4]	
N10 R1 = GETVATDIM("myReal",1)	R1 = 5
N15 R2 = GETVATDIM("myReal",2)	R2 = 4

ler valor padrão

Sintaxe:
```
<Status>=GETVARDFT(<nome>,<resultado>[,<índice_1>,<índice_2>,<índice_3>])
```

Significado:

<Status>:	Status da função		
	Tipo de dado:	INT	
	Faixa de valores:	1	O. K.
		-1	nenhum valor padrão disponível p. ex. porque o <Resultado> tem o tipo de <Nome> incorreto)
		-2	O <Name> especificado não se encontra em nenhum parâmetro de sistema nem em nenhuma variável de usuário.
		-3	valor incorreto para <Índice_1>, Dimensão menor que um (= nenhum Array = Skalar)
		-4	valor incorreto para <Índice_2>
		-5	valor incorreto para <Índice_3>
GETVARDFT:	Leitura do valor padrão de uma variável de sistema/de usuário		
<Nome>:	Nome da variável de sistema / de usuário		
	Tipo de dado:	STRING	

<Resultado>:	Devolução do valor padrão	
	Tipo de dado:	VAR REAL (na leitura do valor padrão das variáveis do tipo INT, REAL , BOOL, AXIS)
		VAR STRING (na leitura do valor padrão das variáveis do tipo STRING e CHAR)
		VAR FRAME (na leitura do valor padrão das variáveis do tipo FRAME)
<Índice_1>:	Índice da primeira dimensão (opcional)	
	Tipo de dado:	INT
	não programado significa = 0	
<Índice_2>:	Índice da segunda dimensão (opcional)	
	Tipo de dado:	INT
	não programado significa = 0	
<Índice_3>:	Índice da terceira dimensão (opcional)	
	Tipo de dado:	INT
	não programado significa = 0	

Exemplo:

```
Código de programa                                  Comentário
DEF INT state=0
DEF REAL resultR=0                                  ; variáveis para inclusão dos valores
                                                    padrão dos tipos INT, REAL, BOOL, AXIS.
DEF FRAME resultF=0                                 ; Variáveis para inclusão dos valores
                                                    padrão do tipo FRAME


IF (GETVARTYP("$MA_MAX_AX_VELO") <> 4)
GOTOF error
state=GETVARDFT("$MA_MAX_AX_VELO",                  ; Defina o valor padrão do
resultR, AXTOINT(X))                                eixo "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 )
```

ler tipo de dados

Sintaxe:

```
<resultado>=GETVARTYP(<Nome>)
```

Significado:

<Resultado>:	Tipo de dados das variáveis de sistema / de usuário especificadas		
	Tipo de dado:	INT	
	Faixa de valores:	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		Em caso de erro:	
		< 0	O <Name> especificado não se encontra em nenhum parâmetro de sistema nem em nenhuma variável de usuário.
GETVARTYP:	Leitura do tipo de dados em uma variável de sistema/de usuário		
<Nome>:	Nome da variável de sistema / de usuário		
	Tipo de dado:	STRING	

Exemplo:

Código de programa	**Comentário**
`DEF INT result=0`	
`DEF STRING name="R"`	
`result=GETVARTYP(name)`	`; Define o tipo do parâmetro R.`
`IF (result < 0) GOTOF error`	

O resultado será 4. Este corresponde ao tipo de dado REAL.

2.2 Programação indireta

2.2.1 Programação indireta de endereços

Na programação indireta de endereços o endereço ampliado (<índice>) é substituído por uma variável de tipo adequado.

Indicação

A programação indireta de endereços não é possível com:

- N (número de bloco)
- L (sub-rotina)
- Endereços ajustáveis
  (p. ex. não é permitido X[1] no lugar de X1)

Sintaxe

```
<ENDEREÇO>[<índice>]
```

Significado

<ENDEREÇO>[...]:	Endereço fixo com extensão (índice)
<índice>:	Variável, p. ex. para número de fuso, eixo, ...

Exemplos

Exemplo 1: Programação indireta de um número de fuso

Programação direta:

Código de programa	Comentário
S1=300	; Rotação de 300 rpm para o fuso de número 1.

Programação indireta:

Código de programa	Comentário
DEF INT SPINU=1	; Definição das variáveis do tipo INT e atribuição de valores.
S[SPINU]=300	; Rotação de 300 rpm para o fuso, cujo número está armazenado na variável SPINU (neste exemplo temos o fuso de número 1).

Exemplo 2: Programação indireta de um eixo

Programação direta:

Código de programa	Comentário
FA[U]=300	; Avanço 300 para o eixo "U".

Programação indireta:

Código de programa	Comentário
DEF AXIS AXVAR2=U	; Definição de uma variável do tipo AXIS e atribuição de valores.
FA[AXVAR2]=300	; Avanço 300 para o eixo, cujo nome de endereço está armazenado na variável de nome AXVAR2.

Exemplo 3: Programação indireta de um eixo

Programação direta:

Código de programa	Comentário
$AA_MM[X]	; Ler o valor de medição do apalpador de medição (MCS) do eixo "X".

Programação indireta:

Código de programa	Comentário
DEF AXIS AXVAR3=X	; Definição de uma variável do tipo AXIS e atribuição de valores.
$AA_MM[AXVAR3]	; Ler o valor de medição do apalpador de medição (MCS) do eixo, cujo nome está armazenado na variável AXVAR3.

Exemplo 4: Programação indireta de um eixo

Programação direta:

Código de programa
X1=100 X2=200

Programação indireta:

Código de programa	Comentário
DEF AXIS AXVAR1 AXVAR2	; Definição de duas variáveis do tipo AXIS.
AXVAR1=(X1) AXVAR2=(X2)	; Atribuição dos nomes de eixo.
AX[AXVAR1]=100 AX[AXVAR2]=200	; Deslocamento dos eixos, cujos nomes de endereço estão armazenados nas variáveis de nome AXVAR1 e AXVAR2.

Exemplo 5: Programação indireta de um eixo

Programação direta:

Código de programa
G2 X100 I20

Programação indireta:

Código de programa	Comentário
DEF AXIS AXVAR1=X	; Definição de uma variável do tipo AXIS e atribuição de valores.

Código de programa	Comentário
`G2 X100 IP[AXVAR1]=20`	`; Programação indireta da indicação do centro para o eixo, cujo nome de endereço está armazenado na variável de nome AXVAR1`

Exemplo 6: Programação indireta de elementos de campo

Programação direta:

Código de programa	Comentário
`DEF INT CAMPO1[4,5]`	`; Definição do campo 1.`

Programação indireta:

Código de programa	Comentário
`DEFINE DIM1 AS 4`	`; No caso das dimensões de campo os tamanhos de campo devem ser especificados como valores fixos.`
`DEFINE DIM2 AS 5`	
`DEF INT CAMPO[DIM1,DIM2]`	
`CAMPO[DIM1-1,DIM2-1]=5`	

Exemplo 7: Chamada de sub-rotina indireta

Código de programa	Comentário
`CALL "L" << R10`	`; Chamada do programa, cujo número está no R10 (encadeamento de string).`

2.2.2 Programação indireta de comandos G

A programação indireta de comandos G possibilita uma programação efetiva de ciclos.

Sintaxe

```
G[<grupo>]=<número>
```

Significado

`G[...]`:	Comando G com extensão (índice)	
`<grupo>`:	Parâmetro de índice: Grupo G	
	Tipo:	INT
`<número>`:	Variável para o número de comando G	
	Tipo:	INT ou REAL

Indicação

Geralmente somente podem ser programados de forma indireta os comandos G que não são determinantes na sintaxe.

Dos comandos G que são determinantes na sintaxe, apenas são possíveis aqueles do grupo 1 de G.
Os comandos G determinantes de sintaxe dos grupos 2, 3 e 4 de G não são aplicáveis.

Indicação

Na programação indireta de comando G não é permitida nenhuma função aritmética. Um cálculo necessário do número de comando G devem ser realizados em uma linha própria do programa de peça, ainda antes da programação indireta do comando G.

Exemplos

Exemplo 1: Deslocamento de ponto zero ajustável (grupo 8 de G)

```
Código de programa                    Comentário
N1010 DEF INT INT_VAR
N1020 INT_VAR=2
...
N1090 G[8]=INT_VAR G1 X0 Y0           ; G54
N1100 INT_VAR=INT_VAR+1               ; cálculo do comando G
N1110 G[8]=INT_VAR G1 X0 Y0           ; G55
```

Exemplo 2: Seleção de plano (grupo 6 de G)

```
Código de programa                    Comentário
N2010 R10=$P_GG[6]                    ; ler comando ativo de G do grupo 6 de G
...
N2090 G[6]=R10
```

Literatura

Para informações sobre os grupos de G, veja:
manual de programação Fundamentos; capítulo "Grupos de G"

2.2.3 Programação indireta de atributos de posição (GP)

Os atributos de posição, como a programação incremental ou absoluta da posição do eixo, podem ser programados indiretamente como variáveis em conjunto com a palavra-chave `GP`.

Aplicação

A programação indireta de atributos de posição tem aplicação nos **ciclos de substituição**, tendo a seguinte vantagem sobre a programação de atributos de posição como palavra-chave (p. ex. IC, AC, ...):

Através da programação indireta como variáveis **não** será necessária nenhuma instrução CASE que se bifurca através de todos possíveis atributos de posição.

Sintaxe

```
<COMANDO_POSICIONAMENTO>[<eixo/fuso>]=
GP(<posição>,<atributo de posição)
<eixo/fuso>=GP(<posição>,<atributo de posição>)
```

Significado

<COMANDO_POSICIONAMENTO>[]:	Os seguintes comandos de posicionamento podem ser programados juntos com a palavra-chave GP: POS, POSA, SPOS, SPOSA Além disso também é possível: • todos identificadores de eixo/fuso disponíveis no canal: <eixo/fuso> • identificador variável de eixo/fuso AX
<eixo/fuso>:	Eixo/fuso que deve ser posicionado
GP():	Palavra-chave para posicionamento
<posição>:	Parâmetro 1 Posição de eixo/fuso como constante ou variável
<atributo de posição>:	Parâmetro 2 Atributo de posição (p. ex. modo de aproximação da posição) como variável (p. ex. $P_SUB_SPOSMODE) ou como palavra-chave (IC, AC, ...)

Os valores retornados das variáveis possuem o seguinte significado:

Valor	Significado	Permitido para:
0	Nenhuma alteração do atributo de posição	
1	AC	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
2	IC	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
3	DC	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
4	ACP	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
5	ACN	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
6	OC	-
7	PC	-
8	DAC	POS, POSA, AX, endereço de eixo
9	DIC	POS, POSA, AX, endereço de eixo
10	RAC	POS, POSA, AX, endereço de eixo

Valor	Significado	Permitido para:
11	RIC	POS, POSA, AX, endereço de eixo
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

Exemplo

Em um acoplamento ativo de fusos sincronizados entre o fuso mestre S1 e o fuso escravo S2, através do comando SPOS no programa principal é realizada a chamada do seguinte ciclo de substituição para posicionamento dos fusos.

O posicionamento é realizado através da instrução no N2230:
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE) SPOS[2]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)

A posição a ser aproximada é lida da variável de sistema $P_SUB_SPOSIT, o modo de aproximação da posição da variável de sistema $P_SUB_SPOSMODE.

```
Código de programa                          Comentário
N1000 PROC LANG_SUB DISPLOF SBLOF
...
N2100 IF($P_SUB_AXFCT==2)
N2110                                       ; Substituição do comando SPOS / SPO-
                                            SA / M19 com o acoplamento do fuso sincro-
                                            nizado ativo
N2185 DELAYFSTON                            ; Início da área stop-delay
N2190 COUPOF(S2,S1)                         ; Desativar o acoplamento de fusos sincro-
                                            nizados
N2200                                       ; Posicionar fuso mestre e fuso escravo
N2210 IF($P_SUB_SPOS==TRUE) OR
($P_SUB_SPOSA==TRUE)
N2220                                       ; Posicionar fuso com SPOS:
N2230    SPOS[1]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
         SPOS[2]=GP($P_SUB_SPOSIT,
$P_SUB_SPOSMODE)
N2250 ELSE
N2260                                       ; Posicionar fuso com M19:
N2270 M1=19 M2=19                           ; Posicionar fuso mestre e fuso escravo
N2280 ENDIF
N2285 DELAYFSTOF                            ; Fim da área stop-delay
N2290 COUPON(S2,S1)                         ; Ativar acoplamento de fusos sincroniza-
                                            dos
N2410 ELSE
N2420                                       ; Consulta quanto a outras substituições
...
```

Código de programa	Comentário
N3300 ENDIF	
...	
N9999 RET	

Condições gerais

- A programação indireta de atributos de posição não é possível em ações sincronizadas.

Literatura

Manual de funcionamento das funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1),
Capítulo: Substituição de funções NC através de sub-rotinas

2.2.4 Programação indireta de linhas de programa de peça (EXECSTRING)

Com o comando de programa de peça `EXECSTRING` é possível executar uma variável de String, que foi gerada anteriormente, como uma linha do programa de peça.

Sintaxe

`EXECSTRING` é programado em uma linha de programa de peça própria:
`EXECSTRING(<variável String>)`

Significado

`EXECSTRING`:	Comando para execução de uma variável de String como linha de programa de peça
`<variável String>`:	Variável do tipo STRING que contém a própria linha de programa de peça que deve ser executada

Indicação

Com o `EXECSTRING` podem ser, com exceção das Estruturas de controle (Página 119), descartadas todas as construções de programa de peças, que podem ser programadas **em uma parte do programa de peças**. Excluídas disso estão as instruções `PROC` e `DEF`, bem como o uso frequente em arquivos INI e DEF.

Exemplo

Código de programa	Comentário
N100 DEF STRING[100] MY_BLOCK	; Definição das variáveis de String para gravar as linhas de programa de peça que devem ser executadas.
N110 DEF STRING[10] MFCT1="M7"	

Código de programa	**Comentário**
...	
N200 EXECSTRING(MFCT1 << "M4711")	; Execução da linha de programa de peça "M7 M4711".
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF(R10)	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(MY_BLOCK)	; Execução da linha de programa de peça "M3 M7".

2.3 Funções de cálculo

Operador / Função de cálculo	Significado
+	Adição
–	Subtração
*	Multiplicação
/ [1]	Divisão [1]
DIV [1]	Divisão por número inteiro [1]
MOD [1]	Divisão de módulo (fornece o resto da divisão por número inteiro) [1]
:	Operador de concatenação para as variáveis FRAME
SIN( )	Seno
COS()	Cosseno
TAN()	Tangente
ASIN()	Arco seno
ACOS()	Arco cosseno
ATAN2(,) [1]	Arco tangente 2 [1]
SQRT()	Raiz quadrada
ABS()	Valor
POT()	2ª Potência (quadrado)
TRUNC()	Parte inteira Precisão em comandos de comparação ajustável com TRUNC (veja "Correção da precisão em erros de comparação (TRUNC) (Página 83)")
ROUND()	Arredondamento para um número inteiro
LN()	Logaritmo natural
EXP()	Função exponencial
MINVAL()	menor valor de duas variáveis (veja "Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 85)")
MAXVAL()	Maior valor de duas variáveis (veja "Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 85)")
BOUND()	Valor de variável que está na faixa de valores definida (veja "Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 85)")
CTRANS()	Deslocamento
CROT()	Rotação
CSCALE()	Alteração de escala
CMIRROR()	Espelhamento
1) Consultar o parágrafo "Exemplos"	

Programação

Nas funções de cálculo se aplica a forma escrita usual de matemática. As prioridades na execução são definidas através de parênteses. Para as funções trigonométricas e suas funções inversas se aplica a indicação em graus (ângulo reto = 90°).

Exemplos

Divisão: /

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

Exemplo: 3 / 4 = 0.75

Divisão de número inteiro: DIV

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

Exemplo: 7 DIV 4.1 = 1

Divisão de módulo (fornece o resto da divisão por número inteiro): MOD

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

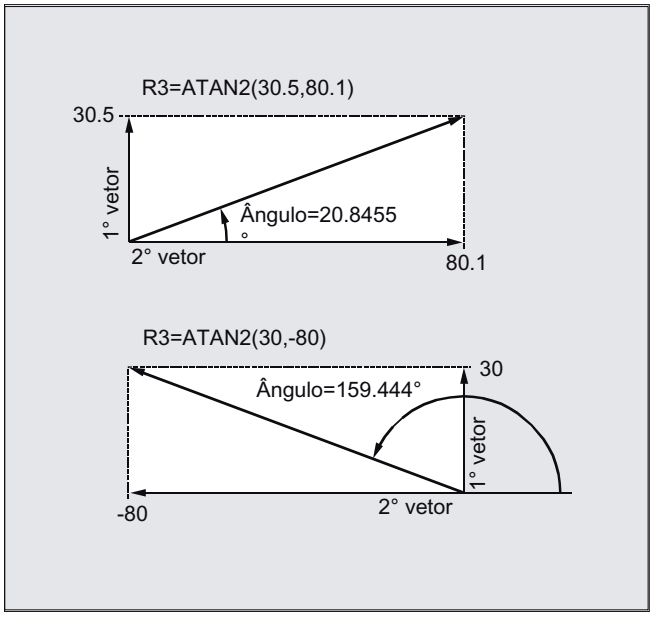
Exemplo: 7 MOD 4.1 = 2.9

Arco tangente 2: ATAN2

A função de cálculo ATAN2 calcula o ângulo do vetor formado por dois vetores posicionados perpendicularmente.

O resultado está na área de quatro quadrantes (-180° < 0 < +180°).

A base para a referência de ângulo sempre é o 2º valor em sentido positivo.



Exemplos de programação

Código de programa	**Comentário**
R1=R1+1	; Novo R1 = antigo R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; O cálculo com pontos tem prioridade sobre o cálculo com traços.
R14=(R1+R2)*R3	; as expressões entre parênteses/colchetes são calculadas primeiro.
R15=SQRT(POT(R1)+POT(R2))	; os parênteses internos são resolvidos primeiro: R15 = raiz quadrada ((R1^2 + R2^2))
RESFRAME=FRAME1:FRAME2	; interligação FRAME com operador de concatenação
FRAME3=CTRANS(…):CROT(…)	; Atribuição de valor a um componente FRAME

2.4 Operadores de comparação e operadores lógicos

Operações de comparação podem ser usadas, por exemplo, para formulação de uma condição de salto. Também podem ser comparadas expressões complexas.

As operações de comparação são aplicáveis para variáveis do tipo `CHAR, INT, REAL` e `BOOL`. No tipo `CHAR` é comparado o valor do código.
Nos tipos `STRING`, `AXIS` e `FRAME` são possíveis: == e <>, que também podem ser aplicados em operações do tipo `STRING` em ações sincronizadas.

O resultado das operações comparadas sempre será do tipo `BOOL`.

Operadores lógicos servem para ligação de valores lógicos.

As operações lógicas somente podem ser aplicadas em variáveis do tipo `BOOL`. Através da conversão interna de tipos elas também podem ser aplicadas nos tipos de dados `CHAR, INT` e `REAL`.

Nas operações lógicas (boolschen) o seguinte é valido para os tipos de dados `BOOL`, `CHAR, INT` e `REAL`:

- 0 corresponde a: FALSE
- diferente de 0 corresponde a: TRUE

Operadores lógicos por Bits

Com as variáveis do tipo `CHAR` e `INT` também podem ser empregados operadores lógicos por Bits. Eventualmente é realizada uma conversão automática de tipos.

Programação

Operador de comparação	Significado
==	igual
<>	diferente
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual

Operador lógico	Significado
`AND`	E
`OR`	OU
`NOT`	Negação
`XOR`	OU exclusivo

Operador lógico por Bits	Significado
`B_AND`	E bit a bit
`B_OR`	OU bit a bit
`B_NOT`	Negação bit a bit
`B_XOR`	OU exclusivo bit a bit

Indicação

Em expressões aritméticas se pode definir a ordem de execução através de parênteses, e com isso, fugir das regras convencionais de prioridade.

Indicação

Entre operandos e operadores BOOLEANOS devem ser escritos espaços.

Indicação

O operador `B_NOT` refere-se apenas a um operando. Este se encontra após o operador.

Exemplos

Exemplo 1: Operadores de comparação

```
IF R10>=100 GOTOF DESTINO
```

ou

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

O resultado da comparação `R10>=100` primeiro é armazenado temporariamente no `R11`.

Exemplo 2: Operadores lógicos

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

ou

```
IF NOT R10 GOTOB PARTIDA
```

`NOT` refere-se apenas a um operando.

Exemplo 3: Operadores lógicos por Bits

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

2.5 Correção da precisão em erros de comparação (TRUNC)

O comando TRUNC corta o operando multiplicado por um fator de precisão.

Precisão ajustável em comandos de comparação

Os dados de programa de peça do tipo REAL são representados internamente em formato IEEE com 64 Bits. Devido a esta forma de representação as casas decimais podem ser imprecisas, o que pode gerar resultados inesperados se compararmos aos valores calculados.

Igualdade relativa

Para que a precisão ressaltada através da forma de representação não falsifique o fluxo do programa, então com os comandos de comparação não é realizado o controle sobre a igualdade absoluta, e sim sobre a igualdade relativa.

Sintaxe

Correção da precisão em erros de comparação

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

Significado

TRUNC:	Corte das casas decimais

Igualdade relativa considerada de 10^{-12} em

- Igualdade: (==)
- Desigualdade: (<>)
- Maior-Igual: (>=)
- Menor-Igual: (<=)
- Maior/Menor: (><) com igualdade absoluta
- Maior: (>)
- Menor: (<)

Compatibilidade

Por motivos de compatibilidade o controle sobre a igualdade relativa pode ser desativada no caso do (>) e (<) através da definição do dado de máquina MD10280 $MN_ PROG_FUNCTION_MASK Bit0 = 1.

Indicação

As comparações com dados do tipo REAL são condenados com uma certa imprecisão devido aos motivos já conhecidos. No caso de desvios não aceitáveis deve-se desviar para o cálculo INTEGER, onde os operandos são multiplicados por um fator de precisão e depois cortados com TRUNC.

Ações sincronizadas

O comportamento descrito dos comandos de comparação também se aplica nas ações sincronizadas.

Exemplos

Exemplo 1: Considerações de precisão

Código de programa	Comentário
`N40 R1=61.01 R2=61.02 R3=0.01`	`; atribuição dos valores iniciais`
`N41 IF ABS(R2-R1) > R3 GOTOF ERRO`	`; O salto seria executado até então`
`N42 M30`	`; fim do programa`
`N43 ERRO: SETAL(66000)`	
`R1=61.01 R2=61.02 R3=0.01`	`; atribuição dos valores iniciais`
`R11=TRUNC(R1*1000) R12=TRUNC(R2*1000) R13=TRUNC(R3*1000)`	`; Correção da precisão`
`IF ABS(R12-R11) > R13 GOTOF ERRO`	`; O salto não será mais executado`
`M30`	`; fim do programa`
`ERRO: SETAL(66000)`	

Exemplo 2: Formação e avaliação do quociente dos dois operandos

Código de programa	Comentário
`R1=61.01 R2=61.02 R3=0.01`	`; atribuição dos valores iniciais`
`IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF ERRO`	`; O salto não será executado`
`M30`	`; fim do programa`
`ERRO: SETAL(66000)`	

2.6 Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND)

Com os comandos MINVAL e MAXVAL os valores de duas variáveis podem ser comparados entre si. Como resultado é retornado o menor valor (com MINVAL) ou o maior valor (com MAXVAL).

Com o comando BOUND pode ser realizado o controle se o valor de uma variável de controle está dentro da faixa de valores definida.

Sintaxe

```
<menor valor>=MINVAL(<variável1>,<variável2>)
<maior valor>=MAXVAL(<variável1>,<variável2>)
<valor retornado>=<BOUND>(<mínimo>,<máximo>,<variável de controle>)
```

Significado

MINVAL:	Determina o **menor** valor de duas variáveis (<variável1>, <variável2>)
<menor valor>:	Variável de resultado para o comando MINVAL É passado para o menor valor de variável.
MAXVAL:	Determina o **maior** valor de duas variáveis (<variável1>, <variável2>)
<maior valor>:	Variável de resultado para o comando MAXVAL É passado para o maior valor de variável.
BOUND:	Controla se a variável (<variável de controle>) está dentro de uma faixa de valores definida.
<mínimo>:	Variável que é definida com o valor mínimo da faixa de valores
<máximo>:	Variável que é definida com o valor máximo da faixa de valores
<valor de retorno>:	Variável de resultado para o comando BOUND Quando o valor da variável de controle estiver dentro da faixa de valores definida, então a variável de resultado é passada para o valor da variável de controle. Quando o valor da variável de controle for maior que o valor máximo, então a variável de resultado é passada para o valor máximo da faixa de definição. Quando o valor da variável de controle for menor que o valor mínimo, então a variável de resultado é passada para o valor mínimo da faixa de definição.

Indicação

MINVAL, MAXVAL e BOUND também podem ser programadas em ações síncronas.

Indicação

Comportamento na igualdade

Em caso de igualdade, com MINVAL/MAXVAL é fornecido este mesmo valor. Com BOUND é retornado o valor da variável a ser controlada.

Exemplo

Código de programa	**Comentário**
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4,	rVar5, rValMin, rValMax, rRetVar
rValMin=MINVAL(rVar1,rVar2)	; rValMin é definido para o valor 10.5.
rValMax=MAXVAL(rVar1,rVar2)	; rValMax é definido para o valor 33.7.
rVar3=19.7	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 situa-se no interior dos limites, rRetVar será definido para 19.7.
rVar3=1.8	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 situa-se abaixo do limite mínimo, rRetVar será definido para 10.5.
rVar3=45.2	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 situa-se acima do limite máximo, rRetVar será definido para 33.7.

2.7 Prioridade das operações

Cada operador é ordenado por um nível de prioridade. Na avaliação de uma expressão sempre são aplicados primeiro os operadores de nível de prioridade mais alto. No caso de operadores de mesma propriedade, a avaliação é realizada da esquerda para direita.

Em expressões aritméticas se pode definir a ordem de execução através de parênteses, e com isso, fugir das regras convencionais de prioridade.

Sequência dos operadores

Do nível de prioridade mais alto até o mais baixo

1.	NOT, B_NOT	Negação, negação por Bits
2.	*, /, DIV, MOD	Multiplicação, divisão
3.	+, –	Adição, subtração
4.	B_AND	E bit a bit
5.	B_XOR	OU exclusivo bit a bit
6.	B_OR	OU bit a bit
7.	AND	E
8.	XOR	OU exclusivo
9.	OR	OU
10.	<<	Encadeamento de Strings, tipo de resultado STRING
11.	==, <>, >, <, >=, <=	Operadores de comparação

Indicação

O operador de encadeamento ":" para Frames não podem surgir com outros operadores em uma expressão. Por isso que uma categorização de prioridades para este operador não se faz necessária.

Exemplo de instrução If

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

2.8 Possíveis conversões de tipo

Função

Conversão de tipos na atribuição

O valor numérico constante, a variável ou a expressão, que se atribui a uma variável, deve ser compatível com o tipo desta variável. Se esta condição for preenchida, então o tipo será convertido automaticamente durante a atribuição.

Possíveis conversões de tipos

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

Explanações

* Na conversão de tipos de REAL para INT um valor fracionado >= 0.5 é arredondado para cima e, em caso contrário, arredondado para baixo (veja a função ROUND)

1) O valor <> 0 corresponde à TRUE, o valor == 0 corresponde à FALSE

2) Se o valor estiver na faixa de números admissíveis

3) Se apenas 1 caractere

4) Tamanho de String 0 = >FALSE, caso contrário é TRUE

Indicação

Se durante a conversão um valor for maior que a faixa de destino, será emitida uma mensagem de erro.

Se aparecerem tipos mistos em uma expressão, então ser realiza automaticamente uma adaptação de tipos. As conversões de tipos também são possíveis em ações sincronizadas; veja o capítulo "Ações sincronizadas de movimentos, conversão implícita de tipos".

2.9 Operações de String

Operações de String

Além das operações clássicas "Atribuição" e "Comparação" também são possíveis as seguintes operações de String:

- Conversão de tipos para STRING (AXSTRING) (Página 89)
- Conversão de tipos de STRING (NUMBER, ISNUMBER, AXNAME) (Página 90)
- Encadeamento de Strings (<<) (Página 91)
- Mudança para letras minúsculas / letras maiúsculas (TOLOWER, TOUPPER) (Página 92)
- Definir o tamanho de uma Strings (STRLEN) (Página 93)
- Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH) (Página 94)
- Seleção de uma String parcial (SUBSTR) (Página 95)
- Leitura e gravação dos caracteres individuais (Página 96)
- Formatação de String (SPRINT) (Página 97)

Significado especial do caractere 0

O caractere 0 é interpretado internamente como identificação final de uma String. Se um caractere é substituído pelo caractere 0, então a String será encurtada com isso.

Exemplo:

```
Código de programa                                  Comentário
DEF STRING[20] STRG="Eixo . está parado"
STRG[6]="X"
MSG(STRG)                                           ; Fornece a mensagem "Eixo X parado".
STRG[6]=0
MSG(STRG)                                           ; Fornece a mensagem "Eixo".
```

2.9.1 Conversão de tipos para STRING (AXSTRING)

Através da função "Conversão de tipos para STRING" se pode utilizar variáveis de diferentes tipos como parte integrante de uma mensagem (MSG).

Ocorre durante a utilização do operador << implícito para os tipos de dados INT, REAL, CHAR e BOOL (consultar " Encadeamento de Strings (<<) (Página 91) ").

Um valor INT é convertido na forma normal e legível. Para valores REAL são indicadas até 10 casas decimais.

Com o comando `AXSTRING` se pode converter variáveis do tipo AXIS para o tipo STRING.

Sintaxe

```
<STRING_ERG> = << <ocup._tipo>
<STRING_ERG> = AXSTRING(<identificador de eixo>)
```

Significado

<STRING_ERG>:	Variável para o resultado da conversão de tipos	
	Tipo:	STRING
<ocup._tipo>:	Tipos de variáveis INT, REAL, CHAR, STRING e BOOL	
AXSTRING:	O comando AXSTRING retorna o identificador de eixo indicado como String.	
<identificador de eixo>:	Variável para identificador de eixo	
	Tipo:	AXIS

Indicação

As variáveis FRAME não podem ser convertidas.

2.9.2 Conversão de tipos de STRING (NUMBER, ISNUMBER, AXNAME)

Com o comando NUMBER se converte de STRING para REAL. A capacidade de conversão pode ser verificada através do comando ISNUMBER.

Com o comando AXNAME se converte uma String no tipo de dado AXIS.

Sintaxe

```
<REAL_ERG>=NUMBER("<String>")
<BOOL_ERG>=ISNUMBER("<String>")
<AXIS_ERG>=AXNAME("<String>")
```

Significado

NUMBER:	O comando NUMBER retorna o número representado através da <String> como valor REAL.	
<String>:	Variável a ser convertida do tipo STRING	
<REAL_ERG>:	Variável para o resultado da conversão de tipos com NUMBER	
	Tipo:	REAL
ISNUMBER:	Com o comando ISNUMBER pode ser verificado se a <String> pode ser convertida em um número válido.	

<BOOL_ERG>:	Variável para o resultado da consulta com ISNUMBER		
	Tipo:	BOOL	
	Valor:	TRUE	ISNUMBER retorna o valor TRUE se a <String> apresenta um número REAL válido conforme as regras da linguagem.
		FALSE	Se ISNUMBER retornar o valor FALSE, então na chamada do NUMBER é emitido o alarme com a mesma <String>.
AXNAME:	O comando AXNAME converte a <String> especificada em um identificador de eixo. **Nota:** Se a <String> não pode ser associada a nenhum identificador de eixo configurado, então será emitido um alarme.		
<AXIS_ERG>:	Variável para o resultado da conversão de tipos com AXNAME		
	Tipo:	AXIS	

Exemplo

```
Código de programa                              Comentário
DEF BOOL BOOL_ERG
DEF REAL REAL_ERG
DEF AXIS AXIS_ERG
BOOL_ERG=ISNUMBER("1234.9876Ex-7")              ; BOOL_ERG == TRUE
BOOL_ERG=ISNUMBER("1234XYZ")                    ; BOOL_ERG == FALSE
REAL_ERG=NUMBER("1234.9876Ex-7")                ; REAL_ERG == 1234.9876Ex-7
AXIS_ERG=AXNAME("X")                            ; AXIS_ERG == X
```

2.9.3 Encadeamento de Strings (<<)

A função "Encadeamento de Strings" oferece a capacidade de compor uma String a partir de partes individuais.

O encadeamento é realizado através do operador "**<<**". Este operador tem como tipo de destino STRING para todas combinações dos tipos básicos CHAR, BOOL, INT, REAL e STRING. Uma eventual conversão que for necessária será realizada conforme as regras existentes.

Sintaxe

```
<ocup._tipo> << <ocup._tipo>
```

Significado

<ocup._tipo> :	Variável do tipo CHAR, BOOL, INT, REAL ou STRING
<< :	Operador para o encadeamento de variáveis (<ocup._tipo) para uma sequência de caracteres composta (tipo STRING). Este operador também está disponível como uma variante "unária". Dessa forma é possível executar uma conversão explícita de tipos para STRING (não para FRAME e AXIS): << <ocup._tipo>

Por exemplo, se pode compor uma mensagem destas ou um comando a partir de listas de texto e inserir parâmetros (algo como um nome de módulo):
`MSG(STRG_TAB[LOAD_IDX]<<NOME_MODULO)`

Indicação

Os resultados intermediários no encadeamento de Strings não podem exceder o tamanho máximo de String.

Indicação

Os tipos FRAME e AXIS não podem ser utilizados juntos com o operador "**<<**".

Exemplos

Exemplo 1: Encadeamento de Strings

Código de programa	Comentário
`DEF INT IDX=2`	
`DEF REAL VALUE=9.654`	
`DEF STRING[20] STRG="ÍNDICE:2"`	
`IF STRG=="Índice:"<<IDX GOTOF NO_MSG`	
`MSG("Índice:"<<IDX<<"/valor:"<<VALUE)`	`; indicação: "Index:2/valor:9.654"`
`NO_MSG:`	

Exemplo 2: Conversão explícita de tipos com <<

Código de programa	Comentário
`DEF REAL VALUE=3.5`	
`<<VALUE`	`;a variável indicada do tipo REAL será convertida no tipo STRING.`

2.9.4 Mudança para letras minúsculas / letras maiúsculas (TOLOWER, TOUPPER)

A função "Mudança para letras minúsculas / maiúsculas" permite a mudança de todas as letras de uma sequência de caracteres para uma representação uniforme.

Sintaxe

```
<STRING_ERG>=TOUPPER("<String>")
<STRING_ERG>=TOLOWER("<String>")
```

Significado

TOUPPER:	Com o comando TOUPPER todas as letras de uma sequência de caracteres muda para letras **maiúsculas**.	
TOLOWER:	Com o comando TOLOWER todas as letras de uma sequência de caracteres muda para letras **minúsculas**.	
<String>:	Sequência de caracteres que deve ser convertida	
	Tipo:	STRING
<STRING_ERG>:	Variável para o resultado da conversão	
	Tipo:	STRING

Exemplo

Visto que também é possível mostrar as especificações de usuário na interface de operação, a representação uniforme se obtém através de letras minúsculas ou maiúsculas:

Código de programa

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

2.9.5 Definir o tamanho de uma Strings (STRLEN)

Com o comando STRLEN é possível definir o tamanho de uma sequência de caracteres.

Sintaxe

```
<INT_ERG>=STRLEN("<STRING>")
```

Significado

STRLEN:	Com o comando STRLEN se define o tamanho da sequência de caracteres especificada. É retornado o número de caracteres que são contados desde o início da sequência de caracteres e que não são caracteres 0.	
<String>:	Sequência de caracteres cujo tamanho deve ser determinado	
	Tipo:	STRING
<INT_ERG>:	Variável para o resultado da definição	
	Tipo:	INT

Exemplo

A função junto com o acesso dos caracteres individuais permite determinar o fim de uma sequência de caracteres:

Código de programa

```
IF (STRLEN(NOME_MODULO)>10) GOTOF ERRO
```

2.9.6 Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH)

Esta funcionalidade permite localizar um único caractere ou uma String dentro de outra String. Os resultados de função indicam em qual posição da String foi encontrado o caractere / a String dentro da String analisada.

Sintaxe

`INT_ERG=INDEX(STRING,CHAR)` ; **Tipo de resultado: INT**

`INT_ERG=RINDEX(STRING,CHAR)` ; **Tipo de resultado: INT**

`INT_ERG=MINDEX(STRING,STRING)` ; **Tipo de resultado: INT**

`INT_ERG=MATCH(STRING,STRING)` ; **Tipo de resultado: INT**

Semântica

Funções de busca: Elas retornam a posição na String (primeiro parâmetro) onde a localização teve êxito. Se o caractere ou a String não puder ser localizado, então se retorna o valor -1. Neste caso o primeiro caractere possui a posição 0.

Significado

`INDEX`:	Busca (do começo) o caractere especificado como segundo parâmetro no primeiro parâmetro.
`RINDEX`:	Busca (do fim) o caractere especificado como segundo parâmetro no primeiro parâmetro.
`MINDEX`:	Corresponde à função INDEX, com exceção de que se transmite uma lista de caracteres (como String) de onde o índice do primeiro caractere encontrado é retornado.
`MATCH`:	Busca uma String dentro de uma String.

Dessa forma as Strings podem ser desmembradas conforme determinados critérios, algo como posições com espaço vazio ou sinais separadores de caminhos / atalhos ("/").

Exemplo

Desmembramento de uma entrada em nomes de caminho e de módulo

Código de programa	**Comentário**
`DEF INT PFADIDX, PROGIDX`	
`DEF STRING[26] ESPECIFICACAO`	

```
Código de programa                          Comentário
DEF INT LISTIDX
ESPECIFICACAO = "/_N_MPF_DIR/_N_EXE-
CUTE_MPF"
LISTIDX = MINDEX (ESPECIFICACAO,            ; Como valor no LISTIDX se retorna o 3; pois o
"M,N,O,P") + 1                                "N" é o primeiro caractere no parâmetro ESPECI-
                                              FICAÇÃO, da lista de seleção partindo-se do iní-
                                              cio.
PFADIDX = INDEX (ESPECIFICACAO,             ; com isto aplica-se: PFADIDX = 1
"/") +1
PROGIDX = RINDEX (ESPECIFICACAO,            ; com isto aplica-se: PROGIDX = 12
"/") +1
                                            ; Com a ajuda da função SUBSTR inserida na pró-
                                              xima secção, é possível desmembrar
                                              a variável ESPECIFICAÇÃO nos componentes ;"ata-
                                              lho";e "módulo":
VARIABLE = SUBSTR (ESPECIFICACAO,           ; retorna então "_N_MPF_DIR"
PFADIDX, PROGIDX-PFADIDX-1)
VARIABLE = SUBSTR (ESPECIFICACAO,           ; retorna então "_N_EXECUTE_MPF"
PROGIDX)
```

2.9.7 Seleção de uma String parcial (SUBSTR)

Com a função SUBSTRING é possível a leitura de quaisquer partes dentro de um String.

Sintaxe

```
<STRING_ERG>=SUBSTR(<String>,<Índice>,<comp.>)
<STRING_ERG>=SUBSTR(<String>,<Índice>)
```

Significado

SUBSTR:	A função fornece do <String> uma parte do string,que sai do <Índice> com o <comprimento> especificado. Se o parâmetro <comprimento> não estiver especificado, então a função fornece uma parte do string que sai do <Índice> até o final do string.
<Índice>:	Posição inicial da parte do string dentro do string. Se a posição inicial estiver após o fim da String, será retornada a String vazia (" "). Primeiro caractere do String: Índice = 0 Faixa de valores: 0 ... (comprimento String - 1)
<comprimento>:	comprimento da parte do string. Se for dada um comprimento muito maior, então será utilizado apenas a parte do string até o seu final. Faixa de valores: 1 ... (comprimento String - 1)

Exemplo

Código de programa	Comentário
`DEF STRING[29] ERG`	
`;  1`	
`;  0123456789012345678`	
`ERG = SUBSTR("CONFIRMAÇÃO: 10 até 99", 10, 2)`	`; ERG == "10"`
`ERG = SUBSTR("CONFIRMAÇÃO: 10 até 99", 10)`	`; ERG == "10 a 99"`

2.9.8 Leitura e gravação dos caracteres individuais

Dentro de um String é possível ler e gravar caracteres individualmente.

porém deve-se observar as seguintes condições:

- possível apenas para variáveis definidas por usuário e não em variáveis de sistema
- Caracteres individuais de um String somente são fornecidos em chamadas de subprogramas "call by value"

Sintaxe

```
<caractere>=<String>[<Índice>]
<caractere>=<String_Array>[<Array_Índice>,<Índice>]
<String>[<Índice>]=<caractere>
<String_Array>[<Array_Índice>,<Índice>]=<caractere>
```

Significado

`<String>`:	String qualquer
`<caractere>`:	variável do tipo CHAR
`<Índice>`:	Posição do caractere dentro do Strings. Primeiro caractere do String: Índice = 0 Faixa de valores: 0 ... (comprimento String - 1)

Exemplos

Exemplo 1: Notificação de variável

Código de programa	Comentário
`;  0123456789`	
`DEF STRING [50] MENSAGEM = "O eixo n alcançou a posição"`	
`MENSAGEM [6] = "X"`	
`MSG (MENSAGEM)`	`; "O eixo X atingiu a posição"`

Exemplo 2: Interpretação de uma variável de sistema

Código de programa	Comentário
DEF STRING [50] STRG	; Memória temporária para as variáveis de sistema
...	
STRG = $P_MMCA	; Carregar as variáveis de sistema
IF STRG[0] == "E" GOTO ...	; Interpretação da variável de sistema

Exemplo 3: Acionamento do parâmetro "call by value" e "call by reference"

Código de programa	Comentário
; 0123456	
DEF STRING[50] STRG = "eixo X"	
DEF CHAR CHR	
...	
EXTERN UP_VAL(EIXO)	; Definição da sub-rotina com o parâmetro "call by value"
EXTERN UP_REF(VAR EIXO)	; Definição da sub-rotina com o parâmetro "call by ref."
...	
UP_VAL(STRG[6])	; Transferência do parâmetro "by value"
...	
CHR = STRG[6]	; Memória temporária
UP_REF(CHR)	; Transferência de parâmetro "by reference"

2.9.9 Formatação de String (SPRINT)

Com a função pré-definida SPRINT as sequências de caracteres podem ser formatadas e, por exemplo, preparadas para emissão e envio até os dispositivos externos (veja também em "Process DataShare - Saída para um dispositivo/arquivo externo (EXTOPEN, WRITE, EXTCLOSE) (Página 689)").

Sintaxe

```
"<resultado_string>"=SPRINT("<formato_string>",<valor_1>,<valor_2>,...,<valor_n>)
```

Significado

SPRINT:	Identificador para uma função pré-definida que retorna um valor do tipo STRING.
"<formato_string>":	Sequência de caracteres que contém partes fixas e variáveis. As partes variáveis são definidas através do caractere de controle de formatos % e uma descrição de formato subsequente.

< valor_1>,< valor_2>,…,< valor_n>:	Valor na forma de uma constante ou variável do NC, que é inserido na posição onde está o n caractere de controle de formato %, de acordo com a descrição de formato dada no <formato_string>.
"<resultado_string>":	Sequência de caracteres formatada (máximo de 400 Bytes)

Descrições de formato disponíveis

%B:	Transformação para a String "TRUE", se o valor a ser transformado:
	• for diferente de 0.
	• não for uma String vazia (no caso de valores string).
	Transformação para a String "FALSE", se o valor a ser transformado:
	• for igual a 0.
	• for uma String vazia.
	Exemplo:
	`N10 DEF BOOL BOOL_VAR=1`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF BOOL_VAR:TRUE".

%C:	Transformação em um caractere ASCII.
	Exemplo:
	`N10 DEF CHAR CHAR_VAR="X"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C",CHAR_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF CHAR_VAR:X".

%D:	Transformação em uma String com um valor inteiro (INTEGER).
	Exemplo:
	`N10 DEF INT INT_VAR=123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D",INT_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF INT_VAR:123".

%<m>D:	Transformação em uma String com um valor inteiro (INTEGER). A String possui uma extensão mínima de <m> caracteres. As casas ausentes são preenchidas à esquerda por caracteres vazios.
	Exemplo:
	`N10 DEF INT INT_VAR=-123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D",INT_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF INT_VAR:xx-123" (onde o "x" equivale a um caractere vazio no exemplo dado).

%F:	Transformação em uma String com um número decimal com 6 casas decimais. Se necessário as casas decimais serão arredondadas ou preenchidas com 0 (zero).
	Exemplo:
	`N10 DEF REAL REAL_VAR=-1.2341234EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR: -1234.123400".

%<m>F:	Transformação em uma String com um número decimal com 6 casas decimais e com uma extensão total de pelo menos <m> caracteres. Se necessário as casas decimais serão arredondadas ou preenchidas com 0 (zero). Os caracteres ausentes na extensão total <m> são preenchidos à esquerda por caracteres vazios.
	Exemplo:
	`N10 DEF REAL REAL_VAR=-1.23412345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR: xxx-1234.123457" gravada ("x"está no exemplo substituindo um lugar em branco).

%.<n>F:	Transformação em uma String com um número decimal com <n> casas decimais. Se necessário as casas decimais serão arredondadas ou preenchidas com 0 (zero).
	Exemplo:
	`N10 DEF REAL REAL_VAR=-1.2345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR: -1234.568".

%<m>.<n>F:	Transformação em uma String com um número decimal com <n> casas decimais e com uma extensão total de pelo menos <m> caracteres. Se necessário as casas decimais serão arredondadas ou preenchidas com 0 (zero). Os caracteres ausentes na extensão total <m> são preenchidos à esquerda por caracteres vazios.
	Exemplo:
	`N10 DEF REAL REAL_VAR=-1.2341234567890EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:xx-1234.12" (onde o "x" equivale a um caractere vazio no exemplo dado).

%E:	Transformação em uma String com um número decimal em representação exponencial. A mantissa é armazenada de maneira normalizada com um dígito antes da vírgula e 6 casas decimais. Se necessário as casas decimais serão arredondadas ou preenchidas com 0 (zero). A exponente é iniciada com a palavra-chave "EX". Depois é seguido por um sinal ("+" ou "-") e um número de dois ou três dígitos.
	Exemplo:
	`N10 DEF REAL REAL_VAR=-1234.567890`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:-1.234568EX+03".

%<m>E:	Transformação em uma String com um número decimal em representação exponencial e com uma extensão total de pelo menos <m> caracteres. Os caracteres ausentes são preenchidos à esquerda por caracteres vazios. A mantissa é armazenada de maneira normalizada com um dígito antes da vírgula e 6 casas decimais. Se necessário as casas decimais serão arredondadas ou preenchidas com 0 (zero). A exponente é iniciada com a palavra-chave "EX". Depois é seguido por um sinal ("+" ou "-") e um número de dois ou três dígitos.
	Exemplo:
	`N10 DEF REAL REAL_VAR=-1234.5`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03" (onde o "x" equivale a um caractere vazio no exemplo dado)

%.<n>E:	Transformação em uma String com um número decimal em representação exponencial. A mantissa é armazenada de maneira normalizada com um dígito antes da vírgula e <n> casas decimais. Se necessário as casas decimais serão arredondadas ou preenchidas com 0 (zero). A exponente é iniciada com a palavra-chave "EX". Depois é seguido por um sinal ("+" ou "-") e um número de dois ou três dígitos.
	Exemplo:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:-1.23EX+03".

%<m>.<n>E:	Transformação em uma String com um número decimal em representação exponencial e com uma extensão total de pelo menos <m> caracteres. Os caracteres ausentes são preenchidos à esquerda por caracteres vazios. A mantissa é armazenada de maneira normalizada com um dígito antes da vírgula e <n> casas decimais. Se necessário as casas decimais serão arredondadas ou preenchidas com 0 (zero). A exponente é iniciada com a palavra-chave "EX". Depois é seguido por um sinal ("+" ou "-") e um número de dois ou três dígitos.
	Exemplo:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:xx-1.23EX+03" (onde o "x" equivale a um caractere vazio no exemplo dado).

%G:	Transformação em uma String com um número decimal de acordo com a faixa de valores em representação decimal ou exponencial: Em termos de quantia, se o valor apresentado for menor que 1.0EX-04 ou maior ou igual a 1.0EX(+06), será adotada a representação exponencial, caso contrário a representação decimal. No máximo são exibidos seis dígitos significantes, e com eventual arredondamento.
	Exemplo com representação decimal:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR: 0.000123457".
	Exemplo com representação exponencial:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:1.23457EX+06".

%<m>G:	Transformação em uma String com um número decimal de acordo com a faixa de valores em representação decimal ou exponencial (como %G). A String possui uma extensão total de pelo menos <m> caracteres. Os caracteres ausentes são preenchidos à esquerda por caracteres vazios.
	Exemplo com representação decimal:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:xxxx0.000123457" (onde o "x" equivale a um caractere vazio no exemplo dado).
	Exemplo com representação exponencial:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:xxx1.23457EX+06" (onde o "x" equivale a um caractere vazio no exemplo dado).

%.<n>G:	Transformação em uma String com um número decimal de acordo com a faixa de valores em representação decimal ou exponencial. No máximo são exibidos <n> dígitos significantes, e com eventual arredondamento. Em termos de quantia, se o valor apresentado for menor que 1.0EX-04 ou maior ou igual a 1.0EX(+<n>), será adotada a representação exponencial, caso contrário a representação decimal.
	Exemplo com representação decimal:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR: 0.000123".
	Exemplo com representação exponencial:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:1.23EX+03".

%<m>.<n>G:	Transformação em uma String com um número decimal de acordo com a faixa de valores em representação decimal ou exponencial (como %.<n>G). A String possui uma extensão total de pelo menos <m> caracteres. Os caracteres ausentes são preenchidos à esquerda por caracteres vazios.
	Exemplo com representação decimal:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:xxx0.0001235" (onde o "x" equivale a um caractere vazio no exemplo dado).
	Exemplo com representação exponencial:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF REAL_VAR:xx1.235EX+06" (onde o "x" equivale a um caractere vazio no exemplo dado).

%.<n>P:	Transformação de um valor REAL em um valor INTEGER que considera <n> casas decimais. O valor INTEGER é retornado como valor binário de 32 Bit. Se o valor a ser transformado não permite ser representado com 32 Bit, o processamento será cancelado com um alarme.
	Visto que uma sequência de Bytes gerada com a instrução de formato %.<n>P também pode conter zeros binários, então a String total gerada não corresponderá mais às convenções do tipo de dados STRING do NC. Por isso que ele não poderá ser gravado em uma variável do tipo STRING, nem ser processado com os comandos de String da linguagem NC. A única utilização possível é a transferência de parâmetros para o comando WRITE com o envio para um dispositivo externo compatível (veja o exemplo a seguir).
	Assim que o <formato_string> portar uma descrição de formato do tipo %P, a String inteira, com exceção do número binário gerado com %.<n>P, será emitida de acordo com o dado MD10750 $MN_SPRINT_FORMAT_P_CODE em código de caracteres ASCII, ISO (DIN6024) ou EIA (RS244). Se for programado um caractere que não permite ser convertido, o processamento será cancelado com alarme.
	Exemplo:
	`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`
	`...  ; Tratamento de erros`
	`N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)`
	`N210 IF ERROR <> 0`
	`… ; Tratamento de erros`
	Resultado: O String "INTEGER BINARY CODED: 'H0001E23A'" é transferida para o dispositivo de destino /ext/dev/1. O valor hexadecimal 0x0001E23A corresponde ao valor decimal 123450.

%<m>.<n>P:	Transformação de um valor REAL de acordo com o ajuste realizado no dado de máquina MD10751 $MN_SPRINT_FORMAT_P_DECIMAL em uma String com:
	• um número inteiro de <m> + <n> dígitos **ou**
	• um número decimal com no máximo <m> dígitos antes da vírgula e exatamente <n> casas decimais.
	Como na descrição de formato %.<n>P, a String inteira é armazenada no código de caracteres definido pelo dado MD10750 $MN_SPRINT_FORMAT_P_CODE.
	Transformação com o MD10751 = 0:
	O valor REAL é transformado em uma String com um número inteiro de <m> + <n> dígitos. As casas decimais serão arredondadas em <n> dígitos ou preenchidas com 0 (zero) conforme necessidade. As casas decimais ausentes são preenchidas por caracteres vazios. O sinal de menos é posicionado na extrema esquerda, no lugar do sinal de mais é colocado um caractere vazio.
	Exemplo:
	`N10 DEF REAL REAL_VAR=-123.45`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "PUNCHED TAPE FORMAT:-xx123450" (onde o "x" equivale a um caractere vazio no exemplo dado).
	Transformação com o MD10751 = 1:
	O valor REAL é transformado em uma String com um número decimal com no máximo <m> dígitos antes da vírgula e exatamente <n> casas decimais. Se necessário, os dígitos antes da vírgula serão truncados e as casas decimais arredondadas ou preenchidas com 0 (zero). Se <n> for igual a 0, também será descartado o ponto decimal.
	Exemplo:
	`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)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "PUNCHED TAPE FORMAT:-123.450 VAR2:23".

%S:	Inserção de uma String.
	Exemplo:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF STRING_VAR:ABCDEFG".

%<m>S:	Inserção de uma String com pelo menos <m> caracteres. Os dígitos ausentes são preenchidos por caracteres vazios.
	Exemplo:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)`
	Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF STRING_VAR:xxxABCDEFG" (onde o "x" equivale a um caractere vazio no exemplo dado).

%.<n>S:	Inserção de <n> caracteres de uma String (iniciando-se pelo primeiro caractere). **Exemplo:** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR) Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF STRING_VAR:ABC".
%<m>.<n>S:	Inserção de <n> caracteres de uma String (iniciando-se pelo primeiro caractere). A extensão total da String gerada possui pelo menos <m> caracteres. Os dígitos ausentes são preenchidos por caracteres vazios. **Exemplo:** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR) Resultado: A variável de String RESULT é gravada com a sequência de caracteres "CONTENT OF STRING_VAR:xxxxxABCDE" (onde o "x" equivale a um caractere vazio no exemplo dado).
%X:	Transformação de um valor INTEGER em uma String com representação hexadecimal. **Exemplo:** N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR) Resultado: A variável de String RESULT é gravada com a sequência de caracteres "INTEGER HEXADECIMAL:A5B8".

Indicação

A propriedade da linguagem NC, relacionada ao fato de não haver diferenciação entre letras maiúsculas e letras minúsculas nos identificadores e palavras-chave, também é aplicada nas descrições de formato. Por isso que eles podem ser programados sem diferença funcional tanto com letras minúsculas como letras maiúsculas.

Possibilidades de combinação

A seguinte tabela oferece informações sobre quais tipos de dados NC podem ser combinados com qual descrição de formato. São aplicadas as regras para transformação implícita de tipos de dados (veja em "Tipos de dados (Página 61)").

	Tipos de dados NC						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-

	Tipos de dados NC						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

Indicação

A tabela mostra que os tipos de dados AXIS e FRAME do NC não podem ser utilizados diretamente na função SPRINT. Porém, também é possível:

- transformar o tipo de dado AXIS com a função AXSTRING em uma String, que então pode ser processada com o SPRINT.
- ler os valores individuais do tipo de dado FRAME através do acesso de componente do Frame. Dessa forma recebemos um dado do tipo REAL, que pode ser processado com o SPRINT.

2.10 Saltos e ramificações de programa

2.10.1 Salto de retorno ao início do programa (GOTOS)

Com o comando GOTOS é possível saltar retornando ao início de um programa principal ou de uma sub-rotina para repetição do programa.

Através de dados de máquina se pode ajustar para que em cada salto de retorno ao início do programa:

- o tempo de execução do programa passe novamente para "0" (seja zerado).
- a contagem de peças seja incrementada com o valor "1".

Sintaxe

```
GOTOS
```

Significado

GOTOS:	Instrução de salto com destino de salto no início do programa.	
	A execução é controlada através do sinal de interface NC/PLC: DB21, ... DBX384.0 (controlar a ramificação do programa)	
	Valor:	Significado:
	0	Nenhum salto de retorno ao início do programa. A execução do programa é continuada com o próximo bloco de programa de peça após o GOTOS.
	1	Salto de retorno ao início do programa. Se repete o programa de peça.

Condições gerais

- GOTOS inicia internamente um STOPRE (parada de pré-processamento).
- Em um programa de peça com definições de dados (variáveis LUD) com o GOTOS se realiza o salto para o primeiro bloco de programa após a parte das definições, isto é, as definições não são executadas novamente. Por isso que as variáveis definidas preservam o valor obtido no bloco do GOTOS e não são resetadas nos valores padrão programados na parte das definições.
- O comando GOTOS não está disponível em ações sincronizadas e em ciclos tecnológicos.

Exemplo

```
Código de programa        Comentário
N10 ...                   ; Início do programa.
...
N90 GOTOS                 ; Salto até o início do programa.
...
```

2.10.2 Saltos de programa até marcadores de salto (GOTOB, GOTOF, GOTO, GOTOC)

Em um programa podem ser colocados marcadores de salto (Labels) de onde se pode saltar para outros pontos dentro do mesmo programa através dos comandos `GOTOF`, `GOTOB`, `GOTO` e `GOTOC`. A execução do programa é continuada com a instrução que seguir imediatamente após o marcador de salto. Com isso as bifurcações são realizáveis dentro do programa.

Além dos marcadores de salto também é possível o uso de números de bloco principal e secundário como destinos de salto.

Se estiver formulada a condição de salto (`IF ...`) antes da instrução de salto, então o salto de programa somente será realizado quando a condição de salto for preenchida.

Sintaxe

```
GOTOB <destino de salto>
IF <condição de salto> == TRUE GOTOB <destino de salto>

GOTOF <destino de salto>
IF <condição de salto> == TRUE GOTOF <destino de salto>

GOTO <destino de salto>
IF <condição de salto> == TRUE GOTO <destino de salto>

GOTOC <destino de salto>
IF <condição de salto> == TRUE GOTOC <destino de salto>
```

Significado

`GOTOB`:	Instrução de salto com destino de salto em direção ao início do programa.
`GOTOF`:	Instrução de salto com destino de salto em direção ao fim do programa.
`GOTO`:	Instrução de salto com localização do destino de salto. A localização é realizada primeiro em direção ao fim do programa, depois em direção ao início do programa.
`GOTOC`:	Efeito como o GOTO, mas com a diferença que o alarme 14080 "Destino de salto não encontrado" é omitido. Isto significa que a execução do programa não é interrompida em um caso de localização de destino de salto sem resultado, mas através do comando GOTOC se prossegue com a execução da próxima linha do programa.

<destino de salto>:	Parâmetro do destino de salto As possíveis indicações são:	
	<marcador de salto>:	O destino de salto é o marcador de salto colocado no programa com nome definido pelo usuário: <Marcador de salto>:
	<número de bloco>:	O destino de salto é um número de bloco principal ou secundário (p. ex.: 200, N300)
	Variável do tipo STRING:	Destino de salto variável. A variável serve para um marcador de salto ou um número de bloco.
IF:	Palavra-chave para formulação da condição de salto. A condição de salto permite todas operações de comparação e operações lógicas (resultado: TRUE ou FALSE). O salto de programa é executado quando o resultado desta operação for TRUE.	

Indicação

Marcadores de salto (Labels)

Os marcadores de salto sempre estão no início de um bloco. Se um número de programa estiver presente, o marcador de salto está imediatamente após o número do bloco.

Para a atribuição de nomes de marcadores de salto são aplicadas as seguintes regras:

- Número de caracteres:
  - pelo menos 2
  - no máximo 32
- Os caracteres permitidos são:
  - Letras
  - Números
  - Sublinhados
- Os primeiros dois caracteres devem ser letras ou sublinhados.
- Após o nome do marcador de salto segue um duplo ponto (":").

Condições gerais

- O destino de salto somente pode ser um bloco com marcador de salto ou número de bloco que está **dentro** do programa.
- Uma instrução de salto sem condição de salto deve ser programada em um bloco separado. Esta restrição não se aplica em instruções de salto com condição de salto. Aqui podem ser formuladas várias instruções de salto em um mesmo bloco.
- Nos programas com instruções de salto sem condições de salto não é obrigatório que o fim de programa M2/M30 esteja no fim do programa.

Exemplos

Exemplo 1: Saltos para marcadores de salto

```
Código de programa                Comentário
N10 …
N20 GOTOF Label_1                 ; salto no sentido final do programa até
                                  ; Marcador de salto "Label_1".
N30 …
N40 Label_0: R1=R2+R3             ; Marcador de salto "Label_0" definido.
N50 …
N60 Label_1:                      ; Marcador de salto "Label_1" definido.
N70 …
N80 GOTOB Label_0                 ; salto no sentido início do programa
                                  ; até marcador de salto "Label_0".
N90 …
```

Exemplo 2: Salto indireto até o número do bloco

```
Código de programa                Comentário
IF <condição> == TRUE
   R10=100                        ; atribuir destino de salto
ELSE
   R10=110                        ; atribuir destino de salto
ENDIF
; salto no sentido final do programa no bloco, cujo número está no R10
N10 GOTOF "N"<<R10
...
N90 ...
N100 ...                          ; Destino de salto
N110 ...
...
```

Exemplo 3: Salto até o destino de salto variável

```
Código de programa                Comentário
DEF STRING[20] DESTINO
IF <condição> == TRUE
   DESTINO = "Marcador1"          ; atribuir destino de salto
ELSE
   DESTINO = "Marcador2"          ; atribuir destino de salto
ENDIF
; salto no sentido final do programa até as variáveis destino de salto "Conteúdo de
DESTINO"
GOTOF DESTINO
Marcador1: T="Broca1"             ; destino de salto 1
...
```

Código de programa	Comentário
`Marcador2: T="Broca2"`	`; destino de salto 2`
`...`	

Exemplo 4: Salto com condição de salto

Código de programa	Comentário
`N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20`	`; atribuição dos valores iniciais`
`N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6`	`; Marcador de salto LA1`
`N42 R1=R1+R3 R4=R4-1`	
`; IF, condição de salto == TRUE`	
`; THEN salto no sentido do início do programa até o marcador de salto LA1`	
`N43 IF R4>0 GOTOB LA1`	
`N44 M30`	`; fim do programa`

2.10.3 Bifurcação do programa (CASE ... OF ... DEFAULT ...)

A função CASE oferece a opção de testar o valor atual (tipo: INT) INT) de uma variável ou de uma função de cálculo e, em função do resultado, saltar para diferentes pontos no programa.

Sintaxe

```
CASE(<expressão>) OF <constante_1> GOTOF <destino do salto_1>
<constante_2> GOTOF <destino do salto_2> ... DEFAULT GOTOF <destino
do salto_n>
```

Significado

`CASE`:	Instrução de salto	
`<expressão>`:	Variável ou função de cálculo	
`OF`:	Palavra-chave para formulação das bifurcações condicionais de programa	
`<constante_1>`:	Primeiro valor constante especificado para a variável ou função de cálculo	
	Tipo:	INT
`<constante_2>`:	Segundo valor constante especificado para a variável ou função de cálculo	
	Tipo:	INT
`DEFAULT`:	Para os casos em que a variável ou a função de cálculo não aceitar nenhum dos valores constantes especificados, se pode definir um destino de salto com a instrução DEFAULT. **Nota:** Se a instrução DEFAULT não for programada, então nestes casos o bloco seguinte da instrução CASE passará a ser o destino de salto.	
`GOTOF`:	Instrução de salto com destino de salto em direção ao fim do programa. Ao invés do `GOTOF` também podem ser programados todos os demais comandos GOTO (veja o assunto "Saltos de programa até os marcadores de salto").	

<destino_salto_1>:	A bifurcação vai para este destino de salto se o valor da variável ou da função de cálculo corresponder à primeira constante especificada. O destino de salto pode ser especificado da seguinte forma:	
	<marcador de salto>:	O destino de salto é o marcador de salto colocado no programa com nome definido pelo usuário: <marcador de salto>:
	<número de bloco>:	O destino de salto é um número de bloco principal ou secundário (p. ex.: 200, N300)
	Variável do tipo STRING:	Destino de salto variável. A variável serve para um marcador de salto ou um número de bloco.
<destino_salto_2>:	A bifurcação vai para este destino de salto se o valor da variável ou da função de cálculo corresponder à segunda constante especificada.	
<destino_salto_n>:	A bifurcação vai para este destino de salto se o valor da variável não corresponder a nenhum dos valores constantes especificados.	

Exemplo

Código de programa

```
...
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
...
```

A instrução CASE do N30 define as seguintes possibilidades de bifurcação de programa:

1. Se o valor da função de cálculo VAR1+VAR2-VAR3 = 7, então se salta até o bloco com a definição de marcador de salto "Label_1" (→ N40).
2. Se o valor da função de cálculo VAR1+VAR2-VAR3 = 9, então se salta até o bloco com a definição de marcador de salto "Label_2" (→ N50).
3. Se o valor da função de cálculo VAR1+VAR2-VAR3 não for 7 nem 9, então se salta até o bloco com a definição de marcador de salto "Label_3" (→ N60).

2.11 Repetição de partes do programa (REPEAT, REPEATB, ENDLABEL, P)

A função de repetição de partes do programa possibilita a repetição de partes do programa já escritas dentro do programa em qualquer tipo de ordem ou formação.

As linhas e áreas de programa que devem ser repetidas precisam ser marcadas com marcadores de salto (Labels).

Indicação

Marcadores de salto (Labels)

Os marcadores de salto sempre estão no início de um bloco. Se um número de programa estiver presente, o marcador de salto está imediatamente após o número do bloco.

Para a atribuição de nomes de marcadores de salto são aplicadas as seguintes regras:

- Número de caracteres:
  - pelo menos 2
  - no máximo 32
- Os caracteres permitidos são:
  - Letras
  - Números
  - Sublinhados
- Os primeiros dois caracteres devem ser letras ou sublinhados.
- Após o nome do marcador de salto segue um duplo ponto (":").

Sintaxe

1. Repetição de linha de programa individual:

```
<marcador de salto>: ...
...
REPEATB <marcador de salto> P=<n>
...
```

2. Repetição de área de programa entre marcador de salto e a instrução REPEAT:

```
<marcador de salto>: ...
...
REPEAT <marcador de salto> P=<n>
...
```

3. Repetição da área entre dois marcadores de salto:

```
<marcador de salto inicial>: ...
...
<marcador de salto final>: ...
...
```

```
REPEAT <marcador de salto inicial> <marcador de salto
final> P=<n>
...
```

Indicação

Não é possível colocar a instrução `REPEAT` com os dois marcadores de salto entre colchetes/parênteses. Se o `<marcador de salto inicial>` for encontrado antes da instrução `REPEAT` e o `<marcador de salto final>` não for alcançado antes da instrução `REPEAT`, então é executada a repetição entre o `<marcador de salto inicial>` e a instrução `REPEAT`.

4. Repetição entre o marcador de salto e o ENDLABEL:

```
<marcador de salto>: ...
...
ENDLABEL: ...
...
REPEAT <marcador de salto> P=<n>
...
```

Indicação

Não é possível colocar a instrução `REPEAT` com o `<marcador de salto>` e o `ENDLABEL` entre colchetes/parênteses. Se o `<marcador de salto>` for encontrado antes da instrução `REPEAT` e o `ENDLABEL` não for alcançado antes da instrução `REPEAT`, então é executada a repetição entre o `<marcador de salto>` e a instrução `REPEAT`.

Significado

`REPEATB`:	Comando para repetição de uma linha de programa
`REPEAT`:	Comando para repetição de uma área de programa

<marcador de salto>:	O <marcador de salto> identifica: • a linha de programa que deve ser repetida (com REPEATB) ou • o início da área de programa que deve ser repetida (com REPEAT)
	A linha de programa marcada com o <marcador de salto> pode estar antes ou depois da instrução REPEAT/REPEATB. A localização é realizada primeiramente em direção ao início do programa. Se o marcador de salto não for encontrado nesta direção, então a localização será realizada em direção ao fim do programa. **Exceção:** Se a área de programa entre o marcador de salto e a instrução REPEAT deve ser repetida (veja item 2. em Sintaxe), então a linha de programa marcada com o <marcador de salto> deve estar **antes** da instrução REPEAT, pois neste caso a localização **somente** é realizada em direção ao início do programa. Se a linha com o <marcador de salto> contém outras instruções, então estas serão executadas em toda repetição.
ENDLABEL:	Palavra-chave que marca o fim de uma área de programa que deve ser repetida Se a linha com o ENDLABEL contém outras instruções, então estas serão executadas em toda repetição. O ENDLABEL pode ser utilizado várias vezes no programa.
P:	Endereço para especificação da quantidade de repetições
<n>:	Quantidade de repetições da parte do programa
	Tipo: INT
	A parte do programa a ser repetida é repetida <n> vezes. Após a última repetição, o programa é continuado com a linha seguinte à linha da instrução REPEAT/REPEATB. **Nota:** Se nenhum P=<n> for especificado, então a parte do programa a ser repetida será repetida apenas uma vez.

Exemplos

Exemplo 1: Repetição de linha de programa individual

```
Código de programa                  Comentário
N10 POSITION1: X10 Y20
N20 POSITION2: CYCLE(0,,9,8)        ; Ciclo de posição
N30 ...
N40 REPEATB POSITION1 P=5           ; Execute cinco vezes o BLOCO N10.
N50 REPEATB POSITION2               ; Execute uma vez o bloco N20.
N60 ...
N70 M30
```

Exemplo 2: Repetição de área de programa entre marcador de salto e a instrução REPEAT

```
Código de programa          Comentário
N5 R10=15
```

Código de programa	Comentário
N10 Begin: R10=R10+1	; Largura
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	; Execute quatro vezes a área do N10 a N70.
N90 Z10	
N100 M30	

Exemplo 3: Repetição da área entre dois marcadores de salto

Código de programa	Comentário
N5 R10=15	
N10 Begin: R10=R10+1	; Largura
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	; Execute três vezes a área do N10 a N70.
N110 Z10	
N120 M30	

Exemplo 4: Repetição entre o marcador de salto e o ENDLABEL

Código de programa	Comentário
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	; Execute três vezes a área do N110 a N120.
N140 REPEAT BEGIN2 P=2	; Execute duas vezes a área N50 a N80.

Código de programa	Comentário
N150 M100	
N160 REPEAT BEGIN1 P=2	; Execute duas vezes a área N20 a N80.
N170 Z10	
N180 X0 Y0	
N190 M30	

Exemplo 5: Execução de operação de fresamento, posição de furação com diferentes tecnologias

Código de programa	Comentário
N10 BROCACENTRAGEM()	; Carregar a broca de centragem.
N20 POS_1:	; Posições de perfuração 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL:	
N80 POS_2:	; Posições de perfuração 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	
N130 BROCA()	; Substituir a broca e ciclo de perfuração.
N140 ROSCA(6)	; Carregar o macho M6 e ciclo de rosqueamento.
N150 REPEAT POS_1	; Repita uma vez a seção de programa a partir de POS_1 até ENDLABEL.
N160 BROCA()	; Substituir a broca e ciclo de perfuração.
N170 ROSCA(8)	; Carregar o macho M8 e ciclo de rosqueamento.
N180 REPEAT POS_2	; Repita uma vez a seção de programa a partir de POS_2 até ENDLABEL.
N190 M30	

Outras informações

- A repetição de partes do programa pode ser chamada de forma aninhada. Cada chamada ocupa um nível de sub-rotina.
- Se durante a usinagem for programada uma repetição de uma parte do programa `M17` ou um `RET`, então a repetição da parte do programa será cancelada. O programa é continuado no bloco após a linha do `REPEAT`.
- Na atual exibição do programa se indica a repetição da parte do programa como um nível próprio de sub-rotina.
- Se durante a execução da parte do programa for ativada uma interrupção de nível, então a execução do programa continua após a chamada da execução da parte do programa. Exemplo:

Código de programa	**Comentário**
N5 R10=15	
N10 BEGIN: R10=R10+1	; Largura
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	; Interrupção de plano
N50 X=-R10	
N60 Y=-R10	
N70 END: Z10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	
N120 Z10	; Continuar com a execução do programa.
N130 M30	

- É possível combinar estruturas de controle e repetições de partes de programa. Entretanto, não deve haver nenhuma sobreposição. Uma repetição de partes de programa deve estar situada em uma bifurcação de estrutura de controle ou uma estrutura de controle deve estar situada em uma repetição de partes de programa.
- Se for realizada a mistura de saltos e repetições de partes de programa, os blocos são executados de forma sequencial pura. Por exemplo, se um salto for realizado a partir de uma repetição de uma parte do programa, então a execução será realizada até ser encontrado o fim programado desta parte do programa.
  Exemplo:

Código de programa
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

Indicação

A instrução `REPEAT` deve estar situada após os blocos de deslocamento.

2.12 Estruturas de controle

Como padrão, o comando numérico executa os blocos NC na sequência programada.

Esta sequência pode ser variada mediante programação de blocos de programa alternativos e loops de programa. A programação dessas estruturas de controle ocorre através de senhas `IF`, `ELSE`, `ENDIF`, `LOOP`, `FOR`, `WHILE` e `REPEAT`.

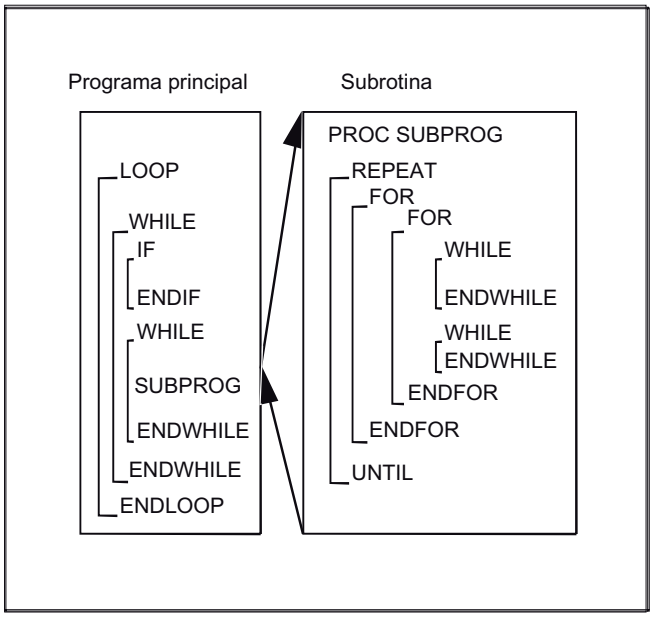
ATENÇÃO
erro de programação
As estruturas de controle somente são possíveis dentro da parte de instrução de um programa. As definições no cabeçalho não podem ser executadas de forma condicional ou de forma repetida.
Da mesma forma, as palavras-chave para estruturas de controle não podem ser sobrepostas com macros. Não é realizado nenhum controle na definição de macros.

Efeito

Uma estrutura de controle não pode ser utilizada de modo a reter um programa.

Nível de aninhamento

Dentro de cada nível de sub-rotina é possível alcançar um nível de aninhamento de até 16 estruturas de controle.



Relação do tempo de execução

Com o modo de interpretação padrão ativado, mediante uso de saltos de programa, se pode alcançar uma execução mais rápida do que com estruturas de controle.

Nos ciclos pré-compilados não existe nenhuma diferença entre saltos de programa e estruturas de controle.

amostra atual do bloco no refinamento do programa

Se dentro de um refinamento do programa forem realizados apenas blocos de avanço, então será mostrada no atual bloco o último bloco principal **antes** do término do refinamento.

Com isso p. ex. a fim de diagnosticar o procedimento dos blocos atuais que estão sendo usinados, é necessário ativar o decoder de bloco individual SBL2.

Literatura

Manual de funcionamento das funções básicas, Capítulo: Manual de funcionamento, capítulo: BAG, Canal, funcionamento do programa, Resetamento (K1) > bloco individual> decoder de bloco individual SBL2 com interrupção de avanço implícita

Retificação sem bloco principal

Se dentro de uma retificação do programa não estiver nenhum bloco principal programado, a retificação é executada no pré-processamento, até que a condição de retificação seja preenchida.

Com isso, ocorre uma alta utilização e a exibição pode ser afetada.

Como solução, é oferecido o comando STOPRE ou a adição de um tempo de espera G04 de 0 segundo **na** retificação.

Condições gerais

- Os blocos com elementos de estrutura de controle não podem ser omitidos.
- Os marcadores de salto (Labels) não são permitidos em blocos com elementos de estrutura de controle.
- As estruturas de controle são executadas de modo interpretativo. Ao identificar o fim de um loop, sob consideração das estruturas de controle encontradas, será realizada uma busca pelo início do loop. Por isso que no modo de interpretação não se controla completamente a estrutura de bloco de um programa.
- Sempre se recomenda evitar o uso misto de estruturas de controle e saltos de programa.
- No pré-processamento de ciclos pode ser realizado o controle quanto ao aninhamento correto de estruturas de controle.

2.12.1 Instrução condicional e bifurcação (IF, ELSE, ENDIF)

Instrução condicional: IF - bloco agrupado de programa - ENDIF

Em uma instrução condicional, o bloco agrupado de programa entre `IF` e `ENDIF` só será executados se a condição for atendida.

Bifurcação IF - bloco agrupado de programa_1 - ELSE - bloco agrupado de programa_2 - ENDIF

Em uma bifurcação será executado sempre um dos dois blocos agrupados de programas.

Caso a condição seja atendida, o bloco agrupado de programa_1 entre `IF` e `ELSE` será executado.

Caso a condição **não** seja atendida, o bloco agrupado de programa_2 entre `ELSE` e `ENDIF` será executado.

Sintaxe

Declaração de condição

```
IF <condição>
   Bloco agrupado de programa              ; Execução em: <Condição> == TRUE
ENDIF
```

Bifurcação

```
IF <condição>
   Blocos agrupados de programa_1          ; Execução em: <Condição> == TRUE
ELSE
   Blocos agrupados de programa_2          ; Execução em: <Condição> == FALSE
ENDIF
```

Significado

`IF`:	Inicia a instrução condicional e a bifurcação.
`ELSE`:	Inicia o bloco de programa alternativo.
`ENDIF`:	Marca o fim da instrução condicional e bifurcação.
`<condição>`:	Expressão lógica que resulta em TRUE ou FALSE.

Exemplo: Sub-rotina de troca de ferramentas

Código de programa	**Comentário**
`PROC L6`	`Rotina de troca de ferramentas`
`N500 DEF INT TNR_ATUAL`	`Variável para número T ativo`
`N510 DEF INT TNR_PRESELECAO`	`Variável para número T pré-selecionado`
	`Determinar a ferramenta atual`

Código de programa	Comentário
N520 STOPRE	
N530 IF $P_ISTEST	No modo de teste de programa se ...
N540 TNR_ATUAL = $P_TOOLNO	... realiza a leitura da "atual" ferramenta a partir do contexto do programa.
N550 ELSE	Caso contrário se ...
N560 TNR_ATUAL = $TC_MPP6[9998,1]	... lê a ferramenta do fuso.
N570 ENDIF	
N580 GETSELT(TNR_PRESELECAO)	Leitura do número T da ferramenta pré-selecionada no fuso.
N590 IF TNR_ATUAL <> TNR_PRESELECAO	Se a ferramenta pré-selecionada ainda não for a ferramenta atual, então se deve ...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... Aproximar ponto de troca de ferramentas ...
N610 M206	... e executar a troca de ferramenta.
N620 ENDIF	
N630 M17	

2.12.2 Loop de programa infinito (LOOP, ENDLOOP)

O loop infinito se aplica em programas de processamento infinito. No fim do loop sempre se executa novamente o salto de retorno ao início do loop.

Sintaxe

```
LOOP
...
ENDLOOP
```

Significado

LOOP:	Inicia o loop infinito.
ENDLOOP:	Marca o fim do loop e realiza o salto de retorno ao início do loop.

Exemplo

Código de programa
...
LOOP
MSG("nenhum corte de ferramenta ativo")
M0

Código de programa
STOPRE
ENDLOOP
...

2.12.3 Loop contador (FOR ... TO ..., ENDFOR)

O loop contador se utiliza quando uma execução de trabalho deve ser repetida com um número fixo de execuções.

Sintaxe

```
FOR <variável> = <valor inicial> TO <valor final>
...
ENDFOR
```

Significado

FOR:	Inicia o loop contador.	
ENDFOR:	Marca o fim do loop e realiza o salto de retorno ao início do loop, enquanto o valor final da contagem não for alcançado.	
<variável>:	Variável de contagem que é incrementada do valor inicial ao valor final pelo valor "1" em cada execução realizada.	
	Tipo	INT ou REAL **Nota:** Por exemplo, o tipo REAL é adotado quando são programados parâmetros R para um loop contador. Se a variável de contagem for do tipo REAL, seu valor será arredondado para um valor de número inteiro.
<valor inicial>:	Valor inicial da contagem Condição: O valor inicial deve ser menor que o valor final.	
<valor final>:	Valor final da contagem	

Exemplos

Exemplo 1: Variável INTEGER ou parâmetro R como variável de contagem

Variável INTEGER como variável de contagem:

Código de programa	**Comentário**
DEF INT iVARIABLE1	
R10=R12-R20*R1 R11=6	
FOR iVARIABLE1= R10 TO R11	; Variável de contagem = variável INTEGER
R20=R21*R22+R33	
ENDFOR	

Código de programa	Comentário
M30	

Parâmetro R como variável de contagem:

Código de programa	Comentário
R11=6	
FOR R10=R12-R20*R1 TO R11	; Variável de contagem = parâmetro R (variável Real)
R20=R21*R22+R33	
ENDFOR	
M30	

Exemplo 2: Produção de uma quantidade fixa de peças

Código de programa	Comentário
DEF INT QTDEPECAS	; Define a variável do tipo INT com o nome "QTDE-PECAS".
FOR QTDEPECAS = 0 TO 100	; Inicia o loop contador. A variável "QTDEPECAS" é incrementada do valor inicial "0" até o valor final "100".
G01 …	
ENDFOR	; Fim do loop contador.
M30	

2.12.4 Loop de programa com condição no início do loop (WHILE, ENDWHILE)

Em um loop WHILE a condição está situada no início do loop. O loop WHILE será executado enquanto a condição estiver preenchida.

Sintaxe

```
WHILE <condição>
...
ENDWHILE
```

Significado

WHILE:	Inicia o loop de programa.
ENDWHILE:	Marca o fim do loop e realiza o salto de retorno ao início do loop.
<condição>:	Condição que deve ser preenchida para que o loop WHILE seja executado.

Exemplo

Código de programa	Comentário
...	
WHILE $AA_IW[EIXOFURACAO] > -10	; Chamada do loop WHILE sob a seguinte condição: o atual valor nominal WCS do eixo de furação deve ser maior que -10.
G1 G91 F250 AX[EIXOFURACAO] = -1	
ENDWHILE	
...	

2.12.5 Loop de programa com condição no fim do loop (REPEAT, UNTIL)

Em um loop REPEAT a condição está situada no fim do loop. O loop REPEAT é executado uma vez e será repetido enquanto a condição for preenchida.

Sintaxe

```
REPEAT
...
UNTIL <condição>
```

Significado

REPEAT:	Inicia o loop de programa.
UNTIL:	Marca o fim do loop e realiza o salto de retorno ao início do loop.
<condição>:	Condição que deve ser preenchida para que o loop REPEAT não seja mais executado.

Exemplo

Código de programa	Comentário
...	
REPEAT	; Chamada do loop REPEAT.
...	
UNTIL ...	; Verificação, se a condição foi atendida.
...	

2.12.6 Exemplo de programa com estruturas de controle aninhadas

Código de programa	Comentário
LOOP	

Código de programa	**Comentário**
IF NOT $P_SEARCH	; IF nenhuma pesquisa bloco
G1 G90 X0 Z10 F1000	
WHILE $AA_IM[X] <= 100	; WHILE (Valor nominal eixo X <= 100)
G1 G91 X10 F500	; Gabarito de furação
Z-5 F100	
Z5	
ENDWHILE	
ELSE	; ELSE pesquisa em blocos
MSG("Na localização não se executa a furação")	
ENDIF	; ENDIF
$A_OUT[1] = 1	; próxima placa de furação
G4 F2	
ENDLOOP	
M30	

2.13 Comandos de coordenação (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

Um canal do NC pode, em princípio, processar o programa iniciado nele independente dos outros canais de seu grupo de modos de operação (BAG). Porém, se vários programas em vários canais do BAG estiverem envolvidos simultaneamente na fabricação de uma peça de trabalho, as execuções dos programas precisam ser coordenadas com os seguintes comandos de coordenação nos diferentes canais.

Pré-requisito

Todos os canais participantes da coordenação do programa precisam pertencer aos **mesmos** grupos de tipo de operação (BAG):

MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<canal>] = <número BAG>

Nome do canal em vez do número do canal

Também podem ser usados como parâmetros dos procedimentos pré-definidos da coordenação do programa, os nomes dos canais introduzidos em MD20000 $MC_CHAN_NAME[<índice do canal>] em vez dos números dos canais. A utilização dos nomes dos canais em programas NC precisa ser primeiro liberada:

MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE

Sintaxe

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

Significado

`INIT()`:	Procedimentos pré-definidos para a seleção do programa NC, o qual deve ser processado no canal especificado
`START()`:	Procedimento pré-definido para iniciar os programas escolhidos nos outros canais
`WAITM()`:	Procedimento pré-definido para espera pelo atingimento de um marcador nos canais especificados O marcador especificado é definido através de `WAITM` no próprio canal. O bloco anterior será finalizado com parada exata. O marcador é eliminado após a sincronização. Simultaneamente podem ser colocados no máx. 10 marcadores por canal.
`WAITE()`:	Procedimento pré-defino para a espera pelo fim do programa em um ou mais canais
`WAITMC()`:[1)]	Procedimento pré-definido para espera pelo atingimento de um marcador nos canais especificados Ao contrário de `WAITM`, a frenagem dos eixos até uma parada exata somente é iniciada se os outros canais ainda não tiverem alcançado o marcador.

SETM(): [1]	Procedimento pré-definido para posicionar um ou mais marcadores para coordenação do canal A usinagem no próprio canal não é influenciada. SETM mantém a sua validade através de reset de canal e início NC.		
CLEARM(): [1]	Procedimento pré-definido para eliminar um ou mais marcadores para coordenação do canal A usinagem no próprio canal não é influenciada. CLEARM() elimina todos os marcadores no canal. CLEARM(0) elimina apenas o marcador "0". CLEARM mantém a sua validade através de reset de canal e início NC.		
<ChanNr>:	Número de canal O número do próprio canal não deve ser indicado.		
	Tipo:	INT	
<Prog>:	Indicação do caminho absoluto ou relativo (**opcional**) + nome do programa		
	Tipo:	STRING	
	Sobre indicação do caminho ver: **Literatura** Manual de programação de preparação do trabalho; capítulo "Administração de arquivos e programas" > "Memória de programas" > "endereçamento de arquivos da memória de programas" (Página 227)		
<AckMode>:	Modo de término (**opcional**)		
	Tipo:	CHAR	
	Valores:	"N"	Sem confirmação A execução do programa é continuada após o envio do comando. O remetente não é informado se o comando não pode ser executado com sucesso.
		"S"	confirmação síncrona A execução do programa é mantida parada até que o componente receptor confirmar o comando. Em caso positivo de confirmação se executa o próximo comando. Uma mensagem de erro aparecerá em caso de confirmação negativa.
<MarkNr>:	Número do marcador de espera **Indicação** Em um sistema com vários canais há no máximo 100 marcadores disponíveis (marcador 0 ... 99). Em um sistema com um único canal há apenas o marcador 0.		

1) Para comunicação específica do usuário e / ou coordenação de canais, podem ser introduzidos marcadores através de SETM / CLEARM, inclusive sem utilização dos comandos de espera condicionais WAITMC. Daí em diante, os marcadores mantém os seus valores através de reset de canal e início NC.

Exemplos

START através de nomes de canal do MD20000

- Parametrização
```
MD10280 $MN_PROG_FUNCTION_MASK, Bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "PROCESSAMENTO" ; nome para canal 1
$MC_CHAN_NAME[ 1 ] = "ALIMENTAÇÃO"  ; nome para canal 2
```
- Programação

Código de programa	Comentário
START(PROCESSAMENTO)	; Início do Canal 1
START(ALIMENTAÇÃO)	; Início do Canal 2

Início do local "nome do canal" e variáveis do usuário

Código de programa	Comentário
DEF INT MÁQUINA = 1	; Definição das variáveis de usuários para o canal 1
DEF INT LADER = 2	; Definição das variáveis de usuários para o canal 2
...	
START(MAQUINA)	; Início do Canal 1
INÍCIO (CARREGADOR)	; Início do Canal 2

Início com o local "nome do canal", variáveis de usuários e o nome do canal parametrizado

Código de programa	Comentário
DEF INT chanNo1	; Definição das variáveis de usuários para o canal 1
DEF INT chanNo2	; Definição das variáveis de usuários para o canal 2
chanNo1 = CHAN_1	; Atribuição parametrizada do nome do canal, canal 1
chanNo2 = CHAN_2	;Atribuição parametrizada do nome do canal, canal 2
...	
START(chanNo1)	; Início do Canal 1
START(chanNo2)	; Início do Canal 2

Comando INIT com indicação absoluta do caminho

Seleção do programa /_N_MPF_DIR/_N_ABSPAN1_MPF no canal 2.

Código de programa
```
INIT(2,"/_N_WKS_DIR/_N_WELLE1_WPD/_N_ABSPAN1_MPF")
```

Comando INIT com o nome do programa

Seleção do programa com o nome "MYPROG". O sistema de comando busca o programa com base no atalho de busca.

Código de programa
```
INIT(2,"MYPROG")
```

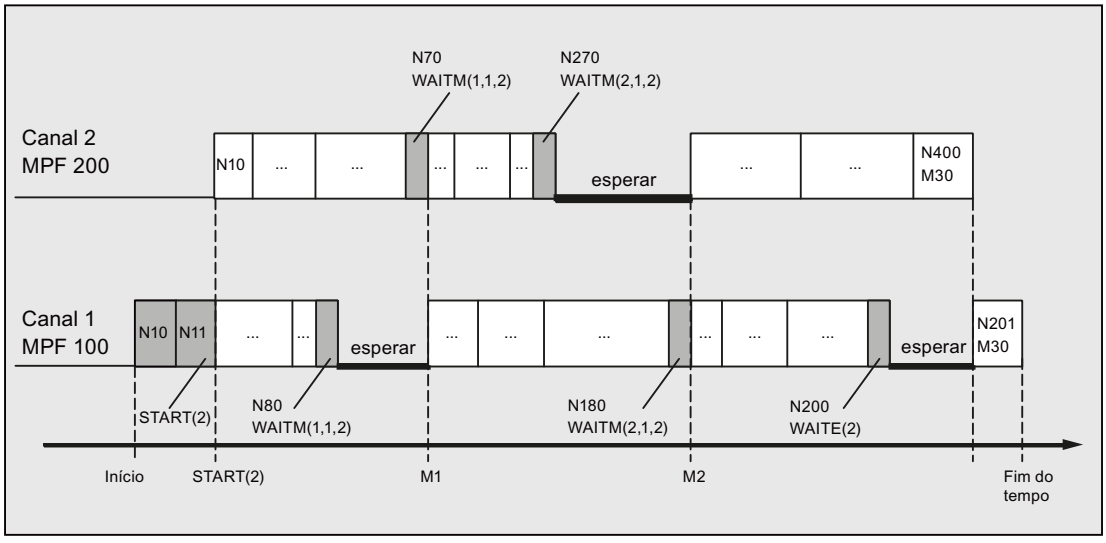
Coordenação de programa com WAITM

- Canal 1: O programa/_N_MPF_DIR/_N_MPF100_MPF é selecionado e iniciado.

Código de programa	**Comentário**
	; Programa MPF100
N10 INIT(2,"MPF200","N")	; Escolha do Programa MPF200, canal 2
N11 START(2)	; Início do Canal 2
...	
N80 WAITM(1,1,2)	; Aguarda a marca-ESPERA 1 no canal 1 e 2
N81 ...	; Canal 1, N81 e Canal 2, N71 iniciarão ; a sincronização
...	
N180 WAITM(2,1,2)	; Aguarda a marca-ESPERA 2 no canal 1 e 2
N181 ...	; canal 1, N181 e canal 2, N271 iniciarão ; a sincronização
...	
N200 WAITE(2)	; Espera pelo final do programa no canal 2
N201 ...	; O N201 só será após o final do programa ; início do MPF200 no canal 2
N201 M30	; final do programa canal 1

- Canal 2: No canal 1 o N10 e N20 do programa MPF200_MPF para o canal 2 será selecionado e iniciado.

Código de programa	**Comentário**
;$PATH=/_N_MPF_DIR	; Programa MPF200
...	
N70 WAITM(1,1,2)	Aguardo da marca-ESPERA 1 no canal 1 e 2
N71 ...	; Canal 1, N81 e Canal 2, N71 iniciarão ; a sincronização
...	
N270 WAITM(2,1,2)	Aguarda a marca-ESPERA 2 no canal 1 e 2
N271 ...	; canal 1, N181 e canal 2, N271 iniciarão ; a sincronização
...	
N400 M30	Fim do programa do canal 2



Condições gerais

Início do trabalho não sincronizado como princípio a seguir de acordo com a marca-ESPERA

No caso de uma coordenação de canal por meio das marcas WAIT pode ocorrer um início não síncrono do processamento dos registros posteriores. Este comportamento ocorre, quando imediatamente antes de atingir a marca WAIT em comum for iniciada uma ação nos canais a serem sincronizados, que resulta na anulação de curso restante com reposicionamento implícito (REPOSA).

Suposição: Atribuição atual nos eixos dos canais 1 e 2

- Canal 1: Eixos X1 e U
- Canal 2: Eixo X2

Tabelas 2-2 Tempo de percurso nos canais 1 e 2

Canal 1	Canal 2	Descrição
`...`	`...`	Qualquer processamento no canal 1 e 2
`N100 MARCA-ESPERA (20,1,2)`		Canal 1: atinge a `marca`-ESPERA e aguarda a sincronização com o canal 2
Início do processo GETD(U): • *Troca de eixos* • *Anulação de curso restante* • *REPOSA* *Fim*	`N200 GETD(U)`	Canal 2: pede pelo eixo U do canal 1 Canal 1: Processamento de `GET(U)` em segundo plano
	`N210 WAITM(20,1,2)`	Canal 2: atingiu a `marca`-ESPERA. ⇒ A sincronização dos canais 1 e 2 está completa
	`N220 G0 X2=100`	Canal 2: Início do trabalho N220
`N110 G0 X1=100`		Canal 1: Atraso no início do trabalho N110

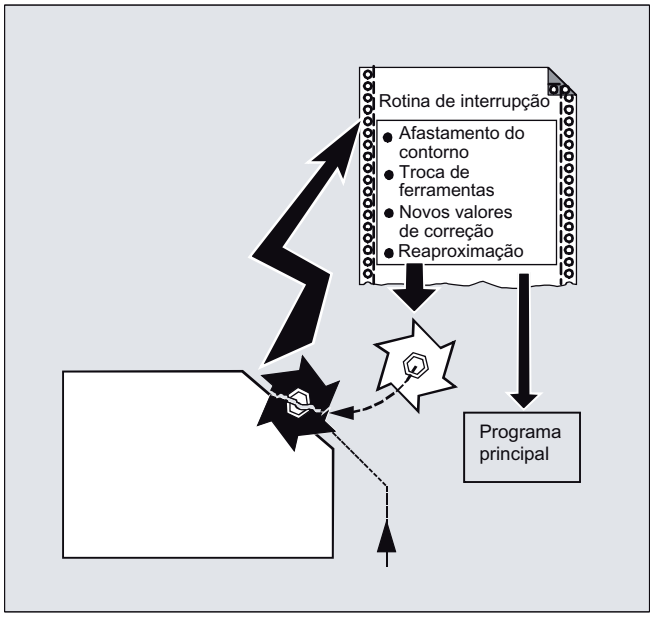
2.14 Rotina de interrupção (ASUP)

2.14.1 Funcionamento de uma rotina de interrupção

Indicação

Os termos "Sub-rotina assíncrona (ASUP)" e "Rotina de interrupção" que aparecem alternadamente na seguinte descrição significam a mesma funcionalidade.

O funcionamento de uma rotina de interrupção deve ser explanada com base em um exemplo típico:



A ferramenta quebra durante a usinagem. Com isso se dispara um sinal que para o processo de usinagem em andamento e ao mesmo tempo é iniciada uma sub-rotina – chamada de rotina de interrupção. Nesta sub-rotina estão contidas as instruções que deverão ser executadas para este caso.

Uma vez terminada a execução da sub-rotina (e com isso restabelecida a operabilidade), em função do comando `REPOS`, o comando retorna para o programa principal e continua com a usinagem a partir do ponto de interrupção (veja " Reaproximação do contorno (Página 554) ").

⚠ CUIDADO

Perigo de colisão

Se na sub-rotina for programado nenhum comando `REPOS`, então se executa o posicionamento no ponto final do bloco que segue o bloco interrompido.

Literatura

Manual de funcionamento das funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1), Capítulo: "Sub-rotinas assíncronas (ASUPs), Rotinas de interrupção"

2.14.2 Criar rotina de interrupção

Criar rotina de interrupção como sub-rotina

Durante a definição, a rotina de interrupção é identificada como se fosse uma sub-rotina.

Exemplo:

Código de programa	**Comentário**
PROC RETRAC_Z	; Nome de programa "RETRAC_Z"
N10 ...	, Em seguida, seguem os blocos NC.
...	
N50 M17	; No final, fim do programa e retorno ao programa principal.

Salvar comandos modais G (SAVE)

Na definição, a rotina de interrupção pode ser identificada com `SAVE`.

O atributo `SAVE` faz com que os comandos modais G sejam salvas antes da chamada da rotina de interrupção e novamente ativadas no fim da sub-rotina de interrupção (veja " Sub--rotinas com mecanismo SAVE (SAVE) (Página 174) ").

Com isso é possível continuar a usinagem do ponto de interrupção depois da rotina de interrupção ser concluída.

Exemplo:

Código de programa
PROC RETRAC_Z SAVE
N10 ...
...
N50 M17

Atribuir outras rotinas de interrupção (SETINT)

Dentro da rotina de interrupção se pode programar instruções `SETINT` (veja "Atribuir e iniciar rotinas de interrupção (SETINT)" (Página 134)) e com isso podem ser ativadas outras rotinas de interrupção. A inicialização só ocorre com a entrada.

Literatura

Para mais informações sobre a criação de sub-rotinas, veja o capítulo "Técnica de sub-rotinas, técnica de macros".

2.14.3 Atribuição e partida de rotinas de interrupção (SETINT, PRIO, BLSYNC)

O sistema de comando dispõe de várias entradas rápidas (Entrada 1 ... 8), que iniciam respectivamente uma interrupção (1 ... 8). A cada interrupção pode, através do comando `SETINT` ser atribuída uma prioridade e uma rotina de interrupção. Se a interrupção for iniciada através da definição da entrada rápida, o processamento atual no canal é interrompido e a rotina de interrupção é iniciada.

Prioridade de interrupção

Se em um programa de peças forem atribuídas várias entradas de interrupção, devem ser atribuídas diversas prioridades às interrupções.

A uma interrupção podem ser atribuídos os valores de prioridade de 1 ... 128. O valor de prioridade 1 corresponde a prioridade mais alta, 128 a mais baixa.

Sintaxe

```
SETINT(<n>) <NOME>
SETINT(<n>) PRIO=<valor> <NOME>
SETINT(<n>) PRIO=<valor> <NAME> BLSYNC
SETINT(<n>) PRIO=<valor> <NAME> LIFTFAST
```

Significado

`SETINT(<n>)`:	O sinal de interrupção <n> será atribuído no programa NC (ASUP) <Nome>. A rotina de interrupção atribuída é iniciada, assim que o sinal de interrupção <n> == 1 for reconhecido. **Nota:** se ao sinal de interrupção <n> for atribuído uma outra rotina de interrupção, a atribuição anterior será revogada.	
`<n>`:	Número do sinal de interrupção	
	Tipo:	INT
	Faixa de valores:	1 ... 32
`PRIO= :`	Prioridade da interrupção (opcional)	
`<valor>`:	(**opcional**) valor de prioridade	
	Tipo:	INT
	Faixa de valores:	1 ... 128 (1 ⇒ prioridade mais alta)
`<NOME>`:	Nome do programa NC (ASUP)	
`BLSYNC`:	(**opcional**) `BLSYNC` é responsável pelo fato de que após o início da interrupção, primeiro espera-se até que o bloco atual seja processado. Somente depois, a rotina de interrupção será executada.	
`LIFTFAST`:	(**opcional**) `LIFTFAST` é responsável pelo fato de que após o início da interrupção, ocorre primeiro uma retração rápida (consultar o capítulo "Retração rápida do contorno (SETINT LIFTFAST, ALF) (Página 137)"). Somente depois, a rotina de interrupção será executada.	

Condições gerais

Regras de interrupção

1. Para cada interrupção que não puder ser processada imediatamente ou já estiver em processamento atualmente, será armazenada uma outra requisição de interrupção. Além disto, as requisições de interrupção para esta interrupção serão perdidas.
2. Se atualmente for processada uma interrupção e for iniciada uma outra interrupção com uma prioridade mais alta, esta interrompe a interrupção com a prioridade mais baixa. Após a conclusão da interrupção com a prioridade mais alta, a interrupção com a prioridade mais baixa é prosseguida. Se durante o processamento da interrupção com a prioridade mais alta chegarem outras requisições para uma interrupção com prioridade mais baixa, uma requisição será armazenada. As outras serão perdidas.
3. Se atualmente for processada uma interrupção e for iniciada uma outra interrupção com uma prioridade mais alta, esta interrompe a interrupção com a prioridade mais baixa. A interrupção com a prioridade mais alta será processada. Se, por outro lado, for iniciada uma interrupção com a prioridade mais alta, a interrupção atual será interrompida e a interrupção com a prioridade mais alta será processada. São possíveis no máximo seis níveis de interrupção ativos. Um nível de interrupção em processamento atual e cinco níveis de interrupção em espera. Para cada nível de interrupção ativo é armazenada, no máximo, uma outra requisição de interrupção. Todas as outras requisições de interrupção serão perdidas. A requisição de interrupção também será perdida quando esta for requisitada para outros níveis de interrupção (nível de interrupção ≥ 7).

Exemplos

Exemplo 1: Atribuir rotinas de interrupção e definir prioridade

```
Código de programa                    Comentário
N20 SETINT(3) PRIO=1 RETRAC_Z         ; Entrada IF 3 == 1
                                      ; Iniciar THEN Interruptroutine "ABHEB_Z"
N30 SETINT(2) PRIO=2 RETRAC_X         ; Entrada IF 2 == 1
                                      ; Iniciar THEN Interruptroutine "ABHEB_X".
```

As rotinas de interrupção são executadas uma a uma na sequência dos valores de prioridade, se as entradas forem disponibilizadas simultaneamente: primeiro "RETRAC_Z", depois "RETRAC_X".

Exemplo 2: Nova atribuição de rotinas de interrupção

```
Código de programa                    Comentário
N20 SETINT(3) PRIO=2 RETRAC_Z         ; Entrada IF 3 == 1
                                      ; Iniciar THEN Interruptroutine "ABHEB_Z"
...
N80 SETINT(3) PRIO=1 RETRAC_X         ; Entrada IF 3 == 1
                                      ; Iniciar THEN Interruptroutine "ABHEB_X"
```

2.14.4 Desativar / ativar a atribuição de uma rotina de interrupção (DISABLE, ENABLE)

Uma instrução `SETINT` é desativada com `DISABLE` e novamente ativada com `ENABLE` sem perder a associação Entrada → Rotina de interrupção.

Sintaxe

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

Significado

`DISABLE(<n>)`:	Comando: **Desativação** da associação da rotina de interrupção com a entrada <n>	
`ENABLE(<n>)`:	Comando: **Reativação** da associação da rotina de interrupção com a entrada <n>	
`<n>`:	Parâmetro: Número do sinal de interrupção	
	Tipo:	INT
	Faixa de valores:	1 ... 32

Exemplo

Código de programa	**Comentário**
`N20 SETINT(3) PRIO=1 RETRAC_Z`	`; Se a entrada 3 liga, então`
	`; a Interruptroutine "ABHEB_Z" deverá iniciar.`
`...`	
`N90 DISABLE(3)`	`; A instrução SETINT do N20 é desativada.`
`...`	
`N130 ENABLE(3)`	`; A instrução SETINT do N20 é novamente ativada.`
`...`	

2.14.5 Deletar a atribuição de uma rotina de interrupção (CLRINT)

Uma com atribuição definida `SETINT` de um sinal de interrupção a um programa NC (ASUP) pode ser apagada com `CLRINT`.

Sintaxe

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

Significado

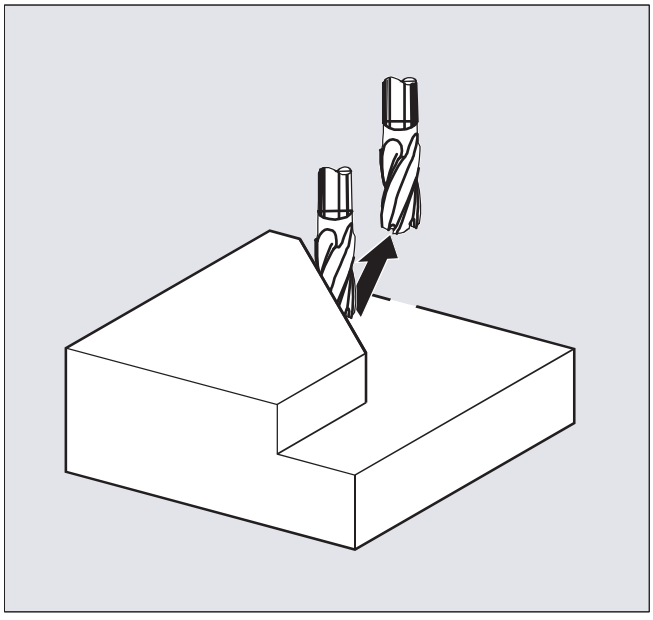
CLRINT(<n>):	Comando: Exclusão da atribuição do sinal de interrupção <n> ao programa NC definido com SETINT NC (ASUP) <n>	
<n>:	Parâmetro: Número do sinal de interrupção	
	Tipo:	INT
	Faixa de valores:	1 ... 32

Exemplo

Código de programa	Comentário
N20 SETINT(3) PRIO=2 RETRAC_Z	
...	
N50 CLRINT(3)	; A atribuição entre a entrada "3" e a rotina de interrupção "RETRAC_Z" foi excluída.

2.14.6 Retração rápida do contorno (SETINT LIFTFAST, ALF)

Através de uma instrução SETINT com LIFTFAST se realiza o afastamento da ferramenta mediante retração rápida assim que a entrada for ativada.



A execução restante depende se a instrução SETINT, além da LIFTFAST, também contiver uma rotina de interrupção:

Com rotina de interrupção: A rotina de interrupção é executada **após** a retração rápida.

Sem rotina de interrupção: A usinagem é parada com alarme após a retração rápida.

Sintaxe

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

Significado

SETINT(<n>):	Comando: Atribuir a entrada <n> de uma rotina de interrupção. A rotina de interrupção atribuída é iniciada assim que a entrada <n> for ativada.	
<n>:	Parâmetro: Número da entrada	
	Tipo:	INT
	Faixa de valores:	1 ... 8
PRIO= :	Definição da prioridade	
<valor>:	Valor de prioridade	
	Faixa de valores:	1 ... 128
	A prioridade 1 corresponde ao mais alto nível de prioridade.	
<NOME>:	Nome da sub-rotina (rotina de interrupção) que deve ser executada.	
LIFTFAST:	Comando: Retração rápida do contorno	
ALF=…:	Comando: Sentido de deslocamento programável (no bloco de deslocamento) Para conhecer as opções de programação com ALF, veja o assunto " Sentido de deslocamento na retração rápida do contorno (Página 139) ".	

Condições gerais

Comportamento com Frame ativo com espelhamento

Na determinação do sentido de retração é verificado se um Frame está ativo com espelhamento. Neste caso a direita e a esquerda são invertidas para o sentido de retração em função do sentido de tangente. Os componentes de sentido no sentido da ferramenta não são espelhados. Este comportamento é ativado através do ajuste de dado de máquina:

MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

Exemplo

Uma ferramenta cancelada deve ser substituída automaticamente por uma ferramenta gêmea. A usinagem é continuada com a nova ferramenta.

Programa principal:

Programa principal	**Comentário**
N10 SETINT(1) PRIO=1 TROCA_F LIFTFAST	; Quando a entrada 1 é ativada, a ferramenta é imediatamente afastada do contorno com retração rápida (código nº 7 para correção do raio da ferramenta G41). Em seguida é executada a rotina de interrupção "TROCA_F".
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	

Programa principal	**Comentário**
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	

Sub-rotina:

Sub-rotina	**Comentário**
PROC TROCA_F SAVE	; Sub-rotina com armazenamento do atual estado operacional
N10 G0 Z100 M5	; Posição de troca de ferramenta, parada do fuso
N20 T11 M6 D1 G41	; Trocar a ferramenta
N30 REPOSL RMBBL M3	; Reaproximação do contorno e salto de retorno para o programa principal (é programado em um bloco)

2.14.7 Sentido de deslocamento na retração rápida do contorno

Movimento de retrocesso

O plano do movimento de retrocesso é definida através dos seguintes comandos G:

- LFTXT
  O plano do movimento de retrocesso é determinado a partir da tangente da trajetória e do sentido da ferramenta (ajuste padrão).
- LFWP
  O plano do movimento de retrocesso é o plano de trabalho ativo que se seleciona com os comandos G G17, G18 ou G19. O sentido do movimento de retrocesso não depende da tangente da trajetória. Com isso pode-se programar uma retração rápida paralela ao eixo.
- LFPOS
  Retrocesso do eixo identificado com POLFMASK / POLFMLIN na posição de eixo absoluta programada com POLF.
  O ALF não tem nenhuma influência sobre o sentido de retração para vários eixos assim como para vários eixos de contexto linear.
  Literatura:
  Manual básico de programação; capítulo: "retrocesso rápido durante o rosqueamento"

Sentido de deslocamento programável (ALF=...)

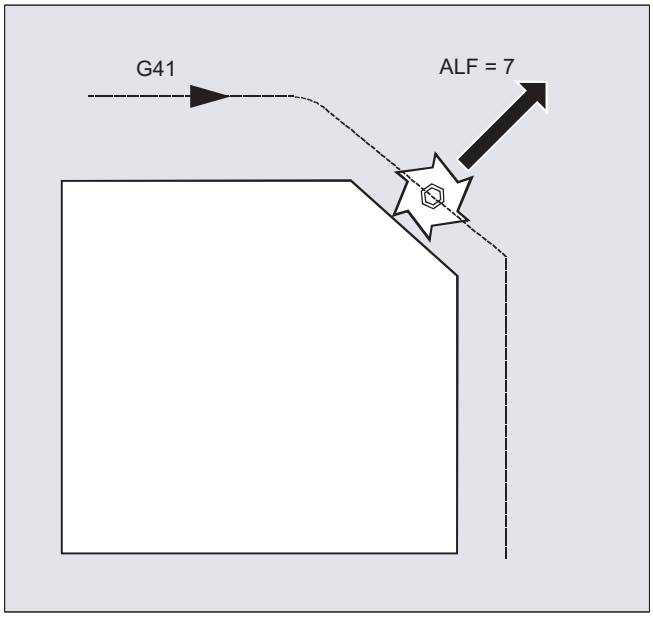
No plano do movimento de retrocesso o sentido é programado com ALF, em discretos passos de 45 graus.

Os possíveis sentidos de deslocamento estão armazenados no comando numérico através de códigos especiais e eles podem ser chamados através deste número.

Exemplo:

Código de programa
N10 SETINT(2) PRIO=1 RETRAC_Z LIFTFAST
ALF=7

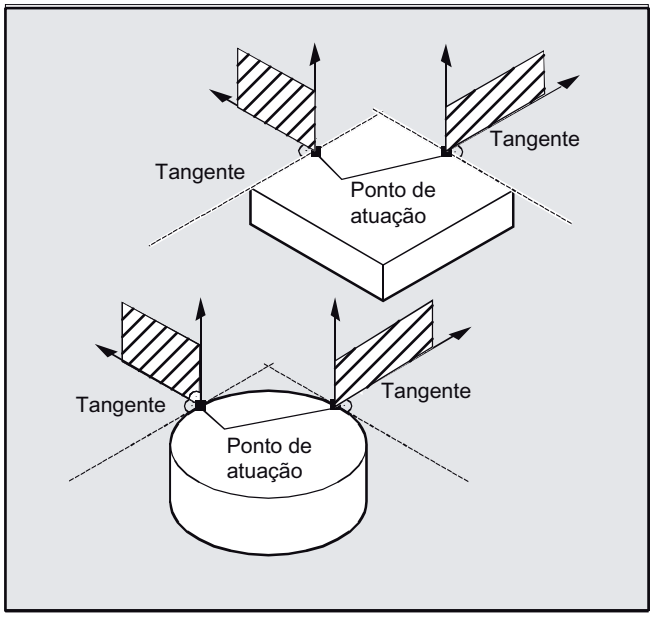
Com o `G41` (sentido de usinagem à esquerda do contorno) ativado, a ferramenta se afasta perpendicularmente do contorno.



Plano de referência para a descrição dos sentidos de deslocamento com LFTXT

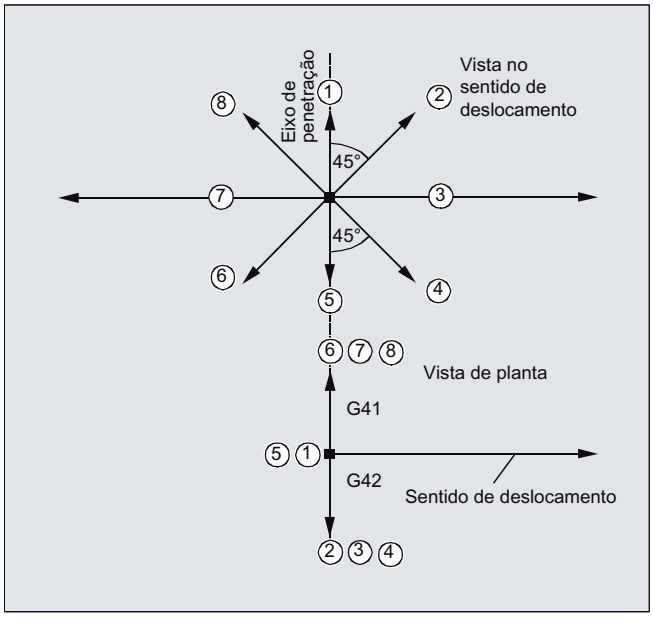
No ponto de aplicação da ferramenta no contorno programado é fixado um plano que serve de referência para a especificação do movimento de retração com o respectivo código.

O plano de referência é fixado a partir de um eixo longitudinal da ferramenta (sentido de penetração) e um vetor que está posicionado perpendicularmente a esse eixo e para o ponto de aplicação da ferramenta no contorno.



Códigos com sentidos de deslocamento com LFTXT

Os números de código com os sentidos de deslocamento a partir do plano de referência estão indicados na figura a seguir.



Para `ALF=1` se define o retrocesso no sentido da ferramenta.

Com `ALF=0` se desativa a função de "Retração rápida".

⚠ CUIDADO

Perigo de colisão

Com a correção do raio da ferramenta ativada:

- os códigos 2, 3, 4 com `G41`
- os códigos 6, 7, 8 com `G42`

não devem ser utilizados, pois nestes casos a ferramenta poderia colidir com a peça enquanto se dirige até o contorno.

Códigos com sentidos de deslocamento com LFWP

Com `LFWP` o sentido do plano de trabalho é atribuído como segue:

- `G17`: Plano X/Y
  `ALF=1`: Retrocesso no sentido X
  `ALF=3`: Retrocesso no sentido Y
- `G18`: Plano Z/X
  `ALF=1`: Retrocesso no sentido Z
  `ALF=3`: Retrocesso no sentido X
- `G19`: Plano Y/Z
  `ALF=1`: Retrocesso no sentido Y
  `ALF=3`: Retrocesso no sentido Z

2.14.8 Execução de movimentos com rotinas de interrupção

Rotina de interrupção sem LIFTFAST

Os movimentos dos eixos são desacelerados na trajetória até pararem. Em seguida é iniciada a rotina de interrupção.

A posição inicial é gravada como posição de interrupção e será ativada com `REPOS RMIBL` no final da rotina ininterrupta.

Rotina de interrupção com LIFTFAST

Os movimentos dos eixos são freados na trajetória. Ao mesmo tempo se executa o movimento `LIFTFAST` como movimento sobreposto. A rotina de interrupção é iniciada quando for parado o movimento de percurso e o movimento `LIFTFAST`.

Como posição de interrupção é memorizada a posição no contorno onde foi iniciado o movimento `LIFTFAST`, onde o percurso também foi abandonado.

Com `LIFTFAST` e `ALF=0` a rotina de interrupção se comporta de modo idêntico como a rotina de interrupção sem `LIFTFAST`.

Indicação

O valor com que os eixos geométricos se afastam do contorno na retração rápida, pode ser ajustado através de um dado de máquina.

2.15 Troca de eixos, troca de fusos (RELEASE, GET, GETD)

Um ou vários eixos ou fusos sempre podem interpolados em apenas um canal. Se um eixo precisa alternar entre dois canais diferentes (p. ex. trocador de paletes), então se deve liberar primeiro o atual canal e depois passado para o outro canal. O eixo é trocado entre os canais.

Ampliação da troca de eixos

Um eixo/fuso pode ser trocado com parada de pré-processamento e aça sincronizada entre o pré-processamento e o processamento principal, ou também, alternativamente sem a parada de pré-processamento. Além disso, uma troca de eixos também é possível através de

- Rotação de contentor de eixo AXCTSWE ou AXCTWED mediante `GET/GETD` implícito.
- Frame com rotação, se através disso este eixo estiver ligado com outro eixo.
- Ações sincronizadas; veja em ações sincronizadas de movimentos, "Troca de eixos `RELEASE`, `GET`".

Fabricante da máquina

Observe as instruções do fabricante da máquina. Através de dados de máquina configuráveis, um eixo deve ser definido de forma clara em todos os canais para a troca de eixos, e o comportamento de troca de eixos também podem ser ajustado de forma modificável através de dados de máquina.

Sintaxe

`RELEASE(nome de eixo, nome de eixo, ...)` ou `RELEASE(S1)`

`GET(nome de eixo, nome de eixo, ...)` ou `GET(S2)`

`GETD(nome de eixo, nome de eixo, ...)` ou `GETD(S3)`

Com GETD (GET Directly) um eixo é buscado diretamente de outro canal. Isto significa que para este GETD não é necessário programar um RELEASE adequado em outro canal. Isto também significa que agora se deve estabelecer outra comunicação de canais (p. ex. marcadores Wait).

Significado

`RELEASE (nome de eixo, nome de eixo, …):`	Habilitação do eixo (n)
`GET (nome de eixo, nome de eixo, …):`	Aceitação do eixo(n)
`GETD (nome de eixo, nome de eixo, …):`	Aceitação direta do eixo(n)
`Nome de eixo:`	Atribuição de eixos no sistema: AX1, AX2, … ou especificação dos nomes de eixo da máquina
`RELEASE(S1):`	Liberação dos fusos S1, S2, …
`GET(S2):`	Aceitação dos fusos S1, S2, …
`GETD(S3):`	Aceitação direta dos fusos S1, S2, …

Solicitação GET sem parada de pré-processamento

Se, após uma solicitação GET **sem** parada de pré-processamento, o eixo for novamente habilitado com `RELEASE(eixo)` ou `WAITP(eixo)`, então um `GET` seguinte conduzirá para um `GET` **com** parada de pré-processamento.

⚠ CUIDADO

Classificação dos eixos modificado

Um eixo aceito com GET permanece atribuído neste canal, mesmo após uma tecla ou programa RESET.

Com uma inicialização do programa a atribuição dos eixos ou fusos trocados deverá ocorrer via programa, caso o eixo não seja necessário em seu canal básico.

Com POWER ON ele será atribuído ao canal definido nos dados da máquina.

Exemplos

Exemplo 1: Troca entre dois canais

Para o processamento no canal 1 são utilizados de 6 eixos os seguintes eixos: 1., 2., 3. e 4º eixo.
Os 5º e 6º eixos são utilizados no canal 2 para a troca de peças de trabalho.

O eixo 2 deve ser trocado entre dois canais e depois do POWER ON ser associado ao canal 1.

Programa "MAIN" no canal 1:

Código de programa	**Comentário**
`INIT (2,"TROCA2")`	`; Selecionar o programa TROCA2 no canal 2.`
`N… START (2)`	`; Iniciar o programa no canal 2.`
`N… GET (AX2)`	`; Aceitar eixo AX2.`
`...`	
`N… RELEASE (AX2)`	`; Liberar eixo AX2.`
`N… WAITM (1,1,2)`	`; Espera pelo marcador WAIT no canal 1 e 2 para sincronização em ambos canais.`
`...`	`; Continuação do processo após a troca de eixos.`
`N… M30`	

Programa "TROCA2" no canal 2:

Programação	**Comentário**
`N… RELEASE(AX2)`	
`N160 WAITM(1,1,2)`	`; Espera pelo marcador WAIT no canal 1 e 2 para sincronização em ambos canais.`
`N150 GET(AX2)`	`; Aceitar eixo AX2.`
`...`	`; Continuação do processo após a troca de eixos.`
`N… M30`	

Exemplo 2: Troca de eixos sem sincronização

Se o eixo não precisa ser sincronizado, não é criada nenhuma parada de pré-processamento pelo GET.

Programação	**Comentário**
N01 G0 X0	
N02 RELEASE(AX5)	
N03 G64 X10	
N04 X20	
N05 GET(AX5)	; Se nenhuma sincronização for necessária, este não é um bloco executável.
N06 G01 F5000	; Nenhum bloco executável.
N07 X20	; Nenhum bloco executável, pois a posição X é igual no N04.
N08 X30	; Primeiro bloco executável após N05.
...	

Exemplo 3: Ativação de uma troca de eixos sem parada de pré-processamento

Requisito: A troca de eixos sem parada de pré-processamento deve ser configurada através de um dado de máquina.

Programação	**Comentário**
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP(B)	; O eixo B passa a ser eixo neutro.
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; O eixo não dispara nenhuma parada de pré-processamento/REORG.
N041 G4 F2	
N050 M5	
N099 M30	

Se o fuso ou eixo B for deslocado imediatamente após o bloco N023 como **eixo PLC** p. ex. desloca 180 graus e depois retorna até 1 grau, então este eixo novamente passa a ser um eixo neutro e não dispara nenhuma parada de pré-processamento no bloco N40.

Outras informações

Pré-requisitos para a troca de eixos

- O eixo deverá estar definido em dados da máquina para todos canais por ele utilizados.
- Através do dado de máquina específico de **eixo** deverá estar definido qual canal o eixo deve ser atribuído após POWER ON.

Descrição

Liberar eixo: RELEASE

Na liberação do eixo deve-se observar:

1. O eixo não pode participar de nenhuma transformação.
2. Nos acoplamentos de eixo (comando tangencial) todos eixos envolvidos precisam ser liberados.
3. Um eixo de posicionamento concorrente não pode ser trocado neste estado.
4. Em um eixo mestre Gantry também ocorre a troca para todos eixos escravos.
5. Em acoplamentos de eixo (movimento acoplado, acoplamento de valor mestre, transmissão eletrônica) somente pode ser liberado o eixo mestre do grupo.

Aceitar eixo: GET

Com este comando é executada a troca de eixos propriamente dita. A responsabilidade para o eixo está totalmente no canal onde foi programado o comando.

Efeitos do GET:

Troca de eixos com sincronização:

Um eixo sempre precisa ser sincronizado quando ele estava atribuído em um outro canal ou no PLC, e se antes do GET ocorreu uma sincronização pelo "WAITP", G74 ou se foi anulado o curso restante.

- Ocorre uma parada de pré-processamento (como no SOPRE).
- O processamento permanece interrompido até que a troca seja completamente executada.

"GET" automático

Se a princípio um eixo está disponível no canal, mas nesse momento não como "Eixo de canal", o GET é executado automaticamente. Se o eixo ou os eixos já estão sincronizados, não é criada nenhuma parada de pré-processamento.

Ajustar o comportamento de eixos de forma modificável

O momento de entrega dos eixos pode ser ajustado através de um dado de máquina da seguinte forma:

- A troca automática de eixos entre dois canais também é executada se o eixo tiver sido colocado em estado neutro (transferência convencional) com WAITP
- Na solicitação de uma rotação de contentor de eixo todos os eixos do contentor que estiverem atribuídos ao canal executado serão chamados no canal através do GET ou GETD implícito. Uma troca de eixos seguinte somente será permitida após a conclusão da rotação do contentor de eixo.
- Após um bloco intermediário colocado no processamento principal se controla se uma reorganização é ou não necessária. Uma reorganização somente se faz necessária quando os estados dos eixos deste bloco **não** coincidirem com os atuais estados dos eixos.
- Ao invés de um bloco GET com parada de pré-processamento e ação sincronizada entre o pré-processamento e o processamento principal, também pode ser realizada uma troca de eixos sem o pré-processamento. Então apenas se gera um bloco intermediário com a solicitação GET. No processamento principal, durante a execução deste bloco, se controla se os estados do eixo no bloco estão de acordo com os estados atuais de eixo.

Para mais informações sobre a funcionalidade de uma troca de eixos ou de fusos, veja o Manual de funções ampliadas; BAGs, Canais, Troca de eixos (K5).

2.16 Transferir eixo de outro canal (AXTOCHAN)

Com o comando de linguagem AXTOCHAN se pode solicitar um eixo que será transferido para outro canal. O eixo pode ser transferido ao canal correspondente tanto do programa de peça NC como de uma ação sincronizada.

Sintaxe

```
AXTOCHAN(nome de eixo,número de canal[,nome de eixo,número de
canal[,...]])
```

Significado

Elemento	Descrição
AXTOCHAN:	Solicitar eixo para um determinado canal
Nome de eixo:	Atribuição de eixos no sistema: X, Y, … ou especificação do nome de eixo de máquina envolvido. O canal a ser executado não precisa ser o próprio canal e ele também não precisa ser o canal que atualmente detém o direito de interpolação do eixo
Número de canal:	Número do canal que deve ser atribuído ao eixo

Indicação

Eixo de posicionamento concorrente e eixo controlado exclusivamente por PLC

Um eixo PLC não pode mudar de canal como eixo de posicionamento concorrente. Um eixo controlado exclusivamente pelo PLC não pode ser atribuído ao programa NC.

Literatura
Manual de funções ampliadas; Eixos de posicionamento (P2)

Exemplo

AXTOCHAN no programa NC

Os eixos X e Y são conhecidos no 1º canal e no 2º canal. Atualmente o canal 1 detém o direito de interpolação e no canal 1 se inicia o seguinte programa:

```
Código de programa                  Comentário
N110 AXTOCHAN(Y,2)                  ; Deslocar o eixo Y para o 2° canal.
N111 M0
N120 AXTOCHAN(Y,1)                  ; Trazer novamente o eixo Y (neutro).
N121 M0
N130 AXTOCHAN(Y,2,X,2)              ; Deslocar o eixo Y e o eixo X para o 2° canal (eixos
                                    neutros).
N131 M0
N140 AXTOCHAN(Y,2)                  ; Deslocar o eixo Y para o 2° canal (programa NC).
N141 M0
```

Outras informações

AXTOCHAN no programa NC

Neste caso, apenas com uma solicitação do eixo para o programa NC em canal próprio é que se executa um `GET` e com isso também se espera pela real alteração de estados. Se o eixo for solicitado para outro canal ou se ele deve ser um eixo neutro no próprio canal, então apenas será cancelada a solicitação.

AXTOCHAN a partir de uma ação sincronizada

Se um eixo for solicitado para o próprio canal, então o `AXTOCHAN` de uma ação sincronizada forma o `GET` a partir de uma ação sincronizada. Neste caso o eixo passa a ser o eixo neutro na primeira solicitação do próprio canal. Na segunda solicitação o eixo do programa NC é atribuído de forma análoga à solicitação GET no programa NC. Para solicitação GET a partir de uma ação sincronizada, veja o capítulo "Ações sincronizadas de movimentos".

2.17 Ativar dados de máquina (NEWCONF)

Com o comando NEWCONF, são ativados todos os dados da máquina. A função também pode ser ativada na interface de operação HMI através da tecla "Ativar MD".

Com a execução da função "NEWCONF" é executada uma parada de pré-processamento implícita, isto é, se interrompe o movimento de percurso.

Sintaxe

```
NEWCONF
```

Significado

NEWCONF:	Comando para tornar ativos todos os dados de máquina do nível de ativação "NEW_CONFIG"

Executar o NEWCONF a partir de um programa de peça entre vários canais

Se dados de máquina de eixos forem alterados a partir do programa de peça e, sem seguida, ativados com NEWCONF, então o NEWCONF ativa apenas os dados de máquina que realizam alterações para o canal do programa de peça.

Indicação

Para permitir que todas alterações sejam ativadas, se deve executar o comando NEWCONF em todo canal onde atualmente são calculados os eixos ou funções alterados que foram afetados pelos dados de máquina.

Com NEWCONF não se ativa nenhum dado de máquina de eixos.

Para eixos controlados por PLC deve ser executado um RESET de eixo.

Exemplo

Fresamento: Execução da posição de furação com diferentes tecnologias

Código de programa	Comentário
N10 $MA_CONTOUR_TOL[AX]=1.0	; Alterar o dado da máquina.
N20 NEWCONF	; Ativar o dado da máquina.
...	

2.18 Gravar arquivo (WRITE)

Com o comando `WRITE` é possível que blocos e dados de um programa NC sejam gravados em um arquivo que se situa no final de um sistema passivo de arquivos ou em uma memória de programa externa (arquivo de protocolo). Isto também pode ser feito com o programa que estiver em execução naquele momento.

Indicação

Um arquivo que deve ser gravado com o comando `WRITE` será recém criado, caso ele ainda não exista na memória do programa.

Pré-requisito

O atual nível de proteção ajustado deve ser igual ou mais alto do que o direito READ do arquivo. Se este não for o caso, o acesso será negado com mensagem de erro (valor retornado das variáveis com erro = 13).

Sintaxe

```
DEF INT <erro>
...
WRITE(<erro>,"<nome de arquivo>"/"<dispositivo externo>","<bloco/dados>")
```

Significado

`WRITE`:	Comando para adicionar um bloco ou dados no fim do arquivo indicado		
`<erro>`:	**Parâmetro 1:** Variável para o retorno do valor do erro		
	Tipo:	INT	
	Valor:	0	nenhum erro
		1	Caminho não permitido
		2	Caminho não encontrado
		3	Arquivo não encontrado
		4	Tipo de arquivo incorreto
		10	Arquivo está cheio
		11	Arquivo está sendo usado
		12	Nenhum recurso livre
		13	Sem direito de acesso
		14	EXTOPEN ausente ou com falha para o dispositivo de destino
		15	Erro durante a gravação no dispositivo externo
		16	programado caminho externo inválido

<nome do arquivo>:	**Parâmetro 2:** Nome do arquivo onde o bloco especificado ou os dados especificados devem ser adicionados	
	Tipo:	STRING
	Antes do nome efetivo do arquivo pode ser indicado o atalho absoluto. Sem a indicação do caminho, o arquivo será procurado no atual diretório (=diretório do programa selecionado). Regras para a indicação de atalho, consultar "Endereçamento dos arquivos da memória do programa (Página 227)".	
<dispositivo externo>:	Se os dados tiverem que ser transferidos para um dispositivo/arquivo externo através da função "Process DataShare", deve ser indicado, ao invés do nome do arquivo, o identificador simbólico para o dispositivo/arquivo externo a ser aberto.	
	Tipo:	STRING
	Para mais informações, veja em "Process DataShare - Saída para um dispositivo/arquivo externo (EXTOPEN, WRITE, EXTCLOSE) (Página 689)". **Nota:** O identificador deve ser idêntico ao identificador especificado no comando EXTOPEN.	
<bloco/dados>:	**Parâmetro 3:**Bloco ou dados que devem ser adicionados no arquivo indicado.	
	Tipo:	STRING

Indicação

Durante a gravação no sistema passivo de arquivos em uma memória de programa externa, o comando WRITE insere de maneira implícita um caractere "LF" (LINE FEED = quebra de linha) no final da string de saída.

Para a saída no dispositivo/arquivo externo através da função "Process DataShare" este comportamento não se aplica. Se for necessário emitir um "LF", isto deverá ser informado de maneira implícita na String de saída.

→ Para isso veja o exemplo 3: LF implícito/explícito!

Condições gerais

- **Tamanho de arquivo máximo (→ Fabricante da máquina!)**
  O tamanho de arquivo máximo possível para arquivos de protocolo no sistema passivo de arquivos é configurado com o dado de máquina:
  MD11420 $MN_LEN_PROTOCOL_FILE
  O tamanho de arquivo máximo vale para todos os arquivos que foram criados no sistema passivo de arquivos com o comando WRITE. No caso de ser excedido, aparece uma mensagem de erro e o bloco ou os dados não serão armazenados. Enquanto houver espaço de memória, ainda se pode criar um novo arquivo.

Exemplos

Exemplo 1: Comando WRITE no sistema passivo de arquivos sem indicação absoluta do caminho

Código de programa	Comentário
N10 DEF INT ERROR	;Definição das variáveis de erro.

Código de programa	Comentário
N20 WRITE(ERROR,"PROT","PROTOCOLO DE 7.2.97")	; Grave o texto "PROTOCOLO DE 7.2.97" no arquivo _N_PROT_MPF.
N30 IF ERROR	; Avaliação de erro.
N40 MSG ("Erro no comando WRITE:" <<ERROR)	
N50 M0	
N60 ENDIF	
...	

Exemplo 2: Comando WRITE no sistema passivo de arquivos com indicação absoluta do caminho

```
Código de programa
...
WRITE(ERROR,"/_N_WCS_DIR/_N_PROT_WPD/_N_PROT_MPF","PROTOCOLO DE
7.2.97")
...
```

Exemplo 3: "LF" implícito/explícito

a, gravação no sistema passivo de arquivos com "LF" gerada de maneira implícita

```
Código de programa
...
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
```

Resultado da emissão:

MY_STRING

MY_STRING

b, gravação em arquivo externo sem "LF" gerada de maneira implícita

```
Código de programa
...
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)
N270 M30
```

Resultado da emissão:

MY_STRINGMY_STRING

c, gravação em arquivo externo com "LF" programada de maneira explícita

Para conseguir o mesmo resultado como em a, deve-se programar o seguinte:

Código de programa

```
...
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
```

Resultado da emissão:

MY_STRING

MY_STRING

2.19 Deletar arquivo (DELETE)

Com o comando `DELETE` se pode deletar todos os arquivos, independente deles serem criados ou não com o comando `WRITE`. Mesmo os arquivos atribuídos com um nível de acesso mais alto podem ser deletados com `DELETE`.

Sintaxe

```
DEF INT <erro>
DELETE(<erro>,"<nome de arquivo>")
```

Significado

`DELETE`:	Comando para deletar o arquivo indicado		
`<erro>`:	Variável para o retorno do valor do erro		
	Tip.	INT	
	Valor:	0	nenhum erro
		1	Caminho não permitido
		2	Caminho não encontrado
		3	Arquivo não encontrado
		4	Tipo de arquivo incorreto
		11	Arquivo está sendo usado
		12	Nenhum recurso livre
		20	Outros erros
`<nome do arquivo>`:	Nome do arquivo que deve ser deletado		
	Tipo:	STRING	
	Antes do nome efetivo do arquivo pode ser indicado o atalho absoluto. Sem a indicação do caminho, o arquivo será procurado no atual diretório (=diretório do programa selecionado). Regras para a indicação de atalho, consultar "Endereçamento dos arquivos da memória do programa (Página 227)".		

Exemplo

Código de programa	**Comentário**
`N10 DEF INT ERROR`	`;Definição das variáveis de erro.`
`N15 STOPRE`	`; Parada de pré-processamento.`
`N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")`	`Deleta o arquivo TEST1 no diretório de sub-rotinas.`
`N30 IF ERROR`	`; Avaliação de erro.`
`N40 MSG ("Erro no comando DELETE:" <<ERROR)`	
`N50 M0`	
`N60 ENDIF`	

2.20 Ler linhas no arquivo (READ)

O comando `READ` lê uma ou mais linhas no arquivo especificado e armazena as informações lidas em um campo de tipo STRING. Cada linha lida ocupa um elemento deste campo.

Pré-requisito

O atual nível de proteção ajustado deve ser igual ou mais alto do que o direito READ do arquivo. Se este não for o caso, o acesso será negado com mensagem de erro (valor retornado das variáveis com erro = 13).

Sintaxe

```
DEF INT <erro>
DEF STRING[<tamanho de String>] <resultado>[<n>,<m>]
READ(<erro>,"<nome de arquivo>",<linha inicial>,<número de
linhas>,<resultado>)
```

Significado

`READ`:	Comando para leitura de linhas do arquivo especificado e para armazenamento destas linhas em um campo de variável.		
`<erro>`:	Variável para o retorno do valor de erro (parâmetro Call-By-Reference)		
	Tip.	INT	
	Valor:	0	nenhum erro
		1	Caminho não permitido
		2	Caminho não encontrado
		3	Arquivo não encontrado
		4	Tipo de arquivo incorreto
		11	Arquivo está sendo usado
		13	Direito de acesso insuficiente
		21	Linha não existente (parâmetro `<linha inicial>` ou `<número de linhas>` maior que o número de linhas no arquivo indicado).
		22	O tamanho de campo das variáveis de resultado (`<resultado>`) é muito pequeno.
		23	Área de linhas muito grande (o parâmetro `<número de linhas>` deve ser configurado de modo que seja permitida a leitura em toda extensão até o fim do arquivo).
`<nome do arquivo>`:	Nome do arquivo que deve ser lido (parâmetro Call-By-Value)		
	Tipo:	STRING	
	Antes do nome efetivo do arquivo pode ser indicado o atalho absoluto. Sem a indicação do caminho, o arquivo será procurado no atual diretório (=diretório do programa selecionado). Regras para a indicação de atalho, consultar "Endereçamento dos arquivos da memória do programa (Página 227)".		

<linha inicial>:	Linha inicial da área do arquivo que deve ser lida (parâmetro Call-By-Value)		
	Tipo:	INT	
	Valor:	0	Antes do fim do arquivo é realizada a leitura do número de linhas especificado com o parâmetro <número de linhas>.
		1 … n	Número da primeira linha a ser lida.
<número de linhas>:	Número de linhas que devem ser lidas (parâmetro Call-By-Value)		
	Tipo:	INT	
<Resultado>:	Variável de resultado (parâmetro Call-By-Reference) Campo de variável onde o texto lido é armazenado.		
	Tipo:	STRING (comprimento máximo: 255)	
	Se no parâmetro <número de linhas> forem especificadas menos linhas que o tamanho do campo [<n>,<m>] das variáveis de resultado, então os elementos de campo restantes não serão alterados. A terminação de uma linha com o caractere de controle "LF" (Line Feed) ou "CR LF" (Carriage Return Libe Feed) **não** é armazenada na variável de resultado. As linhas lidas serão truncadas se a linha for maior que o tamanho de String definido. Não se emite nenhuma mensagem de erro.		

Indicação

Arquivos binários não podem ser lidos. É retornado o erro "tipo de arquivo incorreto" (valor de retorno da variável de erro = 4). Não se pode ler os seguintes tipos de arquivo: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Exemplo

Código de programa	Comentário
N10 DEF INT ERROR	;Definição das variáveis de erro.
N20 DEF STRING[255] RESULT[5]	;Definição das variáveis de resultado.
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF", 1,5,RESULT)	; Nome de arquivo com identificadores de domínio e indicação de atalho.
N40 IF ERROR <>0	; Avaliação de erro.
N50 MSG("ERRO"<<ERROR<<"COM COMANDO READ")	
N60 M0	
N70 ENDIF	
...	

2.21 Controle de presença de um arquivo (ISFILE)

Com o comando `ISFILE`-pode ser verificada a existência de um arquivo na memória do programa.

Sintaxe

```
<Resultado>=ISFILE("<Nome do arquivo>")
```

Significado

`ISFILE`:	Comando para a verificação da existência de um arquivo		
`<Nome do arquivo>`:	Nome do arquivo, cuja existência deve ser verificada.		
	Tipo:	STRING	
	Antes do nome efetivo do arquivo pode ser indicado o atalho absoluto. Sem a indicação do caminho, o arquivo será procurado no atual diretório (=diretório do programa selecionado). Regras para a indicação de atalho, consultar "Endereçamento dos arquivos da memória do programa (Página 227)".		
`<Resultado>`:	Variável de resultado para receber o resultado do controle		
	Tip.	BOOL	
	Valor:	TRUE	Arquivo presente
		FALSE	Arquivo não presente

Exemplos

Exemplo 1

Código de programa	**Comentário**
`N10 DEF BOOL RESULT`	`; Definição das variáveis de resultado.`
`N20 RESULT=ISFILE("TESTFILE")`	
`N30 IF(RESULT==FALSE)`	
`N40  MSG("ARQUIVO NÃO PRESENTE")`	
`N50  M0`	
`N60 ENDIF`	
`...`	

Exemplo 2

Código de programa	**Comentário**
`N10 DEF BOOL RESULT`	`; Definição das variáveis de resultado.`
`N20 RESULT=ISFILE("TESTFILE")`	
`N30 IF(NOT ISFILE("TESTFILE"))`	
`N40  MSG("ARQUIVO NÃO PRESENTE")`	
`N50  M0`	
`N60 ENDIF`	
`...`	

2.22 Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Através dos comandos `FILEDATE`, `FILETIME`, `FILESIZE`, `FILESTAT` e `FILEINFO` é possível ler (extrair) determinadas informações de arquivo como data / horário do último acesso de gravação, atual tamanho de arquivo, estado de arquivo ou a soma destas informações.

Pré-requisito

O atual nível de proteção ajustado deve ser igual ou mais alto do que o direito Show do diretório de um nível acima. Se este não for o caso, o acesso será negado com mensagem de erro (valor retornado das variáveis com erro = 13).

Sintaxe

```
FILE....(<Erro>,"<Nome do arquivo>",<Resultado>)
```

Significado

`FILEDATE`:	Fornece a **Data** do último acesso ao arquivo		
`FILETIME`:	Retorna o **horário** do último acesso de gravação do arquivo		
`FILESIZE`:	Fornece o **tamanho atual** de um arquivo		
`FILESTAT`:	Fornece o **Status** de um arquivo referente os seguintes **direitos** : • Leitura (r: read) • Gravação (w: write) • Execução (x: execute) • Exibição (s: show) • Deleção (d: delete) **Nota:** estes estágios de proteção são características especiais do sistema passivo de arquivos. Ao acessar a memória externa do programa, o `FILESTAT` fornece, portanto, apenas os direitos de acesso Default (77777).		
`FILEINFO`:	Retorna para um arquivo o **soma das informações**, que são lidas através de `FILEDATE`, `FILETIME`, `FILESIZE` e `FILESTAT`		
`<erro>`:	Variável para o retorno do valor de erro (parâmetro Call-By-Reference)		
	Tip.	VAR INT	
	Valor:	0	nenhum erro
		1	Caminho não permitido
		2	Caminho não encontrado
		3	Arquivo não encontrado
		4	Tipo de arquivo incorreto
		13	Direito de acesso insuficiente
		22	O tamanho de String das variáveis de resultado (`<resultado>`) é muito pequeno.

<nome do arquivo>:	Nome do arquivo de onde devem ser lidas (extraídas) as informações de arquivo			
	Tipo:	CHAR[160]		
	Antes do nome efetivo do arquivo pode ser indicado o atalho absoluto. Sem a indicação do caminho, o arquivo será procurado no atual diretório (=diretório do programa selecionado). Regras para a indicação de atalho, consultar "Endereçamento dos arquivos da memória do programa (Página 227)".			
<Resultado>:	Variável de resultado (parâmetroCall-By-Reference) Variável onde a informação de arquivo solicitada é armazenada.			
	Tipo:	VAR CHAR[8]	no	FILEDATE Formato: "dd.mm.yy"
		VAR CHAR[8]	no	FILETIME Formato: "hh:mm.ss"
		VAR INT	no	FILESIZE O tamanho de arquivo é retornado em Bytes.
		VAR CHAR[5]	no	FILESTAT Formato: "rwxsd" (r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete)
		VAR CHAR[32]	no	FILEINFO Formato: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

Exemplo

Código de programa	**Comentário**
N10 DEF INT ERROR	; Definição das variáveis de erro.
N20 STRING[32] RESULT	; Definição das variáveis de resultado.
N30 FILEINFO(ERROR,"/_N_MPF_DIR/_N_TESTFILE_MPF",RESULT)	Nome de arquivo com identificadores de domínio e de arquivo e indicação do caminho.
N40 IF ERROR <> 0	; Avaliação de erro
N50 MSG("ERRO"<<ERROR<<"NO COMANDO FILEINFO")	
N60 M0	
N70 ENDIF	
...	

P. ex., o exemplo poderia retornar nas seguintes variáveis de resultado RESULT:

"77777 12345678 26.05.00 13:51:30"

2.23 Arredondamento (ROUNDUP)

Com a função "ROUNDUP" se pode arredondar os valores de entrada de tipo REAL (números fracionados com ponto decimal) para o próximo número inteiro maior.

Sintaxe

```
ROUNDUP(<valor>)
```

Significado

ROUNDUP:	Comando para arredondar um valor de entrada
<valor>:	Valor de entrada de tipo REAL

Indicação

Valores de entrada de tipo INTEGER (um número inteiro) são retornados sem alteração.

Exemplos

Exemplo 1: Diversos valores de entrada e seus resultados de arredondamento

Exemplo	Resultado de arredondamento
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

Exemplo 2: ROUNDUP no programa NC

Código de programa

```
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 Técnica de subrotinas

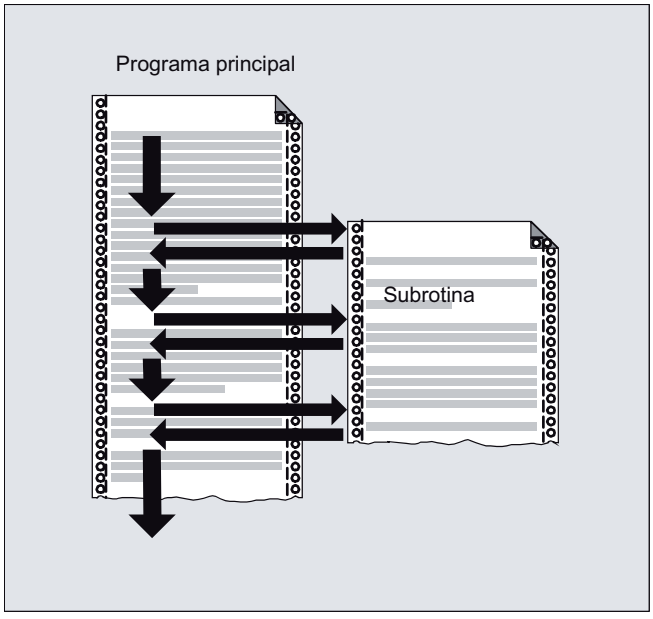
2.24.1 Informações gerais

2.24.1.1 Subrotina

A denominação "sub-rotina" é uma herança do tempo em que se dividia os programas de peças em programas principais e sub-rotinas. No caso, os programas principais eram os programas de peça que eram selecionados no comando para execução e depois era dada a partida dos mesmos. As sub-rotinas eram os programas de peça que eram chamados a partir do programa principal.

Esta divisão não existe mais na atual linguagem NC do SINUMERIK. Cada programa de peça pode, em princípio, ser selecionado e iniciado como programa principal, ou então ser chamado como sub-rotina a partir de outro programa de peça.

Com isso, nos demais procedimentos, uma sub-rotina é denominada como um programa de peça que pode ser chamado a partir de outro programa de peça.



Aplicação

Como em todas as linguagens de programação de nível mais alto, na linguagem NC as sub-rotinas também são aplicadas para armazenar (comportar) programas fechados e independentes, que utilizam partes de programa múltiplas vezes.

As sub-rotinas oferecem as seguintes vantagens:

- Melhoram a clareza e leitura dos programas
- Melhoram a qualidade através do reaproveitamento de partes de programa já testados

- Oferecem a opção para aquisição de bibliotecas específicas de usinagem
- Economizam espaço na memória

2.24.1.2 Nomes de subrotina

Regras de designação

O nome da sub-rotina pode ser selecionado livremente, respeitando-se as seguintes regras:

- Caracteres permitidos:
  - Letras: A ... Z, a ... z
  - Números: 0 ... 9
  - Traço inferior: _
- Os primeiros dois caracteres devem ser duas letras ou um traço inferior seguido por uma letra.

  Indicação

  Quando esta condição estiver preenchida, então um programa NC, através do nome de programa, pode ser chamado como subrotina por outro programa. Se o nome do programa for iniciado com números, então não será possível realizar a chamada de subrotina através da instrução CALL.

- Comprimento máximo: 24 caracteres

Indicação

Letras maiúsculas / minúsculas

Na linguagem NC do SINUMERIK **não** é feita nenhuma distinção entre as letras maiúsculas e as letras minúsculas.

Indicação

Nomes de programas não permitidos

Para evitar problemas com os aplicativos do windows, **não** é permitido o uso dos seguintes nomes de programas:

- CON, PRN, AUX, NUL
- COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9
- LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

Extensões internas do comando

O nome da sub-rotina atribuído durante a criação do programa é ampliado internamente no comando com um prefixo e um sufixo:

- Prefixo: `_N_`
- Pós-fixo: `_SPF`

Utilização do nome de programa

Para a utilização do nome de programa, por exemplo em uma chamada de sub-rotina, são possíveis todas as combinações de prefixo, nome de programa e sufixo.

Exemplo:

A sub-rotina com o nome de programa `SUB_PROG` pode ser acessada através dos seguintes identificadores:

1. `SUB_PROG`
2. `_N_SUB_PROG`
3. `SUB_PROG_SPF`
4. `_N_SUB_PROG_SPF`

Igualdade de nomes nos programas principais e sub-rotinas

Se existir um programa principal (.MPF) e uma sub-rotina (.SPF) com o mesmo nome de programa, no caso de utilização do nome do programa no programa NC deve ser indicado junto a respectiva extensão de arquivo para a identificação inequívoca. Caso contrário, será utilizado o programa, que for localizado primeiro com o nome de programa indicado ao longo do atalho de busca.

2.24.1.3 Aninhamento de subrotinas

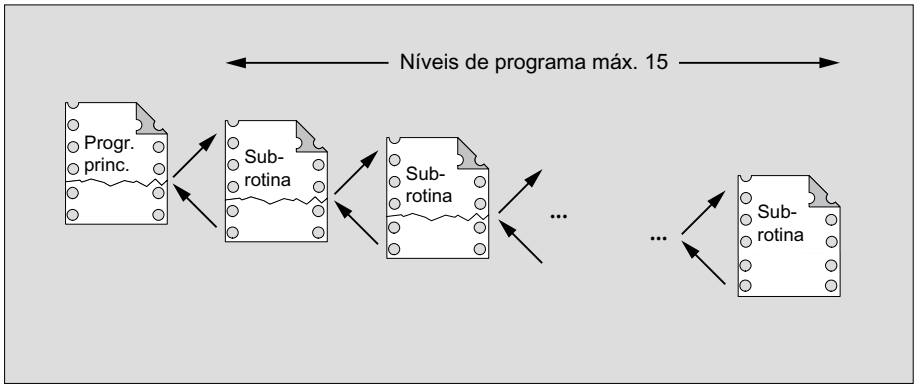
Um programa principal pode chamar subrotinas, que por sua vez podem chamar outras subrotinas. Dessa forma as execuções dos programas estão agrupadas uma dentro da outra. Por isso que cada programa é processado em um nível de programa próprio.

Nível de agrupamento

Atualmente a linguagem NC coloca 16 níveis de programa à disposição. O programa principal sempre é processado no nível de programa mais alto, o nível 0. Uma subrotina sempre é processada no próximo nível de programa mais baixo a partir da chamada. Assim o nível de programa 1 é o primeiro nível de subrotina.

Divisão dos níveis de programa:

- Nível de programa 0: Nível de programa principal
- Nível de programa 1 - 15: Nível de subrotina 1 - 15



Rotinas de interrupção (ASUP)

Se uma subrotina é chamada durante uma rotina de interrupção, esta não será processada no atual nível de programa (n) ativo no canal, mas no próximo nível de programa mais baixo (n+1). Para que isto também seja possível no nível de programa mais baixo, estão disponíveis 2 níveis de programa extras (16 e 17) quando relacionados com rotinas de interrupção.

Se forem necessários mais que 2 níveis de programa, isto deve ser considerado de modo explícito na estruturação do programa de peça executado no canal. Isto significa que somente pode ser utilizado um número de níveis de programa que ainda proporcione a quantidade de níveis de programa suficiente para a execução da interrupção.

Por exemplo, se a execução da interrupção precisar de 4 níveis de programa, o programa de peça deve ser estruturado de modo que ele ocupe no máximo até o nível de programa 13. Quando ocorre uma interrupção, estes 4 níveis de programa (14 até 17) necessários estarão disponíveis.

Ciclos da Siemens

Os ciclos da Siemens requerem 3 níveis de programa. Por isso que a chamada de um ciclo da Siemens deve ocorrer no máximo até:

- Na execução do programa de peça: Nível de programa 12
- Na rotina de interrupção: Nível de programa 14

2.24.1.4 Caminho de localização

No caso de chamada de uma sub-rotina sem a indicação do atalho, o comando examinará as memórias de programa existentes de acordo com uma sequência de busca predefinida (consultar "Atalho de busca na chamada de subrotinas (Página 232)").

2.24.1.5 Parâmetros formais e parâmetros atuais

Os parâmetros formais e parâmetros atuais estão relacionados com a definição e a chamada de sub-rotinas com transferência de parâmetros.

Parâmetro formal

Durante a definição de uma sub-rotina os parâmetros que devem ser transferidos à sub-rotina, os chamados parâmetros formais, devem ser definidos com tipo e nome de parâmetro.

Com isso os parâmetros formais definem a interface da sub-rotina.

Exemplo:

Código de programa	**Comentário**
`PROC CONTORNO (REAL X, REAL Y)`	`; Parâmetro formal: X e Y ambos do tipo REAL`
`N20 X1=X Y1=Y`	`; Deslocamento do eixo X1 até a posição X e do eixo Y1 até a posição Y`
`...`	
`N100 RET`	

Parâmetro atual

Durante a chamada de uma sub-rotina devem ser transferidos à sub-rotina valores absolutos ou variáveis, chamados de parâmetros atuais.

Com isso os parâmetros atuais alimentam os valores atuais durante a chamada da interface da sub-rotina.

Exemplo:

Código de programa	**Comentário**
`N10 DEF REAL LARGURA`	`; Definição de variáveis`
`N20 LARGURA=20.0`	`; Atribuição de variáveis`
`N30 CONTORNO(5.5, LARGURA)`	`; Chamada de sub-rotina com parâmetros atuais: 5.5 e LARGURA`
`...`	
`N100 M30`	

2.24.1.6 Transferência de parâmetros

Definição de uma sub-rotina com transferência de parâmetros

A definição de uma sub-rotina com transferência de parâmetros é realizada com a palavra-chave `PROC` e uma listagem completa de todos os parâmetros esperados pela sub-rotina.

Transferência incompleta de parâmetros

Na chamada da sub-rotina nem sempre precisam ser transferidos de modo explícito todos os parâmetros definidos na interface da sub-rotina. Quando um parâmetro é desconsiderado, para este parâmetro é transferido o valor padrão "0".

Entretanto, para uma identificação única da ordem dos parâmetros sempre devem ser indicadas as vírgulas como caracteres de separação dos parâmetros. Uma exceção é o último parâmetro. Se este não for considerado na chamada, também pode ser descartada a última vírgula.

Exemplo:

Sub-rotina:

Código de programa	Comentário
`PROC SUB_PROG (REAL X, REAL Y, REAL Z)`	`; Parâmetro formal: X, Y e Z`
`...`	
`N100 RET`	

Programa principal:

Código de programa	Comentário
`PROC MAIN_PROG`	
`...`	
`N30 SUB_PROG(1.0,2.0,3.0)`	`; Chamada de sub-rotina com transferência de parâmetros completa:` `X=1.0, Y=2.0, Z=3.0`
…	
`N100 M30`	

Exemplos para a chamada de sub-rotina no `N30` com transferência de parâmetros incompleta:

```
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
```

ATENÇÃO

Transferência de parâmetros Call-by-Reference

Os parâmetros que são transferidos através de Call-by-Reference, não podem ser desconsiderados na chamada de sub-rotina.

ATENÇÃO

Tipo de dados AXIS

Os parâmetros do tipo de dados AXIS não podem ser desconsiderados na chamada de sub-rotina.

Verificação dos parâmetros de transferência

Através da variável de sistema $P_SUBPAR [n] com n = 1, 2, ... pode ser verificado na sub-rotina se um parâmetro foi transferido de modo explícito ou se ele foi desconsiderado. O índice n refere-se à ordem dos parâmetros formais. O índice n = 1 refere-se ao 1º parâmetro formal, o índice n = 2 ao 2º parâmetro formal, e assim por diante.

O seguinte segmento de programa mostra no exemplo do 1º parâmetro formal, como que uma verificação pode ser realizada:

Programação	**Comentário**
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; Parâmetro formal: X, Y e Z
N20 IF $P_SUBPAR[1]==TRUE	; Verificação do 1º parâmetro formal X.
...	; Estas ações são executadas, quando o parâmetro formal X foi transferido explicitamente.
N40 ELSE	
...	; Estas ações são executadas, quando o parâmetro formal X não for transferido.
N60 ENDIF	
...	; Ações gerais
N100 RET	

2.24.2 Definição de uma subrotina

2.24.2.1 Subrotina sem transferência de parâmetros

Para a definição de sub-rotinas sem transferência de parâmetros pode ser descartada a linha de definição no início do programa.

Sintaxe

```
[PROC <nome de programa>]
...
```

Significado

PROC:	Instrução de definição no início de um programa
<nome de programa>:	Nome do programa

Exemplo

Exemplo 1: Sub-rotina com instrução PROC

Código de programa	**Comentário**
PROC SUB_PROG	; Linha de definição

Código de programa	Comentário
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; salto de retorno da sub-rotina

Exemplo 2: Sub-rotina sem instrução `PROC`

Código de programa	Comentário
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; salto de retorno da sub-rotina

Veja também

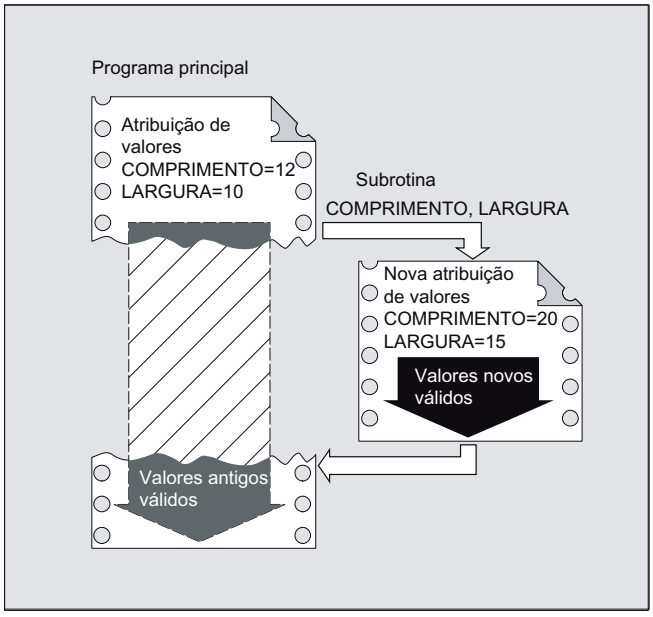
Chamada de subrotina sem transferência de parâmetros (Página 197)

2.24.2.2 Subrotina com transferência de parâmetros Call-by-Value (PROC)

A definição de uma sub-rotina com transferência de Call-by-Value é realizada com a palavra-chave `PROC` seguida de nomes de programas e uma listagem completa de todos os parâmetros, com indicação de tipo e nome. A instrução de definição deve estar na primeira linha do programa.

Call-by-Value

O programa de chamada transmite apenas o valor de uma variável à sub-rotina em uma transferência de parâmetros Call-by-Value. Assim, o subprograma não recebe acesso direto à variável. Desse modo, só o valor visível da sub-rotina é alterado em caso de mudança do valor do parâmetro. O valor das variáveis definidas no programa de chamada permanece inalterado. A transferência de parâmetros Call-by-Value não tem com isso nenhuma reação sobre o programa de chamada.



Sintaxe

```
PROC <nome de programa> (<tipo de parâmetro> <nome de
parâmetro>=<valor_inicialização>, ...)
```

Indicação

Podem ser transferidos até 127 parâmetros.

Significado

`PROC`:	Instrução de definição no início de um programa
`<nome de programa>`:	Nome do programa
`<tipo de parâmetro>`:	Tipo de dados do parâmetro (p. ex. REAL, INT, BOOL)
`<nome de parâmetro>`:	Nome do parâmetro
`<valor_inicialização>`:	Valor opcional para inicialização do parâmetro (**opcional**) Se não for especificado nenhum parâmetro na chamada do subprograma, então o valor de inicialização do parâmetro será o determinado.

Exemplos

Exemplo 1

Definição de uma sub-rotina SUB_PROG com 3 parâmetros do tipo REAL com valores Default:

Código de programa

```
PROC SUB_PROG(REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

Exemplo 2

Diferentes variantes de chamada

Código de programa

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ;variantes de chamada
  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
```

Veja também

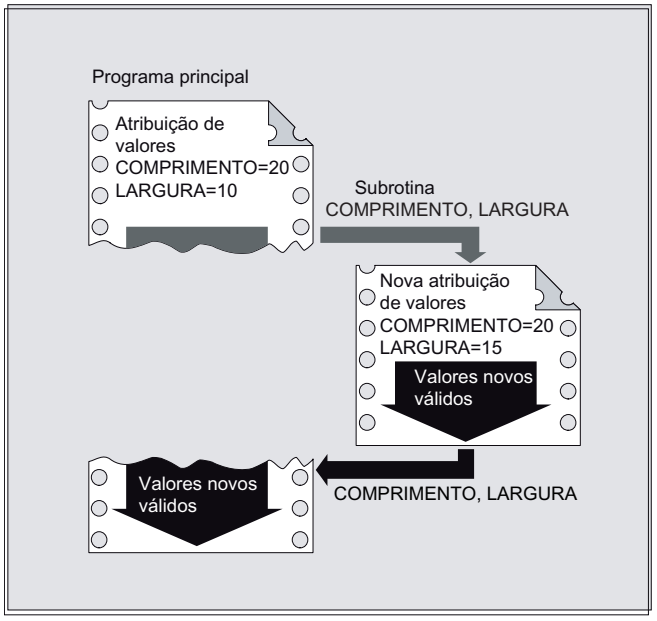
Chamada de subrotina com transferência de parâmetros (EXTERN) (Página 199)

2.24.2.3 Subrotina com transferência de parâmetros Call-by-Reference (PROC, VAR)

A definição de uma sub-rotina com transferência de Call-by-Reference é realizada com a palavra-chave `PROC`, seguida de nomes de programa e uma listagem completa de todos os parâmetros com a palavra-chave`VAR`, tipo e nome. A instrução de definição deve estar na primeira linha do programa. Como parâmetro também podem ser transmitidas as referências nos campos.

Call-by-Reference

O programa de chamada não transmite o valor de uma variável em uma transferência de parâmetros Call-by-Reference, mas sim uma referência (indicador). Isto dá ao sub-programa o acesso direto à variável. Desta forma, não só o valor visível da subrotina será alterado em uma mudança do valor do parâmetro, mas também o valor da variável definida no programa de chamada. Assim, a transferência de parâmetros Call-by-Reference tem reações sobre o programa de chamada, mesmo após finalização do subprograma.



Indicação

A transferência de parâmetros Call-by-Reference somente é necessária se a variável de transferência for definida localmente no programa de chamada (LUD). As variáveis globais de canal ou as variáveis globais do NC não precisam ser transferidas, pois estas podem ser acessadas diretamente pela sub-rotina.

Sintaxe

```
PROC <nome de programa> (VAR <tipo de parâmetro> <nome de
parâmetro>, ...)
PROC <nome de programa> (VAR <tipo de campo> <nome de campo>
[<m>,<n>,<o>], ...)
```

Indicação

Podem ser transferidos até 127 parâmetros.

Significado

PROC:	Instrução de definição no início de um programa
VAR:	Palavra-chave para a transferência de parâmetros por referência
<nome de programa>:	Nome do programa
<tipo de parâmetro>:	Tipo de dados do parâmetro (p. ex. REAL, INT, BOOL)
<nome de parâmetro>:	Nome do parâmetro
<tipo de campo>:	Tipo de dados dos elementos de campo (p. ex. REAL, INT, BOOL)
<nome de campo>:	Nome do campo

[<m>,<n>,<o>]:	Tamanho de campo Atualmente, no máximo são possíveis campos de 3 dimensões:	
	<m>:	Tamanho de campo para 1ª Dimensão
	<n>:	Tamanho de campo para 2ª Dimensão
	<o>:	Tamanho de campo para 3ª Dimensão

Indicação

- O nome de programa informado após a palavra-chave `PROC` deve coincidir com o nome de programa dado na interface de operação.
- Com campos de tamanho indefinido como parâmetros formais se pode processar sub-rotinas de campos de tamanho variável. Para isto, não é dado na definição, p. ex. de um campo bidimensional como parâmetro formal, o comprimento da 1ª dimensão. Porém, a vírgula precisa ser escrita.
  Exemplo: `PROC <nome de programa> (VAR REAL CAMPO[ ,5])`

Exemplo

Definição de um subprograma com dois parâmetros como referência ao tipo REAL:

Código de programa

```
; Parâmetro 1: Referência ao tipo: REAL, nome: COMPRIMENTO
; Parâmetro 2: Referência ao tipo: REAL, nome: LARGURA
PROC SUB_PROG(VAR REAL COMPRIMENTO, VAR REAL LARGURA)
```

Veja também

Chamada de subrotina com transferência de parâmetros (EXTERN) (Página 199)

2.24.2.4 Salvar funções G modais (SAVE)

O atributo `SAVE` faz com que todos os comandos modais G ativos antes da chamada da sub-rotina sejam salvas e novamente ativadas após o fim da sub-rotina.

ATENÇÃO

Interrupção do modo de controle da trajetória

Se com o modo de controle da trajetória ativo uma sub-rotina for chamada com o atributo `SAVE`, o modo de controle da trajetória é interrompido no fim da sub-rotina (salto de retorno).

Sintaxe

```
PROC <nome da sub-rotina> SAVE
```

Significado

SAVE:	Salvamento dos comandos G antes da chamada da sub-rotina e seu restabelecimento após o fim da sub-rotina

Exemplo

Na sub-rotina CONTORNO atua o comando modal G G91 (dimensão incremental). No programa principal atua o comando modal G G90 (dimensão absoluta). Através da definição de sub-rotina com o SAVE o G90 torna-se novamente ativo no programa principal após o fim da sub-rotina.

Definição de sub-rotina:

Código de programa	Comentário
PROC CONTORNO (REAL VALOR1) SAVE	; Definição de sub-rotina com parâmetro SAVE
N10 G91 ...	; Comando modal G G91: Dimensão incremental
N100 M17	; Fim de sub-rotina

Programa principal:

Código de programa	Comentário
N10 G0 X... Y... G90	; Comando modal G G90: Dimensão absoluta
N20 ...	
...	
N50 CONTORNO (12.4)	;chamada de sub-rotina
N60 X... Y...	; Comando modal G G90 reativado através do SAVE

Condições gerais

Frames

A relação dos Frames em relação às sub-rotinas com o atributo SAVE depende do tipo do Frame e pode ser ajustada através de dados de máquina.

Literatura

Manual de funcionamento das funções básicas, eixos, sistemas de coordenadas, Frames (K2),
capítulo: "Salto de retorno de sub-rotinas com SAVE"

2.24.2.5 Supressão de processamento bloco a bloco (SBLOF, SBLON)

Supressão de bloco a bloco para o programa inteiro

Os programas identificados com SBLOF são executados inteiros, mesmo com o processamento bloco a bloco ativo, ou seja, o processamento bloco a bloco é suprimido em todo o programa.

SBLOF está na linha PROC e vale até o fim ou cancelamento da sub-rotina. Com o comando de salto de retorno se decide se a parada deve ser feita no fim da sub-rotina ou não:

Salto de retorno com M17: Parada no fim da sub-rotina
Salto de retorno com RET: Nenhuma parada no fim da sub-rotina

Supressão de bloco a bloco dentro do programa

O SBLOF deve constar isolado em um bloco. A partir deste bloco se desativa o bloco a bloco até:

- o próximo SBLON ou
  ou
- o fim do nível ativo da sub-rotina

Sintaxe

Supressão de bloco a bloco para o programa inteiro:
```
PROC ... SBLOF
```

Supressão de bloco a bloco dentro do programa:

```
SBLOF
...
SBLON
```

Significado

PROC:	Primeira instrução de um programa
SBLOF:	Comando para desativação do processamento bloco a bloco O SBLOF pode estar em um bloco PROC ou isolado em um bloco.
SBLON:	Comando para ativar o processamento bloco a bloco O SBLON deve estar em um bloco próprio.

Condições gerais

- **Supressão de bloco a bloco e exibição de bloco**
  A atual exibição de bloco pode ser ocultada em ciclos/sub-rotinas com `DISPLOF`. Se o `DISPLOF` for programado junto com o `SBLOF`, então será exibida a chamada do ciclo/sub-rotina na parada do bloco a bloco durante o ciclo/sub-rotina.
- **Supressão de bloco a bloco na ASUP de sistema na ASUP de usuário**
  Se a parada de bloco a bloco na ASUP de sistema ou de usuário for suprimida através das configurações no dado de máquina MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (Bit0 = 1 e Bit1 = 1), então a parada de bloco a bloco pode ser ativada novamente através da programação do `SBLON` na ASUP.
  Se a parada de bloco a bloco for suprimida na ASUP de usuário através da configuração no dado de máquina MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP, então a parada de bloco a bloco **não** pode ser ativada novamente através da programação de `SBLON` na ASUP.
- **Particularidades da supressão de bloco a bloco nos diversos tipos de processamento bloco a bloco**
  Com o processamento bloco a bloco SBL2 (parada após cada bloco de programa de peça) ativo **não** se executa a parada no bloco do `SBLON` se no MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (evitar parada de bloco a bloco) o Bit 12 estiver ajustado em "1".
  Com o processamento bloco a bloco SBL3 (parada após cada bloco de programa de peça e também no ciclo) se suprimir o comando `SBLOF`.

Exemplos

Exemplo 1: Supressão de bloco a bloco dentro de um programa

```
Código de programa            Comentário
N10 G1 X100 F1000
N20 SBLOF                     ; Desativar o bloco individual
N30 Y20
N40 M100
N50 R10=90
N60 SBLON                     ; Ativar novamente o bloco individual
N70 M110
N80 ...
```

A área entre `N20` e `N60` é processada como um passo em modo bloco a bloco.

Exemplo 2: O ciclo deve atuar como um comando para o usuário

Programa principal:

```
Código de programa
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
```

Código de programa
N50 M30

Ciclo CYCLE1:

Código de programa	**Comentário**
N100 PROC CYCLE1 DISPLOF SBLOF	; Supressão do bloco individual
N110 R10=3*SIN(R20)+5	
N120 IF (R11 <= 0)	
N130 SETAL(61000)	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

O ciclo CYCLE1 é executado com processamento bloco a bloco ativo, isto é, deve-se pressionar uma vez a tecla Start para execução do CYCLE1.

Exemplo 3: Uma ASUP iniciada pelo PLC para a ativação do deslocamento modificado do ponto zero e para correções da ferramenta não deverá estar visível.

Código de programa
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

Exemplo 4: Com MD10702 Bit 12 = 1 são se executa a parada

Situação inicial:

- O processamento bloco a bloco está ativo.
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

Programa principal:

Código de programa	Comentário
N10 G0 X0	; Parar nesta linha do programa de peças.
N20 X10	; Parar nesta linha do programa de peças.
N30 CYCLE	; Bloco de deslocamento gerado pelo ciclo.
N50 G90 X20	; Parar nesta linha do programa de peças.
M30	

Ciclo CYCLE:

Código de programa	Comentário
PROC CYCLE SBLOF	; Suprimir a parada do bloco individual
N100 R0 = 1	
N110 SBLON	; Devido a MD10702 Bit12=1 não será parado nesta linha do programa de peças.
N120 X1	; Nesta linha do programa de peças será parado.
N140 SBLOF	
N150 R0 = 2	
RET	

Exemplo 5: Supressão de bloco a bloco com aninhamento de programas

Situação inicial:

O processamento bloco a bloco está ativo.

Aninhamento de programas:

Código de programa	Comentário
N10 X0 F1000	; Neste bloco se executa a parada.
N20 UP1(0)	
PROC UP1(INT _NR) SBLOF	; Suprimir a parada do bloco individual.
N100 X10	
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	
N210 SBLON	; Ativar parada do bloco individual.
N220 X22	; Neste bloco se executa a parada.
N230 UP3(0)	
PROC UP3(INT _NR)	
N300 SBLOF	; Suprimir a parada do bloco individual.
N305 X30	
N310 SBLON	; Ativar parada do bloco individual.

Código de programa	Comentário
N320 X32	; Neste bloco se executa a parada.
N330 SBLOF	; Suprimir a parada do bloco individual.
N340 X34	
N350 M17	; SBLOF está ativo.
N240 X24	; Neste bloco se executa a parada. SBLON está ativo.
N250 M17	; Neste bloco se executa a parada. SBLON está ativo.
N120 X12	
N130 M17	; Neste bloco de salto de retorno será executada a parada. O SBLOF da instrução PROC está ativo.
N30 X0	; Neste bloco se executa a parada.
N40 M30	; Neste bloco se executa a parada.

Outras informações

Bloqueio de bloco a bloco para sub-rotinas assíncronas

Para executar uma ASUP em um passo no modo bloco a bloco, na ASUP deve ser programada uma instrução `PROC` com `SBLOF`. Isto também se aplica para a função "ASUP editável de sistema" (MD11610 $MN_ASUP_EDITABLE).

Exemplo para uma ASUP editável de sistema:

Código de programa	Comentário
N10 PROC ASUP1 SBLOF DISPLOF	
N20 IF $AC_ASUP=='H200'	
N30 RET	; Sem REPOS na mudança de modo de operação.
N40 ELSE	
N50 REPOSA	; REPOS em todos os demais casos.
N60 ENDIF	

Controles de programa em bloco a bloco

No processamento bloco a bloco o usuário pode executar o programa de peça por blocos. Existem os seguintes tipos de ajuste:

- SBL1: Bloco a bloco IPO com parada após cada bloco de função da máquina.
- SBL2: Bloco a bloco com parada após cada bloco.
- SBL3: Parada no ciclo (com a seleção do SBL3 se suprime o comando `SBLOF`).

Supressão de bloco a bloco com aninhamento de programas

Se em uma sub-rotina foi programado `SBLOF` na instrução `PROC`, então a parada é feita com `M17` no salto de retorno da sub-rotina. Com isso evita-se que seja executado o próximo bloco do programa de chamada. Se em uma sub-rotina com `SBLOF` for ativada uma supressão de bloco a bloco sem `SBLOF` na instrução `PROC`, então a parada somente será realizada após o

próximo bloco de função da máquina do programa chamado. Se isto não for desejado, então na sub-rotina se deve programar novamente o SBLON, ainda antes do retorno (M17). A parada não será realizada no caso de um salto de retorno com RET para um programa de nível superior.

2.24.2.6 Supressão da atual exibição de bloco (DISPLOF, DISPLON, ACTBLOCNO)

Como padrão, na exibição de bloco se exibe o atual bloco de programa. A exibição do atual bloco pode ser ocultada nos ciclos e nas sub-rotinas com o comando DISPLOF. Ao invés do atual bloco se exibe a chamada do ciclo ou da sub-rotina. Com o comando DISPLON cancela-se novamente a supressão da exibição de bloco.

O DISPLOF e o DISPLON é programado na linha de programa com a instrução PROC e tem efeito sobre toda a sub-rotina e de forma implícita para todas as sub-rotinas chamadas por esta sub-rotina, que não tiverem o comando DISPLON ou DISPLOF. Este comportamento também é válido para ASUPs.

Sintaxe

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

Significado

DISPLOF:	Comando para ocultar a atual exibição de bloco.	
	Posicionamento:	No fim da linha de programa com a instrução PROC
	Efeito:	Até o salto de retorno da sub-rotina ou fim de programa.
	Nota: Se mais sub-rotinas forem chamadas a partir da sub-rotina com o comando DISPLOF, então nesta também se oculta a atual exibição de bloco, isto se nesta não for programado explicitamente o DISPLON.	
DISPLON:	Comando para cancelar a supressão da atual exibição de bloco	
	Posicionamento:	No fim da linha de programa com a instrução PROC
	Efeito:	Até o salto de retorno da sub-rotina ou fim de programa.
	Nota: Se mais sub-rotinas forem chamadas a partir da sub-rotina com o comando DISPLON, então nesta também se exibe o atual bloco de programa, isto se neste não for programado explicitamente o DISPLOF.	
ACTBLOCNO:	O DISPLOF junto com o atributo ACTBLOCNO faz com que, em caso de um alarme, seja retornado o número do atual bloco em que ocorreu o alarme. Isto também se aplica se em um nível de programa mais baixo apenas estiver programado o DISPLOF. Em contrapartida, com DISPLOF sem ACTBLOCNO, o número do bloco da chamada do ciclo ou da sub-rotina não é exibido no último nível de programa marcado com DISPLOF.	

Exemplos

Exemplo 1: Ocultar a atual exibição de bloco no ciclo

Código de programa	Comentário
PROC CYCLE(AXIS TOMOV, REAL POSITION) SAVE DISPLOF	; Suprimir a indicação do bloco atual. O invés disso, deve ser exibida a chamada do ciclo, p. ex.: CYCLE(X,100.0)
DEF REAL DIFF	; Conteúdo dos ciclos
G01 ...	
...	
RET	; Salto de retorno da sub-rotina. Na exibição de bloco se exibe o bloco seguinte à chamada do ciclo.

Exemplo 2: Exibição de bloco na emissão de alarme

Sub-rotina SUBPROG1 (com ACTBLOCNO):

Código de programa	Comentário
PROC SUBPROG1 DISPLOF ACTBLOC- NO	
N8000 R10 = R33 + R44	
...	
N9040 R10 = 66 X100	; Disparar o alarme 12080
...	
N10000 M17	

Sub-rotina SUBPROG2 (sem ACTBLOCNO):

Código de programa	Comentário
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; Disparar o alarme 12080
...	
N7000 M17	

Programa principal:

Código de programa	Comentário
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; Emissão de alarme = "12080 canal K1 bloco N9040 erro de sintaxe no texto R10="

Código de programa	Comentário
N2060 ...	
N2350 SUBPROG2	; Emissão de alarme = "12080 canal K1 bloco N2350 erro de sintaxe no texto R10="
...	
N3000 M30	

Exemplo 3: Cancelamento da supressão da atual exibição de blocos

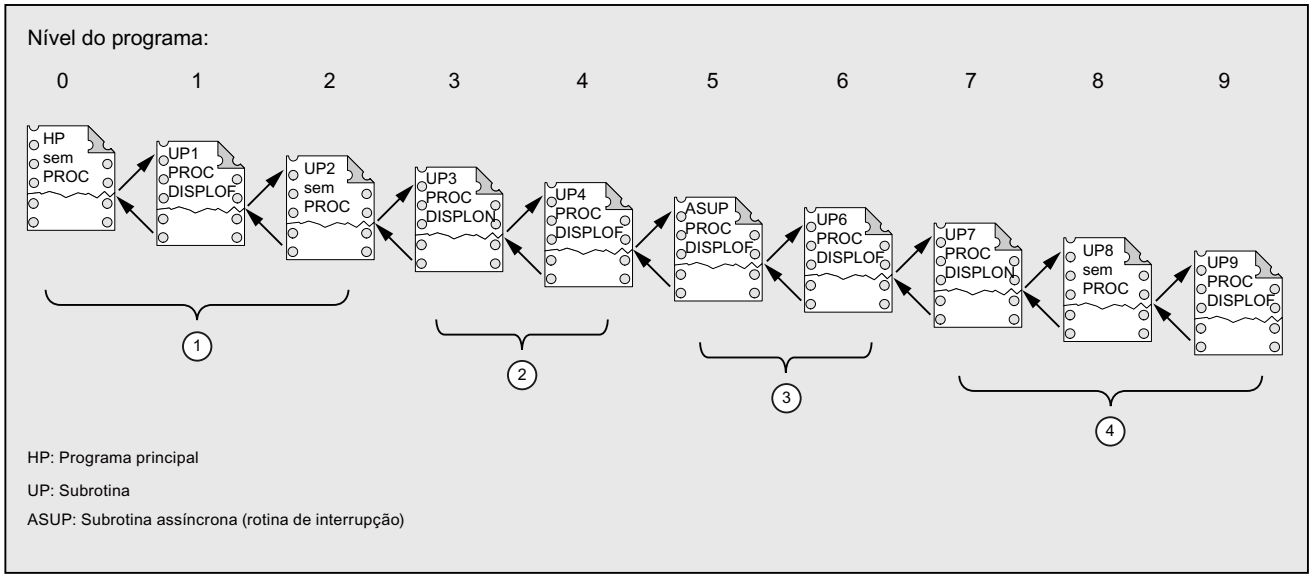
Sub-rotina SUB1 com supressão:

Código de programa	Comentário
PROC SUB1 DISPLOF	; Suprimir a indicação atual do bloco na sub-rotina SUB1. Ao invés disso, o bloco deve ser exibido com a chamada da SUB1.
...	
N300 SUB2	; Acessar a sub-rotina SUB2.
...	
N500 M17	

Sub-rotina SUB2 sem supressão:

Código de programa	Comentário
PROC SUB2 DISPLON	; Revogar a supressão da indicação atual do bloco na sub-rotina SUB2.
...	
N200 M17	; Salto de retorno para a sub-rotina SUB1. Na SUB1 é novamente suprimida a atual exibição de bloco.

Exemplo 4: Comportamento de exibição em diferentes combinações do DISPLON/DISPLOF



① Na atual exibição de bloco são mostradas as linhas de programa de peça do nível de programa 0.
② Na atual exibição de bloco são mostradas as linhas de programa de peça do nível de programa 3.
③ Na atual exibição de bloco são mostradas as linhas de programa de peça do nível de programa 3.
④ Na atual exibição de bloco são mostradas as linhas de programa de peça do nível de programa 7/8.

2.24.2.7 Identificar subrotinas com preparação (PREPRO)

Com a palavra-chave `PREPRO` todos os arquivos podem ser identificados no fim da linha da instrução `PROC` durante a inicialização.

Indicação

Este tipo de preparação de programa depende do ajuste do dado de máquina correspondente. Observar as informações do fabricante da máquina.

Literatura:
Manual de funcionamento, funções especiais; Pré-processamento (V2)

Sintaxe

```
PROC … PREPRO
```

Significado

`PREPRO`:	Palavra-chave para identificação de todos arquivos preparados na inicialização, os programas NC armazenados nos diretórios dos ciclos

Carregar sub-rotinas com preparação e chamada de sub-rotina

Tanto as sub-rotinas preparadas com parâmetros na inicialização como na chamada de sub-rotina os diretórios dos ciclos são tratados na mesma ordem:

1. _N_CUS_DIR ciclos de usuário
2. _N_CMA_DIR ciclos do fabricante
3. _N_CST_DIR ciclos padrão

No caso de programas NC de mesmo nome e diferente formação, ativa-se a primeira instrução `PROC` encontrada e a outra instrução `PROC` será saltada sem mensagem de alarme.

2.24.2.8 Salto de retorno para a subrotina M17

No fim de uma sub-rotina existe um comando de salto de retorno `M17` (ou o comando de fim de programa de peça `M30`). Ele executa o salto de retorno para o programa de chamada no bloco de programa de peça após a chamada da sub-rotina.

Indicação

O `M17` e o `M30` são tratados com o mesmo valor na linguagem NC.

Sintaxe

```
PROC <nome de programa>
...
M17/M30
```

Condições gerais

Efeito do salto de retorno para sub-rotina no modo de controle da trajetória

Se o `M17` (ou o `M30`) estiver sozinho no bloco de programa de peça, então um modo de controle da trajetória que estiver ativo no canal será interrompido.

Para evitar que o modo de controle da trajetória seja interrompido, o `M17` (ou o `M30`) deve ser escrito no último bloco de deslocamento. Adicionalmente, o seguinte dado de máquina deve ser ajustado para "0":

MD20800 $MC_SPF_END_TO_VDI = 0 (sem saída de M30/M17 para a interface NC/PLC)

Exemplo

1. Sub-rotina com `M17` em um bloco próprio

Código de programa	Comentário
`N10 G64 F2000 G91 X10 Y10`	
`N20 X10 Z10`	
`N30 M17`	`; Salto de retorno com interrupção do modo de controle da trajetória.`

2. Sub-rotina com `M17` no último bloco de deslocamento

Código de programa	Comentário
`N10 G64 F2000 G91 X10 Y10`	
`N20 X10 Z10 M17`	`; Salto de retorno sem interrupção do modo de controle da trajetória.`

2.24.2.9 Salto de retorno para subrotina RET

Como substituto para o comando de salto de retorno `M17` também pode ser utilizado na sub-rotina o comando `RET`. O `RET` deve ser programado em um bloco de programa de peça próprio. Como o `M17` o `RET` executa o salto de retorno para o programa de chamada no bloco de programa de peça após a chamada da sub-rotina.

Indicação

Através da programação de parâmetros é possível alterar o comportamento do salto de retorno do `RET` (veja "Salto de retorno parametrizável da subrotina (RET ...) (Página 187)").

Aplicação

A instrução `RET` deve ser utilizada quando um modo de controle da trajetória G64 (`G641` ... `G645`) não deve ser interrompido através do salto de retorno.

Pré-requisito

O comando `RET` somente pode ser utilizado em sub-rotinas que não foram definidas com o atributo `SAVE`.

Sintaxe

```
PROC <nome de programa>
...
RET
```

Exemplo

Programa principal:

Código de programa	Comentário
`PROC MAIN_PROGRAM`	`; Início do programa`
`...`	
`N50 SUB_PROG`	`Chamada de sub-rotina: SUB_PROG`
`N60 ...`	
`...`	

Código de programa	Comentário
N100 M30	; fim do programa

Sub-rotina:

Código de programa	Comentário
PROC SUB_PROG	
...	
N100 RET	O salto de retorno é realizado no bloco N60 no programa principal.

2.24.2.10 Salto de retorno parametrizável da subrotina (RET ...)

Geralmente, com o comando `RET` é efetuado um salto de retorno a partir de uma sub-rotina para o programa a ser acessado. O processamento então é prosseguido em uma linha de programa seguinte à chamada da sub-rotina. Se o processamento do programa tiver que ser prosseguido em outro local, são disponibilizadas as seguintes possibilidades:

- Continuação da execução do programa após a chamada do ciclo de desbaste em modo de dialeto ISO (após a descrição do contorno).
- Salto de retorno ao programa principal a partir de qualquer nível da sub-rotina (também após a ASUP) para tratamento de erros.
- Salto de retorno através de vários níveis de programa para aplicações especiais em ciclos de compilação e em modo de dialeto ISO.

Para isto o comando `RET` deve ser programado com outros parâmetros.

Direção da busca

Com a indicação do parâmetro `<bloco de destino>`, o salto de retorno ocorre primeiro para o bloco posterior ao bloco de chamada. Em seguida, o destino é buscado no sentido do **final** do programa, para o qual foi efetuado o salto de retorno. Se a busca não for bem sucedida, a próxima busca será efetuada no sentido do início do programa.

Sintaxe

```
RET("<bloco de destino>")
RET("<bloco de destino>",<bloco após bloco de destino>)
RET("<bloco de destino>",<bloco após bloco de
destino>,<número de níveis de salto de retorno>)
RET("<bloco de destino>", ,<número de níveis de salto de retorno>)
RET("<bloco de destino>",<bloco após bloco de
destino>,<número de níveis de salto de retorno>),
<salto de retorno ao início do programa>)
RET( , ,<número de níveis de salto de retorno>,
<salto de retorno ao início do programa>)
```

Significado

RET:	Fim de sub-rotina		
<bloco de destino>:	No parâmetro é indicado o bloco como destino do salto, no qual deve ser prosseguido o processamento do programa. Se o parâmetro <número de planos do salto de retorno> não tiver sido programado, então o destino do salto encontra-se no programa, a partir do qual a sub-rotina atual foi acessada. As possíveis indicações são:		
	<número do bloco>	Número do bloco de destino. A busca pelo número do bloco ocorre no programa para o qual é efetuado o salto de retorno, primeiro no sentido do final do programa.	
	<marcador de salto>	O marcador de salto deve estar presente no programa para o qual é efetuado o salto de retorno. A busca pelo marcador de salto ocorre no programa para o qual é efetuado o salto de retorno, primeiro no sentido do final do programa.	
	<sequência de caracteres>	Sequência de caracteres que deve estar presente no programa para o qual é efetuado o salto de retorno (por ex., nome do programa ou nome da variável). A busca pela sequência de caracteres ocorre no programa para o qual é feito o salto de retorno, primeiro no sentido do final do programa. Para a programação da sequência de caracteres no bloco de destino são aplicadas as seguintes regras: • **Espaço vazio no fim** (em contrapartida ao marcador de salto, que é identificado com um ":" no final). • **Antes da sequência de caracteres** somente pode ser colocado um número de bloco e/ou um marcador de salto, **nenhum comando de programa**.	
<bloco após bloco de destino>:	No parâmetro é indicado, se o processamento do programa deve ser prosseguido no bloco indicado no parâmetro bloco de <destino> ou no bloco seguinte.		
	Tipo:	INT	
	Valor:	0	O salto de retorno é realizado para o bloco indicado no parâmetro <bloco de destino>.
		> 0	O salto de retorno é realizado para o bloco seguinte ao indicado no parâmetro <bloco de destino>.

<número de níveis de saltos de retorno>:	No parâmetro é indicada a quantidade de níveis de programa, sobre a qual deve ser efetuado o salto de retorno, para buscar ali o bloco de destino e prosseguir com o processamento do programa.		
	Tipo:	INT	
	Valor:	1	O programa é continuado "no atual nível do programa - 1" (como RET sem parâmetros).
		2	O programa é continuado no "atual nível do programa - 2", isto é, é realizado o salto de um nível.
		3	O programa é continuado no "atual nível do programa - 3", isto é, é realizado o salto de dois níveis.
		...	
	Faixa de valores:	1 ... 15	
<salto de retorno ao início do programa>:	No parâmetro é indicado se durante um salto de retorno para o programa principal e o **modo de dialeto ISO** ativo ali, o programa deve ser prosseguido no início do programa.		
	Tipo:	BOOL	
	Valor:	1	Se o salto de retorno for realizado no programa principal e ali estiver ativo um **modo de dialeto ISO**, é realizada a bifurcação para o início do programa.

Indicação

Em um salto de retorno da sub-rotina com uma sequência de caracteres, como indicação para a localização de destino sempre se busca por um marcador de salto no programa de chamada.

Se um destino de salto deve ser definido claramente através de uma sequência de caracteres, a sequência de caracteres não pode coincidir com o nome de um marcador de salto, pois o salto de retorno da sub-rotina sempre será executado para o marcador de salto e não para a sequência de caracteres (veja o exemplo 2).

Condições gerais

Para o salto de retorno através de vários níveis de programa são avaliadas as instruções SAVE dos diversos níveis de programa.

Se em um retorno através de vários níveis de programa uma sub-rotina modal estiver ativa, e, se em uma das sub-rotinas saltadas foi programado o comando de desseleção `MCALL` para a sub-rotina modal, então a sub-rotina modal ainda permanece ativa.

ATENÇÃO
erro de programação No caso de um salto de retorno através de vários níveis de programa, é de responsabilidade própria do usuário providenciar para que seja prosseguido com as configurações modais necessárias. Isto é obtido, por exemplo, com a programação de um bloco principal correspondente.

Exemplos

Exemplo 1: Continuação no programa principal após a execução da ASUP

Programação	**Comentário**
N10010 CALL "UP1"	; Nível de programa 0 (programa principal)
N11000 PROC UP1	; Nível de programa 1
N11010 CALL "UP2"	
N12000 PROC UP2	; Nível de programa 2
...	
N19000 PROC ASUP	; Nível de programa 3 (execução da ASUP)
...	
N19100 RET("N10900", ,$P_STACK)	; Salto de retorno da sub-rotina para o programa principal
	; $P_STACK: Nível atual do programa
N10900	; Bloco de destino no programa principal
N10910 MCALL	; Desligar a chamada de sub-rotina modal
N10920 G0 G60 G40 M5	; Inicializar outras configurações modais

Exemplo 2: Sequência de caracteres (<String>) como indicação para a localização do bloco de destino

Programa principal:

Código de programa	**Comentário**
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; Chamada da sub-rotina "subProg1"
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; Chamada da sub-rotina "subProg2"
N1410 M3 S500 Y20	
N1420 ..	

Código de programa	Comentário
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1: G1 X30	; Aqui p "subProg1" está definido como marcador de salto.
N1560 ...	
N1600 subProg3	; Chamada da sub-rotina "subProg3"
N1610 ...	
N1900 M30	

Sub-rotina subProg1:

Código de programa	Comentário
PROC subProg1	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; Salto de retorno ao programa principal no bloco N1400

Sub-rotina subProg2:

Código de programa	Comentário
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; Salto de retorno ao programa principal no bloco N1500

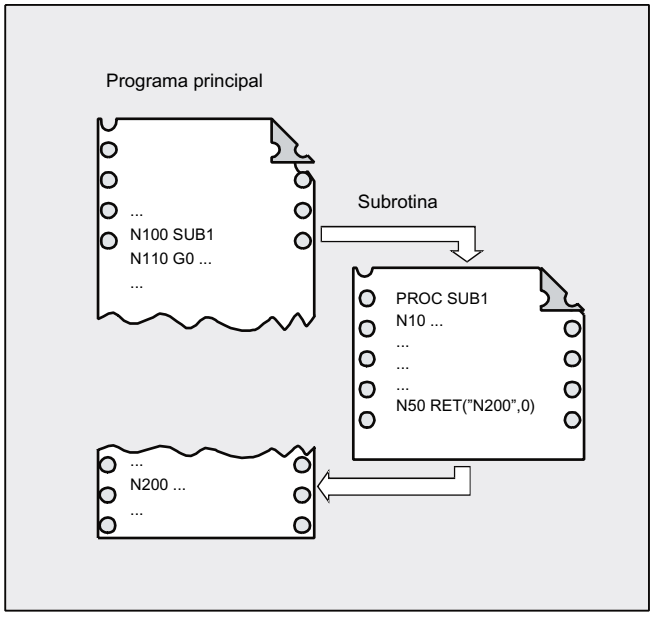
Sub-rotina subProg3:

Código de programa	Comentário
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; Salto de retorno ao programa principal no bloco N1550

Outras informações

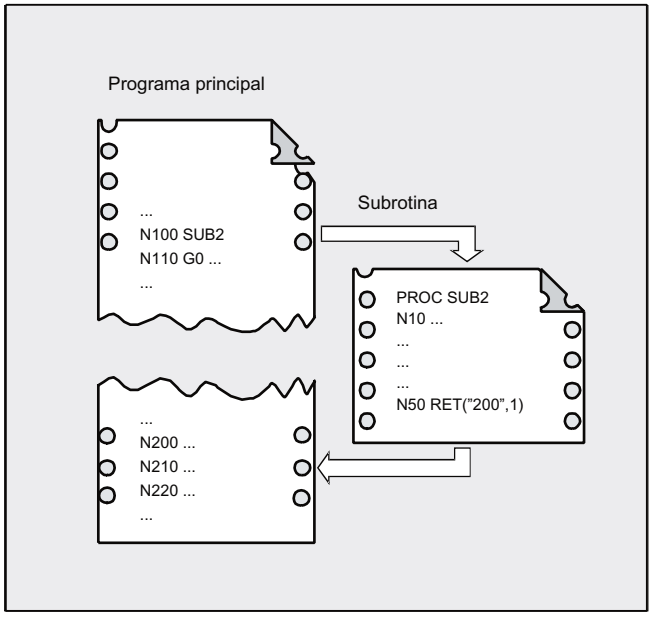
Os gráficos seguintes mostram os diversos efeitos do parâmetro de salto de retorno

1. <Bloco de destino> = "N200", <Bloco após o bloco de destino> = 0



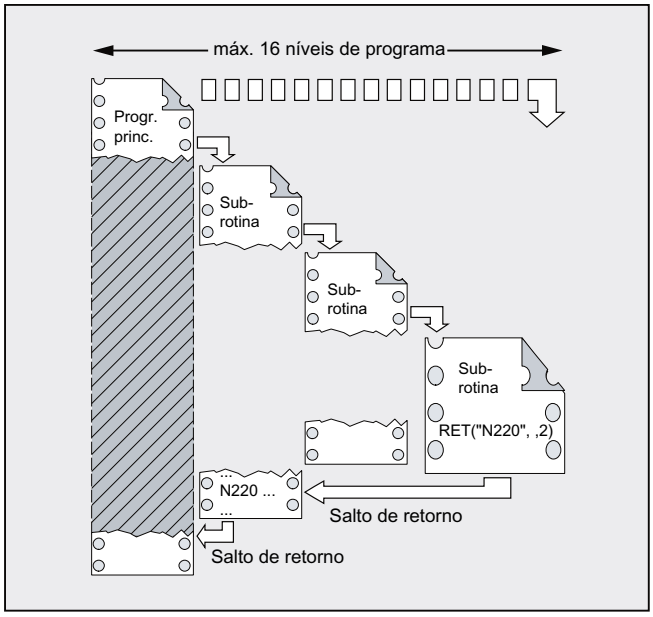
Após o comando `RET` é realizada a continuação da execução do programa com o bloco `N200` no programa principal.

2. <Bloco de destino> = "N200", <Bloco após o bloco de destino> = 1



Após o comando `RET` é realizada a continuação da execução do programa com o bloco (`N210`) que segue o bloco `N200` no programa principal.

3. <Bloco de destino> = "N220", <Quantidade de níveis de salto de retorno> = 2



Após o comando `RET` se realiza o salto retornando dois níveis e a execução do programa é realizada com o bloco `N220`.

2.24.2.11 Salto de retorno parametrizável da subrotina (RET ...)

Geralmente, com o comando `RETB` é efetuado um salto de retorno a partir de uma sub-rotina para o programa a ser acessado. O processamento então é prosseguido em uma linha de programa seguinte à chamada da sub-rotina. Se o processamento do programa tiver que ser prosseguido em outro local, são disponibilizadas as seguintes possibilidades:

- Continuação da execução do programa após a chamada do ciclo de desbaste em modo de dialeto ISO (após a descrição do contorno).
- Salto de retorno ao programa principal a partir de qualquer nível da sub-rotina (também após a ASUP) para tratamento de erros.
- Salto de retorno através de vários níveis de programa para aplicações especiais em ciclos de compilação e em modo de dialeto ISO.

Para isto o comando `RETB` deve ser programado com outros parâmetros.

Direção da busca

Com a indicação do `<bloco de destino`, o salto de retorno ocorre primeiro para o bloco posterior ao bloco de chamada. Em seguida, o destino é buscado no sentido do **início** do programa, para o qual é efetuado o salto de retorno. Se a busca não for bem sucedida, a próxima busca será efetuada no sentido do final do programa.

Sintaxe

```
RETB("<bloco de destino>")
RETB("<bloco de destino>",<bloco posterior ao bloco de destino>)
```

```
RETB("<bloco de destino>",<bloco posterior ao bloco de
destino>,<número de planos do salto de retorno>)
RETB("<bloco de destino>", ,<número de planos do salto de retorno>)
RETB("<bloco de destino>",<bloco posterior ao bloco de
destino>,<número de planos do salta de retorno>,
<salto de retorno ao início do programa>)
RETB( , ,<número de planos do salto de retorno>,<salto de retorno ao
início do programa>)
```

Significado

RETB:	Fim de sub-rotina	
<bloco de destino>:	No parâmetro é indicado o bloco como destino do salto, no qual deve ser prosseguido o processamento do programa. Se o parâmetro <número de planos do salto de retorno> não tiver sido programado, então o destino do salto encontra-se no programa, a partir do qual a sub-rotina atual foi acessada. As possíveis indicações são:	
	<número do bloco>	Número do bloco de destino. A busca pelo número do bloco ocorre no programa para o qual é efetuado o salto de retorno, primeiro no sentido do início do programa.
	<marcador de salto>	O marcador de salto deve estar presente no programa para o qual é efetuado o salto de retorno. A busca pelo marcador de salto ocorre no programa para o qual é efetuado o salto de retorno, primeiro no sentido do início do programa.
	<sequência de caracteres>	Sequência de caracteres que deve estar presente no programa para o qual é efetuado o salto de retorno (por ex., nome do programa ou nome da variável). A busca pela sequência de caracteres ocorre no programa para o qual é efetuado o salto de retorno, primeiro no sentido do início do programa. Para a programação da sequência de caracteres no bloco de destino são aplicadas as seguintes regras: • **Espaço vazio no fim** (em contrapartida ao marcador de salto, que é identificado com um ":" no final). • **Antes da sequência de caracteres** somente pode ser colocado um número de bloco e/ou um marcador de salto, **nenhum comando de programa**.

<bloco após bloco de destino>:	No parâmetro é indicado, se o processamento do programa deve ser prosseguido no bloco indicado no parâmetro bloco de <destino> ou no bloco seguinte.		
	Tipo:	INT	
	Valor:	0	O salto de retorno é realizado para o bloco indicado no parâmetro <bloco de destino>.
		> 0	O salto de retorno é realizado para o bloco seguinte ao indicado no parâmetro <bloco de destino>.
<número de níveis de saltos de retorno>:	No parâmetro é indicada a quantidade de níveis de programa, sobre a qual deve ser efetuado o salto de retorno, para buscar ali o bloco de destino e prosseguir com o processamento do programa.		
	Tipo:	INT	
	Valor:	1	O programa é continuado "no atual nível do programa - 1" (como `RET` sem parâmetros).
		2	O programa é continuado no "atual nível do programa - 2", isto é, é realizado o salto de um nível.
		3	O programa é continuado no "atual nível do programa - 3", isto é, é realizado o salto de dois níveis.
		...	
	Faixa de valores:	1 ... 15	
<salto de retorno ao início do programa>:	No parâmetro é indicado se durante um salto de retorno para o programa principal e o **modo de dialeto ISO** ativo ali, o programa deve ser prosseguido no início do programa.		
	Tipo:	BOOL	
	Valor:	1	Se o salto de retorno for realizado no programa principal e ali estiver ativo um **modo de dialeto ISO**, é realizada a bifurcação para o início do programa.

Indicação

Em um salto de retorno da sub-rotina com uma sequência de caracteres, como indicação para a localização de destino sempre se busca por um marcador de salto no programa de chamada.

Se um destino de salto deve ser definido claramente através de uma sequência de caracteres, a sequência de caracteres não pode coincidir com o nome de um marcador de salto, pois o salto de retorno da sub-rotina sempre será executado para o marcador de salto e não para a sequência de caracteres (veja o exemplo 2).

Condições gerais

Para o salto de retorno através de vários níveis de programa são avaliadas as instruções `SAVE` dos diversos níveis de programa.

Se em um retorno através de vários níveis de programa uma sub-rotina modal estiver ativa, e, se em uma das sub-rotinas saltadas foi programado o comando de desseleção `MCALL` para a sub-rotina modal, então a sub-rotina modal ainda permanece ativa.

ATENÇÃO
erro de programação No caso de um salto de retorno através de vários níveis de programa, é de responsabilidade própria do usuário providenciar para que seja prosseguido com as configurações modais necessárias. Isto é obtido, por exemplo, com a programação de um bloco principal correspondente.

Exemplo

Código de programa	**Comentário**
EXEMPLO.MPF	
…	
N3000 START_CYC(parâm1, parâm2, …)	
N3010 TECH_CYC1(parâm1, parâm2, …)	
N3020 TECH_CYC2(parâm1, parâm2, …)	
N3030 TECH_CYC3(parâm1, parâm2, …)	
N3040 END_CYC(parâm1, parâm2, …)	
N3040 **END_CYC**(parâm1, parâm2, …)	
N3050 …	
N4500 START_CYC(parâm11, parâm12, …)	
N4510 …	
N4590 END_CYC(parâm11, parâm12, ..)	
N5000 …	
…	
N6000 M30	

Código de programa	**Comentário**
PROC	; Chamada no programa principal linha N3040
N10000 …	
N15000 if status == 1	
N15010 RETB(“START_CYC”)	; Salto de retorno para o programa a ser acessado EXEMPLO.MPF
	; Busca pela sequência de caracteres "START_CYC"
	; Direção da busca: em retorno no sentido do início do programa
	; Continuação do processamento do programa com a linha N3000
N15020 endif	
N15030 if status == 0	

Código de programa	Comentário
N15040 RET	; Salto de retorno para o programa a ser acessado EXEMPLO.MPF
	; Continuação do processamento do programa com a linha N3050
N15050 endif	
N16000 RET("START_CYC")	; Salto de retorno para o programa a ser acessado EXEMPLO.MPF
	; Busca pela sequência de caracteres "START_CYC"
	; Direção da busca: para frente no sentido do final do programa
	; Continuação do processamento do programa com a linha N4500
N17060 RETB	; Salto de retorno para o programa a ser acessado EXEMPLO.MPF
	; Continuação do processamento do programa com a linha N3050
	; RETB sem o parâmetro é idêntico ao RET

2.24.3 Chamada de uma subrotina

2.24.3.1 Chamada de subrotina sem transferência de parâmetros

A chamada de uma sub-rotina é realizada com o endereço L e o número de sub-rotina ou através da indicação do nome do programa.

Um programa principal também pode ser chamado como se fosse uma sub-rotina. Neste caso o fim de programa `M2` ou `M30` definido no programa principal é avaliado como `M17` (fim de programa com retorno para o programa de chamada).

Indicação

Da mesma forma uma sub-rotina pode ser iniciada como se fosse um programa principal.

Estratégia de localização do comando numérico:

Existe *_MPF?

Existe *_SPF?

Disto resulta: Se o nome da sub-rotina a ser chamada for idêntico ao nome do programa principal, então o programa principal de chamada é chamado novamente. Este feito, que normalmente não é desejado, deve ser evitado através de escolha clara de nomes através de sub-rotinas e programas principais.

Indicação

As sub-rotinas que não requerem nenhuma transferência de parâmetros, também podem ser chamadas a partir de um arquivo de inicialização.

Sintaxe

```
L<número>/<nome de programa>
```

Indicação

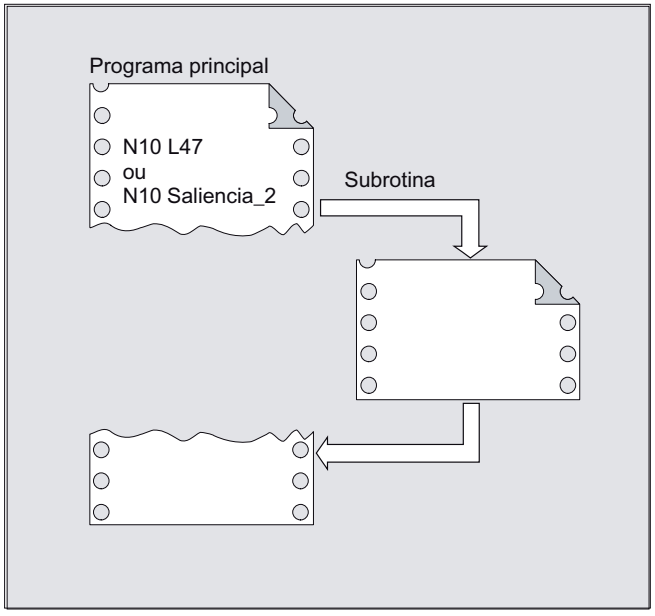
A chamada de uma sub-rotina sempre deve ser programada em um bloco NC próprio.

Significado

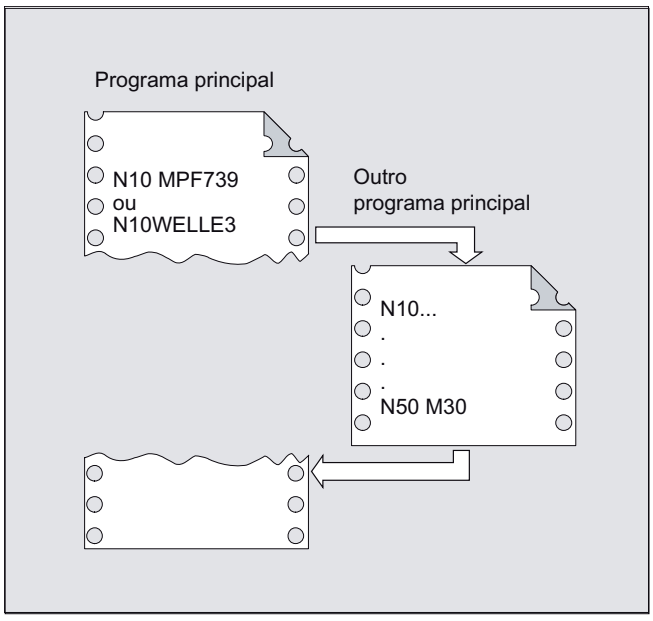
L:	Endereço para a chamada da sub-rotina	
<número>:	Número da sub-rotina	
	Tipo:	INT
	Valor:	Máximo de 7 casas decimais **Atenção:** Os zeros à esquerda têm significado na definição de nomes (⇒ L123, L0123 e L00123 são três sub-rotinas diferentes).
<nome de programa>:	Nome da sub-rotina (ou programa principal)	

Exemplos

Exemplo 1: Chamada de uma sub-rotina sem transferência de parâmetros



Exemplo 2: Chamada de um programa principal como sub-rotina



Veja também

Subrotina sem transferência de parâmetros (Página 169)

2.24.3.2 Chamada de subrotina com transferência de parâmetros (EXTERN)

Na chamada da sub-rotina com transferência de parâmetros podem ser transferidas variáveis ou valores de forma direta (não para parâmetros `VAR`).

Sub-rotinas com transferência de parâmetros devem ser identificadas no programa principal com `EXTERN` antes da chamada (p. ex. no início do programa). Aqui se especificam o nome da sub-rotina e os tipos das variáveis na sequência da transferência.

ATENÇÃO
Risco de troca
Tanto os tipos de variáveis como a ordem da transferência devem coincidir com as definições declaradas sob `PROC` na sub-rotina. Os nomes de parâmetros podem ser diferentes no programa principal e na sub-rotina.

Sintaxe

```
EXTERN <no do programa>(<Tipo_Par1>,<Tipo_Par2>,<Tipo_Par3>)
...
<nome do programa>(<Valor_Par1>,<Valor_Par2>,<Valor_Par3>)
```

Indicação

A chamada da sub-rotina sempre deve ser programada em um bloco NC próprio.

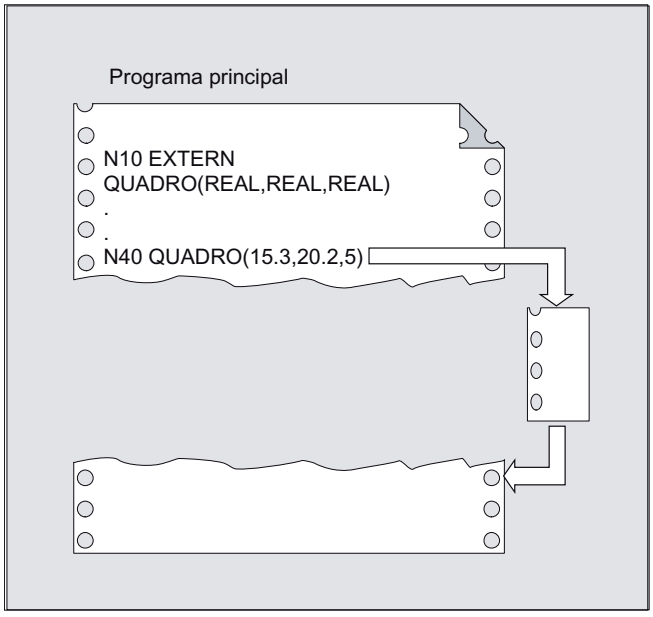
Significado

<nome de programa>:	Nome da sub-rotina
EXTERN:	Palavra-chave para a identificação de uma sub-rotina com transferência de parâmetros **Nota:** EXTERN somente deve ser especificado se a sub-rotina estiver no diretório de peças ou no diretório de sub-rotinas global. Os ciclos não precisam ser declarados como EXTERN.
<Tipo_Par1>,<Tipo_Par2>,<Tipo_Par3>:	Tipos de variáveis dos parâmetros de transferência na sequência da transferência
<Valor_Par1>,<Valor_Par2>,<Valor_Par3>:	Valores de variáveis dos parâmetros de transferência

Exemplos

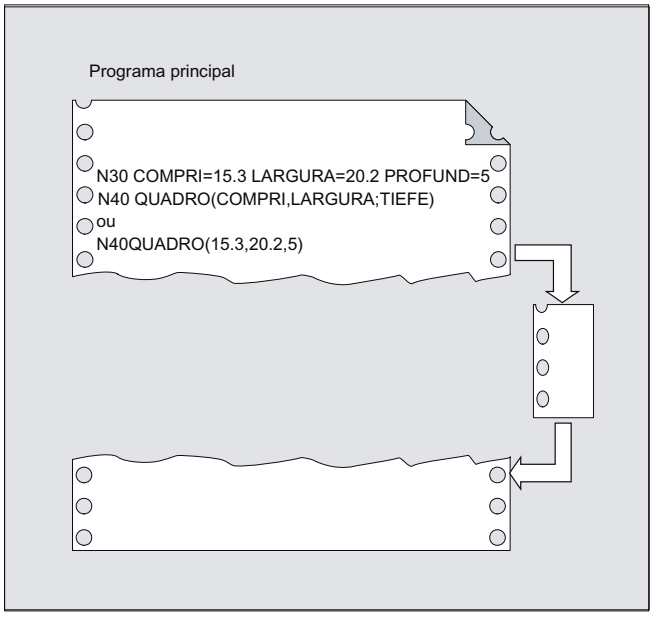
Exemplo 1: Chamada de sub-rotina com identificação prévia

Código de programa	**Comentário**
N10 EXTERN QUADRO(REAL,REAL,REAL)	; Indicação da sub-rotina.
...	
N40 QUADRO(15.3,20.2,5)	; Chamada da sub-rotina com transferência de parâmetros.



Exemplo 2: Chamada de sub-rotina sem identificação

Código de programa	**Comentário**
`N10 DEF REAL COMPRIMENTO, LARGURA, PROFUNDIDADE`	
`N20 …`	
`N30 COMPRIMENTO=15.3 LARGURA=20.2 PROFUNDIDADE=5`	
`N40 QUADRO(COMPRIMENTO,LARGURA,PROFUNDIDADE)`	`; ou: N40 QUADRO(15.3,20.2,5)`



Veja também

Subrotina com transferência de parâmetros Call-by-Value (PROC) (Página 170)

Subrotina com transferência de parâmetros Call-by-Reference (PROC, VAR) (Página 172)

2.24.3.3 Número de repetições de programa (P)

Se uma sub-rotina for executada várias vezes em sucessão, então se pode programar o número desejado de repetições do programa no endereço `P` do bloco com a chamada da sub-rotina.

⚠ CUIDADO

Chamada de sub-rotina com repetição de programa e transferência de parâmetros

Os parâmetros somente são transferidos na chamada do programa ou primeiro processamento. Para as demais repetições os parâmetros permanecem inalterados. Se houver a necessidade de alterar os parâmetros durante as repetições de programa, então se deve definir os ajustes correspondentes na sub-rotina.

Sintaxe

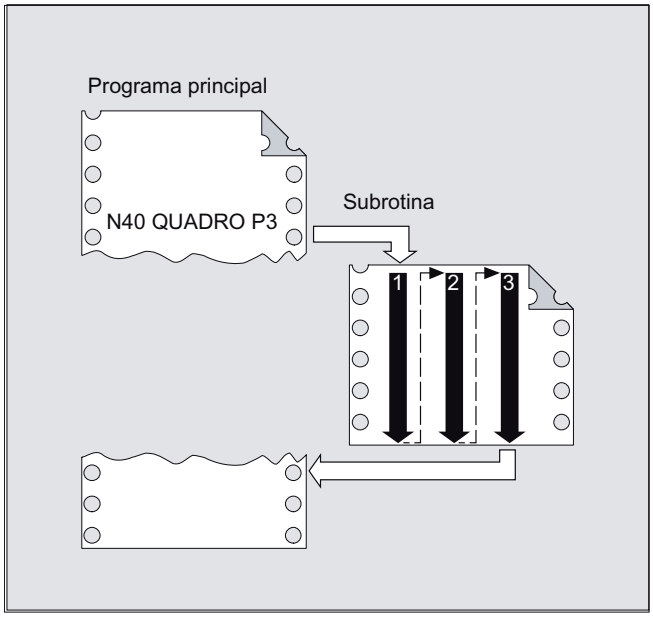
```
<nome de programa> P<valor>
```

Significado

`<nome de programa>`:	Chamada de sub-rotina	
`P`:	Endereço para programação de repetições de programa	
`<valor>`:	Número de repetições de programa	
	Tipo:	INT
	Faixa de valores:	1 … 9999 (sem sinal)

Exemplo

Código de programa	**Comentário**
...	
N40 QUADRO P3	; A sub-rotina QUADRO deve ser executada três vezes consecutivas.
...	



2.24.3.4 Chamada modal de subrotina (MCALL)

Através da chamada de sub-rotina modal `MCALL(<nome do programa>)` a sub-rotina indicada não é acessada imediatamente. Ao invés disto, o acesso ocorre a partir deste momento automaticamente no programa de peças após cada bloco de deslocamento com movimentos de percurso. Também passando pelos níveis de programa.

Indicação

Em uma execução de programa atua sempre apenas a última chamada de sub-rotina modal `MCALL(<nome do programa>)`. A chamada de sub-rotina modal atual substitui a ativa até agora.

Se forem transmitidos parâmetros para a sub-rotina, a transmissão de parâmetros ocorre somente durante a chamada `MCALL(<nome do programa>(Par1, Par2, ...))`.

ATENÇÃO

Chamadas de sub-rotinas modais sem movimento de percurso

Mesmo sem a programação de um movimento, a sub-rotina modal é chamada nas seguintes situações:

- Na programação de endereços `S` ou `F`, quando `G0` ou `G1` estiver ativo
- Se `G0` ou `G1` estiver sozinho no bloco ou for programado junto com outros comandos G.

Sintaxe

```
MCALL <nome de programa>
...
MCALL
```

Significado

`MCALL <nome do programa>`:	Ativar a função "Camada de sub-rotina modal"
`<nome de programa>`:	Nome da sub-rotina
`MCALL`:	Com `MCALL` sem a indicação de um nome de programa, a função "Chamada de sub-rotina modal" é desligada.

Condições gerais

ASUP

Se o processamento de um programa de peças for interrompido através de um ASUP (consultar o capítulo "Rotina de interrupção (ASUP) (Página 132)"), neste ASUP não serão executadas quaisquer chamadas de sub-rotina modais .

Se um ASUP for iniciado em estado de canal "Reset", ele se comportará em relação às chamadas de sub-rotinas modais como um programa de peças normal.

Ciclo de troca de ferramentas

Se durante o ciclo de troca de ferramenta ocorrer um cancelamento da seleção da função "Chamada de sub-rotina modal", então deve-se observar se o ciclo de troca de ferramentas, possivelmente também após a busca por blocos, é acessado de modo implícito através do processo de busca ASUP ou manualmente através de armazenamento. Nesta situação a seleção da função "Acesso de sub-rotina modal" não deve ser cancelada, caso contrário, o resultado do processo de pesquisa será adulterado. Por isto, recomenda-se, programar o cancelamento da seleção da função "Acesso de sub-rotina modal" no ciclo de troca de ferramenta, como segue:

```
Código de programa          Comentário
...
  IF $AC_ASUP == 0          ; A chamada não é realizada através do processo de pesquisa
                            ASUP ou sobre-armazenamento.
    MCALL                   ; Desativar a função "Acesso de sub-rotina modal".
  ENDIF
...
```

Exemplos

Exemplo 1

```
Código de programa          Comentário
N10 G0 X0 Y0
N20 MCALL L70               ; Ligar a chamada de sub-rotina modal para L70.
N30 X10 Y10                 ; Será deslocado para X10 Y10, em seguida, L70 será chamado.
N40 X20 Y20                 ; Será deslocado para X20 Y20, em seguida, L70 será chamado.
...
N100 MCALL                  ; Desativar a função "Acesso de sub-rotina modal".
```

Código de programa	Comentário
N110 X0 Y0	; Será deslocado para X0 Y0, L70 **não** será chamado.

Exemplo 2

Código de programa
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

Neste exemplo os blocos NC subsequentes com eixos de percurso programados estão na sub-rotina L80. L70 é chamada pela L80.

2.24.3.5 Chamada de subrotinas indireta (CALL)

Em função das condições dadas se pode chamar diferentes sub-rotinas em uma posição. Para isso se armazena o nome da sub-rotina em uma variável de tipo STRING. A chamada da sub-rotina é realizada com `CALL` e o nome da variável.

Indicação

A chamada indireta de sub-rotinas somente é possível para sub-rotinas sem transferência de parâmetros. Para a chamada direta de uma sub-rotina armazene o nome em uma constante de STRING.

Sintaxe

```
CALL <nome do programa>
```

Significado

`CALL`:	Comando para a chamada de sub-rotina indireta	
`<nome de programa>`:	Nome da sub-rotina (variável ou constante)	
	Tipo:	STRING

Exemplo

Chamada direta com constante de STRING:

Código de programa	Comentário
…	
CALL "/_N_WCS_DIR/_N_SUBPROG_WPD/_N_PECA1_SPF"	; Chamada direta da sub-rotina PECA1 com o CALL.
…	

Chamada indireta através de variável:

Código de programa	Comentário
…	
DEF STRING[100] PROGNAME	; Definição de variável.
PROGNAME="/_N_WCS_DIR/_N_SUBPROG_WPD/_N_PECA1_SPF"	; Associação da sub-rotina PECA1 à variável PROGNAME.
CALL PROGNAME	; Chamada indireta da sub-rotina PECA1 através do CALL e da variável PROGNAME.
…	

2.24.3.6 Chamada de subrotina indireta com indicação da parte de programa que deve ser executada (CALL BLOCK ... TO ...)

Com `CALL` e a combinação de palavras-chave `BLOCK ... TO` se chama uma sub-rotina de forma indireta e se executa a parte do programa demarcada entre o marcador inicial e o marcador final.

Sintaxe

```
CALL <nome do programa> BLOCK <marcador inicial> TO <marcador final>
CALL BLOCK <marcador inicial> TO <marcador final>
```

Significado

`CALL`:	Comando para a chamada de sub-rotina indireta	
`<nome de programa>`:	Nome da sub-rotina (variável ou constante) que contém a parte do programa a ser executada (**Indicação opcional**).	
	Tipo:	STRING
	Nota: Se não for programado nenhum `<nome de programa>`, será executada a parte do programa demarcada entre o `<marcador inicial>` e o `<marcador final>` no atual programa.	
`BLOCK ... TO ...` :	Combinação de palavras-chave para execução indireta de partes de programa	
`<marcador inicial>`:	Variável que faz referência ao início da parte do programa a ser executada.	
	Tipo:	STRING
`<marcador final>`:	Variável que faz referência ao fim da parte do programa a ser executada.	
	Tipo:	STRING

Exemplo

Programa principal:

Código de programa	Comentário
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; Definição de variáveis para o marcador inicial e o final.
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTORNO_1" BLOCK STARTLABEL TO ENDLABEL	; Chamada de sub-rotina indireta e identificação da parte do programa que deve ser executada.
...	

Sub-rotina:

Código de programa	Comentário
PROC CONTORNO_1 ...	
LABEL_1	; Marcador inicial: Início da execução da parte do programa
N1000 G1 ...	
...	
LABEL_2	; Marcador final: Fim da execução da parte do programa
...	

2.24.3.7 Chamada indireta de um programa programado em linguagem ISO (ISOCALL)

Com a chamada indireta de programas `ISOCALL` se pode chamar um programa criado em linguagem ISO. Neste caso se ativa o modo ISO ajustado nos dados da máquina. No fim do programa se ativa novamente o modo de execução original. Se nos dados da máquina não foi ajustado nenhum modo ISO, a chamada da sub-rotina será realizada em modo Siemens.

Para mais informações sobre o modo ISO, veja no(a):
Literatura:
Descrição do funcionamento de dialetos ISO

Sintaxe

```
ISOCALL <nome do programa>
```

Significado

`ISOCALL`:	Palavra-chave para chamada indireta de sub-rotina que é ativada com o modo ISO ajustado nos dados da máquina
`<nome de programa>`:	Nome de um programa programado em linguagem ISO (variável ou constante de tipo STRING)

Exemplo: Contorno com programação de ciclos chamado a partir do modo ISO

Código de programa	Comentário
0122_SPF	; Descrição do contorno em modo ISO
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME = "0122"	; Programa de peças Siemens (ciclo)
...	
N2000 R11 = $AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	; Executar o programa 0122.spf em modo ISO
...	
N2400 M30	

2.24.3.8 Chamada de subrotina com indicação de caminho e parâmetros (PCALL)

Com PCALL se pode chamar sub-rotinas com indicação absoluta do caminho e transferência de parâmetros.

Sintaxe

```
PCALL <caminho/nome do programa>(<parâmetro 1>,…,<parâmetro n>)
```

Significado

PCALL:	Palavra-chave para chamada de sub-rotinas com indicação absoluta do caminho.
<caminho/nome de programa>:	Indicação de atalho absoluta incluindo o nome da sub-rotina. Regras para a indicação de atalho, consultar "Endereçamento dos arquivos da memória do programa (Página 227)". Se não foi especificado nenhum caminho absoluto, o PCALL age como uma chamada padrão de sub-rotina com identificadores de programa. O nome do programa é especificado sem o prefixo e sem a identificação do arquivo. Se for necessário programar o nome do programa com prefixo e a identificação do arquivo, então ele deve ser explicitamente declarado com o comando EXTERN com o prefixo e a identificação do arquivo.
<Parâmetro 1>, ...:	Parâmetro atual de acordo com a instrução PROC da sub-rotina.

Exemplo

Código de programa
PCALL/_N_WCS_DIR/_N_EIXO_WPD/EIXO(parâmetro1,parâmetro2,…)

2.24.3.9 Ampliar o caminho de localização na chamada de subrotinas (CALLPATH)

Com o comando CALLPATH se pode ampliar o caminho de localização da chamada de sub--rotinas. Dessa forma também podem ser chamadas sub-rotinas de um diretório de peças de trabalho não selecionado, sem precisar especificar o caminho completo da sub-rotina.

Uma outra possibilidade de aplicação é oferecida no modo operacional EES "EES sem GDIR", se for utilizado um diretório em uma memória de programa externa para armazenar as sub--rotinas globais. Neste caso, o atalho de busca pode ser ampliado com CALLPATH por este diretório de sub-rotina.

A ampliação do caminho de localização é realizada antes da entrada para ciclos de usuário (_N_CUS_DIR).

Através dos seguintes eventos se desfaz a seleção do caminho de localização:

- CALLPATH com espaços vazios
- CALLPATH sem parâmetros
- Fim do programa de peça
- Reset

Sintaxe

```
CALLPATH("<nome do caminho>")
```

Significado

CALLPATH:	Palavra-chave para a ampliação programável do caminho de localização. Se programa em uma linha própria do programa de peça.
<nome de caminho>:	Constante ou variável de tipo STRING. Contém a indicação absoluta do caminho de um diretório que deve ampliar o caminho de localização. Regras para a indicação de atalho, consultar "Endereçamento dos arquivos da memória do programa (Página 227)".

Exemplo

O atalho de busca deve ser ampliado em um determinado diretório de peça de trabalho:

Código de programa
...

Código de programa
`CALLPATH("/_N_WCS_DIR/_N_MYWPD_WPD")`
`...`

Com isso se configura o seguinte caminho de localização (a posição 5. é nova):

1. Diretório atual/*nome*
2. Diretório atual/*nome_SPF*
3. Diretório atual/*nome_MPF*
4. //NC:/_N_SPF_DIR/*nome_SPF*
5. **/_N_WKS_DIR/_N_MYWPD_WPD/nome_SPF**
6. /_N_CST_DIR / *nome_SPF*
7. /_N_CMA_DIR / *nome_SPF*
8. /_N_CST_DIR / *nome_SPF*

Condições gerais

- O `CALLPATH` verifica se os caminhos programados realmente existem. Em caso de erro, a execução do programa de peças será cancelada com o alarme de bloco de correção 14009.
- O `CALLPATH` também pode ser programado em arquivos INI. Ele atua durante o período de execução do arquivo INI (arquivo INI WPD ou programa de inicialização para dados de NC ativos, p. ex. Frames no 1º canal _N_CH1_UFR_INI). Em seguida o caminho de localização é resetado novamente.

2.24.3.10 Executar subrotina externa (840D sl) (EXTCALL)

Com o comando `EXTCALL` é possível recarregar e processar um programa de usinagem de um diretório externo.

Diretórios externos disponíveis:

- Unidade de leitura local
- Unidade de rede
- Unidade USB

Indicação

Só é possível uma interface processar um programa externo encontrado num unidade USB, se a interface da USB usar o painel de controle frontal ou TCU.

ATENÇÃO

Dano na ferramenta/peça de trabalho devido ao FlashDrive USB

É aconselhado não usar um USB-FlashDrive no processamento de um subprograma externo. Uma interrupção de comunicação com o USB-FlashDrive durante a execução do programa de usinagem por causa de falhas de contato, ejeção acidental, interrupção por choques ou retirada proposital, resulta na parada imediata do processo. A ferramenta e/ou a peça de trabalho podem ser danificadas com isso.

pré-configuração do caminho externo do programa

O caminho para o diretório externo do programa pode ser pré-configurado com o dado de configuração:

SD42700 $SC_EXT_PROG_PATH

Junto com o caminho ou com o identificador de programa indicado na chamada `EXTCALL` é formado o caminho geral do programa de usinagem chamado.

Indicação

Parâmetro

Na chamada de um programa externo estes não poderão transferir nenhum parâmetro.

Sintaxe

```
EXTCALL("<caminho/><nome do programa>")
```

Significado

EXTCALL:	Comando para chamada de uma sub-rotina externa	
"<caminho/><nome de programa>":	Constante/variável de tipo STRING	
	<caminho/>:	Indicação de caminho absoluta ou relativa (**opcional**)
	<nome de programa>:	O nome do programa é especificado sem o prefixo "_N_". A extensão de arquivo ("MPF", "SPF") pode ser adicionada em nomes de programa com o caractere "_" ou "." (**opcional**). Exemplo: "WELLE" "WELLE_SPF" "WELLE.SPF"

Indicação de caminho: Designações abreviadas

Na indicação do caminho podem ser utilizadas as seguintes abreviações:

- Unidade de disco local: "**LOCAL_DRIVE:**"
- Cartão CF: "**CF_CARD:**"
- Unidade USB (painel de controle frontal): "**USB:**"

As designações abreviadas "CF_CARD:" e "LOCAL_DRIVE:" são de utilização alternativa

Exemplo

Execução a partir de unidade local

O programa principal "MAIN.MPF" encontra-se na memória NC e é selecionado para execução.

Subprograma "SP_1"

O subprograma externo "SP_1.SPF" ou "SP_1.MPF" encontra-se na unidade de disco local no diretório "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

O caminho para o diretório externo do programa deve ser configurado com:

SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

Indicação

Indicação do caminho para a chamada do subprograma externo:

- Sem utilização da pré-configuração: "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- Com utilização da pré-configuração: "SP_1"

Subprograma "SP_2"

O subprograma externo "SP_2.SPF" ou "SP_2.MPF" encontra-se no diretório WKS.DIR/WST1.WPD da unidade USB. A pré-configuração do caminho para o diretório externo do

programa é utilizada para o caminho do subprograma "SP_1" e não será transcrito no programa principal. Por isso é preciso informar o completo caminho para a chamada do subprograma "SP_2".

Programa principal "MAIN"

```
Código de programa
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

Outras informações

Chamada EXTCALL com indicação absoluta do caminho

Se o subprograma encontra-se no caminho especificado, então este será executado com a chamada `EXTCALL`. Se o subprograma não estiver no caminho especificado, então a execução do programa será encerrada com o chamado `EXTCALL`.

Chamada EXTCALL com indicação relativa do caminho / sem indicação de caminho

Em caso de uma chamada `EXTCALL` com indicação relativa do caminho ou sem a indicação do caminho é realizada uma localização nas memórias disponíveis de programas de acordo com o seguinte modelo:

1. Se no SD42700 $SC_EXT_PROG_PATH existir um caminho pré-configurado, então primeiramente será procurado uma indicação no chamado `EXTCALL`partida deste caminho (nome do programa evtl. indicação de caminho relativo). O caminho absoluto resulta da sequência de caracteres dada pelo(a):
   - indicação de caminho pré-configurada no SD42700 $SC_EXT_PROG_PATH
   - Separador"/"
   - Caminho e nome do subprograma no comando `EXTCALL`
2. Se o subprograma não foi encontrado no 1., então a procura será feita no diretório do usuário.

A localização é encerrada assim que a subrotina for encontrada pela primeira vez. Se o subprograma não for encontrado, então a execução do programa será encerrada com o chamado `EXTCALL`.

Memória temporária configurável (memória FIFO)

Para o processamento de um subprograma externo será necessário uma memória de recarregamento. O tamanho da memória de recarregamento é pré-configurada com 30 kByte

e apenas pode ser alterada pelo fabricante da máquina. (sobre MD18360 MM_EXT_PROG_BUFFER_SIZE).

Indicação

Subprogramas com instruções de salto

Em subprogramas externos, que contenham instruções de salto (`GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, `REPEAT`, `IF`, `ELSE`, `ENDIF` etc.), os saltos têm de estar dentro da memória de carregamento.

Indicação

Programas ShopMill/ShopTurn

Programas ShopMill e ShopTurn devem estar completamente dentro da memória recarregável devido às descrições de contornos adicionadas no fim do arquivo.

Para cada subprograma externo processado paralelamente será necessário uma memória de recarregamento.

Reset / final do programa / POWER ON

Com RESET e POWER ON as chamadas externas de subrotinas são canceladas e a memória temporária é apagada.

Uma subrotina selecionada para modo "Executar externamente" permanece selecionada além do Reset, fim do programa ou POWER ON para o modo "Executar externamente". O comportamento não se diferencia dos programas selecionados internamente, desde que a memória externa do programa continue disponível.

Literatura

Para mais informações sobre "Execução externa", veja:

Manual de funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1)

2.24.3.11 Executar subrotina externa (828D) (EXTCALL)

Com o comando `EXTCALL` é possível recarregar e processar um programa de usinagem de um diretório externo.

Diretórios externos disponíveis:

- Cartão CF de usuário
- Unidade de rede
- Unidade USB

Indicação

Só é possível uma interface processar um programa externo encontrado numa unidade USB, se a interface da USB usar o painel de controle frontal ou TCU.

ATENÇÃO

Dano na ferramenta/peça de trabalho devido ao FlashDrive USB

É aconselhado não usar um USB-FlashDrive no processamento de um subprograma externo. Uma interrupção de comunicação com o USB-FlashDrive durante a execução do programa de usinagem por causa de falhas de contato, ejeção acidental, interrupção por choques ou retirada proposital, resulta na parada imediata do processo. A ferramenta e/ou a peça de trabalho podem ser danificadas com isso.

pré-configuração do caminho externo do programa

O caminho para o diretório externo do programa pode ser pré-configurado com o dado de configuração:

SD42700 $SC_EXT_PROG_PATH

Junto com o caminho ou com o identificador de programa indicado na chamada `EXTCALL` é formado o caminho geral do programa de usinagem chamado.

Indicação

Parâmetro

Na chamada de um programa externo estes não poderão transferir nenhum parâmetro.

Sintaxe

```
EXTCALL("<caminho/><nome do programa>")
```

Significado

EXTCALL:	Comando para chamada de uma sub-rotina externa	
"<caminho/><nome de programa>":	Constante/variável de tipo STRING	
	<caminho/>:	Indicação de caminho absoluta ou relativa (**opcional**)
	<nome de programa>:	O nome do programa é especificado sem o prefixo "_N_". A extensão de arquivo ("MPF", "SPF") pode ser adicionada em nomes de programa com o caractere "_" ou "." (**opcional**). Exemplo: "WELLE" "WELLE_SPF" "WELLE.SPF"

Indicação de caminho: Designações abreviadas

Na indicação do caminho podem ser utilizadas as seguintes abreviações:

- Cartão CF de usuário: "**CF_CARD:**"
- Unidade USB (painel de controle frontal): "**USB:**"

Exemplo

O programa principal "MAIN.MPF" encontra-se na memória NC e é selecionado para execução.

Subprograma "SP_1"

O subprograma externo "SP_1.SPF" ou "SP_1.MPF" encontra-se no carão CF de usuário no diretório "/WKS.DIR/WST1.WPD".

O caminho para o diretório externo do programa deve ser configurado com:

SD42700 $SC_EXT_PROG_PATH = CF_CARD:WKS.DIR/WST1.WPD

Indicação

Indicação do caminho para a chamada do subprograma externo:

- Sem utilização da pré-configuração: "CF_CARD:WKS.DIR/WST1.WPD/SP_1"
- Com utilização da pré-configuração: "SP_1"

Subprograma "SP_2"

O subprograma externo "SP_2.SPF" ou "SP_2.MPF" encontra-se no diretório WKS.DIR/WST1.WPD da unidade USB. A pré-configuração do caminho para o diretório externo do programa é utilizada para o caminho do subprograma "SP_1" e não será transcrito no programa principal. Por isso é preciso informar o completo caminho para a chamada do subprograma "SP_2".

Programa principal "MAIN"

```
Código de programa
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

Outras informações

Chamada EXTCALL com indicação absoluta do caminho

Se o subprograma encontra-se no caminho especificado, então este será executado com a chamada EXTCALL. Se o subprograma não estiver no caminho especificado, então a execução do programa será encerrada com o chamado EXTCALL.

Chamada EXTCALL com indicação relativa do caminho / sem indicação de caminho

Em caso de uma chamada EXTCALL com indicação relativa do caminho ou sem a indicação do caminho é realizada uma localização nas memórias disponíveis de programas de acordo com o seguinte modelo:

1. Se no SD42700 $SC_EXT_PROG_PATH existir um caminho pré-configurado, então primeiramente será procurado uma indicação no chamado EXTCALLpartida deste caminho (nome do programa evtl. indicação de caminho relativo). O caminho absoluto resulta da sequência de caracteres dada pelo(a):
   - indicação de caminho pré-configurada no SD42700 $SC_EXT_PROG_PATH
   - Separador"/"
   - Caminho e nome do subprograma no comando EXTCALL
2. Se o subprograma não foi encontrado no 1., então a procura será feita no diretório do usuário.

A localização é encerrada assim que a subrotina for encontrada pela primeira vez. Se o subprograma não for encontrado, então a execução do programa será encerrada com o chamado EXTCALL.

Memória temporária configurável (memória FIFO)

Para o processamento de um subprograma externo será necessário uma memória de recarregamento. O tamanho da memória recarregável está pré-definida (ver MD18360 MM_EXT_PROG_BUFFER_SIZE).

Indicação

Subprogramas com instruções de salto

Em subprogramas externos, que contenham instruções de salto (GOTOF, GOTOB, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDIF etc.), os saltos têm de estar dentro da memória de carregamento.

Indicação

Programas ShopMill/ShopTurn

Programas ShopMill e ShopTurn devem estar completamente dentro da memória recarregável devido às descrições de contornos adicionadas no fim do arquivo.

Para cada subprograma externo processado paralelamente será necessário uma memória de recarregamento.

Reset / final do programa / POWER ON

Com RESET e POWER ON as chamadas externas de subrotinas são canceladas e a memória temporária é apagada.

Uma subrotina selecionada para modo "Executar externamente" permanece selecionada além do Reset, fim do programa ou POWER ON para o modo "Executar externamente". O comportamento não se diferencia dos programas selecionados internamente, desde que a memória externa do programa continue disponível.

Literatura

Para mais informações sobre "Execução externa", veja:

Manual de funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1)

2.25 Técnica de macros (DEFINE ... AS)

ATENÇÃO
A técnica de macros aumenta a complexidade da programação
Com a técnica de macros se pode alterar consideravelmente a linguagem de programação do comando. A técnica de macros dó pode ser implementada com grande exatidão.

Como macro definimos o agrupamento de instruções individuais como uma instrução geral de nome próprio. Na chamada da macro na execução do programa os comandos programados no nome da macro são executados sequencialmente.

De acordo com a área de validade, ou seja, a área em que a definição de macros tem efeito, existem as seguintes categorias de macros:

- Macros locais
  Macros locais são macros definidas no início de programa de um programa NC, o qual não é o programa principal no momento do processamento. São criadas no momento da chamada do programa NC e apagadas com o reset do fim do programa ou com a próxima inicialização do sistema de comando. Macros locais somente podem ser acessadas no contexto do programa NC em que foram definidas.
- Macros globais de programa
  Macros globais de programa são macros definidas no início de programa de um programa NC que é usado como programa principal. São criadas no momento da chamada do programa NC e apagadas com o reset do fim do programa ou com a próxima inicialização do sistema de comando. Macros globais de programa podem ser acessadas no programa principal e em todos os subprogramas.

  Indicação

  Disponibilidade de macros globais de programa

  Macros globais de programa definidas no programa principal somente estão disponíveis também nos subprogramas quando o seguinte dado de máquina estiver ajustado:

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  Com MD11120 = 0, macros globais de programa, definidas no programa principal, estão disponíveis somente no programa principal.

- Macros globais
  Macros globais são macros globais de NC ou de canal definidas em um arquivo de definições (arquivo macro) que são conservadas também após o reset do final do programa ou após a próxima inicialização do sistema de comando. Chamada e processamento de macros globais são possíveis em qualquer programa principal ou subprograma.

  Indicação

  Para poder utilizar macros de um arquivo de macros **externo** no programa NC. o arquivo de macros precisa ser carregado em NC.

É necessário definir macros antes da sua utilização. Aqui devem ser observadas as seguintes regras:

- Nas macros podem-se definir quaisquer identificadores, funções G, M e H e nomes de subprogramas L.
- A definição de macros pode ocorrer no início do programa ou em um arquivo de definições próprio (arquivo macro).
- A definição de macros locais e macros globais de programa ocorre no início do programa.
- Macros globais devem ser definidas em um arquivo macro, por ex. _N_DEF_DIR/ _N_MGUD_DEF.
- Macros de comando G somente podem ser definidas como macros globais.
- As funções H e L são programáveis com 2 dígitos.
- Os comandos M e G podem ser programadas com 3 dígitos.

Indicação

Palavras-chave e nomes reservados não devem ser sobrescritos com macros.

Sintaxe

Definição de macro:
```
DEFINE <Macro_Name> AS <Operation_1> <Operation_2> ...
```

Chamada no programa NC:
```
<Macro_Name>
```

Significado

`DEFINE ... AS`:	Combinação de palavras-chave para definição de uma macro
`<Macro_Name>`:	Nome da macro Como nomes de macros somente são permitidos identificadores. Com o nome da macro se chama a macro a partir do programa NC.
`<Operação_1>`:	Primeira instrução de programação na macro
`<Operação_2>`:	Segunda instrução de programação na macro

Condições gerais

Aninhamento

Não é possível fazer um aninhamento de macros.

Exemplos

Exemplo 1: Definição de macro no início do programa

Código de programa	Comentário
DEFINE LINHA AS G1 G94 F300	; Definição de macro
...	
N70 LINHA X10 Y20	; Chamada de macro
...	

Exemplo 2: Definições de macro em um arquivo de macro

Código de programa	Comentário
DEFINE M6 AS L6	; Para a troca de ferramentas se chama uma sub-rotina que assume a transferência de dados necessária. Na sub-rotina é executada a função M de troca de ferramentas propriamente dita (p. ex. M106).
DEFINE G81 AS DRILL(81)	; Réplica do comando DIN G.
DEFINE G33 AS M333 G333	; Para a usinagem de roscas se solicita a sincronização com o PLC. O comando G original G33 foi renomeada para G333 pelo MD, a programação permanece a mesma para o usuário.

Exemplo 3: Arquivo de macro externo

Após o carregamento do arquivo de macro externo no comando, o arquivo de macro deve ser carregado no NC. Somente então se pode utilizar as macros no programa NC.

Código de programa	Comentário
%_N_UMAC_DEF	
;$PATH=/_N_DEF_DIR	; Macros específicas de usuário
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; Fuso gira à direita, líquido refrigerante ligado
DEFINE M14 AS M4 M7	; Fuso gira à esquerda, líquido refrigerante ligado
DEFINE M15 AS M5 M9	; Fuso parado, líquido refrigerante desligado
DEFINE M6 AS L6	; Chamada do programa de troca de ferramentas
DEFINE G80 AS MCALL	; Desseleção do ciclo de furação
M30	

Gerenciamento de arquivos e programas 3

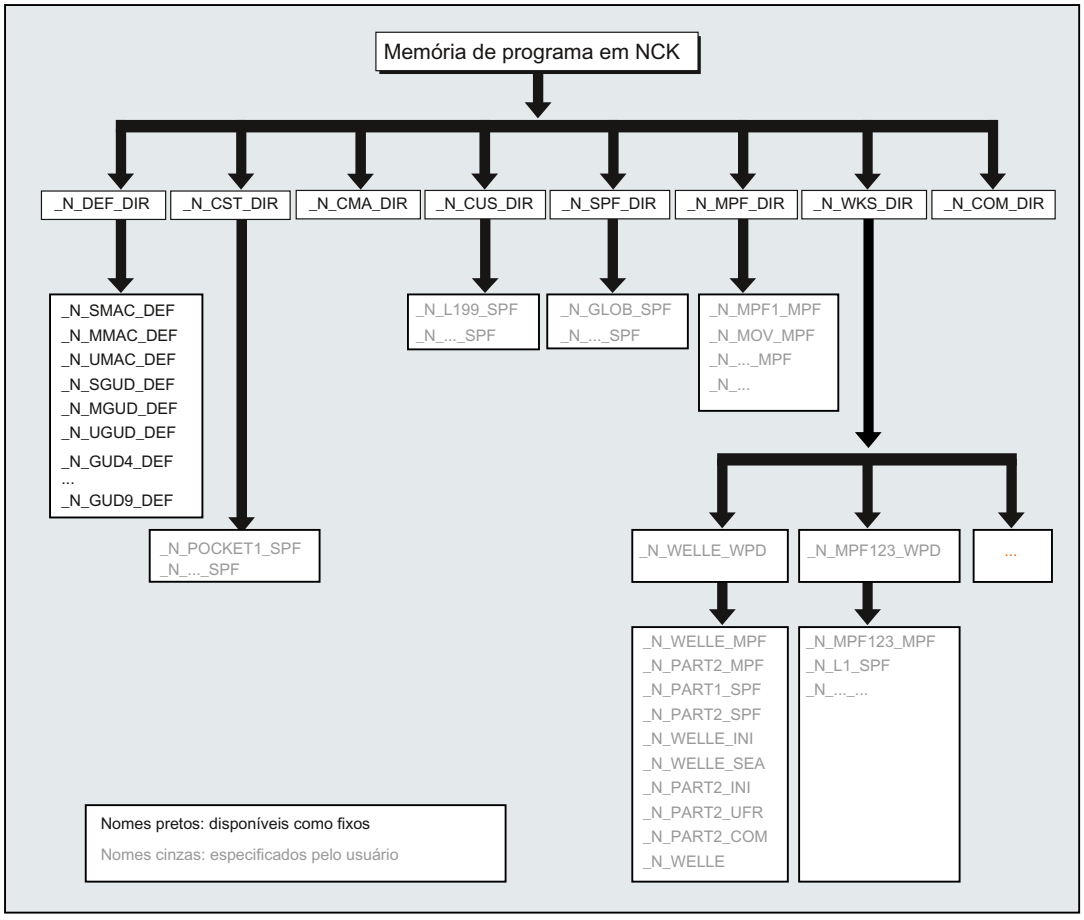
3.1 Memória de programa

3.1.1 Memória de programa em NC

Na memória de programas são armazenados arquivos e programas (p. ex. programas principais e sub-rotinas, definições de macro) de forma permanente (→ Sistema passivo de arquivos).

Literatura:
Manual de funções ampliadas; Configuração de memória (S7)

Também existe uma quantidade de tipos de arquivos que pode ser armazenada temporariamente e, em caso de necessidade (p. ex. para a usinagem de uma determinada peça), estes arquivos são transferidos para a memória de trabalho (p. ex. para inicializações).



Diretórios padrão

Como padrão estão disponíveis os seguintes diretórios:

Diretório	Conteúdo
_N_DEF_DIR	Módulos de dados e módulos de macros
_N_CST_DIR	Ciclos padronizados
_N_CMA_DIR	Ciclos de fabricante
_N_CUS_DIR	Ciclos de usuário
_N_WCS_DIR	Peças de trabalho
_N_SPF_DIR	Sub-rotinas globais
_N_MPF_DIR	Programas principais
_N_COM_DIR	Comentários

Tipos de arquivo

Na memória de programa podem ser inseridos os seguintes tipos de arquivos:

Tipo de arquivo	Descrição
<Nome>_MPF	Programa principal
<Nome>_SPF	Sub-rotina
<Nome>_TEA	Dados de máquina
<Nome>_SEA	Dados de ajuste
<Nome>_TOA	Correções de ferramentas
<Nome>_UFR	Deslocamentos de ponto zero/Frame
<Nome>_INI	Arquivo de inicialização
<Nome>_GUD	Dados de usuário globais
<Nome>_RPA	Parâmetros R
<Nome>_COM	Comentário
<Nome>_DEF	Definições de dados de usuário globais e macros

Diretório principal das peças de trabalho (_N_WCS_DIR)

O diretório principal das peças de trabalho está configurado com a denominação _N_WKS_DIR na memória de programa. O diretório principal das peças de trabalho contém todos os diretórios correspondentes de todas peças de trabalho que foram programadas.

Diretórios de peça de trabalho (..._WPD)

Um diretório de peça de trabalho contém todos arquivos necessários para a usinagem de uma peça de trabalho. Estes podem ser programas principais, sub-rotinas, quaisquer programas de inicialização e arquivos de comentários.

Após a seleção do programa, os programas de inicialização são executados uma vez no primeiro Start do programa de peça (de acordo com o dado de máquina MD11280 $MN_WPD_INI_MODE).

Exemplo:

O diretório de peça de trabalho _N_EIXO_WPD, que foi criado para a peça EIXO, contém os seguintes arquivos:

Arquivo	Descrição
_N_EIXO_MPF	Programa principal
_N_PART2_MPF	Programa principal
_N_PART1_SPF	Sub-rotina
_N_PART2_SPF	Sub-rotina
_N_EIXO_INI	Programa geral de inicialização dos dados para a peça de trabalho
_N_EIXO_SEA	Programa de inicialização dos dados de ajuste
_N_PART2_INI	Programa geral de inicialização dos dados para o programa Part 2
_N_PART2_UFR	Programa de inicialização dos dados de Frame para o programa Part 2
_N_WELLE_COM	Arquivo de comentários

Além disso, em um diretório de peças de trabalho, podem ser também armazenados dados que não são diretamente necessários para o trabalho com o NC. Estes podem ser junto com os arquivos ASCII também arquivos binários com p. exemplo figuras no formato JPG ou descrições no formato PDF. A fim de que estes possam ser interpretados como arquivos binários pelo NC, as extensões dos arquivos devem ser reconhecidas no NC (ajuste no ativação via MD17000 $MN_ EXTENSIONS_OF_BIN_FILES; no ajuste básico, as seguintes extensões de arquivo são pré-alocadas: JPG, GIF, PNG, BMP, PDF, ICO, HTM, TXT): JPG, GIF, PNG, BMP, PDF, ICO, HTM).

Selecionar peça de trabalho para usinagem

Um diretório de peça de trabalho pode ser selecionado para a execução em um canal. Se neste diretório encontra-se um programa principal de **mesmo nome** ou apenas um único programa principal (MPF), então este será automaticamente selecionado para execução.

Literatura:
Manual de operação

3.1.2 Memória externa de programas

Além do sistema de arquivos passivo em NC, também podem estar disponíveis as memórias de programa externas em uma máquina (p. ex., na unidade de disco local ou na unidade de rede).

Por meio das funções "Executar externamente" ou "EES (Execution from External Storage)" os programas de peças podem ser executados **diretamente** a partir das memórias externas de programas.

Literatura:
Manual de funcionamento das funções básicas; K1: BAG, Canal, Operação do programa, Comportamento Reset

Memória global do programa de peças (GDIR)

No caso de unificação das unidades de disco, uma das unidades de disco pode ser designada como memória do programa de peças global (GDIR).

Literatura:
manual de operação; capítulo: "Gerenciar os programas" > "Criar as unidades de disco"

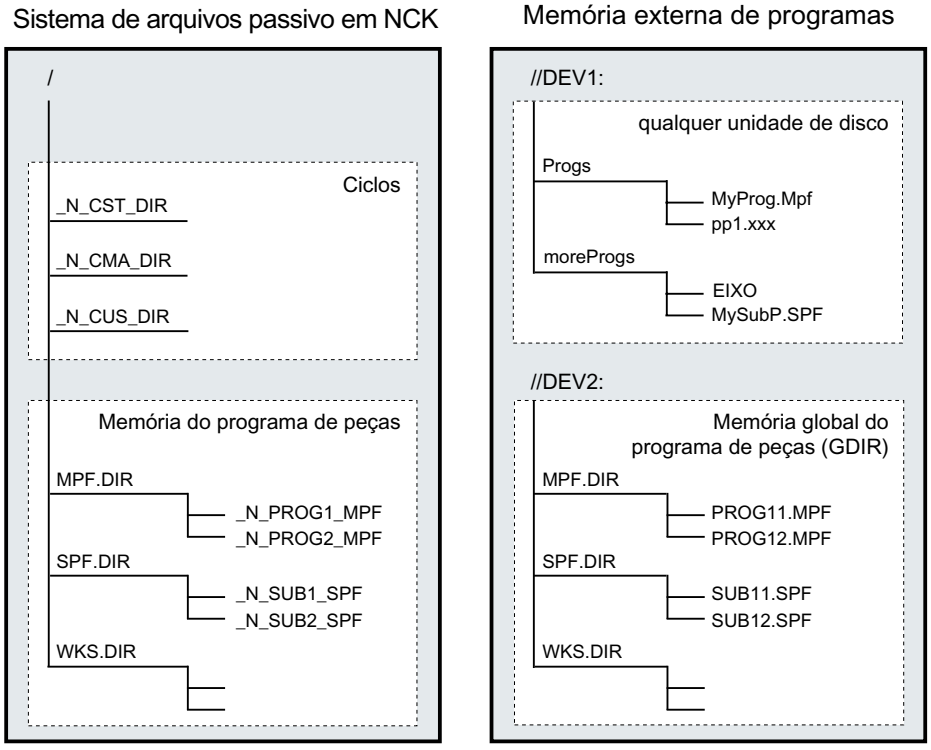
Pelo sistema então serão criados automaticamente os diretórios MPF.DIR, SPF.DIR e WKS.DIR na unidade de disco. Estes três diretórios formam o GDIR.

O GDIR representa um papel exclusivamente para a função EES. Dependendo da configuração da unidade de disco, o GDIR substitui ou amplia a memória do programa de peças NC. A criação de um GDIR, no entanto, não é obrigatoriamente necessária para a operação EES.

Os diretórios e arquivos do GDIR podem ser endereçados em um programa de peças do mesmo modo do que em um sistema de arquivos passivo. Assim, um alocamento compatível de um programa NC com indicações de atalho a partir de um sistema de arquivos passivo é possível para o GDIR. O diretório SPF.DIR do GDIR é suprimido no atalho de busca para as sub-rotinas.

Organização do programa

A ilustração seguinte deve ilustrar a organização de programas nas memórias externas de programa:



Sistema de arquivos Case-insensitive

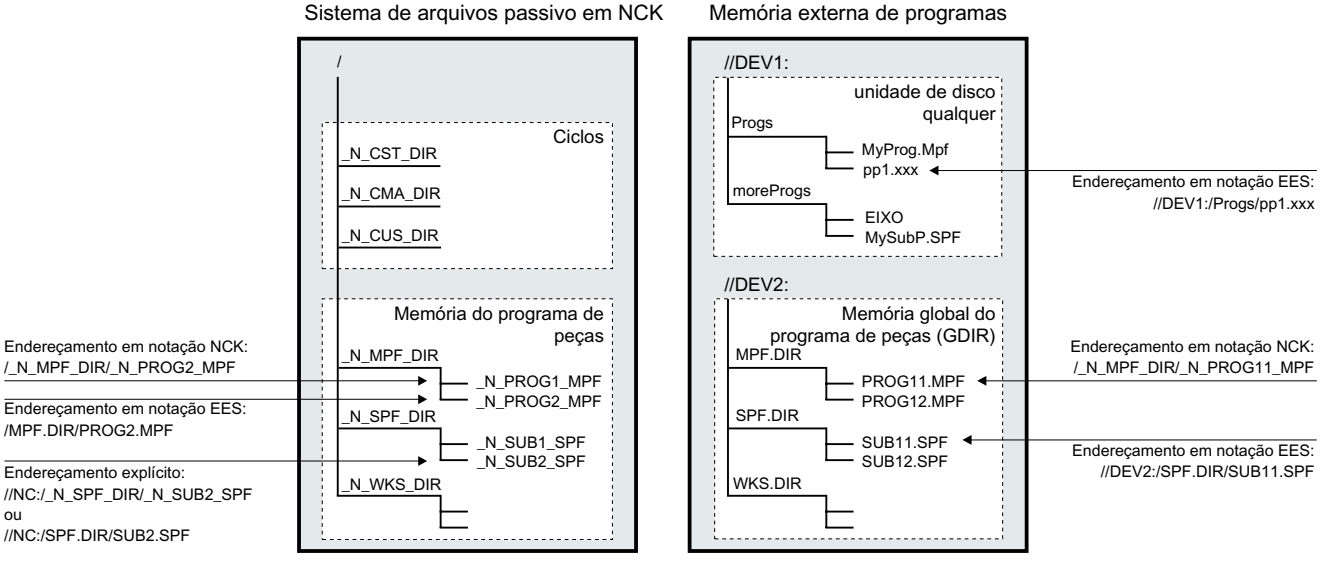
Indicação

Para evitar os problemas devido à escrita em maiúsculas e minúsculas durante o endereçamento do arquivo (consultar "Endereçamento dos arquivos da memória do programa (Página 227)"), os sistemas de arquivo **case-insensitive** devem ser utilizados como memória externa de programa.

3.1.3 Endereçamento dos arquivos da memória do programa

Um arquivo na memória de programas que é abordado através de um comando de manipulação de arquivo (por ex., WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO), será referenciado, ou através de um atalho absoluto mais o nome do arquivo, ou apenas através do nome do arquivo. No segundo caso é utilizado o atalho do programa selecionado como atalho do arquivo.

Endereçamento em notação NC/EES



Endereçamento dos arquivos do sistema de arquivos passivo

O endereçamento de arquivos do sistema de arquivos passivo é realizado normalmente em **notação NC** (nomes do diretório e do arquivo iniciando com a identificação de domínio "_N_", o delimitador para a extensão de diretório/arquivo é "_") sem a indicação do nome da unidade de disco. Um endereçamento em **notação EES** (sem a identificação de domínio "_N_", o delimitador para a extensão de diretório/arquivo é ".") porém também é permitido.

Exemplo:

- Notação NC: "/_N_SPF_DIR/_N_SUB1_SPF"
- Notação EES: "/SPF.DIR/SUB1.SPF"

Indicação

Os endereçamentos de arquivos do sistema de arquivos passivo em notação EES são convertidos internamente para a notação NC, conforme as seguintes regras:

- O nome do diretório e do arquivo são ampliados com a identificação de domínio "_N_".
- Se o quarto último caractere no nome do diretório ou do arquivo for um ponto ("."), ele será convertido em um underline ("_").

Através do nome da unidade de disco "//NC:" pré-definido, o sistema de arquivos passivo também pode ser endereçado objetivamente.

Exemplo:

- Notação NC: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- Notação EES: "//NC:/SPF.DIR/SUB1.SPF"

Endereçamento dos arquivos em uma memória de programa externa

O endereçamento de arquivos de uma memória externa do programa, que não for identificada como GDIR, deve ser feito em notação EES. No início do atalho de endereçamento deve ser indicado o nome da unidade de disco (p. ex. "//DEV1:"). Todos os nomes de dispositivos simbólicos projetados em /user/sinumerik/hmi/cfg/logdrive.ini são permitidos.

Exemplo:

- Notação EES: "//DEV1:/MyProgDir/pp1.xxx"
- Notação NC: não permitida

Endereçamento dos arquivos da memória global do programa de peças (GDIR)

No caso de endereçamento de arquivos do GDIR, além da indicação do atalho em notação EES, também é permitida uma indicação do atalho em notação NC.

Exemplo:

- Notação EES: "//DEV2:/MPF.DIR/PROG11.MPF"
- Notação NC: "/_N_MPF_DIR/_N_PROG11_MPF"

Indicação

Os endereçamentos de arquivos do GDIR em notação NC são convertidos para a notação EES, conforme as seguintes regras:

- A identificação de domínio "_N_" no nome do diretório e do arquivo é excluída.
- Se o quarto último caractere no nome do diretório ou do arquivo for um underline ("_"), ele será convertido em um ponto (".").

Regras para a indicação do atalho

Uma completa indicação do atalho é constituído pelo nome da unidade de disco, o atalho do diretório e o nome do arquivo.

Nome da unidade

Para a indicação do nome da unidade de disco aplicam-se as seguintes regras:

- Todos os nomes de dispositivos simbólicos projetados em /user/sinumerik/hmi/cfg/logdrive.ini são permitidos.
- No início encontra-se o símbolo "//", seguido de pelo menos uma letra ou um dígito.
- Os caracteres seguintes podem se utilizados em qualquer combinação de letras, dígitos, "_" e espaços.
- O nome é concluído com uma letra ou um dígito, seguido por um ":".
- Outros caracteres especiais não são permitidos.

Indicação

Para o sistema de arquivos passivo é pré-definido o nome da unidade de disco "//NC:".

Exemplos:

- Memória externa de programa:
  - //Drive1:
  - //Drive_1:
  - //Drive 1:
  - //A B:
  - //1 B C 2:

Atalho de diretório

Para a indicação do atalho do diretório aplicam-se as seguintes regras:

- No início e no final do atalho do diretório e como delimitador para as seções individuais do atalho é utilizado "/".

  Indicação

  Uma barra dupla ("//") no interior do atalho do diretório **não** é permitida!

- Nome do diretório:
  - Os nomes de diretório devem começar com uma letra ou um dígito. Somente em caso de endereçamento em notação NC, os nomes de diretório começam com a identificação de domínio "_N_".
  - Os caracteres seguintes podem ser utilizados em qualquer combinação de letras, dígitos e "_".

    Indicação

    No caso das memórias externas do programa também são permitidos os espaços no nome do diretório. No entanto, isto não se aplica, quando a memória externa do programa tiver sido criada como memória do programa de peças global (GDIR).

  - Outros caracteres especiais não são permitidos.
- Extensões do diretório:
  - As extensões do diretório devem ser constituídos exatamente por três letras/dígitos.
  - Elas são separadas dos nomes de diretório com "_" (notação NC) ou "." (notação EES).

  Indicação

  No sistema de arquivos passivo existem somente as extensões de diretório _DIR e _WPD.

Exemplos:

- Sistema de arquivos passivo ou GDIR:
  - Notação NC: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
  - Notação EES: WKS.DIR/MYPROGS.WPD/...
- Memória externa de programa:
  - /abc
  - /ab_c.def
  - /ab c1.def
  - /a b c .d11
  - /abc.def/ghi.klm

Nome de arquivo

Para o nome de arquivo aplicam-se as seguintes regras:

- Somente em caso de endereçamento em notação NC, os nomes de arquivos começam com a identificação de domínio "_N_".
- Os dois caracteres seguintes devem ser duas letras ou um traço inferior seguido por uma letra.

  Indicação

  Quando esta condição estiver preenchida, então um programa NC, através do nome de programa, pode ser chamado como subrotina por outro programa. Se o nome do programa for iniciado com números, então não será possível realizar a chamada de subrotina através da instrução CALL.

- Os caracteres seguintes podem ser utilizados em qualquer combinação de letras, dígitos e "_".
- Extensão de arquivo:
  - A extensão do arquivo deve ser constituída exatamente por três letras/dígitos.

    Indicação

    Consultar as extensões de arquivo permitidas no sistema de arquivos passivo em "Memória de programa em NC (Página 223)".

  - Elas são separadas dos nomes de arquivo com "_" (notação NC) ou "." (notação EES).

Exemplos:

- Sistema de arquivos passivo ou GDIR:
  - Notação NC: _N_SUB1_SPF
  - Notação EES: SUB1.SPF
- Memória externa de programa:
  - Parte 1
  - _Parte 1
  - Teil_1.spf
  - Teil1.mpf

Nome da sub-rotina DIN

Para os nomes de sub-rotinas DIN aplicam-se as seguintes regras:

- O primeiro caractere é a letra "L".
- Os caracteres seguintes são dígitos (pelo menos um).
- Extensão de arquivo:
  - A extensão do arquivo deve ser constituída exatamente por três letras.
  - Elas são separadas dos nomes de arquivo com "_" (notação NC) ou "." (notação EES).

Exemplos:

- L123
- L1_SPF (notação NC) ou L1.SPF (notação EES)

Comprimento máximo do atalho

Para a indicação do nome da unidade de disco e do atalho do diretório estão disponíveis no máximo 128 Bytes, o comprimento do nome do arquivo deve ser de no máximo 31 Bytes. Para o completo atalho resulta um comprimento máximo de 159 Bytes.

3.1.4 Atalho de busca na chamada de subrotinas

Durante as chamadas de sub-rotinas sem a indicação de atalho é determinado o atalho absoluto por meio da execução de um atalho fixo de busca.

As memórias de programa são pesquisadas nesta ocasião na seguinte sequência:

		Diretório	Descrição
1		diretório atual / *nome*	O diretório atual é o diretório, no qual ocorreu a seleção do programa. Isto pode ser: • um diretório da peça de trabalho ou o diretório padrão _N_MPF_DIR na memória do programa de peças NC ou na memória global do programa de peças ou • qualquer diretório de uma memória de programa externa
2		Diretório atual / *nome_SPF*	
3		Diretório atual / *nome_MPF*	
4	a	//NC:/_N_SPF_DIR / *nome_SPF*	Diretório da sub-rotina na memória do programa de peças NC
	b	//DEV2:/_N_SPF_DIR / nome_SPF [1]	Diretório da sub-rotina na memória global do programa de peças **Nota:** esta etapa de busca é omitida quando não tiver sido criada nenhuma memória global do programa de peças ou a seleção do programa tiver ocorrido na memória do programa de peças NC.
5		Extensão do atalho de busca programada com `CALLPATH` (consultar "Ampliar o caminho de localização na chamada de subrotinas (CALLPATH) (Página 209)"). **Nota:** esta etapa de busca é omitida quando não tiver sido programado nenhum `CALLPATH`.	
6		/_N_CUS_DIR / *nome_SPF*	Diretório dos ciclos de usuário
7		/_N_CMA_DIR / *nome_SPF*	Diretório dos ciclos do fabricante
8		/_N_CST_DIR / *nome_SPF*	Diretório dos ciclos padrão

[1] //DEV2:" é colocado por exemplo, para a unidade, na qual foi criada a memória global do programa de peças.

Para a busca aplicam-se as seguintes regras:

- O atalho de busca é percorrido para cada chamada de sub-rotina individual, isto é, será irrelevante, onde o programa secundário se encontra.
- Dependendo do diretório são considerados diversos tipos de arquivo.
- Fundamentalmente é pesquisado em um diretório, não no diretório de nível mais baixo, isto é, secundário.

3.1.5 Consulta do atalho e do nome do arquivo

Para a consulta do atalho e do nome do arquivo de um programa NC encontram-se disponíveis as seguintes variáveis de sistema legíveis no programa de peças:

Variável de sistema	Tipo	Significado
$P_STACK	INT	Fornece os níveis de programa, nos quais o programa NC atual é executado.
$P_PATH[<n>]	STRING	Fornece o atalho do programa NC, que é executado através do índice de campo <n> do nível selecionado do programa. Exemplos: $P_PATH[0] fornece o atalho para o programa principal, por ex., "/_N_WKS_DIR/_N_WELLE_WPD/". $P_PATH[$P_STACK - 1] fornece o atalho do programa a ser chamado. Se o atalho se referir a um programa NC, que estiver armazenado no sistema de arquivos passivo do NC ou na memória global do programa de peças (GDIR), o atalho será fornecido em notação NC. Se o atalho se referir a um programa NC que for executado a partir de uma outra memória de programa externa como memória global do programa de peças, $P_PATH fornece o atalho em EES-Notation.
$P_PROG[<n>]	STRING	Fornece o nome do programa NC que é executado através do índice de campo <n> do nível selecionado do programa. Se o programa NC estiver armazenado no sistema de arquivos passivo do NC ou na memória global do programa de peças, o nome do programa será fornecido em notação NC. Se o programa NC for executado a partir de uma outra unidade de disco externa como a memória global do programa de peças, $P_PROG fornece o nome em EES-Notation.
$P_PROGPATH	STRING	Fornece o atalho do programa NC, que está sendo executado no momento. A chamada de $P_PROGPATH é idêntica ao $P_PATH[$P_STACK].

Variável de sistema	Tipo	Significado	
$P_IS_EES_PATH[<n>]	BOOL	Consulta, se o atalho fornecido por $P_PATH[<n>] ou o nome de programa fornecido por $P_PROG[<n>] corresponde à notação NC ou à notação EES.	
		= FALSE	$P_PATH[<n>] e $P_PROG[<n>] fornecem notação NC. Isto é, cada identificador é precedido de um prefixo "_N_". O traço de separação para a identificação do arquivo é "_". Exemplos: • Atalho na notação NC: "/_N_WKS_DIR/_N_MYWPD_WPD/" • Nome do programa em notação NC:"_N_MYPROG_MPF" Um atalho em notação NC tanto pode se referir ao sistema de arquivos passivo em NC, como também à memória global do programa de peças.
		= TRUE	$P_PATH[<n>] e $P_PROG[<n>] fornece notação EES. Isto é, o identificador não é precedido de nenhum prefixo "_N_". O traço de separação para a identificação do arquivo é ".". Exemplos: • Atalho na notação EES: "//DEV1:/WKS.DIR/MYWPD.WPD/" • Nome do programa em EES-Notation: "MYPROG.MPF"

<n>: O índice <n> define o nível de programa, a partir do qual a informação sobre o atalho deverá ser lida (faixa de valores: 0 ... 17)

Indicação

Em operação EES, fora da memória global do programa de peças (GDIR), são fornecidos pelas variáveis do sistema $P_PROG, $P_PATH, e $P_PROGPATH os nomes de atalho em EES-Notation. Os programas de usuário, que avaliam e processam estes nomes de atalho posteriormente, devem ser ampliados por isto para a operação EES de tal modo, que eles também possam processar os nomes de atalho em EES-Notation.

3.2 Memória de trabalho (CHANDATA, COMPLETE, INITIAL)

Função

A memória de trabalho contém os atuais dados de sistema do usuário, com os quais se opera o comando (sistema de arquivos ativo), p. ex.:

- Dados de máquina ativos
- Dados de corretores de ferramentas
- Deslocamentos de ponto zero
- ...

Programas de inicialização

Aqui trata-se de programas com os quais os dados da memória de trabalho são inicializados. Para isso podem ser utilizados os seguintes tipos de arquivos:

Tipo de arquivo	Descrição
nome_TEA	Dados de máquina
nome_SEA	Dados de ajuste
nome_TOA	Correções de ferramentas
nome_UFR	Deslocamentos de ponto zero/Frame
nome_INI	Arquivo de inicialização
nome_GUD	Dados de usuário globais
nome_RPA	Parâmetros R

Áreas de dados

Os dados podem ser organizados em diferentes áreas onde serão aplicados. Por exemplo, um comando pode dispor de vários canais ou, como de costume, dispor de vários eixos.

Existe:

Identificação	Áreas de dados
NC	Dados específicos NC
CH<n>	Dados específicos de canal (<n> indica o número do canal)
AX<n>	Dados específicos de eixo (<n> especifica o número do eixo da máquina)
TO	Dados de ferramenta
COMPLETE	Todos dados

Criar programa de inicialização em um PC externo

Com o auxílio do identificador da área de dados e do tipo de arquivo podem ser determinadas áreas que serão consideradas como uma unidade durante a gravação de segurança dos dados:

_N_AX5_TEA_INI	Dados da máquina para 5 eixos
_N_CH2_UFR_INI	Frames do canal 2
_N_COMPLETE_TEA_INI	Todos dados de máquina

Após a colocação em funcionamento do comando existe um bloco de dados na memória de trabalho que assegura o funcionamento correto do comando.

Procedimento com comandos de vários canais (CHANDATA)

O `CHANDATA(<número do canal>)` para vários canais somente é permitido no arquivo _N_INITIAL_INI. Este é o arquivo de colocação em funcionamento com o qual são inicializados todos dados do comando.

Código de programa	**Comentário**
`%_N_INITIAL_INI`	
`CHANDATA(1)`	
	`; Atribuição de eixo de máquina do canal 1:`
`$MC_AXCONF_MACHAX_USED[0]=1`	
`$MC_AXCONF_MACHAX_USED[1]=2`	
`$MC_AXCONF_MACHAX_USED[2]=3`	
`CHANDATA(2)`	
	`; Atribuição de eixo de máquina do canal 2:`
`$MC_AXCONF_MACHAX_USED[0]=4`	
`$MC_AXCONF_MACHAX_USED[1]=5`	
`CHANDATA(1)`	
	`; Dados de máquina de eixo:`
	`; Janela de parada exata aproximada:`
`$MA_STOP_LIMIT_COARSE[AX1]=0.2`	`; Eixo 1`
`$MA_STOP_LIMIT_COARSE[AX2]=0.2`	`; Eixo 2`
	`; Janela de parada exata fina`
`$MA_STOP_LIMIT_FINE[AX1]=0.01`	`; Eixo 1`
`$MA_STOP_LIMIT_FINE[AX1]=0.01`	`; Eixo 2`

ATENÇÃO

Instrução CHANDATA

No programa de peças, a instrução `CHANDATA` somente pode ser definida para o canal onde o programa NC será executado. Isto é, o comando pode ser utilizado para proteger programas NC para que não sejam executados acidentalmente em outro canal.

A execução do programa será cancelada em caso de erro.

Indicação

Os arquivos INI em listas de tarefas não contém nenhuma instrução `CHANDATA`.

Salvar programas de inicialização (COMPLETE, INITIAL)

Os arquivos da memória de trabalho podem ser salvos em um PC externo e deste recarregados.

- Os arquivos são salvos com `COMPLETE`.
- Com `INITIAL` se gera um arquivo INI (_N_INITIAL_INI) que se estende em todas as áreas.

Carregar programas de inicialização

ATENÇÃO

Perda de dados

Quando se carrega o arquivo de nome "INITIAL_INI", todos dados que não são alimentados com o arquivo serão inicializados com dados padrão. Disso estão excluídos apenas os dados de máquina. Os **dados de ajuste, dados de ferramenta, NPV, valores GUD, ...** também são alimentados com dados padrão (normalmente é o "ZERO").

Por exemplo, o arquivo COMPLETE_TEA_INI é apropriado para o carregamento de dados individuais de máquina. Deste arquivo o comando apenas espera os dados de máquina. Dessa forma as demais áreas de dados permanecem inalteradas neste caso.

Carregar programas de inicialização

Os programas INI também podem ser selecionados e executados como programas de peças, isto se utilizarem apenas os dados de um canal. Assim também é possível inicializar dados controlados pelo programa.

Áreas de proteção

4

4.1 Definir áreas de proteção (CPROTDEF, NPROTDEF)

Áreas de proteção que devem proteger elementos da máquina de colisões, são definidas no programa de peças em respectivos blocos. Estes contêm os seguintes elementos:

1. Determinação do plano de trabalho
   Antes da definição de fato da área de proteção, precisa ser selecionado o plano de trabalho ao qual deve-se referir a descrição do contorno da área de proteção.
2. Início da definição
   Conforme o comando NC, é criado uma área de proteção específica ou do canal ou da máquina.
3. Descrição do contorno da área de proteção
   O contorno de uma área de proteção é descrita por meio de movimentos de deslocamento. Estes não são executados e não têm relação com descrições geométricas anteriores ou posteriores. Apenas definem a área de proteção.
4. Fim da definição

Sintaxe

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/... X/Y/Z...
...
EXECUTE(<Var>)
```

Significado

`DEF INT <Var>`:	Definição de uma variável auxiliar local do tipo de dados INTEGER
`<Var>`:	Nome da variável auxiliar
`G17/G18/G19`:	Plano de trabalho **Aviso:** O plano de trabalho não deve ser alterado antes do fim da definição. Uma programação da aplicada entre o início e o fim da definição não é permitida.
`CPROTDEF()`:	Procedimento predefinido para a definição de uma área de proteção específica do **canal**
`NPROTDEF()`:	Procedimento predefinido para a definição de uma área de proteção específica da **máquina**

<n>:	Número da área de proteção definida		
	Tipo de dado:	INT	
<t>:	Tipo da área de proteção		
	Tipo de dado:	BOOL	
	Valor:	TRUE	Área de proteção em função da **ferramenta**
		FALSE	Área de proteção em função da **peça de trabalho**
<AppLim>:	Tipo de limitação na 3ª dimensão		
	Tipo de dado:	INT	
	Valor:	0	Sem limitação
		1	Limitação no sentido positivo
		2	Limitação no sentido negativo
		3	Limitação nos sentidos positivo e negativo
<AppPlus>:	Valor da limitação no sentido positivo da 3ª dimensão		
	Tipo de dado:	REAL	
<AppMinus>:	Valor da limitação no sentido negativo da 3ª dimensão		
	Tipo de dado:	REAL	

`G0/G1/... X/Y/Z... ...:`	O contorno de uma área de proteção é descrito por meio de no máximo 11 movimentos de deslocamento dentro do plano de trabalho selecionado. Neste caso o primeiro movimento de percurso é o movimento no contorno. O último ponto da descrição do contorno sempre deve coincidir com o primeiro ponto da descrição do contorno. Como área de proteção vale a área à esquerda do contorno. • Área interna de proteção O contorno para uma área interna de proteção deve ser descrito no sentido anti-horário. • Área externa de proteção (admissível somente para áreas de proteção relativas a peças de trabalho!) O contorno para uma área externa de proteção deve ser descrito no sentido horário. Os seguintes elementos de contorno são admissíveis: • `G0`, `G1` para elementos de contorno retos • `G2` para segmentos circulares no sentido horário Admissíveis somente para áreas de segurança relativas a peças de trabalho! Não admissíveis para áreas de proteção relativas a ferramentas, uma vez que só podem ser convexas. • `G3` para segmentos circulares no sentido anti-horário **Aviso:** Uma área de proteção não pode ser descrita por um círculo inteiro. Um círculo inteiro precisa ser dividido em dois círculos parciais. **Aviso:** A ordem `G2` → `G3` ou. `G3` → `G2` não é admissível! Entre os dois blocos circulares deve ser inserido um bloco `G1` curto.
`EXECUTE(<Var>):`	Procedimento predefinido que marca o fim da definição Com `EXECUTE` aciona-se a mudança de volta ao processamento normal do programa.

Exemplo

Veja o exemplo em "Ativar/desativar áreas de proteção (CPROT, NPROT) (Página 244)".

Outras informações

Áreas de proteção específicas da máquina

A definição de uma área de proteção específica da máquina, respectivamente do seu contorno, ocorre por meio dos eixos geométricos, ou seja em relação ao sistema básico de coordenadas (BCS) de um canal. Para que seja possível o monitoramento correto da área de proteção em todos os canais, onde a área de proteção específica da máquina se encontra ativa, é necessário que o sistema básico de coordenadas (BCS) de todos os canais envolvidos seja idêntico:

- Posição da origem das coordenadas em relação ao ponto zero da máquina
- Orientação dos eixos de coordenada

Ponto de referência da descrição do contorno

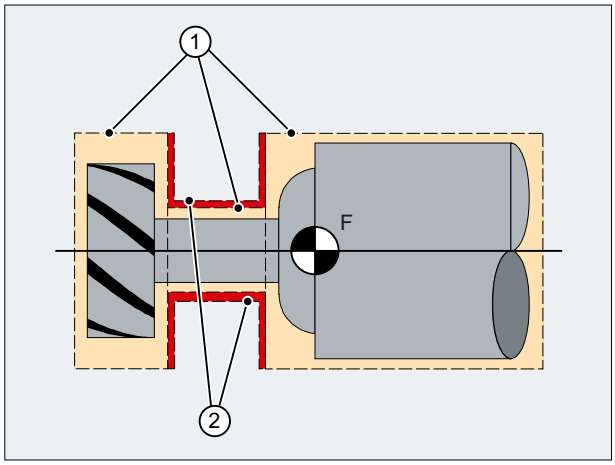
- Áreas de proteção relativas à ferramenta
  As coordenadas para áreas de proteção relativas a **ferramentas** devem ser indicadas de modo absoluto, com referência ao **ponto de referência do porta-ferramenta F**.
- Áreas de proteção relativas à peça de trabalho
  As coordenadas para áreas de proteção relativas à **peça de trabalho**devem ser indicadas de modo absoluto, com referência ao ponto zero do **sistema básico de coordenadas (BCS)**.

Áreas de proteção simétricas na rotação

Para áreas de proteção simétricas na rotação (por ex. mandril do fuso) precisa ser descrito o contorno inteiro, não apenas o contorno até o centro da rotação.

Áreas de proteção relativas à ferramenta

Áreas de proteção relativas à ferramenta sempre devem ser convexas. Se for desejada uma área de proteção côncava, esta deve ser distribuída em várias áreas de proteção convexas.



① Áreas de proteção convexas
② Áreas de proteção côncavas (**não admissíveis!**)
F Ponto de referência do porta-ferramenta

Condições gerais

Durante a definição de uma área de proteção, as seguintes funções não devem estar ativas ou sendo utilizadas:

- Correção de raio de ferramenta (correção de raio de fresa, correção de raio de aresta cortante)
- Transformação
- Aproximação do ponto de referência (`G74`)
- Aproximação do ponto fixo (`G75`)
- Tempo de espera (`G4`)
- Parada de pré-processamento do bloco (`STOPRE`)

- Fim do programa (`M17`, `M30`)
- Funções M: `M0`, `M1`, `M2`

4.2 Ativar/desativar áreas de proteção (CPROT, NPROT)

Áreas de proteção definidas previamente no programa de peças podem ser ativadas a qualquer momento ou podem ser pré-ativadas através do programa do usuário CLP para uma ativação posterior. Áreas de proteção ativas também podem ser desativadas a qualquer momento.

Na ativação, ou pré-ativação respectivamente, existe a possibilidade adicional de deslocar, de modo relativo, o ponto de referência da área de proteção.

Indicação

Uma área de proteção somente é considerada após referenciar todos os eixos geométricos do canal, no qual foi ativada.

Indicação

Monitoramento das áreas de proteção

Quando nenhuma área de proteção relativa à ferramenta está ativa, o trajeto da ferramenta é verificado em relação às áreas de proteção relativas à peça de trabalho.

Quando nenhuma área de proteção relativa à peça de trabalho está ativa, não há monitoramento de áreas de proteção.

Sintaxe

```
CPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
NPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
```

Significado

CPROT:	Procedimento predefinido para a ativação de uma área de proteção específica do **canal**		
NPROT:	Procedimento predefinido para a ativação de uma área de proteção específica da **máquina**		
<n>:	Número da área de proteção		
	Tipo de dado:	INT	
<Status>:	Por meio deste parâmetro coloca-se o status de ativação específico do canal		
	Tipo de dado:	INT	
	Valor:	0	Desativação da área de proteção
		1	Pré-ativação da área de proteção
		2	Ativação da área de proteção
		3	Pré-ativação da área de proteção com parada condicional

<XMov>,<YMov>,<ZMov>:	Valores aditivos de deslocamento no sentido X/Y/Z O deslocamento pode ser realizado em 1, 2 ou 3 dimensões. Os valores de deslocamento referem-se: • ao ponto zero da máquina no caso de uma área de proteção relativa à **peça de trabalho** • ao ponto de referência do porta-ferramenta F no caso de uma área de proteção relativa à **ferramenta**	
	Tipo de dado:	REAL

Exemplo

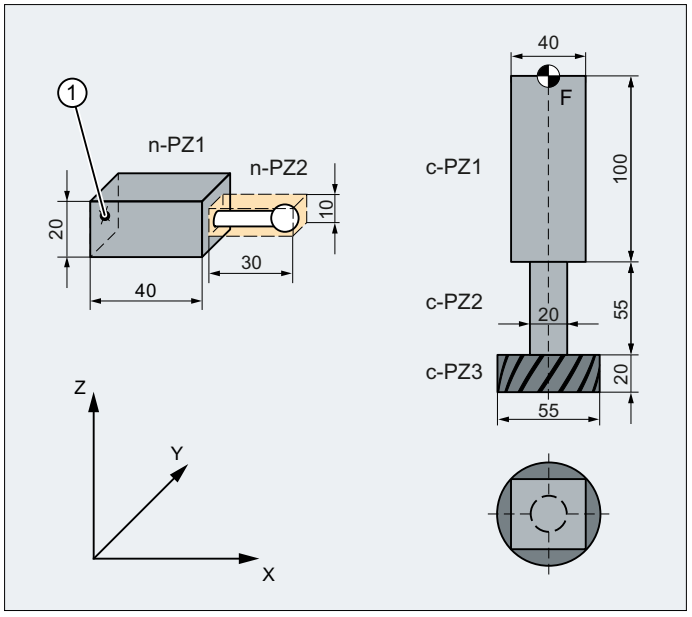
Para uma fresadora deve ser monitorada uma possível colisão da fresa com o apalpador de medição. A posição do apalpador de medição deve ser especificado por um deslocamento para a ativação.

Para isso são definidas as seguintes áreas de proteção:

- Uma área de proteção cada, específica da máquina e relativa à peça de trabalho para o suporte do apalpador de medição (n-PZ1) e para o próprio apalpador de medição (n-PZ2).
- Uma área de proteção cada, específica do canal e relativa à ferramenta para o porta-fresa (c-PZ1), a haste da fresa (c-PZ2) e para a própria fresa (c-PZ3).

A orientação de todas áreas de proteção está no sentido Z.

A posição do ponto de referência do apalpador de medição para a ativação deve estar em X = -120, Y = 60 e Z = 80.



① Ponto de referência para a área de proteção do apalpador de medição

F Ponto de referência do porta-ferramenta

Código de programa	**Comentário**
DEF INT PROTZONE	; Definição de uma variável auxiliar
G17	; Plano de trabalho XY
; Definição das áreas de proteção:	
NPROTDEF(1,FALSE,3,10,-10)	; Área de proteção n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE(PROTZONE)	
NPROTDEF(2,FALSE,3,5,-5)	; Área de proteção n-PZ2
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE(PROTZONE)	
CPROTDEF(1,TRUE,3,0,-100)	; Área de proteção c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	
Y-20	
EXECUTE(PROTZONE)	
CPROTDEF(2,TRUE,3,-100,-150)	; Área de proteção c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE(PROTZONE)	
CPROTDEF(3,TRUE,3,-150,-170)	; Área de proteção c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE(PROTZONE)	
; Ativação das áreas de proteção:	
NPROT(1,2,-120,60,80)	; Ativar a área de proteção -PZ1 com deslocamento
NPROT(2,2,-120,60,80)	; Ativar a área de proteção -PZ2 com deslocamento
CPROT(1,2,0,0,0)	; Ativar a área de proteção c-PZ1
CPROT(2,2,0,0,0)	; Ativar a área de proteção c-PZ2
CPROT(3,2,0,0,0)	; Ativar a área de proteção c-PZ3

Outras informações

Status da ativação após inicialização do sistema de comando

Uma área de proteção já pode estar ativa após a inicialização do sistema de comando e após referenciar os eixos. Isto é o caso quando a seguinte variável de sistema para a área de proteção estiver ajustada para TRUE:

- $SN_PA_ACTIV_IMMED[<n>] (para área de proteção específica da máquina) ou
- $SC_PA_ACTIV_IMMED[<n>] (para área de proteção específica do canal)
  O índice "<n>" corresponde ao número da área de proteção: 0 = 1. Área de proteção

A área de proteção é ativada com o status = 2 e sem deslocamento.

Ativações múltiplas de uma área de proteção

Uma área de proteção também pode ser efetiva simultaneamente em vários canais (por ex. área de proteção do eixo tubular em duas unidades de avanço em lados opostos). A monitoração das áreas de proteção somente será realizada quando todos eixos geométricos estiverem referenciados.

Uma área de proteção não pode ser ativada em um canal simultaneamente com diferentes deslocamentos.

Monitoração da área de proteção com correção do raio da ferramenta ativada

Com a correção do raio da ferramenta ativada somente é possível uma monitoração da área de proteção que funcione, se o plano da correção do raio da ferramenta for idêntico ao plano das definições da área de proteção.

4.3 Verificação quanto à violação da área de proteção, limite da área de trabalho e chave fim de curso de software (CALCPOSI)

Função

A função CALCPOSI verifica no sistema de coordenadas da peça de trabalho (WCS), se, partindo da posição inicial, os **eixos geométricos** podem percorrer uma trajetória especificada sem violar limitações ativas. Se o deslocamento não pode ser completado devido a limitações, um valor de status positivo, codificado de modo decimal, e o deslocamento máximo possível são devolvidos.

Definição

```
INT CALCPOSI(VAR REAL[3] <Início>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

Sintaxe

```
<Condição> = CALCPOSI(VAR <Início>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

Significado

`CALCPOSI(...)`:	Função predefinida para a verificação quanto a violações de limitações relativas aos eixos geométricos	
	Parada de pré-processamento:	não
	Sozinho no bloco:	sim

<Status>: (Parte 1)	Valor de devolução da função. Valores negativos mostram condições de erro.		
	Tipo de dado:		INT
	Faixa de valores:		-8 ≤ x ≤ 100000
	Valor:	0	O deslocamento pode ser completamente inicializado
		-1	No <Limit>, pelo menos um componente é negativo.
		-2	Erro no cálculo de transformação. Exemplo: O curso do deslocamento passa por uma singularidade, de maneira que as posições dos eixos não estão definidas.
		-3	O deslocamento indicado <Dist> e o deslocamento máximo possível <MaxDist> são interdependentes no sentido linear. **Nota** ode apenas ocorrer em relação com <TestLim>, Bit 4 == 1.
		-4	A projeção na direção de deslocamento sobre a área limitante contida no <Dist> é o vetor zero e a direção do deslocamento está perpendicular à área limitante violada. **Nota** ode apenas ocorrer em relação com <TestLim>, Bit 5 == 1.
		-5	Em<TestLim> são Bit 4 == 1 UND Bit 5 == 1
		-6	Pelo menos um eixo da máquina, que deve ser considerado para a verificação dos limites de deslocamento, não está referenciado.
		-7	Função de prevenção de colisão: Definição não válida da cadeia cinemática ou das áreas de proteção.
		-8	Função de prevenção de colisão: a função pode não ser executada devido insuficiência de memória.
<Status>: (Parte 2)	Posição da unidade **Aviso** Em caso de violação simultânea de várias limitações, aquela que causa a restrição mais severa do deslocamento previsto é comunicada.		
	Valor:	1	Chave fim de curso de software limita o deslocamento
		2	Limitação do campo de trabalho limita o deslocamento
		3	Áreas de proteção limitam o deslocamento
		4	Função de prevenção de colisão: Áreas de proteção limitam o deslocamento
	Posição da dezena		
	Valor:	1x	O valor inicial viola o limite
		2x	A reta especificada viola o limite. Este valor também é devolvido, se o próprio ponto final não viola nenhum limite, porém uma violação de um valor limite ocorreria no percurso do ponto inicial ao ponto final (por ex., ao atravessar uma área de proteção, chaves de fim do curso do software deformadas no WCS por transformações não lineares, p. ex. Transmit).

<Status>: (Parte 3)	Posição da centena		
	Valor:	1xx	UND posição unidade == 1 ou 2: O valor-limite positivo está violado.
			E posição da unidade == 3 [1]: Espec. ao NO. Área de proteção violada.
		2xx	UND posição unidade == 1 ou 2: O valor-limite positivo está violado.
			UND posição unidade == 3 [1]: Encontra-se violada uma área de proteção específica do canal.
<Status>: (Parte 4)	Posição da milhar		
	Valor:	1xxx	UND posição unidade == 1 ou 2: Fator, com o qual é multiplicado o número do eixo que viola a limitação. A numeração do eixo se inicia em 1. Referência: • Chave fim de curso de software: Eixos de máquina • Limite da área de trabalho: Eixos geométricos UND posição unidade == 3 [1]: Fator com o que o número de áreas de proteção violadas é multiplicado.
<Status>: (Partel 5)	Posição centena de milhar		
	Valor:	0xxxxx	Posição de centenas de milhar == 0: <Dist> permanece sem alteração
		1xxxxx	Em <Dist> é devolvido um vetor de direção, o qual define a continuação da direção de movimento na área da limitação. Pode ocorrer apenas nas seguintes condições: • Chave fim de curso do software ou limite do campo de trabalho violados (não no ponto inicial) • Uma transformação não é ativa • <TestID>, Bit 4 ou Bit 5 == 1
<início>:	Referência sobre um vetor com as posições iniciais: • <Start> [0]: 1. Eixo geométrico • <Start> [1]: 2º Eixo geométrico • <Start> [2]: 3. Eixo geométrico		
	Tipo de parâmetro:		Entrada
	Tipo de dado:		VAR REAL [3]
	Faixa de valores:		máx. Valor REAL ≤ x[<n>] ≤ +máx. Valor REAL

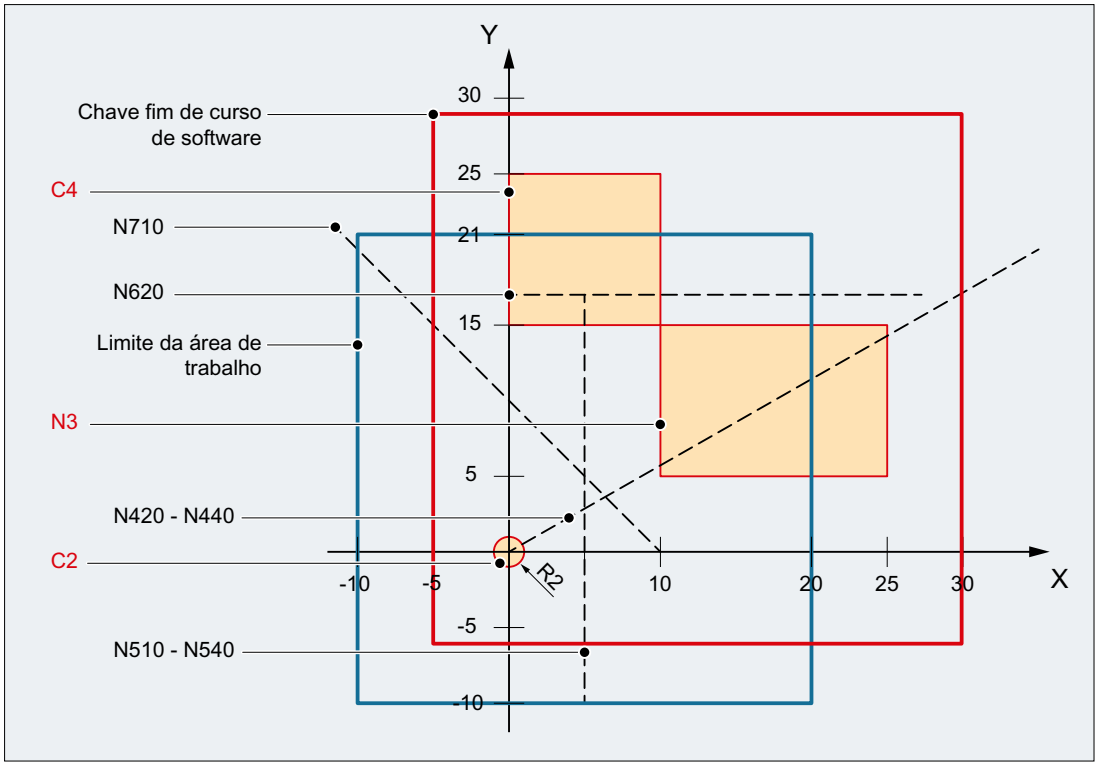
<Dist>:	Referência sobre um vetor	
	<< : deslocamento incremental • <Dist> [0]: 1. Eixo geométrico • <Dist> [1]: 2º Eixo geométrico • <Dist> [2]: 3. Eixo geométrico	
	Saída (apenas com posição de centena de milhar definida no <Status>):	
	<Dist> contém como valor de saída um vetor unitário **v**, o qual define a continuação da direção de deslocamento no WCS.	
	Caso 1: Formação do vetor **v** em<TestID>, Bit 4 == 1 Os vetores de entrada <Dist> e <MaxDist> fixam o plano de movimento. Este plano será cortado com a área limite violada. A linha de interseção dos dois planos define a direção do vetor **v**. Neste processo, a orientação (sinal) é selecionada de maneira que o ângulo entre o vetor de entrada <MaxDist> e **v** não exceda os 90 graus.	
	Caso 2: Formação do vetor **v** em<TestID>, Bit 5 == 1 O vetor **v** é o vetor de unidade na direção da projeção do vetor de deslocamento contido em<Dist> da área limite. Se a projeção do vetor de deslocamento sobre a área limite for o vetor nulo, será devolvido um erro.	
	Tipo de parâmetro:	Entrada/saída
	Tipo de dado:	VAR REAL [3]
	Faixa de valores:	máx. Valor REAL ≤ x[<n>] ≤ +máx. Valor REAL
<Limit>:	Referência sobre um campo do comprimento 5. • <Limit> [0 - 2]: Distâncias mínimas dos eixos geométricos em relação aos limites: – <Limit> [0]: 1. Eixo geométrico – <Limit> [1]: 2º Eixo geométrico – <Limit> [2]: 3. Eixo geométrico As distâncias mínimas serão mantidas em: – Limite da área de trabalho: Sem restrições – Chave fim de curso de software: Sem nenhuma transformação ativa, ou com uma transformação ativa na qual é possível uma atribuição inequívoca dos eixos geométricos aos eixos lineares da máquina, por ex. transformações de 5 eixos. • <Limit> [3]: Contém a distância mínima para eixos de máquinas lineares que, por exemplo, devido a uma transformação não-linear de nenhum eixo geométrico não pode ser classificada. Este valor também é utilizado como valor limite no monitoramento das áreas de proteção convencionais e das áreas de proteção para prevenção de colisões. • <Limit> [4]: Contém a distância mínima para eixos de máquinas rotativos que, por exemplo, devido a uma transformação não-linear de nenhum eixo geométrico não pode ser classificada. **Nota** Este valor apenas se torna válido na monitorização do chave fim de curso do software de transformações especiais.	
	Tipo de parâmetro:	Entrada
	Tipo de dado:	VAR REAL [5]
	Faixa de valores:	máx. Valor REAL ≤ x[n] ≤ +máx. Valor REAL

<MaxDist>:	Referência sobre um vetor com um deslocamento incremental, no qual a distância mínima de um limite de eixo de todos os eixos de máquinas participantes não será pequena demais: • <Dist> [0]: 1. Eixo geométrico • <Dist> [1]: 2º Eixo geométrico • <Dist> [2]: 3. Eixo geométrico Se o deslocamento estiver reduzido, o conteúdo desse parâmtro de devolução é igual ao conteúdo de <Dist>.		
	Em <TestID>, Bit 4 == 1: <Dist> e <MaxDist> <MaxDist> e<Dist> devem conter vetores como valores de entrada que fixem um plano de movimento. Ambos os vetores devem ser independentes um do outro linearmente. O valor de <MaxDist> é almejado. Para o cálculo da direção do movimento, veja a descrição para <Dist>.		
	Tipo de parâmetro:		Saída
	Tipo de dado:		VAR REAL [3]
	Faixa de valores:		máx. Valor REAL ≤ x[<n>] ≤ +máx. Valor REAL
<MeasSys>:	Sistema de medida (inch / métrico) para indicações de posição e comprimento (**opcional**)		
	Tipo de dado:		BOOL
	Valor:	FALSE (Default)	Sistema de medidas correspondente ao atualmente ativo comando G do grupo G 13 (G70, G71, G700, G710). **Aviso** Com G70 ativo e sistema básico métrico ou G71 ativo e sistema básico de polegadas, as variáveis de sistema $AA_IW e $AA_MW são fornecidas no sistema básico e devem ser convertidas na utilização para CALCPOSI, se necessário.
		TRUE	Sistema de medidas conforme o sistema básico ajustado: MD52806 $MN_ISO_SCALING_SYSTEM
<TestLim>:	Seleção codificada em bit dos limites a serem monitorados (**opcional**)		
	Tipo de dado:		INT
	Valor padrão:		Bit 0, 1, 2, 3, 6, 7 == 1 (207)
	Bit	**Decimal**	**Significado**
	0	1	Fim de curso de Software
	1	2	Limite da área de trabalho
	2	4	Áreas de proteção convencionais ativadas
	3	8	Áreas de proteção convencionais pré-ativadas
	4	16	Em chave fim de curso de software e limites de campo de trabalho no <Dist> a direção de deslocamento correspondente **caso 1** (ver acima) restituir.
	5	32	Em chave fim de curso de software e limites de campo de trabalho no <Dist> a direção de deslocamento correspondente **caso 2** (ver acima) restituir.
	6	64	Áreas de proteção ativas da prevenção de colisão.
	7	128	Áreas de proteção da prevenção de colisão pré-ativadas.
	8	256	Pares de zonas de proteção de prevenção de colisão ativadas e pré-ativadas

[1] Havendo violação de várias áreas de proteção, aquela área de proteção que causa a mais severa restrição da trajetória prevista de deslocamento é devolvida.

Exemplo

Limitações



No exemplo estão representadas as chaves fim de curso de software efetivos e as limitações do campo de trabalho no plano X-Y e as seguintes três áreas de proteção:

- C2: Área de proteção relativa à ferramenta, específica do canal, ativa, circular, raio = 2 mm
- C4: Área de proteção relativa à ferramenta, específica do canal, pré-ativada, quadrada, comprimento do lado = 10 mm
- N3: Área de proteção específica da máquina, ativa, retangular, dimensões = 10 mm x 15 mm

Programa NC

No programa NC serão definidos primeiramente as áreas de proteção e os limites do campo de trabalho. Em seguida, será chamada a função`CALCPOSI()` com diferentes padronizações.

Código de programa

```
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
```

Código de programa

```
; Área de proteção relativa à ferramenta C2
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
; Área de proteção relativa a peça de trabalho C4
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
; Área de proteção específica da máquina N3
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
;ativar e pré-ativar áreas de proteção
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
; Limites de áreas de trabalho à definir
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 _CONDIÇÃO = CALCPOSI(_INÍCIO, _DIST, _LIMIT, _MAXDIST)
N430 _CONDIÇÃO = CALCPOSI(_INÍCIO, _DIST, _LIMIT, _MAXDIST,,3)
N440 _CONDIÇÃO = CALCPOSI(_INÍCIO, _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 _CONDIÇÃO = CALCPOSI(_INÍCIO, _DIST, _LIMIT, _MAXDIST,,14)
N520 _CONDIÇÃO = CALCPOSI(_INÍCIO, _DIST, _LIMIT, _MAXDIST,, 6)
N530 _LIMIT[1] = 2.
```

Código de programa

```
N540 _CONDIÇÃO = CALCPOSI(_INÍCIO, _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 _CONDIÇÃO = CALCPOSI(_INÍCIO, _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 _CONDIÇÃO = CALCPOSI(_INÍCIO,_DIST, _LIMIT, _MAXDIST)
; Apagar quadros de N690 e N700
N720 TRANS
N730 _START[0] =  0.
N740 _START[1] = 10.
N750 _START[2] =  0.
; Vetores _DIST e _MAXDIST definem o plano de movimento
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 _CONDIÇÃO = CALCPOSI(_INÍCIO, _DIST, _LIMIT, _MAXDIST,,17)
N830 M30
```

Resultados de CALCPOSI()

N...	<Condição>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ X	Observações
420	3123	8.040	4.594	N3 será violada.
430	1122	20.000	11.429	Sem monitoramento das áreas de proteção, limitação do campo de trabalho é violada.
440	1121	30.000	17.143	Ainda ativo apenas o monitoramento da chave fim de curso de software.
510	4213	0,000	0,000	Ponto inicial viola C4
520	0000	0,000	-27.000	C4 pré-ativado não será monitorado. O caminho especificado pode ser atravessado completamente.
540	2222	0,000	-25.000	Devido a _DLIMIT[1] = 2, a trajetória do deslocamento é restringida pela limitação do campo de trabalho.

N...	<Condição>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ X	Observações
620	4223	-13.000	0,000	Distância total para C4 é de 4 mm, devido a C2 e _LIMIT[3]. Distância C2 → N3 de 0,1 mm não leva à redução do deslocamento.
710	1221	0,000	21.213	Frame ativo com translação e rotação. A trajetória do deslocamento admissível em _DIST vale no WCS deslocado e girado.
820	102121	18.000	18.000	A chave fim de curso do eixo Y será violada. Com <_TESTLIM> = 17 é solicitado o cálculo de uma direção para a continuação do deslocamento. Esta direção está em <_DIST> (0.707, 0.0, 0.707). Ela vale, já que em <_STATUS> está ajustado o dígito de centena de milhar.

Outras informações

Status do eixo "Referenciado"

Todos os eixos de máquina `CALCPOSI()` considerados devem estar referenciados

Especificações de deslocamento referente ao círculo

Todas as especificações de deslocamento referente ao círculo são **sempre** interpretados como especificação de raio. Isto deve ser considerado especialmente nos eixos planos com programação de diâmetro ativas (`DIAMON` / `DIAM90`.

Redução do deslocamento

Se a trajetória do deslocamento indicada de um eixo for limitada, no valor de devolução `<MaxDist>` também as trajetórias de deslocamento dos outros eixos ficam proporcionalmente reduzidos. O ponto final resultante Der resultierende Endpunkt liegt dadurch weiterhin auf der vorgegebenen Bahn.

Eixos rotativos

Os eixos de rotação são monitorados somente se não há módulo de eixo rotativo.

É admissível que um ou mais eixos não tenham chave fim de curso de software e as limitações da área de trabalho ou zonas de proteção sejam definidas.

Chave fim de curso de software e status de limite da área de trabalho

Chave fim de curso de software e limite da área de trabalho são apenas considerados se eles estão ativos durante a execução do `CALCPOSI()`. O status pode ser influenciado por, por ex.:

- Dados de máquina: MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS
- Dados de ajuste: $AC_WORKAREA_CS_...
- NC/PLC-Sinais da Interface: DB31, ... DBX12.2 / 3
- Comandos: `WALIMON` / `WALIMOF`

Chave fim de curso de software e transformações

Com CALCPOSI(), em caso de diferentes transformações cinemáticas (por ex. TRANSMIT) e por causa de ambiguidades em determinados pontos da área de deslocamento, as posições dos eixos da máquina (MCS) não podem ser determinadas sempre de modo inequívoco a partir das posições dos eixos geométricos (WCS). No deslocamento normal, a clareza geralmente resulta dos antecendentes e da condição que de um movimento contínuo em WCS deve corresponder um movimento contínuo dos eixos da máquina em MCS. No monitoramento da chave fim de curso de software, a posição existente da máquina CALCPOSI() para o momento de execução é utilizada em tais casos para a resolução de ambiguidade.

Indicação

Parada de pré-processamento

Na utilização de CALCPOSI() em relação a transformações, é de responsabilidade exclusiva do usuário, programar, para a sincronização das posições dos eixos da máquina com o avanço, antes de CALCPOSI() uma parada de pré-processamento (STOPRE).

Âmbito e distância de zonas de protecção convencionais

Em áreas de proteção convencionais **não** é garantido que, no caso de um movimento de deslocamento dentro da trajetória de deslocamento predeterminada, seja mantida a distância de segurança em relação a todas as áreas de proteção, indicada no parâmetro <Limit>[3], Só é garantido que com uma extensão, pela distância de segurança na direção do deslocamento, do ponto final devolvido em <Dist>, não ocorra nenhuma violação de área de proteção. A reta pode seguir sua trajetória, mas qualquer proximidade de uma área protegida além de executada.

Âmbito e distância de zonas de proteção anti-colisão

Em áreas de proteção para prevenção de colisões é garantido, que durante um movimento de deslocamento na trajetória do deslocamento predefinida, seja mantida, em relação a todas as áreas de proteção, a distância de segurança indicada no parâmetro <Limit>[3].

A distância de segurança indicada no parâmetro <Limit>[3] se torna efetiva somente com a vigência de:

<Limit>[3] > (MD10619 $MN_COLLISION_TOLERANCE)

Se, no parâmetro <TestLim> estiver definido o bit 4 (cálculo da direção para a continuação do deslocamento), o vetor de direção contido em <DIST> só terá validade se estiver definido o dígito de centena de milhar no valor de devolução da função (<Condição>). Se não for possível determinar tal direção, seja por violação de áreas de proteção ou por transformação ativa, o valor de entrada em <DIST> permanece inalterado. Não ocorre nenhuma outra mensagem de erro.

Comandos especiais de curso 5

5.1 Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN)

Através dos seguintes comandos podemos deslocar eixos lineares e rotativos através de números de posição que estão registrados como posições fixas de eixo em tabelas de dados da máquina. Este tipo de programação é denominado de "Aproximação de posições codificadas".

Sintaxe

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

Significado

`CAC(<n>)`:	Aproximar posição codificada de número de posição n
`CIC(<n>)`:	Aproximar a posição codificada, partindo-se do atual número de posição, n locais de posição para frente (+n) ou para trás (–n)
`CDC(<n>)`:	Aproximar posição codificada de número de posição n pelo percurso mais curto (apenas para eixos rotativos)
`CACP(<n>)`:	Aproximar posição codificada de número de posição n em sentido positivo (apenas para eixos rotativos)
`CACN(<n>)`:	Aproximar posição codificada de número de posição n em sentido negativo (apenas para eixos rotativos)
`<n>`:	Número de posição dentro da tabela de dados da máquina Faixa de valores: 0, 1, … (número máx. de locais na tabela - 1)

Exemplo: Aproximação de posições codificadas de um eixo de posicionamento

Código de programação	**Comentário**
`N10 FA[B]=300`	`; Avanço para eixo de posicionamento B`
`N20 POS[B]=CAC(10)`	`; Aproximar a posição codificada de número de posição 10`
`N30 POS[B]=CIC(-4)`	`; Aproximar a posição codificada do "número atual de posição" - 4`

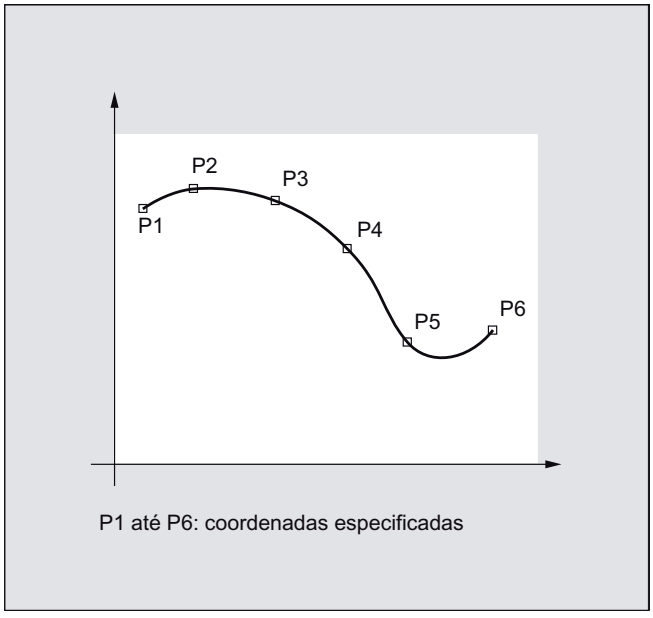
Literatura

- Manual de funções ampliadas; Eixos de indexação (T1)
- Manual de funcionamento para ações sincronizadas

5.2 Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Muitos contornos curvados nas peças de trabalho não podem ser descritos analiticamente com precisão. Por isso que os contornos desse tipo são aproximados através de um número limitado de pontos de interpolação, como no caso da digitalização de superfícies. Para gerar a superfície digitalizada de uma peça de trabalho os pontos de interpolação devem ser ligados em uma descrição de contorno. Isso permite a interpolação de Spline.

Uma Spline define uma curva formada por polinômios de 2º e 3º grau. As propriedades nos pontos de interpolação de uma Splines podem ser definidas **em função do tipo de Spline utilizado**.



P1 até P6: coordenadas especificadas

No SINUMERIK solution line estão disponíveis os seguintes tipos de Spline:

- A-Spline
- B-Spline
- C-Spline

Sintaxe

Geral:

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

Na B-Spline também se programa:

```
PW=<n>
SD=2
PL=<valor>
```

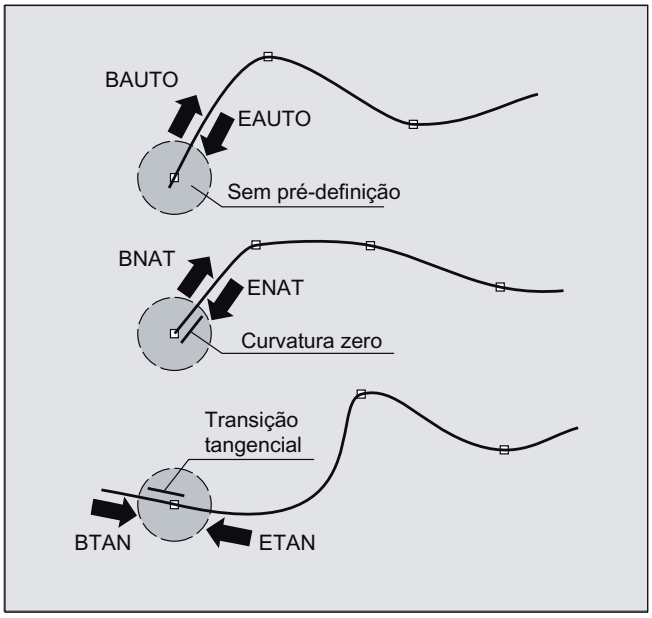
Na A-Spline e C-Spline também se programa:

```
BAUTO / BNAT / BTAN
```

EAUTO / ENAT / ETAN

Significado

Tipo de interpolação de Spline:			
ASPLINE:	Comando para ativar a interpolação de A-Spline		
BSPLINE:	Comando para ativar a interpolação de B-Spline		
CSPLINE:	Comando para ativar a interpolação de C-Spline		
	Os comandos ASPLINE, BSPLINE e CSPLINE são ativados modalmente e pertencem ao grupo dos comandos de curso.		
Pontos de interpolação e pontos de controle:			
X... Y... Z... A... B... C...	Posições em coordenadas cartesianas		
Peso de ponto (apenas para B-Spline):			
PW:	Com o comando PW é possível ser programado um chamado "Peso de ponto" para cada ponto de interpolação.		
<n>:	"Peso de ponto"		
	Faixa de valores:	0 ≤ n ≤ 3	
	Incremento:	0.0001	
	Efeito:	n > 1	A curva é guiada de forma mais acentuada pelo ponto de controle.
		n < 1	A curva é guiada de forma menos acentuada pelo ponto de controle.
Grau Spline (apenas para B-Spline):			
SD:	Como padrão é usado um polinômio de 3º grau. Através da programação do SD=2, mas também pode ser utilizado um polinômio de 2º grau.		
Distância dos nós (apenas para B-Spline):			
PL:	As distâncias entre os nós são calculadas internamente de forma adequada. O comando também pode processar as distâncias indicadas dos nós, que são indicadas como distância parâmetro-intervalo com o comando PL.		
<valor>:	Comprimento de intervalo do parâmetro		
	Faixa de valores:	como dimensão de curso	
Comportamento de transição no início da curva Spline (apenas para A-Spline ou C-Spline):			
BAUTO:	Nenhuma especificação para comportamento de transição. O início resulta da posição do primeiro ponto.		
BNAT:	Curvatura zero		
BTAN:	Transição tangencial para o bloco anterior (ajuste inicial)		
Comportamento de transição no fim da curva Spline (apenas para A-Spline ou C-Spline):			
EAUTO:	Nenhuma especificação para comportamento de transição. O fim resulta da posição do último ponto.		

ENAT:	Curvatura zero
ETAN:	Transição tangencial para o bloco anterior (ajuste inicial)
	

Indicação

O comportamento de transição programável não exerce nenhuma influência sobre a B-Spline. A B-Spline sempre é tangencial ao polígono de controle no ponto de partida e ponto final.

Condições gerais

- A correção do raio da ferramenta é aplicável.
- A monitoração de colisões é realizada na projeção sobre o plano.

Exemplos

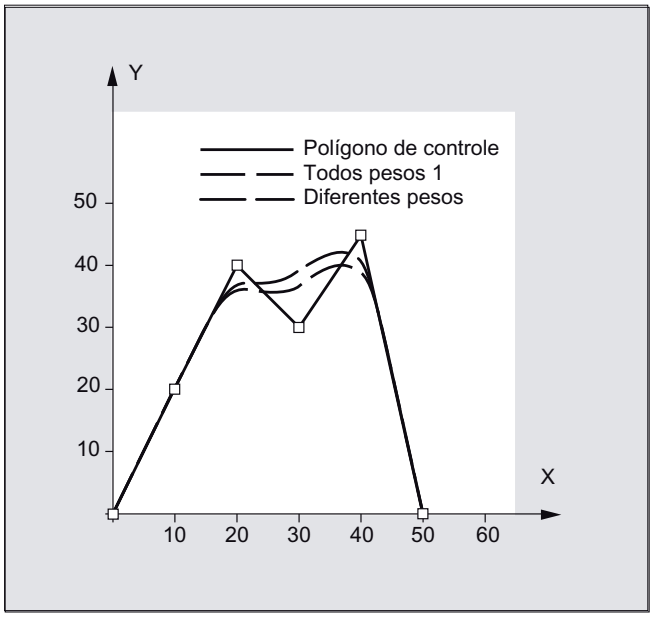
Exemplo 1: B-Spline

```
Código de programa 1 (todos os pesos 1)
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

Código de programa 2 (diferentes pesos)

```
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
```

Código de programa 3 (polígono de controle)	**Comentário**
N10 G1 X0 Y0 F300 G64	
N20	; suprimido
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	

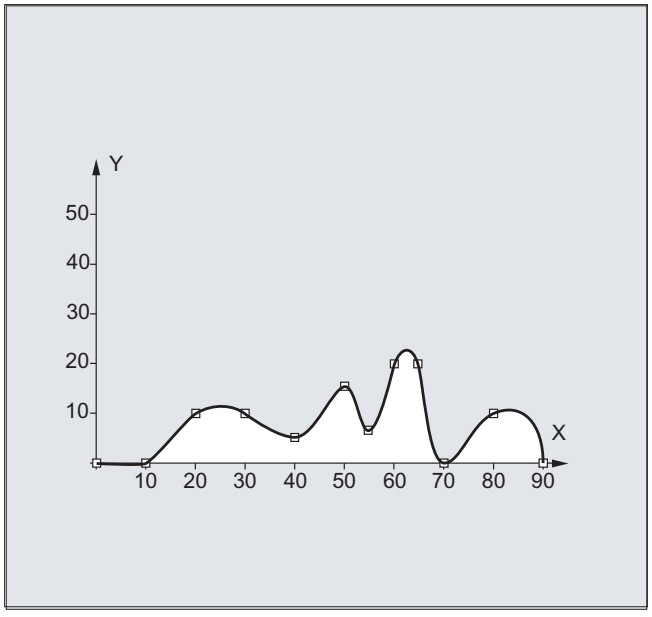


Exemplo 2: C-Spline, no início e no fim da curvatura zero

Código de programa

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
```

```
Código de programa
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
```



Exemplo 3: Interpolação de Spline (A-Spline) e transformação de coordenadas (ROT)

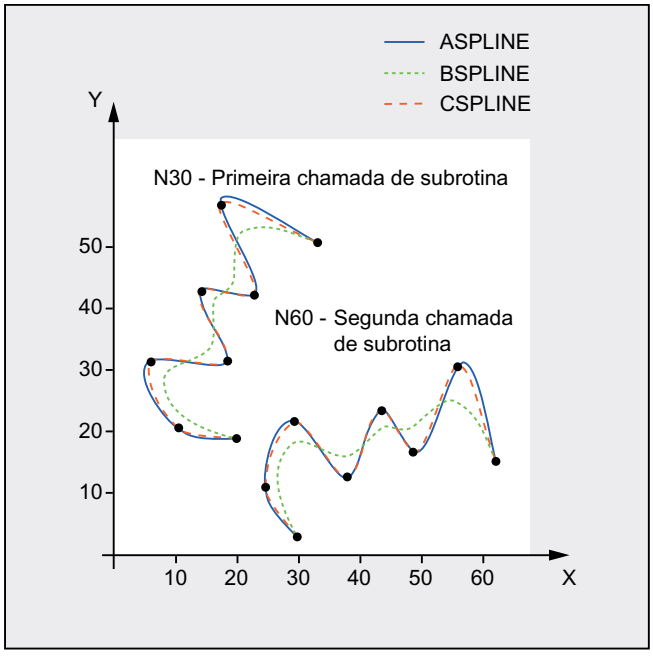
Programa principal:

Código de programa	Comentário
N10 G00 X20 Y18 F300 G64	; Aproximação do ponto de partida.
N20 ASPLINE	; Ativar o tipo de interpolação A-Spline.
N30 CONTORNO	; Primeiro acesso da sub-rotina.
N40 ROT Z-45	; Transformação de coordenadas: Rotação do WCS em -45° em torno do eixo Z.
N50 G00 X20 Y18	; Aproximar do ponto de partida do contorno.
N60 CONTORNO	; Segunda chamada da sub-rotina.
N70 M30	; fim do programa

Sub-rotina "Contorno" (contém as coordenadas dos pontos de interpolação):

```
Código de programa
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
```

Na figura a seguir, além da curva Spline, resultante do exemplo de programa (ASPLINE), também estão contidas as curvas Spline que resultariam da ativação de uma interpolação B--Spline ou C-Spline (BSPLINE, CSPLINE):



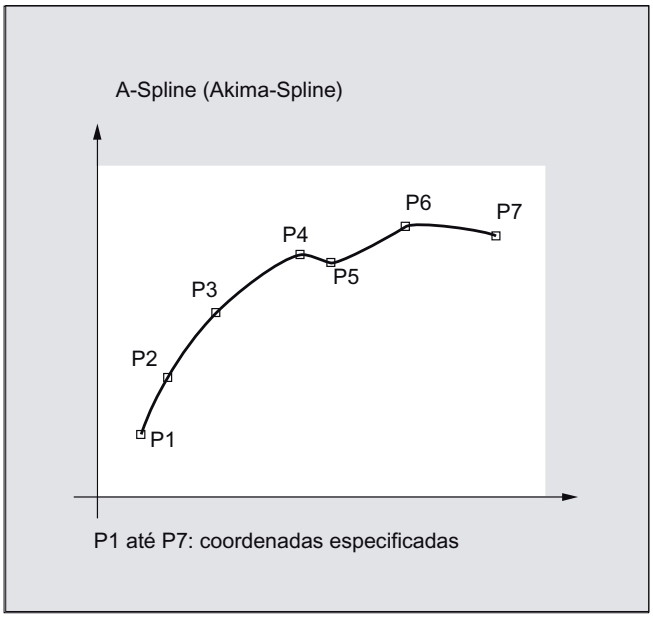
Outras informações

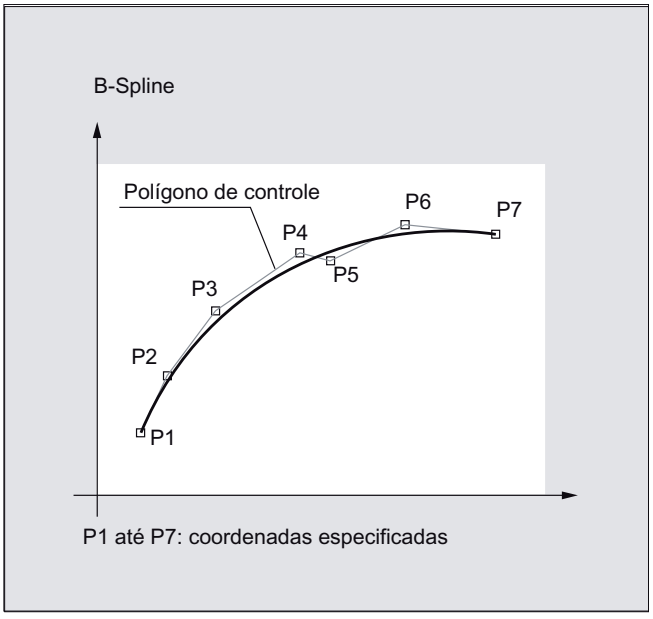
Vantagens da interpolação de Spline

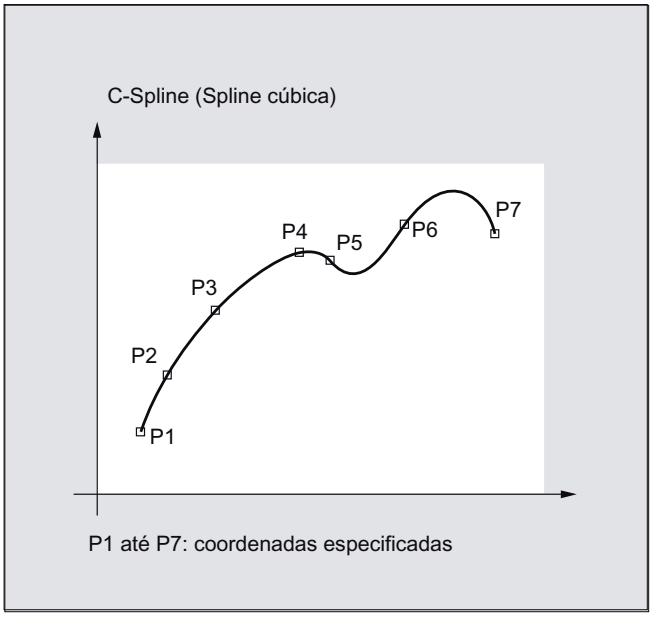
Através do uso da interpolação de Spline são obtidas as vantagens mencionadas a seguir, em contrapartida ao uso de blocos de retas `G01`:

- Redução do número de blocos de programa de peça necessários para descrição do contorno
- Trajetória suave de curvas e de melhor aspecto estético nas transições entre os blocos de programa de peça

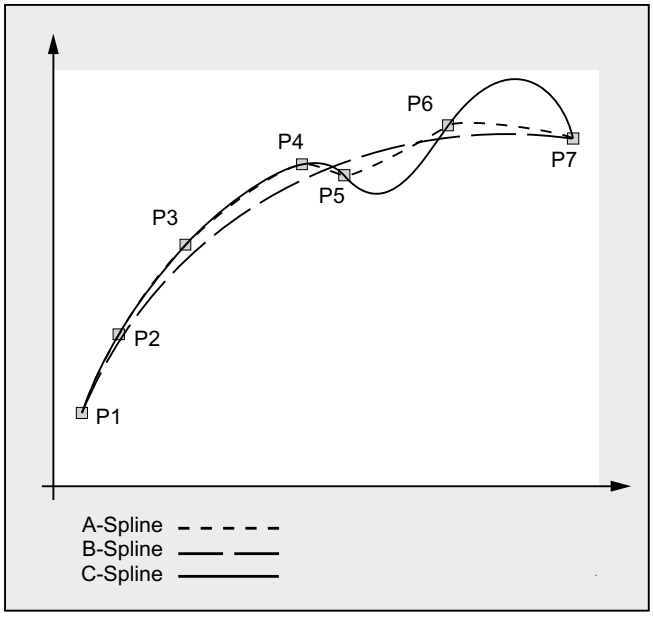
Propriedades e aplicação dos diversos tipos de Spline

Tipo de Spline	Propriedades e aplicação
A-Spline	 **Propriedades:** • Percorre exatamente pelos pontos de interpolação especificados. • A trajetória da curva é tangencial, mas nem sempre curvada. • Quase não gera nenhuma ondulação indesejada. • A área de influência de alterações de pontos de interpolação é local, isto é, a alteração de um ponto de interpolação somente tem efeito até 6 pontos de interpolação vizinhos. **Aplicação:** A A-Spline é principalmente indicada para interpolação de trajetórias de curvas com grandes variações de inclinação (p. ex. trajetórias de curvas em forma de escada).

Tipo de Spline	Propriedades e aplicação
B-Spline	 **Propriedades:** • Não percorre pelos pontos de interpolação especificados, mas apenas na proximidade dos mesmos. As curvas são direcionadas pelos pontos de interpolação. Também se pode controlar a trajetória de curvas através da avaliação de peso dos pontos de interpolação com um fator. • A trajetória de curvas é tangencial e sempre curvada. • Não gera nenhuma ondulação indesejada. • A área de influência de alterações de pontos de interpolação é local, isto é, a alteração de um ponto de interpolação somente tem efeito até 6 pontos de interpolação vizinhos. **Aplicação:** A princípio o B-Spline é indicado como interface para sistemas CAD.

Tipo de Spline	Propriedades e aplicação
C-Spline	 **Propriedades:** • Percorre exatamente pelos pontos de interpolação especificados. • A trajetória de curvas é tangencial e sempre curvada. • Gera ondulações indesejadas com frequência, principalmente em pontos com grandes variações de inclinação. • A área de influência das variações de pontos de interpolação é global, isto é, a variação de um ponto de interpolação atua pela trajetória inteira da curva. **Aplicação:** O C-Spline pode ser bem empregado se os pontos de interpolação estiverem em uma curva analítica conhecida (círculo, parábola, hipérbole)

Comparação dos três tipos de Spline usando os mesmos pontos de interpolação



Número mínimo de blocos de Spline

Os comandos G `ASPLINE`, `BSPLINE` e `CSPLINE` ligam os pontos de fim de bloco com Splines. Para isso, antecipadamente, deve ser calculada simultaneamente uma série de blocos (pontos finais). O tamanho padrão do buffer para os cálculos é de 10 blocos. Nem toda informação de bloco é um ponto final de Spline. Entretanto, o comando requer de 10 blocos uma determinada quantidade de blocos de ponto final Spline:

Tipo de Spline	Número mínimo de blocos de Spline
A-Spline:	De cada 10 blocos, no mínimo **4** devem ser blocos Spline. Os blocos de comentário e cálculos de parâmetros não são considerados neste caso.
B-Spline:	De cada 10 blocos, no mínimo **6** devem ser blocos Spline. Os blocos de comentário e cálculos de parâmetros não são considerados neste caso.
C-Spline:	A quantidade mínima requerida em blocos Spline resulta da seguinte somatória: Valor do MD20160 $MC_CUBIC_SPLINE_BLOCKS + 1 No MD20160 se especifica a quantidade de pontos através dos quais é calculado o segmento de Spline. O ajuste padrão é 8. De cada 10 blocos, em caso padrão, no mínimo **9** devem ser blocos Spline.

Indicação

Se o valor ficar abaixo do tolerável será emitido um alarme, e o mesmo ocorre se um eixo envolvido com a Spline for programado como eixo de posicionamento.

Agrupamento de blocos Spline curtos

Na interpolação de Spline podem existir blocos Spline curtos, que causam uma redução desnecessária da velocidade de percurso. Com a função "Agrupamento de blocos Spline curtos" estes blocos podem ser agrupados de modo que o tamanho resultante dos blocos seja suficiente e que não resulte em uma redução da velocidade de percurso.

A função é ativada através do dado de máquina específico de canal:

MD20488 $MC_SPLINE_MODE (ajuste da interpolação de Spline)

Literatura:
Manual de funcionamento das funções básicas; Modo de controle da trajetória, Parada exata, LookAhead (B1),
Capítulo: Agrupamento de blocos Spline curtos

5.3 Agrupamento de Spline (SPLINEPATH)

Os eixos que são interpolados no agrupamento de Spline são selecionados com o comando `SPLINEPATH`. A interpolação de Splines é possível até com oito eixos.

Indicação

Se o `SPLINEPATH` não for programado de forma explícita, então são deslocados os primeiros três eixos do canal do agrupamento de Spline.

Sintaxe

A definição do agrupamento de Spline é realizado em um bloco especial:

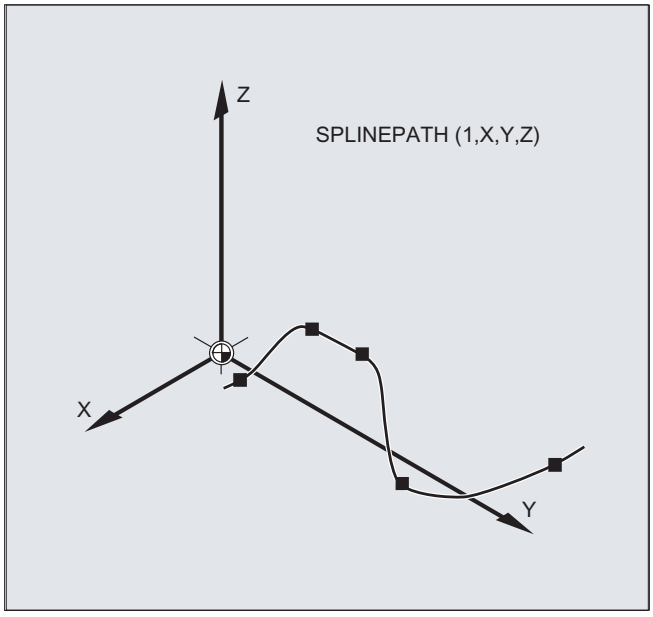
```
SPLINEPATH(n,X,Y,Z,…)
```

Significado

`SPLINEPATH`:	Comando para definição de um agrupamento de Spline
`n`:	=1 (valor fixo)
`X,Y,Z,…` :	Identificador dos eixos de trajetória interpolados no agrupamento de Spline

Exemplo: Agrupamento de Spline com três eixos de percurso

Código de programa	**Comentário**
`N10 G1 X10 Y20 Z30 A40 B50 F350`	
`N11 SPLINEPATH(1,X,Y,Z)`	`; Agrupamento Spline`
`N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60`	`; C-Spline`
`N14 X30 Y40 Z50 A60 B70`	`; Pontos de apoio`
…	
`N100 G1 X… Y…`	`; Cancelamento da seleção de interpolação Spline`


Z
SPLINEPATH (1,X,Y,Z)
X
Y

5.4 Ligar/desligar a compressão de blocos NC (COMPON, COMPCURV, COMPCAD, COMPSURF, COMPOF)

As funções para a compressão de blocos lineares (e em função da parametrização também de blocos circulares e/ou de processamento rápido) são ligadas/desligadas com os comandos G do grupo G 30. Os comandos estão ativos de forma modal.

Sintaxe

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

Significado

COMPON:	Ligar a função de compressor COMPON
COMPCURV:	Ligar a função de compressor COMPCURV
COMPCAD:	Ligar a função de compressor COMPCAD
COMPSURF:	Ligar a função de compressor COMPSURF
COMPOF:	Desligar a função de compressor atualmente ativa

Indicação

Para uma melhoria adicional da qualidade superficial pode ser usada a função de suavização G642 e a limitação de solavancos SOFT. Estes comandos devem ser escritos no início do programa.

Exemplo: COMPCAD

```
Código de programa          Comentário
N10 G00 X30 Y6 Z40
N20 G1 F10000 G642          ; Ligar: Função de suavização G642
N30 SOFT                    ; Ligar: Limitação de solavancos SOFT
N40 COMPCAD                 ; Ligar: Função de compressor COMPCAD
N50 STOPFIFO
N24050 Z32.499              ; 1. Bloco de deslocamento
N24051 X41.365 Z32.500      ; 2. Bloco de deslocamento
...
N99999 X... Z...            ; Último bloco de deslocamento
COMPOF                      ; Função de compressor desligada.
...
```

5.5 Interpolação de polinômios (POLY, POLYPATH, PO, PL)

Na verdade trata-se de uma interpolação de polinômios (POLY) e não de um tipo de interpolação de Splines. Em primeira linha ela age como interface para a programação de curvas Spline criadas externamente. Aqui os segmentos Spline podem ser programados diretamente.

Este tipo de interpolação alivia o NC dos cálculos dos coeficientes de polinômio. Ela pode perfeitamente ser aplicada se os coeficientes são diretamente fornecidos de um sistema CAD ou pós-processador.

Sintaxe

Polinômio de 3º grau:
```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

Polinômio de 5º grau e nova sintaxe de polinômio:
```
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")
```

Indicação

A soma dos coeficientes de polinômio e eixos programados em um bloco NC não pode exceder o número máximo permitido de eixos por bloco.

Significado

POLY:	Ativação da interpolação de polinômios com um bloco com POLY.
POLYPATH:	A interpolação de polinômios pode ser selecionada para os dois grupos de eixos, AXIS ou VECT
PO[identificador de eixo/variável]:	Pontos finais e coeficientes de polinômio
X, Y, Z:	Identificador de eixo
xe, ye, ze :	Especificação da posição final do respectivo eixo; faixa de valores como dimensão de curso
a2, a3, a4, a5:	Os coeficientes a_2, a_3, a_4, e a_5 são escritos com seu respectivo valor; faixa de valores como na dimensão de curso. O último coeficiente em cada caso pode ser omitido se o valor for zero.
PL:	Comprimento do intervalo de parâmetro no qual está definido o polinômio (faixa de definição da função f(p)). O intervalo sempre inicia em 0, p pode aceitar valores de 0 até PL. Faixa de valores teórica para PL: 0,0001 … 99 999,9999 **Nota:** O valor PL vale para o bloco em que ele se encontra. Se não foi programado PL, então PL=1.

Ativação e desativação da interpolação de polinômios

A interpolação de polinômios é ativada no programa de peça através do comando G `POLY`.

O comando G `POLY` pertence ao 1º grupo G, junto com o `G0`, `G1`, `G2`, `G3`, `ASPLINE`, `BSPLINE` e `CSPLINE`.

Os eixos que foram programados apenas com nomes e ponto final (p. ex. `X10`) são deslocados linearmente. Se todos os eixos de um bloco NC foram programados desta forma, o comando se comporta como no caso do G1.

De modo implícito a interpolação de polinômios é novamente desativada através da programação de outro comando do 1º grupo G (p. ex. `G0`, `G1`).

Coeficiente de polinômio

Os valores `PO (PO[]=)` e `...=PO(...)` especifica todos os coeficientes de polinômio para um eixo. Dependendo do grau do polinômio mais valores são especificados separados por vírgula. Dentro de um bloco são possíveis diferentes graus de polinômio para diferentes eixos.

Sub-rotina POLYPATH

Com `POLYPATH(...)` é possível liberar a interpolação de polinômios de modo seletivo para determinados grupos de eixos

Apenas os eixos de percurso e eixos adicionais:	`POLYPATH("AXES")`
Somente eixos de orientação: (no processo com transformação de orientação)	`POLYPATH ("VECT")`

Os eixos que não forem liberados são deslocados linearmente.

Como padrão a interpolação de polinômios é liberada para os dois grupos de eixos.

Através da programação sem indicar o parâmetro `POLYPATH( )` desativa-se a interpolação de polinômios para todos os eixos.

Exemplo

```
Código de programa                                          Comentário
N10 G1 X… Y… Z… F600
N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2)                ; Interpolação de polinômios ati-
PL=1.5                                                      vada
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                             ; Especificações mistas para os
                                                            eixos
N27 PO[X]=(10,2.5) PO[Y]=(2.3)                              ; Sem PL programado; atuando PL=1
N30 G1 X… Y… Z.                                             ; Interpolação de polinômios desa-
                                                            tivada
…
```

Exemplo: Nova sintaxe de polinômio

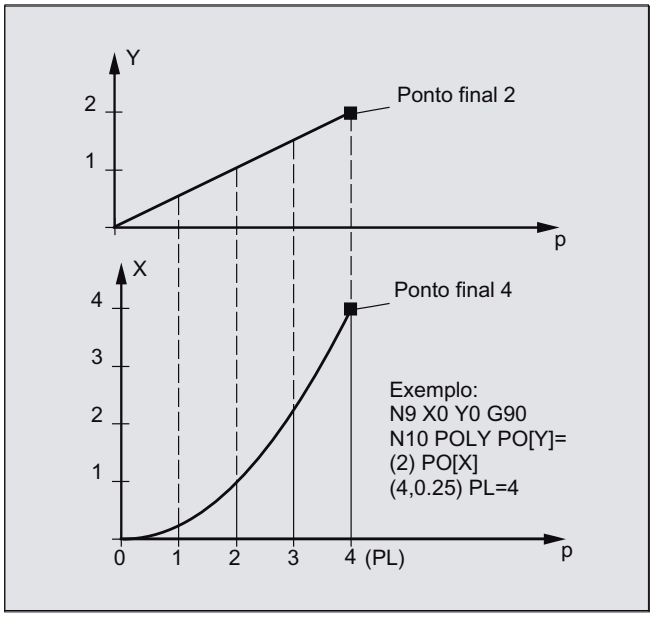
Sintaxe de polinômio ainda aplicada	Nova sintaxe de polinômio
`PO[identificador de eixo]=(.. , ..)`	`Identificador de eixo=PO(.. , ..)`
`PO[PHI]=(.. , ..)`	`PHI=PO(.. , ..)`
`PO[PSI]=(.. , ..)`	`PSI=PO(.. , ..)`
`PO[THT]=(.. , ..)`	`THT=PO(.. , ..)`
`PO[]=(.. , ..)`	`PO(.. , ..)`
`PO[variável]=IC(.. , ..)`	`variável=PO IC(.. , ..)`

Exemplo: Curva no plano X/Y

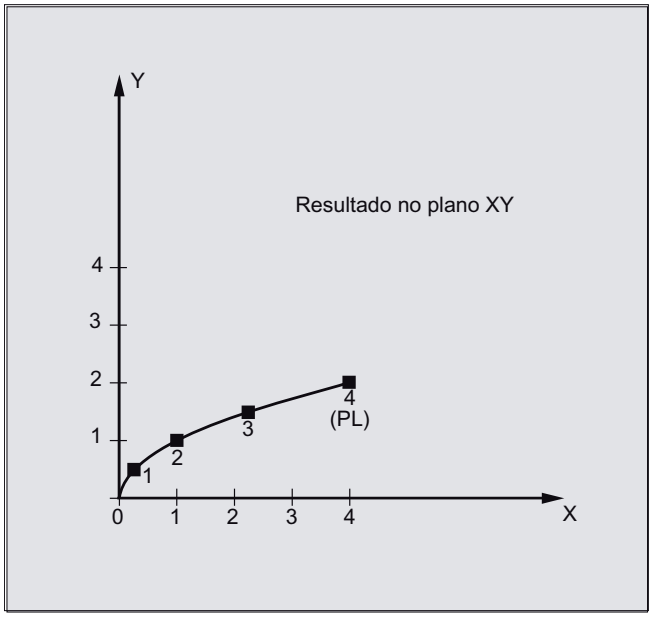
Programação

Código de programa
`N9 X0 Y0 G90 F100`
`N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4`

Trajetória das curvas X(p) e Y(p)



Trajetória da curva no plano XY



Descrição

A equação geral da função de polinômio é:

$f(p)= a_0 + a_1p + a_2p^2 + . . . + a_np^n$

com: a_i: coeficientes constantes (i = 0, 1, ..., n)
p: Parâmetro

No sistema de comando, é possível programar polinômios de 5º grau, no máximo:

$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$

Atribuindo valores concretos aos coeficientes, diferentes curvaturas assim como retas, parábolas e funções exponenciais podem ser criadas.

Uma reta é gerada através de $a_2 = a_3 = a_4 = a_5 = 0$:

$f(p) = a_0 + a_1p$

Ainda se aplica:

$a_{0:}$ Posição de eixo no fim do bloco precedente

p = PL

$a_1 = (x_E - a_0 - a_2*p^2 - a_3*p^3) / p$

É possível programar polinômios **sem** que a interpolação de polinômios seja ativada através do comando G `POLY`. Porém, neste caso os polinômios programados não serão interpolados, apenas os pontos finais programados dos eixos serão aproximados linearmente (`G1`). Somente depois de uma ativação explícita da interpolação de polinômios no programa de peça (`POLY`) é que os polinômios programados serão executados como tais.

Particularidade: Polinômio de denominador

Para eixos geométricos também se pode programar um polinômio de denominador comum sem especificar um nome de eixo com `PO[]=(…)`, isto é, se interpola o movimento dos eixos geométricos quociente de dois polinômios.

Com isso é possível representar de forma exata segmentos cônicos (circunferência, elipse, parábola, hipérbole).

Exemplo:

Código de programa	**Comentário**
`POLY G90 X10 Y0 F100`	`; Eixos geométricos deslocam-se linear-mente até a posição X10 Y0.`
`PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)`	`; Eixos geométricos deslocam-se em qua-drante até X0 Y10.`

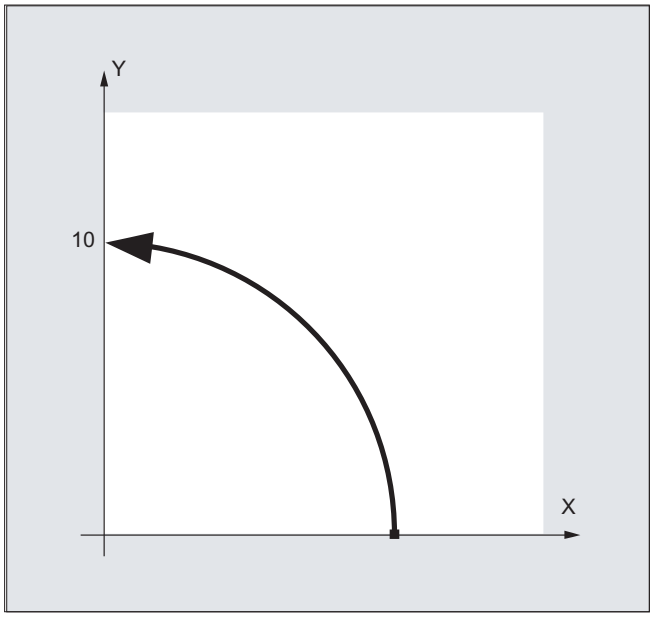
O coeficiente constante (a_0) do polinômio de denominador sempre é adotado com 1. O ponto final programado independe do `G90` / `G91`.

A partir dos valores programados o X(p) e U(p) são calculados como:

$X(p) = (10 - 10 * p^2) / (1 + p^2)$
$Y(p) = 20 * p / (1 + p^2)$
com $0 \le p \le 1$

Com base na programação dos pontos iniciais, pontos finais, coeficiente a_2 e PL=1 temos os seguintes resultados intermediários:

Numerador (X) $=10 + 0 * p - 10 * p^2$
Numerador (Y) = $0 + 20 * p + 0 * p^2$
Denominador = $1 + p^2$



É emitido um alarme se for programado um polinômio de denominador com zeros dentro do intervalo `[0,PL]`. O polinômio de denominador não tem nenhum efeito sobre o movimento de eixos adicionais.

Indicação

Uma correção do raio de compensação pode ser ativada com `G41` e `G42` em conjunto com a interpolação de polinômios, depois ser aplicada para interpolação linear ou circular.

5.6 Referência ajustável do percurso (SPATH, UPATH)

No caso da interpolação de polinômios (POLY, ASPLINE, BSPLINE, CSPLINE, COMPON, COMPCURV) as posições dos eixos de percurso i são prescritas pelos polinômios $p_i(U)$. O parâmetro de curva U desloca-se, nesta ocasião, no interior de um bloco NC de 0 a 1.

Através de FGROUP são selecionados os eixos (eixos FGROUP), aos quais se refere o avanço de percurso F. Uma interpolação com velocidade constante no percurso S dos eixos FGROUP significa, durante a interpolação de polinômios, entretanto, geralmente uma alteração não constante do parâmetro de curva U. Para os eixos não inclusos no FGROUP, pode ser selecionado deste modo, entre duas possibilidades, como estes devem seguir os eixos FGROUP:

- De modo síncrono ao percurso S (SPATH)
- De modo síncrono ao parâmetro de curva U (UPATH)

Sintaxe

```
SPATH
UPATH
```

Significado

SPATH:	Os eixos não inclusos no FGROUP são deslocados com relação ao percurso S
UPATH:	Os eixos não inclusos no FGROUP são deslocados com relação ao parâmetro de curva U

Indicação

UPATH e SPATH também determinam a relação do polinômio de palavra F (FPOLY, FCUB, FLIN) com o movimento de percurso.

Condições gerais

SPATH ou UPATH não apresentam nenhum significado no caso de:

- Interpolação linear (G1)
- Interpolação circular (G2, G3)
- Blocos de roscas (G33, G34, G35, G33x, G63)
- Todos os eixos de percurso estão inclusos em FGROUP

Exemplo

O exemplo a seguir mostra a diferença entre os dois tipos de controle de movimento.

Código de programa

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 SPATH                ; SPATH
```

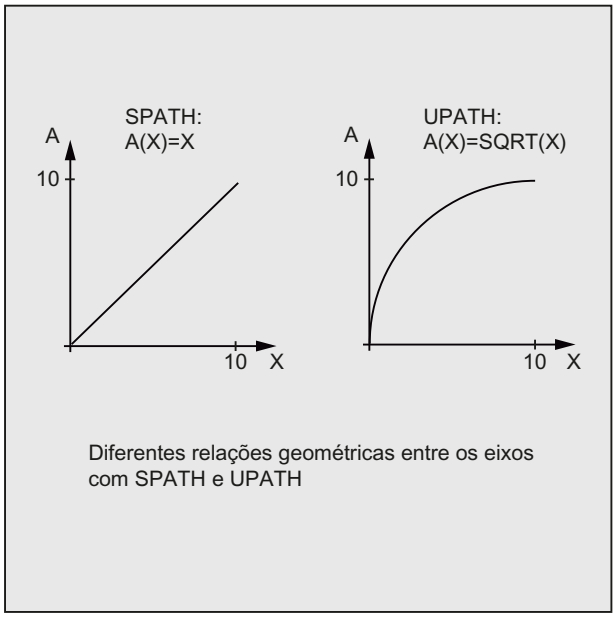
Código de programa

```
N20 POLY PO[X]=(10,10) A10
```

Código de programa

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH                ; UPATH
N20 POLY PO[X]=(10,10) A10
```

Em ambos os segmentos de programa, em `N20` o curso S dos eixos FGROUP depende do quadrado do parâmetro de curva U. Por isto que resultam X diferentes posições do eixo sincronizado A ao longo do curso, dependendo se `SPATH` ou `UPATH` está ativo.



Diferentes relações geométricas entre os eixos com SPATH e UPATH

Outras informações

Comportamento de comando com Reset e dados de máquina e dados de opcionais

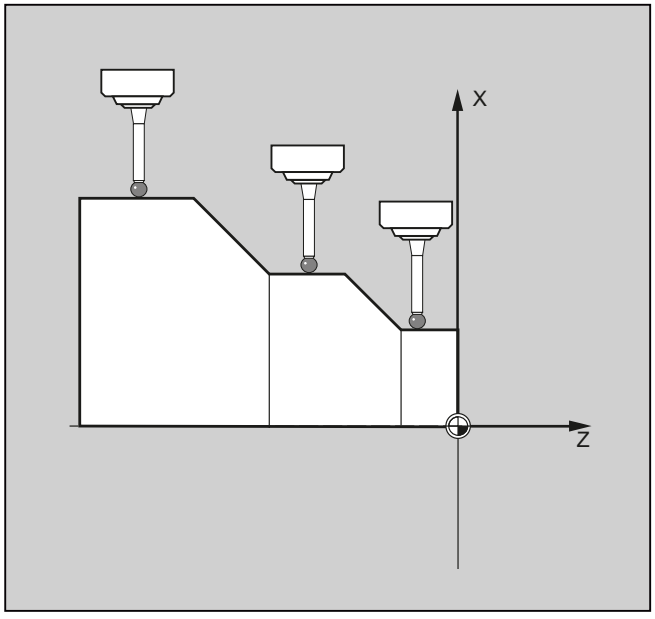
Após o Reset, o comando G determinado pelo MD20150 $MC_GCODE_RESET_VALUES[44] está ativo(45º grupo de G).

O valor de ajuste básico para o tipo de suavização é definido com o MD20150 $MC_GCODE_RESET_VALUES[9] (10º grupo de G).

O dado da máquina específico de eixo MD33100 $MA_COMPRESS_POS_TOL[<n>] possui um significado mais amplo: eles contém as tolerâncias para a função de compressor e para a suavização com G642.

5.7 Medição com apalpador comutável (MEAS, MEAW)

Com a função "Medição com apalpador comutável" as posições reais na peça de trabalho são aproximadas e na borda de contato do apalpador de medição é realizada a medição das posições para todos os eixos programados no bloco de medição e depois realizada a gravação dos valores para cada eixo na célula de memória correspondente.



Para a programação da função estão disponíveis ambos endereços fixos:

- `MEAS`
  Com `MEAS` é cancelado o curso restante entre a posição real e a nominal.
- `MEAW`
  `MEAW` é empregado em tarefas de medição onde a posição programada deve ser aproximada em todas as ocasiões.

O `MEAS` e o `MEAW` estão ativos por blocos e são programados juntos com instruções de movimento. Os avanços e tipos de interpolação (G0, G1, ...), assim como o número de eixos, devem ser adaptados ao respectivo problema de medição.

Sintaxe

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

Significado

`MEAS`:	Comando: Medição **com** anulação de curso restante	
	Efeito:	por blocos
`MEAW`:	Comando: Medição **sem** anulação de curso restante	
	Efeito:	por blocos

<TE>:	Evento de disparo para partida da medição		
	Tipo:	INT	
	Faixa de valores:	-2, -1, 1, 2	
	Significado:		
	(+)1	Flancos crescentes do apalpador de medição 1 (na entrada de medição 1)	
	-1	Flancos decrescentes do apalpador de medição 1 (na entrada de medição 1)	
	(+)2	Flancos crescentes do apalpador de medição 2 (na entrada de medição 2)	
	-2	Flancos decrescentes do apalpador de medição 2 (na entrada de medição 2)	
	Nota: Existem no máximo 2 apalpadores de medição (dependendo do nível de expansão).		
G...:	Tipo de interpolação, p. ex. G0, G1, G2 ou G3		
X... Y... Z...:	Ponto final em coordenadas cartesianas		

Exemplo

Código de programa	**Comentário**
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; Bloco de medição com apalpador da primeira entrada de medição e interpolação linear. A parada de pré-processamento é gerada automaticamente.
...	

Outras informações

Estado do pedido de medição

Se no programa for necessária uma avaliação para saber se o sensor de medição realmente está ativo, então se pode consultar a variável de estado $AC_MEA[<n>] (<n> = Numero do sensor):

Valor	Significado
0	Tarefa de medição não preenchida
1	Tarefa de medição concluída com sucesso (o apalpador de medição foi acionado)

Indicação

Quando o apalpador de medição for defletido, no programa, a variável passa para 1. Na partida de um bloco de medição a variável é automaticamente passada para o estado inicial do apalpador.

Coleta de valores de medição

São coletadas as posições de todos eixos de percurso e de posicionamento do bloco (número máximo de eixos está em função da configuração do comando). Com o `MEAS` o movimento é desacelerado controladamente depois de ser ativado o apalpador de medição.

Indicação

Se em um bloco de medição foi programado um eixo geométrico, então são armazenados os valores de medição dos atuais eixos geométricos.

Se em um bloco de medição foi programado um eixo envolvido em uma transformação, então são armazenados os valores de medição de todos eixos envolvidos com esta transformação.

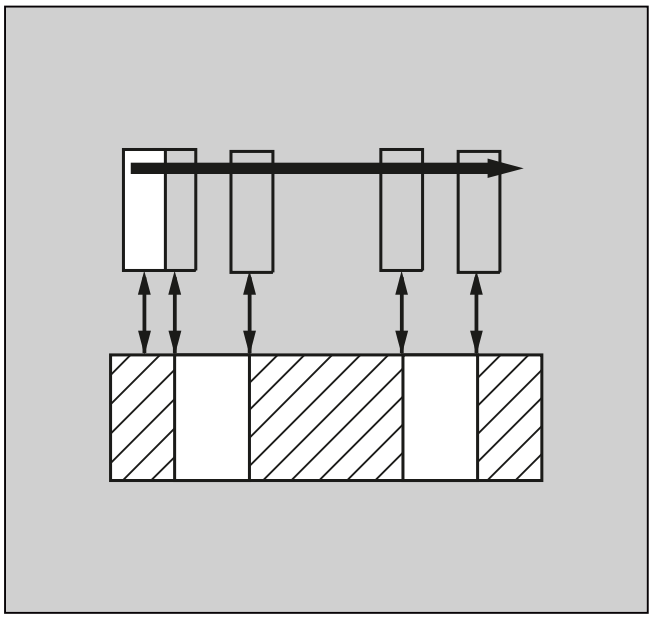
Leitura dos resultados de medição

Os resultados da medição para os eixos analisados com sensor de medição podem ser lidos através das seguintes variáveis de sistema:

- $AA_MM[<eixo>]
  Resultados de medição no sistema de coordenadas da máquina
- $AA_MW[<eixo>]
  Resultados de medição no sistema de coordenadas da peça de trabalho

5.8 Medição específica de eixos (MEASA, MEAWA, MEAC) (Opção)

Na medição específica de eixos podem ser usados vários apalpadores de medição e vários sistemas de medição.



Para a programação da função estão disponíveis palavras-chave MEASA, MEAWE e MEAC.

Com MEASA ou MEAWA são coletados até quatro valores de medição para o eixo programado em cada medição e estes são armazenados em variáveis de sistema conforme o evento de disparo.

Pedidos contínuos de medição podem ser realizados com MEAC. Neste caso os resultados das medições são armazenados em variáveis FIFO.

Sintaxe

```
MEASA[<eixo>]=(<modo>,<TE1>,...,<TE4>)
MEAWA[<eixo>]=(<modo>,<TE1>,...,<TE4>)
MEAC[<eixo>]=(<modo>,<memória de medição>,<TE1>,...,<TE4>)
```

Indicação

MEASA e MEAWA estão ativos por blocos e podem ser programados juntos em um bloco. Se MEASA/MEAWA for programado junto com MEAS/MEAW em um bloco, será gerada uma mensagem de erro.

Significado

MEASA:	Palavra-chave: Medição específica de eixos com eliminação de curso restante	
	Efeito:	por blocos

MEAWA:	Palavra-chave: Medição específica de eixos sem eliminação de curso restante	
	Efeito:	por blocos
MEAC:	Palavra-chave: Medição contínua, específica de eixos sem eliminação de curso restante	
	Efeito:	por blocos
<eixo>:	Nome do eixo de canal utilizado na medição	
<modo>:	Número de dois dígitos para especificação do modo de operação (modo de medição e sistema de medição)	
	década única (Modo de medição):	
	0	Cancelar pedido de medição.
	1	Até 4 eventos de disparo diferentes ativados simultaneamente.
	2	Até 4 eventos de disparo ativados sucessivamente.
	3	Até 4 eventos de disparo ativados consecutivamente, mas sem monitoração do evento de disparo 1 no START (os alarmes 21700/21703 são suprimidos). **Nota:** Este modo não é possível com o MEAC.
	década decimal (sistema de medição):	
	0 (ou sem indicação)	sistema de medição ativo
	1	Sistema de medição 1
	2	Sistema de medição 2
	3	ambos sistemas de medição
<TE>:	Evento de disparo para partida da medição	
	Tipo:	INT
	Faixa de valores:	-2, -1, 1, 2
	Significado:	
	(+)1	Flancos crescentes do apalpador de medição 1
	-1	Flancos decrescentes do apalpador de medição 1
	(+)2	Flancos crescentes do apalpador de medição 2
	-2	Flancos decrescentes do apalpador de medição 2
<Memória de medição>:	Número do FIFO (memória de circulação)	

Exemplos

Exemplo 1: Medição específica de eixos com eliminação de curso restante no modo 1 (avaliação em sequência cronológica)

a) com 1 sistema de medição

Código de programa	Comentário
...	
N100 MEASA[X]=(1,1,-1) G01 X100 F100	; Medição em modo 1 com sistema de medição ativo. Esperar pelo sinal de medição com flanco crescente/decrescente do apalpador de medição 1 no percurso até X=100.

Código de programa	Comentário
N110 IF $AC_MEA[1]==FALSE GOTOF FIM	; Controlar o sucesso da medição.
N120 R10=$AA_MM1[X]	; Salvar o valor de medição correspondente ao primeiro evento de disparo (flanco crescente) programado.
N130 R11=$AA_MM2[X]	; Salvar o valor de medição correspondente ao segundo evento de disparo (flanco decrescente) programado.
N140 FIM:	

b) com 2 sistemas de medição

Código de programa	Comentário
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; Medição no modo 1 com ambos os sistemas de medição. Esperar pelo sinal de medição com flanco crescente/decrescente do apalpador de medição 1 no percurso até X=100.
N210 IF $AC_MEA[1]==FALSE GOTOF FIM	; Controlar o sucesso da medição.
N220 R10=$AA_MM1[X]	; Salvar o valor de medição do sistema de medição 1 com flanco crescente.
N230 R11=$AA_MM2[X]	; Salvar o valor de medição do sistema de medição 2 com flanco crescente.
N240 R12=$AA_MM3[X]	; Salvar o valor de medição do sistema de medição 1 com flanco decrescente.
N250 R13=$AA_MM4[X]	; Salvar o valor de medição do sistema de medição 2 com flanco decrescente.
N260 FIM:	

Exemplo 2: Medição específica de eixos com eliminação de curso restante no modo 2 (avaliação em sequência programada)

Código de programa	Comentário
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; Medição em modo 2 com sistema de medição ativo. Esperar pelo sinal de medição na ordem dos flancos crescentes do apalpador de medição 1, flancos decrescentes do apalpador de medição 1, flancos crescentes do apalpador de medição 2, flancos decrescentes do apalpador de medição 2 no percurso até X=100.
N110 IF $AC_MEA[1]==FALSE GOTOF APALPADOR2	; Controlar o sucesso da medição com o apalpador de medição 1.
N120 R10=$AA_MM1[X]	; Salvar o valor de medição correspondente ao primeiro evento de disparo (flanco crescente do sensor 1) programado.

Código de programa	Comentário
N130 R11=$AA_MM2[X]	; Salvar o valor de medição correspondente ao segundo evento de disparo (flanco crescente do sensor 1) programado.
N140 APALPADOR2:	
N150 IF $AC_MEA[2]==FALSE GOTOF FIM	; Controlar o sucesso da medição com o apalpador de medição 2.
N160 R12=$AA_MM3[X]	; Salvar o valor de medição correspondente ao terceiro evento de disparo (flanco crescente do sensor 2) programado.
N170 R13=$AA_MM4[X]	; Salvar o valor de medição correspondente ao quarto evento de disparo (flanco crescente do sensor 2) programado.
N180 FIM:	

Exemplo 3: Medição contínua, específica de eixos no modo 1 (avaliação em sequência cronológica)

a) Medição de até 100 valores medição

Código de programa	Comentário
...	
N110 DEF REAL VALOR_MEDICAO[100]	
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	; Medição no modo 1 com sistema de medição ativo, salvar os valores de medição em $AC_FIFO1, esperar pelo sinal de medição com flancos decrescentes do apalpador de medição 1 no percurso até X=1000.
N135 STOPRE	
N140 MEAC[X]=(0)	; Cancelar a medição após alcançar a posição do eixo.
N150 R1=$AC_FIFO1[4]	; Salvar o número de valores de medição acumulados no parâmetro R1.
N160 FOR loop=0 TO R1-1	
N170 VALOR_MEDICAO[loop]=$AC_FIFO1[0]	; Ler e salvar os valores de medição de $AC_FIFO1.
N180 ENDFOR	

b) Medição com cancelamento de curso restante após 10 valores de medição

Código de programa	Comentário
...	
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; Excluir o curso restante.
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC[X]=(0)	
N40 R1=$AC_FIFO1[4]	; Número de valores de medição.

Código de programa	Comentário
...	

c) Medição de um flanco dentado decrescente/crescente com 2 sensores de medida

Código de programa	Comentário
...	
N110 DEF REAL VALOR_MEDICAO[16]	
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	; Medição no modo 1 com sistema de medição ativo, salvar os valores de medição em $AC_FIFO1, esperar pelo sinal de medição na sequência de flancos decrescentes do sensor 1, flancos crescentes do sensor 2, no percurso até X=100.
N140 STOPRE	; Parada de pré-processamento
N150 MEAC[X]=(0)	; Cancelar a medição após alcançar a posição do eixo.
N160 R1=$AC_FIFO1[4]	; Salvar o número de valores de medição acumulados no parâmetro R1.
N170 FOR loop=0 TO R1-1	
N180 VALOR_MEDICAO[loop]=$AC_FIFO1[0]	; Ler e salvar os valores de medição de $AC_FIFO1.
N190 ENDFOR	

Outras informações

Pedido de medição

A programação de uma tarefa de medição pode ser realizada no programa de peça ou a partir de uma ação síncrona (veja o capítulo "Ações síncronas (Página 631)"). Apenas um pedido de medição pode estar ativo ao mesmo tempo para cada eixo.

Indicação

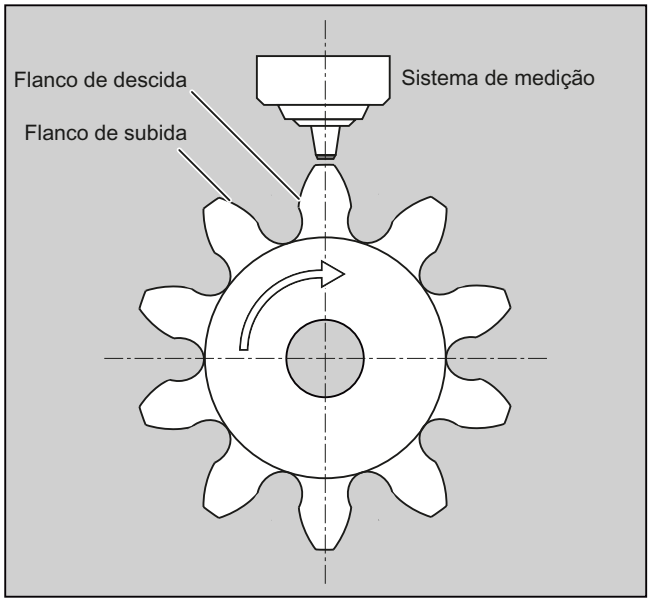
O avanço deve ser adaptado ao respectivo problema de medição.

Para MEASA e MEAWA os resultados corretos somente podem ser garantidos com avanços, nos quais não ocorre mais do que um mesmo evento de disparo e não mais do que 4 eventos diferentes de disparo em cada ciclo de controle de posição.

Na medição contínua com MEAC a relação entre o ciclo de interpolação e o ciclo de controle de posição não pode ser maior que 1:8.

Evento de disparo

Um evento de disparo é formado a partir do número do apalpador e o critério de ativação (flanco crescente ou decrescente) do sinal de medição.



Para cada medição podem ser processados até 4 eventos de disparo do apalpador de medição solicitado, portanto, até dois apalpadores de medição com dois flancos de medição cada. A sequência do processamento assim como o número máximo de eventos de disparo depende do modo selecionado.

Indicação

Para modo de medição 1 aplica-se: Um mesmo evento de disparo somente pode ser programado uma vez em uma tarefa de medição!

Com `MEAC`, o número de valores de medição por evento de disparo pode ser aumentado através da utilização de PROFIBUS Telegramm 395 para um total de 8 valores de medição com flanco ascendente e 8 com flanco descendente por evento de disparo e ciclo do regulador de posição.

- Um apalpador de medição: 8 valores de medição com flanco ascendente e 8 com flanco descendente
- Dois apalpadores de medição: 4 valores de medição com flanco ascendente e 4 com flanco descendente por apalpador de medição

Isto permite realizar, com a utilização de PROFIBUS Telegramm 395, avanços ou rotações mais elevadas.

Literatura:
Funktionshandbuch Erweiterungsfunktionen; Messen (M5), Kapitel:Manual de funções ampliadas; Medição (M5), Capítulo: Medição axial

Modo de operação

Com o primeiro dígito (casa da dezena) do modo de operação é selecionado o sistema de medição desejado. Se houver apenas um sistema de medição, mas houver um segundo programado, se aplica automaticamente o existente.

Com o segundo dígito (casa da unidade) é selecionado o modo de medição desejado. Com isso o processo de medição é adaptado às possibilidades do respectivo comando:

- **Modo 1**
  A avaliação dos eventos de disparo é realizada na sequência cronológica da ocorrência. Neste modo apenas um evento de disparo pode ser programado para módulos de seis eixos, e no caso de haverem mais eventos de disparo especificados, passa-se automaticamente para o modo 2 (sem mensagem).
- **Modo 2**
  A avaliação dos eventos de disparo é realizada na sequência programada.
- **Modo 3**
  A avaliação dos eventos de disparo é realizada na ordem programada, mas sem monitoração do evento de disparo 1 no START.

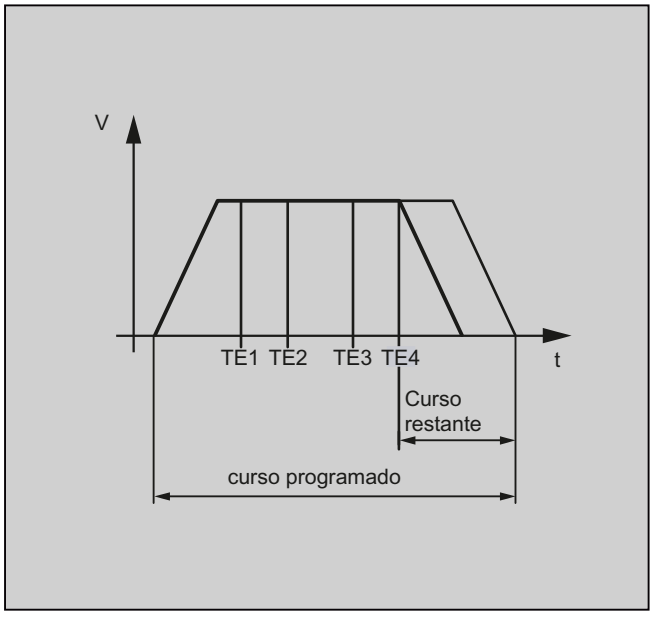
Indicação

No emprego de 2 sistemas de medição apenas se pode programar dois eventos de disparo.

Medição com e sem anulação de curso restante

Para a programação do `MEASA` a anulação de curso restante apenas será executada após a coleta dos valores de medição solicitados.

Para pedidos especiais de medição, onde para cada caso deve ser aproximada a posição programada, se aplica a função `MEAWA`.



Indicação

O `MEASA` não pode ser programado em ações síncronas. Como alternativa se pode programar o `MEAWA` mais a anulação de curso restante como ação síncrona.

Se o pedido de medição for iniciado com `MEAWA` a partir de ações síncronas, os valores de medição somente estarão disponíveis no sistema de coordenadas da máquina.

Resultados de medição para MEASA, MEAWA

Os resultados de medição estão disponíveis nas seguintes variáveis de sistema:

- No sistema de coordenadas da máquina:

`$AA_MM1[<eixo>]`	Valor de medição do sistema de medição programado com evento de disparo 1
`...`	...
`$AA_MM4[<eixo>]`	Valor de medição do sistema de medição programado com evento de disparo 4

- No sistema de coordenadas da peça de trabalho:

`$AA_MW1[<eixo>]`	Valor de medição do sistema de medição programado com evento de disparo 1
`...`	...
`$AA_MW4[<eixo>]`	Valor de medição do sistema de medição programado com evento de disparo 4

Eixos geométricos / Transformações

Se a medição axial deve ser iniciada para um eixo geométrico, deve ser programada explicitamente o mesmo pedido para os demais eixos geométricos. O mesmo vale para os eixos envolvidos em uma transformação.

Exemplos:
```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

ou
```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

Pedido de medição com 2 sistemas de medição

Se um pedido de medição for executado com dois sistemas de medição, então se coleta os dois possíveis eventos de disparo de ambos sistemas de medição do respectivo eixo. Com isso se especifica a atribuição das variáveis reservadas:

`$AA_MM1[<eixo>]`	ou	`$AA_MW1[<eixo>]`	Valor de medição do sistema de medição 1 no evento de disparo 1
`$AA_MM2[<eixo>]`	ou	`$AA_MW2[<eixo>]`	Valor de medição do sistema de medição 2 no evento de disparo 1
`$AA_MM3[<eixo>]`	ou	`$AA_MW3[<eixo>]`	Valor de medição do sistema de medição 1 no evento de disparo 2
`$AA_MM4[<eixo>]`	ou	`$AA_MW4[<eixo>]`	Valor de medição do sistema de medição 2 no evento de disparo 2

Variáveis de sistema

O estado de medição está disponível nas seguintes variáveis de sistema:

$A_PROBE[<n>]

Valor	Significado
1	Apalpador de medição defletido
0	Apalpador de medição não defletido

O limite de medição do sensor está disponível nas seguintes variáveis de sistema:

$A_PROBE_LIMITED[<n>]

Valor	Significado
1	limite de medição do sensor ativo
0	limite de medição do sensor inativo

<n> = sensor de medição

Literatura:
Manual de listas de Variáveis de sistema

Estado do pedido de medição no MEASA, MEAWA

Quando uma avaliação for necessária, então no programa pode ser consultado o estado do pedido de medição através do $AC_MEA[<n>], com <n> = número do sensor de medição. Assim que todos os eventos de disparo dos apalpadores de medição <n>, programados em um bloco, estão executados, esta variável fornece o valor 1. Caso contrário, o valor é 0.

Indicação

O $AC_MEA não será mais atualizado se a medição for iniciada a partir de ações síncronas. Neste caso devem ser consultados novos sinais da interface PLC DB31, … DBX62.3 ou a variável equivalente $AA_MEAACT[<eixo>].

Significado:

$AA_MEAACT==1: Medição ativa

$AA_MEAACT==0: Medição não ativa

Medição contínua (MEAC)

Os valores de medição estão disponíveis no `MEAC` no sistema de coordenadas da máquina e são armazenados na memória FIFO[n] (memória de circulação) especificada. Se forem configurados dois apalpadores para realizar a medição, os valores de medição do segundo apalpador são armazenados separadamente na memória FIFO[n+1] especialmente configurada (através de MD).

A memória FIFO é uma memória de circulação, na qual os valores de medida no princípio de circulação são registrados no $AC_FIFO-Variáveis (Consulte "Ações síncronas (Página 631)").

Indicação

O conteúdo FIFO somente pode ser lido uma vez da memória de circulação. Para a aplicação múltipla dos dados de medição, estes deverão ser armazenados temporariamente nos dados do usuário.

Se o número de valores de medição exceder o número máximo definido no dado da máquina para a memória FIFO, então a medição é encerrada automaticamente.

A medição contínua (sem fim) pode ser realizada através da leitura cíclica dos valores de medição. Neste caso a leitura de dados deve ser realizada na mesma frequência da entrada dos novos valores de medição.

Literatura:

- Manual de funções ações síncronas, descrição detalhada, Capítulo: Parâmetro ($AC_FIFO)
- Manual de funções ampliadas; Medição (M5), capítulo: Medição axial

Proteção contra erro de programação

As seguintes programações com erro são detectadas e exibidas com um erro:

- MEASA/MEAWA programado junto com MEAS/MEAW em um bloco
  Exemplo:
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA com número de parâmetros <2 ou >5
  Exemplo:
  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA com evento de disparo diferente de 1/ -1/ 2/ -2
  Exemplo:
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```
- MEASA/MEAWA em modo errado
  Exemplo:
  ```
  N01 MEAWA[B]=(4,1) B100
  ```
- MEASA/ MEAWAcom evento de disparo programado duas vezes
  Exemplo:
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```
- MEASA/MEAWA e eixos geométricos faltantes
  Exemplo:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;Eixos GEO X/Y/Z
  ```
- Ordem de medição inconsistente nos eixos geométricos
  Exemplo:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

5.9 Funções especiais para o usuário OEM (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

Endereços OEM

O significado dos endereços OEM é definida pelo usuário OEM. A funcionalidade é incorporada através de ciclos de compilação. Foram reservados 5 endereços OEM (`OMA1` ... `OMA5`). O identificador de endereço pode ser ajustado. Endereços OEM são permitidos em cada bloco.

Chamadas de comando G reservados

Para o usuário OEM foram reservadas as seguintes chamadas de comando G:

- `OEMIPO1`, `OEMIPO2` (do grupo 1 de G)
- `G810` ... `G819` (grupo 31 de G)
- `G820` ... `G829` (grupo 32 de G)

A funcionalidade é incorporada através de ciclos de compilação.

Funções e sub-rotinas

Além disso, os usuários OEM também podem criar funções e sub-rotinas pré-definidas com transferência de parâmetros.

Indicação

Simulação de peça de trabalho

Até o SW 4.4 não há nenhum suporte, e a partir do SW 4.4 apenas ciclos de compilação (CC) selecionados recebem suporte na simulação de peça de trabalho.

Por isso que os comandos de linguagem em programas de peça com ciclos de compilação não suportados (`OMA1` ... `OMA5`, `OEMIPO1/2`, `G810` ... `G829`, próprios procedimentos e funções) resultam em mensagens de alarme e o cancelamento da simulação, sem haver tratamento individual.

Solução: Tratar individualmente os elementos de linguagem ausentes e específicos de ciclo de compilação (CC) no programa de peça (consulta $P_SIM).
Exemplo:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ;para simulação CC não ativa
ELSE
N5 X300 OMA1=10
ENDIF
```

5.10 Redução de avanço com desaceleração nos cantos (FENDNORM, G62, G621)

Na desaceleração automática nos cantos o avanço é breve e gradativamente reduzido antes de alcançar o respectivo canto. Além disso se pode parametrizar a dimensão do comportamento de ferramenta relevante à usinagem através de dados de ajuste. São eles:

- Início e fim da redução do avanço
- Override com o qual o avanço é reduzido
- Detecção do canto relevante

Como cantos relevantes consideramos os cantos cujo ângulo interno é menor do que o canto projetado através do dado de ajuste.

A função do override automático de cantos é desativada com o valor padrão `FENDNORM`.

Literatura:
/FBFA/ Descrição do funcionamento dos dialetos ISO

Sintaxe

```
FENDNORM
G62 G41
G621
```

Significado

`FENDNORM`:	Desaceleração automática de cantos desativada
`G62`:	Desaceleração nos cantos internos com correção de raio da ferramenta ativada
`G621`:	Desaceleração em todos cantos com correção de raio da ferramenta ativada

G62 apenas tem efeito nos cantos internos com

- correção de raio de ferramenta ativada `G41`, `G42` e
- modo de controle da trajetória ativado `G64`, `G641`

O respectivo canto é aproximado com avanço reduzido, resultante de:

F * (override para redução do avanço) * override de avanço

A redução de avanço máxima possível é alcançada exatamente quando a ferramenta realizar a mudança de sentido no respectivo canto, com referência ao centro do percurso.

G621 age de forma semelhante ao G62 em cada canto dos eixos definidos por `FGROUP`.

5.11 Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Semelhante ao critério de mudança de blocos na interpolação de percurso (G601, G602 e G603), com a interpolação de eixos individuais, o critério de fim de movimento pode ser programado em um programa de peças ou ações síncronas para eixos de comando e PLC.

Dependendo de qual critério de fim de movimento for programado, os blocos do programa de peça e os blocos de ciclo de tecnologia com movimentos individuais de eixo requerem tempos diferentes para serem concluídos. O mesmo se aplica para o PLC através do FC15/16/18.

Sintaxe

```
FINEA[<eixo>]
COARSEA[<eixo>]
IPOENDA[<eixo>]
IPOBRKA(<eixo>[,<momento>])
ADISPOSA(<eixo>[,<modo>,<tamanho de janela>])
```

Significado

FINEA:	Critério de fim de movimento: "Parada exata fina"		
	Efeito:	modal	
COARSEA:	Critério de fim de movimento: "Parada exata aproximada"		
	Efeito:	modal	
IPOENDA:	Critério de fim de movimento: "Parada de interpolador"		
	Efeito:	modal	
IPOBRKA:	Critério de mudança de blocos: Rampa de frenagem		
	Efeito:	modal	
ADISPOSA:	Janela de tolerância para critério de fim de movimento		
	Efeito:	modal	
<eixo>:	Nome de eixo de canal (X, Y, ...)		
<momento>:	Momento da mudança de blocos relacionado à rampa de frenagem em %: • 100% = Início da rampa de frenagem • 0% = Fim da rampa de frenagem, de mesmo significado como o IPOENDA		
	Tipo:	REAL	
<modo>:	Referência da janela de tolerância		
	Faixa de valores:	0	Janela de tolerância não ativa
		1	Janela de tolerância em relação à posição nominal
		2	Janela de tolerância em relação à posição real
	Tipo:	INT	
<tamanho de janela>:	Tamanho da janela de tolerância		
	Tipo:	REAL	

Exemplos

Exemplo 1: Critério de fim de movimento: "Parada de interpolador"

```
Código de programa
; Desloca o eixo de posicionamento X até 100, com velo-
cidade de 200 m/min e aceleração de 90%,
; Critério de fim de movimento: Parada de interpolador
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]

; Ação síncrona:
; SEMPRE QUE: Estiver definida a entrada 1
; ENTÃO: o eixo de posicionamento X até 50, com veloci-
dade de 200 m/min e aceleração de 140%,
; Critério de fim de movimento: Parada de interpolador
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]
```

Exemplo 2: Critério de mudança de blocos: "Rampa de frenagem"

Código de programa	Comentário
	; Configuração Default ativa
N40 POS[X]=100	; Movimento de posicionamento de X para a posição 100
	Critério de mudança de blocos: Parada exata fina
N20 IPOBRKA(X,100)	; Critério de mudança de blocos: "Rampa de frenagem",
	100% = Início da rampa de frenagem
N30 POS[X]=200	; A mudança de blocos ocorre, assim que o eixo X começa a desacelerar
N40 POS[X]=250	; O eixo X não continua a desacelerar até a posição 200, mas sim prossegue o deslocamento até a posição 250.
	Assim que o eixo começa a desacelerar, ocorre a mudança de blocos.
N50 POS[X]=0	; O eixo X desacelera e desloca-se, retornando para a posição 0.
	A mudança de blocos ocorre na posição 0 e com "Parada exata fina"
N60 X10 F100	; O eixo X desloca-se como eixo de percurso até a posição 10

Outras informações

Variável de sistema para critério de fim de movimento

O critério de fim de movimento em ação pode ser interpretado através da variável de sistema $AA_MOTEND.

Literatura: /LIS2sl/ Manual de listas, Livro 2

Critério de mudança de blocos: "Rampa de frenagem" (IPOBRKA)

Se na ativação do critério de mudança de blocos "Rampa de frenagem" para o momento opcional de mudança de blocos for programado um valor, este estará ativo no próximo movimento de posicionamento e será gravado no dado de ajuste de modo sincronizado com o processamento principal. Se não for especificado nenhum valor para o momento da mudança de blocos, então o atual valor contido no dado de ajuste entrará em ação.

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Com a próxima programação de um critério de fim de movimento em eixos (`FINEA`, `COARSEA` , `IPOENDA`) é desativado o `IPOBRKA` do eixo correspondente.

Critério adicional para mudança de blocos: "Janela de tolerância" (ADISPOSA)

Com o `ADISPOSA` é possível definir uma janela de tolerância em torno do ponto final do bloco (opcionalmente posição real ou nominal) como um critério de mudança de blocos. Para a mudança de blocos é necessário que as duas condições sejam preenchidas:

- Critério de mudança de blocos: "Rampa de frenagem"
- Critério de mudança de blocos: "Janela de tolerância"

Literatura

Para mais informações sobre o critério de mudança de blocos dos eixos de posicionamento veja:

- Manual de funções ampliadas; Eixos de posicionamento (P2)
- Manual de programação Fundamentos; capítulo "Controle de avanço"

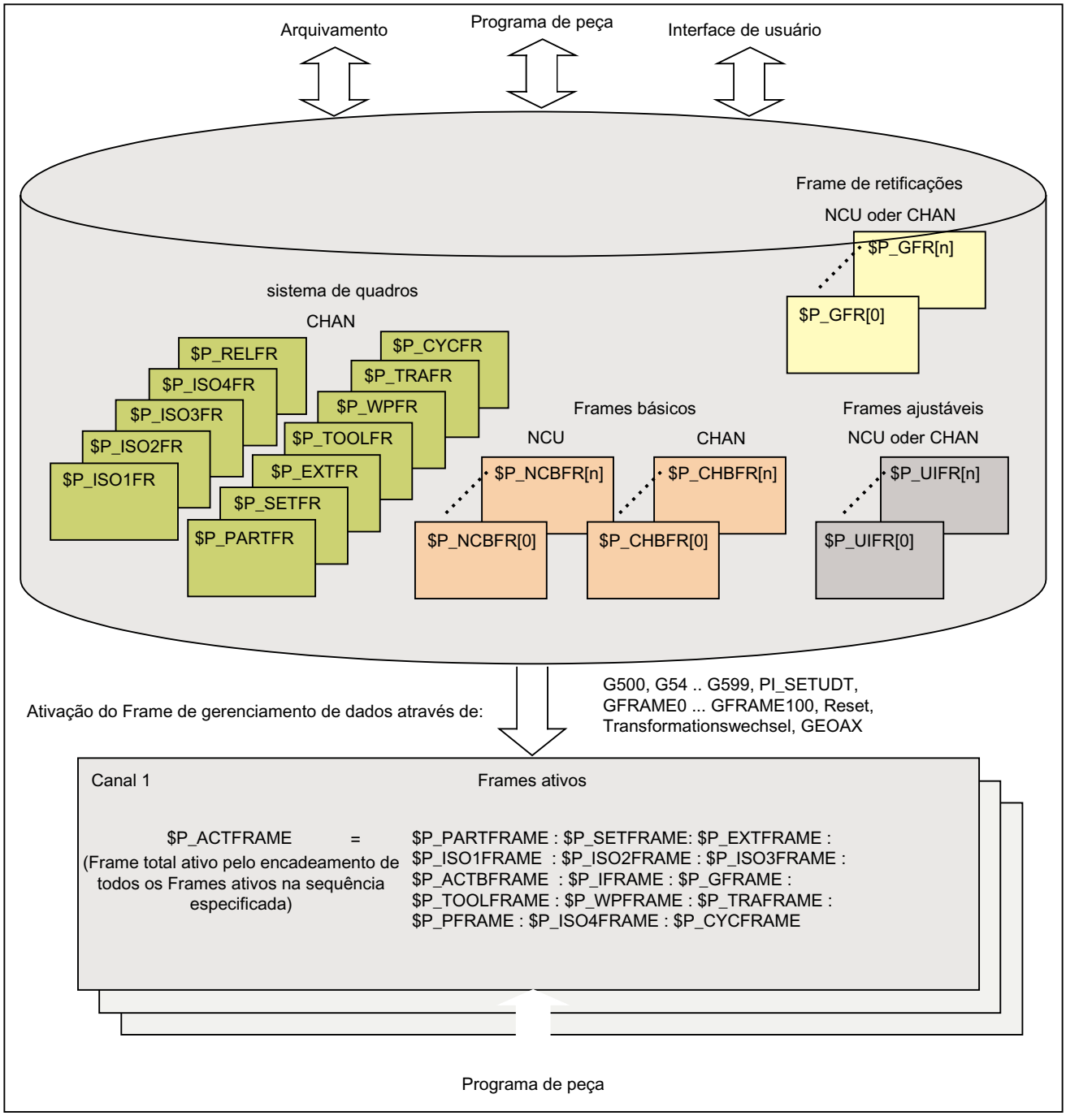
Transformações de coordenadas (Frames) 6

6.1 Transformação de coordenadas através de variável Frame

Além dos comandos descritos nos fundamentos no manual de programação, capítulo "Transformações de coordenadas (Fr ames)", como por ex., `ROT`, `AROT`, `SCALE`, etc., o sistema de coordenadas da peça de trabalho (WKS) também pode ser transformado pela variáveis Frame $P_...FR (frames de gerenciamento dos dados) e $P_...FRAME (Fr ames ativos).

A figura seguinte apresenta uma visão geral sobre a estruturação das variáveis Frame:

- Frames de gerenciamento de dados
- Frames ativos
- Frame total ativo: Interligação de todos os Fr ames ativos
- Frames básicos globais da NCU
- Frames específicos de canal



Esquema 6-1 Visão geral das variáveis Frame

6.1.1 Variável Frame pré-definida ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)

Ativo: Fr ames básicos específicos do canal $P_CHBFRAME[<n>] ($P_BFRAME)

Indicação

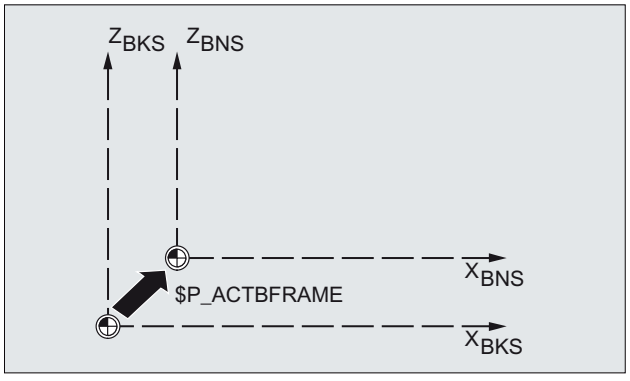
O do frame básico atual `$P_BFRAME` e o frame básico de gerenciamento dos dados `$P_UBFR` são mantidos por motivos de compatibilidade.

- `$P_BFRAME` □ `$P_CHBFRAME[0]`
- `$P_UBFR` □ `$P_CHBFR[0]`.

As variáveis Frame $P_CHBFRAME[<n>] definem a relação entre o sistema de coordenadas básico (BKS) e o sistema de ponto zero básico (BNS).

Se o frame básico específico do canal atual $P_CHBFRAME[`<n>`] tiver que ter efeito imediato no programa NC, estão disponíveis as seguintes possibilidades

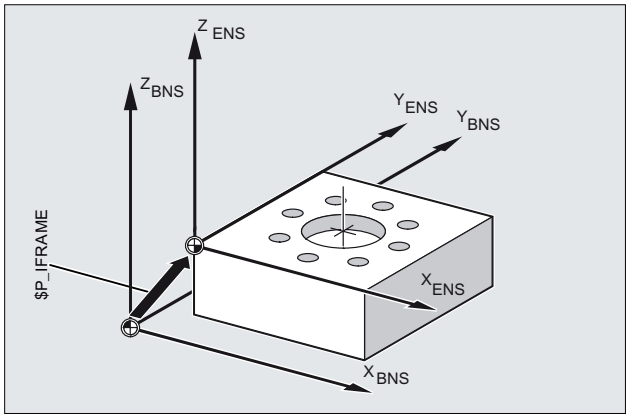
- Comandos:
  - `G500` (desativação de todos os Fr ames configuráveis, os frames básicos permanecem ativos)
  - `G54` ... `G599` (deslocamentos de ponto zero configuráveis)
- Atribuição de um frame básico específico do canal de gerenciamento dos danos a um frame básico específico do canal atual:
  `$P_CHBFRAME[<n>] = $P_CHBFR[<m>]`



Ativo: Frame configurável específico do canal $P_IFRAME

A variável Frame $P_IFRAME, define a relação entre o sistema de ponto zero básico (BNS) e o sistema de ponto zero configurável (ENS).

- `$P_IFRAME` corresponde ao `$P_UIFR[$P_IFRNUM]`
- `$P_IFRAME` depois da programação de, por ex., `G54` contém a translação, rotação, escala e espelhamento definido por `G54`.

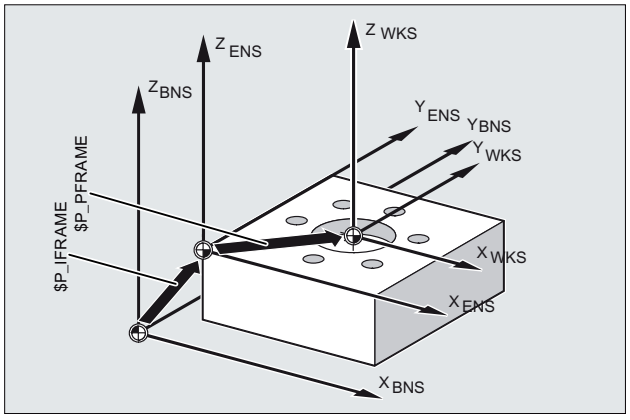


Ativo: Frame programável específico do canal $P_PFRAME

A variável Frame $P_PFRAME, define a relação entre o sistema de ponto zero configurável (ENS) e o sistema de coordenadas da peça de trabalho (WKS).

`$P_PFRAME` contém o Frame resultante, que

- resulta **da programação** do `TRANS/ATRANS`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR` e
- **a partir da atribuição** do `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` no FRAME programável

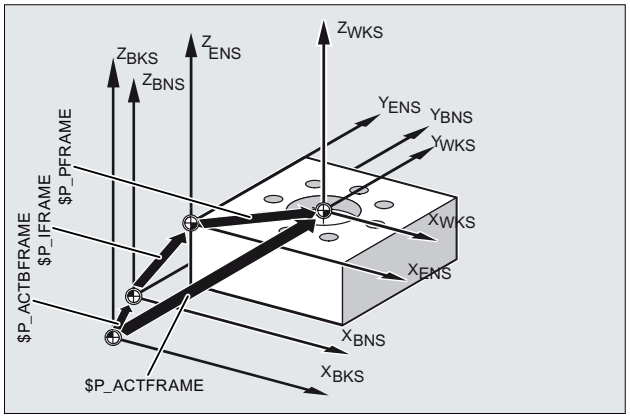


Ativo: Frame total $P_ACTFRAME

O frame total que tem efeito no canal, resulta da interligação de todos os frames que atuam no canal.

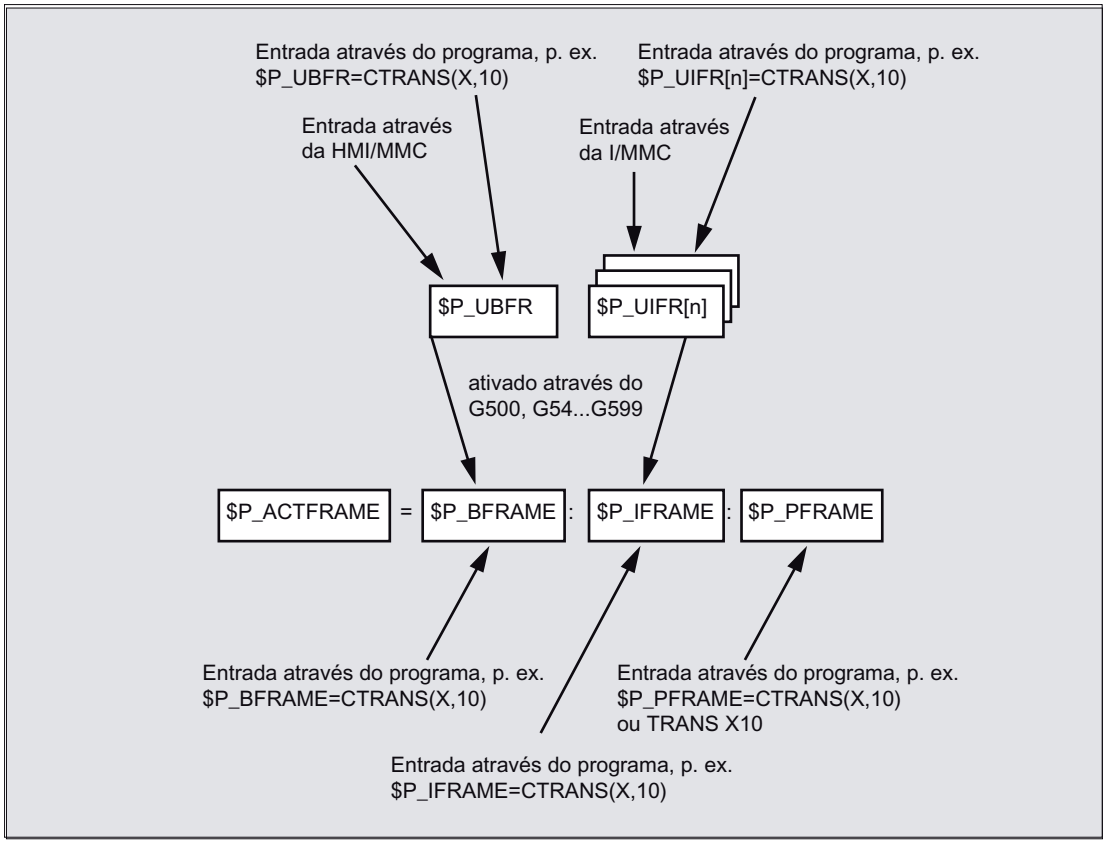
```
$P_ACTFRAME =    $P_PARTFRAME: $P_SETFRAME: $P_EXTFRAME:
                 $P_ISO1FRAME: $P_ISO2FRAME : $P_ISO3FRAME:
                 $P_ACTBFRAME: $P_IFRAME: $P_GFRAME:
                 $P_TOOLFRAME: $P_WPFRAME: $P_TRAFRAME:
                 $P_PFRAME   : $P_ISO4FRAME: $P_CYCFRAME
```

`$P_ACTFRAME` descreve o ponto zero do sistema de coordenadas da peça de trabalho atual.



Esquema 6-2 Variável Frame $P_ACTFRAME

Se um dos seguintes Fr ames `$P_BFRAME` / `$P_CHBFRAME[<n>]`, `$P_IFRAME` ou `$P_PFRAME` for alterado, o frame total atual `$P_ACTFRAME` será recalculado.



O Frame básico e o Frame ajustável tem efeito após Reset, se o MD 20110 RESET_MODE_MASK estiver ajustado da seguinte forma:

Bit0=1, Bit14=1 -->atua o `$P_UBFR` (Frame básico)

Bit0=1, Bit5=1 -->atua o `$P_UIFR[$P_UIFRNUM]` (Frame ajustável)

Gerenciamento dos dados: Fr ames básicos específicos do canal $P_CHBFR[<n>]

Através das variáveis Frame $P_CHBFR[<n>], os frames básicos são carregados / gravados no gerenciamento dos dados. O frame de gerenciamento dos dados não se torna imediatamente ativo no canal pela gravação. A ativação do Frame gravado é realizado no caso de:

- Reset do canal e MD20110 $MC_RESET_MODE_MASK,Bit0 == 1 e Bit14 == 1
- Comando `G500`, `G54` ... `G57`, `G505` ... `G599` (ativar/desativar os frames básicos com o recálculo seguinte do frame total atual)

Gerenciamento dos dados: Fr ames configuráveis específicos do canal $P_UIFR[<n>]

Através das variáveis Frame $P_UIFR[<n>], os Fr ames configuráveis são carregados / gravados no gerenciamento dos dados. O frame não se torna imediatamente ativo no canal pela gravação. O cálculo do Frame gravado no canal é realizado no caso de:

- Comando `G500` (desativar todos os Fr ames ou deslocamentos de ponto zero configuráveis)
- Comando `G54` ... `G57`, `G505` ... `G599` (ativar um Frame ou deslocamento do ponto zero configurável)

Frame ajustável ativo	Frame de gerenciamento dos dados	(corresponde ao comando)
`$P_IFRAME` =	`$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 Atribuições de valores nos frames

6.2.1 Atribuir valores diretos (valor de eixo, ângulo, escala)

No programa NC podemos atribuir valores diretamente em Frames ou em variáveis Frame.

Sintaxe

Sintaxe

```
$P_PFRAME = CTRANS(X, <Valor de deslocamento>, Y, <Valor de deslocamento>, Z, <Valor de deslocamento>, ...)
$P_PFRAME = ROT(X, <ângulo>, Y, <ângulo>, Z, <ângulo>, ...)
$P_UIFR[..] = CROT(X, <ângulo>, Y, <ângulo>, Z, <ângulo>,  ...
$P_PFRAME = CSCALE(X, <escala>, Y, <escala>, Z, <escala>, ...)
$P_PFRAME=CMIRROR (X, Y, Z)
```

A sintaxe para `$P_CHBFRAME[<n>]` é idêntica a `$P_PFRAME`.

Significado

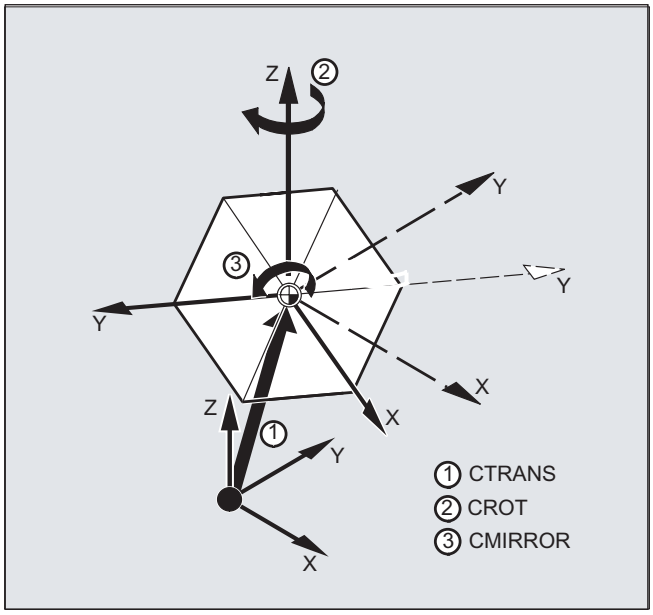
`CTRANS`:	Deslocamento nos eixos especificados
`CROT`:	Rotação em torno dos eixos especificados
`CSCALE`:	Alteração de escala nos eixos especificados
`CMIRROR`:	Mudança de sentido dos eixos especificados
`X, Y, Z`:	Valor de deslocamento no sentido do eixo geométrico especificado
`<Valor de deslocamento>`:	valor do deslocamento
`<ângulo>`:	Ângulo segundo o qual é girado
`<Escala>`:	Indicação da escala

Exemplos

Atribuições de valor aos componentes Frame do frame programável atual

Atribuição de valor aos componentes Frame para a translação, a rotação e a reflexão do frame programável atual:

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5): CROT(Z,45): CMIRROR(Y)
```



Gravar os componentes de rotação de um frame

Atribuição de valores a todos os três eixos do componente de rotação do frame de gerenciamento de dados configurável `$P_UIFR` com `CROT`:

```
$P_UIFR[5]=CROT (X, 0, Y, 0, Z, 0)
```

Como alternativa a isto, atribuição direta dos valores individuais diretamente ao respectivo eixo do componente de rotação do frame de gerenciamento de dados:

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

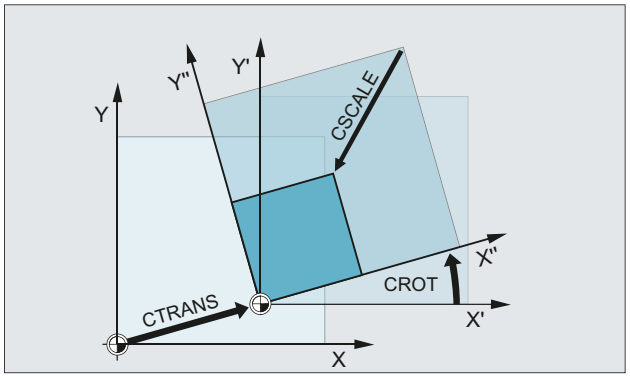
Descrição

Várias operações em um frame podem ser ligadas pelo operador de concatenação `:` entre si. As operações então são executadas em sequência da esquerda para a direita.

Exemplo

Operações encadeadas no $P_PFRAME com deslocamento, rotação e escalonamento:

```
$P_PFRAME = CTRANS(...): CROT(...): CSCALE...
```



6.2.2 Ler e alterar componentes de Frame (TR, FI, RT, SC, MI)

Temos a possibilidade de acessar **determinados** dados de um Frame como um determinado valor de deslocamento ou ângulo de rotação. Estes valores podem ser modificados ou atribuídos em outra variável.

Sintaxe

`R10=$P_UIFR[$P_UIFNUM,X,RT]`	O ângulo de rotação RT em torno do eixo X do atual deslocamento ajustável do ponto zero $P_UIFR-NUM deve ser atribuído na variável R10.
`R12=$P_UIFR[25,Z,TR]`	O valor do deslocamento TR em Z do bloco de dados do Frame nº 25 ajustado deve ser atribuído na variável R12.
`R15=$P_PFRAME[Y,TR]`	O valor do deslocamento TR em Y do atual Frame programável deve ser atribuído na variável R15.
`$P_PFRAME[X,TR]=25`	O valor do deslocamento TR em X do atual Frame programável deve ser modificado. X25 é válido de imediato.

Significado

`$P_UIFRNUM`:	`Com estas variáveis se estabelece automaticamente a relação do atual deslocamento ajustável de ponto zero válido.`
`P_UIFR[n,…,…]`:	`Com a especificação do número de Frame n se pode acessar o Frame ajustável n° n.`
	`Indicação dos componentes que devem ser lidos ou modificados:`
`TR`:	`TR Translation`
`FI`:	`FI Translation Fine`
`RT`:	`RT Rotation`
`SC`:	`Alteração de escala SC Scale`

MI:	Espelhamento MI
X, Y, Z:	Também (veja exemplos) é especificado o eixo X, Y e Z correspondente.

Faixa de valores para rotação RT

Rotação em torno do 1º eixo geomé-trico:	-180° até +180°
Rotação em torno do 2º eixo geomé-trico:	-90° até +90°
Rotação em torno do 3º eixo geomé-trico:	-180° até +180°

Descrição

Chamar Frame

Com a indicação das variáveis $P_UIFRNUM pode-se acessar diretamente o atual deslocamento de ponto zero
(o $P_UIFRNUM contém o número do atual Frame definido) ajustado com $P_UIFR ou G54, G55, ...

Todos demais Frames ajustáveis $P_UIFR que estiverem armazenados serão chamados com a indicação do no respectivo número $P_UIFR[n].

Para as variáveis Frame pré-definidas e os Frames de definição própria especifique o nome, p. ex. $P_IFRAME.

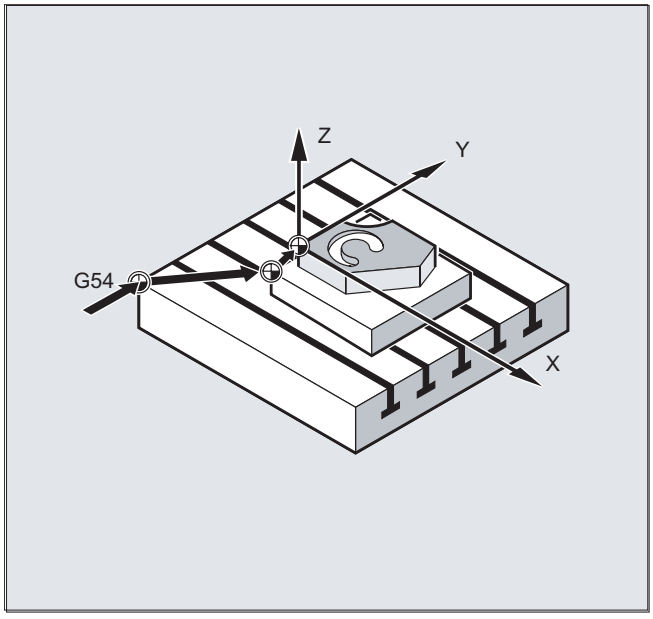
Chamar dados

Entre os colchetes constam o nome do eixo e o componente de Frame do valor que se deseja acessar ou modificar, p. ex. [X, RT] ou [Z, MI].

6.2.3 Cálculo com Fr ames

No programa NC um frame pode ser atribuído a outro frame ou os frames podem ser encadeados entre si.

Por exemplo, o encadeamento de Fr ames é indicado para descrição de várias peças de trabalho, que estão dispostas sobre um palete e devem ser usinadas em um processo de usinagem.



Para a descrição das tarefas dos paletes, por exemplo, os componentes de Frame poderiam conter apenas determinados valores parciais, através do encadeamento dos componentes podem ser gerados vários pontos zero da peça de trabalho.

Exemplos

Atribuições

Código de programa	Comentário
`DEF FRAME EINSTELLUNG_1`	`; Definição de uma variável frame local`
`EINSTELLUNG_1 = CTRANS(X,10)`	`; Atribuição do resultado de uma função à variável Frame`
`$P_PFRAME = EINSTELLUNG_1`	`; Atribuição da variável Frame ao Frame atual`
`DEF FRAME EINSTELLUNG_4`	`; Definição de uma variável frame local`
`EINSTELLUNG_4 = $P_PFRAME`	`; Armazenamento temporário do Frame atual na variável Frame`
`...`	
`$P_PFRAME = EINSTELLUNG_4`	`; Carregamento de retorno do Frame atual a partir da variável Frame`

Encadeamentos

Os Fr ames são encadeados na sequência programável pelo operador : entre si. Os componentes Frame, como por ex., deslocamentos, rotações, etc., são executadas por adição em sequência.

Código de programa	Comentário

`$P_IFRAME = $P_UIFR[15]: $P_UIFR[16]`	`; Atribuição do Frame de resultado a partir do encadeamento de`
	`; ambos os frames de gerenciamento de dados configuráveis ao`
	`; Frame total configurável ativo.`
	`; Exemplo de aplicação:`
	`; $P_UIFR[15]: Deslocamento`
	`; $P_UIFR[16]: Rotação`

`$P_UIFR[3] = $P_UIFR[4]: $P_UIFR[5]`	`; Atribuição do Frame de resultado a partir do encadeamento de`
	`; ambos os frames de gerenciamento de dados configuráveis a um`
	`; outro frame de gerenciamento de dados configurável`

6.2.4 Definição das variáveis Frame (DEF FRAME)

Além das variáveis Frame predefinidas, também podem ser definidas as variáveis Frame próprias. As variáveis Frame de definição própria são variáveis de usuários do tipo FRAME. O nome do Frame pode ser concedido livremente no âmbito das regras para as variáveis de usuários.

Com as funções CTRANS, CROT, CSCALE, CMIRROR, podem ser atribuídos valores às variáveis Frame de definição própria.

Sintaxe

```
DEF FRAME <nome>
```

Significado

`DEF FRAME`:	Definir as variáveis de usuários do tipo FRAME.
<Nome>:	Nome da variável frame

Exemplo

Definição de uma variável Frame "PALETTE" e atribuição dos valores de deslocamento e de rotação:

Código de programa	Comentário
`DEF FRAME PALETTE`	`; Definir a variável Frame PALETTE`

Código de programa	**Comentário**
`PALETTE = CTRANS(...): CROT(...)`	`; Atribuição do frame de resultado do encadeamento de` `; deslocamento e rotação à variável Frame PALETTE`

6.3 Deslocamento aproximado e preciso (CTRANS, CFINE)

Deslocamento fino

Um deslocamento preciso `CFINE(...)` pode ser utilizado no seguinte Frame:

- Fr ames configuráveis: $P_UIFR ou $P_IFRAME
- Fr ames básicos: $P_NCBFR[<n>], $P_CHBFR[<n>] ou $P_CHBFRAMES[<n>] ou $P_ACTBFRAME
- Frame programável: $P_PFRAME

o deslocamento leve de um quadro é programado através do comando `CFINE(...)`.

Deslocamento aproximado

O deslocamento aproximado `CTRANS(...)` pode ser utilizado em todos os Fr ames.

Deslocamento total

O deslocamento total resulta da adição do deslocamento aproximado e do preciso.

Dados de máquina

Liberação do deslocamento preciso

O deslocamento preciso é liberado com o dado da máquina:

MD18600 $MN_MM_FRAME_FINE_TRANS = 1

Sintaxe

Deslocamento fino

- Frame total
  - `<Frame> = CFINE(<K_1>,<valor>)`
  - `<Frame> = CFINE(<K_1>,<valor>, <K_2>, <valor>)`
  - `<Frame> = CFINE(<K_1>,<valor>, <K_2>, <valor>, <K_3>, <valor>)`
- Componentes do quadro
  - `<Frame>[<n>, <K_1>, FI] = <valor>`

Deslocamento aproximado

- Frame total
  - `<Frame> = CTRANS(<K_1>,<valor>)`
  - `<Frame> = CTRANS(<K_1>,<valor>, <K_2,<valor>)`
  - `<Frame> = CTRANS(<K_1>,<valor>, <K_2,<valor>, <K_3,<valor>)`
- Componentes do quadro
  - `<Frame>[<n>,<K_1>,TR] = <valor>`

Especial para o Frame programável `$P_PFRAME`:

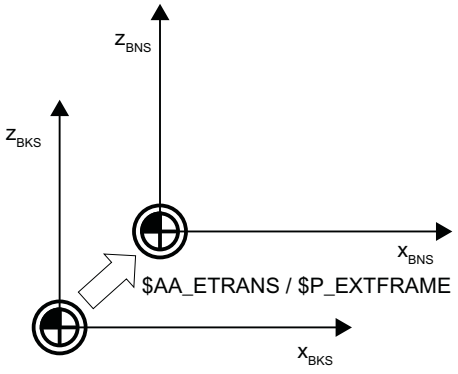
- `TRANS <K_1> <valor>`
- `TRANS <K_1> <valor> <K_2> <valor>`
- `TRANS <K_1> <valor> <K_2> <valor> <K_3> <valor>`

Significado

`<Frame>`:	Frame, por ex., Frame configurável do gerenciamento de dados `$P_UIFR[<n>]`
`CFINE`:	Deslocamento preciso, deslocamento aditivo.
`CTRANS`:	Deslocamento aproximado, deslocamento absoluto
`TRANS`:	Somente Frame programável: Deslocamento aproximado, deslocamento absoluto
`<K_n>`:	Eixos de coordenadas X, Y, Z
`<valor>`:	valor do deslocamento

6.4 Deslocamento externo de ponto zero ($AA_ETRANS)

O deslocamento do ponto zero externo é um deslocamento linear entre o sistema de coordenadas básico (BKS) e o sistema de ponto zero básico (BNS).



O deslocamento de ponto zero externo por meio de $AA_ETRANS tem efeito, dependendo da parametrização dos dados da máquina, de dois modos:

1. A variável de sistema $AA_ETRANS tem efeito após a ativação pelo sinal de interface NC/PLC diretamente como valor de deslocamento
2. O valor da variável de sistema $AA_ETRANS, após a ativação do sinal de interface NC/PLC, é transferido ao frame de sistema ativo $P:EXTFRAME e ao frame de gerenciamento de dados $P_EXTFR. Em seguida, o frame total ativo $P_ACTFRAME é recalculado.

Dados de máquina

Em conexão com a variável de sistema $AA_ETRANS deve ser diferenciado entre dois procedimentos, que são selecionados pelo seguinte dado da máquina:

MD28082 $MC_MM_SYSTEM_FRAME_MASK,Bit1 = <valor>

<valor>	Significado
0	Função: Gravação direta de $AA_ETRANS[<eixo>] pelo PLC, HMI ou programa NC. Liberação para o afastamento do deslocamento de ponto zero de $A_ETRANS[<eixo>] no próximo bloco de deslocamento possível: DB31, ... DBX3.0
1	Função: Ativação do frame de sistema ativo $P:EXTFRAME e do frame de gerenciamento de dados $P_EXTFR Liberação para o afastamento do deslocamento de ponto zero de $AA_ETRANS[<eixo>] por: DB31, ... DBX3.0. Por conseguinte, no canal ocorre: • Parada de todos os movimentos de deslocamento no canal (exceto dos eixos de comando e PLC) • Parada de avanço com reorganização seguinte (STOPRE) • Deslocamento aproximado do Frame ativo $P_EXTFRAME[<eixo>] = $AA_ETRANS[<eixo>] • Deslocamento aproximado do frame de gerenciamento de dados $P_EXTFR[<eixo>] = $AA_ETRANS[<eixo>] • Recálculo do frame total ativo $P_ACTFRAME • Afastamento do deslocamento nos eixos programados. • Continuação do movimento de deslocamento interrompido ou do programa NC

Programação

- Sintaxe
  `$AA_ETRANS[<eixo>] = <valor>`
- Significado

`$AA_ETRANS`:	Variável de sistema para o armazenamento temporário do deslocamento de ponto zero externo
`<eixo>`:	Eixo de canal
`<valor>`:	valor do deslocamento

Sinal de interface NC/PLC

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<eixo>] = $P_EXTFR[<eixo>] = $AA_ETRANS[<eixo>]

6.5 Definição do valor real com perda do status de referenciamento (PRESETON)

O procedimento `PRESETON()` define para um ou vários eixos um novo valor real no sistema de coordenadas da máquina (MKS). Isto corresponde a um deslocamento do ponto zero do MKS do eixo. Com isto, o eixo não é deslocado.

Pelo `PRESETON` é iniciada uma parada de avanço com sincronização. A posição real é atribuído ao eixo somente com a imobilização.

Se o eixo com o `PRESETON` não for atribuído ao canal, o procedimento contínuo dependerá da execução de projeto específica do eixo do comportamento de troca de eixo:

MD30552 $MA_AUTO_GET_TYPE

Status de referenciamento

Através da definição de um novo valor real no sistema de coordenadas da máquina, o status de referenciamento do eixo da máquina é zerado:

DB31, ... DBX60.4 / .5 = 0 (sistema de medição referenciado / sincronizado 1 / 2)

Recomenda-se, utilizar o `PRESETON` apenas para os eixos sem obrigação do ponto de referência.

Para restaurar o sistema de coordenadas da máquina original, o sistema de medição do eixo da máquina deve ser novamente referenciado, por ex., pela marcha de ponto de referência do programa de peças (`G74`).

⚠ CUIDADO

Perda do status de referenciamento

Através da definição de um novo valor real no sistema de coordenadas da máquina com `PRESETON`, o status de referenciamento do eixo da máquina é restaurado para "não referenciado / sincronizado".

Programação

Sintaxe

```
PRESETON(<eixo_1>, <valor_1> [, <eixo_2>, <valor_2>, ... <eixo_8>,
<valor_8>])
```

Significado

`PRESETON`:	Definição de valor real com perda do status de referenciamento	
	Parada de pré-processamento:	sim
	Sozinho no bloco:	sim
`<eixo_x>`:	Nome de eixo da máquina	
	Tipo:	AXIS
	Faixa de valores:	Nome do eixo da máquina definido no canal

<valor_x>:	Novo valor real do eixo da máquina no sistema de coordenadas da máquina (MKS) A especificação é realizada no sistema de medidas em vigor atualmente (polegadas / métrico) Uma programação de diâmetro ativa (DIAMON) será considerada	
	Tipo:	REAL

Literatura

PRESETONS nos programas NC

Uma descrição detalhada do PRESETON nos programas NC pode ser encontrada em:

Manual de funcionamento das funções básicas; capítulo "K2: Eixos, sistemas de coordenadas, Fr ames" > "Sistemas de coordenadas" > "Sistema de coordenadas da máquina (MKS)" > "Definição do valor real com perda do status de referenciamento (PRESETON)"

PRESETONS nas ações de sincronização

Uma descrição detalhada do PRESETON nas ações de sincronização pode ser encontrada em:

Manual de funcionamento para ações síncronas; capítulo: "Descrição detalhada" > "Ações nas ações de sincronização" > "Definição do valor real com perda do status de referenciamento (PRESETON)"

6.6 Definição do valor real sem a perda do status de referenciamento (PRESETONS)

O procedimento `PRESETONS()` define para um ou vários eixos um novo valor real no sistema de coordenadas da máquina (MKS). Isto corresponde a um deslocamento do ponto zero do MKS do eixo. Com isto, o eixo não é deslocado.

Através do `PRESETONS` é iniciada uma parada de avanço com sincronização. A posição real é atribuído ao eixo somente com a imobilização.

Se o eixo com o `PRESETONS` não for atribuído ao canal, o procedimento contínuo dependerá da execução de projeto específica do eixo do comportamento de troca de eixo:

MD30552 $MA_AUTO_GET_TYPE

Status de referenciamento

Através da definição de um novo valor real no sistema de coordenadas da máquina (MKS) com `PRESETONS`, o status de referenciamento do eixo da máquina **não** é alterado.

Pré-requisitos

- **Tipo de codificador**
  `PRESETONS` é possível somente no caso dos seguintes tipos de sensor do sistema de medição ativo:
  - MD30240 $MA_ENC_TYPE[<sistema de medição>] = 0 (sensor simulado)
  - MD30240 $MA_ENC_TYPE[<sistema de medição>] = 1 (sensor do sinal bruto)
- **Modo de referenciamento**
  `PRESETONS` é possível somente no caso dos seguintes modos de referenciamento do sistema de medição ativo:
  - MD34200 $MA_ENC_REFP_MODE[<sistema de medição>] = 0 (não é possível nenhuma marcha ao ponto de referência)
  - MD34200 $MA_ENC_REFP_MODE[<sistema de medição>] = 1 (referenciamento dos sistemas de medição incrementais, rotativos ou lineares: impulso zero na faixa do sensor)

Programação

Sintaxe

```
PRESETONS(<eixo_1>, <valor_1> [, <eixo_2>, <valor_2>, ... <eixo_8>,
<valor_8>])
```

Significado

`PRESETONS`:	Definição do valor real sem perda do status de referenciamento	
	Parada de pré-processamento:	sim
	Sozinho no bloco:	sim
`<eixo_x>`:	Nome de eixo da máquina	
	Tipo:	AXIS
	Faixa de valores:	Nome do eixo da máquina definido no canal

<valor_x>:	Novo valor real atual do eixo da máquina no sistema de coordenadas da máquina (MKS) A especificação ocorre no sistema de medidas ativo (polegadas / métrico) Uma programação de diâmetro ativa (DIAMON) será considerada	
	Tipo:	REAL

Literatura

PRESETONS nos programas NC

Uma descrição detalhada do PRESETONS nos programas NC pode ser encontrada em:

Manual de funcionamento das funções básicas; capítulo "K2: Eixos, sistemas de coordenadas, Fr ames" > "Sistemas de coordenadas" > "Sistema de coordenadas da máquina (MKS)" > "Definição do valor real sem perda do status de referenciamento (PRESETONS)"

PRESETONS nas ações de sincronização

Uma descrição detalhada do PRESETONS nas ações de sincronização pode ser encontrada em:

Manual de funcionamento para ações síncronas; capítulo: "Descrição detalhada" > "Ações nas ações de sincronização" > "Definição do valor real sem perda do status de referenciamento (PRESETONS)"

6.7 Cálculo de Frame a partir de 3 pontos de medição no espaço (MEAFRAME)

A função MEAFRAME é usada como apoio para os ciclos de medição. Ela calcula o quadro a partir de três pontos ideais e dos pontos medidos correspondentes.

Se uma peça é posicionada para a usinagem, então sua posição relativa ao sistema de coordenadas cartesiano da máquina normalmente está deslocada e também girada em relação à sua posição ideal. Para a usinagem ou medição exata, torna-se necessário ou um ajuste físico dispendioso ou então a modificação dos movimentos no programa de peça.

Um Frame pode ser definido através do apalpamento de três pontos no espaço, cujas posições ideais são conhecidas. O apalpamento é realizado com um sensor de contato ou óptico, que entra especialmente em contato com os furos de precisão ou esferas de medição da placa de suporte.

Sintaxe

```
MEAFRAME(<ponto ideal>,<ponto de medida>,<Qualidade>)
```

Significado

`MEAFRAME`:	Chamada da função		
`<ponto ideal>`:	2-dim. Campo REAL, que contém as três coordenadas dos pontos ideais		
`<ponto de medida>`:	2-dim. Campo REAL, que contém as três coordenadas dos pontos medidos		
`<Qualidade>`:	Variável, com a qual será devolvida a informação da qualidade do calculo do quadro		
	Tipo:	VAR REAL	
	Valor:	-1	Os pontos ideais estão próximos de uma reta: O Frame não pôde ser calculado. A variável do quado retornado contém um quadro neutro.
		-2	Os pontos medidos estão próximos de uma reta: O Frame não pôde ser calculado. A variável do quado retornado contém um quadro neutro.
		-4	O cálculo da matriz de rotação falhou por outro motivo
		≥ 0.0	Soma das distorções (distâncias entre os pontos) que são necessárias para a transformar o triângulo medido em um triângulo ideal e congruente.

Indicação

Qualidade da medição

Para que as coordenadas medidas sejam atribuídas às coordenadas ideais com uma rotação/translação combinada, o triângulo formado pelos pontos de medição deverá ser congruente com o triângulo ideal. Isto se alcança com um algoritmo de compensação que minimiza a somatória do quadrado dos desvios que reformam o triângulo medido para o triângulo ideal.

A distorção efetiva necessária dos pontos de medição pode servir de indicador para a qualidade da medição e por isso é especificada como variável adicional do MEAFRAME.

Indicação

O quadro produzido pelo MEAFRAME pode ser transformado através da função ADDFRAME em um outro quadro da sequência de quadros (veja exemplo "Interligação com ADDFRAME").

Exemplos

Exemplo 1:

Programa de usinagem 1:

```
Código de programa
...
DEF FRAME CORR_FRAME
```

Definição de pontos de medição:

```
Código de programa                                  Comentário
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]=                           ; Para teste.
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                          ; Permitidos no máx. 5 graus de ro-
                                                    tação da posição da peça.
DEF REAL FIT_QUALITY_LIMIT=3                        ; Permitidos no máx. um deslocamen-
                                                    to de 3 mm entre
                                                    o triângulo ideal e o medido.
DEF REAL SHOW_MCS_POS1[3]
DEF REAL SHOW_MCS_POS2[3]
DEF REAL SHOW_MCS_POS3[3]
```

Código de programa	**Comentário**
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	; Limitação do 1° ângulo RPY.
SETAL(65020)	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT	; Limitação do 2° ângulo RPY.
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; Limitação do 3° ângulo RPY.
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 $P_IFRAME=CORR_FRAME	; Ativar o Frame de varredura com um Frame de definição. ; Verificar o Frame com o posicionamento dos eixos geométricos nos pontos ideais.
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=$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	; Desativar o Frame de definição, já que com Frame zero (pré-ocupado sem valor especificado).

Código de programa	Comentário
No_FRAME	; Desativar o Frame de definição, já que pré-ocupado com Frame zero (sem valor especificado).
M0	
M30	

Exemplo 2: Interligação de quadros

Encadeamento de MEAFRAME para correções

A função MEAFRAME fornece uma correção do quadro. Se esta correção for interligada com o quadro configurado $P_UIFR[1], o qual estava ativo na chamada da função (ex. G54), então obtemos um quadro configurável para outros cálculos de deslocamento e usinagem.

Encadeamento com ADDFRAME

Se esta correção atuar em outra posição na interligação dos quadros, ou se antes do quadro ajustável ainda existirem outros quadros ativos, então se pode utilizar a função ADDFRAME para interligar um dos quadros básicos de canal ou um quadro de sistema.

Neste caso, no Frame não podem estar ativos:

- Com MIRROR/AMIRROR
- Escalonamento com SCALE

Os parâmetros de entrada para valores nominais e reais são as coordenadas da peça de trabalho. No sistema base de controle, essas coordenadas estão especificadas metricamente ou em polegadas (G71/G70) e também como medida de referência ao raio (DIAMOF).

Literatura:
Mais informações sobre ADDFRAME, veja:
Manual de funcionamento das funções básicas; K2: Eixos, sistema de coordenadas, quadros

6.8 Frames globais da NCU

Para cada NCU os Frames globais da NCU ocorre apenas uma vez em todos canais. Os Frames globais da NCU podem ser gravados e lidos a partir de todos canais. A ativação dos Frames globais da NCU é realizada no respectivo canal.

Através dos Frames globais os **eixos de canal e eixos da máquina** podem ter escala alterada e serem espelhados com os deslocamentos.

Relações geométricas e encadeamento de Frames

Para Frames globais não existe nenhuma relação geométrica entre os eixos. Por isso que não pode ser executada nenhuma rotação e nenhuma programação de identificadores de eixos geométricos.

- As rotações não podem ser aplicadas em Frames globais. A programação de uma rotação será rejeitada com o alarme: "Canal 18310 %1 bloco %2 quadro: rotação não permitida".
- A realização do encadeamento de Frames globais e Frames específicos de canal é possível. O Frame resultante contém todos componentes de Frame, inclusive as rotações de todos eixos. A atribuição de um quadro com componentes de rotação em um quadro global é negada com o alarme: "Frame: Rotação não permitida".

Frames básicos globais da NCU

Frames básicos globais da NCU $P_NCBFR[n]

Podem ser configurados até 8 Frames básicos globais da NCU:

Ao mesmo tempo podem existir Frames básicos específicos de canal.

Frames globais podem ser gravados e lidos a partir de todos canais de uma NCU. O usuário deve providenciar uma coordenação de canais ao gravar Frames globais. Este pode ser realizado, p.ex. através dos marcadores Wait-Marken (WAITMC).

Fabricante da máquina

A quantidade de quadros base é projetada através dos dados da máquina.

Literatura:
Manual de funcionamento das funções básicas; Eixos, Sistemas de coordenadas, Frames (K2)

Frames ajustáveis globais da NCU $P_UIFR[n]

Todos os quadros ajustáveis G500, G54...G599 podem ser configurados como globais da NCU ou específicos de canal.

Fabricante da máquina

Todos os quadros ajustáveis podem ser reconfigurados em quadros globais com o auxílio de um dado de máquina MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES.

Como identificador de eixo nos comandos de programação de Frames podem ser utilizados identificadores específicos de canal e identificadores de eixo de máquina. A programação dos identificadores de eixos geométricos é negada com um alarme.

6.8.1 Frames específicos de canal ($P_CHBFR, $P_UBFR)

Quadros configuráveis ou base podem ser gravados e lidos através do programa de usinagem e do BTSS da condição e do PLC.

O deslocamento fino também é possível nos Frames globais. A supressão dos quadros globais é realizada da mesma forma como dos quadros específicos de canal, através do G53, G153, SUPA e G500.

Fabricante da máquina

Através do MD28081 $MC_MM_NUM_BASE_FRAMES é possível projetar o números de quadros base no canal. A configuração padrão está definida de modo que exista pelo menos um Frame básico por canal. No máximo são possíveis 8 Frames básicos. Além dos 8 Frames básicos no canal ainda podem existir 8 Frames básicos globais da NCU.

Frames específicos de canal

$P_CHBFR[n]

Através da variável de sistema $P_CHBFR[n] se pode ler e gravar os Frames básicos. Ao se gravar um quadro básico, este total encadeado não é ativado, a ativação somente ocorre com a execução de uma instrução G500, G54 ... G599. A variável serve especialmente como memória de gravações no Frame básico da HMI ou PLC. Estas variáveis Frame são salvas através do backup de dados.

Primeiro Frame básico no canal

Uma gravação na variável pré-definida $P_UBFR não ativa simultaneamente o quadro base com o índice de campo 0, a ativação somente é realizada com a execução de uma instrução G500, G54 G599. A variável também pode ser gravada e lida no programa.

$P_UBFR

$P_UBFR é idêntico ao $P_CHBFR[0]. Como padrão sempre existe um Frame básico no canal, de modo que a variável de sistema seja compatível com versões anteriores. Se não existir um quadro básico específico de canal, durante a gravação ou leitura será emitido o alarme "Frame: Instrução inválida".

6.8.2 Frames ativos no canal

Os Frames ativos no canal são especificados pelo programa de peça através das respectivas variáveis de sistema destes Frames. Aqui também pertencem os Frames de sistema. Através destas variáveis de sistema pode-se ler e gravar o atual Frame de sistema no programa de peça.

Atuais Frames ativos no canal

Vista geral

Atuais Frames de sistema	para:
$P_PARTFRAME	TCARR e PAROT

$P_SETFRAME	Definir valor real e contato de referência
$P_EXTFRAME	Deslocamento externo de ponto zero
$P_NCBFRAME[n]	Atuais Frames básicos globais da NCU
$P_CHBFRAME[n]	Atuais Frames básicos de canal
$P_BFRAME	Atual 1º Frame básico no canal
$P_ACTBFRAME	Frame básico total
$P_CHBFRMASK e $P_NCBFRMASK	Frame básico total
$P_IFRAME	Atual Frame ajustável
Atuais Frames de sistema	para:
$P_TOOLFRAME	TOROT e TOFRAME
$P_WPFRAME	Pontos de referência de peça
$P_TRAFRAME	Transformações
$P_PFRAME	Atual Frame programável
Atual Frame de sistema	para:
$P_CYCFRAME	Ciclos
P_ACTFRAME	Atual Frame total
Encadeamento de FRAMEs	O atual Frame é formado pelo Frame básico total

$P_NCBFRAME[n] Atuais Frames básicos globais da NCU

Através da variável de sistema P_NCBFRAME[n] podem ser lidos e gravados os atuais elementos de campo de Frames básicos globais. O Frame básico total resultante é processado pelo processo de gravação no canal.

O Frame alterado somente estará ativo no canal em que o Frame foi programado. Se o Frame deve ser alterado em todos canais de uma NCU, então $P_NCBFR[n] e $P_NCBFRAME[n] devem ser gravados simultaneamente. Os outros canais ainda devem ativar o Frame, por exemplo com G54. Ao gravar um Frame básico, será recalculado o Frame básico total.

$P_CHBFRAME[n] Atuais Frames básicos de canal

Através da variável de sistema $P_NCBFRAME[n] podem ser lidos e gravados os atuais elementos de campo de Frames básicos globais. O Frame básico total resultante é processado pelo processo de gravação no canal. Ao gravar um Frame básico, será recalculado o Frame básico total.

$P_BFRAME Atual 1º Frame básico no canal

Através da variável Frame pré-definida $P_BFRAME o atual Frame básico com o índice de campo 0 válido no canal, poderá ser lido e gravado no programa de peça. O Frame básico gravado é imediatamente processado.

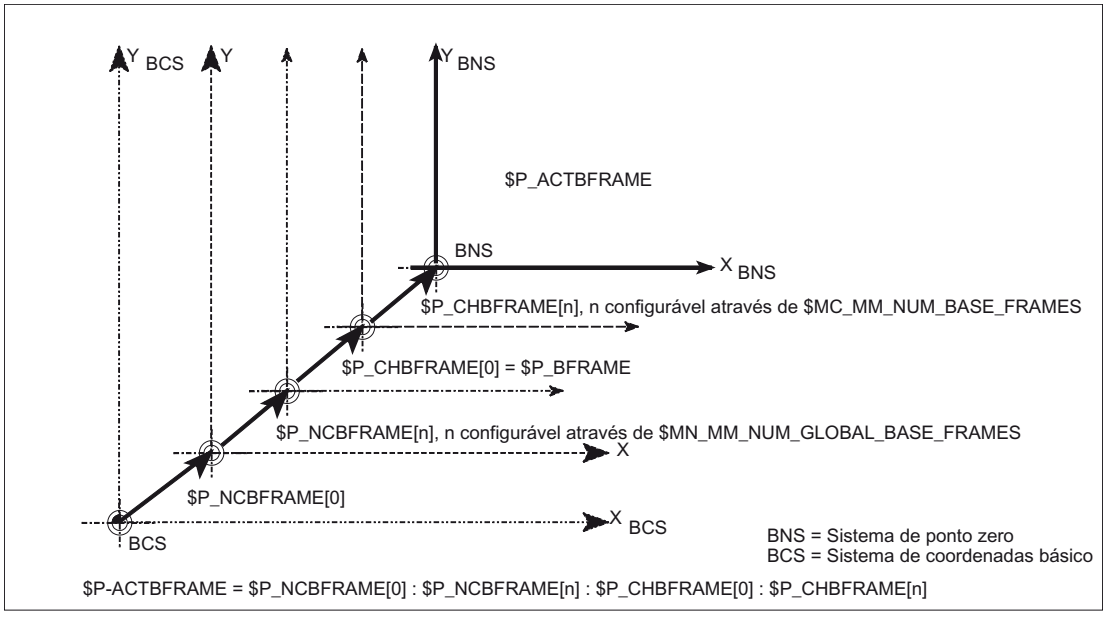
$P_BFRAME é idêntico ao $P_CHBFRAME[0]. Como padrão a variável de sistema sempre tem um valor válido. Se não existir um quadro básico específico de canal, durante a gravação ou leitura será emitido o alarme "Frame: Instrução inválida".

$P_ACTBFRAME Frame básico total

A variável $P_ACTBFRAME determina o Frame básico total encadeado. A variável somente pode ser lida.

$P_ACTFRAME equivale:

$P_NCBFRAME[0]: ... : $P_NCBFRAME[n]: $P_CHBFRAME[0]: ... : $P_CHBFRAME[n].



$P_CHBFRMASK e $P_NCBFRMASK Frame básico total

Através das variáveis de sistema $P_CHBFRMASK e $P_NCBFRMASK o usuário pode escolher quais quadro base ele quer adicionar ao cálculo do "total". As variáveis somente podem ser programadas uma vez no programa e lidas através do BTSS. O valor das variáveis é interpretado como máscara de Bit e indica qual elemento de campo do Frame básico de $P_ACTFRAME entra no cálculo.

Com $P_CHBFRMASK se pode especificar quais quadros base específicos de canal, e com $P_NCBFRMASK quais quadros base globais da NCU, serão processados.

Com a programação das variáveis é recalculado o Frame básico total e o Frame total. Após o Reset e a configuração básica o valor de $P_CHBFRMASK e $P_NCBFRMASK é o seguinte:

$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK

$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK

Exemplo:

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0]: $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0]: $P_CHBFRAME[4]
```

$P_IFRAME Atual Frame ajustável

Através da variável Frame pré-definida $P_IFRAME, o atual Frame ajustável, o que estiver válido no canal, pode ser lido e gravado no programa de peça. O Frame ajustável que está gravado é imediatamente processado.

No caso dos Frames ajustáveis globais da NCU, o Frame alterado apenas atua no canal em que foi programado. Se o quadro deve ser modificado para todos os canais de uma NCU, então $P_UIFR[n] e $P_IFRAME precisam ser gravados simultaneamente. Os outros canais ainda devem ativar o Frame correspondente, por exemplo com G54.

$P_PFRAME Atual Frame programável

$P_PFRAME é o quadro programável, que resulta da programação do TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR ou da atribuição do CTRANS, CROT, CMIRROR, CSCALE ao quadro programável.

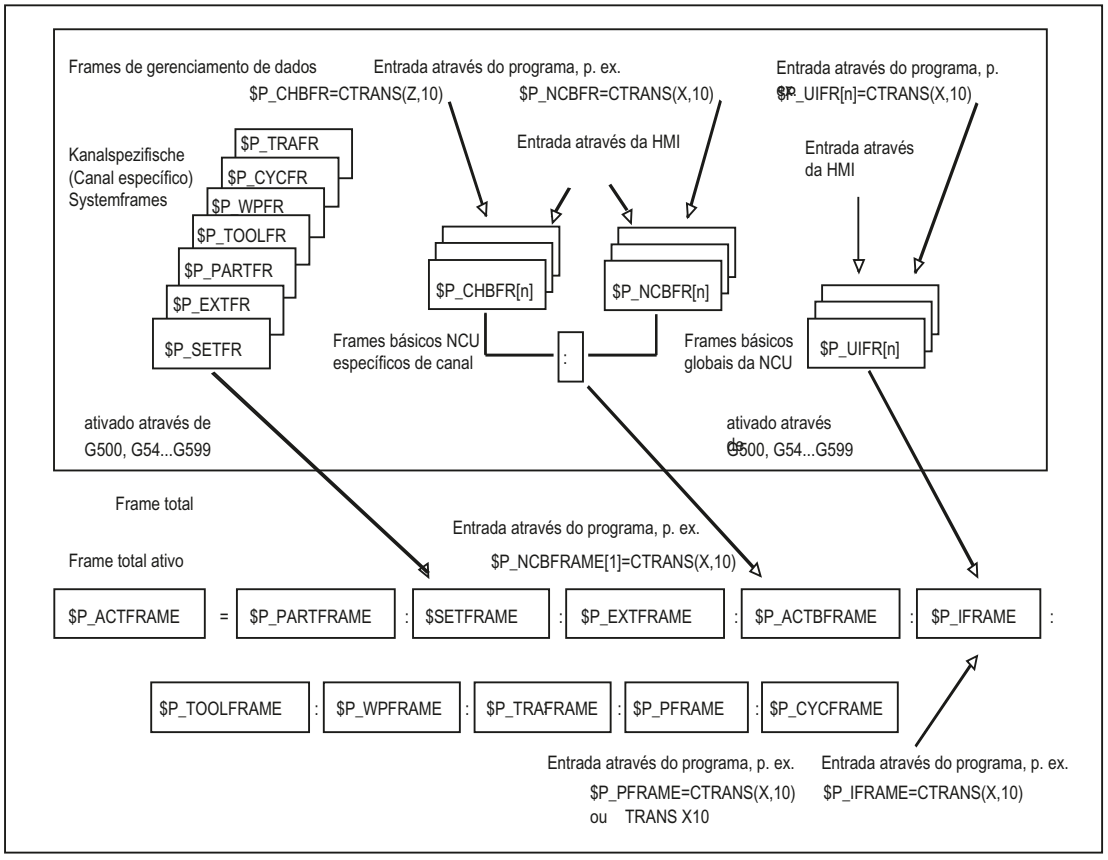
Atual variável programável que estabelece a relação entre o sistema ajustável do ponto zero (ENS) e o sistema de coordenadas da peça de trabalho (WCS).

P_ACTFRAME Atual Frame total

O atual Frame total $P_ACTFRAME resultante agora resulta como encadeamento de todos Frames básicos, do atual Frame ajustável e do Frame programável. O atual Frame sempre será atualizado quando um componente do Frame for alterado.

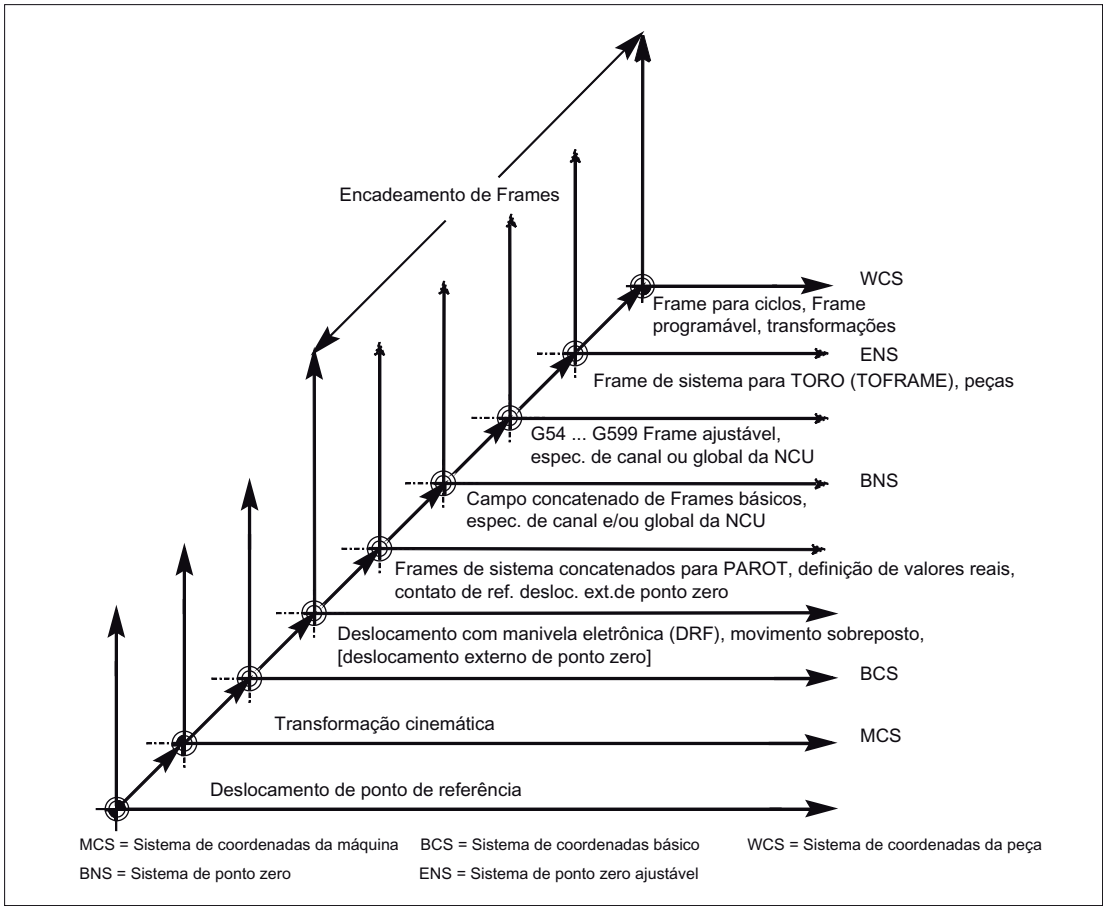
`$P_ACTFRAME` equivale:

```
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :
$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
```


Frames de gerenciamento de dados
Entrada através do programa, p. ex.
$P_CHBFR=CTRANS(Z,10)
$P_NCBFR=CTRANS(X,10)
Entrada através do programa, p.
$P_UIFR[n]=CTRANS(X,10)
Kanalspezifische
(Canal específico)
Systemframes
$P_TRAFR
$P_CYCFR
$P_WPFR
$P_TOOLFR
$P_PARTFR
$P_EXTFR
$P_SETFR
Entrada através da HMI
Entrada através
da HMI
$P_CHBFR[n]
$P_NCBFR[n]
$P_UIFR[n]
Frames básicos NCU
específicos de canal
Frames básicos
globais da NCU
ativado através de
G500, G54...G599
ativado através
G500, G54...G599
Frame total
Entrada através do programa, p. ex.
Frame total ativo
$P_NCBFRAME[1]=CTRANS(X,10)
$P_ACTFRAME
=
$P_PARTFRAME
$SETFRAME
$P_EXTFRAME
$P_ACTBFRAME
$P_IFRAME
$P_TOOLFRAME
$P_WPFRAME
$P_TRAFRAME
$P_PFRAME
$P_CYCFRAME
Entrada através do programa, p. ex.
$P_PFRAME=CTRANS(X,10)
ou TRANS X10
Entrada através do programa, p. ex.
$P_IFRAME=CTRANS(X,10)

Encadeamento de Frames

O atual Frame é formado a partir do Frame básico total, do Frame ajustável, do Frame de sistema e do Frame programável conforme o atual Frame total acima mencionado.



Transformações 7

7.1 Programação geral de tipos de transformação

Função geral

Para adaptação do comando em diversas cinemáticas de máquina existe a possibilidade de se programar a seleção dos tipos de transformações com parâmetros adequados. Através destes parâmetros pode ser acordado para a transformação selecionada tanto a orientação da ferramenta no espaço como os movimentos da orientação dos eixos rotativos.

Em transformações de três, quatro e cinco eixos, as indicações de posição programadas sempre estão relacionadas à ponta da ferramenta que é acompanhada de forma ortogonal à superfície de usinagem disponível no espaço. As coordenadas cartesianas são convertidas do sistema de coordenadas básico para o sistema de coordenadas da máquina e estão relacionadas aos eixos geométricos. Elas descrevem o ponto de trabalho. Os eixos rotativos virtuais descrevem as orientações da ferramenta no espaço e são programadas com TRAORI.

Na transformação cinemática as posições podem ser programadas no sistema de coordenadas cartesiano. O comando numérico transforma os movimentos de deslocamento do sistema de coordenadas cartesiano programados com TRANSMIT, TRACYL e TRAANG em movimentos de deslocamento dos eixos reais da máquina.

Programação

Transformações de três, quatro e cinco eixos TRAORI

A transformação de orientação acordada é ativada com o comando TRAORI e com os três possíveis parâmetros para número de transformação, vetor de orientação e Offset de eixo rotativo.

```
TRAORI(número de transformação, vetor de orientação, Offsets de eixo rotativo)
```

Transformações cinemáticas

Às transformações cinemáticas pertencem as transformações acordadas
`TRANSMIT(número de transformação)`

`TRACYL(diâmetro de trabalho, número de transformação)`

`TRAANG(ângulo do eixo inclinado, número de transformação)`

Desativar a transformação ativa

Com `TRAFOOF` se desativa a transformação que estiver ativa no momento.

Transformação de orientação

Transformações de três, quatro e cinco eixos TRAORI

Para uma usinagem otimizada de superfícies de formas espaciais na área de trabalho da máquina, as máquinas-ferramenta requerem eixos adicionais aos três eixos lineares X, Y e Z. Os eixos adicionais descrevem a orientação no espaço e a seguir serão chamados de eixos de orientação. Eles estão disponíveis como eixos de rotação em quatro tipos de máquina de diferentes cinemáticas.

1. Cabeçote orientável de dois eixos, p. ex. cabeçote de ferramentas cardânico com um eixo rotativo paralelo à um eixo linear e com mesa fixa de ferramenta.
2. Mesa giratória de dois eixos, p. ex. cabeçote orientável fixo com mesa giratória de ferramenta e dois eixos.
3. Cabeçote orientável de um eixo e mesa giratória de um eixo, p. ex. um cabeçote giratório com ferramenta rotativa com mesa giratória de ferramenta em torno de um eixo.
4. Cabeçote orientável de dois eixos e mesa giratória de um eixo, p. ex. uma mesa giratória de ferramenta em torno de um eixo e um cabeçote giratório com ferramenta rotativa em torno de si mesmo.

As **transformações de 3 e 4 eixos** são formas especiais da transformação de 5 eixos e são programadas de forma análoga às transformações de 5 eixos.

A "**transformação genérica de 3, 4, 5, 6 eixos**", com suas funções, cobre os eixos rotativos dispostos perpendicularmente e as transformações para o cabeçote de fresar cardânico, e pode ser ativada com TRAORI para qualquer outra transformação de orientação, assim como para estes quatro tipos de máquina. Na transformação genérica de 5 e 6 eixos, a orientação da ferramenta possui mais um terceiro grau de liberdade, onde a ferramenta pode ser girada em torno de seu próprio eixo em qualquer direção dentro do espaço.

Literatura: /FB3/ Manual de funções especiais; Transformações de 3 a 5 eixos (F2)

Posição básica da orientação da ferramenta independente da cinemática

ORIRESET

Se uma transformação de orientação estiver ativa com TRAORI, então com ORIRESET se pode especificar as posições básicas de até 3 eixos de orientação com os parâmetros opcionais A, B e C. A associação da seqüência dos parâmetros programados com os eixos rotativos é realizada de acordo com a seqüência dos eixos de orientação definida através da transformação. A programação do ORIRESET(A, B, C) faz com que os eixos de orientação se desloquem de modo linear e sincronizado de sua posição momentânea até a posição básica especificada.

Transformações cinemáticas

TRANSMIT e TRACYL

No caso das operações de fresamento em tornos, para a transformação acordada se pode

1. programar uma usinagem frontal na fixação rotativa com TRANSMIT ou
2. uma usinagem de um número qualquer de ranhuras em corpos cilíndricos com TRACYL

.

TRAANG

Por exemplo, se o eixo de penetração da tecnologia de retificação também deve penetrar de modo inclinado, então se pode programar um ângulo parametrizável com TRAANG para a transformação acordada.

Deslocamento PTP cartesiano

O "Deslocamento PTP cartesiano" também pertence às transformações cinemáticas, onde se pode programar até 8 diferentes posições articuladas STAT= . As posições são programadas no sistema de coordenadas cartesiano, onde o movimento da máquina é realizado em coordenadas da máquina.

Literatura:
/FB2/ Manual de funções ampliadas; Transformação cinemática (M1)

Transformações encadeadas

Sempre podem ser ativadas duas transformações sucessivamente. Na segunda transformação encadeada desse modo se assumem as partes de movimentos dos eixos da primeira transformação.

Como primeira transformação são possíveis:

- Transformação de orientação TRAORI
- Transformação polar TRANSMIT
- Transformação cilíndrica TRACYL
- Transformação de eixo inclinado TRAANG

A segunda transformação deve ser de eixo inclinado TRAANG

7.1.1 Movimentos de orientação nas transformações

Movimentos de deslocamento e movimentos de orientação

Os movimentos de deslocamentos das orientações programáveis dependem primeiramente do tipo de máquina. Na transformação de três, quatro e cinco eixos com TRAORI são os eixos rotativos ou os eixos lineares orientáveis que descrevem os movimentos de orientação da ferramenta.

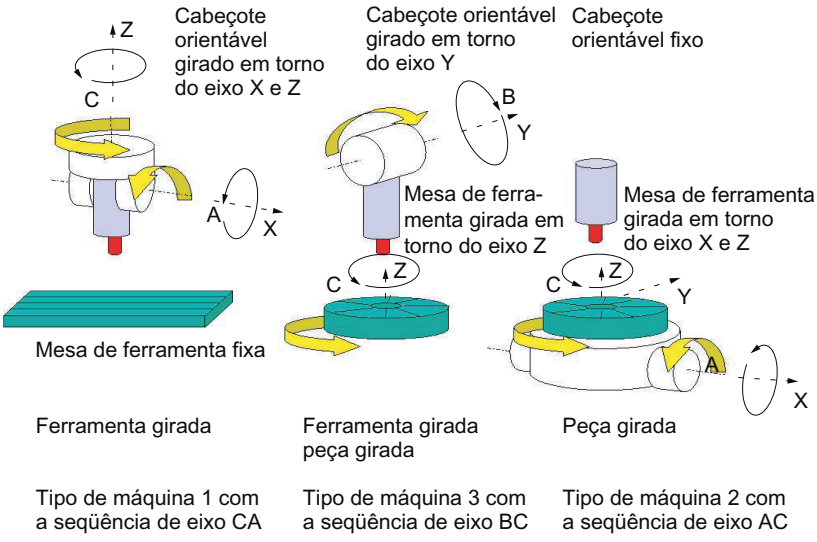
As alterações nas posições dos eixos rotativos que participam na transformação da orientação geram movimentos de compensação nos demais eixos da máquina. A posição da ponta da ferramenta permanece inalterada com isso.

Os movimentos de orientação da ferramenta podem ser programados através dos identificadores de eixo rotativo A…, B…, C… dos eixos virtuais, e dependendo da aplicação, através da indicação de ângulos eulerianos e ângulos RPY ou através de vetores de direção e vetores normais de superfície, vetores normalizados para o eixo de rotação de um cone ou para orientação intermediária em uma superfície periférica cônica.

Na transformação cinemática com TRANSMIT, TRACYL e TRAANG o comando numérico transforma os movimentos de deslocamento programados do sistema de coordenadas cartesiano em movimentos de deslocamento dos eixos reais da máquina.

Cinemática da máquina na transformação de três, quatro e cinco eixos TRAORI

Se pode girar a ferramenta ou a mesa de ferramenta mediante até dois eixos rotativos. Uma combinação de um cabeçote orientável de um eixo e a mesa giratória também é possível.



Tipo de máquina	Programação da orientação
Transformação de três eixos para tipos de máquina 1 e 2	Programação da orientação de ferramenta apenas no plano que estiver perpendicular ao eixo rotativo. Existem **dois** eixos translatórios (eixos lineares) e **um** eixo rotatório (eixo rotativo).
Transformação de quatro eixos para tipos de máquina 1 e 2	Programação da orientação de ferramenta apenas no plano que estiver perpendicular ao eixo rotativo. Existem **três** eixos translatórios (eixos lineares) e **um** eixo rotatório (eixo rotativo).
Transformação de cinco eixos para tipo de máquina 3 Cabeçote orientável de um eixo e mesa giratória de um eixo	Programação da transformação de orientação. Cinemática com **três** eixos lineares e **dois** eixos rotativos ortogonais. Os eixos rotativos são paralelos à dois dos três eixos lineares. O primeiro eixo rotativo é movimentado por dois eixos lineares cartesianos. Ele gira o terceiro eixo linear com a ferramenta. O segundo eixo rotativo gira a peça de trabalho.

Transformações genéricas de 5 e 6 eixos

Tipo de máquina	Programação da transformação de orientação
Transformação genérica de cinco e seis eixos para tipo de máquina 4 Cabeçote orientável de dois eixos com ferramenta rotativa em torno de si mesma e mesa giratória de um eixo	Programação da transformação de orientação. Cinemática com **três** eixos lineares e **três** eixos rotativos ortogonais. Os eixos rotativos são paralelos à dois dos três eixos lineares. O primeiro eixo rotativo é movimentado por dois eixos lineares cartesianos. Ele gira o terceiro eixo linear com a ferramenta. O segundo eixo rotativo gira a peça de trabalho. A orientação básica da ferramenta pode ser programada através de um giro adicional em torno de si mesma com o ângulo de giro THETA.

Na chamada da "Transformação genérica de três, quatro e cinco eixos" também se pode passar a orientação básica da ferramenta. Não se pode aplicar mais as restrições relacionadas às direções dos eixos rotativos. Se os eixos rotativos não estiverem exatamente perpendiculares entre si ou se os eixos rotativos existentes não estiverem exatamente paralelos com os eixos lineares, a "Transformação genérica de cinco e seis eixos" pode oferecer melhores resultados de orientação da ferramenta.

Transformações cinemáticas TRANSMIT, TRACYL e TRAANG

Para operações de fresamento em tornos ou para um eixo que avança inclinado na operação de retificação se aplicam as seguintes disposições de eixo (em função da transformação em casos padrão):

TRANSMIT	Ativação da transformação polar
usinagem frontal na fixação rotativa	um eixo rotativo um eixo de penetração perpendicular ao eixo de rotação um eixo longitudinal paralelo ao eixo de rotação

TRACYL	Ativação da transformação de superfície cilíndrica
Usinagem de qualquer ranhura que percorre no corpo cilíndrico	um eixo rotativo um eixo de penetração perpendicular ao eixo de rotação um eixo longitudinal paralelo ao eixo de rotação

TRAANG	Ativação da transformação de eixos inclinados
Usinagem com eixo de penetração inclinado	um eixo rotativo um eixo de penetração com ângulo parametrizável um eixo longitudinal paralelo ao eixo de rotação

Deslocamento PTP cartesiano

O movimento da máquina é realizado em coordenadas da máquina e se programa com:

TRAORI	Ativação da transformação
PTP avançar de ponto a ponto	Aproximação da posição em sistema de coordenadas cartesiano (MCS)
CP	Movimento de percurso dos eixos cartesianos em (BCS)
STAT	A posição das articulações está em função da transformação
TU	O ângulo com que os eixos percorrem o curso mais curto

Deslocamento PTP na transformação de 5 e 6 eixos

O movimento da máquina é realizado em coordenadas da máquina e a orientação da ferramenta podem tanto ser programada com posições de eixo rotativo como com vetores eulerianos e vetores RPY independentes de cinemática ou com vetores de direção.

Desse modo é possível realizar a interpolação de eixos rotativos, interpolação de vetores com interpolação de grande circunferência ou interpolação do vetor de orientação sobre uma superfície periférica cônica.

Exemplo de transformação de três a cinco eixos em um cabeçote de fresar cardânico

A máquina-ferramenta possui pelo 5 eixos, destes

- Três eixos translatórios para movimentos em linhas retas, que movimentam o ponto de trabalho em qualquer posição na área de trabalho.
- Dois eixos orientáveis rotativos que estão posicionados conforme um ângulo projetável (frequentemente 45 graus) permitem a realização da orientação da ferramenta no espaço, restrita na posição de 45 graus sobre uma meia esfera.

7.1.2 Vista geral da transformação de orientação TRAORI

Possíveis tipos de programação em conjunto com o TRAORI

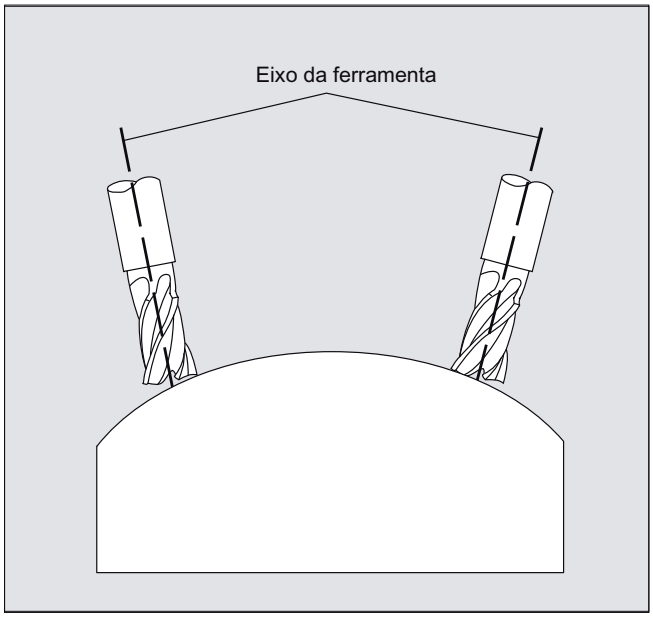
Tipo de máquina	Programação com transformação TRAORI ativa
Tipos de máquina 1, 2 ou 3 Cabeçote orientável de dois eixos ou mesa giratória de dois eixos ou uma combinação de cabeçote orientável de um eixo e mesa giratória de um eixo.	A seqüência de eixos de orientação e o sentido de orientação da ferramenta podem ser projetados **relacionados à máquina** através de dados de máquina e em função da cinemática da máquina ou projetados **relacionados à ferramenta** com orientação programável e independente da cinemática da máquina Os sentidos de giro dos eixos de orientação no sistema de referência são programados com: - Sistema de referência ORIMKS = Sistema de coordenadas da máquina - Sistema de referência ORIWKS = Sistema de coordenadas da peça O ajuste básico é ORIWKS. Programação dos eixos de orientação com: A, B, C direto das posições de eixo da máquina A2, B2, C2 programação de ângulos de eixos virtuais com - ORIEULER através de ângulo euleriano (Standard) - ORIRPY através de ângulo RPY - ORIVIRT1 através de eixos de orientação virtuais de 1ª definição - ORIVIRT2 através de eixos de orientação virtuais de 2ª definição com diferença no tipo de interpolação: **Interpolação linear** - ORIAXES de eixos de orientação dos eixos da máquina **Interpolação de grande circunferência** (interpolação do vetor de orientação) - ORIVECT de eixos de orientação Programação dos eixos de orientação através da indicação A3, B3, C3 dos componentes de vetor (normal de sentido/superfície) Programação da orientação de ferramenta resultante A4, B4, C4 do vetor normal de superfície no início do bloco A5, B5, C5 do vetor de normal de superfície no fim do bloco Ângulo de avanço LEAD para orientação da ferramenta Ângulo lateral TILT para orientação de ferramenta

Tipo de máquina	Programação com transformação TRAORI ativa
	Interpolação do vetor de orientação sobre uma superfície periférica cônica Alterações de orientação sobre uma superfície periférica cônica **qualquer no espaço** através da interpolação: - ORIPLANE no plano (interpolação de grande circunferência) - ORICONCW sobre uma superfície periférica cônica em sentido horário - ORICONCCW sobre uma superfície periférica cônica em sentido anti-horário Vetores de direção A6, B6, C6 (eixo de rotação do cone) - Interpolação OICONIO sobre uma superfície periférica cônica com: Vetores intermediários A7, B7, C7 (orientação inicial e final) ou - ORICONTO sobre uma superfície periférica cônica de transição tangencial Alterações de orientação relacionadas **à uma trajetória** com - Especificação ORICURVE do movimento de dois pontos de contato através de Polinômios de orientação de até 5º grau PO[XH]=(xe, x2, x3, x4, x5) Polinômios de orientação de até 5º grau PO[YH]=(ye, y2, y3, y4, y5) Polinômio de orientação de até 5º grau PO[ZH]=(ze, z2, z3, z4, z5) - A suavização do decurso de orientação ORIPATHS com fase de reorientação A8, B8, C8 da ferramenta corresponde ao(à): Direção e distância de curso da ferramenta no movimento de retração
Tipos de máquina 1 e 3 Os demais tipos de máquina com giro adicional da ferramenta em torno de si mesma requerem um 3º eixo rotativo Transformação de orientação como na transformação genérica de 6 eixos. Rotações do vetor de orientação.	Programação **das rotações** da orientação da ferramenta com ângulo de avanço LEAD relativo ao vetor normal de superfície PO[PHI] Programação de um polinômio de até 5º grau TILT Ângulo lateral da rotação em torno da tangente da trajetória (sentido Z) PO[PSI] Programação de um polinômio de 5º grau THETA Ângulo de rotação (rotação em torno da direção da ferramenta em Z) THETA= Valor a ser alcançado no final do bloco THETA=AC(...) Comutar por blocos para dimensões absolutas THETA=IC(...) Comutar por blocos para dimensões incrementais THETA=Θ_e Interpolar ângulo programado G90/G91 PO[THT]=(..) Programação de um polinômio de 5º grau Programação do vetor de rotação - ORIROTA Rotação absoluta - ORIROTR Vetor de rotação relativo - ORIROTT Vetor de rotação tangencial
Orientação relativa à trajetória para alterações de orientação relativas à trajetória ou rotação do vetor de rotação tangencial à trajetória	Alterações de orientação **relativas à trajetória** com - Orientação de ferramenta ORIPATH relativa à trajetória - ORIPATHS adicionalmente em uma dobra no decurso da orientação Programação do vetor de rotação - Vetor de rotação tangencial ORIROTC, rotação com a tangente da trajetória

7.2 Transformação de três, quatro e cinco eixos (TRAORI)

7.2.1 Relações gerais do cabeçote de ferramenta cardânico

Para obter as condições ideais de corte na usinagem de superfícies curvadas, se deve permitir a variação do ângulo de ataque da ferramenta.

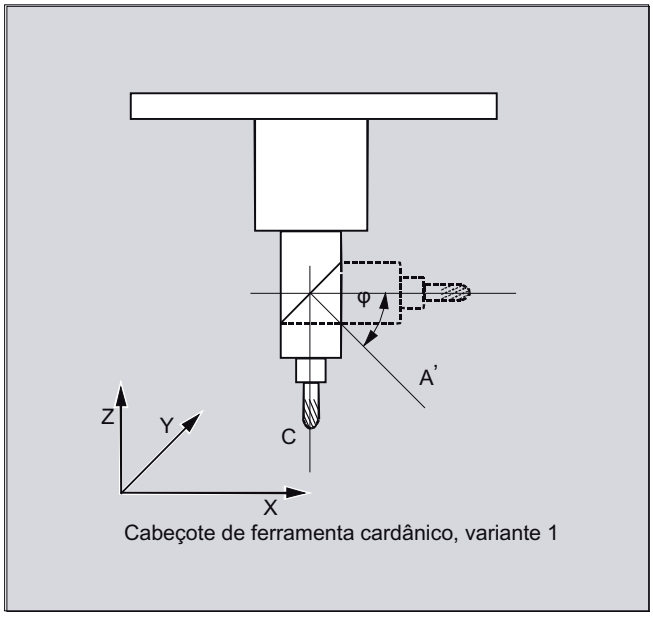


Com qual forma construtiva de máquina isso é obtido, está armazenado nos dados do eixo.

Transformação de 5 eixos

Cabeçote de ferramenta cardânico

Aqui estão definidos três eixos lineares (X, Y, Z) e dois eixos de orientação (C, A) do ângulo de ataque e ponto de trabalho da ferramenta. Um dos dois eixos de orientação está definido como eixo inclinado, aqui no exemplo A', como na maioria dos casos, na disposição de 45°.

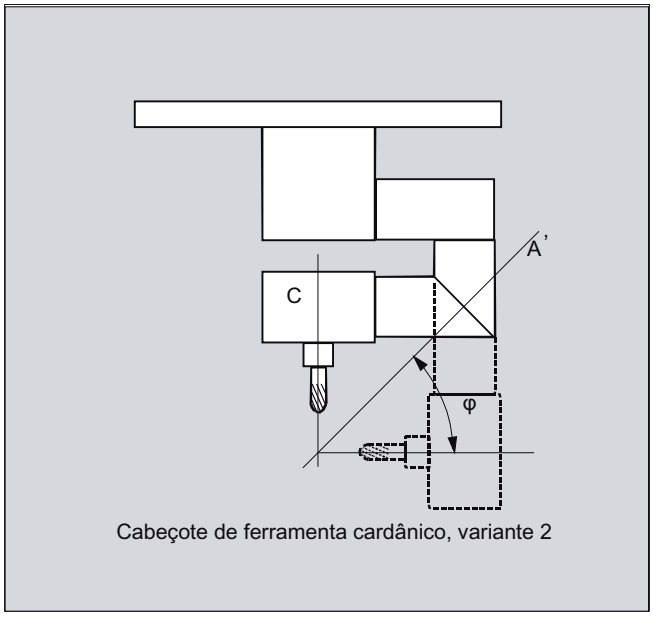


Cabeçote de ferramenta cardânico, variante 1

Nos exemplos aqui apresentados temos as disposições no exemplo da cinemática de máquina CA com o cabeçote de ferramenta cardânico!

Fabricante da máquina

A sequência dos eixos de orientação e o sentido de orientação da ferramenta pode ser ajustado em função da cinemática da máquina através dos dados de máquina.



Cabeçote de ferramenta cardânico, variante 2

Neste exemplo o A' encontra-se a um ângulo φ em relação ao eixo X

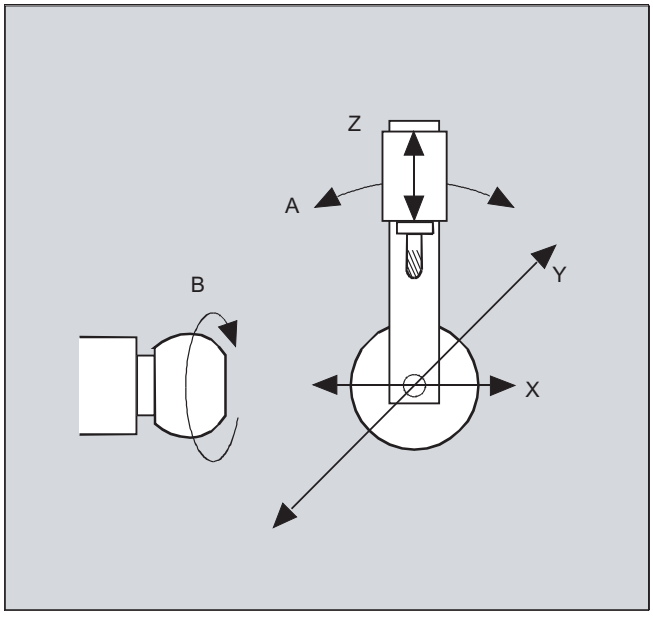
Normalmente são aplicadas as seguintes possíveis relações:

A' encontra-se a um ângulo φ em relação ao	Eixo X
B' encontra-se a um ângulo φ em relação ao	Eixo Y
C' encontra-se a um ângulo φ em relação ao	Eixo Z

O ângulo φ pode ser configurado na faixa de 0° até +89° através de dados de máquina.

Com eixo linear giratório

Aqui trata-se de uma disposição com peça de trabalho em movimento e ferramenta em movimento. A cinemática é formada a partir de três eixos lineares (X, Y, Z) e dois eixos de rotação dispostos ortogonalmente. Por exemplo, o primeiro eixo rotativo é movimentado por uma unidade de avanço cruzada com dois eixos lineares, a ferramenta está paralela ao terceiro eixo linear. O segundo eixo de rotação gira a peça de trabalho. O terceiro eixo linear (eixo que gira) está no plano da unidade de avanço cruzada.



A sequência dos eixos rotativos e o sentido de orientação da ferramenta pode ser ajustado em função da cinemática da máquina através dos dados de máquina.

São aplicadas as seguintes possíveis relações:

Eixos:	Sequências de eixos:
1. Eixo rotativo	A A B B C C
2. Eixo rotativo	B C A C A B
Eixo linear girado	Z Y Z X Y X

Para mais explicações sobre a configuração de sequências de eixos para o sentido de orientação da ferramenta, veja

Literatura: /FB3/ Manual de funcionamento, funções especiais; Transformações de 3 a 5 eixos (F2), Capítulo Cabeçote de fresar cardânico, "Parametrização".

7.2.2 Transformação de três, quatro e cinco eixos (TRAORI)

O usuário pode configurar dois ou três eixos translatórios e um eixo rotatório. As transformações partem do princípio de que o eixo rotatório esteja em posição ortogonal ao plano de orientação.

A orientação da ferramenta somente é possível no plano que estiver perpendicular ao eixo rotatório. A transformação dá suporte aos tipos de máquina com ferramenta articulada e peça de trabalho móvel.

A projeção e programação de transformações de três e quatro eixos é realizada de forma análoga às transformações de cinco eixos.

Literatura:
Manual de funcionamento, funções especiais; Transformações múltiplas (F2)

Sintaxe

```
TRAORI(<n>)
TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)
TRAFOOF
```

Significado

`TRAORI`:	Ativa a primeira transformação de orientação especificada	
`TRAORI(<n>)`:	Ativa a transformação de orientação especificada com n	
`<n>`:	Número da transformação	
	Valor:	1 ou 2
	Exemplo: TRAORI(1) ativa a transformação de orientação 1	
`<X>,<Y>,<Z>`:	Componente do vetor de orientação em que a ferramenta aponta.	
`<A>,<B>`:	Offset programável para os eixos rotativos	
`TRAFOOF`:	Desativação da transformação	

Orientação da ferramenta

Em função do sentido de orientação selecionado da ferramenta, no programa NC o plano de trabalho ativo (G17, G18, G19) deve ser ajustado de modo que a correção do comprimento da ferramenta atue no sentido da orientação da ferramenta.

Indicação

Após a ativação da transformação as posições (Z, Y, Z) sempre estão relativas à ponta da ferramenta. Alteração das posições dos eixos rotativos envolvidos na transformação que gera os tais movimentos de compensação dos demais eixos da máquina, de modo que a posição da ponta da ferramenta permaneça inalterada.

A transformação da orientação sempre está alinhada da ponta da ferramenta para a fixação da ferramenta.

Offset para eixos de orientação

Na ativação da transformação da orientação se pode programar diretamente um deslocamento adicional para eixos de orientação.

Os parâmetros podem ser omitidos se na programação for mantida a sequência correta.

Exemplo:

`TRAORI(, , , ,A,B)` ; se apenas um único Offset deve ser especificado

Como alternativa à programação direta, o Offset adicional para eixos de orientação também pode ser transferido automaticamente do atual deslocamento do ponto zero ativo. A transferência é configurada através de dados de máquina.

Exemplos

```
TRAORI(1,0,0,1)          ; A orientação básica da ferramenta aponta para o sentido Z
TRAORI(1,0,1,0)          ; A orientação básica da ferramenta aponta para o sentido Z
TRAORI(1,0,1,1)          ; A orientação básica da ferramenta aponta para o sentido Y/
                         Z (corresponde a posição de 45°)
```

7.2.3 Variantes da programação da orientação e posição básica (ORIRESET)

Programação de orientação da orientação de ferramenta com TRAORI

Em conexão com uma transformação programável de orientação TRAORI, é possível programar, adicionalmente aos eixos lineares X, Y, Z, também através dos identificadores de eixo rotativo A.., B..., C... posições de eixos ou eixos virtuais com ângulos ou componentes de vetor. Para eixos de orientação e eixos de máquina são possíveis diversos tipos de interpolação. Independentemente de quais polinômios de orientação PO[ângulo] e polinômios de eixo PO[eixo] estão ativos no momento, se pode programar diversos tipos diferentes de polinômio como G1, G2, G3, CIP ou POLY.

A alteração da orientação da ferramenta também pode ser programada através de vetores de orientação. Neste caso, a orientação final de cada bloco pode ser realizada através de programação direta do vetor ou através da programação das posições de eixo rotativo.

Variantes da programação de orientação em transformações de três a cinco eixos

As seguintes variantes da programação de orientação se excluem mutuamente.

`A, B, C`	Indicação direta das posições de eixos rotativos
`A2, B2, C2`	Programação de ângulos de eixos virtuais através de ângulos Euler ou ângulos RPY
`A3, B3, C3`	Indicação dos componentes de vetor
`LEAD, TILT`	Indicação dos ângulos de avanço e laterais em relação à trajetória e superfície
`A4, B4, C4` `A5, B5, C5`	Vetor normal de superfície no início e no fim do bloco

A6, B6, C6 A7, B7, C7	Interpolação do vetor de orientação sobre uma superfície cônica periférica
A8, B8, C8	Reorientação da ferramenta, direção e curso do movimento de elevação

Avançar até a posição básica da orientação da ferramenta (ORIRESET)

Através de ORIRESET(...), os eixos de orientação da respectiva cinemática da máquina são deslocados de modo linear e síncrono da sua posição atual para a posição básica programada. Se a posição básica para um eixo não estiver programada, utiliza-se a posição definida no dado correspondente de máquina $MC_TRAFO5_ROT_AX_OFFSET_1/2.

Frames ativos dos eixos rotativos não são considerados.

Exemplos para cinemática CA da máquina (nomes dos eixos de canal C, A)

Comandos	Descrição
ORIRESET(90, 45)	Eixo C: 90° Eixo A: 45°
ORIRESET(, 30)	Eixo C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] Eixo A: 30°
ORIRESET( )	Eixo C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] Eixo A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

Exemplos para cinemática CAC da máquina (nomes dos eixos de canal C, A, B)

Comandos	Descrição
ORIRESET(90, 45, 90)	Eixo C: 90° Eixo A: 45° Eixo B: 90°
ORIRESET( )	Eixo C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] Eixo A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1] Eixo B: $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]

Indicação

O avanço até a posição básica da orientação da ferramenta com ORIRESET(...) somente é permitido com a transformação de orientação TRAORI(...) ativa.

Programação das rotações LEAD, TILT e THETA

Ângulo de avanço LEAD ângulo lateral TILT

As rotações da orientação da ferramenta são programadas na transformação de três a cinco eixos como ângulo de avanço LEAD e o ângulo lateral TILT.

Ângulo de rotação THETA

Numa transformação **com terceiro eixo rotativo** é possível programar, tanto para a orientação com componentes de vetor quanto com programação dos ângulos LEAD, TILT, a rotação da ferramenta em torno de seu próprio eixo com o ângulo de giro THETA.

7.2.4 Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT)

Para a programação da orientação da ferramenta existem as seguintes opções:

1. Programação direta do movimento dos eixos rotativos. A alteração da orientação sempre é realizada no sistema de coordenadas básico ou no sistema de coordenadas da máquina. Os eixos de orientação são movimentados como eixos sincronizados.
2. Programação em ângulos eulerianos ou RPY conforme definição angular através do `A2, B2, C2`.
3. Programação do vetor de direção através do `A3, B3, C3`. O vetor de direção aponta para da ponta da ferramenta no sentido da fixação da ferramenta.
4. Programação do vetor normal de superfície no início do bloco com `A4, B4, C4` e no fim do bloco com `A5, B5, C5` (fresamento de topo).
5. Programação através de ângulo de avanço `LEAD` e ângulo lateral `TILT`
6. Para programação do eixo de rotação do cone como vetor normalizado através do `A6, B6, C6` ou da orientação intermediária sobre a superfície periférica cônica através do `A7, B7, C7`,
   veja o capítulo "Programação da orientação ao longo da superfície periférica cônica (ORIPLANE, ORICONxx)".
7. Para programação da reorientação, direção e distância de curso da ferramenta durante o movimento de retração através do `A8, B8, C8`,
   veja o capítulo "Suavização do decurso de orientação (ORIPATHS A8=, B8=, C8=)"

Indicação

Em todos casos a programação da orientação somente é permitida se uma transformação de orientação estiver ativada.

Vantagem: Estes programas podem ser transferidos para qualquer cinemática de máquina.

Definição da orientação da ferramenta através de comando G

Indicação

Fabricante da máquina

Através do dado da máquina se pode comutar entre ângulos eulerianos ou ângulos RPY. De acordo com as respectivas configurações de dados de máquina, uma comutação é possível tanto dependente como independente do comando G ativo do grupo 50. Estão disponíveis as seguintes opções de configuração:

1. Se os dois dados de máquina estiverem ajustados com valor zero para a definição dos eixos de orientação e definição do ângulo de orientação através de comando G:
   Os ângulos programados com `A2, B2, C2` podem ser interpretados como ângulos eulerianos ou como ângulos RPY **em função do dado de máquina** da definição de ângulos da programação de orientação.
2. Se o dado de máquina para a definição dos eixos de orientação através de comando G estiver ajustado com valor um, a comutação é realizada
   em função do comando G ativo do grupo 50:
   Os ângulos programados com `A2, B2, C2` são interpretados de acordo com um dos comandos G ativos `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS` e `ORIPY2` . Os valores programados com os eixos de orientação também são interpretados como ângulo de orientação de acordo com o comando G ativo do grupo 50.
3. Se o dado de máquina para a definição do ângulo de orientação através de comando G for ajustado com valor um e o dado de máquina para a definição dos eixos de orientação através de comando G for ajustado com valor zero, a comutação é realizada de forma **independente** do comando G ativo do grupo 50:
   Os ângulos programados com `A2, B2, C2` são interpretados de acordo com um dos comandos G ativos `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2 ORIAXPOS` e `ORIPY2` . Os valores programados com os eixos de orientação sempre são interpretados como posições de eixo rotativo independentemente do comando G ativo do grupo 50.

Sintaxe

Posições dos eixos rotativos

```
G1 X<valor> Y<valor> Z<valor> A<valor> B<valor> C<valor>
```

Ângulo euleriano

```
G1 X<valor> Y<valor> Z<valor> A2=<valor> B2=<valor> C2=<valor>
```

Vetor de direção

```
G1 X<valor> Y<valor> Z<valor> A3=<valor> B3=<valor> C3=<valor>
```

Vetor normal de superfície no início do bloco

```
G1 X<valor> Y<valor> Z<valor> A4=<valor> B4=<valor> C4=<valor>
```

Vetor normal de superfície no fim do bloco

```
G1 X<valor> Y<valor> Z<valor> A5=<valor> B5=<valor> C5=<valor>
```

Ângulo de avanço

`LEAD=<valor>`

Ângulo lateral

`TILT=<valor>`

Significado

`G1`:	Interpolação linear
`X, Y, Z`:	Posições dos eixos lineares
`A, B, C`:	Posições dos eixos rotativos
`A2=, B2=, C2=`:	Programação de ângulos (ângulo Euler ou ângulo RPY)
`A3=, B3=, C3=`:	Vetor de direção em coordenadas X, Y e Z do WCS
`A4=, B4=, C4=`:	Vetor normal de superfície no início do bloco em coordenadas X, Y e Z do WCS
`A5=, B5=, C5=`:	Vetor normal de superfície no fim do bloco em coordenadas X, Y e Z do WCS
`LEAD= `:	Ângulo de avanço [1)]
`TILT=`:	Ângulo lateral [1)]
1) A interpretação das indicações dos ângulos depende do ajuste em MD21094 $MC_ORIPATH_MODE	

Outras informações

Normalmente os programas de 5 eixos são gerados a partir de sistemas CAD/CAM e não são especificados no comando. Por isso que as seguintes explicações estão voltadas principalmente aos programadores de pós-processadores.

Estão disponíveis os seguintes comandos para a programação de orientação:

Comando	Significado
`ORIEULER`:	Ângulo Euler com sequência de rotação ZX'Z''
`ORIRPY`:	Ângulo RPY com sequência de rotação XY'Z''
`ORIRPY2`:	Angulo RPY com sequência de rotação ZY'X''
`ORIVIRT1`:	Eixos virtuais de orientação com sequência de rotação livremente definível através de: MD21120 $MC_ORIAX_TURN_TAB_1
`ORIVIRT2`:	Eixos virtuais de orientação com sequência de rotação livremente definível através de: MD21130 $MC_ORIAX_TURN_TAB_2
`ORIAXPOS`:	Eixos virtuais de orientação com posições de eixo rotativo

Indicação

Através de dados de máquina podem ser definidos diversas variantes pelo fabricante da máquina. Observe as instruções do fabricante da máquina.

Programação em ângulos Euler ORIEULER, sequência de rotação Z X' Z"

Os valores programados na programação de orientação ORIEULER com A2, B2, C2 são interpretados como ângulos Euler (em graus).

O novo vetor de orientação resulta através das seguintes três rotações do vetor original de orientação

1. com o eixo rotativo A2 em torno do eixo da coordenada Z
2. com o eixo rotativo B2 em torno do novo eixo da coordenada X'
3. com o eixo rotativo C2 em torno do novo eixo da coordenada Z"

Neste caso o valor de C2 (rotação em torno do novo eixo Z) não tem significado e não precisa ser programado.

Programação em ângulos RPY ORIRPY, sequência de rotação X Y' Z"

Os valores programados na programação de orientação ORIRPY com A2, B2, C2 são interpretados como ângulos RPY (em graus) com a sequência de rotação X Y' Z".

Indicação

Contrastando com a programação com ORIEULER, com ORIRPY todos os três valores influenciam o vetor de orientação.

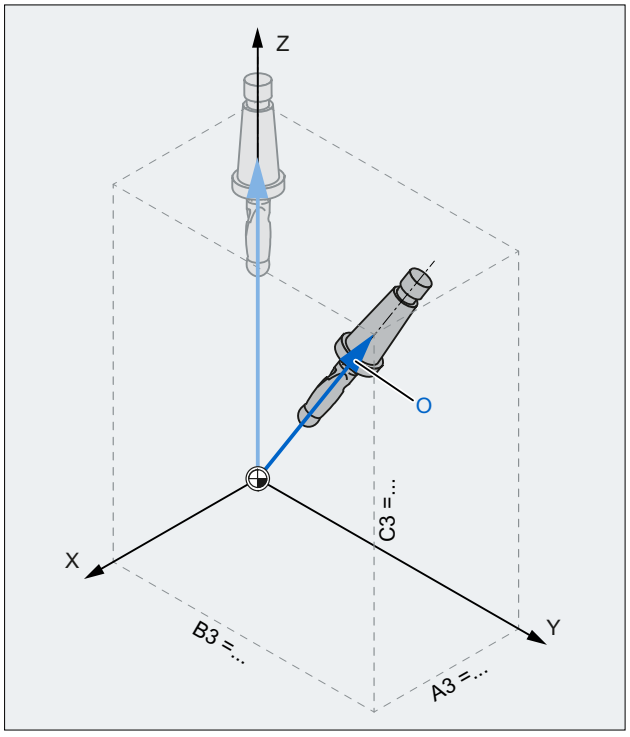
O novo vetor de orientação resulta através das seguintes três rotações do vetor original de orientação

1. com o eixo rotativo A2 em torno do eixo de coordenada X
2. com o eixo rotativo B2 em torno do novo eixo de coordenada Y'
3. com o eixo rotativo C2 em torno do novo eixo de coordenada Z"

Programação do vetor de direção

Os componentes do vetor de direção são programados com A3, B3, C3. O vetor aponta para a direção da fixação da ferramenta; o comprimento do vetor é irrelevante neste caso.

Os componentes não programados são definidos igual a zero.



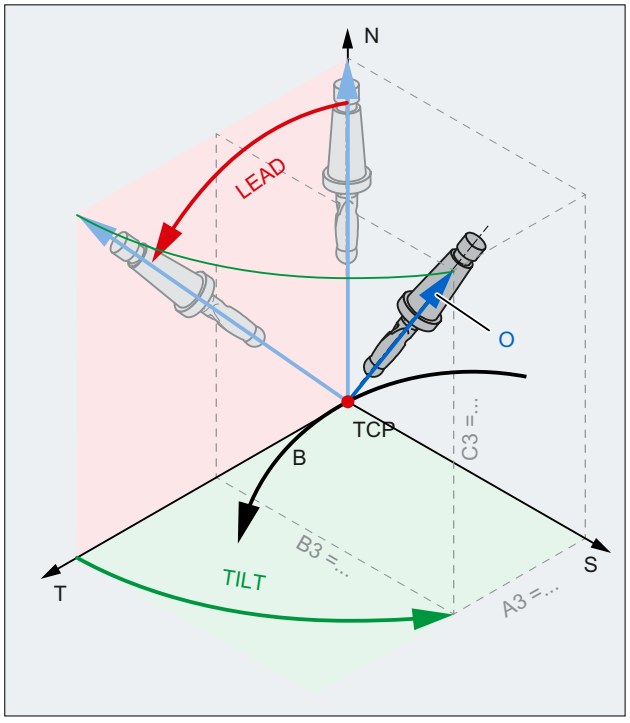
X, Y, Z	Eixos de coordenada do WCS
A3, B3, C3	Componentes do vetor de direção
O	vetor de orientação

Esquema 7-1 Programação do vetor de direção

Programação da orientação da ferramenta com LEAD e TILT

A orientação da ferramenta resultante é obtida a partir do(a):

- Tangente da trajetória
- Vetor da normal à superfície
  No início do bloco `A4, B4, C4` e no final do bloco `A5, B5, C5`
- Ângulo de avanço `LEAD`
  Ângulo no plano fixado pela tangente da trajetória e pelo vetor normal à superfície
- Ângulo lateral `TILT` no fim do bloco
  Ângulo no plano perpendicular à tangente da trajetória relativa ao vetor normal à superfície



T	Tangente da trajetória
S	Perpendicular à tangente da trajetória
N	Normal de superfície
B	de percurso
TCP	Tool Center Point
O	vetor de orientação

Esquema 7-2 Programação de LEAD TILT

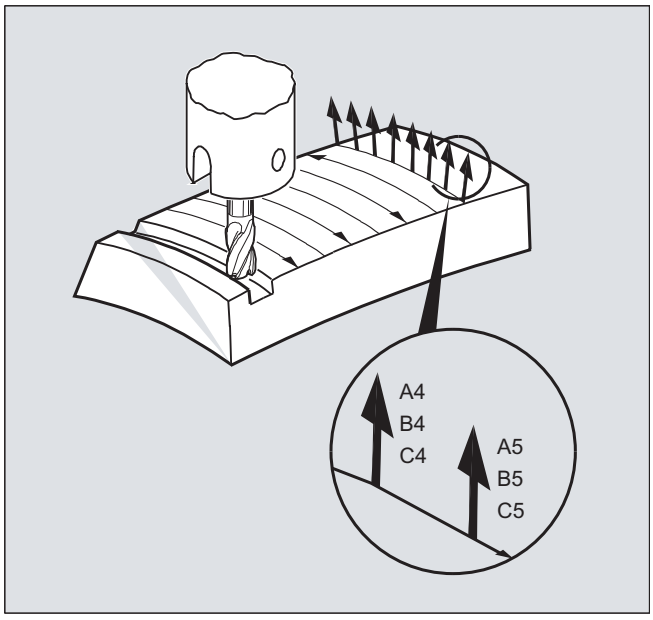
Indicação

Comportamento com cantos internos com 3D-WZK

Se o bloco for encurtado em um canto interno, a orientação da ferramenta programada é assumida da mesma maneira no fim do bloco.

7.2.5 Fresamento de topo (A4, B4, C4, A5, B5, C5)

O fresamento de topo serve para usinagem de qualquer superfície curvada.



Para este tipo de fresamento 3D precisamos da descrição das trajetórias 3D sobre a superfície da peça de trabalho.

Normalmente no CAM os cálculos são executados considerando-se a forma e dimensões da ferramenta. Depois os blocos NC calculados são inseridos no comando numérico através de pós-processamento.

Programação da curvatura de trajetória

Descrição das superfícies

A descrição da curvatura de trajetória é realizada através de vetores normais de superfície com os seguintes componentes:

`A4, B4, C4` Vetor inicial no início do bloco

`A5, B5, C5` Vetor final no fim do bloco

Se em um bloco existir apenas o vetor inicial, o vetor normal de superfície permanecerá constante por todo bloco. Se em um bloco existir apenas o vetor final, então será realizada a interpolação de grande circunferência a partir do valor final do bloco anterior até o valor final programado.

Se foram programados vetor inicial e final, então também se realiza a interpolação de grande circunferência entre os dois sentidos. Com isso são obtidos percursos suaves e constantes.

No ajuste básico os vetores normais de superfície apontam para o sentido Z, independentemente do plano ativo G17 até G19.

O comprimento de um vetor é irrelevante.

Os componentes de vetor não programados são definidos como zero.

Em ORIWKS ativo (veja "Relação dos eixos de orientação (ORIWKS, ORIMKS) (Página 354)") os vetores normais da superfície referem-se aos quadros ativos e giram juntos na rotação de quadros.

Fabricante da máquina

O vetor normal de superfície deve estar perpendicular em relação à tangente do percurso, dentro de um valor limite ajustável via dado da máquina, caso contrário será emitido um alarme.

7.2.6 Relação dos eixos de orientação (ORIWKS, ORIMKS)

Na programação da orientação no sistema de coordenadas da peça através

- do ângulo euleriano e ângulo RPY ou
- vetor de orientação

se pode ajustar o decurso do movimento de rotação através do `ORIMKS`/`ORIWKS`.

Indicação

Fabricante da máquina

O tipo de interpolação para a orientação é definido com o dado de máquina:

MD21104 $MC_ORI_IPO_WITH_G_CODE

=FALSE: A referência são os comandos G ORIWKS e ORIMKS

=TRUE: A referência são os comandos G do 51º grupo (ORIAXES, ORIVECT, ORIPLANE, ...)

Sintaxe

```
ORIMKS=...
ORIWKS=...
```

Significado

`ORIMKS`:	Rotação no sistema de coordenadas da máquina
`ORIWKS`:	Rotação no sistema de coordenadas da peça de trabalho

Indicação

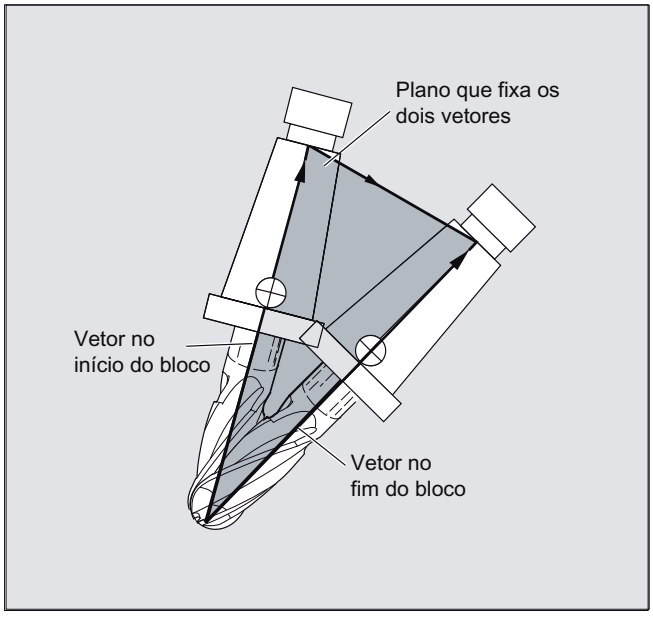
ORIWKS é ajuste básico. No caso de um programa de cinco eixos onde ainda não se sabe em qual máquina ele deverá ser executado, sempre se deve selecionar ORIWKS. Quais movimentos que a máquina realmente executa, depende da cinemática da máquina.

Por exemplo, com `ORIMKS` podem ser programados os movimentos reais da máquina para evitar colisões com dispositivos ou similares.

Outras informações

Com o ORIMKS o movimento executado pela ferramenta **depende** da cinemática da máquina. Para a alteração da orientação com a ponta da ferramenta fixa no espaço, se realiza a interpolação de modo linear entre as posições dos eixos rotativos.

Com o ORIWKS o movimento executado pela ferramenta **não depende** da cinemática da máquina. Para a alteração da orientação com a ponta da ferramenta fixa no espaço, a ferramenta movimenta-se no plano formado pelos vetores inicial e final.



Posições singulares

Indicação

ORIWKS

Movimentos de orientação na área da posição singular da máquina de cinco eixos requerem grandes movimentos dos eixos da máquina. (Por exemplo, em um cabeçote rotativo tendo C como eixo de rotação e A como eixo de orientação, todas posições com A=0 são singulares.)

Fabricante da máquina

Para não sobrecarregar os eixos da máquina, o controle de velocidade reduz bastante a velocidade de percurso próximo às posições singulares.

Com os dados da máquina

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

a transformação pode ser parametrizada de modo que os movimentos de orientação na proximidade do polo sejam colocados pelo polo e com isso seja possível realizar uma usinagem rápida.

As posições singulares são tratadas apenas com o MD $MC_TRAFO5_POLE_LIMIT .

Literatura:
/FB3/ Manual de funcionamento, funções especiais; Transformação de 3 a 5 eixos (F2), Capítulo "Posições singulares e seu tratamento".

7.2.7 Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

A função de eixos de orientação descreve a orientação da ferramenta no espaço e é alcançada através da programação dos Offsets para eixos rotativos. Um terceiro grau de liberdade pode ser obtido através da rotação adicional da ferramenta em torno de si mesma. Esta orientação de ferramenta é realizada em qualquer ponto no espaço através de um terceiro eixo rotativo e requer a transformação de seis eixos. A autorrotação do equipamento depende do tipo de interpolação dos vetores de rotação definido com os ângulos rotativos THETA (veja "Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 366)".

Os eixos de orientação são programados através dos identificadores de eixo A2, B2 e C2.

Sintaxe

```
N... ORIAXES/ORIVECT              Interpolação linear ou de grande
                                  circunferência
N...  G1 X Y Z A B C

N... ORIPLANE                     ; Interpolação de orientação do plano

N... ORIEULER/ORIRPY/ORIRPY2      ; Ângulo de orientação através de ângulo
                                  euleriano/RPY
N... G1 X Y Z A2= B2= C2=         ; Programação de ângulo dos eixos virtuais

N... ORIVIRT1/ORIVIRT2            ; eixos virtuais de orientação Def. 1/2
N... G1 X Y Z A3= B3= C3=         ; programação de direção do vetor
```

Indicação

Para mudanças na orientação ao longo de uma superfície de cone no local, podem ser programadas outras offsets de placas rotativas dos eixos de orientação (veja "Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 358)".

Significado

ORIAXES:	Interpolação linear dos eixos da máquina ou dos eixos de orientação
ORIVECT:	Interpolação de grande circunferência (idêntico ao ORIPLANE)

ORIMKS: ORIWKS:	Rotação no sistema de coordenadas da máquina Rotação no sistema de coordenadas da peça de trabalho Veja "Relação dos eixos de orientação (ORIWKS, ORIMKS) (Página 354)".
A= B= C=:	Programação da posição de eixo da máquina
ORIEULER:	Programação da orientação através de ângulo euleriano
ORIRPY:	Programação da orientação através de ângulo RPY A sequência de rotação é XYZ, onde se aplica: • A2 é o eixo de rotação em volta de X • B2 é o eixo de rotação em volta de Y • C2 é o eixo de rotação em volta de Z
ORIRPY2:	Programação da orientação através de ângulo RPY A sequência de rotação é ZYX, onde se aplica: • A2 é o eixo de rotação em volta de Z • B2 é o eixo de rotação em volta de Y • C2 é o eixo de rotação em volta de X
A2= B2= C2=:	Programação de ângulo dos eixos virtuais
ORIVIRT1/ORIVIRT2:	Programação da orientação através de eixos virtuais de orientação Definição 1: conforme MD21120 $MC_ORIAX_TURN_TAB_1 Definição 2: conforme MD21130 $MC_ORIAX_TURN_TAB_2
A3= B3= C3=:	Programação do vetor de direção do eixo de sentido

Outras informações

Fabricante da máquina

Com MD21102 $MC_ORI_DEF_WITH_G_CODE se define, como os ângulos A2, B2, C2 serão definidos:

A definição é realizada conforme o MD21100 $MC_ORIENTATION_IS_EULER (Standard) ou conforme o grupo 50 de códigos G (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Com MD21104 $MC_ORI_IPO_WITH_G_CODE se define qual tipo de interpolação está ativa: ORIWKS/ORIMKS ou ORIAXES/ORIVECT.

Modo de operação JOG

Os ângulos de orientação sempre são interpolados linearmente neste modo de operação. Com o deslocamento contínuo e incremental através das teclas de deslocamento somente pode ser movimentado um eixo de orientação. Através das manivelas eletrônicas os eixos de orientação podem ser movimentados simultaneamente.

Para o deslocamento manual dos eixos de orientação, o interruptor de override de avanço ou o interruptor de override de avanço rápido tem efeito na correção do avanço rápido.

Com os seguintes dados da máquina é possível realizar uma especificação separada de velocidade:

MD21160 $MC_JOG_VELO_RAPID_GEO

MD21165 $MC_JOG_VELO_GEO

MD21150 $MC_JOG_VELO_RAPID_ORI

MD21155 $MC_JOG_VELO_ORI

Indicação

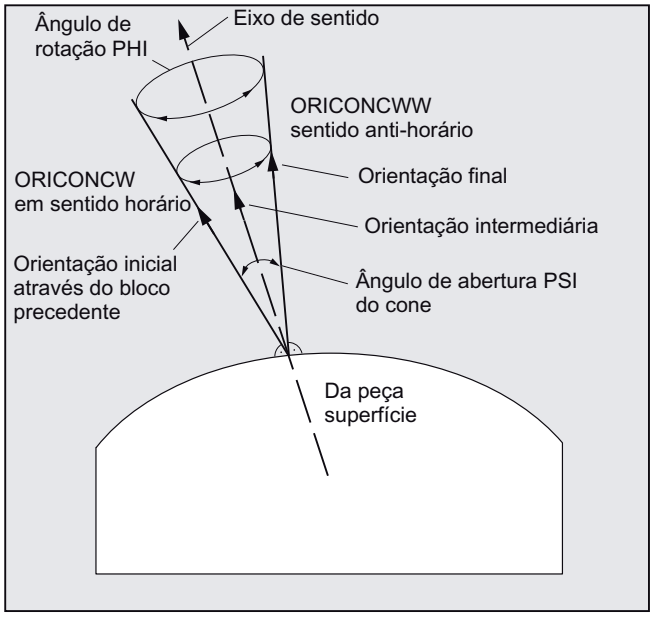
SINUMERIK 840D sl com "Pacote de transformação Handling"

Com a função "Deslocamento manual cartesiano", em modo JOG, se pode ajustar separadamente a translação dos eixos geométricos no sistema de referência MCS, WCS e TCS.

Literatura:
Manual de funções ampliadas; Transformações cinemáticas (M1)

7.2.8 Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

Com a orientação ampliada é possível executar alterações da orientação ao longo de uma superfície periférica cônica no espaço. A interpolação do vetor de orientação sobre uma superfície periférica cônica é realizada com os comandos modais ORICONxx. Para a interpolação em um plano se pode programar a orientação final com ORIPLANE. Geralmente a orientação inicial é definida através dos blocos anteriores.



Programação

A orientação final é definida através da indicação da programação em ângulos eulerianos ou RPY com A2, B2, C2 ou através da programação das posições de eixo rotativo com A, B, C. Para os eixos de orientação ao longo da superfície periférica cônica são necessárias outras informações de programação:

- Eixo de rotação do cone como vetor com `A6, B6, C6`
- Ângulo de abertura `PSI` com o identificador `NUT`
- Orientação intermediária com superfície cônica com `A7, B7, C7`

Indicação

Programação do vetor de direção A6, B6, C6 par ao eixo de rotação do cone

A programação de uma orientação final não é extremamente necessária. Se não for especificada nenhuma orientação final, então se interpola uma superfície cônica inteira com 360 graus.

Programação do ângulo de abertura do cone com NUT=ângulo

A indicação de uma orientação final é extremamente necessária.

Uma superfície cônica completa com 360 graus não pode ser interpolada dessa forma.

Programação da orientação intermediária A7, B7, C7 na superfície cônica

A indicação de uma orientação final é extremamente necessária. A alteração da orientação e o sentido de giro são claramente definidos através dos três vetores de orientação inicial, final e intermediário. Neste caso todos os três vetores devem ser diferentes entre si. Se a orientação intermediária programada for paralela à orientação inicial ou final, então será realizada uma interpolação linear de grande circunferência da orientação no plano que foi fixado pelos vetores inicial e final.

Interpolação ampliada de orientação sobre uma superfície periférica cônica

```
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)
```

Interpolação sobre uma superfície cônica com

vetor de direção em sentido horário/anti-horário do cone e orientação final ou

transição tangencial e
indicação da orientação final

ou
indicação da orientação final e uma orientação intermediária na superfície cônica com

polinômios para ângulo de rotação e polinômios para ângulo de abertura

Parâmetro

`ORIPLANE`:	Interpolação no plano (interpolação de grande circunferência)
`ORICONCW`:	Interpolação sobre uma superfície periférica cônica no sentido horário

ORICONCCW:	Interpolação sobre uma superfície periférica cônica no sentido anti-horário
ORICONTO:	Interpolação sobre uma superfície periférica cônica de transição tangencial
A6= B6= C6=:	Programação do eixo de rotação do cone (vetor normalizado)
NUT=ângulo:	Ângulo de abertura do cone em graus
NUT=+179:	Ângulo de deslocamento menor ou igual a 180 graus
NUT=-181:	Ângulo de deslocamento maior ou igual a 180 graus
ORICONIO:	Interpolação sobre uma superfície periférica cônica
A7= B7= C7=:	Orientação intermediária (programação como vetor normalizado)
PHI:	Ângulo de rotação da orientação em torno do eixo de sentido do cone
PSI:	Ângulo de abertura do cone
possíveis polinômios PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	Além dos respectivos ângulos também se pode programar polinômios de até 5º grau

Exemplo: diferentes alterações de orientação

Código de programa	**Comentário**
…	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	; Transformação de orientação ativada.
N30 ORIVECT	; ;interpolar a orientação de ferramenta como vetor.
…	; Orientação de ferramenta no plano.
N40 ORIPLANE	; Selecionar a interpolação de grande circunferência.
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; Orientação no plano Y/Z girada em 45 graus, ao final do bloco é atingida a orientação (0,1/√2,1/√2).
…	
N70 ORICONCW	; Programação da orientação na superfície cônica:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	; O vetor de orientação é interpolado em sentido horário sobre uma superfície cônica com o sentido (0,0,1) até a orientação (1/√2,0,1/√2), sendo que o ângulo de rotação é de 270 graus.
N90 A6=0 B6=0 C6=1	; A orientação de ferramenta percorre uma rotação completa sobre a mesma superfície cônica.

Outras informações

Quando devem ser descritas alterações de orientação em uma superfície periférica cônica qualquer no espaço, então o vetor com que a orientação de ferramenta deve ser girado deve ser conhecido. Além disso, se deve especificar a orientação inicial e final. A orientação inicial resulta do bloco anterior e a orientação final deve ser programada ou definida por outras condições.

A programação no plano ORIPLANE corresponde ao ORIVECT

A programação da interpolação de grande circunferência junto com polinômios angulares corresponde à interpolação linear e de polinômios de contornos. A orientação de ferramenta é interpolada neste plano, que foi fixado pela orientação inicial e orientação final. Se também forem programados polinômios, então o vetor de orientação também pode ser excluído do plano.

Programação de círculos em um plano G2/G3, CIP e CT

A orientação ampliada corresponde à interpolação de círculos em um plano. Sobre as respectivas possibilidades de programação de círculos com indicação de centro ou indicação de raio como G2/G3, círculo através de ponto intermediário CIP e círculos tangenciais CT, veja

Literatura: Manual de programação Fundamentos, "Programar comandos de curso".

Programação da orientação

Interpolação do vetor de orientação sobre uma superfície periférica cônica ORICONxx

Para a interpolação de orientações sobre uma superfície periférica cônica podem ser selecionados quatro tipos diferentes de interpolação do grupo 51 de G:

1. Interpolação sobre uma superfície cônica em sentido horário `ORICONCW` com indicação da orientação final e do sentido do cone ou do ângulo de abertura. O vetor de direção é programado com os identificadores `A6, B6, C6` e o ângulo de abertura do cone é programado com o identificador NUT= na faixa de valores de 0 a 180 graus.
2. Interpolação sobre uma superfície cônica em sentido anti-horário `ORICONCWW` com indicação da orientação final e do sentido do cone ou do ângulo de abertura. O vetor de direção é programado com os identificadores `A6, B6, C6` e o ângulo de abertura do cone é programado com o identificador NUT= na faixa de valores de 0 a 180 graus.
3. Interpolação sobre uma superfície cônica `ORICONIO` com indicação da orientação final e uma orientação intermediária, que é programada com os identificadores `A7, B7, C7`.
4. Interpolação sobre uma superfície cônica `ORICONTO` com transição tangencial e indicação da orientação final. O vetor de direção é programado com os identificadores `A6, B6, C6`.

7.2.9 Especificação de orientação de dois pontos de contato (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Programação da alteração de orientação através da segunda curva espacial ORICURVE

Uma outra possibilidade de programação de alterações de orientação é que além da ponta da ferramenta também se pode, ao longo de uma curva espacial, programar o movimento de um segundo ponto de contato da ferramenta com `ORICURVE`. Com isso se pode definir claramente as alterações de orientação da ferramenta, como na programação do próprio vetor de ferramenta.

Fabricante da máquina

Por favor, observe as instruções do fabricante da máquina sobre os identificadores de eixo ajustáveis através do dado de máquina para programação da 2ª trajetória de orientação da ferramenta.

Programação

Neste tipo de interpolação se pode programar pontos com `G1` e polinômios com `POLY` para as duas curvas espaciais. Círculos e evolventes não são permitidos. Adicionalmente pode ser ativada uma interpolação de Spline com `BSPLINE` e a função "Agrupamento de blocos Spline curtos".

Literatura:
Manual de funcionamento das funções básicas; Modo de controle da trajetória, Parada exata, Look Ahead (B1), Capítulo: Agrupamento de blocos Spline curtos

Não são permitidos outros tipos de Spline `ASPLINE` e `CSPLINE`, assim como a ativação de um compressor com `COMPON`, `COMPCURV` ou `COMPCAD`.

O movimento de dois pontos de contato da ferramenta pode ser especificado na programação dos polinômios de orientação para coordenadas de até 5º grau.

Interpolação de orientação ampliada com curva espacial adicional e polinômios para coordenadas

```
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)
```

Indicação do movimento do segundo ponto de contato da ferramenta e polinômios adicionais das respectivas coordenadas

Parâmetro

`ORICURVE`	`Interpolação da orientação com especificação do movimento de dois pontos de contato da ferramenta.`
`XH YH ZH`	`Identificador das coordenadas do segundo ponto de contato da ferramenta do contorno adicional como curva espacial`
`possíveis polinômios PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)`	`Além dos respectivos pontos finais, também se pode programar as curvas espaciais com polinômios.`

```
xe, ye, ze                Pontos finais da curva espacial
xi, yi, zi                Coeficientes dos polinômios de até 5° grau, no máximo
```

Indicação

Identificadores XH YH ZH para programação de uma 2ª trajetória de orientação

Os identificadores devem ser selecionados de modo que não resulte em nenhum conflito com outros identificadores dos eixos lineares

Eixos X Y Z

e eixos rotativos como

Ângulo euleriano A2 B2 C2 e ângulo RPY

Vetores de direção A3 B3 C3

Vetores normais de superfície A4 B4 C4 e A5 B5 C5

Vetores de rotação A6 B6 C6 e coordenadas de pontos intermediários A7 B7 C7

ou outros parâmetros de interpolação.

7.3 Polinômios de orientação (PO[ângulo], PO[coordenadas])

Independente de qual interpolação de polinômios do grupo 1 de G está ativo, podem ser programados dois tipos diferentes de polinômios de orientação de até 5º grau em uma transformação de três a cinco eixos.

1. Polinômios para **ângulo:** Ângulo de avanço LEAD, ângulo lateral TILT em relação ao plano que foi fixado pela orientação inicial e orientação final.
2. Polinômios para **coordenadas:** XH, YH, ZH da segunda curva espacial para orientação de ferramenta de um ponto de referência na ferramenta.

Em uma transformação de seis eixos, adicionalmente à orientação de ferramenta, pode ser programada a rotação do vetor de rotação THT com polinômios de até 5º grau para rotações da própria ferramenta.

Sintaxe

Polinômios de orientação de tipo 1 para **ângulo**

```
N… PO[PHI]=(a2, a3, a4, a5)
N… PO[PSI]=(b2, b3, b4, b5)
```

Transformação de três a cinco eixos

Polinômios de orientação de tipo 2 para **coordenadas**

```
N… PO[XH]=(xe, x2, x3, x4, x5)
N… PO[YH]=(ye, y2, y3, y4, y5)
N… PO[ZH]=(ze, z2, z3, z4, z5)
```

Identificador para coordenadas da segunda trajetória de orientação para orientação da ferramenta

Nos dois casos também se pode programar um polinômio para a **rotação** em transformações de seis eixos com

```
N… PO[THT]=(c2, c3, c4, c5)
```

Interpolação da rotação relativa à trajetória

ou

```
N… PO[THT]=(d2, d3, d4, d5)
```

interpolação absoluta, relativa e tangencial à alteração da orientação

do vetor de orientação. Isso é possível quando a transformação de um vetor de rotação com um Offset programável e interpolável pelo ângulo de rotação THETA oferecer o suporte necessário.

Significado

`PO[PHI]`	Ângulo no plano entre a orientação inicial e a orientação final
`PO[PSI]`	Ângulo que descreve a inclinação da orientação a partir do plano entre a orientação inicial e orientação final
`PO[THT]`	Ângulo de rotação resultante da rotação do vetor de rotação de um comando G do grupo 54 programado com THETA
`PHI`	Ângulo de avanço LEAD

PSI	Ângulo lateral TILT
THETA	Rotação no sentido da ferramenta em Z
PO[XH]	Coordenada X do ponto de referência na ferramenta
PO[YH]	Coordenada Y do ponto de referência na ferramenta
PO[ZH]	Coordenada Z do ponto de referência na ferramenta

Outras informações

Os polinômios de orientação não podem ser programados

- quando as interpolações de Spline ASPLINE, BSPLINE, CSPLINE estiverem ativas.
  Os polinômios do tipo 1 para ângulo de orientação são possíveis para cada tipo de interpolação, exceto para Spline, isto é, possível na interpolação linear com avanço rápido G00 ou com avanço G01
  na interpolação de polinômios com POLY e
  na interpolação de círculos e evolventes com G02, G03, CIP, CT, INVCW e INCCCW
  .
  Em contrapartida, os polinômios do tipo 2 para coordenadas de orientação somente são possíveis se a
  interpolação linear com avanço rápido G00 ou com avanço G01 ou a
  interpolação de polinômios com POLY estiver ativa.
- quando a orientação for interpolada através da interpolação de eixos ORIAXES. Neste caso se pode programar os polinômios diretamente com PO[A] e PO[B] para os eixos de orientação A e B.

Polinômios de orientação do tipo 1 com ORIVECT, ORIPLANE e ORICONxx

Na interpolação de grande circunferência e interpolação de superfície cônica com ORIVECT, ORIPLANE e ORICONxx são possíveis apenas os polinômios de orientação do tipo 1.

Polinômios de orientação do tipo 2 com ORICURVE

Se a interpolação estiver ativa com a curva espacial ORICURVE adicional, os componentes cartesianos do vetor de orientação são interpolados e apenas são possíveis polinômios de orientação do tipo 2.

7.4 Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)

Se em máquinas com ferramenta articulada a orientação da ferramenta também deverá ser possível de ser alterada, então esse bloco também é programado com uma orientação final. Em função da cinemática da máquina também se pode programar o sentido da orientação dos eixos ou a direção de rotação do vetor de orientação THETA. Para estes vetores de rotação podem ser programados vários tipos de interpolação:

- ORIROTA: Ângulo de rotação de um sentido de rotação especificado como absoluto.
- ORIROTR: Ângulo de rotação relativo ao plano entre a orientação inicial e final.
- ORIROTT: Ângulo de rotação relativo à alteração do vetor de orientação.
- ORIROTC: Ângulo de rotação tangencial à tangente da trajetória.

Sintaxe

Somente se o tipo de interpolação ORIROTA estiver ativo é que o ângulo de rotação ou o vetor de rotação pode ser programado nos seguintes quatro modos:

1. Diretamente as posições de eixo rotativo `A, B, C`
2. Ângulo euleriano (em graus) através do `A2, B2, C2`
3. Ângulo RPY (em graus) através do `A2, B2, C2`
4. Vetor de direção através do `A3, B3, C3` (ângulo de rotação com `THETA=<valor`)

Se ORIOTR ou ORIOTT estiver ativo, o ângulo de rotação somente poderá ser programado diretamente com `THETA`.

Uma rotação também pode ser programada separadamente em um bloco, sem a ocorrência de uma alteração da orientação. ORIROTR e ORIROTT não é relevante. Neste caso o ângulo de rotação sempre é interpretado relativo ao sentido absoluto (ODIROTA).

`N... ORIROTA`	Definir a interpolação do vetor de rotação
`N... ORIROTR`	
`N... ORIROTT`	
`N... ORIROTC`	
`N... A3= B3= C3= THETA=<valor>`	Definir a rotação do vetor de orientação
`N... PO[THT]=(`d_2`, `d_3`, `d_4`, `d_5`)`	Interpolar o ângulo de rotação com polinômio de 5º grau

Significado

`ORIROTA`:	Ângulo de rotação de um sentido de rotação especificado como absoluto
`ORIROTR`:	Ângulo de rotação relativo ao plano entre a orientação inicial e a orientação final
`ORIROTT`:	Ângulo de rotação como vetor de rotação tangencial à alteração da orientação
`ORIROTC`:	Ângulo de rotação como vetor de rotação tangencial à tangente da trajetória

`THETA`:	Rotação do vetor de orientação
`THETA=<valor>`:	Ângulo de rotação em graus alcançado no fim do bloco
`THETA=`Θe:	Ângulo de rotação com ângulo final Θ_e do vetor de rotação
`THETA=AC(…)`:	Comutar para dimensão absoluta por blocos
`THETA=AC(…)`:	Comutar para dimensão incremental por blocos
Θe:	O ângulo final do vetor de rotação está ativo tanto absoluto com G90 como relativo com G91 (dimensão incremental)
`PO[THT]=(....)`:	Polinômio para ângulo de rotação

Exemplo: rotações das orientações

Código de programa	**Comentário**
`N10 TRAORI`	`; Ativar transformação de orientação`
`N20 G1 X0 Y0 Z0 F5000`	`; Orientação da ferramenta`
`N30 A3=0 B3=0 C3=1 THETA=0`	`; No sentido Z com ângulo de rotação 0`
`N40 A3=1 B3=0 C3=0 THETA=90`	`; No sentido X e rotação de 90 graus`
`N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)`	`; Orientação`
`N60 A3=0 B3=1 C3=0 THETA=IC(-90)`	`; No sentido Y e rotação de 180 graus`
`N70 ORIROTT`	`; Permanece constante e rotação de 90 graus`
`N80 A3=1 B3=0 C3=0 THETA=30`	`; Ângulo de rotação relativo à alteração de orientação`
	`; Vetor de rotação em ângulo de 30 graus em relação ao plano X-Y`

Na interpolação do bloco `N40` o ângulo de rotação é interpolado linearmente do valor inicial de 0 grau até o valor final de 90 graus. No bloco `N50` se altera o ângulo de rotação de 90 graus para 180 graus de acordo com a parábola θ(u) = +90u2. No `N60` também pode ser executada uma rotação sem a ocorrência de uma alteração de orientação.

No `N80` se gira a orientação da ferramenta do sentido Y para sentido X. Neste caso a alteração da orientação está no plano X–Y e o vetor de rotação forma um ângulo de 30 graus com este plano.

Outras informações

ORIROTA

O ângulo de rotação `THETA` é interpolado relativo a uma direção absoluta definida no espaço. O sentido básico de rotação é realizado através de dados da máquina

ORIROTR

O ângulo de rotação `THETA` é interpretado relativo ao plano fixado pelas orientações inicial e final.

ORIROTT

O ângulo de rotação `THETA` é interpretado relativo à alteração da orientação. Para `THETA=0` o vetor de rotação é interpolado de modo tangencial à alteração da orientação e apenas será diferente do `ORIROTR` se para a orientação foi programado pelo menos um polinômio para o "ângulo de inclinação PSI". Com isso resulta uma alteração da orientação que não percorre

no plano. Através de um ângulo de rotação `THETA` programado adicionalmente, o vetor de rotação, por exemplo, pode ser interpolado de modo que ele sempre forma um determinado valor para a alteração da orientação.

ORIROTC

O vetor de rotação é interpolado relativo à tangente da trajetória através de um Offset programável pelo ângulo `THETA`. Para o ângulo de Offset também pode ser programado um polinômio `PO[THT]=(c2, c3, c4, c5)` de até 5º grau.

7.5 Orientações relativas à trajetória

7.5.1 Tipos de orientação relativos à trajetória

Com esta função ampliada não se alcança a orientação relativa apenas no fim do bloco, mas por todo o decurso da trajetória. É realizada a transferência da orientação alcançada no bloco precedente através da interpolação de grande circunferência para a orientação final programada. Basicamente existem duas opções de se programar a orientação desejada de modo relativo à trajetória:

1. A orientação da ferramenta como a rotação da ferramenta é interpolada de modo relativo à trajetória com ORIPATH e ORPATHTS.
2. O vetor de orientação é programado e interpolado da forma usual. Com ORIROTC se posiciona a rotação do vetor de orientação de modo relativo à tangente da trajetória.

Sintaxe

O tipo de interpolação da orientação e da rotação da ferramenta se programa com:

`N... ORIPATH`	Orientação relativa à trajetória
`N... ORIPATHS`	Orientação relativa à trajetória com suavização do decurso de orientação
`N... ORIROTC`	Interpolação do vetor de rotação relativa à trajetória

Uma dobra da orientação ressaltada através de uma esquina no decurso da trajetória pode ser suavizada com `ORIPATHS`. A direção da distância de curso do movimento de retração é programada através do vetor com os componentes `A8=X, B8=Y C8=Z .`

Com `ORIPATH/ORIPATHS` se pode programar diversas referências com a tangente da trajetória através dos três ângulos

- `LEAD=` Indicação do ângulo de avanço relativo à trajetória e superfície
- `TILT=` Indicação do ângulo lateral relativo à trajetória e superfície
- `THETA=` Ângulo de rotação

para o decurso de trajetória total. Ao ângulo de rotação `THETA` também podem ser programados polinômios de até 5º grau com `PO[THT]=(...)`.

Indicação

Fabricante da máquina

Observe as instruções do fabricante da máquina. Através de dados projetáveis de máquina e de ajuste podem ser realizadas outros ajustes para o tipo de orientação relativo à trajetória. Para mais explicações, veja

Literatura:
/FB3/ Manual de funcionamento, funções especiais; Transformação de 3 a 5 eixos (F2), Capítulo "Orientação"

Significado

A interpolação dos ângulos `LEAD` e `TILT` é ajustada de modo diversificado através de dado de máquina:

- A referência da orientação da ferramenta programada com `LEAD` e `TILT` é conservada por todo o bloco.
- Ângulo de avanço `LEAD`: Rotação em torno da direção perpendicular à tange e vetor normal `TILT`: Rotação da orientação em torno do vetor normal.
- Ângulo de avanço `LEAD`: Rotação em torno da direção perpendicular à tange e vetor normal do ângulo lateral `TILT`: Rotação da orientação em torno da direção da tangente da trajetória.
- Ângulo de rotação `THETA`: Rotação da ferramenta em torno de si mesma com um terceiro eixo rotativo extra como eixo de orientação na transformação de seis eixos.

Indicação

A orientação relativa à trajetória não é permitida junto com OSC, OSS, OSSE, OSD, OST

A interpolação de orientação relativa à trajetória `ORIPATH` ou `ORIPATHS` e `ORIOTC` não pode ser programada junto com a suavização do decurso de orientação com um dos comando G do grupo 34. Para isso o OSOF deve estar ativo.

7.5.2 Rotação da orientação da ferramenta relativa à trajetória (ORIPATH, ORIPATHS, ângulo de rotação)

Em uma transformação de seis eixos, a ferramenta também pode ser girada em torno de si mesma com um terceiro eixo rotativo para qualquer orientação de ferramenta no espaço. Na rotação da orientação de ferramenta relativa à trajetória com ORIPATH ou ORIPATHS, se pode programar a rotação adicional através do ângulo de rotação THETA. De modo alternativo, se pode programar os ângulos LEAD e TILT através de um vetor que está perpendicular à direção da ferramenta no plano.

Fabricante da máquina

Observe as instruções do fabricante da máquina. Através de dado de máquina se pode ajusta de forma diferente a interpolação dos ângulos LEAD e TILT.

Sintaxe

Rotação da orientação da ferramenta e da própria ferramenta

O tipo de orientação de ferramenta relativo à trajetória é ativado com ORIPATH ou ORIPATHS.

`N... ORIPATH`	Ativar o tipo de orientação relativo à trajetória
`N... ORIPATHS`	Ativação do tipo de orientação relativo à trajetória com suavização do decurso de orientação

Ativação dos três possíveis ângulos com efeito de rotação:

`N... LEAD=`	Ângulo para orientação programada relativa ao vetor normal de superfície

`N... TILT=`	Ângulo para orientação programada no plano perpendicular à tangente da trajetória relativa ao vetor normal de superfície
`N... THETA=`	Ângulo de rotação relativo à alteração da orientação em torno da direção de ferramenta do terceiro eixo rotativo

Os valores do ângulo no fim do bloco são programados com `LEAD=valor`, `TILT=valor` ou `THETA=valor`. Adicionalmente aos ângulos constantes podem ser programados polinômios de até 5º grau para todos os três ângulos.

`N... PO[PHI]=(a2, a3, a4, a5)`	Polinômio para ângulo de avanço LEAD
`N... PO[PSI]=(b2, b3, b4, b5)`	Polinômio para ângulo lateral TILT
`N... PO[THT]=(d2, d3, d4, d5)`	Polinômio para ângulo de rotação THETA

Na programação podem ser descartados os coeficientes de polinômio mais altos, que são igual a zero. O exemplo `PO[PHI]=a2` resulta em uma parábola para o ângulo de avanço `LEAD`.

Significado

Orientação de ferramenta relativa à trajetória

`ORIPATH`:	Orientação da ferramenta relativa à trajetória
`ORIPATHS`:	A orientação de ferramenta relativa à dobra de trajetória no decurso de orientação é suavizada
`LEAD`:	Ângulo relativo ao vetor normal de superfície, no plano fixado pela tangente da trajetória e vetor normal de superfície
`TILT`:	Rotação da orientação em torno do sentido Z ou rotação em torno da tangente da trajetória
`THETA`:	Rotação em torno do sentido da ferramenta para Z
`PO[PHI]`:	Polinômio de orientação para ângulo de avanço LEAD
`PO[PSI]`:	Polinômio de orientação para ângulo lateral TILT
`PO[THT]`:	Polinômio de orientação para ângulo de rotação THETA

Indicação

Ângulo de rotação THETA

Para a rotação da ferramenta com terceiro eixo rotativo como eixo de orientação em torno de si mesma é necessária uma transformação de seis eixos.

7.5.3 Interpolação relativa à trajetória da rotação da ferramenta (ORIROTC, THETA)

Interpolação com vetores de rotação

À rotação de ferramenta relativa à tangente de trajetória programada com ORIROTC se pode interpolar o vetor de direção com um Offset programável através do ângulo de rotação THETA. Neste caso se pode programar um polinômio de até 5º grau com PO[THT] para o ângulo de Offset.

Sintaxe

`N... ORIROTC`	Posicionar a rotação da ferramenta relativa à tangente de trajetória
`N... A3= B3= C3= THETA=valor`	Definir a rotação do vetor de orientação
`N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)`	Interpolar o ângulo Offset com polinômio de até 5º grau

Uma rotação também pode ser programada separadamente em um bloco, sem a ocorrência de uma alteração da orientação.

Significado

Interpolação da rotação da ferramenta relativa à trajetória na transformação de seis eixos

`ORIROTC`:	posicionar vetor de rotação tangencial à tangente de trajetória
`THETA=valor`:	Ângulo de rotação em graus alcançado no fim do bloco
`THETA=`Θ`e`:	Ângulo de rotação com ângulo final Θ_e do vetor de rotação
`THETA=AC(…)`:	Comutar para dimensão absoluta por blocos
`THETA=IC(…)`:	Comutar para dimensão incremental por blocos
`PO[THT]=(c2, c3, c4, c5)`:	Interpolar o ângulo de Offset com polinômio de 5º grau

Indicação

Interpolação do vetor de rotação ORIROTC

Se a rotação da ferramenta relativa à tangente de trajetória também deve ser posicionada contra o sentido da orientação da ferramenta, então isso será possível apenas com uma transformação de seis eixos.

Com ORIROTC ativo

O vetor de rotação ORIROTA não pode ser programado. No caso de uma programação, será emitido o ALARME 14128 "Programação absoluta da rotação da ferramenta com ORIROTC ativo".

Sentido de orientação da ferramenta na transformação de três a cinco eixos

O sentido de orientação da ferramenta pode ser programado da forma usual, através de ângulo euleriano e ângulo RPY ou através de vetores de direção, como na transformação de

três a cinco eixos. Também são possíveis as alterações de orientação da ferramenta no espaço através da programação da interpolação de grande circunferência ORIVECT, interpolação linear dos eixos de orientação ORIAXES, todas interpolações sobre uma superfície periférica cônica ORICONxx, e também a interpolação adicional à curva espacial com dois pontos de contato da ferramenta ORICURVE.

`G....:`	Indicação do tipo de movimento dos eixos rotativos
`X, Y, Z:`	Indicação dos eixos lineares
`ORIAXES:`	Interpolação linear dos eixos da máquina ou dos eixos de orientação
`ORIVECT:`	Interpolação de grande circunferência (idêntico ao ORIPLANE)
`ORIMKS:` `ORIWKS:`	Rotação no sistema de coordenadas da máquina Rotação no sistema de coordenadas da peça de trabalho Veja a descrição no cap. Rotações da orientação da ferramenta
`A= B= C=:`	Programação da posição de eixo da máquina
`ORIEULER:`	Programação da orientação através de ângulo euleriano
`ORIRPY:`	Programação da orientação através de ângulo RPY
`A2= B2= C2=:`	Programação de ângulo dos eixos virtuais
`ORIVIRT1:` `ORIVIRT2:`	Programação da orientação através de eixos virtuais de orientação (definição 1), definição conforme MD $MC_ORIAX_TURN_TAB_1 (definição 2), definição conforme MD $MC_ORIAX_TURN_TAB_2
`A3= B3= C3=:`	Programação do vetor de direção do eixo de sentido
`ORIPLANE:`	Interpolação no plano (interpolação de grande circunferência)
`ORICONCW:`	Interpolação sobre uma superfície periférica cônica no sentido horário
`ORICONCCW:`	Interpolação sobre uma superfície periférica cônica no sentido anti-horário
`ORICONTO:`	Interpolação sobre uma superfície periférica cônica de transição tangencial
`A6= B6= C6=:`	Programação do eixo de rotação do cone (vetor normalizado)
`NUT=ângulo`	Ângulo de abertura do cone em graus
`NUT=+179`	Ângulo de deslocamento menor ou igual a 180 graus
`NUT=-181`	Ângulo de deslocamento maior ou igual a 180 graus
`ORICONIO:`	Interpolação sobre uma superfície periférica cônica
`A7= B7= C7=:`	Orientação intermediária (programação como vetor normalizado)
`ORICURVE` `XH YH ZH p. ex. com polinômios PO[XH]=(xe, x2, x3, x4, x5)`	Interpolação da orientação com especificação do movimento de dois pontos de contato da ferramenta. Além dos respectivos pontos finais também se pode programar polinômios às curvas espaciais extras.

Indicação

Se a orientação de ferramenta com ORIAXES ativo for interpolada através dos eixos de orientação, então o posicionamento relativo à trajetória do ângulo de rotação somente será realizado no fim do bloco.

7.5.4 Suavização do decurso de orientação (ORIPATHS A8=, B8=, C8=)

Nas alterações de orientação com aceleração constante no contorno existem interrupções nos movimentos de percurso que ocorrem principalmente nos cantos (esquinas) de um contorno, que não são desejadas. A dobra resultante disso no decurso da orientação pode ser suavizada através da inserção de um bloco intermediário próprio. A alteração de orientação é realizada com aceleração constante se durante a reorientação o ORIPATHS também estiver ativo. Nesta fase pode ser realizado um movimento de retração da ferramenta.

Fabricante da máquina

Observe as instruções do fabricante da máquina sobre eventuais dados de máquina e dados de ajuste pré-definidos com os quais se ativa esta função.

Através de dado de máquina se pode ajustar como o vetor de retração deve ser interpretado:

1. No sistema de coordenadas da ferramenta as coordenadas em Z estão definidas através do sentido da ferramenta.
2. No sistema de coordenadas da peça as coordenadas em Z estão definidas através do plano ativo.

Outras explanações sobre a função da "Orientação da trajetória relativa" consulte
Literatura:
Manual de funcionamento, funções especiais, transformações multieixo (F2)

Sintaxe

Para orientações de ferramenta relativas à trajetória inteira, realizadas constantemente, são necessárias outras informações de programação em um canto do contorno. O sentido e a distância de curso deste movimento são programados através do vetor com os componentes A8=X, B8=Y, C8=Z:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Significado

`ORIPATHS`:	Orientação de ferramenta relativa à trajetória, uma dobra é suavizada no decurso da orientação.
`A8= B8= C8=`:	Componentes de vetor para sentido e distância de curso
`X, Y, Z`:	Movimento de retração no sentido da ferramenta

Indicação

Programação do vetor de direção A8, B8, C8

Se a distância deste vetor for igual a zero, não se realiza nenhum movimento de retração.

ORIPATHS

A orientação de ferramenta relativa à trajetória é ativada com ORIPATHS. Caso contrário se transporta a orientação através de interpolação linear de grande circunferência da orientação inicial para a orientação final.

7.6 Compressão da orientação (COMPON, COMPCURV, COMPCAD, COMPSURF)

Os programas NC que possuem uma transformação de orientação (TRAORI) ativa, e neles existem orientações de ferramenta (independente do tipo) programadas, poderão ser comprimidos com a preservação das tolerâncias especificadas.

Programação

Orientação da ferramenta

Se uma transformação de orientação (TRAORI) estiver ativa, a orientação de ferramenta em máquinas de 5 eixos pode ser programada da seguinte forma (independente da cinemática):

- Programação do **vetor** de direção através do:
  `A3=<...> B3=<...> C3=<...>`
- Programação do **ângulo euleriano** ou **ângulo RPY** através de:
  `A2=<...> B2=<...> C2=<...>`

Rotação da ferramenta

Para máquinas de **6 eixos** também se pode programar a rotação da ferramenta para orientação de ferramenta.

A programação do ângulo de rotação é realizada com:

`THETA=<...>`

Veja " Rotações da orientação da ferramenta (Página 366) ".

Indicação

Blocos NC, em qual uma rotação adicional está programada, só pode ser comprimido, se o ângulo de rotação se alterar **linearmente**. Ou seja, nenhum polinômio com PO[THT]=(...) pode ser programado para o ângulo de rotação.

Forma geral de um bloco NC compressível

Por isso que a forma geral de um bloco NC compressível pode se apresentar da seguinte maneira:

```
N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>
```

ou

```
N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

Indicação

Os valores de posição podem ser especificados de modo direto (p. ex. X90) ou indireto através da atribuição de parâmetros (p. ex. X=R1*(R2+R3)).

Programação da orientação de ferramenta através de posições de eixo rotativo

A orientação da ferramenta também pode ser especificada através de posições de eixo rotativo, por exemplo, na seguinte forma:

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

Neste caso a compressão é executada de dois modos diferentes, em função de ser executada uma interpolação de grande circunferência ou não. Se não ocorre nenhuma interpolação de grande circunferência, então a alteração de orientação comprimida é representada através de polinômios axiais para eixos rotativos, como no procedimento usual.

Precisão de contorno

Em função do modo de compressão ajustado (MD20482 $MC_COMPRESSOR_MODE), durante a compressão, para eixos geométricos e eixos de orientação são ativadas as tolerâncias específicas de eixo configuradas (MD33100 $MA_COMPRESS_POS_TOL) ou as seguintes tolerâncias específicas de canal ajustadas através de dados de ajuste:

SD42475 $SC_COMPRESS_CONTUR_TOL (desvio máximo do contorno)

SD42476 $SC_COMPRESS_ORI_TOL (desvio angular máximo para orientação da ferramenta)

SD42477 $SC_COMPRESS_ORI_ROT_TOL (desvio máximo do ângulo de rotação da ferramenta) (disponível apenas em máquinas de 6 eixos)

Literatura:
Manual de funcionamento das funções básicas; Transformação de 3 a 5 eixos (F2), Capítulo: "Compressão da orientação"

Ativação/desativação

As funções de compressor são ativadas através dos comandos modais G `COMPON`, `COMPCURV`, `COMPCAD` e `COMPSURF`.

A função de compressor é encerrada com `COMPOF`.

Veja " Compressão de blocos NC (COMPON, COMPCURV, COMPCAD) (Página 273) ".

Indicação

O movimento de orientação somente é comprimido com a interpolação de grande circunferência ativa (isto é, a alteração da orientação da ferramenta é realizada no plano fixado pelas orientações inicial e final).

Uma interpolação de grande circunferência é executada sob as seguintes condições:

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  o ORIWKS está ativo e
  a orientação é programada através de vetores (com A3, B3, C3 ou A2, B2, C2).
- MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 e o
  ORIVECT ou ORIPLANE está ativo.
  A orientação da ferramenta pode ser programada como vetor de direção ou com posições de eixo rotativo. Se um dos comandos G `ORICONxx` ou `ORICURVE` estiver ativo ou se foram programados polinômios para o ângulo de orientação (`PO[PHI]` e `PO[PSI]`), não será executada nenhuma interpolação de grande circunferência.

Exemplo

No exemplo de programação a seguir se comprime uma circunferência que é aproximada por um polinômio. A orientação da ferramenta se movimenta de modo sincronizado sobre uma superfície periférica cônica. Mesmo se as alterações de orientação forem programadas em sequência, e não serem contínuas, a função do compressor gera um movimento suave da orientação.

```
Programação                                  Comentário
DEF INT QUANTIDADE=60
DEF REAL RAIO=20
DEF INT COUNTER
DEF REAL ANGULO
N10 G1 X0 Y0 F5000 G64

$SC_COMPRESS_CONTUR_TOL=0.05                 ; Divergência máxima do contorno = 0.05 mm
$SC_COMPRESS_ORI_TOL=5                       ; Divergência máxima da orientação = 5 graus

TRAORI
COMPCURV                                     ; É realizado um deslocamento circular, que
                                             é formado a partir dos polígonos. A orienta-
                                             ção movimenta-se sobre um cone em torno do
                                             eixo Z com um ângulo de abertura de 45 graus.
N100 X0 Y0 A3=0 B3=-1 C3=1
N110 FOR COUNTER=0 TO QUANTIDADE
N120 ANGULO=360*COUNTER/QUANTIDADE
N130 X=RAIO*cos(ANGULO) Y=RAIO*sin(ANGULO)
A3=sin(ANGULO) B3=-cos(ANGULO) C3=1
N140 ENDFOR
```

7.7 Ligar/desligar a suavização do decurso de orientação (ORISON, ORISOF)

A "Suavização do decurso de orientação" no programa de peças é ligado/desligado através dos comandos do grupo G 61. Os comandos estão ativos de forma modal.

Pré-requisitos

- Sistema com transformação de 5/6 eixos
- A função de compressor COMPCAD está ativa.

Sintaxe

```
ORISON
...
ORISOF
```

Significado

ORISON:	Ligar a suavização do decurso de orientação
ORISOF:	Desligar a suavização do decurso de orientação

Exemplo

```
Código de programa                Comentário
...
TRAORI()                          ; Ativação da transformação de orientação.
COMPCAD                           ; Ligar a função de compressor COMPCAD.
ORISON                            ; Ativação da suavização de orientação.
$SC_ORISON_TOL=1.0                ; Desvio máximo do ângulo de orientação da ferramenta
                                  = 1,0 grau.
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                            ; Desativação da suavização de orientação.
...
```

A orientação é girada em 90 graus no plano XZ de -45 até +45 graus. Através da suavização do decurso de orientação, esta não alcança mais os valores angulares máximos de -45 e +45 graus, respectivamente.

7.8 Transformação cinemática

7.8.1 Ligar a transformação dos lados frontais (TRANSMIT)

A transformação de lado frontal (TRANSMIT) é ligada no programa de peças ou na ação de sincronização através da instrução `TRANSMIT`.

Sintaxe

```
TRANSMIT
TRANSMIT(<n>)
```

Significado

`TRANSMIT`:	Ligar TRANSMIT com primeiro conjunto de dados TRANSMIT
`TRANSMIT(n)`:	Ligar TRANSMIT com <n>º bloco de dados TRANSMIT

Indicação

Uma transformação TRANSMIT ativa no canal é desligada através de:

- Desativar a transformação: `TRAFOOF`
- Ativação de uma outra transformação: por ex., `TRACYL`, `TRAANG`, `TRAORI`

7.8.2 Ligar a transformação de superfície cilíndrica (TRACYL)

A transformação do revestimento do cilindro (TRACYL) é ligada no programa de peças ou na ação de sincronização através da instrução `TRACYL`.

Sintaxe

```
TRACYL(<d>)
TRACYL(<d>,<n>)
TRACYL(<d>,<n>,<k>)
```

Significado

`TRACYL(<d>)`:	Ligar TRACYL com primeiro bloco de dados TRACYL e o diâmetro de trabalho <d>
`TRACYL (<d>,<n>)`:	Ligar TRACYL com <n>º bloco de dados TRACYL e o diâmetro de trabalho <d>
`<d>`:	Diâmetro de referência ou trabalho O valor deve ser superior a 1.

<n>:	Número de conjunto de dados TRACYL (opcional)	
	Faixa de valores:	1, 2
<k>:	O parâmetro <k> é relevante apenas para o tipo de transformação 514	
	k = 0:	Sem correção da parede da ranhura
	k = 1:	com correção da parede da ranhura
	Se o parâmetro não for indicado, atuará a posição normal parametrizada: $MC_TRACYL_DEFAULT_MODE_<n> com <n> = número do bloco de dados TRACYL	

Indicação

Uma transformação TRACYL ativa no canal é desligada através de:

- Desativar a transformação: `TRAFOOF`
- Ativação de uma outra transformação: por ex., `TRAANG`, `TRANSMIT`, `TRAORI`

Exemplo

Código de programa	Comentário
...	
N40 TRACYL(40.)	; TRACYL é ligado com o primeiro bloco de dados TRACYL e o diâmetro de trabalho 40 mm.
...	

Outras informações

Estrutura do programa

Um programa de peça para fresar uma ranhura com transformação TRACYL 513 (TRACYL com correção da parede da ranhura) é constituído, geralmente, pelos seguintes passos:

1. Selecionar ferramenta.
2. Selecionar TRACYL.
3. Selecionar o deslocamento de coordenadas (FRAME) adequado.
4. Posicionar.
5. Programar OFFN.
6. Selecionar a WRK.
7. Bloco de aproximação (entrada da correção do raio da ferramenta e aproximação da parede da ranhura).
8. Contorno da linha de centro da ranhura.
9. Cancelar a seleção da correção do raio da ferramenta.
10. Bloco de afastamento (saída da correção do raio da ferramenta e afastamento da parede da ranhura).

11.Posicionar.

12.TRAFOOF.

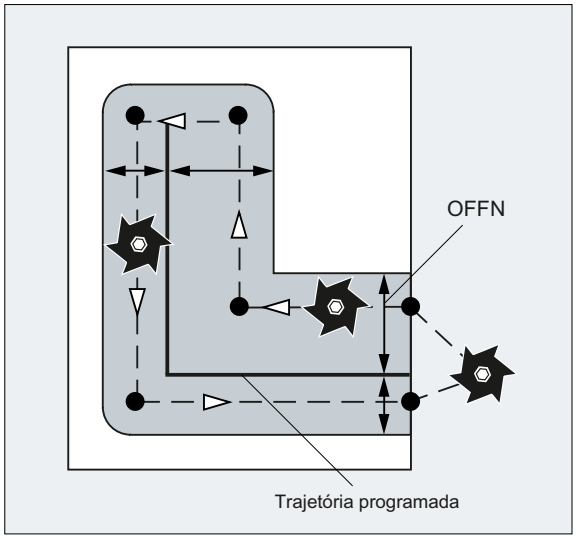
13.Selecionar novamente o deslocamento de coordenadas (FRAME) original.

Offset de contorno (OFFN)

Para fresar ranhuras com a transformação TRACYL 513, no programa de peça programa-se a linha central da ranhura e através do OFFN **meia largura da ranhura**.

Para evitar um dano na parede da ranhura, o OFFN somente é ativado com a seleção da correção do raio da ferramenta.

É possível alterar o OFFN dentro do programa de peça. Com isto, a linha central da ranhura pode ser deslocada do centro:



Indicação

OFFN deveria ser pelo menos tão grande como o raio da ferramenta, para excluir um dano na parede oposta da ranhura.

Indicação

O OFFN com TRACYL atua diferente como quando sem TRACYL. Visto que OFFN também é considerado sem TRACYL com a compensação ativa do raio da ferramenta, então o OFFN deveria ser passado novamente para zero após TRAFOOF.

ATENÇÃO
O efeito de OFFN depende do tipo de transformação No caso da transformação TRACYL 513 (TRACYL com correção da parede da ranhura) é programado para OFFN meia largura da ranhura. No caso da transformação TRACYL 512 (TRACYL sem correção da parede da ranhura) o valor de OFFN, ao contrário, atua como acréscimo para WRK.

Correção do raio da ferramenta (WRK)

No caso da transformação TRACYL 513, a WRK não se torna relativa à parede da ranhura, mas sim, é calculado com o centro programado da ranhura. Para que a ferramenta se afaste à esquerda da parede da ranhura, deve ser programado ao invés da instrução G41 a G42 ou o valor de OFFN com sinal prévio negativo.

Diâmetro da ferramenta

Com TRACYL e uma ferramenta, cujo diâmetro é menor do que a largura da ranhura, não é criada a mesma geometria da parede da ranhura, como com uma ferramenta, cujo diâmetro é igual à largura da ranhura. Para melhorar a precisão recomenda-se, selecionar o diâmetro da ferramenta somente um pouco menor do que a largura da ranhura.

Utilização de eixo

Indicação

Os seguintes eixos não podem ser utilizados como eixo de posicionamento ou eixo oscilante:

- o eixo geométrico no sentido da superfície periférica do cilindro (eixo Y)
- o eixo linear adicional na correção da parede da ranhura (eixo Z)

7.8.3 Ligar a transformação de ângulo oblíquo com ângulo programável (TRAANG)

A transformação de ângulo oblíquo com ângulo programável é ligada no programa de peças ou na ação síncrona através da instrução `TRAANG`.

Sintaxe

```
TRAANG
TRAANG()
TRAANG(, <n>)
TRAANG(<α>)
TRAANG(<α>,<n>)
```

Significado

`TRAANG`: `TRAANG()`:	Ligar TRAANG com o primeiro bloco de dados TRAANG e o último ângulo válido <α>	
`TRAANG(,<n>)`:	Ligar TRAANG com <n>º bloco de dados TRAANG e o último ângulo válido <α>	
`TRAANG(<α>)`:	Ligar TRAANG com primeiro bloco de dados TRAANG e o ângulo <α>	
`TRAANG(<α>,<n>)`:	Ligar TRAANG com <n>º bloco de dados TRAANG e o ângulo <α>	
<α>:	Ângulo do eixo que está inclinado (opcional)	
	Faixa de valores:	-90° < α < + 90°
	Sem indicação do ângulo atua a posição básica parametrizada no dado de máquina: MD2xxxx $MC_TRAANG_ANGLE_<n>	

<n>:	Número de blocos de dados TRAANG (opcional)	
	Faixa de valores:	1, 2

Indicação

Uma transformação de ângulo oblíquo TRAANG, ativa no canal, é desligada através de:

- Desativar a transformação: TRAFOOF
- Ativação de uma outra transformação: por ex., TRACYL, TRANSMIT, TRAORI

Exemplo

Código de programa	Comentário
N20 TRAANG(45)	; Ligar TRAANG com o primeiro bloco de dados TRAANG e o ângulo 45°.

7.8.4 Execução inclinada em máquinas de acabamento (G5, G7)

Os comandos G, G7 e G5 servem para simplificar a programação para a execução inclinada em máquinas de acabamento com transformação "Eixo inclinado" (TRAANG), de forma a que a penetração seja efetuada exclusivamente pelo eixo inclinado.

Apenas deve ser programada a posição final pretendida do movimento de penetração em X e Z. A posição inicial correspondente é calculada e executada pelo NC em G7, partindo da posição atual do eixo X, a posição final programada e o ângulo α do eixo inclinado.

A posição inicial é composta pelo ponto de interseção das duas retas:

- Reta paralela ao eixo Z na distância da posição atual do eixo X
- Reta paralela ao eixo inclinado através da posição final programada

Com o G5 posterior, o eixo inclinado é movido para a posição final programada.

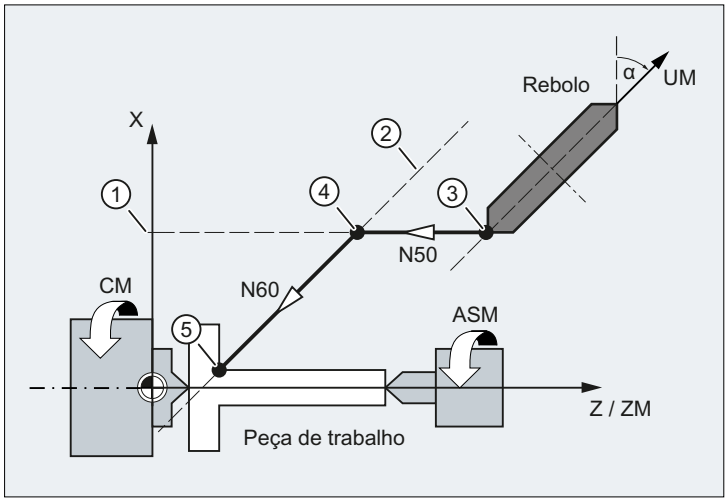
Sintaxe

```
G7 <pos. final_X> <pos. final_Z>
G5 <pos, final_X>
```

Significado

G7:	Calcular posição inicial para penetração inclinada e aproximar
G5:	Mover o eixo inclinado para a posição final programada
<pos. final_X>:	Posição final do eixo X
<pos. final_Z>:	Posição final do eixo Z

Exemplo



① Paralelo ao eixo Z na distância da posição atual do eixo X
② Paralelo ao eixo inclinado através da posição final programada
③ Posição de saída
④ Penetrar: Posição de partida
⑤ Penetrar: Posição final
X Eixo geométrico
Z Eixo geométrico
ZM Eixo de máquina
UM Eixo de máquina

Código de programa	**Comentário**
`N... G18`	`; Selecionar plano XZ.`
`N40 TRAANG(45.0)`	`; Ligar transformação TRAANG, ângulo = 45°.`
`N50 G7 X40 Z70 F4000`	`; Calcular e aproximar posição inicial.`
`N60 G5 X40 F100`	`; Mover eixo inclinado para posição final.`
`N70 ...`	

7.9 Ligar a transformação encadeada (TRACON)

A transformação encadeada é ligada no programa de peças ou na ação de sincronização através da instrução `TRACON`.

Sintaxe

```
TRACON(<Trafo_Nr>,<Par_1>,<Par_2,...)
...
TRAFOOF
```

Significado

`TRACON`:	Ligar a transformação encadeada Outra transformação ativada antes é desativada de modo implícito por `TRACON()`.		
`<Trafo_Nr>`:	Número da transformação encadeada		
	Tipo:	INT	
	Faixa de valores:	0 ... 2	
	Valor:	0, 1	Primeira/[única transformação encadeada
		2	Segunda transformação encadeada
		Nenhuma indicação	mesmo significado que 0 ou 1
		Nota: os valores diferentes 0, 1, 2 geram uma alarme de falha.	
`<Par_1>,<Par_2,...`:	Parâmetros para as transformações encadeadas entre si (por ex., ângulo do eixo inclinado) No caso de parâmetros não especificados serão ativados os pré-ajustes ou os parâmetros que foram usados por último. Com a colocação de vírgulas deve-se providenciar para que os parâmetros sejam avaliados na sequência em que forem solicitados, se os pré-ajustes devem atuar nos parâmetros precedentes. Em especial, no caso da indicação deve haver pelo menos um parâmetro antes desta vírgula, mesmo que a indicação de `<Trafo_Nr>` não seja necessária, assim, por exemplo, `TRACON( , 3.7)`. No TRACON com Transmit ou TRAORI (5ª usinagem do eixo), o segundo parâmetro Par_2 não atual como ângulo para o eixo inclinado. No TRACON com TRACYL (usinagem da superfície periférica), o Par para o diâmetro da unidade é utilizado.		
`TRAFOOF`:	Desligar a transformação (encadeada) ligada por último		

Exemplo

Código de programa	**Comentário**
`...`	

Código de programa	**Comentário**
N230 TRACON(1,45.)	; Ligar a primeira transformação encadeada.
	; A seleção da transformação ativa antes será cancelada automaticamente.
	; O parâmetro para o eixo inclinado é 45°.
...	
N330 TRACON(2,40.)	; Ligar a segunda transformação encadeada.
	; O parâmetro para o eixo inclinado é 40°.
...	
N380 TRAFOOF	. Desligar a segunda transformação encadeada.
...	

7.10 Deslocamento PTP cartesiano

7.10.1 Alternar o curso cartesiano PTP (PTP, PTPG0, PTPWOC, CP)

O curso ponto a ponto, ou seja, curso PTP será ligado/desligado no programa NC com os comandos do grupo G 49.

Os comandos estão ativos de forma modal. A predefinição é o procedimento com um movimento de percurso cartesiano (CP).

Ao contrário de CP, é realizado apenas uma transformação do ponto de destino cartesiano com PTP ativo e os eixos da máquina são movidos de forma síncrona.

Para que o ponto cartesiano possa ser calculado claramente nos valores do eixo de máquina, requer além de informações sobre posição e ângulo, que marcam as posições do eixo. As informações se dão através dos endereços ajustáveis STAT (Página 389) e TU (Página 393).

Pré-requisito

Transformação TRAORI, TRANSMIT, RCTRA ou ROBX está ativa.

Sintaxe

```
PTP / PTPG0 / PTPWOC
...
CP
```

Significado

PTP:	Ligar movimento ponto a ponto PTP A posição cartesiana programada em blocos G0 e G1 é conduzida com movimento sincronizado.
PTPG0:	Ligar o movimento ponto a ponto PTPG0 Somente nos blocos G0, a posição cartesiana programada será conduzida com um movimento sincronizado. Nos blocos G1 será comutado no movimento de percurso CP.
PTPWOC:	Ligar o movimento de ponto a ponto PTPWOC (somente possível com a transformação de orientação ativada) Como PTP, mas sem movimentos de compensação causados pelos movimentos dos eixos rotativos e eixos de orientação.
CP:	Desligar o movimento ponto a ponto e ligar o movimento de percurso CP Com o CP será executado o movimento de percurso cartesiano.

Indicação

PTPWOC

Em combinação com uma transformação RCTRA ou ROBX, a utilização de PTPWOC não faz sentido.

Exemplos

Veja:

- Exemplo 1: Deslocamento PTP de um robô de 6 eixos com transformação ROBX (Página 396)
- Exemplo 2: Deslocamento PTP com transformação genérica de 5 eixos (Página 397)
- Exemplo 3: PTPG0 e TRANSMIT (Página 397)

7.10.2 Indicar a posição das articulações (STAT)

A indicação da posição com coordenadas cartesianas e indicação da orientação da ferramenta não são suficientes para estabelecer claramente a posição da máquina, porque várias posições de articulação estão disponíveis para mesma orientação de ferramenta. Dependendo da cinemática tratada, podem existir até 8 posições de articulação diferentes. Estas posições de articulações diferentes são específicas de acordo a transformação.

Para evitar ambiguidade, a posição das articulações é fornecida no endereço STAT.

Indicação

O comando considera os valores STAT programados apenas para movimentos PTP. Em movimentos CP eles serão ignorados porque no processo de transformação ativa de uma mudança de posição, isto não é normalmente possível. A posição para o ponto de destino é tomada do ponto de partida do processo com um CP ativo.

Sintaxe

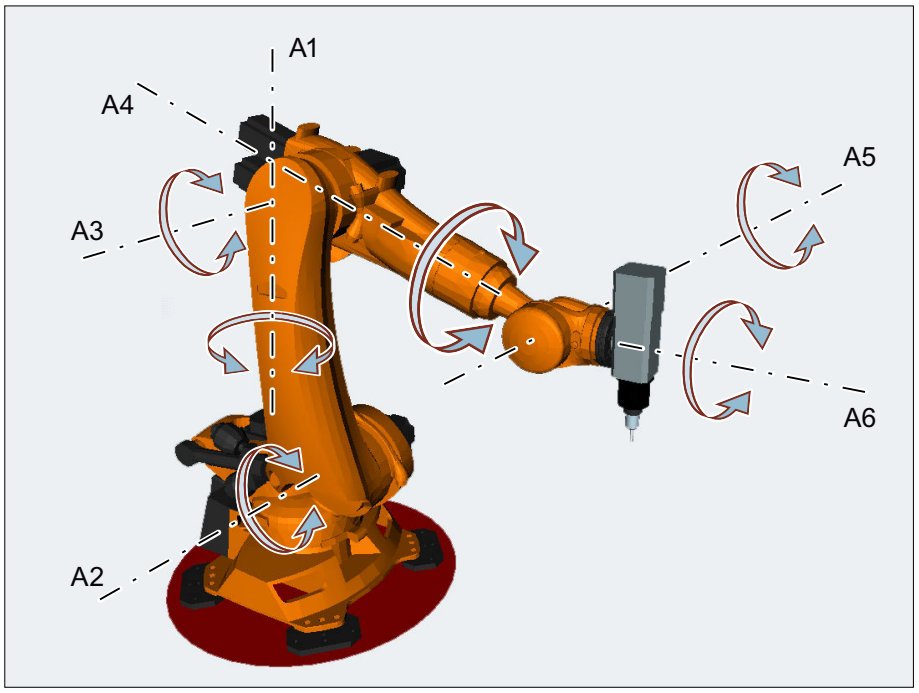
```
STAT=<valor>
```

Significado

STAT:	Endereço ajustável para especificação das posições de articulações
<valor>:	Valor binário ou decimal Contém um bit para cada uma das posições possíveis. O significado dos bits será definido por cada transformação.

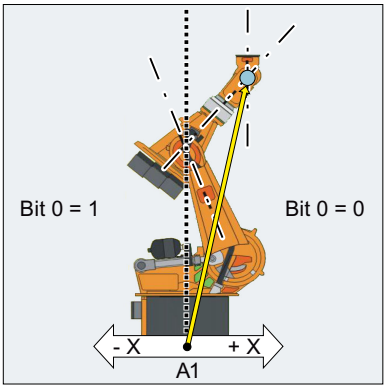
A aplicação de STAT é ilustrada pelo exemplo de um robô de braço articulado de 6 eixos com fuso de fresagem. A transformação cinemática deve ocorrer através da transformação robótica

ROBX (requisito prévio: ciclo de compilação "RMCC/ROBX Transformação robótica ampliada" carregado e ativo).

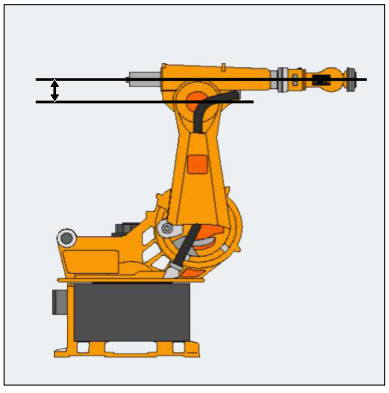


Os eixos A1, A2 e A3 são os eixos principais do robô de braço articulado. Com os eixos principais são posicionados, dentro do espaço de trabalho, os eixos A4, A5 e A6 denominadas eixos de cabeça ou eixos de mão. Através das possibilidades adicionais dos eixos de mão, o fuso de fresagem pode ser orientado no espaço de modo adequado para a tarefa de usinagem. Neste processo, com a mesma orientação da ferramenta são possíveis diferentes posições da articulação.

A seleção das posições da articulação, que devem ser utilizadas para a usinagem, ocorre através da programação de Bit 0 ... 2 do endereço ajustável STAT:

<table>
<tr><td rowspan="3">Bit 0</td><td colspan="2">Posição do ponto de interseção dos eixos manuais (A4, A5, A6)</td><td rowspan="3"> 

Exemplo: O ponto de interseção dos eixos manuais encontra-se na área de base</td></tr>
<tr><td>=0</td><td>Área de base (Shoulder Right)
O robô se localiza na área de base, se o valor X do ponto de interseção dos eixos manuais, associados ao sistema de coordenação A1, for positivo.</td></tr>
<tr><td>= 1</td><td>Área suspensa sobre a cabeça (Shoulder Left)
O robô se localiza na área suspensa, se o valor X do ponto de interseção, associado ao sistema de coordenação A1, for negativo.</td></tr>
</table>

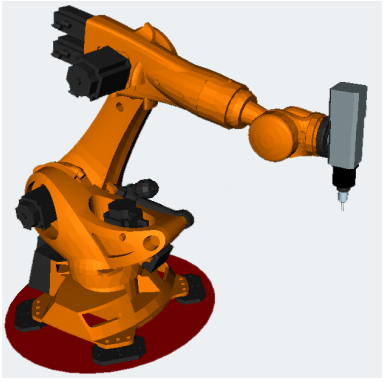
Bit 1	posição do eixo 3 O ângulo no qual o valor de bit 1 muda depende do tipo de robô. Para robôs, cujos eixos 3 e 4 se cruzam, são aplicáveis:	
	=0	A3 < 0° (**Elbow Down**)
	= 1	A3 ≥ 0° (**Elbow Up**)
	Nota: Nos robôs com um deslocamento entre o eixo 3 e 4, o ângulo no qual o valor de bit 1 muda, é dependente do tamanho desse deslocamento.	 Offset entre A3 e A4
Bit 2	posição do eixo 5	
	=0	A5 ≤ 0° (**No Handflip**)
	= 1	A5 > 0° (**Handflip**)

Exemplo de programa:

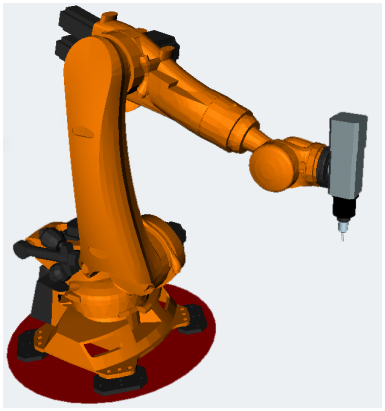
Código de programa	**Comentário**
...	
N14 T="T8MILLD20" D1	; $TC_DP3[1,1]=132.95
N16 ORIMKS	
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 **STAT=...** F2000	; O valor STAT determina as posições da articulação (veja abaixo).
...	

`STAT=1 ('B001')`

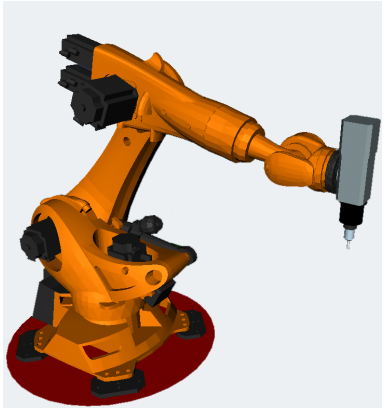
→ Shoulder Left
→ Elbow Down
→ No Handflip



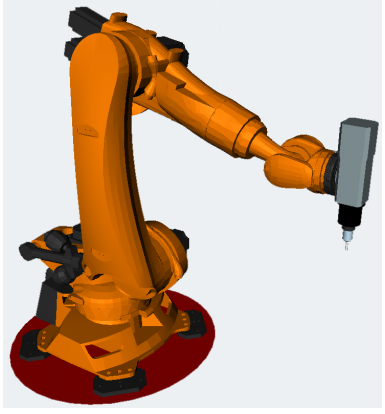
`STAT=2 ('B010')`
→ Shoulder Right
→ Elbow Up
→ No Handflip



`STAT=5 ('B101')`
→ Shoulder Left
→ Elbow Down
→ Handflip



`STAT=6 ('B110')`
→ Shoulder Right
→ Elbow Up
→ Handflip



TRANSMIT

Com TRANSMIT, o endereço STAT é usado para desfazer a ambiguidade relativa ao polo.

Se o eixo rotativo precisa ser girado em 180º ou quando o contorno passaria no CP através do polo, vale:

Bit 0	Relevante apenas para $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 e 2:	
	=0	eixo rotativo movido em +180º e gira no sentido dos ponteiros do relógio.
	= 1	eixo rotativo movido em -180º e gira no sentido contrário ao dos ponteiros do relógio.
Bit 1	Relevante apenas para $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0:	
	=0	Será movido através do polo. O eixo rotativo não gira.
	= 1	Será movido em torno do polo. O bit 0 do STAT é relevante.

7.10.3 Especificar o ângulo do eixo (TU)

Para mover eixos rotacionais, também ângulo do eixo, maiores que + 180 ° ou menores - 180° graus, sem estratégia processual especial (tais como pontos intermediários) sob do endereço ajustável TU, os sinais do ângulo do eixo devem ser especificados.

Indicação

O comando considera os valores TU programados apenas para movimentos PTP. Eles serão ignorados nos movimentos CP.

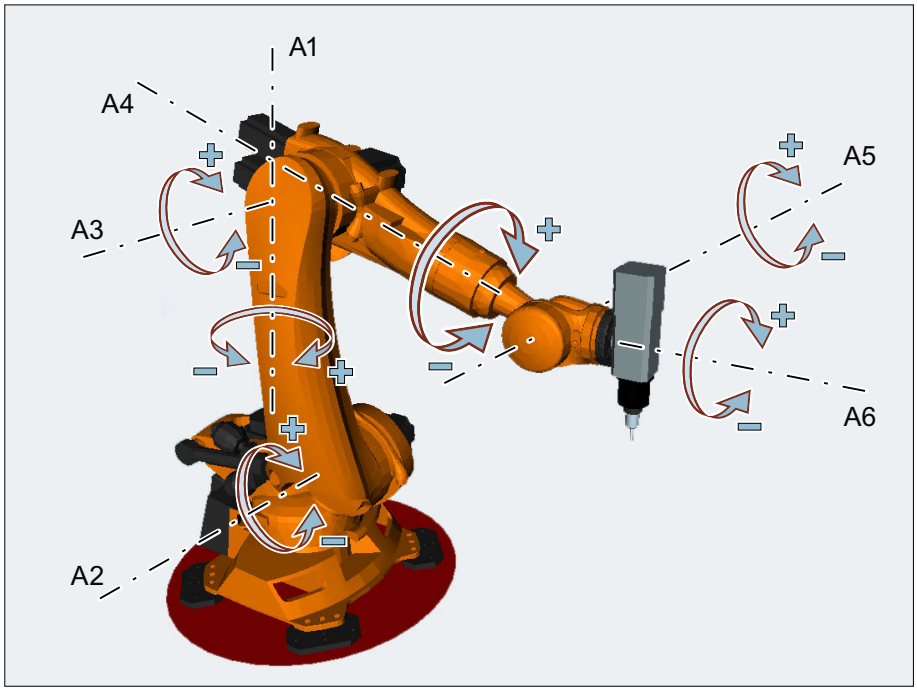
Sintaxe

```
TU=<valor>
```

Significado

TU:	Endereço ajustável para especificação dos sinais do ângulo do eixo			
<valor>:	Valor binário ou decimal Para cada eixo que entra em a transformação, há um bit indicando o sinal do ângulo do eixo (θ) e, por conseguinte, a direção da travessia.			
	Bit	=0	Sinal do ângulo do eixo: +	Área do ângulo do eixo: 0° ≤ θ < 360°
		= 1	Sinal do ângulo do eixo: -	Área do ângulo do eixo: - 360° < θ < 0°

Exemplo: Robô com braço articulado de 6 eixos



Bit	Significado	Valor	Sinal do ângulo do eixo	Ângulo do eixo
Bit 0 [1]	Sinal para o ângulo do eixo de A1	=0	+	≥ 0°
		= 1	-	< 0°
Bit 1 [1]	Sinal para o ângulo do eixo de A2	=0	+	≥ 0°
		= 1	-	< 0°
Bit 2 [1]	Sinal para o ângulo do eixo de A3	=0	+	≥ 0°
		= 1	-	< 0°
Bit 3 [1]	Sinal para o ângulo do eixo de A4	=0	+	≥ 0°
		= 1	-	< 0°
Bit 4 [1]	Sinal para o ângulo do eixo de A5	=0	+	≥ 0°
		= 1	-	< 0°
Bit 5 [1]	Sinal para o ângulo do eixo de A6	=0	+	≥ 0°
		= 1	-	< 0°

[1] Os números reais TU-Bit resultam dos números dos eixos de canal dos eixos do robô! No exemplo, os eixos do robô (A1 até A6) são os primeiros seis eixos no canal, consequentemente são utilizados TU-Bits 0 ... 5. Em uma atribuição diferente de eixo de canal dos eixos do robô, os números TU-Bit dos eixos do robô mudariam de modo correspondente (por ex.: os eixos do robô são o 3º até o 8º eixo de canal, significa que são utilizados TU-Bits 2 ... 7 para os eixos do robô).

TU=19 (corresponde a TU='B010011) significaria:

Bit	Valor		Ângulo do eixo
0	= 1	⇒	$\theta_{A1} < 0°$
1	= 1	⇒	$\theta_{A2} < 0°$

2	=0	⇒	$\theta_{A3} \geq 0°$
3	=0	⇒	$\theta_{A4} \geq 0°$
4	= 1	⇒	θA5 < 0°
5	=0	⇒	$\theta_{A6} \geq 0°$

Indicação

Para eixos com uma área de deslocamento > ±360° sempre será percorrido no caminho mais curto, já que a posição do eixo não é determinada unicamente pela informação TU.

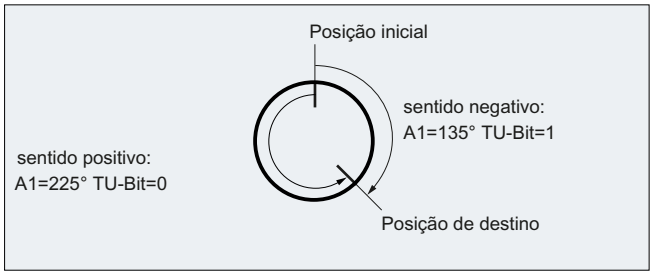
Na ausência de programação TU para uma posição, o deslocamento ocorre pelo percurso mais curto ou mais longo em dependência de MD30455 $MA_MISC_FUNCTION_MASK (veja o capítulo "Consideração dos Limites SW no Deslocamento PTP" no Manual de Funcionamento Funções de Ampliação).

TRANSMIT

No percurso PTP no TRANSCom ativo, o endereço TU não tem significado!

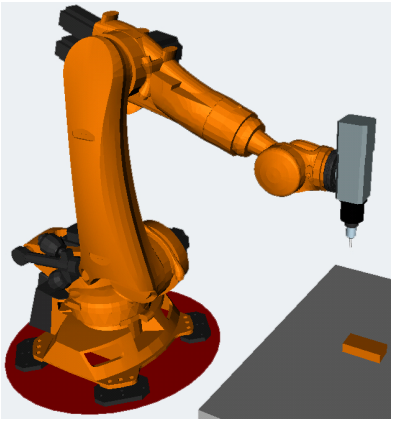
Exemplo

A posição de eixo rotativo na ilustração a seguir pode ser aproximada em sentido negativo ou positivo. A posição angular é programada no endereço A1. Só através das indicações do TU que o sentido do percurso ficará claro.



7.10.4 Exemplo 1: Deslocamento PTP de um robô de 6 eixos com transformação ROBX

No exemplo de aplicação a seguir são mostrados o deslocamento PTP e os comandos NC relacionados a ele.



Esquema 7-3 Robô com braço articulado de 6 eixos com fuso de fresagem

```
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 Exemplo 2: Deslocamento PTP com transformação genérica de 5 eixos

Suposição: Existe uma cinemática ortogonal CA como base.

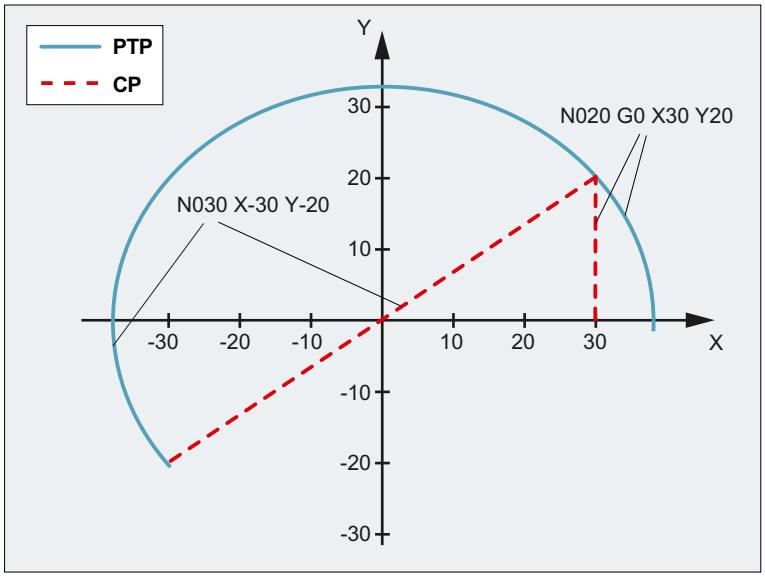
Código de programa	Comentário
TRAORI	; Transformação de cinemática CA ativa
PTP	; Ativar deslocamento PTP
N10 A3=0 B3=0 C3=1	; posições dos eixos rotativos C=0 A=0
N20 A3=1 B3=0 C3=1	; posições dos eixos rotativos C=90 A=45
N30 A3=1 B3=0 C3=0	; posições dos eixos rotativos C=90 A=90
N40 A3=1 B3=0 C3=1 **STAT=1**	; posições dos eixos rotativos C=270 A=-45

Selecionar uma posição de aproximação única da posição de eixo rotativo:

No bloco N40, os eixos rotativos percorrem, através da programação **STAT=1**, o percurso mais longo a partir de seu ponto de partida (C=90, A=90) ao ponto final (C=270, A=-45). Com **STAT=0**, os eixos rotativos, porém, percorreriam o percurso mais curto ao ponto final (C=90, A=45).

7.10.6 Exemplo 3: PTPG0 e TRANSMIT

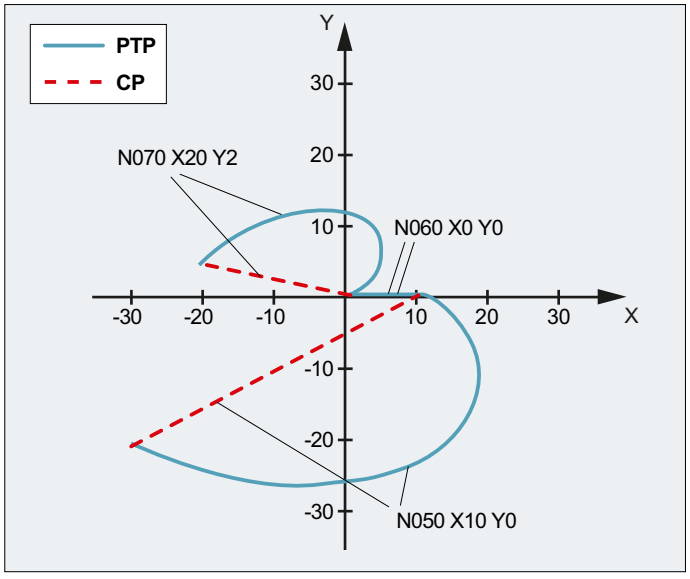
Contornar o polo com PTPG0 e TRANSMIT



Código de programa	Comentário
N001 G0 X30 Z0 F10000 T1 D1 G90	; Posição de saída da dimensão absoluta
N002 SPOS=0	
N003 **TRANSMIT**	; Transformação TRANSMIT
N010 **PTPG0**	; Para cada bloco G0 automaticamente PTP e depois novamente CP.

Código de programa	Comentário
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Afastar do polo com PTPG0 e TRANSMIT



Programação	Comentário
N001 G0 X90 Z0 F10000 T1 D1 G90	; Posição de saída
N002 SPOS=0	
N003 **TRANSMIT**	; Transformação TRANSMIT
N010 **PTPG0**	; Para cada bloco G0 automaticamente PTP e depois novamente 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 Condições gerais na seleção de uma transformação

Função

A seleção das transformações é possível através do programa de peça ou MDA. Aqui se deve observar:

- Não se insere um bloco intermediário de movimento (chanfros/raios).
- Uma sucessão de blocos Spline deve estar concluída, caso contrário aparece uma mensagem.
- A correção fina de ferramenta deve estar desselecionada (FTOCOF); caso contrário aparece uma mensagem.
- A correção de raio da ferramenta deve estar desselecionada (G40); caso contrário aparece uma mensagem.
- Um correção ativa do comprimento de ferramenta é assumida pelo comando na transformação.
- O atual Frame ativo antes da transformação é desselecionado pelo comando.
- Um limite ativo de área de trabalho é desselecionado pelo comando nos eixos afetados pela transformação (equivale ao WALIMOF).
- A monitoração da área de proteção é desselecionada.
- O modo de controle da trajetória e a suavização são cancelados.
- Todos os eixos especificados no dado de máquina devem ser sincronizados relativos ao bloco.
- Eixos trocados são destrocados; caso contrário aparece uma mensagem.
- É emitida uma mensagem para os eixos dependentes.

Troca de ferramentas

Uma troca de ferramentas somente é permitida com a correção de raio da ferramenta desselecionada.

Uma troca da correção do comprimento da ferramenta e uma seleção/desseleção da correção do raio da ferramenta não podem ser programadas no mesmo bloco.

Mudança de Frames

São permitidas todas instruções relativas ao sistema de coordenadas básico (FRAME, correção do raio da ferramenta). Porém, uma mudança de Frames com G91 (dimensão incremental) não é tratada de modo especial, diferente como na transformação inativa. O incremento a ser executado é avaliado no sistema de coordenadas da peça de trabalho do novo Frame, independente de qual Frame estava estiver no bloco anterior.

Exceções

Os eixos afetados pela transformação não podem ser utilizados:

- como eixo Preset (alarme),
- para a aproximação do ponto fixo (alarme),
- para referenciar (alarme).

7.12 Desselecionar a transformação (TRAFOOF)

Com o procedimento pré-definido TRAFOOF são desativadas todas transformações e Frames ativos.

Indicação

Para a desativação da transformação, aplicam-se as mesmas limitações (Página 399) que para a seleção.

Em seguida, os Frames necessários devem ser ativados através de uma nova programação.

Sintaxe

```
...
TRAFOOF
```

Significado

`TRAFOOF`:	Desligar todas as transformações ativas / Frames

Cadeias cinemáticas 8

8.1 Supressão de componentes (DELOBJ)

A função `DELOBJ()` "deleta" componentes através da redefinição da variáveis de sistema dispostas no seu valor padrão:

- Elementos da cadeia cinemática
- Áreas de proteção, elementos da área de proteção e pares de colisão
- Dados de transformação

Sintaxe

```
[<RetVal>=] DELOBJ(<CompType>[,,,<NoAlarm>)])
[<RetVal>=] DELOBJ(<CompType>,<Index1>[,,<NoAlarm>])
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][,<Index2>][,<NoAlarm>])
```

Significado

DELOBJ:	Supressão de elementos de cadeias cinemáticas, áreas de proteção, elementos da área de proteção, pares de colisão e dados de transformação
<CompType>:	Tipo do componente a ser suprimido
	Tipo de dado: STRING
	Valor: "KIN_CHAIN_ELEM" **significado**: variáveis de sistema de todos os elementos cinemáticos: $NK_...
	Valor: "KIN_CHAIN_SWITCH" **significado**: variável do sistema $NK_SWITCH[<i>]
	Valor: "KIN_CHAIN_ALL" **Significado**: Todos os elementos cinemáticos e interruptores. Equivalentecom a chamada sucessiva do DELOBJ com "KIN_CHAIN_ELEM" e "KIN_CHAIN_SWITCH"
	Valor: "PROT_AREA" **significado**: variáveis de sistema das áreas de proteção: • $NP_PROT_NAME • $NP_CHAIN_NAME • $NP_CHAIN_ELEM • $NP_1ST_PROT
	Valor: "PROT_AREA_ELEM" **significado**: variáveis do sistema dos elementos de áreas de proteção da máquina e / ou áreas de proteção da ferramenta automática: • $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
	Valor: "PROT_AREA_COLL_PAIRS" **significado**: variáveis de sistema dos pares de colisão: • $NP_COLL_PAIR • $NP_SAFETY_DIST
	Valor: "PROT_AREA_ALL" **significado**: Todas as áreas de proteção, elementos da área de proteção e pares de colisão (variáveis de sistema $NP_...) Equivalente com a chamada sucessiva do DELOBJ com "PROT_AREA", "PROT_AREA_ELEM" e "PROT_AREA_COLL_PAIRS"
	Valor: "TRAFO_DATA" **significado**: variáveis de sistema de todas as transformações $NT_...

<Index1>:	Índice do primeiro componente a ser suprimido (**opcional**)	
	Tipo de dado:	INT
	Valor padrão:	-1
	Faixa de valores:	-1 ≤ x ≤ (número máximo de componentes projetados -1)
	Valor	**Significado**
	0, 1, 2, ...	Índice do componente a ser eliminado.
	-1	Todos os componentes do tipo especificado serão excluídos. <Índice2> não será avaliado.
<Index2>:	Índice dos últimos componentes a serem eliminados (**opcional**) Se não estiver <Index2> programado, apenas as variáveis de sistema dos componentes referenciados <Index1> serão apagadas.	
	Tipo de dado:	INT
	Valor padrão:	Somente as variáveis de sistema do componente referenciado <Index1> serão excluídas.
	Faixa de valores:	<Index1> < x ≤ (número máximo de componentes projetados -1)
<NoAlarm>:	Eliminação do alarme (**opcional**)	
	Tipo de dado:	BOOL
	Valor padrão:	FALSE
	Valor	**Significado**
	FALSE	Em caso de erro (<RetVal> < 0) a execução do programa será interrompida e um alarme será mostrado.
	TRUE	Em caso de erro a execução do programa será interrompida e nenhum alarme será mostrado. Caso de aplicação: Reação específica do usuário corresponde ao valor de devolução
<RetVal>:	Valor de devolução da função	
	Tipo de dado:	INT
	Faixa de valores:	0, -1, -2, ... -7
	Valor	**Significado**
	0	Nenhuma ocorrência de erro.
	-1	Chamada da função sem parâmetro. Ao menos o parâmetro <CompType> deve ser indicado.
	-2	<CompType> indica componentes desconhecidos
	-3	<Índice1> é menor que -1
	-4	<Índice1> é maior que o número projetado de componentes
	-5	<Índice1> tem na supressão de um componentes **grupo** um valor diferente de -1
	-6	<Índice2> é menor que <Índice1>
	-7	<Índice2> é maior que o número projetado de componentes

8.2 Averiguação do índice por nome (NAMETOINT)

Nos campos variáveis do sistema de tipoSTRING estão inseridos nomes específicos ao usuário. Com base no identificador das variáveis do sistema bem como o nome, a função `NAMETOINT()` () determina o valor do índice do nome pertencente, abaixo do qual é armazenado no campo variável sistema.

Sintaxe

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

Significado

`NAMETOINT`:	Pesquisa do índice de variável do sistema	
`<SysVar>`:	Nome do campo de variável do sistema do tipo STRING	
	Tipo de dado:	STRING
	Faixa de valores:	Nome do campo de variável do sistema do tipo NC STRING
`<Name>`:	A cadeia de caracteres e nome para o índice variável do sistema devem ser pesquisados.	
	Tipo de dado:	STRING
`<NoAlarm>`:	Eliminação do alarme (**opcional**)	
	Tipo de dado:	BOOL
	Valor padrão:	FALSE
	Valor	**Significado**
	TRUE	Em caso de erro a execução do programa será interrompida e nenhum alarme será mostrado. Caso de aplicação: Reação específica do usuário corresponde ao valor de devolução
	FALSE	Em caso de erro(<RetVal> < 0) a execução do programa será interrompida e um alarme será mostrado.
`<RetVal>`:	Índice das variáveis de sistema ou mensagem de erro	
	Tipo de dado:	INT
	Faixa de valores:	-1 ≤ x ≤ (número máximo de componentes projetados -1)
	Valor	**Significado**
	≥ 0	O nome procurado é encontrado sob o índice de variáveis de sistema especificado.
	-1	O nome buscado não foi encontrado e ocorreu um erro.

Exemplo

Código de programa	**Comentário**
`DEF INT Índice`	
`$NP_PROT_NAME[27] = "Cobertura"`	
`...`	

Código de programa	Comentário
`Índice = NAMETOINT("$NP_PROT_NAME","Cobertura")`	`Posição centena de milhar == 27`

Prevenção de colisão com cadeias cinemáticas 9

Indicação

Áreas de proteção

As áreas de proteção referidas nos seguintes capítulos referem-se à função "Modelação geométrica da máquina"

Literatura:
Manual de funções especiais, capítulo "Modelação geométrica da máquina"

9.1 Teste sobre o par de colisão(COLLPAIR)

A função `COLLPAIR(...)` apura se duas áreas formam um par de colisão.

Sintaxe

```
[<RetVal> =] COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>)])
```

Significado

`COLLPAIR`:	Comprovar que pertence a um par de colisão		
`<RetVal>`:	Valor de devolução da função		
	Tipo de dado:	INT	
	Valor:	≥ 0	Ambas as áreas de proteção formam um par de colisão. Valor de devolução == índice de par de colisão m (veja $NP_COLL_PAIR)
		-1	Serão indicados menos de dois Strings ou pelo menos um dos dois é o String zero.
		-2	A área de proteção indicada no primeiro parâmetro não foi encontrada.
		-3	A área de proteção indicada no segundo parâmetro não foi encontrada.
		-4	Nenhuma das duas áreas de proteção indicadas foram encontradas.
		-5	Amas as áreas de proteção foram encontradas, mas não junto com um par de colisão.
`<Name_1>`:	Nome da primeira área de proteção		
	Tipo de dado:	STRING	
	Faixa de valores:	Nome da área de proteção parametrizada	
`<Name_2>`:	Nome da segunda área de proteção		
	Tipo de dado:	STRING	
	Faixa de valores:	Nome da área de proteção parametrizada	
`<NoAlarm>`:	Eliminação de alarme **(opcional)**		
	Tipo de dado:	BOOL	
	Valor:	FALSE (Default)	Em caso de erro(<RetVal> < 0), a execução do programa será interrompida e um alarme será mostrado.
		TRUE	Em caso de erro, a execução do programa será interrompida e nenhum alarme será mostrado. Caso de aplicação: Reação específica do usuário corresponde ao valor de devolução

9.2 Solicitar novo cálculo do modelo de máquina da prevenção de colisão (PROTA)

Quando são gravadas variáveis de sistema da cadeia cinemática $NK_..., da modelação geométrica da máquina ou da prevenção de colisão $NP_... no programa de peças, é necessário acessar, em seguida, o procedimento `PROTA`, para que a alteração no modelo de máquina interno de NC da prevenção de colisão entre em vigor.

Sintaxe

```
PROTA[(<Par>)]
```

Significado

`PROTA`:	Solicitar novo cálculo do modelo de máquina da prevenção de colisão		
	• Aciona uma parada de pré-processamento. • Deve estar em posição isolada dentro do bloco.		
`<Par>`:	Parâmetro (**opcional**)		
	Tipo de dado:	STRING	
	Valor:	---	Sem parâmetro. Realiza-se um novo cálculo do modelo de máquina. Os status das áreas de proteção são conservados.
		"R"	Realiza-se um novo cálculo do modelo de máquina. As áreas de proteção são colocadas nos seus status de inicialização conforme $NP_INIT_STAT.

Condições gerais

Simulação

O procedimento `PROTA` não deve ser utilizado nos programas de peças em relação à simulação (simNC).

Exemplo: Evitar a chamada de `PROTA`, durante o período de ativação da simulação.

```
Código de programa            Comentário
...
IF $P_SIM == FALSE            ; IF Simulação não ativa
   PROTA                      ; THEN recalcular o modelo de colisão
ENDIF                         ; ENDIF
...
```

Veja também

Estabelecer a condição da área de proteção (PROTS) (Página 410)

9.3 Estabelecer a condição da área de proteção (PROTS)

O procedimento `PROTS(...)` ajusta o status das áreas de proteção para o valor indicado.

Sintaxe

```
PROTS(<State>[,<Name_1>,...,<Name_n>])
```

Significado

`PROTS`:	Ajustar o status de áreas de proteção		
	• Deve estar em posição isolada dentro do bloco.		
`<State>`:	Status que deve ser estabelecido nas áreas de proteção		
	Tipo de dado:	CHAR	
	Valor:	"A" ou "a"	Status: Ativo
		"I"ou "i"	Status: Inativo
		"P"ou "p"	Status: Pré-ativado ou sob comando PLC [1]
		"R"ou "r"	Status: Valor interno de NC do status de inicialização [2]
`<Name_1> ... <Name_n>`:	Nome de uma ou várias áreas de proteção que devem ser ajustadas para o status indicado (**opcional**) Sem indicação de nome, o status indicado é ajustado para todas as áreas de proteção definidas.		
	Tipo de dado:	STRING	
	Faixa de valores:	Nome da área de proteção parametrizada	
	Aviso: O número máximo de áreas de proteção que podem ser indicadas como parâmetro depende apenas do número máximo possível de caracteres por linha do programa.		

[1] A ativação / desativação ocorre através de: DB10.DBX234.0 - DBX241.7

[2] O status é ajustado para o valor interno de NC do status de inicialização, ou seja para o valor que a variável de sistema $NP_INIT_STAT tinha no momento da última chamada de PROTA(...) (Página 409).

9.4 Determinação da distância de duas áreas de proteção (PROTD)

A função `PROTD(...)` calcula a distância de duas áreas de proteção.

Características da função:

- O cálculo da distância acontece independente do status da área de proteção (ativado, desativado, pré-ativado).
- Para o cálculo da distância das duas áreas de proteção, apenas são chamados tais elementos de áreas de proteção, os quais são identificados com $NP_USAGE = "C" ou "A". Elementos de áreas de proteção da área de proteção, os quais são identificados com $NP_USAGE = "V", não são considerados.
- Áreas de proteção, nas quais todos elementos de áreas de proteção da área de proteção são identificados com $NP_USAGE = "V", não podem ser chamados para o cálculo da distância.
- O cálculo da distância acontece com as posições válidas no final do bloco de processo.
- Sobreposições que são incluídas no cálculo do curso principal, por ex. deslocamento DRF ou deslocamentos externos do ponto zero entram no cálculo da distância com os valores válidos no **momento da interpretação** da função.

Indicação

Sincronização

Na utilização da função `PROTD`, é de exclusiva responsabilidade do usuário sincronizar o curso principal e o avanço por meio de uma parada de pré-processamento `STOPRE`.

Colisão

Se houver uma colisão entre as áreas de proteção indicadas, a função fornece uma distância de 0,0. Uma colisão existe quando há contato ou penetração entre as duas áreas de proteção.

A distância de segurança para o teste de colisão (MD10622 $MN_COLLISION_SAFETY_DIST) não é considerada no cálculo da distância.

Sintaxe

```
[<RetVal> =] PROTD([<Name_1>],[<Name_2>],VAR <Vector>[,<System>])
```

Significado

`PROTD`:	Calcular a distância das duas áreas de proteção indicadas	
	• Deve estar em posição isolada dentro do bloco.	
`<RetVal>`:	Valor de devolução da função: Valor absoluto da distância das duas zonas de proteção ou 0,0 em Colisão (ver acima: Parágrafo de Colisão)	
	Tipo de dado:	REAL
	Faixa de valores:	0,0 ≤ x ≤ +máx. Valor REAL

<Name_1>, <Name_2>:	Nomes das duas áreas de proteção para as quais a distância entre uma e outra deve ser calculada (**opcional**)		
	Tipo de dado:	STRING	
	Faixa de valores:	Nome da área de proteção parametrizada	
	Valor padrão:	"" (string vazio) Caso não seja indicada nenhuma área de proteção, a função calcula a menor distância atual de todas as áreas de proteção ativadas e pré-ativadas contidas no modelo de colisão.	
<Vector>:	Valor de retorno: Vetor de distância tridimensional da área de proteção<Name_2> para área de proteção<Name_1> com: • <Vector>[0]: Coordenada X no sistema mundial de coordenadas • <Vector>[1]: Coordenada Y no sistema mundial de coordenadas • <Vector>[2]: Coordenada Z no sistema mundial de coordenadas Em caso de colisão: <Vector> == vetor zero		
	Tipo de dado:	VAR REAL [3]	
	Faixa de valores:	<Vector> [n]: 0,0 ≤ x ≤ ±max. REAL-Wert	
<System>:	Sistema de medica (inch / métrico) para distância e vetor de distância (**opcional**)		
	Tipo de dado:	BOOL	
	Valor:	FALSE (Default)	Sistema de medidas correspondente ao comando G, atualmente ativa, a partir do grupo G 13 (G70, G71, G700, G710).
		TRUE	Sistema de medida correspondente ao sistema básico ajustado: MD52806 $MN_ISO_SCALING_SYSTEM

Transformações com cadeias cinemáticas 10

10.1 Ligar transformação (TRAFOON)

Uma transformação definida com sequências cinemáticas é ativada com o procedimento pré-definido TRAFOON. A chamada precisa permanecer sozinha em um bloco.

Indicação

Alternativamente, uma transformação definida com sequências cinemáticas também pode ser ativada através de comandos de voz convencionais, como por ex. TRAORI ou TRANSMIT. Para isso, precisa ser introduzido nas variáveis de sistema $NT_TRAFO_INDEX um valor correspondente diferente de zero.

Para mais informações sobre $NT_TRAFO_INDEX ver "Manual de listagem de variáveis de sistema".

Sintaxe

```
TRAFOON(<Trafoname>,<Diameter>,<k>)
```

Significado

TRAFOON:	Procedimento para ativação de uma transformação definida com sequências cinemáticas		
<Trafoname>:	Nome do bloco de dados da transformação		
	Tipo de dado:	STRING	
	Faixa de valores:	Todos os nomes de blocos de dados da transformação definidos através de $NT_NAME	
	Indicação: o nome do bloco de dados da transformação precisa ser único. Ele deve ocorrer apenas uma vez em $NT_NAME.		
<Diameter>:	Diâmetro de referência ou de trabalho (somente TRACYL)		
	Tipo de dado:	REAL	
	O valor deve ser > 1.		
<k>:	Define a utilização da correção da parede da ranhura (somente TRACYL).		
	Tipo de dado:	BOOL	
	Valor:	FALSE	Sem correção da parede da ranhura
		TRUE	com correção da parede da ranhura
	Corresponde ao tipo de transformação TRACYL 514 (correção da parede da ranhura programável). Se <k> não for especificado, fica ativo o ajuste parametrizado do bit 10 em $NT_CNTRL[<n>].		

Exemplo

Código de programa	Comentário
`TRAFOON["Trans_1"]`	`Ativa a Transformação com o nome Trans_1.`

10.2 Modificação da transformação de orientação após a medição da máquina (CORRTRAFO)

Em máquinas com transformações de orientação definidas por meio de sequências cinemáticas, o usuário pode utilizar a função pré-definida CORRTRAFO para modificar os vetores offset ou os vetores de direção, após uma medição da máquina, no modelo cinemático da máquina.

Sintaxe

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode> [,
<No_Alarm>])
```

Significado

CORRTRAFO:	Chamada da função

<Corr_Condição>:	Valor de devolução da função		
	Tipo de dado:	INT	
	Valores:	0	A função foi executada sem erro.
		1	Não há nenhuma transformação ativa.
		2	A transformação ativa atual não é uma transformação de orientação.
		3	A transformação de orientação ativa não foi definida com sequências cinemáticas.
		10	O parâmetro da chamada <Corr_Index> é negativo.
		11	O parâmetro da chamada <Corr_Mode> é negativo.
		12	Referência não válida em uma seção de uma sequência da peça (1º dígito de <Corr_Index>). O valor não pode ser maior que a quantidade de eixos de orientação na sequência parcial.
		13	Referência não válida no eixo de orientação de uma sequência da peça (1º dígito de <Corr_Index>). O valor precisa ser menor do que a quantidade de eixos de orientação na sequência parcial.
		14	Referência não válida em uma sequência da peça (10º dígito de <Corr_Index>). Somente os valores 0 e 1 são permitidos (referência a sequência da peça ou de ferramenta). Esse número de erro também ocorre quando a sequência da peça que se refere a <Corr_Index> não existe.
		15	Na seção que deve ser referida com o parâmetro <Corr_Index>, não há nenhum elemento de correção definido ($NT_CORR_ELEM_P ou $NT_CORR_ELEM_T).
		20	Modo de correção não válido (1º dígito de <Corr_Mode>). Apenas os valores 0 e 1 são permitidos.
		21	Modo de correção não válido (10º e/ou 100º dígito de <Corr_Mode>). Na gravação de uma direção de eixo, apenas o 1º dígito precisa ser diferente de zero.
		30	O 100º dígito de <Corr_Mode> não é válido. Apenas os valores 0 e 1 são permitidos.
		31	O 1000º dígito de <Corr_Mode> não é válido. Apenas os valores 0 e 1 são permitidos.
		40	O vetor de direção que deve ser adotado como direção do eixo, é o vetor nulo. Esse erro somente deve ocorrer se o 1000º dígito de <Corr_Mode> for igual a 0. Se o 1000º dígito desse parâmetro for igual a 1 (monitoramento da correção máxima desativado), o vetor nulo também pode ser gravado.
		41	Na correção de um vetor de offset, o desvio com relação aos valores atuais precisa ser no mínimo uma coordenada maior do que o valor máximo especificado pelo dado de ajuste SD41610 $SN_CORR_TRAFO_LIN_MAX. O parâmetro <Corr_Vect> é sobrescrito por um vetor de erro. Isso também é válido se a usinagem for interrompida com alarme (ver parâmetro <No_Alarm>). O vetor de erro contém a diferença com o sinal correto, entre o valor da correção determinado e o limite, nos componentes onde o valor de correção ultrapassou o limite permitido.

			O conteúdo dos componentes que não ultrapassaram o seu limite é zero.
		42	Na correção de um vetor de direção, a diferença de ângulo com relação à direção atual é maior do que o valor máximo especificado através do dado de ajuste SD41611 $SN_CORR_TRAFO_DIR_MAX.
		43	A busca de uma variável de sistema para ser descrita foi desprezada devido à falta de direitos de gravação.

<Corr_Vect>:	Vetor de correção O conteúdo do vetor de correção foi definido por meio dos seguintes parâmetros **<**Corr_Index**>** e <Corr_Mode>. Se <Corr_Condição> = 41, o conteúdo do vetor de correção será sobrescrito (s. o.).	
	Tipo de dado:	REAL
<Corr_Index**>**:	Seção onde esse elemento de correção deve ser modificado / Index do eixo de orientação cujo vetor de direção deve ser modificado	
	Tipo de dado:	INT
	O parâmetro <Corr_Index> tem codificação decimal (1º até o 10º dígito):	
	1º dígito:	Contém o Index da seção ou do eixo de orientação na sequência da peça.
	10º dígito:	Refere-se à sequência da peça.
		0x Sequência da peça
		1x Sequência da ferramenta

<Corr_Mode>:	Modo de correção		
	Tipo de dado:	INT	
	O parâmetro <Corr_Mode> tem codificação decimal (1º até o 1000º dígito):		
	1º dígito:	Determina qual elemento precisa ser corrigido.	
		xxx**0**	Correção de um vetor de offset linear
		xxx**1**	Correção do vetor de direção de um eixo de orientação
	10º dígito:	Determina como o elemento de correção, que se refere ao conteúdo de <Corr_Index> , deve ser modificado.	
		xx**0**x	O vetor de correção é gravado imediatamente no elemento de correção. Então, essa variante poderá ser usada para gravar imediatamente o elemento de correção, sem que o Index <n> do dado do sistema ($NK_OFF_DIR[<n>, ...]) em questão seja conhecido.
		xx**1**x	Como em 0, contudo, com a diferença de que o valor da correção informado é interpretado em coordenadas globais. Uma diferença entre as variantes 0 e 1 pode, no entanto, sempre ocorrer se a sequência cinemática na posição normal (posições de todos os eixos de orientação igual a 0) contiver mais rotações.
		xx**2**x	Como em 1, contudo, com a diferença de que o valor da correção se refere a toda a seção, ou seja, esse valor é introduzido no elemento de correção de modo que toda a seção atinja um comprimento definido através do valor de correção.
		Indicação: os valores 1 e 2 não são permitidos na gravação do vetor de direção de um eixo de orientação.	
	100º dígito:	Determina como o conteúdo do parâmetro <Corr_Vect> deve ser interpretado.	
		x**0**xx	O vetor de correção <Corr_Vect> informado contém todo o novo comprimento do elemento de correção ou da seção à qual o <Corr_Index> se refere, em conjunto com o 10º dígito de <Corr_Mode> (correção absoluta).
		x**1**xx	O vetor de correção <Corr_Vect> informado contém somente a diferença com relação ao comprimento atual do elemento de correção ou da seção à qual o <Corr_Index> se refere, em conjunto com o 10º dígito de <Corr_Mode> (correção incremental).
		Indicação: na correção do vetor de direção de um eixo de orientação, o conteúdo do 100º dígito deve ser 0.	
	1000º dígito:	Determina se a correção precisa ser limitada pelo seguinte valor máximo: • SD41610 $SN_CORR_TRAFO_LIN_MAX ou • SD41611 $SN_CORR_TRAFO_DIR_MAX	
		0xxx	A supervisão da correção máxima está ativa.

		1xxx	A supervisão da correção máxima não está ativa.
<No_Alarm>:	Comportamento em caso de erro (valor retornado > 0) (**opcional**)		
	Tipo de dado:	BOOL	
	Valor:	FALSE (Default)	Em caso de erro, a execução do programa será interrompida e o alarme 14103 será mostrado.
		TRUE	Em caso de erro a execução do programa será interrompida e nenhum alarme será mostrado. Caso de aplicação: Reação específica do usuário corresponde ao valor de devolução

Indicação

Se ocorrer um erro na chamada da função, ou é emitido um alarme ou é retornado um número de falha (ver parâmetro <No_Alarm>), de modo que o próprio usuário possa reagir adequadamente à condição de erro. A causa da falha é mais bem determinada por um parâmetro de alarme. Um número de falha retornado em vez de um alarme é idêntico ao parâmetro do alarme.

Mais informações sobre CORRTRAFO

A estrutura cinemática de uma máquina com transformação de orientação é gravada por uma ou duas sequências cinemáticas (sequência da peça) que partem do ponto zero do sistema de coordenadas global. Uma das duas sequências, a **sequência da ferramenta**, termina no ponto de referência da ferramenta, a outra, a **sequência da peça**, termina no ponto zero do sistema de coordenadas básico.

A função CORRTRAFO omite comprimentos de braço de alavanca e direções de eixo em máquinas com transformação de orientação e grava esses valores em elementos de correção especiais. Uma sequência cinemática é gravada, entre outros, com elementos do tipo OFFSET definidos através de $NK_TYPE.

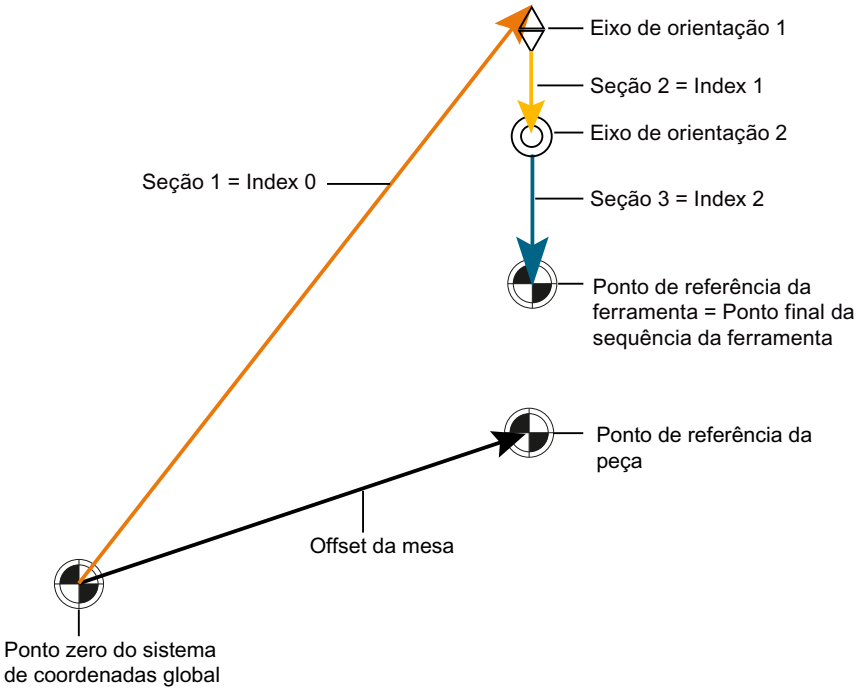
CORRTRAFO trabalha com seções

Ambas as sequências da peça podem se dividir em no máximo quatro seções de cada vez:

- A seção 1 começa no ponto inicial da sequência e termina no primeiro eixo de orientação.
- A seção 2 é a seção entre o eixo de orientação 1 e o eixo de orientação 2.
- A seção 3 é a seção entre o eixo de orientação 2 e o eixo de orientação 3.
- A seção 4 é a seção entre eixo de orientação 3 e o final da sequência da ferramenta ou da sequência da peça.

As seções podem conter, conforme o caso, elementos de sequência constantes do tipo OFFSET ou ROT_CONST.

O seguinte gráfico mostra uma transformação de orientação com 2 eixos de orientação.



Esquema 10-1 CORRTRAFO exemplo

As seções são definidas de modo inequívoco: Movendo-se a sequência cinemática da peça, do seu ponto inicial até o ponto final, a primeira seção terá o Index 0, a próxima o Index 1 e assim por diante. O Index da última seção é então sempre igual à quantidade de eixos de orientação.

Elementos de correção

Um elemento constante da sequência cinemática (elemento da sequência do tipo $NK_TYPE[<n>] = "OFFSET")pode ser referenciado, em cada uma dessas seções, com as variáveis de sistema $NT_CORR_ELEM_T[<n>, 0 ... 3] ou $NT_CORR_ELEM_P[<n>, 0 ... 3]. Os valores de correção que foram determinados na medição da máquina, são gravados nesse elemento assim identificado, com auxílio da função CORRTRAFO.

Exemplo com Index de transformação = 1:

- $NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"; o offset do eixo C (eixo de orientação 1) na seção 1 é definido como elemento de correção.
- $NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"; o offset do eixo B (eixo de orientação 2) na seção 2 é definido como elemento de correção.
- $NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"; o offset do eixo B até o ponto de referência da ferramenta, na seção 3, é definido como elemento de correção.

A sequência das referências em $NT_CORR_ELEM_T/P[<n>, 0 ... 3] precisa corresponder às seções acima descritas, ou seja, em $NT_CORR_ELEM_T/P [<n>, 0] só pode haver um elemento da sequência que se encontre antes do primeiro eixo de orientação, e assim por diante.

A função CORRTRAFO grava no elemento de correção assim definido, os valores determinados através da medição da máquina. A modificação dos valores de correção é definida em CORRTRAFO através do parâmetro <Corr_Mode>.

Fechar a cadeia

Se o bit 7 ou o bit 8 estiver definido nas variáveis de sistema $NT_CNTRL[<n>], serão introduzidas automaticamente constantes adicionais no final da sequência da peça (bit 7) ou antes do ponto inicial da sequência da ferramenta (bit 8), que estabelecem uma ligação do ponto final da sequência com o ponto zero da máquina ("Fechamento da sequência").

Esses elementos introduzidos automaticamente não podem ser gravados externamente, mas apenas lidos (ver sobre isso as variáveis de sistema $AC_TRAFO_CORR_ELEM_P/T).

Ponto para fechamento da sequência da ferramenta

Se as variáveis de sistema $NT_CLOSE_CHAIN_T não estiverem vazias, a sequência da ferramenta não é fechada no ponto final da sequência, mas sim no ponto final dos elementos da sequência marcados. Outros elementos da sequência localizados atrás desse ponto, causam então um deslocamento correspondente do ponto zero na ativação da transformação.

Index de um eixo de orientação

Além dos offsets constantes entre os eixos de orientação, os vetores de direção dos eixos de orientação também podem ser gravados com a função CORRTRAFO. Com relação ao Index de um eixo de orientação, o Index é aqui entendido como o que resulta quando a sequência cinemática da peça se desloca do seu início até o final, enquanto a contagem começa no zero. O Index de um eixo de orientação é, portanto, sempre igual ao Index da seção anterior a ele.

O Index de um eixo de orientação também pode ser determinado com as variáveis de sistema $AC_TRAFO_ORIAX_LOC.

Alteração máxima permitida de um elemento da sequência

A alteração máxima permitida de um elemento da sequência pode ser limitada através de ambos os dados de ajuste, SD41610 $SN_CORR_TRAFO_LIN_MAX para vetores offset e SD41611 $SN_CORR_TRAFO_DIR_MAX para vetores de direção de eixos de orientação. SD41610 $SN_CORR_TRAFO_LIN_MAX fornece até qual valor máximo cada componente individual do vetor pode ser alterado com relação ao seu valor limite. SD41611 $SN_CORR_TRAFO_DIR_MAX fornece até qual ângulo máximo a direção do vetor do eixo pode ser alterada com relação ao seu valor limite. O valor limite é sempre o valor correspondente que está atuante na transformação ativa, na chamada de CORRTRAFO. Ou seja, o conteúdo dos dados cinemáticos no gerenciamento de dados, eventualmente alterado após a ativação da transformação, não tem nenhuma influência na efetividade da função CORRTRAFO.

Corretores de ferramentas 11

11.1 Memória de correções

Estrutura da memória de correções

Cada campo de dados pode ser chamado com um número T ou D (exceto "Nº D plano") e, além das informações geométricas da ferramenta, também contém outras especificações como por exemplo o tipo de ferramenta.

Estrutura plana de números D

A "estrutura plana de números D" é utilizada se o gerenciamento de ferramentas for realizado fora da NC. Neste caso os números D são criados com os respectivos blocos de correção de ferramenta sem a atribuição às ferramentas.

No programa de peça ainda se pode programar o T. Mas este T não tem relação com o número D programado.

Dados de corte do usuário

Através de dados de máquina podem ser configurados dados de corte do usuário. Observe as instruções do fabricante da máquina.

Parâmetro da ferramenta

Indicação

Valores individuais na memória de correções

Os diversos valores da memória de correções P1 até P25 podem ser lidos e gravados no programa através de variáveis de sistema. Os demais parâmetros estão reservados.

Os parâmetros de ferramenta $TC_DP6 até $TC_DP8, $TC_DP10 e $TC_DP11, $TC_DP15 até $TC_DP17, $TC_DP19 e $TC_DP20 têm outro significado, independentemente do tipo de ferramenta.

Parâmetro de ferramenta Número (DP)	Significado das variáveis de sistema	Observação
$TC_DP1	Tipo de ferramenta	Como vista geral veja a lista
$TC_DP2	Posição de corte	apenas para ferramentas de tornear
Geometria	**Correção de comprimentos**	
$TC_DP3	Comprimento 1	Processamento por
$TC_DP4	Comprimento 2	Tipo e plano
$TC_DP5	Comprimento 3	
Geometria	**Raio**	

Parâmetro de ferramenta Número (DP)	Significado das variáveis de sistema	Observação
$TC_DP6 [1] $TC_DP6 [2]	Raio 1 / comprimento 1 diâmetro d	Ferram. de fresar/tornear/retificar Serra para ranhuras
$TC_DP7 [1] $TC_DP7 [2]	Comprimento 2 / raio de canto de fresa cônica Largura de ranhura b raio de canto	Fresas Serra para ranhuras
$TC_DP8 [1] $TC_DP8 [2]	Raio de arredondamento 1 para ferramentas de fresar Ressalto k	Fresas Serra para ranhuras
$TC_DP9 [1] [3]	Raio de arredondamento 2	reservado
$TC_DP10 [1]	Ângulo 1 lado frontal da ferramenta	fresas cônicas
$TC_DP11 [1]	Ângulo 2 eixo longitudinal da ferramenta	fresas cônicas
Desgaste	**Correção de comprimento e raio**	
$TC_DP12	Comprimento 1	
$TC_DP13	Comprimento 2	
$TC_DP14	Comprimento 3	
$TC_DP15 [1] $TC_DP15 [2]	Raio 1 / comprimento 1 diâmetro d	Ferramenta de fresar/tornear/retificar Serra para ranhuras
$TC_DP16 [1] $TC_DP16 [3]	Comprimento 2 / Raio de canto de fresa cônica Largura de ranhura b raio de canto	Fresas Serra para ranhuras
$TC_DP17 [1] $TC_DP17 [2]	Raio de arredondamento 1 para fresas Ressalto k	Fresas / fresamento de topo 3D Serra para ranhuras
$TC_DP18 [1] [3]	Raio de arredondamento 2	reservado
$TC_DP19[1]	Ângulo 1 lado frontal da ferramenta	fresas cônicas
$TC_DP20[1]	Ângulo 2 eixo longitudinal da ferramenta	fresas cônicas
Medida básica/Adaptador	**Correções dos comprimentos**	
$TC_DP21	Comprimento 1	
$TC_DP22	Comprimento 2	
$TC_DP23	Comprimento 3	
Tecnologia		
$TC_DP24	Ângulo livre	apenas para ferramentas de tornear
$TC_DP25		reservado

[1] serve também para equipamentos de fresamento para fresamento de topo 3D

[2] para equipamentos do tipo Serra de entalhe

[3] reservado (não é usado pelo SINUMERIK 840D sl)

Observações

Para as grandezas geométricas (p. ex. comprimento 1 ou raio) existem vários componentes de entrada. Estes são processados aditivamente para uma grandeza resultante (p. ex. comprimento total 1, raio total), que entrará em ação.

As correções desnecessárias devem ser especificadas com o valor zero.

Parâmetros de ferramenta $TC-DP1 até $TC-DP23 com ferramentas de contorno

Indicação

Os parâmetros do equipamento, que não estão registrados na tabela, como por exemplo: B. $TC_DP7, não serão avaliados, d. h. seu conteúdo é irrelevante.

Parâmetro de ferramenta Número (DP)	Significado	Cortes Dn		Observação
$TC_DP1	Tipo de ferramenta			400 até 599
$TC_DP2	Posição de corte			
Geometria	**Correção de comprimentos**			
$TC_DP3	Comprimento 1			
$TC_DP4	Comprimento 2			
$TC_DP5	Comprimento 3			
Geometria	**Raio**			
$TC_DP6	Raio			
Geometria	**Ângulo limite**			
$TC_DP10	Ângulo limite mínimo			
$TC_DP11	Ângulo limite máximo			
Desgaste	**Correção de comprimento e raio**			
$TC_DP12	Desgaste de comprimento 1			
$TC_DP13	Desgaste de comprimento 2			
$TC_DP14	Desgaste de comprimento 3			
$TC_DP15	Desgaste do raio			
Desgaste	**Ângulo limite**			
$TC_DP19	Ângulo limite mín. de desgaste			
$TC_DP20	Ângulo limite máx. de desgaste			
Medida básica/Adaptador	**Correções dos comprimentos**			
$TC_DP21	Comprimento 1			
$TC_DP22	Comprimento 2			
$TC_DP23	Comprimento 3			

Valor básico e valor de desgaste

Os tamanhos resultantes sempre resultam da soma do valor básico com o valor de desgaste (p. ex. $TC_DP6 + $TC_DP15 para o raio). Para o comprimento da ferramenta ainda é adicionada a dimensão básica ($TC_DP21 – $TC_DP23) do primeiro corte. Neste comprimento de ferramenta também terão efeito todas as demais grandezas que podem influenciar o comprimento efetivo de uma ferramenta convencional (como adaptadores, porta-ferramenta orientável, dados de ajuste).

Ângulo limite 1 e 2

Os ângulos limite 1 e 2 sempre têm sua referência no vetor do centro do corte ao ponto de referência do corte e são contados em sentido anti-horário.

11.2 Corretores aditivos

11.2.1 Ativar correções aditivas (DL)

As correções aditivas podem ser consideradas como correções programáveis do processo de usinagem. Elas têm referência nos dados geométricos de um corte com isso são parte integrante dos dados de corte da ferramenta.

Os dados de uma correção aditiva são ativados por meio de um número DL (DL: (Location dependent; correções referentes ao respectivo local de aplicação) e especificados por meio da interface de operação.

Aplicação

Através das correções aditivas podem ser compensados erros de medição condicionados ao local de aplicação.

Sintaxe

```
DL=<número>
```

Significado

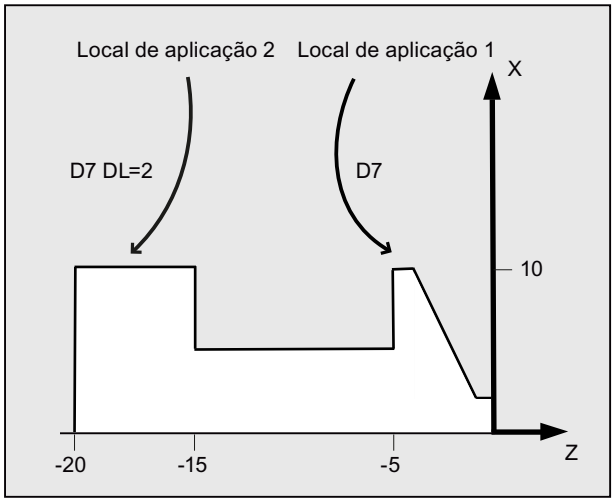
`DL`:	Comando para ativação de uma correção aditiva
`<Número>`:	Através do parâmetro `<número>` é especificado o bloco de dados de correção da ferramenta a ser ativado

Indicação

A definição da quantidade e ativação das correções aditivas são realizadas através de dados de máquina (→ Observar as informações do fabricante da máquina!).

Exemplo

O mesmo corte é utilizado para 2 assentos de rolamento:



Código de programa	Comentário
`N110 T7 D7`	`O revólver está posicionado no alojamento 7. D7 e DL=1 são ativados e executados no próximo bloco.`
`N120 G0 X10 Z1`	
`N130 G1 Z-6`	
`N140 G0 DL=2 Z-14`	`De forma aditiva ao D7 se ativa o DL=2 e executado no próximo bloco.`
`N150 G1 Z-21`	
`N160 G0 X200 Z200`	`Aproximar o ponto de troca de ferramentas.`
`...`	

11.2.2 Definir valores de desgaste e de ajuste ($TC_SCPxy[t,d], $TC_ECPxy[t,d])

Os valores de desgaste e de ajuste podem ser lidos e gravados através de variáveis de sistema. Aqui a lógica se orienta na lógica das respectivas variáveis de sistema para ferramentas e cortes.

Variáveis de sistema

`$TC_SCPxy[<t>,<d>]:`	Valores de desgaste que estão atribuídos aos respectivos parâmetros geométricos através de xy, onde x corresponde ao número do valor de desgaste e y estabelece a relação com o parâmetro geométrico.
`$TC_ECPxy[<t>,<d>]:`	Valores de ajuste que estão atribuídos aos respectivos parâmetros geométricos através de xy, onde x corresponde ao número do valor de ajuste e y estabelece a relação com o parâmetro geométrico.
`<t>`: Número T da ferramenta `<d>`: Número D do corte da ferramenta	

Indicação

Os valores definidos de desgaste e de ajuste são adicionados aos parâmetros geométricos e aos demais parâmetros de correção (número D).

Exemplo

O valor de desgaste do comprimento 1 é definido para o corte <d> da ferramenta <t> no valor 1.0.

Parâmetro: $TC_DP**3** (Comprimento 1, para ferramentas de tornear)

Valores de desgaste: $TC_SCP1**3** até $TC_SCP6**3**

Valores de ajuste: $TC_ECP1**3** até $TC_ECP6**3**

$TC_SCP4**3** [<t>,<d>] = 1.0

11.2.3 Deletar correções aditivas (DELDL)

Com o comando `DELDL` são deletadas correções aditivas para o corte de uma ferramenta (liberação de memória). Aqui são deletados tanto os valores de desgaste definidos como os valores de ajuste definidos.

Sintaxe

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<estado>=DELDL[<t>,<d>]
```

Significado

`DELDL`:	Comando para deletar correções aditivas	
`<t>`:	Número T da ferramenta	
`<d>`:	Número D do corte da ferramenta	
`DELDL[<t>,<d>]`:	São deletadas todas as correções aditivas do corte `<d>` da ferramenta `<t>`.	
`DELDL[<t>]`:	São deletadas todas as correções aditivas de todos os cortes da ferramenta `<t>`.	
`DELDL`:	São deletadas todas as correções aditivas de todos cortes de todas ferramentas da unidade TO (para o canal em que o comando for programado).	
`<Status>`:	Estado de deletação	
	Valor:	Significado:
	0	A deletação foi realizada com sucesso.
	-	A deletação não foi executada (quando a parametrização denomina exatamente um corte), ou a deletação não foi executada totalmente (quando a parametrização denomina vários cortes).

Indicação

Os valores de desgaste e de ajuste de ferramentas ativas não podem ser deletados (comportamento semelhante à deletação do `D` e de dados de ferramenta).

11.3 Tratamentos especiais de corretor de ferramenta

Com os dados de ajuste SD42900 até SD42960 se pode controlar a avaliação do sinal prévio para comprimento de ferramenta e desgaste.

Isto também se aplica ao comportamento dos componentes de desgaste no espelhamento de eixos geométricos ou na mudança de plano de usinagem e ainda para compensação de temperatura no sentido da ferramenta.

Valores de desgaste

Se a seguir forem tomados como referência valores de desgaste, entendemos sempre como a soma dos próprios valores de desgaste ($TC_DP12 bis $TC_DP20) e das correções aditivas com os valores de desgaste ($SCPX3 até $SCPX11) e os valores de ajuste ($ECPX3 até $ECPX11).

Para mais informações sobre as correções aditivas, veja:
Literatura:
Manual de funcionamento para gerenciamento de ferramentas

Dados de ajuste

SD42900 $SC_MIRROR_TOOL_LENGTH	Espelhamento de componentes do comprimento de ferramentas e de componentes da dimensão básica.
SD42910 $SC_MIRROR_TOOL_WEAR	Espelhamento de valores de desgaste dos componentes do comprimento de ferramentas.
SD42920 $SC_WEAR_SIGN_CUTPOS	Avaliação do sinal precedente de componentes de desgaste em função da posição do corte.
SD42930 $SC_WEAR_SIGN	Inverte o sinal das dimensões de desgaste.
SD42935 $SC_WEAR_TRANSFORM	Transformação dos valores de desgaste.
SD42940 $SC_TOOL_LENGTH_CONST	Associação dos componentes de comprimento de ferramenta com os eixos geométricos.
SD42950 $SC_TOOL_LENGTH_TYPE	Associação dos componentes de comprimento de ferramenta independentemente do tipo de ferramenta.
SD42960 $SC_TOOL_TEMP_COMP	Valor de compensação de temperatura no sentido da ferramenta. Também está ativo com a orientação de ferramenta presente.

Literatura

Manual de funcionamento das funções básicas; Correção de ferramenta (W1)

Outras informações

Ativação dos dados de ajuste alterados

A nova avaliação de componentes de ferramentas em uma alteração dos dados de ajuste descritos somente se ativa na próxima vez que o corte de ferramenta for selecionado. Se uma

ferramenta está ativa e a avaliação dos dados alterados desta ferramenta deve ser ativada, esta ferramenta deve ser selecionada novamente.

O mesmo se aplica no caso em que o comprimento resultante da ferramenta sofrer alteração, pois o estado do espelhamento de um eixo foi alterado. A ferramenta deve ser selecionada novamente depois do comando de espelhamento, para que os componentes alterados do comprimento da ferramenta possam ser ativados.

Porta-ferramenta orientável e novos dados de ajuste

Os dados de ajuste SD42900 até SD42940 não atuam sobre os componentes de um porta--ferramenta orientável que eventualmente esteja ativo. Entretanto, uma ferramenta com seu comprimento total resultante (comprimento da ferramenta + desgaste + dimensão básica) sempre entra no processamento com um porta-ferramenta orientável. No processamento do comprimento total resultante são consideradas todas alterações geradas através dos dados de ajuste, ou seja, os vetores do porta-ferramenta orientável independem do plano de usinagem.

Indicação

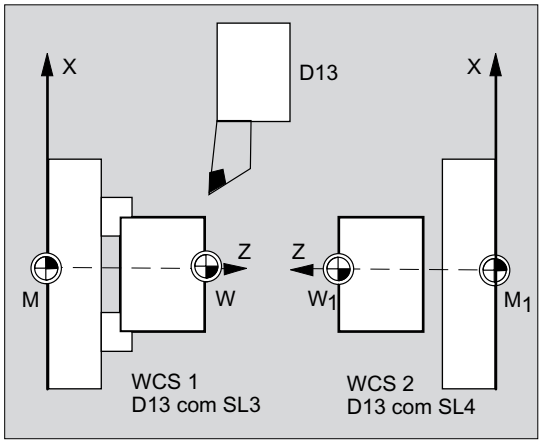
Frequentemente, ao serem empregados porta-ferramentas orientáveis, será mais conveniente definir todas ferramentas para um sistema básico não espelhado, mesmo para aquelas que serão utilizadas em operações espelhadas. Na usinagem com eixos espelhados o porta-ferramenta será girado de modo que a posição real da ferramenta seja descrita de forma correta. Todos componentes de comprimento da ferramenta atuam automaticamente no sentido correto, de modo que para o comando apenas resta a avaliação de alguns componentes através de dados de ajuste em função do estado de espelhamento de determinados eixos.

Outras possibilidades de aplicação

O uso da funcionalidade de porta-ferramentas orientáveis também pode ser bastante útil, se na máquina não houver nenhuma possibilidade física de girar as ferramentas, mas as ferramentas serem instaladas definitivamente com diferentes orientações. O dimensionamento da ferramenta pode ser realizado em uma orientação, e as dimensões relevantes à usinagem resultam através de rotações do porta-ferramenta virtual.

11.3.1 Espelhar comprimentos de ferramentas

Com os dados de ajuste SD42900 $SC_MIRROR_TOOL_LENGTH e SD42910 $SC_MIRROR_TOOL_WEAR definidos diferente de zero, podemos espelhar componentes de comprimento de ferramentas e componentes da dimensão básica com valores de desgaste em seus respectivos eixos.



SD42900 $SC_MIRROR_TOOL_LENGTH

Dado de ajuste **diferente** de zero:

Os componentes de comprimento de ferramenta ($TC_DP3, $TC_DP4 e $TC_DP5) e os componentes da dimensão básica ($TC_DP21, $TC_DP22 e $TC_DP23), cujos eixos estão espelhados, também são espelhados através da inversão de sinais.

Os valores de desgaste **não** são espelhados juntos. Se estes também devem ser espelhados, então se deve empregar o dado de ajuste SD42910 $SC_MIRROR_TOOL_WEAR.

SD42910 $SC_MIRROR_TOOL_WEAR

Dado de ajuste **diferente** de zero:

Os valores de desgaste dos componentes de comprimento da ferramenta, cujos eixos estão espelhados, também são espelhados através da inversão de sinais.

11.3.2 Avaliação de sinais de desgaste

Com os dados de ajuste SD42920 $SC_WEAR_SIGN_CUTPOS e SD42930 $SC_WEAR_SIGN definidos diferente de zero, podemos inverter a avaliação de sinais dos componentes de desgaste.

SD42920 $SC_WEAR_SIGN_CUTPOS

Dado de ajuste **diferente** de zero:

Nas ferramentas com posição de corte relevante (ferramentas de tornear e de retificar, tipos de ferramenta 400) a avaliação de sinais dos componentes de desgaste no plano de usinagem depende da posição do corte. Nos tipos de ferramenta sem posição de corte relevante, este dado de ajuste não tem nenhum significado.

Na seguinte tabela estão marcadas com um X as dimensões cujo sinal é invertido através do SD42920 (diferente de 0):

Posição de corte	Comprimento 1	Comprimento 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Indicação

A avaliação de sinais através do SD42920 e do SD42910 é independente uma da outra. Por exemplo, se o sinal de uma dimensão se alterar através dos dois dados de ajuste, então o sinal resultante permanece inalterado.

SD42930 $SC_WEAR_SIGN

Dado de ajuste **diferente** de zero:

O sinal de todas dimensões de desgaste é invertido. O efeito ocorre tanto no comprimento da ferramenta como nas demais grandezas como raio de ferramenta, raio de arredondamento, etc.

Se for especificada uma dimensão de desgaste, então a ferramenta será "mais curta" e "mais estreita", veja o capítulo "Correção de ferramenta, tratamento especial", "Ativação dos dados de ajuste alterados".

11.3.3 Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

Em função da cinemática da máquina ou da presença de um porta-ferramenta orientável, os valores de desgaste medidos em um destes sistemas de coordenadas serão transportados e transformados para um sistema de coordenadas adequado.

Sistemas de coordenadas da usinagem ativa

A partir dos seguintes sistemas de coordenadas podem resultar Offsets do comprimento da ferramenta, com os quais se inclui o componente desgaste de comprimento de ferramenta através do respectivo comando G do grupo 56 em uma ferramenta ativa:

- Sistema de coordenadas da máquina (MCS)
- Sistema de coordenadas básico (BCS)
- Sistema de coordenadas da peça (WCS)

- Sistema de coordenadas da ferramenta (TCS)
- Sistema de coordenadas da ferramenta da transformação cinemática (KCS)

Sintaxe

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

Significado

TOWSTD:	Valor de ajuste básico para correções no valor de desgaste do comprimento da ferramenta
TOWMCS:	Correções no comprimento da ferramenta em MCS
TOWWCS:	Correções no comprimento da ferramenta em WCS
TOWBCS:	Correções no comprimento da ferramenta em BCS
TOWTCS:	Correções do comprimento da ferramenta no ponto de referência do porta-ferramenta (porta-ferramenta orientável)
TOWKCS:	Correções do comprimento da ferramenta do cabeçote de ferramenta (transformação cinemática)

Outras informações

Características de diferenciação

Na seguinte tabela estão representadas as características mais importantes de diferenciação:

Comando G	Valor de desgaste	Porta-ferramenta orientável ativo
TOWSTD:	Valor de ajuste básico, comprimento da ferramenta	Os valores de desgaste estão sujeitos à rotação.
TOWMCS:	Valor de desgaste em MCS. O TOWMCS é idêntico ao TOWSTD, quando nenhum porta-ferramenta orientável estiver ativo.	Apenas se gira o vetor do comprimento de ferramenta resultante, sem consideração do desgaste.
TOWWCS:	O valor de desgaste é convertido do WCS para MCS.	O vetor da ferramenta é processado sem consideração do desgaste, como no TOWMCS.
TOWBCS:	O valor de desgaste é convertido do BCS para MCS.	O vetor da ferramenta é processado sem consideração do desgaste, como no TOWMCS.
TOWTCS:	O valor de desgaste é convertido do sistema de coordenadas da ferramenta para o MCS.	O vetor da ferramenta é processado sem consideração do desgaste, como no TOWMCS.

TOWWCS, TOWBCS, TOWTCS: O vetor de desgaste é adicionado ao vetor da ferramenta.

Transformação linear

A definição do comprimento da ferramenta somente é conveniente em MCS se o MCS resultar do BCS através de uma transformação linear.

Transformação não linear

Se, por exemplo, uma transformação não linear estiver ativa com TRANSMIT, então se utiliza automaticamente o BCS quando se especifica o MCS como sistema de coordenadas desejado.

Nenhuma transformação cinemática e nenhum porta-ferramenta orientável

Se nem uma transformação cinemática nem um porta-ferramenta orientável estiver ativo, então, com exceção do WCS, os demais quatro sistemas de coordenadas coincidirão. Com isso apenas o WCS se diferencia dos demais. Visto que são avaliados somente os comprimentos de ferramentas, as transformações entre os sistemas de coordenadas não têm nenhuma importância.

Literatura:
Para mais informações sobre correções de ferramentas, veja:
Manual de funcionamento das funções básicas; Correções de ferramenta (W1)

Inclusão dos valores de desgaste

O dado de ajuste **SD42935 $SC_WEAR_TRANSFORM** define quais dos três componentes de desgaste:

- Desgaste
- Correções aditivas finas
- Correções aditivas aproximadas

Serão submetidas a uma rotação através de uma transformação de adaptador ou de um porta-ferramenta orientável, se um dos seguintes comandos G estiverem ativos:

- `TOWSTD`
  Posição inicial. para correções no comprimento da ferramenta.
- `TOWMCS`
  Valores de desgaste no sistema de coordenadas da máquina (MCS).
- `TOWWCS`
  Valores de desgaste no sistema de coordenadas da peça de trabalho (WCS).
- `TOWBCS`
  Valores de desgaste no sistema de coordenadas básico (BCS).

- TOWTCS
  Valores de desgaste no sistema de coordenadas da ferramenta no assento do porta-ferramenta (referência do porta-ferramenta T).
- TOWKCS
  Valores de desgaste no sistema de coordenadas do cabeçote da ferramenta para transformação cinemática.

Indicação

A avaliação dos diversos componentes de desgaste (atribuição aos eixos geométricos, avaliação de sinais) é influenciada através dos seguintes fatores:

- Plano ativo
- Transformação de adaptador
- Dados de ajuste:
  - 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 Comprimento de ferramenta e mudança de planos

Com o dado de ajuste SD42940 $SC_TOOL_LENGTH_CONST definida diferente de zero, podemos atribuir os componentes de comprimento de ferramenta como comprimento, desgaste e dimensão básica aos eixos geométricos para ferramentas de tornear e retificar em uma mudança de planos.

SD42940 $SC_TOOL_LENGTH_CONST

Dado de ajuste **diferente** de zero:

A atribuição dos componentes de comprimento de ferramenta (comprimento, desgaste e dimensão básica) com os eixos geométricos não se altera na mudança do plano de usinagem (G17 - G19).

A seguinte tabela mostra a associação dos componentes de comprimento de ferramenta com os eixos geométricos para ferramentas de tornear e de retificar (tipo de ferramenta 400 até 599):

Conteúdo	Comprimento 1	Comprimento 2	Comprimento 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Cada valor diferente de 0, que não é igual a um dos seis valores mencionados, será avaliado como o valor 18.

A seguinte tabela mostra a associação dos componentes de comprimento de ferramenta com os eixos geométricos para todas demais ferramentas (tipo de ferramenta < 400 e > 599):

Plano de usinagem	Comprimento 1	Comprimento 2	Comprimento 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Cada valor diferente de 0, que não é igual a um dos seis valores mencionados, será avaliado como o valor 17.

Indicação

Na representação nas tabelas se pressupõe que os eixos geométricos até 3 sejam identificados com X, Y e Z. Determinante para a atribuição de uma correção para um eixo não é o identificador de eixo, mas a sequência dos eixos.

11.4 Compensação de ferramenta Online

11.4.1 Definir função polinomial (FCTDEF)

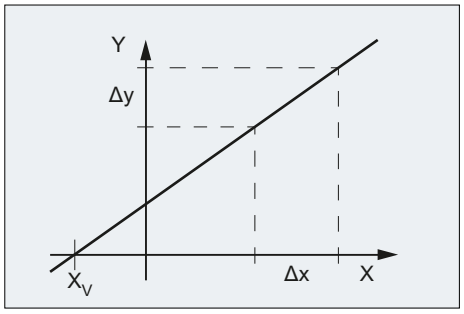
Certas estratégias do dressador (p. ex., do rolo dressador) distingue-se pelo fato de que o rebolo diminui o raio continuamente (linearmente) com a entrega do rolo dressador. Para isso, é preciso uma função linear entre a entrega do rolo dressador e gravação do valor do desgaste do comprimento. A definição da função linear ocorre através do procedimento pré--definidoFCTDEF(...) para funções polinomiaisaté um máximo de terceiro grau.

Equação linear

$y = f(x) = a_0 + a_1 * x_1$

a_1: Declive das retas, com $a_1 = \Delta x / \Delta y$

a_0: Deslocamento das retas ao longo do eixo X, com $a_0 = -a1 * X_V$



Sintaxe

```
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
```

Significado

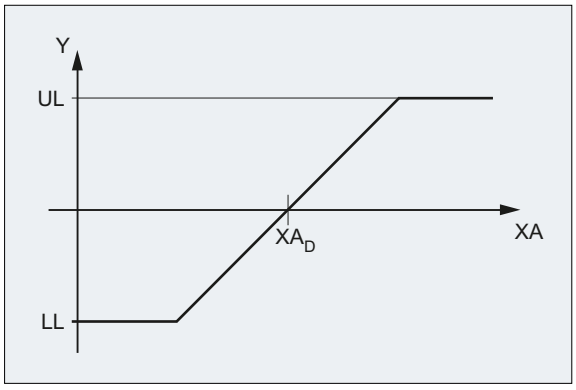
`FCTDEF(...)`:	Definição de uma função polinomial para PUTFTOCF(...): $y = f(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3$	
<`Func`**>**:	Número de função	
	Tipo de dado:	INT
	Faixa de valores:	1, 2, 3
`<LLimit>`:	Valor limite inferior	
	Tipo de dado:	REAL
`<ULimit>`:	Valor limite superior	
	Tipo de dado:	REAL
`<a0>,<a1>,<a2>,<a3>`:	Coeficientes da função de polinômio	
	Tipo de dado:	REAL

Exemplo

Definições

- Número de função: 1
- Valor-limite superior e inferior: -100, 100
- Elevação da curva característica: $a_1 = 1$
- O ponto operacional deve situar-se no meio da curva característica. A curva característica tem que ser movida com base na posição de destino dos eixos XA no WCS no momento da definição de função no programa NC na direção de Y negativo: $a_0 = -a_1 * XA_D = -1 *$ $AA_IW
- $a_2 = a_3 = 0$

Curva característica



UL Valor limite superior

LL Valor limite inferior

XA_D Valor nominal do eixo XA no momento da definição da função no programa NC

Programação

Código de programa	Comentário
`FCTDEF(1,-100,100,-$AA_IW[XA],1)`	`; Definição de função`

11.4.2 Gravar continuamente a correção de ferramenta Online (PUTFTOCF)

Com o procedimento pré-definido PUTFTOCF(...), é realizada uma correção de ferramenta on-line uma função polinomial definida anteriormente com FCTDEF(...) (Página 438).

Indicação

A correção de ferramenta on-line pode ser feita através de uma ação sincronizada.

Para mais informações, veja o guia/manual de função de ações sincronizadas.

Sintaxe

```
PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)
```

Significado

`PUTFTOCF(...)`:	Escrever a correção de ferramenta on-line, continuamente por blocos com base na função polinomial definida com FCTDEF(...)	
`<Func>`:	Número de função, definido na definição de função com FCTDEF(...)	
	Tipo de dado:	INT
	Faixa de valores:	1, 2, 3
`<RefVal>`:	Valor de referência, de qual deve ser derivada a correção (p. ex. valor nominal de um eixo)	
	Tipo de dado:	VAR REAL
`<ToolPar>`:	Número do parâmetro de desgaste (comprimento 1, 2 ou 3), onde o valor de correção deve ser calculado	
	Tipo de dado:	INT
`<Chan>`:	Número do canal, onde a correção de ferramenta Online deve ser ativada **Nota**: necessário apenas se a correção não for ativada no canal ativo.	
	Tipo de dado:	INT
`<Sp>`:	Número do fuso, onde a correção de ferramenta Online deve ser ativada **Nota:** Uma especificação somente é necessária se, ao invés do rebolo ativo, deve ser corrigido um rebolo não ativo no momento da ferramenta empregada.	
	Tipo de dado:	INT

11.4.3 Gravar discretamente a correção de ferramenta on-line (PUTFTOC)

Função

Com o procedimento pré-definidoPUTFTOC(...) uma correção de ferramenta on-line será feita com um valor fixo de correção.

Sintaxe

```
PUTFTOC(<CorrVal>,<ToolPar>,<Chan>,<Sp>)
```

Significado

`PUTFTOC(...)`:	Gravação da correção de ferramenta on-line	
`<CorrVal>`:	Valor de correção que é adicionado aos parâmetros de desgaste	
	Tipo de dado:	REAL

<ToolPar>:	Número do parâmetro de desgaste (comprimento 1, 2 ou 3), onde o valor de correção deve ser calculado	
	Tipo de dado:	INT
<Chan>:	Número do canal, onde a correção de ferramenta Online deve ser ativada **Nota:** necessário apenas se a correção não for ativada no canal ativo.	
	Tipo de dado:	INT
<Sp>:	Número do fuso, onde a correção de ferramenta Online deve ser ativada **Nota:**Uma especificação somente é necessária se, ao invés do rebolo ativo, deve ser corrigido um rebolo não ativo no momento da ferramenta empregada.	
	Tipo de dado:	INT

11.4.4 Ligar/desligar a correção de ferramenta on-line (FTOCON/FTOCOF)

Com os comandos GFTOCON eFTOCOF a correção de ferramenta on-line é ligada ou desligada.

Sintaxe

```
FTOCON
...
FTOCOF
```

Significado

FTOCON:	Ligar a correção de ferramenta on-line O comando deve ser programado no canal em que o a correção da ferramenta on-line deverá ser ligada.
FTOCOF:	Desligar a correção de ferramenta on-line O comando deve ser programado no canal em que o a correção da ferramenta on-line deverá ser desligada. **Nota:** com o FTOCOF a correção de ferramenta não continuará sendo movida. Nos dados de correção de corte específicos permanece o com valor calculado com PUTFTOC/PUTFTOCF. Para a desativação definitiva da correção de ferramenta Online, após o FTOCOF, ainda precisa ocorrer uma ativação/desativação da ferramenta (T...).

11.5 Correção do raio de ferramenta 3D

11.5.1 Selecionar correção do raio da ferramenta em 3D para fresamento periférico em 3D (CUT3DC, CUT3DCD, ISD)

A correção do raio da ferramenta 3D (WRK 3D) para fresamento de periférico 3D, sem consideração das superfícies limite, é selecionada com o modo de comando G ativo, CUT3DC ou CUT3DCD.

A ativação real é realizada com G41 ou G42. Com G40, a correção do raio da ferramenta é desligada.

Sintaxe

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

Significado

CUT3DC:	WRK 3D para fresamento periférico (apenas na transformação de 5 eixos ativa)	
CUT3DCD:	Em uma ferramenta de diferença relativa à WRK 3D para fresamento periférico (apenas na transformação de 5 eixos ativa) A diferença de raio é determinada através do parâmetro WZ $TC_DP15.	
G41/G42 X... Y... Z...:	Ligação da correção do raio da ferramenta	
	G41:	Correção do raio da ferramenta à esquerda do contorno
	G42:	Correção do raio da ferramenta à direita do contorno
	Indicação: a ligação precisa ser realizada em um bloco linear (G0/G1).	
CDOF2:	Ligarsupervisão de colisão para fresamento periférico 3D	
ORIC/ORID:	Através dos comandos G ORIC e ORID, determina-se o comportamento na alteração de orientação em cantos externos.	
	ORIC:	As alterações de orientação em cantos externos são sobrepostas no bloco circular a ser inserido.
	ORID:	As alterações de orientação em cantos externos são realizadas antes do bloco circular a ser inserido.

ISD=<Value>:	Com o endereço ISD (InSertion Depth) é possível, na fresagem periférica e com correção de raio de ferramenta 3D **ativa**, alterar a profundidade de inserção da ferramenta.	
	<Valor>:	Comprimento da profundidade de inserção
G40 X... Y... Z...:	Desativar a correção do raio da ferramenta **Indicação:** o desligamento precisa ser realizado em um bloco linear (G0/G1) com deslocamento do eixo geométrico.	

Indicação

Os comandos G para ativação da WRK 3D são avaliados no bloco de aproximação, ou seja, normalmente no bloco que contém G41 ou G42.

G41 ou G42 também podem ser programados em blocos sem movimento de deslocamento nos eixos geométricos relevantes para a correção. Nesse caso, o bloco de início é o primeiro em um bloco desse tipo após um bloco de deslocamento.

Uma mudança da variante WRK 3D com a correção do raio da ferramenta ativa é ignorada sem alarme.

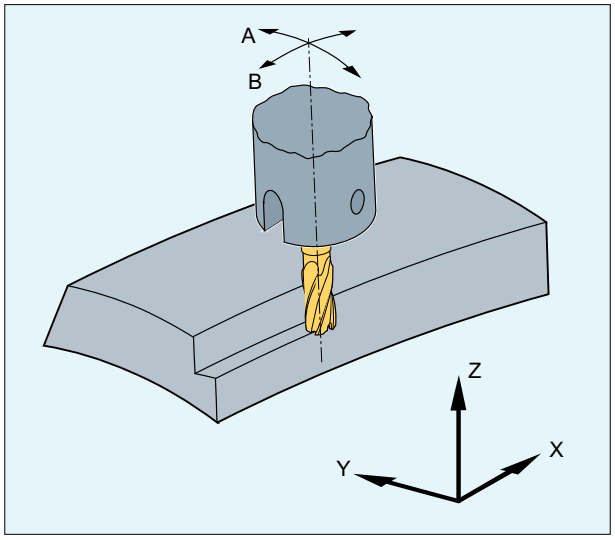
Exemplo

Código de programa	**Comentário**
	; definição da ferramenta D1:
$TC_DP1[1,1]=120	; tipo (fresa com haste)
$TC_DP3[1,1]=20	; vetor de correção de comprimento
$TC_DP6[1,1]=8	; Raio
N10 X0 Y0 Z0 T1 D1 F12000	; seleção da ferramenta.
N20 TRAORI(1)	; ligação da transformação.
N30 G42 ORIC ISD=10 **CUT3DC** G64 X30	; ativação do fresamento periférico 3D,
	; alterações de orientação continuadas nos cantos externos,
	; profundidade de inserção: 10mm
N40 ORIWKS A30 B15	; alterações de orientação em um canto pela especificação das posições do eixo.
N50 Y20 A3=1 C3=1	; bloco com alteração de orientação,
	; especificação da orientação com o vetor de direção.
N60 X50 Y30	; bloco com orientação constante.
N70 Y50 A3=0.5 B3=1 C3=5	; bloco com alteração de orientação.
N80 M63	; bloco sem informações de deslocamento.
N90 X0 ISD=20	; bloco com alteração da profundidade de inserção.
N100 G40 Y0	; desativação da correção do raio da ferramenta.
N110 M30	

Outras informações

Trajetória e orientação

A variante de fresamento periférico utilizada neste caso é realizada com a especificação de um percurso (linha guia) e da orientação correspondente. Neste tipo de usinagem o tipo de ferramenta no percurso é irrelevante. O único fator decisivo é o raio no ponto de ataque da ferramenta.



Esquema 11-1 Fresamento periférico

Comportamento de aproximação

O comportamento de aproximação é sempre NORM com as variantes em 3D da correção do raio da ferramenta.

Comportamento em cantos externos

No fresamento periférico com WRK 3D, os comandos G do grupo 18 são avaliados, em um canto externo, do mesmo modo que as relações em WRK 2½D (comportamento de canto da correção da ferramenta):

- G450: Círculo de transição (a ferramenta percorre os cantos da peça em uma trajetória circular)
  Ao contrário da solução em WRK 2½D, o elemento de contorno introduzido em um canto externo é sempre um círculo com raio 0, no qual a correção do raio da ferramenta age como qualquer outra trajetória programada. Não é possível introduzir um tronco de cone em vez do círculo. Por isso, o endereço DISC não tem nenhum significado nesse caso e não é avaliado.
- G451: Ponto de intersecção das equidistantes (a ferramenta usina para retirada do canto da peça)
  A intersecção é definida quando as curvas de deslocamento de ambos blocos envolvidos são prolongadas e sua intersecção definida no canto perpendicular à orientação da ferramenta no plano.

O procedimento do ponto de corte (G451) não é utilizado quando, entre os blocos do processamento, é introduzido pelo menos um bloco que contém uma alteração da orientação da ferramenta. Nesses casos, é sempre introduzido um círculo no canto.

Comportamento na alteração de orientação em cantos externos

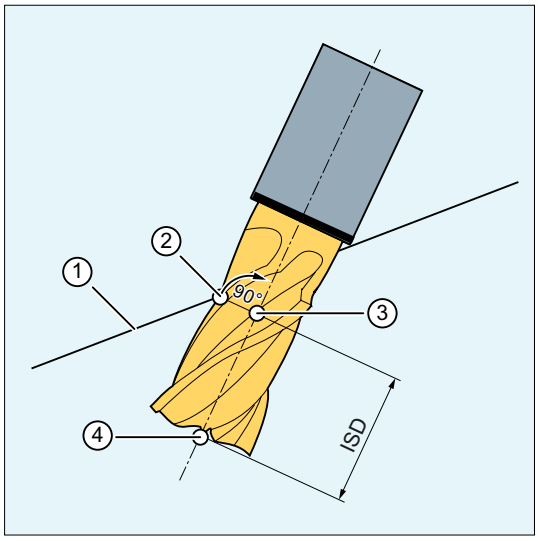
Através dos comandos G ORIC e ORID, pode-se estabelecer se foram realizadas alterações de orientação que foram programadas entre ambos os blocos que constituem o canto, antes do início do bloco do círculo introduzido (ORID) ou simultaneamente a ele (ORIC).

Profundidade de imersão

A profundidade de inserção da fresa é a distância do ponto auxiliar da fresa da ponta da ferramenta.

O ponto auxiliar da fresa é a projeção perpendicular do ponto de usinagem da fresa no trajeto programado sobre o eixo longitudinal da ferramenta.

Deste modo, com a profundidade de inserção, ajusta-se a posição do ponto de usinagem na superfície periférica da ferramenta.



① Trajetória programada
② Ponto de usinagem da fresa
③ Ponto auxiliar da fresa
④ Ponta da fresa
ISD Profundidade de inserção (InSertion Depth)

Esquema 11-2 Profundidade de imersão

Correção do raio da ferramenta relacionada a uma ferramenta de diferença

A WRK 3D relacionada a uma ferramenta de diferença será ativada pelo comando CUT3DCD. Ela deve ser usada, se o contorno programado for relacionado ao percurso do ponto central de uma ferramenta de normalizada e o processamento ocorre com ferramenta diferente. Então, no cálculo da correção do raio de ferramenta 3D, será incluído somente o valor do desgaste do raio da ferramenta ativa ($ TC_ DP15) e os offsets de correção da ferramenta eventualmente programados OFFN e TOFFR. O raio da base($TC_DP6) da ferramenta ativa **não** será calculado.

Fresamento de bolsões com paredes laterais inclinadas para fresamento periférico com CUT3DC

Nesta correção do raio da ferramenta 3D se compensa o desvio do raio da ferramenta ao passo que o avanço é realizado no sentido da normal da superfície com a superfície usinada. Aqui o plano onde está o lado frontal da fresa permanece inalterado se a profundidade de imersão ISD for a mesma. Por exemplo, se comparada à uma ferramenta normalizada, uma fresa de raio menor não alcançaria a base do bolsão, este que também forma a superfície de limitação. Para uma penetração automática da ferramenta, a superfície de limitação deve ser conhecida pelo comando, ver o capítulo "Fresamento periférico 3D levando em conta uma superfície de limitação (CUT3DCC, CUT3DCCD) (Página 452)".

Advanced Surface / Top Surface

Indicação

Na utilização da correção do raio de ferramenta CUT3DCD em combinação com a opção de licença obrigatória "Advanced Surface" ou "Top Surface" devem ser observadas as recomendações de ajuste relacionadas ao "Advanced Surface" / "Top Surface"!

Para a verificação dos dados ajustados, estão disponíveis pelo portal SIOS programas especiais de verificação:

- Programa de verificação para Advanced Surface (https://support.industry.siemens.com/cs/ww/en/view/78956392)
- Programa de verificação para Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)

11.5.2 Selecionar correção do raio da ferramenta 3D para fresamento de topo 3D (CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

A correção do raio da ferramenta 3D (WRK 3D) para fresamento de topo 3D é selecionada com o modo de comando G ativo CUT3DF, CUT3DFS, CUT3DFF ou CUT3DFD.

A ativação real é realizada com G41 ou G42.

Para poder calcular a correção do raio da ferramenta no fresamento de topo 3D, é necessária a definição da normal à superfície do plano a ser usinado. Isso precisa ser realizado no bloco com G41 ou G42 através dos endereços A4, B4, C4 e A5, B5, C5.

Com G40, a correção do raio da ferramenta é desligada.

Sintaxe

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

Significado

`CUT3DFS`:	WRK 3D para fresamento de topo com orientação constante. A orientação da ferramenta é definida pelo G17 - G19 e não influenciada por Frames.	
`CUT3DFF`:	WRK 3D para fresamento de topo com orientação constante. A orientação da ferramenta é definida pelo G17 - G19 e, se necessário, através de um sentido girado por um Frame.	
`CUT3DF`:	WRK 3D para fresamento de topo com alteração de orientação (apenas na transformação de 5 eixos ativa)	
`CUT3DFD`:	Em uma WRK 3D associada a uma ferramenta de diferença, para fresamento de topo com alteração de orientação (apenas na transformação de 5 eixos ativa) A diferença de raio é determinada através do parâmetro WZ $TC_DP15. **Indicação**: CUT3DFD só é possível em combinação com o "alisamento da normal à superfície no fresamento 3D". Este é ativado pelo acesso à função de licença obrigatória "Top Surface" através de CYCLE832(...) (Página 815).	
`G41/G42 X... Y... Z...`:	Ligação da correção de ferramenta O comportamento com G41 e com G42 é idêntico no fresamento de topo 3D. **Indicação:** a ligação precisa ser realizada em um bloco linear (G0/G1).	
`A4/5=... B4/5=... C4/5=...`:	Definição da normal à superfície do plano a ser usinado	
	`A4=... B4=... C4=...`:	Definição no início do bloco
	`A5=... B5=... C5=...`:	Definição no final do bloco
`ORIC/ORID`:	Através dos comandos G `ORIC` e `ORID`, determina-se o comportamento na alteração de orientação em cantos externos.	
	`ORIC`:	As alterações de orientação em cantos externos são sobrepostas no bloco circular a ser inserido.
	`ORID`:	As mudanças de orientação são executadas antes de um bloco circular.
`G40 X... Y... Z...`:	Desativar a correção do raio da ferramenta **Indicação:** o desligamento precisa ser realizado em um bloco linear (G0/G1) com deslocamento do eixo geométrico.	

Indicação

G41 ou G42 também podem ser programados em blocos sem movimento de deslocamento nos eixos geométricos relevantes para a correção. Nesse caso, o bloco de início é o primeiro em um bloco desse tipo após um bloco de deslocamento.

Uma mudança da variante WRK 3D com a correção do raio da ferramenta ativa é ignorada sem alarme.

Exemplos

Exemplo 1: Fresamento de topo 3D com CUT3DF

Código de programa	Comentário
N10	; definição da ferramenta D1:
N20 $TC_DP1[1,1]=121	; tipo da ferramenta (fresa toroidal)
N30 $TC_DP3[1,1]=20	; correção de comprimentos
N40 $TC_DP6[1,1]=5	; Raio
N50 $TC_DP7[1,1]=3	; raio de arredondamento
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; seleção da ferramenta.
N90 TRAORI(1)	; selecionar transformação da orientação.
N100 B4=-1 C4=1	; definição do plano.
N110 **G41** ORID **CUT3DF** G64 X10 Y0 Z0	; ligar correção de ferramenta.
N120 X30	
N130 Y20 A4=1 C4=1	; canto externo, nova definição do plano.
N140 B3=1 C3=5	; mudança de orientação com ORID.
N150 B3=1 C3=1	; mudança de orientação com ORID.
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; mudança de orientação com ORIC
N180 A3=-1 C3=1	; mudança de orientação com ORIC
N190 Y-10 A4=-1 C4=3	; nova definição do plano.
N200 X-20 Y-20 Z10	; canto interno com bloco anterior.
N210 X-30 Y10 A4=1 C4=1	; canto interno, nova definição do plano.
N220 A3=1 B3=0.5 C3=1.7	; mudança de orientação com ORIC.
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; mudança de orientação.
N250 X0 Y30 C4=1	; deslocamento da trajetória, novo plano, ; orientação com programação relativa.
N260 BSPLINE X20 Z15	; início do Spline, programação relativa da orientação
N270 X30 Y25 Z18	; permanece com Spline ativo.
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; hélice, orientação com programação de eixos.
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; deslocamento da trajetória, orientação, plano não constante.
N330 X-20 G40	; Desativação da correção do raio da ferramenta.
N340 M30	

Exemplo 2: Um programa NC criado em um sistema CAD (recorte) com CUT3DFD

Código de programa	Comentário
N01 G710	

Código de programa	**Comentário**
N03 T="12"	
N06 S5305 M03	
N07 G642	
; aproximação da posição de partida em MKS levando em conta o comprimento da ferramenta.	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; final do posicionamento em MKS.	
;	
TRAORI(1)	; selecionar transformação da orientação.
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; chamada CYCLE832 com: ; tolerância de contorno = 0,01 mm, ; tipo de usinagem: Top Surface com alisamento, ; acabamento com introdução de uma tolerância de orientação, ; tolerância de orientação = 1 grau
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; os blocos N09 até N10 são a parte rápida do movimento de aproximação na peça com orientação constante.
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; nos blocos N11 até N21 é realizada a parte lenta do movimento de aproximação. A ferramenta está pronta próxima à peça e, além disso, as configurações para construção da forma (por ex. COMPSURF) do CYCLE832 já estão ativas (pela ativação de G01). A trajetória dessa assim chamada fase de balanço para o comportamento da construção da forma, deve ser de aprox.1000 vezes a tolerância do contorno (neste exemplo é de 10 mm).
N12 X-247.69126 Y33.82182 Z24.78219 F1061.00000	
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	

Código de programa	Comentário
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	; a partir de N22, a normal à superfície é, pela primeira vez, totalmente definida através do bloco completo (ou seja, a normal à superfície no final do bloco anterior N21 está disponível, assim como a normal à superfície no início do bloco N22 e a normal à superfície no final do bloco N22). Desse modo, o pré-requisito para a ligação da correção da ferramenta com G41/G42 é preenchido.
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	
...	

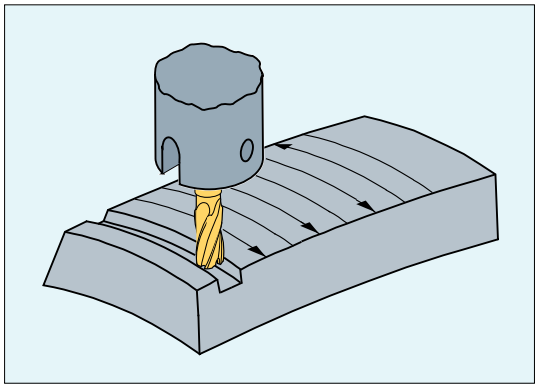
Indicação

No fresamento de topo 3D com CUT3DFD, a definição da normal à superfície é necessária para a ligação da correção da ferramenta com G41/G42. A programação de G41/G42 sem definição da normal à superfície causa a emissão de um alarme.

Outras informações

Fresamento de topo 3D

Para este tipo de fresamento 3D é necessária a descrição das trajetórias 3D sobre a superfície da peça de trabalho. Normalmente no CAM os cálculos são executados considerando-se a forma e dimensões da ferramenta. O pós-processador, além dos blocos NC, grava no programa de peças as orientações da ferramenta (com transformação de 5 eixos ativada) e o comando G da correção de ferramenta 3D desejada. Com isso, o operador da máquina tem a opção de empregar ferramentas muito menores do que as empregadas no cálculo dos percursos NC.



Esquema 11-3 Fresamento de topo

Comportamento de aproximação

O comportamento de aproximação é sempre NORM com as variantes em 3D da correção do raio da ferramenta.

Comportamento em cantos externos

Os cantos externos são tratados como círculos de raio 0 no fresamento de topo, onde o plano do círculo da tangente final do primeiro e a tangente inicial do segundo bloco é estendido. Por isso, uma mudança de orientação na transição do bloco também é possível. Com isso, um círculo é sempre inserido como elemento de contorno em um canto externo. O procedimento de ponto de corte não está disponível para fresamento de topo.

Comportamento na alteração de orientação em cantos externos

Através dos comandos G ORIC e ORID, pode-se estabelecer se foram realizadas alterações de orientação que foram programadas entre ambos os blocos que constituem o canto, antes do início do bloco do círculo introduzido (ORID) ou simultaneamente a ele (ORIC).

Correção do raio da ferramenta relacionada a uma ferramenta de diferença

Uma correção do raio de ferramenta 3D relacionada a uma ferramenta de diferença é ativada pelo comando CUT3DFD. Ela deve ser usada, se o contorno programado for relacionado ao percurso do ponto central de uma ferramenta de normalizada e o processamento ocorre com ferramenta diferente. Então, no cálculo da correção do raio de ferramenta 3D, será incluído somente o valor do desgaste do raio da ferramenta ativa ($ TC_ DP15) e os offsets de correção da ferramenta eventualmente programados OFFN e TOFFR. O raio da base($TC_DP6) da ferramenta ativa **não** será calculado.

O fresamento de topo 3D com CUT3DFD só é possível em combinação com o "alisamento da normal à superfície no fresamento de topo 3D". Este é ativado pelo acesso à função de licença obrigatória "Top Surface" através de CYCLE832(...) (Página 815). A ativação precisa ser realizada **antes** da ligação da correção da ferramenta com G41/G42, e obviamente não logo antes da entrada da ferramenta, mas antes em um comprimento de trajetória que corresponda a aprox. 1000 vezes a tolerância do contorno (por ex. 1000 x 0,01 mm = 10 mm). A desativação precisa ocorrer em sequência inversa: primeiro, desligar o corretor de ferramenta com G40, então, conforme um comprimento de trajetória, que corresponde cerca de 1000 à tolerância do contorno, a desativação, através, por exemplo, CUT2D (ou semelhante).

Para poder aplicar o "alisamento da normal à superfície no fresamento de topo 3D", é necessária também a liberação da função "Interpolação da normal à superfície através de polinômios":

MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

Indicação

Para o fresamento de topo 3D com CUT3DFD em combinação com "Top Surface" devem ser observadas as recomendações de ajuste em relação à "Top Surface"!

Para a verificação dos dados ajustados, estão disponíveis (https://support.industry.siemens.com/cs/ww/en/view/109738423) pelo portal SIOS programas especiais de verificação.

11.5.3 Fresamento periférico 3D levando em conta uma superfície de limitação (CUT3DCC, CUT3DCCD)

No fresamento periférico 3D com alteração contínua ou constante da orientação da ferramenta, muitas vezes se programa o percurso do centro da ferramenta para uma ferramenta normalizada definida. Visto que, na prática, frequentemente não há disponibilidade de ferramentas normalizadas apropriadas, a utilização de uma ferramenta não excessivamente divergente (≤ 5 %) á admissível.

Com CUT3DCCD se considera uma superfície de limitação para uma ferramenta diferencial, que descreveria a ferramenta normalizada programada. O programa NC descreve a trajetória do centro da ferramenta normalizada.

Com CUT3DCC se considera uma superfície de limitação com o uso de ferramentas cilíndricas, que teria alcançado a ferramenta normalizada programada. O programa NC descreve o contorno na superfície de usinagem.

O vetor da normal à superfície das superfícies de limitação, é definido para fresamento de topo 3D por A4, B4, C4 e A5, B5, C5.

Sintaxe

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

Significado

CUT3DCCD:	WRK 3D para fresamento periférico levando em conta uma superfície de limitação, com ferramenta de diferença, na trajetória do centro da ferramenta: Penetração até a superfície de limitação
CUT3DCC:	WRK 3D para fresamento periférico levando em conta uma superfície de limitação com correção de raio 3D: Contorno na superfície de usinagem

G41/G42 X... Y... Z...:	Ligação da correção do raio da ferramenta	
	G41:	Correção do raio da ferramenta à esquerda do contorno
	G42:	Correção do raio da ferramenta à direita do contorno
	Indicação: a ligação precisa ser realizada em um bloco linear (G0/G1).	
CDOF2:	Ligarsupervisão de colisão para fresamento periférico 3D	
A4/5=... B4/5=... C4/5=...:	Definição da normal à superfície da superfície de limitação	
	A4=... B4=... C4=...:	Definição no início do bloco
	A5=... B5=... C5=...:	Definição no final do bloco
G40 X... Y... Z...:	Desativar a correção do raio da ferramenta **Indicação:** o desligamento precisa ser realizado em um bloco linear (G0/G1) com deslocamento do eixo geométrico.	

Indicação

Os comandos G para ativação da WRK 3D são avaliados no bloco de aproximação, ou seja, normalmente no bloco que contém G41 ou G42.

G41 ou G42 também podem ser programados em blocos sem movimento de deslocamento nos eixos geométricos relevantes para a correção. Nesse caso, o bloco de início é o primeiro em um bloco desse tipo após um bloco de deslocamento.

Uma mudança da variante WRK 3D com a correção do raio da ferramenta ativa é ignorada sem alarme.

Exemplo

```
Código de programa                              Comentário
N10 $TC_DP1[1,1]=120                            ; fresa cilíndrica
N20 $TC_DP6[1,1]=10
N30 $TC_DP15[1,1]=-3
...
; usinagem com fresa cilíndrica e CUT3DCCD
N110 TRAORI                                     ; ligação da transformação.
N120 A4=0 B4=0 C4=1                             ; definição da normal à superfície da su-
                                                perfície de limitação no início do bloco.
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000
N140 X10 Y0 Z0 G41 CUT3DCCD CDOF2 G64           ; ativação do fresamento periférico 3D le-
                                                vando em conta a superfície de limitação
                                                + desligamento da supervisão de colisão.
N150 X20
N160 X30 A45                                    ; ângulo obtuso ==> nenhuma penetração
N170 X40 A-45                                   ; ângulo agudo ==> penetração
N180 X55
N190 Y10 Z10                                    ; movimento na direção da ferramenta.
N200 Y20
```

Código de programa	**Comentário**
N210 C45	; apenas mudança de orientação.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; alteração da superfície.
N240 Y50 G40	; desativação da correção do raio da ferramenta.
...	

Outras informações

Tipo de ferramenta

O tipo de ferramenta (parâmetro da ferramenta $TC_DP1) é avaliado. Apenas fresas com eixo cilíndrico são permitidas (fresa cilíndrica ou fresa de topo, fresa toroidal e, como caso limite, fresa cilíndrica para matrizes). Isso corresponde aos tipos de ferramenta 1 - 399 com exceção dos números 111 e 155 até 157.

Ferramenta normalizada com arredondamento de cantos

O arredondamento de cantos da ferramenta normalizada é descrito pelo parâmetro de ferramenta $TC_DP7. A partir do parâmetro de ferramenta $TC_DP16, resulta o desvio do arredondamento de canto da ferramenta real comparado com a ferramenta normalizada.

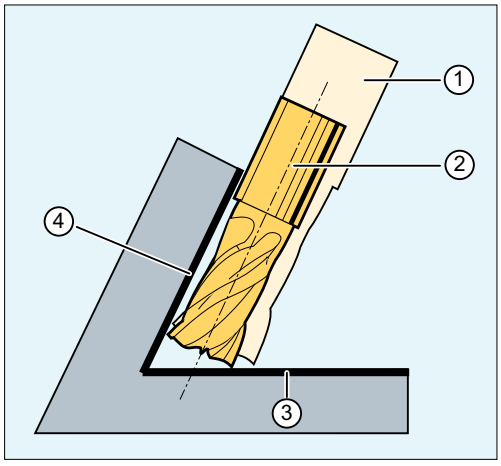
Exemplo: Fresa toroidal com raio reduzido se comparada com uma ferramenta normal

Tipo de ferramenta	Raio da haste (R)	Raio do canto (r)
Ferramenta normalizada com arredondamento de cantos	R = $TC_DP6	r = $TC_DP7
Ferramenta real com arredondamento nos cantos Tipos de ferramenta 121 e 131, fresa toroidal (fresa de topo com arredondamento nos cantos)	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16

Neste exemplo tanto o $TC_DP15 + OFFN como o $TC_DP16 são negativos.

WRK 3D com CUT3DCCD: Trajetória do centro da ferramenta com penetração até a superfície de limitação

Se for utilizada uma ferramenta com um raio menor em relação à ferramenta normalizada correspondente, então uma fresa penetrada no sentido longitudinal será conduzida até que atinja novamente a base do bolsão. Com isso, o canto formado pela superfície de usinagem e a superfície de limitação será desbastado até onde a ferramenta permitir. Neste caso trata-se de um modo de usinagem misto de fresamento periférico e fresamento de topo. Semelhante à uma ferramenta com raio reduzido, em uma ferramenta com raio aumentado a penetração é realizada no sentido contrário.



① Ferramenta normalizada
② Ferramenta de menor raio penetrada até a superfície de limitação
③ Superfície de limitação
④ Superfície de usinagem

Em comparação com as outras correções da ferramenta do grupo 22 de G, um parâmetro de ferramenta $TC_DP6 especificado para CUT3DCCD não é relevante para o raio da ferramenta e não tem influência sobre a correção resultante. O offset de correção é determinado pela soma do valor de desgaste do raio da ferramenta (parâmetro da ferramenta $TC_DP15) e um offset da ferramenta programado OFFN para o cálculo do offset perpendicular à superfície de limitação.

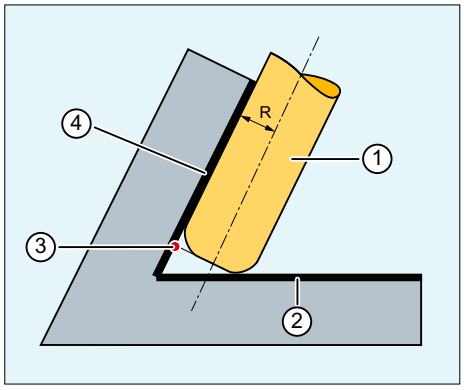
A partir do programa de peças criado não é possível saber se a superfície de usinagem está à esquerda ou à direita do percurso. Por isso que se parte do princípio de um raio positivo e um valor negativo de desgaste da ferramenta original. Um valor negativo de desgaste sempre descreve uma ferramenta com diâmetro reduzido.

Utilização de ferramentas cilíndricas

Na utilização de ferramentas cilíndricas uma penetração apenas será necessária se a superfície de usinagem e a superfície de limitação formarem um ângulo agudo (menor que 90 graus). Se forem utilizadas fresas toroidais (fresas de topo com arredondamento de cantos), então a penetração no sentido longitudinal da ferramenta será necessária tanto em ângulos agudos como obtusos.

WRK 3D com CUT3DCC: Contorno na superfície de usinagem

Se CUT3DCC estiver ativo com uma fresa toroidal, então a trajetória programada toma como referência uma fresa cilíndrica fictícia de mesmo diâmetro. O ponto de referência do percurso resultante é representado na figura a seguir com a utilização de uma fresa toroidal.



① Fresa toroidal
② Superfície de limitação
③ Ponto de referência da trajetória
④ Superfície de usinagem
R Raio da haste (raio da ferramenta)

É permitido que o ângulo entre as superfícies de usinagem e de limitação também pode passar de um ângulo agudo para um ângulo obtuso ou vice-versa dentro de um bloco.

Ao contrário da ferramenta normalizada, a ferramenta real utilizada pode ser tanto maior do que menor. Neste caso o raio de canto resultante não pode se tornar negativo e o sinal do raio de ferramenta resultante deve ser preservado.

No CUT3DCC o programa NC de peça tem sua referência no contorno da superfície de usinagem. Aqui, como na correção convencional do raio da ferramenta, trata-se do raio total, que é constituído pelos seguintes componentes:

- Raio de ferramenta (parâmetro de ferramenta $TC_DP6)
- Valor de desgaste (parâmetro de ferramenta $TC_DP15)
- Um offset da ferramenta OFFN programado para o cálculo do offset perpendicular à superfície de limitação

A posição da superfície de limitação é determinada a partir da seguinte diferença:

Dimensões da ferramenta normalizada - raio da ferramenta (parâmetro WZ $TC_DP6)

Advanced Surface / Top Surface

Indicação

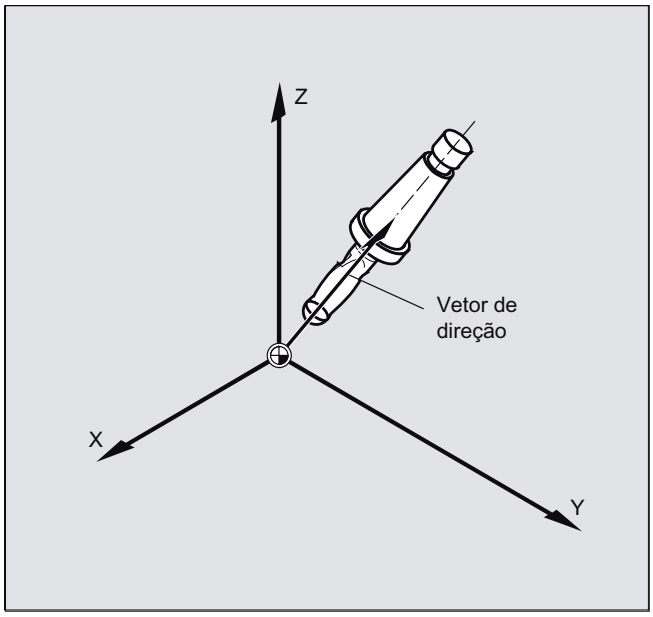
Na utilização da correção do raio de ferramenta CUT3DCC / CUT3DCCD em combinação com a função de licença obrigatória "Advanced Surface" ou "Top Surface" devem ser observadas as recomendações de ajuste relacionadas ao "Advanced Surface" / "Top Surface"!

Para a verificação dos dados ajustados, estão disponíveis pelo portal SIOS programas especiais de verificação:

- Programa de verificação para Advanced Surface (https://support.industry.siemens.com/cs/ww/en/view/78956392)
- Programa de verificação para Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)

11.6 Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Como orientação da ferramenta entendemos o alinhamento geométrico da ferramenta no espaço. Em uma máquina de usinagem de 5 eixos a orientação da ferramenta pode ser ajustada através de comandos de programa.



Os movimentos de suavização ativados com `OSD` e `OST` são formados de modo diferente dependendo do tipo de interpolação da orientação da ferramenta.

Com a interpolação de vetor ativa também se interpola o decurso de orientação suavizado através da interpolação de vetor. Em contrapartida, na interpolação de eixos rotativos ativa, a orientação é suavizada através de movimentos de eixo rotativo.

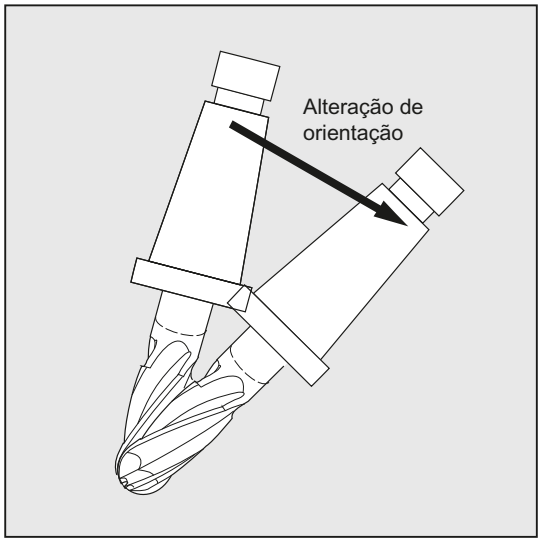
Programação

Programação da alteração de orientação:

Uma alteração da orientação da ferramenta pode ser programada através de:

- programação direta dos eixos rotativos `A`, `B` e `C` (interpolação de eixos rotativos)
- Ângulo euleriano ou ângulo RPY
- Vetor de direção (interpolação de vetor através da especificação do `A3` ou `B3` ou `C3`)
- `LEAD/TILT` (fresamento de topo)

O sistema de coordenadas de referência é o sistema de coordenadas da máquina (`ORIMKS`) ou o atual sistema de coordenadas da peça (`ORIWKS`).



Programação da orientação da ferramenta:

ORIC:	Orientação e movimento de percurso paralelo
ORID:	Orientação e movimento de percurso sucessivamente
OSOF:	sem suavização da orientação
OSC:	Orientação constante
OSS:	Suavização da orientação apenas no começo do bloco
OSSE:	Suavização da orientação no começo e no fim do bloco
ORIS:	Velocidade da alteração da orientação com suavização de orientação ativada em graus por mm (vale para OSS e OSSE)
OSD:	Suavização da orientação através da especificação da extensão de suavização com o dado de ajuste: SD42674 $SC_ORI_SMOOTH_DIST
OST:	Suavização da orientação através da especificação da tolerância angular em graus na interpolação de vetor com o dado de ajuste: SD42676 $SC_ORI_SMOOTH_TOL Na interpolação de eixos rotativos se assume a tolerância especificada como o desvio máximo dos eixos de orientação.

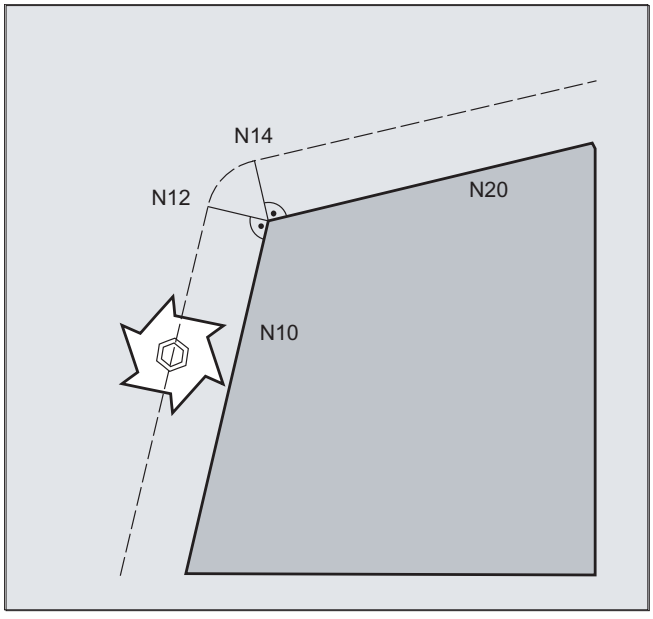
Indicação

Todos os comandos para suavização da orientação de ferramenta (OSOF, OSC, OSS, OSSE, OSD e OST) estão agrupados no grupo de G 34. Elas estão ativas modalmente, ou seja, sempre apenas um destes comandos poderá atuar.

Exemplos

Exemplo 1: ORIC

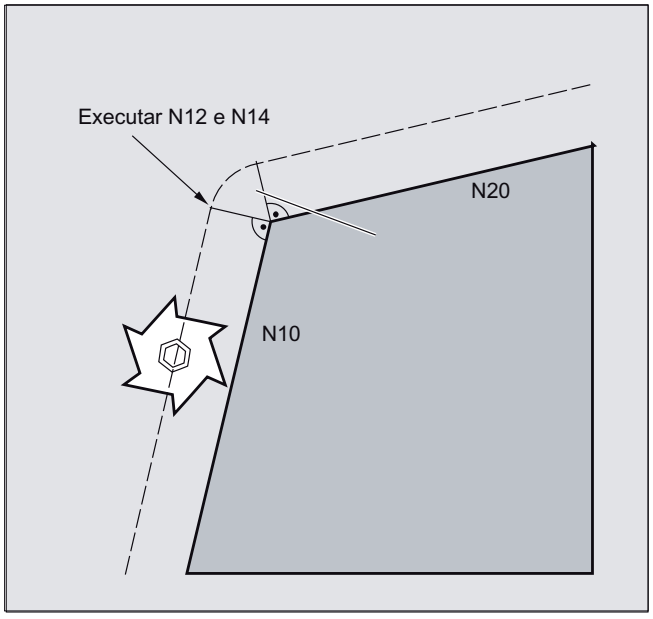
Se entre os blocos de deslocamento `N10` e `N20` foram programados dois ou mais blocos com alterações de orientação (p. ex. `A2=... B2=... C2=...`) e o `ORIC` estiver ativo, então o bloco circular inserido será divido nestes blocos intermediários de acordo com o valor das alterações angulares.



Código de programa	Comentário
ORIC	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 C2=… B2=…	; O bloco circular, que é inserido no canto externo, distribui-se entre o N12 e o N14, conforme a alteração da orientação. Neste caso o movimento circular e a alteração da orientação são executados paralelamente.
N14 C2=… B2=…	
N20 X =…Y=… Z=… G1 F200	

Exemplo 2: ORID

Se o `ORID` estiver ativo, então serão executados todos blocos entre os dois blocos de deslocamento no fim do primeiro bloco de deslocamento. O bloco circular com orientação constante é executado imediatamente antes do segundo bloco de deslocamento.



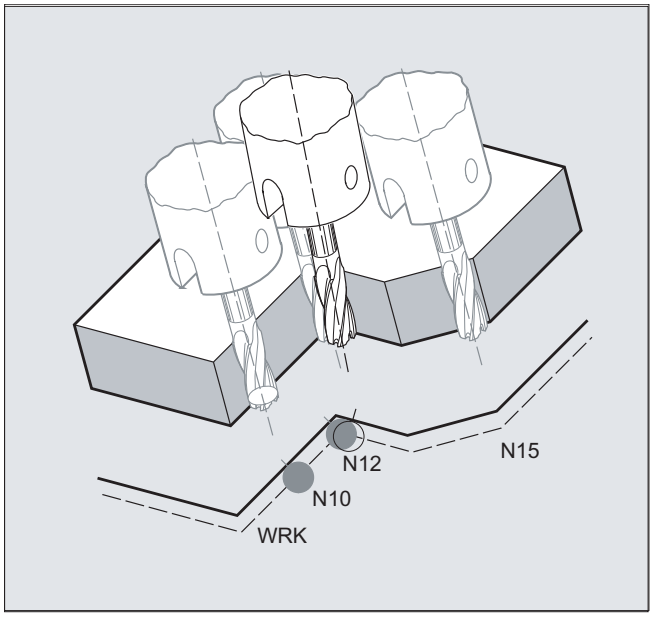
Código de programa	**Comentário**
ORID	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 A2=… B2=… C2=…	; O bloco N12 e N14 é executado no fim do N10. Em seguida é executado o bloco circular com a atual orientação.
N14 M20	; Funções auxiliares, etc.
N20 X… Y… Z…	

Indicação

Para o tipo da alteração da orientação em um canto externo será determinante o comando de programa que estiver ativo no primeiro bloco de deslocamento de um canto externo.

Sem alteração da orientação: Se a orientação no limite de bloco não for alterada, então a secção transversal da ferramenta será um círculo que tem contato em ambos contornos.

Exemplo 3: Alteração da orientação em um canto interno



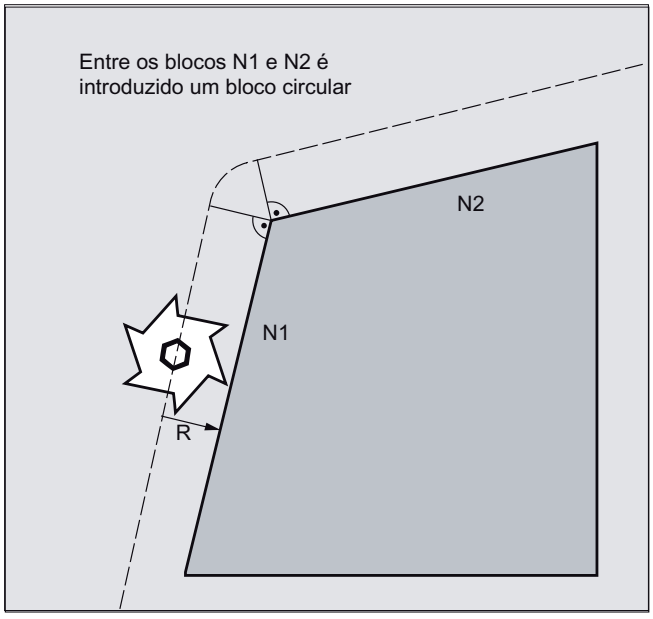
Código de programa
ORIC
N10 X …Y… Z… G1 F500
N12 X …Y… Z… A2=… B2=… C2=…
N15 X …Y… Z… A2=… B2=… C2=…

Outras informações

Comportamento em cantos externos

Em um canto externo sempre é inserido um bloco circular com o raio da fresa.

Com os comandos de programa `ORIC` ou `ORID` se pode definir se as alterações de orientação que foram programadas entre os blocos `N1` e `N2` serão executadas antes do início do bloco circular inserido, ou simultaneamente com este.



Se uma alteração de orientação for necessária em cantos externos, então ela pode ser realizada paralelamente à interpolação ou separadamente com o movimento de percurso.

Com `ORID` são executados primeiramente os blocos inseridos sem movimento de percurso. O bloco circular é inserido imediatamente antes do segundo dos dois blocos que formam o canto.

Se em um canto externo foram inseridos vários blocos de orientação e selecionado o `ORIC`, então o movimento circular será distribuído sobre os vários blocos de acordo com os valores das alterações de orientação dos diversos blocos inseridos.

Suavização da orientação com OSD ou OST

Na suavização com `G642` o desvio máximo para eixos de contorno e eixos de orientação não pode ser muito diferente. A menor tolerância dos dois determina a forma do movimento de suavização ou tolerância angular, para suavizar o decurso de orientação relativamente forte, sem precisar de maiores desvios de contorno.

Através da ativação do `OSD` ou do `OST` é possível, com uma extensão de suavização especificada ou tolerância angular, suavizar com facilidade desvios muito pequenos do decurso de orientação sem desvios de contorno muito grandes.

Indicação

Diferente da suavização do contorno (e do decurso de orientação) com `G642`, na suavização da orientação com `OSD` ou `OST` não se forma nenhum bloco próprio, mas se insere o movimento da suavização diretamente nos blocos originais programados.

Com `OSD` ou `OST` não pode ser suavizada nenhuma transição de blocos onde existe uma mudança de tipos de interpolação para orientação da ferramenta (vetor → eixo rotativo, eixo rotativo → vetor). Estas transições de blocos podem, eventualmente, ser suavizados com as funções convencionais de suavização `G641`, `G642` e `G643`.

11.7 Seleção livre de número D, número de gume

11.7.1 Atribuição livre de números D, número de corte (endereço CE)

Número D

Os números D podem ser utilizados como números de correção. Além disso se pode endereçar o número do corte através do endereço CE. Através da variável de sistema $TC_DPCE se pode descrever o número de corte.

Pré-ajuste: Número de correção == número de cortes

Através dos dados de máquina se pode definir o número máximo de números D (números de corte) e o número máximo de cortes por ferramenta (→ Fabricante da máquina). Os comandos a seguir apenas têm relevância se o número máximo de cortes (MD18105) for definido maior do que o número de cortes por ferramenta (MD18106). Observe as informações do fabricante da máquina.

Indicação

Além da atribuição relativa de números D, os números D também podem ser atribuídos como números D "planos" ou "absolutos" (1-32000) sem referência a um número T (dentro da função "Estrutura plana de números D").

Literatura

Manual de funções básicas; Correção de ferramenta (W1)

11.7.2 Atribuição livre de números D: Verificar números D (CHKDNO)

Com o comando `CKKDNO` verificamos se os números D presentes foram especificados de forma única. Os números D de todas ferramentas definidas dentro de uma unidade TO podem aparecer apenas uma vez. As ferramentas de reposição não são consideradas neste caso.

Sintaxe

```
state=CHKDNO(Tno1,Tno2,Dno)
```

Significado

`state`:	=TRUE:	Os números D foram especificados exclusivamente para a faixa verificada.
	=FALSE:	Ocorreu uma colisão de números D ou a parametrização é inválida. Através de Tno1, Tno2 e Dno são transmitidos os parâmetros que geram a colisão. Estes dados podem ser avaliados no programa de peça.

CHKDNO(Tno1,Tno2):	São verificados todos os números D das ferramentas mencionadas.
CHKDNO(Tno1):	São verificados todos números D de Tno1 contra todas outras ferramentas.
CHKDNO:	São verificados todos números D de todas ferramentas contra todas outras ferramentas.

11.7.3 Atribuição livre de números D: Renomear números D (GETDNO, SETDNO)

Os números D devem ser especificados como únicos. Dois cortes diferentes de uma ferramenta não podem ter o mesmo número D.

GETDNO

Este comando retorna o número D de um determinado corte (ce) de uma ferramenta de número T (t). Se não existir nenhum número D para os parâmetros especificados, se define d=0. Se o número D for inválido, será retornado um valor acima de 32000.

SETDNO

Com este comando atribuímos o valor d do número D de um corte ce da ferramenta t. Através do state é retornado o resultado desta instrução (TRUE ou FALSE). Se não houver nenhum bloco de dados para os parâmetros especificados, será retornado FALSE. Os erros de sintaxe geram um alarme. O número D não pode ser passado explicitamente para 0.

Sintaxe

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

Significado

d:	Número D do corte da ferramenta
T:	Número T da ferramenta
ce:	Número do corte (número CE) da ferramenta
state:	Indica se o comando poderia ser executado sem erros (TRUE ou FALSE).

Exemplo de renomeação de um número D

Programação	**Comentário**
$TC_DP2[1,2]=120	
$TC_DP3[1,2] = 5.5	
$TC_DPCE[1,2] = 3	; Número de corte CE
...	
N10 def int DNoAnt, DNoNov = 17	
N20 NoDAnt = GETDNO(1,3)	
N30 SETDNO(1,3,NoDNov)	

Com isso o novo valor D 17 é atribuído para o corte CE=3. Agora os dados deste corte são endereçados através do número D 17; tanto através de variáveis de sistema como na programação com o endereço NC.

11.7.4 Atribuição livre de números D: Determinar o número T para o número D especificado (GETACTTD)

Com o comando `GETACTTD` determinamos o número T pertencente a um número F absoluto. Não é realizada nenhuma verificação quanto à exclusividade. Se existirem vários números D iguais dentro de uma unidade TO, será retornado o número T da primeira ferramenta encontrada. Com a utilização de números D "planos" não existe relevância no uso do comando, pois aqui sempre é retornado o valor 1 (nenhum número T no banco de dados).

Sintaxe

```
status=GETACTTD(Tno,Dno)
```

Significado

`Dnr`:	Número D que deve ser procurado para o número T.	
`Tnr`:	Número T encontrado	
`status`:	Valor:	Significado:
	0	O número T foi encontrado. O Tno contém o valor do número T.
	-1	Para o número D indicado não existe nenhum número T; Tno=0.
	-2	O número D não é absoluto. O Tno contém o valor da primeira ferramenta encontrada que contém o número D com o valor Dno.
	-5	A função não pôde ser executada por outro motivo.

11.7.5 Atribuição livre de números D: Invalidar números D (DZERO)

O comando `DZERO` serve para dar suporte durante a mudança de ferramentas. Os blocos de dados de correção marcados não são mais verificados pelo comando `CHKDNO`. Para acessá-los novamente, o número D deve ser ajustado novamente com `SETDNO`.

Sintaxe

```
DZERO
```

Significado

`DZERO`:	Marca todos números D da unidade TO como inválidos.

11.8 Cinemática do porta-ferramenta

Pré-requisitos

Um porta-ferramenta somente pode orientar uma ferramenta sem todas possíveis direções no espaço se

- dois eixos de rotação V_1 e V_2 estiverem disponíveis.
- os eixos de rotação estão perpendiculares entre si.
- o eixo longitudinal da ferramenta está perpendicular em relação ao segundo eixo de rotação V_2.

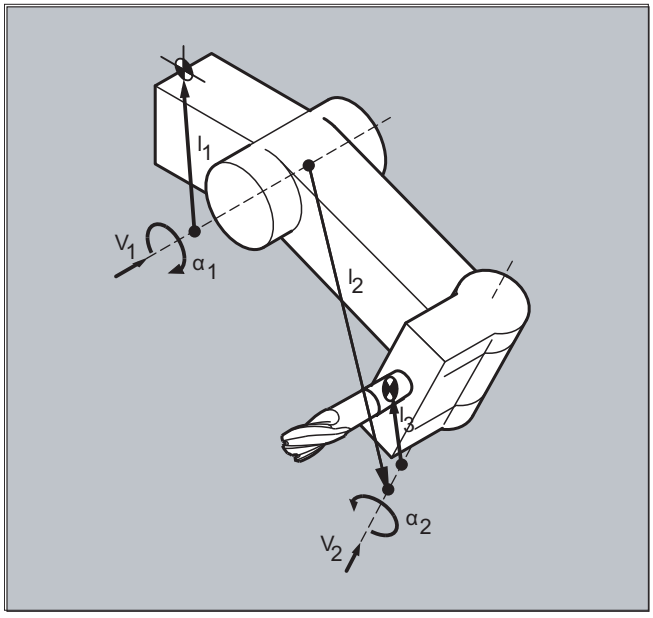
Em máquinas onde todas possíveis orientações devem ser ajustáveis, também se aplica o seguinte:

- a orientação da ferramenta deve estar perpendicular em relação ao primeiro eixo de rotação V_1.

Função

A cinemática da ferramenta com no máximo dois eixos de rotaçãov_1 ou v_2 é descrita pelas 17 variáveis de sistema $TC_CARR1[m] bis $TC_CARR17[m]. A descrição do porta-ferramenta é composta por:

- a distância vetorial do primeiro eixo de rotação até o ponto de referência do porta-ferramenta I_1, a distância vetorial do primeiro e segundo eixo de rotação I_2, a distância vetorial do segundo eixo de rotação até o ponto de referência da ferramenta I_3.
- os vetores de direção dos dois eixos de rotação V_1, V_2.
- ângulo de rotação α1, α2 em torno dos dois eixos. Os ângulos de rotação são contados positivos no sentido de visualização dos vetores dos eixos rotativos em sentido horário.



Para máquinas **com cinemática dissipada** (gira tanto a ferramenta como a peça) as variáveis de sistema são ampliadas com os registros $TC_CARR18[m] para $TC_CARR23[m] .

Parâmetro

Função das variáveis de sistema para porta-ferramentas orientáveis			
Denominação	**Componente x**	**Componente y**	**Componente z**
l_1 Vetor de Offset	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
l_2 Vetor de Offset	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
v_1 Eixo de rotação	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
v_2 Eixo de rotação	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
α_1 Ângulo de rotação α_2 Ângulo de rotação	$TC_CARR13[m] $TC_CARR14[m]		
l_3 Vetor de Offset	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]

Ampliações das variáveis de sistema para porta-ferramentas orientáveis			
Denominação	**Componente x**	**Componente y**	**Componente z**
l_4 Vetor de Offset	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
Identificador de eixo do eixo de rotação v_1 do eixo de rotação v_2	Identificador de eixo dos eixos de rotação v_1e v_2 (a ocupação prévia é zero) $TC_CARR21[m] $TC_CARR22[m]		
Tipo de cinemática **T**ool **P**art **M**ixed mode	$TC_CARR23[m]		
	Tipo de cinemática T ->	Tipo de cinemática P ->	Tipo de cinemática M
	Apenas a ferramenta pode ser girada (ocupação prévia)	Apenas a peça de trabalho gira	A peça de trabalho &e a ferramenta giram
Offset do eixo de rotação v_1 do eixo de rotação v_2	Ângulo em graus dos eixos de rotação v_1e v_2 com a introdução da posição inicial $TC_CARR24[m] $TC_CARR25[m]		
Offset de ângulo do eixo de rotação v_1 do eixo de rotação v_2	Offset dos dentes Hirth em graus dos eixos de rotação v_1e v_2 $TC_CARR26[m] $TC_CARR27[m]		
Increm. angular v_1 eixo de rotação v_2 eixo de rotação	Incremento dos dentes Hirth em graus dos eixos de rotação v_1e v_2 $TC_CARR28[m] $TC_CARR29[m]		
Posição mín. do eixo de rotação v_1 do eixo de rotação v_2	Limite de software para posição mínima dos eixos de rotação v_1e v_2 $TC_CARR30[m] $TC_CARR31[m]		

Ampliações das variáveis de sistema para porta-ferramentas orientáveis			
Posição máx. do eixo de rotação v_1 do eixo de rotação v_2	Limite de software para posição máxima dos eixos de rotação v_1e v_2 $TC_CARR32[m] $TC_CARR33[m]		
Nome do porta-ferramentas	Ao invés de um número um porta-ferramenta pode receber um nome. $TC_CARR34[m]		
Usuário: Nome de eixo 1 Nome de eixo 2 Identificação **Posição**	Uso intencional durante os ciclos de medição do usuário. $TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
Desloc. fino	Parâmetros que podem ser adicionados **aos valores nos parâmetros básicos**.		
l_1 Vetor de Offset	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
l_2 Vetor de Offset	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
l_3 Vetor de Offset	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
l_4 Vetor de Offset	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
v_1 Eixo de rotação	$TC_CARR64[m]		
v_2 Eixo de rotação	$TC_CARR65[m]		

Indicação

Explicações para os parâmetros

Com "m" é especificado o número do porta-ferramenta a ser descrito.

O $TC_CARR47 para $TC_CARR54 assim como o $TC_CARR61 para $TC_CARR63 não estão definidos e geram um alarme quando é realizada uma tentativa de leitura ou gravação.

Os pontos inicial e final dos vetores de distância podem ser selecionados livremente nos eixos. Os ângulos de rotação α_1, α_2 em torno dos dois eixos são definidos com 0° no estado inicial do porta-ferramenta. A cinemática de um porta-ferramenta pode ser descrita de várias formas diferentes.

Porta-ferramentas com apenas um ou nenhum eixo de rotação podem ser descritos para um ou ambos eixos de rotação quando zeramos os vetores de direção.
Em um porta-ferramenta sem eixo de rotação atuam os vetores de distância assim como as correções adicionais do equipamento, cujos componentes não são afetados durante a mudança dos planos de usinagem (G17 para G19)°.

Ampliação dos parâmetros

Parâmetros dos eixos de rotação

As variáveis de sistema foram ampliadas com os registros $TC_CARR24[m] até $TC_CARR33[m] e descritas como segue:

O **Offset** dos eixos de rotação v_1, v_2	Alteração da posição do eixo de rotação v_1 ou v_2 na posição inicial do porta-ferramenta orientável.
O **Offset angular/ incremento angular** dos eixos de rotação v_1, v_2	Offset do incremento angular dos dentes Hirth dos eixos de rotação v_1 e v_2. O ângulo programado ou calculado é arredondado para o próximo valor que, no caso do valor inteiro n, resulta do phi = s + n * d.
Posição mínima e máxima dos eixos de rotação v_1, v_2	A posição mínima/máxima do eixo de rotação é o ângulo de limite (limite de software) dos eixos de rotação v1 e v2.

Parâmetros para o usuário

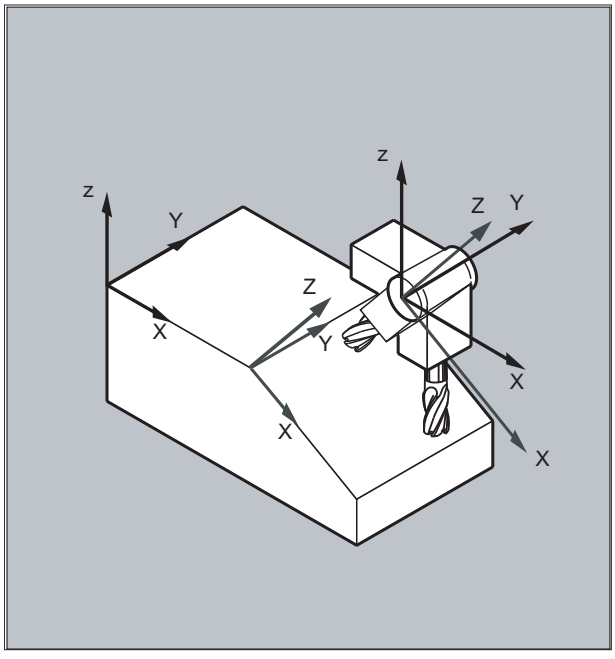
O $TC_CARR34 até $TC_CARR40 contém parâmetros que o usuário tem livre acesso e que, até o SW 6.4, como padrão na NCK, não são mais avaliados ou são irrelevantes.

Parâmetros do deslocamento fino

O $TC_CARR41 até $TC_CARR65 contém parâmetros de deslocamento fino que podem ser adicionados aos valores nos parâmetros básicos. O valor de deslocamento fino atribuído em um parâmetro básico é formado quando se adiciona o valor 40 ao número de parâmetro.

Exemplo

O porta-ferramenta utilizado no exemplo a seguir pode ser completamente descrito por uma rotação em torno do eixo Y.



Código de programa	Comentário
N10 $TC_CARR8[1]=1	; Definição do componente Y do primeiro eixo de rotação do porta-ferramenta 1.
N20 $TC_DP1[1,1]=120	; Definição da fresa de topo.
N30 $TC_DP3[1,1]=20	; Definição de uma fresa de topo com 20 mm de comprimento.
N40 $TC_DP6[1,1]=5	; Definição de uma fresa de topo com 5 mm de raio.
N50 ROT Y37	; Definição de Frame com rotação de 37° em torno do eixo Y.
N60 X0 Y0 Z0 F10000	; Aproximar da posição de partida.
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; Ajustar a correção do raio, correção do comprimento da ferramenta no Frame girado, selecionar porta-ferramenta 1 e ferramenta 1.
N80 X40	; Executar a usinagem com a rotação de 37°.
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Outras informações

Cinemática resolvida

Para máquinas com cinemática resolvida (se gira tanto a ferramenta como a peça) as variáveis de sistema são ampliadas com os registros `$TC_CARR18[m]` até `$TC_CARR23[m]` e descritas como segue:

A mesa giratória de ferramentas é composta pelo(a):

- distância vetorial do segundo eixo rotativo V_2 até o ponto de referência de uma mesa giratória de ferramenta I_4 do terceiro eixo de rotação.

Os eixos rotativos são compostos pelo(a):

- dois identificadores de canal para a referência dos eixos de rotação V_1e V_2, cuja posição se acessa eventualmente na definição da orientação do porta-ferramenta orientável.

O tipo de cinemática com um dos valores T, P ou M:

- Tipo de cinemática T: Apenas a ferramenta gira.
- Tipo de cinemática P: Apenas a peça de trabalho gira.
- Tipo de cinemática M: A ferramenta e a peça de trabalho giram.

Deletação dos dados do porta-ferramenta

Com `$TC_CARR1[0]=0` pode-se deletar os dados de todos blocos de dados do porta-ferramenta.

O tipo de cinemática `$TC_CARR23[T]=T` pode ser especificado com uma das três letras maiúsculas ou minúsculas permitidas (T,P,M) e por este motivo não deveria ser deletado.

Alteração dos dados do porta-ferramenta

Cada um dos valores descritos pode ser alterado com a atribuição de um novo valor no programa de peça. Todo caractere diferente de T, P ou M gera um alarme na tentativa de se ativar o porta-ferramenta orientável.

Leitura dos dados do porta-ferramenta

Cada um dos valores descritos pode ser lido no programa de peça através da atribuição em uma variável.

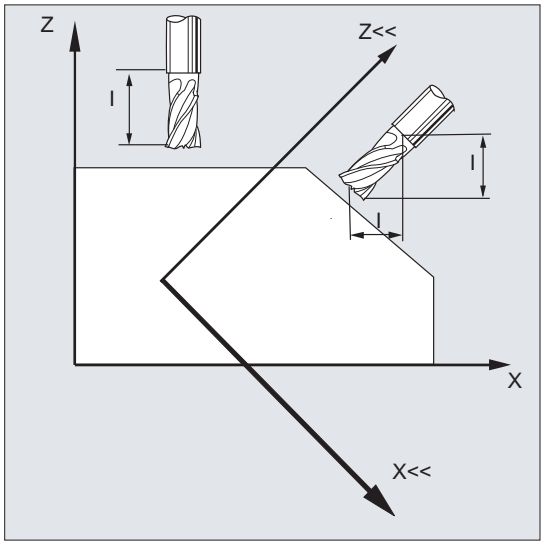
Deslocamentos finos

Um valor de deslocamento fino não permitido apenas será detectado quando estiver ativado um porta-ferramenta orientável que tem este tipo de valor e ao mesmo tempo o dado de ajuste SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE.

O valor do deslocamento fino permitido é limitado em um valor máximo permitido através de dados de máquina.

11.9 Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Com a orientação espacial alterada da ferramenta também são alterados os componentes das mesmas.



Após uma mudança, p. ex. através de ajuste manual ou troca do porta-ferramentas com alinhamento fixo no espaço, se deve determinar novamente os componentes de comprimento da ferramenta. Isto se realiza com os comandos de curso `TCOABS` e `TCOFR`.

Em um porta-ferramenta orientável de um Frame, na seleção de ferramenta com `TCOFRZ`, `TCOFRY` e `TCOFRX`, se pode definir o sentido em que a ferramenta deve apontar.

Sintaxe

```
TCARR=[<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

Significado

`TCARR=[<m>]`:	Solicitar porta-ferramenta com o número "m"
`TCOABS`:	Calcular os componentes de comprimento da ferramenta a partir da atual orientação do porta-ferramenta
`TCOFR`:	Definir os componentes de comprimento da ferramenta a partir da orientação do Frame ativo
`TCOFRZ`:	Porta-ferramenta orientável do Frame ativo cuja ferramenta aponta para o sentido Z

TCOFRY:	Porta-ferramenta orientável do Frame ativo cuja ferramenta aponta para o sentido Y
TCOFRX:	Porta-ferramenta orientável do Frame ativo cuja ferramenta aponta para o sentido X

Outras informações

Correção de comprimento de ferramenta a partir da orientação do suporte (TCOABS)

O TCOABS calcula a correção de comprimento da ferramenta a partir dos atuais ângulos de orientação do porta-ferramenta; armazenada nas variáveis de sistema $TC_CARR13 e $TC_CARR14.

Para definição da cinemática do porta-ferramenta com variáveis de sistema, veja " Cinemática do porta-ferramenta (Página 466) ".

Para recálculo da correção de comprimento da ferramenta na mudança de Frames a ferramenta deve ser selecionada novamente.

Direção da ferramenta a partir do Frame ativo

O porta-ferramenta orientável pode ser ajustado de modo que a ferramenta aponte para as seguintes direções:

- com TCOFR ou TCOFRZ no sentido Z
- com TCOFRY no sentido Y
- com TCOFRX no sentido X

Uma comutação entre TCOFR e TCOABS aciona um recálculo da correção de comprimento da ferramenta.

Solicitar porta-ferramenta (TCARR)

Com TCARR são solicitados os dados geométricos (memória de correções) do número de porta-ferramenta m.

Com m=0 se desseleciona o porta-ferramenta ativo.

Os dados geométricos do porta-ferramenta são ativados apenas após a chamada de uma ferramenta. A ferramenta selecionada permanece ativa durante a mudança de um porta-ferramenta.

Os atuais dados geométricos do porta-ferramenta também podem ser definidos no programa de peça através das respectivas variáveis de sistema.

Recálculo da correção de comprimento da ferramenta (TCOABS) na mudança de Frames

Para recálculo da correção de comprimento da ferramenta na mudança de Frames a ferramenta deve ser selecionada novamente.

Indicação

A orientação de ferramenta deve ser adaptada manual ao Frame ativo.

No recálculo da correção de comprimento da ferramenta também é calculado o ângulo de rotação do porta-ferramenta em um passo intermediário. Visto que porta-ferramentas com

dois eixos de rotação normalmente existem em dois pares de ângulo de rotação com os quais a orientação da ferramenta pode ser adaptada ao Frame ativo, os valores dos ângulos de rotação devem, de forma aproximada, corresponder pelo menos aos ângulos de rotação ajustados mecanicamente.

Indicação

Orientação da ferramenta

O comando não pode verificar o ângulo de rotação calculado através da orientação de Frame com o ajustado na máquina.

Se os eixos de rotação do porta-ferramenta construtivamente não puderem alcançar a orientação de ferramenta calculada através da orientação de Frame, então será emitido um alarme.

Não é permitida combinação da correção fina da ferramenta e as funcionalidades para correção de comprimento de ferramenta em porta-ferramentas móveis. Numa tentativa de ativar as duas funções ao mesmo tempo, ocorre uma mensagem de erro.

Com `TOFRAME` é possível definir um Frame com base no sentido de orientação do porta-ferramenta selecionado. Para informações mais detalhadas, veja o capítulo "Frames".

Com a transformação de orientação (transformação de 3, 4 e 5 eixos) ativa, um porta-ferramenta de orientação fora da posição zero pode ser selecionado sem que um alarme seja emitido.

Parâmetro de transferência de ciclos padrão e ciclos de medição

Para o parâmetro de transferência de ciclos padrão e ciclos de medição são aplicadas faixas de valores definidas.

No caso dos valores angulares a faixa de valores está definida da seguinte forma:

- Rotação em torno do 1º eixo geométrico: -180 graus até +180 graus
- Rotação em torno do 2º eixo geométrico: -90 graus até +90 graus
- Rotação em torno do 3º eixo geométrico: -180 graus até +180 graus

Veja o capítulo Frames, "Rotação programável (ROT, AROT, RPL)".

Indicação

Na transferência de valores angulares em um ciclo padrão ou ciclo de medição deve-se observar o seguinte:

Valores menores que a unidade de cálculo do NC devem ser arredondados para zero!

A unidade de cálculo do NC para posições angulares está definida no dado de máquina:

MD10210 $MN_INT_INCR_PER_DEG

11.10 Correção Online de comprimento de ferramenta (TOFFON, TOFFOF)

Através da variável de sistema $AA_TOFF[<n>] os comprimentos efetivos de ferramenta podem ser sobrepostos em tempo real de acordo com os três sentidos da ferramenta.

Como índice <n> são utilizados os três identificadores de eixo geométrico. Com isso se define o número de sentidos de correção ativos através dos eixos geométricos durante o mesmo tempo.

Todas correções podem estar ativas simultaneamente.

A função da correção de comprimento de ferramenta Online pode ser aplicada no(a):

- Transformação de orientação TRAORI
- Porta-ferramenta orientável TCARR

Indicação

A correção de comprimento de ferramenta Online é um **opcional** que deve ser liberado antes. Esta função apenas tem utilidade se trabalhar em conjunto com uma transformação ativa da orientação ou com um porta-ferramenta orientável ativo.

Sintaxe

```
TRAORI
TOFFON(<sentido de correção>[,<valor de Offset>])
WHEN TRUE DO $AA_TOFF[<sentido de correção>]            ; Em ações síncronas.
...
TOFFOF(<sentido de correção>)
```

Para mais explanações sobre programação da correção de comprimento de ferramenta Online em ações síncronas de movimentos, veja o capítulo "Ações síncronas (Página 631)".

Significado

TOFFON:	**Ativação** da correção de comprimento de ferramenta Online	
	<sentido de correção>:	Sentido de correção (X, Y, Z) onde a correção de comprimento da ferramenta Online deve estar ativo.
	<valor Offset>:	Na ativação se pode especificar um valor de Offset para o respectivo sentido de correção que será imediatamente executado.
TOFFOF:	**Resetamento** da correção de comprimento de ferramenta Online Os valores de correção no sentido de correção especificado são resetados e se ativa uma parada de pré-processamento.	

Exemplos

Exemplo 1: Seleção da correção de comprimento da ferramenta

Código de programa	Comentário
MD21190 $MC_TOFF_MODE =1	; Os valores absolutos são aproximados.
MD21194 $MC_TOFF_VELO[0] =1000	
MD21196 $MC_TOFF_VELO[1] =1000	
MD21194 $MC_TOFF_VELO[2] =1000	
MD21196 $MC_TOFF_ACCEL[0] =1	
MD21196 $MC_TOFF_ACCEL[1] =1	
MD21196 $MC_TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	; Ativar a transformação.
N20 TOFFON(Z)	; Ativação da correção on-line do comprimento da ferramenta para o sentido Z da ferramenta.
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; Para o sentido Z da ferramenta é interpolada uma correção do comprimento da ferramenta de 10.
...	
N100 XOFFSET=$AA_TOFF_VAL[X]	; Atribuir atual correção no sentido X.
N120 TOFFON(X,-XOFFSET) G4 F5	; Para o sentido X da ferramenta a correção do comprimento da ferramenta é novamente retornada como 0.

Exemplo 2: Desativação da compensação de comprimento de ferramenta

Código de programa	Comentário
N10 TRAORI(1)	; Ativar a transformação.
N20 TOFFON(X)	; Ativação da correção on-line do comprimento da ferramenta para o sentido X da ferramenta.
N30 WHEN TRUE DO $AA_TOFF[X] = 10 G4 F5	; Para o sentido X da ferramenta é interpolada uma correção do comprimento da ferramenta de 10.
...	
N80 TOFFOF(X)	; O Offset de posição do sentido X da ferramenta é deletado: ...$AA_TOFF[X]=0 Não é movimentado nenhum eixo. Para a atual posição em WCS é adicionado o deslocamento de posição conforme a atual orientação.

Outras informações

Preparação de blocos

Para preparação de blocos no pré-processamento é considerado o atual deslocamento de comprimento da ferramenta ativo no processamento principal. Para permitir um uso extenso das velocidades máximas permitidas dos eixos, é necessário parar a preparação de blocos com uma parada de pré-processamento `STOPRE` enquanto um deslocamento de ferramenta é constituído.

O deslocamento da ferramenta sempre é conhecido no momento do pré-processamento se as correções do comprimento da ferramenta não forem mais alteradas após o início do programa ou se após uma alteração das correções do comprimento da ferramenta não forem mais processados blocos que o buffer IPO pode suportar entre o pré-processamento e o processamento principal.

Variável $AA_TOFF_PREP_DIFF

Na variável $AA_TOFF_PREP_DIFF[<n>] é possível consultar a medida da diferença entre a atual correção ativa no interpolador e a correção que estava ativa no momento da preparação de blocos.

Ajustar dados da máquina e dados de ajuste

Para correção de comprimento de ferramenta Online estão disponíveis os seguintes dados de sistema:

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE (Reserva de aceleração para movimento sobreposto)
- MD21190 $MC_TOFF_MODE
  O conteúdo da variável de sistema $AA_TOFF[<n>] é executado ou integrado como valor absoluto.
- MD21194 $MC_TOFF_VELO (Velocidade da correção do comprimento de ferramenta Online)
- MD21196 $MC_TOFF_ACCEL (Aceleração da correção do comprimento de ferramenta Online)
- Dado de ajuste para especificação de valores de limite:
  SD42970 $SC_TOFF_LIMIT (Limite superior do valor de correção do comprimento da ferramenta)

Literatura:
Manual de funcionamento, funções especiais; F2: Transformações múltiplas

11.11 Modificação dos dados de correção em ferramentas giratórias

11.11.1 Cálculo das orientações (ORISOLH)

A função pré-definida ORISOLH auxilia o usuário a ajustar as posições do eixo rotativo de uma máquina, de modo que uma ferramenta rotativa seja colocada em uma posição definida e cinematicamente independente com relação à peça. O pré-requisito é que uma transformação de 6 eixos esteja ativa, a qual é parametrizada com sequências cinemáticas.

Existem duas funções básicas disponíveis:

- Alinhar ferramenta
  Os dois ângulos β e γ são pré-definidos. A função calcula os ângulos dos três eixos de orientação necessários para isso.
- Alinhamento direto da ferramenta
  Os ângulos do segundo e do terceiro eixo de orientação são pré-fixados. Então, a função calcula os respectivos ângulos β e γ, assim como o ângulo que falta do primeiro eixo de orientação.

Indicação

Sequência dos eixos de orientação

Quando se percorre a sequência cinemática descrita pela configuração da máquina, desde a peça até a ferramenta, então valem as seguintes afirmações para a sequência dos três eixos de orientação de uma transformação de 6 eixos:

- O eixo de orientação que se encontra mais próximo da**peça** é o **primeiro** eixo de orientação.
- O eixo de orientação mais próximo à**ferramenta** é o **terceiro** eixo de orientação.

O primeiro eixo de orientação é normalmente um fuso e a sua rotação é, por isso e nesse caso, realizada por um Frame rotativo.

Sintaxe

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

Significado

ORISOLH:	Chamada da função		
<RetVal>:	Valor de devolução da função		
	Tipo de dado:	INT	
	Faixa de valores:	0, -2, -3, ..., -17	
	Valores:	0	A função foi concluída sem erro.
		-2	Nenhuma transformação válida (transformação de orientação de 6 eixos) está ativa.
		-3	O primeiro parâmetro (<Cntrl>) é negativo.
		-4	O primeiro dígito do primeiro parâmetro (<Cntrl>) não é válido. Apenas os valores 0 e 1 são permitidos.
		-5	O décimo dígito do primeiro parâmetro (<Cntrl>) não é válido. Apenas os valores de 0 até 3 são permitidos.
		-6	O centésimo dígito do primeiro parâmetro (<Cntrl>) não é válido. Apenas os valores 0 e 1 são permitidos.
		-7	O milésimo dígito do primeiro parâmetro (<Cntrl>) não é válido. Apenas os valores de 0 até 3 são permitidos.
		-8	Na função "Alinhamento direto da ferramenta" o ângulo γ é muito grande.
		-9	Na função "Alinhamento direto da ferramenta", pelo menos uma das posições de eixo pré-definidas ultrapassa um limite de eixo.
		-10	Nenhuma ferramenta ativa.
		-11	A orientação estendida não é ajustável.
		-12	O ajuste do ângulo de eixo livre no acoplamento Hirth não é possível para a primeira solução ou solução única.
		-13	O ajuste do ângulo de eixo livre no acoplamento Hirth não é possível para a segunda solução.
		-14	O ajuste do ângulo de eixo livre no acoplamento Hirth não é possível para nenhuma das duas soluções.
		-15	O primeiro eixo de orientação é parametrizado como um eixo Hirth.
		-16	E tanto o segundo como o terceiro eixo rotativo são parametrizados como eixo Hirth. No máximo um dos dois eixos pode ser um eixo Hirth.
		-17	Na função "Rotação direta", pelo menos uma das posições de eixo pré-definidas não é compatível com o respectivo acoplamento Hirth.

<Cntrl>:	Comando do comportamento da função		
	Tipo de dado:	INT	
	O parâmetro <Cntrl> tem codificação decimal (1º até o 1000º dígito):		
	1º dígito:	O 1º dígito comanda o comportamento em caso de erros.	
		xxx**0**	Em caso de erro (valor retornado < 0), o processamento do programa é interrompido e o alarme 14106 é emitido. **Indicação:** o alarme também é emitido, independente do valor do 1º dígito, se o parâmetro <Cntrl> for negativo.
		xxx**1**	Em caso de erro (valor retornado < 0) **nenhum** alarme é emitido. O usuário tem a possibilidade de reagir adequadamente por si mesmo no programa.
	10º dígito:	Comanda o comportamento no caso de haver um eixo de orientação com acoplamento Hirth. **Indicação:** esse parâmetro só é avaliado na função "Alinhamento da ferramenta" (ou seja, quando o 100º dígito tiver o valor "0").	
		xx**0**x	A posição do eixo é arredondada para a próxima posição existente.
		xx**1**x	As posições do eixo são arredondadas de modo que o desvio do ângulo β com relação ao seu valor programado seja mínimo.
		xx**2**x	As posições do eixo são arredondadas de modo que o ângulo β seja igual ao valor máximo possível, que é menor que o valor programado (β é arredondado).
		xx**3**x	As posições do eixo são arredondadas de modo que o ângulo β seja igual ao valor mínimo possível, que é maior que o valor programado (β é arredondado).
	100º dígito:	Indica qual função precisa ser executada ou qual significado tem os dois parâmetros seguintes <W1> e <W2>.	
		x**0**xx	Função "Alinhar ferramenta" Os parâmetros <W1> e <W2> têm o seguinte significado: • <W1> = β • <W2> = γ O ângulo correspondente do eixo de orientação é calculado.
		x**1**xx	Função "Alinhamento direto da ferramenta" <W1> é a posição pré-definida para o segundo eixo de orientação, <W2> é a posição pré-definida para o terceiro eixo de orientação de uma transformação de 6 eixos. A posição do primeiro eixo de orientação e os ângulos β e γ são determinados de modo a serem compatíveis com as duas posições pré-definidas. Caso ocorra um erro, são fornecidas sempre duas soluções nas variáveis do sistema $P_ORI_POS[<n>, <m>]. Então o primeiro índice <n> (0 ou 1) corresponde à solução e o segundo índice <m> (0 ... 2) ao eixo de orientação: • $P_ORI_POS[0/1, **0**]: Posição do primeiro eixo de orientação • $P_ORI_POS[0/1, **1**]: Ângulo β

			• $P_ORI_POS[0/1, **2**]: Ângulo γ É verificado se as posições pré-definidas <W1> e <W2> são compatíveis com eventuais acoplamentos Hirth ativos ou limites de software ativos. Se isso não for o caso, é retornado o número de erro correspondente (ver parâmetro <RetVal>). No caso de uma seleção livre dos dois ângulos <W1> e <W2>, o corte da ferramenta em geral não pertence ao plano de usinagem. O ângulo γ, pelo qual o corte está desviado para fora do plano de usinagem, não pode ser maior que o valor limite determinado pelo dado de ajuste SD42999 $SC_ORISOLH_INCLINE_TOL.
	1000º dígito:	Indica como as posições das soluções precisam ser eventualmente modificadas, quando o 100º dígito tiver o valor "0", ou seja, na função "Alinhamento da ferramenta".	
		0xxx	As posições dos eixos calculadas precisam estar o mais perto possível das posições dos eixos da máquina.
		1xxx	As posições de eixo calculadas para eixos módulo precisam estar o mais perto possível do centro do intervalo módulo e, para os outros eixos, o mais perto possível de 0. Para eixos não módulo, isso significa que as posições dos eixos precisam ser reduzidas para o intervalo -180° … +180°.
		2xxx	As posições de eixo calculadas, precisam ser reduzidas para o intervalo -180° … +180° independente do tipo do eixo.
<W1>:	Primeiro ângulo O significado é obtido do 100º dígito do parâmetro <Cntrl>.		
	Tipo de dado:	REAL	
<W1>:	Segundo ângulo O significado é obtido do 100º dígito do parâmetro <Cntrl>.		
	Tipo de dado:	REAL	

Indicação

Parâmetros não programados têm o valor padrão "0".

Outras informações

A quantidade das soluções encontradas, junto com outros dados de estado, na execução da função ORISOLH, pode ser lida através das seguintes variáveis de sistema:

Variável de sistema	Significado		
$P_ORI_POS [<n>, <m>]	Fornece o ângulo dos eixos de orientação obtidos na programação da orientação.		
	<n>:	índice da solução	
		Faixa de valores:	0, 1
	<m>:	Índice do eixo de orientação	
		Faixa de valores:	0 ... 2
			A sequência dos eixos de orientação (1 ... 3) refere-se à definição dos eixos em $NT_ROT_AX_NAME.
	Chamando-se a função ORISOLH no modo "Alinhamento direto da ferramenta", as variáveis $P_ORI_POS[0/1, 1] e P_ORI_POS[0/1, 2] contêm os valores pertencentes às duas soluções dos dois ângulos β e γ. A primeira solução registrada em $P_ORI_POS[<n>, <m>], ou seja, com o índice <n> = 0, é sempre aquela selecionada pelo comando quando se faz uma aproximação direta para a orientação requerida. O segundo índice <m> refere-se ao eixo de orientação, ou seja, a $NT_ROT_AX_NAME. As posições dos eixos registradas em $P_ORI_POS[<n>, <m>] levam em conta os offsets registrados em $NK_OFF e $NK_OFF_FINE, ou seja, esses ângulos de eixo podem ser usados nos seguintes blocos sem outras modificações, para ajustar a orientação desejada. Se o eixo rotativo for um eixo Hirth, as posições das soluções são arredondadas para a próxima posição de repouso disponível do acoplamento Hirth. Nas variáveis de sistema $P_ORI_DIFF, as diferenças das posições dos eixos podem ser lidas, com eixos rotativos com dentes Hirth, para as soluções exatas e para as soluções ajustadas para engates Hirth.		
$P_ORI_DIFF [<n>, <m>]	Fornece a diferença entre as posições exatas e as posições definidas disponíveis em $P_ORI_POS, dos eixos de orientação obtidos na programação da orientação.		
	<n>:	índice da solução	
		Faixa de valores:	0, 1
	<m>:	Índice do eixo de orientação	
		Faixa de valores:	0 ... 2
			A sequência dos eixos de orientação (1 ... 3) refere-se à definição dos eixos em $NT_ROT_AX_NAME.
	Nesse caso, o conteúdo só pode ser diferente de zero quando as posições são engatadas (dentes Hirth), ou seja, quando o dado do sistema $NT_HIRTH_INCR do eixo em questão for diferente de zero e quando esse eixo for um eixo rotativo manual.		

Variável de sistema	Significado		
$P_ORI_SOL	Se for calculado o ângulo do eixo em uma transformação de orientação com mais de um eixo de orientação, que deve conduzir a uma orientação pré-programada, existe em geral mais de uma solução. Nas variáveis de sistema $P_ORI_SOL está incluída a quantidade de soluções válidas, junto com dados de estado complementares. O conteúdo de $P_ORI_SOL é codificado como segue:		
	Valores < 0	Condições gerais de erro	
		-1	Para transformação ativa, mais nenhuma solução é calculada (falta chamada de ORISOLH).
		-2	Se não houver nenhuma transformação ativa, ou se a transformação ativa não for uma transformação de orientação (transformação de 6 eixos), que possa fornecer as posições a uma programação de orientação pré-definida.
		-4	A orientação desejada não é ajustável com a cinemática especificada.
		-5	Chamando-se a função ORISOLH no modo "Alinhamento direto da ferramenta", nenhuma solução foi encontrada.
		-6	Chamando-se a função ORISOLH no modo "Alinhamento direto da ferramenta", o ângulo γ é muito grande.
		-7	Chamando-se a função ORISOLH no modo "Alinhamento direto da ferramenta", foi pré-definido um ângulo que não é ajustável devido ao acoplamento Hirth.
		-8	O primeiro eixo de orientação (eixo Frame) não pode ser parametrizado como um eixo Hirth.
		-9	E tanto o segundo como o terceiro eixo rotativo são parametrizados como eixo Hirth. No máximo um dos dois eixos pode ser um eixo Hirth.
		-10	Não foi encontrado nenhum ajuste da(s) solução(ões) para os dentes do eixo Hirth.
	Valores > 0 1º dígito	Quantidade de soluções matemáticas possíveis sem levar em conta limitações de eixo e eventuais condições de erro.	
		0	Não há nenhuma solução, ou seja, a orientação estendida não é ajustável. Esse caso pode ter três causas diferentes: • A orientação estendida não é, em princípio, alcançável devido à cinemática da máquina (os eixos de orientação não são perpendiculares), inclusive com uma faixa de movimento qualquer dos eixos de orientação. Nesse caso, o 10º e o 100º dígito de $P_ORI_SOL são nulos, e as variáveis de status $P_ORI_STAT associadas aos eixos de orientação contém o valor "-4". • As soluções calculadas não são alcançáveis porque elas ultrapassam os limites dos eixos. As posições dos eixos de orientação, obtidas sem limitações de eixo, podem ser lidas em $P_ORI_POS. • Chamando-se a função ORISOLH no modo "Alinhamento direto da ferramenta", são pré-definidas posições de eixo de modo que o vetor de orientação, ou o vetor normal de

Variável de sistema	Significado		
			orientação, da ferramenta esteja alinhado em paralelo ao primeiro eixo de orientação do qual a posição precisa ser calculada. A posição desse eixo é nesse caso não definida.
		1	Existe uma solução. Esse caso pode ter três causas diferentes: • Devido à orientação pré-definida e à cinemática da máquina, mesmo sem levar em conta as limitações de eixo, existe somente uma solução (do ponto de vista matemático, trata-se de duas soluções coincidentes). Esse caso ocorre, para cinemáticas não perpendiculares, no limite do intervalo de orientação. $P_ORI_POS contém ambas as soluções (idênticas). • Existe apenas uma solução, porque uma segunda solução não seria válida devido à ultrapassagem das limitações de eixo. A solução válida é, por isso, sempre a primeira solução em $P_ORI_POS. A segunda solução, obtida sem levar em conta as limitações de eixo, pode ser igualmente lida em $P_ORI_POS. • Chamar a função ORISOLH no modo "Alinhamento direto da ferramenta", é o caso normal. Para as posições de eixo pré-definidas com dois eixos de orientação, obtém-se em geral apenas uma posição válida a ser calculada para o eixo de orientação que falta.
		2	Existem duas soluções.
		8	Existem infinitas soluções, ou seja, a posição de um eixo de orientação (do eixo polar) é livre. Das duas posições possíveis do(s) outro(s) eixo(s), uma é, no entanto, excluída devido à ultrapassagem das limitações de eixo.
		9	Existem infinitas soluções, ou seja, a posição de um eixo de orientação (do eixo polar) não é determinada. O eixo não determinado pode ser determinado a partir do 100º dígito ou a partir das variáveis de sistema $P_ORI_STAT.
	Valores > 0 10º dígito	Indicação codificada em bits para limitações de eixo ultrapassadas. A causa exata da falha pode ser determinada a partir das variáveis de sistema $P_ORI_STAT.	
		Bit 0 (valor 10):	Para pelo menos uma solução existe pelo menos um limite de eixo do 1º eixo de orientação ultrapassado.
		Bit 1 (valor 20):	Para pelo menos uma solução existe pelo menos um limite de eixo do 2º eixo de orientação ultrapassado.
		Bit 2 (valor 40):	Para pelo menos uma solução existe pelo menos um limite de eixo do 3º eixo de orientação ultrapassado.
	Valores > 0 100º dígito	Indicação codificada em bits para posições de eixo não definidas (somente pode ocorrer quando houverem infinitas soluções, ou seja, quando o 1º dígito for igual a "9").	

Variável de sistema	Significado		
		Bit 0 (valor 100):	A posição do 1º eixo de orientação não está definida.
		Bit 1 (valor 200):	A posição do 2º eixo de orientação não está definida.
		Bit 2 (valor 400):	A posição do 3º eixo de orientação não está definida.
	As designações 1º, 2º e 3º eixo de orientação referem-se à definição dos eixos em $NT_ROT_AX_NAME.		

Variável de sistema	Significado			
$P_ORI_STAT [<n>]	Fornece para cada um de no máximo 3 eixos de orientação, a condição após a chamada de ORISOLH.			
	<n>:	Índice do eixo de orientação (corresponde ao índice do eixo de orientação em questão em $NT_ROT_AX_NAME)		
		Faixa de valores:	0 ... 2	
			A sequência dos eixos de orientação (1 ... 3) refere-se à definição dos eixos em $NT_ROT_AX_NAME.	
	O conteúdo de $P_ORI_STAT é codificado como segue:			
	Valores < 0	Condições gerais de erro		
		-1	A condição não está definida (falta a chamada de ORISOLH).	
		-2	Se não houver nenhuma transformação ativa, ou se a transformação ativa não for uma transformação de orientação (transformação de 6 eixos), que possa fornecer as posições a uma programação de orientação pré-definida.	
		-3	O eixo não está incluído na transformação ativa.	
		-4	A posição do eixo não pode ser calculada porque, com a cinemática calculada, a orientação requerida não é alcançável, mesmo com a faixa de movimento do eixo livremente adotada.	
		-5	Chamando-se a função ORISOLH no modo "Alinhamento direto da ferramenta", são pré-definidas posições de eixo de modo que o vetor de orientação, ou o vetor normal de orientação, da ferramenta esteja alinhado em paralelo ao primeiro eixo de orientação do qual a posição precisa ser calculada. A posição desse eixo é nesse caso não definida.	
		-6	Chamando-se a função ORISOLH no modo "Alinhamento direto da ferramenta", o ângulo γ é muito grande.	
		-7	Chamando-se a função ORISOLH no modo "Alinhamento direto da ferramenta", foi pré-definido um ângulo que não é ajustável devido ao acoplamento Hirth.	
		-8	O primeiro eixo de orientação (eixo Frame) não pode ser parametrizado como um eixo Hirth.	
		-9	E tanto o segundo como o terceiro eixo rotativo são parametrizados como eixo Hirth. No máximo um dos dois eixos pode ser um eixo Hirth.	
		-10	Não foi encontrado nenhum ajuste da(s) solução(ões) para os dentes do eixo Hirth.	
	Valores > 0 1º dígito	Indicação codificada em bits para limitações de eixo violadas da primeira solução.		
		Bit 0 (valor 1):	A primeira solução viola o limite inferior do eixo.	
		Bit 1 (valor 2):	A primeira solução viola o limite superior do eixo.	
	Valores > 0 10º dígito	Indicação codificada em bits para limitações de eixo violadas da segunda solução.		
		Bit 0 (valor 10):	A segunda solução viola o limite inferior do eixo.	
		Bit 1 (valor 20):	A segunda solução viola o limite superior do eixo.	
	Valores > 0	Indicação de uma posição de eixo não definida.		

Variável de sistema	Significado		
	100º dígito	Bit 0 (valor 100):	A posição do eixo de orientação não está definida, ou seja, a orientação requerida pode ser obtida com qualquer ajuste do eixo rotativo (posição polar). Essa informação também está contida nas variáveis de sistema $P_ORI_SOL.
	A partir dos números de erro que indicam uma violação das limitações de eixo, vários podem ocorrer simultaneamente. Se no caso da violação de um limite de eixo for tentada, pela adição ou subtração múltiplos de 360°, uma posição a ser atingida dentro das limitações de eixo permitidas e, se isso não for possível, não fica perfeitamente definido se o limite inferior do eixo ou o limite superior do eixo foi violado. Não existe nenhuma solução para a orientação requerida ($P_ORI_SOL = 0) se a condição dos eixos de orientação contidos na transformação for "0".		

Indicação

$NT_ROT_AX_NAME

Com essas variáveis de sistema, no máximo 3 eixos são eliminados, que são usados para ajustar a orientação. Eles contêm os nomes dos elos de corrente ($NK_NAME) que definem qualquer eixo da máquina (eixos rotativos) que precise realizar movimentos de orientação e que sejam obtidos a partir de uma transformação cinemática. A sequência na qual no máximo três eixos rotativos estão contidos nessas variáveis de sistema, não tem significado para a cinemática da máquina, porque eles foram removidos da estrutura da sequência cinemática. Mas como eles definem a sequência com a qual as outras variáveis têm acesso aos eixos rotativos, a sequência dos eixos de orientação em $NT_ROT_AX_NAME deve coincidir com a descrição cinemática.

Indicação

Informações de status

As informações de condição que indicam, por ex., que uma orientação não é alcançável ou seria alcançável apenas com a violação de limitações de eixo relevantes, não conduzem a um alarme NC. É de responsabilidade do usuário reagir adequadamente às condições citadas.

11.11.2 Ativação da alteração dos dados de correção para ferramentas orientáveis (CUTMOD, CUTMODK)

A alteração dos dados de correção para ferramentas orientáveis é ativada no programa NC através do comando de voz CUTMOD (em conjunto com porta-ferramentas orientáveis) ou CUTMODK (para transformações de orientação que são definidas através de sequências cinemáticas).

Indicação

Como porta-ferramentas orientáveis e transformações de orientação que são definidas através de sequências cinemáticas, não podem estar ativos simultaneamente, não há conflitos entre as duas variantes.

Sintaxe

```
CUTMOD = <Value>
```

ou

```
CUTMODK = <Command>
```

Significado

CUTMOD:	Chamada de função em conjunto com porta-ferramentas orientáveis		
<Valor>:	Valor designado		
	Tipo de dado:	INT	
	Valor:	0	A função está desativada. Os valores retornados pelas variáveis $P_AD... são iguais aos parâmetros de ferramenta correspondentes.
		> 0	A função é ativada caso um porta-ferramenta orientável com o número especificado esteja ativo, ou seja, a ativação está associada a um determinado porta-ferramenta orientável. Os valores retornados pelas variáveis de sistema $P_AD..., ao contrário dos parâmetros de ferramenta correspondentes, eventualmente, são modificados em função da rotação ativa. A desativação do porta-ferramenta orientável identificado apenas desativa temporariamente a função, a ativação de outro porta-ferramenta orientável o desativa permanentemente. Por isso, no primeiro caso a função é reativada com uma nova seleção do mesmo porta-ferramenta, e no segundo caso, a nova seleção é necessária, mesmo se mais tarde o porta-ferramenta orientável for ativado novamente com o número especificado. A função não é influenciada através do Reset.
		-1	A função sempre é ativada se um porta-ferramenta orientável estiver ativo. Na troca do porta-ferramenta ou no caso desse ser desativado e depois novamente ativado, se deve ativar novamente o CUTMOD.
		-2	A função sempre é ativada se um porta-ferramenta orientável estiver ativo e seu número for igual ao atual porta-ferramenta orientável ativo. Se nenhum porta-ferramenta orientável estiver ativo, isto tem o mesmo significado que CUTMOD=0. Se um porta-ferramenta orientável estiver ativo, isto tem o mesmo significado que a indicação imediata do atual número de porta-ferramenta.
		< -2	Valores menores que 2 são ignorados, isto é, este caso é tratado como se o CUTMOD não fosse programado. **Indicação:** este intervalo de valores não deve ser utilizado, pois ele está reservado para eventuais futuras ampliações.
CUTMODK:	Chamada de função em conjunto com transformações de orientação que são definidas através de sequências cinemáticas		

<Command>:	Comando designado		
	Tipo de dado:	STRING	
	Valor:	"NEW"	Para a "Alteração dos dados de correção", são armazenadas condições relevantes de uma transformação ativa definida com sequências cinemáticas, o nome da transformação e o Frame de contorno atual. **Indicação:** esse comando só é permitido se uma transformação adequada (TRAORI_DYN, TRAORI_STAT ou TRAANG_K) estiver ativa.
		"OFF"	Desliga a "Alteração dos dados de correção". Os dados salvos anteriormente com "NEW“ são mantidos. **Indicação:** esse comando também é permitido se CUTMODK não estiver ativo. Então ele permanece sem ação. Um eventual bloco de dados existente para "Alteração dos dados de correção" é mantido.
		"ON"	Com esse comando, a "Modificação dos dados de correção" é reativada com um bloco de dados armazenado anteriormente com o comando "NEW". Se na execução desse comando, estiver ativa uma transformação com o nome do bloco de dados armazenado, a "Alteração dos dados de correção" é ativada imediatamente. Caso contrário, a alteração é retardada até que uma transformação ativa seja ativada.
		"CLEAR"	Desliga, como o comando "OFF", a "Alteração dos dados de correção" e elimina também os blocos de dados armazenados. **Indicação:** esse comando também é permitido se CUTMODK não estiver ativo.

Indicação

SD42984 $SC_CUTDIRMOD

A função ativável através do comando CUTMOD ou CUTMODK, substitui a função ativável através do dado de ajuste SD42984 $SC_CUTDIRMOD. Entretanto esta função continua disponível sem ser alterada. Visto que não faz sentido utilizar as duas funções paralelamente, ela somente pode ser ativada se CUTMOD for igual a zero e CUTMODK for igual à sequência nula.

Outras informações

Leitura dos dados de correção alterados

Os dados de correção alterados estão disponíveis nas seguintes variáveis de sistema e variáveis BTSS:

Significado	Variável de sistema	Variável BTSS
Posição de corte	$P_AD[2]	cuttEdgeParam2
Ângulo do suporte	$P_AD[10]	cuttEdgeParam10
Sentido de corte	$P_AD[11]	cuttEdgeParam11
Ângulo livre	$P_AD[24]	cuttEdgeParam24

Por isso, os dados são sempre alterados com relação aos parâmetros de ferramenta correspondentes ($TC_DP2[..., ...], ...), se a função "Alteração dos dados de correção em ferramentas orientáveis" for ativada com o comando `CUTMOD` ou `CUTMODK` e se a ferramenta for orientada através de um porta-ferramenta orientável ou uma transformação de orientação adequada.

Outras variáveis de sistema relevantes para a função

Variável de sistema	Significado
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	Fornece o ângulo com o qual uma ferramenta no plano de usinagem ativo precisa ser orientada e no qual se baseia a determinação dos dados de corte alterados nas funções CUTMOD ou CUTMODK.
$P_CUTMOD / $AC_CUTMOD	Lê o valor válido atual que foi programado por último com o comando CUTMOD (número do porta-ferramenta para o qual é necessário ativar a alteração de dados de corte). Se o último valor programado era `CUTMOD=-2` (ativação com o porta-ferramenta orientável atual ativo), as variáveis de sistema não retornam o valor "-2", mas o número do porta-ferramenta orientável ativo neste momento da programação.
$P_CUTMODK / $AC_CUTMODK	Lê o nome da transformação sob a qual o bloco de dados válido atual foi introduzido na "Alteração dos dados de correção".

Variável de sistema	Significado	
$P_CUT_INV / $AC_CUT_INV	Retorna o valor TRUE se a ferramenta for girada de modo que o sentido de giro do fuso seja invertido. Para isso devem ser preenchidas as seguintes quatro condições no bloco relacionado à respectiva operação de leitura: 1. Se uma ferramenta de tornear ou de retificar estiver ativa (tipos de ferramenta 400 até 599 e / ou SD42950 $SC_TOOL_LENGTH_TYPE = 2). 2. A alteração dos dados de correção foi ativada com o comando `CUTMOD` ou `CUTMODK`. 3. Se um porta-ferramenta orientável ou uma transformação de orientação definida com sequências cinemáticas estiver ativa(o), o/a qual foi selecionado(a) através do comando CUTMOD ou CUTMODK. 4. A ferramenta será orientada, através do porta-ferramenta orientável ou da transformação de orientação cinemática, de modo que a normal resultante do corte da ferramenta esteja orientada em mais de 90º com relação à posição de saída (normalmente 180°). Se pelo menos uma das quatro condições mencionadas não for preenchida, a variável emite o valor FALSE. Para ferramentas cuja posição de corte não está definida, o valor da variável é sempre FALSE.	
$P_CUTMOD_ERR	Condição de erro após a última chamada da função CUTMOD A função CUTMOD também pode ser chamada implicitamente na troca de ferramenta. A variável é retornada a zero com Reset. Ela é reiniciada em cada troca de ferramenta e, se necessário, novamente gravada. A variável é codificada em bits. Os bits têm os seguintes significados:	
	Bit 0:	Para a ferramenta ativa não há nenhum sentido de corte definido.
	Bit 1:	O ângulo de corte (ângulo livre e ângulo do suporte) da ferramenta ativa são ambos nulos.
	Bit 2:	O ângulo livre da ferramenta ativa tem um valor não permitido (< 0° ou > 180°).
	Bit 3:	O ângulo do suporte da ferramenta ativa tem um valor não permitido (< 0° ou > 90°).
	Bit 4:	O ângulo da placa da ferramenta ativa tem um valor não permitido (< 0° ou > 90°).
	Bit 5:	A combinação posição de corte - ângulo do suporte da ferramenta ativa não é permitida (nos locais de corte 1 até 4, o ângulo do suporte precisa ser ≤ 90° e nos locais de corte 5 até 8 precisa ser ≥ 90°).
	Bit 6:	Orientação não permitida da ferramenta ativa. A ferramenta foi orientada em ± 90° (com uma tolerância de aprox. 1°) para fora do plano de usinagem ativo. Por isso, a posição de corte não está mais definida no plano de usinagem.
	Bit 7:	A placa de corte não está no plano de usinagem e o ângulo entre a placa de corte e o plano de usinagem ultrapassa o limite superior especificado com o dado de ajuste SD42998 $SC_CUTMOD_PLANE_TOL.
	Bit 8:	A placa de corte não está no plano de usinagem. O valor do ângulo α é maior que 1°. O ângulo α é o ângulo de orientação para que o eixo coordenado, assim como o eixo de rotação do ângulo β e também o eixo de rotação do ângulo γ, fique perpendicular (para G18 o eixo X).

$P_ ...: Variáveis de pré-processamento

$AC_...: Variáveis de processamento principal

Todas as variáveis do processamento principal podem ser lidas em ações sincronizadas. Um acesso de leitura a partir do pré-processamento gera uma parada de pré-processamento.

Mudança de planos

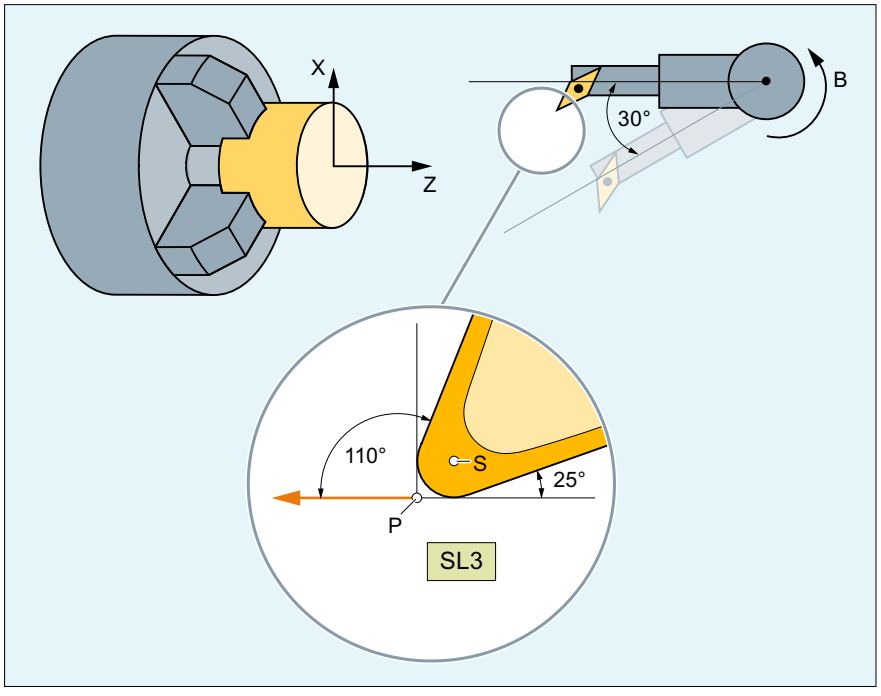
Para a definição da posição de corte, sentido de corte e ângulo de suporte ou ângulo livre modificados é determinante se considerar o corte no respectivo plano ativado (G17 - G19).

Entretanto, se o dado de ajuste SD42940 $SC_TOOL_LENGTH_CONST (troca dos componentes de comprimento da ferramenta na mudança de planos) contém um valor válido diferente de zero (positivo ou negativo 17, 18 ou 19), então este conteúdo determinará o plano que as grandezas relevantes devem ser consideradas.

A regra de preferência dos dados de ajuste através do código G, pode ser facilmente estabelecida pela definição do bit 18 dos dados da máquina $MC_TOOL_PARAMETER_DEF_MASK. Ou seja, se esse bit estiver definido, valerá sempre o plano definido com o comando G do grupo 6.

Ação dos dados de corte modificados

A posição de corte modificada e o ponto de referência do corte são ativados imediatamente na programação também para uma ferramenta ativa. Uma nova seleção de ferramenta neste caso não é necessária.



S: Centro do corte

P: Ponto de referência do corte

SL: Posição de corte

Esquema 11-4 Ferramentas com posição de corte 3 e um porta-ferramenta orientável que pode orientar a ferramenta em torno do eixo B

```
Código de programa                                          Comentário
N10 $TC_DP1[1,1]=500
N20 $TC_DP2[1,1]=3                                          ; posição de corte
N30 $TC_DP3[1,1]=12
N40 $TC_DP4[1,1]=1
N50 $TC_DP6[1,1]=6
N60 $TC_DP10[1,1]=110                                       ; Ângulo do suporte
N70 $TC_DP11[1,1]=3                                         ; Sentido de corte
N80 $TC_DP24[1,1]=25                                        ; Ângulo livre

N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0            ; Eixo B
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0                        ; Eixo C
$TC_CARR12[2]=1
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
```

Os valores numéricos nos comentários indicam as posições finais do bloco em coordenadas de máquina (MCS) na sequência X → Y → Z.

Explicações

No bloco `N180` se seleciona primeiro a ferramenta com `CUTMOD=0` e não o porta-ferramenta orientável girado. Visto que todos vetores de Offset do porta-ferramenta orientável têm valor

0, se aproxima a posição que estiver especificada no `$TC_DP3[1,1]` e `$TC_DP4[1,1]` que corresponde aos comprimentos de ferramenta.

No bloco `N200` é ativado o porta-ferramenta orientável com uma rotação de 30° em torno do eixo B. Visto que a posição do corte não é alterada devido o `CUTMOD=0`, prevalece o ponto de referência antigo dos cortes. Por isso que no bloco `N210` é aproximada a posição que contém o antigo ponto de referência de corte no ponto zero (isto é, o vetor (1, 12) é girado 30° no plano Z/X).

No bloco `N260` atua uma diferença em relação ao bloco `N200 CUTMOD=2`. Por causa da rotação do porta-ferramenta orientável, a posição de corte modificada passa a ser 8. Disto também resultam posições de eixo divergentes.

Nos blocos `N220` e `N270` sempre se ativa a correção do raio da ferramenta (WRK). A posição de corte diferente nos dois segmentos do programa não tem nenhuma influência sobre as posições finais dos blocos ativos na WRK, sendo que as posições correspondentes são idênticas. Somente nos blocos de desativação `N260` e `N300` que atuam novamente as posições de corte diferentes.

11.12 Trabalhar com ambientes de ferramenta

Visão geral das funções

- Armazenar o ambiente de ferramenta (TOOLENV) (Página 496)
- Eliminar um ambiente de ferramenta (DELTOOLENV) (Página 499)
- Leitura de números T, D e DL (GETTENV) (Página 500)
- Leitura de comprimentos de ferramenta ou componentes de comprimento de ferramenta (GETTCOR) (Página 502)
- Modificar componentes de ferramenta (SETTCOR) (Página 508)

Visão geral das variáveis de sistema

- Leitura de informações sobre ambientes de ferramenta armazenadas ($P_TOOLENVN, ($P_TOOLENV) (Página 501)

11.12.1 Armazenar o ambiente de ferramenta (TOOLENV)

A função TOOLENV serve para armazenar os estados atuais que são importantes para a avaliação dos dados de ferramentas armazenados na memória

Em detalhes, estes são os seguintes dados:

- O comando G ativo do grupo:
  - 6 (G17, G18, G19)
  - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- O eixo transversal ativo
- Os dados de máquina:
  - MD18112 $MN_MM_KIND_OF_SUMCORR (propriedades das correções de somas na área TO)
  - MD20360 $MC_TOOL_PARAMETER_DEF_MASK (definição dos parâmetros da ferramenta)

- Os dados de ajuste:
  - SD42900 $SC_MIRROR_TOOL_LENGTH (troca de sinal comprimento de ferramenta ao espelhar)
  - SD42910 $SC_MIRROR_TOOL_WEAR (troca de sinal desgaste de ferramenta ao espelhar)
  - SD42920 $SC_WEAR_SIGN_CUTPOS (sinal do desgaste em ferramentas com dispositivos para arestas cortantes)
  - SD42930 $SC_WEAR_SIGN (sinal do desgaste)
  - SD42935 $SC_WEAR_TRANSFORM (transformações para componentes de ferramenta)
  - SD42940 $SC_LENGTH_CONST (troca dos componentes de comprimento da ferramenta na troca de plano)
  - SD42942 $SC_TOOL_LENGTH_CONST_T (troca dos componentes de comprimento da ferramenta para ferramentas de tornear, em caso de troca de planos)
  - SD42950 $SC_TOOL_LENGTH_TYPE (atribuição dos componentes de comprimento de ferramenta independentemente do tipo de ferramenta)
  - SD42954 $SC_TOOL_ORI_CONST_M (troca dos componentes de orientação da ferramenta para fresas, em caso de troca de planos)
  - SD42956 $SC_TOOL_ORI_CONST_T (troca do componentes de orientação da ferramenta para ferramentas de tornear, em caso de troca de planos)
- A parcela de orientação do frame total atual (rotações e espelhamentos, sem deslocamentos de ponto zero ou escalamentos)
- A parcela de orientação e o comprimento resultante do porta-ferramenta orientável ativo
- A parcela de orientação e o comprimento resultante de uma transformação ativa

Além dos dados mencionados que descrevem o ambiente da ferramenta, também são armazenados o número T, o número D e o número DL da ferramenta ativa, para poder acessar esta ferramenta posteriormente no mesmo ambiente que ao acessar TOOLENV, sem necessidade de designar a ferramenta novamente.

Sintaxe

```
<Status> = TOOLENV(<Name>)
```

Significado

`TOOLENV(...)`:	Função predefinida para armazenar um ambiente de ferramenta	
	Sozinho no bloco:	sim

<Status>:	Valor de devolução da função. Valores negativos mostram condições de erro.		
	Tipo de dado:		INT
	Valor:	0	Função OK
		-1	Nenhum espaço de memória reservado para ambientes de ferramenta: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Significa que a funcionalidade "Ambientes de Ferramenta" não está presente.
		-2	Não há mais espaços de memória livres para ambientes de ferramenta.
		-3	String zero não é admissível como nome de um ambiente de ferramenta.
		-4	Nenhum parâmetro (<nome>) indicado.
Parâmetros			
1 <Nome>:	Nome, pelo qual o bloco de dados atual deve ser arquivado Se já existe um bloco de dados com o mesmo nome, este é sobrescrito. Neste caso, o status é "0".		
	Tipo de dado:		STRING

Outras informações

Medida básica/medida do adaptador - correção de comprimento de ferramenta

Se o comprimento do adaptador ou a medida básica (parâmetros específicos de aresta cortante $TC_DP21, $TC_DP22 e $TC_DP23) entram ou não no cálculo de comprimento de ferramenta, é decidido, com a administração ativa de ferramentas (apenas disponível com a opção "Administração de Ferramentas"!), pelo valor do dado de máquina:

MD18104 $MN_MM_NUM_TOOL_ADAPTER (adaptador da ferramenta na área TO)

Uma vez que uma alteração deste dado de máquina somente pode entrar em vigor na inicialização do sistema de comando, este dado não é armazenado no ambiente de ferramenta.

Comprimento resultante de porta-ferramentas orientáveis e transformações

Indicação

Tanto nos porta-ferramentas orientáveis quanto nas transformações existem variáveis de sistema ou dados de máquina que atuam como componentes adicionais de comprimento de ferramenta, e que são, totalmente ou parcialmente, subordinadas às rotações realizáveis por eles. As parcelas adicionais de comprimento de ferramenta, resultantes disto, devem ser arquivadas também na chamada de TOOLENV, uma vez que formam uma parte do ambiente no qual a ferramenta é utilizada.

Transformação de adaptador

A transformação de adaptador é uma propriedade do adaptador de ferramenta e, portanto, da ferramenta global. Por este motivo, ela não é parte de um ambiente de ferramenta que pode ser aplicado a uma outra ferramenta.

Pelo armazenamento de todos os dados necessários para a determinação do comprimento total de ferramenta existe a possibilidade de calcular posteriormente o comprimento efetivo de uma ferramenta, mesmo se esta não estiver mais ativa nesse momento ou se houve mudanças nas condições do ambiente (por ex. dos comandos G ou dos dados de ajuste). Também, o comprimento efetivo de uma outra ferramenta pode ser calculado com a suposição que seria utilizada sob as mesmas condições que a ferramenta para a qual o status foi armazenado.

Quantidade máxima de blocos de dados para ambientes de ferramenta

Através do dado de máquina MD18116 $MN_MM_NUM_TOOL_ENV é determinado o número máximo de blocos de dados para descrição de ambientes de ferramenta que podem ser armazenados. Os dados estão na área TOA. Eles são conservados mesmo ao desligar o sistema de comando.

Uma garantia de dados não é possível. Isto significa que estes dados não podem ser transferidos entre diferentes sistemas de comando.

11.12.2 Eliminar um ambiente de ferramenta (DELTOOLENV)

Com a função DELTOOLENV é possível eliminar blocos de dados para a descrição de ambientes de ferramenta. Eliminar significa que não é mais possível acessar um bloco de dados armazenado sob determinado nome (a tentativa de acesso dispara um alarme).

Indicação

Blocos de dados somente podem ser eliminados com a função DELTOOLENV, através de um INITIAL.INI-download ou uma partida a frio (inicialização NC com dados padrão de máquina). Não existem outros processos automáticos de eliminação.

Sintaxe

```
<Status> = DELTOOLENV(<Name>)
<Status> = DELTOOLENV()
```

Significado

DELTOOLENV(...):	Função predefinida para eliminar um ambiente de ferramenta	
	Sozinho no bloco:	sim

<Status>:	Valor de devolução da função. Valores negativos mostram condições de erro.		
	Tipo de dado:		INT
	Valor:	0	Função OK
		-1	Nenhum espaço de memória reservado para ambientes de ferramenta: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Significa que a funcionalidade "Ambientes de Ferramenta" não está presente.
		-2	Um ambiente de ferramenta com o nome indicado não existe.
Parâmetros			
1 <Nome>:	Nome do bloco de dados a ser eliminado		
	Tipo de dado:		STRING
DELTOOLENV():	DELTOOLENV() sem indicação de um nome elimina todos os blocos de dados para descrição de ambientes de ferramenta		

11.12.3 Leitura de números T, D e DL (GETTENV)

A função GETTENV serve para a leitura de números T, D e DL armazenados em um ambiente de ferramenta.

Sintaxe

```
<Status> = GETTENV(<Name>, <TDDL>)
```

Significado

GETTENV(...):	Função predefinida para leitura de números T, D e DL em um bloco de dados para descrição de um ambiente de ferramenta		
	Sozinho no bloco:		sim
<Status>:	Valor de devolução da função. Valores negativos mostram condições de erro.		
	Tipo de dado:		INT
	Valor:	0	Função OK
		-1	Nenhum espaço de memória reservado para ambientes de ferramenta: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Significa que a funcionalidade "Ambientes de Ferramenta" não está presente.
		-2	Um ambiente de ferramenta com o nome indicado não existe.
Parâmetros			
1 <Nome>:	Nome do bloco de dados, a partir do qual os números T, D e DL devem ser lidos		
	Tipo de dado:		STRING

2	<TDDL>:	O campo deste parâmetro de resultado contém os números T, D, DL da ferramenta cujo ambiente de ferramenta se encontra armazenado no bloco de dados indicado. • <TDDL> [0]: Número T • <TDDL> [1]: Número D • <TDDL> [2]: Número DL	
		Tipo de dado:	INT[3]
GETTENV(,<TDDL>), GETTENV("",<TDDL>):		Na chamada da função GETTENV, é admissível omitir o primeiro parâmetro ou entrar como primeiro parâmetro com o string zero. Nestes dois casos especiais, são devolvidos em <TDDL> os números T, D e DL da ferramenta **ativa**.	

11.12.4 Leitura de informações sobre ambientes de ferramenta armazenadas ($P_TOOLENVN, ($P_TOOLENV)

Informações sobre ambientes de ferramenta armazenados podem ser lidas através das seguintes variáveis de sistema:

$P_TOOLENVN:	Fornece a quantidade de blocos de dados definidos por meio de TOOLENV (e ainda não eliminados) para descrição de ambientes de ferramenta			
	Sintaxe:	<n> = $P_TOOLENVN		
	Significado:	<n>:	Quantidade de blocos de dados definidos	
			Tipo de dado:	INT
			Faixa de valores:	0 ... MD18116 $MN_MM_NUM_TOOL_ENV
	O acesso a esta variável de sistema também á admissível quando não são possíveis os ambientes de ferramenta (MD18116 = 0). Neste caso, o valor de devolução é "0".			
$P_TOOLENV:	Fornece o nome do bloco de dados <i> para descrição de um ambiente de ferramenta			
	Sintaxe:	<Nome> = $P_TOOLENV[<i>]		
	Significado:	<Nome>:	Nome do bloco de dados com o número <i>	
			Tipo de dado:	STRING
		<i>:	Número do bloco de dados	
			Tipo de dado:	INT
			Faixa de valores:	1 ... $P_TOOLENVN
	A atribuição dos números aos blocos de dados não e fixa, ela pode ser alterada por eliminação ou criação de novos blocos de dados. A numeração dos blocos de dados ocorre internamente. Se <i> indica um bloco de dados não definido, o string zero é devolvido. Se o índice <i> está inválido, significa <i> abaixo de 1 ou acima da quantidade máxima de blocos de dados para ambientes de ferramenta (MD18116 $MN_MM_NUM_TOOLENV), o seguinte alarme é acionado: Alarme 17020 "Índice Array 1 não permitido"			

11.12.5 Leitura de comprimentos de ferramenta ou componentes de comprimento de ferramenta (GETTCOR)

A função GETTCOR serve para a leitura de comprimentos de ferramenta ou componentes de comprimento de ferramenta.

Neste processo, é possível indicar por meio de parametrização, quais dos componentes devem ser considerados e sob que condições de utilização a ferramenta deve ser avaliada.

Sintaxe

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

Significado

`GETTCOR(...)`:	Função predefinida para a leitura de comprimentos de ferramenta ou componentes de comprimento de ferramenta.		
	Sozinho no bloco:		sim
`<Status>`:	Valor de devolução da função. Valores negativos mostram condições de erro.		
	Tipo de dado:		INT
	Valor:	0	Função OK
		-1	Nenhum espaço de memória reservado para ambientes de ferramenta: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Significa que a funcionalidade "Ambientes de Ferramenta" não está presente.
		-2	Um ambiente de ferramenta com o nome indicado em <Stat> não existe.
		-3	String inválido no parâmetro <Comp>. Causa deste erro podem ser caracteres inválidos ou programados em duplicidade.
		-4	Número T inválido
		-5	Número D inválido
		-6	Número DL inválido
		-7	Tentativa de acesso a um módulo de memória inexistente.
		-8	Tentativa de acesso a uma opção inexistente (por ex. orientação programável de ferramenta, administração de ferramentas).
		-9	O string <Comp> contém dois pontos (identificação para especificação de um sistema de coordenadas), porém não seguidos por um caractere válido de designação do sistema de coordenadas solicitado.
Parâmetros			

1	<Len>:	Vetor de resultado:
		Tipo de dado: REAL[11]
		Os componentes de vetor estão dispostos na seguinte ordem: • <Len> [0]: Tipo de ferramenta • <Len> [1]: Posição de corte • <Len> [2]: Abscissa • <Len> [3]: Ordenada • <Len> [4]: Aplicada • <Len> [5]: Raio da ferramenta Válido como sistema de coordenadas de referência para componentes de comprimento é o sistema de coordenadas definido em <Comp> e <Stat>. Na ausência de um sistema de coordenadas definido em <Comp>, os comprimentos de ferramenta são representados no sistema de coordenadas da máquina. A atribuição de abcissa, ordenada e aplicada aos eixos geométricos depende do plano ativo no ambiente de ferramenta utilizado. Significa que em G17 a abcissa está paralela com X, em G18 com Z, etc. Os componentes <Len>[6] até <Len>[10] contêm os parâmetros adicionais, que podem ser indicados para a descrição da geometria de uma ferramenta (por ex. $TC_DP7 até $TC_DP11 para a geometria ou os correspondentes componentes para desgaste ou correções de somas e de ajuste). Estes 5 elementos adicionais e o raio da ferramenta foram definidos somente para os componentes E, G, S e W. A avaliação deles não depende de <Stat>. Por este motivo, os correspondentes valores em <Len>[6] até <Len>[10] somente podem ser diferentes de zero, quando pelo menos um dos quatro componentes mencionados participa do cálculo de comprimento de ferramenta. Os demais componentes não têm influência sobre o resultado. As indicações de medidas referem-se ao sistema básico do sistema de comando (polegada ou métrico).

2	<Comp>:	Componentes de comprimento de ferramenta (opcional)			
		Tipo de dado:	STRING		
		A sequência de caracteres consiste em dois strings parciais, separados por dois pontos. Forma geral: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	O **primeiro string parcial** identifica os componentes de comprimento de ferramenta, que devem ser considerados no cálculo de comprimento de ferramenta. A ordem dos caracteres no string parcial assim como seu modo de escrever (maiúscula ou minúscula) é de livre escolha. Entre os caracteres podem ser inseridos espaços ou sinais de tabulação (white spaces) em número de livre escolha. **Aviso:** Os caracteres no string parcial **não** devem ser programados em duplicidade!		
			Caractere:	–	Sinal de menos (admissível somente como primeiro caractere!) É calculado o comprimento total de ferramenta menos os componentes especificados no string a seguir.
				C	Medida de adaptador ou medida de base (aquele dos dois componentes existentes de modo alternativo, que está ativo para a ferramenta utilizada)
				E	Correções de ajuste
				G	Geometria
				K	Transformação cinemática (é avaliada somente na transformação genérica de 3, 4 e 5 eixos!)
				S	Correções de somas
				T	Porta-ferramenta orientável
				W	Desgaste
			O primeiro string parcial estando vazio (exceto white spaces) significa que o comprimento total de ferramenta deve ser calculado, levando em consideração todos os componentes. O mesmo vale também quando o parâmetro <Comp> não é indicado.		
		<Substr_2>:	O **segundo string parcial** opcional designa o sistema de coordenadas, no qual o comprimento de ferramenta deve ser emitido. O segundo string parcial consiste apenas em um único caractere relevante.		
			Caractere:	A	Sistema ajustável de coordenadas (ACS)
				B	Sistema de coordenadas básico (BCS)
				K	Sistema de coordenadas da ferramenta da transformação cinemática (KCS)
				M	Sistema de coordenadas da máquina (MCS)
				T	Sistema de coordenadas da ferramenta (TCS)
				W	Sistema de coordenadas da peça (WCS)

			Sem indicação do sistema de coordenadas, a avaliação ocorre no sistema de coordenadas da máquina (MCS). As rotações que devem ser consideradas, quando necessário, são determinadas pelo ambiente de ferramenta definido em <Stat>.
3	<_Stat>:	Nome do bloco de dados para descrição de um ambiente de ferramenta (opcional)	
		Tipo de dado:	STRING
		Se o valor deste parâmetro é o string zero ("") ou se ele não é indicado, o estado atual é utilizado. Sem especificação de nenhuma ferramenta, a ferramenta atual é utilizada.	
4	<T>:	Número T interno da ferramenta (opcional)	
		Tipo de dado:	INT
		Sem indicação deste parâmetro ou se o valor dele é "0", a ferramenta arquivada em <Stat> é utilizada. Se o valor deste parâmetro é "-1", o número T da ferramenta ativa é utilizado. Também é admissível indicar de modo explícito o número da ferramenta ativa. **Aviso:** Sem indicação de <Stat>, o estado atual é utilizado como ambiente de ferramenta. Uma vez que, através de <T> = 0, existe a referência ao número T armazenado no ambiente de ferramenta, nele utiliza-se a ferramenta ativa, o que significa que, neste caso especial, as indicações <T> = 0 e <T> = -1 têm o mesmo significado.	
5	<D>:	Número da aresta cortante de ferramenta (opcional)	
		Tipo de dado:	INT
		Sem indicação deste parâmetro ou se o seu valor é "0", o número D utilizado varia em função da fonte do número T. Se o número T a partir do ambiente de ferramenta é utilizado, também o número D do ambiente de ferramenta é lido, caso contrário, o número da ferramenta atualmente ativa.	
6	<DL>:	Número da correção em função do local (opcional)	
		Tipo de dado:	INT
		Sem indicação deste parâmetro, o número DL utilizado varia em função da fonte do número T. Se o número T a partir do ambiente de ferramenta é utilizado, também o número D do ambiente de ferramenta é lido, caso contrário, o número da ferramenta atualmente ativa.	

Exemplos

`GETTCOR(_LEN)`	Calcula o comprimento de ferramenta atualmente ativa no sistema de coordenadas da máquina, levando em consideração todos os componentes.
`GETTCOR(_LEN,"CGW:W")`	É calculado o comprimento de ferramenta que consiste em medida de adaptador ou medida básica, geometria e desgaste para a ferramenta ativa. Outras partes como por ex. porta-ferramenta orientável ou transformações cinemáticas não são consideradas. A emissão ocorre no sistema de coordenadas da peça de trabalho.

`GETTCOR (_LEN,"-K:B")`	É calculado o comprimento total de ferramenta ativa sem considerar componentes de comprimento de uma transformação cinemática eventualmente ativa. Emissão no sistema básico de coordenadas.
`GETTCOR (_LEN,":M","Testenv1",,3)`	É calculado o comprimento total de ferramenta para a ferramenta armazenada no ambiente de ferramenta com o nome "Testenv1" no sistema de coordenadas da máquina. O cálculo, porém, ocorre, de modo independente do número de aresta cortante arquivado, para a número de aresta cortante D3.

Outras informações

Transformação de adaptador / porta-ferramentas orientáveis / transformação cinemática

As rotações e trocas de componentes que, quando necessário, são executadas pela transformação de adaptador, pelo porta-ferramenta orientável e por uma transformação cinemática, são componentes do ambiente de ferramenta. Por este motivo, elas são executadas sempre, mesmo quando não se deve considerar o respectivo componente de comprimento. Se isto não é desejado, é necessário definir ambientes de ferramenta, nos quais as correspondentes transformações não estejam ativas. Em muitos casos, (sempre quando, em uma máquina, não são utilizados transformações ou porta-ferramentas orientáveis), os blocos de dados armazenados para ambientes de ferramenta cumprem estas condições automaticamente sem requerer atenção especial do usuário.

Ferramentas de tornear e retificar: Cálculo de comprimento de ferramenta em dependência de MD20360 $MC_TOOL_PARAMETER_DEF_MASK

Para ferramentas de tornear e retificar, o modo de avaliar o desgaste ou o comprimento de ferramenta em um eixo de diâmetro eventualmente existente é determinado no dado de máquina:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK (definição dos parâmetros da ferramenta)

Bit	Valor	
0	Para as ferramentas de tornear e de retificar, o **parâmetro de desgaste** do eixo transversal entra no cálculo como valor de diâmetro:	
	= 0 (Default)	não
	= 1	sim
1	Para as ferramentas de tornear e de retificar, o **componente de comprimento de ferramenta** do eixo transversal entra no cálculo como valor de diâmetro:	
	= 0 (Default)	não
	= 1	sim

Estando ajustados os respectivos bits, o registro pertencente a eles é avaliado com o fator 0,5. Esta avaliação também influencia o comprimento de ferramenta fornecido por GETTCOR.

Exemplo:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 $MC_DIAMETER_AX_DEF (eixo geométrico com função de eixo transversal) = "X"

X é o eixo de diâmetro (configuração padrão para máquina de tornear)

```
Código de programa                      Comentário
N30 $TC_DP1[1,1]=500
N40 $TC_DP2[1,1]=2
N50 $TC_DP3[1,1]=3.0                    ; Geometria L1
N60 $TC_DP4[1,1]=4.0
N70 $TC_DP5[1,1]=5.0
N80 $TC_DP12[1,1]=12.0                  ; Desgaste 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
```

Componentes de comprimento de transformação cinemática e porta-ferramenta orientável

Quando no cálculo do comprimento da ferramenta é considerado um **porta-ferramenta orientável**, os seguintes vetores entram no cálculo de comprimento de ferramenta:

Tipo	Vetores
M	l1 e l2
T	l1 l2 e l3
P	O comprimento da ferramenta não é influenciado pelo porta-ferramenta orientável.

Na **transformação genérica de 5 eixos**, com os tipos de transformação 24 e 56, os seguintes dados de máquina entram no cálculo de comprimento de ferramenta:

Tipo de transformação	Dados de máquina
24	MD24550/24650 $MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 MD24558/24658 $MC_TRAFO5_PART_OFFSET_1/2
56	MD24550/24650 $MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2

O tipo de transformação 56 (ferramenta móvel e peça de trabalho móvel) corresponde ao tipo M do porta-ferramenta orientável.

Nesta transformação de 5 eixos, nas versões de software até agora, o vetor MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 (vetor do deslocamento cinemático da 1ª/2ª transformação de 5 eixos no canal) corresponde à soma dos dois vetores l_1 e l_3 no porta-ferramenta orientável do tipo M.

Em ambos os casos, apenas a soma é relevante para a transformação. O tipo de composição a partir dos dois componentes individuais é insignificante. Importantes, porém, no cálculo de comprimento de ferramenta são as respectivas parcelas que devem ser atribuídas à ferramenta e à mesa da ferramenta. Por este motivo foi implantado o dado de máquina

MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 (vetor do deslocamento cinemático na mesa). Corresponde ao vetor l3. O dado de máquina MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 agora não mais corresponde à soma de l1 e l3, mas apenas ao vetor l1. Com o dado de máquina MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 igual zero, o comportamento é igual ao comportamento até agora.

Compatibilidade

A função GETTCOR, em conjunto com as funções TOOLENV e SETTCOR, tem a finalidade, entre outras, de substituir partes da funcionalidade, que até agora era realizada nos ciclos externos de medição.

Nos ciclos de medição, apenas uma parte dos parâmetros, que em última instância determinam o comprimento efetivo de ferramenta, era avaliada. As funções mencionadas podem ser parametrizadas de modo que o comportamento dos ciclos de medição relativos ao cálculo de comprimento de ferramenta possa ser reproduzido.

11.12.6 Modificar componentes de ferramenta (SETTCOR)

A função SETTCOR serve para alterar componentes de ferramenta sob consideração de todas condições marginais que podem entrar na avaliação dos diversos componentes.

Indicação

Sobre a terminologia: Quando a seguir, no contexto de comprimento de ferramenta, se fala de componentes de ferramenta, faz-se a referência aos componentes que, contemplados de modo vetorial, compõem o comprimento total de ferramenta (por ex. geometria ou desgaste). Componentes deste tipo, portanto, consistem em três valores individuais (L1, L2, L3) os quais, a continuação, são denominados valores de coordenada.

"Geometria", como componente de ferramenta, portanto, consiste nos três valores de coordenada $TC_DP3 até $TC_DP5.

Sintaxe

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>, <GeoAx>,
<Stat>, <T>, <D>, <DL>])
```

Significado

`SETTCOR(...)`:	Função predefinida para alterar componentes de ferramenta.	
	Sozinho no bloco:	sim

<Status>:	Valor de devolução da função. Valores negativos mostram condições de erro.		
	Tipo de dado:		INT
	Valor:	0	Função OK
		-1	Nenhum espaço de memória reservado para ambientes de ferramenta: MD18116 $MN_MM_NUM_TOOL_ENV = 0 Significa que a funcionalidade "Ambientes de Ferramenta" não está presente.
		-2	Um ambiente de ferramenta com o nome indicado em <Stat> não existe.
		-3	String inválido no parâmetro <Comp>. Causa deste erro podem ser caracteres inválidos ou programados em duplicidade.
		-4	Número T inválido.
		-5	Número D inválido.
		-6	Número DL inválido.
		-7	Tentativa de acesso a um módulo de memória inexistente.
		-8	Tentativa de acesso a uma opção inexistente (por ex. orientação programável de ferramenta, administração de ferramentas).
		-9	Valor numérico inadmissível para o parâmetro <CorComp>.
		-10	Valor numérico inadmissível para o parâmetro <CorMode>.
		-11	Os conteúdos dos parâmetros <Comp> e <CorComp> são contraditórios.
		-12	Os conteúdos dos parâmetros <Comp> e <CorMode> são contraditórios.
		-13	O conteúdo do parâmetro <GeoAx> não se refere a um eixo geométrico
		-14	Tentativa de gravar em correção de ajuste inexistente.
Parâmetros			
1	<CorVal>:	Vetor de correção No sistema de coordenadas da peça de trabalho (WCS) definido por <Stat> vale a seguinte atribuição: • <CorVal> [0]: Abscissa • <CorVal> [1]: Ordenada • <CorVal> [2]: Aplicada Se apenas um componente de ferramenta deve ser corrigido (ou seja nenhuma correção vetorial, veja parâmetro <CorMode>), o valor da correção sempre consta em <CorVal>[0], independente do eixo, sobre o qual atua. Os conteúdos dos dois componentes restantes não são avaliados neste caso. Quando <CorVal> ou um componente de <CorVal> se refere ao eixo transversal, o dado é avaliado como **medida de raio**. Isto significa que uma ferramenta fica por ex. "mais comprida" na medida indicada, o que leva a uma alteração do dobro do tamanho no diâmetro da peça de trabalho. As indicações das medidas se referem ao **sistema básico** (polegada ou métrico) do sistema de comando.	
		Tipo de dado:	REAL[3]

2	<Comp>:	Componente(s) de ferramenta			
		Tipo de dado:	STRING		
		A sequência de caracteres consiste em dois strings parciais, separados por dois pontos. Forma geral: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	O **primeiro string parcial** sempre deve estar presente e pode consistir em um ou dois caracteres. Estão presentes primeiro ou o único dígito para o 1º componente (val_1) e o segundo dígito para o 2º componente (Val_2), os quais devem ser processados conforme os seguintes parâmetros <CorComp> e <CorMode> .		
			Caracte-re:	C	Medida de adaptador ou medida de base (aquele dos dois componentes existentes de modo alternativo, que está ativo para a ferramenta utilizada)
				E	Correções de ajuste
				G	Geometria
				S	Correções de somas
				W	Desgaste
		<Substr_2>:	O **segundo string parcial** é opcional. Pode, de modo alternativo, consistir na letra (única) "W" ou "T".		
			Caracte-re:	W	Quando o segundo string parcial está vazio ou quando contém a letra "W", os valores de correção são calculados como se tivessem sido medidos no sistema de coordenadas da **peça**de trabalho (WCS).
				T	Quando o segundo string parcial contém a letra "T", os valores de correção são calculados como se tivessem sido medidos no sistema de coordenadas da **ferramenta** (**T**ool Coordinate System, TCS).
		O modo de escrever os caracteres no string (letras maiúsculas ou minúsculas) é de livre escolha. Entre os caracteres podem ser inseridos espaços ou sinais de tabulação (white spaces) em número de livre escolha.			

3	<CorComp>:	Especifica o(s) componente(s) do bloco de dados da ferramenta que deve(m) ser descrito(s) (opcional)		
		Tipo de dado:		INT
		Valor:	0	O valor de correção <CorVal>[0] refere-se ao eixo geométrico entregue no parâmetro <GeoAx> no sistema de coordenadas da peça de trabalho ou no sistema de coordenadas da ferramenta (veja neste contexto a descrição do parâmetro <Comp>). Significa que o valor de correção deve entrar no cálculo dos componentes designados de ferramenta de modo que resulte, levando em consideração todos os parâmetros que podem influenciar o cálculo de comprimento de ferramenta, em uma alteração pelo valor prescrito do comprimento total de ferramenta, no sentido dado do eixo. Esta alteração deve ser realizada através da correção do componente indicado em <Comp> e através da instrução simbólica de cálculo indicada em <CorMode> (veja o parâmetro a seguir). A correção resultante, portanto, pode ter efeito em todos os três componentes de eixo
			1	Como "0", porém vetorial. O conteúdo do vetor <CorVal> refere-se a abcissa, ordenada e aplicada no sistema de coordenadas da peça de trabalho ou no sistema de coordenadas da ferramenta (veja neste contexto a descrição do parâmetro <Comp>). O parâmetro subsequente <GeoAx> não é avaliado.
			2	Correção vetorial, significa que L1, L2 e L3 podem ser alterados simultaneamente. Contrastando com as variantes "0" e "1", os valores de correção em <CorVal> referem-se às coordenadas do componente Val_1 (veja o parâmetro a seguir <CorMode>) da ferramenta. Uma possível posição inclinada da ferramenta em relação ao sistema de coordenadas da peça de trabalho não afeta a correção.
			3 - 5	Correção de comprimento de ferramenta L1 até L3 ($TC_DP3 até $TC_DP5) ou dos correspondentes valores para desgaste, correções de ajuste ou de somas. O valor da correção encontra-se em <CorVal>[0]. Foi medido nas coordenadas do componente Val_1 (veja o parâmetro a seguir <CorMode>) da ferramenta. Uma possível posição inclinada da ferramenta em relação ao sistema de coordenadas da peça de trabalho não afeta a correção.
			6	Correção de raio de ferramenta ($TC_DP6) ou dos correspondentes valores para desgaste, correções de ajuste ou de somas. Os bits 10 e 11 (avaliação da indicação de diâmetro ou desgaste do diâmetro opcionalmente como indicação de raio ou de diâmetro) no dado de máquina MD20360 $MC_TOOL_PARAMETER_DEF_MASK são levados em consideração.
			7 – 11	Correção de $TC_DP7 até $TC_DP11 ou dos correspondentes valores para desgaste, correções de ajuste ou de somas. Estes parâmetros são tratados como o raio da ferramenta
		Sem indicação deste parâmetro, seu valor é "0".		

<table>
<tr><td rowspan="8">4</td><td rowspan="8"><CorMode>:</td><td colspan="3">Especifica o tipo da operação de gravação a ser executada (opcional)</td></tr>
<tr><td colspan="2">Tipo de dado:</td><td>INT</td></tr>
<tr><td rowspan="4">Valor:</td><td>0</td><td>Val_{1novo} = <CorVal></td></tr>
<tr><td>1</td><td>Val_{1novo} = $Val_{1antigo}$ + <CorVal></td></tr>
<tr><td>2</td><td>Val_{1novo} = <CorVal>
Val_{2novo} = 0</td></tr>
<tr><td>3</td><td>Val_{1novo} = $Val_{1antigo}$ + $Val_{2antigo}$ + <CorVal>
Val_{2novo} = 0</td></tr>
<tr><td colspan="3">O modo de escrever $Val_{1antigo}$ + $Val_{2antigo}$ deve ser entendido de maneira simbólica. Se os dois componentes (por causa do estado de <_Stat>) são avaliados de modo diferente, ou seja se entre os dois componentes atua uma rotação, então $Val_{2antigo}$ é transformado, antes da adição, de modo que o comprimento de ferramenta resultante permaneça inalterado após a eliminação de Val_{2novo} e antes da adição de <CorVal>.
<CorVal> refere-se sempre a Val_1. <CorVal>é um valor que, em função da segunda parte do parâmetro <Comp>, é medido no sistema de coordenadas da peça de trabalho (WCS) ou no sistema de coordenadas (TCS). Ele, dado o caso, já pode estar transformado em relação ao componente de ferramenta no qual deve ser calculado. Por este motivo, ele não pode ser calculado diretamente com o valor armazenado, ele precisa ser retransformado antes da adição a Val_1 ou Val_2. Isto pode ter como consequência que a correção tenha efeito sobre um eixo diferente daquele definido através de <CorComp> ou que atue sobre vários eixos.
Para o caso <CorComp> = 0, ou seja quando <CorVal> não contém nenhum vetor, mas sim um valor individual, as operações descritas são executadas nas coordenadas nas quais <CorVal> foi medido (WCS / TCS). Isto vale particularmente para a colocação em zero de Val_{2novo} nas variantes 2 e 3. Este resultado, então, é retransformado para as coordenadas da ferramenta. Isto pode ter como consequência que nenhum dos valores das coordenadas (L1, L2, L3) que devem ser colocados em zero realmente se torne zero ou que os valores das coordenadas, anteriormente zero, deixem de ser zero. Se, porém, as correspondentes operações são executadas sucessivamente para todos os três eixos geométricos, fica assegurado que também os três valores das coordenadas para o componente a ser eliminado fiquem com valor zero. Se a ferramenta não foi girada em relação ao sistema de coordenadas da peça de trabalho, ou se foi girada de uma maneira, que todos os componentes da ferramenta permaneceram em posição paralela em relação aos eixos de coordenada (transposição de eixos), fica assegurado que somente uma das respectivas coordenadas da ferramenta se altere.
A execução sucessiva da mesma operação (<CorMode>) com <CorComp> = 0 para todos os três eixos das coordenadas em ordem aleatória é idêntica a uma única execução da mesma operação com <CorComp>=1.
Para os valores de parâmetro "0" e "1", o parâmetro <Comp> deve conter um caractere e para os valores de parâmetro "2" e "3" dois caracteres.
Exemplo:
<Comp> contém o string "ES", <CorMode> o valor "2"
⇒ Correção de ajuste$_{novo}$ = <CorVal>, correção de somas$_{novo}$ = 0
Sem indicação do parâmetro <CorMode>, seu valor é "0".</td></tr>
</table>

5	<GeoAx>:	Indica o índice do eixo geométrico, no qual o valor de correção <CorVal>[0] foi medido (opcional)	
		Tipo de dado:	INT
		Faixa de valores:	0 ... 2
		Os índices 0 até 2 referem-se a abcissa, ordenada, e aplicada no plano efetivo (G17/G18/G19) do ambiente de ferramenta atual. O conteúdo deste parâmetro somente é avaliado quando o parâmetro <CorComp> tem valor "0".	
6	<Stat>:	Nome do bloco de dados para descrição de um ambiente de ferramenta (opcional)	
		Tipo de dado:	STRING
		Se o valor deste parâmetro é o string zero ("") ou se ele não é indicado, o estado atual é utilizado. Sem especificação de nenhuma ferramenta, a ferramenta atual é utilizada.	
7	<T>:	Número T interno da ferramenta (opcional)	
		Tipo de dado:	INT
		Sem indicação deste parâmetro ou se o valor dele é "0", a ferramenta arquivada em <Stat> é utilizada. Se o valor deste parâmetro é "-1", o número T da ferramenta ativa é utilizado. Também é admissível indicar de modo explícito o número da ferramenta ativa. **Aviso:** Sem indicação de <Stat>, o estado atual é utilizado como ambiente de ferramenta. Uma vez que, através de <T> = 0, existe a referência ao número T armazenado no ambiente de ferramenta, nele utiliza-se a ferramenta ativa, o que significa que, neste caso especial, as indicações <T> = 0 e <T> = -1 têm o mesmo significado.	
8	<D>:	Número da aresta cortante de ferramenta (opcional)	
		Tipo de dado:	INT
		Sem indicação deste parâmetro ou se o seu valor é "0", o número D utilizado varia em função da fonte do número T. Se o número T a partir do ambiente de ferramenta é utilizado, também o número D do ambiente de ferramenta é lido, caso contrário, o número da ferramenta atualmente ativa.	
9	<TL>:	Número da correção em função do local (opcional)	
		Tipo de dado:	INT
		Sem indicação deste parâmetro, o número DL utilizado varia em função da fonte do número T. Se o número T a partir do ambiente de ferramenta é utilizado, também o número D do ambiente de ferramenta é lido, caso contrário, o número da ferramenta atualmente ativa. Quando T, D e DL especificam uma ferramenta sem correções em função do local, no parâmetro <Comp> não devem ser especificadas correções de somas ou de ajuste (código de erro em <Status>).	

Indicação

Nem todas as possíveis combinações dos três parâmetros <Comp>, <CorComp> e <CorMode> são úteis. A instrução de cálculo 3, por exemplo, requer em <CorComp> a indicação de dois caracteres em <Comp>. Em caso de indicação de uma combinação inadmissível de parâmetros, em <Status> ocorre a devolução do correspondente código de erro.

Exemplos

Exemplo 1

Código de programa	Comentário
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Ferramenta de fresar
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Desgaste 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	; ==> MKS-Position X0.000 Y0.000 **Z1.333**
N90 M30	

<CorComp> é "0", por isto, o valor da coordenada que atua na direção Z do componente geométrico é substituído pelo valor de correção 0,333.

Portanto, o comprimento de ferramenta resultante é: L1 = 0,333 + 1,000 = 1,333

Exemplo 2

Código de programa	Comentário
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Ferramenta de fresar
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Desgaste 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	; ==> MKS-Position X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> é "1", por isto. o valor da correção de 0,333 que atua na direção Z é somado ao valor de desgaste de 1,0.

Portanto, o comprimento de ferramenta resultante é: L1 = 10,0 + 1,333 = 11,333

Exemplo 3

Código de programa	Comentário
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Ferramenta de fresar
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Desgaste 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	; ==> MKS-Position X0.000 Y0.000 Z0.333
N90 M30	

<CorComp> é "2", por isto, a correção que atua na direção Z é registrada no componente geométrico (o valor antigo é sobrescrito), e o valor de desgaste é apagado.

Portanto, o comprimento de ferramenta resultante é: L1 = 0,333 + 0,0 = 0,333

Exemplo 4

Código de programa	Comentário
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Ferramenta de fresar
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Desgaste 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	; ==> MKS-Position X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> é "3", por isto, o valor de desgaste e o valor da correção são somados ao componente geométrico e o componente de desgaste é apagado.

Portanto, o comprimento de ferramenta resultante é: L1 = 11,333 + 0,0 = 11,333.

Exemplo 5

Código de programa	Comentário
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; Ferramenta de fresar
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP12[1,1]=1.0	; Desgaste 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	; ==> MKS-Position X0.333 Y0.000 Z11.000
N90 M30	

<CorComp> é "3" como no exemplo anterior, a correção, porém, atua agora sobre o eixo geométrico com o índice "0" (eixo X), ao qual, por causa de G17 no caso de uma ferramenta de fresar, está atribuído o componente de ferramenta L3. A chamada de SETTCOR, portanto, não tem influência sobre os parâmetros de ferramenta $TC_DP3 e $TC_DP12. O valor da correção, em vez disso, é registrado em $TC_DP5.

Exemplo 6

Código de programa	Comentário
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; Ferramenta de tornear
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP4[1,1]=15.0	; Geometria L2
N50 $TC_DP12[1,1]=10.0	; Desgaste L1
N60 $TC_DP13[1,1]=0.0	; Desgaste L2

Código de programa	Comentário
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	; ==> MKS-Position X24.330 Y0.000 Z17.500
N120 M30	

A ferramenta é uma ferramenta de tornear. No bloco N80 é ativado a rotação do frame, de modo que o sistema básico de coordenadas (BCS) em relação ao sistema de coordenadas da peça de trabalho (WCS) esteja girado. O valor da correção (N70) atua no WCS sobre o eixo geométrico com o índice "1", ou seja sobre o eixo X, dada a ativação de G18 Valendo <CorMode> = 3, o desgaste de ferramenta na direção do eixo X do WCS deve chegar a zero após a execução de N100.

Portanto, o conteúdo dos parâmetros relevantes de ferramenta no final do programa é:

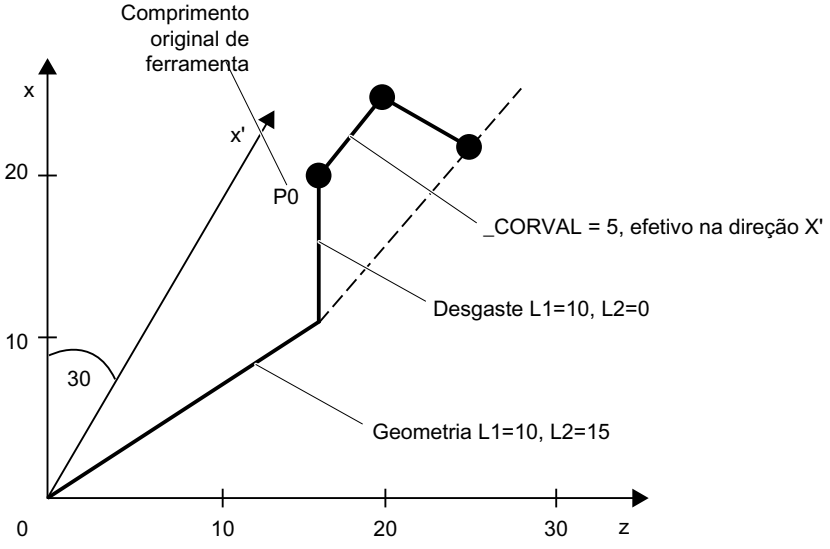
$TC_DP3[1,1] : 21.830 ; Geometria L1

$TC_DP4[1,1] : 21.830 ; Geometria L2

$TC_DP12[1,1] : 2.500 ; Desgaste L1

$TC_DP13[1,1] : -4.330 ; Desgaste L2

As relações geométricas estão representadas na seguinte ilustração. O desgaste total inclusive _CORVAL é projetado sobre a direção X' no WCS. O resultado é o ponto P2. As coordenadas deste ponto (medidas em coordenadas X-Y) são registradas no componente geométrico da ferramenta. No desgaste resta o vetor de diferença $P_2 - P_1$. Com isto, o desgaste não tem mais componente na direção de _CORVAL.



Continuando o programa do exemplo após N110 com as seguintes instruções, o desgaste restante é assumido totalmente pela geometria, uma vez que agora a correção atua no eixo Z' (parâmetro <GeoAx> = 0):

```
N120 _CORVAL[0]=0.0
```

```
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> MKS-Position X24.330 Y0.000 Z17.500
```

Uma vez que o novo valor da correção é "0", não deve haver alterações no comprimento total de ferramenta e com isto também na posição alcançada em N140. Se _CORVAL em N120 fosse desigual "0", resultaria um novo comprimento total de ferramenta e com isto também uma posição alterada em N140, a parte de desgaste do comprimento de ferramenta, porém, seria em todos os casos zero, significa: o subsequente comprimento total de ferramenta, em todos os casos, está incluído na parte geométrica da ferramenta.

O mesmo resultado da chamada por duas vezes da função SETTCOR com o parâmetro <CorComp> = 0 também é alcançado por uma única chamada com <CorComp> = 1 (correção vetorial):

Código de programa	**Comentário**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; Ferramenta de tornear
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP4[1,1]=15.0	; Geometria L2
N50 $TC_DP12[1,1]=10.0	; Desgaste L1
N60 $TC_DP13[1,1]=0.0	; Desgaste 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	; ==> MKS-Position X24.330 Y0.000 Z17.500
N120 M30	

Neste caso, todos os componentes de desgaste da ferramenta chegam a zero imediatamente após a primeira chamada de SETTCOR em N100.

Exemplo 7

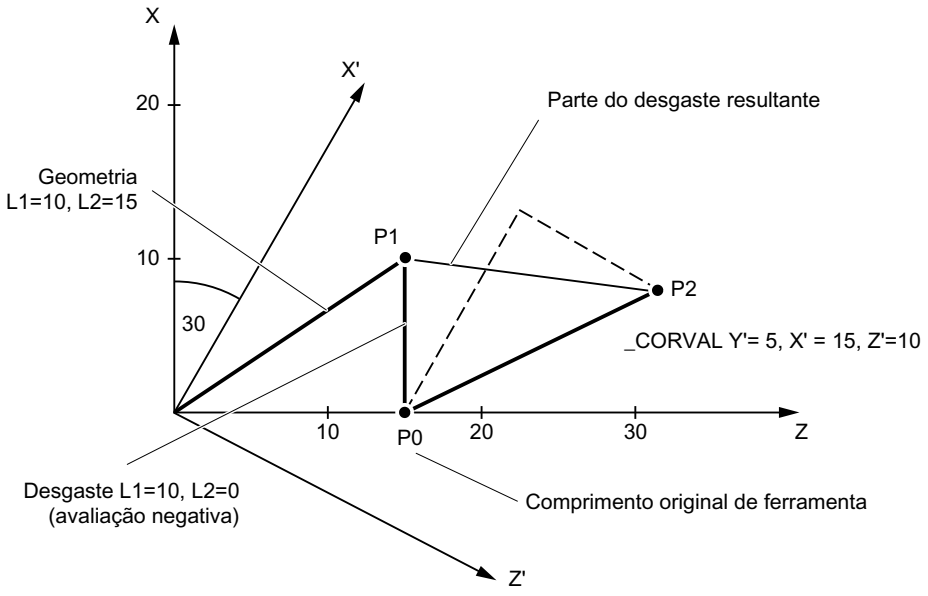
Código de programa	**Comentário**
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; Ferramenta de tornear
N30 $TC_DP3[1,1]=10.0	; Geometria L1
N40 $TC_DP4[1,1]=15.0	; Geometria L2
N50 $TC_DP12[1,1]=10.0	; Desgaste L1
N60 $TC_DP13[1,1]=0.0	; Desgaste L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",3,3)**	
N110 T1 D1 X0 Y0 Z0	; ==> MKS-Position X25.000 Y0.000 Z15.000
N120 M30	

Contrário ao exemplo 6, aqui o parâmetro <CorComp> = 3, por isto, pode-se omitir a indicação do parâmetro <GeoAx>. O valor contido em _CORVAL[0] agora atua diretamente sobre o componente L1 de ferramenta, a rotação em N80 não influencia o resultado, o componente de desgaste em $TC_DP12 é assumido junto com _CORVAL[0] pela parte geométrica, de modo que, devido a $TC_DP13, o comprimento total de ferramenta esteja já após a primeira chamada de SETTCOR em N100 na parte geométrica da ferramenta.

Exemplo 8

```
Código de programa                          Comentário
N10 DEF REAL _CORVAL[3]
N20 $TC_DP1[1,1]=500                        ; Ferramenta de tornear
N30 $TC_DP3[1,1]=10.0                       ; Geometria L1
N40 $TC_DP4[1,1]=15.0                       ; Geometria L2
N50 $TC_DP5[1,1]=20.0                       ; Geometria L3
N60 $TC_DP12[1,1]=10.0                      ; Desgaste L1
N70 $TC_DP13[1,1]=0.0                       ; Desgaste L2
N80 $TC_DP14[1,1]=0.0                       ; Desgaste 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                         ; ==> MKS-Position X7.990 Y25.000 Z31.160
N170 M30
```

Em N90 encontra-se ajustado o dado de ajuste SD42930 $SC_WEAR_SIGN, isto significa que o desgaste deve ser avaliado com sinal negativo. A correção é vetorial (<CorComp> = 1), e o vetor da correção deve ser somado ao desgaste (<CorMode> = 1). As relações geométricas no plano Z-X estão representadas na seguinte ilustração.



Devido a <CorMode> = 1, a parte geométrica da ferramenta permanece inalterada. O vetor da correção definido no WCS (rotação em torno do eixo y) deve ser assumido pela parte do desgaste de modo que o comprimento total de ferramenta na fig. 3 indique para o ponto P_2. A parte resultante do desgaste da ferramenta, portanto, está definida pela distância dos dois pontos P_1 e P_2.
Uma vez que o desgaste, devido ao dado de ajuste SD42930 $SC_WEAR_SIGN, é avaliado de modo negativo, a correção assim apurada deve ser registrada com sinal negativo na memória da correção.. Portanto, o conteúdo dos parâmetros relevantes da ferramenta no final do programa é:

$TC_DP3[1,1] : 10.000 ; Geometria L1 (sem alteração)

$TC_DP4[1,1] : 15.000 ; Geometria L2 (sem alteração)

$TC_DP5[1,1] : 10.000 ; Geometria L3 (sem alteração)

$TC_DP12[1,1] : 2.010 ; Desgaste L1 (= 10 - 15 * cos(30) + 10 * sin(30))

$TC_DP13[1,1] : -16.160 ; Desgaste L2 (= -15 * sin(30) - 10 * cos(30))

$TC_DP14[1,1] : -5.000 ; Desgaste L3

Pelo componente L3 na direção Y pode-se reconhecer o efeito do dado de ajuste SD42930 $SC_WEAR_SIGN sem a complicação adicional da rotação do frame.

Outras informações

Ferramenta de tornear / retificar: Cálculo de comprimento de ferramenta em dependência de MD20360 $MC_TOOL_PARAMETER_DEF_MASK

Para ferramentas de tornear / retificar, o modo de avaliar o desgaste ou o comprimento de ferramenta em um eixo de diâmetro eventualmente existente é determinado no dado de máquina:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <Wert>

<bit>	<valor>	Significado
0	0	Em ferramentas de tornear / retificar, o **parâmetro de desgaste** do eixo transversal é levado em conta no **valor do raio**:
	1	Em ferramentas de tornear / retificar, o **parâmetro de desgaste** do eixo transversal é levado em conta como **valor de diâmetro**:
1	0	Em ferramentas de tornear / retificar, o **componente de comprimento da ferramenta** do eixo transversal é levado em conta como **valor de raio**:
	1	Em ferramentas de tornear / retificar, o **componente de comprimento da ferramenta** do eixo transversal é levado em conta como **valor de diâmetro**:

Estando ajustados os respectivos bits, o registro pertencente a eles é avaliado com o fator 0,5. A correção por meio de SETTCOR é executada de modo que a alteração total efetiva de comprimento de ferramenta seja igual ao valor entregue em <CorVal>. Se um comprimento, devido ao dado de máquina MD20360 $MC_TOOL_PARAMETER_DEF_MASK, é avaliado no cálculo de comprimento com o fator 0,5, a correção deste componente, portanto, deve ser executado com o dobro do valor entregue.

Exemplo

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 2 (o comprimento de ferramenta deve ser avaliado com o fator 0,5 no eixo do diâmetro).

O eixo X é o eixo de diâmetro.

```
Código de programa                          Comentário
N10 DEF REAL _CORVAL[11]
N20 DEF REAL _CORVAL[3]
N30 $TC_DP1[1,1]=500                        ; tipo de ferramenta
N40 $TC_DP2[1,1]=2                          ; posição de corte
N50 $TC_DP3[1,1]=3.                         ; geometria - comprimento 1
N60 $TC_DP4[1,1]=4.                         ; geometria - comprimento 2
N70 $TC_DP5[1,1]=5.                         ; geometria - comprimento 3
N80 _CORVAL[0]=1.
N90 _CORVAL[1]=1.
N100 _CORVAL[2]=1.
N110 T1 D1 G18 G0 X0 Y0 Z0                  ; ==> MKS-Position X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)
N130 T1 D1 X0 Y0 Z0                         ; ==> MKS-Position X2.5 Y6 Z5
N140 R3=$TC_DP3[1,1]                        ; = 5. = (3.000 + 2.*1.000)
N150 R4=$TC_DP4[1,1]                        ; = 5. = (4.000 + 1.000)
N160 R5=$TC_DP5[1,1]                        ; = 6. = (5.000 + 1.000)
N170 M30
```

A correção de comprimento de ferramenta deve ser de 1 mm em cada eixo (N80 até N100). Nos comprimentos L2 e L3, portanto, deve ser somado 1 mm ao comprimento original em cada caso. Ao comprimento original de ferramenta em L1, porém, o dobro do valor da correção (2 mm) deve ser somado para que o comprimento total da ferramenta se modifique em 1 mm, como exigido. Na comparação das posições alcançadas nos blocos N110 e N130 é reconhecível que cada posição de eixo se alterou em 1 mm.

11.13 Leitura da atribuição de comprimentos de eixo L1, L2, L3 aos eixos de coordenada (LENTOAX)

A função LENTOAX fornece informações sobre a atribuição dos comprimentos de ferramenta L1, L2 e L3 da ferramenta **ativa** à abcissa, ordenada e aplicada. A atribuição de abcissa, ordenada e aplicada aos eixos geométricos é influenciada por frames e pelo plano ativo (G17 - G19).

Aqui é considerada somente a parte geométrica de uma ferramenta ($TC_DP3[<t>,<d>] até $TC_DP5[<t>,<d>]), significa que uma atribuição de eixo eventualmente divergente de outros componentes (por ex. desgaste) não influencia o resultado.

Sintaxe

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

Princípio

LENTOAX(...):	Função predefinida para a leitura da atribuição dos comprimentos de ferramenta L1, L2 e L3 da ferramenta ativa ao eixos de coordenada		
	Sozinho no bloco:		sim
<Status>:	Valor de devolução da função. Valores negativos mostram condições de erro.		
	Tipo de dado:		INT
	Valor:	0	Função OK A informação em <AxInd> é suficiente para a descrição (todos os componentes de comprimento de ferramenta estão em posição paralela em relação aos eixos geométricos).
		1	A função está OK, para uma descrição correta, porém, o conteúdo de <Matrix> precisa ser avaliado (os componentes de comprimento de ferramenta não estão em posição paralela em relação aos eixos geométricos).
		-1	String inválido no parâmetro <Coord>.
		-2	Nenhuma ferramenta ativa.
Parâmetros			

1	<AxInd>:	Quando os componentes de comprimento de ferramenta estão em posição paralela em relação aos eixos geométricos, os índices dos eixos, atribuídos aos componentes de comprimento L1 até L3, são devolvidos no campo **<**AxInd**>**: • **<**AxInd**>** [0]: Abscissa • **<**AxInd**>** [1]: Ordenada • **<**AxInd**>** [2]: Aplicada		
		Tipo de dado:		INT[3]
		Valor:	0	Sem atribuição (eixo não existe)
			1 ... 3 ou -1 ... -3	Número do comprimento que atua no correspondente eixo de coordenada. O sinal é negativo quando o componente de comprimento de ferramenta aponta na direção negativa de coordenada.
		Quando nem todos os componentes de comprimento estão em posição paralela ou antiparalela em relação aos eixos geométricos, em **<**AxInd**>** ocorre a devolução do respectivo índice daquele eixo que contém a maior parte de um componente de comprimento de ferramenta. Neste caso (desde que a função não produza um erro por algum outro motivo), o valor de devolução é **<**Status**>** = 1. A ilustração dos componentes de comprimento da ferramenta L1 até L3 sobre os eixos geométricos 1 até 3, é então descrita completamente pelo conteúdo do 2º parâmetro **<**Matrix**>**.		
2	<Matrix>:	Matrix, que reproduz o vetor dos comprimentos de ferramenta (L1=1, L2=1, L3=1) no vetor dos eixos de coordenada (abcissa, ordenada, aplicada), significa: às colunas encontram-se atribuídos os componentes de comprimento de ferramenta na ordem L1, L2, L3, às linhas os eixos na ordem abcissa, ordenada, aplicada.		
		Tipo de dado:		REAL
		Sempre são válidos todos os elementos em Matrix, também quando o eixo geométrico pertencente a um eixo de coordenada não está presente, ou seja quando o correspondente registro em **<**AxInd**>** é zero.		
3	<Coord>:	Sistema de coordenadas para o qual a atribuição vale (opcional)		
		Tipo de dado:		STRING
		Caractere:	MCS M	Reprodução do comprimento de ferramenta no sistema de coordenadas de máquina
			BCS B	Reprodução do comprimento de ferramenta no sistema básico de coordenadas
			WCS W	Reprodução do comprimento de ferramenta no sistema de coordenadas da peça de trabalho (default)
			KCS K	Reprodução do comprimento de ferramenta no sistema de coordenadas da ferramenta da transformação cinemática
			TCS T	Reprodução do comprimento de ferramenta no sistema de coordenadas da ferramenta
		O modo de escrever os caracteres no string (letras maiúsculas ou minúsculas) é de livre escolha. Sem indicação do parâmetro **<**Coord**>** é utilizado o WCS (default).		

Indicação

Em TCS, todos os componentes de comprimento de ferramenta estão em posição paralela ou antiparalela em relação aos eixos.

Posição antiparalela dos componentes somente é possível com espelhamento ativo e com o seguinte dado de ajuste:

SD42900 $SC_MIRROR_TOOL_LENGTH (troca de sinal comprimento de ferramenta ao espelhar)

Exemplo

Caso padrão ferramenta de fresar em G17.

L1 atua em Z (aplicada), L2 atua em Y (ordenada), L3 atua em X (abcissa).

Chamada da função na forma:
```
<Status>=LENTOAX(<AxInd>,<Matrix>,"WCS")
```

O parâmetro do resultado **<AxInd>**, então, contém os valores:

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

ou abreviado: (3, 2, 1)

Matrix pertencente (**<Matrix>**) neste caso é:

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Uma mudança de G17 para G18 ou G19 não altera o resultado, uma vez que a atribuição dos componentes de comprimento aos eixos geométricos se altera da mesma forma que a atribuição de abcissa, ordenada e aplicada.

Agora, com G17 ativo, uma rotação de 60 graus do frame em torno de Z é programada, por ex.:
```
ROT Z60
```

A direção da aplicada (direção Z) permanece inalterada, a parte principal de L2 agora está na direção do novo eixo X, a parte principal de L1 na direção do eixo Y negativo. O valor de devolução (**<Status>**), portanto, é "1", **<AxInd>** contém os valores (2, -3, 1).

Matrix pertencente (**<Matrix>**) neste caso é:

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

Comportamento de percurso 12

12.1 Controle tangencial

12.1.1 Definição de acoplamento (TANG)

Com o procedimento pré-definido TANG (...) será definido um acoplamento tangencial entre um eixo rotativo como eixo escravo e dois eixos geométricos como eixo condutor. o eixo escravo é alinhado continuamente até a tangente da trajetória dos eixos condutores.

Indicação

Fator de acoplamento

Um fator de acoplamento de 1 não precisa ser programado de forma explícita.
Através do fator de acoplamento -1, a direção do eixo tangencial é girada.

Sintaxe

```
TANG(<eixo escravo>, <eixo condutor_1>, <eixo condutor_2>, <fator de acoplamento>, <sistema de coordenação>, <otimização>)
```

Significado

`TANG(...)`:	Definição de um acoplamento tangencial	
`<eixo escravo>`:	Nome do eixo escravo (eixo rotativo)	
	Tipo de dado:	AXIS
	Faixa de valores:	Nomes do eixos de canal
`<eixo condutor_1>` `<eixo condutor_2>`:	Nome dos eixos condutores (eixos geométricos) [1)]	
	Tipo de dado:	AXIS
	Faixa de valores:	Nomes dos eixos geométricos do canal
`<fator de acoplamento>`:	Fator n da mudança de ângulo do eixo escravo da alteração da tangente de trajetória do eixos condutores: alteração do ângulo, $_{\text{eixo escravo}}$ = alteração do ângulo, $_{\text{tangente de trajetória}}$ * n	
	Tipo de dado:	REAL
	Valor padrão:	1,0

<sistema de coordenação>:	Sistema efetivo de coordenação [2]		
	Tipo de dado:	CHAR	
	Valor:	"B":	Sistema básico de coordenação (valor padrão)
		"W":	Sistema de coordenação de peças (não disponível)
<otimização>:	Tipo de otimização		
	Tipo de dado:	CHAR	
	Valor:	"S":	Standard (valor padrão) A dinâmica do eixo rotativo não tem efeito sobre os eixos condutores. Quando a dinâmica do eixo rotativo é maior que o necessário para o acompanhamento, este método é suficientemente preciso. Se a dinâmica do eixo rotativo não for suficientemente elevada para seguir a mudança da tangente de trajetória, o alinhamento do eixo rotativo desvia ao longo de um curso de suavização não definido da orientação desejada.
		"P":	A dinâmica do eixo rotativo é tida em conta no planejamento da trajetória dos eixos condutores. Ao ligar o acoplamento tangencial com TANGON(), é preciso indicar dois parâmetros adicionais: • curso de suavização • tolerância angular Veja o capítulo "Ligar acoplamento (TANGON) (Página 528)" **Nota** Em relação às transformações de cinemáticas, é recomendável usar o tipo de otimização "P".

Indicação

Os valores padrão não precisam ser programados de forma explícita.

[1] **Nota**

Como eixos condutores do acoplamento tangencial devem ser usados os eixos geométricos, que se movem nas trajetórias programadas relacionadas à posição normal da máquina, no sistema de coordenação de máquinas (MKS). Se, p. ex., for usado um ciclo giratório CYCLE800 em uma fresa com cabeçote giratório, ocorre, dependente da configuração do ciclo, a interpolação no WCS p. ex., com os eixos geométricos X e Y. O acoplamento tangencial deve ser definido com os eixos geométricos como eixos condutores, que movem no MCS as trajetórias programadas. Para este efeito, os eixos condutores deverão ser usados como eixos geométricos no estado **não inclinado** da máquina.

[2] **Nota**

O sistema básico de coordenação (BKS) não deve estar girado em relação ao MCS. Se o BKS gira, p. ex. através do comando VERMELHO ou através do ciclo giratório CYCLE800, o comando tangencial não funciona corretamente.

12.1.2 Ligar a geração de ponte (TLIFT)

Se a alteração de tangente do eixo escravo exceder o valor limite parametrizável na data da máquina MD37400 $MA_EPS_TLIFT_TANG_STEP em um ponto da trajetória programada do eixos condutores, ocorre o outro planejamento de trajetória dependendo do comportamento de cantos em conjunto. Sem a utilização do procedimento pré-definido TLIFT(...), será conduzido de acordo ao comportamento de suavização programado em relação ao TANG(...) (Página 525) e TANGON(...) (Página 528).

Ligar geração de ponte

Com a programação do TLIFT(...) em conexão no TANG(...) será inserido um bloco intermediário gerado automaticamente quando um canto neste ponto da trajetória é reconhecido pelo comando no pré-curso.

Naexecução do programa, os eixos mestre então são parados ao alcançar o bloco intermediário. No bloco intermediário, o eixo escravo é girado com máxima dinâmica de eixo no sentido da tangente de trajetória do bloco seguinte. Posteriormente, os eixos condutores continuarão sendo movidos na trajetória programada.

Desligar a geração de ponte

Para desligar a geração de ponte, o acoplamento tangencial deve ser redefinido com o TANG(...), mas sem ligação posterior da geração de ponte com TLIFT(...).

Sintaxe

```
TLIFT(<eixo escravo>)
```

Significado

`TLIFT(...)`:	Ligar o reconhecimento de cantos com cálculo de ponte intermediária	
`<eixo escravo>`:	Nome do eixo escravo (eixo rotativo)	
	Tipo de dado:	AXIS
	Faixa de valores:	Nomes do eixos de canal

Velocidade de rotação do eixo escravo

Eixo de trajetória

Se o eixo escravo foi movido antes da ativação do acoplamento tangencial já como eixo de trajetória, o movimento de rotação no bloco intermediário ocorre como eixo de trajetória.

Ao especificar o raio de referência com `FGREF[<eixo>]=0.001`, ocorre movimento de rotação com a velocidade máxima parametrizável do eixo:

MD32000 $MA_MAX_AX_VELO[<eixo escravo>]

Eixo de posicionamento

Se o eixo escravo foi movido antes da ativação do acoplamento tangencial já como eixo de trajetória, a rotação no bloco intermediário ocorre como eixo de posição.

O movimento de rotação ocorre com a velocidade de eixo de posição parametrizável:

MD32060 $MA_POS_AX_VELO[<eixo escravo>]

12.1.3 Ligar acoplamento (TANGON)

Com o procedimento pré-definido TANGON(...) o acoplamento tangencial, definido antes com TANG(...) (Página 525), será ligado. O eixo escravo será alinhado no deslocamento posterior dos eixos condutores continuamente até a tangente de trajetória.

Ângulo do eixo escravo

O ângulo, que ocupa o eixo escravo em relação à tangente de trajetória, depende da relação de desmultiplicação prescrita no TANG(...), do ângulo de desvio parametrizável na data da máquina MD37402 $MA_TANG_OFFSET e ângulo de desvio aditivo prescrito, efetivo no TANGON(...).

Otimização "P"

Se na definição do acoplamento tangencial (TANG(...)) foi dado o valor "P" como parâmetro de otimização, ao ligar deve-se especificar o acoplamento do parâmetro "curso de suavização" e opcionalmente o parâmetro "tolerância angular".

Se a tolerância angular dada tiver o valor 0, só o parâmetro "curso de suavização" terá efeito.

Se a tolerância angular tiver o valor maior que 0, o curso de suavização dar-se do mínimo do curso de suavização parametrizável e do curso de suavização devido à tolerância angular parametrizável.

Se a dinâmica do eixo escravo não é suficiente para seguir as condições parametrizáveis, a velocidade de trajetória dos eixos condutores será reduzida de acordo.

Sintaxe

```
TANGON(<eixo escravo>, <ângulo de desvio>, <curso de suavização>,
<tolerância angular>)
```

Significado

`TANGON(...)`:	Ligar um acoplamento tangencial	
`<eixo escravo>`:	Nome do eixo escravo (eixo rotativo)	
	Tipo de dado:	AXIS
	Faixa de valores:	Nomes do eixos de canal
`<ângulo de desvio>`:	Ângulo de desvio do eixo escravo à tangente de trajetória O ponto de referência é o ponto de zero do eixo rotativo.	
	Tipo de dado:	REAL
`<curso de suavização>`:	Curso de suavização máximo admissível Se o curso de suavização fosse maior devido às condições de dinâmica, a velocidade de trajetória do eixos condutores será reduzida.	
	Tipo de dado:	REAL

<tolerância angular>:	Tolerância máxima admissível em relação ao ângulo predeterminado entre posição zero do eixo escravo e a tangente do percurso	
	Tipo de dado:	REAL

12.1.4 Desligar acoplamento (TANGOF)

Com o procedimento pré-definido TANGOF(...) o acoplamento tangencial, definido com TANG(...) (Página 525) e com TANGON(...) (Página 528) será ligado. O eixo escravo não será mais alinhado sobre a tangente de trajetória do eixo condutor. O acoplamento do eixo escravo aos eixos condutores permanece mesmo após desligamento,o que evita p. ex., as seguintes funções:

- Mudança de planos
- Comutação de eixo geométrico
- Definição de um novo acoplamento tangencial do eixo escravo

A suspensão completa do eixo escravo aos eixos condutores ocorre somente após apagar o acoplamento através do TANGDEL(...) (Página 529).

Programação

```
TANGOF(<eixo escravo>)
```

Significado

TANGOF(...):	Desligar um acoplamento tangencial	
<eixo escravo>:	Nome do eixo escravo (eixo rotativo)	
	Tipo de dado:	AXIS
	Faixa de valores:	Nomes do eixos de canal

12.1.5 Deletar acoplamento (TANGDEL)

Um acoplamento tangencial definido com TANG(...) (Página 525) permanece mesmo após desligar o acoplamento tangencial com TANGOF(...) (Página 529). O acoplamento existente tangencial evita então p. ex., as seguintes funções:

- Mudança de planos
- Comutação de eixo geométrico
- Definição de um novo acoplamento tangencial do eixo escravo

Com o procedimento pré-definido TANGDEL(...) após desligar o acoplamento tangencial com TANGOF(...), o acoplamento tangencial existente será apagado.

Sytax

```
TANGDEL(<eixo escravo>)
```

Significado

TANGDEL(...):	Apagar um acoplamento tangencial definido com TANG()	
	Efeito:	Por blocos
<eixo escravo>:	Nome do eixo escravo, cujo acoplamento tangencial deverá ser apagado	
	Tipo de dado:	AXIS
	Faixa de valores:	Nomes do eixos de canal

Exemplos

Substituição de eixo condutor

Antes de poder definir o acoplamento tangencial novo para o eixo escravo com outro eixo condutor, o acoplamento tangencial existente deve ser apagado primeiro.

```
Código de programa                Comentário
N10 TANG(A, X, Y,1)               ; Definir o acoplamento tangencial para o eixo escravo
                                  A: A até X e Y
N20 TANGON(A)                     ; ligar acoplamento tangencial do eixo escravo A
N30 X10 Y20
...
N80 TANGOF(A)                     ; desligar acoplamento tangencial do eixo escravo A
N90 TANGDEL(A)                    ; eliminar acoplamento tangencial do eixo escravo A
...
N120 TANG(A, X, Z)                ; definir novo acoplamento tangencial do eixo escravo A
N130 TANGON(A)                    ; ligar novo acoplamento tangencial do eixo escravo A
...
```

Comutação de eixo geométrico

Antes que uma transição do eixo geométrico do acoplamento existente possa ser feita, o acoplamento deve ser apagado primeiro.

```
Código de programa                Comentário
N10 GEOAX(2, Y1)                  ; 2. Eixo geométrico = eixo de máquina Y1
N20 TANG(A, X, Y)                 ; Definir o acoplamento tangencial para o eixo escra-
                                  vo A
N30 TANGON(A, 90)                 ; ligar acoplamento tangencial do eixo escravo A
N40 G2 F8000 X0 Y0 I0 J50         ; Bloco
N50 TANGOF(A)                     ; desligar acoplamento tangencial do eixo escravo A
N60 TANGDEL(A)                    ; eliminar acoplamento tangencial do eixo escravo A
N70 GEOAX(2, Y2)                  ; 2. Eixo geométrico = eixo de máquina Y2
N80 TANG(A, X, Y)                 ; definir novo acoplamento tangencial do eixo escravo
                                  A
N90 TANGON(A, 90)                 ; ligar novo acoplamento tangencial do eixo escravo A
...
```

12.2 Característica de avanço (FNORM, FLIN, FCUB, FPO)

Para uma especificação flexível da característica de avanço a programação do avanço conforme DIN 66025 é ampliada em características lineares e cúbicas.

As características cúbicas podem ser programadas diretamente ou como Splines interpoladoras. Com isso pode-se programar características de velocidade suaves e contínuas, em função da curvatura da peça de trabalho usinada.

Estas características de velocidade permitem mudanças de aceleração isentas de solavancos tendo como resultado uma usinagem uniforme das superfícies das peças de trabalho.

Sintaxe

```
F… FNORM
F… FLIN
F… FCUB
F=FPO(…,…,…)
```

Significado

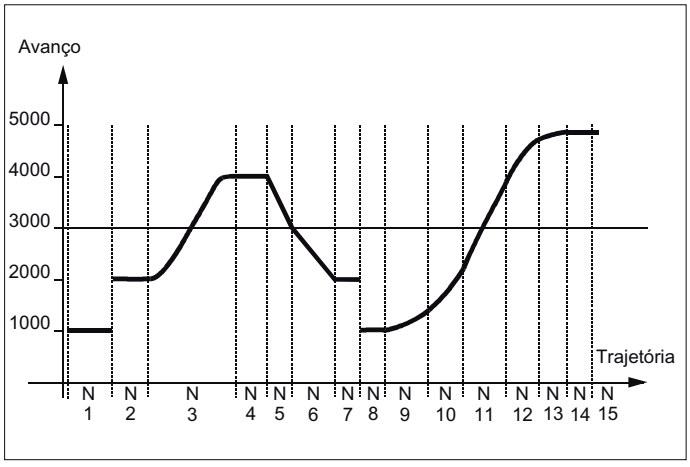
FNORM:	Ajuste básico. O valor do avanço é especificado através do percurso do bloco e depois ele vale como valor modal.
FLIN:	Perfil de velocidade de percurso **linear:** O valor de avanço é executado linearmente pelo percurso a partir do atual valor no início do bloco até o fim do bloco e depois ele vale como valor modal. Esta característica pode ser combinada com G93 e G94.
FCUB:	Perfil da velocidade de percurso **cúbica:** Os valores F programados por blocos são, em função do ponto final do bloco, ligados por uma Spline. A Spline inicia e finda tangencialmente com os avanços anterior e posterior definidos e tem efeito com G93 e G94. Se faltar o endereço F em um bloco, então neste caso se utiliza o último valor F programado.
F=FPO…:	Perfil da velocidade de percurso **através de polinômio:** O endereço F define a característica de avanço através de um polinômio a partir do atual valor até o fim do bloco. Depois o valor final vale como valor modal.

Otimização de avanço em trechos de percurso curvados

O polinômio de avanço F=FPO e a Spline de avanço FCUB sempre deveriam ser afastados com velocidade de corte constante CFC. Com isso é possível produzir um perfil de avanço nominal com aceleração isenta de solavancos.

Exemplo: Diferentes perfis de avanço

Neste exemplo temos a programação e a representação gráfica de diversos perfis de avanço.



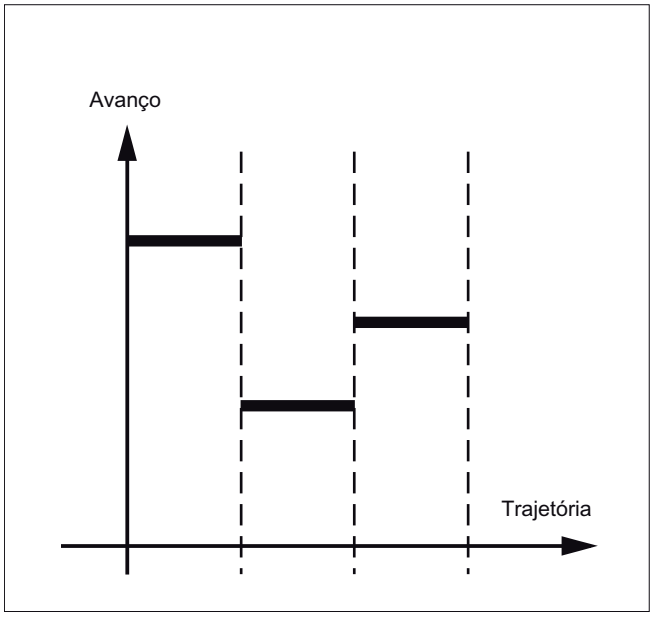
Código de programa	**Comentário**
N1 F1000 FNORM G1 X8 G91 G64	; Perfil de avanço constante, especificação incremental
N2 F2000 X7	; Mudança brusca de velocidade nominal
N3 F=FPO(4000, 6000, -4000)	; Perfil de avanço através de polinômio com avanço 4000 no fim do bloco.
N4 X6	; O avanço de polinômio 4000 vigora como valor modal.
N5 F3000 FLIN X5	; Perfil de avanço linear
N6 F2000 X8	; Perfil de avanço linear
N7 X5	; O avanço linear vale como valor modal
N8 F1000 FNORM X5	; Perfil constante de avanço com mudança brusca de aceleração.
N9 F1400 FCUB X8	; Todos os valores F programados por blocos a seguir serão ligados com Splines.
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; Desativar perfil de Spline.
N14 FNORM X5	
N15 X20	

Outras informações

FNORM

O endereço de avanço F define o avanço de trajetória como valor constante conforme DIN 66025.

Para mais informações sobre este assunto, veja no Manual de programação "Fundamentos".

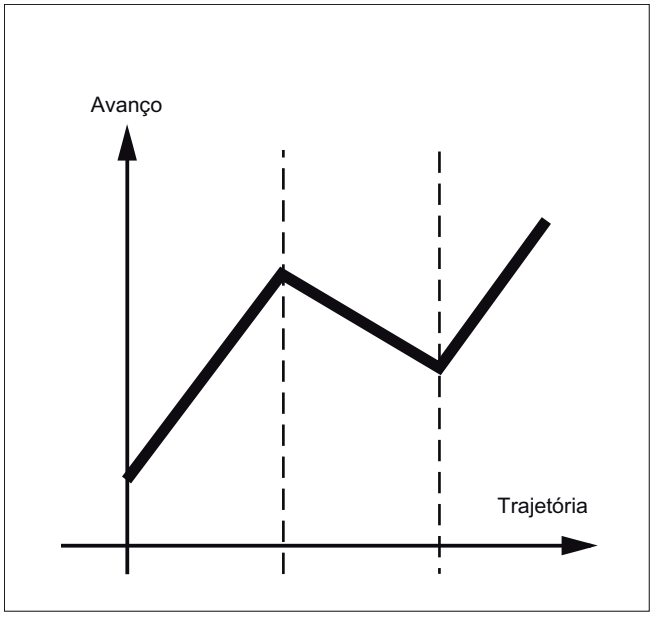


FLIN

A característica de avanço é executada linearmente do atual valor de avanço o valor F programado até atingir o final do bloco.

Exemplo:

```
N30 F1400 FLIN X50
```



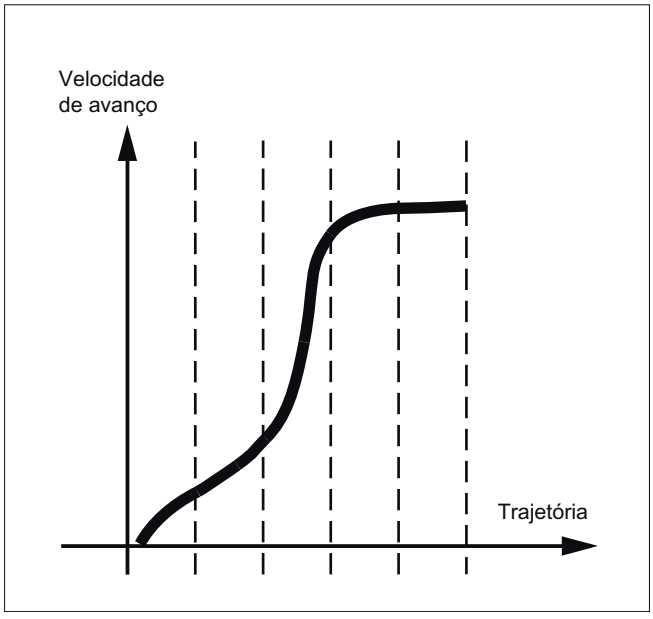
FCUB

O avanço é executado de acordo com a característica cúbica do atual valor de avanço para o valor F até o final do bloco. O comando liga com Splines todos valores de avanço

programados por blocos com FCUB ativo. Aqui os valores de avanço servem como pontos de apoio para o cálculo da interpolação de Spline.

Exemplo:

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```



F=FPO(...,...,...)

A característica de avanço é programada diretamente através de um polinômio. A especificação do coeficiente de polinômio é realizado de forma similar à interpolação de polinômios.

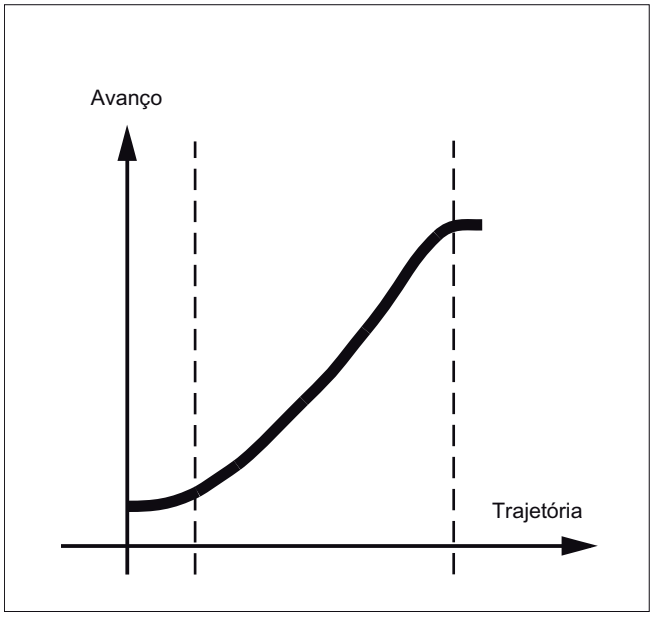
Exemplo:

```
F=FPO(endfeed, quadf, cubf)
```

endfeed, quadf e cubf são variáveis que são definidas antes.

endfeed:	Avanço no fim do bloco
quadf:	Coeficiente quadrático de polinômios
cubf:	Coeficiente cúbico de polinômios

Com `FCUB` ativo a Spline liga tangencialmente à característica definida por `FPO`, no início do bloco e no fim do bloco.



Condições gerais

- Independentemente da característica de avanço programada, aplicam-se as funções de programação do comportamento de percurso.
- A característica de avanço programável sempre é aplicada de forma absoluta, independentemente de `G90` ou `G91`.
- A característica de avanço `FLIN` e `FCUB` tem efeito com `G93` e `G94`, **não** com `G95`, `G96/G961` e `G97/G971`.
- No caso do compressor ativo `COMPON` e agrupamento de vários blocos em uma segmento Spline aplica-se o seguinte:

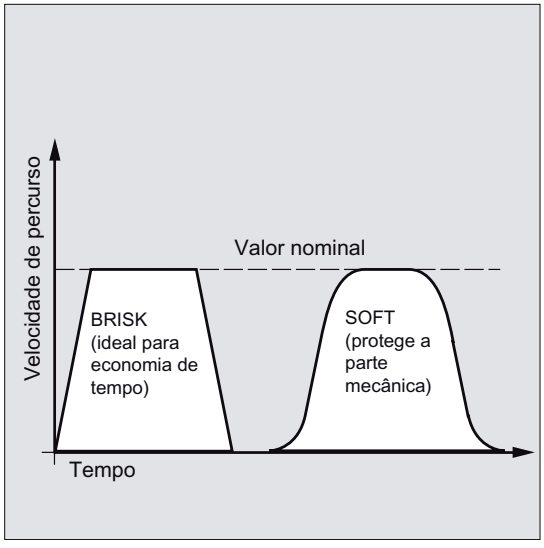
`FNORM`:	Para o segmento de Spline se aplica a palavra F do último bloco correspondente.
`FLIN`:	Para o segmento de Spline se aplica a palavra F do último bloco correspondente. O valor F programado vale até o fim do segmento e depois ele é aproximado linearmente.
`FCUB`:	A Spline de avanço tem um desvio máximo equivalente ao valor definido no dado de máquina MD20172 $MC_COMPRESS_VELO_TOL referente aos pontos finais programados.
`F=FPO(…,…,…)`:	Estes blocos não são comprimidos.

12.3 Comportamento da aceleração

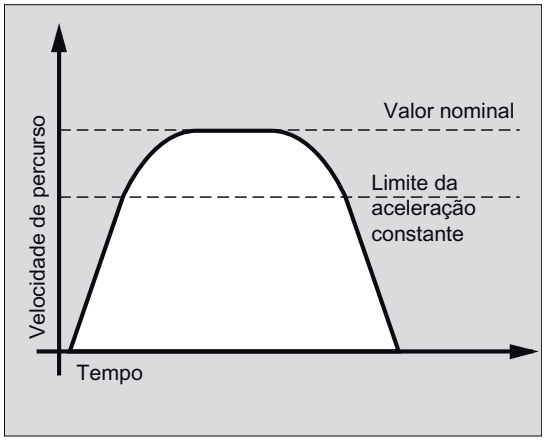
12.3.1 Modo de aceleração (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

Para programação do modo de aceleração estão disponíveis os seguintes comandos de programa de peça:

- "BRISK, BRISKA"
  Os eixos individuais e os eixos de percurso são deslocados com a máxima aceleração até alcançarem a velocidade de avanço programada (**Aceleração sem limitação de solavancos**).
- "SOFT, SOFTA"
  Os eixos individuais e os eixos de percurso são deslocados com aceleração constante até alcançarem a velocidade de avanço programada (**Aceleração com limitação de solavancos**).
- "DRIVE, DRIVEA"
  Os eixos individuais e os eixos de percurso são deslocados com a aceleração máxima até um determinado limite de velocidade projetado (ajuste de dado de máquina!). Em seguida é realizada uma redução de aceleração (ajuste de dado de máquina!) até ser alcançada a velocidade de avanço programada.



Esquema 12-1 Desenvolvimento da velocidade de percurso com `BRISK` e `SOFT`



Esquema 12-2 Desenvolvimento da velocidade de percurso com DRIVE

Sintaxe

```
BRISK
BRISKA(<eixo1>,<eixo2>,…)
SOFT
SOFTA(<eixo1>,<eixo2>,…)
DRIVE
DRIVEA(<eixo1>,<eixo2>,…)
```

Significado

BRISK:	Comando para ativação da "Aceleração sem limitação de solavancos" para eixos de percurso.
BRISKA:	Comando para ativação da "Aceleração sem limitação de solavancos" para movimentos de eixo individual (JOG, JOG/INC, eixo de posicionamento, eixo oscilante, etc.).
SOFT:	Comando para ativação da "Aceleração com limitação de solavancos" para os eixos de percurso.
SOFTA:	Comando para ativação da "Aceleração com limitação de solavancos" para movimentos de eixo individual (JOG, JOG/INC, eixo de posicionamento, eixo oscilante, etc.).
DRIVE:	Comando para ativação da aceleração reduzida acima de um determinado limite de velocidade projetado (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) para os eixos de percurso.
DRIVEA:	Comando para ativação da aceleração reduzida acima de um determinado limite de velocidade projetado (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) para movimentos de eixo individual (JOG, JOG/INC, eixo de posicionamento, eixo oscilante, etc.).
(<eixo1>,<eixo2>,…):	Eixos individuais que devem ser aplicados para o modo de aceleração chamado.

Condições gerais

Mudança do modo de aceleração durante a usinagem

Se em um programa de peça o modo de aceleração for mudado durante o processo de usinagem (BRISK ↔ SOFT), também será realizada uma mudança de blocos com parada exata no fim do bloco durante o modo de controle da trajetória na transição.

Exemplos

Exemplo 1: SOFT e BRISKA

Código de programa
N10 G1 X… Y… F900 SOFT
N20 BRISKA(AX5,AX6)
...

Exemplo 2: DRIVE e DRIVEA

Código de programa
N05 DRIVE
N10 G1 X… Y… F1000
N20 DRIVEA (AX4, AX6)
...

Literatura

Manual de funcionamento das funções básicas; Aceleração (B2)

12.3.2 Influência da aceleração em eixos escravos (VELOLIMA, ACCLIMA, JERKLIMA)

Em acoplamentos de eixos (Acompanhamento tangencial, movimento acoplado, acoplamento de valor mestre, caixa de transmissão eletrônica; consulte "Acoplamentos de eixo (Página 573)") os eixos/fusos escravos são deslocados em função de um ou mais eixos/fusos mestres.

A limitação de dinâmica dos eixos/fusos escravos podem ser controlados com as funções VELOLIMA, ACCLIMA e JERKLIMA a partir do programa de peça ou a partir de ações sincronizadas, mesmo com um acoplamento de eixo já ativo.

Indicação

A função JERLIMA não está disponível para todos tipos de acoplamento.

Literatura:

- Manual de funcionamento, funções especiais; Acoplamentos de eixos (M3)
- Manual de funções ampliadas; Fuso sincronizado (S3)

Indicação

Disponibilidade no SINUMERIK 828D

As funções VELOLIMA, ACCLIMA e JERKLIMA somente podem ser utilizadas no SINUMERIK 828D junto com a função "Movimento acoplado"!

Sintaxe

```
VELOLIMA(<eixo>)=<valor>
ACCLIMA(<eixo>)=<valor>
JERKLIMA(<eixo>)=<valor>
```

Significado

VELOLIMA:	Comando para correção da **velocidade** máxima parametrizada
ACCLIMA:	Comando para correção da **aceleração** máxima parametrizada
JERKLIMA:	Comando para correção do **solavanco** máximo parametrizado
<eixo>:	Eixo escravo, cujas limitações de dinâmica devem ser corrigidas
<valor>:	Valor de correção percentual

Exemplos

Exemplo 1: Correção das limitações de dinâmica para um eixo escravo (AX4)

Código de programa	Comentário
...	
VELOLIMA[AX4]=75	; Correção de limitação para 75% da velocidade máxima por eixo armazenada no dado de máquina.
ACCLIMA[AX4]=50	; Correção de limitação para 50% da aceleração máxima por eixo armazenada no dado de máquina.
JERKLIMA[AX4]=50	; Correção de limitação para 50% do solavanco máximo por eixo armazenada no dado de máquina para movimento de percurso.
...	

Exemplo 2: Caixa de transmissão eletrônica

O eixo 4 é acoplado ao eixo X através de um acoplamento da "caixa de transmissão eletrônica". O valor de aceleração do eixo escravo é limitado em 70 % da aceleração máxima. A velocidade máxima permitida é limitada em 50 % da velocidade máxima. Após a comutação de acoplamento ser executada a velocidade máxima é retornada novamente em 100 %.

Código de programa	Comentário
...	
N120 ACCLIMA[AX4]=70	; Aceleração máxima reduzida.
N130 VELOLIMA[AX4]=50	; Velocidade máxima reduzida.
...	

Código de programa	Comentário
N150 EGON(AX4,"FINE",X,1,2)	; Ativação do acoplamento de caixa de transmissão eletrônica.
...	
N200 VELOLIMA[AX4]=100	; Velocidade máxima cheia.
...	

Exemplo 3: Controle do acoplamento de valor mestre por ação síncrona estática

O eixo 4 é acoplado ao X através do acoplamento de valor mestre. O comportamento de aceleração é limitado em 80 % por ação síncrona estática 2 a partir da posição 100.

Código de programa	Comentário
...	
N120 IDS=2 WHENEVER $AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80	; Ação síncrona
N130 LEADON(AX4, X, 2)	; Acoplamento de valor mestre ativado
...	

12.3.3 Ativação de valores de dinâmica específicos de tecnologia (DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH)

Através do grupo G "Tecnologia" podem ser ativados 5 passos diferentes de usinagem tecnológicos para a dinâmica adequada.

Os valores de dinâmica e os códigos G são configuráveis, e por isso dependem dos ajustes dos dados de máquina (→ Fabricante da máquina!).

Literatura:
Manual de funcionamento das funções básicas; Modo de controle da trajetória, Parada exata, LookAhead (B1)

Sintaxe

Ativação de valores de dinâmica:
```
DYNNORM
DYNPOS
DYNROUGH
DYNSEMIFIN
DYNFINISH
```

Indicação

Os valores de dinâmica são ativados no bloco em que o respectivo comando G for programado. Não se executa nenhuma parada na usinagem.

Leitura ou gravação de um determinado elemento de campo:
```
R<m>=$MA...[n,X]
```

```
$MA...[n,X]=<valor>
```

Significado

DYNNORM:	Comando G para ativação da **dinâmica normal**	
DYNPOS:	Comando G para ativação da dinâmica para **modo de posicionamento**, **rosqueamento com macho**	
DYNROUGH:	Comando G para ativação da dinâmica para **desbaste**	
DYNSEMIFIN:	Comando G para ativação da dinâmica para **acabamento**	
DYNFINISH:	Comando G para ativação da dinâmica para **acabamento fino**	
R<m>:	Parâmetro de cálculo com número <m>	
$MA...[n,X]:	Dado de máquina com elemento de campo determinante da dinâmica	
<n>:	Índice de campo	
	Faixa de valores:	0 ... 4
	0	Dinâmica normal (DYNNORM)
	1	Dinâmica para modo de posicionamento (DYNPOS)
	2	Dinâmica para desbaste (DYNROUGH)
	3	Dinâmica para acabamento (DYNSEMIFIN)
	4	Dinâmica para acabamento fino (DYNFINISH)
<X>:	Endereço de eixo	
<valor>:	Valor de dinâmica	

Exemplos

Exemplo 1: Ativação de valores de dinâmica

Código de programa	Comentário
DYNNORM G1 X10	; Posição inicial
DYNPOS G1 X10 Y20 Z30 F…	; Modo de posicionamento, rosqueamento
DYNROUGH G1 X10 Y20 Z30 F10000	;desbaste
DYNSEMIFIN G1 X10 Y20 Z30 F2000	;acabamento
DYNFINISH G1 X10 Y20 Z30 F1000	; Acabamento fino

Exemplo 2: Leitura ou gravação de um determinado elemento de campo

Aceleração máxima para desbaste, eixo X.

Código de programa	Comentário
R1=$MA_MAX_AX_ACCEL[2, X]	; Leitura
$MA_MAX_AX_ACCEL[2, X]=5	; Gravação

12.4 Deslocamento com controle antecipado (FFWON, FFWOF)

Através do controle feedforward o erro de seguimento dependente da velocidade é reduzido até um valor próximo de zero. O deslocamento com controle feedforward, permite uma maior precisão de trajetória e consequentemente melhores resultados de acabamento.

Sintaxe

```
FFWON
FFWOF
```

Significado

FFWON:	Comando para **ativar** o controle feedforward
FFWOF:	Comando para **desativar** o controle feedforward

Indicação

Através dos dados de máquina define-se o tipo de controle feedforward e quais eixos de percurso devem ser movimentados com este controle.

Padrão: Controle feedforward em função da velocidade

Opcional: Controle feedforward em função da aceleração

Exemplo

Código de programa

```
N10 FFWON
N20 G1 X… Y… F900 SOFT
```

12.5 Precisão de contorno programável (CPRECON, CPRECOF)

A função "precisão de contorno programável" reduz o erro do trajeto em contornos contorcidos através da adequação automática da velocidade.

A precisão de contorno a ser mantida dependerá da projeção da máquina (MD20470 $MC_MC_CPREC_WITH_FFW; veja as indicações do fabricante) ou através dos dados de configuração $SC_CONTPREC ou da tolerância de contorno programado CTOL. Quanto menor o valor e o fator K_V dos eixos geométricos, menor será impacto do avanço nos contornos.

A função "precisão de contorno programável" é ativada e desativada através das instruções `CPRECON` e `CPRECOF` no programa NC.

Sintaxe

```
CPRECON
...
CPRECOF
```

Significado

`CPRECON`:	Chamada de comando G: **Ativar** "precisão de contorno programável"	
	Efeito:	modal
`CPRECOF`:	Chamada de comando G: **Desativar** "Precisão de contorno programável"	
	Efeito:	modal

CPRECON e CPRECOF formam juntos o grupo de função G39 (precisão de contorno programável).

Indicação

Através dos dados de configuração $SC_MINFEED (avanço de trajeto minimo no CPRECON) o usuário pode determinar uma velocidade mínima para o avanço da trajetória.

O avanço não é limitado a este valor, a não ser que um valor F menor tenha sido programado ou a limitação dinâmica dos eixos obriguem uma velocidade mais baixa do percurso.

Indicação

A função "precisão de contorno programável" observa apenas os eixos geométricos do trajeto. Não influenciando portanto a velocidade dos eixos de posicionamento.

Exemplo

Código de programa	**Comentário**
`N10 G0 X0 Y0`	
`N20 CPRECON`	`; Ativação da "precisão de contorno programável".`

Código de programa	**Comentário**
N30 G1 G64 X100 F10000	; Usinagem com 10 m/min em modo de controle da trajetória.
N40 G3 Y20 J10	; Limite automático de avanço no bloco circular.
N50 G1 X0	; Avanço novamente sem limite (10 m/min).
...	
N100 CPRECOF	; Desativação da "Precisão de contorno programável".
N110 G0 ...	

Literatura

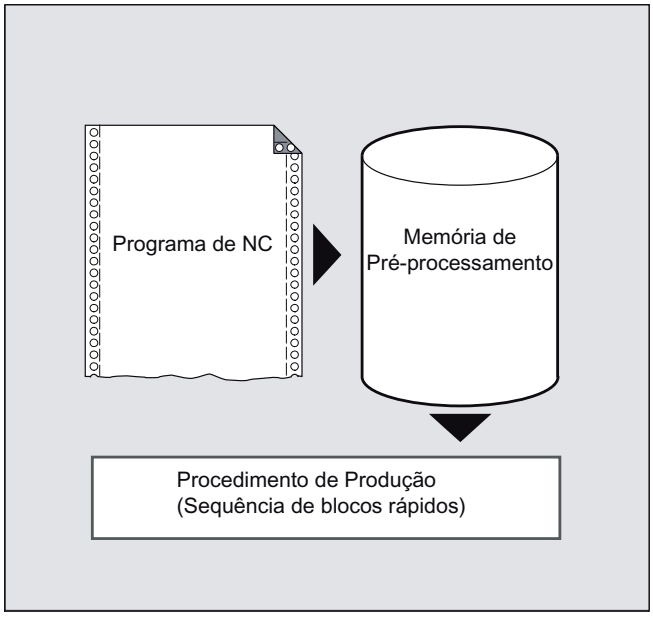
Para programação de CTOL consulte "Programar tolerância de contorno/orientação (CTOL, OTOL, ATOL) (Página 567)"

Para informações detalhadas sobre a função "precisão de contorno programável", veja:

Manual de funcionamento, funções especiais; monitoramento do túnel de contorno (K6), Capítulo: "Precisão de contorno programável"

12.6 Execução do programa com memória de pré-processamento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

Dependendo do nível de expansão, o comando dispõe através de uma determinada quantidade da assim chamada memória de pré-processamento na qual são armazenados os blocos antes deles serem executados, depois são disparadas sequências rápidas de blocos durante a execução da usinagem. Com isso se pode percorrer pequenos cursos em alta velocidade. Assim que o tempo restante do comando permitir, a memória de pré-processamento é continuamente alimentada.



Marcar o segmento de processamento

O segmento de usinagem que deve ser armazenado temporariamente na memória de pré-processamento, é identificado no início do programa de peças com "STOPFIFO" e no fim com "STARTFIFO". A execução dos blocos preparados e armazenados temporariamente somente se inicia após o comando "STARTFIFO" ou quando a memória de pré-processamento estiver cheia.

Controle automático de memória de pré-processamento

O controle automático de memória de pré-processamento é acessada com o comando "FIFOCTRL". Em princípio o "FIFOCTRL" atua da mesma forma como o "STOPFIFO". Em cada programação espera-se até a memória de pré-processamento estar cheia, para então ser iniciada a execução. De modo contrário, a diferença fica no comportamento do esvaziamento da memória de pré-processamento: com o "FIFOCTRL" existe uma redução crescente da velocidade de percurso a partir de um nível de enchimento de 2/3, para evitar um esvaziamento total e a desaceleração até a imobilização total (parada).

Parada de pré-processamento

A preparação e o armazenamento temporário dos blocos são parados, se no bloco estiver programado o comando "STOPRE". O bloco seguinte somente será executado se todos

blocos anteriormente pré-processados e armazenados foram totalmente executados. O bloco anterior é parado na parada exata (como o G9).

ATENÇÃO
Interrupção do programa Com a compensação de ferramenta ativada e em interpolações de Spline não deve ser programado nenhum "STOPRE", pois as sequências de blocos associadas seriam interrompidas.

Sintaxe

Tabelas 12-1 Marcação do segmento de processamento:

```
STOPFIFO
...
STARTFIFO
```

Tabelas 12-2 Controle automático de memória de pré-processamento:

```
...
FIFOCTRL
...
```

Tabelas 12-3 Parada de pré-processamento:

```
...
STOPRE
...
```

Indicação

Os comandos "STOPFIFO", "STARTFIFO", "FIFOCTRL" e "STOPRE" devem ser programados em um bloco próprio.

Significado

STOPFIFO:	O "STOPFIFO" marca o início de um segmento de usinagem, que deve ser armazenado temporariamente na memória de pré-processamento. Com o "STOPFIFO", a usinagem é parada e a memória de pré-processamento é abastecida até: • que seja identificado "STARTFIFO" ou "STOPRE" ou • a memória de pré-processamento estiver cheia ou • o fim do programa for alcançado.
STARTFIFO:	Com o "STARTFIFO" é iniciada uma execução rápida do segmento de usinagem, paralelamente a isto ocorre o abastecimento da memória de pré-processamento
FIFOCTRL:	Ativação do controle automático de memória de pré-processamento
STOPRE:	Parada do pré-processamento

Indicação

O abastecimento da memória de pré-processamento não é executado ou é interrompido, se o segmento de processamento conter comandos que forçam uma operação sem memória temporária (aproximação do ponto de referência, funções de medição, …).

Indicação

Ao acessar dados de estado da máquina ($SA...) o comando numérico gera uma parada interna do pré-processamento.

Exemplo: Parada do pré-processamento

Código de programa	**Comentário**
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	; Bloco de medição com apalpador da primeira entrada de medição e interpolação linear.
N40 STOPRE	; Parada de pré-processamento.
...	

12.7 Definir áreas de stop-delay (DELAYFSTON, DELAYFSTOF)

Para definir uma área de stop-delay, condicionalmente sujeita a interrupção, no programa de peças (área de stop-delay) aplicam-se os procedimentos predefinidos DELAYFSTON e DELAYFSTOF.

Indicação

Nas ações sincronizadas DELAYFSTON e DELAYFSTOF **não** são permitidos!

Sintaxe

```
DELAYFSTON
...
DELAYFSTOF
```

Significado

DELAYFSTON:	Definir o início de uma área de stop-delay	
	Sozinho no bloco:	sim
DELAYFSTOF:	Definição do fim de uma área Stop-Delay	
	Sozinho no bloco:	sim

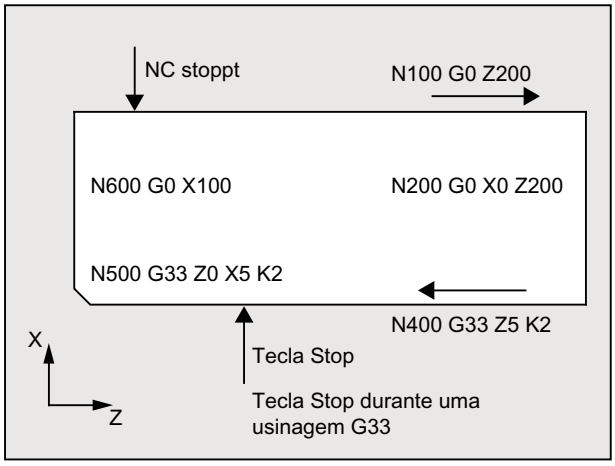
Exemplo de programação

Em um loop é repetido o seguinte bloco de programa:

Código de programa

```
...
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
...
```

Na seguinte figura é visível que o usuário pressiona "Stop" na área de stop-delay, e NC inicia o processo de frenagem fora da área de stop-delay, ou seja, no bloco N100. Com isso o NC é parado na área anterior do N100.



Outras informações

Fim de sub-rotina

Com o fim da subrotina onde foi chamado o DELAYFSTON, se ativa implicitamente o DELAYFSTOF.

Aninhamento

Se o subprograma 1 chamar o subprograma 2 em uma área Stop-Delay, então o subprograma 2 passa a ser uma completa área Stop-Delay. Especialmente o DELAYFSTOF não tem efeito no subprograma 2.

Exemplo:

```
Código de programa          Comentário
N10010 DELAYFSTON           ; Blocos com N10xxx do nível de programa 1.
N10020 R1 = R1 + 1
N10030 G4 F1                ; É iniciada a área Stop-Delay.
...
N10040 Subrotina2
...
...                         ; Interpretação da sub-rotina 2.
N20010 DELAYFSTON           ; Inativo, início repetido, 2. Nível.
...
N20020 DELAYFSTOF           ; Inativo, fim em outro plano.
N20030 RET
N10050 DELAYFSTOF           ; Fim da área de stop-delay no mesmo plano.
...
N10060 R2 = R2 + 2
N10070 G4 F1                ; A área Stop-Delay termina. A partir de agora as para-
                              das tem efeito imediato.
```

Variáveis de sistema

Para ver se no momento da execução o programa de peça está em uma seção de Stop-Delay, é possível consultar usando as seguintes variáveis do sistema:

- no programa de peça com $P_DELAYFST
- em ações sincronizadas com $AC_DELAYFST

Valor	Significado
0	Área de stop-delay não está ativa
1	Área de stop-delay está ativa

12.8 Evitar posição de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK)

Para determinadas situações mecanicamente complicadas da máquina se faz necessário evitar a localização de blocos SERUPRO.

Com um indicador de interrupção programável existe a possibilidade de intervenção, na "Localização na posição de interrupção", antes da posição que não deve ser pesquisada.

Também se pode definir áreas do programa de peça que não devem ser pesquisadas, onde o NC não deve entrar novamente. Com o cancelamento do programa o NC marca o último bloco processado que pode ser localizado através da interface do operador HMI.

Sintaxe

```
IPTRLOCK
IPTRUNLOCK
```

Os comandos estão isolados em uma linha do programa de peça e permitem um indicador de interrupção programável

Significado

`IPTRLOCK`:	Início do segmento de programa que não deve ser pesquisado
`IPTRUNLOCK`:	Fim do segmento de programa que não deve ser pesquisado

Os dois comandos somente estão em programas de peça, mas **não** são permitidos em ações sincronizadas.

Exemplo

Aninhamento de segmentos de programa que não devem ser pesquisados em dois níveis de programa com "IPTRUNLOCK" implícito. O "IPTRUNLOCK" implícito na sub-rotina 1 finaliza a área que não deve ser pesquisada.

Código de programa	**Comentário**
`N10010 IPTRLOCK()`	
`N10020 R1 = R1 + 1`	
`N10030 G4 F1`	`; Bloco de parada que inicia o segmento de programa não apto a pesquisa.`
`...`	
`N10040 Subrotina2`	
`...`	`; Interpretação da sub-rotina 2.`
`N20010 IPTRLOCK ()`	`; Inativo, início repetido.`
`...`	
`N20020 IPTRUNLOCK ()`	`; Inativo, fim em outro plano.`
`N20030 RET`	
`...`	
`N10060 R2 = R2 + 2`	
`N10070 RET`	`; Fim do segmento de programa que não deve ser pesquisado.`

Código de programa	Comentário
N100 G4 F2	; O programa principal será prosseguido.

Uma interrupção em 100 retorna novamente o indicador de interrupção.

Outras informações

Detectar e localizar áreas que não devem ser pesquisadas

Os segmentos de programa que não devem ser pesquisados são identificados com os comandos de linguagem "IPTRLOCK" e "IPTRUNLOCK" .

O comando "IPTRLOCK" congela o indicador de interrupção em um processamento principal de um bloco individual (SBL1). A partir daqui este bloco será nomeado como bloco de parada. Se ocorrer um cancelamento do programa após o "IPTRLOCK", então este assim chamado bloco de parada pode ser localizado na interface do operador HMI.

Continuar novamente do bloco atual

O indicador de interrupção é posicionado no bloco atual com "IPTRUNLOCK" como o ponto de interrupção para o segmento de programa seguinte.

Após um destino de busca ser encontrado se pode realizar uma nova busca com o mesmo bloco de parada.

Um indicador de interrupção editado pelo usuário deve ser removido novamente através da HMI.

Regras de aninhamento

Os itens a seguir regularizam a ação conjunta dos comandos de linguagem "IPTRLOCK" e "IPTRUNLOCK" com aninhamentos e o fim da sub-rotina:

1. Com o fim da sub-rotina onde foi chamado o "IPTRLOCK", se ativa implicitamente o "IPTRUNLOCK".
2. O "IPTRLOCK" em uma área não apta para pesquisa permanece sem efeito.
3. Se a sub-rotina1 chama a sub-rotina2 em uma área que não deve ser pesquisada, então a sub-rotina2 inteira não será pesquisada. Especialmente o "IPTRUNLOCK" está sem efeito na sub-rotina 2.

Para mais informações sobre este assunto, veja o
/FB1/ Manual de funcionamento das funções básicas; BAG, Canal, Modo de operação do programa (K1).

Variável de sistema

Uma área que não deve ser pesquisada pode ser detectada no programa de peça com "$P_IPTRLOCK".

Indicador de interrupção automático

A função de indicador de interrupção automático define automaticamente um tipo de acoplamento definido anteriormente para localização. Através de dado de máquina, para

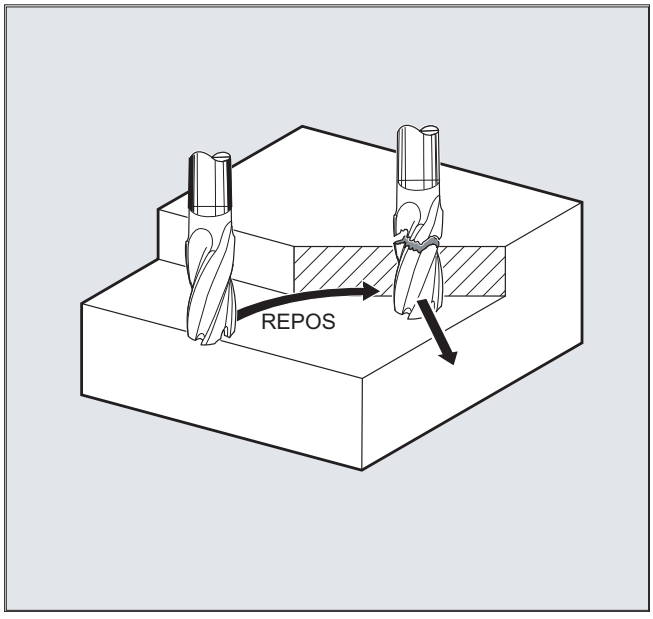
- Transmissão eletrônica com "EGON"
- Acoplamento axial do valor mestre com "LEADON"

se ativa o indicador de interrupção automático. Se o indicador de interrupção programado e o ativado automaticamente através de dado de máquina se sobreporem, então se forma a maior área possível que não deve ser pesquisada.

12.9 Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

Se o programa em execução for interrompido durante a usinagem e a ferramenta estiver sendo recuada (p. ex. por conta de uma quebra de ferramenta ou para uma eventual medição), o contorno pode ser reaproximado em um ponto selecionável de forma controlada.

O comando `REPOS` tem efeito em um ASUP, como um salto de regresso à sub-rotina (por ex., M17). Os blocos seguintes não são mais executados. Para interrupção do programa consulte "Rotina de interrupção (ASUP) (Página 132)".



Sintaxe

```
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=…
```

Significado

Seleção da abordagem

REPOSA:	Reaproximação até o contorno com os eixos geométricos em uma reta. Todos os outros eixos de canal também serão reposicionados.
REPOSL:	Reaproximação até o contorno com os eixos geométricos em uma reta. Todos os outros eixos de canal também devem ser explicitamente programados.
REPOSQ DISR=… :	Reaproximação no contorno com eixos geométricos em um quadrante com raio DISR. Todos os outros eixos de canal também devem ser explicitamente programados.
REPOSQA DISR=… :	Reaproximação no contorno com eixos geométricos em um quadrante com raio DISR. Todos os outros eixos de canal também serão reposicionados.
REPOSH DISR=… :	Reaproximação no contorno com eixos geométricos em um semicírculo com diâmetro DISR. Todos os outros eixos de canal também devem ser explicitamente programados.
REPOSHA DISR=… :	Reaproximação no contorno com eixos geométricos em um semicírculo com raio DISR. Todos os outros eixos de canal também serão reposicionados.

Seleção do novo ponto de partida

RMIBL:	Aproximar ponto de interrupção
RMIBL DISPR=…:	Ponto de entrada com distância DISPR em mm/polegadas antes do ponto de interrupção
RMBBL:	Aproximação do ponto inicial do bloco
RMEBL:	Aproximação do ponto final do bloco
RMIBL DISPR=…:	Aproximação do ponto final do bloco com distância DISPR antes do ponto final
RMNBL:	Aproximação do ponto de percurso mais próximo
A0 B0 C0 :	Eixos em que se deve realizar a aproximação

Indicação

Compatibilidade

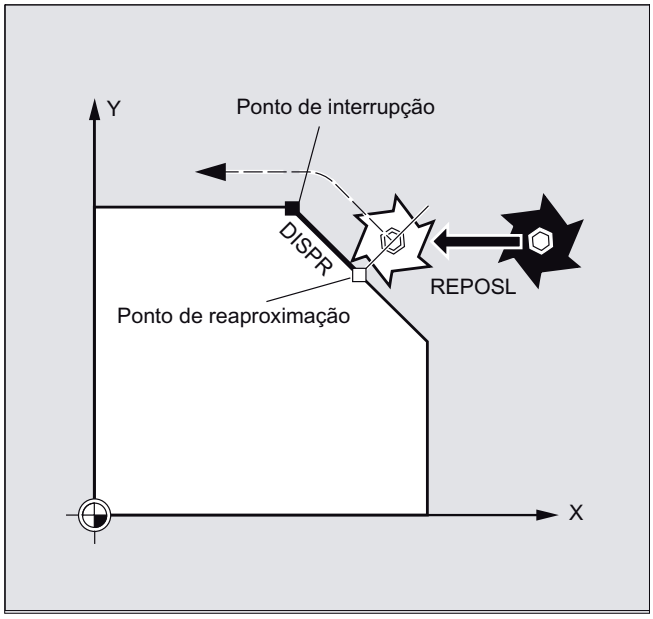
A fim de permanecer compatível com os níveis de software mais antigos, o modo de aproximação REPOS também pode continuar a ser programado por meio dos comandos modais RMI, RMB, RME e RMN. No caso de utilização dentro de um ASUP, este deve ser guarnecido com o atributo SAVE na instrução PROC. Caso contrário, o modo de aproximação REPOS utilizado de forma modal em ASUP, se ele divergir da configuração prévia RMI, também terá efeito nos processos REPOS seguintes.

Aproximação ao contorno em uma reta, REPOSA, REPOSL

A ferramenta aproxima-se do ponto de reaproximação diretamente em uma linha reta.

Exemplo

```
REPOSL RMIBL DISPR=6 F400
```

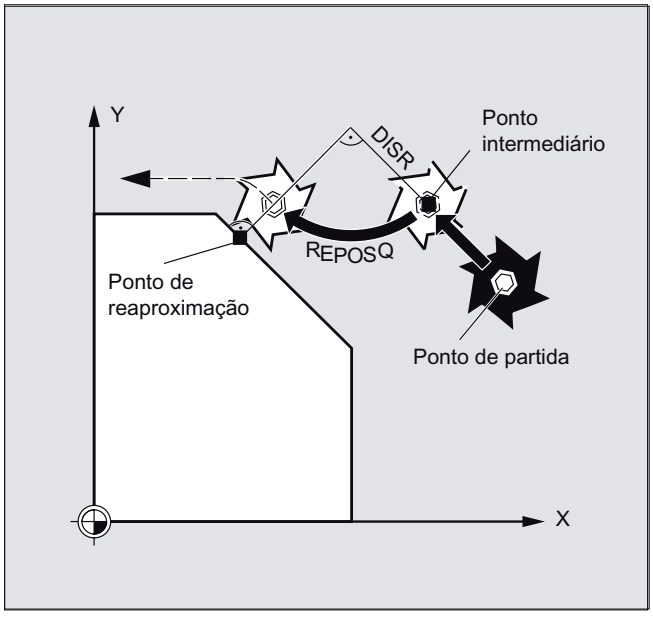


Reaproximação no contorno com quadrante, REPOSQ, REPOSQA

A ferramenta aproxima-se do ponto de reaproximação em um quadrante com o raio `DISR=...` . O comando calcula automaticamente o ponto intermediário necessário entre o ponto de partida e o ponto de reaproximação.

Exemplo

```
REPOSQ RMIBL DISR=10 F400
```

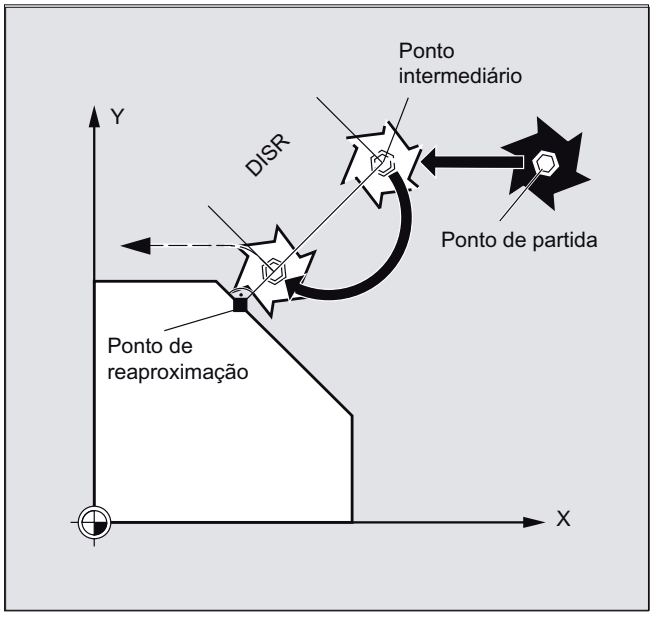


Reaproximação até o contorno com semicírculo, REPOSH, REPOSHA

A ferramenta aproxima-se do ponto de reaproximação em um semicírculo com o diâmetro `DISR=...`. O comando calcula automaticamente o ponto intermediário necessário entre o ponto de partida e o ponto de reaproximação.

Exemplo

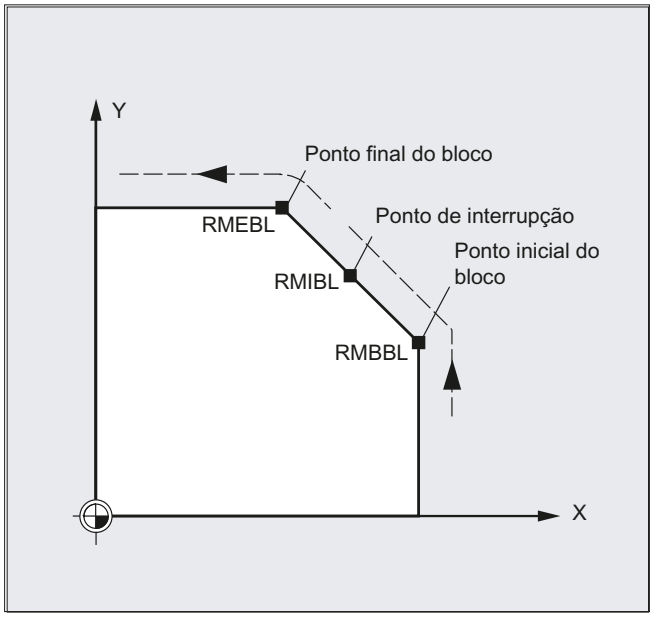
```
REPOSH RMIBL DISR=20 F400
```



Definir o ponto de reaproximação (não para aproximação SERUPRO com RMNBL)

Dependendo do bloco NC onde o processo do programa foi interrompido, pode-se escolher entre um dos três pontos de reaproximação:

- RMIBL, ponto de interrupção
- RMBBL, ponto inicial do bloco ou o último ponto final
- RMEBL, ponto final do bloco



Com RMIBL DISPR=... ou com RMEBL DISPR=... podemos definir o ponto de reaproximação que está antes do ponto de interrupção ou antes do ponto final do bloco.

Com DISPR=... descrevemos o trajeto do contorno em mm/polegadas em torno do qual está o ponto de reaproximação antes do ponto de interrupção ou o ponto final. Este ponto no máximo pode estar no ponto inicial do bloco, também para valores mais altos.

Se nenhum DISPR=... programado, aplica-se DISPR=0 e com isso programa-se o ponto de interrupção (para RMIBL) ou o ponto final do bloco (para RMEBL).

Sinal do DISPR

O sinal do DISPR é avaliado. Se o sinal for positivo permanece o procedimento descrito anteriormente.

Se o sinal for negativo a reaproximação será realizada após o ponto de interrupção e para RMBBL atrás do ponto de partida.

A distância do ponto de interrupção para o ponto de aproximação resulta do valor do DISPR. Para valores maiores, este ponto também pode estar no máximo no ponto final do bloco.

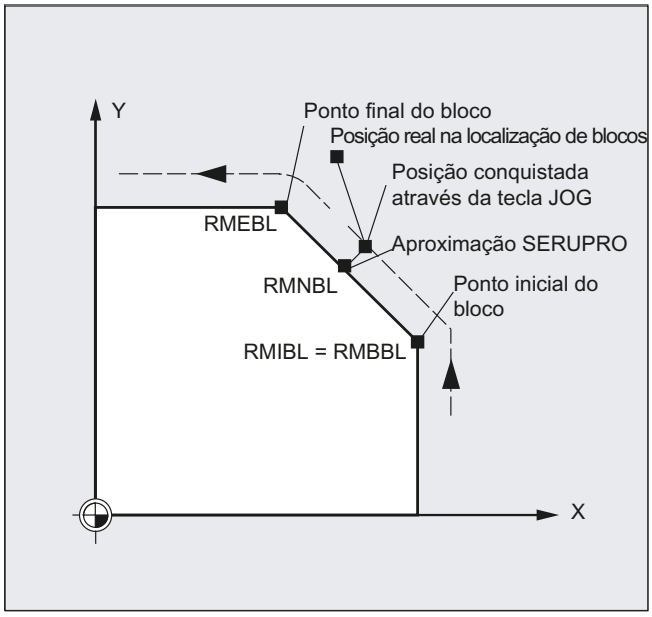
Exemplo de aplicação:

Através de um sensor se detecta a aproximação de uma garra de fixação. É iniciada uma ASUP com a qual se desvia da garra de fixação.

Em seguida é feito o reposicionamento em um ponto atrás da garra de fixação com DISPR negativo e o programa é prosseguido.

Aproximação SERUPRO com RMNBL

Se para a usinagem em uma posição qualquer for forçado um cancelamento, então será realizada a aproximação SERUPRO sob RMNBL pelo curso mais curto até o ponto de interrupção, depois será executado apenas o curso restante. Para isso o usuário inicia um processo SERUPRO no bloco de interrupção e realiza o posicionamento através das teclas JOG antes do ponto danificado do bloco de destino.



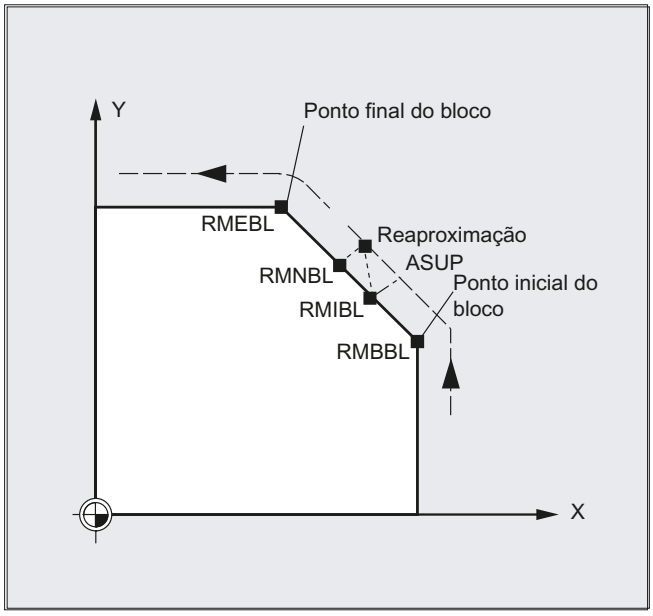
Indicação

SERUPRO

Para SERUPRO o RMIBL e o RMBBL são idênticos. O RMNBL não é restrito apenas ao SERUPRO, mas sim geralmente aplicado.

Aproximação do ponto de percurso mais próximo RMNBL

Quando o REPOSA é interpretado, após uma interrupção, o bloco de reaproximação com RMNBL não é completamente reiniciado, apenas executa-se o curso restante. É realizada a aproximação do ponto de percurso mais próximo do bloco interrompido.



Estado do modo REPOS válido

O modo REPOS válido do bloco interrompido pode ser lido através de ações sincronizadas com a variável $AC_REPOS_PATH_MODE:

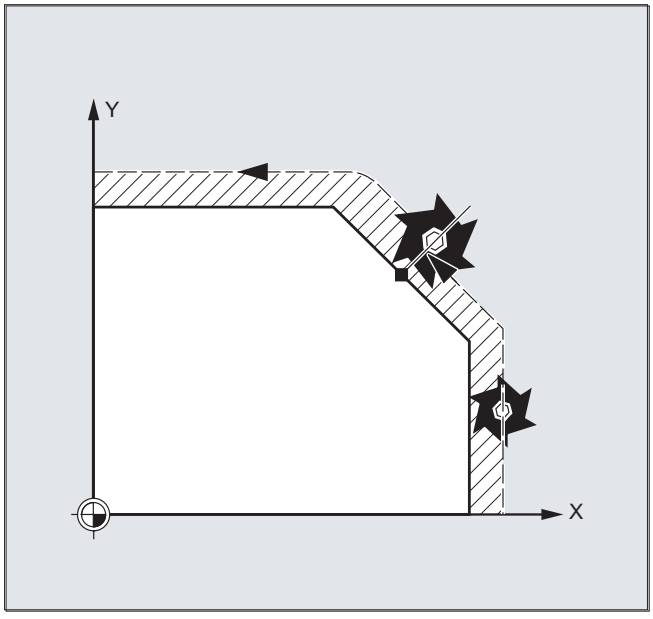
0 Aproximação não definida
1 RMBBL: Aproximação no início
2 RMIBL: Aproximação no ponto de interrupção
3 RMEBL: Aproximação no ponto final do bloco
4 RMNBL: Aproximação no ponto de percurso mais próximo do bloco interrompido

Aproximação com nova ferramenta

Se interrompemos a execução do programa devido uma quebra de ferramenta:

Com a programação do novo número D, o programa será continuado a partir do ponto de reaproximação com os valores alterados da correção da ferramenta.

No caso de valores alterados da correção da ferramenta, o ponto de interrupção possivelmente não poderá mais ser aproximado. Neste caso é aproximado o ponto do novo contorno mais próximo do ponto de interrupção (eventualmente modificado pelo DISPR).



Aproximar contorno

O movimento com que a ferramenta é reaproximada no contorno pode ser programado. Os endereços dos eixos de movimento indicam o valor zero.

Com os comandos REPOSA, REPOSQA e REPOSHA todos eixos são reposicionados automaticamente. Não será necessário especificar nenhum eixo.

Com a programação do REPOSL, REPOSQ e REPOSH todos os eixos geométricos são aproximados automaticamente, portanto, também sem a especificação no comando. Todos demais eixos devem ser especificados no comando.

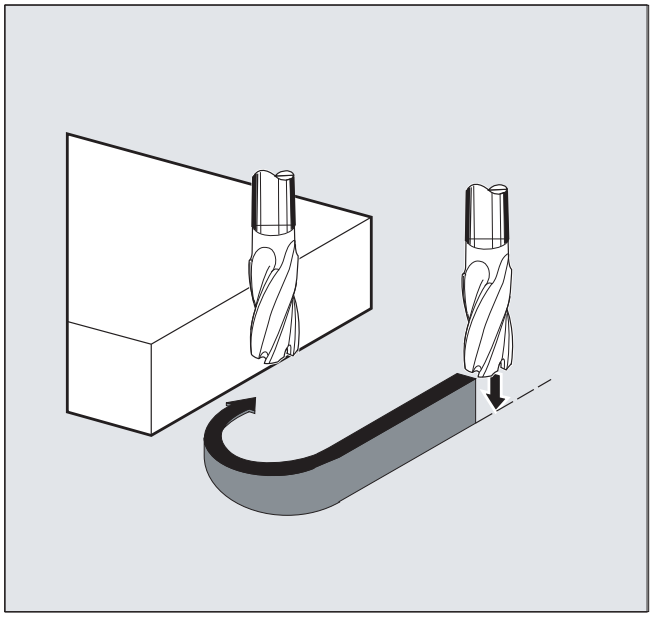
Para os movimentos circulares REPOSH e REPOSQ aplica-se:

O círculo é percorrido com o plano de trabalho especificado G17 até G19 .

Se for especificado o terceiro eixo geométrico (sentido da penetração) no bloco de aproximação, então para este caso o ponto de reaproximação é aproximado em uma linha espiral, onde a posição da ferramenta e a posição programada no sentido de avanço não coincidem.

Nos seguintes caso será comutado automaticamente em aproximação linear REPOSL:

- Se não foi especificado nenhum valor para DISR;.
- Não existe nenhum sentido de aproximação definido (interrupção de programa em um bloco sem informação de movimento).
- Para sentido de aproximação perpendicular ao atual plano de trabalho.



12.10 Controle na condução do movimento

12.10.1 Correção percentual de solavancos (JERKLIM)

Com o comando NC "JERKLIM" é possível que o solavanco máximo permitido de um eixo ajustado por meio do dado da máquina seja reduzido ou acelerado no movimento de percurso em seções de programa críticas.

Pré-requisito

O modo de aceleração SOFT deverá estar ativo.

Efeito

A função tem efeito:

- em modos de operação AUTOMÁTICO.
- somente em eixos de percurso.

Sintaxe

```
JERKLIM[<eixo>]=<valor>
```

Significado

JERKLIM:	Comando para correção de solavancos	
<eixo>:	Eixo da máquina cujo valor limite de solavanco deve ser adaptado.	
<valor>:	Valor de correção percentual, relacionado ao solavanco de eixo máximo configurado no movimento de percurso (MD32431 $MA_MAX_AX_JERK).	
	Faixa de valores:	1 ... 200
	O valor 100 não tem nenhuma influência sobre o solavanco.	

Indicação

O comportamento do JERKLIM no fim do programa de peça e no Reset de canal é configurado com Bit 0 no dado de máquina MD32320 $MA_DYN_LIMIT_RESET_MASK:

- Bit 0 = 0:
  O valor programado para o JERKLIM é resetado em 100 % com o Reset de canal/M30.
- Bit 0 = 1:
  O valor programado para JERKLIM é preservado além do Reset de canal/M30.

Exemplo

Código de programa	Comentário
...	

Código de programa	Comentário
`N60 JERKLIM[X]=75`	`; O carro de eixo no sentido X deve ser acelerado/desacelerado apenas com o máximo de 75% do solavanco permitido para o eixo.`
`...`	

12.10.2 Correção percentual da velocidade (VELOLIM)

com o comando VELOLIM é possível reduzir, no programa de usinagem ou na ação síncrona, a velocidade máxima possível de um eixo ou a rotação máxima de um fuso programado no dado da máquina.

Efeito

A função tem efeito:

- em modos de operação AUTOMÁTICO.
- em eixos de percurso e eixos de posicionamento.
- em fusos em modo de fuso/eixo

Sintaxe

```
VELOLIM[<eixo/fuso>]=<valor>
```

Significado

`VELOLIM`:	Comando para correção da velocidade
`<eixo/fuso>`:	Eixo ou fuso, cujo limite de velocidade ou rotação deve ser adaptado. **VELOLIM para fusos** Através do dado da máquina (MD30455 $MA_MISC_FUNCTION_MASK, Bit 6) é possível definir no programa de peças, se a programação do "VELOLIM" deve atuar independentemente da utilização atual como fuso ou eixo (Bit 6 = 1) ou se ele pode ser programado separadamente para cada modo de operação (Bit 6 = 0). Se for projetado um efeito separado, então a seleção é realizada na programação através do identificador: • Identificador de fuso `S<n>` para modos de operação de fuso • Identificador de eixo, p. ex. "`C`", para o modo de eixo

<valor>:	Valor de correção percentual O valor de correção refere-se: • Em eixos / fusos na usinagem de eixo (MD30455, Bit 6 == 0): na velocidade de eixo máxima projetada (MD32000 $MA_MAX_AX_VELO). • Em fusos na usinagem de fuso ou de eixo (MD30455 Bit 6 == 1): na rotação máxima da gama de velocidade ativa (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])	
	Faixa de valores:	1 ... 100
	O valor 100 não tem nenhuma influência sobre a velocidade e rotação.	

Indicação

Comportamento no fim do programa de peça e Reset de canal

É possível ajustar a ação do "VELOLIM" no dado da máquina, ao final do programa de peças no reset do canal: MD32320 $MA_DYN_LIMIT_RESET_MASK, Bit 0

Reconhecimento de um limite ativo de rotação na usinagem do fuso

É possível identificar um limite de rotação através do "VELOLIM" (inferior a 100 %) em operação de fuso através das seguintes variáveis de sistema:

- $AC_SMAXVELO (rotação máxima possível do fuso)
- $AC_SMAXVELO_INFO (Identificação do motivo de limitação da rotação)

Exemplos

Exemplo 1: Limitação de velocidade de eixos de máquina

Código de programa	Comentário
...	
N70 VELOLIM[X]=80	; O carro de eixo no sentido X somente deve ser movimentado com o máximo de 80% de velocidade permitida para o eixo.
...	

Exemplo 2: Limite de rotação do fuso

Código de programa	Comentário
N05 VELOLIM[S1]=90	; Limitação da rotação máxima do fuso 1 em 90% de 1000 rpm.
...	
N50 VELOLIM[C]=45	; Limitação da rotação em 45% de 1000 rpm, C é o identificador de eixo do S1.
...	

Ajuste dos dados da máquina para Fuso 1 (AX5)

- rotação máxima do nível de transmissão 1 = 1000 U/min:
  MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
- A programação de "VELOLIM" apresenta um efeito comum para a operação do fuso e do eixo independente do identificador programado:
  MD30455 $MA_MISC_FUNCTION_MASK[AX5], Bit 6 = 1

12.10.3 Exemplo de programa para JERKLIM e VELOLIM

O programa a seguir apresenta um exemplo de aplicação para a limitação em porcentagem do número de solavancos e da velocidade:

Código de programa	**Comentário**
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20	
ACC[Y]=30	
N1200 G1 Y20 VELOLIM[X]=5	; O carro de eixo no sentido X somente deve ser movimentado com o máx. 5% da velocidade permitida para o eixo.
JERKLIM[Y]=200	; O carro de eixo no sentido Y deve ser acelerado/desacelerado no máx. com 200% do solavanco permitido para o eixo.
N1300 G1 X0 JERKLIM[X]=2	; O carro de eixo no sentido X deve ser acelerado/desacelerado apenas como máx. 2% do solavanco permitido para o eixo.
N1400 G1 Y0	
M30	

12.11 Programar tolerância de contorno/orientação (CTOL, OTOL, ATOL)

Com os endereços CTOL, OTOL e ATOL podem ser adaptadas as tolerâncias de usinagem, parametrizadas por meio de dados de máquina e de ajuste para funções de compressor, para suavização e suavização de orientação no programa de peças.

Os valores de tolerância programados permanecem válidos até nova programação ou eliminação através de atribuição de um valor negativo. Também são eliminados no final do programa ou reset. Após a eliminação, novamente os valores parametrizados são efetivos.

Sintaxe

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

Significado

CTOL:	Endereço para programação da **tolerância de contorno**			
	Área de Aplicação:	• todas funções de compressor • todos tipos de suavização exceto G641 e G644		
	Parada de pré-processamento:	não		
	Efeito:	modal		
	<Value>:	O valor para a tolerância de contorno é uma especificação de comprimento.		
		Tipo:	REAL	
		Unidade:	Inch/mm (em função do atual ajuste da indicação de dimensões)	
		Faixa de valores:	≥ 0:	Valor de tolerância
			< 0:	Eliminação do valor de tolerância programado ⇒ Vale novamente o valor de tolerância parametrizado no dado de máquina ou no dado de ajuste.

OTOL:	Endereço para programação da **tolerância de orientação**			
	Área de Aplicação:	• todas funções de compressor • Suavização de orientação ORISON • todos os tipos de suavização exceto G641 e G644 e OSD		
	Parada de pré-processamento:	não		
	Efeito:	modal		
	<Value>:	O valor para a tolerância de orientação é uma especificação angular.		
		Tipo:	REAL	
		Unidade:	Graus	
		Faixa de valores:	≥ 0:	Valor de tolerância
			< 0:	Eliminação do valor de tolerância programado ⇒ Vale novamente o valor de tolerância parametrizado no dado de máquina ou no dado de ajuste.
ATOL:	Endereço para programação de uma **tolerância específica de eixo**			
	Área de Aplicação:	• todas funções de compressor • Suavização de orientação ORISON • todos os tipos de suavização exceto G641 e G644 e OSD		
	Parada de pré-processamento:	não		
	Efeito:	modal		
	<Axis>:	Nome do eixo de canal para o qual a tolerância programada deve atuar		
	<Value>:	O valor para a tolerância de eixo pode ser uma especificação de comprimento ou angular dependendo do tipo de eixo (eixo linear ou rotativo).		
		Tipo:	REAL	
		Unidade:	para eixos lineares:	Inch/mm (em função do atual ajuste da indicação de dimensões)
			para eixos rotativos:	Graus
		Faixa de valores:	≥ 0:	Valor de tolerância
			< 0:	Eliminação do valor de tolerância programado ⇒ Vale novamente o valor de tolerância parametrizado no dado de máquina ou no dado de ajuste.

Indicação

Os valores de tolerância específicos do canal programados com CTOL e OTOL têm prioridade superior que os valores de tolerância específicos do eixo programados com ATOL

Indicação

Frames escalonáveis

Os Frames escalonáveis atuam sobre as tolerâncias programadas da mesma forma como nas posições de eixo, isto é, a tolerância relativa permanece a mesma.

Exemplo

Código de programa	Comentário
COMPCAD G645 G1 F10000	; Ativar a função de compressor COMPCAD.
X... Y... Z...	; Aqui atuam os dados de máquina e de configuração.
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; A partir deste ponto atua a tolerância de contorno de 0,02 mm.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
ASCALE X0.25 Y0.25 Z0.25	; A partir deste ponto atua a tolerância de contorno de 0,005 mm.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=-1	; A partir destes ponto atuam novamente os dados de máquina e de configuração.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Variáveis de sistema

Leitura com parada de pré-processamento

Através das seguintes variáveis de sistema é possível ler as tolerâncias atualmente válidas no programa de peças e ação síncrona:

- $AC_CTOL
  Tolerância de contorno específica do canal que estava efetiva durante a preparação do atual bloco de processamento principal
  Sem nenhuma tolerância de contorno efetiva, $AC_CTOL fornece a raiz da soma dos quadrados das tolerâncias dos eixos geométricos.
- $AC_OTOL
  Tolerância de orientação específica do canal, que estava efetiva durante a preparação do atual bloco de processamento principal
  Sem nenhuma tolerância de orientação efetiva, $AC_OTOL fornece, durante uma transformação de orientação ativa, a raiz da soma dos quadrados das tolerâncias dos eixos de orientação, caso contrário, o valor "-1".
- $AA_ATOL[<eixo>]
  Tolerância de contorno específica do eixo, que estava efetiva durante a preparação do atual bloco de processamento principal.
  Com uma tolerância de contorno ativa, $AA_ATOL[<eixo geométrico>] fornece a tolerância de contorno dividida pela raiz do número de eixos geométricos.
  Com uma tolerância de orientação e uma transformação de orientação ativas, $AA_ATOL[<eixo de orientação>] fornece a tolerância de orientação dividida pela raiz do número de eixos de orientação.

Indicação

Se não houve programação de valores de tolerância, as variáveis $A não são suficientemente diferenciadas para distinguir as tolerâncias das diferentes funções.

Tais casos podem ocorrer quando os dados de máquina e de ajuste ajustam diferentes tolerâncias para funções de compressor, suavização e suavização de orientação. Neste caso, as variáveis de sistema fornecem o maior valor presente nas funções ativas no momento. Quando, por ex., uma função de compressor está ativa com a tolerância de orientação 0,1° e uma suavização de orientação ORISON está ativa com 1°, $AC_OTOL fornece o valor "1". Ao desligar a suavização de orientação $AC_OTOL fornece o valor "0,1".

Leitura sem parada de pré-processamento

Através das seguintes variáveis de sistema é possível ler as tolerâncias atualmente válidas no programa de peças:

- $P_CTOL
  Tolerância de contorno específica do canal atualmente efetiva.
- $P_OTOL
  Tolerância de orientação específica do canal atualmente efetiva.
- $PA_ATOL
  Tolerância de contorno específica do eixo atualmente efetiva.

Condições gerais

As tolerâncias programadas com CTOL, OTOL e ATOL atuam também sobre funções indiretamente dependentes destas tolerâncias:

- A limitação do erro de corda no cálculo do valor nominal
- A função básica do modo de superfície de forma de alívio

As seguintes funções de suavização **não** são influenciadas pela programação de CTOL, OTOL e ATOL:

- Suavização da orientação com OSD
  OSD não utiliza nenhuma tolerância, mas sim uma distância para a transição do bloco.
- Suavização com G644
  G644 não atua na usinagem, mas sim na otimização das trocas de ferramentas e outros movimentos sem usinagem.
- Suavização com G645
  O comportamento de G645 é quase sempre igual ao de G642 e, portanto, utiliza as tolerâncias programadas. Somente nas transições de bloco de tangentes constantes com um salto de curvatura, por ex. uma transição tangencial círculo-reta, é utilizado o valor de tolerância a partir do dado de máquina MD33120 $MA_PATH_TRANS_POS_TOL. O trajeto da suavização nestes pontos também pode estar localizado no lado externo do contorno programado, onde muitas aplicações são menos tolerantes. Além disso, normalmente uma pequena tolerância, ajustada de maneira fixa, é suficiente para compensar as alterações de curvatura, que não precisam causar preocupações ao programador NC.

12.12 Ação de mudança de bloco no acoplamento ativo (CPBC)

Com o comando CPBC, o critério da mudança de bloco a ser preenchido é dado e uma mudança de bloco no programa de usinagem com acoplamento ativo é realizada.

Sintaxe

```
CPBC[<Folgeachse>] = <Critério>
```

Significado

CPBC:	Critério de troca dos blocos em acoplamento ativo	
<eixo escravo>:	Identificador de eixo do eixo escravo	
<Critério>:	Critério de mudança de blocos	
	Tipo:	STRING
	Valor	**Significado:** A mudança de bloco ocorre
	"NOC"	independente do estado do acoplamento
	"IPOSTOP"	na sincronização de valor nominal
	"COARSE"	na sincronização de valor real "amplo"
	"FINE"	na sincronização de valor real "fino"

Exemplo

Código de programa

```
; A mudança de bloco ocorre em:
; -Acoplagem para eixo escravo X2 == ativo
; -sincronização de valor nominal == ativo
CPBC[X2]="IPOSTOP"
```

Acoplamentos de eixo 13

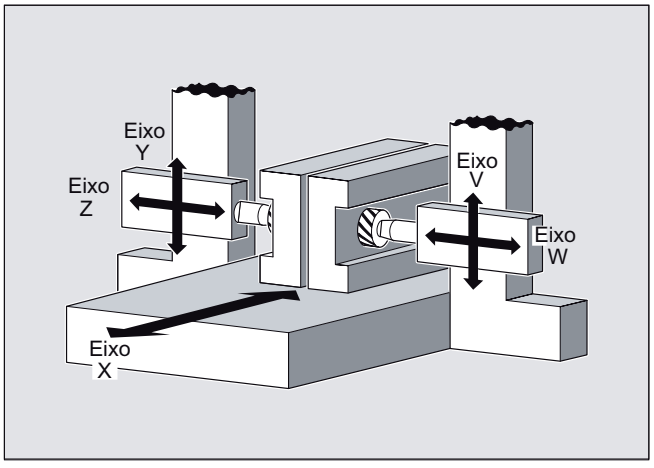
13.1 Movimento acoplado (TRAILON, TRAILOF)

Ao ser movimentado um eixo mestre definido, os eixos acoplados (= eixos escravos) à ele associados são movimentados nos percursos descritos pelo eixo mestre, sob a consideração de um fator de acoplamento.

Eixo mestre e eixos escravos juntos formam um grupo de movimento acoplado.

Áreas de aplicação

- Movimento de um eixo através de um eixo simulado. O eixo mestre é um eixo simulado e o eixo acoplado um eixo real. Com isso o eixo real pode ser movimentado sob a consideração de um fator de acoplamento.
- Usinagem bilateral com 2 grupos de movimento acoplado:
  1. Eixo mestre Y, eixo acoplado V
  2. Eixo mestre Z, eixo acoplado W



Sintaxe

```
TRAILON(<eixo escravo>,<eixo mestre>,<fator de acoplamento>)
TRAILOF(<eixo escravo>,<eixo mestre>,<eixo escravo 2>)
TRAILOF(<eixo escravo>)
```

Significado

TRAILON:	Comando para ativar e definir um grupo de movimento acoplado	
	Efeito:	modal

`<eixo escravo>`:	Parâmetro 1: Denominação do eixo de movimento acoplado **Nota:** Um eixo de movimento acoplado também pode ser eixo mestre para outros eixos acoplados. Desse modo podem ser construídos diferentes grupos de movimento acoplado.
`<eixo mestre>`:	Parâmetro 2: Denominação do eixo mestre
`<fator de acoplamento>`:	Parâmetro 3: Fator de acoplamento O fator de acoplamento estabelece a relação desejada dos cursos do eixo acoplado e eixo mestre: <fator de acoplamento> = Curso do eixo acoplado/curso do eixo mestre
	Tipo: REAL
	Pré-ajuste: 1
	A especificação de um valor negativo desencadeia um movimento inverso de percurso dos eixos mestres e acoplados. Se o fator de acoplamento não for especificado na programação, então será aplicado automaticamente o fator de acoplamento 1.
`TRAILOF`:	Comando para desativar um grupo de movimento acoplado
	Efeito: modal

	O `TRAILOF` com 2 parâmetros apenas desativa o acoplamento para o eixo mestre indicado:
	`TRAILOF(<eixo escravo>,<eixo mestre>)`
	Quando um eixo acoplado possui 2 eixos mestres, para desativação dos dois acoplamentos o `TRAILOF` pode ser chamado com 3 parâmetros:
	`TRAILOF(<eixo escravo>,<eixo mestre>,<eixo escravo 2>)`
	O mesmo resultado é oferecido pela programação do `TRAILOF` sem a indicação de um eixo mestre:
	`TRAILOF(<eixo escravo>)`

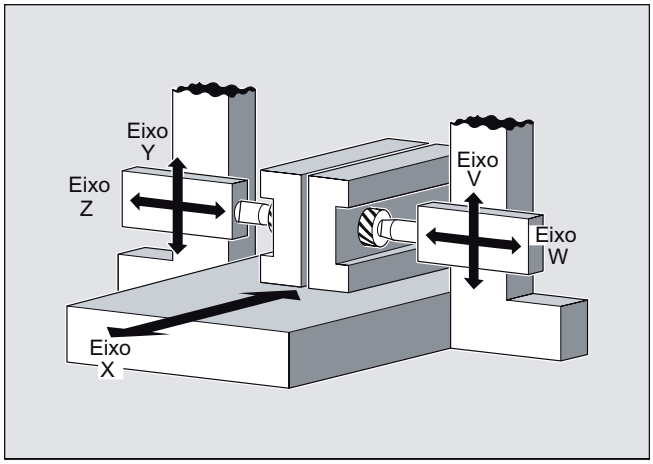
Indicação

O movimento acoplado sempre é realizado no sistema de coordenadas básico (BCS).

O número de grupos de movimento acoplado ativados simultaneamente apenas é limitado pelas opções de combinação dos eixos disponíveis na máquina.

Exemplo

A peça de trabalho deve ser usinada nos dois lados com o conjunto de eixos mostrado. Para isso crie 2 grupos de movimento acoplado.



Código de programa	**Comentário**
…	
N100 TRAILON(V,Y)	; Ativação do 1° grupo de movimento acoplado
N110 TRAILON(W,Z,-1)	; Ativação do 2° grupo de movimento acoplado. Fator negativo de acoplamento: O eixo acoplado movimenta-se no sentido oposto ao eixo mestre.
N120 G0 Z10	; Penetração do eixo Z e eixo W no sentido oposto ao eixo.
N130 G0 Y20	; Penetração do eixo Y e eixo V no mesmo sentido do eixo.
…	
N200 G1 Y22 V25 F200	; Sobreposição de um movimento dependente e um independente do eixo acoplado V.
…	
TRAILOF(V,Y)	; Desativação do 1° grupo de movimento acoplado.
TRAILOF(W,Z)	; Desativação do 2° grupo de movimento acoplado.

Outras informações

Tipos de eixos

Um grupo movimento acoplado pode conter um número qualquer de combinações de eixos lineares e rotativos. Como eixo mestre também se pode definir um eixo simulado.

Eixos acoplados

Um eixo acoplado pode ser atribuído no máximo à 2 eixos mestres simultaneamente. A atribuição é realizada em diferentes grupos de movimento acoplado.

Um eixo acoplado pode ser programado com todos comandos de movimento disponíveis (G0, G1, G2, G3, …). Além do curso independente definido, o eixo acoplado percorre os cursos derivados de seus eixos mestres com os fatores de acoplamento.

Limitação de dinâmica

A limitação de dinâmica depende do tipo de ativação do grupo de movimento acoplado:

- Ativação no programa de peça
  Se a ativação é realizada no programa de peça e todos os eixos mestres são eixos de programa no canal ativado, a dinâmica de todos eixos acoplados será considerada durante o deslocamento dos eixos mestres, para que nenhum eixo acoplado seja sobrecarregado. Se a ativação é realizada no programa de peça com eixos mestres, que não estão ativos como eixos de programa no canal ativo ($AA_TYP ≠ 1), a dinâmica do eixo acoplado não será considerada durante o deslocamento dos eixos mestres. Com isso pode ocorrer uma sobrecarga nos eixos acoplados com uma dinâmica um pouco menor do que a necessária para o acoplamento.
- Ativação na ação síncrona
  Se a ativação é realizada em uma ação síncrona, a dinâmica dos eixos acoplados não será considerada durante o deslocamento dos eixos mestres. Com isso pode ocorrer uma sobrecarga nos eixos acoplados com uma dinâmica um pouco menor do que a necessária para o acoplamento.

⚠ CUIDADO

Sobrecarga no eixo

Quando um grupo de movimento acoplado

- em ações síncronas
- no programa de peça com eixos mestres, que não são eixos de programa no canal do eixo acoplado,

é ativado, então o usuário/fabricante da máquina tem uma responsabilidade especial para empregar medidas adequadas, para que não ocorra nenhuma sobrecarga dos eixos acoplados através dos movimentos do eixo mestre.

Estado do acoplamento

O estado de acoplamento de um eixo pode ser consultado no programa de peça com as variáveis de sistema:

$AA_COUP_ACT[<eixo>]

Valor	Significado
0	Nenhum acoplamento ativo
8	Movimento acoplado ativo

Trajetória restante do eixo de movimento acoplado no módulo de eixo rotativo.

São os módulos dos eixos rotativos, eixos-mestre ou de movimento acoplado, soma-se o movimento rotacional ao eixo-mestre de n * 360° com n = 1, 2, 3... , na trajetória restante do eixo de movimento acoplado até a desativação do acoplamento.

Exemplo: Parte do programa com TRAILON e eixo-mestre B e eixo escravo C

```
Código de programa              Comentário
TRAILON(C,B,1)                  ; Ativar acoplamento.
G0 B0                           ; Posição de saída
                                ; Trajetória restante no início do bloco:
```

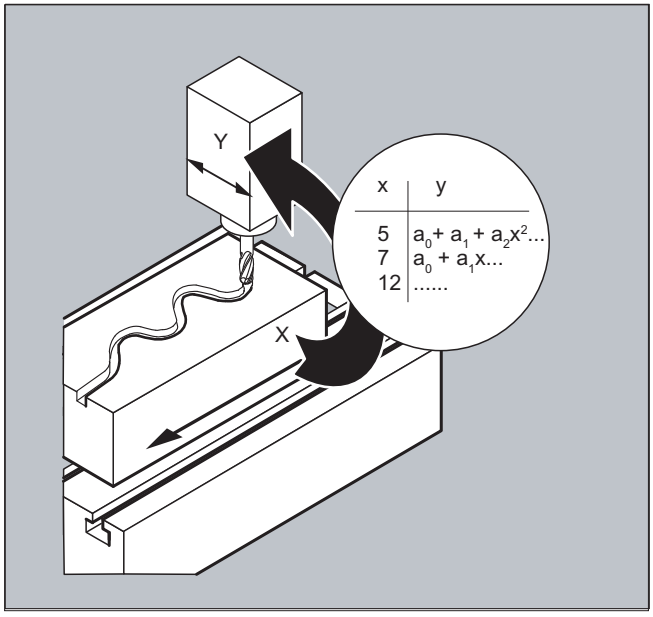
Código de programa	**Comentário**
G91 B360	; B=360, C=360
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

13.2 Tabelas de curvas (CTAB)

Com a ajuda das tabelas de curvas é possível programar relações de posição e velocidade entre dois eixos (eixo mestre e eixo escravo). A definição da tabela de curvas é realizada no programa de peça.

Aplicação

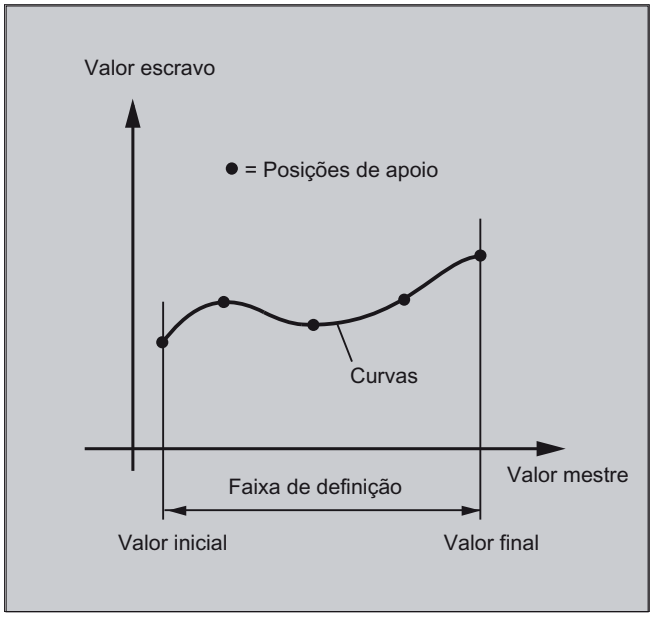
As tabelas de curvas substituem os discos de cames. Neste caso a tabela de curvas forma a base para o acoplamento axial de valor mestre, no qual se estabelece a relação funcional entre valor mestre e valor escravo: Conforme a programação o comando calcula, a partir das posições atribuídas para eixo mestre e eixo escravo, um polinômio que corresponde ao disco de cames.



13.2.1 Definição de tabelas de curvas (CTABDEF, CATBEND)

Uma tabela de curvas representa um programa de peça ou um segmento do programa de peça que está marcado no início com o comando CTABDEF e no fim com o CTABEND.

Neste segmento do programa de peça são associadas posições únicas de eixos escravos para cada posição do eixo mestre através de instruções de movimento, e estas posições servem de pontos de apoio para o cálculo da curva em forma de um polinômio de até 5º grau.



Pré-requisito

Para a definição de tabelas de curvas deve ser reservado o espaço de memória necessário através da configuração de dado de máquina adequada (→ Fabricante da máquina!).

Sintaxe

```
CTABDEF(<eixo escravo>,<eixo mestre>,<n>,<periodicidade>[,<local de
armazenamento>])
...
CTABEND
```

Significado

`CTABDEF( )`:	Início da definição de tabela de curvas	
`CTABEND`:	Fim da definição de tabela de curvas	
`<eixo escravo>`:	Eixo, cujo movimento deve ser calculado através da tabela de curvas	
`<eixo mestre>`:	Eixo, que fornece os valores mestres para o cálculo do movimento do eixo escravo	
`<n>`:	Número (**ID**) da tabela de curvas O número de uma tabela de curvas é único e independe do local de armazenamento. Não pode existir nenhuma tabela de mesmo número na memória estática e na memória dinâmica do NC.	
`<Periodicidade>`:	Periodicidade da tabela	
	`0`	A tabela não é periódica (é processada apenas uma vez, mesmo para eixos rotativos)
	`1`	A tabela é periódica relativa ao eixo mestre
	`2`	A tabela é periódica relativa ao eixo mestre e ao eixo escravo

<Local de armazenamento>:	Especificação do local de armazenamento (opcional)	
	"SRAM"	A tabela de curvas é criada na memória **estática** do NC.
	"DRAM"	A tabela de curvas é criada na memória **dinâmica** do NC.
	Nota: Se não for programado nenhum valor para este parâmetro, então é utilizado o local de armazenamento padrão definido no dado MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	

Indicação

Sobrescrever

Uma tabela de curvas é intitulada assim que uma nova definição de tabela utilizar seu número (<n>) (Exceção: Uma tabela de curvas está ativa em um acoplamento de eixo ou bloqueada com CTABLOCK. **Na sobregravação não é emitido nenhum aviso correspondente!**

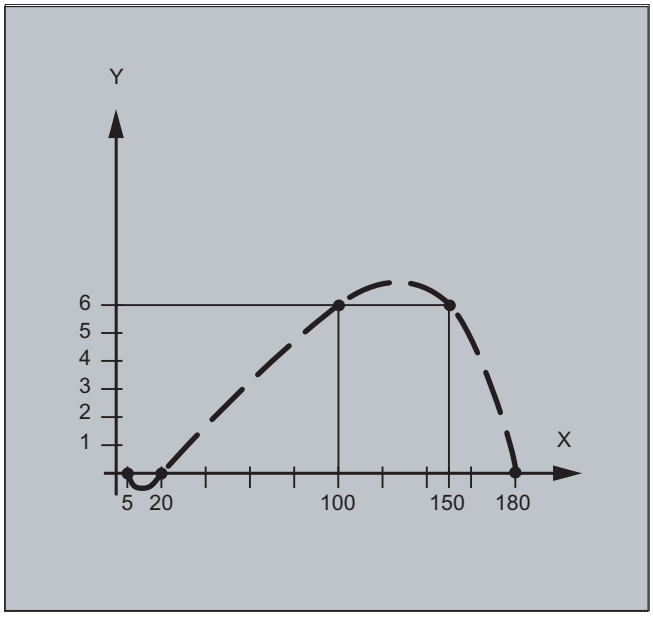
Exemplos

Exemplo 1: Segmento de programa como definição de tabela de curvas

Um segmento do programa deve ser utilizado sem alterações para definição de uma tabela de curvas. O comando ali contido e responsável pela parada do pré-processamento STOPRE pode permanecer e ele será imediatamente ativado quando o segmento do programa não for utilizado para definição da tabela e se CTABDEF e CTABEND foram removidos.

```
Código de programa              Comentário
…
CTABDEF(Y,X,1,1)                ; Definição de uma tabela de curvas.
…
IF NOT ($P_CTABDEF)
STOPRE
ENDIF
…
CTABEND
```

Exemplo 2: Definição de uma tabela de curvas não periódica



Código de programa	Comentário
N100 CTABDEF(Y,X,3,0)	; Início da definição de uma tabela de curvas não periódica com o número 3.
N110 X0 Y0	; 1ª instrução de movimento, define valores iniciais e a 1ª posição de apoio: Valor mestre: 0, valor escravo: 0
N120 X20 Y0	; 2ª Posição de apoio: Valor mestre: 0…20, valor escravo: Valor inicial…0
N130 X100 Y6	; 3ª Posição de apoio: Valor mestre: 20…100, valor escravo: 0…6
N140 X150 Y6	; 4ª Posição de apoio: Valor mestre: 100…150, valor escravo: 6…6
N150 X180 Y0	; 5ª Posição de apoio: Valor mestre: 150…180, valor escravo: 6…0
N200 CTABEND	; Fim da definição. A tabela de curvas é gerada em sua representação interna como polinômio de até 5° grau. O cálculo da curva com as posições de apoio indicadas depende do tipo de interpolação modal escolhida (interpolação circular, interpolação linear, interpolação Spline). O estado do programa de peça antes do início da definição é restabelecido.

Exemplo 3: Definição de uma tabela de curvas periódica

Definição de uma tabela de curvas periódica com número 2, faixa de valores mestres de 0 até 360, movimento do eixo escravo de 0 para 45 e retorno ao 0:

Código de programa	Comentário
N10 DEF REAL DEPPOS	

Código de programa	**Comentário**
N20 DEF REAL GRADIENT	
N30 CTABDEF(Y,X,2,1)	; Início da definição.
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	; Fim da definição.
;Teste da curva através do acoplamento do Y no 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)	; Leitura da função de tabela com valor mestre 75.0.
N180 G0 X75 Y=DEPPOS	; Posicionamento do eixo mestre e eixo escravo.
;Após a ativação do acoplamento não é necessária nenhuma sincronização do eixo escravo.	
N190 LEADON(Y,X,2)	
N200 G1 X110 F1000	
N210 LEADOF(Y,X)	
N220 M30	

Outras informações

Valor inicial e final da tabela de curvas

Como valor inicial para o início da faixa de definição da tabela de curvas vale a primeira especificação das posições de eixo envolvidas (a primeira instrução de movimento) dentro da definição da tabela de curvas. O valor final da faixa de definição da tabela de curvas é determinado de acordo com o último comando de movimento.

Escopo de linguagem disponível

Na definição da tabela de curvas todo escopo de linguagem NC está disponível.

Indicação

As seguintes especificações não são permitidas em definições de tabelas de curvas:

- Parada de pré-processamento
- Saltos no movimento de eixos mestres (p. ex. na mudança de transformações)
- Uma instrução de movimento exclusivamente para o eixo escravo
- Reversão de movimento do eixo mestre, isto é, a posição do eixo mestre sempre deve ser única
- Instruções CTABDEF e CTABEND em diferentes níveis de programa.

Efeito de instruções modais

Todas instruções ativas modalmente que existem na definição da tabela de curvas são invalidadas no final da definição de tabelas. Com isso o programa de peça onde se realiza a definição de tabelas encontra-se no mesmo estado antes e após a definição de tabelas.

Atribuições no parâmetro R

As atribuições no parâmetro R dentro da definição de tabelas são desfeitas após o CTABEND.

Exemplo:

Código de programa	**Comentário**
...	
R10=5 R11=20	; R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	; R10=25
X=R10	
CTABEND	
...	; R10=5

Ativação do ASPLINE, BSPLINE, CSPLINE

Se no interior de uma definição da tabela de curvas CTABDEF ... CTABEND for ativado um ASPLINE, BSPLINE ou CSPLINE, então, antes desta ativação Spline deve ser programado pelo menos um ponto de partida. Uma ativação imediata após CTABDEF deve ser evitada, caso contrário a Spline da atual posição de eixo será cancelada antes da definição da tabela de curvas.

Exemplo:

Código de programa
...
CTABDEF(Y,X,1,0)
X0 Y0

Código de programa
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND

Utilização repetida de tabelas de curvas

A relação funcional do eixo mestre e o eixo escravo, calculada através da tabela de curvas, é mantida no número de tabela selecionado além do fim do programa de peça e além do POWER OFF, isto se a tabela estiver armazenada na memória NC estática (SRAM).

Uma tabela que foi criada na memória dinâmica (DRAM) é deletada com POWER ON e, quando solicitada, deverá ser criada novamente.

A tabela de curvas criada pode ser aplicada em qualquer combinação de eixos mestres e escravos e está em função da configuração de eixos utilizados para criar a tabela de curvas.

Sobregravação de tabelas de curvas

Uma tabela de curvas é sobregravada assim que uma nova definição de tabela utilizar seu número.

Exceção: Uma tabela de curvas está ativa em um acoplamento de eixo ou bloqueada com CTABLOCK.

Indicação

Na sobregravação de tabelas de curvas não é emitido nenhum aviso correspondente!

Definição de tabela de curvas ativa?

Com a variável de sistema `$P_CTABDEF` pode ser consultado a qualquer momento do programa de peça se uma definição de tabela de curvas está ativa.

Cancelamento da definição de tabela de curvas

O segmento do programa de peça pode ser utilizado novamente como programa de peça real depois de serem retiradas as instruções referentes à definição da tabela de curvas.

Carregamento de tabelas de curvas através da função "Executar externamente"

Para a execução externa de tabelas de curvas deve-se selecionar o buffer de recarga (DRAM) através do MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE de modo que a definição total da tabela de curvas possa ser armazenada ao mesmo tempo no buffer de recarga. Caso contrário a execução do programa de peça é cancelada com um alarme.

Saltos do eixo escravo

Em função do dado de máquina:
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION
os saltos do eixo escravo podem ser tolerados durante a ausência de movimento do eixo mestre.

13.2.2 Controle de presença de uma tabela de curvas (CTABEXISTS)

Com o comando CTABEXISTS pode ser verificado se um determinado número de tabela de curvas está presente na memória NC.

Sintaxe

```
CTABEXISTS(<n>)
```

Significado

CTABEXISTS:	Verifica se a tabela de curvas de número <n> está presente na memória estática ou dinâmica do NC	
	0	A tabela não existe
	1	A tabela existe
<n>:	Número (**ID**) da tabela de curvas	

13.2.3 Eliminação de tabelas de curvas (CTABDEL)

Com o CTABDEL pode-se deletar tabelas de curvas.

Indicação

As tabelas de curvas ativadas em um acoplamento de eixos não podem ser deletadas.

Sintaxe

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<local de armazenamento>)
CTABDEL()
CTABDEL(,,<local de armazenamento>)
```

Significado

CTABDEL:	Comando para deletar tabelas de curvas
<n>:	Número (**ID**) da tabela de curvas que deve ser deletada Durante a eliminação de uma área de tabela de curvas CTABDEL(<n>,<m>) é indicado com <n> o número da primeira tabela de curvas da área.
<m>:	Durante a eliminação de uma área de tabela de curvas CTABDEL(<n>,<m>) é indicado com <m> o número da última tabela de curvas da área. O <m> deve ser maior que o <n>!

`<Local de armazenamento>`:	Especificação do local de armazenamento (opcional) Durante a eliminação **sem** indicação do local de armazenamento as tabelas de curvas indicadas são deletadas da memória estática e dinâmica do NC. Durante a eliminação **com** indicação do local de armazenamento apenas são deletadas as tabelas de curvas presentes na memória indicada. As demais são mantidas.	
	`"SRAM"`	Eliminação na memória **estática** do NC
	`"DRAM"`	Eliminação na memória **dinâmica** do NC

Se o `CTABDEL` for programado sem indicação da tabela de curvas que deve ser deletada, então **todas** as tabelas de curvas serão deletadas, inclusive as da memória indicada:

`CTABDEL()`:	Deleta todas as tabelas da memória estática e dinâmica do NC
`CTABDEL(,,"SRAM")`:	Deleta todas as tabelas da memória estática do NC
`CTABDEL(,,"DRAM")`:	Deleta todas as tabelas da memória dinâmica do NC

Indicação

Se durante a eliminação múltipla `CTABDEL(<n>,<m>)` ou `CTABDEL()` existir pelo menos um acoplamento ativo das tabelas de curvas a serem deletadas, então o comando de eliminação não será executado, isto é, **nenhuma** das tabelas de curvas endereçadas será deletada.

13.2.4 Bloqueio de tabelas de curvas contra eliminação e sobregravação (CTABLOCK, CTABUNLOCK)

As tabelas de curvas podem ser protegidas contra uma eliminação e sobregravação acidental através da definição de um bloqueio. Um bloqueio definido pode ser cancelado a qualquer momento.

Sintaxe

Definição de bloqueio:

```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<local de armazenamento>)
CTABLOCK()
CTABLOCK(,,<local de armazenamento>)
```

Cancelamento de bloqueio:

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<local de armazenamento>)
CTABUNLOCK()
CTABUNLOCK(,,<local de armazenamento>)
```

Significado

CTABLOCK:	Comando para **definição** de um bloqueio contra eliminação/sobregravação	
CTABUNLOCK:	Comando para **cancelamento** de um bloqueio contra eliminação/sobregravação O CTABUNLOCK libera novamente as tabelas bloqueadas com CTABLOCK. As tabelas de curvas que estiverem envolvidas em um acoplamento ativo permanecem bloqueadas e não podem ser deletadas. O bloqueio é cancelado com CTABLOCK assim que o travamento for cancelado com a desativação do acoplamento que estiver ativo. Com isso esta tabela pode ser deletada. Não será mais necessária uma nova chamada do CTABUNLOCK.	
<n>:	Número (**ID**) da tabela de curvas que deve ser bloqueada/desbloqueada Durante o bloqueio/desbloqueio de uma área de tabela de curvas CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) é indicado com <n> o número da primeira tabela de curvas da área.	
<m>:	Durante o bloqueio/desbloqueio de uma área de tabela de curvas CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) é indicado com <m> o número da última tabela de curvas da área. O <m> deve ser maior que o <n>!	
<Local de armazenamento>:	Especificação do local de armazenamento (opcional) Durante a definição/cancelamento de um bloqueio **sem** indicação do local de armazenamento as tabelas de curvas indicadas são bloqueadas/desbloqueadas na memória estática e dinâmica do NC. Durante a definição/cancelamento de um bloqueio **com** indicação do local de armazenamento apenas são bloqueadas/desbloqueadas as tabelas de curvas presentes na memória indicada. As demais não são bloqueadas/desbloqueadas.	
	"SRAM"	Definição/cancelamento de bloqueio na memória **estática** do NC
	"DRAM"	Definição/cancelamento de bloqueio na memória **dinâmica** do NC

Se o CTABLOCK/CTABUNLOCK for programado sem indicação da tabela de curvas que deve ser bloqueada/desbloqueada, então **todas** as tabelas de curvas serão bloqueadas/ desbloqueadas, inclusive as da memória indicada:

CTABLOCK():	Bloqueia todas as tabelas da memória estática e dinâmica do NC
CTABLOCK(,,"SRAM"):	Bloqueia todas as tabelas da memória estática do NC
CTABLOCK(,,"DRAM"):	Bloqueia todas as tabelas da memória dinâmica do NC
CTABUNLOCK():	Desbloqueia todas as tabelas da memória estática e dinâmica do NC
CTABUNLOCK(,,"SRAM"):	Desbloqueia todas as tabelas da memória estática do NC
CTABUNLOCK(,,"DRAM"):	Desbloqueia todas as tabelas da memória dinâmica do NC

13.2.5 Tabelas de curvas: Determinação de propriedades de tabela (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Com estes comandos podem ser consultadas propriedades importantes de uma tabela de curvas (número de tabela, estado de bloqueio, local de armazenamento, periodicidade).

Sintaxe

```
CTABID(<p>)
CTABID(<p>,<local de armazenamento>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

Significado

CTABID:	Fornece o **número de tabela** que está registrado na memória indicada como a <p>[a] tabela de curvas. Exemplo: O CTABID(1,"SRAM") fornece o número da primeira tabela de curvas na memória estática do NC. Neste caso, a primeira tabela de curvas não corresponde ao número de tabela mais alto. **Nota:** Se a ordem das tabelas de curvas é alterada entre duas chamadas consecutivas de CTABID, por exemplo, através da eliminação de tabelas de curvas com CTABDEL, o CTABID(<p>,...) pode fornecer uma tabela diferente da tabela anterior de mesmo número <p>.	
CTABISLOCK:	Retorna o **estado de bloqueio** da tabela de curvas com o número <n>:	
	0	A tabela não está bloqueada
	1	A tabela está bloqueada através do CTABLOCK
	2	A tabela está bloqueada através de acoplamento ativo
	3	A tabela está bloqueada através do CTABLOCK e acoplamento ativo
	-1	A tabela não existe
CTABMEMTYP:	Fornece o **local de armazenamento** da tabela de curvas com o número <n>:	
	0	Tabela na memória estática do NC
	1	Tabela na memória dinâmica do NC
	-1	A tabela não existe
CTABPERIOD:	Fornece a **periodicidade** da tabela de curvas com o número <n>:	
	0	A tabela não é periódica
	1	A tabela é periódica no eixo mestre
	2	A tabela é periódica no eixo mestre e no eixo escravo
	-1	A tabela não existe
<p>:	Número de registro na memória	
<n>:	Número (**ID**) da tabela de curvas	

<Local de armazenamento>:	Especificação do local de armazenamento (opcional)	
	"SRAM"	Memória **estática** do NC
	"DRAM"	Memória **dinâmica** do NC
	Nota: Se não for programado nenhum valor para este parâmetro, então é utilizado o local de armazenamento padrão definido no dado MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	

13.2.6 Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Os seguintes valores de tabela de curvas podem ser lidos no programa de peça:

- Valores de eixo escravo e eixo mestre no início e no fim de uma tabela de curvas
- Valores de eixo escravo no início e no fim de um segmento de curva
- Valor de eixo escravo para um valor de eixo mestre
- Valor de eixo mestre para um valor de eixo escravo
- Valor mínimo e valor máximo do eixo escravo
  - na faixa de definição inteira da tabela de curvas
    ou
  - em um intervalo definido da tabela de curvas

Sintaxe

```
CTABTSV(<n>,<gradiente>[,<eixo escravo>])
CTABTEV(<n>,<gradiente>[,<eixo escravo>])
CTABTSP(<n>,<gradiente>[,<eixo mestre>])
CTABTEP(<n>,<gradiente>[,<eixo mestre>])
CTABSSV(<valor mestre>,<n>,<gradiente>[,<eixo escravo>])
CTABSEV(<valor mestre>,<n>,<gradiente>[,<eixo escravo>])
CTAB(<valor mestre>,<n>,<gradiente>[,<eixo escravo>,<eixo mestre>]
CTABINV(<valor escravo>,<valor aproximado>,<n>,<gradiente>[,<eixo
escravo>,<eixo mestre>]
CTABTMIN(<n>[,<eixo escravo>])
CTABTMAX(<n>[,<eixo escravo>])
CTABTMIN(<n>,<a>,<b>[,<eixo escravo>,<eixo mestre>])
CTABTMAX(<n>,<a>,<b>[,<eixo escravo>,<eixo mestre>])
```

Significado

CTABTSV:	Leitura do valor de eixo escravo no **início** da tabela de curvas nº <n>
CTABTEV:	Leitura do valor de eixo escravo no **fim** da tabela de curvas nº <n>
CTABTSP:	Leitura do valor de eixo mestre no **início** da tabela de curvas nº <n>
CTABTEP:	Leitura do valor de eixo mestre no **fim** da tabela de curvas nº <n>

CTABSSV:	Leitura do valor de eixo escravo no **início** do segmento de curva pertencente ao valor de eixo mestre especificado (<valor mestre>)
CTABSEV:	Leitura do valor de eixo escravo no **fim** do segmento de curva pertencente ao valor de eixo mestre especificado (<valor mestre>)
CTAB:	Leitura do valor de eixo escravo para o valor de eixo mestre especificado (<valor mestre>)
CTABINV:	Leitura do valor de eixo mestre para o valor de eixo escravo especificado (<valor escravo>)
CTABTMIN:	Determinação do **valor mínimo** do eixo escravo: • na faixa de definição inteira da tabela de curvas ou • em um intervalo definido <a> ... <b>
CTABTMAX:	Determinação do **valor máximo** do eixo escravo: • na faixa de definição inteira da tabela de curvas ou • em um intervalo definido <a> ... <b>
<n>:	Número (**ID**) da tabela de curvas
<gradiente>:	No parâmetro <gradiente> é retornada a **inclinação** da função de tabela de curvas na posição determinada
<eixo escravo>:	Eixo, cujo movimento deve ser calculado através da tabela de curvas (opcional)
<eixo mestre>:	Eixo, que fornece os valores mestres para o cálculo do movimento do eixo escravo (opcional)
<valor escravo>:	Valor de eixo escravo para leitura do valor de eixo mestre correspondente com CTABINV
<valor mestre>:	Valor de eixo mestre: • para leitura do valor de eixo escravo correspondente com CTAB ou • para a seleção do segmento de curvas com CTABSSV/CTABSEV
<valor aproximado>:	A associação de um valor de eixo mestre com um valor de eixo escravo com CTABINV nem sempre precisa ser única. Por isso o CTABINV requer um valor aproximado como parâmetro para o valor de eixo mestre esperado.
<a>:	Limite **inferior** do intervalo de valor mestre com CTABTMIN/CTABTMAX
<b>:	Limite **superior** do intervalo de valor mestre com CTABTMIN/CTABTMAX
	Nota: O intervalo de valores mestres <a> ... <b> deve estar dentro da faixa de definição da tabela de curvas.

Exemplos

Exemplo 1:

Determinação dos valores de eixo escravo e valores de eixo mestre no início e no fim da tabela de curvas, assim como do valor mínimo e do valor máximo do eixo escravo na faixa de definição inteira da tabela de curvas.

```
Código de programa                          Comentário
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)                       ; Início da definição de tabela
N110 X0 Y10                                 ; Posição de partida do 1° segmento de tabela
N120 X30 Y40                                ; Posição final do 1° segmento de tabela =
                                            Posição de partida do 2° segmento de tabela
N130 X60 Y5                                 ; Posição final do 2° segmento de tabela = ...
N140 X70 Y30
N150 X80 Y20
N160 CTABEND                                ; Fim da definição de tabela.
...
N200 STARTPOS=CTABTSV(1,GRADIENT)           ; Valor de eixo escravo no início da tabela de
                                            curvas = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)             ; Valor de eixo escravo no fim da tabela de
                                            curvas = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)          ; Valor de eixo mestre no início da tabela de
                                            curvas = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)            ; Valor de eixo mestre no fim da tabela de
                                            curvas = 80
N240 MINVAL=CTABTMIN(1)                     ; Valor mínimo do eixo escravo com Y=5
N250 MAXVAL=CTABTMAX(1)                     ; Valor máximo do eixo escravo com Y=40
```

Exemplo 2:

Determinação dos valores de eixo escravo no início e no fim do segmento de curva pertencente ao valor de eixo mestre X=30.

```
Código de programa                          Comentário
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                       ; Início da definição de tabela.
N110 X0 Y0                                  ; Posição de partida do 1° segmento de tabela
```

Código de programa	Comentário
N120 X20 Y10	; Posição final do 1° segmento de tabela = Posição de partida do 2° segmento de tabela
N130 X40 Y40	; Posição final do 2° segmento de tabela = ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	; Fim da definição de tabela.
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; Posição de partida Y no 2° segmento = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	; Posição final Y no 2° segmento = 40

Outras informações

Utilização em ações síncronas

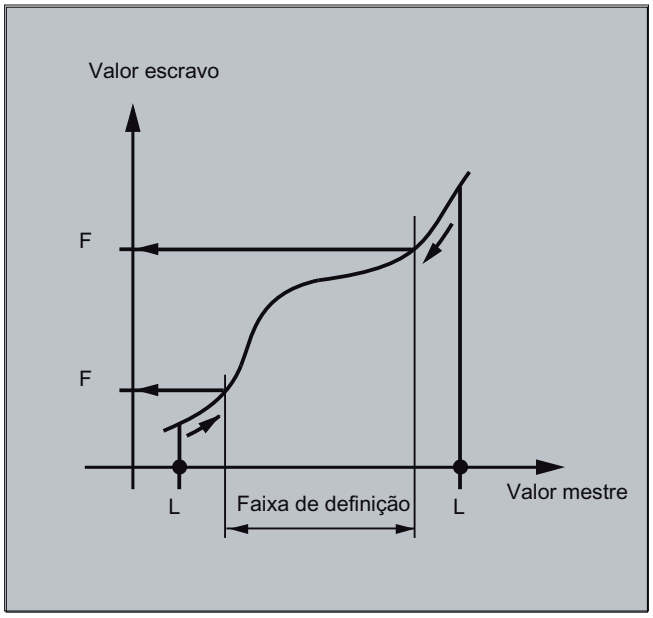
Todos comandos para leitura de valores de tabelas de curvas também podem ser utilizados em ações síncronas (veja também o capítulo "Ações sincronizadas de movimentos").

Para utilização dos comandos CTABINV, CTABTMIN e CTABTMAX deve-se prestar atenção para que:

- no momento de execução uma capacidade suficiente do NC esteja disponível
  ou
- o número de segmentos da tabela de curvas seja consultado antes da chamada para, eventualmente, dividir a tabela afetada

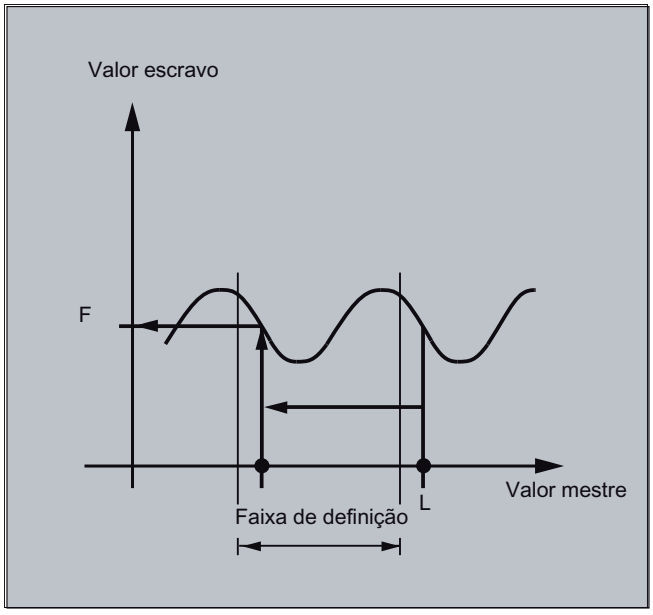
CTAB em tabelas de curvas não periódicas

Se o <valor mestre> especificado estiver fora da faixa de definição, é retornado o limite superior ou inferior como valor escravo:



CTAB em tabelas de curvas periódicas

Se o <valor mestre> especificado estiver fora da faixa de definição, é avaliado o valor mestre do Modulo da faixa de definição e retornado o valor escravo correspondente:



Valor aproximado para CTABINV

O comando CTABINV requer um valor aproximado para o valor mestre esperado. O CTABINV retorna o valor mestre que está mais próximo do valor aproximado. Por exemplo, o valor aproximado pode ser o valor mestre de um ciclo de interpolação anterior.

Inclinação da função de tabela de curvas

A emissão da inclinação (<gradiente>) permite o cálculo da velocidade do eixo mestre ou eixo escravo na respectiva posição.

Especificação do eixo mestre ou eixo escravo

A especificação opcional do eixo mestre ou eixo escravo é importante se o eixo mestre e o eixo escravo estiverem configurados em diferentes unidades lineares.

CTABSSV, CTABSEV

Os comandos CTABSSV e CTABSEV nos seguintes casos **não** são adequados para a consulta dos segmentos programados:

- Foram programados círculos e evolventes.
- Chanfros e arredondamentos estão ativos com CHF/RND.
- A suavização está ativa com G643.
- O compressor de blocos NC está ativo com COMPON/COMPCURV/COMPCAD.

13.2.7 Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Com estes comandos o programador tem a possibilidade de se manter atualizado com informações sobre a ocupação dos recursos para tabelas de curvas, segmentos de tabelas e polinômios.

Sintaxe

```
CTABNO
CTABNOMEM(<local de armazenamento>)
CTABFNO(<local de armazenamento>)
CTABSEGID(<n>,<local de armazenamento>)
CTABSEG(<local de armazenamento>,<tipo de segmento>)
CTABFSEG(<local de armazenamento>,<tipo de segmento>)
CTABMSEG(<local de armazenamento>,<tipo de segmento>)
CTABPOLID(<n>)
CTABPOL(<local de armazenamento>)
CTABFPOL(<local de armazenamento>)
CTABMPOL(<local de armazenamento>)
```

Significado

CTABNO:	Determinação do número total das tabelas de curvas **definidas** (na memória estática e dinâmica do NC)
CTABNOMEM:	Determinação das tabelas de curvas **definidas** no <local de armazenamento> indicado
CTABFNO:	Determinação das tabelas de curvas **ainda possíveis** no <local de armazenamento> indicado
CTABSEGID:	Determinação dos segmentos de curva do <tipo de segmento> indicado, que são utilizados pela tabela de curvas de número <n>
CTABSEG:	Determinação dos segmentos de curva **utilizados** do <tipo de segmento> indicado no <local de armazenamento> indicado
CTABFSEG:	Determinação dos segmentos de curva **ainda possíveis** do <tipo de segmento> indicado no <local de armazenamento> indicado
CTABMSEG:	Determinação dos segmentos de curva **máximos possíveis** do <tipo de segmento> indicado no <local de armazenamento> indicado
CTABPOLID:	Determinação da quantidade de polinômios de curvas, que são utilizados pela tabela de curvas de número <n>
CTABPOL:	Determinação da quantidade de polinômios de curvas **utilizados** no <local de armazenamento> indicado
CTABFPOL:	Determinação da quantidade de polinômios de curvas **ainda possível** no <local de armazenamento> indicado
CTABMPOL:	Determinação da quantidade de polinômios de curvas **máxima possível** no <local de armazenamento> indicado
<n>:	Número (**ID**) da tabela de curvas

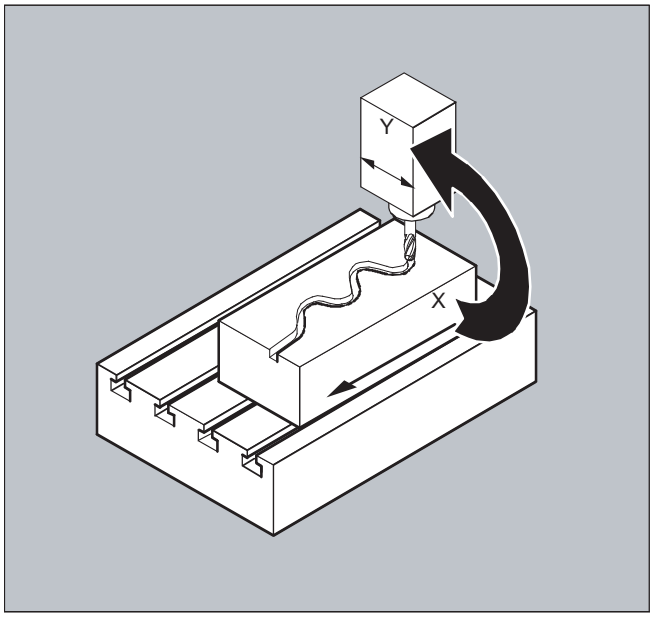
<Local de armazenamento>:	Especificação do local de armazenamento (opcional)	
	"SRAM"	Memória **estática** do NC
	"DRAM"	Memória **dinâmica** do NC
	Nota: Se não for programado nenhum valor para este parâmetro, então é utilizado o local de armazenamento padrão definido no dado MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.	
<Tipo de segmento>:	Especificação do tipo de segmento (opcional)	
	"L"	Segmentos lineares
	"P"	Segmentos de polinômio
	Nota: Se não for programado nenhum valor para este parâmetro, então é retornada a soma dos segmentos lineares e de polinômio.	

13.3 Acoplamento axial de valor mestre (LEADON, LEADOF)

Indicação

Esta função não está disponível para o SINUMERIK 828D!

No acoplamento axial de valor mestre o eixo mestre e o eixo escravo são movimentados de forma sincronizada. Neste caso a respectiva posição do eixo escravo é atribuída para uma posição, eventualmente simulada, do eixo mestre através de uma tabela de curvas e de um polinômio calculado a partir desta.



O **eixo mestre** é o eixo que fornece os valores de entrada para a tabela de curvas. O **eixo escravo** é o eixo que assume as posições calculadas na tabela de curvas.

Acoplamento de valor real e valor nominal

Como valores mestres, portanto valores de saída para determinar a posição do eixo escravo, podem ser utilizados:

- Valores reais da posição do eixo mestre: Acoplamento de valor real
- Valores reais da posição do eixo mestre: Acoplamento do valor nominal

O acoplamento de valor mestre sempre é aplicado no sistema de coordenadas básico.

Para criação de tabelas de curvas consulte o capítulo "Tabelas de curvas".

Sintaxe

```
LEADON(<eixo escravo>,<eixo-mestre>,<n>)
LEADOF(<eixo escravo>,<eixo-mestre>)
```

ou desativação sem indicação do eixo mestre:

```
LEADOF(<eixo escravo>)
```

O acoplamento de valor-mestre pode ser ativado e desativado tanto a partir do programa como durante o movimento de ações sincronizadas, consulte o capítulo "Ações sincronizadas de movimentos".

Significado

`LEADON`:	Ativação do acoplamento de valor mestre
`LEADOF`:	Desativação do acoplamento de valor mestre
`<eixo escravo>`:	Eixo escravo
`<eixo mestre>`:	Eixo mestre
`<n>`:	Número da tabela de curvas
`$SA_LEAD_TYPE`:	Comutação entre acoplamento de valor nominal e valor real

Desativar o acoplamento de valor mestre, LEADOF

Com a desativação do acoplamento de valor mestre o eixo escravo torna-se novamente um eixo normal de comando!

Acoplamento axial de valor mestre e diversos estados operacionais, RESET

Independentemente do ajuste no dado de máquina, os acoplamentos de valor mestre são desativados com RESET.

Exemplo de acoplamento de valor mestre a partir da ação sincronizada

Em uma prensa um acoplamento mecânico convencional entre um eixo mestre (eixo de estampagem) e os eixos de um sistema Transfer dotado de eixos Transfer e eixos auxiliares deve ser substituído por um sistema de acoplamento eletrônico.

Aqui demonstra-se como se substitui um sistema Transfer mecânico por um sistema Transfer eletrônico em uma prensa. Os processos de acoplamento e desacoplamento são realizados como **ações sincronizadas estáticas**.

A partir do eixo mestre de valor mestre (eixo de estampagem) os eixos Transfer e eixos auxiliares são definidos como eixos escravos através de tabelas de curvas.

Eixos escravos

X - Eixo de avanço e longitudinal
YL - Eixo de fechamento e eixo transversal
ZL - Eixo de elevação
U - Avanço de rolo, eixo auxiliar
V - cabeçote de guia, eixo auxiliar
W - Aplicação de graxa, eixo auxiliar

Ações

As ações ocorrem, por exemplo, em ações sincronizadas:

- Acoplar, `LEADON(<eixo escravo>,<eixo-mestre>,<nr. de tabelas de curvas>)`
- Desacoplar, `LEADOF(<eixo escravo>,<eixo-mestre>)`
- Inserir valor real, `PRESETON(<eixo>,<valor>)`

- Definir marcador, `$AC_MARKER[i]=<valor`
- Tipo de acoplamento: valor mestre real/virtual
- Definindo as posições dos eixos, `POS[<eixo>]=<valor>`

Condições

Como condições são avaliadas as entradas digitais rápidas, variáveis de tempo real `$AC_MARKER` e comparações de posição que estiverem interligadas com o operador lógico AND.

Indicação

No capítulo a seguir foram utilizadas quebras de linhas, recuos e parágrafos em **negrito** especialmente para facilitar a leitura da programação. Para o comando tudo que estiver em um número de linha constitui uma linha separada.

Comentário

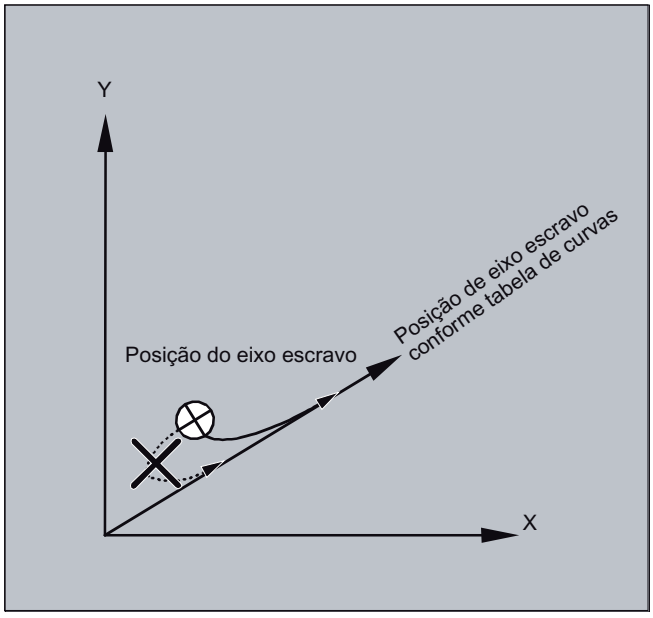
Código de programa	**Comentário**
	; Define todas as ações síncronas estáticas.
	; ****Resetar o marcador
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
	; **** E1 0=>1 acoplamento Transfer ATIVADO
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_MARKER[0]==0) DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) $AC_MARKER[0]=1	
	**** E1 0=>1 acoplamento do avanço de rolos ATIVADO
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0) DO LEADON(U,LW,4) PRESETON(U,0) $AC_MARKER[5]=1	
	**** E1 0->1 acoplamento do cabeçote guia ATIVADO
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0) DO LEADON(V,LW,4) PRESETON(V,0) $AC_MARKER[6]=1	
	**** E1 0->1 acoplamento da aplicação de graxa ATIVADO
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[7]==0) DO LEADON(W,LW,4) PRESETON(W,0) $AC_MARKER[7]=1	
	**** E2 0=>1 acoplamento DESATIVADO
N30 IDS=3 EVERY ($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	

Descrição

O acoplamento de valor mestre requer a sincronização do eixo mestre e do eixo escravo. Esta sincronização somente pode ser alcançada se o eixo escravo estiver dentro da faixa de tolerância da curva calculada pela tabela de curvas durante a ativação do acoplamento do valor mestre.

A faixa de tolerância para a posição do eixo escravo é definida pelo dado da máquina `MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE` definido.

Se com a ativação do acoplamento de valor mestre o eixo escravo ainda não estiver na respectiva posição, então o processo sincronizado será realizado automaticamente assim que o valor nominal de posição calculado para o eixo escravo realmente se aproximar da posição do eixo escravo. Neste caso o eixo escravo é movimentado durante o processo de sincronização para o sentido que foi definido pela velocidade nominal do eixo escravo (calculado a partir da velocidade do eixo mestre e conforme a tabela de curvas CTAB).

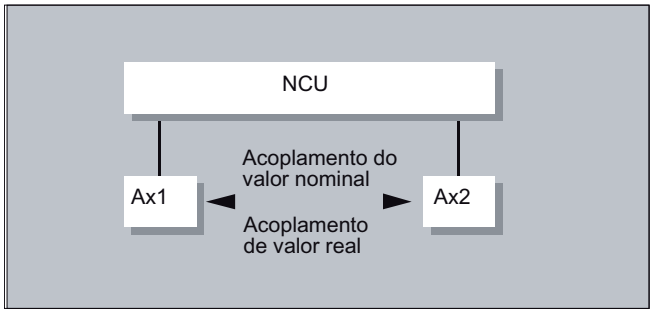


Sem sincronismo

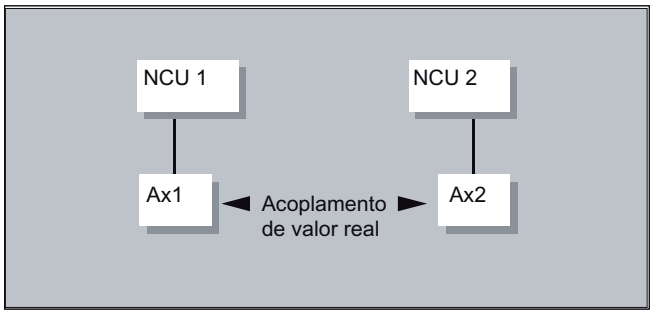
Se com a ativação do acoplamento de valor mestre a posição nominal calculada se afastar da atual posição de eixo escravo, não será realizada nenhuma sincronização.

Acoplamento de valor real e valor nominal

Quando comparado ao acoplamento de valor real, o acoplamento de valor nominal oferece uma melhor sincronização entre o eixo mestre e o eixo escravo e por isso está ajustado como padrão.



O acoplamento de valor nominal somente é possível se o eixo mestre e o eixo escravo forem interpolados pela mesma NCU. No caso de um eixo mestre externo, o eixo escravo apenas pode ser acoplado no eixo mestre através de valores reais.



Uma **comutação** somente é possível através do dado de ajuste $SA_LEAD_TYPE.

A comutação entre o acoplamento de valor real e o de valor nominal sempre deve ser realizado com o eixo escravo imobilizado. Pois apenas em estado parado é realizada uma nova sincronização após a comutação.

Exemplos de aplicação

A leitura dos valores reais não pode ser executada isenta de falhas com grandes vibrações da máquina. Na aplicação do acoplamento do valor mestre em uma linha Transfer de prensas, para os passos de trabalho de maior vibração pode ser necessário comutar do acoplamento de valor real para o acoplamento de valor nominal.

Simulação do valor mestre no acoplamento de valor nominal

O interpolador do eixo mestre pode ser separado do servo através de dado de máquina. Desse modo no acoplamento do valor nominal podem ser gerados valores nominais sem movimento efetivo do eixo mestre.

Os valores mestres gerados através do acoplamento de valor nominal podem ser lidos a partir das seguintes variáveis para utilização em ações sincronizadas:

- $AA_LEAD_P	Posição de valor mestre
- $AA_LEAD_V	Velocidade de valor mestre

Gerar valores mestres

Os valores mestres também podem ser gerados com outros procedimentos próprios programados. Os valores mestres gerados são gravados na variável

- $AA_LEAD_SP	Posição de valor mestre
- $AA_LEAD_SV	Velocidade de valor mestre

e também lidos da própria. Para a utilização destas variáveis deve-se definir o dado de ajuste $SA_LEAD_TYPE = 2.

Estado do acoplamento

No programa de peça NC se pode consultar o estado do acoplamento com as seguintes variáveis de sistema:

`$AA_COUP_ACT[eixo]`
0: Nenhum acoplamento ativo
16: Acoplamento de valor mestre ativo

Gerenciamento de estado em ações sincronizadas

Processos de ativação e de acoplamento são gerenciados através das variáveis de tempo real:

`$AC_MARKER[i] = n`
com:
número de marcador i
valor de estado n

13.4 Caixa de transmissão eletrônica (EG)

Com o auxílio da função "caixa de transmissão eletrônica" é possível controlar o movimento de um **eixo escravo** conforme o bloco de movimento linear em função de até cinco **eixos mestres**. As relações entre os eixos mestres e o eixo escravo são definidas para cada eixo mestre através do fator de acoplamento.

A proporção de movimento do eixo escravo é formada mediante a adição com base nas proporções de movimento dos eixos mestres multiplicada pelos fatores de acoplamento correspondentes. Com a ativação de um grupo de eixos EG a sincronização do eixo escravo pode ser forçada para uma posição definida. A partir do programa de peça um grupo de caixa de transmissão pode ser:

- definido,
- ativado,
- desativado,
- deletado

.

O movimento do eixo escravo pode ser, alternativamente, derivado dos

- valores nominais do eixos mestres assim como os
- valores reais dos eixos mestres.

Como ampliação também se pode realizar relações entre os eixos mestres e o eixo escravo mediante **tabelas de curvas** (veja o capítulo sobre comportamento de percurso). As caixas de transmissão eletrônicas podem ser conectadas em cascata, isto é, o eixo escravo de uma caixa de transmissão eletrônica pode ser o eixo mestre de outra caixa de transmissão eletrônica.

13.4.1 Definir caixa de transmissão eletrônica (EGDEF)

Um grupo de eixos EG é definido com a especificação do eixo escravo, por pelo menos um, mas no máximo, cinco eixos mestres com o tipo de acoplamento correspondente.

Pré-requisito

Condição para uma definição de grupo de eixos EG:

Para o eixo escravo ainda não pode ser definido nenhum acoplamento de eixo (eventualmente, um existente deve ser deletado antes com EGDEL).

Sintaxe

```
EGDEF(eixo escravo,eixo mestre1,tipo de acoplamento1,eixo
mestre2,tipo de acoplamento2,...)
```

Significado

`EGDEF`:	Definição de uma caixa de transmissão eletrônica	
`Eixo escravo`:	Eixo que é controlado por eixos mestres	

`Eixo mestre1`
`,...,`
`Eixo mestre5`
	Eixos que controlam os eixos escravos	

`Tipo de acoplamento1`
`,...,`
`Tipo de acoplamento5`
	Tipo de acoplamento
	O tipo de acoplamento deve ser o mesmo para todos eixos mestres e por isso que deve ser especificado individualmente para cada eixo mestre.	

	Valor:	**Significado:**
	0	O eixo escravo é controlado pelo **valor real** do eixo mestre correspondente.
	1	O eixo escravo é controlado pelo **valor nominal** do eixo mestre correspondente.

Indicação

Os fatores de acoplamento são pré-definidos com zero na definição do grupo de acoplamento EG.

Indicação

O EGDEF ativa a parada de pré-processamento. A definição da caixa de transmissão com EGDEF também pode ser utilizada sem alteração se atua um ou vários eixos mestres através da **tabela de curvas** no eixo escravo.

Exemplo

Código de programa	**Comentário**
`EGDEF(C,B,1,Z,1,Y,1)`	`; Definição de um grupo de eixos EG. Os eixos mestres B, Z e Y controlam o eixo escravo C através do valor nominal.`

13.4.2 Ativar caixa de transmissão eletrônica (EGON, EGONSYN, EGONSYNE)

Para a ativação de um grupo de eixos EG existem 3 variantes.

Sintaxe

Variante 1:

O grupo de eixos EG é ativado seletivamente sem sincronização com:

```
EGON(FA,"modo de mudança de
blocos",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)
```

Variante 2:

O grupo de eixos EG é ativado seletivamente com sincronização com
`EGONSYN(FA,"modo de mudança de eixos",SynPosFA, [,LAi,SynPosLAi,Zi,Ni])`

Variante 3:

O grupo de eixos EG é ativado seletivamente com sincronização e especificado o modo de aproximação com:
`EGONSYNE(FA,"modo de mudança de blocos",SynPosFA,modo de aproximação[,LAi,SynPosLAi,Zi,Ni])`

Significado

Variante 1:

FA	Eixo escravo	
`Modo de mudança de blocos:`	Podem ser utilizados os seguintes modos:	
	`"NOC"`	A mudança de blocos é realizada imediatamente
	`"FINE"`	A mudança de blocos é realizada com "sincronização fina"
	`"COARSE"`	A mudança de blocos é realizada com "sincronização aproximada"
	`"IPOSTOP"`	A mudança de blocos é realizada com sincronização de valor nominal
LA1, ... LA5	Eixos mestres	
Z1, ... Z5	Numerador para o fator de acoplamento i	
N1, ... N5	Denominador para o fator de acoplamento i Fator de acoplamento i = numerador i / denominador i	

Somente podem ser programados os eixos mestres que foram previamente especificados com EGDEF. Deve ser programado pelo menos um eixo mestre.

Variante 2:

`FA`	Eixo escravo	
`Modo de mudança de blocos:`	Podem ser utilizados os seguintes modos:	
	`"NOC"`	A mudança de blocos é realizada imediatamente
	`"FINE"`	A mudança de blocos é realizada com "sincronização fina"
	`"COARSE"`	A mudança de blocos é realizada com "sincronização aproximada"
	`"IPOSTOP"`	A mudança de blocos é realizada com sincronização de valor nominal
`[,LAi,SynPosLAi,Zi,Ni]`	(não escrever colchetes) Mín. 1, máx. 5 sequências de:	
`LA1, ... LA5`	Eixos mestres	
`SynPosLAi`	Posição sincronizada para o eixo mestre i	

`Z1, ... Z5`	Numerador para o fator de acoplamento i
`N1, ... N5`	Denominador para o fator de acoplamento i Fator de acoplamento i = numerador i / denominador i

Somente podem ser programados eixos mestres que foram previamente especificados com EGDEF. Com as "posições sincronizadas" programadas para o eixo escravo (SynPosFA) e para os eixos mestres (SynPosLA) são definidas posições onde o grupo de acoplamento é considerado como *sincronizado*. Se a caixa de transmissão eletrônica não estiver em estado síncrono ao ser ativada, então o eixo escravo movimenta-se até sua posição sincronizada definida.

Variante 3:

Os parâmetros correspondem aos da variante 2 além de:

`Modo de aproximação:`	Podem ser utilizados os seguintes modos:	
	`"NTGT"`	Aproximar o próximo vão entre dentes com otimização de tempo
	`"NTGP"`	Aproximar o próximo vão entre dentes com otimização de curso
	`"ACN"`	Deslocar o eixo rotativo no sentido de giro negativo e de modo absoluto
	`"ACP"`	Deslocar o eixo rotativo no sentido de giro positivo e de modo absoluto
	`"DCT"`	Com otimização de tempo para a posição sincronizada programada
	`"DCP"`	Com otimização de curso para a posição sincronizada programada

A variante 3 apenas atua em eixos escravos Modulo que estão acoplados em eixos mestres Modulo. A otimização de tempo considera os limites de velocidade do eixo escravo.

Outras informações

Descrição das variantes de ativação

Variante 1:

No momento da ativação, as posições dos eixos mestres assim como o eixo escravo são armazenados como "posições sincronizadas". As "posições sincronizadas" podem ser lidas com a variável de sistema `$AA_EG_SYN`.

Variante 2:

Quando os módulos de eixo estiverem no grupo de acoplamento, seus valores de posição serão reduzidos no módulo. Desse modo se garante uma aproximação mais rápida possível da posição sincronizada (chamada *sincronização relativa*: p. ex. o próximo vão entre dentes). Se para o eixo escravo não for especificado o sinal de interface "liberação da sobreposição do eixo escravo" DB(30 +número de eixo), DBX 26 Bit 4, então não será realizado o deslocamento até a posição sincronizada. Em vez disso, o programa é parado no bloco EGONSYN e emitida a mensagem de alarme 16771 auto-deletável, até que o sinal acima mencionado seja ativado.

Variante 3:

A distância dos dentes (graus) resulta de: 360 * Zi/Ni. Para o caso em que o eixo escravo estiver parado no momento da chamada, a otimização de curso oferece o mesmo comportamento como na otimização de tempo.

Se o eixo escravo estiver em marcha, se realiza a sincronização com NTGP, independentemente da velocidade atual do eixo escravo, até o próximo vão entre dentes. Se o eixo escravo estiver em marcha, se realiza a sincronização com NTGT, dependentemente da velocidade atual do eixo escravo, até o próximo vão entre dentes. Com esse propósito o eixo também será freado quando necessário.

Tabelas de curvas

Se para um dos eixos mestres for utilizada uma **tabela de curvas**, então deve-se:

Ni	passar para 0 o denominador do fator de acoplamento para acoplamentos lineares. (o denominador 0 não seria permitido para acoplamentos lineares). O denominador zero indica para o comando que
Zi	se deve interpretar como número da tabela de curvas a ser utilizada. A tabela de curvas com o número indicado deve estar definida no momento da ativação.
LAi	A especificação do eixo mestre corresponde à especificação do eixo mestre no caso de acoplamento com fator de acoplamento (acoplamento linear).

Para mais informações sobre a utilização de tabelas de curvas e a conexão em cascata de caixas de transmissão eletrônicas e suas sincronização, veja na:
Literatura:
Manual de funcionamento, funções especiais; Acoplamentos de eixo e ESR (M3), Capítulo "Movimento acoplado e acoplamento de valor mestre".

Comportamento da caixa de transmissão eletrônica com Power On, RESET, mudança de modo de operação, bloco de pesquisa

- Após Power On **nenhum** acoplamento está ativo.
- Os acoplamentos ativos são mantidos mesmo com RESET e a mudança de modo de operação.
- Na localização de blocos os comandos de ativação, deletação, definição da caixa de transmissão eletrônica não são executados nem coletados, mas ignorados.

Variáveis de sistema da caixa de transmissão eletrônica

Com o auxílio de variáveis de sistema da transmissão eletrônica, o programa de peça pode verificar, e se for necessário, reagir aos estados do grupo de eixos EG.

As variáveis de sistema da caixa de transmissão eletrônica são identificadas da seguinte forma:

$AA_EG_ ...

ou

$VA_EG_ ...

Literatura:
Manual das variáveis de sistema

13.4.3 Desativar transmissão eletrônica (EGOFS, EGOFC)

Para a desativação de um grupo de eixos EG existem 3 variantes.

Programação

Variante 1:

Sintaxe	Significado
`EGOFS(eixo escravo)`	A caixa de transmissão eletrônica é desativada. O eixo escravo é desacelerado até parar. A chamada ativa a parada de pré-processamento.

Variante 2:

Sintaxe	Significado
`EGOFS (eixo escravo,eixo mestre 1, …,eixo mestre 5)`	Esta parametrização do comando permite eliminar **seletivamente** a influência que exerce cada eixo mestre sobre o movimento do eixo escravo.

Deve-se especificar pelo menos um eixo mestre. A influência dos eixos mestres especificados sobre o eixo escravo é desativada de forma controlada. A chamada ativa a parada de pré-processamento. Se ainda permanecerem eixos mestres ativos, então o eixo escravo continua em marcha sob esta influência. Uma vez eliminadas todas influências de eixos mestres desta forma, então o eixo escravo será desacelerado até parar.

Variante 3:

Sintaxe	Significado
`EGOFC(fuso escravo1)`	A caixa de transmissão eletrônica é desativada. O eixo escravo continua em marcha com a rotação e velocidade ativas no momento da desativação. A chamada ativa a parada de pré-processamento.

Indicação

Esta função somente é permitida para fusos.

13.4.4 Deletar a definição de uma caixa de transmissão eletrônica (EGDEL)

Um grupo de eixos EG deve estar desativado antes de sua definição ser deletada.

Programação

Sintaxe	Significado
`EGDEL(eixo escravo)`	A definição de acoplamento do grupo de eixos é deletada. Até alcançar o número máximo de grupos de eixos ativos simultaneamente pode-se definir novamente outros grupo de eixos com `EGDEF`. A chamada ativa a parada de pré-processamento.

13.4.5 Avanço por rotação (G95) / caixa de transmissão eletrônica (FPR)

Com o comando FPR também se pode especificar o eixo escravo de uma transmissão eletrônica como eixo determinante para o avanço por rotação. Para este caso se aplica o seguinte comportamento:

- O avanço depende da velocidade nominal do eixo escravo da caixa de transmissão eletrônica.
- A velocidade nominal é calculada a partir das velocidades dos fusos mestres e eixos mestres Modulo (que não são eixos de percurso) e seus fatores de acoplamento associados.
- Não são considerados os componentes de velocidade de eixos lineares e eixos que não são Modulo e os movimentos sobrepostos do eixo escravo.

13.5 Fuso sincronizado

No modo sincronizado existe um fuso mestre (LS) e um fuso escravo (FS), o assim chamado **par de fusos sincronizados**. Com o acoplamento ativo (modo sincronizado) o fuso escravo acompanha os movimentos do fuso mestre de acordo com a relação de funcionamento definida.

O par de fusos sincronizados pode ser configurado de forma permanente para qualquer máquina através dos dados de máquina específicos de canal, ou definido pelo usuário através do programa de peça CNC. Por canal NC podem ser definidos até 2 pares de fusos sincronizados simultaneamente.

A partir do programa de peça um grupo de caixa de transmissão pode ser

- definido e alterado
- ativado
- desativado
- deletado

.

Além disso, em função do software se pode

- esperar pela condição de sincronismo
- alterar o comportamento da mudança de blocos
- selecionar o tipo de acoplamento entre nominal ou real ou especificar o deslocamento angular entre fuso mestre e fuso escravo
- assumir uma programação especificada do fuso escravo ao ativar o acoplamento
- corrigir um desvio de sincronismo medido ou conhecido

.

13.5.1 Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

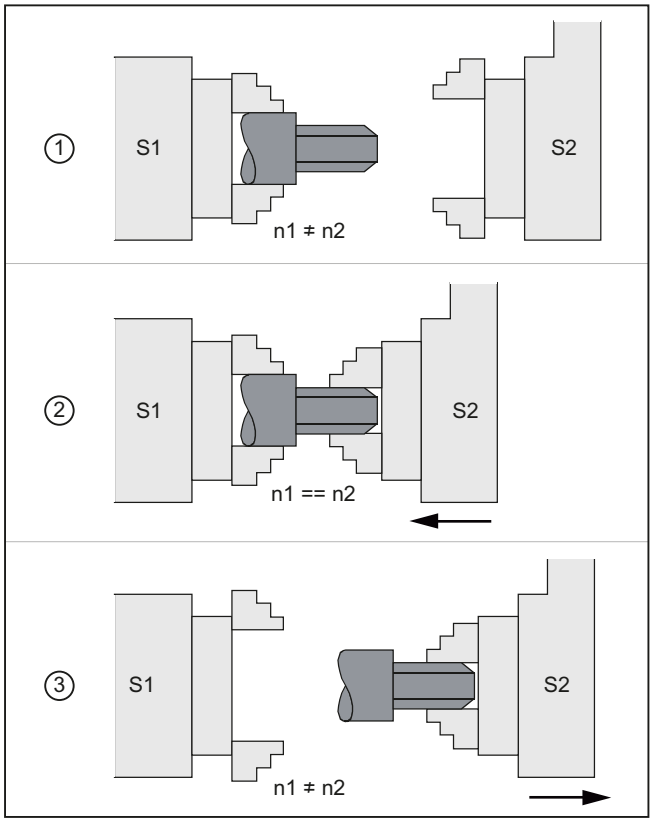
A função "fuso síncrone" possibilita um processo síncrone de número de rotações de fuso escravo (FS) e fuso mestre (LS) com relação de transmissão programável.

A função oferece os seguintes modos:

- Sincronismo de rotação ($n_{FS} = n_{LS}$)
- Sincronismo de posição ($\phi_{FS} = \phi_{LS}$)
- Sincronismo de posição com deslocamento angular ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

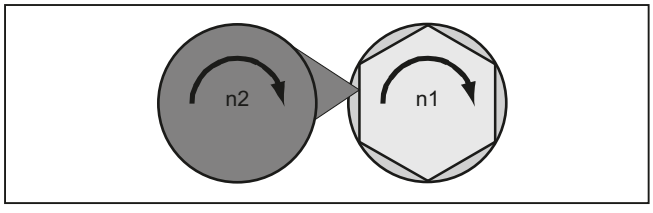
Exemplos de utilização:

- Transmissão flutuante de peças de trabalho ex. para usinagem de lado posterior, relação de transmissão: 1:1



① Sincronizar número de rotações
② Transposição da peça de trabalho
③ Usinar lado posterior

- Usinagem de cantos múltiplos (rotações poligonais), sincronicidade rotacional, relação de transmissão: n_1:n_2



Sintaxe

```
COUPDEF(<FS>,<LS>,<ZFS>,<NLS>,<troca de bloco>,<tipo de
acoplamento>)
COUPON(<FS>,<LS>,<POSFS>)
COUPONC(<FS>,<LS>)
COUPOF(<FS>,<LS>,<POSFS>,<POSLS>)
COUPOFS(<FS>,<LS>)
COUPOFS(<FS>,<LS>,<POSFS>)
```

```
COUPRES(<FS>,<LS>)
COUPDEL(<FS>,<LS>)
WAITC(<FS>,<mudança de blocos>,<LS>,<mudança de blocos>)
```

Indicação

Forma escrita abreviada

Para as instruções COUPOF, COUPOFS, COUPRES e COUPDEL também é possível uma forma escrita abreviada sem a indicação do fuso mestre.

Significado

COUPDEF:	Definição/alteração de acoplamento específica de usuário
COUPON:	Ativar acoplamento. A partir da atual rotação o fuso escravo sincroniza com o fuso mestre
COUPONC:	Aceitação do acoplamento na ativação com programação anterior do M3 S... ou M4 S.... Uma rotação diferencial do fuso escravo é incluída imediatamente.
COUPOF:	Desativação de acoplamento. • com mudança de blocos imediata: COUPOF(<S2>,<S1>) • Mudança somente depois de passar a(s) posição(ões) de desativação <POSFS> e <POSLS>: COUPOF(<S2>,<S1>,<POSFS>) COUPOF(<S2>,<S1>,<POSFS>,<POSLS>)
COUPOFS:	Desativação de um acoplamento com parada do eixo escravo. Mudança de blocos realizada o mais rápido possível com a mudança imediata: COUPOFS(<S2>,<S1>) Mudança de blocos apenas ao ultrapassar a posição de desativação: COUPOFS(<S2>,<S1>,<POSFS>)
COUPRES:	Resetamento do parâmetro de acoplamento no MD e no SD configurado
COUPDEL:	Deletação do acoplamento definido pelo usuário
WAITC:	Esperar a condição de sincronização (NOC são cancelados no IPO na mudança de blocos)
<FS>:	Denominação do fuso escravo
Parâmetros opcionais:	
<LS>:	Denominação do fuso mestre Especificação com número de fuso p. ex. S2, S1
<ZFS>, <NLS>:	Relação de transmissão entre FS e LS. <ZFS> / <NLS> = Numerador / Denominador Pré-ajuste: <ZFS> / <NLS> = 1.0 ; Especificação de denominador opcional

<mudança de blocos>:	Comportamento de mudança de blocos A relação de mudança de blocos é realizada pelo(a):	
	"NOC"	imediatamente
	"FINE"	com o alcance da "Sincronização fina"
	"COARSE"	com o alcance da "Sincronização aproximada"
	"IPOSTOP"	com alcance do IPOSTOP, isto é, após a sincronização de valor nominal (pré-ajuste)
	O comportamento de mudança de blocos está ativo modalmente.	
<tipo de acoplamento>:	Tipo de acoplamento: Acoplamento entre FS e LS	
	"DV"	Acoplamento de valor nominal (pré-ajuste)
	AV	Acoplamento de valor real
	"VV"	Acoplamento de velocidade
	O tipo de acoplamento está ativo modalmente.	
<POSFS>:	Deslocamento angular entre fuso mestre e fuso escravo	
	Faixa de valores:	0°… 359,999°
<POSFS>,<POSLS>:	Posições de desativação do fuso escravo e fuso mestre "A mudança de blocos é liberada após a ultrapassagem do POS_{FS}, POS_{LS}"	
	Faixa de valores:	0°… 359,999°

Exemplos

Trabalho com fuso mestre e fuso escravo

Código de programa	**Comentário**
	Fuso de comando = fuso mestre = fuso 1
	Fuso escravo = fuso 2
N05 M3 S3000 M2=4 S2=500	O fuso mestre gira com 3000 rpm, o fuso escravo gira com 500 rpm.
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	Definição do acoplamento (também pode ser projetada).
...	
N70 SPCON	Definição de fuso mestre no controle de posição (acoplamento de valor nominal).
N75 SPCON(2)	Definição de fuso escravo no controle de posição.
N80 COUPON(S2,S1,45)	Acoplamento flutuante na posição de deslocamento = 45 graus.
...	
N200 FA[S2]=100	Velocidade de posicionamento = 100 graus/min
N205 SPOS[2]=IC(-90)	Deslocamento de 90 graus sobrepostos no sentido negativo.
N210 WAITC(S2,"Fine")	Espera pela sincronização "fina".
N212 G1 X... Y... F...	Usinagem
...	
N215 SPOS[2]=IC(180)	Desloca 180 graus sobrepostos no sentido positivo.

Código de programa	Comentário
N220 G4 S50	Tempo de espera = 50 rotações do fuso mestre
N225 FA[S2]=0	Ativação da velocidade configurada (MD).
N230 SPOS[2]=IC(-7200)	20 rotações. Deslocamento com velocidade configurada no sentido negativo.
...	
N350 COUPOF(S2,S1)	Desacoplamento flutuante, S=S2=3000
N355 SPOSA[2]=0	Parada do FS em zero grau.
N360 G0 X0 Y0	
N365 WAITS(2)	Espera pelo fuso 2.
N370 M5	Parada do FS.
N375 M30	

Programação de uma rotação diferencial

Código de programa	Comentário
	Fuso de comando = fuso mestre = fuso 1
	Fuso escravo = fuso 2
N01 M3 S500	O fuso mestre gira com 500 rpm.
N02 M2=3 S2=300	O fuso escravo gira com 300 rpm.
...	
N10 G4 F1	Tempo de espera do fuso mestre.
N15 COUPDEF (S2,S1,-1)	Fator de acoplamento com relação de transmissão -1:1
N20 COUPON(S2,S1)	Ativar acoplamento. A rotação do fuso escravo resulta da rotação do fuso escravo e do fator de acoplamento.
...	
N26 M2=3 S2=100	Programação de uma rotação diferencial.

Exemplos da transferência de um movimento para rotação diferencial

Ativação do 1º acoplamento com programação anterior do fuso escravo com COUPON

Código de programa	Comentário
	Fuso de comando = fuso mestre = fuso 1
	Fuso escravo = fuso 2
N05 M3 S100 M2=3 S2=200	O fuso mestre gira com 100 rpm, o fuso escravo gira com 200 rpm.
N10 G4 F5	Tempo de espera = 5 segundos do fuso mestre
N15 COUPDEF(S2,S1,1)	A relação de transmissão do FS com o LS é 1,0 (pré-ajuste).
N20 COUPON(S2,S1)	Acoplamento flutuante no fuso mestre.
N10 G4 F5	O fuso escravo gira com 100 rpm.

Ativação do 2º acoplamento com programação anterior do fuso escravo com COUPONC

Código de programa	Comentário
	Fuso de comando = fuso mestre = fuso 1

Código de programa	Comentário
	Fuso escravo = fuso 2
N05 M3 S100 M2=3 S2=200	O fuso mestre gira com 100 rpm, o fuso escravo gira com 200 rpm.
N10 G4 F5	Tempo de espera = 5 segundos do fuso mestre
N15 COUPDEF(S2,S1,1)	A relação de transmissão do FS com o LS é 1,0 (pré-ajuste).
N20 COUPONC(S2,S1)	Acoplar flutuante no fuso mestre e assumir a rotação anterior para S2.
N10 G4 F5	O S2 gira com 100 rpm + 200 rpm = 300 rpm

Ativação do 3º acoplamento com fuso escravo parado com `COUPON`

Código de programa	Comentário
	Fuso de comando = fuso mestre = fuso 1
	Fuso escravo = fuso 2
N05 SPOS=10 SPOS[2]=20	Fuso escravo S2 em modo de posicionamento.
N15 COUPDEF(S2,S1,1)	A relação de transmissão do FS com o LS é 1,0 (pré-ajuste).
N20 COUPON(S2,S1)	Acoplamento flutuante no fuso mestre.
N10 G4 F1	O acoplamento é encerrado, o S2 permanece parado em 20 graus.

Ativação do 4º acoplamento com fuso escravo parado com `COUPONC`

Indicação

Modo de posicionamento ou modo de eixo

Se o fuso escravo estiver em modo de posicionamento ou em modo de eixo antes do acoplamento, então o fuso escravo se comporta de maneira igual para `COUPON(<FS>,<LS>)` e `COUPONC(<FS>,<LS>)`.

Indicação

Fuso mestre e modo de eixo

Se o fuso mestre estiver em modo de eixo antes da definição do acoplamento, vigorará, mesmo após a ativação do acoplamento, o valor limite de velocidade do dado de máquina:

MD32000 $MA_MAX_AX_VELO (velocidade de eixo máxima)

Para evitar este comportamento, o eixo deve ser ativado em modo de fuso (`M3 S...` ou `M4 S...`) antes da definição do acoplamento.

Outras informações

Acoplamento configurado

Com o acoplamento configurado os fusos mestre e escravo são definidos através de dado de máquina. Os fusos configurados não podem ser alterados no programa de peça. A parametrização do acoplamento pode ocorrer com `COUPDEF` no programa (Condição: proteção escrita não estabelecida).

Acoplamento definido pelo usuário

Com COUPDEF um acoplamento pode ser redefinido ou alterado no programa de peça. Se já existe um acoplamento ativo, este deve ser cancelado primeiro com COUPDEL, antes da definição de um novo acoplamento.

Um acoplamento é totalmente definido por:

```
COUPDEF(<FS>,<LS>,<ÜFS>,<ÜLS>, comportamento de mudança de blocos,
tipo de acoplamento)
```

Fuso escravo (FS) e fuso mestre (LS)

O acoplamento é definido de forma clara com os nomes de eixo para fuso escravo (FS) e fuso mestre (LS). Os nomes de eixo devem ser programados com cada instrução COUPDEF. Os demais parâmetros de acoplamento estão ativos modalmente e somente precisam ser programados, quando forem alterados.

Exemplo:

```
COUPDEF(S2,S1)
```

Relação de transmissão

A relação de transmissão é especificada como relação de rotação entre o fuso escravo e o fuso mestre

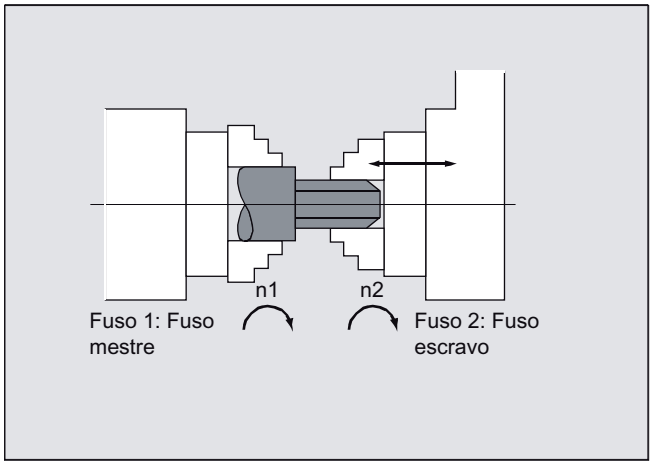
Fuso escravo / fuso mestre= numerador / denominador

O numerador deve ser programado. O denominador não deve ser programado. Para o denominador fica estabelecido o valor padrão de 1.0 .

Exemplo:

Fuso escravo S2 e fuso mestre S1, relação de transmissão = 1 / 1

```
COUPDEF(S2, S1, 1.0)
```



Indicação

A relação de transmissão também pode ser alterada com o acoplamento ativo e os fusos girando.

Comportamento de mudança de blocos NOC, FINE, COARSE, IPOSTOP

Para a programação do comportamento de mudança de blocos é possível a seguinte forma escrita abreviada:

- "NO": imediatamente (pré-ajuste)
- "FI": com o alcance da "Sincronização fina"
- "CO": com o alcance da "Sincronização aproximada"
- "IP": com alcance do IPOSTOP, isto é, após a sincronização de valor nominal

Tipo de acoplamento

Indicação

O tipo de acoplamento somente pode ser modificado com o acoplamento desativado.

Ativação do modo sincronizado COUPON, <POSFS>

- Ativação do acoplamento com qualquer deslocamento angular entre LS e FS:
  - `COUPON(S2,S1)`
  - `COUPON(S2)`
- Ativação do acoplamento com deslocamento angular <POSFS>
  <POSFS> refere-se à posição de 0° do fuso-mestre no sentido de giro positivo
  Faixa de valores <POSFS>: 0°… 359,999°
  - `COUPON(S2,S1,30)`

Indicação

O deslocamento do ângulo também pode ser alterado por acoplamento ativo.

Posicionamento do fuso escravo

Com o acoplamento de fusos sincronizados ativado, também se pode posicionar fusos escravos dentro de ±180° independentemente do movimento realizado pelo fuso mestre.

- Posicionamento do fuso FS com SPOS
  Exemplo: `SPOS[2] = IC(-90)`
  Mais informações sobre o SPOS estão disponíveis no(a):
  Literatura:
  Manual básico de programação

Rotações diferenciais

As rotações diferenciais ocorrem no modo de controle de rotação e de acoplamento do fuso síncrono ativo através da sobreposição assinada de uma rotação FS devido ao movimento LS e de uma rotação FS devido à programação do fuso:

- Acoplamento de fusos síncronos com `COUPONC`
- `S<FS>=<Rotação>` [`M<FS>=<Sentido da rotação>`]

Indicação

Condições gerais

- Com o sentido de rotação `M3` ou `M4` a rotação `S`... também deve ser reprogramada.
- Uma sobreposição da rotação do fuso (`M <sentido de rotação>` S<FS>) através do acoplamento de fuso síncrono `COUPONC` somente estará ativa se a sobreposição estiver liberada.
- A dinâmica do fuso mestre deve ser restringida de modo que no caso de uma sobreposição do fuso escravo, os valores limite de dinâmica deste último não sejam ultrapassados.

Para mais informações sobre rotação diferencial, veja:
Literatura:
Manual de funções ampliadas; Fuso sincronizado (S3)

Velocidade, aceleração: FA, ACC, OVRA, VELOLIMA

A velocidade axial e aceleração axial de um fuso escravo são programáveis com:

- `FA[SPI(S<n>)]` e `FA[S<n>]` (velocidade axial), respectivamente
- `ACC[SPI(S<n>)]` e `ACC[S<n>]` (aceleração axial), respectivamente
- `OVRA[SPI(S<n>)]` e `OVRA[S<n>]` (Override axial), respectivamente
- `VELOLIMA[SPI(S<n>)]` e `VELOLIMA[S<n>]` (aceleração e redução de velocidade axial), respectivamente

Com `<n>` = 1, 2, 3, ... (número de fuso dos fusos escravos)

Literatura:
Manual básico de programação

Indicação

Uma redução ou aumento do puxão axial máximo não é eficaz com fusos.

Mais informações sobre a dinâmica axial estão disponíveis no(a):
Literatura:
Manual de funções ampliadas; Eixos rotativos (R2)

Comportamento de mudança de blocos programável WAITC

Com o `WAITC` define-se o comportamento de mudança de blocos, por exemplo, após a alteração de parâmetros de acoplamento ou processos de posicionamento, com diferentes condições de sincronização (aproximada, fina, `IPOSTOP`). Se não for especificada nenhuma condição de sincronização, é aplicado o comportamento de mudança de blocos indicado na definição `COUPDEF`.

Exemplos

- Espera pelo alcance (preenchimento) da condição de sincronização `FINE` para fuso escravo S2 e `COARSE` para fuso escravo S4: `WAITC(S2,"FINE",S4,"COARSE")`
- Espera pelo alcance (preenchimento) da condição de sincronização de acordo com o `COUPDEF`: `WAITC( )`

Desativação de acoplamento COUPOF

Com o `COUPOF` define-se o comportamento de desativação do acoplamento:

- Desativação do acoplamento com mudança de blocos imediata:
  - `COUPOF(S2,S1)` (com indicação do fuso mestre)
  - `COUPOF(S2)` (sem indicação do fuso mestre)
- Desativação do acoplamento depois de ultrapassar as posições de desativação. A mudança de blocos é realizada depois da ultrapassagem das posições de desativação.
  - `COUPOF(S2,S1,150)` (posição de desativação do FS: 150°)
  - `COUPOF(S2,S1,150,30)` (posição de desativação do FS: 150°, LS: 30°)

Desativação do acoplamento com parada do fuso escravo COUPOFS

Com o `COUPOFS` define-se o comportamento de desativação do acoplamento com parada do fuso escravo:

- Desativação do acoplamento com parada do fuso escravo e mudança de blocos imediata:
  - `COUPOFS(S2,S1)` (com indicação do fuso mestre)
  - `COUPOFS(S2)` (sem indicação do fuso mestre)
- Desativação do acoplamento após ultrapassar as posições de desativação com parada do fuso escravo. A mudança de blocos é realizada depois da ultrapassagem das posições de desativação.
  - `COUPOFS(S2,S1,150)` (posição de desativação do FS: 150°)

Deletação de acoplamentos COUPDEL

Com o `COUPDEL` deleta-se o acoplamento:

- `COUPDEL(S2,S1)` (com indicação do fuso mestre)
- `COUPDEL(S2)` (sem indicação do fuso mestre)

Resetamento dos parâmetros de acoplamento COUPRES

Com o `COUPRES` são ativados os valores do acoplamento parametrizados nos dados de máquina e de ajuste:

- `COUPRES(S2,S1)` (com indicação do fuso mestre)
- `COUPRES(S2)` (sem indicação do fuso mestre)

Variáveis de sistema

- Atual estado de acoplamento do fuso escravo
  O estado atual de um acoplamento de um fuso escravo pode ser lido através de bits codificados:
  `<valor> = $AA_COUP_ACT[<FS>]`

Bit	<valor>	Significado
-	0	Nenhum acoplamento ativo
2	4	Acoplamento de fusos sincronizados ativo

Nota

- Todos os outros valores se relacionam com o funcionamento do eixo
- Caso o fuso seja um fuso escravo de acoplamentos múltiplos, será devolvido como valor da condição de acoplamento todos os acoplamentos como condição total.

- Atual deslocamento angular
  O atual deslocamento angular de um fuso escravo em relação ao fuso mestre pode ser lido em:
  - `$AA_COUP_OFFS[<FS>]` (deslocamento angular por parte do valor nominal)
  - `$VA_COUP_OFFS[<FS>]` (deslocamento angular por parte do valor real)

  Exemplos de aplicação
  Correção da diferença do deslocamento do ângulo no programa NC após suspensão do funcionamento condutor:
  Diferença do deslocamento do ângulo = deslocamento do ângulo programado- variável do sistema

Literatura

Para informações detalhadas sobre as variáveis de sistema, veja:

Lista de manuais Variáveis de sistema

13.6 Acoplamento genérico (CP...)

O "acoplamento genérico" é uma função geral de acoplamento na qual estão resumidas todas as características de acoplamento dos tipos de acoplamento existentes (movimento acoplado, acoplamento de valor mestre, caixa de transmissão eletrônica e fuso sincronizado).

A função permite uma forma flexível de programação:

- O utilizador pode selecionar livremente as características de acoplamento necessárias para a sua aplicação (princípio modular).
- Cada característica de acoplamento pode ser programada individualmente.
- As características de acoplamento de um acoplamento definido (p. ex. fator de acoplamento) podem ser alterados.
- É possível uma utilização posterior de outras características de acoplamento.
- O sistema de referência de coordenadas do eixo escravo (sistema de coordenadas básicas ou sistema de coordenadas da máquina) é programável.
- Certas características de acoplamento também podem ser programadas em ações sincronizadas.
  Literatura: Manual de funções para ações síncronas

Indicação

As chamadas de acoplamentos até agora para o movimento acoplado (TRAIL*), acoplamento de valor mestre (LEAD*), caixa de transmissão eletrônica (EG*) e fuso sincronizado (COUP*) continuam a ser suportados através dos ciclos de adaptação.

Vista geral de todas as palavras-chave e características de acoplamento

A tabela seguinte representa uma vista geral de todas as palavras-chave do acoplamento genérico e das características de acoplamento programáveis com o mesmo:

Palavra-chave:	Característica de acoplamento / significado	Sintaxe
`CPDEF`	Criação de um modulo de acoplamento	`CPDEF=(<FAx>)`
`CPDEL`	Exclusão de um modulo de acoplamento	`CPDEL=(<FAx>)`
`CPLA`	Definição de um eixo mestre	`CPLA[<FAx>]=(<LAx>)`

`CPLDEF`	Definição de um eixo mestre e criação de um modulo de acoplamento
	(também possível com `CPDEF` + `CPLA`)
		`CPLDEF[<FAx>]=(<LAx>)`
		ou
		`CPDEF=(<FAx>) CPLA[<FAx>]=(<LAx>)`

`CPLDEL`	Exclusão de um eixo mestre de um modulo de acoplamento
	(também possível com `CPDEL` + `CPLA`)
		`CPLDEL[<FAx>]=(<LAx>)`
		ou
		`CPDEL=(<FAx>) CPLA[<FAx>]=(<LAx>)`

Palavra-chave:	Característica de acoplamento / significado	Sintaxe		
CPON	ativação modulo de acoplagem	CPON=(<FAx>)		
CPOF	Desativação de modulo de acoplagem	CPOF=(<FAx>)		
CPLON	ativação de um eixo mestre de um modulo de acoplagem	CPLON[<FAx>]=<LAx>		
CPLOF	Desativação de um eixo mestre de um modulo de acoplagem	CPLOF[<FAx>]=<LAx>		
CPLNUM	Nominador do fator de acoplagem	CPLNUM[FAx,LAx]=<valor>		
CPLDEN	Denominador de um fator de acoplagem	CPLDEN[FAx,LAx]=<valor>		
CPLCTID	Número da tabela de curvas	CPLCTID[FAx,LAx]=<valor>		
CPLSETVAL	referência de acoplagem	CPLSETVAL[FAx,LAx]="<referência de acoplamento>"		
		"<referência de acoplamento>":	"CMDPOS"	Acoplamento do valor nominal
			"CMDVEL"	Acoplamento de velocidade
			"ACTPOS"	Acoplamento de valor real
CPFRS	sistema de referência de coordenadas	CPFRS[FAx]="<referência de coordenadas>"		
		"<referência de coordenadas>":	"BCS"	Sistema de coordenadas básico
			"MCS"	Sistema de coordenadas da máquina
CPBC	Critério de mudança de blocos	CPBC[FAx]="<critério de mudança de blocos>"		
		"<critério de mudança de blocos>":	"NOC"	Mudança de blocos independentemente de estado de acoplamento.
			"IPOSTOP"	A mudança de blocos é realizada com sincronização de valor nominal.
			"COARSE"	A mudança de blocos é realizada com sincronização de valor real "aproximado".
			"FINE"	A mudança de blocos é realizada com sincronização de valor real "preciso".
CPFPOS + CPON	Posição síncrona do eixo-escravo na ativação	CPON=FAx CPFPOS[FAx]=<valor>		
CPLPOS + CPON	Posição síncrona do eixo-mestre na ativação	CPLPOS[FAx,LAx]=<valor>		

Palavra-chave:	Característica de acoplamento / significado	Sintaxe		
CPFMSON	modo de sincronização	CPFMSON[FAx]="<modo de sincronização>"		
		"<modo de sincronização>":	"CFAST"	O acoplamento é fechado com otimização de tempo.
			"CCOARSE"	O acoplamento é desligado apenas quando a posição do eixo escravo se encontra na área da posição atual do eixo escravo, conforme as leis do acoplamento.
			"NTGT"	A próxima folga entre dentes é aproximada com tempo otimizado.
			"NTGP"	A próxima folga de dentes é aproximada com percurso otimizado.
			"NRGT"	O próximo segmento é aproximado com tempo otimizado, conforme a relação da quantidade de passos com o número de dentes.
			"NRGP"	O próximo segmento é aproximado com percurso otimizado, conforme a relação da quantidade de passos com o número de dentes.
			"ACN"	Apenas para eixos rotativos! O eixo rotativo desloca-se no sentido negativo do eixo para a posição síncrona. A sincronização ocorre imediatamente.
			"ACP"	Apenas para eixos rotativos! O eixo rotativo desloca-se no sentido positivo do eixo para a posição síncrona. A sincronização ocorre imediatamente.
			"DCT"	Apenas para eixos rotativos! O eixo rotativo desloca-se otimizado em tempo para a posição síncrona pen. A sincronização ocorre imediatamente.
			"DCP"	Apenas para eixos rotativos!

Palavra-chave:	Característica de acoplamento / significado	Sintaxe		
				O eixo rotativo desloca-se otimizado em curso para a posição síncrona programada. A sincronização ocorre imediatamente.
CPFMON	Ação do eixo escravo na ativação	CPFMON[FAx]="<comportamento de ativação>"		
		"<comportamento de ativação>":	"STOP"	Apenas para fusos! Um movimento ativo dos fusos escravos é parado antes da ativação.
			"CONT"	Apenas para fusos e eixos principais! O movimento atual do eixo/fuso escravo é assumido no acoplamento como movimento inicial.
			"ADD"	Apenas para fusos! As partes de movimentos do acoplamento atuam ainda sobre o movimento atual sobreposto, ou seja, o movimento atual do eixo/fuso escravo é mantido como movimento sobreposto.
CPFMOF	Ação do eixo escravo na desativação total	CPFMOF[FAx]="<comportamento de desativação>"		
		"<comportamento de desativação>":	"STOP"	Paragem do eixo/fuso escravo. Um movimento sobreposto ativo é desacelerado até a imobilização. Em seguida, o acoplamento é aberto
			"CONT"	Apenas para fusos e eixos principais! O eixo escravo continua em marcha com a rotação e velocidade ativas no momento da desativação.
CPFPOS + CPOF	Posição de desativação do eixo escravo durante a desativação	CPOF=(FAx) CPFPOS[FAx]=<valor>		

Palavra-chave:	Característica de acoplamento / significado	Sintaxe		
CPMRESET	ação de acoplagem no RESET	CPMRESET[FAx]="<comportamento de reset>"		
		"<comportamento de reset>":	"NONE"	O estado atual do acoplamento é mantido.
			"ON"	Se o respectivo módulo de acoplamento tiver sido criado, então o acoplamento será ativado. Serão ativadas todas as relações do eixo mestre definidas. Isto também ocorre, quanto todas ou partes destas relações do eixo mestre estiverem ativas, isto é, no caso de um acoplamento totalmente ativado também ocorre uma nova sincronização.
			"OF"	Um movimento sobreposto ativo é desacelerado até a imobilização. Em seguida, o acoplamento é desativado. Se o respectivo módulo de acoplamento tiver sido criado sem a definição explícita (CPDEF), então o módulo de acoplamento será excluído. Em outro caso, ele permanece criado, isto é, ele pode continuar a ser utilizado.
			"OFC"	Apenas possível com fusos! O eixo escravo continua em marcha com a rotação e velocidade ativas no momento da desativação. O acoplamento será desativado. Se o respectivo módulo de acoplamento tiver sido criado sem a definição explícita (CPDEF), então o módulo de acoplamento será excluído. Em outro caso, ele permanece criado, isto é, ele pode continuar a ser utilizado.
			"DEL"	Um movimento sobreposto ativo é desacelerado até a imobilização. Em seguida, o acoplamento é desativado e, por fim, excluído.

Palavra-chave:	Característica de acoplamento / significado	Sintaxe		
			"DELC"	Apenas possível com fusos! O eixo escravo continua em marcha com a rotação e velocidade ativas no momento da desativação. O acoplamento é desativado e, por fim, apagado.
CPMSTART	ação de acoplagem no inicio do programa de usinagem	CPMSTART[FAx]="<comportamento de início>"		
		"<comportamento de início>":	"NONE"	O estado atual do acoplamento é mantido.
			"ON"	Acoplamento ativado. Serão ativadas todas as relações do eixo mestre definidas. Isto também ocorre, quanto todas ou partes destas relações do eixo mestre estiverem ativas, isto é, no caso de um acoplamento totalmente ativado também ocorre uma nova sincronização.
			"OF"	O acoplamento será desativado. Se o respectivo módulo de acoplamento tiver sido criado sem a definição explícita (CPDEF), então o módulo de acoplamento será excluído. Em outro caso, ele permanece criado, isto é, ele pode continuar a ser utilizado.
			"DEL"	O acoplamento é desativado e, por fim, apagado.
CPMPRT	ação do acoplamento no início do programa de usinagem sob bloco de pesquisa via teste de programa	CPMPRT[FAx]="<comportamento de início>"		
		"<comportamento de início>":	ver CPMSTART	
CPLINTR	Valor de deslocamento para o valor inicial do eixo-mestre	CPLINTR[FAx,LAx]=<valor>		
CPLINSC	Fator de escala para o valor inicial do eixo-mestre	CPLINSC[FAx,LAx]=<valor>		
CPLOUTTR	valor de deslocamento para o valor final de um acoplamento	CPLOUTTR[FAx,LAx]=<valor>		
CPLOUTSC	fator de escala para o valor final de um acoplamento	CPLOUTSC[FAx,LAx]=<valor>		
CPSYNCOP	limiar para o andamento síncrono de posição "amplo"	CPSYNCOP[FAx]=<valor>		

Palavra-chave:	Característica de acoplamento / significado	Sintaxe		
CPSYNFIP	limiar para o andamento síncrono de posição "preciso"	CPSYNFIP[FAx]=<valor>		
CPSYNCOP2	Segundo limiar para o andamento síncrono de posição "amplo"	CPSYNCOP2[FAx]=<valor>		
CPSYNFIP2	Segundo limiar para o andamento síncrono de posição "preciso"	CPSYNFIP2[FAx]=<valor>		
CPSYNCOV	limiar para o andamento síncrono de velocidade "amplo"	CPSYNCOV[FAx]=<valor>		
CPSYNFIV	limiar para o andamento síncrono de velocidade "suave"	CPSYNFIV[FAx]=<valor>		
CPMBRAKE	Ação do eixo escrava em específicos sinais e comandos de parada	CPMBRAKE[FAx]=<valor codificado por Bits>		
CPMVDI	Ação do eixo escravo de específicos sinais de interface NC/PLC	CPMVDI[FAx]=<valor codificado por Bits>		
CPMALARM	supressão de saída do alarme especialmente referente à acoplagem	CPMALARM[FAx]=<valor codificado por Bits>		
CPSETTYPE	Tipo de acoplamento	CPSETTYPE[FAx]="<tipo de acoplamento>"		
		"<tipo de acoplamento>":	"CP"	Programação livre
			"TRAIL"	Tipo de acoplamento "movimento acoplado"
			"LEAD"	Tipo de acoplamento "acoplamento de valor mestre"
			"EG"	Tipo de acoplamento "caixa de transmissão eletrônica"
			"COUP"	Tipo de acoplamento "fuso sincronizado"

FAx: : Eixo/fuso escravo

LAx: : Eixo/fuso mestre

Indicação

Características de acoplamento que não são programadas (no programa de peça ou em ações de sincronização) tornam-se válidas com os seus ajustes padrão.

Dependendo do ajuste da palavra-chave CPSETTYPE, também podem tornar-se válidas características pré-definidas de acoplamento em vez dos ajustes padrão (CPSETTYPE="CP").

Literatura

Para informações detalhadas sobre o acoplamento genérico, ver:

- Manual de funcionamento, funções especiais; M3: Acoplamentos de eixos, capítulo: "Acoplamento genérico"

13.7 Acoplamento-mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

O "acoplamento-mestre/escravo" possibilita:

- O acoplamento dos eixos escravos em seu eixo-mestre é realizado apenas com a imobilização dos eixos envolvidos.
- o acoplar e desacoplar dos fusos giratórios.
- a projeção dinâmica.

Indicação

Operação de posicionamento

Nos eixos e fusos em modo de posicionamento, o acoplamento somente é encerrado e separado em estado parado.

Sintaxe

```
MASLON(<escravo_1>,<escravo_2>,...)
MASLOF(<escravo_1>,<escravo_2>,...)
MASLOFS(<escravo_1>,<escravo_2>,...)
```

Projeção dinâmica:

```
MASLDEF(<escravo_1>,<escravo_2>,...,<Mestre>)
MASLDEL(<escravo_1>,<escravo_2>,...)
```

Significado

MASLON:	Ativar um acoplamento-mestre/escravo temporário	
	<Slave_x>,...:	Eixo Slave 1 ... n
MASLOF:	Separar um acoplamento ativo mestre/escravo	
	<escravo_1>,...:	Eixo Slave 1 ... n
MASLOFS:	Separar acoplamento-mestre/escravo e frear os fusos-escravo automaticamente (veja observação "condução no acoplamento de fusos em modo de rotação"!)	
	<escravo_1>,...:	Eixo Slave 1 ... n
MASLDEF:	Aplicar/modificar grupo-mestre/escravo do programa de peças	
	<escravo_1>,...:	Eixo Slave 1 ... n
	<Mestre>:	Eixo-mestre
MASLDEL:	Separar o grupo de eixos-mestre/escravo e deletar a definição do grupo	
	<escravo_1>,...:	Eixo Slave 1 ... n
	Observação: As definições de mestre/escravo acordadas nos dados da máquina são conservadas.	

Indicação

Condução no acoplamento de fusos em modo de rotação

A condução do acoplamento nos fusos em modo de rotação MASLON, MASLOF, MASLOFS e MASLDEL é explicitamente determinado pelos dados da máquina:

MD37263 $MA_MS_SPIND_COUPLING_MODE

No ajuste padrão com MD37263 = 0 o acoplamento e a separação dos eixos escravos são realizados exclusivamente com os eixos participantes parados. MASLOFS corresponde ao MASLOF.

Com MD37263 = 1 executa-se imediatamente a instrução de acoplamento e também sua movimentação. O acoplamento é imediatamente encerrado com MASLON e imediatamente separado com MASLOFS ou MASLOF. Os fusos-escravo que estão em rotação neste momento mantêm MASLOF suas rotações até a reprogramação da rotação. Ao contrário, o MASLOFS é freado automaticamente.

Indicação

Com MASLOF/MASLOFS é suprimida a parada de pré-processamento implícita. Condicionadas à falta de parada de pré-processamento, as variáveis de sistema $P não enviam valores atualizados aos eixos escravos até o momento de uma nova programação.

Indicação

O valor real para o eixo escravo pode ser sincronizado com PRESETON no mesmo valor do eixo mestre. Para isso, o acoplamento permanente de mestre e escravo deve ser desativado brevemente para ajustar o valor real do eixo escravo não referenciado com o valor do eixo mestre com o Power On. Em seguida é restabelecido o acoplamento permanente.

O acoplamento-mestre/escravo permanente é ativado com o ajuste do dado da máquina MD:
MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1
e não influi nos comandos de linguagem do acoplamento temporário.

Exemplos

Exemplo 1: Definição do valor real no caso do eixo Slave em um acoplamento mestre/escravo

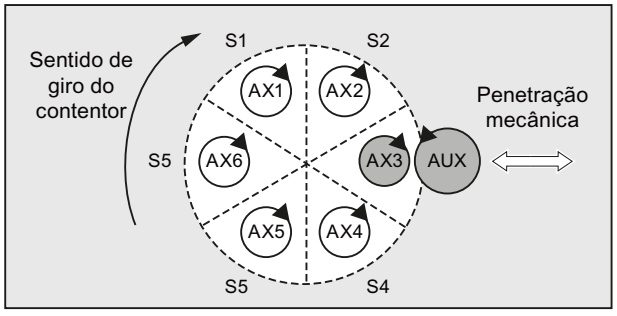
No caso de um acoplamento mestre/escravo permanente, com PRESETON é definido o valor real do eixo escravo sobre o valor do eixo mestre.

Código de programa	Comentário
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; Desativar o acoplamento permanente do eixo escravo
NEWCONF	; Ativar a alteração dos dados da máquina
STOPRE	
MASLOF(Y1)	; Desativar o acoplamento temporário
PRESETON(AX2,$VA_IM(M_AX))	; Valor real do eixo escravo = valor real do eixo mestre

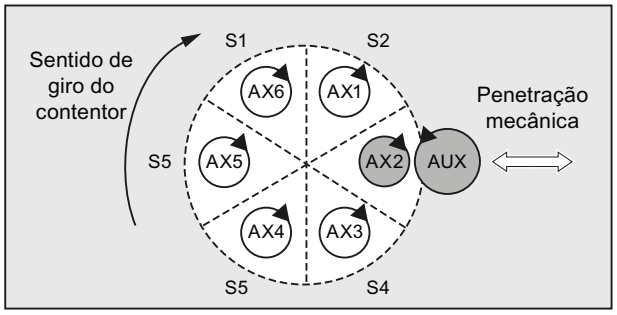
Código de programa	Comentário
`$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1`	`; Ativar o acoplamento permanente do eixo escravo`
`NEWCONF`	`; Ativar a alteração dos dados da máquina`

Exemplo 2: Configuração dinâmica de um acoplamento mestre/escravo a partir do programa de peça

Para permitir o restabelecimento do acoplamento com outro fuso após o giro do contentor do eixo, primeiro o acoplamento antigo deve ser separado, a configuração ser deletada e ser configurado o novo acoplamento.



Esquema 13-1 Antes do giro do contentor do eixo



Esquema 13-2 Após o giro do contentor do eixo em um Slot

Código de programa	Comentário
`MASLDEF(AUX,S3)`	`; AUX: Slave, S3: Master = AX3`
`MASLON(AUX)`	`; Acoplamento ativado`
`M3=3 S3=4000`	`; Girar o mestre`
`MASLDEL(AUX)`	`; Separar e deletar o acoplamento`
`AXCTSWE(CT1)`	`; Liberar o giro do contentor de eixo`
`MASLDEF(AUX,S3)`	`; AUX: Slave, S3: Master = AX2`

Literatura

- Manual de funcionamento, funções especiais, capítulo "TE3: Acoplamento de rotação/ torque, mestre/escravo"
- Manual de funções ampliadas, capítulo "B3: Sistemas descentralizados - somente 840D sl" > "NCU-Link" > "Contentor de eixo"

Ações síncronas

14

14.1 Definição de uma ação síncrona

Uma ação síncrona é definida em um bloco de um programa de peças. Dentro deste bloco não pode ser programado nenhum outro comando que não seja componente da ação síncrona.

Uma ação síncrona consiste dos seguintes componentes:

Validade, Nº de identificação (opcional)	Peça de operação (opcional)			Peça de ação		
	Frequência	Comando G (opcional)	Condição	Palavra-chave:	Comando G (opcional)	Ações
--- [1] ID=<Nr> IDS=<Nr>	--- [1] WHENEVER FROM WHEN EVERY	G...	Termo lógico	DO	G...	Ação 1 ... Ação n

[1] não programado

Sintaxe

```
DO <Aktion 1> ... <Aktion n>
<Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ... <Aktion n>
ID=<Nr> <Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ...
<Aktion n>
IDS=<Nr> <Häufigkeit> [<G-Funktion>] <Bedingung> DO <Aktion 1> ...
<Aktion n>
```

Literatura

Uma descrição detalhada da funcionalidade das ações síncronas encontra-se no:

Manual de funcionamento para ações síncronas

Oscilação 15

15.1 Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Um eixo oscilante alterna seu movimento entre os dois pontos de reversão 1 e 2 com o avanço especificado, até o movimento alternado ser desativado.

Os demais eixos podem ser interpolados quantas vezes for necessário durante o movimento alternado. Uma penetração contínua é alcançada através de um movimento de percurso ou com um eixo de posicionamento. Entretanto, neste caso não existe **nenhuma relação** entre os movimentos alternado e de penetração.

Propriedades da oscilação assíncrona

- A oscilação assíncrona está ativa em função dos eixos, além dos limites de bloco.
- Através do programa de peça se garante uma ativação do movimento alternado sincronizada com os blocos.
- Uma interpolação comum de vários eixos e uma sobreposição de trechos de oscilação não são possíveis.

Programação

Através dos seguintes comandos é possível fazer uma ativação e controle da oscilação assíncrona durante a execução do programa NC a partir do programa de peça.

Os valores programados são registrados nos respectivos dados de ajustes de modo sincronizadamente aos blocos no processamento principal e permanecem ativos até a próxima alteração.

Sintaxe

```
OSP1[<eixo>]=<valor> OSP2[<eixo>]=<valor>
OST1[<eixo>]=<valor> OST2[<eixo>]=<valor>
FA[<eixo>]=<valor>
OSCTRL[<eixo>]=(<opção de definição>,<opção de resetamento>)
OSNSC[<eixo>]=<valor>
OSE[<eixo>]=<valor>
OSB[<eixo>]=<valor>
OS[<eixo>]=1
OS[<eixo>]=0
```

Significado

<eixo>:	Nome do eixo oscilante		
OS:	Ativação/desativação da oscilação		
	Valor:	1	**A**tivação da oscilação
		0	**Desa**tivação da oscilação

OSP1:	Definição da posição do ponto de reversão 1		
OSP2:	Definição da posição do ponto de reversão 2 **Nota:** Se nenhum deslocamento incremental estiver ativo, então a posição incremental será calculada para a última posição de reversão correspondente programada no programa NC.		
OST1:	Definição de tempo de parada no ponto de reversão 1 em [s]		
OST2:	Definição de tempo de parada no ponto de reversão 2 em [s]		
	<valor>:	-2	A interpolação é continuada sem espera pela parada exata
		-1	Espera pela parada exata aproximada
		0	Espera pela parada exata fina
		>0	Espera pela parada exata fina e em seguida espera pelo tempo de parada especificado **Nota:** A unidade do tempo de espera é idêntica à do tempo de parada programado com G4.
FA:	Definição de velocidade de avanço Como velocidade de avanço se aplica a que foi definida para o eixo de posicionamento. Se nenhuma velocidade de avanço foi definida, será aplicado o valor armazenado no dado de máquina.		

OSCTRL:	Indicação de opções de definição e de resetamento Os valores de opção 0 - 3 classificam o comportamento nos pontos de reversão na desativação. Pode ser selecionada uma das variantes 0 - 3. As demais configurações podem ser combinadas conforme necessidade com a variante selecionada. Várias opções são anexadas através de sinais positivos (+).		
	<valor>:	0	Ao ser desativado o movimento oscilante, a parada ocorre no próximo ponto de reversão (pré-ajuste) **Nota:** Possível somente com o resetamento dos valores 1 e 2.
		1	Quando o movimento alternado é desativado, parar no ponto de reversão 1
		2	Quando o movimento alternado é desativado, parar no ponto de reversão 2
		3	Quando o movimento alternado é desativado, não aproximar nenhum ponto de reversão, isto se nenhum curso de passada final foi programado
		4	Após a passada final aproximar a posição final
		8	Se o movimento oscilante for cancelado através da anulação de curso restante, os cursos de passada final devem ser executados em seguida e, se necessário, executada a aproximação da posição final.
		16	Se o movimento oscilante é cancelado através da anulação de curso restante, a posição de reversão correspondente deve ser aproximada como no caso da desativação.
		32	O avanço alterado só estará ativo a partir do próximo ponto de reversão
		64	FA igual a 0, FA = 0: A sobreposição de cursos está ativa FA igual a 0, FA <> 0: A sobreposição de velocidade está ativa
		128	Para eixo rotativo DC (curso mais curto)
		256	O curso de passada final é executado como curso duplo.(padrão) 1=curso de passada final é executado como curso simples.
		512	Primeiro aproximar a posição de partida
OSNSC:	Definição do número de cursos de passada final		
OSE:	Definição da posição final (no WCS), que deve ser aproximada depois de ser desativado o movimento oscilante **Nota:** no caso de programação de "OSE", para "OSCTRL" torna-se eficaz implicitamente a opção 4.		
OSB:	Definição da posição de partida (no WCS), que deve ser aproximada antes de ser ativado o movimento oscilante A posição de partida é aproximada antes do ponto de reversão 1. Quando a posição de partida coincide com a posição de reversão 1, então a aproximação ocorre na próxima posição de reversão 2. Ao ser alcançada a posição de partida nenhum tempo de parada entra em ação, mesmo se a posição de partida coincidir com a posição de reversão 1, ao invés disso, espera-se a parada exata fina. Uma condição de parada exata ajustada será mantida. **Nota:** Para que a posição de partida seja aproximada, no dado de ajuste SD43770 $SA_OSCILL_CTRL_MASK deve estar definido o Bit 9.		

Exemplos

Exemplo 1: O eixo oscilante deve oscilar entre dois pontos de reversão

O eixo oscilante Z deve oscilar entre as posições 10 e 100. O ponto de reversão 1 deve ser aproximado com parada exata fina, e o ponto de reversão 2 com parada exata aproximada. O avanço para o eixo oscilante deve ser 250. No final da usinagem devem ser executados 3 cursos de passada final e o eixo oscilante deve aproximar-se da posição final 200. O avanço para o eixo de penetração deve ser 1, o fim da penetração no sentido X deve ser alcançado na posição 15.

Código de programa	**Comentário**
`WAITP(X,Y,Z)`	`; Posição de saída.`
`G0 X100 Y100 Z100`	`; Comutação para o modo de eixo de posicionamento.`
`WAITP(X,Z)`	
`OSP1[Z]=10 OSP2[Z]=100`	`; Ponto de reversão 1, ponto de reversão 2.`
`OSE[Z]=200`	`; Posição final.`
`OST1[Z]=0 OST2[Z]=-1`	`; Tempo de parada em U1: Parada exata fina`
	`; Tempo de parada em U2: Parada exata aproximada`
`FA[Z]=250 FA[X]=1`	`; Avanço do eixo oscilante, avanço do eixo de penetração.`
`OSCTRL[Z]=(4,0)`	`; Opções de definição.`
`OSNSC[Z]=3`	`; 3 cursos de passada final.`
`OS[Z]=1`	`; Partida da oscilação.`
`WHEN $A_IN[3]==TRUE DO DELDTG(X)`	`; Anulação de curso restante.`
`POS[X]=15`	`; Posição de saída do eixo X`
`POS[X]=50`	`; Posição final do eixo X.`
`OS[Z]=0`	`; Parada da oscilação.`
`M30`	

Indicação

A sequência de comandos "OSP1[Z]=..." até "OSNCS[Z]=..." também pode ser programada em um bloco.

Exemplo 2: Oscilação com alteração Online da posição de reversão

Os dados de ajuste necessários para a oscilação assíncrona podem ser ajustados no programa de peça.

Se os dados de ajuste forem escritos diretamente no programa de peça, então a alteração estará ativa no momento do pré-processamento. O comportamento síncrono pode ser obtido com uma parada de pré-processamento (`STOPRE`).

Código de programa	**Comentário**
`$SA_OSCILL_REVERSE_POS1[Z]=-10`	
`$SA_OSCILL_REVERSE_POS2[Z]=10`	

Código de programa	Comentário
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0	; Se o valor real do eixo oscilante ultrapassou o ponto de reversão, então o eixo de penetração será parado.
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; Ativação da oscilação.
OS[Z]=0	; Desativar a oscilação.
M30	

Outras informações

Eixo oscilante

Para o eixo oscilante aplica-se:

- Todo eixo pode ser utilizado como eixo oscilante.
- Vários eixos oscilantes podem estar ativos ao mesmo tempo (máximo: número de eixos de posicionamento).
- Para o eixo oscilante sempre está ativa a interpolação linear `G1`, independentemente do atual código G.

O eixo oscilante pode:

- ser eixo de entrada para a transformação dinâmica
- ser eixo guia para eixos Gantry e eixos acoplados
- no deslocamento:
  - sem limitação de solavancos "BRISK"
    ou
  - com limitação de solavancos "SOFT"
    ou
  - com curva característica de aceleração dobrada (como eixos de posicionamento)

Pontos de reversão da oscilação

Para definição das posições oscilantes devem ser observados os atuais deslocamentos:

- Especificação absoluta
  "OSP1[Z]=<valor>"
  Posição do ponto de reversão = Soma dos deslocamentos + valor programado
- Especificação relativa
  "OSP1[Z]=IC(<valor>)"
  Posição do ponto de reversão = Ponto de reversão 1 + valor programado

Exemplo:

Código de programa
N10 OSP1[Z] = 100 OSP2[Z] = 110
...

Código de programa
`N40 OSP1[Z] = IC(3)`

WAITP

Se a oscilação deve ser realizada por um eixo geométrico, então este deve ser liberado com "WAITP" para oscilação.

Com WAITP, depois de finalizada a oscilação, o eixo oscilante será registrado novamente como eixo de posicionamento e poderá ser utilizado normalmente.

Oscilação com ações síncronas de movimento e tempos de parada

Depois de expirar os tempos de parada ajustados, é realizada uma mudança interna de blocos durante a oscilação (visível nos novos cursos dos eixos). Na mudança de blocos é verificada a função de desativação. Neste caso é definida a função de desativação, após o ajuste de controle da sequência de movimento (`OSCTRL`). *Este comportamento de tempo pode ser controlado pelo Override de avanço.*

Talvez depois disso ainda seja executado um curso oscilante antes dos cursos de passada final serem iniciados ou da posição final ser aproximada. *Neste caso temos a impressão que o comportamento de desativação mudou. Mas este não é o caso.*

15.2 Oscilação controlada através de ações síncronas (OSCILL)

Neste tipo de oscilação apenas é permitido um movimento de penetração nos pontos de reversão ou dentro de áreas de reversão definidas.

Dependendo da necessidade, o movimento da oscilação pode, durante a penetração,

- ser continuado ou
- parado até que a penetração seja totalmente executada.

Sintaxe

1. **Definição do parâmetro para a oscilação**
2. **Definição de ações síncronas de movimentos**
3. **Atribuição de eixos, definição da penetração**

Significado

`OSP1[<eixo oscilante>]=`	Posição do ponto de reversão 1
`OSP2[<eixo oscilante>]=`	Posição do ponto de reversão 2
`OST1[<eixo oscilante>]=`	Tempo de parada no ponto de reversão 1 em segundos
`OST2[<eixo oscilante>]=`	Tempo de parada no ponto de reversão 2 em segundos
`FA[<eixo oscilante>]=`	Avanço do eixo oscilante
`OSCTRL[<eixo oscilante>]=`	Opções de definição e de resetamento
`OSNSC[<eixo oscilante>]=`	Número de cursos de passada final
`OSE[<eixo oscilante>]=`	Posição final
`WAITP(<eixo oscilante>)`	Liberação do eixo para a oscilação

Atribuição de eixos, penetração

```
OSCILL[<eixo oscilante>]=(<eixo de penetração 1>,<eixo de
penetração 2>,<eixo de penetração 3>)
POSP[<eixo de penetração>]=(<posição final>,<comprimento
parcial>,<modo>)
```

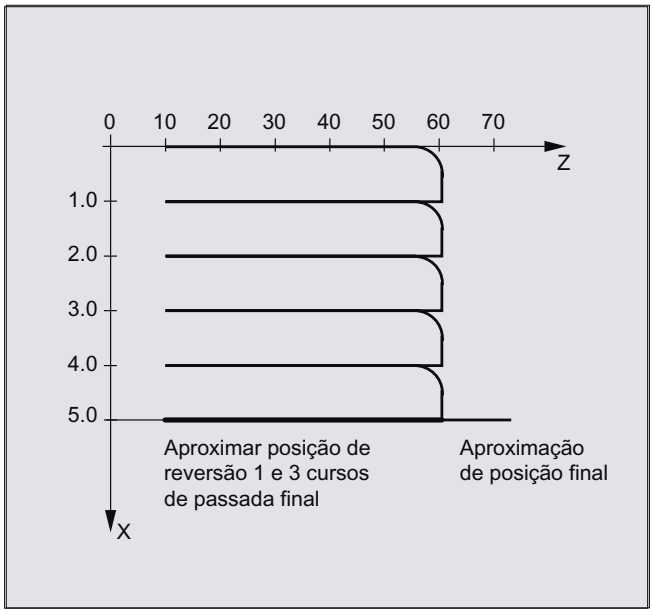
`OSCILL`:	Atribuir eixo(s) de penetração para eixo oscilante
`POSP`:	Definição das penetrações totais e parciais (veja capítulo Gerenciamento de arquivos e programas)
`Posição final`:	Posição final do eixo de penetração, depois de todas penetrações parciais serem executadas.
`Comprimento parcial`:	Tamanho da penetração parcial no ponto/área de reversão
`Modo`:	Divisão da penetração total em penetrações parciais = dois passos restantes de mesmo tamanho (pré-ajuste); = todas as penetrações parciais de mesmo tamanho

Ações sincronizadas de movimentos

WHEN… … DO	quando…, então…
WHENEVER … DO	sempre que…, então…

Exemplo

No ponto de reversão 1 não deve ser realizada nenhuma penetração. Para o ponto de reversão 2 a penetração deve ser realizada na distância ii2 antes do ponto de reversão 2 e o eixo oscilante não deve esperar no ponto de reversão pela finalização da penetração parcial. O eixo Z é o eixo oscilante e o eixo X é o eixo de penetração.



1. Parâmetro para a oscilação

Código de programa	**Comentário**
DEF INT ii2	; Definição da variável para a área de reversão 2
OSP1[Z]=10 OSP2[Z]=60	; Definição do ponto de reversão 1 e 2
OST1[Z]=0 OST2[Z]=0	; Ponto de reversão 1: Parada exata fina
	Ponto de reversão 2: Parada exata fina
FA[Z]=150 FA[X]=0.5	; Avanço do eixo oscilante Z, avanço do eixo de penetração X
OSCTRL[Z]=(2+8+16,1)	; Desativação do movimento alternado no ponto de reversão 2; após o RWL realização da passada final e aproximação da posição final; após o RWL aproximação da respectiva posição de reversão
OSNC[Z]=3	; Cursos de passada final
OSE[Z]=70	; Posição final = 70
ii2=2	; Ajuste da área de reversão
WAITP(Z)	; Permite a oscilação do eixo Z

2. Ação sincronizada de movimentos

Código de programa	Comentário
`WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z] DO ->` `$AA_OVR[X]=0 $AC_MARKER[0]=0`	; Sempre que a posição atual do eixo oscilante Z em MCS for menor que o início da área de reversão 2, então defina o Override axial do eixo de penetração X em 0% e o marcador com o índice 0 para o valor 0.
`WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0`	; Sempre que a posição atual do eixo oscilante Z em MCS for maior ou igual à posição de reversão 2, então defina o Override axial do eixo oscilante Z em 0%.
`WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0] = 1`	; Sempre que o curso restante da penetração parcial for igual a 0, então defina o marcador com o índice 0 para o valor 1.
`WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100`	; Sempre que o marcador com o índice 0 for igual a 1, então defina o override axial do eixo de penetração X para 0%. Com isso é evitada uma penetração muito antecipada (o eixo oscilante Z ainda não saiu da área de reversão 2, mas o eixo de penetração X está pronto para uma nova penetração). Ajuste o override axial do eixo oscilante Z de 0% (ação da 2ª ação síncrona) para o deslocamento novamente para 100%.

-> deve ser programado em um bloco

3. Início da oscilação

Código de programa	Comentário
`OSCILL[Z]=(X) POSP[X]=(5,1,1)`	; Partida dos eixos Para o eixo oscilante Z é atribuído o eixo X como eixo ;de penetração. O eixo X deve movimentar-se até a posição final 5 em passos de 1.
`M30`	; fim do programa

Outras informações

1. **Definir parâmetro de oscilação**
   Antes do bloco de movimento que contém tanto a atribuição do eixo de penetração e do eixo oscilante como a definição da penetração, devem ser definidos os parâmetros da oscilação (veja "Oscilação assíncrona").
2. **Definir ações sincronizadas de movimentos**
   Através das condições de sincronização ocorre:
   Supressão da penetração até que o eixo oscilante estiver dentro de uma área de reversão (ii1, ii2) ou em um ponto de reversão (U1, U2).
   Parar o **movimento alternado** no ponto de reversão durante a penetração.
   Reiniciar o **movimento alternado** depois de concluir a penetração parcial.
   Definir o início da próxima penetração parcial.
3. **Atribuir eixo oscilante e eixo de penetração** assim como definir a **penetração total e parcial**.

Definir parâmetro de oscilação

Atribuição do eixo oscilante e eixo de penetração: OSCILL

```
OSCILL[<eixo oscilante>]=(<Sentido de penetração1>,<Sentido de
penetração2>,<Sentido de penetração3>)
```

Com o comando "OSCILL" são realizadas as atribuições de eixo e a partida do movimento alternado.

Podem ser atribuídos no máximo 3 eixos de penetração para um eixo oscilante.

Indicação

Antes da partida da oscilação devem ser definidas as condições de sincronização para a relação dos eixos.

Definir penetrações: POSP

```
POSP[<eixo de penetração>]=(<posição final>,<comprimento
parcial>,<modo>)
```

Com o comando "POSP" são comunicados ao comando:

- Penetração total (através da posição final)
- O tamanho da respectiva penetração parcial no ponto de reversão ou na área de reversão
- O comportamento da penetração parcial ao alcançar a posição final (através do modo)

Modo = 0	Para as duas últimas penetrações parciais é realizada uma divisão dos cursos restantes até o ponto de destino em dois passos restantes de mesmo tamanho (pré-ajuste).
Modo = 1	Todas penetrações parciais são de mesmo tamanho. Elas são calculadas a partir da penetração total.

Definir ações sincronizadas de movimentos

As ações sincronizadas de movimentos mencionadas a seguir são utilizadas de forma bem geral na oscilação.

Aqui encontramos exemplos para solução em determinadas solicitações que servirão como módulos para a criação de movimentos alternados específicos de usuário.

Indicação

Em casos específicos, as condições de sincronização também podem ser programadas de outra forma.

Palavras-chave

`WHEN … DO …`	quando…, então…
`WHENEVER … DO`	sempre que…, então…

Funções

Com os meios de linguagem descritos detalhadamente a seguir, as seguintes funções podem realizar:

1. Penetração no ponto de reversão.
2. Penetração na área de reversão.
3. Penetração nos dois pontos de reversão.
4. Parada no movimento alternado no ponto de reversão.
5. Reiniciar o movimento alternado.
6. Não iniciar antecipadamente a penetração parcial.

Para todas ações sincronizadas aqui apresentadas como exemplo estão valendo as suposições:

- Ponto de reversão 1 < ponto de reversão 2
- Z = eixo oscilante
- X = eixo de penetração

Indicação

Para mais explanações, veja o capítulo Ações sincronizadas de movimentos.

Atribuir eixo oscilante e eixo de penetração assim como definir a penetração total e parcial

Penetração na área de reversão

O movimento de penetração deve iniciar dentro de uma área de reversão, antes do ponto de reversão ser alcançado.

As ações sincronizadas evitam o movimento de penetração até que o eixo oscilante entre na área de reversão.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções:

Área de reversão 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X]=0
```

Sempre que a atual posição do eixo oscilante em MCS for maior que o início da área de reversão 1, então defina o Override axial do eixo de penetração em 0%.

Área de reversão 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X]=0
```

Sempre que a atual posição do eixo oscilante em MCS for menor que o início da área de reversão 2, então defina o Override axial do eixo de penetração em 0%.

Penetração no ponto de reversão

Não é executado nenhum movimento do eixo de penetração enquanto o eixo oscilante não alcançar o ponto de reversão.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções:

Área de reversão 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Sempre que a atual posição do eixo oscilante Z em MCS for maior ou menor que a posição do ponto de reversão 1, então defina o Override axial do eixo de penetração X em 0% e o Override axial do eixo oscilante Z em 100%.

Área de reversão 2:

Para ponto de reversão 2:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Sempre que a atual posição do eixo oscilante Zu em MCS for maior ou menor que a posição do ponto de reversão 2, então defina o Override axial do eixo de penetração X em 0% e o Override axial do eixo oscilante Z em 100%.

Parada do movimento alternado no ponto de reversão

O eixo oscilante é parado no ponto de reversão, ao mesmo tempo é iniciado o movimento de penetração. O movimento alternado é continuado assim que o movimento de penetração for concluído.

Simultaneamente esta ação sincronizada pode ser utilizada para iniciar o movimento de penetração, caso este tenha sido parado por uma ação sincronizada anterior que ainda está ativa.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções:

Área de reversão 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Sempre que a atual posição do eixo oscilante em MCS for igual à posição de reversão 1, então defina o Override axial do eixo oscilante em 0% e o Override axial do eixo de penetração em 100%.

Área de reversão 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

Sempre que a atual posição do eixo oscilante Zu em MCS for igual à posição de reversão 2, então defina o Override axial do eixo oscilante X em 0% e o Override axial do eixo de penetração em 100%.

Avaliação Online do ponto de reversão

Se ao lado direito da comparação estiver uma variável de processamento principal marcada com `$$`, então as duas variáveis serão avaliadas durante o ciclo de interpolação e comparadas entre si.

Indicação

Para mais informações sobre este assunto, veja o capítulo "Ações sincronizadas de movimentos".

Reiniciar o movimento alternado

Esta ação sincronizada é utilizada para dar continuidade ao movimento do eixo oscilante quando o movimento de penetração parcial estiver concluído.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções:

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z] = 100
```

Sempre que o curso restante para a penetração parcial do eixo de penetração X em WCS for igual a zero, então defina o override axial do eixo oscilante em 100%.

Próxima penetração parcial

Após a realização da penetração se deve evitar uma partida antecipada da próxima penetração parcial.

Para isso é utilizado um marcador específico de canal (`$AC_MARKER[índice]`) que é especificado no fim da penetração parcial (curso restante parcial = 0) e deletado depois de deixar a área de reversão. Então o próximo movimento de penetração é evitado com uma ação sincronizada.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções, por exemplo, para o ponto de reversão 1:

1. Definir marcador:

```
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[1] = 1
```

Sempre que o curso restante da penetração parcial do eixo de penetração X em WS for igual a zero, então defina o marcador de índice 1 em 1.

2. Deletar marcador

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1]=0
```

Sempre que a atual posição do eixo oscilante Z em MCS for maior ou menor que a posição do ponto de reversão 1, então defina o marcador 1 em 0.

3. Evitar a penetração

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X] = 0
```

Sempre que o marcador 1 for igual, então defina o Override axial do eixo de penetração X em 0%.

Estampagem e puncionamento 16

16.1 Ativação/desativação

16.1.1 Ativar/desativar estampagem e puncionamento (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Ativação/desativação da estampagem e puncionamento

A função de estampagem e de puncionamento é ativada com PON e SON. O SPOF encerra todas funções específicas de estampagem e de puncionamento. Os comandos PON e SON ativados modalmente se encerram mutuamente, isto é, o PON desativa o SON e vice-versa.

Estampagem/puncionamento com guia

As funções SONS e PONS também ativam as funções de estampagem e de puncionamento.

Ao contrário do controle de curso no nível de interpolação ativo com SON/PON, para estas funções é realizado um controle baseado em sinais da inicialização do curso no nível servo. Com isso pode-se trabalhar com frequências mais elevadas de curso, consequentemente com uma maior capacidade de estampagem.

Durante a avaliação de sinais na guia, todas funções que conduzem a uma alteração de posição dos eixos de puncionamento ou de estampagem (p. ex. deslocamento manual por manivela, alterações de frames via PLC, funções de medição) são bloqueadas.

Estampagem com retardo

O PDELAYON realiza a execução com retardo do curso de estampagem. O comando ativado modalmente tem função preparatória e normalmente está antes do PON. A estampagem é normalmente continuada após o PDELAYOF.

Indicação

O tempo de retardo é ajustado no dado de ajuste SD42400 $SC_PUNCH_DWELLTIME.

Aceleração em função do curso

Com PUNCHACC pode ser definida uma curva característica de aceleração, que define diferentes acelerações em função da distância entre os furos.

Segunda interface de estampagem

As máquinas que devem utilizar alternadamente uma segunda interface de estágio (segunda unidade de estampagem ou uma mídia equivalente), podem ser comutadas para um segundo par de entradas e saídas digitais rápidas do comando (par de I/O). A total funcionalidade de estampagem/puncionamento está disponível para as duas interfaces de estágio. A comutação

entre a primeira e a segunda interface de estampagem é realizada através dos comandos `SPIF1` e `SPIF2`.

Indicação

Pré-requisito: Um segundo par de I/O precisa estar definido para a funcionalidade de estampagem através de dados de máquina (→ veja as informações do fabricante da máquina!).

Sintaxe

```
PON G... X... Y... Z...
SON G... X... Y... Z...
SONS G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYON
PDELAYOF
PUNCHACC(<Smin>,<Amin>,<Smax>,<Amax>)
SPIF1/SPIF2
SPOF
```

Significado

`PON`:	Ativação da estampagem	
`SON`:	Ativação do puncionamento	
`PONS`:	Ativação da estampagem com guia	
`SONS`:	Ativação do puncionamento com guia	
`SPOF`:	Desativação da estampagem/puncionamento	
`PDELAYON`:	Ativação da estampagem com retardo	
`PDELAYOF`:	Desativação da estampagem com retardo	
`PUNCHACC`:	Ativação da aceleração em função do curso Parâmetro:	
	`<Smin>`	Menor distância entre furos
	`<Amin>`	Aceleração inicial A aceleração inicial `<Amin>` pode ser maior que o **<**`Amax>`.
	`<Smax>`	Maior distância entre furos
	`<Amax>`	Aceleração final `<Amax>` pode ser maior que o **<**`Amin>`.
`SPIF1`:	Ativação da **primeira** interface de estampagem O controle de curso é realizado através do primeiro par de I/O rápido.	
`SPIF2`:	Ativação da **segunda** interface de estampagem O controle de curso é realizado através do segundo par de I/O rápido.	
	Nota: Após o RESET ou na inicialização do comando é sempre a primeira interface de estampagem que está ativa. Se apenas uma interface de estampagem for usada, então esta programação não precisa ser feita.	

Exemplos

Exemplo 1: Ativação do puncionamento

Código de programa	Comentário
...	
N70 X50 SPOF	; Posicionamento sem disparo de estampagem.
N80 X100 SON	; Ativação de puncionamento, disparo de um curso antes do movimento (X=50) e no fim do movimento programado (X=100).
...	

Exemplo 2: Estampagem com retardo

Código de programa	Comentário
...	
N170 PDELAYON X100 SPOF	; Posicionamento sem disparo de estampagem, ativação do disparo de estampagem com retardo.
N180 X800 PON	; Ativação da estampagem. Depois de ser alcançada a posição final, o curso de estampagem é executado com retardo.
N190 PDELAYOF X700	; Desativação da estampagem com retardo, disparo de estampagem normal no fim do movimento programado.
...	

Exemplo 3: Estampagem com duas interfaces de estágio

Código de programa	Comentário
...	
N170 SPIF1 X100 PON	; No fim do bloco ocorre um disparo de curso na primeira saída rápida. O sinal "Curso ativo" é monitorado na primeira entrada.
N180 X800 SPIF2	; O segundo disparo de curso ocorre na segunda saída rápida. O sinal "Curso ativo" é monitorado na segunda entrada.
N190 SPIF1 X700	; O controle de curso para os demais cursos é realizado com a primeira interface.
...	

Outras informações

Estampagem e puncionamento com guia (PONS/SONS)

A estampagem e puncionamento com guias simultâneos não são possíveis em vários canais. Somente o `PONS` ou o `SONS` pode estar ativo em um canal.

Aceleração em função do curso (PUNCHACC)

Exemplo:
`PUNCHACC(2,50,10,100)`

Distâncias entre fusos abaixo de 2mm:

O deslocamento é executado com uma aceleração de 50% da aceleração máxima.

As distâncias entre furos de 2mm até 10mm:

A aceleração é aumentada proporcionalmente à distância em 100%.

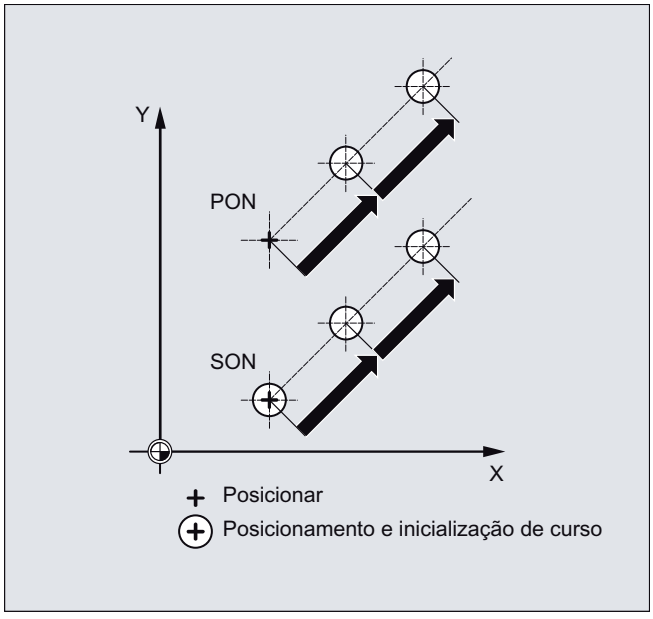
Distâncias entre furos maior que 10mm:

Deslocamento com uma aceleração de 100%.

Disparo do primeiro curso

O disparo do primeiro curso após a ativação da função é realizado em tempos diferentes para o puncionamento e estampagem:

- `PON/PONS`:
  - Todos cursos são realizados no fim do bloco, mesmo o primeiro bloco após a ativação.
- `SON/SONS`:
  - O primeiro curso após a ativação do puncionamento é realizado no início do bloco.
  - Todos demais cursos são ativados no fim do bloco.



Estampagem e puncionamento no local

Uma inicialização do curso somente ocorre quando o primeiro bloco contém uma informação de deslocamento para os eixos de estampagem ou puncionamento (eixos do plano ativo).

Mesmo assim, para disparar um curso no mesmo local, deve ser programado um dos eixos de estampagem/puncionamento com percurso 0.

Trabalhar com ferramentas rotativas

Indicação

Para empregar ferramentas rotativas tangencialmente ao percurso programado, utilize o controle tangencial.

Utilização de comandos M

Com a ajuda da tecnologia de macros ainda é possível utilizar funções M especiais ao invés dos comandos de linguagem (compatibilidade). Neste caso são aplicadas as seguintes correspondências com os sistemas antigos:

```
M20, M23   ≙  SPOF
M22        ≙  SON
M25        ≙  PON
M26        ≙  PDELAYON
```

Exemplo para arquivo de macro:

Código de programa	**Comentário**
`DEFINE M25 AS PON`	`; Estampagem ativada`
`DEFINE M125 AS PONS`	`; Estampagem com guia ativada`
`DEFINE M22 AS SON`	`; Puncionamento ativado`
`DEFINE M122 AS SONS`	`; Puncionamento com guia ativado`
`DEFINE M26 AS PDELAYON`	`; Estampagem com retardo ativada`
`DEFINE M20 AS SPOF`	`; Estampagem, puncionamento desativado`
`DEFINE M23 AS SPOF`	`; Estampagem, puncionamento desativado`

Exemplo de programação:

Código de programa	**Comentário**
`...`	
`N100 X100 M20`	`; Posicionamento sem disparo de estampagem.`
`N110 X120 M22`	`; Ativação de puncionamento, disparo de curso antes e depois do movimento.`
`N120 X150 Y150 M25`	`; Ativação de estampagem, disparo de curso no fim do movimento.`
`...`	

16.2 Segmentação automática de curso

Segmentação em trechos

Para a estampagem e puncionamento ativados, tanto o SPP como o SPN realizam uma segmentação do trecho total de deslocamento para os eixos de percurso programados obtendo um número de trechos de mesmo comprimento (segmentação de curso equidistante). Internamente cada trecho corresponde à um bloco.

Número de cursos

Para a estampagem o primeiro curso é realizado no ponto final do primeiro trecho, já para o puncionamento é realizado no ponto de partida do primeiro trecho. Através do trecho total temos como resultado os seguintes números:

Estampagem: Número de cursos = número de trechos

Puncionamento: Número de cursos = número de trechos + 1

Funções auxiliares

As funções auxiliares são executadas no primeiro bloco gerado.

Sintaxe

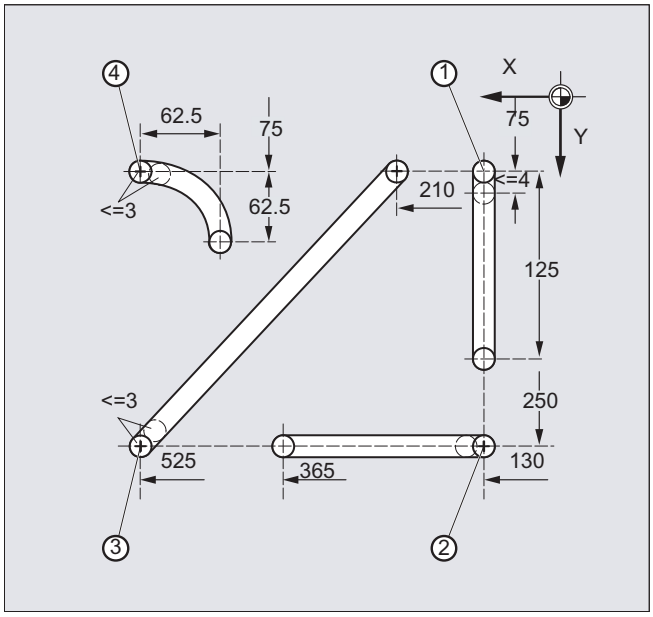
```
SPP=
SPN=
```

Significado

SPP:	Tamanho do trecho (distância máxima de curso); ativado modalmente
SPN:	Número de trechos por bloco; ativado por blocos

Exemplo 1

Os trechos de puncionamento programados devem ser automaticamente segmentados em trechos de mesmo tamanho.

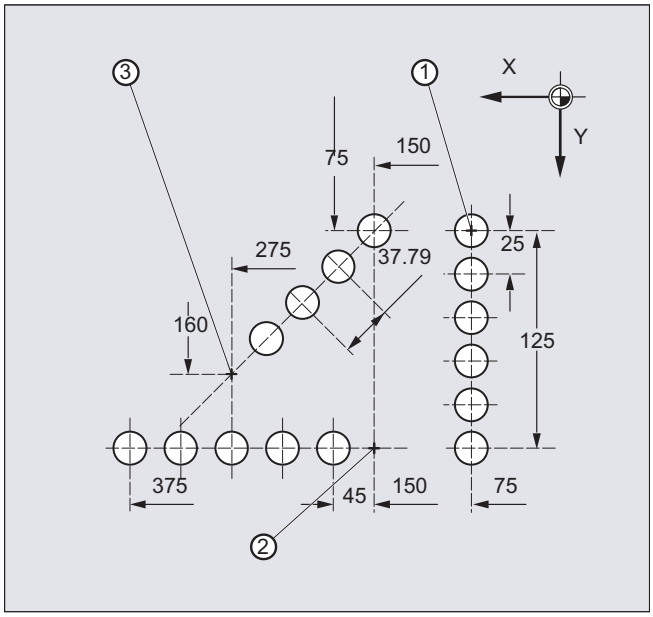


Código de programa	**Comentário**
N100 G90 X130 Y75 F60 SPOF	; Posicionar no ponto de partida 1
N110 G91 Y125 SPP=4 SON	; Puncionamento ativado; comprimento máximo do trecho para a segmentação automática do curso: 4 mm
N120 G90 Y250 SPOF	; Puncionamento desativado; posicionar no ponto de partida 2
N130 X365 SON	; Puncionamento ativado; comprimento máximo do trecho para a segmentação automática do curso: 4 mm
N140 X525 SPOF	; Puncionamento desativado; posicionar no ponto de partida 3
N150 X210 Y75 SPP=3 SON	; Puncionamento ativado; comprimento máximo do trecho para a segmentação automática do curso: 3 mm
N160 X525 SPOF	; Puncionamento desativado; posicionar no ponto de partida 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; Puncionamento ativado; comprimento máximo do trecho para a segmentação automática do curso: 3 mm

Código de programa	Comentário
N180 G00 G90 Y300 SPOF	; Puncionamento desativado

Exemplo 2

Para as fileiras individuais de furos deve ser realizada uma segmentação automática do curso. Para a segmentação é especificado o comprimento máximo do trecho (valor SPP).



Código de programa	Comentário
N100 G90 X75 Y75 F60 PON	; Posicionar no ponto de partida 1; estampagem ativada, puncionar furo individual
N110 G91 Y125 SPP=25	; Comprimento máximo do trecho para segmentação automática do curso: 25 mm
N120 G90 X150 SPOF	; Estampagem desativada; posicionar no ponto de partida 2
N130 X375 SPP=45 PON	; Estampagem ativada, comprimento máximo do trecho para a segmentação automática do curso: 45 mm
N140 X275 Y160 SPOF	; Estampagem desativada; posicionar no ponto de partida 3
N150 X150 Y75 SPP=40 PON	; Estampagem ativada; no lugar do comprimento do trecho programado de 40 mm é utilizado o comprimento calculado de trecho de 37,79 mm.
N160 G00 Y300 SPOF	; Estampagem desligada; posicionamento

16.2.1 Segmentação de curso para eixos de percurso

Comprimento do trecho SPP

Com o `SPP` especificamos a distância máxima do curso e com isso o comprimento máximo do trecho com que o trecho total de deslocamento deverá ser segmentado. A desativação do comando é feita com `SPOF` ou `SPP=0`.

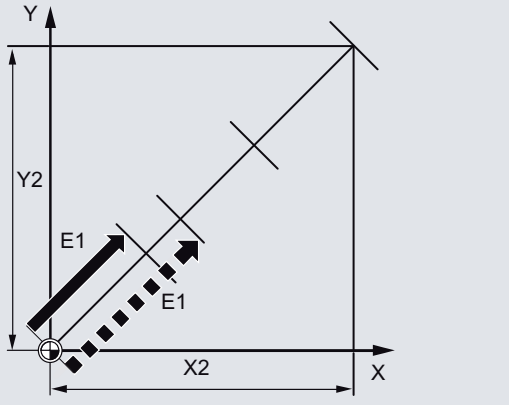
Exemplo:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

O trecho total de deslocamento de 10 mm é segmentado em 5 trechos de 2 mm cada (`SPP=2`).

Indicação

A segmentação de curso com `SPP` sempre é realizada de forma eqüidistante: todos trechos têm o mesmo comprimento. Isto significa que o tamanho programado do trecho (valor do `SPP`) somente será aplicado se o quociente do trecho total de deslocamento e o valor SPP forem números inteiros. Se não for este o caso, então o tamanho do trecho será reduzido internamente de modo que obtenha um quociente de número inteiro.



X2/Y2	Percurso programado (bloco de puncionamento ou estampagem)
E1	Comprimento programado de trecho
E1'	Comprimento de trecho arredondado automaticamente

Exemplo:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

Com um trecho total de deslocamento de 15 mm e um trecho com comprimento de 3,5 mm não é obtido um quociente de número inteiro (4.28). Assim é realizada uma redução do valor

SPP até o quociente de número inteiro mais próximo. Neste caso temos como resultado um trecho com comprimento de 3mm.

Número de trechos SPN

Com `SPN` definimos o número de trechos que devem ser gerados a partir do trecho total de deslocamento. O comprimento dos trechos é calculado automaticamente. Visto que o `SPN` é ativado por blocos, antes se deve ativar a estampagem ou o puncionamento com `PON` ou `SON`.

16.2.2 Segmentação de curso para eixos individuais

Se além dos eixos de percurso também forem definidos eixos individuais como eixo de estampagem/puncionamento, então estes também podem estar submetidos à segmentação automática de curso.

Comportamento do eixo individual com SPP

O comprimento programado do trecho (`SPP`) sempre é relativo aos eixos de percurso. Por isso que o valor SPP é ignorado em um bloco onde não existe nenhum eixo de percurso programado além do movimento do eixo individual e do valor SPP.

Se forem programados tanto o eixo individual como o eixo de percurso, então o comportamento do eixo individual orienta-se conforme o ajuste do respectivo dado da máquina.

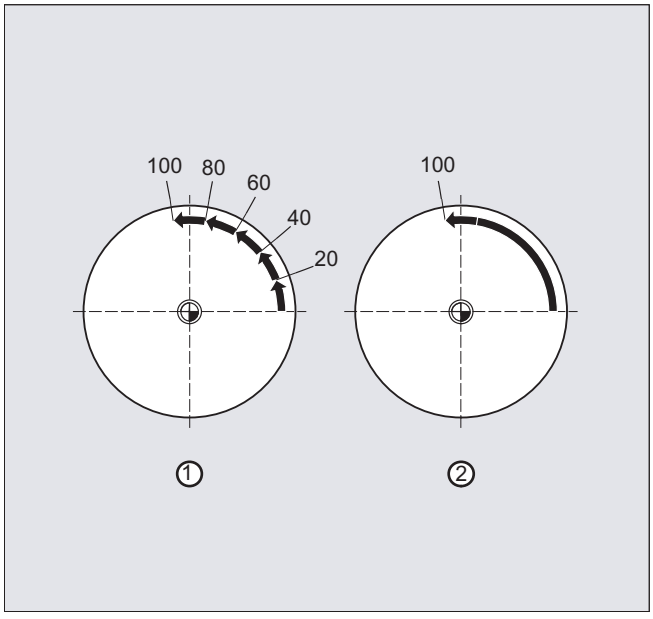
1. Ajuste padrão
   O curso do eixo individual é segmentado uniformemente pelos blocos intermediários gerados pelo `SPP`.

Exemplo:

```
N10 G1 SON X10 A0
N20 SPP=3 X25 A100
```

Com um trecho de 3 mm são gerados 5 blocos a partir de um trecho total de deslocamento do eixo X (eixo de percurso) de 15 mm.

O eixo A faz um giro de 20° em cada bloco.



1. Eixo individual sem segmentação de curso
   O eixo individual movimenta-se pelo seu curso total no primeiro dos blocos gerados.
2. Segmentação de curso diferenciada
   O comportamento do eixo individual depende da interpolação dos eixos de percurso:

- Interpolação circular: Segmentação de curso
- Interpolação linear: sem segmentação de curso

Comportamento para SPN

O número programado de trechos também será aplicado se ao mesmo tempo não for programado um eixo de percurso.

Pré-requisito: Eixo individual definido como eixo de estampagem/puncionamento.

Retificação 17

17.1 Ligar/desligar monitoramento da ferramenta específica de desbaste (TMON, TMOF)

Com os procedimentos pré-definidos TMON(...) e TMOF(...), o monitoramento da ferramenta específica de desbaste (monitoramento geométrico e de velocidade de giro) será ligado ou desligado.

Pré-requisito

Os parâmetros específicos de ferramenta $TC_TPG1 bis $TC_TPG9 devem ser definidos.

Sintaxe

```
TMON(<TNr>)
...
TMOF(<TNr>)
```

Significado

`TMON(...)`:	Ligar o monitoramento da ferramenta específica de desbaste O comando deve ser programado no canal em que o monitoramento da ferramenta específica de desbaste deverá ser ligado.
`TMOF(...)`:	Desligar o monitoramento da ferramenta específica de desbaste O comando deve ser programado no canal em que o monitoramento da ferramenta específica de desbaste deverá ser desligado.
`<TNr>`:	Número T **Nota:** necessário apenas, se o monitoramento do rebolo tiver que ser ligada ou desligada para um rebolo não ativo, ao invés da ferramenta ativa, em uso.
`TMOF(0)`:	Desativar a monitoração para todas ferramentas

Outras funções 18

18.1 Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

Por exemplo, o "AXNAME" é utilizado na criação de ciclos de uso geral, se o nome dos eixos não forem conhecidos.

O "AX" é utilizado para a programação indireta de eixos geométricos e eixos sincronizados. Aqui o identificador de eixo é armazenado em uma variável do tipo AXIS ou fornecida por um comando como o "AXNAME" ou "SPI".

O "SPI" é utilizado, quando forem programadas as funções de eixo para um fuso, por ex. fuso sincronizado.

"AXTOSPI" é utilizado, para converter um identificador de eixo em um índice de fuso (função inversa ao "SPI").

"AXSTRING" é utilizado, para converter um identificador de eixo (tipo de dado AXIS) em um String (função inversa ao "AXNAME").

O "ISAXIS" é utilizado em todos ciclos válidos, para assegurar que um determinado eixo geométrico esteja presente e com isso evitar com que uma chamada de $P_AXNX não seja cancelada com falha.

O "MODAXVAL" é utilizado para determinar a posição de módulo no caso dos eixos rotativos de módulo.

Sintaxe

```
AXNAME("String")
AX[AXNAME("String")]
SPI(n)

AXTOSPI(A) ou AXTOSPI(B) ou AXTOSPI(C)
AXSTRING(SPI(n))
ISAXIS(<número do eixo geométrico>)
<posição Modulo>=MODAXVAL(<eixo>,<posição de eixo>)
```

Significado

AXNAME:	Converte uma string de entrada no identificador de eixo; a string de entrada deve conter o nome de eixo válido.
AX:	Identificador variável de eixo
SPI:	Converte o número de fuso em identificador de eixo; o parâmetro de transferência deve conter um número de fuso válido.
n:	Número do fuso
AXTOSPI:	Converte um identificador de eixo em um índice de fuso de tipo Integer. "AXTOSPI" corresponde a função inversa ao "SPI".
X, Y, Z:	Identificador de eixo de tipo AXIS como variável ou constante

AXSTRING:	Se retorna a string com o número de fuso atribuído.
ISAXIS:	Verifica se o eixo geométrico especificado está presente.
MODAXVAL:	Determina em eixos rotativos Modulo a posição Modulo; esta corresponde ao resto Modulo relacionado à área Modulo parametrizada (no ajuste padrão compreende a faixa de 0 a 360 graus; através do MD30340 MODULO_RANGE_START e MD30330 $MA_MODULO_RANGE se pode alterar o início e o tamanho da área Modulo).

Indicação

Ampliações SPI

A função de eixo SPI(n) também pode ser aplicada para a leitura e gravação de componentes de Frame. Com isso podem ser gravados Frames com a sintaxe `$P_PFRAME[SPI(1),TR]=2.22`.

Com a programação adicional de posições de eixo através do endereço `AX[SPI(1)]=<posição de eixo>` nenhum eixo pode ser movimentado. O requisito para isso é que o fuso esteja em modo de posicionamento ou em modo de eixo.

Exemplos

Exemplo 1: AXNAME, AX, ISAXIS

Código de programa	Comentário
`OVRA[AXNAME("eixo transversal")]=10`	`; Override para eixo transversal`
`AX[AXNAME("eixo transversal")]=50.2`	`; Posição do final para eixo transversal`
`OVRA[SPI(1)]=70`	`; Override para fuso 1`
`AX[SPI(1)]=180`	`; Posição final para fuso 1`
`IF ISAXIS(1)==FALSE GOTOF CONTINUA`	`; Abscissa presente?`
`AX[$P_AXN1]=100`	`; Percorrer abscissa`
`CONTINUA:`	

Exemplo 2: AXSTRING

Na programação com AXSTRING[SPI(n)] o índice de eixo do eixo atribuído ao fuso não é mais retornado como número de fuso, mas retornada a string "Sn".

Código de programa	Comentário
`AXSTRING[SPI(2)]`	`; Será emitido o String "S2".`

Exemplo 3: MODAXVAL

Deve ser determinada a posição Modulo do eixo rotativo Modulo A.

O valor de saída para o cálculo é a posição de eixo 372.55.

A faixa Modulo parametrizada é de 0 a 360 graus:

MD30340 MODULO_RANGE_START = 0

MD30330 $MA_MODULO_RANGE = 360

Código de programa	Comentário
R10=MODAXVAL(A,372.55)	; Posição de módulo calculada R10 = 12.55.

Exemplo 4: MODAXVAL

Se o identificador de eixo programado não se referir à um eixo rotativo Modulo, então o valor a ser convertido (<posição de eixo>) será retornado como sem ser alterado.

Código de programa	Comentário
R11=MODAXVAL(X,372.55)	; X é o eixo linear; R11 = 372.55.

18.2 Eixos geométricos comutáveis (GEOAX)

Com a função "Eixos geométricos comutáveis", os eixos geométricos parametrizados por meio dos dados da máquina podem ser substituídos por outros eixos de canal.

Sintaxe

```
GEOAX(<n>,<eixo de canal>,<n>,<eixo de canal>,<n>,<eixo de canal>)
GEOAX()
```

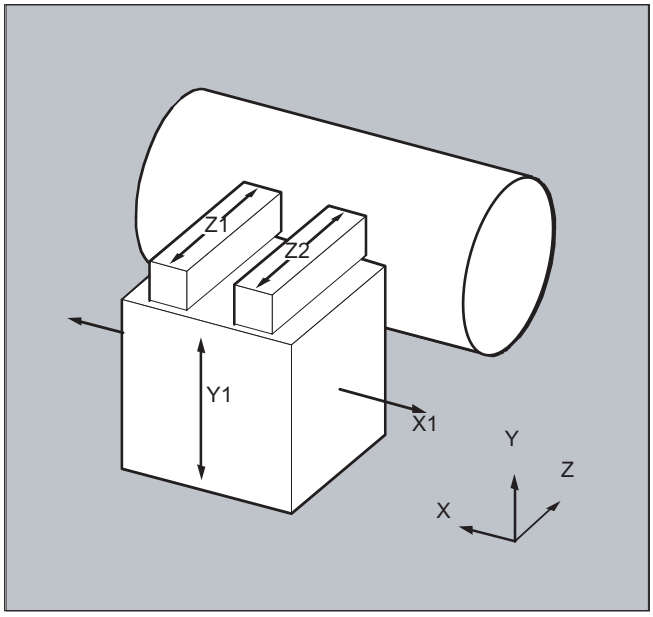
Significado

`GEOAX(...)`	Função para comutar os eixos geométricos. **Nota:** `GEOAX()` sem indicação de parâmetro ativa novamente a configuração básica dos eixos geométricos parametrizada nos dados da máquina.
`<n>`	Número do eixo geométrico, que será substituído pelo eixo de canal indicado. Faixa de valores: 0, 1, 2, 3 **Nota:** 0: O eixo de canal indicado será removido sem substituição do conjunto de eixos geométricos 1: 1. Eixo geométrico ≙ eixo de coordenadas X (abscissa) da WKS 2: 2. Eixo geométrico ≙ eixo de coordenadas Y (ordenada) da WKS 3: 3. Eixo geométrico ≙ eixo de coordenadas Z (coordenada) da WKS
`<eixo de canal>`	Nome do eixo de canal, que deve ser incorporado ao conjunto de eixos geométricos.

Exemplos

Exemplo 1: Ativar dois eixos alternadamente como eixo geométrico

Um carro de ferramentas pode ser movido através dos eixos de canal X1, Y1, Z1, Z2:



Os eixos geométricos são projetados de forma que, depois de ligar o sistema, seja ativado primeiro o Z1 como 3º eixo geométrico que recebe o nome de eixo geométrico "Z", formando um grupo de eixos geométricos com o X1 e o Y1.

Agora no programa de peça os eixos Z1 e Z2 devem ser aplicados alternadamente como eixo geométrico Z:

Código de programa	**Comentário**
...	
N100 GEOAX(3,Z2)	; Como 3° eixo geométrico (Z) atua o eixo de canal Z2.
N110 G1 ...	
N120 GEOAX(3,Z1)	; Como 3° eixo geométrico (Z) atua o eixo de canal Z1.
...	

Exemplo 2: Comutação dos eixos geométricos com 6 eixos de canal

Uma máquina possui 6 eixos de canal com os nomes XX, YY, ZZ, U, V e W.

O ajuste básico da configuração de eixos geométricos através de dados de máquina é:

Eixo de canal XX = 1º eixo geométrico (eixo X)

Eixo de canal YY = 2º eixo geométrico (eixo Y)

Eixo de canal ZZ = 3º eixo geométrico (eixo Z)

Código de programa	**Comentário**
N10 GEOAX()	; A configuração básica dos eixos geométricos está ativa.
N20 G0 X0 Y0 Z0 U0 V0 W0	; Todos eixos em avanço rápido até a posição 0.
N30 GEOAX(1,U,2,V,3,W)	; O eixo de canal U passa para primeiro (X), V para o segundo (Y)
	; e W para o terceiro eixo geométrico (Z).

Código de programa	Comentário
N40 GEOAX(1,XX,3,ZZ)	; O eixo de canal XX passa para primeiro (X), ZZ para terceiro
	; eixo geométrico (Z). O eixo de canal V permanece o segundo
	; eixo geométrico (Y).
N50 G17 G2 X20 I10 F1000	; Círculo inteiro no plano X/Y. Deslocam-se os
	; eixos de canal XX e V.
N60 GEOAX(2,W)	; O eixo de canal W passa a ser o segundo eixo geométrico (Y).
N80 G17 G2 X20 I10 F1000	; Círculo inteiro no plano X/Y. Deslocam-se os
	; eixos de canal XX e W.
N90 GEOAX()	; Reset para a posição normal.
N100 GEOAX(1,U,2,V,3,W)	; O eixo de canal U passa a ser o primeiro (X), V o segundo
	; (Y) e W o terceiro eixo geométrico (Z).
N110 G1 X10 Y10 Z10 XX=25	; Os eixos de canal U, V, W deslocam-se cada qual para a
	; posição 10. XX como eixo suplementar desloca-se para a posição 25.
N120 GEOAX(0,V)	; V é retirado do grupo de eixos geométricos.
	; U e W continuam a ser o primeiro (X) e o terceiro
	; eixo geométrico (Z).
	; O segundo eixo geométrico (Y) permanece desocupado.
N130 GEOAX(1,U,2,V,3,W)	; O eixo de canal U permanece o primeiro (X), V passa a ser o
	; segundo (Y), W permanece o terceiro eixo geométrico (Z).
N140 GEOAX(3,V)	; V passa a ser o terceiro eixo geométrico (Z), sendo que W
	; é sobrescrito e assim retirado do grupo de
	; eixos geométricos. O segundo eixo geométrico (Y)
	; continua a estar desocupado.

Dados de máquina

Configuração de eixo

Atribuição dos eixos geométricos, suplementares e da máquina aos eixos de canal:

- MD10000 $MN_AXCONF_MACHAX_NAME_TAB
- MD20050 $MC_AXCONF_GEOAX_ASIGN_TAB
- MD20060 $MC_AXCONF_GEOAX_NAME_TAB
- MD20070 $MC_AXCONF_MACHAX_USED
- MD20080 $MC_AXCONF_CHANAX_NAME_TAB
- MD35000 $MA_SPIND_ASSIGN_TO_MACHAX

Comportamento de Reset

Comportamento de Reset das atribuições modificadas dos eixos geométricos:

- MD20110 $MC_RESET_MODE_MASK, Bit 12
- MD20118 $MC_GEOAX_CHANGE_RESET

Comportamento de início NC

- MD20112 $MC_START_MODE_MASK, Bit 12

Informação ao programa de usuário de PLC

Possibilidade de parametrização do comando M, que no caso de uma comutação de eixo geométrico, será emitido ao ponto de junção NC/PLC:

- MD22532 $MC_GEOAX_CHANGE_M_CODE

Condições gerais

Sem comutação de eixo geométrico

- Se uma das seguintes funções estiver ativa, não é possível nenhuma comutação do eixo geométrico:
  - Transformação
  - Interpolação de Spline
  - Correção do raio da ferramenta
  - Correção fina de ferramenta
- O eixo geométrico e um outro eixo de canal possuem o mesmo nome.
- Um dos eixos envolvidos na comutação do eixo geométrico está envolvido em uma ação, que demora além dos limites do bloco. Por ex., eixo de posicionamento extensivo ao bloco ou eixo seguinte de um acoplamento de eixos.

Eixos rotativos

Os eixos rotativos não podem se tornar eixos geométricos.

Estado do eixo após a substituição

Um eixo substituído no grupo de eixos geométricos através da comutação pode ser programado como eixo adicional através de seu nome de eixo de canal e após o processo de comutação.

Frames, áreas de proteção, limites da área de trabalho

Com a comutação dos eixos geométricos são deletados todos os Frames, áreas de proteção e limites da área de trabalho.

Coordenadas polares

Uma troca de eixos geométricos com `GEOAX` ajusta as coordenadas polares modais para o valor 0 de forma similar à numa mudança de planos com `G17-G19`.

DRF, NPV

Um eventual deslocamento com manivela eletrônica (DRF) ou um deslocamento externo do ponto zero permanece ativo após a comutação.

Configuração básica dos eixos geométricos

O comando `GEOAX()` chama a configuração básica do grupo de eixos geométricos.

Após o POWER ON e na mudança para o modo de operação "Aproximação do ponto de referência" se restabelece automaticamente a configuração básica.

Correção do comprimento da ferramenta

Uma correção ativa do comprimento da ferramenta também está ativa após o processo de comutação. Porém, ela vale como ainda não executada para os eixos geométricos que foram adotados ou trocados de posição. No primeiro comando de movimento para estes eixos geométricos o percurso resultante é composto da soma da correção do comprimento da ferramenta e o percurso programado.

Os eixos geométricos que mantém sua posição no grupo de eixos em uma comutação, também mantém seu estado relativo à correção do comprimento da ferramenta.

Configuração de eixos geométricos com transformação ativa

- A configuração de eixos geométricos parametrizada para uma transformação ativa através dos dados da máquina de transformação não pode ser alterada através da função "Eixos geométricos comutáveis".
- As diferentes configurações de eixos geométricos para uma transformação, devem ser parametrizadas nos dados da máquina de transformação por diferentes blocos de dados.
- Uma configuração de eixo geométrico alterada através de `GEOAX` é eliminada pela ativação de uma transformação.
- Com referência aos eixos geométricos, as parametrizações de eixos geométricos específicas das transformações ativas apresentam prioridade sobre as parametrizações referentes à comutação dos eixos geométricos.
  Exemplo: Uma transformação está ativa. Segundo os dados da máquina, a transformação deve ser mantida no caso de reset do canal. Simultaneamente, em caso de um reset de canal entretanto, a configuração básica dos eixos geométricos devem ser criada. Será mantida a configuração dos eixos geométricos, que tiver sido determinada para a transformação.
- Com a desativação de uma transformação será ativada novamente o ajuste básico parametrizado da configuração dos eixos geométricos.

Modo de operação JOG, função da máquina REF

Ao comutar para o modo de operação JOG, função da máquina REF (deslocar para o ponto de referência), a configuração dos eixos geométricos parametrizada nos dados da máquina é ativada..

18.3 Contentor de eixos (AXCTSWE, AXCTSWED, AXCTSWEC)

Com os comandos "AXCTSWE" ou "AXCTSWED" é liberada a rotação dos contentores de eixos indicados.

Com o comando "AXCTSWEC" é possível retomar novamente uma liberação feita para a rotação do contentor de eixo.

Sintaxe

```
AXCTSWE(<ID>)
AXCTSWED(<ID>)
AXCTSWEC(<ID>)
```

Significado

AXCTSWE:	Liberação para rotação do contentores de eixo A execução do programa não será parada pelo "AXCTSWE". A rotação acontece assim que todos os canais pertencentes ao contentor de eixo transmitirem a liberação.	
AXCTSWED:	Liberação da rotação do contentor de eixo sem tomada em consideração os outros canais pertencentes ao contentor de eixo **Nota** • Variante de comando para simplificação da operação do programa de peça e/ou ação síncrona. • O comportamento referente aos outros canais pertencentes ao contentor, pode ser definido por meio de: MD12760 $MN_ AXCT_FUNCTION_MASK, Bit 0	
AXCTSWEC:	Retomada da liberação para rotação de contentores de eixo **Observação:** A liberação de uma rotação de contentor de eixo somente poderá ser retomada se a rotação ainda não tiver sido iniciada: $AN_AXCTSWA[<contentor de eixo>] == 0 Para variáveis de sistema, consulte em "Contentor de eixos (AXCTSWE, AXCTSWED, AXCTSWEC) (Página 669)"	
<ID>:	Identificador de contentores de eixo ou de um eixo contentor:	
	CT<Numero>:	Identificador de erro de contentor de eixo: MD12750 $MN_AXCT_NAME_TAB Exemplo: "CT1"
	<Contentor>:	Contentor específico de usuário de um eixo contentor: MD12750 $MN_AXCT_NAME_TAB Exemplo: "CONTAINER_1"
	<eixo>:	Identificador de um eixo contentor reconhecido no canal

Indicação

Incremento

A distância da rotação do contentor de eixo é programado por meio do Setting:

SD41700 $SN_AXCT_SWWIDTH

Outras informações

Diagnóstico

O atual estado de um contentor de eixo pode ser interpretado por meio da seguinte variável de sistema:

Variável de sistema	Tipo	Descrição
$AC_AXCTSWA[<Nome>]	BOOL	Estado específico de canal do contentor de eixo
$AN_AXCTSWA[<contentor de eixo>]	BOOL	Estado específico de NCU do contentor de eixo
$AN_AXCTSWE[<contentor de eixo>]	INT	Estado específico de Slot da rotação de contentor de eixo A variável de sistema retorna o estado **por Bit** do Slots do contentor de eixo. Cada Bit corresponde a um Slot.
$AN_AXCTAS[<contentor de eixo>]	INT	Número de entradas (Slots) através das quais o contentor de eixo foi comutado atualmente.

Rotação de contentor de eixo com GET / GETD implícito

Por meio do seguinte dado da máquina é possível programar, para que o comando "AXCTSWE" que todos os contentores de eixos do canal sejam trazidos por meio de "GET / GETD" implícito para o canal. Uma troca de eixo só é possível depois uma rotação bem-sucedida do contentor.

MD10722 $MN_AXCHANGE_MASK, Bit 1 = 1

Indicação

A rotação do contentor de eixo com "GET / GETD" implícitos **não** é executado para um eixo no estado de "eixo principal" por ex., eixo PLC), visto que o eixo deveria abandonar o estado para a rotação do contentor de eixo.

18.4 Espera pela posição de eixo válida (WAITENC)

Com o comando de linguagem "WAITENC", é possível esperar no programa NC, até que os eixos configurados com MD34800 $MA_WAIT_ENC_VALID = 1 estejam nas posições de eixo síncronas ou restauradas.

Em estado de espera pode ocorrer uma interrupção, p. ex. através da partida de uma ASUP ou pela mudança de modos de operação para JOG. Com a continuação do programa o estado de espera é novamente cancelado.

Indicação

O estado de espera é indicado na interface de operação através do estado de parada "Esperar pelo sistema de medição".

Sintaxe

O "WAITENC" pode ser programado na seção de programa de qualquer programa NC.

A programação deve ser realizada em um bloco próprio

```
...
WAITENC
...
```

Exemplo

Por exemplo, o "WAITENC" é utilizado no programa de usuário controlado por eventos .../_N_CMA_DIR/_N_PROG_EVENT_SPF, como mostra o seguinte exemplo de aplicação.

Exemplo de aplicação: Retrocesso de ferramenta após POWER OFF com transformação de orientação

Uma usinagem com orientação de ferramenta foi interrompida através de queda de energia. Em seguida, ao inicializar o sistema é ativado o programa de usuário controlado por eventos .../_N_CMA_DIR/_N_PROG_EVENT_SPF.

No programa de usuário controlado por eventos existe uma espera pelas posições de eixo síncronas ou restauradas através do "WAITENC", para depois poder calcular um Frame, que alinha o WCS no sentido da ferramenta.

Código de programa	Comentário
...	
IF $P_PROG_EVENT == 4	; Inicialização.
IF $P_TRAFO <> 0	; A transformação foi selecionada.
WAITENC	; Esperar pelas posições de eixo válidas dos eixos de orientação.
TOROTZ	; Rotação do eixo Z do WCS no sentido do eixo de ferramenta.
ENDIF	
M17	

Código de programa	Comentário
ENDIF	
...	

Em seguida, em modo de operação JOG, a ferramenta pode ser afastada através de um movimento de retrocesso no sentido do eixo da ferramenta.

18.5 Bloco programável de alteração de parâmetros (SCPARA)

Com o comando "SCPARA" pode ser requisitado para um eixo a comutação para um determinado bloco de parâmetros.

Indicação

Não alterar o bloco de parâmetro durante o rosqueamento

Nos rosqueamentos G33 eG331 / G332 o bloco do parâmetro é escolhido pelo controle e não pode ser modificado.

bloqueada a alteração do bloco do parâmetro

Uma alteração do bloco do parâmetro também pode ser solicitada através da interface NC/PLC. Para evitar conflitos na alteração, é possível bloquear a alteração do bloco do parâmetro do NC (SCPARA) através da interface NC/PLC:

DB31, ... DBX9.3 (especificação do bloco de parâmetros bloqueada por NC)

Indicação

Se for requisitada uma comutação do bloco de parâmetros através do "SCPARA", enquanto a comutação do bloco de parâmetros estiver bloqueada pela interface NC/PLC, a comutação será rejeitada sem mensagem de erro.

Sintaxe

```
SCPARA[<eixo>]=<valor>
```

Significado

SCPARA:	Comando: alterar bloco do parâmetro	
<eixo>:	Identificador de eixo (eixo canal)	
	Tipo:	AXIS
<valor>:	número do bloco do parâmetro: 1, 2, 3, ... número máx. do bloco do parâmetro	

Exemplo

Código de programa	**Comentário**
...	
N110 SCPARA[X]= 3	; Seleção: Eixo X, 3° bloco do parâmetro
...	

Outras informações

alteração do bloco do parâmetro liberada

A alteração do bloco do parâmetro do eixo precisa ser liberada explicitamente:

MD35590 $MA_PARAMSET_CHANGE_ENABLE[<Eixo>]

Leitura do número do bloco do parâmetro

O número do bloco do parâmetro selecionado (Parâmetro nominal) pode ser lido através da variável de sistema $AA_SCPAR .

Literatura

Para informações detalhadas sobre os blocos do parâmetro, veja:

Manual de funcionamento das funções básicas; Capítulo "Velocidades,Sistema variante, Regulagem (G2)" > "Regulagem" > "blocos do parâmetro do regulador de posição"

18.6 Controlar o escopo disponível de linguagem NC (STRINGIS)

Com a função "STRINGIS(...)" é possível verificar, se o String indicado como elemento de linguagem de programação NC está disponível no atual escopo de linguagem.

Definição

```
INT STRINGIS(STRING <nome>)
```

Sintaxe

```
STRINGIS(<nome>)
```

Significado

STRINGIS:	Função com valor de retorno
<nome>:	Nome do elemento verificado da linguagem de programação NC
Valor de retorno:	O formato do valor de retorno é yxx (decimal).

Elementos da linguagem de programação NC

Os seguintes elementos da linguagem de programação NC podem ser verificados:

- Comandos G de todos os grupos de G existentes, p. ex. "G0", "INVCW", "POLY", "ROT", "KONT", "SOFT", "CUT2D", "CDON", "RMBBL", "SPATH"
- Endereços DIN ou NC, como por ex., "ADIS", "RNDM", "SPN", "SR", "MEAS"
- Funções, por ex., "TANG(...)" ou "GETMDACT"
- Procedimentos, por ex., "SBLOF".
- Palavras chave, por ex., "ACN", "DEFINE" ou "SETMS"
- Dados de sistema, p. ex. dados de máquina $M... , dados de ajuste $S... ou dados de opção $O...
- Variável de sistema $A... , $V... , $P...
- Parâmetros de cálculo R...
- Nomes de ciclo de ciclos ativados
- Variáveis GUD e LUD
- Nomes de macro
- Nomes de Label

Valor de retorno

Um valor de retorno somente é relevante nas primeiras 3 casas decimais. O formato do valor de retorno é yxx, com y = informação básica e xx = informação de detalhe.

Valor de retorno	Significado	
000	A String 'name' não é reconhecida pelo presente sistema [1]	
100	A String 'name' é um elemento da linguagem de programação NC, mas atualmente não é programável (opção/função inativa)	
2xx	A String 'name' é um elemento programável da linguagem de programação NC (opção/função ativa). A informação de detalhe xx contém outras informações sobre o tipo do elemento:	
	xx	**Significado**
	01	Endereço DIN ou endereço NC [2]
	02	Comando G (p. ex. G04, INVCW)
	03	Função com valor de retorno
	04	Função sem valor de retorno
	05	Palavra-chave (p. ex. DEFINE)
	06	Dados de máquina ($M...), dados de ajuste ($S...) ou dados de opção $O...)
	07	Parâmetros de sistema, p. ex. variável de sistema $...) ou parâmetro de cálculo R...)
	08	Ciclo (O ciclo deve estar carregado no NC e os programas de ciclo ativados [3])
	09	Variável GUD (A variável GUD deve estar definida nos arquivos de definição GUD e ativada nas variáveis GUD)
	10	Nome de macro (A macro estar definida nos arquivos de definição de macro; as macros devem estar ativadas) [4]
	11	Variável LUD do atual programa de peça
	12	Comando G ISO (O modo de linguagem ISO deve estar ativo)
400	A String 'name' é um endereço NC, que não foi reconhecido como xx == 01 ou xx == 10 e que não é G nem R [2]	
y00	Nenhuma atribuição específica possível	

1) Dependendo do comando, sob determinadas circunstâncias, apenas uma menor quantidade de comandos de linguagem NC da Siemens são reconhecidos, SINUMERIK 802D sl. Nestes comandos numéricos é retornado o valor 0 para as Strings que são, em princípio, comandos de linguagem NC da Siemens. Este comportamento pode ser alterado através do MD10711 $MN_NC_LANGUAGE_CONFIGURATION. Com MD10711 = 1 sempre é retornado o valor 100 para comandos de linguagem NC da Siemens.

2) Os endereços NC são as seguintes letras: A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Estes endereços NC também podem ser programados com uma extensão de endereço. A extensão de endereço pode ser especificada na verificação realizada com STRINGIS. Exemplo: 201 == STRINGIS("A1").
Letras: D, F, H, L, M, N, O, P, S, T são endereços NC ou funções auxiliares que são usadas por definição de usuário. Para elas sempre é retornado o valor 400. Exemplo: 400 == STRINGIS("D"). Estes endereços NC não podem ser especificados com extensão de endereço na verificação realizada com STRINGIS.
Exemplo: 000 == STRINGIS("M02"), mas 400 == STRINGIS("M").

3) Os nomes de parâmetros de ciclos não podem ser verificados com STRINGIS.
4) Endereço definido como macro, por exemplo o G, H, M e L são identificados como macro

Exemplos

Nos seguintes exemplos considera-se que o elemento de linguagem NC especificado como String seja programável no comando numérico, isto se não houver outra observação.

1. A String "T" é definida como função auxiliar:
   ```
   400 == STRINGIS("T")
   000 == STRINGIS ("T3")
   ```
2. A String "X" é definida como eixo:
   ```
   201 == STRINGIS("X")
   201 == STRINGIS("X1")
   ```
3. A String "A2" é definida como endereço NC com extensão:
   ```
   201 == STRINGIS("A")
   201 == STRINGIS("A2")
   ```
4. A string "INVCW" é definida como comando G nomeado:
   ```
   202 == STRINGIS("INVCW")
   ```
5. A String "$MC_GCODE_RESET_VALUES" é definida como dado de máquina:
   ```
   206 == STRINGIS("$MC_GCODE_RESET_VALUES")
   ```
6. A String "GETMDACT" é uma função de linguagem NC:
   ```
   203 == STRINGIS("GETMDACT ")
   ```
7. A String "DEFINE" é uma palavra-chave:
   ```
   205 == STRINGIS("DEFINE")
   ```
8. A String "$TC_DP3" é um parâmetro de sistema (componente do comprimento da ferramenta):
   ```
   207 == STRINGIS("$TC_DP3")
   ```
9. A String "$TC_TP4" é um parâmetro de sistema (tamanho da ferramenta):
   ```
   207 == STRINGIS("$TC_TP4")
   ```
10. A String "$TC_MPP4" é um parâmetro de sistema (estado de alojamento do magazine):
    - O gerenciamento do magazine de ferramentas está ativo: `207 == STRINGIS("$TC_MPP4") ;`
    - O gerenciamento do magazine de ferramentas não está ativo: `000 == STRINGIS("$TC_MPP4")`

    Veja também o parágrafo abaixo: Gerenciamento de magazine de ferramentas.
11. A String "MACHINERY_NAME" é definida como variável GUD:
    ```
    209 == STRINGIS("MACHINERY_NAME")
    ```
12. A String "LONGMACRO" é definida como macro:
    ```
    210 == STRINGIS("LONGMACRO")
    ```
13. A String "MYVAR" é definida como variável LUD:
    ```
    211 == STRINGIS("MYVAR")
    ```
14. A string "XYZ" não é nenhum comando, variável GUS, nome de macro ou de ciclo conhecido no NC:
    ```
    000 == STRINGIS("XYZ")
    ```

Gerenciamento de magazine de ferramentas

Se a função de gerenciamento de magazine de ferramentas não estiver ativa, o STRINGIS retorna para os parâmetros de sistema do gerenciamento do magazine de ferramentas, independentemente do dado de máquina:

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION

sempre o valor 000.

Modo ISO

Se a função "Modo ISO" estiver ativa:

- MD18800 $MN_MM_EXTERN_LANGUAGE (ativação de linguagens NC externas)
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM (para sistema de comando que deve ser adaptado)

o STRINGIS verifica a string indicada primeiro como comando G do SINUMERIK. Se a string não for nenhum comando G do SINUMERIK, então ela será verificada como comando G de linguagem ISO.

As comutações programadas (`G290` (modo SINUMERIK), `G291` (modo ISO)) não têm nenhum efeito sobre o STRINGIS.

Exemplo

Os dados de máquina relevantes para o STRINGIS(...) possuem os seguintes valores:

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION = 2 (Somente são reconhecidos os comandos de linguagem NC cujas opções estiverem definidas)
- MD19410 $ON_TRAFO_TYPE_MASK = 'H0' (Opção: Transformações)
- MD10700 $MN_PREPROCESSING_LEVEL='H43' (Pré-processamento de ciclos ativo)

O próximo programa de exemplo é processado sem mensagem de erro:

Código de programa	**Comentário**
`N1 R1=STRINGIS("TRACYL")`	`; R1 == 0, visto que o TRACYL devido à falta da opção de transformação, é tido como "não reconhecido" no reconhecimento`
`N2 IF STRINGIS("TRACYL") == 204`	
`N3  TRACYL(1,2,3)`	`; N3 será ignorado`
`N4 ELSE`	
`N5  G00`	`; e ao invés disso é executado o N5`
`N6 ENDIF`	
`N7 M30`	

18.7 Chamada interativa a partir do programa de peça (MMC)

Através do subprograma predefinido MMC(...) podem ser indicados, a partir de um programa NC, diálogos específicos de usuário na interface de usuário de, por ex., SINUMERIK Operate.

Projetos de diálogos podem ser executados para os seguintes tipos de diálogo:

- Run MyScreens
- Easy XML
- XML de usuário

Literatura

- Manual de programação Run MyScreens
- Manual de programação Easy XML

Sintaxe

```
MMC("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>","<QUIT>")
```

Significado

`MMC(...)`:	identificador de subprograma Os parâmetros são indicados com codificação por localização e separados por vírgulas dentro de dois strings, o string de comando e o string de confirmação.	
Parâmetros dentro do string de comando:		
`<ADDRESS>`:	Área de operação na qual são executados os diálogos de usuário projetados	
	Função	**Áreas de operação**
	Diálogo de usuário "Run MyScreens"	`CICLOS`
	Diálogo de usuário "Easy XML"	`CICLOS`
	XML de usuário	`XML`
	Popup "Run MyScreens"	`POPUPDLG`
	Popup "Easy XML"	`POPUPDLG`
`<COMMAND>`:	Comando a ser executado	
	Função	**Comandos**
	Diálogo de usuário "Run MyScreens"	`PICTURE_ON`, `PICTURE_OFF`
	Diálogo de usuário "Easy XML"	`PICTURE_ON`, `PICTURE_OFF`
	XML de usuário	`XML_ON`, `XML_OFF`
	Popup "Run MyScreens"	`PICTURE_ON`, `PICTURE_OFF`
	Popup "Easy XML"	`PICTURE_ON`, `PICTURE_OFF`

<FILE>:	Nome do arquivo, no qual está programado o diálogo a ser indicado	
	Função	**Arquivos**
	Diálogo de usuário "Run MyScreens"	<name>.com
	Diálogo de usuário "Easy XML"	<name>.xml
	XML de usuário	<name>.xml
	Popup "Run MyScreens"	<name>.com
	Popup "Easy XML"	<name>.xml
	Popup "Easy XML" com execução do projeto diretamente no programa NC (veja exemplo 2)	xmldial_emb.xml
<DIALOG>:	Nome do diálogo a ser indicado	
	Função	**Nome do diálogo**
	Todas as funções exceto popup "Easy XML" com execução do projeto diretamente no programa NC	No arquivo <FILE> nome projetado do diálogo
	Popup "Easy XML" com execução do projeto diretamente no programa NC (veja exemplo 3)	main
Parâmetros dentro do string de confirmação:		
<QUIT>:	Tipo de confirmação	
	N:	Nenhuma confirmação. A execução do programa é continuada após o envio do comando. Nenhum sinal de retorno ocorre quando não foi possível executar o comando com sucesso. **Indicação** O tipo de confirmação "N" deve ser utilizado, quando no programa NC é programado um tempo de indicação (período de permanência) (veja abaixo exemplo 2)
	"A":	Confirmação assíncrona A execução do programa é continuada após o envio do comando. O valor de devolução é arquivado em uma variável de confirmação específica do usuário, definida no contexto da execução do projeto do diálogo (variável GUD), e pode ser lido no programa NC.

Exemplo

Exemplo 1

Indicação de um diálogo e da reação ao comando do usuário em programa NC.

```
Código de programa                                  Comentário
; A variável de confirmação QUIT já foi configurada como variável global de usuário
(GUD)
; do tipo STRING na execução do projeto do diálogo:
; DEF NCK STRING[20] QUIT
QUIT = "XXX"                                       ; Inicializar a variável de confirmação
G4 F5
```

Código de programa	**Comentário**
MMC("CYCLES,PICTURE_ON,test.com,test1","A")	; Indicar diálogo
	; - Área de operação: CICLOS
	; - Status da imagem: PICTURE_ON (indicar)
	; - Arquivo de imagem do diálogo: test.com
	; - Imagem do diálogo: test1
INPUT:	; Esperando o comando do usuário
STOPRE	; Parada de pré-processamento
IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK	; Softkey "RUN"
IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK	; Softkey "CHK"
GOTOB INPUT	; => Espera
WORK:	; Softkey "RUN" pressionado
MSG("Continuar processamento -> inicialização NC ")	; Sinalizar mensagem
MMC("CYCLES,PICTURE_OFF","N")	; Fechar diálogo
M0	; Esperando inicialização NC
GOTOF END	; => Fim do programa
CHECK:	; Softkey "CHK" pressionado
MSG("Aproximar posição -> inicialização NC")	; Sinalizar mensagem
MMC("CYCLES,PICTURE_OFF","N")	; Fechar diálogo
M0	; Esperando inicialização NC
GOTOF END	; => Fim do programa
END:	
...	

Exemplo 2

O tempo de indicação de um diálogo é determinado no programa NC, por ex., por meio de um período de permanência.

Código de programa	**Comentário**
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_**ON**,xmldial_emb.xml,main","N")	; Indicar diálogo
X200	
Z40	
MMC("POPUPDLG,PICTURE_**OFF**","N")	; Fechar diálogo

Exemplo 3

Inserção de um PopUp-Script em um programa NC e a sua utilização.

Código de programa

```
PROC POPUP_TEST
; ---------------------------- 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>
```

Código de programa

```
;          <while>
;             <condition>count < num</condition>
;             <print name="field_name" text="R-Parameter%d">count</print>
;             <text xpos = "$xpos" ypos = "$ypos" >$$$field_name</text>
;             <op>
;                ypos = ypos + 24;
;                count = count + 1;
;             </op>
;          </while>
;       </paint>
;    </form>
; </main_dialog>
; ========================= Parte do programa ============================
...
G94 F100
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")
G4 F4
X200
MMC("POPUPDLG,PICTURE_OFF","N")
G4 F2
X0
...
```

Condições gerais

- Os arquivos de definição *.com dos diálogos devem ser arquivados na pasta "proj".
- Os arquivos de definição Easy XML *.com dos diálogos devem ser arquivados na pasta "appl".
  Se os arquivos são arquivados em outro índice, a indicação do caminho deve ocorrer de modo indireto, a partir do índice "appl".
- Diálogos definidos para os usuários não podem ser indicados simultaneamente a partir de canais diferentes.

18.8 Tempo de processamento do programa / contador de peças

Para suporte ao operador da máquina-ferramenta são disponibilizadas informações sobre o tempo de processamento e a quantidade de peças produzidas.

Estas informações podem ser processadas como variáveis de sistema no programa NC e/ou PLC. Simultaneamente estas informações são disponibilizadas para exibição na interface de operação.

18.8.1 Tempo de processamento do programa

A função "tempo de processamento do programa" disponibiliza um relógio interno do NC para supervisão de processos tecnológicos, que pode ser lido através de variáveis de sistema de NC e específicos de canal em programas de peça e em ações sincronizadas.

O trigger para medição do tempo de processamento ($AC_PROG_NET_TIME_TRIGGER) é a única variável de sistema gravável da função e serve para a medição seletiva de etapas do programa. Significa que pela programação do trigger no programa NC, a medição de tempo pode ser ativada e desativada;

Variável de sistema	Significado	Atividade
Específico de NC		
$AN_SETUP_TIME	Tempo desde a última inicialização do comando com valores Standard ("partida a frio") em minutos É automaticamente resetado para "0" com valores Standard em cada inicialização do comando.	• sempre ativo
$AN_POWERON_TIME	Tempo desde a última inicialização normal do comando ("reinicialização") em minutos É automaticamente resetado para "0" em cada inicialização normal do comando.	
Específico de canal		
$AC_OPERATING_TIME	Tempo total de processamento de programas NC em modo automático, dado em segundos O valor é automaticamente resetado para "0" em cada inicialização do comando.	• Ativação através do MD27860 • apenas modo de operação AUTOMÁTICO
$AC_CYCLE_TIME	Tempo de processamento do programa NC selecionado, em segundos O valor é automaticamente resetado para "0" com a partida de um novo programa NC.	
$AC_CUTTING_TIME	Tempo de processamento em segundos Aqui se mede o tempo de movimento dos eixos de percurso (pelo menos um está ativo) sem avanço rápido ativo em todos programas NC entre a partida do NC e o fim do programa / NC-Reset. A medição também é interrompida se o tempo de espera estiver ativo. O valor é automaticamente resetado para "0" com valores Standard em cada inicialização do comando.	

Variável de sistema	Significado		Atividade
$AC_ACT_PROG_NET_TIME	Atual tempo de processamento líquido do atual programa NC em segundos É automaticamente resetado para "0" com a partida de um programa NC.		• sempre ativo • apenas modo de operação AUTOMÁTICO
$AC_OLD_PROG_NET_TIME	Tempo líquido de processamento do programa encerrado corretamente com M30 indicado em segundos		
$AC_OLD_PROG_NET_TIME_COUNT	Alterações no $AC_OLD_PROG_NET_TIME Após o POWER ON o $AC_OLD_PROG_NET_TIME_COUNT está em "0". $AC_OLD_PROG_NET_TIME_COUNT sempre é aumentado, se o comando gravar novamente o $AC_OLD_PROG_NET_TIME.		
$AC_PROG_NET_TIME_TRIGGER	Disparo para medição do tempo de processamento:		• apenas modo de operação AUTOMÁTICO
	0	Estado neutro O disparo não está ativo.	
	1	Finalizar Encerra a medição e copia o valor a partir de $AC_ACT_PROG_NET_TIME em $AC_OLD_PROG_NET_TIME. $AC_ACT_PROG_NET_TIME é colocado em "0" e continua funcionando depois.	
	2	Start Inicia a medição e no processo coloca $AC_ACT_PROG_NET_TIME em "0". $AC_OLD_PROG_NET_TIME não é alterado.	
	3	Stop Cessa a medição. Não altera o $AC_OLD_PROG_NET_TIME e mantém o $AC_ACT_PROG_NET_TIME sempre constante até a continuação.	
	4	Continuação Continuação da medição, ou seja uma medição interrompida previamente é retomada. $AC_ACT_PROG_NET_TIME continua funcionando. $AC_OLD_PROG_NET_TIME não é alterado.	
Através do POWER ON todas variáveis de sistema são passadas para "0"!			

Indicação

Fabricante da máquina

A ativação do temporizador ativável é realizada através do dado de máquina MD27860 $MC_PROCESSTIMER_MODE.

O comportamento das medições de tempo ativas em determinadas funções (p. ex. GOTOS, Override = 0%, avanço de teste ativo, teste de programa, ASUP, PROG_EVENT, …) é configurado através dos dados de máquina MD27850 $MC_PROG_NET_TIMER_MODE e MD27860 $MC_PROCESSTIMER_MODE.

Literatura:
Manual de funcionamento das funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1), Capítulo: Tempo de processamento do programa

Indicação

Tempo restante para uma peça de trabalho

Quando peças de trabalho iguais são produzidas em sequência, é possível apurar o tempo restante para uma peça de trabalho com os seguintes valores do timer:

- Tempo de processamento da última peça de trabalho produzida (veja $AC_OLD_PROG_NET_TIME)
- Tempo atual de processamento (veja $AC_ACT_PROG_NET_TIME)

O tempo restante é indicado na interface de operação, junto com o atual tempo de processamento.

Indicação

Utilização do STOPRE

As variáveis de sistema $AC_OLD_PROG_NET_TIME e $AC_OLD_PROG_NET_TIME_COUNT não geram nenhuma parada implícita de pré-processamento. Na utilização no programa de peça isto não é crítico, se o valor das variáveis de sistema são herdadas da execução de programa anterior. Entretanto, se o disparo para medição de tempo de processamento ($AC_PROG_NET_TIME_TRIGGER) for gravado com alta frequência e com isso o $AC_OLD_PROG_NET_TIME for alterado com muita frequência, então se deve utilizar um `STOPRE` implícito no programa de peça.

Condições gerais

- **Localização de blocos**
  Na localização de blocos não é determinado nenhum tempo de processamento do programa.
- **REPOS**
  O tempo de um processo REPOS é considerado no cálculo do atual tempo de processamento ($AC_ACT_PROG_NET_TIME).

Exemplos

Exemplo 1: Medição do tempo do "mySubProgrammA"

Código de programa

```
...
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
```

Depois que o programa processar a linha N80, no $AC_OLD_PROG_NET_TIME estará gravado o tempo líquido de processamento do "mySubProgrammA".

O valor do $AC_OLD_PROG_NET_TIME:

- é mantido além do M30.
- é atualizado depois de cada execução de loop.

Exemplo 2: Medição de tempo do "mySubProgrammA" e do "mySubProgrammC"

Código de programa

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
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 Contador de peças

A função "Contador de peças" disponibiliza diversos contadores, que podem ser utilizados especialmente para a contagem de peças, interna do comando.

Os contadores existem como variáveis de sistema específicas de canal com acesso de gravação e leitura, em uma faixa de valores de 0 até 999 999 999.

Variável de sistema	Significado
$AC_REQUIRED_PARTS	Número de peças que devem ser produzidas (número de peças nominal) Neste contador pode-se definir o número de peças que, ao ser alcançado, passa o número de peças real ($AC_ACTUAL_PARTS) para "0"..
$AC_TOTAL_PARTS	Número total de peças produzidas (número de peças real total) Este contador indica o número de todas peças produzidas desde o momento da partida. O valor é automaticamente resetado para "0" com valores Standard somente com uma inicialização do comando.
$AC_ACTUAL_PARTS	Número de peças produzidas (número de peças real) Neste contador é registrado o número de todas peças produzidas desde o momento da partida. Ao ser alcançado o número de peças nominal ($AC_REQUIRED_PARTS) o contador é automaticamente passado para "0" ($AC_REQUIRED_PARTS > 0 é a condição prévia).
$AC_SPECIAL_PARTS	Número de peças contado pelo usuário Este contador permite ao usuário uma contagem de peças conforme sua própria definição. Uma emissão de alarme pode ser definida com o alcance do número de peças nominal ($AC_REQUIRED_PARTS). O zeramento do contador deve ser realizado pelo próprio usuário.

Indicação

Todos contadores de peças são passados para "0" com valores padrão na fase de inicialização do comando e podem ser lidos e gravados independentemente de sua ativação.

Indicação

Através de dados de máquina específicos de canal pode-se influenciar sobre a ativação de contadores, o momento do zeramento e o algoritmo de contagem.

Indicação

Contagem de peças com comando M definido por usuário

Os dados de máquina podem ser ajustados de modo que os impulsos de contagem para os diversos contadores de peças sejam disparados através de comandos M definidos pelo usuário ao invés do fim do programa `M2`/`M30`.

18.9 Process DataShare - Saída para um dispositivo/arquivo externo (EXTOPEN, WRITE, EXTCLOSE)

A gravação de dados de um programa de peça em um aparelho/arquivo externo ocorre em três passos:

1. Abertura do dispositivo/arquivo externo
   Com o comando EXTOPEN é aberto o dispositivo/arquivo externo para tarefas de gravação para o canal.
2. Gravação de dados
   Os dados de saída podem ser preparados com as funções de string da linguagem NC, p. ex. SPRINT. A gravação propriamente dita é realizada através do comando WRITE.
3. Fechamento do dispositivo/arquivo externo
   Com o comando EXTCLOSE ou com a chegada do fim do programa (M30), assim como no caso de Reset de canal, o dispositivo ou arquivo externo ocupado com o canal será novamente liberado.

Sintaxe

```
DEF INT <Result>
DEF STRING[<n>] <Output>
…
EXTOPEN(<Result>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
…
<Output>="Dados de saída"
WRITE(<Result>,<ExtDev>,<Output>)
…
EXTCLOSE(<Result>,<ExtDev>)
```

Significado

EXTOPEN:	procedimento pré-definido para abrir um aparelho/arquivo externo		
<Resultado>:	**Parâmetro 1:** Variável de resultado Com base no valor de variáveis de resultado no programa torna-se possível interpretar o êxito da operação no programa e, então, prosseguir de acordo.		
	Tipo:	INT	
	Valores:	0	nenhum erro
		1	dispositivo externo não pode ser aberto
		2	dispositivo externo não foi configurado
		3	dispositivo externo configurado com caminho incorreto
		4	sem direitos de acesso para dispositivo externo
		5	Modo de utilização: dispositivo externo já reservado como "exclusivo"
		6	Modo de utilização: dispositivo externo já reservado como "shared"
		7	Tamanho de arquivo maior que LOCAL_DRIVE_MAX_FILESIZE
		8	número máximo de dispositivos externos excedido
		9	Opção para LOCAL_DRIVE não foi criada
		11	A interface V.24 já se encontra ocupada pela função EasyMessage (apenas 828D)
		12	Modo de gravação: Informação contraditória ao extdev.ini
		16	programado caminho externo inválido
		22	dispositivo externo não fixado em circuito

<ExtDev>:	**Parâmetro 2:** Identificador simbólico para o dispositivo/arquivo externo a ser aberto	
	Tipo: STRING	
	O identificador simbólico é composto de: 1. nomes de dispositivo lógicos 2. eventualmente seguido por um caminho de arquivo (anexado com "/"). Estão definidos os seguintes **nomes de dispositivo lógicos**:	
	"LOCAL_DRIVE":	Cartão CompactFlash local (pré-definido)
	"CYC_DRIVE":	indicação de unidade reservada para utilização em ciclos da SIEMENS (pré-definida)
	"/dev/ext/1",... "/dev/ext/9":	Unidades de rede disponíveis **Nota:** Configuração disponível no arquivo extdev.ini!
	"/dev/cyc/1", "/dev/cyc/2":	indicações de unidade reservadas para utilização em ciclos da SIEMENS **Nota:** Configuração disponível no arquivo extdev.ini!
	"/dev/v24":	Interface V.24 **Nota:** Configuração disponível no arquivo extdev.ini!
	Caminho do arquivo: • Para "LOCAL_DRIVE" e "CYC_DRIVE" deve ser informado um caminho de arquivo, p. ex.: "LOCAL_DRIVE/my_dir/my_file.txt" • Por configuração, os nomes de dispositivo lógicos "/dev/ext/1...9" e "/dev/cyc/1...2" podem: – ser atribuídos diretamente a um arquivo, onde apenas o nome de dispositivo lógico deverá ser informado, p. ex.: "/dev/ext/4" – ou a um diretório, onde será necessário informar um caminho de arquivo, p. ex.: "/dev/ext/5/my_dir/my_file.txt" • Ao "/dev/v24" não deve ser adicionado nenhum caminho de arquivo.	
	Nota: Para os nomes de dispositivo lógicos "/dev/ext/1...9", "/dev/v24" e "/dev/cyc/1...2" é ignorado o modo de escrita em letras maiúsculas e minúsculas, e na indicação do caminho de um arquivo a escrita maiúscula e minúscula já é relevante. Para "LOCAL_DRIVE" e "CYC_DRIVE" somente é permitida a escrita em letras maiúsculas.	

<SyncMode>:	**Parâmetro 3:** Modo de processamento para os comandos WRITE para este dispositivo/arquivo		
	Tipo:	STRING	
	Valores:	"SYN":	Gravação síncrona A execução do programa é cessada até o processo de gravação ser concluído. A finalização da gravação síncrona realizada com êxito pode ser verificada através da interpretação da variável de erro do comando WRITE.
		"ASYN":	Gravação assíncrona A execução do programa não é interrompida através do comando WRITE. **Nota:** A variável de resultado do comando WRITE não é declarável neste modo e sempre terá o valor 0 (sem erros). Neste modo não há nenhuma segurança de que o comando WRITE foi executado com sucesso.
<AccessMode>:	**Parâmetro 4:** Modo de utilização para este dispositivo/arquivo		
	Tipo:	STRING	
	Valores:	"SHARED":	O dispositivo/arquivo é solicitado em modo "Shared". Outros canais também utilizar o dispositivo, isto é, também abrir neste modo.
		"EXCL":	O dispositivo/arquivo é utilizado no canal de modo exclusivo, nenhum outro canal poderá utilizar o dispositivo.
<WriteMode>:	**Parâmetro 5:** Modo de gravação para os comandos WRITE para este dispositivo/arquivo (opcional)		
	Tipo:	STRING	
	Valores:	"APP":	Anexos O arquivo permanece com seu conteúdo preservado, as chamadas de gravação são adicionadas no fim.
		"OVR":	Sobrescrever O conteúdo do arquivo é apagado e recriado pelas chamadas de gravação subsequentes.
	Nota: Com este parâmetro o modo de gravação configurado no arquivo extdev.ini não poderá ser sobrescrito. Em caso de conflito, a chamada EXTOPEN é quitada com erro.		

WRITE:	Procedimento pré-definido para gravar os dados de saída

EXTCLOSE:	procedimento pré-definido para fechar um aparelho/arquivo externo aberto		
<Resultado>:	**Parâmetro 1:** Variável de resultado		
	Tipo:	INT	
	Valores:	0	nenhum erro
		16	programado caminho externo inválido
		21	Erro durante o fechamento do dispositivo externo
<ExtDev>:	**Parâmetro 2:** identificador simbólico para o dispositivo/descrição de arquivo externo a ser fechado, veja em EXTOPEN! **Nota:** O identificador deve ser idêntico ao identificador especificado na solicitação EXTOPEN!		

Exemplo

Código de programa

```
N10       DEF INT RESULT
N20       DEF BOOL EXTDEVICE
N30       DEF STRING[80] OUTPUT
N40       DEF INT FASE
N50       EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60       IF RESULT > 0
N70          MSG("Erro no EXTOPEN:" << RESULT)
N80       ELSE
N90          EXTDEVICE=TRUE
N100      ENDIF
…
N200      FASE=4
N210      IF EXTDEVICE
N220         OUTPUT=SPRINT("Fim de fase: %D",PHASE)
N230         WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240      ENDIF
…
```

Veja também

Operações de String (Página 89)

Gravar arquivo (WRITE) (Página 152)

18.10 Alarmes (SETAL)

Em um programa NC podem ser introduzidos alarmes. Estes são representados na interface de operação por um campo especial. Com um alarme sempre está associada uma reação do comando em função da categoria do alarme.

Literatura:
Para mais informações sobre reações de alarmes, veja o manual de colocação em funcionamento.

Sintaxe

```
SETAL(<número do alarme>[,<sequência de caracteres>])
```

Significado

SETAL:	Palavra-chave para programação de um alarme. O SETAL deve ser programado em um bloco NC próprio.	
<número de alarme>:	Variável de tipo INT. Contém o número do alarme. A faixa válida de números de alarme está entre 60000 e 69999, sendo que os números do 60000 ao 64999 estão reservados para ciclos da SIEMENS e do 65000 ao 69999 estão disponíveis para o usuário.	
<sequência de caracteres>:	Na programação de alarmes de ciclos de usuário também se pode indicar uma sequência de caracteres de até 4 parâmetros. Nestes parâmetros podem ser definidos textos variáveis de usuários. Entretanto, também estão disponíveis os seguintes parâmetros pré-definidos:	
	Parâmetro	**Significado**
	%1	Número de canal
	%2	Número do bloco, Label
	%3	Índice de texto para alarmes de ciclo
	%4	parâmetros de alarme adicionais

Indicação

Os textos de alarme devem ser configurados na interface de operação.

Indicação

Para disparar uma emissão de alarme no idioma que estiver ativado na interface de operação, o usuário precisa de informações sobre o atual idioma configurado na HMI. Esta informação pode ser consultada no programa de peça e nas ações síncronas através da variável de sistema $AN_LANGUAGE_ON_HMI (veja em "Atual idioma na HMI (Página 960)").

Exemplo

```
Código de programa                  Comentário
...
N100 SETAL(65000)                   ; Definir o alarme nº 65000
```

Código de programa	Comentário
...	

18.11 Parada e retrocesso ampliados (ESR)

A função "Parada e processo ampliados", denominado com ESR, oferece a oportunidade de reagir de modo flexível em situações de erro:

- **Parada ampliada**
  Enquanto a situação específica de erro permitir, será parado em ordem todos os eixos liberados para a parada ampliada.
- **Retrocesso**
  A ferramenta encontrada no Override será puxada o mais rápido possível pela peça.
- **Operação do gerador (SINAMICS-função de acionamento "Regulamento Vdc")**
  No subdesvio de uma valor parametrizável da tensão do círculo intermediário,ex. Na queda de tensão da rede, será gerado um acionamento (gerador) para a retração da energia elétrica necessária através da alimentação da energia de freio.

Circuitos de disparo

Fontes gerais (NC externo/NC global ou específico de BAG/canal)

- Entradas digitais (ex. no módulo NCU) e/ou a imagem reinterpretável e de controle interno das saídas digitais
- Condição do canal ($AC_STAT)
- Sinais VDI ($A_DBB)
- Mensagens coletivas de um número de alarmes ($AC_ALARM_STAT)

Fontes de eixos

- Valor-limite de retrocesso de emergência do eixo-escravo (sincronização do acoplamento eletrônico, $VC_EG_SYNCDIFF[<eixo-escravo>])
- Acion.: Valor-limite de advertência do circuito intermediário (ameaça de subtensão), $AA_ESR_STAT[<eixo>]
- Acion.: Valor-limite da rotação mínima do gerador (falta de energia cinética alimentável), $AA_ESR_STAT[<eixo>].

Lógica de ligação das ações sincronizadas estáticas: Ligação de fontes e reações

As possibilidades de ligação flexíveis das ações sincronizadas estáticas são utilizadas para ativar determinadas reações em função das fontes.

A ligação de todas fontes relevantes com o auxílio das ações sincronizadas estáticas é de responsabilidade do usuário. Este pode avaliar as variáveis de sistema das fontes de forma integral ou seletiva com a ajuda das telas de Bits, e ligar as reações desejadas. As ações sincronizadas estáticas estão ativas em todos modos de operação.

Literatura:
Manual de funcionamento para ações sincronizadas

Ativação

Liberação da função:

As funções de operação do gerador, parada e retrocesso são liberados ativando-se o respectivo sinal de controle $AA_ESR_ENABLE. Este sinal de controle pode ser alterado por ações sincronizadas.

Desarme da função

ESR será disparado para todos os eixos liberados por meio do posicionamento da variável de sistema $AC_ESR_TRIGGER.

O modo gerador é "automaticamente" ativado no acionamento quando se detecta a ameaça de subtensão do circuito intermediário.

A parada e/ou retrocesso independente do acionamento é ativado quando se detecta uma falha de comunicação (entre NC e o acionamento) assim como quando se detecta uma subtensão do circuito intermediário no acionamento (sempre que haja configuração e liberação).

A parada e/ou retrocesso independente do acionamento também pode ser ativado a partir do lado NC ativando-se o respectivo sinal de controle $AN_ESR_TRIGGER (comando broadcast em todos acionamentos).

Literatura

Para informações detalhadas sobre ESR, consulte:

Manual de funcionamento, funções especiais; parada e retrocesso ampliados (R3)

18.11.1 ESR controlado por NC

18.11.1.1 Retrocessos controladas por NC (POLF, POLFA, POLFMASK, POLFMLIN)

Para retrocesso controlado pelo NC é necessário certos critérios de saída (veja "Retrocessos controladas por NC (POLF, POLFA, POLFMASK, POLFMLIN) (Página 697)"). Se esses pré-requisitos estiverem preenchidos, então será ativado o levantamento rápido (LIFTFAST) do(s) eixo(s) de retrocesso configurado(s) através do posicionamento da variável de sistema $AC_ESR_TRIGGER (ou $AA_ESR_TRIGGER para eixos únicos).

Sintaxe

```
POLF(<eixo>)=<Posição>
POLFA(<eixo>,<Tipo>,<Posição>)
POLFMASK(<eixo_1>,<eixo_2>,...)
POLFMLIN(<eixo_1>,<eixo_2>,...)
```

Para `POLFA` são permitidas as seguintes formas abreviadas:

```
POLFA(<eixo>,<Tipo>) ; Forma abreviada para retrocesso de eixo
individual
```

```
POLFA(eixo,0/1/2) ; Desativação ou ativação rápida
POLFA(eixo,0,$AA_POLFA[eixo]) ; Causa uma parada de avanço
POLFA(eixo,0) ; Causa uma parada de avanço
```

Significado

POLF:	Endereço para informar a posição-alvo do eixo de retrocesso POLF é ativado modalmente.			
	<eixo>:	Nome do eixo geométrico ou eixo de canal/máquina que se retrocede		
	<posição>:	Posição de retrocesso		
		Tipo:	REAL	
		Para o eixo geométrico vale WKS, senão MKS. Para identificadores iguais para eixo geométrico, de canal e de máquina será retirado no WKS.		
POLFA:	Chamada de subprograma predefinido para indicação da posição de retração dos eixos únicos			
	<eixo>:	Identificador de canal		
	<tipo>:	Modo de indicação de posição		
		Tipo:	INT	
		Valor:	0:	Marcar valor de posição como inválido
			1:	O valor de posição é absoluto
			2:	O valor de posição é incremental (distância)
		Observação: Se o eixo não for um eixo individual ou se faltar o tipo ou o tipo=0, então será emitido o alarme correspondente.		
	<posição>:	Posição de retrocesso (s. o.)		
		Observação: O valor também é aceito com tipo=0. Só que este valor é marcado como inválido e ele deve ser reprogramado para o retrocesso.		
POLFMASK:	Chamada definida do subprograma para escolha dos eixos, os quais devem recuar independentes um do outro após o início da retração rápida **independentemente** de ser aplicado.			
	<eixo_1>,…:	Nome dos eixos que devem se deslocar à suas posições definidas com POLF na retração rápida. Todos eixos especificados devem estar no mesmo sistema de coordenadas.		
	POLFMASK() sem indicação de um eixo, a retração rápida desativa para todos os eixos que foram recuados.			
POLFMLIN:	Chamada definida do subprograma para escolha dos eixos, os quais devem recuar independentes um do outro após o início da retração rápida **deve ser retirada** na relação linear.			
	<eixo_1>,...:	s. o.		
	POLFMLIN() sem indicação de um eixo, a retração rápida desativa para todos os eixos que foram recuados com contexto linear.			

Indicação

Antes da retração rápida ser liberada em uma posição fixa através do POLFMASK ou POLFMLIN, deve ser programada uma posição com POLF para os eixos selecionados.

Indicação

Se os eixos foram liberados sucessivamente com POLFMASK, POLFMLIN ou POLFMLIN, POLFMASK, sempre se aplica a última definição do respectivo eixo.

Indicação

As posições programadas com POLF e a ativação através do POLFMASK ou POLFMLIN são canceladas ao ser iniciado o programa de peça. Isto significa que o usuário deve reprogramar em todo programa de peça os valores de POLF e os eixos selecionados no POLFMASK ou POLFMLIN.

Indicação

Se apenas for alterado o tipo na utilização das formas abreviadas POLFA, então o usuário deve garantir que a posição de retrocesso ou o percurso de retrocesso tenha um valor racional. Principalmente a posição de retrocesso e o percurso de retrocesso devem ser redefinidos após o Power On.

Exemplo

Retrocesso de um eixo individual:

Código de programa	Comentário
MD37500 $MA_ESR_REACTION[AX1]=21	; Retrocesso controlado pelo NC.
...	
$AA_ESR_ENABLE[AX1]=1	
POLFA(AX1,1,20.0)	; O AX1 é atribuído à posição axial de retrocesso 20.0 (absoluta).
$AA_ESR_TRIGGER[AX1]=1	; A partir daqui começa o retrocesso.

Outras informações

Requisitos para retrocesso controlado pelo NC

- No canal está programado um eixo de retrocesso para o retrocesso controlado pelo NC:
  MD37500 $MA_ESR_REACTION = 21
- ESR precisa estar livre para o eixo:
  $AA_ESR_ENABLE = 1
- definições de atraso:
  MD21380 $MC_ESR_DELAY_TIME1
  MD21381 $MC_ESR_DELAY_TIME2

- No programa de peças, os eixos específicos de posições de retrocesso estão programados com POLF
- Os eixos são escolhidos com POLFMASK/POLFMLIN para o retrocesso controlado pelo NC.
- Para o movimento de retrocesso devem ser ativados e conservados os sinais de liberação.

Liberar e iniciar o retrocesso controlado pelo NC

Se for definida a variável de sistema $AC_ESR_TRIGGER = 1 e se neste canal houver um eixo de retrocesso configurado (isto é MD37500 $MA_ESR_REACTION = 21) e para este eixo estiver definido $AA_ESR_ENABLE = 1, então neste canal se ativará a retração rápida (LIFTFAST).

O movimento de retração do(s) eixo(s) selecionado com POLF (ou. LFPOS) configurado com POLFMASK ou POLFMLIN substitui o movimento de percurso definido para este(s) eixo(s) no programa de peça.

Para o retrocesso está disponível, no máximo, a soma dos tempos do MD21380 $MC_ESR_DELAY_TIME1 e MD21381 $MC_ESR_DELAY_TIME2 Uma vez transcorrido este tempo, também é iniciada uma desaceleração rápida do eixo de retrocesso, seguido do acompanhamento.

Indicação

O movimento de retrocesso ampliado (isto é, ativado pelo $AC_ESR_TRIGGER LIFTFAST/ LFPOS) **não pode ser interrompido** e somente pode cancelado previamente com PARADA DE EMERGÊNCIA.

Indicação

Através do retrocesso ativado pelo $AC_ESR_TRIGGER</code> é realizado o travamento contra o retrocesso múltiplo.

Retrocesso de eixo individual

No retrocesso de eixo individual com POLFA a posição de retrocesso do eixo individual precisa ter sido programada e as seguintes condições precisam ser mantidas:

- $AA_ESR_ENABLE = 1
- <eixo> precisa ser um eixo único no momento do disparo ($AA_ESR_TRIGGER = 1)
- <Tipo> precisa ser ou 1 ou 2.

Sentido de retrocesso na retração rápida

Considera-se o Frame válido no momento da ativação da retração rápida.

Indicação

Os Frames com rotação também controlam o sentido de retração através do POLF.

Troca de eixos

Os eixos de retrocesso sempre devem ser atribuídos exatamente para um canal e não podem ser trocados entre os canais. Na tentativa de trocar um eixo de retração em um outro canal,

um alarme será acionado. Somente depois que este eixo for desativado novamente com $AA_ESR_ENABLE[AX] = 0 que o eixo pode ser trocado para um novo canal. Após a conclusão da troca de eixos que os eixos podem ser reativados com $AA_ESR_ENABLE[AX] = 1.

Eixos neutros

Os eixos neutros não podem executar nenhum ESR controlado por NC.

18.11.1.2 Parada controlada por NC

Para o eixo de parada configurado no canal será ativada a parada pelo NC pelo posicionamento da variável de sistema $AC_ESR_TRIGGER (ou $AA_ESR_TRIGGER para eixos únicos).

Pré-requisitos

- No canal está programado um eixo de parada para parada controlada pelo NC:
  MD37500 $MA_ESR_REACTION = 22
- ESR precisa estar livre para o eixo:
  $AA_ESR_ENABLE = 1
- definições de atraso:
  MD21380 $MC_ESR_DELAY_TIME1 (atraso dos eixos ESR)
  MD21381 $MC_ESR_DELAY_TIME2 (Tempo ESR para frenagem interpolar)

Sequência de operação

Para o período de tempo no MD21380, o eixo segue interpolando sem perturbações de acordo com a programação. Uma vez transcorrido este período no MD21380 é iniciada a desaceleração por interpolação (parada por rampa). Para a frenagem interpolatória está disponível o intervalo máximo no MD21381. Após este intervalo, é iniciada uma desaceleração rápida seguida do acompanhamento.

Exemplo

Parada de um eixo individual

Código de programa	Comentário
MD37500 $MC_ESR_REACTION[AX1] = 22	; Parada controlada por NC.
MD21380 $MC_ESR_DELAY_TIME1[AX1]=0.3	
MD21381 $MC_ESR_DELAY_TIME2[AX1]=0.06	
...	
$AA_ESR_ENABLE[AX1]=1	
$AA_ESR_TRIGGER[AX1]=1	; A partir daqui começa a parada.

18.11.2 Comando autárquico ESR

18.11.2.1 Configuração da parada independente de acionamento (ESRS)

Com a função `ESRS(...)` são configurados os parâmetros de acionamento para a "Parada" da função ESR independente de acionamento.

Sintaxe

```
ESRS(<eixo_1>,<tempo_de_parada_1>[,...,<eixo_n>,<tempo_de_parada_n>]
)
```

Significado

`ESRS(...)`:	Função para gravar os parâmetros de acionamento para a "Parada" da função ESR A função: • deve estar isolada em um bloco. • ativa a parada de pré-processamento. • não pode ser utilizada em ações síncronas.	
`<eixo_1>`, `...`, `<eixo_n>`:	Eixo para o qual deve ser configurada a parada independente de acionamento O parâmetro de acionamento p0888 (configuração) é gravado no acionamento para este eixo: p0888 = 1	
	Tipo:	AXIS
	Faixa de valores:	Identificador de canal
`<tempo_de_parada_1>`, `...`, `<tempo_de_parada_n>`:	Duração em que o acionamento deve funcionar constante com o atual setpoint de rotação logo após a ocorrência de uma falha O parâmetro de acionamento p0892 (estágio de temporizador) é gravado no acionamento para o eixo indicado: p0892 = <tempo de parada>	
	Unidade:	s
	Tipo:	REAL
	Faixa de valores:	0.00 - 20.00
Em uma chamada de função podem ser programados no máximo 5 eixos; n = 5		

18.11.2.2 Configuração do retrocesso independente de acionamento (ESRR)

Com a função `ESRR(...)` são configurados os parâmetros de acionamento para o "Retrocesso" da função ESR independente de acionamento.

Sintaxe

```
ESRR(<eixo_1>,<curso_de_retrocesso_1>,<velocidade_de_retrocesso_1>[,
...,<eixo_n>,<curso_de_retrocesso_n>,<velocidade_de_retrocesso_n>])
```

Significado

`ESRR(...)`:	Função para gravar os parâmetros de acionamento para o "Retrocesso" da função ESR A função: • deve estar isolada em um bloco. • ativa a parada de pré-processamento. • não pode ser utilizada em ações síncronas.	
`<eixo_1>, ..., <eixo_n>`:	Eixo para o qual deve ser configurado o retrocesso independente de acionamento O parâmetro de acionamento p0888 (configuração) é gravado no acionamento para este eixo: p0888 = 2	
	Tipo:	AXIS
	Faixa de valores:	Identificador de canal
`<curso_de_retrocesso_1>, ..., <curso_de_retrocesso_n>`:	O curso de retrocesso é convertido em um número de rotações de retrocesso para o acionamento. O valor é gravado no parâmetro de acionamento p0893 (rotação) para o eixo indicado: p0893 = (<curso_de_retrocesso_n> convertido em número de rotações de retrocesso)	
	Unidade:	mm/min, pol./min, graus/min (em função da unidade do eixo)
	Tipo:	REAL
	Faixa de valores:	MIN - MAX
`<velocidade_de_ retrocesso_1>, ..., <velocidade_de_ retrocesso_n>`:	A velocidade de retrocesso é convertida em um período de tempo para o acionamento. O valor é gravado no parâmetro de acionamento p0892 (estágio de temporizador) [s] para o eixo indicado: p0892 = <curso_de_retrocesso_n> / <velocidade_de_retrocesso_n>	
	Unidade:	mm/min, pol./min, graus/min (em função da unidade do eixo)
	Tipo:	REAL
	Faixa de valores:	0.00 - MAX
Em uma chamada de função podem ser programados no máximo 5 eixos; n = 5		

18.12 Definir a peça bruta (WORKPIECE)

O sistema de comando precisa conhecer a forma e o tamanho de uma peça bruta para poder indicá-la em uma simulação gráfica. O usuário, portanto, tem a possibilidade de definir peças brutas através da interface de usuário ou diretamente no programa NC. As definições de peças brutas são conservadas após o reset (do fim do programa/canal/BAG). Na próxima inicialização do sistema de comando são eliminadas automaticamente.

Sintaxe

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

Significado

`WORKPIECE(...)`:		Procedimento predefinido para a definição de uma peça bruta		
		Parada de pré-processamento:	sim	
		Sozinho no bloco:	sim	
Parâmetro:				
1	`"<WP>"`:	Nome da peça de trabalho (opcional)		
		Tipo de dado:	STRING	
		Uma indicação somente é necessária quando em um canal podem existir várias peças de trabalho. Sem indicação, "WORKP<n>" é adotado automaticamente, sendo <n> o número do canal declarante.		
2	`"<RefP>"`:	Fixação (opcional; somente para máquinas de fresar)		
		Tipo de dado:	STRING	
		Faixa de valores:	"Mesa"	Fixação na mesa fixa
			"A"	Fixação no eixo rotativo A
			"B"	Fixação no eixo rotativo B
			"C"	Fixação no eixo rotativo C
		Pré-requisito: A mesa ou o eixo rotativo precisa estar ativado através do correspondente dado de máquina para a fixação da peça bruta (veja Manual de colocação em funcionamento SINUMERIK Operate).		
3	`"<ZeroOffset>"`:	Deslocamento ajustável do ponto zero para o posicionamento da peça bruta (não programável) A seleção de um deslocamento ajustável do ponto zero para o posicionamento da peça bruta é oferecida na inserção da peça bruta através da interface de usuário. Na definição direta da peça bruta no programa de peças, a peça bruta, porém, refere-se sempre ao deslocamento do ponto zero atualmente válido.		

4	"<Tipo>":	Forma de peça bruta		
		Tipo de dado:	STRING	
		Faixa de valores:	"CYLINDER":	Cilindro
			"PIPE":	Tubo
			"RECTANGLE":	Quadro centralizado
			"BOX":	Quadro
			"N_CORNER":	Polígono
5 ... 12	<Par5> ... <Par12>:	Parâmetro para a descrição da forma da peça bruta		
		Tipo de dado:	REAL	
		O número de parâmetros necessários e seu significado dependem da respectiva forma da peça bruta e do valor do parâmetro de bits. Veja: • Tabela "Parâmetro para a descrição da forma da peça bruta" • Tabela "Parâmetro de bits"		
WORKPIECE():		Uma chamada de WORKPIECE sem parâmetro elimina todas as definições de peças brutas.		
WORPIECE(<WP>):		Uma chamada de WORKPIECE com nome da peça de trabalho elimina apenas esta definição de peça bruta,		

Tabelas 18-1 Parâmetro para a descrição da forma da peça bruta

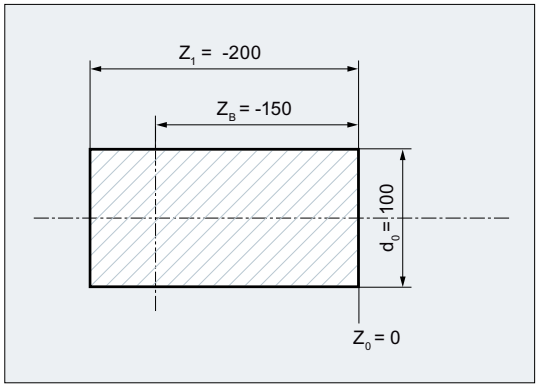
Forma de peça bruta	Parâmetros							
	<Par5>	<Par6>	<Par7>	<Par8>	<Par9>	<Par10>	<Par11>	<Par12>
Cilindro	Parâmetro de bits Valor real, interpretado como número inteiro codificado por bits. Os bits definem o significado dos seguintes parâmetros (veja a tabela "Parâmetros de bits").	Ponto de referência Z_0	Comprimento Z_1	Medida de processamento Z_B	Diâmetro externo d_0	-	Rotação em torno do eixo rotativo	-
Tubo		Ponto de referência Z_0	Comprimento Z_1	Medida de processamento Z_B	Diâmetro externo d_0	Espessura da parede (inc) / diâmetro interno d_1 (abs)	Rotação em torno do eixo rotativo	-
Quadro centralizado		Ponto de referência Z_0	Comprimento Z_1	Medida de processamento Z_B	Largura W	Comprimento L	Rotação em torno do eixo rotativo	-
Quadro		Ponto de referência Z_0	Comprimento Z_1	Medida de processamento Z_B	X_0	Y_0	X_1	Y_1
Polígono		Ponto de referência Z_0	Comprimento Z_1	Medida de processamento Z_B	Número de cantos	Abertura de chave	Rotação em torno do eixo rotativo	-

Tabelas 18-2 Parâmetro de bits

Bit	Significado	
4 (0x0010)	Cuboide X_1	
	=0	inc
	= 1	abs
5 (0x0020)	Cuboide Y_1	
	=0	inc
	= 1	abs
6 (0x0040)	Comprimento Z_1 (medida final)	
	=0	inc
	= 1	abs
Bit 7 (0x0080)	Medida de processamento Z_B	
	=0	inc
	= 1	abs
Bit 8 (0x0100)	Tubo: Espessura da parede / diâmetro interno	
	=0	inc
	= 1	abs
9 (0x0200)	Polígono	
	=0	Abertura de chave
	= 1	Comprimento de canto
12 (0x1000)	Fixação para máquinas de tornear	
	=0	Fuso principal
	= 1	Contrafuso
13 (0x2000)	Contrafuso	
	=0	com espelhamento
	= 1	sem espelhamento

Exemplos

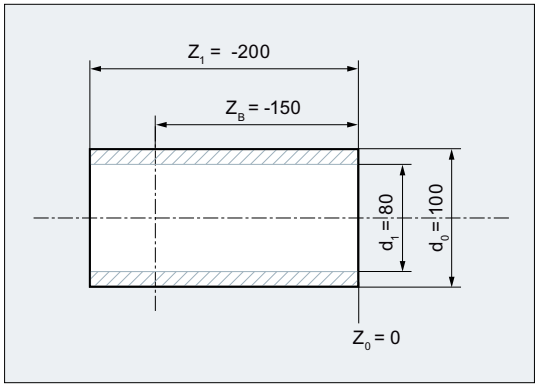
Exemplo 1: Peça bruta cilíndrica em uma máquina de tornear



Código de programa	Comentário
...	

Código de programa	Comentário
WORKPIECE(,,,"CYLINDER",0,0,-200,-150,100)	; **Definição da peça bruta:** ; Forma da peça bruta: Cilindro ; Parâmetro de bits=0(nenhum bit ajustado) → valores para comprimento e medida de processamento são incrementais, peça bruta no fuso principal ; Ponto de referência(Z0)=0 ; Comprimento(Z1)=-200 ; Medida de processamento(ZB)=-150 ; Diâmetro externo(d0)=100
...	

Exemplo 2: Peça bruta em forma de tubo em uma máquina de tornear



Código de programa	Comentário
...	
WORKPIECE(,,,"PIPE",256,0,-200,-150,100,80)	; **Definição da peça bruta:** ; Forma da peça bruta: Tubo ; Parâmetro de bits=256(Bit8=1) → diâmetro interno é absoluto, comprimento e medida de processamento são incrementais, peça bruta no fuso principal ; Ponto de referência(Z0)=0 ; Comprimento(Z1)=-200 ; Medida de processamento(ZB)=-150 ; Diâmetro externo(d0)=100 ; Diâmetro interno(d1)=80
...	

18.13 Comutar modo de linguagem (G290, G291)

O sistema de comando oferece a possibilidade de leitura e processamento de programas de peças provenientes de sistemas CNC externos. Condição prévia é que o correspondente modo de linguagem NC (dialeto ISO) foi determinado na colocação em serviço.

Literatura:
Manual de funcionamento Dialetos ISO

O modo do dialeto ISO pode ser ativado separadamente para cada canal. Por exemplo: canal 1 pode funcionar no modo do dialeto ISO, enquanto canal 2 está ativo no modo SINUMERIK.

A comutação entre o modo SINUMERIK e o modo do dialeto ISO ocorre no programa NC por meio dos comandos do grupo G 47. A ferramenta ativa, as correções de ferramenta e os deslocamentos de ponto zero não são afetados pela comutação.

Sintaxe

```
G291
...
G290
```

Significado

G290:	Ativação do modo de linguagem SINUMERIK	
	Sozinho no bloco:	sim
	Efeito:	modal
G291:	Ativação do modo de linguagem ISO	
	Sozinho no bloco:	sim
	Efeito:	modal

Condições

Modo SINUMERIK

- O ajuste prévio dos comandos G pode ser determinado para cada canal através de um dado de máquina.
- No modo SINUMERIK não é possível programar comandos de linguagem a partir dos dialetos ISO.

Modo do dialeto ISO

- O modo do dialeto ISO como ajuste básico do sistema de comando pode ser ajustado através de dados de máquina. Como padrão, o comando numérico sempre será inicializado em modo de dialeto ISO.
- Somente comandos G a partir do dialeto ISO podem ser programados. A programação de funções SINUMERIK não é possível no modo do dialeto ISO.
- Uma mistura de dialeto ISO e linguagem SINUMERIK no mesmo bloco NC não é possível.

- Uma comutação por meio de comandos G entre dialeto ISO M (fresar) e dialeto ISO-T (tornear) não é possível.
- É possível chamar subprogramas programados no modo SINUMERIK.
- Quando funções SINUMERIK devem ser utilizadas, é necessário comutar primeiro para o modo SINUMERIK (veja o exemplo).

Exemplo

Compressão de blocos lineares no modo do dialeto ISO

Código de programa	**Comentário**
N5 G290	; Ativar o modo de linguagem SINUMERIK
N10 COMPON	; COMPON é um comando da linguagem Siemens que ativa uma função de compressor, a qual substitui blocos lineares consecutivos por blocos de polinômios com os maiores comprimentos de trajeto possíveis.
N15 G291	; Ativar modo de linguagem ISO
N20 G01 X100 Y100 F1000	; Uma vez que a ativação de COMPON ocorreu no modo SINUMERIK, é possível comprimir com esta função também blocos lineares no modo do dialeto ISO.
...	

Programas de desbaste próprios 19

19.1 Funções de apoio para o desbaste

Para a operação de desbaste nos são oferecidos ciclos de usinagem prontos para serem aplicados. Além disso existe a possibilidade de se utilizar as seguintes funções para criação de próprios programas de desbaste:

- Criar tabela de contorno (CONTPRON)
- Criar tabela de contorno codificada (CONTDCON)
- Desativar a preparação de contorno (EXECUTE)
- Determinar a intersecção entre dois elementos de contorno (INTERSEC)
  (Apenas para tabelas que foram criadas com CONTPRON.)
- Executar por blocos os elementos de contorno de uma tabela (EXECTAB)
  (Apenas para tabelas que foram criadas com CONTPRON.)
- Calcular dados de círculos (CALCDAT)

Indicação

Estas funções podem ser utilizadas não apenas para o desbaste, mas de modo universal.

Pré-requisitos

Antes da chamada das funções CONTPRON ou CONTDCON se deve:

- aproximar um ponto de partida que permita uma usinagem livre de colisões.
- desativar a correção do raio da ferramenta com `G40`.

19.2 Criar tabela de contorno (CONTPRON)

Com CONTPRON pode ser ligada a preparação de contorno. Os blocos NC chamados em seguida não serão executados, mas divididos em movimentos individuais e armazenados na tabela de contorno. Cada um dos elementos de contorno corresponde à uma linha na tabela em um campo bidimensional da tabela de contorno. É retornado o número de rebaixados determinados.

Sintaxe

Ativar a preparação de contorno:
```
CONTPRON(<tabela do contorno>,<tipo de usinagem>,<rebaixados>,
<sentido de usinagem>)
```

Desativar a preparação de contorno e retornar ao modo de execução normal:
```
EXECUTE(<ERRO>)
```

Veja "Desativar a preparação de contorno (EXECUTE) (Página 727)"

Significado

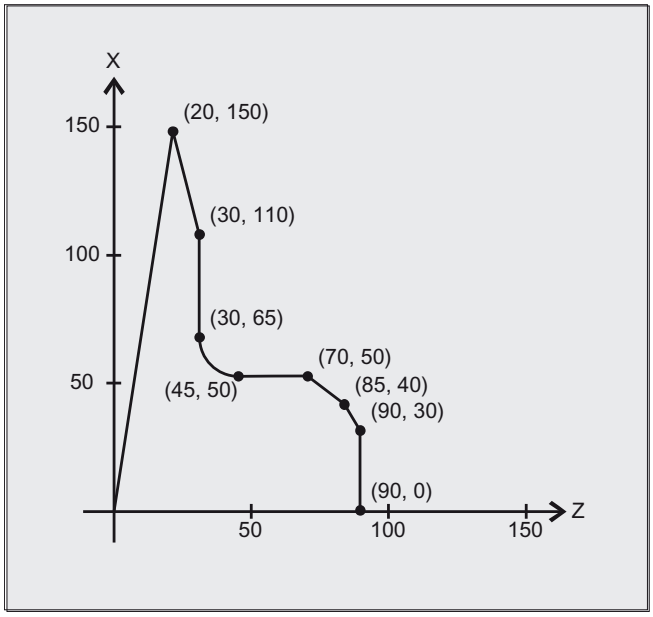
CONTPRON:	Processo predefinido para ligar a preparação de contorno para a criação de uma tabela de contornos		
<Tabela de contornos>:	Nome da tabela de contorno		
<Tipo de usinagem>:	Parâmetro para o tipo de usinagem		
	Tipo:	CHAR	
	Valor:	"G":	Torneamento longitudinal: Usinagem interna
		"L":	Torneamento longitudinal: Usinagem externa
		"N":	Torneamento transversal: Usinagem interna
		"P":	Torneamento transversal: Usinagem externa
<Rebaixados>:	Variável de resultado para o número de elementos de rebaixado ocorridos		
	Tipo:	INT	
<Sentido da usinagem>:	Parâmetro para o sentido da usinagem		
	Tipo:	INT	
	Valor:	0	Preparação de contorno para frente (valor padrão)
		1	Preparação de contorno nos dois sentidos

Exemplo 1

Criação de uma tabela de contorno com:

- nome "KTAB"
- máx. 30 elementos de contorno (círculos, retas)

- uma variável para o número de elementos de rebaixado ocorridos
- uma variável para mensagens de erro



Programa NC:

Código de programa	**Comentário**
N10 DEF REAL KTAB[30,11]	; Tabela de contorno de nome KATB e com o máx. de 30 elementos de contorno, o valor de parâmetro 11 (número de colunas da tabela) é um tamanho fixo.
N20 DEF INT NUMREBAIXADOS	; Variável para a quantidade de elementos de rebaixado de nome NUMREBAIXADOS.
N30 DEF INT ERRO	; Variável para resposta de erro (0=nenhum erro, 1=erro).
N40 G18	
N50 CONTPRON(KTAB,"G",NUMREBAIXADOS)	; Ativar a preparação de contorno.
N60 G1 X150 Z20	; N60 até N120: Descrição do contorno
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(ERRO)	; Finalizar o preenchimento da tabela do contorno, comutar para o modo normal do programa.
N140 …	; Processamento seguinte da tabela.

Tabela de contorno KTAB:

Índice Linha	Coluna									
(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

Explicação do conteúdo das colunas:

(0) Ponteiro no elemento de contorno seguinte (no número de linha do próprio)

(1) Ponteiro no elemento de contorno anterior

(2) Codificação do modo de contorno para o movimento

Possíveis valores para 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) Ponto inicial dos elementos de contorno

(3) = abscissa, (4) = ordenada no plano atual

(5), (6) Ponto final dos elementos de contorno

(5) = abscissa, (6) = ordenada no plano atual

(7) Indicador máx./mín.: identifica os máximos e mínimos no contorno

(8) Valor máximo entre o elemento de contorno e a abscissa (na usinagem longitudinal) ou ordenada (na usinagem transversal). O ângulo está em função do tipo de usinagem programado.

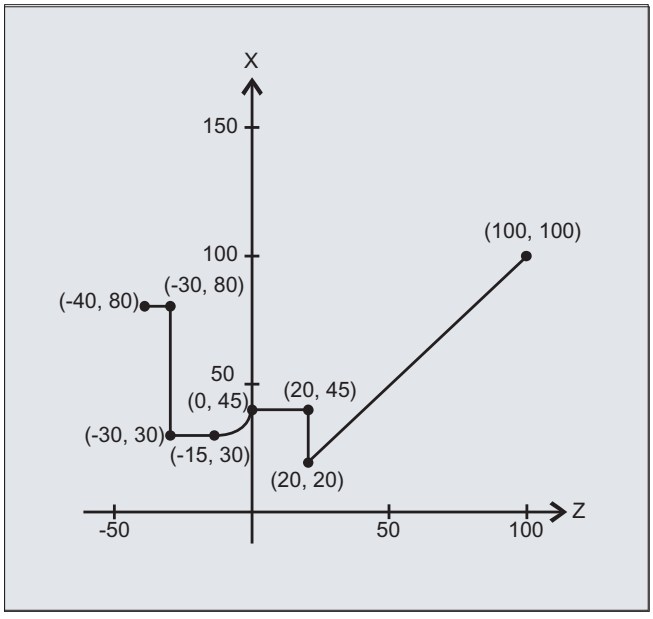
(9), (10) Coordenadas do centro do elemento de contorno, no caso de um bloco de circunferência.

(9) = abscissa, (10) = ordenada

Exemplo 2

Criação de uma tabela de contorno com

- nome KTAB
- máx. 92 elementos de contorno (círculos, retas)
- Modo de operação: Torneamento longitudinal, usinagem externa
- Preparação para frente e para trás



Programa NC:

Código de programa	Comentário
N10 DEF REAL KTAB[92,11]	; Tabela de contorno com nome KTAB e por exemplo máx. de 92 elementos de contorno; o valor de parâmetro 11 é um tamanho fixo.
N20 DEF CHAR BT="L"	; Modo de operação para CONTPRON: Torneamento longitudinal, usinagem externa
N30 DEF INT HE=0	; Número de elementos de rebaixado=0
N40 DEF INT MODE=1	; Preparação para frente e para trás
N50 DEF INT ERR=0	; Resposta de erro
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; Ativar a preparação de contorno.
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)	; Finalizar o preenchimento da tabela do contorno, comutar para o modo normal do programa.
...	

Tabela de contorno KTAB:

Após a finalização da preparação do contorno, o contorno se encontra disponível nos dois sentidos.

Índice	Coluna										
Linha	(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

Explanação dos conteúdos de colunas e das observações sobre as linhas 0, 1, 6, 8, 83, 85 e 91

São aplicadas as explanações de conteúdo de coluna mencionadas no exemplo 1.

Sempre na linha de tabela 0:

1) Precedente: A linha n contém o fim do contorno para frente

2) Posterior: A linha n é o fim da tabela de contorno para frente

Em cada caso uma vez dentro dos elementos de contorno para frente:

3) Precedente: Início do contorno (para frente)

4) Posterior: Fim do contorno (para frente)

Sempre na linha do fim da tabela do contorno (para frente) +1:

5) Precedente: Número de rebaixados para frente

6) Posterior: Número de rebaixados para trás

Em cada caso uma vez dentro dos elementos de contorno para trás:

7) Posterior: Fim de contorno (para trás)

8) Precedente: Início de contorno (para trás)

Sempre na última linha da tabela:

9) Precedente: A linha n é o início da tabela de contorno (para trás)

10) Posterior: A linha n contém o início do contorno (para trás)

Outras informações

Comandos de deslocamento permitidos, sistema de coordenadas

Para programação do contorno são permitidos os seguintes comandos G:

- Grupo G 1: `G0`, `G1`, `G2`, `G3`

Também são possíveis:

- Arredondamento e chanfro
- Programação de círculo através do `CIP` e `CT`

As funções Spline, polinômio e rosca conduzem à erros.

Não são permitidas alterações do sistema de coordenadas através da ativação de um Frame entre `CONTPRON` e `EXECUTE`. O mesmo vale para uma mudança entre `G70` e `G71` ou `G700` e `G710`.

Uma troca dos eixos geométricos com `GEOAX` durante a preparação da tabela do contorno gera um alarme.

Elementos de detalonado

Opcionalmente a descrição do contorno dos elementos de detalonado pode ser realizada em uma subrotina ou em blocos individuais.

Desbaste em função do sentido de contorno programado

A preparação de contorno `CONTPRON` é ampliada de modo que, após sua chamada, a tabela do contorno esteja disponível independentemente do sentido programado.

19.3 Criar tabela de contorno codificada (CONTDCON)

Na preparação de contorno ativado com `CONTDCON` os blocos NC seguintes são armazenados em uma tabela de contorno de 6 colunas que permite a economia de espaço. Cada elemento de contorno corresponde à uma linha na tabela do contorno. Conhecendo-se as regras de codificação especificadas, por exemplo, podemos compor ciclos através de programas de código DIN a partir das linhas da tabela. Na linha de tabela com o número 0 são armazenados os dados do ponto de saída.

Sintaxe

Ativar a preparação de contorno:
`CONTDCON(<tabela de contorno>,<sentido de usinagem>)`

Desativar a preparação de contorno e retornar ao modo de execução normal:
`EXECUTE(<ERRO>)`

Veja "Desativar a preparação de contorno (EXECUTE) (Página 727)"

Significado

`CONTDCON`:	Processo predefinido para ligar a preparação de contorno para a criação de uma tabela de contornos codificada		
`<Tabela de contornos>`:	Nome da tabela de contorno		
`<Sentido da usinagem>`:	Parâmetro para sentido de usinagem		
	Tipo:	INT	
	Valor:	0	Preparação de contorno conforme sequência de contornos (valor padrão)
		1	inadmissível

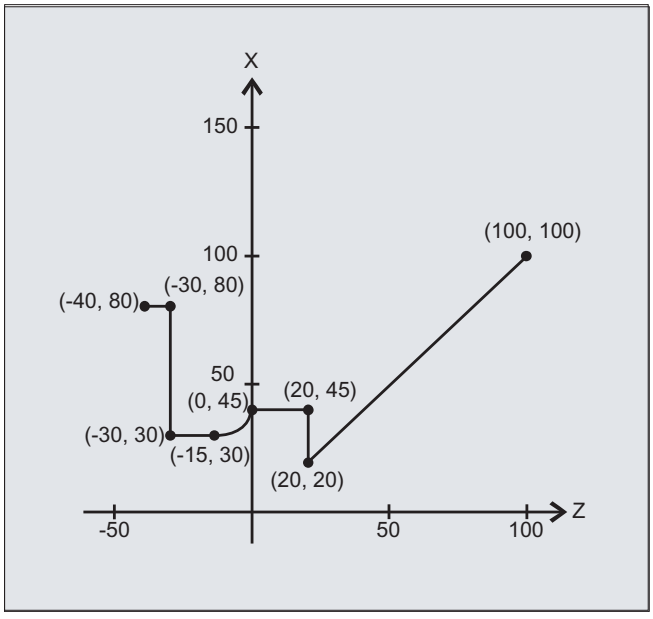
Indicação

Os comandos G permitidos para o `CONTDCON` no segmento de programa a ser tabelado são mais completos que no `CONTPRON`. Além disso, os avanços e o tipo de avanço podem ser salvos juntos por segmento de contorno.

Exemplo

Criação de uma tabela de contorno com:

- nome "KTAB"
- Elementos de contorno (círculos, retas)
- Modo de operação: Torneamento
- Sentido de usinagem: para frente



Programa NC:

Código de programa	**Comentário**
N10 DEF REAL KTAB[9,6]	; Tabela de contorno com nome KTAB e 9 linhas de tabela. Estas permitem 8 blocos de contorno. O valor de parâmetro 6 (número de colunas da tabela) é um tamanho fixo.
N20 DEF INT MODE = 0	; Variável para o sentido de usinagem. Valor padrão 0: somente no sentido programado do contorno.
N30 DEF INT ERROR = 0	; Variável para a resposta de erro.
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; Chamada da preparação de contorno (MODE pode ser suprimido).
N110 G1 Z20 X20 F200	; Descrição do contorno.
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)	; Finalizar o preenchimento da tabela do contorno, comutar para o modo normal do programa.
...	

Tabela de contorno KTAB:

	Índice de coluna					
	0	1	2	3	4	5
Índice de linha	Modo de contorno	Ponto final Abscissa	Ponto final Ordenada	Centro Abscissa	Centro Ordenada	Avanço
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

Explicação do conteúdo das colunas:

Linha 0: Codificações para o **ponto de partida:**

Coluna 0: 10^0 (posição da unidade): G0 = 0
10^1 (posição da dezena): G70 = 0, G71 = 1, G700 = 2, G710 = 3
Coluna 1: Ponto de partida da abscissa
Coluna 2: Ponto de partida da ordenada
Coluna 3-4: 0
Coluna 5: Índice de linha do último segmento de contorno na tabela

Linhas 1-n: Registros dos **segmentos de contorno**

Coluna 0: 10^0 (posição da unidade): G0 = 0, G1 = 1, G2 = 2, G3 = 3
10^1 (posição da dezena): G70 = 0, G71 = 1, G700 = 2, G710 = 3
10^2 (posição da centena): G90 = 0, G91 = 1
10^3 (posição da milhar): G93 = 0, G94 = 1, G95 = 2, G96 = 3
10^4 (posição da dezena de milhar): G60 = 0, G44 = 1, G641 = 2, G642 = 3
10^5 (posição da centena de milhar): G9 = 1
Coluna 1: Ponto final da abscissa
Coluna 2: Ponto final da ordenada
Coluna 3: Centro da abscissa na interpolação circular
Coluna 4: Centro da ordenada na interpolação circular
Coluna 5: Avanço

Outras informações

Comandos de deslocamento permitidos, sistema de coordenadas

Para programação do contorno são permitidos os seguintes grupos G e comandos G:

Grupo G 1:	G0, G1, G2, G3
Grupo G 10:	G60, G64, G641, G642
Grupo G 11:	G9
Grupo G 13:	G70, G71, G700, G710
Grupo G 14:	G90, G91
Grupo G 15:	G93, G94, G95, G96, G961

Também são possíveis:

- Arredondamento e chanfro
- Programação de círculo através do CIP e CT

As funções Spline, polinômio e rosca conduzem à erros.

Não são permitidas alterações do sistema de coordenadas através da ativação de um Frame entre CONTDCON e EXECUTE. O mesmo vale para uma mudança entre G70 e G71 ou G700 e G710.

Uma troca dos eixos geométricos com GEOAX durante a preparação da tabela do contorno gera um alarme.

Sentido de usinagem

A tabela de contorno gerada com CONTDCON está prevista para o desbaste no sentido programado do contorno.

19.4 Determinar a intersecção entre dois elementos de contorno (INTERSEC).

O INTERSEC determina a intersecção de dois elementos de contorno normalizados, obtidos da tabela de contorno gerada com CONTPRON.

Sintaxe

```
<estado>=INTERSEC(<tabela de contorno_1>[<elemento de contorno_1>],
<tabela de contorno_2>[<elemento de contorno_2>],<intersecção>,<tipo
de usinagem>)
```

Significado

INTERSEC:	Função predefinida para a determinação da intersecção de dois elementos de contorno a partir das tabelas criadas com CONTPRON		
<Status>:	Variável para o estado da intersecção		
	Tipo:	BOOL	
	Valor:	TRUE	intersecção encontrada
		FALSE	nenhuma intersecção encontrada
<Tabela de contorno_1>:	Nome da primeira tabela de contorno		
<Elemento de contorno_1>:	Número do elemento de contorno da primeira tabela de contorno		
<Tabela de contorno_2>:	Nome da segunda tabela de contorno		
<Elemento de contorno_2>:	Número do elemento de contorno da segunda tabela de contorno		
<Intersecção>:	Coordenadas da intersecção no plano ativo (G17 / G18 / G19)		
	Tipo:	REAL	
<Tipo de usinagem>:	Parâmetro para o tipo de usinagem		
	Tipo:	INT	
	Valor:	0	Cálculo da intersecção no plano ativo com o parâmetro 2 (valor padrão)
		1	Cálculo de intersecção independente do plano transferido

Indicação

Observe que as variáveis devem ser definidas antes de sua utilização.

A transferência dos contornos requer a conservação dos valores definidos com CONTPRON:

Parâmetro	Significado
2	Codificação do modo de contorno para o movimento
3	Ponto de início do contorno, abscissa
4	Ponto de início do contorno, ordenada
5	Ponto final do contorno, abscissa
6	Ponto final do contorno, ordenada

Parâmetro	Significado
9	Coordenada do centro para a abscissa (somente para contorno circular)
10	Coordenada do centro para a ordenada (somente para contorno circular)

Exemplo

Determinar a intersecção do elemento de contorno 3 da tabela TABNAME1 com o elemento de contorno 7 da tabela TABNAME2. As coordenadas da intersecção no plano ativo são armazenadas na variável ISCOORD (1º elemento = abscissa, 2º elemento = ordenada). Se não existir nenhuma intersecção, ocorre um salto para NOINTERSECT (nenhuma intersecção encontrada).

Código de programa	**Comentário**
`DEF REAL TABNAME1[12,11]`	`; Tabela de contorno 1`
`DEF REAL TABNAME2[10,11]`	`; Tabela de contorno 2`
`DEF REAL ISCOORD[2]`	`; Variável para coordenadas da intersecção.`
`DEF BOOL ISPOINT`	`; Variável para estado da intersecção.`
`DEF INT MODE`	`; Variável para tipo de usinagem.`
…	
`MODE=1`	`; Cálculo independente do plano ativo.`
`N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE)`	`; Chamada da intersecção dos elementos de contorno.`
`N20 IF ISPOINT==FALSE GOTOF NOINTERSECT`	`; Salto até o NOINTERSECT.`
…	

19.5 Executar por blocos os elementos de contorno de uma tabela (EXECTAB)

Com EXECTAB podem ser executados por blocos, os elementos de contorno de uma tabela que, por ex., foi criada com CONTPRON.

Sintaxe

```
EXECTAB(<tabela de contorno>[<elemento de contorno>])
```

Significado

EXECTAB:	Processo predefinido para a execução de um elemento de contorno
<Tabela de contornos>:	Nome da tabela de contorno
<Elemento de contorno>:	Número do elemento de contorno

Exemplo

Os elementos de contorno 0 até 2 da tabela KTAB devem ser executados por blocos.

Código de programa	**Comentário**
N10 EXECTAB(KTAB[0])	; Executar o elemento 0 da tabela KTAB.
N20 EXECTAB(KTAB[1])	; Executar o elemento 1 da tabela KTAB.
N30 EXECTAB(KTAB[2])	; Executar o elemento 2 da tabela KTAB.

19.6 Calcular dados de círculos (CALCDAT)

Com `CALCDAT` podem ser calculados o raio e as coordenadas do centro do círculo através de três ou quatro pontos conhecidos do círculo. Os pontos indicados devem ser diferentes um do outro.

No caso de 4 pontos, que não estão exatamente posicionados sobre o círculo, é utilizado um valor médio para o cálculo do centro e o raio do círculo.

Indicação

Regra de cálculo para a criação do valor médio

O cálculo do arco do círculo é executado 4 x:

1. com o ponto do círculo 1, 2, 3
2. com o ponto do círculo 1, 2, 4
3. com o ponto do círculo 1, 3, 4
4. com o ponto do círculo 2, 3, 4

As coordenadas do centro do círculo, o valor da abscissa e o da ordenada são calculados, somando-se os valores da abscissa ou da ordenada aos quatro cálculos do arco do círculo e dividindo-se por 4.

O raio é calculado, pela raiz quadrada da soma dos quatro raios dos cálculos do arco do círculo e multiplicando o resultado por 0,5.

Sintaxe

```
<estado>=CALCDAT(<pontos do
círculo>[<número>,<tipo>],<número>,<resultado>)
```

Significado

`CALCDAT`:	Função predefinida para o cálculo do raio e das coordenadas do centro de um círculo a partir de 3 ou 4 pontos		
`<Status>`:	Variável para o estado do cálculo do círculo		
	Tipo:	BOOL	
	Valor:	TRUE	Os pontos indicados estão em um círculo.
		FALSE	os pontos indicados **não** estão em um círculo.
`<Pontos do círculo>[]`:	Variável para indicação dos pontos do círculo com os parâmetros:		
	`<Quantidade>`:	Número de pontos do círculo (3 ou 4)	
	`<Tipo>`:	Tipo da indicação de coordenada, p. ex. 2 para coordenadas de 2 pontos	
`<Quantidade>`:	Parâmetro para o número de pontos utilizados no cálculo (3 ou 4)		

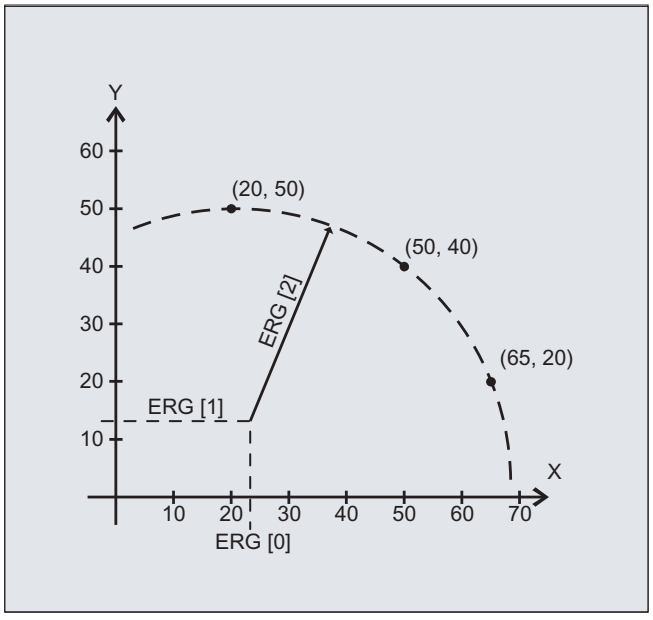
<Resultado>[3]:	Variável para resultado: Indicação de coordenadas do centro do círculo e raio	
	0	Coordenada do centro do círculo: Valor da abscissa
	1	Coordenada do centro do círculo: Valor da ordenada
	2	Raio

Indicação

Observe que as variáveis devem ser definidas antes de sua utilização.

Exemplo

A partir de três pontos deve-se calcular se estes encontram-se em um segmento do círculo.



Código de programa	**Comentário**
N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)	; Variável para indicação dos pontos do círculo.
N20 DEF REAL RES[3]	; Variável para o resultado.
N30 DEF BOOL STATUS	; Variável para estado.
N40 STATUS=CALCDAT(PTS,3,RES)	; Chamada dos dados do círculo calculados.
N50 IF STATUS == FALSE GOTOF ERROR	; Salto para o erro.

19.7 Desativar a preparação de contorno (EXECUTE)

Com EXECUTE a preparação de contorno é desativada e simultaneamente é comutado para o modo normal de execução.

Sintaxe

```
EXECUTE(<ERRO>)
```

Significado

EXECUTE:	Procedimento predefinido para finalizar a preparação de contorno	
<ERRO>:	Variável para resposta de erro	
	Tipo:	INT
	O valor das variáveis indica se o contorno pode ser preparado sem erros:	
	0	Erro
	1	Nenhum erro

Exemplo

Código de programa

```
...
N30 CONTPRON(...)
N40 G1 X... Z...
...
N100 EXECUTE(...)
...
```

Programação externa de ciclos 20

20.1 Ciclos tecnológicos

20.1.1 Introdução

Conteúdo

Este capítulo contém a descrição dos ciclos para as tecnologias de torneamento, fresa, suavização.

Estrutura

A descrição de um ciclo é estruturado do seguinte modo:

- Sintaxe
  Nome de ciclo e sequência de chamada dos parâmetros de transferência
- Parâmetros
  Tabela para explanação dos diversos parâmetros

Descrição dos parâmetros

Na tabela são indicados os seguintes dados a respeito de um parâmetro: Nome, descrição, faixa de valores e relações com os outros parâmetros.

A coluna para a referência ao parâmetro na tela serve para facilitar o recontro dos valores programados pelo sistema de comando, no caso de recompilação das chamadas de ciclo criadas externamente.

Parâmetro "somente para a interface"

Na tabela estão marcados os parâmetros com "somente para a interface". Estes não têm nenhum significado para o funcionamento do ciclo. Eles somente são utilizados para recompilar totalmente as chamadas dos ciclos. Se eles não forem programados, o ciclo também poderá ser recompilado, mas os campos vem marcados com suas cores relativas, e requerem o preenchimento na tela.

Parâmetro "reservado"

Os parâmetros que são descritos com "reservado" devem ser programados com o valor 0 (zero) ou vírgula nula, para que os parâmetros de chamada seguintes estejam associados corretamente com os parâmetros de ciclo internos. Exceção: nos parâmetros de String de valor "" ou vírgula nula.

Repetição de ciclos no modelo de posições

Os ciclos de furação e fresamento podem ser repetidos no modelo de posições (chamadas modais). Então, antes do ciclo, deve ser escrito `MCALL` na mesma linha, p. ex. `MCALL CYCLE83(...)`.

Indicação

Se determinados parâmetros de transferência (por ex., `<_VARI>`, `<_GMODE>`, `<_DMODE>`, `<_AMODE>`) forem programados indiretamente como parâmetros, então a tela de especificação será aberta ao recompilar, porém não pode ser salva, pois não haverá uma atribuição inequívoca para determinados campos de seleção.

20.1.2 Visão geral da tecnologia específica

Na tabela de visão geral a seguir estão os ciclos tecnológicos programáveis externamente disponíveis e disposto a cada tecnologia:

Tecnologia	Ciclo tecnológico
Furação	• CYCLE81 - Furação, centragem (Página 773) • CYCLE82 - Furação, escareamento plano (Página 774) • CYCLE85 - Alargamento (Página 783) • CYCLE86 - Mandrilamento (Página 784) • CYCLE83 - Furação profunda (Página 777) • CYCLE830 - Furação de furo profundo 2 (Página 809) • CYCLE84 - Rosqueamento com macho sem mandril de compensação (Página 780) • CYCLE840 - Rosqueamento com macho com mandril de compensação (Página 818) • CYCLE78 - Fresar rosca (Página 767) • CYCLE802 - Posição definida pelo usuário (Página 806) • HOLES1 - Fileira de furos (Página 732) • CYCLE801 - Grade ou quadro (Página 805) • HOLES2 - Fileira de furos (Página 732)
Torneamento	• CYCLE951- Desbaste (Página 829) • CYCLE930 - Canal (Página 824) • CYCLE940 - Formas de alívio (Página 827) • CYCLE99 - Torneamento de roscas (Página 793) • CYCLE98 - encadeamento de roscas (Página 789) • CYCLE92 - Separação (Página 786)
Torneamento de contorno	• CYCLE62 - Chamada de contorno (Página 752) • CYCLE952 - Usinagem de canal de contorno (Página 832)

Tecnologia	Ciclo tecnológico
Fresamento	• CYCLE61- Fresamento de facear (Página 750) • POCKET3 - Fresamento de bolsão retangular (Página 735) • POCKET4 - Fresamento de bolsão circular (Página 737) • CYCLE76 - Fresar a saliência retangular (Página 763) • CYCLE77 - Fresar a saliência circular (Página 765) • CYCLE79 - Poliedro (Página 771) • SLOT1 - Ranhura longitudinal (Página 740) • SLOT2 - Ranhura circular (Página 743) • CYCLE899 - Fresar calha (Página 821) • LONGHOLE - Furo oblongo (Página 745) • CYCLE70 - Fresamento de roscas (Página 757) • CYCLE60 - Ciclo de gravação (Página 747)
Fresamento de contorno	• CYCLE62 - Chamada de contorno (Página 752) • CYCLE72 - Fresamento de percurso (Página 758) • CYCLE63 - Fresar bolsão de contorno (Página 753) • CYCLE64 -Pré-furação do bolsão de contorno (Página 755)
Retificar	• CYCLE495 - Perfilar (Página 798) • CYCLE435 - definir o sistema de coordenadas de ajuste (Página 798) • CYCLE4071 - Retificação longitudinal com penetração no ponto de inversão (Página 839) • CYCLE4072 - Retificação longitudinal com penetração no ponto de inversão e sinal de interrupção (Página 840) • CYCLE4073 - Retificação longitudinal com penetração contínua (Página 844) • CYCLE4074 - Retificação longitudinal com penetração contínua e sinal de interrupção (Página 845) • CYCLE4075 - Retificação plana com penetração no ponto de inversão (Página 848) • CYCLE4077 - Retificação plana com penetração no ponto de inversão e sinal de interrupção (Página 851) • CYCLE4078 - Retificação plana com penetração contínua (Página 854) • CYCLE4079 - Retificação plana com penetração intermitente (Página 856)
Outros	• CYCLE800 - Rotação (Página 800) • CYCLE832 - High Speed Settings (Página 815)
Todos	• GROUP_BEGIN - Início bloco de programa (Página 858) • GROUP_END - Final bloco de programa (Página 859) • GROUP_ADDEND - Final adendo de entrada (Página 859)

20.1.3 HOLES1 - Fileira de furos

Sintaxe

```
HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Parâmetro

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	X0	<SPCA>	REAL	Ponto de referência para fileira de furos no 1º eixo (abs)		
2	Y0	<SPCO>	REAL	Ponto de referência para fileira de furos no 2º eixo (abs)		
3	α0	<STA1>	REAL	Ângulo de giro básico (ângulo para o 1º eixo)		
4	L0	<FDIS>	REAL	Distância entre o 1º furo e o ponto de referência		
5	L	<DBH>	REAL	Distância entre os furos		
6	N	<NUM>	INT	Número de furos		
7		<_VARI>	INT	reservado		
8		<_UMODE>	INT	reservado		
9		<_HIDE>	STRING [200]	Posições suprimidas • máx. 198 caracteres • Indicação do número de posição subsequente, p. ex. "1,3" (as posições 1 e 3 não são executadas)		
10		<_NSP>	INT	reservado		
11		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)

20.1.4 HOLES2 - Fileira de furos

Sintaxe

```
HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros interno	Tipo de dado	Significado			
1	X0	<CPA>	REAL	Centro para círculo de furos no 1º eixo (abs) Ponto de referência no Eixo 1			(em XY) (em XA, YB, ZC)
2	Y0	<CPO>	REAL	Centro para círculo de furos no 2º eixo (abs) Ponto de referência no Eixo 2			(em XY) (em XA, YB, ZC)
3	R	<RAD>	REAL	Raio do círculo de furos			(em XY)
4	α0	<STA1>	REAL	Ângulo inicial ou 1ª posição do eixo circular			(em XY) (em XA, YB, ZC)
5	α1	<INDA>	REAL	Ângulo de indexação (somente para círculo parcial)			(emi XY, XA, YB, ZC)
					< 0 =	Sentido horário	
					> 0 =	Sentido anti-horário	
6	N	<NUM>	INT	Número de posições			

Nº.	Tela de parâmetros	Parâmetros interno	Tipo de dado	Significado			
7		<_VARI>	INT	Tipo de usinagem			
				UNIDADE:	reservado		
				DEZENA:	Tipo de posicionamento		
					0 =	Aproximação da posição - linear	
					1 =	Aproximação da posição - em percurso circular	
				CENTENA:	reservado		
				MILHAR:	Modelo de círculo		
					0 =	Modo de compatibilidade, se INDA = 0, então temos o círculo inteiro, e INDA <> 0, então será um círculo parcial)	
					1 =	Círculo inteiro	
					2 =	Círculo parcial	
				DEZENA DA MILHAR:	Modelo de posição com eixo circular		
					0 =	XY (sem eixo rotativo)	(em XY)
					1 =	XA (eixo X e eixo rotativo em torno de X)	(apenas com XA)
					2 =	YB (eixo Y e eixo rotativo em torno de Y)	(apenas comi YB)
					3 =	ZC (eixo Z e eixo rotativo em torno de C)	(apenas com ZC)
				UM MILHÃO+ CENTENA DE MILHAR:	Offset (com mais eixos circulares em torno de um mesmo eixo; se o índice for grande demais, então 1º eixo)		
					00 =	1. Eixo A, B ou C	
					01 =	2. Eixo A, B ou C	
					...		
					10 =	20. Eixo A, B ou C	
8		<_UMODE>	INT	reservado			
9		<_HIDE>	STRING [200]	reservado			
10		<_NSP>	INT	reservado			
11		<_DMODE>	INT	Modo de Exibição			
				UNIDADE:	Plano de usinagem G17/18/19		
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo	
					1 =	G17 (ativo apenas no ciclo)	
					2 =	G18 (ativo apenas no ciclo)	
					3 =	G19 (ativo apenas no ciclo)	

20.1.5 POCKET3 - Fresamento de bolsão retangular

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)		
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Profundidade da bolsa (abs/inc), consulte <_AMODE>)		
5	L	<_LENG>	REAL	Comprimento do bolsão (inc, especificar com sinal)		
6	W	<_WID>	REAL	Largura do bolsão (inc, especificar com sinal)		
7	R	<_CRAD>	REAL	Raio de canto do bolsão		
8	X0	<_PA>	REAL	Ponto de referência, 1º Eixo (abs)		
9	YO	<_PO>	REAL	Ponto de referência, 2º eixo (abs)		
10	α0	<_STA>	REAL	Ângulo de giro, ângulo entre o eixo longitudinal (L) e o 1º Eixo		
11	DZ	<_MID>	REAL	penetração máxima em profundidade		
12	UXY	<_FAL>	REAL	Sobremetal de acabamento do plano		
13	UZ	<_FALD>	REAL	Sobremetal de acabamento da profundidade		
14	F	<_FFP1>	REAL	Avanço no plano		
15	FZ	<_FFD>	REAL	Valor do avanço de penetração em profundidade		
16		<_CDIR>	INT	Sentido de fresamento:	0 =	Concordante
					1 =	Discordante
17		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:		
					1 =	Desbaste
					2 =	Acabamento
					4 =	Acabamento da borda
					5 =	Chanframento
				DEZENA:		
					0 =	pré-furado, penetração com G0
					1 =	perpendicular, penetração com G1
					2 =	helicoidal
					3 =	alternado no eixo longitudinal do bolsão
				CENTENA:	reservado	

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
18	DXY	<_MIDA>	REAL	penetração máxima no plano, consulte a unidade em <_AMODE>		
19	L1	<_AP1>	REAL	Comprimento da pré-usinagem (inc)		
20	W1	<_AP2>	REAL	Largura da pré-usinagem (inc)		
21	AZ	<_AD>	REAL	Profundidade da pré-usinagem (inc)		
22	ER	<_RAD1>	REAL	Raio da trajetória helicoidal na imersão helicoidal		
	EW			ângulo máximo de imersão para movimento alternado		
23	EP	<_DP1>	REAL	Passo da espiral na imersão helicoidal		
24		<_UMODE>	INT	reservado		
25	FS	<_FS>	REAL	Largura do chanfro (inc)		
26	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>		
27		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem/apenas cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal
				MILHAR:	Dimensionamento através do centro/canto	
					0 =	Modo de compatibilidade
					1 =	Dimensionamento através do centro
					2 =	Dimensionamento pelo canto, posição do bolsão +LENG/+WID
					3 =	Dimensionamento pelo canto, posição do bolsão -LENG/+WID
					4 =	Dimensionamento pelo canto, posição do bolsão +LENG/-WID
					5 =	Dimensionamento pelo canto, posição do bolsão -LENG/-WID
				DEZENA DA MILHAR:	Usinagem completa/retrabalho (reusinagem)	
					0 =	Modo de compatibilidade (<_AP1>, <_AP2> e <_AD> tratar como até então)
					1 =	Usinagem completa
					2 =	Retrabalho

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
28		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Tipo de avanço: Grupo G (G94/G95) para avanço em planos e em profundidade	
					0 =	Modo de compatibilidade
					1 =	Comando G como antes da chamada do ciclo. G94/G95 igual para avanço em planos e em profundidade
				CENTENA:	---	reservado
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples
29		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade do bolsão (Z1)	
					0 =	Absoluto (Modo de compatibilidade)
					1 =	incremental
				DEZENA:	Unidade para penetração no plano (DXY)	
					0 =	mm
					1 =	% do diâmetro da ferramenta
				CENTENA:	Profundidade de imersão no chanframento (ZFS)	
					0 =	Absoluto
					1 =	incremental

20.1.6 POCKET4 - Fresamento de bolsão circular

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)		
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Profundidade da bolsa (abs/inc), consulte <_AMODE>)		
5	∅	<_CDIAM>	REAL	Diâmetro ou raio do bolsão, consulte <_DMODE>		
6	X0	<_PA>	REAL	Ponto de referência, 1º eixo (abs)		
7	Y0	<_PO>	REAL	Ponto de referência, 2º eixo (abs)		
8	DZ	<_MID>	REAL	penetração máxima em profundidade, consulte<_VARI> = por planos passo máximo de espiral, consulte <_VARI> = helicoidal		
9	UXY	<_FAL>	REAL	Sobremetal de acabamento do plano		
10	UZ	<_FALD>	REAL	Sobremetal de acabamento da profundidade		
11	F	<_FFP1>	REAL	Avanço para usinagem de superfícies		
12	FZ	<_FFD>	REAL	Valor do avanço de penetração em profundidade		
13		<_CDIR>	INT	Sentido de fresamento	0 =	Concordante
					1 =	Discordante
14		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Usinagem	
					1 =	Desbaste
					2 =	Acabamento
					4 =	Acabamento da borda
					5 =	Chanframento
				DEZENA:	Tipo de penetração (desbaste e acabamento)	
					0 =	pré-furado, penetração com G0 (o bolsão foi pré-usinado)
					1 =	perpendicular, penetração com G1
					2 =	helicoidal
				CENTENA:	reservado	
				MILHAR:		
					0 =	em planos
					1 =	helicoidal
15	DXY	<_MIDA>	REAL	penetração máxima no plano, consulte <_AMODE>, 0 = 0,8 x diâmetro da ferramenta		
16	∅	<_AP1>	REAL	Diâmetro/raio da pré-usinagem (inc)		
17	AZ	<_AD>	REAL	Profundidade da pré-usinagem (inc)		
18	ER	<_RAD1>	REAL	Raio da trajetória helicoidal na imersão helicoidal		
19	EP	<_DP1>	REAL	Passo da espiral na imersão na trajetória helicoidal		
20		<_UMODE>	INT	reservado		
21	FS	<_FS>	REAL	Largura do chanfro (inc)		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
22	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>		
23		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Usinagem/cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal
				MILHAR:	reservado	
				DEZENA DA MILHAR:	Usinagem completa/retrabalho (reusinagem)	
					0 =	Modo de compatibilidade (<_AP1> e <_AD> tratar como até então)
					1 =	Usinagem completa
					2 =	Retrabalho
24		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Tipo de avanço: Grupo G (G94/G95) para avanço em planos e em profundidade	
					0 =	Modo de compatibilidade
					1 =	Comando G como antes da chamada do ciclo. G94/G95 igual para avanço em planos e em profundidade
				CENTENA:		
					0 =	Modo de compatibilidade (<_CDIAM>/<_AP1> especificado como raio)
					1 =	<_CDIAM>/<_AP1> especificado como diâmetro
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
25		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade do bolsão (Z1)	
					0 =	Absoluto (Modo de compatibilidade)
					1 =	incremental
				DEZENA:	Unidade para largura de penetração (DXY)	
					0 =	mm
					1 =	% do diâmetro da ferramenta
				CENTENA:	Profundidade de imersão no chanframento (ZFS)	
					0 =	Absoluto
					1 =	incremental

20.1.7 SLOT1 - Ranhura longitudinal

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	RP	<RTP>	REAL	Plano de retrocesso (abs)
2	Z0	<RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)
4	Z1	<_DP>	REAL	Profundidade da ranhura (abs)
5		<_DPR>	REAL	Profundidade da ranhura (ink), relativa ao Z0 (especificar sem sinal)
6		<NUM>	INT	Número de ranhuras = 1
7	L	<LENG>	REAL	Comprimento da ranhura
8	W	<WID>	REAL	Largura da ranhura
9	X0	<_CPA>	REAL	Ponto de referência, 1º eixo do plano
10	Y0	<_CPO>	REAL	Ponto de referência, 2º eixo do plano
11		<RAD>	REAL	reservado
12	α	<STA1>	REAL	Ângulo de giro
13		<INDA>	REAL	reservado
14	FZ	<FFD>	REAL	Valor do avanço de penetração em profundidade

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
15	F	<FFP1>	REAL	Avanço		
16	DZ	<_MID>	REAL	penetração máxima em profundidade		
17		<CDIR>	INT	Sentido de fresamento	0 =	Concordante
					1 =	Discordante
18	UXY	<_FAL>	REAL	Sobremetal de acabamento no plano ou borda da ranhura		
19		<VARI>	INT	Tipo de usinagem		
				UNIDADE:		
					0 =	reservado
					1 =	Desbaste
					2 =	Acabamento
					4 =	Acabamento da borda (usinar apenas a borda)
					5 =	Chanframento
				DEZENA:	Aproximação	
					0 =	pré-furado, penetração com G0 (a ranhura foi pré-usinada)
					1 =	perpendicular, penetração com G1
					2 =	helicoidal
					3 =	oscilante
				CENTENA:	reservado	
20	DZF	<_MIDF>	REAL	reservado		
21	FF	<FFP2>	REAL	reservado		
22	SF	<SSF>	REAL	reservado		
23	UZ	<_FALD>	REAL	Sobremetal de acabamento da profundidade		
24	ER	<_STA2>	REAL	Raio da trajetória helicoidal na imersão helicoidal		
	EW			ângulo máximo de imersão para movimento alternado		
25	EP	<_DP1>	REAL	Profundidade de imersão por rotação para espiral		
26		<_UMODE>	INT	reservado		
27	FS	<_FS>	REAL	Largura de chanfro (inc) no chanframento		
28	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
29		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem ou apenas cálculo do ponto de partida	
					1 =	Usinagem normal
				MILHAR:	Dimensionamento do ponto de referência, posição da ranhura	
					0 =	Centro
					1 =	interno à esquerda +L
					2 =	interno à direita -L
					3 =	borda esquerda +L
					4 =	borda direita -L
30		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	Identificação da versão de SW	
					1 =	Ampliação da função SLOT1
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples
31		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final Z1 (abs/inc)	
					0 =	Compatibilidade
					1 =	Z1:
					2 =	Z1 (abs)
				DEZENA:	reservado	
				CENTENA:	Profundidade de imersão no chanframento ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (inc)

Indicação

O ciclo foi aprimorado com novas funções, se comparado com versões anteriores de SW. Isto tem como resultado que, determinados parâmetros na tela de especificação não sejam mais exibidos (<NUM>, <RAD>, <INDA>). As ranhuras múltiplas sobre um modelo de posição são programáveis através do "MCALL" e chamada do modelo de posição desejado, p. ex. HOLES2.

20.1.8 SLOT2 - Ranhura circular

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)		
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Profundidade da ranhura (abs)		
5		<_DPR>	REAL	Profundidade da ranhura (ink), relativa ao Z0 (especificar sem sinal)		
6	N	<NUM>	INT	Número de ranhuras		
7	α1	<AFSL>	REAL	Ângulo de abertura da ranhura		
8	W	<WID>	REAL	Largura da ranhura		
9	X0	<_CPA>	REAL	Ponto de referência = centro do círculo, 1º eixo do plano		
10	Y0	<_CPO>	REAL	Ponto de referência = centro do círculo, 2º eixo do plano		
11	R	<RAD>	REAL	Raio do círculo		
12	α0	<STA1>	REAL	Ângulo inicial		
13	α2	<INDA>	REAL	Ângulo de indexação		
14	FZ	<FFD>	REAL	Valor do avanço de penetração em profundidade		
15	F	<FFP1>	REAL	Avanço		
16	DZ	<_MID>	REAL	penetração máxima em profundidade		
17		<CDIR>	INT	Sentido de fresamento	0 =	Concordante
					1 =	Discordante
18	UXY	<_FAL>	REAL	Sobremetal de acabamento no plano ou borda da ranhura		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
19		<VARI>	INT	Tipo de usinagem		
				UNIDADE:		
					0 =	Usinagem completa
					1 =	Desbaste
					2 =	Acabamento
					3 =	Acabamento da borda
					5 =	Chanframento
				DEZENA:		
					0 =	posicionamento intermediário com reta G0
					1 =	posicionamento intermediário em percurso circular
				CENTENA:	reservado	
				MILHAR:		
					0 =	Modo de compatibilidade, quando <INDA> = 0 então circulo interior,<INDA> **<>** 0 então círculo parcial)
					1 =	Círculo inteiro
					2 =	Círculo parcial
20	DZF	<_MIDF>	REAL	reservado		
21		<FFP2>	REAL	reservado		
22		<SSF>	REAL	reservado		
23	FF	<_FFCP>	REAL	reservado		
24		<_UMODE>	INT	reservado		
25	FS	<_FS>	REAL	Largura do chanfro (inc)		
26	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>		
27		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem ou apenas cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
28		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	Identificação da versão de SW	
					1 =	Funções SLOT2 a partir de SW 2.5
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples
29		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final Z1 (abs/inc)	
					0 =	Compatibilidade
					1 =	Z1:
					2 =	Z1 (abs)
				DEZENA:	reservado	
				CENTENA:	Profundidade de imersão no chanframento ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (inc)

20.1.9 LONGHOLE - Furo oblongo

Sintaxe

```
LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>,
<_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>,
<_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	RP	<RTP>	REAL	Plano de retrocesso (abs)
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Profundidade do oblongo (abs)		
5		<_DPR>	REAL	Profundidade o oblongo (ink), relativa ao Z0 (especificar sem sinal)		
6		<NUM>	INT	Quantidade de oblongos = 1		
7	L	<LENG>	REAL	Comprimento do oblongo		
8	X0	<_CPA>	REAL	Ponto de referência, 1º eixo do plano		
9	Y0	<_CPO>	REAL	Ponto de referência, 2º eixo do plano		
10		<RAD>	REAL	reservado		
11	α0	<STA1>	REAL	Ângulo de giro		
12		<INDA>	REAL	reservado		
13	FZ	<FFD>	REAL	Valor do avanço de penetração em profundidade		
14	F	<FFP1>	REAL	Avanço		
15	DZ	<MID>	REAL	penetração máxima em profundidade		
16		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Tipo de penetração	
					1 =	perpendicular com G1
					3 =	oscilante
				CENTENA:	reservado	
17		<_UMODE>	INT	reservado		
18		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem ou apenas cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal
				MILHAR:	Dimensionamento do ponto de referência, posição da ranhura	
					0 =	Centro
					1 =	interno à esquerda +L
					2 =	interno à direita -L
					3 =	borda esquerda +L
					4 =	borda direita -L

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
19		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Tipo de avanço: Grupo G (G94/G95) para avanço em planos e em profundidade	
					0 =	Modo de compatibilidade
					1 =	Comando G como antes da chamada do ciclo. G94/G95 igual para avanço em planos e em profundidade
				CENTENA:	reservado	
				MILHAR:	Identificação da versão de SW	
					1 =	Extensão das funções do LONGHOLE (dimensionamento do ponto de referência)
20		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final Z1 (abs/inc)	
					0 =	Compatibilidade
					1 =	Z1:
					2 =	Z1 (abs)

Indicação

O ciclo foi aprimorado com novas funções, se comparado com versões anteriores de SW. Isto tem como resultado que, determinados parâmetros na tela de especificação não sejam mais exibidos (<NUM>, <RAD>, <INDA>). As ranhuras múltiplas sobre um modelo de posição são programáveis através do "MCALL" e chamada do modelo de posição desejado, p. ex. HOLES2.

20.1.10 CYCLE60 - Ciclo de gravação

Sintaxe

```
CYCLE60(<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parâmetros

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1		<_TEXT>	STRING [200]	texto de gravação (máximo 100 caracteres)
2	RP	<_RTP>	REAL	Plano de retrocesso (abs)
3	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)
4	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao plano de referência, especificar sem sinal)
5	Z1	<_DP>	REAL	Profundidade (abs), consulte <_AMODE>
6	Z1	<_DPR>	REAL	Profundidade (inc), consulte <_AMODE>
7	X0	<_PA>	REAL	Ponto de referência, 1º eixo do plano (abs) - ortogonal, consulte <_VARI>
	R			Ponto de referência, comprimento (raio) - polar, consulte <_VARI>
8	Y0	<_PO>	REAL	Ponto de referência, 2º eixo do plano (abs) - ortogonal, consulte <_VARI>
	α0			Ponto de referência, ângulo relativo ao 1º eixo - polar, consulte <_VARI>
9	α1	<_STA>	REAL	Direção do texto, ângulo da linha de texto relativo ao 1º eixo), consulte <_VARI>
10	XM	<_CP1>	REAL	Ponto central do círculo de texto, 1º eixo do plano (abs) - ortogonal, consulte <_VARI>
	LM			Centro do círculo de texto, comprimento (raio) relativo ao WNP - polar, consulte <_VARI>
11	YM	<_CP2>	REAL	Ponto central do círculo de texto, 2º eixo do plano (abs) - ortogonal, consulte <_VARI>
	αM			Ponto central do círculo de texto, ângulo relativo ao 1º eixo - polar, consulte <_VARI>
12	W	<_WID>	REAL	Altura dos caracteres (especificar sem sinal)
13	DX1 DX2	<_DF>	REAL	Espaçamento entre caracteres / largura total, consulte <_VARI>
	α2			Ângulo de abertura, consulte <_VARI>
14	FZ	<_FFD>	REAL	Avanço de penetração em profundidade, consulte <_DMODE>
15	F	<_FFP1>	REAL	Avanço para usinagem de superfícies

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
16		<_VARI>	INT	Usinagem (alinhamento e ponto de referência do texto de gravação)		
				UNIDADE:	Ponto de referência	
					0 =	ortogonal
					1 =	polar
				DEZENA:	Orientação do texto	
					0 =	Texto sobre uma linha
					1 =	Texto sobre um arco para cima
					2 =	Texto sobre um arco para baixo
				CENTENA:	reservado	
				MILHAR:	Ponto de referência do texto horizontal	
					0 =	esquerda
					1 =	centralizado
					2 =	direita
				DEZENA DA MILHAR:	Ponto de referência do texto vertical	
					0 =	embaixo
					1 =	centralizado
					2 =	em cima
				CENTENA DA MILHAR:	Comprimento do texto	
					0 =	Espaçamento de caracteres
					1 =	Largura total do texto (apenas para texto linear)
					2 =	Ângulo de abertura (apenas para texto sobre arco)
				UNIDADE DE MILHÃO:	Centro do círculo	
					0 =	Ortogonal (cartesiano)
					1 =	polar
				DEZENA DE MILHÃO:	Texto espelhado	
					0 =	Compatibilidade
					1 =	Texto espelhado LIGA
					2 =	Texto espelhado DESLIGA
17		<_CODEP>	INT	Número do Codepage para a escrita (por enquanto apenas o 1252)		
18		<_UMODE>	INT	reservado		
19		_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem/apenas cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
20		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Tipo de avanço: Grupo G (G94/G95) para avanço em planos e em profundidade	
					0 =	Modo de compatibilidade
					1 =	Comando G como antes da chamada do ciclo. G94/G95 igual para avanço em planos e em profundidade
21		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final (<_DP>, <_DPR>)	
					0 =	Compatibilidade
					1 =	incremental (<_DPR>)
					2 =	absoluto (<_DP>)

20.1.11 CYCLE61- Fresamento de facear

Sintaxe

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta, altura da peça bruta (abs)
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)
4	Z1	<_DP>	REAL	Altura da peça pronta (abs/inc), consulte <_AMODE>
5	X0	<_PA>	REAL	Canto 1 no 1º eixo (abs)
6	Y0	<_PO>	REAL	Canto 1 no 2º eixo (abs)
7	X1	<_LENG>	REAL	Canto 2 no 1º eixo (abs/inc), consulte <_AMODE>
8	Y1	<_WID>	REAL	Canto 2 no 2º eixo (abs/inc), consulte <_AMODE>
9	DZ	<_MID>	REAL	penetração máxima em profundidade

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
10	DXY	<_MIDA>	REAL	penetração máxima no plano (unidade, consulte a unidade em <_AMODE>)		
11	UZ	<_FALD>	REAL	Sobremetal de acabamento da profundidade		
12	F	<_FFP1>	REAL	Avanço de usinagem		
13		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Usinagem	
					1 =	Desbaste
					2 =	Acabamento
				DEZENA:	Sentido de usinagem	
					1 =	paralelo com o 1º eixo, um sentido
					2 =	paralelo com o 2º eixo, um sentido
					3 =	paralelo com o 1º eixo, sentido alternado
					4 =	paralelo com o 2º eixo, sentido alternado
14		<_LIM>	INT	Delimitações		
				UNIDADE:	Delimitação do 1º eixo negativo	
					0 =	não
					1 =	sim
				DEZENA:	Delimitação do 1º eixo positivo	
					0 =	não
					1 =	sim
				CENTENA:	Delimitação do 2º eixo negativo	
					0 =	não
					1 =	sim
				MILHAR:	Delimitação do 2º eixo positivo	
					0 =	não
					1 =	sim
15		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
16		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final (<_DP>)	
					0 =	Absoluto
					1 =	incremental
				DEZENA:	Unidade para penetração no plano (<_MIDA>)	
					0 =	mm
					1 =	% do diâmetro da ferramenta
				CENTENA:	reservado	
				MILHAR:	Comprimento da superfície	
					0 =	incremental
					1 =	Absoluto
				DEZENA DA MILHAR:	Largura da superfície	
					0 =	incremental
					1 =	Absoluto

20.1.12 CYCLE62 - Chamada de contorno

Sintaxe

```
CYCLE62(<_KNAME>,<_TYPE>, <_LAB1>, <_LAB2>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	PRG/CON	<_KNAME>	STRING [140]	Nome de contorno ou nome de subrotina, mas não precisa ser programado no _TYPE = 2		
2		<_TYPE>	INT	Determinação da especificação do contorno		
					0 =	Subrotina
					1 =	Nome do contorno
					2 =	Labels
					3 =	Labels na subrotina
3	LAB1	<_LAB1>	STRING[32]	Label 1, início do contorno		
4	LAB2	<_LAB2>	STRING[32]	Label 2, fim do contorno		

20.1.13 CYCLE63 - Fresar bolsão de contorno

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	PRG	<_PRG>	STRING [100]	Nome do programa de remoção de material		
2		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Tecnologias de usinagem	
					1 =	Desbaste
					3 =	Acabamento da base
					4 =	Acabamento da borda
					5 =	Chanframento
				DEZENA:	Tipo de penetração	
					0 =	Imersão centralizada
					1 =	Imersão helicoidal
					2 =	Imersão oscilante
				CENTENA:	reservado	
				MILHAR:	Modo de retração	
					0 =	Retração até o plano de retrocesso
					1 =	retração até o ponto de referência + distância de segurança
				DEZENA DA MILHAR:	Ponto de partida no desbaste e acabamento da base	
					0 =	auto
					1 =	manual
3	RP	<_RP>	REAL	Plano de retrocesso (abs)		
4	Z0	<_Z0>	REAL	Ponto de referência do eixo da ferramenta (abs)		
5	SC	<_SC>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
6	Z1	<_Z1>	REAL	Profundidade final, (consulte <_AMODE> UNIDADE)		
7	F	<_F>	REAL	Avanço no plano do desbaste/acabamento		
8	FZ	<_FZ>	REAL	Valor do avanço de penetração em profundidade		
9	DXY	<_DXY>	REAL	Penetração no plano - unidade, (consulte <_AMODE> DEZENA)		
10	DZ	<_DZ>	REAL	Avanço em profundidade		
11	UXY	<_UXY>	REAL	Sobremetal de acabamento do plano		
12	UZ	<_UZ>	REAL	Sobremetal de acabamento da profundidade		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
13		<_CDIR>	INT	Sentido de fresamento	0 =	Concordante
					1 =	Discordante
14	XS	<_XS>	REAL	Ponto de partida X, absoluto		
15	YS	<_YS>	REAL	Ponto de partida Y, absoluto		
16	ER	<_ER>	REAL	Imersão helicoidal: Raio		
17	EP	<_EP>	REAL	Imersão helicoidal: Passo		
18	EW	<_EW>	REAL	Imersão alternada: ângulo máximo de imersão		
19	FS	<_FS>	REAL	Largura de chanfro (inc) no chanframento		
20	ZFS	<_ZFS>	REAL	Profundidade de imersão da ponta da ferramenta no chanframento, (consulte<_AMODE> CENTENA)		
21	TR	<_TR>	STRING[32]	Nome da ferramenta de referência na usinagem do material residual		
22	DR	<_DR>	INT	Número D da ferramenta de referência na usinagem do material residual		
23		<_UMODE>	INT	reservado		
24		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem/apenas cálculo do ponto de partida	
					0 =	usinagem normal (nenhum modo de compatibilidade necessário)
					1 =	Usinagem normal
					2 =	reservado
25		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	reservado	
				CENTENA:	Modo de tecnologia	
					1 =	Bolsão
					2 =	Saliência
				MILHAR:	Usinagem do material residual	
					0 =	não
					1 =	sim
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
26		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final (Z1)	
					0 =	Absoluto (Modo de compatibilidade)
					1 =	incremental
				DEZENA:	Unidade para penetração no plano (DXY)	
					0 =	mm
					1 =	% do diâmetro da ferramenta
				CENTENA:	Profundidade de imersão no chanfra-mento (ZFS)	
					0 =	Absoluto
					1 =	incremental
				MILHAR:	---	reservado

20.1.14 CYCLE64 -Pré-furação do bolsão de contorno

Sintaxe

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_DXY>,
<_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	PRG	<_PRG>	STRING [100]	Nome do programa de furação e centragem		
2		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	Modo de retração	
					0 =	Retração até o plano de retrocesso
					1 =	retração até o ponto de referência + distância de segurança
3	RP	<_RP>	REAL	Plano de retrocesso (abs)		
4	Z0	<_Z0>	REAL	Ponto de referência (abs)		
5	SC	<_SC>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
6	Z1	<_Z1>	REAL	Profundidade de furação/centragem, (consulte<_AMODE> UNIDADE)		
7	F	<_F>	REAL	Avanço de furação/centragem		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
8	DXY	<_DXY>	REAL	Penetração no plano - unidade, (consulte<_AMODE> DEZENA)		
9	UXY	<_UXY>	REAL	Sobremetal de acabamento do plano		
10	UZ	<_UZ>	REAL	Sobremetal de acabamento da profundidade		
11		<_CDIR>	INT	Sentido de fresamento	0 =	Concordante
					1 =	Discordante
12	TR	<_TR>	STRING[20]	Nome da ferramenta de referência		
13	DR	<_DR>	INT	Número D da ferramenta de referência		
14		<_UMODE>	INT	reservado		
15		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem/apenas cálculo do ponto de partida	
					0 =	usinagem normal (nenhum modo de compatibilidade necessário)
					1 =	Usinagem normal
					2 =	reservado
25		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Modo de tecnologia	
					1 =	Pré-furação
					2 =	Centragem
26		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade de furação/centragem Z1	
					0 =	Absoluto (Modo de compatibilidade)
					1 =	incremental
				DEZENA:	Unidade para penetração no plano (DXY)	
					0 =	mm
					1 =	% do diâmetro da ferramenta

20.1.15 CYCLE70 - Fresamento de roscas

Sintaxe

```
CYCLE70(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)		
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Comprimento da rosca (abs, inc), consulte <_AMODE> Considerar a saída na base do furo (pelo menos meio passo)		
5	∅	<_DIATH>	REAL	Diâmetro nominal da rosca		
6	H1	<_H1>	REAL	Profundidade da rosca		
7	U	<_FAL>	REAL	Sobremetal de acabamento		
8	P	<_PIT>	REAL	Passo de rosca (seleção <_PITA>: mm, inch, MODUL, Cursos/polegada)		
9	NT	<_NT>	INT	Número de dentes na pastilha de corte O comprimento da ferramenta sempre é referenciado no dente inferior!		
10	DXY	<_MID>	REAL	Penetração máxima por corte <_MID> > <_H1>: tudo com um corte		
11	F	<_FFR>	REAL	Avanço de fresamento		
12		<_TYPTH>	INT	Tipo de rosca	0 =	Rosca interna
					1 =	Rosca externa
13	X0	<_PA>	REAL	Centro do círculo, 1º eixo (abs)		
14	Y0	<_PO>	REAL	Centro do círculo, 2º eixo (abs)		
15	αS	<_NSP>	REAL	Ângulo de partida (rosca de múltiplas entradas)		
16		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:		
					1 =	Desbaste
					2 =	Acabamento
				DEZENA:		
					1 =	de cima para baixo
					2 =	de baixo para cima
				CENTENA:		
					0 =	Rosca à direita
					1 =	Rosca à esquerda

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
17		<_PITA>	INT	Avaliação do passo da rosca		
					0 =	Modo de compatibilidade
					1 =	Passo em mm
					2 =	Passo em voltas por polegada (TPI)
					3 =	Passo em polegadas
					4 =	Passo como MÓDULO
18		<_PITM>	STRING[15]	String como marcador para entrada do passo de rosca (apenas para a interface)		
19		<_PTAB>	STRING[20]	String para tabela de roscas ("", "ISO", "BSW", "BSP", "UNC") (apenas para a interface)		
20		<_PTABA>	STRING[20]	String para a seleção da tabela de roscas (por ex., "M 10", "M 12", ...) (apenas para a superfície)		
21		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Usinagem/cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal
22		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
23		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Comprimento da rosca (<_DP>)	
					0 =	Absoluto
					1 =	incremental

20.1.16 CYCLE72 - Fresamento de percurso

Sintaxe

```
CYCLE72(<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>,<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1		<_KNAME>	STRING [141]	Nome da subrotina do contorno
2	RP	<_RTP>	REAL	Plano de retrocesso (abs)
3	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)
4	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)
5	Z1	<_DP>	REAL	Ponto final, profundidade final (abs/inc), consulte <_AMODE>
6	DZ	<_MID>	REAL	Profundidade máxima de penetração (inc, sem especificar o sinal)
7	UXY	<_FAL>	REAL	Sobremetal de acabamento no plano (inc), sobremetal no contorno da borda
8	UZ	<_FALD>	REAL	Sobremetal de acabamento na profundidade (inc), sobremetal na base (especificar sem sinal)
9	FX	<_FFP1>	REAL	Avanço no contorno
10	FZ	<_FFD>	REAL	Avanço para penetração em profundidade (sem penetração tridimensional)

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
11		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Usinagem	
					1 =	Desbaste
					2 =	Acabamento
					5 =	Chanframento
				DEZENA:		
					0 =	Cursos intermediários com G0
					1 =	Cursos intermediários com G1
				CENTENA:	Retrocesso no final do contorno	
					0 =	Retrocesso no final do contorno ao ponto de referência
					1 =	Retrocesso no final do contorno ao ponto de referência + <_SDIS>
					2 =	Retrocesso no final do contorno em <_SDIS>
					3 =	Sem retrocesso no final do contorno, o próximo ponto de partida é aproximado com avanço de contorno
				MILHAR:	reservado	
				DEZENA DA MILHAR:	Usinar o contorno	
					0 =	Usinar o contorno avançando
					1 =	Usinar o contorno retrocedendo Restrições no movimento para trás: • máx. 170 elementos de contorno (inclusive chanfros ou arredondamentos) • somente são interpretados os valores no plano (X/Y) e F
12		<_RL>	INT	Sentido de usinagem		
					40 =	Centralizado no contorno (G40, aproximação e afastamento: reta ou vertical)
					41 =	À esquerda do contorno (G41, aproximação e afastamento: reta ou círculo)
					42 =	À direita do contorno (G42, aproximação e afastamento: reta ou círculo)

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
13		<_AS1>	INT	Movimento de aproximação do contorno		
				UNIDADE:		
					1 =	Reta
					2 =	Quadrante:
					3 =	semicírculo
					4 =	Aproximação e afastamento vertical
				DEZENA:		
					0 =	Último movimento, no plano
					1 =	Último movimento, no espaço
14	L1	<_LP1>	REAL	Trajetória de aproximação ou raio de aproximação (inc, sem especificar o sinal)		
15	FZ	<_FF3>	REAL	Avanço para percursos intermediários (G94/G95 como no contorno)		
16		<_AS2>	INT	Movimento de afastamento do contorno (não na aproximação e afastamento na vertical)		
				UNIDADE:		
					1 =	Reta
					2 =	Quadrante:
					3 =	semicírculo
				DEZENA:		
					0 =	Último movimento, no plano
					1 =	Último movimento, no espaço
17	L2	<_LP2>	REAL	Trajetória de afastamento ou raio de afastamento (inc, sem especificar o sinal)		
18		<_UMODE>	INT	reservado		
19	FS	<_FS>	REAL	Largura do chanfro (inc)		
20	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>		
21		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem/apenas cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
22		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Tipo de avanço: Grupo G (G94/G95) para avanço em planos e em profundidade	
					0 =	Modo de compatibilidade
					1 =	Comando G como antes da chamada do ciclo. G94/G95 igual para avanço em planos e em profundidade
				MILHAR:		
					0 =	Modo de compatibilidade: O nome do contorno está em <_KNAME>
					1 =	O nome do contorno é programado em CYCLE62 e transferido para _SC_CONT_NAME
23		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Ponto final Z1 (<_DP>)	
					0 =	Absoluto (Modo de compatibilidade)
					1 =	incremental
				DEZENA:	Unidade para penetração no plano	
					0 =	mm, pol.
					1 =	reservado
				CENTENA:	Profundidade de imersão no chanframento (<_ZFS>)	
					0 =	Absoluto
					1 =	incremental

Indicação

Se os seguintes parâmetros de transferência forem programados indiretamente (como parâmetros), a tela de especificação não será recompilada:

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

20.1.17 CYCLE76 - Fresar a saliência retangular

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)		
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Profundidade da saliência (abs)		
5		<_DPR>	REAL	Profundidade da saliência (ink), relativa ao Z0 (especificar sem sinal)		
6	L	<_LENG>	REAL	Comprimento da saliência, consulte<_GMODE> (especificar sem sinal prévio)		
7	W	<_WID>	REAL	Largura da saliência, consulte<_GMODE> (especificar sem sinal prévio)		
8	R	<_CRAD>	REAL	Raio de canto da saliência (especificar sem sinal)		
9	X0	<_PA>	REAL	Ponto de referência da saliência, 1º eixo do plano (abs)		
10	Y0	<_PO>	REAL	Ponto de referência da saliência, 2º eixo do plano (abs)		
11	α0	<_STA>	REAL	Ângulo de giro, ângulo entre o eixo longitudinal (L) e o 1º eixo do plano		
12	DZ	<_MID>	REAL	Profundidade máxima de penetração (inc, sem especificar o sinal)		
13	UXY	<_FAL>	REAL	Sobremetal de acabamento no plano (inc), sobremetal no contorno da borda		
14	UZ	<_FALD>	REAL	Sobremetal de acabamento na profundidade (inc), sobremetal na base (especificar sem sinal)		
15	FX	<_FFP1>	REAL	Avanço no contorno		
16	FZ	<_FFD>	REAL	Valor do avanço de penetração em profundidade		
17		<_CDIR>	INT	Sentido de fresamento (especificar sem sinal)		
				UNIDADE:		
					0 =	Concordante
					1 =	Discordante
18		<_VARI>	INT	Usinagem		
				UNIDADE:		
					1 =	Desbaste
					2 =	Acabamento
					5 =	Chanframento
19	L1	<_AP1>	REAL	Comprimento da saliência bruta		
20	W1	<_AP2>	REAL	Largura da saliência bruta		
21	FS	<_FS>	REAL	Largura do chanfro (inc)		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
22	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>		
23		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem ou apenas cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal
				MILHAR:	Dimensionamento da saliência através do centro ou canto	
					0 =	Modo de compatibilidade
					1 =	Dimensionamento através do centro
					2 =	Dimensionamento do canto, saliência +L +W
					3 =	Dimensionamento do canto, saliência -L +W
					4 =	Dimensionamento do canto, saliência +L -W
					5 =	Dimensionamento do canto, saliência -L -W
				DEZENA DA MILHAR:	Usinagem completa ou retrabalho (reusinagem)	
					0 =	Modo de compatibilidade
					1 =	Usinagem completa
					2 =	Retrabalho
24		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	---	reservado
				CENTENA:	---	reservado
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
25		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final Z1 (DP)	
					0 =	Compatibilidade
					1 =	incremental
					2 =	Absoluto
				DEZENA:	reservado	
				CENTENA:	Profundidade de imersão no chanframento (ZFS)	
					0 =	Absoluto
					1 =	incremental

20.1.18 CYCLE77 - Fresar a saliência circular

Sintaxe

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>,
<_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)
4	Z1	<_DP>	REAL	Profundidade da saliência (abs)
5		<_DPR>	REAL	Profundidade da saliência (ink), relativa ao Z0 (especificar sem sinal)
6	∅	<_CDIAM>	REAL	Diâmetro da saliência (especificar sem sinal)
7	X0	<_PA>	REAL	Ponto de referência da saliência, 1º eixo do plano (abs)
8	Y0	<_PO>	REAL	Ponto de referência da saliência, 2º eixo do plano (abs)
9	DZ	<_MID>	REAL	Profundidade máxima de penetração (inc, sem especificar o sinal)
10	UXY	<_FAL>	REAL	Sobremetal de acabamento no plano (inc), sobremetal no contorno da borda
11	UZ	<_FALD>	REAL	Sobremetal de acabamento na profundidade (inc), sobremetal na base (especificar sem sinal)
12	FX	<_FFP1>	REAL	Avanço no contorno
13	FZ	<_FFD>	REAL	Valor do avanço de penetração em profundidade

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
14		<_CDIR>	INT	Sentido de fresamento (especificar sem sinal)		
				UNIDADE:		
					0 =	Concordante
					1 =	Discordante
15		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Usinagem	
					1 =	Desbaste até o sobremetal de acabamento
					2 =	Acabamento (sobremetal em X/Y/Z = 0)
					5 =	Chanframento
16	∅1	<_AP1>	REAL	Diâmetro da saliência bruta		
17	FS	<_FS>	REAL	Largura do chanfro (inc)		
18	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>		
19		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem/apenas cálculo do ponto de partida	
					0 =	Modo de compatibilidade
					1 =	Usinagem normal
				MILHAR:	reservado	
				DEZENA DA MILHAR:	Usinagem completa/retrabalho (reusinagem)	
					0 =	Modo de compatibilidade (<_AP1> tratar como até então)
					1 =	Usinagem completa
					2 =	Retrabalho

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
20		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	---	reservado
				CENTENA:	---	reservado
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples
21		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final Z1 (DP)	
					0 =	Absoluto (Modo de compatibilidade)
					1 =	incremental
					2 =	Absoluto
				DEZENA:	reservado	
				CENTENA:	Profundidade de imersão no chanfra-mento (ZFS)	
					0 =	Absoluto
					1 =	incremental

20.1.19 CYCLE78 - Fresar rosca

Sintaxe

```
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>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Profundidade final de furação (abs/ink), consultar <_AMODE>		
5		<_ADPR>	REAL	Profundidade de pré-furação com avanço de furação reduzido (inc), ativo com <_VARI> DEZENA DE MILHAR		
6	D	<_FDPR>	REAL	penetração máxima em profundidade (inc) D ≥ Z1 ⇒ uma penetração na profundidade final de furação D < Z1 ⇒ ciclo de furação profunda com várias penetrações e remoção de cavacos		
7	ZR	<_LDPR>	REAL	Profundidade de furação restante na furação passante (inc), com avanço FR		
8	∅	<_DIAM>	REAL	Diâmetro nominal da rosca		
9	P	<_PIT>	REAL	Passo de rosca como valor numérico		
10		<_PITA>	INT	Interpretação do passo de rosca P		
					1 =	Passo em mm/U
					2 =	Passo em voltas/polegada
					3 =	Passo em polegada/U
					4 =	Passo como MÓDULO
11	DF	<_DAM>	REAL	Quantia/valor em porcentagem para cada penetração seguinte (decréscimo), consultar <_AMODE>		
12	V1	<_MDEP>	REAL	penetração mínima (inc), ativa apenas no decréscimo		

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
13		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	reservado	
				DEZENA:	Remoção de cavacos antes do fresamento de rosca	
					0 =	sem remoção de cavacos antes do fresamento de rosca (atua somente na profundidade final de furação)
					1 =	remoção de cavacos antes do fresamento de rosca (atua somente na profundidade final de furação)
				CENTENA:	Rosca à direita / à esquerda	
					0 =	Rosca à direita
					1 =	Rosca à esquerda
				MILHAR:	Profundidade de furação restante com avanço de furação	
					0 =	Sem profundidade de furação restante com avanço de furação FR
					1 =	Profundidade de furação restante com avanço de furação FR
				DEZENA DA MILHAR:	Pré-furação com avanço reduzido	
					0 =	sem pré-furação com avanço reduzido
					1 =	Pré-furação com avanço reduzido avanço de pré-furação = 0,3 F1, se F1 < 0,15 mm/U avanço de pré-furação = 0,1 mm/U, se F1 ≥0,15 mm/U
14		<_CDIR>	INT	Sentido de fresamento	0 =	Concordante
					1 =	Discordante
					4 =	Discordante + concordante (combinação de desbaste + acabamento)
15	Z2	<_GE>	REAL	Valor de retrocesso antes do fresamento de rosca (inc)		
16	F1	<_FFD>	REAL	Avanço de furação (mm/min e pol./min ou mm/rot.)		
17	FR	<_FRDP>	REAL	Avanço de furação para profundidade de furação restante (mm/min ou mm/rot.)		
18	F2	<_FFR>	REAL	Avanço para fresamento de rosca (mm/min ou mm/dente)		
19	FS	<_FFP2>	REAL	Avanço de acabamento para <_CDIR> = 4 (mm/min ou mm/dente)		

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
20		<_FFA>	INT	Avaliação dos avanços		
				UNIDADE:	Avanço de furação F1	
				DEZENA:	Avanço de furação para profundidade de furação restante FR	
				CENTENA:	Avanço para fresamento de rosca F2	
				MILHAR:	Avanço de acabamento FS	
21		<_PITM>	STRING[15]	String como marcador para entrada do passo de rosca (apenas para a interface)[1)]		
22		<_PTAB>	STRING[20]	String para tabela de roscas ("", "ISO", "BSW", "BSP", "UNC") (apenas para a interface)[1)]		
23		<_PTABA>	STRING[20]	String para seleção na tabela de roscas (por ex., "M 10", "M 12", ...) apenas para superfície) [1)]		
24		<_GMODE>	INT	Modo geométrico (avaliação dos valores geométricos programados), reservado		
25		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
26		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade de furação = profundidade final de furação Z1 abs/inc	
					0 =	Absoluto
					1 =	incremental
				DEZENA:	Quantia/valor em porcentagem do DF para cada penetração seguinte (decréscimo)	
					0 =	Valor
					1 =	Porcentagem (0.001 a 100 %)

Indicação

[1)] Os parâmetros 21, 22 e 23 somente são utilizados na seleção de roscas nas tabelas de roscas da tela de especificação. Um acesso às tabelas de roscas através da definição de ciclo durante o processamento do ciclo não é possível.

20.1.20 CYCLE79 - Poliedro

Sintaxe

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)		
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Profundidade do poliedro (abs/inc), consulte <_AMODE>		
5	N	<_NUM>	INT	Número de cantos (1...n)		
6	SW / L	<_SWL>	REAL	Abertura de chave ou comprimento de canto (dependendo de <_VARI>) ("SW" para abertura de chave, "L" para comprimento de canto) Abertura de chave somente para número par de cantos, e um só canto		
7	X0	<_PA>	REAL	Ponto de referência da saliência, 1º eixo (abs)		
8	Y0	<_PO>	REAL	Ponto de referência da saliência, 2º eixo (abs)		
9	α0	<_STA>	REAL	Ângulo de giro no centro do canto contra o 1º eixo (eixo X)		
10	R1/FS1	<_RC>	REAL	Arredondamento de canto com<_NUM> > 2 (raio/chanfro, consulte <_AMODE>) (inc, especificar sem sinal prévio) ("R1" para raio, "FS1" para chanfro)		
11	∅	<_AP1>	REAL	Diâmetro bruto da saliência		
12	DXY	<_MIDA>	REAL	Largura máxima de penetração (unidade, consulte <_AMODE>)		
13	DZ	<_MID>	REAL	penetração máxima em profundidade		
14	UXY	<_FAL>	REAL	Sobremetal de acabamento do plano		
15	UZ	<_FALD>	REAL	Sobremetal de acabamento da profundidade		
16	F	<_FFP1>	REAL	Avanço de usinagem		
17		<_CDIR>	INT	Sentido de fresamento	0 =	Concordante
					1 =	Discordante

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
18		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Usinagem	
					1 =	Desbaste
					2 =	Acabamento
					3 =	Acabamento da borda
					5 =	Chanframento
				DEZENA:	Abertura de chave ou comprimento do canto	
					0 =	Abertura de chave
					1 =	Comprimento de canto
19	FS	<_FS>	REAL	Largura do chanfro (inc)		
20	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>		
21		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem ou apenas cálculo do ponto de partida	
					1 =	Usinagem normal
22		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/18/19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	---	reservado
				CENTENA:	---	reservado
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
23		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade final (<_DP>)	
					0 =	Absoluto
					1 =	incremental
				DEZENA:	Unidade para penetração no plano (<_MIDA>)	
					0 =	mm
					1 =	% do diâmetro da ferramenta
				CENTENA:	Profundidade de imersão no chanfra-mento (<_ZFS>)	
					0 =	Absoluto
					1 =	incremental
				MILHAR:	Arredondamento de canto (<_RC>)	
					0 =	Raio
					1 =	Chanfro

20.1.21 CYCLE81 - Furação, centragem

Sintaxe

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

Parâmetro

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<RFP>	REAL	Ponto de referência (abs)		
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1/∅	<DP>	REAL	Profundidade de furação (abs) / diâmetro da centragem (abs), consultar <_GMODE>		
5	Z1	<DPR>	REAL	Profundidade de furação (inc)		
6	DT	<DTB>	REAL	Tempo de permanência na profundidade final de furação, consultar <_AMODE>		
7		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	Centragem relacionada à profundidade/ ao diâmetro	
					0 =	Compatibilidade, profundidade
					1 =	Diâmetro

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
8		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
9		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade de furação Z1 (abs/inc)	
					0 =	Compatibilidade, a partir da programação DP/DPR
					1 =	incremental
					2 =	Absoluto
				DEZENA:	Tempo de espera na profundidade final de furação DT em segundos/rotações	
					0 =	Compatibilidade, do sinal do DTB (> 0 segundos ou < 0 rotações)
					1 =	Em segundos
					2 =	Em rotações

20.1.22 CYCLE82 - Furação, escareamento plano

Sintaxe

```
CYCLE82(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

Parâmetro

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	RP	<RTP>	REAL	Plano de retrocesso (abs)
2	Z0	<RFP>	REAL	Ponto de referência (abs)
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)
4	Z1	<DP>	REAL	Profundidade de furação (abs), consultar <_AMODE>
5	Z1	<DPR>	REAL	Profundidade de furação (ink), consultar <_AMODE>
6	DT	<DTB>	REAL	Tempo de permanência na profundidade final de furação, consultar <_AMODE>

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
7		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	Profundidade de furação relativa à ponta/haste	
					0 =	Compatibilidade, pico
					1 =	Haste
8		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	reservado	
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
9		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade de furação Z1 (abs/inc)	
					0 =	Compatibilidade a partir da programação DP/DPR
					1 =	incremental
					2 =	Absoluto
				DEZENA:	Tempo de espera DT na profundidade final de furação em segundos/rotações	
					0 =	compatibilidade, a partir do sinal prévio de DT (> 0 segundos / < 0 rotações)
					1 =	Em segundos
					2 =	Em rotações
				CENTENA:	Profundidade da furação ZA abs/inc	
					0 =	incremental
					1 =	Absoluto
				MILHAR:	Avaliação do avanço da furação	
					0 =	Em % do avanço da furação
					1 =	F/min
					2 =	F/U
				DEZENA DA MILHAR:	Profundidade da furação restante ZD abs/ink	
					0 =	incremental
					1 =	Absoluto
				CENTENA DA MILHAR:	Avaliação do avanço da furação	
					0 =	Em % do avanço da furação
					1 =	F/min
					2 =	F/U
10		<_VARI>	INT	Tipo de trabalho furação/perfuração		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	Perfuração	
					0 =	Perfurar "Não"
					1 =	Perfurar "Sim"
				DEZENA DA MILHAR:	Furação	
					0 =	Furar "Não"
					1 =	Furar "Sim"
11	ZA	<S_ZA>	REAL	Profundidade da furação incremental relacionada ao ponto de referência ou absoluto (consultar<_AMODE> CENTENA)		
12	FA	<S_FA>	REAL	Avanço de furação como valor ou em % (em conexão com<_AMODE> MILHAR)		

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado
13	ZD	<S_ZD>	REAL	Profundidade da furação restante incremental relacionada à profundidade da furação final ou absoluto (em conexão com<_AMODE> DEZENA DE MILHAR)
14	FD	<S_FD>	REAL	Avanço de furação restante como valor ou em % (em conexão com<_AMODE> CENTENA DE MILHAR)

20.1.23 CYCLE83 - Furação profunda

Sintaxe

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>, <_DAM>,
<DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>, <_DTD>,
<_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<RFP>	REAL	Ponto de referência (abs)		
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<DP>	REAL	Profundidade final de furação (abs), consultar <_AMODE>		
5	Z1	<DPR>	REAL	Profundidade final de furação (ink), consultar <_AMODE>		
6	D	<FDEP>	REAL	1. Profundidade de perfuração (abs), veja <_AMODE>		
7	D	<FDPR>	REAL	1. Profundidade de perfuração (ink), veja <_AMODE>		
8	DF	<_DAM>	REAL	Quantidade de degressão/em porcentagem para cada entrega a mais, veja <_AMODE>		
9	DTB	<DTB>	REAL	Tempo de permanência na profundidade de furação, consultar <_AMODE>		
10	DTS	<DTS>	REAL	Tempo de permanência no ponto inicial (apenas para remoção de cavacos), consultar <_AMODE>		
11	FD1	<FRF>	REAL	Porcentagem para o avanço na primeira penetração, consultar <_AMODE>		
12		<VARI>	INT	Tipo de usinagem		
				UNIDADE:	Quebra de cavacos / remoção de cavacos	
					0 =	Quebra de cavacos
					1 =	Remoção de cavacos

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
13		<_AXN>	INT	Eixo da ferramenta		
					0 =	3º eixo geométrico
					1 =	1º eixo geométrico
					2 =	2º eixo geométrico
					>2	3º eixo geométrico
14	V1	<_MDEP>	REAL	penetração mínima (apenas para valor em porcentagem para decréscimo)		
15	V2	<_VRT>	REAL	Valor de retrocesso depois de cada usinagem (somente para quebra de cavacos)		
					>0	Valor de retrocesso variável
					0 =	Valor padrão 1 mm
16	DT	<_DTD>	REAL	Tempo de permanência na profundidade final de furação, consultar <_AMODE>		
17	V3	<_DIS1>	REAL	Distância de parada prévia (somente para remoção de cavacos), consultar <_AMODE>		
18		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	Profundidade de furação relativa à ponta/haste	
					0 =	Ponta
					1 =	Haste
19		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	---	reservado
				CENTENA:	---	reservado
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
20		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade de furação = profundidade final de furação Z1 (abs/inc)	
					0 =	Compatibilidade, a partir da programação <DP>/<DPR>
					1 =	incremental
					2 =	Absoluto
				DEZENA:	Tempo de espera na profundidade de furação DTB em segundos/rotações	
					0 =	Compatibilidade a partir do sinal prévio de DTB (> 0 segundos ou < 0 rotações)
					1 =	Em segundos
					2 =	Em rotações
				CENTENA:	Tempo de espera no ponto inicial do DTS em segundos/rotações	
					0 =	Compatibilidade do sinal de DTS (> 0 segundos ou < 0 rotações)
					1 =	Em segundos
					2 =	Em rotações
				MILHAR:	Tempo de permanência na profundidade final de furação DTD em segundos/rotações	
					0 =	Compatibilidade do sinal de DTD (> 0 segundos ou < 0 rotações)
					1 =	Em segundos
					2 =	Em rotações
				DEZENA DA MILHAR:	1. Profundidade de perfuração D (abs/ink)	
					0 =	Compatibilidade, a partir da programação <FDEPF>/<DPR>
					1 =	incremental
					2 =	Absoluto
				CENTENA DA MILHAR:	Quantidade de degressão/em porcentagem <_DAM> para cada entrega a mais	
					0 =	Compatibilidade, a partir do sinal da<_DAM> (> 0 quantidade de degressão ou < 0 fator 0,001 até 1,0)
					1 =	Quantidade de degressão
					2 =	Porcentagem (0,001 a 100 %)
				UNIDADE DE MILHÃO:	Distância de parada prévia V3 automática/manual	
					0 =	Compatibilidade a partir do sinal de<_DIS1> (= 0 automático ou > 0 manual)

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
					1 =	Automático (é calculado no ciclo)
					2 =	Manual (valor programado)
				DEZENA DE MILHÃO:	Fator de avanço para a primeira penetração <FRF> como fator/porcentagem	
					0 =	Compatibilidade, como fator (0,001 a 1,0, FRF = 0 significa 100 %)
					1 =	Porcentagem (0,001 a 999,999 %)

20.1.24 CYCLE84 - Rosqueamento com macho sem mandril de compensação

Sintaxe

```
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>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<RFP>	REAL	Ponto de referência (abs)		
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<DP>	REAL	Profundidade de furação = profundidade final de furação (abs), consultar <_AMODE>		
5	Z1	<DPR>	REAL	Profundidade de furação = profundidade final de furação (ink), consultar <_AMODE>		
6	DT	<DTB>	REAL	Tempo de espera da profundidade de furação em segundos		
7	SDE	<SDAC>	INT	Sentido de rotação após fim do ciclo		
8		<MPIT>	REAL	Tamanho de rosca somente para "ISO métrico" (o passo é calculado internamente durante o processamento)		
9	P	<PIT>	REAL	Passo de rosca como valor, consultar a unidade de medida <_PITA>		
10	αS[1)]	<POSS>	REAL	Posição do fuso para parada orientada do fuso		
11	S	<SST>	REAL	Rotação do fuso para rosqueamento com macho		
12	SR	<SST1>	REAL	Rotação do fuso para retrocesso		
13		<_AXN>	INT	Eixo de furação	0 =	3. Eixo geométrico
					1 =	1. Eixo geométrico
					2 =	2. Eixo geométrico
					≥ 3 =	3. Eixo geométrico

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado			
14		<_PITA>	INT	Unidade de medida do passo de rosca (avaliação <PIT> e <MPIT>)			
					0 =	Passo em mm	- Avaliação <MPIT>/<PIT>
					1 =	Passo em mm	- Avaliação <PIT>
					2 =	Passo em TPI	- Avaliação <PIT> (passos de rosca por polegada)
					3 =	Passo em polegadas	- Avaliação <PIT>
					4 =	MÓDULO	- Avaliação <PIT>
15		<_TECHNO>	INT	Tecnologia[1)]			
				UNIDADE:	Comportamento de parada exata		
					0 =	Comportamento de parada exata como o ativado antes da chamada do ciclo	
					1 =	Parada exata G601	
					2 =	Parada exata G602	
					3 =	Parada exata G603	
				DEZENA:	Controle feedforward		
					0 =	com/sem controle feedforward como o ativado antes da chamada do ciclo	
					1 =	com Feedforward FFWON	
					2 =	sem Feedforward FFWOF	
				CENTENA:	Aceleração		
					0 =	SOFT/BRISK/DRIVE como ativado antes da chamada do ciclo	
					1 =	com limitação de solavancos SOFT	
					2 =	sem limitação de solavancos BRISK	
					3 =	aceleração reduzida DRIVE	
				MILHAR:	Modo de fuso com MCALL		
					0 =	reativação do modo de fuso com MCALL	
					1 =	permanência em controle de posição com MCALL	
16		<_VARI>	INT	Tipo de usinagem			
				UNIDADE:	0 =	1 corte	
					1 =	Quebra de cavacos (rosqueamento com macho para furo profundo)	
					2 =	Remoção de cavacos (rosqueamento com macho para furo profundo)	
				MILHAR:	Modo ISO/SIEMENS não relevante para tela de especificação		
					0 =	Chamada a partir da compatibilidade ISO	
					1 =	Chamada a partir do contexto SIEMENS	
17	D	<_DAM>	REAL	penetração máxima em profundidade (apenas para remoção de cavacos / quebra de cavacos)			

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
18	V2	<_VRT>	REAL	Valor de retrocesso depois de cada usinagem (somente para quebra de cavacos), consultar <_AMODE>		
19		<_PITM>	STRING[15]	String como marcador para entrada do passo da rosca[2)]		
20		<_PTAB>	STRING[5]	String para tabela de roscas ("", "ISO", "BSW", "BSP", "UNC")[2)]		
21		<_PTABA>	STRING[20]	String para seleção na tabela de roscas (p. ex. "M 10", "M 12", ...)[2)]		
22		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
23		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	Modo de compatibilidade (apenas para tela de especificação para recompilação), se o dado MD 52216 Bit0 = 1[1)]	
					0 =	Os parâmetros de tecnologia são exibidos (compatibilidade): Sob efeito dos parâmetros TECHNO
					1 =	Os parâmetros de tecnologia não são exibidos: Sob efeito da tecnologia "como o programado antes da chamada do ciclo"
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
24		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade de furação = profundidade final de furação Z1 (abs/inc)	
					0 =	Compatibilidade, a partir da programação <DP>/<DPR>
					1 =	incremental
					2 =	Absoluto
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	Sentido de giro da rosca à direita/esquerda	
					0 =	Compatibilidade, a partir do sinal PIT/MPIT
					1 =	direita
					2 =	esquerda
				DEZENA DA MILHAR:	reservado	
				CENTENA DA MILHAR:	reservado	
				UNIDADE DE MILHÃO:	Valor de retrocesso após cada usinagem V2 manual/automática	
					0 =	Compatibilidade, a partir da programação <_VRT> (> 0 valor variável ou ≤ 0 valor padrão 1 mm/0,0394 polegada)
					1 =	automático (valor padrão de 1mm/0,0394 polegada)
					2 =	manual (como programado em V2)

[1] Os campos de tecnologia podem ser ocultados em função do dado de ajuste SD52216 $MCS_FUNCTION_MASK_DRILL

[2] Os parâmetros 19, 20 e 21 somente são utilizados na seleção de roscas nas tabelas de roscas da tela de especificação. Um acesso às tabelas de roscas através da definição de ciclo durante o processamento do ciclo não é possível.

20.1.25 CYCLE85 - Alargamento

Sintaxe

```
CYCLE85(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

Parâmetro

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	RP	<RTP>	REAL	Plano de retrocesso (abs)
2	Z0	<RFP>	REAL	Ponto de referência (abs)

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<DP>	REAL	Profundidade de furação (abs), consultar <_AMODE>		
5	Z1	<DPR>	REAL	Profundidade de furação (ink), consultar <_AMODE>		
6	DT	<DTB>	REAL	Tempo de permanência na profundidade final de furação, consultar <_AMODE>		
7	F	<FFR>	REAL	Avanço		
8	FR	<RFF>	REAL	Avanço no retrocesso		
9		<_GMODE>	INT	reservado		
10		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
11		<_AMODE>	INT	Modo alternativo (furação)		
				UNIDADE:	Profundidade de furação Z1 (abs/inc)	
					0 =	Compatibilidade, a partir da programação DP/DPR
					1 =	incremental
					2 =	Absoluto
				DEZENA:	Tempo de espera DT na profundidade final de furação em segundos/rotações	
					0 =	Compatibilidade, a partir do sinal de DT (> 0 segundos ou < 0 rotações)
					1 =	Em segundos
					2 =	Em rotações

20.1.26 CYCLE86 - Mandrilamento

Sintaxe

```
CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>,
<RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<RFP>	REAL	Ponto de referência (abs)		
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<DP>	REAL	Profundidade de furação (abs), consultar <_AMODE>		
5	Z1	<DPR>	REAL	Profundidade de furação (ink), consultar <_AMODE>		
6	DT	<DTB>	REAL	Tempo de permanência na profundidade final de furação, consultar <_AMODE>		
7	DIR	<SDIR>	INT	Sentido de giro do fuso	3 =	M3
					4 =	M4
8	DX	<RPA>	REAL	Valor de retração no sentido X		
9	DY	<RPO>	REAL	Valor de retração no sentido Y		
10	DZ	<RPAP>	REAL	Valor de retração no sentido Z		
11	SPOS	<POSS>	REAL	Posição do fuso para retração (para parada de fuso orientada, em graus)		
12		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	Modo de retração	
					0 =	retrair, compatibilidade
					1 =	não retrair
13		<_DMODE>	INT	Modo de exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
14		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade de furação Z1 (abs/inc)	
					0 =	Compatibilidade, a partir da programação <DP>/<DPR>
					1 =	incremental
					2 =	Absoluto
				DEZENA:	Tempo de espera na profundidade final de furação DT em segundos/rotações	
					0 =	Compatibilidade, a partir do sinal de DT (> 0 segundos ou < 0 rotações)
					1 =	Em segundos
					2 =	Em rotações

20.1.27 CYCLE92 - Separação

Sintaxe

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>, <_SV1>,
<_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>, <_VARI>, <_DN>,
<_DMODE>, <_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	X0	<_SPD>	REAL	Ponto de referência (abs, sempre em diâmetro)		
2	Y0	<_SPL>	REAL	Ponto de referência (abs)		
3	X1	<_DIAG1>	REAL	Profundidade para redução da rotação, consulte<_AMODE> (UNIDADE)		
4	X2	<_DIAG2>	REAL	Profundidade final, consulte<_AMODE> (DEZENA)		
5	R/FS	<_RC>	REAL	Raio de arredondamento ou largura do chanfro, consulte<_AMODE> (MILHAR)		
6	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
7	S	<_SV1>	REAL	Rotação constante do fuso, consulte <_AMODE> (DEZENA DE MILHAR)		
	V			Velocidade de corte constante		
8	SV	<_SV2>	REAL	Rotação máxima com velocidade de corte constante		
9	DIR	<_SDAC>	INT	Sentido de giro do fuso	3 =	para M3
					4 =	para M4
10	F	<_FF1>	REAL	Avanço até a profundidade para redução da rotação		
11	FR	<_FF2>	REAL	avanço reduzido até a profundidade final		
12	SR	<_SS2>	REAL	rotação reduzida até a profundidade final		
13	XM	<_DIAGM>	REAL	Avanço da profundidade da garra de peças (abs, sempre em diâmetro)		
14		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Retrocesso	
					0 =	Retrocesso até<_SPD> + <_SDIS>
					1 =	Sem retrocesso no final
				DEZENA:	Garra de peças	
					0 =	Não executar nenhum comando M
					1 =	Sim, acesso de CUST_TECHCYC(101)- avanço de gaveta, CUST_TECHCYC(102)- fechamento da gaveta
15		<_DN>	INT	Número D para o 2º corte, se não estiver programado ⇒ D+1		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
20		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
21		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade para redução da rotação (<_DIAG1>)	
					0 =	Absoluto, valor do eixo transversal em diâmetro
					1 =	Incremental, valor do eixo transversal em raio
				DEZENA:	Profundidade final (<_DIAG2>)	
					0 =	Absoluto, valor do eixo transversal em diâmetro
					1 =	Incremental, valor do eixo transversal em raio
				CENTENA:	reservado	
				MILHAR:	Raio/chanfro (<_RC>)	
					0 =	Raio
					1 =	Chanfro
				DEZENA DA MILHAR:	Rotação do fuso/velocidade de corte (<_SV1>)	
					0 =	Rotação constante do fuso
					1 =	Velocidade de corte constante

20.1.28 CYCLE95 - Desbaste do contorno

Sintaxe

```
CYCLE95(<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>,
<_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	CON	<NPP>	STRING [140]	Nome do contorno		
2	D	<MID>	REAL	penetração máxima em profundidade no desbaste, consulte <_GMODE>		
3	UZ	<FALZ>	REAL	Sobremetal de acabamento em Z		
4	UX	<FALX>	REAL	Sobremetal de acabamento em X		
5	U	<FAL>	REAL	Sobremetal de acabamento paralelo ao contorno (atua em ambos os eixos)		
6	F	<FF1>	REAL	Avanço para desbaste		
7	FY	<FF2>	REAL	Avanço de imersão detalonados		
8	FS	<FF3>	REAL	Avanço de acabamento		
9		<_VARI>	INT	Tipo de usinagem		
				UNIDADE e DEZENA:		
					1 =	Desbaste, longitudinal, externo
					2 =	Desbaste, plano, externo
					3 =	Desbaste, longitudinal, interno
					4 =	Desbaste, plano, interno
					5 =	Acabamento, longitudinal, externo
					6 =	Acabamento, plano, externo
					7 =	Acabamento, longitudinal, interno
					8 =	Acabamento, plano, interno
					9 =	Usinagem completa, longitudinal, externa
					10 =	Usinagem completa, plana, externa
					11 =	Usinagem completa, longitudinal, interna
					12 =	Usinagem completa, plana, interna
				CENTENA:		
					0 =	Com retoque no contorno, sem cantos residuais
					1 =	Sem retoque de contornos
					2 =	Retoque apenas até o ponto de corte anterior, podem se formar cantos residuais
10	DT	<DT>	REAL	Tempo de espera na interrupção de avanço		
11	DI	<DAM>	REAL	Distância de interrupção de avanço		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
12	VRT	<_VRT>	REAL	Curso de retração do contorno		
					0 =	Internamente é utilizado um curso de retração de 1 mm independentemente do sistema de medidas polegada/métrico
					> 0 =	Curso de retração
13		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	Avaliação da profundidade da penetração	
					0 =	A profundidade de penetração será compensada de acordo com o grupo G DIAMON/DIAMOF
					1 =	A profundidade de penetração atua como valor de raio (independente de DIAMON/DIAMOF)
14		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				MILHAR:		
					0 =	Modo de compatibilidade: O nome do contorno se encontra em NPP
					1 =	O nome do contorno é programado em CYCLE62 e transferido para _SC_CONT_NAME

20.1.29 CYCLE98 - encadeamento de roscas

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	Z0	<_PO1>	REAL	Ponto de referência em Z (abs)		
2	X0	<_DM1>	REAL	Ponto de referência em X (abs), em diâmetro		
3	Z1	<_PO2>	REAL	Ponto intermediário 1 em Z (abs/inc), consulte<_AMODE> (UNIDADE)		
4	X1	<_DM2>	REAL	Ponto intermediário 1 em X (abs/inc), consulte<_AMODE> (DEZENA) ou		
	X1α			Inclinação de rosca 1 (-90° até 90°) abs sempre em diâmetro, inc sempre em raio		
5	Z2	<_PO3>	REAL	Ponto intermediário 2 em Z (abs/inc), consulte<_AMODE> (CENTENA)		
6	X2	<_DM3>	REAL	Ponto intermediário 2 em X (abs/inc), consulte<_AMODE> (MILHAR) ou		
	X2α			Inclinação de rosca 2 (-90° até 90°) abs sempre em diâmetro, inc sempre em raio		
7	Z3	<_PO4>	REAL	Ponto final em Z (abs/inc), consulte <_AMODE> (DEZENA DE MILHAR)		
8	X3	<_DM4>	REAL	Ponto final em X (abs/inc), consulte<_AMODE> (CENTENA DE MILHAR) ou		
	X3α			Inclinação de rosca 3 (-90° até 90°) abs sempre em diâmetro, inc sempre em raio		
9	LW	<APP>	REAL	Curso precedente da rosca (inc, especificar com sinal)		
10	LR	<ROP>	REAL	Saida da rosca (inc, especificar com sinal)		
11	H1	<TDEP>	REAL	Profundidade da rosca (inc, especificar com sinal)		
12	U	<FAL>	REAL	Sobremetal de acabamento em X e Z		
13	DP	<_IANG>	REAL	Inclinação de penetração como distância ou ângulo, consulte <_AMODE> (UNIDADE DE MILHÃO)		
	αP			A inclinação de penetração atua de acordo com a configuração do parâmetro<_VARI> (CENTENA).		
					Definição parar <_VARI_CENTENA = 0 - modo de compatibilidade:	
					> 0 =	penetração de flancos em um flanco
					0 =	penetração vertical na rosca
					< 0 =	penetração de flancos com alternância dos flancos
					Definição para _VARI_CENTENA<>0:	
					> 0 =	penetração no flanco positivo
					0 =	Penetração centralizada
					< 0 =	penetração no flanco negativo
14	α0	<NSP>	REAL	Deslocamento do ângulo de partida para o 1º passo da rosca		
15		<NRC>	INT	Número de cortes de desbaste, consulte<_VARI> (DEZENA DE MILHAR)		
16	NN	<NID>	INT	Número de cortes em vazio		
17	P0	<_PP1>	REAL	Passo da rosca do 1º segmento de rosca, consulte <_PITA>		
18	P1	<_PP2>	REAL	Passo da rosca do 2º segmento de rosca, consulte <_PITA>		
19	P2	<_PP3>	REAL	Passo da rosca do 3º segmento de rosca, consulte <_PITA>		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
20		<_VARI>	INT	Usinagem		
				UNIDADE:	Tecnologia	
					1 =	Rosca externa com penetração linear
					2 =	Rosca interna com penetração linear
					3 =	Rosca externa com penetração degressiva, e a secção de corte permanece constante
					4 =	Rosca interna com penetração degressiva, e a secção de corte permanece constante
				DEZENA:	reservado	
				CENTENA:	Tipo de penetração	
					0 =	Modo de compatibilidade para <_IANG>
					1 =	Penetração unilateral
					2 =	Penetração alternada
				MILHAR:	reservado	
				DEZENA DA MILHAR:	Penetração em profundidade alternativa	
					0 =	Compatibilidade, especificação do número de cortes de desbaste (<_NRC>)
					1 =	Especificação do valor para a 1ª penetração (<_MID>)
				CENTENA DA MILHAR:	Tipo de usinagem	
					0 =	Compatibilidade (desbaste e acabamento)
					1 =	Desbaste
					2 =	Acabamento
					3 =	Desbaste e acabamento
				UNIDADE DE MILHÃO:	Sequência de usinagem para roscas de múltiplas entradas	
					0 =	Sequência dos passos crescente
					1 =	Sequência dos passos em relação a
21	N	<_NUMTH>	INT	Número de passos de rosca		
22		<_VRT>	REAL	Distância de retrocesso (inc)		
					0 =	Internamente é utilizado um curso de retração de 1 mm independentemente do sistema de medidas polegada/métrico
					> 0 =	Curso de retração
23	D1	<_MID>	REAL	Primeira penetração, consulte<_VARI> (DEZENA DE MILHAR)		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
24	DA	<_GDEP>	REAL	Profundidade de mudança de passo (apenas atua na versão "uma entrada")		
					0 =	Sem considerar a profundidade de mudança de passo
					> 0 =	Considerar a profundidade de mudança de passo
25		<_IFLANK>	REAL	Inclinação de penetração como largura (apenas para a interface)		
26		<_PITA>	INT	Avaliação do passo da rosca		
					0 =	Modo de compatibilidade para o passo de rosca, avaliação<_PP1> até<_PP3> como atualmente de acordo com o sistema ativo métrico/polegadas
					1 =	Passo em mm
					2 =	Passo em TPI (passos de rosca por polegada)
					3 =	Passo em polegadas
					4 =	MÓDULO
27		<_PITM1>	STRING[15]	String como marcador para entrada do passo de rosca (apenas para a interface)		
28		<_PITM2>	STRING[15]	String como marcador para entrada do passo de rosca (apenas para a interface)		
29		<_PITM3>	STRING[15]	String como marcador para entrada do passo de rosca (apenas para a interface)		
30		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	---	reservado
				CENTENA:	---	reservado
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
31		<_AMODE>	INT	modo alternativo		
				UNIDADE:	1. Ponto intermediário em Z (Z1)	
					0 =	Absoluto
					1 =	incremental
				DEZENA:	1. Ponto intermediário em X (X1)	
					0 =	Absoluto
					1 =	incremental
					2 =	α
				CENTENA:	2. Ponto intermediário em Z (Z2)	
					0 =	Absoluto
					1 =	incremental
				MILHAR:	2. Ponto intermediário em X (X2)	
					0 =	Absoluto
					1 =	incremental
					2 =	α
				DEZENA DA MILHAR:	Ponto final em Z (Z3)	
					0 =	Absoluto
					1 =	incremental
				CENTENA DA MILHAR:	Ponto final em X (X3)	
					0 =	Absoluto
					1 =	incremental
					2 =	α
				UNIDADE DE MILHÃO:	Seleção da inclinação de penetração como ângulo ou largura	
					0 =	Ângulo de penetração<_IANG>
					1 =	Inclinação de penetração <_IFLANK>
				DEZENA DE MILHÃO:	de uma entrada/de múltiplas entradas	
					0 =	Modo de compatibilidade (ângulo inicial<_NSP> será avaliado)
					1 =	Uma entrada (com desvio do ângulo inicial <_NSP>)
					2 =	múltiplas entradas

20.1.30 CYCLE99 - Torneamento de roscas

Sintaxe

```
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>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	Z0	<_SPL>	REAL	Ponto de referência (abs)		
2	X0	<_SPD>	REAL	Ponto de referência (abs, sempre em diâmetro)		
3	Z1	<_FPL>	REAL	Ponto final em ligação com<_AMODE> (UNIDADE)		
4	X1	<_FPD>	REAL	Ponto final em ligação com<_AMODE> (DEZENA)		
5	LW/LW2	<_APP>	REAL	Curso precedente da rosca em ligação com<_AMODE> (CENTENA) ou Entrada da rosca = saída da rosca em ligação com<_AMODE> (CENTENA)		
6	LR	<_ROP>	REAL	Saída da rosca		
7	H1	<_TDEP>	REAL	Profundidade da rosca		
8	U	<_FAL>	REAL	Sobremetal de acabamento em X e Z		
9	DP	<_IANG>	REAL	Inclinação de penetração como distância ou ângulo em ligação com<_AMODE> (MILHAR)		
	αP				> 0 =	penetração no flanco positivo
					< 0 =	penetração no flanco negativo
					0 =	Penetração centralizada
10	α0	<_NSP>	REAL	Deslocamento do ângulo de partida (apenas atua na versão "uma entrada")		
11	ND	<_NRC>	INT	Número de cortes de desbaste, em ligação com<_VARI> (DEZENA DE MILHAR)		
12	NN	<_NID>	INT	Número de cortes em vazio		
13	P	<_PIT>	REAL	Passo de rosca como valor, em ligação com <_PITA>		

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
14		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Tecnologia	
					1 =	Rosca externa com penetração linear
					2 =	Rosca interna com penetração linear
					3 =	Rosca externa com penetração degressiva, e a secção de corte permanece constante
					4 =	Rosca interna com penetração degressiva, e a secção de corte permanece constante
				DEZENA:	reservado	
				CENTENA:	Tipo de penetração	
					1 =	Penetração unilateral
					2 =	Penetração alternada
				MILHAR:	reservado	
				DEZENA DA MILHAR:	Penetração em profundidade alternativa	
					0 =	Especificação do número de cortes de desbaste (<_NRC>)
					1 =	Especificação do valor para a 1ª penetração (<_MID>)
				CENTENA DA MILHAR:	Tipo de usinagem	
					1 =	Desbaste
					2 =	Acabamento
					3 =	Desbaste e acabamento
				UNIDADE DE MILHÃO:	Sequência de usinagem para roscas de múltiplas entradas	
					0 =	Sequência dos passos crescente
					1 =	Sequência dos passos em relação a
15	N	<_NUMTH>	INT	Número de passos de rosca		
16	VR	<_SDIS>	REAL	Distância de retrocesso, inc		
17	D1	<_MID>	REAL	Primeira profundidade de penetração, em ligação com<_VARI> (DEZENA DE MILHAR)		
18	DA	<_GDEP>	REAL	Profundidade de mudança de passo (apenas atua na versão "múltiplas entradas")		
					0 =	Sem considerar a profundidade de mudança de passo
					> 0 =	Considerar a profundidade de mudança de passo

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
19	G	<_PIT1>	REAL	Alteração do passo da rosca por rotação		
					0 =	O passo de rosca é constante (G33)
					> 0 =	O passo da rosca é cada vez maior (G34)
					< 0 =	O passo da rosca é cada vez menor (G35)
20		<_FDEP>	REAL	Profundidade de imersão (especificar sem sinal)		
21	N1	<_GST>	INT	Passo inicial N1 = 1...N, em ligação com<_AMODE> (CENTENA DE MILHAR)		
22		<_GUD>	INT	reservado		
23		<_IFLANK>	REAL	Inclinação de penetração como largura (apenas para a interface)		
24		<_PITA>	INT	Unidade de medida do passo da rosca (interpretação do PIT e/ou MPIT)		
					0 =	Passo em mm - avaliação MPIT/PIT
					1 =	Passo em mm - avaliação PIT
					2 =	Passo em TPI - avaliação PIT (passos de rosca por polegada)
					3 =	Passo em polegada - avaliação PIT
					4 =	MÓDULO - avaliação PIT
25		<_PITM>	STRING[15]	String como marcador para entrada do passo de rosca (apenas para a interface)[1)]		
26		<_PTAB>	STRING[20]	String para tabela de roscas (apenas para a interface) [1)]		
27		<_PTABA>	STRING[20]	String para seleção na tabela de roscas (apenas para a interface) [1)]		
28		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Tipo de rosca	
					0 =	Rosca longitudinal
					1 =	Rosca transversal
					2 =	Rosca cônica
				CENTENA:	---	reservado
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
29		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Comprimento da rosca em Z	
					0 =	Absoluto
					1 =	incremental
				DEZENA:	Comprimento da rosca em X	
					0 =	Absoluto, valor do eixo transversal em diâmetro
					1 =	Incremental, valor do eixo transversal em raio
					2 =	α
				CENTENA:	Avaliação de curso precedente/de entrada <_APP>	
					0 =	Curso precedente da rosca <_APP>
					1 =	Entrada da rosca = saída da rosca <_APP> = -<_ROP>
					2 =	Curso de entrada da rosca predefinido <_APP> = -<_APP>
				MILHAR:	Seleção da inclinação de penetração como ângulo ou largura	
					0 =	Ângulo de penetração<_IANG>
					1 =	Inclinação de penetração <_IFLANK>
				DEZENA DA MILHAR:	de uma entrada/de múltiplas entradas	
					0 =	Uma entrada (com desvio do ângulo inicial <_NSP>)
					1 =	múltiplas entradas
				CENTENA DA MILHAR:	Passo inicial <_GST>	
					0 =	Usinagem completa
					1 =	Iniciar a usinagem a partir deste passo
					2 =	Apenas usinar este passo
				UNIDADE DE MILHÃO:	Compensação de queda na rosca longitudinal	
					0 =	Altura de segmento da rosca boleada XS
					1 =	Raio da rosca boleada RS
30	XS / RS	<_S_XRS>	REAL	Compensação de queda na rosca longitudinal em ligação com <_AMODE>: UNIDADE DE MILHÃO		

Indicação

[1] Os parâmetros <_PITM>,<_PTAB> e<_PTABA> são utilizados apenas na seleção da rosca nas tabelas de roscas da tela de especificação.
Um acesso às tabelas de roscas através da definição de ciclo durante o processamento do ciclo não é possível.

20.1.31 CYCLE435 - definir o sistema de coordenadas de ajuste

Sintaxe

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1		<_T>	STRING[32]	Nome da ferramenta do rebolo		
2		<_DD>	INT	Número da lâmina do rebolo		
3		<S_TA>	STRING[32]	Ponto de referência do retificador - Nome do retificador		
4		<S_DA>	INT	Número da lâmina do retificador		
5		<S_AD>	REAL	Valor de retífica em diâmetro		
6		<S_AL>	REAL	Valor de retífica em plano		
7		<S_PVD>	REAL	Deslocamento do perfilamento em diâmetro		
8		<S_PVL>	REAL	Deslocamento do perfilamento em plano		
9		<S_PD>	REAL	Medida excedente de perfilamento em diâmetro		
10		<S_PL>	REAL	Medida excedente de perfilamento em plano		
11		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Ferramenta ativa ao final do ciclo	
					0 =	Retificador ativo
					1 =	Rebolo ativo

20.1.32 CYCLE495 - Perfilar

Sintaxe

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1		<_T>	STRING[20]	Nome da ferramenta do rebolo		
2		<_DD>	INT	Número da lâmina do rebolo		
3		<_SC>	REAL	Valor de retração para desviar de obstáculos, incremental		
4		<_F>	REAL	Avanço do perfilamento		
5		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Tipo de perfilamento	
					1 =	paralelo ao eixo
					2 =	paralelo ao contorno
				DEZENA:	Sentido de usinagem	
					0 =	por tração possível com as posições de corte 1 a 4
					1 =	por impacto possível com as posições de corte 1 a 4
					2 =	alternado possível com as posições de corte 1 a 8
					3 =	Início → Fim possível com as posições de corte 1 a 8
					4 =	Fim → Início possível com as posições de corte 1 a 8
				CENTENA:	Sentido de penetração	
					1 =	penetração X- com G18 ou Y- com G19
					2 =	penetração X+ com G18 ou Y+ com G19
					3 =	penetração Z- com G18 e com G19
					4 =	penetração Z+ com G18 e com G19
6		<_D>	REAL	Valor de retífica no caso do tipo de perfilamento paralelo ao eixo		
7		<_DX>	REAL	Valor de retífica X com G18 ou Y com G19 no caso do tipo de perfilamento paralelo ao contorno		
8		<_DZ>	REAL	Valor de retífica Z com G18 e G19 no caso do tipo de perfilamento paralelo ao contorno		
9		<S_PA>	REAL	Dimensão do perfilamento		
10		<S_N>	INT	Quantidade de cursos no programa de perfilamento		

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
11		<_DMODE>	INT	Modo de exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
12		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Seleção do perfilamento Novo/Prosseguir	
					1 =	Novo
					2 =	Continuação
				DEZENA:	Seleção da dimensão de perfilamento	
					0 =	Do contorno bruto até o ponto mais profundo do contorno
					1 =	Do contorno bruto até o ponto mais alto do contorno
13		<S_FW>	REAL	Ângulo de incidência do retificador		
14		<S_HW>	REAL	Ângulo de suporte do retificador		

20.1.33 CYCLE800 - Rotação

Sintaxe

```
CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>,
<_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)
```

Parâmetro

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1		<_FR>	INT	Modo de afastamento:	0 =	Nenhum afastamento
					1 =	Afastamento do eixo da máquina Z
					2 =	Afastamento do eixo de máquina Z e depois XY
					3 =	reservado
					4 =	Afastamento máximo no sentido da ferramenta
					5 =	Afastamento incremental no sentido da ferramenta

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
2		<_TC>	STRING[32]	Nome do bloco de dados da rotação:	""	"" (sem nome) se houver apenas 1 bloco de dados de rotação
					"0"	Cancelamento da seleção do bloco de dados de rotação (exclui o frame de rotação)

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
3		<_ST>	INT	Estado das transformações		
				UNIDADE:		
					0 =	Novo, o plano de rotação é apagado e recalculado com os parâmetros atuais
					1 =	Aditivo, o plano de rotação é adicionado ao plano de rotação ativo
				DEZENA:	Acompanhamento da ponta da ferramenta sim/não (somente está ativo se a função foi configurada na colocação em funcionamento da ROTAÇÃO)	
					0 =	Não acompanhar a ponta da ferramenta
					1 =	Acompanhar a ponta da ferramenta (TRAORI)
				CENTENA:	Posicionamento e alinhamento da ferramenta (a função é exibida na tela de especificação para ROTAÇÃO da ferramenta)	
					0 =	Não posicionar a ferramenta
					1 =	Posicionar a ferramenta (preferencialmente fresas de raios)
					2 =	Alinhar a **ferramenta de tornear** (se a cinemática do eixo B estiver configurada para a tecnologia de torno durante a colocação em funcionamento da rotação)
					3 =	Alinhar a **ferramenta de fresa** (se a cinemática do eixo B estiver configurada para a tecnologia de torno durante a colocação em funcionamento da rotação)
				MILHAR:	Parâmetro interno da rotação em JOG	
				DEZENA DA MILHAR:	Consultar o parâmetro de sentido <_DIR>	
					0 =	Rotação "sim"
					1 =	Rotação "não" sentido "negativo" [3]
					2 =	Rotação "não" sentido "positivo" [3]
				CENTENA DA MILHAR:	Consultar o parâmetro de sentido <_DIR>	
					0 =	Compatibilidade
					1 =	Seleção de direção "Menos" otimizada (somente para interface de usuário) [4]

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
					2 =	Seleção de direção "Mais" otimizada (somente para interface de usuário) [4]
4		<_MODE> [5]	INT	Modo de rotação: Interpretação do ângulo de rotação e a sequência de rotação (codificados por Bit!)		
				Bit: 7 6	0 0:	Ângulo de rotação por eixo -> consultar o parâmetro <_A>, <_B>, <_C>
					0 1:	Ângulo espacial -> consulte o parâmetro<_A>, <_B> [1]
					1 0:	Ângulo de projeção -> consulte o parâmetro <_A>, <_B>, <_C> [1]
					1 1:	Modo de rotação direto dos eixos rotativos -> consulte o parâmetro<_A>, <_B> [1]
				Bit: 5 4 3 2 1 0 (sem importância para ângulos espaciais!)	x x x x 0 1	1.Rotação _A em torno de X
					x x x x 1 0	1.Rotação _A em torno de Y
					x x x x 1 1	1.Rotação _A em torno de Z
					x x 0 1 x x	2.Rotação _B em torno de X
					x x 1 0 x x	2.Rotação _B em torno de Y
					x x 1 1 x x	2.Rotação _B em torno de Z
					0 1 x x x x	3.Rotação _C em torno de X
					1 0 x x x x	3.Rotação _C em torno de Y
					1 1 x x x x	3.Rotação _C em torno de Z
5	X0	<_X0>	REAL	Ponto de referência em X antes da rotação		
6	Y0	<_Y0>	REAL	Ponto de referência em Y antes da rotação		
7	Z0	<_Z0>	REAL	Ponto de referência em Z antes da rotação		
8	X(A)	<_A>	REAL	1ª Rotação conforme ajuste no parâmetro <_MODE>		
9	Y(B)	<_B>	REAL	2ª Rotação conforme ajuste no parâmetro <_MODE>		
10	Z(C)	<_C>	REAL	3ª Rotação conforme ajuste no parâmetro <_MODE>		
11	X1	<_X1>	REAL	Ponto de referência em X após a rotação		
12	Y1	<_Y1>	REAL	Ponto de referência em Y após a rotação		
13	Z1	<_Z1>	REAL	Ponto de referência em Z após a rotação		
14	- ou +	<_DIR>	INT	Disparar o movimento de deslocamento dos eixos rotativos (default = -1!)		
					-1 =	Posicionar o eixo rotativo 1 ou 2 no menor valor[2]
					+1 =	Posicionar o eixo rotativo 1 ou 2 no maior valor[2]
					0 =	Rotação não (apenas cálculo do frame de rotação) [1] [3]
15	FR	<_FR_I>	REAL	O valor (inc) do afastamento no sentido da ferramenta é incremental		

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
16		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Representação do valor beta ao alinhar a ferramenta	
					0 =	Valor
					1 =	Seta

Indicação

Se os seguintes parâmetros de transferência forem programados indiretamente (como parâmetros), a tela de especificação não será recompilada: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

[1)]Seleção possível, se a função foi configurada na colocação em funcionamento da ROTAÇÃO.

[2)] Seleção possível, se a referência de sentido para o eixo rotativo 1 ou 2 foi configurada na colocação em funcionamento da ROTAÇÃO.

sem campo de seleção para referência de sentido "não"

[3)] A seleção rotação "não" pode ser ocultada pelo dado SD 55221 Bit 0

Rotação "não" e sentido "negativo" corresponde a<_DIR> = 0 e _ST DEZENA DE MILHAR = 1

Rotação "não" e sentido "positivo" corresponde a<_DIR> = 0 e _ST DEZENA DE MILHAR = 2

[4)] A seleção do sentido do eixo rotativo 1 ou 2 também é realizada se o eixo rotativo estiver na posição de polo (valor de posição igual a zero) com a referência de sentido.

[5)] Exemplo de codificação: Rotação por eixos, sequência de rotação ZYX

Binário: 00011011; Decimal: 27

Os identificadores de eixo XYZ correspondem aos eixos geométricos do canal NC. Rotações em torno dos eixos XYZ podem ser executadas individualmente. Por ex. Sequência de rotação em torno de ZXZ não está permitida em uma chamada de CYCLE800

20.1.34 CYCLE801 - Grade ou quadro

Sintaxe

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros interno	Tipo de dado	Significado		
1	X0	<_SPCA>	REAL	Ponto de referência para modelo de posição (Grade/Quadro) no 1º eixo (abs)		
2	Y0	<_SPCO>	REAL	Ponto de referência para modelo de posição (Grade/Quadro) no 2º eixo (abs)		
3	α0	<_STA>	REAL	Ângulo de rotação básico (ângulo em relação ao 1º eixo)	< 0 =	Rotação no sentido horário
					> 0 =	Rotação no sentido anti-horário
4	L1	<_DIS1>	REAL	Distância das colunas (distância de posição do 1º eixo, especificar sem sinal)		
5	L2	<_DIS2>	REAL	Distância das linhas (distância de posição do 2º eixo, especificar sem sinal)		
6	N1	<_NUM1>	INT	Número de colunas		
7	N2	<_NUM2>	INT	Número de linhas		
8		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Modelo de posição	
					0 =	Grade
					1 =	Quadro
				DEZENA:	reservado	
				CENTENA:	reservado	
9		<_UMODE>	INT	reservado		
10	αX	<_ANG1>	REAL	Ângulo de esquadro para 1º eixo (inclinação das linhas em relação ao 1º eixo)		
					< 0 =	Medição no sentido horário (0 a -90 graus)
					> 0 =	Medição no sentido anti-horário (0 a 90 graus)
11	αY	<_ANG2>	REAL	Ângulo de esquadro para 2º eixo (inclinação das colunas em relação ao 2º eixo)		
					< 0 =	Medição no sentido horário (0 a -90 graus)
					> 0 =	Medição no sentido anti-horário (0 a 90 graus)

Nº.	Tela de parâmetros	Parâmetros interno	Tipo de dado	Significado		
12		<_HIDE>	STRING [200]	Posições suprimidas • máx. 198 caracteres • Indicação do número de posição subsequente, p. ex. "1,3" (as posições 1 e 3 não são executadas)		
13		<_NSP>	INT	reservado		
14		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)

20.1.35 CYCLE802 - Posição definida pelo usuário

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros interno	Tipo de dado	Significado		
1		<_XA>	INT	Alternativa para todas posições X (valor decimal de 9 dígitos) Número de dígitos: 876543210 (O dígito corresponde à posição de furação Xn)		
				Número de dígitos:	1 =	Absoluto (1ª posição programada é sempre absoluta)
					2 =	incremental
2		<_YA>	INT	Alternativa para todas posições Y (valor decimal de 9 dígitos) Número de dígitos: 876543210 (O dígito corresponde à posição de furação Yn)		
				Número de dígitos:	1 =	Absoluto (1ª posição programada é sempre absoluta)
					2 =	incremental
3	X0	<_X0>	REAL	1. Posição X		
4	Y0	<_Y0>	REAL	1. Posição Y		
5	X1	<_X1>	REAL	2. Posição X		

Nº.	Tela de parâmetros	Parâmetros interno	Tipo de dado	Significado
6	Y1	<_Y1>	REAL	2. Posição Y
7	X2	<_X2>	REAL	3. Posição X
8	Y2	<_Y2>	REAL	3. Posição Y
9	X3	<_X3>	REAL	4. Posição X
10	Y3	<_Y3>	REAL	4. Posição Y
11	X4	<_X4>	REAL	5. Posição X
12	Y4	<_Y4>	REAL	5. Posição Y
13	X5	<_X5>	REAL	6. Posição X
14	Y5	<_Y5>	REAL	6. Posição Y
15	X6	<_X6>	REAL	7. Posição X
16	Y6	<_Y6>	REAL	7. Posição Y
17	X7	<_X7>	REAL	8. Posição X
18	Y7	<_Y7>	REAL	8. Posição Y
19	X8	<_X8>	REAL	9. Posição X
20	Y8	<_Y8>	REAL	9. Posição Y

Nº.	Tela de parâmetros	Parâmetros interno	Tipo de dado	Significado		
21		<_VARI>	INT	Usinagem		
				CENTENA:	(apenas para chamada do Jobshop) (por enquanto apenas 0 e 2 avaliados)	
					0 =	Sem travamento do fuso
					1 =	Travar o fuso apenas no caso de imersão perpendicular com G00 ou G01
					2 =	Travar o fuso durante toda a usinagem
				MILHAR:	reservado	
				DEZENA DA MILHAR:	Modelo de posição com/sem eixo rotativo – combinação de eixos (com <_VARI> CENTENA DE MILHAR)	
					0 =	XY (apenas XY sem eixo rotativo, compatibilidade)
					1 =	X,Y ou Z e eixo rotativo: XA, YB, ZC (1 eixo rotativo com eixo geométrico, em torno do qual o eixo rotativo gira)
					2 =	XY e eixo rotativo: XYA, XYB, XYC (1 eixo rotativo 1º e 2º eixo geométrico, sem TRACYL)
				CENTENA DA MILHAR:	Eixo rotativo	
					0 =	Sem eixo rotativo (apenas XY, compatibilidade)
					1 =	Eixo A (eixo rotativo em torno de X)
					2 =	Eixo B (eixo rotativo em torno de Y)
					3 =	Eixo C (eixo rotativo em torno de Z)
				DEZENA DE MILHÃO + UNIDADE DE MILHÃO:	modelo de posição com eixo circular – Offset (com mais eixos circulares em torno de uma mesmo eixo; se o índice for grande demais, então 1º eixo)	
					00 =	1. A, B ou C ou com compatibilidade
					01 =	2. Eixo A, B ou C
					...	
					19 =	20. Eixo A, B ou C
22		<_UMODE>	INT	Seleção do fuso a ser fixado (apenas para chamada do Jobshop) (chamada do ciclo aplicado CUST_TECHCYC)		
					3 =	Fixar/soltar o fuso principal
					23 =	Fixar/soltar o contra-fuso

Nº.	Tela de parâmetros	Parâmetros interno	Tipo de dado	Significado		
23		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
24		<S_ABA>	INT	Alternativa para todas posições AB (valor decimal de 9 dígitos) Número de dígitos: 876543210 (O dígito corresponde à posição ABn)		
				Número de dígitos:	1 =	Absoluto (1ª posição programada é sempre absoluta)
					2 =	incremental
25	A0	<S_AB0>	REAL	1. Posição do eixo rotativo em modelo de posição com eixo rotativo (em ligação com <_VARI>)		
26	A1	<S_AB1>	REAL	2. Posição do eixo circular em modelo de posição com eixo rotativo		
27	A2	<S_AB2>	REAL	3. Posição do eixo circular em modelo de posição com eixo rotativo		
28	A3	<S_AB3>	REAL	4. Posição do eixo circular em modelo de posição com eixo rotativo		
29	A4	<S_AB4>	REAL	5. Posição do eixo circular em modelo de posição com eixo rotativo		
30	A5	<S_AB5>	REAL	6. Posição do eixo circular em modelo de posição com eixo rotativo		
31	A6	<S_AB6>	REAL	7. Posição do eixo circular em modelo de posição com eixo rotativo		
32	A7	<S_AB7>	REAL	8. Posição do eixo circular em modelo de posição com eixo rotativo		
33	A8	<S_AB8>	REAL	9. Posição do eixo circular em modelo de posição com eixo rotativo		

Indicação

As posições desnecessárias dos parâmetros X1/Y1/A1 até X8/Y8/A8 podem ser descartadas. Os valores alternativos para<_XA>,<_YA> e <S_ABA>, entretanto, devem ser especificados totalmente para todas as 9 posições.

Com modelos de posição XA, YB ouZC (um eixo geométrico e eixo circular), o eixo da etapa de usinagem que não se deslocará sobre o modelo de posição (com G17 e XA é o Y) deve ser posicionado antes da chamada de ciclo.

20.1.36 CYCLE830 - Furação de furo profundo 2

Sintaxe

```
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>)
```

Parâmetro

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<RFP>	REAL	Ponto de referência (abs)		
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, sem sinal)		
4	Z1	<_DP>	REAL	Profundidade final de furação abs/inc (consultar<_AMODE> UM)		
5	D	<FDEP>	REAL	1. Profundidade da furação para quebra / remoção de cavacos absoluta ou incremental relacionada ao ponto de referência na furação/sem furação ou relacionada à profundidade da furação piloto (ver <_AMODE> ZEHNTAUSENDER)		
6	DF	<_DAM>	REAL	Quantia/valor em porcentagem para cada penetração seguinte valor de decréscimo ou porcentagem (consultar<_AMODE> CENTENA DE MILHAR)		
7	DTB	<DTB>	REAL	Tempo de permanência em cada profundidade de furação (consultar<_AMODE> DEZENA)		
8	DTS	<DTS>	REAL	Tempo de permanência ao remover os cavacos no ponto inicial (consultar<_AMODE> CENTENA)		
9	FD1	<FRF>	REAL	Porcentagem para o avanço na primeira penetração (consultar<_AMODE> DEZENA DE MILHÕES)		
10		<VARI>	INT	Usinagem		
				UNIDADE:	Quebra de cavacos / remoção de cavacos	
					0 =	em um corte
					1 =	Quebra de cavacos
					2 =	Remoção de cavacos
					3 =	Quebra de cavacos e remoção de cavacos
				DEZENA:	Retrocesso na remoção de cavacos	
					0 =	Na profundidade de furação piloto
					1 =	Na distância de segurança
				CENTENA:	entalhe maleável	
					0 =	não
					1 =	sim
				MILHAR:	Perfuração	
					0 =	não
					1 =	sim
				DEZENA DA MILHAR:	Furação/furação piloto	
					0 =	sem furação
					1 =	com furação
					2 =	com furação piloto
				CENTENA DA MILHAR:	Retrocesso	
					0 =	Na profundidade de furação piloto
					1 =	até o plano de retrocesso

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
11	V1	<_MDEP>	REAL	Penetração mínima incremental (apenas para valor em porcentagem para decréscimo)		
12	V2	<_VRT>	REAL	Valor de retrocesso depois de cada usinagem incremental (somente para quebra de cavacos)		
					0 =	Valor padrão 1 mm
					> 0 =	Valor de retrocesso variável
13	DT	<_DTD>	REAL	Tempo de permanência na profundidade final de furação (consultar<_AMODE> MILHAR)		
14	V3	<_DIS1>	REAL	Distância de parada prévia incremental (somente para remoção de cavacos), ver <_AMODE> MILHÃO)		
15	FP	<S_FP>	REAL	Avanço para inserção na furação piloto como valor ou em % (em conexão com<S_AMODE2> CENTENA)		
16		<S_SDAC2>	INT	Aproximação com o sentido de giro do fuso		
					3 =	M3
					4 =	M4
					5 =	M5 (Default)
17	SP	<S_SV2>	REAL	Aproximação com o	Rotação constante do fuso (consultar<S_AMODE2> DEZ MILHÕES)	
	V4				Velocidade de corte constante	
				número de giros do furo em % no número de giros da furação		
18	F	<S_FB>	REAL	Avanço de furação (consultar<S_AMODE2> UM)		
19		<_SDAC>	REAL	Furação no sentido de giro do fuso		
					3 =	M3
					4 =	M4
20	S	<_SV1>	REAL	Furação com	Rotação constante do fuso (consultar<S_AMODE2> UM MILHÃO)	
	V5				Velocidade de corte constante	
21	SPOS	<S_SPOS>	REAL	Posição do fuso,apenas quando partindo com M5		
22	ZA	<S_ZA>	REAL	Profundidade da furação incremental relacionada ao ponto de referência ou absoluto (consultar<S_AMODE3> UM)		
23	FA	<S_FA>	REAL	Avanço de furação como valor ou em % (em conexão com<S_AMODE2> DEZENA)		
24	ZP	<S_ZP>	REAL	Furação piloto incremental relacionada ao ponto de referência, absoluto ou fator do diâmetro da furação (consultar<S_AMODE3> DEZENA)		
25	FS	<S_FS>	REAL	Avanço do entalhe como valor ou em % (em conexão com<S_AMODE2> MILHAR)		
26	ZS1	<S_ZS1>	REAL	Profundidade de cada entalhe com avanço constante (inc)		
27	ZS2	<S_ZS2>	REAL	Profundidade de cada entalhe para o aumento do avanço (inc)		
28	N	<S_N>	INT	Número de cursos de quebra de cavacos antes de cada remoção de cavacos		
29	ZD	<S_ZD>	REAL	Profundidade da furação restante incremental relacionada à profundidade da furação final ou absoluto (em conexão com<S_AMODE3> CENTENA)		
30	FD	<S_FD>	REAL	Avanço de furação restante como valor ou em % (em conexão com<S_AMODE2> DEZENA DE MILHAR)		

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
31	FR	<S_FR>	REAL	Avanço do retrocesso (em conexão com<S_AMODE2> CENTENA DE M ILHAR)		
32		<S_SDAC3>	INT	Retrocesso no sentido de giro do fuso		
					3 =	M3
					4 =	M4
					5 =	M5
33	SR	<S_SV3>	REAL	Retrocesso com o	Rotação constante do fuso (consultar<S_AMODE2> CEM MILHÕES)	
	V6				Velocidade de corte constante	
				número de giros do furo em % no número de giros da furação		
34	Refrigeração um	<S_CON>	STRING[10]	Refrigeração ligada, comando M ou chamada da sub-rotina		
35	Refrigeração de	<S_COFF>	STRING[10]	Refrigeração de comando M ou chamada da sub-rotina		
36		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	Profundidade de furação relativa à ponta/haste	
					0 =	Ponta
					1 =	Haste
37		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	reservado	
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
38		<_AMODE>	INT	Modo alternativo 1		
				UNIDADE:	Profundidade de furação = profundidade final de furação Z1 abs/inc	
					0 =	incremental
					1 =	Absoluto
				DEZENA:	Tempo de espera em cada profundidade de furação DTB em segundos/rotações	
					0 =	Em segundos
					1 =	Em rotações
				CENTENA:	Tempo de espera para remoção de cavacos em segundos/rotações	
					0 =	Em segundos
					1 =	Em rotações
				MILHAR:	Tempo de espera na profundidade final de furação DT em segundos/rotações	
					0 =	Em segundos
					1 =	Em rotações
				DEZENA DA MILHAR:	1. Profundidade de perfuração D abs/incr	
					0 =	incremental
					1 =	Absoluto
				CENTENA DA MILHAR:	Quantia/valor em porcentagem do DF para cada penetração seguinte (decréscimo)	
					0 =	Valor
					1 =	Porcentagem (0,001 a 100 %)
				UNIDADE DE MILHÃO:	Distância de parada prévia V3 automática/manual	
					0 =	Automático (é calculado no ciclo)
					1 =	Manual (valor programado)

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
39		<S_AMODE2>	INT	Modo alternativo 2		
				UNIDADE:	UNIDADE: Avanço de furação F	
					0 =	F/min
					1 =	F/U
				DEZENA:	Avaliação do avanço da furação FA	
					0 =	Em % do avanço da furação
					1 =	F/min
					2 =	F/U
				CENTENA:	Avaliação do avanço para a inserção em furação piloto FP	
					0 =	Em % do avanço da furação
					1 =	F/min
					2 =	F/U
				MILHAR:	Avaliação do avanço do entalhe FS	
					0 =	Em % do avanço da furação
					1 =	F/min
					2 =	F/U
				DEZENA DA MILHAR:	Avaliação do avanço da perfuração FD	
					0 =	Em % do avanço da furação
					1 =	F/min
					2 =	F/U
				CENTENA DA MILHAR:	Avanço do retrocesso FR	
					0 =	F/min
					1 =	Avanço rápido
				UNIDADE DE MILHÃO:	Furação - Rotação do fuso/velocidade de corte (_SV5)	
					0 =	Rotação constante do fuso
					1 =	Velocidade de corte constante
				DEZENA DE MILHÃO:	Aproximação com a rotação do fuso/velocidade de corte (SP/V4)	
					0 =	Rotação constante do fuso
					1 =	Velocidade de corte constante
					2 =	número de giros do furo em % no número de giros da furação
				CEM MILHÕES	Retrocesso - Rotação do fuso/velocidade de corte (SR/V6)	
					0 =	Rotação constante do fuso
					1 =	Velocidade de corte constante
					2 =	número de giros do furo em % no número de giros da furação

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
40		<S_AMODE3>	INT	Modo alternativo 3		
				UNIDADE:	Profundidade da furação ZA abs/inc	
					0 =	incremental
					1 =	Absoluto
				DEZENA:	Profundidade da Furação Piloto ZP	
					0 =	incremental
					1 =	Absoluto
					2 =	Fator do diâmetro da furação
				CENTENA:	Profundidade da furação restante ZD inc/abs	
					0 =	incremental
					1 =	Absoluto
41	ZPV	<S_ZPV>	REAL	Distância de parada prévia incremental relativa à profundidade do furo piloto		

20.1.37 CYCLE832 - High Speed Settings

Sintaxe

```
CYCLE832(<S_TOL>, <S_TOLM>, <S_OTOL>)
```

Indicação

O CYCLE832 poupa o fabricante de máquinas de tarefas de otimização durante a colocação em funcionamento da máquina. Isto envolve a otimização dos eixos utilizados na usinagem e os ajustes da NCU (controle feedforward, limitação de solavancos, etc.).

Parâmetro

Nº	Máscara de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	Tolerância	<S_TOL>	REAL	Tolerância do contorno A tolerância do contorno corresponde à tolerância de eixo dos eixos geométricos.		
2		<S_TOLM>	INT	Tipo de usinagem (tecnologia)		
				UNIDADE:		
					0 =	Desativação
					1 =	Acabamento (Finish)
					2 =	Pré-acabamento (Semifinish)
					3 =	Desbaste (Rough)
				DEZENA:		
					0 =	Compatibilidade[1] ou nenhuma tolerância de orientação
					1 =	Tolerância de orientação no parâmetro <S_OTOL>
				CENTENA ... CENTENA-MILHAR	Ocupado devido à Compatibilidade	
				UNIDADE DE MILHÃO:		
					0 =	Compatibilidade. Automaticamente a melhor função de construção de formas disponível será escolhida: • opção Top Surface não ativa: ⇒ Advanced Surface • opção Top Surface ativa: ⇒ Top Surface com alisamento
					1 =	Top Surface sem alisamento
					2 =	Top Surface com alisamento
3	Tolerância ORI	<S_OTOL>	REAL	Tolerância de orientação ou indicador de versão CYCLE832 parâmetro de tolerância para a orientação da ferramenta. É necessário no processamento de um programa de usinagem de alta velocidade de máquinas com transformação de orientação dinâmica. ex. Usinagem de 5 eixos). O parâmetro<S_OTOL> **deve** ser programado. Isto também se aplica para as aplicações em máquinas de 3 eixos, nos programas sem orientação da ferramenta (<S_OTOL> = 1).		

[1] tolerância de orientação derivada da tolerância de contorno multiplicada com o fator do ciclo dos dados SD55441 até SD55443.
Literatura:
manual de colocação em funcionamento; SINUMERIK Operate (IM9), Capítulo"Configurar Funktion High Speed Setting (CYCLE832) "

Entrada de texto claro

para melhorar a leitura da chamada do ciclo, pode ser implantado no texto também o parâmetro <S_TOLM> (tipo de edição). Textos claros são independentes de linguagem. As seguintes entradas são permitidas:

_OFF	para	0	Desativação
_FINISH	para	1	Acabamento
_SEMIFIN	para	2	Pré-acabamento
_ROUGH	para	3	Desbaste
_ORI_FINISH	para	11	Acabamento com entrada de uma tolerância de orientação
_ORI_SEMIFIN	para	12	Pré-acabamento com entrada de uma tolerância de orientação
_ORI_ROUGH	para	13	Desbaste com entrada de uma tolerância de orientação
_TOP_SURFACE_SMOOTH_OFF	para	1000000	Top Surface sem alisamento
_TOP_SURFACE_SMOOTH_ON	para	2000000	Top Surface com alisamento

Para a entrada de texto no Top Surface, os textos claros serão combinados assim como no seguinte exemplo:

```
CYCLE832(0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

Indicação

Os textos devem se apoiar nos nomes da função do grupo G 59 (modo dinâmico para interpolação da trajetória). Com estes textos, as máquinas de 3 eixos e as com transformação orientada de multieixos (TRAORI) serão claramente separadas na aplicação.

Desativação do CYCLE832

Com a desativação do CYCLE832, o parâmetro<S_TOL> deve ser transmitido com zero.

Exemplo: `CYCLE832(0,0,1)`

A sintaxe `CYCLE832()` também é permitida para a desativação do CYCLE832.

Exemplos

Exemplo 1: CYCLE832 na máquina de 3 eixos sem transformação da orientação

a) chamada de ciclos com entrada de texto claro

Código de programa	Comentário
G710	; Sistema de medida é métrico.
CYCLE832(0.004,_FINISH,1)	; Chamada CYCLE832 com: tolerância de contorno = 0,004 mm, tipo de usinagem: Acabamento

Código de programa	Comentário
...	; Execução de um programa de usinagem de alta velocidade

b) chamada de ciclos sem entrada de texto claro

Código de programa	Comentário
G710	; assim ou
CYCLE832(0.004,1,1)	; assim ou
...	; assim ou

Exemplo 2: CYCLE832 na máquina de 5 eixos com transformação da orientação

a) chamada de ciclos e desativação com entrada de texto claro

Código de programa	Comentário
G710	; Sistema de medida é métrico.
TRAORI	; Ativar transformação de orientação.
CYCLE832(0.3,_ORI_ROUGH,0.8)	; Chamada CYCLE832 com: tolerância de contorno = 0,3 mm, tipo de usinagem: Desbaste com entrada de uma tolerância de orientação, tolerância de orientação = 0,8 Graus
...	; Execução de um programa de usinagem de alta velocidade
CYCLE832(0,_OFF,1)	; Tolerância de contorno = 0, tipo de usinagem: Desativação CYCLE832, tolerância de orientação = 0 Graus

b) chamada de ciclos e desativação sem entrada de texto claro

Código de programa	Comentário
G710	; assim ou
TRAORI	; assim ou
CYCLE832(0.3,13,0.8)	; assim ou
...	; assim ou
CYCLE832(0,0,1)	; assim ou

20.1.38 CYCLE840 - Rosqueamento com macho com mandril de compensação

Sintaxe

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>,
<_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado			
1	RP	<RTP>	REAL	Plano de retrocesso (abs)			
2	Z0	<RFP>	REAL	Ponto de referência (abs)			
3	SC	<SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)			
4	Z1	<DP>	REAL	Profundidade de furação (abs), consultar <_AMODE>			
5	Z1	<DPR>	REAL	Profundidade de furação (ink), consultar <_AMODE>			
6	DT	<DTB>	REAL	Tempo de permanência na profundidade de furação/na distância de segurança após o retrocesso em segundos, consultar <ENC>			
7		<SDR>	INT	Sentido de rotação para o retrocesso			
8	SDE	<SDAC>	INT	Sentido de rotação após fim do ciclo			
9		<ENC>	INT	Rosqueamento com encoder de fuso (G33)/rosqueamento sem encoder de fuso(G63)			
					0 =	com encoder de fuso	- Passo de <MPIT>/<PIT> - sem DT
					20 =	com encoder de fuso	- Passo de <MPIT>/<PIT> - com DT após o retrocesso para a distância de segurança
					11 =	sem encoder do fuso	- Passo de <MPIT>/<PIT> - com DT na profundidade de furação
					1 =	sem encoder do fuso	- Passo do avanço programado - com DT na profundidade de furação (avanço = Rotação Passo)
10		<MPIT>	REAL	Tamanho de rosca somente para "ISO métrico" (o passo é calculado internamente durante o processamento) Faixa de valores: 3 a 48 (para M3 até M48), alternativa para <PIT>			
11		<PIT>	REAL	Passo de rosca como valor, consultar a unidade de medida <_PITA> Faixa de valores: > 0, alternativa para o MPIT			
12		<_AXN>	INT	Eixo de furação	0 =	3. Eixo geométrico	
					1 =	1. Eixo geométrico	
					2 =	2. Eixo geométrico	
					≥ 3 =	3. Eixo geométrico	

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado			
13		<_PITA>	INT	Unidade de medida do passo de rosca (avaliação <PIT> e <MPIT>)			
					0 =	Passo em mm	- Avaliação <MPIT>/<PIT>
					1 =	Passo em mm	- Avaliação <PIT>
					2 =	Passo em TPI	- Avaliação <PIT> (passos de rosca por polegada)
					3 =	Passo em polegadas	- Avaliação <PIT>
					4 =	MÓDULO	- Avaliação <PIT>
14		<_TECHNO>	INT	Tecnologia[1]			
				UNIDADE:	Comportamento de parada exata		
					0 =	Comportamento de parada exata como o ativado antes da chamada do ciclo	
					1 =	Parada exata G601	
					2 =	Parada exata G602	
					3 =	Parada exata G603	
				DEZENA:	Controle feedforward		
					0 =	com/sem controle feedforward como o ativado antes da chamada do ciclo	
					1 =	com Feedforward FFWON	
					2 =	sem Feedforward FFWOF	
15		<_PITM>	STRING[15]	String como marcador para entrada do passo da rosca[2]			
16		<_PTAB>	STRING[5]	String para tabela de roscas ("", "ISO", "BSW", "BSP", "UNC")[2]			
17		<_PTABA>	STRING[20]	String para seleção na tabela de roscas (p. ex. "M 10", "M 12", ...)[2]			
18		<_GMODE>	INT	reservado			

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
19		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:	Modo de compatibilidade (apenas para tela de especificação para recompilação), se o dado MD 52216 Bit0 = 1[1)]	
					0 =	Os parâmetros de tecnologia são exibidos (compatibilidade): Sob efeito dos parâmetros TECHNO
					1 =	Os parâmetros de tecnologia não são exibidos: Sob efeito da tecnologia "como o programado antes da chamada do ciclo"
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples
20		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade de furação Z1 (abs/inc)	
					0 =	Compatibilidade, a partir da programação <DP>/<DPR>
					1 =	incremental
					2 =	Absoluto

[1)] Os campos de tecnologia podem ser ocultados em função do dado de ajuste SD52216 MCS_FUNCTION_MASK_DRILL

[2)] Os parâmetros 15, 16 e 17 somente são utilizados na seleção de roscas nas tabelas de roscas da tela de especificação. Um acesso às tabelas de roscas através da definição de ciclo durante o processamento do ciclo não é possível!

20.1.39 CYCLE899 - Fresar calha

Sintaxe

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>, <_CDIR>,
<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>, <_ZFS>)
```

Parâmetros

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	RP	<_RTP>	REAL	Plano de retrocesso (abs)		
2	Z0	<_RFP>	REAL	Ponto de referência do eixo da ferramenta (abs)		
3	SC	<_SDIS>	REAL	Distância de segurança (adicionada ao ponto de referência, especificar sem sinal)		
4	Z1	<_DP>	REAL	Profundidade da ranhura (abs/inc), consulte <_AMODE>		
5	L	<_LENG>	REAL	Comprimento da ranhura (inc)		
6	W	<_WID>	REAL	Largura da ranhura (inc)		
7	X0	<_PA>	REAL	Ponto de referência/posição de partida do 1º eixo (abs)		
8	Y0	<_PO>	REAL	Ponto de referência/posição de partida do 2º eixo (abs)		
9	α0	<_STA>	REAL	Ângulo de giro para o 1º eixo		
10	DZ	<_MID>	REAL	profundidade de penetração máxima (inc), somente para fresamento vortical		
11	DXY	<_MIDA>	REAL	Penetração máxima no plano, consulte <_AMODE>		
12	UXY	<_FAL>	REAL	Sobremetal de acabamento do plano		
13	UZ	<_FALD>	REAL	Sobremetal de acabamento da profundidade		
14	F	<_FFP1>	REAL	Avanço		
15		<_CDIR>	INT	Sentido de fresamento		
				UNIDADE:		
					0 =	Concordante
					1 =	Discordante
					4 =	Alternado
16		<_VARI>	INT	Usinagem		
				UNIDADE:		
					1 =	Desbaste
					2 =	Acabamento
					3 =	Acabamento da base
					4 =	Acabamento da borda
					5 =	Pré-acabamento
					6 =	Chanframento
				DEZENA:	reservado	
				CENTENA:	reservado	
				MILHAR:		
					1 =	Fresamento de vórtice
					2 =	Fresamento por imersão

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
17		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem/apenas cálculo do ponto de partida	
					1 =	Usinagem normal
				MILHAR:	Dimensionamento através do centro/ aresta	
					0 =	Dimensionamento através do centro
					1 =	Dimensionamento através da aresta "esquerda" ("-" para sentido do 1º eixo)
					2 =	Dimensionamento através da aresta "direita" ("+" para sentido do 1º eixo)
18		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	---	reservado
				CENTENA:	---	reservado
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples
19		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Profundidade da ranhura Z1	
					0 =	Absoluto
					1 =	incremental
				DEZENA:	Unidade para penetração no plano (<_MIDA>) DXY	
					0 =	mm
					1 =	% do diâmetro da ferramenta
				CENTENA:	Profundidade de imersão no chanfra-mento ZFS	
					0 =	Absoluto
					1 =	incremental
20		<_UMODE>	INT	reservado		

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
21	FS	<_FS>	REAL	Largura do chanfro (inc)
22	ZFS	<_ZFS>	REAL	Profundidade de imersão (ponta da ferramenta) no chanframento (abs/inc), consulte <_AMODE>

20.1.40 CYCLE930 - Canal

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	X0	<_SPD>	REAL	Ponto de referência no eixo transversal (sempre em diâmetro)
2	Z0	<_SPL>	REAL	Ponto de referência no eixo longitudinal
3	B1	<_WIDG>	REAL	Largura de canal inferior
4	B2	<_WIDG2>	REAL	Largura do canal superior (apenas para a interface)
5	T1	<_DIAG>	REAL	Profundidade de canal no ponto de referência, com abs e usinagem longitudinal = diâmetro, caso contrário inc
6	T2	<_DIAG2>	REAL	Profundidade do canal oposta ao ponto de referência (apenas para a interface), com abs e usinagem longitudinal = diâmetro, caso contrário inc
7	α0	<_STA>	REAL	Ângulo da inclinação (-180 ≤<_STA> ≤ 180)
8	α1	<_ANG1>	REAL	Ângulo do flanco 1 (0 ≤<_ANG1> < 90) no lado do canal determinado pelo ponto de referência
9	α2	<_ANG2>	REAL	Ângulo do flanco 2 (0 ≤<_ANG2> < 90) oposto ao ponto de referência
10	R1/FS1	<_RCO1>	REAL	Raio de arredondamento ou largura do chanfro 1, externo no ponto de referência
11	R2/FS2	<_RCI1>	REAL	Raio de arredondamento ou largura do chanfro 2, interno no ponto de referência
12	R3/FS3	<_RCI2>	REAL	Raio de arredondamento ou largura do chanfro 3, interno e oposto ao ponto de referência
13	R4/FS4	<_RCO2>	REAL	Raio de arredondamento ou largura do chanfro 4, externo e oposto ao ponto de referência
14	U	<_FAL>	REAL	Sobremetal de acabamento em X e Z, consulte<_VARI> (DEZENA DE MILHAR) (especificar sem sinal prévio)

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
15	D	<_IDEP1>	REAL	Profundidade máxima de penetração na imersão (sem especificar o sinal)		
					0 =	1. Corte direto até a profundidade total
					> 0 =	1. Corte <_IDEP1>, 2. corte 2 ·<_IDEP1> etc.
16	SC	<_SDIS>	REAL	Distância de segurança (especificar sem sinal)		
17		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	reservado	
				DEZENA:	Tecnologias de usinagem	
					1 =	Desbaste
					2 =	Acabamento
					3 =	Desbaste e acabamento
				CENTENA:	Posição longitudinal/transversal externa/interna +Z/+Z e +X/-X	
					1 =	Longitudinal/externo +Z
					2 =	Plano/interno -X
					3 =	Longitudinal/interno +Z
					4 =	Plano/interno +X
					5 =	Longitudinal/externo -Z
					6 =	Plano/externo -X
					7 =	Longitudinal/interno -Z
					8 =	Plano/externo +X
				MILHAR:	Posição do ponto de referência	
					0 =	Ponto de referência superior
					1 =	Ponto de referência inferior
				DEZENA DA MILHAR:	Definição de como atuam os sobremetais de acabamento	
					0 =	Sobremetal de acabamento U paralelo ao contorno
					1 =	Sobremetais de acabamento UX e UZ separados
18		<_DN>	INT	Número D para o 2º corte da ferramenta		
					> 0 =	Número D para correção de ferramenta do 2º corte da ferramenta para abrir canais
					0 =	Nenhum 2º corte programado
19	N	<_NUM>	INT	Número de canais (0 = 1 canal)		
20	DP	<_DBH>	REAL	Distância dos canais (necessário apenas, se<_NUM> > 1)		
21	F	<_FF1>	REAL	Avanço		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
22		<_NR>	INT	Identificação para a forma do canal, que corresponde às softkeys verticais para seleção da forma		
					0 =	Flancos de 90° sem chanfros/arredondamentos
					1 =	Flancos inclinados com chanfros/arredondamentos (sem α0)
					2 =	Como 1, mas sobre cone (com α0)
23	UX	<_FALX>	REAL	Sobremetal de acabamento no eixo X, consulte<_VARI> (DEZENA DE MILHAR) (especificar sem sinal prévio)		
24	UZ	<_FALZ>	REAL	Sobremetal de acabamento no eixo Z, consulte<_VARI> (DEZENA DE MILHAR) (especificar sem sinal prévio)		
25		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
26		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Dimensionamento da profundidade (apenas para a interface)	
					0 =	No ponto de referência
					1 =	Oposto ao ponto de referência
				DEZENA:	Profundidade	
					0 =	Absoluto
					1 =	incremental
				CENTENA:	Dimensionamento da largura (apenas para a interface)	
					0 =	No diâmetro externo (superior)
					1 =	No diâmetro interno (inferior)
				MILHAR:	Raio/chanfro 1 (<_RCO1>)	
					0 =	Raio
					1 =	Chanfro
				DEZENA DA MILHAR:	Raio/chanfro 2 (<_RCI1>)	
					0 =	Raio
					1 =	Chanfro
				CENTENA DA MILHAR:	Raio/chanfro 3 (<_RCI2>)	
					0 =	Raio
					1 =	Chanfro
				UNIDADE DE MILHÃO:	Raio/chanfro 4 (<_RCO2>)	
					0 =	Raio
					1 =	Chanfro

20.1.41 CYCLE940 - Formas de alívio

Com o ciclo CYCLE940 é possível programar vários alívios diferentes. Estes, em parte, têm grandes diferenças na parametrização.

As colunas adicionais na tabela mostram quais parâmetros são necessários para cada forma de alívio. Elas correspondem às softkeys verticais de seleção na tela dos ciclos:

- E: Alívio forma E
- F: Alívio forma F
- A-D: Alívio de rosca DIN (formas A-D)
- T: Alívio de rosca (livre definição da forma)

Sintaxe

```
CYCLE940(<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>, <_VARI>,
<_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>, <_FAL>,
<_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>, <_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Progr. na forma				Significado		
				E	F	A-D	T			
1	X0	<_SPD>	REAL	x	x	x	x	Ponto de referência no eixo transversal (sempre em diâmetro)		
2	Z0	<_SPL>	REAL	x	x	x	x	Ponto de referência do eixo longitudinal (abs)		
3	FORM	<_FORM>	CHAR	x	x	x	x	Forma do alívio (letras maiúsculas, p. ex. "T") Seleção de qual tabela que os valores do alívio devem ser usados		
									A =	externo, referência DIN76, A = normal
									B =	externo, referência DIN76, B = curto
									C =	interno, referência DIN76, C = normal
									D =	interno, referência DIN76, D = curto
									E =	Referência DIN509
									F =	Referência DIN509
									T=	Forma de alívio
4	LAGE	<_LAGE>	INT	x	x	x	x	Posição do alívio (paralelo a Z)	0 =	Externo +Z: ____|
									1 =	Externo -Z: ____|
									2 =	Interno +Z: /-----|
									3 =	Interno -Z: |-----\
5	SC	<_SDIS>		x	x	x	x	Distância de segurança (inc)		
6	F	<_FFP>		x	x	x	x	Avanço de usinagem (mm/rot.)		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Progr. na forma				Significado			
7		<_VARI>	INT	-	-	x	x	Tipo de usinagem			
								UNIDADE:	Usinagem		
									1 =	Desbaste	
									2 =	Acabamento	
									3 =	Desbaste + acabamento	
								DEZENA:	Estratégia de usinagem		
									0 =	paralelo ao contorno	
									1 =	longitudinal	
				Os alívios de forma E e F sempre são usinados em um passe como no acabamento.							
8	X1	<_EPD>		x	x	-	-	Sobremetal X (abs/inc), consulte <_AMODE>			
				-	-	-	x	Profundidade do alívio (abs/inc), consulte <_AMODE>			
9	Z1	<_EPL>		-	x	-	-	Sobremetal em Z			
				-	-	-	x	Largura do alívio (abs/inc), consulte <_AMODE>			
10	R1	<_R1>		-	-	-	x	Raio de arredondamento nas inclinações			
11	R2	<_R2>		-	-	-	x	Raio de arredondamento no canto			
12	α	<_STA>		-	-	x	x	Ângulo de imersão			
13	VX	<_VRT>		x	x	-	-	Avanço transversal X (abs/inc), consulte <_AMODE>			
				-	-	x	x	Avanço transversal X no acabamento (abs/inc), consulte <_AMODE>			
14	D	<_MID>		-	-	x	x	Penetração em profundidade			
15	U	<_FAL>		-	-	x	x	Sobremetal de acabamento paralelo ao contorno, consulte <_AMODE>			
16	UX	<_FALX>		-	-	x	x	Sobremetal de acabamento em X			
17	UZ	<_FALZ>		-	-	x	x	Sobremetal de acabamento em Z			
18	P	<_PITI>	INT	-	-	x	-	Seleção do passo, forma A-D, corresponde 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	-	-	Seleção do raio/profundidade, forma 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]					String para tabela de roscas ("", "ISO", "BSW", "BSP", "UNC") (apenas para a interface)			
20		<_PTABA>	STRING [20]					String para a seleção na tabela de roscas (por ex., "M 10", "M 12", ...) (apenas para a superfície)			

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Progr. na forma				Significado		
21		<_DMODE>	INT					Modo de Exibição		
				x	x	x	x	UNIDADE:	Plano de usinagem G17/G18/G19	
									0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
									1 =	G17 (ativo apenas no ciclo)
									2 =	G18 (ativo apenas no ciclo)
									3 =	G19 (ativo apenas no ciclo)
22		<_AMODE>	INT					modo alternativo		
				x	x	-	x	UNIDADE:	Parâmetro<_EPD> do sobremetal X ou a profundidade do alívio	
									0 =	Absoluto (sempre em diâmetro)
									1 =	incremental
				x	x	-	x	DEZENA:	Parâmetro<_EPL> do sobremetal Z ou a largura do alívio	
									0 =	Absoluto
									1 =	incremental
				x	x	x	x	CENTENA:	Parâmetro<_VRT> do avanço transversal X	
									0 =	Absoluto (sempre em diâmetro)
									1 =	incremental
				-	-	x	x	MILHAR:	Sobremetal de acabamento	
									0 =	Sobremetal de acabamento paralelo ao contorno (<_FAL>)
									1 =	Sobremetal de acabamento separado (<_FALX>/ <_FALZ>)

20.1.42 CYCLE951- Desbaste

Sintaxe

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>,
<_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
1	X0	<_SPD>	REAL	Ponto de referência (abs, sempre em diâmetro)		
2	Z0	<_SPL>	REAL	Ponto de referência (abs)		
3	X1	<_EPD>	REAL	Ponto final		
4	Z1	<_EPL>	REAL	Ponto final		
5	XM α1 α2	<_ZPD>	REAL	Ponto intermediário, consulte<_DMODE> (DEZENA)		
6	ZM α1 α2	<_ZPL>	REAL	Ponto intermediário, consulte<_DMODE> (DEZENA)		
7	Posição	<_LAGE>	INT	Posição do canto de desbaste	0 =	Externo/traseiro
					1 =	Externo/dianteiro
					2 =	Interno/traseiro
					3 =	Interno/dianteiro
8	D	<_MID>	REAL	avanço máximo em profundidade na imersão		
9	UX	<_FALX>	REAL	Sobremetal de acabamento em X		
10	UZ	<_FALZ>	REAL	Sobremetal de acabamento em Z		
11		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Sentido do desbaste (longitudinal ou transversal) no sistema de coordenadas	
					1 =	longitudinal
					2 =	transversal
				DEZENA:		
					1 =	Desbaste até o sobremetal de acabamento
					2 =	Acabamento
				CENTENA:	reservado	
				MILHAR:	reservado	
				DEZENA DA MILHAR:	reservado	
12	R1/FS1	<_RF1>	REAL	Raio de arredondamento ou largura de chanfro 1, consulte<_AMODE> (DEZENA DE MILHAR)		
13	R2/FS2	<_RF2>	REAL	Raio de arredondamento ou largura de chanfro 2, consulte<_AMODE> (CENTENA DE MILHAR)		
14	R3/FS3	<_RF3>	REAL	Raio de arredondamento ou largura de chanfro 3, consulte<_AMODE> (UNIDADE DE MILHÃO)		
15	SC	<_SDIS>	REAL	Distância de segurança		
16	F	<_FF1>	REAL	Avanço para desbaste/acabamento		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
17		<_NR>	INT	Identificação do tipo de desbaste (corresponde às softkeys verticais para seleção da forma):		
					0 =	Desbaste 1, canto de 90 graus sem chanfros/arredondamentos
					1 =	Desbaste 2, canto de 90 graus com chanfros/arredondamentos
					2 =	Desbaste 3, qualquer canto com chanfros/arredondamentos
18		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Forma de entrada <_ZPD>/<_ZPL>	
					0 =	Xm/Zm
					1 =	Xm/α1
					2 =	Xm/α2
					3 =	α1/Zm
					4 =	α2/Zm
					5 =	α1/α2

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
21		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Ponto intermediário em X	
					0 =	Absoluto, valor do eixo transversal em diâmetro
					1 =	Incremental, valor do eixo transversal em raio
				DEZENA:	Ponto intermediário em Z	
					0 =	Absoluto
					1 =	incremental
				CENTENA:	Ponto final em X	
					0 =	Absoluto, valor do eixo transversal em diâmetro
					1 =	Incremental, valor do eixo transversal em raio
				MILHAR:	Ponto final em Z	
					0 =	Absoluto
					1 =	incremental
				DEZENA DA MILHAR:	Raio/chanfro 1	
					0 =	Raio
					1 =	Chanfro
				CENTENA DA MILHAR:	Raio/chanfro 2	
					0 =	Raio
					1 =	Chanfro
				UNIDADE DE MILHÃO:	Raio/chanfro 3	
					0 =	Raio
					1 =	Chanfro

20.1.43 CYCLE952 - Usinagem de canal de contorno

Sintaxe

```
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>)
```

Parâmetros

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado
1	PRG	<_PRG>	STRING[100]	Nome do programa de desbaste
2	CON	<_CON>	STRING[100]	Nome do programa a partir do qual é interpretado o contorno da peça bruta atualizado (na usinagem residual)
3	CONR	<_CONR>	STRING[100]	Nome do programa, no qual será gravado o contorno da peça bruta atualizado (consulte<_AMODE> DEZENA DE MILHAR)

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
4		<_VARI>	INT	Tipo de usinagem		
				UNIDADE:	Tipo de desbaste	
					1 =	longitudinal
					2 =	transversal
					3 =	paralelo ao contorno
				DEZENA:	Usinagem tecnológica (consulte<_GMODE> CENTENA)	
					1 =	Desbaste
					2 =	Acabamento
					3 =	reservado
					4 =	Desbaste de dois canais
					5 =	Acabamento de dois canais
				CENTENA:	Sentido de usinagem	
					1 =	Sentido de usinagem X -
					2 =	Sentido de usinagem X +
					3 =	Sentido de usinagem Z -
					4 =	Sentido de usinagem Z +
				MILHAR:	Sentido de penetração	
					1 =	Externo X -
					2 =	Interno X +
					3 =	Lado frontal Z -
					4 =	Lado traseiro Z +
				DEZENA DA MILHAR:	Definição de como atuam os sobremetais de acabamento	
					0 =	Sobremetais de acabamento UX e UZ separados
					1 =	Sobremetal de acabamento U paralelo ao contorno
				CENTENA DA MILHAR:	Retoque	
					0 =	Compatibilidade, retoque automático
					1 =	Com retoque no contorno
					2 =	Sem retoque
					3 =	Retoque automático
				UNIDADE DE MILHÃO:	Detalonados	
					0 =	A posição não será avaliada na abertura de canais, no torneamento restante e de canal, canal residual
					1 =	Usinagem de detalonados
					2 =	Não efetuar a usinagem de detalonados
				DEZENA DE MILHÃO:	após/antes do centro de rotação	

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
					0 =	Usinagem antes do centro de rotação
					1 =	reservado
5	F	<_F>	REAL	Avanço para desbaste/acabamento		
	FZ			Avanço na abscissa para torneamento de canal		
6	FR	<_FR>	REAL	Avanço para imersão em detalonados para desbaste		
	FX			Avanço na ordenada para torneamento de canal		
7	RP	<_RP>	REAL	Plano de retrocesso na usinagem interna (abs, sempre em diâmetro)		
8	D	<_D>	REAL	Penetração de desbaste (consulte<_AMODE> UNIDADE)		
9	DX	<_DX>	REAL	Penetração X (consulte<_AMODE> UNIDADE)		
10	DZ	<_DZ>	REAL	Penetração Z (consulte<_AMODE> UNIDADE)		
11	UX	<_UX>	REAL	Sobremetal de acabamento X (consulte<_VARI> DEZENA DE MILHAR)		
12	UZ	<_UZ>	REAL	Sobremetal de acabamento Z (consulte<_VARI> DEZENA DE MILHAR)		
13	U	<_U>	REAL	Sobremetal de acabamento paralelo ao contorno (consulte<_VARI> DEZENA DE MILHAR)		
14	U1	<_U1>	REAL	Sobremetal de acabamento adicional no acabamento (consulte<_AMODE> MILHAR)		
15	BL	<_BL>	INT	Definição de peça bruta	1 =	Cilindro com sobremetal
					2 =	Sobremetal no contorno da peça pronta
					3 =	O contorno de peça bruta foi indicado
16	XD	<_XD>	REAL	Definição de peça bruta X (consulte<_AMODE> CENTENA DE MILHAR)		
17	ZD	<_ZD>	REAL	Definição de peça bruta Z (consulte<_AMODE> UNIDADE DE MILHÃO)		
18	XA	<_XA>	REAL	Limite 1 em X (abs, sempre em diâmetro)		
19	ZA	<_ZA>	REAL	Limite 1 em Z (abs)		
20	XB	<_XB>	REAL	Limite 2 X (consulte <_AMODE> DEZENA DE MILHÕES)		
21	ZB	<_ZB>	REAL	Limite 2 Z (consulte <_AMODE>CENTENA DE MILHÕES)		
22	XDA	<_XDA>	REAL	Limite para canal 1 para a 1ª posição do canal no lado frontal (abs., sempre diâmetro)		
23	XDB	<_XDB>	REAL	Limite para canal 2 para a 1ª posição do canal no lado frontal (abs., sempre diâmetro)		
24	N	<_N>	INT	Número de canais		
25	DP	<_DP>	REAL	Distância dos canais	Chanframento em série: paralelo ao eixo Z	
					Canal transversal: paralelo ao eixo X	
26	DI	<_DI>	REAL	Distância para interrupção de avanço	0 =	sem interrupção
					> 0 =	com interrupção
27	SC	<_SC>	REAL	Distância de segurança para desviar obstáculos, incremental		
28	D2	<_DN>	INT	Número D para a 2ª aresta cortante, quando não programado ⇒ D+1		

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
29		<_GMODE>	INT	GeometryMode (modo de geometria, avaliação dos valores de geometria programados)		
				UNIDADE:	reservado	
				DEZENA:	reservado	
				CENTENA:	Seleção da usinagem/apenas cálculo do ponto de partida	
					0 =	usinagem normal (nenhum modo de compatibilidade necessário)
					1 =	Usinagem normal
					2 =	Cálculo da posição de partida - sem usinagem (apenas para chamada a partir do ShopMill/ShopTurn)
				MILHAR:	Delimitação	
					0 =	não
					1 =	sim
				DEZENA DA MILHAR:	Especificação do limite 1 em X	
					0 =	não
					1 =	sim
				CENTENA DA MILHAR:	Especificação do limite 2 em X	
					0 =	não
					1 =	sim
				UNIDADE DE MILHÃO:	Especificação do limite 1 em Z	
					0 =	não
					1 =	sim
				DEZENA DE MILHÃO:	Especificação do limite 2 em Z	
					0 =	não
					1 =	sim

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
30		<_DMODE>	INT	Modo de Exibição		
				UNIDADE:	Plano de usinagem G17/G18/G19	
					0 =	Compatibilidade, permanece ativo o plano em ação antes da chamada do ciclo
					1 =	G17 (ativo apenas no ciclo)
					2 =	G18 (ativo apenas no ciclo)
					3 =	G19 (ativo apenas no ciclo)
				DEZENA:	Modo de tecnologia	
					1 =	Desbaste do contorno
					2 =	Usinagem de canal de contorno
					3 =	Torneamento de canal
				CENTENA:	Usinagem do material residual	
					0 =	não
					1 =	sim
				MILHAR:	---	reservado
				DEZENA DA MILHAR:	Escala tecnológica no interior das telas de ciclo (Página 860)	
					0 =	Especificação: completa
					1 =	Especificação: simples

Nº.	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
31		<_AMODE>	INT	modo alternativo		
				UNIDADE:	Seleção da penetração	
					0 =	Penetração DX e DZ para o tipo de desbaste paralelo ao contorno
					1 =	Penetração D
				DEZENA:	Estratégia de penetração	
					0 =	Variável da profundidade de corte (90 ... 100 %)
					1 =	profundidade de corte constante
				CENTENA:	Segmentação do corte	
					0 =	uniforme
					1 =	alinhamento nas bordas
				MILHAR:	Seleção do sobremetal de contorno U1, acabamento duplo	
					0 =	não
					1 =	sim
				DEZENA DA MILHAR:	Seleção da atualização da peça bruta	
					0 =	não
					1 =	sim
				CENTENA DA MILHAR:	Seleção do sobremetal na peça bruta XD	
					0 =	Absoluto, valor do eixo transversal em diâmetro
					1 =	Incremental, valor do eixo transversal em raio
				UNIDADE DE MILHÃO:	Seleção do sobremetal na peça bruta ZD	
					0 =	Absoluto
					1 =	incremental
				DEZENA DE MILHÃO:	Seleção do limite 2 em XB	
					0 =	Absoluto, valor do eixo transversal em diâmetro
					1 =	Incremental, valor do eixo transversal em raio
				CENTENA DE MILHÃO:	Seleção do limite 2 em ZB	
					0 =	Absoluto
					1 =	incremental
				UNIDADE DE BILHÃO		
					0 =	Canal guia
					1 =	Canal seguinte
32		<_PK>	INT	Número do canal parceiro, quando houver mais do que 2 canais na máquina		
33	DCH	<_DCH>	REAL	deslocamento de canal		
34	FS	<_FS>	REAL	Avanço de acabamento na usinagem completa		

20.1.44 CYCLE4071 - Retificação longitudinal com penetração no ponto de inversão

Sintaxe

```
CYCLE4071(<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

Parâmetro

Nº	Parâmetro	Tipo de dado	Significado
1	<S_A>	REAL	Profundidade de penetração no início
2	<S_B>	REAL	Profundidade de penetração no final
3	<S_W>	REAL	Largura da retificação
4	<S_U>	REAL	Tempo de passada final
5	<S_I>	REAL	Avanço para a penetração
6	<S_K>	REAL	Avanço para a penetração transversal
7	<S_H>	INT	Número de repetições
8	<S_A1>	AXIS	Eixo de penetração (opcional) e 1º eixo geométrico
9	<S_A2>	AXIS	Eixo oscilante (opcional) e 2º eixo geométrico

Função

O ciclo serve para a execução de <S_H>penetrações. Nesta ocasião, a profundidade de penetração no início e no final pode ser diferente. Entre as penetrações é realizado um movimento tangencial.

Sequência de operação

1. Início do ciclo na posição atual do eixo oscilante.
2. Procedimento do eixo de penetração na profundidade de penetração no início P1 **<**S_A**>** com o avanço para a penetração P5 <S_I>.
3. Passada final no tempo P4 **<**S_U**>**.
4. Procedimento do eixo oscilante com a largura de rebolo P3 **<**S_W**>** como percurso e avanço da penetração transversal P6 <S_K>.
5. Procedimento do eixo de penetração na profundidade de penetração no final P2 **<**S_B**>** com o avanço para a penetração P5 <S_I>.
6. Passada final no tempo P4 **<**S_U**>**.
7. Procedimento do eixo oscilante com a largura de rebolo P4 **<**S_A**>** como percurso do ponto inicial e avanço para penetração transversal P6 <S_K>.

A sequência de operação não pode ser interrompida com um bloco individual.

A sequência será repetida de acordo o número programado para as repetições P7 (<S_H>).

Exemplo

Executar dois movimentos pendulares com os seguintes parâmetros do ciclo:

- Profundidade de penetração no início: 0,02 mm
- Profundidade de penetração no final: 0,01 mm
- Hub: 100 mm
- Tempo de passada final: 1 s
- Avanço da penetração: 1 mm/min
- Avanço transversal: 1000 mm/min
- Repetições: 2
- Eixos pendulares e eixo de penetração: Eixos geométricos padrão

Código de programa

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.45 CYCLE4072 - Retificação longitudinal com penetração no ponto de inversão e sinal de interrupção

Sintaxe

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

Parâmetro

Nº	Parâmetro	Tipo de dado	Significado
1	<S_GAUGE>	STRING	Condições de interrupção para a penetração: 1. Número de uma entrada rápida 2. Termo lógico
2	<S_A>	REAL	Profundidade de penetração no início
3	<S_B>	REAL	Profundidade de penetração no final
4	<S_W>	REAL	Largura da retificação
5	<S_U>	REAL	Tempo de passada final
6	<S_I>	REAL	Avanço para a penetração
7	<S_K>	REAL	Avanço para a penetração transversal
8	<S_H>	INT	Número de repetições
9	<S_A1>	AXIS	Eixo de penetração (opcional) e 1º eixo geométrico
10	<S_A2>	AXIS	Eixo oscilante (opcional) e 2º eixo geométrico

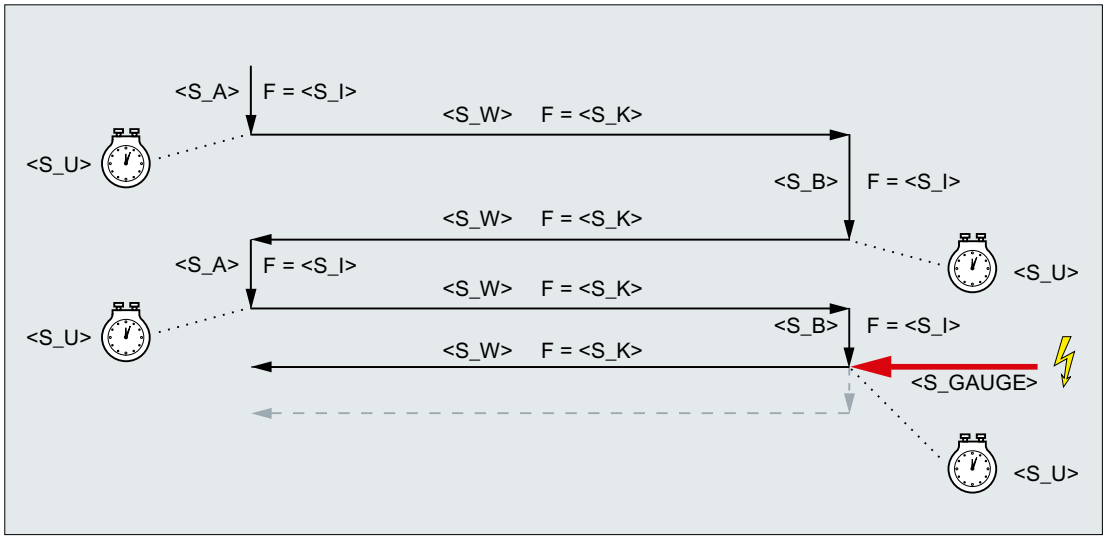
Função

O ciclo serve para a execução de <S_H>penetrações sob condições de um sinal externo de interrupções. A profundidade de penetração no início e no final pode ser diferente. Entre as penetrações é realizado um movimento tangencial. A penetração profunda é interrompida, quando a condição de interrupção for satisfeito. Após a interrupção da penetração profunda é executado sempre um curso completo.

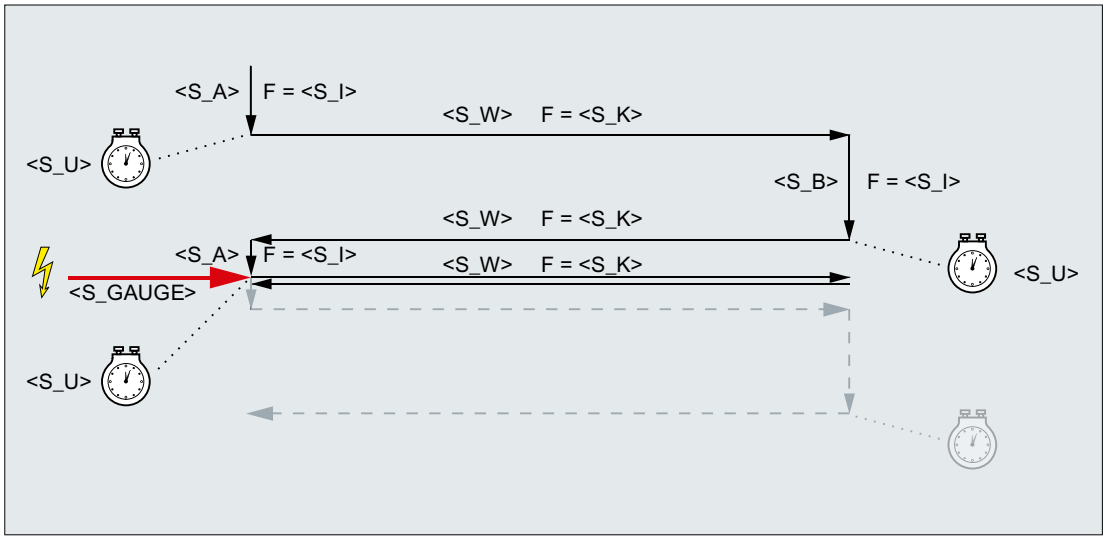
Sequência de operação

1. Início do ciclo na posição atual do eixo oscilante.
2. Procedimento do eixo de penetração na profundidade de penetração no início P2 **<**S_A**>** com o avanço para a penetração P6 <S_I>.
3. Passada final no tempo P5 **<**S_U**>**.
4. Procedimento do eixo oscilante com a largura de rebolo P4 **<**S_W**>** como percurso e avanço da penetração transversal P7 <S_K>.
5. Procedimento do eixo de penetração na profundidade de penetração no final P3 **<**S_B**>** com o avanço para a penetração P6 <S_I>.
6. Passada final no tempo P5 **<**S_U**>**.
7. Procedimento do eixo oscilante com a largura de rebolo P4 **<**S_W**>** como percurso no ponto inicial e avanço para penetração transversal P7 <S_K>.
8. Sem cancelamento: A sequência mencionada acima será repetida até que o número programado para as repetições P7 seja alcançado (<S_H>).
   Com cancelamento: O processamento é encerrado quando se alcança o próximo ponto inicial.

A sequência de operação não pode ser interrompida pelo bloco individual.



Esquema 20-1 Interrupção da penetração no final



Esquema 20-2 Interrupção da penetração no início

Recursos

Como recursos, o ciclo utiliza uma ação de sincronização extensiva aos blocos e uma variável de ação de sincronização. A ação de sincronização é determinada de forma dinâmica a partir da área livre da banda de ação sincronizada (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Como variável da ação de sincronização é utilizado SYG_IS[1].

Exemplos

Exemplo 1: Oscilação com dois avanços

Parâmetros de ciclo:

- Profundidade de penetração no início: 0,02 mm
- Profundidade de penetração no final: 0,01 mm
- Hub: 100 mm
- Tempo de passada final: 1 s
- Avanço da penetração: 1 mm/min
- Avanço transversal: 1000 mm/min
- Repetições: 2
- Eixos pendulares e eixo de penetração: Eixos geométricos padrão

Sinal de interrupção: Entrada rápida 1 ($A_IN[1])

Código de programa

```
N10 T1 D1
N20 CYCLE4072("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

Exemplo 2: Oscilação com dois avanços

Parâmetros de ciclo:

- Profundidade de penetração no início: 0,02 mm
- Profundidade de penetração no final: 0,01 mm
- Hub: 100 mm
- Tempo de passada final: 1 s
- Avanço da penetração: 1 mm/min
- Avanço transversal: 1000 mm/min
- Repetições: 2
- Eixos pendulares e eixo de penetração: Eixos geométricos padrão

Sinal de interrupção: Variável $A_DBR[20] < 0,01

Código de programa

```
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 - Retificação longitudinal com penetração contínua

Sintaxe

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>, <S_A2>)
```

Parâmetro

Nº	Parâmetro	Tipo de dado	Significado
1	<S_A>	REAL	Profundidade de penetração no início
2	<S_B>	REAL	Profundidade de penetração no final
3	<S_W>	REAL	Largura da retificação
4	<S_U>	REAL	Tempo de passada final
5	<S_K>	REAL	Avanço para a penetração transversal
6	<S_H>	INT	Número de repetições
7	<S_A1>	AXIS	Eixo de penetração (opcional) e 1º eixo geométrico
8	<S_A2>	AXIS	Eixo oscilante (opcional) e 2º eixo geométrico

Função

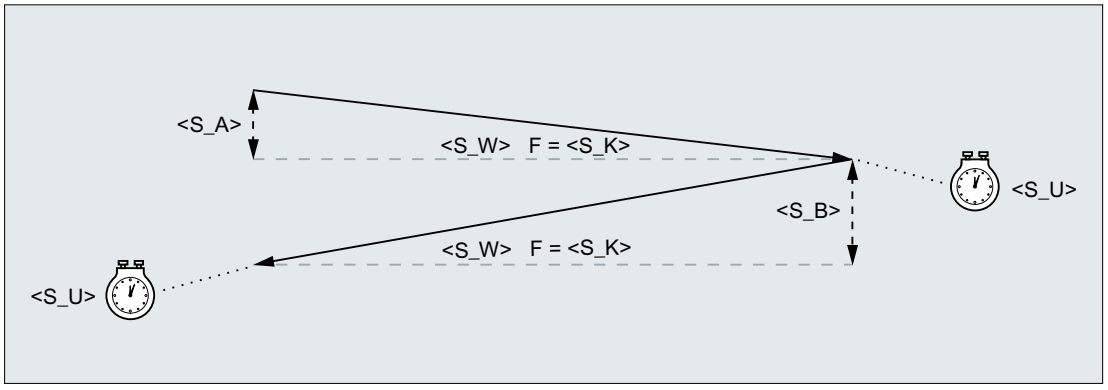
O ciclo serve para a execução de <S_H> penetrações. Nesta ocasião, a penetração pode ser diferente do início para o final e do final para o início.

Sequência de operação

1. Início do ciclo na posição atual do eixo oscilante com profundidade de penetração 0
2. Procedimento do eixo oscilante com a largura de rebolo P3 <S_W> como percurso e avanço para penetração transversal P5 <S_K> com o aumento contínuo da profundidade de penetração para a profundidade definida no início de P1<S_A>.
3. Passada final no tempo P4 <S_U>.
4. Procedimento do eixo oscilante com a largura de rebolo P3 <S_W> como percurso no ponto inicial e avanço para penetração transversal P5 <S_K> com o aumento contínuo da profundidade de penetração para a profundidade definida no final de P2 <S_B>.
5. Passada final no tempo P4 <S_U>.

A sequência de operação não pode ser interrompida pelo bloco individual.

A sequência será repetida de acordo o número programado para as repetições P7 (<S_H>).



Exemplo

Oscilação com dois avanços

Parâmetros de ciclo:

- Profundidade de penetração no início: 0,02 mm
- Profundidade de penetração no final: 0,01 mm
- Hub: 100 mm
- Tempo de passada final: 1 s
- Avanço transversal: 1000 mm/min
- Repetições: 2
- Eixos pendulares e eixo de penetração: Eixos geométricos padrão

Código de programa

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.47 CYCLE4074 - Retificação longitudinal com penetração contínua e sinal de interrupção

Sintaxe

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

Parâmetro

Nº	Parâmetro	Tipo de dado	Significado
1	<S_GAUGE>	STRING	Condições de interrupção para a penetração: 1. Número de uma entrada rápida 2. Termo lógico
2	<S_A>	REAL	Profundidade de penetração no início
3	<S_B>	REAL	Profundidade de penetração no final
4	<S_W>	REAL	Largura da retificação
5	<S_U>	REAL	Tempo de passada final
6	<S_K>	REAL	Avanço para a penetração transversal
7	<S_H>	INT	Número de repetições
8	<S_A1>	AXIS	Eixo de penetração (opcional) e 1º eixo geométrico
9	<S_A2>	AXIS	Eixo oscilante (opcional) e 2º eixo geométrico

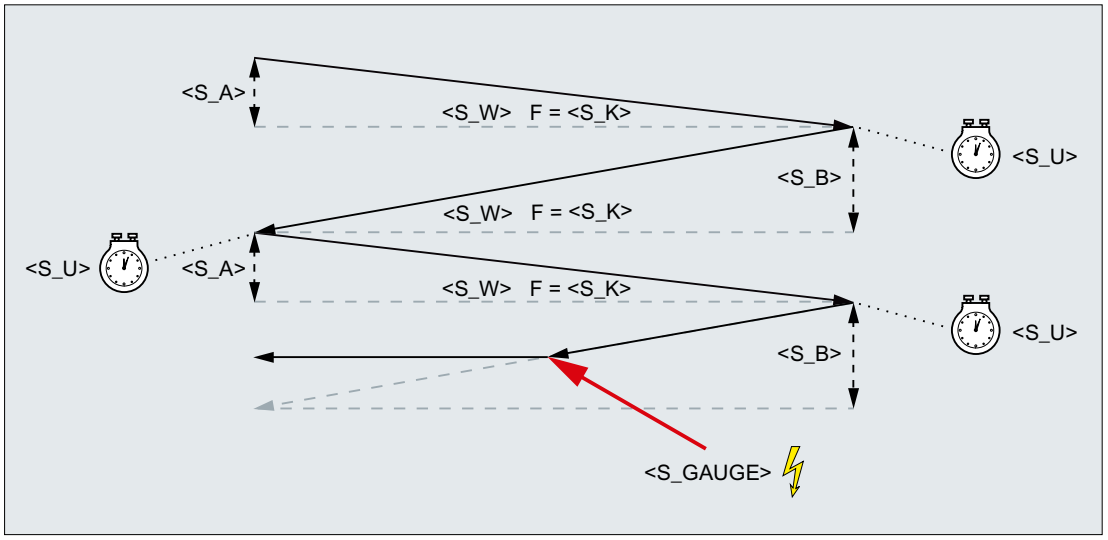
Função

O ciclo serve para a execução de <S_H>penetrações sob condições, por ex., de um sinal externo de interrupções. A profundidade de penetração no início e no final pode ser diferente. A penetração profunda é interrompida, quando a condição de interrupção for satisfeito. Após a interrupção da penetração profunda é executado sempre um curso completo.

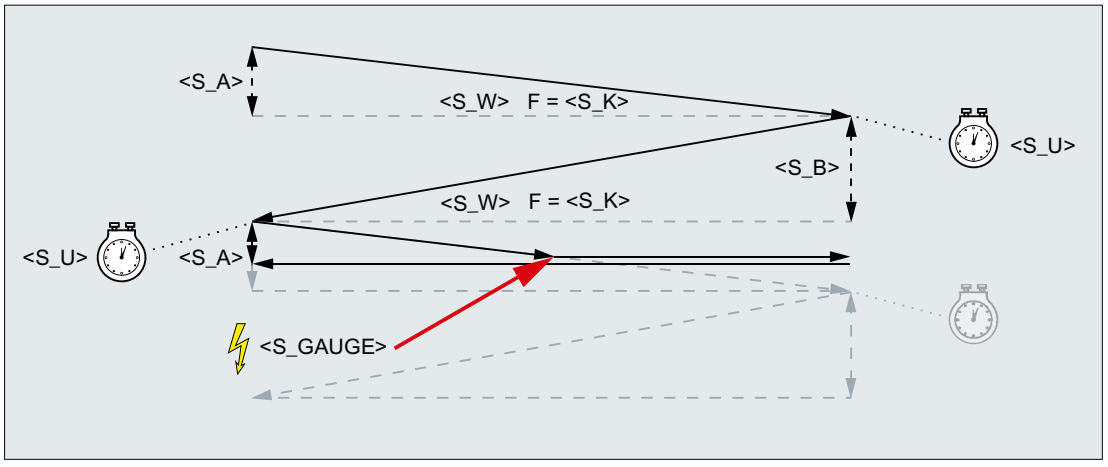
Sequência de operação

1. Início do ciclo na posição atual do eixo oscilante com profundidade de penetração 0
2. Procedimento do eixo oscilante com a largura de rebolo P4 <S_W> como percurso e avanço para penetração transversal P6 <S_K> com o aumento contínuo da profundidade de penetração para a profundidade definida no início de P2<S_A>.
3. Passada final no tempo P5 <S_U>.
4. Procedimento do eixo oscilante com a largura de rebolo P4 <S_W> como percurso no ponto inicial e avanço para penetração transversal P6 <S_K> com o aumento contínuo da profundidade de penetração para a profundidade definida no final de P3 <S_B>.
5. Passada final no tempo P5 <S_U>.
6. Sem cancelamento: A sequência mencionada acima será repetida até que o número programado para as repetições P7 seja alcançado (<S_H>).
   Com cancelamento: A penetração profunda é interrompida. O processamento é encerrado com o alcance ao próximo ponto inicial.

A sequência de operação não pode ser interrompida pelo bloco individual.



Esquema 20-3 Interrupção da penetração do final para o início



Esquema 20-4 Interrupção da penetração do início para o final

Recursos

Como recursos, o ciclo utiliza uma ação de sincronização extensiva aos blocos e uma variável de ação de sincronização. A ação de sincronização é determinada de forma dinâmica a partir da área livre da banda de ação sincronizada (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Como variável da ação de sincronização é utilizado SYG_IS[1].

Exemplos

Exemplo 1: Oscilação com dois avanços

Parâmetros de ciclo:

- Profundidade de penetração no início: 0,02 mm
- Profundidade de penetração no final: 0,01 mm

- Hub: 100 mm
- Tempo de passada final: 1 s
- Avanço transversal: 1000 mm/min
- Repetições: 2
- Eixos pendulares e eixo de penetração: Eixos geométricos padrão

Sinal de interrupção: Entrada rápida 1 ($A_IN[1])

Código de programa

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

Exemplo 2: Oscilação com dois avanços

Parâmetros de ciclo:

- Profundidade de penetração no início: 0,02 mm
- Profundidade de penetração no final: 0,01 mm
- Hub: 100 mm
- Tempo de passada final: 1 s
- Avanço transversal: 1000 mm/min
- Repetições: 2
- Eixos pendulares e eixo de penetração: Eixos geométricos padrão

Sinal de interrupção: Variável $A_DBR[20] < 0,01

Código de programa

```
N10 T1 D1
N20 CYCLE4074("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.48 CYCLE4075 - Retificação plana com penetração no ponto de inversão

Sintaxe

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Parâmetro

Nº	Parâmetro	Tipo de dado	Significado
1	<S_I>	REAL	Profundidade de penetração no início
2	<S_J>	REAL	Profundidade de penetração no final

Nº	Parâmetro	Tipo de dado	Significado
3	<S_K>	REAL	Profundidade de penetração total
4	<S_A>	REAL	Largura da retificação
5	<S_R>	REAL	Avanço para a penetração
6	<S_F>	REAL	Avanço para a penetração transversal
7	<S_P>	REAL	Tempo de passada final
8	<S_A1>	AXIS	Eixo de penetração (opcional)
9	<S_A2>	AXIS	Eixo oscilante (opcional)

Função

O ciclo serve para o processamento com uma profundidade de penetração P3 <S_K> em etapas de penetração. A profundidade de penetração no início P1 <S_I> e final P2 <S_J> pode ser diferente. Entre as penetrações é realizado um movimento tangencial.

As indicações de curso P1 a P4 podem ser negativas ou positivas.

As indicações de curso P8 <S_A1> e/ou do eixo oscilante P9 <S_A2> são opcionais. Se um ou ambos os parâmetros não forem indicados, o ciclo utiliza os primeiros dois eixos geométricos do canal.

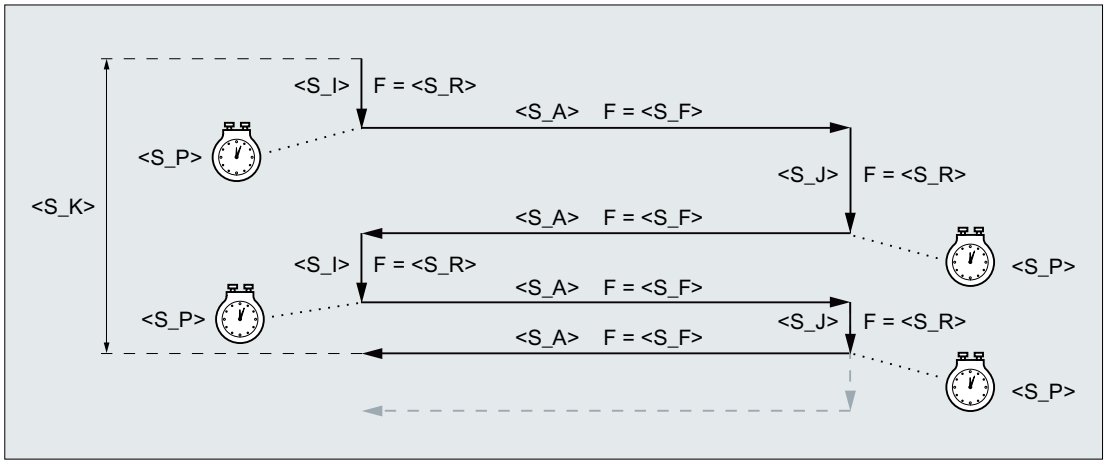
Se a soma da profundidade de penetração no início P1 <S_I> e no final P2 <S_J> for igual a 0, e a profundidade de penetração total P3 for <S_K> igual a 0, só um avanço de passada final será executado.

Sequência de operação

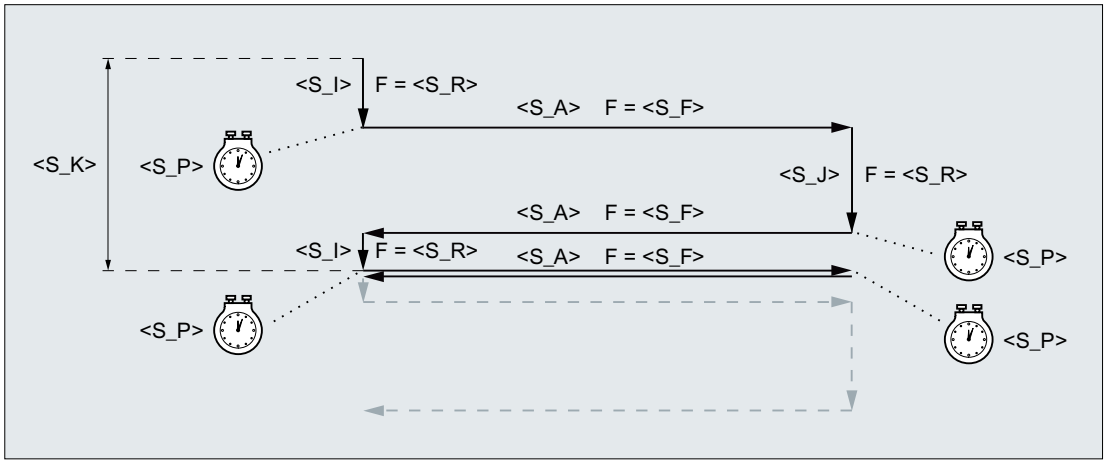
1. Início do ciclo na posição atual do eixo oscilante.
2. Procedimento do eixo de penetração na profundidade de penetração no início P1 <S_I> com o avanço para a penetração P5 <S_R>.
3. Passada final no tempo P7 <S_P>
4. Procedimento do eixo oscilante com a largura de rebolo P4 <S_A> como percurso e avanço da penetração transversal P6 <S_F>.
5. Procedimento do eixo de penetração na profundidade de penetração no final P2 <S_J> com o avanço para a penetração P5 <S_R>.
6. Passada final no tempo P7 <S_P>.
7. Procedimento do eixo oscilante com a largura de rebolo P4 <S_A> como percurso do ponto inicial e avanço para penetração transversal P6 <S_F>.

A sequência de operação não pode ser interrompida com um bloco individual.

A sequência será repetida até que a profundidade de penetração total P3 <S_K> seja alcançada. O último curso então será dividido irregularmente.



Esquema 20-5 Profundidade de penetração total atingida na penetração no segundo ponto de inversão



Esquema 20-6 Profundidade de penetração total atingida na penetração no primeiro ponto de inversão

Exemplo

Oscilação com:

- 0,02 mm de profundidade de penetração no início
- 0,01 mm de profundidade de penetração no final
- Profundidade de penetração total 1 mm
- Curso de 100 mm
- Avanço da penetração 1 mm/min
- Avanço transversal 1000 mm/min
- Tempo de passada final 1 segundo
- Eixos geométricos padrão

Código de programa

```
N10 T1 D1
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.49 CYCLE4077 - Retificação plana com penetração no ponto de inversão e sinal de interrupção

Sintaxe

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

Parâmetro

Nº	Parâmetro	Tipo de dado	Significado
1	<S_GAUGE>	STRING	Condição de interrupção para a penetração: • Número de uma entrada rápida • Termo lógico
2	<S_I>	REAL	Profundidade de penetração no início
3	<S_J>	REAL	Profundidade de penetração no final
4	<S_K>	REAL	Profundidade de penetração total
5	<S_A>	REAL	Largura da retificação
6	<S_R>	REAL	Avanço para a penetração
7	<S_F>	REAL	Avanço para a penetração transversal
8	<S_P>	REAL	Tempo de passada final
9	<S_A1>	AXIS	Eixo de penetração (opcional)
10	<S_A2>	AXIS	Eixo oscilante (opcional)

Função

O ciclo serve para o processamento com uma profundidade de penetração P4 <S_K> em etapas de penetração. A profundidade de penetração no início P2 <S_I> e final P3 <S_J> pode ser diferente. Entre as penetrações é realizado um movimento tangencial. A penetração profunda é interrompida, quando o sinal de interrupção da entrada rápida for igual a 1 ou a condição de interrupção for satisfeita. Após a interrupção, será executado um hub completo.

As indicações de curso P2 a P5 podem ser negativas ou positivas.

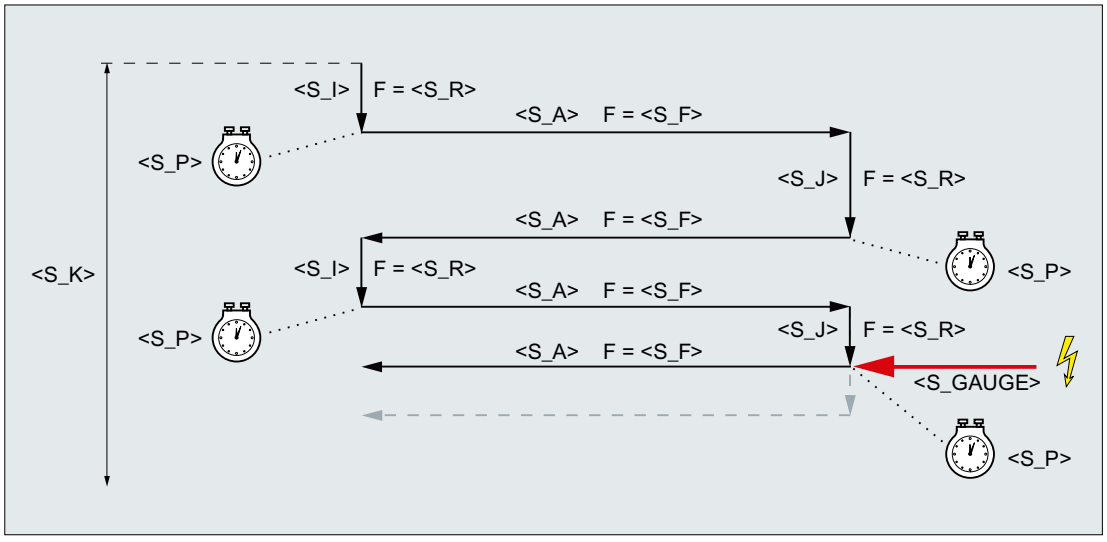
As indicações de curso P9 <S_A1> e/ou do eixo oscilante P10 <S_A2> são opcionais. Se um ou ambos os parâmetros não forem indicados, o ciclo utiliza os primeiros dois eixos geométricos do canal.

Se a soma da profundidade de penetração no início P2 <S_I> e no final P3 <S_J> for igual a 0, e a profundidade de penetração total P4 for <S_K> igual a 0, só um avanço de passada final será executado.

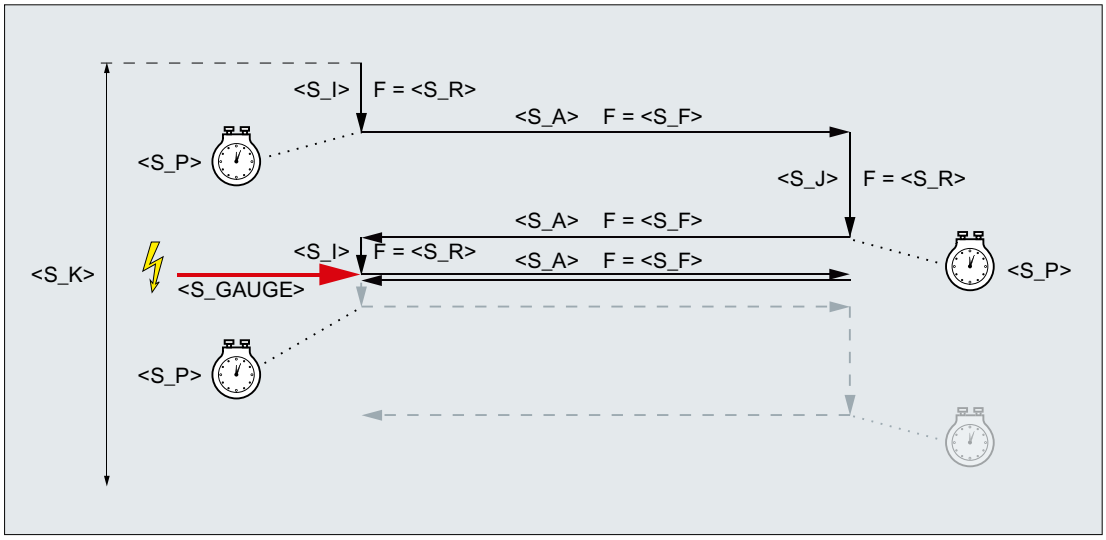
Sequência de operação

1. Início do ciclo na posição atual do eixo oscilante.
2. Procedimento do eixo de penetração na profundidade de penetração no início P2 **<**S_I**>** com o avanço para a penetração P6 <S_R>.
3. Passada final no tempo P8 **<**S_P**>**.
4. Procedimento do eixo oscilante com a largura de rebolo P5 **<**S_A**>** como percurso e avanço da penetração transversal P7 <S_F>.
5. Procedimento do eixo de penetração na profundidade de penetração no final P3 **<**S_J**>** com o avanço para a penetração P6 <S_R>.
6. Passada final no tempo P8 **<**S_P**>**.
7. Procedimento do eixo oscilante com a largura de rebolo P5 **<**S_A**>** como percurso no ponto inicial e avanço para penetração transversal P7 <S_F>.
8. Sem cancelamento: A sequência mencionada acima será repetida até que a profundidade de penetração total P4 **<**S_K**>** seja alcançada. O último curso então será dividido irregularmente.
   Com cancelamento: A execução será finalizada no ponto inicial.

A sequência de operação não pode ser interrompida com um bloco individual.



Esquema 20-7 Interrupção da penetração no final



Esquema 20-8 Interrupção da penetração no início

Recursos

Como recursos, o ciclo utiliza uma ação de sincronização extensiva aos blocos e uma variável de ação de sincronização. A ação de sincronização é determinada de forma dinâmica a partir da área livre da banda de ação sincronizada (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). Como variável da ação de sincronização é utilizado SYG_IS[1].

Exemplos

Exemplo 1

Oscilação com:

- 0,02 mm de profundidade de penetração no início
- 0,01 mm de profundidade de penetração no final
- Profundidade de penetração total 1 mm
- Curso de 100 mm
- Avanço da penetração 1 mm/min
- Avanço transversal 1000 mm/min
- Tempo de passada final 1 segundo
- Eixos geométricos padrão

Sinal de interrupção: Entrada rápida 1 ($A_IN[1])

Código de programa

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

Exemplo 2

Oscilação com:

- 0,02 mm de profundidade de penetração no início
- 0,01 mm de profundidade de penetração no final
- Profundidade de penetração total 1 mm
- Curso de 100 mm
- Avanço da penetração 1 mm/min
- Avanço transversal 1000 mm/min
- Tempo de passada final 1 segundo
- Eixos geométricos padrão

Sinal de interrupção: Variável Dualport-RAM 20 inferior a 0,01 ($A_DBR[20] < 0,01)

Código de programa

```
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 - Retificação plana com penetração contínua

Sintaxe

```
CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)
```

Parâmetro

Nº	Parâmetro	Tipo de dado	Significado
1	<S_I>	REAL	Profundidade de penetração do início ao fim
2	<S_J>	REAL	Profundidade de penetração do fim ao início
3	<S_K>	REAL	Profundidade de penetração total
4	<S_A>	REAL	Largura da retificação
5	<S_F>	REAL	Avanço
6	<S_P>	REAL	Tempo de passada final
7	<S_A1>	AXIS	Eixo de penetração (opcional)
8	<S_A2>	AXIS	Eixo oscilante (opcional)

Função

O ciclo serve para o processamento com uma profundidade de penetração total P3 <S_K> por meio de penetração contínua. A profundidade de penetração do início ao final P1 <S_I> e final ao início P2 <S_J> pode ser diferente.

As indicações de curso P1 a P4 podem ser negativas ou positivas.

As indicações de curso P8 <S_A1> e/ou do eixo oscilante P9 <S_A2> são opcionais. Se um ou ambos os parâmetros não forem indicados, o ciclo utiliza os primeiros dois eixos geométricos do canal.

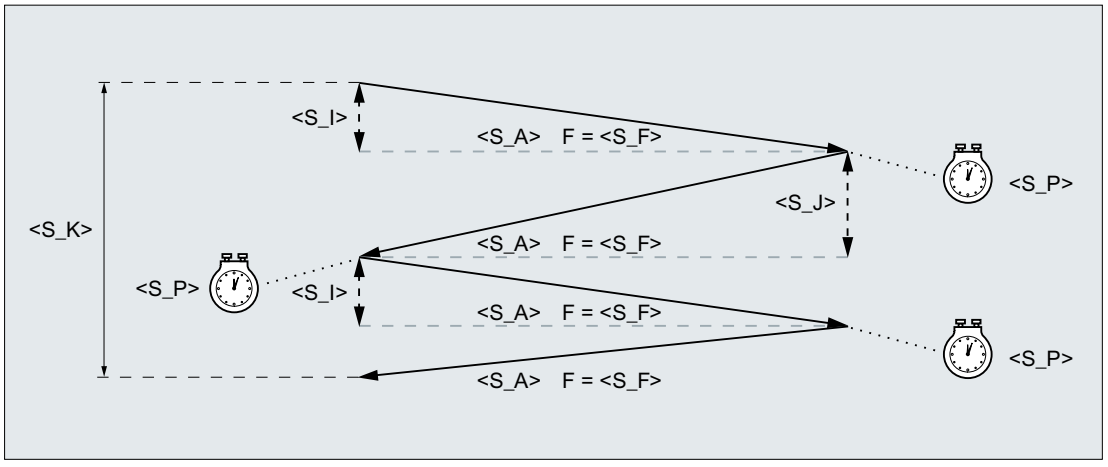
Se a soma da profundidade de penetração P1 <S_I> e P2 <S_J> for igual a 0, e a profundidade de penetração total P3 for <S_K> igual a 0, só um avanço de passada final será executado.

Sequência de operação

1. Início do ciclo na posição atual do eixo oscilante com profundidade de penetração 0
2. Procedimento do eixo oscilante com a largura de rebolo P4 <S_A> como percurso e avanço P5 <S_F> com o aumento contínuo da profundidade de penetração para a profundidade definida no início de P1 <S_I>.
3. Passada final no tempo P7 <S_P>
4. Procedimento do eixo oscilante com a largura de rebolo P4 <S_A> como percurso no ponto inicial e avanço P5 <S_F> com o aumento contínuo da profundidade de penetração para a profundidade definida no final de P2 <S_J>.
5. Passada final no tempo P7 <S_P>.
6. Procedimento do eixo oscilante com a largura de rebolo P4 <S_A> como percurso no ponto inicial e avanço P5 <S_F>.

A sequência de operação não pode ser interrompida com um bloco individual.

A sequência será repetida até que a profundidade de penetração total P3 <S_K> seja alcançada. O último curso então será dividido irregularmente.



Exemplo

Oscilação com:

- 20 mm de profundidade de penetração no início
- 10 mm de profundidade de penetração no final
- Profundidade de penetração total 100 mm

- Curso de 100 mm
- Avanço de 1000 mm/min
- Tempo de passada final 1 segundo
- Eixos geométricos padrão

Código de programa

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

20.1.51 CYCLE4079 - Retificação plana com penetração intermitente

Sintaxe

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

Parâmetro

Nº	Parâmetro	Tipo de dado	Significado
1	<S_I>	REAL	Profundidade de penetração no início
2	<S_J>	REAL	Profundidade de penetração no final
3	<S_K>	REAL	Profundidade de penetração total
4	<S_A>	REAL	Largura da retificação
5	<S_R>	REAL	Avanço para a penetração
6	<S_F>	REAL	Avanço para a penetração transversal
7	<S_P>	REAL	Tempo de passada final
8	<S_A1>	AXIS	Eixo de penetração (opcional)
9	<S_A2>	AXIS	Eixo oscilante (opcional)

Função

O ciclo serve para o processamento com uma profundidade de penetração P3 <S_K> em etapas de penetração. A profundidade de penetração no início P1 <S_I> e final P2 <S_J> pode ser diferente. Entre as penetrações é realizado um movimento tangencial.

As indicações de curso P1 a P4 podem ser negativas ou positivas.

As indicações de curso P8 <S_A1> e/ou do eixo oscilante P9 <S_A2> são opcionais. Se um ou ambos os parâmetros não forem indicados, o ciclo utiliza os primeiros dois eixos geométricos do canal.

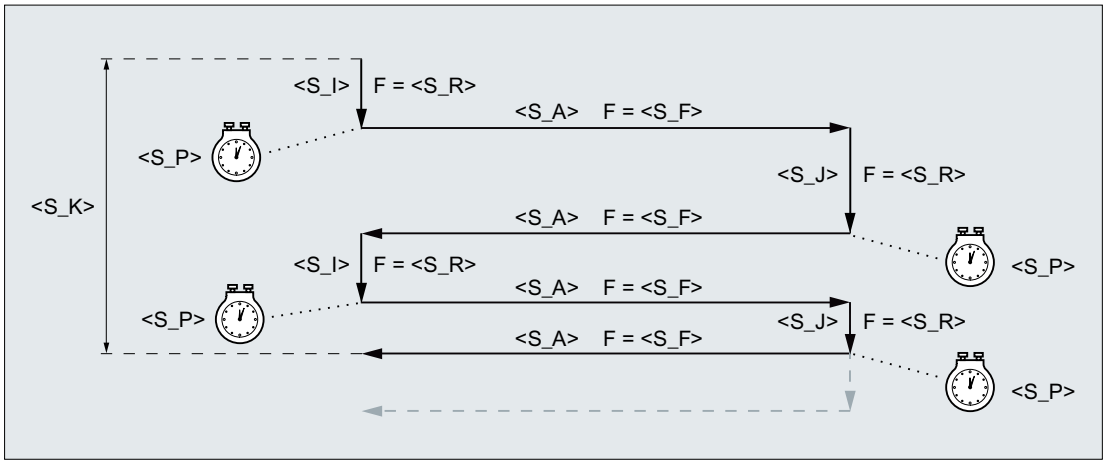
Se a soma da profundidade de penetração no início P1 <S_I> e no final P2 <S_J> for igual a 0, e a profundidade de penetração total P3 for <S_K> igual a 0, só um avanço de passada final será executado.

Sequência de operação

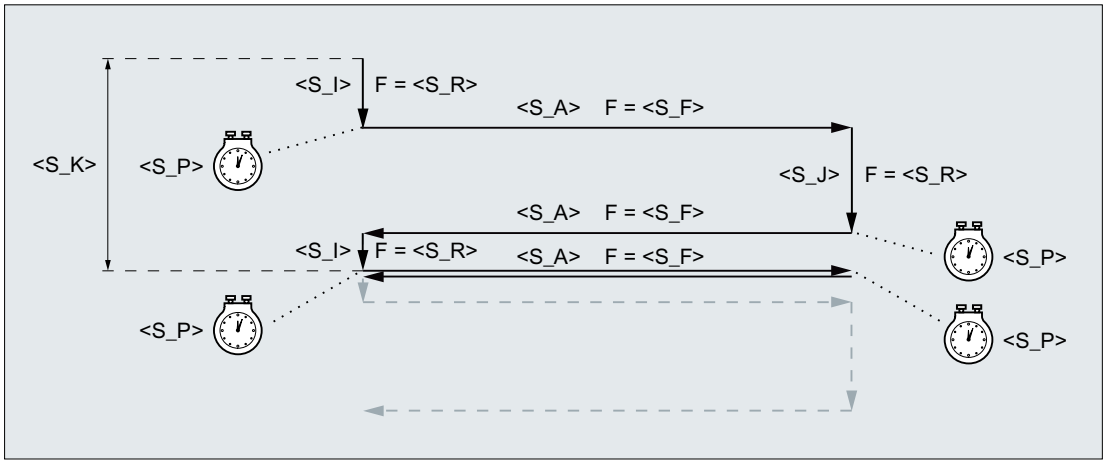
1. Início do ciclo na posição atual do eixo oscilante.
2. Procedimento do eixo de penetração na profundidade de penetração no início P1 **<**S_I**>** com o avanço para a penetração P5 <S_R>.
3. Passada final no tempo P7 **<**S_P**>**
4. Procedimento do eixo oscilante com a largura de rebolo P4 **<**S_A**>** como percurso e avanço da penetração transversal P6 <S_F>.
5. Procedimento do eixo de penetração na profundidade de penetração no final P2 **<**S_J**>** com o avanço para a penetração P5 <S_R>.
6. Passada final no tempo P7 **<**S_P**>**.
7. Procedimento do eixo oscilante com a largura de rebolo P4 **<**S_A**>** como percurso do ponto inicial e avanço para penetração transversal P6 <S_F>.

A sequência de operação não pode ser interrompida com um bloco individual.

A sequência será repetida até que a profundidade de penetração total P3 **<**S_K**>** seja alcançada. O último curso então será dividido irregularmente.



Esquema 20-9 Profundidade de penetração total atingida na penetração no segundo ponto de inversão



Esquema 20-10 Profundidade de penetração total atingida na penetração no primeiro ponto de inversão

Exemplo

Oscilação com:

- 0,02 mm de profundidade de penetração no início
- 0,01 mm de profundidade de penetração no final
- Profundidade de penetração total 1 mm
- Curso de 100 mm
- Avanço da penetração 1 mm/min
- Avanço transversal 1000 mm/min
- Tempo de passada final 1 segundo
- Eixos geométricos padrão

Código de programa

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.52 GROUP_BEGIN - Início bloco de programa

Sintaxe

```
GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado	
1		<_LEVEL>	INT	Plano	
				0 =	Nível principal
				1 =	1º subnível
2		<_NAME>	STRING[128]	Nome do bloco	
3		<_SP>	INT	Fuso	
				0 =	sem fuso
				1 =	Fuso principal
				2 =	Contrafuso

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado		
4		<_MODE>	INT	Modo		
				Bit 0	= 1	GROUP_ADDEND existe
				Bit 1	= 1	ShopTurn: Retrocesso automático (deslocar ao o ponto de troca de ferramenta)
				Bit 12	reservado	
				Bit 13	reservado	
5		<S_ICON>	STRING[32]	Nome do ícone (somente para interface de usuário)		

20.1.53 GROUP_END - Final bloco de programa

Sintaxe

```
GROUP_END(<_LEVEL>, <_SP>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado	
1		<_LEVEL>	INT	Plano	
				0 =	Nível principal
				1 =	1º subnível
2		<_SP>	INT	Fuso	
				0 =	sem fuso
				1 =	Fuso principal
				2 =	Contrafuso

20.1.54 GROUP_ADDEND - Final adendo de entrada

Sintaxe

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

Parâmetro

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado	
1		<_LEVEL>	INT	Plano	
				0 =	Nível principal
				1 =	1º subnível

Nº	Tela de parâmetros	Parâmetros internos	Tipo de dado	Significado	
2		<_SP>	INT	Fuso	
				0 =	sem fuso
				1 =	Fuso principal
				2 =	Contrafuso

20.1.55 Condições gerais

20.1.55.1 Escala tecnológica no interior das telas de ciclo

Se a escala tecnológica estiver ativa, nas diversas telas de ciclos pode ser selecionada a especificação simples, na qual são exibidos apenas os parâmetros de ciclos fundamentais.

Nas seguintes telas de ciclos pode, por ex., ser selecionada a especificação simples:

Tecnologia	Tela de ciclos
Perfurar	Furação profunda
	Rosqueamento com macho
Fresamento	Bolsão retangular
	Fresamento de contorno: Bolsão
Torneamento	Rosqueamento: longitudinal
	Torneamento de contorno: Desbaste
	Torneamento de contorno: Abertura de canais
	Torneamento de contorno: Torneamento de canal

Nas telas de ciclos em questão, na interface do usuário será disponibilizada a opção de seleção "Especificação: **simples**" e "Especificação: **completa**".

Parâmetros de ciclos não exibidos

Os parâmetros de ciclos não exibidos na especificação simples são pré-atribuídos com valores fixos, tecnologicamente razoáveis, porém inalteráveis. Ou, aos parâmetros de ciclos são atribuídos, através dos dados de configuração de ciclos específicos de canal, os valores parametrizáveis. Consultar a seção abaixo "Colocação em funcionamento" > "Dados de configuração dos ciclos específicos de canal"

Comutação "Especificação": completa" > "Especificação: simples"

Se uma tela de ciclos for preenchida na configuração "Especificação completa" e em seguida, for comutado para "Especificação simples", para a criação da chamada do ciclo para os parâmetros que não forem mais exibidos, serão utilizados os valores de dados Default ou de configuração.

Colocação em funcionamento

Dados de máquina de configuração específicos de canal

Com o dado da máquina é ativada a possibilidade para a escala tecnológica dentro das telas de ciclos:

MD52210 $MCS_FUNCTION_MASK_DISP, Bit 9 = 1 (Exibir a seleção "Especificação simples")

Dados de ajuste para ciclos específicos de canal

Se a especificação simples estiver ativa em uma tela de ciclo, pode ser predefinidos os valores para determinados parâmetros de ciclo através dos seguintes dados de configuração:

Número	Identificador	Significado
SD55300	$SCS_EASY_SAFETY_CLEARANCE	Distância de segurança
SD55301	$SCS_EASY_DWELL_TIME	Tempo de espera
SD55305	$SCS_EASY_DRILL_DEEP_FD1	Furação profunda: Porcentagem: 1. Avanço
SD55306	$SCS_EASY_DRILL_DEEP_DF	Furação profunda: Porcentagem: Penetração
SD55307	$SCS_EASY_DRILL_DEEP_V1	Furação profunda: penetração mínima em profundidade
SD55308	$SCS_EASY_DRILL_DEEP_V2	Furação profunda: valor de retrocesso
SD55309	$SCS_EASY_THREAD_RETURN_DIST	Rosqueamento: Distância de retrocesso

20.2 Ciclos de medição

Os ciclos de medição são sub-rotinas disponibilizadas especialmente pela Siemens para a solução de determinadas tarefas de medição. Como geralmente é usual nos ciclos, os ciclos de medição também podem ser adaptados ao problema concreto por meio de parâmetros.

São disponibilizados os ciclos de medição para as medições nos seguintes setores e tecnologias:

- Medições de ferramentas de Torneamento / fresamento
- Medições de peças de trabalho de Torneamento / fresamento

Literatura

A descrição detalhada dos ciclos de medição pode ser encontrada em:

Manual de programação de ciclos de medição

Tabelas

21

21.1 Instruções

Indicação

Ciclos

A lista das instruções contém todos os ciclos pode ocorrer no programa NC (código G), isto é, pode ser programado no editor do programa através da tela ou deve ser programado retificação sem o apoio de programação. Não foram considerados ciclos, que ainda estão disponíveis por motivos de compatibilidade no comando, mas não podem mais ser editados através do editor do programa do SINUMERIK Operate ("ciclos de compatibilidade").

Instruções A ... C

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
:	O	NC- número de bloco principal, encerramento de marca de transferência, operador de encadeamento		+		PGAsl
*	O	Operador para multiplicação		+		PGAsl
+	O	Operador para adição		+		PGAsl
-	O	Operador para subtração		+		PGAsl
<	O	Operador de comparação, menor		+		PGAsl
<<	O	Operador de concatenação para Strings		+		PGAsl
<=	O	Operador de comparação, menor igual		+		PGAsl
=	O	Operador de atribuição		+		PGAsl
>=	O	Operador de comparação, maior igual		+		PGAsl
/	O	Operador para divisão		+		PGAsl
/0 … … /7		Bloco será extraído (1º nível de extração)° ... ... Bloco será extraído (8º bloco de extração)		+		PGsl
A	A	Nome de eixo	m/b	+		PGAsl
A2	A	Orientação da ferramenta: Ângulo RPY ou ângulo euleriano	s	+		PGAsl
A3	A	Orientação da ferramenta: 1. Componente do vetor de direção	s	+		PGAsl
A4	A	Orientação da ferramenta: 1. Componente do vetor normal de superfície no início do bloco	s	+		PGAsl
A5	A	Orientação da ferramenta: 1. Componente do vetor normal de superfície no fim do bloco	s	+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
A6	A	Orientação da ferramenta: 1. Componente do vetor de direção para o eixo de rotação do cone	s	+		PGAsl
A7	A	Orientação da ferramenta: 1. Componente do vetor para orientação intermediária na superfície periférica cônica	s	+		PGAsl
ABS	F	Valor absoluto (quantia)		+	+	PGAsl
AC	K	Especificação de dimensão absoluta de coordenadas/posições	s	+		PGsl
ACC	K	Controle da atual aceleração axial	m	+	+	PGsl
ACCLIMA	K	Controle da atual aceleração axial máxima	m	+	+	PGAsl
ACN	K	Especificação de dimensão absoluta para eixos rotativos, aproximação da posição no sentido negativo	s	+		PGsl
ACOS	F	Arco cosseno (função. trigonométrica)		+	+	PGAsl
ACP	K	Especificação de dimensão absoluta para eixos rotativos, aproximação da posição no sentido positivo	s	+		PGsl
ACTBLOCNO	P	Retorna o atual número de bloco de um bloco de alarme, mesmo se "ocultar atual exibição de blocos" (DISPLOF) estiver ativo!		+		PGAsl
ADDFRAME	F	Inclusão e eventual ativação de um Frame medido		+	-	PGAsl, FB1sl (K2)
ADIS	A	Distância de suavização para funções de trajetória G1, G2, G3, ...	m	+		PGsl
ADISPOS	A	Distância de suavização para avanço rápido G0	m	+		PGsl
ADISPOSA	P	Tamanho da janela de tolerância para IPOBRKA	m	+	+	PGAsl
ALF	A	Ângulo de retração rápida	m	+		PGAsl
AMIRROR	G	Espelhamento programável	s	+		PGsl
AND	K	"E" lógico		+		PGAsl
ANG	A	Ângulo de sucessão de elementos de contorno	s	+		PGsl
AP	A	Ângulo polar	m/b	+		PGsl
APR	K	Proteção de acesso para leitura / exibição		+		PGAsl
APRB	K	Direito de acesso para leitura, BTSS		+		PGAsl
APRP	K	Direito de acesso para leitura, programa de peça		+		PGAsl
APW	K	Proteção de acesso para gravação		+		PGAsl
APWB	K	Direito de acesso para gravação, BTSS		+		PGAsl
APWP	K	Direito de acesso para gravação, programa de peça		+		PGAsl
APX	K	Definição da proteção de acesso para a execução do elemento de linguagem indicado		+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
AR	A	Ângulo de abertura	m/b	+		PGsl
AROT	G	Rotação programável	s	+		PGsl
AROTS	G	Rotações de Frame programáveis com ângulos espaciais	s	+		PGsl
AS	K	Definição de macro		+		PGAsl
ASCALE	G	Escala programável	s	+		PGsl
ASIN	F	Função de cálculo, arco seno		+	+	PGAsl
ASPLINE	G	Akima-Spline	m	+		PGAsl
ATAN2	F	Arco tangente 2		+	+	PGAsl
ATOL	A	tolerância específica de eixo para funções de compressor, suavização de orientação e tipos de suavização	M	+		PGAsl
ATRANS	G	Deslocamento de ponto zero programável aditivo	s	+		PGsl
AUXFUDEL	P	excluir da lista global a função de ajuda específica de canal		+	-	FB1sl (H2)
AUXFUDELG	P	excluir da lista global todas funções de ajuda de um grupo de funções de ajuda específico de canal		+	-	FB1sl (H2)
AUXFUMSEQ	P	determinar sequência de visualização para funções de ajuda - M		+	-	FB1sl (H2)
AUXFUSYNC	P	gerar, da lista global de funções de ajuda, um completo bloco de programa de usinagem para SERUPRO-ENDE-ASUP como String		+	-	FB1sl (H2)
AX	K	Identificador de eixo variável	m/b	+		PGAsl
AXCTSWE	P	Rotação de contentor de eixo		+	-	PGAsl
AXCTSWEC	P	Liberação para cancelar a rotação de contentor de eixos		+	+	PGAsl
AXCTSWED	P	Rotação de contentor de eixos (variante de comando para a colocação em funcionamento!)		+	-	PGAsl
AXIS	K	Identificador de eixo, endereço de eixo		+		PGAsl
AXNAME	F	Converte a String de entrada em identificador de eixo		+	-	PGAsl
AXSTRING	F	Converte a String em número de fuso		+	-	PGAsl
AXTOCHAN	P	Solicita o eixo para um determinado canal. É possível a partir do programa NC e da ação sincronizada.		+	+	PGAsl
AXTOSPI	F	converte o identificador de eixo em um índice de fuso		+	-	PGAsl
B	A	Nome de eixo	m/b	+		PGAsl
B2	A	Orientação da ferramenta: Ângulo RPY ou ângulo euleriano	s	+		PGAsl
B3	A	Orientação da ferramenta: . Componente do vetor de direção	s	+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
B4	A	Orientação da ferramenta: 2. Componente do vetor normal de superfície no início do bloco	s	+		PGAsl
B5	A	Orientação da ferramenta: 2. Componente do vetor normal de superfície no fim do bloco	s	+		PGAsl
B6	A	Orientação da ferramenta: 2. Componente do vetor de direção para o eixo de rotação do cone	s	+		PGAsl
B7	A	Orientação da ferramenta: 2. Componente do vetor para orientação intermediária na superfície periférica cônica	s	+		PGAsl
B_AND	O	"E" por Bits		+		PGAsl
B_OR	O	"OU" por Bits		+		PGAsl
B_NOT	O	Negação por Bits		+		PGAsl
B_XOR	O	"OU" exclusivo por Bits		+		PGAsl
BAUTO	G	Definição do primeiro segmento Spline através dos 3 pontos seguintes	m	+		PGAsl
BLOCK	K	Definição junto com a palavra-chave TO a parte de programa que deve ser executada em uma execução de subrotina indireta		+		PGAsl
BLSYNC	K	O processamento da rotina de interrupção apenas deve começar com a próxima mudança de blocos		+		PGAsl
BNAT [6]	G	Transição natural para o primeiro bloco de Spline	m	+		PGAsl
BOOL	K	Tipo de dado: Valores lógicos TRUE/FALSE ou 1/0		+		PGAsl
BOUND	F	Controla se o valor está dentro da faixa de valores definida. A igualdade retorna o valor de controle.		+	+	PGAsl
BRISK [6]	G	Aceleração de trajetória de forma brusca	m	+		PGAsl
BRISKA	P	Ativação da aceleração de trajetória brusca para os eixos programados		+	-	PGAsl
BSPLINE	G	B-Spline	m	+		PGAsl
BTAN	G	Transição tangencial para o primeiro bloco Spline	m	+		PGAsl
C	A	Nome de eixo	m/b	+		PGAsl
C2	A	Orientação da ferramenta: Ângulo RPY ou ângulo euleriano	s	+		PGAsl
C3	A	Orientação da ferramenta: 3. Componente do vetor de direção	s	+		PGAsl
C4	A	Orientação da ferramenta: 3. Componente do vetor normal de superfície no início do bloco	s	+		PGAsl
C5	A	Orientação da ferramenta: 3. Componente do vetor normal de superfície no fim do bloco	s	+		PGAsl
C6	A	Orientação da ferramenta: 3. Componente do vetor de direção para o eixo de rotação do cone	s	+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
C7	A	Orientação da ferramenta: 3. Componente do vetor para orientação intermediária na superfície periférica cônica	s	+		PGAsl
CAC	K	Aproximação absoluta de uma posição		+		PGAsl
CACN	K	O valor armazenado na tabela é aproximado de forma absoluta em sentido negativo		+		PGAsl
CACP	K	O valor armazenado na tabela é aproximado de forma absoluta em sentido positivo		+		PGAsl
CALCDAT	F	Calcula o raio e o centro de um círculo a partir de 3 ou 4 pontos		+	-	PGAsl
CALCPOSI	F	Verificação quanto à violação da área de proteção, limite da área de trabalho e limites de software		+	-	PGAsl
CALL	K	Chamada de subrotina indireta		+		PGAsl
CALLPATH	P	Caminho programável de localização para chamada de subrotinas		+	-	PGAsl
CANCEL	P	Cancelamento de ação síncrona modal		+	-	FBSYsl
CASE	K	Bifurcação de programa condicionada		+		PGAsl
CDC	K	Aproximação direta de uma posição		+		PGAsl
CDOF [6]	G	Desligar monitoramento de colisão	m	+		PGsl
CDOF2	G	Desligar monitoramento de colisão, no fresamento periférico 3D	m	+		PGsl
CDON	G	Ligar monitoramento de colisão	m	+		PGsl
CFC [6]	G	Avanço constante no contorno	m	+		PGsl
CFIN	G	Avanço constante somente para curvaturas internas, não para curvaturas externas	m	+		PGsl
CFINE	F	Atribuição do deslocamento fino em uma variável FRAME		+	-	PGAsl
CFTCP	G	Avanço constante no ponto de referência de corte da ferramenta, trajetória do centro	m	+		PGsl
CHAN	K	Especificação da área de validade de dados		+		PGAsl
CHANDATA	P	Ajuste do número de canal para acesso de dados no canal		+	-	PGAsl
CHAR	K	Tipo de dado: Caracteres ASCII		+		PGAsl
CHF	A	Chanfro; valor = comprimento do chanfro	s	+		PGsl
CHKDM	F	Controle da condição inequívoca dentro de um magazine		+	-	FBWsl
CHKDNO	F	Verificação de condição inequívoca de números D		+	-	PGAsl
CHR	A	Chanfro; valor = comprimento do chanfro no sentido de movimento		+		PGsl
CIC	K	Aproximação incremental de uma posição		+		PGAsl
CIP	G	Interpolação circular através do ponto intermediário	m	+		PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
CLEARM	P	Reset de um ou vários marcadores para coordenação de canal		+	+	PGAsl
CLRINT	P	Cancelamento de Interrupt		+	-	PGAsl
CMIRROR	F	Espelhamento em um eixo de coordenadas		+	-	PGAsl
COARSEA	K	Fim de movimento ao alcançar a "Parada exata aproximada"	m	+		PGAsl
COLLPAIR	F	Comprovar que pertence a um par de colisão		+		PGAsl
COMPCAD	G	Ligar função de compressor COMPCAD	m	+		PGAsl
COMPCURV	G	Ligar a função de compressor COMPCURV	m	+		PGAsl
COMPLETE		Instrução do comando para a saída e entrada de dados		+		PGAsl
COMPOF [6]	G	Desligar compressão de blocos NC	m	+		PGAsl
COMPON	G	Ligar a função de compressor COMPON	m	+		PGAsl
COMPSURF	G	Ligar a função de compressor COMPSURF	m	+		PGAsl
CONTDCON	P	Ligar decodificação de contorno em formato de tabela		+	-	PGAsl
CONTPRON	P	Ativação da preparação de referência		+	-	PGAsl
CORROF	P	Todos os movimentos sobrepostos ativos são cancelados.		+	-	PGsl
CORRTRAFO	F	Modificar vetores offset ou vetores de direção dos eixos de orientação no modelo cinemático da máquina		+	-	PGAsl
COS	F	Cosseno (trigon. trigonométrica)		+	+	PGAsl
COUPDEF	P	Definição do grupo ELG / grupo de fusos síncronos		+	-	PGAsl
COUPDEL	P	Exclusão do grupo ELG		+	-	PGAsl
COUPOF	P	Grupo ELG / Par de fusos sincronizados		+	-	PGAsl
COUPOFS	P	Desativação do grupo ELG / par de fusos síncronos com parada do fuso escravo		+	-	PGAsl
COUPON	P	Ligar grupo ELG / par de fusos sincronizados		+	-	PGAsl
COUPONC	P	Aceitação da ativação do grupo ELG / par de fusos síncronos com programação precedente		+	-	PGAsl
COUPRES	P	Reset do grupo ELG		+	-	PGAsl
CP [6]	G	Movimento de percurso	m	+		PGAsl
CPBC	K	acoplamento genérico: Critério de mudança de blocos		+	+	FB3sl (M3)
CPDEF	K	acoplamento genérico: Criação de um modulo de acoplamento		+	+	FB3sl (M3)
CPDEL	K	acoplamento genérico: Exclusão de um modulo de acoplamento		+	+	FB3sl (M3)
CPFMOF	K	acoplamento genérico: Ação do eixo escravo na desativação total		+	+	FB3sl (M3)
CPFMON	K	acoplamento genérico: Ação do eixo escravo na ativação		+	+	FB3sl (M3)

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
CPFMSON	K	acoplamento genérico: modo de sincronização		+	+	FB3sl (M3)
CPFPOS	K	acoplamento genérico: posição síncrona do eixo escravo		+	+	FB3sl (M3)
CPFRS	K	acoplamento genérico: sistema de referência de coordenadas		+	+	FB3sl (M3)
CPLA	K	acoplamento genérico: Definição de um eixo mestre		+	-	FB3sl (M3)
CPLCTID	K	acoplamento genérico: Número da tabela de curvas		+	+	FB3sl (M3)
CPLDEF	K	acoplamento genérico: Definição de um eixo mestre e criação de um modulo de acoplamento		+	+	FB3sl (M3)
CPLDEL	K	acoplamento genérico: Exclusão de um eixo mestre de um modulo de acoplamento		+	+	FB3sl (M3)
CPLDEN	K	acoplamento genérico: Denominador de um fator de acoplagem		+	+	FB3sl (M3)
CPLINSC	K	acoplamento genérico: Fator de escala para o valor inicial do eixo-mestre		+	+	FB3sl (M3)
CPLINTR	K	acoplamento genérico: Valor de deslocamento para o valor inicial do eixo-mestre		+	+	FB3sl (M3)
CPLNUM	K	acoplamento genérico: Nominador do fator de acoplagem		+	+	FB3sl (M3)
CPLOF	K	acoplamento genérico: Desativação de um eixo mestre de um modulo de acoplagem		+	+	FB3sl (M3)
CPLON	K	acoplamento genérico: ativação de um eixo mestre de um modulo de acoplagem		+	+	FB3sl (M3)
CPLOUTSC	K	acoplamento genérico: fator de escala para o valor final de um acoplamento		+	+	FB3sl (M3)
CPLOUTTR	K	acoplamento genérico: valor de deslocamento para o valor final de um acoplamento		+	+	FB3sl (M3)
CPLPOS	K	acoplamento genérico: posição síncrona do eixo mestre		+	+	FB3sl (M3)
CPLSETVAL	K	acoplamento genérico: referência de acoplagem		+	+	FB3sl (M3)
CPMALARM	K	acoplamento genérico: supressão de saída do alarme especialmente referente à acoplagem		+	+	FB3sl (M3)
CPMBRAKE	K	acoplamento genérico: Ação do eixo escrava em específicos sinais e comandos de parada		+	-	FB3sl (M3)
CPMPRT	K	acoplamento genérico: Ação do acoplamento no início do programa de usinagem em bloco via teste de programa		+	+	FB3sl (M3)
CPMRESET	K	acoplamento genérico: ação de acoplagem no RESET		+	+	FB3sl (M3)
CPMSTART	K	acoplamento genérico: ação de acoplagem no inicio do programa de usinagem		+	+	FB3sl (M3)

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
CPMVDI	K	acoplamento genérico: Ação do eixo escravo de específicos sinais de interface NC/PLC		+	+	FB3sl (M3)
CPOF	K	acoplamento genérico: Desativação de modulo de acoplagem		+	+	FB3sl (M3)
CPON	K	acoplamento genérico: ativação modulo de acoplagem		+	+	FB3sl (M3)
CPRECOF [6]	G	Desativação da precisão de contorno programável	m	+		PGAsl
CPRECON	G	Ativação da precisão de contorno programável	m	+		PGAsl
CPRES	K	acoplamento genérico: Ativa os dados projetados do acoplamento síncrono		+	-	FB3sl (M3)
CPROT	P	Área de proteção específica de canal ON/OFF		+	-	PGAsl
CPROTDEF	P	Definição de uma área de proteção específica de canal		+	-	PGAsl
CPSETTYPE	K	acoplamento genérico: Tipo de acoplamento		+	+	FB3sl (M3)
CPSYNCOP	K	acoplamento genérico: limiar para o andamento síncrono de posição "amplo"		+	+	FB3sl (M3)
CPSYNCOP2	K	acoplamento genérico: limiar para o andamento síncrono de posição "amplo" 2		+	+	FB3sl (M3)
CPSYNCOV	K	acoplamento genérico: Limiar para o andamento síncrono de velocidade "amplo"		+	+	FB3sl (M3)
CPSYNFIP	K	acoplamento genérico: limiar para o andamento síncrono de posição "suave"		+	+	FB3sl (M3)
CPSYNFIP2	K	acoplamento genérico: limiar para o andamento síncrono de posição "suave" 2		+	+	FB3sl (M3)
CPSYNFIV	K	acoplamento genérico: Limiar para o andamento síncrono de velocidade "suave"		+	+	FB3sl (M3)
CR	A	Raio do círculo	s	+		PGsl
CROT	F	Rotação do atual sistema de coordenadas		+	-	PGAsl
CROTS	F	Rotações de Frames programáveis com ângulos espaciais (rotações nos eixos especificados)	s	+	-	PGsl
CRPL	F	Rotação de Frame em um plano qualquer		+	-	FB1sl (K2)
CSCALE	F	Fator de escala para vários eixos		+	-	PGAsl
CSPLINE	F	Spline cúbica	m	+		PGAsl
CT	G	Círculo com transição tangencial	m	+		PGsl
CTAB	F	Posição de eixo escravo determinada com base na posição do eixo mestre a partir da tabela de curvas		+	+	PGAsl
CTABDEF	P	Ligar definição de tabela		+	-	PGAsl
CTABDEL	P	Eliminação de tabela de curvas		+	-	PGAsl
CTABEND	P	Desligar definição de tabela		+	-	PGAsl
CTABEXISTS	F	Verifica a tabela de curva de número n		+	+	PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
CTABFNO	F	Número de tabelas de curvas possíveis na memória		+	+	PGAsl
CTABFPOL	F	Número de polinômios possíveis na memória		+	+	PGAsl
CTABFSEG	F	Número de segmentos de curva possíveis na memória		+	+	PGAsl
CTABID	F	Fornece o número de tabela da tabela de curvas n		+	+	PGAsl
CTABINV	F	Posição de eixo mestre determinada com base na posição do eixo escravo a partir da tabela de curvas		+	+	PGAsl
CTABISLOCK	F	Retorna o estado de bloqueio da tabela de curvas de número n		+	+	PGAsl
CTABLOCK	P	Eliminação e sobrescrição, bloqueio		+	+	PGAsl
CTABMEMTYP	F	Retorna a memória em que está armazenada a tabela de curva de número n.		+	+	PGAsl
CTABMPOL	F	Número máximo de polinômios possíveis na memória		+	+	PGAsl
CTABMSEG	F	Número máximo de segmentos de curva possíveis na memória		+	+	PGAsl
CTABNO	F	Número de tabelas de curvas definidas na SRAM ou DRAM		+	+	FB3sl (M3)
CTABNOMEM	F	Número de tabelas de curvas definidas na SRAM ou DRAM		+	+	PGAsl
CTABPERIOD	F	Retorna a periodicidade de tabela da tabela de curvas de número n		+	+	PGAsl
CTABPOL	F	Número de polinômios efetivamente utilizados na memória		+	+	PGAsl
CTABPOLID	F	Número de polinômios de curvas utilizados pela tabela de curva de número n		+	+	PGAsl
CTABSEG	F	Número de segmentos de curva efetivamente utilizados na memória		+	+	PGAsl
CTABSEGID	F	Número de segmentos de curva utilizados na tabela de curva de número n		+	+	PGAsl
CTABSEV	F	Fornece o valor final do eixo escravo de um segmento da tabela de curva		+	+	PGAsl
CTABSSV	F	Fornece o valor inicial do eixo escravo de um segmento da tabela de curvas		+	+	PGAsl
CTABTEP	F	Fornece o valor do eixo mestre no fim da tabela de curvas		+	+	PGAsl
CTABTEV	F	Fornece o valor do eixo escravo no fim da tabela de curvas		+	+	PGAsl
CTABTMAX	F	Fornece o valor máximo do eixo escravo da tabela de curvas		+	+	PGAsl
CTABTMIN	F	Fornece o valor mínimo do eixo escravo da tabela de curvas		+	+	PGAsl
CTABTSP	F	Fornece o valor do eixo mestre no início da tabela de curvas		+	+	PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
CTABTSV	F	Fornece o valor do eixo escravo no início da tabela de curvas		+	+	PGAsl
CTABUNLOCK	P	Cancelamento do bloqueio contra eliminação e sobrescrição		+	+	PGAsl
CTOL	A	Tolerância de contorno para funções de compressor, suavização de orientação e tipos de suavização	M	+		PGAsl
CTRANS	F	Deslocamento de ponto zero para vários eixos		+	-	PGAsl
CUT2D [6]	G	Corretor do raio de ferramenta 2D	m	+		PGsl
CUT2DD	G	Corretor do raio de ferramenta 2½D relacionada a uma ferramenta de diferença	m	+		PGsl
CUT2DF	G	Corretor de ferramenta 2D atuam de forma relativa ao atual Frame (plano inclinado)	m	+		PGsl
CUT2DFD	G	Corretor do raio de ferramenta 2½D relacionada a uma ferramenta de diferença, relativa ao atual Frame (plano inclinado)	m	+		PGsl
CUT3DC	G	Corretor do raio de ferramenta 3D para fresamento periféricos	m	+		PGAsl
CUT3DCC	G	Corretor do raio da ferramenta 3D para fresamento periférico considerando uma superfície de limitação com corretor de raio 3D: Contorno na superfície de usinagem	m	+		PGAsl
CUT3DCCD	G	Corretor do raio da ferramenta 3D para fresamento periférico considerando uma superfície de limitação com ferramenta diferencial na trajetória do centro da ferramenta: Penetração até a superfície de limitação	m	+		PGAsl
CUT3DCD	G	Em uma ferramenta relacionada a corretor do raio da ferramenta 3D para fresamento periférico	m	+		PGAsl
CUT3DF	G	Corretor do raio de ferramenta 3D para fresamento de topo com alteração de orientação	m	+		PGAsl
CUT3DFD	G	Em uma ferramenta relacionada a corretor do raio da ferramenta 3D para fresamento de topo com alteração de orientação	m	+		PGAsl
CUT3DFF	G	Corretor do raio de ferramenta 3D para fresamento de topo com orientação constante. A orientação da ferramenta é definida pelo G17 - G19 e, se necessário, através de um sentido girado por um Frame.	m	+		PGAsl
CUT3DFS	G	Corretor do raio de ferramenta 3D para fresamento de topo com orientação constante. A orientação da ferramenta é definida pelo G17 - G19 e não influenciada por Frames.	m	+		PGAsl
CUTCONOF [6]	G	Desligar correção de raio constante	m	+		PGsl
CUTCONON	G	Ligar correção de raio constante	m	+		PGsl
CUTMOD	A	Acionar a modificação dos dados de correção em ferramentas giratórias (em conexão com porta-ferramentas orientáveis)	m	+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
CUTMODK	A	Acionar a modificação dos dados de correção em ferramentas giratórias (em conexão com transformações de orientação que foram definidas por meio de sequências cinemáticas)	m	+		PGAsl
CYCLE60	C (T)	Ciclo de gravação		+		PGAsl
CYCLE61	C (T)	Fresamento de facear		+		PGAsl
CYCLE62	C (T)	Chamada de contorno		+		PGAsl
CYCLE63	C (T)	Fresamento de bolsão de contorno		+		PGAsl
CYCLE64	C (T)	"Pré-furação" do bolsão de contorno		+		PGAsl
CYCLE70	C (T)	Fresamento de roscas		+		PGAsl
CYCLE72	C (T)	Fresamento de percurso		+		PGAsl
CYCLE76	C (T)	Fresamento de saliências retangulares		+		PGAsl
CYCLE77	C (T)	Fresamento de saliências circulares		+		PGAsl
CYCLE78	C (T)	Fresas de rosqueamento e furação		+		PGAsl
CYCLE79	C (T)	Poliedro		+		PGAsl
CYCLE81	C (T)	Furação, centragem		+		PGAsl
CYCLE82	C (T)	Furação, escareamento plano		+		PGAsl
CYCLE83	C (T)	Furação profunda		+		PGAsl
CYCLE84	C (T)	Rosqueamento com macho sem mandril de compensação		+		PGAsl
CYCLE85	C (T)	Alargamento		+		PGAsl
CYCLE86	C (T)	Mandrilamento		+		PGAsl
CYCLE92	C (T)	Separação		+		PGAsl
CYCLE95	C (T)	Desbaste do contorno		+		PGAsl
CYCLE98	C (T)	Sequência de roscas		+		PGAsl
CYCLE99	C (T)	Rosqueamento		+		PGAsl
CYCLE150	C (M)	Indicar / protocolar os resultados da medição		+		BNMsl
CYCLE435	C (T)	Calcular a posição do retificador		+		PGAsl
CYCLE495	C (T)	Perfilamento		+		PGAsl
CYCLE750	C (A)	Ciclo de trabalho interno para CYCLE751 ... CYCLE759 (é recebido pelo comando MMC para solicitar a função real)		-		FB3sl (T4)
CYCLE751	C (A)	Abrir / executar / fechar sessão de otimização		M		FB3sl (T4)
CYCLE752	C (A)	Acrescentar eixo à sessão de otimização		M		FB3sl(T4)
CYCLE753	C (A)	Selecionar modo de otimização		M		FB3sl (T4)
CYCLE754	C (A)	Acrescentar / remover registro		M		FB3sl (T4)
CYCLE755	C (A)	Salvar / restaurar registro		M		FB3sl (T4)
CYCLE756	C (A)	Ativar resultados de otimização		M		FB3sl (T4)
CYCLE757	C (A)	Salvar dados de otimização		M		FB3sl (T4)
CYCLE758	C (A)	Alterar valores de parâmetro		M		FB3sl (T4)
CYCLE759	C (A)	Ler valores de parâmetro		M		FB3sl (T4)
CYCLE800	C (T)	Rotação		+		PGAsl
CYCLE801	C (T)	Grade ou Quadro		+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
CYCLE802	C (T)	Posição definida pelo usuário		+		PGAsl
CYCLE830	C (T)	Furação de furo profundo 2		+		PGAsl
CYCLE832	C (T)	High Speed Settings		+		PGAsl
CYCLE840	C (T)	Rosqueamento com macho com mandril de compensação		+		PGAsl
CYCLE899	C (T)	fresamento de ranhuras abertas		+		PGAsl
CYCLE930	C (T)	Canal		+		PGAsl
CYCLE940	C (T)	formas em baixo relevo		+		PGAsl
CYCLE951	C (T)	Desbaste		+		PGAsl
CYCLE952	C (T)	Usinagem de canal de contorno		+		PGAsl
CYCLE961	C (M)	Determinar a posição de um canto da peça de trabalho (interno ou externo) e aplicar como deslocamento do ponto zero		+		BNMsl
CYCLE971	C (M)	Calibrar a sonda da ferramenta, medir o comprimento da ferramenta e/ou o raio da ferramenta (somente para a tecnologia de fresa)		+		BNMsl
CYCLE973	C (M)	Calibrar a sonda da ferramenta em uma superfície da peça de trabalho ou em uma ranhura (somente para a tecnologia de torno)		+		BNMsl
CYCLE974	C (M)	Determinar o ponto zero da peça de trabalho no eixo de medição selecionado, determinar a correção da ferramenta com a medição de 1 ponto (somente para a tecnologia de torno)		+		BNMsl
CYCLE976	C (M)	Calibrar a sonda da peça de trabalho em um anel de calibração ou em uma esfera de calibração totalmente no plano de trabalho ou em um canto para um determinado eixo e sentido		+		BNMsl
CYCLE977	C (M)	Determinar o ponto central no plano e a largura ou o diâmetro		+		BNMsl
CYCLE978	C (M)	Medir a posição de um canto no sistema de coordenadas da peça de trabalho		+		BNMsl
CYCLE979	C (M)	Determinar o ponto central em um plano, medir o raio dos segmentos circulares		+		BNMsl
CYCLE982	C (M)	Calibrar a sonda da ferramenta, dimensionar as ferramentas de torno, perfuração e fresa (somente para a tecnologia de torno)		+		BNMsl
CYCLE994	C (M)	Determinar o ponto zero da peça de trabalho no eixo de medição selecionado com a medição de 2 pontos (somente para a tecnologia de torno)		+		BNMsl
CYCLE995	C (M)	Medir a angularidade do fuso em uma máquina de ferramenta		+		BNMsl
CYCLE996	C (M)	Determinar os dados relevantes a transformação para as transformações cinemáticas com os eixos rotativos		+		BNMsl
CYCLE997	C (M)	Determinar o ponto central e o diâmetro de uma esfera, medir os pontos centrais de três esferas distribuídas		+		BNMsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
CYCLE998	C (M)	Determinar a posição angular de uma superfície (plana) com relação ao plano de trabalho, determinar os ângulos dos cantos no sistema de coordenadas da peça de trabalho		+		BNMsl
CYCLE4071	C (T)	Retificação longitudinal com penetração no ponto de inversão		+		PGAsl
CYCLE4072	C (T)	Retificação longitudinal com penetração no ponto de inversão e sinal de interrupção		+		PGAsl
CYCLE4073	C (T)	Retificação longitudinal com penetração contínua		+		PGAsl
CYCLE4074	C (T)	Retificação longitudinal com penetração contínua e sinal de interrupção		+		PGAsl
CYCLE4075	C (T)	Retificação plana com penetração no ponto de inversão		+		PGAsl
CYCLE4077	C (T)	Retificação plana com penetração no ponto de inversão e sinal de interrupção		+		PGAsl
CYCLE4078	C (T)	Retificação plana com penetração contínua		+		PGAsl
CYCLE4079	C (T)	Retificação plana com penetração intermitente		+		PGAsl

Instruções D ... F

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
D	A	Número de correção da ferramenta		+		PGsl
D0	A	Com D0 as correções da ferramenta estão inativas		+		PGsl
DAC	K	Programação em diâmetro específica de eixo, absoluta e por blocos	s	+		PGsl
DC	K	Dimensão absoluta para eixos rotativos, aproximação direta da posição	s	+		PGsl
DCI	K	Organizar a classe de dados I (= Individual) (apenas SINUMERIK 828D!)		+		PGAsl
DCM	K	Organizar a classe de dados M (= fabricante) (apenas SINUMERIK 828D!)		+		PGAsl
DCU	K	Organizar a classe de dados U (= Usuário) (apenas SINUMERIK 828D!)		+		PGAsl
DEF	K	Definição de variáveis		+		PGAsl
DEFAULT	K	Deriva na bifurcação CASE		+		PGAsl
DEFINE	K	Palavra-chave para definições de macro		+		PGAsl
DELAYFSTOF	P	Definição do fim de uma área Stop-Delay	m	+	-	PGAsl
DELAYFSTON	P	Definição do início de uma área Stop-Delay	m	+	-	PGAsl
DELDL	F	Eliminação de correções aditivas		+	-	PGAsl
DELDTG	P	Anulação de curso restante		-	+	FBSYsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1][2][3][4][5] Explicações ver legenda (Página 902).						
DELETE	P	Deleta o arquivo especificado. O nome do arquivo pode ser especificado com caminho e extensão de arquivo.		+	-	PGAsl
DELMLOWNER	F	deletar local do magazine do proprietário da ferramenta		+	-	FBWsl
DELMLRES	F	deletar local de reserva do magazine		+	-	FBWsl
DELMT	P	Exclusão de Multitool		+	-	FBWsl
DELOBJ	F	Supressão de elementos de cadeias cinemáticas, áreas de proteção, elementos da área de proteção, pares de colisão e dados de transformação		+		PGAsl
DELT	P	Apagar ferramenta		+	-	FBWsl
DELTC	P	deletar registro do porta ferramenta		+	-	FBWsl
DELTOOLENV	F	Eliminação de blocos de dados para descrição de ambientes de ferramentas		+	-	PGAsl
DIACYCOFA	K	Programação em diâmetro específica de eixo ativa modalmente: OFF em ciclos	m	+		FB1sl (P1)
DIAM90	G	Programação em diâmetro para G90, programação em raio para G91	m	+		PGAsl
DIAM90A	K	Programação em diâmetro específica de eixo ativa modalmente para G90 e AC, e em raio para G91 e IC	m	+		PGsl
DIAMCHAN	K	Aceitação de todos eixos a partir de funções de eixo de dado de máquina no estado de canal da programação em diâmetro		+		PGsl
DIAMCHANA	K	Aceitação do estado de canal da programação em diâmetro		+		PGsl
DIAMCYCOF	G	Programação em diâmetro específica de canal: OFF em ciclos	m	+		FB1sl (P1)
DIAMOF [6]	G	Programação em diâmetro: OFF Para posição inicial, vejas as informações do fabricante da máquina	m	+		PGsl
DIAMOFA	K	Programação em diâmetro específica de eixo ativa modalmente: OFF Para posição inicial, vejas as informações do fabricante da máquina	m	+		PGsl
DIAMON	G	Programação em diâmetro: ON	m	+		PGsl
DIAMONA	K	Programação em diâmetro específica de eixo ativa modalmente: ON Para ativação, veja as informações do fabricante da máquina	m	+		PGsl
DIC	K	Programação em diâmetro específica de eixo, relativa e por blocos	s	+		PGsl
DILF	A	Curso de retrocesso (comprimento)	m	+		PGsl
DISABLE	P	Interrupt OFF		+	-	PGAsl
DISC	A	Aceleração do círculo de transição, compensação do raio da ferramenta	m	+		PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1][2][3][4][5] Explicações ver legenda (Página 902).						
DISCL	A	Distância do ponto final do movimento de penetração rápido a partir do plano de usinagem		+		PGsl
DISPLOF	PA	Supressão da atual exibição de blocos		+		PGAsl
DISPLON	PA	Cancelamento da supressão da atual exibição de blocos		+		PGAsl
DISPR	A	Diferença de percurso de reposicionamento	s	+		PGAsl
DISR	A	Distância de reposicionamento	s	+		PGAsl
DISRP	A	distância do plano de retração do plano de usinagem na aproximação e no afastamento suave		+		PGsl
DITE	A	Curso de saída da rosca	m	+		PGsl
DITS	A	Curso de entrada da rosca	m	+		PGsl
DIV	K	Divisão Integer		+		PGAsl
DL	A	Seleção de corretor de ferramenta aditivo dependente de local (DL, corretor aditivo, corretor de ajuste)	m	+		PGAsl
DO	A	Palavra-chave para ação síncrona, ativa a ação quando a condição for preenchida		-	+	FBSYsl
DRFOF	P	Desativação dos deslocamentos com manivela eletrônica (DRF)	m	+	-	PGsl
DRIVE	G	Aceleração de trajetória dependente da velocidade	m	+		PGAsl
DRIVEA	P	Ativação da curva característica de aceleração dobrada para os eixos programados		+	-	PGAsl
DYNFINISH	G	Dinâmica para acabamento fino	m	+		PGAsl
DYNNORM [6]	G	Dinâmica normal	m	+		PGAsl
DYNPOS	G	Dinâmica para modo de posicionamento, rosqueamento com macho	m	+		PGAsl
DYNROUGH	G	Dinâmica para desbaste	m	+		PGAsl
DYNSEMIFIN	G	Dinâmica para acabamento	m	+		PGAsl
DZERO	P	Marca todos números D da unidade TO como inválidos		+	-	PGAsl
EAUTO	G	Definição do último segmento Spline através dos últimos 3 pontos	m	+		PGAsl
EGDEF	P	Definição de uma caixa de transmissão eletrônica		+	-	PGAsl
EGDEL	P	Eliminação da definição de acoplamento para o eixo escravo		+	-	PGAsl
EGOFC	P	Desativação contínua da caixa de transmissão eletrônica		+	-	PGAsl
EGOFS	P	Desativação seletiva da caixa de transmissão eletrônica		+	-	PGAsl
EGON	P	Ativação da caixa de transmissão eletrônica		+	-	PGAsl
EGONSYN	P	Ativação da caixa de transmissão eletrônica		+	-	PGAsl
EGONSYNE	P	Ativação da caixa de transmissão eletrônica, com especificação do modo de aproximação		+	-	PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1][2][3][4][5] Explicações ver legenda (Página 902).						
ELSE	K	Bifurcação do programa, se a condição IF não for preenchida		+		PGAsl
ENABLE	P	Interrupt ON		+	-	PGAsl
ENAT [6]	G	Transição de curvas natural para o próximo bloco de deslocamento	m	+		PGAsl
ENDFOR	K	Linha final do loop de contagem FOR		+		PGAsl
ENDIF	K	Linha final da bifurcação IF		+		PGAsl
ENDLABEL	K	Marcador final para repetições de programa de peça através do REPEAT		+		PGAsl, FB1sl (K1)
ENDLOOP	K	Linha final do loop contínuo de programa LOOP		+		PGAsl
ENDPROC	K	Linha final de um programa com linha inicial PROC		+		
ENDWHILE	K	Linha final do loop WHILE		+		PGAsl
ESRR	P	Parametrização do retrocesso independente de acionamento ESR no acionamento		+		PGAsl
ESRS	P	Parametrização da parada independente de acionamento ESR no acionamento		+		PGAsl
ETAN	G	Transição tangencial de curva para o próximo bloco de deslocamento no início de Spline	m	+		PGAsl
EVERY	K	Execução de ação síncrona na passagem da condição de FALSE para TRUE		-	+	FBSYsl
EX	K	Palavra-chave para a atribuição de valor na forma escrita exponencial		+		PGAsl
EXECSTRING	P	Transferência de uma variável String com a linha de programa de peça a ser executada		+	-	PGAsl
EXECTAB	P	Execução de um elemento a partir de uma tabela de movimentos		+	-	PGAsl
EXECUTE	P	Execução de programa ON		+	-	PGAsl
EXP	F	Função exponencial ex		+	+	PGAsl
EXTCALL	A	Execução de subrotina externa		+	+	PGAsl
EXTCLOSE	P	Fechamento do dispositivo/arquivo aberto para executar a gravação		+	-	PGAsl
EXTERN	K	Identificação de uma subrotina com transferência de parâmetros		+		PGAsl
EXTOPEN	P	Abertura do dispositivo/arquivo externo para o canal para executar a gravação		+	-	PGAsl
F	A	Valor de avanço (em associação com G4 também é programado o tempo de espera com o F)		+	+	PGsl
FA	K	Avanço axial	m	+	+	PGsl
FAD	A	Avanço de penetração para aproximação suave e afastamento suave		+		PGsl
FALSE	K	Constante lógica: incorreto		+	+	PGAsl
FB	A	Avanço por bloco		+		PGsl
FCTDEF	P	Definição de função polinomial		+	-	PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1][2][3][4][5] Explicações ver legenda (Página 902).						
FCUB	G	Avanço variável conforme Spline cúbica	m	+		PGAsl
FD	A	Avanço de trajetória para sobreposição com manivela eletrônica	s	+		PGsl
FDA	K	Avanço axial para sobreposição de manivela eletrônica	s	+		PGsl
FENDNORM [6]	G	Desaceleração de cantos OFF	m	+		PGAsl
FFWOF [6]	G	Controle feedforward OFF	m	+		PGAsl
FFWON	G	Controle feedforward ativado	m	+		PGAsl
FGREF	K	Raio de referência em eixos rotativos ou fatores de referência de trajetória em eixos de orientação (interpolação de vetores)	m	+		PGsl
FGROUP	P	Definição dos eixos com avanço de trajetória		+	-	PGsl
FI	K	Parâmetro para acesso aos dados de Frame: Deslocamento fino		+		PGAsl
FIFOCTRL	G	Controle da memória de pré-processamento	m	+		PGAsl
FILEDATE	P	Retorna a data do último acesso de gravação do arquivo		+	-	PGAsl
FILEINFO	P	Retorna a soma de FILEDATE, FILESIZE, FILESTAT e FILETIME juntos		+	-	PGAsl
FILESIZE	P	Retorna o tamanho atual do arquivo		+	-	PGAsl
FILESTAT	P	Retorna o estado de arquivo dos direitos de leitura, gravação, execução, exibição e exclusão (rwxsd)		+	-	PGAsl
FILETIME	P	Retorna o horário do último acesso de gravação do arquivo		+	-	PGAsl
FINEA	K	Fim de movimento ao alcançar a "Parada exata fina"	m	+		PGAsl
FL	K	Velocidade limite para eixos síncronos	m	+		PGsl
FLIN	G	Avanço linear modificável	m	+		PGAsl
FMA	K	Vários avanços axiais	m	+		PGsl
FNORM [6]	G	Avanço normal conforme DIN66025	m	+		PGAsl
FOC	K	Retentor de torque/força eficaz bloco a bloco	s	-	+	FBSYsl
FOCOF	K	Desativar retentor de torque/força modal	m	-	+	FBSYsl
FOCON	K	Ativar retentor de torque/força modal	m	-	+	FBSYsl
FOR	K	Loop de contagem com número fixo de passadas		+		PGAsl
FP	A	Ponto fixo: Número do ponto fixo a ser aproximado	s	+		PGsl
FPO	K	Característica de avanço programada através de um polinômio		+		PGAsl
FPR	P	Identificação do eixo rotativo		+	-	PGsl
FPRAOF	P	Desativação do avanço por rotação		+	-	PGsl
FPRAON	P	Ativação do avanço por rotação		+	-	PGsl
FRAME	K	Tipo de dado para definição de sistemas de coordenadas		+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1][2][3][4][5] Explicações ver legenda (Página 902).						
FRC	A	Avanço para raio e chanfro	s	+		PGsl
FRCM	A	Avanço para raio e chanfro modal	m	+		PGsl
FROM	K	A ação é executada quando a condição é pre-enchida uma vez e permanece ativa por toda a ação síncrona		-	+	FBSYsl
FTOC	P	Modificação da correção fina de ferramenta		-	+	FBSYsl
FTOCOF [6]	G	Correção fina de ferramenta ativa Online OFF	m	+		PGAsl
FTOCON	G	Correção fina de ferramenta ativa Online ON	m	+		PGAsl
FXS	K	Deslocamento até o encosto fixo ativado	m	+	+	PGsl
FXST	K	Limite de torque para deslocamento até o encosto fixo	m	+	+	PGsl
FXSW	K	Janela de monitoramento para deslocamento até o encosto fixo		+	+	PGsl
FZ	K	Avanço por dente	m	+		PGsl

Instruções G ... L

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
G0	G	Interpolação linear com avanço (movimento) rápido	m	+		PGsl
G1 [6]	G	Interpolação linear com avanço (interpolação de retas)	m	+		PGsl
G2	G	Interpolação circular em sentido horário	m	+		PGsl
G3	G	Interpolação circular em sentido anti-horário	m	+		PGsl
G4	G	Tempo de espera, pré-definido	s	+		PGsl
G5	G	Retificação inclinada de canal	s	+		PGAsl
G7	G	Movimento de compensação na retificação inclinada de canal	s	+		PGAsl
G9	G	Parada exata - redução de velocidade	s	+		PGsl
G17 [6]	G	Seleção do plano de trabalho X/Y	m	+		PGsl
G18	G	Seleção do plano de trabalho Z/X	m	+		PGsl
G19	G	Seleção do plano de trabalho Y/Z	m	+		PGsl
G25	G	Limite inferior da área de trabalho	s	+		PGsl
G26	G	Limite da área de trabalho superior	s	+		PGsl
G33	G	Rosqueamento com passo constante	m	+		PGsl
G34	G	Rosqueamento com passo linear e crescente	m	+		PGsl
G35	G	Rosqueamento com passo linear e decrescente	m	+		PGsl
G40 [6]	G	Correção do raio da ferramenta OFF	m	+		PGsl
G41	G	Correção do raio da ferramenta à esquerda do contorno	m	+		PGsl
G42	G	Correção do raio da ferramenta à direita do contorno	m	+		PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
G53	G	Supressão do atual deslocamento de ponto zero (por bloco)	s	+		PGsl
G54	G	1º deslocamento de ponto zero ajustável	m	+		PGsl
G55	G	2º deslocamento do ponto zero ajustável	m	+		PGsl
G56	G	3º deslocamento de ponto zero ajustável	m	+		PGsl
G57	G	4º deslocamento de ponto zero ajustável	m	+		PGsl
G58 (840D sl)	G	Deslocamento de ponto zero absoluto programável (deslocamento brusco)	s	+		PGsl
G58 (828D)	G	5º deslocamento de ponto zero ajustável	m	+		PGsl
G59 (840D sl)	G	Deslocamento de ponto zero programável aditivo (deslocamento fino)	s	+		PGsl
G59 (828D)	G	6º deslocamento de ponto zero ajustável	m	+		PGsl
G60 [6]	G	Parada exata - redução de velocidade	m	+		PGsl
G62	G	Desaceleração em cantos internos com compensação do raio da ferramenta ativa (G41, G42)	m	+		PGAsl
G63	G	Rosqueamento com macho com mandril de compensação	s	+		PGsl
G64	G	Modo de controle da trajetória	m	+		PGsl
G70	G	Especificação em polegadas para dimensões geométricas (comprimentos)	m	+	+	PGsl
G71 [6]	G	Especificação métrica para dimensões geométricas (comprimentos)	m	+	+	PGsl
G74	G	Aproximação do ponto de referência	s	+		PGsl
G75	G	Aproximação do ponto fixo	s	+		PGsl
G90 [6]	G	Especificação de dimensão absoluta	m/b	+		PGsl
G91	G	Indicação de dimensão incremental	m/b	+		PGsl
G93	G	Avanço em função do tempo 1/min (rpm)	m	+		PGsl
G94 [6]	G	Avanço linear F em mm/min ou pol./min e graus/min	m	+		PGsl
G95	G	Avanço por rotação F em mm/rot. ou pol./rot	m	+		PGsl
G96	G	Avanço por rotação (como no caso de G95) e velocidade de corte constante	m	+		PGsl
G97	G	Avanço por rotação e rotação do fuso constante (velocidade de corte constante DESLIGADA)	m	+		PGsl
G110	G	Programação polar relativa à última posição nominal programada	s	+		PGsl
G111	G	Programação polar relativa ao ponto zero do atual sistema de coordenadas da peça de trabalho	s	+		PGsl
G112	G	Programação polar relativa ao último polo válido	s	+		PGsl
G140 [6]	G	Sentido de aproximação WAB definido por G41/G42	m	+		PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
G141	G	Sentido de aproximação WAB à esquerda do contorno	m	+		PGsl
G142	G	Sentido de aproximação WAB à direita do contorno	m	+		PGsl
G143	G	Sentido de aproximação WAB em função da tangente	m	+		PGsl
G147	G	Aproximação suave em linha reta	s	+		PGsl
G148	G	Afastamento suave em linha reta	s	+		PGsl
G153	G	Supressão do atual Frame inclusive Frame básico	s	+		PGsl
G247	G	Aproximação suave em quadrante	s	+		PGsl
G248	G	Afastamento suave em quadrante	s	+		PGsl
G290 [6]	G	Comutação para modo SINUMERIK ON	m	+		FBWsl
G291	G	Comutação para modo ISO2/3 ON	m	+		FBWsl
G331	G	Rosqueamento com macho sem mandril de compensação, passo positivo, giro horário (direito)	m	+		PGsl
G332	G	Rosqueamento com macho sem mandril de compensação, passo negativo, giro anti-horário (esquerdo)	m	+		PGsl
G335	G	rotação de uma rosca boleada no sentido horário	m	+		PGsl
G336	G	girar a rosca boleada em anti-sentido horário	m	+		PGsl
G340 [6]	G	Bloco de aproximação espacial (simultâneo em profundidade e no plano (espiral))	m	+		PGsl
G341	G	Primeiro penetração no eixo perpendicular (z), depois aproximação no plano	m	+		PGsl
G347	G	Aproximação suave em semicírculo	s	+		PGsl
G348	G	Afastamento suave em semicírculo	s	+		PGsl
G450 [6]	G	Círculo de transição	m	+		PGsl
G451	G	Intersecção das equidistâncias	m	+		PGsl
G460 [6]	G	Ativação do monitoramento de colisão para bloco de aproximação e de afastamento	m	+		PGsl
G461	G	Inserção de um círculo no bloco de compensação do raio de ferramenta (WRK)	m	+		PGsl
G462	G	Inserção de uma reta no bloco de compensação do raio de ferramenta (WRK)	m	+		PGsl
G500 [6]	G	Desativação de todos os Frames ajustáveis, Frames básicos estão ativos	m	+		PGsl
G505 ... G599	G	5. ... 99. Deslocamento de ponto zero ajustável	m	+		PGsl
G601 [6]	G	Mudança de bloco com parada exata fina	m	+		PGsl
G602	G	Mudança de bloco com parada exata aproximada	m	+		PGsl
G603	G	Mudança de bloco para fim de bloco de interpolação IPO	m	+		PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
G621	G	Desaceleração de cantos em todos os cantos	m	+		PGAsl
G641	G	Modo de controle da trajetória com suavização conforme critério de percurso (= distância de suavização programável)	m	+		PGsl
G642	G	Modo de controle da trajetória com suavização com a conservação de tolerâncias definidas	m	+		PGsl
G643	G	Modo de controle da trajetória com suavização com a conservação de tolerâncias definidas (interno de bloco)	m	+		PGsl
G644	G	Modo de controle da trajetória com suavização com o máximo possível de dinâmica	m	+		PGsl
G645	G	Modo de controle da trajetória com suavização de cantos e transições de blocos tangenciais com conservação de tolerâncias definidas	m	+		PGsl
G700	G	Especificação em polegadas para dimensões geométricas e tecnológicas (comprimentos, avanço)	m	+	+	PGsl
G710 [6]	G	Especificação métrica para dimensões geométricas e tecnológicas (comprimentos, avanço)	m	+	+	PGsl
G810 [6], ..., G819	G	Grupo G reservado para o usuário OEM		+		PGAsl
G820 [6], ..., G829	G	Grupo G reservado para o usuário OEM		+		PGAsl
G931	G	Especificação de avanço pelo tempo de deslocamento, desativação de velocidade constante de percurso	m	+		
G942	G	Congelamento do avanço linear e velocidade de corte constante ou rotação de fuso	m	+		
G952	G	Congelamento do avanço por rotação e velocidade de corte constante ou rotação de fuso	m	+		
G961	G	Avanço linear (como no caso de G94) e velocidade de corte constante	m	+		PGsl
G962	G	Avanço linear ou avanço por rotação e velocidade de corte constante	m	+		PGsl
G971	G	Avanço linear e rotação do fuso constante (velocidade de corte constante DESLIGADA)	m	+		PGsl
G972	G	Avanço linear ou avanço por rotação e rotação do fuso constante (velocidade de corte constante DESLIGADA)	m	+		PGsl
G973	G	Avanço por rotação sem limitação de rotação do fuso e rotação do fuso constante (G97 sem LIMS para modo ISO)	m	+		PGsl
GEOAX	P	Atribui novos canais de eixo para os eixos geométricos 1 - 3		+	-	PGAsl
GET	P	Troca de eixos liberados entre canais		+	+	PGAsl
GETACTT	F	Determina a ferramenta ativa de um grupo de ferramentas de mesmo nome		+	-	FBWsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
GETACTTD	F	Determina para um número D absoluto seu número T correspondente		+	-	PGAsl
GETD	P	Troca de eixo direta entre canais		+	-	PGAsl
GETDNO	F	Fornece o número D de um corte (CE) de uma ferramenta (T)		+	-	PGAsl
GETEXET	P	Leitura do número T carregado		+	-	FBWsl
GETFREELOC	P	Localização de um alojamento vazio no magazine para uma ferramenta especificada		+	-	FBWsl
GETSELT	P	Fornecer números T pré-selecionados		+	-	FBWsl
GETT	F	Definir número T para nome de ferramenta		+	-	FBWsl
GETTCOR	F	Leitura de comprimentos de ferramenta ou componentes de comprimentos de ferramenta		+	-	PGAsl
GETTENV	F	Leitura de números T, D e DL		+	-	PGAsl
GETVARAP	F	Leitura do direito do acesso em uma variável de sistema/de usuário		+	-	PGAsl
GETVARDFT	F	Leitura do valor padrão de uma variável de sistema/de usuário		+	-	PGAsl
GETVARLIM	F	Leitura do valor padrão de uma variável de sistema/de usuário		+	-	PGAsl
GETVARPHU	F	Leitura da unidade física de uma variável de sistema/de usuário		+	-	PGAsl
GETVARTYP	F	Leitura do tipo de dados em uma variável de sistema/de usuário		+	-	PGAsl
GFRAME0 ... GFRAME100	G	Ativação do frame de afiação <n> da gestão de dados no canal	m	+		PGsl
GOTO	K	Instrução de salto primeiro para frente depois para trás (sentido primeiro para o fim e depois para o início do programa)		+		PGAsl
GOTOB	K	Instrução de salto para trás (sentido no início do programa)		+		PGAsl
GOTOC	K	Como GOTO, mas com supressão do alarme 14080 "Destino de salto não encontrado"		+		PGAsl
GOTOF	K	Instrução de salto para frente (sentido no fim do programa)		+		PGAsl
GOTOS	K	Salto de retorno ao início do programa		+		PGAsl
GP	K	Palavra-chave para programação indireta de atributos de posição		+		PGAsl
GROUP_ ADDEND	C (T)	Final do acréscimo de entrada		+		PGAsl
GROUP_BEGIN	C (T)	Início do bloco do programa		+		PGAsl
GROUP_END	C (T)	Final do bloco do programa		+		PGAsl
GWPSOF	P	De-selecionar a velocidade constante da circunferência do disco (SUG)	s	+	-	PGsl
GWPSON	P	Selecionar a velocidade constante da circunferência do disco (SUG)	s	+	-	PGsl
H...	A	Emissão de função auxiliar no PLC		+	+	PGsl/FB1sl (H2)

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
HOLES1	C (T)	Fileira de furos		+		PGAsl
HOLES2	C (T)	Círculo de furos		+		PGAsl
I	A	Parâmetro de interpolação	s	+		PGsl
I1	A	Coordenada de ponto intermediário	s	+		PGsl
IC	K	Especificação de dimensões incrementais	s	+		PGsl
ICYCOF	P	Execução de todos blocos de um ciclo de tecnologia conforme ICYCOF em um ciclo IPO		+	+	FBSYsl
ICYCON	P	Execução de cada bloco de um ciclo de tecnologia conforme ICYCOF em um ciclo IPO separado		+	+	FBSYsl
ID	K	Identificação para ações síncronas modais	m	-	+	FBSYsl
IDS	K	Identificação para ações síncronas estáticas modais		-	+	FBSYsl
IF	K	Introdução de um salto condicional no programa de peça / ciclo de tecnologia		+	+	PGAsl
INDEX	F	Determinação do índice de um caractere na String de entrada		+	-	PGAsl
INICF	K	Inicialização das variáveis com NEWCONF		+		PGAsl
INIPO	K	Inicialização das variáveis com PowerOn		+		PGAsl
INIRE	K	Inicialização das variáveis com Reset		+		PGAsl
INIT	P	Seleção de um determinado programa NC para execução em um determinado canal		+	-	PGAsl
INITIAL		Criação de um INI-File através de todas as áreas		+		PGAsl
INT	K	Tipo de dado: Valor inteiro com sinal		+		PGAsl
INTERSEC	F	Calcula a intersecção entre dois elementos de contorno		+	-	PGAsl
INVCCW	G	Deslocamento de evolvente, no sentido anti-horário	m	+		PGsl
INVCW	G	Deslocamento de evolvente, no sentido horário	m	+		PGsl
INVFRAME	F	Cálculo do Frame inverso a partir de um Frame		+	-	FB1sl (K2)
IP	K	Parâmetro de interpolação variável		+		PGAsl
IPOBRKA	P	Critério de movimento a partir do ponto de ativação da rampa de frenagem	m	+	+	
IPOENDA	K	Fim de movimento ao alcançar "IPO-Stop"	m	+		PGAsl
IPTRLOCK	P	Congela o início do segmento do programa que não deve ser pesquisado até o próximo bloco de função da máquina.	m	+	-	PGAsl
IPTRUNLOCK	P	Define o fim do segmento do programa que não deve ser pesquisado no bloco atual no momento da interrupção.	m	+	-	PGAsl
IR	A	Coordenada de ponto intermediário do círculo (sentido X) ao tornear roscas esféricas		+		PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
ISAXIS	F	Verifica se o eixo geométrico especificado como parâmetro é 1		+	-	PGAsl
ISD	A	Profundidade de imersão	m	+		PGAsl
ISFILE	F	Verificar, se há um arquivo na memória de aplicativos NC		+	-	PGAsl
ISNUMBER	F	Verifica se a String de entrada pode ser convertida em número		+	-	PGAsl
ISOCALL	K	Chamada indireta de um programa programado em linguagem ISO		+		PGAsl
ISVAR	F	Verifica se o parâmetro de transferência contém uma variável conhecida do NC		+	-	PGAsl
J	A	Parâmetro de interpolação	s	+		PGsl
J1	A	Coordenada de ponto intermediário	s	+		PGsl
JERKA	P	Ativação do comportamento de aceleração ajustado através de MD para os eixos programados		+	-	
JERKLIM	K	Redução ou aceleração do solavanco axial máximo	m	+		PGAsl
JERKLIMA	K	Redução ou aceleração do solavanco axial máximo	m	+	+	PGAsl
JR	A	Coordenada de ponto intermediário do círculo (sentido Y) ao tornear roscas esféricas		+		PGsl
K	A	Parâmetro de interpolação	s	+		PGsl
K1	A	Coordenada de ponto intermediário	s	+		PGsl
KONT	G	Percorre o contorno com compensação da ferramenta	m	+		PGsl
KONTC	G	Aproximação/afastamento com polinômio de curvatura contínua	m	+		PGsl
KONTT	G	Aproximação/afastamento com polinômio de tangente constante	m	+		PGsl
KR	A	Coordenada de ponto intermediário do círculo (sentido Z) ao tornear roscas esféricas		+		PGsl
L	A	Número da subrotina	s	+	+	PGAsl
LEAD	A	Ângulo de avanço 1. Orientação da ferramenta 2. Polinômios de orientação	m	+		PGAsl
LEADOF	P	Acoplamento axial de valor mestre OFF		+	+	PGAsl
LEADON	P	Acoplamento axial de valor mestre ON		+	+	PGAsl
LENTOAX	F	Fornece informações sobre a associação dos comprimentos de ferramenta L1, L2 e L3 da ferramenta com a abscissa, ordenada e aplicada		+	-	PGAsl
LFOF [6]	G	Retrocesso rápido para rosqueamento OFF	m	+		PGsl
LFON	G	Retrocesso rápido para rosqueamento ON	m	+		PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
LFPOS	G	Retrocesso do eixo identificado com POLFMASK ou POLFMLIN na posição de eixo absoluta programada com POLF	m	+		PGsl
LFTXT [6]	G	O plano do movimento de retrocesso na retração rápida é determinado a partir da tangente da trajetória e do atual sentido de ferramenta	m	+		PGsl
LFWP	G	O plano do movimento de retrocesso na retração rápida é determinado através do atual plano de trabalho (G17/G18/G19)	m	+		PGsl
LIFTFAST	K	Retração rápida		+		PGsl
LIMS	K	Limite de rotação com G96/G961 e G97	m	+		PGsl
LLI	K	Valor limite inferior de variáveis		+		PGAsl
LN	F	Logaritmo natural		+	+	PGAsl
LOCK	P	Bloqueio de ação síncrona com ID (parar ciclo de tecnologia)		-	+	FBSYsl
LONGHOLE	C (T)	Oblongo		+		PGAsl
LOOP	K	Introdução de um loop sem fim		+		PGAsl

Instruções M ... R

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
M0		Parada programada		+	+	PGsl
M1		Parada opcional		+	+	PGsl
M2		Fim do programa Programa principal (como M30)		+	+	PGsl
M3		Sentido de giro do fuso à direita (horário)		+	+	PGsl
M4		Sentido de giro do fuso à esquerda (anti-horário)		+	+	PGsl
M5		Parada do fuso		+	+	PGsl
M6		Troca de ferramentas		+	+	PGsl
M17		Fim da subrotina		+	+	PGsl
M19		Posicionamento de fuso na posição registrada no SD43240		+	+	PGsl
M30		Fim do programa Programa principal (como M2)		+	+	PGsl
M40		Mudança automática da gama de velocidade		+	+	PGsl
M41 ... M45		Gama de velocidade 1 ... 5		+	+	PGsl
M70		Passagem para o modo de eixo		+	+	PGsl
MASLDEF	P	Definição de grupo de eixos mestres/escravos		+	+	PGAsl
MASLDEL	P	Separação de grupo de eixos mestres/escravos e cancelamento da definição do grupo		+	+	PGAsl
MASLOF	P	Desativação de um acoplamento temporário		+	+	PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
MASLOFS	P	Desativação de um acoplamento temporário com parada automática do eixo escravo		+	+	PGAsl
MASLON	P	Ativação de um acoplamento temporário		+	+	PGAsl
MATCH	F	Localização de uma String em Strings		+	-	PGAsl
MAXVAL	F	Maior valor de duas variáveis (função. trigonométrica)		+	+	PGAsl
MCALL	K	Chamada de subrotina modal		+		PGAsl
MEAC	K	Medição contínua por eixo sem anulação do curso restante	s	+	+	PGAsl
MEAFRAME	F	Cálculo de Frame a partir de pontos de medição		+	-	PGAsl
MEAS	A	Medição com anulação de curso restante	s	+		PGAsl
MEASA	K	Medição por eixo com anulação do curso restante	s	+	+	PGAsl
MEASURE	F	Método de cálculo para medição de peça e medição de ferramenta		+	-	FB1sl (M5)
MEAW	A	Medição sem anulação de curso restante	s	+		PGAsl
MEAWA	K	Medição por eixo sem anulação do curso restante	s	+	+	PGAsl
MI	K	Acesso aos dados de Frame: Espelhamento		+		PGAsl
MINDEX	F	Determinação do índice de um caractere na String de entrada		+	-	PGAsl
MINVAL	F	Menor valor de duas variáveis (função. trigonométrica)		+	+	PGAsl
MIRROR	G	Espelhamento programável	s	+		PGAsl
MMC	P	Chamada da janela de diálogo interativa na HMI a partir do programa de peça		+	-	PGAsl
MOD	K	Divisão Modulo		+		PGAsl
MODAXVAL	F	Determinação da posição Modulo de um eixo rotativo Modulo		+	-	PGAsl
MOV	K	Partida de eixo de posicionamento		-	+	FBSYsl
MOVT	A	indicar o ponto final de um deslocamento no sentido da ferramenta				FB1(K2)
MSG	P	Mensagens programáveis	m	+	-	PGsl
MVTOOL	P	Comando de linguagem para movimentar uma ferramenta		+	-	FBWsl
N	A	Número de bloco secundário NC		+		PGsl
NAMETOINT	F	Índice de variável do sistema		+		PGAsl
NC	K	Especificação da área de validade de dados		+		PGAsl
NEWCONF	P	Aceitação dos dados de máquina modificados (corresponde à "Ativação do dado de máquina")		+	-	PGAsl
NEWMT	F	criar novo Multitool		+	-	FBWsl
NEWT	F	Criação de nova ferramenta		+	-	FBWsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
NORM [6]	G	Ajuste normal dos pontos inicial e final com compensação de ferramenta	m	+		PGsl
NOT	K	NÃO lógico (Negation)		+		PGAsl
NPROT	P	Área de proteção específica de máquina ON/ OFF		+	-	PGAsl
NPROTDEF	P	Definição de uma área de proteção específica de máquina		+	-	PGAsl
NUMBER	F	Converte a String de entrada em número		+	-	PGAsl
OEMIPO1	G	Interpolação OEM 1	m	+		PGAsl
OEMIPO2	G	Interpolação OEM 2	m	+		PGAsl
OF	K	Palavra-chave na bifurcação CASE		+		PGAsl
OFFN	A	Sobremetal para contorno programado	m	+		PGsl
OMA1	A	Endereço OEM 1	m	+		PGAsl
OMA2	A	Endereço OEM 2	m	+		PGAsl
OMA3	A	Endereço OEM 3	m	+		PGAsl
OMA4	A	Endereço OEM 4	m	+		PGAsl
OMA5	A	Endereço OEM 5	m	+		PGAsl
OR	K	Operador lógico, operador lógico OU		+		PGAsl
ORIAXES	G	Interpolação linear dos eixos de máquina ou eixos de orientação	m	+		PGAsl
ORIAXPOS	G	Ângulo de orientação através de eixos virtuais de orientação com posições de eixo rotativo	m	+		PGAsl
ORIC [6]	G	As mudanças de orientação em cantos externos são sobrepostas no bloco circular a ser inserido	m	+		PGAsl
ORICONCCW	G	Interpolação em uma superfície periférica circular em sentido anti-horário	m	+		PGAsl/FB3sl (F3)
ORICONCW	G	Interpolação em uma superfície periférica circular em sentido horário	m	+		PGAsl/FB3sl (F4)
ORICONIO	G	Interpolação em uma superfície periférica circular com especificação de uma orientação intermediária	m	+		PGAsl/FB3sl (F4)
ORICONTO	G	Interpolação em uma superfície periférica circular na transição tangencia (especificação da orientação final)	m	+		PGAsl/FB3sl (F5)
ORICURVE	G	Interpolação da orientação com especificação do movimento de dois pontos de contato da ferramenta	m	+		PGAsl/FB3sl (F6)
ORID	G	As mudanças de orientação são executadas antes de um bloco circular	m	+		PGAsl
ORIEULER [6]	G	Ângulo de orientação através de ângulo euleriano	m	+		PGAsl
ORIMKS	G	Orientação de ferramenta no sistema de coordenadas da máquina	m	+		PGAsl
ORIPATH	G	Orientação da ferramenta relativa à trajetória	m	+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
ORIPATHS	G	Orientação de ferramenta relativa à trajetória, uma dobra é suavizada no decurso da orientação	m	+		PGAsl
ORIPLANE	G	Interpolação num plano (corresponde ao ORIVECT) Interpolação de grande circunferência	m	+		PGAsl
ORIRESET	P	Posição inicial da orientação de ferramenta com até 3 eixos de orientação		+	-	PGAsl
ORIROTA [6]	G	Ângulo de rotação de um sentido de rotação especificado como absoluto	m	+		PGAsl
ORIROTC	G	Vetor de rotação tangencial à tangente da trajetória	m	+		PGAsl
ORIROTR	G	Ângulo de rotação relativo ao plano entre a orientação inicial e a orientação final	m	+		PGAsl
ORIROTT	G	Ângulo de rotação relativo à alteração do vetor de orientação	m	+		PGAsl
ORIRPY	G	Ângulo de orientação através de ângulo RPY (XYZ)	m	+		PGAsl
ORIRPY2	G	Ângulo de orientação através de ângulo RPY (ZYX)	m	+		PGAsl
ORIS	A	Alteração de orientação	m	+		PGAsl
ORISOF [6]	G	Suavização do decurso de orientação OFF	m	+		PGAsl
ORISOLH	F	Calcular orientação		+		PGAsl
ORISON	G	Suavização do decurso de orientação ON	m	+		PGAsl
ORIVECT [6]	G	Interpolação de grande circunferência (idêntico ao ORIPLANE)	m	+		PGAsl
ORIVIRT1	G	Ângulo de orientação através de eixos virtuais de orientação (Definition 1)	m	+		PGAsl
ORIVIRT2	G	Ângulo de orientação através de eixos virtuais de orientação (Definition 1)	m	+		PGAsl
ORIWKS [6]	G	Orientação de ferramenta no sistema de coordenadas da peça de trabalho	m	+		PGAsl
OS	K	Oscilação ativada/desativada		+		PGAsl
OSB	K	Oscilação: Ponto de partida	m	+		FB1sl (P5)
OSC	G	Suavização constante da orientação da ferramenta	m	+		PGAsl
OSCILL	K	Axis: 1 - 3 eixos de penetração	m	+		PGAsl
OSCTRL	K	Opções da oscilação	m	+		PGAsl
OSD	G	Suavização da orientação de ferramenta através da especificação da extensão de suavização com SD	m	+		PGAsl
OSE	K	Ponto final da oscilação	m	+		PGAsl
OSNSC	K	Oscilação: Número de passadas finais	m	+		PGAsl
OSOF [6]	G	Suavização da orientação de ferramenta OFF	m	+		PGAsl
OSP1	K	Oscilação: ponto de reversão esquerdo	m	+		PGAsl
OSP2	K	Ponto de reversão direito da oscilação	m	+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
OSS	G	Suavização da orientação da ferramenta no fim do bloco	m	+		PGAsl
OSSE	G	Suavização da orientação de ferramenta no início e no fim do bloco	m	+		PGAsl
OST	G	Suavização da orientação de ferramenta através da especificação da tolerância angular em graus com SD (desvio máximo do decurso de. orientação programado)	m	+		PGAsl
OST1	K	Oscilação: Ponto de parada no ponto de reversão esquerdo	m	+		PGAsl
OST2	K	Oscilação: Ponto de parada no ponto de reversão direito	m	+		PGAsl
OTOL	A	Tolerância de orientação para funções de compressor, suavização de orientação e tipos de suavização	M	+		PGAsl
OVR	K	Correção de rotação	m	+		PGAsl
OVRA	K	Correção de rotação axial	m	+	+	PGAsl
OVRRAP	K	Correção do avanço rápido	m	+		PGAsl
P	A	Quantidade de passagens de subrotinas		+		PGAsl
PAROT	G	Alinhamento do sistema de coordenadas à peça de trabalho	m	+		PGsl
PAROTOF [6]	G	Desativação da rotação de Frame relativa à peça de trabalho	m	+		PGsl
PCALL	K	Chamada de subrotinas com indicação absoluta do caminho e transferência de parâmetros		+		PGAsl
PDELAYOF	G	Retardamento na estampagem OFF	m	+		PGAsl
PDELAYON [6]	G	Retardamento na estampagem ON	m	+		PGAsl
PHI	K	Ângulo de rotação da orientação em torno do eixo de sentido do cone		+		PGAsl
PHU	K	Unidade física de uma variável		+		PGAsl
PL	A	1. B-Spline: Distância de nós 2. Interpolação de polinômios: Comprimento do intervalo de parâmetros na interpolação de polinômios	s	+		PGAsl
PM	K	por minuto		+		PGsl
PO	K	Coeficiente de polinômio na interpolação de polinômios	s	+		PGAsl
POCKET3	C (T)	Fresamento de bolsão retangular		+		PGAsl
POCKET4	C (T)	Fresamento de bolsão circular		+		PGAsl
POLF	K	Posição de retrocesso LIFTFAST	m	+		PGsl/PGAsl
POLFA	P	Início da posição de retrocesso dos eixos individuais com $AA_ESR_TRIGGER	m	+	+	PGsl
POLFMASK	P	Liberar eixos para o retrocesso sem haver relação entre eixos	m	+	-	PGsl
POLFMLIN	P	Liberação de eixos para o retrocesso com relação linear entre os eixos	m	+	-	PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
POLY	G	Interpolação de polinômios	m	+		PGAsl
POLYPATH	P	A interpolação de polinômios pode ser selecionada para os grupos de eixos AXIS ou VECT	m	+	-	PGAsl
PON	G	Estampagem ON	m	+		PGAsl
PONS	G	Estampagem ON no ciclo IPO	m	+		PGAsl
POS	K	Posicionamento de eixo		+	+	PGsl
POSA	K	Posicionamento de eixo além dos limites de bloco		+	+	PGsl
POSM	P	Posicionamento do magazine		+	-	FBWsl
POSMT	P	posicionar Multitool no suporte de ferramenta no nº. da posição		+	-	FBWsl
POSP	K	Posicionamento em segmentos (oscilação)		+		PGsl
POSRANGE	F	Determinação se a atual posição nominal interpolada de um eixo encontra-se em uma janela e uma posição de referência pré-definida		+	+	FBSYsl
POT	F	Quadrado (função aritmética)		+	+	PGAsl
PR	K	Por rotação		+		PGsl
PREPRO	PA	Identificação de subrotinas com preparação		+		PGAsl
PRESETON	P	Definir valor nominal com perda do estado de referenciação		+	+	PGAsl
PRESETONS	P	Definir valor nominal sem perda do estado de referenciação		+	+	PGAsl
PRIO	K	Palavra-chave para definir a prioridade no tratamento de interrupções		+		PGAsl
PRLOC	K	Inicialização das variáveis no Reset somente após a alteração local		+		PGAsl
PROC	K	Primeira instrução de um programa		+		PGAsl
PROTA	P	Pedir novo cálculo do modelo de colisão		+		PGAsl
PROTD	F	Calcular distância entre duas áreas de proteção		+		PGAsl
PROTS	P	Definir o estado da área de proteção		+		PGAsl
PSI	K	Ângulo de abertura do cone		+		PGAsl
PTP	G	Movimento ponto a ponto (deslocamento PTP)	m	+		PGAsl
PTPG0	G	Movimento ponto a ponto somente com G0, senão movimento de percurso CP	m	+		PGAsl
PTPWOC	G	Movimento ponto a ponto sem movimentos de compensação,que são causadas por mudanças de orientação	m	+		PGAsl
PUNCHACC	P	Aceleração em função do curso para puncionamento		+	-	PGAsl
PUTFTOC	P	Correção fina de ferramenta para dressagem paralela		+	-	PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
PUTFTOCF	P	Correção fina de ferramenta em função de uma função definida com FCTDEF para dressagem paralela		+	-	PGAsl
PW	A	B-Spline, peso do ponto	s	+		PGAsl
QU	K	Emissão rápida de função (auxiliar) adicional		+		PGsl
R...	A	Parâmetro de cálculo também como identificador de eixo ajustável e com extensão numérica		+		PGAsl
RAC	K	Programação de raio específica de eixo, absoluta e por blocos	s	+		PGsl
RDISABLE	P	Bloqueio de leitura (entrada)		-	+	FBSYsl
READ	P	Lê uma ou várias linhas de um arquivo especificado e carrega estas informações no campo		+	-	PGAsl
REAL	K	Tipo de dado: Variável de vírgula flutuante com sinal (números reais)		+		PGAsl
REDEF	K	Redefinição das variáveis do sistema, variáveis de usuário e comandos de linguagem NC		+		PGAsl
RELEASE	P	Liberação de eixos de máquina para troca de eixos		+	+	PGAsl
REP	K	Palavra-chave para inicialização de todos elementos de um campo com o mesmo valor		+		PGAsl
REPEAT	K	Repetição de um loop de programa		+		PGAsl
REPEATB	K	Repetição de uma linha do programa		+		PGAsl
REPOSA	G	Reaproximação no contorno linear com todos os eixos	s	+		PGAsl
REPOSH	G	Reaproximação até o contorno com semicírculo	s	+		PGAsl
REPOSHA	G	Reaproximação até o contorno com todos os eixos; eixos geométricos em semicírculo	s	+		PGAsl
REPOSL	G	Reaproximação linear no contorno	s	+		PGAsl
REPOSQ	G	Reaproximação no contorno em quadrante	s	+		PGAsl
REPOSQA	G	Reaproximação até o contorno linear com todos os eixos; eixos geométricos em quadrante	s	+		PGAsl
RESETMON	P	Comando de linguagem para ativação de valor nominal		+	-	FBWsl
RET	P	Fim de subrotina		+	+	PGAsl
RETB	P	Fim de subrotina		+	+	PGAsl
RIC	K	Programação em raio relativa por bloco e específica de eixo	s	+		PGsl
RINDEX	F	Determinação do índice de um caractere na String de entrada		+	-	PGAsl
RMB	G	Reaproximação no ponto inicial do bloco	m	+		PGAsl
RMBBL	G	Reaproximação no ponto inicial do bloco	s	+		PGAsl
RME	G	Reaproximação no ponto final do bloco	m	+		PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
RMEBL	G	Reaproximação no ponto final do bloco	s	+		PGAsl
RMI [6]	G	Reaproximação no ponto de interrupção	m	+		PGAsl
RMIBL [6]	G	Reaproximação no ponto de interrupção	s	+		PGAsl
RMN	G	Reaproximação no ponto de percurso mais próximo	m	+		PGAsl
RMNBL	G	Reaproximação no ponto de percurso mais próximo	s	+		PGAsl
RND	A	Arredondamento do canto do contorno	s	+		PGsl
RNDM	A	Arredondamento modal	m	+		PGsl
ROT	G	Rotação programável	s	+		PGsl
ROTS	G	Rotações de Frame programáveis com ângulos espaciais	s	+		PGsl
ROUND	F	Arredondamento das casas decimais		+	+	PGAsl
ROUNDUP	F	Arredondamento para cima de um valor de entrada		+	+	PGAsl
RP	A	Raio polar	m/b	+		PGsl
RPL	A	Rotação no plano	s	+		PGsl
RT	K	Parâmetro para acesso aos dados de Frame: Rotação		+		PGAsl
RTLIOF	G	G0 sem interpolação linear (interpolação de eixos individuais)	m	+		PGsl
RTLION [6]	G	G0 com interpolação linear	m	+		PGsl

Instruções S ... Z

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
S	A	Rotação do fuso (com G4, G96/G961 tem outro significado)	m/b	+	+	PGsl
SAVE	PA	Atributo para salvar informações em chamadas de subrotinas		+		PGAsl
SBLOF	P	Supressão de bloco a bloco		+	-	PGAsl
SBLON	P	Cancelamento da supressão de bloco a bloco		+	-	PGAsl
SC	K	Parâmetro para acesso aos dados de Frame: Escala		+		PGAsl
SCALE	G	Escala programável	S	+		PGsl
SCC	K	Atribuição seletiva de um eixo transversal ao G96/G961/G962. Os identificadores de eixo podem ser de eixo geométrico, de canal ou de máquina.		+		PGsl
SCPARA	K	Programação de bloco de parâmetros servo		+	+	PGAsl
SD	A	Grau de Spline	S	+		PGAsl
SET	K	Palavra-chave para inicialização de todos elementos de um campo com valores listados		+		PGAsl

Instrução	tipo [1]	Significado	W[2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
SETAL	P	Definição de alarme		+	+	PGAsl
SETDNO	F	Atribuição de número D do corte (CE) de uma ferramenta (T)		+	-	PGAsl
SETINT	K	Determinação, qual rotina Interrupt deve ser ativada, se houver uma entrada NC		+		PGAsl
SETM	P	Definição de marcadores em canal próprio		+	+	PGAsl
SETMS	P	Retorna para o fuso mestre definido no dado de máquina		+	-	PGsl
SETMS(n)	P	O fuso n deve valer como fuso mestre		+		PGsl
SETMTH	P	definição de número do suporte da ferramenta mestre		+	-	FBWsl
SETPIECE	P	Consideração do número de peças para todas ferramentas atribuídas ao fuso		+	-	FBWsl
SETTA	P	Definição de uma ferramenta do grupo de desgaste como ativa		+	-	FBWsl
SETTCOR	F	Alteração de componentes de ferramenta sob consideração de todas condições gerais		+	-	PGAsl
SETTIA	P	Definição de uma ferramenta do grupo de desgaste como inativa		+	-	FBWsl
SF	A	Deslocamento do ponto de partida para rosqueamento	m	+		PGsl
SIN	F	Seno (função. trigonométrica)		+	+	PGAsl
SIRELAY	F	Ativação das funções de segurança parametrizadas com SIRELIN, SIRELOUT e SIRELTIME		-	+	FBSIsl
SIRELIN	P	Inicialização de grandezas de entrada do módulo de função		+	-	FBSIsl
SIRELOUT	P	Inicialização de grandezas de saída do módulo de função		+	-	FBSIsl
SIRELTIME	P	Inicialização do Timer do módulo de função		+	-	FBSIsl
SLOT1	C (T)	Ranhura longitudinal		+		PGAsl
SLOT2	C (T)	Ranhura circular		+		PGAsl
SOFT	G	Aceleração de trajetória suave	m	+		PGAsl
SOFTA	P	Ativação da aceleração de eixo brusca para os eixos programados		+	-	PGAsl
SON	G	Puncionamento ON	m	+		PGAsl
SONS	G	Puncionamento ON no ciclo IPO	m	+		PGAsl
SPATH [6]	G	A referência de percurso para eixos FGROUP é o comprimento do arco	m	+		PGAsl
SPCOF	P	Comutação do fuso mestre ou fuso (n) de controle de posição para controle de rotação	m	+	-	PGsl
SPCON	P	Comutação do fuso mestre ou fuso (n) de controle de rotação para controle de posição	m	+	-	PGAsl
SPI	F	Converte o número de fuso em identificador de eixo		+	-	PGAsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
SPIF1 [6]	G	Limite entradas/saídas NC para estampagem/puncionamento Byte 1	m	+		FB2sl (N4)
SPIF2	G	Limite entradas/saídas NC para estampagem/puncionamento Byte 2	m	+		FB2sl (N4)
SPLINEPATH	P	Definição de grupo de Spline		+	-	PGAsl
SPN	A	Número de trechos por bloco	S	+		PGAsl
SPOF [6]	G	Curso OFF, estampagem, puncionamento OFF	m	+		PGAsl
SPOS	K	Posição do fuso	m	+	+	PGsl
SPOSA	K	Posição do fuso além dos limites do bloco	m	+		PGsl
SPP	A	Comprimento de um trecho	m	+		PGAsl
SPRINT	F	Retorna uma string de entrada formatada		+		PGAsl
SQRT	F	Raiz quadrada (função aritmética) (square root)		+	+	PGAsl
SR	A	Curso de retrocesso oscilante para ação síncrona	S	+		PGsl
SRA	K	Curso de retrocesso oscilante na entrada externa axial para ação síncrona	m	+		PGsl
ST	A	Tempo de passada final oscilante para ação síncrona	S	+		PGsl
STA	K	Tempo de passada final oscilante axial para ação síncrona	m	+		PGsl
START	P	Inicialização dos programas selecionados em vários canais, simultaneamente a partir do programa em andamento		+	-	PGAsl
STARTFIFO [6]	G	Execução; paralelo à isso, abastecimento da memória de pré-processamento	m	+		PGAsl
STAT		Posição das articulações	S	+		PGAsl
STOLF	K	Fator de tolerância G0	m	+		PGAsl
STOPFIFO	G	Parada do processamento; abastecimento da memória de pré-processamento até ser detectado o STARTFIFO, memória cheia ou fim de programa	m	+		PGAsl
STOPRE	P	Parada de pré-processamento até todos os bloco preparados serem executados pelo processamento principal		+	-	PGAsl
STOPREOF	P	Cancelamento da parada de pré-processamento		-	+	FBSYsl
STRING	K	Tipo de dado: Sequência de caracteres		+		PGAsl
STRINGIS	F	Verifica o escopo de linguagem NC disponível e especialmente verifica a existência, validade, definição e ativação dos nomes de ciclo NC, variáveis de usuário, macros e nomes de Label pertencentes a este comando.		+	-	PGAsl

Instrução	tipo [1]	Significado	W[2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
STRLEN	F	Determinação do comprimento de uma string		+	-	PGAsl
SUBSTR	F	Determinação do índice de um caractere na string de entrada		+	-	PGAsl
SUPA	G	Supressão do atual deslocamento de ponto zero, inclusive os deslocamentos programados, Frames de sistema, deslocamentos com manivela eletrônica (DRF), deslocamento de ponto zero externo e movimento sobreposto	S	+		PGsl
SVC	K	Velocidade de corte da ferramenta	m	+		PGsl
SYNFCT	P	Avaliação de um polinômio em função de uma condição na ação síncrona de movimentos		-	+	FBSYsl
SYNR	K	A leitura da variável é síncrona, isto é, ocorre no momento da execução		+		PGAsl
SYNRW	K	A leitura e gravação da variável são sincronizadas, isto é, ocorrem no momento da execução		+		PGAsl
SYNW	K	A gravação da variável é sincronizada, isto é, ocorre no momento da execução		+		PGAsl
T	A	Chamada de ferramenta (a troca somente ocorre se estiver definida nos dados de máquina; senão será necessário o comando M6)		+		PGsl
TAN	F	Tangente (função trigonométrica)		+	+	PGAsl
TANG	P	Controle tangencial: Definir acoplamento		+	-	PGAsl
TANGDEL	P	Controle tangencial: Deletar acoplamento		+	-	PGAsl
TANGOF	P	Controle tangencial: Desativação de acoplamento		+	-	PGAsl
TANGON	P	Controle tangencial: Ativar acoplamento		+	-	PGAsl
TCA (828D: _TCA)	P	Seleção de ferramenta / troca de ferramentas independente do estado da ferramenta		+	-	FBWsl
TCARR	A	Solicitação de porta-ferramenta (número "m")		+		PGAsl
TCI	P	Troca a ferramenta do alojamento intermediário para o magazine		+	-	FBWsl
TCOABS [6]	G	Determinação de componentes de comprimento da ferramenta a partir da atual orientação de ferramenta	m	+		PGAsl
TCOFR	G	Determinação de componentes de comprimento da ferramenta a partir da orientação do Frame ativo	m	+		PGAsl
TCOFRX	G	Determinação da orientação de ferramenta de um Frame ativo na seleção de ferramenta, a ferramenta aponta para o sentido X	m	+		PGAsl
TCOFRY	G	Determinação da orientação de ferramenta de um Frame ativo na seleção de ferramenta, a ferramenta aponta para o sentido Y	m	+		PGAsl
TCOFRZ	G	Determinação da orientação de ferramenta de um Frame ativo na seleção de ferramenta, a ferramenta aponta para o sentido Z	m	+		PGAsl

Instrução	tipo [1]	Significado	W[2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
THETA	A	Ângulo de giro	S	+		PGAsl
TILT	A	Ângulo lateral	m	+		PGAsl
TLIFT	P	Controle tangencial: Ligar a geração de ponte		+	-	PGAsl
TML	P	Seleção de ferramenta com nº. de posição do magazine		+	-	FBWsl
TMOF	P	Cancelamento do monitoramento de ferramentas		+	-	PGAsl
TMON	P	Ativação do monitoramento de ferramentas		+	-	PGAsl
TO	K	Identifica o valor final em um loop de contagem FOR		+		PGAsl
TOFF	K	Offset de comprimento de ferramenta no sentido do componente de comprimento da ferramenta, que atua paralelo ao eixo geométrico especificado no índice.	m	+		PGsl
TOFFL	K	Offset de comprimento de ferramenta no sentido do componente de comprimento da ferramenta L1, L2 ou L3	m	+		PGsl
TOFFOF	P	Reset de correção online de comprimentos de ferramentas		+	-	PGAsl
TOFFON	P	Ativar a correção online de comprimentos de ferramentas		+	-	PGAsl
TOFFR	A	Offset do raio da ferramenta	m	+		PGsl
TOFRAME	G	Alinhamento do eixo Z do WCS através da rotação de Frame paralelamente à orientação de ferramenta	m	+		PGsl
TOFRAMEX	G	Alinhamento do eixo X do WCS através da rotação de Frame paralelamente à orientação de ferramenta	m	+		PGsl
TOFRAMEY	G	Alinhamento do eixo Y do WCS através da rotação de Frame paralelamente à orientação de ferramenta	m	+		PGsl
TOFRAMEZ	G	como o TOFRAME	m	+		PGsl
TOLOWER	F	Transformação das letras de uma string em letras minúsculas		+	-	PGAsl
TOOLENV	F	Salvamento dos atuais estados importantes para a avaliação dos dados de ferramenta armazenados na memória		+	-	PGAsl
TOOLGNT	F	definir a quantidade de ferramentas de um grupo de ferramentas		+	-	FBWsl
TOOLGT	F	definir o Número T de uma ferramenta de um grupo de ferramenta		+	-	FBWsl
TOROT	G	Alinhamento do eixo Z do WCS através da rotação de Frame paralelamente à orientação de ferramenta	m	+		PGsl
TOROTOF [6]	G	Rotações de Frame no sentido da ferramenta OFF	m	+		PGsl

Instrução	tipo [1]	Significado	W[2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
TOROTX	G	Alinhamento do eixo X do WCS através da rotação de Frame paralelamente à orientação de ferramenta	m	+		PGsl
TOROTY	G	Alinhamento do eixo Y do WCS através da rotação de Frame paralelamente à orientação de ferramenta	m	+		PGsl
TOROTZ	G	como o TOROT	m	+		PGsl
TOUPPER	F	Transformação das letras de uma string em letras maiúsculas		+	-	PGAsl
TOWBCS	G	Valores de desgaste no sistema de coordenadas básico (BCS)	m	+		PGAsl
TOWKCS	G	Valores de desgaste no sistema de coordenadas do cabeçote da ferramenta para transformação cinemática (difere do MCS pela rotação da ferramenta)	m	+		PGAsl
TOWMCS	G	Valores de desgaste no sistema de coordenadas da máquina (MCS)	m	+		PGAsl
TOWSTD [6]	G	Valor de posição inicial para correções no comprimento da ferramenta	m	+		PGAsl
TOWTCS	G	valores de desgaste no sistema de coordenadas da ferramenta (ponto de referência T do suporte da ferramenta na recepção do suporte da ferramenta)	m	+		PGAsl
TOWWCS	G	Valores de desgaste no sistema de coordenadas da peça de trabalho (WCS)	m	+		PGAsl
TR	K	Componente de deslocamento de uma variável Frame		+		PGAsl
TRAANG	P	Transformação de eixo inclinado		+	-	PGAsl
TRACON	P	Transformação concatenada		+	-	PGAsl
TRACYL	P	Cilindro: Transformação de superfície periférica		+	-	PGAsl
TRAFOOF	P	Desativação das transformações ativas no canal		+	-	PGAsl
TRAFOON	P	Acionar transformação com sequências cinemáticas definidas		+	-	PGAsl
TRAILOF	P	Movimento acoplado assíncrono de eixo OFF		+	+	PGAsl
TRAILON	P	Movimento acoplado assíncrono de eixo ON		+	+	PGAsl
TRANS	G	Deslocamento de ponto zero programável absoluto	S	+		PGsl
TRANSMIT	P	Transformação polar (usinagem de face)		+	-	PGAsl
TRAORI	P	Transformação de 4 e 5 eixos, transformação genérica		+	-	PGAsl
TRUE	K	Constante lógica: verdadeiro		+		PGAsl
TRUNC	F	Corte das casas decimais		+	+	PGAsl
TU		Ângulo do eixo	S	+		PGAsl
TURN	A	Número de voltas para linha helicoidal	S	+		PGsl

Instrução	tipo [1]	Significado	W[2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
ULI	K	Valor limite superior de variáveis		+		PGAsl
UNLOCK	P	Liberação de ação síncrona com ID (continuação do ciclo de tecnologia)		-	+	FBSYsl
UNTIL	K	Condição para finalização de um loop REPEAT		+		PGAsl
UPATH	G	A referência de percurso para eixos FGROUP é o parâmetro de curva	m	+		PGAsl
VAR	K	Palavra-chave: Tipo de transferência de parâmetros		+		PGAsl
VELOLIM	K	Redução da velocidade axial máxima	m	+		PGAsl
VELOLIMA	K	Redução ou aceleração da velocidade axial máxima do eixo escravo	m	+	+	PGAsl
WAITC	P	Espera até o critério de mudança de blocos de acoplamento ser preenchido para os eixos / fusos		+	-	PGAsl
WAITE	P	Espera pelo fim do programa em outro canal.		+	-	PGAsl
WAITENC	P	Espera pelas posições de eixo sincronizadas e restauradas		+	-	PGAsl
WAITM	P	Espera pelo marcador no canal especificado; finaliza o bloco especificado com parada exata.		+	-	PGAsl
WAITMC	P	Aguardando marca especificada. Canal; Parada exata apenas quando a marca não foi atingida nos outros canais.		+	-	PGAsl
WAITP	P	Espera pelo fim do deslocamento do eixo de posicionamento		+	-	PGsl
WAITS	P	Espera para alcançar a posição do fuso		+	-	PGsl
WALCS0 [6]	G	Limite da área de trabalho WCS desativado	m	+	-	PGsl
WALCS1	G	WKS- grupo de limitação do campo de trabalho 1 ativo	m	+	-	PGsl
WALCS2	G	WKS- grupo de limitação do campo de trabalho 2 ativo	m	+	-	PGsl
WALCS3	G	WKS- grupo de limitação do campo de trabalho 3 ativo	m	+	-	PGsl
WALCS4	G	WKS- grupo de limitação do campo de trabalho 4 ativo	m	+	-	PGsl
WALCS5	G	WKS- grupo de limitação do campo de trabalho 5 ativo	m	+	-	PGsl
WALCS6	G	WKS- grupo de limitação do campo de trabalho 6 ativo	m	+	-	PGsl
WALCS7	G	WKS- grupo de limitação do campo de trabalho 7 ativo	m	+	-	PGsl
WALCS8	G	WKS- grupo de limitação do campo de trabalho 8 ativo	m	+	-	PGsl
WALCS9	G	WKS- grupo de limitação do campo de trabalho 9 ativo	m	+	-	PGsl

Instrução	tipo [1]	Significado	W [2]	TP [3]	SA [4]	Para descrição, veja [5]
[1] [2] [3] [4] [5] Explicações ver legenda (Página 902).						
WALCS10	G	WKS- grupo de limitação do campo de trabalho 10 ativo	m	+	-	PGsl
WALIMOF	G	Limite da área de trabalho no BCS OFF	m	+	-	PGsl
WALIMON [6]	G	Limite da área de trabalho no BCS ON	m	+	-	PGsl
WHEN	K	A ação é executada apenas uma vez quando a condição for preenchida.		-	+	FBSYsl
WHENEVER	K	A ação é executada ciclicamente em cada ação interpolar, enquanto a condição for preenchida.		-	+	FBSYsl
WHILE	K	Início do loop de programa WHILE		+		PGAsl
WRITE	P	Gravação de texto no sistema de arquivos. Anexa um bloco no fim do arquivo especificado.		+	-	PGAsl
WRTPR	P	Escrever sequência de caracteres na variável BTSS		+	-	PGsl
X	A	Nome de eixo	m/b	+	+	PGsl
XOR	O	OU lógico exclusivo		+		PGAsl
Y	A	Nome de eixo	m/b	+	+	PGsl
Z	A	Nome de eixo	m/b	+	+	PGsl

Legende

[1] Tipo de instrução:

A Endereço

Identificador, para o qual é atribuído um valor, (z. ex. OVR=10). TEs indica alguns endereços, que são acionados ou desativados sem a atribuição do valor de uma função por ex. CPLON e CPLOF).

C (A) Ciclo AST

Programa NC predefinido para a otimização posterior automática com AST (= otimização servo automática). A adequação a uma situação concreta de otimização ocorre através de parâmetros, que são transmitidos ao ciclo durante a chamada.

C (M) Ciclo de medição

Programa NC predefinido, no qual um determinado processo de medição, como por ex., a determinação do diâmetro interno de uma peça de trabalho cilíndrica, é programado com validade geral. A adequação a uma situação concreta de medição ocorre através de parâmetros, que são transmitidos ao ciclo durante a chamada.

C (T) Ciclo tecnológico

Programa NC predefinido, no qual um determinado processo de usinagem, como por ex. a perfuração de uma rosca ou o fresamento de uma bolsa, é programado com validade geral. A adequação a uma situação concreta de usinagem ocorre através de parâmetros que são transmitidos ao ciclo durante a chamada.

F Função pré-definida (fornece um valor de devolução)

O chamado da função pré-definida pode ficar em termo operante.

G Comando G

Os comandos G são divididos em grupos. Em um bloco somente pode ser escrito um comando G de um grupo. Um comando G pode estar ativo modalmente (até ser cancelada por outro comando do mesmo grupo) ou ele está ativo apenas para o bloco onde ele se encontra, ativo por bloco.

K Palavra-chave

Identificador, que determina a sintaxe de um bloco. Não é atribuído nenhum valor à uma palavra chave e com ela também não é possível ativar/desativar nenhuma função NC.

Exemplos: estruturas de controle (IF, ELSE, ENDIF, WHEN, ...), andamento do programa (GOTOB, GOTO, RET …)

O Operador

Operador para uma operação matemática, comparativa ou lógica

P Procedimento pré-definido (não fornece um valor de devolução)

PA Atributo do programa

os atributos se encontram no final da linha de definição de um subprograma:
`PROC <nome do programa>(...) <Atributo do programa>`

Eles determinam a ação durante o andamento do subprograma.

[2] Efeito da instrução:

m modal

s por bloco

[3] Programabilidade no programa de usinagem:

\+ programável

\- não programável

m programável apenas pelo fabricante da máquina

[4] Programabilidade em ações síncronas:

+	programável
-	não programável
T	programável apenas em ciclos de tecnologia

[5] Referência para o documento que contém a descrição completa da instrução:

PGsl	Manual de Fundamentos de Programação
PGAsl	Manual de Programação Avançada
BNMsl	Manual de programação de ciclos de medição
BHDsl	Manual de Operação para Torneamento
BHFsl	Manual de operação para Fresamento
FB1sl ( )	Manual de funções básicas (com a abreviação alfanumérica da respectiva descrição de funcionamento entre parênteses)
FB2sl ( )	Manual de funções ampliadas (com a abreviação alfanumérica da respectiva descrição de funcionamento entre parênteses)
FB3sl ( )	Manual de funções especiais (com a abreviação alfanumérica da respectiva descrição de funcionamento entre parênteses)
FBSIsl	Manual de funções para Safety Integrated
FBSYsl	Manual de funções para ações síncronas
FBWsl	Manual de funções para gerenciamento de ferramentas

[6] Ajuste padrão no início do programa (versão de comando fornecida de fábrica, se não houver nada diferente programado).

Esquema 21-1 Significado das notas de rodapé nas tabelas de instruções

21.2 Instruções Disponibilidade no SINUMERIK 828D

Indicação

Ciclos

Ciclos foram identificados como "opcional", se depender das seguintes opções licenciadas:

- Funções ampliadas de tecnologia (número de artigo: 6FC5800-0AP58-0YB0)
- Ciclos de medição (número de artigo: 6FC5800-0AP28-0YB0)
- Medição de cinemática (número de artigo: 6FC5800-0AP18-0YB0)
- SINUMERIK retificação Advanced (número de artigo: 6FC5800-0AS35-0YB0)

Não foram identificados, se ciclos contêm apenas subfuncionalidades através da opção "Funções ampliadas de tecnologia" em determinados sistemas.

21.2.1 Variantes de comando, fresamento / torneamento

Instruções A ... C

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
:	●	●	●	●	●	●	●	●
*	●	●	●	●	●	●	●	●
+	●	●	●	●	●	●	●	●
-	●	●	●	●	●	●	●	●
<	●	●	●	●	●	●	●	●
<<	●	●	●	●	●	●	●	●
<=	●	●	●	●	●	●	●	●
=	●	●	●	●	●	●	●	●
>=	●	●	●	●	●	●	●	●
/	●	●	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●	●	●
A	●	●	●	●	●	●	●	●
A2	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-
A4	-	-	-	-	-	-	-	-
A5	-	-	-	-	-	-	-	-
A6	-	-	-	-	-	-	-	-

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
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	●	●	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
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	○	○	○	○	○	○	○	○
BLOCK	●	●	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●	●	●
BNAT	○	○	○	○	○	○	○	○
BOOL	●	●	●	●	●	●	●	●
BOUND	●	●	●	●	●	●	●	●
BRISK	●	●	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○	○	○
BTAN	○	○	○	○	○	○	○	○
C	●	●	●	●	●	●	●	●
C2	-	-	-	-	-	-	Nome do eixo do canal	Nome do eixo do canal
C3	-	-	-	-	-	-	-	-

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
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	●	-	●	-	●	●	-	-
COMPCURV	●	-	●	-	●	●	-	-

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
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	○	○	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○	○	○

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
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	-	-	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-	-	-

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
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	-	-	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-	-	-

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
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	●	●	●	●	●	●	●	●
CYCLE758	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
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	-	-	-	-	-	-	-	-

Instruções D ... F

Instrução	SINUMERIK 828D							
• Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
D	•	•	•	•	•	•	•	•
D0	•	•	•	•	•	•	•	•
DAC	•	•	•	•	•	•	•	•
c.c.	•	•	•	•	•	•	•	•
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	•	•	•	•	•	•	•	•

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
FLIN	●	●	●	●	●	●	●	●
FMA	●	●	●	●	●	●	●	●
FNORM	●	●	●	●	●	●	●	●
FOCOF	○	○	○	○	○	○	○	○
FOCON	○	○	○	○	○	○	○	○
FOR	●	●	●	●	●	●	●	●
FP	●	●	●	●	●	●	●	●
FPO	-	-	-	-	-	-	-	-
FPR	●	●	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●	●	●
FRAME	●	●	●	●	●	●	●	●
FRC	●	●	●	●	●	●	●	●
FRCM	●	●	●	●	●	●	●	●
FROM	●	●	●	●	●	●	●	●
FTOC	●	●	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●	●	●
FXS	●	●	●	●	●	●	●	●
FXST	●	●	●	●	●	●	●	●
FXSW	●	●	●	●	●	●	●	●
FZ	●	●	●	●	●	●	●	●

Instruções G ... L

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
G0	●	●	●	●	●	●	●	●
G1	●	●	●	●	●	●	●	●
G2	●	●	●	●	●	●	●	●
G3	●	●	●	●	●	●	●	●
G4	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
ISNUMBER	●	●	●	●	●	●	●	●
ISOCALL	●	●	●	●	●	●	●	●
ISVAR	●	●	●	●	●	●	●	●
J	●	●	●	●	●	●	●	●
J1	●	●	●	●	●	●	●	●
JERKA	●	●	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●	●	●
JR	●	●	●	●	●	●	●	●
K	●	●	●	●	●	●	●	●
K1	●	●	●	●	●	●	●	●
KONT	●	●	●	●	●	●	●	●
KONTC	●	●	●	●	●	●	●	●
KONTT	●	●	●	●	●	●	●	●
KR	●	●	●	●	●	●	●	●
L	●	●	●	●	●	●	●	●
LEAD Orientação da ferramenta Polinômio de orientação	- -	- -	- -	- -	- -	- -	- -	- -
LEADOF	-	-	-	●	-	-	●	●
LEADON	-	-	-	●	-	-	●	●
LENTOAX	●	●	●	●	●	●	●	●
LFOF	●	●	●	●	●	●	●	●
LFON	●	●	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●	●	●
LFWP	●	●	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●	●	●
LIMS	●	●	●	●	●	●	●	●
LLI	●	●	●	●	●	●	●	●
LN	●	●	●	●	●	●	●	●
LOCK	●	●	●	●	●	●	●	●
LONGHOLE	●	●	●	●	●	●	●	●
LOOP	●	●	●	●	●	●	●	●

Instruções M ... R

Instrução	SINUMERIK 828D							
• Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	•	•	•	•	•	•	•	•

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	-	-	-	-	-	-	-	-

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
ROTS	●	●	●	●	●	●	●	●
ROUND	●	●	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●	●	●
RP	●	●	●	●	●	●	●	●
RPL	●	●	●	●	●	●	●	●
RT	●	●	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●	●	●
RTLION	●	●	●	●	●	●	●	●

Instruções S ... Z

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
S	●	●	●	●	●	●	●	●
SAVE	●	●	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●	●	●
SBLON	●	●	●	●	●	●	●	●
SC	●	●	●	●	●	●	●	●
SCALE	●	●	●	●	●	●	●	●
SCC	●	●	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●	●	●
SD	○	○	○	○	○	○	○	○
SET	●	●	●	●	●	●	●	●
SETAL	●	●	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●	●	●
SETINT	●	●	●	●	●	●	●	●
SETM	-	-	-	-	-	○	-	●
SETMS	●	●	●	●	●	●	●	●
SETMS(n)	●	●	●	●	●	●	●	●
SETMTH	●	●	●	●	●	●	●	●
SETPIECE	●	●	●	●	●	●	●	●
SETTA	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (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	●	●	●	●	●	●	●	●

Instrução	SINUMERIK 828D							
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW fresamento exportação (me42)	SW24x(5) CNC-SW torneamento exportação (te42)	SW26x(3) CNC-SW fresamento exportação (me62)	SW26x(3) CNC-SW torneamento exportação (te62)	SW28x(2) CNC-SW fresamento exportação (me821)	SW28x(1) CNC-SW fresamento Adv. Exportação (me822)	SW28x(2) CNC-SW torneamento exportação (te821)	SW28x(1) CNC-SW torneamento Adv. Exportação (te822)
Y	●	●	●	●	●	●	●	●
Z	●	●	●	●	●	●	●	●

21.2.2 Variantes de comando, retificar

Instruções A ... C

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exportação (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exportação (gse82)
:	●	●	●	●	●	●
*	●	●	●	●	●	●
+	●	●	●	●	●	●
-	●	●	●	●	●	●
<	●	●	●	●	●	●
<<	●	●	●	●	●	●
<=	●	●	●	●	●	●
=	●	●	●	●	●	●
>=	●	●	●	●	●	●
/	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●
A	●	●	●	●	●	●
A2	-	-	-	-	-	-
A3	-	-	-	-	-	-
A4	-	-	-	-	-	-
A5	-	-	-	-	-	-
A6	-	-	-	-	-	-
A7	-	-	-	-	-	-
ABS	●	●	●	●	●	●
AC	●	●	●	●	●	●
ACC	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
AXIS	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●
B	●	●	●	●	●	●
B2	-	-	-	-	-	-
B3	-	-	-	-	-	-
B4	-	-	-	-	-	-
B5	-	-	-	-	-	-
B6	-	-	-	-	-	-
B7	-	-	-	-	-	-
B_AND	●	●	●	●	●	●
B_OR	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○
BLOCK	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●
BNAT	○	○	○	○	○	○
BOOL	●	●	●	●	●	●
BOUND	●	●	●	●	●	●
BRISK	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○
BTAN	○	○	○	○	○	○
C	●	●	●	●	●	●
C2	-	-	Nome do eixo do canal	-	-	-
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
C6	-	-	-	-	-	-
C7	-	-	-	-	-	-
CAC	●	●	●	●	●	●
CACN	●	●	●	●	●	●
CACP	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	○	○	○	○	○	○

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
CYCLE971	-	-	-	-	-	-
CYCLE973	-	-	-	-	-	-
CYCLE974	-	-	-	-	-	-
CYCLE976	-	-	-	-	-	-
CYCLE977	-	-	-	-	-	-
CYCLE978	-	-	-	-	-	-
CYCLE979	-	-	-	-	-	-
CYCLE982	-	-	-	-	-	-
CYCLE994	-	-	-	-	-	-
CYCLE995	-	-	-	-	-	-
CYCLE996	-	-	-	-	-	-
CYCLE997	-	-	-	-	-	-
CYCLE998	-	-	-	-	-	-
CYCLE4071	●	●	●	●	●	●
CYCLE4072	●	●	●	●	●	●
CYCLE4073	●	●	●	●	●	●
CYCLE4074	●	●	●	●	●	●
CYCLE4075	●	●	●	●	●	●
CYCLE4077	●	●	●	●	●	●
CYCLE4078	●	●	●	●	●	●
CYCLE4079	●	●	●	●	●	●

Instruções D ... F

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
D	●	●	●	●	●	●
D0	●	●	●	●	●	●
DAC	●	●	●	●	●	●
DC	●	●	●	●	●	●
DCI	●	●	●	●	●	●
DCM	●	●	●	●	●	●
DCU	●	●	●	●	●	●
DEF	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
• Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
FRCM	•	•	•	•	•	•
FROM	•	•	•	•	•	•
FTOC	•	•	•	•	•	•
FTOCOF	•	•	•	•	•	•
FTOCON	•	•	•	•	•	•
FXS	•	•	•	•	•	•
FXST	•	•	•	•	•	•
FXSW	•	•	•	•	•	•
FZ	•	•	•	•	•	•

Instruções G ... L

Instrução	SINUMERIK 828D					
• Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
G0	•	•	•	•	•	•
G1	•	•	•	•	•	•
G2	•	•	•	•	•	•
G3	•	•	•	•	•	•
G4	•	•	•	•	•	•
G5	•	•	•	•	•	•
G7	•	•	•	•	•	•
G9	•	•	•	•	•	•
G17	•	•	•	•	•	•
G18	•	•	•	•	•	•
G19	•	•	•	•	•	•
G25	•	•	•	•	•	•
G26	•	•	•	•	•	•
G33	•	•	•	•	•	•
G34	•	•	•	•	•	•
G35	•	•	•	•	•	•
G40	•	•	•	•	•	•
G41	•	•	•	•	•	•
G42	•	•	•	•	•	•
G53	•	•	•	•	•	•

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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	-	-	●	-	-	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
GETDNO	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●
GETSELT	●	●	●	●	●	●
GETT	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●
GETVARAP	●	●	●	●	●	●
GETVARDFT	●	●	●	●	●	●
GETVARLIM	●	●	●	●	●	●
GETVARPHU	●	●	●	●	●	●
GETVARTYP	●	●	●	●	●	●
GFRAME0 ... GFRA-ME100	< 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	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (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 Orientação da ferra-menta Polígono de orientação	- -	- -	- -	- -	- -	- -

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
LEADOF	-	-	-	-	-	-
LEADON	-	-	-	-	-	-
LENTOAX	●	●	●	●	●	●
LFOF	●	●	●	●	●	●
LFON	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●
LFWP	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●
LIMS	●	●	●	●	●	●
LLI	●	●	●	●	●	●
LN	●	●	●	●	●	●
LOCK	●	●	●	●	●	●
LONGHOLE	-	-	-	-	-	-
LOOP	●	●	●	●	●	●

Instruções M ... R

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
M0	●	●	●	●	●	●
M1	●	●	●	●	●	●
M2	●	●	●	●	●	●
M3	●	●	●	●	●	●
M4	●	●	●	●	●	●
M5	●	●	●	●	●	●
M6	●	●	●	●	●	●
M17	●	●	●	●	●	●
M19	●	●	●	●	●	●
M30	●	●	●	●	●	●
M40	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●
M70	●	●	●	●	●	●
MASLDEF	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
MASLDEL	-	-	-	-	-	-
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	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
OF	●	●	●	●	●	●
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	○	○	○	○	○	○

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
OSB	○	○	○	○	○	○
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	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
PON	-	-	-	-	-	-
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	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
REPOSH	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●
RET	●	●	●	●	●	●
RETB	●	●	●	●	●	●
RIC	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●
RMB	●	●	●	●	●	●
RME	●	●	●	●	●	●
RMI	●	●	●	●	●	●
RMN	●	●	●	●	●	●
RND	●	●	●	●	●	●
RNDM	●	●	●	●	●	●
ROT	●	●	●	●	●	●
ROTS	●	●	●	●	●	●
ROUND	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●
RP	●	●	●	●	●	●
RPL	●	●	●	●	●	●
RT	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●
RTLION	●	●	●	●	●	●

Instruções S ... Z

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
S	●	●	●	●	●	●
SAVE	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●
SBLON	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
SC	●	●	●	●	●	●
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	-	-	-	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
SPOS	●	●	●	●	●	●
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	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
TCOFR	●	●	●	●	●	●
TCOFRX	●	●	●	●	●	●
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	○	○	○	-	-	-

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
TRACON	○	○	○	-	-	-
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	●	●	●	●	●	●

Instrução	SINUMERIK 828D					
● Padrão ○ Opção - não disponível	SW24x(5) CNC-SW G-Tech exportação (gce42)	SW26x(3) CNC-SW G-Tech exportação (gce62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gce82)	SW24x(5) CNC-SW G-Tech exportação (gse42)	SW26x(3) CNC-SW G-Tech exportação (gse62)	SW28x(1) CNC-SW G-Tech Adv. Exporta-ção (gse82)
WALIMOF	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●
WHEN	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●
WHILE	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●
WRITE	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●
X	●	●	●	●	●	●
XOR	●	●	●	●	●	●
Y	●	●	●	●	●	●
Z	●	●	●	●	●	●

21.3 Atual idioma na HMI

A seguinte tabela contém todos os idiomas disponíveis na interface de operação.

O idioma configurado atualmente pode ser consultado no programa de peça e nas ações síncronas através da seguinte variável de sistema:

$AN_LANGUAGE_ON_HMI = <Wert>

<valor>	Idioma	Abreviação de idioma
1	Alemão (Alemanha)	DEU
2	Francês	FRA
3	Inglês (Reino Unido)	ENG
4	Espanhol	ESP
6	Italiano	ITA
7	Holandês	NLD
8	Chinês (simplificado)	CHS
9	Sueco	SVE
18	Húngaro	HUN
19	Finlandês	FIN
28	Tcheco	CSY
50	Português (Brasil)	PTB
53	Polonês	PLK
55	Dinamarquês	DAN
57	Russo	RUS
68	Eslovaco	SKY
72	Romeno	ROM
80	Chinês (tradicional)	CHT
85	Coreano	KOR
87	Japonês	JPN
89	Turco	TRK

Indicação

Uma atualização do $AN_LANGUAGE_ON_HMI é realizada:

- após a inicialização do sistema.
- após o reset do NC e/ou PLC.
- após a comutação para outro NC dentro do M2N.
- após a comutação de idiomas na HMI.

Apêndice

A

A.1 Lista de abreviações

A	
A	Saída
ADI4	Analog Drive Interface para 4 eixos
AC	Adaptive Control
ALM	Active Line Module
ARM	Motor rotativo asíncrono
AS	Sistema de automação
ASCII	American Standard Code for Information Interchange: Norma americana de códigos para troca de informações
ASIC	Application Specific Integrated Circuit: Circuito de aplicação do usuário
ASUP	Subrotina assíncrona
AUXFU	Função auxiliar: Função auxiliar
AWL	Lista de instruções
AWP	Programa de usuário

B	
BA	Modo de operação
BAG	Grupo de modos de operação
BCD	Binary Coded Decimals: Números decimais codificados em código binário
BERO	Chave de aproximação sem contato
BI	Binector Input
BICO	Conector Bionector
BIN	Arquivos binários: dados binários
BIOS	Basic Input Output System
BCS	Sistema de coordenadas básico
BO	Binector Output
BTSS	Interface de painel de comando

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: Ciclos compilados
CEC	Cross Error Compensation
CI	Connector Input
Cartão CF	Cartão Compact Flash

C	
CNC	Computerized Numerical Control: Comando numérico computadorizado
CO	Connector Output
CoL	Certificado de Licença
COM	Communication
CPA	Compiler Projecting Data: Dados de projeção do compilador
CRT	Cathode Ray Tube: Tubos de raios catódicos
CSB	Central Service Board: Unidade de PLC
CU	Control Unit
CP	Processador de comunicação
CPU	Central Processing Unit: Unidade de processamento central
CR	Carriage Return
CTS	Clear To Send: Mensagem de pronto para enviar em interfaces de dados seriais
CUTCOM	Cutter radius compensation: Correção do raio da ferramenta

D	
DAU	Conversor digital-analógico
DB	Módulo de dados (PLC)
DBB	Módulo de dados - Byte(PLC)
DBD	Módulo de dados - palavra dupla (PLC)
DBW	Módulo de dados - palavra (PLC)
DBX	Módulo de dados - Bit (PLC)
DDE	Dynamic Data Exchange
DDS	Drive Data Set: Grupo de dados de acionamento
DIN	Instituto Alemão de Normas
DIO	Data Input/Output: Exibição da transferência de dados
DIR	Directory: Diretório
DLL	Dynamic Link Library
DO	Drive Object
DPM	Dual Port Memory
DPR	Dual Port RAM
DRAM	Memória dinâmica (sem tampão)
DRF	Differential Resolver Function: Função de resolução diferencial (manivela eletrônica)
DRIVE-CLiQ	Drive Component Link with IQ
DRY	Dry Run: Avanço de teste
DSB	Decoding Single Block: Bloco a bloco de decodificação
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Palavra de dados
DWORD	palavra dupla (atual 32 Bit)

E	
E	Entrada
EES	Execution from External Storage
E/A	Entrada/saída
ENC	Encoder: Gerador de valor real
EFP	simples módulo periférico (PLC–E/A–grupo)
EGB	Módulos e elementos sensíveis a cargas eletrétricas
EMV	Compatibilidade electromagnética
EN	Europäische Norm (norma européia)
ENC	Encoder: Gerador de valor real
EnDat	interface do codificador
EPROM	Erasable Programmable Read Only Memory: Memória de leitura deletável e eletricamente programável apenas memória de leitura
ePS Network Services	Serviços para manutenção a distância de máquinas protegidas da internet
EQN	Descrição do tipo de um valor absoluto do codificador com 2048 sinais Sinus/rotação
ES	Engineering System
ESR	Parada e retrocesso ampliados
ETC	Tecla ETC">"; Expansão da barra de softkeys no mesmo menu

F	
FB	Módulo de função (PLC)
FC	Function Call: Módulo de função (PLC)
FEPROM	Flash–EPROM: Memória de leitura e gravação
FIFO	First In First Out: Memória, que opera sem indicação de endereço e cujos dados podem ser lidos na mesma sequência em que vão sendo armazenados
FIPO	Interpolador fino
FPU	Floating Point Unit: Unidade de ponto flutuante
FRK	Correção do raio da fresa
FST	Feed Stop: Parada de avanço
FUP	Plano de funcionamento (método de programação para PLC)
FW	Firmware

G	
GC	Global Control (PROFIBUS: Broadcast-Telegramm)
GDIR	Memória global do programa de peças
GEO	Geometria, ex. eixo geométrico
GIA	Gear Interpolation Data: dados da transmissão interpolar
GND	Signal Ground
GP	Programa básico (PLC)
GS	Gama de velocidade
GSD	Dados principais do equipamento para descrição de um PROFIBUS escravo

G	
GSDML	Generic Station Description Markup Language: linguagem descritiva baseada em XML para edição de um arquivo GSD
GUD	Global User Data: Dados de usuário globais

H	
HEX	Abreviação para número hexadecimal
HiFu	Função auxiliar
HLA	comando linear hidráulico
HMI	Human Machine Interface: SINUMERIK-interface de operação
HSA	Acionamento do fuso principal
HW	Hardware

I	
IBN	Colocação em funcionamento
IKA	Compensação interpolatória
IM	Módulo de interface: Módulo de interface
IMR	Interface-Modul Receive: Módulo de interface para modo de recepção
IMS	Interface-Modul Send: Módulo de interface para modo de envio
INC	Increment: Incremento
INI	Initializing Data: Dados de inicialização
IPO	Interpolador
ISA	International Standard Architecture
ISO	International Standard Organization

J	
JOG	Jogging: Modo de ajuste

K	
K_V	reforço do circuito de controle
K_P	reforço proporcional
$K_Ü$	Relação de transmissão
KOP	Plano de contatos (método de programação para PLC)

L	
LAI	Logic Machine Axis Image: Panorama dos eixos lógicos da máquina
LAN	Local Area Network
LCD	Liquid-Crystal Display: Display de cristal líquido
LED	Light Emitting Diode: Light emitting diode
LF	Line Feed

L	
LMS	Sistema de medição de posição
LR	Controlador de posição
LSB	Bit menos significativo: Bit de valor baixo
LUD	Local User Data: dados do usuário (local)

M	
MAC	Media Access Control (Controle de acesso de mídia)
MAIN	Main program: Programa principal (OB1, PLC)
MB	Megabyte
MCI	Motion Control Interface
MCIS	Motion–Control–Information–System
MCP	Machine Control Panel: Painel de comando da máquina
MD	dados da máquina
MDA	Manual Data Automatic: Entrada manual
MDS	Motor Data Set: Grupo de dados do motor
MELDW	palavra de aviso
MCS	Sistema de coordenadas da máquina
MM	Módulo de motor
MPF	Main Program File: Programa principal (NC)
MSTT	Painel de comando da máquina

N	
NC	Numerical Control: Comando numérico com processamento de dados, faixa de deslocamento etc.
NCU	Numerical Control Unit: Unidade de hardware do NC
NRK	Denominação do sistema operacional do NC
NST	Sinal de interface
NURBS	Non-Uniform Rational B-Spline
NV	Deslocamento de ponto zero
NX	Numerical Extension: módulo de extensão de eixo

O	
OB	Módulo de organização no PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Painel de operação
OPI	Operation Panel Interface: Interface do painel de comando
OPT	Options: Opcionais
OLP	Optical Link Plug: Conector de bus para fibra ótica
OSI	Open Systems Interconnection: Norma para comunicação do processador

P	
PAA	Fluxograma de processo das saídas
PAE	Fluxograma de processo das entradas
PC	Personal Computer
PCIN	Nome do SW para troca de dados com o comando
PCMCIA	Personal Computer Memory Card International Association: Nominação para o cartão de memória
PCU	PC Unit: PC-Box (unidade de processamento)
PG	Dispositivo de programação
PKE	Identificação de parâmetro: parte de um PKW
PKW	Identificação de parâmetro: Valor (parte parametrial de um PPO)
PLC	Programmable Logic Control: Controle de adaptação (programável)
PN	PROFINET
PNO	PROFIBUS-organização do usuário
PO	POWER ON
POE	Unidade de organização do programa
POS	Posição/posicionar
POSMO A	Positioning Motor Actuator: motor de posicionamento
POSMO CA	Positioning Motor Compact AC: unidade completa de comando com módulo integrado de regulagem e capacidade assim como unidade de posicionamento e memória de programa, alimentação de corrente alternada
POSMO CD	Positioning Motor Compact DC: como CA, porém alimentação de corrente direta
POSMO SI	Positioning Motor Servo Integrated: motor de posicionamento; alimentação de corrente direta
PPO	Objeto dos dados do processo do parâmetro; dados de telegrama cíclicos na transmissão com PROFIBUS–DP e seção "comandos de rotação alteráveis"
PPU	Panel Processing Unit (Hardware central de um comando CNC baseado em Panel z.B. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Databus serial
PRT	Teste do programa
PSW	palavra de controle do programa
PTP	Point to Point: Ponto a ponto
PUD	Program Global User Data: Variáveis de usuário globais de programa
PZD	Dados de processo: Divisão de dados do processo de um PPO

Q	
QFK	compensação de erro quadrático

R	
RAM	Random Access Memory: Memória de leitura e gravação
REF	Função de aproximação do ponto de referência
REPOS	Função de reposicionamento

R	
RISC	Reduced Instruction Set Computer: Tipo de processador de bloco de comandos reduzido e rápido processamento dos comandos
ROV	Rapid Override: Correção de avanço rápido
RP	Parâmetro R, parâmetro de rede, variável de usuário pré-definida
RPA	R-Parameter Active: Área de memória no NC para números de parâmetro R
RPY	Roll Pitch Yaw: Tipo de rotação de um sistema de coordenadas
RTLI	Rapid Traverse Linear Interpolation: interpolação linear no movimento rápido
RTS	Request To Send: Ativa uma parte de envio, sinal de controle de interfaces seriais de dados
RTCP	Real Time Control Protocol

S	
SA	Ação síncrona
SBC	Safe Break Control: controle seguro dos freios
SBL	Single Block: Bloco a bloco
SBR	Subrotina: Subrotina (PLC)
SD	dados de configuração
SDB	Módulo de dados de sistema
SEA	Setting Data Active: Identificação (tipo de arquivo) para dados de ajuste
SERUPRO	Search-Run by Program Test: Bloco de pesquisa através de teste de programa
SFB	Módulo de função do sistema
SFC	System Function Call
SGE	Entrada de segurança
SGA	saída de segurança
SH	Parada segura
SIM	Single in Line Module
SK	Softkey
SKP	Skip: função para desvanecer um bloco do programa de peças
SLM	motor linear síncrono
SM	Motor de passo
SMC	Sensor Module Cabinet Mounted
SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SPF	Sub Program File: Subrotina (NC)
SPS	Comando lógico programável = PLC
SRAM	Memória estática (armazenada)
SRK	Correção do raio de corte
SRM	Motor rotativo síncrono
SSFK	Compensação de erro de passo do fuso
SSI	Serial Synchron Interface: Interface serial síncrona
SSL	Localização de blocos
STW	Palavra de controlo

S	
SUG	Velocidade periférica do rebolo
SW	Software
SYF	System Files: Arquivos de sistema
SYNACT	ação sincronizada: Ação síncrona

T	
TB	Terminal Board (SINAMICS)
TCP	Tool Center Point: Ponta da ferramenta
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active: Identificação para dados de máquina
TIA	Totally Integrated Automation
TM	Terminal Board (SINAMICS)
TO	Tool Offset: Correção de ferramenta
TOA	Tool Offset Active: Identificação (tipo de arquivo) para correções de ferramenta
TRANSMIT	Transform Milling Into Turning: transformação de coordenadas para processo de fresamento em uma máquina rotativa
TTL	Transistor–Transistor–Logik (tipo de interface)
TZ	Ciclo tecnológico

U	
UFR	User Frame: Deslocamento de ponto zero
UP	Subrotina
USB	Universal Serial Bus
USV	fornecimento de energia não interruptível

V	
VDI	Interface interna de comunicação entre o NC e o CLP
VDI	Associação dos engenheiros alemães
VDE	Associação dos técnicos de eletrônica da Alemanha
VI	Voltage Input
VO	Voltage Output
VSA	Acionamento de avanço

W	
WAB	Função de aproximação e afastamento suaves
WCS	Sistema de coordenadas da peça de trabalho
WKZ	Ferramenta
WLK	Correção do comprimento da ferramenta
WOP	Programação orientada para oficinas

W	
WPD	Work Piece Directory: Diretório de peças de trabalho
WRK	Correção do raio da ferramenta
T	Ferramenta
WZK	Correção de ferramenta
WZV	Gerenciamento de ferramentas
WZW	Troca de ferramentas

X	
XML	Extensible Markup Language

Z	
ZOA	Zero Offset Active: Identificador para deslocamentos do ponto zero
ZSW	Palavra de condição do comando

A.2 Vista Geral da documentação

Documentação geral

Material promocional
– SINUMERIK 840D sl
– SINUMERIK 828D
– SINUMERIK 828D BASIC

Catálogo NC 62
SINUMERIK 840D sl

catálogo NC 82
SINUMERIK 828D

Catálogo PM 21
SIMOTION,
SINAMICS S120

Manual do sistema
Ferramenta inteligente
operação da máquina

Manual do
sistema
Ctrl Energy

Documentação do usuário

Manual de operação
– Universal
– Torneamento
– Fresamento
– Retificação

Manual de programação
– Princípios básicos
– Preparação para o trabalho
– Ciclos de medição

Manual de programação
– Torneamento ISO
– fresamento ISO

Manual de
diagnósticos
Alarmes

Manual de
diagnósticos
Alarmes

Documentação do fabricante / de serviço

Manual do aparelho
– NCU
– Componente de operação e interconexão
– ADI4

Manual do aparelho
Manual de colocação em funcionamento
Manual de serviço

Manual de colocação em funcionamento
– CNC: NCK, CLP, Acionamento
– Software de base e software de comando

Guia das listras
– Dados da máquina
– Sinais de interface
– Variáveis

Guia das listras
– Dados da máquina
– sinais de interface
– parâmetros
– variáveis

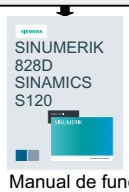
Manual do sistema
Instruções para
Máquinas
- desenvolvimento

Documentação do fabricante / de serviço

Manual de função
– Funções básicas
– Funções avançadas
– Funções específicas
– Ações sincronizadas
– Linguagem ISO

Manual de função
Gerenciamento de ferramentas

Manual de função
Safety Integrated



Manual de função
Safety Integrated

Manual de execução do projeto
– Diretrizes sobre a estrutura EMC
– Industrial Security

Informação / treinamento

Material de treinamento
– Fresamento simples com ShopMill
– Torneamento simples com ShopTurn

Manual
ferramenta e
Construção de moldes

Documentação eletrônica

DOConCD

Industry Online
Suporte (SIOS)

Industry Mall

Glossário

Aceleração com limitação de torque

Para otimizar a resposta de aceleração na máquina, e simultaneamente proteger a mecânica, pode-se alternar no programa de usinagem entre a aceleração rápida e a aceleração constante (sem jerk).

Acionamento

O acionamento é o componente do CNC que controla o torque e a rotação do motor baseado em comandos do NC.

Ações sincronizadas

1. Emissão de função auxiliar
   Durante a usinagem de uma peça pode-se solicitar funções tecnológicas externas (→ Funções auxiliares) do programa de CNC ao CLP. Por exemplo, estas funções auxiliares são utilizadas para controlar equipamentos auxiliares da máquina-ferramenta, como mandril, garras de fixação, porta-ferramenta, etc.
2. Apresentação de funções rápidas de ajuda
   Com relação ao tempo crítico de alteração de funções, o tempo de reconhecimento para as → funções auxiliares pode ser minimizado e paradas desnecessárias no processo de usinagem são evitados.

Alarmes

Todas → Mensagens e alarmes são indicados no painel de operação com data e hora, e o símbolo correspondente para indicar o critério de eliminação. Alarmes e mensagens são mostrados separadamente.

1. Alarmes e mensagens em programas de usinagem
   Alarmes e mensagens podem ser geradas diretamente de programas de usinagem.
2. Alarmes e mensagens do PLC
   Alarmes e mensagens de máquina podem ser geradas pelo programa de PLC. Para isso nenhum pacote adicional de blocos de função é necessário.

Aproximação de ponto fixo

Máquina-ferramenta pode definir pontos fixos para troca de ferramenta, carregamento, troca de paletes, etc. As coordenadas para estes pontos são armazenadas no comando. O comando movimenta os eixos envolvidos, se possível, em → avanço rápido.

Área de offset de ferramenta ativa (TOA)

Na área de offset de ferramenta ativa contém todos os dados de ferramenta e magazine. Por padrão, esta área coincide com a área do → canal considerando o objetivo dos dados. No entanto, os dados de máquina podem ser utilizados para permitir que vários canais compartilhem de uma unidade → TOA, então dados comuns do gerenciador de ferramentas ficam disponíveis para estes canais.

Área de trabalho

Área tri-dimensional, na qual a ponta da ferramenta pode se mover, com base na construção da máquina-ferramenta. Vide → Área de proteção.

Arquivar

Transmissão de arquivos ou diretórios para um dispositivo **externo** de armazenamento.

Aterramento

No terra é conectado todas as partes inativas de um equipamento, o qual mesmo em caso de mal funcionamento não se tornará ativa gerando risco de contato com alguma tensão.

Automático

Modo de operação do comando (Operação em sequência de blocos de acordo com a DIN): Modo de operação do sistema NC, em que um → Programa de usinagem é selecionado e processado de forma contínua.

Avanço de tempo inverso

No lugar da velocidade de avanço pode ser programado para o movimento axial, o tempo necessário do trajeto de um blocol (G93).

Avanço de trajetória

Avanço de trajetória influência → eixos de trajetória. Ele representa a soma geométrica dos avanços dos → eixos geométricos envolvidos.

Avanço rápido

Avanço mais rápido de um eixo. É utilizado quando, por exemplo, a ferramenta está se aproximando de um → contorno da peça de uma posição de descanso ou está sendo recuada. O avanço rápido é definido em uma base de máquina específica através de um dado de máquina.

Bateria reserva

A bateria reserva garante que → o programa de usuário na → CPU será protegido de falhas na alimentação e mantém fixas as áreas de dados e indicadores, temporizadores e contadores.

Bloco

Todas as configurações para as necessidades programação e execução dos programas são realizadas nos blocos.

Bloco de dados

1. A unidade de dados do → PLC, que pode acessar → programas HIGHSTEP.
2. Unidade de dados do → NC: Bloco de dados que contém definições de dados para usuários globais. Os dados podem ser inicializados diretamente em sua configuração.

Bloco de programa

Bloco de programa contém o programa principal e sub-rotinas do → programa de peça.

Bloco principal

antecedido por ":" bloco introdutório, contém todos os parâmetros necessários para iniciar a execução de um → programa de usinagem.

Bloco secundário

Bloco introduzido por "N" com informação sobre a etapa do processo, por exemplo, um dado de posição.

Blocos intermediários

Operação de movimentação com a seleção → de compensação de ferramenta (`G41`/`G42`) pode ser interrompida por uma limitação na quantidade de blocos intermediários (Bloco sem movimentação de eixo no plano de compensação), de forma que a compensação de ferramenta ainda possa ser corretamente realizada. A quantidade de blocos intermediários permitidos, que o comando lê antecipadamente, é ajustável através dos parâmetros de sistema.

Boot

Carrega os programas de sistema ao ligar.

Canal

Um canal é caracterizado pelo fato de que um → programa de usinagem pode ser processado independentemente de outros canais. Um canal controla exclusivamente os eixos e fusos associados à ele. Programas de usinagem trabalham em canais diferentes podem ser coordenados através de → sincronização.

Canal de execução

A estrutura do canal pode ser utilizada para redução de tempo não produtivo através de sequências de movimentos em paralelo, por exemplo, movimento em uma porta de carregamento simultâneo à usinagem. Um canal de CNC deve ser considerado como um comando de CNC separado, com decodificação, preparação de bloco e interpolação.

Chaves

As chaves no → painel de comando da máquina possui quatro posições, as quais possuem funções definidas no sistema de operação do comando. Ao interruptor das chaves são associadas três chaves de cores diferentes, que podem ser movimentadas para as posições específicas.

Chaves de programação

Caracteres ou strings, que possuem um significado fixo na linguagem de programação do → programa de usinagem.

Ciclos

Subrotinas protegidas para suporte na realização de usinagens repetitivas em uma → peça.

Ciclos padrão

Para tarefas de usinagem utilizadas frequentemente são disponíveis os ciclos padrão:

- Para a tecnologia de torneamento/fresamento
- Para a tecnologia de torneamento

Na área de operação "Programa" no menu "Ciclos de auxílio" são listados os ciclos disponíveis. Após a seleção do ciclo desejado são apresentados parâmetros necessários para preenchimento.

CLP

Controle **L**ógico **P**rogramável: → Controlador lógico programável. Componente do → NC: Controlador programável para processar o controle lógico da máquina-ferramenta.

CNC

Vide → NC

Computericed Numerical Control: abrange os componentes → NC, → PLC, HMI, → COM.

CNC

Vide → NC

Computericed Numerical Control: abrange os componentes → NC, → PLC, HMI, → COM.

COM

Componente do comando de NC para realização e coordenação de comunicação.

Compensação de erro no passo do fuso

Compensação de irregularidades mecânicas em um avanço de todas as peças relacionadas ao fuso esférico com base nas variações dos valores medidos.

Compensação de folga

Compensação de folga mecânica da máquina, por exemplo folga no fuso esférico reverso. Para cada eixo a compensação de folga deve ser especificada separadamente.

Compensação do erro de quadrante

Erro de contorno na transição de quadrantes, que aumenta pela alteração no atrito das guias de condução, pode ser virtualmente eliminado com a compensação de erro de quadrante. A parametrização da compensação de erro de quadrante ocorre através de um teste circular.

Compensação do raio da ferramenta

Visando a programação do → contorno de peça desejado, o comando deve percorrer o contorno programado em uma trajetória eqüidistante considerando o raio da ferramenta utilizada, (G41/G42).

Compensação interpolatória

Por meio de compensações interpolatórias como → compensação de falha de elevação de fuso, de flecha, de angularidade e de temperatura, as falhas mecânicas da máquina são compensadas.

Contorno

Forma da → Peça

Contorno acabado

Contorno da peça usinada. Vide → Peça bruta.

Contorno da peça de trabalho

Contorno desejado para → peças a serem criadas/usinadas.

Controle de avanço dinâmico

Irregularidades no contorno devido à erros de contorno podem ser praticamente eliminadas utilizando o controle de avanço dependente da aceleração. Isto resulta em uma excelente

precisão de usinagem mesmo em altas → velocidades. O controle pode selecionar ou retirar a seleção de um eixo específico no → programa de peça.

Controle de velocidade

Visando a obtenção de uma velocidade aceitável em caso de movimentos irrelevantes por bloco, uma análise antecipada em vários blocos pode ser realizada (→ Look Ahead).

Controle Lógico Programável

Controles programáveis (CLP) são controles eletrônicos, no qual suas funções são armazenadas em forma de programa na unidade de controle. A estrutura e a fiação do equipamento não dependem da função do controlador. O controlador programável tem a mesma estrutura que um computador; ele consiste em uma CPU (unidade central) com memória, grupo de entradas e saídas e um bus-system interno. Os periféricos e a linguagem de programação estão alinhados aos interesses do controle.

Coordenadas polares

Sistema de coordenadas, que especifica o local do ponto no plano através de sua distância do ponto zero e o ângulo formado pelo vetor de direção com o eixo fixo.

Corretor do raio de corte

Através da programação de um contorno a ponta da ferramenta é desconsiderada. Na prática isto não é realizado, o raio da ferramenta selecionada deve ser indicado no controle e é considerado desta forma. Desta forma o centro do raio é deslocado de forma eqüidistante do contorno.

Corretores de ferramenta

Consideração das dimensões da ferramenta para o cálculo do trajeto.

CPU

Central Processing Unit, vide → Controle programável

C-Spline

O C-Spline é o Spline mais conhecido e utilizado. O trajeto pelos pontos base são tangentes e com curvatura constante. É utilizado um polinômio de 3. grau.

Curvatura

A curvatura k de um contorno é o inverso do raio r da aproximação em círculo em um ponto de contorno (k = 1/r).

Dados de ajuste

Dados, que comunica propriedades da máquina-ferramenta ao NC, como foi definida pelo Systemsoftware.

Definição de variáveis

Na definição de uma variável contém o tipo de dado e o nome da variável. Com o nome da variável pode-se endereçar o valor desta.

Deslocamento de ponto zero

Padronizar um novo ponto de referência para um sistema de coordenadas através da aquisição de um ponto zero existente e um → Frame.

1. Ajuste
   Existe uma certa quantidade de deslocamentos de zero ajustáveis para cada eixo CNC à disposição. Os deslocamentos selecionados através de comandos G são ativados opcionalmente.
2. Externo
   Adicionalmente para todos deslocamentos, que determinam a localização do ponto zero da peça, um deslocamento de zero externo pode ser sobreposto através da manivela eletrônica (deslocamento DRF) ou através do PLC.
3. Programável
   Com a instrução `TRANS` pode-se programar o deslocamento de ponto zero para todos os eixos de posicionamento e trajetória.

Deslocamento externo de ponto zero

Deslocamento de ponto zero especificado pelo → PLC.

Diagnóstico

1. Área de operação do controle
2. O controle possui tanto um auto-diagnóstico quanto um teste auxiliar para o trabalho: Indicações de estado, alarme e trabalho

Dimensão absoluta

O destino para movimento de um eixo é definido por cotas que se referem ao sistema de coordenadas atualmente ativo. Vide → Sequência de medição.

Dimensão incremental

Também medidas incrementais: O destino de um eixo transversal é definido através de uma distância e direção orientadas a partir de um ponto já alcançado. Vide → medição absoluta.

DRF

Differential Resolver Function: Funções de NC, que gera um deslocamento de ponto zero incremental em modo automático com utilização de uma manivela eletrônica.

Editor

O editor permite a criação, alteração, complementação, junção e inserção de programas/ textos/blocos.

Editor de texto

Vide → Editor

Eixo base

Eixos, para os quais o valor de referência ou o valor atual de posição gera uma base de cálculo para um valor de compensação.

Eixo C

Eixo, ao redor do qual a ferramenta do fuso descreve uma rotação e movimentos de posicionamento controlados.

Eixo de compensação

Eixo, o qual o valor atual ou desejado é alterado de acordo com o valor de compensação.

Eixo de posicionamento

Eixo, que realiza um movimento auxiliar na máquina-ferramenta. (por exemplo, Magazine de ferramentas, transporte de paletes). Eixos de posicionamento são eixos, que não interpolam com → os eixos de trajetória.

Eixo de sincronismo

O eixo de sincronismo é o → eixo gantry, cujas posições desejadas derivam continuamente dos movimentos do → eixo mestre, e move-se de forma sincronizada à este. À vista do operador e do programador, o eixo de sincronismo "não é presente".

Eixo de trajetória

Eixos de trajetória são todos os eixos processados pelo → canal que são controlados pelo → interpolador de forma que iniciam, aceleram, param e atingem o ponto final simultaneamente.

Eixo geométrico

Os eixos geométricos criam o sistema de coordenadas da peça de trabalho de 2 ou 3 dimensões → no qual o → programas de peças programa a geometria da peça de trabalho.

Eixo linear

O eixo linear descreve uma linha reta diferente do eixo rotativo.

Eixo mestre

O eixo mestre é o → eixo gantry, que existe do ponto de vista do operador e do programador e por isso é manipulado como um eixo de NC padrão.

Eixo rotativo

Eixos rotativos produzem um giro da peça ou da ferramenta de acordo com um ângulo especificado.

Eixos

De acordo com suas funções, os eixos de CNC são classificados como:

- Eixos: eixos de interpolação de trajetória
- Eixos auxiliares: Eixos de posicionamento sem interpolação com avanço programado individualmente. Eixos auxiliares não participam de usinagem, Ex. Trocador de ferramentas, Magazine de ferramentas.

Eixos de curvatura

Eixos curvatura produzem um giro da peça ou da ferramenta de acordo com um passo definido. Ao atingir o passo definido, o eixo está "posicionado".

Eixos de máquina

Eixos físicos existentes em uma máquina-ferramenta.

Eixos sincronizados

Eixos síncronos necessitam para seu trajeto o mesmo tempo que um eixo geométrico necessita.

Endereço

Um endereço é o identificador para um certo operando ou faixa de operandos, Ex entrada, saída etc.

Endereço de eixo

Ver → nome do eixo

Escala

Componentes de um → frame, que causa variações de escala.

Especificação de dimensão métrica e em polegadas

No programa de usinagem podem ser programados valores de posições ou passos em polegadas. Independentemente da indicação de medição do programa (G70/G71) o controle é ajustado em um sistema base.

Espelhamento

Através do espelhamento os valores das coordenadas de um contorno são alteradas de forma oposta com relação à um eixo. Pode ser espelhado em vários eixos ao mesmo tempo.

Faixa de deslocamento

A faixa máxima de deslocamento permitida para um eixo linear é de ± 9 décadas. O valor absoluto depende da entrada selecionada, da resolução de controle de posição e do sistema de medição (polegadas ou metro).

Ferramenta

Peça ativa na máquina-ferramenta, que realiza a usinagem (por exemplo, ferramenta de corte, fresa, broca, feixe de laser ...).

Fim de curso de Software

Fim de curso de software limita a faixa de deslocamento de um eixo e previne colisões da mecânica nos limites de hardware. Para cada eixo existe um par, que pode ser ativado separadamente através → do CLP.

Frame

Um frame é representado por uma fórmula aritmética, que transfere um sistema de coordenadas cartesianas para outro sistema de coordenadas cartesianas. Um frame contém as seguintes funções → deslocamento de ponto zero, → rotação, → alteração de escala, → espelhamento.

Frames programáveis

Os → frames programáveis permitem a definição dinâmica de novos pontos de início do sistema de coordenadas durante a execução do programa de usinagem. É utilizado uma definição absoluta com frames novos ou uma definição adicional referente à um ponto de início existente.

Funções auxiliares

Funções auxiliares permite ao programa de usinagem transferir → parâmetros ao → CLP que podem disparar reações definidas pelo fabricante de máquina.

Funções de segurança

O controle contém um monitoramento constante, que detecta falhas no → CNC, no → CLP e na máquina de maneira que é amplamente prevenida alguma danificação das peças, ferramentas ou da máquina. Em caso de falha, a usinagem é interrompida e os acionamentos são parados, a causa do mau funcionamento é armazenada e o alarme é apresentado. Simultanemante, o CLP é informado que um alarme de CNC é apresentado.

Geometria

Descrição de uma → peça em → um sistema de coordenada de peça.

Gerenciamento de programas de usinagem

O gerenciamento de programas de usinagem pode ser organizado por → peça. O tamanho da memória determina o número de programas e dados que poderão ser gerenciados. Cada arquivo (programa ou dado) pode ter um nome com no máximo 24 caracteres alfa numéricos.

Grupo de modos de operação

Eixos e fusos que são tecnologicamente acoplados podem ser combinados em um mesmo grupo de operação (BAG). Eixos/Fusos de um BAG podem ser controlados por um ou vários → canais. O mesmo → modo de operação é sempre atribuído aos canais do BAG.

HIGHSTEP

Sumário para as possibilidades de programação para o → PLC do sistema AS300/AS400.

HW-Konfig

SIMATIC S7-Tool para configuração e parametrização dos componentes de Hardware dentro de um projeto S7.

Identificador

De acordo com a DIN 66025, palavras são complementadas utilizando indicadores (nomes) para variáveis (variáveis de cálculo, variáveis de sistema, variáveis de usuário), para sub-rotinas, palavras-chaves e palavras com várias letras de endereçamentos suplementares. Este complemento tem o mesmo significado das palavras respeitando a construção do bloco. Os identificadores devem ser únicos. O mesmo identificador não pode ser utilizado por diferentes objetos.

Incremento

Indicação de distância do movimento de acordo com o valor do incremento. Valor do incremento pode ser definido pelos → dados de ajuste e/ou selecionado através das teclas 10, 100, 1000, 10000.

Interface de operação

A interface homem-máquina (IHM) é um indicador do comando CNC com auxilio de telas. É composta por softkeys horizontais e verticais.

Interpolação circular

A → ferramenta deve movimentar-se em círculo entre pontos definidos do contorno com um avanço estipulado e então a peça é usinada.

Interpolação de polinômios

Com a interpolação de polinômios os trajetos e curvas mais variados podem ser gerados, como **funções lineares, parábolas, funcões exponenciais** (SINUMERIK 840D).

Interpolação de Spline

Com a interpolação Spline o controle pode gerar uma curva característica bem definida, com apenas alguns pontos base.

Interpolação helicoidal

A interpolação helicoidal é apropriada particularmente à usinagem de rosca interna ou externa com fresa para chanfro e para fresamento de ranhuras de lubrificação.

O movimento helicoidal consiste em dois movimentos em conjunto:

- Movimento circular em um plano
- Movimento linear perpendicular à este plano

Interpolação linear

A ferramenta irá se movimentar por uma linha reta até o destino enquanto usina a peça.

Interpolador

Unidade lógica do → NC que, de acordo com os dados de posição de destino no programa de usinagem, determina os valores intermediários para os movimentos a serem executados nos diversos eixos.

JOG

Modo de operação do controle (Ajuste modo de operação): A máquina pode ser ajustada no modo de operação JOG. Eixos individuais e fusos podem ser movimentados em JOG através das teclas de direção. Outras funções para o modo de operação JOG são: → referenciamento, → Repos e → Preset (ajuste de posição atual).

KV

Fator de ganho do servo, variável de controle em uma malha fechada

Limite de área de trabalho

Com o auxílio da limitação da área de trabalho, o deslocamento dos eixos pode ser limitado além das chaves fim de curso. Cada eixo possui um par de valores para definição da área de trabalho protegida.

Limite de parada exata

Quando todos os eixos atingem o limite de parada exata, o controlador se comporta como se tivesse atingido seu ponto exato de destino. Ocorre um avanço de bloco no → programa de peça.

Limite de Velocidade

Velocidade máxima/mínima do fuso: Através dos dados de máquina, o → PLC ou → os dados de configuração podem limitar a rotação máxima do fuso.

Limite programável da área de trabalho

Limitação da área de movimentação da ferramenta através da programação de limites da área definida.

Linguagem alto-nível do CNC

A linguagem de alto nível serve para escrita de programas NC, → ações sincronizadas e → ciclos. Ela oferece: Estruturas de controle, → variáveis definidas de usuário, → variáveis de sistema, → tecnologia de macros.

Look Ahead

Com a função **Look Ahead** consegue-se otimizar a velocidade de usinagem, através da visualização antecipada de uma certa quantidade de blocos.

MDA

Modo de operação do comando: Manual Data Automatic. No modo de operação MDA, blocos de programa individuais ou sequência de blocos, sem ter referência à um programa ou sub-rotina, podem ser definidos e instantaneamente executados pela tecla NC-Start.

Memória de carregamento

A memória de carregamento é igual à → RAM para a CPU 314 do → CLP.

Memória de compensação

Área de dados do comando, onde são armazenados os dados de corretores de ferramenta.

Memória de programação de CLP

SINUMERIK 840D sl: O programa de usuário, dados de usuários e o programa base de CLP são armazenados juntos na memória de usuário do CLP.

Memória de sistema

A memória de sistema é uma memória da CPU, onde os seguintes dados são arquivados:

- Dados, que são requeridos pelo sistema
- Os operandos de tempo, contador, indicador

Memória de trabalho

A memória de trabalho é uma memória RAM dentro da → CPU, que o processador acessa durante a execução do programa de usuário.

Memória de usuário

Todos os programas e dados como programas de usinagem, sub-rotinas, comentários, correção de ferramenta, deslocamento de ponto zero/frames, assim como, dados de usuário de programa e canal podem ser armazenados na memória comum de usuário do CNC.

Mensagens

Todas as mensagens programadas em um programa de usinagem e → alarmes detectados pelo sistema são indicados no painel de operação com data e hora e com o símbolo correspondente para seu cancelamento. A indicação de alarmes e mensagens são apresentadas separadamente.

Modo contínuo de trajetória

A função do Modo contínuo de trajetória é evitar desacelerações substanciais → dos eixos de trajetória nas fronteira entre blocos do programa de usinagem e continuar com o avanço o mais próximo possível no bloco seguinte.

Modo de operação

Conceito de operação do comando SINUMERIK. Os seguintes modos são definidos: → Jog → MDA → Automático.

Módulo periférico

Módulos I/O realizam a conexão entre CPU e o processo.

Módulos I/O são:

- → Módulos de entrada/saída digital
- → Módulos de entrada/saída analógica
- → Módulos de simulação

Monitoramento de contorno

O erro de contorno é monitorado considerando-se uma faixa de valores de tolerância pré--definidos como precisão do contorno. Um erro de contorno ilegal pode causar, por exemplo, do sobrecarregamento do acionamento. Neste caso apresentará um alarme e o eixo será parado.

NC

Componente de controle numérico do → CNC, que → trabalha o programa de peças e coordena os processos de movimentação da máquina de ferramenta.

Nível de programa

Um programa de peça iniciado no canal é processado como → Programa principal no nível de programa 0 (nível do programa principal). Cada programa de peça chamado no programa principal é processado como → Subrotina em um nível de programa 1 ... n próprio.

Nome de eixo

Para uma identificação clara, todos os eixos de canal e da máquina → do comando com controle de canal ou comando remoto devem ser claramente nomeados. Os → eixos geométricos são nomeados com X, Y, Z. Os eixos geométricos dos eixos rotativos → são nomeados com A, B, C.

NRK

Numeric Robotic Kernel (sistema operacional do → NC)

NURBS

O controle de movimentação e interpolação de trajetória que ocorre no comando é feito com base em NURBS (**N**on **U**niform **R**ational **B**-**S**plines). Como resultado, um movimento uniforme é disponível no comando para todas as interpolações.

OEM

O escopo para implementação de soluções individuais (aplicações OEM) é fornecido pelos fabricantes de máquina que desejem criar sua própria IHM ou incluir funções de processo específicas no comando.

Override

Controle manual ou programável, que permite ao operador alterar avanços programáveis ou rotações, de acordo com a peça ou material.

Override de avanço

A velocidade programada é sobreposta pelo ajuste da velocidade atual feita através → do painel de comando da máquina ou pelo → PLC (0-200%). O avanço de velocidade pode igualmente ser corrigido no programa de usinagem através de uma faixa de porcentagem (1-200%).

Painel de comando da máquina

Painel de operação da máquina-ferramenta com teclas de controle, potenciômetros, etc, e um indicador simples com LEDs. Ele serve para interagir diretamente com a máquina-ferramenta, através do PLC.

Palavra de dados

Unidade de dados de dois bytes dentro de um → bloco de dados.

Palavras-chave

Palavras com sintaxe definida, que tem um significado definido na linguagem de programação para → o programa de peça .

Parada exata

Quando uma parada exata é programada, a posição especificada no bloco é atingida de forma exata e, se necessário, muito lentamente. Para redução do tempo de aproximação são definidos limites de parada exata para avanço rápido ou → avanço.

Parada orientada de fuso

Parada do fuso em uma posição angular pré-determinada, por exemplo, para ser feita uma usinagem auxiliar em uma área específica.

Parâmetros R

Parâmetro de cálculo, que pode ser ajustado ou requisitado → no programa de usinagem para qualquer finalidade.

Peça

Peças a serem fornecidas ou usinadas pela máquina-ferramenta.

Peça bruta

Peça antes de ser usinada.

Ponto de referência

Ponto na máquina-ferramenta, que é referência para o sistema de medição dos → eixos da máquina.

Ponto fixo da máquina

Ponto único definido da máquina-ferramenta, por exemplo, ponto de referência da máquina.

Ponto zero da máquina

Ponto fixo da máquina-ferramenta, no qual permite atribuir à todo sistema de medição (derivado).

Ponto zero da peça

O ponto zero da peça forma o ponto de início para o → sistema de coordenadas da peça. É definido pela distância do → ponto zero da máquina.

Pré-coincidência

Troca de bloco ocorre quando a distância do trajeto aproxima-se de um valor que é igual à um delta especificado com relação à posição final.

Procura de blocos

Para testar um programa de usinagem ou no cancelamento do processo de usinagem, qualquer parte do programa pode ser selecionada utilizando a função "Procura de blocos", da qual o processo de usinagem pode iniciar ou continuar.

Programa de usinagem

Seqüência de instruções do comando NC, que em conjunto resultam na produção de uma → peça específica. E igualmente conduz a usinagem específica para a → peça bruta desejada.

Programa de usuário

Programa de usuário para o sistema de automação S7-300 são criados com a linguagem de programação STEP 7. O programa de usuário possui estrutura modular e consiste de blocos individuais.

Os tipos de básicos de blocos são:

- Blocos de códigos
  Estes blocos contêm as instruções em STEP 7.
- Blocos de dados
  Estes blocos contêm constantes e variáveis para o programa STEP 7.

Programa para transferência de dados PCIN

PCIN é um programa que auxilia no envio e recebimento de dados de usuário do CNC através de uma interface serial, como por exemplo programa de usinagem, corretor de ferramenta etc. O programa PCIN é executável em MS-DOS em computadores industriais padrão.

Programa principal

A denominação "programa principal" é uma herança do tempo em que se dividia os programas de peça em programas principais e → Subrotinas. Esta divisão não existe mais na atual linguagem NC do SINUMERIK. A princípio cada programa de peça pode ser selecionado e executado no canal. Então é processado o → Nível de programa 0 (nível do programa principal). No programa principal podem ser chamados outros programas de peça ou os → Ciclos podem ser chamados como subrotinas

Programação de CLP

O CLP é programado com o software **STEP 7**. O software de programação STEP 7 tem como base o sistema padrão **WINDOWS** e contém as funções de programação de STEP 5 com inovações.

Rede

Uma rede é a conexão de múltiplos S7-300 e outros terminais, por exemplo, uma PG, através → de cabos de conexão. Através da rede ocorre uma troca de dados entre os dispositivos conectados.

Reset geral

Através do reset geral toda a memória da → CPU é apagada:

- → Memória de trabalho
- Área de escrita/leitura da → memória de armazenamento
- → Memória de sistema
- → Memória de back-up

Retração rápida do contorno

Com uma interrupção o programa de usinagem do CNC pode-se introduzir um movimento, que torna possível uma retirada rápida da ferramenta de um contorno da peça que esteja

sendo usinado. Adicionalmente pode-se parametrizar o ângulo e a distância de retração. Após uma retração rápida pode-se adicionalmente executar uma rotina de interrupção

Rosqueamento sem mandril de compensação

Com esta função permite-se fazer rosqueamento sem mandril de compensação. Através de métodos de interpolação do fuso como um eixo rotativo e o eixo de rosqueamento, o rosqueamento é feito com precisão na profundidade final da rosca. Por exemplo, roscas cegas (Condição: Fuso em operação de eixo).

Rotação

Componente de um → frame, que define uma rotação no sistema de coordenadas ao redor de um ângulo específico.

Rotina de interrupção

Rotinas de interrupção são → sub-rotinas especiais, que podem ser iniciadas através da execução de um evento (sinal externo) no processo de usinagem. Um bloco do programa de usinagem é interrompido, e a posição dos eixos são automaticamente armazenadas.

Saídas e entradas digitais rápidas

Em um entrada digital pode-se, por exemplo iniciar uma rotina rápida de CNC (rotina de interrupção). Através de uma saída digital do CNC pode-se rapidamente ativar funções de comutação controladas pelo programa

Sentença do programa de usinagem

Parte de um → programa de peça, demarcado através de Line Feed. Os → blocos principais e os → subblocos são diferenciados.

Sincronização

Instruções em → programas de usinagem para seqüências coordenadas em → canais diferentes em certos pontos de usinagem.

Sistema de coordenadas

Vide → Sistema de coordenadas de máquina, → Sistema de coordenada de peça

Sistema de coordenadas básico

Sistema de coordenadas cartesianas, que é gerado através de uma transformação do sistema de coordenadas da máquina.

O programador utiliza os nomes dos eixos do sistema de coordenadas base nos → programas de peça. O sistema de coordenadas base é paralelo ao → sistema de coordenadas de máquina

caso nenhuma → transformação esteja ativa. A diferença para estes dois sistemas de coordenadas são os → nomes dos eixos.

Sistema de coordenadas da máquina

Sistema de coordenadas, pelo qual os eixos da máquina-ferramenta são orientados.

Sistema de coordenadas da peça

O sistema de coordenadas de peça tem seu ponto de início no → ponto zero da peça. Para programação de usinagem no sistema de coordenadas de peça, a distância e direção da movimentação referem-se à este sistema.

Sistema de medição métrico

Sistema padrão de unidade: para comprimento, por exemplo, mm (milímetro), m (metro).

Sistema de unidade em polegadas

Sistema de unidade, as distâncias em "polegadas" e suas frações.

Softkey

Teclas, as quais são representadas por um campo na tela, e são dinamicamente adaptadas à situação atual de operação. As teclas (Softkeys) que estão disponíveis (livres) são atribuídas funções definidas pelo software.

SRT

Relação de transmissão

Subrotina

A denominação "subrotina" é uma herança do tempo em que se dividia os programas de peça em → Programas principais e subrotinas. Esta divisão não existe mais na atual linguagem NC do SINUMERIK. A princípio cada programa de peça ou cada → Ciclo pode ser chamado como uma subrotina em outro programa de peça. Então o processamento ocorre no próximo → Nível de programa (x+1) (Nível de subrotina (x+1)).

Subrotina assíncrona

Programa de usinagem, que pode ser iniciado de forma assíncrona (independentemente) do programa atual, através de um sinal de interrupção (ex. um sinal "Entrada rápida de NC").

Tabela de compensação

Tabela de pontos de interpolação. É fornecido os valores de compensação dos eixos de compensação para posições selecionadas dos eixos base.

Técnica de macros

Agrupar uma certa quantidade de instruções sob um identificador. O identificador representa instruções agrupadas, em um programa.

Transformação

Deslocamento de ponto zero de um eixo absoluto ou incremental.

Unidade de offset de ferramenta ativa (TOA)

Cada → área de offset de ferramenta ativa pode conter várias unidades desta. A quantidade de TOA disponíveis é limitada de acordo com a quantidade de → canal ativos. Uma unidade TOA contém um bloco de dados de máquina e um bloco de dados de magazine. Adicionalmente também pode conter mais um bloco de dados de máquina de porta ferramenta (opcional).

Usinagem em superfície inclinada

Furação e fresamento na superfície da peça, que não está localizado no plano de coordenadas da máquina, pode ser conduzido confortavelmente com auxilio da função "usinagem em superfície inclinada".

V.24

interface serial para entrada e saída de dados Através desta interface podem ser carregados ou salvos tanto programas de usinagem quanto dados de máquina de fabricante e usuário.

Valor de compensação

Diferença entre a medição da posição do eixo medida, através do encoder, e a programada.

Variáveis de sistema

Uma variável que existe sem precisar da declaração do programador no → programa de usinagem. É definida através de um tipo de dado e de um nome de variável, precedida por **$**. Vide → Variáveis de usuário definidas.

Variáveis definidas pelo usuário

O usuário pode declarar suas próprias variáveis para qualquer propósito em → programas de usinagem ou bloco de dados (Dados de usuário globais). Uma definição contém o tipo de dados e os nomes das variáveis. Vide → Variáveis de sistema.

Velocidade de percurso

A máxima velocidade programável depende da resolução do campo de introdução. Uma resolução de por exemplo 0.1 mm permite um avanço programado de no máximo 1000 m/min.

Velocidade de transmissão

Velocidade para a transferência de dados (Bit/s).

WinSCP

WinSCP é um programa gratuito disponível para Windows, para transferência de arquivos.

Zona de Proteção

Área tri-dimensional dentro da → área de trabalho, na qual não é permitida a passagem da ponta da ferramenta.

Índice

$

$AA_ATOL, 570
$AA_COUP_ACT
  Acoplamento axial de valor-mestre, 600
  Em movimento acoplado, 576
$AA_ESR_ENABLE, 697
$AA_LEAD_SP, 600
$AA_LEAD_SV, 600
$AC_ACT_PROG_NET_TIME, 685
$AC_ACTUAL_PARTS, 688
$AC_AXCTSWA, 670
$AC_AXCTSWE, 670
$AC_CTOL, 570
$AC_CUT_INV, 492
$AC_CUTMOD, 491
$AC_CUTMOD_ANG, 491
$AC_CUTMODK, 491
$AC_CUTTING_TIME, 684
$AC_CYCLE_TIME, 684
$AC_DELAYFST, 550
$AC_ESR_TRIGGER, 697
$AC_OLD_PROG_NET_TIME, 685
$AC_OLD_PROG_NET_TIME_COUNT, 685
$AC_OPERATING_TIME, 684
$AC_OTOL, 570
$AC_PROG_NET_TIME_TRIGGER, 685
$AC_REPOS_PATH_MODE, 560
$AC_REQUIRED_PARTS, 688
$AC_SMAXVELO, 565
$AC_SMAXVELO_INFO, 565
$AC_SPECIAL_PARTS, 688
$AC_TOTAL_PARTS, 688
$AC_TRAFO_CORR_ELEM_P, 422
$AC_TRAFO_CORR_ELEM_T, 422
$AC_TRAFO_ORIAX_LOC, 422
$AN_AXCTAS, 670
$AN_AXCTSWA, 670
$AN_ESR_TRIGGER, 697
$AN_LANGUAGE_ON_HMI, 960
$AN_POWERON_TIME, 684
$AN_SETUP_TIME, 684
$NT_CLOSE_CHAIN_T, 422
$NT_CNTRL, 422
$NT_CORR_ELEM_P, 421
$NT_CORR_ELEM_T, 421
$NT_NAME, 413
$NT_ROT_AX_NAME, 487
$NT_TRAFO_INDEX, 413
$P_ACTBFRAME, 329
$P_AD, 491
$P_BFRAME, 328
$P_CHBFRAME, 328
$P_CHBFRMASK, 329
$P_CTOL, 570
$P_CUT_INV, 492
$P_CUTMOD, 491
$P_CUTMOD_ANG, 491
$P_CUTMOD_ERR, 492
$P_CUTMODK, 491
$P_DELAYFST, 550
$P_IFRAME, 330
$P_IS_EES_PATH, 234
$P_NCBFRAME, 328
$P_NCBFRMASK, 329
$P_ORI_DIFF, 482
$P_ORI_POS, 482
$P_ORI_SOL, 483
$P_ORI_STAT, 486
$P_OTOL, 570
$P_PATH, 233
$P_PFRAME, 330
$P_PROG, 233
$P_PROGPATH, 233
$P_SIM, 295
$P_STACK, 233
$P_SUBPAR, 169
$P_TOOLENV, 501
$P_TOOLENVN, 501
$PA_ATOL, 570
$SA_LEAD_TYPE, 600
$SC_CONTPREC, 543
$SC_MINFEED, 543
$SC_PA_ACTIV_IMMED, 247
$SN_PA_ACTIV_IMMED, 247
$TC_CARR1...14, 466
$TC_CARR18...23, 467
$TC_CARR18[m], 470
$TC_DP1 ... 25, 423
$TC_ECPxy, 427
$TC_SCPxy, 427

*

* (função de cálculo), 79

/

/ (função de cálculo), 79

+

+ (função de cálculo), 79

<

< (operador de comparação), 81
<< (operador de encadeamento), 91
<= (operador de comparação), 81
<> (operador de comparação), 81

=

== (operador de comparação), 81

>

> (operador de comparação), 81
>= (operador de comparação), 81

A

ABS, 79
ACCLIMA, 538
Acoplamento
    genérico, 620
Acoplamento axial de valor mestre, 596
Acoplamento de valor mestre
    Acoplamento de valor real e valor nominal, 599
    Sincronização do eixo mestre e eixo escravo, 598
Acoplamento de valor real, 612
Acoplamento de velocidade, 612
Acoplamento do valor nominal, 612
ACOS, 79
ACTBLOCNO, 181
ACTFRAME, 303
ADISPOSA, 297
Agrupamento de Spline, 271
Alargamento - CYCLE85
    programação externa, 783
Alarmes
    Inserir no programa NC, 694
Alarmes de ciclo, 694
ALF
    par retração rápida do contorno, 139
Alívio - CYCLE940
    programação externa, 827
AND, 81
Ângulo euleriano, 350
Ângulo RPY, 350
Anulação de curso restante, 291
APR, 48
APRB, 48
APRP, 48
APW, 48
APWB, 48
APWP, 48
Áreas de proteção, 239
Arquivo
    Informações, 160
Array, 55
Arredondamento, 162
AS, 219
ASIN, 79
ASPLINE, 260
A-Spline, 266
Atalho de diretório, 230
ATAN2, 79
ATOL, 567
Atributos de posição
    Programação indireta, 73
AV, 609
AX, 661
AXCTSWE, 669
AXCTSWEC, 669
AXCTSWED, 669
AXIS, 31
AXNAME, 90
AXSTRING, 661
AXTOCHAN, 149
AXTOSPI, 661

B

B_AND, 81
B_NOT, 81
B_OR, 81
B_XOR, 81
BAUTO, 260
BFRAME, 303
BLOCK, 206
Bloco a bloco
    - Supressão, 176
Bloco de parada, 552
BLSYNC, 134

BNAT, 260
Bolsão circular - POCKET4
programação externa, 737
Bolsão retangular - POCKET3
programação externa, 735
BOOL, 31
BOUND, 85
BRISK, 536
BRISKA, 536
BSPLINE, 260
B-Spline, 267
BTAN, 260

C

CAC, 259
CACN, 259
CACP, 259
Caixa de transmissão eletrônica, 602
CALL, 205
CALLPATH, 209
Caminho de localização
Caminho de localização programável, 209
na chamada de sub-rotinas, 232
Campo
- Elemento, 56
Definição, 55
Canal - CYCLE930
programação externa, 824
Caractere 0, 89
CASE, 110
case-insensitiv, 227
CDC, 259
Centragem - CYCLE81
programação externa, 773
CFINE, 314
Chamada de contorno - CYCLE62
programação externa, 752
Chamada de sub-rotina com indicação de caminho e parâmetros, 208
CHAN, 31
CHANDATA, 235
CHAR, 31
CHKDNO, 463
CIC, 259
Cinemática
Resolvida, 470
Classe de dados, 53
CLEARM, 127
CLRINT, 136
COARSE, 609
COARSEA, 297
Código G
Programação indireta, 72
Coeficiente de polinômio, 275
COLLPAIR, 408
COMPCAD, 273
COMPCURV, 273
COMPLETE, 235
COMPOF, 273
COMPON, 273
Componente de Frame
FI, 309
MI, 309
RT, 309
SC, 309
TR, 309
COMPSURF, 273
Condições gerais em transformações, 399
CONTDCON, 718
Contorno
- Codificação, 718
Preparação, 712
Reaproximação, 554
Tabela, 712
CONTPRON, 712
controle
Estruturas de, 119
Correção de comprimento de ferramenta
Online, 475
Correção de ferramenta
Memória de correções, 423
Sistema de coordenadas para valores de desgaste, 433
Correção do raio da ferramenta
Desaceleração nos cantos, 296
Correções de ferramentas
aditivo, 426
CORRTRAFO, 415
COS, 79
COUPDEF, 609
COUPDEL, 609
COUPOF, 609
COUPOFS, 609
COUPON, 609
COUPONC, 609
COUPRES, 609
CP, 388
CPBC, 621
CPDEF, 620
CPDEL, 620
CPFMOF, 623
CPFMON, 623
CPFMSON, 622

CPFPOS + CPOF, 623
CPFPOS + CPON, 621
CPFRS, 621
CPLA, 620
CPLCTID, 621
CPLDEF, 620
CPLDEL, 620
CPLDEN, 621
CPLINSC, 625
CPLINTR, 625
CPLNUM, 621
CPLOF, 621
CPLON, 621
CPLOUTSC, 625
CPLOUTTR, 625
CPLPOS, 621
CPLSETVAL, 621
CPMALARM, 626
CPMBRAKE, 626
CPMPRT, 625
CPMRESET, 624
CPMSTART, 625
CPMVDI, 626
CPOF, 621
CPON, 621
CPRECOF, 543
CPRECON, 543
CPROT, 244
CPROTDEF, 239
CPSETTYPE, 626
CPSYNCOP, 625
CPSYNCOP2, 626
CPSYNCOV, 626
CPSYNFIP, 626
CPSYNFIP2, 626
CPSYNFIV, 626
Critério de fim de movimento
programável, 297
CSPLINE, 260
C-Spline, 268
CTAB, 589
CTABDEF, 578
CTABDEL, 585
CTABEND, 578
CTABEXISTS, 585
CTABFNO, 594
CTABFPOL, 594
CTABFSEG, 594
CTABID, 588
CTABINV, 589
CTABISLOCK, 588
CTABLOCK, 586
CTABMEMTYP, 588
CTABMPOL, 594
CTABMSEG, 594
CTABNO, 594
CTABNOMEM, 594
CTABPERIOD, 588
CTABPOL, 594
CTABPOLID, 594
CTABSEG, 594
CTABSEGID, 594
CTABSEV, 589
CTABSSV, 589
CTABTEP, 589
CTABTEV, 589
CTABTMAX, 589
CTABTMIN, 589
CTABTSP, 589
CTABTSV, 589
CTABUNLOCK, 586
CTOL, 567
CTRANS, 314
CUT3DC, 442
CUT3DCC, 452
CUT3DCCD, 452
CUT3DCD, 442
CUT3DF, 446
CUT3DFD, 446
CUT3DFF, 446
CUT3DFS, 446
CUTMOD, 488
CUTMODK, 488
CYCLE4071
programação externa, 839
CYCLE4072
programação externa, 840
CYCLE4073
programação externa, 844
CYCLE4074
programação externa, 845
CYCLE4075
programação externa, 848
CYCLE4077
programação externa, 851
CYCLE4078
programação externa, 854
CYCLE4079
programação externa, 856
CYCLE435 - definir o sistema de coordenadas de ajuste
programação externa, 798
CYCLE495 - Perfilamento
programação externa, 798

CYCLE60 - Gravação
programação externa, 747
CYCLE61- Fresamento de facear
programação externa, 750
CYCLE63 - Fresar bolsão de contorno
programação externa, 753
CYCLE64 -Pré-furação do bolsão de contorno
programação externa, 755
CYCLE70 - Fresamento de roscas
programação externa, 757
CYCLE72 - Fresamento de percurso
programação externa, 758
CYCLE76 - Saliência retangular
programação externa, 763
CYCLE77 - Saliência circular
programação externa, 765
CYCLE78 - Fresar rosca
programação externa, 767
CYCLE79 - Poliedro
programação externa, 771
CYCLE800 - Rotação
programação externa, 800
CYCLE801 - Modelo de posição Grade/Quadro
programação externa, 805
CYCLE802 - Posição definida pelo usuário
programação externa, 806
CYCLE81 - Centragem
programação externa, 773
CYCLE82 - Furação
programação externa, 774
CYCLE83 - Furação profunda
programação externa, 777
CYCLE832 - High Speed Settings
programação externa, 815
CYCLE84 - Rosqueamento com macho sem mandril de compensação
programação externa, 780
CYCLE840 - Rosqueamento com macho com mandril de compensação
programação externa, 818
CYCLE85 - Alargamento
programação externa, 783
CYCLE86 - Mandrilamento
programação externa, 784
CYCLE899 - Fresar calha
programação externa, 821
CYCLE92 - Separação
programação externa, 786
CYCLE930 - Canal
programação externa, 824
CYCLE940 - Alívio
programação externa, 827
CYCLE95 - Desbaste do contorno
programação externa, 787
CYCLE951- Desbaste
programação externa, 829
CYCLE952 - Usinagem de canal de contorno
programação externa, 832
CYCLE98 - encadeamento de roscas
programação externa, 789
CYCLE99 - Torneamento de roscas
programação externa, 793

D

Dados de círculo
calcular, 725
DCI, 53
DCM, 53
DCU, 53
de estruturas de controle
Comportamento do tempo de processamento, 120
de Máquinas
Movimento acoplado, 575
DEF, 31
DEFAULT, 110
DEFINE ... AS, 219
Definição de peça bruta, 704
Definir o sistema de coordenadas de ajuste - CYCLE435
programação externa, 798
DELAYFSTOF, 548
DELAYFSTON, 548
DELDL, 428
DELETE, 156
DELOBJ, 401
DELTOOLENV, 499
Desaceleração de cantos em cantos internos, 296
Desaceleração de cantos em todos os cantos, 296
Desbaste
Funções de apoio, 711
Desbaste - CYCLE951
programação externa, 829
Desbaste do contorno - CYCLE95
programação externa, 787
Deslocamento de ponto a ponto, 388
Deslocamento de ponto zero
Externo, 316
Deslocamento externo de ponto zero, 316
Deslocamento PTP cartesiano, 388
Detectar e localizar áreas que não devem ser pesquisadas, 552
DISABLE, 136

DISPLOF, 181
DISPLON, 181
Disponibilidade
    Dependente do sistema, 5
Disponibilidade dependente de
    Sistema, 5
DISPR, 554
DIV, 79
DL, 426
DO, 631
DRIVE, 536
DRIVEA, 536
DV, 609
DYNFINISH, 540
DYNNORM, 540
DYNPOS, 540
DYNROUGH, 540
DYNSEMIFIN, 540

E

Easy XML, 679
EAUTO, 260
EES, 225
EES-Notation, 227
EG
    Caixa de transmissão eletrônica, 602
EGDEF, 602
EGDEL, 607
EGOFC, 607
EGOFS, 607
EGON, 603
EGONSYN, 603
EGONSYNE, 603
Eixo
    - Troca, 144
Eixo escravo
    Acoplamento axial de valor-mestre, 596
Eixo geométrico
    comutar, 664
Eixo mestre
    Acoplamento axial de valor-mestre, 596
Eixos acoplados, 575
Eixos de orientação, 356
Eixos de rotação
    Ângulo de rotação, 466
    vetores de direção, 466
    vetores de distância, 466
Eixos geométricos comutáveis, 664
Elemento de contorno
    executar, 724
ELSE, 121
ENABLE, 136
ENAT, 260
Encadeamento
    de Strings, 91
Endereçamento, 227
Endereços OEM, 295
ENDFOR, 123
ENDIF, 121
ENDLABEL, 112
ENDLOOP, 122
ENDWHILE, 124
ESR, 696
ESRR, 702
ESRS, 702
Estado do acoplamento
    Acoplamento axial de valor-mestre, 600
    Em movimento acoplado, 576
Estado do pedido de medição, 293
Estampagem, 647
ETAN, 260
Evento de disparo
    Durante a medição, 289
EVERY, 631
EXECSTRING, 76
EXECTAB, 724
EXECUTE, 727
Exibição de bloco
    suprimir, 181
EXP, 79
EXTCALL
    para SINUMERIK 828D, 214
    para SINUMERIK 840D sl, 210
EXTCLOSE, 689
EXTERN, 199
EXTOPEN, 689

F

FALSE, 31
Fator de acoplamento, 573
FCTDEF, 438
FCUB, 531
FENDNORM, 296
Ferramenta
    - Orientação na mudança de Frames, 474
    Orientação, 457
    Parâmetro, 423
FFWOF, 542
FFWON, 542
FIFOCTRL, 545
FILEDATE, 160
FILEINFO, 160

FILESIZE, 160
FILESTAT, 160
FILETIME, 160
Final adendo de entrada - GROUP_ADDEND
programação externa, 859
Final bloco de programa - GROUP_END
programação externa, 859
FINE, 609
FINEA, 297
FLIN, 531
FNORM, 531
FOR, 123
FPO, 531
FRAME, 31
chamar,
-Ligação,
Frames
atribuir, 311
Encadeamento de Frames, 312
Específico de canal, 327
NCU-global, 326
sistema-, 327
Fresamento de bolsão de perfil – CYCLE63
programação externa, 753
Fresamento de facear - CYCLE61
programação externa, 750
Fresamento de furo roscado - CYCLE78
programação externa, 767
Fresamento de percurso - CYCLE72
programação externa, 758
Fresamento de ranhura aberta - CYCLE899
programação externa, 821
Fresamento de roscas - CYCLE70
programação externa, 757
Fresamento de topo, 353
FROM, 631
FTOCOF, 441
FTOCON, 441
Funções OEM, 295
Furação - CYCLE82
programação externa, 774
Furação profunda - CYCLE83
programação externa, 777
Furação profunda 2 - CYCLE830
programação externa, 809
Fuso
- Troca, 144
Fuso sincronizado
Acoplamento, 609
Definição de par, 614

G

G290, 708
G291, 708
G5, 384
G62, 296
G621, 296
G7, 384
G810 ... G819, 295
G820 ... G829, 295
GEOAX, 664
GET, 144
GETACTTD, 465
GETD, 144
GETDNO, 464
GETTCOR, 502
GETTENV, 500
GETVARAP, 65
GETVARDFT, 67
GETVARDIM, 67
GETVARLIM, 66
GETVARPHU, 64
GETVARTYP, 68
GOTO, 107
GOTOB, 107
GOTOC, 107
GOTOF, 107
GOTOS, 106
GP, 73
Gravação - CYCLE60
programação externa, 747
GROUP_ADDEND - final adendo de entrada
programação externa, 859
GROUP_BEGIN - início bloco de programa
programação externa, 858
GROUP_END - final bloco de programa
programação externa, 859
Grupo de movimento acoplado, 573
Grupo G
Tecnologia, 540
GUD, 32

H

High Speed Settings - CYCLE832
programação externa, 815
HOLES1 - Modelo de posição Linha
programação externa, 732
HOLES2 - Modelo de posição Círculo
programação externa, 732

I

ID, 631
IDS, 631
IF, 121
IFRAME, 303
INDEX, 94
Indicação de caminho, 229
Indicador de interrupção automático, 552
Índice de campo, 58
INICF, 31
Inicialização
De campos, 56
Início bloco de programa - GROUP_BEGIN, 858
INIPO, 31
INIRE, 31
INIT, 127
INITIAL, 235
INITIAL_INI, 235
INT, 31
Interpolação de polinômios, 274
Interpolação do vetor de rotação, 366
INTERSEC, 722
IPOBRKA, 297
IPOENDA, 297
IPOSTOP, 609
IPTRLOCK, 551
IPTRUNLOCK, 551
ISAXIS, 661
ISFILE, 159
ISNUMBER, 90
ISOCALL, 207
ISVAR, 62

J

JERKLIM, 563
JERKLIMA, 538

L

L..., 197
Label, 112
LEAD, 347
LEADOF, 596
LEADON, 596
LENTOAX, 521
LIFTFAST, 137
Link
-Variáveis, 29
LLI, 44
LN, 79
LONGHOLE - Oblongo
programação externa, 745
LOOP, 122
Loop contador, 123
Loop de programa
Loop contador, 123
Loop final, 122
Loop IF, 121
Loop REPEAT, 125
Loop WHILE, 124
Loop infinito, 122
LUD, 32

M

M17, 185
M30, 185
Macro, 219
Mandrilamento - CYCLE86
programação externa, 784
Marcador de salto
Nas repetição de partes do programa, 112
no programa salto, 108
MASLDEF, 628
MASLDEL, 628
MASLOF, 628
MASLOFS, 628
MASLON, 628
MATCH, 94
MAXVAL, 85
MCALL, 203
MD10010, 127
MD10280, 127
MD15800, 28
MD18104, 498
MD18116, 499
MD18156, 28
MD20360, 506
MD24558, 508
MD24658, 508
MEAC, 285
MEAFRAME, 322
MEAS, 282
MEASA, 285
MEAW, 282
MEAWA, 285
Memória
de trabalho, 235
Módulo de, 223
Pré-processamento, 545

Memória de correções, 423
Memória de programa
  Diretórios padrão, 224
  Tipos de arquivo, 224
Memória de trabalho, 235
Memória global do programa de peças (GDIR), 226
MINDEX, 94
MINVAL, 85
MMC, 679
MOD, 79
MODAXVAL, 661
Modelo de posição Círculo - HOLES2
  programação externa, 732
Modelo de posição Grade/Quadro - CYCLE801
  programação externa, 805
Modelo de posição Linha - HOLES1
  programação externa, 732
modo de aceleração, 536
Modo de linguagem, 708
Modo de operação
  Durante a medição, 290
Movimento acoplado, 573
Movimento alternado
  Área de reversão, 642
  Penetração no ponto de reversão, 644
  Ponto de reversão, 642
MPF, 224

N

NAMETOINT, 404
NCK, 31
NCK-Notation, 227
NEWCONF, 151
Nível de aninhamento
  de estruturas de controle, 119
NOC, 609
Nome da sub-rotina DIN, 231
Nome da unidade, 229
Nome de arquivo, 231
NOT, 81
NPROT, 244
NPROTDEF, 239
NUMBER, 90
Número D
  atribuir livremente, 463
Número de corte, 463
Números D
  renomear, 464
  verificar, 463
NUT, 358

O

Oblongo - LONGHOLE
  programação externa, 745
OEMIPO1/2, 295
OMA1 ... OMA5, 295
Operadores de comparação, 81
Operadores lógicos, 81
OR, 81
ORIAXES, 356
ORIC, 457
ORICONCCW, 358
ORICONCW, 358
ORICONIO, 358
ORICONTO, 358
ORICURVE, 362
ORID, 457
Orientação da ferramenta
  trajetória relativa, 369
Orientação de ferramenta relativa à trajetória, 369
ORIEULER, 356
ORIMKS, 354
ORIPATH, 370
ORIPATHS, 370
ORIPLANE, 358
ORIRESET(A, B, C), 345
ORIROTA, 366
ORIROTC
  na interpolação da rotação da ferramenta, 372
  na rotação da orientação da ferramenta, 366
ORIROTR, 366
ORIROTT, 366
ORIRPY, 356
ORIRPY2, 356
ORIS, 457
ORISOF, 378
ORISOLH, 478
ORISON, 378
ORIVECT, 356
ORIVIRT1, 356
ORIVIRT2, 356
ORIWKS, 354
OS, 633
OSB, 633
OSC, 457
Oscilação
  Assíncrono(a), 633
  controlar através de ação sincronizada, 639
  Oscilação assíncrona, 633
  Oscilação sincronizada, 639
  Penetração parcial, 642

Oscilação assíncrona, 633
Oscilação sincronizada
Ações sincronizadas, 643
Atribuição do eixo oscilante e eixo de penetração, 642
Avaliação, ciclo de interpolação, 645
Definir penetrações, 642
Movimento de penetração, 643
Penetração na área de reversão, 643
Próxima penetração parcial, 645
OSCILL, 639
OSCTRL, 633
OSD, 457
OSE, 633
OSNSC, 633
OSOF, 457
OSP1, 633
OSP2, 633
OSS, 457
OSSE, 457
OST, 457
OST1, 633
OST2, 633
OTOL, 567

P

P..., 202
P_ACTFRAME, 330
Parada
comando autárquico, 702
controlado pelo NC, 701
Parâmetro
Atuais, 167
Ferramenta, 423
Formal, 167
Transferência na chamada de sub-rotina, 167, 199
Parâmetros de cálculo
Específico de canal, 26
Global, 27
parte do programa
Repetição, 112
PCALL, 208
PDELAYOF, 647
PDELAYON, 647
Peça de trabalho
- Diretório principal, 224
- Diretórios, 224
Contador de, 687
Perfilamento - CYCLE495
programação externa, 798
PFRAME, 303
PHI
na orientação ao longo da superfície do cone, 358
Polinômios de orientação, 364
PHU, 45
PL
Na interpolação de polinômios, 274
Na interpolação de Spline, 260
PO, 274
PO[PHI]
na orientação ao longo da superfície do cone, 358
na rotação da orientação da ferramenta, 370
Polinômios de orientação, 364
PO[PSI]
na orientação ao longo da superfície do cone, 358
na rotação da orientação da ferramenta, 370
Polinômios de orientação, 364
PO[THT]
na rotação da orientação da ferramenta, 370
Polinômios de orientação, 364
PO[XH]
na definição da orientação de dois pontos de contato, 362
Polinômios de orientação, 365
PO[YH]
na definição da orientação de dois pontos de contato, 362
Polinômios de orientação, 365
PO[ZH]
na definição da orientação de dois pontos de contato, 362
Polinômios de orientação, 365
POCKET3 - Bolsão retangular
programação externa, 735
POCKET4 - Bolsão circular
programação externa, 737
POLF
Para retrocesso controlado pelo NC, 697
POLFA, 697
POLFMASK
Para retrocesso controlado pelo NC, 697
POLFMLIN
Para retrocesso controlado pelo NC, 697
Poliedro - CYCLE79
programação externa, 771
Polinômio de denominador, 278
POLY, 274
POLYPATH, 274
PON, 656

PONS, 647
Ponta da fresa, 445
Ponto auxiliar da fresa, 445
Ponto de usinagem da fresa, 445
Porta-ferramenta
- Cinemática, 466
Orientável, 472
Porta-ferramenta orientável, 466
POSFS, 609
Posição básica da orientação da ferramenta
ORIRESET, 346
Posições singulares, 355
POT, 79
Precisão de contorno
Programável, 543
Pré-furação de bolsão de contorno – CYCLE64
programação externa, 755
Preparação de contorno
Resposta de erro, 727
PREPRO, 184
Pré-processamento
Memória, 545
PRESETON, 318
PRESETONS, 320
PRIO, 134
PRLOC, 31
Process DataShare, 689
Profundidade de imersão, 445
Programa
- Saltos, 107
Bifurcação, 110
de inicialização, 235
Endereçamento do,
Memória, 224
Repetição, 202
Tempos de processamento, 684
Programa de inicialização, 235
Programação da orientação, 356
programação externa, 858
Programação indireta
das linhas do programa de usinagem, 76
de códigos G, 72
de endereços, 70
dos Atributos de posição, 73
PROTA, 409
PROTD, 411
PROTS, 410
PSI
na orientação ao longo da superfície do cone, 358
Polinômios de orientação, 365
PTP, 388
PTPG0, 388
PTPWOC, 388
PUD, 32
PUNCHACC, 647
Puncionamento, 647
PUTFTOC, 440
PUTFTOCF, 439
PW, 260

Q

Qualquer posição - CYCLE802
programação externa, 806

R

Ranhura circular - SLOT2
programação externa, 743
Ranhura longitudinal - SLOT1
programação externa, 740
READ, 157
REAL, 31
REDEF, 37
RELEASE, 144
REP, 55
REPEAT, 112
REPEATB, 112
Repetição de partes do programa
com programação indireta CALL, 206
REPOSA, 554
REPOSH, 554
REPOSHA, 554
REPOSL, 554
REPOSQ, 554
REPOSQA, 554
RET, 186
RET (parametrizável), 187
RETB (parametrizável), 193
Retificação inclinada de canal, 384
Retração rápida do contorno, 137
Retrocesso
comando autárquico, 702
controlado pelo NC, 697
RG, 27
RINDEX, 94
RMBBL, 554
RMEBL, 554
RMIBL, 554
RMNBL, 554

Rosqueamento com macho com mandril de compensação - CYCLE840
programação externa, 818
Rosqueamento com macho sem mandril de compensação - CYCLE84
programação externa, 780
Rotação
do vetor de orientação, 366
Rotação – CYCLE800
programação externa, 800
Rotina de interrupção
Ativação / desativação, 136
Deletar, 136
Movimento de retrocesso, 139
Nova atribuição, 135
Retração rápida do contorno, 137
Sentido de deslocamento programável, 139
ROUND, 79
ROUNDUP, 162
Run MyScreens, 679

S

Saída
no dispositivo/arquivo externo, 689
Saliência circular - CYCLE77
programação externa, 765
Saliência retangular - CYCLE76
programação externa, 763
Salto
Marcadores de salto, 107
Vai para o início do programa, 106
SAVE, 174
SBLOF, 176
SBLON, 176
SCPARA, 673
SD, 260
SD41610, 422
SD41611, 422
SD42475, 376
SD42476, 376
SD42477, 376
SD42900, 432
SD42910, 432
SD42920, 432
SD42930, 433
SD42935, 435
SD42940, 436
SD42984, 490
Segmentação automática de curso, 652
Separação - CYCLE92
programação externa, 786
Sequência da ferramenta, 420
Sequência da peça, 420
Sequência de roscas - CYCLE98
programação externa, 789
SET, 55
SETAL, 694
SETDNO, 464
SETINT, 134
SETM, 127
SETTCOR, 508
Simulação de valor mestre, 600
SIN, 79
Sincronismo de posição, 609
Sincronismo de posição com deslocamento de ângulo, 609
Sincronismo do número rotações, 609
Sincronização
(fino), 612
aproximado, 612
sistema de quadros, 327
SLOT1 - Ranhura longitudinal
programação externa, 740
SLOT2 - Ranhura circular
programação externa, 743
SOFT, 536
SOFTA, 536
Solavanco
Correção do, 563
Limitação, 536
SON, 647
SONS, 647
SPF, 224
SPI, 661
SPIF1, 647
SPIF2, 647
Spline
- Tipos, 266
Interpolação, 260
SPLINEPATH, 271
SPN, 652
SPOF, 647
SPP, 652
SPRINT, 97
SQRT, 79
START, 127
STARTFIFO, 545
STAT, 389
STOPFIFO, 545
STOPRE, 545
String,
- Encadeamento, 91
- Operações, 89

comprimento, 93
formatar, 97
STRINGIS, 675
STRLEN, 93
Suavização
Do decurso de orientação, 378
Subrotina
Nome, 164
Sub-rotina
- Chamada com transferência de parâmetros, 199
- Chamada sem transferência de parâmetros, 197
- Chamada, indireta, 205
- Chamada, modal, 203
Aplicação, 163
Caminho de localização programável, 209
repetição, 202
-salto de retorno parametrizável (RET ...), 187
-salto de retorno parametrizável (RETB...), 193
SUBSTR, 95
SYNR, 31
SYNRW, 31
SYNW, 31

T

TAN, 79
TANG, 525
TANGDEL, 529
TANGOF, 529
TANGON, 528
TCARR, 472
TCOABS, 472
TCOFR, 472
TCOFRX, 472
TCOFRY, 472
TCOFRZ, 472
Tempo de processamento, 684
Tempo restante
para uma peça de trabalho, 686
THETA
na interpolação da rotação da ferramenta, 372
na rotação da orientação da ferramenta, 366
TILT, 347
Tipo de acoplamento, 612
Tipo de cinemática, 470
Tipos de transformação
Função geral, 333
TLIFT, 527
TMOF, 659
TMON, 659
TOFFOF, 475
TOFFON, 475
TOLOWER, 92
TOOLENV, 496
Torneamento de roscas - CYCLE99
programação externa, 793
TOUPPER, 92
TOWBCS, 433
TOWKCS, 433
TOWMCS, 433
TOWSTD, 433
TOWTCS, 433
TOWWCS, 433
TRAANG
com ângulo programável, 383
TRACON, 386
TRACYL, 380
TRAFOOF, 400
TRAFOON, 413
TRAILOF, 573
TRAILON, 573
Transformação com eixo linear giratório, 343
Transformação de ângulo oblíquo (TRAANG)
com ângulo programável, 383
Transformação de orientação TRAORI
Cinemática da máquina, 336
Movimentos de orientação, 335
Programação da orientação, 345
Transformações genéricas de 5 e 6 eixos, 336
Variantes da programação da orientação, 345
Transformação de superfície cilíndrica, 337
Transformação polar, 337
Transformações
encadeadas, 386
Posição básica da orientação da ferramenta independente da cinemática, 334
Transformação de orientação, 333
Transformação de três, quatro e cinco eixos, 344
Transformações cinemáticas, 334
Transformações encadeadas, 335
TRANSMIT, 380
TRAORI, 344
TRUE, 31
TRUNC, 79
TU, 393

U

ULI, 44
UNTIL, 125
Usinagem de canal de contorno - CYCLE952
programação externa, 832

V

Valor de ajuste, 427
Valor de desgaste, 427
Variáveis
    Conversão de tipos, 89
    definido(a) pelo usuário, 31
Variáveis de sistema
    Estado do apalpador de medição, 292
    limite de medição do sensor, 293
Variável
    Conversão de tipos, 88
Variável Frame
    Atribuir valores, 307
    Chamada de transformações de coordenadas, 301
    Variável Frame pré-definida, 303
VELOLIM, 564
VELOLIMA, 538
Vetor de direção, 350
vetor de orientação THETA, 366

W

WAITC, 609
WAITE, 127
WAITENC, 671
WAITM, 127
WAITMC, 127
WHEN, 631
WHEN-DO, 643
WHENEVER, 631
WHENEVER-DO, 643
WHILE, 124
WORKPIECE, 704
WRITE, 152

X

XML de usuário, 679
XOR, 81