SIEMENS

SINUMERIK

SINUMERIK 840D sl/828D
工作計畫

編程手冊

適用範圍

控制
SINUMERIK 840D sl / 840DE sl / 828D

CNC 軟體　版本 4.8 SP3

08/2018
6FC5398-2BP40-6MA2

前言
基本安全指示 1
彈性的 NC 程式設計 2
檔案與程式管理 3
保護區 4
特殊動作指令 5
座標轉換（框架） 6
轉換 7
動態鍊 8
動態鍊的碰撞避免 9
動態鍊的轉換 10
刀具偏移量 11
路徑移動行為 12
軸耦合 13
同步動作 14
振盪 15
沖孔與切片 16
研磨 17
附加功能 18
使用者材料移除程式 19
外部程式設計循環 20
表格 21
附錄 A

Промэлектроника, sales@prom-elec.com

法律聲明

警告事項意涵

為了您的人身安全以及避免財產損失，必須注意本手冊中的提示。有關人身安全的提示通過一個警告三角表示，僅與財產損失有關的提示不帶警告三角。

⚠ 危險
表示如果不采取相應的小心措施，將會導致死亡或者嚴重的人身傷害。

⚠ 警告
表示如果不采取相應的小心措施，可能導致死亡或者嚴重的人身傷害。

⚠ 小心
表示如果不采取相應的小心措施，可能導致輕微的人身傷害。

注意
表示如果不采取相應的小心措施，可能導致財產損失。

當出現多個危險等級的情況下，每次總是使用最高等級（較低數字）的警告提示。如果在某個警告提示中帶有警告可能導致人身傷害的警告三角，則可能在該警告提示中另外還附帶有可能導致財產損失的警告。

合格的崙業人員

唯有與各項工作要求**資格符合的人員**才能操作本文件所屬產品/系統，遵照各附帶文件說明，特別是其中的安全及警告提示。資格符合的人員由於具備相關訓練及經驗，可以察覺本產品/系統的風險，並避免可能的危險。

Siemens 產品

請注意下列說明：

⚠ 警告
Siemens 產品只允許用於目錄和相關技術文件中規定的使用情況。如果要使用其他公司的產品和組件，必須得到 Siemens 推薦和允許。正確的運輸、儲存、組裝、裝配、安裝、調試、操作和維護是產品安全、正常運行的前提。必須保證允許的環境條件。必須注意相關檔中的提示。

商標

所有帶有標記符號 ® 的都是 Siemens AG 的注冊商標。標簽中的其他符號可能是一些其他商標，任何第三方將其用于其他目的都會損坏所有者的利益。

責任免除

我們已對印刷品中所述內容與硬件和軟件的一致性作過檢查。然而不排除存在偏差的可能性，因此我們不保証印刷品中所述內容與硬件和軟件完全一致。印刷品中的數據都按規定經過檢測，必要的修正值包含在下一版本中。同時歡迎您提出改進建議。

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
德國

文檔訂購號: 6FC5398-2BP40-6MA2
℗ 08/2018 本公司保留更改的權利

西門子股份有限公司版權所有 Siemens AG 1995 - 2018.
保留所有權利

前言

SINUMERIK 文件

SINUMERIK 文件可分為以下類別：

- 一般文件 / 目錄
- 使用者文件
- 製造商 / 服務文件

其他資訊

您可以在以下位址 (https://support.industry.siemens.com/cs/de/en/view/108464614)找到下列主題的資訊：

- 訂購文件/文件簡介
- 其他下載文件的連接
- 使用線上文件（在手冊/資訊中尋找及搜尋）

若您有關於技術文件的任何問題（例如建議事項、修正），請傳送電子郵件至下列位址 (mailto:docu.motioncontrol@siemens.com)。

mySupport / 文件

您可以在下列位址 (https://support.industry.siemens.com/My/cn/zh/documentation)，了解如何根據西門子內容建立自己的個別文件，並修改成自己的機台文件：

訓練

您可以在下列位址 (http://www.siemens.com/sitrain)，找到有關 SITRAIN 的資訊（產品、系統、自動化與驅動器解決方案的相關西門子訓練）。

FAQ

產品支援 (https://support.industry.siemens.com/cs/de/en/ps/faq)的「維修支援中心」頁面上，提供了常見問題集。

SINUMERIK

您可以在以下位址 (http://www.siemens.com/sinumerik)找到 SINUMERIK 的相關資訊：

目標群組

本出版品適用於下列對象：

- 程式設計師
- 專案工程師

用途

使用本編程手冊，目標群組群得以研發、撰寫、測試以及修正程式與軟體使用者介面。

標準範圍

本程式設計手冊說明標準範圍的功能。 機器刀具製造商所作之功能擴充或更改由機器刀具製造商自行提供文件。

未於文件中述及的其他功能，有可能會在控制系統中執行。 然而，並不代表應於新控制系統內或維修時提供此類功能。

此外，為了讓文件清晰，本文件不包含所有產品類型的詳細資訊，且無法涵蓋所有可理解的安裝案例、操作或維護。

技術支援

網站上「聯絡資訊」區域的下列地址 (https://support.industry.siemens.com/sc/cn/zh/sc/-/oid2090)中有各國特定的技術支援服務電話號碼。

結構與內容的資訊

程式設計手冊，基本原理 / 工作計畫

NC 程式設計說明分成兩份手冊：

1. **基礎**
   此“基礎”編程手冊乃針對具鑽孔、銑削及車削操作方面適當專業技能的操作員使用。 簡易的程式設計範例用於解釋指令及敘述，同樣也是根據 DIN 66025 定義。
2. **工作計畫**
   “進階”編程手冊係針對具進階程式設計知識的技術人員。 藉著特殊程式設計語言，SINUMERIK 控制系統讓使用者能進行複雜的工件程式設計（例如：任意形狀的表面、通道協調,...），並可輕易完成複雜作業的程式設計。

所敘述的 NC 語言元素的可用性

在手冊中所敘述的 NC 語言元素在 SINUMERIK 840D sl 都有提供。 與 SINUMERIK 828D 相關的可用性應從表 "操作：SINUMERIK 828D 的支援度 (頁 965)" 取得。

目錄

前言 3

1 基本安全指示 19

1.1 一般安全指示 19

1.2 應用範例的保證與責任 20

1.3 工業安全性 21

2 彈性的 NC 程式設計 23

2.1 變數 23
2.1.1 系統資料 23
2.1.2 預定義使用者變數：算術參數 26
2.1.2.1 通道專屬算術參數 (R) 26
2.1.2.2 全域算術參數 (RG) 27
2.1.3 預定義使用者變數： 連結變數 29
2.1.4 使用者變數的定義（DEF） 31
2.1.5 系統資料、使用者資料及 NC 指令重新定義 (REDEF) 37
2.1.6 屬性：初始化值 41
2.1.7 屬性： 臨界值（LLI、ULI） 45
2.1.8 屬性： 實體元件（PHU） 46
2.1.9 屬性： 存取權力（APR、APW、APRP、APWP、APRB、APWB） 49
2.1.10 屬性： 資料層級（DCM、DCI、DCU）- 僅限 SINUMERIK 828D 54
2.1.11 可定義與可重新定義屬性之概觀 55
2.1.12 陣列變數（DEF，SET，REP）之定義與初始化 56
2.1.13 陣列變數（DEF，SET，REP）之定義與初始化： 其它資訊 61
2.1.14 資料類型 63
2.1.15 檢查變數 (ISVAR) 的可用性 64
2.1.16 讀取屬性值 / 資料類型（GETVARPHU、GETVARAP、GETVARLIM、GETVARDIM、GETVARDFT、GETVARTYP） 65

2.2 間接程式設計 72
2.2.1 間接程式設計位址 72
2.2.2 間接程式設計 G 指令 74
2.2.3 間接程式設計定位屬性 (GP) 76
2.2.4 間接程式設計工件程式行（EXECSTRING） 79

2.3 運算功能 80

2.4 比較與邏輯運算 83

2.5 比較錯誤（TRUNC）的精確性修正 85

2.6 變數最小，最大與範圍（MINVAL，MAXVAL 與 BOUND） 87

2.7 運算優先順序....89
2.8 可行類型轉換....90
2.9 字串運算....91
2.9.1 類型轉換為 STRING（AXSTRING）....91
2.9.2 從 STRING 類型轉換（NUMBER、ISNUMBER、AXNAME）....92
2.9.3 串鍊字串（<<）....94
2.9.4 轉換為大小寫字母（TOLOWER、TOUPPER）....95
2.9.5 決定字串長度（STRLEN）....96
2.9.6 在字串中搜尋字元 / 字串（INDEX、RINDEX、MINDEX、MATCH）....96
2.9.7 選擇子字串（SUBSTR）....97
2.9.8 個別字元的讀取和寫入....98
2.9.9 格式化字串 (SPRINT)....100
2.10 程式跳躍與分支....110
2.10.1 傳回跳躍至程式開始處（GOTOS）....110
2.10.2 程式跳躍至跳躍標記（GOTOB、GOTOF、GOTO、GOTOC）....111
2.10.3 程式分支（CASE ... OF ... DEFAULT ...）....114
2.11 重複程式區段（REPEAT，REPEATB，ENDLABEL，P）....117
2.12 檢查結構....124
2.12.1 條件式陳述及分支（IF、ELSE、ENDIF）....125
2.12.2 連續程式迴圈（LOOP，ENDLOOP）....127
2.12.3 計數迴圈（FOR ... TO ...，ENDFOR）....127
2.12.4 在迴圈起點處，附有條件的程式迴圈（WHILE，ENDWHILE）....129
2.12.5 在迴圈結尾處，附有條件的程式迴圈（REPEAT，UNTIL）....130
2.12.6 具有巢狀檢查結構的程式範例....131
2.13 一致性指令（INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM）....132
2.14 中斷常式（ASUB）....138
2.14.1 中斷常式的函數....138
2.14.2 建立一個中斷常式....139
2.14.3 指派並啟動中斷常式（SETINT，PRIO，BLSYNC）....140
2.14.4 停用 / 重新啟用指派中斷常式（DISABLE，ENABLE）....142
2.14.5 刪除指派的中斷常式（CLRINT）....142
2.14.6 從輪廓快速回退（SETINT LIFTFAST，ALF）....143
2.14.7 從輪廓快速回退的移動方向....146
2.14.8 供中斷程式所用的順序....149
2.15 座標軸替換，主軸替換（RELEASE，GET，GETD）....150
2.16 將座標軸傳輸至其他通道（AXTOCHAN）：....155
2.17 啟動機械參數（NEWCONF）....157
2.18 寫入檔案（WRITE）....158
2.19 刪除檔案（DELETE）....163

2.20 在檔案中讀入行（READ）...165
2.21 確認檔案（ISFILE）是否存在...168
2.22 讀出檔案資訊（FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO）...170
2.23 無條件進位（ROUNDUP）...173
2.24 副程式技術...174
2.24.1 一般資訊...174
2.24.1.1 副程式...174
2.24.1.2 副程式名稱...175
2.24.1.3 巢狀的副程式...176
2.24.1.4 搜尋路徑...177
2.24.1.5 形式與實際參數...177
2.24.1.6 參數傳輸...178
2.24.2 副程式的定義...180
2.24.2.1 沒有參數傳輸的副程式：...180
2.24.2.2 具有傳值呼叫參數傳輸（PROC）的副程式...181
2.24.2.3 具有傳址呼叫參數傳輸（PROC、VAR）的副程式...182
2.24.2.4 儲存模態 G 碼功能（SAVE）...185
2.24.2.5 抑制獨立單節執行（SBLOF，SBLON）...186
2.24.2.6 抑制目前的單節顯示（DISPLOF、DISPLON、ACTBLOCNO）...191
2.24.2.7 以準備（PREPRO）定義副程式...195
2.24.2.8 副程式返回 M17...195
2.24.2.9 RET 副程式返回...196
2.24.2.10 可參數化的副程式傳回跳躍（RET ...）...198
2.24.2.11 可參數化的子程式傳回跳躍（RETB ...）...204
2.24.3 副程式呼叫...208
2.24.3.1 沒有參數傳輸的呼叫：...208
2.24.3.2 以參數傳輸（EXTERN）進行副程式呼叫...210
2.24.3.3 程式重覆的次數（P）...213
2.24.3.4 模態副程式呼叫（MCALL）...214
2.24.3.5 間接子程式呼叫（CALL）...216
2.24.3.6 使用呼叫程式部份的規格間接進行副程式呼叫（CALL BLOCK ... TO ...）...217
2.24.3.7 間接呼叫以 ISO 語言（ISOCALL）進行程式設計的程式...218
2.24.3.8 以路徑規格和參數 (PCALL) 呼叫副程式...219
2.24.3.9 供副程式呼叫（CALLPATH）所用的延伸搜尋路徑...220
2.24.3.10 執行外部副程式 (840D sl) (EXTCALL)...221
2.24.3.11 執行外部副程式 (828D) (EXTCALL)...225
2.25 巨集技術（DEFINE ... AS）...230
3 檔案與程式管理...233
3.1 程式記憶體...233
3.1.1 NC 程式記憶體...233
3.1.2 外部程式記憶體...236

3.1.3 定址程式記憶體檔案 ... 237
3.1.4 搜尋子程式呼叫的路徑 ... 242
3.1.5 詢問路徑及檔案名稱 ... 243
3.2 工作記憶體（CHANDATA，COMPLETE，INITIAL） ... 245
4 保護區 ... 249
4.1 定義保護區（CPROTDEF，NPROTDEF） ... 249
4.2 啟動 / 停用保護區（CPROT、NPROT） ... 254
4.3 檢查保護區差異、工作區限制與軟體極限開關（CALCPOSI） ... 258
5 特殊動作指令 ... 271
5.1 逼近編碼位置（CAC、CIC、CDC、CACP、CACN） ... 271
5.2 曲線插補（ASPLINE、BSPLINE、CSPLINE、BAUTO、BNAT、BTAN、EAUTO、ENAT、ETAN、PW、SD、PL） ... 273
5.3 曲線群組 (SPLINEPATH) ... 285
5.4 啟用 / 停用 NC 單節壓縮（COMPON、COMPCURV、COMPCAD、COMPSURF、COMPOF） ... 287
5.5 多項式插補（POLY、POLYPATH、PO、PL） ... 288
5.6 可設定路徑參考（SPATH、UPATH） ... 294
5.7 以觸發式探針量測 (MEAS, MEAW) ... 297
5.8 軸專屬測量（MEASA、MEAWA、MEAC）（選項） ... 300
5.9 OEM 使用者的特殊函數 (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829). ... 310
5.10 含轉角減速的進給率降低功能 (FENDNORM, G62, G621) ... 311
5.11 可程式設計的動作結尾條件 (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) ... 312
6 座標轉換（框架） ... 315
6.1 透過框架變數座標轉換 ... 315
6.1.1 預先定義的框架變數（$P_CHBFRAME、$P_IFRAME、$P_PFRAME、$P_ACTFRAME） ... 317
6.2 值指定至框架 ... 322
6.2.1 指派直接值（軸值、角度、比例） ... 322
6.2.2 讀取及變更框架元件（TR、FI、RT、SC、MI） ... 324
6.2.3 使用框架計算 ... 325
6.2.4 框架變數定義 (DEF FRAME) ... 327
6.3 粗調或微調偏移量（CTRANS、CFINE） ... 328
6.4 外部零點偏移量 ($AA_ETRANS) ... 330
6.5 設定喪失參考狀態的實際值 (PRESETON) ... 332

6.6 設定並未喪失參考狀態的實際值 (PRESETONS)..334
6.7 從空間中的三個量測點計算框架（MEAFRAME）..336
6.8 NCU 全域框架..340
6.8.1 通道專屬框架（$P_CHBFR、$P_UBFR）..341
6.8.2 通道中有效框架..341
7 轉換..347
7.1 轉換類型的一般程式設計..347
7.1.1 轉換的方向移動..349
7.1.2 方向轉換概觀 TRAORI..353
7.2 三、四及五軸轉換（TRAORI）..356
7.2.1 通用刀具頭的一般關係..356
7.2.2 三、四及五軸轉換（TRAORI）..359
7.2.3 方向程式設計及初始設定的變數（ORIRESET）..360
7.2.4 編寫刀具方向 (A..., B..., C..., LEAD, TILT)..362
7.2.5 平面銑削 (A4, B4, C4, A5, B5, C5)..368
7.2.6 方向軸參考（ORIWKS、ORIMKS）：..369
7.2.7 程式設計方向軸（ORIAXES，ORIVECT，ORIEULER，ORIRPY，ORIRPY2，ORIVIRT1，ORIVIRT2）..371
7.2.8 沿著錐形的柱面進行方向程式設計（ORIPLANE，ORICONCW，ORICONCCW，ORICONTO，ORICONIO）..374
7.2.9 兩接點的方向規格（ORICURVE、PO[XH]=、PO[YH]=、PO[ZH]=）..377
7.3 方向多項式（PO[角度]、PO[座標]）..380
7.4 刀具方向的旋轉（ORIROTA、ORIROTR、ORIROTT、ORIROTC、THETA）..382
7.5 相對於路徑的方向..385
7.5.1 相對於路徑的方向類型..385
7.5.2 相對於路徑的刀具方向旋轉（ORIPATH、ORIPATHS、旋轉角度）..386
7.5.3 相對於路徑的刀具旋轉插補（ORIROTC、THETA）..388
7.5.4 方向特性的平滑化（ORIPATHS A8=、B8=、C8=）..390
7.6 方向的壓縮（COMPON、COMPCURV、COMPCAD、COMPSURF）..392
7.7 啟用 / 停用方向特性（ORISON，ORISOF）..395
7.8 動態轉換..397
7.8.1 啟動平面結尾處轉換 (TRANSMIT)..397
7.8.2 啟動柱體表面轉換 (TRACYL)..397
7.8.3 使用可程式設計角度 (TRAANG) 啟動傾斜角轉換..400
7.8.4 研磨機的傾斜直進切削（G5、G7）..401
7.9 啟動序連轉換 (TRACON)..404
7.10 直角座標 PTP 移動..406
7.10.1 啟動 / 停用直角座標 PTP 移動（PTP、PTPG0、PTPWOC、CP）..406
7.10.2 指定接點的位置 (STAT)..407

7.10.3 指定軸角度 (TU) 的符號 ..412
7.10.4 範例 1：含 ROBX 轉換的 6 軸機械手臂 PTP 移動 ..415
7.10.5 範例 2：含一般 5 軸轉換之 PTP 移動 ..416
7.10.6 範例 3：PTPG0 與 TRANSMIT ..416
7.11 選擇轉換時的限制 ..419
7.12 取消選擇轉換 (TRAFOOF) ..420
8 動態鍊 ..421
8.1 刪除元件 (DELOBJ) ..421
8.2 以名稱 (NAMETOINT) 的方法判定索引 ..426
9 動態鍊的碰撞避免 ..429
9.1 檢查碰撞對 (COLLPAIR) ..430
9.2 要求重新計算碰撞避免的機台模型 (PROTA) ..432
9.3 設定保護區狀態 (PROTS) ..433
9.4 判定兩個指定保護區的間隙 (PROTD) ..434
10 動態鍊的轉換 ..437
10.1 啟動轉換 (TRAFOON) ..437
10.2 在機台測量之後，修改方向轉換 (CORRTRAFO) ..439
11 刀具偏移量 ..447
11.1 偏移記憶體 ..447
11.2 附加偏移 ..451
11.2.1 選擇附加偏移（DL） ..451
11.2.2 指定磨損及設定值（$TC_SCPxy[t，d]，$TC_ECPxy[t，d]） ..452
11.2.3 刪除附加偏移（DELDL） ..453
11.3 刀具補正的特殊處理方式 ..455
11.3.1 刀長的鏡像 ..456
11.3.2 磨損符號評估 ..457
11.3.3 生效加工操作的座標系統（TOWSTD、TOWMCS、TOWWCS、TOWBCS、TOWTCS、TOWKCS） ..458
11.3.4 刀長及平面變更 ..461
11.4 線上刀具偏移量 ..462
11.4.1 定義多項式函數 (FCTDEF) ..462
11.4.2 持續寫入線上刀具偏移 (PUTFTOCF) ..464
11.4.3 分散寫入線上刀具偏移量 (PUTFTOC) ..465
11.4.4 啟用/停用線上刀具偏移量 (FTOCON/FTOCOF) ..465
11.5 3D 刀具半徑補正 ..467
11.5.1 選擇 3D 刀具半徑補正進行 3D 圓周銑削（CUT3DC、CUT3DCD、ISD） ..467

11.5.2 選擇 3D 刀具半徑補正進行 3D 端面銑削（CUT3DF、CUT3DFS、CUT3DFF、CUT3DFD）...471
11.5.3 3D 圓周銑削將限制表面考量在內 (CUT3DCC, CUT3DCCD)...477

11.6 刀具方向（ORIC、ORID、OSOF、OSC、OSS、OSSE、ORIS、OSD、OST）...483

11.7 自由定義 D 編號,切削刀刃編號...489
11.7.1 任意指派 D 數量、刀刃數量（CE 位址）...489
11.7.2 任意指派 D 數量 檢查 D 數量（CHKDNO）...489
11.7.3 任意指派 D 數量 重新命名 D 數量（GETDNO，SETDNO）...490
11.7.4 任意指派 D 數量 決定指定 D 數量的 T 數量（GETACTTD）...491
11.7.5 任意指派 D 數量無效的 D 數量（DZERO）...491

11.8 刀把動力學...493

11.9 用於可定向刀把的刀具長度補正（TCARR，TCOABS，TCOFR，TCOFRX，TCOFRY，TCOFRZ）...499

11.10 線上刀長補正（TLC）（TOFFON，TOFFOF）...502

11.11 可旋轉刀具偏移資料之修改...505
11.11.1 計算方向 (ORISOLH)...505
11.11.2 啟動可旋轉刀具的偏移資料之修改 (CUTMOD, CUTMODK)...515

11.12 使用刀具環境工作...524
11.12.1 儲存刀具環境 (TOOLENV)...524
11.12.2 刪除刀具環境 (DELTOOLENV)...526
11.12.3 讀取 T、D 與 DL 編號 (GETTENV)...527
11.12.4 讀取有關已儲存刀具環境的資訊 ($P_TOOLENVN, ($P_TOOLENV)...528
11.12.5 讀取刀長及 / 或刀長元件 (GETTCOR)...529
11.12.6 變更刀具元件 (SETTCOR)...536

11.13 將刀具長度 L1、L2、L3 的指派讀入座標軸 (LENTOAX)...551

12 路徑移動行為...555

12.1 切線控制...555
12.1.1 定義耦合 (TANG)...555
12.1.2 啟用中間單節產生 (TLIFT)...557
12.1.3 啟用耦合 (TANGON)...558
12.1.4 停用耦合 (TANGOF)...559
12.1.5 刪除耦合 (TANGDEL)...560

12.2 進給率特性 (FNORM, FLIN, FCUB, FPO)...562

12.3 加速度行為...567
12.3.1 加速模式（BRISK、BRISKA、SOFT、SOFTA、DRIVE、DRIVEA）...567
12.3.2 在跟隨軸（VELOLIMA，ACCLIMA，JERKLIMA）上加速的影響...569
12.3.3 啟用技術專屬動態值（DYNNORM、DYNPOS、DYNROUGH、DYNSEMIFIN、DYNFINISH）...571

12.4 具有前饋控制的移動 (FFWON, FFWOF)...573

12.5 可程式設計的輪廓精確度 (CPRECON, CPRECOF)....574
12.6 具有前置處理記憶體（STOPFIFO，STARTFIFO，FIFOCTRL，STOPRE）的程式順序....576
12.7 定義一個停止延遲範圍 (DELAYFSTON, DELAYFSTOF)....579
12.8 防止 SERUPRO 的程式定位 (IPTRLOCK, IPTRUNLOCK)....582
12.9 重新定位至輪廓（REPOSA，REPOSL，REPOSQ，REPOSQA，REPOSH，REPOSHA，DISR，DISPR，RMIBL，RMBBL，RMEBL，RMNBL）....584
12.10 影響動作控制....593
12.10.1 比例震動校正（JERKLIM）....593
12.10.2 比例速率修正（VELOLIM）....594
12.10.3 JERKLIM 和 VELOLIM 的程式範例....596
12.11 程式設計輪廓 / 方向允差 (CTOL, OTOL, ATOL)....597
12.12 啟用耦合功能的單節變更行為 (CPBC)....602
13 軸耦合....603
13.1 耦合動作（TRAILON，TRAILOF）....603
13.2 曲線表（CTAB）....608
13.2.1 定義曲線表（CTABDEF、CATBEND）....608
13.2.2 確認曲線表是否存在（CTABEXISTS）....614
13.2.3 刪除曲線表（CTABDEL）....615
13.2.4 鎖住曲線表以防止刪除與覆寫（CTABLOCK、CTABUNLOCK）....616
13.2.5 曲線表： 決定取線表特性（CTABID、CTABISLOCK、CTABMEMTYP、CTABPERIOD）....617
13.2.6 讀取曲線表值（CTABTSV、CTABTEV、CTABTSP、CTABTEP、CTABSSV、CTABSEV、CTAB、CTABINV、CTABTMIN、CTABTMAX）....619
13.2.7 曲線表： 檢查資源的使用（CTABNO，CTABNOMEM，CTABFNO，CTABSEGID，CTABSEG，CTABFSEG，CTABMSEG，CTABPOLID，CTABPOL，CTABFPOL，CTABMPOL）....624
13.3 軸主動值耦合 (LEADON, LEADOF)....626
13.4 電子齒輪（EG）....632
13.4.1 定義電子齒輪（EGDEF）....632
13.4.2 切換至電子變速箱（EGON、EGONSYN、EGONSYNE）....633
13.4.3 切換至電子變速箱（EGOFS，EGOFC）....636
13.4.4 刪除電子齒輪的定義（EGDEL）....637
13.4.5 旋轉進給率（G95）/ 電子齒輪（FPR）....638
13.5 同步主軸....639
13.5.1 同步主軸： 程式設計（COUPDEF、COUPDEL、COUPON、COUPONC、COUPOF、COUPOFS、COUPRES、WAITC）....639
13.6 一般耦合 (CP...)....650

13.7 主 / 從耦合 (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)...657
14 同步動作...661
14.1 同步動作之定義...661
15 振盪...663
15.1 非同步震盪（OS、OSP1、OSP2、OST1、OST2、OSCTRL、OSNSC、OSE、OSB）...663
15.2 由同步動作控制的震盪（OSCILL）...669
16 沖孔與切片...677
16.1 啟用 / 停用...677
16.1.1 啟用 / 停用沖孔與切片（SPOF、SON、PON、SONS、PONS、PDELAYON、PDELAYOF、PUNCHACC）...677
16.2 自動路徑分段...683
16.2.1 路徑軸的路徑分割...685
16.2.2 單一軸的路徑分割...687
17 研磨...689
17.1 啟動 / 停用研磨專屬刀具監控（TMON、TMOF）...689
18 附加功能...691
18.1 軸函數（AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL）...691
18.2 可更換的幾何軸（GEOAX）...694
18.3 軸容器（AXCTSWE、AXCTSWED、AXCTSWEC）...699
18.4 等候有效的軸位置 (WAITENC)...701
18.5 可程式參數集轉換 (SCPARA)...703
18.6 檢查顯示 NC 語言之範圍（STRINGIS）...705
18.7 由工件程式相互呼叫視窗（MMC）...709
18.8 程式執行時間 / 工件計數器...715
18.8.1 程式執行時間...715
18.8.2 工件計數器...719
18.9 Process DataShare - 輸出至外部裝置 / 檔案（EXTOPEN、WRITE、EXTCLOSE）：...721
18.10 警報（SETAL）...726
18.11 延伸停止及回退 (ESR)...728
18.11.1 NC 控制 ESR...729
18.11.1.1 NC 控制回退 (POLF, POLFA, POLFMASK, POLFMLIN)...729
18.11.1.2 NC 控制停止...733
18.11.2 驅動整合式 ESR...734

18.11.2.1 設定整合式驅動停止 (ESRS)....734
18.11.2.2 設定整合式驅動回退 (ESRS)....735
18.12 定義坯料 (WORKPIECE)....737
18.13 切換語言模式 (G290, G291)....742
19 使用者材料移除程式....745
19.1 材料移除的支援函數....745
19.2 產生輪廓表（CONTPRON）....746
19.3 產生編碼輪廓表（CONTDCON）....753
19.4 決定兩輪廓元素間的交點（INTERSEC）....757
19.5 按單節執行表的輪廓元素（EXECTAB）....759
19.6 計算圓弧資料（CALCDAT）....760
19.7 停用輪廓準備（EXECUTE）....762
20 外部程式設計循環....763
20.1 技術循環....763
20.1.1 簡介....763
20.1.2 技術專屬概觀....764
20.1.3 HOLES1 - 排孔....766
20.1.4 HOLES2 - 孔洞圓....766
20.1.5 POCKET3 - 銑削矩形凹槽....769
20.1.6 POCKET4 - 銑削圓形凹槽....773
20.1.7 SLOT1－長形槽....777
20.1.8 SLOT2－環槽....780
20.1.9 LONGHOLE - 細長孔洞....783
20.1.10 CYCLE60 - 刻字循環....786
20.1.11 CYCLE61－平面銑削....788
20.1.12 CYCLE62 - 輪廓呼叫....791
20.1.13 CYCLE63－銑削輪廓挖槽....792
20.1.14 CYCLE64－預鑽孔輪廓挖槽....796
20.1.15 CYCLE70－螺紋銑削....797
20.1.16 CYCLE72－路徑銑削....800
20.1.17 CYCLE76 - 矩形島嶼銑削....804
20.1.18 CYCLE77 - 圓弧島嶼銑削....806
20.1.19 CYCLE78－鑽孔螺紋銑削....809
20.1.20 CYCLE79－多邊緣....812
20.1.21 CYCLE81 - 鑽孔、定心....815
20.1.22 CYCLE82 - 鑽孔、逆鏜孔....816
20.1.23 CYCLE83－深孔鑽....819
20.1.24 CYCLE84－不搭配補正夾頭進行攻牙....823
20.1.25 CYCLE85－鉸孔....827
20.1.26 CYCLE86－鏜孔....829

20.1.27 CYCLE92－切離 ..830
20.1.28 CYCLE95－輪廓切削 ..832
20.1.29 CYCLE98－螺紋鍊 ..834
20.1.30 CYCLE99－螺紋車削 ..841
20.1.31 CYCLE435 - 設定修整器座標系統 ..846
20.1.32 CYCLE495 - 形狀修整 ..847
20.1.33 CYCLE800 - 旋轉 ..849
20.1.34 CYCLE801 - 網格或框架 ..853
20.1.35 CYCLE802－任意位置 ..855
20.1.36 CYCLE830 - 深孔鑽 2 ..859
20.1.37 CYCLE832 - 高速設定 ..865
20.1.38 CYCLE840 - 搭配補正夾頭進行攻牙 ..869
20.1.39 CYCLE899－銑削開孔槽 ..872
20.1.40 CYCLE930－槽 ..875
20.1.41 CYCLE940 - 底切形狀 ..878
20.1.42 CYCLE951－材料移除 ..882
20.1.43 CYCLE952 - 輪廓切槽 ..885
20.1.44 CYCLE4071 - 在反轉點進給的縱向研磨 ..893
20.1.45 CYCLE4072 - 於反轉點進給的縱向研磨和取消訊號 ..895
20.1.46 CYCLE4073 - 連續進給之縱向研磨 ..898
20.1.47 CYCLE4074 - 連續進給的縱向研磨和取消訊號 ..899
20.1.48 CYCLE4075 - 在反轉點進給的表面研磨 ..903
20.1.49 CYCLE4077 - 於反轉點進給的表面研磨和取消訊號 ..905
20.1.50 CYCLE4078 - 連續進給之表面研磨 ..908
20.1.51 CYCLE4079 - 間歇進給之表面研磨 ..910
20.1.52 GROUP_BEGIN - 程式單節開始處 ..913
20.1.53 GROUP_END - 程式單節結束 ..914
20.1.54 GROUP_ADDEND - 嘗試切削添加 ..914
20.1.55 補充條件 ..915
20.1.55.1 循環畫面表單的技術刻度比例 ..915
20.2 測量循環 ..917
21 表格 ..919
21.1 操作 ..919
21.2 操作：SINUMERIK 828D 的支援度 ..965
21.2.1 控制器版本銑削 / 車削 ..965
21.2.2 控制器版本研磨 ..1004
21.3 目前設定在 HMI 中的語言 ..1042
A 附錄 ..1045
A.1 縮寫表 ..1045
A.2 文件概觀 ..1056

字彙表..... 1057
索引..... 1079

基本安全指示

1

1.1 一般安全指示

⚠ 警告

若未遵守安全指示與殘餘風險，可能會造成人員生命危險

若未遵守硬體文件相關的安全指示與殘餘風險，可能會造成人員重傷或死亡。

- 請遵守硬體文件內的安全指示。
- 將殘餘風險納入風險評估。

⚠ 警告

錯誤或經變更的參數設定會造成機器故障

錯誤或經變更的參數設定會導致機台故障，並導致傷害或死亡。

- 保護參數設定（參數指派）不受未經授權的存取。
- 處理可能的故障應採取適當的措施，例如緊急停止或緊急關閉。

1.2 應用範例的保證與責任

應用範例不具約束力，未聲稱設定、設備為完整內容，亦不得用於任何可能衍生的不可測性。應用範例並不代表客戶專屬解決方案，但僅能用於針對一般工作提供支援。

作為使用者必須負責確保產品依說明正確地操作。應用範例不得解除您在使用、安裝、操作及維護設備時，對於安全處理應負之責任。

1.3 工業安全性

說明

工業安全性

西門子提供具備工業安全性功能的產品及解決方案，可支援工廠、系統、機器及網路的安全操作。

為保護工廠、系統、機器及網路免於網路威脅，務必實行並持續維護完整且最先進的工業安全概念。西門子的產品及解決方案僅為此概念的其中一部分構成要素。

客戶需負責避免未經授權之人存取工廠、系統、機器及網路。這些系統、機械及元件只能在有需要且於必要範圍內並採取適當安全措施（例如，使用防火牆及／或網路區隔）之情況下，才能連接至企業網路或網際網路及其延伸網路。

如需可能需要實行的工業安全性措施相關資訊，請造訪：

工業安全 (http://www.siemens.com/industrialsecurity)

為提升產品安全性，西門子持續研發產品和解決方案。西門子強烈建議客戶於產品更新發佈後立即更新，並使用最新的產品產品。使用已不再支援的產品，和未能套用最新的更新，可能使客戶更容易受到網路威脅。

如需關於產品更新的最新資訊，請訂閱西門子工業安全性 RSS 摘要：

工業安全 (http://www.siemens.com/industrialsecurity)

若需進一步資訊，請至網站：

工業安全設定手冊 (https://support.industry.siemens.com/cs/cn/zh/view/108862708/en)

⚠ 警告

因軟體操作而導致的不安全操作狀態

軟體操縱（例如，病毒、特洛伊木馬程式、惡意軟體或蠕蟲）可能導致系統發生不安全的操作狀態，進而引發生命危險、嚴重傷害及財產損失。

- 將軟體保持在最新的狀態。
- 將安裝環境或機台的自動化與驅動元件整合至完整且最先進的工業安全概念。
- 確保您已將所有已安裝的產品納入完整工業安全概念。
- 請利用合適的保護措施（如病毒掃描程式）來保護儲存在可交換式儲存媒體上的檔案遠離惡意軟體。
- 啟動「Know-how protection」驅動功能以保護設備不受未經授權的變更。

彈性的 NC 程式設計

2

2.1 變數

使用區域系統資料和使用者資料中的變數，尤其搭配算術函數和檢查結構時，可寫入高彈性的 NC 程式及循環。

- 系統資料
  系統資料包含系統中預先定義的變數。這些變數有定義的意義。主要供系統軟體使用。使用者可讀取這些變數並寫入 NC 程式和循環。範例：機械參數、設定資料、系統變數。
  雖然系統資料項目的意義是固定的，但使用者可在一定的限度內以重新定義來修改其屬性。
  請參閱「系統資料、使用者資料及 NC 指令重新定義 (REDEF) (頁 37)」
- 使用者資料
  使用者資料包含使用者所定義的這些變數，其意義完全由使用者定義。系統不會對其進行評估。
  使用者資料分成：
  - 預定義使用者變數
    預定義使用者變數為系統中已定義的變數且透過機械參數將其編號參數化。使用者可對這些變數的屬性做改變。
    請參閱「系統資料、使用者資料及 NC 指令重新定義 (REDEF) (頁 37)」。
  - 使用者自訂變數
    使用者自訂變數為由使用者定義且系統在未達執行時間不會產生的變數。這些編號、資料類型、可見性、及其他特性均僅能由使用者定義。
    請參閱「使用者變數的定義（DEF） (頁 31)」

2.1.1 系統資料

系統資料包含變數，變數為系統預先定義且可存取 NC 程式及循環目前的控制參數設定，及存取機台、控制和處理狀態的變數。

前置處理變數

前置處理變數為前置處理期間讀取和寫入的系統資料；換言之便是轉譯包含變數的單節。前置處理變數不會觸發前置處理停止。

主要執行變數

主要執行變數為主要執行期間讀取和寫入的系統資料；換言之便是執行包含變數的單節。以下為主要執行變數：

- 可在同步動作（讀取 / 寫入）中程式設計的變數
- 可程式設計在 NC 程式中並觸發前置處理停止（讀取 / 寫入）的變數
- 可程式設計在 NC 程式中且其值可在前置處理時加以計算但在未達主要執行時不會寫入的變數（主要執行同步：僅可寫入）

前置系統

為能區別系統資料與其他資料，其名稱前面通常含有 $ 號並接著一或兩個字母及底線。

$ + 1.字母	**含義：資料類型**
預處理資料（在前置處理時讀取 / 寫入的系統資料）	
$M	機械參數 [1)]
$S	設定資料，保護區 [1)]
$T	刀具管理資料
$P	程式設計值
$C	ISO 封套循環的循環變數
$O	選項資料
R	R 參數（算術參數）[2)]
主要執行資料（在主要執行時讀取 / 寫入的系統資料）	
$$M	機械參數 [1)]
$$S	設定資料 [1)]
$A	目前主要執行資料
$V	定位控制器資料
$R	R 參數（算術參數）[2)]
[1)] 依據寫入值包含的 $ 字元數（一個或兩個），決定機械參數和設定資料為前置處理或主要執行變數。特定應用的標記可自由選擇。 [2)] R 參數在工件程式 / 循環中作為前置處理變數使用時，會省略前置符號，例如 R10。當其在同步動作中做為主要執行變數使用時，會在前面加上一個$號，例如$R10。	

第二個字母	含義：可見性
N	NC 全域變數 (NC)
C	通道專屬變數（**通道**）
A	軸專屬變數（**軸**）

補充條件

前置系統的例外

以下的變數系統與上述指定的前置系統有所不同：

- $TC_...:此處第二個字母 C 並非表示通道專屬系統變數，而是指刀把專屬系統變數（TC= 刀盤）。
- $P_ ...:通道專屬系統變數

在同步動作中使用機械參數及設定資料

若於同步動作中使用機械參數及設定資料，前置字元可用於定義是否在前置處理執行或主要執行時，同時讀取 / 寫入機械參數或設定資料。

若在加工時該資料維持不變，其可與前置處理執行同時讀取。為此，機械參數或設定資料前面會加上 1 個 $ 號：

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

若在加工時該資料改變，則其必須與主要執行同時讀取/寫入。為此，機械參數或設定資料前面會加上 2 個 $ 號：

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

說明

寫入機械參數與設定資料

當寫入機械參數或設定資料時，確保在工件程式/循環執行時啟動的存取層級允許寫入且該資料能「立即」生效是很重要的。

參考資料

所有系統變數的完整概觀請參閱：

清單手冊，系統變數

另請參見

變數 (頁 23)

2.1.2 預定義使用者變數：算術參數

2.1.2.1 通道專屬算術參數 (R)

通道專屬算術參數或 R 參數為具有 R 符號的預定義使用者變數，其定義為一個實數資料類型的陣列。基於歷史因素，R 參數可同時標記為有陣列索引，例如 R[10]，及無陣列索引，例如 R10。

當採用同步動作時，前置字元需含$符號，例如$R10。

語法

當作為前置處理變數使用時：
R<n>
R[<算式>]

當作為主要執行變數使用時：
$R<n>
$R[<算式>]

含義

R：	做為前置處理變數時的識別碼，例如在工件程式中	
$R：	做為主要執行變數時的識別碼，例如在同步動作中	
	類型：	REAL
	值域：	對於非指數標記法： ± (0.000 0001 ... 9999 9999) **注意：** 最多 8 個小數點位數
		對於指數標記法： ± ($1*10^{-300}$ ... $1*10^{+300}$) **注意事項：** ● 標記法：<mantissa>EX<exponent> 如 8.2EX-3 ● 包含符號與小數點，最多允許 10 個字元。

<n>：	R 參數的編號	
	類型：	INT
	值域：	0 - MAX_INDEX **請注意** MAX_INDEX 是從參數化的 R 參數編號計算得到的： MAX_INDEX =（MD28050 $MN_MM_NUM_R_PARAM）- 1
<算式>：	陣列索引 所有的算式均可作為陣列索引，只要該算式結果能轉換成整數資料類型（整數、實數、布林值、字元）即可。	

範例

在數學函數中配置並應用 R 參數：

程式碼	註解
R0=3.5678	；在前置處理中配置
R[1]=-37.3	；在前置處理中配置
R3=-7	；在前置處理中配置
$R4=-0.1EX-5	；在主程式執行中配置：R4 = -0.1 * 10^-5
$R[6]=1.874EX8	；在主程式執行中配置：R6 = 1.874 * 10^8
R7=SIN(25.3)	；在前置處理中配置
R[R2]=R10	；使用 R 參數間接定址
R[(R1+R2)*R3]=5	；使用數學算式間接定址
X=(R1+R2)	；將軸 X 移動至 R1 及 R2 加總所產生的位置
Z=SQRT（R1*R1+R2*R2）	將軸 Z 移動至（R1^2 + R2^2）的平方根位置

另請參見

變數 (頁 23)

2.1.2.2 全域算術參數 (RG)

功能

除了通道專屬 R 參數，使用者可存取全域 R 參數，其會在控制單元內存在一次，可由所有通道讀取及寫入。

例如全域 R 參數可用於由通道傳輸資訊至下一個通道。另一項範例則是應針對所有通道評估全域設定，例如由主軸開始的原始工件懸長。

全域 R 參數在前置處理期間，由使用者介面或 NC 程式讀取及寫入。無法使用同步行動和技術循環。

說明

沒有同步在讀取及寫入全域 R 參數時於通道之間進行。

這是因為讀取及寫入於前置處理期間執行，並未定義通道寫入值在其他通道生效時的時間點。

範例：

在通道 1，迴路以全域 R 參數執行作為迴路計數器。通道 2 寫入值至此全域 R 參數；這樣會在通道 1 造成迴路取消。不過於通道 1 前置處理可轉譯的所有迴路仍可執行。迴路數量並未定義，需視通道載入等因素而定。

使用者必須在通道之間實作同步作為應用，例如使用 WAIT 旗標！

語法

在 NC 程式中寫入

```
RG[<n>]=<值>
RG[<算式>]=<值>
```

在 NC 程式中讀取

```
R...=RG[<n>]
R...=RG[<算式>]
```

含義

RG ：	**全域** R 參數 NC 位址預設名稱 **說明：** NC 位址名稱可透過 MD15800 $MN_R_PARAM_NCK_NAME 設定	
<n>：	全域 R 參數的編號	
	類型：	INT
	值域：	0 ... MAX_INDEX **附註** MAX_INDEX 是從參數化的全域 R 參數編號計算而得： MAX_INDEX = (MD18156 $MN_MM_NUM_R_PARAM_NCK) - 1

<算式>：	所有的算式均可作為陣列索引，只要該算式結果能轉換成 INT 資料類型（INT、REAL、BOOL、CHAR）即可。	
<值>：	全域 R 參數的值	
	類型：	REAL
	值域：	對於非指數標記法： ± (0.000 0001 ... 9999 9999) **附註：** 最多 8 個小數點位數
		對於指數標記法： ± ($1*10^{-300}$ ... $1*10^{+300}$) **附註：** ● 標記法：<mantissa>EX<exponent> 如 8.2EX-3 ● 包含符號與小數點，最多允許 10 個字元。

2.1.3 預定義使用者變數： 連結變數

連結變數可用於「NCU 連結」函數中作為連結在同一網路上 NCU 間的循環資料交換。 其可加速對連結變數記憶體的資料格式專屬存取。連結變數記憶體由使用者 / 機台製造商以系統專屬的方式同時透過大小及資料結構加以定義。

連結變數為系統全域使用者變數，可由在已設定連結通訊的連結中所涵蓋的所有 NCU 讀取及寫入至工件程式及循環中。 有別於全域使用者變數 (GUD)，連結變數亦可用在同步動作中。

在無啟用 NCU 連結的系統中，連結變數可用在控制系統本機上作為全域使用者變數 (GUD) 旁的額外全域使用者變數。

語法

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

含義

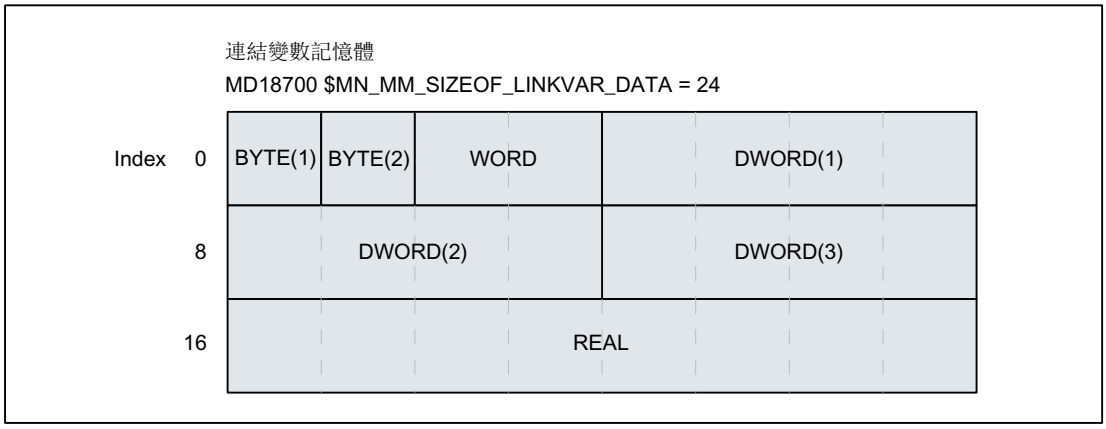
$A_DLB:	BYTE 資料格式（1 位元組）的連結變數	
	資料類型：	UINT
	值域：	0 ... 255
$A_DLW:	WORD 資料格式（2 位元組）的連結變數	
	資料類型：	INT
	值域：	-32768 ... 32767
$A_DLD:	DWORD 資料格式（4 位元組）的連結變數	
	資料類型：	INT
	值域：	-2147483648 ... 2147483647
$A_DLR:	REAL 資料格式（8 位元組）的連結變數	
	資料類型：	REAL
	值域：	±(2,2*10^{-308} … 1,8*10^{+308})
<索引>：	位址索引以位元組為單位，從連結變數記憶體的起點開始計算	
	資料類型：	INT
	值域：	0 - MAX_INDEX **附註** • MAX_INDEX 可從連結變數記憶體的參數化大小計算得到： MAX_INDEX = (MD18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1 • 僅有索引可程式設計故定址在連結變數記憶體中的位元組位於資料格式臨界值 ⇒ 索引= n * 位元組，其中 n = 0、1、2 等。 – $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, ...

範例

一個自動化系統含有 2 個 NCU（NCU1 及 NCU2）。 機械軸 AX2 連接至 NCU1。 其以 NCU2 的連結軸方式移動。

NCU1 反覆的將軸 AX2 的實際電流值 ($VA_CURR) 寫入至連結變數記憶體中。 NCU2 反覆讀取透過連結通訊傳送的實際電流值，並在超過臨界值時顯示警報 61000。

連結變數記憶體的資料結構如下圖所示。 實際電流值以 REAL 值傳輸。



NCU1

NCU1 利用連結變數 $A_DLR[16] 在靜態同步動作中的插補循環時，反覆將軸 AX2 的實際電流值寫入至連結變數記憶體中。

程式碼

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

NCU2

NCU2 利用連結變數 $A_DLR[16] 在靜態同步動作中的插補循環時，反覆將軸 AX2 的實際電流值讀取至連結變數記憶體。 若實際電流值大於 23.0 A，便會顯示警報 61000。

程式碼

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

另請參見

變數 (頁 23)

2.1.4 使用者變數的定義（DEF）

使用 DEF 指令，可定義使用者專屬變數，或使用者變數（使用者資料），並為其指派值。

根據有效範圍區分（換言之，便是該變數的可見範圍），使用者變數分為以下幾類：

- 局部使用者變數 (LUD)
  局部使用者變數 (LUD) 為定義在執行時不屬於主要程式之 NC 程式中的變數。其會在呼叫 NC 程式時產生，並在程式結束重置或下次控制系統開啟電源時刪除。局部使用者變數僅能在定義他們的 NC 程式中存取。
- 程式全域使用者變數 (PUD)
  程式全域使用者變數 (PUD) 為在做為主要程式之 NC 程式中定義的使用者變數。其會在呼叫 NC 程式時產生，並在程式結束重置或下次控制系統開啟電源時刪除。可以在主程式和主程式的所有子程式中存取 PUD。

  說明

  程式全域使用者變數 (PUD) 的可用性

  定義在主程式中的程式全域使用者變數 (PUD) 僅在設定以下機械資料時於子程式中提供：

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  若 MD11120 = 0，則主程式中所定義的程式全域使用者變數只可於主程式中提供。
- 全域使用者變數 (GUD)
  全域使用者變數 (GUD) 為在資料單節（SGUD、MGUD、UGUD、GUD4 至 GUD9）中定義的 NC 或通道全域變數，甚至在程式結束重置後，或是下次控制系統開機時還能保持。所有 NC 程式都可以對 GUD 進行存取。

使用者變數需在使用（讀取 / 寫入）前先做定義。此時，必須注意以下規則：

- 必須在定義檔案中定義 GUD，例如 _N_DEF_DIR/_N_UGUD_DEF。
- PUD 及 LUD 必須在 NC 程式的定義區段中定義。
- 該資料必須在專用的單節中定義。
- 每個資料定義僅能使用一種資料類型。
- 相同資料類型的數個變數可針對各個資料定義做定義。

語法

LUD 和 PUD

```
DEF <類型> <物理單位> <臨界值> <名稱>[<值_1>，<值_2>，<值_3>]=<初始值>
```

GUD

```
DEF <範圍> <前置處理停止> <存取權限> <資料層級> <類型> <物理單位> <條件數值> <名稱>[<值_1>，<值_2>，<值_3>]=<初始值>
```

含義

DEF：	用於定義 GUD、PUD、LUD 使用者變數的指令	
<範圍>：	有效範圍，僅和 GUD 有關：	
	NC：	NC 全域使用者變數
	CHAN：	通道全域使用者變數
<PP_stop>：	前置處理停止，僅與 GUD（選用）有關	
	SYNR：	讀取時前置處理停止
	SYNW：	寫入時前置處理停止
	SYNRW：	當讀取 / 寫入時，前置處理停止
<存取權限>：	透過 NC 程式或 OPI（選用）讀取 / 寫入 GUD 的保護等級	
	APRP <保護等級>：	讀取：NC 程式
	APWP <保護等級>：	寫入：NC 程式
	APRB <保護等級>：	讀取：OPI
	APWB <保護等級>：	寫入：OPI
	<保護等級>：	值域：0 ... 7
	請參閱「屬性： 存取權力（APR、APW、APRP、APWP、APRB、APWB） (頁 49)」	
<資料層級>：	資料層級指派（僅限 SINUMERIK 828D）	
	DCM：	資料層級 M（= 製造商）
	DCI：	資料層級 I（= 個人）
	DCU：	資料層級 U（= 使用者）
	請參閱「屬性： 資料層級（DCM、DCI、DCU）- 僅限 SINUMERIK 828D (頁 54)」。	
<類型>：	資料類型：	
	INT：	含正負號之整數
	REAL：	實數（LONG REAL 至 IEEE）
	BOOL：	真值 TRUE （=1），FALSE （=0）
	CHAR：	ASCII 字元
	STRING[最大長度>]：	已定義長度的字元字串
	AXIS：	軸 / 主軸識別碼
	FRAME：	靜態座標轉換的幾何資料
	請參閱「資料類型 (頁 63)」	

<物理單位>：	實體元件（選用）	
	PHU <單位>：	實體元件
	請參閱「屬性： 實體元件（PHU） (頁 46)」	
<臨界值>：	上 / 下臨界值（選用）	
	LLI <臨界值>：	下臨界值（下臨界值）
	ULI <臨界值>：	上臨界值（上臨界值）
	請參閱「屬性： 臨界值（LLI、ULI） (頁 45)」	
<名稱>：	變數名稱 **附註** • 最多 31 個字元 • 前兩個字元必須為一個字母及 / 或一個底線。 • $符號僅保留給系統變數故不可使用。	
[<值_1>, <值_2>, <值_3>]：	1 至最大 3 相陣列變數的陣列大小規格（選用） 有關陣列變數的初始化，請參閱「陣列變數（DEF，SET，REP）之定義與初始化 (頁 56)」	
<init_value>：	初始化值（選用） 請參閱「屬性：初始化值 (頁 41)」 有關陣列變數的初始化，請參閱「陣列變數（DEF，SET，REP）之定義與初始化 (頁 56)」	

範例

範例 1：機台製造商資料單節中的使用者變數定義

程式碼	註解
%_N_MGUD_DEF	；GUD 單節：機台製造商
$PATH=/_N_DEF_DIR	
DEF CHAN REAL PHU 24 LLI 0 ULI 10 STROM_1, STROM_2	
；說明	
；2 個 GUD 項目的定義：STROM_1, STROM_2	
；有效範圍：貫穿通道	
；資料類型：REAL	
PP 停止：未程式設計=>預設值=無 PP 停止	
；實體單元：24 = [A]	
；臨界值：低 = 0.0，高 = 10.0	
；存取權限：未程式設計=>預設值= 7 =按鍵操作開關位置 0	

程式碼	註解
；初始化值：未程式設計=>預設值= 0.0	
DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 ZEIT_1=12, ZEIT_2=45	
；說明	
；2 個 GUD 項目的定義：ZEIT_1, ZEIT_2	
；有效範圍：整個 NC	
；資料類型：REAL	
PP 停止：未程式設計=>預設值=無 PP 停止	
；實體單元：13 = [s]	
；臨界值：低= 10.0，高=未程式設計=>上定義範圍臨界值	
；存取權限：	
；NC 程式：寫入 / 讀取 = 3 = 終端使用者	
；OPI：寫入 = 0 = Siemens，讀取 = 3 = 終端使用者	
；初始化值：ZEIT_1 = 12.0, ZEIT_2 = 45.0	
DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "計數器"	
；說明	
；1 個 GUD 項目的定義：GUD5_NAME	
；有效範圍：整個 NC	
；資料類型：字串，最大 5 個字元	
PP 停止：未程式設計=>預設值=無 PP 停止	
；實體單元：未程式設計=>預設值= 0 =無實體元件	
；臨界值：未程式設計的 => 定義範圍臨界值：低 = 0，高 = 255	
；存取權限：	
；NC 程式：寫入 / 讀取 = 3 = 終端使用者	
；OPI：寫入 = 0 = Siemens，讀取 = 3 = 終端使用者	
；初始化值：「計數器」	
M30	

範例 2：全域程式及局部使用者變數 (PUD/LUD)

程式碼	註解
PROC MAIN	；主程式
DEF INT VAR1	；PUD 定義
...	
SUB2	；子程式呼叫
...	
M30	

程式碼	註解
PROC SUB2	；子程式 SUB2
DEF INT VAR2	；LUD DEFINITION
...	

程式碼	註解
IF (VAR1==1)	；讀取 PUD
VAR1=VAR1+1	；讀寫 PUD
VAR2=1	；寫入 LUD
ENDIF	
SUB3	；子程式呼叫
...	
M17	

程式碼	註解
PROC SUB3	；子程式 SUB3
...	
IF (VAR1==1)	；讀取 PUD
VAR1=VAR1+1	；讀寫 PUD
VAR2=1	；錯誤：從 SUB2 來的 LUD 不明
ENDIF	
...	
M17	

範例 3：資料類型軸的使用者變數定義與用法

程式碼	註解
DEF AXIS ABSCISSA	；第一幾何軸
DEF AXIS SPINDLE	;主軸
...	
IF ISAXIS(1) == FALSE GOTOF CONTINUE	
ABSCISSA = $P_AXN1	
CONTINUE：	
...	
SPINDLE=(S1)	；第一主軸
OVRA[SPINDLE]=80	；主軸手動倍率調整 = 80%
SPINDLE=(S3)	；第三主軸

補充條件

全域使用者變數 (GUD)

在全域使用者變數（GUD）的定義關係中，需將以下機械資料列入考慮：

編號	識別碼：$MN_	含義
11140	GUD_AREA_ SAVE_TAB	GUD 單節的輔助儲存
18118 [1)]	MM_NUM_GUD_MODULES	在啟動檔案系統中的數個 GUD 檔案
18120 [1)]	MM_NUM_GUD_NAMES_NCK	數個全域 GUD 名稱

編號	識別碼：$MN_	含義
18130 1)	MM_NUM_GUD_NAMES_CHAN	通道專屬 GUD 名稱的數目
18140 1)	MM_NUM_GUD_NAMES_AXIS	數個軸專屬 GUD 名稱
18150 1)	MM_GUD_VALUES_MEM	全域 GUD 值的記憶體位置
18660 1)	MM_NUM_SYNACT_GUD_REAL	REAL 資料類型的可設定 GUD 數目
18661 1)	MM_NUM_SYNACT_GUD_INT	INT 資料類型的可設定 GUD 數目
18662 1)	MM_NUM_SYNACT_GUD_BOOL	BOOL 資料類型的可設定 GUD 數目
18663 1)	MM_NUM_SYNACT_GUD_AXIS	AXIS 資料類型的可設定 GUD 數目
18664 1)	MM_NUM_SYNACT_GUD_CHAR	CHAR 資料類型的可設定 GUD 數目
18665 1)	MM_NUM_SYNACT_GUD_STRING	STRING 資料類型的可設定 GUD 數目

1) 對 SINUMERIK 828D 而言，MD 僅可讀取！

軸資料類型的 NC 全域使用者變數之跨通道應用

在資料單節使用軸識別碼定義時，所產生的 軸 資料類型 NC 全域使用者變數，僅有在其他 NC 通道軸具有相同編號時，才可在其他 NC 通道中使用。

若否，則該變數需置於 NC 程式起點，否則，需如下列範例一樣使用 AXNAME(...) 函數（請參閱「軸函數（AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL） (頁 691)」）。

程式碼	註解
`DEF NCK STRING[5] ACHSE="X"`	；資料單節內的定義
`...`	
`N100 AX[AXNAME(ACHSE)]=111 G00`	；在 NC 程式中使用

2.1.5 系統資料、使用者資料及 NC 指令重新定義 (REDEF)

REDEF 指令可變更系統資料、使用者資料及 NC 指令的屬性。重新定義的基本條件為其對應的定義需已過期。

在重新定義時不可同時變更多的屬性。需針對要變更的各個屬性程式設計個別的 REDEF 指令。

若已程式設計多個同時發生的屬性變更，最後的變更永遠有效。

重設屬性值

隨著 REDEF 改變的存取權限及初始化時間的屬性可以藉由重新程式設計 REDEF 重設為預設值，其後接著變數名稱或 NC 語言指令：

- 存取權限：保護層級 7
- 初始化時間：現值未初始化或保留

可重新定義的屬性

請參閱「可定義與可重新定義屬性之概觀 (頁 55)」。

區域使用者變數 (PUD/LUD)

局部使用者變數不可重新定義 (PUD/LUD)。

語法

```
REDEF <名稱> <前置處理停止>
REDEF <名稱> <物理單位>
REDEF <名稱> <臨界值>
REDEF <名稱> <存取權限>
REDEF <名稱> <初始時間>
REDEF <名稱> <初始時間> <初始值>
REDEF <名稱> <資料類別>
REDEF <名稱>
```

含義

REDEF：	某些屬性重新定義的指令，或重設系統變數、使用者變數及 NC 語言指令的「存取權限」及 / 或「初始化時間」屬性	
<名稱>：	已定義的變數或 NC 語言指令之名稱	
<前置處理停止>：	前置處理停止	
	SYNR：	讀取時前置處理停止
	SYNW：	寫入時前置處理停止
	SYNRW：	當讀取 / 寫入時，前置處理停止

<物理單位>：	實體元件	
	PHU <單位>：	實體元件
	請參閱「屬性： 實體元件（PHU） (頁 46)」。 **附註** 無法重新定義的項目： • 系統變數 • 全域使用者資料 (GUD) 的資料類型：布林、軸、字串、框架	
<臨界值>：	上 / 下臨界值	
	LLI <臨界值>：	下臨界值（下臨界值）
	ULI <臨界值>：	上臨界值（上臨界值）
	請參閱「屬性： 臨界值（LLI、ULI） (頁 45)」。 **附註** 無法重新定義的項目： • 系統變數 • 全域使用者資料 (GUD) 的資料類型：布林、軸、字串、框架	
<存取權限>：	透過工件程式或 OPI 讀取 / 寫入的存取權限	
	APX <保護等級>：	執行：NC 語言元件
	APRP <保護等級>：	讀取：工件程式
	APWP <保護等級>：	寫入：工件程式
	APRB <保護等級>：	讀取：OPI
	APWB <保護等級>：	寫入：OPI
	<保護等級>：	值域：0 ... 7
	請參閱「屬性： 存取權力（APR、APW、APRP、APWP、APRB、APWB） (頁 49)」。	
<初始時間>：	變數重新初始化的時間點	
	INIPO：	啟動
	INIRE：	主程式結束，NC 重置或啟動
	INICF：	NEWCONF 或主程式尾端，NC 重置或啟動
	PRLOC：	主程式結束，NC 在本機變更或啟動之後重置
	請參閱「屬性：初始化值 (頁 41)」。	

<init_value>：	初始化值 每當初始化值重新定義時，亦需指定初始化時間（請參閱 <初始時間>）。 請參閱「屬性：初始化值 (頁 41)」。 有關陣列變數的初始化，請參閱「陣列變數（DEF，SET，REP）之定義與初始化 (頁 56)」。 **附註** 系統變數無法重新定義，設定資料除外。
<資料層級>：	資料層級指派（僅限 SINUMERIK 828D）
	DCM： 資料層級 M（= 製造商）
	DCI： 資料層級 I（= 個人）
	DCU： 資料層級 U（= 使用者）
	請參閱「屬性： 資料層級（DCM、DCI、DCU）- 僅限 SINUMERIK 828D (頁 54)」。

範例

重新定義機台製造商資料單節中的系統變數 $TC_DPCx

程式碼

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

關於 N100： 寫入存取：OPI = 保護等級 2，工件程式 = 保護等級 3

關於 N200： 實體元件 [%]

關於 N300： 下臨界值 = 0，上臨界值 = 200

關於　　陣列變數在啟動時以四個值產生。
N400：

關於　　重設「存取權限」及 / 或「初始化時間」屬性值
N800 /
N900

說明

使用存取檔案

如果使用存取檔案，需將存取權限從 _N_MGUD_DEF 重新分配至 _N_MACCESS_DEF。

補充條件

粒度

重新定義每次均套用至以名稱做為識別的整個變數中。其中，陣列變數並不支援將不同屬性指派至個別陣列元素中。

2.1.6　屬性：初始化值

定義 (DEF) 使用者變數

在定義時可先將初始化值指派給以下使用者變數：

- 全域使用者變數 (GUD)
- 程式全域使用者變數 (PUD)
- 局部使用者變數 (LUD)

重新定義 (REDEF) 系統及使用者變數

在重新定義時可先將初始化值指派給以下變數：

- 系統資料
  - 設定資料
- 使用者資料
  - R 參數
  - 同步動作變數（$AC_PARAM、$AC_MARKER、$AC_TIMER）
  - 同步動作 GUD（SYG_xy[ ]，其中 x=R、I、B、A、C、S 而 y=S、M、U、4 至 9）
  - EPS 參數
  - 刀具資料 OEM
  - 刀庫資料 OEM
  - 全域使用者變數 (GUD)

重新初始化時間

在重新定義時可指定變數重新初始化（即重置為初始化值）的時間點。

- INIPO（啟動）
  該變數在啟動時重新初始化。
- INIRE（重置）
  該變數在 NC 重置、模式群組重置、工件程式 (M02/M30) 結束或啟動時重新初始化。
- INICF (NEWCONF)
  對於功能「設定機械資料啟用」，變數係透過 HMI、工件程式指令 NEWCONF 或 NC 重置、模式群組重置、工件程式尾端 (M02 / M30) 或電源開啟重新初始化。
- PRLOC（程式局部變更）
  該變數僅在 NC 重置、模式群組重置或工件程式（M02/M30）結束時若在目前的工件程式中有變更時才重新初始化。
  PRLOC 屬性僅能和可程式設計的設定資料一同變更（請參閱下表）。

表格 2-1　可程式設計的設定資料

編號	識別碼	G 指令 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

編號	識別碼	G 指令 1)
43220	$SA_SPIND_MAX_VELO_G26	G26
43230	$SA_SPIND_MAX_VELO_LIMS	LIMS
43300	$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
43420	$SA_WORKAREA_LIMIT_PLUS	G26
43430	$SA_WORKAREA_LIMIT_MINUS	G25
43510	$SA_FIXED_STOP_TORQUE	FXST
43520	$SA_FIXED_STOP_WINDOW	FXSW
43700	$SA_OSCILL_REVERSE_POS1	OSP1
43710	$SA_OSCILL_REVERSE_POS2	OSP2
43720	$SA_OSCILL_DWELL_TIME1	OST1
43730	$SA_OSCILL_DWELL_TIME2	OST2
43740	$SA_OSCILL_VELO	FA
43750	$SA_OSCILL_NUM_SPARK_CYCLES	OSNSC
43760	$SA_OSCILL_END_POS	OSE
43770	$SA_OSCILL_CTRL_MASK	OSCTRL
43780	$SA_OSCILL_IS_ACTIVE	OS
43790	$SA_OSCILL_START_POS	OSB
1) 此 G 指令用於處理設定資料。		

補充條件

初始化值：全域使用者變數 (GUD)

- 僅 INIPO（啟動）可定義具 NC 有效範圍之全域使用者變數 (GUD) 的初始化時間。
- 除了 INIPO（啟動），INIRE（重置）或 INICF (NEWCONF)可用於定義具 CHAN 有效範圍之全域使用者變數 (GUD) 的初始化時間。
- 在具有 CHAN 有效範圍的全域使用者變數 (GUD) 及 INIRE（重置）或 INICF (NEWCONF) 初始化時間的情況中，當 NC 重置、模式群組重置及「啟動機械資料」時，這些變數僅可在已命名事件觸發時方可在通道中重新初始化。

初始化值：FRAME 資料類型

不可對 FRAME 資料類型的變數指定初始化值。FRAME 資料類型的變數不會明確的初始化且通常配合預設框架進行。

初始化值：CHAR 資料類型

CHAR 資料類型的變數，除了可用 ASCII 碼 (0...255) 表示以外，也可以藉由把對應的字元放在雙引號中來表示，例如"A"。

初始化值：資料類型 STRING

STRING 資料類型的變數，字串必須放在雙引號中，例如...= "MACHINE_1"

初始化值：AXIS 資料類型

軸 資料類型的變數，採延伸位址標記，軸識別碼必須放在括號中，例如...= (X3)。

初始化值：系統變數

針對系統變數，重新定義功能不可用於定義使用者專屬初始化值。系統變數的初始化值有系統指定且不可變更。然而，重新定義可用於在系統變數重新初始化時變更時間點（INIRE、INICF）。

不明確初始化值：AXIS 資料類型

針對 AXIS 資料類型的變數，會使用以下的不明確初始化值：

- 系統資料：「第一幾何軸」
- 同步動作 GUD（記為：SYG_A*）、PUD、LUD：
  以下機械資料中的軸指定：MD20082 $MC_AXCONF_CHANAX_DEFAULT_NAME

不明確初始化值：刀具與刀庫資料

刀具與刀庫資料的初始化值可用以下機械資料加以定義：MD17520 $MN_TOOL_DEFAULT_DATA_MASK

說明

同步

當全域變數讀入至不同位置為使用者 / 機台製造商的專屬責任時，由重新初始化該變數所觸發的事件之同步。

另請參見

變數 (頁 23)

2.1.7 屬性： 臨界值（LLI、ULI）

定義範圍的上下臨界值僅可定義為下列資料類型：

- INT
- REAL
- CHAR

定義（DEF）使用者變數： 臨界值及不明確初始化值

若在定義上述資料類型的使用者變數時未定義明確的初始化值，該變數會設定為該資料類型的不明確初始化值。

- INT： 0
- REAL： 0.0
- CHAR： 0

若該不明確初始化值超出程式設計的臨界值所定義之範圍，則該變數為以最接近不明確初始化值的臨界值進行初始化：

- 不明確初始化值 **<** 下臨界值 (LLI) ⇒
  初始化值 **=** 下臨界值
- 不明確初始化值 **>** 上臨界值列 (ULI) ⇒
  初始化值 **=** 上臨界值

範例：

程式碼	註解
`DEF REAL GUD1`	;下臨界值 = 定義範圍臨界值 ;上臨界值 = 定義範圍臨界值 ;程式未設定初始化值 ;=> 不明確初始化值 = 0.0
`DEF REAL LLI 5.0 GUD2`	;下臨界值 = 5.0 ;上臨界值 = 定義範圍臨界值 ;=> 初始化值 = 5.0
`DEF REAL ULI -5 GUD3`	;下臨界值 = 定義範圍臨界值 ;上臨界值 = -5.0 ;=> 初始化值 = -5.0

重新定義 (REDEF) 使用者變數： 臨界值與目前實際值

當重新定義了使用者變數的臨界值時，若其變更使目前實際值超出新定義的範圍，則會發出警報並拒絕該臨界值。

說明

重新定義 (REDEF) 使用者變數

當重新定義了使用者變數的臨界值時，需特別小心以確保後續的值均能一致地變更：

- 臨界值
- 實際值
- 以 INIPO、INIRE 或 INICF 為基礎重新定義或自動重新初始化時的初始化值

另請參見

變數 (頁 23)

2.1.8 屬性： 實體元件（PHU）

僅下列資料類型的變數可指定物理單位：

- INT
- REAL

可程式設計的實體元件 (PHU)

物理單位係指定為定點數： `PHU <單位>`：

可程式設計的實體元件如下：

<單位>	含義	實體元件
0	非實體元件	-
1	線性或角度位置 1) 2)	[mm]、[inch]、[degree]
2	線性位置 2)	[mm]、[inch]
3	角度位置	[degree]
4	線性或角度速率 1) 2)	[mm/min]、[英吋 / 分]、[rpm]
5	線性速率 2)	[mm/min]
6	角度速率	[rpm]
7	線性或角度加速度 1) 2)	[m/s^2]、[$inch/s^2$]、[rev/s^2]

<單位>	含義	實體元件
8	線性加速度 2)	[m/s²]、[inch/s²]
9	角度加速度	[rev/s²]
10	線性或旋轉 JERK 1) 2)	[m/s³]、[inch/s³]、[rev/s³]
11	線性加加速度 2)	[m/s³]、[inch/s³]
12	角度加加速度	[rev/s³]
13	時間	[s]
14	位置控制器增益	[16.667/s]
15	迴轉進給率 2)	[mm/rev]、[inch/rev]
16	溫度補正 1) 2)	[mm]、[inch]
18	力	[N]
19	重量	[kg]
20	重量慣性矩 3)	[kgm²]
21	百分比	[%]
22	頻率	[Hz]
23	電壓	[V]
24	電流	[A]
25	溫度	[°C]
26	角度	[degree]
27	KV	[1000/min]
28	線性或角度位置 3)	[mm]、[inch]、[degree]
29	切削率 2)	[m/min]、[feet/min]
30	圓周速率 2)	[m/s]、[feet/s]
31	阻抗	[ohm]
32	電感	[mH]
33	扭矩 3)	[Nm]
34	扭矩常數 3)	[Nm/A]
35	電流控制器增益	[V/A]
36	速度控制器增益 3)	[Nm/(rad*s)]
37	速度	[rpm]
42	功率	[kW]
43	電流，低	[µA]

<單位>	含義	實體元件
46	扭矩，低 3)	[µNm]
48	每 mil	-
49	-	[Hz/s]
65	流率	[l/min]
66	壓力	[bar]
67	體積 3)	[cm^3]
68	控制系統增益 3)	[mm/(V*min)]
69	力控制器控制系統增益	[N/V]
155	螺距 3)	[mm/rev]、[inch/rev]
156	螺距變化 3)	[mm/rev / rev]、[inch/rev / rev]
1) 物理單位視軸類型而定： 線性或旋轉軸		
2) 單位系統轉換 G70/G71（英制 / 公制） 使用 G70/G71 轉換基本系統之後 (MD10240 $MN_SCALING_SYSTEM_IS_METRIC)，在對於與長度有關的系統及使用者變數進行讀 / 寫操作時，數值**不**轉換（實際值、預設值和臨界值） G700/G710（英制 / 公制） 使用 G700/G710 轉換基本系統之後 (MD10240 $MN_SCALING_SYSTEM_IS_METRIC)，在對於與長度有關的系統及使用者變數進行讀 / 寫操作時，數值**會**轉換（實際值、預設值和臨界值）		
3) 該變數**不會**自動轉換成 NC 目前的度量衡系統（英制 / 公制）。 轉換為使用者 / 機台製造商的工作。		

說明

因格式轉換而造成溢位

所有具實體元件長度的使用者變數 (GUD/PUD/LUD) 之內部儲存格式為公制。 在 NCK 的主執行中，例如同步動作，過度使用這些類型的變數，會在切換度量衡系統時在插補階段造成 CPU 時間溢位，而產生警報 4240。

說明

元件的相容性

使用變數（配置、比較、計算等）時，需檢查相關元件的相容性。必要時需轉換，此為使用者 / 機台製造商的責任。

另請參見

變數 (頁 23)

2.1.9 屬性： 存取權力（APR、APW、APRP、APWP、APRB、APWB）

名稱

存取屬性 AP... 的名稱包括：

1. A：**存**取
2. P：**保**護
3. R / W：**讀**取 / **寫**入
4. P / O：**程**式 / **B**TSS (OPI)

存取權限 / 存取等級

在程式設計時需指定下列與存取權限對應的存取等級：

存取權益	保護等級
系統密碼	0
機台製造商密碼	1
服務密碼	2
最終使用者密碼	3
按鍵操作開關位置 3	4
按鍵操作開關位置 2	5
按鍵操作開關位置 1	6
按鍵操作開關位置 0	7

定義 (DEF) 使用者資料

以下資料可定義存取權限 (APR.../APW...)：

- 全域使用者資料 (GUD)

重新定義 (REDEF) 系統及使用者資料

以下資料可重新定義存取權限 (APR.../APW...)：

- 系統資料
  - 機械資料

    說明

    重新定義機械資料的讀取權限

    讀取機械資料的保護等級通常只能利用關鍵字 APR 在工件程式及 OPI 上設定。

    讀取權限的重新定義並不支援關鍵字 APRP 及 APRB，導致出現中斷訊息 12490「存取權限 APRP/APRB <保護等級> 並未設定」。
  - 設定資料
  - 系統變數
  - 製程資料
  - 刀庫資料
  - 刀具資料
- 使用者資料
  - R 參數
  - 同步動作變數（$AC_PARAM、$AC_MARKER、$AC_TIMER）
  - 同步動作 GUD（SYG_xy[ ]，其中 x=R、I、B、A、C、S 而 y=S、M、U、4 至 9）
  - EPS 參數
  - 刀具資料 OEM
  - 刀庫資料 OEM
  - 全域使用者變數 (GUD)

    說明

    在重新定義時，可任意的將該存取權限指派給介於最低保護等級 7 及特定保護等級，例如 1（機台製造商），間的變數中。

重新定義 (REDEF) NC 語言指令

下列 NC 語言指令的存取或執行權限 (APX) 可重新定義：

- G 指令 / 預備函數
  參考資料
  基礎編程手冊，章節：G 指令 / 預備函數
- 預先定義之函數
  參考資料
  基礎編程手冊，章節：預先定義之函數
- 預定義之子程式呼叫
  參考資料
  基礎編程手冊，章節：預定義之子程式呼叫
- DO 搭配同步動作操作
- 循環程式識別碼
  該循環需儲存在循環目錄中且必須含 PROC 操作。

與 NC 程式及循環有關的存取權限（APRP、APWP）

不同的存取權限可從 NC 程式或循環中的存取加速以下項目：

- APRP 0/APWP 0
  - 在 NC 程式處理時需設定系統密碼。
  - 該循環需儲存在_N_CST_DIR 目錄（系統）中。
  - 該執行權限需設定在系統_N_CST_DIR 目錄的 MD11160 $MN_ACCESS_EXEC_CST 中。
- APRP 1/APWP 1 或 APRP 2/APWP 2
  - 在 NC 程式處理時需設定機台製造商或服務密碼。
  - 該循環需儲存在_N_CMA_DIR（機台製造商）或_N_CST_DIR 目錄中。
  - 該執行權限至少需設定在機台製造商之 _N_CMA_DIR 或 _N_CST_DIR 目錄的機械資料 MD11161 $MN_ACCESS_EXEC_CMA 或 MD11160 $MN_ACCESS_EXEC_CST 中。

- APRP 3/APWP 3
  - 在 NC 程式執行期間，終端使用者的密碼必須加以設定。
  - 該循環需儲存在_N_CUS_DIR（使用者）、_N_CMA_DIR 或_N_CST_DIR 目錄中。
  - 該執行權限至少需設定在最終使用者之_N_CUS_DIR、_N_CMA_DIR 或 _N_CST_DIR 目錄的機械資料 MD11162 $MN_ACCESS_EXEC_CUS、MD11161 $MN_ACCESS_EXEC_CMA 或 MD11160 $MN_ACCESS_EXEC_CST 中。
- APRP 4...7/APWP 4...7
  - 在 NC 程式處理期間，鑰匙操作開關必須設為 3 ... 0。
  - 該循環需儲存在 _N_CUS_DIR、_N_CMA_DIR 或 _N_CST_DIR 目錄中。
  - 該執行權限至少需設定在對應按鍵操作開關位置之_N_CUS_DIR、_N_CMA_DIR 或 _N_CST_DIR 目錄的機械資料 MD11162 $MN_ACCESS_EXEC_CUS、MD11161 $MN_ACCESS_EXEC_CMA 或 MD11160 $MN_ACCESS_EXEC_CST 中。

與 OPI 有關的存取權限（APRB、APWB）

該存取權限（APRB、APWB）透過 OPI 平等的限制了所有系統元件 （HMI、PLC、外部計算機、EPS 服務、等）對系統的存取及使用者變數。

說明

局部 HMI 存取權限

當變更對系統資料的存取權限時，需特別小心以確保這些變更與使用 HMI 機制所定義的存取權限一致。

APR/APW 存取屬性

考量相容性，APR 和 APW 係不明確對映至屬性 APRP / APRB 和 APWP / APWB ：

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

使用 ACCESS 檔案的存取權限

當使用 ACCESS 檔案指派存取權限時，對於系統資料、使用者資料、及 NC 語言指令之存取權限僅能在這些 ACCESS 檔案中重新定義。但全域使用者資料 (GUD) 除外。針對此資料，需在對應的定義檔案 *_DEF 中重新定義存取權限。

為持續進行存取保護，該執行權限的機械資料及對應目錄的存取保護需一併修改。

原則上，步驟如下：

1. 建立必要的定義檔案：
   - _N_DEF_DIR/_N_SACCESS_DEF
   - _N_DEF_DIR/_N_MACCESS_DEF
   - _N_DEF_DIR/_N_UACCESS_DEF
2. 將定義檔案的寫入權限設定為重新定義所需要的值：
   - MD11170 $MN_ACCESS_WRITE_SACCESS = <保護等級>
   - MD11171 $MN_ACCESS_WRITE_MACCESS = <保護等級>
   - MD11172 $MN_ACCESS_WRITE_UACCESS = <保護等級>
3. 為存取循環中受保護的元件，需修改循環目錄_N_CST_DIR、_N_CMA_DIR 以及 _N_CST_DIR 的執行與寫入權限。
   執行權限
   - MD11160 $MN_ACCESS_EXEC_CST = <保護等級>
   - MD11161 $MN_ACCESS_EXEC_CMA = <保護等級>
   - MD11162 $MN_ACCESS_EXEC_CUS = <保護等級>

   寫入權限
   - MD11165 $MN_ACCESS_WRITE_CST = <保護等級>
   - MD11166 $MN_ACCESS_WRITE_CMA = <保護等級>
   - MD11167 MN_ACCESS_WRITE_CUS = <保護等級>

   該執行權限的保護等級至少需設定為與該元件所使用的保護等級相同。
   該寫入權限的保護等級至少需設定為該執行權限所使用的相同。
4. 局部 HMI 循環目錄之寫入權限的保護等級至少需設定為與局部 NC 循環目錄所使用的相同。
   參考資料
   操作手冊

ACCESS 檔案中的子程式呼叫

為進一步建構存取保護，可在 ACCESS 檔案中呼叫（SPF 或 MPF 識別碼）子程式。該子程式繼承了呼叫之 ACCESS 檔案的存取權限。

說明

ACCESS 檔案中僅能重新定義存取權限。其他的屬性需繼續在對應的定義檔案中程式設計/重新定義。

另請參見

變數 (頁 23)

2.1.10 屬性： 資料層級（DCM、DCI、DCU）- 僅限 SINUMERIK 828D

為了在機台和機台系列調試、系列啟動及升版期間簡化資料處理，所有 NC 的系統及使用者資料都會分為各個資料層級。

資料層級	資料
S = 系統	西門子提供的系統資料，例如機械資料和設定資料、標準及測量循環、定義 (SGUD) 和巨集 (SMAC) 等。
M = 製造商 （機台製造商）	機台系列專屬調試資料，例如定義機台功能範圍的製造商循環、定義 (MGUD) 和巨集 (MMAC) 及機械資料。
I = 個人（機台專屬）	機台專屬調試資料，例如補正資料參考點偏移量。
U = 使用者	操作機台期間產生的機台專屬資料，例如刀具資料、設定資料、工件程式、使用者循環、定義 (UGUD) 和巨集 (UMAC)。

參考資料：
SINUMERIK 828D 調試手冊，車削及銑削，「簡介及使用資料層級」章節

定義 (DEF) 使用者資料

資料項目的資料層級，會透過定義使用者資料的檔案或目錄資料層級以隱性方式指定。 資料項目的資料層級無法變更。

不過對使用者資料的定義 (DEF) 而言，可針對**資料值**指定與資料項目不同的資料層級。

資料項目資料層級必須套用下列條件：

資料值資料層級優先順序 ≤ 資料項目資料層級優先順序

範例：

定義探針的 GUD 定義應位於資料層級 M（= 製造商），供執行製造商循環時使用。 不過資料項目值應屬於資料層級 I（= 個人），因為機台之間的探針類型可能不同。

MGUD.DEF（資料層級 M）

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

重新定義 (REDEF) 系統資料

系統資料的資料層級可透過重新定義 (REDEF) 變更。 重新定義必須在資料層級 S 或 M 的定義檔案中執行。

使用 ACCESS 檔案時，可能只能以 ACCESS 檔案執行重新定義。

如需機械資料、設定及選項資料的個別資料層級及系統變數，請參閱

- 清單手冊，詳細機械資料說明，參數：「類別」
- 清單手冊，系統變數

2.1.11 可定義與可重新定義屬性之概觀

下列表顯示出可定義的屬性（DEF）與對資料類型重定義的（REDEF）。

系統資料

資料類型	初始值	極限值	實體元件	存取權限	資料層級（僅限 828D）
機械參數	---	---	---	REDEF	REDEF
設定資料	REDEF	---	---	REDEF	---
框架資料	---	---	---	REDEF	---
製程資料	---	---	---	REDEF	---
導螺桿錯誤元件（EEC）	---	---	---	REDEF	---
懸垂補正（CEC）	---	---	---	REDEF	---
象限錯誤補正（ QEC）	---	---	---	REDEF	---
刀庫資料	---	---	---	REDEF	---
刀具資料	---	---	---	REDEF	---
保護區	---	---	---	REDEF	---
刀把，具定向功能	---	---	---	REDEF	---
動態鍊	---	---	---	REDEF	---
3D 保護區	---	---	---	REDEF	---
工作區限制	---	---	---	REDEF	---

使用者資料

資料類型	初始值	極限值	實體元件	存取權限	資料層級
R-參數	REDEF	REDEF	REDEF	REDEF	---
同步動作變數（$AC_...）	REDEF	REDEF	REDEF	REDEF	---

資料類型	初始值	極限值	實體元件	存取權限	資料層級
同步動作 GUD（SYG_...）	REDEF	REDEF	REDEF	REDEF	---
EPS 參數	REDEF	REDEF	REDEF	REDEF	---
刀具資料 OEM	REDEF	REDEF	REDEF	REDEF	---
刀庫資料 OEM	REDEF	REDEF	REDEF	REDEF	---
全域使用者變數（GUD）	DEF/REDEF	DEF	DEF	DEF/REDEF	DEF/REDEF
區域使用者變數 (PUD/LUD)	DEF	DEF	DEF	---	---

另請參見

變數 (頁 23)

2.1.12 陣列變數（DEF，SET，REP）之定義與初始化

使用者變數可定義成 1 維至最多 3 維陣列。

- 1 維： `DEF <資料類型> <變數名稱>[<n>]`
- 2 維： `DEF <資料類型> <變數名稱>[<n>,<m>]`
- 3 維： `DEF <資料類型> <變數名稱>[<n>,<m>,<o>]`

說明

STRING 資料類型使用者變數可被定義成最多 2 維陣列。

資料類型

使用者變數可被定義成供下列資料類型所用的陣列： BOOL，CHAR，INT，REAL，STRING，AXIS，FRAME

指派值給陣列元件

可在下列時間即時將值指派給陣列元件：

- 在陣列定義期間（初始值）
- 在程式執行期間

可藉由下列方式指派值：

- 陣列元件的明確規格
- 作為啟動元件與值清單的規格之陣列元件的明確規格（SET）
- 作為啟動元件、值的規格及其重複之處的頻率之陣列元件的明確規格（REP）

說明

框架資料類型使用者變數，無法被指派為初始值。

語法（DEF）

```
DEF <資料類型> <變數名稱>[<n>,<m>,<o>]
DEF STRING[<字串長度>] <變數名稱>[<n>,<m>]
```

語法（DEF...=SET...）

使用值清單：

- 在定義期間：
  DEF <資料類型> <變數名稱>[<n>,<m>,<o>]=SET（<值 1>，<值 2>，等等）
  等同於：
  DEF <資料類型> <變數名稱>[<n>,<m>,<o>]=（<值 1>，<值 2>，等等）

說明

SET 不需透過值清單來指定初始化。

- 在值指派期間：
  <變數名稱>[<n>,<m>,<o>]=SET（<值 1>，<值 2>，等等）

語法 (DEF...=REP...)

使用具有重複的值

- 在定義期間：
  DEF <資料類型> <變數名稱>[<n>,<m>,<o>]=REP（<值>）
  DEF <資料類型> <變數名稱>[<n>，<m>，<o>]=REP（<值>，<number_array_elements>）
- 在值指派期間：
  <變數名稱>[<n>,<m>,<o>]=REP（<值>）
  DEF <資料類型> <變數名稱>[<n>，<m>，<o>]=REP（<值>，<number_array_elements>）

含義

DEF：	用來定義變數的指令	
<資料類型>：	變數的資料類型	
	值域： • 供系統變數 BOOL，CHAR，INT，REAL，STRING，AXIS • 對於 GUD 或 LUD 變數： BOOL，CHAR，INT，REAL，STRING，AXIS，FRAME	
<字串長度>：	STRING 資料類型的最大字元數	
<變數名稱>：	變數名稱。	
[<n>,<m>,<o>]：	陣列大小或陣列索引	
<n>：	1 維的陣列大小或陣列索引	
	類型：	INT（對於系統變數，以及 AXIS）
	值域：	最大陣列大小：65535 陣列索引： 0 ≤ n ≤ 65534
<m>：	2 維的陣列大小或陣列索引	
	類型：	INT（對於系統變數，以及 AXIS）
	值域：	最大陣列大小：65535 陣列索引： 0 ≤ m ≤ 65534
<o>：	3 維的陣列大小或陣列索引	
	類型：	INT（對於系統變數，以及 AXIS）
	值域：	最大陣列大小：65535 陣列索引： 0 ≤ o ≤ 65534
SET：	使用特定值清單的值指派	
(<值 1>，<值 2> 等)：	值清單	
REP：	使用特定 <值> 指派數值	

<值>：	以 REP 初始化時，陣列元素應被寫入的值。
<number_array_elements>：	以特定 <值> 寫入的陣列元素數目。 下列對於剩餘陣列元件的使用，依時間點而定： ● 在定義陣列時初始化： →零點被寫入至剩餘的陣列元件。 ● 在程式執行期間的指派： →陣列元素的實際值保持不變。 若參數未程式設計，則所有陣列元件都會以 <值> 寫入。 若參數等於零點，則下列應用依時間點而定： ● 在定義陣列時初始化： →所有元素皆被預指派為零 ● 在程式執行期間的指派： →陣列元素的實際值保持不變。

陣列索引

陣列元件的明確順序，例如在使用 SET 或 REP 進行值指派時，因為陣列索引的重複，故為由右至左。

範例： 具有 24 個陣列元件的 3 維陣列之初始化：

```
DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1     1. 陣列元件
  FELD[0,0,1] = 1     2. 陣列元件
  FELD[0,0,2] = 1     3. 陣列元件
  FELD[0,0,3] = 1     4. 陣列元件
  ...
  FELD[0,1,0] = 1     5. 陣列元件
  FELD[0,1,1] = 1     6. 陣列元件
  ...
  FELD[0,2,3] = 1     12. 陣列元件
  FELD[1,0,0] = 1     13. 陣列元件
  FELD[1,0,1] = 1     14. 陣列元件
  ...
  FELD[1,2,3] = 1     24. 陣列元件
```

對應至：

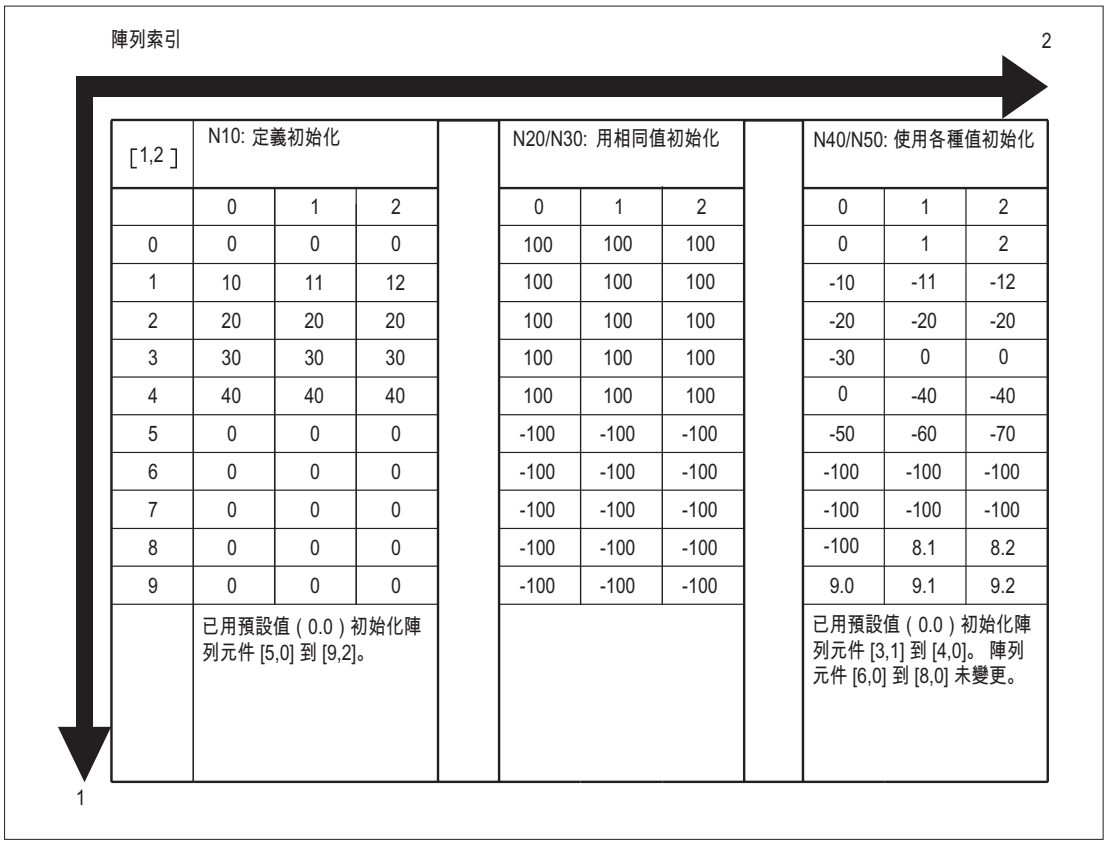
```
FOR n=0 TO 1
  FOR m=0 TO 2
    FOR o=0 TO 3
      FELD[n，m，o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

範例： 初始化完整的變數陣列

對於實際指派情形，請參考圖示。

程式碼

```
N10 DEF REAL FELD1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)
N20 ARRAY1[0,0] = REP(100)
N30 ARRAY1[5,0] = REP(-100)
N40 FELD1[0,0]=SET(0,1,2,-10,-11,-12,-20,-20,-20,-30, , , ,-40,-40,-50,-60,-70)
N50 FELD1[8,1]=SET(8.1,8.2,9.0,9.1,9.2)
```



[1,2]	N10: 定義初始化		
	0	1	2
0	0	0	0
1	10	11	12
2	20	20	20
3	30	30	30
4	40	40	40
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
	已用預設值 (0.0) 初始化陣列元件 [5,0] 到 [9,2]。		

N20/N30: 用相同值初始化		
0	1	2
100	100	100
100	100	100
100	100	100
100	100	100
100	100	100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100

N40/N50: 使用各種值初始化		
0	1	2
0	1	2
-10	-11	-12
-20	-20	-20
-30	0	0
0	-40	-40
-50	-60	-70
-100	-100	-100
-100	-100	-100
-100	8.1	8.2
9.0	9.1	9.2
已用預設值 (0.0) 初始化陣列元件 [3,1] 到 [4,0]。 陣列元件 [6,0] 到 [8,0] 未變更。		

另請參見

陣列變數（DEF，SET，REP）之定義與初始化： 其它資訊 (頁 61)

變數 (頁 23)

2.1.13 陣列變數（DEF，SET，REP）之定義與初始化： 其它資訊

詳細資訊（SET）

在定義期間初始化

- 從第 1 陣列元件開始，因為在值清單中有已程式設計的值清單，所以有許多陣列元件，會以值清單中的值來指派。
- 將值 0 指派至陣列元件，沒有在值清單中明確宣告值（值清單中的空白處）。
- 對於 AXIS 類型的變數，不允許在值清單中有空白處存在。
- 若值清單中包含的值，超過了已定義的陣列元件時，會顯示警報。

在程式執行期間的值指派

在程式設計期間的值指派時，上述的規則係用於定義。 下列選項亦提供支援：

- 算式亦可作為值清單中的元件。
- 值指派會以已程式設計的陣列索引開始。 可以將值選擇性指派到子陣列中。

範例：

程式碼	註解
`DEF INT ARRAY[5,5]`	；陣列定義
`ARRAY[0,0]=SET(1,2,3,4,5)`	；值指派至前 5 個陣列元件 [0,0] - [0,4]
`FELD[0,0]=SET(1,2, , ,5)`	；具有空白處的值指派至前 5 個陣列元件 [0,0] - [0,4]，陣列元件 [0,2] 及 [0,3] = 0
`ARRAY[2,3]=SET(VARIABLE,4*5.6)`	；具有變數的值指派和在陣列索引 [2,3] 啟動的算式： [2,3] = VARIABLE [2,4] = 4 * 5.6 = 22.4

詳細資訊（REP）

在定義期間初始化

- 所有或可選擇的特定數量的陣列元件，以特定值（常數）初始化。
- 無法初始化 FRAME 類型的變數。

範例：

程式碼	註解
`DEF REAL varName[10]=REP(3.5,4)`	；初始化陣列定義與陣列元件 [0] 至 [3] 具有值 3.5。

在程式執行期間的值指派

在程式設計期間的值指派時，上述的規則係用於定義。下列選項亦提供支援：

- 算式亦可作為值清單中的元件。
- 值指派會以已程式設計的陣列索引開始。 可以將值選擇性指派到子陣列中。

範例：

程式碼	註解
`DEF REAL varName[10]`	；陣列定義
`varName[5]=REP(4.5,3)`	；陣列元件元 [5] 至 [7] = 4.5
`R10=REP(2.4,3)`	；R 參數 R10 至 R12 = 2.4
`DEF FRAME FRM[10]`	；陣列定義
`FRM[5] = REP(CTRANS (X,5))`	；陣列元件 [5] 至 [9] = CTRANS(X,5)

另請參見

陣列變數（DEF，SET，REP）之定義與初始化 (頁 56)

2.1.14 資料類型

NC 有下列可用的資料類型：

資料類型	含義	值域
INT	含正負號之整數	-2147483648 ... +2147483647
REAL	實數（LONG REAL 至 IEEE）	±(~2,2*10^{-308} … ~1,8*10^{+308})
BOOL	真值 TRUE (1) 而 FALSE (0)	1, 0
CHAR	ASCII 字元	ASCII 代碼 0 至 255
STRING	已定義長度的字元字串	最多 200 個字元（無特殊字元）
AXIS	軸 / 主軸識別碼	通道軸識別碼
FRAME	用於靜態座標轉換的幾何參數（轉譯，旋轉，刻度，鏡像）	---

明確的資料類型轉換

下列的資料類型轉換是可行的，且會在指派和參數轉換期間明確執行：

從 ↓ / 至 →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x
x： 可行，無限制 o：因為值域過衝的可能的資料損失 ⇒ 警報； 倒圓角： 小數點位數值 ≥ 0.5 ⇒ 將進位，小數點位數值 < 0.5 ⇒ 將捨去 &: 值 ≠ 0 ⇒ TRUE，值== 0 ⇒ FALSE			

另請參見

變數 (頁 23)

2.1.15 檢查變數 (ISVAR) 的可用性

預先定義的 ISVAR 函數可用於檢查系統 / 使用者變數（例如機械資料、設定資料、系統變數、如 GUD 等一般變數）是否為 NC 中已知的變數。

變數

質問的變數之結構如下：

無維度變數：	**<變數>**
無陣列索引的一維變數：	**<變數>[ ]**
有陣列索引 n 的一維變數：	**<變數>[<n>]**
無陣列索引的二維變數：	**<變數>[,]**
有陣列索引 n 及 m 的二維變數：	**<變數>[<n>,<m>]**

語法

```
<結果>=ISVAR(<變數>[<n>,<m>])
```

含義

<結果>：	傳回值		
	資料類型：	BOOL	
	值域：	1	可用變數
		0	未知變數
ISVAR：	檢查指定的系統 / 使用者變數是否為 NC 中已知的變數。		
<變數>：	系統 / 使用者變數的名稱		
	資料類型：	STRING	
<n>：	第一維度的陣列索引（**選配**）		
	資料類型：	INT	
<m>：	第二維度的陣列索引（**選配**）		
	資料類型：	INT	

根據傳輸參數進行下列檢查：

- 名稱是否已知？
- 變數是否為陣列？

- 是否為一維或二維陣列？
- 個別的陣列索引是否在可允許範圍內？

只有在所有檢查為正值時，傳回 TRUE (1)。

若檢查為負值或發生語法錯誤，傳回 FALSE (0)。

範例

程式碼	註解
DEF INT VAR1	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VAR1")	;此情況下 IS_VAR 為 TRUE。

程式碼	註解
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VARARRAY[,]")	;此情況下 IS_VAR 為 TRUE，是二維陣列。
N20 IS_VAR=ISVAR("VARARRAY")	;IS_VAR 為 TRUE，變數存在。
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	;IS_VAR 為 FALSE，不允許陣列索引。
N40 IS_VAR=ISVAR("VARARRAY[8,8")	;IS_VAR 為 FALSE，遺失「]」（語法錯誤）。
N50 IS_VAR=ISVAR("VARARRAY[,8]")	;IS_VAR 為 TRUE，允許陣列索引。
N60 IS_VAR=ISVAR("VARARRAY[8,]")	;IS_VAR 為 TRUE，允許陣列索引。

程式碼	註解
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"	;傳輸參數是一種機械資料項目，IS_VAR 為 TRUE。

程式碼	註解
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("$P_EP")	;此情況下 IS_VAR 為 TRUE。
N20 IS_VAR=ISVAR("$P_EP[X]")	;此情況下 IS_VAR 為 TRUE。

2.1.16 讀取屬性值 / 資料類型（GETVARPHU、GETVARAP、GETVARLIM、GETVARDIM、GETVARDFT、GETVARTYP）

系統 / 使用者變數的屬性值可利用預先定義的 GETVARPHU、GETVARAP、GETVARLIM 和 GETVARDFT 函數讀取，系統 / 使用者變數的資料類型則利用 GETVARTYP 讀取。

讀取物理單位

語法：
<結果>=GETVARPHU（<名稱>）

含義：

<結果>：	物理單位的數值		
	資料類型：	INT	
	值域：	請參閱 "屬性： 實體元件（PHU） (頁 46)" 的表格	
		錯誤情況	
		- 2	指定的 <名稱> 尚未指派至系統參數或使用者變數。
GETVARPHU：	讀取系統 / 使用者變數的物理單位		
<名稱>：	系統 / 使用者變數的名稱		
	資料類型：	STRING	

範例：

NC 包含下列 GUD 變數：

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

程式碼	註解
DEF INT result=0	
result=GETVARPHU("electric"	；決定 GUD 變數的物理單位。
)	
IF (result < 0) GOTOF error	

值 42 為回傳的結果，對應於物理單位 [kW]。

說明

例如，指派變數 a = b 時，GETVARPHU 可用於檢查兩個變數是否具有預期的物理單位。

讀取存取權限

語法：
<結果>=GETVARAP（<名稱>，<存取>）

含義：

<結果>：	特定 **<**存取**>** 的保護等級		
	資料類型：	INT	
	值域：	0 ... 7	請參閱「屬性： 存取權力（APR、APW、APRP、APWP、APRB、APWB） (頁 49)」。
		錯誤情況	
		-1	無法寫入（僅與 **<**存取**>** "WP" 及 "WB" 相關）
		- 2	指定的 <名稱> 尚未指派至系統參數或使用者變數。
		- 3	不正確的 <存取> 值
GETVARAP：	讀取系統 / 使用者變數的存取權限		
<名稱>：	系統 / 使用者變數的名稱		
	資料類型：	STRING	
<存取>：	存取類型		
	資料類型：	STRING	
	值域：	"RP"	經由工件程式讀取
		"WP"	經由工件程式寫入
		"RB"	經由 OPI 讀取
		"WB"	經由 OPI 寫入

範例：

程式碼	註解
`DEF INT result=0`	
`result=GETVARAP("$TC_MAP8","WB")`	；決定關於透過 OPI 寫入系統參數「刀庫定位」的存取保護。
`IF (result < 0) GOTOF error`	

值 7 為回傳的結果，對應於按鍵開關位置 0（= 無存取保護）。

說明

例如，GETVARAP 可作為檢查程式，用於檢查應用程式預期的存取權限。

讀取臨界值

語法：
<狀態>=GETVARLIM（<名稱>，<臨界值>，<結果>）

含義：

<狀態>：	函數狀態		
	資料類型：	INT	
	值域：	1	沒有問題
		-1	無定義的臨界值 （類型 AXIS、STRING、FRAME 的變數）
		-2	指定的 <名稱> 尚未指派至系統參數或使用者變數。
		-3	不正確的 <臨界值> 值
GETVARLIM：	讀取系統 / 使用者變數之上 / 下臨界值		
<名稱>：	系統 / 使用者變數的名稱		
	資料類型：	STRING	
<臨界值>：	指定應讀出的臨界值		
	資料類型：	CHAR	
	值域：	「L」：	= 下臨界值
		"U"：	= 上臨界值
<結果>：	回傳臨界值		
	資料類型：	VAR REAL	

範例：

程式碼	註解
DEF INT state=0	
DEF REAL result=0	
state=GETVARLIM("$MA_MAX_AX_VELO","L",result)	定義 MD32000 $MA_MAX_AX_VELO 的下臨界值。
IF (result < 0) GOTOF error	

讀取屬性 / 資料類型

語法：
<結果>=GETVARDIM(<名稱>, 索引)

意義：

<結果>：	陣列的維度 / 編號 **<**索引**>**	
	資料類型：	INT
GETVARDIM：	讀取系統 / 使用者變數之上 / 下臨界值	
<名稱>：	讀取陣列元素的編號	
	資料類型：	STRING
<索引>：	陣列的編號，最大 3。	
	資料類型：	INT

範例：

程式碼	備註
N5 DEF REAL myReal[5,4]	
N10 R1 = GETVATDIM("myReal",1)	R1 = 5
N15 R2 = GETVATDIM("myReal",2)	R2 = 4

讀取預設值

語法：
<狀態>=GETVARDFT（<名稱>，<結果>[，<索引_1>，<索引_2>，<索引_3>]）

含義：

<狀態>：	函數狀態		
	資料類型：	INT	
	值域：	1	沒有問題
		-1	無可用的預設值 （例如，因為 **<**結果**>** 包含錯誤的 **<**名稱**>** 類型）
		-2	指定的 <名稱> 尚未指派至系統參數或使用者變數。
		-3	不正確的 <索引_1> 值，維度小於一（**=** 無陣列 **=** 純量）
		-4	不正確的 <索引_2> 值
		-5	不正確的 <索引_3> 值

GETVARDFT：	讀取系統 / 使用者變數的預設值	
<名稱>：	系統 / 使用者變數的名稱	
	資料類型：	STRING
<結果>：	回傳預設值	
	資料類型：	VAR REAL （當讀取類型 INT、REAL、BOOL、AXIS 的變數預設值時）
		VAR STRING （當讀取類型 STRING 和 CHAR 的變數預設值時）
		VAR FRAME （當讀取類型 FRAME 的變數預設值時）
<索引_1>：	第一維度的索引（選用）	
	資料類型：	INT
	未程式設計表示 = 0	
<索引_2>：	第二維度的索引（選用）	
	資料類型：	INT
	未程式設計表示 = 0	
<索引_3>：	第三維度的索引（選用）	
	資料類型：	INT
	未程式設計表示 = 0	

範例：

程式碼	註解
`DEF INT state=0`	
`DEF REAL resultR=0`	；接受 INT、REAL、BOOL、AXIS 類型之預設值的變數
`DEF FRAME resultF=0`	；接受 FRAME 類型之預設值的變數
`IF (GETVARTYP("$MA_MAX_AX_VELO") <> 4) GOTOF error`	
`state=GETVARDFT("$MA_MAX_AX_VELO", resultR, AXTOINT(X))`	；決定「X」軸的預設值。
`IF (result < 0) GOTOF error`	
`IF (GETVARTYP("$TC_TP8") <> 3) GOTOF error`	
`state=GETVARDFT("$TC_TP8", resultR)`	
`IF (GETVARTYP("$P_UBFR") <> 7) GOTOF error`	

程式碼	註解
`state=GETVARDFT("$P_UBFR", resultF )`	

讀取資料類型

語法：
<結果>=GETVARTYP（<名稱>）

含義：

<結果>：	指定系統 / 使用者變數的資料類型		
	資料類型：	INT	
	值域：	1	= 布林值 (BOOL)
		2	= 字元 (CHAR)
		3	= 整數 (INT)
		4	= 實數 (REAL)
		5	= 字串 (STRING)
		6	= 軸 (AXIS)
		7	= 框架 (FRAME)
		錯誤情況	
		< 0	指定的 <名稱> 尚未指派至系統參數或使用者變數。
GETVARTYP：	讀取系統 / 使用者變數的資料類型		
<名稱>：	系統 / 使用者變數的名稱		
	資料類型：	STRING	

範例：

程式碼	註解
`DEF INT result=0`	
`DEF STRING name="R"`	
`result=GETVARTYP(name)`	；決定 R 參數的類型。
`IF (result < 0) GOTOF error`	

值 4 為回傳的結果，對應於 REAL 資料類型。

2.2 間接程式設計

2.2.1 間接程式設計位址

間接程式設計位址時，會以適當類型的變數替換延伸位址（<索引>）。

說明

無法間接程式設計位址供以下使用：

- N（單節號碼）
- L（子程式）
- 可設定的位址
  （例如 X[1]不可以取代 X1）

語法

```
<ADDRESS>[<索引>]
```

含義

<ADDRESS>[...]：	具有副檔名（索引）的固定位址
<索引>：	變數，例如，主軸號碼、座標軸，....

範例

範例 1： 間接程式設計一個主軸號碼

直接程式設計：

程式碼	註解
S1=300	；1 號主軸的 rpm 速度。

間接程式設計：

程式碼	註解
DEF INT SPINU=1	；定義變數，類型 INT，以及值指派。
S[SPINU]=300	；主軸速度為 300 rpm，該數值儲存在 SPINU 變數（在此範例 1 中，採用的是 1 號主軸）。

範例 2： 間接程式設計軸

直接程式設計：

程式碼	註解
FA[U]=300	；軸「U」的進給率為 300。

間接程式設計：

程式碼	註解
DEF AXIS AXVAR2=U	；定義變數，類型 AXIS，以及值指派。
FA[AXVAR2]=300	；將進給率 300 指定給軸；該軸係以 AXVAR2 為位址名稱，儲存在變數中。

範例 3： 間接程式設計軸

直接程式設計：

程式碼	註解
$AA_MM[X]	；讀取軸「X」的探針測量值 (MCS)。

間接程式設計：

程式碼	註解
DEF AXIS AXVAR3=X	；定義變數，類型 AXIS，以及值指派。
$AA_MM[AXVAR3]	；讀取探針測量值 (MCS) ，其名稱係儲存在變數 AXVAR3 中。

範例 4： 間接程式設計軸

直接程式設計：

程式碼
X1=100 X2=200

間接程式設計：

程式碼	註解
DEF AXIS AXVAR1 AXVAR2	；定義兩個 AXIS 類型的變數。
AXVAR1=（X1） AXVAR2=（X2）	；指派軸名稱。
AX[AXVAR1]=100 AX[AXVAR2]=200	；移動軸，該軸係以 AXVAR1 和 AXVAR2 作為位址名稱，儲存在變數中

範例 5： 間接程式設計軸

直接程式設計：

程式碼
G2 X100 I20

間接程式設計：

程式碼	註解
DEF AXIS AXVAR1=X	；定義變數，類型 AXIS，以及值指派。
G2 X100 IP[AXVAR1]=20	；對軸進行間接程式設計中心點資料，該軸係以 AXVAR1 作為位址名稱，儲存於變數中。

範例 6： 間接程式設計陣列元件

直接程式設計：

程式碼	註解
DEF INT ARRAY1[4,5]	；定義陣列 1

間接程式設計：

程式碼	註解
DEFINE DIM1 AS 4	；對於陣列維度，陣列大小必須指派為固定值。
DEFINE DIM2 AS 5	
DEF INT ARRAY[DIM1，DIM2]	
ARRAY[DIM1-1，DIM2-1]=5	

範例 7：間接子程式呼叫

程式碼	註解
CALL "L" << R10	；呼叫程式，該程式之號碼位於 R10（字串串連）。

2.2.2 間接程式設計 G 指令

間接程式設計 G 指令，允許對循環進行有效的程式設計。

語法

G[<群組>]=<編號>

含義

G[...]：	具有副檔名（索引）的 G 指令	
<群組>：	索引參數：G 群組	
	類型：	INT

<編號>：	供 G 指令數字所用的變數	
	類型：	INT 或 REAL

說明

一般而言，僅不決定語法的 G 指令可以間接程式設計。

決定語法的 G 指令中，僅 G 群組 1 可以間接程式設計。
決定語法的 G 指令中，G 群組 2、3 和 4 無法間接程式設計。

說明

進行間接 G 指令程式設計時，不可使用算術函數。若要計算 G 指令數字，則必須在進行間接 G 指令程式設計之前，在分開的工件程式行先行計算完成。

範例

範例 1：可調整的工作偏移量（G 群組 8）

程式碼	註解
`N1010 DEF INT INT_VAR`	
`N1020 INT_VAR=2`	
`...`	
`N1090 G[8]=INT_VAR G1 X0 Y0`	`;G54`
`N1100 INT_VAR=INT_VAR+1`	`;G 指令計算`
`N1110 G[8]=INT_VAR G1 X0 Y0`	`;G55`

範例 2：層級選取（G 群組 6）

程式碼	註解
`N2010 R10=$P_GG[6]`	`;讀取 G 群組 6 的生效 G 指令`
`...`	
`N2090 G[6]=R10`	

參考資料

若需 G 群組的資訊，請參閱：
基礎程式設計手冊；「G 群組」章節

2.2.3 間接程式設計定位屬性 (GP)

定位屬性（例如，座標軸定位的增量或絕對程式設計）可以連同關鍵字組 GP 間接編寫為變數。

應用

定位屬性的間接程式設計係用於**替換循環**，在此情況中，程式設計定位屬性為關鍵字組（例如 IC，AC，...），會有下列優點：

由於間接程式設計為變數，因此**不需要** CASE 陳述式，否則將為所有可能的定位屬性分支。

語法

```
<定位指令>[<軸 / 主軸>]=
GP（<位置>，<定位屬性>）
<軸 / 主軸>=BP（<位置>，<定位屬性>）
```

含義

<定位指令>[]：	下列定位指令可以和關鍵字組 GP 一起編寫： POS，POSA，SPOS，SPOSA 也有可能： ● 所有的座標軸和主軸識別碼在通道中出現： <軸 / 主軸> ● 變數座標軸 / 主軸識別碼 AX
<軸 / 主軸>：	待定位的座標軸 / 主軸
GP()：	用以定位的關鍵字
<位置>：	參數 1 座標軸 / 主軸定位為常數或變數
<定位屬性>：	參數 2 定位屬性（例如，定位逼近模式為變數（例如，$P_SUB_SPOSMODE）或為關鍵字組 IC，AC，...）

由變數所提供的值，具有下列特性：

值	含義	允許給：
0	定位屬性保持不變	
1	AC	POS，POSA，SPOS，SPOSA，AX，座標軸位址
2	IC	POS，POSA，SPOS，SPOSA，AX，座標軸位址
3	DC	POS，POSA，SPOS，SPOSA，AX，座標軸位址
4	ACP	POS，POSA，SPOS，SPOSA，AX，座標軸位址
5	ACN	POS，POSA，SPOS，SPOSA，AX，座標軸位址
6	OC	-
7	PC	-
8	DAC	POS，POSA，AX， 座標軸位址
9	DIC	POS，POSA，AX， 座標軸位址
10	RAC	POS，POSA，AX， 座標軸位址
11	RIC	POS，POSA，AX， 座標軸位址
12	CAC	POS，POSA
13	CIC	POS，POSA
14	CDC	POS，POSA
15	CACP	POS，POSA
16	CACN	POS，POSA

範例

對於介於螺距主軸 S1 和後續主軸 S2 之間的主動同步主軸耦合，會使用主程式中的 SPOS 指令，呼叫後續的替換循環，來定位主軸。

定位是使用 N2230 中的陳述式來完成：
```
SPOS[1]=GP（$P_SUB_SPOSIT，$P_SUB_SPOSMODE） SPOS[2]=GP
（$P_SUB_SPOSIT，$P_SUB_SPOSMODE）
```

待逼近的定位，會由從系統變數$P_SUB_SPOSIT 中來讀取；而定位逼近模式，則是從系統變數$P_SUB_SPOSMODE 中讀取。

程式碼	註解
N1000 PROC LANG_SUB DISPLOF SBLOF	
...	
N2100 IF($P_SUB_AXFCT==2)	
N2110	；SPOS / SPOSA / M19 指令的替換，供主動同步主軸耦合所用
N2185 DELAYFSTON	；停止延遲區域起點
N2190 COUPOF(S2,S1)	；停用同步主軸耦合
N2200	；定位螺距與後續主軸
N2210 IF($P_SUB_SPOS==TRUE) OR ($P_SUB_SPOSA==TRUE)	
N2220	；以 SPOS 定位主軸：
N2230 SPOS[1]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)	
SPOS[2]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)	
N2250 ELSE	
N2260	；使用 M19 定位主軸：
N2270 M1=19 M2=19	；定位螺距與後續主軸
N2280 ENDIF	
N2285 DELAYFSTOF	；停止延遲區域結尾
N2290 COUPON(S2,S1)	；啟動同步主軸耦合
N2410 ELSE	
N2420	；查詢進一步替換
...	
N3300 ENDIF	
...	
N9999 RET	

補充條件

- 定位屬性的間接程式設計，無法在同步動作中進行。

參考資料

功能手冊基本功能；BAG，通道，程式操作，重置回應 (K1)，
章節： 藉由子程式替換 NC 函數一章

2.2.4 間接程式設計工件程式行（EXECSTRING）

使用工件程式指令 EXECSTRING 時，可將先前產生的字串變數視為工件程式行並加以執行。

語法

EXECSTRING 編寫於分開的工件程式行中：
EXECSTRING（<字串變數>）

含義

EXECSTRING：	將字串變數視為工件程式行加以執行的指令
<字串變數>：	字串類型的變數，其中包含待執行的工件程式行

說明

使用 EXECSTRING 時，可抽取所有能夠於**工件程式程式區段**編寫的工件程式句（控制結構 (頁 124)除外）。 這表示 PROC 與 DEF 陳述式也會和一般的使用一樣，被排除在 INI 與 DEF 檔案之外。

範例

程式碼	註解
N100 DEF STRING[100] MY_BLOCK	；定義用來接受待執行工件程式行的字串變數。
N110 DEF STRING[10] MFCT1="M7"	
...	
N200 EXECSTRING(MFCT1 << "M4711")	；執行工件程式行「M7 M4711」。
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF（R10）	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(MY_BLOCK)	；執行工件程式行「M3 M7」。

2.3 運算功能

運算子 / 算術函數	含義
+	加法
–	減法
*	乘法
/ [1)]	除法 [1)]
DIV [1)]	整數除法 [1)]
MOD [1)]	模數除法（提供剩餘的整數除法）[1)]
:	FRAME 變數的串鍊運算子
SIN( )	正弦
COS()	餘弦
TAN()	正切
ASIN()	反正弦
ACOS()	反餘弦
ATAN2(,) [1)]	反正切 2 [1)]
SQRT()	平方根
ABS()	絕對值
POT()	二次方（平方）
TRUNC()	整數元件 比較指令的精確度，可用 TRUNC（參閱"比較錯誤（TRUNC）的精確性修正 (頁 85)"）來設定
ROUND()	標準整數法
LN()	自然對數
EXP()	指數功能
MINVAL ()	兩變數的較小值 （請參閱"變數最小，最大與範圍（MINVAL，MAXVAL 與 BOUND） (頁 87)"）
MAXVAL ()	兩變數的較大值 （請參閱"變數最小，最大與範圍（MINVAL，MAXVAL 與 BOUND） (頁 87)"）

BOUND ()	在所定義值域中的變數值 （請參閱"變數最小，最大與範圍（MINVAL，MAXVAL 與 BOUND） (頁 87)"）
CTRANS()	偏移量
CROT ()	旋轉
CSCALE()	變更比例
CMIRROR()	鏡射
1) 請參閱「範例」段落	

程式設計

一般的算術標記法可用於算術函數。執行的優先順序係由圓括號來表示。角度係用於三角函數及其反函數（直角 = 90°）。

範例

除法：/

（類型 REAL）=（類型 INT 或類型 REAL）/（類型 INT 或類型 REAL）；

範例：3 / 4 = 0.75

整數除法：DIV

（類型 INT）=（類型 INT 或 REAL）/（類型 INT 或 REAL）；

範例：7 DIV 4.1 = 1

模數除法（提供整數除法的餘數）：MOD

（類型 REAL）=（類型 INT 或 REAL）MOD（類型 INT 或 REAL）；

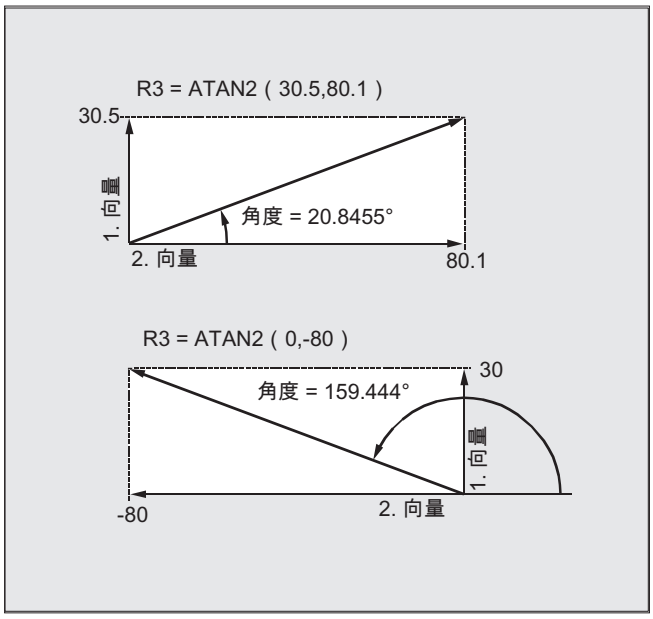
範例：7 MOD 4.1 = 2.9

反正切 2：ATAN2

算術函數 ATAN2 會計算兩個互相垂直的向量之間的總向量角度。

結果會落在四個象限之一（–180° < 0 < +180°）。

角度的參考會根據正向的第二個值。



程式設計範例

程式碼	註解
R1=R1+1	；新 R1 = 舊 R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN（25.3）	
R14=R1*R2+R3	；先乘除後加減。
R14=（R1+R2）*R3	；運算式及圓括號先運算。
R15=SQRT（POT（R1）+POT（R2））	；內部圓括號先運算： R15 = ((R1^2 + R2^2)) 平方根
RESFRAME=FRAME1：FRAME2	；以串鍊運算子進行 FRAME 邏輯運算
FRAME3=CTRANS（...）：CROT（...）	在 FRAME 元件指派值

2.4 比較與邏輯運算

舉例來說，**比較運算**可以用來制訂跳躍條件。 也可比較複雜運算式。

比較運算適用於變數類型 CHAR，INT，REAL 與 BOOL。 代碼值會與 CHAR 類型做比較。對於 STRING，AXIS 與 FRAME 等類型，下列為可行的運算： **==** 與 **<>**，可用於 STRING 類型的運算，即使在同步動作中亦然。

比較運算的結果永遠會是 BOOL 類型。

邏輯運算子係用於連結真值。

邏輯運算僅能用於 BOOL 類型的變數。 然而，透過內部類型轉換，他們還可用於 CHAR，INT 與 REAL 等資料類型。

對於邏輯（布林）運算，下列可用於 BOOL、CHAR、INT 與 REAL 等資料類型：

- 0 對應至： FALSE
- 不等於 0 即表示： TRUE

位元邏輯運算子

邏輯運算也能用於 CHAR 與 INT 類型的單位元。 類型轉換是自動進行的。

程式設計

關係運算子	含義
==	等於
<>	不等於
>	大於
<	小於
>=	大於或等於
<=	小於或等於

邏輯運算子	含義
AND	AND
OR	OR
NOT	否
XOR	排除 OR

位元邏輯運算子	含義
B_AND	逐一位元 AND
B_OR	逐一位元 OR
B_NOT	逐一位元否定
B_XOR	逐一位元排除 OR

說明

在算術運算式中，可藉由圓括號，來指定各運算子的執行順序，藉此手動超調一般的優先順序規則。

說明

在 BOOLEAN 運算元和運算子中間必須空一格。

說明

運算子 B_NOT，僅參照一個運算元。 位於運算子之後。

範例

範例 1： 比較運算子

```
IF R10>=100 GOTOF DEST
```

或

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

R10>=100 的比較結果，會先暫存在 R11 之中。

範例 2： 邏輯運算子

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

或

```
IF NOT R10 GOTOB START
```

NOT，僅參照一個運算元。

範例 3： 位元邏輯運算子

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

2.5 比較錯誤（TRUNC）的精確性修正

TRUNC 指令會去除運算元與精確係數相乘後的尾數。

用於比較指令的可設定的精確性

REAL 類型的程式資料會在 IEEE 格式中，以 64 位元的方式於內部顯示。 這種顯示格式會造成顯示出來的小數位不夠精確，並會導致在與理想計算值進行比較時，出現無法預期的結果。

相對等式

為了防止這種因為顯示格式所導致的不精確情形，干擾了程式流程，比較指令會檢查相對等式，而非絕對等式。

語法

比較錯誤的精確性修正

```
TRUNC （R1*1000）
```

含義

TRUNC：	去掉小數點位數

相對等式 10^{-12} 會考慮到

- 等式： **(==)**
- 不等式： **(<>)**
- 大於或等於： **(>=)**
- 小於或等於： **(<=)**
- 大於或小於： **(><)** 絕對等式
- 大於： **(>)**
- 小於： **(<)**

相容性

為求一致性，檢查相對等式 (>) 和 (<)，可以透過設定機械資料 MD10280 $MN_PROG_FUNCTION_MASK Bit0 = 1 來停用。

說明

與 REAL 類型的資料做比較，會因為上述原因而造成一定程度的不精確。 若誤差超過了接受範圍，可藉由將一個精確係數與運算元相乘，使用 INTEGER 計算，然後再用 TRUNC 去掉尾數。

同步動作

定義比較指令的回應，也可套用至同步動作。

範例

範例 1： 精確性考量

程式碼	註解
`N40 R1=61.01 R2=61.02 R3=0.01`	；指派初始值
`N41 IF ABS(R2-R1) > R3 GOTOF ERROR`	；跳躍一直執行
`N42 M30`	；程式結束
`N43 ERROR： SETAL(66000)`	
`R1=61.01 R2=61.02 R3=0.01`	；指派初始值
`R11=TRUNC(R1*1000) R12=TRUNC(R2*1000) R13=TRUNC(R3*1000)`	；精確度校正
`IF ABS(R12-R11) > R13 GOTOF ERROR`	；不再執行跳躍
`M30`	；程式結束
`ERROR:SETAL(66000)`	

範例 2： 計算並求出兩個運算元的商數

程式碼	註解
`R1=61.01 R2=61.02 R3=0.01`	；指派初始值
`IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF ERROR`	；未執行跳躍
`M30`	；程式結束
`ERROR:SETAL(66000)`	

2.6 變數最小，最大與範圍（MINVAL，MAXVAL 與 BOUND）

MINVAL 及 MAXVAL 指令可用於比較兩個變數的值。 較小的值（在 MINVAL）的情形中，或較大的值（在 MAXVAL 的情形中）會各自被傳送。

BOUND 指令可測試變數的值是否落在已定義的值域中。

語法

<較小值>=MINVAL（<變數 1>，<變數 2>）
<較大值>=MAXVAL（<變數 1>，<變數 2>）
<傳回值>=<BOUND>（<最小>，<最大>，<測試變數>）

含義

MINVAL：	取得二變數的**較小**值（<變數 1>，<變數 2>）
<較小值>：	用於 MINVAL 指令的結果變數 設定較小的變數值。
MAXVAL：	取得二變數的**較大**值（<變數 1>，<變數 2>）
<較大值>：	用於 MAXVAL 指令的結果變數 設定較大的變數值。
BOUND：	測試（<測試變數）這個變數是否在已定義的值域之中。
<最小>：	定義了值域中最小值的變數。
<最大>：	定義了值域中最大值的變數。
<傳回值>：	用於 BOUND 指令的結果變數 若測試變數的值在已定義的值域之中，結果變數會設定為測試變數的值。 若測試變數的值大於最大值，結果變數會設定為定義域中的最大值。 若測試變數的值小於最小值，結果變數會設定為定義域中的最小值。

說明

MINVAL，MAXVAL，與 BOUND 也可同步程式設計。

說明

若值相等時的行為

若值相等，MINVAL/MAXVAL 會設定此相等值。 在 BOUND 的情形中，要測試的變數值會再次傳回。

範例

程式碼	註解
DEF REAL rVar1 = 10.5，rVar2 = 33.7，rVar3，rVar4，rVar5，rValMin，rValMax，rRetVar	
rValMin=MINVAL（rVar1，rVar2）	；rValMin 設定為值 10.5。
rValMax=MAXVAL（rVar1，rVar2）	；rValMax 設定為值 33.7。
rVar3=19.7	
rRetVar=BOUND（rVar1，rVar2，rVar3）	；rVar3 在臨界值中，rRetVar 設定為 19.7。
rVar3=1.8	
rRetVar=BOUND（rVar1，rVar2，rVar3）	；rVar3 低於最小臨界值，rRetVar 設定為 10.5。
rVar3=45.2	
rRetVar=BOUND（rVar1，rVar2，rVar3）	；rVar3 高於最大臨界值，rRetVar 設定為 33.7。

2.7 運算優先順序

每個運算子都已分配優先順序。 當運算式已求出，具有最高優先權的運算子，永遠都要優先套用。. 若運算子具有同等的優先順序，演算時必須由左至右。

在算術運算式中，可藉由圓括號，來指定各運算子的執行順序，藉此手動超調一般的優先順序規則。

運算子的順序

優先順序由高至低

1.	NOT，B_NOT	否定，逐一位元否定
2.	*，/，DIV，MOD	乘法，除法
3.	+, –	加法，減法
4.	B_AND	逐一位元 AND
5.	B_XOR	逐一位元排除 OR
6.	B_OR	逐一位元 OR
7.	AND	AND
8.	XOR	排除 OR
9.	OR	OR
10.	<<	串鍊字串，結果 STRING 類型
11.	==, <>, >, <, >=, <=	比較運算子

說明

串鍊運算子「：」用於框架，不可在相同的運算式中做為其他運算子用。 因此這種運算子不需要具有優先順序層級。

範例： IF 敘述

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

2.8 可行類型轉換

功能

指派時的類型轉換

常數值，變數，或指派給變數的運算式，必須要有相容的變數類型。 在這種情形中，當值被指派時，會自動轉換類型。

可行類型轉換

至	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
從							
REAL	是	是*	是 1)	是*	–	–	–
INT	是	是	是 1)	是 2)	–	–	–
BOOL	是	是	是	是	是	–	–
CHAR	是	是	是 1)	是	是	–	–
STRING	–	–	是 4)	是 3)	是	–	–
AXIS	–	–	–	–	–	是	–
FRAME	–	–	–	–	–	–	是

說明

* 從 REAL 轉換至 INT 類型時，>=0.5 的小數值會被進位，若小於則會被捨去（參見 ROUND 函數）。

1) 值 <> 0 為 TRUE；值 == 0 為 FALSE

2) 若值在許可範圍

3) 若只有一個字元

4) 字串長度 0 = >FALSE，其餘為 TRUE

說明

若轉換產生的值，大於目標範圍，則會輸出錯誤訊息。

若在運算式中發生混合類型的情形，則會自動進行類型轉換。 同步動作時也可進行類型轉換，請參考“動作同步動作，固有類型轉換”一章。

2.9 字串運算

字串運算

除了典型的運算“指派”與“比較”以外，下列為可用的字串運算：

- 類型轉換為 STRING（AXSTRING）(頁 91)
- 從 STRING 類型轉換（NUMBER、ISNUMBER、AXNAME）(頁 92)
- 串鍊字串（<<）(頁 94)
- 轉換為大小寫字母（TOLOWER、TOUPPER）(頁 95)
- 決定字串長度（STRLEN）(頁 96)
- 在字串中搜尋字元 / 字串（INDEX、RINDEX、MINDEX、MATCH）(頁 96)
- 選擇子字串（SUBSTR）(頁 97)
- 個別字元的讀取和寫入 (頁 98)
- 格式化字串 (SPRINT) (頁 100)

0 字元的特殊意涵

在內部，0 字元會解讀為字串辨識碼的終點。 若某字元被替換成 0 字元，則該字串會被去掉尾數。

範例：

程式碼	註解
`DEF STRING[20] STRG="axis . stationary"`	
`STRG[6]="X"`	
`MSG（STRG）`	；產生訊息「座標軸 X 固定」。
`STRG[6]=0`	
`MSG（STRG）`	；產生訊息「軸」。

2.9.1 類型轉換為 STRING（AXSTRING）

使用函數「type conversion to STRING」，不同類型的變數便可用來作為訊息 (MSG) 元件。

使用 << 運算子時，可供 INT、REAL、CHAR 及 BOOL 等資料類型內部通用（請參閱「串鍊字串（<<）(頁 94)」）。

一個 INT 值將會轉換成為一般可讀取的格式。 REAL 值會轉換成最多具有 10 個小數點位數。

使用 AXSTRING 指令，可將類型 AXIS 變數轉換成為 STRING。

語法

```
<STRING_RES> = << <any_type>
<STRING_RES> = AXSTRING(<軸識別碼>)
```

含義

<STRING_RES>：	供類型轉換結果所用的變數	
	類型：	STRING
<any_type>：	變數類型 INT，REAL，CHAR，STRING 與 BOOL	
AXSTRING：	AXSTRING 指令提供了特定的座標軸識別碼作為字串。	
<軸識別碼>：	作為座標軸識別碼的變數	
	類型：	AXIS

說明

FRAME 變數無法轉換。

2.9.2 從 STRING 類型轉換（NUMBER、ISNUMBER、AXNAME）

使用 NUMBER 指令，可將 STRING 類型轉換成 REAL。 使用 ISNUMBER 指令，可檢查轉換的能力。

使用 AXNAME 指令，可將字串轉換成座標軸資料類型。

語法

```
<REAL_RES>=NUMBER("<字串>")
<BOOL_RES>=ISNUMBER("<字串>")
<AXIS_RES>=AXNAME("<字串>")
```

含義

號碼：	NUMBER 指令會傳回由 <string> 所呈現的數字，作為 REAL 值。		
<字串>：	待轉換的 STRING 變數		
<REAL_RES>：	供具有 NUMBER 的類型轉換之結果的變數		
	類型：	REAL	
ISNUMBER：	ISNUMBER 指令可檢查 <字串> 是否能轉換為有效數字。		
<BOOL_RES>：	供具有 ISNUMBER 之詢問結果的變數		
	類型：	BOOL	
	值：	TRUE	若 <string> 依照語言規則，呈現了有效的 REAL 數字，ISNUMBER 將提供 TRUE 值。
		FALSE	若 ISNUMBER 提供 FALSE 值，以相同的 <string> 呼叫 NUMBER 時，則會產生警報。
AXNAME：	AXNAME 指令會將特定的 <string> 轉換成座標軸識別碼。 **附註：** 若無法將 <string> 指派給一個已配置的座標軸識別碼，則會產生警報。		
<AXIS_RES>：	供具有 AXNAME 的類型轉換之結果的變數		
	類型：	AXIS	

範例

```
程式碼                                          註解
DEF BOOL BOOL_RES
DEF REAL REAL_RES
DEF AXIS AXIS_RES
REAL_RES == 1234.9876Ex-7                       ;BOOL_RES == TRUE
BOOL_RES=ISNUMBER("1234XYZ")                    ;BOOL_RES == FALSE
REAL_RES=NUMBER("1234.9876Ex-7")                ;REAL_RES == 1234.9876Ex-7
AXIS_RES=AXNAME("X")                            ;AXIS_RES == X
```

2.9.3 串鍊字串（<<）

“串鍊字串”函數允許從獨立元件中配置字串。

使用運算子「**<<**」來表示串鍊。 這個運算子，以 STRING 為一目標類型，作為所有基礎類型 CHAR、BOOL、INT、REAL 以及 STRING，結合時的基本類型。 所有可能需要的轉換，會依照現有的規則來實行。

語法

```
<any_type> << <any_type>
```

含義

<any_type>：	變數類型 CHAR、BOOL、INT、REAL 或 STRING
<< ：	運算子用來串連變數 (<any_type>)，以配置一個字元字串（STRING 類型）。 這個運算子也可單獨使用，作為所謂的「unary」變數。 並可用來做精確的類型轉換為 STRING（不是 FRAME 和 AXIS）： << <any_type>

例如，可以從文字清單配置的訊息或指令，或可以被插入（例如一個單節名稱）的參數：
`MSG(STRG_TAB[LOAD_IDX]<<BLOCK_NAME)`

說明

這種字串串鍊的中間結果，不可以超過最大字串長度。

說明

FRAME 和 AXIS 類型，無法和運算子「**<<**」一起使用。

範例

範例 1： 串鍊字串

程式碼	註解
`DEF INT IDX=2`	
`DEF REAL VALUE=9.654`	
`DEF STRING[20] STRG="INDEX：2"`	
`IF STRG=="Index："<<IDX GOTOF NO_MSG`	

程式碼	註解
MSG（"Index："<<IDX<<"/value："<<VALUE）	；顯示： "Index：2/value：9.654"
NO_MSG：	

範例 2： 以 << 進行精確的類型轉換

程式碼	註解
DEF REAL VALUE=3.5	
<<VALUE	；特定 REAL 類型變數會轉換成一個 STRING 類型。

2.9.4 轉換為大小寫字母（TOLOWER、TOUPPER）

「conversion to lowercase/uppercase letters」（轉換至大小寫字母）函數允許所有字串的字母，轉換成標準表示法。

語法

```
<STRING_RES>=TOUPPER("<字串>")
<STRING_RES>=TOLOWER("<字串>")
```

含義

TOUPPER：	使用 TOUPPER 指令，所有字元字串中的字母，都會轉換成**大寫**字母。	
TOLOWER：	使用 TOLOWER 指令，所有字元字串中的字母，都會轉換成**小寫**字母。	
<字串>：	待轉換的字元字串	
	類型：	STRING
<STRING_RES>：	供轉換結果所用的變數	
	類型：	STRING

範例

使用者輸入可以從使用者介面開始，因此可以給予標準大小寫（大寫或小寫）：

程式碼

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

2.9.5 決定字串長度（STRLEN）

STRLEN 指令可決定字元字串的長度。

語法

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

含義

STRLEN：	STRLEN 指令可決定特定字元字串的長度。 字元數字不是 0 的字元，會從傳回的字串開頭，開始計數。	
<字串>：	長度待決定的字元字串	
	類型：	STRING
<INT_RES>：	供維度結果所用的變數	
	類型：	INT

範例

和單字元存取一起，此函數允許字元字串的結尾待決定：

程式碼

```
IF (STRLEN(BLOCK_NAME)>10) GOTOF ERROR
```

2.9.6 在字串中搜尋字元 / 字串（INDEX、RINDEX、MINDEX、MATCH）

此功能會在字串中搜尋單字元或字串。 此函數結果會指定已搜尋過的字串中，已定位字元 / 字串的所在處。

語法

`INT_RES=INDEX(STRING,CHAR)`；**結果類型： INT**

`INT_RES=RINDEX(STRING,CHAR)`；**結果類型：INT**

`INT_RES=MINDEX(STRING,STRING)`；**結果類型：INT**

`INT_RES=MINDEX(STRING,STRING)`；**結果類型：INT**

語義

搜尋函數： 提供了字串（第一參數），已成功搜尋到的位置。 若無法找到字元 / 字串，則會傳回值-1。 第一字元具有位置 0。

含義

INDEX：	在第一參數中，搜尋已指定為第二參數（從頭開始找）的字元。
RINDEX：	在第一參數中，搜尋已指定為第二參數（從結尾開始找）的字元。
MINDEX：	對應至 INDEX 函數，除了已傳輸字元清單（作為字串），且當中找到的字元索引會被傳回的情形以外。
MATCH：	在字串中搜尋字串。

這允許字串能根據特定條件來打斷，例如，在空白處或路徑分隔符號（「/」）之處。

範例

打斷一個輸入，使其進入路徑和單節名稱

程式碼	註解
`DEF INT PFADIDX，PROGIDX`	
`DEF STRING[26] INPUT`	
`DEF INT LISTIDX`	
`INPUT = "/_N_MPF_DIR/_N_EXECUTE_MPF"`	
`LISTIDX = MINDEX （INPUT，"M，N，O，P"） + 1`	；傳回 LISTIDX 的值為 3；因為從清單選擇開始處起算，「N」是參數 INPUT 中的第一個字元。
`PFADIDX = INDEX （INPUT，"/"） +1`	；因此下列均適用： PFADIDX = 1
`PROGIDX = RINDEX （INPUT，"/"） +1`	；因此下列均適用： PROGIDX = 12
	；下一章節會介紹的 SUBSTR 函數，可以用來打斷在「路徑」和「模組」元件中的 INPUT 變數：
`VARIABLE = SUBSTR (INPUT, PFADIDX, PROGIDX-PFADIDX-1)`	；隨後傳回「_N_MPF_DIR」
`VARIABLE = SUBSTR （INPUT，PROGIDX）`	；隨後傳回「_N_EXECUTE_MPF」

2.9.7 選擇子字串（SUBSTR）

字串內的任意部分可使用 SUBSTRING 函數讀取。

語法

```
<STRING_RES>=SUBSTR(<字串>,<索引>,<長度>)
```

```
<STRING_RES>=SUBSTR(<字串>,<索引>)
```

含義

SUBSTR：	此函數將回傳來自 **<**字串**>** 的子字串，從 **<**索引**>** 開始，具有特定的 **<**長度**>**。 若未指定 **<**長度**>** 參數，該函數將回傳從 **<**索引**>** 開始直到字串結尾的子字串。
<索引>：	字串內子字串的起始位置。 若起始位置在字串結尾之後，則會傳回空字串 (" ")。 字串的第一個字元： 索引 = 0 值域： 0 ...（字串長度 - 1）
<長度>：	子字串的長度。 若指定的長度過長，回傳的子字串最多只會到字串結尾。 值域： 1 ...（字串長度 - 1）

範例

程式碼	註解
`DEF STRING[29] RES`	
`;                    1`	
`;          0123456789012345678`	
`RES = SUBSTR("QUITTUNG: 10 to 99", 10, 2)`	`; RES == "10"`
`RES = SUBSTR("QUITTUNG: 10 to 99", 10)`	`; RES == "10 to 99"`

2.9.8 個別字元的讀取和寫入

可讀取和寫入字串內的個別字元。

必須遵守以下的補充條件：

- 僅適用於使用者定義變數，無法用於系統變數
- 字串的個別字元僅使用子程式「傳值呼叫 (call by value)」的方式傳送

語法

```
<字元>=<字串>[<索引>]
<字元>=<字串陣列>[<陣列索引>，<索引>]
<字串>[<索引>]=<字元>
<字串陣列>[<陣列索引>，<索引>]=<字元>
```

含義

<字串>：	任何字串
<字元>：	類型 CHAR 的變數
<索引>：	字串內字元的位置。 字串的第一個字元： 索引 = 0 值域： 0 ...（字串長度 - 1）

範例

範例 1： 變數訊息

程式碼	註解
`;  0123456789`	
`DEF STRING [50] MESSAGE = "Axis n has reached position"`	
`MESSAGE [6] = "X"`	
`MSG （MESSAGE）`	`; "Axis X has reached position"`

範例 2： 評估系統變數

程式碼	註解
`DEF STRING [50] STRG`	；系統變數的緩衝
`...`	
`STRG = $P_MMCA`	；載入系統變數
`IF STRG[0] == "E" GOTO ...`	；評估系統變數

範例 3： 參數以「傳值呼叫 (call by value)」和「傳址呼叫 (call by reference)」的方式傳送

程式碼	註解
`;  0123456`	
`DEF STRING[50] STRG = "Axis X"`	
`DEF CHAR CHR`	
`...`	
`EXTERN UP_VAL(ACHSE)`	；具「傳值呼叫」參數的子程式定義
`EXTERN UP_REF(VAR ACHSE)`	；具「傳址呼叫」參數的子程式定義
`...`	
`UP_VAL(STRG[6])`	；參數以「傳值」的方式傳送
`...`	
`CHR = STRG[6]`	；緩衝區
`UP_REF(CHR)`	；參數以「傳址」的方式傳送

2.9.9 格式化字串 (SPRINT)

使用預先定義的 SPRINT 功能，可以將字串格式化，例如準備由外部裝置輸出（另請參閱「Process DataShare - 輸出至外部裝置 / 檔案（EXTOPEN、WRITE、EXTCLOSE）：(頁 721)」）。

語法

```
"<Result_string>"=SPRINT("<Format_string>",<value_1>,<value_2>,...
., <value_n>)
```

含義

`SPRINT`：	提供數值給預先定義功能的識別碼，類型 STRING。
`"<Format_String>"`：	包含固定及變數元件的字元字串。 變數元件使用格式的控制字元 % 及接續的格式說明定義。
`< value_1>,< value_2>,…,< value_n>`：	常數或 NC 變數格式中的數值，插入位置在第 n 個格式控制字元 % 的所在處，對應於 <format_string> 的格式說明。
`"<result_string>"`：	格式化的字元字串（最大 400 位元組）

提供格式說明

%B：	轉換成「TRUE」字串，如果將數值轉換為：
	● 不等於 0。
	● 非空字串（字串數值使用）。
	轉換成「FALSE」字串，如果將數值轉換為：
	● 等於 0。
	● 為空字串。
	範例：
	`N10 DEF BOOL BOOL_VAR=1`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)`
	結果： 將字元字串「CONTENT OF BOOL_VAR:TRUE」寫入 RESULT 字串變數。

%C：	轉換成 ASCII 字元。
	範例：
	`N10 DEF CHAR CHAR_VAR="X"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C",CHAR_VAR)`
	結果： 將字元字串「CONTENT OF CHAR_VAR:X」寫入字串變數 RESULT。

%D：	轉換成整數值 (INTEGER) 的字串。
	範例：
	`N10 DEF INT INT_VAR=123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D",INT_VAR)`
	結果： 將字元字串「CONTENT OF INT_VAR:123」寫入字串變數 RESULT。

%<m>D：	轉換成整數值 (INTEGER) 的字串。 字串最小長度為 **<m>** 個字元。 缺少的位置用空格填滿，向左對齊。
	範例：
	`N10 DEF INT INT_VAR=-123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D",INT_VAR)`
	結果： 將字元字串「CONTENT OF INT_VAR:xx-123」寫入字串變數 RESULT（範例中的「x」代表空格）。

%F：	轉換成有 6 個小數點位數的小數字串。 必要時，小數點位數四捨五入或填入 0。
	範例：
	`N10 DEF REAL REAL_VAR=-1.2341234EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR)`
	結果： 字串變數 RESULT 寫入時使用字元字串「CONTENT OF REAL_VAR: -1234.123400".

%<m>F：	轉換成有 6 個小數點位數的小數字串，總長度至少 **<m>** 個字元。 必要時，小數點位數四捨五入或填入 0。缺少的字元使用空格填滿至總長度 **<m>**，向左對齊。
	範例：
	`N10 DEF REAL REAL_VAR=-1.23412345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)`
	結果： 字串變數 RESULT 寫入時使用字元字串「CONTENT OF REAL_VAR: xxx-1234.123457」（其中「x」為空格佔位符號）。

%.<n>F：	轉換成有 **<n>** 個小數點位數的小數字串。 必要時，小數點位數四捨五入或填入 0。
	範例：
	`N10 DEF REAL REAL_VAR=-1.2345678EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:-1234.568」寫入字串變數 RESULT。

%<m>.<n>F：	轉換成有 **<n>** 個小數點位數的小數字串，總長度至少 **<m>** 個字元。 必要時，小數點位數四捨五入或填入 0。缺少的字元使用空格填滿至總長度 **<m>**，向左對齊。
	範例：
	`N10 DEF REAL REAL_VAR=-1.2341234567890EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:xx-1234.12」寫入字串變數 RESULT（範例中的「x」代表空格）。

%E：	轉換以指數表示的小數字串。 尾數保存，前一個小數點及 6 個小數點位數標準化。 必要時，小數點位數四捨五入或填入 0。指數用關鍵字「EX」開頭。 緊跟在後的是符號（「+」或「-」），以及兩位數或三位數。
	範例：
	`N10 DEF REAL REAL_VAR=-1234.567890`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:-1.234568EX+03」寫入字串變數 RESULT。

%<m>E：	轉換成指數表示式的小數字串，總長度至少 **<m>** 個字元。 缺少的字元用空格填滿，向左對齊。 尾數保存，前一個小數點及 6 個小數點位數標準化。 必要時，小數點位數四捨五入或填入 0。指數用關鍵字「EX」開頭。 緊跟在後的是符號（「+」或「-」），以及兩位數或三位數。
	範例：
	`N10 DEF REAL REAL_VAR=-1234.5`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:xxxxxx-1.234500EX +03」寫入字串變數 RESULT（範例中的「x」代表空格）。

%.<n>E：	轉換以指數表示的小數字串。 尾數保存，前一個小數點及 **<n>** 個小數點位數標準化。 必要時，小數點位數四捨五入或填入 0。指數用關鍵字「EX」開頭。 緊跟在後的是符號（「+」或「-」），以及兩位數或三位數。
	範例：
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:-1.23EX+03」寫入字串變數 RESULT。

%<m>.<n>E：	轉換成指數表示式的小數字串，總長度至少 **<m>** 個字元。 缺少的字元用空格填滿，向左對齊。 尾數保存，前一個小數點及 **<n>** 個小數點位數標準化。 必要時，小數點位數四捨五入或填入 0。指數用關鍵字「EX」開頭。 緊跟在後的是符號（「+」或「-」），以及兩位數或三位數。
	範例：
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:xx-1.23EX+03」寫入字串變數 RESULT（範例中的「x」代表空格）。

%G：	轉換成小數的字串 – 視數值範圍而定 – 以小數或指數表示： 如果表示的絕對值小於 1.0EX-04 或大於等於 1.0EX+06，則會選取指數標記，否則為小數標記。 最多顯示六個數值位數，必要時四捨五入。
	小數標記的範例：
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:0.000123457」寫入字串變數 RESULT。
	指數標記的範例：
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:1.23457EX+06」寫入字串變數 RESULT。

%<m>G：	轉換成小數的字串 – 視數值範圍而定 – 以小數或指數標記表示（就像 %G）。 字串總長度至少有 **<m>** 個字元。 缺少的字元用空格填滿，向左對齊。
	小數標記的範例：
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:xxxx0.000123457」寫入字串變數 RESULT（範例中的「x」代表空格）。
	指數標記的範例：
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:xxx1.23457EX+06」寫入字串變數 RESULT（範例中的「x」代表空格）。

%.<n>G：	轉換成小數的字串 – 視數值範圍而定 – 以小數或指數表示。 最多顯示 **<n>** 個數值位數，必要時四捨五入。 如果表示的絕對值小於 1.0EX-04 或大於等於 1.0EX(+<n>)，則會選取指數標記，否則為小數標記。
	小數標記的範例：
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:0.000123」寫入字串變數 RESULT。
	指數標記的範例：
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:1.23EX+03」寫入字串變數 RESULT。

%<m>.<n>G：	轉換成小數的字串 – 視數值範圍而定 – 以小數或指數標記表示（就像 %.<n>G）。 字串總長度至少有 **<m>** 個字元。 缺少的字元用空格填滿，向左對齊。
	小數標記的範例：
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:xxx0.0001235」寫入字串變數 RESULT（範例中的「x」代表空格）。
	指數標記的範例：
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	結果： 將字元字串「CONTENT OF REAL_VAR:xx1.235EX+06」寫入字串變數 RESULT（範例中的「x」代表空格）。

%.<n>P：	將 REAL 數值轉換成 INTEGER 數值，並考量 **<n>** 個小數點位數。INTEGER 數值是以 32 位元的二進制數字輸出。 如果轉換的數值無法以 32 位元表示，則發出警報並中斷處理。
	由於使用格式陳述式 %.<n>P 產生的位元組順序也含有二進制零位，依此方法產生的總字串不再對應於 NC 資料類型 STRING 的規則。 結果將無法儲存於變數、類型 STRING，也無法再使用 NC 語言的字串指令進一步處理。 唯一僅存的用處是將參數傳送至 WRITE 指令，由適當的外部裝置輸出（請參閱以下範例）。
	一旦 **<Format_String>** 包含格式說明以後，藉由 %.<n>P 產生的二進制數字例外，輸出的類型 %P 再接著完整的字串對應於 ASCII 字元代碼、ISO (DIN6024) 或 EIA (RS244) 的 MD10750 $MN_SPRINT_FORMAT_P_CODE。 如果程式設計無法轉換的字元，則發出警報中斷處理。
	範例：
	`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`
	`... ; 處理錯誤`
	`N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)`
	`N210 IF ERROR <> 0`
	`... ; 處理錯誤`
	結果： 將字串「INTEGER BINARY CODED: 'H0001E23A'」傳輸至輸出裝置 /ext/dev/1。十六進位數值 0x0001E23A 對應於十進位數值 123450。

%<m>.<n>P：	以下列方式將對應於機械資料 MD10751 $MN_SPRINT_FORMAT_P_DECIMAL 設定的 REAL 數值轉換成字串：
	• **<m> + <n>** 位數的整數值**或**
	• 小數點前最多 **<m>** 位數的十進位數字，及精確的 **<n>** 個小數點位數。
	如同格式說明 %.<n>P 一般，完整的字串係儲存在 MD10750 $MN_SPRINT_FORMAT_P_CODE 所定義的字元代碼中。
	MD10751 = 0 的轉換：
	將 REAL 數值以 **<m> + <n>** 個位數的整數值轉換成字串。 必要時，小數點位數四捨五入至 **<n>** 位數，或以 0 填滿。缺少的預小數點位數則以空白填滿。 加上負號，向左對齊；正號則以空格取代。
	範例：
	`N10 DEF REAL REAL_VAR=-123.45`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)`
	結果： 將字元字串「PUNCHED TAPE FORMAT:-xx123450」寫入字串變數 RESULT（範例中的「x」代表空格）。
	MD10751 = 1 的轉換：
	以最大 **<m>** 個預十進制位數的十進位數字，及精確的 **<n>** 個小數點位數，將 REAL 數值轉換成字串。 必要時，預十進制位數將截斷，並將小數點位數四捨五入或填滿 0。如果 **<n>** 等於 0，則小數點位數也一併省略。
	範例：
	`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)`
	結果： 將字元字串「PUNCHED TAPE FORMAT:-123.450 VAR2:23」寫入字串變數 RESULT。

%S：	插入字串。
	範例：
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)`
	結果： 將字元字串「CONTENT OF STRING_VAR:ABCDEFG」寫入字串變數 RESULT。

%<m>S：	插入最少 **<m>** 個字元的字串。 缺少的位置用空格填滿。 **範例：** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR) 結果： 將字元字串「CONTENT OF STRING_VAR:xxxABCDEFG」寫入字串變數 RESULT（範例中的「x」代表空格）。
%.<n>S：	插入 **<n>** 個字元的字串（從第一個字元開始）。 **範例：** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR) 結果： 將字元字串「CONTENT OF STRING_VAR:ABC」寫入字串變數 RESULT。
%<m>.<n>S：	插入 **<n>** 個字元的字串（從第一個字元開始）。 產生的字串總長度至少有 **<m>** 個字元。 缺少的位置用空格填滿。 **範例：** N10 DEF STRING[16] STRING_VAR="ABCDEFG" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR) 結果： 將字元字串「CONTENT OF STRING_VAR:xxxxxABCDE」寫入字串變數 RESULT（範例中的「x」代表空格）。
%X：	將 INTEGER 數值轉換成十六進位標記的字串。 **範例：** N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR) 結果： 將字元字串「INTEGER HEXADECIMAL:A5B8」寫入字串變數 RESULT。

說明

未區分識別碼及關鍵字文字大小寫的 NC 語言屬性也適用於格式說明。 因此，可以使用小寫或大寫的文字程式設計，功能毫無差別。

組合選項

下表說明 NC 資料類型與格式說明如何搭配。有關不明確資料類型的轉換規則亦適用（請參閱「資料類型 (頁 63)」）。

	NC 資料類型						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

說明

表格指出 NC 資料類型 AXIS 及 FRAME 無法直接用在 SPRINT 功能上。 但是，它可以：

- 將 AXIS 資料類型使用 AXSTRING 功能轉換成字串，接著再以 SPRINT 處理。
- 讀取每個框架元件存取的 FRAME 資料類型個別數值。 因此，可以取得用 SPRINT 處理之後的 REAL 資料類型。

2.10 程式跳躍與分支

2.10.1 傳回跳躍至程式開始處（GOTOS）

GOTOS 指令能用來跳回主程式或子程式的開始處，以便能重複執行該程式。

機械資料可以用來設定每一次的傳回，要跳到程式開始處：

- 程式執行時間設定為「0」。
- 工件計數因為值「1」增加。

語法

GOTOS

含義

GOTOS：	目的地為程式開始處的跳躍陳述式。	
	執行是透過 NC/PLC 介面訊號來控制： DB21，至 DBX384.0（控制程式分支）	
	值：	含義：
	0	沒有將跳躍傳回程式開始處。 會在 GOTOS 之後的下一個工件程式單節重新開始執行程式。
	1	將跳躍傳回程式開始處。 已重複執行工件程式。

補充條件

- GOTOS 會內部初始一個 STOPRE （預先處理的停止）。
- 對於具有資料定義（LUD 變數）的子程式，若該資料定義具有 GOTOS，則會在定義區段之後，跳躍至第一個程式單節，換言之，資料定義不會被重複執行。 這就是所定義的變數，在到達 GOTOS 單節時，仍可保持其值，並且不會被重置為定義區段中所程式設計標準值的原因。
- GOTOS 指令不可用於同步動作和技術循環。

範例

程式碼	註解
N10 ...	；程式開始處。
...	
N90 GOTOS	；跳躍至程式開始處。
...	

2.10.2 程式跳躍至跳躍標記（GOTOB、GOTOF、GOTO、GOTOC）

可在程式中設定跳躍標籤，透過指令 GOTOF、GOTOB、GOTO 或 GOTOC，便能在相同程式中，從其他位置跳躍至此。程式會從馬上接在跳躍標籤之後的陳述式，重新開始執行。這表示程式中可以實現分支程式。

除了跳躍標籤以外，主單節和副單節數字也可作為跳躍指定。

若在跳躍陳述式之前，已形成一個跳躍條件（IF ...），則只會在滿足跳躍條件時，才會執行程式跳躍。

語法

```
GOTOB <跳躍目的地>
IF <跳躍條件> = TRUE GOTOB <跳躍目的地>

GOTOF <跳躍目的地>
IF <跳躍條件> = TRUE GOTOF <跳躍目的地>

GOTO <跳躍目的地>
IF <跳躍條件> = TRUE GOTO <跳躍目的地>

GOTOC <跳躍目的地>
IF <跳躍條件> = TRUE GOTOC <跳躍目的地>
```

含義

GOTOB：	具有朝向程式開始處的跳躍目的地，其跳躍陳述式。
GOTOF：	具有朝向程式結尾處的跳躍目的地，其跳躍陳述式。
GOTO：	具有跳躍目的地搜尋的跳躍陳述式。搜尋會先朝向程式結尾處進行，然後再朝向程式開始處進行。

GOTOC：	有些影響，例如具有差距的 GOTO，不會發出警報 14080「找不到跳躍指定」。 這表示在搜尋跳躍目的地失敗的情況中，不會中斷程式執行，且會繼續依照 GOTOC 指令執行程式行。	
<跳躍目的地>：	跳躍目的地參數 可能的資料包含：	
	<跳躍標籤**>**：	跳躍目的地為程式中設定的跳躍標籤，含有使用者定義名稱：**<**跳躍標籤>：
	<單節號碼**>**：	跳躍目的地為主單節或副單節號碼（例如：200，N300）
	字串類型變數：	變數跳躍目的地。此變數表示跳躍標籤或單節號碼。
IF：	形成跳躍條件的關鍵字。 跳躍條件允許使用所有的比較和邏輯運算（結果：TRUE 或 FALSE）。若運算結果為 TRUE，則會執行程式跳躍。	

說明

跳躍標籤

跳躍標籤永遠位於單節開始處。若同時有程式編號存在，則跳躍標籤會立即位於在單節編號之後。

當為跳躍標籤命名時，需套用下列規則：

- 字元數：
  - 最小 2
  - 最大 32
- 可允許的字元包括：
  - 字母
  - 號碼
  - 底線
- 前兩個字元必須為字母或底線。
- 跳躍標籤的名稱後方接冒號（「：」）。

補充條件

- 跳躍目的地僅能是具有跳躍標籤或單節號碼的單節，或是位於程式內的單節。
- 不具有跳躍條件的跳躍陳述式，必須在獨立單節進行程式設計。此限制不適用於具有跳躍條件的跳躍陳述式。在這種情況中，可在單節中形成數個跳躍陳述式。
- 具有跳躍陳述式，但不具有跳躍條件的程式，程式結尾 M2/M30 不一定要放在程式結尾。

範例

範例 1：跳躍至跳躍標籤

程式碼	註解
N10...	
N20 GOTOF Label_1	；朝程式結尾跳躍至
	；跳躍標籤「Label_1」。
N30 …	
N40 Label_0：R1=R2+R3	；跳躍標籤「Label_0」已設定。
N50 …	
N60 Label_1：	；跳躍標籤「Label_1」已設定。
N70 …	
N80 GOTOB Label_0	；朝程式開始處跳躍
	；至跳躍標籤「Label_0」。
N90 ...	

範例 2：間接跳躍至單節號碼

程式碼	註解
IF <條件> == TRUE	
R10=100	；指定跳躍目的地
ELSE	
R10=110	；指定跳躍目的地
ENDIF	
；朝程式結尾處跳躍至單節號碼位於 R10 的單節	
N10 GOTOF "N"<<R10	
...	
N90 …	
N100 …	；跳躍目的地
N110 …	
...	

範例 3：跳躍至變數跳躍目的地

程式碼	註解
DEF STRING[20] DESTINATION	
IF <條件> == TRUE	
DESTINATION = "Label1"	；指定跳躍目的地
ELSE	
DESTINATION = "Label2"	；指定跳躍目的地
ENDIF	
；朝程式結尾跳躍至變數跳躍目的地「DESTINATION 內容」。	
GOTOF DESTINATION	
Label1：T="Drill1"	；跳躍目的地 1
...	
Label2：T="Drill2"	；跳躍目的地 2
...	

範例 4：以跳躍條件跳躍

程式碼	註解
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	；指派初始值
N41 LA1：G0 X=R2*COS（R1）+R5 Y=R2*SIN（R1）+R6	；跳躍標籤 LA1
N42 R1=R1+R3 R4=R4-1	
；IF 跳躍條件 == TRUE	
；THEN 朝程式開始處跳躍至跳躍標籤 LA1	
N43 IF R4>0 GOTOB LA1	
N44 M30	；程式結束

2.10.3 程式分支（CASE ... OF ... DEFAULT ...）

CASE 函數提供了檢查（類型： INT）變數或算術函數的實際值的可能性，並依照此結果，跳躍至程式中不同的位置。

語法

```
CASE(<算式>) OF <常數_1> GOTOF <跳躍目標_1> <常數_2> GOTOF <跳躍目標_2> ... DEFAULT GOTOF <跳躍目標_n>
```

含義

CASE：	跳躍敘述
<算式>：	變數或算術函數
OF：	形成有條件的分支程式的關鍵字。

<常數_1>：	變數或算術函數的第一指定常數值	
	類型：	INT
<常數_2>：	變數或算術函數的第二指定常數值	
	類型：	INT
DEFAULT：	若變數或算術函數沒有採用任何特定常數值，則 DEFAULT 陳述式可用來決定跳躍目標。 **附註：** 若尚未程式設計 DEFAULT 陳述式，則在此情形中，跟在 CASE 陳述式後面的單節，就會成為跳躍目標。	
GOTOF：	具有朝向程式結尾處的跳躍目標，其跳躍指令。 不使用 GOTOF，可程式設計所有其他的 GOTO 指令（請參考「程式跳躍至跳躍標記」）。	
<跳躍目標_1>：	若變數值或算術函數對應到第一特定常數，則會對此跳躍目標執行一個分支。 可將跳躍目標指定為：	
	<跳躍標記**>**：	跳躍目標是以使用者自訂之名稱於程式中設定之跳躍標記（標籤），該自訂名稱為：**<**跳躍標記>：
	<單節號碼**>**：	跳躍目標為主單節或副單節號碼（例如：200，N300）
	字串類型變數：	變數跳躍目標。 此變數表示跳躍標記或單節號碼。
<跳躍目標_2>：	若變數值或算術函數對應到第二特定常數，則會對此跳躍目標執行一個分支。	
<跳躍目標_n>：	若變數值或算術函數不採用特定常數值，則會對此跳躍目標執行一個分支。	

範例

程式碼

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

程式碼
...

CASE 陳述式，從 N30 而來，定義了下列分支程式的可能性：

1. 若算術函數的值 VAR1+VAR2-VAR3 = 7，便跳躍至具有跳躍標記定義「Label_1」的單節（→ N40）。
2. 若算術函數的值 VAR1+VAR2-VAR3 = 9，便跳躍至具有跳躍標記定義「Label_2」的單節（→ N50）。
3. 若算術函數的值 VAR1+VAR2-VAR3 不等於 7 也不等於 9，便跳躍至具有跳躍標記定義「Label_3」的單節（→N60）。

2.11 重複程式區段（REPEAT，REPEATB，ENDLABEL，P）

程式區段重複允許您在所有程式中，以任何順序重複現有的程式區段。

要重複的程式行或程式區段，由跳躍標記來識別（標籤）。

說明

跳躍標記（標籤）

跳躍標記永遠位於單節開始處。 若同時有程式編號存在，則跳躍標記會立即位於在單節編號之後。

當為跳躍標記命名時，需套用下列規則：

- 字元數：
  - 最小 2
  - 最大 32
- 可允許的字元包括：
  - 字母
  - 號碼
  - 底線
- 前兩個字元必須為字母或底線。
- 跳躍標記的名稱在冒號（「：」）後。

語法

1. 重複獨立程式行：

```
<跳躍標記>： ...
...
REPEATB <跳躍標記> P=<n>
...
```

2. 重複介於跳躍標記和 REPEAT 陳述式之間的程式區段：

```
<跳躍標記>： ...
...
REPEAT <跳躍標記> P=<n>
...
```

3. 重複介於兩跳躍標記之間的區段：

```
<開始跳躍標記>： ...
...
<結尾跳躍標記>： ...
...
```

```
REPEAT <開始跳躍標記> <結尾跳躍標記> P=<n>
...
```

說明

不可以在圓括號中，將具有兩個跳躍標記的 REPEAT 敘述，套巢狀使用在一起。 若 <開始跳躍標記> 在 REPEAT 敘述之前出現，且 <結尾跳躍標記> 未在 REPEAT 敘述之前到達，則會重複介於 <開始跳躍標記> 與 REPEAT 敘述之間的區段。

4. 介於跳躍標記和 ENDLABEL 之間的重複區段：

```
<跳躍標記>： ...
...
ENDLABEL：...
...
REPEAT <跳躍標記> P=<n>
...
```

說明

不可在圓括號中，將具有 <跳躍標記> 與 ENDLABEL 的 REPEAT 敘述套巢狀使用。 若 <跳躍標記> 在 REPEAT 敘述之前出現，且 ENDLABEL 未在 REPEAT 敘述之前到達，則會重複介於 <跳躍標記> 與 REPEAT 敘述之間的區段。

含義

REPEATB：	用來重複程式行的指令
REPEAT：	用來重複程式區段的指令

<跳躍標記>：	<跳躍標記> 識別： • 要重複的程式行（在 REPEATB 的情況下） 或 • 要重複的程式區段開始處（在 REPEAT 的情況下）
	程式行係由<跳躍標記>識別，可在 REPEAT/REPEATB 敘述之前或之後出現。 此搜尋會朝程式開始處方向進行。 若此方向沒有找到跳躍標記，則會朝向程式結尾繼續進行搜尋。 **例外：** 若介於跳躍標記與 REPEAT 敘述之間的程式區段，需要被重複（請參考 2. 語法），由 <跳躍標記> 所識別的程式行，必須出現在 REPEAT 敘述**之前**，因為在此情況中，搜尋**只會**朝向程式開始處執行。 若具有<跳躍標記>的程式行中，包含了進一步的操作，則會在各個重複時再次執行。
ENDLABEL：	關鍵字標記要重複的程式區段結尾。 若具有<ENDLABEL>的程式行中，包含了進一步的操作，則會在各個重複時再次執行。 ENDLABEL 在可程式中使用超過一次。
P：	用來指定重複次數的位址
<n>：	程式區段重複的號碼
	類型： INT
	要重複的程式區段重複了 <n> 次。 在最後一次重複後程式會在 REPEAT/REPEATB 行之後，繼續執行。 **附註：** 若少了指定給 P=<n> 的號碼，程式區段只會重複一次。

範例

範例 1： 重複獨立程式行

程式碼	註解
N10 POSITION1： X10 Y20	
N20 POSITION2： CYCLE(0,,9,8)	；位置循環
N30 ...	
N40 REPEATB POSITION1 P=5	；執行 BLOCK N10 五次。
N50 REPEATB POSITION2	；執行單節 N20 一次。
N60 ...	

程式碼	註解
N70 M30	

範例 2： 重複介於跳躍標記和 REPEAT 陳述式之間的程式區段：

程式碼	註解
N5 R10=15	
N10 Begin： R10=R10+1	；寬度
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	；執行從 N10 到 N70 的區段四次。
N90 Z10	
N100 M30	

範例 3： 重複介於兩跳躍標記之間的區段

程式碼	註解
N5 R10=15	
N10 Begin：R10=R10+1	；寬度
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	；執行從 N10 到 N70 的區段三次。
N110 Z10	
N120 M30	

範例 4： 介於跳躍標記和 ENDLABEL 之間的重複區段

程式碼	註解
N10 G1 F300 Z-10	
N20 BEGIN1：	
N30 X10	
N40 Y10	
N50 BEGIN2：	
N60 X20	
N70 Y30	
N80 ENDLABEL： Z10	

程式碼	註解
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3： X20	
N120 Y30	
N130 REPEAT BEGIN3 P=3	；執行從 N110 到 N120 的區段三次。
N140 REPEAT BEGIN2 P=2	；執行從 N50 到 N80 的區段兩次。
N150 M100	
N160 REPEAT BEGIN1 P=2	；執行從 N20 到 N80 的區段兩次。
N170 Z10	
N180 X0 Y0	
N190 M30	

範例 5： 銑削，以不同技術進行機台鑽頭定位

程式碼	註解
N10 CENTER DRILL（）	；載入中心鑽頭。
N20 POS_1：	；鑽孔定位 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL：	
N80 POS_2：	；鑽孔定位 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL：	
N130 DRILL（）	；變更鑽頭與鑽孔循環。
N140 THREAD（6）	；載入螺牙 M6 以及螺紋循環。
N150 REPEAT POS_1	；從 POS_1 向上到 ENDLABEL，重複程式區段一次。
N160 DRILL（）	；變更鑽頭與鑽孔循環。
N170 THREAD（8）	；載入螺牙 M8 以及螺紋循環。
N180 REPEAT POS_2	；從 POS_2 向上到 ENDLABEL，重複程式區段一次。
N190 M30	

其他資訊

- 可以巢狀使用程式區段重複。 每個呼叫使用一個子程式層級。
- 若在進行程式區段重複的過程中，程式設計了 M17 或 RET，則該重複將會取消。 程式會接在 REPEAT 行之後的單節，重新開始。

- 在實際的程式顯示中，程式區段重複會以獨立的子程式層級顯示。
- 若在程式區段重複時取消層級，則程式會在程式區段重複呼叫之後，重新進行。
  範例：

程式碼	註解
N5 R10=15	
N10 BEGIN： R10=R10+1	；寬度
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	；中斷層級
N50 X=-R10	
N60 Y=-R10	
N70 END： Z10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	
N120 Z10	；重新進行程式執行。
N130 M30	

- 檢查結構與程式區段重複可配合使用。 然而，二者之間不能有所重疊。 程式區段重複應出現在檢查結構的範圍之內，或檢查結構應當出現在程式區段重複之內。
- 若跳躍與程式區段重複混和，則該單節僅會單純地依序執行。 例如，若從程式區段重複處，執行一個跳躍，則會繼續進行，直到找到了程式設計的程式區段結尾為止。
  範例：

程式碼
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

說明

REPEAT 敘述應出現在移動單節之後。

2.12 檢查結構

控制會將 NC 單節，處理為已程式設計的順序之標準。

此順序，可能會因為程式設計替代的程式單節和程式迴圈，而有所不同。使用關鍵字 IF、ELSE、ENDIF、LOOP，FOR，WHILE 以及 REPEAT，來程式設計這些檢查結構。

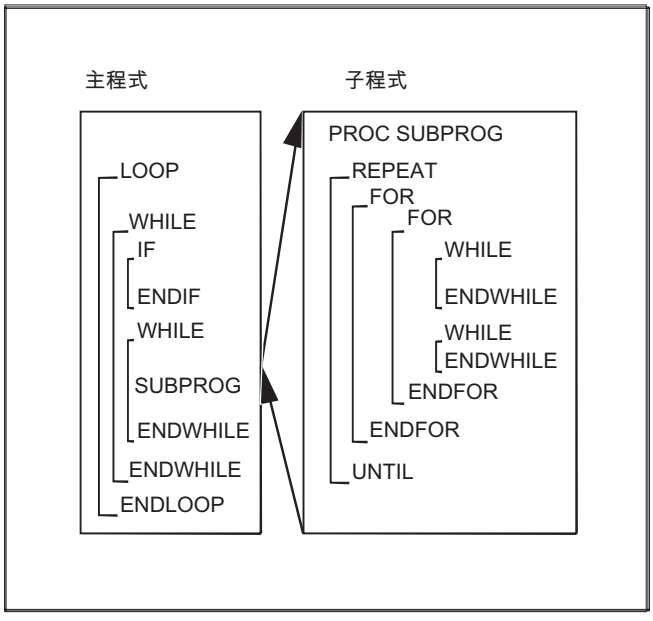
注意
程式設計錯誤 檢查結構僅能在程式的敘述區段中插入。程式表頭中的定義，不可附加條件或重複地執行。 不允許將巨集，加在用來檢查結構的關鍵字，或跳躍目標之上。定義巨集時，尚未做過這種檢查。

效果

無法在整個程式中使用檢查結構。

巢狀深度

在各子程式層級中，可設定至多 16 個檢查結構的巢狀深度。



檢查結構的

在解譯器模式（啟用為標準）中，藉由使用分支程式，能夠比用檢查結構更有效地縮短程式執行時間。

在預編譯的循環中，分支程式和檢查結構之間沒有差異存在。

顯示程式迴圈的目前單節

若程式迴圈中只有選取的單節執行，則程式迴圈**之前**的最後一個主要執行單節會如目前的單節顯示所示。

因此，目前的單節顯示中也可以看到已處理的選取單節，例如診斷用的單節，解碼的單一單節 SBL2 必需啟用。

參考資料

功能手冊，基本功能，章節：模式群組、通道、程式操作、重設回應 (K1) > 單一單節 > 使用隱性前置處理停止的解碼單一單節 SBL2

不含主要執行單節的研磨

在程式迴圈中若無主要執行單節經過程式設計，則迴圈會預處理直到迴圈條件滿足為止。

因此可能導致高度的使用率，對於顯示將有不良的影響。

可以將 STOPRE 指令或是 0 秒的停頓時間 G04 **插入**迴圈中作為因應對策。

補充條件

- 具有檢查結構元素的單節，無法被停用。
- 在具有檢查結構元素的單節中，不允許有跳躍標記（標籤）。
- 檢查結構會被解譯進行。當偵測到迴圈結尾，會允許在程序中找到的檢查結構進行迴圈開始搜尋。基於這個理由，在解譯器模式中，程式的單節結構不會被徹底檢查。
- 通常不建議混合使用檢查結構與分支程式。
- 會進檢查，以確保當處理到循環時，檢查結構為正確巢狀。

2.12.1 條件式陳述及分支（IF、ELSE、ENDIF）

條件式陳述： IF - 程式單節 - ENDIF

在程式單節 `IF` 及 `ENDIF` 之間使用的條件式陳述，只有在滿足條件時才會執行。

分支： IF - 程式單節_1 - ELSE - 程式單節_2 - ENDIF

使用分支，會執行兩個程式單節之一。

若滿足條件，會執行 IF 及 ELSE 之間的程式單節_1。

若條件不滿足，會執行 ELSE 及 ENDIF 之間的程式單節_2。

語法

條件式陳述

```
IF <條件>
程式單節                                    ;執行時機： <條件> == TRUE
ENDIF
```

分支

```
IF <條件>
   程式單節_1                               ;執行時機： <條件> == TRUE
ELSE
   程式單節_2                               ;執行時機： <條件> == FALSE
ENDIF
```

含義

IF：	引進條件式陳述或分支。
ELSE：	引進可供選擇的程式單節。
ENDIF：	標示條件式陳述或分支的結尾。
<條件>：	邏輯表達式被視為 TRUE 或 FALSE。

範例： 換刀子程式

程式碼	註解
PROC L6	換刀常式
N500 DEF INT TNR_AKTUELL	供有效 T 數字所用的變數
N510 DEF INT TNR_VORWAHL	預選擇 T 編號的變數
	決定當前刀具
N520 STOPRE	
N530 IF $P_ISTEST	在程式測試模式中...
N540 TNR_AKTUELL = $P_TOOLNO	... 從程式文字讀取「當前」刀具。
N550 ELSE	否則...
N560 TNR_AKTUELL = $TC_MPP6[9998,1]	... 已讀取了主軸的刀具。

程式碼	註解
N570 ENDIF	
N580 GETSELT（TNR_VORWAHL）	在主軸中讀取預選擇刀具之 T 號碼。
N590 IF TNR_AKTUELL <> TNR_VORWAHL	若預選擇刀具還不是當前刀具，那麼...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... 逼近刀具更換定位...
N610 M206	... 以及執行一個換刀。
N620 ENDIF	
N630 M17	

2.12.2 連續程式迴圈（LOOP，ENDLOOP）

無窮迴圈係用於無窮程式。 在迴圈結尾，永遠會有一個回到開始處的分支。

語法

```
LOOP
...
ENDLOOP
```

含義

LOOP：	初始化無窮迴圈。
ENDLOOP：	標記迴圈的結尾，以及造成傳回跳躍返回至迴圈開始的結果。

範例

程式碼
...
LOOP
MSG ("no tool cutting edge active")
M0
STOPRE
ENDLOOP
...

2.12.3 計數迴圈（FOR ... TO ...，ENDFOR）

當必須以執行的固定數字重複運算時，需使用計數迴圈。

語法

```
FOR <變數> = <初始值> TO <結束值>
...
ENDFOR
```

含義

FOR：	初始化計數迴圈。	
ENDFOR：	只要尚未達到計數結束值，就標記迴圈的結尾，並造成傳回跳躍到迴圈開始處的結果。	
<變數>：	計數變數，係從初始值漸增到結束值，且每次執行時以值「1」增加。	
	類型	INT 或 REAL **附註：** 例如，當 R 參數被程式設計供計數迴圈所用，則需使用 REAL 類型。 若計數變數為 REAL 類型，必須將其值去尾數，成為整數。
<初始值>：	計數初始值 條件： 起始值必須小於結束值。	
<完整比例值>：	計數結束值	

範例

範例 1： INTEGER 變數或作為計數變數的 R 參數

作為計數變數的 INTEGER 變數：

程式碼	註解
`DEF INT iVARIABLE1`	
`R10=R12-R20*R1 R11=6`	
`FOR iVARIABLE1 = R10 TO R11`	`；計數變數 = INTEGER 變數`
`R20=R21*R22+R33`	
`ENDFOR`	
`M30`	

作為計數變數的 R 參數：

程式碼	註解
R11=6	
FOR R10=R12-R20*R1 TO R11	；計數變數 = R 參數（實數變數）
R20=R21*R22+R33	
ENDFOR	
M30	

範例 2： 生產固定品質的零件

程式碼	註解
DEF INT WKPCCOUNT	；以「WKPCCOUNT」名稱，定義類型 INT 變數。
FOR WKPCCOUNT = 0 TO 100	；初始化計數迴圈。 「WKPCCOUNT」變數，從初始值「0」，朝結束值「100」增加。
G01 ...	
ENDFOR	；計數迴圈結尾
M30	

2.12.4 在迴圈起點處，附有條件的程式迴圈（WHILE，ENDWHILE）

對於 WHILE 迴圈，條件就在迴圈開始處。 只要滿足條件，就會執行 WHILE 迴圈。

語法

```
WHILE <條件>
...
ENDWHILE
```

含義

WHILE：	初始化程式迴圈。
ENDWHILE：	標記迴圈的結尾，以及造成傳回跳躍返回至迴圈開始的結果。
<條件>：	必須滿足條件，才會執行 WHILE 迴圈。

範例

程式碼	註解
...	
WHILE $AA_IW[DRILL_AXIS] > -10	；在下列情況下，呼叫 WHILE 迴圈： 供鑽孔座標軸所用的實際工件座標系統 WCS 設定點，必須大於-10。
G1 G91 F250 AX[DRILL_AXIS] = -1	
ENDWHILE	
...	

2.12.5 在迴圈結尾處，附有條件的程式迴圈（REPEAT，UNTIL）

對於 REPEAT 迴圈，條件就在迴圈結尾處。 REPEAT 迴圈會執行一次，並不斷重複直到滿足條件為止。

語法

```
REPEAT
...
UNTIL <意義>
```

含義

REPEAT：	初始化程式迴圈。
UNTIL：	標記迴圈的結尾，以及造成傳回跳躍返回至迴圈開始的結果。
<條件>：	必須滿足條件，REPEAT 迴圈才會不再執行。

範例

程式碼	註解
...	
REPEAT	；呼叫 REPEAT 迴圈。
...	
UNTIL ...	；檢查條件是否滿足。
...	

2.12.6 具有巢狀檢查結構的程式範例

程式碼	註解
LOOP	
IF NOT $P_SEARCH	;IF 無單節搜尋
G1 G90 X0 Z10 F1000	
WHILE $AA_IM[X] <= 100	;WHILE（設定點 X 軸 <= 100）
G1 G91 X10 F500	;鑽孔樣式
Z-5 F100	
Z5	
ENDWHILE	
ELSE	;ELSE 單節搜尋
MSG ("No drilling during block search")	
ENDIF	;ENDIF
$A_OUT[1] = 1	;下個鑽孔盤
G4 F2	
ENDLOOP	
M30	

2.13 一致性指令（INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM）

原則上，NC 通道可獨立執行其中啟動的程式，不受其模式群組的其他通道影響。不過若模式群組多個通道之中的多個程式參與工件加工，不同通道的程式順序就必須與下列的一致性指令協調。

需求

參與程式一致性的所有通道都必須屬於**相同的**模式群組：

MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<通道>] = <模式群組編號>

取代通道編號的通道名稱

取代通道編號，利用將通道名稱輸入至 MD20000 $MC_CHAN_NAME[<通道索引>] 做為程式一致性的預先定義程序之參數。在 NC 程式中使用的通道名稱必須首先啟用：

MD10280 $MN_PROG_FUNCTION_MASK，位元 1 = TRUE

語法

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

意義

INIT()：	在指定通道中，選擇待執行 NC 程式的預先定義程序
START()：	預先定義的程序用於啟動在個別通道中選取的程式
WAITM()：	在指定通道中，等待執行到等候標記的預先定義程序 在相同通道中，以 WAITM 設定指定的等候標記。上一個單節已透過精確停止終止。在同步後刪除等候標記。 各通道可同步設定 10 個標記。
WAITE()：	在一個或多個其他通道中，等待程式結尾的預先定義程序
WAITMC()：[1)]	在指定通道中，等待執行到等候標記的預先定義程序 與 WAITM 不同的是，唯有在其他通道尚未執行到等候標記時，才會啟動精確停止的軸煞車。

SETM()：[1)]	為通道一致性設定一個或多個等候標記的預先定義程序 自身通道的執行不受此影響。 通道重置及 NC 啟動之後，SETM 仍為有效。		
CLEARM()：[1)]	為通道一致性刪除一個或多個等候標記的預先定義程序 自身通道的執行不受此影響。 CLEARM() 刪除通道中的所有等候標記。 CLEARM(0) 只刪除等候標記「0」。 通道重置及 NC 啟動之後，CLEARM 仍為有效。		
<ChanNr>：	通道編號 不必指派本身通道的編號。		
	類型：	INT	
<Prog>：	絕對或相對路徑指定（**選配**）+ 程式名稱		
	類型：	STRING	
	有關路徑指定，請參閱： **參考資料** 編程手冊，工作規劃，章節「檔案與程式管理」>「程式記憶體」>「程式記憶體的檔案定址」 (頁 237)		
<AckMode>：	確認模式（**選配**）		
	類型：	CHAR	
	值：	「N」	無確認 指令傳送之後，程式會繼續執行。若指令無法成功執行，傳送者不會收到通知。
		「S」	同步確認 程式執行會停止，直到指令確認接收元件。若確認結果為肯定，將執行下一指令。若確認結果為否定，則輸出錯誤訊息。

<MarkNr> ：	等候標記的編號 **附註** 在多重通道系統中，提供最多 100 個等候標記（等候標記 0...99）。 單通道系統只提供等候標記 0。
1) 對於使用者專屬通訊及 / 或通道的一致性，可以使用 SETM/CLEARM 部署等候標記，同時不需要使用條件式的等候指令 WAITMC。同時，在通道重置及 NC 啟動之後，等候標記也會保持其數值。	

範例

START 使用來自 MD20000 的通道名稱

- 參數化
  MD10280 $MN_PROG_FUNCTION_MASK，位元 1 = TRUE
  $MC_CHAN_NAME[0] = "MACHINING"；通道 1 名稱
  $MC_CHAN_NAME[1] = "INFEED"；通道 2 名稱
- 程式設計

程式碼	備註
START (MACHINING)	；通道 1 起點
START(INFEED)	；通道 2 起點

START 使用本機「通道名稱」及使用者變數

程式碼	備註
DEF INT MACHINE = 1	；通道 1 的使用者變數定義
DEF INT LOADER = 2	；通道 2 的使用者變數定義
...	
START(MACHINE)	；通道 1 起點
START(LOADER)	；通道 2 起點

START 使用本機「通道名稱」、使用者變數及參數化通道名稱

程式碼	備註
DEF INT chanNo1	；通道 1 的使用者變數定義
DEF INT chanNo2	；通道 2 的使用者變數定義
chanNo1 = CHAN_1	；指派參數化的通道 1 通道名稱
chanNo2 = CHAN_2	；指派參數化的通道 2 通道名稱
...	
START(chanNo1)	；通道 1 起點
START(chanNo2)	；通道 2 起點

具有絕對路徑規格的 INIT 指令

選擇通道 2 中的程式 /_N_MPF_DIR/_N_ABSPAN1_MPF。

程式碼

```
INIT(2,"/_N_WCS_DIR/_N_SHAFT1_WPD/_N_CUT1_MPF")
```

INIT 指令含程式名稱

選擇名稱為「MYPROG」的程式。控制會使用搜尋路徑搜尋程式。

程式碼

```
INIT(2,"MYPROG")
```

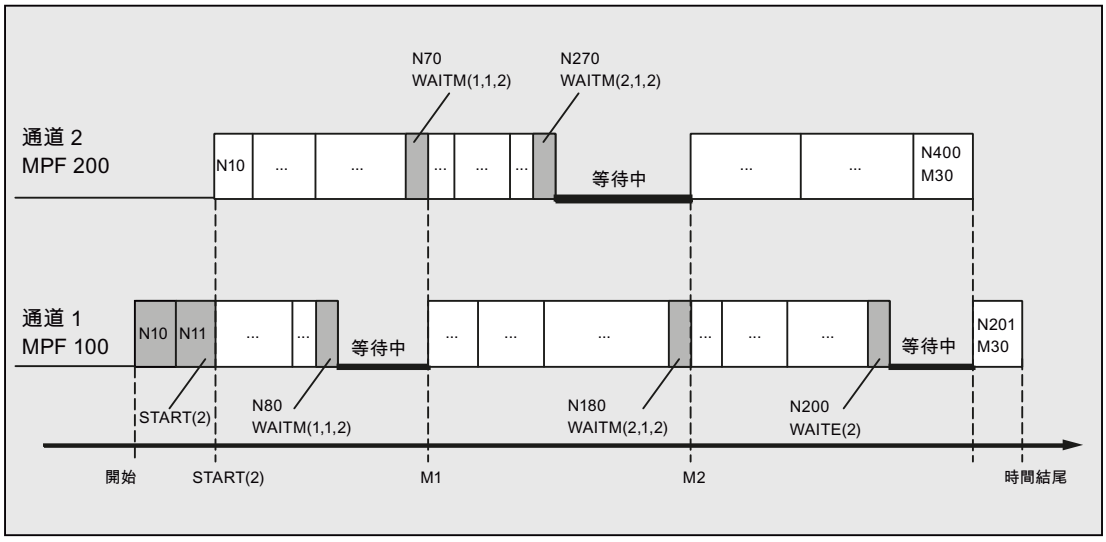
以 WAITM 編寫一致性

- 通道 1：已選取並啟動程式 /_N_MPF_DIR/_N_MPF100_MPF。

程式碼	備註
	；程式 MPF100
N10 INIT(2,"MPF200","N")	；選擇程式 MPF200，通道 2
N11 START(2)	；通道 2 起點
...	
N80 WAITM(1,1,2)	；等待通道 1 和 2 的 WAIT 標記 1
N81...	；通道 1 N81 及通道 2 N71 會 ；同步啟動
...	
N180 WAITM(2,1,2)	；等待通道 1 和 2 的 WAIT 標記 2
N181...	；通道 1 N181 及通道 2 N271 會 ；同步啟動
...	
N200 WAITE(2)	；等待通道 2 的程式結束
N201...	；N201 只會在程式結束後啟動 ；通道 2 中的 MPF200
N201 M30	；通道 1 程式結束

- 通道 2：在通道 1 中，會使用單節 N10 及 N20 針對通道 2 選擇及啟動的程式 MPF200_MPF。

程式碼	備註
;$PATH=/_N_MPF_DIR	；程式 MPF200
...	
N70 WAITM(1,1,2)	等待通道 1 和 2 的 WAIT 標記 1
N71...	；通道 1 N81 及通道 2 N71 會 ；同步啟動
...	
N270 WAITM(2,1,2)	等待通道 1 和 2 的 WAIT 標記 2
N271...	；通道 1 N181 及通道 2 N271 會 ；同步啟動
...	
N400 M30	通道 2 程式結束



補充條件

在 WAIT 標記之後非同步啟動執行隨後的單節

在使用 WAIT 標記進行通道一致性的情況下，執行以下單節時，可能會發生非同步啟動。如果在到達共同 WAIT 標記前立即在其中一個通道觸發動作，就會產生這項行為，因而造成在刪除剩餘距離時，產生隱性重新定位 (REPOSA) 的結果。

假設：在通道 1 和 2 目前的軸指派

- 通道 1：軸 X1 及 U
- 通道 2：軸 X2

表格 2-2　通道 1 和 2 的時間順序

通道 1	通道 2	說明
...	...	通道 1 和 2 的任意處理
N100 WAITM(20,1,2)		通道 1：到達 WAIT 標記，等待與通道 2 同步
開始 GETD(U) 處理： • *軸交換* • *刪除剩餘距離* • *REPOSA* *結束*	N200 GETD(U)	通道 2：從通道 1 要求軸 U 通道 1：在背景處理 GET(U)
	N210 WAITM(20,1,2)	通道 2：抵達 WAIT 標記。⇒ 可以完成通道 1 和 2 的同步
	N220 G0 X2=100	通道 2：開始處理 N220
N110 G0 X1=100		通道 1：交錯啟動處理 N110

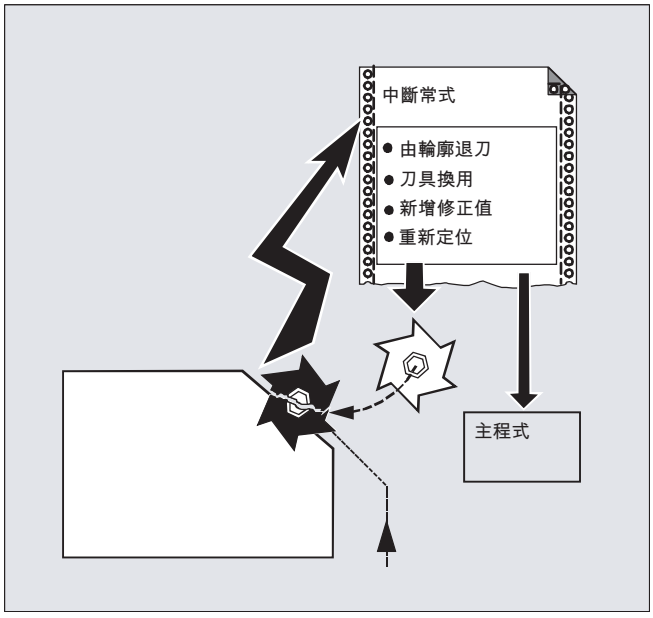
2.14 中斷常式（ASUB）

2.14.1 中斷常式的函數

說明

詞語"非同步子程式（ASUB）"與"中斷常式"，在底下的說明中可交替使用，以參考相同的功能。

典型的範例，應能解釋中斷常式的函數：



刀具在加工時停止。 這會觸發訊號來停止目前的加工處理，並同步啟動一個子程式 - 所謂的中斷常式。 中斷常式包含了所有在此情況中，有待執行的敘述。

當中斷常式執行完畢後，且機台已準備好繼續操作，則控制會跳躍返回至主程式並繼續在中斷處加工－視 REPOS 指令（請參考「 重新定位輪廓 (頁 584) 」）而定。

⚠ 小心

碰撞風險

若在子程式中，還沒執行到 REPOS 指令，那麼控制會到跟在中斷的單節後面的單節結束點。

參考資料

功能手冊，基本功能；模式群組，通道，程式操作，重置回應 (K1)，章節：「非同步子程式 (ASUB)，中斷常式」

2.14.2 建立一個中斷常式

建立中斷常式作為子程式

在此定義中，中斷常式係定義為子程式。

範例：

程式碼	註解
PROC LIFT_Z	；程式名稱「ABHEB_Z」
N10 ...	；NC 單節隨後跟著：
...	
N50 M17	；最後請結束程式，並返回主程式。

儲存模態 G 指令 (SAVE)

藉由以 SAVE 定義，來標明中斷常式。

SAVE 屬性表示在呼叫中斷常式之前，就先儲存了主動模態 G 指令，並在中斷常式結尾之後，可以重新啟用（請參考「具有 SAVE 機制 (SAVE) 的子程式 (頁 185)」）。

這表示在中斷常式結束後，可以在中斷點重新開始處理。

範例：

```
程式碼
PROC LIFT_Z SAVE
N10 ...
...
N50 M17
```

指派其他的中斷常式（SETINT）

SETINT 陳述式可以在中斷常式的範圍之內被程式設計（請參考「指派並啟動中斷常式（SETINT）」 (頁 140)），因此可以啟動其他的中斷常式。他們係透過輸入來觸發。

參考資料

您將在「子程式，巨集」章節中發現更多關於如何建立子程式的資訊。

2.14.3 指派並啟動中斷常式（SETINT，PRIO，BLSYNC）

控制具有多個快速輸入（輸入 1 ... 8）起始中斷（1 ... 8）。使用 SETINT 指令可指派每個中斷的優先順序及中斷程序。若中斷是由設定快速輸入所起始，則此通道的處理就會中斷，並啟動中斷程序。

中斷優先順序

若在工件程式中有多個輸入指派中斷，就必須針對中斷指定不同的優先順序。

可針對中斷指派 1 至 128 的優先順序值。優先順序值 1 對應最高的優先順序，128 則為最低。

語法

```
SETINT(<n>) <NAME>
SETINT(<n>) PRIO=<值> <名稱>
SETINT(<n>) PRIO=<值> <名稱> BLSYNC
SETINT(<n>) PRIO=<值> <名稱> LIFTFAST
```

含義

SETINT (<n>)：	針對輸入 **<n>** 指派中斷程序 **<**名稱**>**。只要偵測到輸入 **<n> == 1**，就會立即啟動指派的中斷程序。 **附註：** 若針對已程式設計的輸入 **<n>** 指派其他中斷程序，則之前指派就不再有效。	
<n>：	輸入號碼	
	類型：	INT
	值域：	1 … 8
PRIO= ：	中斷優先順序 （選擇性）	
<值>：	優先順序值 （選擇性）	
	類型：	INT
	值域：	1 ... 128（1 ⇒ 最高優先順序）

<名稱>：	中斷程序名稱（子程式）
BLSYNC：	BLSYNC 可在起始中斷後，確保系統一定會先等待，直到實際單節完成為止。然後才會執行中斷程序。 （選擇性）
LIFTFAST：	LIFTFAST 可在起始中斷之後，確保產生最初的快速回退（請參閱「從輪廓快速回退（SETINT LIFTFAST，ALF）(頁 143)」章節）。然後才會執行中斷程序。 （選擇性）

補充條件

中斷規則

1. 對於無法立即執行，或目前正在處理的每個中斷，會儲存額外的中斷要求。此中斷的所有其他中斷要求都會遺失。
2. 若中斷目前正在處理，並起始更高優先順序的其他中斷，就會中斷較低優先順序的中斷。優先順序較高的中斷完成後，優先順序較低的中斷就會繼續進行。若較高優先順序中斷處理期間，收到較低優先順序中斷的其他要求，就會儲存一項要求。所有其他要求將遺失。
3. 若中斷目前正在處理，並起始更高優先順序的其他中斷，就會中斷較低優先順序的中斷。優先順序較高的中斷將進行處理。若起始較高優先順序的中斷，就會中斷實際中斷，並處理較高優先順序的中斷。最多可有六個生效中斷層級。其中一個是目前正在處理的中斷層級，另五個則是等待中的中斷層級。每個生效的中斷層級，最多可儲存一個額外中斷要求。所有其他中斷要求都將遺失。如果要求額外的中斷層級（中斷層級 ≥ 7），中斷要求也會遺失。

範例

範例 1：指派中斷常式並定義優先順序

程式碼	註解
...	
N20 SETINT(3) PRIO=1 ABHEB_Z	;IF 輸入 3 == 1 THEN 啟動中斷程序「ABHEB_Z」
N30 SETINT(2) PRIO=2 ABHEB_X	;IF 輸入 2 == 1 THEN 啟動中斷程序「ABHEB_X」。
...	

若輸入同步變成可用（同步啟用），中斷常式會以優先值的順序來執行，：第一「ABHEB_Z」，然後「ABHEB_X」。

範例 2：新指派中斷常式

程式碼	註解
...	
N20 SETINT(3) PRIO=2 ABHEB_Z	;IF 輸入 3 == 1 THEN 啟動中斷程序「**ABHEB_Z**」

程式碼	註解
...	
N80 SETINT(3) PRIO=1 ABHEB_X	；IF 輸入 3 == 1 THEN 啟動中斷程序「**ABHEB_X**」
...	

2.14.4 停用 / 重新啟用指派中斷常式（DISABLE，ENABLE）

SETINT 陳述式，可以 DISABLE 來將其停用，並可以 ENABLE 將其重新啟用，而不會失去輸入 → 中斷常式指派。

語法

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

含義

DISABLE(<n>)：	指令：**停用**輸入的中斷常式指派 **<n>**	
ENABLE(<n>)：	指令：**重新啟用**輸入的中斷常式指派 **<n>**	
<n>：	參數：中斷訊號編號	
	類型：	INT
	值域：	1 ... 32

範例

程式碼	註解
N20 SETINT(3) PRIO=1 ABHEB_Z	；若輸入 3 開關，則中斷 ；常式「ABHEB_Z」應啟動。
...	
N90 DISABLE(3)	；從 N20 來的 SETINT 陳述式停用。
...	
N130 ENABLE(3)	；從 N20 來的 SETINT 陳述式重新啟用。
...	

2.14.5 刪除指派的中斷常式（CLRINT）

使用 NC 程式 (ASUP) 的 SETINT 定義的中斷訊號指定可用 CLRINT 加以刪除。

語法

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

含義

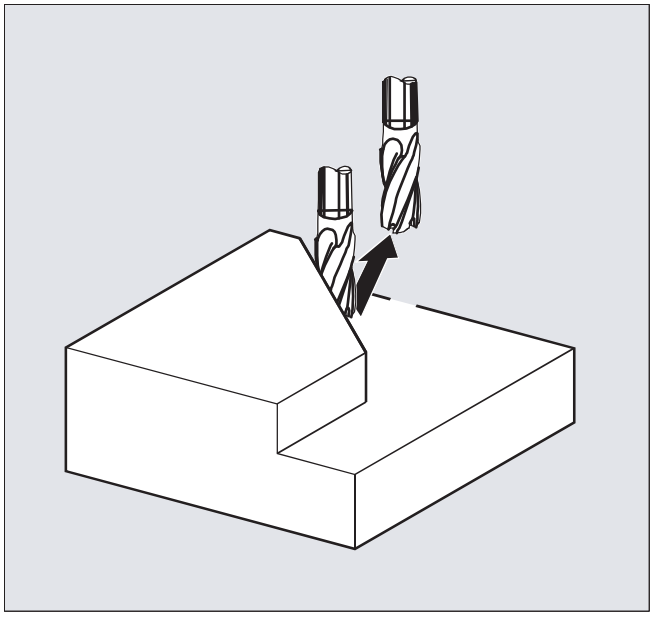
CLRINT(<n>)：	指令：刪除使用 SETINT <n> 定義指定至 NC 程式 (ASUP) 的中斷訊號 <n>	
<n>：	參數：中斷訊號編號	
	類型：	INT
	值域：	1 ... 32

範例

程式碼	註解
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	；刪除介於輸入「3」和中斷常式「ABHEB_Z」之間的指派。

2.14.6 從輪廓快速回退（SETINT LIFTFAST，ALF）

對於 SETINT 敘述，其具有 LIFTFAST，切換輸入時，使用快速回退，刀具從工件輪廓退出。



後續順序是，依照中斷常式是否包含 SETINT 敘述，除了 LIFTFAST 之外，而有所不同：

具有中斷常式：　　在快速回退**之後**，會執行中斷常式。
不具有中斷常式：　　在快速回退之後，停止加工，並輸出警報。

語法

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

含義

SETINT (<n>)：	指令： 指派輸入**<n>**至一個中斷常式。 所指派的中斷常式，會在輸入**<n>**開關時啟動。	
<n>：	參數： 輸入號碼	
	類型：	INT
	值域：	1 ... 8
PRIO= ：	定義優先順序	
<值>：	優先順序值	
	值域：	1 ... 128
	優先順序 1 表示最高優先權。	
<名稱>：	將執行的子程式（中斷常式）名稱。	
LIFTFAST：	指令： 從輪廓快速退刀	
ALF=... ：	指令： 可程式設計移動方向（在動作單節中） 關於以 ALF 進行程式設計的可能性，請參考「從輪廓快速回退的移動方向 (頁 146)」。	

補充條件

具有主動框架的動作

當決定回退方向，會檢查看看是否有鏡像框架啟用中。 在這種情況中，回退方向會參考切線方向，而左右交換。 刀具方向中的元件方向，並非鏡像。 此行為會以使用 MD 設定來啟用：

MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

範例

破損的刀具應當自動被備用刀具替換。 隨後會繼續以新的刀具加工。

主程式：

主程式	註解
N10 SETINT（1） PRIO=1 W_WECHS LIFTFAST	；切換為輸入 1，刀具會立即以快速回退的方式，從輪廓回退（代碼為 7 號，供刀具半徑補正 G41 所用）。 執行中斷常式「W_WECHS」。
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

子程式：

子程式	註解
PROC W_CHANGE SAVE	；儲存實際運算狀態的子程式
N10 G0 Z100 M5	；換刀定位，主軸停止
N20 T11 M6 D1 G41	；變更刀具
N30 REPOSL RMBBL M3	；在輪廓處重新定位，並跳躍返回至主程式（這是在單節中的程式設計）

2.14.7 從輪廓快速回退的移動方向

回退移動

下列 G 指令定義了回退移動平面：

- LFTXT
  回退移動平面係由路徑切線和刀具方向（預設設定）來決定。
- LFWP
  退刀移動平面，是用 G 指令 G17、G18 或 G19 所選擇的主動工作平面。回退移動的方向，不相依於路徑切線。這允許和座標軸平行的快速回退可做程式設計。
- LFPOS
  已宣告的座標軸的回退，使用 POLFMASK/POLFMLIN，對以 POLF 進行程式設計的絕對座標軸定位。
  ALF 對於數個軸和在線性系統中的數個軸，其回退方向不具有影響。
  參考資料：
  基礎程式設計手冊，章節：「螺紋切削時的快速退刀」

可程式設計移動方向（ALF=...）

程式設計的方向，會在回退移動平面中採用 ALF，以 45 度的離散步驟進行。

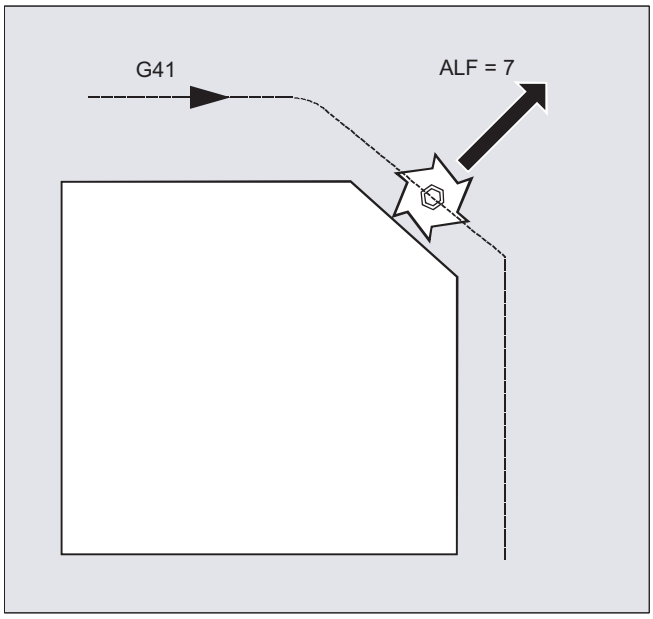
可能的移動方向會儲存在控制上的特殊代碼數字中，並可使用三個數字來呼叫。

範例：

程式碼

```
N10 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST
ALF=7
```

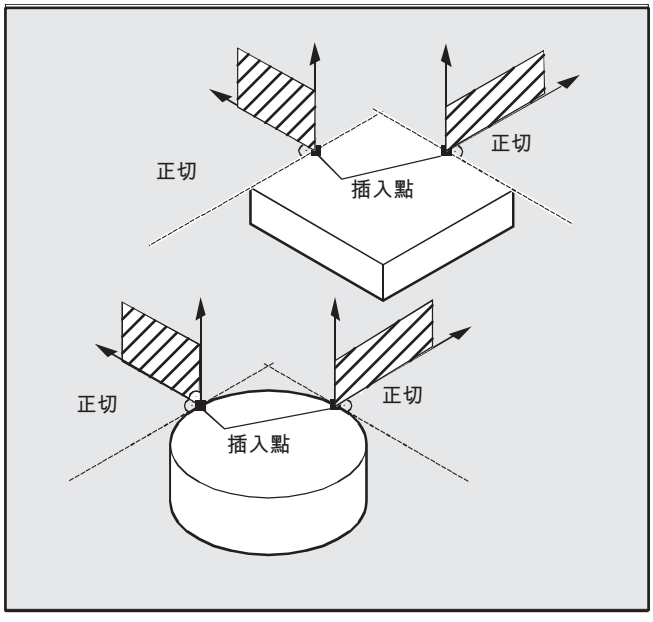
隨著 G41 啟動（對於左側輪廓的加工方向）刀具從輪廓垂直移出。



用來定義移動方向 LFTXT 的參考平面

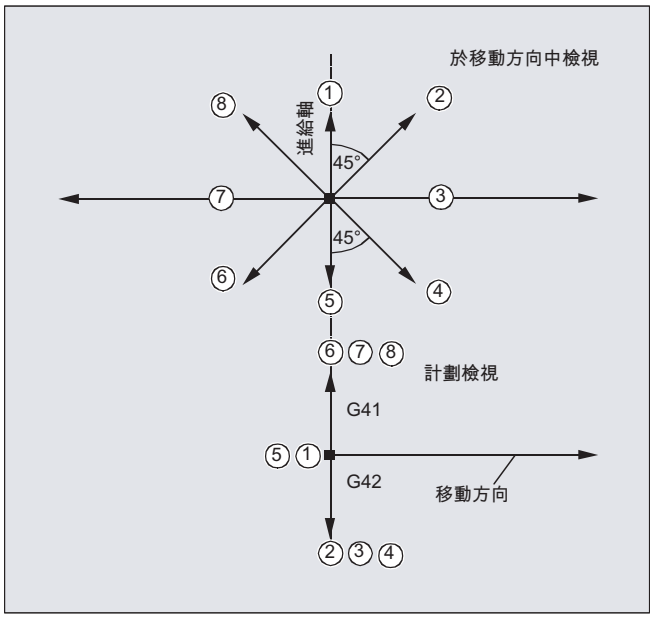
在將刀具應用在程式設計的輪廓處，刀具會被夾在用來作為參考的平面，以便具體指定以對應代碼數字所回退的移動。

參考平面係從縱向刀具座標軸（進給方向）而來，並有一個向量對準此軸，以及對準應用刀具的切線來定位。



具有供 LFTXT 所用的移動方向之代碼數字

從參考平面開始，您將在下列圖示中，發現具有移動方向的代碼數字。



刀具的回退方向係定義為 ALF=1。

「快速回退」函數係以 ALF=0 停用。

⚠ 小心

碰撞風險

當刀具半徑補正啟用時，則：

- 對於 G41 代碼 2、3、4
- 對於 G42 代碼 6、7、8

在這種情況下**不可以**使用，刀具會移動到輪廓，且會與工件抵觸。

具有供 LFWP 所用的移動方向之代碼數字

使用 LFWP 時，工作平面中的方向從以下配置中衍生而來：

- G17：X/Y 平面
  ALF=1：X 方向之回退
  ALF=3：Y 方向之回退
- G18：Z/X 平面
  ALF=1：Z 方向之回退
  ALF=3：X 方向之回退
- G19：Y/Z 平面
  ALF=1：Y 方向之回退
  ALF=3：Z 方向之回退

2.14.8 供中斷程式所用的順序

不具有 LIFTFAST 的中斷程式

座標軸的動作係沿著路徑減速並停止（零速度）。 隨後啟動中斷程式。

靜止位置會儲存為中斷位置，並且以 RMIBL 在中斷程式結尾逼近 REPOS。

具有 LIFTFAST 的中斷程式

座標軸的動作係沿著路徑減速。 LIFTFAST 動作會同步執行，作為覆蓋動作。 若路徑動作與 LIFTFAST 動作已靜止（零速），則會啟動中斷程式。

輪廓上的位置會儲存起來作為中斷位置，該中斷位置就是 LIFTFAST 動作的開始處，也是路徑離開之處。

具有 LIFTFAST 和 ALF=0 的中斷程式，和不具 LIFTFAST 的中斷程式，以同樣精確的方式運作。

說明

使用機械參數，能將從輪廓快速回退時，幾何軸移動的絕對值，加以設定。

2.15 座標軸替換，主軸替換（RELEASE，GET，GETD）

一或多個軸或主軸，僅能在一個通道中插補。 若座標軸必須要從兩個不同的通道之間做出選擇（例如，工作台交換），則必須先在目前的通道中啟用，隨後傳輸至其他通道。 座標軸替換在兩個通道中有效。

座標軸替換副檔名

欲替換座標軸 / 主軸，若非以介於前置處理和主程式之間的前置處理停止和同步，就是不能使用前置處理停止。 座標軸替換，也可能可以透過：

- 座標軸空間旋轉 AXCTSWE 或 AXCTWED 使用內建的 GET/GETD
- 若此程序將座標軸和其他軸連結，則框架能旋轉。
- 同步動作，請參考動作－同步動作，「座標軸替換 RELEASE，GET」。

機台製造商

請參閱機台製造商說明。 若要達成座標軸替換的目的，必須要在配置有機械資料的所有通道中，各自定義一個座標軸，並使用機械資料，來設定座標軸替換特性。

語法

```
RELEASE （axis name，axis name，...） 或 RELEASE （S1）
GET （axis name，axis name，...） 或 GET （S2）
GETD（axis name，axis name，...） 或 GETD（S3）
```

隨著 GETD（直接 GET），座標軸會直接從其他通道拿走。 這表示對於其他通道中的 GETD，沒有適當的 RELEASE，必須進行程式設計。 也表示其他通道通訊，有待建立（例如，等待標記）。

含義

RELEASE（軸名稱，軸名稱，等等）：	釋放座標軸（軸）
GET（軸名稱，軸名稱，等等）：	存取座標軸（軸）
GETD（軸名稱，軸名稱，等等）：	直接存取座標軸（軸）
軸名稱：	系統中的座標軸： AX1、AX2、...或特定機械軸名稱
RELEASE （S1）：	釋放主軸 S1、S2、...
GET（S2）：	接受主軸 S1、S2、...
GETD（S3）：	直接存取主軸 S1、S2、...

不具前置處理停止的 GET 要求

若跟在**不具**前置處理的 GET 要求之後，座標軸再次以 RELEASE（axis）或 WAITP（axis）啟用，則後續的 GET 會產生一個 GET，**具有**前置處理停止。

⚠ 小心
軸指派已變更 以 GET 接受的座標軸或主軸，仍然指派至此通道，即使在關鍵或程式 RESET 之後。 當程式重新啟動被替換的軸或主軸，若該座標軸仍須在原通道中使用，則必須在程式中重新指派。 指派給在 POWER ON 時，定義在機械資料中的通道。

範例

範例 1： 介於兩通道之間的座標軸交換

在六軸中，下列可用於通道 1 的加工： 1., 2., 3. 第一、第二、第三和第四軸。
第五和第六軸在通道 2 中，會用於工件更改。

在 POWER ON 後，座標軸 2 應當可在兩通道之間交換，並可指派給通道 1。

通道 1 中的程式「MAIN」：

程式碼	註解
INIT （2，"TRANSFER2"）	；在通道 2 中選擇程式 TRANSFER2。
N... START （2）	；在通道 2 中啟動程式。
N... GET （AX2）	；接受座標軸 AX2。
...	
N... RELEASE （AX2）	；釋放座標軸 AX2。
N... WAITM (1,1,2)	；等待通道 1 和 2 中的 WAIT 標記，用來在兩個通道中同步。
...	；在座標軸替換之後的其他程式。
N... M30	

通道 2 中的程式「TRANSFER2」：

程式設計	註解
N... RELEASE （AX2）	
N160 WAITM(1,1,2)	；等待通道 1 和 2 中的 WAIT 標記，用來在兩個通道中同步。
N150 GET（AX2）	；接受座標軸 AX2。
...	；在座標軸替換之後的其他程式。

程式設計	註解
N... M30	

範例 2： 不具同步的座標軸替換

若座標軸不必同步，則 GET 不會產生任何前置處理。

程式設計	註解
N01 G0 X0	
N02 RELEASE（AX5）	
N03 G64 X10	
N04 X20	
N05 GET（AX5）	；若不需要同步，那麼就無法執行這個單節。
N06 G01 F5000	；無法被執行的單節。
N07 X20	；無法被執行的單節，原因是 N04 中的 X 定位。
N08 X30	；在 N05 之後第一個可以執行的單節。
...	

範例 3： 啟動不具有前置處理的座標軸交換

需求： 不具有前置處理的座標軸替換，必須要透過機械資料來配置。

程式設計	註解
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP（B）	；座標軸 B 成為中立座標軸。
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	；座標軸並未觸發前置處理停止/REORG
N041 G4 F2	
N050 M5	
N099 M30	

例如，若主軸或座標軸 B 以 180 度來移動，並在單節 N023 後立即退回 1 度，作為 **PLC 座標軸**，此軸將返回其中立狀態，並不會在單節 N40 中觸發前置處理停止。

其他資訊

座標軸替換的需求

- 該座標軸必須在所有通道中被定義，且該通道必須在機械資料中使用到這些座標軸。
- POWER ON 之後，在**軸**專屬機械資料之中，必須要定義座標軸將指派至哪一個通道。

說明

釋放座標軸： RELEASE

當啟用了座標軸，請注意：

1. 此軸不可以被包含在轉換中。
2. 所有包含在座標軸連結（切線控制）中的軸，必須被啟用。
3. 在此情況中，無法替換並行定位軸。
4. 門型軸的所有跟隨軸，會以第一主要軸傳輸。
5. 至於耦合軸（耦合動作，主動值耦合，電動齒輪），只有群組的引導軸可以啟用。

接受座標軸： GET

實際的座標軸替換會以此指令執行。 用來執行指令的通道，必須對座標軸負起全責。

GET 的效果：

同步進行座標軸替換：

若軸同時指派給了其他通道或 PLC，且在 GET 之前，尚未以「WAITP」，G74 或刪除剩餘距離，進行再同步，則該軸必須永遠被同步。

- 接著出現一個前置處理停止（STOPRE）。
- 會中斷執行，直到完成替換。

自動「GET」

若通道中基本上有一個座標軸，但目前沒有定義為「通道軸」，則會自動執行 GET。 若座標軸 / 軸已經同步，則不會產生前置處理停止。

驗證座標軸替換行為

可以使用機械資料，將軸的轉換點設定為：

- 當座標軸已經被 WAITP（像之前那樣回應）還原成中立狀態，則隨後會發生介於兩個通道之間的自動座標軸替換
- 當要求一個座標軸空間旋轉，則可被指定給執行通道的座標軸空間其所有的軸，同時會被內建的 GET 或 GETD 帶入通道。 一旦完成了座標軸空間旋轉，則只會認可後續的座標軸替換。
- 當在主程式中插入一個中間單節，則會進行檢查，以決定是否需要重新制訂。 唯有當此單節的座標軸狀態，**不符合**目前的座標軸狀態時，才需進行重新制訂。
- 以介於前置處理和主程式之間的前置處理停止和同步，取代 GET 單節，則不需前置處理停止，也能進行軸替換。 在此情況中，會以 GET 要求，產生一個中間單節。 在主程式中，當執行了此單節，則系統會檢查單節中的軸狀態是否符合目前的座標軸狀態。

若要更多有關座標軸或主軸替換如何運作的資訊，請參考
功能手冊，延伸函數，模式群組，通道，座標軸替換 (K5)。

2.16 將座標軸傳輸至其他通道（AXTOCHAN）：

AXTOCHAN 語言指令可用來要求一個軸，將其移動到其他通道。 此座標軸，可以從 NC 工件程式，也可以從同步動作，來移動到對應的通道。

語法

```
AXTOCHAN（axis name，channel number[，axis name，channel
number[，...]]）
```

含義

元件	說明
AXTOCHAN：	要求座標軸作為特定通道
軸名稱：	系統中的座標軸： 與 X、Y、... 或機械軸名稱的輸入有關。 執行通道不必是同一個通道，或甚至可以是目前正在處理插補供座標軸所用的通道也行。
通道號碼：	通道號碼對應至那個有待指派的座標軸

說明

競爭定位座標軸和 PLC 控制座標軸互斥

PLC 座標軸無法將通道替換為競爭定位座標軸。 被 PLC 互斥控制的座標軸，無法被指派給 NC 程式。

參考資料：
功能手冊，延伸功能；位置軸（P2）

範例

NC 程式內的 AXTOCHAN

軸 X 和 Y 已在第二通道中宣告。 目前，通道 1 具有插補，且後續程式在此通道中開始。

程式碼	註解
N110 AXTOCHAN（Y，2）	；將 Y 座標軸移動至第二通道
N111 M0	
N120 AXTOCHAN（Y，1）	；重新取回 Y 座標軸（中立）。
N121 M0	
N130 AXTOCHAN（Y，2，X，2）	；將 Y 座標軸和 X 座標軸移動至第二通道（軸為中立）。
N131 M0	
N140 AXTOCHAN（Y，2）	；將 Y 座標軸移動至第二通道（NC 程式）。

程式碼	註解
N141 M0	

其他資訊

NC 程式內的 AXTOCHAN

只有在座標軸被相同通道中的 NC 程式要求時，才會執行 GET。（這表示系統等待狀態會實際改變）。 若是其他通道要求座標軸，或是座標軸在相同通道中變成中立軸，則會送出此要求。

從同步動作而來的 AXTOCHAN

當座標軸被相同通道要求時，從同步動作而來的 AXTOCHAN，會對應至從同步動作而來的 GET。 在此情形中，座標軸會在第一次被相同通道邀請時，成為中立座標軸。 第二次要求時，座標軸會以相同方式被指派至 NC 程式，作為在 NC 程式之中的 GET 要求。 若要更多關於從同步動作而來的 GET 要求之資訊，請參考“動作－同步動作”。

2.17 啟動機械參數（NEWCONF）

NEWCONF 指令啟動所有機械資料。在 HMI 使用者介面中，也能藉由按下「MD 資料有效」軟鍵，來啟用函數。

執行「NEWCONF」函數時，會有一個內建的前置處理停止，亦即表示中斷了路徑移動。

語法

```
NEWCONF
```

含義

NEWCONF：	用來設定所有有效層級的「NEW_CONFIG」的機械資料，皆被設定為有效

從工件程式，跨通道執行 NEWCONF

若對工件程式中的機械資料進行變更，接著使用 NEWCONF 啟用，那麼 NEWCONF 將只會啟用包含了對工件程式通道會產生影響之變更的機械資料。

說明

為了確保已套用所有變更，必須在所有通道中執行 NEWCONF 指令，而通道中受到機械資料變更影響的軸或函數正在進行計算。

軸機械資料對 NEWCONF 無效。

軸的 RESET，必須對受 PLC 控制的軸執行。

範例

銑削：以不同技術進行機台鑽頭定位

程式碼	註解
N10 $MA_CONTOUR_TOL[AX]=1.0	；變更機械資料。
N20 NEWCONF	；啟動機械資料。
...	

2.18 寫入檔案（WRITE）

WRITE 指令可由 NC 程式將資料集 / 資料寫入被動檔案系統的檔案結尾（日誌檔）或外部程式記憶體， 或是目前正在執行的程式。

說明

如果程式記憶體沒有存在這種檔案，則會建立一個，並可使用 WRITE 指令將其寫入。

需求條件

目前的設定保護等級，必須等於或大於檔案的 WRITE 權限。 在此情形中，會以錯誤訊息（錯誤變數的傳回值=13）來否決存取。

語法

```
DEF INT<錯誤>
...
WRITE(<錯誤>,"<檔案名稱>"/"<ExtG>","<資料組/資料>")
```

含義

寫入：	用來在特定檔案結尾，附加一個單節或資料的指令。		
<錯誤>：	**參數 1：** 用來傳回錯誤值的變數		
	類型：	INT	
	值：	0	無錯誤
		1	路徑未獲得允許
		2	找不到路徑
		3	找不到檔案
		4	不正確的檔案類型
		10	檔案已滿
		11	檔案使用中
		12	無可用資源
		13	無存取權限
		14	輸出裝置 EXTOPEN 遺失或失敗
		15	寫入外部裝置時錯誤
		16	程式設計無效的外部路徑
<檔案名稱>：	**參數 2：** 要被特定單節或特定資料加入的檔案名稱。		
	類型：	STRING	
	絕對路徑可於實際檔案名稱之前指定。 若不指定特定路徑，則會在目前的目錄中搜尋檔案（= 所選程式的目錄）。 路徑資料規則請參閱「定址程式記憶體檔案 (頁 237)」。		
<ExtG>：	若要使用「處理 DataShare」功能將資料輸出至外部裝置 / 檔案，則必須以符號識別碼取代檔案名稱，以開啟外部裝置 / 檔案。		
	類型：	STRING	
	如需進一步資訊，請參閱「Process DataShare - 輸出至外部裝置 / 檔案（EXTOPEN、WRITE、EXTCLOSE）： (頁 721)」。 **附註：** 識別碼必須與 EXTOPEN 指令指定的識別碼相同。		

<單節/資料>：	**參數 3：** 要被加入至特定檔案的區塊或資料。	
	類型：	STRING

說明

寫入至被動檔案系統或外部程式記憶體時，WRITE 指令會將「LF」字元（LINE FEED = 換行）隱性插入輸出字串的結尾。

這項行為不適用於使用「處理 DataShare」功能輸出至外部裝置 / 檔案。 如果也要輸出 "LF"，則必須在輸出字串中明確指定。

→ 亦請參考範例 3： 隱性/明確 "LF"！

補充條件

- **最大檔案大小（機台製造商）**
  被動檔案系統中，日誌檔最大檔案的大小係以機械資料設定：
  MD11420 $MN_LEN_PROTOCOL_FILE
  最大檔案長度可用於所有以 WRITE 指令建立在被動檔案系統中的檔案。 若超過，會輸出一個錯誤訊息，且該單節或資料不會被儲存。 若自由記憶體足夠，則可建立一個新的檔案。

範例

範例 1： 以 WRITE 指令寫入被動檔案系統，無絕對路徑資料

程式碼	註解
N10 DEF INT ERROR	；錯誤變數的定義。
N20 WRITE(ERROR,"PROT","LOG FROM 7.2.97")	；將文字「LOG FROM 7.2.97」寫入至檔案 _N_PROT_MPF。
N30 IF ERROR	；錯誤評估。
N40 MSG ("Error with WRITE command:" <<ERROR)	
N50 M0	
N60 ENDIF	
...	

範例 2： 以 WRITE 指令寫入被動檔案系統，有絕對路徑資料

程式碼
...
WRITE（ERROR，"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF"，"LOG FROM 7.2.97"）

程式碼

```
...
```

範例 3： 隱性/明確 "LF"

a) 以隱性產生的「LF」寫入被動檔案系統

程式碼

```
...
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
```

輸出結果：

MY_STRING

MY_STRING

b) 寫入外部檔案，無隱性產生的「LF」

程式碼

```
...
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
```

輸出結果：

MY_STRINGMY_STRING

c) 以隱性產生的「LF」寫入外部檔案

若要達到與 a 相同的結果，必須如下程式設計：

程式碼

```
...
N200 DEF STRING[30] DEV_1
```

程式碼

```
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
```

輸出結果：

MY_STRING

MY_STRING

2.19 刪除檔案（DELETE）

DELETE 指令可刪除所有檔案，不論這些檔案是否使用 WRITE 指令建立。 使用較高存取權限來建立的檔案，也能用 DELETE 來刪除。

語法

```
DEF INT<錯誤>
DELETE（<錯誤>，"<檔案名稱>"）
```

含義

DELETE：	用來刪除特定檔案的指令。		
<錯誤>：	用於傳回錯誤值的變數。		
	類型。	INT	
	值：	0	無錯誤
		1	不認可的路徑
		2	找不到路徑
		3	找不到檔案
		4	不正確的檔案類型
		11	檔案使用中
		12	無可用資源
		20	其他錯誤
<檔案名稱>：	待刪除之檔案的名稱		
	類型：	STRING	
	絕對路徑可於實際檔案名稱之前指定。 若不指定特定路徑，則會在目前的目錄中搜尋檔案（= 所選程式的目錄）。 路徑資料規則請參閱「定址程式記憶體檔案 (頁 237)」。		

範例

程式碼	註解
N10 DEF INT ERROR	；錯誤變數的定義。
N15 STOPRE	；前置處理停止。
N20 DELETE（ERROR，"/_N_SPF_DIR/_N_TEST1_SPF"）	；在子程式目錄刪除檔案 TEST1。
N30 IF ERROR	；錯誤評估。
N40 MSG（"error for DELETE command：" <<ERROR)	

程式碼	註解
N50 M0	
N60 ENDIF	

2.20 在檔案中讀入行（READ）

READ 指令會在特定檔案中，讀入一或多行，並儲存陣列類型 STRING 的資訊。 在此陣列中，每一個讀入行佔用了一個陣列元素。

需求條件

目前的設定保護等級，必須等於或大於檔案的 READ 權限。 在此情形中，會以錯誤訊息（錯誤變數的傳回值=13）來否決存取。

語法

```
DEF INT<錯誤>
DEF STRING[<字串長度>] <結果>[<n>,<m>]
READ（<錯誤>，"<檔案名稱>"，<起始指令列>，<指令列數>，<結果>）
```

含義

讀取：	用於從特定檔案中讀取指令列並將這些指令列儲存在變數陣列中。		
<錯誤>：	用於傳回錯誤值的變數（傳址呼叫參數）		
	類型。	INT	
	值：	0	無錯誤
		1	不認可的路徑
		2	找不到路徑
		3	找不到檔案
		4	不正確的檔案類型
		11	檔案使用中
		13	存取權限不足
		21	指令列不存在（<起始指令列>或<指令列數>參數超過指定檔案中的指令列數）。
		22	結果變數（<結果>）的欄位長度過短。
		23	指令列範圍過大（選擇了<指令列數>個參數故讀取時會超出檔案的終點）。

<檔案名稱>：	欲讀取之檔案的名稱（傳值呼叫參數）		
	類型：	STRING	
	絕對路徑可於實際檔案名稱之前指定。若不指定特定路徑，則會在目前的目錄中搜尋檔案（= 所選程式的目錄）。 路徑資料規則請參閱「定址程式記憶體檔案 (頁 237)」。		
<起始指令列>：	要讀取之檔案區段的起始指令列（傳值呼叫參數）		
	類型：	INT	
	值：	0	在結束檔案前讀取以<指令列數>參數指定的指令列數。
		1 到 n	待讀取的第一行數字。
<指令列數>：	欲讀取之指令列數（傳值呼叫參數）		
	類型：	INT	
<結果>：	結果變數（傳址呼叫參數） 儲存了讀取之文字的變數陣列。		
	類型：	字串（最大長度： 255)	
	若在<指令列數>參數中指定的列數小於結果變數的陣列大小[<n>，<m>]，則不會修改剩餘的陣列元素。 藉由控制字元「LF」（換行）或「CR LF」（歸位換行），所終止的行，**不會**儲存在變數「結果」之中。 當讀取的指令列較定義的字串長度更長時則會進行裁切。 未輸出錯誤訊息。		

說明

無法讀入二元檔案。會輸出"資料類型不正確"錯誤訊息（錯誤變數的傳回值= 4）。 下列類型的檔案無法讀取： _BIN，_EXE，_OBJ，_LIB，_BOT，_TRC，_ACC，_CYC，_NCK.

範例

程式碼	註解
N10 DEF INT ERROR	；錯誤變數的定義。
N20 DEF STRING[255] RESULT[5]	；結果變數的定義。
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF", 1,5,RESULT)	；有域及檔案識別碼的檔名及路徑名稱。
N40 IF ERROR <>0	；錯誤評估。
N50 MSG("ERROR"<<ERROR<<"ON READ COMMAND")	
N60 M0	

程式碼	註解
N70 ENDIF	
...	

2.21 確認檔案（ISFILE）是否存在

ISFILE 指令可檢查檔案是否存在於程式記憶體。

語法

<結果>=ISFILE("<檔案名稱>")

含義

ISFILE：	檢查檔案可用性的指令		
<檔案名稱>：	要檢查可用性的檔案名稱。		
	類型：	STRING	
	絕對路徑可於實際檔案名稱之前指定。若不指定特定路徑，則會在目前的目錄中搜尋檔案（= 所選程式的目錄）。 路徑資料規則請參閱「定址程式記憶體檔案 (頁 237)」。		
<結果>：	用於指派檢查結果的結果變數。		
	類型。	BOOL	
	值：	TRUE	檔案存在
		FALSE	檔案不存在

範例

範例 1

程式碼	註解
N10 DEF BOOL RESULT	；結果變數的定義。
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(RESULT==FALSE)	
N40 MSG("檔案不存在")	
N50 M0	
N60 ENDIF	
...	

範例 2

程式碼	註解
N10 DEF BOOL RESULT	；結果變數的定義。
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(NOT ISFILE("TESTFILE"))	

程式碼	註解
N40 MSG("檔案不存在")	
N50 M0	
N60 ENDIF	
...	

2.22 讀出檔案資訊（FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO）

FILEDATE、FILETIME、FILESIZE、FILESTAT 及 FILEINFO 指令可讀出最後存取的日期/時間、目前檔案的大小、檔案狀態等特定檔案資訊或以上全部的資訊。

需求條件

目前的設定保護等級，必須等於或大於上級目錄的顯示權限。在此情形中，會以錯誤訊息（錯誤變數的傳回值=13）來否決存取。

語法

FILE....(<錯誤>,"<檔案名稱>",<結果>)

含義

FILEDATE:	回傳最近寫入檔案的**日期**
FILETIME:	回傳最近寫入檔案的**時間**
FILESIZE:	回傳檔案**目前的大小**
FILESTAT:	回傳以下相關的檔案**狀態權限**： • 讀取 (r: read) • 寫入 (w: write) • 執行 (x: execute) • 顯示 (s: show) • 刪除 (d: delete) **附註：** 這些保護等級為被動檔案系統的專屬特性。因此存取外部程式記憶體時，FILESTAT 會提供預設存取層級 (77777)。
FILEINFO:	提供**資訊總和**的檔案，可透過 FILEDATE, FILETIME、FILESIZE 及 FILESTAT 讀取

<錯誤>：	用於傳回錯誤值的變數（傳址呼叫參數）			
	類型。	VAR INT		
	值：	0	無錯誤	
		1	不認可的路徑	
		2	找不到路徑	
		3	找不到檔案	
		4	不正確的檔案類型	
		13	存取權限不足	
		22	結果變數（<結果>）的字串長度過短。	
<檔案名稱>：	欲讀出檔案資訊的檔案名稱。			
	類型：	CHAR[160]		
	絕對路徑可於實際檔案名稱之前指定。若不指定特定路徑，則會在目前的目錄中搜尋檔案（= 所選程式的目錄）。 路徑資料規則請參閱「定址程式記憶體檔案 (頁 237)」。			
<結果>：	結果變數（Call-By-Reference 參數） 儲存了要求之檔案資訊的變數。			
	類型：	VAR CHAR[8]	於	FILEDATE 格式："dd.mm.yy"
		VAR CHAR[8]	於	FILETIME 格式："hh.mm.ss"
		VAR INT	於	FILESIZE 將檔案大小以位元輸出
		VAR CHAR[5]	於	FILESTAT 格式："rwxsd" (r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete)
		VAR CHAR[32]	於	FILEINFO 格式："rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

範例

程式碼	註解
`N10 DEF INT ERROR`	；錯誤變數的定義。

程式碼	註解
N20 STRING[32] RESULT	；結果變數的定義。
N30 FILEINFO(ERROR,"/_N_MPF_DIR/ _N_TESTFILE_MPF",RESULT)	；具有網域、檔案識別碼及路徑資料的檔案名稱。
N40 IF ERROR <> 0	；錯誤評估
N50 MSG("錯誤"<<ERROR<<"檔案資訊指令")	
N60 M0	
N70 ENDIF	
...	

在結果變數 RESULT 中，範例會提供下列結果：

"77777 12345678 26.05.00 13:51:30"

2.23 無條件進位（ROUNDUP）

輸入值，類型 REAL（具有小數點），使用 ROUNDUP 函數，會將值無條件進位成下一個較大的整數。

語法

```
ROUNDUP（<值>）
```

含義

ROUNDUP：	作為無條件進位輸入值的指令
<值>：	輸入值，類型 REAL

說明

輸入值，類型 INTEGER（整數），則不會改變。

範例

範例 1： 多種輸入值，及其無條件進位的結果

範例	無條件進位結果
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

範例 2： 在 NC 程式中的 ROUNDUP

程式碼

```
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 副程式技術

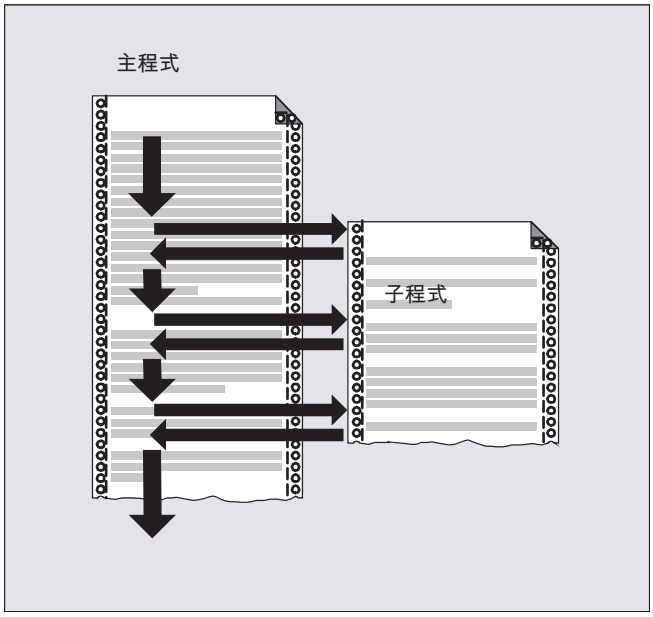
2.24.1 一般資訊

2.24.1.1 副程式

當工件程式嚴謹的區分為主程式與子程式時，「子程式」具有其原點。主程式為選擇用於在控制時處理並啟動的工件程式。 子程式為從主程式中呼叫的工件程式。

在今日的 SINUMERIK NC 程式語言中已無嚴格的區別。 基本上，每個工件程式均可選擇作為主程式並執行或作為子程式從其他工件程式中呼叫。

相對的，子程式可用於代表從其他工件程式中所呼叫的工件程式。



應用

與在高階程式語言中一樣，在 NC 程式語言中，子程式可取代在獨立、內含之程式中使用超過一次以上的程式區段。

子程式提供下列優點：

- 增加了程式的透明度和可讀取性
- 透過再利用已測試過的程式段增加品質

- 對於建立特定的加工庫，提供了可能性
- 儲存記憶體空間

2.24.1.2 副程式名稱

命名規則

若遵循下列規則，即可自由選擇子程式名稱：

- 可允許的字元：
  - 字母：A ... Z、a ... z
  - 數字：0 ... 9
  - 底線：_
- 前兩個字元應為兩個字母，或是在一個字母後接著一個底線。

  說明

  若滿足此條件，則可從另一個程式中以指定的程式名稱呼叫該 NC 程式作為子程式。然而，若該程式名稱開頭為數字，則僅能透過 CALL 敘述呼叫該子程式。

- 最大長度：24 個字元

說明

大寫 / 小寫字母

SINUMERIK NC 語言**並不**區分大小寫。

說明

不容許的程式名稱

為防止與 Windows 的應用程式產生問題，請**避免**使用下列程式名稱：

- CON、PRN、AUX、NUL
- COM1、COM2、COM3、COM4、COM5、COM6、COM7、COM8、COM9
- LPT1、LPT2、LPT3、LPT4、LPT5、LPT6、LPT7、LPT8、LPT9

控制：內部副檔名

在建立子程式時所指派的程式名稱在控制系統中可加上附加的字首與字尾作擴張：

- 字首：_N_
- 字尾：_SPF

使用程式名稱

在使用程式名稱時，例如在子程式呼叫中，所有的字首、程式名稱、及字尾組合均可使用。

範例：

使用程式名稱 SUB_PROG 的子程式可使用下列識別碼啟動：

1. SUB_PROG
2. _N_SUB_PROG
3. SUB_PROG_SPF
4. _N_SUB_PROG_SPF

具有相同名稱的主程式與子程式

若主程式 (.MPF) 和子程式 (.SPF) 的程式名稱相同，則在 NC 程式使用程式名稱時，必須指定適當的副檔名作為獨特識別。否則就會使用在搜尋路徑中，第一個發現具有指定名稱的程式。

2.24.1.3 巢狀的副程式

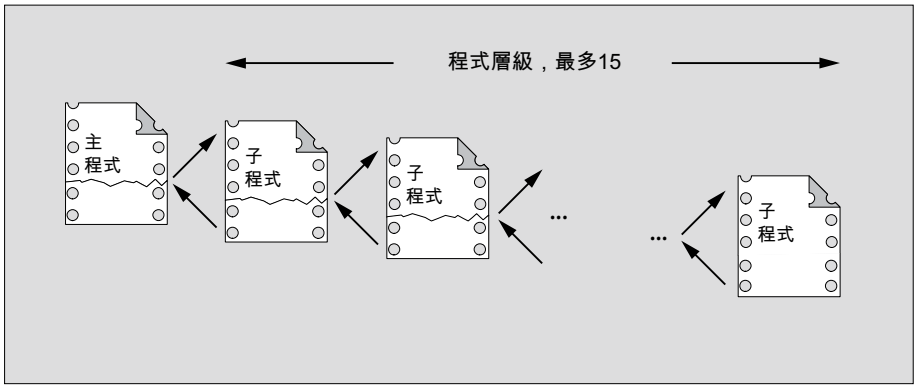
一個主程式可呼叫會呼叫更多副程式的副程式。此時，這些程式的執行順序彼此為巢狀結構。每個程式均在專屬的程式層次上執行。

巢狀深度

NC 程式語言目前提供 16 個程式層次。主程式永遠在最上層的程式層次，0 上執行。副程式永遠在呼叫後次低的程式層級上執行。因此程式層級 1 為第一副程式層級。

程式層級的分配：

- 程式層級 0：主程式層級
- 程式層級 1 至 15：副程式層級 1 至 15



中斷程式（ASUB）

若在中斷程式中呼叫副程式，則其不會在通道（n）中目前啟用的程式層級上執行，而是會在次低的程式層級（n+1）上執行。因此即便在最低的程式層級，在中斷程式中仍可有 2 個額外的程式層級（16 與 17）。

若需要超過 2 個程式層級，則必須明確的考慮建立工件程式在通道中執行的架構。換言之，僅能使用眾多程式層級中最大的層級使中斷處理能有足夠的程式層級可用。

例如，若中斷處理需要 4 個程式層級，則需將工件程式建構為最大使用程式層級 13 的架構。在中斷時，其便可使用所需要的 4 個程式層級（14 至 17）。

Siemens 週期

Siemens 週期需 3 個程式層級。因此，Siemens 週期最晚的呼叫點為：

- 工件程式處理：程式層級 12：
- 中斷程式：程式層級 14：

2.24.1.4 搜尋路徑

呼叫沒有路徑細節的子程式時，控制系統會使用預先定義的搜尋順序搜尋可用程式記憶體（請參閱「搜尋子程式呼叫的路徑 (頁 242)」）。

2.24.1.5 形式與實際參數

形式與實際參數會與使用參數傳輸對子程式進行定義與呼叫時發生。

形式參數

當定義子程式時，欲傳送至該子程式中的參數（稱為形式參數）需定義有類型與參數名稱。

因此，形式參數定義了子程式的介面。

範例：

程式碼	註解
`PROC CONTOUR（實數 X，實數 Y）`	`；形式參數： X 與 Y，均為實數類型`
`N20 X1=X Y1=Y`	`；將軸 X1 移動至位置 X 並將 Y1 移動至位置 Y`
`...`	
`N100 RET`	

實際參數

當呼叫子程式時，需將絕對值或變數傳（稱為實際參數）送至子程式中。

此時，實際參數會在稍後呼叫時指派最新的值至子程式介面中。

範例：

程式碼	註解
N10 DEF REAL WIDTH	；變數定義
N20 WIDTH=20.0	；變數指派
N30 CONTOUR（5.5，WIDTH）	；使用實際參數進行子程式呼叫： 5.5 及 WIDTH
...	
N100 M30	

2.24.1.6 參數傳輸

定義具有參數傳輸的子程式

具有參數傳輸的子程式可使用 PROC 關鍵字及子程式所需的所有參數列表進行定義。

不完整參數傳輸

當呼叫子程式時，並非所有定義在子程式介面中的參數均需明確的傳輸。 若有參數被省略，則傳輸時會以預設值"0"取代。

如此一來便可獨立識別參數的順序，但是，需永遠包含作為參數分隔字元的逗號。 最後的參數除外。 若在呼叫時忽略它，則最後的逗號亦會被省略。

範例：

子程式：

程式碼	註解
PROC SUB_PROG（實數 X、實數 Y、實數 Z）	；形式參數： X、Y、和 Z
...	
N100 RET	

主程式：

程式碼	註解
PROC MAIN_PROG	
...	

程式碼	註解
N30 SUB_PROG(1.0,2.0,3.0)	；具完整參數傳輸的子程式呼叫： X=1.0，Y=2.0，Z=3.0
…	
N100 M30	

在具有不完整參數傳輸的 N30 中進行子程式呼叫的範例：

```
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
```

注意

傳址呼叫參數傳輸

使用傳址呼叫傳輸的參數在子程式呼叫時不可省略。

注意

AXIS 資料類型

AXIS 資料類型參數在子程式呼叫時不可省略。

確認傳輸參數

系統變數$P_SUBPAR [n] where n = 1、2、等，可用於確認參數是有明確的傳輸或是在子程式中被省略。 索引 n 代表形式參數的順序。 Index n = 1 表示第一形式參數，index n = 2 表示第二形式參數，依此類推。

以下程式摘錄顯示如何以第一形式參數為基礎執行確認的範例：

程式設計	註解
PROC SUB_PROG（實數 X、實數 Y、實數 Z）	；形式參數： X、Y、和 Z
N20 IF $P_SUBPAR[1]==TRUE	；確認第一形式參數 X。
...	；若形式參數已明確傳輸時會執行這些動作。
N40 ELSE	
...	；若形式參數未傳輸時會執行這些動作。
N60 ENDIF	
...	；一般動作
N100 RET	

2.24.2 副程式的定義

2.24.2.1 沒有參數傳輸的副程式：

在定義沒有參數傳輸的子程式時，可忽略位在程式開頭處的定義指令列。

語法

```
[PROC <程式名稱>]
...
```

含義

PROC：	在程式開始處的定義操作
<程式名稱>：	程式的名稱

範例

範例 1： 具有 PROC 操作的子程式

程式碼	註解
PROC SUB_PROG	；定義行
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	；子程式傳回

範例 2： 沒有 PROC 操作的子程式

程式碼	註解
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	；子程式傳回

另請參見

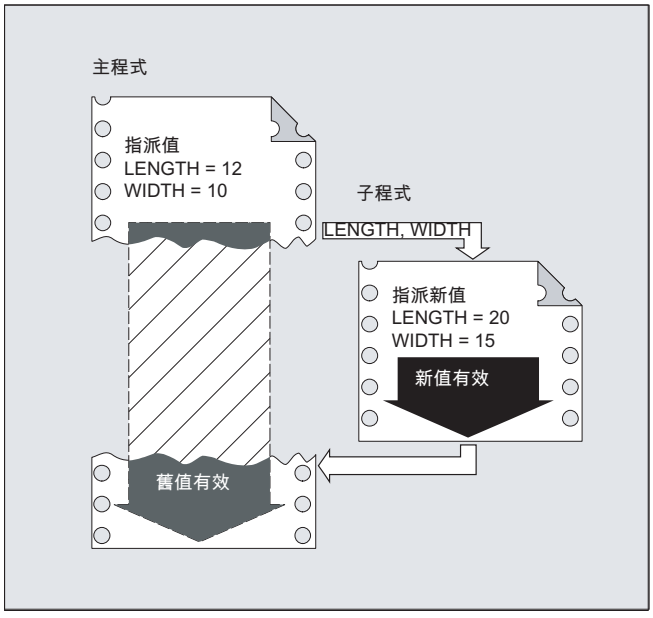
沒有參數傳輸的呼叫： (頁 208)

2.24.2.2 具有傳值呼叫參數傳輸（PROC）的副程式

具有傳值呼叫參數傳輸的子程式可利用在程式名稱後的 PROC 關鍵字以及所有具有其類型與名稱的完整參數清單進行定義。該定義操作需出現在程式指令列的第一列。

傳值呼叫

呼叫程式在傳值呼叫參數時僅傳輸變數數值至子程式。因此子程式不會取得變數的直接存取。於是，當參數數值變更時，只有子程式內可見的數值會修改。呼叫程式內定義的變數數值仍保持不變。結果，傳值呼叫參數傳輸對呼叫程式並無影響。



語法

```
PROC <程式名稱>（<參數類型> <參數名稱>=<init_value>...）
```

說明

最大可傳輸 127 個參數。

含義

PROC：	在程式開始處的定義操作
<程式名稱>：	程式的名稱
<參數類型>：	參數的資料類型（例如實數、整數、布林值）

<參數名稱>：	參數的名稱
<init_value>：	參數初始化的選用值（**選擇性**） 呼叫子程式時，若參數尚未指定，參數將配置為初始化值。

範例

範例 1

子程式 SUB_PROG 的定義具有三個含預設值的 REAL 類型參數：

程式碼

```
PROC SUB_PROG(REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

範例 2

各種呼叫版本

程式碼

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ；呼叫變化
  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
```

另請參見

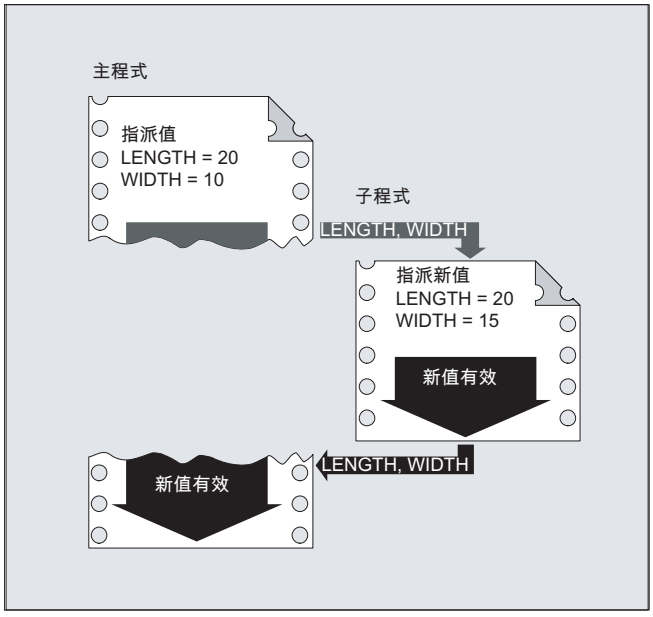
以參數傳輸（EXTERN）進行副程式呼叫 (頁 210)

2.24.2.3 具有傳址呼叫參數傳輸（PROC、VAR）的副程式

具有傳址呼叫參數傳輸的子程式可利用在程式名稱後的 PROC 關鍵字以及所有具有 VAR 關鍵字、類型與名稱的完整參數清單進行定義。該定義操作需出現在程式指令列的第一列。如為參數，亦會傳輸陣列的參考位址。

傳址呼叫

呼叫程式在傳址呼叫參數傳輸中不是傳輸變數數值至子程式，而是傳輸參考（指標）至變數。子程式將取得變數的直接存取。因此，變更參數數值時，不只修改子程式中可見的數值，也會修改呼叫程式中定義的變數數值。傳址呼叫參數傳輸會影響呼叫程式，即使子程式已結束。



說明

傳址呼叫參數傳輸只有傳輸變數是在呼叫程式 (LUD) 本機定義時需要。通道全域或 NC 全域變數不需傳輸，因其無法從子程式中直接進行存取。

語法

```
PROC <程式名稱>（VAR <參數類型> <參數名稱>，等等）
PROC <程式名稱>（VAR <陣列類型> <陣列名稱>，[<m>,<n>,<o>]，等等）
```

說明

最大可傳輸 127 個參數。

含義

PROC：	在程式開始處的定義操作
VAR：	透過參考位址傳輸之參數的關鍵字

<程式名稱>：	程式的名稱	
<參數類型>：	參數的資料類型（例如實數、整數、布林值）	
<參數名稱>：	參數的名稱	
<陣列類型>：	陣列元素的資料類型（例如實數、整數、布林值）	
<陣列名稱>：	陣列的名稱	
[<m>,<n>,<o>]：	陣列大小 目前，最多可使用 3 維陣列：	
	<m>：	1 維陣列的維度
	<n>：	2 維陣列的維度
	<o>：	3 維陣列的維度

說明

- 指定在 PROC 關鍵字之後的程式名稱需與使用者介面上指派的程式名稱相符。
- 對於未定義陣列長度的陣列，子程式可以處理將變數長度作為正式參數的陣列。當定義二維陣列作為正式參數時，舉例來說，不會指定一維的長度。然而，必須寫入逗號。
  範例：PROC <程式名稱> (VAR REAL ARRAY[,5])

範例

；以參考至實數類型的 2 個參數定義子程式：

程式碼

```
；參數 1：參考的類型：實數，名稱：長度
；參數 2：參考的類型：實數，名稱：寬度
PROC SUB_PROG(VAR REAL LENGTH, VAR REAL WIDTH)
```

另請參見

以參數傳輸（EXTERN）進行副程式呼叫 (頁 210)

2.24.2.4 儲存模態 G 碼功能（SAVE）

SAVE 屬性表示在子程式呼叫之前，儲存了已啟動的 G 指令，並在子程式結尾之後，重新啟用。

注意
中斷連續路徑模式 如連續路徑模式已生效，並以 SAVE 屬性呼叫子程式，則在該子程式結束時（跳躍回原呼叫程式），連續路徑模式將中斷。

語法

```
PROC <子程式名稱> SAVE
```

含義

SAVE：	在子程式呼叫之前，儲存模態 G 指令，並在子程式結尾之後，重新儲存。

範例

模態 G 指令 G91 在 CONTOUR 子程式的增量尺寸為有效。模態 G 指令 G90 在主程式（絕對尺寸）中有效。在子程式結尾之後的主程式之中，G90 再次有效，因為具有 SAVE 子程式定義的。

子程式定義：

程式碼	註解
PROC CONTOUR （REAL VALUE1） SAVE	；具有 SAVE 參數的子程式定義
N10 G91 ...	；模態 G 指令 G91：增量尺寸
N100 M17	；子程式結束

主程式：

程式碼	註解
N10 G0 X...Y...G90	；模態 G 指令 G90：絕對尺寸
N20 ...	
...	
N50 CONTOUR （12.4）	；子程式呼叫
N60 X...Y...	；模態 G 指令 G90 使用 SAVE 重新啟動：

補充條件

框架

關於具有 SAVE 屬性的子程式，其框架之行為，會依照框架時間而有所不同，並可設定使用機械資料。

參考資料

功能手冊，基本功能；座標軸、座標系統、框架 (K2)，
章節：「以 SAVE 傳回的子程式」

2.24.2.5 抑制獨立單節執行（SBLOF，SBLON）

用來完成程式的單一單節停用

當啟用了單一單節時，以 SBLOF 設計的程式會被完整執行，就像一個單節一樣。換言之，會停用單一單節執行，以完成程式。

SBLOF 係在 PROC 行之中，並保持有效，直到子程式結尾或直到它被中斷為止。 在傳回指令處，會決定是否要在子程式結尾停止：

以 M17 傳回跳躍：	在子程式結尾停止
以 RET 傳回跳躍：	在子程式結尾不停止

在程式中的單一單節停用

SBLOF 必須單獨保持在單節中。 會停用在此單節之後的單一單節，直到：

- 下一個 SBLON
  或
- 生效子程式層級結尾

語法

用來完成程式的單一單節停用：

```
PROC ... SBLOF
```

在程式中的單一單節停用：

```
SBLOF
...
SBLON
```

含義

PROC：	程式中的第一個操作
SBLOF：	用來停用單一單節執行的指令 SBLOF 可以寫入在 PROC 單節之中，或單獨存在於單節之中。
SBLON：	用來啟用單一單節執行的指令 SBLON 必須為在獨立單節中。

補充條件

- **單一單節停用與單節顯示**
  目前的單節顯示，可使用 DISPLOF，在循環 / 子程式中停用。若 DISPLOF 和 SBLOF 一起被程式設計，那麼循環 / 子程式呼叫會繼續在單一單節在循環 / 子程式中的停止處繼續顯示。
- **在系統 ASUB 或使用者 ASUB 中的單一單節停用**
  若單一單節在系統中停止，或使用機械資料設定 MD10702 $MN_IGNORE_SINGLEBLOCK_MASK（bit0 = 1 或 bit1 = 1），停用了使用者 ASUB，那麼可藉由 ASUB 中的 SBLON 來進行程式設計，重新啟用該單一單節停止。
  若使用機械資料設定 MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP，停用了使用者 ASUB 中的單一單節停止，那麼該單一單節停止，將**無法**藉由在 ASUB 中的 SBLON 進行程式設計而重新啟用。
- **對於數個不同的單一單節執行類型，其單一單節停用的特殊部分**
  當啟用單一單節執行 SBL2（在各工件程式單節後停止）時，**沒有**執行停止在 SBLON 單節中（若在 MD10702 $MN_IGNORE_SINGLEBLOCK_MASK 之中將位元 12 設定為「1」（防止單一單節停止））。
  當單一單節執行 SBL3 啟用時（在各工件程式單節後，還在該循環中），SBLOF 指令會被停用。

範例

範例 1： 在程式中的單一單節停用

程式碼	註解
N10 G1 X100 F1000	
N20 SBLOF	；停用單一單節。
N30 Y20	
N40 M100	

程式碼	註解
N50 R10=90	
N60 SBLON	；重新啟用單一單節。
N70 M110	
N80 ...	

在 N20 與 N60 之間的區域，會以在單一單節模式中的一個步驟來執行。

範例 2： 循環會以類似指令的方式來對使用者動作

主程式：

程式碼
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30

循環 CYCLE1：

程式碼	註解
N100 PROC CYCLE1 DISPLOF SBLOF	；停用單一單節
N110 R10=3*SIN（R20）+5	
N120 IF (R11 <= 0)	
N130 SETAL（61000）	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

CYCLE1 會進入啟用的單一單節執行，換言之，必須按下一次開始鍵來進行 CYCLE1。

範例 3： 為了要啟用一個修正過的零點偏移量以及刀具偏移量，由自行執行一個 PLC 啟動的 ASUB。

程式碼	
N100 PROC ZO 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

程式碼
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

範例 4： 沒有以 MD10702 位元 12 = 1 停止

初始狀態：

- 單一單節已啟用。
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

主程式：

程式碼	註解
N10 G0 X0	；在此工件程式行中停止。
N20 X10	；在此工件程式行中停止。
N30 CYCLE	；由循環產生的移動單節。
N50 G90 X20	；在此工件程式行中停止。
M30	

循環 CYCLE：

程式碼	註解
PROC CYCLE SBLOF	；停用單一單節停止。
N100 R0 = 1	
N110 SBLON	；因為 MD10702 bit12=1，執行不會在工件程式行中停止。
N120 X1	；執行在此工件程式行中停止。
N140 SBLOF	
N150 R0 = 2	
RET	

範例 5： 用於程式巢狀嵌套的單一單節停用

初始狀態：

單一單節已啟用。

巢狀程式

程式碼	註解
N10 X0 F1000	；執行在此單節中停止。
N20 UP1（0）	
PROC UP1（INT _NR） SBLOF	；停用單一單節停止。
N100 X10	
N110 UP2（0）	
PROC UP2（INT _NR）	
N200 X20	
N210 SBLON	；啟用單一單節停止。
N220 X22	；執行在此單節中停止。
N230 UP3（0）	
PROC UP3（INT _NR）	
N300 SBLOF	；停用單一單節停止。
N305 X30	
N310 SBLON	；啟用單一單節停止。
N320 X32	；執行在此單節中停止。
N330 SBLOF	；停用單一單節停止。
N340 X34	
N350 M17	；SBLOF 為啟用。
N240 X24	；執行在此單節中停止。 SBLON 為啟用。
N250 M17	；執行在此單節中停止。SBLON 為啟用。
N120 X12	
N130 M17	；執行在此傳回跳躍單節中停止。 PROC 的 SBLOF 陳述式為啟用。
N30 X0	；執行在此單節中停止。
N40 M30	；執行在此單節中停止。

其他資訊

對非同步的子程式禁用單一單節

為了在一個步驟中執行 ASUB，必須在 ASUB 中以 SBLOF 陳述式，程式設計一個 PROC 陳述式。 亦可套用至函數“可編輯的系統 ASUB”（MD11610 $MN_ASUP_EDITABLE）。

可編輯系統 ASUB 之範例：

程式碼	註解
N10 PROC ASUB1 SBLOF DISPLOF	
N20 IF $AC_ASUP==’H200’	
N30 RET	；沒有 REPOS 供模式變更所用。
N40 ELSE	
N50 REPOSA	；REPOS 在所有其他情形中。
N60 ENDIF	

於單一單節模式中程式控制

藉由單一單節執行函數，使用者可以一個單節一個單節的方式，執行工件程式。 下列設定類型存在於：

- SBL1： 在各機台函數單節之後，具有停止的 IPO 單一單節。
- SBL2： 在各單節之後，具有停止的單一單節。
- SBL3： 在循環中停止（選擇 SBL3，停用 SBLOF 指令）。

用於程式巢狀嵌套的單一單節停用

若 SBLOF 在子程式中的 PROC 陳述式進行程式設計，那麼執行會在子程式以 M17 傳回跳躍處停止。 此舉使得在呼叫中的程式之中的下一個單節不會被執行。 若 SBLOF，沒有 SBLOF，在子程式中的 PROC 陳述式進行程式設計，那麼單一單節停用會啟用，執行只會在呼叫中程式的下一個加工函數單節之後，才會停止。 若不想這樣，則必須在子程式中，程式設計 SBLON，且必須再傳回（M17）之前。 執行不會因為以 RET 傳回跳躍到更高層級的程式而停止。

2.24.2.6 抑制目前的單節顯示（DISPLOF、DISPLON、ACTBLOCNO）

目前程式單節會顯示於單節顯示中作為標準。 目前單節的顯示可使用 DISPLOF 指令在循環與子程式中加以抑制。 會顯示呼叫的循環或子程式，而不是目前的單節。 DISPLON 指令可撤回對單節顯示的抑制。

DISPLOF 與 DISPLON 和 PROC 操作一同程式設計在程式指令列中，且對整個子程式均有效，且對於從其中所呼叫的不含 DISPLON 或 DISPLOF 指令的子程式亦有潛在的效果。 此對於所有的 ASUB 均適用。

語法

```
PROC ... DISPLOF
PROC ... DISPLOF ACTBLOCNO
PROC ... DISPLON
```

含義

DISPLOF：	用來停用目前單節顯示的指令。	
	位置：	在具有 PROC 操作的程式指令列結尾
	有效：	往上到從子程式傳回的跳躍或程式結尾。
	附註： 若使用 DISPLOF 指令從該子程式中呼叫了其他子程式，則目前的單節顯示在這些子程式中亦會受到抑制，除非其中有明確的程式設計了 DISPLON。	
DISPLON：	用於撤回對目前單節顯示的抑制	
	位置：	在具有 PROC 操作的程式指令列結尾
	有效：	往上到從子程式傳回的跳躍或程式結尾。
	附註： 若使用 DISPLON 指令從該子程式中呼叫了其他子程式，則目前的單節在這些子程式中亦會顯示，除非其中有明確的程式設計了 DISPLOF。	
ACTBLOCNO：	DISPLOF 和 ACTBLOCNO 屬性，表示在警報的狀態下，於警報發生處，輸出實際的單節號碼。 當 DISPLOF 在較低程式層級中進行程式設計時，也能套用。 另一方面，DISPLOF，不含 ACTBLOCNO，從上一個沒有以 DISPLOF 指派的程式層級而來的循環或子程式呼叫，會顯示其單節號碼。	

範例

範例 1： 在循環中停用目前的單節顯示

程式碼	註解
PROC CYCLE （AXIS TOMOV，REAL POSITION） SAVE DISPLOF	；取代停用目前的單節顯示，應顯示循環呼叫，例如： CYCLE（X，100.0）
DEF REAL DIFF	；循環內容
G01 ...	
...	
RET	；子程式返回跳躍。 跟在循環呼叫之後的單節，顯示在單節顯示中。

範例 2： 用來輸出警報的單節顯示

子程式 SUBPROG1（含 ACTBLOCNO）：

程式碼	註解
PROC SUBPROG1 DISPLOF ACTBLOCNO	
N8000 R10 = R33 + R44	
...	
N9040 R10 = 66 X100	；觸發警報 12080
...	
N10000 M17	

子程式 SUBPROG2（含 ACTBLOCNO）：

程式碼	註解
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	；觸發警報 12080
...	
N7000 M17	

主程式：

程式碼	註解
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	；警報輸出 =「12080 通道 K1 單節 N9040 語法錯誤文字 R10=」
N2060 ...	
N2350 SUBPROG2	；警報輸出 =「12080 通道 K1 單節 N2350 語法錯誤文字 R10=」
...	
N3000 M30	

範例 3：撤回對目前單節顯示的抑制

具有抑制的子程式 SUB1：

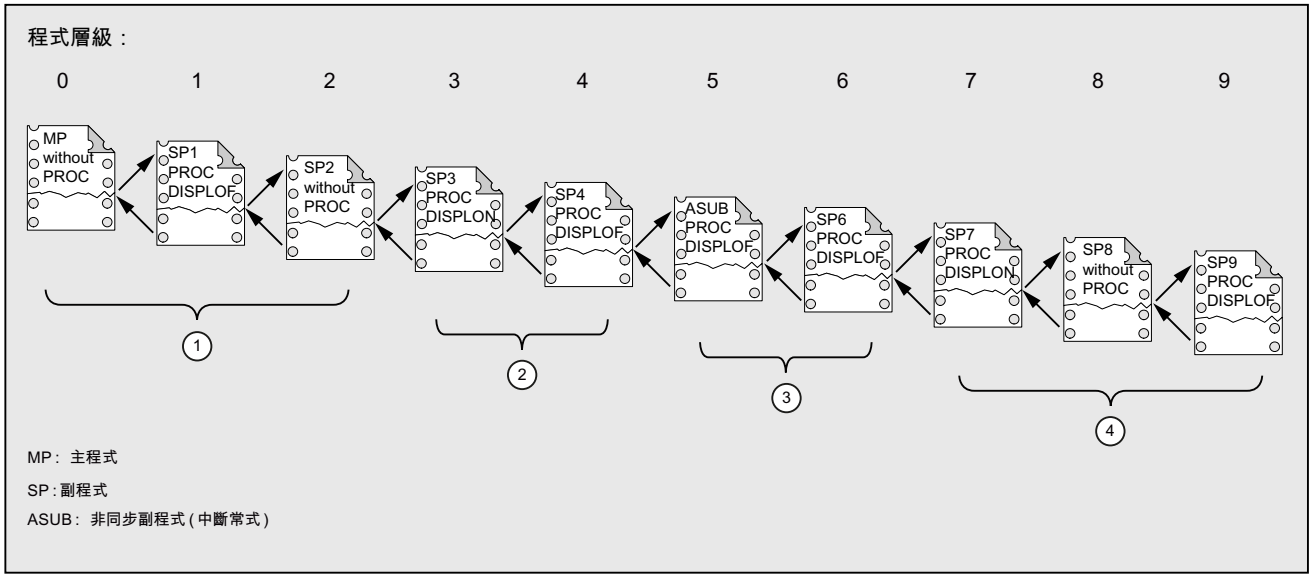
程式碼	註解
PROC SUB1 DISPLOF	；抑制 SUB1 子程式中的目前單節顯示。 取而代之的是，該單節會在 SUB1 呼叫時顯示。

程式碼	註解
...	
N300 SUB2	；呼叫子程式 SUB2。
...	
N500 M17	

不含抑制的子程式 SUB2：

程式碼	註解
PROC SUB2 DISPLON	；撤回對子程式 SUB2 中目前單節顯示的抑制。
...	
N200 M17	；返回子程式 SUB1。 對目前單節顯示的抑制重新儲存在 SUB1 中。

範例 4： 針對不同的 DISPLON/DISPLOF 組合顯示回覆



① 來自程式層級 0 的工件程式指令列會顯示在目前單節顯示中。

② 來自程式層級 3 的工件程式指令列會顯示在目前單節顯示中。

③ 來自程式層級 3 的工件程式指令列會顯示在目前單節顯示中。

④ 來自程式層級 7/8 的工件程式指令列會顯示在目前單節顯示中。

2.24.2.7 以準備（PREPRO）定義副程式

在啟動過程中，所有檔案皆可在 PROC 行結尾處，以 PREPRO 關鍵字來定義。

說明

這種程式準備，依機械資料的相關設定而有所不同。 請遵守製造商說明。

參考資料：

功能手冊，特殊函數，前置處理（V2）

語法

```
PROC ... PREPRO
```

含義

PREPRO：	在啟動期間，準備了用來辨識所有檔案的關鍵字（對於儲存在循環目錄中的 NC 程式）

以準備和子程式呼叫來讀取子程式

循環目錄會以相同的順序處理，提供在啟動期間和在子程式呼叫時，以參數處理的子程式所用。

1. _N_CUS_DIR 使用者循環
2. _N_CMA_DIR 製造商循環
3. _N_CST_DIR 標準循環

在 NC 程式共用相同名稱但有不同特性時，會啟用第一個發現的 PROC 操作，而其他的 PROC 操作則會被忽略且不發出警報訊息。

2.24.2.8 副程式返回 M17

返回指令 M17（或工件程式結尾指令 M30）會出現在子程式結尾。 其會提示返回至工件程式單節中位於子程式呼叫之後的呼叫程式中。

說明

M17 與 M30 在 NC 程式語言中視為相同。

語法

```
PROC <程式名稱>
...
M17/M30
```

補充條件

在連續路徑模式中子程式返回的影響

若 M17（或 M30）單獨出現在工件程式單節中，會中斷通道中所啟用的連續路徑模式。

為避免連續路徑模式中斷，最後移動的單節中需含有 M17（或 M30）。 更進一步，需將以下機械資料設為「0」：

MD20800 $MC_SPF_END_TO_VDI = 0（沒有 M30/M17 輸出至 NC/PLC 介面）

範例

1. 在個別單節中具有 M17 的子程式

程式碼	註解
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10	
N30 M17	；利用中斷連續路徑模式的方式返回跳躍點。

2. 在最後移動單節中具有 M17 的子程式

程式碼	註解
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	；不使用中斷連續路徑模式的方式返回跳躍點。

2.24.2.9 RET 副程式返回

RET 指令亦可用在子程式中替代 M17 返回眺躍指令。 RET 必須程式設計在獨立的工件程式單節中。 與 M17 類似，RET 會提示返回至工件程式單節中緊接在子程式呼叫之後的呼叫程式中。

說明

可程式設計參數以變更 RET 的跳躍行為（請參閱"$可參數化的副程式傳回跳躍（RET ...）(頁 198)"）。

應用

若希望 G64 連續路徑模式（G641 到 G645）在程式跳躍回來時不中斷，則必須使用 RET 操作。

需求條件

RET 指令只可以在未以 SAVE 屬性定義的子程式中使用。

語法

```
PROC <程式名稱>
...
RET
```

範例

主程式：

程式碼	註解
PROC MAIN_PROGRAM	；程式開始處
...	
N50 SUB_PROG	；子程式呼叫： SUB_PROG
N60 ...	
...	
N100 M30	；程式結束

子程式：

程式碼	註解
PROC SUB_PROG	
...	
N100 RET	；傳回跳躍至主程式中的單節 N60。

2.24.2.10 可參數化的副程式傳回跳躍（RET ...）

一般來說，傳回跳躍是由子程式使用 RET 指令進入呼叫程式。之後會繼續進行處理，程式行跟隨在子程式呼叫之後。若程式處理要在其他位置繼續進行，可使用下列選項：

- 在 ISO 同源模式（在描述輪廓後）中，呼叫材料移除循環之後，重新執行程式。
- 因為錯誤掌控，而從任何子程式層級（就算在 ASUB 之後）返回到主程式。
- 透過數個程式層級傳回跳躍，作為提供編譯循環和 ISO 同源模組中的特殊應用。

為了達到這個目標，RET 指令應使用其他參數編程設定。

搜尋方向

指定參數 <目標單節> 時，會在呼叫單節後先於單節進行傳回跳躍。之後會在進行傳回跳躍的程式中，往程式**結尾**方向進行目標搜尋。若搜尋不成功，就會往程式起點搜尋。

語法

```
RET（「<目標單節>」）
RET（"<目標單節"」，<目標單節之後之單節>）
RET（"<目標單節"」，<目標單節之後之單節> <傳回跳躍階層之數量>）
RET（"<目標單節>"， ，<傳回跳躍階層之數量>）
RET（"<目標單節>"，<目標單節之後之單節>，<傳回跳躍階層之數量>，
<傳回跳躍至程式開始處>）
RET（ ，，<傳回跳躍階層之數量>，<傳回跳躍至程式開始處>）
```

含義

RET：	子程式結束		
<目標單節>：	宣告為跳躍目標，程式執行的單節，應當重新啟動。 若參數 **<**傳回跳躍階層之數量**>** 並未編程設定，則跳躍目標就在目前呼叫子程式的程式之中。 可能的資料包含：		
	<單節編號**>**	目標單節的編號。 會在最初往程式結尾方向進行傳回跳躍的程式中，進行單節編號搜尋。	
	<跳躍標記**>**	在進行傳回跳躍的程式中，必須存在跳躍標記。 會在最初往程式結尾方向進行傳回跳躍的程式中，進行跳躍標記搜尋。	
	<字串**>**	在進行傳回跳躍的程式中，必須存在字串（例如程式或變數名稱）。 會在最初往程式結尾方向進行傳回跳躍的程式中，進行字串搜尋。 當程式設計字元字串時，套用下列規則： • **在結尾處空白**（與跳躍標記相反，係在結尾以「:」辨識）。 • **在字元字串之前**僅有一個單節號碼或跳躍標記可供設定，**沒有程式指令**。	
<目標單節之後之單節>：	參數指定程式處理是否應在參數 **<**目標單節**>** 指定單節或後續單節中繼續進行。		
	類型：	INT	
	值：	0	傳回跳躍會在參數 **<**目標單節**>** 指定的單節中進行。
		> 0	傳回跳躍會在參數 **<**目標單節**>** 指定的下個單節中進行。

<傳回跳躍階層之數量>：	參數指定應跳躍通過的程式階層數量（傳回跳躍），在其中搜尋目標單節，並繼續處理程式。		
	類型：	INT	
	值：	1	程式會在「目前程式層級－1」之處重新開始（例如沒有參數的 RET）。
		2	程式會在「現行的程式層級－2」重新開始，換言之，會跳過一個層級。
		3	程式會在「現行的程式層級－3」重新開始，換言之，會跳過一個層級。
		...	
	值域：	1...15	
<傳回跳躍至程式開始處>：	參數針對進入主程式的傳回跳躍，指定程式是否應以生效的 **ISO 用語模式**於程式起點繼續進行。		
	類型：	BOOL	
	值：	1	若傳回跳躍至主程式中，且 **ISO 同源模式**在該處主動，那麼程式會朝向該程式開始處分支。

說明

對於子程式傳回跳躍，以字元字串指定目標單節搜尋，一開始總是都會在呼叫程式中，搜尋跳躍標記。

若要使用字串將跳躍目標獨自定義，則不允許讓該字串的名稱，和跳躍標記的名稱相同，否則子程式傳回跳躍，永遠只會跳躍到跳躍標記，而不會跳躍到字串（請參考範例 2）。

補充條件

當透過數個程式層級進行傳回跳躍，則會求得獨立程式層級 SAVE 敘述。

若有一個傳回跳躍，跳過了數個程式層級，而另有一個模態子程式啟用，如果其中有一個被跳過的程式中有取消選擇指令 MCALL，並已對模態子程式進行程式設計，那麼模態子程式仍保持有效。

注意
程式設計錯誤
對穿越多個程式層級的傳回跳躍而言，使用者有責任確保處理以必要的模態設定繼續進行。這是可行的，例如程式設計一個適當的主單節。

範例

範例 1：在 ASUB 執行後，重新進行主程式

程式設計	註解
N10010 CALL "UP1"	；程式層級 0（主程式）
N11000 PROC UP1	；程式層級 1
N11010 CALL "UP2"	
N12000 PROC UP2	；程式層級 2
...	
N19000 PROC ASUP	；程式層級 3（ASUB 執行）
...	
N19100 RET("N10900", ,$P_STACK)	；子程式傳回跳躍進入主程式
	；$P_STACK：實際程式階層
N10900	；主程式之中的目標單節
N10910 MCALL	；停用模態子程式呼叫
N10920 G0 G60 G40 M5	；初始化額外模態設定

範例 2：作為指定目標單節搜尋的字元字串（字串>）

主程式：

程式碼	註解
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1，iVar2=4	
N1010 ...	
N1200 subProg1	；呼叫子程式「subProg1」
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	；呼叫子程式「subProg2」
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1：iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1：G1 X30	；「subProg1」在此定義為跳躍標記。
N1560 ...	
N1600 subProg3	；呼叫子程式「subProg3」
N1610 ...	
N1900 M30	

子程式 subProg1：

程式碼	註解
`PROC subProg1`	
`N2000 R10=R20+100`	
`N2010 ...`	
`N2200 RET("subProg2")`	；在單節 N1400 將跳躍傳回給主程式

子程式 subProg2：

程式碼	註解
`PROC subProg2`	
`N2000 R10=R20+100`	
`N2010 ...`	
`N2200 RET("iVar1")`	；在單節 N1500 將跳躍傳回給主程式

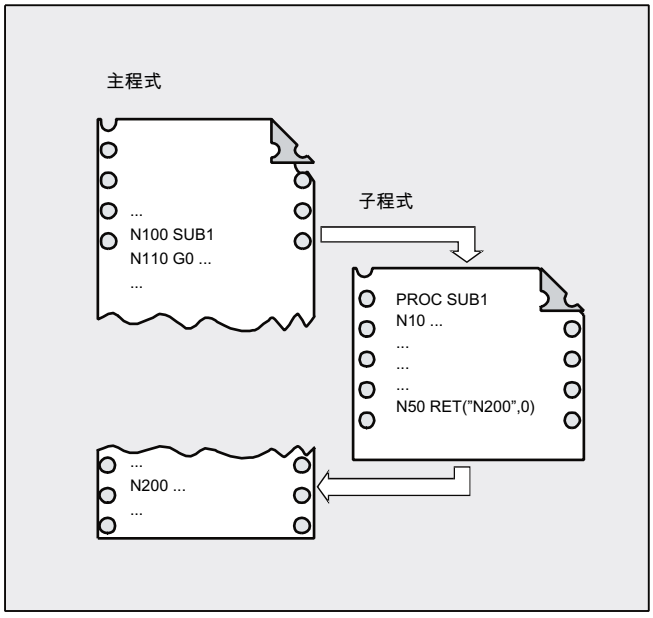
子程式 subProg3：

程式碼	註解
`PROC subProg3`	
`N2000 R10=R20+100`	
`N2010 ...`	
`N2200 RET("subProg1")`	；在單節 N1550 將跳躍傳回給主程式

其他資訊

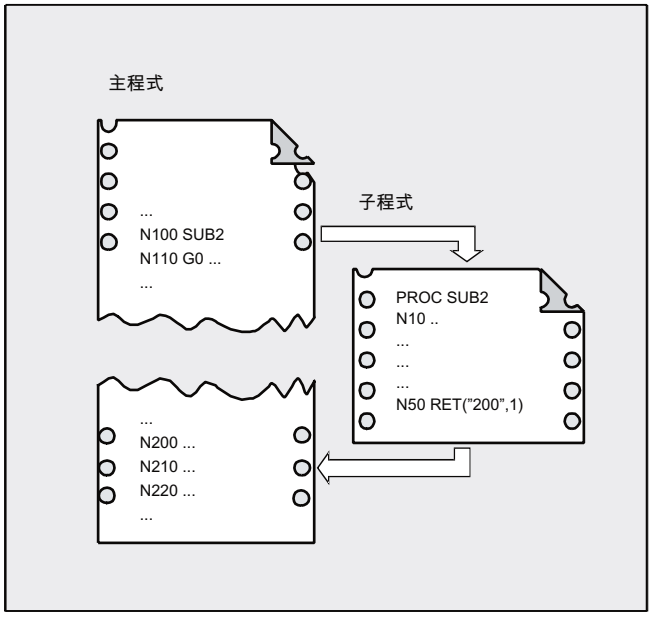
下圖顯示傳回跳躍參數的不同效應

1. <目標單節> = "N200"，<目標單節之後之單節> = 0



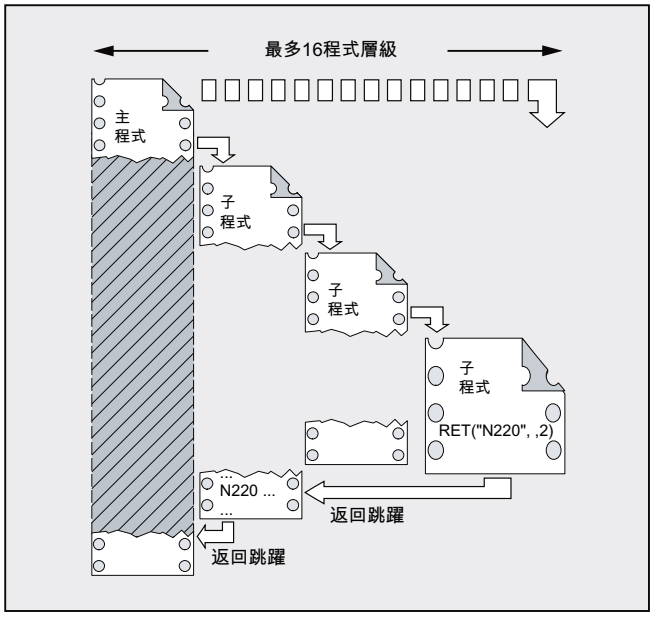
在 RET 指令之後，程式會繼續在主程式中，以 N200 單節執行。

2. <目標單節> = "N200"，<目標單節之後之單節> = 1



在 RET 指令之後，程式會繼續在主程式中，以 N210）單節執行，該單節係跟在 N200 單節之後。

3. <目標單節> = "N220"，<傳回跳躍階層之數量> = 2



在 RET 指令之後，兩個程式層級會跳躍，並繼續以 N220 單節執行程式。

2.24.2.11 可參數化的子程式傳回跳躍（RETB ...）

一般來說，傳回跳躍是由子程式使用 RETB 指令進入呼叫程式。之後會繼續進行處理，程式行跟隨在子程式呼叫之後。若程式處理要在其他位置繼續進行，可使用下列選項：

- 在 ISO 用語模式（在描述輪廓後）中，呼叫材料移除循環之後，重新執行程式。
- 因為錯誤掌控，而從任何子程式層級（就算在 ASUB 之後）返回到主程式。
- 透過數個程式層級傳回跳躍，作為提供編譯循環和 ISO 同源模組中的特殊應用。

為了達到這個目標，RETB 指令應使用其他參數編程設定。

搜尋方向

指定參數 <目標單節> 時，會在呼叫單節後先於單節進行傳回跳躍。之後會在進行傳回跳躍的程式中，往程式**開始**方向進行目標搜尋。若搜尋不成功，就會往程式結尾搜尋。

語法

```
RETB（"<目標單節>"）
RETB（"<目標單節>"，<目標單節之後之單節>）
RETB（"<目標單節>"，<目標單節之後之單節> <傳回跳躍階層之數量>）
RETB（"<目標單節>"， ，<傳回跳躍階層之數量>）
RETB（"<目標單節>"，<目標單節之後之單節>，<傳回跳躍階層之數量>，
<傳回跳躍至程式開始處>）
RETB（ ， ，<傳回跳躍階層之數量>，<傳回跳躍至程式開始處>）
```

含義

RETB：	子程式結束		
<目標單節>：	宣告為跳躍目標，程式執行的單節，應當重新啟動。 若參數 **<**傳回跳躍階層之數量**>** 並未編程設定，則跳躍目標就在目前呼叫子程式的程式之中。 可能的資料包含：		
	<單節編號**>**	目標單節的編號。 會在最初往程式開始處方向進行傳回跳躍的程式中，進行單節編號搜尋。	
	<跳躍標記**>**	在進行傳回跳躍的程式中，必須存在跳躍標記。 會在最初往程式開始處方向進行傳回跳躍的程式中，進行跳躍標記搜尋。	
	<字串**>**	在進行傳回跳躍的程式中，必須存在字串（例如程式或變數名稱）。 會在最初往程式開始處方向進行傳回跳躍的程式中，進行字串搜尋。 當程式設計字元字串時，套用下列規則： • **在結尾處空白**（與跳躍標記相反，係在結尾以「:」辨識）。 • **在字元字串之前**僅有一個單節號碼或跳躍標記可供設定，**沒有程式指令**。	
<目標單節之後之單節>：	參數指定程式處理是否應在參數 **<**目標單節**>** 指定單節或後續單節中繼續進行。		
	類型：	INT	
	值：	0	傳回跳躍會在參數 **<**目標單節**>** 指定的單節中進行。
		> 0	傳回跳躍會在參數 **<**目標單節**>** 指定的下個單節中進行。

<傳回跳躍階層之數量>：	參數指定應跳躍通過的程式階層數量（傳回跳躍），在其中搜尋目標單節，並繼續處理程式。		
	類型：	INT	
	值：	1	程式會在「目前程式層級－1」之處重新開始（例如沒有參數的 RET）。
		2	程式會在「現行的程式層級－2」重新開始，換言之，會跳過一個層級。
		3	程式會在「現行的程式層級－3」重新開始，換言之，會跳過一個層級。
		...	
	值域：	1...15	
<傳回跳躍至程式開始處>：	參數針對進入主程式的傳回跳躍，指定程式是否應以生效的 **ISO 用語模式**於程式起點繼續進行。		
	類型：	BOOL	
	值：	1	若傳回跳躍至主程式中，且 **ISO 同源模式**在該處主動，那麼程式會朝向該程式開始處分支。

說明

對於子程式傳回跳躍，以字元字串指定目標單節搜尋，一開始總是都會在呼叫程式中，搜尋跳躍標記。

若要使用字串將跳躍目標獨自定義，則不允許讓該字串的名稱，和跳躍標記的名稱相同，否則子程式傳回跳躍，永遠只會跳躍到跳躍標記，而不會跳躍到字串（請參考範例 2）。

補充條件

當透過數個程式層級進行傳回跳躍，則會求得獨立程式層級 SAVE 敘述。

若有一個傳回跳躍，跳過了數個程式層級，而另有一個模態子程式啟用，如果其中有一個被跳過的程式中有取消選擇指令 MCALL，並已對模態子程式進行程式設計，那麼模態子程式仍保持有效。

注意
程式設計錯誤
對穿越多個程式層級的傳回跳躍而言，使用者有責任確保處理以必要的模態設定繼續進行。這是可行的，例如程式設計一個適當的主單節。

範例

程式碼	註解
EXAMPLE.MPF	
…	
N3000 START_CYC(parm1, param2, …)	
N3010 TECH_CYC1(param1, param2, …)	
N3020 TECH_CYC2(param1, param2, …)	
N3030 TECH_CYC3(param1, param2, …)	
N3040 END_CYC(param1, param2, …)	
N3040 **END_CYC**(param1, param2, …)	
N3050…	
N4500 START_CYC(param11, param12, …)	
N4510…	
N4590 END_CYC(param11, param12, ..)	
N5000…	
…	
N6000 M30	

程式碼	註解
PROC END_CYC(…)	；在主程式 N3040 行呼叫
N10000…	
N15000 if status == 1	
N15010 RETB("START_CYC")	；傳回跳躍至呼叫程式 EXAMPLE.MPF
	；搜尋字串「START_CYC」
	；搜尋方向：於程式起始方向向後搜尋
	；程式處理以 N3000 行繼續進行
N15020 endif	
N15030 if status == 0	
N15040 RET	；傳回跳躍至呼叫程式 EXAMPLE.MPF
	；程式處理以 N3050 行繼續進行
N15050 endif	
N16000 RET("START_CYC")	；傳回跳躍至呼叫程式 EXAMPLE.MPF
	；搜尋字串「START_CYC」
	；搜尋方向：於程式結尾方向向前搜尋
	；程式處理以 N4500 行繼續進行
N17060 RETB	；傳回跳躍至呼叫程式 EXAMPLE.MPF
	；程式處理以 N3050 行繼續進行
	；無參數的 RETB 與 RET 相同

2.24.3 副程式呼叫

2.24.3.1 沒有參數傳輸的呼叫：

子程式會以位址 L 及子程式編號或透過指定程式名稱的方式進行呼叫。

主程式也能被呼叫作為子程式。 在此情況中，在主程式中設定的程式結尾 M2 或 M30，會被評估為 M17（具有傳回呼叫程式的程式結尾）。

說明

相對的，子程式也能被啟用作為主程式。

控制的搜尋策略：

有無任何*_MPF？

有無任何*_SPF？

這代表若待呼叫的子程式名稱，和主程式名稱相同，則已發出呼叫的主程式會再被呼叫一次。 一而言，這是無法預期的影響，且必須藉由指派獨特的名稱給子程式和主程式來避免。

說明

亦可從初始化檔案中呼叫不需參數傳輸的子程式。

語法

```
L <編號>/<程式名稱>：
```

說明

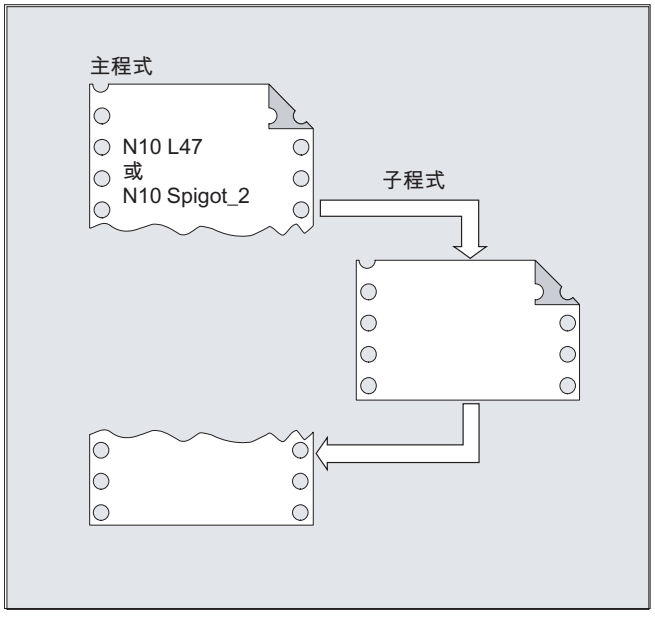
子程式呼叫必須永遠在 NC 單節中被程式設計。

含義

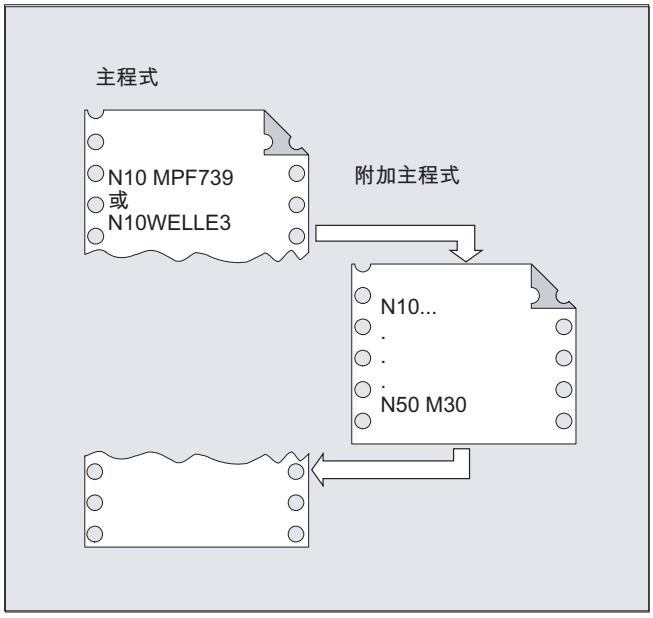
L：	子程式呼叫的位址	
<編號>：	子程式的名稱	
	類型：	INT
	值：	最多 7 個小數點位數 **注意：** 名稱中的前置零是有意義的（⇒ L123，L0123 和 L00123 是三個不同的子程式）。
<程式名稱>：	子程式（或主程式）的名稱	

範例

範例 1：沒有參數傳輸的子程式呼叫



範例 2： 以呼叫子程式的方式呼叫主程式



另請參見

沒有參數傳輸的副程式： (頁 180)

2.24.3.2 以參數傳輸（EXTERN）進行副程式呼叫

對於具有參數傳輸的子程式呼叫，可以直接傳輸變數或值（但不是 VAR 參數）。

具有參數傳輸的子程式，必須在他們被主程式呼叫之前，先在主程式之中，以 EXTERNAL 宣告（例如，在程式開始處）。 子程式和變數類型的名稱，從而會依照他們被傳輸的順序來指定。

注意
混淆風險 變數類型和傳輸順序，二者必須符合在子程式中，PROC 底下所宣告的定義。 在主程式和子程式中的參數名稱可以不同。

語法

```
外部 <程式名稱>(<type_Par1>,<type_Par2>,<type_Par3>)
...
<程式名稱>(<value_Par1>,<value_Par2>,<value_Par3>)
```

說明

子程式呼叫必須永遠在 NC 單節中被程式設計。

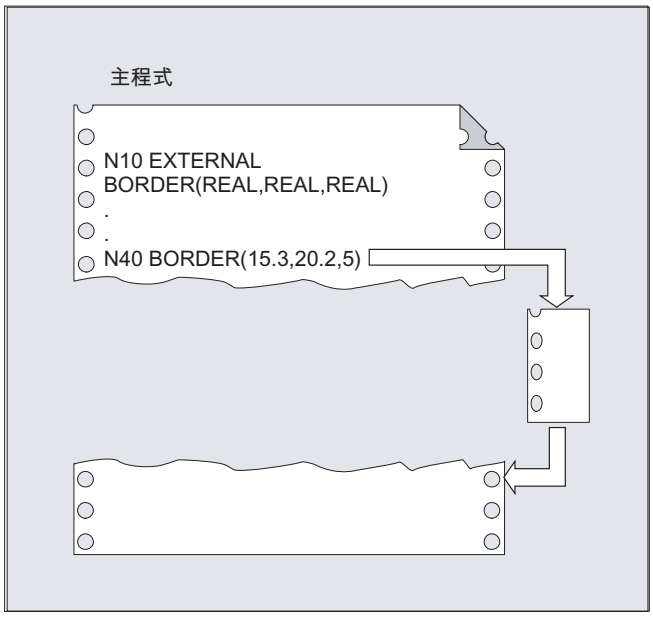
含義

<程式名稱>：	子程式的名稱
EXTERNAL：	作為以參數傳輸宣告子程式的關鍵字。 **附註：** 若子程式在工件中，或在全域子程式目錄中，您只需要指定 EXTERNAL。 循環不需宣告為 EXTERNAL。
<type_par1>、<type_par2>、<type_par3>：	依序待傳輸參數之變數類型
<value_par1>、<value_par2>、<value_par3>：	供待傳輸的參數所用的變數類型

範例

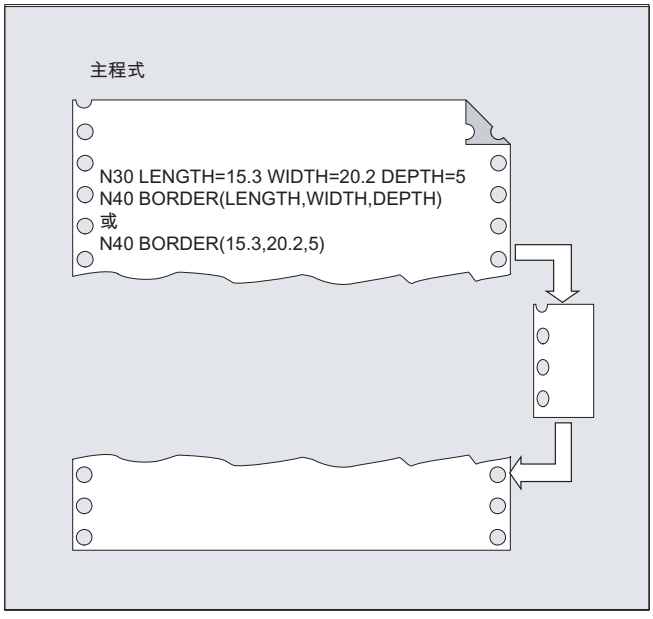
範例 1： 子程式呼叫在宣告之後進行

程式碼	註解
N10 EXTERNAL BORDERS(REAL,REAL,REAL)	；指定子程式。
...	
N40 BORDER(15.3,20.2,5)	；呼叫具有參數傳輸的子程式。



範例 2： 子程式呼叫時不使用宣告

程式碼	註解
`N10 DEF REAL LENGTH，WIDTH，DEPTH`	
`N20...`	
`N30 LENGTH=15.3 WIDTH=20.2 DEPTH=5`	
`N40 BORDER(LENGTH,WIDTH,DEPTH)`	；或： `N40 BORDER(15.3,20.2,5)`



另請參見

具有傳值呼叫參數傳輸（PROC）的副程式 (頁 181)

具有傳址呼叫參數傳輸（PROC、VAR）的副程式 (頁 182)

2.24.3.3 程式重覆的次數（P）

若有一個子程式連續執行數次，則程式重複所需的次數可用子程式呼叫的方式，在單節中的位址 P 輸入。

⚠ 小心
具有程式重複和參數傳輸的子程式呼叫 唯有當程式被呼叫時，才會傳輸參數，例如在第一次執行時。 對於其餘的重複，參數會維持不變。 若您想要在程式重複期間變更參數，您必須在子程式中適當提出。

語法

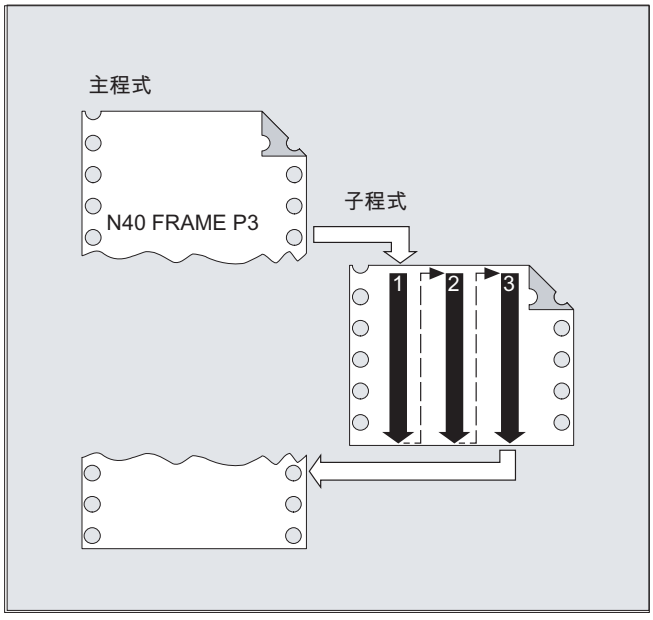
<程式名稱> P<值>

含義

<程式名稱>：	子程式呼叫	
P：	作為程式設計程式重複的位址	
<值>：	程式重覆之次數	
	類型：	INT
	值域：	1 … 9999 （沒有符號）

範例

程式碼	註解
...	
N40 FRAME P3	；BORDER 子程式要連續執行三次。
...	



2.24.3.4 模態副程式呼叫（MCALL）

由於模態子程式呼叫 MCALL(<程式名稱>) 的緣故，不會立即呼叫指定子程式，而是在每個移動單節進行路徑動作後，於此時在工件程式中執行呼叫。此外也會跨越程式層級。

說明

程式執行期間，只有最後一個模態子程式呼叫 MCALL(<程式名稱>) 有效（一定如此）。目前的模態子程式呼叫將取代在此之前生效的呼叫。

若參數傳輸至子程式，則參數就只會與呼叫傳輸 MCALL(<程式名稱>(Par1, Par2, ...))。

注意

沒有路徑動作的模態子程式

在下列情況中，模態子程式仍以沒有路徑動作的方式呼叫：

- 程式設計位址 S 或 F，若 G0 或 G1 生效。
- 若 G0 或 G1 獨自在單節中或與其他 G 指令進行程式設計。

語法

```
MCALL <程式名稱>
...
MCALL
```

含義

MCALL <程式名稱>：	啟動「模態子程式呼叫」函數
<程式名稱>：	子程式的名稱
MCALL：	「模態子程式呼叫」函數會以 MCALL 停用，無需指定程式名稱。

補充條件

ASUB

若 ASUB 中斷工件程式處理（請參閱「中斷常式（ASUB） (頁 138)」章節），就不會在此 ASUB 執行模態子程式呼叫。

若 ASUB 於「重置」通道狀態啟動，則在模態子程式呼叫方面，其行為就會和一般工件程式一樣。

換刀循環

若換刀循環期間取消選擇「模態子程式呼叫」函數，請注意即使在單節搜尋後，換刀循環會透過搜尋 ASUB 以隱性方式呼叫，或透過過量儲存手動呼叫。在這種情況下不得取消選擇「模態子程式呼叫」函數，否則搜尋結果就是錯誤的。因此建議針對取消選擇換刀循環「模態子程式呼叫」函數，以下列方式進行程式設計：

程式碼	註解
...	
IF $AC_ASUP == 0	；呼叫不會透過搜尋 ASUB 或過量儲存執行。
MCALL	；取消選擇「模態子程式呼叫」函數。
ENDIF	
...	

範例

範例 1

程式碼	註解
N10 G0 X0 Y0	
N20 MCALL L70	；啟動 L70 的模態子程式呼叫。
N30 X10 Y10	；逼近 X10 Y10，然後呼叫 L70。
N40 X20 Y20	；逼近 X20 Y20，然後呼叫 L70。
...	
N100 MCALL	；取消選擇「模態子程式呼叫」函數。
N110 X0 Y0	；逼近 X0 Y0，**不會**呼叫 L70。

範例 2

程式碼
N10 G0 X0 Y0 N20 MCALL L70 N30 L80

在此範例中，下列具有程式設計路徑軸的 NC 單節，係在子程式 L80 中。L70 由 L80 呼叫

2.24.3.5 間接子程式呼叫（CALL）

依照在程式中，特定點的一般條件而定，不同的子程式可以取消。 子程式的名稱儲存在 STRING 類型的變數中。 子程式呼叫可用 CALL 和變數名稱表示。

說明

唯有不具參數傳輸的子程式，才能使用間接子程式呼叫。 對於直接子程式呼叫，將名稱儲存在 STRING 常數中。

語法

```
CALL <程式名稱>
```

含義

CALL：	用於間接子程式呼叫的指令	
<程式名稱>：	子程式的名稱（變數或常數）	
	類型：	STRING

範例

以 STRING 常數直接呼叫：

程式碼	註解
… CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PART1_SPF" … …	；以 CALL 直接呼叫至子程式 PART1。

透過變數間接呼叫：

程式碼	註解
…	
`DEF STRING[100] PROGNAME`	：定義變數。
`PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PART1_SPF"`	；將子程式 PART1 指派至 PROGNAME 變數中。
`CALL PROGNAME`	；透過 CALL 及 PROGNAME 變數間接呼叫至子程式 PART1。
…	

2.24.3.6 使用呼叫程式部份的規格間接進行副程式呼叫（CALL BLOCK ... TO ...）

CALL 及關鍵字組合 BLOCK ... TO 可用於間接呼叫子程式並執行由起始與結尾標籤所標示的程式區段。

語法

```
CALL <程式名稱> BLOCK <起始標籤> TO <結尾標籤>
CALL BLOCK <起始標籤> TO <結尾標籤>
```

含義

CALL：	用於間接子程式呼叫的指令	
<程式名稱>：	包含待執行之程式部分的子程式名稱（變數或常數）（**規格 選配**）。	
	類型：	STRING
	附註： 若有一個 <程式名稱> 尚未編寫，由 <起始標籤> 和 <結尾標籤> 標明的程式區段，會在目前的程式中搜尋並執行。	
BLOCK ... TO ...：	供間接程式區段執行所用的關鍵字組合	
<起始標籤>：	參考有待執行的程式區段開始之變數。	
	類型：	STRING
<結尾標籤>：	參考有待執行的程式區段結尾之變數。	
	類型：	STRING

範例

主程式：

程式碼	註解
...	
DEF STRING[20] STARTLABEL，ENDLABEL	；起始與結尾標籤的變數定義。
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	；間接子程式呼叫及與呼叫程式區段搭配的識別碼。
...	

子程式：

程式碼	註解
PROC CONTUR_1 ...	
LABEL_1	；起始標籤： 程式區段執行的開始。
N1000 G1 ...	
...	
LABEL_2	；結尾標籤： 程式區段執行的結尾。
...	

2.24.3.7 間接呼叫以 ISO 語言（ISOCALL）進行程式設計的程式

使用間接程式呼叫 ISOCALL，可以呼叫出用 ISO 語言程式設計的程式。隨後會啟用在機械資料中設定的 ISO 模式。在程式結尾，原始的執行模式再度變為有效。若機械資料中沒有設定任何 ISO 模式，則會在 Siemens 模式中呼叫子程式。

若需要對於 ISO 模式的進一步資訊，請參考
參考：
ISO 同源函數說明

語法

```
ISOCALL <程式名稱>
```

含義

ISOCALL：	用以設定在有效機械資料中的 ISO 模式，進行間接子程式呼叫的關鍵字。
<程式名稱>：	以 ISO 語言程式設計的程式名稱（變數或常數，類型 STRING）

範例：從 ISO 模式，呼叫具有循環程式設計的輪廓

程式碼	註解
0122_SPF	；在 ISO 模式中的輪廓說明
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME =「0122「	；Siemens 工件程式（循環）
...	
N2000 R11 = $AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	；在 ISO 模式中執行程式 0122.spf
...	
N2400 M30	

2.24.3.8 以路徑規格和參數 (PCALL) 呼叫副程式

藉由 PCALL，您可以絕對路徑和參數傳輸呼叫子程式。

語法

PCALL <路徑 / 程式名稱>（<參數 1>，...，<參數 n>）

含義

PCALL：	作為以絕對路徑名稱呼叫子程式的關鍵字
<路徑 / 程式名稱>：	絕對路徑資料包括子程式名稱。 路徑資料規則請參閱「定址程式記憶體檔案 (頁 237)」。 若沒有指定絕對路徑名稱，PCALL 會以具有程式識別碼的標準子程式呼叫的方式運作。 指定程式名稱，沒有前置碼及檔案識別碼。 若程式名稱程式設計時要具有前置碼及檔案識別碼，就必須使用 EXTERN 指令以前置碼及檔案識別碼明確宣告。
<參數 1>，...：	遵守子程式之 PROC 操作的實際參數。

範例

程式碼
`PCALL/_N_WKS_DIR/_N_SHAFT_WPD/SHAFT(parameter1,parameter2,…)`

2.24.3.9 供副程式呼叫（CALLPATH）所用的延伸搜尋路徑

使用指令，可以延伸供子程式呼叫所用的搜尋路徑 CALLPATH。 這表示子程式也能從非選擇的工件目錄呼叫，而不必指定子程式的完整的、絕對的路徑名稱。

若其他目錄於儲存全域子程序的外部程式記憶體使用，就可在 EES 模式「EES 不含 GDIR」中進行另一項應用選項。 在此狀況下，使用 CALLPATH 可由此子程式目錄延伸搜尋路徑。

搜尋路徑副檔名在輸入使用者循環（_N_CUS_DIR）之前便已完成。

搜尋路徑副檔名會因為下列事件再次被取消選擇：

- 具有空白的 CALLPATH
- 具有參數的 CALLPATH
- 工件程式結束
- 重設

語法

CALLPATH（"<路徑名稱>"）

含義

CALLPATH：	可程式設計搜尋路徑副檔名的關鍵字 在分離的工件程式行中是可程式設計的。
<路徑名稱>：	常數或變數，STRING 類型 包含目錄的絕對路徑名稱，搜尋路徑應以此延伸。 路徑資料規則請參閱「定址程式記憶體檔案 (頁 237)」。

範例

搜尋路徑應由特定工件目錄延伸：

程式碼

```
...
CALLPATH ("/_N_WKS_DIR/_N_MYWPD_WPD")
...
```

表示會設定下列的搜尋路徑（位置 5 為新增）：

1. 實際目錄 / *名稱*
2. 實際目錄 / *名稱_SPF*
3. 實際目錄 / *名稱_MPF*
4. //NC:/_N_SPF_DIR / *名稱_SPF*
5. **/_N_WKS_DIR / _N_MYWPD_WPD/名稱_SPF**
6. /N_CUS_DIR / *名稱_SPF*
7. /_N_CMA_DIR / *名稱_SPF*
8. /_N_CST_DIR / *名稱_SPF*

補充條件

- CALLPATH 檢查程式設計路徑名稱是否確實存在。 當發生工件程式錯誤時，會以修正單節警報 14009 中斷執行。
- CALLPATH 亦能程式設計為 INI 檔案。 只有在需要執行 INI 檔案時有效（WPD-INI 檔案或給 NC 主動資料，例如在第 1 通道中的框架_N_CH1_UFR_INI）。 搜尋路徑會再次重置。

2.24.3.10 執行外部副程式 (840D sl) (EXTCALL)

可從外部記憶體載入工件程式，並使用 EXTCALL 指令執行。

以下項目可作為外部記憶體：

- 本機磁碟
- 網路磁碟
- USB 磁碟

說明

僅有操作面板正面的 USB 介面或 TCU 可用作 USB 磁碟處理外部程式的介面。

注意

使用 USB 隨身碟時發生刀具 / 工件受損

建議不要使用 USB 隨身碟執行外部子程式。執行子程式時，若隨身碟因為遭到意外敲擊或不慎拔除造成接觸問題及離線等狀況，就會中斷與 USB 隨身碟之間的通訊，加工將立即停止。刀具及 / 或工件可能因此受損。

外部程式路徑的預設設定

您可利用設定資料，預設外部程式目錄的路徑：

SD42700 $SC_EXT_PROG_PATH

集合程式路徑和使用 EXTCALL 呼叫指定的識別碼，形成呼叫子程式的完整路徑。

說明

參數

呼叫外部程式時，這些參數都無法傳輸給它。

語法

```
EXTCALL ("<¸ô®|/><µ{¦¡¦W°Ù>")
```

含義

EXTCALL：	用於呼叫外部子程式的指令。	
"<路徑/><程式名稱>"：	類型 STRING 的常數 / 變數	
	<路徑>：	絕對或相對路徑資料（**選用**）
	<程式名稱>：	程式名稱以前置符號「_N_」指定。 可利用「_」或「.」字元在程式名稱後加上副檔名 (「MPF」、「SPF」) (**選用**)。 範例： "SHAFT" "SHAFT_SPF" "SHAFT.SPF"

路徑指定：代號

可以使用下列代號來指定路徑：

- 本機磁碟：「**LOCAL_DRIVE:**」
- CF 卡：「**CF_CARD:**」
- USB 磁碟（操作面板正面）：「**USB:**」

或者，也可使用縮寫「CF_CARD:」及「LOCAL_DRIVE:」。

範例

從本機磁碟執行

「MAIN.MPF」主程式儲存在 NC 記憶體中，並已選取為希望執行的程式。

子程式「SP_1」

外部子程式「SP_1.SPF」或「SP_1.MPF」位於本機磁碟目錄「/user/sinumerik/data/prog/WKS.DIR/WST1.WPD」中。

外部程式目錄的路徑可設定為：

SD42700 $SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

說明

呼叫外部子程式的路徑規格：

- 不使用預設設定："LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- 使用預設設定："SP_1"

子程式「SP_2」

外部子程式「SP_2.SPF」或「SP_2.MPF」位於 USB 磁碟的 WKS.DIR /WST1.WPD 目錄中。外部程式目錄的預設路徑設定為子程式「SP_1」的路徑，在主程式中也不會改寫。因此，呼叫子程式「SP_2」時，需要指定完整的路徑。

主程式「MAIN」

程式碼

```
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

其他資訊

含絕對路徑名稱的 EXTCALL 呼叫

如果副程式存在於指定路徑，則將在 EXTCALL 呼叫時執行。 如果副程式不在指定路徑，使用 EXTCALL 呼叫時程式將取消執行。

含相對路徑名稱 / 沒有路徑名稱的 EXTCALL 呼叫

如果 EXTCALL 呼叫使用相對路徑名稱或沒有路徑名稱，系統將搜尋下列程式記憶體：

1. 如果 SD42700 $SC_EXT_PROG_PATH 中已預先設定路徑名稱，則系統會先從這個路徑開始搜尋 EXTCALL 呼叫（程式名稱或相對路徑名稱）中所指定的資料。 結合下列字元後即可獲得絕對路徑：
   - 在 SD42700 $SC_EXT_PROG_PATH 的預設路徑規格
   - 分隔符號「/」
   - 在 EXTCALL 指令的路徑規格和副程式名稱
2. 若在 1. 底下沒有找到副程式，將搜尋使用者記憶體目錄。

找到第一個副程式後，搜尋結束。 如果找不到副程式，使用 EXTCALL 呼叫時程式將取消執行。

可調式重新載入記憶體（FIFO 緩衝）

執行外部副程式時需要重新載入記憶體。 重新載入記憶體的大小預設為 30 KB，只能由機台製造商進行變更（使用 MD18360 MM_EXT_PROG_BUFFER_SIZE）。

說明

使用跳躍指令的副程式

對於包含跳躍指令的外部副程式（GOTOF、GOTOB、CASE、FOR、LOOP、WHILE、REPEAT、IF、ELSE、ENDIF 等），跳躍目的地必須存在後載入記憶體內。

說明

ShopMill/ShopTurn 程式

新增在檔案結尾的輪廓說明表示 ShopMill 及 ShopTurn 程式必須完全儲存在唯讀記憶體中。

外部副程式平行執行時需要獨立的重新載入記憶體。

重置 / 程式結束 / 啟動

重置 (Reset) 及開機 (POWER ON) 將導致外部子程式呼叫中斷，並釋出相關的載入記憶體。

重置 / 程式結束後，「從外部資源執行」所選擇的子程式仍保持不變。 假設外部程式記憶體仍然可用，則行為並不會不同於內部選擇的程式。

參考資料

「從外部資源執行」的進一步資訊，請參閱：

功能手冊、基本功能、模式群組、通道、程式操作、重置行為 (K1)

2.24.3.11 執行外部副程式 (828D) (EXTCALL)

可從外部記憶體載入工件程式，並使用 EXTCALL 指令執行。

以下項目可作為外部記憶體：

- 使用者 CF 卡
- 網路磁碟
- USB 磁碟

說明

由於執行外部程式的介面位於 USB 磁碟，只能使用操作面板正面 (PPU) 的 USB 介面。

注意
使用 USB 隨身碟時發生刀具 / 工件受損 建議不要使用 USB 隨身碟執行外部子程式。執行子程式時，若隨身碟因為遭到意外敲擊或不慎拔除造成接觸問題及離線等狀況，就會中斷與 USB 隨身碟之間的通訊，加工將立即停止。刀具及 / 或工件可能因此受損。

外部程式路徑的預設設定

您可利用設定資料，預設外部程式目錄的路徑：

SD42700 $SC_EXT_PROG_PATH

集合程式路徑和使用 EXTCALL 呼叫指定的識別碼，形成呼叫子程式的完整路徑。

說明

參數

呼叫外部程式時，這些參數都無法傳輸給它。

語法

```
EXTCALL ("<¸ô®|/><µ{¦¡¦W°Ù>")
```

含義

EXTCALL：	用於呼叫外部子程式的指令。	
"<路徑/><程式名稱>"：	類型 STRING 的常數 / 變數	
	<路徑>：	絕對或相對路徑資料（**選用**）
	<程式名稱>：	程式名稱以前置符號「_N_」指定。 可利用「_」或「.」字元在程式名稱後加上副檔名 (「MPF」、「SPF」) (**選用**)。 範例： "SHAFT" "SHAFT_SPF" "SHAFT.SPF"

路徑指定：代號

可以使用下列代號來指定路徑：

- 使用者 CF 卡：「**CF_CARD:**」
- USB 磁碟（操作面板正面）：「**USB:**」

範例

「MAIN.MPF」主程式儲存在 NC 記憶體中，並已選取為希望執行的程式。

子程式「SP_1」

外部子程式「SP_1.SPF」或「SP_1.MPF」位於使用者 CF 卡的「/WKS.DIR / WST1.WPD」目錄中。

外部程式目錄的路徑可設定為：

SD42700 $SC_EXT_PROG_PATH = CF_CARD:WKS.DIR/WST1.WPD

說明

呼叫外部子程式的路徑規格：

- 不使用預設設定："CF_CARD:WKS.DIR/WST1.WPD/SP_1"
- 使用預設設定："SP_1"

子程式「SP_2」

外部子程式「SP_2.SPF」或「SP_2.MPF」位於 USB 磁碟的 WKS.DIR /WST1.WPD 目錄中。外部程式目錄的預設路徑設定為子程式「SP_1」的路徑，在主程式中也不會改寫。因此，呼叫子程式「SP_2」時，需要指定完整的路徑。

主程式「MAIN」

程式碼

```
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...
N060 M30
```

其他資訊

含絕對路徑名稱的 EXTCALL 呼叫

如果副程式存在於指定路徑，則將在 EXTCALL 呼叫時執行。如果副程式不在指定路徑，使用 EXTCALL 呼叫時程式將取消執行。

含相對路徑名稱 / 沒有路徑名稱的 EXTCALL 呼叫

如果 EXTCALL 呼叫使用相對路徑名稱或沒有路徑名稱，系統將搜尋下列程式記憶體：

1. 如果 SD42700 $SC_EXT_PROG_PATH 中已預先設定路徑名稱，則系統會先從這個路徑開始搜尋 EXTCALL 呼叫（程式名稱或相對路徑名稱）中所指定的資料。結合下列字元後即可獲得絕對路徑：
   - 在 SD42700 $SC_EXT_PROG_PATH 的預設路徑規格
   - 分隔符號「/」
   - 在 EXTCALL 指令的路徑規格和副程式名稱
2. 若在 1. 底下沒有找到副程式，將搜尋使用者記憶體目錄。

找到第一個副程式後，搜尋結束。 如果找不到副程式，使用 EXTCALL 呼叫時程式將取消執行。

可調式重新載入記憶體（FIFO 緩衝）

執行外部副程式時需要重新載入記憶體。 重新載入的記憶體大小已預設（請參閱 MD18360 MM_EXT_PROG_BUFFER_SIZE）。

說明

使用跳躍指令的副程式

對於包含跳躍指令的外部副程式（GOTOF、GOTOB、CASE、FOR、LOOP、WHILE、REPEAT、IF、ELSE、ENDIF 等），跳躍目的地必須存在後載入記憶體內。

說明

ShopMill/ShopTurn 程式

新增在檔案結尾的輪廓說明表示 ShopMill 及 ShopTurn 程式必須完全儲存在唯讀記憶體中。

外部副程式平行執行時需要獨立的重新載入記憶體。

重置 / 程式結束 / 啟動

重置 (Reset) 及開機 (POWER ON) 將導致外部子程式呼叫中斷，並釋出相關的載入記憶體。

重置 / 程式結束後，「從外部資源執行」所選擇的子程式仍保持不變。 假設外部程式記憶體仍然可用，則行為並不會不同於內部選擇的程式。

參考資料

「從外部資源執行」的進一步資訊，請參閱：

功能手冊、基本功能、模式群組、通道、程式操作、重置行為 (K1)

2.25 巨集技術（DEFINE ... AS）

注意
巨集增加程式設計的複雜性 巨集可能大幅更改控制的程式設計語言。使用巨技術必需非常謹慎。

巨集是一連串的獨立敘述，並一起被指派一個屬於自己的名稱。在程式執行期間呼叫巨集時，在程式名稱底下程式設計的敘述，會一個接著一個執行。

根據有效範圍區分（換言之，便是該巨集定義啟用的範圍），巨集分為以下幾類：

- 局部巨集
  局部巨集為定義在 NC 程式開始的巨集，執行時並非主程式。其會在呼叫 NC 程式時產生，並在程式結束重置或下次控制系統開啟電源時刪除。局部巨集僅能在定義他們的 NC 程式中存取。
- 程式全域巨集
  程式全域巨集為定義在 NC 程式開始的巨集，會當作主程式使用。其會在呼叫 NC 程式時產生，並在程式結束重置或下次控制系統開啟電源時刪除。程式全域巨集可以在主程式及所有子程式中存取。

  說明

  程式全域巨集的可用性

  定義在主程式中的程式全域巨集僅在設定了以下機械資料時於子程式中提供：
  MD11120 $MN_LUD_EXTENDED_SCOPE = 1
  若 MD11120 = 0，則主程式中所定義的程式全域巨集只可於主程式中提供。
- 全域巨集
  全域巨集為在定義檔（巨集檔）中定義的 NC 或通道全域巨集，甚至在程式結束重置後，或是下次控制系統開機時還能保持。可在任何主程式或子程式中，呼叫全域巨集並執行。

  說明

  為了使用 NC 程式中的**外部**巨集檔案之巨集，必須將巨集檔案下載至 NC。

巨集必須在使用前先行定義。此時，必須注意以下規則：

- 任何識別碼，G、M、H 函數與 L 子程式名稱皆可在巨集中定義。
- 巨集可以在程式開始完成定義，或是儲存在指定的定義檔（巨集檔）中。
- 在程式開始的局部及程式全域巨集已定義。

- 全域巨集必須定義在巨集檔中，例如 _N_DEF_DIR/_N_UMAC_DEF。
- G 指令巨集只能定義為全域巨集。
- 可用 2 位數對 H 和 L 函數進行程式設計。
- M 和 G 指令可用 3 位數進行程式設計。

說明

巨集不能覆寫關鍵字和保留名稱。

語法

巨集定義：
`DEFINE <巨集_名稱> AS <操作_1> <操作_2> ...`

在 NC 程式中呼叫：
<巨集_名稱>

含義

`DEFINE ... AS`：	用來定義巨集的關鍵字組合
<巨集_名稱>：	巨集名稱 只有識別碼允許作為巨集名稱。 使用巨集名稱，從 NC 程式呼叫巨集。
<操作_1>：	巨集中的第一個程式設計指令
<操作_2>：	巨集中的第二個程式設計指令

補充條件

巢狀

不認可巢狀巨集。

範例

範例 1：在程式開始處的巨集定義

程式碼	註解
`DEFINE LINE AS G1 G94 F300`	；巨集定義
`...`	

程式碼	註解
N70 LINE X10 Y20	；巨集呼叫
...	

範例 2：在巨集檔案中的巨集定義

程式碼	註解
DEFINE M6 AS L6	；在換刀時，呼叫子程式來掌控所需的資料傳輸。在子程式中輸出實際的 M 功能（例如，M106）。
DEFINE G81 AS DRILL（81）	；仿效 DIN-G 指令。
DEFINE G33 AS M333 G333	；在螺紋切削同步期間，需要 PLC。藉由機械資料，將 G 指令 G33，重新命名為 G333，讓使用者可以進行程式設計。

範例 3：外部巨集檔案

在將外部巨集檔案讀取控制之後，必須將巨集檔案下載至 NC。只有這樣，才能在 NC 程式中使用巨集。

程式碼	註解
%_N_UMAC_DEF	
；$PATH=/_N_DEF_DIR	；用戶專屬巨集
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	；主軸順時針，冷卻液開啟
DEFINE M14 AS M4 M7	；主軸逆時針，冷卻液開啟
DEFINE M15 AS M5 M9	；主軸停止，冷卻液關閉
DEFINE M6 AS L6	；呼叫換刀程式
DEFINE G80 AS MCALL	；取消選擇鑽孔循環
M30	

檔案與程式管理

3

3.1 程式記憶體

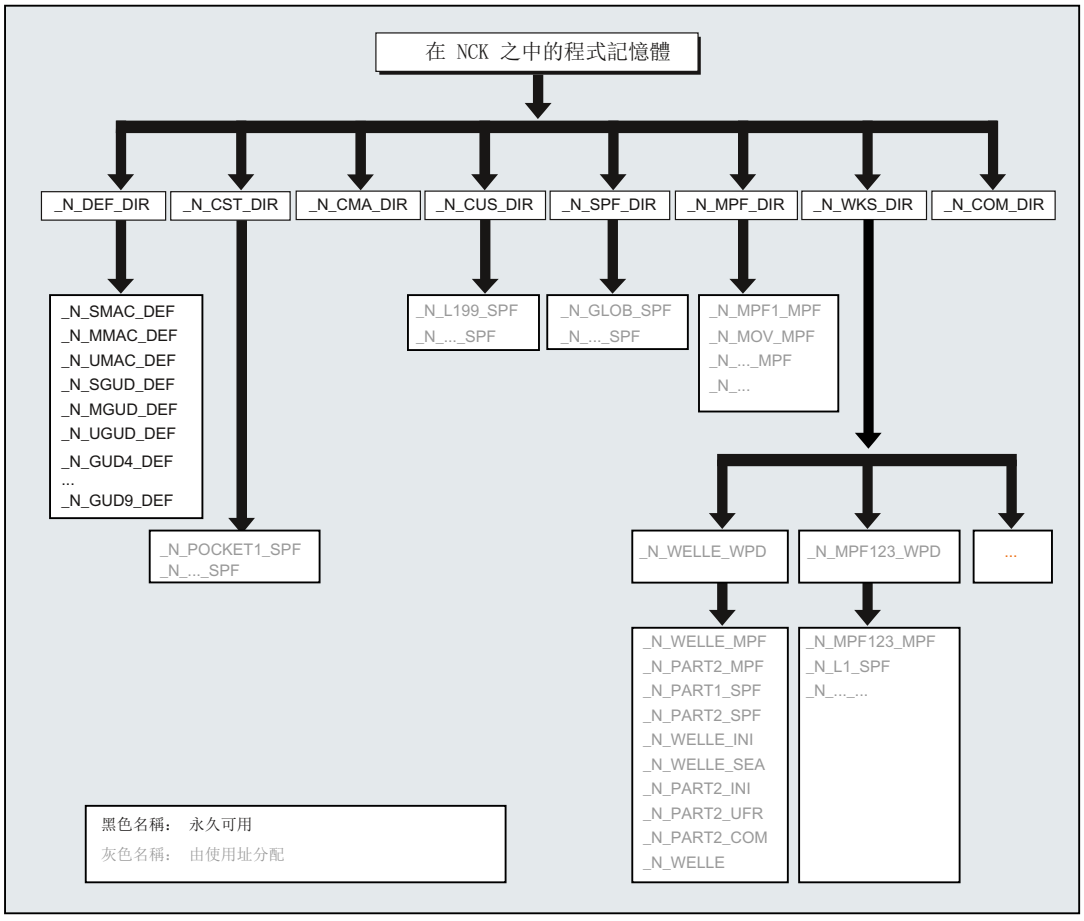
3.1.1 NC 程式記憶體

檔案與程式（例如，主程式與子程式、巨集定義）會儲存在非揮發性程式記憶體中（→ 被動檔案系統）。

參考資料：

功能手冊，延伸功能，記憶體設定（S7）

有些檔案類型也會暫時儲存於此；如有需要，它們也可以傳輸至工作記憶體（例如，對於在特定工件上加工的初始化目的）。



標準目錄

可使用下列標準目錄：

目錄	內容
_N_DEF_DIR	資料模組與巨集模組
_N_CST_DIR	標準循環
_N_CMA_DIR	製造商循環
_N_CUS_DIR	使用者循環
_N_WKS_DIR	工件
_N_SPF_DIR	全域子程式
_N_MPF_DIR	主程式
_N_COM_DIR	註解

檔案類型

主記憶體中可以儲存下列檔案類型：

檔案類型	說明
<名稱>_MPF	主程式
<名稱>_SPF	子程式
<名稱>_TEA	機械資料
<名稱>_SEA	設定資料
<名稱>_TOA	刀具偏移量
<名稱>_UFR	零點偏移量 / 框架
<名稱>_INI	Initialization files
<名稱>_GUD	全域使用者資料
<名稱>_RPA	R－參數
<名稱>_COM	註解
<名稱>_DEF	全域使用者資料與巨集之定義

工件主目錄（_N_WKS_DIR）

工件主目錄存在於程式記憶體的標準設定中，在_N_WKS_DIR 名稱底下。工件主目錄包含所有您已程式設計的工件之工件目錄。

工件目錄（..._WPD）

工件目錄包含了所有需要用來加工工件的檔案。這些可以是主程式、子程式、任何初始化程式及註解檔案。

第一次啟動工件程式時，會執行一次初始化程式，依選擇程式（依照機械資料 MD11280 $MN_WPD_INI_MODE）有所不同。

範例：

為 SHAFT 工件所建立的工件目錄 _N_SHAFT_WPD，包含了下列檔案：

檔案	說明
_N_SHAFT_MPF	主程式
_N_PART2_MPF	主程式
_N_PART1_SPF	子程式
_N_PART2_SPF	子程式
_N_SHAFT_INI	工件資料的一般初始化程式
_N_SHAFT_SEA	設定資料初始化程式
_N_PART2_INI	工件 2 程式資料的一般初始化程式
_N_PART2_UFR	工件 2 程式框架資料的初始化程式
_N_SHAFT_COM	註解檔案

資料也可以儲存在工件目錄中，並非 NC 直接要求做為加工之用。除了 ASCII 檔案之外，也可以是二進制檔案，例如 JPG 格式的圖像，或是 PDF 格式的說明。為使 NC 能夠將其解譯為二進制檔案，延伸檔必需對 NC 為已知（透過 MD17000 $MN_EXTENSIONS_OF_BIN_FILE S 調試時設定；下列的延伸檔在基本設定時預先設定：JPG、GIF、PNG、BMP、PDF、ICO、HTM）。

選擇加工用的工件

可以選擇一個工件目錄，用來在通道中執行。若有一個具有**相同名稱**的主程式，或只有一個單一的主程式（_MPF）儲存在此目錄中，便會自動被選擇執行。

參考資料：
操作手冊

3.1.2 外部程式記憶體

除了 NC 之中的被動檔案系統，外部程式記憶體也可於機台提供（例如本機磁碟或網路磁碟）。

使用「由外部執行」或「EES（由外部儲存）」功能，就可**直接**由外部程式記憶體執行工件程式。

參考資料：
功能手冊基本功能；K1：模式群組、通道、程式操作、重置回應

全域工件程式記憶體 (GDIR)

宣告驅動器時，可指定其中一個磁碟機至全域工件程式記憶體 (GDIR)。

參考資料：
操作手冊：章節：「管理程式」>「設定磁碟機」

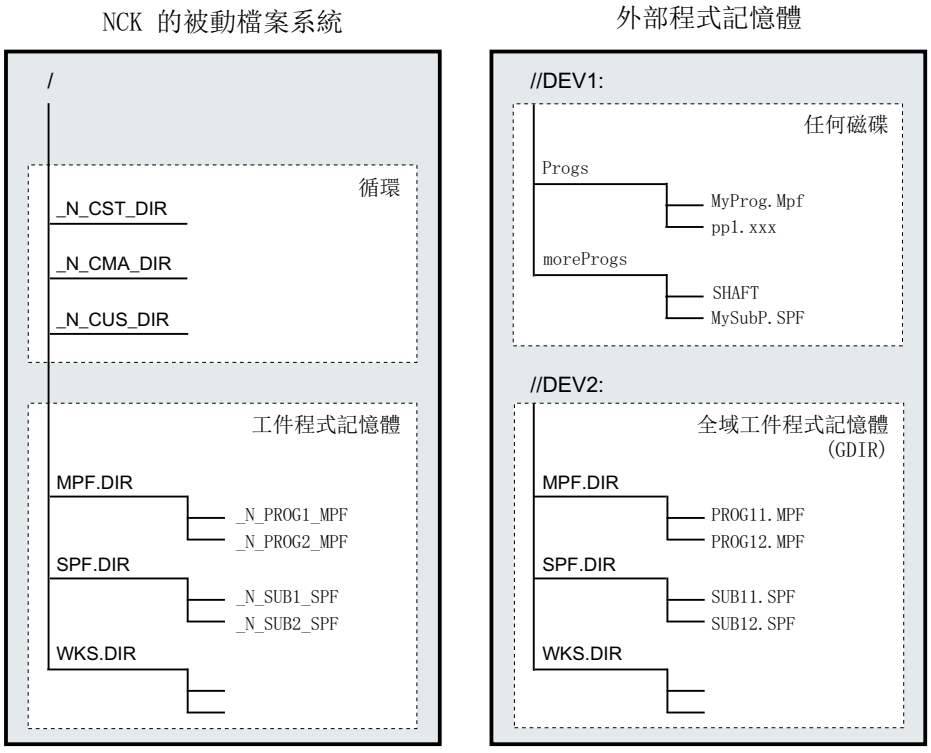
系統會在磁碟自動建立 MPF.DIR、SPF.DIR 及 WKS.DIR 目錄。這三個目錄來自 GDIR。

GDIR 僅針對 EES 函數扮演角色。視磁碟組態而定，GDIR 可取代或延伸 NC 工件程式記憶體。不過對 EES 操作而言，並不一定要建立 GDIR。

GDIR 的目錄及檔案可利用與被動檔案系統的相同方式，在工件程式之中定址。這樣可以讓具有路徑細節的 NC 程式，由被動檔案系統相容傳輸至 GDIR。GDIR 的目錄 SPF.DIR 位於子程式的搜尋路徑之中。

程式結構

下圖顯示外部程式記憶體的程式組織：



不區分大小寫的檔案系統

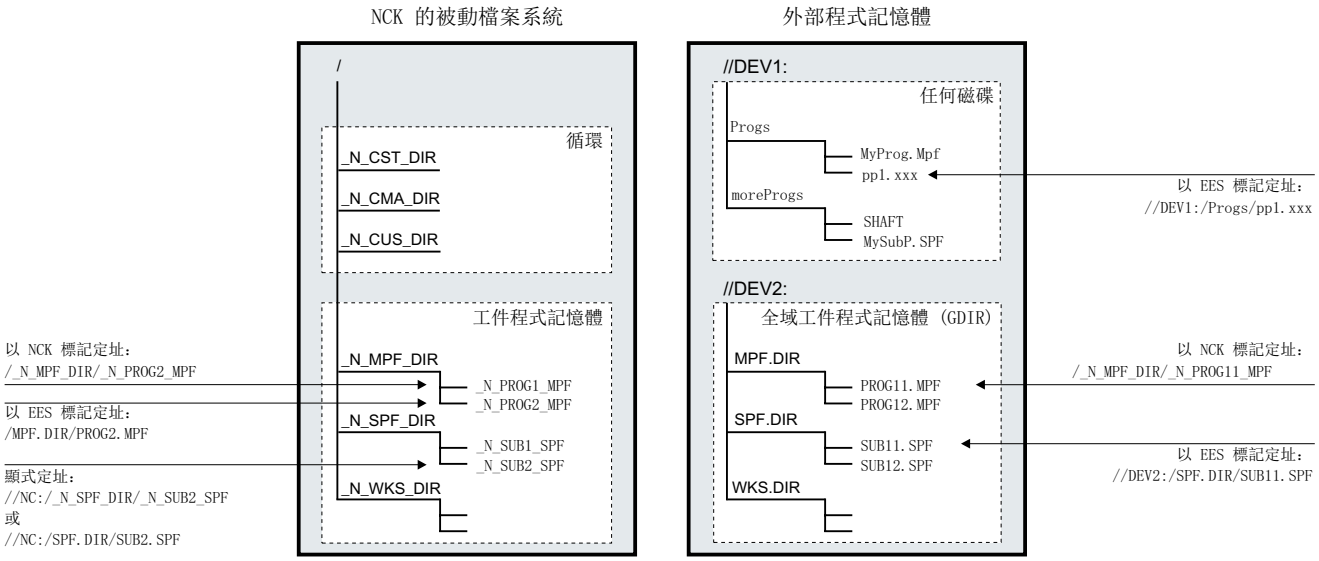
說明

為了避免檔案定址時發生大小寫問題（請參閱「定址程式記憶體檔案 (頁 237)」），應使用**不區分大小寫**檔案系統作為外部程式記憶體。

3.1.3 定址程式記憶體檔案

程式記憶體之中的檔案，是以檔案處理指令定址（例如 WRITE、DELETE、READ、ISFILE、FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO），並使用絕對路徑加上檔案名稱，或僅使用檔案名稱進行參照。在第二種情況下，選取程式的路徑會作為檔案路徑。

以 NC/EES 標記定址



定址被動檔案系統的檔案

被動檔案系統一般以 **NC 標記定址**（以網域識別碼「_N_」、「_」開頭的目錄及檔案名稱，是檔案識別碼分隔符號），不指定磁碟名稱。不過也允許以 **EES 標記**（目錄/檔案副檔名為「.」者無網域識別「_N_」）進行定址。

範例：

- NC 標記："/_N_SPF_DIR/_N_SUB1_SPF"
- EES 標記："/SPF.DIR/SUB1.SPF"

說明

被動檔案系統檔案以 EES 標記的定址方法，會依據下列規則於內部轉換為 NC 標記：

- 目錄及檔案名稱以網域識別「_N_」延伸。
- 若目錄或檔案名稱的倒數第四個字元為句點（「.」），將會轉換為底線（「_」）。

使用預先定義的驅動器名稱 "//NC:"，被動檔案系統也可以明確定址。

範例：

- NC 標記： "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- EES 標記："//NC:/SPF.DIR/SUB1.SPF"

定址外部程式記憶體檔案

未記錄為 GDIR 的外部程式記憶體檔案必須以 EES 標記定址。驅動器名稱（例如 "//DEV1:"）必須於定址路徑開始處指定。允許以 /user/sinumerik/hmi/cfg/logdrive.ini 設定的所有符號裝置名稱。

範例：

- EES 標記："//DEV1:/MyProgDir/pp1.xxx"
- NC 標記：不允許

定址全域工件程式記憶體 (GDIR) 檔案

定址 GDIR 檔案時，除了在 EES 標記指定路徑，也可使用 NC 標記指定路徑。

範例：

- EES 標記："//DEV2:/MPF.DIR/PROG11.MPF"
- NC 標記："/_N_MPF_DIR/_N_PROG11_MPF"

說明

GDIR 檔案於 NC 標記的定址方法，會依據下列規則於內部轉換為 EES 標記：
- 移除目錄及檔案名稱的網域識別「_N_」。
- 若目錄或檔案名稱的倒數第四個字元為底線（「_」），將轉換為句點（「.」）。

路徑規格的規則

完整路徑規格包含磁碟名稱、目錄路徑及檔案名稱。

磁碟名稱

磁碟名稱規格遵循下列規則：

- 允許以 /user/sinumerik/hmi/cfg/logdrive.ini 設定的所有符號裝置名稱。
- 字元「//」位於開始處，之後接續至少一個字母或一個數字。
- 下列字元可為任何字母、數字、「_」及空格組合。
- 名稱以字母或數字為結尾，並於之後接續「:」。
- 不允許使用其他特殊字元。

說明

驅動器名稱 "//NC:" 針對被動檔案系統預先定義。

範例：

- 外部程式記憶體：
  - //Drive1:
  - //Drive_1:
  - //Drive 1:
  - //A B:
  - //1 B C 2:

目錄路徑

目錄路徑規格遵循下列規則：

- 「/」位於目錄路徑的開始及結尾處，並作為個別路徑區段的分隔符號。

  說明

  目錄路徑之中**不允許**使用雙斜線（「//」）！

- 目錄名稱：
  - 目錄名稱必須以字母或數字開始。只有以 NC 標記定址時，目錄名稱才能以網域識別「_N_」開始。
  - 下列字元可為任何字母、數字及「_」組合。

    說明

    外部程式記憶體的目錄名稱也允許使用空格。不過若建立外部程式記憶體作為全域工件程式記憶體 (GDIR)，就不允許使用空格。

  - 不允許使用其他特殊字元。
- 目錄副檔名：
  - 目錄副檔名必須包含三個字母/數字，
  - 並以來自目錄名稱的「_」（NC 標記）或「.」（EES 標記）分隔。

  說明

  被動檔案系統只有 _DIR 及 _WPD 目錄副檔名。

範例：

- 被動檔案系統或 GDIR：
  - NC 標記： _N_WKS_DIR/_N_MYNCPROGS_WPD/...
  - EES 標記：WKS.DIR/MYPROGS.WPD/...
- 外部程式記憶體：
  - /abc
  - /ab_c.def
  - /ab c1.def
  - /a b c .d11
  - /abc.def/ghi.klm

檔案名稱

以下原則適用於檔案名稱：

- 只有以 NC 標記定址時，檔案名稱才能以網域識別「_N_」開始。
- 接下來兩個字元應為兩個字母，或是在一個字母後接著一個底線。

  說明

  若滿足此條件，則可從另一個程式中以指定的程式名稱呼叫該 NC 程式作為子程式。然而，若該程式名稱開頭為數字，則僅能透過 CALL 敘述呼叫該子程式。

- 下列字元可為任何字母、數字及「_」組合。
- 檔案副檔名：
  - 檔案副檔名必須包含三個字母/數字。

    說明

    被動檔案系統允許的檔案副檔名請參閱「NC 程式記憶體 (頁 233)」。

  - 並以來自檔案名稱的「_」（NC 標記）或「.」（EES 標記）分隔。

範例：

- 被動檔案系統或 GDIR：
  - NC 標記： _N_SUB1_SPF
  - EES 標記：SUB1.SPF
- 外部程式記憶體：
  - 第一部分
  - _Part1
  - Part_1.spf
  - Part1.mpf

DIN 子程式名稱

以下原則適用於 DIN 子程式名稱：

- 第一個字元必須為「L」。
- 後續字元為數字（至少一個）。
- 檔案副檔名：
  - 檔案副檔名必須包含三個字母。
  - 並以來自檔案名稱的「_」（NC 標記）或「.」（EES 標記）分隔。

範例：

- L123
- L1_SPF（NC 標記）或 L1.SPF（EES 標記）

最長路徑長度

指定磁碟名稱和目錄長度最多可使用 128 位元組，檔案名稱最長為 31 位元組。完整路徑最長為 159 位元組。

3.1.4 搜尋子程式呼叫的路徑

對於沒有路徑資料的子程式呼叫，絕對路徑是以處理固定搜尋路徑的方式決定。

之後會以下列順序在程式記憶體進行搜尋：

		目錄	說明
1		目前的目錄 / *名稱*	目前目錄為選取程式所在的目錄。 這可能是： • 在 NC 工件程式記憶體或全域工件程式記憶體的工件目錄或標準目錄 _N_MPF_DIR 或 • 外部程式記憶體的任何目錄
2		目前的目錄 / *名稱_SPF*	
3		目前的目錄 / *名稱_MPF*	
4	a	//NC:/_N_SPF_DIR / *名稱_SPF*	NC 工件程式記憶體的子程式目錄
	b	//DEV2:/_N_SPF_DIR / 名稱_SPF [1)]	全域工件程式記憶體的子程式目錄 **附註：** 若全域工件程式記憶體尚未建立，或是已在 NC 工件程式記憶體選擇程式，就不會執行此搜尋步驟。

	目錄	說明
5	以 CALLPATH 編程設定的搜尋路徑副檔名（請參閱「供副程式呼叫（CALLPATH）所用的延伸搜尋路徑 (頁 220)」）。 **附註：** 若 CALLPATH 尚未編程設定，就不會執行此搜尋步驟。	
6	/_N_CUS_DIR / *名稱_SPF*	使用者循環目錄
7	/_N_CMA_DIR / *名稱_SPF*	製造商循環目錄
8	/_N_CST_DIR / *名稱_SPF*	標準循環目錄

1) //DEV2:"對磁碟而言也是一樣，全域工件程式記憶體已於其中建立。

搜尋適用以下規定：

- 針對各個子程式呼叫執行搜尋路徑，也就表示這與較高層級程式所在位置無關。
- 會考慮不同的檔案類型，視目錄而定。
- 會在目錄進行搜尋，不會在較低層級（例如巢狀目錄）搜尋。

3.1.5 詢問路徑及檔案名稱

下列系統變數可於工件程式之中讀取，可用於詢問 NC 程式的路徑及檔案名稱：

系統變數	類型	含義
$P_STACK	INT	提供目前 NC 程式執行時所在的程式層級。
$P_PATH[<n>]	STRING	提供 NC 程式路徑；該路徑使用欄位索引 <n> 於選定的程式層級處理。 範例： $P_PATH[0] 提供主程式路徑，例如「/_N_WKS_DIR/_N_WELLE_WPD/」。 $P_PATH[$P_STACK - 1] 提供呼叫程式路徑。 若路徑指向 NC 程式，且該程式儲存於 NC 被動檔案系統或全域工件程式記憶體 (GDIR)，就會在 NC 標記提供路徑。 若路徑指向 NCK 程式，且該程式由全域工件程式記憶體以外的外部程式記憶體執行，則 $P_PATH 就會在 EES 標記提供路徑。

系統變數	類型	含義	
$P_PROG[<n>]	STRING	提供 NC 程式名稱；該名稱使用欄位索引 <n> 於選定的程式層級處理。 若 NC 程式儲存於 NC 的被動檔案系統，或是全域工件程式記憶體，就會在 NC 標記提供程式名稱。 若 NC 程式是由全域工件程式記憶體以外的外部磁碟執行，則 $P_PROG 就會在 EES 標記提供名稱。	
$P_PROGPATH	STRING	提供目前正在處理的 NC 程式路徑。 呼叫 $P_PROGPATH 與 $P_PATH[$P_STACK] 相同。	
$P_IS_EES_PATH[<n>]	BOOL	詢問 $P_PATH[<n>] 提供的路徑或 $P_PROG[<n>] 提供的程式名稱，是否與 NC 標記或 EES 標記相符。	
		= FALSE	$P_PATH[<n>] 及 $P_PROG[<n>] 提供 NC 標記。也就是說每個識別碼都有前置碼「_N_」。檔案識別碼的分隔符號為「_」。 範例： • NC 標記的路徑："/_N_WKS_DIR/_N_MYWPD_WPD/" • NC 標記之中的程式名稱：「_N_MYPROG_MPF」 NC 標記之中的路徑可指向 NC 的被動檔案系統，也可指向全域工件程式記憶體。
		= TRUE	$P_PATH[<n>] 及 $P_PROG[<n>] 提供 EES 標記。也就是說識別碼沒有前置碼「_N_」。檔案識別碼的分隔符號為「.」。 範例： • EES 標記的路徑："//DEV1:/WKS.DIR/MYWPD.WPD/" • EES 標記的程式名稱：「MYPROG.MPF」

<n>：索引 <n> 定義程式層級，應可由此讀取路徑資訊（值範圍：0 ... 17）

說明

在 EES 模式中，系統變數 $P_PROG, $P_PATH 及 $P_PROGPATH 路徑名稱位於 EES 標記，在全域工件程式記憶體 (GDIR) 之外。對 EES 模式而言，評估及處理這些路徑名稱的使用者程式必須延伸，才能處理 EES 標記之中的路徑名稱。

3.2 工作記憶體（CHANDATA，COMPLETE，INITIAL）

功能

工作記憶體包含目前的系統，以及和控制（主動檔案系統）一起操作的使用者資料，例如：

- 主動機械參數
- 刀具偏移量資料
- 零點偏移量
- ...

初始化程式

這些是和工作記憶體資料一起被初始化的程式。下列檔案類型可供使用：

檔案類型	說明
name_TEA	機械參數
name_SEA	設定資料
name_TOA	刀具偏移量
name_UFR	零點偏移量 / 框架
name_INI	Initialization files
name_GUD	全域使用者資料
name_RPA	R－參數

資料區

可在將要套用的不同區域中組織資料。例如，一個控制系統可以具有數個通道，或像大多數的情況一樣，會有數個軸供其自由使用。

包括：

識別碼	資料區
NC	NC 專屬資料
CH<n>	通道專屬資料（<n>指定通道名稱）
AX<n>	軸專屬資料（<n>指定機械軸的數量）
TO	刀具資料
COMPLETE	所有資料

在外部 PC 建立初始化程式

資料區域辨識碼和資料類型辨識碼，可以用來決定那個區域將在儲存資料時成為一個元件：

_N_AX5_TEA_INI	軸 5 的機械參數
_N_CH2_UFR_INI	通道 2 框架
_N_COMPLETE_TEA_INI	所有機械參數

初始啟動控制時，會自動載入一個集合的資料，以確保控制的適當操作。

多通道控制的步驟（CHANDATA）

提供數個通道使用的 CHANDATA（<通道編號>），僅在檔案_N_INITIAL_INI 中認可。這是已初始化的參考，其中所有資料的調試檔。

程式碼	註解
%_N_INITIAL_INI	
CHANDATA(1)	
	;加工軸指派，通道 1：
$MC_AXCONF_MACHAX_USED[0]=1	
$MC_AXCONF_MACHAX_USED[1]=2	
$MC_AXCONF_MACHAX_USED[2]=3	
CHANDATA(2)	
	;加工軸指派，通道 2：
$MC_AXCONF_MACHAX_USED[0]=4	
$MC_AXCONF_MACHAX_USED[1]=5	
CHANDATA(1)	
	;軸機械參數：
	;精確停止視窗粗調：
$MA_STOP_LIMIT_COARSE[AX1]=0.2	;軸 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2	;軸 2
	;精確停止視窗微調：
$MA_STOP_LIMIT_FINE[AX1]=0.01	;軸 1
$MA_STOP_LIMIT_FINE[AX1]=0.01	;軸 2

注意

CHANDATA 敘述

在工件程式中，CHANDATA 敘述只能針對執行 NC 程式的通道進行設定。這表示敘述可用於保護 NC 程式，使其不在錯誤通道中執行。

若發生錯誤則取消處理程式。

說明

工作清單中的 INI 檔案，不包含任何 CHANDATA 敘述。

儲存初始化程式（COMPLETE，INITIAL）

可將工作記憶體的檔案，儲存在外部 PC，並可再次從該處讀入。

- 以 COMPLETE 來儲存檔案。
- INITIAL 係用來在所有區域之上建立一個 INI 檔案（_N_INITIAL_INI）。

讀入初始化程式

注意
資料丟失 若以名稱「INITIAL_INI」讀入檔案，那麼檔案中所有未供應的資料，都會使用標準資料來初始化。唯有機械參數是例外。這表示**設定資料、刀具資料、ZO、GUD 值、...**係由標準資料（通常為「零」）供應。

例如，檔案 COMPLETE_TEA_INI 適用於讀入獨立機械參數。控制僅能預期在此檔案中的機械參數。這就是其他資料區域在此情況中仍保持不受影響的原因。

載入初始化程式

若 INI 程式僅使用一個通道的資料，那麼也能被選為，和被呼叫成為工件程式。這表示也有可能初始化程式控制資料。

保護區 4

4.1 定義保護區（CPROTDEF，NPROTDEF）

保護機台元件避免碰撞的保護區是定義在工件程式的單節中。這些將包含下列元素：

1. 加工平面的定義
   定義實際的保護區之前，必須先選擇加工平面，使得保護區的輪廓說明有所參考。
2. 定義的開始處
   依據特定的 NC 指令，可以建立通道專屬或機台專屬的保護區。
3. 保護區的輪廓說明
   保護區的輪廓是以移動動作定義。這些並不會執行，而且與前一個或接下來的幾何說明毫無關聯。只會定義保護區。
4. 定義結束

語法

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/...X/Y/Z...
...
EXECUTE(<Var>)
```

含義

DEF INT <Var>：	本地輔助說明變數、INTEGER 資料類型的定義
<Var>：	輔助說明變數名稱
G17/G18/G19：	加工平面 **說明：** 在定義結尾之前，不允許變更加工平面。在定義開始處與結尾之間，不允許程式設計備選座標。
CPROTDEF()：	預先定義的程序以定義一個**通道**專屬的保護區
NPROTDEF()：	預先定義的程序以定義一個**機台**專屬的保護區

<n>：	已定義之保護區號碼		
	資料類型：	INT	
<t>：	保護區類型		
	資料類型：	BOOL	
	值：	TRUE	**刀具**相關保護區
		FALSE	**工件**相關保護區
<AppLim>：	第三尺寸的限制類型		
	資料類型：	INT	
	值：	0	無限制
		1	增加方向上的限制
		2	減少方向上的限制
		3	正與負方向限制
<AppPlus>：	3 維中正向的限制值		
	資料類型：	REAL	
<AppMinus>：	3 維中負向的限制值		
	資料類型：	REAL	

G0/G1/...X/Y/Z... ...：	保護區輪廓於選擇的加工平面最多可指定 11 個移動動作。移動至輪廓會是第一個移動動作。輪廓說明最後的點必須與輪廓說明的第一個點相同。 有效的保護區是輪廓剩下的區域： ● 內部保護區 內部保護區的輪廓必須以逆時針方向說明。 ● 外部保護區（僅允許工件相關保護區） 外部保護區的輪廓必須以順時針方向說明。 允許使用以下的輪廓元素： ● G0、G1 表示直線輪廓元素 ● G2 表示順時針方向的圓弧節段 僅允許工件相關保護區。不允許刀具相關的保護區，因為必須為凸起的。 ● G3 表示於逆時針方向之圓弧區段 **說明：** 保護區不能用完整的圓形說明。完整的圓形必須分成兩個半圓形。 **說明：** 不允許 G2 → G3 或 G3 → G2 的順序！兩個圓形單節之間必須插入一個簡短的 G1 單節。
EXECUTE(<Var>)：	標示定義結尾的預先定義程序 使用 EXECUTE 切換返回一般的程式程序。

範例

請參閱「啟動 / 停用保護區（CPROT、NPROT） (頁 254)」下的範例。

其他資訊

機台專屬保護區

機台專屬的保護區或其輪廓係使用幾何軸定義，亦即參照通道的基本座標系統 (BCS)。為了讓機台專屬保護區已啟用的所有通道都能執行正確的保護區監控，所有相關聯通道的基本座標系統 (BCS) 都必須完全相同：

- 座標原點的位置參照至機台零點
- 座標軸的方向

輪廓說明的參考點

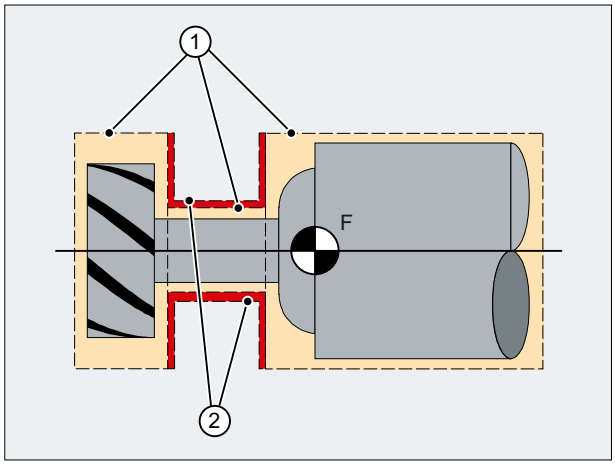
- 刀具相關保護區
  刀具相關保護區的座標必須指定為參照至**刀把參考點 F** 的絕對值。
- 工件相關保護區
  工件相關保護區的座標必須指定為參照至**基本座標系統 (BCS)** 零點的絕對值。

對稱於旋轉中心的保護區

如為軸對稱或旋轉對稱的保護區（例如主軸夾頭），則必須描述完整的輪廓，而不只是輪廓的旋轉中心。

刀具相關保護區

刀具相關保護區需永遠為凸面。若需要凹面保護區，它應被次分為數個凸面保護區。



① 凸起保護區
② 凸起保護區（**不允許！**）
F 刀把參考點

一般條件

定義保護區時，以下的功能切勿啟用或使用：

- 刀具半徑補正（銑刀半徑補正、刀鼻半徑補正）
- 轉換
- 參考點逼近 (G74)
- 逼近固定點 (G75)
- 停頓時間 (G4)
- 單節查找停止 (STOPRE)

- 程式結束（M17、M30）
- M 功能：M0、M1、M2

4.2 啟動 / 停用保護區（CPROT、NPROT）

先前在工件程式中定義的保護區可以隨時啟用，或者也可以預先啟用，以供 PLC 使用者程式稍後啟動。啟用保護區可以隨時停用。

在啟用或預先啟用時，也可以相對平移保護區的參考點。

說明

只有在保護區已啟用的通道之所有幾何軸都參照以後，才會將保護區納入考量。

說明

監控保護區

如果與刀具相關的保護區並未啟用，則刀具路徑的檢核將針對與工件相關的保護區。
如果工件導向的保護區已啟用，則沒有保護區監控。

語法

```
CPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
NPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
```

含義

CPROT：	預先定義的程序用來啟動**通道**專屬的保護區		
NPROT：	預先定義的程序用來啟動**機台**專屬的保護區		
<n>：	保護區編號		
	資料類型：	INT	
<狀態>：	通道專屬的啟用狀態是利用此參數設定		
	資料類型：	INT	
	值：	0	停用保護區
		1	預啟動保護區
		2	啟動保護區
		3	預啟動含條件停止之保護區

<XMov>、<YMov>、 <ZMov>：	在 X/Y/Z 方向上的附加偏移量 偏移量可於 1、2 或 3 尺寸發生。偏移量值參照至： • **工件**相關保護區的機械零點 • **刀具**專屬保護區的刀盤參考點 F	
	資料類型：	REAL

範例

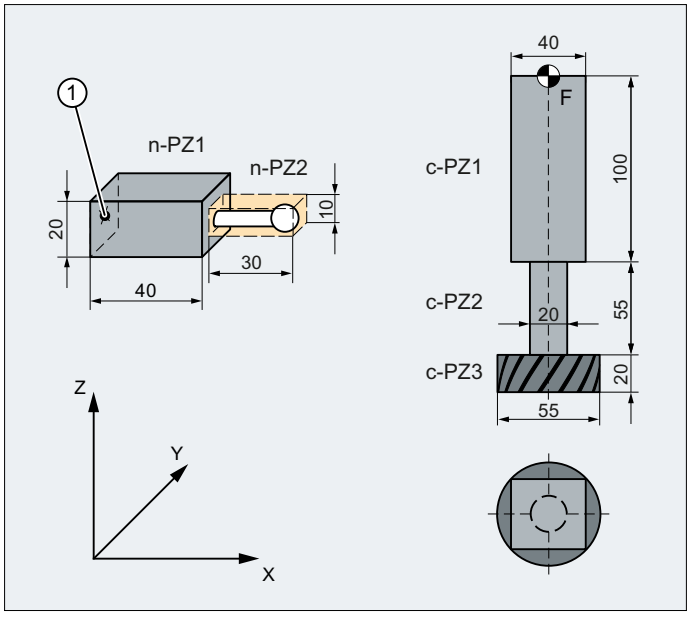
銑刀與量測探針間可能的碰撞會在銑削機台上監控。量測探針之位置可在功能啟動時以偏移量定義。

為此，已定義下列保護區：

- 測量探針座 (n-PZ1) 與測量探針本身 (n-PZ2) 之機台專屬與工件相關的保護區。
- 銑刀刀盤 (c-PZ1)、銑刀握把 (c-PZ2) 及銑刀本身 (c-PZ3) 之通道專屬與刀具相關保護區。

於 Z 方向之所有保護區的方向。

功能啟用時量測探針之參考點位置必須為 X = -120、Y = 60 及 Z = 80。



① 探針保護區名稱

F 刀把參考點

程式碼	註解
DEF INT PROTZONE	；輔助說明變數定義
G17	；加工平面 XY
；定義保護區：	
NPROTDEF(1,FALSE,3,10,-10)	；保護區 n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE(PROTZONE)	
NPROTDEF(2,FALSE,3,5,-5)	；保護區 n-PZ2
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE(PROTZONE)	
CPROTDEF(1,TRUE,3,0,-100)	；保護區 c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	
Y-20	
EXECUTE(PROTZONE)	
CPROTDEF（2，TRUE，3，-100，-150）	；保護區 c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE(PROTZONE)	
CPROTDEF（3，TRUE，3，-150，-170）	；保護區 c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE(PROTZONE)	
；啟動保護區：	
NPROT(1,2,-120,60,80)	；以偏移量啟動保護區 n-PZ1
NPROT(2.2,-120,60,80)	；以偏移量啟動保護區 n-PZ2
CPROT(1,2,0,0,0)	；啟動保護區 c-PZ1
CPROT(2,2,0,0,0)	；啟動保護區 c-PZ2
CPROT(3,2,0,0,0)	；啟動保護區 c-PZ3

其他資訊

控制器開啟電源後啟動狀態

控制系統開啟電源後保護區可能已經啟用，軸已經被參照。如果保護區是處於這種狀態，則將以下的系統變數設定為 TRUE：

- $SN_PA_ACTIV_IMMED[<n>]（機台專屬保護區）或
- $SC_PA_ACTIV_IMMED[<n>]（通道專屬保護區）
  索引「<n>」對應於保護區編號：0 = 1。保護區

以狀態 = 2 啟動保護區，且無偏移量。

保護區的多重啟動

機台專屬保護區可於數個通道中同時啟動（例如有兩個相反面的尾座保護區）。保護區僅當已參考所有幾何軸之後監控。

在單一通道中保護區無法以不同的偏移量同時啟動。

生效刀具補正之保護區監控

進行刀具半徑補正時，僅在刀具半徑補正平面與保護區定義平面相同時，保護區監控功能方可產生作用。

4.3 檢查保護區差異、工作區限制與軟體極限開關（CALCPOSI）

功能

在工件座標系統 (WCS) 中，CALCPOSI 功能會檢查**幾何軸**是否可以從起點開始移動指定的距離而沒有違反啟用限制。在因為限制而造成距離無法完全移動的情況下，則傳回十進位編碼的狀態正值及最大可能移動距離。

定義

```
INT CALCPOSI(VAR REAL[3] <起始>, VAR REAL[3] <Dist>, VAR REAL[5] <限制>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

語法

```
<狀態> = CALCPOSI(VAR <起始>, VAR <Dist>, VAR <限制>, VAR <MaxDist>, <MeasSys>, <TestLim>)
```

含義

CALCPOSI(...)：	用來測試幾何軸是否違反限制的預先定義功能	
	前置處理停止：	否
	單獨在單節中：	是

<狀態>： （第 1 部分）	函數傳回值。負值代表錯誤狀態。		
	資料類型：		INT
	值域：		-8 ≤ x ≤ 100000
	值：	0	距離可以完全移動。
		-1	在<限制>中至少一元件為負。
		-2	轉換計算中的錯誤。 範例：移動距離通過奇點，因此無法定義軸位置。
		-3	指定的移動距離<Dist>及最大可能移動距離 <MaxDist> 為線性相依。 **注意** 只能與 <TestLim> 搭配使用，位元 4 == 1。
		-4	包含在<Dist>中的移動方向在限制表面上的投影為零向量，或者移動方向垂直於違反的限制表面。 **注意** 只能與 <TestLim> 搭配使用，位元 5== 1。
		-5	在 <TestLim> 中，位元 4 == 1 及位元 5 == 1
		-6	檢查移動限制是否未被參考時，需將至少一機械軸納入考量。
		-7	碰撞避免功能運動鏈或保護區的無效定義。
		-8	碰撞避免功能由於記憶體不足，無法執行此一指令。
<狀態>： （第二部分）	個位數 **注意** 若同時違反多個限制，則指定的移動距離上最大限制會發出警報。		
	值：	1	軟體限制開關限制了移動距離
		2	工作區限制限制了移動距離
		3	保護區限制了移動距離
		4	碰撞避免功能：保護區限制了移動路徑
	十位數：		
	值：	1x	初始值違反限制
		2x	指定的直線違反限制。 即使終點無違反任何限制本身，但自起點至終點之路徑會導致一臨界值違反（如經過保護區、非線性轉換於 WCS 弧線軟體限制開關），亦將傳回該值。

<狀態>： （第三部分）	百位數		
	值：	1xx	AND 個位數 == 1 或 2： 已違反正臨界值。
			及個位數 == 3 [1]： 已違反 NC 專屬的保護區。
		2xx	AND 個位數 == 1 或 2： 已違反負臨界值。
			AND 個位數 == 3 [1]： 違反了通道專屬保護區。
<狀態>： （第四部分）	千位數		
	值：	1xxx	AND 個位數 == 1 或 2： 違反限制的軸編號相乘的係數。軸從 1 開始編號。 參考： • 軟體限制開關機械軸 • 工作區限制：幾何軸 AND 個位數 == 3 [1]： 已違反保護區編號相乘的係數。
<狀態>： （第 5 部分）	十萬位數		
	值：	0xxxxx	十萬位數 == 0：<Dist> 保持不變
		1xxxxx	以 <Dist> 傳回方向向量，定義限制平面上下一步動作的方向。 只發生於以下的補充條件： • 已違反軟體限制開關或工作區限制（未在起點） • 轉換未啟用 • <TestID>，位元 4 或位元 5 == 1
<起始>：	參考起始位置的向量： • <起始> [0]：第一幾何軸 • <起始> [1]：第二幾何軸 • <起始> [2]：第三幾何軸		
	參數類型：		輸入
	資料類型：		VAR REAL [3]
	值域：		-最大 REAL 值 ≤ x[<n>] ≤ +最大 REAL 值

<Dist>：	參考向量。	
	輸入：增量移動距離 • **<Dist>** [0]：第一幾何軸 • **<Dist>** [1]：第二幾何軸 • **<Dist>** [2]：第三幾何軸	
	輸出（僅供 **<**狀態**>** 設定十萬位數）：	
	<Dist> 包含一個做為輸出值的單位向量 **v**，定義在 WCS 中的下一個移動方向。	
	案例 1：**<TestID>** 的向量 **v** 格式，位元 4 == 1 輸入向量 **<Dist>** 及 **<MaxDist>** 跨越動作平面。此平面由已違反的限制平面所切割。兩個平面的交線定義了向量 **v** 的方向。選取方位（符號），使得輸入向量 **<MaxDist>** 與 **v** 之間的角度不大於 90 度。	
	案例 2：**<TestID>** 的向量 **v** 格式，位元 5 == 1 向量 **v** 包是含在限制平面上 **<Dist>** 的移動向量投影方向的單位向量。若限制平面上移動向量的投影是零向量，傳回錯誤。	
	參數類型：	輸入/輸出
	資料類型：	VAR REAL [3]
	值域：	-最大 REAL 值 ≤ x[<n>] ≤ +最大 REAL 值

<限制>：	參考長度 5 的陣列。 ● **<限制> [0 - 2]**：幾何軸至限制的最小間隙： – **<限制> [0]**：第一幾何軸 – **<限制> [1]**：第二幾何軸 – **<限制> [2]**：第三幾何軸 如下觀察最小間隙： – 工作區限制：無限制 – 軟體限制開關若未啟用轉換，或是啟用轉換而幾何軸可以清楚指派至線性機械軸，例如 5 軸轉換。 ● **<限制> [3]**：包含線性機械軸的最小間隙，例如，因為非線性轉換而無法指派幾何軸。此數值亦用來當作監控傳統保護區以及碰撞避免保護區的臨界值。 ● **<限制> [4]**：包含旋轉機械軸的最小間隙，例如，因為非線性轉換而無法指派幾何軸。 **注意** 此數值只有在監控特殊轉換的軟體極限開關時啟用。	
	參數類型：	輸入
	資料類型：	VAR REAL [5]
	值域：	-最大 REAL 值 ≤ x[n] ≤ +最大 REAL 值
<MaxDist>：	參考使用增量移動距離的向量，其中沒有任何一個相關的機械軸違反軸臨界值指定的最小間隙。 ● **<Dist> [0]**：第一幾何軸 ● **<Dist> [1]**：第二幾何軸 ● **<Dist> [2]**：第三幾何軸 若未限制移動距離，此傳回參數之內容與 **<Dist>** 的內容相同。	
	<TestID>，位元 4 == 1：**<Dist>** 及 **<MaxDist>** **<MaxDist>** 及 **<Dist>** 必須包含跨越動作平面的向量做為輸入值。這兩個向量必須彼此線性獨立。**<MaxDist>** 的絕對值為任意值。計算動作方向，請參閱 **<Dist>** 的說明。	
	參數類型：	輸出
	資料類型：	VAR REAL [3]
	值域：	-最大 REAL 值 ≤ x[<n>] ≤ +最大 REAL 值

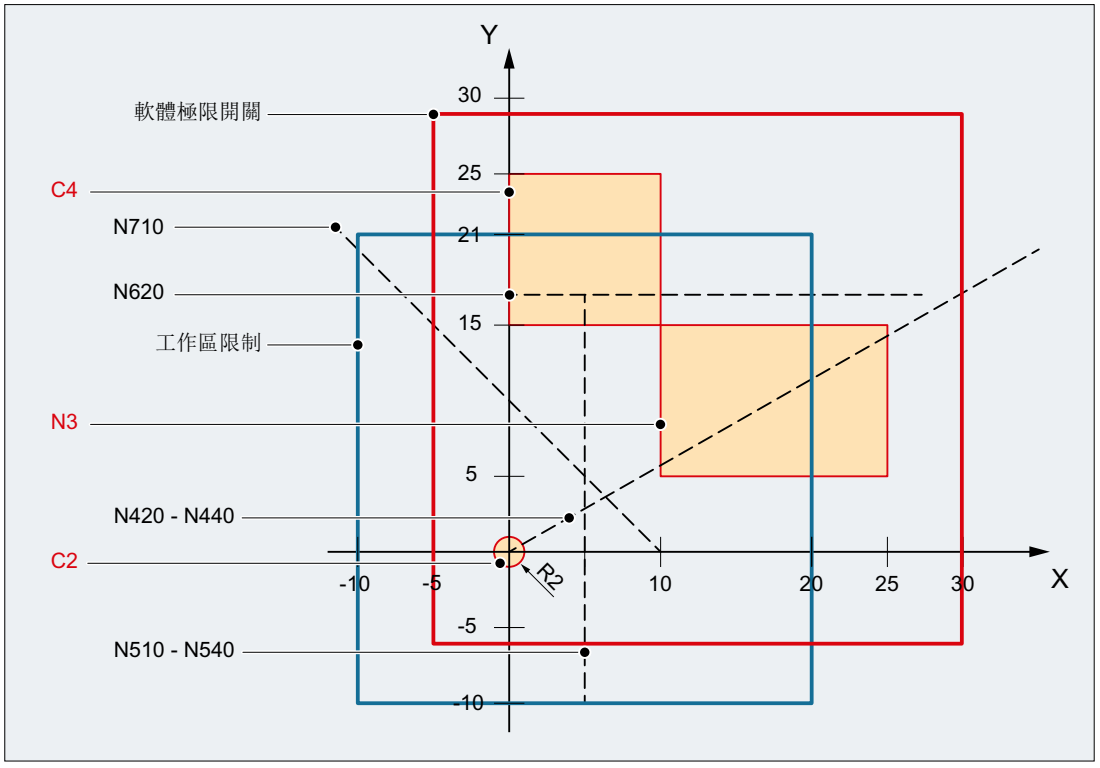
<MeasSys>：	位置及距離規範的量測系統（英制/公制）（**選配**）		
	資料類型：		BOOL
	值：	FALSE（預設）	對應於 G 群組 13 目前啟用的 G 指令（G70、G71、G700、G710）的單位系統。 **附註** 若已啟用 G70 且基本系統為公制（或 G71 已啟用且基本系統為英制），將於基本系統中提供系統變數 $AA_IW 及 $AA_MW，且使用則必需轉換為 CALCPOSI。
		TRUE	依據設定基本系統的單位系統： MD52806 $MN_ISO_SCALING_SYSTEM

<TestLim>：	欲監控之臨界值的位元編碼（**選配**）		
	資料類型：		INT
	預設值：		位元 0, 1, 2, 3, 6, 7 == 1 (207)
	Bit	**十進制**	**含義**
	0	1	軟體極限開關
	1	2	工作區限制
	2	4	已啟用的傳統保護區
	3	8	預先啟用的傳統保護區
	4	16	使用 **<Dist>** 中已違反的軟體極限開關或工作區限制，如**案例 1**（如上所示）傳回移動方向。
	5	32	使用 **<Dist>** 中已違反的軟體極限開關或工作區限制，如**案例 2**（如上所示）傳回移動方向。
	6	64	已啟用的碰撞避免保護區
	7	128	預先啟用的碰撞避免保護區
	8	256	已啟用及預先啟用的碰撞避免保護區

[1] 若同時違反多個保護區，則傳回指定的移動距離上最大限制的保護區。

範例

限制



在範例中，顯示了啟用的軟體極限開關及 X-Y 平面中的工作區限制，以及下列三個保護區：

- C2：刀具相關，通道專屬保護區，啟用，圓弧，半徑 = 2 mm
- C4：工件相關，通道專屬保護區，預先啟用，正方形，側邊長度 = 10 mm
- N3：機台專屬保護區，啟用，長方形，側邊長度 = 10 mm x 15 mm

NC 程式

首先在 NC 程式中定義保護區及工作區限制。接著用不同的參數指派呼叫 CALCPOSI() 功能。

程式碼

```
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
```

程式碼

```
；刀具相關的保護區 C2
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
；工件相關保護區 C4
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
；機台專屬保護區 N3
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
；啟用或預先啟用保護區
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
；定義工作區限制
N290 G25 XX=-10 YY=-10
N300 G26 XX=20 YY=21
N310 _START[0] = 0.
N320 _START[1] = 0.
N330 _START[2] = 0.
N340 _DIST[0] = 35.
N350 _DIST[1] = 20.
N360 _DIST[2] = 0.
N370 _LIMIT[0] = 0.
N380 _LIMIT[1] = 0.
N390 _LIMIT[2] = 0.
N400 _LIMIT[3] = 0.
N410 _LIMIT[4] = 0.
N420 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)
N430 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,3)
N440 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,1)
N450 _START[0] = 5.
N460 _START[1] = 17.
N470 _START[2] = 0.
```

程式碼

```
N480 _DIST[0] = 0.
N490 _DIST[1] =-27.
N500 _DIST[2] = 0.
N510 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,14)
N520 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N530 _LIMIT[1] = 2.
N540 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N550 _START[0] = 27.
N560 _START[1] = 17.1
N570 _START[2] = 0.
N580 _DIST[0] =-27.
N590 _DIST[1] = 0.
N600 _DIST[2] = 0.
N610 _LIMIT[3] = 2.
N620 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,12)
N630 _START[0] = 0.
N640 _START[1] = 0.
N650 _START[2] = 0.
N660 _DIST[0] = 0.
N670 _DIST[1] = 30.
N680 _DIST[2] = 0.
N690 TRANS X10
N700 AROT Z45
N710 _STATUS = CALCPOSI(_START,_DIST, _LIMIT, _MAXDIST)
；再次從 N690 及 N700 中刪除框架
N720 TRANS
N730 _START[0] = 0.
N740 _START[1] = 10.
N750 _START[2] = 0.
；向量 _DIST 及 _MAXDIST 定義動作平面
N760 _DIST[0] = 30.
N770 _DIST[1] = 30.
N780 _DIST[2] = 0.
N790 _MAXDIST[0] = 1.
N800 _MAXDIST[1] = 0.
N810 _MAXDIST[2] = 1.
N820 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,17)
N830 M30
```

CALCPOSI() 的結果

N...	<狀態>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ Y	備註
420	3123	8.040	4.594	N3 已違反。
430	1122	20.000	11.429	未監控任何保護區， 違反工作區限制。
440	1121	30.000	17.143	只有軟體限制監控仍生效。
510	4213	0.000	0.000	起點違反 C4
520	0000	0.000	-27.000	不監控預先啟用 C4。指定的距離可以完全移動。
540	2222	0.000	-25.000	由於 _LIMIT[1] = 2，移動距離受到工作區限制。
620	4223	-13.000	0.000	由於 C2 與 _LIMIT[3]，至 C4 之間隙共 4 mm。間隙 C2 → N3 為 0.1 mm 且不會導致移動距離受到限制。
710	1221	0.000	21.213	啟用平移及旋轉的框架。_DIST 中可允許的移動距離套用在偏移與旋轉的 WCS。
820	102121	18.000	18.000	已違反 Y 軸的軟體極限開關。需使用 <_TESTLIM> = 17 計算進一步的移動方向。此方向在_DIST (0.707, 0.0, 0.707)。由於十萬位數設定在 <_STATUS> 中，因此有效。

其他資訊

「參考」軸狀態

所有 CALCPOSI() 考量的機械軸都必須返回原點。

圓形相關的距離規範

所有的圓形相關距離規範**必須**解譯為半徑規格。對於已啟用直徑程式設計的移動軸，尤其必需納入考量 (DIAMON/DIAM90)。

移動距離縮短

如果軸的指定移動距離受限制，其他軸的移動距離也在<MaxDist>返回值中等比例縮短。因此，結果終點仍在指定路徑上。

旋轉軸

僅在旋轉軸不是模數旋轉軸的情況下進行監控。

允許不針對一或多個相關軸定義軟體極限開關、工作區限制或保護區。

軟體極限開關及工作區限制狀態

軟體極限開關及工作區限制只有在執行 CALCPOSI() 時啟用，才會納入考量。狀態會受到影響，例如，透過：

- 機台參數：MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS
- 設定資料：$AC_WORKAREA_CS_...
- NC/PLC 介面訊號 DB31、 ... DBX12.2 / 3
- 指令：WALIMON / WALIMOF

軟體極限開關及轉換

使用 CALCPOSI()，機械軸的位置 (MCS) 無法在各種動態轉換時（例如 TRANSMIT）從幾何軸 (WCS) 的位置唯一決定，因為在移動距離的特定位置不明確。於一般移動操作中，唯一性通常從 WCS 中對應至 MCS 連續移動之連續移動記錄與條件所產生。因此，監控軟體極限開關時，使用 CALCPOSI() 執行時的機台位置以解決這些情況中不明確的情形。

說明

前置處理停止

搭配使用 CALCPOSI() 轉換時，使用者必需自行負責程式設計前置處理停止 (STOPRE) 在 CALCPOSI() 前置處理之前，以同步化機械軸位置。

保護區間隙及傳統保護區

使用傳統保護區，**無法**保證指定路徑執行移動動作時，所有的保護區都可以維持設定在參數 <Limit>[3] 中的安全間距。它只能保證從 <Dist> 傳回的終點在移動方向上延伸安全間距時，不會有保護區違反。但是，直線可以非常靠近保護區通過。

保護區間隙及碰撞避免保護區

使用碰撞避免保護區，可以保證指定的移動路徑執行移動動作時，所有的保護區都可以維持設定在參數 <Limit>[3] 中的安全間距。

指定在參數 <Limit>[3] 的安全間距只有以下應用時才會生效：

<Limit>[3] > (MD10619 $MN_COLLISION_TOLERANCE)

若在參數 <TestLim> 中設定位元 4（計算接下來的方向），則 <DIST> 接收到的方向向量只有在十萬位數設定在函數傳回值（<狀態>）時有效。若是無法判定這樣的方向，可能是因為違反了保護區，也可能是因為啟用了轉換，導致 <DIST> 的輸入值維持不變。未輸出其他的錯誤訊息。

特殊動作指令

5

5.1 逼近編碼位置（CAC、CIC、CDC、CACP、CACN）

您可使用下列指令透過儲存於機械資料表中的固定軸位置之位置編號移動線性及旋轉軸。此類型的程式設計稱為「逼近已編碼位置」。

語法

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

含義

CAC(<n>)：	從位置編號 n 逼近已編碼位置
CIC(<n>)：	從實際位置編號開始，逼近已編碼位置 n 向前（+n）或向後（-n）定位
CDC(<n>)：	沿最短路徑從位置編號 n 逼近位置 （僅供旋轉軸）
CACP(<n>)：	從位置編號 n 以正方向逼近已編碼位置 （僅供旋轉軸）
CACN(<n>)：	從位置編號 n 以負方向逼近已編碼位置 （僅供旋轉軸）
<n>：	於機械資料表中的位置編號 值域： 0、1、...（表位置的最大編號－1）

範例： 逼近定位軸的已編碼位置

程式設計碼	註解
N10 FA[B]=300	；定位軸 B 的進給率
N20 POS[B]=CAC（10）	；從位置編號 10 逼近已編碼位置
N30 POS[B]=CIC（-4）	；從「目前位置編號」- 4 逼近已編碼位置

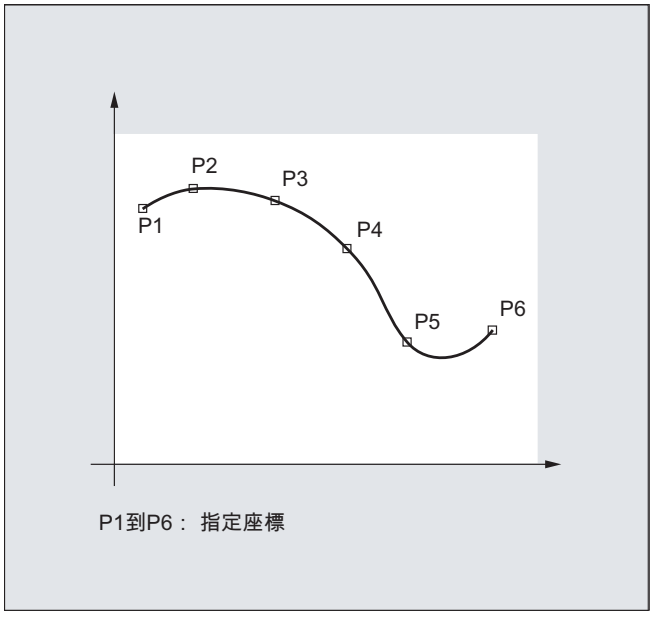
參考資料

- 功能手冊延伸功能；索引軸（T1）
- 功能手冊，已同步動作

5.2 曲線插補（ASPLINE、BSPLINE、CSPLINE、BAUTO、BNAT、BTAN、EAUTO、ENAT、ETAN、PW、SD、PL）

隨機曲線工件輪廓無法在分析形狀中精確地定義。 這就是這些輪廓類型僅沿著曲線使用有限點近似之原因，例如數位化表面時。 沿著曲線的點必須相連已定義輪廓，已產生工件的數位化平面。 曲線插補允許此動作。

曲線能定義二次或三次多項式形成的曲線。 沿著曲線之曲線的點特性可視所使用之曲線類型定義。



SINUMERIK solution line 提供下列曲線類型：

- A 曲線
- B 曲線
- C 曲線

語法

一般：

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

可為 B 曲線額外程式設計：

```
PW=<n>
SD=2
PL=<value>
```

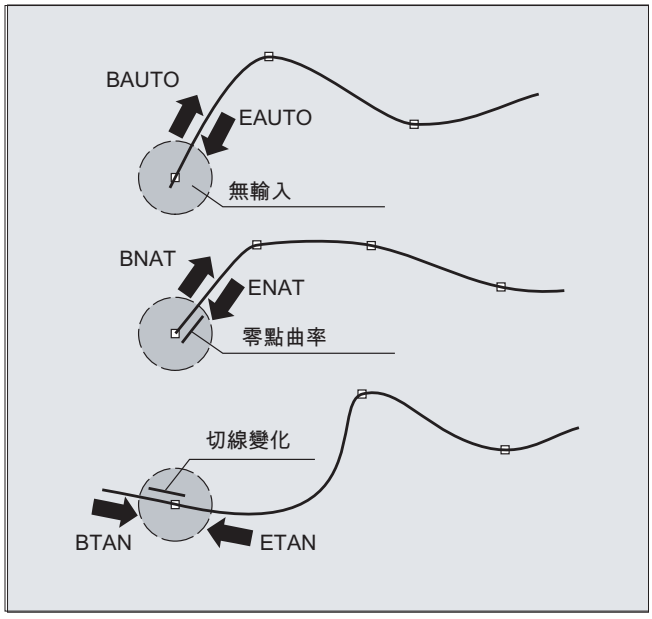
可為 A 與 C 曲線額外程式設計：

BAUTO / BNAT / BTAN

EAUTO / ENAT / ETAN

含義

曲線插補類型：			
ASPLINE：	啟動 A 曲線插補的指令		
BSPLINE：	啟動 B 曲線插補的指令		
CSPLINE：	啟動 C 曲線插補的指令		
	ASPLINE，BSPLINE 與 CSPLINE 指令模態有效且屬於動作指令群組。		
沿著曲線的點與檢查點：			
X... Y... Z... A... B... C...	直角座標位置		
點重（僅於 B 曲線）：			
PW：	使用 PW 指令，可為沿著曲線上之所有點程式設計所謂的「點重」。		
<n>：	「點重」		
	值域：	0 ≤ n ≤ 3	
	Increment：	0.0001	
	影響：	n > 1	檢查點會更吸引曲線。
		n < 1	檢查點會較不吸引曲線。
曲線角度（僅 B 曲線）：			
SD：	標準會使用三度多邊形，不過若是程式設計 SD=2，則將使用二度多邊形。		
節點間的距離（僅 B 曲線）：			
PL：	節點間的距離適合於內部計算。控制也可加工預定義所謂使用 PL 指令於參數間隔長度指定之節點間距。		

<值>：	參數間隔長度	
	值域：	作為路徑尺寸
於曲線開始的變化行為（僅 A 或 C 曲線）：		
BAUTO：	變化行為無規格。 開始由第一點的位置決定。	
BNAT：	無彎曲	
BTAN：	切線變化至上個單節（刪除位置）	
於曲線結尾的變化行為（僅 A 或 C 曲線）：		
EAUTO：	變化行為無規格。 結尾由最後一點的位置決定。	
ENAT：	無彎曲	
ETAN：	切線變化至上個單節（刪除位置）	
		

說明

可程式設計之變化行為對於 B 曲線無影響。 B 曲線永遠為切線以檢查多邊形的起點與終點。

補充條件

- 可能會使用刀具半徑補正。
- 會於投射之平面上進行碰撞監控。

範例

範例 1：B 曲線

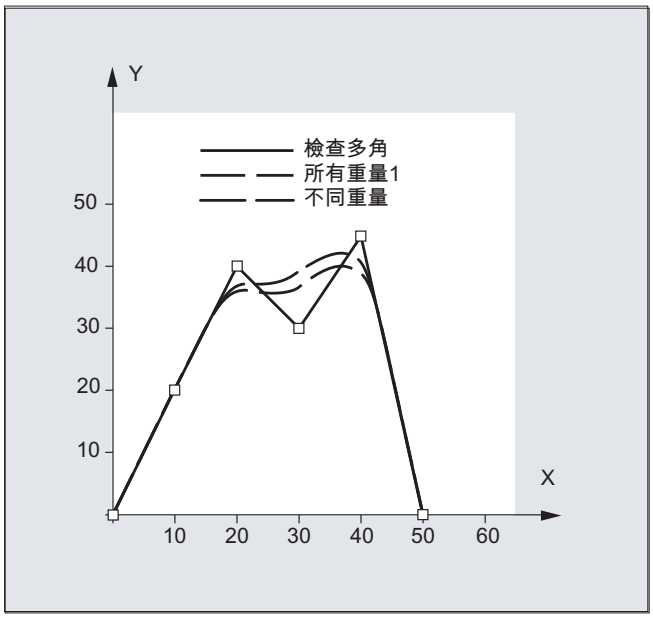
程式碼 1（所有重量 1）

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

程式碼 2（不同重量）

```
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
```

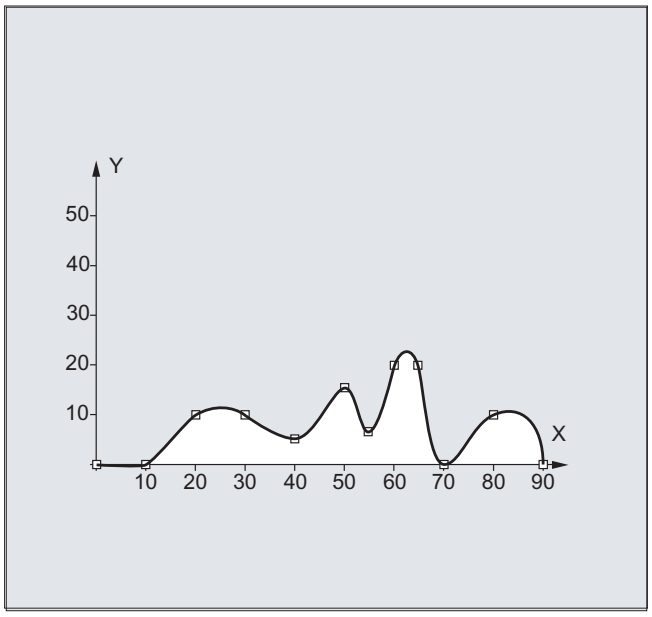
程式碼 3（檢查多邊形）	註解
N10 G1 X0 Y0 F300 G64	
N20	；不適用
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	



範例 2： C 曲線，於開始及結尾無彎曲

程式碼

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
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
```



範例 3： 曲線插補（A 曲線）及座標轉換（ROT）

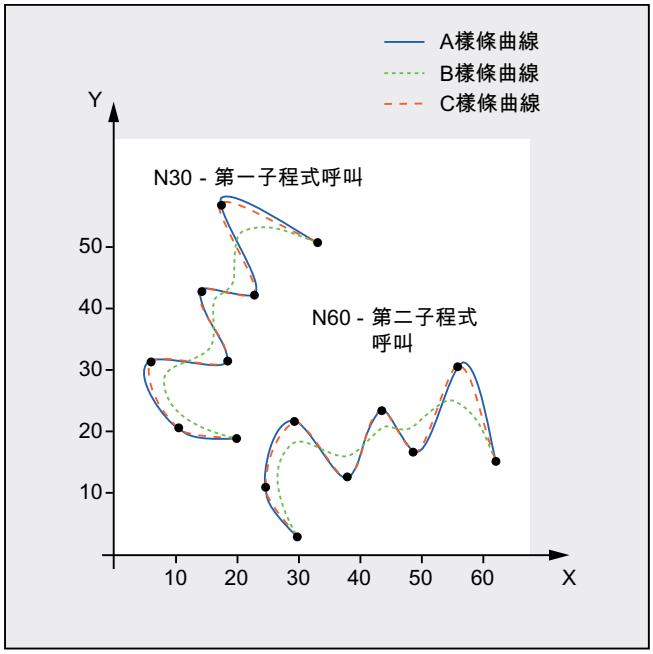
主程式：

程式碼	註解
N10 G00 X20 Y18 F300 G64	；逼近起點。
N20 ASPLINE	；啟動插補類型 A 曲線。
N30 CONTOUR	；第一個子程式呼叫。
N40 ROT Z-45	；座標轉換： WCS 透過 Z 軸-45°的旋轉。
N50 G00 X20 Y18	；逼近輪廓起點。
N60 CONTOUR	；第二個子程式呼叫。
N70 M30	；程式結束

子程式「contour」（包括沿著曲線之點的座標）：

程式碼
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

除了曲線，從範例程式（ASPLINE）產生的結果，下列圖示包含了啟動 B 或 C 曲線插補（BSPLINE、CSPLINE）可獲得的曲線：



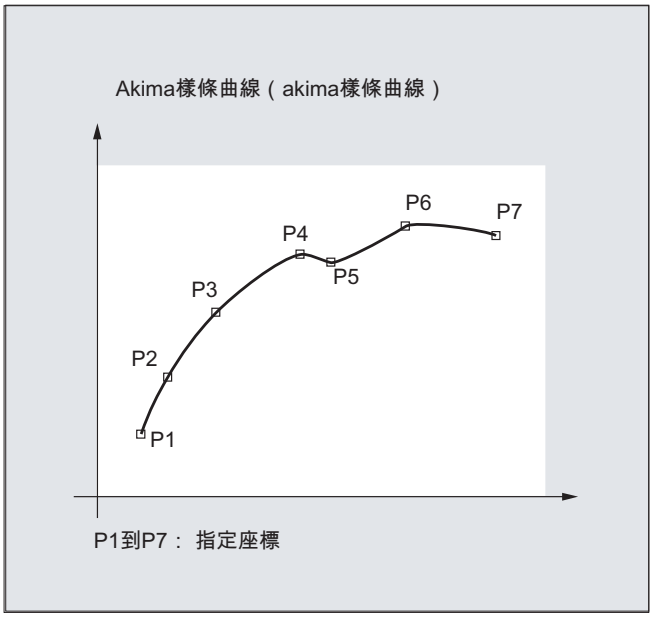
其他資訊

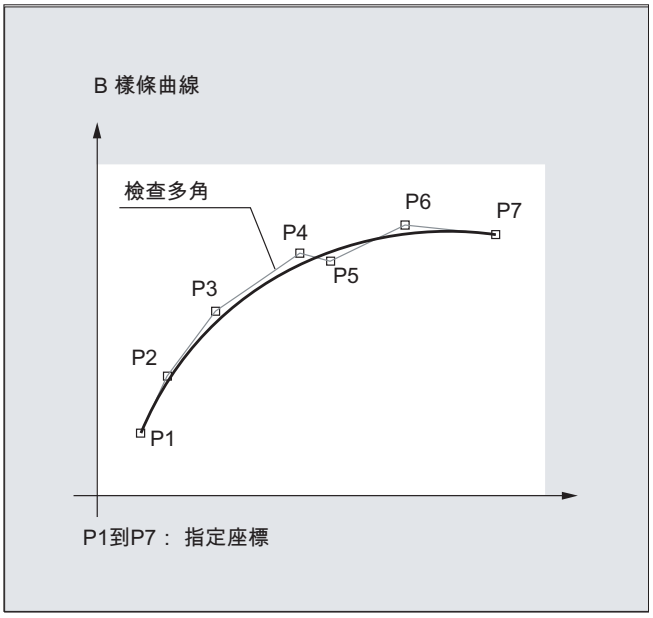
曲線插補的優點

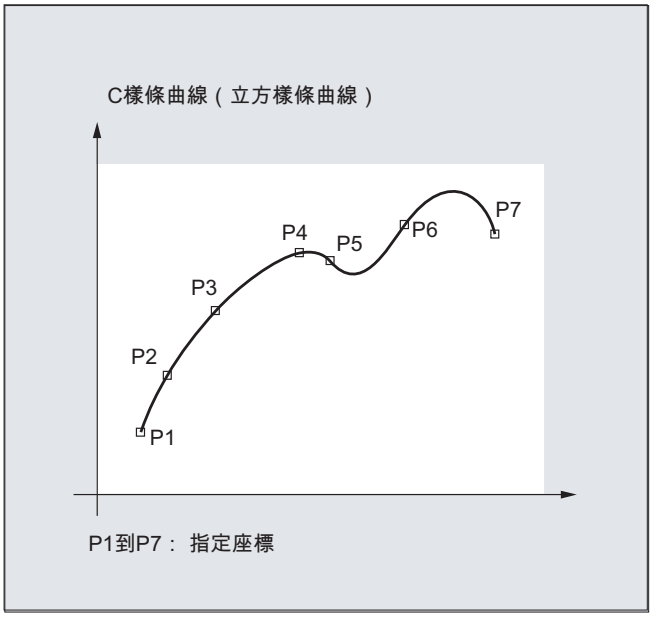
相較於使用直線單節 G01，使用曲線插補含下列優點：

- 減少需用來定義輪廓的工件程式單節數量
- 於工件程式單節間轉換時，能減少機械系統壓力的平緩、曲線特性。

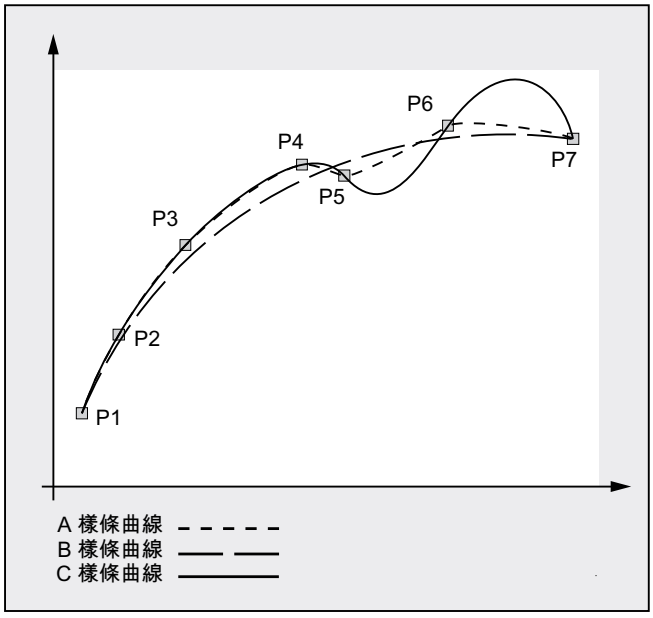
多種曲線的特性及使用

曲線類型	特性及使用
A 曲線	 **特性：** ● 精確地沿曲線穿越指定的中間點。 ● 曲線特性為切線，但無連續曲率。 ● 幾乎無產生任何非預期振盪。 ● 沿曲線中間點的變更僅影響本機。表示沿曲線中間點的變更，最多僅影響 6 個相鄰的中間點。 **應用：** A 曲線特別適合插補含大變化梯度的曲線（例如梯形曲線及特性）。

曲線類型	特性及使用
B 曲線	 **特性：** ● 請勿通過指定的沿曲線中間點，僅可靠近。不吸引曲線的中間點。透過使用係數增加中間點的重量，可另外影響曲線特性。 ● 曲線特性為含連續曲率的切線。 ● 不會產生任何非預期振盪。 ● 沿曲線中間點的變更僅影響本機。表示沿曲線中間點的變更，最多僅影響 6 個相鄰的中間點。 **應用：** 其 B 曲線主要作為與 CAD 系統間的介面。

曲線類型	特性及使用
C 曲線	 **特性：** ● 精確地沿曲線穿越指定的中間點。 ● 曲線特性為含連續曲率的切線。 ● 特別在梯度變化的位置，會經常產生非預期振盪。 ● 中間點的變更影響為全域。表示若變更中間點，則將影響全部的曲線特性。 **應用：** 若中間點位於曲線上，且可透過分析定義（圓弧、拋物線、雙曲線），則可善用 C 曲線。

以相同的插補點比較三種曲線類型



曲線單節的最少數量

G 碼 ASPLINE、BSPLINE 及 CSPLINE 使用曲線連接單節結尾。為此，需同時計算一系列的單節（終點）。以 10 個單節為計算的緩衝大小標準。並非每個單節資訊皆為曲線終點。然而，每 10 個單節，控制便需某些數量的曲線終點單節：

曲線類型	曲線單節的最少數量
A 曲線：	每 10 個單節中，至少需有 **4** 個單節為曲線單節。 不包含註解單節或參數計算。
B 曲線：	每 10 個單節中，至少需有 **6** 個單節為曲線單節。 不包含註解單節或參數計算。
C 曲線：	所需最小曲線單節數量為下列總和的結果： MD20160 $MC_CUBIC_SPLINE_BLOCKS + 1 的值 在 MD20160 輸入計算曲線區段的點數量。預設設定為 8%。每 10 個單節中，至少需有 **9** 個單節為曲線單節。

說明

若容許值為下衝，且當涉入於曲線中的其中一座標軸，經程式設計為定位軸時，將輸出警報。

結合短曲線單節

曲線插補可產生短曲線單節，無需減少路徑速率。“結合短曲線單節”函數，可讓您結合單節，因此單節長度足夠，且不會減低路徑速率。

透過通道專屬的機械資料啟用函數：

MD20488 $MC_SPLINE_MODE（曲線插補設定）。

參考資料：

功能手冊，基本功能；連續路徑模式，精確停止，預見控制（B1），
一章：結合短曲線單節

5.3 曲線群組 (SPLINEPATH)

使用 SPLINEPATH 指令選擇曲線群組中欲插補之軸。 曲線插補群組中最多可包含八個路徑軸。

說明

若未明確程式設計 SPLINEPATH，則以曲線群組般移動通道的前三個軸。

語法

在獨立單節中定義曲線群組：

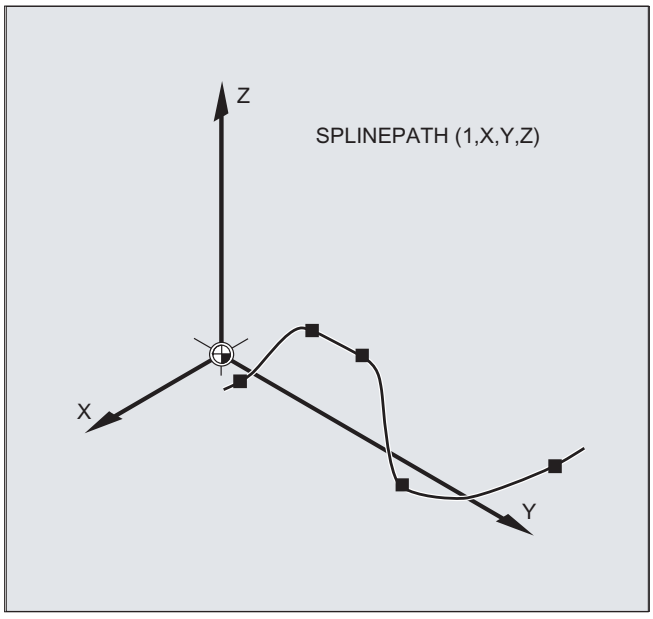
```
SPLINEPATH（n，X，Y，Z，...）
```

含義

SPLINEPATH：	定義樣條群組的指令
n：	=1（固定值）
X,Y,Z,…：	曲線群組中欲插補路徑軸的識別碼

範例： 含三個路徑軸的曲線群組

程式碼	註解
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH（1，X，Y，Z）	；曲線群組
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	；C 曲線
N14 X30 Y40 Z50 A60 B70	；中間點
…	
N100 G1 X... Y...	；取消選擇曲線插補


Z
SPLINEPATH (1,X,Y,Z)
X
Y

5.4 啟用 / 停用 NC 單節壓縮（COMPON、COMPCURV、COMPCAD、COMPSURF、COMPOF）

單節逼近的線性單節功能（並且依據參數化，也有圓弧及 / 或快速移動單節）可以使用 G 群組 30 的 G 指令啟用 / 停用。指令為模態。

語法

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

意義

COMPON：	啟用壓縮程序函數 COMPON
COMPCURV：	啟用壓縮程序函數 COMPCURV
COMPCAD：	啟用壓縮程序函數 COMPCAD
COMPSURF：	啟用壓縮程序函數 COMPSURF
COMPOF ：	停用目前生效的壓縮程序函數

說明

倒圓角函數 G642 及震動停止 SOFT 可進一步改善表面品質。需於程式開頭寫入指令。

範例：COMPCAD

程式碼	註解
N10 G00 X30 Y6 Z40	
N20 G1 F10000 G642	；啟用：倒圓角函數 G642
N30 SOFT	；啟用：震動限制 SOFT
N40 **COMPCAD**	；啟用：壓縮程序函數 COMPCAD
N50 STOPFIFO	
N24050 Z32.499	；第 1 個移動單節
N24051 X41.365 Z32.500	；第 2 個移動單節
...	
N99999 X...Z...	；上一個移動單節
COMPOF	；壓縮程序函數關閉。
...	

5.5 多項式插補（POLY、POLYPATH、PO、PL）

實際上是多項式插補 (POLY) 而非曲線插補的類型。 其主要目的，是要當作由外部程式設計已產生的曲線所用之介面，且在該處可直接程式設計曲線區段。

此插補模式減少 NC 計算多項式係數的作業。 在係數是直接由 CAD 系統或後處理器提供的情況下，可以達到最佳化運用。

語法

3 次多項式：

```
POLY PO[X]=（xe，a2，a3） PO[Y]=（ye，b2，b3） PO[Z]=（ze，c2，c3）
PL=n
```

五次多項式及新多項式語法：

```
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"）
```

說明

在一個 NC 單節中，程式設定的多項式係數和軸的總數，不可以超過每一單節的最大允許軸數。

含義

POLY：	使用包含 POLY 的單節啟用多項式插補。
POLYPATH：	可為 AXIS 或 VECT 軸群組選擇多項式插補
PO[軸識別碼 / 變數]：	終點及多項式係數
X、Y、Z：	軸識別碼
xe、ye、ze：	特定軸的結束位置規格；值域與路徑尺寸相同

a2、a3、a4、a5：	係數 a_2、a_3、a_4、及 a_5 以其值寫入；值域與路徑尺寸相同。 若係數等於零，則可忽略各情況下的最後係數。
PL：	定義多項式的參數間隔長度（函數 f（p）定義範圍）。 間隔永遠從 0 開始，p 可假設值為 0 到 PL。 PL 的理論值域： 0.0001 ... 99 999.9999 **附註：** PL 值將套用至其所在之單節。 若未程式設計 PL，則套用 PL=1。

啟用 / 停用多項式插補

多項式插補係在工件程式中以 POLX G 指令啟用。

POLY G 指令連同 G0、G1、G2、G3、ASPLINE、BSPLINE 和 CSPLINE 都屬於第一個群組。

軸僅以名稱和端點程式設計（例如 X10），並以線性移動。若一 NC 單節中全部的軸都以此種方式程式設計，則控制方式與 G1 相同。

藉由程式設計第一個群組 G0、G1 的其他指令，可再次將多項式插補間接停用。

多項式係數

PO 值（PO[]=）或 ...=PO（...）為一個軸指定所有多項式係數。 以相應多項式次方的逗號指定、分隔數個值。 一個單節中的各軸可有不同的多項式次方。

POLYPATH 子程式

使用 POLYPATH(...)，即可選擇性地釋放特定軸群組的多項式插補：

僅路徑軸和輔助軸：	POLYPATH（"AXES"）
僅定向軸： （利用方向轉換作移動時）	POLYPATH（"VECT"）

未釋放的軸以線性移動。

兩種軸群組的多項式插補預設為啟用。

程式設計不需 POLYPATH( ) 參數，即可將所有軸的多項式插補停用。

範例

程式碼	註解
N10 G1 X... Y... Z... F600	
N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2) PL=1.5	；開啟多項式插補
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	；軸的混合資料
N27 PO[X]=(10,2.5) PO[Y]=(2.3)	；未程式設計 PL，套用 PL=1
N30 G1 X... Y... Z.	；關閉多項式插補
…	

範例： 新增的多項式語法

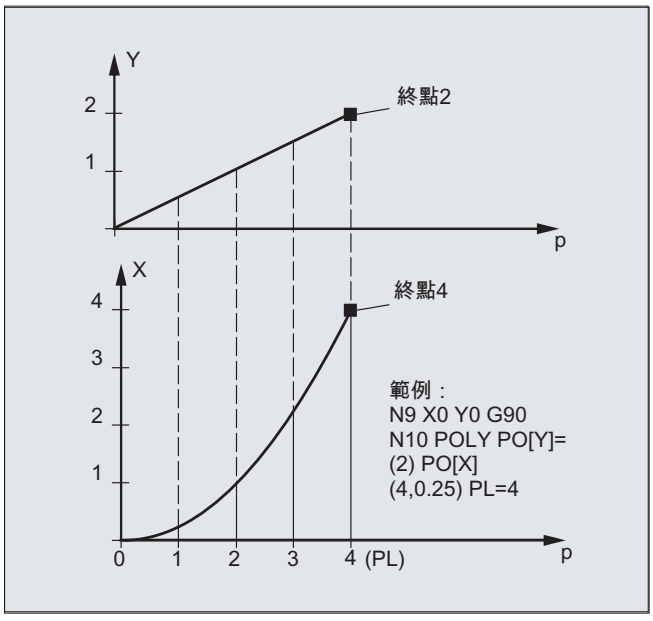
仍然有效的多項式語法	新增的多項式語法
PO[axis identifier]=(.. ,..)	Axis identifier=PO(.. ,..)
PO[PHI]=(.. ,..)	PHI=PO(.. ,..)
PO[PSI]=(.. ,..)	PSI=PO(.. ,..)
PO[THT]=(.. ,..)	THT=PO(.. ,..)
PO[]=(.. ,..)	PO(.. ,..)
PO[variable]=IC(.. ,..)	variable=PO IC(.. ,..)

範例： X/Y 平面上曲線。

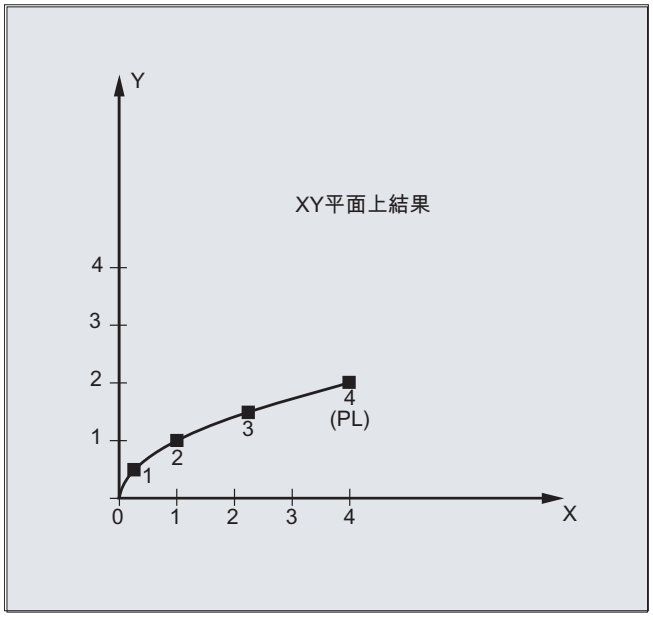
程式設計

程式碼
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

曲線 X(p) 和 Y(p) 的形狀


Y
2
1
終點2
p
X
4
3
2
1
終點4
範例：
N9 X0 Y0 G90
N10 POLY PO[Y]=
(2) PO[X]
(4,0.25) PL=4
0
1
2
3
4 (PL)
p

XY 平面中曲線的形狀


Y
XY平面上結果
4
3
2
1
4
(PL)
3
2
1
0
1
2
3
4
X

說明

表示多項式函數的方程式通常如下：

$$f(p)= a_0 + a_1p + a_2p^2 + \ldots + a_np^n$$

其　a_i：常數係數 (i = 0, 1, ..., n)
中：　p：參數

控制器系統中，可程式設計的最大多項式次數為 5：

$$f（p）= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

指派具體值給這些係數，即可產生各種不同的曲線形狀，例如直線、拋物線和冪函數。

$a_2 = a_3 = a_4 = a_5 = 0$ 時，可產生直線：

$$f（p） = a_0 + a_1p$$

以下仍適用：

$a0$：前一單節結尾的軸位置

$p = PL$

$a_1 = （x_E - a_0 - a_2*p^2 - a_3*p^3）/p$

可在**未**使用 G 指令 POLY 啟用多項式插補的情況下，程式設計多項式。此情況下，程式設計的多項式並未作插補，所有程式設計的軸端點都以線性逼近 (G1)。必須在工件程式 (POLY) 中，明確地將多項式插補啟用後，已編寫的多項式才會以這種方式移動。

特殊功能：分母多項式

可使用 PO[]=(...) 指令為幾何軸（無需指定軸名稱）編寫一個普通的分母多項式，意即將幾何軸的動作插補為兩個多項式的商數。

使用此程式設計選項，可精確呈現如圓錐曲線（圓、橢圓、拋物線、雙曲線）的形狀。

範例：

程式碼	註解
POLY G90 X10 Y0 F100	；幾何軸線性移動至位置 X10 Y0。
PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)	；幾何軸沿象限移動至 X0 Y10。

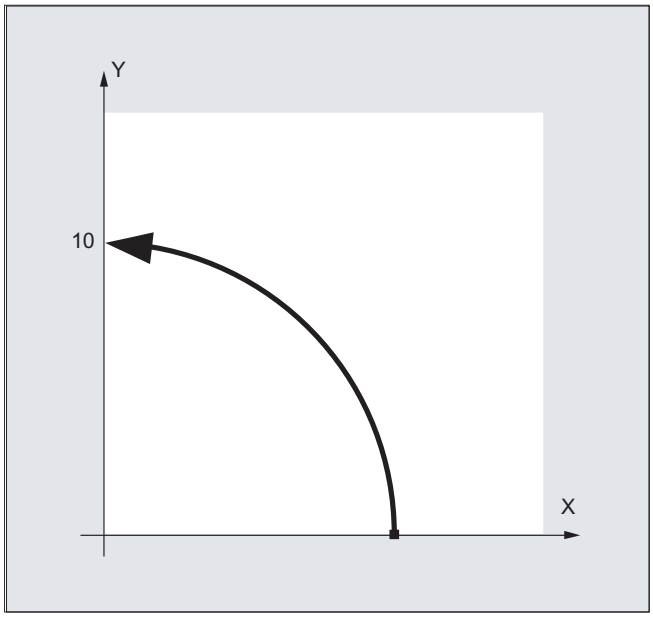
分母多項式的常數係數 (a_0) 一律假設為 1。程式設計的終點與 G90 / G91 無關。

X(p) 和 Y(p) 以程式設計的值計算如下：

$X(p) = (10 - 10 * p^2) / (1 + p^2)$
$Y(p) = 20 * p / (1 + p^2)$
且 $0 \le p \le 1$

以程式設計的起點、終點，係數 a_2 和 PL=1 計算，得到下列中間值：

分子（X）= $10 + 0 * p - 10 * p^2$
分子 (Y) = $0 + 20 * p + 0 * p^2$
分母 = $1 + p^2$



若多項式插補有效，且已在間隔 [0，PL] 內使用零點程式設計分母多項式，則拒絕並輸出警報。分母多項式不會影響特殊軸的動作。

說明

可使用 G41、G42 啟動刀具半徑補正，搭配多項式插補，並可與在線性或圓弧插補模式以相同的方式套用。

5.6 可設定路徑參考（SPATH、UPATH）

對多項式插補而言（POLY、ASPLINE、BSPLINE、CSPLINE、COMPON、COMPCURV），軸路徑 i 的位置指定為多項式 $p_i(U)$。 曲線參數 U 在 NC 單節之中由 0 移至 1。

FGROUP 選擇要套用路徑進給 F 的軸（FGROUP 軸）。 在 FGROUP 軸路徑 S 的恆速插補，代表在多項式插補期間，通常會產生曲線參數 U 的非恆定變更。因此若選擇不在 FGROUP 之中的軸，其跟隨 FGROUP 軸的方式有兩種可能：

- 同步至路徑 S (SPATH)
- 同步至曲線參數 U (UPATH)

語法

```
SPATH
UPATH
```

含義

SPATH：	不在 FGROUP 之中的軸，會以參考路徑 S 的方式移動
UPATH：	不在 FGROUP 之中的軸，會以參考曲線參數 U 的方式移動

說明

UPATH 及 SPATH 也使用路徑動作定義 F 字組多項式（FPOLY、FCUB、FLIN）的相互關係。

補充條件

SPATH 及 UPATH 對以下項目沒有意義：

- 線性插補（G1）
- 電路插補（G2、G3）
- 螺紋單節（G33、G34、G35、G33x、G63）
- 所有路徑軸均在 FGROUP

範例

下列範例顯示不同類型動作控制間的差異。

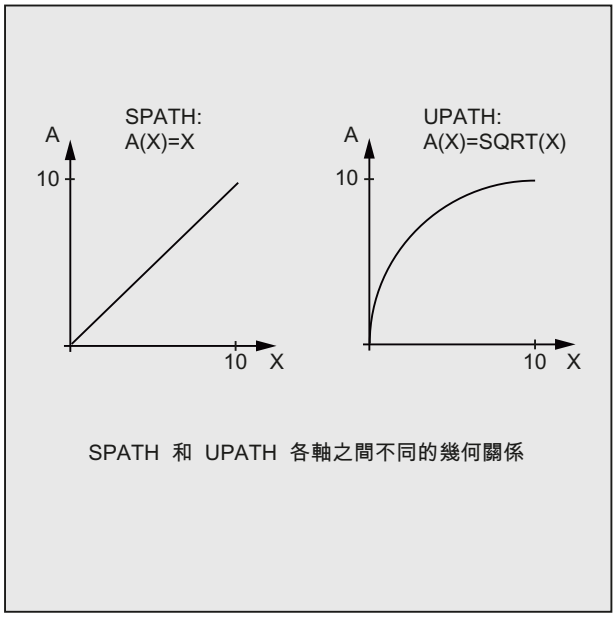
程式碼

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 SPATH                ; SPATH
N20 POLY PO[X]=(10,10) A10
```

程式碼

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH                ; UPATH
N20 POLY PO[X]=(10,10) A10
```

在兩個程式區段中，N20 之中的 FGROUP 軸路徑 S，需視曲線參數 U 的平方而定。因此，沿著路徑 X 的同步軸 A 形成的不同位置，根據 SPATH 或 UPATH 而有效。



SPATH 和 UPATH 各軸之間不同的幾何關係

其他資訊

重置及機械資料 / 選項資料控制行為

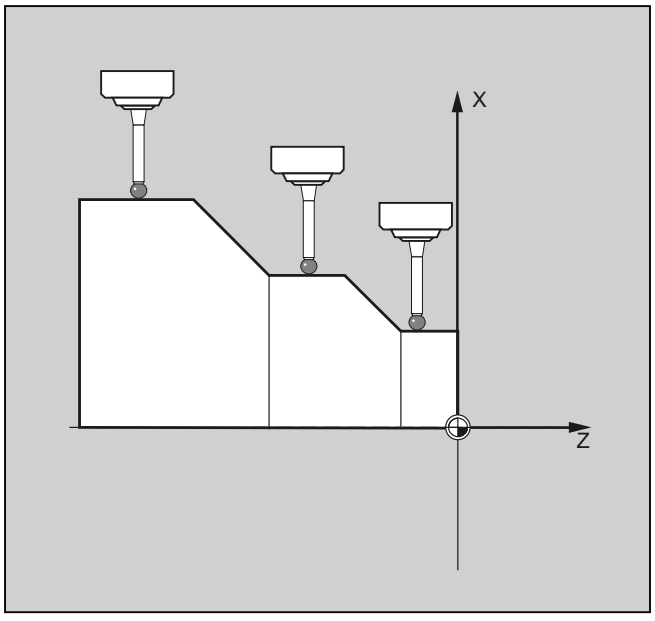
以 MD20150 $MC_GCODE_RESET_VALUES[44] 定義的 G 指令，將於重置後生效（第 45 G 群組）。

平滑化類型的初始狀態係以 MD20150 $MC_GCODE_RESET_VALUES[9]（第 10 個 G 群組）定義。

軸專屬機械資料 MD33100 $MA_COMPRESS_POS_TOL[<n>] 具備其他重要意義：為壓縮功能和利用 G642 進行平滑化的允差值設定。

5.7 以觸發式探針量測 (MEAS, MEAW)

「使用觸發式探針量測」可用於在逼近工件上的實際位置。 在探針的切換稜邊，會測量所有程式設計在量測單節中的軸之位置並寫入至各個軸適當的記憶體中。



以下兩個固定位址可用於編寫此功能：

- MEAS
  MEAS 可刪除實際與設定點位置間的剩餘距離。
- MEAW
  MEAW 可用在編寫的位置永遠需要逼近的量測作業中。

MEAS 與 MEAW 為非模態；其會與動作操作一同程式設計。 針對個別量測作業改寫進給率、插補類型（G0、G1，等等）以及軸數量。

語法

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

含義

MEAS：	指令： 含刪除剩餘距離的量測	
	有效：	非模態

MEAW：	指令： 量測時**不刪除**剩餘距離	
	有效：	非模態
<TE>：	觸發開始量測的事件	
	類型：	INT
	值域：	-2, -1, 1, 2
	含義：	
	(+)1	探針 1 的上升稜邊（量測輸入 1）
	-1	探針 1 的下降稜邊（量測輸入 1）
	(+)2	探針 2 的上升稜邊（量測輸入 2）
	-2	探針 2 的下降稜邊（量測輸入 2）
	附註： 最多可有 2 個探針（視設定層級而定）。	
G...：	插補類型，例如 G0、G1、G2 或 G3	
X... Y... Z...：	直角座標的終點	

範例

程式碼	註解
N10 MEAS=1 G1 F1000 X100 Y730 Z40 ...	；使用探針於第一個量測輸入及線性插補量測單節。 自動產生前置處理停止。

其他資訊

量測作業狀態

若需要在程式中評估是否已觸發探針，可查詢狀態變數 $AC_MEA[<n>]（<n> = 量測探針數量）：

值	含義
0	量測作業未完成。
1	量測作業已完成（已觸發探針）。

說明

若在程式中偏移程式，則變數設為 1。當啟動量測單節時，變數自動設為探針的初始狀態。

讀取量測值

取得單節中所有移動路徑及定位軸之位置（最大軸數量視控制設定而定）。在使用 MEAS 時，該動作會以定義的方式在觸發探針後減速。

說明

若在量測單節中編寫幾何軸，則會儲存該量測值，用於目前所有的幾何軸。
若在量測單節中程式設計轉換中的軸，則會記錄所有轉換中的軸量測值。

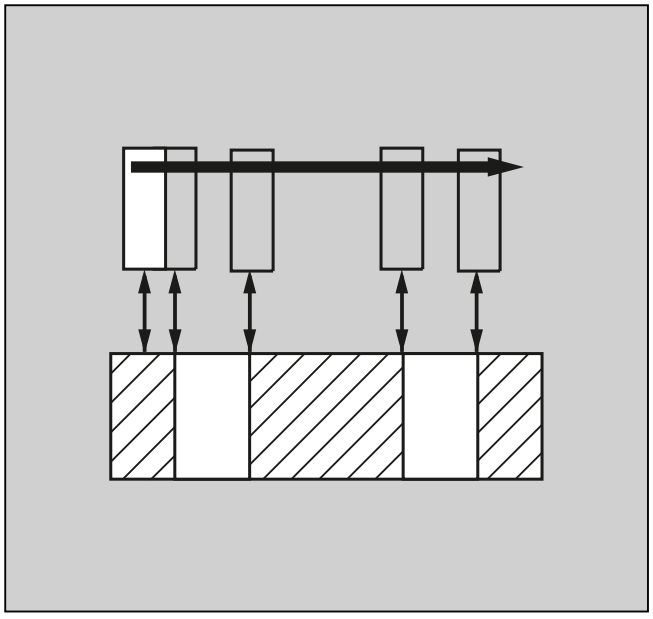
讀取量測結果

經由下列系統變數可讀取探針的軸量測值：

- $AA_MM[<軸>]
  使用機械座標系統的量測結果
- $AA_MW[<軸>]
  使用工件座標系統的量測結果

5.8 軸專屬測量（MEASA、MEAWA、MEAC）（選項）

可使用數個探針及測量系統用於軸專屬測量。



關鍵字 MEASA、MEAWE 和 MEAC 可用於編寫此功能。

以 MEASA 或 MEAWA 進行軸的程式設計，每次量測最多可取得四個量測值，並根據觸發事件儲存於系統變數中。

可使用 MEAC 執行量測操作。在此情況下，量測結果會儲存於 FIFO 變數中。

語法

```
MEASA[<軸>]=(<模式>,<TE1>,...,<TE4>)
MEAWA[<軸>]=(<模式>,<TE1>,...,<TE4>)
MEAC[<軸>]=(<模式>,<量測記憶體>,<TE1>,...,<TE4>)
```

說明

MEASA 與 MEAWA 為非模態，其可一同程式設計在一個單節中。然而，若於同一單節中與 MEAS/MEAW 一同程式設計 MEASA/MEAWA，將輸出錯誤訊息。

含義

MEASA：	關鍵字：含刪除剩餘距離的軸專屬測量	
	有效：	非模態

MEAWA：	關鍵字：不含刪除剩餘距離的軸專屬測量	
	有效：	非模態
MEAC：	關鍵字：不含刪除剩餘距離的軸專屬持續測量	
	有效：	非模態
<軸>：	用於量測的通道軸名稱	
<模式>：	二位數字顯示操作模式（量測模式與量測系統）	
	個位數字（量測模式）：	
	0	取消量測作業。
	1	最多可同時啟動 4 個不同的觸發事件。
	2	最多可連續啟動 4 個不同的觸發事件。
	3	最多可連續啟動 4 個不同的觸發事件，但開始時不監控觸發事件 1（警報 21700/21703 停用）。 **附註：** 此模式 MEAC 並不支援。
	十位數字（量測系統）：	
	0（或無資料）	主動量測系統
	1	量測系統 1
	2	量測系統 2
	3	兩個量測系統
<TE>：	觸發開始量測的事件	
	類型：	INT
	值域：	-2，-1，1，2
	含義：	
	(+)1	探針 1 的上升稜邊
	-1	探針 1 的下降稜邊
	(+)2	探針 2 的上升稜邊
	-2	探針 2 的下降稜邊
<量測記憶體>：	FIFO 數量（循環儲存）	

範例

範例 1：在模式 1 中刪除剩餘距離的軸專屬測量（以時間順序評估）

a) 使用一個量測系統

程式碼	註解
...	
N100 MEASA[X]=(1,1,-1) G01 X100 F100	；於模式 1 使用主動量測系統量測。等候使用上升 / 下降稜邊的量測訊號，以從探針 1 移動路徑至 X=100。
N110 IF $AC_MEA[1]==FALSE GOTOF END	；檢查量測是否成功。
N120 R10=$AA_MM1[X]	；儲存於第一個已程式設計的觸發事件（上升稜邊）取得的量測值。
N130 R11=$AA_MM2[X]	；儲存於第二個已程式設計的觸發事件（下降稜邊）取得的量測值。
N140 END：	

b) 使用兩個量測系統

程式碼	註解
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	；於模式 1 使用兩個量測系統量測。等候使用上升 / 下降稜邊的量測訊號，以從探針 1 移動路徑至 X=100。
N210 IF $AC_MEA[1]==FALSE GOTOF END	；檢查量測是否成功。
N220 R10=$AA_MM1[X]	；儲存上升稜邊上量測系統 1 的量測值。
N230 R11=$AA_MM2[X]	；儲存上升稜邊上量測系統 2 的量測值。
N240 R12=$AA_MM3[X]	；儲存下降稜邊上量測系統 1 的量測值。
N250 R13=$AA_MM4[X]	；儲存下降稜邊上量測系統 2 的量測值。
N260 END：	

範例 2：在模式 2 中刪除剩餘距離的軸專屬測量（以程式設計的順序評估）

程式碼	註解
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	；於模式 2 使用主動量測系統量測。當移動路徑至 X=100 時，以上升稜邊探針 1、下降稜邊探針 1、上升稜邊探針 2、下降稜邊探針 2 的順序等候量測訊號。
N110 IF $AC_MEA[1]==FALSE GOTOF PROBE2	；檢查是否成功使用探針 1 量測。
N120 R10=$AA_MM1[X]	；儲存在第一個程式觸發事件（探針 1 的上升稜邊）取得的量測值。
N130 R11=$AA_MM2[X]	；儲存在第二個程式觸發事件（探針 1 的上升稜邊）取得的量測值。
N140, PROBE2：	
N150 IF $AC_MEA[2]==FALSE GOTOF END	；檢查是否成功使用探針 2 量測。

程式碼	註解
N160 R12=$AA_MM3[X]	；儲存在第三個程式觸發事件（探針 2 的上升稜邊）取得的量測值。
N170 R13=$AA_MM4[X]	；儲存在第四個程式觸發事件（探針 2 的上升稜邊）取得的量測值。
N180 END：	

範例 3：使用模式 1 的軸專屬連續測量（以時間順序評估）

a) 最多 100 個量測值的量測

程式碼	註解
...	
N110 DEF REAL MEASVALUE[100]	
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	；在模式中使用主動量測系統量測，儲存量測值；於 $AC_FIFO1 下，等候下降稜邊，從探針 1 移動路徑至 X=1000 的量測系統。
N135 STOPRE	
N140 MEAC[X]=（0）	；達到軸位置時終止量測。
N150 R1=$AC_FIFO1[4]	；於參數 R1 中儲存累計的量測值。
N160 FOR loop=0 TO R1-1	
N170 MEASURED VALUE[loop]=$AC_FIFO1[0]	；從 $AC_FIFO1 讀出並儲存量測值。
N180 ENDFOR	

b) 在 10 個量測值後，進行刪除剩餘距離的量測

程式碼	註解
...	
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	；刪除剩餘距離。
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC [X]=(0)	
N40 R1 = $AC_FIFO1[4]	；量測值數量。
...	

c) 使用兩個探針量測下降 / 上升的齒腹

程式碼	註解
...	
N110 DEF REAL MEASVALUE[16]	
N120 DEF INT loop=0	

程式碼	註解
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	；在模式 1 中使用主動量測系統量測，將量測值儲存於 $AC_FIFO1，當移動路徑至 X=100 時，等待的量測訊號依序為探針 1 的下降稜邊和探針 2 的上升稜邊。
N140 STOPRE	；前置處理停止
N150 MEAC[X]=(0)	；達到軸位置時終止量測。
N160 R1=$AC_FIFO1[4]	；於參數 R1 中儲存累計的量測值。
N170 FOR loop=0 TO R1-1	
N180 MEASURED VALUE[loop]=$AC_FIFO1[0]	；從 $AC_FIFO1 讀出並儲存量測值。
N190 ENDFOR	

其他資訊

量測工作

可在工件程式中或從同步動作中編寫量測作業（請參閱「同步動作 (頁 661)」章節）。.請注意在任何時間下各軸每次僅啟用一個量測工作。

說明

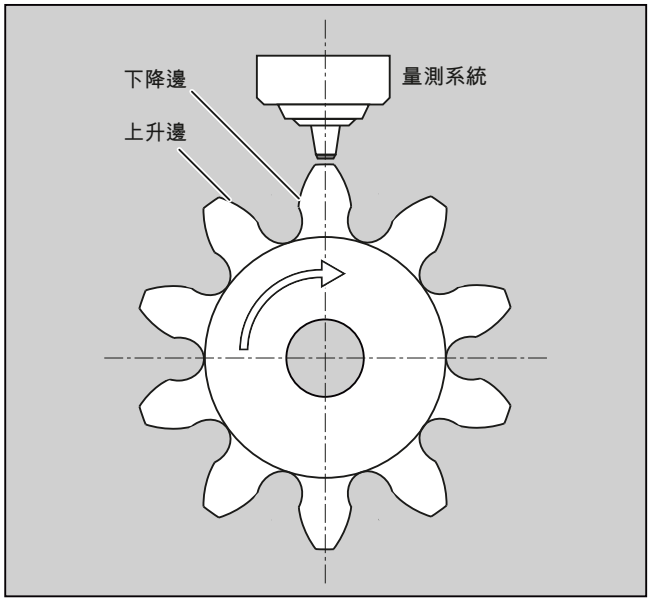
需調整進給以符合目前的量測作業。

以 MEASA 與 MEAWA 為例，在每個位置控制循環中，沒有發生超過 1 個相同類型的觸發事件，且沒有超過 4 種不同類型的觸發事件時，才能保證進給率結果的正確性。

在使用 MEAC 連續量測的情況下，插補時脈及位置控制循環間的比例不可超過 1：8。

觸發事件

觸發事件包含量測訊號的探針及觸發條件（上升或下降稜邊）的數量。



每次量測最多可處理定址探針中 4 個觸發事件；換言之，即 2 個探針最多各含 2 個量測訊號稜邊。處理順序及觸發事件最大數量視選擇模式而定。

說明

以下資料套用至量測模式 1：在一個量測作業中，相同的觸發事件僅能程式設計一次。

針對 MEAC，每個觸發事件的測量值數量可以利用 PROFIBUS 電報 395 增加至總共 8 個測量值的上升稜邊，以及每個觸發事件及定位控制器循環 8 個測值量的下降稜邊。

- 單探針：8 個上升稜邊測量值及 8 個下降稜邊測量值
- 雙探針：每個探針有 4 個上升稜邊測量值及 4 個下降稜邊測量值

也就是說可以使用 PROFIBUS 電報 395 達到更高的進給率或更高的速度。

參考資料：

功能手冊，延伸功能，量測 (M5)，章節：軸量測

操作模式

操作模式的第一位（十進位）可選擇需要的量測系統。若僅安裝一個量測系統，但已程式設計第二個，則會自動選擇安裝的系統。

第二位（十位數）可選擇需要的量測模式。量測程序可藉此針對相關控制系統所支援的選項做調整：

- **模式 1**
  依發生的時間順序評估觸發事件。選擇此模式時，僅可為六軸模組程式設計一個觸發事件。若已指定一個以上的觸發事件，則會自動切換模式選擇至模式 2（無訊息）。
- **模式 2**
  依已程式設計的順序評估觸發事件。
- **模式 3**
  依已程式設計的順序評估觸發事件，但不包括 START 監控觸發事件 1。

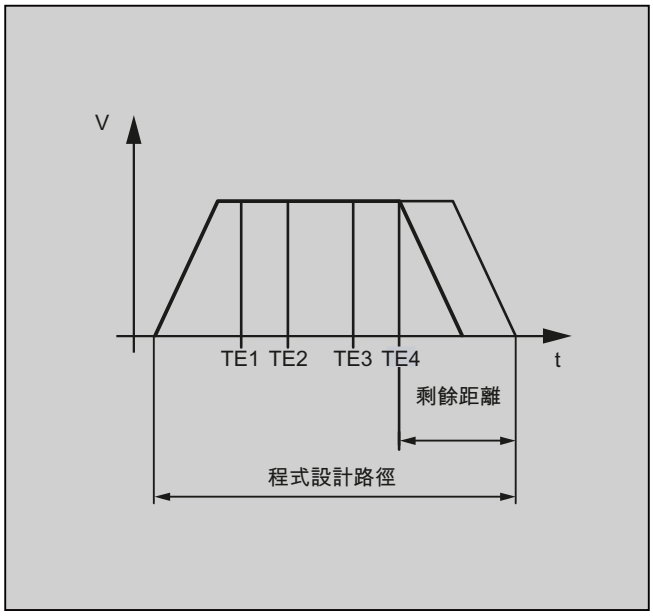
  說明

  若正在使用量測系統，則無法程式設計兩個以上的觸發事件。

量測時含及不含已刪除的剩餘距離

程式設計 MEASA 指令時，在記錄所有所需量測的值之前，不會刪除剩餘距離。

在需逼近特殊量測作業的已程式設計位置時，可執行 MEAWA 函數。



說明

同步動作中無法程式設計 MEASA。此外，MEAWA 加上刪除剩餘距離可編寫為同步動作。
若已從同步動作中啟動含 MEAWA 的量測工作，則量測值僅在機械座標系統中可用。

MEASA、MEAWA 的量測結果

量測結果於下列系統變數中可用：

- 在機械座標系統中：

$AA_MM1[<軸>]	觸發事件 1 上已程式設計的量測系統量測值
...	...
$AA_MM4[<軸>]	觸發事件 4 上已程式設計的量測系統量測值

- 在工件座標系統中：

$AA_MW1[<軸>]	觸發事件 1 上已程式設計的量測系統量測值
...	...
$AA_MW4[<軸>]	觸發事件 4 上已程式設計的量測系統量測值

幾何軸/轉換

若欲為幾何軸啟動軸量測，需明確地為所有剩餘幾何軸程式設計相同的量測工作。同樣套用至涉入轉換的軸。

範例：

```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

或

```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

使用兩個量測系統的量測工作

若由兩個量測系統執行量測工作，則會取得兩個相關軸量測系統中，兩個可能觸發事件之一。因此預設保留變數的指派為：

$AA_MM1[<軸>]	或	$AA_MW1[<軸>]	觸發事件 1 量測系統 1 之量測值
$AA_MM2[<軸>]	或	$AA_MW2[<軸>]	觸發事件 1 量測系統 2 之量測值
$AA_MM3[<軸>]	或	$AA_MW3[<軸>]	觸發事件 2 量測系統 1 之量測值
$AA_MM4[<軸>]	或	$AA_MW4[<軸>]	觸發事件 2 量測系統 2 之量測值

系統變數

探針狀態可存在下列系統變數中：

$A_PROBE[<n>]

值	含義
1	探針偏移
0	探針未偏移

探針限制存在下列系統變數中：

$A_PROBE_LIMITED[<n>]

值	含義
1	探針限制啟用
0	探針限制停用

<n> = 探針

參考資料：

清單手冊，系統變數

MEASA、MEAWA 量測工作狀態

若程式中需要評估，可透過 $AC_MEA[<n>] 查詢量測作業狀態，其中 <n> = 探針編號。一旦單節程式中所有探針 <n> 的觸發事件皆發生，此變數會傳回「1」值。否則，數值為 0。

說明

若從同步動作開始量測，則不會再更新$AC_MEA。在此情況下，需查詢新的 PLC 介面信號 DB31、 ... DBX62.3，或等效變數 $AA_MEAACT[<軸>]。

含義：

$AA_MEAACT==1：量測啟用

$AA_MEAACT==0：量測未啟用

連續量測（MEAC）

MEAC 量測值在機械座標系統中可用，並儲存在已程式設計的 FIFO[m]記憶體（循環緩衝器）中。若設定兩個用於量測的探針，則第二個探針的量測值，會分別儲存於特別為此而設的 FIFO[n+1]記憶體中（在機械參數中定義）。

FIFO 記憶體為循環緩衝器，根據循環原則將量測值寫入 $AC_FIFO 變數中，請參閱 "同步動作 (頁 661)" 章節。

說明

僅可從循環儲存中讀取 FIFO 內容一次。若將重複使用該量測資料，則需在使用者資料中進行緩衝。

若 FIFO 記憶體量測值的數量，超過機械參數中定義的最大值，則自動終止量測。

可循環讀出量測值以進行無限量測流程。在此狀況下，讀出資料的頻率需與讀入新量測值的頻率相同。

參考資料：

- 功能手冊，同步動作；詳細說明，
  章節：參數 ($AC_FIFO)
- 功能手冊，延伸功能，量測 (M5)，章節：軸量測

防止程式設計錯誤

適當偵測並指出下列程式設計錯誤：

- MEASA/MEAWA 與 MEAS/MEAW 程式設計在相同單節中
  範例：
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA 的參數數目 <2 或 >5
  範例：
  ```
  N01 MEAWA[X]=（1） G01 F100 POS[X]=100
  ```
- MEASA/MEAWA 觸發事件不等於 1/ -1/ 2/ -2。
  範例：
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```
- 具有無效模式的 MEASA/MEAWA
  範例：
  ```
  N01 MEAWA[B]=(4,1) B100
  ```
- 具有程式設計了兩次觸發事件的 MEASA/MEAWA
  範例：
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```
- MEASA/MEAWA 和遺失的幾何軸
  範例：
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100；幾何軸 XY/Z
  ```
- 量測作業與幾何軸不一致
  範例：
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

5.9 OEM 使用者的特殊函數 (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

OEM 位址

由 OEM 使用者決定 OEM 位址的意義。透過編譯循環結合其函數性。保留了 5 個 OEM 位址 (OMA1 ... OMA5)。可設定位址識別碼。可於任何單節中程式設計 OEM 位址。

保留的 G 指令呼叫

OEM 使用者保留了以下的 G 指令呼叫：

- OEMIPO1, OEMIPO2（從 G 群組 1）
- G810 ... G819（G 群組 31）
- G820 ... G829（G 群組 32）

透過編譯循環結合其函數性。

函數及子程式

OEM 使用者亦可使用參數傳輸，設定預先定義的函數及子程式。

說明

工件模擬

SW 4.4 以下不支援編譯循環，自 SW 4.4 版本僅支援選定的編譯循環 (CC) 用於工件模擬。

不支援在編譯循環工件程式中的語言指令（OMA1 ... OMA5、OEMIPO1/2、G810 ... G829，自己的製程及功能）；因此將導致警報訊息、取消模擬且無個別處理。

解決方案：個別處理遺失的 CC - 工件程式中的特定語言元件 ($P_SIM query)。
範例：

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300；CC 模擬未啟用
ELSE
N5 X300 OMA1=10
ENDIF
```

5.10 含轉角減速的進給率降低功能 (FENDNORM, G62, G621)

使用自動轉角減速時，將在到達轉角前依據鐘形曲線減慢進給率。 也可透過設定資料，參數化與加工相關的刀具行為範圍。 分別為：

- 進給率減慢功能的開始與結束
- 進給率減慢時所遵照的手動倍率內容
- 偵測相關轉角

相關轉角的內角小於設定資料中參數化的轉角。

預設值 FENDNORM 會停用自動轉角手動倍率的函數。

參考資料：
/FBFA/「函數說明 ISO 語言代碼」

語法

```
FENDNORM
G62 G41
G621
```

含義

FENDNORM：	自動轉角減速 OFF
G62：	刀具半徑偏移量有效時，轉角在內角減速
G621：	刀具半徑偏移量有效時，轉角在所有轉角減速

G62 僅適用於

- 具有有效的刀具半徑偏移量 G41、G42 以及
- 有效的連續路徑模式 G64、G641 的內角

逼近轉角所使用的減慢進給率是利用以下公式計算得出：

F*（為了減慢進給率所作的手動倍率）*進給率調整

最大可能進給率的減慢幅度，取自轉角上刀具要進行轉向的準確位置，並且參考（刀具）中心點路徑。

類似於 G62，G621 適用於 FGROUP 之指定軸上的每一個轉角。

5.11 可程式設計的動作結尾條件 (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

類似路徑插補的單節變更條件（G601、G602 及 G603），也可為工件程式中的單軸插補，或同步動作中的指令 / PLC 軸，程式設計移動結尾條件。

結尾動作條件集會影響單軸移動的工件程式單節及科技循環單節，其完成的快慢。 透過 FC15/16/18 同樣套用於 PLC。

語法

```
FINEA[<軸>]
COARSEA[<軸>]
IPOENDA[<軸>]
IPOBRKA（<軸>[，<即時>]）
ADISPOSA[<軸>]=(<模式>,<視窗大小>)
```

含義

FINEA：	動作結束條件：「精確停止細值」	
	有效：	模態
COARSEA：	動作結束條件：「精確停止粗值」	
	有效：	模態
IPOENDA：	動作結束條件："插補停止"	
	有效：	模態
IPOBRKA：	單節變更條件： 煞車斜率	
	有效：	模態
ADISPOSA：	動作結束條件的允差視窗	
	有效：	模態
<軸>：	通道軸名稱（X、Y、....）	
<即時>：	單節變更的時間，參考煞車斜率為 %： • 100% = 煞車斜率的起始 • 0% = 煞車斜率的結束，與 IPOENDA 意義相同	
	類型：	REAL

<模式>：	允差視窗的原點復歸		
	值域：	0	允差視窗未啟用
		1	允差視窗與設定位置相關
		2	允差視窗與實際位置相關
	類型：	INT	
<視窗尺寸>：	允差視窗尺寸		
	類型：	REAL	

範例

範例 1： 動作結束條件： "插補停止"

程式碼

```
；定位軸 X 移動至 100，速率 200 m/min、加速度 90%，
；動作結束條件： 插補停止
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]

；同步動作：
；ALWAYS IF： 設定輸入 1
；接著將定位軸 X 移動至 50，速率 200 m/min、加速度 140%，
；動作結束條件： 插補停止
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]
```

範例 2： 單節變更條件： "煞車斜率"

程式碼	註解
	；預設設定有效
N40 POS[X]=100	；定位動作從 X 至位置 100。
	單節變更條件： 精確停止微調
N20 IPOBRKA（X，100）	；單節變更條件： "煞車斜率"，
	100% = 煞車斜率的起始
N30 POS[X]=200	；一旦 X 軸開始煞車，則變更單節
N40 POS[X]=250	；X 軸不再煞車停在位置 200，而是繼續移動至位置 250。
	一旦軸開始煞車，單節變更。
N50 POS[X]=0	；軸 X 煞車並返回位置 0。
	位置 0 發生單節變更及 "精確停止微調"
N60 X10 F100	；軸 X 以路徑軸移動至位置 10

其他資訊

動作結束條件的系統變數

使用系統變數 $AA_MOTEND 讀取有效的動作結束條件。

參考： /LIS2sl/ 清單手冊，第 2 冊

單節變更條件： "煞車斜率" (IPOBRKA)

啟動單節變更條件 "煞車斜率" 時，如果程式設計選用的即時單節變更數值，此動作會使得下一個定位動作生效，並且寫入與主要執行同步的設定資料。 若即時單節變更未指定數值，則設定資料的實際值生效。

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

如果接下來將軸動作結束條件（FINEA、COARSEA、IPOENDA ）程式設計在軸上，IPOBRKA 即停用對應的存取。

其他的單節變更條件： "允差視窗" (ADISPOSA)

使用 ADISPOSA 可以將單節結尾（實際位置或設定點位置）周圍的允差視窗定義為其他的單節變更條件。 然後，單節變更必須履行兩個條件：

- 單節變更條件： "煞車斜率"
- 單節變更條件： "允差視窗"

參考

若需更多有關定位軸的單節變更條件之資訊，請參閱：

- 功能手冊，延伸功能；定位軸（P2）
- 程式設計手冊，基礎；"進給率控制"一章。

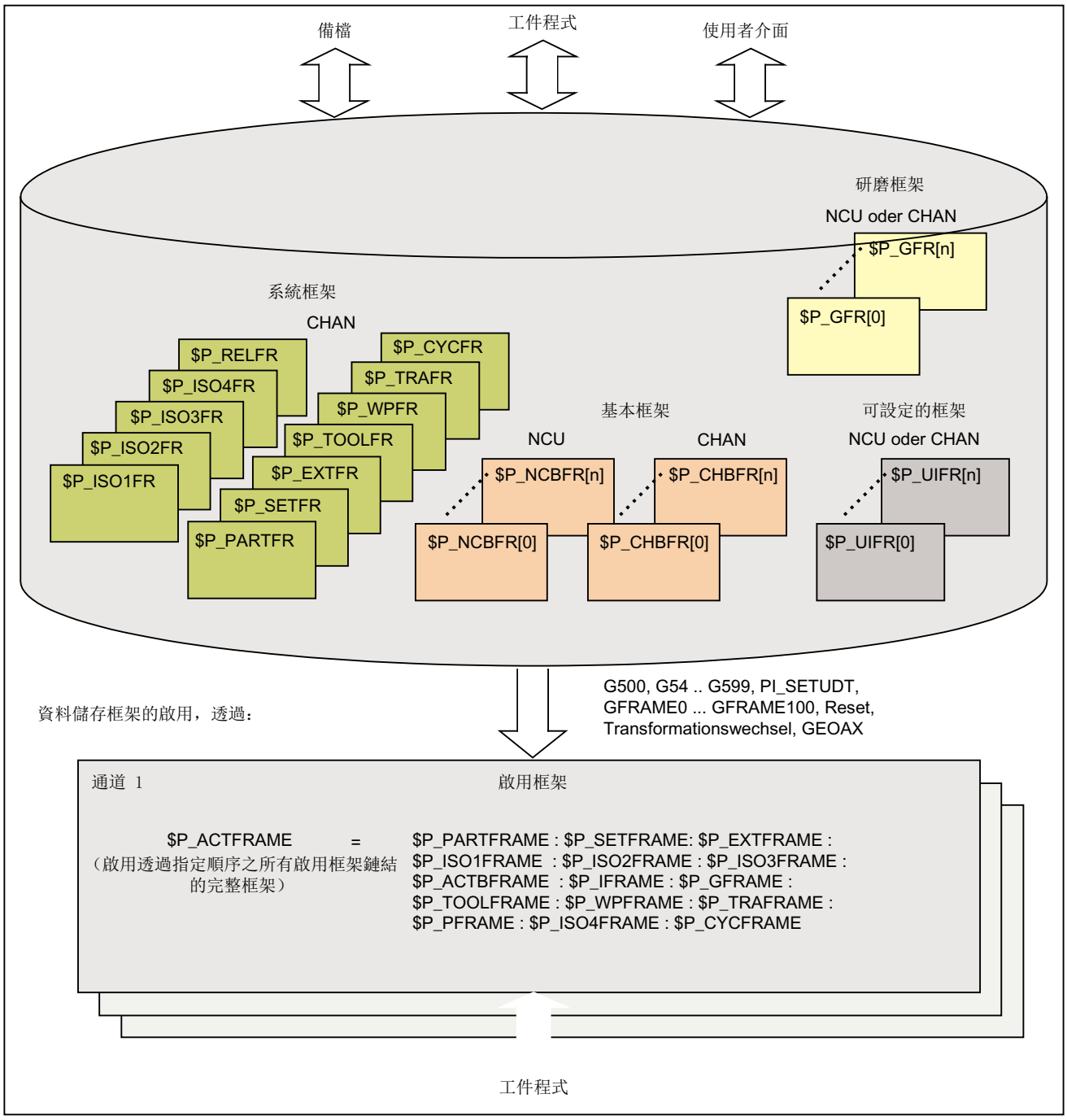
座標轉換（框架） 6

6.1 透過框架變數座標轉換

除了基礎編程手冊「座標轉換（框架）」章節所述的 ROT、AROT 及 SCALE 指令以外，工件座標系統 (WCS) 也可由框架變數 $P_...FR（資料儲存框架）及 $P_...FRAME（生效框架）轉換。

下圖概述結構框架變數：

- 資料管理框架
- 有效框架
- 生效的總框架： 所有生效框架鏈
- NCU 全域框架
- 通道專屬框架


備檔
工件程式
使用者介面
研磨框架
NCU oder CHAN
$P_GFR[n]
$P_GFR[0]
系統框架
CHAN
$P_RELFR
$P_ISO4FR
$P_ISO3FR
$P_ISO2FR
$P_ISO1FR
$P_CYCFR
$P_TRAFR
$P_WPFR
$P_TOOLFR
$P_EXTFR
$P_SETFR
$P_PARTFR
基本框架
NCU
CHAN
$P_NCBFR[n]
$P_CHBFR[n]
$P_NCBFR[0]
$P_CHBFR[0]
可設定的框架
NCU oder CHAN
$P_UIFR[n]
$P_UIFR[0]
資料儲存框架的啟用，透過：
G500, G54 .. G599, PI_SETUDT,
GFRAME0 ... GFRAME100, Reset,
Transformationswechsel, GEOAX
通道 1
啟用框架
$P_ACTFRAME =
（啟用透過指定順序之所有啟用框架鏈結的完整框架）
$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
工件程式

圖像 6-1　框架變數概觀

6.1.1 預先定義的框架變數（$P_CHBFRAME、$P_IFRAME、$P_PFRAME、$P_ACTFRAME）

啟用： 通道專屬的基準框架 $P_CHBFRAME[<n>] ($P_BFRAME)

說明

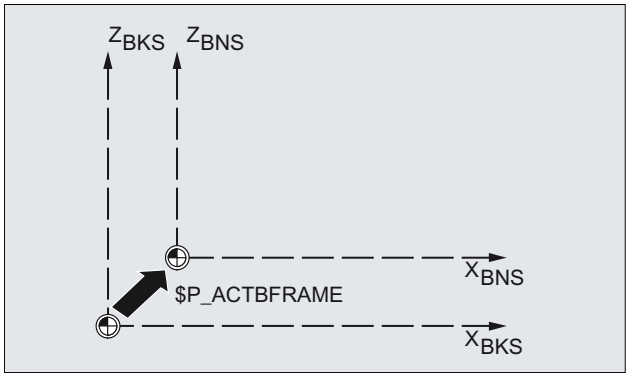
目前基準框架 $P_BFRAME 及資料儲存基準框架 $P_UBFR 因為相容性而保留。

- $P_BFRAME □ $P_CHBFRAME[0]
- $P_UBFR □ $P_CHBFR[0]。

框架變數 $P_CHBFRAME[<n>] 定義基本座標系統 (BCS) 和基本原始系統 (BOS) 之間的參考。

若目前通道專屬基準框架 $P_CHBFRAME[<n>] 應於 NC 程式之中立即生效，就可能出現下列情形。

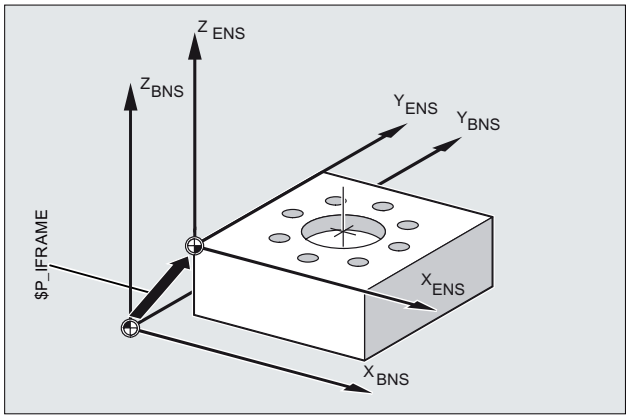
- 指令：
  - G500（停用所有可設定框架，基準框架維持生效）
  - G54 至 G599（可設定的零點偏移量）
- 將資料儲存的通道專屬基準框架，指派至目前的通道專屬基準框架：
  $P_CHBFRAME[<n>] = $P_CHBFR[<m>]



啟用： 通道專屬可設定框架 $P_IFRAME

框架變數 $P_IFRAME 定義基本原始系統 (BOS) 及可設定零點系統 (SZS) 之間的參考。

- $P_IFRAME 對應於$P_UIFR[$P_IFRNUM]
- 程式設計 G54 之後，例如 $P_IFRAME 包含由 G54 定義的轉譯、旋轉、比例縮放及鏡像。

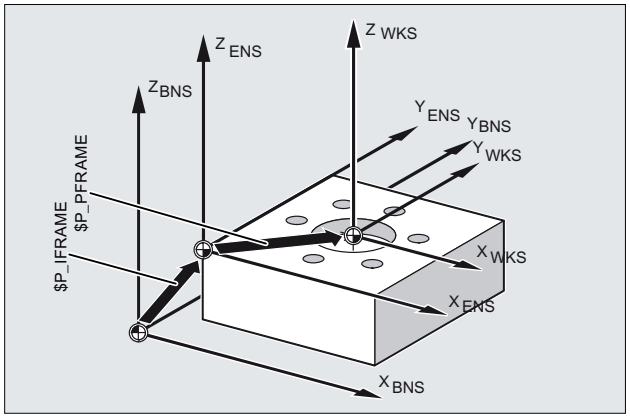


啟用： 通道專屬的可程式設計框架 $P_PFRAME

$P_PFRAME 框架變數定義可設定零點系統 (SZS) 與工件座標系統 (WCS) 之間的參考。

$P_PFRAME 包含結果框架，造成

- **程式設計** TRANS/ATRANS、ROT/AROT、SCALE/ASCALE、MIRROR/AMIRROR 或
- **指派** CTRANS、CROT、CMIRROR、CSCALE 至已程式設計的 FRAME 產生的結果框架。

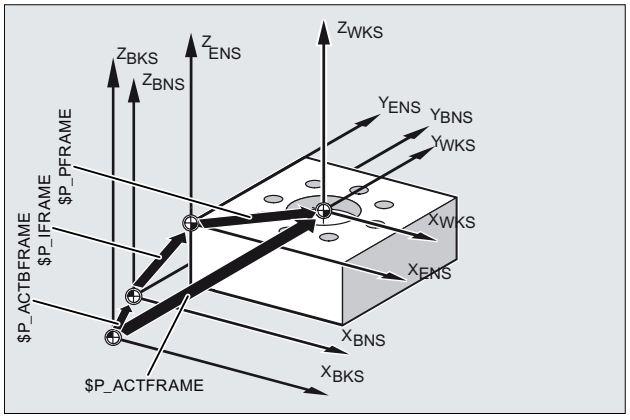


啟用： 總框架 $P_ACTFRAME

通道中生效的總框架，是將通道中作用的所有框架鏈結而成。

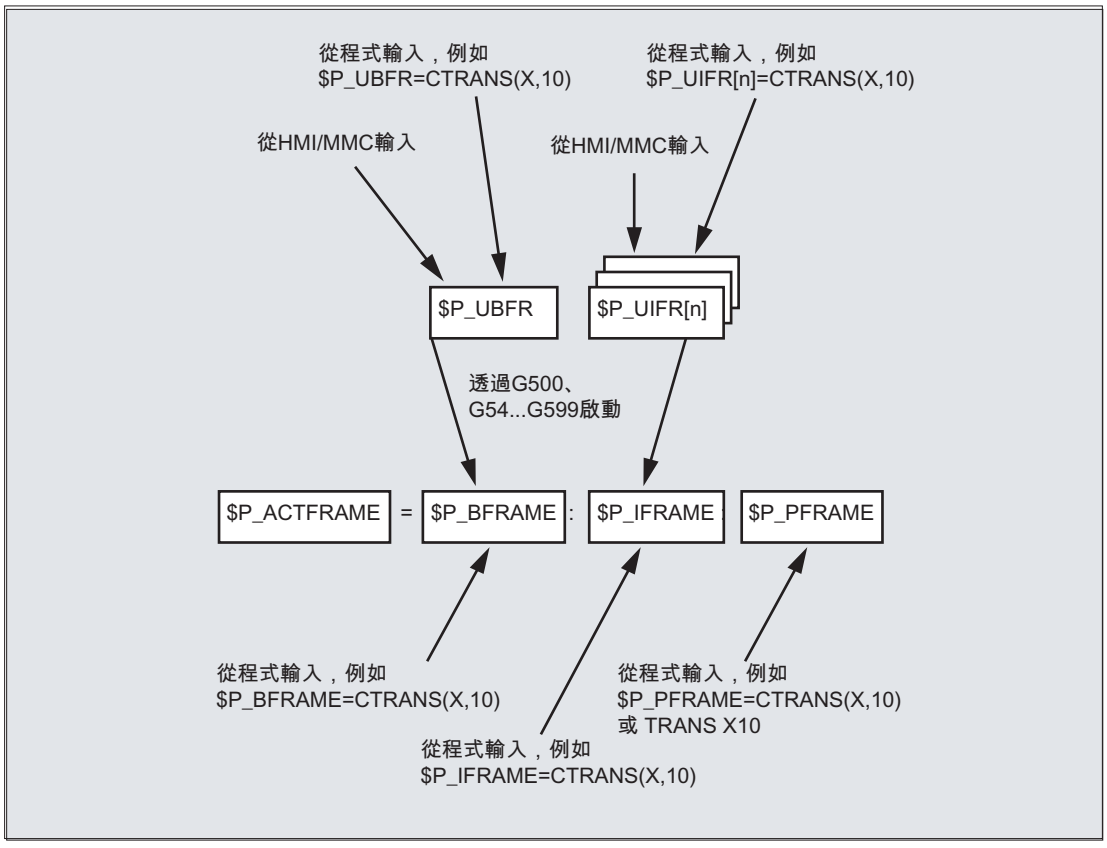
```
$P_ACTFRAME =   $P_PARTFRAME： $P_SETFRAME： $P_EXTFRAME：
                $P_ISO1FRAME： $P_ISO2FRAME： $P_ISO3FRAME：
                $P_ACTBFRAME： $P_IFRAME： $P_GFRAME：
                $P_TOOLFRAME： $P_WPFRAME： $P_TRAFRAME：
                $P_PFRAME： $P_ISO4FRAME：$P_CYCFRAME
```

$P_ACTFRAME 說明目前生效的工件座標系統。



圖像 6-2　框架變數 $P_ACTFRAME

若下列框架 $P_BFRAME / $P_CHBFRAME[<n>]、$P_IFRAME 或 $P_PFRAME 其中之一變更，就會重新計算目前的總框架 $P_ACTFRAME。



若 MD 20110 RESET_MODE_MASK 如下設定，則基本框架及可設定框架在重置後有效：

Bit0=1，bit14=1 --> $P_UBFR（基本框架）動作

Bit0=1，bit5=1 --> $P_UIFR [$P_UIFRNUM]（可設定框架）動作

資料儲存： 通道專屬基準框架 $P_CHBFR[<n>]

框架變數 $P_CHBFR[<n>] 在資料儲存讀取/寫入基準框架。 資料儲存框架寫入時，並不會在通道中立即生效。 寫入框架以下列方式啟動：

- 通道重置及 MD20110 $MC_RESET_MODE_MASK、Bit0 == 1 及 Bit14 == 1
- 指令 G500、G54 ... G57、G505 ... G599（啟用/停用基準框架及後續的目前總框架重新計算）

資料儲存： 通道專屬可設定框架 $P_UIFR[<n>]

框架變數 $P_UIFR[<n>] 在資料儲存讀取/寫入可設定的基準框架。 框架寫入時並不會在通道中立即生效。 通道中的寫入框架是以下列方式計算：

- G500 指令（停用所有可設定框架或零點偏移量）
- G54 ... G57、G505 ... G599 指令（啟動可設定框架或零點偏移量）

主動可設定的框架	資料儲存框架	（符合於指令）
$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 值指定至框架

6.2.1 指派直接值（軸值、角度、比例）

您可在 NC 程式中直接指派值至框架或框架變數。

語法

語法

```
$P_PFRAME = CTRANS(X, <偏移值>, Y, <偏移值>, Z, <偏移值>, ...)
$P_PFRAME = ROT(X, <角度>, Y, <角度>, Z, <角度>, ...)
$P_UIFR[..] = CROT(X, <角度>, Y, <角度>, Z, <角度>,  ...
$P_PFRAME = CSCALE(X, <比例>, Y, <比例>, Z, <比例>, ...)
$P_PFRAME = CMIRROR(X, Y, Z)
```

$P_CHBFRAME[<n>] 的語法和 $P_PFRAME 相同。

含義

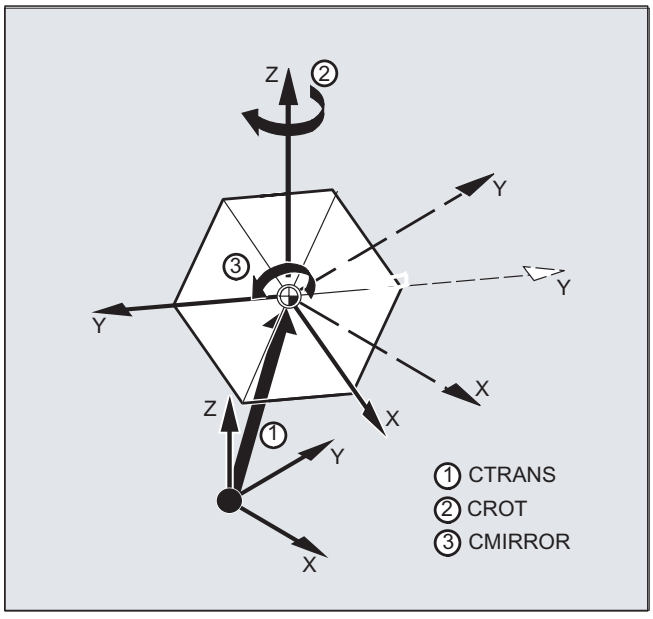
CTRANS：	指定軸之轉譯
CROT：	繞指定軸旋轉
CSCALE：	指定軸比例變更
CMIRROR：	指定軸反轉方向
X、Y、Z：	指定幾何軸之方向偏移量值
<偏移值>：	偏移值
<角度>：	旋轉角度
<比例>：	比例值

範例

指派值至目前可程式設計框架的框架元件

指派值至目前可程式設計框架的轉譯、旋轉和鏡像框架元件：

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) : CROT(Z,45) : CMIRROR(Y)
```



寫入框架的旋轉元件

以 CROT 指派值至可設定資料儲存框架 $P_UIFR 旋轉元件的所有三個軸：

```
$P_UIFR[5] = CROT(X,0,Y,0,Z,0)
```

或是直接指派個別值至資料儲存框架旋轉元件的相關軸：

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

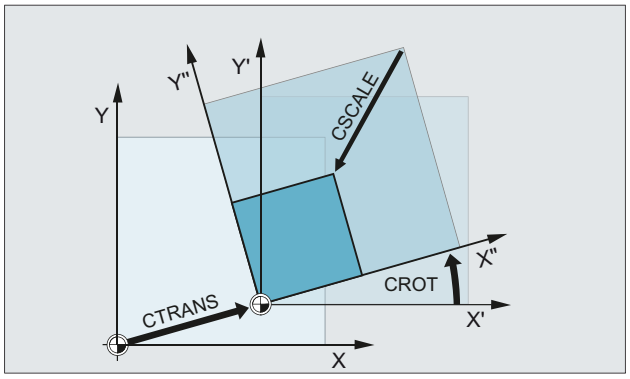
說明

鏈結運算子： 將框架上的多個運算互相結合。 之後運算將由左至右處理。

範例

在具有偏移、旋轉及比例縮放的 $P_PFRAME 進行鏈結運算：

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```



6.2.2 讀取及變更框架元件（TR、FI、RT、SC、MI）

此功能讓您可存取框架的**個別**資料，例如，特定的偏移量值或旋轉角度。 可修改這些值或將它們指派至另一變數。

語法

`R10=$P_UIFR[$P_UIFNUM，X，RT]`	從目前有效的可設定零點偏移量 $P_UIFRNUM，指派繞 X 軸的旋轉角度 RT 至變數 R10。
`R12=$P_UIFR[25，Z，TR]`	從設定框架第 25 號的資料集，指派 Z 中的偏移量值 TR 至變數 R12。
`R15=$P_PFRAME[Y，TR]`	指派目前可程式設計框架 Y 中的偏移量值 TR 至變數 R15。
`$P_PFRAME[X，TR] = 25`	修改目前可程式設計框架 X 中的偏移量值 TR。 立即套用 X25。

含義

`$P_UIFRNUM`：	該指令會自動建立參考至目前有效的可設定零點偏移量。
`P_UIFR[n,…,…]`：	指定框架編號 n 以存取可設定框架編號 n。
	指定欲讀取或修改的元件：
`TR`：	TR 轉譯
`FI`：	FI 轉譯微調
`RT`：	RT 旋轉
`SC`：	SC 調整比例修改

MI：	MI 鏡像
X、Y、Z：	亦指定對應軸 X、Y、Z（請見範例）。

RT 旋轉值域

繞第 1 幾何軸旋轉： -180°到+180°
繞第 2 幾何軸旋轉： -90°到+90°
繞第 3 幾何軸旋轉： -180°到+180°

說明

呼叫框架

藉由指定系統變數$P_UIFRNUM，您可使用$P_UIFR 或 G54、G55、...
存取目前零點偏移量集（$P_UIFRNUM 包含目前設定框架數量）。

可透過指定適當數量的$P_UIFR[n]，呼叫其它已儲存的可設定$P_UIFR 框架。

對預先定義的框架變數或使用者自訂框架，指定其名稱，例如，$P_IFRAME。

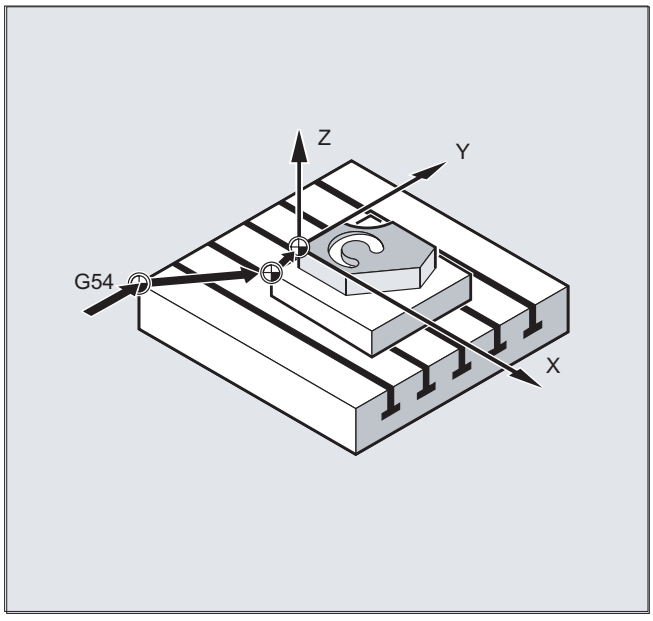
呼叫資料

將欲存取或修改之值的軸名稱及框架元件，寫入至方括號中，例如，[X，RT] 或 [Z，MI]。

6.2.3 使用框架計算

框架可指派至其它框架，或可於 NC 程式中相互鏈結框架。

框架鏈結適合說明多個工件，其配置於同個程序中待加工的棘爪。



框架元件僅可包含棘爪作業說明的中間值。 並鏈結以產生多個工件零點。

範例

指定

程式碼	註解
DEF FRAME SETTING_1	；本機框架變數定義
SETTING_1 = CTRANS(X,10)	；指派函數結果至框架變數
$P_PFRAME = SETTING_1	；指派框架變數至目前框架
DEF FRAME SETTING_4	；本機框架變數定義
SETTING_4 = $P_PFRAME	；緩衝框架變數之中的目前框架
...	
$P_PFRAME = SETTING_4	；由框架變數提取目前框架

鏈結

運算子： 以設定順序讓框架互相鏈結。 像是偏移和旋轉等框架元件，會在之後附加執行。

程式碼	註解
$P_IFRAME = $P_UIFR[15] : $P_UIFR[16]	；指派結果框架：鏈結 ；兩個可設定的資料儲存框架，在生效的 ；可設定總框架上。 ；應用範例： ；$P_UIFR[15]：偏移量 ；$P_UIFR[16]：旋轉

程式碼	註解
$P_UIFR[3] = $P_UIFR[4]： $P_UIFR[5]	；指派結果框架：鏈結 ；兩個可設定的資料儲存框架，在 ；不同的可設定資料儲存框架上

6.2.4 框架變數定義 (DEF FRAME)

除了預先定義的框架變數，也可以定義使用者框架變數。 使用者定義框架變數為 FRAME 類型的使用者變數。 框架名稱可依據使用者變數規則自由指派。

CTRANS、CROT、CSCALE 及 CMIRROR 函數指派值至使用者定義框架變數。

語法

```
DEF FRAME <名稱>
```

含義

DEF FRAME：	定義 FRAME 類型的使用者變數。
<名稱>：	框架變數名稱

範例

定義「PALETTE」框架變數，並指派偏移和旋轉值：

程式碼	註解
DEF FRAME PALETTE	；定義 PALETTE 框件變數
PALETTE = CTRANS(...)： CROT(...)	；指派鏈結的結果框架，適用於 ；PALETTE 框架變數的偏移及旋轉

6.3 粗調或微調偏移量（CTRANS、CFINE）

微調偏移量

微調偏移 CFINE(...) 可套用至下列框架：

- 可設定框架： $P_UIFR 或 $P_IFRAME
- 基本框架： $P_NCBFR[<n>]、$P_CHBFR[<n>]、$P_CHBFRAMES[<n>] 或 $P_ACTBFRAME
- 可程式設計框架：$P_PFRAME

使用 CFINE(...) 指令對框架的微調偏移量進行程式設計。

粗調偏移量

粗調偏移 CTRANS(...) 可套用至所有框架。

總偏移量量

總偏移是由加上粗調及微調偏移而產生。

機械資料

啟用微調偏移

微調偏移是以下列機械資料啟用：

MD18600 $MN_MM_FRAME_FINE_TRANS = 1

語法

微調偏移量

- 完整框架
  - <框架> = CFINE(<K_1>,<值>)
  - <框架> = CFINE(<K_1>,<值>, <K_2>, <值>)
  - <框架> = CFINE(<K_1>,<值>, <K_2>, <值>, <K_3>, <值>)
- 框架元件
  - <框架>[<n>, <K_1>, FI] = <值>

粗調偏移量

- 完整框架
  - <框架> = CTRANS(<K_1>,<值>)
  - <框架> = CTRANS(<K_1>,<值>, <K_2,<值>)
  - <框架> = CTRANS(<K_1>,<值>, <K_2,<值>, <K_3,<值>)
- 框架元件
  - <框架>[<n>,<K_1>,TR] = <值>

特別是可程式設計框架 $P_PFRAME：

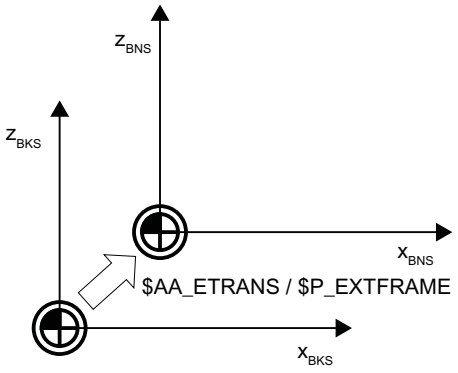
- TRANS <K_1> <值>
- TRANS <K_1> <值> <K_2> <值>
- TRANS <K_1> <值> <K_2> <值> <K_3> <值>

含義

<框架>：	框架，例如資料儲存的可設定框架 $P_UIFR[<n>]
CFINE：	微調偏移，附加偏移。
CTRANS：	粗調偏移，絕對偏移。
TRANS：	僅限可程式設計框架： 粗調偏移，絕對偏移。
<K_n>：	座標軸 X、Y、Z
<值>：	偏移值

6.4 外部零點偏移量 ($AA_ETRANS)

外部零點偏移量是基準座標系統 (BCS) 和基本原始系統 (BOS) 之間的線性偏移。



外部零點偏移量搭配 $AA_ETRANS 會以兩種方式作用，需視機械資料的參數設定而定：

1. 由 NC/PLC 介面訊號啟用之後，系統變數 $AA_ETRANS 會直接作為偏移值
2. 由 NC/PLC 介面訊號啟用後，系統變數值 $AA_ETRANS 將傳輸至生效系統框架 $P:EXTFRAME 及資料儲存框架 $P_EXTFR。之後將重新計算生效的總框架 $P_ACTFRAME。

機械資料

下列機械資料可搭配系統變數 $AA_ETRANS，在選擇的兩種步驟之間做出區隔：

MD28082 $MC_MM_SYSTEM_FRAME_MASK,Bit1 = <值>

<值>	含義
0	功能：$AA_ETRANS[<軸>] 直接以 PLC、HMI 或 NC 程式寫入。 在下一個可能的移動單節中，針對 $AA_ETRANS[<軸>] 啟用零點偏移量回退：DB31, ... DBX3.0
1	功能：啟動生效系統框架 $P:EXTFRAME 及資料儲存框架 $P_EXTFR 以下列方式啟用 $AA_ETRANS[<軸>] 的零點偏移量回退：DB31、... DBX3.0。 以下於通道中執行： ● 停止通道中的所有移動（除了指令及 可程式邏輯控制器軸） ● 前置處理停止及後續重組 (STOPRE) ● 生效框架 $P_EXTFRAME[<軸>] = $AA_ETRANS[<軸>] 的粗調偏移 ● 資料儲存框架 $P_EXTFR[<軸>] = $AA_ETRANS[<軸>] 的粗調偏移 ● 重新計算生效的總框架 $P_ACTFRAME ● 於編程設定軸的偏移回退。 ● 繼續中斷的移動動作或 NC 程式

程式設計

- 語法
  $AA_ETRANS[<軸>] = <值>
- 含義

$AA_ETRANS：	緩衝外部零點偏移量的系統變數
<軸>：	通道軸
<值>：	偏移值

NC/PLC 介面訊號

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<軸>] = $P_EXTFR[<軸>] = $AA_ETRANS[<軸>]

6.5 設定喪失參考狀態的實際值 (PRESETON)

PRESETON() 步驟可於機械座標系統 (MCS) 針對一個以上的軸設定新實際值。 這符合於軸機械座標系統 (MCS) 的零點偏移量。 這不會造成軸移動。

PRESETON 以同步起始前置處理停止。 實際位置僅於停止時指派至軸。

若 PRESETON 軸並未指派至通道，後續步驟需視軸更換行為的軸專屬設定而定：

MD30552 $MA_AUTO_GET_TYPE

參考狀態

在機械座標系統設定新的實際值，可重置機械軸的參考狀態：

DB31、... DBX60.4/.5 = 0（參考/同步的量測系統 1/2）

因此建議 PRESETON 僅用於不需要參考點的軸。

如欲還原原始的機械座標系統，就必須再次參考機械軸的量測系統，例如由工件程式 (G74) 透過生效參考進行。

⚠ 小心
喪失參考狀態 使用 PRESETON 在機械座標系統設定新的實際值，會將機械軸的參考狀態重置為「不參考/同步」。

程式設計

語法

```
PRESETON(<軸_1>, <值_1> [, <軸_2>, <值_2>, ... <軸_8>, <值_8>])
```

含義

PRESETON：	設定喪失參考狀態的實際值	
	前置處理停止：	是
	單獨在單節中：	是
<axis_x>：	機械軸名稱	
	類型：	AXIS
	值域：	於通道中定義的機械軸名稱

<value_x>：	機械座標系統 (MCS) 中機械軸的新實際值 以目前有效的量測系統（英吋/公制）進行輸入 要考量生效的直徑程式設計 (DIAMON)	
	類型：	REAL

參考資料

NC 程式之中的 PRESETONS

NC 程式之中的 PRESETON 詳細說明請參閱：

功能手冊基本功能，章節「K2： 軸、座標系統、框架」>「座標系統」>「機械座標系統 (MCS)」>「設定喪失參考狀態的實際值 (PRESETON)」

同步動作中的 PRESETONS

同步動作中的 PRESETON 詳細說明請參閱：

功能手冊，已同步動作；章節： 「詳細說明」>「同步動作中的動作」>「設定喪失參考狀態的實際值 (PRESETON)」

6.6 設定並未喪失參考狀態的實際值 (PRESETONS)

PRESETONS() 步驟可於機械座標系統 (MCS) 針對一個以上的軸設定新實際值。 這符合於軸機械座標系統 (MCS) 的零點偏移量。 這不會造成軸移動。

PRESETONS 以同步起始前置處理停止。 實際位置僅於停止時指派至軸。

若 PRESETONS 軸並未指派至通道，後續步驟需視軸更換行為的軸專屬設定而定：

MD30552 $MA_AUTO_GET_TYPE

參考狀態

以 PRESETONS 在機械座標系統 (MCS) 設定新實際值，**不會**變更機械軸的參考狀態。

條件

- **編碼器類型**
  PRESETONS 僅適用於下列生效量測系統的編碼器類型：
  - MD30240 $MA_ENC_TYPE[<量測系統>] = 0（模擬編碼器）
  - MD30240 $MA_ENC_TYPE[<量測系統>] = 1（原始訊號編碼器）
- **參考模式**
  PRESETONS 僅適用於下列生效量測系統的參考模式：
  - MD34200 $MA_ENC_REFP_MODE[<量測系統>] = 0（無法原點復歸）
  - MD34200 $MA_ENC_REFP_MODE[<量測系統>] = 1（參考增量、旋轉或線性量測系統： 在編碼器軌的零點脈衝）

程式設計

語法
```
PRESETONS(<軸_1>, <值_1> [, <軸_2>, <值_2>, ... <軸_8>, <值_8>])
```

含義

PRESETONS：	設定並未喪失參考狀態的實際值	
	前置處理停止：	是
	單獨在單節中：	是
<axis_x>：	機械軸名稱	
	類型：	AXIS
	值域：	於通道中定義的機械軸名稱

<value_x>：	機械座標系統 (MCS) 中機械軸目前的新實際值 以生效的量測系統（英吋/公制）進行輸入 要考量生效的直徑程式設計 (DIAMON)	
	類型：	REAL

參考資料

NC 程式之中的 PRESETONS

NC 程式之中的 PRESETONS 詳細說明請參閱：

功能手冊基本功能，章節「K2： 軸、座標系統、框架」>「座標系統」>「機械座標系統 (MCS)」>「設定並未喪失參考狀態的實際值 (PRESETONS)」

同步動作中的 PRESETONS

同步動作中的 PRESETONS 詳細說明請參閱：

功能手冊，已同步動作；章節： 「詳細說明」>「同步動作中的動作」>「設定並未喪失參考狀態的實際值 (PRESETONS)」

6.7 從空間中的三個量測點計算框架（MEAFRAME）

MEAFRAME 功能用於支援量測循環。 從三處理想點及對應的量測點計算框架。

當工件已定位待加工，一般來說，其相對於直角機械座標系統之位置會參考其理想位置移動及旋轉。 對精確加工或量測，需費時實體調整工件或工件程式中定義的動作必須變更。

可透過取樣空間中已知其理想位置的三點定義框架。 觸發式探針或光學感測器可用來取樣，觸碰在支撐板或探頭精確固定的特別孔洞。

語法

```
MEAFRAME（<理想點>，<量測點>，<品質>）
```

含義

MEAFRAME：	功能呼叫		
<理想點>：	二維 包含理想點三座標的 REAL 陣列		
<量測點>：	二維 包含量測點三座標的 REAL 陣列		
<品質>：	回傳 FRAME 計算的品質資訊變數		
	類型：	VAR REAL	
	值：	-1	理想點幾乎位於一條直線上： 無法計算框架。 傳回的 FRAME 變數包含一個中性框架。
		-2	量測點幾乎位於一條直線上： 無法計算框架。 傳回的 FRAME 變數包含一個中性框架。
		-4	旋轉矩陣的計算因不同原因而失敗。
		≥ 0.0	需要失真總和（點之間的距離）將量測的三角形轉換為與理想三角形全等的三角形。

說明

量測品質

為使用旋轉及轉譯將已量測之座標繪製於理想座標上，經量測點形成之三角形需與理想三角形全等。 可透過補正演算法達成，以將量測三角形重塑為理想三角形所需的離差平方和減至最低。

由於可使用有效扭曲來判斷量測的品質，MEAFRAME 將其視為額外變數傳回。

說明

由 MEAFRAME 建立的框架可由 ADDFRAME 函數在框架鏈結中轉換為另一框架。（請參閱範例〈以 ADDFRAME 鏈結〉）

範例

範例 1：

工件程式 1：

程式碼
...
DEF FRAME CORR_FRAME

設定量測點：

程式碼	註解
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]= SET(10.1,0.2,-0.2,-0.2,10.2,0.1,-0.2,0.2,9.8)	；用於測試。
DEF REAL FIT_QUALITY=0	
DEF REAL ROT_FRAME_LIMIT=5	；最多允許工件位置旋轉五度。
DEF REAL FIT_QUALITY_LIMIT=3	；理想與量測三角形 間最大允許三毫米的偏移量。
DEF REAL SHOW_MCS_POS1[3]	
DEF REAL SHOW_MCS_POS2[3]	
DEF REAL SHOW_MCS_POS3[3]	

程式碼	註解
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	

程式碼	註解
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT	
SETAL（65010）	
GOTOF NO_FRAME	
ENDIF	
N250 IF CORR_FRAME[X，RT] > ROT_FRAME_LIMIT	；限制第一 RPY 角度。
SETAL（65020）	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y，RT] > ROT_FRAME_LIMIT	；限制第二 RPY 角度。
SETAL（65021）	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z，RT] > ROT_FRAME_LIMIT	；限制第三 RPY 角度。
SETAL（65022）	
GOTOF NO_FRAME	
ENDIF	
N300 $P_IFRAME=CORR_FRAME	；啟動含可設定框架之樣本框架。
	；將幾何軸定位於理想點以檢查框架。
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=$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	；停用可設定框架，如同用零點框架（無輸入、預先指派之值）。
No_FRAME	；停用可設定框架，如同以零點框架預先指派（無輸入值）。
M0	
M30	

範例 2： 框架鏈結

偏移量 MEAFRAME 鏈結

MEAFRAME 函數回傳一個偏移框架。 若此偏移框架鏈結至可設定框架 $P_UIFR[1]（如 G54 等函數呼叫時啟用），可設定框架則可用於進一步的移動或加工轉換。

以 ADDFRAME 鏈結

若您希望該框架鏈結中的偏移框架套用至其他位置，或是其他框架比可設定框架先啟用，可使用 ADDFRAME 函數鏈結至其中一個通道基本框架或系統框架。

下列為框架中不可啟用的：

- 以 MIRROR 產生鏡像
- 以 SCALE 產生比例

設定點及實際值之輸入參數為工件座標。 在基本的控制系統中，上述座標必須以公制設定，或以英吋 (G71/G70) 和相關半徑 (DIAMOF) 來定義。

參考資料：
如需有關 ADDFRAME 更進一步的資訊，請參閱：
功能手冊，基本功能，K2： 軸類型，座標系統，框架

6.8 NCU 全域框架

NCU 所有通道僅使用一組 NCU 全域框架。 可於所有通道讀寫 NCU 全域框架。 NCU 全域框架在個別通道中啟用。

含偏移量之**通道軸及機械軸**可透過全域框架產生比例及鏡像。

幾何關係及框架鏈結

使用全域框架時，軸之間並無幾何關係。 因此無法執行旋轉或程式設計幾何軸辨識碼。

- 全域框架上無法使用旋轉。 拒絕程式設計旋轉，並發出警報： 顯示「18310 通道%1 單節%2 框架： 不允許旋轉」。
- 可鏈結全域框架及通道專屬框架。 結果框架包括內含所有軸旋轉的所有框架元件。 拒絕指派含旋轉元件的框架至全域框架，並輸出警報「框架： 不允許旋轉」。

NCU 全域框架

NCU 全域基本框架 $P_NCBFR[n]

最多可設定八個 NCU 全域基本框架：

亦可用通道專屬基本框架。

可於所有 NCU 通道讀寫全域框架。 寫入全域框架時，使用者需確保通道座標。 例如，可透過等待標記 (WAITMC) 實現。

機台製造商

經由機械資料設定全域基本框架的數量。

參考資料：
功能手冊，基本功能；座標軸，座標系統，框架 (K2)

NCU 全域可設定框架 $P_UIFR[n]

可經由 NCU 全域設定或針對各通道設定所有的可設定框架 G500、G54...G599。

機台製造商

可透過機械資料 MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES 的協助，重新將所有可設定框架設定為全域框架。

可使用通道軸識別碼及機械軸識別碼，作為框架程式指令中的座標軸識別碼。 拒絕程式設計幾何識別碼，並發出警報。

6.8.1 通道專屬框架（$P_CHBFR、$P_UBFR）

操作者和 PLC 可透過工件程式和 OPI 讀取和寫入可設定框架或基本框架。

微調偏移量也可供全域框架使用。 透過 G53、G153、SUPA 及 G500 亦產生全域框架停止，與通道專屬框架情況相同。

機台製造商

可在通道中透過機械資料 MD28081 $MC_MM_NUM_BASE_FRAMES 設定基本框架的數量。 標準設定為每個通道至少一個基本框架。 每個通道最多支援八個基本框架。 除八個基本框架外，通道中也可有八個 NCU 全域基本框架。

通道專屬框架

$P_CHBFR[n]

可使用系統變數 $P_CHBFR[n] 讀寫基本框架。 當寫入基本框架，執行 G500、G54...G599 陳述式前不會啟動鏈結的總基本框架。 變數主要係用來儲存寫入作業至 HMI 或 PLC 上的基本框架。 該框架變數係由資料備份儲存。

通道中第一個基本框架

寫入至預先定義的 $P_UBFR 變數時，不會同時啟動含陣列索引 0 的基本框架，而是執行 G500、G54...G599 陳述式後才會啟動。 也可於程式中讀寫該變數。

$P_UBFR

$P_UBFR 與 $P_CHBFR[0] 相同。 預設的通道中永遠存在一個基本框架，因此系統變數可與較舊版本相容。 若無通道專屬基本框架，讀取 / 寫入時會發出警報：「框架： 不允許陳述式」。

6.8.2 通道中有效框架

從工件程式透過該框架相關系統變數，輸入通道中的有效框架。 亦包括系統框架。 可透過上述系統變數，在工件程式中讀寫目前的系統框架。

目前通道中的有效框架

概觀

目前系統框架	為：
$P_PARTFRAME	TCARR 及 PAROT
$P_SETFRAME	預設實際值記憶體和清除

$P_EXTFRAME	外部零點偏移量
$P_NCBFRAME[n]	目前 NCU 全域基本框架
$P_CHBFRAME[n]	目前通道基本框架
$P_BFRAME	目前通道中第一個基本框架
$P_ACTBFRAME	完整基本框架
$P_CHBFRMASK 及$P_NCBFRMASK	完整基本框架
$P_IFRAME	目前可設定框架
目前系統框架	為：
$P_TOOLFRAME	TOROT 及 TOFRAME
$P_WPFRAME	工件參考點
$P_TRAFRAME	轉換
$P_PFRAME	目前可程式設計框架
目前系統框架	為：
$P_CYCFRAME	循環
P_ACTFRAME	目前總框架
框架鏈結	目前框架由完整的基本框架所組成

$P_NCBFRAME [n] 目前 NCU 全域基本框架

可使用系統變數$P_NCBFRAME[n]讀寫目前全域基本框架欄位元件。 由通道中的寫入程序計算出所得的總基本框架。

僅當通道中之框架已程式設計時，才會啟動已修改之框架。 若欲為 NCU 中所有通道修改框架，需同時寫入$P_NCBFR[n]及$P_NCBFRAME[n]。 此時其他通道必須啟動框架，例如，使用 G54。 無論何時寫入基本框架，皆會再次計算完整基本框架。

$P_CHBFRAME[n]目前通道基本框架

可使用系統變數$P_CHBFRAME[n]讀寫目前通道基本框架欄位元件。 由通道中的寫入程序計算出所得的完整基本框架。 無論何時寫入基本框架，皆會再次計算完整基本框架。

$P_BFRAME 目前通道中第一個基本框架

可使用預先定義的框架變數 $P_BFRAME，讀寫目前含陣列索引 0 的基本框架，其於通道、工件程式中有效。 寫入的基本框架立即包含於計算中。

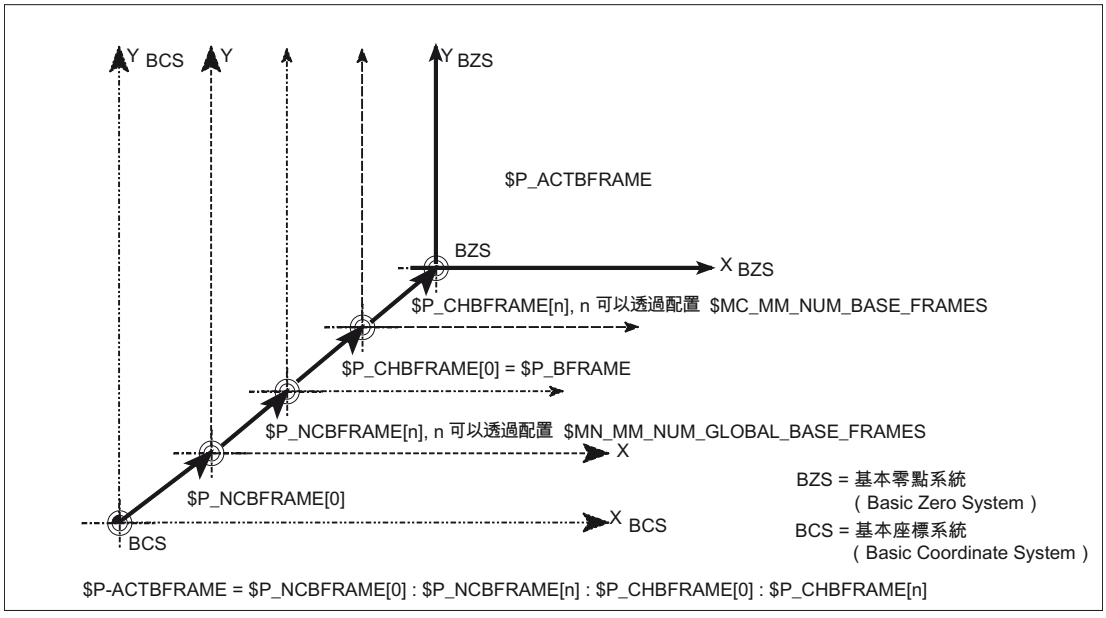
$P_BFRAME 與 $P_CHBFRAME[0] 相同。 系統變數永遠有一個有效的預設值。 若無通道專屬基本框架，讀取 / 寫入時會發出警報：「框架： 不允許陳述式」。

$P_ACTBFRAME 完整基本框架

$P_ACTFRAME 變數決定鏈結的完整基本框架。 變數為唯讀。

$P_ACTFRAME 對應於：

$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n].



$P_CHBFRMASK 及 $P_NCBFRMASK 完整基本框架

使用者可透過系統變數 $P_CHBFRMASK 和 $P_NCBFRMASK，選擇「完整」基本框架計算中包含的基本框架。 變數僅可於程式中編寫，並透過 OPI 來讀取。 變數值經轉譯為位元遮罩，並指定計算中所包含的 $P_ACTFRAME 基本框架欄位元件。

可使用 $P_CHBFRMASK 指定計算中所包含的通道專屬基本框架，並可使用 $P_NCBFRMASK 指定計算中所包含的 NCU 全域基本框架。

完整的基本框架和完整框架將以變數程式重新計算。 重置後，$P_CHBFRMASK 和 $P_NCBFRMASK 的基本設定值如下：

$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK

$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK

範例：

```
$P_NCBFRMASK = 'H81'  ；$P_NCBFRAME[0] ： $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11'  ；$P_CHBFRAME[0] ： $P_CHBFRAME[4]
```

$P_IFRAME 目前可設定框架

可使用預先定義的框架變數 $P_IFRAME 讀寫目前可設定框架，其於通道、工件程式中有效。 寫入的可設定框架立即包含於計算中。

在 NCU 全域可設定框架的範例中，修改過的框架僅在已編寫框架的通道中才會動作。 若想為 NCU 的所有通道修改框架，必須同時寫入 $P_UIFR[n] 及 $P_IFRAME。 而此時其他通道必須啟動對應的框架，例如，使用 G54。

$P_PFRAME 目前可程式設計框架

$P_PFRAME 為可程式設計的框架，來自 TRANS/ATRANS、G58/G59、ROT/AROT、SCALE/ASCALE、MIRROR/AMIRROR 的程式設計，或是從 CTRANS、CROT、CMIRROR、CSCALE 指定至可程式設計的框架。

目前可程式設計的框架變數，會建立可設定零點系統（SZS）與工件座標系統（WCS）間的參考資料。

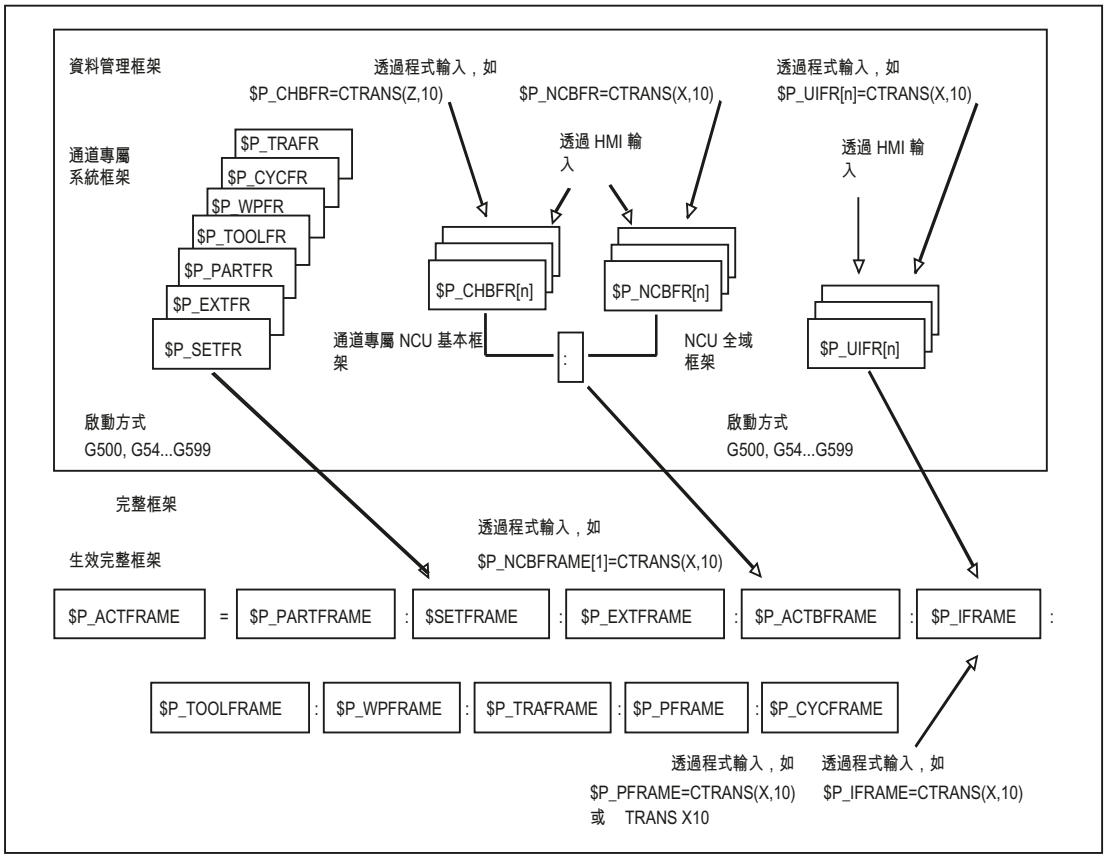
P_ACTFRAME 目前完整框架

所得的目前完整框架$P_ACTFRAME，現為所有基本框架、目前可設定框架及可程式設計的框架間的鏈結。 只要變更框架元件，便會更新目前框架。

$P_ACTFRAME 對應於：

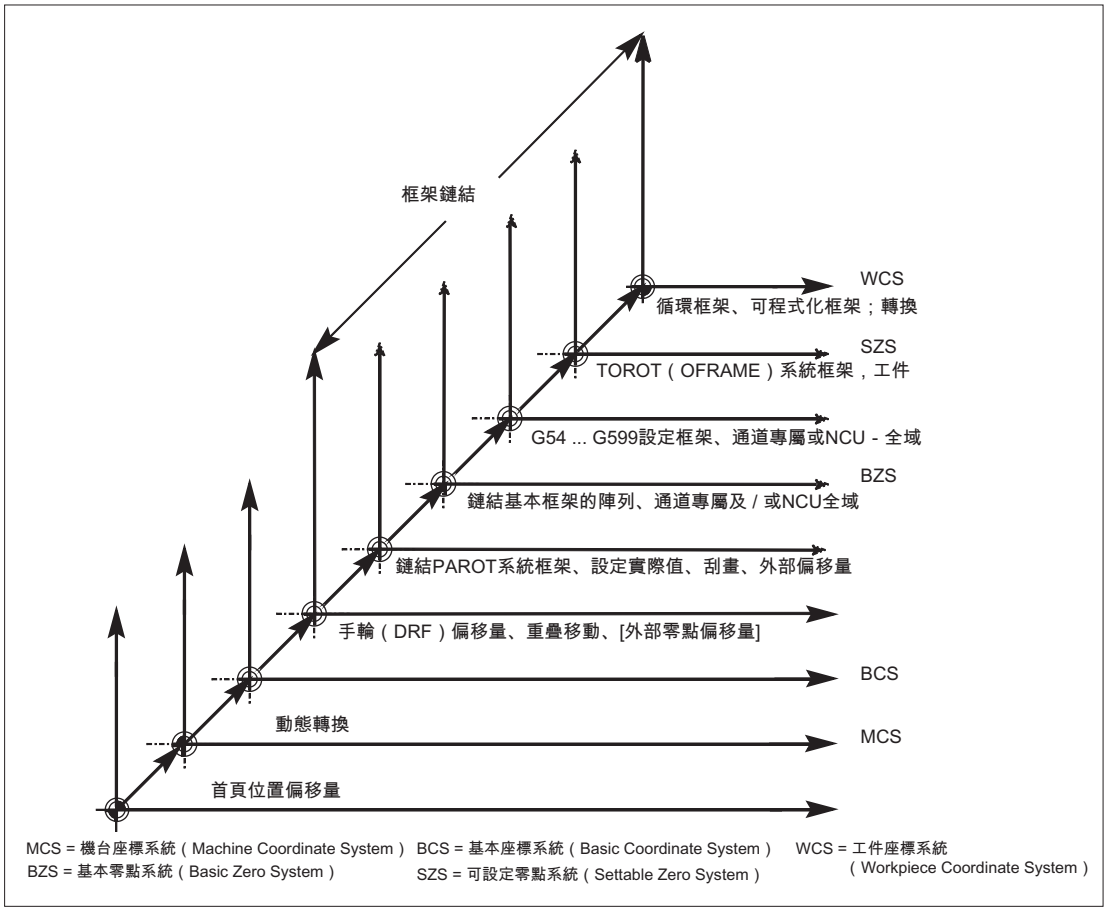
$P_PARTFRAME： $P_SETFRAME： $P_EXTFRAME： $P_ACTBFRAME： $P_IFRAME：

$P_TOOLFRAME： $P_WPFRAME： $P_TRAFRAME： $P_PFRAME：$P_CYCFRAME


資料管理框架
透過程式輸入，如
$P_CHBFR=CTRANS(Z,10)
$P_NCBFR=CTRANS(X,10)
透過程式輸入，如
$P_UIFR[n]=CTRANS(X,10)
通道專屬
系統框架
$P_TRAFR
$P_CYCFR
$P_WPFR
$P_TOOLFR
$P_PARTFR
$P_EXTFR
$P_SETFR
透過 HMI 輸入
透過 HMI 輸入
$P_CHBFR[n]
$P_NCBFR[n]
$P_UIFR[n]
通道專屬 NCU 基本框架
:
NCU 全域框架
啟動方式
G500, G54...G599
啟動方式
G500, G54...G599
完整框架
透過程式輸入，如
$P_NCBFRAME[1]=CTRANS(X,10)
生效完整框架
$P_ACTFRAME
=
$P_PARTFRAME
:
$SETFRAME
:
$P_EXTFRAME
:
$P_ACTBFRAME
:
$P_IFRAME
:
$P_TOOLFRAME
:
$P_WPFRAME
:
$P_TRAFRAME
:
$P_PFRAME
:
$P_CYCFRAME
透過程式輸入，如
$P_PFRAME=CTRANS(X,10)
或 TRANS X10
透過程式輸入，如
$P_IFRAME=CTRANS(X,10)

框架鏈結

根據上述的目前完整框架，目前框架由完整基本框架、可設定框架、系統框架及可程式設計框架組成。



轉換 7

7.1 轉換類型的一般程式設計

一般功能

您可選擇使用適當的參數編寫轉換類型，讓控制器適應多種機械動態。 可使用該參數，根據所選的轉換，宣告空間中的刀具方向及旋轉軸的方向移動。

三、四及五軸轉換中，已程式設計的位置資料永遠與刀具的刀尖相關，在空間中與已加工的平面成直角移動。 直角座標從基本座標系統轉換至機械座標系統，且與幾何軸相關。 由此定義操作點。 虛擬旋轉軸定義空間中的刀具方向，並且使用 TRAORI 程式設計。

在動態轉換的情況下，可於直角座標系統中程式設計位置。 控制器將使用 TRANSMIT、TRACYL 及 TRAANG 編寫的直角座標系統移動動作，安排至實際機械軸的移動動作上。

程式設計

三、四及五軸轉換（TRAORI）

使用 TRAORI 指令及轉換數量、方向向量及旋轉軸偏移量三個可能的參數，啟動宣告的方向轉換。

TRAORI（轉換數量，方向向量，旋轉軸偏移量）

動態轉換

TRANSMIT（轉換數量）經宣告的轉換為動態轉換的範例。

TRACYL（作業直徑，轉換數量）

TRAANG（偏移軸角度，轉換數量）

停用有效轉換

可使用 TRAFOOF 來停用目前有效轉換。

方向轉換

三、四及五軸轉換（TRAORI）

對於機台工作區空間中，所設定的表面最佳加工，加工刀具除 X、Y 及 Z 三個線性軸外，還需其他的軸。其他軸定義空間中的方向，且於後續區段中稱做方向軸。 它們可利用為四種具有不同動力學的機台上之旋轉軸。

1. 兩軸迴轉頭，例如，在固定刀具台上，一個旋轉軸與線性軸平行的萬向刀具頭。
2. 兩軸旋轉台，例如，固定迴轉頭搭配可繞兩軸旋轉的刀具台。
3. 單軸迴轉頭及單軸旋轉台，例如，一個可旋轉迴轉頭搭配旋轉刀具，其刀具台可繞一軸旋轉。
4. 兩軸迴轉頭及單軸旋轉台，例如，可繞一軸旋轉的刀具台，搭配一個刀具可自轉的可旋轉迴轉頭。

3 及 4 軸轉換為 5 軸轉換的特殊類型，並以與 5 軸轉換相同的方式程式設計。

「**一般 3/4/5/6 軸轉換**」的函數範圍，適合直角旋轉軸及通用銑削頭之轉換，與其他的方向轉換相同，也可使用 TRAORI 啟動這四種機台類型。 在一般 5/6 軸轉換情況下，刀具方向有第三度自由度，刀具可相對於刀具方向自轉，因此可在空間中依所需導向。

參考資料： /FB3/ 功能手冊，特殊函數；3 至 5 軸轉換（F2）

不管動力學，初始刀具方向設定

ORIRESET

若使用 TRAORI 啟用方向轉換，則 ORIRESET 可用來以選配的參數 A、B、C，指定最多三方向軸的初始設定。已程式設計的參數順序，依照由轉換所定義的方向軸順序指派至圓軸。程式設計 ORIRESET（A，B，C），使方向軸以線性及同步動作，從其目前位置移動至指定的初始設定位置。

動態轉換

TRANSMIT 與 TRACYL

對車削機台上銑削，

1. 使用 TRANSMIT 於車削夾具平面加工，或
2. 使用 TRACYL 在圓柱體上以任何路徑溝槽加工

可為宣告轉換程式設計。

TRAANG

若需要傾斜進給的設定進給軸選項（例如，用於研磨技術），可使用 TRAANG 為已宣告的轉換，程式設計一可設定角度。

直角座標 PTP 移動

動態轉換亦包括所謂的「直角座標 PTP 移動」，最多可程式設計 8 個不同的關節接頭位置 STAT=。 雖然位置已程式設計在直角座標系統中，但在機械座標系統中會發生機台移動。

參考資料：
/FB2/ 功能手冊，延伸功能，動態轉換 (M1)

鏈結轉換

可接連切換兩轉換。 在此鏈接的第二個轉換，會從第一個轉換取得軸的動作零件。

第一個轉換可為：

- 方向轉換 TRAORI
- 極轉換 TRANSMIT
- 圓柱轉換 TRACYL
- 傾斜軸轉換 TRAANG

第二個轉換需為傾斜軸 TRAANG 類型的轉換。

7.1.1 轉換的方向移動

移動動作及方向移動

已程式設計方向的移動動作主要由機台類型決定。對使用 TRAORI 之三、四及五軸類型的轉換，旋轉軸或於樞軸上的線性軸定義刀具的方向移動。

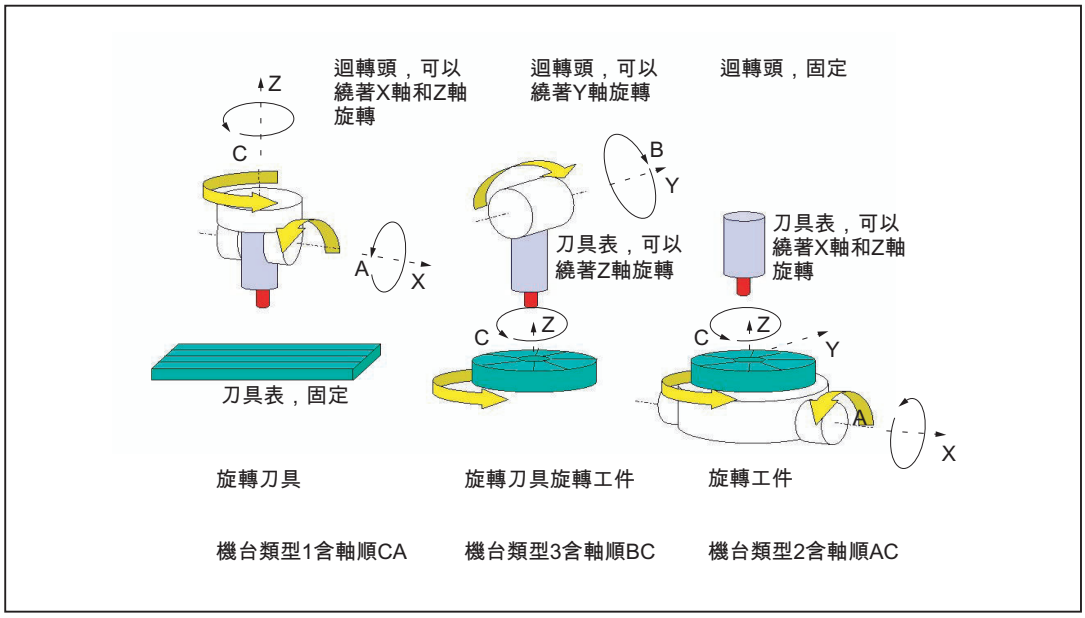
涉入方向轉換的旋轉軸位置中之變更，會導致剩餘機械軸的補正移動。刀具刀尖的位置維持不變。

可使用虛擬軸的旋轉軸辨識碼 A...、B...、C...，視情況應用程式設計刀具的方向移動，經由輸入 Euler 或 RPY 角度或方向，或表面法線向量、錐形旋轉軸或錐形柱面上中間方向的正規化向量。

使用 TRANSMIT、TRACYL 及 TRAANG 動態轉換的情況下，控制器會將編寫的直角座標系統移動動作，安排至實際機械軸的移動動作上。

三、四及五軸轉換之機械動態（TRAORI）

最多皆可使用兩個旋轉軸旋轉刀具及刀具表。亦可迴轉頭及旋轉表的組合（單軸情況下）。



機台類型	程式設計方向
三軸轉換機台類型 1 及 2	僅在與旋轉軸垂直的平面中，程式設計刀具方向。有 **兩個**平移軸（線性軸）及 **一個**旋轉的軸（旋轉軸）。
四軸轉換機台類型 1 及 2	僅在與旋轉軸垂直的平面中，程式設計刀具方向。有 **三個**平移軸（線性軸）及 **一個**旋轉的軸（旋轉軸）。
五軸轉換機台類型 3 單軸迴轉頭及單軸旋轉表	程式設計方向轉換。含 **三個** 線性軸及**兩個**直角旋轉軸的動力學。 旋轉軸與三個線性軸其中兩軸平行。兩由兩個直角線性軸移動第一個旋轉軸。它使用刀具旋轉第三個線性軸。第二個旋轉軸旋轉工件。

一般 5/6 軸轉換

機台類型	程式設計方向轉換
一般五 / 六軸轉換機台類型 4 含刀具的兩軸迴轉頭會自轉並繞單軸旋轉表旋轉	程式設計方向轉換。含 **三個**線性軸及**三個**直角旋轉軸的動力學。 旋轉軸與三個線性軸其中兩軸平行。兩由兩個直角線性軸移動第一個旋轉軸。它使用刀具旋轉第三個線性軸。第二個旋轉軸旋轉工件。可使用其他以 THETA 旋轉角度自轉的刀具旋轉，程式設計基本刀具方向。

呼叫「一般三、四及五 / 六軸轉換」時，亦可傳輸刀具的基本方向。旋轉軸方向的相關限制不再適用。若旋轉軸間不完全垂直，或現有旋轉軸與線性軸間不完全平行，「一般五 / 六軸轉換」能為刀具方向提供更好的結果。

動態轉換 TRANSMIT、TRACYL 及 TRAANG

對車削機台上的銑削，或研磨時可供傾斜進給設定之軸，應套用下列預設的符合已宣告之轉換軸配置：

TRANSMIT	極轉換之啟動
車削夾具中的平面加工	旋轉軸 與旋轉軸垂直的進給軸 與旋轉軸平行的縱向軸

TRACYL	圓柱表面轉換的啟動
在圓柱體以任何路徑溝槽加工。	旋轉軸 與旋轉軸垂直的進給軸 與旋轉軸平行的縱向軸

TRAANG	傾斜軸轉換的啟動
以傾斜進給軸來進行加工	旋轉軸 含可參數化角度的進給軸 與旋轉軸平行的縱向軸

直角座標 PTP 移動

機台於機械座標中移動且使用下列項目程式設計：

TRAORI	轉換啟動
PTP 點對點移動	於直角座標系統（MCS）中逼近位置
CP	BCS 中直角座標軸的路徑動作
STAT	關節接頭位置與轉換相關
TU	軸以最短路徑移動的角度

含一般 5/6 軸轉換之 PTP 橫向移動

機台使用機械座標及刀具方向移動，可使用圓軸位置及 Euler 及 / 或 RPY 角度向量程式設計該移動，不管動力學或方向向量。

在這些情況下，可於錐形柱面上，使用大圓弧插補或方向向量插補，進行圓軸插補、向量插補。

範例：通用銑削頭上的三至五軸轉換

機台工具至少有五個軸：

- 三個用來在直線裡移動的平移軸，將操作點移動至工作區的任何位置上。
- 以可設定角度（通常為 45 度）配置兩旋轉迴轉軸，使刀具迴轉至限制設定為 45 度的半圓位置空間中。

7.1.2 方向轉換概觀 TRAORI

可與 TRAORI 搭配的程式設計類型

機台類型	含有效轉換 TRAORI 的程式設計
機台類型 1、2 或 3 兩軸迴轉頭或兩軸旋轉表，或單軸迴轉頭與單軸旋轉表的組合。	可以使用依照機械動態的機械參數之 **特定機台**為基準， 或根據 **特定工件**，以獨立於機械動態的可程式設計之方向 設定方向軸的軸順序及刀具的軸方向。 參考系統中，以下列項目程式設計方向軸的旋轉方向： －ORIMKS 參考系統 = 機械座標系統 －ORIWKS 參考系統 = 工件座標系統 預設設定為 ORIWKS。 使用下列項目程式設計方向軸： 機械軸位置 A、B、C 管理 A2、B2、C2 角度程式設計虛擬軸 －使用 ORIEULER 透過尤拉角（標準） －ORIPY 透過 RPY 角 －ORIVIRT1 透過虛擬方向軸第一個定義 －ORIVIRT2 透過虛擬方向軸第二個定義， 包含插補類型間之差異： **線性插補** －方向軸 ORIAXES 或機械軸 **大半徑圓弧插補**（方向向量的插補） －從方向軸的 ORIVECT， 經由指定向量元件（方向 / 表面正常）A3、B3、C3 程式設計方向軸。 程式設計最終刀具方向。 在單節開始時，程式設計表面正常向量的 A4、B4、C4。 在單節結束時，程式設計單節垂直於平面的向量 A5、B5、C5。 LEAD 刀具方向的導角。 TILT 刀具方向的傾斜角度。

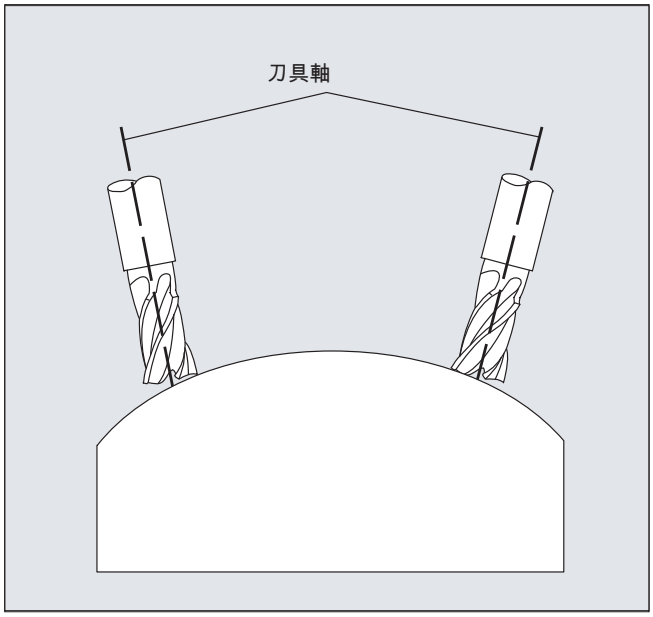
機台類型	含有效轉換 TRAORI 的程式設計
	錐形柱面上的方向向量插補。 使用插補，方向變更至空間中任何一處的 錐形柱面上： －平面中 ORIPLANE（大半徑圓弧插補） －錐形柱面順時針方向上的 ORICONCW －錐形柱面逆時針方向上的 ORICONCCW。 A6、B6、C6 取向向量（錐形旋轉軸） －OICONIO 於錐形柱面插補，使用： A7、B7、C7 中間向量（初始及最終方向）或 －錐形柱面上的 ORICONTO，切線變化 變更方向與含下列項目的路徑相關 －ORICURVE 的兩接點移動使用 PO[XH]=（xe，x2，x3，x4，x5），方向多項式最高達五次 PO[YH]=（ye，y2，y3，y4，y5）方向多項式最高達五次 PO[ZH]=（ze，z2，z3，z4，z5）方向多項式最高達五次 －ORIPATHS 使用 A8、B8、C8 刀具再定向階段， 方向特性的平滑化對應於：回退移動時的刀具方向及路徑長度

機台類型	含有效轉換 TRAORI 的程式設計
機台類型 1 及 3 其它另含刀具自轉的機台類型，需要第三旋轉軸 方向轉換，例如一般 6 軸轉換。方向向量旋轉。	使用 LEAD 角度**程式設計**刀具方向旋轉， 相對於表面正常向量的角度。 PO[PHI]程式設計多項式最高達五次； TILT 角度，繞路徑切線旋轉（Z 方向）； PO[PSI] 程式設計多項式最高達五次； THETA 旋轉角度（依 Z 繞刀具方向旋轉）； THETA = 到達單節結束之值； THETA=AC（...） 絕對非模態切換的至尺寸； THETA=IC（...）非模態切換至鏈結尺寸； THETA=Θ_e 插補已程式設計之角度 G90/G91； PO[THT]=（..） 程式設計多項式最高達五次。 程式設計旋轉向量 －ORIROTA 旋轉，絕對 －ORIROTR 相對旋轉向量 －ORIROTT 切線旋轉向量。
為有關路徑之方向變更所用的有關路徑之方向，或切線至路徑的旋轉向量之旋轉	在**相對於路徑**的方向中變更，使用 - ORIPATH 刀具方向相對於路徑 - ORIPATHS 亦於方向特性中的起伏 程式設計旋轉向量 - ORIROTC 切線旋轉向量，至路徑切線的旋轉

7.2 三、四及五軸轉換（TRAORI）

7.2.1 通用刀具頭的一般關係

當加工表面具有三維曲度時，為了取得最理想的切削條件，必須能改變刀具的設定角度。

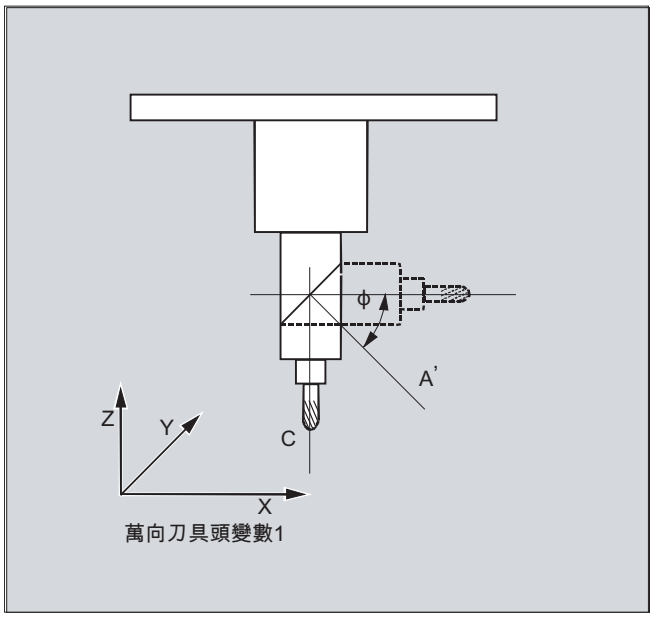


座標軸資料中儲存可達成本項目標的機台設計。

五軸轉換

萬向刀具頭

在此，由三線性軸（X、Y、Z）及兩方向軸（C、A）定義刀具的設定角度和操作點。 將兩方向軸之一設為傾斜軸，在我們的範例 A'中－在許多情況下，都是置於 45°角。

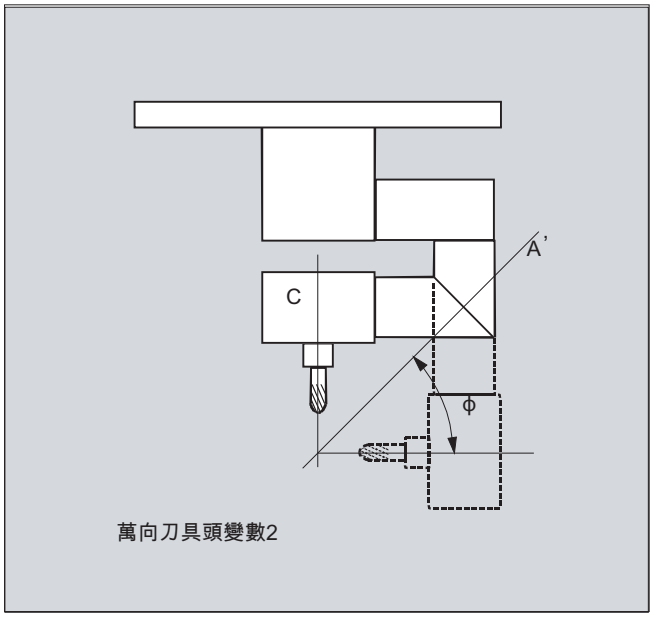


萬向刀具頭變數1

如此處範例中所示，您可看到圖示中以 CA 機台動力學配合萬向刀具頭的配置！

機台製造商

方向軸的軸順序及刀具的定位方向，可使用機械資料，視情況為機台動力學設定。



萬向刀具頭變數2

在本例中，A'位於與 X 軸成 φ 角度之下。

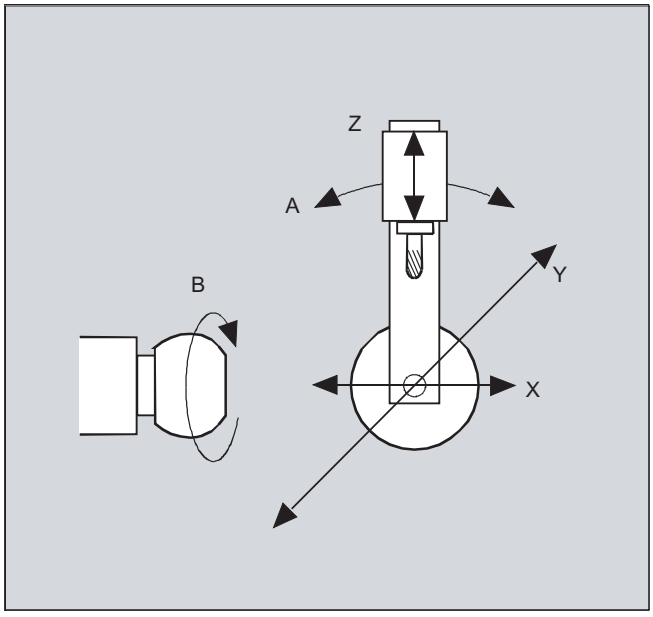
下列可能關係一般皆有效：

A'位於與 X 軸成 φ 角度之下	X 軸
B'位於與 Y 軸成 φ 角度之下	Y 軸
C'位於與 Z 軸成 φ 角度之下	Z 軸

φ 角度可使用機械資料設定為 0° 至 +89° 的範圍。

使用迴轉線性軸

此為具移動工件及移動刀具的配置。 其動力學包含三個線性軸（X、Y、Z）及兩個成直角配置的旋轉軸。 例如，第一個旋轉軸移過兩個線性軸的複式滑座，而刀具則與第三個線性軸平行。 第二個旋轉軸轉動工件。 第三個線性軸（迴轉軸）則位於複式滑座平面中。



旋轉軸的軸順序及刀具的定位方向，可使用機械資料，視情況為機台動力學設定。

有下列可能關係：

軸：	軸順序：
1. 旋轉軸	A A B B C C
2. 旋轉軸	B C A C A B
迴轉線性軸	Z Y Z X Y X

如需有關刀具定位方向可設定之軸順序的更多詳細資訊，請參閱

參考資料： /FB3/ 功能手冊，特殊函數；三至五軸轉換 (F2)，通用銑削頭章節「參數化」。

7.2.2 三、四及五軸轉換（TRAORI）

使用者可設定兩個或三個平移軸和一個旋轉軸。 轉換會假設旋轉軸與方向平面成直角。

僅在垂直於旋轉軸的平面中才有刀具方向。 轉換支援具可移動刀具和可移動工件的機台類型。

三及四軸轉換的設定，及程式設計的方式，與五軸轉換相同。

參考資料：
功能手冊、特殊功能；多軸轉換 (F2)

語法

```
TRAORI(<n>)

TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)

TRAFOOF
```

含義

TRAORI：	啟動第一個指定的方向轉換	
TRAORI(<n>)：	啟動由 n 指定的方向轉換	
<n>：	轉換數量	
	值：	1 或 2
	範例： TRAORI(1) 啟動方向轉換 1	
<X>、<Y>、<Z>：	刀具指向的方向向量元件	
<A>、<B>：	旋轉軸的可程式設計偏移	
TRAFOOF：	停用轉換運作	

刀具方向

依為刀具所選的定位方向而定，需在 NC 程式中設定有效工作平面（G17、G18、G19），讓刀長偏移量朝向刀具定位方向工作。

說明

啟用轉換時，位置資料（X、Y、Z）永遠與刀具刀尖相關。 改變牽涉轉換之旋轉軸的位置，會造成其餘機械軸發生補正運動，刀尖位置將維持不變。

方向轉換永遠從刀尖指向刀具接頭。

方向軸的偏移

啟動方向轉換後，可為方向軸直接程式設計其他偏移。

若程式設計時使用正確的順序，則可省略參數。

範例：

```
TRAORI(, , , ,A,B)  ；若只輸入單一的偏移。
```

作為直接程式設計的替代方案，方向軸的其他偏移，也可以從目前啟用的零點偏移量自動傳輸。 在機械資料中設定傳輸。

範例

```
TRAORI(1,0,0,1)          ；刀具的基本方向是 Z 方向
TRAORI(1,0,1,0)          ；刀具的基本方向是 Z 方向
TRAORI(1,0,1,1)          ；刀具的基本方向是 Y/Z 方向（符合於 -45° 位置）
```

7.2.3 方向程式設計及初始設定的變數（ORIRESET）

使用 TRAORI 之刀具方向的方向程式設計

結合可程式設計 TRAORI 方向轉換時，除線性軸 X、Y、Z 外，也可使用旋轉軸識別碼 A...、B...、C...，以角度或向量元件程式設計軸位置或虛擬軸。方向軸及機械軸可有各種類型的插補。無論目前啟用何種 PO[angle] 方向多項式及 PO[axis] 軸多項式，皆可程式設計許多不同類型的多項式。這些包含 G1、G2、G3、CIP 或 POLY。

在某些情況下，甚至可使用方向向量程式設計刀具方向中的變更。在此情況下，可透過直接程式設計向量，或是程式設計旋轉軸位置，來設定各單節的最終方向。

三至五軸轉換的方向程式設計變數

下列的方向程式設計版本相互排斥。

A、B、C	旋轉軸位置直接輸入內容。
A2、B2、C2	透過尤拉角度或 RPY 角度的虛擬軸之角度程式設計
A3、B3、C3	向量元件名稱
LEAD、TILT	參照至路徑及表面的導程及傾斜角度規格
A4、B4、C4 A5、B5、C5	在單節起點及單節結尾的表面法線向量

A6、B6、C6 A7、B7、C7	錐形柱面轉換上，方向向量的插補。
A8、B8、C8	刀具重新定向、退刀移動時的刀具方向及路徑長度

刀具方向的逼近初始設定 (ORIRESET)

透過 ORIRESET(...)，相關機台動態的方向軸從目前位置以線性同步移動至程式設計的初始狀態位置。若未程式設計軸的基本位置，則會使用相關的機械資料 $MC_TRAFO5_ROT_AX_OFFSET_1/2 中的位置。

忽略旋轉軸的啟用框架。

機械動態 CA（通道軸名稱 C、A）之範例

指令	說明
ORIRESET（90，45）	軸 C：90° 軸 A：45°
ORIRESET（，30）	軸 C：$MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A：30°
ORIRESET（ ）	軸 C：$MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A：$MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

機械動態 CAC（通道軸名稱 C、A、B）之範例

指令	說明
ORIRESET（90，45，90）	軸 C：90° 軸 A：45° 軸 B：90°
ORIRESET（ ）	軸 C：$MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A：$MC_TRAFO5_ROT_AX_OFFSET_1/2[1] 軸 B：$MC_TRAFO5_ROT_AX_OFFSET_1/2[2]

說明

使用 ORIRESET...）移動至刀具方向的初始狀態，只有在使用生效的方向轉換 TRAORI...）時才會發生作用。

程式設計 LEAD、TILT 及 THETA 旋轉

導程角度 LEAD 及傾斜角度 TILT。

關於三至五軸轉換，使用 LEAD 及 TILT 角度程式設計刀具方向的旋轉。

旋轉角度 THETA

轉換時**利用第三個旋轉軸**，可以用 THETA 旋轉角度程式設計刀具自轉，同時用在向量元件的方向及程式設計角度 LEAD、TILT 上。

7.2.4 編寫刀具方向 (A..., B..., C..., LEAD, TILT)

程式設計刀具方向時，可使用下列選項：

1. 直接程式設計旋轉軸動作。方向變更永遠發生在基本或機械座標系統中。並依照同步軸移動方向軸。
2. 尤拉或 RPY 角中的程式設計依照使用 A2、B2、C2 的角度定義
3. 使用 A3、B3、C3 程式設計方向向量，方向向量由刀具刀尖指向刀具接頭。
4. 用 A4、B4、C4 在單節開始程式設計表面法線向量，並用 A5、B5、C5 在單節結束程式設計（平面銑削）。
5. 使用螺距角度 LEAD 及傾斜角度 TILT 進行程式設計
6. 使用 A6、B6、C6 程式設計錐形旋轉軸為正規化向量，或使用 A7、B7、C7 程式設計錐形柱面上的中間方向
   請參閱「沿錐形柱面進行方向程式設計（ORIPLANE、ORICONxx）」。
7. 在回退移動期間，使用 A8、B8、C8
   ，程式設計重新定向、方向及刀具路徑長度請參閱「平滑化方向特性（ORIPATHS A8=、B8=、C8=）」

說明

在所有情況下，僅在啟用方向轉換時，才可程式設計方向。

優點：可傳輸該程式至任何機械動態。

透過 G 指令定義刀具方向

說明

機台製造商

可使用機械資料在尤拉角或 RPY 角間切換。若機械資料依此設定，則無論是否依照群組 50 的有效 G 指令，都可進行變換。您可選擇下列設定選項：

1. 若定義方向軸及方向角度的機械資料，皆透過 G 指令設定為零點：
   使用 A2、B2、C2 程式設計的角度皆**依照機械資料而定**；方向程式設計的角度定義則轉譯為尤拉角或 RPY 角。
2. 若透過 G 指令定義方向軸的機械資料設定為一，則依
   群組 50 的有效 G 指令來變換：
   使用 A2、B2、C2 程式設計的角度皆依照有效 G 指令 ORIEULER、ORIRPY、ORIVIRT1、ORIVIRT2、ORIAXPOS 及 ORIPY2 轉譯。使用方向軸程式設計的值，亦依照群組 50 的有效 G 指令，轉譯為方向軸。
3. 若定義方向角的機械資料設定為一，而透過 G 指令定義方向軸的機械資料是設定為零點，**則不依**群組 50 的有效 G 指令：
   使用 A2、B2、C2 程式設計的角度皆依照有效 G 指令 ORIEULER、ORIRPY、ORIVIRT1、ORIVIRT2、ORIAXPOS 及 ORIPY2 其中之一轉譯。使用方向軸程式設計的值則會轉譯為圓軸位置，與群組 50 的有效 G 指令無關。

語法

旋轉軸位置

```
G1 X<值> Y<值> Z<值> A<值> B<值> C<值>
```

尤拉角

```
G1 X<值> Y<值> Z<值> A2<值> B2<值> C2<值>
```

方向向量

```
G1 X<值> Y<值> Z<值> A3<值> B3<值> C3<值>
```

在單節起點的表面法線向量

```
G1 X<值> Y<值> Z<值> A4<值> B4<值> C4<值>
```

在單節結尾的表面法線向量

```
G1 X<值> Y<值> Z<值> A5<值> B5<值> C5<值>
```

螺距角度

```
LEAD=<值>
```

傾斜角度

```
TILT=<值>
```

意義

G1：	線性插補
X、Y、Z：	線性軸位置
A、B、C：	旋轉軸位置
A2=、B2=、C2=：	角度程式設計（尤拉角或 RPY 角）
A3=、B3=、C3=：	WCS 的 X、Y 與 Z 座標之方向向量。
A4=、B4=、C4=：	WCS 的 X、Y 與 Z 座標在單節起點的表面法線向量。
A5=、B5=、C5=：	WCS 的 X、Y 與 Z 座標在單節結尾的表面法線向量。
LEAD= ：	先導角度 1)
TILT=：	傾斜角度 1)
1) 角度表示的解譯係依據 MD21094 $MC_ORIPATH_MODE 中的設定	

其他資訊

五軸程式一般是由 CAD/CAM 系統產生，而非在控制系統中輸入。故下列說明，主要是針對後處理器的程式設計人員。

方向程式設計有以下指令：

指令	意義
ORIEULER：	尤拉角旋轉順序為 ZX'Z"
ORIRPY：	RPY 角旋轉順序為 XY'Z"
ORIRPY2：	RPY 角旋轉順序為 ZY'X"
ORIVIRT1：	虛擬方向軸使用可隨意定義的旋轉順序，如下： MD21120 $MC_ORIAX_TURN_TAB_1
ORIVIRT2：	虛擬方向軸使用可隨意定義的旋轉順序，如下： MD21130 $MC_ORIAX_TURN_TAB_2
ORIAXPOS：	虛擬方向軸搭配旋轉軸位置

說明

機台製造商可使用機械資料來定義各種變數。請參閱機台製造商說明。

以尤拉角 ORIEULER 程式設計，旋轉順序為 Z X' Z"

在使用 A2、B2、C2 方向程式設計 ORIEULER 期間，轉譯程式設計的值為尤拉角（以度為單位）。

新方向向量來自於下列原始方向向量的三次旋轉

1. 以旋轉軸 A2 繞著座標軸 Z
2. 以旋轉軸 B2 繞著新座標軸 X'
3. 以旋轉軸 C2 繞著座標軸 Z"

在此 C2 值的情況（繞新 Z 軸旋轉）毫無意義，且無需程式設計。

以 RPY 角 ORIRPY 程式設計，旋轉順序為 X Y' Z"

在使用 A2、B2、C2 方向程式設計 ORIEULER 期間，轉譯程式設計的值為 RPY 角（以度為單位），旋轉順序為 X Y' Z"。

說明

相較於 ORIEULER 程式設計，使用 ORIRPY 在此三個值皆會影響方向向量。

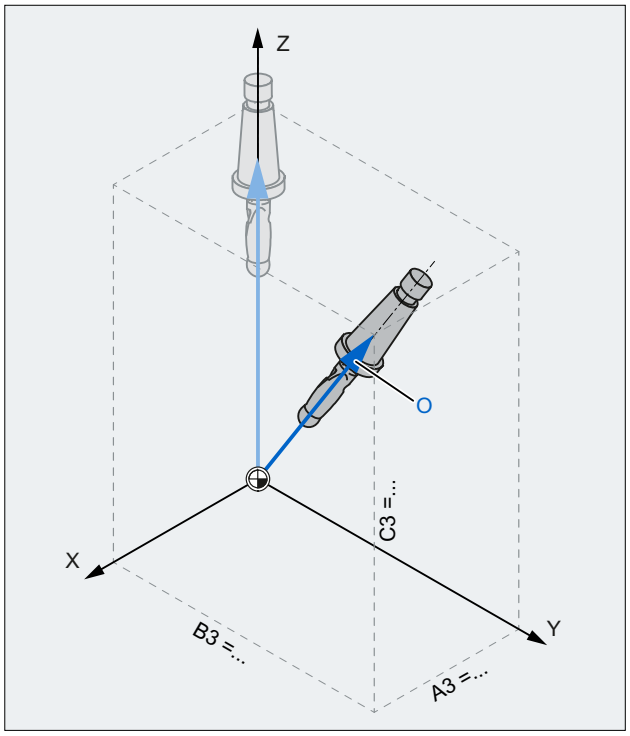
新方向向量來自於下列原始方向向量的三次旋轉

1. 以旋轉軸 A2 繞著座標軸 X
2. 以旋轉軸 B2 繞著新座標軸 Y'
3. 以旋轉軸 C2 繞著座標軸 Z"

程式設計方向性向量

以 A3、B3、C3 程式設計方向向量的元件。向量指向刀具轉接頭，向量的長度不重要。

未程式設計的向量元件，經設定等於零點。



X、Y、Z　WCS 的座標軸

A3、B3、C3　方向向量的分量

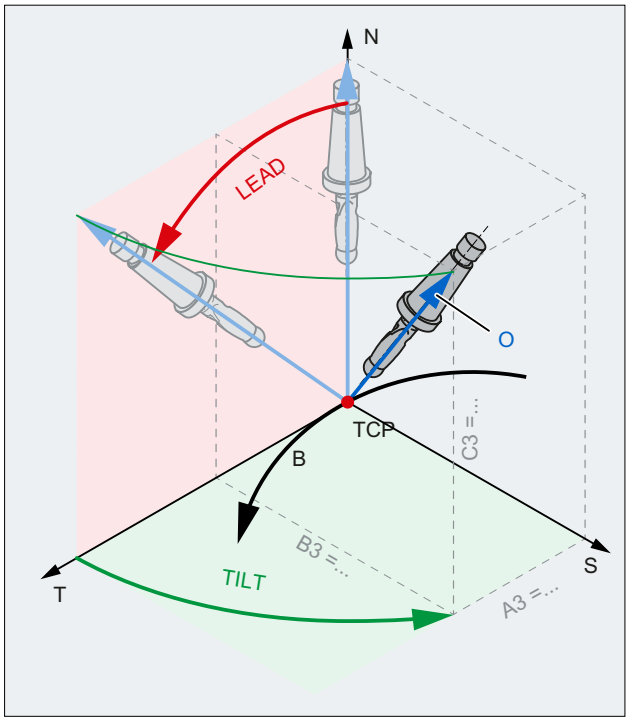
O　方向向量

圖像 7-1　程式設計方向向量

以 LEAD 及 TILT 程式設計刀具方向

由下列各項決定出刀具方向：

- 路徑切線
- 表面法線向量
  在單節 A4、B4、C4 的起點，及在單節 A5、B5、C5 的結尾
- 螺距角度 LEAD
  由路徑切線及表面法線向量，定義平面中的角度
- 傾斜角度 TILT 在單節的結尾
  平面中的角度，和相對於表面法線向量的路徑切線垂直



T 路徑切線

S 與路徑切線垂直

N 表面法向

B 路徑

TCP 刀具中心點

O 方向向量

圖像 7-2 LEAD TILT 的程式設計

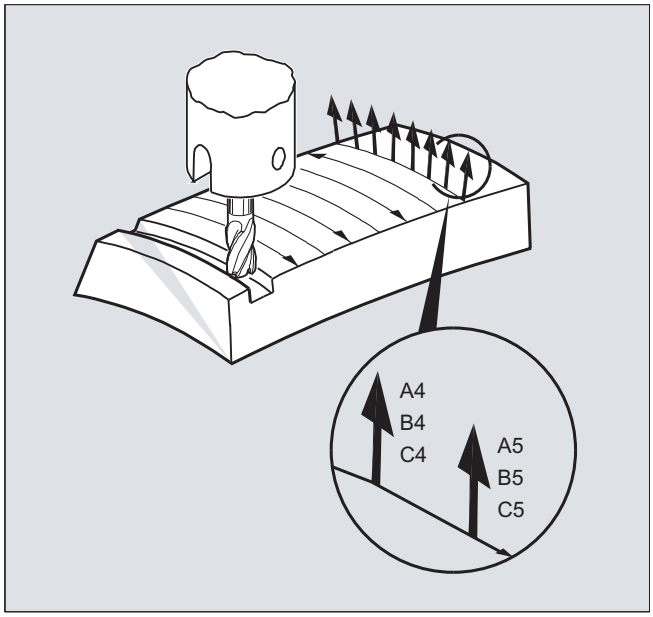
說明

使用 3D 刀具偏移量的內角行為

若在內角處縮短單節，則程式設計的刀具方向仍會在單節結尾處取代。

7.2.5 平面銑削 (A4, B4, C4, A5, B5, C5)

平面銑削用於所有加工曲面。



該類型 3D 銑削，在工件表面上需有 3D 路徑逐行說明。

計算需考慮刀具形狀及規格，其一般是在 CAM 中執行計算；然後再將計算完成的 NC 單節，透過後處理器讀入控制系統中。

程式設計路徑曲率

表面說明

以下列元件由表面法線向量說明路徑曲率：

A4、B4、C4 在單節起點的起始向量

A5、B5、C5 在單節結尾的結束向量

若單節僅含起始向量，則單節中所有表面法線向量將維持恆定。 若單節僅含結束向量，將透過大半徑圓弧插補，從前一單節的結束值到已編寫的結束值執行插補。

若起始及結束向量皆已編寫，則亦透過大半徑圓弧插補，在兩個方向間執行插補。 如此可持續平滑化產生路徑。

不管有效的 G17 到 G19 層級為何，在初始設定中，表面法線向量指向 Z 方向。

向量的長度毫無意義。

設定未已程式設計的向量元件為零點。

當 ORIWKS 啟用（請參閱 "方向軸參考（ORIWKS、ORIMKS）： (頁 369)"），表面法線向量參考現用框架，亦透過框架旋轉進行旋轉。

機台製造商

表面法線向量需需垂直於路徑切線，透過機械資料設定在臨界值內，否則將輸出警報。

7.2.6 方向軸參考（ORIWKS、ORIMKS）：

對於在工件座標系統中程式設計的方向時，當使用

- 尤拉角或 RPY 角，或
- 方向向量

可使用 ORIMKS/ORIWKS 來設定旋轉動作的課程。

說明

機台製造商

以機械資料指定的方向之插補類型：

MD21104 $MC_ORI_IPO_WITH_G_CODE

= FALSE：參考點由 G 指令 ORIWKS 與 ORIMKS 提供。

= TRUE：參考點是第 51 群組的 G 指令 (ORIAXES, ORIVECT ,ORIPLANE, ...)

語法

```
ORIMKS=...

ORIWKS=...
```

含義

ORIMKS：	在機械座標系統中旋轉
ORIWKS：	在工件座標系統中旋轉

說明

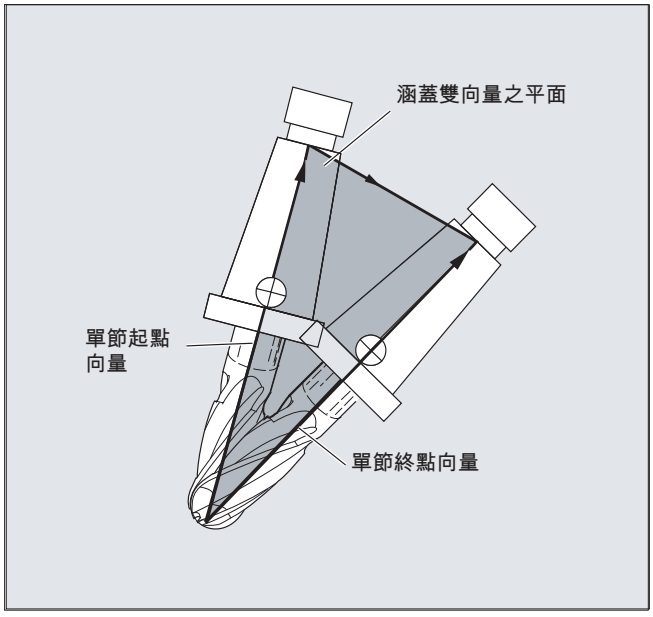
ORIWKS 為基本設定。在五軸程式的情況下，若欲執行的機台未立即明確，則必需選擇 ORIWKS。機台實際執行的動作依照機械動態而定。

ORIMKS 可被用於程式設計實際的機台移動（用來避免和設備碰撞或類似情形，例如）。

其他資訊

使用 ORIMKS 時，刀具執行動作**依照**機械動態。 在空間中固定點刀具刀尖方向變更的情況下，會在旋轉軸位置間進行線性插補。

使用 ORIWKS 時，刀具執行動作**不依照**機械動態。 含固定刀具刀尖的方向變更，在依起始及結束向量設定的平面中移動刀具。



單一位置

說明

ORIWKS

五軸機台單一設定區中的方向動作需要大量機械軸動作（例如，以 C 為旋轉軸，而以 A 為迴轉軸的旋轉迴轉頭，具 A = 0 的所有位置皆為單一位置）。

機台製造商

速率控制會大幅降低接近單一位置的刀具路徑速率，以避免機械軸過載。

使用以下機械資料時

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

可參數化轉換，讓接近極點的方向動作穿透極點，且能迅速加工。

只能使用 MD $MC_TRAFO5_POLE_LIMIT 處理單一位置。

參考資料：
/FB3/ 功能手冊，特殊函數：三至五軸轉換（F2），
“單一點及其處理方式”一節。

7.2.7 程式設計方向軸（ORIAXES，ORIVECT，ORIEULER，ORIRPY，ORIRPY2，ORIVIRT1，ORIVIRT2）

「方向軸」函數定義刀具在空間中的方向，並藉由旋轉軸偏移的程式設計來達成。 也可讓刀具本身旋轉，而達成額外的第三度自由度。 在此情況下，透過第三個旋轉軸在空間定向刀具，該動作需六軸轉換。刀具繞自身旋轉是依據旋轉向量的插補類型，使用 THETA 旋轉角度來定義。（請參閱 "刀具方向的旋轉（ORIROTA、ORIROTR、ORIROTT、ORIROTC、THETA） (頁 382)"）。

使用軸識別碼 A2、B2 及 C2 程式設計方向軸。

語法

```
N... ORIAXES/ORIVECT                ；線性或大半徑圓弧插補
N... G1 X Y Z A B C

N... ORIPLANE                       ；平面的方向插補

N... ORIEULER/ORIRPY/ORIRPY2        ： 方向角度尤拉 / RPY 角
N... G1 X Y Z A2= B2= C2=           ；虛擬軸的角度程式設計

N... ORIVIRT1/ORIVIRT2              ；虛擬方向軸定義 1/2
N... G1 X Y Z A3= B3= C3=           ；方向向量程式設計
```

說明

可沿空間中錐形柱面，為方向變更編寫方向軸的其他旋轉軸偏移，請參閱 "沿著錐形的柱面進行方向程式設計（ORIPLANE，ORICONCW，ORICONCCW，ORICONTO，ORICONIO） (頁 374)"。

含義

ORIAXES：	機台或方向軸的線性插補
ORIVECT：	大半徑圓弧插補（與 ORIPLANE 相同）

ORIMKS： ORIWKS：	在機械座標系統中旋轉 在工件座標系統中旋轉 相關說明請參閱 "方向軸參考（ORIWKS、ORIMKS）：(頁 369)"。
A= B= C=：	程式設計機械軸位置
ORIEULER：	透過尤拉角的方向程式設計
ORIRPY：	透過 RPY 角的方向程式設計 旋轉順序為 XYZ，且： • A2 是繞 X 的旋轉角度 • B2 是繞 Y 的旋轉角度 • C2 是繞 Z 的旋轉角度
ORIRPY2：	透過 RPY 角的方向程式設計 旋轉順序為 ZYX，且： • A2 是繞 Z 的旋轉角度 • B2 是繞 Y 的旋轉角度 • C2 是繞 X 的旋轉角度
A2= B2= C2=：	虛擬軸的角度程式設計
ORIVIRT1/ORIVIRT2：	使用虛擬方向軸的方向程式設計 定義 1： 依據 MD21120 $MC_ORIAX_TURN_TAB_1 定義 定義 2： 依據 MD21130 $MC_ORIAX_TURN_TAB_2 定義
A3= B3= C3=：	方向軸的方向向量程式設計

其他資訊

機台製造商

使用 MD21102 $MC_ORI_DEF_WITH_G_CODE 指定已編寫的角度 A2、B2、C2 的定義方式：

根據 MD21100 $MC_ORIENTATION_IS_EULER（標準），或根據 G 群組 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2) 來定義。

MD21104 $MC_ORI_IPO_WITH_G_CODE 定義有效的插補模式類型：ORIWKS/ORIMKS 或 ORIAXES/ORIVECT。

寸動進給（JOG）模式

在本操作模式中的方向角度插補永遠為線性。在透過移動鍵持續及增量移動期間，僅可移動一個方向軸。可同時使用手輪移動方向軸。

當方向軸是以手動移動時，套用以快速移動超調的通道專屬進給率超調開關或快速移動超調開關。

可用下列機械資料進行獨立的速率設定：

MD21160 $MC_JOG_VELO_RAPID_GEO

MD21165 $MC_JOG_VELO_GEO

MD21150 $MC_JOG_VELO_RAPID_ORI

MD21155 $MC_JOG_VELO_ORI

說明

SINUMERIK 840D sl 與「處理轉換套件」

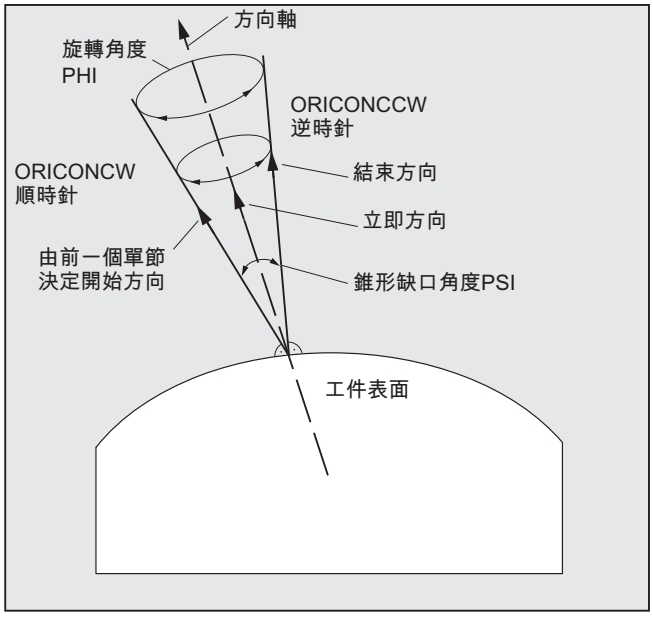
使用「直角座標手動移動」功能，可以從參考系統－機械座標系統（MCS）、工件座標系統（WCS）及 TCS 中，互相設定在寸動進給模式中的幾何軸轉譯。

參考資料：

功能手冊，延伸功能，動態轉換 (M1)

7.2.8 沿著錐形的柱面進行方向程式設計（ORIPLANE，ORICONCW，ORICONCCW，ORICONTO，ORICONIO）

使用延伸方向，可沿空間中的錐形柱面執行方向變更。 使用 ORICONxx 模態指令，在錐形柱面上插補方向向量。 可在平面上，使用插補的 ORIPLANE 程式設計結束方向。 通常由前一個單節定義起始方向。



程式設計

使用 A2、B2、C2，在尤拉角或 RPY 角中指定角度程式設計，或使用 A、B、C 程式設計旋轉軸位置來定義結束方向。沿錐形柱面的方向軸需要進一步的程式設計細節：

- 將錐形旋轉軸做為使用 A6、B6、C6 的向量
- 使用識別碼 NUT 之缺口角度 PSI
- 使用 A7、B7、C7 之中間方向錐形外部

說明

為錐形的旋轉軸程式設計方向向量 A6、B6、C6

結束方向的程式設計絕非必要。 若無指定結束方向，則會插補 360 度的完整外錐形。

用 NUT=angle 程式設計錐形缺口角度

需指定結束方向。

無法以此方式插補 360 度的完整外錐形。

在錐形外部程式設計中間方向 A7、B7、C7

需指定結束方向。 僅可透過三個向量：「起始」方向、「結束」方向及「中間」方向，各自定義變更方向及旋轉方向。 三個向量需為不同。 若已編寫的中間方向與起始或結束方向平行，便在以起始及結束向量定義的平面中，執行方向的線性大半徑圓弧插補。

錐形柱面上的延伸方向插補

N... ORICONCW 或 ORICONCCW	在錐形外部的**插補**，使用
N... A6= B6= C6= A3= B3= C3=	在錐形順時針 / 逆時針方向和結束方向的方向向量，或
或	
N... ORICONTO	結束方向的
N... G1 X Y Z A6= B6= C6=	切線變化及規格
或	或
N... ORICONIO	錐形外部的結束方向及
N... G1 X Y Z A7= B7= C7=	中間方向規格，使用
N... PO[PHI]=（a2，a3，a4，a5）	旋轉角度的多項式及
N... PO[PSI]=（b2，b3，b4，b5）	缺口角度的多項式

參數

ORIPLANE：	平面中插補（大半徑圓弧插補）
ORICONCW：	順時針方向的錐形柱面插補

ORICONCCW：	逆時針方向的錐形柱面插補
ORICONTO：	含切線變化的錐形柱面插補
A6= B6= C6=：	錐形旋轉軸的程式設計（正規化向量）
NUT=angle：	錐形的缺口角度，以度為單位
NUT=+179：	移動角度小於或等於 180 度
NUT=-181：	移動角度大於或等於 180 度
ORICONIO：	錐形柱面上的插補
A7= B7= C7=：	中間方向（程式設計為正規化向量）
PHI：	繞錐形方向軸的方向旋轉角度
PSI：	錐形缺口角度
Possible polynomials PO[PHI]=（a2，a3，a4，a5） PO[PSI]=（b2，b3，b4，b5）	除不同角度外，也可程式設計多項式最高達 五次

範例： 不同的方向變更

程式碼	註解
…	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	；方向轉換 ON
N30 ORIVECT	；插補刀具方向為向量。
…	；平面中刀具方向。
N40 ORIPLANE	；選擇大半徑圓弧插補。
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	；在 Y/Z 平面的方向旋轉 45 度，在單節結束時方向變為 (0,1/√2,1/√2)。
…	
N70 ORICONCW	；在錐形外部進行方向程式設計：
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	方向向量是以從 (0,0,1) 直到 (1/√2,0,1/√2) 的順時針方向插補在錐形的外部，旋轉角度為 270 度。
N90 A6=0 B6=0 C6=1	；刀具方向在相同的錐形外部通過完整的轉數。

其他資訊

若要在空間任何一處，沿錐形柱面說明變更方向，需先知道欲旋轉刀具方向的向量。亦需指定起始及結束方向。從前一個單節產生起始方向，並透過其他條件程式設計或定義結束方向。

在與 ORIVECT 相同的 ORIPLANE 平面中程式設計

大半徑圓弧插補的程式設計，與角度多項式一同相當於輪廓的線性及多項式插補。在由起始及結束方向定義的平面中插補刀具方向。若程式設計了其他多項式，則方向向量也可傾斜至平面外。

在平面 G2/G3、CIP 及 CT 中圓弧的程式設計

與平面中圓弧插補相符合的延伸方向。關於具中心或半徑（如 G2/G3）的圓弧、經過中間點 CIP 的圓弧、及切線圓弧 CT 的對應程式設計選項，請參閱

參考資料：程式設計手冊基礎，「程式設計動作指令」。

方向程式設計

錐形柱面上，方向向量的插補 ORICONxx

可從 G 指令群組 51 選擇四種不同類型插補，在錐形柱面上插補方向：

1. 以結束方向及錐形方向或缺口角度的規格，依順時針方向在錐形外部插補 ORICONCW。以識別碼 A6、B6、C6 程式設計方向向量，並以識別碼 NUT=值範圍，間隔 0 度到 180 度，程式設計錐形缺口角度。
2. 以結束方向及錐形方向或缺口角度的規格，依逆時針方向在錐形外部插補 ORICONCCW。以識別碼 A6、B6、C6 程式設計方向向量，並以識別碼 NUT=值範圍，間隔 0 度到 180 度，程式設計錐形缺口角度。
3. 以識別碼 A7、B7、C7 程式設計的結束及中間方向規格，在錐形外部插補 ORICONIO。
4. 以切線變化及結束方向的規格，在錐形外部插補 ORICONTO。以識別碼 A6、B6、C6 程式設計方向向量。

7.2.9 兩接點的方向規格（ORICURVE、PO[XH]=、PO[YH]=、PO[ZH]=）

使用空間中第二個曲度程式設計方向變更 ORICURVE

除使用刀尖沿空間中曲度外，另一個程式設計方向變更的方式，為使用 ORICURVE 程式設計第二個接點的動作。在此方式中，僅可定義刀具方向變更，與程式設計刀具本身向量時一樣。

機台製造商

關於可以透過機械資料，設定為程式設計刀具第二個方向路徑的軸識別碼，請參閱機台製造商的說明。

程式設計

可使用該類型插補程式設計空間中兩曲度的點（使用 G1）或多項式（使用 POLY）。不得使用圓弧與漸開線。也可啟動 BSPLINE 曲線插補及「結合短曲線單節」函數。

參考資料：
功能手冊，基本功能，連續路徑模式，精確停止，預見控制 (B1)，章節：結合短曲線單節

不得使用其他曲線類型 ASPLINE 及 CSPLINE，以及使用 COMPON、COMPCURV 或 COMPCAD 啟動壓縮機。

程式設計座標的方向多項式時，可預先定義刀具的兩接點動作最高達五次。

具空間中其他曲度及座標多項式的延伸方向插補

```
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)
```

刀具的第二個接點動作規格，及該座標的其他多項式

參數

ORICURVE	指定刀具兩接點間動作的方向插補。
XH YH ZH	做為空間中曲度之其他輪廓的第二個刀具接點之座標識別碼
Possible polynomials PO[XH]=（xe，x2，x3，x4，x5） PO[YH]=（ye，y2，y3，y4，y5） PO[ZH]=（ze，z2，z3，z4，z5）	除使用適當的終點外，也可使用多項式程式設計空間中曲度。
xe，ye，ze	空間中曲度的終點
xi，yi，zi	多項式的係數最高達五次

說明

程式設計第二個方向路徑的識別碼 XH YH ZH

識別碼的需經選擇，才不會與其他識別碼或線性軸產生衝突

X Y Z 軸

及旋轉軸，例如

A2 B2 C2 尤拉角或 RPY 角

A3 B3 C3 方向向量

A4 B4 C4 或 A5 B5 C5 表面法線向量

A6 B6 C6 旋轉向量或 A7 B7 C7 中間點座標

或其他插補參數。

7.3 方向多項式（PO[角度]、PO[座標]）

與目前啟用的 G 群組 1 的多項式插補無關，可為三軸或至五軸轉換程式設計兩個不同類型的方向多項式最高達五次。

1. **角度**的多項式：螺距角度 LEAD、傾斜角度 TILT
   與以起始及結束方向所定義平面的關係。
2. **座標**的多項式：刀具上參考點的刀具方向之空間中第二個曲度的 XH、YH、ZH。

有了六軸轉換，除刀具方向外，可為刀具本身旋轉，以最高達五次的多項式，程式設計旋轉向量 THT 之旋轉。

語法

類型 1 **角度**的方向多項式

```
N... PO[PHI]=(a2，a3，a4，a5)        三軸至五軸轉換
N... PO[PSI]=(b2，b3，b4，b5)
```

類型 2 **座標**的方向多項式

```
N... PO[XH]=(xe，x2，x3，x4，x5)   刀具方向之第二個方向路徑的座標識別碼
N... PO[YH]=(ye，y2，y3，y4，y5)
N... PO[ZH]=(ze，z2，z3，z4，z5)
```

在這兩種情況下，有了六軸轉換，也可為**旋轉**使用方向向量程式設計一多項式

```
N... PO[THT]=(c2，c3，c4，c5)        相對於路徑的旋轉插補
```

或

```
N... PO[THT]=(d2，d3，d4，d5)        方向變更的絕對、相對及切線插補
```

方向向量。若轉換支援含可程式設計偏移的旋轉向量，並使用 THETA 旋轉角度進行插補，即可進行上述作業。

含義

PO[PHI]	平面中起始與結束方向間的角度
PO[PSI]	定義由平面起始與結束方向間方向角度的傾斜
PO[THT]	旋轉的角度，是由旋轉群組 54 的 G 指令之一的旋轉向量而產生，而該指令是用 THETA 所程式設計
PHI	螺距角度 LEAD
PSI	傾斜角度 TILT
THETA	依 Z 繞刀具方向的旋轉
PO[XH]	刀具上參考點的 X 座標
PO[YH]	刀具上參考點的 Y 座標
PO[ZH]	刀具上參考點的 Z 座標

其他資訊

以下情況無法程式設計方向多項式：

- 若已啟用 ASPLINE、BSPLINE、CSPLINE 曲線插補。
  類型 1 方向角度的多項式可用於曲線外所有類型的插補，即，含快速移動 G00 或進給率 G01 的線性插補
  與使用 POLY 及
  圓弧 / 漸開線插補 G02、G03、CIP、CT、INVCW 及 INCCCW 的多項式插補
  。
  然而，類型 2 方向座標的多項式僅在若
  含快速移動 G00 或進給率 G01 的線性插補，或
  已啟用 POLY 的多項式插補下使用。
- 若使用 ORIAXES 軸插補來插補方向。 在此情況下，可直接使用方向軸 A 及 B 的 PO[A] 及 PO[B]程式設計多項式。

含 ORIVECT、ORIPLANE 及 ORICONxx 類型 1 方向多項式

僅類型 1 方向多項式可用於大半徑圓弧插補，及以 ORIVECT、ORIPLANE 及 ORICONxx 在錐形外部插補。

含 ORICURVE 類型 2 方向多項式

若啟用具空間中其他曲度的插補 ORICURVE，則會插補方向向量的直角座標元件，且僅能使用類型 2 方向多項式。

7.4 刀具方向的旋轉（ORIROTA、ORIROTR、ORIROTT、ORIROTC、THETA）

若您亦欲在具有可移動刀具的機台類型上變更刀具方向，請程式設計含結束方向的每個單節。 依照機械動態，您可程式設計方向軸的定位方向，或方向向量的旋轉方向 THETA。可為下列旋轉向量程式設計不同插補類型：

- ORIROTA： 至絕對旋轉方向的旋轉角度。
- ORIROTR： 相對於平面起始與結束方向間的旋轉角度。
- ORIROTT： 相對於方向向量中變更的旋轉角度。
- ORIROTC： 至路徑切線的旋轉切線角度。

語法

只有啟用插補類型 ORIROTA 時，才可在以下四個模式中設定旋轉角度或旋轉向量：

1. 直接做為旋轉軸位置 A、B、C
2. 使用 A2、B2、C2 之尤拉角（以度為單位）
3. 使用 A2、B2、C2 之 RPY 角（以度為單位）
4. 透過 A3、B3、C3 的方向向量（使用 THETA=<值> 的旋轉角度）

若已啟用 ORIOTR 或 ORIOTT，僅能直接使用 THETA 設定旋轉角度。

旋轉也可在未發生方向變更的獨立單節中，進行程式設計。 在此情況下，ORIROTR 及 ORIROTT 不相關。 在此情況下，皆參考絕對方向解讀旋轉角度 (ORIROTA)。

N... ORIROTA	定義旋轉向量的插補
N... ORIROTR	
N... ORIROTT	
N... ORIROTC	
N... A3= B3= C3= THETA=<值>	定義方向向量的旋轉
N... PO[THT]=（d_2，d_3，d_4，d_5）	使用五次多項式的旋轉插補角度

含義

ORIROTA：	絕對旋轉方向的旋轉角度
ORIROTR：	相對於起始角度及結束角度間之平面的旋轉角度

ORIROTT：	做為方向變更之切線旋轉向量的旋轉角度
ORIROTC：	做為路徑切線之切線旋轉向量的旋轉角度
THETA：	方向向量的旋轉
THETA=<值>：	單節結尾達到的旋轉角度度數
THETA=Θe：	含旋轉向量結束角度 Θ_e 的旋轉角度
THETA=AC(...)：	絕對尺寸的非模態切換
THETA=AC(...)：	增量尺寸的非模態切換
Θe：	啟用含 G90 絕對及 G91 相對（增量尺寸標註）旋轉向量的結束角度
PO[THT]=(....)：	旋轉角度的多項式

範例： 方向旋轉

程式碼	註解
N10 TRAORI	；啟動方向轉換
N20 G1 X0 Y0 Z0 F5000	；刀具方向
N30 A3=0 B3=0 C3=1 THETA=0	；以旋轉角度 0，依 Z 方向旋轉
N40 A3=1 B3=0 C3=0 THETA=90	；依 X 方向，並旋轉約 90 度
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	；方向
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	；依 Y 方向，並旋轉約 180 度
N70 ORIROTT	；維持恆定，並旋轉至 90 度
N80 A3=1 B3=0 C3=0 THETA=30	；相對於方向變更的旋轉角度
	；以 30 度旋轉向量至 X/Y 平面

插補單節 N40 時，以線性插補由初始值 0 度至最終值 90 度的旋轉角度。 在單節 N50 中，根據拋物線 θ(u) = +90u2，將旋轉角度由 90 度變更為 180 度。 在 N60 中，也可在未發生方向變更下執行旋轉。

在 N80 中，由 Y 方向朝 X 方向旋轉刀具方向。 在 X/Y 平面中進行方向變更，而旋轉向量定義與此平面成 30 度之角度。

其他資訊

ORIROTA

使用空間中絕對方向的參考，插補旋轉角度 THETA。 基本旋轉方向在機械資料中定義。

ORIROTR

相對於由起始及結束方向定義的平面，轉譯旋轉角度 THETA。

ORIROTT

相對於方向變更，轉譯旋轉角度 THETA。 當 THETA=0 時，會切線插補旋轉向量至方向變更，且僅在若為方向的“傾斜角度 PSI”至少程式設計一個多項式時，才與 ORIROTR 不同。結果為在平面中並未執行的方向變更。 然後可使用旋轉 THETA 的其他角度來插補旋轉向量，如此將產生參考方向變更的特定值。

ORIROTC

旋轉向量係以用 THETA 角度程式設計的偏移，插補到相對的路徑。 多項式 PO[THT]=（c2，c3，c4，c5）也可為偏移角度，程式設計最高達五次。

7.5 相對於路徑的方向

7.5.1 相對於路徑的方向類型

使用此展開函數，相對方向不僅在單節結尾處，且會跨越整個軌道而達成。使用大半徑圓弧插補，將前一個單節中達成的方向傳送至編寫的結束方向。基本上，有兩種方式可以程式設計相對於路徑所要方向：

1. 如同刀具旋轉，使用 ORIPATH、ORPATHTS 插補相對於路徑的刀具方向。
2. 以一般方式程式設計及插補方向向量。使用 ORIROTC，開始相對於路徑切線的方向向量旋轉。

語法

使用用下列語法程式設計方向的插補類型及刀具的旋轉：

N...ORIPATH	相對於路徑的方向
N...ORIPATHS	相對於具方向特性的平滑化路徑之方向
N...ORIROTC	相對於路徑之旋轉向量的插補

可使用 ORIPATHS 來平滑化在軌道上由轉角造成的方向起伏。使用元件 A8=X、B8=Y、C8=Z，透過向量程式設計回退移動的方向及路徑長度。

可使用 ORIPATH/ORIPATHS 透過三個角度，為整個軌道程式設計各種路徑切線的參考

- LEAD=相對於路徑及表面的螺距角度規格
- TILT=相對於路徑及表面的傾斜角度規格
- THETA=旋轉角度

用於整個軌道。除使用旋轉角度 THETA 外，也可使用 PO[THT]=（...）程式設計最高達五次的多項式。

說明

機台製造商

請參閱機台製造商說明。透過可設定的機械及設定資料，產生其他相對於路徑的方向設定。如需更多詳細資訊，請參考

參考資料：
/FB3/ 功能手冊，特殊函數，三至五軸轉換 (F2)，
〈方向〉章節 "Orientation"

含義

可透過機械資料產生角度 LEAD 及 TILT 插補的各種設定：

- 使用 LEAD 及 TILT 將程式設計的刀具方向參考保留在整個單節中。
- 螺距角度 LEAD：繞垂直於切線及法線向量 TILT 的方向旋轉：繞法線向量的方向旋轉。
- 螺距角度 LEAD：繞垂直於切線及法線向量「傾斜」角度 TILT 的方向旋轉：依路徑切線方向的方向旋轉。
- 旋轉角度 THETA：以另外第三個旋轉軸做為六軸轉換中的方向軸，繞刀具本身旋轉。

說明

相對於路徑的方向不得搭配 OSC、OSS、OSSE、OSD 及 OST

不得搭配以群組 34 的 G 指令方向特性平滑化，程式設計相對於路徑的方向插補，即 ORIPATH 或 ORIPATHS 及 ORIOTC。此動作需啟用 OSOF。

7.5.2 相對於路徑的刀具方向旋轉（ORIPATH、ORIPATHS、旋轉角度）

使用六軸轉換時，以第三個旋轉軸在空間中，依所需的定向刀具，可使刀具繞本身旋轉。使用 ORIPATH 或 ORIPATHS 相對於路徑的刀具方向旋轉時，可透過 THETA 旋轉角度程式設計其他的旋轉。 或者可使用位於與刀具方向垂直的平面向量，程式設計 LEAD 及 TILT 角度。

機台製造商

請參閱機台製造商說明。 可使用機械資料將 LEAD 及 TILT 角度的插補設定為不同插補。

語法

刀具方向及刀具的旋轉

使用 ORIPATH 或 ORIPATHS，啟動相對於路徑的刀具方向類型。

N... ORIPATH	啟動相對於路徑的方向類型
N... ORIPATHS	啟動含方向特性的平滑化，相對於路徑的方向類型

啟動可旋轉的三個角度：

N... LEAD=	相對於表面法線向量，已程式設計的方向角度

N... TILT=	與相對於表面法線向量的路徑切線垂直，且在平面中已程式設計的方向角度
N... THETA=	在第三個旋轉軸的刀具方向中，相對於方向變更的旋轉角度

在單節結尾使用 LEAD=value、TILT=value 或 THETA=value 程式設計角度值。 除恆定角外，皆可為三個角度程式設計多項式最高達五次。

N... PO[PHI]=(a2，a3，a4，a5)	螺距角度 LEAD 的多項式
N... PO[PSI]=(b2，b3，b4，b5)	傾斜角度 TILT 的多項式
N... PO[THT]=(d2，d3，d4，d5)	旋轉角度 THETA 的多項式

較高的多項式係數為零點時，則可在程式設計時省略。 範例： PO[PHI]=a2 會產生 LEAD 角度的拋物線。

含義

相對於路徑的刀具方向

ORIPATH：	與路徑相關的刀具方向
ORIPATHS：	刀具相對於路徑之方向，方向特性中的上下變動經平滑化化
LEAD：	由路徑切線及表面法線向量定義的平面中，相對於表面法線向量的角度
TILT：	依 Z 方向的方向，或繞路徑切線旋轉
THETA：	朝 Z 繞刀具方向旋轉
PO[PHI]：	LEAD 角度的方向多項式
PO[PSI]：	TILT 角度的方向多項式
PO[THT]：	THETA 旋轉角度的方向多項式

說明

旋轉角度 THETA

需有六軸轉換，才能用以繞自身方向軸的第三個旋轉軸，來旋轉刀具。

7.5.3 相對於路徑的刀具旋轉插補（ORIROTC、THETA）

使用旋轉向量的插補

以 ORIROTC 程式設計刀具旋轉的旋轉向量，相對於路徑切線，亦可以偏移插補，而此偏移可以使用 THETA 旋轉角度來進行程式設計。 因此，可使用 PO[THT]，為偏移角度程式設計最高達五次的多項式。

語法

```
N... ORIROTC                                    啟動相對於路徑切線的刀具旋轉
N... A3= B3= C3= THETA=value                    定義方向向量的旋轉
N... A3= B3= C3= PO[THT]=(c2，c3，c4，c5)       以最高達五度的多項式插補偏移角度
```

旋轉也可在未發生方向變更的獨立單節中，進行程式設計。

含義

六軸轉換中，相對於路徑的刀具旋轉插補

ORIROTC：	啟動相對於路徑切線的切線旋轉向量
THETA=value：	單節結尾達到的旋轉角度度數
THETA=θe：	含旋轉向量結束角度 Θ_e 的旋轉角度
THETA=AC(...)：	絕對尺寸的非模態切換
THETA=IC(…)：	增量尺寸的非模態切換
PO[THT]=(c2, c3, c4, c5)：	以五度多項式插補偏移角度

說明

旋轉向量的插補 ORIROTC

僅能以六軸轉換，啟動與刀具方向相反方向中，相對於路徑切線的刀具旋轉。

使用有效 ORIROTC

無法程式設計旋轉向量 ORIROTA。 若正進行程式設計，便會輸出「警報」14128「使用有效 ORIROTC 的刀具旋轉絕對程式設計」。

三軸至五軸轉換的刀具定位方向

與三軸至五軸轉換相同，可透過尤拉角、RPY 角或方向向量，程式設計刀具的定位方向。也可透過程式設計達成空間中的刀具方向變更，程式設計指令包括大半徑圓弧插補 ORIVECT、方向軸的線性插補 ORIAXES、錐形柱面上所有插補 ORICONxx，及空間中含刀具兩接點之曲度外的插補 ORICURVE。

G....：	旋轉軸動作的細節
X、Y、Z：	線性軸的細節
ORIAXES：	機台或方向軸的線性插補
ORIVECT：	大半徑圓弧插補（與 ORIPLANE 相同）
ORIMKS： ORIWKS：	在機械座標系統中旋轉 在工件座標系統中旋轉 說明，請參閱「刀具方向的旋轉」一節
A= B= C=：	程式設計機械軸位置
ORIEULER：	透過尤拉角的方向程式設計
ORIRPY：	透過 RPY 角的方向程式設計
A2= B2= C2=：	虛擬軸的角度程式設計
ORIVIRT1： ORIVIRT2：	使用虛擬方向軸的方向程式設計 （定義 1），根據 MD $MC_ORIAX_TURN_TAB_1 的定義 （定義 2），根據 MD $MC_ORIAX_TURN_TAB_2 的定義
A3= B3= C3=：	方向軸的方向向量程式設計
ORIPLANE：	平面中插補（大半徑圓弧插補）
ORICONCW：	順時針方向的錐形柱面插補
ORICONCCW：	逆時針方向的錐形柱面插補
ORICONTO：	含切線變化的錐形柱面插補

A6= B6= C6=：	錐形旋轉軸的程式設計（正規化向量）
NUT=angle	錐形的缺口角度，以度為單位
NUT=+179	移動角度小於或等於 180 度
NUT=-181	移動角度大於或等於 180 度
ORICONIO：	錐形柱面上的插補
A7= B7= C7=：	中間方向（程式設計為正規化向量）
ORICURVE XH YH ZH，例如，使用多項式 PO[XH]=(xe, x2, x3, x4, x5)	指定刀具兩接點間動作的方向插補。 除終點外，也可程式設計其他曲度多項式。

說明

若透過方向軸補插含有效 ORIAXES 的刀具方向，僅在單節結束啟動相對於路徑的旋轉角度。

7.5.4 方向特性的平滑化（ORIPATHS A8=、B8=、C8=）

在輪廓上－特別在輪廓轉角處，以恆定加速進行的方向變更，可能造成不想要的路徑動作中斷。 可插入不同的中間單節，以平滑化在方向特性中形成的起伏。 若重新定向期間已啟用 ORIPATHS，則在恆定加速時發生方向變更。 刀具可在此階段回退。

機台製造商

請參考機台製造商，關於使用任何預先定義之機械資料及設定資料，啟用本函數的說明。

可使用機械資料來設定轉譯回退向量的方式：

1. 在 TCS 中，依刀具方向定義 Z 座標。
2. 在工件座標系統（WCS）中，依有效平面來定義 Z 座標。

關於「相對於路徑方向」函數的進一步解釋，請參閱

參考資料：

功能手冊，特殊功能，多軸轉換 (F2)

語法

需在輪廓轉角上進一步程式設計細節，才能恆定相對於路徑為整體的刀具方向。 此動作的方向及路徑長度，是透過向量，使用元件 A8=X、B8=Y C8=Z，來進行程式設計。

```
N... ORIPATHS A8=X B8=Y C8=Z
```

含義

ORIPATHS：	相對於路徑的刀具方向；已平滑化方向特性中的起伏
A8= B8= C8=：	方向及路徑長度的向量元件
X、Y、Z：	刀具方向中的回退移動

說明

程式設計方向向量 A8、B8、C8

若向量的長度正好為零點，則不會執行回退移動。

ORIPATHS

使用 ORIPATHS 啟動相對於路徑的刀具方向。 否則，方向會以線性大半徑圓弧插補，由起始方向傳送至結束方向。

7.6 方向的壓縮（COMPON、COMPCURV、COMPCAD、COMPSURF）

啟用方向轉換 (TRAORI) 且程式設計該刀具方向轉換的 NC 程式（無論何種類型），若維持在指定的臨界值內，則可進行壓縮。

程式設計

刀具方向

若在 5 軸機台啟用方向轉換 (TRAORI)，可依下列方式程式設計刀具方向（獨立於動力學）：

- 方向**向量**的程式設計透過：
  `A3=<...> B3=<...> C3=<...>`
- 尤拉**角**或 RPY **角**的程式設計透過：
  `A2=<...> B2=<...> C2=<...>`

刀具旋轉

在**六軸**機台上，除刀具方向外，亦可程式設計刀具旋轉。

程式設計旋轉角度，使用：

`THETA=<...>`

請參閱「刀具方向的旋轉 (頁 382)」。

說明

NC 單節中，僅若以**線性方式**變更旋轉角度時，才可壓縮程式設計的旋轉。即表示不應程式設計含旋轉角度 PO[THT]=(...) 的多項式。

可壓縮 NC 單節的一般結構

因此，可壓縮 NC 單節的一般結構如下所示：

```
N...X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>
```

或

```
N...X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

說明

可直接輸入位置值（例如 X90），或透過參數設定間接輸入（例如 X=R1*（R2+R3））。

使用旋轉軸位置程式設計刀具方向

也可使用旋轉軸位置指定刀具方向，例如，使用以下結構：

```
N...X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

在此情況下，以兩種不同方式執行壓縮，視是否執行大半徑圓弧插補而定。若無進行大半徑圓弧插補，則透過旋轉軸的軸多項式，以一般方式呈現已壓縮的方向變更。

輪廓準確度

依所選的壓縮模式 (MD20482 $MC_COMPRESSOR_MODE) 而定，設定的軸專屬公差 (MD33100 $MA_COMPRESS_POS_TOL)，或下列通道專屬公差－使用設定資料所設定－在壓縮幾何軸及方向軸時有效：

SD42475 $SC_COMPRESS_CONTUR_TOL（最大輪廓偏差）

SD42476 $SC_COMPRESS_ORI_TOL（刀具方向的最大角度偏差）

SD42477 $SC_COMPRESS_ORI_ROT_TOL（刀具旋轉角度的最大角度偏差）（僅適用於六軸機台）

參考資料：
功能手冊，基本功能；三至五軸轉換 (F2)，
章節：「路徑的壓縮」一章

啟用 / 停用

壓縮程序函數由模態 G 指令 COMPON、COMPCURV、COMPCAD 或 COMPSURF 啟動。

COMPOF 終止壓縮功能。

請參閱「NC 單節壓縮（COMPON、COMPCURV、COMPCAD） (頁 287)」。

說明

僅在啟用大半徑圓弧插補時，才壓縮方向動作（即在由起始及結束方向決定的平面中變更刀具方向）。

依下列條件執行大半徑圓弧插補：

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0，
  啟用 ORIWKS，且
  程式設計方向為向量（使用 A3、B3、C3 或 A2、B2、C2）。
- 啟用 MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 及
  ORIVECT 或 ORIPLANE。
  可程式設計刀具方向為方向向量，或以旋轉軸位置程式設計刀具方向。若已啟用 G 指令 ORICONxx 或 ORICURVE 其中之一，或若程式設計方向角度的多項式（PO[PHI] 及 PO[PSI]），則將不會執行大型圓弧插補。

範例

在以下範例程式中，會壓縮以多邊形定義逼近的圓弧。刀具方向同時在錐形外部移動。雖然逐一執行已程式設計的方向變更，但並不穩定，故壓縮函數會產生方向的平滑化動作。

程式設計	註解
DEF INT NUMBER=60	
DEF REAL RADIUS=20	
DEF INT COUNTER	
DEF REAL ANGLE	
N10 G1 X0 Y0 F5000 G64	
$SC_COMPRESS_CONTUR_TOL=0.05	；最大輪廓偏差 = 0.05 毫米
$SC_COMPRESS_ORI_TOL=5	；最大方向偏差 = 5 度
TRAORI	
COMPCURV	；此動作定義由多邊形產生的圓弧。方向以 45 度缺口角度在錐形上繞 Z 軸移動。
N100 X0 Y0 A3=0 B3=-1 C3=1	
N110 FOR COUNTER=0 TO NUMBER	
N120 ANGLE=360*COUNTER/NUMBER	
N130 X=RADIUS*cos（angle） Y=RADIUS*sin（angle）	
A3=sin（angle） B3=-cos（angle） C3=1	
N140 ENDFOR	

7.7 啟用 / 停用方向特性（ORISON，ORISOF）

使用指令 G 群組 61 在工件程式中啟用 / 停用「方向特性平滑化」。指令為模態。

先決條件

- 含 5/6 軸轉換的系統。
- 壓縮函數 COMPCAD 啟用。

語法

```
ORISON
...
ORISOF
```

含義

ORISON：	啟用方向特性平滑化
ORISOF：	停用方向特性平滑化

範例

程式碼	註解
...	
TRAORI()	；程式設計方向啟用。
COMPCAD	；啟用 COMPCAD 壓縮函數。
ORISON	；啟用方向平滑化。
$SC_ORISON_TOL=1.0	；刀具方向 = 1.0 度的最大角度偏差。
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	
...	

程式碼	註解
ORISOF	；方向平滑化停用。
...	

在 XZ 平面上，方向旋轉了超過 90 度，從-45 度到+45 度。因為方向特性的平滑化，方向不再能達到-45 度或+45 度的最大角度值。

7.8 動態轉換

7.8.1 啟動平面結尾處轉換 (TRANSMIT)

正面轉換 (TRANSMIT) 使用 TRANSMIT 敘述於工件程式或同步動作中啟動。

語法

```
TRANSMIT
TRANSMIT(<n>)
```

含義

TRANSMIT：	使用第一個 TRANSMIT 資料集啟動 TRANSMIT
TRANSMIT(n)：	使用第 n 個 TRANSMIT 資料集啟動 TRANSMIT

說明

通道中生效的 TRANSMIT 轉換會以下列方式啟動：

- 停用轉換運作： TRAFOOF
- 啟動其他轉換： 例如 TRACYL、TRAANG、TRAORI

7.8.2 啟動柱體表面轉換 (TRACYL)

圓柱表面轉換 (TRACYL) 使用 TRACYL 敘述於工件程式或同步動作中啟動。

語法

```
TRACYL(<d>)
TRACYL(<d>,<n>)
TRACYL(<d>,<n>,<k>)
```

含義

TRACYL(<d>)：	使用第一個 TRACYL 資料集及工作直徑 **<d>** 啟動 TRACYL	
TRACYL (<d>,<n>)：	使用第 **<n>** 個 TRACYL 資料集及工作直徑 **<d>** 啟動 TRACYL	
<d>：	參考或工作直徑 數值必需大於 1。	
<n>：	TRACYL 資料集編號（選配）	
	值域：	1, 2
<k>：	參數 **<k>** 只跟轉換類型 514 相關：	
	k = 0：	不含溝槽壁修正
	k = 1：	含溝槽壁修正
	若未指定參數，則將套用參數化基本位置： $MC_TRACYL_DEFAULT_MODE_<n> 其中 **<n>** = TRACYL 資料集編號	

說明

通道中生效的 TRACYL 轉換會以下列方式關閉：

- 停用轉換運作：TRAFOOF
- 啟動其他轉換： 例如 TRAANG、TRANSMIT、TRAORI

範例

程式碼	註解
...	
N40 TRACYL(40.)	；使用第一個 TRACYL 資料集及工作直徑 40 毫米啟動 TRACYL
...	

其他資訊

程式結構

使用 TRACYL 轉換 513 銑削溝槽的工件程式（TRACYL 搭配溝槽邊偏移）一般由下列步驟組成：

1. 選擇刀具。
2. 選擇 TRACYL。
3. 選擇合適的座標偏移（框架）

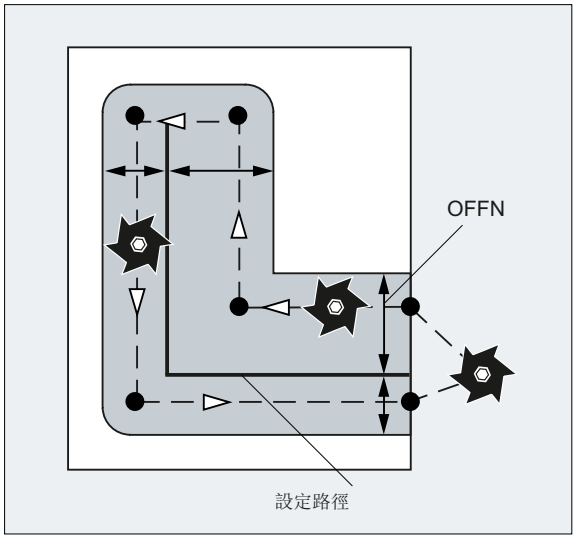
4. 定位。
5. 程式設定 OFFN。
6. 選擇 TRC。
7. 逼近單節（定位 TRC 並逼近溝槽邊）。
8. 溝槽中心線輪廓。
9. 取消選擇 TRC。
10. 回退單節（退出 TRC 並離開溝槽邊）。
11. 定位。
12. TRAFOOF。
13. 重新選擇座標偏移（框架）。

輪廓偏移 (OFFN)

為了使用 TRACYL 轉換 513 銑削溝槽，會在工件程式中編程設定透過 OFFN 位址的溝槽中心線及**一半溝槽寬度**。

OFFN 僅在啟用刀具半徑補正時才動作，避免損壞溝槽側邊。

可變更工件程式內的 OFFN。這讓溝槽中心線可以偏離中心：



說明

OFFN 至少要和刀具半徑一樣，以免損壞溝槽壁的對側。

說明

OFFN 含 TRACYL 與不含 TRACYL 的動作不同。因為，即使沒有 TRACYL，當 TRC 啟用後，OFFN 會加入，TRAFOOF 後 OFFN 需重置為零。

注意
OFFN 效應需視轉換類型而定 對 TRACYL 轉換 513（含溝槽邊偏移的 TRACYL）而言，會針對 OFFN 編程設定一半的溝槽寬度。 對 TRACYL 轉換 512（含溝槽邊偏移的 TRACYL）而言，會以 OFFN 值作為 TRC 允差。

刀具半徑補正 (TRC)

對 TRACYL 轉換 513 而言，將 TRC 納入考量的不是有關溝槽邊的部分，而是編程設定的溝槽中心。為了讓刀具移動至溝槽邊左側，必須編程設定敘述 G42 取代 G41，或是指定具有負號的 OFFN 值。

刀具直徑

使用 TRACYL 及直徑小於溝槽寬度的刀具，因為與溝槽寬度相同的刀具直徑，無法產生相同的溝槽邊幾何。若欲改善精度，建議選用直徑稍微小於溝槽寬度的刀具。

軸利用

說明

不得使用跟隨軸為定位軸或往返軸：

- 柱體柱面（Y 軸）的柱面方向中的幾何軸。
- 溝槽側邊補正（Z 軸）的其他線性軸。

7.8.3 使用可程式設計角度 (TRAANG) 啟動傾斜角轉換

使用 TRAANG 敘述在工件程式或同步動作中，以可程式設計角度啟動傾斜角轉換。

語法

```
TRAANG
TRAANG()
TRAANG(, <n>)
TRAANG(<α>)
TRAANG(<α>,<n>)
```

含義

TRAANG： TRAANG()：	使用第一個 TRAANG 資料集及最後一個有效角度 <α> 啟動 TRAANG	
TRAANG(, <n>)：	使用第 <n> 個 TRAANG 資料集及最後一個有效角度 <α> 啟動 TRAANG	
TRAANG(<α>)：	使用第一個 TRAANG 資料集及角度 <α> 啟動 TRAANG	
TRAANG(<α>,<n>)：	使用第 <n> 個 TRAANG 資料集及角度 <α> 啟動 TRAANG	
<α>：	斜式軸的角度（選配）	
	值域：	-90° < α < + 90°
	若未指定角度，在機械資料中的參數化初始狀態為有效： MD2xxxx $MC_TRAANG_ANGLE_<n>	
<n>：	TRAANG 資料集編號（選配）	
	值域：	1、2

說明

通道中生效的 TRAANG 傾斜角轉換如下使用已停用：

- 停用轉換運作：TRAFOOF
- 啟動其他轉換：例如 TRACYL、TRANSMIT、TRAORI

範例

程式碼	註解
N20 TRAANG(45)	；使用第一個 TRAANG 資料集和角度 45° 啟動 TRAANG

7.8.4 研磨機的傾斜直進切削（G5、G7）

使用 G 碼功能 G7 及 G5，簡化以「傾斜軸 (TRAANG)」程式設計研磨機的傾斜直進切削，因此在直進切削時，只有傾斜軸移動。

X 及 Z 只需要程式設計直進切削動作所要求的結束位置，從 X 軸實際位置啟動的 G7，NC 會計算及逼近程式設計的傾斜軸之結束位置及角度 α。

起始位置是以兩條直線的交點計算出來：

- 平行於 Z 軸的直線，與 X 軸實際位置的距離
- 平行於傾斜軸的直線通過程式設計的結束位置

使用後續的 G5，將傾斜軸移動至程式設計的結束位置。

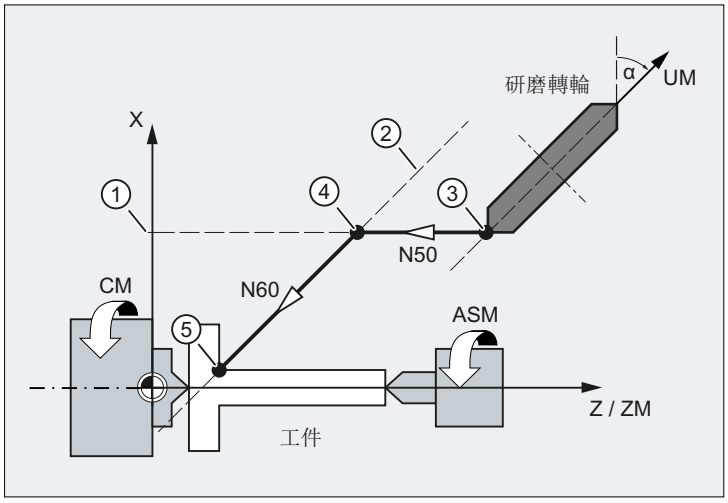
語法

```
G7 <Endpos_X> <Endpos_Z>
G5 <Endpos_X>
```

含義

G7：	計算傾斜直進切削的起始位置及逼近
G5：	將傾斜軸移動至程式設計的結束位置
<Endpos_X>：	X 軸結束位置
<Endpos_Z>：	Z 軸的結束位置

範例



① 平行於 Z 軸，與 X 軸實際位置的距離

② 平行於傾斜軸，通過程式設計的結束位置

③ 起始位置

④ 直進切削：起始位置

⑤ 直進切削：結束位置

X 幾何軸

Z 幾何軸

ZM 機械軸

UM 機械軸

程式碼	註解
`N...G18`	；選擇 XZ 平面。
`N40 TRAANG (45.0)`	；啟動 TRAANG 轉換，角度 = 45°
`N50 G7 X40 Z70 F4000`	；計算起始位置及逼近
`N60 G5 X40 F100`	；將傾斜軸移動至結束位置。
`N70 ...`	

7.9 啟動序連轉換 (TRACON)

TRAANG 轉換使用 TRACON 敘述在工件程式或同步動作中啟動。

語法

```
TRACON(<Trafo_No>,<Par_1>,<Par_2,...)
...
TRAFOOF
```

含義

TRACON：	啟動序連轉換 若已先啟動其他轉換，則由 TRACON() 隱含停用。		
<Trafo_No>：	序連轉換的編號：		
	類型：	INT	
	值域：	0 ... 2	
	值：	0、1	第一 / 唯一的序連轉換
		2	第二個序連轉換
		未指定	與 0 或 1 的意義相同
		附註： 值不等於 0、1、2 會產生錯誤警報。	
<Par_1>、 <Par_2,...：	序連轉換參數（例如傾斜軸角度） 若未設定參數，則預設值或上次使用的參數會生效。 若先前參數的預設設定會生效，則需使用逗號，以確保會依預期的順序評估指定的參數。特別是雖無需指定 <Trafo_No>，但至少一個參數前需一個逗號。例如 TRACON(, 3.7)。 適用於含 Transmit 或 TRAORI（5 軸加工）的 TRACON，第二參數 Par_2 不能當作傾斜軸的角度。 對於含 TRACYL（柱面加工）的 TRACON，使用 Par 作為單元直徑。		
TRAFOOF：	停用上次啟動的（序連）轉換		

範例

程式碼	註解
...	
N230 TRACON(1,45.)	；啟動第一個序連轉換。
	；之前生效的轉換會自動取消選擇。
	；傾斜軸參數為 45°。
...	
N330 TRACON(2,40.)	；啟動第二個序連轉換。
	；傾斜軸參數為 40°。
...	
N380 TRAFOOF	；停用第二個序連轉換。
...	

7.10 直角座標 PTP 移動

7.10.1 啟動 / 停用直角座標 PTP 移動（PTP、PTPG0、PTPWOC、CP）

直角座標點對點或 PTP 移動是在 NC 程式內使用 G 群組 49 指令啟動 / 停用。

指令為模態。預設設定為使用直角座標路徑動作 (CP) 移動。

相對於 CP，啟用中 PTP 移動只有直角座標目標點會轉換，且機械軸將與主導軸同步運行。

為了讓直角座標目標點可唯一轉換為機械軸數值，除了位置及角度資料，也需要識別軸位置的資訊。此資料是從可調整位址 STAT (頁 407) 及 TU (頁 412) 擷取。

先決條件

啟用轉換 TRAORI、TRANSMIT、RCTRA 或 ROBX。

語法

```
PTP / PTPG0 / PTPWOC
...
CP
```

意義

PTP：	啟動點對點動作 PTP G0 及 G1 單節的程式設計直角座標位置可用同步軸動作接近。
PTPG0：	啟動點對點動作 PTPG0 只有 G0 單節的程式設計直角座標位置可用同步軸動作接近。G1 單節，需切換到 CP 路徑動作。
PTPWOC：	啟動點對點移動 PTPWOC（只有方向轉換啟用時為可能） 與 PTP 相同的是，不需由旋轉軸和方向軸動作引發的任何補正動作。
CP：	停用點對點動作並啟動路徑動作 CP 直角座標路徑動作由 CP 執行。

說明

PTPWOC

將 PTPWOC 搭配使用 RCTRA 或 ROBX 轉換完全不可行！

範例

請參閱：

- 範例 1：含 ROBX 轉換的 6 軸機械手臂 PTP 移動 (頁 415)
- 範例 2：含一般 5 軸轉換之 PTP 移動 (頁 416)
- 範例 3：PTPG0 與 TRANSMIT (頁 416)

7.10.2 指定接點的位置 (STAT)

含直角座標的位置資料和指定刀具方向不足以唯一識別機台位置，就相同的刀具方向來說，可有多個接點位置。依與其有關的動力學而定，可能有多達 8 個不同的接點位置。這些不同的接點位置為轉換專屬。

為避免任何不明狀況，接點位置在 STAT 位址下指定。

說明

控制只會將 PTP 動作程式設計的 STAT 數值納入考量。CP 動作將忽略，因為使用啟用中轉換移動時，通常不允許改變位置。使用啟用中 CP 移動時，目標點的位置將從開始點取得。

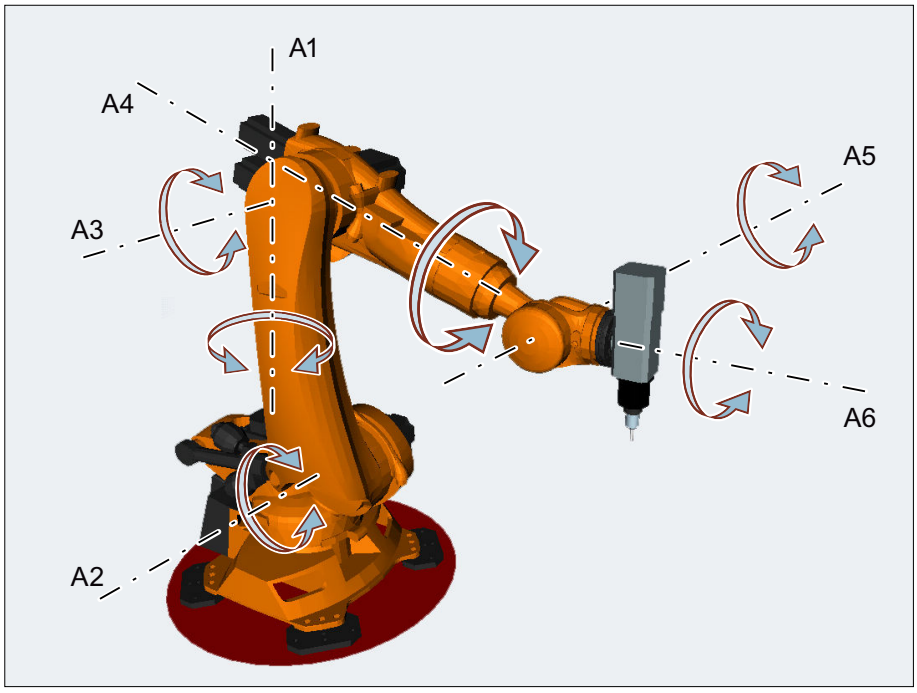
語法

```
STAT=<Value>
```

意義

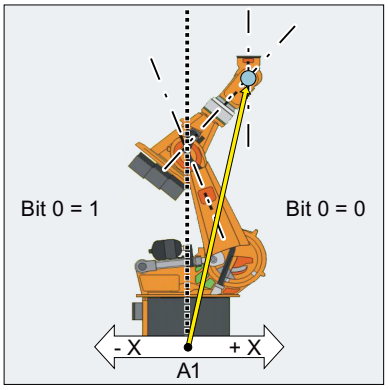
STAT：	用於指定接點位址的可調整位址
<Value>：	二進制或十進制值 包含各可能位置的一個位元。位元的重要性由特定轉換指定。

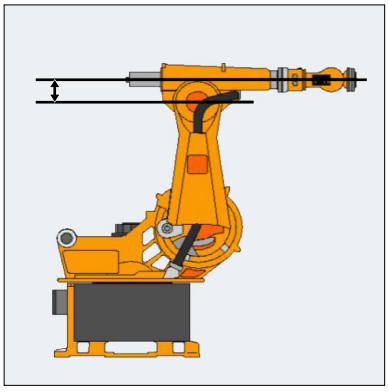
使用 STAT 必須使用 6 軸關節機械手臂以銑削主軸加以說明。動態轉換必須使用 ROBX 機械手臂轉換履行（先決條件：載入並啟用編譯循環「RMCC/ROBX 轉換擴充機械手臂」）。



軸 A1、A2 及 A3 是關節機械手臂的主要軸。除了主要軸以外，軸 A4、A5 及 A6 分別命名為頭軸或手軸 / 腕軸，都用來在加工空間中定位。加上其他的手軸 / 腕軸的動作結果，銑削主軸可以針對特定的加工工作需求，在空間定出方向。可以利用各種關節的接頭位置，做到相同的刀具方向。

加工所需的關節接頭位置，可以選擇可調整 STAT 位址的程式設計位元 0 ... 2：

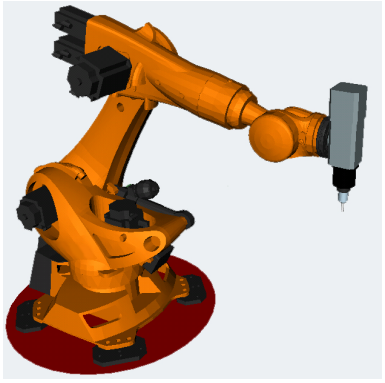
<table>
<tr><td rowspan="3">位元 0</td><td colspan="2">手動軸交叉點位置（A4、A5、A6）</td><td rowspan="3"> 

範例：手軸 / 腕軸的交點位置落在基本範圍內</td></tr>
<tr><td>= 0</td><td>基本範圍（肩部向右）
如果手動軸交叉點的 X 值相對於 A1 座標系統為正值，則機器人在基本範圍內。</td></tr>
<tr><td>= 1</td><td>額外範圍（肩部向左）
如果手動軸交叉點的 X 值相對於 A1 座標系統為負值，則機器人在額外範圍內。</td></tr>
<tr><td rowspan="4">位元 1</td><td colspan="2">軸 3 位置
位元 1 數值變更的角度取決於特定的機器人類型。以下適用於軸 3 和軸 4 交叉的機器人：</td></tr>
<tr><td>= 0</td><td>A3 < 0°（肘部向下）</td></tr>
<tr><td>= 1</td><td>A3 ≥ 0°（肘部向上）</td></tr>
<tr><td colspan="2">附註：
軸 3 和 4 之間有偏移量的機器人，位元 1 數值變更的角度取決於偏移量的大小。</td><td> 
A3 與 A4 之間的偏移量</td></tr>
<tr><td rowspan="3">位元 2</td><td colspan="2">軸 5 位置</td></tr>
<tr><td>= 0</td><td>A5 ≤ 0°（手沒有翻轉）</td></tr>
<tr><td>= 1</td><td>A5 > 0°（手翻轉）</td></tr>
</table>

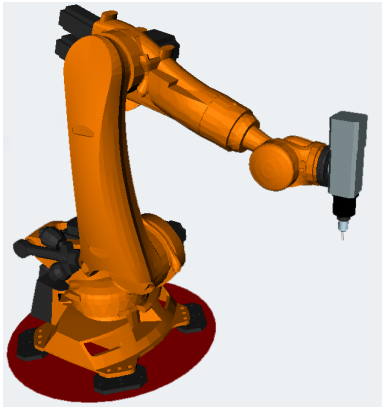
程式範例：

程式碼	註解
...	
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	；STAT 值定義關節接頭位置（請見下方）
...	

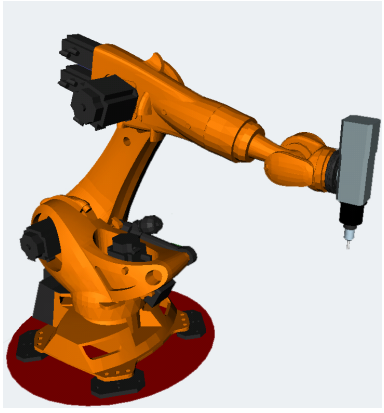
STAT=1 ('B001') → 肩部向左
→ 肘部向下
→ 手沒有翻轉



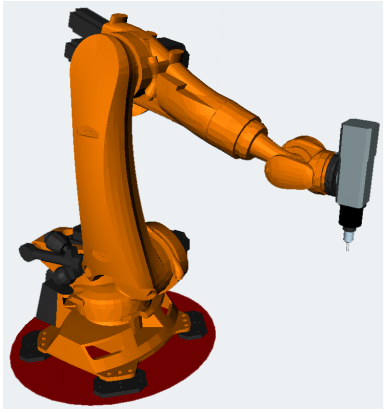
STAT=2 ('B010') → 肩部向右
→ 肘部向上
→ 手沒有翻轉



STAT=5 ('B101') → 肩部向左
→ 肘部向下
→ 手翻轉



STAT=6 ('B110') → 肩部向右
→ 肘部向上
→ 手翻轉



TRANSMIT

TRANSMIT 使用 STAT 位址初始化關於極點的等價性。

如果旋轉軸必須旋轉 180º 或 CP，輪廓將經過極點，下列將適用：

位元 0	僅相關於 $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 或 2：	
	= 0	旋轉軸移動經過 +180º 或順時針旋轉。
	= 1	旋轉軸旋轉經過 -180º 或逆時針旋轉。
位元 1	僅相關於 $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0：	
	= 0	軸移動經過極點。旋轉軸不旋轉。
	= 1	軸沿著極點旋轉。STAT 的位元 0 為相關。

7.10.3 指定軸角度 (TU) 的符號

為了讓旋轉軸不需特殊的移動策略（如中間點）也能接近超出 +180° 或小於 -180° 的軸角度，必須在可調整位址的 TU 底下指定軸角度的符號。

說明

控制只會考慮 PTP 動作的程式設計 TU 數值。忽略 CP 動作。

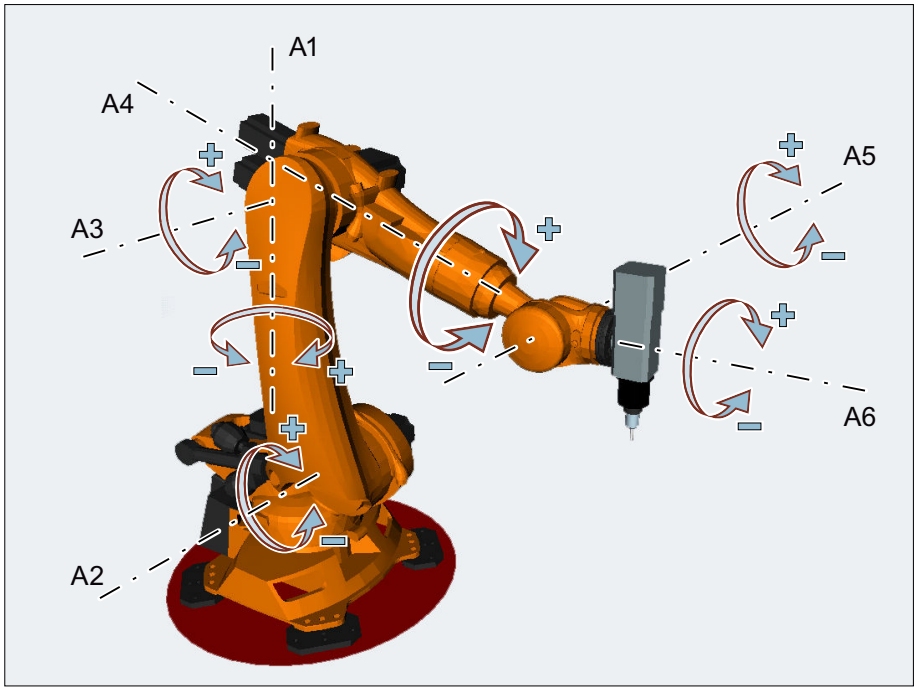
語法

```
TU=<Value>
```

含義

TU：	用於指定軸角度符號的可調整位址			
<Value>：	二進制或十進制值 針對與轉換有關的每個軸，都有一個位元標示軸角度符號 (θ)，以及移動方向。			
	位元	= 0	軸角度符號：+	軸角度範圍：0° ≤ θ < 360°
		= 1	軸角度符號：-	軸角度範圍：- 360° < θ < 0°

範例：6 軸關節機械手臂



位元	意義	值	軸角度符號	軸角度
位元 0 [1]	A1 的軸角度符號	= 0	+	≥ 0°
		= 1	-	< 0°
位元 1 [1]	A2 的軸角度符號	= 0	+	≥ 0°
		= 1	-	< 0°
位元 2 [1]	A3 的軸角度符號	= 0	+	≥ 0°
		= 1	-	< 0°
位元 3 [1]	A4 的軸角度符號	= 0	+	≥ 0°
		= 1	-	< 0°
位元 4 [1]	A5 的軸角度符號	= 0	+	≥ 0°
		= 1	-	< 0°
位元 5 [1]	A6 的軸角度符號	= 0	+	≥ 0°
		= 1	-	< 0°

[1] 從機械手臂軸的通道軸編號獲得的實際 TU 位元編號！在範例中，機械手臂軸（A1 至 A6）為通道內的前六個軸；因此使用了 TU 位元 0 ... 5。對於機械手臂軸的另一個通道軸指派，機械手臂軸的 TU 位元編號也會隨之變更（例如：機械手臂軸為第三至第八通道軸，亦即使用 TU 位元 2 ... 7 為機械手臂軸）。

TU=19（對應至 TU='B010011'）因此代表：

位元	值		軸角度
0	= 1	①	$\theta_{A1} < 0°$
1	= 1	①	$\theta_{A2} < 0°$
2	= 0	①	$\theta_{A3} \geq 0°$
3	= 0	①	$\theta_{A4} \geq 0°$
4	= 1	①	$\theta_{A5} < 0°$
5	= 0	①	$\theta_{A6} \geq 0°$

說明

假如軸的移動範圍 > ±360°，由於 TU 資訊無法唯一指定軸位置，因此軸將永遠沿著最短的路徑移動。

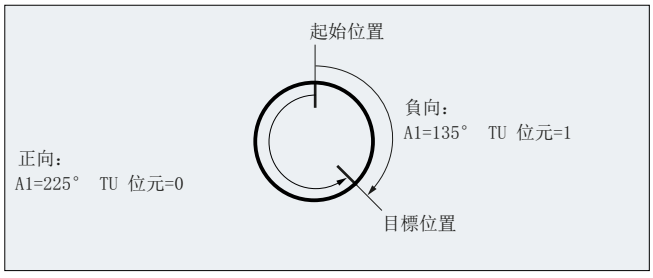
若位置並未程式設計 TU，則根據 MD30455 $MA_MISC_FUNCTION_MASK，移動較短或較長的路徑（請參閱在延伸函數功能手冊中的章節「將 PTP 移動的軟體限制納入考量」）。

TRANSMIT

如為啟用 TRANSMIT 的 PTP 移動，TU 位址則無意義！

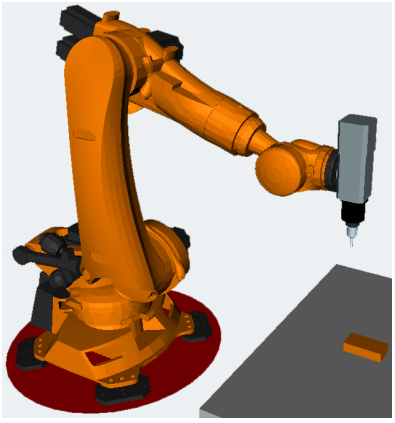
範例

可依正向或負向逼近下圖中所示的旋轉軸位置。在位址 A1 下程式設計角度位置。移動方向只有在指定 TU 時為絕對清楚。



7.10.4 範例 1：含 ROBX 轉換的 6 軸機械手臂 PTP 移動

在下列的應用範例中，直角座標 PTP 移動及相關的 NC 指令如範例之形式所示。



圖像 7-3　含銑削主軸的 6 軸關節機械手臂

```
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 範例 2：含一般 5 軸轉換之 PTP 移動

假設：用作基礎的直角 CA 動力學。

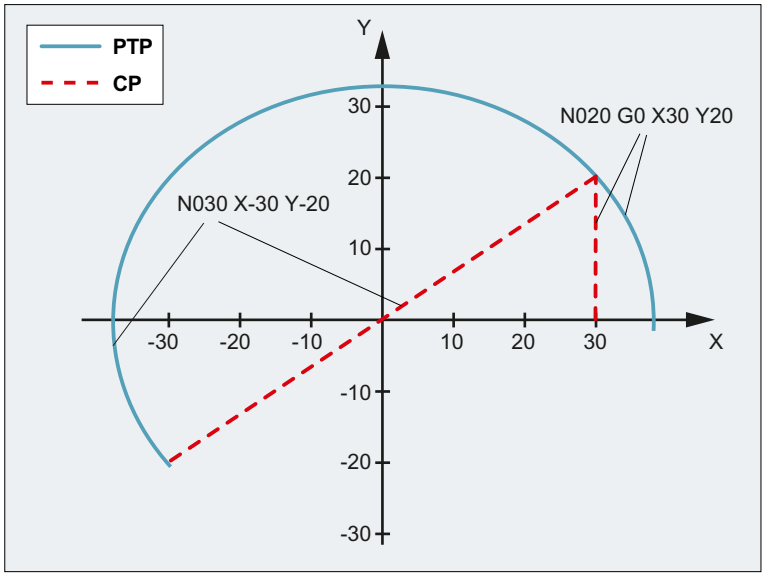
程式碼	註解
TRAORI	；轉換 CA 動力學啟動
PTP	；啟動 PTP 移動
N10 A3=0 B3=0 C3=1	；旋轉軸位置 C=0 A=0
N20 A3=1 B3=0 C3=1	；旋轉軸位置 C=90 A=45
N30 A3=1 B3=0 C3=0	；旋轉軸位置 C=90 A=90
N40 A3=1 B3=0 C3=1 **STAT=1**	；旋轉軸位置 C=270 A=-45

選擇旋轉軸位置的準確逼近位置：

在單節 N40 中，程式設計 **STAT=1** 時，旋轉軸會從起點（C=90，A=90）長距離移動到終點（C=270，A=–45）。另一方面，**STAT=0** 時，旋轉軸則沿著最短的路徑移動到終點（C=90，A=45）。

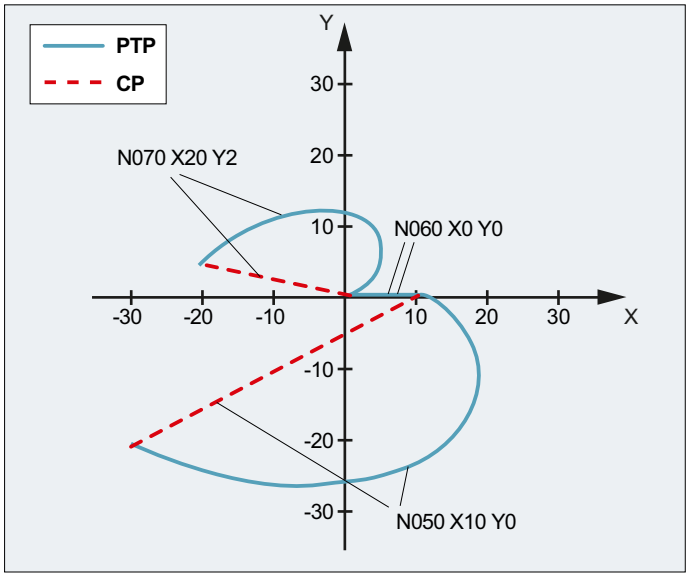
7.10.6 範例 3：PTPG0 與 TRANSMIT

使用 PTPG0 與 TRANSMIT 沿著極點移動



程式碼	註解
N001 G0 X30 Z0 F10000 T1 D1 G90	；初始設定，絕對尺寸
N002 SPOS=0	
N003 **TRANSMIT**	；TRANSMIT 轉換
N010 **PTPG0**	；每一個 G0 單節，自動 PTP，然後再次 CP。
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

使用 PTPG0 與 TRANSMIT 從極點移動



程式設計	註解
N001 G0 X90 Z0 F10000 T1 D1 G90	；初始設定
N002 SPOS=0	
N003 **TRANSMIT**	；TRANSMIT 轉換
N010 **PTPG0**	；每一個 G0 單節，自動 PTP，然後再次 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	

程式設計	註解
N150 X-30 Y-20	
M30	

7.11 選擇轉換時的限制

功能

可透過工件程式或 MDA（手動輸入）選擇轉換。請注意：

- 不可插入中間動作單節（倒角 / 半徑）。
- 需排除曲線單節順序；否則會顯示訊息。
- 需取消選擇微調刀具補正（FTOCOF）；否則會顯示訊息。
- 需取消選擇刀具半徑補正（G40）；否則會顯示訊息。
- 已啟動的刀長偏移量，係由控制系統包含在轉換中。
- 控制系統取消選擇目前啟用的框架，再進行轉換。
- 控制系統為受轉換影響的軸，取消選擇啟用的操作範圍限制（與 WALIMOF 相同）。
- 取消選擇保護區監控。
- 中斷連續路徑控制以及倒圓角作業。
- 需相對於單節同步處理所有在機械資料中指定的軸。
- 將已交換的軸交換回來；否則會顯示訊息。
- 輸出相關軸的訊息。

換刀

僅在取消選擇刀具半徑補正函數時，才能變更刀具。

刀長偏移量中的變更，及刀具半徑補正選擇 / 取消選擇，不得在同一單節中進行程式設計。

框架變更

允許所有只參考基本座標系統的敘述（FRAME、刀具半徑補正）。然而，使用 G91 的框架變更（增量尺寸）不同於未啟用的轉換，不分別做處理。在新框架的工件座標系統中評估欲移動的增量，不管前一個單節中為哪個有效框架。

例外情況

不能使用受轉換影響的軸

- 為預先設定的軸（警報）、
- 逼近檢查點（警報）、
- 供參考（警報）。

7.12 取消選擇轉換 (TRAFOOF)

預先定義的 TRAFOOF 程序停用所有有效的轉換與框架。

說明

如要取消選取轉換，也適用選取時相同的次要條件 (頁 419)。
需由更新的程式設計啟動之後所需的框架。

語法

```
...
TRAFOOF
```

含義

TRAFOOF：	停用所有有效的轉換/框架

動態鍊 8

8.1 刪除元件 (DELOBJ)

DELOBJ() 函數會將指定的系統變數重置為預設值，以刪除「元件」：

- 動態鍊元素
- 保護區、保護區元素、碰撞對
- 轉換資料

語法

```
[<RetVal>=] DELOBJ(<CompType>[,,,<NoAlarm>)])
[<RetVal>=] DELOBJ(<CompType>,<Index1>[,,<NoAlarm>])
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][,<Index2>][,<NoAlarm>])
```

意義

DELOBJ：	刪除動態鍊元素、保護區、保護區元素、碰撞對及轉換資料

<CompType>：	欲刪除的元件類型	
	資料類型：	STRING
	值："KIN_CHAIN_ELEM" **意義**：所有運動元素的系統變數：$NK_...	
	值："KIN_CHAIN_SWITCH" **意義**：系統變數 $NK_SWITCH[<i>]	
	值："KIN_CHAIN_ALL" **意義**：所有運動元素 及開關。 DELOBJ 連續呼叫與「KIN_CHAIN_ELEM」和 「KIN_CHAIN_SWITCH」相同	
	值："PROT_AREA" **意義**：保護區系統變數： • $NP_PROT_NAME • $NP_CHAIN_NAME • $NP_CHAIN_ELEM • $NP_1ST_PROT	
	值："PROT_AREA_ELEM" **意義**：機台保護區和/或自動刀具保護區的保護區元素系統變數： • $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	
	值："PROT_AREA_COLL_PAIRS" **意義**：碰撞對系統變數： • $NP_COLL_PAIR • $NP_SAFETY_DIST	

	值："PROT_AREA_ALL" **意義**：所有保護區、保護區元素、碰撞對（系統變數 $NP_...） DELOBJ 連續呼叫與「PROT_AREA」、「PROT_AREA_ELEM」和「PROT_AREA_COLL_PAIRS」相同	
	值："TRAFO_DATA" **意義**：所有轉換的系統變數：$NT_...	
<Index1>：	欲刪除第一個元件的索引（**選擇性**）	
	資料類型：	INT
	預設值：	-1
	值域：	-1 ≤ x ≤（已設定元件的最大編號 -1）
	值	**意義**
	0、1、2...	欲刪除元件索引。
	-1	刪除已指定類型的所有元件。未評估 **<Index2>**。
<Index2>：	欲刪除最後一個元件的索引（**選擇性**） 若未程式設計 **<Index2>**，只刪除 **<Index1>** 中參考的元件系統變數。	
	資料類型：	INT
	預設值：	只刪除 **<Index1>** 中參考的元件系統變數。
	值域：	**<Index1>** < x ≤（已設定元件的最大編號 -1）
<NoAlarm>：	警報抑制（**選擇性**）	
	資料類型：	BOOL
	預設值：	FALSE
	值	**意義**
	FALSE	錯誤的狀況中 (**<RetVal>** < 0)，會停止程式處理，並顯示警報.
	TRUE	錯誤的狀況中，不會停止程式處理，也不顯示警報。 應用：對應於傳回值的使用者專屬反應

<RetVal>：	函數傳回值	
	資料類型：	INT
	值域：	0、-1、-2、... -7
	值	**意義**
	0	未發生任何錯誤。
	-1	呼叫無參數的功能。至少必須指定參數 **<CompType>**。
	-2	**<CompType>** 識別未知的元件
	-3	**<Index1>** 小於 -1
	-4	**<Index1>** 大於元件設定的編號
	-5	**<Index1>** 刪除元件**群組**時，數值不等於 -1
	-6	**<Index2>** 小於 **<Index1>**
	-7	**<Index2>** 大於元件設定的編號

8.2 以名稱 (NAMETOINT) 的方法判定索引

將使用者專屬名稱輸入 STRING 類型的系統變數陣列中。依據系統變數及名稱的識別碼，NAMETOINT() 函數可以判定系統變數陣列中所儲存之名稱所屬的索引值。

語法

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

意義

NAMETOINT：	判定系統變數索引	
<SysVar>：	STRING 類型的系統變數陣列名稱	
	資料類型：	STRING
	值域：	STRING 類型的所有 NC 系統變數陣列名稱
<Name>：	欲判定系統變數索引的字元字串或名稱。	
	資料類型：	STRING
<NoAlarm>：	警報抑制（**選擇性**）	
	資料類型：	BOOL
	預設值：	FALSE
	值	**意義**
	TRUE	錯誤的狀況中，不會停止程式處理，也不顯示警報。 應用：對應於傳回值的使用者專屬反應
	FALSE	錯誤的狀況中 (<RetVal> < 0)，會停止程式處理，並顯示警報。
<RetVal>：	系統變數索引或錯誤訊息	
	資料類型：	INT
	值域：	-1 ≤ x ≤（已設定元件的最大編號 -1）
	值	**意義**
	≥ 0	搜尋名稱已在指定的系統變數索引下找到。
	-1	未搜尋到名稱，或發生錯誤。

範例

程式碼	註解
DEF INT INDEX	
$NP_PROT_NAME[27]="Cover"	
...	
INDEX = NAMETOINT("$NP_PROT_NAME","Cover")	; INDEX == 27

動態鍊的碰撞避免

9

說明

保護區

有關以下章節所指的保護區，請參閱「幾何機台模型」功能

參考資料：

功能手冊，特殊函數，章節「幾何機台模型」

9.1 檢查碰撞對 (COLLPAIR)

COLLPAIR() 功能判定兩個保護區是否組成碰撞對。

語法

```
[<RetVal> =] COLLPAIR(<Name_1>,<Name_2>[,<NoAlarm>)])
```

含義

COLLPAIR：	檢查是否碰撞對的一部份		
<RetVal>：	函數傳回值		
	資料類型：	INT	
	值：	≥ 0	兩個保護區組成碰撞對。 傳回值 == 碰撞對索引 m（參閱 $NP_COLL_PAIR）
		-1	或是兩個字串未經指定，或是其中一個字串為零字串。
		-2	未找到第一個參數指定的保護區。
		-3	未找到第二個參數指定的保護區。
		-4	兩個指定保護區均未發現。
		-5	已找到兩個指定的保護區，但未在同一碰撞對中。
<Name_1>：	第一個保護區的名稱		
	資料類型：	STRING	
	值域：	參數化的保護區名稱	
<Name_2>：	第二個保護區的名稱		
	資料類型：	STRING	
	值域：	參數化的保護區名稱	

<NoAlarm>：	警報抑制（**選配**）		
	資料類型：	BOOL	
	值：	FALSE（預設）	錯誤的狀況中 (<RetVal> < 0)，會停止程式處理，並顯示警報。
		TRUE	錯誤的狀況中，不會停止程式處理，也不顯示警報。 應用：對應於傳回值使用者專屬反應

9.2 要求重新計算碰撞避免的機台模型 (PROTA)

若是將動態鏈 $NK_... 的系統變數、幾何機台模型或碰撞避免 $NP_... 寫入工件程式，PROTA 程序必須接著呼叫，使得 NC 內部的碰撞避免機台模型變更成為有效。

語法

```
PROTA[(<Par>)]
```

含義

PROTA：	要求重新計算碰撞避免的機台模型		
	● 觸發前置處理停止。 ● 必須單獨放在單節中。		
<Par>：	參數（**選用**）		
	資料類型：	STRING	
	值：	---	無參數。 重新計算機台模型。保留保護區的狀態。
		「R」	重新計算機台模型。將保護區設定為對應於 $NP_INIT_STAT 的初始化狀態。

補充條件

模擬

在工件程式中 PROTA 程序不得搭配模擬 (simNC) 一起使用。

範例：啟用模擬時要避免呼叫 PROTA。

程式碼	註解
...	
IF $P_SIM == FALSE	;IF 模擬未啟用
PROTA	THEN 重新計算碰撞模型
ENDIF	; ENDIF
...	

另請參見

設定保護區狀態 (PROTS) (頁 433)

9.3 設定保護區狀態 (PROTS)

PROTS() 程序將保護區狀態設定為指定值。

語法

```
PROTS(<State>[,<Name_1>,...,<Name_n>])
```

含義

PROTS：	設定保護區的狀態		
	● 必須單獨放在單節中。		
<State>：	欲設定之指定保護區的狀態		
	資料類型：	CHAR	
	值：	「A」或「a」	狀態：啟用中
		「I」或「i」	狀態：非使用中
		「P」或「p」	狀態：預先啟動或 PLC 控制 [1]
		「R」或「r」	狀態：初始化狀態的 NC 內部值 [2]
<Name_1> ... <Name_n>：	欲設定至指定狀態的一個或多個保護區名稱（**選用**） 若未指定名稱，則指定的狀態設定給所有定義的保護區使用。		
	資料類型：	STRING	
	值域：	參數化的保護區名稱	
	附註： 保護區可以指定為參數的最大編號，只跟每個程式行最大可能的字元數有關。		

[1] 啟用/停止啟用如下執行：DB10.DBX234.0 - DBX241.7

[2] 狀態設定為 NC 內部的初始化狀態值，亦即系統變數 $NP_INIT_STAT 在最後一次 PROTA( ) (頁 432) 呼叫時的數值。

9.4 判定兩個指定保護區的間隙 (PROTD)

PROTD() 功能計算兩個保護區的間隙。

功能屬性：

- 執行間隙計算時與保護區的狀態無關（已啟用、停用、預先啟用）。
- 若要計算兩個保護區的間隙，只能使用保護區元素，其以 $NP_USAGE = "C" 或 "A" 標記。保護區的保護區元素是以 $NP_USAGE = "V" 標記，並不納入考量。
- 對於所有以 $NP_USAGE = "V" 標記的保護區元素的保護區，不能拿來用在間隙計算上。
- 執行間隙計算時前一個單節結尾的位置有效。
- 包含在主要執行計算之中的覆蓋功能（例如 DRF 偏移量或外部零點偏移量）係包括在間隙計算中，位在功能**解譯時間**的數值為有效。

說明

同步

在使用 PROTD() 功能時，使用者唯一的責任就是同步化主要的執行及前置處理，必要時使用 STORE 前置處理停止。

碰撞

如果兩個指定的保護區之間發生碰撞， 函數即傳回 0.0 間隙。若是兩個保護區接觸或彼此交錯，則會有碰撞。

碰撞檢查的安全間距 (MD10622 $MN_COLLISION_SAFETY_DIST) 在間隙計算中不予考慮。

語法

```
[<RetVal> =] PROTD([<Name_1>],[<Name_2>],VAR <Vector>[,<System>])
```

含義

PROTD:	計算兩個指定保護區的間隙。	
	• 必須單獨放在單節中。	
<RetVal>：	函數傳回值：兩個保護區的間隙絕對值或碰撞的 0.0（如上所示：碰撞一節）	
	資料類型：	REAL
	值域：	0.0 ≤ x ≤ +最大 REAL 值

<Name_1>, <Name_2>：	欲計算間隙的兩個保護區名稱（**選配**）。		
	資料類型：	STRING	
	值域：	參數化的保護區名稱	
	預設值：	""（空字串） 若未指定保護區，功能會從碰撞模型中的所有已啟用及預先啟用的保護區，計算出目前最小的間隙。	
<Vector>：	傳回值：3 維間隙向量，從保護區 **<Name_2>** 至保護區 **<Name_1>**，其中： • **<Vector>[0]**:世界座標系統中的 X 座標 • **<Vector>[1]**:世界座標系統中的 Y 座標 • **<Vector>[2]**:世界座標系統中的 Z 座標 碰撞：**<Vector> ==** 零向量		
	資料類型：	VAR REAL [3]	
	值域：	**<Vector>** [n]:0.0 ≤ x ≤ ± 最大 REAL 值	
<System>：	間隙及間隙向量的量測系統（英制/公制）（**選配**）		
	資料類型：	BOOL	
	值：	FALSE （預設）	量測系統對應於 G 群組 13 目前啟用的 G 指令（G70、G71、G700、G710)。
		TRUE	量測系統對應於設定基本系統： MD52806 $MN_ISO_SCALING_SYSTEM

動態鍊的轉換 10

10.1 啟動轉換 (TRAFOON)

使用預先定義的 TRAFOON 程序，啟動以動態鏈定義的轉換。呼叫必須單獨存在於單節中。

說明

或者，也可以透過傳統的 NC 指令，啟動以動態鏈定義的轉換，例如 TRAORI 或 TRANSMIT。為達此目的，必須輸入不等於零的適當值至 $NT_TRAFO_INDEX 系統變數中。

有關 $NT_TRAFO_INDEX 的其他資訊，請參閱「系統變數參數手冊」。

語法

```
TRAFOON(<Trafoname>,<Diameter>,<k>)
```

意義

TRAFOON:	啟動以動態鏈定義的轉換之程序		
<Trafoname>：	轉換資料集的名稱		
	資料類型：	STRING	
	值域：	透過 $NK_NAME 定義的轉換資料集之所有名稱	
	附註： 轉換資料集的名稱必須是唯一。只能在 $NT_NAME 中發生一次。		
<Diameter>：	參考或工作直徑（僅適用 TRACYL）		
	資料類型：	REAL	
	數值必須 > 1。		
<k>：	定義溝槽側偏移量的使用（僅適用 TRACYL）。		
	資料類型：	BOOL	
	值：	FALSE	無溝槽側偏移量
		TRUE	含溝槽側偏移量
	對應於 TRACYL 的轉換類型 514（溝槽側偏移量可以程式設計）。若未指定 <k>，套用 $NT_CNTRL[<n>] 中位元 10 的參數化設定。		

範例

程式碼	註解
`TRAFOON["Trans_1"]`	以名稱 `Trans_1` 啟動轉換。

10.2 在機台測量之後，修改方向轉換 (CORRTRAFO)

對於利用動態鏈的方式定義方向轉換的機台，使用者可以使用預先定義的 CORRTRAFO 功能，以便在機台測量之後，修改機台動態模式中方向軸的偏移向量或方向向量。

語法

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode>
[, <No_Alarm>])
```

意義

CORRTRAFO:	功能呼叫

<Corr_Status>：	函數傳回值		
	資料類型：	INT	
	值：	0	功能執行時無錯誤。
		1	無轉換啟用。
		2	目前啟用的轉換並非方向轉換。
		3	啟用的方向轉換並非以動態鏈定義。
		10	<Corr_Index> 呼叫參數為負值。
		11	<Corr_Mode> 呼叫參數為負值。
		12	子鏈結節段的參考無效（<Corr_Index> 的個位數位置）。數值必須不得大於子鏈結中方向軸的編號。
		13	子鏈結方向軸的參考無效（<Corr_Index> 的個位數位置）。數值必須小於子鏈結中方向軸的編號。
		14	子鏈結的參考無效（<Corr_Index> 的十位數位置）。只允許數值 0 及 1（參考零件或刀具鏈）。若是 <Corr_Index> 所參考的子鏈結不存在，會產生此錯誤編號。
		15	節段中沒有修正元素參考包含 <Corr_Index> 的參數（$NT_CORR_ELEM_P 或 $NT_CORR_ELEM_T）。
		20	無效的修正模式（<Corr_Mode> 的個位數位置）。只允許數值 0 及 1。
		21	無效的修正模式（<Corr_Mode> 的十位數及 / 或百位數位置）。寫入軸方向時，只有個位數位置可以不等於零。
		30	<Corr_Mode> 的百位數位置無效。只允許數值 0 及 1。
		31	<Corr_Mode> 的千位數位置無效。只允許數值 0 及 1。
		40	用來當作軸方向的方向向量為零向量。此錯誤只有在 <Corr_Mode> 的千位數位置等於 0 時才會發生。若是此參數的千位數位置等於 1（停用最大修正的監控），零向量也可以寫入。
		41	對於偏移向量的修正，與現值的差異，至少在一個座標上大於設定資料 SD41610 $SN_CORR_TRAFO_LIN_MAX 所指定的最大值。

			<Corr_Vect> 參數將被一個錯誤向量覆寫。在處理因為警報而中止時，這也適用（請參閱 <No_Alarm> 參數）。 在修正值超過可允許臨界值的元件中，在決定的修正值與臨界值之間，錯誤向量會有修正符號的差異。 未超過其臨界值的元件內容為零。
		42	對於方向向量的修正，相較於目前方向的角度位移大於設定資料 SD41611 $SN_CORR_TRAFO_DIR_MAX 所指定的最大值。
		43	由於寫入權限遺失，試圖寫入系統變數遭拒。
<Corr_Vect>： 修正向量 修正向量的內容係由下列參數 **<Corr_Index>** 及 <Corr_Mode> 定義。 若 <Corr_Status> = 41，向量的內容將被覆寫（如上所示）。			
	資料類型：	REAL	
<Corr_Index**>**： 修正元件即將修改的節段 / 方向向量即將修改的方向軸索引			
	資料類型：	INT	
	<Corr_Index> 參數是以十進制編碼（個位數至十位數位置）：		
	個位數位置：	包含節段的索引，或子鏈結中的方向軸。	
	十位數位置：	請參閱子鏈結。	
		0x	工件鏈
		1x	刀具鏈

<Corr_Mode>：	修正模式		
	資料類型：	INT	
	<Corr_Index> 參數是以十進制編碼（個位數至千位數位置）：		
	個位數位置：	指定要修正的元件。	
		xxx0	修正線性偏移向量
		xxx1	修正方向軸的方向向量
	十位數位置：	指定 <Corr_Index> 的內容所參考的修正元件要如何修改。	
		xx0x	將修正向量立刻寫入修正元件。 可以使用此版本將修正元件立刻寫入，而不需要已知相關系統資料 ($NK_OFF_DIR[<n>, ...]) 的索引 <n>。
		xx1x	與 0 相同，差別在於傳輸的修正值在世界座標中解譯。 版本 0 與 1 之間的差異，在動態鏈處於包含其他旋轉的初始狀態時（所有方向軸的位置等於 0），絕對會發生。
		xx2x	與 1 相同，差別在於修正值參考整個節段，亦即，輸入數值至修正元件，使得整個節段達到修正值定義的長度。
		附註： 寫入方向軸的方向向量時，不允許使用數值 1 及 2。	
	百位數位置：	指定 <Corr_Vect> 參數的內容將如何編譯。	
		x0xx	傳輸的修正向量 <Corr_Vect> 包含修正元件的整個新長度，或是 <Corr_Index> 搭配參考 <Corr_Mode> 十位數位置的節段（絕對修正）。
		x1xx	傳輸的修正向量 <Corr_Vect> 只包含相較於修正元件目前長度的差異，或是 <Corr_Index> 搭配參考 <Corr_Mode> 十位數位置的節段（增量修正）。
		附註： 對於方向軸的方向向量修正，百位數位置的內容必須為 0。	
	千位數位置：	指定修正是否必須受到下列最大值的限制： • SD41610 $SN_CORR_TRAFO_LIN_MAX	

		或 • SD41611 $SN_CORR_TRAFO_DIR_MAX	
		0xxx	啟用最大修正的監控。
		1xxx	不啟用最大修正的監控。
<No_Alarm>：	在發生錯誤狀況時的行為（傳回值 **>** 0）（**選配**）		
	資料類型：	BOOL	
	值：	FALSE（預設值）	在錯誤的狀況中，停止程式處理，並顯示警報 14103。
		TRUE	在錯誤的狀況中，不會停止程式處理，也不顯示警報。 應用：對應於傳回值的使用者專屬反應

說明

在呼叫功能時發生錯誤狀況中，或是輸出警報或是傳回錯誤編號（請參閱 **<No_Alarm>** 參數），使得使用者可以適當方式回應錯誤狀態。錯誤的肇因可以透過警報參數更詳細說明。傳回錯誤編號以取代警報，與警報參數完全相同。

CORRTRAFO 的其他資訊

含方向轉換的機台之動態結構係由一個或兩個動態鏈（子鏈結）加以說明，從世界座標系統的零點開始。兩個鏈結之一的**刀具鏈**在刀具的參考點結束，另一個鏈結的**工件鏈**則在基本座標系統的零點結束。

CORRTRAFO 功能測量含方向轉換機台上的支桿臂長度及軸方向，並將這些數值寫入特殊的修正元件中。例如，以透過 $ NK_TYPE 定義之類型 OFFSET 的元件，說明動態鏈。

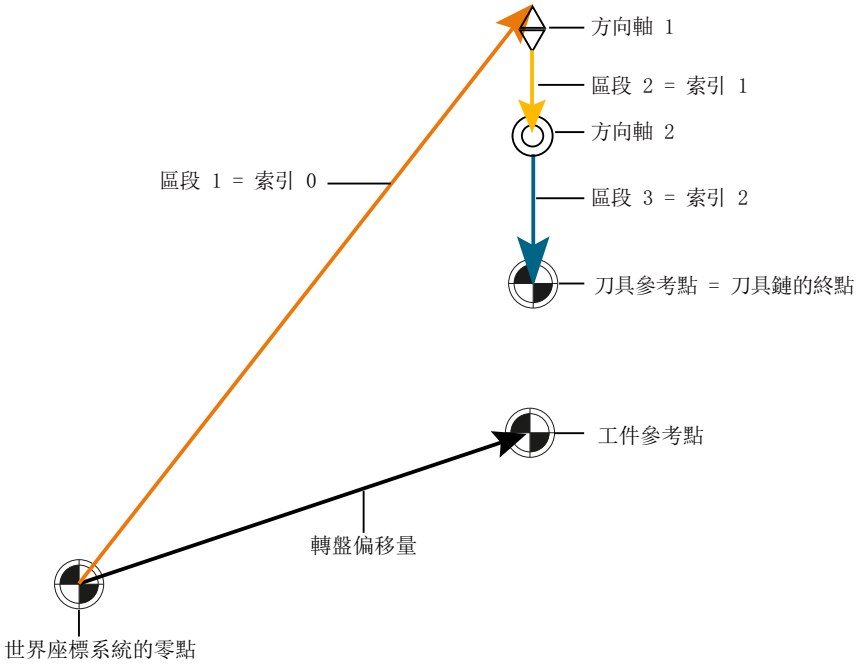
CORRTRAFO 搭配節段一起使用

兩個子鏈結每個最多可以分割成四個節段：

- 節段 1 從鏈結的起點開始，在第一方向軸結束。
- 節段 2 是方向軸 1 與方向軸 2 之間的節段。
- 節段 3 是方向軸 2 與方向軸 3 之間的節段。
- 節段 4 是方向軸 3 與刀具鏈或工件鏈結束之間的節段。

每個節段會包含類型 OFFSET 或 ROT_CONST 的常數鏈結元件。

下圖顯示 2 個方向軸的方向轉換。



圖像 10-1　CORRTRAFO 範例

節段已清楚定義：若您從起點至終點執行過動態子鏈結，第一個節段有索引 0，下一個索引 1，依此類推。然後，最後一個節段的索引永遠等於方向軸的編號。

修正元件

常數動態鏈元件（類型 $NK_TYPE[<n>] = "OFFSET" 的鏈結元件）可以產生參照，在每個節段中含有 $NT_CORR_ELEM_T[<n>, 0...3] 或 $NT_CORR_ELEM_P[<n>, 0 ... 3] 系統變數。在機台測量期間決定的修正值，則寫入這些含有 CORRTRAFO 功能的元件中。

轉換索引 = 1 的範例：

- $NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"；節段 1 中 C 軸（方向軸 1）的偏移量定義為修正元件。
- $NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"；節段 2 中 B 軸（方向軸 2）的偏移量定義為修正元件。
- $NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"；來自節段 3 中刀具參考點的 B 軸偏移量定義為修正元件

$NT_CORR_ELEM_T/P[<n>, 0...3] 中的參考順序必須對應至上述的節段，亦即在第一方向軸之前的 $NT_CORR_ELEM_T/P [<n>, 0] 中只能有一個鏈結元件等等。

CORRTRAFO 功能藉由將測量機台決定的數值，依此定義方式寫入修正元件。修正值的修改係透過 <Corr_Mode> 參數在 CORRTRAFO 中定義。

關閉一個鏈結

若設定 $NT_CNTRL[<n>] 系統變數中的位元 7 或位元 8，建立從鏈結的終點至機台零點的其他常數鏈結元件會自動內部插入至工件鏈的結尾（位元 7），或是插入至刀具鏈起點之前（位元 8）（「封閉鏈結」）。

這些自動插入的元件無法外部寫入，僅能讀取（請參閱 $AC_TRAFO_CORR_ELEM_P/T 系統變數）。

關閉刀具鏈的點

若是 $NT_CLOSE_CHAIN_T 系統變數並非空白，刀具鏈在鏈結的終點不會關閉，而是會在指定的鏈結元件終點關閉。在此點之後的其他鏈結元件會在啟動轉換時，導致對應的工作偏移量。

方向軸的索引

除了方向軸之間的常數偏移量以外，方向軸的方向向量也可以用 CORRTRAFO 功能寫入。方向軸的索引可以是動態子鏈結從原點執行至結束所得到的結果之索引，其中的計數從零開始。因此，方向軸的索引永遠與前一個節段的索引相同。

方向軸的索引也可以使用 $AC_TRAFO_ORIAX_LOC 系統變數決定。

鏈結元件的最大允許變更

鏈結元件的最大允許變更可以用兩個設定資料加以限制，亦即偏移向量的 SD41610 $SN_CORR_TRAFO_LIN_MAX 及方向軸的方向向量之 SD41611 $SN_CORR_TRAFO_DIR_MAX。SD41610 $SN_CORR_TRAFO_LIN_MAX 指定最大量，其中每個各別的向量元件可以對應於其參考值進行變更。SD41611 $SN_CORR_TRAFO_DIR_MAX 指定最大角度，其中軸向量的方向可以對應於其參考值進行變更。所謂的參考值永遠是在呼叫 CORRTRAFO 時啟用的轉換中，其所啟用的對應值。也就是說，動態資料的變更內容，有可能在轉換啟動之後，對 CORRTRAFO 功能的操作方法毫無影響。

刀具偏移量 11

11.1 偏移記憶體

偏移記憶體的結構

可使用 T 及 D 編號呼叫（「Flat D 編號」除外）每個資料欄位；除刀具的幾何資料外，也包含其他資訊，例如，刀具類型。

Flat D 編號結構

若在 NC，則使用「Flat D 編號結構」。在此情況下，使用未指派至工具且對應的刀具補正單節，產生 D 編號。

可繼續在工件程式中程式設計 T。然而，T 不會參考已程式設計的 D 編號。

使用者刀刃資料

可透過機械資料設定使用者刀刃資料。請參閱機台製造商說明。

刀具參數

說明

偏移記憶體中的各值

可透過系統變數，由程式讀取及寫入偏移記憶體 P1 到 P25 的各值。保留其他所有參數。

刀具參數$TC_DP6 到$TC_DP8、$TC_DP10 及$TC_DP11，以及$TC_DP15 到 $TC_DP17、$TC_DP19 及$TC_DP20 依刀具類型而有其他意義。

刀具參數編號（DP）	系統變數的意義	註解
$TC_DP1	刀具類型	概觀請參閱清單
$TC_DP2	刀刃位置	僅用於車刀
幾何	**長度補正**	
$TC_DP3	長度 1	配置方式
$TC_DP4	長度 2	類型及層級
$TC_DP5	長度 3	
幾何	**半徑**	

刀具參數編號（DP）	系統變數的意義	註解
$TC_DP6 [1] $TC_DP6 [2]	半徑 1 / 長度 1 直徑 d	銑削 / 車削 / 研磨刀具 切槽鋸
$TC_DP7 [1] $TC_DP7 [2]	長度 2 / 轉角半徑，錐形銑削刀具 槽寬度 b 轉角半徑	銑削刀具 切槽鋸
$TC_DP8 [1] $TC_DP8 [2]	銑削刀具的倒圓角半徑 1 投影長度 k	銑削刀具 切槽鋸
$TC_DP9 [1] [3]	倒圓角半徑 2	保留
$TC_DP10 [1]	角度 1 刀具端面	錐形銑削刀具
$TC_DP11 [1]	角度 2 刀具縱向軸	錐形銑削刀具
磨損	**長度及半徑補正**	
$TC_DP12	長度 1	
$TC_DP13	長度 2	
$TC_DP14	長度 3	
$TC_DP15 [1] $TC_DP15 [2]	半徑 1 / 長度 1 直徑 d	銑削 / 車削 / 研磨刀具 切槽鋸
$TC_DP16 [1] $TC_DP16 [3]	長度 2 / 轉角半徑、錐形銑削刀具、槽寬度 b 轉 角半徑	銑削刀具 切槽鋸
$TC_DP17 [1] $TC_DP17 [2]	銑削刀具的倒圓角半徑 1 投影長度 k	銑削 / 3D 面銑削 切槽鋸
$TC_DP18 [1] [3]	倒圓角半徑 2	保留
$TC_DP19[1]	角度 1 刀具端面	錐形銑削刀具
$TC_DP20[1]	角度 2 刀具縱向軸	錐形銑削刀具
刀具基本尺寸 / 轉接頭	**長度偏移**	
$TC_DP21	長度 1	
$TC_DP22	長度 2	
$TC_DP23	長度 3	
技術		
$TC_DP24	間隙角度	僅用於車刀
$TC_DP25		保留

[1] 亦適用於 3D 面銑削的銑削刀具

[2] 可用於切槽鋸刀具類型

[3] 已保留（SINUMERIK 840D sl 未使用）

註解

有多個可供幾何變數使用的輸入元件（例如，長度 1 或半徑）。相加以產生數值（例如，全長 1、總半徑），然後使用該值計算。

無需的偏移值需指派為零。

刀具參數$TC-DP1 到$TC-DP23，搭配輪廓刀具

說明

未列於表中的刀具參數，例如$TC_DP7，將不予評估，即其內容無意義。

刀具參數編號（DP）	含義	切削 Dn		註解
$TC_DP1	刀具類型			400 到 599
$TC_DP2	刀刃位置			
幾何	**長度補正**			
$TC_DP3	長度 1			
$TC_DP4	長度 2			
$TC_DP5	長度 3			
幾何	**半徑**			
$TC_DP6	半徑			
幾何	**角度限制**			
$TC_DP10	最小角度限制			
$TC_DP11	最大角度限制			
磨損	**長度及半徑補正**			
$TC_DP12	磨損長度 1			
$TC_DP13	磨損長度 2			
$TC_DP14	磨損長度 3			
$TC_DP15	磨損半徑			
磨損	**角度限制**			
$TC_DP19	最小磨損角度限制			
$TC_DP20	最大磨損角度限制			
刀具基本尺寸 / 轉接頭	**長度偏移**			

刀具參數編號（DP）	含義	切削 Dn		註解
$TC_DP21	長度 1			
$TC_DP22	長度 2			
$TC_DP23	長度 3			

基本值及磨損值

每個結果值皆為基本值及磨損值的總合，（例如，半徑為$TC_DP6 + $TC_DP15）。第一個刀刃的刀長也會加入基本量測（$TC_DP21 – $TC_DP23）。其他所有參數可能會影響標準刀具的有效刀長，也會影響該刀長（轉接頭、方向性刀把、設定資料）。

角度限制 1 及 2

角度限制 1 及 2 各與圓心參考點的刀刃中心點向量相關，並依順時針計算。

11.2 附加偏移

11.2.1 選擇附加偏移（DL）

附加偏移可視為程序偏移，可在加工中程式設計該偏移。 參考自刀刃的幾何資料，因此為刀具切削資料的元件。

使用 DL 編號定址附加偏移的資料（DL： 位置相關；關於使用位置的偏移），並透過使用者介面輸入。

應用

因使用位置而造成的尺寸錯誤可用附加偏移補償。

語法

```
DL=<編號>
```

含義

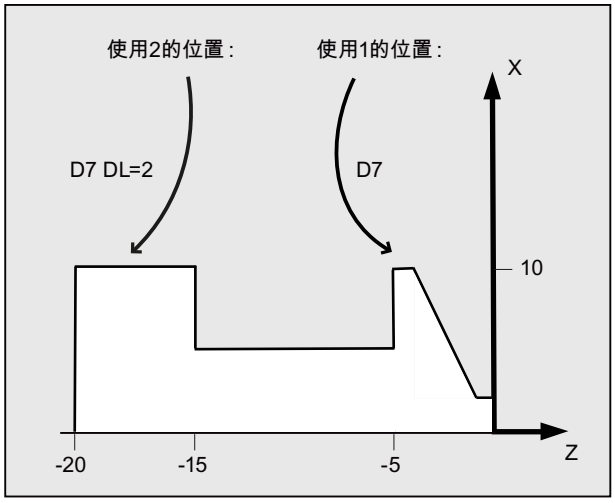
DL：	啟動附加偏移的指令
<編號>：	使用 <編號> 參數指定欲啟動的附加刀具偏移量資料

說明

使用機械資料定義附加偏移的數量，並加以啟動（→請小心觀察機台 OEM'的資料！）。

範例

兩軸承座使用相同的刀刃：



程式碼	註解
N110 T7 D7	；快轉區所在位置為 7. D7，並啟動 DL=1，且移動通過下一個單節。
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	；除 D7 外也啟動 DL=2，並移動通過下一個單節。
N150 G1 Z-21	
N160 G0 X200 Z200	；逼近換刀點。
...	

11.2.2 指定磨損及設定值（$TC_SCPxy[t，d]，$TC_ECPxy[t，d]）

可使用系統變數讀取及寫入磨損及設定值。 以刀具及刀鼻的對應系統變數邏輯為其邏輯基準。

系統變數

$TC_SCPxy[<t>,<d>]：	透過 xy 指派磨損值給特定的幾何參數，x 與磨損值的數量相同，而 y 則建立幾何參數的參考。
$TC_ECPxy[<t>,<d>]：	透過 xy 指派設定值給特定的幾何參數，x 與設定值的數量相同，而 y 則建立幾何參數的參考。
<t>： 刀具的 T 編號 <d>： 刀具刀刃的 D 編號	

說明

將定義的磨損及設定值加入至幾何參數及其他偏移參數中（D 編號）。

範例

長度 1 的磨損值針對刀具 **<t>** 的刀刃 **<d>** 設定為 1.0。

參數： **$TC_DP3**（長度 1，搭配車刀）

磨損值： **$TC_SCP13** 到**$TC_SCP63**

設定值： **$TC_ECP13** 到**$TC_ECP63**

$TC_SCP43 [<t>,<d>] = 1.0

11.2.3 刪除附加偏移（DELDL）

DELDL 指令會刪除刀具刀刃的附加偏移（以釋出記憶體空間）。 定義的磨損值及設定值均刪除。

語法

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<狀態>=DELDL[<t>,<d>]
```

含義

DELDL：	刪除附加偏移的指令
<t>：	刀具的 T 編號
<d>：	刀具刀刃的 D 編號
DELDL[<t>,<d>]：	刪除刀具 <t> 刀刃 <d> 的所有附加偏移。
DELDL[<t>]：	刪除刀具 <t> 所有刀刃的全部附加偏移。
DELDL：	刪除 TO 元件所有刀具之所有刀刃的全部附加偏移（在已程式設計指令的通道中）。

<狀態>：	刪除狀態	
	值：	含義：
	0	刪除已順利完成。
	-	偏移未刪除（若參數設定正好指定一個刀具稜邊），或未完全刪除（若參數設定指定多個刀刃）。

說明

無法刪除啟用刀具的磨損及設定值（基本上與 D 或刀具資料的刪除行為相同）。

11.3 刀具補正的特殊處理方式

使用設定資料 SD42900 到 SD42960 控制刀長及磨損的符號評估。

進行幾何軸鏡像或變更加工平面，及依刀具方向補正溫度時，同樣適用於磨損元件的行為。

磨損值：

若參考下述情形中的磨損值，則應視為實際磨損值的總和（$TC_DP12 到$TC_DP20），及含磨損值的總和偏移（$SCPX3 到$SCPX11），以及設定值（$ECPX3 到$ECPX11）。

更多關於總和偏移的資訊，請參閱：

參考資料：

功能手冊，刀具管理

設定資料

SD42900 $SC_MIRROR_TOOL_LENGTH	刀長元件的鏡像及刀具基準尺寸的元件。
SD42910 $SC_MIRROR_TOOL_WEAR	刀長元件磨損值的鏡像。
SD42920 $SC_WEAR_SIGN_CUTPOS	將磨損值元件的符號評估為刀刃位置的函數。
SD42930 $SC_WEAR_SIGN	反轉磨損尺寸的符號。
SD42935 $SC_WEAR_TRANSFORM	磨損值的轉換。
SD42940 $SC_TOOL_LENGTH_CONST	將刀長元件指派給幾何軸。
SD42950 $SC_TOOL_LENGTH_TYPE	獨立於刀具類型之刀長元件指派。
SD42960 $SC_TOOL_TEMP_COMP	刀具方向的溫度補正值。程式設計刀具方向時，也會運作。

參考資料

功能手冊基本功能；刀具偏移量 (W1)

其他資訊

啟用修改的設定資料

當上述說明的設定資料已修改，需到下次選擇刀具稜邊時，才會重新評估刀具元件。 若已啟用刀具，且欲重新評估該刀具的資料，則需再次選擇刀具。

該原則亦適用於，因軸鏡像狀態變更，而修改結果刀長的情況。 需在鏡像指令後再次選擇刀具，才能啟動修改的刀長元件。

可定向刀把及新的設定資料

設定資料 SD42900 到 SD42940 對啟用含方向功能的刀把元件無任何影響。 然而，以可定向刀把計算，可得出刀具及其總結果長度（刀長+磨損+刀具基準尺寸）。 以設定資料啟動的所有修改，皆包含在結果總長度的計算中，即可定向刀把的向量獨立於加工平面。

說明

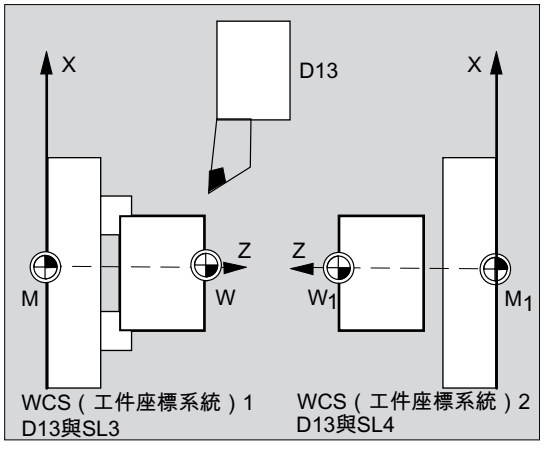
使用可定向刀把時，為非鏡像基本系統定義所有刀具非常實用，即使是僅用於鏡像加工的刀具。 以鏡像軸加工時，刀把會旋轉，以使刀具的實際位置被正確地說明。 然後所有刀長元件會自動依正確方向動作，依各個軸的鏡像狀態而定，不需透過設定資料控制各個元件評估。

其他應用程式選項

若機台上無車刀的實體選項，即使已永久安裝不同方向的刀具，可定向刀把功能的使用也可以很實用。 可在基本方向中一起執行刀具尺寸標註，根據虛擬刀把的旋轉計算與加工相關的尺寸。

11.3.1 刀長的鏡像

當設定資料 SD42900 $SC_MIRROR_TOOL_LENGTH 及 SD42910 $SC_MIRROR_TOOL_WEAR 未設定為零點時，即可進行刀長元件及含磨損值與其相關軸的基本尺寸元件之鏡像。



SD42900 $SC_MIRROR_TOOL_LENGTH

設定資料**不等於**零點：

該刀長元件（$TC_DP3、$TC_DP4 及$TC_DP5）及基本尺寸的元件（$TC_DP21、$TC_DP22 及$TC_DP23）是對相關軸進行鏡像，也由反轉符號進行鏡像。

磨損值**不會**進行鏡像。 若欲進行鏡像該值，則需加以設定設定資料 SD42910 $SC_MIRROR_TOOL_WEAR。

SD42910 $SC_MIRROR_TOOL_WEAR

設定資料**不等於**零點：

刀長元件的磨損值－其關聯軸會進行鏡像－也由反轉符號進行鏡像。

11.3.2 磨損符號評估

當設定資料 SD42920 $SC_WEAR_SIGN_CUTPOS 及 SD42930 $SC_WEAR_SIGN 未設定為零點時，即可反轉磨損元件的符號評估。

SD42920 $SC_WEAR_SIGN_CUTPOS

設定資料**不等於**零點：

對於含相關刀刃位置的刀具（車削及研磨刀具，刀具類型 400），此時加工平面中磨損元件的符號評估視刀刃位置而定。 該設定資料對於不含相關刀刃位置的刀具類型無意義。

在下表中，用 SD42920（不等於零點）反轉的尺寸符號係使用 X 設計：

刀刃位置	長度 1	長度 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

說明

使用 SD42920 及 SD42910 的符號評估相互獨立。 例如若使用兩個設定資料變更尺寸的符號，則結果符號維持不變。

SD42930 $SC_WEAR_SIGN

設定資料**不等於**零點：

反轉所有磨損尺寸的符號。 此會影響刀長及其他變數，例如刀具半徑、倒圓角半徑等等。

若輸入正號磨損尺寸，刀具則變為「較短」且「較細」，請參閱「刀具偏移量，特殊處理，啟動變更的設定資料」一章。

11.3.3 生效加工操作的座標系統（TOWSTD、TOWMCS、TOWWCS、TOWBCS、TOWTCS、TOWKCS）

依機台的動力學或可定向刀盤的可用性而定，將在座標系統之一中量測的磨損值轉換成適當的座標系統。

主動加工操作的座標系統

使用下列會產生刀長偏移量的座標系統，將刀長磨損元件，透過相對應群組 56 的 G 指令，結合至生效刀具中：

- 機械座標系統 (MCS)
- 基本座標系統 (BCS)
- 工件座標系統 (WCS)
- 刀具座標系統 (TCS)
- 動態轉換的刀具座標系統 (KCS)

語法

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

意義

TOWSTD：	刀長磨損值中偏移的初始設定值
TOWMCS：	機械座標系統（MCS）中刀長偏移
TOWWCS：	工件座標系統（WCS）中刀長偏移
TOWBCS：	基本座標系統（BCS）中刀長偏移

TOWTCS：	刀盤參考點上的刀長偏移（可定向刀盤）
TOWKCS：	刀具頭上的刀長補正（動態轉換）

其他資訊

區分特性

最重要的區分特性顯示於下表中：

G 指令	磨損值	啟用的可定向刀盤
TOWSTD：	初始值，刀長	磨損值受制於旋轉。
TOWMCS：	機械座標系統（MCS）中的磨損值。若未啟用可定向的刀盤，則 TOWMCS 與 TOWSTD 完全相同。	僅旋轉在不計入磨損下所得出的刀長向量。
TOWWCS：	磨損值在工件座標系統（WCS）中轉換成機械座標系統（MCS）。	在不計入磨損下如 TOWMCS 般計算刀具向量。
TOWBCS：	磨損值會在基準座標系統（BCS）中轉換成機械座標系統（MCS）。	在不計入磨損下如 TOWMCS 般計算刀具向量。
TOWTCS：	磨損值在工件座標系統中轉換成機械座標系統（MCS）。	在不計入磨損下如 TOWMCS 般計算刀具向量。

TOWWCS、TOWBCS、TOWTCS：將磨損向量加入刀具向量中。

線性轉換

僅在由 BCS 線性轉換產生機械座標系統（MCS）時，在機械座標系統（MCS）中定義刀長才有意義。

非線性轉換

例如，若以 TRANSMIT 啟用非線性轉換，則指定機械座標系統（MCS）為所求座標系統時，即自動使用 BCS。

無動態轉換及可定向刀盤

若無啟用動態轉換及可定向刀盤，則會結合其他四個座標系統（工件座標系統（WCS）除外）。所以只有工件座標系統（WCS）與其他系統不同。由於只需評估刀長，座標系統間的轉譯互不相關。

參考資料：
如需有關刀具補正的更多資訊，請參閱：
功能手冊基本功能；刀具偏移量 (W1)

計算中包含磨損值

設定資料 **SD42935 $SC_WEAR_TRANSFORM** 定義三個磨損元件之一：

- 磨損
- 總偏移微調
- 總偏移粗調

若已啟用下列 G 指令之一，則應受制於使用轉接頭轉換或可定向的刀盤所進行的旋轉：

- TOWSTD
  基本位置。刀長中的改正。
- TOWMCS
  機械座標系統 (MCS) 中之磨耗值。
- TOWWCS
  工件座標系統 (WCS) 中之磨耗值。
- TOWBCS
  基本座標系統 (BCS) 中之磨損值。
- TOWTCS
  在刀盤夾具（T 刀盤參考）刀具座標系統中的磨損值。
- TOWKCS
  動態轉換刀具頭座標系統中的磨損值

說明

各個磨損元件的評估（指派給幾何軸，符號評估）受以下因素的影響：
- 啟用的平面
- 接頭轉換
- 設定資料：
  - 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 刀長及平面變更

設定資料 SD42940 $SC_TOOL_LENGTH_CONST 未設定等於零點時，即可在變更平面時，指派刀長元件－例如，長度、磨損及基本尺寸－至車削及研磨刀具的幾何軸。

SD42940 $SC_TOOL_LENGTH_CONST

設定資料**不等於**零點：

變更加工平面時，不會變更指派刀長元件（長度、磨損及刀具基本尺寸）至幾何軸（G17 - G19）。

下表顯示指派給車削及研磨刀具（刀具類型 400 到 599）幾何軸的刀長元件：

內容	長度 1	長度 2	長度 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*）不等於零也不等於列出的六個值，會評估為值 18。

下表顯示指派給其他刀具（刀具類型 < 400 或 > 599）幾何軸的刀長元件：

操作平面	長度 1	長度 2	長度 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*）不等於零也不等於列出的六個值，會評估為值 17。

說明

表中代表的意義為，假設最多用 X、Y、Z 指定 3 個幾何軸。以軸順序而非軸識別碼決定補正與軸間的指派。

11.4 線上刀具偏移量

11.4.1 定義多項式函數 (FCTDEF)

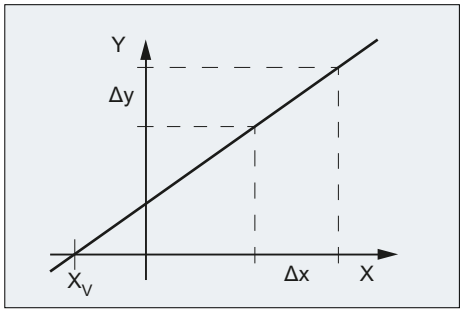
某些修飾策略（如修飾輪）的特色是其研磨輪半徑會隨修飾輪的進給連續（線性）縮小。此策略要求修飾輪的進給與各長度磨損值的寫入之間必須為線性函數。線性函數使用預先定義的程序 FCTDEF(...) 定義，最多到三分之一的多項式函數。

直線等式

$y = f(x) = a_0 + a_1 * x_1$

a_1：直線斜度，含 $a_1 = \Delta x / \Delta y$

a_0：沿 X 軸直線移動，含 $a_0 = -a1 * X_V$



語法

```
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
```

意義

FCTDEF(...)：	定義 PUTFTOCF(...) 的多項式函數： $y = f(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3$	
<Func>：	函數編號	
	資料類型：	INT
	值域：	1、2、3
<LLimit>：	下臨界值	
	資料類型：	REAL
<ULimit>：	上臨界值	
	資料類型：	REAL

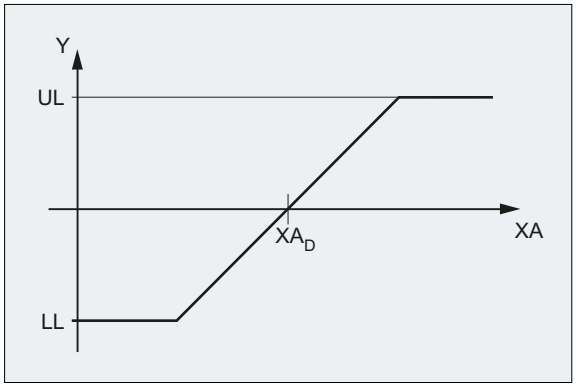
<a0>,<a1>,<a2>,<a3>：	多項式函數的係數	
	資料類型：	REAL

範例

定義

- 函數編號：1
- 下臨界與上臨界值：-100、100
- 特性的斜度：$a_1 = 1$
- 操作點應位於特性的中心。依據在 NC 程式中定義函數時 WCS 中的 XA 軸設定點位置，特性必須往負 Y 的方向移動：$a_0 = -a_1 * XA_D = -1 * \$AA_IW$
- $a_2 = a_3 = 0$

特性



UL　上臨界值

LL　下臨界值

XA_D　在 NC 程式中定義函數時 XA 軸的設定點

程式設計

程式碼	註解
`FCTDEF(1,-100,100,-$AA_IW[XA],1)`	；函數定義

11.4.2 持續寫入線上刀具偏移 (PUTFTOCF)

使用預先定義的程序 PUTFTOCF(...)，線上刀具偏移量會依照先前使用 FCTDEF(...) (頁 462) 定義的多項式函數執行。

說明

線上刀具偏移量也可用同步動作實現。
如需進一步資訊，請參閱功能手冊同步動作。

語法

```
PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)
```

意義

PUTFTOCF(...)：	使用 FCTDEF(...) 定義的多項式函數按單節順序持續寫入線上刀具偏移量	
<Func>：	函數編號，使用 FCTDEF(...) 定義於函數定義	
	資料類型：	INT
	值域：	1、2、3
<RefVal>：	從衍生偏移量（例如，軸設定點）得出的參考值。	
	資料類型：	VAR REAL
<ToolPar>：	加入偏移量至磨耗參數（長度 1、2 或 3）的號碼。	
	資料類型：	INT
<Chan>：	線上刀具偏移量發生影響的通道號碼。 **附註：** 只有偏移量沒有在有效通道中產生效果時需要。	
	資料類型：	INT
<Sp>：	線上刀具偏移量發生影響的主軸號碼。 **附註：** 只有偏移量要被套用至無效的研磨輪，而不是正在使用中的有效刀具時需要。	
	資料類型：	INT

11.4.3 分散寫入線上刀具偏移量 (PUTFTOC)

功能

使用預先定義的程序 PUTFTOC(...)，線上刀具偏移量會依照固定偏移值執行。

語法

```
PUTFTOC(<CorrVal>,<ToolPar>,<Chan>,<Sp>)
```

意義

PUTFTOC(...)：	寫入線上刀具偏移量	
<CorrVal>：	偏移值會新增磨耗參數中。	
	資料類型：	REAL
<ToolPar>：	加入偏移量至磨耗參數（長度 1、2 或 3）的號碼。	
	資料類型：	INT
<Chan>：	線上刀具偏移量發生影響的通道號碼。 **附註：** 只有偏移量沒有在有效通道中產生效果時需要。	
	資料類型：	INT
<Sp>：	線上刀具偏移量發生影響的主軸號碼。 **附註：** 只有偏移量要被套用至無效的研磨輪，而不是正在使用中的有效刀具時需要。	
	資料類型：	INT

11.4.4 啟用/停用線上刀具偏移量 (FTOCON/FTOCOF)

線上刀具偏移量可用 G 指令 FTOCON 和 FTOCOF 啟用或停用。

語法

```
FTOCON
...
```

```
FTOCOF
```

意義

FTOCON：	啟用線上刀具偏移量 指令必須程式設計至線上刀具偏移量要啟用的通道中。
FTOCOF：	停用線上刀具偏移量 指令必須程式設計至線上刀具偏移量要停用的通道中。 **附註：** 使用 FTOCOF 時，軸不會為刀具偏移量往外移動。但使用 PUTFTOC/PUTFTOCF 計算的值會保留在裁切專屬偏移量資料內。 若要停用線上刀具偏移量，刀具 (T...) 仍需要在 FTOCOF 之後，被選擇或取消選擇。

11.5 3D 刀具半徑補正

11.5.1 選擇 3D 刀具半徑補正進行 3D 圓周銑削（CUT3DC、CUT3DCD、ISD）

未將限制表面納入考量的 3D 圓周銑削之 3D 刀具半徑補正 (3D TRC) 係利用模式生效的 G 指令 CUT3DC 或 CUT3DCD 選擇。

實際的啟動係以 G41 或 G42 執行，刀具半徑補正是以 G40 停用。

語法

```
G41/G42 ORIC/ORID ISD=...CUT3DC/CUT3DCD CDOF2 X...Y...Z...
...
G40 X...Y...Z...
```

含義

CUT3DC：	圓周銑削的 3D TRC（只有在 5 軸轉換啟用時）	
CUT3DCD：	圓周銑削的 3D TRC 參考差別刀具（只有在 5 軸轉換啟用時） 半徑差異係由刀具參數 $TC_DP15 指定。	
G41/G42 X...Y...Z...：	啟動刀具半徑補正	
	G41：	刀具半徑補正輪廓左側
	G42：	刀具半徑補正輪廓右側
	附註： 啟動必須在線性區塊內執行 (G0/G1)。	
CDOF2：	停用 3D 周長銑削的碰撞偵測	
ORIC/ORID：	位於外角的方向變更行為係透過 G 指令 ORIC 及 ORID 指定。	
	ORIC：	在待插入之圓弧單節上外角處的方向變更是疊印的。
	ORID：	在待插入圓弧單節之前，執行外角處的方向變更。

ISD=<值>：	利用 ISD 位址，刀具可以變更圓周銑削的刀具插入深度，並**啟用** 3D 刀具半徑補正。	
	<值>：	插入深度的長度
G40 X...Y...Z...：	停用刀具半徑補正 **附註：** 停用必須在含有幾何軸移動的線性區塊內執行 (G0/G1)。	

說明

選擇 3D TRC 的 G 指令是在逼近單節中評估，亦即，通常是在含有 G41 或 G42 的單節中。

G41 或 G42 也可以程式設計在沒有移動與補正相關的幾何軸動作單節中。在此情況下，逼近單節是第一個移動單節，該單節緊跟在後。

含啟用刀具半徑補正的 3D TRC 版本變更將忽略，沒有警報。

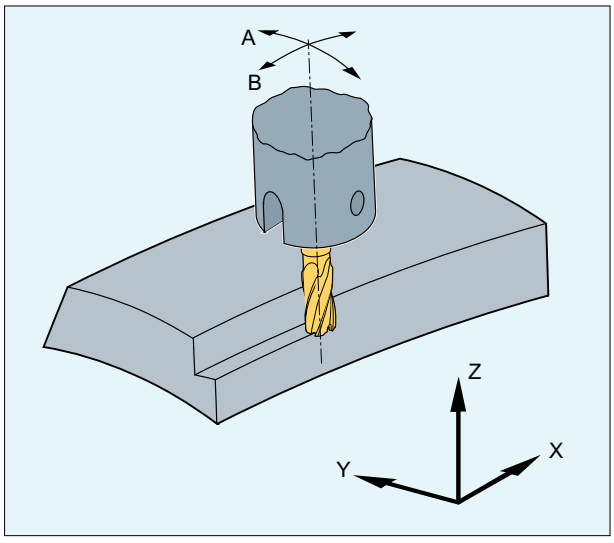
範例

程式碼	註解
	；刀具 D1 的定義：
$TC_DP1[1,1]=120	；類型（端面銑刀）
$TC_DP3[1,1]=20	；長度補正向量
$TC_DP6[1,1]=8	；半徑
N10 X0 Y0 Z0 T1 D1 F12000	；選擇刀具。
N20 TRAORI(1)	；啟動轉換。
N30 G42 ORIC ISD=10 **CUT3DC** G64 X30	；啟動 3D 圓周銑削，
	；位於外角的連續方向變更，
	；插入深度：10 mm
N40 ORIWKS A30 B15	；透過軸位置的規格，在轉角的方向變更。
N50 Y20 A3=1 C3=1	；含方向向量的移動單節，
	；含方向向量的方向規格。
N60 X50 Y30	；含常數方向的移動單節。
N70 Y50 A3=0.5 B3=1 C3=5	；含方向變更的移動單節。
N80 M63	；不含移動資訊的單節。
N90 X0 ISD=20	；含插入深度變更的移動單節。
N100 G40 Y0	；停用刀具半徑補正。
N110 M30	

其他資訊

路徑及方向

透過定義路徑（引導線）及相關的方向，進行用於此處的圓周銑削類型。在該類型加工中，與路徑上的刀具形狀無關。只有確定刀具切入點的半徑。



圖像 11-1　圓周銑削

逼近行為

刀具半徑補正的 3D 版本，其逼近行為永遠是 NORM。

外轉角的行為

使用與 2½D TRC 條件相同的方法，在外轉角處，評估含有 3D TRC 圓周銑削的群組 18 之 G 指令（角落行為、刀具偏移量）：

- G450：變化圓弧（刀具以圓弧路徑繞工件轉角移動）
  與 2½D TRC 解決方案不同的是，在外轉角處插入的輪廓元件永遠是 0 半徑的圓形，其中的刀具半徑補正以相同的方式作用在任何其他的程式設計路徑上。只能插入圓形，無法插入圓錐。在這種情況下，DISC 位置並不明顯，因此不納入評估。
- G451：等距路徑（刀具自工件轉角返回）的交點
  在與刀具方向垂直的平面中，於角落由延伸兩相關單節的偏移曲線，及定義兩單節的交點，來決定交點。

若是二個相關的移動單節之間，插入了至少一個含有刀具方向變更的單節，則不使用交點程序 (G451)。在此狀況下，轉角處永遠都是插入圓形。

在外部角落處的方向變更行為

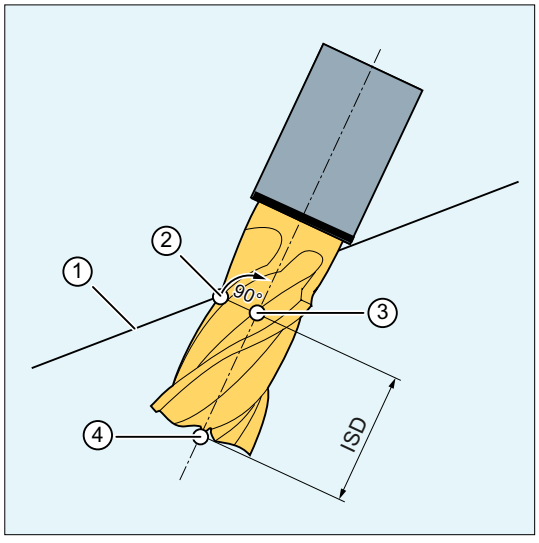
使用 ORIC 及 ORID G 指令，以決定兩個單節之間程式設計的方向變更所形成的轉角，會在進行插入圓形單節 (ORID) 之前處理，或同一時間 (ORIC) 時執行。

插入深度

銑削刀具的插入深度是從刀具尖端至銑削刀具參考點的距離。

銑削刀具參考點是銑削刀具在程式設計路徑上的加工點在刀具長軸上的垂直投影。

加工點在刀具錐形柱面上的位置是以插入深度設定。



① 程式設計路徑
② 銑削刀具加工點
③ 銑削刀具參考點
④ 銑削刀尖
ISD 插入深度（InSertion 深度）

圖像 11-2 插入深度

刀具半徑補正參考差別刀具

透過 CUT3DCD 指令，參考差別刀具，以啟動圓周銑削的 3D TRC。假如程式設計的輪廓參考標準刀具的圓心路徑，並使用差別刀具以外的刀具進行加工，則適用此設定。計算 3D 刀具半徑補正時，只有啟用中刀具 ($TC_DP15) 的半徑磨損值及任何程式設計的刀具偏移量 OFFN 和 TOFFR 會納入考量。啟用中刀具的基本半徑 ($TC_DP6) 並不列入考量。

為含 CUT3DC 之圓周銑削，以傾斜側壁進行口袋型銑削

在此 3D 刀具半徑補正中，銑削半徑的偏差，係由朝欲加工之表面法線向量的進給補償。若插入深度 ISD 維持不變，則銑削刀具平面所在的平面也維持不變。例如，用小於標準刀具半徑的銑削刀具無法到達口袋基部－此處亦為限制表面。對於自動刀具進給，控制系統

需知道限制表面，請參閱章節「3D 圓周銑削將限制表面考量在內 (CUT3DCC, CUT3DCCD) (頁 477)」。

Advanced Surface / Top Surface

說明

套用刀具半徑補正 CUT3DCD 並搭配「Advanced Surface」或「Top Surface」（需要授權）選項時，必須注意與「Advanced Surface」/「Top Surface」有關的設定建議。

檢查設定資料時，SIOS 入口網頁提供特殊的測試程式：

- Advanced Surface (https://support.industry.siemens.com/cs/ww/en/view/78956392) 的測試程式
- Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423) 的測試程式

11.5.2 選擇 3D 刀具半徑補正進行 3D 端面銑削（CUT3DF、CUT3DFS、CUT3DFF、CUT3DFD）

利用模式生效的 G 指令 CUT3DF、CUT3DFS、CUT3DFF 或 CUT3DFD，選擇 3D 端面銑削的 3D 刀具半徑補正 (3D TRC)。

實際的啟動係以 G41 或 G42 執行

使用 3D 端面銑削時，必須定義所要加工平面的表面法線向量，以便計算刀具半徑補正。這必須在含有 G41 或 G42 的單節中，透過 A4、B4、C4 及 A5、B5、C5 位址執行。

刀具半徑補正是以 G40 停用。

語法

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD
X...Y...Z...A4=...B4=...C4=...A5=...B5=...C5=...
...
G40 X...Y...Z...
```

含義

CUT3DFS：	含恆定方向之端面銑削的 3D TRC。以 G17 - G19 定義刀具方向，且不受框架影響。	
CUT3DFF：	含恆定方向之端面銑削的 3D TRC。刀具方向為以 G17 - G19 定義的方向，且在某些情況下，由框架旋轉。	
CUT3DF：	含方向變更之端面銑削的 3D TRC（僅使用啟用的 5 軸轉換）	
CUT3DFD：	含參考差別刀具方向變更的端面銑削之 3D TRC（僅使用啟用的 5 軸轉換） 半徑差異係由刀具參數 $TC_DP15 指定。 **附註**： CUT3DFD 只有在搭配「表面平滑化的 3D 端面銑削」時才有作用。這需要透過 CYCLE832(...) (頁 865) 呼叫「頂端表面加工」功能（需要授權）啟動。	
G41/G42 X...Y...Z...：	啟用刀具偏移量 使用 G41 及使用 G42 的行為與 3D 端面銑削完全相同。 **附註：** 啟動必須在線性區塊內執行 (G0/G1)。	
A4/5=...B4/5=...C4/5=...：	所要加工平面的表面法線向量之定義	
	A4=...B4=...C4=...：	單節起點的定義
	A5=...B5=...C5=...：	在單節結尾的定義
ORIC/ORID：	位於外角的方向變更行為係透過 G 指令 ORIC 及 ORID 指定。	
	ORIC：	在待插入之圓弧單節上外角處的方向變更是疊印的。
	ORID：	方向變更在圓弧單節之前進行。
G40 X...Y...Z...：	停用刀具半徑補正 **附註：** 停用必須在含有幾何軸移動的線性區塊內執行 (G0/G1)。	

說明

G41 或 G42 也可以程式設計在沒有移動與補正相關的幾何軸動作單節中。在此情況下，逼近單節是第一個移動單節，該單節緊跟在後。

含啟用刀具半徑補正的 3D TRC 版本變更將忽略，沒有警報。

範例

範例 1：使用 CUT3DF 的 3D 端面銑削

程式碼	備註
N10	；刀具 D1 的定義：
N20 $TC_DP1[1,1]=121	；刀具類型（環型銑刀）
N30 $TC_DP3[1,1]=20	；長度補正
N40 $TC_DP6[1,1]=5	；半徑
N50 $TC_DP7[1,1]=3	；倒圓角半徑
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	；選擇刀具。
N90 TRAORI(1)	；選擇方向轉換。
N100 B4=-1 C4=1	；平面的定義。
N110 **G41** ORID **CUT3DF** G64 X10 Y0 Z0	；啟用刀具偏移量。
N120 X30	
N130 Y20 A4=1 C4=1	；外轉角，新平面定義。
N140 B3=1 C3=5	；含 ORID 的方向變更。
N150 B3=1 C3=1	；含 ORID 的方向變更。
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	；含 ORIC 的方向變更。
N180 A3=-1 C3=1	；含 ORIC 的方向變更。
N190 Y-10 A4=-1 C4=3	；新平面定義。
N200 X-20 Y-20 Z10	；內轉角含前一單節。
N210 X-30 Y10 A4=1 C4=1	；內轉角，新平面定義。
N220 A3=1 B3=0.5 C3=1.7	；含 ORIC 的方向變更。
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	；方向的變更。
N250 X0 Y30 C4=1	；路徑移動，新平面，
	；方向含相對的程式設計。
N260 BSPLINE X20 Z15	；曲線開始，相對的方向程式設計
N270 X30 Y25 Z18	；曲線過程中維持生效。
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	；螺旋線，方向含軸程式設計。

程式碼	備註
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	；路徑移動，方向，非常數平面。
N330 X-20 G40	；停用刀具半徑補正。
N340 M30	

範例 2：使用 CUT3DFD 從 CAD 系統產生 NC 程式（節段）

程式碼	註解
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
；逼近將刀具長度納入考量的 MCS 起點位置。	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
；在 MCS 中的定位結束。	
；	
TRAORI(1)	；選擇方向轉換。
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	；呼叫 CYCLE832，其中： ；輪廓允差 = 0.01 mm， ；處理類型：平滑化的頂端表面加工， ；精加工，有方向允差輸入， ；方向允差 = 1 度
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	；單節 N09 至 N10 是以恆定方向快速逼近移動至工件的部分。
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	；在單節 N11 至 N21 中，是進行緩慢逼近移動的部分。刀具已經接近工件；此外，CYCLE832 的模具製造設定（例如 COMPSURF）現在啟用（由於啟用 G01 的緣故）。此所謂模具製造行為的暫態階段之路徑應該是輪廓允差的大約 1000 倍（本範例為 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	

程式碼	註解
N19 X-248.01282 Y30.84230 Z11.58085 A3=-.216336015 B3=.934194446 C3=-.283689030	
N20 X-248.03357 Y30.65006 Z10.72910 A3=-.216233089 B3=.933833626 C3=-.284952647	
N21 X-248.05086 Y30.48986 Z10.01931 A5=-.060687572 B5=.974940255 C5=-.214029243 A3=-.215712821 B3=.932005189 C3=-.291263295	
N22 **G41** X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	；從 N22，將表面法線向量初次完全定義在整個單節上（亦即，前一個單節 N21 結尾的表面法線向量仍存在，因此單節 N22 開始的表面法線向量，及單節 N22 結尾的表面法線向量繼續沿用）。因此滿足了使用 G41/G42 切換刀具偏移量的先決條件。
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	
...	

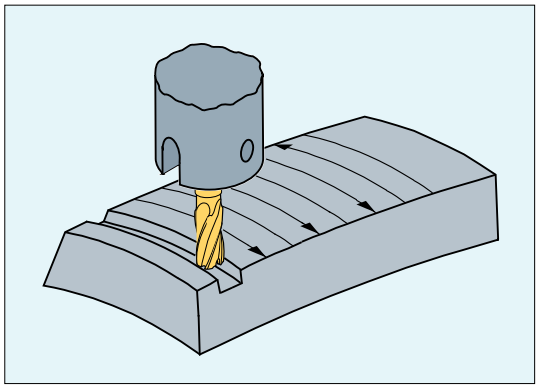
說明

對於使用 CUT3DFD 的 3D 端面銑削，需要表面法線向量的定義，以便利用 G41/G42 啟動刀具偏移量。沒有表面法線向量定義的 G41/G42 程式設計，將導致輸出警報。

其他資訊

3D 端面銑削

該類型的 3D 銑削，在工件表面上需有 3D 路徑逐行說明。計算需考慮刀具形狀及規格，其一般是在 CAM 中執行計算。後處理器寫入工件程式（除 NC 單節外）刀具方向（啟用的 5 軸轉換）及所需 3D 刀具偏移量的 G 指令。代表機台操作員可使用較小一點的刀具，脫離用來計算 NC 路徑的刀具。



圖像 11-3　端面銑削

逼近行為

刀具半徑補正的 3D 版本，其逼近行為永遠是 NORM。

外轉角的行為

外轉角可以用半徑 0 的端面銑削加以處理，其中圓形平面從第一個單節的結尾正切延伸至第二個單節的起始正切。以此方式，方向可以在單節變化期間變更。因此，永遠會在外轉角插入一個圓弧做為輪廓元件。端面銑削並不提供交點程序。

在外部角落處的方向變更行為

使用 ORIC 及 ORID G 指令，以決定兩個單節之間程式設計的方向變更所形成的轉角，會在進行插入圓形單節 (ORID) 之前處理，或同一時間 (ORIC) 時執行。

刀具半徑補正參考差別刀具

3D 刀具半徑補正，參考差別刀具，係使用 CUT3DFD 指令選擇。假如程式設計的輪廓參考標準刀具的圓心路徑，並使用差別刀具以外的刀具進行加工，則適用此設定。計算 3D 刀具半徑補正時，只有啟用中刀具 ($TC_DP15) 的半徑磨損值及任何程式設計的刀具偏移量 OFFN 和 TOFFR 會納入考量。啟用中刀具的基本半徑 ($TC_DP6) 並不列入考量。

使用 CUT3DFD 進行 3D 端面銑削，只有在搭配「表面平滑化的 3D 端面銑削」時才有作用。這需要透過 CYCLE832(...) (頁 865) 呼叫「Top Surface」功能（需要授權）啟動。啟動必須在刀具偏移量以 G41/G42 啟用之前發生；不是直接在刀具介入之前，在其之前應有一個路徑長度，對應於輪廓允差的大約 1000 倍（例如 1000 x 0.01 mm = 10 mm）。停用必須以反向順序執行：首先利用 G40 關閉刀具偏移量，接著在路徑長度之後，以例如 CUT2D（或類似）停用，其對應於大約輪廓允差的 1000 倍。

為了能夠使用「3D 端面銑削的平滑化表面法線向量」，也必須啟用「透過多項式插補表面法線向量」功能：

MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

說明

對於使用 CUT3DFD 的 3D 端面銑削，並搭配「Top Surface」，必須注意與「Top Surface」相關的設定建議。

檢查設定資料時，SIOS 入口網頁提供 (https://support.industry.siemens.com/cs/ww/en/view/109738423)特殊的測試程式。

11.5.3 3D 圓周銑削將限制表面考量在內 (CUT3DCC, CUT3DCCD)

在刀具方向裡有一持續或經常變更的 3D 圓周銑削中，經常依已定義的標準刀具程式設計刀具中心點路徑。由於在實務上經常無適當的標準刀具可用，故可使用與標準刀具偏差不大的刀具 (≤ 5%)。

CUT3DCCD 將程式設計標準刀具所定義，有實際差距的刀具限制表面考慮在內。NC 程式會定義標準刀具的中心點路徑。

CUT3DCC 使用圓柱刀具時，將程式設計標準刀具可能到達的限制表面考慮在內。NC 程式定義加工表面的輪廓。

限制表面的表面法線向量以 A4、B4、C4 及 A5、B5、C5 指定 3D 端面銑削。

語法

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X...Y...Z...A4=...B4=...C4=...A5=...B5=...C5=...
...
G40 X...Y...Z...
```

意義

CUT3DCCD：	將限制表面納入考量的圓周銑削之 3D TRC，使用刀具中心點路徑上的差別刀具：進給至限制表面
CUT3DCC：	將限制表面納入考量的圓周銑削之 3D TRC，使用 3D 半徑補正：加工表面的輪廓

G41/G42 X...Y...Z...：	啟動刀具半徑補正	
	G41：	刀具半徑補正輪廓左側
	G42：	刀具半徑補正輪廓右側
	附註： 啟動必須在線性區塊內執行 (G0/G1)。	
CDOF2：	停用 3D 圓周銑削的碰撞偵測	
A4/5=...B4/5=...C4/5=...：	限制平面的表面法線向量之定義	
	A4=...B4=...C4=...：	單節起點的定義
	A5=...B5=...C5=...：	在單節結尾的定義
G40 X...Y...Z...：	停用刀具半徑補正 **附註：** 停用必須在含有幾何軸移動的線性區塊內執行 (G0/G1)。	

說明

選擇 3D TRC 的 G 指令是在逼近單節中評估，亦即，通常是在含有 G41 或 G42 的單節中。

G41 或 G42 也可以程式設計在沒有移動與補正相關的幾何軸動作單節中。在此情況下，逼近單節是第一個移動單節，該單節緊跟在後。

含啟用刀具半徑補正的 3D TRC 版本變更將忽略，沒有警報。

範例

程式碼	註解
N10 $TC_DP1[1,1]=120	；圓柱銑削刀具
N20 $TC_DP6[1,1]=10	
N30 $TC_DP15[1,1]=-3	
...	
；以圓柱銑削刀具及 CUT3DCCD 處理	
N110 TRAORI	；啟動轉換。
N120 A4=0 B4=0 C4=1	；限制平面在單節起點的表面法線向量之定義。
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	
N140 X10 Y0 Z0 G41 **CUT3DCCD** CDOF2 G64	；啟動 3D 圓周銑削，將限制表面 + 關閉碰撞偵測考量在內。
N150 X20	
N160 X30 A45	；鈍角 ==> 無進給
N170 X40 A-45	；銳角 ==> 進給

程式碼	註解
N180 X55	
N190 Y10 Z10	；朝刀具方向移動。
N200 Y20	
N210 C45	；單純的方向變更。
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	；表面的變更。
N240 Y50 G40	；停用刀具半徑補正。
...	

其他資訊

刀具類型

評估刀具型號（刀具參數 $TC_DP1）。只認可具圓柱端的銑削刀具（圓柱或端銑刀、環型銑刀，在有限情況下的柱形模銑刀）。這對應於刀具類型 1 - 399，類型 111 及 155 至 157 除外。

含倒圓角磨圓的標準刀具

以刀具參數 $TC_DP7 定義使用標準刀具的倒圓角磨圓。刀具參數 $TC_DP16 說明與標準刀具相比下，實際刀具的倒圓角磨圓偏差。

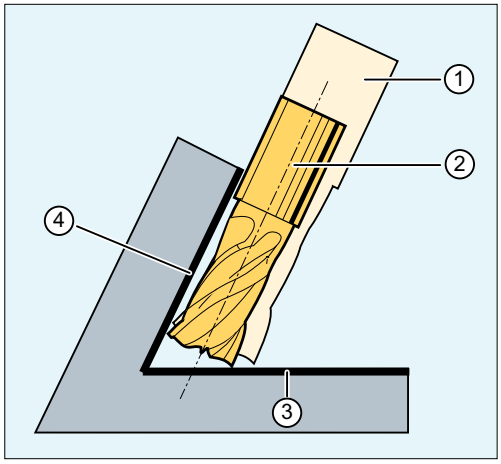
範例：相較於標準刀具半徑縮小的環型銑刀

刀具類型	軸半徑 (R)	圓角半徑 (r)
含倒圓角磨圓的標準刀具	R = $TC_DP6	r = $TC_DP7
具角磨圓的實際刀具 刀具類型 121 及 131 環型銑刀（以角磨圓的端面銑削）	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16

在本範例中，$TC_DP15 + OFFN 及 $TC_DP16 皆為負數。

使用 CUT3DCCD 的 3D TRC：含進給直到限制表面的刀具中心點路徑

若使用半徑小於適當標準刀具的刀具，則使用銑削刀具，依縱向饋入持續加工直到口袋底部（基部）為止。刀具會從加工表面及限制表面形成的倒圓角，盡可能切削材料。此牽涉到結合圓周及平面銑削的加工類型。對於半徑漸增的刀具，類比於含較小半徑的刀具，進給在反方向。



① 標準刀具

② 含較小半徑進給直到限制表面的刀具

③ 限制表面

④ 加工平面

與所有 G 群組 22 的其他刀具偏移量相反，為 CUT3DCCD 指定的刀具參數 $TC_DP6 與刀具半徑不相關，不會影響其產生的補正。刀具半徑磨耗值總和產生的補正偏移量（刀具參數 $TC_DP15），以及程式設計以計算限制表面之垂直偏移的刀具偏移 OFFN。

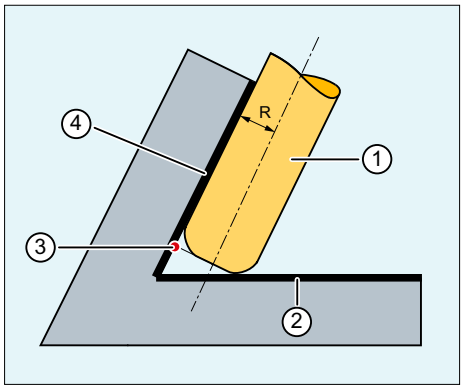
產生的工件程式未指定欲加工的平面在路徑的右側或左側。因此假設半徑為正值，而原始刀具的磨損值為負值。負磨耗值永遠以縮小的直徑定義刀具。

使用圓柱刀具

使用圓柱刀具時，僅在自銳角（小於 90 度）加工表面及限制表面時，才需進給。若使用環型銑刀（含角磨圓的端面銑削），銳角及鈍角皆需縱向中的刀具進給。

使用 CUT3DCC 的 3D TRC：加工表面的輪廓

若環型銑刀啟用 CUT3DCC，則已程式設計的路徑，會參考具有相同直徑的虛設圓柱銑削刀具。得出的路徑參考點顯示於下列環型銑刀的圖示中。



① 環型銑刀
② 限制表面
③ 路徑參考點
④ 加工平面
R 軸半徑（刀具半徑）

即使在同一單節中，加工與限制表面間的角度可能從銳角變成鈍角，或從鈍角變銳角。

實際使用的刀具可能大於或小於標準刀具。然而，所得的倒圓角半徑不得為負值，且需保留所得的刀具半徑符號。

CUT3DCC，NC 工件程式會參考加工表面的輪廓。至於傳統的刀具半徑補正，使用包含下列元件的總刀具半徑：

- 刀具半徑（刀具參數 $TC_DP6）
- 磨耗值（刀具參數 $TC_DP15）
- 程式設計以計算限制表面之垂直偏移的刀具偏移量 OFFN

依下列差距來定義限制表面的位置：

標準刀具的尺寸 - 刀具半徑（刀具參數 $TC_DP6）

Advanced Surface / Top Surface

說明

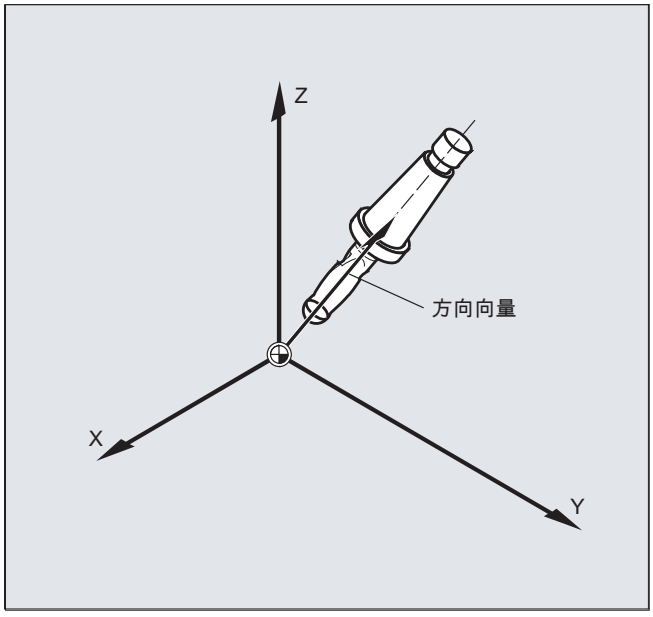
套用刀具半徑補正 CUT3DCC / CUT3DCCD 並搭配「Advanced Surface」或「Top Surface」（需要授權）功能時，必須注意與「Advanced Surface」/「Top Surface」有關的設定建議。

檢查設定資料時，SIOS 入口網頁提供特殊的測試程式：

- Advanced Surface (https://support.industry.siemens.com/cs/ww/en/view/78956392) 的測試程式
- Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423) 的測試程式

11.6 刀具方向（ORIC、ORID、OSOF、OSC、OSS、OSSE、ORIS、OSD、OST）

刀具方向一詞定義在空間中刀具的幾何基準。在五軸工具機上的刀具方向可以透過程式指令設定。



以 OSD 和 OST 啟動的倒圓角動作會依刀具方向的插補類型而以不同方式形成。

如果已啟用向量插補，已平滑化的方向特性也是使用向量插補進行插補。另一方面，如果已啟用旋轉軸插補，是直接使用旋轉軸動作平滑化方向。

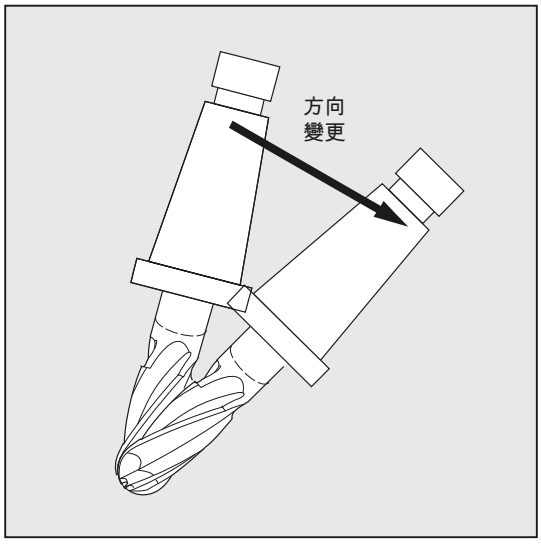
程式設計

將方向變更程式設計：

刀具方向的變更可以依下列方式程式設計：

- 直接將旋轉軸 A、B、C 程式設計（旋轉軸插補）
- 尤拉角或 RPY 角
- 方向向量（指定 A3 或 B3 或 C3 進行向量插補）
- LEAD/TILT（平面銑削）

參考座標系統是機械座標（ORIMKS）或目前的工件機械座標（ORIWKS）。



程式設計刀具方向：

ORIC：	方向與路徑動作平行
ORID：	方向與路徑動作連續進行
OSOF：	沒有方向平滑化
OSC：	經常定向
OSS：	只在單節開始處進行方向平滑化
OSSE：	只在單節開始與結尾處進行方向平滑化
ORIS：	已啟動方向平滑化的方向變更速率是每毫米以度計算（對 OSS 與 OSSE）有效
OSD：	以設定資料來指定平滑化距離的方向平滑化： SD42674 $SC_ORI_SMOOTH_DIST
OST：	使用設定資料，透過為向量插補以度指定角度允差來平滑化方向： SD42676 $SC_ORI_SMOOTH_TOL 進行旋轉軸插補時，就會取指定的允差做為方向軸的最大變異。

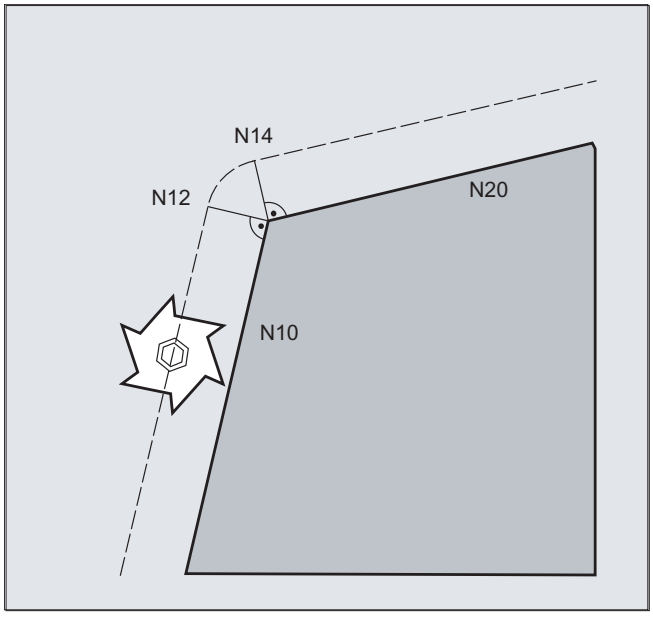
說明

所有用於平滑化刀具方向的指令（OSOF，OSC，OSS，OSSE，OSD，與 OST）皆歸納在 G 群組 34 中。他們為模態；換句話說，一次只有一個指令可以生效。

範例

範例 1：ORIC

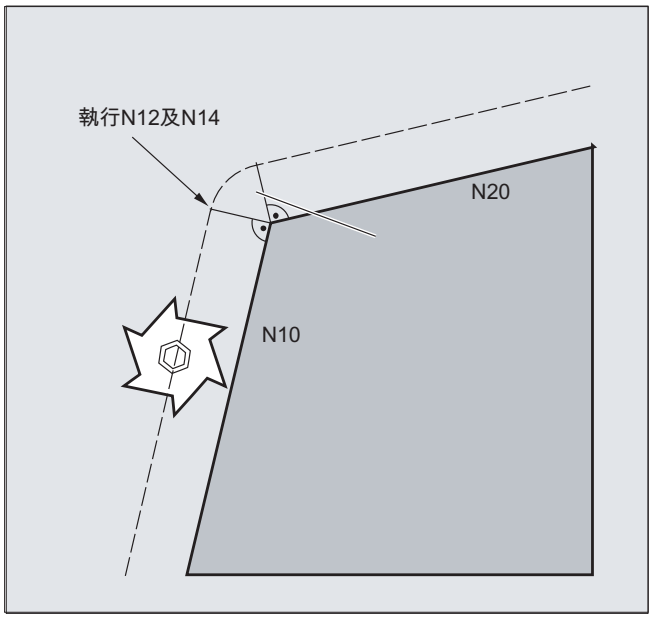
若將含方向變更的兩個或以上單節程式設計在移動單節 N10 與 N20 之間（例如 A2=...B2=...C2=...）且 ORIC 已啟用，則會根據角度中的絕對變更，將插入的圓弧單節散佈在這些中間單節之間。



程式碼	註解
ORIC	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 C2=... B2=...	；插入外部角的圓弧單節是散佈於 N12 與 N14 之間，對應於方向中的變更。圓弧動作和方向變更是平行執行。
N14 C2=... B2=...	
N20 X =...Y=... Z=... G1 F200	

範例 2：ORID

如果已啟用 ORID，則兩個移動單節之間的所有單節都會在第一個移動單節結束時執行。具有恆定方向的圓弧單節就在第二個移動單節之前執行。



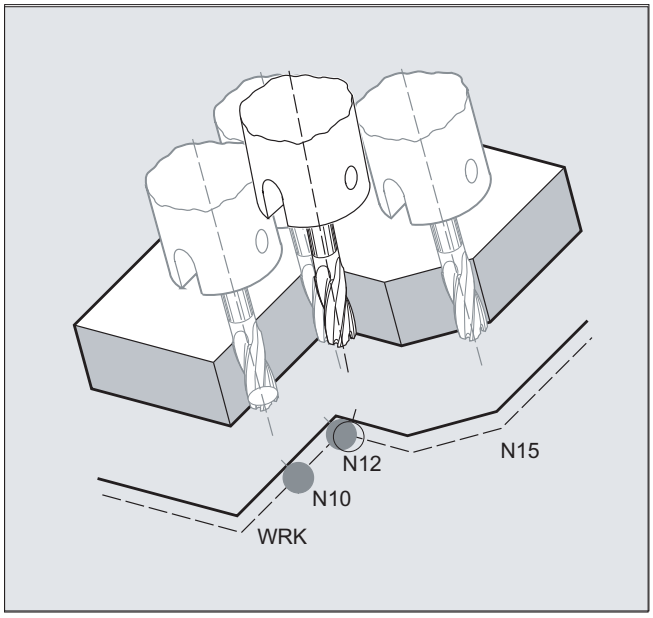
程式碼	註解
`ORID`	
`N8 A2=... B2=... C2=...`	
`N10 X... Y... Z...`	
`N12 A2=... B2=... C2=...`	；N12 和 N14 單節是在 N10 結束時執行，然後再執行具有實際方向的圓弧單節。
`N14 M20`	；輔助說明函數等。
`N20 X... Y... Z...`	

說明

用來在外角變更方向的方法是使用在外角的第一個移動單節中啟用的程式指令來決定。

沒有變方向時：如果沒有在單節界限變更方向，刀具的斷面是圓弧，會同時接觸兩個輪廓。

範例 3：在內部轉角變更方向



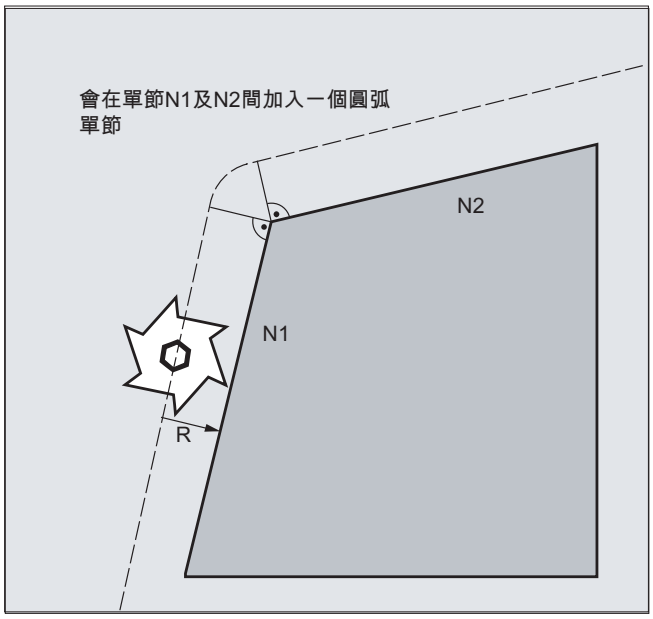
程式碼
ORIC
N10 X ...Y... Z... G1 F500
N12 X ...Y... Z... A2=... B2=... C2=...
N15 X ...Y... Z... A2=... B2=... C2=...

其他資訊

在外角的行為

具有銑刀半徑的圓弧單節永遠都插入外角。

ORIC 和 ORID 程式指令是用來決定 N1 和 N2 單節之間已程式設計的方向變更是在插入圓弧單節之前處理，或是同時處理。



如果外角需要方向變更，可以與插補同時執行，或是分別配合路徑動作執行。

當 ORID 已程式時，插入的單節會在沒有路徑動作之前先執行。 產生角的圓弧單節就插入於兩個移動單節的第二個之前。

如果在外部角插入多個方向單節，而且已選擇 ORIC，圓弧動作就會根據方向變更的絕對值，散佈在各個插入的單節之間。

以 OSD 或 OST 平滑化方向

使用 G642 調合時，輪廓軸的最大變異與方向軸很不相同。 兩個之中較小的允差決定平滑化動作類型及 / 或角度允差，以堅定地平滑化方向特性，而不必接受較高的輪廓偏差。

OSD 與 OST 可啟動為"很全面性地"平滑極小的偏差，從具有特定平滑化距離的方向特性，以及不具有具體輪廓偏差的有角度允差。

說明

與使用 G642 的輪廓磨圓（和方向特性）處理不同，使用 OSD 及 / 或 OST 磨圓方向時，不會形成獨立單節，而是直接將磨圓動作加入已程式設計的原始單節中。

使用 OSD 和 / 或 OST 時，如果刀具方向的插補類型中有變更（向量→旋轉軸，旋轉軸→向量）時，就無法磨圓單節變化。 必要時，這些單節變化可以使用標準的磨圓函數 G641、G642 和 G643 進行磨圓。

11.7 自由定義 D 編號,切削刀刃編號

11.7.1 任意指派 D 數量、刀刃數量（CE 位址）

D 數量

D 數量可用來做為輪廓數量。您也可以經由位址 CE，定出刀刃數量的位址。您可以使用系統變數$TC_DPCE，定義刀刃數量。

預設值：補正數量==刀具稜邊數量

機械參數是用來定義最大的 D 數量（刀刃數量），以及每個刀具最大的刀刃數量（→機台製造 OEM）。只有在最大刀刃數量（MD18105）指定為大於每個刀具的刀刃數量（MD18106）時，下列指令才實際可用。請參閱機台製造商規格。

說明

除了相對 D 數量配置以外，也可以指派 D 數量做為“均一”或“絕對”D 數量（1-32000），而不需要參考 T 數量（在“均一 D 數量結構”函數之中）。

參考

功能手冊基本功能；刀具偏移（W1）

11.7.2 任意指派 D 數量 檢查 D 數量（CHKDNO）

您可以使用 CKKDNO 指令，檢查是現有 D 數量是否為唯一指派。 在 TO 元件中定義的所有刀具 D 數量不可能出現超過一次。 沒有為替換刀具定出允差。

語法

```
state=CHKDNO（Tno1，Tno2，Dno）
```

含義

狀態：	= TRUE：	D 數量是唯一地指派給已檢查的區域。
	= FALSE：	發生 D 數量衝撞，或參數無效。 Tno1、Tno2 和 Dno 會傳回導致衝撞的參數。 這些資料現在可以在工件程式中進行評估。

CHKDNO(Tno1,Tno2)：	指定的工件所有 D 數量都已檢查。
CHKDNO(Tno1)：	Tno1 的所有 D 數量都已依其他所有刀具檢查。
CHKDNO：	所有刀具的所有 D 數量都已依其他所有刀具檢查。

11.7.3 任意指派 D 數量 重新命名 D 數量（GETDNO，SETDNO）

您必須指派唯一的 D 數量。 刀具的兩個不同刀刃不能有相同的 D 數量。

GETDNO

這個指令會以刀具數量 t 傳回刀具之特定刀刃的 D 數量（ce）；如果輸入的參數沒有 D 數量，就會設定 d=0。 如果 D 數量無效，就會傳回大於 32000 的值。

SETDNO

這個指令指派 D 數量的 d 值給刀具 t 的刀刃 (ce)，此敘述的結果是經由狀態（TRUE 或 FALSE）傳回。 如果指定的參數沒有資料單節，就會傳回值 FALSE。 語法錯誤會產生警報。 D 數量不可明確地設定為 0。

語法

```
d = GETDNO （t，ce）
state = SETDNO （t，ce，d）
```

含義

d：	刀具稜邊的 D 編號
t：	刀具的 T 編號
ce：	刀具的刀刃編號（CE 編號）
狀態：	指出是否可執行指令（TRUE 或 FALSE）。

重新命名 D 編號的範例

程式設計	註解
$TC_DP2[1.2]=120	
$TC_DP3[1,2] = 5.5	
$TC_DPCE[1,2] = 3	；刀刃編號 CE
...	
N10 def int DNoOld，DNoNew = 17	

程式設計	註解
N20 DNoOld = GETDNO(1,3)	
N30 SETDNO(1,3,DNoNew)	

然後，新的 D 值 17 就會指派給刀刃 CE=3。現在刀刃的資料是經由 D 數量 17 定位址；經由系統變數，也在 NC 位址的程式設計中。

11.7.4 任意指派 D 數量 決定指定 D 數量的 T 數量（GETACTTD）

您可以使用 GETACTTD 指令，決定與絕對 D 數量相關聯的 T 數量。 不會有唯一性的檢查。 如果在 TO 元件之中有多個相同的 D 數量，會傳回搜尋中所找到第一個刀具的 T 數量。 這個指令不適合搭配「均一」D 數量使用，因為在此情況下，永遠都會傳回值「1」（資料庫中沒有 T 數量）。

語法

```
status=GETACTTD（Tnr，Dnr）
```

含義

Dnr：	會進行 T 數量搜尋的 D 數量。	
Tnr：	找到 T 數量	
狀態：	值：	含義：
	0	已經找到 T 數量。 Tno 包含 T 數量的值。
	-1	指定的 D 數量中沒有 T 數量；Tno=0。
	-2	D 數量不是絕對的。 Tno 接到所找到第一個刀具的值，其中包含具有 Dno 值的 D 數量。
	-5	函數因為其他原因而無法執行。

11.7.5 任意指派 D 數量無效的 D 數量（DZERO）

DZERO 指令是在重新用工具加工時用來支援。 以這個指令標記的補正資料集不再以 CHKDNO 指令進行驗證。 這些資料集可以透過再次使用 SETDNO 設定 D 數量再次存取。

語法

```
DZERO
```

含義

DZERO：	將 TO 元件的所有 D 數量標記為無效。

11.8 刀把動力學

條件

刀把可以將刀具定向為空間中所有可能方向的條件是

- 同時有兩個旋轉軸 V_1 和 V_2。
- 旋轉軸彼此垂直相交。
- 刀具縱向軸垂直於第二個旋轉軸 V_2。

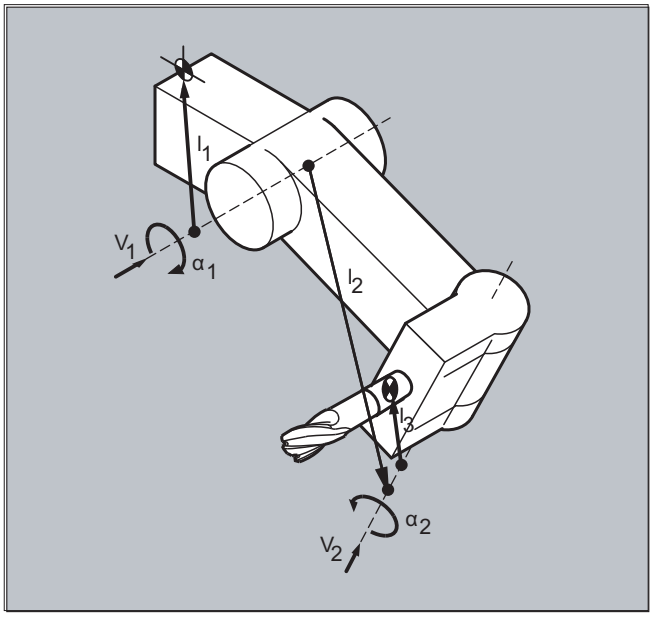
此外，下列先決條件也適用於必須可設定所有可能方向的機台：

- 刀具縱向軸必須垂直於第一個旋轉軸 V_1。

功能

最多兩旋轉軸 v_1 或 v_2 的刀把動力學是使用 17 個系統變數 $TC_CARR1[m] 到 $TC_CARR17[m] 來定義。 刀把的說明包含：

- 離刀把 I_1 第一個旋轉軸的向量距離、從第一個旋轉軸到第二個旋轉軸 I_2 的向量距離、從第二個旋轉軸到刀具 I_3 之參考點的向量距離。
- 兩旋轉軸 V_1、V_2 的方向向量。
- 以兩軸為中心的旋轉角度 α1、α2。 旋轉角度是依旋轉軸向量的觀看方向、正向、旋轉的順時針方向計算。



對於具有**解析結構配置**的機台（刀具和工件都可以旋轉），系統變數已使用項目 $TC_CARR18[m] 到 $TC_CARR23[m] 擴充。

參數

可定向刀把之系統變數的函數			
指定	**x 元件**	**y 元件**	**z 元件**
l_1 偏移向量	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
l_2 偏移向量	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
v_1 旋轉軸	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
v_2 旋轉軸	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
旋轉的 α_1 角度 旋轉的 α_2 角度	$TC_CARR13[m] $TC_CARR14[m]		
l_3 偏移向量	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]

可定向刀把之系統變數的延伸			
指定	**x 元件**	**y 元件**	**z 元件**
l_4 偏移向量	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
軸識別碼旋轉軸 v_1 旋轉軸 v_2	旋轉軸 v_1 和 v_2 的軸識別碼（以零點初始化） $TC_CARR21[m] $TC_CARR22[m]		
動力學類型 **刀**具 **工**件 **混合**模式	$TC_CARR23[m]		
	動力學類型 T ->	動力學類型 P ->	動力學類型 M
	只有刀具可以旋轉（預設值）。	只有工件可以旋轉	工件和刀具都可以旋轉
偏移 旋轉軸 v_1 旋轉軸 v_2	旋轉軸 v_1 和 v_2 的角度度數，其初始設定取為 $TC_CARR24[m] $TC_CARR25[m]		
角度偏移，旋轉軸 v_1 旋轉軸 v_2	旋轉軸 v_1 和 v_2 之 Hirth 齒系統的偏移度數，其初始設定取為 $TC_CARR26[m] $TC_CARR27[m]		

可定向刀把之系統變數的延伸			
角度增量 v_1 旋轉軸 v_2 旋轉軸	旋轉軸 v_1 和 v_2 之 Hirth 齒系統的偏移度數，其初始設定取為 $TC_CARR28[m] $TC_CARR29[m]		
最小位置，旋轉軸 v_1 旋轉軸 v_2	旋轉軸 v_1 和 v_2 之最小位置的軟體限制 $TC_CARR30[m] $TC_CARR31[m]		
最大位置，旋轉軸 v_1 旋轉軸 v_2	旋轉軸 v_1 和 v_2 之最大位置的軟體限制 $TC_CARR32[m] $TC_CARR33[m]		
刀把名稱	刀把可以命名而不用編號。 $TC_CARR34[m]		
使用者： 軸名稱 1 軸名稱 2 識別碼 **位置**	預計使用於使用者量測循環$TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
微調 **偏移量**	可加入**基本參數**中 **值**的參數。		
l_1 偏移向量	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
l_2 偏移向量	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
l_3 偏移向量	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
l_4 偏移向量	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
v_1 旋轉軸	$TC_CARR64[m]		
v_2 旋轉軸	$TC_CARR65[m]		

說明

參數的說明

「m」指定要程式設計的刀把編號。

$TC_CARR47 到$TC_CARR54 以及$TC_CARR61 到$TC_CARR63 並未定義，而且如果嘗試讀取或寫入存取時會產生警報。

軸上距離向量的開始/結尾可以任意選擇。 繞著兩個軸的旋轉角度 α_1、α_2 是在刀把的初始狀態中以 0° 定義。 以這種方式，刀把的動力學可以為任何可能數量進行程式設計。

只有一個旋轉軸或沒有旋轉軸的刀把可以透過將一個或兩個旋轉軸的方向向量設定為零點來說明。

如果刀把沒有旋轉軸，距離向量可做為附加刀具偏移量，其元件不能受加工平面的變更影響 (G17 到 G19)。

參數延伸

旋轉軸的參數

系統變數已由整個輸入$TC_CARR24[m]到$TC_CARR33[m]加以延伸，並說明如下：

旋轉軸 v_1、v_2 的偏移	為已定向刀把的初始設定變更旋轉軸 v_1 或 v_2 的位置。
旋轉軸 v_1、v_2 的**角度偏移 / 角度增量**	旋轉軸 v_1 和 v_2 之 Hirth 齒系統的偏移或角度增量。 程式設計或計算的角度捨入為當 n 為整數時 phi = s + n * d 所產生的下一個值。
旋轉軸 v_1、v_2 的**最小和最大位置**	旋轉軸 v1 和 v2 之旋轉軸限制角度（軟體限制）的最小和最大位置。

使用者參數

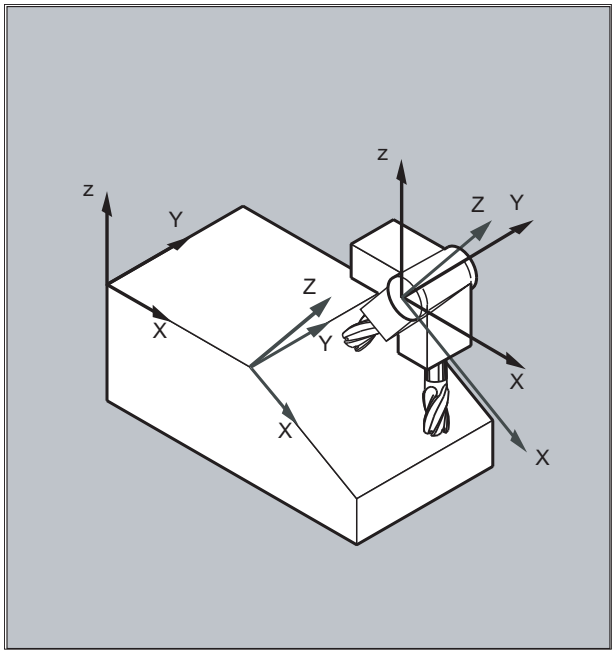
$TC_CARR34 到$TC_CARR40 包含使用者可自由使用的參數，一直到 SW 6.4 都與標準相同，在 NCK 沒有進一步評估，也無有效性。

微調偏移參數

$TC_CARR41 到$TC_CARR65 包含微調偏移參數，可加入至基本參數中的值。 指派給基本參數的微調偏移值是在值 40 加入參數數字時取得。

範例

使用於下列範例中的刀把可以完全由繞著 Y 軸的旋轉加以說明。



程式碼	註解
N10 $TC_CARR8[1]=1	；刀把 1 第一個旋轉軸之 Y 元件的定義。
N20 $TC_DP1[1,1] = 120	；軸銑刀的定義。
N30 $TC_DP3[1,1]=20	；軸銑刀的定義，20 毫米長。
N40 $TC_DP6[1,1]=5	；軸銑刀的定義，5 毫米半徑。
N50 ROT Y37	；繞著 Y 軸 37° 旋轉的框架定義。
N60 X0 Y0 Z0 F10000	；逼近起始位置。
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	設定半徑補正、旋轉框架中的刀長補正（TLC）、選擇刀把 1、刀具 1。
N80 X40	；在 37° 旋轉下執行加工。
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

其他資訊

解析動力學

在具解析動力學的機台（刀具和工作都可以旋轉）上，系統變數已經由輸入 $TC_CARR18[m]擴充到$TC_CARR23[m]，而說明如下：

可旋轉刀具表包含：

- 第二個旋轉軸 V_2 到可旋轉第三個旋轉軸之 I_4 的刀具表參考點之間的向量間隙。

旋轉軸包含：

- 旋轉軸 V_1 和 V_2 參考的兩個通道識別碼，需要時，即存取其位置，以決定可定向刀把的方向。

含 T、P 或 M 其中一個值的動力學類型：

- 動力學類型 T： 只有刀具可以旋轉。
- 動力學類型 P： 只有工件可以旋轉。
- 動力學類型 M： 工件和刀具都可以旋轉。

清除刀把資料

所有刀把資料集的資料都可以使用$TC_CARR1[0]=0 刪除。

動力學類型$TC_CARR23[T]=T 必須指派給三個允許的大寫或小寫字母（T、P、M），因此不可刪除。

變更刀把資料

每個已說明值都可以透過在工件程式中指派新值來修改。 嘗試啟動可以定向的刀把時，T、P 或 M 以外的任何字元都會產生警報。

讀取刀把資料

每個已說明值都可以透過指派給工件程式中的變數來讀取。

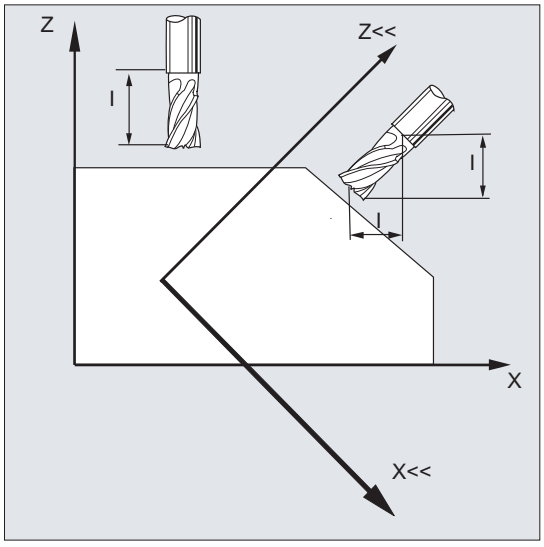
微調偏移量

只有在啟動可以定向的刀把時，才會偵測所有不合規定的微調偏移值，此刀盤中包含不合規定的值，同時設定資料 SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE。

最大允許的微調偏移量受限於機械資料中允許的值。

11.9 用於可定向刀把的刀具長度補正（TCARR，TCOABS，TCOFR，TCOFRX，TCOFRY，TCOFRZ）

當刀具的空間方向變更時，其刀長元件也會跟著變更。



在重置之後（例如對具有固定空間方向的刀把進行手動設定或變更），刀長元件也必須重新定義。這是使用 TCOABS 和 TCOFR 路徑指令執行。

對可以定向主動框架的刀把來說，使用 TCOFRZ、TCOFRY 和 TCOFRX 選擇刀具時，可以定義刀具應該指的方向。

語法

```
TCARR=[<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

含義

TCARR=[<m>]：	以數量「m」要求刀把
TCOABS：	從目前刀把的方向決定刀長元件
TCOFR：	從主動框架的方向決定刀長元件
TCOFRZ：	主動框架的可定向刀把，含指著 Z 方向的刀具

TCOFRY：	主動框架的可定向刀把，含指著 Y 方向的刀具
TCOFRX：	主動框架的可定向刀把，含指著 X 方向的刀具

其他資訊

從刀把的方向決定刀長偏移量 (TCOABS)

TCOABS 從刀把的目前方向角度計算刀長偏移量；儲存於系統變數 $TC_CARR13 及 $TC_CARR14。

關於含系統變數的刀把動力學定義，請參閱「刀盤動力學 (頁 493)」。

框架變更時，若要重新計算刀長偏移量，必須再次選擇刀具。

主動框架的刀具方向

設定了含定向功能的刀把，讓刀具指向下列方向：

- 使用 TCOFR 或 TCOFRZ，指向 Z 方向
- 使用 TCOFRY，指向 Y 方向
- 使用 TCOFRX，指向 X 方向

刀長偏移量是在 TCOFR 與 TCOABS 之間變換時重新計算。

要求刀把（TCARR）

使用 TCARR 時，刀把數量 m 是以其幾何資料要求（補正記憶體）。

使用 m=0 時，則取消選擇已啟用刀把。

只有在呼叫刀具之後，刀把的幾何資料才會有效。 選擇的刀具會在刀把變更發生後維持啟用。

刀把目前的幾何資料也可以在工件程式中，經由對應系統變數定義。

重新計算框架變更的刀長偏移量 (TCOABS)

框架變更時，若要重新計算刀長偏移量，必須再次選擇刀具。

說明

刀具方向必須手動手動倍率以適應主動框架。

計算刀長偏移量 (TLC) 時，刀把的旋轉角度是在一個中間步進計算。 使用有兩個旋轉軸的刀把時，一般是有兩組旋轉軸，可用來手動倍率刀具方向以適應主動框架，因此儲存在系統變數中的旋轉角度值必須至少大約對應於以機械方式設定的旋轉角度。

說明

刀具方向

控制系統不可能檢查旋轉角度是否經由機台可設定的框架方向來計算。

如果刀把的旋轉軸排列方式，使得刀具方向是經由無法到達的框架方向來計算，就會輸出警報。

不允許在可移動刀把上結合使用刀具精準補正和刀長偏移量的函數。 如果同時呼叫這兩個函數，就會發出錯誤訊息。

TOFRAME 函數允許根據所選擇刀把的定位方向來定義框架。 如需其他詳細資訊，請參閱「框架」一章。

啟用方向轉換（三、四或五軸轉換）時，可以選擇方向是從零點位置衍生而來的刀把，而不會造成警報輸出。

標準和量測循環的傳輸參數

在標準和量測循環的傳輸參數上，適用下列已定義的值域。

角度值的已定義值域如下：

- 繞第 1 幾何軸旋轉： -180 度到+180 度
- 繞第 2 幾何軸旋轉： -90 度到+90 度
- 繞第 3 幾何軸旋轉： -180 度到+180 度

請參閱「框架」一章的「可程式設計旋轉（ROT，AROT，RPL）」。

說明

當傳輸角度值至標準或量測循環時，應該小心觀察下列規則：

值若小於 NC 之計算的解析度就應該捨入至零點！

角度位置之 NC 的計算解析度是在機械資料中定義：

MD10210 $MN_INT_INCR_PER_DEG

11.10 線上刀長補正（TLC）（TOFFON，TOFFOF）

利用系統變數 $AA_TOFF[<n>]，依照三個刀具方向，以三度空間即時調整有效刀長。

使用三個幾何軸識別碼當作索引 **<n>**。 因此，有效方向偏移數是由同時啟用之幾何軸決定。

可同時啟用所有偏移。

線上刀長偏移量函數可以用於：

- 方向轉換 TRAORI
- 可定向刀把 TCARR

說明

線上刀長偏移量是**選項**，必須事先啟用。 只有結合有效方向轉換或已啟用可定向刀把時，此函數才有實效可用。

語法

```
TRAORI
TOFFON(<補正方向>[,<偏移值>])
WHEN TRUE DO $AA_TOFF[<補正方向>]                                    ；在同步動作中。
...
TOFFOF(<補正方向>)
```

更多關於運動同步動作中線上刀長偏移量的程式設計資訊，請參閱「同步動作 (頁 661)」。

含義

TOFFON：	**啟用**線上刀長偏移量	
	<補正方向>：	線上刀長偏移量必須生效的刀具方向 (X、Y、Z)。
	<偏移值>：	啟動時，可以為補正的相關方向指定偏移值，這項指定會立即恢復。
TOFFOF：	**重置**線上刀長偏移量 將所指定補正方向上的補正值重置，並啟動前置處理停止。	

範例

範例 1： 選擇刀長補正

程式碼	註解
MD21190 $MC_TOFF_MODE = 1	；逼近絕對值。
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）	；轉換啟動。
N20 TOFFON（Z）	；啟用刀具 Z 軸方向的線上刀具長度補正。
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	；刀具 Z 軸方向的 TLC 插補為 10。
...	
N100 XOFFSET=$AA_TOFF_VAL[X]	；指派 X 軸方向的實際補正。
N120 TOFFON(X,-XOFFSET) G4 F5	；刀具 X 軸方向的 TLC 退為 0。

範例 2： ；取消選擇刀長偏移量

程式碼	註解
N10 TRAORI（1）	；轉換啟動。
N20 TOFFON（X）	；啟用刀具 X 軸方向的線上刀具長度補正。
N30 WHEN TRUE DO $AA_TOFF[X] = 10 G4 F5	；刀具 X 軸方向的 TLC 插補為 10。
...	
N80 TOFFOF（X）	；刪除刀具 X 軸方向的位置偏移： ...$AA_TOFF[X]=0 軸不移動。 將位置偏移值加到與實際方向對應的工作中實際位置上。

其他資訊

單節準備

在前置處理中的單節準備期間，也會將目前在主要執行中啟用的刀長偏移量考慮在內。如要進一步使用最大容許軸速率，必須在設定刀具偏移量時，使用 STOPRE 前置處理停止來停止單節準備。

在程式啟動後未變更刀長偏移量時，或是如果變更刀具偏移量後，所處理過的單節超過 IPO 緩衝區在旋進與主要旋進之間所能容納的單節，刀具偏移量在旋進時永遠都是已知。

變數$AA_TOFF_PREP_DIFF

插補器中目前生效的補正，以及在單節準備時啟用的補正，兩者之間尺寸的差異，可用變數 $AA_TOFF_PREP_DIFF[<n>] 調整。

手動倍率機械資料和設定資料

補正線上刀長偏移量時，可利用下列系統資料：

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE（覆疊動作的加速度極限）
- MD21190 $MC_TOFF_MODE
  系統變數 $AA_TOFF[<n>] 以絕對值移動或整合。
- MD21194 $MC_TOFF_VELO（線上刀長偏移量的速度）
- MD21196 $MC_TOFF_ACCEL（線上刀長偏移量的加速度）
- 預設臨界值的設定資料：
  SD42970 $SC_TOFF_LIMIT（刀長偏移量值的上限）

參考資料：

功能手冊，特殊功能；F2：多軸轉換

11.11 可旋轉刀具偏移資料之修改

11.11.1 計算方向 (ORISOLH)

預先定義的 ORISOLH 功能可以幫助使用者設定機台的旋轉軸位置，進而將車刀帶入相對於工件的已定義動態獨立位置。先決條件為啟用已經以動態鏈參數化的 6 軸轉換。

提供的二個基本功能為：

- 刀具對齊
  指定 β 及 γ 角。本功能計算出此項所需的三個旋轉軸之角度。
- 直接刀具對齊
  指定第二及第三個方向軸的角度。功能會計算相關的 β 及 γ 角度，以及遺失的第一方向軸角度。

說明

方向軸的順序

若您執行過說明機台結構的動態鏈，從工件至刀具，則以下規格套用至 6 軸轉換的三個方向軸順序：

- 最靠近**工件**的方向軸是**第一**方向軸。
- 最靠近**刀具**的方向軸是**第三**方向軸。

一般來說，第一方向軸為主軸，因此其所對應的旋轉係透過旋轉框架，在這些案例中履行。

語法

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

意義

ORISOLH：	功能呼叫		
<RetVal>：	函數傳回值		
	資料類型：	INT	
	值域：	0、-2、-3、...、-17	
	值：	0	功能結束，無錯誤。
		-2	沒有啟用的有效轉換（6 軸方向轉換）。
		-3	第一參數 (<Cntrl>) 為負值。
		-4	第一參數 (<Cntrl>) 的個位數位置為無效。 只允許數值 0 及 1。
		-5	第一參數 (<Cntrl>) 的十位數位置為無效。 只允許數值 0 至 3。
		-6	第一參數 (<Cntrl>) 的百位數位置為無效。 只允許數值 0 及 1。
		-7	第一參數 (<Cntrl>) 的千位數位置為無效。 只允許數值 0 至 3。
		-8	角度 γ 對「直接刀具對齊」功能而言角度過大。
		-9	指定的軸位置至少有一個違反「直接刀具對齊」功能的軸限制。
		-10	無用的刀具。
		-11	要求的方向無法設定。
		-12	Hirth 接頭的自由軸角度調整對第一或唯一的解決方案不可行。
		-13	Hirth 接頭的自由軸角度調整對第二解決方案不可行。
		-14	Hirth 接頭的自由軸角度調整對兩個解決方案皆不可行。
		-15	第一方向軸參數化成 Hirth 軸。
		-16	第二及第三旋轉軸參數化成 Hirth 軸。只有兩個軸的其中一個可以是 Hirth 軸。
		-17	指定的軸位置至少有一個與「直接迴轉」功能的相關 Hirth 接頭不相容。

<Cntrl>：	功能的行為控制項		
	資料類型：	INT	
	<Cntrl> 參數是以十進制編碼（個位數至千位數位置）：		
	個位數位置：	個位數位置控制對於錯誤的反應。	
		xxx0	錯誤的狀況中（傳回值 **< 0**），輸出警報 14106 並且放棄停止程式處理。 **附註：** 在 **<Cntrl>** 參數為負值時，即使與個位數位置的值無關，仍然輸出警報。
		xxx1	錯誤的狀況中（傳回值 **< 0**），不輸出警報。使用者可以在程式中適當反應。
	十位數位置：	在含 Hirth 接頭的方向軸存在時，控制其行為。 **附註：** 此參數僅對「刀具對齊」功能進行評估（亦即，百位數位數的值為「0」）。	
		xx0x	軸位置四捨五入至最近的位置。
		xx1x	軸位置四捨五入，使得 β 角度與其程式設計的值之間差異最小化。
		xx2x	軸位置四捨五入，使得 β 角度等於最大可能值，此值小於程式設計的值（β 向下取整數）。
		xx3x	軸位置四捨五入，使得 β 角度等於最小可能值，此值大於程式設計的值（β 向上取整數）。
	百位數位置：	指定要執行哪一個功能，或是以下兩個參數 **<W1>** 與 **<W2>** 的顯著差異。	
		x0xx	「刀具對齊」功能 參數 **<W1>** 及 **<W2>** 具有以下意義： ● **<W1>** = β ● **<W2>** = γ 計算方向軸的相關角度。
		x1xx	「直接刀具對齊」功能 **<W1>** 是第二方向軸的位置規格，**<W2>** 是 6 軸轉換的第三方向軸之位置規格。定義與兩個位置規格相容的第一方向軸位置，及 β 與 γ 角度。 若未發生錯誤，永遠會在 $P_ORI_POS[<n>, <m>] 系統變數中輸出兩個解決方案。第一個索引 **<n>**（0

			或 1）參考解決方案，而第二個索引 **<m>** (0...2) 參照至方向軸： • $P_ORI_POS[0/1, **0**]：第一方向軸的位置 • $P_ORI_POS[0/1, **1**]：角度 β • $P_ORI_POS[0/1, **2**]：角度 γ 進行檢查，以確認位置規格 **<W1>** 及 **<W2>** 是否與任何 Hirth 接頭或啟用的軟體限制相容。若狀況並非如此，將傳回對應的錯誤編號（參閱 **<RetVal>** 參數）。 若任意選擇角度 **<W1>** 及 **<W2>**，刀具的刀刃一般來說不會在加工平面上。環繞加工平面旋轉的刀刃之角度 γ，不得大於以設定資料 SD42999 $SC_ORISOLH_INCLINE_TOL 定義的臨界值。
	千位數位置：	百位數位置的數值為「0」時，指定可以修改那一個解決方案的位置，亦即「刀具對齊」功能。	
		0xxx	計算的軸位置應盡可能靠近目前機台的軸位置。
		1xxx	計算的模數軸之軸位置應該盡可能靠近模數範圍的中點，至於其他軸則盡可能靠近 0。對非模數軸而言，這意謂著軸位置將減少至範圍 -180°…+180° 之間。
		2xxx	計算的軸位置應減少至範圍 -180°…+180° 之間，與軸類型無關。
<W1>：	第一角度 源自於 **<Cntrl>** 參數的百位數位置的意義。		
	資料類型：	REAL	
<W1>：	第二角度 源自於 **<Cntrl>** 參數的百位數位置的意義。		
	資料類型：	REAL	

說明

尚未程式設計的參數，其預設值為「0」。

其他資訊

所找到的解決方案數量，以及執行 ORISOLH 功能時的其他狀態資訊，可以透過以下系統變數讀取：

系統變數	意義		
$P_ORI_POS [<n>, <m>]	傳回來自方向程式設計的方向軸之角度。		
	<n>： 解決方案的索引		
		值域：	0、1
	<m>： 方向軸的索引		
		值域：	0...2
			方向軸的順序 (1...3) 參考 $NT_ROT_AX_NAME 中軸的定義。
	在「直接刀具對齊」模式中呼叫 ORISOLH 功能時，$P_ORI_POS[0/1, 1] 及 P_ORI_POS[0/1, 2] 變數包含了屬於兩個解決方案的兩個角度 β 與 γ 的值。 輸入至 $P_ORI_POS[<n>, <m>] 的第一解決方案，亦即索引 <n> = 0，會在要求的方向直接逼近時，控制項永遠選擇的解決方案。第二個索引 <m> 參考方向軸，亦即在 $NT_ROT_AX_NAME 上。 輸入至 $P_ORI_POS[<n>, <m>] 中的軸位置將輸入至 $NK_OFF 及 $NK_OFF_FINE 中的偏移量納入考量，亦即這些軸角度可以使用在以下的單節中，設定要求的方向而無須任何其他修改。 若旋轉軸為 Hirth 軸，解決方案的位置四捨五入至 Hirth 接頭其餘的最近位置上。對於 Hirth 接頭的旋轉軸，您可以讀取確實解決方案的軸位置與解決方案調整至在 $P_ORI_DIFF 系統變數中增量的 Hirth 之間的差異。		
$P_ORI_DIFF [<n>, <m>]	傳回方向軸確實位置與來自於方向程式設計，由 $P_ORI_POS 提供的那些位置之間的差異。		
	<n>： 解決方案的索引		
		值域：	0、1
	<m>： 方向軸的索引		
		值域：	0...2
			方向軸的順序 (1...3) 參考 $NT_ROT_AX_NAME 中軸的定義。
	位置為增量時，內容只能不等於零（Hirth 接頭），亦即，相關軸的系統資料 $NT_HIRTH_INCR 不等於零時，以及此軸為手動旋轉軸時。		

系統變數	意義		
$P_ORI_SOL	若是方向轉換含一個以上的旋轉軸，計算的軸角度應該導致指定的方向，一般而言會有一個以上的解決方案。$P_ORI_SOL 系統變數包含有效解決方案的編號，以及其他的狀態資訊。 $P_ORI_SOL 的內容如下編碼：		
	值 < 0	一般的錯誤狀態	
		-1	啟用的轉換尚未計算出任何解決方案（ORISOLH 的遺失呼叫）。
		-2	轉換未啟用、或啟用的轉換並非可以提供指定方向程式設計的位置之方向轉換（6 軸轉換）。
		-4	要求的方向無法以現有的動態設定。
		-5	在「直接刀具對齊」模式中呼叫 ORISOLH 功能時，未發現解決方案。
		-6	在「直接刀具對齊」模式中呼叫 ORISOLH 功能時，角度 γ 過大。
		-7	在由於 Hirth 接頭而無法設定的「直接刀具對齊」模式中，呼叫 ORISOLH 功能時，指定某個角度。
		-8	第一方向軸（框架軸）不得參數化成 Hirth 軸。
		-9	第二及第三旋轉軸參數化成 Hirth 軸。只有兩個軸的其中一個可以是 Hirth 軸。
		-10	Hirth 接頭的解決方案未發現調整。
	值 > 0 個位數位置	數學式可能解決方案的數量，未考慮軸限制及任何錯誤條件。	
		0	沒有解決方案，亦即無法設定要求的方向。 此狀況有三種不同的肇因： • 原則上，由於機台的動態，即使方向軸以任意的移動範圍，所要求的方向亦無法達成（方向軸的配置並非直角）。在此狀況下，$P_ORI_SOL 的十位數及百位數位置皆為零，指派給方向軸的 $P_ORI_STAT 狀態變數數值為「-4」。 • 計算的解決方案無法達成，因為違反軸限制。所得結果的方向軸位置不含軸限制，可以在 $P_ORI_POS 中讀取。 • 在「直接刀具對齊」模式中呼叫 ORISOLH 功能時可以指定軸位置，其將導致方向向量或刀具的方向法線向量在計算位置時，平行對齊第一方向軸。在這些狀況中，未定義此軸的位置。

系統變數	意義		
		1	有解決方案。 此狀況有三種不同的肇因： ● 根據指定的方向及機台的動態，即使沒有考慮軸限制，也只有一個解決方案（從數學式的觀點，兩個一致的解決方案）。此狀況發生在非直角的動態方向範圍之邊界，$P_ORI_POS 包含兩個（相同）解決方案。 ● 由於違反軸限制，第二個解決方案無效，因此只有一個解決方案。有效的解決方案永遠是 $P_ORI_POS 中的第一個解決方案。若未將軸限制納入考量，將得出第二個解決方案，也可以在 $P_ORI_POS 中讀取。 ● 在「直接刀具對齊」模式中呼叫 ORISOLH 功能時，這屬於正常狀況。對於兩個方向軸的指定軸位置而言，一般而言遺失的方向軸只有計算一個有效位置。
		2	有兩個解決方案。
		8	有無限個解決方案，亦即，方向軸的位置（極座標軸）為任意的。不過其他軸的兩個可能解決方案，會排除其中一個，因為違反了軸限制。
		9	有無限個解決方案，亦即，方向軸的位置（極座標軸）為不確定的。不確定軸可以從百位數位置，或是從 $P_ORI_STAT 系統變數決定。
	值 > 0 十位數位置	違反的軸限制之位元編碼顯示。精確的錯誤肇因可以從 $P_ORI_STAT 系統變數決定。	
		位元 0（值 10）：	至少一個解決方案，第一方向軸至少違反一個軸限制。
		位元 1（值 20）：	至少一個解決方案，第二方向軸至少違反一個軸限制。
		位元 2（值 40）：	至少一個解決方案，第三方向軸至少違反一個軸限制。
	值 > 0 百位數位置	未定義軸位置的位元編碼顯示（只有在解決方案有無限個時才發生，亦即個位數位置等於「9」時）。	
		位元 0（值 100）：	第一方向軸的位置未定義。
		位元 1（值 200）：	第二方向軸的位置未定義。

系統變數	意義		
		位元 2（值 400）：	第三方向軸的位置未定義。
	指定的第一、第二及第三方向軸參考 $NT_ROT_AX_NAME 中的軸定義。		

系統變數	意義			
$P_ORI_STAT [<n>]	呼叫 ORISOLH 之後，傳回最多三個方向軸的每個狀態。			
	<n>： 方向軸的索引 （對應至 $NT_ROT_AX_NAME 中相關方向軸的索引）			
		值域：	0...2	
			方向軸的順序 (1...3) 參考 $NT_ROT_AX_NAME 中軸的定義。	
	$P_ORI_STAT 的內容如下編碼：			
	值 < 0	一般的錯誤狀態		
		-1	狀態未定義（遺失 ORISOLH 呼叫）。	
		-2	轉換未啟用、或啟用的轉換並非可以提供指定方向程式設計的位置之方向轉換（6 軸轉換）。	
		-3	軸未包含在啟用的轉換內。	
		-4	由於現有的動態，即使任意假設軸的移動範圍，要求的方向仍無法達成，因此無法計算軸的位置。	
		-5	在「直接刀具對齊」模式中呼叫 ORISOLH 功能時可以指定軸位置，其將導致方向向量或刀具的方向法線向量在計算位置時，平行對齊第一方向軸。在這些狀況中，未定義此軸的位置。	
		-6	在「直接刀具對齊」模式中呼叫 ORISOLH 功能時，角度 γ 過大。	
		-7	在由於 Hirth 接頭而無法設定的「直接刀具對齊」模式中，呼叫 ORISOLH 功能時，指定某個角度。	
		-8	第一方向軸（框架軸）不得參數化成 Hirth 軸。	
		-9	第二及第三旋轉軸參數化成 Hirth 軸。只有兩個軸的其中一個可以是 Hirth 軸。	
		-10	Hirth 接頭的解決方案未發現調整。	
	值 > 0	第一個解決方案違反的軸限制之位元編碼顯示。		
	個位數位置	位元 0（值 1）：	第一個解決方案違反軸下限。	
		位元 1（值 2）：	第一個解決方案違反軸上限。	
	值 > 0	第二個解決方案違反的軸限制之位元編碼顯示。		

系統變數	意義		
	十位數位置	位元 0（值 10）：	第二個解決方案違反軸下限。
		位元 1（值 20）：	第二個解決方案違反軸上限。
	值 > 0	未定義軸位置的顯示。	
	百位數位置	位元 0（值 100）：	方向軸的位置未定義，亦即，含旋轉軸方向的每個任意設定（極座標位置）之要求方向均已達成。此項資訊也包含在 $P_ORI_SOL 系統變數中。
	在指出違反軸限制的錯誤編號中，有一些會同時發生。在軸限制違反時，藉由增減 360° 的倍數，試圖在可允許的軸限制範圍內，抵達位置。若仍無法達成，意即沒有明確的定義是否違反了軸的上下限。 若是要求的方向沒有解決方案 ($P_ORI_SOL = 0)，則轉換中的方向軸狀態為「0」。		

說明

$NT_ROT_AX_NAME

此系統變數參考用來設定方向的最多三個軸。它包含了鏈結元素的名稱 ($NK_NAME)，定義了必須執行來自動態轉換之方向移動的機械軸（旋轉軸）。包含在此系統變數中之最多三個旋轉軸的順序與機台動態無關，因為它是來自於動態鏈的結構。不過，由於它定義了其他變數存取旋轉軸的順序，因此在 $NT_ROT_AX_NAME 中的方向軸順序必須符合動態說明。

說明

狀態資訊

例如，顯示方向無法達成、或只能在相關的軸限制違反時達成的狀態資訊，不會觸發 NC 警報。對於指定條件如何適當反應，係屬於使用者的責任。

11.11.2 啟動可旋轉刀具的偏移資料之修改 (CUTMOD, CUTMODK)

可旋轉刀具的偏移資料之修改係透過 CUTMOD（搭配方向刀盤使用）、或 CUTMODK 程式語言指令（以動態鍵的方式定義的方向轉換），在 NC 程式中啟動。

說明

由於可旋轉刀盤及以動態鍵方式定義的方向轉換無法同時啟用，這兩個版本之間並無衝突。

語法

```
CUTMOD = <值>
```

或

```
CUTMODK = <指令>
```

意義

CUTMOD：	功能呼叫搭配可旋轉刀盤		
<值>：	指派的值		
	資料類型：	INT	
	值：	0	函數已停用。 從系統變數 $P_AD... 供應的值與對應刀具參數相同。
		> 0	若是啟用含有指定編號的可旋轉刀盤，功能即啟動，亦即啟動係連結至特定的可旋轉刀盤。 從系統變數 $P_AD... 供應的值可以配合對應刀具參數修改，取決於已啟用的旋轉。 停用所指定的可旋轉刀盤會暫時停用此功能；啟用其他可旋轉刀盤則永遠停用此功能。因此在第一種情況下，功能會在再次選擇相同可旋轉刀盤時重新啟動；而在第二種情況下，即使在後續時都必須要有新的選擇，可旋轉刀盤會以指定的數量重新啟動。 此函數不會受重置影響。
		-1	如果已啟用可旋轉刀盤，永遠都會啟動此功能。 變更刀盤時，或取消選擇刀盤而後續新增選擇時，不必再次設定 CUTMOD。
		-2	如果已啟用可旋轉刀盤，而其編號與目前啟用的可旋轉刀盤相同，則永遠都會啟動此功能。 如果未啟用可旋轉刀盤，則其有效性與 CUTMOD=0 相同。 如果啟用可旋轉刀盤，則其有效性與直接指定的實際刀盤編號相同。
		< -2	小於 2 的值將不予理會，亦即，這種情況視同未程式設計 CUTMOD 處理。 **附註：** 不應該使用此值域，因為已保留作為可能的後續擴充。
CUTMODK：	功能呼叫搭配以動態鍵方式定義的方向轉換一起使用		

<指令>：	指派的指令		
	資料類型：	STRING	
	值：	「NEW」	以動態鏈定義的啟用轉換之狀態與「偏移資料的修改」相關，儲存轉換的名稱及目前的輪廓框架。 **附註：** 此指令只有在適當的轉換（TRAORI_DYN、TRAORI_STAT 或 TRAANG_K）啟用時才允許。
		「OFF」	關閉啟用的「偏移資料的修改」。先前以「NEW」儲存的資料予以保留。 **附註：** 此指令在 CUTMODK 未啟用時也可允許。它會保持毫無影響。「偏移資料的修改」所產生的任何資料集都會保留。
		「ON」	使用此指令，「偏移資料的修改」會利用先前「NEW」指令儲存的資料集重新啟動。 若是執行此指令時，以所儲存的資料集名稱啟用轉換，「偏移資料的修改」會立即生效。否則，將延遲啟動，直到啟用的轉換啟動為止。
		「CLEAR」	如同「OFF」指令，關閉「偏移資料的修改」，同時刪除儲存的資料集。 **附註：** 此指令在 CUTMODK 未啟用時也可允許。

說明

SD42984 $SC_CUTDIRMOD

CUTMOD 或 CUTMODK 指令取代可使用設定資料 SD42984 $SC_CUTDIRMOD 啟動的函數。但是此函數的可用性維持不變。但因為平行使用兩個函數沒有意義，所以只有在 CUTMOD 等於零，以及 CUTMODK 為零字串時，才能加以啟動。

其他資訊

讀取修改的偏移資料

在下列的系統變數及 OPI 變數中，提供修改的偏移資料：

意義	系統變數	OPI 變數
刀刃位置	$P_AD[2]	cuttEdgeParam2
刀盤角度	$P_AD[10]	cuttEdgeParam10
切削方向	$P_AD[11]	cuttEdgeParam11
間隙角度	$P_AD[24]	cuttEdgeParam24

在「可旋轉刀具的偏移資料修改」功能以 CUTMOD 或 CUTMODK 指令啟動，並且刀具以可旋轉刀盤或適當的方向轉換旋轉時，資料會永遠相對於對應的刀具參數（$TC_DP2[..., ...] 等）進行修改。

其他的功能相關系統變數

系統變數	意義
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	刀具透過傳回的角度，在啟用的加工平面上旋轉，並且提供 CUTMOD 及 CUTMODK 功能的修改後刀刃資料。
$P_CUTMOD / $AC_CUTMOD	讀取上次使用 CUTMOD 指令程式設計的目前有效值（應該為啟動偏移資料的修改之刀盤編號）。 如果上次程式設計的值為 CUTMOD = -2（以目前啟用的可旋轉刀盤啟動），則系統變數中不會傳回值「-2」，而會傳回程式設計時的已啟用可旋轉刀盤編號。
$P_CUTMODK / $AC_CUTMODK	在「偏移資料的修改」建立的目前有效資料集底下，讀取轉換的名稱。

系統變數	意義
$P_CUT_INV / $AC_CUT_INV	旋轉刀具時提供值 TRUE，讓旋轉的主軸方向必須逆反。若要執行這項作業，必須在讀取操作所指的單節中滿足下列四個條件： 1. 若是車刀或研磨刀具為啟用（刀具類型 400 至 599 及/或 SD42950 $SC_TOOL_LENGTH_TYPE = 2）。 2. 偏移資料的修改利用 CUTMOD 或 CUTMODK 指令啟動。 3. 啟用可旋轉刀盤或是由動態鏈所定義的方向轉換，這是使用 CUTMOD 或 CUTMODK 指令選擇。 4. 藉由可旋轉刀盤或動態方向轉換旋轉刀具，使得刀具刀刃的所得法線向量繞著相對於初始位置旋轉超過 90° 以上（一般是 180°）。 若是指定的四項條件有至少一項未能履行，變數將傳回數值 FALSE。如果是未定義刀刃位置的刀具，則變數的值永遠是 FALSE。

系統變數	意義	
$P_CUTMOD_ERR	CUTMOD 功能最後一次呼叫之後的錯誤狀態 CUTMOD 功能也可以隱性呼叫，進行換刀。重置時，變數會重設為零。必要時，每次換刀都會重置及重寫。 變數係以位元編碼。位元具有下列意義：	
	位元 0：	未對啟用的刀具定義有效的切削方向。
	位元 1：	啟用的刀具之刀刃角度（間距角度及刀把角度）皆為零。
	位元 2：	啟用的刀具之間距角度有可允許值（< 0° 或 > 180°）。
	位元 3：	啟用的刀具之刀把角度有可允許值（< 0° 或 > 90°）。
	位元 4：	啟用的刀具之板角度有可允許值（< 0° 或 > 90°）。
	位元 5：	不允許刀刃位置 - 刀把角度與啟用刀具的組合（刀刃位置在 1 至 4 時，刀把角度必須 ≤ 90°；刀刃位置 5 至 8 時則必須 ≥ 90°）。
	位元 6：	啟用刀具的不合規定旋轉。 刀具繞著啟用的加工平面旋轉 ± 90°（允差大約 1°）。因此，刀刃位置不再定義在加工平面中。
	位元 7：	切削板未在加工平面中，切削板與加工平面之間的角度超過設定資料 SD42998 $SC_CUTMOD_PLANE_TOL 指定的上限。
	位元 8：	切削板未在加工平面中。角度 α 大於 1°。角度 α 是繞著座標軸旋轉的角度，與繞著角度 β 旋轉的軸，以及繞著角度 γ 旋轉的軸（G18 的 X 軸）垂直。

$P_...：前置處理變數

$AC_...：主要執行變數

所有主要執行變數都可以讀入同步化動作中。前置處理的讀取存取操作會產生前置處理停止。

平面變更

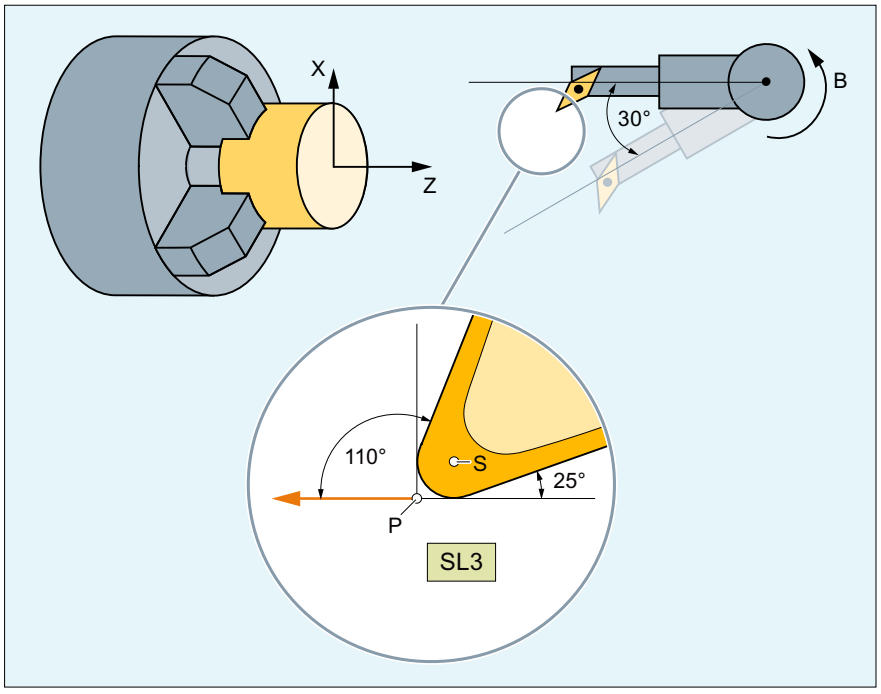
為了決定修改的刀刃位置、切削方向與刀盤或間隙角度，有效平面中刀刃的評估（G17 - G19）是決定性的。

但是，如果設定資料 SD42940 $SC_TOOL_LENGTH_CONST（選擇平面時的刀長元件變更）的有效值不等於零點（正或負 17、18 或 19），則其內容會定義平面，在其中評估相關的數量。

此項設定資料在 G 碼的優先順序規則，可以透過機械資料 $MC_TOOL_PARAMETER_DEF_MASK 的設定位元 18 加以停用。也就是說在設定位元時，以群組 6 的 G 指令定義之平面仍然有效。

已修改切削資料的有效性

修改的刀刃位置和修改的刀刃參考點是在程式設計時立即生效，即使對已經啟用的刀具也一樣有效。不必特別再重新選擇刀具。



S： 刀刃中心點

P： 刀刃參考點

SL： 刀刃位置

圖像 11-4 刀刃位置 3 的刀具，以及可以繞著 B 軸旋轉的刀具之可旋轉刀盤。

程式碼	備註
N10 $TC_DP1[1,1]=500	
N20 $TC_DP2[1,1]=3	；刀刃位置
N30 $TC_DP3[1,1]=12	
N40 $TC_DP4[1,1]=1	
N50 $TC_DP6[1,1]=6	

程式碼	備註
N60 $TC_DP10[1,1]=110	；刀盤角度
N70 $TC_DP11[1,1]=3	；切削方向
N80 $TC_DP24[1,1]=25	；間隙角度
N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0	；B 軸
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0 $TC_CARR12[2]=1	；C 軸
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	

註解中的數值是依 X → Y → Z 順序，定機械座標系統（MCS）的單節位置結尾。

說明

在單節 N180 中，剛開始是為 CUTMOD=0 和非旋轉的可定向刀把選擇刀具。由於所有可定向刀把的偏移向量都是 0，因此是逼近對應於 $TC_DP3[1,1] 和 $TC_DP4[1,1] 中所指定長度的位置。

可以繞著 B 軸旋轉 30° 定向的刀把是在單節 N200 中啟動。由於因 CUTMOD=0 而未修改刀刃位置，舊的刀刃參考點仍固定如前不改動。因此會在單節 N210 中逼近此位置，而讓舊刀鼻參考點保持在零點（也就是，向量（1，12）是在 Z/X 平面中穿過 30° 旋轉）。

在單節 N260 中，與單節 N200 相反，是 CUTMOD=2 生效。由於可定向刀把旋轉的結果，修改的刀刃位置變成 8，因此也是不同的軸位置。

刀具半徑補正 (TRC) 是在單節 N220 和 / 或 N270 中啟動。在這兩個程式區段中不同的刀刃位置不會影響已啟用 TRC 的單節結尾位置，因此對應的位置完全相同。不同的刀刃位置只會在取消選擇單節 N260 及 / 或 N300 時再度生效。

11.12 使用刀具環境工作

函式概觀

- 儲存刀具環境 (TOOLENV) (頁 524)
- 刪除刀具環境 (DELTOOLENV) (頁 526)
- 讀取 T、D 與 DL 編號 (GETTENV) (頁 527)
- 讀取刀長及 / 或刀長元件 (GETTCOR) (頁 529)
- 變更刀具元件 (SETTCOR) (頁 536)

系統變數概觀

- 讀取有關已儲存刀具環境的資訊 ($P_TOOLENVN, ($P_TOOLENV) (頁 528)

11.12.1 儲存刀具環境 (TOOLENV)

使用 TOOLENV 函式儲存目前狀態，以供評估儲存在記憶體中的刀具所需。

個別的資料如下：

- 作用中 G 指令群組：
  - 6 (G17, G18, G19)
  - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- 作用中橫向軸
- 機械資料：
  - MD18112 $MN_MM_KIND_OF_SUMCORR（在 TO 區的加總偏移量屬性）
  - MD20360 $MC_TOOL_PARAMETER_DEF_MASK（刀具參數的定義）。
- 設定資料：
  - SD42900 $SC_MIRROR_TOOL_LENGTH（鏡射時的刀長符號變更）
  - SD42910 $SC_MIRROR_TOOL_WEAR（鏡射時的刀具磨耗符號變更）
  - SD42920 $SC_WEAR_SIGN_CUTPOS（含刀刃系統的刀具磨耗符號）
  - SD42930 $SC_WEAR_SIGN（磨耗符號）
  - SD42935 $SC_WEAR_TRANSFORM（刀具元件的轉換）
  - SD42940 $SC_LENGTH_CONST（平面變更的刀具長度元件變更）
  - SD42942 $SC_TOOL_LENGTH_CONST_T（平面變更的車刀長度元件變更）
  - SD42950 $SC_TOOL_LENGTH_TYPE（獨立於刀具類型之刀具長度元件的配置）
  - SD42954 $SC_TOOL_ORI_CONST_M（平面變更的銑刀長度元件變更）
  - SD42956 $SC_TOOL_ORI_CONST_T（平面變更的車刀長度元件變更）

- 目前完整框架的方向元件（旋轉及鏡射、無偏移量或縮放）
- 具有方向功能的作用中刀把之方向元件及得出的長度
- 作用中轉換的方向元件及得出的長度

除了說明刀具環境的資料以外，也儲存作用中刀具的 T 編號、D 編號及 DL 編號，如此一來在稍後相同環境下，可以利用 TOOLENV 存取而無須再次命名刀具。

語法

```
<狀態> = TOOLENV(<名稱>)
```

意義

TOOLENV(...)：		預先定義的函式，儲存刀具環境		
		單獨在單節中：		是
<狀態>：		函數傳回值。負值代表錯誤狀態。		
		資料類型：		INT
		值：	0	功能 OK
			-1	沒有保留記憶體供刀具環境使用： MD18116 $MN_MM_NUM_TOOL_ENV = 0 亦即未提供「刀具環境」的功能。
			-2	刀具環境未能提供更多的可用記憶體位置。
			-3	刀具環境名稱使用空白字串不合規定。
			-4	無指定參數（<名稱>）
參數				
1	<名稱>：	名稱，目前資料應儲存在其中。 若已有相同名稱的資料集存在，則會被覆蓋。在此情況中，狀態為「0」。		
		資料類型：		STRING

其他資訊

底座尺寸 / 接頭尺寸 - 刀長補正

啟用刀庫管理時（僅適用於「刀具管理」選項！），下列的機械資料數值將定義接頭長度或刀座尺寸（刀刃專屬參數 $TC_DP21、$TC_DP22 及 $TC_DP23）是否納入刀具長度的計算當中：

MD18104 $MN_MM_NUM_TOOL_ADAPTER（TO 區中的刀具接頭）。

由於變更此機械資料只有在控制系統開機後才會生效，它並非儲存在刀具環境中。

具有方向功能及轉換的刀把所得長度：

說明

兼具方向功能與轉換的刀把可以使用系統變數或機械資料，充當為其他的刀長元件，並且部分或全部受限於執行的旋轉。所得的其他刀長元件也必須在呼叫 TOOLENV 時儲存，因為他們代表了所使用刀具的一部分環境。

接頭轉換

接頭轉換是刀具接頭的屬性，因此也是完整刀具的屬性。因此並非刀具環境的一部分，可以套用至另一個刀具。

藉由儲存判定整個刀具長度所需的完整資料，可以稍後再計算刀具的有效長度，即使當時刀具已經失效，或是環境的條件（例如 G 碼或設定資料）已變更。同樣地，不同刀具的有效長度在計算時可以假設是在相同條件下使用狀態已儲存的刀具。

刀具環境的資料集最大數目

使用機械資料 MD18116 $MN_MM_NUM_TOOL_ENV 定義可以被儲存以說明刀具環境的資料集最大數目。資料位在 TOA 區。這些資料即使在控制系統關機時仍可保存。

資料無法備份。也就是說，此資料無法在不同的控制系統間傳輸。

11.12.2 刪除刀具環境 (DELTOOLENV)

使用 DELTOOLENV 函式刪除用來說明刀具環境的資料集。刪除意指儲存在特定名稱下的資料集無法再存取（試圖存取將觸發警報）。

說明

資料集只能使用 DELTOOLENV 函式刪除，必須是透過 INITIAL.INI 下載或冷開機（以預設的機械資料 NC 開機）。並無其他的自動刪除操作方式。

語法

```
<狀態> = DELTOOLENV(<名稱>)
<狀態> = DELTOOLENV()
```

意義

DELTOOLENV(...)：	預先定義的函式，刪除刀具環境		
	單獨在單節中：	是	
<狀態>：	函數傳回值。負值代表錯誤狀態。		
	資料類型：	INT	
	值：	0	功能 OK
		-1	沒有保留記憶體供刀具環境使用： MD18116 $MN_MM_NUM_TOOL_ENV = 0 亦即未提供「刀具環境」的功能。
		-2	含指定名稱的刀具環境不存在。
參數			
1 <名稱>：	待刪除的資料集名稱		
	資料類型：	STRING	
DELTOOLENV()：	DELTOOLENV() 刪除不含指定名稱之刀具環境說明的資料集		

11.12.3 讀取 T、D 與 DL 編號 (GETTENV)

使用 GETTENV 函式讀取儲存於刀具環境之 T、D 和 DL 編號。

語法

```
<狀態> = GETTENV(<名稱>, <TDDL>)
```

意義

GETTENV(...)：	預先定義的函式，讀取資料集中的 T、D 和 DL 編號以說明刀具環境	
	單獨在單節中：	是

<狀態>：	函數傳回值。負值代表錯誤狀態。		
	資料類型：		INT
	值：	0	功能 OK
		-1	沒有保留記憶體供刀具環境使用： MD18116 $MN_MM_NUM_TOOL_ENV = 0 亦即未提供「刀具環境」的功能。
		-2	含指定名稱的刀具環境不存在。
參數			
1 <名稱>：	資料集名稱，讀取其中的 T、D 及 DL 編號		
	資料類型：		STRING
2 <TDDL>：	此所得參數的欄位包含刀具的 T、D 及 DL 編號，其刀具環境儲存在指定的資料集中： • **<TDDL>** [0]：T 編號 • **<TDDL>** [1]：D 編號 • **<TDDL>** [2]：DL 編號		
	資料類型：		INT[3]
GETTENV(,<TDDL>)、GETTENV("",<TDDL>)：	呼叫函式 GETTENV 時，可允許忽略第一個參數，或是將空白字串當作第一個參數傳輸。在這兩個特殊情況的 **<TDDL>** 中，會傳回**作用中**刀具的 T、D 及 DL 編號。		

11.12.4 讀取有關已儲存刀具環境的資訊 ($P_TOOLENVN, ($P_TOOLENV)

有關已儲存刀具環境的資訊可使用以下的系統變數來讀取：

$P_TOOLENVN：	使用 TOOLENV 定義，供應資料集的編號（尚未被刪除者）以說明刀具環境			
	語法：	<n> = $P_TOOLENVN		
	意義：	<n>：	已定義資料集的編號	
			資料類型：	INT
			值域：	0 ... MD18116 $MN_MM_NUM_TOOL_ENV
	即使無任何可用的刀具環境，此系統變數也可以存取 (MD18116 = 0)。在此情況中，傳回值為「0」。			

$P_TOOLENV：	提供第 **<i>** 個資料集的名稱，以便說明刀具環境			
	語法：	<名稱> = $P_TOOLENV[<i>]		
	意義：	<名稱>：	編號 **<i>** 的資料集名稱	
			資料類型：	STRING
		<i>：	資料集編號	
			資料類型：	INT
			值域：	1 ... $P_TOOLENVN
	指派給資料集的編號並未固定，但是可以因為刪除或建立資料集等種種結果而加以變更。資料集是以內部編號。 若 **<i>** 參照至尚未定義的資料集，則將傳回空白字串。 若索引 **<i>** 為無效，亦即 **<i>** 小於 1 或高於刀具環境的資料集的最大編號 (MD18116 $MN_MM_NUM_TOOLENV)，則輸出以下警報： 警報 17020「不可接受之陣列索引 1」			

11.12.5 讀取刀長及 / 或刀長元件 (GETTCOR)

使用 GETTCOR 函式讀出刀長或刀長元件。

可使用參數指定有哪些元件將納入考量，以及在哪些指定條件下使用了刀具。

語法

```
<狀態> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

意義

GETTCOR(...)：	預先定義的函式，讀取刀長或讀取刀長元件	
	單獨在單節中：	是

<狀態>：	函數傳回值。負值代表錯誤狀態。		
	資料類型：		INT
	值：	0	功能 OK
		-1	沒有保留記憶體供刀具環境使用： MD18116 $MN_MM_NUM_TOOL_ENV = 0 亦即未提供「刀具環境」的功能。
		-2	在 **<Stat>** 中指定名稱的刀具環境不存在。
		-3	參數 **<Comp>** 中的無效字串。 造成此錯誤的肇因可能是無效字元或字元程式設計兩次。
		-4	無效的 T 編號
		-5	無效的 D 編號
		-6	無效的 DL 編號
		-7	嘗試存取不存在的記憶體模組。
		-8	嘗試存取不存在的選項（可程式設計的刀具方向、刀具管理）。
		-9	**<Comp>** 字串包含一個分號（座標系統規格的識別碼），但其後並未緊跟著一個代表座標系統的有效字元。
參數			

1	<Len>：	結果向量	
		資料類型：	REAL[11]
		向量元件的安排順序如下： • **<Len> [0]**：刀具類型 • **<Len> [1]**：刀刃位置 • **<Len> [2]**：橫座標 • **<Len> [3]**：縱座標 • **<Len> [4]**：備選座標 • **<Len> [5]**：刀具半徑 使用定義在 **<Comp>** 及 **<Stat>** 中的座標系統，當作長度元件的參考座標系統。若是在 **<Comp>** 中未定義座標系統，則刀長會顯示在機械座標系統中。 依據使用在刀具環境中的作用平面，將橫座標、縱座標及備選座標指派給幾何軸。也就是說，使用 G17 橫座標平行於 X，使用 G18 則平行於 Z 等等。 元件 **<Len>[6]** 至 **<Len>[10]** 包含其他的參數，可以用來指定刀具的幾何說明（例如 $TC_DP7 至 $TC_DP11 的幾何及磨耗的對應元件，或是總和與設定偏移量）。 這 5 個其他元素及刀具半徑只定義給元件 E、G、S 及 W。這些元素的評估並非根據 **<Stat>**。因此，如果四個指定元件之一有關於刀具長度的計算，則 **<Len>[6]** 至 **<Len>[10]** 的對應值只能為非零。其餘的元件並不會影響結果。尺寸參照至控制器的基本系統（英制或公制）。	

2	<Comp>：	刀長元件（選擇性）			
		資料類型：	STRING		
		字元字串包括兩個子字串，彼此用分號隔開。 一般格式："<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>： ：	**第一個子字串**指定在計算刀具長度時，必須納入考量的刀長元件。 子字串中的字元順序及其標記法（大小寫）可以隨意。可以在字元之間插入任何數量的空格或空白。 **說明：** 子字串中的字元**不允許**重複程式設計。		
			字元：	-	減符號（僅允許為第一個字元） 計算整個刀具長度，減掉下一個字串指定的元件。
				C	接頭或刀座尺寸（無論是這兩者交替元件任一者為使用中刀具有效）
				E	設定偏移量
				G	幾何
				K	動態轉換（僅供一般的 3、4 及 5 軸轉換評估之用）
				S	加總偏移量
				T	具有方向功能的刀把
				W	磨耗
			若第一個子字串為空白（除了空白以外），則允許所有元件計算完整的刀具長度。即使未指定 **<Comp>** 參數也適用。		
		<Substr_2>：	選擇性的**第二個子字串**用來識別座標系統，以便輸出刀具長度。 第二個子字串只包含一個單一的相關字元。		
			字元：	A	可調整座標系統 (ACS)
				B	基本座標系統 (BCS)
				K	動態轉換的刀具座標系統 (KCS)
				M	機械座標系統 (MCS)
				T	刀具座標系統 (TCS)
				W	工件座標系統 (WCS)

			若未指定座標系統，則於 MCS（機械座標系統）中執行評估。若要將任何旋轉納入考量，可以將其指定在 **<Stat>** 中所定義的刀具環境。
3	<_Stat>：	說明刀具環境的資料集名稱（選擇性）	
		資料類型：	STRING
		若是此參數值為空白字串 ("") 或是未指定，則使用目前的狀態。若未指定刀具則使用目前的刀具。	
4	<T>：	刀具的內部 T 編號（選擇性）。	
		資料類型：	INT
		若此參數並未指定或是其值為「0」，則使用儲存在 **<Stat>** 中的刀具。 若此參數值為「-1」，則使用作用中刀具的 T 編號。也可以明確指定作用中刀具的編號。 **說明：** 若未指定 **<Stat>**，則使用實際的狀態為刀具環境。由於 **<T> = 0** 參照至儲存在刀具環境中的 T 編號，在這個環境中使用作用中的刀具，亦即參數 **<T> = 0** 與 **<T> = -1** 在此一特殊案例中具有相同的含義。	
5	<D>：	刀具的刀刃（選擇性）	
		資料類型：	INT
		若此參數並未指定或是其值為「0」，則使用的 D 編號是以 T 編號的來源為基礎。若使用刀具環境中的 T 編號，則一併使用刀具環境的 D 編號，否則就會讀取目前作用中刀具的 D 編號。	
6	<DL>：	根據位置的偏移量編號（選擇性）	
		資料類型：	INT
		若是未指定此參數，則使用的 DL 編號是以 T 編號的來源為基礎。若使用刀具環境中的 T 編號，則一併使用刀具環境的 D 編號，否則就會讀取目前作用中刀具的 D 編號。	

範例

`GETTCOR(_LEN)`	計算目前作用刀具在允許所有元件的機械座標系統中之刀具長度。
`GETTCOR(_LEN,"CGW:W")`	計算作用中刀具的刀具長度，包含了接頭或刀座尺寸、幾何及磨耗。其他元件（例如具有方向功能的刀把或動態轉換）則不予以考慮。在工件座標系統中輸出。

GETTCOR (_LEN,"-K:B")	計算作用中刀具的完整刀具長度，不允許可能作用中動態轉換的長度元件。在基本座標系統中輸出。
GETTCOR (_LEN,":M","Testenv1",,3)	計算在機械座標系統中的完整刀具長度，其中刀具是儲存在刀具環境名稱為「Testenv1」中。不過是以刀刃編號 D3 執行計算，與儲存的刀刃編號無關。

其他資訊

轉接頭轉換/含方向功能的刀把/動態轉換

由轉接頭轉換、含方向功能的刀把及動態轉換所初始化的任何旋轉及元件交換，都是刀具環境的一部分。因此，即使不應該包括在內的對應長度元件也絕對會執行。若想避免發生，必須定義刀具環境，讓對應的轉換不會啟用。在許多情況下（亦即機台在任何時候都未使用轉換或含方向功能的刀把），儲存在刀具環境的資料組會自動滿足這些條件，其結果是使用者完全不需要特殊提供。

車刀及研磨刀具：依據 MD20360 $MC_TOOL_PARAMETER_DEF_MASK 計算刀具長度

下列的機械資料定義了如果使用直徑軸在車刀及研磨刀具上時，磨耗及刀具長度如何評估。

MD20360 $MC_TOOL_PARAMETER_DEF_MASK（刀具參數的定義）。

位元	值	
0	對於車刀及研磨刀具，將橫向軸的**磨耗參數**納入直徑值的考量中：	
	= 0（預設值）	否
	= 1	是
1	對於車刀及研磨刀具，將橫向軸的**刀長元件**納入直徑值的考量中：	
	= 0（預設值）	否
	= 1	是

若有關的位元已設定，相關的輸入內容使用 0.5 係數加權。此加權會反應在 GETTCOR 回傳的刀具長度中。

範例：

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 $MC_DIAMETER_AX_DEF（具移動軸功能的幾何軸）= "X"

X 為直徑軸（標準車床設定）

程式碼	備註
N30 $TC_DP1[1,1]=500	
N40 $TC_DP2[1,1]=2	
N50 $TC_DP3[1,1]=3.0	；幾何 L1
N60 $TC_DP4[1,1]=4.0	
N70 $TC_DP5[1,1]=5.0	
N80 $TC_DP12[1,1]=12.0	；磨耗 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	

動態轉換的長度元件及具有方向功能的刀把

若在刀具長度計算時將**具有方向功能的刀把**納入考量，則以下的向量要包含在計算中：

類型	向量
M	I1 及 I2
T	I1、I2 及 I3
P	刀具長度不受具有方向功能的刀把影響。

在一般的 **5 軸轉換**中，下列的機械資料會包含在轉換類型 24 及 56 的刀具長度計算內：

轉換類型	機械資料
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

轉換類型 56（移動刀具及移動工件）對應於具有方向功能的類型 M 刀把。

對於此一五軸轉換，在前一版本的軟體中，向量 MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2（通道的第一/第二 5 軸轉換之動態偏移量向量）係對應於具有方向功能的類型 M 刀盤的兩個向量 l_1 及 l_3 總和。

只有總和才與兩種情況的轉換相關。至於這兩個個別元件的組成方式，則無關緊要。不過，在計算刀具長度時，應分別指派給刀具和刀盤的元件為何，就有所相關。這也就是為何需要使用機械資料 MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2（刀盤的向量動態偏移量）的原因。它對應至向量 I3。機械資料：MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 不再對應於 I1 及 I3 的總和，而是僅對應至向量 I1。若是機械資料 MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2 等於零，其行為與前相同。

相容性

將 GETTCOR 功能搭配 TOOLENV 及 SETTCOR 功能一併使用，取代部份的功能性，之前是外部運用在測量循環中。

只有實際決定有效刀具長度的特定參數才會運用在測量循環上。上述的功能可以經過設定以重新產生相對於刀具長度計算的測量循環行為。

11.12.6 變更刀具元件 (SETTCOR)

使用 SETTCOR 函式變更刀具元件，並將個別元件評估時會涉及的所有一般條件納入考慮。

說明

有關的專有名詞：若涉及下列與刀具長度相關連的刀具元件，則這些組建完整刀具長度的元件可以從向量的角度加以考慮（例如幾何或磨耗）。該元件包括三項個別值（L1、L2、L3），分別如下稱之為座標值。

因此，刀具元件「幾何」包含三個座標值 $TC_DP3 至 $TC_DP5。

語法

```
<狀態> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>, <GeoAx>,
<Stat>, <T>, <D>, <DL>])
```

意義

SETTCOR(...)：	變更刀具元件的預先定義函式	
	單獨在單節中：	是

<狀態>：	函數傳回值。負值代表錯誤狀態。		
	資料類型：		INT
	值：	0	功能 OK
		-1	沒有保留記憶體供刀具環境使用： MD18116 $MN_MM_NUM_TOOL_ENV = 0 亦即未提供「刀具環境」的功能。
		-2	在 **<Stat>** 中指定名稱的刀具環境不存在。
		-3	參數 **<Comp>** 中的無效字串。 造成此錯誤的肇因可能是無效字元或字元程式設計兩次。
		-4	無效的 T 編號。
		-5	無效的 D 編號。
		-6	無效的 DL 編號。
		-7	嘗試存取不存在的記憶體模組。
		-8	嘗試存取不存在的選項（可程式設計的刀具方向、刀具管理）。
		-9	參數 **<CorComp>** 的數值不合規定。
		-10	參數 **<CorMode>** 的數值不合規定。
		-11	參數 **<Comp>** 與 **<CorComp>** 的內容互相矛盾。
		-12	參數 **<Comp>** 與 **<CorMode>** 的內容互相矛盾。
		-13	**<GeoAx** 參數的內容並未指定幾何軸。
		-14	嘗試寫入不存在的設定偏移量。
參數			

1	<CorVal>： 	修正向量 在 **<Stat>** 定義的工件座標系統 (WCS) 中，套用以下的指派： • **<CorVal>** [0]：橫座標 • **<CorVal>** [1]：縱座標 • **<CorVal>** [2]：備選座標 若是只有一個元件需要修正（亦即無向量修正、參閱參數 **<CorMode>**），修正值一定是在 **<CorVal>**[0]，與軸所作用的對象無關。接著，不評估另兩個元件的內容。 若是 **<CorVal>** 或是 **<CorVal>** 的元件參照至橫向軸，則資料將評估為**半徑尺寸**。也就是說，例如刀具比指定的尺寸「更長」；這會對應地導致工件的直徑變成兩倍大。 尺寸參照至控制系統的**基本系統**（英制或公制）。	
		資料類型：	REAL[3]

2	<Comp>：	刀具元件			
		資料類型：	STRING		
		字元字串包括兩個子字串，彼此用分號隔開。 一般格式："<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>：	**第一個子字串**絕對要提供，可以一個或合併兩個字元。第一個元件的第一個或僅有的字元 (Val_1)，以及第二個元件的第二個字元 (Val_2)，會根據隨後的參數 **<CorComp>** 與 **<CorMode>** 進行處理。		
			字元：	C	接頭或刀座尺寸（無論是這兩者交替元件任一者為使用中刀具有效）
				E	設定偏移量
				G	幾何
				S	加總偏移量
				W	磨耗
		<Substr_2>：	**第二個子字串**為選擇性。或者，它可以合併（個別）文字「W」或「T」。		
			字元：	W	若是第二個子字串為空白，或是包含文字「W」，則偏移量的值會納入考量，如同已經在**工件**座標系統 (WCS) 中已測量。
				T	若是第二個子字串為空白，或是包含文字「T」，則偏移量的值會納入考量，如同已經在**刀具**座標系統（刀具座標系統，TCS）中已測量。
		字串中的字元標記法（大小寫）可以隨意。可以插入任何數量的空格或跳格（空白）。			

3	<CorComp>：	指定即將說明的刀具資料集元件（選擇性）。		
		資料類型：		INT
		值：	0	偏移量數值 **<CorVal>[0]** 參照至在工件座標系統或是刀具座標系統中的參數 **<GeoAx>** 所轉換之幾何軸（另請參閱參數 **<Comp>** 的說明）。這表示說偏移量數值必須在指定的刀具元件中計算，因此在考慮會影響刀具長度計算的所有參數之後，即可得知在指定軸方向上因為指定值所造成的刀具長度變更。 這項變更必須透過在 **<Comp>** 中指定的元件修訂，以及在 **<CorMode>** 中指定的符號演算法來達成（參閱下列參數）。因此，所得的修訂才能對全部的三個軸元件產生效果。
			1	不過，像「0」是向量。向量 **<CorVal>** 的內容參照至工件座標系統或刀具座標系統的橫座標、縱座標及備選座標（參閱參數 **<Comp>** 的說明）。 接下來的參數 **<GeoAx>** 則不評估。
			2	向量偏移量，亦即 L1、L2 及 L3 可以同時變更。 不同於來自「0」與「1」的版本，包含在 **<CorVal>** 中的偏移量數值參照至刀具的 Val_1 元件座標（參閱下列參數 **<CorMode>**）。 相較於工件座標系統，現有刀具的任何可能的傾角都對偏移量毫無影響。
			3 - 5	刀具長度 L1 至 L3（$TC_DP3 至 $TC_DP5）的修正，或是磨耗、設定或附加偏移量的對應值。 偏移量數值包含在 **<CorVal>[0]** 中。它是在刀具的 Val_1 元件座標內測量（參閱以下的參數 **<CorMode>**）。相較於工件座標系統，現有刀具的任何可能的傾角都對偏移量毫無影響。
			6	刀具半徑 ($TC_DP6) 的修正，或是磨耗、設定或附加偏移量的對應值。將機械資料 MD20360 $MC_TOOL_PARAMETER_DEF_MASK 的位元 10 及 11（直徑及 / 或直徑磨耗資料的評估，指定為半徑或直徑）納入考量。
			7 – 11	$TC_DP7 至 $TC_DP11 的修正，或是磨耗、設定或附加偏移量的對應值。將這些參數視同刀具半徑一樣處理。

		若是未指定此參數，則其值為「0」。		
4	<CorMode>：	指定執行寫入操作的類型（選擇性）。		
		資料類型：		INT
		值：	0	Val_{1new} = <CorVal>
			1	Val_{1new} = Val_{1old} + <CorVal>
			2	Val_{1new} = <CorVal> Val_{2new} = 0
			3	Val_{1new} = Val_{1old} + Val_{2old} + <CorVal> Val_{2new} = 0
		Val_{1old} + Val_{2old} 為一種符號標記法。若是兩個元件（由於 **<_Stat>** 的狀態）以不同的方法評估，亦即若是旋轉在兩個元件之間為有效，則 Val_{2old} 在相加之前先行轉換，如此一來所得的刀具長度在刪除 Val_{2new} 之後及 **<CorVal>** 餘數相加之前，都會保持不變。 **<CorVal>** 永遠參照至 Val_1。**<CorVal>** 的值是根據參數 **<Comp>** 的第二部分，在工件座標系統 (WCS) 或是刀具座標系統 (TCS) 中測量。因此，數值均已相對於刀具元件進行轉換及計算。因此，並不能與儲存的數值直接一併計算，而是必須在添加至 Val_1 或 Val_2 前先予以轉換。這代表作用在單一軸或幾個軸上的偏移量不同於由 **<CorComp>** 所定義的偏移量。 對於 **<CorComp> = 0** 的案例，亦即 **<CorVal>** 不含向量而僅含個別數值時，則在測量 **<CorVal>** 的座標 (WCS/TCS) 中，執行說明的操作。尤其是，在版本 2 及 3 中，這也套用至將 Val_{2new} 設定為零。接著，將此結果返回轉換至刀具的座標。這可以表示設定為零（L1、L2、L3）的座標值都沒有變為零；或者先前為零的座標值現在不等於零。不過，若是對應的操作在全部三個幾何軸上成功執行，那就可以保證所要刪除的元件所有三個座標值均為零。若是刀具並未相對於工件座標系統旋轉，或是旋轉時所有的刀具元件仍然保持平行於座標軸（軸交換操作），這樣也可以確認僅有一個刀具座標有變更。 以 **<CorComp> = 0** 連續以任何順序在全部三個座標軸上執行相同的操作 (**<CorMode>**)，等同於使用 **<CorComp>=1** 單次執行相同的操作。 對於參數值「0」及「1」，參數 **<Comp>** 必須包含一個字元，而對於參數值「2」及「3」則是兩個字元。 範例： **<Comp>** 包含字串「ES」，**<CorMode>** 數值為「2」 ⇒ 設定偏移量 $_{new}$ = **<CorVal>**，加總的偏移量 $_{new}$ = 0 若是未指定參數 **<CorMode>**，則其值為「0」。		

5	<GeoAx>：	指定幾何軸的索引，讀取偏移量的值 <CorVal>[0]（選擇性）	
		資料類型：	INT
		值域：	0 ... 2
		索引 0 至 2 參照至目前刀具環境中，有效平面的橫座標、縱座標及備選座標 (G17/G18/G19)。 此參數的內容只有在參數 <CorComp> 的數值為「0」時才會評估。	
6	<Stat>：	說明刀具環境的資料集名稱（選擇性）	
		資料類型：	STRING
		若是此參數值為空白字串 ("") 或是未指定，則使用目前的狀態。若未指定刀具則使用目前的刀具。	
7	<T>：	刀具的內部 T 編號（選擇性）。	
		資料類型：	INT
		若此參數並未指定或是其值為「0」，則使用儲存在 <Stat> 中的刀具。 若此參數值為「-1」，則使用作用中刀具的 T 編號。也可以明確指定作用中刀具的編號。 **說明：** 若未指定 <Stat>，則使用實際的狀態為刀具環境。由於 <T> = 0 參照至儲存在刀具環境中的 T 編號，在這個環境中使用作用中的刀具，亦即參數 <T> = 0 與 <T> = -1 在此一特殊案例中具有相同的含義。	
8	<D>：	刀具的刀刃（選擇性）	
		資料類型：	INT
		若此參數並未指定或是其值為「0」，則使用的 D 編號是以 T 編號的來源為基礎。若使用刀具環境中的 T 編號，則一併使用刀具環境的 D 編號，否則就會讀取目前作用中刀具的 D 編號。	
9	<TL>：	根據位置的偏移量編號（選擇性）	
		資料類型：	INT
		若是未指定此參數，則使用的 DL 編號是以 T 編號的來源為基礎。若使用刀具環境中的 T 編號，則一併使用刀具環境的 D 編號，否則就會讀取目前作用中刀具的 D 編號。若是 T、D 及 DL 指定一個不含位置相依偏移量的刀具，參數 <Comp> 可能不會指定加總或設定偏移量（<狀態> 中的錯誤碼）。	

說明

三個參數 **<Comp>**、**<CorComp>** 及 **<CorMode>** 的所有可能組合均不成立。例如，**<CorComp>** 中的演算法 3 要求在 **<Comp>** 中指定兩個字元。若是指定一個無效的參數組合，則對應的錯誤碼會返回至 **<**狀態**>** 中。

範例

範例 1

程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	；銑削刀具
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP12[1,1]=1.0	；磨耗 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	；==> MCS 位置 X0.000 Y0.000 **Z1.333**
N90 M30	

<CorComp> 為「0」，因此，幾何元件作用在 Z 方向上的座標值必須用偏移量數值 0.333 取代。

所得的總刀具長度則為：L1 = 0.333 + 1.000 = 1.333

範例 2

程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	；銑削刀具
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP12[1,1]=1.0	；磨耗 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	；==> MCS 位置 X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> 為「1」，代表作用在 Z 軸上的偏移量數值 0.333 加總磨耗值 1.0。

所得的總刀具長度則為：L1 = 10.0 + 1.333 = 11.333

範例 3

程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	；銑削刀具
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP12[1,1]=1.0	；磨耗 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	；==> MCS 位置 X0.000 Y0.000 Z0.333
N90 M30	

<CorComp> 為「2」，因此將 Z 軸的有效偏移量輸入幾何元件（將舊值覆蓋），並刪除磨耗值。

所得的總刀具長度則為：L1 = 0.333 + 0.0 = 0.333

範例 4

程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	；銑削刀具
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP12[1,1]=1.0	；磨耗 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	；==> MCS 位置 X0.000 Y0.000 Z11.333
N90 M30	

<CorComp> 為「3」，因此將磨耗值及補正值添加至幾何元件，並刪除磨耗元件。

所得的總刀具長度則為：L1 = 11.333 + 0.0 = 11.333

範例 5

程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	；銑削刀具
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP12[1,1]=1.0	；磨耗 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	；==> MCS 位置 X0.333 Y0.000 Z11.000
N90 M30	

<CorComp> 為「3」，同前一範例所述，但現在將補正作用在幾何軸上，索引為「0」（X 軸），適用於銑削刀具，並且由於 G17 而指派給刀具元件 L3。其結果為呼叫 SETTCOR 時，刀具參數 $TC_DP3 及 $TC_DP12 不受影響。反之將補正值輸入至 $TC_DP5。

範例 6

程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	；車刀
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP4[1,1]=15.0	；幾何 L2
N50 $TC_DP12[1,1]=10.0	；磨耗 L1
N60 $TC_DP13[1,1]=0.0	；磨耗 L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",0,3,1)**	
N110 T1 D1 X0 Y0 Z0	；==> MCS 位置 X24.330 Y0.000 Z17.500
N120 M30	

刀具為車刀。在 N80 中啟用框架旋轉，造成基本座標系統 (BCS) 相對於工件座標系統 (WCS) 旋轉。在 WCS 中，補正值 (N70) 使用索引 1 作用在幾何軸上，亦即 X 軸上，因為 G18 為有效。由於 <CorMode> = 3，在 WCS 的 X 軸方向的刀具磨耗一旦執行 N100 時必須變成零。

因此，在程式結尾的相關刀具參數內容為：

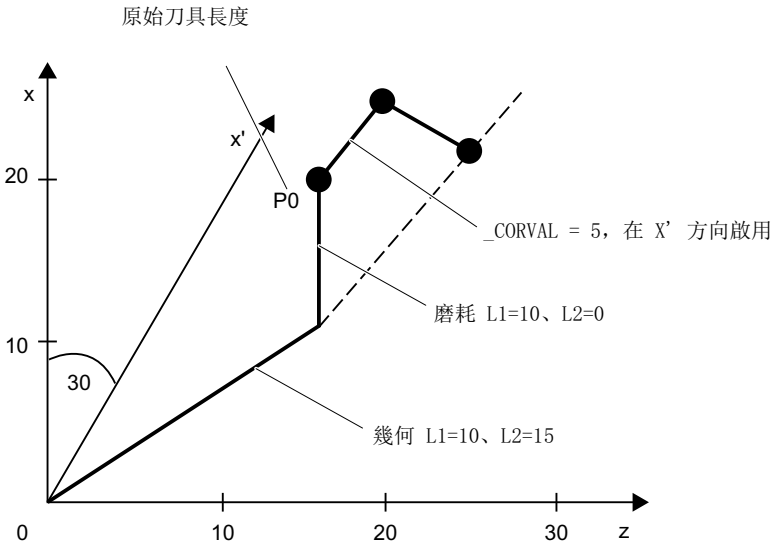
$TC_DP3[1,1]：21.830；幾何 L1

$TC_DP4[1,1]：21.830；幾何 L2

$TC_DP12[1,1]：2.500；磨耗 L1

$TC_DP13[1,1]：-4.330；磨耗 L2

下圖顯示幾何關係：包含 _CORVAL 的總磨耗對映至 WCS 的 X' 方向上。這會產生 P2 點。將此點的座標（在 X/Y 座標上測量）輸入至刀具的幾何元件。差異向量 P_2 - P_1 保留在磨耗中。因此，磨耗在 _CORVAL 方向上不再有分量。



若程式範例在 N110 之後以下列指示繼續執行，則剩餘的磨耗會完全保留在幾何中，因為 Z' 軸的補正現在開始生效（參數 <GeoAx> = 0）：

```
N120 _CORVAL[0]=0.0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0；==> MCS 位置 X24.330 Y0.000 Z17.500
```

由於新的補正值為「0」，因此總刀具長度及逼近 N140 的位置不會變更。若是 _CORVAL 在 N120 不等於「0」，則產生新的總刀具長度及 N140 的新位置，不過刀具長度的磨耗元件將永遠為零，亦即最終的總刀具長度將永遠保留在刀具的幾何元件中。

使用 <CorComp> = 0 參數兩次，經由呼叫 SETTCOR 函式得到的相同結果，也可以藉由只要呼叫 <CorComp> = 1（垂直補正）一次就可以達到：

程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	；車刀
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP4[1,1]=15.0	；幾何 L2
N50 $TC_DP12[1,1]=10.0	；磨耗 L1
N60 $TC_DP13[1,1]=0.0	；磨耗 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	；==> MCS 位置 X24.330 Y0.000 Z17.500
N120 M30	

此例中，刀具的所有磨耗元件在 N100 中第一次呼叫 SETTCOR 之後，立即設定為零。

範例 7

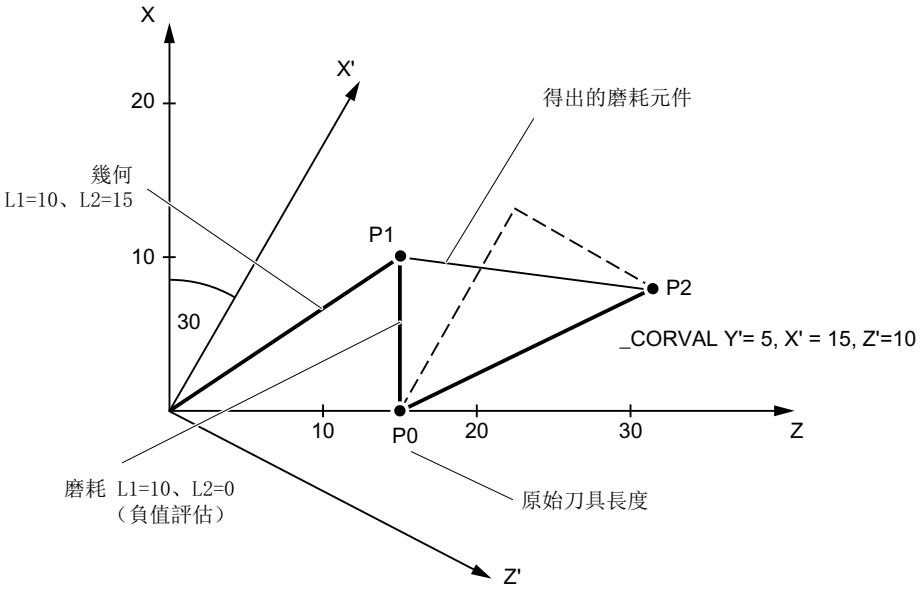
程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	；車刀
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP4[1,1]=15.0	；幾何 L2
N50 $TC_DP12[1,1]=10.0	；磨耗 L1
N60 $TC_DP13[1,1]=0.0	；磨耗 L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",3,3)**	
N110 T1 D1 X0 Y0 Z0	；==> MCS 位置 X25.000 Y0.000 Z15.000
N120 M30	

與範例 6 比較時，參數 <CorComp> = 3，因此可以忽略 <GeoAx> 參數。現在，包含在 _CORVAL[0] 中的值立即作用在刀具元件 L1 上，在 N80 的旋轉並不影響結果，在 $TC_DP12 中的磨耗元件與 _CORVAL[0] 一起包含在幾何元件中。所得的總刀具長度結果會因為 $TC_DP13 而在第一次呼叫 N100 中的 SETTCOR 之後，儲存在刀具的幾何元件中。

範例 8

程式碼	註解
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	；車刀
N30 $TC_DP3[1,1]=10.0	；幾何 L1
N40 $TC_DP4[1,1]=15.0	；幾何 L2
N50 $TC_DP5[1,1]=20.0	；幾何 L3
N60 $TC_DP12[1,1]=10.0	；磨耗 L1
N70 $TC_DP13[1,1]=0.0	；磨耗 L2
N80 $TC_DP14[1,1]=0.0	；磨耗 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	；==> MCS 位置 X7.990 Y25.000 Z31.160
N170 M30	

設定資料：SD42930 $SC_WEAR_SIGN 在 N90 中啟用，亦即評估磨耗時必須為負號。補正為向量 (<CorComp> = 1)，且必須將補正向量加到磨耗中 (<CorMode> = 1)。在 Z/X 平面上的幾何關係如下圖所示：



由於 <CorMode> = 1，刀具的幾何元件仍舊維持不變。定義在 WCS 中的補正向量（繞著 y 軸旋轉）必須包含在磨耗元件中，如此一來圖 3 當中的總刀具長度會參照至 P_2 點。因此，二個點 P_1 與 P_2 之間的距離可以得出所得的刀具磨耗元件。
不過，由於是以負值評估磨耗，由於設定資料 SD42930 $SC_WEAR_SIGN，依此方法判定出的補正必須使用負號輸入至補正記憶體中。因此，在程式結尾的相關刀具參數內容為：

$TC_DP3[1,1]：10.000；幾何 L1（無變更）

$TC_DP4[1,1]：15.000；幾何 L2（無變更）

$TC_DP5[1,1]：10.000；幾何 L3（無變更）

$TC_DP12[1,1]：2.010；磨耗 L1 (= 10 - 15 * cos(30) + 10 * sin(30))

$TC_DP13[1,1]：-16.160；磨耗 L2 (= -15 * sin(30) - 10 * cos(30))

$TC_DP14[1,1]：-5.000；磨耗 L3

設定資料 SD42930 $SC_WEAR_SIGN 在 Y 方向上的 L3 元件效果可以輕易辨識，不會因為框架旋轉而造成其他混淆。

其他資訊

車刀 / 研磨刀具：依據 MD20360 $MC_TOOL_PARAMETER_DEF_MASK 計算刀具長度

下列的機械資料定義了如果使用直徑軸在車刀 / 研磨刀具上時，磨耗及刀具長度如何評估：

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<位元> = <值>

<位元>	<值>	意義
0	0	對於車刀 / 研磨刀具，將橫向軸的**磨耗參數**納入**半徑值**的考量中：
	1	對於車刀 / 研磨刀具，將橫向軸的**磨耗參數**納入**直徑值**的考量中：
1	0	對於車刀 / 研磨刀具，將橫向軸的**刀長元件**納入**半徑值**的考量中：
	1	對於車刀 / 研磨刀具，將橫向軸的**刀長元件**納入**直徑值**的考量中：

若有關的位元已設定，相關的輸入內容使用 0.5 係數加權。使用 SETTCOR 執行修訂，使得總有效刀具長度變更等於 <CorVal> 傳輸的值。計算長度時，若是使用係數 0.5 評估長度作為機械資料 MD20360 $MC_TOOL_PARAMETER_DEF_MASK 的結果，則此元件的補正必須透過傳輸數值二次來完成。

範例

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 2（刀具長度必須使用係數 0.5 在直徑軸中評估）

軸 X 為直徑軸。

```
程式碼                                  註解
N10 DEF REAL _LEN[11]
N20 DEF REAL _CORVAL[3]
N30 $TC_DP1[1,1]=500                    ;刀具類型
N40 $TC_DP2[1,1]=2                      ;刀刃位置
N50 $TC_DP3[1,1]=3。                    ;幾何 - 長度 1
N60 $TC_DP4[1,1]=4。                    ;幾何 - 長度 2
N70 $TC_DP5[1,1]=5。                    ;幾何 - 長度 3
N80 _CORVAL[0]=1。
N90 _CORVAL[1]=1。
N100 _CORVAL[2]=1。
N110 T1 D1 G18 G0 X0 Y0 Z0              ;==> MCS 位置 X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)
N130 T1 D1 X0 Y0 Z0                     ;==> MCS 位置 X2.5 Y6 Z5
N140 R3=$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
```

在每個軸中，刀長補正應為 1 毫米（N80 至 N100）。因此，將 1 毫米加到長度 L2 與 L3 的原始長度中。將兩倍的補正值（2 毫米）加到 L1 的原始刀具長度中，以便依要求變更總長度 1 毫米。若是單節 N110 與 N130 中的逼近點相互比較，可以發現每個軸的位置都變更 1 毫米。

11.13 將刀具長度 L1、L2、L3 的指派讀入座標軸 (LENTOAX)

LENTOAX 函式將有關**啟用**刀具的刀長配置資訊 L1、L2 與 L3 提供給橫座標、縱座標及備選座標幾何座標軸的橫座標、縱座標及備選座標之指派會受框架及啟用中平面之影響 (G17 - G19)。

只考慮刀具 ($TC_DP3[<t>,<d>] 至 $TC_DP5[<t>,<d>]) 的幾何元件，亦即對其他元件的不同軸指派（例如磨耗），其結果不受影響。

語法

<狀態> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])

原則

LENTOAX(...)：	預先定義函式，將作用中刀具的刀具長度 L1、L2 及 L3 指派讀入座標軸		
	單獨在單節中：		是
<狀態>：	函數傳回值。負值代表錯誤狀態。		
	資料類型：		INT
	值：	0	功能 OK <AxInd> 提供的資訊已足供說明（所有的刀長補正平行於幾何軸）。
		1	功能 OK，不過必須評估 <矩陣> 的內容以供正確說明（刀長元件於幾何軸不平行）。
		-1	參數 <Coord> 中的無效字串。
		-2	沒有作用中刀具。
參數			

1	<AxInd>：	如果刀具長度補正平行於幾何軸，則指派給刀具元件 L1 至 L3 的軸索引將返回至 **<AxInd>** 陣列。 • **<AxInd>** [0]：橫座標 • **<AxInd>** [1]：縱座標 • **<AxInd>** [2]：備選座標		
		資料類型：		INT[3]
		值：	0	指派不存在（軸並不存在）
			1 ... 3 或 -1 ... -3	長度編號在對應的座標軸中有效。 若刀長元件指向座標的負向，符號為負。
		若是所有的長度元件並非完全平行或反平行於幾何軸，則包含最多部分刀長元件的軸索引將返回至 **<AxInd>** 中。此例中（若是因為其他原因，函式並未返回錯誤），則返回值為 **<狀態> = 1**。刀長元件 L1 至 L3 對映至幾何軸 1 至 3 則由第 2 參數 **<矩陣>** 的內容完整說明。		
2	<矩陣>：	矩陣代表對座標軸（橫座標、縱座標、備選座標）向量的刀具長度 (L1=1, L2=1, L3=1) 向量，亦即刀具長度元件會依照 L1、L2、L3 的順序指派給列，而軸依照橫座標、縱座標及備選座標的順序指派給行。		
		資料類型：		REAL
		矩陣中的所有元素總是有效，即使是屬於座標軸的幾何軸並未提供，亦即假設在 **<AxInd>** 中的輸入內容為 0。		
3	<Coord>：	指派適用的座標系統（選擇性）		
		資料類型：		STRING
		字元：	MCS M	在機械座標系統中表示刀具長度。
			BCS B	在基本座標系統中表示刀具長度。
			WCS W	在工件座標系統中表示刀具長度（預設設定）。
			KCS K	在動態轉換的刀具座標系統中表示刀具長度。
			TCS T	在刀具座標系統中表示刀具長度。
		字串中的字元標記法（大小寫）可以隨意。 若未指定參數 **<Coord>**，則使用 WCS（預設設定）。		

說明

在 TCS 中所有的刀具長度元件都會平行或反向平行於軸。
啟用鏡射時元件只能反向平行，並且啟動下列的設定資料：
SD42900 $SC_MIRROR_TOOL_LENGTH（鏡射時的刀長符號變更）

範例

標準應用，銑削刀具 G17。

L1 套用至 Z（備選座標）、L2 套用至 Y（縱座標）、L3 套用至 X（橫座標）。

以下列格式函式呼叫：
<狀態>=LENTOAX(<AxInd>,<矩陣>,"WCS")

所得的參數 **<AxInd>** 將包含以下數值：

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

或簡言之：(3, 2, 1)

在此情況中，相關的矩陣（**<**矩陣**>**）為：

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

從 G17 變更至 G18 或 G19 並不會改變結果，因為將長度元件指派給幾何軸的變更與橫座標、縱座標及備選座標的指派，方法完全相同。

現在，框架旋轉繞著 Z 轉 60 度可使用作用中的 G17 程式設計，例如
ROT Z60

備選座標的方向（Z 方向）保持不變；L2 的主元件落在新的 X 軸方向上；L1 的主元件現在落在負 Y 軸方向上。其結果，返回值 (**<**狀態**>**) 為「1」，**<AxInd>** 包含數值 (2, -3, 1)。

在此情況中，相關的矩陣（**<**矩陣**>**）為：

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

路徑移動行為 12

12.1 切線控制

12.1.1 定義耦合 (TANG)

透過預先定義的程序 TANG(...)，旋轉軸之間的切線耦合會定義為跟隨軸，兩個幾合軸會定義為先導軸。跟隨軸持續對齊至沿著主導軸切線的路徑。

說明

耦合係數

耦合係數 1 不必明確地進行程式設計。
切線軸的方向是使用 -1 的耦合係數進行旋轉。

語法

```
TANG(<跟隨軸>, <先導軸_1>, <先導軸_2>, <耦合係數>, <座標系統>, <最佳化>)
```

意義

TANG(...)：	定義切線耦合	
<跟隨軸>：	跟隨軸的軸名稱（旋轉軸）	
	資料類型：	AXIS
	值域：	通道軸名稱
<先導軸_1> <先導軸_2>：	先導軸的軸名稱（幾何軸）[1)]	
	資料類型：	AXIS
	值域：	通道的幾何軸名稱
<耦合係數>：	變更先導軸路徑切線的跟隨軸角度變更係數 n： $角度變更_{跟隨軸} = 角度變更_{路徑切線} * n$	
	資料類型：	REAL
	預設值：	1.0

<座標系統>：	啟用的座標系統 2)		
	資料類型：	CHAR	
	值：	「B」：	基本的座標系統（預設值）
		「W」：	工件座標系統（不適用）
<最佳化>：	最佳化類型		
	資料類型：	CHAR	
	值：	「S」：	標準（預設值） 旋轉軸的動態回應不影響先導軸。假如旋轉軸的動態回應大於所需要的追蹤，則此方法已足夠精確。假如旋轉軸的動態回應不足以跟隨路徑切線的變更，旋轉軸的方向將沿著未定義的倒圓角間隙偏離目標方向。
		「P」：	在計劃先導軸路徑時納入旋轉軸的動態回應。 因此，使用 TANGON() 啟動切線耦合時，必須指定兩個額外參數： • 磨圓間隙 • 角度允差 請參閱「啟用耦合 (TANGON) (頁 558)」章節 **附註** 進行動態轉換時，建議使用最佳化方法「P」。
附註 預設值不必明確地進行程式設計。			

[1] **附註** 作為切線耦合的先導軸，幾何軸必須用在機台座標系統 (MCS) 的程式設計路徑中移動，並參考機台初始位置。例如，迴轉循環 CYCLE800 用在有旋轉頭部的銑床，視循環的設定方式而定，插補將在 WCS 中執行，亦即使用幾何軸 X 與 Y。不過，切線耦合必須定義成以幾何軸作為先導軸，沿著 MCS 中程式設計的路徑移動。因此必須使用機台**非迴轉**狀態的幾何軸做為先導軸。
[2] **附註** 基本座標系統 (BCS) 不得對著 MCS 旋轉。例如，假如 BCS 使用 ROT 指令或迴轉循環 CYCLE800 旋轉，切線控制便不再正確。

12.1.2 啟用中間單節產生 (TLIFT)

假如先導軸程式設計路徑任何位置的跟隨軸切線變更超出機台資料 MD37400 $MA_EPS_TLIFT_TANG_STEP 中參數化的限制，進一步的路徑規劃將取決於角的設定行為。在不使用預先定義的程序 TLIFT(...)，路徑將依照使用 TANG(...) (頁 555) 與 TANGON(...) (頁 558) 程式設計的倒圓角行為移動。

啟用中間單節產生

如果 TLIFT(...) 在 TANG(...) 之後程式設計，控制自動產生的中間單節將於預處理期間偵測到角時插入此點。

程式執行時，先導軸會在達到中間單節時停止。在中間單節中，跟隨軸將以最大軸動態朝跟隨軸路徑切線旋轉。先導軸接著將在程式設計的路徑上移動。

停用中間單節產生

若要停用中間單節產生，切線耦合必須使用 TANG(...) 再次定義，但後續不使用 TLIFT(...) 啟用中間單節產生。

語法

```
TLIFT(<跟隨軸>)
```

意義

TLIFT(...)：	啟用含中間單節計算的角偵測	
<跟隨軸>：	跟隨軸的軸名稱（旋轉軸）	
	資料類型：	AXIS
	值域：	通道軸名稱

跟隨軸旋轉速度

路徑軸

如果跟隨軸已在啟用切線耦合之前作為路徑軸移動，將在作為路徑軸的中間單節執行旋轉移動。

如果使用 FGREF[<軸>]=0.001 指定參考半徑，旋轉移動將以參數化的最大軸速率執行：

MD32000 $MA_MAX_AX_VELO[<跟隨軸>]

定位軸

如果跟隨軸尚未在啟用切線耦合之前作為路徑軸移動，將在作為定位軸的中間單節執行旋轉。

旋轉移動將以參數化的定位軸速率執行：

MD32060 $MA_POS_AX_VELO[<跟隨軸>]

12.1.3 啟用耦合 (TANGON)

透過預先定義的程序 TANGON(...)，先前使用 TANG(...) (頁 555) 定義的切線耦合將啟用。跟隨軸接著在後續移動時持續對齊至路徑切線。

跟隨軸的角度

相對於路徑切線的跟隨軸角度取決於 TANG(...) 指定的轉換率、機械資料 MD37402 $MA_TANG_OFFSET 中參數化的偏移角，及 TANGON(...) 所指定額外適用的偏移角。

最佳化「P」

如果在切線耦合 (TANG(...)) 定義中指定值「P」為最佳化參數，啟用耦合時必須設定參數「磨圓間隙」和選配參數「角度允差」。

如果角度允差指定 0 值，將只會啟用參數「磨圓間隙」。

如果角度允差指定大於 0 的值，啟用的磨圓間隙將產生自參數化磨圓間隙最小值和以參數化角度允差為基礎的磨圓間隙。

如果跟隨軸的動態回應不足以跟隨參數化條件，將隨之降低先導軸的路徑速率。

語法

TANGON(<跟隨軸>, <偏移角>, <磨圓間隙>, <角度允差>)

意義

TANGON(...)：	啟動切線耦合	
<跟隨軸>：	跟隨軸的軸名稱（旋轉軸）	
	資料類型：	AXIS
	值域：	通道軸名稱
<偏移角>：	跟隨軸與路徑切線的偏移角度 參考點為旋轉軸的零點。	
	資料類型：	REAL
<磨圓間隙>：	最大允許的磨圓間隙 如果磨圓間隙因動態條件增加，先導軸的路徑速率將下降。	
	資料類型：	REAL
<角度允差>：	跟隨軸零設定與路徑切線間指定角度的最大允許允差	
	資料類型：	REAL

12.1.4 停用耦合 (TANGOF)

透過預先定義的程序 TANGOF(...)，使用 TANG(...) (頁 555) 定義並使用 TANGON(...) (頁 558) 啟用的切線耦合將停用。跟隨軸將不再對齊至沿著主導軸切線的路徑。但跟隨軸與先導軸的耦合在停用後仍保留，因此無法使用下列函數，例如：

- 平面變更
- 幾何軸切換
- 定義跟隨軸的新切線耦合

跟隨軸與先導軸耦合連接的最終取消，需使用 TANGDEL(...) (頁 560) 刪除耦合後才完成。

程式設計

```
TANGOF(<跟隨軸>)
```

意義

TANGOF(...)：	停用切線耦合	
<跟隨軸>：	跟隨軸的軸名稱（旋轉軸）	
	資料類型：	AXIS
	值域：	通道軸名稱

12.1.5 刪除耦合 (TANGDEL)

使用 TANG(...) (頁 555) 定義的切線耦合即使在使用 TANGOF(...) (頁 559) 停用切線耦合後仍會保留。現有的切線耦合將繼續使下列函數無法使用，例如：

- 平面變更
- 幾何軸切換
- 定義跟隨軸的新切線耦合

使用預先定義的程序 TANGDEL(...)，現有的切線耦合將在使用 TANGOF(...) 停用切線耦合後刪除。

語法

```
TANGDEL(<跟隨軸>)
```

意義

TANGDEL(...)：	刪除使用 TANG() 定義的切線耦合	
	有效：	非模態
<跟隨軸>：	將刪除切換耦合的跟隨軸軸名稱	
	資料類型：	AXIS
	值域：	通道軸名稱

範例

先導軸變更

在使用跟隨軸的其他先導軸定義新切線耦合之前，必須先刪除現有的切換耦合。

程式碼	註解
N10 **TANG**(A, X, Y, 1)	；定義跟隨軸 A 的切線耦合：A 到 X 與 Y
N20 TANGON(A)	；啟用跟隨軸 A 的切線耦合
N30 X10 Y20	
...	
N80 TANGOF(A)	；停用跟隨軸 A 的切線耦合
N90 **TANGDEL**(A)	；刪除跟隨軸 A 的切線耦合
...	
N120 **TANG**(A, X, Z)	；定義跟隨軸 A 的新切線耦合
N130 TANGON(A)	；啟用跟隨軸 A 的新切線耦合
...	

幾何軸切換

執行現有耦合的幾何軸切換之前，必須先刪除耦合。

程式碼	註解
N10 GEOAX(2,Y1)	；第二幾何軸 = 機械軸 Y1
N20 TANG(A, X, Y)	；定義跟隨軸 A 的切線耦合
N30 TANGON(A, 90)	；啟用跟隨軸 A 的切線耦合
N40 G2 F8000 X0 Y0 I0 J50	；動作單節
N50 TANGOF(A)	；停用跟隨軸 A 的切線耦合
N60 **TANGDEL**(A)	；刪除跟隨軸 A 的切線耦合
N70 **GEOAX**(2, Y2)	；第二幾何軸 = 機械軸 Y2
N80 TANG(A, X, Y)	；定義跟隨軸 A 的新切線耦合
N90 TANGON(A, 90)	；啟用跟隨軸 A 的新切線耦合
...	

12.2 進給率特性 (FNORM, FLIN, FCUB, FPO)

為了允許進給率特性的彈性定義，根據 DIN 66025 程式設計的進給率已由線性和三次特性加以延伸。

三次特性可以直接或做為插補曲線進行程式設計。 由於這些附加的特性，使連續的平滑化速度特性得以依照待加工工件的曲率進行程式設計。

由於這些附加的特性，使連續的平滑化速度特性得以依照待加工工件的曲率進行程式設計。

語法

```
F... FNORM
F... FLIN
F... FCUB
F=FPO(...,...,...)
```

含義

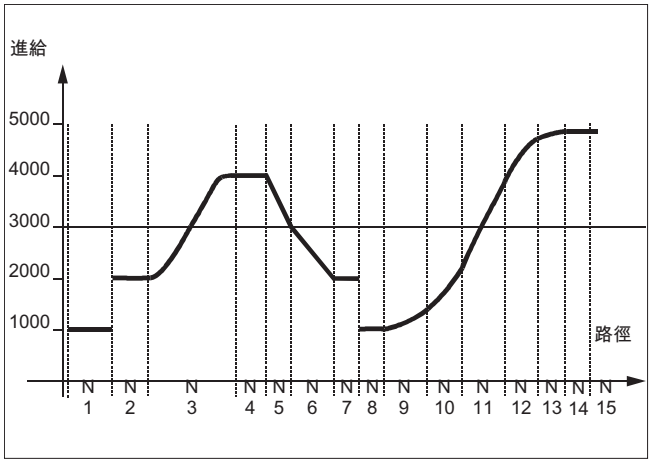
FNORM：	基本設定。 進給值會被指定為該單節中移動路徑的函數，隨後成為一個有效的模態值。
FLIN：	路徑速率輪廓**線性**： 進給值是經由在單節開始和單節結束目前值的移動路徑，以線性方式逼近，然後成為有效的模態值。 回應可以結合 G93 和 G94。
FCUB：	路徑速率輪廓**三次**： 整個單節已程式設計的 F 值（相對於單節結尾）是以曲線連接。 曲線是與前後定義的進給率形成切線開始及結束，並使用 G93 和 G94 生效。 如果單節中少了 F 位址，就使用要已程式設計的最後 F 值。
F=FPO…：	**多項式**路徑速率輪廓： F 位址經由多項式定義從目前值到單節結尾的進給特性。 此後結束值即為有效模態值。

曲線路徑區段上的進給最佳化

進給多項式 F=FPO 和進給曲線 FCUB 應該永遠都以恆定切削速率 CFC 移動，從而允許產生無震動設定點量變曲線。 這樣才能夠建立持續加速度設定點進給輪廓。

範例： 各種不同的進給輪廓

這個範例為您示範，各種不同範例輪廓的程式設計和圖形表示。



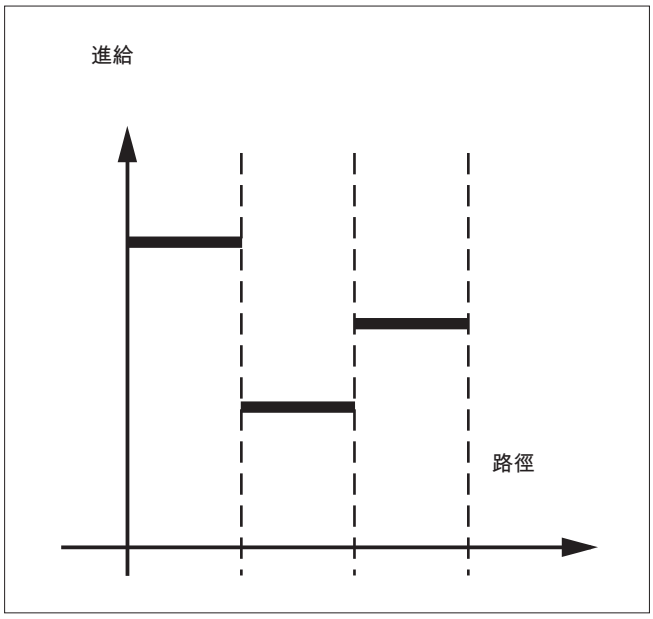
程式碼	註解
N1 F1000 FNORM G1 X8 G91 G64	；恆定進給率輪廓，增量尺寸資料
N2 F2000 X7	；設定點速率步進變更
N3 F=FPO（4000，6000，-4000）	；在單節結尾使用進給 4000 經由多項式進給輪廓
N4 X6	；多項式進給率 4000 為有效模態值
N5 F3000 FLIN X5	；線性進給率輪廓
N6 F2000 X8	；線性進給率輪廓
N7 X5	；線性進給率有效模態值
N8 F1000 FNORM X5	；含加速度步進變更的恆定進給率
N9 F1400 FCUB X8	；在單節中已程式設計的下列所有 F 值是與曲線連接
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	；轉出曲線外形
N14 FNORM X5	
N15 X20	

其他資訊

FNORM

根據 DIN 66025，進給位址 F 將路徑進給率定義為常數值。

請參閱程式設計手冊「基本概念」，取得有關此主題的詳細資訊。

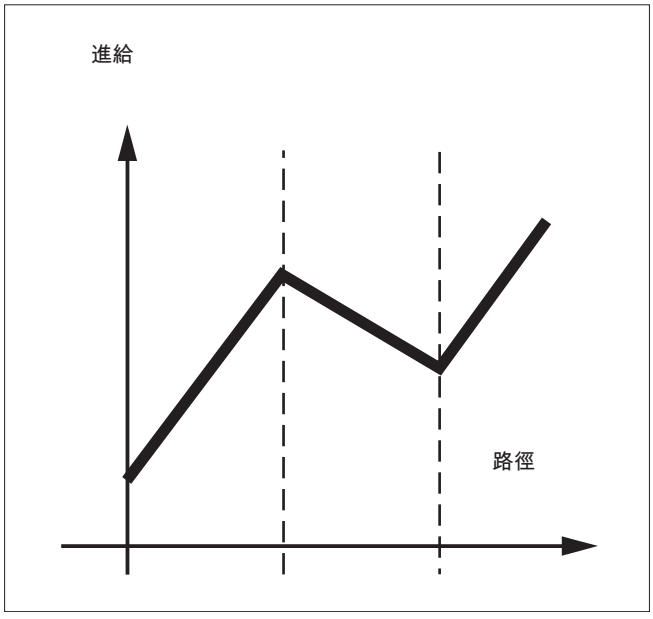


FLIN

進給率特性是以線性方式，從目前進給率值逼近已設定的 F 值，直到單節結尾。

範例：

```
N30 F1400 FLIN X50
```

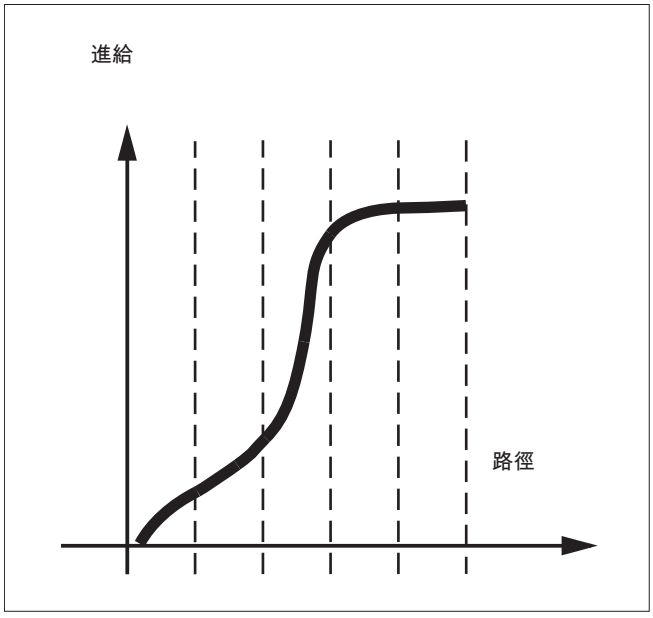


FCUB

進給率根據三次特性，從目前進給率值逼近已設定的 F 值，直到單節結尾。 控制使用曲線，連接具有效 FCUB 且以非模態方式設定的所有進給率值。 進給率值在這裡是做為插補點，以進行曲線插補的計算。

範例：

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```



F=FPO（...，...，...）

進給率特性是經由多項式直接進行程式設計。 多項式係數是根據用於多項式插補的相同方法指定。

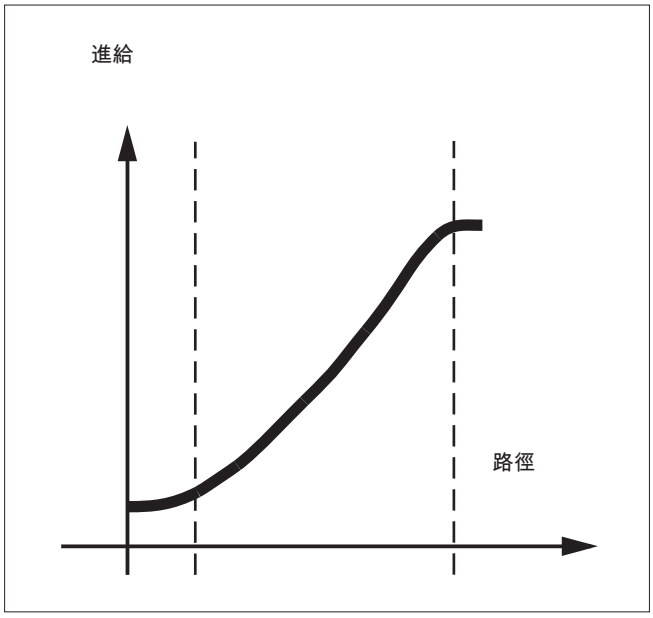
範例：

```
F=FPO（endfeed，quadf，cubf）
```

endfeed、quadf 和 cubf 是事先定義的變數。

endfeed：	單節結尾的進給率
quadf：	二次多項式係數
cubf：	三次多項式係數

使用有效 FCUB，曲線是沿切線連結至在單節開始和單節結束時經由 FPO 定義的特性。



補充條件

- 不管設定的進給率特性為何，編寫路徑移動特性的函數都適用。
- 不管是 G90 或 G91，設定的進給率特性永遠都是絕對的。
- 進給特性曲線 FLIN 及 FCUB **不會**與 G93 及 G94 作用於 G95、G96/G961 及 G97/G971。
- 使用已啟用的壓縮 COMPON，會在多個單節結合形成曲線區段時，套用下列各項：

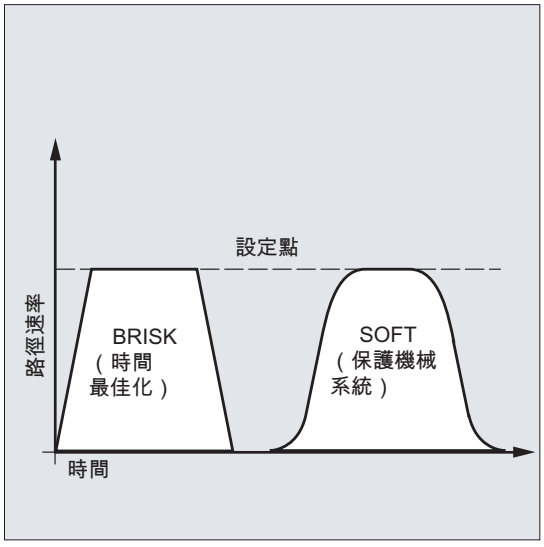
FNORM：	群組中最後單節的 F 字組套用於曲線區段。
FLIN：	群組中最後單節的 F 字組套用於曲線區段。 已程式設計的 F 值一直套用到區段結尾，然後再以線性方式逼近。
FCUB：	產生的進給率曲線偏離設定之終點，程度不超過機械資料 MD20172 $MC_COMPRESS_VELO_TOL 中所設定的數值。
F=FPO(…,…,…)：	這些單節不會進行壓縮。

12.3 加速度行為

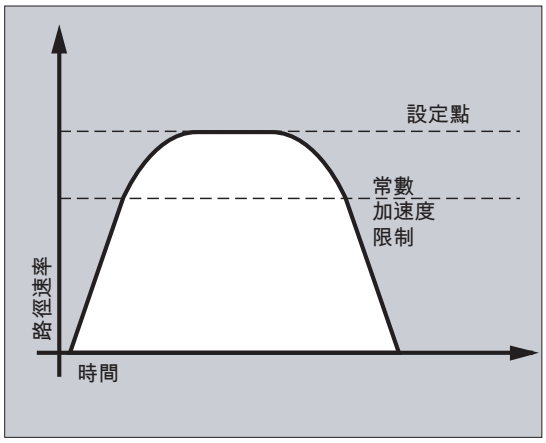
12.3.1 加速模式（BRISK、BRISKA、SOFT、SOFTA、DRIVE、DRIVEA）

下列工件程式指令可用於程式設計目前的加速模式：

- 「BRISK、BRISKA」
  單一軸或路徑軸以最大加速度進行移動直到達到程式設計的進給率為止（**不含震動限制的加速**）。
- 「SOFT、SOFTA」
  單一軸或路徑軸以恆定加速度進行移動直到達到程式設計的進給率為止（**含震動限制的加速**）。
- 「DRIVE、DRIVEA」
  單一軸或路徑軸以最高不超過程式設計之速率限制的加速度進行移動（MD 設定！）。 接著加速會降低（MD 設定）直到達到程式設計的進給率為止。



圖像 12-1　使用 BRISK 與 SOFT 的路徑速率曲線



圖像 12-2　使用 DRIVE 的路徑速率曲線

語法

```
BRISK
BRISKA（<軸 1>，<軸 2>，...）
SOFT
SOFTA（<軸 1>，<軸 2>，...）
DRIVE
DRIVEA（<軸 1>，<軸 2>，...）
```

含義

BRISK：	用於啟動路徑軸的「不含震動限制的加速」功能指令。
BRISKA：	用於啟用單一軸移動的「不含震動限制的加速」功能指令（JOG、JOG/INC、定位軸、擺動軸等）。
SOFT：	用於啟動路徑軸的「含震動限制的加速」功能指令。
SOFTA：	用於啟用單一軸移動的「含震動限制的加速」功能指令（JOG、JOG/INC、定位軸、擺動軸等）。
驅動器：	用於啟用超過設定之路徑軸速率限制的減低加速指令（MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT）。
DRIVEA：	用於啟用超過設定之單一軸移動的速率限制 (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) 的減低加速指令（JOG、JOG/INC、定位軸、擺動軸，等）。
（<軸 1>，<軸 2> 等）：	套用已呼叫之加速模式的單一軸。

補充條件

在加工時變更加速模式

若加工時加速模式在工件程式中有變更（BRISK ↔ SOFT），則即在使用連續路徑模式進行轉換時，在單節結束時會有具精確停止的單節變更。

範例

範例 1： SOFT 及 BRISKA

程式碼
N10 G1 X... Y... F900 SOFT
N20 BRISKA(AX5,AX6)
...

範例 2： DRIVE 及 DRIVEA

程式碼
N05 DRIVE
N10 G1 X... Y... F1000
N20 DRIVEA (AX4, AX6)
...

參考資料

功能手冊，基本功能；加速（B2）

12.3.2 在跟隨軸（VELOLIMA，ACCLIMA，JERKLIMA）上加速的影響

在軸耦合的情況下（切線修正、耦合動作、主從耦合、電子齒輪；請參閱 「軸耦合 (頁 603)」）。

跟隨軸/主軸的動態臨界值可利用工件程式或同步動作的 VELOLIMA、ACCLIMA、及 JERKLIMA 功能加以控制，即便已啟用軸耦合亦同。

說明

JERKLIMA 功能並不適用於所有類型的耦合。

參考資料：

- 功能手冊，特殊功能；軸耦合（M3）
- 功能手冊，延伸功能；同步主軸（S3）

說明

SINUMERIK 828D 的支援度

VELOLIMA、ACCLIMA 及 JERKLIMA 功能只有在 SINUMERIK 828D 搭配「耦合動作」的功能下，方能使用！

語法

```
VELOLIMA（<軸>）=<值>
ACCLIMA（<軸>）=<值>
JERKLIMA（<軸>）=<值>
```

含義

VELOLIMA：	用於修正參數化最大**速率**的指令
ACCLIMA：	用於修正參數化最大**加速度**的指令
JERKLIMA：	用於修正參數化最大**震動**的指令
<軸>：	動態臨界值需要修正的跟隨軸
<值>：	修正值百分比

範例

範例 1： 修正跟隨軸（AX4）的動態臨界值

程式碼	註解
...	
VELOLIMA[AX4]=75	；將儲存在機械資料中的最大軸速率臨界值修正為 75%
ACCLIMA[AX4]=50	；將儲存在機械資料中的最大軸加速度臨界值修正為 50%
JERKLIMA[AX4]=50	；將儲存在機械資料中的最大軸震動臨界值修正為 50%
...	

範例 2：電子齒輪

座標軸 4 係透過「電動齒輪」耦合，耦合至座標軸 X。 跟隨軸的加速度大小限制在最大加速度的 70%。 可允許的最大速率限制在最大速率的 50%。 一旦成功啟用耦合後，便會將最大容許速率重設為 100%。

程式碼	註解
...	
N120 ACCLIMA[AX4]=70	；已降低的最大加速度。
N130 VELOLIMA[AX4]=50	；已降低的最大速率。

程式碼	註解
...	
N150 EGON（AX4,"FINE",X,1,2）	；啟用 EG 耦合。
...	
N200 VELOLIMA[AX4]=100	；完全最大速率。
...	

範例 3： 透過靜態同步動作影響主動值耦合

座標軸 4，透過主動值耦合，與 X 耦合。 透過靜態同步動作 2，加速度回應從位置 100 被限制在位置 80%。

程式碼	註解
...	
N120 IDS=2 WHENEVER $AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80	；同步動作
N130 LEADON(AX4, X, 2)	；主動值耦合開啟
...	

12.3.3 啟用技術專屬動態值（DYNNORM、DYNPOS、DYNROUGH、DYNSEMIFIN、DYNFINISH）

使用「技術」G 群組，可針對五個不同的技術加工步驟，啟用合適的動態回應。

動態值及 G 指令可視機械資料的設定值（機台製造商）加以設定。

參考資料：
功能手冊，基本功能、連續路徑模式，精確停止，預讀控制（B1）

語法

啟動動態值：
```
DYNNORM
DYNPOS
DYNROUGH
DYNSEMIFIN
DYNFINISH
```

說明

動態值已於單節中啟用，而其中相關的 G 指令已被程式製作。 加工未停止。

讀取或寫入指定欄位元件：
```
R<m>=$MA...[n,X]
$MA...[n,X]=<值>
```

含義

DYNNORM：	用於啟動**正常動態回應的 G 指令**	
DYNPOS：	用於啟動**定位模式 攻牙**之動態回應的 G 指令	
DYNROUGH：	用於啟用**粗加工**之動態回應的 G 指令	
DYNSEMIFIN：	用於啟用**精加工**之動態回應的 G 指令	
DYNFINISH：	用於啟用**平滑精加工**之動態回應的 G 指令	
R<m>：	具編號 <m> 的 R 參數	
$MA...[n,X]：	具有欄位元件的機械資料會影響動態回應	
<n>：	陣列索引	
	值域：	0 ... 4
	0	正常動態回應（DYNNORM）
	1	定位模式（DYNPOS）的動態回應
	2	粗加工（DYNROUGH）的動態回應
	3	半精加工（DYNSEMIFIN）的動態回應
	4	平滑精加工（DYNFINISH）的動態回應
<X> ：	軸位址	
<值>：	動態值	

範例

範例 1： 啟動動態值

程式碼	註解
DYNNORM G1 X10	；初始設定
DYNPOS G1 X10 Y20 Z30 F...	；定位模式，攻牙
DYNROUGH G1 X10 Y20 Z30 F10000	;粗加工
DYNSEMIFIN G1 X10 Y20 Z30 F2000	;精加工
DYNFINISH G1 X10 Y20 Z30 F1000	；平滑精加工

範例 2： 讀取或寫入指定欄位元件

粗加工，座標軸 X 的最大加速度

程式碼	註解
R1=$MA_MAX_AX_ACCEL[2, X]	；讀取
$MA_MAX_AX_ACCEL[2, X]=5	；寫入

12.4 具有前饋控制的移動 (FFWON, FFWOF)

前饋控制減少了對當輪廓朝零點前進時，發生依速率而過度移動的情況。 使用前饋控制移動能容許更高的路徑精度並改善工結果。

語法

```
FFWON
FFWOF
```

含義

FFWON：	下指令來**啟動**前饋控制
FFWOF：	下指令來**停用**前饋控制

說明

前饋控制的類型，以及要使用哪條路徑軸來配合前饋控制移動，係透過機械資料來說明。
預設值： 依速率的前饋控制
選項： 依加速度的前饋控制

範例

程式碼

```
N10 FFWON
N20 G1 X... Y... F900 SOFT
```

12.5 可程式設計的輪廓精確度 (CPRECON, CPRECOF)

「可程式設計的輪廓精確度」功能透過速率自動調整，減少了曲線輪廓上的路徑錯誤。

透過設定資料 $SC_CONTPREC 或已設定的輪廓允差 CTOL，依機台組態（MD20470 $MC_MC_CPREC_WITH_FFW；請參閱機台製造商的規格）指定欲維持的輪廓精確度。若數值和幾何軸的 K_V 係數愈小，曲線輪廓上路徑進給率降低的幅度愈大。

經由 NC 程式中的 CPRECON 和 CPRECOF 指令，可啟用或停用「可程式設計的輪廓精確度」功能。

語法

```
CPRECON
...
CPRECOF
```

含義

CPRECON：	G 指令呼叫：**啟用**「可程式設計的輪廓精確度」	
	有效：	模態
CPRECOF：	G 指令呼叫：**停用**「可程式設計的輪廓精確度」	
	有效：	模態

CPRECON 和 CPRECOF 一同組成 G 功能群組 39（可程式設計的輪廓精確度）。

說明

使用者可透過設定資料 $SC_MINFEED（使用 CPRECON 指令的最小路徑進給率）指定路徑進給率的最小速率。

低於此值進給率不受限制，除非設定了較低的 F 值，或軸的動態臨界值需要較低的路徑速率。

說明

「可程式設計的輪廓精確度」功能僅考慮路徑的幾何軸。不會影響定位軸的速率。

範例

程式碼	註解
N10 G0 X0 Y0	
N20 CPRECON	；啟用「可程式設計的輪廓精密度」。
N30 G1 G64 X100 F10000	；以 10 米/分連續路徑模式加工。
N40 G3 Y20 J10	；圓弧單節中的自動進給限制。
N50 G1 X0	；進給率再次不受限制（10 米 / 分）。
...	
N100 CPRECOF	；停用「可程式設計的輪廓精密度」。
N110 G0 ...	

參考資料

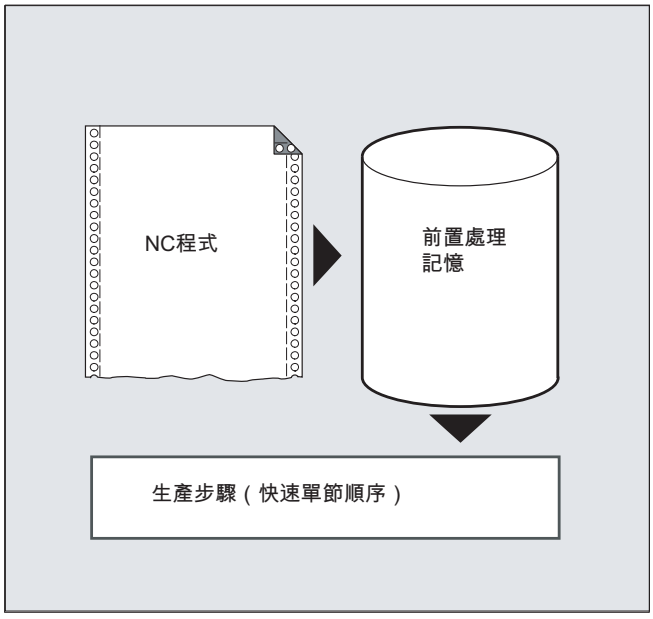
有關 CTOL 的程式設計，請參閱 「程式設計輪廓 / 方向允差 (CTOL, OTOL, ATOL) (頁 597)」

至於「可程式設計的輪廓精確度」功能的詳細資訊，請參閱：

功能手冊，特殊功能；輪廓通道監控 (K6)，章節：「可程式設計的輪廓精確度」

12.6 具有前置處理記憶體（STOPFIFO，STARTFIFO，FIFOCTRL，STOPRE）的程式順序

依其擴充程度而定，控制系統具有一定數量的所謂前置處理記憶體，程式執行之前已準備的單節即儲存在其中，然後在加工進行中，以高速單節順序輸出。 這些順序可以讓短路徑以高速率移動。 如果有足夠的剩餘控制時間可供使用，前置處理記憶體永遠都會填滿。



指定加工步進

要在前置處理記憶體中緩衝的加工步驟之開頭與結尾，會各自以 STOPFIFO 與 STARTFIFO，在工件程式中定義。 前置處理的程序，以及緩衝的單節，只會在 STARTFIFO 指令之後開始，或是如果是前置處理記憶體滿載。

自動前置處理記憶體控制

自動前置處理記憶體控制係以 FIFOCTRL 指令來呼叫。「FIFOCTRL」最初的作用就像是「STOPFIFO」。 不論怎麼程式設計，都不會開始處理，直到前置處理記憶體滿載為止。 然而，對空的前置處理記憶體的回應會有所不同： 以「FIFOCTRL」，路徑速率會逐漸降低，直到填滿層級達到 2/3，才能防止全空以及減速到靜止狀態。

前置處理停止

在單節中程式設計「STOPRE」指令，會停止單節前置處理與緩衝。 一直到所有前置處理以及儲存的單節，都已經被執行了，才會執行下列單節。 前面的單節會以精確停止暫停（如同使用 G9）。

注意
程式放棄 若已啟用刀具偏移量或曲線插補時，不可程式設計「STOPRE」指令，因為這會導致連續單節順序中斷。

語法

表格 12-1　定義加工步驟：

```
STOPFIFO
...
STARTFIFO
```

表格 12-2　自動前置處理記憶體控制：

```
...
FIFOCTRL
...
```

表格 12-3　前置處理停止：

```
...
STOPRE
...
```

說明

「STOPFIFO」、「STARTFIFO」、「FIFOCTRL」及「STOPRE」必須於本身的單節中程式設計。

含義

STOPFIFO：	「STOPFIFO」對要在前置處理記憶體中，被緩衝的加工步驟之起點，進行定義。「STOPFIFO」停止處理並填滿前置處理記憶體，直到： • 「STARTFIFO」或「STOPRE」被偵測到 或 • 前置處理記憶體滿載 或 • 到達程式結尾
STARTFIFO：	「STARTFIFO」開始加工步驟的快速處理；前置處理記憶體以平行於此的方式被填滿。
FIFOCTRL：	啟用自動前置處理記憶體控制系統
STOPRE：	停止前置處理

說明

前置處理記憶體未填滿，或是，如果加工步驟包含需要取消緩衝操作的指令（搜尋參照、量測函數等），填入會中斷。

說明

控制會在對狀態資料（$SA...）進行存取的事件中，產生一個內部的前置處理停止。

範例： 停止前置處理

程式碼	註解
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	；使用探針於第一個量測輸入及線性插補量測單節。
N40 STOPRE	；前置處理停止。
...	

12.7 定義一個停止延遲範圍 (DELAYFSTON, DELAYFSTOF)

使用預先定義的 DELAYFSTON 及 DELAYFSTOF 程序，在工件程式中定義條件式可中斷的範圍（停止延遲範圍）。

說明

DELAYFSTON 與 DELAYFSTOF 在同步動作中並不認同！

語法

```
DELAYFSTON
...
DELAYFSTOF
```

意義

DELAYFSTON：	定義停止延遲範圍的開始	
	單獨在單節中：	是
DELAYFSTOF：	定義停止延遲區域的結束	
	單獨在單節中：	是

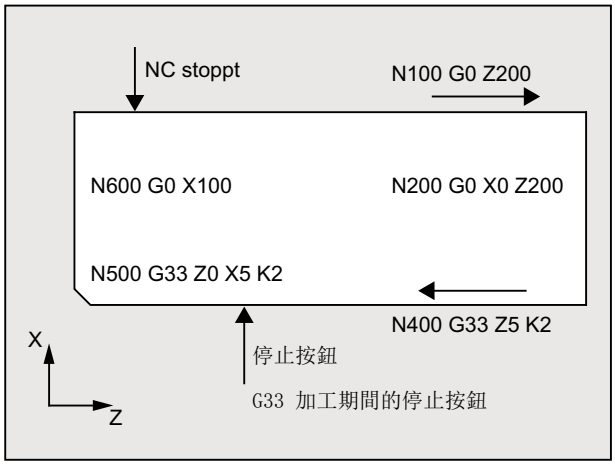
程式設計範例

後面的程式單節是在迴圈中重複：

程式碼

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

在下列圖表中，可以看到使用者在停止延遲範圍中按下「停止」，然後 NC 在停止延遲範圍之外開始煞車，亦即在單節 N100 中。因而導致 NC 在 N100 開始時停止。



其他資訊

子程式結束

DELAYFSTOF 是在呼叫 DELAYFSTON 的子程式結束時，以隱含方式啟動。

巢狀

如果子程式 1 在停止延遲區域中呼叫子程式 2，則整個子程式 2 都是停止延遲區域。尤其是子程式 2 的 DELAYFSTOF 沒有作用。

範例：

程式碼	註解
N10010 **DELAYFSTON**	；使用 N10xxx 程式層級 1 的單節。
N10020 R1 = R1 + 1	
N10030 G4 F1	；開始停止延遲區域。
...	
N10040 subprogram2	
...	
...	；子程式 2 的解譯。
N20010 **DELAYFSTON**	；不生效，重複開始，第二層級。
...	
N20020 **DELAYFSTOF**	；不生效，在其他層級結束。
N20030 RET	
N10050 **DELAYFSTOF**	；相同等級下的停止延遲範圍結尾。
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	；結束停止延遲區域。從現在起，停止立即運作。

系統變數

可查詢下列系統變數，以判定工件程式處理目前是否在停止延遲區中：

- 在工件程式中使用 $P_DELAYFST
- 在同步動作中使用 $AC_DELAYFST

值	意義
0	延遲停止範圍未啟用
1	延遲停止區域啟用

12.8 防止 SERUPRO 的程式定位 (IPTRLOCK, IPTRUNLOCK)

在機台上某些複雜的加工情況下，必須停止單節搜尋 SERUPRO。

透過使用可程式設計的中斷指示器，可以在無法追蹤點之前干預「在中斷點搜尋」。

也可以在 NC，定義無法追蹤的區段。當程式被中斷時，NC 會記下最後處理的單節，然後可以經由 HMI 使用者介面進行搜尋。

語法

```
IPTRLOCK
IPTRUNLOCK
```

這些指令位於工件程式行中，並允許可程式設計的中斷指示器

含義

IPTRLOCK：	無法追蹤的程式區段開始
IPTRUNLOCK：	無法追蹤的程式區段結尾

只有在工件程式中才認可這兩個指令，但在同步動作中並不認可。

範例

將兩個程式層級中無法追蹤的程式區段，使用隱含的「IPTRUNLOCK」組成巢狀嵌套。子程式 1 中的隱含「IPTRUNLOCK」會結束無法追蹤的區段。

程式碼	註解
N10010 IPTRLOCK ( )	
N10020 R1 = R1 + 1	
N10030 G4 F1	；暫停開始搜尋停用程式區段的單節。
...	
N10040 subprogram2	
...	；子程式 2 的解譯。
N20010 IPTRLOCK ( )	；沒有作用，重複開始。
...	
N20020 IPTRUNLOCK ( )	；不生效，在其他層級結束。
N20030 RET	
...	
N10060 R2 = R2 + 2	
N10070 RET	；搜尋停用程式區段結束。
N100 G4 F2	；繼續主程式。

中斷指示器就會再度在 100 產生中斷。

其他資訊

取得及尋找無法追蹤的區段

非可搜尋程式區段是使用語言指令「IPTRLOCK」和「IPTRUNLOCK」進行識別。

「IPTRLOCK」指令會在主要執行中可執行的單一單節凍結中斷指示器（SBL1）。 下文中將此單節稱為暫停單節。 如果在「IPTRLOCK」之後取消程式，此暫停單節可以從 HMI 使用者介面進行搜尋。

從目前單節繼續

中斷指示器是使用「IPTRUNLOCK」放置於目前的單節上，做為後續程式區段的中斷點。

找到搜尋目標之後，就可以使用暫停單節重複新的搜尋目標。

必須再度經由 HMI 移除使用者編輯的中斷指示器。

巢狀的規則：

利用巢狀嵌套與子程式結尾，透過以下各點規範程式語言指令「IPTRLOCK」及「IPTRUNLOCK」之間的互動。

1. 「IPTRLOCK」是在呼叫「IPTRUNLOCK」的子程式結尾，以隱含方式啟動。
2. 在停止延遲區段中的「IPTRLOCK」沒有作用。
3. 如果子程式 1 在無法追蹤的區段中呼叫子程式 2，整個子程式 2 都會無法追蹤。 特別是「IPTRUNLOCK」在子程式 2 中就會沒有作用。

如需詳細資訊，請參閱
/FB1/ 功能手冊，基本功能；模式群組、通道、程式操作模式（K1）。

系統變數

無法追蹤的區段可以在工件程式中，使用「$P_IPTRLOCK」進行偵測。

自動中斷指示器

自動中斷指示器會自動將先前已定義的耦合類型定義為無法追蹤。 以下項目的機械資料

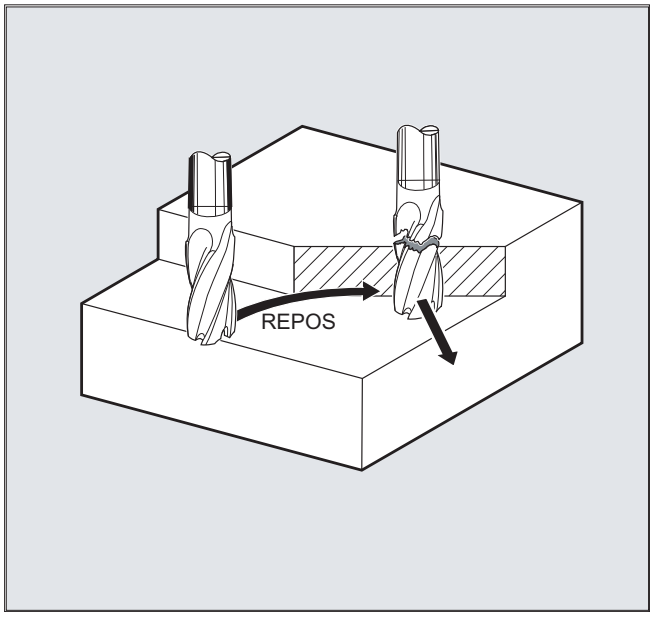
- 「EGON」的電子齒輪
- 「LEADON」的軸主動值耦合

會用來啟動自動中斷指示器。 如果已程式設計的中斷指示器與使用自動中斷指示器啟動的中斷指示器覆疊，就會產生可能的最大無法追蹤區段。

12.9 重新定位至輪廓（REPOSA，REPOSL，REPOSQ，REPOSQA，REPOSH，REPOSHA，DISR，DISPR，RMIBL，RMBBL，RMEBL，RMNBL）

如果在加工操作期間中斷程式執行，並讓刀具回退，例如，由於刀具已破損，或您要測量工件，可以在輪廓上選擇任何一點在程式控制下重新定位。

指令 REPOS 在 ASUB 之中的作用為子程式返回（例如 M17）。下列單節不會執行。有關中斷程式執行的資訊，亦請參閱 「中斷常式（ASUB） (頁 138)」。



語法

```
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=...
```

含義

選擇逼近路徑

REPOSA：	幾何軸沿著直線重新定位輪廓。 亦重新定位所有其他通道軸。
REPOSL：	幾何軸沿著直線重新定位輪廓。 其他軸必須明確地進行程式設計。
REPOSQ DISR=…：	幾何軸沿著半徑 DISR 的四分之一圓重新定位輪廓。 其他軸必須明確地進行程式設計。
REPOSQA DISR=… ：	幾何軸沿著半徑 DISR 的四分之一圓重新定位輪廓。 亦重新定位所有其他通道軸。
REPOSH DISR=… ：	幾何軸沿著直徑 DISR 的半圓重新定位輪廓。 其他軸必須明確地進行程式設計。
REPOSHA DISR=… ：	幾何軸沿著半徑 DISR 的半圓重新定位輪廓。 亦重新定位所有其他通道軸。

選擇重新定位點

RMIBL：	逼近中斷點
RMIBL DISPR=…：	在中斷點之前，毫米 / 英吋 DISPR 距離的進入點
RMBBL：	逼近單節起點
RMEBL：	逼近單節結尾
RMEBL DISPR=… ：	在終點之前距離 DISPR 逼近單節結束點
RMNBL：	在最接近的路徑點逼近
A0 B0 C0 ：	要進行逼近的軸

說明

相容性

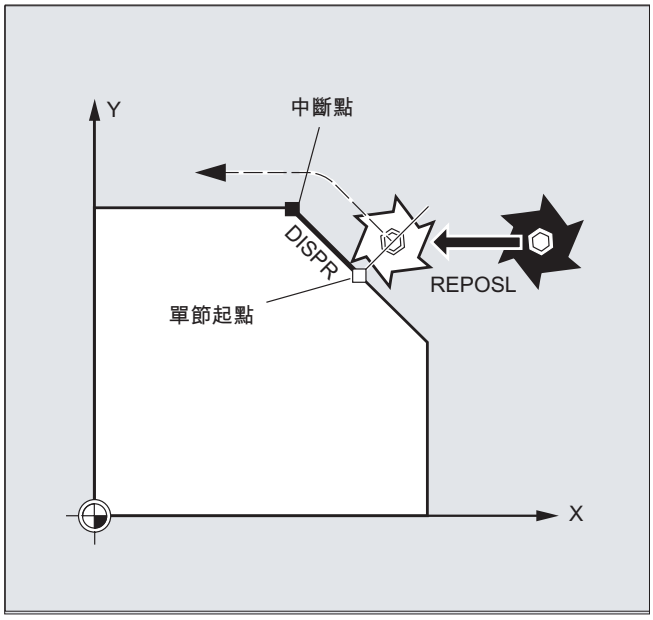
為保持較早軟體版本的相容性，您仍然可以透過模態指令 RMI、RMB、RME 及 RMN 程式設計 REPOS 逼近模式。在 ASUB 內部使用時，這種作法應在 PROC 陳述式之中分配 SAVE 屬性，否則在 ASUB 使用的模態 REPOS 逼近模式，也會在後續的 REPOS 程序中生效（如果偏離預先設定的 RMI）。

沿直線重新定位輪廓 REPOSA，REPOSL

刀具沿直線逼近重新定位點。

範例

```
REPOSL RMIBL DISPR=6 F400
```

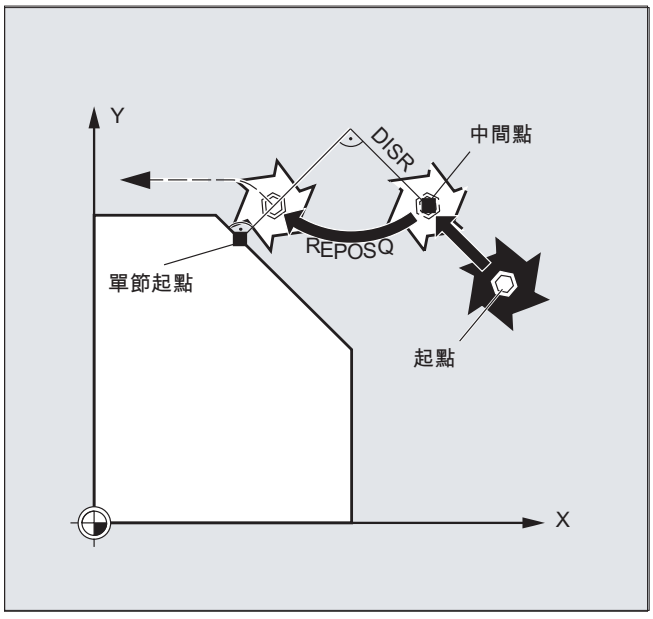


沿四分之一圓重新定位輪廓 REPOSQ，REPOSQA

刀具以 DISR=...的半徑，沿象限弧逼近重新定位點。控制系統會自動計算起點與重新定位點之間的必要中間點。

範例

```
REPOSQ RMIBL DISR=10 F400
```

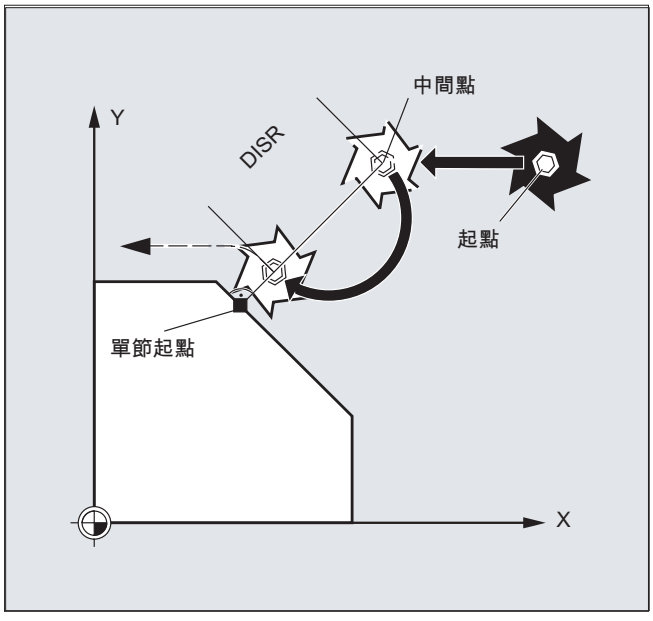


沿半圓重新定位輪廓 REPOSH，REPOSHA

刀具以 DISR=...的直徑，沿半圓弧逼近重新定位點。控制系統會自動計算起點與重新定位點之間的必要中間點。

範例

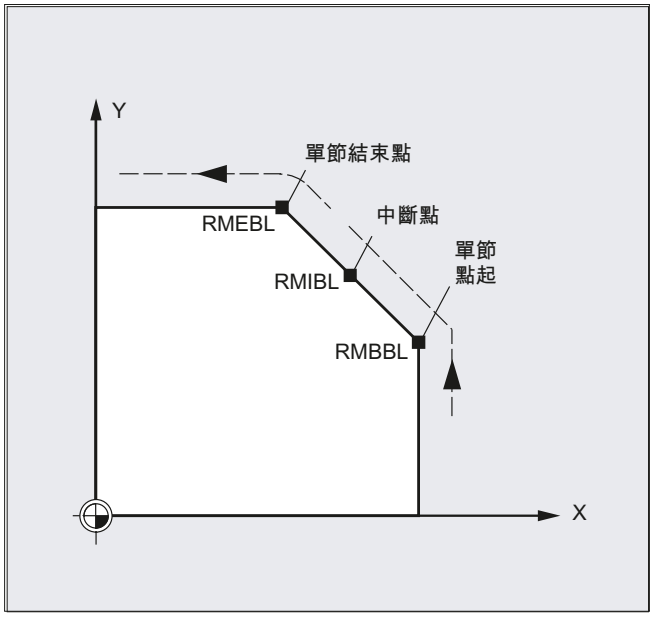
```
REPOSH RMIBL DISR=20 F400
```



指定重新定位點（不適用於 SERUPRO 使用 RMNBL 逼近）

在參考已中斷程式執行的 NC 單節下，可以選擇三個不同重新定位點的其中一個：

- RMIBL，中斷點
- RMBBL，單節起點或上一個終點
- RMEBL，單節結束點



RMIBL DISPR=...或 RME DISPR=... 允許您選擇位於中斷點或單節結束點之前的重新定位點。

DISPR=...允許您以毫米 / 英吋定義重新定位點與終點前中斷之間的輪廓距離。 即使是高值，這個點也不能超過單節起點。

如果沒有程式設計 DISPR=...指令，則套用 DISPR=0，加上中斷點（使用 RMIBL）或單節結束點（使用 RMEBL）。

DISPR 符號

DISPR 符號會進行評估。 如果是加號，行為如前。

如果是減號，逼近是在中斷點之後，或使用 RMBBL 時，在單節起點之後。

中斷點與逼近點之間的距離取決於 DISPR 的值。 即使值較高，這個點最遠只能位於單節結束點。

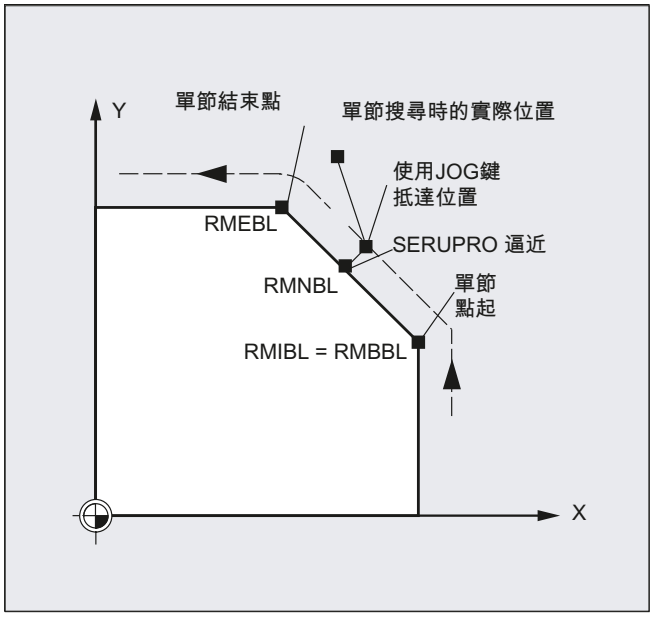
應用範例：

感應器會識別逼近夾具。 會啟動 ASUB，以略過夾具。

之後，負值 DISPR 在夾具之後一點上重新定位，然後程式繼續執行。

SERUPRO 使用 RMNBL 逼近

如果在任何位置進行加工期間強迫取消，則使用 SERUPRO 逼近和 RMNBL，逼近離取消點最短的路徑，以便之後只處理剩餘距離。 使用者在中斷單節啟動 SERUPRO 程序，然後使用寸動進給鍵，以移至目標單節的問題元件之前。



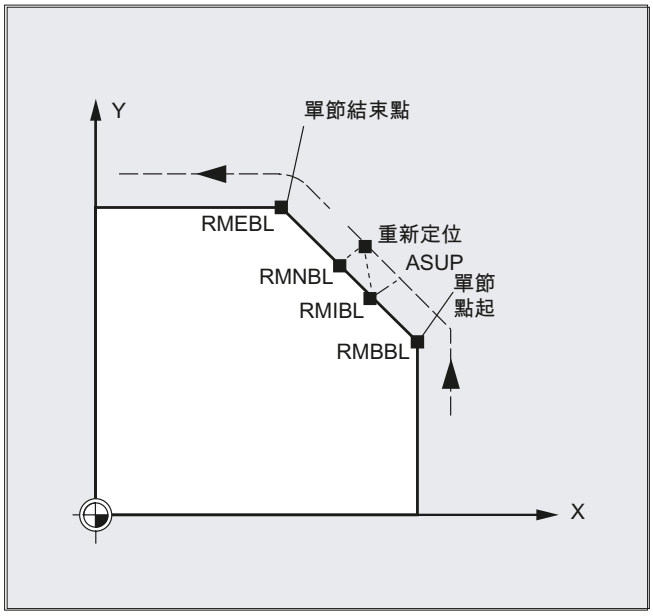
說明

SERUPRO

對 SERUPRO 來說，RMIBL 和 RMBBL 是完全相同的。 RMNBL 並非僅限用於 SERUPRO，亦適用於一般情況。

從最接近的路徑點逼近 RMNBL

解譯 REPOSA 時，使用 RMNBL 的重新定位單節在中斷後並不是完整啟動，而是只處理剩餘距離。 逼近的是中斷單節的最接近路徑點。



有效 REPOS 模式的狀態

中斷單節的有效 REPOS 模式可以使用同步動作和變數$AC_ REPOS_PATH_MODE 進行讀取：

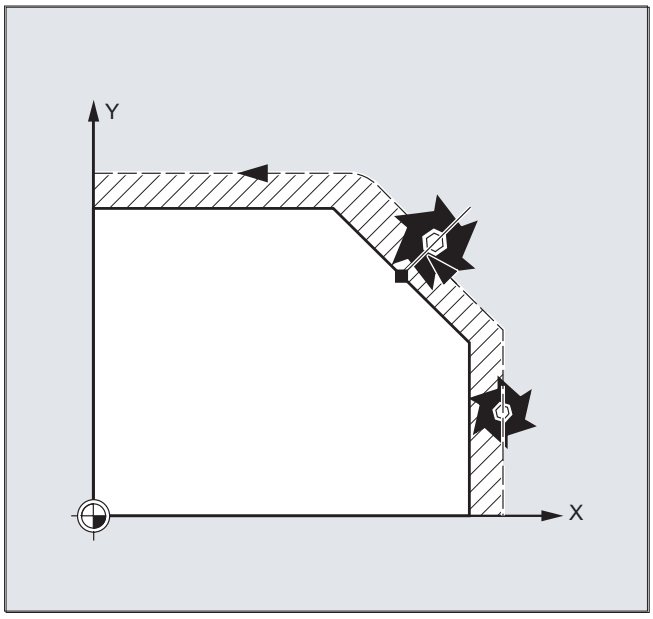
0 未定義逼近
1 RMBBL： 逼近開始處
2 RMIBL： 逼近中斷點
3 RMEBL： 逼近單節結尾
4 RMNBL： 逼近中斷單節的下一個路徑點

使用新增刀具逼近

如果由於刀具破損而停止程式執行，則適用下列情形：

已程式設計新增 D 數量時，加工程式是以修改的刀具偏移值，在重新定位點繼續執行。

已修改刀具偏移量值之處，也許無法重新逼近中斷點。 在此情況下，就逼近新輪廓上最接近中斷點的點（可能是由 DISPR 修改）。



逼近輪廓

刀具在輪廓上重新定位的動作可以程式設計。 為要移動的軸位址輸入零點。

REPOSA、REPOSQA 和 REPOSHA 指令會自動將所有軸重新定位。 各個軸名稱並不需要重新指定。

當 REPOSL、REPOSQ 和 REPOSH 指令設定完成後，所有幾何軸都會自動移動，也就是不需要在指令中指定。 其他所有軸都必須在指令中指定。

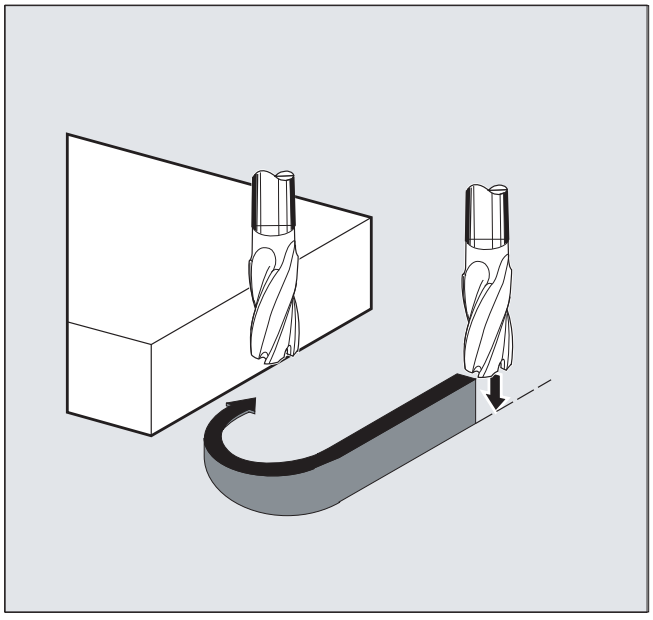
下列情形適用於 REPOSH 和 REPOSQ 圓弧動作：

圓弧是在指定的工作平面 G17 到 G19 中移動。

如果在逼近單節中指定第三個幾何軸（進給方向），如果刀具位置與進給方向的程式設計位置並未重疊，就會沿螺旋線逼近重新定位點。

在下列情況中，具有自動切換可直線逼近 REPOSL：

- 您並未指定 DISR 的值。
- 沒有定義的逼近方向可使用（在單節中程式設計中斷而沒有移動資訊）。
- 逼近方向與目前的工作平面垂直相交。



12.10 影響動作控制

12.10.1 比例震動校正（JERKLIM）

若使用 NC 指令「JERKLIM」，即可在關鍵程式區段降低或提高路徑動作的最大軸震動（以機械資料作設定）。

需求條件

加速度模式 SOFT 必須有效。

效果

此功能在以下情況中有效：

- 於自動 (AUTOMATIC) 操作模式。
- 僅在路徑軸上。

語法

```
JERKLIM[<軸>]=<值>
```

含義

JERKLIM：	震動校正的指令	
<軸>：	要手動倍率震動臨界值的機械軸。	
<值>：	修正值百分比；參考路徑動作的最大軸震動設定值 (MD32431 $MA_MAX_AX_JERK)。	
	值域：	1 ... 200
	值 100 不會影響震動。	

說明

JERKLIM 在工件程式結束以及通道重置時的行為，係利用機械資料 MD32320 $MA_DYN_LIMIT_RESET_MASK 的位元 0 設定：

- Bit 0 = 0：
  JERKLIM 的程式設定值，由通道重置 / M30 重置為 100%。
- Bit 0 = 1：
  JERKLIM 的程式設定值，在通道重置 / M30 之後仍維持不變。

範例

程式碼	註解
...	
N60 JERKLIM[X]=75	；軸在 X 方向的滑動，至多僅可以該軸震動允許值的 75% 加速／減速。
...	

12.10.2 比例速率修正（VELOLIM）

可在工件程式或同步動作中，利用 VELOLIM 指令，降低透過機械資料設定的最大可能軸速率或齒輪檔位相關之最大可能主軸轉速。

效果

此功能在以下情況中有效：

- 於自動 (AUTOMATIC) 操作模式。
- 於路徑及定位軸上。
- 於主軸／軸操作的主軸上

語法

```
VELOLIM[<軸/主軸>]=<值>
```

含義

VELOLIM：	速率校正的指令
<軸 / 主軸>：	速率或轉速臨界值應調整的軸或主軸。 **主軸的 VELOLIM 設定** 使用機械資料 （MD30455 $MA_MISC_FUNCTION_MASK，位元 6）在工件程式中進行程式設計時，可將其設為不論是用作主軸或軸（位元 6 = 1），「VELOLIM」皆有效。也可設為針對每一種操作模式，分別進行程式設計（位元 6 = 0）。 如果要分別使其有效，則進行程式設計時，須選擇使用識別碼： • 利用主軸識別碼 S<n> ，選擇主軸操作模式 • 利用軸識別碼，例如 「C」，選擇軸操作模式

<值>：	修正值百分比 修正值參考： • 軸操作的軸／主軸（MD30455 位元 6 == 0）： 最大軸速率設定 (MD32000 $MA_MAX_AX_VELO)。 • 主軸或軸操作的主軸（MD30455 位元 6 == 1）： 生效齒輪單元階段的最大速度 (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])	
	值域：	1 ... 100
	值 100 不影響速率或速度。	

說明

在工件程式結束或通道重置時的行為

「VELOLIM」在工件程式結束以及通道重置時的行為，可利用機械資料進行設定： MD32320 $MA_DYN_LIMIT_RESET_MASK，位元 0

偵測主軸操作的生效速度限制

主軸操作時，可利用以下系統變數，偵測透過「VELOLIM」（低於 100 %）設定的速度限制：

- $AC_SMAXVELO（最大可能主軸速度）
- $AC_SMAXVELO_INFO（由於速度限制所需的識別碼）

範例

範例 1： 機械軸速率限制

程式碼	註解
...	
N70 VELOLIM[X]=80	；軸在 X 方向的滑動，至多可以該軸速率允許值的 80% 移動。
...	

範例 2： 主軸轉速限制

程式碼	註解
N05 VELOLIM[S1]=90	；將主軸 1 的最大速度限制在 1000 rpm 的 90% 以下。
...	
N50 VELOLIM[C]=45	；將速度限制在 1000 rpm 的 45% 以下，C 為 S1 的軸識別碼。
...	

主軸 1 的機械資料設定 (AX5)

- 齒輪檔位 1 的最大轉速 = 1000 rpm:
  MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
- 設定「VELOLIM」使主軸和軸操作一起動作，不論編寫的識別碼為何：
  MD30455 $MA_MISC_FUNCTION_MASK[AX5]，位元 6 = 1

12.10.3 JERKLIM 和 VELOLIM 的程式範例

下列程式示範依比例限制震動和速率的應用程式範例：

程式碼	註解
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20	
ACC[Y]=30	
N1200 G1 Y20 VELOLIM[X]=5	；以 X 方向滑動的軸應該只有在軸所允許的最多 5% 速率時才移動。
JERKLIM[Y]=200	；以 Y 方向滑動的軸可以在軸所允許的最大 200% 震動值下進行加速 / 減速。
N1300 G1 X0 JERKLIM[X]=2	；以 X 方向滑動的軸應該只有在軸所允許的最大 2%震動值下才進行加速 / 減速。
N1400 G1 Y0	
M30	

12.11 程式設計輪廓 / 方向允差 (CTOL, OTOL, ATOL)

可以使用位址 CTOL、OTOL 及 ATOL 調整加工允差，使用機械及設定資料參數化，套用在工件程式的壓縮程序函式、平滑化及方向平滑化上。

程式設計的允差值一直有效，直到重新程式設計或是透過指派一個負值將其刪除為止。另外，要在程式結尾將其刪除或重置。參數化允差值在刪除後會再度恢復有效。

語法

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

含義

CTOL：	程式設計**輪廓允差**的位址			
	應用：	● 所有的壓縮函數 ● 除了 G641 與 G644 以外的所有倒圓角類型		
	前置處理停止：	否		
	有效：	模態		
	<Value>：	輪廓允差的值指定為長度。		
		類型：	REAL	
		單位：	inch/mm（視目前的尺寸設定而定）	
		值域：	≥ 0：	允差值
			< 0：	刪除程式設計的允差值 ⇒ 在機械或設定資料中參數化的允差值再度恢復有效。

OTOL：	程式設計**方向允差**的位址			
	應用：	• 所有的壓縮函數 • ORISON 方向平滑化 • 所有的平滑化類型，除 G641、G644 及 OSD 以外		
	前置處理停止：	否		
	有效：	模態		
	<Value>：	方向允差的值指定為角度。		
		類型：	REAL	
		單位：	度	
		值域：	≥ 0：	允差值
			< 0：	刪除程式設計的允差值 ⇒ 在機械或設定資料中參數化的允差值再度恢復有效。
ATOL：	程式設計**軸專屬允差**的位址			
	應用：	• 所有的壓縮函數 • ORISON 方向平滑化 • 所有的平滑化類型，除 G641、G644 及 OSD 以外		
	前置處理停止：	否		
	有效：	模態		
	<Axis>：	套用澄社設計允差的通道軸名稱		
	<Value>：	軸允差的值會視軸的類型（線性或旋轉軸）指定為長度或角度。		
		類型：	REAL	
		單位：	用於線性軸：	inch/mm（視目前的尺寸設定而定）
			用於旋轉軸：	度
		值域：	≥ 0：	允差值
			< 0：	刪除程式設計的允差值 ⇒ 在機械或設定資料中參數化的允差值再度恢復有效。

說明

使用 CTOL 及 OTOL 程式設計的通道專屬允差值比使用 ATOL 程式設計的軸專屬允差值有更高的優先權。

說明

比例調整框架

比例調整框架會以和軸位置相同的方式影響程式設計的允差，換言之，相對允差會維持不變。

範例

程式碼	註解
COMPCAD G645 G1 F10000	；啟用 COMPCAD 壓縮函數。
X...Y...Z...	；機械資料與設定資料套用於此。
X...Y...Z...	
X...Y...Z...	
CTOL=0.02	；從此處開始套用 0.02 毫米的輪廓允差。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	
ASCALE X0.25 Y0.25 Z0.25	；從此處開始套用 0.005 毫米的輪廓允差。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	
CTOL=-1	；機械資料與設定資料再次從此處開始套用。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	

系統變數

以前置處理停止讀取

使用下列系統變數，可以在工件程式中讀取目前作用中的遠差，並同步化動作：

- $AC_CTOL
  當處理實際的主要執行單節時，通道專屬輪廓允差啟動。
  若無有效的輪廓允差，$AC_CTOL 會傳回從幾何軸允差平方和得到的平方根值。
- $AC_OTOL
  當處理實際的主要執行單節時，通道專屬方向允差啟動。
  若無有效的方向允差，$AC_OTOL 會傳回從在啟動方向轉換時方向軸允差平方和得到的平方根值。否則，其會傳回值「-1」。
- $AA_ATOL[<軸>]
  當處理實際的主要執行單節時，軸專屬輪廓允差啟動。
  若無有效的輪廓允差，$AA_ATOL[<幾何軸>] 會傳回除以幾何軸數根值的輪廓允差。
  若啟用了方向允差及方向轉換，$AA_ATOL[<方向軸>] 會傳回除以方向軸數根值的方向允差。

說明

若現在已程式設計允差值，$A 變數不會與個別函數中的允差有足夠的不同。

此種情況會在機械資料與設定資料為壓縮函數、平滑化及方向平滑化設定不同允差時出現。接著系統變數以目前同時發生的作用中函式，傳回最大值。例如，若壓縮函數啟用時的方向允差為 0.1° 且 ORISON 方向平滑化為 1°，則 $AC_OTOL 變數會傳回值「1」。若停用了方向平滑化，$AC_OTOL 會傳回數值「0.1」。

讀取時沒有前置處理停止

使用下列系統變數，可以在工件程式中讀取目前作用中的遠差：

- $P_CTOL
  目前作用中通道專屬輪廓允差。
- $P_OTOL
  目前作用中通道專屬方向允差。
- $PA_ATOL
  目前作用中軸專屬輪廓允差。

補充條件

使用 CTOL、OTOL 及 ATOL 程式設計的允差也會影響間接根據這些允差的函式：

- 限制在設定點數值計算中的弦誤差
- 自由格式表面模式的基本函式

下列的平滑化函式**不受** CTOL、OTOL 及 ATOL 程式設計的影響：

- 以 OSD 平滑化方向
  OSD 不使用允差，而是使用單節變化的距離。
- 以 G644 平滑化
  G644 不是用來平滑化，它是用來最佳化刀具變更及其他與加工無關的動作。
- 以 G645 平滑化
  G645 的行為幾乎與 G642 完全一致，因此是使用程式設計的允差。機械資料 MD33120 $MA_PATH_TRANS_POS_TOL 的允差值僅使用在曲率上跳躍的均勻切線單節變化上，例如一個切線圓/直線變化。在這些點上的倒圓角路徑也可能會落在程式設計輪廓的外側，這其中有許多的應用較少允差。另外，補正這類型曲率中的變更通常只需要極小的固定允差，NC 程式設計師不需要特別關注。

12.12 啟用耦合功能的單節變更行為 (CPBC)

CPBC 指令可指定必須滿足的單節變更條件，以便透過啟用的耦合功能在工件程式中執行單節變更。

語法

```
CPBC[<跟隨軸>] = <條件>
```

含義

CPBC：	啟用耦合功能的單節變更條件	
<跟隨軸>：	跟隨軸的軸識別碼	
<條件>：	單節變更條件	
	類型：	STRING
	值	**含義：** 進行單節變更
	"NOC"	不考慮耦合狀態
	"IPOSTOP"	用於設定點同步
	"COARSE"	用於實際值同步「粗略」
	"FINE"	用於實際值同步「精細」

範例

程式碼

```
；單節發生變化：
；- 耦合至跟隨軸 X2 == 啟用
；- 設定點同步 == 啟用
CPBC[X2]="IPOSTOP"
```

軸耦合 13

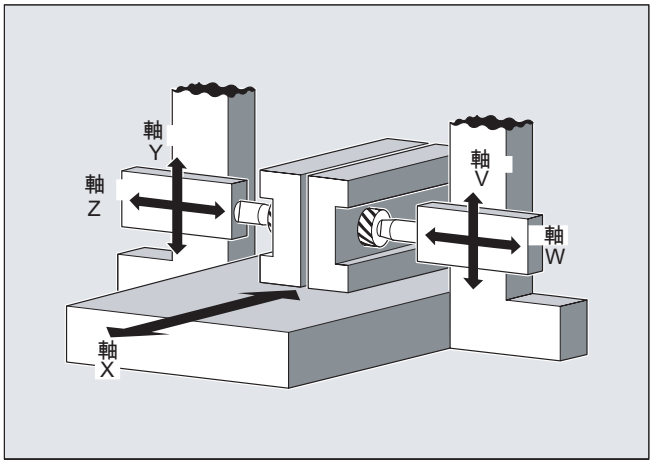
13.1 耦合動作（TRAILON，TRAILOF）

已定義的先導軸移動時，指派給它的耦合動作軸（=跟隨軸）會移動通過由先導軸說明的距離，以允許耦合係數。

主導軸與及隨動軸同在，便構成耦合軸。

應用

- 藉由模擬軸移跨一個軸。 主導軸是模擬軸，而耦合軸則是實軸。 以此方式，在耦合係數功能下，實軸可被移跨。
- 使用 2 個耦合動作群組進行雙側加工：
  1. 先導軸 Y，耦合動作軸 V
  2. 先導軸 Z，耦合動作軸 W



語法

```
TRAILON(<跟隨軸>,<主導軸>,<耦合係數>)
TRAILOF(<跟隨軸>,<主導軸>,<主導軸 2>)
TRAILOF(<跟隨軸>)
```

含義

TRAILON：	用於啟用及定義已耦合的軸群組之指令	
	有效：	模態

<跟隨軸>：	參數 1： 尾隨軸的軸名稱 **附註：** 已耦合的動作軸亦可作為其他已耦合之動作軸的先導軸。 以此方法，可以創造一系列不同的耦合軸組合。	
<先導軸>：	參數 2： 尾隨軸的軸名稱	
<耦合係數>：	參數 3： 耦合係數 耦合係數指定所想要的耦合動作軸之路徑與先導軸之間關係： **<**耦合係數**> =** 耦合動作軸的路徑 / 先導軸的路徑	
	類型：	REAL
	預設值：	1
	輸入負值會導致主動及耦合軸以相反方向移動。 如果耦合係數未程式設計，則會自動套用耦合係數 1。	
TRAILOF：	用於停用已耦合之軸群組的指令	
	有效：	模態

	具有 2 個參數的 TRAILOF 僅會啟用指定先導軸的耦合：
	`TRAILOF(<跟隨軸>,<主導軸>)`
	若已耦合的動作軸有 2 個先導軸，則可使用 3 個參數呼叫 TRAILOF 以便將這兩個耦合均停用。
	`TRAILOF(<跟隨軸>,<主導軸>,<主導軸 2>)`
	程式設計 TRAILOF 時不指定先導軸會產生相同的結果：
	`TRAILOF(<跟隨軸>)`	

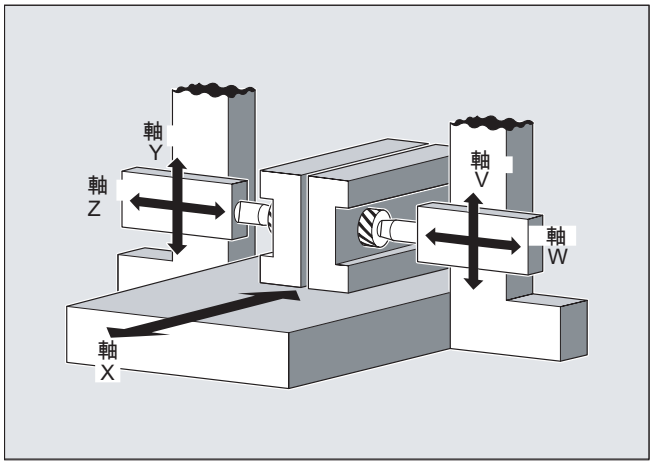
說明

耦合軸動作永遠都是在基準座標系統（BCS）中執行。

能同時被啟動的耦合軸組數量，僅被機台上軸組合的最大可能數量所限制。

範例

工件要使用圖示中顯示的軸設定，在雙側進行加工。 若要執行這項作業，要建立兩個耦合軸組合。



程式碼	註解
…	
N100 TRAILON（V，Y）	；啟用第一個耦合軸群組。
N110 TRAILON（W，Z，-1）	；啟用第 2 個已耦合的軸群組，負耦合係數： ；耦合動作軸移動；與先導軸的方向相反。
N120 G0 Z10	；朝與軸方向相反的方向進給 Z 軸與 W 軸。
N130 G0 Y20	；Y 和 V 軸依相同的軸方向進給。
…	
N200 G1 Y22 V25 F200	；將已耦合之動作軸 V 的相依與獨立移動重疊。
…	
TRAILOF（V，Y）	；停用第一個耦合軸群組。
TRAILOF（W，Z）	；；停用第二個耦合軸群組。

其他資訊

軸類型

耦合軸群組可以包含任何想要的線性軸和旋轉軸組合。 模擬軸也可以定義為先導軸。

耦合動作軸

最多兩個先導軸可以同時指派給尾隨軸。 指派是以不同的耦合軸組合進行。

已耦合之動作軸可使用完整範圍的可用動作指令（G0、G1、G2、G3、等）進行程式設計。耦合軸會根據耦合係數，不但移動獨立定義的路徑，也移動自其先導軸衍生的路徑。

動態臨界值

動態臨界值視已耦合之軸群組的啟用類型而定：

- 在工件程式中啟用
  若在工件程式中執行啟用且所有的先導軸均在啟用的通道中啟用作為程式設計軸，則在移動先導軸時請將所有已耦合的動作軸之動態回應列入考慮以避免已耦合的動作軸過載。
  若在工件程式中未將啟用通道中的先導軸啟用為程式設計軸（$AA_TYP ≠ 1）而執行啟用，則在先導軸移動時不需將已耦合的動作軸之動態回應列入考慮。 如此會造成已耦合之動作軸具有少於比耦合所需要的動態回應而過載。
- 已同步動作啟用
  若在同步動作中執行啟用，已耦合之動作軸的動態回應在先導軸移動時不需列入考慮。 如此會造成已耦合之動作軸具有少於比耦合所需要的動態回應而過載。

⚠ 小心
軸過載 若啟用了已耦合的軸群組： ● 在同步動作中 ● 在工件程式中，先導軸在耦合動作軸之通道中並非程式軸 採取適當的動作，確保移動先導軸時不會使耦合動作軸過載，為使用者／機台製造商的責任。

耦合狀態

工件程式中的軸耦合狀態可使用以下系統變數確認：

$AA_COUP_ACT[<軸>]

值	含義
0	沒有啟用耦合
8	已啟用耦合動作

顯示模數旋轉軸的耦合動作軸之剩餘距離

若先導軸和耦合動作軸為模數旋轉軸，先導軸的移動動作為 n * 360（其中 n = 1, 2, 3 ...），在耦合動作軸的剩餘距離顯示中累加，直到耦合關閉。

範例： 使用 TRAILON 指令的程式區段以及先導軸 B 和跟隨軸 C

程式碼	註解
TRAILON(C,B,1)	；啟動耦合
G0 B0	；開始位置
	；在單節起始處顯示剩餘距離：

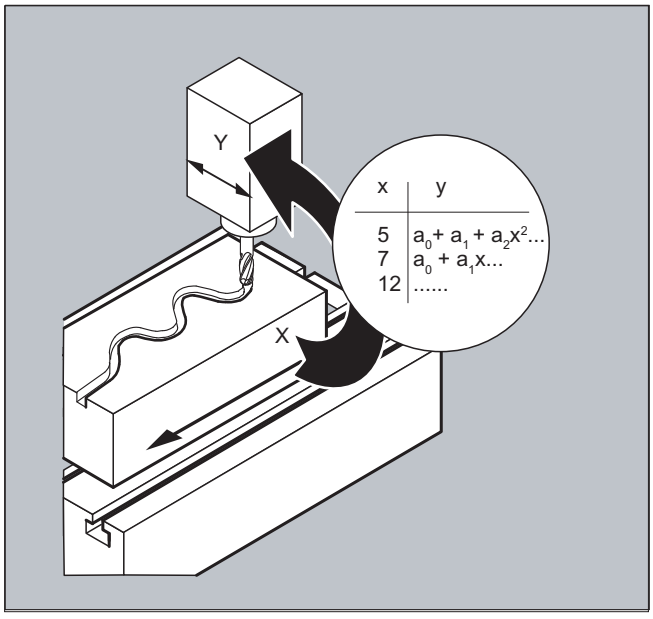
程式碼	註解
G91 B360	；B=360, C=360
G91 B720	；B=720, C=1080
G91 B360	；B=360, C=1440

13.2 曲線表（CTAB）

曲線表可用於程式設計兩軸（先導軸與跟隨軸）間的位置與速率關係。 曲線表是在工件程式中定義的。

應用

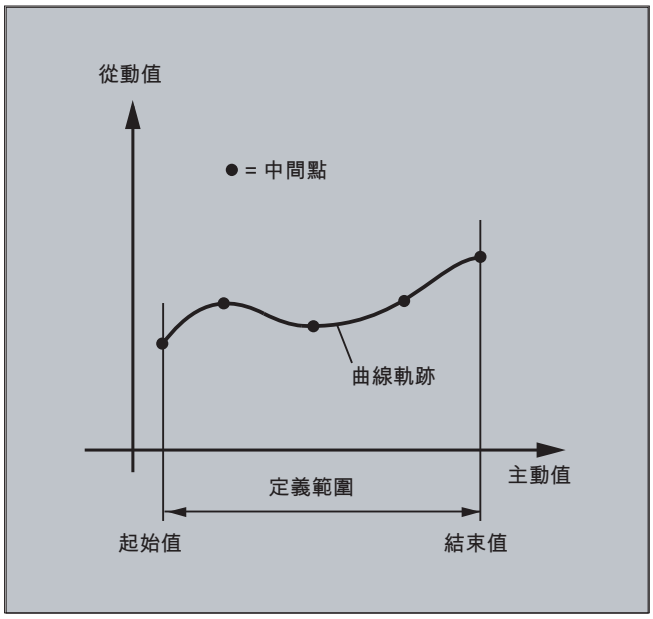
以曲線表取代機械檔塊。 曲線表透過建立先導值與跟隨值之間函數關係，形成軸主動值耦合的基礎： 在適當的程式設計下，控制會從先導軸和跟隨軸的相對位置，計算對應於擋塊的多項式。



13.2.1 定義曲線表（CTABDEF、CATBEND）

曲線表代表工件程式或工件程式的區段，是用 CTABDEF 在開始處，並用 CTABEND 在結尾閉鎖。

在此工件程式區段中，使用動作操作將唯一的跟隨軸位置指派至先導軸的個別位置中；這些跟隨軸位置可在以最多五次多項式的行式計算曲線定義時作為中間點。



需求條件

該 MD 必須做對應的設定以確保有保留足夠的記憶體空間給曲線表的定義使用（機台製造商）。

語法

```
CTABDEF（<跟隨軸>，<先導軸>，<n>，<週期性>[，<記憶體位置>]）
...
CTABEND
```

含義

CTABDEF( )：	曲線表定義的起點
CTABEND：	曲線表定義的結尾
<跟隨軸>：	使用曲線表計算動作的軸
<先導軸>：	提供計算跟隨軸動作所需之主動值的軸
<n>：	曲線表的編號 (ID) 曲線表的編號是唯一且與記憶體位置無關的。 因此，靜態與動態NC記憶體中不可能同時存在相同編號的曲線表。

<週期性>：	表週期性	
	0	表格非週期性（該表僅處理一次，旋轉軸亦同）
	1	與先導軸相關時，表是週期性
	2	與先導軸和跟隨軸相關時，表是週期性
<記憶體位置>：	指定記憶體位置（選用）	
	「SRAM」	該曲線表建立在**靜態** NC 記憶體中。
	「DRAM」	該曲線表建立在**動態** NC 記憶體中。
	附註： 若未替此參數程式設計值，會使用以 MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE 設定的預設記憶體位置。	

說明

覆寫

一旦其編號（<n>）在其他曲線表定義中使用後，便會將曲線表覆寫。（例外： 曲線表在軸耦合中啟用或以 CTABLOCK 加以鎖定）。 **覆寫曲線表時，不會輸出任何警告。**

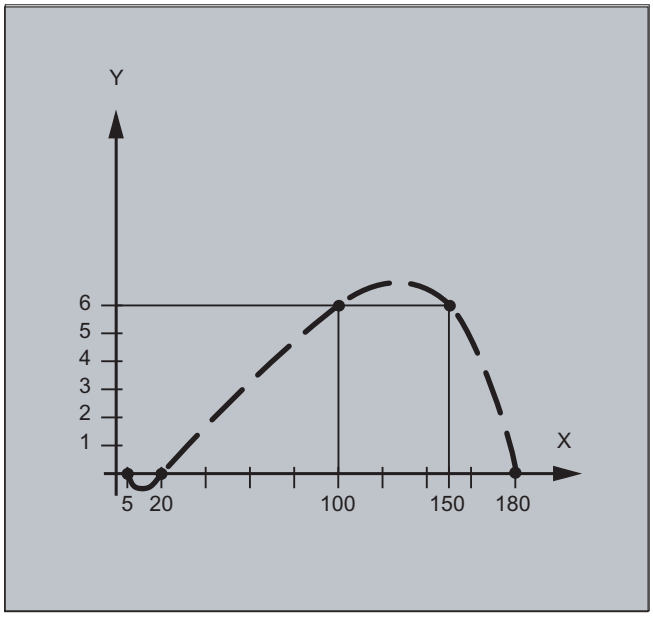
範例

範例 1： 曲線表定義的程式區段

必須使用未變更的程式區段，以供定義曲線表。 處理停止 STOPRE 的指令可以保留，只要程式區段未用於表定義，而且已移除 CTABDEF 和 CTABEND，就會立即再度啟用。

程式碼	註解
…	
CTABDEF(Y,X,1,1)	；曲線表的定義。
…	
IF NOT （$P_CTABDEF）	
STOPRE	
ENDIF	
…	
CTABEND	

範例 2： 非週期性曲線表的定義



程式碼	註解
N100 CTABDEF(Y,X,3,0)	；定義的開始；非週期性曲線表，編號 3。
N110 X0 Y0	；第 1 動作操作，定義起始值和第一個中間點：
	主動值： 0，跟隨值： 0
N120 X20 Y0	；第 2 插補點：
	主動值： 0...20，跟隨值： 起始值...0
N130 X100 Y6	；第 3 插補點：
	主動值： 20...100，跟隨值： 0…6
N140 X150 Y6	；第 4 插補點：
	主動值： 100...150，跟隨值： 6…6
N150 X180 Y0	；第 5 插補點：
	主動值： 150...180，跟隨值： 6…0
N200 CTABEND	；定義結尾。 曲線表以其內在表象最高 5 次多項式的方式產生。 使用指定中間點進行的曲線定義計算視模態選擇的插補類型（圓弧、線性、曲線插補）而定。 開始定義前的工件程式狀態會重新儲存。

範例 3：週期性曲線表的定義

週期性曲線表的定義，編號 2，主動值域 0 到 360，跟隨軸動作從 0 到 45，然後回到 0：

程式碼	註解
N10 DEF REAL DEPPOS	
N20 DEF REAL GRADIENT	
N30 CTABDEF(Y,X,2,1)	；定義起點。
N40 G1 X=0 Y=0	
N50 POLY	

程式碼	註解
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	；定義結尾。
；透過耦合 Y 至 X，測試曲線：	
N120 G1 F1000 X0	
N130 LEADON(Y,X,2)	
N140 X360	
N150 X0	
N160 LEADOF(Y,X)	
N170 DEPPOS=CTAB(75.0,2,GRADIENT)	；讀取主動值 75.0 的表函數。
N180 G0 X75 Y=DEPPOS	；為先導軸和跟隨軸定位。
；啟動耦合之後，不需要跟隨軸的同步處理。	
N190 LEADON（Y，X，2）	
N200 G1 X110 F1000	
N210 LEADOF（Y，X）	
N220 M30	

其他資訊

曲線表的起始值和結束值

曲線表的定義範圍開始的起始值是曲線表定義內第一個指定的相關聯軸位置（第一個移動敘述）。 曲線表定義範圍的結束值是根據最後的移動指令來決定。

可用的語言範圍

在曲線表的定義之內，您可以使用整個 NC 語言。

說明

在曲線表定義中不可使用以下項目：

- 前置處理停止
- 先導軸動作中的跳躍（例如在變更轉換時）
- 只有跟隨軸的移動敘述
- 先導軸的逆轉，也就是先導軸的位置必須永遠都是唯一的
- 在各種不同程式層級的 CTABDEF 和 CTABEND 敘述。

模態操作的效果

表定義完成後，在曲線表定義內做的所有模態敘述都無效。 因此，在其中完成表定義的工件程式在表定義之前和之後都處於相同狀態。

對 R 參數的配置

表格定義中對 R 參數的配置會在 CTABEND 之後重置。

範例：

程式碼	註解
...	
R10=5 R11=20	；R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	；R10=25
X=R10	
CTABEND	
...	；R10=5

啟動 ASPLINE，BSPLINE，CSPLINE

如果已在曲線定義表 CTABDEF ... CTABEND 之內啟動 ASPLINE、BSPLINE 或 CSPLINE，至少有一個起點應該在啟用此曲線之前進行程式設計。 必須避免 CTABDEF 之後立即啟用，否則曲線將取決於曲線表定義之前的目前軸位置。

範例：

程式碼
...
CTABDEF(Y,X,1,0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND

重覆使用曲線表

如果選擇的曲線表已儲存於靜態 NC 記憶體（SRAM），利用曲線表計算得出的先導軸與跟隨軸函數關係，在工件程式結束之後以及關機之後，將仍保留於該表編號之下。

建立於動態記憶體（DRAM）中的曲線表將於開機後刪除，可能必須重新產生。

只要建立了曲線表，便可套用至任一先導軸與尾隨軸軸組合，且該表與建立曲線表時所使用的軸無關。

覆寫曲線表

一旦在其他表定義中使用某曲線表編號，將立刻覆寫該曲線表。

例外情形： 曲線表在軸耦合中啟用或以 CTABLOCK 加以鎖定。

說明

覆寫曲線表時，不會輸出任何警告。

是否已啟用曲線表定義？

在工件程式中可隨時使用$P_CTABDEF 系統變數確認曲線表是否已啟用。

撤回曲線表定義

一旦排除了與曲線表定義有關的操作後，該工件程式區段便可再次作為實數工件程式使用。

使用"從外部來源執行"功能載入曲線表。

若從外部來源執行曲線表，則在 MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE 中，重新載入緩衝器（動態記憶體）大小的選擇需支援能同時儲存重新載入緩衝器中所定義的曲線表項目。 若否，則會取消處理並發出警報。

在跟隨軸中跳躍

依機械資料
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION
中的設定而定，如果先導軸中少了某個動作，可能會容許跟隨軸中的跳躍。

13.2.2 確認曲線表是否存在（CTABEXISTS）

CTABEXISTS 指令可用於確認指定的曲線表編號是否存在 NC 記憶體中。

語法

```
CTABEXISTS(<n>)
```

含義

CTABEXISTS：	確認靜態或動態 NC 記憶體中是否有存在曲線表編號<n>。	
	0	表格不存在
	1	表格存在
<n>：	曲線表的編號 (ID)	

13.2.3 刪除曲線表（CTABDEL）

CTABDEL 可用來刪除曲線表。

說明

在軸耦合中啟用的曲線表無法刪除。

語法

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL（<n>，<m>，<記憶體位置>）
CTABDEL()
CTABDEL（，，<記憶體位置>）
```

含義

CTABDEL：	用於刪除曲線表的指令	
<n>：	待刪除的曲線表編號（**ID**） 當刪除曲線表範圍 CTABDEL（<n>，<m>）時，<n>可用於指定該範圍中第一個曲線表的編號。	
<m>：	當刪除曲線表範圍 CTABDEL（<n>，<m>）時，<m>可用於指定該範圍中最後一個曲線表的編號。 <m>需大於<n>。	
<記憶體位置>：	指定記憶體位置（選用） 在**未**指定記憶體位置而進行刪除時，則會將靜態與動態 NC 記憶體中指定的曲線表刪除。 在**有**指定記憶體位置而進行刪除時，則僅會將位於指定記憶體位置上的曲線表刪除。 其餘會保留。	
	「SRAM」	在**靜態** NC 記憶體中執行刪除
	「DRAM」	在**動態** NC 記憶體中執行刪除

若程式設計了 CTABDEL 但未指定欲刪除的曲線表時，則會將**所有**的曲線表或在指定記憶體中的所有曲線表均刪除：

CTABDEL()：	將靜態與動態 NC 記憶體中的所有曲線表均刪除
CTABDEL(,,"SRAM")：	將靜態 NC 記憶體中所有的曲線表均刪除
CTABDEL(,,"DRAM")：	將動態 NC 記憶體中所有的曲線表均刪除

說明

使用 CTABDEL(<n>,<m>) 或 CTABDEL() 進行多重刪除時，若一個以上的待刪除曲線表啟用耦合狀態，則不會執行刪除指令；換言之，**不會**刪除任何待處理的曲線表。

13.2.4 鎖住曲線表以防止刪除與覆寫（CTABLOCK、CTABUNLOCK）

鎖定可設定用於保護曲線表免於被意外刪除或覆寫。 一旦設定鎖定後，可隨時將其撤回。

語法

Lock：

```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK（<n>，<m>，<記憶體位置>）
CTABLOCK()
CTABLOCK（，，<記憶體位置>）
```

取消鎖定：

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK（<n>，<m>，<記憶體位置>）
CTABUNLOCK()
CTABUNLOCK（，，<記憶體位置>）
```

含義

CTABLOCK：	用於**設定**防止刪除/覆寫之鎖定的指令
CTABUNLOCK：	用於**撤回**防止刪除/覆寫之鎖定的指令 CTABUNLOCK 可將使用 CTABLOCK 鎖定的曲線表取消鎖定。 涉入已啟用耦合的曲線表保持為鎖定，而無法被刪除。 使用 CTABLOCK 所做的鎖定會在因停用耦合而使因耦合所造成的鎖定取消後便取消鎖定。 因此，這個表可以刪除。 不必再次呼叫 CTABUNLOCK。
<n>：	待鎖定/取消鎖定的曲線表編號（**ID**） 將曲線表範圍 CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) 鎖定／解除鎖定時，<n> 可用於指定該範圍中第一個曲線表的編號。

<m>：	將曲線表範圍 CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) 鎖定／解除鎖定時，<m> 可用於指定該範圍中最後一個曲線表的編號。 <m>需大於<n>。	
<記憶體位置>：	指定記憶體位置（選用） 在**未**指定記憶體位置而進行鎖定/取消鎖定時，則會將靜態與動態NC記憶體中指定的曲線表鎖定/取消鎖定。 在**有**指定記憶體位置而進行鎖定/取消鎖定時，則僅會將位於指定記憶體位置上的曲線表鎖定/取消鎖定。 其餘的不鎖定/取消鎖定。	
	「SRAM」	在**靜態** NC 記憶體中進行鎖定/取消鎖定
	「DRAM」	在**動態** NC 記憶體中進行鎖定/取消鎖定

若程式設計了 CTABLOCK/CTABUNLOCK 但未指定欲鎖定/取消鎖定的曲線表時，則會將**所有**的曲線表或指定記憶體中的所有曲線表均鎖定/取消鎖定：

CTABLOCK()：	將靜態與動態 NC 記憶體中的所有曲線表均鎖定
CTABLOCK(,,"SRAM")：	將靜態 NC 記憶體中所有的曲線表均鎖定
CTABLOCK(,,"DRAM")：	將動態 NC 記憶體中所有的曲線表均鎖定
CTABUNLOCK()：	將靜態與動態 NC 記憶體中的所有曲線表均取消鎖定
CTABUNLOCK(,,"SRAM")：	將靜態 NC 記憶體中所有的曲線表均取消鎖定
CTABUNLOCK(,,"DRAM")：	將動態 NC 記憶體中所有的曲線表均取消鎖定

13.2.5 曲線表： 決定取線表特性（CTABID、CTABISLOCK、CTABMEMTYP、CTABPERIOD）

這些指令可用於選取曲線表的重要特性行（表編號、鎖定狀態、記憶體位置、週期性）。

語法

```
CTABID(<p>)
CTABID（<p>，<記憶體位置>）
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

含義

CTABID：	回傳在指定記憶體中輸入為第 <p> 個曲線表的**編號**。 範例： CTABID（1，"SRAM「）會傳回靜態 NC 記憶體中第一個曲線表的編號。 此時，第一個曲線表便是具有最高表編號的曲線表。 **附註：** 若記憶體中的曲線表順序在連續呼叫 CTABID 時改變，例如因使用 CTABDEL 刪除曲線表，CTABID(<p>,...) 會傳回具有相同編號 <p> 的不同曲線表。	
CTABISLOCK：	傳回曲線表編號 <n> 的**鎖定狀態**：	
	0	表未鎖定
	1	表被 CTABLOCK 鎖定
	2	表被啟動耦合鎖定
	3	表被 CTABLOCK 與啟用耦合鎖定
	-1	表格不存在
CTABMEMTYP：	傳回曲線表編號 <n> 的**記憶體位置**：	
	0	表在靜態 NC 記憶體中
	1	表在動態 NC 記憶體中
	-1	表格不存在
CTABPERIOD：	傳回曲線表編號 <n> 的**週期性**：	
	0	表不是週期性
	1	表在先導軸中週期運作
	2	表在先導與跟隨軸中週期運作
	-1	表格不存在
<p>：	記憶體中的輸入編號	
<n>：	曲線表的編號 **(ID)**	
<記憶體位置>：	指定記憶體位置（選用）	
	「SRAM」	**境態** NC 記憶體
	「DRAM」	**動態** NC 記憶體
	附註： 若未替此參數程式設計值，會使用以 MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE 設定的預設記憶體位置。	

13.2.6 讀取曲線表值（CTABTSV、CTABTEV、CTABTSP、CTABTEP、CTABSSV、CTABSEV、CTAB、CTABINV、CTABTMIN、CTABTMAX）

下列曲線表值可在工件程式中讀取：

- 在曲線表之起點與結尾的跟隨軸與先導軸值
- 在曲線區段之起點與結尾的跟隨軸值
- 先導軸值的跟隨軸值
- 跟隨軸值的先導軸值
- 跟隨軸最小與最大值
  - 在曲線表的整個定義範圍中
    或
  - 在已定義的曲線表之間

語法

```
CTABTSV(<n>,<斜度>[,<跟隨軸>])
CTABTEV(<n>,<斜度>[,<跟隨軸>])
CTABTSP(<n>,<斜度>[,<先導軸>])
CTABTEP(<n>,<斜度>[,<先導軸>])
CTABSSV(<主動值>,<n>,<斜度>[,<跟隨軸>])
CTABSEV(<主動值>,<n>,<斜度>[,<跟隨軸>])
CTAB(<主動值>,<n>,<斜度>[,<跟隨軸>,<先導軸>])
CTABINV(<跟隨值>,<近似值>,<n>,<斜度>[,<跟隨軸>,<先導軸>]
CTABTMIN(<n>[，<跟隨軸>])
CTABTMAX(<n>[，<跟隨軸>])
CTABTMIN(<n>,<a>,<b>[,<跟隨軸>,<先導軸>])
CTABTMAX(<n>,<a>,<b>[,<跟隨軸>,<先導軸>])
```

含義

CTABTSV：	在曲線表編號 <n> 的**起點**讀取跟隨軸值
CTABTEV：	在曲線表編號 <n> 的**結尾**讀取跟隨軸值
CTABTSP：	在曲線表編號 <n> 的**起點**讀取先導軸值
CTABTEP：	在曲線表編號 <n> 的**結尾**讀取先導軸值
CTABSSV：	在屬於指定先導軸值的曲線區段**起點**，讀取跟隨軸值（<主動值>）
CTABSEV：	在屬於指定先導軸值的曲線區段**結尾**，讀取跟隨軸值（<主動值>）
CTAB：	讀取指定先導軸值（<主動值>）的跟隨軸值

CTABINV：	讀取指定跟隨軸值（<跟隨值>）的先導軸值
CTABTMIN：	定義跟隨軸**最小值**： ● 在曲線表的整個定義範圍中 或 ● 在定義的區間 <a> ... <b>
CTABTMAX：	定義跟隨軸**最大值**： ● 在曲線表的整個定義範圍中 或 ● 在定義的區間 <a> ... <b>
<n>：	曲線表的編號 **(ID)**
<斜度>：	<斜度>參數傳回曲線表函數於計算位置的**傾斜度**。
<跟隨軸>：	使用曲線表計算動作的軸（選用）
<先導軸>：	提供計算跟隨軸動作所需之主動值的軸（選用）
<跟隨值>：	用於讀取 CTABINV 先導軸值的跟隨軸值
<先導值>：	先導軸值： ● 透過 CTAB 讀取相關的跟隨軸值 或 ● 使用 CTABSSV/CTABSEV 選擇曲線區段
<近似值>：	使用 CTABINV 派指至跟隨軸的先導軸值不可每次是唯一的。 因此，CTABINV 要需針對期望的先導軸值提供一個大略值作為參數。
<a>：	具有 CTABTMIN/CTABTMAX 主的動值區間下臨界值
<b>：	具有 CTABTMIN/CTABTMAX 主的動值區間上臨界值
	附註： 主動值區間 <a> 至 <b> 一定要在曲線表的定義範圍內。

範例

範例 1：

在曲線表的開頭與結尾處定義跟隨軸與先導軸值，其中跟隨軸的最大與最小值均在整個曲線表的定義範圍內。

程式碼	註解
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL STARTPARA	
N40 DEF REAL ENDPARA	

程式碼	註解
N50 DEF REAL MINVAL	
N60 DEF REAL MAXVAL	
N70 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	；表格定義的起點
N110 X0 Y10	；第一表區段起始位置
N120 X30 Y40	；第一表區段結束位置 = 第二表區段開始位置
N130 X60 Y5	；結束位置第 2 表區段 = ...
N140 X70 Y30	
N150 X80 Y20	
N160 CTABEND	；表格定義的結尾。
...	
N200 STARTPOS = CTABTSV(1,GRADIENT)	；曲線表開頭的跟隨軸值 = 10
N210 ENDPOS = CTABTEV(1,GRADIENT)	；曲線表結尾的跟隨軸值 = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)	；曲線表開頭的主軸值 = 0
N230 ENDPARA = CTABTEP(1,GRADIENT)	；曲線表結尾的主軸值 = 80
N240 MINVAL=CTABTMIN(1)	；當 Y=5 時跟隨軸的最小值
N250 MAXVAL=CTABTMAX(1)	；當 Y=40 時跟隨軸的最大值

範例 2：

搭配先導軸值 X=30 決定曲線區段開頭與結尾處的跟隨軸值。

程式碼	註解
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	；表格定義的起點。
N110 X0 Y0	；第一表區段起始位置
N120 X20 Y10	；第一表區段結束位置 = 第二表區段開始位置
N130 X40 Y40	；結束位置第 2 表區段 = ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	；表格定義的結尾。
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	；第二區段開始位置 Y = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	；第二區段結尾位置 Y = 40

其他資訊

於同步動作中使用

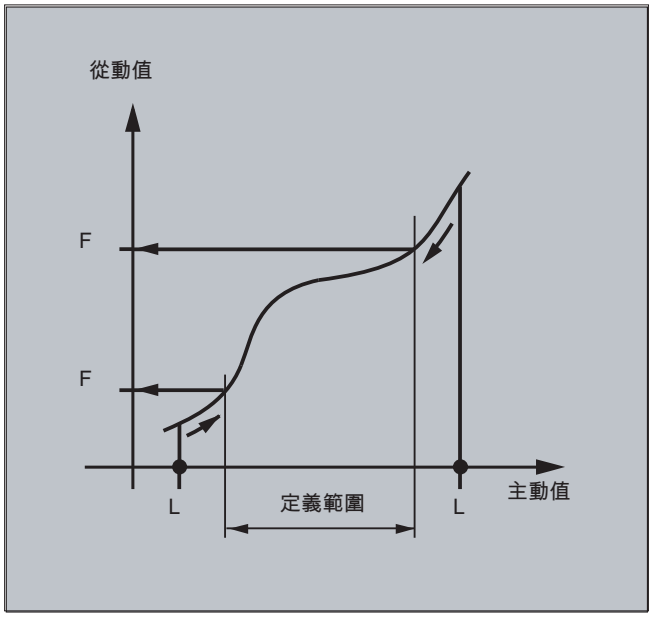
所有用來讀取曲線表值的指令，也可用於同步動作中（請參考"動作-同步動作"章節）。

使用 CTABINV、CTABTMIN 與 CTABTMAX 指令時，請確認：

- 在執行時有足夠的 NC 電源可用
  或
- 在曲線表中的區段號碼，會在呼叫之前被查詢，因此如有必要，可再對相關的表進行子分割。

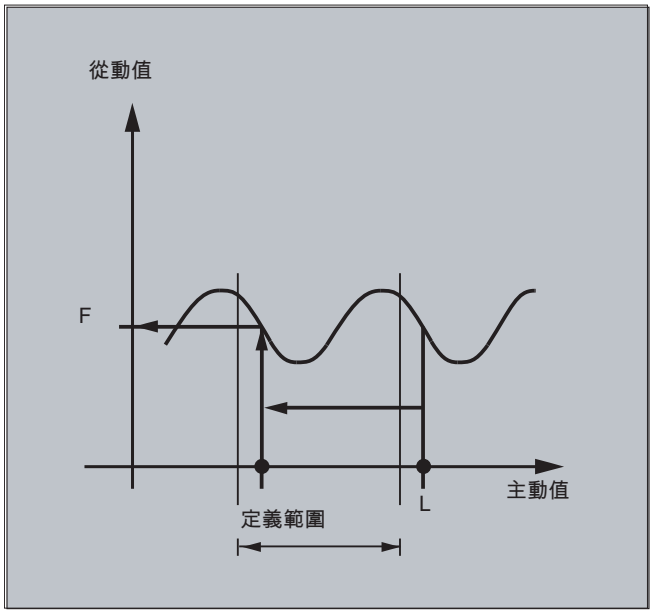
具有非週期性曲線表的 CTAB

若特定的<主動值>落在定義域之外，則上限或下限會被輸出為下列的值：



具有週期性曲線表的 CTAB

若特定的 <主動值> 超出定義範圍，主動值則將估算為定義範圍的模數，並輸入對應的跟隨值：



CTABINV 的大約值

CTABINV 指令，因此，需要一個大約的值來作為預測的主動值。 CTABINV 將傳回最接近概估值的先導值。 例如，大約的值可以是從前一個插補器時脈循環而來的主動值。

曲線表函數的斜度

斜度的輸出（<斜度>），使得計算先導軸或在對應位置的跟隨軸之速率這件事，變得可能。

先導軸或跟隨軸的規格

如果先導軸與跟隨軸的長度單位不同，則先導軸或跟隨軸選配規格就格外重要。

CTABSSV，CTABSEV

在下列情況中，指令 CTABSSV 與 CTABSEV 不適合用來查詢已程式設計的區段：

- 已程式設計圓形或漸開線。
- 含 CHF/ RND 的倒角或倒圓角為啟用狀態
- 含 G643 的平滑為啟用狀態
- 含 COMPON/COMPCURV/COMPCAD 的 NC 單節壓縮為啟用狀態

13.2.7 曲線表： 檢查資源的使用（CTABNO，CTABNOMEM，CTABFNO，CTABSEGID，CTABSEG，CTABFSEG，CTABMSEG，CTABPOLID，CTABPOL，CTABFPOL，CTABMPOL）

程式設計者可使用這些指令，來取得關於曲線表，表區段以及多項式的資源最新使用資訊。

語法

```
CTABNO
CTABNOMEM（<記憶體位置>）
CTABFNO（<記憶體位置>）
CTABSEGID（<n>，<記憶體位置>）
CTABSEG（<記憶體位置>，<區段類型>）
CTABFSEG（<記憶體位置>，<區段類型>）
CTABMSEG（<記憶體位置>，<區段類型>）
CTABPOLID(<n>)
CTABPOL（<記憶體位置>）
CTABFPOL（<記憶體位置>）
CTABMPOL（<記憶體位置>）
```

含義

CTABNO：	決定**定義的**曲線表號碼（在靜態與動態 NC 記憶體中）
CTABNOMEM：	決定在特定的 <記憶體位置> 中**定義的**曲線表號碼
CTABFNO：	決定**可能剩餘**在特定 <記憶體位置> 中的曲線表號碼
CTABSEGID：	決定特定的 <區段類型>的曲線區段之號碼，該區段類型被曲線表號碼<n>所用
CTABSEG：	決定**用**於特定 <記憶體位置> 中特定 <區段類型> 之曲線區段號碼
CTABFSEG：	決定**可能剩餘**在特定 <記憶體位置> 中特定 <區段類型> 之曲線區段號碼
CTABMSEG：	決定在特定 <記憶體位置> 中特定 <區段類型> 之曲線區段的**最大可能**號碼
CTABPOLID：	決定被曲線表號碼<n>所用的曲線多項式之號碼
CTABPOL：	決定**用**於特定 <記憶體位置> 中曲線多項式的號碼
CTABFPOL：	決定**可能剩餘**在特定 <記憶體位置> 中的曲線多項式號碼
CTABMPOL：	決定在特定 <記憶體位置> 中曲線多項式的**最大可能**號碼
<n>：	曲線表的編號 **(ID)**

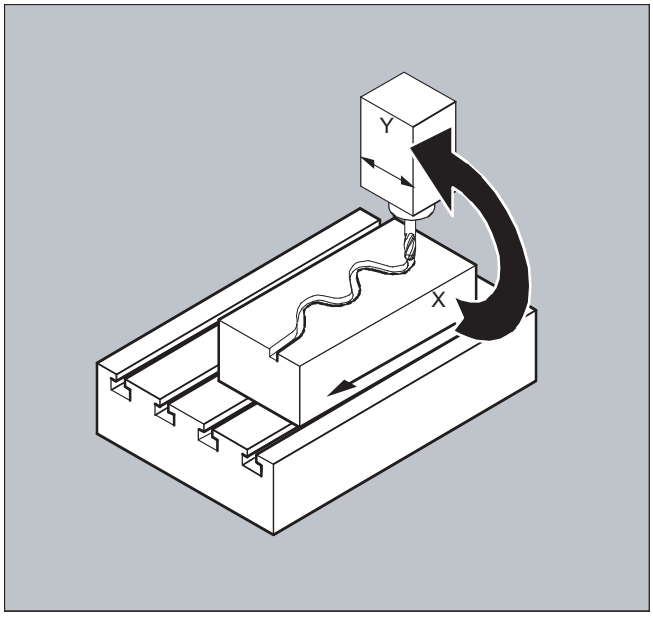
<記憶體位置>：	指定記憶體位置（選用）	
	「SRAM」	**境態** NC 記憶體
	「DRAM」	**動態** NC 記憶體
	附註： 若未替此參數程式設計值，會使用以 MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE 設定的預設記憶體位置。	
<區段類型>：	指定區段類型（選用）	
	「L」	線性區段
	「P」	多項式區段
	附註： 若沒有對此參數程式設計值，線性與多項式區段的總和會輸出。	

13.3 軸主動值耦合 (LEADON, LEADOF)

說明

SINUMERIK 828D 並未提供此功能！

利用軸主動值耦合，可同步移動先導軸與跟隨軸。 透過曲線表或是先導軸（如有必要亦可是模擬軸）某個位置上唯一計算得出的多項式，亦可指派跟隨軸位置。



先導軸是提供輸入值給曲線表的軸。 **跟隨軸**是利用曲線表接收算出之位置的軸。

實際值與設定點耦合

以下各項資料可作為主動值，亦即跟隨軸位置計算結果的輸出值：

- 先導軸位置實際值： 實際值耦合
- 先導軸位置設定點： 設定點值耦合

基本座標系統一律套用主控值耦合。

有關曲線表建立的資訊，請參閱〈曲線表〉章節。

語法

```
LEADON（<跟隨軸>，<先導軸>，<n>）
LEADOF（<跟隨軸>，<先導軸>）
```

或不指定主導軸而停用

```
LEADOF（<跟隨軸>）
```

您可從工件程式以及從同步動作的移動期間，啟動或停用主動值耦合。

含義

LEADON：	啟動主動值耦合
LEADOF：	停用主動值耦合
<跟隨軸>：	跟隨軸
<先導軸>：	主導軸
<n>：	曲線表編號
$SA_LEAD_TYPE：	切換設定點耦合與實際值耦合

停用主動值耦合，LEADOF

當您停用主動值耦合時，跟隨軸會再次成為一般指令軸！

軸主動值耦合與不同的操作狀態，RESET

根據機械資料中的設定，主動值耦合是以 RESET 停用。

來自同步動作的主動值耦合範例

在衝壓廠裏，將傳輸軸與輔助軸所構成之傳輸系統的軸與先導軸（立軸）做一般機械性耦合，是為了讓電子耦合系統能夠取代之。

這示範了電子傳輸系統如何取代機械傳輸系統。 耦合與去耦流程是**靜態同步動作**。

從先導軸 LV（立軸）控制傳輸軸與輔助軸時，其方法等同於利用曲線表加以定義的跟隨軸。

跟隨軸

X 進給或縱向軸
YL 閉合軸或橫向軸
ZL 起重軸
U 滾筒式進給、輔助軸
V 導頭、輔助軸
W 潤滑、輔助軸

動作

舉例來說，發生的動作包含下列同步動作：

- 啟動耦合，LEADON（<跟隨軸>，<先導軸>，<曲線表編號>）
- 停用耦合，LEADOF（<跟隨軸>，<先導軸>）

- 設定實際值，PRESETON（<軸>，<值>）
- 設定標記，$AC_MARKER[i]=<值>
- 耦合類型： 實際 / 虛擬主動值
- 逼近軸位置，POS[<軸>]=<值>

條件

快速數位輸入，利用布林運算子 AND 將時間變數$AC_MARKER 以及位置對照連結起來，作為評估條件。

說明

在以下範例中，換行、縮排以及使用**粗體**字型的唯一目的只是增進程式的可讀性。 對於控制系統而言，行號之後的所有文字便構成單行。

註解

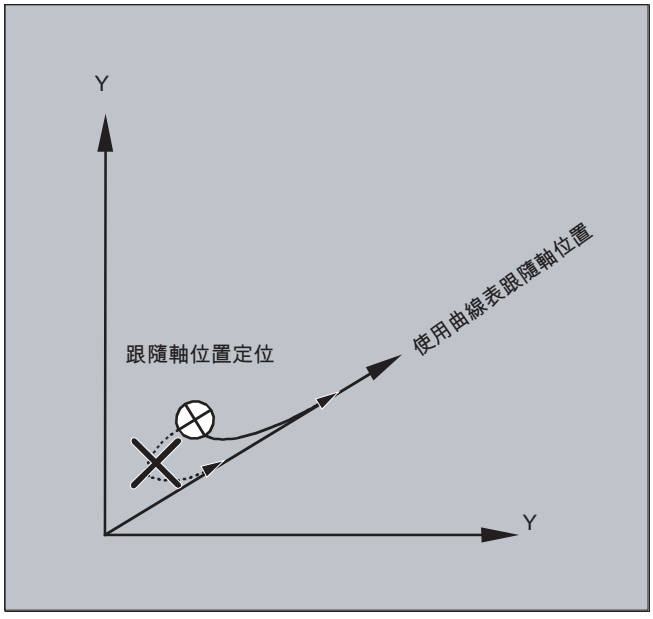
程式碼	註解
	；定義所有靜態同步動作。
	；****重置標記
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
	；**** E1 0=>1 傳輸 ON
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_MARKER[0]==0) DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) $AC_MARKER[0]=1	
	；**** E1 0=>1 耦合滾輪進給 ON
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0) DO LEADON(U,LW,4) PRESETON(U,0) $AC_MARKER[5]=1	
	；**** E1 0->1 耦合對齊準頭 ON
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0) DO LEADON(V,LW,4) PRESETON(V,0) $AC_MARKER[6]=1	
	；**** E1 0->1 潤滑耦合 ON
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[7]==0) DO LEADON(W,LW,4) PRESETON(W,0) $AC_MARKER[7]=1	
	；**** E2 0=>1 耦合 OFF
N30 IDS=3 EVERY ($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	

說明

主動值耦合要求先導軸與跟隨軸必須同步。 啟動主動值耦合時，跟隨軸必須在根據曲線表所計算得出的曲線定義允差範圍內，唯有如此才能達到上述同步要求。

跟隨軸位置的允差範圍是利用機械資料 MD 37200 加以定義：
COUPLE_POS_POL_COARSE A_LEAD_TYPE.

啟動主動值耦合時，若跟隨軸尚未在正確位置上，則等待所計算得出的跟隨軸位置設定點值大約等於真正跟隨軸位置時，將隨即自動開始同步執行。 跟隨軸將於同步程序期間以該軸設定點速度所定義的方向移動（此速度是利用 CTAB 曲線表從主控主軸計算得出）。

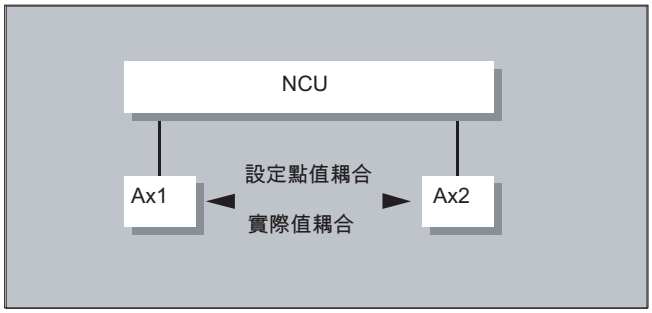


未同步

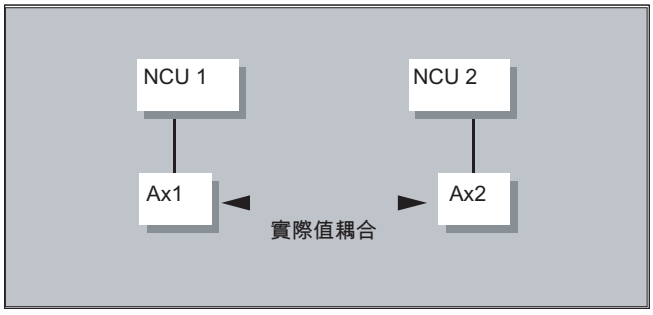
如果啟動主動值耦合時，所計算得出的跟隨軸位置正在遠離目前跟隨軸位置，那麼將無法執行同步。

實際值與設定點耦合

與實際值耦合相較，設定點耦合能夠提供更佳的「先導軸與跟隨軸同步」功能，因此為預設選項。



先導軸與跟隨軸必須由同一 NCU 插補，如此才能進行設定點耦合。 如果使用外部先導軸，則跟隨軸僅能以實際值與先導軸耦合。



切換功能可透過設定資料 $SA_LEAD_TYPE 進行程式設計。

跟隨軸停止時，您永遠必須在實際值耦合與設定點耦合之間進行切換。 當軸靜止不動時，您必須進行切換，然後才能重新同步。

應用範例

機台嚴重震動且無發生錯誤時，您無法讀取實際值。 如果您在壓製傳輸時使用主動值耦合，可能必須在最大震動的加工步驟中，將實際值耦合切換成設定點耦合。

使用設定點耦合時的**主控值模擬**

利用機械資料，您可從伺服中斷插補器與先導軸的連接。 如此一來，您不需實際移動先導軸，就能產生設定點耦合所需的設定點。

您可從下列變數讀取由設定點連結產生的主動值，然後便可在，例如，同步動作中使用該值：

- $AA_LEAD_P	主動值位置
- $AA_LEAD_V	主動值速率

建立主控值

您亦可選擇使用其他自程方式產生主動值。 以此方式產生的主動值是利用變數加以讀寫

- $AA_LEAD_SP 主動值位置
- $AA_LEAD_SV 主動值速率

您必須先將設定資料設定為 $SA_LEAD_TYPE = 2，然後才能開始使用這些變數。

耦合狀態

您可利用下列系變數在 NC 程式中查詢耦合狀態：

```
$AA_COUP_ACT[[axis]]
```

0：沒有啟用耦合
16： 主動值耦合為啟用中

同步動作的狀態管理

切換事件與耦合事件是利用時間變數加以管理：

```
$AC_MARKER[i] = n
```

是以
i 旗標編號與
n 狀態值加以管理

13.4 電子齒輪（EG）

「電子齒輪」函數可讓您根據線性移動單節，控制**跟隨軸**的移動，一個函數可輸入高達五個**先導軸**。 各個先導軸與跟隨軸的關係是以耦合係數加以定義。

將個別先導軸動作部分乘以各自的耦合係數後再加總，即可得出跟隨軸的動作部分。 啟動 EG 軸群組時，您可將已定義之位置的相關跟隨軸同步化。 所謂齒輪群組可以是：

- 已定義
- 啟動
- 停用
- 刪除

.

跟隨軸動作可選擇性取自

- 先導軸的設定點，以及
- 先導軸的實際值。

利用**曲線表**，各先導軸與跟隨軸之間的非線性關係亦可認知為一項延伸性資料（請參閱「路徑移動行為」一節）。 電子齒輪可以是串聯式架構，意即某一電子齒輪的跟隨軸可以是另一電子齒輪的先導軸。

13.4.1 定義電子齒輪（EGDEF）

透過指定跟隨軸以及一至五個先導軸（包括各自相關的耦合類型），即可定義 EG 軸群組。

需求條件

定義 EG 軸群組的先決條件：

不得定義跟隨軸的軸耦合（或：若有軸耦合則必須先以 EGDEL 刪除之）。

語法

```
EGDEF（跟隨軸、先導軸 1、耦合類型 1、先導軸 2、耦合類型 2、...）
```

含義

EGDEF：	定義電子齒輪	
跟隨軸：	受先導軸影響的軸	
先導軸 1 ，...， 先導軸 5	影響跟隨軸的軸	
耦合類型 1 ，...， 耦合類型 5	耦合類型 先導軸的耦合類型不需完全一致，但個別主控值必須各自單獨加以程式設計	
	值：	**含義：**
	0	跟隨軸受到對應先導軸之**實際值**的影響。
	1	跟隨軸受到對應先導軸之**設定點**的影響。

說明

若已定義 EG 軸群組，耦合係數預設為零。

說明

EGDEF 會觸發前置處理停止。 如果系統中有一或多個先導軸透過**曲線表**影響跟隨軸，含 EGDEF 之變速箱定義在使用時亦須保持一致。

範例

程式碼	註解
EGDEF（C，B，1，Z，1，Y，1）	；定義 EG 軸群組。 先導軸 B，Z，Y 透過設定點影響跟隨軸 C。

13.4.2 切換至電子變速箱（EGON、EGONSYN、EGONSYNE）

切換至 EG 軸群組的方法共有三種。

語法

格式變異 1：

選擇以未同步化方式切換至 EG 軸群組時，請使用：
EGON（FA，「單節變更模式」，LA1，Z1，N1，LA2，Z2，N2，...，LA5，Z5，N5）

格式變異 2：

選擇以同步化方式啟動 EG 軸群組時，請使用：
EGONSYN（FA，「單節變更模式」，SynPosFA，[，LAi，SynPosLAi，Zi，Ni]）

格式變異 3：

選擇以同步化方式採逼近模式切換至 EG 軸群組時，請使用：
EGONSYNE（FA，「單節變更模式」，SynPosFA，逼近模式，[，LAi，SynPosLAi，Zi，Ni]）

含義

格式變異 1：

FA	跟隨軸	
單節變更模式：	您可使用下列模式：	
	"NOC"	單節立刻變化
	"FINE"	單節變更是以「微調同步」之方式進行
	"COARSE"	單節變更是以「粗調同步」之方式進行
	"IPOSTOP"	依設定點同步執行單節變更
LA1，... LA5	先導軸	
Z1，... Z5	耦合係數 i 的分子	
N1，... N5	耦合係數 i 的分母 耦合係數 i = 分子 i / 分母 i	

先導軸必須事先以 EGDEF 指令加以指定，如此才能在啟用列上進行程式設計。 您必須程式設計至少一個先導軸。

格式變異 2：

FA	跟隨軸	
單節變更模式：	您可使用下列模式：	
	"NOC"	單節立刻變化
	"FINE"	單節變更是以「微調同步」之方式進行
	"COARSE"	單節變更是以「粗調同步」之方式進行
	"IPOSTOP"	依設定點同步執行單節變更
[，LAi，SynPosLAi，Zi，Ni]	（請勿寫入方括弧） 以下各項可寫上至少一個最多五個：	
LA1，... LA5	先導軸	

SynPosLAi	第 i 先導軸的同步位置
Z1，... Z5	耦合係數 i 的分子
N1，... N5	耦合係數 i 的分母 耦合係數 i = 分子 i / 分母 i

先導軸必須事先以 EGDEF 指令加以指定，如此才能在啟用列上進行程式設計。 透過跟隨軸（SynPosFA）與先導軸（SynPosLA）的已程式設計「同步位置」，其軸群組已轉譯成*同步*的位置可完成定義。 切換至群組時，若電子齒輪不在同步狀態，則跟隨軸將移動至其已定義的同步位置。

格式變異 3：

對應至格式變異 2 的參數外加：

逼近模式：	您可使用下列模式：	
	"NTGT"	逼近下一個時間最佳化的齒槽
	"NTGP"	逼近下一個路徑最佳化的齒槽
	"ACN"	沿負方向移動旋轉軸（絕對）
	"ACP"	沿正方向移動旋轉軸（絕對）
	"DCT"	對於已程式設計之同步位置為時間最佳化的逼近方式
	"DCP"	對於已程式設計之同步位置為距離最佳化的逼近方式

格式變數 3 將僅影響已與「模數先導軸」耦合的「模數跟隨軸」。 所謂時間最佳化將會考量跟隨軸的速限。

其他資訊

切換版本說明

格式變異 1：

切換至群組的瞬間，先導軸與跟隨軸的位置便儲存為“同步位置”。您可利用系統變數 $AA_EG_SYN 讀取“同步位置”。

格式變異 2：

如果耦合群組中包含模數軸，則其位置值為減折模數。如此確保系統是逼近下一個可能的同步位置（這就是所謂的“*相對同步*”：例如下一個齒槽）。如果系統向跟隨軸發佈“啟用跟隨軸手動超調”介面訊號 DB（30 ＋軸編號），DBX 26 位元 4，則將僅逼近同步位置。反之，程式將停止在 EGONSYN 單節上，並等待上述訊號完成後才輸出警報 16771，該警報將自行清除。

格式變異 3：

齒距（角度）的計算方式如下：360 * Zi/Ni.如果跟隨軸在進行呼叫時停止動作，則路徑最佳化傳回的內容將與時間最佳化完全一致。

如果跟隨軸已在動作中，NTGP 將不可考慮跟隨軸目前速率，逕自在下一個齒槽上進行同步化。如果跟隨軸已在動作中，NTGT 將根據跟隨軸目前速率，在下一個齒槽上進行同步化。軸亦會視需要減速。

曲線表

如果**曲線表**是用於其中一個先導軸：

Ni　線性耦合的耦合係數分母必須設為 0。（分母為 0 不合線性耦合之規定。）分子為零等於告訴控制系統：

Zi　是應使用的曲線表編號。指定編號的曲線表在電源開啟時就應已定義。

LAi　指定的先導軸利用耦合係數對應至耦合指定先導軸（線性耦合）。

關於使用曲線表以及串聯並同步化電子齒輪的詳細資訊，請參閱：

參考：

功能手冊，特殊功能；耦合軸與 ESR（M3），“耦合動作與先導值耦合”。

電源開啟時、重置時、操作模式變更時以及單節搜尋時的電子齒輪回應

- 電源開啟後**無**啟用的耦合。
- 耦合的啟用狀態不受重置或操作模式切換的影響。
- 單節搜尋期間，將不會執行或囤集有關切換、刪除與定義電子齒輪等的指令，系統將直接略過。

電子齒輪的系統變數

藉由電子齒輪的系統變數，工件程式可判定 EG 軸群組的目前狀態並視需要採取相關措施。

電子變速箱系統變數的名稱標示如下所示：

$AA_EG_ ...

或

$VA_EG_ ...

參考：

系統變數手冊

13.4.3 切換至電子變速箱（EGOFS，EGOFC）

切換出啟用之 EG 軸群組的方法共有三種。

程式設計

格式變異 1：

語法	含義
EGOFS（跟隨軸）	停用電子齒輪。 跟隨軸開始剎車直到靜止。 此項呼叫將觸發前置處理停止。

格式變異 2：

語法	含義
EGOFS（跟隨軸，先導軸 1， …，先導軸 5）	這樣的指令參數設定可讓您**選擇性**移除個別先導軸對於跟隨軸動作的影響。

您必須指定至少一個先導軸。 選擇性禁止指定之先導軸影響從動裝置。 此項呼叫將觸發前置處理停止。 如果此項呼叫仍包含啟用之先導軸，則從動裝置將在受到其影響下繼續操作。 如果這個方法排除了所有先導軸的影響，則跟隨軸開始剎車直到靜止。

格式變異 3：

語法	含義
EGOFC（跟隨主軸 1）	停用電子齒輪。 跟隨主軸將以停用瞬間當時所套用的速度 / 速率繼續移動。 此項呼叫將觸發前置處理停止。

說明

此項變異僅適用於主軸。

13.4.4 刪除電子齒輪的定義（EGDEL）

您必須切換出 EG 軸群組之後，才能刪除其定義。

程式設計

語法	含義
EGDEL（跟隨軸）	刪除軸群組的耦合定義。 在達到最大同時啟動之軸群組數量以前，您可藉由 EGDEF 定義其他軸群組。 此項呼叫將觸發前置處理停止。

13.4.5 旋轉進給率（G95）/ 電子齒輪（FPR）

FPR 指令可用來指定，某一電子齒輪的跟隨軸決定繞轉進給率。 請注意以下這個指令的相關事項：

- 將由電子齒輪跟隨軸的設定點速率決定進給率。
- 而設定點速率則是從（非路徑軸之）先導主軸與模數軸的速度及其相關耦合係數計算得出。
- 將不考慮線性先導軸或非模數先導軸的速度部分以及跟隨軸被覆蓋的動作。

13.5 同步主軸

同步操作涉及一個跟隨主軸（FS）以及一個先導主軸（LS），它們就合稱為「**同步主軸組**」。 當耦合為啟用狀態時（同步操作），跟隨主軸將依照已定義功能上的相互關係，仿效先導主軸移動。

您可利用通道專屬機械資料指派固定設定給每個機台的同步主軸組，或者透過 CNC 工件程式，針對特定應用定義之。 每條 NC 通道上最多可同時操作兩個同步主軸組。

關於下列耦合動作，請參閱工件程式

- 定義或變更
- 啟動
- 停用
- 刪除

.

此外，根據軟體狀態

- 進入同步狀態之前可能需要等待一段時間
- 單節變更方式可能有所改變
- 可能已選擇設定點耦合類型或實際值耦合類型，或者已指定主動主軸與跟隨主軸之間的角度偏移
- 啟動耦合時，將傳送跟隨軸的上一個程式
- 修正已測得或已知的同步差異

13.5.1 同步主軸： 程式設計（COUPDEF、COUPDEL、COUPON、COUPONC、COUPOF、COUPOFS、COUPRES、WAITC）

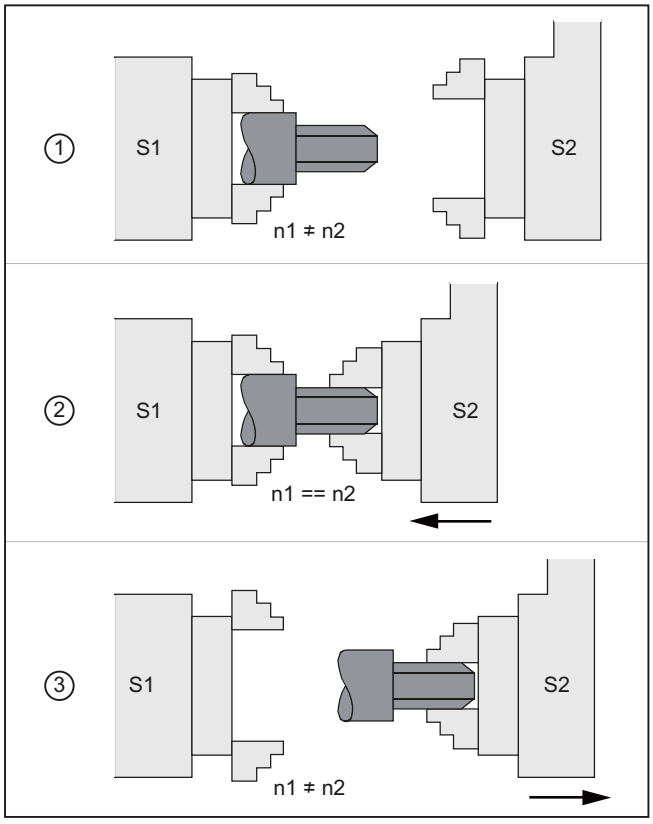
「同步主軸」使用可程式設計的轉換率，讓跟隨主軸 (FS) 及先導主軸 (LS) 速度同步移動。

此函數可支援下列模式：

- 速度同步（$n_{FS} = n_{LS}$）
- 位置同步 ($\phi_{FS} = \phi_{LS}$)
- 位置同步與角度偏移 ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

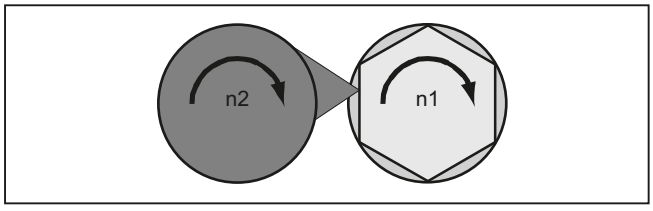
應用範例：

- 工件迅速轉移，例如加工背面，轉換率：1:1



① 同步速度
② 轉移工件
③ 加工背面

- 多稜邊加工（多邊形車削）、速度同步、轉換率：n_1：n_2



語法

```
COUPDEF(<FS>,<LS>,<ZFS>,<NLS>,<單節變更>,<耦合類型>)
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>,<單節變更>,<LS>,<單節變更>)
```

說明

簡化標記

對 COUPOF、COUPOFS、COUPRES 及 COUPDEL 陳述式而言，可使用不指定先導主軸的較短標記。

含義

COUPDEF：	以使用者自訂基準來定義/更改耦合
COUPON：	啟動耦合。跟隨主軸根據實際速度，調整成與先導主軸同步
COUPONC：	以先前的 M3 S...或 M4 S... 程式碼啟動時進行耦合。 立即傳輸跟隨主軸的速度差異。
COUPOF：	停用耦合。 • 以立即單節更改： COUPOF(<S2>,<S1>) • 跨過 POSFS 或 POSLS 停用位置後，單節才會變更： COUPOF(<S2>,<S1>,<POSFS>) COUPOF(<S2>,<S1>,<POSFS>,<POSLS>)
COUPOFS：	跟隨主軸停止時停用耦合。 以即時單節變更之方式儘快進行單節變更： COUPOFS(<S2>,<S1>) 僅於傳遞關閉位置後進行單節變更： COUPOFS(<S2>,<S1>,<POSFS>)
COUPRES：	將耦合參數重置為已設定之 MD 與 SD
COUPDEL：	刪除使用者自訂之耦合
WAITC：	等待同步執行狀態 （單節變更期間 NOC 增加至 IPO）
<FS>：	跟隨主軸指定
選配參數：	
<LS>：	主要主軸指定 含主軸編號之規格：例如 S2、S1

<ZFS>，<NLS>：	介於 FS 和 LS 之間的轉換比率。 <ZFS>/<NLS> = 分子/分母 預設值：<ZFS>/<NLS> = 1.0；可選擇是否指定分母	
<單節變更>：	單節變更行為 單節變更是：	
	"NOC"	立即
	"FINE"	到達「同步微調」
	"COARSE"	到達「同步粗調」
	"IPOSTOP"	到達 IPOSTOP；換句話說，在設定點基準同步之後（預設）
	單節變更行為為模態有效。	
<耦合類型>：	耦合類型：FS 與 LS 間的耦合	
	"DV"	設定點連接（預設）
	"AV"	實際值耦合
	"VV"	速度耦合
	耦合類型為模態。	
<POSFS>：	先導主軸與跟隨主軸間的角度偏移	
	值域：	0°… 359.999°
<POSFS>、 <POSLS>：	跟隨主軸與先導主軸的關閉位置 「待傳遞 POS_{FS}、POS_{LS} 之後立即啟用單節變更」	
	值域：	0°… 359.999°

範例

使用先導主軸與跟隨主軸進行加工

程式碼	註解
	先導主軸 = 主控主軸 = 主軸 1
	跟隨主軸 = 主軸 2
N05 M3 S3000 M2=4 S2=500	先導主軸以 3000 rpm 旋轉， 跟隨主軸以 500 rpm 旋轉
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	定義耦合（亦可加以設定）
...	
N70 SPCON	將先導主軸帶入閉迴圈位置控制 （設定點耦合）。
N75 SPCON(2)	將跟隨主軸帶入閉迴圈位置控制。
N80 COUPON(S2,S1,45)	即時耦合至偏移位置 = 45 度。
...	

程式碼	註解
N200 FA[S2]=100	定位速度 = 100 度 / 分鐘
N205 SPOS[2]=IC(-90)	沿負方向以 90° 覆蓋的方式移動。
N210 WAITC(S2,"Fine")	等待「微調」同步。
N212 G1 X...Y...F...	加工
...	
N215 SPOS[2]=IC(180)	沿正方向以 180° 覆蓋的方式移動。
N220 G4 S50	主控主軸停頓時間 = 50 轉數
N225 FA[S2]=0	啟用已設定的速率 (MD)。
N230 SPOS[2]=IC(-7200)	轉 20 圈。沿負方向以設定的速率移動。
...	
N350 COUPOF(S2,S1)	即時退耦，S=S2=3000
N355 SPOSA[2]=0	將 FS 停在 0 度。
N360 G0 X0 Y0	
N365 WAITS(2)	等待主軸 2。
N370 M5	停止 FS。
N375 M30	

程式設計速度差異

程式碼	註解
	先導主軸 = 主控主軸 = 主軸 1
	跟隨主軸 = 主軸 2
N01 M3 S500	先導主軸以 500 rpm 旋轉。
N02 M2=3 S2=300	跟隨主軸以 300 rpm 旋轉。
...	
N10 G4 F1	主控主軸的停頓時間。
N15 COUPDEF (S2,S1,-1)	耦合係數比為 1：1
N20 COUPON(S2,S1)	啟動耦合。以主要主軸的速度和耦合係數，計算出跟隨主軸的速度。
...	
N26 M2=3 S2=100	程式設計速度差。

速度不同時的移動傳輸範例

1.在上一個以 COUPON 進行跟隨主軸程式設計期間，啟動耦合

程式碼	註解
	先導主軸 = 主控主軸 = 主軸 1
	跟隨主軸 = 主軸 2
N05 M3 S100 M2=3 S2=200	先導主軸以 100 rpm 旋轉，跟隨主軸以 200 rpm 旋轉。
N10 G4 F5	主控主軸停頓時間 = 5 秒
N15 COUPDEF(S2,S1,1)	FS 對 LS 的轉換比率為 1.0 （預設值）。
N20 COUPON(S2,S1)	即時耦合至先導主軸。
N10 G4 F5	跟隨主軸以 100 rpm 旋轉。

2.在上一個以 COUPONC 進行跟隨主軸程式設計期間，啟動耦合

程式碼	註解
	先導主軸 = 主控主軸 = 主軸 1
	跟隨主軸 = 主軸 2
N05 M3 S100 M2=3 S2=200	先導主軸以 100 rpm 旋轉，跟隨主軸以 200 rpm 旋轉。
N10 G4 F5	主控主軸停頓時間 = 5 秒
N15 COUPDEF(S2,S1,1)	FS 對 LS 的轉換比率為 1.0 （預設值）。
N20 COUPONC（S2，S1）	即時耦合至先導主軸，並將先前的速度傳輸至 S2。
N10 G4 F5	S2 以 100 rpm + 200 rpm = 300 rpm 之速度旋轉

3.於跟隨主軸靜止的情況下利用 COUPON 啟動耦合

程式碼	註解
	先導主軸 = 主控主軸 = 主軸 1
	跟隨主軸 = 主軸 2
N05 SPOS=10 SPOS[2]=20	跟隨主軸 S2 處於定位模式。
N15 COUPDEF(S2,S1,1)	FS 對 LS 的轉換比率為 1.0 （預設值）。
N20 COUPON（S2，S1）	即時耦合至先導主軸。
N10 G4 F1	關閉耦合，S2 停在 20 度。

4.於跟隨主軸靜止的情況下利用 COUPONC 啟動耦合

說明

定位模式或軸模式

若耦合前跟隨主軸處於定位或軸模式，則不論使用 COUPON(<FS>,<LS>) 或 COUPONC(<FS>,<LS>)，跟隨主軸的動作皆相同。

說明

前導主軸與軸操作

若在定義耦合之前，前導主軸在軸操作中，從機械資料而來的速率臨界值 MD32000 $MA_MAX_AX_VELO（最大軸速率）將仍然使用，即使在啟用了耦合之後。為了避免此行為，在定義耦合之前，必須將軸切換到主軸模式（M3 S...或 M4 S...）。

其他資訊

設定耦合

對於已設定的耦合，LS 與 FS 透過機械資料定義。 已設定的主軸，無法在工件程式中更改。使用 COUPDEF 可以在工件程式中對耦合參數化（這種情況下，寫入保護皆為無效）。

使用者自訂之耦合

COUPDEF 可備用來定義或更改工件程式中的耦合。 若已啟用耦合，則必須在定義新耦合之前，先以 COUPDEL 來刪除。

於整體定義耦合：

COUPDEF(<FS>,<LS>,<TFS>,<TLS>,單節變更行為,耦合類型)

跟隨主軸（FS）與前導主軸（LS）

耦合以 FS 和 LS 的軸名下定義。 必須以各個 COUPDEF 敘述，來對軸名稱進行程式設計。. 其他耦合參數為模態，且必須被程式設計，若其變更。

範例：

```
COUPDEF(S2,S1)
```

轉換率

轉換率係依據 FS 與 LS 之間的轉速比定義之：

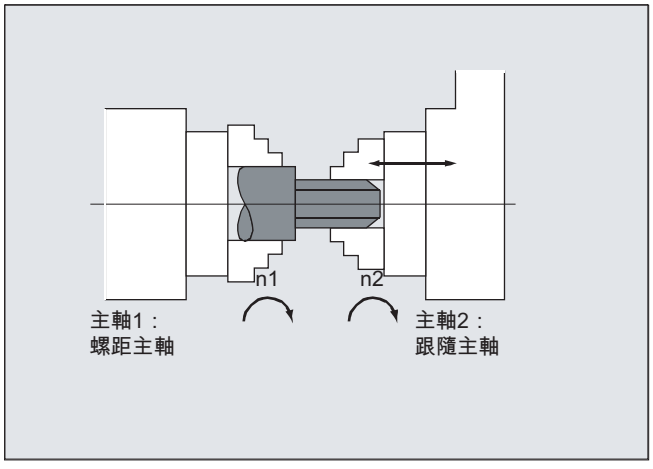
跟隨主軸 / 先導主軸 = 分子/分母

您必須程式設計分子。 分母不得程式設計。 因此分母的預設值設定為 1.0。

範例：

跟隨主軸 S2 與先導主軸 S1，轉換率 = 1/1

```
COUPDEF(S2, S1, 1.0)
```



說明

可立即更改轉換比率（當耦合生效且主軸旋轉中）。

單節變更行為 NOC、FINE、COARSE、IPOSTOP

下列的簡化標記可在程式設計單節更改行為時使用：

- 「NO」： 立即（預設）
- 「FI」： 到達「同步微調」
- 「CO」： 到達「同步粗調」
- 「IP」： 到達 IPOSTOP；亦即，在設定點基準同步之後

耦合類型

說明

耦合類型只有在耦合停用時才能變更。

啟用同步模式 COUPON，<POSFS>

- 以 LS 與 FS 間的任一角度偏移啟動耦合：
  - COUPON(S2,S1)
  - COUPON（S2）
- 以角度偏移 <POSFS> 啟用耦合
  <POSFS> 在正旋轉
  方向的 <POSFS> 值域內參考至先導主軸的 0° 位置： 0°… 359,999°
  - COUPON(S2,S1,30)

說明

角度偏移亦可於耦合為啟用狀態時變更。

定位跟隨主軸

即使已啟用同步主軸耦合，FS 仍可以定位在範圍 ±180° 之內，與 LS 無關。

- 使用 SPOS 的 FS 主軸定位
  範例： SPOS[2] = IC（-90）
  關於 SPOS 的進一步資訊，請參閱：
  參考資料：
  基礎程式設計手冊

差別速率

速度的差異會經由 FS 轉速因為 LS 移動的正負符號重疊，以及因為主軸程式設計的 FS 轉速，導致轉速控制模式及啟用同步主軸的耦合：

- 與 COUPONC 同步主軸耦合
- S<FS>=<轉速> [M<FS>=<旋轉方向>]

說明

補充條件

- 轉速 S... 不得以旋轉方向 M3/M4 程式設計。
- 覆蓋功能必須已經啟用，利用 LS 移動的同步主軸耦合 COUPONC 覆蓋主軸速度（M<旋轉方向> S<FS>）才會生效。
- 當套用覆蓋至跟隨主軸時，前導主軸的動態回應必須被限制在特定範圍，且沒有超過其動態臨界值。

關於速度差異的進一步資訊，請參閱：

參考資料：

功能手冊，延伸功能；同步主軸 (S3)

速率，加速度： FA、ACC、OVRA、VELOLIMA

軸速率與跟隨主軸的加速度，可用下列來程式設計：

- FA[SPI(S<n>)] 或 FA[S<n>] （軸速率）
- ACC[SPI(S<n>)] 或 ACC[S<n>] （軸加速度）
- OVRA[SPI(S<n>)] 與 OVRA[S<n>] (軸手動超調)
- VELOLIMA[SPI(S<n>)] 與 VELOLIMA[S<n>] （分別為軸速率的提高及降低）

其中 <n> = 1, 2, 3, ... (跟隨主軸的主軸號碼)

參考資料：

基礎程式設計手冊

說明

增加或減少最大的軸震動對主軸毫無影響。

關於軸動態反應的進一步資訊，請參閱：

參考資料：

功能手冊，延伸功能；旋轉軸 (R2)

可程式設計單節變更行為 WAITC

WAITC 可用來定義單節更改行為，例如，在以數個不同的同步條件（粗調，微調，IPOSTOP），更改了為耦合參數或位置動作之後。 若沒有指定同步條件，則將套用在 COUPDEF 定義中的單節更改行為。

範例

- 等待要為跟隨軸 S2 滿足的同步條件 FINE，以及要為跟隨軸 S4 滿足的 COARSE：WAITC(S2,"FINE",S4,"COARSE /)
- 根據要被滿足的 COUPDEF，等待同步條件： WAITC（ ）

停用耦合 COUPOF

COUPOF 可被用來定義耦合的關閉行為：

- 以利及的單節更改來停用耦合：
  - COUPOF（S2，S1）（具有前導主軸的規格）
  - COUPOF（S2）（沒有前導主軸的規格）
- 在已跨過關閉位置之後，停用耦合。 在已跨過關閉位置之後，發生了單節更改。
  - COUPOF(S2,S1,150) （關閉位置 FS： 150°)
  - COUPOF(S2,S1,150,30) （關閉位置 FS： 150°，LS： 30°)

以下列主軸停止 COUPOFS 來停用耦合

COUPOFS 可用來定義，以跟隨主軸停止所進行的耦合之關閉行為：

- 以跟隨主軸停止與立即單節更改所進行的停用耦合：
  - COUPOFS（S2，S1）（具有前導主軸的規格）
  - COUPOFS（S2）（沒有前導主軸的規格）
- 在關閉位置已被跟隨主軸停止跨過之後，停用耦合。 在已跨過關閉位置之後，發生了單節更改。
  - COUPOFS(S2,S1,150)（關閉位置 FS： 150°)

刪除耦合 COUPDEL

COUPDEL 刪除耦合：

- COUPDEL(S2,S1)（指定先導主軸）
- COUPDEL（S2）（沒有前導主軸的規格）

重置耦合參數，COUPRES

COUPRES 啟動了在機械資料和設定資料中參數化的耦合值：

- COUPRES(S2,S1)（指定先導主軸）
- COUPRES（S2）（沒有前導主軸的規格）

系統變數

- 跟隨主軸的目前耦合狀態
  目前跟隨主軸的耦合狀態，可透過以下讀取位元編碼：
  <值> = $AA_COUP_ACT[<FS>]

位元	<值>	含義
-	0	沒有啟用耦合
2	4	同步主軸耦合為生效狀態

附註

- 參考軸模式的所有其他數值
- 若主軸為跟隨主軸或數個耦合，則所有耦合的耦合狀態值會以總體狀態傳回。

- 目前角度偏移
  跟隨主軸至先導主軸的目前角度偏移可以透過以下讀取：
  – $AA_COUP_OFFS[<FS>]（設定點側的角度偏移）
  – $VA_COUP_OFFS[<FS>]（實際值側的角度偏移）
  應用範例
  取消跟進模式之後，改正 NC 程式中的角度偏移差距：
  角度偏移差距 = 已邊成的角度偏移 - 系統變數

參考資料

有關系統變數的詳細資訊，請參閱：

清單手冊，系統變數

13.6 一般耦合 (CP...)

「一般耦合」屬於一般的耦合功能，結合了現有的耦合類型之所有耦合特性（耦合動作、主動值耦合、電子齒輪箱及同步化主軸）。

此功能允許彈性的程式設計：

- 使用者可以選擇其應用所需的耦合屬性（建構單節原則）。
- 可以個別程式設計每個耦合的屬性。
- 可以更改已定義耦合的耦合屬性（例如耦合係數）。
- 可以稍後使用其他的耦合屬性。
- 跟隨軸的座標參考系統（基準座標系統或機械座標系統）為可程式設計。
- 特定的耦合屬性也可以使用同步化動作加以程式設計。
  參考資料：功能手冊，同步動作

說明

透過調整循環支援先前耦合呼叫的耦合動作 (TRAIL*)、主動值耦合 (LEAD*)、電子式齒輪箱 (EG*) 及同步化主軸 (COUP*)。

所有關鍵字及耦合特性的概觀

下表提供一般耦合的所有關鍵字，以及可程式設計耦合特性的概觀：

關鍵字	耦合特性 / 意義	語法
CPDEF	建立耦合模組	CPDEF=(<FAx>)
CPDEL	刪除耦合模組	CPDEL=(<FAx>)
CPLA	先導軸定義	CPLA[<FAx>]=(<LAx>)
CPLDEF	先導軸定義及建立耦合模組 （使用 CPDEF + CPLA 也可行）	CPLDEF[<FAx>]=(<LAx>) 或 CPDEF=(<FAx>) CPLA[<FAx>]=(<LAx>)
CPLDEL	刪除耦合模組的先導軸 （使用 CPDEF + CPLA 也可行）	CPLDEL[<FAx>]=(<LAx>) 或 CPDEL=(<FAx>) CPLA[<FAx>]=(<LAx>)
CPON	開啟耦合模組	CPON=(<FAx>)
CPOF	關閉耦合模組	CPOF=(<FAx>)

關鍵字	耦合特性 / 意義	語法		
CPLON	開啟耦合模組的先導軸	CPLON[<FAx>]=<LAx>		
CPLOF	關閉耦合模組的先導軸	CPLOF[<FAx>]=<LAx>		
CPLNUM	耦合係數的分子	CPLNUM[FAx,LAx]=<值>		
CPLDEN	耦合係數的分母	CPLDEN[FAx,LAx]=<值>		
CPLCTID	曲線表的編號	CPLCTID[FAx,LAx]=<值>		
CPLSETVAL	耦合參考	CPLSETVAL[FAx,LAx]="<耦合參考>"		
		"<耦合參考>"：	"CMDPOS"	設定點值耦合
			"CMDVEL"	速度耦合
			"ACTPOS"	實際值耦合
CPFRS	座標參考系統	CPFRS[FAx]="<座標參考>"		
		"<座標參考>"：	"BCS"	基本座標系統
			"MCS"	機械座標系統
CPBC	單節變更條件	CPBC[FAx]="<單節更改條件>"		
		"<單節更改條件>"：	"NOC"	執行單節更改與耦合狀態無關。
			"IPOSTOP"	以設定點同步執行單節更改。
			"COARSE"	以實際值同步「粗調」執行單節更改。
			"FINE"	以實際值同步「微調」執行單節更改。
CPFPOS + CPON	開啟時跟隨軸的同步位置	CPON=FAx CPFPOS[FAx]=<值>		
CPLPOS + CPON	開啟時主導軸的同步位置	CPLPOS[FAx,LAx]=<值>		

關鍵字	耦合特性 / 意義	語法		
CPFMSON	同步模式	CPFMSON[FAx]="<同步模式>"		
		"<同步模式>"：	"CFAST"	耦合為封閉時間最佳化。
			"CCOARSE"	只有根據耦合規則要求的跟隨軸位置位在目前跟隨軸位置的範圍內時，耦合才會封閉。
			"NTGT"	下一個逼近時間最佳化的齒槽。
			"NTGP"	下一個逼近路徑最佳化的齒槽。
			"NRGT"	依據齒輪數與齒數的比率，以時間最佳化的方式逼近下一個區段。
			"NRGP"	依據齒輪數與齒數的比率，以路徑最佳化的方式逼近下一個區段。
			"ACN"	僅適用於旋轉軸！ 旋轉軸以軸的負方向，朝同步化位置移動。立即實現同步化。
			"ACP"	僅適用於旋轉軸！ 旋轉軸以軸的正方向，朝同步化位置移動。立即實現同步化。
			"DCT"	僅適用於旋轉軸！ 旋轉軸以時間最佳化，朝程式設計的同步化位置移動。立即實現同步化。
			"DCP"	僅適用於旋轉軸！ 旋轉軸以路徑最佳化，朝程式設計的同步化位置移動。立即實現同步化。

關鍵字	耦合特性 / 意義	語法		
CPFMON	跟隨軸開啟時的行為	CPFMON[FAx]= "<開啟時行為>"		
		"<開啟時行為>"：	"STOP"	僅適用於主軸！ 開啟之前跟隨主軸的生效動作停止。
			"CONT"	僅適用於主軸及主移動軸！ 跟隨軸/主軸的目前動作移到耦合做為開始動作。
			"ADD"	僅適用於主軸！ 除了目前的重疊動作以外，耦合的動作分量（亦即跟隨軸/主軸的目前動作）將繼續成為重疊動作。
CPFMOF	跟隨軸完全關閉時的行為	CPFMOF[FAx]="<關閉時行為>"		
		"<關閉時行為>"：	"STOP"	停止跟隨軸/主軸。 生效的覆蓋動作也煞停至完全靜止。接著將耦合開啟
			"CONT"	僅適用於主軸及主移動軸！ 跟隨主軸將以停用瞬間當時所套用的速度 / 速率繼續移動。
CPFPOS + CPOF	關閉時跟隨軸的關閉位置	CPOF=(FAx) CPFPOS[FAx]=<值>		

關鍵字	耦合特性 / 意義	語法		
CPMRESET	RESET 的耦合行為	CPMRESET[FAx]="<重置行為>"		
		"<重置行為>"：	"NONE"	繼續耦合的目前狀態。
			"ON"	建立適當的耦合模組以後，即開啟耦合。啟動所有已定義的先導軸。啟動全部或部份的這些先導軸時亦會執行，即使已經完整啟動耦合也會重新同步化。
			"OF"	生效的覆蓋動作也煞停至完全靜止。接著停用耦合。建立不含明確定義的相關耦合模組時 (CPDEF)，會刪除耦合模組。否則將保留，仍可繼續使用。
			"OFC"	可能僅適用於主軸！ 跟隨主軸將以停用瞬間當時所套用的速度 / 速率繼續移動。關閉耦合。建立不含明確定義的相關耦合模組時 (CPDEF)，會刪除耦合模組。否則將保留，仍可繼續使用。
			"DEL"	生效的覆蓋動作也煞停至完全靜止。接著停用耦合，然後刪除。
			"DELC"	可能僅適用於主軸！ 跟隨主軸將以停用瞬間當時所套用的速度 / 速率繼續移動。停用耦合，然後刪除。

關鍵字	耦合特性 / 意義	語法		
CPMSTART	工件程式啟動時的耦合行為	CPMSTART[FAx]="<啟動行為>"		
		"<啟動行為>"：	"NONE"	繼續耦合的目前狀態。
			"ON"	耦合開啟。啟動所有已定義的先導軸。啟動全部或部份的這些先導軸時亦會執行，即使已經完整啟動耦合也會重新同步化。
			"OF"	關閉耦合。建立不含明確定義的相關耦合模組時 (CPDEF)，會刪除耦合模組。否則將保留，仍可繼續使用。
			"DEL"	停用耦合，然後刪除。
CPMPRT	透過程式測試進行單節搜尋執行時，工件程式啟動時的耦合回應	CPMPRT[FAx]="<啟動行為>"		
		"<啟動行為>"：	參閱 CPMSTART	
CPLINTR	先導軸輸入值的偏移值	CPLINTR[FAx,LAx]=<值>		
CPLINSC	先導軸輸入值的調整比例係數	CPLINSC[FAx,LAx]=<值>		
CPLOUTTR	耦合輸出值的偏移值	CPLOUTTR[FAx,LAx]=<值>		
CPLOUTSC	耦合輸出值的調整比例係數	CPLOUTSC[FAx,LAx]=<值>		
CPSYNCOP	位置同步「粗調」的臨界值	CPSYNCOP[FAx]=<值>		
CPSYNFIP	位置同步「細調」的臨界值	CPSYNFIP[FAx]=<值>		
CPSYNCOP2	「粗調」位置同步化的第二門檻值	CPSYNCOP2[FAx]=<值>		
CPSYNFIP2	「微調」位置同步化的第二門檻值	CPSYNFIP2[FAx]=<值>		
CPSYNCOV	速度同步「粗調」的臨界值	CPSYNCOV[FAx]=<值>		
CPSYNFIV	速度同步「細調」的臨界值	CPSYNFIV[FAx]=<值>		

關鍵字	耦合特性 / 意義	語法		
CPMBRAKE	針對特定停止訊號及停止指令的跟隨軸回應	CPMBRAKE[FAx]=<位元編碼值>		
CPMVDI	針對特定 NC/PLC 介面訊號的跟隨軸回應	CPMVDI[FAx]=<位元編碼值>		
CPMALARM	抑制特殊耦合相關警報輸出	CPMALARM[FAx]=<位元編碼值>		
CPSETTYPE	耦合類型	CPSETTYPE[FAx]="<耦合類型>"		
		"<耦合類型>"：	"CP"	可任意編寫程式
			"TRAIL"	耦合類型「耦合動作」
			"LEAD"	耦合類型「主動值耦合」
			"EG"	耦合類型「電子式齒輪箱」
			"COUP"	耦合類型「同步化主軸」

FAx：跟隨軸/主軸

LAx：先導軸/主軸

說明

未明確程式設計的耦合特性（於同步動作的工件程式中），使用其預設設定值成為有效。

依據關鍵字 CPSETTYPE 的設定值，而非預設設定值 (CPSETTYPE="CP")，預設的耦合特性成為有效。

參考資料

若需更多有關一般耦合的詳細資訊，請參閱：

- 功能手冊，特殊功能；M3：軸耦合，章節：「一般耦合」

13.7 主 / 從耦合 (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

「主 / 從耦合」可：

- 將從動軸耦合到主動軸，只能在相關的軸都停止時進行。
- 對**旋轉**、速率控制的主軸進行耦合 / 去耦合動作。
- 進行動態配置。

說明

定位模式

只有在靜止時，才能對定位模式下的軸及主軸執行耦合關閉和開啟動作。

語法

```
MASLON（<從動_1>，<從動_2>，...）
MASLOF（<從動_1>，<從動_2>，...）
MASLOFS（<從動_1>，<從動_2>，...）

動態配置：
MASLDEF（<從動_1>，<從動_2>，...，<主動>）
MASLDEL（<從動_1>，<從動_2>，...）
```

含義

MASLON：	啟用暫時主 / 從耦合	
	<從動_x>，...：	從動軸 1 ... n
MASLOF：	對啟用的主 / 從耦合去耦合	
	<從動_1>，...：	從動軸 1 ... n
MASLOFS：	對主 / 從耦合去耦合，並自動煞住從動主軸（請參閱附註「速度控制模式下的主軸耦合行為！」）	
	<從動_1>，...：	從動軸 1 ... n
MASLDEF：	從工件程式建立或變更主 / 從群組。	
	<從動_1>，...：	從動軸 1 ... n
	<主動>：	主控軸

MASLDEL：	將主控 / 附屬耦合分開並刪除群組定義	
	<從動_1>，...：	從動軸 1 ... n
	附註： 將保留機械資料中所指定的主 / 從定義。	

說明

速度控制模式下的主軸耦合行為

對於速度控制模式下的主軸，可透過以下機械資料明確定義 MASLON、MASLOF、MASLOFS 和 MASLDEL 的耦合行為：

MD37263 $MA_MS_SPIND_COUPLING_MODE

如果將 MD37263 預設為 0，則從動軸僅能在相關軸皆完全靜止的情況下進行耦合及去耦。MASLOFS 等同於 MASLOF。

當 MD37263 = 1 時，將立即執行耦合指令及動作。使用 MASLON 時，將立即關閉耦合；使用 MASLOFS 或 MASLOF 時則將立即開啟耦合。使用 MASLOF 時，從動主軸將保持原來的速度繼續旋轉，直到設定新速度為止。不過，使用 MASLOFS 會自動煞車。

說明

使用 MASLOF/MASLOFS 時，不需隱含前置處理停止。由於少了前置處理停止，在下一個程式設計之前，附屬軸的$P 系統變數不會提供更新後的數值。

說明

您可利用 PRESETON 讓從動軸實際值與主動軸實際值同步化。若要這麼做，您必須短暫關閉永久 / 從動耦合，以便將不具參考點之從動軸的實際值設為電源開啟時的主動 / 軸值。然後耦合便重新建立為永久耦合。

利用以下機械資料設定啟動永久主 / 從耦合：
MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1
此設定對臨時耦合的語言指令無效。

範例

範例 1：設定主/從耦合從動軸的實際值

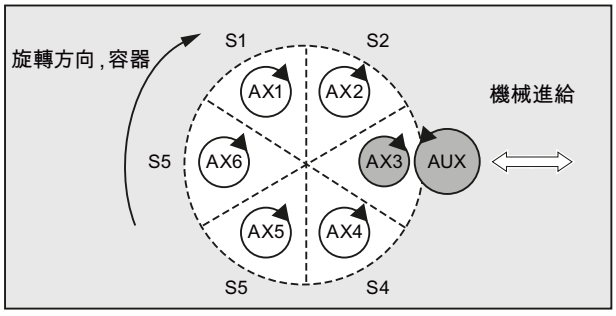
對於永久的主/從耦合，PRESETON 可將從動軸實際值設定為主動軸值。

程式碼	註解
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	；停用從動軸的永久耦合
NEWCONF	；啟動機械資料變更
STOPRE	

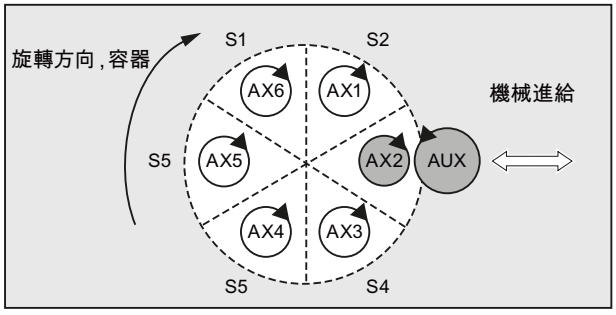
程式碼	註解
MASLOF（Y1）	；停用暫時耦合
PRESETON(AX2,$VA_IM(M_AX))	；從動軸實際值 = 主動軸實際值
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	；啟動從動軸的永久耦合
NEWCONF	；啟動機械資料變更

範例 2：主 / 從耦合的動態設定

軸容器旋轉之後若要啟用其他主軸的耦合，必須先將先前的耦合退耦、清除設定並設定新的耦合。



圖像 13-1　在軸容器旋轉之前



圖像 13-2　在軸容器旋轉一個插槽之後

程式碼	註解
MASLDEF（AUX，S3）	；AUX：從，S3：主 = AX3
MASLON（AUX）	；耦合開啟
M3=3 S3=4000	；旋轉主
MASLDEL（AUX）	；分離及刪除耦合
AXCTSWE（CT1）	；啟用軸容器旋轉
MASLDEF（AUX，S3）	；AUX：從，S3：主 = AX2

參考資料

- 功能手冊特殊功能，章節「TE3：速率/轉矩耦合，主從耦合」
- 功能手冊延伸功能，章節「B3：分散式系統 - 僅限 840D sl」>「NCU 連結」>「軸容器」

同步動作 14

14.1 同步動作之定義

同步動作於工件程式單節中定義。任何不屬於同步動作的進一步指令不得在該單節內設定。

同步動作包含下列元件：

有效性，ID 編號（選擇性）	條件部分（選擇性）			動作部分		
	頻率	G 指令（選擇性）	條件	關鍵字	G 指令（選擇性）	動作
--- [1] ID=<編號> IDS=<編號>	--- [1] WHENEVER FROM WHEN EVERY	G...	邏輯運算式	DO	G...	動作 1 ... 動作 n

[1] 未設定

語法

```
DO <動作 1> ... <動作 n>
<頻率> [<G 功能>] <條件> DO <動作 1> ... <動作 n>
ID=<編號> <頻率> [<G 功能>] <條件> DO <動作 1> ... <動作 n>
IDS=<編號> <頻率> [<G 功能>] <條件> DO <動作 1> ... <動作 n>
```

參考資料

同步動作功能的詳細說明，請參閱：

功能手冊，已同步動作

振盪 15

15.1 非同步震盪（OS、OSP1、OSP2、OST1、OST2、OSCTRL、OSNSC、OSE、OSB）

振盪軸以所定義的進給率在反轉點 1 及 2 之間前後移動，直至振盪運動停用為止。

振盪運動期間，可依所需對其他軸進行插補。 經由路徑移動或定位軸，可進行連續性進給，不過，**無特定關係**存在於振盪運動與進給運動之間。

非同步振盪的屬性

- 非同步振盪不受制於區塊限度，以軸為基礎而進行有效運作。
- 工件程式確保振盪移動之單節導向啟用。
- 不可進行數個軸的聯合插補，也不可將振盪軸重疊運作。

程式設計

下列指令可用於啟用及控制工件程式的非同步震盪。

程式化的數值輸入至含主執行時間中區塊同步化的相關設定資料內，此數值持續有效，直至再次變更為止。

語法

```
OSP1[<軸>]=<值> OSP2[<軸>]=<值>
OST1[<軸>]=<值> OST2[<軸>]=<值>
FA[<軸>]=<值>
OSCTRL[<軸>]=（<設定選項>，<重置選項>）
OSNSC[<軸>]=<值>
OSE[<軸>]=<值>
OSB[<軸>]=<值>
OS[<軸>] = 1
OS[<軸>] = 0
```

含義

<軸>：	震盪軸的名稱		
OS：	啟動/停用震盪		
	值：	1	切換震盪為**開**
		0	切換震盪為**關**

OSP1：	定義反轉點 1 的位置		
OSP2：	定義反轉點 2 的位置 **附註：** 若啟用增量移動，則該位置會增量計算至程式設計在 NC 程式中的最後一個對應反轉位置。		
OST1：	以[秒]為單位定義反轉點 1 的停止時間		
OST2：	以[秒]為單位定義反轉點 2 的停止時間		
	<值>：	-2	不等待精確停止而持續進行插補運作
		-1	等待精確停止粗調
		0	等待精確停止微調
		>0	等待精確停止微調，然後等待特定停止時間 **附註：** 停止時間的單位與使用 G4 程式設計的停止時間相同。
FA：	定義進給率 進給率為所定義的定位軸進給率。 若未定義進給率，則適用儲存於機械資料中的數值。		

<table>
<tr><td>OSCTRL：</td><td colspan="3">指定設定與重置選項
選項值 0 至 3 將停用時反轉點上的行為轉為密碼。 可選擇 0 至 3 其中一個變數。 剩餘的設定可自由與選擇的變數相結合。 多重選項會附加加號（+）字元。</td></tr>
<tr><td></td><td><值>：</td><td>0</td><td>在停用震盪時停在下一個反轉點上（預設）
附註：
只有在重置值 1 與值 2 時才可使用。</td></tr>
<tr><td></td><td></td><td>1</td><td>在振盪運作停用時，在反轉點 1 停止</td></tr>
<tr><td></td><td></td><td>2</td><td>在振盪運作停用時，在反轉點 2 停止</td></tr>
<tr><td></td><td></td><td>3</td><td>在振盪運作停用時，若未程式化滑磨衝程，不可逼近反轉點</td></tr>
<tr><td></td><td></td><td>4</td><td>在滑磨後逼近結束位置</td></tr>
<tr><td></td><td></td><td>8</td><td>若透過刪除剩餘距離的方式將震盪取消，則接下來需執行滑磨衝程並在必要時逼近結束位置。</td></tr>
<tr><td></td><td></td><td>16</td><td>若以刪除剩餘距離的方式取消震盪，則需與關機時一樣逼近對應的反轉點。</td></tr>
<tr><td></td><td></td><td>32</td><td>新進給只在下個反轉點後有效</td></tr>
<tr><td></td><td></td><td>64</td><td>FA 等於 0，FA = 0： 路徑覆蓋啟動中
FA 不等於 0，FA <> 0： 速度覆蓋啟動中</td></tr>
<tr><td></td><td></td><td>128</td><td>用於旋轉軸 DC（最短路徑）</td></tr>
<tr><td></td><td></td><td>256</td><td>滑磨衝程是雙衝程（預設）。 1=單衝程。</td></tr>
<tr><td></td><td></td><td>512</td><td>首先逼近起始位置</td></tr>
<tr><td>OSNSC：</td><td colspan="3">定義滑磨衝程次數</td></tr>
<tr><td>OSE：</td><td colspan="3">定義在停用震盪後要逼近的結束位置（使用工件座標系統）。
附註：
程式設計「OSE」時，選項 4 在「OSCTRL」會不明確的生效。</td></tr>
<tr><td>OSB：</td><td colspan="3">定義在啟用震盪前要逼近的開始位置（使用工件座標系統）。
在反轉點 1 前先逼近開始位置。若開始位置與反轉位置 1 重疊，則接下來逼近反轉位置 2。 當到達開始位置後，不會套用停止時間，即便此位置與反轉位置 1 重疊亦然；取而代之的是，該軸會等候精確停止微調訊號。 已滿足任何已設定的精確停止條件。
附註：
設定資料 SD43770 $SA_OSCILL_CTRL_MASK 中的位元 9 需設定以便初始化至開始位置的逼近。</td></tr>
</table>

範例

範例 1： 將震盪軸在兩個反轉點間震盪

振盪軸 Z 會在位置 10 與 100 間振盪。反轉點 1 會以精確停止微調逼近，反轉點 2 會以精確停止粗調逼近。 震盪軸的進給率需為 250。在加工操作結尾時需執行 3 次滑磨衝程且震盪需逼近結束位置 200。進給軸的進給率需為 1 且以 X 方向進行的進給需在位置 15 結束。

程式碼	註解
WAITP（X，Y，Z）	；初始設定。
G0 X100 Y100 Z100	；切換至定位軸運作。
WAITP（X，Z）	
OSP1[Z]=10 OSP2[Z]=100	；反轉點 1、反轉點 2。
OSE[Z]=200	；結束位置。
OST1[Z]=0 OST2[Z]=-1	；在 U1 的停止時間： 精確停止微調
	；在 U2 的停止時間： 精確停止粗調
FA[Z]=250 FA[X]=1	；振盪軸的進給，進給軸
OSCTRL[Z]=(4,0)	；設定選項。
OSNSC[Z]=3	；3 次火花消除衝程。
OS[Z]=1	；開始震盪。
WHEN $A_IN[3]==TRUE DO DELDTG（X）	；剩餘距離的刪除。
POS[X]=15	；起始位置 X 軸。
POS[X]=50	；結束位置 X 軸。
OS[Z]=0	；停止震盪。
M30	

說明

「OSP1[Z]=...」至「OSNCS[Z]=...」指令順序亦可程式設計在一個單節中。

範例 2： 使用反轉位置的線上修改進行震盪

可於工件程式中設定非同步震盪之必要設定資料。

若直接於程式中說明設定資料，則變更會於前置處理期間生效。 同步回應可透過 STOPRE 前置處理停止達成。

程式碼	註解
$SA_OSCILL_REVERSE_POS1[Z]=-10	
$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP（Z）	
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0	；若震盪軸之實際值超出反轉點，則進給軸將停止。
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0	

程式碼	註解
OS[Z]=1 FA[X]=1000 POS[X]=40	；啟動震盪。
OS[Z]=0	；停用震盪。
M30	

其他資訊

振盪軸

振盪軸有以下特點：

- 所有的軸都可做為振盪軸之用。
- 數個振盪軸可同時有效（至多： 定位軸的數量）。
- 線性插補 G1 對振盪軸總是有效 – 無論 G 指令目前於程式中是否有效。

振盪軸可以：

- 做為動態轉換的輸入軸而運作
- 做為門式（gantry）及聯合運動軸的導軸而運作
- 移動：
  - 無震動限制「BRISK」
    或
  - 具震動限制「SOFT」
    或
  - 含有彎處的加速度曲線（做為定位軸）。

振盪反轉點

在振盪位置如下定義時，必須將目前的偏移量列入考量：

- 絕對規格
  「OSP1[Z]=<值>」
  反轉點位置 = 偏移總量 + 程式化數值
- 相對規格
  「OSP1[Z]=IC(<值>)」
  反轉點位置 = 反轉點 1 + 程式化數值

範例：

程式碼
N10 OSP1[Z]=100 OSP2[Z]=110
...
N40 OSP1[Z]=IC(3)

WAITP

如果是以幾何軸來執行振盪，則必須將此軸為含「WAITP」的振盪而啟用。

當震盪結束時，「WAITP」會再次被用來輸入震盪軸作為定位軸，因此可以繼續正常使用。

同步動作之震盪與停止次數

一旦設定停止時間期滿，內部單節更改會於震盪期間執行（由與軸的新剩餘距離顯示）。單節更改時會檢查關閉功能。 關閉功能係依照動作順序的控制設定所定義（OSCTRL）。*此動態回應可由受進給手動超調影響。*

然後，可能會在火花消除衝程開始或結束位置逼近前執行一震盪衝程。 *儘管看起來關閉回應已更改，然而實際上並非如此。*

15.2 由同步動作控制的震盪（OSCILL）

此震盪模式下，進給動作可能只可於反轉點或於已定義之反轉區域中執行。

視需求而定，震盪移動可為

- 連續的，或
- 在進給執行完成前保持停止狀態。

語法

1. **定義震盪參數**
2. **定義動作－同步動作**
3. **指派軸，定義進給**

含義

OSP1[<震盪軸>]=	反轉點 1 位置
OSP2[<震盪軸>]=	反轉點 2 位置
OST1[<震盪軸>]=	於反轉點 1 的停止時間，單位為秒
OST2[<震盪軸>]=	於反轉點 2 的停止時間，單位為秒
FA[<震盪軸>]=	振盪軸的進給
OSCTRL[<震盪軸>]=	設定或重置選項
OSNSC[<震盪軸>]=	火花消除衝程次數
OSE[<震盪軸>]=	結束位置
WAITP（<震盪軸>）	啟用震盪之軸

軸指派，進給
OSCILL[<震盪軸>]=（<進給軸 1>，<進給軸 2>，<進給軸 3>）
POSP[<進給軸>]=（<結束位置>，<部分長度>，<模式>）

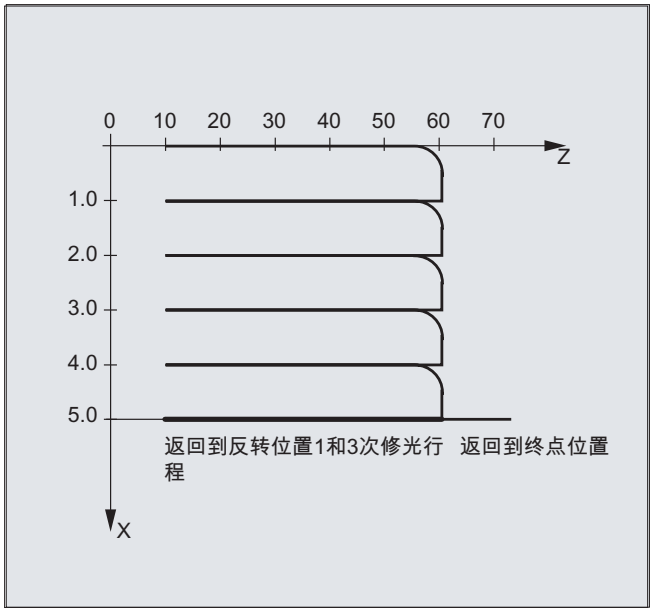
OSCILL：	為震盪軸指派進給軸或軸
POSP：	定義完整與部分進給（請參閱〈檔案與程式管理〉章節）
結束位置：	當所有部分進給均移動後的進給軸結束位置。
部分長度：	於反轉點 / 反轉區域的部分進給長度
模式：	將完整進給分為部分進給的分割 = 兩個相同尺寸的剩餘步進（預設）； = 所有相同尺寸的部分進給

動作－同步動作

WHEN... ... DO	當...，執行...
WHENEVER ... DO	每當...，執行...

範例

於反轉點 1 不需要發生進給。於反轉點 2，進給會在反轉點 2 之前、距離 ii2 處進行，且震盪軸不需於反轉點等待部分進給結束。軸 Z 為震盪軸而軸 X 為進給軸。



1.震盪參數

程式碼	註解
DEF INT ii2	；為反轉區域 2 定義變數
OSP1[Z]=10 OSP2[Z]=60	；定義反轉點 1 和 2
OST1[Z]=0 OST2[Z]=0	；反轉點 1：精確停止微調
	反轉點 2：精確停止微調
FA[Z]=150 FA[X]=0.5	；震盪軸 Z 進給率，進給軸 X 進給率
OSCTRL[Z]=(2+8+16.1)	；於反轉點 2 停用震盪動作；於刪除 DTG 火花消除及逼近結束位置後；於刪除 DTG 逼近反轉位置後
OSNC[Z]=3	；火花消除衝程
OSE[Z]=70	；結束位置 = 70
ii2=2	；設定反轉點範圍
WAITP（Z）	；為 Z 軸啟用震盪

2.同步動作

程式碼	註解
WHENEVER $AA_IM[Z]< $SA_OSCILL_REVERSE_POS2[Z]DO -> -> $AA_OVR[X]=0 $AC_MARKER[0]=0	；於 MCS 中，若震盪軸 Z 的實際位置小於反轉範圍 2 的起始處，則將進給軸 X 的軸手動倍率設為 0%，並將索引 0 之位元記憶體的值設為 0。
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0	；於 MCS 中，若震盪軸 Z 的實際位置大於反轉位置 2，則將震盪軸 Z 的軸手動倍率設為 0%。
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[0]=1	；若部分進給所要前進的剩餘距離為 0，請將索引值為 0 的位元記憶體的值設為 1。
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100	；若索引值為 0 的位元記憶體等於 1，則將進給軸 X 的軸手動超調設為 0%。如此可防止過早進給（震盪軸 Z 尚未離開反轉區 2，但進給軸 X 已準備進行新的進給）。將震盪軸 Z 的軸手動超調從 0% (第二同步動作的動作) 設回 100% 來作移動。

-> 必須在單一單節中程式化。

3.啟動震盪

程式碼	註解
OSCILL[Z]=(X) POSP[X]=(5,1,1)	；啟動軸 已將震盪軸 Z 指派給 X 軸作為進給軸。 在結束位置 5 之前，軸 X 應以步進 1 移動。
M30	；程式結束

其他資訊

1. **定義震盪參數**
   震盪的參數應在含有進給與震盪軸指派，及進給定義的移動單節之前定義（請參閱「非同步震盪」）。
2. **定義運動同步動作**
   可定義下列同步條件：
   抑制進給直到震盪軸設置在反轉區域
   內（ii1、ii2）或位於反轉點（U1、U2）時為止。 於反轉點進給時
   停止震盪動作。 部分進給完成時
   重新開始震盪移動。 定義
   下個部分進給的開始。
3. **指派震盪及進給軸**以及**部分與完整進給**。

定義震盪參數

指派震盪與進給軸： OSCILL

OSCILL[<震盪軸>]=（<進給軸 1>，<進給軸 2>，<進給軸 3>）

使用「OSCILL」指令定義軸指派與震盪移動的開始。

可為一震盪軸指派至多三個進給軸。

說明

震盪開始前，必須先為軸行為定義同步條件。

定義進給： POSP

POSP[<進給軸>]=（<結束位置>，<部分長度>，<模式>）

透過「POSP」指令向控制系統做以下宣告：

- 完整進給（參考終點）
- 部分進給於反轉點或在反轉區域中的長度
- 達到終點時的部分進給回應（並參考模式）

Mode = 0	最後兩次部分進給距離目的地的剩餘距離，會被等分為兩個相同的步進（預設設定）。
Mode = 1	所有部分進給的尺寸相同 會從完整的進給中計算出。

定義動作－同步動作

一般震盪會使用下列之同步－動作動作。

提供個別作業的解決方案，可作為建立使用者專屬之震盪移動的模組

說明

個別案例中，可以不同的方式程式設計同步條件。

關鍵字

WHEN ... DO ...	當...，執行...
WHENEVER ... DO	每當...，執行...

功能

您可使用下列詳述之語言資源建置以下功能：

1. 於反轉點的進給。
2. 於反轉區域的進給
3. 於兩反轉點進給
4. 於反轉點停止震盪移動。

5. 重新開始震盪移動。
6. 切勿過早開始部分進給。

為在此舉出之同步動作範例，提出下列假設：

- 反轉點 1 < 反轉點 2
- Z = 震盪軸
- X = 進給軸

說明

如需詳細資訊，請參閱「動作－同步動作」一節。

指派震盪及進給軸以及部分與完整進給

於反轉點範圍內進給

進給動作必須在抵達反轉點之前，於反轉區域內開始。

這些同步動作在震盪軸到達反轉區域內之前將禁止進給移動。

下列指令受上述假設約束：

反轉範圍 1：

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X] = 0
```

於 MCS 中，每當震盪軸的實際位置大於反轉範圍 1 起始處，則將進給軸的軸手動倍率設為 0%。

反轉範圍 2：

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X] = 0
```

於 MCS 中，每當震盪軸的實際位置小於反轉範圍 2 起始處，則將進給軸的軸手動倍率設為 0%。

於反轉點的進給

只要震盪軸未到達反轉點，進給軸便不會移動。

下列指令受上述假設約束：

反轉範圍 1：

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

於 MCS 中，每當震盪軸 Z 的實際位置大於或小於反轉點 1 的位置，則將進給軸 X 的軸手動倍率設為 0%，並將震盪軸 Z 的軸手動倍率設為 100%。

反轉範圍 2：

針對反轉點 2：

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

於 MCS 中，每當震盪軸 Z 的實際位置大於或小於反轉點 2 的位置，則將進給軸 X 的軸手動倍率設為 0%，並將震盪軸 Z 的軸手動倍率設為 100%。

於反轉點停止震盪移動

震盪軸於反轉點處停止，而進給動作同時開始。 進給移動完成時，震盪動作仍繼續。

同時，若同步動作因目前仍有效的上一個同步動作而中斷，則此同步動作可用來啟動進給移動。

下列指令受上述假設約束：

反轉範圍 1：

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

於 MCS 中，每當震盪軸 Z 的實際位置與反位置 1 相同，則將震盪軸的軸手動倍率設為0%，並將進給軸的軸手動倍率設為 100%。

反轉範圍 2：

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

於 MCS 中，每當震盪軸 Z 的實際位置與反位置 2 相同，則將震盪軸 X 的軸手動倍率設為 0%，並將進給軸的軸手動倍率設為 100%。

反轉點的線上評估

若主要執行變數之比較右側為$$，則兩變數會在 IPO 循環中持續相互評估與比較。

說明

請參考「動作同步動作」一節，取得更多資訊。

震盪移動重新開始

本同步動作的目的是在工件進給移動完成時，使進給軸仍能持續移動。

下列指令受上述假設約束：

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]= 100
```

於 WCS 中，每當進給軸 X 之部分進給的剩餘距離等於零，則將震盪軸的軸手動倍率設為 100%。

下一個部分進給

進給完成時，必須禁止下個部分進給過早開始。

通道專屬標記（$AC_MARKER[Index]）可用來達成此目的。 它於部分進給結束時（部分剩餘距離 ≡ 0）啟用，且於軸離開反轉區域時被刪除。 同步動作會預防下個進給移動發生。

以這些已知假設為基礎，下列指令將適用於反轉點 1：

1. 設定標記：

```
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[1]=1
```

於 WCS 中，每當進給軸 X 之部分進給的剩餘距離等於零，則將含索引 1 的位元記憶體設為 1。

2. 刪除標記

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1] = 0
```

於 MCS 中，每當若震盪軸 Z 的實際位置大於或小於反轉點 1 的位置，則將位元記憶體 1 設為 0。

3. 禁止進給

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

每當位元記憶體 1 相同，則將進給軸 X 的軸手動倍率設為 0%。

沖孔與切片 16

16.1 啟用 / 停用

16.1.1 啟用 / 停用沖孔與切片（SPOF、SON、PON、SONS、PONS、PDELAYON、PDELAYOF、PUNCHACC）

啟用/停用沖孔與切片

PON 與 SON 可用於啟動沖孔與切片函數。SPOF 結束所有沖孔及切片專屬函數。模態指令 PON 和 SON 相互排斥，意即 PON 會停用 SON，反之亦然。

以前導程式進行沖孔/切片

SONS 及 PONS 函數亦可啟動沖孔或切片功能。

與 SON/PON 相反（於補插層級進行衝擊控制），這些函數於伺服層級進行衝擊初始動作的訊號相關控制。這啟用了更高的衝程頻率，並因此增加了要達成的沖孔容量。

當訊號在前導程式中被評估，所有造成函數切片或沖孔軸改變位置的函數（例如，手動手輪移動，透過 PLC 更改至框架，量測函數）皆禁止使用。

含延遲的沖孔

PDELAYON 會延遲沖孔衝程的輸出。模態有效指令具有準備的功能，因此通常放在 PON 之前。執行 PDELAYOF 後會繼續進行正常沖孔。

說明

延遲時間係設定在設定資料 SD42400 $SC_PUNCH_DWELLTIME 中。

路徑相依的加速度

PUNCHACC 可用來指定加速度特定，依照孔洞間距來定義不同的加速度率。

第二沖孔介面

需要使用第二沖孔介面的加工（第二沖孔元件或可相比的媒介），可選擇切換至控制面版上的第二對快速數位輸入與輸出（I/O 組）。兩種介面均提供完整的沖孔 / 切片功能。SPIF1 與 SPIF2 指令可用來在第一和第二沖孔介面之間切換。

說明

需求：第二 I/O 組必須在機械資料中定義給沖孔功能（請參考機台製造商規格'）。

語法

```
PON G...X...Y...Z...
SON G...X...Y...Z...
SONS G...X...Y...Z...
PONS G...X...Y...Z...
PDELAYON
PDELAYOF
PUNCHACC（<S 最小>，<A 最小>，<S 最大>，<A 最大>）
SPIF1/SPIF2
SPOF
```

含義

PON：	啟動沖孔。	
SON：	啟用切片	
PONS：	使用前導程式啟動沖孔。	
SONS：	使用前導程式啟動切片。	
SPOF：	停用沖孔/切片 。	
PDELAYON：	使用延遲啟動沖孔。	
PDELAYOF：	使用延遲停用沖孔。	
PUNCHACC：	啟動移動相依加速度。 參數：	
	<S 最小>	最小孔洞間距
	<A 最小>	初始加速度 <A 最小>可大於<A 最大>。
	<S 最大>	最大孔洞間距
	<A 最大>	最終加速度 <A 最大>可大於<A 最小>。

SPIF1：	啟動**第一**沖孔介面。 衝程係使用第一組快速 I/O 來控制。
SPIF2：	啟動**第二**沖孔介面。 衝程係使用第二組快速 I/O 來控制。
	附註： 重置後或控制系統開機後，第一沖孔介面便永遠為啟用中。如果只使用一個沖孔介面，則不必將其程式設計。

範例

範例 1：啟用切片

程式碼	註解
...	
N70 X50 SPOF	；無初始沖孔定位。
N80 X100 SON	；啟用切片；在動作之前（X=50）以及在完成程式設計之移動時（X=100），啟動衝程。
...	

範例 2：含延遲的沖孔

程式碼	註解
...	
N170 PDELAYON X100 SPOF	；不含沖孔啟動的位置，啟用延遲沖孔啟動。
N180 X800 PON	；啟動沖孔。抵達結束位置時，沖孔衝程會延遲輸出。
N190 PDELAYOF X700	；以延遲停用沖孔，在程式設計移動完成時進行一般沖孔啟動。
...	

範例 3：以兩個沖孔介面進行沖孔

程式碼	註解
...	
N170 SPIF1 X100 PON	；單節結束時，在第一快速輸出上啟動衝程。在第一輸入上監控「衝程啟用」訊號。
N180 X800 SPIF2	；在第二高速輸出上啟動第二衝程。在第二輸入上監控「衝程啟用」訊號。
N190 SPIF1 X700	；使用第一介面控制其他所有衝程。
...	

其他資訊

以前導程式進行冲孔及切片 (PONS/SONS)

不能同時在同一通道中以前導程式進行冲孔及切片。PONS 或 SONS 一次只能在一個通道中啟用。

路徑相依加速度 (PUNCHACC)

範例：

```
PUNCHACC(2,50,10,100)
```

孔洞間距離小於 2 毫米：

軸會以最高加速度的 50%的速率加速。

孔洞間距離介於 2 毫米至 10 毫米間：

加速度會增加到 100%，與間距成比例。

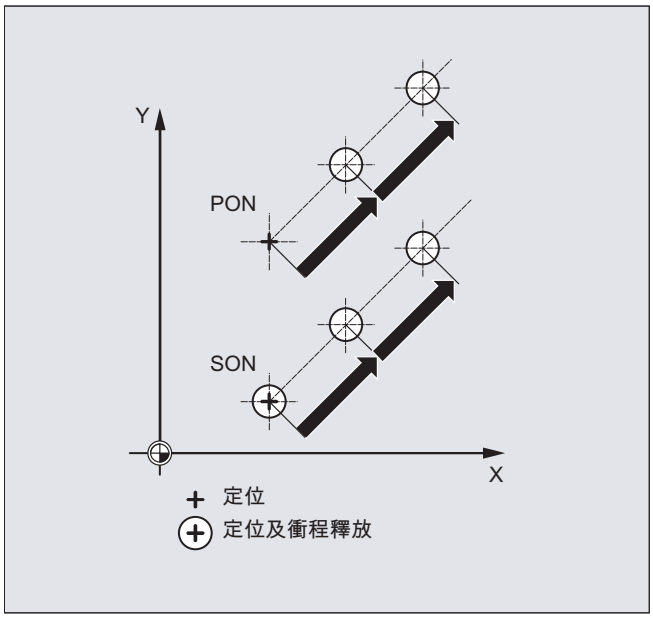
孔洞間距離大於 10 毫米：

以 100%之加速度移動。

初次衝程之初始化

啟用本函數後發生首次衝擊的時間點取決於所選的是切片或冲孔：

- PON/PONS：
  - 所有衝程（包括啟用後位於第一個單節的衝程）皆在單節結尾時執行。
- SON/SONS：
  - 在單節開始時執行的切片函數啟用後之首次衝程。
  - 後續各衝程皆在單節結尾時啟動。



現地沖孔及切片

衝程僅在若單節中包含沖孔或切片軸（位於有效平面的軸）的移動資訊時啟動。

然而，若要在同一位置啟動衝程，可程式設計沖孔 / 切片軸其中一個的移動路徑為 0。

以可旋轉工具加工

說明

若您想將可旋轉工具定位在已程式設計的路徑之切線上，請使用切線控制函數。

M 指令的使用

如同較早的版本，巨集技術允許特殊 M 功能的使用，不必採用語言指令（一致性）。M 功能以及對等的語言指令，亦用於較早的系統如下：

```
M20，M23  ≙  SPOF
M22       ≙  SON
M25       ≙  PON
M26       ≙  PDELAYON
```

巨集檔案範例：

程式碼	註解
DEFINE M25 AS PON	；沖孔開啟

程式碼	註解
DEFINE M125 AS PONS	；含前導的沖孔開啟
DEFINE M22 AS SON	；切片開啟
DEFINE M122 AS SONS	；含前導的切片開啟
DEFINE M26 AS PDELAYON	；含延遲的沖孔開啟
DEFINE M20 AS SPOF	；沖孔、切片關閉
DEFINE M23 AS SPOF	；沖孔、切片關閉

程式設計範例：

程式碼	註解
...	
N100 X100 M20	；無初始沖孔定位。
N110 X120 M22	；啟動切片，在動作之前和之後的初始衝程。
N120 X150 Y150 M25	；啟動沖孔，在動作結尾初始衝程。
...	

16.2 自動路徑分段

分割為路徑區段

沖孔或切片為啟動時，SPP 及 SPN 皆會將為路徑軸程式設計的總移動區段，分割成數個等長的路徑區段（等距路徑區段）。每個路徑區段內部會與一單節對應。

衝程數量

沖孔時，會在第一個路徑區段的終點進行首次衝擊；不過在切片時，則會在第一個路徑區段的起點進行首次衝擊。因此，完整的移動區段會得到以下數量：

沖孔：衝程數量=路徑區段數量

切片：衝程數量=路徑區段數量+1

輔助功能

輔助功能在產生的第一個單節中執行。

語法

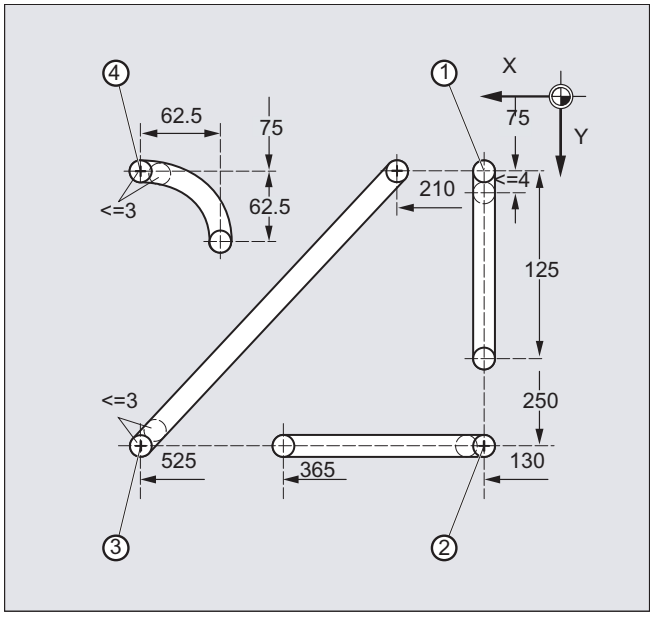
```
SPP=
SPN=
```

含義

SPP：	路徑區段大小（衝程間的最大距離）；模態
SPN：	每一單節之路徑區段數量；模態有效

範例 1

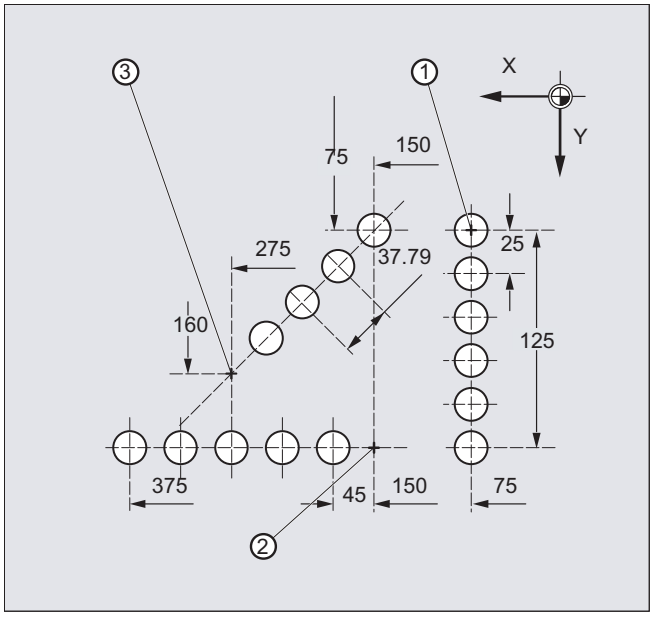
應自動將程式設計切片區段分割成路徑區段。



程式碼	註解
N100 G90 X130 Y75 F60 SPOF	；定位於起點 1
N110 G91 Y125 SPP=4 SON	；切片開啟；自動路徑分段之最長路徑區段長度：4 mm
N120 G90 Y250 SPOF	；切片關閉，定位至起點 2
N130 X365 SON	；切片開啟；自動路徑分段之最長路徑區段長度：4 mm
N140 X525 SPOF	；切片關閉，定位至起點 3
N150 X210 Y75 SPP=3 SON	；切片開啟；自動路徑分段之最長路徑區段長度：3 毫米
N160 X525 SPOF	；切片關閉，定位至起點 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	；切片開啟；自動路徑分段之最長路徑區段長度：3 毫米
N180 G00 G90 Y300 SPOF	；切片關閉

範例 2

應為各組孔洞進行自動路徑分段。並為分段指定最長路徑區段長度（SPP 值）。



程式碼	註解
N100 G90 X75 Y75 F60 PON	；定位至起點 1； 沖孔單一孔洞
N110 G91 Y125 SPP=25	；自動路徑分段之最長路徑區段長度 ：25 mm
N120 G90 X150 SPOF	；沖孔關閉，定位至 起點 2
N130 X375 SPP=45 PON	；沖孔開啟；自動路徑分段之最長路徑區段長度 ：45 mm
N140 X275 Y160 SPOF	；沖孔關閉，定位至 起點 3
N150 X150 Y75 SPP=40 PON	；沖孔開啟，相對於使用已程式設計的路徑區段長度 40 毫米，改用計算出的路徑區段長度 37.79 毫米。
N160 G00 Y300 SPOF	；沖孔關閉；定位

16.2.1 路徑軸的路徑分割

SPP 路徑區段長度

SPP 用來指定衝程間最大間距，也就是其中跨移距離總長所分割的路徑區段之最長長度。以 SPOF 或 SPP=0 停用指令。

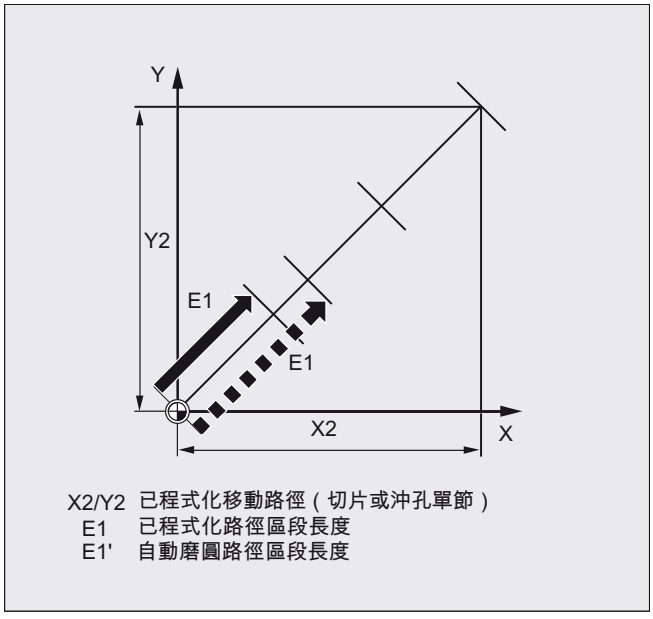
範例：

```
N10 SON X0 Y0
N20 SPP=2 X10
```

總長 10 毫米的移跨距離將分割成五個路徑區段，每段 2 毫米（SPP=2）。

說明

經由 SPP 所產生的路徑區段總是等距，即所有區段的長度都相同。也就是說，只有在總跨移距離的商數（quotient）及 SPP 值是整數時，所程式設計的路徑區段規格（SPP 設定）才有效如果不是如此，路徑區段規格便會為了產生整數商數，在內部減少。



範例：

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

總跨移距離為 15 毫米，而路徑區段長度為 3.5 毫米時，其商數不是整數值（4.28）。在這樣的情況下，SPP 值被降至下個可能的整數商數。此範例的結果會是路徑區段長度為 3 毫米。

SPN 路徑區段數量

SPN 定義由跨移距離總長所產生的路徑區段數量。區段長度自動計算。因為 SPN 為非模態，所以沖孔或沖壓必須事先分別使用 PON 或 SON 啟動。

16.2.2 單一軸的路徑分割

如果除路徑軸之外，單一軸也被定義為沖孔/切片軸，則可為單一軸啟動自動路徑分割函數。

SPP 的單一軸行為

已程式設計的路徑區段長度 (SPP) 基本上是指路徑軸。因此，軸單節中除了單軸移動及 SPP 值之外，沒有路徑軸被程式設計，則 SPP 值將被忽略。

若在單節中程式設計個別軸與路徑軸，則單一軸的行為將依照相關機械資料而定。

1. 預設設定
   單一軸所移跨的路徑，均勻分布在由 SPP 所產生的中間單節上。

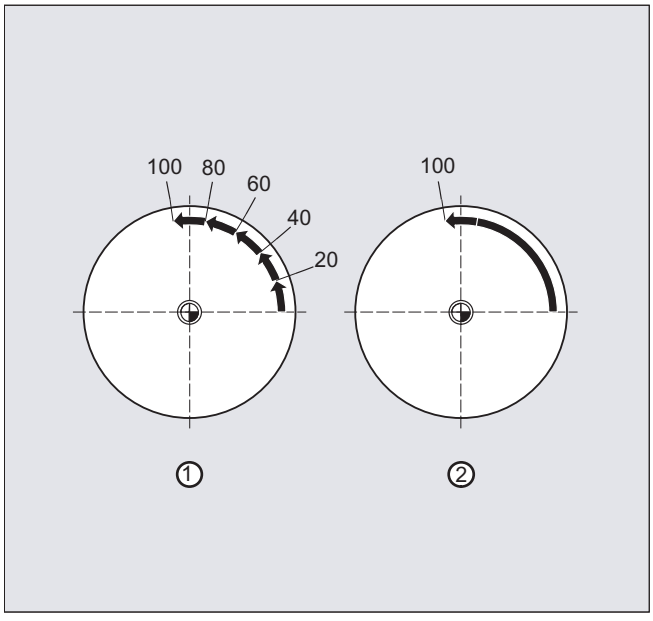
範例：

```
N10 G1 SON X10 A0

N20 SPP=3 X25 A100
```

衝程間距為 3 毫米，X 軸（路徑軸）跨越距離總長為 15 毫米，因此產生 5 個單節。

A 軸便在每個單節中，以 20 度旋轉。



1. 無路徑分割的單一軸
   單一軸在產生的第一個單節之完整長度上移跨。
2. 不同的路徑分段
   單一軸的行為依據路徑軸的插補而定：

- 圓弧插補：路徑分割
- 線性插補：無路徑分段

SPN 的行為

如未同時程式設計路徑軸，亦可套用已程式設計的路徑分段數量。

先決條件：單一軸定義為冲孔-切片軸。

研磨 17

17.1 啟動 / 停用研磨專屬刀具監控（TMON、TMOF）

使用預先定義的程序 TMON(...) 和 TMOF(...)，研磨專屬刀具監控將啟動或停用（幾何與速度監控）。

需求條件

需設定刀具專屬參數 $TC_TPG1 至 $TC_TPG9。

語法

```
TMON(<TNo>)
...
TMOF(<TNo>)
```

含義

TMON(...)：	啟動研磨專屬刀具監控 指令必須程式設計至研磨專屬刀具監控要啟用的通道中。
TMOF(...)：	停用研磨專屬刀具監控 指令必須程式設計至研磨專屬刀具監控要停用的通道中。
<TNo>：	T 編號 **附註：** 只有在要開啟和關閉的是停用的研磨輪而不是目前使用中的研磨輪的監控時，才需要此設定。
TMOF(0)：	停用監控所有刀具

附加功能 18

18.1 軸函數（AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL）

例如，若在軸名稱不明時，可使用「AXNAME」產生一般有效的循環。

使用「AX」可間接程式設計幾何及同步軸。 軸識別碼存在 AXIS 類型的變數中，或由「AXNAME」或「SPI」等指令提供。

若為主軸（例如同步主軸）程式設計軸函數，可使用「SPI」。

使用「AXTOSPI」可將軸識別碼轉換成主軸索引（「SPI」之反函數）。

使用「AXSTRING」可將軸識別碼（資料類型 AXIS）轉換成字串（「AXNAME」之反函數）。

在通用循環中使用「ISAXIS」，可確保特定幾何軸存在，而不會發出錯誤訊息取消後續一切 $P_AXNX 呼叫。

使用「MODAXVAL」可判定模數旋轉軸之模數位置。

語法

```
AXNAME("string")
AX[AXNAME("string")]
SPI(n)

AXTOSPI(A) 或 AXTOSPI(B) 或 AXTOSPI(C)
AXSTRING( SPI(n) )
ISAXIS(<幾何軸編號>)
<模數位置>=MODAXVAL(<軸>,<軸位置>)
```

含義

AXNAME：	將輸入字串轉換成軸辨識碼；輸入字串需含有效的軸名稱。
AX：	變數軸識別碼
SPI：	將主軸編號轉換成軸辨識碼；傳輸參數需含有效的主軸編號。
n：	主軸編號
AXTOSPI：	將軸辨識碼轉換成整數主軸索引。「AXTOSPI」是「SPI」的反函數。
X、Y、Z：	將 AXIS 類型的軸辨識碼當成變數或常數

AXSTRING：	字串會隨相關主軸編號輸出。
ISAXIS：	檢查指定的幾何軸是否存在。
MODAXVAL：	判定模數旋轉軸之模數位置；其符合參考參數化模數範圍的模態休止點（在預設設定中，為 0 到 360 度；可利用 MD30340 MODULO_RANGE_START 及 MD30330 $MA_MODULO_RANGE 變更模數範圍的起點及大小。）

說明

SPI 副檔名

軸函數 SPI（n）亦可用來讀寫框架元件。 這代表可使用例如 $P_PFRAME[SPI(1),TR]=2.22 語法寫入框架。

可另外使用位址 AX[SPI（1）]=<軸位置>程式設計軸位置來移動軸。 前提是主軸需在定位或軸模式之下。

範例

範例 1： AXNAME、AX、ISAXIS

程式碼	註解
OVRA[AXNAME（"Transverse axis"）]=10	；手動調整橫向軸
AX[AXNAME（"Transverse axis"）]=50.2	；橫向軸結束位置
OVRA[SPI（1）]=70	；手動調整主軸 1
AX[SPI（1）]=180	；主軸 1 結束位置
IF ISAXIS(1) == FALSE GOTOF CONTINUE	；可使用橫座標？
AX[$P_AXN1]=100	；移動橫座標
CONTINUE：	

範例 2：AXSTRING

使用 AXSTRING[SPI(n)] 進行程式設計時，指派給主軸的軸索引值將不再以主軸編號形式輸出，而是輸出字串「Sn」。

程式碼	註解
AXSTRING[SPI（2）]	；輸出字串「S2」。

範例 3：MODAXVAL

模數旋轉軸 A 的模數位置待判定。

計算的起始值為軸位置 372.55。

參數化模數範圍為 0 到 360 度：

MD30340 MODULO_RANGE_START = 0

MD30330 $MA_MODULO_RANGE = 360

程式碼	註解
R10=MODAXVAL（A，372.55）	；計算出模數位置 R10 = 12.55。

範例 4：MODAXVAL

若已程式設計的軸辨識碼並未參考模數旋轉軸，則會傳回原先欲轉換之值（<軸位置>）。

程式碼	註解
R11=MODAXVAL（X，372.55）	；X 為線性軸；R11 = 372.55。

18.2 可更換的幾何軸（GEOAX）

「可切換幾何軸」函數可使用其他通道軸取代由機械資料設定的幾何軸。

語法

```
GEOAX（<n>，<通道軸>，<n>，<通道軸>，<n>，<通道軸>）
GEOAX（）
```

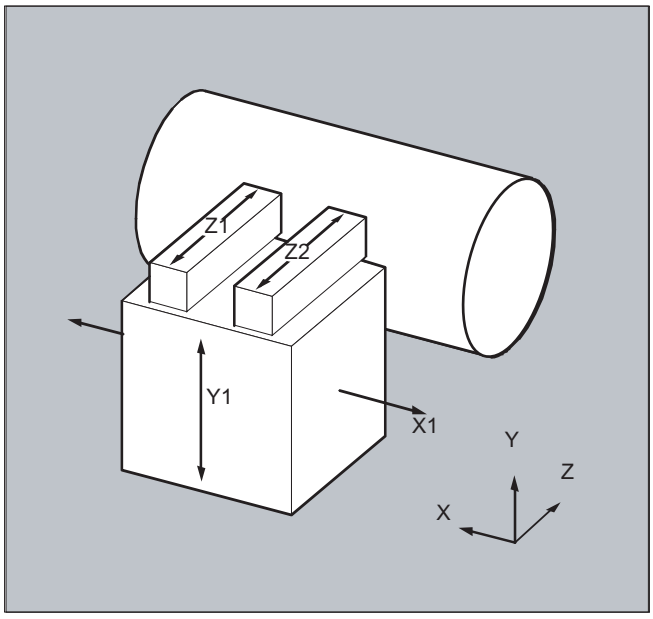
含義

GEOAX（...）	切換幾何軸的函數。 **附註：** GEOAX()（不含參數規格）會再次啟動在機械資料參數化的幾何軸基本組態。
<n>	由指定通道軸取代的幾何軸數量。 值域：0, 1, 2, 3 **附註：** 0: 指定通道軸由幾何軸群組移除，並未取代 1: 1. 幾何軸 ≙ 工件座標系統 (WCS) 座標軸 X（橫座標） 2: 2. 幾何軸 ≙ 工件座標系統 (WCS) 座標軸 Y（縱座標） 3: 3. 幾何軸 ≙ 工件座標系統 (WCS) 座標軸 Z（備選座標）
<通道軸>	新增至幾何軸群組的通道軸名稱

範例

範例 1： 在兩軸間切換決定幾何軸

可使用通道軸 X1、Y1、Z1、Z2 來移動刀具滑座：



幾何軸已設定，故開啟電源後，首先 Z1 便為有效第三幾何軸，幾何軸名稱為「Z」，並與 X1 及 Y1 共同構成幾何軸群組。

現在在工件程式中會使用 Z1 及 Z2，並在兩者間切換作為幾何軸 Z：

程式碼	註解
...	
N100 GEOAX(3, Z2)	;通道軸 Z2 扮演第三幾何軸 (Z)。
N110 G1 ...	
N120 GEOAX(3, Z1)	;通道軸 Z1 扮演第三幾何軸 (Z)。
...	

範例 2： 變更六通道軸之幾何軸

一個機台有六個通道軸，名稱分別為 XX、YY、ZZ、U、V、W。

以機械資料進行座標軸設定的基本設定為：

通道軸 XX = 第一幾何軸（X 軸）

通道軸 YY = 第二幾何軸（Y 軸）

通道軸 ZZ = 第三幾何軸（Z 軸）

程式碼	註解
N10 GEOAX()	;幾何軸的基本組態生效。
N20 G0 X0 Y0 Z0 U0 V0 W0	;所有軸快速移動至位置 0。
N30 GEOAX(1, U, 2, V, 3, W)	;通道軸 U 成為第一 (X)，V 為第二 (Y)
	;而 W 為第三幾何軸 (Z)。

程式碼	註解
N40 GEOAX(1,XX,3,ZZ)	;通道軸 XX 成為第一 (X)，ZZ 為第三
	;幾何軸 (Z)。 通道軸 V 仍為第二
	;幾何軸 (Y)。
N50 G17 G2 X20 I10 F1000	;X/Y 平面中的完整圓。 通道軸
	;XX 及 V 移動。
N60 GEOAX(2,W)	;通道軸 W 成為第二幾何軸 (Y)。
N80 G17 G2 X20 I10 F1000	;X/Y 平面中的完整圓。 通道軸
	;XX 及 W 移動。
N90 GEOAX()	;重置至初始狀態。
N100 GEOAX(1,U,2,V,3,W)	;通道軸 U 成為第一 (X)，V 為第二
	;(Y) 而 W 為第三幾何軸 (Z)。
N110 G1 X10 Y10 Z10 XX=25	;通道軸 U、V、W 各自移動至
	;位置 10。XX 作為特殊軸移動至位置 25。
N120 GEOAX(0,V)	;自幾何軸群組中移除 V。
	;U 及 W 仍為第一 (X) 和第三
	;幾何軸 (Z)。
	;仍未指派第二幾何軸 (Y)。
N130 GEOAX(1,U,2,V,3,W)	;通道軸 U 仍為第一 (X)，V 為
	;第二 (Y)，W 為第三幾何軸 (Z)。
N140 GEOAX(3,V)	;V 為第三幾何軸 (Z)，其中 W
	;遭到覆寫，因此移除離開幾何
	;軸群組。 第二幾何軸 (Y)
	;仍未指派。

機械參數

軸設定

指派幾何、特殊和機械軸至通道軸：

- 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

重置行為

變更幾何軸指派的重置行為：

- MD20110 $MC_RESET_MODE_MASK，位元 12
- MD20118 $MC_GEOAX_CHANGE_RESET

NC 啟動行為

- MD20112 $MC_START_MODE_MASK，位元 12

傳送通知給 PLC 使用者程式

M 指令的參數化選項，於變更幾何軸時在 NC/PLC 介面輸出。

- MD22532 $MC_GEOAX_CHANGE_M_CODE

補充條件

沒有更換幾何軸

- 如果下列函數之一生效，就不可能進行幾何軸更換：
  - 轉換
  - 曲線插補
  - 刀具半徑補正
  - 刀具微調偏移
- 幾何軸及其他通道軸的名稱相同。
- 參與幾何軸更換的其中一個軸參與動作，超越單節限制，例如全單節定位軸或軸耦合的跟隨軸。

旋轉軸

旋轉軸無法編程設定為幾何軸。

更換後之軸狀態

藉由通道軸名稱變更操作後，可在幾何軸群組中將已更換的軸程式設計為輔助軸。

框架、保護區、工作區限制

在變更幾何軸後，會刪除所有框架、保護區及工作區限制。

極座標

以 GEOAX 更換幾何軸會使用 G17-G19 將類比設為層級變更，模態極座標的值會則為 0。

DRF、WO

變更後，合適的手輪偏移（DRF）或外部工件偏移量（ZO）仍然有效。

幾何軸之基本設定

GEOAX（）指令會呼叫幾何軸群組的基本設定。

系統在 POWER ON 後，以及變更為“參考點逼近”模式時，會自動變回基本設定。

刀長補正

變更操作後，啟用刀長補正也生效。 然而，新增或被取代位置的幾何軸，會被認定尚未移動補正距離。 該幾何軸的第一個動作指令，最終的移動距離是總刀長補正及程式設計移動距離的總和。

在替換操作後，幾何軸仍在軸群組中保有其位置，也保有其刀長補正的狀態。

啟用轉換的幾何軸設定

- 透過轉換機械參數針對生效轉換參數化的幾何軸組態，無法使用「可切換幾何軸」函數變更。
- 不同資料集必須在不同幾何軸設定的轉換機械參數進行參數化，以便進行轉換。
- 啟動轉換會刪除使用 GEOAX 變更的幾何軸設定。
- 至於在幾何軸方面，生效轉換的轉換專屬幾何軸參數化，其優先順序高於變更幾何軸的相關參數化。
  範例： 轉換生效。 依據機械參數，應在通道重置時保留轉換。 同時幾何軸的基本組態應在通道重置時還原。 已針對轉換指定的幾何軸組態將保留。
- 若轉換關閉，幾何軸組態的參數化基本設定將再次生效。

寸動進給模式；REF 機台功能

切換為寸動進給模式時，REF 機台功能（原點復歸）、機械參數中參數化的幾何軸組態將會生效

18.3 軸容器（AXCTSWE、AXCTSWED、AXCTSWEC）

「AXCTSWE」或「AXCTSWED」指令可啟用指定軸容器的旋轉。

透過「AXCTSWEC」指令可取消已啟用的軸容器旋轉。

語法

```
AXCTSWE(<ID>)
AXCTSWED(<ID>)
AXCTSWEC(<ID>)
```

含義

AXCTSWE：	啟用軸容器旋轉 使用「AXCTSWE」不會停止程式處理。 一旦軸容器相關的所有通道啟用後，即開始旋轉。	
AXCTSWED：	使軸容器旋轉，不考慮與軸容器相關的其他通道 **附註** ● 變化指令以簡化工件程式或同步動作的調試。 ● 至於與軸容器相關之其他通道的行為，可透過以下指令設定：MD12760 $MN_ AXCT_FUNCTION_MASK，位元 0	
AXCTSWEC：	取消啟用的軸容器旋轉 **附註** 只有在旋轉尚未開始前，才能取消啟用軸容器旋轉： $AN_AXCTSWA[<軸容器>] == 0 關於系統變數，請參閱 "軸容器（AXCTSWE、AXCTSWED、AXCTSWEC）(頁 699)"	
<ID>：	軸容器或容器軸的識別碼：	
	CT<編號>：	軸容器的預設識別碼： MD12750 $MN_AXCT_NAME_TAB 範例：「CT1」
	<容器>：	軸容器的使用者專屬識別碼： MD12750 $MN_AXCT_NAME_TAB 範例：「CONTAINER_1」
	<軸>：	通道中已知容器軸的識別碼

說明

增量

透過設定資料設定軸容器旋轉增量：

SD41700 $SN_AXCT_SWWIDTH

其他資訊

診斷

可透過以下系統變數讀取軸容器的目前狀態：

系統變數	類型	說明
$AC_AXCTSWA[<名稱>]	BOOL	軸容器的通道專屬狀態
$AN_AXCTSWA[<軸容器>]	BOOL	軸容器的 NCU 專屬狀態
$AN_AXCTSWE[<軸容器>]	INT	軸容器旋轉的插槽專屬狀態 系統變數提供軸容器插槽 **bitwise** 的狀態。 每個位元對應至一個插槽。
$AN_AXCTAS[<軸容器>]	INT	剛切換經過的軸容器位置（插槽）編號。

隱性 GET / GETD 的軸容器旋轉

使用「AXCTSWE」指令藉由未指定之「GET/GETD」帶入通道的所有通道容器軸，可利用下列機械資料進行設定。 唯有容器旋轉後，才能再次更換軸。

MD10722 $MN_AXCHANGE_MASK，位元 1 = 1

說明

處於「主執行軸」狀態的軸（如 PLC 軸）**不得**使用未指定之「GET/GETD」進行軸容器旋轉，為了進行軸容器旋轉，此軸必須離開該狀態。

18.4 等候有效的軸位置 (WAITENC)

利用語言命令「WAITENC」，NC 程式可等候，直到已同步或已恢復的軸位置，可供參數值設為 MD34800 $MA_WAIT_ENC_VALID = 1 的軸使用。

可藉由（舉例而言）啟動 ASUB 或將操作模式變更為寸動進給 (JOG)，中斷等候狀態。 程式繼續執行時，會重回等候狀態。

說明

使用者介面中，「等候量測系統」保留狀態會顯示等候狀態。

語法

「WAITENC」可編寫於任何 NC 程式的程式區段中。

程式碼必須編寫於專屬的單節中：

```
...
WAITENC
...
```

範例

「WAITENC」可用於（舉例而言）以事件控制的使用者程式中 (.../_N_CMA_DIR/_N_PROG_EVENT_SPF)，如以下應用範例所示。

應用範例： 關機 (POWER OFF) 後利用方向轉換退出刀具

電力失效造成利用刀具方向的加工發生中斷。
再次開機時，會呼叫以事件控制的使用者程式.../_N_CMA_DIR/_N_PROG_EVENT_SPF。

在以事件控制的使用者程式中，系統利用「WAITENC」等候已同步或已恢復的軸位置，以便隨後能夠計算框架，使工件座標系統 (Work) 對齊刀具方向。

程式碼	註解
...	
IF $P_PROG_EVENT == 4	；預備。
IF $P_TRAFO <> 0	；轉換已選取。
WAITENC	；等候方向軸的有效軸位置。
TOROTZ	；將工件座標系統 (WCS) 的 Z 軸朝刀具軸轉。
ENDIF	
M17	
ENDIF	

程式碼	註解
. . .	

接著可以在寸動進給 (JOG) 模式中，利用朝向刀具軸的回退移動，將刀具回退。

18.5 可程式參數集轉換 (SCPARA)

利用 SCPARA 指令可提出將軸轉換至特定參數集之要求。

說明

螺紋切削時，無法進行參數集轉換

螺紋切削 G33 和攻牙 G331/G332 時，由控制選定參數集，因此無法變更。

停用參數集轉換

透過 NC/PLC 介面，亦可提出參數集轉換之要求。 為了避免轉換衝突，可透過 NC/PLC 介面停用 NC 參數集轉換 (SCPARA)：

DB31, ... DBX9.3（以 NC 停用的參數集規格）

說明

若以 SCPARA 要求參數集轉換，同時透過 NC/PLC 介面停用參數集轉換，轉換要求將被拒絕，且不會產生錯誤訊息。

語法

```
SCPARA[<軸>]=<值>
```

含義

SCPARA：	指令： 變更參數集	
<軸>：	軸識別碼（通道軸）	
	類型：	AXIS
<值>：	參數集編號： 1, 2, 3, ... 最大參數集編號	

範例

程式碼	註解
...	
N110 SCPARA[X]= 3	；選擇： 軸 X，第三參數集
...	

其他資訊

啟用參數集轉換

軸的參數集轉換必須明確啟用：

MD35590 $MA_PARAMSET_CHANGE_ENABLE[<軸>]

讀取參數集編號

可透過系統變數 $AA_SCPAR 讀取選定的參數集編號（指定的參數集）。

參考資料

有關參數集的詳細資訊，請參閱：

功能手冊，基本功能；章節〈速率，設定點 / 實際值系統，閉迴路控制 (G2)〉>〈閉迴路控制〉>〈位置控制器的參數集〉

18.6 檢查顯示 NC 語言之範圍（STRINGIS）

使用「STRINGIS (...)」功能，即可檢查所指定的字串，是否可做為實際語言範圍內的 NC 程式元素。

定義

```
INT STRINGIS (STRING <名稱>)
```

語法

```
STRINGIS (<名稱>)
```

含義

STRINGIS：	有傳回值的函數
<名稱>：	待檢查的 NC 程式設計語言元素名稱。
傳回值：	傳回值的格式為 yxx（十進位）。

NC 程式設計語言元素

以下為可檢查的 NC 程式設計語言元素：

- 全部現存 G 群組的 G 指令，例如「G0」、「INVCW」、「POLY」、「ROT」、「KONT」、「SOFT」、「CUT2D」、「CDON」、「RMBBL」、「SPATH」
- DIN 或 NC 位址，如「ADIS」、「RNDM」、「SPN」、「SR」、「MEAS」
- 功能，例如「TANG (...)」或「GETMDACT」
- 程序，例如「SBLOF」。
- 關鍵字，例如「ACN」、「DEFINE」或「SETMS」
- 系統資料，例如機械資料 $M... 、設定資料 $S... 或選項資料 $O...
- 系統變數 $A...、$V...、$P...
- 算術參數 R...
- 已啟用的循環的循環名稱
- GUD 和 LUD 變數
- 巨集名稱
- 標籤名稱

傳回值

僅採前三個十進位數字。傳回值的格式為 yxx，其中 y = 基本資訊，xx = 詳細資訊。

傳回值	含義	
000	'name' 字串在本系統中是未知的 1)	
100	「name」字串是 NC 程式語言的元素，但目前無法用於程式設計（選項 / 功能停用）	
2xx	'name' 字串是 NC 程式設計語言的可程式設計元素 （選項 / 功能啟用）。詳細資訊 xx 包含元素類型的其他資訊：	
	xx	**含義**
	01	DIN 位址或 NC 位址 2)
	02	G 指令（例如 G04、INVCW）
	03	有傳回值的函數
	04	沒有傳回值的函數
	05	關鍵字，例如 DEFINE
	06	機械資料 ($M...)、設定資料 ($S...) 或選項資料 ($O...)
	07	系統參數，例如系統變數 ($...) 或算術參數 (R...)
	08	循環（循環必須載入到 NC 中，且循環程式必須生效 3)）
	09	GUD 變數（必須在定義檔中定義 GUD 變數，且 GUD 變數已啟用）
	10	巨集名稱（必須在巨集定義檔中定義巨集，且巨集已啟用）4)
	11	實際工件程式的 LUD 變數
	12	ISO G 指令（ISO 語言模式必須生效）
400	'name' 字串為一個非記為 xx == 01 或 xx == 10 且非 G 或 R 2) 的 NC 位址。	
y00	無法作特定的指派	

1) 視控制器而定，某些情況下，僅部份的西門子 NC 語言指令為系統所知，例如 SINUMERIK 802D sl。若為此類控制器，且字串主要是西門子 NC 語言時，會傳回 0。此行為可利用 MD10711 $MN_NC_LANGUAGE_CONFIGURATION 改變。MD10711 = 1 時，西門子 NC 語言指令一定傳回 100。

2) NC 位址為下列字母：A、B、C、E、I、J、K、Q、U、V、W、X、Y、Z。這些 NC 位址亦可以用位址延伸進行程式設計。以 STRINGIS 檢查時，可指定位址延伸。範例：201 == STRINGIS("A1")。
字母：D、F、H、L、M、N、O、P、S、T 等字母是使用者定義的 NC 位址或輔助功能。針對這些位址，一定是傳回 400 。範例：400 == STRINGIS ("D")。以 STRINGIS 檢查時，這些 NC 位址無法利用位址延伸指定。
範例：000 == STRINGIS ("M02")，但 400 == STRINGIS ("M")。

3) 循環參數的名稱無法以 STRINGIS 檢查

4) G、H、M、L 等定義為巨集的位址（例如）等同於巨集。

範例

以下範例假設指定為字串的 NC 語言元素（若無特殊狀況），原則上可用於控制器的程式設計。

1. 字串 "T" 定義為輔助功能：
   ```
   400 == STRINGIS ("T")
   000 == STRINGIS ("T3")
   ```
2. 字串 "X" 定義為軸：
   ```
   201 == STRINGIS ("X")
   201 == STRINGIS ("X1")
   ```
3. 字串 "A2" 定義為具有延伸的位址：
   ```
   201 == STRINGIS ("A")
   201 == STRINGIS ("A2")
   ```
4. 字串「INVCW」定義為 G 指令：
   ```
   202 == STRINGIS ("INVCW")
   ```
5. 字串「$MC_GCODE_RESET_VALUES」定義為機械資料：
   ```
   206 == STRINGIS ("$MC_GCODE_RESET_VALUES")
   ```
6. 字串 "GETMDACT" 為 NC 語言函數：
   ```
   203 == STRINGIS ("GETMDACT ")
   ```
7. 字串 "DEFINE" 為關鍵字
   ```
   205 == STRINGIS ("DEFINE")
   ```
8. 字串 "$TC_DP3" 為系統參數（刀長元件）：
   ```
   207 == STRINGIS ("$TC_DP3")
   ```
9. 字串 "$TC_TP4" 為系統參數（刀具尺寸）：
   ```
   207 == STRINGIS ("$TC_TP4")
   ```
10. 字串 "$TC_MPP4" 為系統參數（刀庫位置狀態）：
    - 刀庫管理生效：207 == STRINGIS ("$TC_MPP4")；
    - 刀庫管理未生效：`000 == STRINGIS ("$TC_MPP4")`

    亦請參見下文：刀庫管理。
11. 字串「MACHINERY_NAME」定義為 GUD 變數：
    ```
    209 == STRINGIS ("MACHINERY_NAME")
    ```
12. 字串「LONGMACRO」定義為巨集：
    ```
    210 == STRINGIS ("LONGMACRO")
    ```
13. 字串「MYVAR」定義為 LUD 變數：
    ```
    211 == STRINGIS ("MYVAR")
    ```
14. 字串 "XYZ" 是 NC、GUD 變數、巨集或循環名稱中未知的指令：
    ```
    000 == STRINGIS ("XYZ")
    ```

刀庫管理

刀庫管理功能若未生效，則由 STRINGIS 供應刀庫管理的系統參數，不論機械資料如何設定

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION

值一定是 000。

ISO 模式

若「ISO 模式」功能生效：

- MD18800 $MN_MM_EXTERN_LANGUAGE（啟用，外部 NC 語言）
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM（控制系統待調適）

STRINGIS 首先檢查指定的字串是否為 SINUMERIK G 指令。若字串非 SINUMERIK G 指令，則檢查其是否為 ISO G 指令。

已程式設計的切換（G290（SINUMERIK 模式）、G291（ISO 模式））對 STRINGIS 沒有影響。

範例

與函數 STRINGIS (...) 有關的機械資料具以下的值：

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION = 2（已設定選項的 NC 語言指令視為已知）
- MD19410 $ON_TRAFO_TYPE_MASK = 'H0'（選擇性：轉換）
- MD10700 $MN_PREPROCESSING_LEVEL='H43'（前置處理或單節生效）

以下程式可執行無誤：

程式碼	註解
N1 R1=STRINGIS ("TRACYL")	；R1 == 0，因為 TRACYL 定義為「未知」，因為未進行轉換選項
N2 IF STRINGIS ("TRACYL") == 204	
N3 TRACYL (1,2,3)	；N3 跳過
N4 ELSE	
N5 G00	；以執行 N5 代替
N6 ENDIF	
N7 M30	

18.7 由工件程式相互呼叫視窗（MMC）

透過預先定義的副程式 MMC(...)，可以從 SINUMERIK Operate 使用者介面上的 NC 程式顯示例如使用者專屬的對話框。

完成對話框的設定以供下列的對話框類型使用：

- Run MyScreens
- 簡易 XML
- 使用者 XML

參考資料

- 編程手冊 Run MyScreens
- 編程手冊簡易 XML

語法

```
MMC("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>","<QUIT>")
```

含義

MMC(...)：	子程式識別碼 指定位置編碼的參數，並且在二個字串、指令字串及確認字串間以逗號分隔。	
指令字串內的參數：		
<ADDRESS>：	在操作區裡建置設定的使用者對話框	
	函式	**操作區**
	「Run MyScreens」使用者對話框	CYCLES
	「簡易 XML」使用者對話框	CYCLES
	使用者 XML	XML
	彈出視窗「Run MyScreens」	POPUPDLG
	彈出視窗「簡易 XML」	POPUPDLG

<COMMAND>：	要執行的指令	
	函式	**指令**
	「Run MyScreens」使用者對話框	PICTURE_ON、PICTURE_OFF
	「簡易 XML」使用者對話框	PICTURE_ON、PICTURE_OFF
	使用者 XML	XML_ON、XML_OFF
	彈出視窗「Run MyScreens」	PICTURE_ON、PICTURE_OFF
	彈出視窗「簡易 XML」	PICTURE_ON、PICTURE_OFF
<FILE>：	包含要程式設計的對話框顯示之檔案名稱	
	函式	**檔案**
	「Run MyScreens」使用者對話框	<名稱>.com
	「簡易 XML」使用者對話框	<名稱>.xml
	使用者 XML	<名稱>.xml
	彈出視窗「Run MyScreens」	<名稱>.com
	彈出視窗「簡易 XML」	<名稱>.xml
	直接在 NC 程式內設定的彈出視窗「簡易 XML」（參閱範例 2）	xmldial_emb.xml
<DIALOG>：	欲顯示的對話框名稱	
	函式	**對話框名稱**
	直接在 NC 程式內設定的所有函式，彈出視窗「簡易 XML」除外	設定在 <FILE> 檔案中的對話框名稱
	直接在 NC 程式內設定的彈出視窗「簡易 XML」（參閱範例 3）	主要
確認字串內的參數：		

<QUIT>：	確認類型	
	N：	無確認。 指令被傳輸時，程式執行會繼續。若是指令無法成功執行，則沒有回應。 **說明** 若是顯示時間（停頓時間）係程式設計在 NC 程式內，則必須使用確認類型「N」（參閱範例 2 如下）
	A：	非同步確認 指令發出後，程式會繼續執行。將傳回值除存在使用者專屬的確認變數中（GUD 變數），該變數定義在對話框設定範圍內，可以在 NC 程式中讀取。

範例

範例 1

顯示對話框，回應給 NC 程式內的使用者操作。

```
;已建立的確認變數 QUIT 為全域使用者變數 (GUD)
;設定對話框時為 STRING 類型：
;DEF NCK STRING[20] QUIT
QUIT = "XXX"                                    ;初始化確認變數
G4 F5
MMC("CYCLES,PICTURE_ON,test.com,test1","A       ;顯示對話框
")                                              ;- 操作區：CYCLES
                                                ;- 圖片狀態：PICTURE_ON（顯示）
                                                ;- 對話框畫面檔案：test.com
                                                ;- 對話框畫面：test1
INPUT：                                         ;等候使用者輸入
   STOPRE                                       ;前置處理停止
   IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK        ;軟鍵「RUN」
   IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK       ;軟鍵「CHK」
GOTOB INPUT                                     ;=> 等候

WORK：                                          ;按住軟鍵「RUN」
MSG("繼續處理 -> NC 啟動")                      ;輸出訊息
MMC("CYCLES,PICTURE_OFF","N")                   ;關閉對話框
M0                                              ;等候 NC 啟動
GOTOF END                                       ;=> 程式結尾

CHECK：                                         ;按住軟鍵「CHK」
```

程式碼	註解
MSG("逼近位置 -> NC 啟動")	；輸出訊息
MMC("CYCLES,PICTURE_OFF","N")	；關閉對話框
M0	；等候 NC 啟動
GOTOF END	；=> 程式結尾
END：	
...	

範例 2

例如，在 NC 程式中透過停頓時間定義對話框的顯示時間。

程式碼	註解
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_**ON**,xmldial_emb.xml,main","N")	；顯示對話框
X200	
Z40	
MMC("POPUPDLG,PICTURE_**OFF**","N")	；關閉對話框

範例 3

在 NC 程式內嵌入快顯畫面命令集及其使用。

程式碼

```
PROC POPUP_TEST
;------------------------------ 命令集 ------------------------------
;<main_dialog entry="rpara_main">
;<let name="xpos" />
;<let name="ypos" />
;<let name="field_name" type="字串" />
;<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>來自 NC 工件程式的遮罩</caption>
;<let name="count" >0</let>
;<op>
;xpos = 120;
```

程式碼

```
;ypos = 34;
;"nck/通道/參數/R[10]" = 10;
;</op>
;<!-- 載入控制器編號 -->
;<op>
;num = "nck/通道/參數/R[10]";
;</op>
;<while>
;<condition> count < num</condition>
;<print name="field_name" text="edit%d">count</print>
;<op>
;ypos = ypos + 24;
;count = count + 1;
;</op>
;</while>
;</init>
;<paint>
;<op>
;xpos = 8;
;ypos = 36;
;count = 0;
;</op>
;<while>
;<condition>count < num</condition>
;<print name="field_name" text="R-Parameter%d">count</print>
;<text xpos = "$xpos" ypos = "$ypos" >$$$field_name</text>
;<op>
;ypos = ypos + 24;
;count = count + 1;
;</op>
;</while>
;</paint>
;</form>
;</main_dialog>
;========================== 程式節段 ============================
...
G94 F100
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")
G4 F4
X200
MMC("POPUPDLG,PICTURE_OFF","N")
```

程式碼

```
G4 F2
X0
...
```

補充條件

- 對話框的定義檔 *.com 必須儲存在「proj」資料夾。
- 對話框的簡易 XML 定義檔 *.xml 必須儲存在「appl」資料夾。
  若是將定義檔儲存在不同的目錄，路徑必須間接指定，從「appl」目錄開始。
- 使用者定義的對話框無法從不同的通道同時顯示。

18.8 程式執行時間 / 工件計數器

提供關於程式執行時間及工件計數器之資訊，以支援機台刀具操作員。

可在 NC 及 / 或 PLC 程式中做為系統變數處理本資訊。亦可在操作員介面上顯示本資訊。

18.8.1 程式執行時間

“程式執行時間”功能提供內部 NC 計時器以監控技術處理，該資訊可透過 NC 及通道專屬系統變數，讀入至工件程式及同步動作。

執行時間測量之觸發 ($AC_PROG_NET_TIME_TRIGGER) 為該函數唯一可寫入的系統變數，並使用於測量選取的程式區段。這表示可透過 NC 程式中寫入 $AC_PROG_NET_TIME_TRIGGER，以再度啟動及停用時間測量：

系統變數	含義	活動
NC 專屬		
$AN_SETUP_TIME	距離上次以預設值（"冷開機"）控制開機的時間，以分鐘計算： 每次以預設值控制開機，會自動重置為"0"。	• 永遠有效
$AN_POWERON_TIME	距離上次正常控制開機（"暖開機"）的時間，以分鐘計算。 每次正常控制開機，會自動重置為"0"。	
通道專屬		

系統變數	含義	活動
$AC_OPERATING_TIME	在自動模式中，NC 程式的總執行時間，以秒計算。 每次控制開機，該值會自動重置為"0"。	• 透過 MD27860 啟動 • 僅有 AUTOMATIC 模式
$AC_CYCLE_TIME	選定 NC 程式的執行時間，單位為秒。 每次有新的 NC 程式開機，該值會自動重置為 "0"。	
$AC_CUTTING_TIME	以秒為單位的處理時間 所有介於 NC 啟動與程式結束／NC 重置之間無快速移動且刀具有效時的所有程式路徑軸（至少一個有效）之執行時間。出現停頓時間時，測量會中斷。 每次以預設值控制開機，該值會自動重置為"0"。	
$AC_ACT_PROG_NET_TIME	目前的 NC 程式之實際執行時間，以秒為單位。 當新的 NC 程式開始時，會自動重置為"0"。	• 永遠有效 • 僅有 AUTOMATIC 模式
$AC_OLD_PROG_NET_TIME	剛以 M30 正確終止的程式執行時間，以秒表示	
$AC_OLD_PROG_NET_TIME_COUNT	變更為$AC_OLD_PROG_NET_TIME_COUNT 在 POWER ON（開機）後，$AC_OLD_PROG_NET_TIME_COUNT 在 “0” 。 若新寫入控制至$AC_OLD_PROG_NET_TIME，則永遠增加 $AC_OLD_PROG_NET_TIME_COUNT。	

系統變數	含義		活動
$AC_PROG_NET_TIME_TRIGGER	執行時間量測之觸發：		● 僅有 AUTOMATIC 模式
	0	中立狀態 觸發未啟用。	
	1	離開 結束測量，並將數值由 $AC_ACT_PROG_NET_TIME 複製到 $AC_OLD_PROG_NET_TIME。將 $AC_ACT_PROG_NET_TIME 設定為「0」然後繼續執行。	
	2	開始 開始測量，並因此將 $AC_ACT_PROG_NET_TIME 設為「0」。$AC_OLD_PROG_NET_TIME 沒有變更。	
	3	停止 量測停止。不變更 $AC_OLD_PROG_NET_TIME 並保持 $AC_ACT_PROG_NET_TIME 恆定，直到繼續執行	
	4	繼續 繼續執行測量，即繼續先前停止的測量。$AC_ACT_PROG_NET_TIME 繼續，$AC_OLD_PROG_NET_TIME 沒有變更。	
POWER ON（開機）後，將所有系統變數重置為 0！			

說明

機台製造商

使用機械資料 MD27860 $MC_PROCESSTIMER_MODE 開啟可啟動計時器。

部分函數（例如 GOTOS、手動超調 = 0%，生效測試運轉進給、程式測試、ASUB、PROG_EVENT ...）的生效時間量測行為，是以機械資料 MD27850 $MC_PROG_NET_TIMER_MODE 和 MD27860 $MC_PROCESSTIMER_MODE 設定。

參考文獻：

功能手冊基本功能；BAG、通道、程式操作、重置回應（K1），章節：程式執行時間一章

說明

工件的剩餘時間

若連續加工同樣的工件，使用下列的計時器數值，即可判定出工件剩下的剩餘時間。

- 處理最後一個工件所花的時間（請參考$AC_OLD_PROG_NET_TIME）
- 目前的處理時間（請參考$AC_ACT_PROG_NET_TIME）

除了目前的處理時間以外，剩餘的時間也會顯示在使用者介面上。

說明

使用 STOPRE

系統變數 $AC_OLD_PROG_NET_TIME 及 $AC_OLD_PROG_NET_TIME_COUNT 不產生任何間接前置處理停止。若系統變數值來自先前的程式執行，則在工件程式中使用時便不重要。然而，若因$AC_OLD_PROG_NET_TIME 改變頻繁，而頻頻寫入執行時間量測之觸發（$AC_PROG_NET_TIME_TRIGGER），則應在工件程式中明確使用 STOPRE。

補充條件

- **單節搜尋**
  沒有程式執行時間透過單節搜尋決定。
- **REPOS**
  REPOS 的處理時間會被加在目前的處理時間上（$AC_ACT_PROG_NET_TIME）。

範例

範例 1：量測「mySubProgrammA」的期間

程式碼

```
...
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
```

在程式處理 N80 行後，「mySubProgrammA」的淨執行時間置於 $AC_OLD_PROG_NET_TIME。

來自$AC_OLD_PROG_NET_TIME 的數值：

- 在 M30 之後仍舊保留。
- 在每次執行迴圈時更新。

範例 2：量測「mySubProgrammA」及「mySubProgrammC」的期間

程式碼

```
...
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 工件計數器

"工件計數"函數使可用的數個技術器，可用於控制中特定的輪廓工件上。

該計數器以通道專屬系統變數形式存在，可在數值範圍 0 到 999 999 999 間讀寫存取。

系統變數	含義
$AC_REQUIRED_PARTS	待生產的工件數量（工件的設定點號碼） 在此計數器中，您可以定義工件數量，在此實際的工件計數器 $AC_ACTUAL_PARTS 重設為"0"。
$AC_TOTAL_PARTS	所有完成的工件總數（實際的工件加總） 此計數器會指定了從開始到現在所生產的所有工件的總數量。當以預設值控制開機，該值只會自動重置為"0"。

系統變數	含義
$AC_ACTUAL_PARTS	所有完成的工件數（實際的工件加總） 此計數器會記錄了從開始到現在所生產的所有工件的總數量。在$AC_REQUIRED_PARTS > 0 的條件下，當達到所需的工件數量（$AC_REQUIRED_PARTS）時，該計數器會自動重置為「0」。
$AC_SPECIAL_PARTS	使用者選擇的工件數量 此計數支援使用者自訂工件計數。當到達了工件的設定點號碼（$AC_REQUIRED_PARTS），可定義輸出一個警報。使用者必須自行重置計數器。

說明

當控制以預設值啟動時，所有工件計數器皆設為「0」，並可獨立於啟動進行讀寫。

說明

通道專屬機械資料可用於控制計數器啟用、計數器重置時間以及計數演算法。

說明

以使用者自訂 M 指令進行工件計數

可設定機械資料，因此用於數個工件輪廓的計數脈衝，便可透過使用者自訂 M 指令來觸發，而不用等到程式結尾（M2/M30）。

18.9 Process DataShare - 輸出至外部裝置 / 檔案（EXTOPEN、WRITE、EXTCLOSE）：

從工件程式寫入資料至外部裝置 / 檔案分成三步驟：

1. 開啟外部裝置/檔案
   開啟通道的外部裝置 / 檔案，使用 EXTOPEN 指令寫入。
2. 寫入資料
   輸出資料可以使用 NC 語言的字串功能處理，例如 SPRINT。使用 WRITE 指令寫入。
3. 關閉外部裝置/檔案
   到達程式結尾 (M30) 或通道重置時，通道指派的外部裝置 / 檔案可以使用 EXTCLOSE 指令再次釋放。

語法

```
DEF INT <Result>
DEF STRING[<n>] <Output>
…
EXTOPEN(<Result>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
…
<Output>="data output"
WRITE(<Result>,<ExtDev>,<Output>)
…
EXTCLOSE(<Result>,<ExtDev>)
```

含義

EXTOPEN：	開啟外部裝置 / 檔案的預先定義程序		
<Result>：	**參數 1：**結果變數 使用結果變數值可以評估程式中的操作是否成功，然後繼續適當處理。		
	類型：	INT	
	值：	0	無錯誤
		1	無法開啟外部裝置
		2	無法設定外部裝置
		3	外部裝置設定了無效的路徑
		4	沒有外部裝置的存取權限
		5	使用模式：外部裝置已被「獨」佔
		6	使用模式：外部裝置已被「分享」
		7	檔案長度超過 LOCAL_DRIVE_MAX_FILESIZE
		8	已經超過外部裝置的最大數量
		9	LOCAL_DRIVE 的選項未設定
		11	V.24 介面已指派至 Easy-Message 功能 (僅限 828D)
		12	寫入模式：資料與 extdev.ini 產生矛盾
		16	程式設計無效的外部路徑
		22	外部裝置未裝載

<table>
<tr><td rowspan="10"><ExtDev>：</td><td colspan="2">參數 2：欲開啟之外部裝置 / 檔案的符號識別碼</td></tr>
<tr><td>類型：</td><td>STRING</td></tr>
<tr><td colspan="2">符號識別碼包括：
1. 邏輯裝置名稱
2. 部份附檔案路徑（使用 "/" 附加）。
下列的邏輯裝置名稱已定義：</td></tr>
<tr><td>"LOCAL_DRIVE"：</td><td>本機 CF 卡（預先定義）</td></tr>
<tr><td>"CYC_DRIVE"：</td><td>保留的磁碟名稱供 SIEMENS 循環使用（預先定義）</td></tr>
<tr><td>"/dev/ext/1"、...
"/dev/ext/9"：</td><td>可用的網路磁碟機
附註：
必須在 extdev.ini 檔案中設定！</td></tr>
<tr><td>"/dev/cyc/1"、
"/dev/cyc/2"：</td><td>保留的磁碟名稱供 SIEMENS 循環使用
附註：
必須在 extdev.ini 檔案中設定！</td></tr>
<tr><td>"/dev/v24"：</td><td>V.24 介面
附註：
必須在 extdev.ini 檔案中設定！</td></tr>
<tr><td colspan="2">檔案路徑：
● 檔案路徑必須指定給「LOCAL_DRIVE」及「CYC_DRIVE」例如：「LOCAL_DRIVE/my_dir/my_file.txt」
● 邏輯裝置名稱「/dev/ext/1...9」及「/dev/cyc/1...2」可以設定為：
– 已指向只能指定邏輯裝置名稱的檔案，例如：「/dev/ext/4」
– 或指向必須指定檔案路徑的目錄，例如：「/dev/ext/5/my_dir/my_file.txt」
● 檔案路徑不允許附屬於「/dev/v24」。</td></tr>
<tr><td colspan="2">附註：
邏輯裝置名稱「/dev/ext/1...9」、「/dev/v24」及「/dev/cyc/1...2」可以忽略大小寫；欲將路徑指定給檔案時才需要注意大小寫。
「LOCAL_DRIVE」及「CYC_DRIVE」只允許大寫字母。</td></tr>
</table>

<SyncMode>：	**參數 3：**寫入此裝置 / 檔案的 WRITE 指令處理模式		
	類型：	STRING	
	值：	"SYN"：	同步寫入 程式執行停止，直到寫入操作完成為止。 評估 WRITE 指令的錯誤變數可以檢查同步寫入是否成功完成。
		"ASYN"：	非同步寫入 程式執行不會被 WRITE 指令中斷。 **附註：** 於此模式 WRITE 指令的結果變數並不會提供任何資訊，數值永遠為 0（無錯誤）。在此特殊模式中，WRITE 指令成功與否無法確定。
<AccessMode>：	**參數 4：**此裝置 / 檔案的使用模式		
	類型：	STRING	
	值：	"SHARED"：	在「共享」模式中要求裝置/檔案。其他通道也可以使用裝置，亦即可在此模式中開啟。
		"EXCL"：	通道單獨使用裝置/檔案；其他通道不能使用該裝置。
<WriteMode>：	**參數 5：**寫入此檔案 / 裝置的 WRITE 指令寫入模式（選擇性）		
	類型：	STRING	
	值：	"APP"：	附加 檔案內容永遠保存；寫入呼叫附加於結尾。
		"OVR"：	覆寫 刪除檔案內容，並使用後續的寫入呼叫重新產生。
	附註： 使用此參數，設定在 extdev.ini 檔案中的寫入模式無法被覆寫。發生衝突的情況下，以錯誤確認 EXTOPEN 呼叫。		

寫入：	寫入輸出資料的預先定義程序

EXTCLOSE：	關閉已開啟之外部裝置 / 檔案的預先定義程序		
<Result>：	**參數 1：**結果變數		
	類型：	INT	
	值：	0	無錯誤
		16	程式設計無效的外部路徑
		21	關閉外部裝置時錯誤
<ExtDev>：	**參數 2：**欲關閉之外部裝置 / 檔案說明的符號識別碼，請參閱 EXTOPEN！ **附註：** 識別碼必須與 EXTOPEN 呼叫指定的識別碼相同！		

範例

程式碼

```
N10       DEF INT RESULT
N20       DEF BOOL EXTDEVICE
N30       DEF STRING[80] OUTPUT
N40       DEF INT PHASE
N50       EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60       IF RESULT > 0
N70          MSG("Error for EXTOPEN:" << RESULT)
N80       ELSE
N90          EXTDEVICE=TRUE
N100      ENDIF
…
N200      PHASE=4
N210      IF EXTDEVICE
N220         OUTPUT=SPRINT("End phase: %D",PHASE)
N230         WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240      ENDIF
…
```

另請參見

字串運算 (頁 91)

寫入檔案（WRITE）(頁 158)

18.10 警報（SETAL）

可在 NC 程式中設定警報。 警報顯示於使用者介面中的另一個欄位。 警報永遠伴隨控制依警報類別所發出的回應。

參考資料：
關於警報回應的其他資訊，請參考「調試手冊」。

語法

```
SETAL(<警報編號>[,<字元字串>])
```

含義

SETAL：	程式設計警報的關鍵字。 SETAL 必須在獨立 NC 單節中進行程式設計。	
<警報編號>：	INT 類別變數。 內含警報編號。 警報編號有效範圍介於 60000 及 69999 間，其中保留 60000 到 64999 供 SIEMENS 循環使用，65000 到 69999 供使用者使用。	
<字元串>：	在程式設計使用者循環警報時，可另外指定最多含四個參數的字元字串。 可在這些參數中定義變數使用者文字。 然而，可使用下列預先定義的參數：	
	參數	**含義**
	%1	通道編號
	%2	單節編號，標籤
	%3	循環警報的文字索引
	%4	附加警報參數

說明
警報文字必須在使用者介面中設定。

說明
如果警報是以使用者介面啟用的語言輸出，則使用者所要求的資訊為 HMI 目前設定的語言。這項資訊可以使用系統變數 $AN_LANGUAGE_ON_HMI 在工件程式及同步動作中查詢（請參閱「目前設定在 HMI 中的語言 (頁 1042)」）。

範例

程式碼	註解
...	
N100 SETAL （65000）	；設定警報 No. 65000
...	

18.11 延伸停止及回退 (ESR)

延伸停止及回退功能（隨後呼叫 ESR）在故障情形發生時，提供靈活反應的可能性，作為製程函數使用：

- **延伸停止**
  假設特定的故障情形允許啟用延伸停止的所有軸依序停止。
- **回退**
  目前正在使用的刀具盡速從工件回退。
- **發電機操作（SINAMICS 驅動功能「Vdc 控制」）**
  若 DC 連結電壓的參數化值過低，例如因為線路電壓失效，則需將此情況專用的驅動煞車能量恢復，以產生回退所需的電能（發電機操作）。

觸發源

一般來源（NC 外部 / 全域或模式群組 / 通道專屬）：

- 數位輸入（如位於 NCU 模組），或控制內部可回讀的數位輸出內容 ($A_IN, $A_OUT)
- 通道狀態 ($AC_STAT)
- VDI 訊號 ($A_DBB)
- 數個警報的群組訊息 ($AC_ALARM_STAT)

軸向來源

- 跟隨軸的緊急回退門檻值（電控耦合之同步，$VA_EG_SYNCDIFF[<跟隨軸>]）
- 驅動器： DC 連結警告門檻值（逼近電壓過低），$AA_ESR_STAT[<軸>]
- 驅動器： 發電機最小速度門檻值（無更多可用再生旋轉能量），$AA_ESR_STAT[<軸>]。

靜態同步動作的閘控邏輯： 來源 / 回應連結

使用靜態同步動作'彈性閘控，依其來源快速觸發專屬反應。

使用靜態同步動作連結所有相關來源是使用者的責任。 操作員可選擇對來源系統變數作整體評估，或以位元記號，然後以其所要反應進行邏輯操作。 靜態同步動作在所有操作模式中皆有效。

參考資料：
功能手冊，同步動作

啟用

功能啟用

設定相對應的控制信號 $AA_ESR_ENABLE，啟用發電機操作、關機、回退等功能。 可用同步動作變更該控制訊號。

功能觸發

設定系統變數 $AC_ESR_TRIGGER 可同時觸發所有啟用軸的 ESR。

偵測到 DC 連結電壓逼近過低時，驅動器內的發電機操作會「自動」啟用。

偵測到通訊故障（NC 與驅動器間）及在驅動器內偵測到 DC 連結電壓過低（需要配置及啟用）時，會啟動獨立驅動停止及 / 或回退。

設定適當的控制訊號 $AN_ESR_TRIGGER（對所有驅動器廣播的指令），亦可由 NC 側觸發獨立驅動停止及 / 或回退。

參考資料

若需更多有關 ESR 的詳細資訊，請參閱：

功能手冊，特殊功能；延伸停止及回退 (R3)

18.11.1 NC 控制 ESR

18.11.1.1 NC 控制回退 (POLF, POLFA, POLFMASK, POLFMLIN)

NC 控制回退所需的特定初始條件（請參閱「NC 控制回退 (POLF, POLFA, POLFMASK, POLFMLIN) (頁 729)」）。 當滿足上述需求後，設定系統變數 $AC_ESR_TRIGGER（單軸則使用 $AA_ESR_TRIGGER），可啟動通道中針對回退軸所設定的快速舉升 (LIFTFAST)。

語法

```
POLF(<軸>)=<位置>
POLFA（<軸>，<類型>，<位置>）
POLFMASK（<軸_1>，<軸_2>，...）
POLFMLIN（<軸_1>，<軸_2>，...）
```

```
POLFA 可使用下列簡化格式：
POLFA（<軸>，<類型>）；單軸回退之簡化格式
POLFA（軸，0/1/2）；快速停用或啟用
POLFA（軸，0，$AA_POLFA[軸]）；造成前置處理停止
```

POLFA（軸，0）；不會造成前置處理停止

含義

POLF：	用於指定回退軸的目標位置 POLF 為模態。			
	<軸>：	回退的幾何或通道 / 機械軸名稱		
	<位置>：	退刀位置		
		類型：	REAL	
		WCS 適用於幾何軸，除了 MCS。 與幾何和通道 / 機械軸具有相同的識別碼，在 WCS 中退刀。		
POLFA：	預先定義的子程式呼叫單軸退刀位置的規格			
	<軸>：	通道軸識別碼		
	<類型>：	位置規格模式		
		類型：	INT	
		值：	0:	將位置值標記為無效
			1:	位置值為絕對
			2:	位置值為增量（距離）
		附註： 若軸非單一軸、類型遺失或類型為 0，將輸出相對應的警報。		
	<位置>：	退刀位置（請參考上述）		
		附註： 位置值的類型亦可為 0，但唯有該值標記為無效時才能接受，且必須針對退刀重新設定。		
POLFMASK：	**彼此互相獨立**的快速舉升作動後，預先定義的子程式將呼叫預定回退的軸選項。			
	<軸_1>，...：	快速舉升期間，預定移動至以 POLF 定義之位置的軸名稱。 指定的軸皆需位於相同座標系統中。		
	未指定軸規格的 POLFMASK() 可停用所有回退軸的快速舉升，彼此互相獨立。			

POLFMLIN：	**具線性關係**的快速舉升作動後，預先定義的子程式將呼叫預定回退的軸選項。	
	<軸_1>，...：	請參考上述。
	未指定軸規格的 POLFMLIN() 可停用所有回退軸的快速舉升，彼此具線性關係。	

說明

透過 POLFMASK 或 POLFMLIN 快速回退至固定位置前，必須先以 POLF 設定選定軸的位置。

說明

若以 POLFMASK、POLFMLIN 或 POLFMLIN、POLFMASK 依序啟用軸，則該軸永遠會套用最後一個定義。

說明

在工件程式啟動後刪除以 POLF 程式設計之位置及 POLFMASK 或 POLFMLIN 啟用之動作。 這表示使用者需在每個工件程式中重新程式設計 POLF 的值及在 POLFMASK 或 POLFMLIN 中選取的軸。

說明

若使用簡化格式 POLFA 時，僅有改變類型，則使用者需確認回退位置或回退路徑含有一實用且合理之值。 特別在電源開啟後，需再次設定回退位置及回退路徑。

範例

個別軸回退：

程式碼	註解
MD37500 $MA_ESR_REACTION[AX1]=21	；NC 控制回退。
...	
$AA_ESR_ENABLE[AX1] = 1	
POLFA(AX1,1,20.0)	；指派 20.0 的軸向回退位置（絕對）給 AX1。
$AA_ESR_TRIGGER[AX1] = 1	；由此開始回退。

其他資訊

NC 控制回退的需求

- 在通道中設定執行 NC 控制回退的回退軸：
  MD37500 $MA_ESR_REACTION = 21
- 該軸的 ESR 必須啟用：
  $AA_ESR_ENABLE = 1
- 定義延遲時間：
  MD21380 $MC_ESR_DELAY_TIME1
  MD21381 $MC_ESR_DELAY_TIME2
- 在工件程式中，以 POLF 設定軸專屬回退位置。
- 使用 POLFMASK/POLFMLIN 選擇 NC 控制回退的軸。
- 需為回退移動設定啟動訊號，並維持該設定。

啟用並啟動 NC 控制反應

若系統變數設定為 $AC_ESR_TRIGGER = 1，並在此通道中設定回退軸（即 MD37500 $MA_ESR_REACTION = 21）以及設定該軸的 $AA_ESR_ENABLE = 1，然後將啟動此通道中的快速舉升 (LIFTFAST)。

在工件程式中，以 POLF（或 POLF）為透過 POLFMASK 或 POLFMLIN 選取之軸設定的舉升動作，可取代軸定義的路徑動作。

MD21380 $MC_ESR_DELAY_TIME1 和 MD21381 $MC_ESR_DELAY_TIME2 的時間總和，為回退可用的最長時間。 時間超過時，也會為回退軸發動含跟進之快速減速。

說明

無法中斷延伸回退（即經由 $AC_ESR_TRIGGER 觸發之 LIFTFAST/LFPOS），僅能透過緊急停止提前終止。

說明

鎖定經 $AC_ESR_TRIGGER 啟動之回退，以防止多次回退。

單軸回退

使用單軸回退時，單軸回退位置必須使用 POLFA 設定，且必須滿足以下條件：

- $AA_ESR_ENABLE = 1
- 觸發 ($AAAA_ESR_TRIGGER = 1) 時，**<軸>** 必須為單軸。
- **<類型>** 必須為 1 或 2。

快速舉升期間的回退方向

會考量啟動升起快速時的有效框架。

說明

含旋轉的框架也會影響透過 POLF 升起的方向。

座標軸替換

永遠只能將回退軸指派給一個 NC 通道，且不能在通道間切換。 嘗試在另一通道交換回退軸時，將會輸出警報。 一旦使用 $AA_ESR_ENABLE[AX] = 0 再次停用該軸後，才能在新通道交換軸。 一旦該軸交換後，便可以 $AA_ESR_ENABLE[AX] = 1 再次動作。

中立軸

中立軸無法進行 NC 控制 ESR。

18.11.1.2 NC 控制停止

設定系統變數 $AC_ESR_TRIGGER（單軸則使用 $AA_ESR_TRIGGER），可對在通道中設定的停止軸啟動 NC 控制停止。

條件

- 在通道中設定執行 NC 控制停止的停止軸：
  MD37500 $MA_ESR_REACTION = 22
- 該軸的 ESR 必須啟用：
  $AA_ESR_ENABLE = 1
- 定義延遲時間：
  MD21380 $MC_ESR_DELAY_TIME1（延遲時間，ESR 軸）
  MD21381 $MC_ESR_DELAY_TIME2（插補煞車的 ESR 時間）

執行

本軸在 MD21380 設定期間，仍依程式設計繼續進行插補： 在超過 MD21380 指定的延遲時間後，將發動控制煞車（斜率停止）： MD21381 內的時段為插補控制煞車的最長可用時間。 該時段結束後，將啟動快速剎車與後續追蹤。

範例

停止單軸：

程式碼	註解
MD37500 $MC_ESR_REACTION[AX1] = 22	；NC 控制停止。
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	；由此開始停止。

18.11.2 驅動整合式 ESR

18.11.2.1 設定整合式驅動停止 (ESRS)

整合式驅動 ESR 功能的「停止」驅動參數係使用 ESRS(...) 功能設定。

語法

```
ESRS(<access_1>,<stopping time_1>[,...,<axis_n>,<stopping
time_n>])
```

含義

ESRS(...)：	將 ESR 功能「停止」寫入驅動參數的功能。 功能： ● 必須單獨放在單節中。 ● 觸發前置處理停止。 ● 無法使用在同步動作上。	
<axis_1>, ..., <axis_n>：	整合式驅動停止應該設定的軸。 將此軸的驅動參數 p0888（組態）寫入驅動器： p0888 = 1	
	類型：	AXIS
	值域：	通道軸識別碼

<stopping time_1>, ..., <stopping time_n>：	驅動器發生故障後，持續以實際速度設定點移動的時間。 將指定軸的驅動參數 p0892（計時器）寫入驅動器： p0892 = <停止時間>	
	單位：	s
	類型：	REAL
	值域：	0.00 - 20.00
一個功能呼叫最多可以程式設計 5 個軸；n = 5		

18.11.2.2 設定整合式驅動回退 (ESRS)

整合式驅動器 ESR 功能的「回退」驅動參數係使用 ESRR(...) 功能設定。

語法

```
ESRR(<axis_1>,<retraction distance_1>,<retraction
velocity_1>[,...,<axis_n>,<retraction distance_n>,<retraction
velocity_n>])
```

含義

ESRR(...)：	將 ESR 功能「回退」寫入驅動參數的功能。 功能： ● 必須單獨放在單節中。 ● 觸發前置處理停止。 ● 無法使用在同步動作上。	
<axis_1>, ..., <axis_n>：	整合式驅動回退應該設定的軸。 將此軸的驅動參數 p0888（組態）寫入驅動器： p0888 = 2	
	類型：	AXIS
	值域：	通道軸識別碼

<retraction distance_1>, ..., <retraction distance_n>：	將驅動器的回退距離轉換成回退速度。 將指定軸的數值寫入驅動參數 p0893（速度）： p0893 =（<retraction distance _n> 轉換成回退速度）	
	單位：	毫米 / 分、英吋/分、度/分（視軸的單位而定）
	類型：	REAL
	值域：	MIN - MAX
<retraction velocity_1>, ..., <retraction velocity_n>：	將驅動器的回退速率轉換成時間。 將指定軸的數值寫入驅動參數 p0892（時間）[s]： p0892 = <retraction distance_n> / <retraction velocity _n>	
	單位：	毫米 / 分、英吋/分、度/分（視軸的單位而定）
	類型：	REAL
	值域：	0.00 - MAX
一個功能呼叫最多可以程式設計 5 個軸；n = 5		

18.12 定義坯料 (WORKPIECE)

控制器必須知道坯料的形狀大小才能夠在圖形模擬中顯示。因此，使用者有能力透過使用者介面或是直接在 NC 程式中定義坯料。坯料的定義即使在重置之後（程式結束 / 通道 / BAG）仍可保留。等到下一次控制系統重新開機時，就會自動刪除。

語法

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

意義

WORKPIECE(...)：		定義坯料的預先定義程序		
		前置處理停止：	是	
		單獨在單節中：	是	
參數：				
1	"<WP>"：	工件名稱（選擇性）		
		資料類型：	STRING	
		若是一個通道中可以有幾個工件，才需要規格。沒有指定，「WORKP<n>」會自動接受，其中 <n> 代表宣告的通道編號。		
2	"<RefP>"：	鉗緊（選擇性，僅適用銑削機台）		
		資料類型：	STRING	
		值域：	「轉盤」	鉗緊固定的轉盤
			「A」	鉗緊旋轉軸 A
			「B」	鉗緊旋轉軸 B
			「C」	鉗緊旋轉軸 C
		先決條件： 轉盤或旋轉軸必須透過鉗緊坯料的對應機械資料啟用（請參閱 SINUMERIK Operate 調試手冊）。		

3	"<ZeroOffset>"：	定位坯料的可設定工作偏移量（無法程式設計） 選擇定位坯料的可設定工作偏移量僅能透過使用者介面提供給坯料的輸入內容。若要在工件程式中直接定義坯料，則坯料總是與目前的有效工作偏移量有關。		
4	"<Type>"：	坯料形狀		
		資料類型：	STRING	
		值域：	"CYLINDER"：	圓柱
			"PIPE"：	圓管
			"RECTANGLE"：	中心矩體
			"BOX"：	矩體
			"N_CORNER"：	n 個稜邊的多邊形
5 ... 12	<Par5> ... <Par12>：	坯料形狀的說明參數		
		資料類型：	REAL	
		需要參數編號，其含義根據對應的坯料形狀及位元的參數值。 請參閱： ● 「坯料形狀說明參數」表 ● 「位元參數」表		
WORPIECE()：		呼叫不含參數的 WORKPIECE 會刪除所有的坯料定義。		
WORPIECE(<WP>)：		呼叫含工件名稱的 WORKPIECE 只會刪除這一個坯料的定義。		

表格 18-1　坯料形狀的說明參數

坯料形狀	參數							
	<Par5>	<Par6>	<Par7>	<Par8>	<Par9>	<Par10>	<Par11>	<Par12>
圓柱	位元參數實數值轉譯成位元編碼的整數值。位元定義下列參數的意義（參閱「位元參數」表）。	參考點 Z_0	長度 Z_1	加工尺寸 Z_B	外直徑 d_0	-	繞著旋轉軸旋轉	-
圓管		參考點 Z_0	長度 Z_1	加工尺寸 Z_B	外直徑 d_0	壁厚（增量）/ 內直徑 d_1（絕對）	繞著旋轉軸旋轉	-
中心矩體		參考點 Z_0	長度 Z_1	加工尺寸 Z_B	寬度 W	長度 L	繞著旋轉軸旋轉	-
矩體		參考點 Z_0	長度 Z_1	加工尺寸 Z_B	X_0	Y_0	X_1	Y_1
n 個稜邊的多邊形		參考點 Z_0	長度 Z_1	加工尺寸 Z_B	角的數目	跨越平面的寬度	繞著旋轉軸旋轉	-

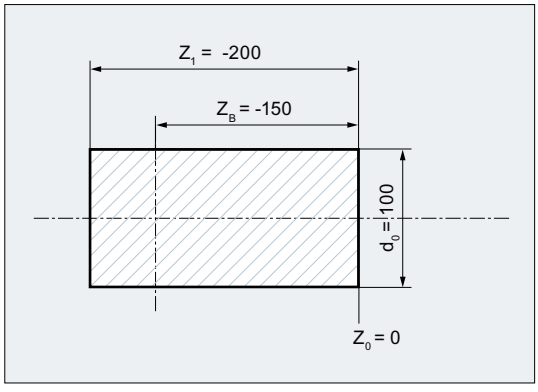
表格 18-2　位元參數

位元	意義	
4 (0x0010)	矩體：X_1	
	= 0	增量
	= 1	絕對
5 (0x0020)	矩體：Y_1	
	= 0	增量
	= 1	絕對
6 (0x0040)	長度 Z_1（最後尺寸）	
	= 0	增量
	= 1	絕對
位元 7 (0x0080)	加工尺寸 Z_B	
	= 0	增量
	= 1	絕對
位元 8 (0x0100)	圓管：壁厚 / 內直徑	
	= 0	增量
	= 1	絕對

位元	意義	
9 (0x0200)	n 個稜邊的多邊形	
	= 0	跨越平面的寬度
	= 1	邊長
12 (0x1000)	車削機台的鉗緊	
	= 0	主要主軸
	= 1	副主軸
13 (0x2000)	副主軸	
	= 0	有鏡射
	= 1	無鏡射

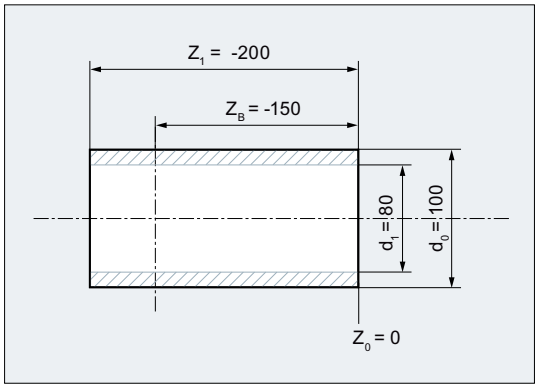
範例

範例 1：車削機台上的圓柱形坯料



程式碼	註解
...	
WORKPIECE(,,,"CYLINDER",0,0,-200,-150,100)	；**坯料定義：** ；坯料形狀：圓柱 ；位元參數 = 0（無位元集）→ 長度值及加工尺寸為增量，坯料在主要主軸 ；參考點 (Z0)=0 ；長度 (Z1)=-200 ；加工尺寸 (ZB)=-150 ，外直徑 (d0)=100
...	

範例 2：車床上的圓管狀坯料



程式碼	註解
...	
WORKPIECE(,,,"PIPE",256,0,-200,-150,100,80)	；**坯料定義：** ；坯料形狀：圓管 ；位元參數 = 256(Bit8=1) → 內直徑為絕對；長度及加工尺寸為增量，坯料在主要主軸 ；參考點 (Z0)=0 ；長度 (Z1)=-200 ；加工尺寸 (ZB)=-150 ，外直徑 (d0)=100 ；內直徑 (d1)=80
...	

18.13 切換語言模式 (G290, G291)

控制器讓您能夠從外部 CNC 系統讀入工件程式及進行處理。先決條件是對應的 NC 語言模式（ISO 用語）在調適期間已經定義。

參考資料：
功能手冊 ISO 用語

ISO 用語模式可以針對每個通道個別啟動。例如，通道 1 可以在 ISO 用語模式中執行，同時通道 2 在 SINUMERIK 模式啟動。

在 SINUMERIK 模式與 ISO 用語模式之間切換可以透過 G 碼群組 47 在 NC 程式中完成。啟用的刀具、刀具補正及工作偏移量不受切換影響。

語法

```
G291
...
G290
```

意義

G290：	啟用 SINUMERIK 語言模式	
	單獨在單節中：	是
	有效：	模態
G291：	啟用 ISO 語言模式	
	單獨在單節中：	是
	有效：	模態

條件

SINUMERIK 模式

- 可透過機械資料定義每一個通道預設的 G 指令。
- 在 SINUMERIK 模式中，無法編寫任何源自 ISO 用語的語言指令。

ISO 用語模式

- 可利用機械資料將 ISO 用語模式設為控制系統的基本設定。在 ISO 用語模式中，控制系統以預設接著重新開機。
- 只能編寫源自於 ISO 用語的 G 指令。在 ISO 用語模式中無法編寫 SINUMERIK G 功能。
- 同一個 NC 單節中，不能混用 ISO 用語和 SINUMERIK 語言。
- 無法使用 G 指令在 ISO 用語 M（銑削）與 ISO 用語 T（車削）之間切換。
- 可以呼叫在 SINUMERIK 模式中編寫的副程式。
- 若是使用 SINUMERIK 功能，必須先切換至 SINUMERIK 模式（參閱範例）。

範例

ISO 用語模式中的線性單節壓縮

程式碼	註解
N5 G290	；啟動 SINUMERIK 語言模式。
N10 COMPON	；COMPON 是西門子語言中的指令，會啟動壓縮程序功能以多項式單節取代接下來的線性單節，並且讓路徑長度盡可能的長。
N15 G291	；啟動 ISO 語言模式。
N20 G01 X100 Y100 F1000	；由於 COMPON 已在 SINUMERIK 模式中啟用了，則甚至 ISO 模式的線性單節也可以利用此單節壓縮功能。
...	

使用者材料移除程式 19

19.1 材料移除的支援函數

為材料移除提供預先程式設計的材料移除程式。 此外，您可使用下列函數產生自己的材料移除程式：

- 產生輪廓表（CONTPRON）
- 產生編碼輪廓表（CONTDCON）
- 停用輪廓準備（EXECUTE）
- 決定兩輪廓元素間的交點（INTERSEC）
  （僅適用使用 CONTPRON 產生的表）
- 按單節執行表的輪廓元素（EXECTAB）
  （僅適用使用 CONTPRON 產生的表）
- 計算圓弧資料（CALCDAT）

說明

該函數不僅可用在材料移除，也可用在所有地方。

條件

呼叫 CONTPRON 或 CONTDCON 函數前，需完成下列項目：

- 必須逼近允許無碰撞加工的起點。
- 需以 G40 停用切削半徑補正。

19.2 產生輪廓表（CONTPRON）

CONTPRON 啟動輪廓準備。 不執行之後呼叫的 NC 單節，而是分隔成個別的移動並存入輪廓表。 每個輪廓元素對應於輪廓表二維陣列中的一列。 傳回凸紋切削的次數。

語法

啟動輪廓準備：
CONTPRON（<輪廓表>，<加工類型>，<凸紋切削>，
<加工方向>）

停用輪廓準備並回到正常執行模式：
EXECUTE(<ERROR>)

請參閱「停用輪廓準備（EXECUTE）(頁 762)」

含義

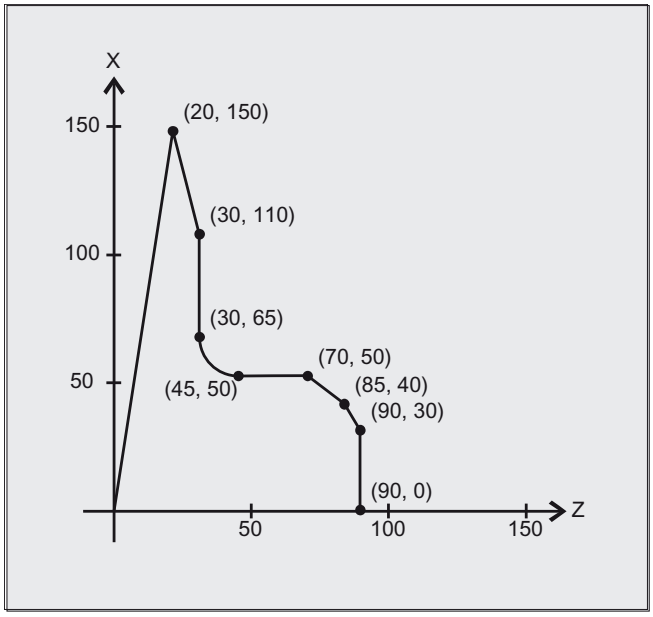
CONTPRON：	預先定義的程序，用於啟動輪廓準備以產生輪廓表		
<輪廓表>：	輪廓表名稱		
<加工類型>：	加工類型的參數		
	類型：	CHAR	
	值：	「G」：	縱向車削： 內部加工
		「L」：	縱向車削： 外部加工
		"N"：	平面車削： 內部加工
		「P」：	平面車削： 外部加工
<凸紋切削>：	發生凸紋切削元件數量的結果變數		
	類型：	INT	
<加工方向>：	加工方向之參數		
	類型：	INT	
	值：	0	輪廓準備，向前（預設值）
		1	輪廓準備，雙向

範例 1

產生輪廓表，使用：

- 名稱「KTAB」
- 最多 30 個輪廓元素（圓弧、直線）

- 發生凸紋切削元件數量的一變數
- 錯誤訊息的一變數



NC 程式：

程式碼	註解
N10 DEF REAL KTAB[30,11]	含 KTAB 名稱及最多 30 個輪廓元素、參數值 11（表欄位數）的輪廓表，為固定數量。
N20 DEF INT ANZHINT	；使用名稱 ANZHINT 的凸紋切削元件數量之變數。
N30 DEF INT ERROR	；錯誤反饋訊號變數（0 = 無錯誤，1 = 錯誤）。
N40 G18	
N50 CONTPRON（KTAB，"G"，ANZHINT）	；啟動輪廓準備。
N60 G1 X150 Z20	；N60 至 N120： 輪廓說明
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE（ERROR）	；停止填寫輪廓表，切換為正常程式模式。
N140 ...	；繼續處理該表。

輪廓表 KTAB：

索引 直線	欄									
(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

欄位內容說明：

(0) 下一個輪廓元素的指標（指向該欄位之列數）

(1) 前一個輪廓元素的指標

(2) 將動作的輪廓模式編碼

X = abc 的可能值

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) 輪廓元素起點

（3）=橫座標、（4）=目前平面的縱座標

(5), (6) 輪廓元素起點

（5）=橫座標、（6）=目前平面的縱座標

(7) 最大 / 最小指示碼： 辨識出本機輪廓中的最大值及最小值。

(8) 輪廓元素及橫座標（用於縱向加工）或縱座標（用於平面切削）間的最大值。角度依已程式設計的加工類型而定。

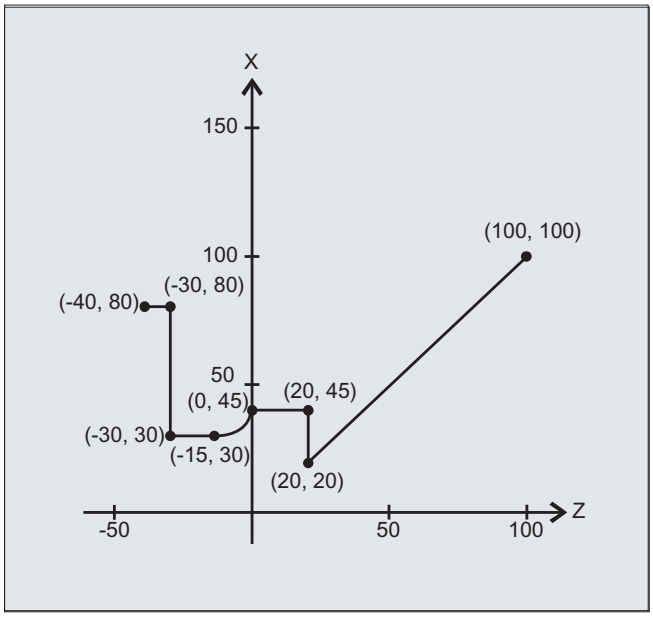
(9), (10) 輪廓元素的中心點座標，若其為圓弧單節。

（9） = 橫座標，（10） = 縱座標

範例 2

產生輪廓表，使用

- 名稱 KTAB
- 最多 92 個輪廓元素（圓弧、直線）
- 操作模式： 縱向車削、外部加工
- 準備，向前及向後



NC 程式：

程式碼	註解
N10 DEF REAL KTAB[92,11]	；含 KTAB 名稱及最多 92 個輪廓元素、參數值 11 的輪廓表，為固定數量。
N20 DEF CHAR BT="L"	；CONTPRON 模式： 縱向車削、外部加工
N30 DEF INT HE=0	；凸紋切削元件數量 = 0
N40 DEF INT MODE=1	；準備，向前及向後
N50 DEF INT ERR=0	；錯誤反饋訊號
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON（KTAB，BT，HE，MODE）	；啟動輪廓準備。

程式碼	註解
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）	；停止填寫輪廓表，切換為正常程式模式。
...	

輪廓表 KTAB：

完成輪廓準備後，可雙向使用輪廓。

索引	欄										
直線	(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

欄位內容說明及第 0、1、6、8、83、85 及 91 行之註解。

適用範例 1 中對欄位內容的說明。

永遠在表列 0：

1） 前置： 第 n 行包含輪廓結尾（向前）

2） 後續： 第 n 行為輪廓表結尾（向前）

一旦輪廓元素中每個皆向前：

3） 前置： 輪廓開始（向前）

4） 後續： 輪廓結尾（向前）

永遠在輪廓表結束（向前）+1 行：

5） 前置： 凸紋切削次數（向前）

6） 後續： 凸紋切削次數（向後）

一旦輪廓元素中每個皆向後：

7） 後續： 輪廓結尾（向後）

8） 前置： 輪廓開始（向後）

永遠在表的最後一行：

9） 前置： 第 n 行為輪廓表開始（向後）

10） 後續： 第 n 行包含輪廓開始（向後）

其他資訊

認可的移動指令、座標系統

可在輪廓程式設計中使用下列 G 指令：

- G 群組 1： G0、G1、G2、G3

除此之外，下列也可能：

- 倒圓角與倒角
- 圓弧程式設計使用 CIP 及 CT

曲線、多項式及螺紋函數會產生錯誤。

在 CONTPRON 及 EXECUTE 間不可透過啟動框架來變更座標系統。 此限制亦適用於在 G70 及 G71 或 G700 及 G710 間的變更。

若準備輪廓表時，以 GEOAX 更換幾何軸，會發生警報。

凸紋切削元件

可在副程式或個別單節中，執行個別凸紋切削元件的輪廓說明。

材料移除獨立於已程式設計的輪廓方向

以 CONTPRON 展開輪廓準備，如此在呼叫後，可用輪廓表獨立於已程式設計的方向。

19.3 產生編碼輪廓表（CONTDCON）

以 CONTDCON 啟動輪廓準備時，下列呼叫的 NC 單節會以編碼形式儲存於六欄輪廓表，以最佳化記憶體使用。每個輪廓元素對應於輪廓表的一列。熟悉下述編碼規則後，例如，您可從表列中結合 DIN 代碼程式作為循環。輸出點的資料，用編號 0 存入表列中。

語法

啟動輪廓準備：
CONTDCON（<輪廓表>，<加工方向>）

停用輪廓準備並回到正常執行模式：
EXECUTE(<ERROR>)

請參閱「停用輪廓準備（EXECUTE）(頁 762)」

含義

CONTDCON：	預先定義的程序，用於啟動輪廓準備以產生編碼輪廓表		
<輪廓表>：	輪廓表名稱		
<加工方向>：	加工方向的參數		
	類型：	INT	
	值：	0	根據輪廓單節順序的輪廓準備（預設值）
		1	不允許

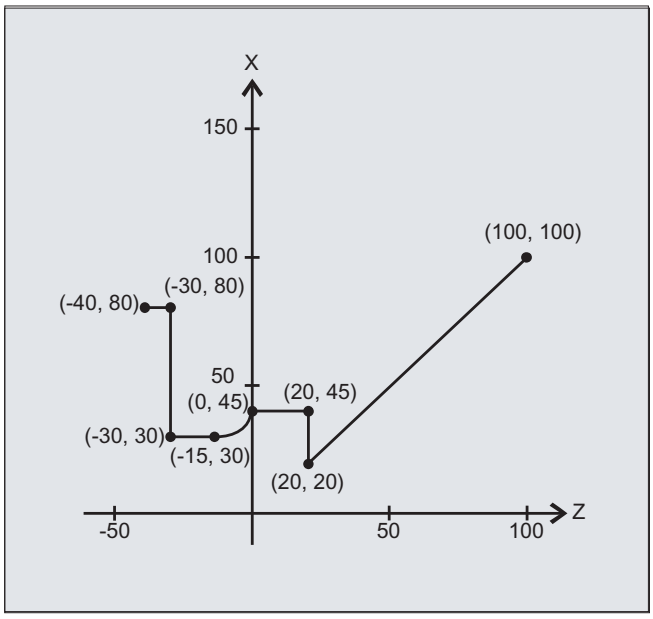
說明

在程式區段中欲包含至該表，為 CONTDCON 所認可的 G 指令，比為 CONTPRON 認可的還要廣泛。此外，為各輪廓區段儲存進給率及進給率類型。

範例

產生輪廓表，使用：

- 名稱「KTAB」
- 輪廓元素（圓弧、直線）
- 操作模式：車削
- 加工方向：向前



NC 程式：

程式碼	註解
N10 DEF REAL KTAB[9,6]	；含名稱 KTAB 及 9 個列表的輪廓表。允許 8 個輪廓集。參數值 6（表中欄位數）為固定尺寸。
N20 DEF INT MODE = 0	；加工方向的變數。標準值 0：僅在輪廓已程式設計的方向中。
N30 DEF INT ERROR = 0	；錯誤反饋訊號變數。
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON （KTAB，MODE）	；呼叫輪廓準備（可忽略 MODE）。
N110 G1 Z20 X20 F200	；輪廓說明。
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）	；停止填寫輪廓表，切換為正常程式模式。
...	

輪廓表 KTAB：

	欄位索引					
	0	1	2	3	4	5

行索引	輪廓模式	終點 橫座標	終點 縱座標	中心點 橫座標	中心點 縱座標	進給率
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

欄位內容說明：

起點的 0 行編碼：

欄 0： 10^0（個位數）：G0 = 0
10^1（十位數）：G70 = 0，G71 = 1，G700 = 2，G710 = 3
欄 1： 起點橫座標
欄 2： 起點縱座標
欄 3-4： 0
欄 5： 表中最後一個輪廓片段的行索引

1 至 n 行：**輪廓片段**項目

欄 0： 10^0（個位數）：G0 = 0，G1 = 1，G2 = 2，G3 = 3
10^1（十位數）：G70 = 0，G71 = 1，G700 = 2，G710 = 3
10^2（百位數）：G90 = 0，G91 = 1
10^3（千位數）：G93 = 0，G94 = 1，G95 = 2，G96 = 3
10^4（萬位數）：G60 = 0，G44 = 1，G641 = 2，G642 = 3
10^5（十萬位數）：G9 = 1
欄 1： 終點橫座標
欄 2： 終點縱座標
欄 3： 圓弧插補的中心點橫座標
欄 4： 圓弧插補的中心點縱座標
欄 5： 進給率

其他資訊

認可的移動指令、座標系統

可在輪廓程式設計中，使用下列 G 群組及 G 指令：

G 群組 1： G0、G1、G2、G3

G 群組 10： G60，G64，G641，G642

G 群組 11： G9

G 群組 13： G70，G71，G700，G710

G 群組 14： G90，G91

G 群組 15： G93，G94，G95，G96，G961

除此之外，下列也可能：

- 倒圓角與倒角
- 圓弧程式設計使用 CIP 及 CT

曲線、多項式及螺紋函數會產生錯誤。

在 CONTDCON 及 EXECUTE 間不可透過啟動框架來變更座標系統。此限制亦適用於在 G70 及 G71 或 G700 及 G710 間的變更。

若準備輪廓表時，以 GEOAX 更換幾何軸，會發生警報。

加工方向

使用 CONTDCON 產生的輪廓表，可用於輪廓已程式設計的方向上之材料移除。

19.4 決定兩輪廓元素間的交點（INTERSEC）

INTERSEC 從使用 CONTPRON 產生的輪廓表，決定兩個正常化輪廓元素間的交點。

語法

<狀態>=INTERSEC（<輪廓表_1>[<輪廓元素_1>]，
<輪廓表_2>[<輪廓元素_2>]，<交叉點>，<加工類型>）

含義

INTERSEC：	為預先定義函數，從 CONTPRON 產生的輪廓表決定兩輪廓元素間交點		
<狀態>：	交點狀態的變數		
	類型：	BOOL	
	值：	TRUE	找到交點
		FALSE	未找到交點
<輪廓表_1>：	第一輪廓表名稱		
<輪廓元素_1>：	第一輪廓表輪廓元素的數量		
<輪廓表_2>：	第二輪廓表名稱		
<輪廓元素_2>：	第二輪廓表輪廓元素的數量		
<交點>：	生效平面中的交點座標（G17 / G18 / G19）		
	類型：	REAL	
<加工類型>：	加工類型的參數		
	類型：	INT	
	值：	0	有效平面中含參數 2 的交點計算點（標準值）
		1	交點計算點獨立於傳輸平面

說明

請注意，變數需在使用前先定義。

傳輸輪廓時，需觀察以 CONTPRON 定義之值：

參數	含義
2	移動輪廓模式的編碼
3	輪廓起點橫座標
4	輪廓起點縱座標
5	輪廓終點橫座標
6	輪廓終點縱座標
9	橫座標的中心點座標（僅用於圓弧輪廓）
10	縱座標的中心點座標（僅用於圓弧輪廓）

範例

計算表 TABNAME1 中輪廓元素 3，與表 TABNAME2 中輪廓元素 7 的交點。 有效平面上的交點座標會儲存於變數 ISCOORD（第一元素 = 橫座標，第二元素 = 縱座標）。 若無交點存在，則程式跳躍至 NOCUT（未找到交點）。

程式碼	註解
`DEF REAL TABNAME1[12,11]`	；輪廓表 1
`DEF REAL TABNAME2[10,11]`	；輪廓表 2
`DEF REAL ISCOORD [2]`	；交點座標變數。
`DEF BOOL ISPOINT`	；交點狀態變數。
`DEF INT MODE`	；加工類型變數。
…	
`MODE=1`	；獨立於有效平面的計算。
`N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE)`	；輪廓元素呼叫交點。
`N20 IF ISPOINT==FALSE GOTOF NOCUT`	；跳躍至 NOCUT。
…	

19.5 按單節執行表的輪廓元素（EXECTAB）

使用 EXECTAB 指令，就可按單節執行表的輪廓元素，例如使用 CONTPRON 產生的元素。

語法

EXECTAB（<輪廓表>[<輪廓元素>]）

含義

EXECTAB：	預先定義的程序，用於執行輪廓元素
<輪廓表>：	輪廓表名稱
<輪廓元素>：	輪廓元素的數量

範例

需按單節執行表 KTAB 中的輪廓元素 0 至 2。

程式碼	註解
N10 EXECTAB（KTAB[0]）	；移動表 KTAB 的元素 0。
N20 EXECTAB（KTAB[1]）	；移動表 KTAB 的元素 1。
N30 EXECTAB（KTAB[2]）	；移動表 KTAB 的元素 2。

19.6 計算圓弧資料（CALCDAT）

使用 CALCDAT，可由圓上已知的三或四個點，計算出半徑及圓心點座標。 需為不同指定點。

四個點並未直接位於圓上，算出圓心點及半徑的平均值。

說明

平均的計算規範

圓弧計算執行四次：

1. 使用圓點 1、2、3
2. 使用圓點 1、2、4
3. 使用圓點 1、3、4
4. 使用圓點 2、3、4

圓心點座標橫座標及縱座標值的計算方式，為加總四個圓弧計算的橫座標及縱座標值，然後再除以四。

半徑的計算方式為開方圓弧計算的四個半徑總和，然後再將結果乘以 0.5。

語法

<狀態>=CALCDAT（<圓點>[<數量>，<類型>]，<數量>，<結果>）

含義

CALCDAT：	預先定義的函數，由三或四個點計算圓的半徑及中心點座標		
<狀態>：	圓弧計算狀態的變數		
	類型：	BOOL	
	值：	TRUE	指定點位於圓上。
		FALSE	指定點**並未**位於圓上。
<圓點>[]：	使用參數指定圓弧點的變數		
	<編號>：	圓弧點的數量（3 或 4）	
	<類型>：	座標資料類型， 例如，2 表示 2 點座標	
<編號>：	用於計算（3 或 4）的點數量之參數		

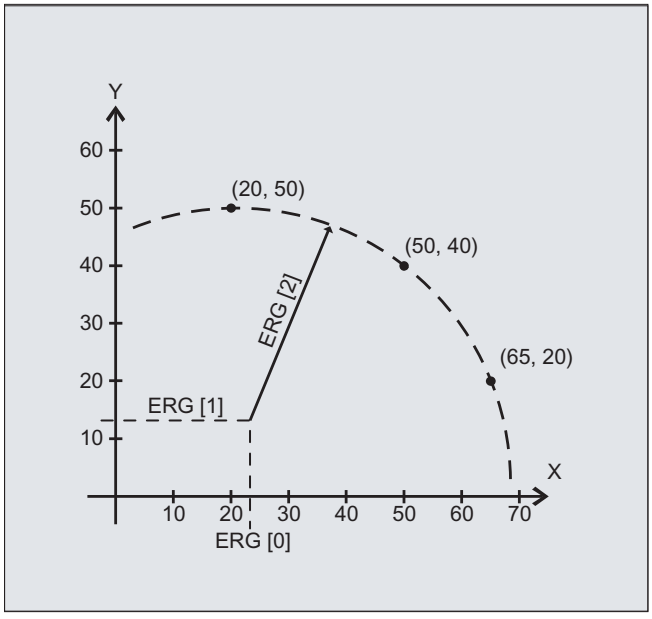
<結果>[3]：	結果的變數： 中心點座標及半徑	
	0	中心點座標： 橫座標值
	1	中心點座標： 縱座標值
	2	半徑

說明

請注意，變數需在使用前先定義。

範例

應使用三個點，以決定三個點是否位在一圓弧區段上。



程式碼	註解
N10 DEF REAL PT[3,2]=(20,50,50,40,65,20)	；指定圓點的變數。
N20 DEF REAL RES[3]	；結果變數。
N30 DEF BOOL STATUS	；狀態變數。
N40 STATUS=CALCDAT（PKT，3，ERG）	；呼叫判定的圓資料。
N50 IF STATUS == FALSE GOTOF ERROR	；跳躍至錯誤。

19.7 停用輪廓準備（EXECUTE）

EXECUTE 可停用輪廓準備，同時系統會回到正常執行模式。

語法

```
EXECUTE(<ERROR>)
```

含義

EXECUTE：	預先定義的程序，用於終止輪廓準備	
<ERROR>：	錯誤反饋訊號變數	
	類型：	INT
	變數值指出輪廓是否能以無錯誤做好準備：	
	0	錯誤
	1	無錯誤

範例

程式碼

```
...
N30 CONTPRON(...)
N40 G1 X... Z...
...
N100 EXECUTE(...)
...
```

外部程式設計循環 20

20.1 技術循環

20.1.1 簡介

內容

本節說明車削、銑削及研磨技術的循環。

結構

循環說明的結構如下：

- **語法**
  傳輸參數的循環名稱及呼叫順序
- **參數**
  說明個別參數的表格

參數說明

以下資料於表中針對參數指定：名稱、說明、值域及對其他參數的依賴性。

重新編譯外部產生的循環呼叫時，可使用參考欄的畫面表單參數，更輕鬆地找出在控制程式設計的值。

「僅適用介面」參數

表格中某些參數標記為「僅適用介面」。這些參數與循環的操作無關。只有在完全重新編譯循環呼叫時才會需要這些參數。若未程式設計則循環仍可以重新編譯；欄位將以顏色辨識並且必須在遮罩中完成。

「已保留」參數

稱之為「已保留」的參數必須以數值 0 或逗號程式設計，這樣一來以下的呼叫參數配置才能符合內部循環參數。例外：具有數值 "" 或逗號的字串參數。

位置樣式上的重複循環

鑽孔及銑削循環可以在位置樣式上重複（模態呼叫）。於此情況下，MCALL 應寫在循環之前的同一行內，例如 MCALL CYCLE83(...)。

說明

如果特定參數（例如 <_VARI>, <_GMODE>, <_DMODE>, <_AMODE>）已經間接程式設計為參數，重新編譯時會開啟畫面表單，但無法儲存，因為定義的選擇欄位並無唯一的配置。

20.1.2 技術專屬概觀

以下概觀表列出所有可用的外部程式設計技術循環及各自指定的技術：

技術	技術循環
鑽孔	• CYCLE81 - 鑽孔、定心 (頁 815) • CYCLE82 - 鑽孔、逆鏜孔 (頁 816) • CYCLE85－鉸孔 (頁 827) • CYCLE86－鏜孔 (頁 829) • CYCLE83－深孔鑽 (頁 819) • CYCLE830 - 深孔鑽 2 (頁 859) • CYCLE84－不搭配補正夾頭進行攻牙 (頁 823) • CYCLE840 - 搭配補正夾頭進行攻牙 (頁 869) • CYCLE78－鑽孔螺紋銑削 (頁 809) • CYCLE802－任意位置 (頁 855) • HOLES1 - 排孔 (頁 766) • CYCLE801 - 網格或框架 (頁 853) • HOLES2 - 孔洞圓 (頁 766)
車削	• CYCLE951－材料移除 (頁 882) • CYCLE930－槽 (頁 875) • CYCLE940 - 底切形狀 (頁 878) • CYCLE99－螺紋車削 (頁 841) • CYCLE98－螺紋鍊 (頁 834) • CYCLE92－切離 (頁 830)
輪廓車削	• CYCLE62 - 輪廓呼叫 (頁 791) • CYCLE952 - 輪廓切槽 (頁 885)

技術	技術循環
銑削	• CYCLE61－平面銑削 (頁 788) • POCKET3 - 銑削矩形凹槽 (頁 769) • POCKET4 - 銑削圓形凹槽 (頁 773) • CYCLE76 - 矩形島嶼銑削 (頁 804) • CYCLE77 - 圓弧島嶼銑削 (頁 806) • CYCLE79－多邊緣 (頁 812) • SLOT1－長形槽 (頁 777) • SLOT2－環槽 (頁 780) • CYCLE899－銑削開孔槽 (頁 872) • LONGHOLE - 細長孔洞 (頁 783) • CYCLE70－螺紋銑削 (頁 797) • CYCLE60 - 刻字循環 (頁 786)
輪廓銑削	• CYCLE62 - 輪廓呼叫 (頁 791) • CYCLE72－路徑銑削 (頁 800) • CYCLE63－銑削輪廓挖槽 (頁 792) • CYCLE64－預鑽孔輪廓挖槽 (頁 796)
研磨	• CYCLE495 - 形狀修整 (頁 847) • CYCLE435 - 設定修整器座標系統 (頁 846) • CYCLE4071 - 在反轉點進給的縱向研磨 (頁 893) • CYCLE4072 - 於反轉點進給的縱向研磨和取消訊號 (頁 895) • CYCLE4073 - 連續進給之縱向研磨 (頁 898) • CYCLE4074 - 連續進給的縱向研磨和取消訊號 (頁 899) • CYCLE4075 - 在反轉點進給的表面研磨 (頁 903) • CYCLE4077 - 於反轉點進給的表面研磨和取消訊號 (頁 905) • CYCLE4078 - 連續進給之表面研磨 (頁 908) • CYCLE4079 - 間歇進給之表面研磨 (頁 910)
其他	• CYCLE800 - 旋轉 (頁 849) • CYCLE832 - 高速設定 (頁 865)
全部	• GROUP_BEGIN - 程式單節開始處 (頁 913) • GROUP_END - 程式單節結束 (頁 914) • GROUP_ADDEND - 嘗試切削添加 (頁 914)

20.1.3 HOLES1 - 排孔

語法

```
HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	X0	<SPCA>	REAL	沿著第一軸排孔的參考點（絕對）		
2	Y0	<SPCO>	REAL	沿著第二軸排孔的參考點（絕對）		
3	α0	<STA1>	REAL	基本旋轉角度（至第一軸的角度）		
4	L0	<FDIS>	REAL	從第 1 孔到參考點的距離		
5	L	<DBH>	REAL	孔間距		
6	N	<NUM>	INT	孔的數量		
7		<_VARI>	INT	已保留		
8		<_UMODE>	INT	已保留		
9		<_HIDE>	STRING [200]	隱藏位置 • 最多 198 個字元 • 連續位置數的規格，例如「1,3」（位置 1 與 3 不執行）		
10		<_NSP>	INT	已保留		
11		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

20.1.4 HOLES2 - 孔洞圓

語法

```
HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

參數

號碼	參數遮罩	參數 內側	資料類型	含義			
1	X0	<CPA>	REAL	沿著第一軸孔圓的中心點（絕對） 第一軸的參考點			（供 XY 之用） （供 XA、YB、ZC 之用）
2	Y0	<CPO>	REAL	沿著第二軸孔圓的中心點（絕對） 第二軸的參考點			（供 XY 之用） （供 XA、YB、ZC 之用）
3	R	<RAD>	REAL	孔圓的半徑			（供 XY 之用）
4	α0	<STA1>	REAL	起始角度 或第一旋轉軸位置			（供 XY 之用） （供 XA、YB、ZC 之用）
5	α1	<INDA>	REAL	前進角（僅適用於截圓）			（供 XY、XA、YB、ZC 之用）
					< 0 =	順時鐘方向	
					> 0 =	逆時針	
6	N	<NUM>	INT	位置數			

號碼	參數遮罩	參數 內側	資料類型	含義			
7		<_VARI>	INT	加工類型			
				單位：	已保留		
				十位數：	定位類型		
					0 =	逼近位置 - 線性	
					1 =	逼近位置 - 圓弧路徑	
				百位數：	已保留		
				千位數：	圓弧模式		
					0 =	相容性模式，若 INDA = 0 則為全圓，INDA <> 0 則為節圓	
					1 =	全圓	
					2 =	節圓	
				萬位數：	有旋轉軸的位置樣式		
					0 =	XY（無旋轉軸）	（供 XY 之用）
					1 =	XA（X 軸及旋轉軸繞著 X）	（僅供 XA 之用）
					2 =	YB（Y 軸及旋轉軸繞著 Y）	（僅供 YB 之用）
					3 =	ZC（Z 軸及旋轉軸繞著 C）	（僅供 ZC 之用）
				百萬位數 + 十萬位數：	偏移量（繞著相同軸的數個旋轉軸；若索引過大，則為第一軸）		
					00 =	1. A、B 或 C 軸	
					01 =	2. A、B 或 C 軸	
					...		
					10 =	20. A、B 或 C 軸	
8		<_UMODE>	INT	已保留			
9		<_HIDE>	STRING [200]	已保留			
10		<_NSP>	INT	已保留			

號碼	參數遮罩	參數 內側	資料類型	含義		
11		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

20.1.5 POCKET3 - 銑削矩形凹槽

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	RP	<_RTP>	REAL	回退平面（絕對）
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<_DP>	REAL	腔深度（絕對 / 增量），請參閱 <_AMODE>
5	L	<_LENG>	REAL	腔長度（增量，輸入時帶正負號）
6	W	<_WID>	REAL	腔寬度（增量，輸入時帶正負號）
7	R	<_CRAD>	REAL	腔的角半徑
8	X0	<_PA>	REAL	參考點第一軸（絕對）
9	YO	<_PO>	REAL	參考點第二軸（絕對）
10	α0	<_STA>	REAL	旋轉角，縱向軸 (L) 與第一軸的夾角
11	DZ	<_MID>	REAL	最大深度進給
12	UXY	<_FAL>	REAL	精加工允差量，平面

號碼	參數遮罩	內部參數	資料類型	含義		
13	UZ	<_FALD>	REAL	精加工允差，深度		
14	F	<_FFP1>	REAL	平面進給率		
15	FZ	<_FFD>	REAL	深度進給速率		
16		<_CDIR>	INT	銑削方向：	0 =	下切
					1 =	上切
17		<_VARI>	INT	加工類型		
				單位：		
					1 =	粗加工
					2 =	精加工
					4 =	邊緣精加工
					5 =	倒角
				十位數：		
					0 =	預先鑽孔，以 G0 進給
					1 =	垂直，以 G1 進給
					2 =	螺旋
					3 =	沿凹槽縱向軸擺動
				百位數：	已保留	
18	DXY	<_MIDA>	REAL	最大平面進給，單位請參閱 <_AMODE>		
19	L1	<_AP1>	REAL	預加工長度（增量）		
20	W1	<_AP2>	REAL	預加工寬度（增量）		
21	AZ	<_AD>	REAL	預加工深度（增量）		
22	ER	<_RAD1>	REAL	螺旋插入的螺旋路徑半徑		
	EW			擺盪的最大插入角		
23	EP	<_DP1>	REAL	螺旋插入的螺旋螺距		
24		<_UMODE>	INT	已保留		
25	FS	<_FS>	REAL	倒角寬度（增量）		
26	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		

號碼	參數遮罩	內部參數	資料類型	含義		
27		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工/僅計算起點	
					0 =	相容性模式
					1 =	正常加工
				千位數：	透過中心/角隅尺寸標註	
					0 =	相容性模式
					1 =	透過中心尺寸標註
					2 =	角點的尺寸標註，腔位置 +LENG/+WID
					3 =	角點的尺寸標註，腔位置 -LENG/+WID
					4 =	角點的尺寸標註，腔位置 +LENG/-WID
					5 =	角點的尺寸標註，腔位置 -LENG/-WID
				萬位數：	完整加工/重新加工	
					0 =	相容性模式（和之前一樣處理 <_AP1>, <_AP2> 及 <_AD> ）
					1 =	完整加工
					2 =	後加工

號碼	參數遮罩	內部參數	資料類型	含義		
28		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	進給率類型：平面及深度進給率的 G 群組 (G94/G95)	
					0 =	相容性模式
					1 =	循環呼叫前的 G 指令平面及深度進給率的 G94/G95 相同
				百位數：	---	已保留
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單
29		<_AMODE>	INT	替代模式		
				單位：	腔深度 (Z1)	
					0 =	絕對（相容性模式）
					1 =	增量
				十位數：	平面進給的單位 (DXY)	
					0 =	mm
					1 =	刀具直徑 %
				百位數：	倒角的插入深度 (ZFS)	
					0 =	絕對值
					1 =	增量

20.1.6 POCKET4 - 銑削圓形凹槽

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<_RTP>	REAL	回退平面（絕對）		
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）		
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1	<_DP>	REAL	腔深度（絕對 / 增量），請參閱 <_AMODE>		
5	∅	<_CDIAM>	REAL	腔直徑或半徑，請參閱 <_DMODE>		
6	X0	<_PA>	REAL	參考點第一軸（絕對）		
7	Y0	<_PO>	REAL	參考點第二軸（絕對）		
8	DZ	<_MID>	REAL	最大深度設定，請參閱 <_VARI> = 依平面 最大螺旋設定，請參閱 <_VARI> = 螺旋		
9	UXY	<_FAL>	REAL	精加工允差量，平面		
10	UZ	<_FALD>	REAL	精加工允差，深度		
11	F	<_FFP1>	REAL	表面加工的進給率		
12	FZ	<_FFD>	REAL	深度進給速率		
13		<_CDIR>	INT	銑削方向	0 =	下切
					1 =	上切

號碼	參數遮罩	內部參數	資料類型	含義		
14		<_VARI>	INT	加工類型		
				單位：	加工	
					1 =	粗加工
					2 =	精加工
					4 =	邊緣精加工
					5 =	倒角
				十位數：	進給類型（粗加工及精加工）	
					0 =	預先鑽孔，以 G0 進給（腔先預加工）
					1 =	垂直，以 G1 進給
					2 =	螺旋
				百位數：	已保留	
				千位數：		
					0 =	按平面
					1 =	螺旋
15	DXY	<_MIDA>	REAL	最大平面進給，請參閱 <_AMODE>, 0 = 0.8 x 刀具直徑		
16	∅	<_AP1>	REAL	預加工的直徑/半徑（增量）		
17	AZ	<_AD>	REAL	預加工深度（增量）		
18	ER	<_RAD1>	REAL	螺旋插入的螺旋路徑半徑		
19	EP	<_DP1>	REAL	螺旋路徑的插入螺旋螺距		
20		<_UMODE>	INT	已保留		
21	FS	<_FS>	REAL	倒角寬度（增量）		
22	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		

號碼	參數遮罩	內部參數	資料類型	含義		
23		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	加工/計算起點	
					0 =	相容性模式
					1 =	正常加工
				千位數：	已保留	
				萬位數：	完整加工/重新加工	
					0 =	相容性模式（和之前一樣處理 <_AP1> 及 <_AD> ）
					1 =	完整加工
					2 =	後加工

號碼	參數遮罩	內部參數	資料類型	含義		
24		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	進給率類型：平面及深度進給率的 G 群組 (G94/G95)	
					0 =	相容性模式
					1 =	循環呼叫前的 G 指令平面及深度進給率的 G94/G95 相同
				百位數：		
					0 =	相容性模式（輸入 <_CDIAM>/<_AP1> 作為半徑）
					1 =	輸入 <_CDIAM>/<_AP1> 作為半徑
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單

號碼	參數遮罩	內部參數	資料類型	含義		
25		<_AMODE>	INT	替代模式		
				單位：	腔深度 (Z1)	
					0 =	絕對（相容性模式）
					1 =	增量
				十位數：	進給寬度的單位 (DXY)	
					0 =	mm
					1 =	刀具直徑 %
				百位數：	倒角的插入深度 (ZFS)	
					0 =	絕對值
					1 =	增量

20.1.7 SLOT1－長形槽

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	RP	<RTP>	REAL	回退平面（絕對）
2	Z0	<RFP>	REAL	刀具軸參考點（絕對）
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<_DP>	REAL	槽深（絕對）
5		<_DPR>	REAL	相對於 Z0 的槽深（增量）（輸入時不帶正負號）
6		<NUM>	INT	槽數 = 1
7	L	<LENG>	REAL	槽長
8	W	<WID>	REAL	槽寬
9	X0	<_CPA>	REAL	平面第一軸的參考點
10	Y0	<_CPO>	REAL	平面第二軸的參考點

號碼	參數遮罩	內部參數	資料類型	含義		
11		<RAD>	REAL	已保留		
12	α	<STA1>	REAL	旋轉角度		
13		<INDA>	REAL	已保留		
14	FZ	<FFD>	REAL	深度進給速率		
15	F	<FFP1>	REAL	進給率		
16	DZ	<_MID>	REAL	最大深度進給		
17		<CDIR>	INT	銑削方向	0 =	下切
					1 =	上切
18	UXY	<_FAL>	REAL	平面或槽稜邊的精加工允差		
19		<VARI>	INT	加工類型		
				單位：		
					0 =	已保留
					1 =	粗加工
					2 =	精加工
					4 =	稜邊精加工（僅加工稜邊）
					5 =	倒角
				十位數：	逼近	
					0 =	預先鑽孔，以 G0 進給（插槽先預加工）
					1 =	垂直，以 G1 進給
					2 =	螺旋
					3 =	振盪
				百位數：	已保留	
20	DZF	<_MIDF>	REAL	已保留		
21	FF	<FFP2>	REAL	已保留		
22	SF	<SSF>	REAL	已保留		
23	UZ	<_FALD>	REAL	精加工允差，深度		
24	ER	<_STA2>	REAL	螺旋插入的螺旋路徑半徑		
	EW			擺盪的最大插入角		
25	EP	<_DP1>	REAL	螺旋每轉的插入深度		
26		<_UMODE>	INT	已保留		

號碼	參數遮罩	內部參數	資料類型	含義		
27	FS	<_FS>	REAL	倒角時的倒角寬度（增量）		
28	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		
29		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工或僅計算起點	
					1 =	正常加工
				千位數：	參考點的尺寸標註，槽長	
					0 =	中心
					1 =	內側左手邊 +L
					2 =	內側右手邊 -L
					3 =	左手稜邊 +L
					4 =	右手稜邊 -L
30		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	已保留	
				百位數：	已保留	
				千位數：	軟體版本識別	
					1 =	函數副檔名 SLOT1
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單

號碼	參數遮罩	內部參數	資料類型	含義		
31		<_AMODE>	INT	替代模式		
				單位：	最終深度 Z1（絕對/增量）	
					0 =	相容性
					1 =	Z1（增量）
					2 =	Z1（絕對）
				十位數：	已保留	
				百位數：	倒角的插入深度 ZFS	
					0 =	ZFS（絕對）
					1 =	ZFS（函數）

說明

提供較早軟體版本所沒有的新功能給循環使用。 因此畫面表單的特定參數（<NUM>, <RAD>, <INDA>）將不再顯示。位置圖樣上的多重槽可以使用 "MCALL" 程式設計，並呼叫所要的位置圖樣，例如 HOLES2。

20.1.8 SLOT2－環槽

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	RP	<RTP>	REAL	回退平面（絕對）
2	Z0	<RFP>	REAL	刀具軸參考點（絕對）
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<_DP>	REAL	槽深（絕對）
5		<_DPR>	REAL	相對於 Z0 的槽深（增量）（輸入時不帶正負號）
6	N	<NUM>	INT	槽編號
7	α1	<AFSL>	REAL	凹槽缺口角度

號碼	參數遮罩	內部參數	資料類型	含義		
8	W	<WID>	REAL	槽寬		
9	X0	<_CPA>	REAL	參考點 = 圓弧的圓心，平面第一軸		
10	Y0	<_CPO>	REAL	參考點 = 圓弧的圓心，平面第二軸		
11	R	<RAD>	REAL	圓弧半徑		
12	α0	<STA1>	REAL	起始角度		
13	α2	<INDA>	REAL	增量角度		
14	FZ	<FFD>	REAL	深度進給速率		
15	F	<FFP1>	REAL	進給率		
16	DZ	<_MID>	REAL	最大深度進給		
17		<CDIR>	INT	銑削方向	0 =	下切
					1 =	上切
18	UXY	<_FAL>	REAL	平面或槽稜邊的精加工允差		
19		<VARI>	INT	加工類型		
				單位：		
					0 =	完整加工
					1 =	粗加工
					2 =	精加工
					3 =	邊緣精加工
					5 =	倒角
				十位數：		
					0 =	使用 G0 行的中間位置
					1 =	圓形路徑上的中間定位
				百位數：	已保留	
				千位數：		
					0 =	相容性模式，若 <INDA> = 0 則為全圓，<INDA> <> 0 則為節圓
					1 =	全圓
					2 =	節圓
20	DZF	<_MIDF>	REAL	已保留		
21		<FFP2>	REAL	已保留		
22		<SSF>	REAL	已保留		

號碼	參數遮罩	內部參數	資料類型	含義		
23	FF	<_FFCP>	REAL	已保留		
24		<_UMODE>	INT	已保留		
25	FS	<_FS>	REAL	倒角寬度（增量）		
26	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		
27		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工或僅計算起點	
					0 =	相容性模式
					1 =	正常加工
28		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	已保留	
				百位數：	已保留	
				千位數：	軟體版本識別	
					1 =	軟體版本 2.5 的 SLOT2 函數
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單

號碼	參數遮罩	內部參數	資料類型	含義		
29		<_AMODE>	INT	替代模式		
				單位：	最終深度 Z1（絕對/增量）	
					0 =	相容性
					1 =	Z1（增量）
					2 =	Z1（絕對）
				十位數：	已保留	
				百位數：	倒角的插入深度 ZFS	
					0 =	ZFS（絕對）
					1 =	ZFS（函數）

20.1.9 LONGHOLE - 細長孔洞

語法

```
LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>,
<_VARI>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	RP	<RTP>	REAL	回退平面（絕對）
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<_DP>	REAL	長孔深度（絕對）
5		<_DPR>	REAL	相對於 Z0 的長孔深度（增量）（輸入時不帶正負號）
6		<NUM>	INT	長孔數 = 1
7	L	<LENG>	REAL	長孔的長度
8	X0	<_CPA>	REAL	參考點平面第一軸
9	Y0	<_CPO>	REAL	參考點平面第二軸
10		<RAD>	REAL	已保留
11	α0	<STA1>	REAL	旋轉角度
12		<INDA>	REAL	已保留

號碼	參數遮罩	內部參數	資料類型	含義		
13	FZ	<FFD>	REAL	深度進給速率		
14	F	<FFP1>	REAL	進給率		
15	DZ	<MID>	REAL	最大深度進給		
16		<_VARI>	INT	加工類型		
				單位：	進給類型	
					1 =	垂直，G1
					3 =	振盪
				百位數：	已保留	
17		<_UMODE>	INT	已保留		
18		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工或僅計算起點	
					0 =	相容性模式
					1 =	正常加工
				千位數：	參考點的尺寸標註，槽長	
					0 =	中心
					1 =	內側左手邊 +L
					2 =	內側右手邊 -L
					3 =	左手稜邊 +L
					4 =	右手稜邊 -L

號碼	參數遮罩	內部參數	資料類型	含義		
19		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	進給率類型：平面及深度進給率的 G 群組 (G94/G95)	
					0 =	相容性模式
					1 =	循環呼叫前的 G 指令平面及深度進給率的 G94/G95 相同
				百位數：	已保留	
				千位數：	軟體版本識別	
					1 =	函數副檔名 LONGHOLE（參考點的尺寸標註）
20		<_AMODE>	INT	替代模式		
				單位：	最終深度 Z1（絕對/增量）	
					0 =	相容性
					1 =	Z1（增量）
					2 =	Z1（絕對）

說明

提供較早軟體版本所沒有的新功能給循環使用。因此畫面表單的特定參數（<NUM>, <RAD>, <INDA>）將不再顯示。位置圖樣上的多重槽可以使用 "MCALL" 程式設計，並呼叫所要的位置圖樣，例如 HOLES2。

20.1.10 CYCLE60 - 刻字循環

語法

```
CYCLE60(<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1		<_TEXT>	STRING [200]	欲雕刻的文字（最多 100 個字元）
2	RP	<_RTP>	REAL	回退平面（絕對）
3	Z0	<_RFP>	REAL	刀具軸參考點（絕對）
4	SC	<_SDIS>	REAL	安全間距（加入到參考平面，輸入時不帶正負號）
5	Z1	<_DP>	REAL	深度（絕對），請參閱 <_AMODE>
6	Z1	<_DPR>	REAL	深度（增量），請參閱 <_AMODE>
7	X0	<_PA>	REAL	平面第一軸上的參考點（絕對）- 直角，請參閱 <_VARI>
	R			參考點，長度（半徑）- 極，請參閱 <_VARI>
8	Y0	<_PO>	REAL	平面第二軸上的參考點（絕對）- 直角，請參閱 <_VARI>
	α0			參考點，相對於第一軸的角度 - 極，請參閱 <_VARI>
9	α1	<_STA>	REAL	相對於第一軸的文字方向，文字行的角度，請參閱 <_VARI>
10	XM	<_CP1>	REAL	圖形文字中心，平面第一軸（絕對）- 直角，請參閱 <_VARI>
	LM			圓形文字的中心，相對於 WNP 的長度（半徑）- 極，請參閱 <_VARI>
11	YM	<_CP2>	REAL	圖形文字中心，平面第二軸（絕對）- 直角，請參閱 <_VARI>
	αM			圓形文字的中心，相對於第一軸的角度- 極，請參閱 <_VARI>
12	W	<_WID>	REAL	字元的高度（輸入時不含正負號）
13	DX1 DX2	<_DF>	REAL	字元之間的距離 / 整體寬度，請參閱 <_VARI>
	α2			缺口角度，請參閱 <_VARI>
14	FZ	<_FFD>	REAL	深度進給率，請參閱 <_DMODE>
15	F	<_FFP1>	REAL	表面加工的進給率

號碼	參數遮罩	內部參數	資料類型	含義	
16		<_VARI>	INT	加工（雕刻文字的對齊及參考點）	
				單位：	參考點
					0 = 直角
					1 = 極座標
				十位數：	文字對齊
					0 = 一行文字
					1 = 朝上指向圓弧上的文字
					2 = 朝下曲線圓弧上的文字
				百位數：	已保留
				千位數：	文字的參考點，水平
					0 = 向左
					1 = 中心
					2 = 向右
				萬位數：	文字的參考點，垂直
					0 = 下方
					1 = 中心
					2 = 靠上
				十萬位數：	文字長度
					0 = 字元間隔
					1 = 整體文字寬度（僅線性文字）
					2 = 缺口角度（僅供圓弧文字）
				百萬位數：	圓心
					0 = 直角（直角座標）
					1 = 極座標
				千萬位數：	反向文字
					0 = 相容性
					1 = 反向文字 ON
					2 = 反向文字 OFF
17		<_CODEP>	INT	寫入的代碼頁碼（目前僅 1252）	
18		<_UMODE>	INT	已保留	

號碼	參數遮罩	內部參數	資料類型	含義		
19		_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工/僅計算起點	
					0 =	相容性模式
					1 =	正常加工
20		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	進給率類型：平面及深度進給率的 G 群組 (G94/G95)	
					0 =	相容性模式
					1 =	循環呼叫前的 G 指令平面及深度進給率的 G94/G95 相同
21		<_AMODE>	INT	替代模式		
				單位：	最終深度（<_DP>, <_DPR>）	
					0 =	相容性
					1 =	增量（<_DPR>）
					2 =	絕對（<_DP>）

20.1.11 CYCLE61－平面銑削

語法

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<_RTP>	REAL	回退平面（絕對）		
2	Z0	<_RFP>	REAL	刀具軸參考點，毛胚高度（絕對）		
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1	<_DP>	REAL	精加工零件的高度（絕對 / 增量），請參閱 <_AMODE>		
5	X0	<_PA>	REAL	第一軸的角點 1（絕對）		
6	Y0	<_PO>	REAL	第二軸的角點 1（絕對）		
7	X1	<_LENG>	REAL	第一軸的角點 2（絕對 / 增量），請參閱 <_AMODE>		
8	Y1	<_WID>	REAL	第二軸的角點 2（絕對 / 增量），請參閱 <_AMODE>		
9	DZ	<_MID>	REAL	最大深度進給		
10	DXY	<_MIDA>	REAL	最大平面進給（單位請參閱 <_AMODE>）		
11	UZ	<_FALD>	REAL	精加工允差，深度		
12	F	<_FFP1>	REAL	機台加工進給速率		
13		<_VARI>	INT	加工類型		
				單位：	加工	
					1 =	粗加工
					2 =	精加工
				十位數：	加工方向	
					1 =	平行於第一軸，單一方向
					2 =	平行於第二軸，單一方向
					3 =	平行於第一軸，不同方向
					4 =	平行於第二軸，不同方向

號碼	參數遮罩	內部參數	資料類型	含義		
14		<_LIM>	INT	極限		
				單位：	限制第一軸負向	
					0 =	否
					1 =	是
				十位數：	限制第一軸正向	
					0 =	否
					1 =	是
				百位數：	限制第二軸負向	
					0 =	否
					1 =	是
				千位數：	限制第二軸正向	
					0 =	否
					1 =	是
15		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

號碼	參數遮罩	內部參數	資料類型	含義		
16		<_AMODE>	INT	替代模式		
				單位：	最終深度（<_DP>）	
					0 =	絕對值
					1 =	增量
				十位數：	平面進給的單位（<_MIDA>）	
					0 =	mm
					1 =	刀具直徑 %
				百位數：	已保留	
				千位數：	平面長度	
					0 =	增量
					1 =	絕對值
				萬位數：	平面寬度	
					0 =	增量
					1 =	絕對值

20.1.12 CYCLE62 - 輪廓呼叫

語法

```
CYCLE62(<_KNAME>,<_TYPE>, <_LAB1>, <_LAB2>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	PRG/CON	<_KNAME>	STRING [140]	輪廓名稱或子程式名稱不需要在 _TYPE = 2 編程設定		
2		<_TYPE>	INT	決定輪廓輸入		
					0 =	副程式
					1 =	輪廓名稱
					2 =	標記
					3 =	副程式中的標籤

號碼	參數遮罩	內部參數	資料類型	含義
3	LAB1	<_LAB1>	STRING[32]	標籤 1，輪廓的起點
4	LAB2	<_LAB2>	STRING[32]	標籤 2，輪廓的結尾

20.1.13 CYCLE63－銑削輪廓挖槽

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	PRG	<_PRG>	STRING [100]	移除程式的名稱		
2		<_VARI>	INT	加工類型		
				單位：	加工程序	
					1 =	粗加工
					3 =	底部精加工
					4 =	邊緣精加工
					5 =	倒角
				十位數：	進給類型	
					0 =	垂直´¡¤J
					1 =	螺旋插入
					2 =	振盪插入
				百位數：	已保留	
				千位數：	拉升模式	
					0 =	拉升至回退平面
					1 =	拉升至參考點 + 安全間距
				萬位數：	粗加工及精加工基準的起點	
					0 =	自動
					1 =	手動
3	RP	<_RP>	REAL	回退平面（絕對）		
4	Z0	<_Z0>	REAL	刀具軸參考點（絕對）		
5	SC	<_SC>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
6	Z1	<_Z1>	REAL	最終深度（請參閱 <_AMODE> UNITS）		
7	F	<_F>	REAL	粗加工/精加工時平面上的進給率		
8	FZ	<_FZ>	REAL	深度進給速率		
9	DXY	<_DXY>	REAL	進給平面 - 單位（請參閱 <_AMODE> TENS）		
10	DZ	<_DZ>	REAL	深度進給：		
11	UXY	<_UXY>	REAL	精加工允差量，平面		
12	UZ	<_UZ>	REAL	精加工允差，深度		

號碼	參數遮罩	內部參數	資料類型	含義		
13		<_CDIR>	INT	銑削方向	0 =	下切
					1 =	上切
14	XS	<_XS>	REAL	起點 X，絕對		
15	YS	<_YS>	REAL	起點 Y，絕對		
16	ER	<_ER>	REAL	螺旋插入： 半徑		
17	EP	<_EP>	REAL	螺旋插入： 螺距		
18	EW	<_EW>	REAL	振盪插入： 最大插入角度		
19	FS	<_FS>	REAL	倒角時的倒角寬度（增量）		
20	ZFS	<_ZFS>	REAL	倒角時刀尖的插入深度（請參閱 <_AMODE> HUNDREDS）		
21	TR	<_TR>	STRING[32]	加工殘料時的參考刀具名稱		
22	DR	<_DR>	INT	加工殘料時的參考刀具 D 編號		
23		<_UMODE>	INT	已保留		
24		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工/僅計算起點	
					0 =	正常加工（不需要相容性模式）
					1 =	正常加工
					2 =	已保留

號碼	參數遮罩	內部參數	資料類型	含義		
25		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	已保留	
				百位數：	技術模式	
					1 =	口袋型
					2 =	島嶼
				千位數：	加工殘料	
					0 =	否
					1 =	是
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單
26		<_AMODE>	INT	替代模式		
				單位：	最終深度 (Z1)	
					0 =	絕對（相容性模式）
					1 =	增量
				十位數：	平面進給的單位 (DXY)	
					0 =	mm
					1 =	刀具直徑 %
				百位數：	倒角的插入深度 (ZFS)	
					0 =	絕對值
					1 =	增量
				千位數：	---	已保留

20.1.14 CYCLE64－預鑽孔輪廓挖槽

語法

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>,
<_DXY>, <_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	PRG	<_PRG>	STRING [100]	鑽孔名稱/定心程式		
2		<_VARI>	INT	加工類型		
				單位：	已保留	
				十位數：	已保留	
				百位數：	已保留	
				千位數：	拉升模式	
					0 =	拉升至回退平面
					1 =	拉升至參考點 + 安全間距
3	RP	<_RP>	REAL	回退平面（絕對）		
4	Z0	<_Z0>	REAL	參考點（絕對）		
5	SC	<_SC>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
6	Z1	<_Z1>	REAL	鑽孔 / 定心深度（請參閱 <_AMODE> UNITS）		
7	F	<_F>	REAL	鑽孔/定心進給率		
8	DXY	<_DXY>	REAL	進給平面 - 單位（請參閱 <_AMODE> TENS）		
9	UXY	<_UXY>	REAL	精加工允差量，平面		
10	UZ	<_UZ>	REAL	精加工允差，深度		
11		<_CDIR>	INT	銑削方向	0 =	下切
					1 =	上切
12	TR	<_TR>	STRING[20]	參考刀具名稱		
13	DR	<_DR>	INT	參考刀具 D 編號		
14		<_UMODE>	INT	已保留		

號碼	參數遮罩	內部參數	資料類型	含義		
15		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工/僅計算起點	
					0 =	正常加工（不需要相容性模式）
					1 =	正常加工
					2 =	已保留
25		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	技術模式	
					1 =	預鑽孔
					2 =	中心定位
26		<_AMODE>	INT	替代模式		
				單位：	鑽孔 / 定心深度 Z1	
					0 =	絕對（相容性模式）
					1 =	增量
				十位數：	平面進給的單位 (DXY)	
					0 =	mm
					1 =	刀具直徑 %

20.1.15 CYCLE70－螺紋銑削

語法

```
CYCLE70(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
```

```
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<_RTP>	REAL	回退平面（絕對）		
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）		
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1	<_DP>	REAL	螺紋深度（絕對 / 增量），請參閱 <_AMODE> 考慮孔洞基準的旋出（至少半個螺距）		
5	∅	<_DIATH>	REAL	螺紋的公稱直徑		
6	H1	<_H1>	REAL	螺紋深度		
7	U	<_FAL>	REAL	精加工允差		
8	P	<_PIT>	REAL	螺距（選擇 <_PITA>： 毫米、英吋、MODULE、螺紋/英吋）		
9	NT	<_NT>	INT	刀具頂端上的齒數 刀具長度一定相對於底部的齒。		
10	DXY	<_MID>	REAL	每次切削之最大進給 <_MID> > <_H1>：全部在一次切削中		
11	F	<_FFR>	REAL	銑削進給		
12		<_TYPTH>	INT	螺紋種類	0 =	內螺紋
					1 =	外螺紋
13	X0	<_PA>	REAL	圓心第一軸（絕對）		
14	Y0	<_PO>	REAL	圓心第二軸（絕對）		
15	αS	<_NSP>	REAL	開始角度（多重起始螺紋）		

號碼	參數遮罩	內部參數	資料類型	含義		
16		<_VARI>	INT	加工類型		
				單位：		
					1 =	粗加工
					2 =	精加工
				十位數：		
					1 =	由上到下
					2 =	由上而下
				百位數：		
					0 =	右旋螺紋
					1 =	左旋螺紋
17		<_PITA>	INT	評估螺紋螺距		
					0 =	相容性模式
					1 =	以毫米為單位的螺距
					2 =	以螺紋 / 英吋為單位的螺距 (TPI)
					3 =	以英吋為單位的螺距
					4 =	以 MODULE 為單位的螺距
18		<_PITM>	STRING[15]	螺距輸入的標記字串（僅供介面使用）		
19		<_PTAB>	STRING[20]	螺紋表的字串（""、"ISO"、"BSW"、"BSP"、"UNC")（僅供介面使用）		
20		<_PTABA>	STRING[20]	從螺紋表選擇的字串（例如 "M 10"、"M 12"，...）（僅供介面使用）		
21		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	加工/計算起點	
					0 =	相容性模式
					1 =	正常加工

號碼	參數遮罩	內部參數	資料類型	含義		
22		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
23		<_AMODE>	INT	替代模式		
				單位：	螺紋長度（<_DP>）	
					0 =	絕對值
					1 =	增量

20.1.16 CYCLE72－路徑銑削

語法

```
CYCLE72(<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>,<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1		<_KNAME>	STRING [141]	輪廓副程式的名稱
2	RP	<_RTP>	REAL	回退平面（絕對）
3	Z0	<_RFP>	REAL	刀具軸參考點（絕對）
4	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
5	Z1	<_DP>	REAL	終點，最終深度（絕對 / 增量），請參閱 <_AMODE>
6	DZ	<_MID>	REAL	最大深度進給（增量；輸入時不帶正負號）
7	UXY	<_FAL>	REAL	精加工允差，平面（增量），稜邊輪廓的允差
8	UZ	<_FALD>	REAL	精加工允差深度（增量），刀塔允差（輸入時不帶正負號）
9	FX	<_FFP1>	REAL	輪廓上的進給率

號碼	參數遮罩	內部參數	資料類型	含義		
10	FZ	<_FFD>	REAL	深度進給的進給率（或空間進給）		
11		<_VARI>	INT	加工類型		
				單位：	加工	
					1 =	粗加工
					2 =	精加工
					5 =	倒角
				十位數：		
					0 =	G0 的中間路徑
					1 =	G1 的中間路徑
				百位數：	在輪廓終點回退	
					0 =	從輪廓終點回退至參考點
					1 =	從輪廓終點回退至 + <_SDIS>
					2 =	從輪廓終點回退：依據 <_SDIS>
					3 =	輪廓終點無回退，以輪廓進給逼近下一個起點
				千位數：	已保留	
				萬位數：	加工輪廓	
					0 =	加工輪廓向前
					1 =	加工輪廓向後 向後加工的限制： • 最多 170 個輪廓元素（包括輪廓或倒圓角） • 只有在 (X/Y) 及 F 平面的數值才加以評估
12		<_RL>	INT	加工方向		
					40 =	輪廓中心（G40，逼近及回退：直線或垂直）
					41 =	輪廓左邊（G41，逼近及回退：直線或圓）
					42 =	輪廓右邊（G42，逼近及回退：直線或圓）

號碼	參數遮罩	內部參數	資料類型	含義		
13		<_AS1>	INT	輪廓逼近移動		
				單位：		
					1 =	直線
					2 =	四分之一圓
					3 =	半圓
					4 =	垂直逼近及回退
				十位數：		
					0 =	最後的移動，平面中
					1 =	最後的移動，空間中
14	L1	<_LP1>	REAL	逼近路徑或逼近半徑（增量；輸入時不帶正負號）		
15	FZ	<_FF3>	REAL	中間路徑的進給率（以 G94/G95 輪廓）		
16		<_AS2>	INT	輪廓逼近移動（非垂直逼近/回退）		
				單位：		
					1 =	直線
					2 =	四分之一圓
					3 =	半圓
				十位數：		
					0 =	最後的移動，平面中
					1 =	最後的移動，空間中
17	L2	<_LP2>	REAL	回退路徑或回退半徑（增量，輸入時不帶正負號）		
18		<_UMODE>	INT	已保留		
19	FS	<_FS>	REAL	倒角寬度（增量）		
20	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		
21		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工/僅計算起點	
					0 =	相容性模式
					1 =	正常加工

號碼	參數遮罩	內部參數	資料類型	含義		
22		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	進給率類型：平面及深度進給率的 G 群組 (G94/G95)	
					0 =	相容性模式
					1 =	循環呼叫前的 G 指令平面及深度進給率的 G94/G95 相同
				千位數：		
					0 =	相容性模式：輪廓名稱在 <_KNAME>
					1 =	輪廓名稱在 CYCLE62 程式設計，並傳送給 _SC_CONT_NAME
23		<_AMODE>	INT	替代模式		
				單位：	終點 Z1（<_DP>）	
					0 =	絕對（相容性模式）
					1 =	增量
				十位數：	平面進給的單位	
					0 =	毫米、英吋
					1 =	已保留
				百位數：	倒角的插入深度（<_ZFS>）	
					0 =	絕對值
					1 =	增量

說明

若下列傳輸參數係間接程式設計（作為參數），畫面表單並未重置：

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

20.1.17 CYCLE76 - 矩形島嶼銑削

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<_RTP>	REAL	回退平面（絕對）		
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）		
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1	<_DP>	REAL	島嶼深度（絕對）		
5		<_DPR>	REAL	相對於 Z0 的島嶼深度（增量）（輸入時輸入時不帶正負號）		
6	L	<_LENG>	REAL	島嶼長度，請參閱 <_GMODE>（輸入時不帶正負號）		
7	W	<_WID>	REAL	島嶼寬度，請參閱 <_GMODE>（輸入時不帶正負號）		
8	R	<_CRAD>	REAL	島嶼角半徑（輸入時不帶正負號）		
9	X0	<_PA>	REAL	平面第一軸上島嶼的參考點（絕對）		
10	Y0	<_PO>	REAL	平面第二軸上島嶼的參考點（絕對）		
11	α0	<_STA>	REAL	旋轉角，縱向軸 (L) 與平面第一軸的夾角		
12	DZ	<_MID>	REAL	最大深度進給（增量；輸入時不帶正負號）		
13	UXY	<_FAL>	REAL	精加工允差，平面（增量），稜邊輪廓的允差		
14	UZ	<_FALD>	REAL	精加工允差深度（增量），刀塔允差（輸入時不帶正負號）		
15	FX	<_FFP1>	REAL	輪廓上的進給率		
16	FZ	<_FFD>	REAL	深度進給速率		
17		<_CDIR>	INT	銑床方向（輸入時不帶正負號）		
				單位：		
					0 =	下切
					1 =	上切

號碼	參數遮罩	內部參數	資料類型	含義		
18		<_VARI>	INT	加工		
				單位：		
					1 =	粗加工
					2 =	精加工
					5 =	倒角
19	L1	<_AP1>	REAL	毛胚島嶼長度		
20	W1	<_AP2>	REAL	毛胚島嶼寬度		
21	FS	<_FS>	REAL	倒角寬度（增量）		
22	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		
23		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工或僅計算起點	
					0 =	相容性模式
					1 =	正常加工
				千位數：	根據中心或角隅的島嶼尺寸標註	
					0 =	相容性模式
					1 =	透過中心尺寸標註
					2 =	角點的尺寸標註， 島嶼 +L +W
					3 =	角點的尺寸標註， 島嶼 -L +W
					4 =	角點的尺寸標註， 島嶼 +L -W
					5 =	角點的尺寸標註， 島嶼 -L -W
				萬位數：	完整加工或重新加工	
					0 =	相容性模式
					1 =	完整加工
					2 =	後加工

號碼	參數遮罩	內部參數	資料類型	含義		
24		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	---	已保留
				百位數：	---	已保留
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單
25		<_AMODE>	INT	替代模式		
				單位：	最終深度 Z1 (DP)	
					0 =	相容性
					1 =	增量
					2 =	絕對值
				十位數：	已保留	
				百位數：	倒角的插入深度 (ZFS)	
					0 =	絕對值
					1 =	增量

20.1.18 CYCLE77 - 圓弧島嶼銑削

語法

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>,
<_VARI>, <_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<_RTP>	REAL	回退平面（絕對）		
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）		
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1	<_DP>	REAL	島嶼深度（絕對）		
5		<_DPR>	REAL	相對於 Z0 的島嶼深度（增量）（輸入時輸入時不帶正負號）		
6	∅	<_CDIAM>	REAL	島嶼直徑（輸入時不帶正負號）		
7	X0	<_PA>	REAL	平面第一軸上島嶼的參考點（絕對）		
8	Y0	<_PO>	REAL	平面第二軸上島嶼的參考點（絕對）		
9	DZ	<_MID>	REAL	最大深度進給（增量；輸入時不帶正負號）		
10	UXY	<_FAL>	REAL	精加工允差，平面（增量），稜邊輪廓的允差		
11	UZ	<_FALD>	REAL	精加工允差深度（增量），刀塔允差（輸入時不帶正負號）		
12	FX	<_FFP1>	REAL	輪廓上的進給率		
13	FZ	<_FFD>	REAL	深度進給速率		
14		<_CDIR>	INT	銑床方向（輸入時不帶正負號）		
				單位：		
					0 =	下切
					1 =	上切
15		<_VARI>	INT	加工類型		
				單位：	加工	
					1 =	粗加工直到達到最終加工允差
					2 =	精加工（允插 X/Y/Z=0）
					5 =	倒角
16	∅1	<_AP1>	REAL	胚料島嶼直徑		
17	FS	<_FS>	REAL	倒角寬度（增量）		
18	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		

號碼	參數遮罩	內部參數	資料類型	含義		
19		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工/僅計算起點	
					0 =	相容性模式
					1 =	正常加工
				千位數：	已保留	
				萬位數：	完整加工/重新加工	
					0 =	相容性模式（和之前一樣處理 <_AP1>）
					1 =	完整加工
					2 =	後加工
20		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	---	已保留
				百位數：	---	已保留
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單

號碼	參數遮罩	內部參數	資料類型	含義		
21		<_AMODE>	INT	替代模式		
				單位：	最終深度 Z1 (DP)	
					0 =	絕對（相容性模式）
					1 =	增量
					2 =	絕對值
				十位數：	已保留	
				百位數：	倒角的插入深度 (ZFS)	
					0 =	絕對值
					1 =	增量

20.1.19 CYCLE78－鑽孔螺紋銑削

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	RP	<_RTP>	REAL	回退平面（絕對）
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<_DP>	REAL	最終鑽孔深度（絕對/增量），請參閱 <_AMODE>
5		<_ADPR>	REAL	以降低的鑽孔進給率預鑽深度（增量），以 <_VARI> TEN THOUSANDS 使其生效
6	D	<_FDPR>	REAL	最大深度進給（增量） D ≥ Z1 ⇒ 一個進給至最終鑽孔深度 D < Z1 ⇒ 鑽深孔循環含多重進給與排屑移除
7	ZR	<_LDPR>	REAL	以 FR 進給通鑽切削（增量）時保持鑽孔深度
8	∅	<_DIAM>	REAL	螺紋的公稱直徑

號碼	參數遮罩	內部參數	資料類型	含義		
9	P	<_PIT>	REAL	螺距的數值		
10		<_PITA>	INT	評估螺紋螺距 P		
					1 =	以毫米 / 轉為單位的螺距
					2 =	以螺距 / 英吋為單位的螺距
					3 =	以英吋 / 轉為單位的螺距
					4 =	以 MODULE 為單位的螺距
11	DF	<_DAM>	REAL	每次額外進給的絕對值 / 百分比（縮減）請參閱 <_AMODE>		
12	V1	<_MDEP>	REAL	最小進給（增量），僅於縮減時啟用		
13		<_VARI>	INT	加工類型		
				單位：	已保留	
				十位數：	銑削螺紋前進行排屑	
					0 =	螺紋銑削前無排屑（只有在最終鑽孔深度時啟用）
					1 =	螺紋銑削前排屑（只有在最終鑽孔深度時啟用）
				百位數：	右旋 / 左旋螺紋	
					0 =	右旋螺紋
					1 =	左旋螺紋
				千位數：	依鑽孔進給率鑽除剩餘深度	
					0 =	依鑽孔進給率 FR 無鑽除剩餘深度
					1 =	依鑽孔進給率鑽除剩餘深度
				萬位數：	以減小的進給率預鑽孔	
					0 =	不以減小的進給率預鑽孔
					1 =	以減小的進給率預鑽孔 預鑽孔進給率 = 0.3 F1，若 F1 < 0.15 毫米 / 轉 預鑽孔進給率 = 0.1 毫米 / 轉，若 F1 ≥ 0.15 毫米 / 轉
14		<_CDIR>	INT	銑削方向	0 =	下切
					1 =	上切
					4 =	上切 + 下切（結合粗加工 + 精加工）

號碼	參數遮罩	內部參數	資料類型	含義		
15	Z2	<_GE>	REAL	銑削螺紋前之回退距離（增量）		
16	F1	<_FFD>	REAL	鑽孔進給率（毫米 / 分或英吋 / 分或毫米 / 轉）		
17	FR	<_FRDP>	REAL	鑽除剩餘深度的鑽孔進給率（毫米 / 分或毫米 / 轉）		
18	F2	<_FFR>	REAL	螺紋銑削的進給率（毫米 / 分或毫米 / 齒）		
19	FS	<_FFP2>	REAL	<_CDIR> = 4 的精加工進給率（毫米 / 分或毫米 / 齒）		
20		<_FFA>	INT	進給率評估		
				單位：	鑽孔進給 F1	
				十位數：	鑽除剩餘深度的鑽孔進給率 FR	
				百位數：	銑削螺紋之進給率 F2	
				千位數：	精加工進給率 FS	
21		<_PITM>	STRING[15]	螺距輸入的標記字串（僅供介面使用）[1)]		
22		<_PTAB>	STRING[20]	螺紋表的字串（""、"ISO"、"BSW"、"BSP"、"UNC")（僅供介面使用）[1)]		
23		<_PTABA>	STRING[20]	從螺紋表選擇的字串（例如 "M 10"、"M 12"，...）（僅供介面使用））[1)]		
24		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料），已保留		
25		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

號碼	參數遮罩	內部參數	資料類型	含義		
26		<_AMODE>	INT	替代模式		
				單位：	鑽孔深度 = 最終鑽孔深度 Z1 絕對/增量	
					0 =	絕對值
					1 =	增量
				十位數：	每次額外進給的絕對值 / 百分比（縮減）	
					0 =	絕對值
					1 =	百分比（0.001 至 100%）

說明

[1] 參數 21、22 及 23 只用在畫面表單螺紋表的螺紋選擇上。 螺紋表無法透過循環執行時間內的循環定義存取。

20.1.20 CYCLE79－多邊緣

語法

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	RP	<_RTP>	REAL	回退平面（絕對）
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<_DP>	REAL	多重稜邊深度（絕對 / 增量），請參閱 <_AMODE>
5	N	<_NUM>	INT	稜邊數 (1...n)
6	SW / L	<_SWL>	REAL	跨越平面或稜邊長度的寬度（視 <_VARI> 而定） （跨越平面的寬度 "SW"，或稜邊長度的寬度 "L"） 只有在稜邊數為偶數及單一稜邊時，跨越平面的寬度

號碼	參數遮罩	內部參數	資料類型	含義		
7	X0	<_PA>	REAL	島嶼參考點，第一軸（絕對）		
8	Y0	<_PO>	REAL	島嶼參考點，第二軸（絕對）		
9	α0	<_STA>	REAL	相對於第一軸（X 軸）的旋轉角，稜邊中心		
10	R1/FS1	<_RC>	REAL	以 <_NUM> > 2 角隅倒圓角（半徑 / 倒角，請參閱 <_AMODE>） （增量，輸入時不帶正負號） （半徑為 "R1"，倒角為 "FS1"）		
11	∅	<_AP1>	REAL	未加工的島嶼直徑		
12	DXY	<_MIDA>	REAL	最大進給寬度（單位請參閱 <_AMODE>）		
13	DZ	<_MID>	REAL	最大深度進給		
14	UXY	<_FAL>	REAL	精加工允差量，平面		
15	UZ	<_FALD>	REAL	精加工允差，深度		
16	F	<_FFP1>	REAL	機台加工進給速率		
17		<_CDIR>	INT	銑削方向	0 =	下切
					1 =	上切
18		<_VARI>	INT	加工類型		
				單位：	加工	
					1 =	粗加工
					2 =	精加工
					3 =	邊緣精加工
					5 =	倒角
				十位數：	跨越平面或邊緣長度的寬度	
					0 =	跨越平面的寬度
					1 =	邊長
19	FS	<_FS>	REAL	倒角寬度（增量）		
20	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		
21		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工或僅計算起點	
					1 =	正常加工

號碼	參數遮罩	內部參數	資料類型	含義		
22		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/18/19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	---	已保留
				百位數：	---	已保留
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單
23		<_AMODE>	INT	替代模式		
				單位：	最終深度（<_DP>）	
					0 =	絕對值
					1 =	增量
				十位數：	平面進給的單位（<_MIDA>）	
					0 =	mm
					1 =	刀具直徑 %
				百位數：	倒角的插入深度（<_ZFS>）	
					0 =	絕對值
					1 =	增量
				千位數：	轉角倒圓角（<_RC>）	
					0 =	半徑
					1 =	倒角

20.1.21 CYCLE81 - 鑽孔、定心

語法

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<RTP>	REAL	回退平面（絕對）		
2	Z0	<RFP>	REAL	參考點（絕對）		
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1/∅	<DP>	REAL	鑽孔深度（絕對）/ 中心直徑（絕對），請參閱 <_GMODE>		
5	Z1	<DPR>	REAL	鑽孔深度（增量）		
6	DT	<DTB>	REAL	最終鑽孔深度的停頓時間，請參閱 <_AMODE>		
7		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	相對於深度/直徑的中心	
					0 =	相容性，深度
					1 =	直徑
8		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

號碼	參數遮罩	內部參數	資料類型	含義		
9		<_AMODE>	INT	替代模式		
				單位：	鑽孔深度 Z1（絕對/增量）	
					0 =	相容性，來自 DP/DPR 程式設計
					1 =	增量
					2 =	絕對值
				十位數：	最終鑽孔深度 DT 停頓時間，以秒/轉為單位	
					0 =	相容性，來自 DTB 符號（> 0 秒或 < 0 轉）
					1 =	以秒為單位
					2 =	以轉為單位

20.1.22 CYCLE82 - 鑽孔、逆鏜孔

語法

```
CYCLE82(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

參數

編號	參數遮罩	內部參數	資料類型	含義
1	RP	<RTP>	REAL	回退平面（絕對）
2	Z0	<RFP>	REAL	參考點（絕對）
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<DP>	REAL	鑽孔深度（絕對），請參閱 <_AMODE>
5	Z1	<DPR>	REAL	鑽孔深度（增量），請參閱 <_AMODE>
6	DT	<DTB>	REAL	最終鑽孔深度的停頓時間，請參閱 <_AMODE>

編號	參數遮罩	內部參數	資料類型	含義		
7		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	保留	
				十位數：	相對於刀尖/刀柄的鑽孔	
					0 =	相容性，刀尖
					1 =	刀柄
8		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	保留	
				百位數：	保留	
				千位數：	保留	
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入： 完整
					1 =	輸入： 基本

編號	參數遮罩	內部參數	資料類型	含義		
9		<_AMODE>	INT	替代模式		
				單位：	鑽孔深度 Z1（絕對/增量）	
					0 =	相容性，來自 DP/DPR 程式設計
					1 =	增量
					2 =	絕對值
				十位數：	最終鑽孔深度 DT 停頓時間，以秒/轉為單位	
					0 =	相容性，來自 DT 符號（> 0 秒/ < 0 轉）
					1 =	以秒為單位
					2 =	以轉為單位
				百位數：	鑽孔深度 ZA 絕對/增量	
					0 =	增量
					1 =	絕對值
				千位數：	評估預鑽孔的進給率	
					0 =	以 % 表示鑽孔進給率
					1 =	F / 分
					2 =	F / 轉
				萬位數：	剩下的鑽孔深度 ZD 絕對/增量	
					0 =	增量
					1 =	絕對值
				十萬位數：	評估剩下的鑽孔進給率	
					0 =	以 % 表示鑽孔進給率
					1 =	F / 分
					2 =	F / 轉

編號	參數遮罩	內部參數	資料類型	含義	
10		<_VARI>	INT	預鑽孔/通鑽的加工類型	
				單位：	保留
				十位數：	保留
				百位數：	保留
				千位數：	通鑽
					0 = 通鑽「否」
					1 = 通鑽「是」
				萬位數：	預鑽孔
					0 = 預鑽孔「是」
					1 = 預鑽孔「否」
11	ZA	<S_ZA>	REAL	相對於參考點或絕對的增量預鑽孔深度（請參閱 <_AMODE> HUNDREDS）	
12	FA	<S_FA>	REAL	預鑽孔進給率為數值或以 % 為單位（搭配 <_AMODE> THOUSANDS 使用）	
13	ZD	<S_ZD>	REAL	相對於最終鑽孔深度或絕對的增量鑽孔深度（請參閱 <_AMODE> TEN THOUSANDS）	
14	FD	<S_FD>	REAL	剩餘鑽孔進給率為數值或以 % 為單位（搭配 <_AMODE> HUNDRED THOUSANDS 使用）	

20.1.23 CYCLE83－深孔鑽

語法

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>,
<_DAM>, <DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>,
<_DTD>, <_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	RP	<RTP>	REAL	回退平面（絕對）
2	Z0	<RFP>	REAL	參考點（絕對）
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<DP>	REAL	最終鑽孔深度（絕對），請參閱 <_AMODE>

號碼	參數遮罩	內部參數	資料類型	含義		
5	Z1	<DPR>	REAL	最終鑽孔深度（增量），請參閱 <_AMODE>		
6	D	<FDEP>	REAL	第一鑽孔深度（絕對），請參閱 <_AMODE>		
7	D	<FDPR>	REAL	第一鑽孔深度（增量），請參閱 <_AMODE>		
8	DF	<_DAM>	REAL	每次額外進給之縮減值 / 百分比，請參閱 <_AMODE>		
9	DTB	<DTB>	REAL	鑽孔深度的停頓時間，請參閱 <_AMODE>		
10	DTS	<DTS>	REAL	起點的停頓時間（僅供排屑移除），請參閱 <_AMODE>		
11	FD1	<FRF>	REAL	第一進給之進給率百分比，請參閱 <_AMODE>		
12		<VARI>	INT	加工類型		
				單位：	斷屑/排屑	
					0 =	斷屑
					1 =	切屑移除
13		<_AXN>	INT	刀具軸		
					0 =	第三幾何座標軸
					1 =	第一幾何座標軸
					2 =	第二幾何座標軸
					> 2	第三幾何座標軸
14	V1	<_MDEP>	REAL	最小進給（僅供縮減百分比）		
15	V2	<_VRT>	REAL	每次加工步驟之後的回退距離（僅供斷屑）		
					> 0	可變回退距離
					0 =	預設值 1 毫米
16	DT	<_DTD>	REAL	最終鑽孔深度的停頓時間，請參閱 <_AMODE>		
17	V3	<_DIS1>	REAL	限制距離（僅供排屑移除），請參閱 <_AMODE>		
18		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	相對於刀尖/刀柄的鑽孔	
					0 =	刀尖
					1 =	刀柄

號碼	參數遮罩	內部參數	資料類型	含義		
19		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	---	已保留
				百位數：	---	已保留
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單

號碼	參數遮罩	內部參數	資料類型	含義		
20		<_AMODE>	INT	替代模式		
				單位：	鑽孔深度 = 最終鑽孔深度 Z1（絕對/增量）	
					0 =	相容性，來自程式設計 <DP>/<DPR>
					1 =	增量
					2 =	絕對值
				十位數：	鑽孔深度 DTB 停頓時間，以秒/轉為單位	
					0 =	相容性，來自 DTB 符號（> 0 秒或 < 0 轉）
					1 =	以秒為單位
					2 =	以轉為單位
				百位數：	DTS 起點的停頓時間，以秒/轉為單位	
					0 =	相容性，來自 DTS 符號（> 0 秒或 < 0 轉）
					1 =	以秒為單位
					2 =	以轉為單位
				千位數：	最終鑽孔深度 DTD 停頓時間，以秒 / 轉為單位	
					0 =	相容性，來自 DTD 符號（> 0 秒或 < 0 轉）
					1 =	以秒為單位
					2 =	以轉為單位
				萬位數：	第一鑽孔深度 D（絕對 / 增量）	
					0 =	相容性，來自程式設計 <FDEPF>/<DPR>
					1 =	增量
					2 =	絕對值
				十萬位數：	每次額外進給之縮減值 / 百分比 <_DAM>	

號碼	參數遮罩	內部參數	資料類型	含義		
					0 =	相容性，來自 <_DAM> 符號（> 0 縮減值或 < 0 係數 0.001 至 1.0）
					1 =	縮減值
					2 =	百分比（0.001 至 100%）
				百萬位數：	限制距離 V3 自動/手動	
					0 =	相容性來自 <_DIS1> 符號（= 0 自動或 > 0 手動）
					1 =	自動（在循環中計算）
					2 =	手動（已設定數值）
				千萬位數：	第一次進給的進給率係數 <FRF> ，以係數 / 百分比為單位	
					0 =	相容性，係數單位（0.001 至 1.0，FRF = 0 代表 100%）
					1 =	百分比（0.001 至 999.999%）

20.1.24 CYCLE84－不搭配補正夾頭進行攻牙

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	RP	<RTP>	REAL	回退平面（絕對）
2	Z0	<RFP>	REAL	參考點（絕對）
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
4	Z1	<DP>	REAL	鑽孔深度 = 最終鑽孔深度（絕對），請參閱 <_AMODE>
5	Z1	<DPR>	REAL	鑽孔深度 = 最終鑽孔深度（增量），請參閱 <_AMODE>

號碼	參數遮罩	內部參數	資料類型	含義			
6	DT	<DTB>	REAL	鑽孔深度上的停頓時間，以秒為單位			
7	SDE	<SDAC>	INT	循環結束後的旋轉方向			
8		<MPIT>	REAL	僅用於「ISO 公制」的螺紋尺寸（螺距在執行時間時內部計算）			
9	P	<PIT>	REAL	螺距數值，單位請參閱 <_PITA>			
10	αS[1)]	<POSS>	REAL	定向主軸停止的主軸位置			
11	S	<SST>	REAL	攻牙的主軸速率			
12	SR	<SST1>	REAL	回退的主軸速率			
13		<_AXN>	INT	鑽孔軸	0 =	第三幾何座標軸	
					1 =	第一幾何座標軸	
					2 =	第二幾何座標軸	
					≥ 3 =	第三幾何座標軸	
14		<_PITA>	INT	螺距單位（ <PIT> 及 <MPIT> 的評估）			
					0 =	以毫米為單位的螺距	- 評估 <MPIT>/<PIT>
					1 =	以毫米為單位的螺距	- 評估 <PIT>
					2 =	以 TPI 為單位的螺距	- <PIT> 的評估（每英吋的螺紋）
					3 =	以英吋為單位的螺距	- 評估 <PIT>
					4 =	MODULUS	- 評估 <PIT>

號碼	參數遮罩	內部參數	資料類型	含義		
15		<_TECHNO>	INT	技術 1)		
				單位：	精確停止反應	
					0 =	循環呼叫前啟用精確停止反應
					1 =	精確停止 G601
					2 =	精確停止 G602
					3 =	精確停止 G603
				十位數：	前饋控制	
					0 =	循環呼叫前，啟用有 / 無前饋控制
					1 =	具前饋控制 FFWON
					2 =	無前饋控制 FFWOF
				百位數：	加速度	
					0 =	循環呼叫前啟用 SOFT/BRISK/ DRIVE
					1 =	具震動限制 SOFT
					2 =	無震動限制 BRISK
					3 =	降低加速度 DRIVE
				千位數：	MCALL 主軸模式	
					0 =	針對 MCALL 重新啟動主軸操作
					1 =	讓 MCALL 維持位置控制
16		<_VARI>	INT	加工類型		
				單位：	0 =	1 次切削
					1 =	斷屑（深孔攻牙）
					2 =	排屑（深孔攻牙）
				千位數：	ISO/SIEMENS 模式與畫面表單無關	
					0 =	從 ISO 相容性呼叫
					1 =	從 SIEMENS 關聯呼叫
17	D	<_DAM>	REAL	最大深度進給（僅適用於排屑/斷屑）		
18	V2	<_VRT>	REAL	每次加工步驟之後的回退距離（僅供斷屑），請參閱 <_AMODE>		
19		<_PITM>	STRING[15]	螺距輸入的標記字串 2)		
20		<_PTAB>	STRING[5]	螺紋表的字串（""、"ISO"、"BSW"、"BSP"、"UNC")2)		

號碼	參數遮罩	內部參數	資料類型	含義		
21		<_PTABA>	STRING[20]	從螺紋表選擇的字串（例如 "M 10"、"M 12"，...）[2]		
22		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
23		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	已保留	
				百位數：	已保留	
				千位數：	相容性模式（或僅重新編譯畫面表單），若 MD 52216 bit0 = 1[1]	
					0 =	顯示技術參數（相容性）：TECHNO 參數有效
					1 =	不顯示技術參數：「循環呼叫前」啟用技術
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單

號碼	參數遮罩	內部參數	資料類型	含義		
24		<_AMODE>	INT	替代模式		
				單位：	鑽孔深度 = 最終鑽孔深度 Z1（絕對/增量）	
					0 =	相容性，來自程式設計 <DP>/<DPR>
					1 =	增量
					2 =	絕對值
				十位數：	已保留	
				百位數：	已保留	
				千位數：	螺紋旋轉方向右旋/左旋	
					0 =	相容性，來自 PIT/MPTI 符號
					1 =	向右
					2 =	向左
				萬位數：	已保留	
				十萬位數：	已保留	
				百萬位數：	每次加工步驟 V2 手動/自動之後的回退距離	
					0 =	相容性，來自 <_VRT> 程式設計（> 0 變數值或 ≤ 0 標準值 1 毫米 / 0.0394 英吋）
					1 =	自動（標準值 1 毫米 / 0.0394 英吋）
					2 =	手動（在 V2 下程式設計）

[1] 根據設定日期 SD52216 $MCS_FUNCTION_MASK_DRILL，技術欄位可以隱藏

[2] 參數 19、20 及 21 只用在或面表單螺紋表的螺紋選擇上。 螺紋表無法透過循環執行時間內的循環定義存取。

20.1.25 CYCLE85－鉸孔

語法

```
CYCLE85(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<RTP>	REAL	回退平面（絕對）		
2	Z0	<RFP>	REAL	參考點（絕對）		
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1	<DP>	REAL	鑽孔深度（絕對），請參閱 <_AMODE>		
5	Z1	<DPR>	REAL	鑽孔深度（增量），請參閱 <_AMODE>		
6	DT	<DTB>	REAL	最終鑽孔深度的停頓時間，請參閱 <_AMODE>		
7	F	<FFR>	REAL	進給率		
8	FR	<RFF>	REAL	回退期間的進給率		
9		<_GMODE>	INT	已保留		
10		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
11		<_AMODE>	INT	替代模式（鑽孔）		
				單位：	鑽孔深度 Z1（絕對/增量）	
					0 =	相容性，來自 DP/DPR 程式設計
					1 =	增量
					2 =	絕對值
				十位數：	最終鑽孔深度 DT 停頓時間，以秒/轉為單位	
					0 =	相容性，來自 DT 符號（> 0 秒或 < 0 轉）
					1 =	以秒為單位
					2 =	以轉為單位

20.1.26 CYCLE86－鏜孔

語法

```
CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>,
<RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<RTP>	REAL	回退平面（絕對）		
2	Z0	<RFP>	REAL	參考點（絕對）		
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1	<DP>	REAL	鑽孔深度（絕對），請參閱 <_AMODE>		
5	Z1	<DPR>	REAL	鑽孔深度（增量），請參閱 <_AMODE>		
6	DT	<DTB>	REAL	最終鑽孔深度的停頓時間，請參閱 <_AMODE>		
7	DIR	<SDIR>	INT	主軸旋轉方向	3 =	M3
					4 =	M4
8	DX	<RPA>	REAL	X 方向的拉升距離		
9	DY	<RPO>	REAL	Y 方向的拉升距離		
10	DZ	<RPAP>	REAL	Z 方向的拉升距離		
11	SPOS	<POSS>	REAL	拉升的主軸位置（供定向主軸停止，以度為單位）		
12		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	拉升模式	
					0 =	舉升，相容性
					1 =	不離開輪廓
13		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

號碼	參數遮罩	內部參數	資料類型	含義		
14		<_AMODE>	INT	替代模式		
				單位：	鑽孔深度 Z1（絕對/增量）	
					0 =	相容性，來自程式設計 <DP>/<DPR>
					1 =	增量
					2 =	絕對值
				十位數：	最終鑽孔深度 DT 停頓時間，以秒/轉為單位	
					0 =	相容性，來自 DT 符號（> 0 秒或 < 0 轉）
					1 =	以秒為單位
					2 =	以轉為單位

20.1.27 CYCLE92－切離

語法

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>,
<_SV1>, <_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>,
<_VARI>, <_DN>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	X0	<_SPD>	REAL	參考點（絕對，永遠為直徑）
2	Y0	<_SPL>	REAL	參考點（絕對）
3	X1	<_DIAG1>	REAL	速率降低的深度，請參閱 <_AMODE> (UNITS)
4	X2	<_DIAG2>	REAL	最終深度，請參閱 <_AMODE> (TENS)
5	R/FS	<_RC>	REAL	倒圓角狀態或倒角寬度，請參閱 <_AMODE> (THOUSANDS)
6	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）
7	S	<_SV1>	REAL	主軸定速，請參閱 <_AMODE> (TEN THOUSANDS)
	V			恆定切削率
8	SV	<_SV2>	REAL	切削定速的最大速度

號碼	參數遮罩	內部參數	資料類型	含義		
9	DIR	<_SDAC>	INT	主軸旋轉方向	3 =	針對 M3
					4 =	針對 M4
10	F	<_FF1>	REAL	速率降低且盡可能深度的進給		
11	FR	<_FF2>	REAL	盡可能最終深度的進給降低		
12	SR	<_SS2>	REAL	盡可能最終深度的速率降低		
13	XM	<_DIAGM>	REAL	收回工件夾具的深度（絕對，永遠為直徑）		
14		<_VARI>	INT	加工類型		
				單位：	回退	
					0 =	回退至 <_SPD> + <_SDIS>
					1 =	不在終點上回退
				十位數：	工件收集	
					0 =	否，請勿執行 M 指令
					1 =	是，從 CUST_TECHCYC(101) 呼叫 - 開啟抽屜，CUST_TECHCYC(102) - 關閉抽屜
15		<_DN>	INT	刀具第二稜邊的 D 編號；如未程式設計 ⇒ D+1		
20		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

號碼	參數遮罩	內部參數	資料類型	含義		
21		<_AMODE>	INT	替代模式		
				單位：	速率降低的深度 (<_DIAG1>)	
					0 =	絕對，直徑中的橫向軸數值
					1 =	增量，半徑中的橫向軸數值
				十位數：	最終深度 (<_DIAG2>)	
					0 =	絕對，直徑中的橫向軸數值
					1 =	增量，半徑中的橫向軸數值
				百位數：	已保留	
				千位數：	半徑 / 倒角 (<_RC>)	
					0 =	半徑
					1 =	倒角
				萬位數：	主軸轉速 / 切削率 (<_SV1>)	
					0 =	主軸定速
					1 =	恆定切削率

20.1.28 CYCLE95－輪廓切削

語法

```
CYCLE95(<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>,
<_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	CON	<NPP>	STRING [140]	輪廓名稱
2	D	<MID>	REAL	粗加工時的最大深度進給，請參閱 <_GMODE>
3	UZ	<FALZ>	REAL	Z 的精加工允差
4	UX	<FALX>	REAL	X 的精加工允差
5	U	<FAL>	REAL	平行於輪廓的精加工允差（雙軸同時有效）
6	F	<FF1>	REAL	粗加工進給率
7	FY	<FF2>	REAL	插入進給率，離隙切削

號碼	參數遮罩	內部參數	資料類型	含義		
8	FS	<FF3>	REAL	精加工進給率		
9		<_VARI>	INT	加工類型		
				UNITS 及 TENS：		
					1 =	粗加工，縱向，外部
					2 =	粗加工，橫向，外部
					3 =	粗加工，縱向，內部
					4 =	粗加工，橫向，內部
					5 =	精加工，縱向，外部
					6 =	精加工，橫向，外部
					7 =	精加工，縱向，內部
					8 =	精加工，橫向，內部
					9 =	完整加工，縱向，外部
					10 =	完整加工，橫向，外部
					11 =	完整加工，縱向，內部
					12 =	完整加工，橫向，內部
				百位數：		
					0 =	輪廓有倒圓角，無剩餘角隅
					1 =	輪廓無倒圓角
					2 =	只有前一個交點的倒圓角、有剩餘角隅
10	DT	<DT>	REAL	進給中斷的停頓時間		
11	DI	<DAM>	REAL	進給中斷的距離		
12	VRT	<_VRT>	REAL	從輪廓開始的拉升距離		
					0 =	內部使用 1 毫米的舉升距離，忽略啟用的系統（英吋或公制）
					> 0 =	舉升距離

號碼	參數遮罩	內部參數	資料類型	含義		
13		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	評估進給深度	
					0 =	對應 G 群組 DIAMON/ DIAMOF 計算進給深度
					1 =	進給深度做為半徑值（與 DIAMON/DIAMOF 無關）
14		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				千位數：		
					0 =	相容性模式： 輪廓名稱在 NPP
					1 =	輪廓名稱在 CYCLE62 程式設計，並傳送給 _SC_CONT_NAME

20.1.29 CYCLE98－螺紋鍊

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	Z0	<_PO1>	REAL	Z 中的參考點（abs）
2	X0	<_DM1>	REAL	X 方向上的參考點（絕對），於直徑
3	Z1	<_PO2>	REAL	Z 方向上的中間點 1（絕對 / 增量），請參閱 <_AMODE> (UNITS)
4	X1	<_DM2>	REAL	X 方向上的中間點 1（絕對 / 增量），請參閱 <_AMODE> (TENS) 或
	X1α			螺紋傾斜 1（-90° 至 90°） 直徑永遠為絕對，半徑永遠為增量
5	Z2	<_PO3>	REAL	Z 方向上的中間點 2（絕對 / 增量），請參閱 <_AMODE> (HUNDREDS)
6	X2	<_DM3>	REAL	X 方向上的中間點 2（絕對 / 增量），請參閱 <_AMODE> (THOUSANDS) 或
	X2α			螺紋傾斜 2（-90° 至 90°） 直徑永遠為絕對，半徑永遠為增量
7	Z3	<_PO4>	REAL	Z 方向上的終點（絕對 / 增量），請參閱 <_AMODE> (TEN THOUSANDS)
8	X3	<_DM4>	REAL	X 方向上的終點（絕對 / 增量），請參閱 <_AMODE> (HUNDRED THOUSANDS) 或
	X3α			螺紋傾斜 3（-90° 至 90°） 直徑永遠為絕對，半徑永遠為增量
9	LW	<APP>	REAL	螺紋旋入（增量，輸入時不帶正負號）
10	LR	<ROP>	REAL	螺紋旋出（增量，輸入時不帶正負號）
11	H1	<TDEP>	REAL	螺紋深度（增量，輸入時不帶正負號）
12	U	<FAL>	REAL	X 及 Z 方向的精加工允差

號碼	參數遮罩	內部參數	資料類型	含義		
13	DP	<_IANG>	REAL	作為距離或角度的進給斜率，請參閱 <_AMODE> (ONE MILLION)		
	αP			依據參數 <_VARI> (HUNDREDS) 的設定套用進給斜率。		
					定義 <_VARI_HUNDREDS = 0 - 相容性模式：	
					> 0 =	單邊側進給
					0 =	螺紋中垂直進給
					< 0 =	以側邊交替側進給
					定義 _VARI_HUNDREDS<>0：	
					> 0 =	正號側的進給
					0 =	中心進給
					< 0 =	負號側的進給
14	α0	<NSP>	REAL	第一螺紋的起始角度偏移		
15		<NRC>	INT	粗加工切削次數，請參閱 <_VARI> (TEN THOUSANDS)		
16	NN	<NID>	INT	空切削次數		
17	P0	<_PP1>	REAL	螺紋第一區段的螺距，請參閱 <_PITA>		
18	P1	<_PP2>	REAL	螺紋第二區段的螺距，請參閱 <_PITA>		
19	P2	<_PP3>	REAL	螺紋第三區段的螺距，請參閱 <_PITA>		

號碼	參數遮罩	內部參數	資料類型	含義		
20		<_VARI>	INT	加工		
				單位：	技術	
					1 =	線性進給的外螺紋
					2 =	線性進給的內螺紋
					3 =	縮減進給的外螺紋，切削的斷面保持固定
					4 =	縮減進給的內螺紋，切削的斷面保持固定
				十位數：	已保留	
				百位數：	進給類型	
					0 =	相容性模式：<_IANG>
					1 =	單邊進給
					2 =	側邊交替進給
				千位數：	已保留	
				萬位數：	交替深度進給	
					0 =	相容性，粗加工切削的預設數量 (<_NRC>)
					1 =	第一進給的預設值 (<_MID>)
				十萬位數：	加工類型	
					0 =	相容性（粗加工及精加工）
					1 =	粗加工
					2 =	精加工
					3 =	粗加工與精加工
				百萬位數：	多重啟動螺紋的加工順序	
					0 =	螺紋的遞增順序
					1 =	螺紋的遞減順序
21	N	<_NUMTH>	INT	螺紋圈數		
22		<_VRT>	REAL	返回距離（增量）		
					0 =	內部使用 1 毫米的舉升距離，忽略啟用的系統（英吋或公制）
					> 0 =	舉升距離

號碼	參數遮罩	內部參數	資料類型	含義		
23	D1	<_MID>	REAL	第一進給，請參閱 <_VARI> (TEN THOUSANDS)		
24	DA	<_GDEP>	REAL	螺紋交換深度（僅於「多重啟動」有效）		
					0 =	勿觀察任何螺紋更換深度
					> 0 =	觀察螺紋更換深度
25		<_IFLANK>	REAL	寬度的進給斜率（僅供介面）		
26		<_PITA>	INT	評估螺紋螺距		
					0 =	螺紋的相容性模式，如前述依據生效系統評估 <_PP1> 至 <_PP3>（公制 / 英吋）
					1 =	以毫米為單位的螺距
					2 =	以 TPI（每英吋的螺紋）為單位的螺距
					3 =	以英吋為單位的螺距
					4 =	MODULUS
27		<_PITM1>	STRING[15]	螺距輸入的標記字串（僅供介面使用）		
28		<_PITM2>	STRING[15]	螺距輸入的標記字串（僅供介面使用）		
29		<_PITM3>	STRING[15]	螺距輸入的標記字串（僅供介面使用）		

號碼	參數遮罩	內部參數	資料類型	含義		
30		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	---	已保留
				百位數：	---	已保留
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單

號碼	參數遮罩	內部參數	資料類型	含義		
31		<_AMODE>	INT	替代模式		
				單位：	Z 的第一中間點 (Z1)	
					0 =	絕對值
					1 =	增量
				十位數：	X 的第一中間點 (X1)	
					0 =	絕對值
					1 =	增量
					2 =	α
				百位數：	Z 的第二中間點 (Z2)	
					0 =	絕對值
					1 =	增量
				千位數：	X 的第二中間點 (X2)	
					0 =	絕對值
					1 =	增量
					2 =	α
				萬位數：	Z 的終點 (Z3)	
					0 =	絕對值
					1 =	增量
				十萬位數：	X 的終點 (X3)	
					0 =	絕對值
					1 =	增量
					2 =	α
				百萬位數：	選擇角度或寬度的進給斜率	
					0 =	進給角度 <_IANG>
					1 =	進給斜率 <_IFLANK>
				千萬位數：	單/多螺紋	
					0 =	相容性模式（評估起始角度 <_NSP>）
					1 =	單螺紋（有起始角度偏移 <_NSP>）
					2 =	多重

20.1.30 CYCLE99－螺紋車削

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	Z0	<_SPL>	REAL	參考點（絕對）		
2	X0	<_SPD>	REAL	參考點（絕對，永遠為直徑）		
3	Z1	<_FPL>	REAL	搭配 <_AMODE> (UNITS) 使用的終點		
4	X1	<_FPD>	REAL	搭配 <_AMODE> (TENS) 使用的終點		
5	LW/LW2	<_APP>	REAL	搭配 <_AMODE> (HUNDREDS) 使用的螺紋旋入，或 螺紋旋入＝螺紋旋出，搭配 <_AMODE> (HUNDREDS) 使用		
6	LR	<_ROP>	REAL	螺紋旋出		
7	H1	<_TDEP>	REAL	螺紋深度		
8	U	<_FAL>	REAL	X 及 Z 方向的精加工允差		
9	DP	<_IANG>	REAL	作為距離或角度的進給斜率，搭配 <_AMODE> (THOUSANDS) 使用		
	αP				> 0 =	正號側的進給
					< 0 =	負號側的進給
					0 =	中心進給
10	α0	<_NSP>	REAL	起始角度偏移（僅在「單一起點」時有效）		
11	ND	<_NRC>	INT	粗加工切削次數，配合 <_VARI> (TEN THOUSANDS)		
12	NN	<_NID>	INT	空切削次數		
13	P	<_PIT>	REAL	螺距數值，搭配使用 <_PITA>		

號碼	參數遮罩	內部參數	資料類型	含義		
14		<_VARI>	INT	加工類型		
				單位：	技術	
					1 =	線性進給的外螺紋
					2 =	線性進給的內螺紋
					3 =	縮減進給的外螺紋，切削的斷面保持固定
					4 =	縮減進給的內螺紋，切削的斷面保持固定
				十位數：	已保留	
				百位數：	進給類型	
					1 =	單邊進給
					2 =	側邊交替進給
				千位數：	已保留	
				萬位數：	交替深度進給	
					0 =	粗加工切削的預設數量 (<_NRC>)
					1 =	第一進給的預設值 (<_MID>)
				十萬位數：	加工類型	
					1 =	粗加工
					2 =	精加工
					3 =	粗加工與精加工
				百萬位數：	多重啟動螺紋的加工順序	
					0 =	螺紋的遞增順序
					1 =	螺紋的遞減順序
15	N	<_NUMTH>	INT	螺紋圈數		
16	VR	<_SDIS>	REAL	返回距離，增量		
17	D1	<_MID>	REAL	第一進給深度，搭配 <_VARI> (TEN THOUSANDS) 使用		
18	DA	<_GDEP>	REAL	螺紋交換深度（僅於「多重啟動」有效）		
					0 =	勿觀察任何螺紋更換深度
					> 0 =	觀察螺紋更換深度

號碼	參數遮罩	內部參數	資料類型	含義		
19	G	<_PIT1>	REAL	每轉的螺距變更		
					0 =	螺距固定 (G33)
					> 0 =	螺距增加 (G34)
					< 0 =	螺距減少 (G35)
20		<_FDEP>	REAL	插入深度（輸入時不帶正負號）		
21	N1	<_GST>	INT	起始螺紋 N1 = 1...N，搭配 <_AMODE> (HUNDRED THOUSANDS) 使用		
22		<_GUD>	INT	已保留		
23		<_IFLANK>	REAL	寬度的進給斜率（僅供介面）		
24		<_PITA>	INT	螺距單位（PIT 及/或 MPIT 的評估）		
					0 =	以毫米為單位的螺距 - MPIT/PIT 評估
					1 =	以毫米為單位的螺距 - PIT 評估
					2 =	以 TPI 為單位的螺距 - PIT 評估（每英吋螺距）
					3 =	以英吋為單位的螺距 - PIT 評估
					4 =	MODULE - PIT 評估
25		<_PITM>	STRING[15]	螺距輸入的標記字串（僅供介面使用）[1)]		
26		<_PTAB>	STRING[20]	螺紋表的字串（僅供介面使用）[1)]		
27		<_PTABA>	STRING[20]	螺紋表的字串選擇（僅供介面使用）[1)]		

號碼	參數遮罩	內部參數	資料類型	含義		
28		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	螺紋類型	
					0 =	縱向螺紋
					1 =	平面螺紋
					2 =	錐形螺紋
				百位數：	---	已保留
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：基本

號碼	參數遮罩	內部參數	資料類型	含義		
29		<_AMODE>	INT	替代模式		
				單位：	Z 方向的螺紋長度	
					0 =	絕對值
					1 =	增量
				十位數：	X 方向的螺紋長度	
					0 =	絕對，直徑中的橫向軸數值
					1 =	增量，半徑中的橫向軸數值
					2 =	α
				百位數：	計算逼近 / 旋進路徑 <_APP>	
					0 =	螺紋旋進 <_APP>
					1 =	螺紋旋進 = 螺紋旋出 <_APP> = -<_ROP>
					2 =	指定螺紋旋進路徑 <_APP> = -<_APP>
				千位數：	選擇角度或寬度的進給斜率	
					0 =	進給角度 <_IANG>
					1 =	進給斜率 <_IFLANK>
				萬位數：	單/多螺紋	
					0 =	單螺紋（有起始角度偏移 <_NSP>）
					1 =	多重
				十萬位數：	啟動螺紋 <_GST>	
					0 =	完整加工
					1 =	由此螺紋啟動加工
					2 =	僅加工此螺紋
				百萬位數：	縱向螺紋懸垂補正	
					0 =	分段高度，冠螺紋 XS
					1 =	半徑，冠螺紋 RS
30	XS/RS	<_S_XRS>	REAL	縱向螺紋懸垂補正，搭配使用 <_AMODE>： 百萬位數		

說明

[1)] 參數 <_PITM>, <_PTAB> 及 <_PTABA> 只用在畫面表單螺紋表的螺紋選擇上。
螺紋表無法透過循環執行時間內的循環定義存取。

20.1.31 CYCLE435 - 設定修整器座標系統

語法

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

參數

編號	參數遮罩	內部參數	資料類型	含義		
1		<_T>	STRING[32]	研磨輪的刀具名稱		
2		<_DD>	INT	研磨輪的刀刃編號		
3		<S_TA>	STRING[32]	修整刀具參考點 - 修整刀具名稱		
4		<S_DA>	INT	修整刀具的刀刃編號		
5		<S_AD>	REAL	修整值，直徑		
6		<S_AL>	REAL	修整值，平面		
7		<S_PVD>	REAL	形狀修整偏移量，直徑		
8		<S_PVL>	REAL	形狀修整偏移量，平面		
9		<S_PD>	REAL	形狀修整允差，直徑		
10		<S_PL>	REAL	形狀修整允差，平面		
11		<_AMODE>	INT	替代模式		
				單位：	循環尾端的生效刀具	
					0 =	修整刀具生效
					1 =	轉輪生效

20.1.32 CYCLE495 - 形狀修整

語法

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

參數

編號	參數遮罩	內部參數	資料類型	含義
1		<_T>	STRING[20]	研磨輪的刀具名稱
2		<_DD>	INT	研磨輪的刀刃編號
3		<_SC>	REAL	避免障礙物的舉升距離，增量
4		<_F>	REAL	形狀修整進給率

編號	參數遮罩	內部參數	資料類型	含義		
5		<_VARI>	INT	加工類型		
				單位：	形狀修整類型	
					1 =	與軸平行
					2 =	與輪廓平行
				十位數：	加工方向	
					0 =	拉出 可使用刀刃位置 1 到 4
					1 =	推入 可使用刀刃位置 1 到 4
					2 =	交錯 可使用刀刃位置 1 到 8
					3 =	開始 → 結尾 可使用刀刃位置 1 到 8
					4 =	結尾 → 開始 可使用刀刃位置 1 到 8
				百位數：	進給方向	
					1 =	G18 的進給 X 或 G19 的進給 Y-
					2 =	G18 的進給 X+ 或 G19 的進給 Y+
					3 =	G18 及 G19 的進給 Z-
					4 =	G18 及 G19 的進給 Z+
6		<_D>	REAL	與軸平行的形狀修整類型修飾值		
7		<_DX>	REAL	平行於輪廓的形狀修整類型，其 G18 的修飾值 X 或 G19 的修飾值 Y		
8		<_DZ>	REAL	平行於輪廓的形狀修整類型，其 G18 及 G19 的修飾值 Z		
9		<S_PA>	REAL	形狀修整允差		
10		<S_N>	INT	形狀修整程式之中的衝程數量		

編號	參數遮罩	內部參數	資料類型	含義		
11		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
12		<_AMODE>	INT	替代模式		
				單位：	形狀修整選擇，新 / 繼續	
					1 =	新增
					2 =	繼續
				十位數：	選擇形狀修整允差	
					0 =	從粗輪廓到輪廓最低點
					1 =	從粗輪廓到輪廓最高點
13		<S_FW>	REAL	修整刀具的淨空角度		
14		<S_HW>	REAL	修整刀具的刀把角度		

20.1.33 CYCLE800 - 旋轉

語法

```
CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>,
<_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)
```

參數

編號	參數遮罩	內部參數	資料類型	含義		
1		<_FR>	INT	回退模式：	0 =	無退刀操作
					1 =	回退機械軸 Z
					2 =	回退機械軸 Z 接著 XY
					3 =	保留
					4 =	沿刀具方向之最大回退
					5 =	沿刀具方向之增量回退

編號	參數遮罩	內部參數	資料類型	含義		
2		<_TC>	STRING[32]	旋轉資料單節的名稱：	""	""（無名稱）若僅存在一個旋轉資料單節
					"0"	取消選擇旋轉資料單節（刪除迴轉框架）
3		<_ST>	INT	狀態轉換		
				單位：		
					0 =	全新，刪除迴轉層級並使用目前參數重新計算
					1 =	附加，新增迴轉層級以啟用迴轉層級
				十位數：	追蹤刀具刀尖是 / 否（僅於在調試時建立 SWIVEL 功能時生效）	
					0 =	不追蹤刀具刀尖
					1 =	追蹤刀具刀尖 (TRAORI)
				百位數：	逼近 / 對齊刀具（功能顯示在刀具迴轉的畫面表單中）	
					0 =	不逼近刀具
					1 =	逼近刀具（徑向銑削較佳）
					2 =	對齊**車**刀（設定 B 軸旋轉機構在調試迴轉進行銑削時）
					3 =	對齊**銑**刀（設定 B 軸旋轉機構在調試迴轉進行銑削時）
				千位數：	內部「在 JOG 模式中迴轉」參數	
				萬位數：	請參閱方向參數 <_DIR>	
					0 =	迴轉「是」
					1 =	迴轉「否」、「負」方向 [3]
					2 =	迴轉「否」、「正」方向 [3]
				十萬位數：	請參閱方向參數 <_DIR>	
					0 =	相容性
					1 =	方向選擇「負」最佳化（僅適用使用者介面）[4]
					2 =	方向選擇「正」最佳化（僅適用使用者介面）[4]

編號	參數遮罩	內部參數	資料類型	含義		
4		<_MODE> 5)	INT	迴轉模式：評估迴轉角度及迴轉順序（位元編碼）		
				位元：7 6	0 0：	逐軸迴轉角度 -> 請參閱參數 <_A>, <_B>, <_C>
					0 1：	立體角 -> 請參閱參數 <_A>, <_B> 1)
					1 0：	投影角 -> 請參閱參數 <_A>, <_B>, <_C> 1)
					1 1：	直接旋轉軸迴轉模式 -> 請參閱參數 <_A>, <_B> 1)
				位元：5 4 3 2 1 0 （這些不適用於實心角）	x x x x 0 1	繞著 X 軸的第一旋轉 _A
					x x x x 1 0	繞著 Y 軸的第一旋轉 _A
					x x x x 1 1	繞著 Z 軸的第一旋轉 _A
					x x 0 1 x x	繞著 X 軸的第二旋轉 _B
					x x 1 0 x x	繞著 Y 軸的第二旋轉 _B
					x x 1 1 x x	繞著 Z 軸的第二旋轉 _B
					0 1 x x x x	繞著 X 軸的第三旋轉 _C
					1 0 x x x x	繞著 Y 軸的第三旋轉 _C
					1 1 x x x x	繞著 Z 軸的第三旋轉 _C
5	X0	<_X0>	REAL	旋轉前的參考點 X		
6	Y0	<_Y0>	REAL	旋轉前的參考點 Y		
7	Z0	<_Z0>	REAL	旋轉前的參考點 Z		
8	X(A)	<_A>	REAL	根據參數 <_MODE> 設定的第一次旋轉		
9	Y(B)	<_B>	REAL	根據參數 <_MODE> 設定的第二次旋轉		
10	Z(C)	<_C>	REAL	根據參數 <_MODE> 設定的第三次旋轉		

編號	參數遮罩	內部參數	資料類型	含義		
11	X1	<_X1>	REAL	旋轉後的參考點 X		
12	Y1	<_Y1>	REAL	旋轉後的參考點 Y		
13	Z1	<_Z1>	REAL	旋轉後的參考點 Z		
14	- 或 +	<_DIR>	INT	旋轉軸的起始移動（預設 = -1！）		
					-1 =	旋轉軸 1 或 2 較小值的位置 2)
					+1 =	旋轉軸 1 或 2 較大值的位置 2)
					0 =	勿迴轉（只是計算迴轉框架） 1) 3)
15	FR	<_FR_I>	REAL	刀具方向增量的回退值（增量）		
16		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	在對齊刀具期間表示 beta 值	
					0 =	值
					1 =	箭頭

說明

若下列傳輸參數係間接程式設計（作為參數），畫面表單並未重置： <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

1) 若於調試建立 SWIVEL 功能就可選擇

2) 如果方向參考至旋轉軸 1 或 2 係設定在 IBN SWIVEL 中，可以選擇

如果方向參考是「否」則無選擇欄位

3) 迴轉選擇「否」會以灰色顯示 SD 55221 Bit 0

迴轉「否」、「負」方向對應於 <_DIR> = 0 及 _ST TEN THOUSANDS = 1

迴轉「否」、「正」方向對應於 <_DIR> = 0 及 _ST TEN THOUSANDS = 2

4) 如果有方向參考的旋轉軸位在極位置（位置值等於零），則旋轉軸 1 或 2 的方向選擇也會發生。

5) 編碼範例：按軸順序旋轉，旋轉順序 ZYX

二進制值：00011011 十進位：27

軸識別碼 XYZ 對應於 NC 通道的幾何軸。允許繞著 XYZ 軸個別旋轉。例如，CYCLE800 的單次呼叫不允許繞著 ZXZ 的旋轉順序

20.1.34 CYCLE801 - 網格或框架

語法

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

參數

號碼	參數遮罩	參數 內側	資料類型	含義		
1	X0	<_SPCA>	REAL	沿著第一軸位置圖樣（網格/框架）的參考點（絕對）		
2	Y0	<_SPCO>	REAL	沿著第二軸位置圖樣（網格/框架）的參考點（絕對）		
3	α0	<_STA>	REAL	基本旋轉角度 （至第一軸的角度）	< 0 =	順時針旋轉
					> 0 =	逆時針旋轉
4	L1	<_DIS1>	REAL	直行的距離（與第一軸的距離，輸入時不帶正負號）		
5	L2	<_DIS2>	REAL	直列的距離（與第二軸的距離，輸入時不帶正負號）		

號碼	參數遮罩	參數 內側	資料類型	含義		
6	N1	<_NUM1>	INT	欄數		
7	N2	<_NUM2>	INT	列數		
8		<_VARI>	INT	加工類型		
				單位：	位置樣式	
					0 =	網格
					1 =	框架
				十位數：	已保留	
				百位數：	已保留	
9		<_UMODE>	INT	已保留		
10	αX	<_ANG1>	REAL	第一軸的剪切角（與第一軸傾斜的直線）		
					< 0 =	順時針測量 （0 至 -90 度）
					> 0 =	逆時針測量 （0 至 90 度）
11	αY	<_ANG2>	REAL	第二軸的剪切角（與第二軸傾斜的直線）		
					< 0 =	順時針測量 （0 至 -90 度）
					> 0 =	逆時針測量 （0 至 90 度）
12		<_HIDE>	STRING [200]	隱藏位置 • 最多 198 個字元 • 連續位置數的規格，例如「1,3」（位置 1 與 3 不執行）		
13		<_NSP>	INT	已保留		
14		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

20.1.35 CYCLE802－任意位置

語法

```
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>)
```

參數

號碼	參數遮罩	參數 內側	資料類型	含義		
1		<_XA>	INT	替代所有的 X 位置（9 位數十進位值） 位數： 876543210（數字位置對應於鑽孔位置 Xn）		
				位置值：	1 =	絕對（程式設計的第一個位置一定要絕對）
					2 =	增量
2		<_YA>	INT	替代所有的 Y 位置（9 位數十進位值） 位數： 876543210（數字位置對應於鑽孔位置 Yn）		
				位置值：	1 =	絕對（程式設計的第一個位置一定要絕對）
					2 =	增量
3	X0	<_X0>	REAL	1. X 位置		
4	Y0	<_Y0>	REAL	1. Y 位置		
5	X1	<_X1>	REAL	2. X 位置		
6	Y1	<_Y1>	REAL	2. Y 位置		
7	X2	<_X2>	REAL	3. X 位置		
8	Y2	<_Y2>	REAL	3. Y 位置		
9	X3	<_X3>	REAL	4. X 位置		
10	Y3	<_Y3>	REAL	4. Y 位置		
11	X4	<_X4>	REAL	5. X 位置		
12	Y4	<_Y4>	REAL	5. Y 位置		
13	X5	<_X5>	REAL	6. X 位置		
14	Y5	<_Y5>	REAL	6. Y 位置		

號碼	參數遮罩	參數 內側	資料類型	含義
15	X6	<_X6>	REAL	7. X 位置
16	Y6	<_Y6>	REAL	7. Y 位置
17	X7	<_X7>	REAL	8. X 位置
18	Y7	<_Y7>	REAL	8. Y 位置
19	X8	<_X8>	REAL	9. X 位置
20	Y8	<_Y8>	REAL	9. Y 位置

號碼	參數遮罩	參數 內側	資料類型	含義		
21		<_VARI>	INT	加工		
				百位數：	（僅用於從 Jobshop 呼叫）（目前只有評估 0 及 2）	
					0 =	勿鉗住主軸
					1 =	僅鉗住主軸以 G00 或 G01 垂直插入
					2 =	整個加工操作期間夾鉗主軸
				千位數：	已保留	
				萬位數：	有 / 無旋轉軸的位置樣式 – 軸組合（使用 <_VARI> HUNDRED THOUSANDS）	
					0 =	XY（僅 XY 無旋轉軸，相容）
					1 =	X、Y 或 Z 及旋轉軸： XA、YB、ZC （一個旋轉軸的幾何軸，繞著旋轉軸旋轉）
					2 =	XY 及旋轉軸： XYA、XYB、XYC （一個旋轉軸的第一及第二幾何軸，無 TRACYL）
				十萬位數：	旋轉軸	
					0 =	無旋轉軸 （僅 XY，相容）
					1 =	A 軸（旋轉軸繞著 X）
					2 =	B 軸（旋轉軸繞著 Y）
					3 =	C 軸（旋轉軸繞著 Z）
				千萬位數 + 百萬位數：	旋轉軸的位置樣式 – 偏移量 （用於繞著相同軸的數個旋轉軸；若索引過大，則為第一軸）	
					00 =	第一 A、B 或 C 軸，或用於相容
					01 =	第二 A、B 或 C 軸
					...	

號碼	參數遮罩	參數 內側	資料類型	含義		
					19 =	第 20 A、B 或 C 軸
22		<_UMODE>	INT	選擇欲鉗住的主軸：（僅用於從 Jobshop 呼叫）（呼叫使用者循環 CUST_TECHCYC）		
					3 =	鉗住 / 釋放主軸
					23 =	鉗住 / 釋放副主軸
23		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
24		<S_ABA>	INT	替代所有的 AB 位置（9 位數十進位值） 位數： 876543210（數字位置對應於位置 ABn）		
				位置值：	1 =	絕對（程式設計的第一個位置一定要絕對）
					2 =	增量
25	A0	<S_AB0>	REAL	具有旋轉軸位置樣式的第一旋轉軸位置（搭配 <_VARI>）		
26	A1	<S_AB1>	REAL	具有旋轉軸位置樣式的第二旋轉軸位置		
27	A2	<S_AB2>	REAL	具有旋轉軸位置樣式的第三旋轉軸位置		
28	A3	<S_AB3>	REAL	具有旋轉軸位置樣式的第四旋轉軸位置		
29	A4	<S_AB4>	REAL	具有旋轉軸位置樣式的第五旋轉軸位置		
30	A5	<S_AB5>	REAL	具有旋轉軸位置樣式的第六旋轉軸位置		
31	A6	<S_AB6>	REAL	具有旋轉軸位置樣式的第七旋轉軸位置		
32	A7	<S_AB7>	REAL	具有旋轉軸位置樣式的第八旋轉軸位置		
33	A8	<S_AB8>	REAL	具有旋轉軸位置樣式的第九旋轉軸位置		

說明

不需要位置的參數 X1/Y1/A1 至 X8/Y8/A8 可以忽略。 不過 <_XA>, <_YA> 及 <S_ABA> 所有 9 個位置都必須提供完全的的替代值。

位置樣式 XA、YB 或 ZC（幾何軸及旋轉軸）在加工平面的軸尚未透過位置樣式（Y 用於 G17 及 XA）移動時，必需在循環呼叫以前先定位。

20.1.36 CYCLE830 - 深孔鑽 2

語法

```
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>)
```

參數

編號	參數遮罩	內部參數	資料類型	含義
1	RP	<RTP>	REAL	回退平面（絕對）
2	Z0	<RFP>	REAL	參考點（絕對）
3	SC	<SDIS>	REAL	安全間距（新增至參考點，不帶正負號）
4	Z1	<_DP>	REAL	最終鑽孔深度絕對 / 增量（請參閱 <_AMODE>UNITS）
5	D	<FDEP>	REAL	相對於參考點有 / 無預鑽孔，或相對於先導孔深度，其絕對或增量斷屑 / 排屑的第一鑽孔深度（請參閱 <_AMODE> TEN THOUSANDS）
6	DF	<_DAM>	REAL	各個額外進給的絕對值 / 百分比、縮減絕對值 / 百分比（請參閱 <_AMODE> HUNDRED THOUSANDS）
7	DTB	<DTB>	REAL	每個鑽孔深度的停頓時間（請參閱 <_AMODE> TENS）
8	DTS	<DTS>	REAL	排屑期間在起點處的停頓時間（請參閱 <_AMODE> HUNDREDS）
9	FD1	<FRF>	REAL	第一進給之進給率百分比（請參閱 <_AMODE> TEN MILLION）

編號	參數遮罩	內部參數	資料類型	含義		
10		<VARI>	INT	加工		
				單位：	斷屑/排屑	
					0 =	一次切削
					1 =	斷屑
					2 =	切屑移除
					3 =	斷屑及排屑
				十位數：	在排屑期間回退	
					0 =	至先導孔深度
					1 =	至安全間距
				百位數：	軟第一切削	
					0 =	否
					1 =	是
				千位數：	通鑽	
					0 =	否
					1 =	是
				萬位數：	預鑽孔 / 先導孔	
					0 =	無預鑽孔
					1 =	有預鑽孔
					2 =	有先導孔
				十萬位數：	回退	
					0 =	至先導孔深度
					1 =	至退刀平面
11	V1	<_MDEP>	REAL	最小增量進給（僅供縮減百分比）		
12	V2	<_VRT>	REAL	每次增量加工步驟之後的回退距離（僅供斷屑）		
					0 =	預設值 1 毫米
					> 0 =	可變回退距離
13	DT	<_DTD>	REAL	最終鑽孔深度的停頓時間（請參閱 <_AMODE> THOUSANDS）		
14	V3	<_DIS1>	REAL	增量限制距離，僅供排屑移除（請參閱 <_AMODE> ONEMILLION）		
15	FP	<S_FP>	REAL	移動至先導孔的進給率為數值或以 % 為單位（搭配 <S_AMODE2> HUNDREDS 使用）		

編號	參數遮罩	內部參數	資料類型	含義		
16		<S_SDAC2>	INT	逼近時的主軸旋轉方向		
					3 =	M3
					4 =	M4
					5 =	M5（預設）
17	SP	<S_SV2>	REAL	逼近，	主軸定速（請參閱 <S_AMODE2> TEN MILLION）	
	V4				恆定切削率	
				鑽孔速度的主軸速度，以 % 為單位		
18	F	<S_FB>	REAL	鑽孔進給率（請參閱 <S_AMODE2> UNITS）		
19		<_SDAC>	REAL	鑽孔時的主軸旋轉方向		
					3 =	M3
					4 =	M4
20	S	<_SV1>	REAL	鑽孔，	主軸定速（請參閱 <S_AMODE2> ONE MILLION）	
	V5				恆定切削率	
21	SPOS	<S_SPOS>	REAL	主軸位置，僅以 M5 逼近時		
22	ZA	<S_ZA>	REAL	相對於參考點或絕對的增量預鑽孔深度（請參閱 <S_AMODE3> UNITS）		
23	FA	<S_FA>	REAL	預鑽孔進給率為數值或以 % 為單位（搭配 <S_AMODE2> TENS 使用）		
24	ZP	<S_ZP>	REAL	相對於參考點或絕對或孔徑係數的增量先導孔（請參閱 <S_AMODE3> TENS）		
25	FS	<S_FS>	REAL	第一切削進給率為數值或以 % 為單位（搭配 <S_AMODE2> THOUSANDS 使用）		
26	ZS1	<S_ZS1>	REAL	以固定進給率的每個第一切削深度（增量）		
27	ZS2	<S_ZS2>	REAL	進給率增加的每個第一切削深度（增量）		
28	N	<S_N>	INT	每次排屑之前的斷屑衝程編號		
29	ZD	<S_ZD>	REAL	相對於最終鑽孔深度或絕對的增量剩餘鑽孔深度（請參閱 <S_AMODE3> HUNDREDS）		
30	FD	<S_FD>	REAL	剩餘鑽孔進給率為數值或以 % 為單位（搭配 <S_AMODE2> TEN THOUSANDS 使用）		
31	FR	<S_FR>	REAL	回退進給率（搭配 <S_AMODE2> HUNDRED THOUSANDS 使用）		

編號	參數遮罩	內部參數	資料類型	含義		
32		<S_SDAC3>	INT	回退時的主軸旋轉方向		
					3 =	M3
					4 =	M4
					5 =	M5
33	SR	<S_SV3>	REAL	回退，	主軸定速（請參閱 <S_AMODE2> HUNDRED MILLION）	
	V6				恆定切削率	
				鑽孔速度的主軸速度，以 % 為單位		
34	冷卻液啟動	<S_CON>	STRING[10]	冷卻液開啟，M 指令或子程式呼叫		
35	冷卻液關閉	<S_COFF>	STRING[10]	冷卻液關閉，M 指令或子程式呼叫		
36		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	保留	
				十位數：	相對於刀尖/刀柄的鑽孔	
					0 =	刀尖
					1 =	刀柄
37		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	保留	
				百位數：	保留	
				千位數：	保留	
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：基本

編號	參數遮罩	內部參數	資料類型	含義		
38		<_AMODE>	INT	替代模式 1		
				單位：	鑽孔深度 = 最終鑽孔深度 Z1 絕對/增量	
					0 =	增量
					1 =	絕對值
				十位數：	每個鑽孔深度 DTB 停頓時間，以秒/轉為單位	
					0 =	以秒為單位
					1 =	以轉為單位
				百位數：	排屑 DTS 停頓時間，以秒/轉為單位	
					0 =	以秒為單位
					1 =	以轉為單位
				千位數：	最終鑽孔深度 DT 停頓時間，以秒/轉為單位	
					0 =	以秒為單位
					1 =	以轉為單位
				萬位數：	第一鑽孔深度 D 絕對 / 增量	
					0 =	增量
					1 =	絕對值
				十萬位數：	每次額外進給的絕對值 / 百分比（縮減）	
					0 =	絕對值
					1 =	百分比（0.001 至 100%）
				百萬位數：	限制距離 V3 自動/手動	
					0 =	自動（在循環中計算）
					1 =	手動（已設定數值）

編號	參數遮罩	內部參數	資料類型	含義		
39		<S_AMODE 2>	INT	替代模式 2		
				單位：	單位：鑽孔進給率 F	
					0 =	F / 分
					1 =	F / 轉
				十位數：	評估預鑽孔的進給率 FA	
					0 =	以 % 表示鑽孔進給率
					1 =	F / 分
					2 =	F / 轉
				百位數：	評估移動至先導孔 FP 內的進給率	
					0 =	以 % 表示鑽孔進給率
					1 =	F / 分
					2 =	F / 轉
				千位數：	評估第一切削的進給率 FS	
					0 =	以 % 表示鑽孔進給率
					1 =	F / 分
					2 =	F / 轉
				萬位數：	評估通鑽的進給率 FD	
					0 =	以 % 表示鑽孔進給率
					1 =	F / 分
					2 =	F / 轉
				十萬位數：	回退進給率 FR	
					0 =	F / 分
					1 =	快速移動
				百萬位數：	鑽孔 - 主軸速度 / 切削率 (S/V5)	
					0 =	主軸定速
					1 =	恆定切削率
				千萬位數：	以主軸速度 / 切削率逼近 (SP/V4)	
					0 =	主軸定速
					1 =	恆定切削率
					2 =	鑽孔速度的主軸速度，以 % 為單位

編號	參數遮罩	內部參數	資料類型	含義		
				億位數：	回退 - 主軸速度 / 切削率 (SR/V6)	
					0 =	主軸定速
					1 =	恆定切削率
					2 =	鑽孔速度的主軸速度，以 % 為單位
40		<S_AMODE3>	INT	替代模式 3		
				單位：	鑽孔深度 ZA 絕對/增量	
					0 =	增量
					1 =	絕對值
				十位數：	先導孔的深度 ZP	
					0 =	增量
					1 =	絕對值
					2 =	孔徑係數
				百位數：	剩下的鑽孔深度 ZD 絕對/增量	
					0 =	增量
					1 =	絕對值
41	ZPV	<S_ZPV>	REAL	自先導孔深度的增量限制距離		

20.1.37 CYCLE832 - 高速設定

語法

```
CYCLE832(<S_TOL>, <S_TOLM>, <S_OTOL>)
```

說明

機台試俥時，CYCLE832 並不能免除機台製造商必要的最佳化工作。這牽涉到涉入加工程序與 NCU 設定的軸最佳化（預先控制、抑制震動等）。

參數

編號	參數遮罩	內部參數	資料類型	含義		
1	允差	<S_TOL>	REAL	輪廓允差 對應於幾何座標軸之軸允差的輪廓允差。		
2		<S_TOLM>	INT	加工類型（技術）		
				單位：		
					0 =	取消選擇
					1 =	精加工
					2 =	半精加工
					3 =	粗加工
				十位數：		
					0 =	相容性 [1] 或無方向允差
					1 =	<S_OTOL> 參數的方向允差
				百位數 ... 十萬位數	指定 有鑑於此 相容性	
				百萬位數：		
					0 =	相容性。自動使用最好的模具製作函數： • 選用的上表面未啟用： ☐ 先進表面加工 • 選用的上表面啟用： ⇒ 平滑化上表面
					1 =	無平滑化上表面
					2 =	平滑化上表面
3	ORI 允差	<S_OTOL>	REAL	方向允差或版本識別碼 CYCLE832 工件方向的允差參數 在動態方向轉換機台上執行高速加工程式時會需要（例如 5 軸加工）。 參數 <S_OTOL> **必須**程式設計，也適用於無刀具方向程式的 3 軸機台應用 (<S_OTOL> = 1)。		

1) 方向允差是從輪廓允差乘以循環設定資料 SD55441 至 SD55443 的參數，所推算得到的。

參考資料：

調試手冊，基礎軟體及操作軟體；SINUMERIK Operate (IM9)，章節「高速加工設定功能組態規劃 (CYCLE832)」

純文字欄位

為增進循環呼叫的可讀性，參數 <S_TOLM>（加工類型）也可用純文字輸入。純文字與任何語言無關。下列為認可的輸入：

_OFF	用於	0	取消選擇
_FINISH	用於	1	精加工
_SEMIFIN	用於	2	粗-精加工
_ROUGH	用於	3	粗加工
_ORI_FINISH	用於	11	精加工，有方向允差輸入
_ORI_SEMIFIN	用於	12	精加工，有方向允差輸入
_ORI_ROUGH	用於	13	粗加工，有方向允差輸入
_TOP_SURFACE_SMOOTH_OFF	用於	1000000	無平滑化上表面
_TOP_SURFACE_SMOOTH_ON	用於	2000000	平滑化上表面

上表面的純文字輸入，純文字的合併如以下範例所示：

```
CYCLE832(0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

說明

純文字根據 G 群組 59（路徑插補的動態模式）的功能名稱。3 軸機台和多軸方向轉換 (TRAORI) 機台，用純文字可在應用中清楚區分。

取消選取 CYCLE832

取消選取 CYCLE832 時，參數 <S_TOL> 必須以零傳送。

範例：CYCLE832(0,0,1)

語法 CYCLE832() 亦允許用於取消選取 CYCLE832。

範例

範例 1：CYCLE832 用於無方向轉換的 3 軸機台

a) 純文字輸入的循環呼叫

程式碼	註解
G710	；尺寸系統為公制。
CYCLE832(0.004,_FINISH,1)	；CYCLE832 呼叫，透過： 輪廓允差 = 0.004 毫米，加工類型：精加工
...	；執行高速加工程式

b) 循環呼叫無純文字輸入

程式碼	註解
G710	；請參考上述
CYCLE832(0.004,1,1)	；請參考上述
...	；請參考上述

範例 2：CYCLE832 用於有方向轉換的 5 軸機台

a) 純文字輸入的循環呼叫與取消選擇

程式碼	註解
G710	；尺寸系統為公制。
TRAORI	；啟動方向轉換。
CYCLE832(0.3,_ORI_ROUGH,0.8)	；CYCLE832 呼叫，透過： 輪廓允差 = 0.3 毫米，加工類型：有方向允差輸入的粗加工； 方向允差 = 0.8 度
...	；執行高速加工程式
CYCLE832(0,_OFF,1)	；輪廓允差 = 0， 加工類型：取消選取 CYCLE832， 方向允差 = 0 度

a) 無純文字輸入的循環呼叫與取消選擇

程式碼	註解
G710	；請參考上述
TRAORI	；請參考上述
CYCLE832(0.3,13,0.8)	；請參考上述
...	；請參考上述
CYCLE832(0,0,1)	；請參考上述

20.1.38 CYCLE840 - 搭配補正夾頭進行攻牙

語法

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>,
<_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義			
1	RP	<RTP>	REAL	回退平面（絕對）			
2	Z0	<RFP>	REAL	參考點（絕對）			
3	SC	<SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）			
4	Z1	<DP>	REAL	鑽孔深度（絕對），請參閱 <_AMODE>			
5	Z1	<DPR>	REAL	鑽孔深度（增量），請參閱 <_AMODE>			
6	DT	<DTB>	REAL	回退後鑽孔深度 / 安全間距以秒為單位的停頓時間，請參閱 <ENC>			
7		<SDR>	INT	退刀旋轉方向			
8	SDE	<SDAC>	INT	循環結束後的旋轉方向			
9		<ENC>	INT	使用主軸編碼器攻牙 (G33)/不使用主軸編碼器攻牙 (G63)			
					0 =	使用主軸編碼器	- 螺距來自 <MPIT>/ <PIT> - 無 DT
					20 =	使用主軸編碼器	- 螺距來自 <MPIT>/ <PIT> - 回退至安全間距後有 DT
					11 =	不使用主軸編碼器	- 螺距來自 <MPIT>/ <PIT> - 鑽孔深度有 DT
					1 =	不使用主軸編碼器	- 螺距來自程式設計的進給率 - 鑽孔深度有 DT（進給率 = 速率 · 螺距）

號碼	參數遮罩	內部參數	資料類型	含義			
10		<MPIT>	REAL	僅用於「ISO 公制」的螺紋尺寸（螺距在執行時間時內部計算） 值域： 3 至 48（M3 至 M48），替代 <PIT>			
11		<PIT>	REAL	螺距數值，單位請參閱 <_PITA> 值域： > 0，替代 MPIT			
12		<_AXN>	INT	鑽孔軸	0 =	第三幾何座標軸	
					1 =	第一幾何座標軸	
					2 =	第二幾何座標軸	
					≥ 3 =	第三幾何座標軸	
13		<_PITA>	INT	螺距單位（ <PIT> 及 <MPIT> 的評估）			
					0 =	以毫米為單位的螺距	- 評估 <MPIT>/<PIT>
					1 =	以毫米為單位的螺距	- 評估 <PIT>
					2 =	以 TPI 為單位的螺距	- <PIT> 的評估（每英吋的螺紋）
					3 =	以英吋為單位的螺距	- 評估 <PIT>
					4 =	MODULUS	- 評估 <PIT>
14		<_TECHNO>	INT	技術 [1]			
				單位：	精確停止反應		
					0 =	循環呼叫前啟用精確停止反應	
					1 =	精確停止 G601	
					2 =	精確停止 G602	
					3 =	精確停止 G603	
				十位數：	前饋控制		
					0 =	循環呼叫前，啟用有 / 無前饋控制	
					1 =	具前饋控制 FFWON	
					2 =	無前饋控制 FFWOF	
15		<_PITM>	STRING[15]	螺距輸入的標記字串 [2]			
16		<_PTAB>	STRING[5]	螺紋表的字串（""、"ISO"、"BSW"、"BSP"、"UNC"）[2]			

號碼	參數遮罩	內部參數	資料類型	含義		
17		<_PTABA>	STRING[20]	從螺紋表選擇的字串（例如 "M 10"、"M 12"，...）[2]		
18		<_GMODE>	INT	已保留		
19		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	已保留	
				百位數：	已保留	
				千位數：	相容性模式（或僅重新編譯畫面表單），若 MD 52216 bit0 = 1[1]	
					0 =	顯示技術參數（相容性）：TECHNO 參數有效
					1 =	不顯示技術參數：「循環呼叫前」啟用技術
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單
20		<_AMODE>	INT	替代模式		
				單位：	鑽孔深度 Z1（絕對/增量）	
					0 =	相容性，來自程式設計 <DP>/<DPR>
					1 =	增量
					2 =	絕對值

[1] 根據設定日期 SD52216 MCS_FUNCTION_MASK_DRILL，技術欄位可以隱藏

[2] 參數 15、16 及 17 只用在畫面表單螺紋表的螺紋選擇上。 螺紋表無法透過循環執行時間內的循環定義存取。

20.1.39 CYCLE899－銑削開孔槽

語法

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>,
<_CDIR>, <_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>,
<_ZFS>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	RP	<_RTP>	REAL	回退平面（絕對）		
2	Z0	<_RFP>	REAL	刀具軸參考點（絕對）		
3	SC	<_SDIS>	REAL	安全間距（新增至參考點，輸入時不帶正負號）		
4	Z1	<_DP>	REAL	槽深度（絕對 / 增量），請參閱 <_AMODE>		
5	L	<_LENG>	REAL	槽長度（增量）		
6	W	<_WID>	REAL	槽寬度（增量）		
7	X0	<_PA>	REAL	參考點/起點第一軸（絕對）		
8	Y0	<_PO>	REAL	參考點/起點第二軸（絕對）		
9	α0	<_STA>	REAL	相對於第一軸的旋轉角度		
10	DZ	<_MID>	REAL	最大進給深度（增量）– 僅適用於渦輪銑削		
11	DXY	<_MIDA>	REAL	最大平面進給，請參閱 <_AMODE>		
12	UXY	<_FAL>	REAL	精加工允差量，平面		
13	UZ	<_FALD>	REAL	精加工允差，深度		
14	F	<_FFP1>	REAL	進給率		
15		<_CDIR>	INT	銑削方向		
				單位：		
					0 =	下切
					1 =	上切
					4 =	交錯

號碼	參數遮罩	內部參數	資料類型	含義		
16		<_VARI>	INT	加工		
				單位：		
					1 =	粗加工
					2 =	精加工
					3 =	底部精加工
					4 =	邊緣精加工
					5 =	粗-精加工
					6 =	倒角
				十位數：	已保留	
				百位數：	已保留	
				千位數：		
					1 =	渦旋銑削
					2 =	直進切削
17		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工/僅計算起點	
					1 =	正常加工
				千位數：	透過中心/稜邊尺寸標註	
					0 =	透過中心尺寸標註
					1 =	使用稜邊「左手邊」尺寸標註（第一軸的「-」方向）
					2 =	使用稜邊「右手邊」尺寸標註（第一軸的「+」方向）

號碼	參數遮罩	內部參數	資料類型	含義		
18		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	---	已保留
				百位數：	---	已保留
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單
19		<_AMODE>	INT	替代模式		
				單位：	槽深 Z1	
					0 =	絕對值
					1 =	增量
				十位數：	平面進給的單位 (<_MIDA>) DXY	
					0 =	mm
					1 =	刀具直徑 %
				百位數：	倒角的插入深度 ZFS	
					0 =	絕對值
					1 =	增量
20		<_UMODE>	INT	已保留		
21	FS	<_FS>	REAL	倒角寬度（增量）		
22	ZFS	<_ZFS>	REAL	倒角的插入深度（刀尖）（絕對 / 增量），請參閱 <_AMODE>		

20.1.40 CYCLE930－槽

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義		
1	X0	<_SPD>	REAL	平面軸中的參考點（永遠為直徑）		
2	Z0	<_SPL>	REAL	參考點沿著縱向軸		
3	B1	<_WIDG>	REAL	槽底端的寬度		
4	B2	<_WIDG2>	REAL	槽頂端的寬度（僅供介面）		
5	T1	<_DIAG>	REAL	參考點處的槽深度 絕對及縱向加工 = 直徑，否則為增量		
6	T2	<_DIAG2>	REAL	與參考點反向的槽深度（僅供介面）， 絕對及縱向加工 = 直徑，否則為增量		
7	α0	<_STA>	REAL	傾斜角 (-180 ≤ <_STA> ≤ 180)		
8	α1	<_ANG1>	REAL	側角 1 (0 ≤ <_ANG1> < 90) 位於由參考點決定之槽的側面		
9	α2	<_ANG2>	REAL	側角 2 (0 ≤ <_ANG2> < 90) 與參考點相反		
10	R1/FS1	<_RCO1>	REAL	倒圓角半徑或倒角寬度 1，參考點外側		
11	R2/FS2	<_RCI1>	REAL	倒圓角半徑或倒角寬度 2，參考點內側		
12	R3/FS3	<_RCI2>	REAL	倒圓角半徑或倒角寬度 3，參考點內側反向		
13	R4/FS4	<_RCO2>	REAL	倒圓角半徑或倒角寬度 4，參考點外側反向		
14	U	<_FAL>	REAL	X 及 Z 方向的精加工允差，請參閱 <_VARI> (TEN THOUSANDS)（輸入時不帶正負號）		
15	D	<_IDEP1>	REAL	插入的最大深度進給（輸入時不帶正負號）		
					0 =	1. 直接切削至完全深度
					> 0 =	1. 切削 <_IDEP1>、第二切削 2 · <_IDEP1> 等
16	SC	<_SDIS>	REAL	安全間距（輸入時不含正負號）		

號碼	參數遮罩	內部參數	資料類型	含義		
17		<_VARI>	INT	加工類型		
				單位：	已保留	
				十位數：	加工程序	
					1 =	粗加工
					2 =	精加工
					3 =	粗加工與精加工
				百位數：	位置縱向/橫向外部/內部 +Z/+Z 及 +X/-X	
					1 =	縱向 / 外部 +Z
					2 =	橫向 / 內部 -X
					3 =	縱向 / 內部 +Z
					4 =	橫向 / 內部 +X
					5 =	縱向 / 外部 -Z
					6 =	橫向 / 外部 -X
					7 =	縱向 / 內部 -Z
					8 =	橫向 / 外部 +X
				千位數：	參考點位置	
					0 =	上方參考點
					1 =	下方參考點
				萬位數：	精加工允差的定義效果	
					0 =	平行於輪廓的精加工允差 U
					1 =	UX 及 UZ 個別的精加工允差
18		<_DN>	INT	刀具第二稜邊的 D 編號		
					> 0 =	切槽刀具第二稜邊刀具偏移量 D 編號
					0 =	並未編程設定第二稜邊
19	N	<_NUM>	INT	槽數（0 = 1 槽）		
20	DP	<_DBH>	REAL	槽之間的距離（僅在 <_NUM> > 1 時需要）		
21	F	<_FF1>	REAL	進給率		

號碼	參數遮罩	內部參數	資料類型	含義		
22		<_NR>	INT	槽類型對應於選擇形狀縱向軟鍵的識別		
					0 =	90° 側邊無倒角 / 倒圓角
					1 =	傾斜邊有倒角 / 倒圓角（無 α0）
					2 =	如 1，但是在錐形上（有 α0）
23	UX	<_FALX>	REAL	X 軸的精加工允差，請參閱 <_VARI> (TEN THOUSANDS)（輸入時不帶正負號）		
24	UZ	<_FALZ>	REAL	Z 軸的精加工允差，請參閱 <_VARI> (TEN THOUSANDS)（輸入時不帶正負號）		
25		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）

號碼	參數遮罩	內部參數	資料類型	含義		
26		<_AMODE>	INT	替代模式		
				單位：	槽頂端的尺寸標註（僅供介面）	
					0 =	位於參考點
					1 =	參考點反向
				十位數：	深度	
					0 =	絕對值
					1 =	增量
				百位數：	寬度的尺寸標註（僅供介面）	
					0 =	在外徑上（頂端）
					1 =	在內徑上（底端）
				千位數：	半徑 / 倒角 1 (<_RCO1>)	
					0 =	半徑
					1 =	倒角
				萬位數：	半徑 / 倒角 2 (<_RCI1>)	
					0 =	半徑
					1 =	倒角
				十萬位數：	半徑 / 倒角 3 (<_RCI2>)	
					0 =	半徑
					1 =	倒角
				百萬位數：	半徑 / 倒角 4 (<_RCO2>)	
					0 =	半徑
					1 =	倒角

20.1.41 CYCLE940 - 底切形狀

各種底切可以使用 CYCLE940 循環程式設計。在某些情況下，這些會因為參數設定而有顯著的不同。

表格中多餘的欄代表各種底切類型要求的參數。這些對應於畫面表單中的垂直選擇軟鍵：

- E：底切形狀 E
- F：底切形狀 F

- A-D：DIN 螺紋底切（形狀 A-D）
- T：螺紋底切（形狀任意定義）

語法

```
CYCLE940(<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>,
<_VARI>, <_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>,
<_FAL>, <_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>,
<_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	形狀的程式設計				含義		
				E	F	A-D	T			
1	X0	<_SPD>	REAL	x	x	x	x	平面軸中的參考點（永遠為直徑）		
2	Z0	<_SPL>	REAL	x	x	x	x	縱向軸參考點（絕對）		
3	FORM	<_FORM>	CHAR	x	x	x	x	底切的形狀（大寫字母，例如「T」） 選擇，採用該底切數值的表格		
									A =	外部，參考 DIN76， A = 正常
									B =	外部，參考 DIN76， B = 短
									C =	內部，參考 DIN76， C = 正常
									D =	內部，參考 DIN76， D = 短
									E =	參考 DIN509
									F =	參考 DIN509
									T=	自由
4	位置	<_LAGE>	INT	x	x	x	x	底切的位置（平行 Z 軸）	0 =	外部 +Z: ____|
									1 =	外部 -Z: |____/
									2 =	內部 +Z: /-----|
									3 =	內部 -Z: |-----\
5	SC	<_SDIS>		x	x	x	x	安全間距（增量）		

號碼	參數遮罩	內部參數	資料類型	形狀的程式設計				含義		
6	F	<_FFP>		x	x	x	x	加工進給率（毫米/轉）		
7		<_VARI>	INT	-	-	x	x	加工類型		
								單位：	加工	
									1 =	粗加工
									2 =	精加工
									3 =	粗加工 + 精加工
								十位數：	加工策略	
									0 =	與輪廓平行
									1 =	縱向
				底切形狀 E 及 F 永遠像精加工一樣單次移動加工。						
8	X1	<_EPD>		x	x	-	-	允差 X（絕對 / 增量），請參閱 <_AMODE>		
				-	-	-	x	底切深度（絕對 / 增量），請參閱 <_AMODE>		
9	Z1	<_EPL>		-	x	-	-	允差 Z		
				-	-	-	x	底切寬度（絕對 / 增量），請參閱 <_AMODE>		
10	R1	<_R1>		-	-	-	x	斜面的倒圓角半徑		
11	R2	<_R2>		-	-	-	x	角隅的倒圓角半徑		
12	α	<_STA>		-	-	x	x	插入角度		
13	VX	<_VRT>		x	x	-	-	交叉進給 X（絕對 / 增量），請參閱 <_AMODE>		
				-	-	x	x	精加工時的交叉進給 X（絕對 / 增量），請參閱 <_AMODE>		
14	D	<_MID>		-	-	x	x	深度進給：		
15	U	<_FAL>		-	-	x	x	平行於輪廓的精加工允差，請參閱 <_AMODE>		
16	UX	<_FALX>		-	-	x	x	精加工允差 X		
17	UZ	<_FALZ>		-	-	x	x	精加工允差 Z		

號碼	參數遮罩	內部參數	資料類型	形狀的程式設計	含義			
18	P	<_PITI>	INT	- - x -	選擇螺距，形狀 A-D，對應於 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 - -	選擇半徑/深度，形狀 E、F			
					0 = 0.6 x 0.3 1 = 1.0 x 0.4 2 = 1.0 x 0.2 3 = 1.6 x 0.3	4 = 2.5 x 0.4 5 = 4.0 x 0.5 6 = 0.4 x 0.2 7 = 0.6 x 0.2	8 = 0.1 x 0.1 9 = 0.2 x 0.1	
19		<_PTAB>	STRING [5]		螺紋表的字串（""、"ISO"、"BSW"、"BSP"、"UNC"） （僅供表面使用）			
20		<_PTABA>	STRING [20]		從螺紋表選擇的字串 （例如「M 10」、「M 12」、...） （僅適用表面）			
21		<_DMODE>	INT		顯示模式			
				x x x x	單位：	加工平面 G17/G18/G19		
						0 =	相容性，在循環呼叫保持啟用前的平面生效	
						1 =	G17（僅於循環生效）	
						2 =	G18（僅於循環生效）	
						3 =	G19（僅於循環生效）	

號碼	參數遮罩	內部參數	資料類型	形狀的程式設計				含義		
22		<_AMODE>	INT					替代模式		
				x	x	-	x	單位：	參數 <_EPD> 允差 X 或底切深度	
									0 =	絕對（永遠為直徑）
									1 =	增量
				x	x	-	x	十位數：	參數 <_EPL>　允差 Z 或底切寬度	
									0 =	絕對值
									1 =	增量
				x	x	x	x	百位數：	參數 <_VRT> 交叉進給 X	
									0 =	絕對（永遠為直徑）
									1 =	增量
				-	-	x	x	千位數：	精加工允差	
									0 =	平行於輪廓的精加工允差 (<_FAL>)
									1 =	個別的加工允差 (<_FALX>/<_FALZ>)

20.1.42 CYCLE951－材料移除

語法

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>,
<_SDIS>, <_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	X0	<_SPD>	REAL	參考點（絕對，永遠為直徑）
2	Z0	<_SPL>	REAL	參考點（絕對）
3	X1	<_EPD>	REAL	終點
4	Z1	<_EPL>	REAL	終點

號碼	參數遮罩	內部參數	資料類型	含義		
5	XM α1 α2	<_ZPD>	REAL	中間點，請參閱 <_DMODE> (TENS)		
6	ZM α1 α2	<_ZPL>	REAL	中間點，請參閱 <_DMODE> (TENS)		
7	位置	<_LAGE>	INT	材料移除角隅的位置	0 =	外部 / 背面
					1 =	外部 / 正面
					2 =	內部 / 背面
					3 =	內部 / 正面
8	D	<_MID>	REAL	插入的最大深度進給		
9	UX	<_FALX>	REAL	X 的精加工允差		
10	UZ	<_FALZ>	REAL	Z 的精加工允差		
11		<_VARI>	INT	加工類型		
				單位：	座標系統中的外徑車削方向（縱向或橫向）	
					1 =	縱向
					2 =	平面
				十位數：		
					1 =	粗加工直到達到最終加工允差
					2 =	精加工
				百位數：	保留	
				千位數：	保留	
				萬位數：	保留	
12	R1/FS1	<_RF1>	REAL	倒圓角半徑或倒角寬度 1，請參閱 <_AMODE> (TEN THOUSANDS)		
13	R2/FS2	<_RF2>	REAL	倒圓角半徑或倒角寬度 2，請參閱 <_AMODE> (HUNDRED THOUSANDS)		
14	R3/FS3	<_RF3>	REAL	倒圓角半徑或倒角寬度 3，請參閱 <_AMODE> (ONE MILLION)		
15	SC	<_SDIS>	REAL	安全間隙		
16	F	<_FF1>	REAL	粗加工/精加工的進給率		

號碼	參數遮罩	內部參數	資料類型	含義		
17		<_NR>	INT	材料移除類型的識別（對應於選擇格式的縱向軟鍵）：		
					0 =	材料移除 1，90 度角隅，無倒角 / 倒圓角
					1 =	材料移除 2，90 度角隅，有倒角 / 倒圓角
					2 =	材料移除 3，任一角隅，有倒角 / 倒圓角
18		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	輸入格式 <_ZPD>/<_ZPL>	
					0 =	Xm/Zm
					1 =	Xm/α1
					2 =	Xm/α2
					3 =	α1/Zm
					4 =	α2/Zm
					5 =	α1/α2

號碼	參數遮罩	內部參數	資料類型	含義		
21		<_AMODE>	INT	替代模式		
				單位：	X 的中間點	
					0 =	絕對，直徑中的橫向軸數值
					1 =	增量，半徑中的橫向軸數值
				十位數：	Z 的中間點	
					0 =	絕對值
					1 =	增量
				百位數：	X 的終點	
					0 =	絕對，直徑中的橫向軸數值
					1 =	增量，半徑中的橫向軸數值
				千位數：	Z 的終點。	
					0 =	絕對值
					1 =	增量
				萬位數：	半徑/倒角 1	
					0 =	半徑
					1 =	倒角
				十萬位數：	半徑/倒角 2	
					0 =	半徑
					1 =	倒角
				百萬位數：	半徑/倒角 3	
					0 =	半徑
					1 =	倒角

20.1.43 CYCLE952 - 輪廓切槽

語法

```
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>)
```

參數

號碼	參數遮罩	內部參數	資料類型	含義
1	PRG	<_PRG>	STRING[100]	材料移除程式的名稱
2	CON	<_CON>	STRING[100]	讀取毛胚更新輪廓的程式名稱（用於剩餘加工）
3	CONR	<_CONR>	STRING[100]	寫入毛胚更新輪廓的程式名稱（請參閱 <_AMODE> TEN THOUSANDS）

號碼	參數遮罩	內部參數	資料類型	含義		
4		<_VARI>	INT	加工類型		
				單位：	材料移除的類型	
					1 =	縱向
					2 =	平面
					3 =	與輪廓平行
				十位數：	加工程序（請參閱 <_GMODE> HUNDREDS）	
					1 =	粗加工
					2 =	精加工
					3 =	已保留
					4 =	粗加工，雙通道
					5 =	精加工，雙通道
				百位數：	加工方向	
					1 =	加工方向 X -
					2 =	加工方向 X +
					3 =	加工方向 Z -
					4 =	加工方向 Z +
				千位數：	進給方向	
					1 =	外部 X -
					2 =	內部 X +
					3 =	正面 Z -
					4 =	背面 Z +
				萬位數：	精加工允差的定義效果	
					0 =	UX 及 UZ 個別的精加工允差
					1 =	平行於輪廓的精加工允差 U
				十萬位數：	倒圓角	
					0 =	相容性，自動倒圓角
					1 =	輪廓有倒圓角
					2 =	無倒圓角
					3 =	自動倒圓角
				百萬位數：	退切	

號碼	參數遮罩	內部參數	資料類型	含義		
					0 =	切槽時未評估位置，- 剩餘及槽車削，- 剩餘部份
					1 =	機械離隙切削
					2 =	無凸紋切削加工
				千萬位數：	車削中心之後/之前	
					0 =	於車削中心前方加工
					1 =	已保留
5	F	<_F>	REAL	粗加工/精加工的進給率		
	FZ			進給橫座標槽車削		
6	FR	<_FR>	REAL	插入至離隙車削的進給率，粗加工		
	FX			進給縱座標槽車削		
7	RP	<_RP>	REAL	內部加工的回退平面（絕對，永遠為直徑）		
8	D	<_D>	REAL	粗加工進給（請參閱 <_AMODE> UNITS）		
9	DX	<_DX>	REAL	X 進給（請參閱 <_AMODE> UNITS）		
10	DZ	<_DZ>	REAL	Z 進給（請參閱 <_AMODE> UNITS）		
11	UX	<_UX>	REAL	精加工允差 X，（請參閱 <_VARI> TEN THOUSANDS）		
12	UZ	<_UZ>	REAL	精加工允差 Z，（請參閱 <_VARI> TEN THOUSANDS）		
13	U	<_U>	REAL	平行於輪廓的精加工允差，（請參閱 <_VARI> TEN THOUSANDS）		
14	U1	<_U1>	REAL	精加工時其他的精加工允差（請參閱 <_AMODE> THOUSANDS）		
15	BL	<_BL>	INT	毛胚的定義	1 =	有允差的柱面
					2 =	最終零件的輪廓允差
					3 =	已指定毛胚輪廓
16	XD	<_XD>	REAL	定義毛胚 X（請參閱 <_AMODE> HUNDRED THOUSANDS）		
17	ZD	<_ZD>	REAL	定義毛胚 Z（請參閱 <_AMODE> ONE MILLION）		
18	XA	<_XA>	REAL	臨界值 1 X（絕對，永遠為直徑）		
19	ZA	<_ZA>	REAL	臨界值 1 Z（絕對）		
20	XB	<_XB>	REAL	臨界值 2 X（請參閱 <_AMODE> TEN MILLIONS）		
21	ZB	<_ZB>	REAL	臨界值 2 Z（請參閱 <_AMODE> HUNDRED MILLIONS）		
22	XDA	<_XDA>	REAL	切槽在前平面上的第一槽臨界值 1（絕對，永遠為直徑）		
23	XDB	<_XDB>	REAL	切槽在前平面上的第一槽臨界值 2（絕對，永遠為直徑）		

號碼	參數遮罩	內部參數	資料類型	含義		
24	N	<_N>	INT	槽數		
25	DP	<_DP>	REAL	溝槽之間的距離	長形槽：平行於 Z 軸	
					橫向槽：平行於 X 軸	
26	DI	<_DI>	REAL	中斷進給的距離	0 =	無中斷
					> 0 =	有中斷
27	SC	<_SC>	REAL	避免障礙物的安全間距，增量		
28	D2	<_DN>	INT	第二刀刃的 D 編號；如未程式設計 ⇒ D+1		

號碼	參數遮罩	內部參數	資料類型	含義		
29		<_GMODE>	INT	幾何模式（評估程式設計的幾何資料）		
				單位：	已保留	
				十位數：	已保留	
				百位數：	選擇加工/僅計算起點	
					0 =	正常加工（不需要相容性模式）
					1 =	正常加工
					2 =	計算起點 - 無加工（僅用於從 ShopMill/ShopTurn 呼叫）
				千位數：	極限	
					0 =	否
					1 =	是
				萬位數：	輸入臨界值 1 X	
					0 =	否
					1 =	是
				十萬位數：	輸入臨界值 2 X	
					0 =	否
					1 =	是
				百萬位數：	輸入臨界值 1 Z	
					0 =	否
					1 =	是
				千萬位數：	輸入臨界值 2 Z	
					0 =	否
					1 =	是

號碼	參數遮罩	內部參數	資料類型	含義		
30		<_DMODE>	INT	顯示模式		
				單位：	加工平面 G17/G18/G19	
					0 =	相容性，在循環呼叫保持啟用前的平面生效
					1 =	G17（僅於循環生效）
					2 =	G18（僅於循環生效）
					3 =	G19（僅於循環生效）
				十位數：	技術模式	
					1 =	沿著輪廓進行材料移除
					2 =	輪廓溝槽
					3 =	溝槽車削
				百位數：	加工殘料	
					0 =	否
					1 =	是
				千位數：	---	已保留
				萬位數：	循環畫面表單的技術刻度比例 (頁 915)	
					0 =	輸入：完整
					1 =	輸入：簡單

號碼	參數遮罩	內部參數	資料類型	含義		
31		<_AMODE>	INT	替代模式		
				單位：	選擇進給	
					0 =	平行於輪廓的 DX 及 DZ 方向材料移除進給
					1 =	D 進給
				十位數：	進給策略	
					0 =	可變切削深度 (90 ... 100%)
					1 =	恆定切削深度
				百位數：	切削分段	
					0 =	統一
					1 =	對齊稜邊
				千位數：	選擇輪廓允差 U1，雙精加工	
					0 =	否
					1 =	是
				萬位數：	更新毛胚選擇	
					0 =	否
					1 =	是
				十萬位數：	選擇毛胚 XD 上的允差	
					0 =	絕對，直徑中的橫向軸數值
					1 =	增量，半徑中的橫向軸數值
				百萬位數：	選擇毛胚 ZD 上的允差	
					0 =	絕對值
					1 =	增量
				千萬位數：	選擇臨界值 2 XB	
					0 =	絕對，直徑中的橫向軸數值
					1 =	增量，半徑中的橫向軸數值
				億位數：	選擇臨界值 2 ZB	
					0 =	絕對值
					1 =	增量
				十億位數：		
					0 =	主導通道

號碼	參數遮罩	內部參數	資料類型	含義		
					1 =	跟隨通道
32		<_PK>	INT	若機台可用的通道超過兩個，可用於計算搭配通道的數量。		
33	DCH	<_DCH>	REAL	通道偏移		
34	FS	<_FS>	REAL	完整加工期間的精加工進給率		

20.1.44 CYCLE4071 - 在反轉點進給的縱向研磨

語法

```
CYCLE4071(<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

參數

號碼	參數	資料類型	含義
1	<S_A>	REAL	開始的進給深度
2	<S_B>	REAL	結尾的進給深度
3	<S_W>	REAL	研磨寬度
4	<S_U>	REAL	滑磨時間
5	<S_I>	REAL	進給的進給率
6	<S_K>	REAL	橫向進給的進給率
7	<S_H>	INT	重複次數
8	<S_A1>	AXIS	進給軸（選配）或第一幾何軸
9	<S_A2>	AXIS	震盪軸（選配）或第二幾何軸

功能

循環用於執行 <S_H> 個進給。開始及結尾的進給深度可以不同。進給之間有切線動作。

順序

1. 從震盪軸目前的位置開始循環。
2. 進給軸以進給 P5 <S_I> 的進給率移動到開始 P1 **<**S_A**>** 的進給深度。
3. 以滑磨時間 P4 **<**S_U**>** 滑磨。

4. 以研磨寬度 P3 <S_W> 為移動路徑的震盪軸以橫向進給 P6 <S_K> 的進給率移動。
5. 進給軸以進給 P5 <S_I> 的進給率移動到結尾 P2 <S_B> 的進給深度。
6. 以滑磨時間 P4 <S_U> 滑磨。
7. 以研磨寬度 P4 <S_A> 為移動路徑的震盪軸以橫向進給 P6 <S_K> 的進給率移動至開始點。

此順序無法以單一單節中斷。

根據設定的重複次數 P7 (<S_H>) 重複此順序。

範例

以下列循環參數執行兩個震盪動作：

- 開始的進給深度：0.02 毫米
- 結尾的進給深度：0.01 毫米
- 衝程：100 毫米
- 滑磨時間：1 秒
- 進給率：1 毫米 / 分鐘
- 移動進給率：1000 毫米 / 分鐘
- 重複：2
- 震盪與進給軸：標準幾何軸

程式碼

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.45 CYCLE4072 - 於反轉點進給的縱向研磨和取消訊號

語法

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

參數

號碼	參數	資料類型	含義
1	<S_GAUGE>	STRING	進給的取消條件： 1. 快速輸入的編號 2. 邏輯算式
2	<S_A>	REAL	開始的進給深度
3	<S_B>	REAL	結尾的進給深度
4	<S_W>	REAL	研磨寬度
5	<S_U>	REAL	滑磨時間
6	<S_I>	REAL	進給的進給率
7	<S_K>	REAL	橫向進給的進給率
8	<S_H>	INT	重複次數
9	<S_A1>	AXIS	進給軸（選配）或第一幾何軸
10	<S_A2>	AXIS	震盪軸（選配）或第二幾何軸

功能

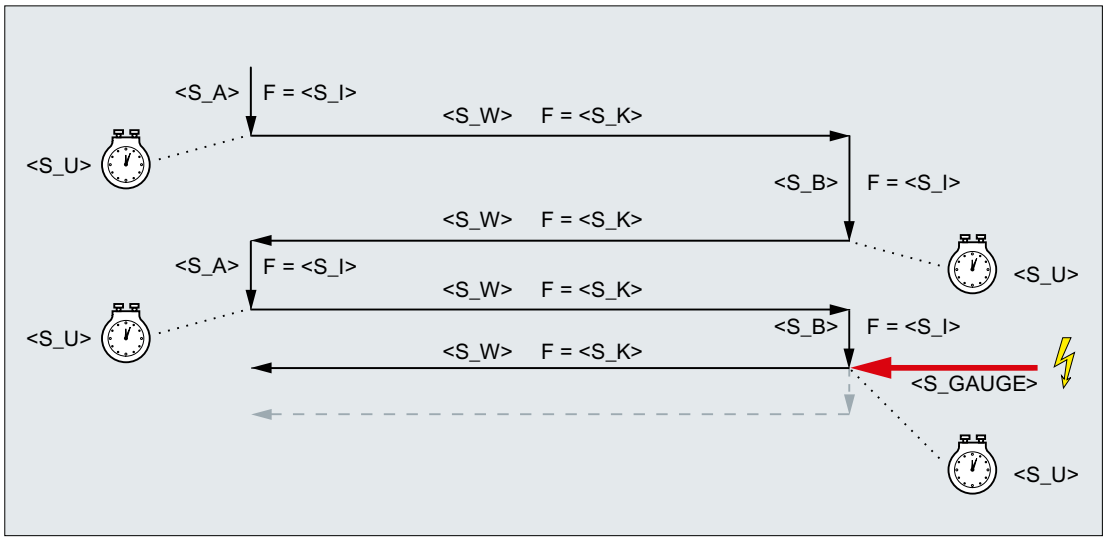
循環用於執行 <S_H> 個進給，將外部取消訊號納入考量。開始及結尾的進給深度可以不同。進給之間有切線動作。滿足取消條件時將取消深度進給。取消深度進給後一定會執行完整衝程。

順序

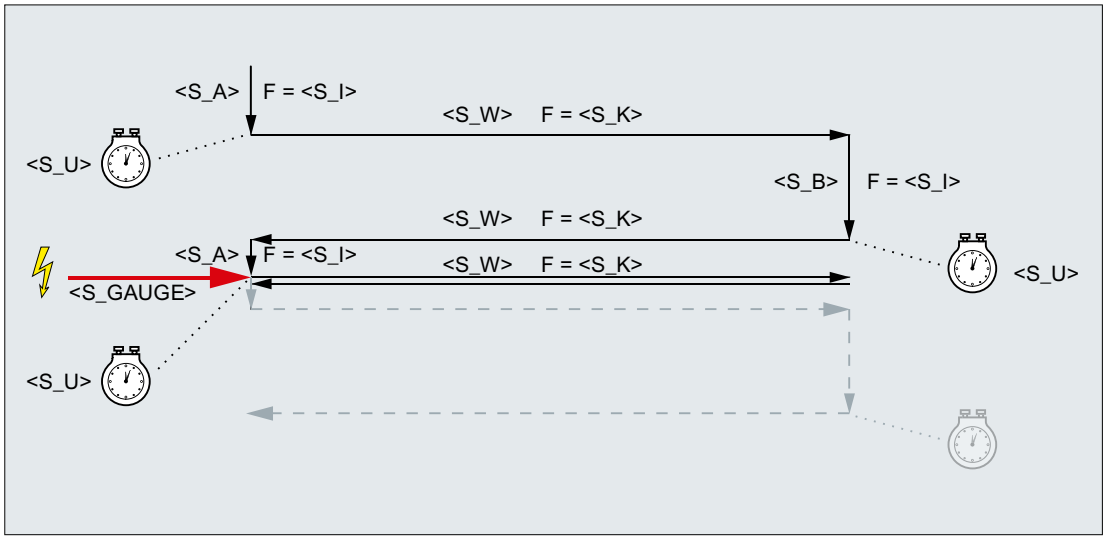
1. 從震盪軸目前的位置開始循環。
2. 進給軸以進給 P6 <S_I> 的進給率移動到開始 P2 <S_A> 的進給深度。
3. 以滑磨時間 P5 <S_U> 滑磨。
4. 以研磨寬度 P4 <S_W> 為移動路徑的震盪軸以橫向進給 P7 <S_K> 的進給率移動。
5. 進給軸以進給 P6 <S_I> 的進給率移動到結尾 P3 <S_B> 的進給深度。
6. 以滑磨時間 P5 <S_U> 滑磨。

7. 以研磨寬度 P4 <S_W> 為移動路徑的震盪軸以橫向進給 P7 <S_K> 的進給率移動至開始點。
8. 不含取消：重複上述的順序，直到達到設定的重複次數 P7 (<S_H>)。
   含取消：機台會在達到下一個開始點時停止。

此順序無法以單一單節中斷。



圖像 20-1　取消結尾的進給



圖像 20-2　取消開始的進給

資源

作為資源時，循環會使用全單節同步動作及同步動作變數。同步動作是由同步動作範圍的任意區域以動態方式決定（CUS.DIR - 1 ...、CMA.DIR - 1000 ...、CST.DIR – 1199 ...）。SYG_IS[1] 作為同步動作變數。

範例

範例 1：有兩個衝程的震盪：

循環參數

- 開始的進給深度：0.02 毫米
- 結尾的進給深度：0.01 毫米
- 衝程：100 毫米
- 滑磨時間：1 秒
- 進給率：1 毫米 / 分鐘
- 移動進給率：1000 毫米 / 分鐘
- 重複：2
- 震盪與進給軸：標準幾何軸

取消訊號快速輸入 1 ($A_IN[1])

程式碼

```
N10 T1 D1
N20 CYCLE4072("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

範例 2：有兩個衝程的震盪：

循環參數

- 開始的進給深度：0.02 毫米
- 結尾的進給深度：0.01 毫米
- 衝程：100 毫米
- 滑磨時間：1 秒
- 進給率：1 毫米 / 分鐘
- 移動進給率：1000 毫米 / 分鐘
- 重複：2
- 震盪與進給軸：標準幾何軸

取消訊號變數 $A_DBR[20] < 0.01

程式碼

```
N10 T1 D1
```

程式碼

```
N20 CYCLE4072("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

20.1.46 CYCLE4073 - 連續進給之縱向研磨

語法

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

參數

號碼	參數	資料類型	含義
1	<S_A>	REAL	開始的進給深度
2	<S_B>	REAL	結尾的進給深度
3	<S_W>	REAL	研磨寬度
4	<S_U>	REAL	滑磨時間
5	<S_K>	REAL	橫向進給的進給率
6	<S_H>	INT	重複次數
7	<S_A1>	AXIS	進給軸（選配）或第一幾何軸
8	<S_A2>	AXIS	震盪軸（選配）或第二幾何軸

功能

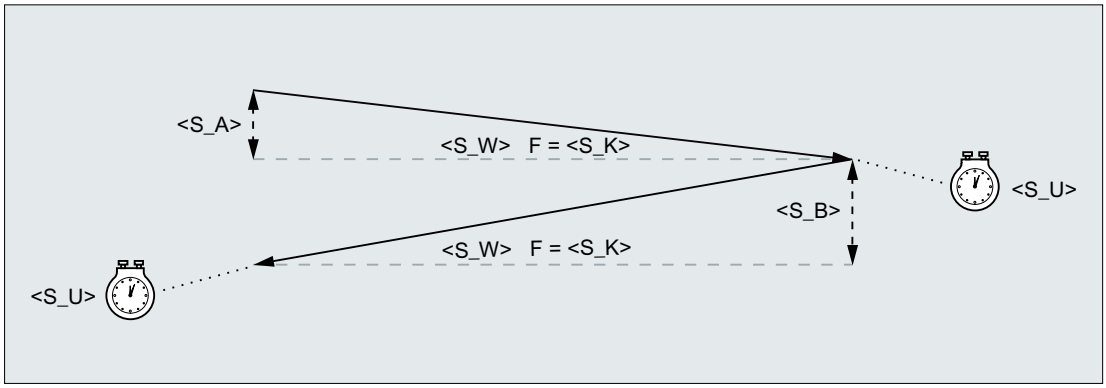
循環用於執行 <S_H> 個進給。由開始至結尾以及由結尾至開始的進給可以不同。

順序

1. 從進給深度 0 的震盪軸目前位置開始循環
2. 以研磨寬度 P3 <S_W> 為移動路徑的震盪軸以移動進給 P5 <S_K> 的進給率移動，並持續增加進給深度，最高到開始 P1 <S_A> 的進給深度。
3. 以滑磨時間 P4 <S_U> 滑磨。
4. 以研磨寬度 P3 <S_W> 為移動路徑的震盪軸以移動進給 P5 <S_K> 的進給率移動到開始點，並持續增加進給深度，最高到結尾 P2 <S_B> 的進給深度。
5. 以滑磨時間 P4 <S_U> 滑磨。

此順序無法以單一單節中斷。

根據設定的重複次數 P7 (<S_H>) 重複此順序。



範例

有兩個衝程的震盪：

循環參數

- 開始的進給深度：0.02 毫米
- 結尾的進給深度：0.01 毫米
- 衝程：100 毫米
- 滑磨時間：1 秒
- 移動進給率：1000 毫米 / 分鐘
- 重複：2
- 震盪與進給軸：標準幾何軸

程式碼

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.47 CYCLE4074 - 連續進給的縱向研磨和取消訊號

語法

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

參數

號碼	參數	資料類型	含義
1	<S_GAUGE>	STRING	進給的取消條件： 1. 快速輸入的編號 2. 邏輯算式
2	<S_A>	REAL	開始的進給深度
3	<S_B>	REAL	結尾的進給深度
4	<S_W>	REAL	研磨寬度
5	<S_U>	REAL	滑磨時間
6	<S_K>	REAL	橫向進給的進給率
7	<S_H>	INT	重複次數
8	<S_A1>	AXIS	進給軸（選配）或第一幾何軸
9	<S_A2>	AXIS	震盪軸（選配）或第二幾何軸

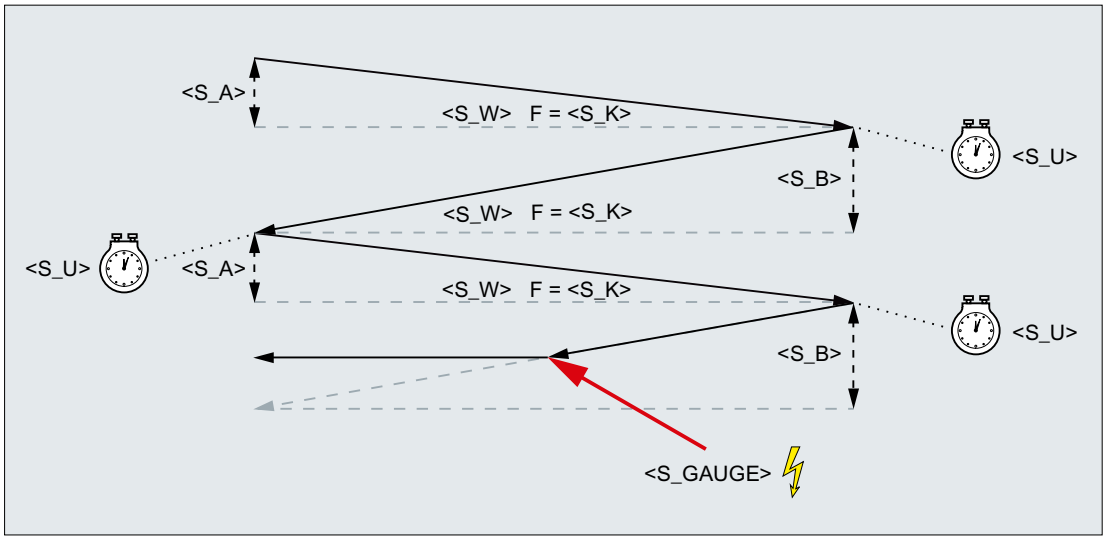
功能

循環用於執行 <S_H> 個進給，例如將外部取消訊號納入考量。開始及結尾的進給深度可以不同。滿足取消條件時將取消深度進給。取消深度進給後一定會執行完整衝程。

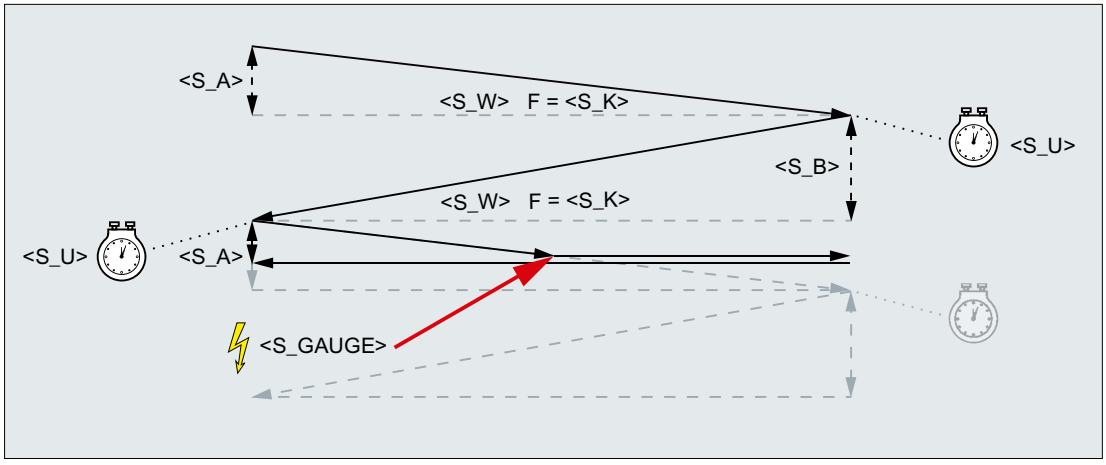
順序

1. 從進給深度 0 的震盪軸目前位置開始循環
2. 以研磨寬度 P4 <S_W> 為移動路徑的震盪軸以移動進給 P6 <S_K> 的進給率移動，並持續增加進給深度，最高到開始 P2 <S_A> 的進給深度。
3. 以滑磨時間 P5 <S_U> 滑磨。
4. 以研磨寬度 P4 <S_W> 為移動路徑的震盪軸以移動進給 P6 <S_K> 的進給率移動到開始點，並持續增加進給深度，最高到結尾 P3 <S_B> 的進給深度。
5. 以滑磨時間 P5 <S_U> 滑磨。
6. 不含取消：重複上述的順序，直到達到設定的重複次數 P7 (<S_H>)。
   含取消：取消深度進給。機台會在達到下一個開始點時停止。

此順序無法以單一單節中斷。



圖像 20-3　取消由結尾至開始的進給



圖像 20-4　取消由開始至結尾的進給 nd

資源

作為資源時，循環會使用全單節同步動作及同步動作變數。同步動作是由同步動作範圍的任意區域以動態方式決定（CUS.DIR - 1 ...、CMA.DIR - 1000 ...、CST.DIR – 1199 ...）。SYG_IS[1] 作為同步動作變數。

範例

範例 1：有兩個衝程的震盪：

循環參數

- 開始的進給深度：0.02 毫米
- 結尾的進給深度：0.01 毫米
- 衝程：100 毫米
- 滑磨時間：1 秒
- 移動進給率：1000 毫米 / 分鐘
- 重複：2
- 震盪與進給軸：標準幾何軸

取消訊號快速輸入 1 ($A_IN[1])

程式碼

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

範例 2：有兩個衝程的震盪：

循環參數

- 開始的進給深度：0.02 毫米
- 結尾的進給深度：0.01 毫米
- 衝程：100 毫米
- 滑磨時間：1 秒
- 移動進給率：1000 毫米 / 分鐘
- 重複：2
- 震盪與進給軸：標準幾何軸

取消訊號變數 $A_DBR[20] < 0.01

程式碼

```
N10 T1 D1
N20 CYCLE4074("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

20.1.48 CYCLE4075 - 在反轉點進給的表面研磨

語法

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>,
<S_A1>, <S_A2>)
```

參數

號碼	參數	資料類型	含義
1	<S_I>	REAL	開始的進給深度
2	<S_J>	REAL	結尾的進給深度
3	<S_K>	REAL	總進給深度
4	<S_A>	REAL	研磨寬度
5	<S_R>	REAL	進給的進給率
6	<S_F>	REAL	橫向進給的進給率
7	<S_P>	REAL	滑磨時間
8	<S_A1>	AXIS	進給軸（選配）
9	<S_A2>	AXIS	震盪軸（選配）

功能

循環用於進給步驟為總進給深度 P3 <S_K> 的加工。開始 P1 <S_I> 及結尾 P2 <S_J> 的進給深度可以不同。進給之間有切線動作。

位置資料 P1 至 P4 可以為負值或正值。

進給軸 P8 <S_A1> 和 / 或震盪軸 P9 <S_A2> 的規格為選擇性。如果未指定一個或兩個參數，循環就會使用通道的前兩個幾何軸。

如果開始 P1 <S_I> 及結尾 P2 <S_J> 的進給深度總和為 0 或總進給深度 P3 <S_K> 為 0，則只會執行滑磨衝程。

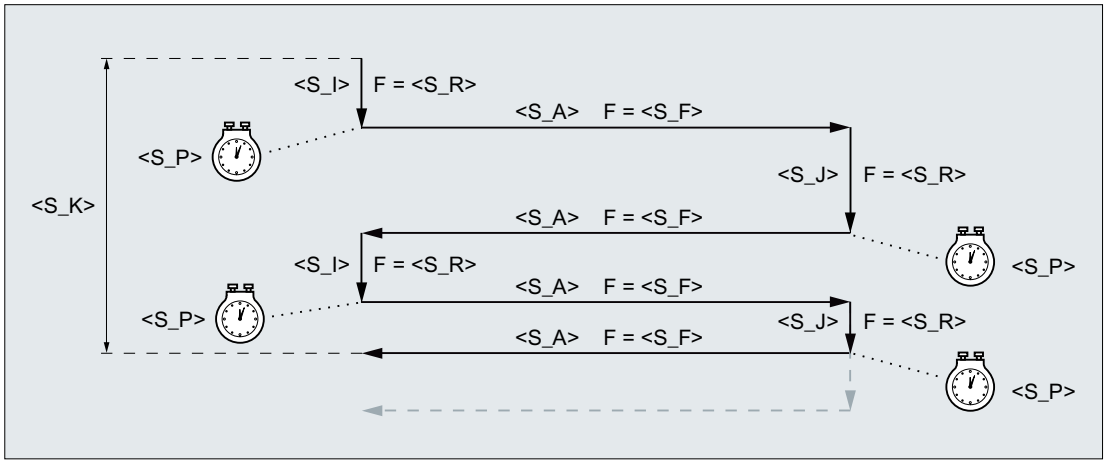
順序

1. 從震盪軸目前的位置開始循環。
2. 進給軸以進給 P5 <S_R> 的進給率移動到開始 P1 <S_I> 的進給深度。
3. 以滑磨時間 P7 <S_P> 滑磨。
4. 以研磨寬度 P4 <S_A> 為移動路徑的震盪軸以橫向進給 P6 <S_F> 的進給率移動。

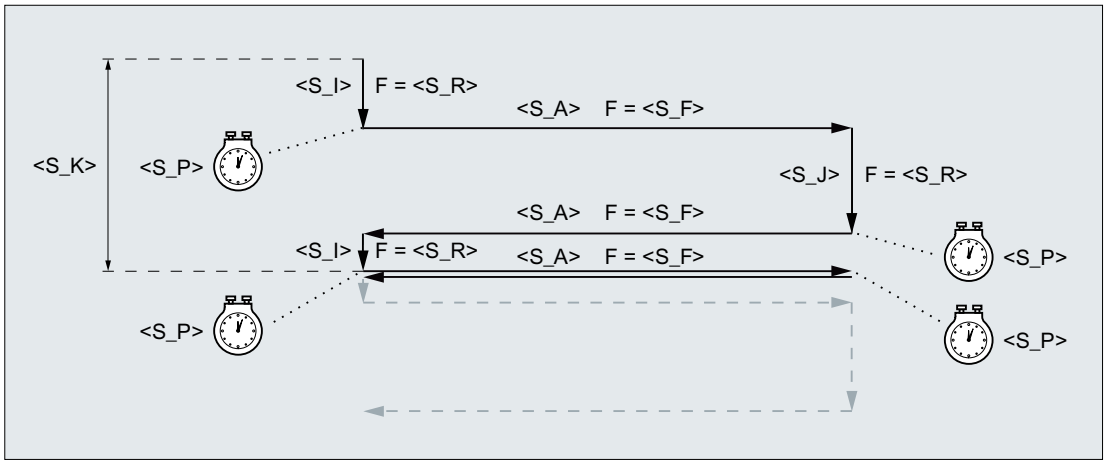
5. 進給軸以進給 P5 <S_R> 的進給率移動到結尾 P2 <S_J> 的進給深度。
6. 以滑磨時間 P7 <S_P> 滑磨。
7. 以研磨寬度 P4 <S_A> 為移動路徑的震盪軸以橫向進給 P6 <S_F> 的進給率移動至開始點。

此順序無法以單一單節中斷。

將重複順序，直到達到總進給深度 P3 <S_K>。最後衝程則會以不平均的方式分佈。



圖像 20-5　以第二反轉點的進給達到總進給深度



圖像 20-6　以第一反轉點的進給達到總進給深度

範例

以下列項目進行震盪：

- 開始處的 0.02 毫米進給深度
- 結尾處的 0.01 毫米進給深度
- 總進給深度 1 毫米

- 100 毫米衝程
- 進給進給率 1 毫米 / 分
- 橫向進給率 1000 毫米 / 分
- 1 秒鐘火花消除時間
- 標準幾何軸

程式碼

```
N10 T1 D1
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.49 CYCLE4077 - 於反轉點進給的表面研磨和取消訊號

語法

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

參數

號碼	參數	資料類型	含義
1	<S_GAUGE>	STRING	進給的取消條件： • 快速輸入的編號 • 邏輯算式
2	<S_I>	REAL	開始的進給深度
3	<S_J>	REAL	結尾的進給深度
4	<S_K>	REAL	總進給深度
5	<S_A>	REAL	研磨寬度
6	<S_R>	REAL	進給的進給率
7	<S_F>	REAL	橫向進給的進給率
8	<S_P>	REAL	滑磨時間
9	<S_A1>	AXIS	進給軸（選配）
10	<S_A2>	AXIS	震盪軸（選配）

功能

循環用於進給步驟為總進給深度 P4 <S_K> 的加工。開始 P2 <S_I> 及結尾 P3 <S_J> 的進給深度可以不同。進給之間有切線動作。快速輸入的取消訊號為 1 或滿足取消條件時將取消深度進給。取消後會執行完整衝程。

位置資料 P2 至 P5 可以為負值或正值。

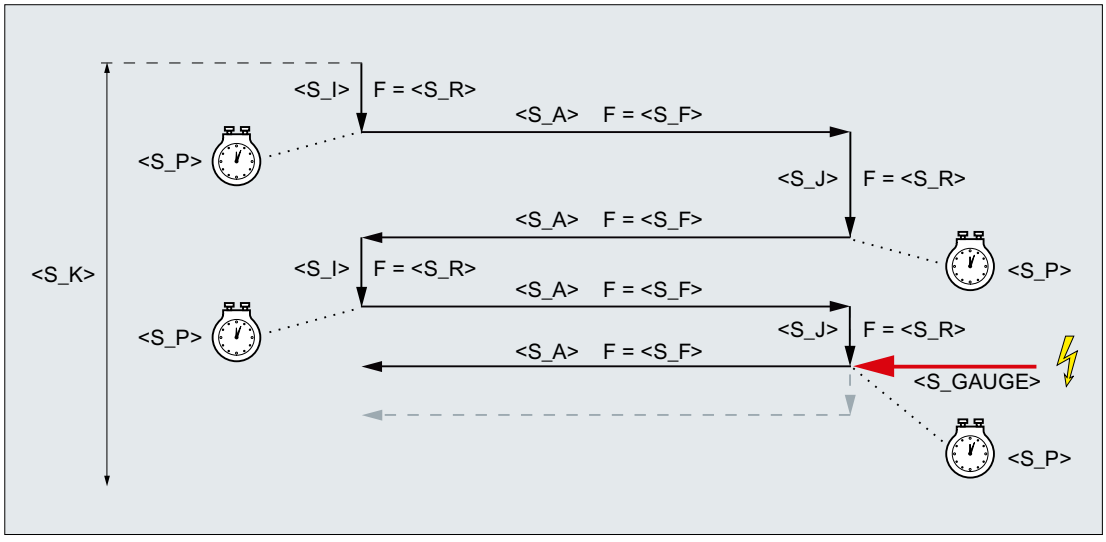
進給軸 P9 <S_A1> 和 / 或震盪軸 P10 <S_A2> 的規格為選擇性。如果未指定一個或兩個參數，循環就會使用通道的前兩個幾何軸。

如果開始 P2 <S_I> 及結尾 P3 <S_J> 的進給深度總和為 0 或總進給深度 P4 <S_K> 為 0，則只會執行滑磨衝程。

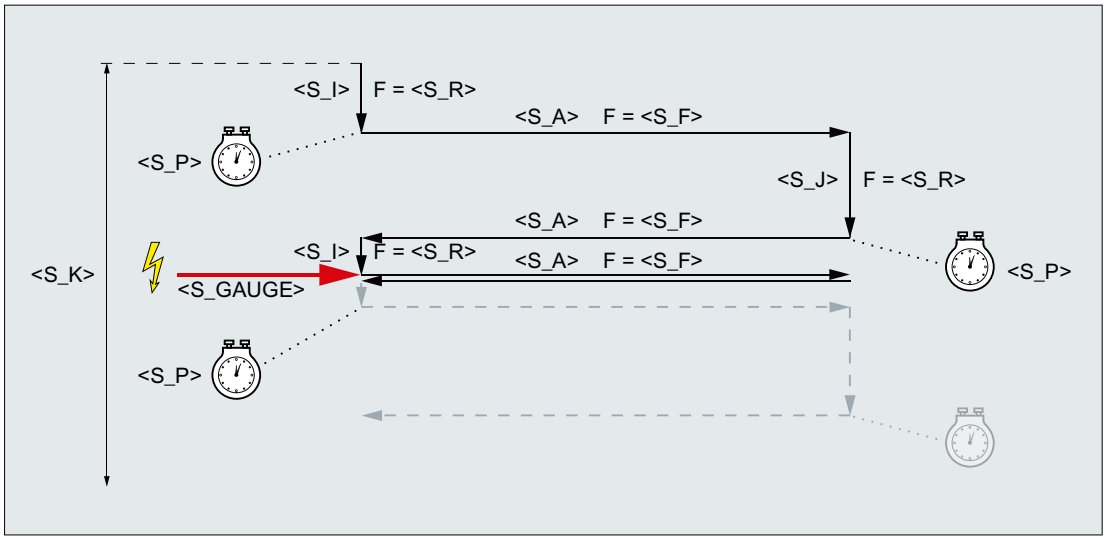
順序

1. 從震盪軸目前的位置開始循環。
2. 進給軸以進給 P6 <S_R> 的進給率移動到開始 P2 <S_I> 的進給深度。
3. 以滑磨時間 P8 <S_P> 滑磨。
4. 以研磨寬度 P5 <S_A> 為移動路徑的震盪軸以橫向進給 P7 <S_F> 的進給率移動。
5. 進給軸以進給 P6 <S_R> 的進給率移動到結尾 P3 <S_J> 的進給深度。
6. 以滑磨時間 P8 <S_P> 滑磨。
7. 以研磨寬度 P5 <S_A> 為移動路徑的震盪軸以橫向進給 P7 <S_F> 的進給率移動至開始點。
8. 不含取消：將重複上述順序，直到達到總進給深度 P4 <S_K>。最後衝程則會以不平均的方式分佈。
   含取消：機台會在開始點停止。

此順序無法以單一單節中斷。



圖像 20-7　取消結尾的進給



圖像 20-8　取消開始的進給

資源

作為資源時，循環會使用全單節同步動作及同步動作變數。同步動作是由同步動作範圍的任意區域以動態方式決定（CUS.DIR - 1 ...、CMA.DIR - 1000 ...、CST.DIR – 1199 ...）。SYG_IS[1] 作為同步動作變數。

範例

範例 1

以下列項目進行震盪：

- 開始處的 0.02 毫米進給深度
- 結尾處的 0.01 毫米進給深度
- 總進給深度 1 毫米
- 100 毫米衝程
- 進給進給率 1 毫米 / 分
- 橫向進給率 1000 毫米 / 分
- 1 秒鐘火花消除時間
- 標準幾何軸

取消訊號快速輸入 1 ($A_IN[1])

程式碼

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

範例 2

以下列項目進行震盪：

- 開始處的 0.02 毫米進給深度
- 結尾處的 0.01 毫米進給深度
- 總進給深度 1 毫米
- 100 毫米衝程
- 進給進給率 1 毫米 / 分
- 橫向進給率 1000 毫米 / 分
- 1 秒鐘火花消除時間
- 標準幾何軸

取消訊號雙通訊埠 RAM 變數 20 小於 0.01 ($A_DBR[20] < 0.01)

程式碼

```
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 - 連續進給之表面研磨

語法

```
CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

參數

編號	參數	資料類型	含義
1	<S_I>	REAL	由開始至結尾的進給深度
2	<S_J>	REAL	由結尾至開始的進給深度

編號	參數	資料類型	含義
3	<S_K>	REAL	總進給深度
4	<S_A>	REAL	研磨寬度
5	<S_F>	REAL	進給率
6	<S_P>	REAL	滑磨時間
7	<S_A1>	AXIS	進給軸（選配）
8	<S_A2>	AXIS	震盪軸（選配）

功能

循環用於以連續進給方式加工總進給深度 P3 <S_K>。由開始至結尾 P1 <S_I> 以及由結尾至開始 P2 <S_J> 的進給深度可以不同。

位置資料 P1 至 P4 可以為負值或正值。

進給軸 P8 <S_A1> 和 / 或震盪軸 P9 <S_A2> 的規格為選擇性。如果未指定一個或兩個參數，循環就會使用通道的前兩個幾何軸。

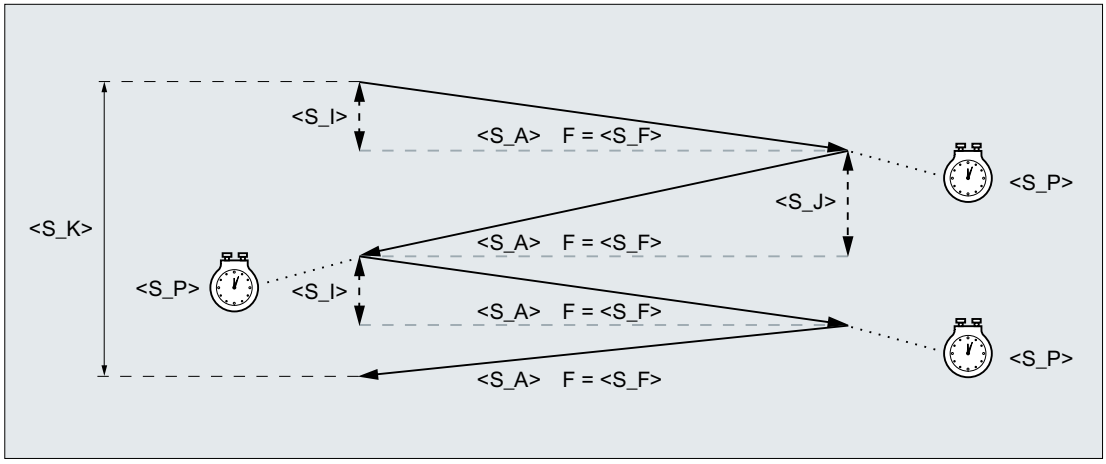
如果進給深度 P1 <S_I> 及 P2 <S_J> 的總和為 0 或總進給深度 P3 <S_K> 為 0，則只會執行滑磨衝程。

順序

1. 從進給深度 0 的震盪軸目前位置開始循環
2. 以研磨寬度 P4 <S_A> 為移動路徑的震盪軸以進給率 P5 <S_F> 移動，並持續增加進給深度，最高到開始 P1 <S_I> 的進給深度。
3. 以滑磨時間 P7 <S_P> 滑磨。
4. 以研磨寬度 P4 <S_A> 為移動路徑的震盪軸以進給率 P5 <S_F> 移動到開始點，並持續增加進給深度，最高到結尾 P2 <S_J> 的進給深度。
5. 以滑磨時間 P7 <S_P> 滑磨。
6. 以研磨寬度 P4 <S_A> 為移動路徑的震盪軸以進給率 P5 <S_F> 移動至開始點。

此順序無法以單一單節中斷。

將重複順序，直到達到總進給深度 P3 <S_K>。最後衝程則會以不平均的方式分佈。



範例

以下列項目進行震盪：

- 開始處的 20 毫米進給深度
- 結尾的 10 毫米進給深度
- 總進給深度 100 毫米
- 100 毫米衝程
- 進給率 1000 毫米 / 分
- 1 秒鐘滑磨時間
- 標準幾何軸

程式碼

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

20.1.51 CYCLE4079 - 間歇進給之表面研磨

語法

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>,
<S_A1>, <S_A2>)
```

參數

號碼	參數	資料類型	含義
1	<S_I>	REAL	開始的進給深度
2	<S_J>	REAL	結尾的進給深度
3	<S_K>	REAL	總進給深度
4	<S_A>	REAL	研磨寬度
5	<S_R>	REAL	進給的進給率
6	<S_F>	REAL	橫向進給的進給率
7	<S_P>	REAL	滑磨時間
8	<S_A1>	AXIS	進給軸（選配）
9	<S_A2>	AXIS	震盪軸（選配）

功能

循環用於進給步驟為總進給深度 P3 <S_K> 的加工。開始 P1 <S_I> 及結尾 P2 <S_J> 的進給深度可以不同。進給之間有切線動作。

位置資料 P1 至 P4 可以為負值或正值。

進給軸 P8 <S_A1> 和 / 或震盪軸 P9 <S_A2> 的規格為選擇性。如果未指定一個或兩個參數，循環就會使用通道的前兩個幾何軸。

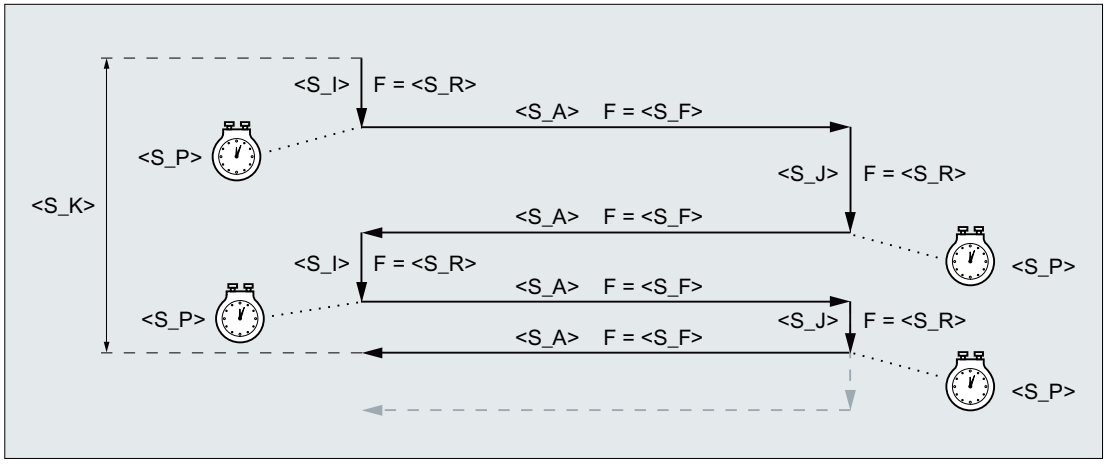
如果開始 P1 <S_I> 及結尾 P2 <S_J> 的進給深度總和為 0 或總進給深度 P3 <S_K> 為 0，則只會執行滑磨衝程。

順序

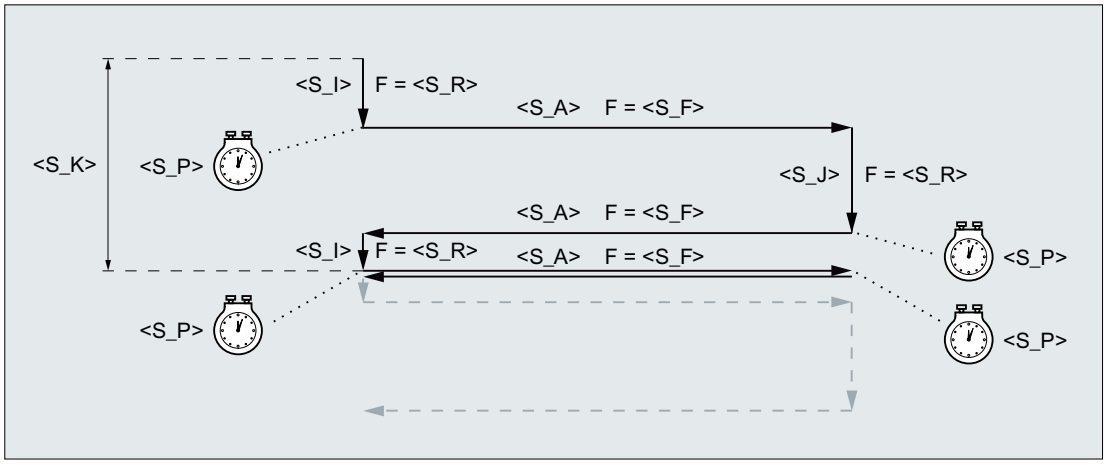
1. 從震盪軸目前的位置開始循環。
2. 進給軸以進給 P5 <S_R> 的進給率移動到開始 P1 <S_I> 的進給深度。
3. 以滑磨時間 P7 <S_P> 滑磨。
4. 以研磨寬度 P4 <S_A> 為移動路徑的震盪軸以橫向進給 P6 <S_F> 的進給率移動。
5. 進給軸以進給 P5 <S_R> 的進給率移動到結尾 P2 <S_J> 的進給深度。
6. 以滑磨時間 P7 <S_P> 滑磨。
7. 以研磨寬度 P4 <S_A> 為移動路徑的震盪軸以橫向進給 P6 <S_F> 的進給率移動至開始點。

此順序無法以單一單節中斷。

將重複順序，直到達到總進給深度 P3 <S_K>。最後衝程則會以不平均的方式分佈。



圖像 20-9　以第二反轉點的進給達到總進給深度



圖像 20-10　以第一反轉點的進給達到總進給深度

範例

以下列項目進行震盪：

- 開始處的 0.02 毫米進給深度
- 結尾處的 0.01 毫米進給深度
- 總進給深度 1 毫米
- 100 毫米衝程
- 進給進給率 1 毫米 / 分
- 橫向進給率 1000 毫米 / 分

- 1 秒鐘火花消除時間
- 標準幾何軸

程式碼

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

20.1.52 GROUP_BEGIN - 程式單節開始處

語法

```
GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)
```

參數

編號	參數遮罩	內部參數	資料類型	含義		
1		<_LEVEL>	INT	層級		
				0 =	主層級	
				1 =	第一次層級	
2		<_NAME>	STRING[128]	單節名稱		
3		<_SP>	INT	主軸		
				0 =	無主軸	
				1 =	主要主軸	
				2 =	副主軸	
4		<_MODE>	INT	模式		
				位元 0	= 1	GROUP_ADDEND 存在
				位元 1	= 1	ShopTurn：自動回退（移動至換刀點）
				位元 12	保留	
				位元 13	保留	
5		<S_ICON>	STRING[32]	圖示名稱（僅適用於操作介面）		

20.1.53 GROUP_END - 程式單節結束

語法

```
GROUP_END(<_LEVEL>, <_SP>)
```

參數

編號	參數遮罩	內部參數	資料類型	含義	
1		<_LEVEL>	INT	層級	
				0 =	主層級
				1 =	第一次層級
2		<_SP>	INT	主軸	
				0 =	無主軸
				1 =	主要主軸
				2 =	副主軸

20.1.54 GROUP_ADDEND - 嘗試切削添加

語法

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

參數

編號	參數遮罩	內部參數	資料類型	含義	
1		<_LEVEL>	INT	層級	
				0 =	主層級
				1 =	第一次層級
2		<_SP>	INT	主軸	
				0 =	無主軸
				1 =	主要主軸
				2 =	副主軸

20.1.55 補充條件

20.1.55.1 循環畫面表單的技術刻度比例

技術比例生效時，就可針對各種循環畫面表單選擇簡化輸入，其中只會顯示最重要的循環參數

例如，下列畫面可選取表單簡化輸入：

技術	循環畫面表單
鑽孔	深孔鑽
	攻牙
銑削	矩形腔
	輪廓銑削：口袋型
車削	螺紋車削：縱向
	輪廓車削：材料移除
	輪廓車削：切槽
	輪廓車削：溝槽車削

在相關循環畫面表單的使用者介面中，提供「輸入：**簡單**」及「輸入：**完整**」選項。

不會顯示的循環參數

簡化輸入不顯示的循環參數，為預先指派固定、具有技術用途但非變數值的參數。或是透過通道專屬循環設定資料指派參數化值的循環參數。請參閱「調試」>「通道專屬循環設定資料」之下的段落。

切換：由「輸入：完整」>「輸入：簡單」

若畫面表單以「輸入完整」設定填寫，然後切換為「輸入簡單」，則在產生循環呼叫時，就不會再顯示參數使用的預設或設定資料值。

調試

通道專屬組態機床資料

循環畫面表單的技術比例可透過機械參數啟動：

MD52210 $MCS_FUNCTION_MASK_DISP，位元 9 = 1（選擇顯示「輸入簡單」）

通道專屬循環設定參數

若循環畫面表單的簡化輸入生效，特定循環參數值就可利用下列設定資料指定：

號碼	識別碼	含義
SD55300	$SCS_EASY_SAFETY_CLEARANCE	安全間隙
SD55301	$SCS_EASY_DWELL_TIME	停頓時間
SD55305	$SCS_EASY_DRILL_DEEP_FD1	深孔鑽：百分比：第一進給率
SD55306	$SCS_EASY_DRILL_DEEP_DF	深孔鑽：百分比：進給
SD55307	$SCS_EASY_DRILL_DEEP_V1	深孔鑽：最小深度進給
SD55308	$SCS_EASY_DRILL_DEEP_V2	深孔鑽：回退距離
SD55309	$SCS_EASY_THREAD_RETURN_DIST	螺紋車削：傳回距離

20.2 測量循環

測量循環是西門子提供的特殊子程式，用於解決特定測量作業。 測量循環和一般循環一樣，也可透過參數設定調整因應特定問題。

以下區域及技術提供測量循環：

- 刀具測量、車削 / 銑削
- 工件測量、車削 / 銑削

參考資料

有關測量循環之詳細說明，請參閱：

測量循環的程式設計手冊

表格 21

21.1 操作

說明

循環

發生在 NC 程式（G 碼）上含有全部循環的清單，亦即可以使用遮罩程式設計在程式編輯器中，或是必須在無程式設計支援之下針對迴圈進行程式設計。考量到相容性的緣故，控制仍然必須提供循環，不過並不需要使用 SINUMERIK Operate 程式編輯器進行編輯（「相容性循環」）將其納入考量。

操作 A ... C

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
:	O	NC 主單節編號，跳躍標籤中止，串鍊運算子		+		PGAsl
*	O	乘法之運算子		+		PGAsl
+	O	加法之運算子		+		PGAsl
-	O	減法之運算子		+		PGAsl
<	O	比較運算子，少於		+		PGAsl
<<	O	字串的串鍊運算子		+		PGAsl
<=	O	比較運算子，少於或等於		+		PGAsl
=	O	指派運算子		+		PGAsl
>=	O	比較運算子，大於或等於		+		PGAsl
/	O	除法之運算子		+		PGAsl
/0 … … /7		單節已略過（第 1 略過層級） ... ... 單節已略過（第 8 略過層級）		+		PGsl
A	A	軸名稱	m/s	+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
A2	A	刀具方向：RPY 或尤拉角	s	+		PGAsl
A3	A	刀具方向：方向向量的第一分量	s	+		PGAsl
A4	A	刀具方向：單節開始表面法線向量的第一分量	s	+		PGAsl
A5	A	刀具方向：單節結束表面法線向量的第一分量	s	+		PGAsl
A6	A	刀具方向：旋轉錐形軸方向向量的第一分量	s	+		PGAsl
A7	A	刀具方向：錐形柱面中間方向的第一向量分量	s	+		PGAsl
ABS	F	絕對值（量）		+	+	PGAsl
AC	K	座標 / 位置的絕對尺寸	s	+		PGsl
ACC	K	目前的軸加速度之效果	m	+	+	PGsl
ACCLIMA	K	目前的最大軸加速度之效果	m	+	+	PGAsl
ACN	K	旋轉軸之絕對尺寸，負向上的逼近位置	s	+		PGsl
ACOS	F	反餘弦 （三角函數）		+	+	PGAsl
ACP	K	旋轉軸之絕對尺寸，正向上的逼近位置	s	+		PGsl
ACTBLOCNO	P	輸出目前警報單節的單節號碼，即使「目前單節顯示抑制」(DISPLOF) 生效亦然！		+		PGAsl
ADDFRAME	F	包含可能的測量框架之啟用		+	-	PGAsl、FB1sl (K2)
ADIS	A	路徑函數 G1、G2、G3、... 之倒圓角間隙	m	+		PGsl
ADISPOS	A	快速移動 G0 之倒圓角間隙	m	+		PGsl
ADISPOSA	P	IPOBRKA 允差視窗之大小	m	+	+	PGAsl
ALF	A	LIFTFAST 角度	m	+		PGAsl
AMIRROR	G	可程式設計的鏡像	s	+		PGsl
AND	K	邏輯 AND		+		PGAsl
ANG	A	輪廓角度	s	+		PGsl
AP	A	極角	m/s	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
APR	K	讀取 / 顯示存取保護		+		PGAsl
APRB	K	讀取存取權限，OPI		+		PGAsl
APRP	K	讀取存取權限，工件程式		+		PGAsl
APW	K	寫入存取保護		+		PGAsl
APWB	K	寫入存取權限，OPI		+		PGAsl
APWP	K	寫入存取權限，工件程式		+		PGAsl
APX	K	用來執行特定語言元件的存取權限之定義		+		PGAsl
AR	A	缺口角度	m/s	+		PGsl
AROT	G	可程式設計的旋轉	s	+		PGsl
AROTS	G	具立體角的可程式設計框架旋轉	s	+		PGsl
AS	K	巨集定義		+		PGAsl
ASCALE	G	可程式設計的比例	s	+		PGsl
ASIN	F	算術函數，反正弦		+	+	PGAsl
ASPLINE	G	Akima 曲線	m	+		PGAsl
ATAN2	F	反正切 2		+	+	PGAsl
ATOL	A	壓縮函數、方向平滑化與平滑化類型的軸專屬允差	m	+		PGAsl
ATRANS	G	附加的可程式設計工件偏移量	s	+		PGsl
AUXFUDEL	P	專門由全域清單刪除輔助功能通道		+	-	FB1sl (H2)
AUXFUDELG	P	專門由全域清單刪除輔助功能群組通道的所有輔助功能		+	-	FB1sl (H2)
AUXFUMSEQ	P	決定 M 輔助功能的輸出順序		+	-	FB1sl (H2)
AUXFUSYNC	P	由輔助功能的全域清單，產生通道專屬 SERUPRO 結尾 ASUB 的完整工件程式單節作為字串		+	-	FB1sl (H2)
AX	K	變數軸識別碼	m/s	+		PGAsl
AXCTSWE	P	旋轉軸容器		+	-	PGAsl
AXCTSWEC	P	取消軸容器旋轉的啟用		+	+	PGAsl
AXCTSWED	P	旋轉軸容器（調試用的指令不同！）		+	-	PGAsl
AXIS	K	軸識別碼，軸位址		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
AXNAME	F	將輸入字串轉換軸識別碼		+	-	PGAsl
AXSTRING	F	轉換字串主軸號碼		+	-	PGAsl
AXTOCHAN	P	請求特定通道之軸。可能來自 NC 程式及同步化動作。		+	+	PGAsl
AXTOSPI	F	將軸識別碼轉換為主軸索引		+	-	PGAsl
B	A	軸名稱	m/s	+		PGAsl
B2	A	刀具方向：RPY 或尤拉角	s	+		PGAsl
B3	A	刀具方向：方向向量的第二分量	s	+		PGAsl
B4	A	刀具方向：單節開始表面法線向量的第二分量	s	+		PGAsl
B5	A	刀具方向：單節結束表面法線向量的第二分量	s	+		PGAsl
B6	A	刀具方向：旋轉錐形軸方向向量的第二分量	s	+		PGAsl
B7	A	刀具方向：錐形柱面中間方向的第二向量分量	s	+		PGAsl
B_AND	O	逐一位元 AND		+		PGAsl
B_OR	O	逐一位元 OR		+		PGAsl
B_NOT	O	逐一位元否定		+		PGAsl
B_XOR	O	逐一位元互斥 OR		+		PGAsl
BAUTO	G	以接下來 3 個點定義最初的曲線區段	m	+		PGAsl
BLOCK	K	和關鍵字 TO 一起定義，在間接子程式呼叫中待處理的程式片段		+		PGAsl
BLSYNC	K	僅隨下一個單節變更開始處理中斷常式		+		PGAsl
BNAT [6]	G	自然變化至第一個曲線單節	m	+		PGAsl
BOOL	K	資料類型：布林值 TRUE/FALSE 或 1/0		+		PGAsl
BOUND	F	測試數值是否介於定義之值域間。若數值相同，將傳回測試值。		+	+	PGAsl
BRISK [6]	G	路徑非平滑化之快速加速	m	+		PGAsl
BRISKA	P	啟動已程式設計軸的路徑輕快加速		+	-	PGAsl
BSPLINE	G	B 曲線	m	+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
BTAN	G	沿切線變化至第一個曲線單節	m	+		PGAsl
C	A	軸名稱	m/s	+		PGAsl
C2	A	刀具方向：RPY 或尤拉角	s	+		PGAsl
C3	A	刀具方向：方向向量的第三分量	s	+		PGAsl
C4	A	刀具方向：單節開始表面法線向量的第三分量	s	+		PGAsl
C5	A	刀具方向：單節結束表面法線向量的第三分量	s	+		PGAsl
C6	A	刀具方向：旋轉錐形軸方向向量的第三分量	s	+		PGAsl
C7	A	刀具方向：錐形柱面中間方向的第三向量分量	s	+		PGAsl
CAC	K	絕對位置逼近		+		PGAsl
CACN	K	列於表中的值，以反方向絕對逼近		+		PGAsl
CACP	K	列於表中的值，以同方向絕對逼近		+		PGAsl
CALCDAT	F	從 3 或 4 個點計算半徑和圓的中心點		+	-	PGAsl
CALCPOSI	F	檢查保護區差異、工作區限制與軟體限制		+	-	PGAsl
CALL	K	間接子程式呼叫		+		PGAsl
CALLPATH	P	子程式呼叫之可程式設計的搜尋路徑		+	-	PGAsl
CANCEL	P	取消模態同步化動作		+	-	FBSYsl
CASE	K	條件式程式分支		+		PGAsl
CDC	K	直接逼近位置		+		PGAsl
CDOF [6]	G	關閉碰撞監控	m	+		PGsl
CDOF2	G	進行 3D 周長銑削時關閉碰撞監控	m	+		PGsl
CDON	G	啟動碰撞監控	m	+		PGsl
CFC [6]	G	輪廓上的恆定進給率	m	+		PGsl
CFIN	G	恆定進給率僅適用內部半徑，不適用外部半徑	m	+		PGsl
CFINE	F	指派微調偏移量給 FRAME 變數		+	-	PGAsl
CFTCP	G	刀具中心點之恆定進給率（中心點路徑）	m	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
CHAN	K	指定資料的有效範圍		+		PGAsl
CHANDATA	P	設定通道編號以存取通道資料		+	-	PGAsl
CHAR	K	資料類型：ASCII 字元		+		PGAsl
CHF	A	倒角； 值 = 倒角長度	s	+		PGsl
CHKDM	F	刀庫中的唯一性檢查		+	-	FBWsl
CHKDNO	F	檢查特有的 D 編號		+	-	PGAsl
CHR	A	倒角； 值 = 移動方向的倒角長度		+		PGsl
CIC	K	藉由增量的逼近位置		+		PGAsl
CIP	G	經由中間點的圓形插補	m	+		PGsl
CLEARM	P	為通道座標重置一個 / 數個標記		+	+	PGAsl
CLRINT	P	取消選取中斷		+	-	PGAsl
CMIRROR	F	座標軸上之鏡射		+	-	PGAsl
COARSEA	K	於達到「精確停止粗調」時動作結束	m	+		PGAsl
COLLPAIR	F	檢查是否碰撞對的一部份		+		PGAsl
COMPCAD	G	啟動壓縮程序函數 COMPCAD	m	+		PGAsl
COMPCURV	G	啟動壓縮程序函數 COMPCURV	m	+		PGAsl
COMPLETE		用來讀取和寫入資料的控制指令		+		PGAsl
COMPOF [6]	G	停用 NC 單節壓縮	m	+		PGAsl
COMPON	G	啟動壓縮程序函數 COMPON	m	+		PGAsl
COMPSURF	G	啟動壓縮程序函數 COMPSURF	m	+		PGAsl
CONTDCON	P	啟動表格化輪廓解碼		+	-	PGAsl
CONTPRON	P	啟動參考前置處理		+	-	PGAsl
CORROF	P	取消選擇所有啟用的重疊移動。		+	-	PGsl
CORRTRAFO	F	修改機台動力模型中方向軸的偏移向量或方向向量		+	-	PGAsl
COS	F	餘弦 （三角函數）		+	+	PGAsl
COUPDEF	P	定義 ELG 群組 / 同步主軸群組		+	-	PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
COUPDEL	P	刪除 ELG 群組		+	-	PGAsl
COUPOF	P	停用 ELG 群組 / 同步主軸對		+	-	PGAsl
COUPOFS	P	停用 ELG 群組 / 同步主軸對，搭配跟隨主軸之停止		+	-	PGAsl
COUPON	P	啟動 ELG 群組 / 同步主軸對		+	-	PGAsl
COUPONC	P	傳輸 ELG 群組 / 同步主軸對，含先前之程式設計		+	-	PGAsl
COUPRES	P	重置 ELG 群組		+	-	PGAsl
CP [6]	G	路徑動作	m	+		PGAsl
CPBC	K	一般耦合：單節變更條件		+	+	FB3sl (M3)
CPDEF	K	一般耦合：建立耦合模組		+	+	FB3sl (M3)
CPDEL	K	一般耦合：刪除耦合模組		+	+	FB3sl (M3)
CPFMOF	K	一般耦合：跟隨軸完全關閉時的行為		+	+	FB3sl (M3)
CPFMON	K	一般耦合：跟隨軸開啟時的行為		+	+	FB3sl (M3)
CPFMSON	K	一般耦合：同步模式		+	+	FB3sl (M3)
CPFPOS	K	一般耦合：跟隨軸的同步位置		+	+	FB3sl (M3)
CPFRS	K	一般耦合：座標參考系統		+	+	FB3sl (M3)
CPLA	K	一般耦合：先導軸定義		+	-	FB3sl (M3)
CPLCTID	K	一般耦合：曲線表的編號		+	+	FB3sl (M3)
CPLDEF	K	一般耦合：先導軸定義及建立耦合模組		+	+	FB3sl (M3)
CPLDEL	K	一般耦合：刪除耦合模組的先導軸		+	+	FB3sl (M3)
CPLDEN	K	一般耦合：耦合係數的分母		+	+	FB3sl (M3)
CPLINSC	K	一般耦合：先導軸輸入值的調整比例係數		+	+	FB3sl (M3)
CPLINTR	K	一般耦合：先導軸輸入值的偏移值		+	+	FB3sl (M3)
CPLNUM	K	一般耦合：耦合係數的分子		+	+	FB3sl (M3)
CPLOF	K	一般耦合：關閉耦合模組的先導軸		+	+	FB3sl (M3)
CPLON	K	一般耦合：開啟耦合模組的先導軸		+	+	FB3sl (M3)
CPLOUTSC	K	一般耦合：耦合輸出值的調整比例係數		+	+	FB3sl (M3)
CPLOUTTR	K	一般耦合：耦合輸出值的偏移值		+	+	FB3sl (M3)

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
CPLPOS	K	一般耦合：先導軸的同步位置		+	+	FB3sl (M3)
CPLSETVAL	K	一般耦合：耦合參考		+	+	FB3sl (M3)
CPMALARM	K	一般耦合：抑制特殊耦合相關警報輸出		+	+	FB3sl (M3)
CPMBRAKE	K	一般耦合：針對特定停止訊號及停止指令的跟隨軸回應		+	-	FB3sl (M3)
CPMPRT	K	一般耦合：透過程式測試進行單節搜尋執行時，工件程式啟動時的耦合回應		+	+	FB3sl (M3)
CPMRESET	K	一般耦合：RESET 的耦合行為		+	+	FB3sl (M3)
CPMSTART	K	一般耦合：工件程式啟動時的耦合行為		+	+	FB3sl (M3)
CPMVDI	K	一般耦合：針對特定 NC/PLC 介面訊號的跟隨軸回應		+	+	FB3sl (M3)
CPOF	K	一般耦合：關閉耦合模組		+	+	FB3sl (M3)
CPON	K	一般耦合：開啟耦合模組		+	+	FB3sl (M3)
CPRECOF [6]	G	停用可程式設計的輪廓精確度	m	+		PGAsl
CPRECON	G	啟動可程式設計的輪廓精確度	m	+		PGAsl
CPRES	K	一般耦合：啟用同步主軸耦合的已設定資料		+	-	FB3sl (M3)
CPROT	P	啟動 / 停用通道專屬保護區		+	-	PGAsl
CPROTDEF	P	定義通道專屬保護區		+	-	PGAsl
CPSETTYPE	K	一般耦合：耦合類型		+	+	FB3sl (M3)
CPSYNCOP	K	一般耦合：位置同步「粗調」的臨界值		+	+	FB3sl (M3)
CPSYNCOP2	K	一般耦合：位置同步「粗調」2 的臨界值		+	+	FB3sl (M3)
CPSYNCOV	K	一般耦合：速度同步「粗調」的臨界值		+	+	FB3sl (M3)
CPSYNFIP	K	一般耦合：位置同步「細調」的臨界值		+	+	FB3sl (M3)
CPSYNFIP2	K	一般耦合：位置同步「細調」2 的臨界值		+	+	FB3sl (M3)
CPSYNFIV	K	一般耦合：速度同步「細調」的臨界值		+	+	FB3sl (M3)
CR	A	圓形半徑	s	+		PGsl
CROT	F	旋轉目前座標系統		+	-	PGAsl

操作	類型 1)	含義	W 2)	TP 3)	SA 4)	說明，請參閱 5)
1) 2) 3) 4) 5) 供說明之用，請參閱圖例 (頁 963)。						
CROTS	F	可程式設計的框架以固定角度旋轉（在指定軸中之旋轉）	s	+	-	PGsl
CRPL	F	任何平面中的框架旋轉		+	-	FB1sl (K2)
CSCALE	F	數個軸的比例係數		+	-	PGAsl
CSPLINE	F	三次曲線	m	+		PGAsl
CT	G	具切線變化的圓	m	+		PGsl
CTAB	F	依據由曲線表取得之先導軸位置定義跟隨軸位置		+	+	PGAsl
CTABDEF	P	啟動表之定義		+	-	PGAsl
CTABDEL	P	清除曲線表		+	-	PGAsl
CTABEND	P	停用表之定義		+	-	PGAsl
CTABEXISTS	F	以數字 n 檢查曲線表		+	+	PGAsl
CTABFNO	F	記憶體中仍可使用的曲線表數量		+	+	PGAsl
CTABFPOL	F	記憶體中仍可使用的多項式數量		+	+	PGAsl
CTABFSEG	F	記憶體中仍可使用的曲線區段數量		+	+	PGAsl
CTABID	F	傳回第 n 個曲線表的表編號		+	+	PGAsl
CTABINV	F	依據由曲線表取得之跟隨軸位置定義先導軸位置		+	+	PGAsl
CTABISLOCK	F	利用數字 n 傳回曲線表的鎖定狀態		+	+	PGAsl
CTABLOCK	P	刪除與覆寫，鎖定		+	+	PGAsl
CTABMEMTYP	F	傳回建立曲線表編號 n 的記憶體。		+	+	PGAsl
CTABMPOL	F	記憶體中仍可使用的最大多項式數量		+	+	PGAsl
CTABMSEG	F	記憶體中仍可使用的最大曲線區段數量		+	+	PGAsl
CTABNO	F	在 SRAM 或 DRAM 中已定義之曲線表數量		+	+	FB3sl (M3)
CTABNOMEM	F	在 SRAM 或 DRAM 中已定義之曲線表數量		+	+	PGAsl
CTABPERIOD	F	傳回曲線表編號 n 的表週期性		+	+	PGAsl
CTABPOL	F	記憶體中已使用的多項式數量		+	+	PGAsl
CTABPOLID	F	編號為 n 之曲線表所使用之曲線多項式數量		+	+	PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
CTABSEG	F	記憶體中已使用的曲線區段數量		+	+	PGAsl
CTABSEGID	F	編號為 n 之曲線表所使用之曲線區段數量		+	+	PGAsl
CTABSEV	F	傳回曲線表中某區段之跟隨軸的最終值		+	+	PGAsl
CTABSSV	F	傳回曲線表中某區段之跟隨軸的初始值		+	+	PGAsl
CTABTEP	F	傳回曲線表結尾處之先導軸的值		+	+	PGAsl
CTABTEV	F	傳回曲線表結尾處之跟隨軸的值		+	+	PGAsl
CTABTMAX	F	傳回曲線表中跟隨軸的最大值		+	+	PGAsl
CTABTMIN	F	傳回曲線表中跟隨軸的最小值		+	+	PGAsl
CTABTSP	F	傳回曲線表起始處之先導軸的值		+	+	PGAsl
CTABTSV	F	傳回曲線表起始處之跟隨軸的值		+	+	PGAsl
CTABUNLOCK	P	撤回刪除與覆寫鎖定		+	+	PGAsl
CTOL	A	壓縮函數、方向平滑化與平滑化類型的輪廓允差	m	+		PGAsl
CTRANS	F	數個軸的零點偏移量		+	-	PGAsl
CUT2D [6]	G	2D TRC	m	+		PGsl
CUT2DD	G	相對於差別刀具的 2½ D TRC	m	+		PGsl
CUT2DF	G	相對於目前框架的 2 D TRC（傾斜平面）	m	+		PGsl
CUT2DFD	G	相對於差別刀具的 2½ D TRC，該差別刀具則與目前框架（傾斜平面）相對	m	+		PGsl
CUT3DC	G	圓周銑削的 3D TRC	m	+		PGAsl
CUT3DCC	G	圓周銑削的 3D TRC，考量極限表面，含 3D 半徑補正，加工表面的輪廓	m	+		PGAsl
CUT3DCCD	G	圓周銑削的 3D TRC，考量極限表面，含刀具中心點路徑的差別刀具，進給至極限表面。	m	+		PGAsl
CUT3DCD	G	圓周銑削差別刀具的 3D TRC	m	+		PGAsl
CUT3DF	G	3D TRC 平面銑削，含變更方向	m	+		PGAsl
CUT3DFD	G	平面銑削差別刀具的 3D TRC，含變更方向	m	+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
CUT3DFF	G	3D TRC 平面銑削，含恆定方向。刀具方向為以 G17 - G19 定義的方向，且在某些情況下，由框架旋轉。	m	+		PGAsl
CUT3DFS	G	3D TRC 平面銑削，含恆定方向。以 G17 - G19 定義刀具方向，且不受框架影響。	m	+		PGAsl
CUTCONOF [6]	G	停用刀具半徑補正	m	+		PGsl
CUTCONON	G	啟動刀具半徑補正	m	+		PGsl
CUTMOD	A	啟動「可旋轉刀具偏移資料之修改」（與可定向刀盤連結）	m	+		PGAsl
CUTMODK	A	啟動「可旋轉刀具偏移資料之修改」（與動態鏈定義之方向轉換連結）	m	+		PGAsl
CYCLE60	C (T)	刻字循環		+		PGAsl
CYCLE61	C (T)	面銑削		+		PGAsl
CYCLE62	C (T)	輪廓呼叫		+		PGAsl
CYCLE63	C (T)	輪廓挖槽銑削		+		PGAsl
CYCLE64	C (T)	輪廓挖槽預鑽		+		PGAsl
CYCLE70	C (T)	螺紋銑削		+		PGAsl
CYCLE72	C (T)	路徑銑削		+		PGAsl
CYCLE76	C (T)	銑削矩形島嶼		+		PGAsl
CYCLE77	C (T)	圓形島嶼銑削		+		PGAsl
CYCLE78	C (T)	銑削切割螺紋		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
CYCLE79	C (T)	多稜邊		+		PGAsl
CYCLE81	C (T)	鑽孔，中心鑽孔		+		PGAsl
CYCLE82	C (T)	鑽孔，逆鏜孔		+		PGAsl
CYCLE83	C (T)	深孔鑽		+		PGAsl
CYCLE84	C (T)	無補正夾頭的攻牙		+		PGAsl
CYCLE85	C (T)	鉸孔		+		PGAsl
CYCLE86	C (T)	鏜孔		+		PGAsl
CYCLE92	C (T)	切離		+		PGAsl
CYCLE95	C (T)	沿著輪廓進行材料移除		+		PGAsl
CYCLE98	C (T)	螺紋鏈		+		PGAsl
CYCLE99	C (T)	螺紋切削		+		PGAsl
CYCLE150	C (M)	顯示 / 記錄測量結果		+		BNMsl
CYCLE435	C (T)	計算修整刀具位置		+		PGAsl
CYCLE495	C (T)	形狀修整		+		PGAsl
CYCLE750	C (A)	CYCLE751 的內部操作循環...CYCLE759（包含實際功能呼叫的 MMC 指令）		-		FB3sl (T4)
CYCLE751	C (A)	開啟 / 進行 / 關閉最佳化工作階段		M		FB3sl (T4)

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
CYCLE752	C (A)	將軸加入最佳化工作階段		M		FB3sl(T4)
CYCLE753	C (A)	選擇最佳化模式。		M		FB3sl (T4)
CYCLE754	C (A)	新增 / 移除語言區塊		M		FB3sl (T4)
CYCLE755	C (A)	備份/還原資料區塊		M		FB3sl (T4)
CYCLE756	C (A)	啟動最佳化結果		M		FB3sl (T4)
CYCLE757	C (A)	儲存最佳化資料		M		FB3sl (T4)
CYCLE758	C (A)	正在變更參數值		M		FB3sl (T4)
CYCLE759	C (A)	讀取參數值		M		FB3sl (T4)
Cycle800	C (T)	迴轉		+		PGAsl
CYCLE801	C (T)	網格或框架		+		PGAsl
CYCLE802	C (T)	任意位置		+		PGAsl
CYCLE830	C (T)	深鑽孔 2		+		PGAsl
CYCLE832	C (T)	高速加工設定		+		PGAsl
CYCLE840	C (T)	搭配補正夾頭進行攻牙		+		PGAsl
CYCLE899	C (T)	開孔槽銑削		+		PGAsl
CYCLE930	C (T)	溝槽		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
CYCLE940	C (T)	底切形狀		+		PGAsl
CYCLE951	C (T)	材料移除		+		PGAsl
CYCLE952	C (T)	輪廓溝槽		+		PGAsl
CYCLE961	C (M)	決定工件角隅（內側或外側）的位置並插入作為工作偏移量。		+		BNMsl
CYCLE971	C (M)	校正刀具探針、測量刀長及 / 或刀具半徑（僅適用於銑削）		+		BNMsl
CYCLE973	C (M)	校正工件表面上或溝槽內的工件探針（僅適用於車削）		+		BNMsl
CYCLE974	C (M)	決定所選測量軸內的工件零點，使用 1 點測量決定刀具偏移量（僅適用於車削）。		+		BNMsl
CYCLE976	C (M)	在完全於工作平面內的校正環內或校正球上校正工件探針，或在特定軸與方向的稜邊上校正		+		BNMsl
CYCLE977	C (M)	決定平面內的中心，以及寬度或直徑		+		BNMsl
CYCLE978	C (M)	測量工件座標系統內稜邊的位置		+		BNMsl
CYCLE979	C (M)	決定平面的中心、測量圓形區段的半徑。		+		BNMsl
CYCLE982	C (M)	校正刀具探針、測量車削、鑽孔與銑刀（僅適用於車銷）		+		BNMsl
CYCLE994	C (M)	使用 2 點測量，決定所選測量軸內的工件零點（僅適用於車削）。		+		BNMsl
CYCLE995	C (M)	測量工具機上主軸的角度		+		BNMsl
CYCLE996	C (M)	決定使用旋轉軸進行座標結構配置轉換之轉換相關資料		+		BNMsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
CYCLE997	C (M)	決定球的中心和直徑、測量三個分散球的中心		+		BNMsl
CYCLE998	C (M)	決定當成工作平面的表面（平面）之角度位置，決定工件座標系統內稜邊的角度。		+		BNMsl
CYCLE4071	C (T)	於反轉點進給之縱向研磨		+		PGAsl
CYCLE4072	C (T)	於反轉點進給和取消訊號之縱向研磨		+		PGAsl
CYCLE4073	C (T)	連續進給之縱向研磨		+		PGAsl
CYCLE4074	C (T)	連續進給和取消訊號之縱向研磨		+		PGAsl
CYCLE4075	C (T)	於反轉點進給之表面研磨		+		PGAsl
CYCLE4077	C (T)	於反轉點進給和取消訊號之表面研磨		+		PGAsl
CYCLE4078	C (T)	連續進給之表面研磨		+		PGAsl
CYCLE4079	C (T)	間歇進給之表面研磨		+		PGAsl

操作 D ... F

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
D	O	刀具偏移編號		+		PGsl
D0	O	使用 D0 時，刀具的偏移量會無效		+		PGsl
DAC	K	絕對非模態、軸專屬直徑程式設計	s	+		PGsl
DC	K	旋轉軸之絕對直徑，直接逼近位置	s	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1][2][3][4][5] 供說明之用，請參閱圖例 (頁 963)。						
DCI	K	指派資料分類 I (= Individual，個人) (僅適用於 SINUMERIK 828D)		+		PGAsl
DCM	K	指派資料分類 M (= Manufacturer，製造商) (僅適用於 SINUMERIK 828D)		+		PGAsl
DCU	K	指派資料分類 U (= User，使用者) (僅適用於 SINUMERIK 828D)		+		PGAsl
DEF	K	變數定義		+		PGAsl
DEFAULT	K	CASE 分支內之分支		+		PGAsl
DEFINE	K	用於巨集定義的關鍵字		+		PGAsl
DELAYFSTOF	P	定義停止延遲區段的結束	m	+	-	PGAsl
DELAYFSTON	P	定義停止延遲區段的開始	m	+	-	PGAsl
DELDL	F	刪除附加的偏移量		+	-	PGAsl
DELDTG	P	刪除剩餘距離		-	+	FBSYsl
DELETE	P	刪除指定的檔案。可以用路徑和檔案識別碼來指定檔名。		+	-	PGAsl
DELMLOWNER	F	刪除刀具的擁有者刀庫位置		+	-	FBWsl
DELMLRES	F	刪除保留的刀庫位置		+	-	FBWsl
DELMT	P	刪除多重刀具		+	-	FBWsl
DELOBJ	F	刪除動態鍊元素、保護區、保護區元素、碰撞對及轉換資料		+		PGAsl
DELT	P	刪除刀具		+	-	FBWsl
DELTC	P	刪除刀把資料記錄		+	-	FBWsl
DELTOOLENV	F	刪除說明刀具環境的資料記錄		+	-	PGAsl
DIACYCOFA	K	軸專屬模態半徑程式設計：於循環中 OFF	m	+		FB1sl (P1)
DIAM90	G	適用於 G90 之直徑程式設計，適用於 G91 之半徑程式設計	m	+		PGAsl
DIAM90A	K	適用於 G90 及 AC 的軸專屬模態直徑程式設計，適用於 G91 及 IC 的半徑程式設計	m	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1][2][3][4][5] 供說明之用，請參閱圖例 (頁 963)。						
DIAMCHAN	K	MD 軸函數中所有處於直徑程式設計通道狀態之軸的傳輸		+		PGsl
DIAMCHANA	K	直徑程式設計通道狀態的傳輸		+		PGsl
DIAMCYCOF	G	通道專屬直徑程式設計：於循環中 OFF	m	+		FB1sl (P1)
DIAMOF [6]	G	直徑程式設計：OFF 正常位置，請參閱機台製造商	m	+		PGsl
DIAMOFA	K	軸專屬模態半徑程式設計：OFF 正常位置，請參閱機台製造商	m	+		PGsl
DIAMON	G	直徑程式設計：開啟	m	+		PGsl
DIAMONA	K	軸專屬模態半徑程式設計：ON 如需啟用，請參閱機台製造商	m	+		PGsl
DIC	K	相對非模態、軸專屬直徑程式設計	s	+		PGsl
DILF	O	回退路徑（長度）	m	+		PGsl
DISABLE	P	中斷 OFF		+	-	PGAsl
DISC	O	變化圓弧過衝刀具半徑補正	m	+		PGsl
DISCL	O	快速進給動作之結束位置與加工平面間的間隙		+		PGsl
DISPLOF	PA	抑制目前單節顯示		+		PGAsl
DISPLON	PA	撤回對目前單節顯示的抑制		+		PGAsl
DISPR	O	重新定位之路徑差異	s	+		PGAsl
DISR	O	重新定位距離	s	+		PGAsl
DISRP	O	在平滑化逼近與回退期間，退刀平面與加工平面之間的距離		+		PGsl
DITE	O	螺紋旋出路徑	m	+		PGsl
DITS	O	螺紋旋進路徑	m	+		PGsl
DIV	K	整數除法		+		PGAsl
DL	O	選擇視位置而定的附加刀具偏移量（DL，總設定偏移量）	m	+		PGAsl
DO	O	同步動作的關鍵字，當條件滿足時會觸發動作		-	+	FBSYsl
DRFOF	P	停用手輪偏移 (DRF)	m	+	-	PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1][2][3][4][5] 供說明之用，請參閱圖例 (頁 963)。						
DRIVE	G	路徑速率相依式加速	m	+		PGAsl
DRIVEA	P	啟動程式設計之軸的膝形加速特性		+	-	PGAsl
DYNFINISH	G	平滑化精加工之動態回應	m	+		PGAsl
DYNNORM [6]	G	標準動態回應	m	+		PGAsl
DYNPOS	G	定位模式、攻牙的動態回應	m	+		PGAsl
DYNROUGH	G	粗加工的動態回應	m	+		PGAsl
DYNSEMIFIN	G	精加工的動態回應	m	+		PGAsl
DZERO	P	將 TO 元件的所有 D 數量標記為無效		+	-	PGAsl
EAUTO	G	以最後 3 個點定義最後的曲線區段	m	+		PGAsl
EGDEF	P	定義電子齒輪		+	-	PGAsl
EGDEL	P	刪除跟隨軸的耦合定義		+	-	PGAsl
EGOFC	P	連續地關閉電子齒輪		+	-	PGAsl
EGOFS	P	選擇性地關閉電子齒輪		+	-	PGAsl
EGON	P	啟動電子齒輪		+	-	PGAsl
EGONSYN	P	啟動電子齒輪		+	-	PGAsl
EGONSYNE	P	使用逼近模式的規格啟動電子齒輪		+	-	PGAsl
ELSE	K	程式分支，若未滿足 IF 條件		+		PGAsl
ENABLE	P	中斷 ON		+	-	PGAsl
ENAT [6]	G	自然轉變至下一個移動單節	m	+		PGAsl
ENDFOR	K	FOR 計數器迴圈之最終指令列		+		PGAsl
ENDIF	K	IF 分支之最終指令列		+		PGAsl
ENDLABEL	K	使用 REPEAT 的工件程式重覆之結尾標籤		+		PGAsl、FB1sl (K1)
ENDLOOP	K	無限程式迴圈 LOOP 之最終指令列		+		PGAsl
ENDPROC	K	含起始指令列 PROC 之程式的最終指令列		+		
ENDWHILE	K	WHILE 迴圈之最終指令列		+		PGAsl
ESRR	P	參數化驅動器中的驅動自主化 ESR 回退		+		PGAsl
ESRS	P	參數化驅動器中的驅動自動化 ESR 關機		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1][2][3][4][5] 供說明之用，請參閱圖例 (頁 963)。						
ETAN	G	沿切線變化至曲線開始處下一個移動單節	m	+		PGAsl
EVERY	K	當條件從 FALSE 變化為 TRUE 時執行同步動作		-	+	FBSYsl
EX	K	以指數標記法配置值的關鍵字		+		PGAsl
EXECSTRING	P	字串變數的傳輸，該字串變數具有執行中的工件程式行		+	-	PGAsl
EXECTAB	P	執行取自動作表之元素		+	-	PGAsl
EXECUTE	P	程式執行 ON		+	-	PGAsl
EXP	F	指數函數範例		+	+	PGAsl
EXTCALL	O	執行外部子程式		+	+	PGAsl
EXTCLOSE	P	關閉已打開供寫入的外部裝置 / 檔案		+	-	PGAsl
EXTERN	K	宣告具有參數傳輸的子程式		+		PGAsl
EXTOPEN	P	打開該通道的外部裝置 / 檔案以供寫入		+	-	PGAsl
F	O	進給率值 （搭配 G4 時，停頓時間亦使用 F 程式設計）		+	+	PGsl
FA	K	軸進給率	m	+	+	PGsl
FAD	O	軟逼近與回退的進給率		+		PGsl
FALSE	K	邏輯常數：不正確		+	+	PGAsl
FB	O	非模態進給率		+		PGsl
FCTDEF	P	定義多項式函數		+	-	PGAsl
FCUB	G	依三次曲線之定進給率變數	m	+		PGAsl
FD	O	手輪手動倍率之路徑進給率	s	+		PGsl
FDA	K	手輪手動倍率軸進給率	s	+		PGsl
FENDNORM [6]	G	角落減速 OFF	m	+		PGAsl
FFWOF [6]	G	前饋控制 OFF	m	+		PGAsl
FFWON	G	前饋控制 ON	m	+		PGAsl
FGREF	K	旋轉軸或方向軸之路徑參考係數的參考半徑（向量插補）	m	+		PGsl
FGROUP	P	含路徑進給率之軸的定義		+	-	PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1][2][3][4][5] 供說明之用，請參閱圖例 (頁 963)。						
FI	K	存取框架資料之參數：微調偏移量		+		PGAsl
FIFOCTRL	G	前置處理緩衝器之控制	m	+		PGAsl
FILEDATE	P	傳回最近對檔案做寫入存取的日期		+	-	PGAsl
FILEINFO	P	傳回列出 FILEDATE、FILESIZE、FILESTAT 與 FILETIME 的摘要資訊		+	-	PGAsl
FILESIZE	P	傳回目前的檔案大小		+	-	PGAsl
FILESTAT	P	傳回檔案之讀取、寫入、執行、顯示、刪除權限之狀態 (rwxsd)		+	-	PGAsl
FILETIME	P	傳回最近對檔案做寫入存取的時間		+	-	PGAsl
FINEA	K	於達到「精確停止微調」時動作結束	m	+		PGAsl
FL	K	同步軸的臨界速率	m	+		PGsl
FLIN	G	進給線性變數	m	+		PGAsl
FMA	K	軸的多重進給率	m	+		PGsl
FNORM [6]	G	垂直於 DIN 66025 之進給率	m	+		PGAsl
FOC	K	非模態扭矩 / 力量限制	s	-	+	FBSYsl
FOCOF	K	關閉模態扭矩 / 力量限制	m	-	+	FBSYsl
FOCON	K	開啟模態扭矩 / 力量限制	m	-	+	FBSYsl
FOR	K	含固定通過次數之計數器迴圈		+		PGAsl
FP	O	固定點：待逼近之固定點數量	s	+		PGsl
FPO	K	透過多項式程式設計的進給率特性		+		PGAsl
FPR	P	旋轉軸識別碼		+	-	PGsl
FPRAOF	P	停用旋轉進給率		+	-	PGsl
FPRAON	P	啟動旋轉進給率		+	-	PGsl
FRAME	K	座標系統定義的資料類型		+		PGAsl
FRC	O	半徑及倒角之進給率	s	+		PGsl
FRCM	O	半徑及倒角之進給率，模態	m	+		PGsl
FROM	K	一旦滿足該條件且啟用同步動作時便會執行該動作		-	+	FBSYsl
FTOC	P	更改微調刀具偏移		-	+	FBSYsl
FTOCOF [6]	G	線上微調刀具偏移量 OFF	m	+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1][2][3][4][5] 供說明之用，請參閱圖例 (頁 963)。						
FTOCON	G	線上微調刀具偏移量 ON	m	+		PGAsl
FXS	K	移動至固定停止點 ON	m	+	+	PGsl
FXST	K	移動至固定停止點的轉矩限制	m	+	+	PGsl
FXSW	K	監控視窗以移動至固定停止點		+	+	PGsl
FZ	K	刀齒進給率	m	+		PGsl

操作 G ... L

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
G0	G	線性插補含快速移動（快速移動動作）	m	+		PGsl
G1 [6]	G	使用進給進行線性插補（線性插補）	m	+		PGsl
G2	G	順時針圓弧插補	m	+		PGsl
G3	G	逆時針圓弧插補	m	+		PGsl
G4	G	停頓時間，預設	s	+		PGsl
G5	G	斜向直面進給研磨	s	+		PGAsl
G7	G	傾斜降切研磨時之補正動作	s	+		PGAsl
G9	G	精確停止－減速	s	+		PGsl
G17 [6]	G	作用平面 X/Y 之選擇	m	+		PGsl
G18	G	作用平面 Z/X 之選擇	m	+		PGsl
G19	G	作用平面 Y/Z 之選擇	m	+		PGsl
G25	G	工作區下限	s	+		PGsl
G26	G	工作區上限	s	+		PGsl
G33	G	以恆定螺距進行螺紋切削	m	+		PGsl
G34	G	使用線性遞增螺距進行螺紋切削	m	+		PGsl
G35	G	使用線性遞減螺距進行螺紋切削	m	+		PGsl
G40 [6]	G	刀具半徑補正 OFF	m	+		PGsl
G41	G	刀具半徑補正輪廓左側	m	+		PGsl
G42	G	刀具半徑補正輪廓右側	m	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
G53	G	抑制目前的零點偏移量（非模態）	s	+		PGsl
G54	G	第 1 可設定零點偏移量	m	+		PGsl
G55	G	第 2 可設定零點偏移量	m	+		PGsl
G56	G	第 3 可設定零點偏移量	m	+		PGsl
G57	G	第 4 可設定零點偏移量	m	+		PGsl
G58 (840D sl)	G	絕對的可程式設計工件偏移量（粗調偏移量）	s	+		PGsl
G58 (828D)	G	第 5 可設定零點偏移量	m	+		PGsl
G59 (840D sl)	G	附加的可程式設計工件偏移量（微調偏移量）	s	+		PGsl
G59 (828D)	G	第 6 可設定零點偏移量	m	+		PGsl
G60 [6]	G	精確停止－減速	m	+		PGsl
G62	G	啟動刀具半徑偏移（G41，G42）時，於內角的轉角減速	m	+		PGAsl
G63	G	搭配補正夾頭進行攻牙	s	+		PGsl
G64	G	連續路徑模式	m	+		PGsl
G70	G	用於幾何規格（長度）的英制尺寸	m	+	+	PGsl
G71 [6]	G	用於幾何規格（長度）的公制尺寸	m	+	+	PGsl
G74	G	參考點搜尋	s	+		PGsl
G75	G	逼近固定點	s	+		PGsl
G90 [6]	G	絕對尺寸	m/s	+		PGsl
G91	G	增量尺寸	m/s	+		PGsl
G93	G	反向時間進給率 rpm	m	+		PGsl
G94 [6]	G	線性進給率 F，單位為毫米 / 分或英吋 / 分以及度 / 分	m	+		PGsl
G95	G	旋轉進給率 F，單位為毫米 / 轉或英吋 / 轉	m	+		PGsl
G96	G	旋轉進給率（供 G95 使用）及恆定切削率	m	+		PGsl
G97	G	旋轉進給率及恆定主軸速度（恆定切削率關閉）	m	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
G110	G	極點程式設計相對於上一次已程式設計之設定點的位置	s	+		PGsl
G111	G	極點程式設計相對於目前工件座標系統的零點	s	+		PGsl
G112	G	極點程式設計相對於上一個有效的極點	s	+		PGsl
G140 [6]	G	以 G41/G42 定義之 SAR 逼近方向	m	+		PGsl
G141	G	朝輪廓左側逼近之 SAR 逼近方向	m	+		PGsl
G142	G	朝輪廓右側逼近之 SAR 逼近方向	m	+		PGsl
G143	G	相依於切線之 SAR 逼近方向	m	+		PGsl
G147	G	以直線方式緩逼近	s	+		PGsl
G148	G	以直線方式緩返回	s	+		PGsl
G153	G	抑制含基本框架的目前框架	s	+		PGsl
G247	G	以四分之一圓方式緩逼近	s	+		PGsl
G248	G	以四分之一圓方式緩返回	s	+		PGsl
G290 [6]	G	切換至 SINUMERIK 模式 ON	m	+		FBWsl
G291	G	切換至 ISO2/3 模式 ON	m	+		FBWsl
G331	G	剛性攻牙、正螺距、順時針	m	+		PGsl
G332	G	剛性攻牙、負螺距、逆時針	m	+		PGsl
G335	G	順時針方向車削凸螺紋	m	+		PGsl
G336	G	逆時針方向車削凸螺紋	m	+		PGsl
G340 [6]	G	立體逼近單節（同時進行深度與平面（螺旋線））	m	+		PGsl
G341	G	垂直軸 (z) 上之初始進給，然後逼近平面	m	+		PGsl
G347	G	以半圓緩逼近	s	+		PGsl
G348	G	以半圓緩返回	s	+		PGsl
G450 [6]	G	圓形變化	m	+		PGsl
G451	G	同距交點	m	+		PGsl
G460 [6]	G	啟動逼近與回退單節的碰撞偵測	m	+		PGsl
G461	G	將圓弧插入至 TRC 單節中	m	+		PGsl
G462	G	將直線插入至 TRC 單節中	m	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
G500 [6]	G	停用所有可調整的框架，啟用基本框架	m	+		PGsl
G505 ... G599	G	5 ... 99 可設定工件偏移量	m	+		PGsl
G601 [6]	G	於精確停止微調處之單節變更	m	+		PGsl
G602	G	於精確停止粗調處之單節變更	m	+		PGsl
G603	G	於 IPO 單節結尾處之單節變更	m	+		PGsl
G621	G	所有角的轉角減速	m	+		PGAsl
G641	G	具有符合路徑條件（= 可程式設計的倒圓角間隙）之平滑化的連續路徑模式	m	+		PGsl
G642	G	具在定義允差內之平滑化的連續路徑模式	m	+		PGsl
G643	G	具有在定義允差內（單節內部）之平滑化的連續路徑模式	m	+		PGsl
G644	G	具有含最大可能動態回應之平滑化的連續路徑模式	m	+		PGsl
G645	G	具有在定義允差內之平滑化與切線單節變化的連續路徑模式	m	+		PGsl
G700	G	用於幾何與技術規格（長度、進給率）的英制尺寸	m	+	+	PGsl
G710 [6]	G	用於幾何與技術規格（長度、進給率）的公制尺寸	m	+	+	PGsl
G810 [6]、...、G819	G	保留給 OEM 使用者的 G 群組		+		PGAsl
G820 [6]、...、G829	G	保留給 OEM 使用者的 G 群組		+		PGAsl
G931	G	以移動時間、停用恆定路徑速率方面指定進給率	m	+		
G942	G	凍結線性進給率及恆定切削速率或主軸速率	m	+		
G952	G	凍結旋轉進給率及恆定切削速率或主軸速率	m	+		
G961	G	線性進給率（供 G94 使用）及恆定切削率	m	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
G962	G	線性進給率或旋轉進給率及恆定切削率	m	+		PGsl
G971	G	線性進給率及恆定主軸速度（恆定切削率關閉）	m	+		PGsl
G972	G	線性進給率或旋轉進給率及恆定主軸速度（恆定切削率關閉）	m	+		PGsl
G973	G	旋轉進給率，無主軸速度限制及恆定主軸速度（G97 無 LIMS，適用於 ISO 模式）	m	+		PGsl
GEOAX	P	指派新通道軸給幾何座標軸 1 - 3		+	-	PGAsl
GET	P	更換通道間啟用的軸		+	+	PGAsl
GETACTT	F	由刀具群組中找出具相同名稱的啟用中刀具		+	-	FBWsl
GETACTTD	F	取得具有絕對 D 編號的 T 編號		+	-	PGAsl
GETD	P	直接更換通道間的軸		+	-	PGAsl
GETDNO	F	傳回刀具 (T) 的刀刃 (CE) 之 D 編號		+	-	PGAsl
GETEXET	P	讀取載入的 T 編號		+	-	FBWsl
GETFREELOC	P	為已知刀具在刀庫中尋找可用空間		+	-	FBWsl
GETSELT	P	傳回選取的 T 編號		+	-	FBWsl
GETT	F	為刀具名稱取得 T 編號		+	-	FBWsl
GETTCOR	F	讀出刀長及 / 或刀長元件		+	-	PGAsl
GETTENV	F	讀取 T、D 與 DL 編號		+	-	PGAsl
GETVARAP	F	讀取系統 / 使用者變數的存取權限		+	-	PGAsl
GETVARDFT	F	讀取系統 / 使用者變數的預設值		+	-	PGAsl
GETVARLIM	F	讀取系統 / 使用者變數的限制值		+	-	PGAsl
GETVARPHU	F	讀取系統 / 使用者變數的實體元件		+	-	PGAsl
GETVARTYP	F	讀取系統 / 使用者變數的資料類型		+	-	PGAsl
GFRAME0 ... GFRAME100	G	啟用通道中 <n> 資料管理的研磨框架	m	+		PGsl
GOTO	K	先向前然後向後之跳躍操作（其初始方向是朝程式結尾，之後朝程式開始）		+		PGAsl
GOTOB	K	向後跳躍（前往程式開始）		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
GOTOC	K	同 GOTO，但抑制警報 14080「找不到跳躍目的地」		+		PGAsl
GOTOF	K	向前跳躍（前往程式結尾）		+		PGAsl
GOTOS	K	向後跳躍至程式開始處		+		PGAsl
GP	K	將位置屬性間接程式設計的關鍵字		+		PGAsl
GROUP_ ADDEND	C (T)	嘗試切削添加結束		+		PGAsl
GROUP_BEGIN	C (T)	程式群組開始處		+		PGAsl
GROUP_END	C (T)	程式群組結束		+		PGAsl
GWPSOF	P	取消選擇研磨輪圓柱恆定速率 (GWPS)	s	+	-	PGsl
GWPSON	P	選擇研磨輪圓柱恆定速率 (GWPS)	s	+	-	PGsl
H...	O	輔助函數輸出至 PLC		+	+	PGsl/FB1sl (H2)
HOLES1	C (T)	排孔		+		PGAsl
HOLES2	C (T)	環孔		+		PGAsl
I	O	插補參數	s	+		PGsl
I1	O	中間點座標	s	+		PGsl
IC	K	增量尺寸	s	+		PGsl
ICYCOF	P	所有單節在 ICYCOF 之後都會在單一插補循環中處理		+	+	FBSYsl
ICYCON	P	ICYCON 之後的每一個技術循環單節皆在不同的插補循環中處理		+	+	FBSYsl
ID	K	模態同步動作的識別碼	m	-	+	FBSYsl
IDS	K	模態靜態同步動作的識別碼		-	+	FBSYsl
IF	K	工件程式 / 技術循環中傳統跳躍函數之介紹		+	+	PGAsl
INDEX	F	定義輸入字串中字元的索引值		+	-	PGAsl
INICF	K	在 NEWCONF 時將變數初始化		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
INIPO	K	在 POWER ON 將變數初始化		+		PGAsl
INIRE	K	在重置時將變數初始化		+		PGAsl
INIT	P	選擇特定 NC 程式，供特定通道執行		+	-	PGAsl
INITIAL		產生一個適用所有區域的 INI 檔		+		PGAsl
INT	K	資料類型：含正負號之整數		+		PGAsl
INTERSEC	F	計算兩個輪廓元素間的交叉點		+	-	PGAsl
INVCCW	G	軌跡漸開線，逆時針	m	+		PGsl
INVCW	G	軌跡漸開線，順時針	m	+		PGsl
INVFRAME	F	從框架計算出反向框架		+	-	FB1sl (K2)
IP	K	變數插補參數		+		PGAsl
IPOBRKA	P	自啟用煞車斜率開始動作條件	m	+	+	
IPOENDA	K	於抵達「IPO 停止」時動作結束	m	+		PGAsl
IPTRLOCK	P	將下一個加工函式單節中無法追蹤之程式區段的啟動予以凍結。	m	+	-	PGAsl
IPTRUNLOCK	P	中斷時，設定在目前單節中無法追蹤之程式區段的結束。	m	+	-	PGAsl
IR	O	車削凸螺紋時的圓座標中心 (X 軸)		+		PGsl
ISAXIS	F	檢查幾何軸 1 是否被指定為參數		+	-	PGAsl
ISD	O	插入深度	m	+		PGAsl
ISFILE	F	檢查檔案在 NC 應用程式記憶體中是否存在		+	-	PGAsl
ISNUMBER	F	檢查輸入字串是否可轉換為數字		+	-	PGAsl
ISOCALL	K	間接呼叫以 ISO 語言程式設計的程式		+		PGAsl
ISVAR	F	檢查傳輸參數是否包含在 NC 中已宣告的變數		+	-	PGAsl
J	O	插補參數	s	+		PGsl
J1	O	中間點座標	s	+		PGsl
JERKA	P	透過 MD 為已程式設計之軸啟動加速度回應		+	-	
JERKLIM	K	最大軸向震動之減少或過衝	m	+		PGAsl
JERKLIMA	K	最大軸向震動之減少或過衝	m	+	+	PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
JR	O	Center of circle coordinate (Y axis) when turning a convex thread		+		PGsl
K	O	插補參數	s	+		PGsl
K1	O	中間點座標	s	+		PGsl
KONT	G	在刀具偏移上沿輪廓移動	m	+		PGsl
KONTC	G	利用連續曲率多項式逼近 / 回退	m	+		PGsl
KONTT	G	利用連續切線多項式逼近 / 回退	m	+		PGsl
KR	O	Center of circle coordinate (Z axis) when turning a convex thread		+		PGsl
L	O	子程式編號	s	+	+	PGAsl
LEAD	O	螺距角度 第一基本刀具方位 第二方向多項式	m	+		PGAsl
LEADOF	P	軸主動值耦合關閉		+	+	PGAsl
LEADON	P	軸主動值耦合開啟		+	+	PGAsl
LENTOAX	F	將有關啟用刀具的刀長配置資訊 L1、L2 與 L3 提供給橫座標、縱座標及備選座標		+	-	PGAsl
LFOF [6]	G	關閉螺紋切削時的快速回退	m	+		PGsl
LFON	G	開啟螺紋切削時的快速回退	m	+		PGsl
LFPOS	G	將以 POLFMASK 或 POLFMLIN 宣告的軸回退至以 POLF 程式設計的絕對軸位置	m	+		PGsl
LFTXT [6]	G	快速回退時的回退移動之平面由目前刀具方向的路徑切線決定	m	+		PGsl
LFWP	G	快速回退時的回退移動之平面由目前的工作平面 (G17/G18/G19) 決定	m	+		PGsl
LIFTFAST	K	快速回退		+		PGsl
LIMS	K	G96/G961 與 G97 的 速度臨界值	m	+		PGsl
LLI	K	變數的下臨界值		+		PGAsl
LN	F	自然對數		+	+	PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
LOCK	P	利用 ID 停用同步化動作（停止技術循環）		-	+	FBSYsl
LONGHOLE	C (T)	長孔形		+		PGAsl
LOOP	K	無限迴圈之介紹		+		PGAsl

操作 M ... R

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
M0		程式設計停止		+	+	PGsl
M1		選擇性停止		+	+	PGsl
M2		程式結束，主程式（如同 M30）		+	+	PGsl
M3		主軸順時針旋轉		+	+	PGsl
M4		主軸逆時針旋轉		+	+	PGsl
M5		主軸停止		+	+	PGsl
M6		換刀		+	+	PGsl
M17		子程式結尾		+	+	PGsl
M19		主軸定位到 SD43240 中的輸入位置		+	+	PGsl
M30		程式結束，主程式（如同 M2）		+	+	PGsl
M40		自動檔位變更		+	+	PGsl
M41 ... M45		變速檔 1 ... 5		+	+	PGsl
M70		變化至軸模式		+	+	PGsl
MASLDEF	P	定義主動 / 從屬軸群組		+	+	PGAsl
MASLDEL	P	解除主動 / 從屬軸群組之耦合並清除分組定義		+	+	PGAsl
MASLOF	P	停用暫時耦合		+	+	PGAsl
MASLOFS	P	透過從屬軸自動停止停用暫時耦合		+	+	PGAsl
MASLON	P	啟用暫時耦合		+	+	PGAsl
MATCH	F	在字串中搜尋字串		+	-	PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
MAXVAL	F	兩變數中較大數值（算術函數）		+	+	PGAsl
MCALL	K	模態子程式呼叫		+		PGAsl
MEAC	K	不含刪除剩餘距離的持續軸量測	s	+	+	PGAsl
MEAFRAME	F	由量測點進行之框架計算		+	-	PGAsl
MEAS	O	測量時含刪除的剩餘距離	s	+		PGAsl
MEASA	K	含刪除剩餘距離的軸量測	s	+	+	PGAsl
MEASURE	F	工件與刀具量測的計算方法		+	-	FB1sl (M5)
MEAW	O	不含刪除剩餘距離的量測	s	+		PGAsl
MEAWA	K	不含刪除剩餘距離的軸量測	s	+	+	PGAsl
MI	K	存取框架資料：鏡射		+		PGAsl
MINDEX	F	定義輸入字串中字元的索引值		+	-	PGAsl
MINVAL	F	兩變數中較小數值（算術函數）		+	+	PGAsl
MIRROR	G	可程式設計的鏡像	s	+		PGAsl
MMC	P	由 HMI 上之工件程式以互動方式呼叫對話方塊視窗		+	-	PGAsl
MOD	K	模態除法		+		PGAsl
MODAXVAL	F	決定模態旋轉軸之模態位置		+	-	PGAsl
MOV	K	啟動定位軸		-	+	FBSYsl
MOVT	O	指定沿刀具方向移動動作的終點				FB1(K2)
MSG	P	可程式設計的訊息	m	+	-	PGsl
MVTOOL	P	用於移動刀具的語言指令		+	-	FBWsl
N	O	NC 輔助單節編號		+		PGsl
NAMETOINT	F	判定系統變數索引		+		PGAsl
NC	K	指定資料的有效範圍		+		PGAsl
NEWCONF	P	套用修改後的機械資料（對應至「啟動機械資料」）		+	-	PGAsl
NEWMT	F	建立新的多重刀具		+	-	FBWsl
NEWT	F	建立新刀具		+	-	FBWsl
NORM [6]	G	起點及終點之標準設定，含刀具偏移量	m	+		PGsl
NOT	K	邏輯 NOT（否定）		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
NPROT	P	機台專屬保護區 ON/OFF		+	-	PGAsl
NPROTDEF	P	定義機台專屬保護區		+	-	PGAsl
NUMBER	F	將輸入字串轉換成數字		+	-	PGAsl
OEMIPO1	G	OEM 插補 1	m	+		PGAsl
OEMIPO2	G	OEM 插補 2	m	+		PGAsl
OF	K	含 CASE 分支之關鍵字		+		PGAsl
OFFN	O	已程式設計之輪廓的允差	m	+		PGsl
OMA1	O	OEM 位址 1	m	+		PGAsl
OMA2	O	OEM 位址 2	m	+		PGAsl
OMA3	O	OEM 位址 3	m	+		PGAsl
OMA4	O	OEM 位址 4	m	+		PGAsl
OMA5	O	OEM 位址 5	m	+		PGAsl
OR	K	邏輯運算子，OR 操作		+		PGAsl
ORIAXES	G	機械軸或方向軸之線性插補	m	+		PGAsl
ORIAXPOS	G	透過虛擬方向軸搭配旋轉軸位置取得之方向角	m	+		PGAsl
ORIC [6]	G	在待插入之圓弧單節上外部角落處的方向變更是疊印的	m	+		PGAsl
ORICONCCW	G	圓柱面上逆時針方向之插補	m	+		PGAsl/FB3sl (F3)
ORICONCW	G	圓柱面上順時針方向之插補	m	+		PGAsl/FB3sl (F4)
ORICONIO	G	在圓柱面上依照中間方向設定進行插補	m	+		PGAsl/FB3sl (F4)
ORICONTO	G	在圓柱面上依照切線變化進行插補（最終方向）	m	+		PGAsl/FB3sl (F5)
ORICURVE	G	搭配刀具兩接點之規格進行方向插補	m	+		PGAsl/FB3sl (F6)
ORID	G	方向變更在圓弧單節之前進行	m	+		PGAsl
ORIEULER [6]	G	透過尤拉角取得之方向角	m	+		PGAsl
ORIMKS	G	機械座標系統中之刀具方向	m	+		PGAsl
ORIPATH	G	與路徑相關的刀具方向	m	+		PGAsl
ORIPATHS	G	刀具相對於路徑之方向，方向特性中的上下變動經平滑化化	m	+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
ORIPLANE	G	平面中插補 （對應於 ORIVECT）， 大半徑圓弧插補	m	+		PGAsl
ORIRESET	P	具有最多 3 個方向軸的初始刀具方向		+	-	PGAsl
ORIROTA [6]	G	絕對旋轉方向的旋轉角度	m	+		PGAsl
ORIROTC	G	相對於路徑切線之切線旋轉向量	m	+		PGAsl
ORIROTR	G	相對於起始方向及結束方向間之平面的旋轉角度	m	+		PGAsl
ORIROTT	G	相對於方向向量上之變更的旋轉角度	m	+		PGAsl
ORIRPY	G	透過 RPY 角度 (XYZ) 取得之方向角度	m	+		PGAsl
ORIRPY2	G	透過 RPY 角度 (ZYX) 取得之方向角度	m	+		PGAsl
ORIS	O	變更方向	m	+		PGAsl
ORISOF [6]	G	方向特性平滑化 OFF	m	+		PGAsl
ORISOLH	F	計算方向		+		PGAsl
ORISON	G	方向特性平滑化 ON	m	+		PGAsl
ORIVECT [6]	G	大圓弧插補（與 ORIPLANE 相同）	m	+		PGAsl
ORIVIRT1	G	透過虛擬方向軸取得之方向角度（定義 1）	m	+		PGAsl
ORIVIRT2	G	透過虛擬方向軸取得之方向角度（定義 1）	m	+		PGAsl
ORIWKS [6]	G	工件座標系統中之刀具方向	m	+		PGAsl
OS	K	振盪開啟 / 關閉		+		PGAsl
OSB	K	振盪：起點	m	+		FB1sl (P5)
OSC	G	連續性刀具方向平滑化處理	m	+		PGAsl
OSCILL	K	軸：1 - 3 進給軸	m	+		PGAsl
OSCTRL	K	振盪選項	m	+		PGAsl
OSD	G	利用 SD 指定平滑化距離進行刀具方向之平滑化	m	+		PGAsl
OSE	K	振盪結束位置	m	+		PGAsl
OSNSC	K	振盪：滑磨循環編號	m	+		PGAsl
OSOF [6]	G	刀具方向平滑化處理 OFF	m	+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
OSP1	K	振盪：左反轉點	m	+		PGAsl
OSP2	K	振盪右反轉點	m	+		PGAsl
OSS	G	於單節結尾進行之刀具方向平滑化處理	m	+		PGAsl
OSSE	G	於單節開始及結尾進行之刀具方向平滑化處理	m	+		PGAsl
OST	G	利用 SD 以角度為單位指定角度允差對刀具方向進行平滑（與設計之方向特性間最大的偏差）	m	+		PGAsl
OST1	K	振盪：左逆轉點之停止點	m	+		PGAsl
OST2	K	振盪：右逆轉點之停止點	m	+		PGAsl
OTOL	A	壓縮函數、方向平滑化與平滑化類型的方向允差	m	+		PGAsl
OVR	K	速度偏移量	m	+		PGAsl
OVRA	K	軸速度偏移量	m	+	+	PGAsl
OVRRAP	K	快速移動手動超調	m	+		PGAsl
P	O	子程式重複數量		+		PGAsl
PAROT	G	對齊工件上的工件座標系統	m	+		PGsl
PAROTOF [6]	G	停用與相對於工件的框架旋轉	m	+		PGsl
PCALL	K	利用絕對路徑及參數傳輸呼叫子程式		+		PGAsl
PDELAYOF	G	含延遲的沖孔 OFF	m	+		PGAsl
PDELAYON [6]	G	含延遲的沖孔 ON	m	+		PGAsl
PHI	K	繞錐形方向軸的方向旋轉角度		+		PGAsl
PHU	K	變數的實體元件		+		PGAsl
PL	O	1.B 曲線：節點間隙 2.多項式插補的參數間隔多項式插補長度	s	+		PGAsl
PM	K	每分鐘		+		PGsl
PO	K	多項式插補的多項式係數	s	+		PGAsl
POCKET3	C (T)	正在銑削矩形腔		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
POCKET4	C (T)	正在銑削圓形腔		+		PGAsl
POLF	K	LIFTFAST 回退位置	m	+		PGsl/PGAsl
POLFA	P	含 $AA_ESR_TRIGGER 之單軸的回退起始位置	m	+	+	PGsl
POLFMASK	P	啟用回退之軸不含軸間連接	m	+	-	PGsl
POLFMLIN	P	啟用回退之軸含軸間線性連接	m	+	-	PGsl
POLY	G	多項式插補	m	+		PGAsl
POLYPATH	P	可為 AXIS 或 VECT 軸群組選擇多項式插補	m	+	-	PGAsl
PON	G	沖孔 ON	m	+		PGAsl
PONS	G	插補循環中之沖孔 ON	m	+		PGAsl
POS	K	軸定位		+	+	PGsl
POSA	K	跨單節界線之定位軸		+	+	PGsl
POSM	P	定位刀庫		+	-	FBWsl
POSMT	P	在位置編號將多重刀具置於刀把		+	-	FBWsl
POSP	K	零件中之定位軸（擺動）		+		PGsl
POSRANGE	F	決定軸目前的插補位置設定點是否落在位於預先定義之參考位置的視窗中		+	+	FBSYsl
POT	F	正方形（算術功能）		+	+	PGAsl
PR	K	每轉數		+		PGsl
PREPRO	PA	定義具有準備的子程式		+		PGAsl
PRESETON	P	實際值設定，含參考狀態的遺失		+	+	PGAsl
PRESETONS	P	實際值設定，含參考狀態的遺失		+	+	PGAsl
PRIO	K	設定中斷處理優先順序之關鍵字		+		PGAsl
PRLOC	K	僅於本機變更後在重置時將變數初始化		+		PGAsl
PROC	K	程式中的第一個操作		+		PGAsl
PROTA	P	要求重新計算碰撞模型		+		PGAsl
PROTD	F	計算兩個保護區間的距離		+		PGAsl
PROTS	P	設定保護區狀態		+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
PSI	K	錐形缺口角度		+		PGAsl
PTP	G	點對點動作（PTP 移動）	m	+		PGAsl
PTPG0	G	點對點動作僅搭配 G0，否則搭配路徑動作 CP	m	+		PGAsl
PTPWOC	G	點對點動作不含由方向變更所產生的補正移動	m	+		PGAsl
PUNCHACC	P	切片的移動相關加速度		+	-	PGAsl
PUTFTOC	P	平行加工之刀具微調偏移量		+	-	PGAsl
PUTFTOCF	P	刀具微調偏移量視以 FCTDEF 定義的平行加工之函數而定		+	-	PGAsl
PW	O	B 曲線，重量點	s	+		PGAsl
QU	K	快速附加 （輔助）函數輸出		+		PGsl
R...	O	算術參數亦為可設定之位址識別碼，含數字副檔名		+		PGAsl
RAC	K	絕對非模態、軸專屬半徑程式設計	s	+		PGsl
RDISABLE	P	讀入停用		-	+	FBSYsl
READ	P	讀取指定檔案中一或數行，並將讀取之資訊儲存到陣列中		+	-	PGAsl
REAL	K	資料類型：含正負號浮點數（實數）		+		PGAsl
REDEF	K	Redefinition of system variables, user variables, and NC language commands		+		PGAsl
RELEASE	P	鬆開機械軸以便進行軸交換		+	+	PGAsl
REP	K	將陣列中具相同數值之所有元素初始化的關鍵字		+		PGAsl
REPEAT	K	重覆程式迴圈		+		PGAsl
REPEATB	K	重覆程式指令列		+		PGAsl
REPOSA	G	以所有軸進行線性重新定位	s	+		PGAsl
REPOSH	G	使用半圓重新定位	s	+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
REPOSHA	G	以所有軸進行重新定位；幾何座標軸以半圓表示	s	+		PGAsl
REPOSL	G	線性重新定位	s	+		PGAsl
REPOSQ	G	在象限中重新定位	s	+		PGAsl
REPOSQA	G	以所有軸進行線性重新定位，幾何座標軸以象限表示	s	+		PGAsl
RESETMON	P	用於設定點啟用的語言指令		+	-	FBWsl
RET	P	子程式結尾		+	+	PGAsl
RETB	P	子程式結尾		+	+	PGAsl
RIC	K	相對非模態、軸專屬半徑程式設計	s	+		PGsl
RINDEX	F	定義輸入字串中字元的索引值		+	-	PGAsl
RMB	G	重新定位至單節起點	m	+		PGAsl
RMBBL	G	重新定位至單節起點	s	+		PGAsl
RME	G	重新定位至單節結尾	m	+		PGAsl
RMEBL	G	重新定位至單節結尾	s	+		PGAsl
RMI [6]	G	重新定位至中斷點	m	+		PGAsl
RMIBL [6]	G	重新定位至中斷點	s	+		PGAsl
RMN	G	重新定位至最近的路徑點	m	+		PGAsl
RMNBL	G	重新定位至最近的路徑點	s	+		PGAsl
RND	O	磨圓輪廓轉角	s	+		PGsl
RNDM	O	模態倒圓角	m	+		PGsl
ROT	G	可程式設計的旋轉	s	+		PGsl
ROTS	G	具立體角的可程式設計框架旋轉	s	+		PGsl
ROUND	F	小數點位數的四捨五入		+	+	PGAsl
ROUNDUP	F	輸入值的向上捨去		+	+	PGAsl
RP	O	極半徑	m/s	+		PGsl
RPL	O	平面旋轉	s	+		PGsl
RT	K	存取框架資料之參數：旋轉		+		PGAsl
RTLIOF	G	G0 不含線性插補（單軸插補）	m	+		PGsl
RTLION [6]	G	用線性插補進行 G0	m	+		PGsl

操作 S ... Z

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
S	A	主軸速度 （包含 G4、G96/G961 不同意義）	m/s	+	+	PGsl
SAVE	PA	當呼叫子程式時，用來儲存資訊的屬性		+		PGAsl
SBLOF	P	抑制單一單節		+	-	PGAsl
SBLON	P	撤回單一單節的抑制		+	-	PGAsl
SC	K	存取框架資料之參數：刻度		+		PGAsl
SCALE	G	可程式設計的比例	s	+		PGsl
SCC	K	選擇指派至 G96/G961/G962 之橫向軸。軸識別碼的形式可為幾何、通道或機械軸。		+		PGsl
SCPARA	K	程式伺服參數設定		+	+	PGAsl
SD	A	曲線度數	s	+		PGAsl
SET	K	將陣列中含列出之數值的所有元素初始化的關鍵字		+		PGAsl
SETAL	P	設定警報		+	+	PGAsl
SETDNO	F	指派刀具 (T) 的刀刃 (CE) 之 D 編號		+	-	PGAsl
SETINT	K	定義於 NC 輸入存在時，應啟動哪一個中斷常式		+		PGAsl
SETM	P	專用通道中，標記的設定		+	+	PGAsl
SETMS	P	將機械資料中定義的主動主軸重置		+	-	PGsl
SETMS(n)	P	將主軸 n 設為主動主軸		+		PGsl
SETMTH	P	設定主刀把號碼		+	-	FBWsl
SETPIECE	P	為指派至該主軸之所有刀具設定工件編號		+	-	FBWsl
SETTA	P	由磨損群組啟動刀具		+	-	FBWsl
SETTCOR	F	刀具元件的修改，將所有補充條件列入考量		+	-	PGAsl
SETTIA	P	由磨損群組停用刀具		+	-	FBWsl
SF	A	螺紋切削用起點偏移	m	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
SIN	F	正弦（三角函數）		+	+	PGAsl
SIRELAY	F	啟動以 SIRELIN、SIRELOUT 與 SIRELTIME 參數化的安全函數		-	+	FBSIsl
SIRELIN	P	函數單節的初始化輸入變數		+	-	FBSIsl
SIRELOUT	P	函數單節的初始化輸出變數		+	-	FBSIsl
SIRELTIME	P	函數單節的初始化計時器		+	-	FBSIsl
SLOT1	C (T)	縱向槽		+		PGAsl
SLOT2	C (T)	圓弧槽		+		PGAsl
SOFT	G	軟路徑加速度	m	+		PGAsl
SOFTA	P	啟動程式設計之軸的震動抑制軸加速		+	-	PGAsl
SON	G	切片 ON	m	+		PGAsl
SONS	G	插補循環中之切片 ON	m	+		PGAsl
SPATH [6]	G	FGROUP 軸的路徑參考為圓弧長	m	+		PGAsl
SPCOF	P	將主動主軸或主軸由定位控制切換至速率控制	m	+	-	PGsl
SPCON	P	將主動主軸或主軸由速率控制切換至定位控制	m	+	-	PGAsl
SPI	F	將主軸號碼轉換為軸識別碼		+	-	PGAsl
SPIF1 [6]	G	快速 NC 輸入 / 輸出給沖孔 / 切片位元組 1 使用	m	+		FB2sl (N4)
SPIF2	G	快速 NC 輸入 / 輸出給沖孔 / 切片位元組 2 使用	m	+		FB2sl (N4)
SPLINEPATH	P	定義曲線群組		+	-	PGAsl
SPN	A	各單節路徑區段數量	s	+		PGAsl
SPOF [6]	G	衝擊 OFF， 切片，沖孔 OFF	m	+		PGAsl
SPOS	K	主軸位置	m	+	+	PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
SPOSA	K	跨單節界線主軸位置	m	+		PGsl
SPP	A	路徑區段長度	m	+		PGAsl
SPRINT	F	傳回格式化的輸入字串		+		PGAsl
sqrt	F	平方根 （算術功能）		+	+	PGAsl
SR	A	同步動作的振盪回退路徑	s	+		PGsl
SRA	K	同步動作含外部軸向輸入之振盪回退路徑	m	+		PGsl
ST	A	同步動作之振盪火花消除時間	s	+		PGsl
STA	K	同步動作之振盪火花消除時間軸	m	+		PGsl
START	P	由目前的程式同時在數個通道中啟動選取之程式		+	-	PGAsl
STARTFIFO [6]	G	執行；同時填充前置處理記憶體	m	+		PGAsl
STAT		接點位置	s	+		PGAsl
STOLF	K	G0 允差係數	m	+		PGAsl
STOPFIFO	G	停止加工；填充前置處理記憶體直到偵測到 STARTFIFO、前置處理記憶體完整或程式結尾	m	+		PGAsl
STOPRE	P	前置處理停止，直到所有預置單節已在主程式中執行		+	-	PGAsl
STOPREOF	P	撤回前置處理停止		-	+	FBSYsl
STRING	K	資料類型：字元字串		+		PGAsl
STRINGIS	F	檢查 NC 語言及 NC 循環名稱、使用者變數、巨集及屬於本指令之標籤名稱目前的範圍，以確定其係存在、有效、已定義或啟用。		+	-	PGAsl
STRLEN	F	定義字串長度		+	-	PGAsl
SUBSTR	F	定義輸入字串中字元的索引值		+	-	PGAsl
SUPA	G	抑制目前的零點偏移量，包括程式設計之偏移量、系統框架、手輪偏移 (DRF)、外部零點偏移量及重疊移動	s	+		PGsl
SVC	K	刀具切削率	m	+		PGsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
SYNFCT	P	評估將多項式作為動作同步動作中條件的函數		-	+	FBSYsl
SYNR	K	同步讀取變數，例如，在執行時		+		PGAsl
SYNRW	K	同步讀取和寫入變數，例如，在執行時		+		PGAsl
SYNW	K	同步寫入變數，例如，在執行時		+		PGAsl
T	A	呼叫刀具 （僅於有在機械資料中指定時改變；否則須使用 M6 指令）		+		PGsl
TAN	F	正切（三角函數）		+	+	PGAsl
TANG	P	切線控制：定義耦合		+	-	PGAsl
TANGDEL	P	切線控制：刪除耦合		+	-	PGAsl
TANGOF	P	切線控制：停用耦合		+	-	PGAsl
TANGON	P	切線控制：啟動耦合		+	-	PGAsl
TCA (828D: _TCA)	P	刀具選擇 / 換刀，不考慮刀具狀態		+	-	FBWsl
TCARR	A	請求刀把（編號「m」）		+		PGAsl
TCI	P	從緩衝器，將刀具載入至刀庫		+	-	FBWsl
TCOABS[6]	G	從目前的刀具方向決定刀長元件	m	+		PGAsl
TCOFR	G	從主動框架的方向決定刀長元件	m	+		PGAsl
TCOFRX	G	選擇刀具及在 X 方向之刀具點來判定啟用中框架的刀具方向	m	+		PGAsl
TCOFRY	G	選擇刀具及在 Y 方向之刀具點來判定啟用中框架的刀具方向	m	+		PGAsl
TCOFRZ	G	選擇刀具及在 Z 方向之刀具點來判定啟用中框架的刀具方向	m	+		PGAsl
THETA	A	旋轉角度	s	+		PGAsl
TILT	A	傾斜角度	m	+		PGAsl
TLIFT	P	切線控制：啟動中間單節產生		+	-	PGAsl
TML	P	利用刀庫位置編號選擇刀具		+	-	FBWsl
TMOF	P	取消選擇刀具監控		+	-	PGAsl
TMON	P	啟用刀具監控		+	-	PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
TO	K	指派 FOR 計數迴圈內之結束值		+		PGAsl
TOFF	K	刀長元件之方向上的刀長偏移，與索引值中指定之幾何軸平行。	m	+		PGsl
TOFFL	K	刀長元件 L1、L2 或 L3 之方向上的刀長偏移	m	+		PGsl
TOFFOF	P	停用線上刀具偏移		+	-	PGAsl
TOFFON	P	啟用線上刀長偏移		+	-	PGAsl
TOFFR	A	刀具半徑偏移量	m	+		PGsl
TOFRAME	G	將框架平行於刀具方向旋轉，對齊 WCS 的 Z 軸	m	+		PGsl
TOFRAMEX	G	將框架平行於刀具方向旋轉，對齊 WCS 的 X 軸	m	+		PGsl
TOFRAMEY	G	將框架平行於刀具方向旋轉，對齊 WCS 的 Y 軸	m	+		PGsl
TOFRAMEZ	G	與 TOFRAME 相同	m	+		PGsl
TOLOWER	F	將字串字母轉換成小寫		+	-	PGAsl
TOOLENV	F	儲存目前狀態，這對於評估儲存在記憶體中的刀具資料非常重要		+	-	PGAsl
TOOLGNT	F	決定刀具群組的刀具數量		+	-	FBWsl
TOOLGT	F	由刀具群組決定刀具 T 編號		+	-	FBWsl
TOROT	G	將框架平行於刀具方向旋轉，對齊 WCS 的 Z 軸	m	+		PGsl
TOROTOF [6]	G	框架在刀具方向上之旋轉 OFF	m	+		PGsl
TOROTX	G	將框架平行於刀具方向旋轉，對齊 WCS 的 X 軸	m	+		PGsl
TOROTY	G	將框架平行於刀具方向旋轉，對齊 WCS 的 Y 軸	m	+		PGsl
TOROTZ	G	與 TOROT 相同	m	+		PGsl
TOUPPER	F	將字串字母轉換成大寫		+	-	PGAsl
TOWBCS	G	基本座標系統 (BCS) 中之磨損值	m	+		PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
TOWKCS	G	刀頭座標系統之磨損值，供動力學轉換之用（與機械座標系統不同點在於刀具旋轉）	m	+		PGAsl
TOWMCS	G	機械座標系統中之磨耗值	m	+		PGAsl
TOWSTD [6]	G	刀長偏移的初始設定值	m	+		PGAsl
TOWTCS	G	刀具座標系統中之磨損值（位於刀把之刀把參考點 T）	m	+		PGAsl
TOWWCS	G	工件座標系統中之磨損值	m	+		PGAsl
TR	K	框架變數的偏移元件		+		PGAsl
TRAANG	P	轉換傾斜軸		+	-	PGAsl
TRACON	P	串連式轉換		+	-	PGAsl
TRACYL	P	圓柱：柱面轉換		+	-	PGAsl
TRAFOOF	P	在通道中停用生效轉換		+	-	PGAsl
TRAFOON	P	啟動動態鏈定義的轉換		+	-	PGAsl
TRAILOF	P	非同步耦合動作 OFF		+	+	PGAsl
TRAILON	P	非同步耦合動作 ON		+	+	PGAsl
TRANS	G	絕對的可程式設計工件偏移量	s	+		PGsl
TRANSMIT	P	極點轉換（平面加工）		+	-	PGAsl
TRAORI	P	4 軸，5 軸轉換，一般轉換		+	-	PGAsl
TRUE	K	邏輯常數：真		+		PGAsl
TRUNC	F	小數點位數的四捨五入		+	+	PGAsl
TU		軸角度	s	+		PGAsl
TURN	A	螺旋旋轉數	s	+		PGsl
ULI	K	變數的上臨界值		+		PGAsl
UNLOCK	P	利用 ID 啟用同步動作（繼續技術循環）		-	+	FBSYsl
UNTIL	K	終止 REPEAT 迴圈之條件		+		PGAsl
UPATH	G	FGROUP 軸之路徑參照為曲線參數	m	+		PGAsl
VAR	K	關鍵字：參數傳輸類型		+		PGAsl
VELOLIM	K	最大軸向速率之減少	m	+		PGAsl
VELOLIMA	K	跟隨軸之最大軸向速率之減少或增加	m	+	+	PGAsl

操作	類型 [1]	含義	W [2]	TP [3]	SA [4]	說明，請參閱 [5]
[1] [2] [3] [4] [5] 供說明之用，請參閱圖例 (頁 963)。						
WAITC	P	等待軸 / 主軸所需滿足的耦合單節更改條件		+	-	PGAsl
WAITE	P	等待另一通道中的程式結束。		+	-	PGAsl
WAITENC	P	等待同步或還原的軸位置		+	-	PGAsl
WAITM	P	等待指定通道中之標記；以精確停止終止先前單節。		+	-	PGAsl
WAITMC	P	等待指定通道中之標記；僅在另一通道尚未抵達標記時進行精確停止。		+	-	PGAsl
WAITP	P	等待定位軸移動結束		+	-	PGsl
WAITS	P	等待有待達到的主軸位置		+	-	PGsl
WALCS0 [6]	G	工件座標系統工作區限制取消選擇	m	+	-	PGsl
WALCS1	G	WCS 工作區限制 1 啟用	m	+	-	PGsl
WALCS2	G	WCS 工作區限制 2 啟用	m	+	-	PGsl
WALCS3	G	WCS 工作區限制 3 啟用	m	+	-	PGsl
WALCS4	G	WCS 工作區限制 4 啟用	m	+	-	PGsl
WALCS5	G	WCS 工作區限制 5 啟用	m	+	-	PGsl
WALCS6	G	WCS 工作區限制 6 啟用	m	+	-	PGsl
WALCS7	G	WCS 工作區限制 7 啟用	m	+	-	PGsl
WALCS8	G	WCS 工作區限制 8 啟用	m	+	-	PGsl
WALCS9	G	WCS 工作區限制 9 啟用	m	+	-	PGsl
WALCS10	G	WCS 工作區限制 10 啟用	m	+	-	PGsl
WALIMOF	G	BCS 工作區限制 OFF	m	+	-	PGsl
WALIMON [6]	G	BCS 工作區限制 ON	m	+	-	PGsl
WHEN	K	條件滿足時，就可以執行動作一次。		-	+	FBSYsl
WHENEVER	K	當條件滿足時，將在每一插補器循環中執行動作。		-	+	FBSYsl
WHILE	K	WHILE 程式迴圈之起點		+		PGAsl
WRITE	P	將文字寫入檔案系統。 將單節附加至指定檔案之結束。		+	-	PGAsl
WRTPR	P	將字串寫入 OPI 變數中		+	-	PGsl
X	A	軸名稱	m/s	+	+	PGsl

操作	類型 1)	含義	W 2)	TP 3)	SA 4)	說明，請參閱 5)
1) 2) 3) 4) 5) 供說明之用，請參閱圖例 (頁 963)。						
XOR	O	邏輯互斥 OR		+		PGAsl
Y	A	軸名稱	m/s	+	+	PGsl
Z	A	軸名稱	m/s	+	+	PGsl

圖例

1) 操作類型：

A 位址

已指定值的識別碼（例如 OVR=10）。也有部分開啟或關閉功能的位址並未指定值（例如 CPLON 及 CPLOF）。

C (A) AST 循環

預先定義的 NC 程式，使用 AST（自動伺服調整）自動後置最佳化（調整）。參數用於調整特定的最佳化狀況；這些參數於呼叫時傳送。

C 測量循環

(M) 預先定義的 NC 程式中，已程式設計特殊一般有效的測量操作，例如決定圓柱工件的內直徑。參數用於調整特定的測量狀況；這些參數於呼叫時傳送。

C (T) 技術循環

預先定義的 NC 程式中，已程式設計特殊一般有效的加工操作，例如螺紋攻牙或銑削凹槽。可透過呼叫期間傳輸至循環的參數針對特定機台狀況進行調整。

F 預先定義的功能（提供傳回值）

呼叫預先定義功能可以是一個算式的運算元。

G G 指令

G 指令劃分在 G 群組中。每個單節中僅能程式設計群組中的一個 G 指令。G 指令可為模態（直到同個群組中的另一個功能將其取消為止）或僅程式設計（非模態）的單節有效。

K 關鍵字

定義單節語法的識別碼。並未指定值至關鍵字，無法使用關鍵字開啟 / 關閉 NC 功能。

範例：控制結構（IF、ELSE、ENDIF、WHEN、...）、程式執行（GOTOB、GOTO、RET ...）

O 運算子

數學、比較或邏輯運算的運算子

P 預先定義的步驟（未提供傳回值）

PA 程式屬性

程式屬性位於子程式定義行的結尾：

```
PROC <程式名稱>(...) <程式屬性>
```

程式屬性決定子程式執行期間的行為。

2) 操作的效果：

m 模態

s 非模態

3) 工件程式的可程式性：

+ 可程式設計

- 不可程式設計

M 僅可由機台製造商進行程式設計

4) 同步動作的可程式性：

+ 可程式設計

- 不可程式設計

T 僅於技術循環進行程式設計

5) 參考包含有詳細操作說明的文件：

PGsl 程式設計手冊，基礎

PGAsl 程式設計手冊，工作計畫

BNMsl 程式設計手冊測量循環

BHDsl 車削操作手冊

BHFsl 銑削操作手冊

FB1sl ( ) 功能手冊，基本功能（具有相對應函數的字母與數字之縮寫，函數以括弧說明）

FB2sl ( ) 功能手冊，延伸功能（具有相對應函數的字母與數字縮寫，函數以括弧說明）

FB3sl ( ) 功能手冊，特殊功能（具有相對應函數的字母與數字縮寫，函數以括弧說明）

FBSIsl 功能手冊，安全整合

FBSYsl 功能手冊，同步動作

FBWsl 功能手冊，刀具管理

6) 程式開始之預設設定（如程式中未程式設計，為該控制項目之出廠設定）。

圖像 21-1 操作表中的註釋含義

21.2 操作：SINUMERIK 828D 的支援度

說明

循環

若是循環係依據下列的選項需要授權的話，選項要標記為「選項」：

- 延伸技術功能（文件編號：6FC5800-0AP58-0YB0）
- 測量循環（文件編號：6FC5800-0AP28-0YB0）
- 測量動力學（文件編號：6FC5800-0AP18-0YB0）
- SINUMERIK 研磨進階版（文件編號：6FC5800-0AS35-0YB0）

若是循環由於選項「延伸技術功能」的結果，只包含部份的功能，則無標記。

21.2.1 控制器版本銑削 / 車削

操作 A ... C

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
:	●	●	●	●	●	●	●	●
*	●	●	●	●	●	●	●	●
+	●	●	●	●	●	●	●	●
-	●	●	●	●	●	●	●	●
<	●	●	●	●	●	●	●	●
<<	●	●	●	●	●	●	●	●
<=	●	●	●	●	●	●	●	●
=	●	●	●	●	●	●	●	●
>=	●	●	●	●	●	●	●	●
/	●	●	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●	●	●
A	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
A2	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-
A4	-	-	-	-	-	-	-	-
A5	-	-	-	-	-	-	-	-
A6	-	-	-	-	-	-	-	-
A7	-	-	-	-	-	-	-	-
ABS	●	●	●	●	●	●	●	●
AC	●	●	●	●	●	●	●	●
ACC	●	●	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●	●	●
ACN	●	●	●	●	●	●	●	●
ACOS	●	●	●	●	●	●	●	●
ACP	●	●	●	●	●	●	●	●
ACTBLOCNO	●	●	●	●	●	●	●	●
ADDFRAME	●	●	●	●	●	●	●	●
ADIS	●	●	●	●	●	●	●	●
ADISPOS	●	●	●	●	●	●	●	●
ADISPOSA	●	●	●	●	●	●	●	●
ALF	●	●	●	●	●	●	●	●
AMIRROR	●	●	●	●	●	●	●	●
AND	●	●	●	●	●	●	●	●
ANG	●	●	●	●	●	●	●	●
AP	●	●	●	●	●	●	●	●
APR	●	●	●	●	●	●	●	●
APRB	●	●	●	●	●	●	●	●
APRP	●	●	●	●	●	●	●	●
APW	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
APWB	●	●	●	●	●	●	●	●
APWP	●	●	●	●	●	●	●	●
APX	●	●	●	●	●	●	●	●
AR	●	●	●	●	●	●	●	●
AROT	●	●	●	●	●	●	●	●
AROTS	●	●	●	●	●	●	●	●
AS	●	●	●	●	●	●	●	●
ASCALE	●	●	●	●	●	●	●	●
ASIN	●	●	●	●	●	●	●	●
ASPLINE	○	○	○	○	○	○	○	○
ATAN2	●	●	●	●	●	●	●	●
ATOL	●	●	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●	●	●
AX	●	●	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-	-	-
AXIS	●	●	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●	●	●
B	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	-	-	-	-	-	-	通道軸名稱	通道軸名稱
C3	-	-	-	-	-	-	-	-
C4	-	-	-	-	-	-	-	-
C5	-	-	-	-	-	-	-	-
C6	-	-	-	-	-	-	-	-

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
CLEARM	-	-	-	-	-	○	-	●
CLRINT	●	●	●	●	●	●	●	●
CMIRROR	●	●	●	●	●	●	●	●
COARSEA	●	●	●	●	●	●	●	●
COLLPAIR	-	-	-	-	-	-	-	-
COMPCAD	●	-	●	-	●	●	-	-
COMPCURV	●	-	●	-	●	●	-	-
COMPLETE	●	●	●	●	●	●	●	●
COMPOF	●	-	●	-	●	●	-	-
COMPON	●	-	●	-	●	●	-	-
COMPSURF	-	-	○	-	○	○	-	-
CONTDCON	●	●	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●	●	●
CORROF	●	●	●	●	●	●	●	●
CORRTRAFO	-	-	-	-	-	-	-	-
COS	●	●	●	●	●	●	●	●
COUPDEF	-	○	-	○	-	-	○	○
COUPDEL	-	○	-	○	-	-	○	○
COUPOF	-	○	-	○	-	-	○	○
COUPOFS	-	○	-	○	-	-	○	○
COUPON	-	○	-	○	-	-	○	○
COUPONC	-	○	-	○	-	-	○	○
COUPRES	-	○	-	○	-	-	○	○
CP	●	●	●	●	●	●	●	●
CPBC	○	○	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○	○	○

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	○	○	○	○	○	○	○	○

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	-	-	-	-	-	-	-	-

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
CUT2DD	●	●	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●	●	●
CUT2DFD	●	●	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-	-	-
CYCLE60	○	○	●	●	●	●	●	●
CYCLE61	●	●	●	●	●	●	●	●
CYCLE62	●	●	●	●	●	●	●	●
CYCLE63	○	○	●	●	●	●	●	●
CYCLE64	○	○	●	●	●	●	●	●
CYCLE70	○	○	●	●	●	●	●	●
CYCLE72	●	●	●	●	●	●	●	●
CYCLE76	●	●	●	●	●	●	●	●
CYCLE77	●	●	●	●	●	●	●	●
CYCLE78	○	○	●	●	●	●	●	●
CYCLE79	○	○	●	●	●	●	●	●
CYCLE81	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	-	-	-	-	-	-	-	-

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
CYCLE4078	-	-	-	-	-	-	-	-
CYCLE4079	-	-	-	-	-	-	-	-

操作 D ... F

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
D	●	●	●	●	●	●	●	●
D0	●	●	●	●	●	●	●	●
DAC	●	●	●	●	●	●	●	●
DC	●	●	●	●	●	●	●	●
DCI	●	●	●	●	●	●	●	●
DCM	●	●	●	●	●	●	●	●
DCU	●	●	●	●	●	●	●	●
DEF	●	●	●	●	●	●	●	●
DEFINE	●	●	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●	●	●
DELDL	●	●	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●	●	●
DELETE	●	●	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
ENDLOOP	●	●	●	●	●	●	●	●
ENDPROC	●	●	●	●	●	●	●	●
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●
FRCM	●	●	●	●	●	●	●	●
FROM	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
FTOC	●	●	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●	●	●
FXS	●	●	●	●	●	●	●	●
FXST	●	●	●	●	●	●	●	●
FXSW	●	●	●	●	●	●	●	●
FZ	●	●	●	●	●	●	●	●

操作 G ... L

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
G0	●	●	●	●	●	●	●	●
G1	●	●	●	●	●	●	●	●
G2	●	●	●	●	●	●	●	●
G3	●	●	●	●	●	●	●	●
G4	●	●	●	●	●	●	●	●
G5	●	●	●	●	●	●	●	●
G7	●	●	●	●	●	●	●	●
G9	●	●	●	●	●	●	●	●
G17	●	●	●	●	●	●	●	●
G18	●	●	●	●	●	●	●	●
G19	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
G95	●	●	●	●	●	●	●	●
G96	●	●	●	●	●	●	●	●
G97	●	●	●	●	●	●	●	●
G110	●	●	●	●	●	●	●	●
G111	●	●	●	●	●	●	●	●
G112	●	●	●	●	●	●	●	●
G140	●	●	●	●	●	●	●	●
G141	●	●	●	●	●	●	●	●
G142	●	●	●	●	●	●	●	●
G143	●	●	●	●	●	●	●	●
G147	●	●	●	●	●	●	●	●
G148	●	●	●	●	●	●	●	●
G153	●	●	●	●	●	●	●	●
G247	●	●	●	●	●	●	●	●
G248	●	●	●	●	●	●	●	●
G290	●	●	●	●	●	●	●	●
G291	●	●	●	●	●	●	●	●
G331	●	●	●	●	●	●	●	●
G332	●	●	●	●	●	●	●	●
G335	●	●	●	●	●	●	●	●
G336	●	●	●	●	●	●	●	●
G340	●	●	●	●	●	●	●	●
G341	●	●	●	●	●	●	●	●
G347	●	●	●	●	●	●	●	●
G348	●	●	●	●	●	●	●	●
G450	●	●	●	●	●	●	●	●
G451	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
GET	●	●	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●	●	●
GETD	-	-	-	-	-	○	-	●
GETDNO	●	●	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●	●	●
GETSELT	●	●	●	●	●	●	●	●
GETT	●	●	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●	●	●
GETVARAP	●	●	●	●	●	●	●	●
GETVARDFT	●	●	●	●	●	●	●	●
GETVARLIM	●	●	●	●	●	●	●	●
GETVARPHU	●	●	●	●	●	●	●	●
GETVARTYP	●	●	●	●	●	●	●	●
GFRAME0 ... GFRAME100	-	-	-	-	-	-	-	-
GOTO	●	●	●	●	●	●	●	●
GOTOB	●	●	●	●	●	●	●	●
GOTOC	●	●	●	●	●	●	●	●
GOTOF	●	●	●	●	●	●	●	●
GOTOS	●	●	●	●	●	●	●	●
GP	●	●	●	●	●	●	●	●
GWPSOF	●	●	●	●	●	●	●	●
GROUP_ ADDEND	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
IPOBRKA	●	●	●	●	●	●	●	●
IPOENDA	●	●	●	●	●	●	●	●
IPTRLOCK	●	●	●	●	●	●	●	●
IPTRUNLOCK	●	●	●	●	●	●	●	●
IR	●	●	●	●	●	●	●	●
ISAXIS	●	●	●	●	●	●	●	●
ISD	-	-	-	-	-	-	-	-
ISFILE	●	●	●	●	●	●	●	●
ISNUMBER	●	●	●	●	●	●	●	●
ISOCALL	●	●	●	●	●	●	●	●
ISVAR	●	●	●	●	●	●	●	●
J	●	●	●	●	●	●	●	●
J1	●	●	●	●	●	●	●	●
JERKA	●	●	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●	●	●
JR	●	●	●	●	●	●	●	●
K	●	●	●	●	●	●	●	●
K1	●	●	●	●	●	●	●	●
KONT	●	●	●	●	●	●	●	●
KONTC	●	●	●	●	●	●	●	●
KONTT	●	●	●	●	●	●	●	●
KR	●	●	●	●	●	●	●	●
L	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
LEAD 刀具方向 方向多項式	- -	- -	- -	- -	- -	- -	- -	- -
LEADOF	-	-	-	●	-	-	●	●
LEADON	-	-	-	●	-	-	●	●
LENTOAX	●	●	●	●	●	●	●	●
LFOF	●	●	●	●	●	●	●	●
LFON	●	●	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●	●	●
LFWP	●	●	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●	●	●
LIMS	●	●	●	●	●	●	●	●
LLI	●	●	●	●	●	●	●	●
LN	●	●	●	●	●	●	●	●
LOCK	●	●	●	●	●	●	●	●
LONGHOLE	●	●	●	●	●	●	●	●
LOOP	●	●	●	●	●	●	●	●

操作 M ... R

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (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	-	-	-	-	-	-	-	-

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
MEASURE	●	●	●	●	●	●	●	●
MEAW	●	●	●	●	●	●	●	●
MEAWA	-	-	-	-	-	-	-	-
MI	●	●	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●	●	●
MMC	●	●	●	●	●	●	●	●
MOD	●	●	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●	●	●
MOV	●	●	●	●	●	●	●	●
MOVT	●	●	●	●	●	●	●	●
MSG	●	●	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●	●	●
N	●	●	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●	●	●
NC	●	●	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●	●	●
NEWT	●	●	●	●	●	●	●	●
NORM	●	●	●	●	●	●	●	●
NOT	●	●	●	●	●	●	●	●
NPROT	●	●	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-	-	-

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
OF	●	●	●	●	●	●	●	●
OFFN	●	●	●	●	●	●	●	●
OMA1	-	-	-	-	-	-	-	-
OMA2	-	-	-	-	-	-	-	-
OMA3	-	-	-	-	-	-	-	-
OMA4	-	-	-	-	-	-	-	-
OMA5	-	-	-	-	-	-	-	-
或	●	●	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-	-	-
ORIC	-	-	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-	-	-
ORID	-	-	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-	-	-

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
OVR	●	●	●	●	●	●	●	●
OVRA	●	●	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●	●	●
P	●	●	●	●	●	●	●	●
PAROT	●	●	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●	●	●
PCALL	●	●	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-	-	-
PHI	-	-	-	-	-	-	-	-
PHU	●	●	●	●	●	●	●	●
PL	-	-	-	-	-	-	-	-
PM	●	●	●	●	●	●	●	●
PO	-	-	-	-	-	-	-	-
POCKET3	●	●	●	●	●	●	●	●
POCKET4	●	●	●	●	●	●	●	●
POLF	●	●	●	●	●	●	●	●
POLFA	●	●	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●	●	●
POLY	-	-	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-	-	-
PON	-	-	-	-	-	-	-	-
PONS	-	-	-	-	-	-	-	-
POS	●	●	●	●	●	●	●	●
POSA	●	●	●	●	●	●	●	●
POSM	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
REAL	●	●	●	●	●	●	●	●
RELEASE	●	●	●	●	●	●	●	●
REP	●	●	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●	●	●
REPOSH	●	●	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●	●	●
RET	●	●	●	●	●	●	●	●
RETB	●	●	●	●	●	●	●	●
RIC	●	●	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●	●	●
RMB	●	●	●	●	●	●	●	●
RME	●	●	●	●	●	●	●	●
RMI	●	●	●	●	●	●	●	●
RMN	●	●	●	●	●	●	●	●
RND	●	●	●	●	●	●	●	●
RNDM	●	●	●	●	●	●	●	●
ROT	●	●	●	●	●	●	●	●
ROTS	●	●	●	●	●	●	●	●
ROUND	●	●	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●	●	●
RP	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
RPL	●	●	●	●	●	●	●	●
RT	●	●	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●	●	●
RTLION	●	●	●	●	●	●	●	●

操作 S ... Z

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
S	●	●	●	●	●	●	●	●
SAVE	●	●	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●	●	●
SBLON	●	●	●	●	●	●	●	●
SC	●	●	●	●	●	●	●	●
SCALE	●	●	●	●	●	●	●	●
SCC	●	●	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●	●	●
SD	○	○	○	○	○	○	○	○
SET	●	●	●	●	●	●	●	●
SETAL	●	●	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●	●	●
SETINT	●	●	●	●	●	●	●	●
SETM	-	-	-	-	-	○	-	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	-	-	-	-	-	-	-	-

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
SPOF	-	-	-	-	-	-	-	-
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●
TOFFOF	●	●	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	-	-	-	-	-	-	-	-

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
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	●	●	●	●	●	●	●	●

操作	SINUMERIK 828D							
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW 銑削 匯出 (me42)	SW24x(5) CNC-SW 車削 匯出 (te42)	SW26x(3) CNC-SW 銑削 匯出 (me62)	SW26x(3) CNC-SW 車削 匯出 (te62)	SW28x(2) CNC-SW 銑削 匯出 (me821)	SW28x(1) CNC-SW 銑削 進階匯出 (me822)	SW28x(2) CNC-SW 車削 匯出 (te821)	SW28x(1) CNC-SW 車削 進階匯出 (te822)
WALCS4	●	●	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●	●	●
WALCS8	●	●	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●	●	●
WHEN	●	●	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●	●	●
WHILE	●	●	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●	●	●
WRITE	●	●	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●	●	●
X	●	●	●	●	●	●	●	●
XOR	●	●	●	●	●	●	●	●
Y	●	●	●	●	●	●	●	●
Z	●	●	●	●	●	●	●	●

21.2.2 控制器版本研磨

操作 A ... C

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
:	●	●	●	●	●	●
*	●	●	●	●	●	●
+	●	●	●	●	●	●
-	●	●	●	●	●	●
<	●	●	●	●	●	●
<<	●	●	●	●	●	●
<=	●	●	●	●	●	●
=	●	●	●	●	●	●
>=	●	●	●	●	●	●
/	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●
A	●	●	●	●	●	●
A2	-	-	-	-	-	-
A3	-	-	-	-	-	-
A4	-	-	-	-	-	-
A5	-	-	-	-	-	-
A6	-	-	-	-	-	-
A7	-	-	-	-	-	-
ABS	●	●	●	●	●	●
AC	●	●	●	●	●	●
ACC	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●
ACN	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
ATOL	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●
AX	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-
AXIS	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●
B	●	●	●	●	●	●
B2	-	-	-	-	-	-
B3	-	-	-	-	-	-
B4	-	-	-	-	-	-
B5	-	-	-	-	-	-
B6	-	-	-	-	-	-
B7	-	-	-	-	-	-
B_AND	●	●	●	●	●	●
B_OR	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
BLOCK	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●
BNAT	○	○	○	○	○	○
BOOL	●	●	●	●	●	●
BOUND	●	●	●	●	●	●
BRISK	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○
BTAN	○	○	○	○	○	○
C	●	●	●	●	●	●
C2	-	-	通道軸名稱	-	-	-
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
C6	-	-	-	-	-	-
C7	-	-	-	-	-	-
CAC	●	●	●	●	●	●
CACN	●	●	●	●	●	●
CACP	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●
CALCPOSI	●	●	●	●	●	●
CALL	●	●	●	●	●	●
CALLPATH	●	●	●	●	●	●
CANCEL	●	●	●	●	●	●
CASE	●	●	●	●	●	●
CDC	●	●	●	●	●	●
CDOF	-	-	-	-	-	-

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	○	○	○	○	○	○

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
CPLNUM	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○
CPLON	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○
CPOF	○	○	○	○	○	○
CPON	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●
CPRES	○	○	○	○	○	○
CPROT	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	-	-	-	-	-	-
CTABPOLID	-	-	-	-	-	-

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
CYCLE751	●	●	●	●	●	●
CYCLE752	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●
CYCLE758	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●
CYCLE800	○	○	○	○	○	○
CYCLE801	-	-	-	-	-	-
CYCLE802	-	-	-	-	-	-
CYCLE830	-	-	-	-	-	-
CYCLE832	●	●	●	●	●	●
CYCLE840	-	-	-	-	-	-
CYCLE899	-	-	-	-	-	-
CYCLE930	-	-	-	-	-	-
CYCLE940	-	-	-	-	-	-
CYCLE951	-	-	-	-	-	-
CYCLE952	-	-	-	-	-	-
CYCLE961	-	-	-	-	-	-
CYCLE971	-	-	-	-	-	-
CYCLE973	-	-	-	-	-	-
CYCLE974	-	-	-	-	-	-
CYCLE976	-	-	-	-	-	-
CYCLE977	-	-	-	-	-	-
CYCLE978	-	-	-	-	-	-

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
CYCLE979	-	-	-	-	-	-
CYCLE982	-	-	-	-	-	-
CYCLE994	-	-	-	-	-	-
CYCLE995	-	-	-	-	-	-
CYCLE996	-	-	-	-	-	-
CYCLE997	-	-	-	-	-	-
CYCLE998	-	-	-	-	-	-
CYCLE4071	●	●	●	●	●	●
CYCLE4072	●	●	●	●	●	●
CYCLE4073	●	●	●	●	●	●
CYCLE4074	●	●	●	●	●	●
CYCLE4075	●	●	●	●	●	●
CYCLE4077	●	●	●	●	●	●
CYCLE4078	●	●	●	●	●	●
CYCLE4079	●	●	●	●	●	●

操作 D ... F

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
D	●	●	●	●	●	●
D0	●	●	●	●	●	●
DAC	●	●	●	●	●	●
DC	●	●	●	●	●	●
DCI	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
DIC	●	●	●	●	●	●
DILF	●	●	●	●	●	●
DISABLE	●	●	●	●	●	●
DISC	●	●	●	●	●	●
DISCL	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●
DISPR	●	●	●	●	●	●
DISR	●	●	●	●	●	●
DISRP	●	●	●	●	●	●
DITE	●	●	●	●	●	●
DITS	●	●	●	●	●	●
DIV	●	●	●	●	●	●
DL	-	-	-	-	-	-
DO	●	●	●	●	●	●
DRFOF	●	●	●	●	●	●
DRIVE	●	●	●	●	●	●
DRIVEA	●	●	●	●	●	●
DYNFINISH	●	●	●	●	●	●
DYNNORM	●	●	●	●	●	●
DYNPOS	●	●	●	●	●	●
DYNROUGH	●	●	●	●	●	●
DYNSEMIFIN	●	●	●	●	●	●
DZERO	●	●	●	●	●	●
EAUTO	○	○	○	○	○	○
EGDEF	○	○	○	○	○	○
EGDEL	○	○	○	○	○	○

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●
EXTERN	●	●	●	●	●	●
EXTOPEN	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	○	○	○	○	○	○

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
FOCON	○	○	○	○	○	○
FOR	●	●	●	●	●	●
FP	●	●	●	●	●	●
FPO	-	-	-	-	-	-
FPR	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●
FRAME	●	●	●	●	●	●
FRC	●	●	●	●	●	●
FRCM	●	●	●	●	●	●
FROM	●	●	●	●	●	●
FTOC	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●
FXS	●	●	●	●	●	●
FXST	●	●	●	●	●	●
FXSW	●	●	●	●	●	●
FZ	●	●	●	●	●	●

操作 G ... L

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
G0	●	●	●	●	●	●
G1	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
G2	●	●	●	●	●	●
G3	●	●	●	●	●	●
G4	●	●	●	●	●	●
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
G332	●	●	●	●	●	●
G335	●	●	●	●	●	●
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	-	-	-	-	-	-

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	< 50	< 100	< 100	< 50	< 100	< 100
GOTO	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●
INIRE	●	●	●	●	●	●
INICF	●	●	●	●	●	●
INIT	-	-	●	-	-	●
INITIAL						

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
KONTT	●	●	●	●	●	●
KR	●	●	●	●	●	●
L	●	●	●	●	●	●
LEAD 刀具方向 方向多項式	 - -	 - -	 - -	 - -	 - -	 - -
LEADOF	-	-	-	-	-	-
LEADON	-	-	-	-	-	-
LENTOAX	●	●	●	●	●	●
LFOF	●	●	●	●	●	●
LFON	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●
LFWP	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●
LIMS	●	●	●	●	●	●
LLI	●	●	●	●	●	●
LN	●	●	●	●	●	●
LOCK	●	●	●	●	●	●
LONGHOLE	-	-	-	-	-	-
LOOP	●	●	●	●	●	●

操作 M ... R

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	-	○	○	-	○	○
量測	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
MEAW	●	●	●	●	●	●
MEAWA	-	○	○	-	○	○
MI	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●
MMC	●	●	●	●	●	●
MOD	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●
MOV	●	●	●	●	●	●
MOVT	●	●	●	●	●	●
MSG	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●
N	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●
NCK	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●
NEWT	-	-	-	-	-	-
NORM	●	●	●	●	●	●
NOT	●	●	●	●	●	●
NPROT	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-
OF	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
OFFN	●	●	●	●	●	●
OMA1	-	-	-	-	-	-
OMA2	-	-	-	-	-	-
OMA3	-	-	-	-	-	-
OMA4	-	-	-	-	-	-
OMA5	-	-	-	-	-	-
或	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-
ORIC	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-
ORID	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
OVRA	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●
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	-	-	-	-	-	-

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
RELEASE	●	●	●	●	●	●
REP	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●
REPOSH	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●
RET	●	●	●	●	●	●
RETB	●	●	●	●	●	●
RIC	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●
RMB	●	●	●	●	●	●
RME	●	●	●	●	●	●
RMI	●	●	●	●	●	●
RMN	●	●	●	●	●	●
RND	●	●	●	●	●	●
RNDM	●	●	●	●	●	●
ROT	●	●	●	●	●	●
ROTS	●	●	●	●	●	●
ROUND	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●
RP	●	●	●	●	●	●
RPL	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
RT	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●
RTLION	●	●	●	●	●	●

操作 S ... Z

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
S	●	●	●	●	●	●
SAVE	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●
SBLON	●	●	●	●	●	●
SC	●	●	●	●	●	●
SCALE	●	●	●	●	●	●
SCC	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●
SD	○	○	○	○	○	○
SET	●	●	●	●	●	●
SETAL	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●
SETINT	●	●	●	●	●	●
SETM	-	-	●	-	-	●
SETMS	●	●	●	●	●	●
SETMS(n)	●	●	●	●	●	●
SETMTH	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
SETPIECE	●	●	●	●	●	●
SETTA	●	●	●	●	●	●
SETTCOR	●	●	●	●	●	●
SETTIA	●	●	●	●	●	●
SF	●	●	●	●	●	●
SIN	●	●	●	●	●	●
SIRELAY	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-
SLOT1	-	-	-	-	-	-
SLOT2	-	-	-	-	-	-
SOFT	●	●	●	●	●	●
SOFTA	●	●	●	●	●	●
SON	-	-	-	-	-	-
SONS	-	-	-	-	-	-
SPATH	●	●	●	●	●	●
SPCOF	●	●	●	●	●	●
SPCON	●	●	●	●	●	●
SPI	●	●	●	●	●	●
SPIF1	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○
SPN	-	-	-	-	-	-
SPOF	-	-	-	-	-	-
SPOS	●	●	●	●	●	●
SPOSA	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
TANG	○	○	○	○	○	○
TANGDEL	○	○	○	○	○	○
TANGOF	○	○	○	○	○	○
TANGON	○	○	○	○	○	○
TCA （828D：_TCA）	●	●	●	●	●	●
TCARR	●	●	●	●	●	●
TCI	●	●	●	●	●	●
TCOABS	●	●	●	●	●	●
TCOFR	●	●	●	●	●	●
TCOFRX	●	●	●	●	●	●
TCOFRY	●	●	●	●	●	●
TCOFRZ	●	●	●	●	●	●
THETA	-	-	-	-	-	-
TILT	-	-	-	-	-	-
TLIFT	○	○	○	○	○	○
TML	●	●	●	●	●	●
TMOF	●	●	●	●	●	●
TMON	●	●	●	●	●	●
TO	●	●	●	●	●	●
TOFF	●	●	●	●	●	●
TOFFL	●	●	●	●	●	●
TOFFOF	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●
TOFFR	●	●	●	●	●	●
TOFRAME	●	●	●	●	●	●
TOFRAMEX	●	●	●	●	●	●
TOFRAMEY	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	○	○	○	○	○	○

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
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	●	●	●	●	●	●

操作	SINUMERIK 828D					
● 標準 ○ 選配 - 不提供	SW24x(5) CNC-SW G-Tech 匯出 (gce42)	SW26x(3) CNC-SW G-Tech 匯出 (gce62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gce82)	SW24x(5) CNC-SW G-Tech 匯出 (gse42)	SW26x(3) CNC-SW G-Tech 匯出 (gse62)	SW28x(1) CNC-SW G-Tech 進階匯出 (gse82)
WALCS8	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●
WHEN	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●
WHILE	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●
WRITE	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●
X	●	●	●	●	●	●
XOR	●	●	●	●	●	●
Y	●	●	●	●	●	●
Z	●	●	●	●	●	●

21.3 目前設定在 HMI 中的語言

下表列出使用者介面提供的所有語言。

可以使用下列的系統變數，在工件程式及同步動作中詢問目前設定的語言：

$AN_LANGUAGE_ON_HMI = <數值>

<值>	語言	語言代碼
1	德文 (德國)	GER
2	法文	FRA
3	英文（英國）	ENG
4	西文	ESP
6	義大利文	ITA
7	荷蘭文	NLD
8	簡體中文	CHS
9	瑞典文	SVE
18	匈牙利文	HUN
19	芬蘭文	FIN
28	捷克	CSY
50	葡文（巴西）	PTB
53	波蘭文	PLK
55	丹麥文	DAN
57	俄文	RUS
68	斯洛伐克文	SKY
72	羅馬尼亞文	ROM
80	繁體中文	CHT
85	韓文	KOR
87	日文	JPN
89	土耳其文	TRK

說明

更新 $AN_LANGUAGE_ON_HMI：

- 系統啟動之後。
- NC 及/或 PLC 重置之後。
- 切換至 M2N 範圍內的另一個 NC 之後。
- 切換 HMI 的語言之後。

附錄 A

A.1 縮寫表

A	
O	輸出
ADI4	（4 軸類比驅動器介面）
AC	適應性控制
ALM	ALM 電源模組
ARM	旋轉感應馬達
AS	自動化系統
ASCII	American Standard Code for Information Interchange：美國資訊交換編碼標準
ASIC	應用程式專屬積體電路：使用者切換電路
ASUB	非同步子程序
AUXFU	輔助功能：輔助功能
STL	敘述清單
UP	使用者程式

B	
OP	操作模式
BAG	模式群組
BCD	二進位編碼小數：以二進位代碼編碼之十進位數字
BERO	非接觸式近接開關
BI	可控開關量輸入
BICO	數位開關量接頭
BIN	BINary files：二進位檔案
BIOS	基本輸出輸入系統
BCS	基本座標系統

B	
BO	可控開關量輸出
OPI	操作面板介面

C	
CAD	電腦輔助設計
CAM	電腦輔助製造
CC	Compile Cycle：編譯循環
CEC	交互錯誤補正
CI	接頭輸入
CF 卡	（用於快閃記憶卡）
CNC	Computerized Numerical Control：電腦支援的數值控制
CO	接頭輸出
授權認證 (CoL)	授權認證
COM	通訊
CPA	編譯器投射資料：設定編譯器資料
CRT	陰極射線管：映像管
CSB	中央機板：PLC 模組
CU	控制單元
CP	通訊處理器
CPU	Central Processing Unit：中央處理單元
CR	歸位
CTS	清除以便傳送：準備傳送序列資料介面信號
CUTCOM	刀具半徑補償：刀具半徑補正

D	
DAC	數位到類比的轉換器
DB	資料單節 (PLC)
DBB	資料單節位元組 (PLC)
DBD	資料單節雙字組 (PLC)
DBW	資料單節字組 (PLC)

D	
DBX	資料單節位元 (PLC)
DDE	動態資料交換
DDS	驅動器資料集：驅動器資料集
DIN	德國工業標準
DIO	資料輸入 / 輸出：資料傳輸顯示
DIR	Directory：目錄
DLL	動態聯結函式庫
DO	驅動物件
DPM	雙通訊埠記憶體
DPR	雙通訊埠隨機存取記憶體
DRAM	動態記憶體（非緩衝）
DRF	Differential Resolver Function：差動解算器功能 (手輪)
DRIVE-CLiQ	與 IQ 連接至驅動元件
DRY	空跑：空跑進給率
DSB	Decoding Single Block：解碼單一單節
DSC	動態伺服控制 / 動態硬度控制
DW	資料字組
DWORD	雙字組（目前為 32 位元）

E	
I	輸入
EES	由外部儲存執行
I/O	輸入 / 輸出
ENC	編碼器：實際值編碼器
EFP	精簡 I/O 模組（PLC I/O 模組）
ESD	靜電感應裝置
EMC	電磁相容性
EN	歐洲標準
ENC	編碼器：實際值編碼器
EnDat	編碼器介面
EPROM	Erasable Programmable Read Only Memory：抹除式唯讀記憶體

E	
ePS 網路服務	網際網路式遠端機台維護服務
EQN	每轉 2048 個正弦信號的絕對編碼器之標識
ES	工程系統
ESR	延伸停止及回退
ETC	ETC 鍵 ">"；相同功能表的軟鍵列延伸

F	
FB	功能單節 (PLC)
FC	Function Call：功能單節 (PLC)
FEPROM	快閃抹除式唯讀記憶體：讀寫記憶體
FIFO	先進先出：不指定位址的記憶體，資料的讀取順序與儲存時的順序相同
FIPO	微調插補器
FPU	Floating Point Unit：浮點運算單元
CRC	銑刀半徑補正
FST	進給停止：進給率停止
FBD	功能單節圖（PLC 編寫方法）
FW	韌體

G	
GC	全域控制（PROFIBUS：廣播電訊）
GDIR	全域工件程式記憶體
GEO	幾何，如幾何軸
GIA	Gear Interpolation dAta：齒輪插補資料
GND	信號地端
GP	基本程式 (PLC)
GS	變速檔
GSD	說明 PROFIBUS 從動裝置的裝置主要檔案
GSDML	GSD 標記語言：建立 GSD 檔案的 XML 描述語言
GUD	Global User Data：全域使用者資料

H	
HEX	十六進位數之縮寫
AuxF	輔助功能
HLA	液壓線性驅動
HMI	人機介面：SINUMERIK 使用者介面
MSD	主軸磁碟
HW	硬體

I	
IBN	調試
ICA	插補補正
IM	Interface Module：互連模組
IMR	介面模組接收：用於接收資料之介面模組
IMS	介面模組傳送：用於傳送資料之介面模組
INC	Increment：增量
INI	Initializing Data：初始化資料
IPO	插補器
ISA	工業標準架構
ISO	國際標準組織

J	
JOG（寸動進給）	寸動進給：設定模式

K	
K_V	控制迴路的增益係數
K_p	比例增益
$K_Ü$	轉換率
LAD	階梯圖（PLC 編寫方法）

L	
LAI	Logic Machine Axis Image：邏輯機械軸圖像
LAN	區域網路
LCD	液晶顯示器：液晶顯示器
LED	Light Emitting Diode：發光二極體
LF	換行
PMS	位置量測系統
LR	定位控制器
LSB	最低有效位元：最低有效位元
LUD	Local User Data：使用者資料（本機）

M	
MAC	媒體存取控制
MAIN	主程式：主程式（OB1、PLC）
MB	百萬位元組
MCI	移動控制介面
MCIS	運動控制資訊系統
MCP	Machine Control Panel：機台控制面板
MD	機械參數
MDA（手動輸入）	手動資料自動執行：手動輸入
MDS	馬達資料集：馬達資料集
MSGW	訊息字組
MCS	機械座標系統
MM	馬達模組
MPF	Main Program File (主程式檔案)：主程式 (NC)
MCP	機台控制面板

N	
NC	Numerical Control：數值控制包含單節準備、移動範圍等
NCU	數值控制裝置：NC 硬體單元
NRK	NC 操作系統之名稱

N	
IS	介面訊號
NURBS	非均勻有理 B－曲線
WO	工件偏移
NX	數值擴充：軸擴充板

O	
OB	PLC　內組織單節
OEM	原始設備製造商
OP	Operator Panel (操作面板)：操作設備
OPI	Operator Panel Interface (操作面板介面)：連接到操作面板的介面
OPT	選項：選項
OLP	光學連結插頭：光纖匯流排連接器
OSI	開放系統內部連結：電腦通訊標準

P	
PIQ	流程影像輸出
PII	流程影像輸入
PC	個人電腦
PCIN	與控制項目進行資料交換時 SW 的名稱
PCMCIA	國際個人電腦記憶卡協會： 插入記憶卡標準化
PCU	PC 元件：PC 主機（電腦元件）
PG	編寫裝置
PKE	參數識別：PIV 的一部分
PIV	參數識別：數值（PPO 的參數化部份）
PLC	可程式邏輯控制器：適應控制
PN	PROFINET
PNO	PROFIBUS 使用者組織
PO	啟動
POU	程式組織單元
POS	位置 / 定位

P	
POSMO A	Positioning Motor Actuator：定位馬達致動器
POSMO CA	精簡 AC 定位馬達：具備整合供電系統和控制模組，以及定位單元和程式記憶體的完整驅動單元；AC 進給
POSMO CD	精簡 DC 定位馬達：類似 CA，但以 DC 進給
POSMO SI	整合式定位馬達伺服系統：定位馬達，DC 進給
PPO	參數流程資料物件：PROFIBUS DP 傳輸和「可變速度驅動器」檔案的循環資料電訊
PPU	面板處理單元（面板型 CNC 的中央硬體，如 SINUMERIK 828D）
PROFIBUS	處理區域匯流排：序列資料匯流排
PRT	程式測試
PSW	程式控制字組
PTP	點對點：點對點
PUD	Program global User Data：程式－全域使用者資料
PZD	流程資料：PPO 的流程資料部分

Q	
QEC	象限錯誤補正

R	
RAM	隨機存取記憶體：讀 / 寫記憶體
REF	參考點逼近功能
REPOS	重新定位功能
RISC	減少指令集電腦：小型指令集的處理器類型，具高速處理指令能力
ROV	快送超調：輸入修正
RP	R 參數，算術參數，預先定義使用者變數
RPA	R Parameter Active (啟用 R 參數)：NC 中用於 R 參數編號的記憶體區
RPY	Roll Pitch Yaw 三方位：座標系統之旋轉類型
RTLI	快速移動線性插補：快速移動動作時的線性插補
RTS	請求傳送：序列資料介面的控制訊號
RTCP	即時控制協定

S	
SA	同步動作
SBC	Safe Brake Control：Safe Brake Control
SBL	Single Block：單節
SBR	子程式：副程式 (PLC)
SD	設定資料
SDB	系統資料單節
SEA	設定資料生效：設定資料的識別碼（檔案類型）
SERUPRO	SEarch RUn by PROgram test：單節搜尋，程式測試
SFB	系統函數單節
SFC	系統函數呼叫
SGE	安全相關輸入
SGA	安全相關輸出
SH	安全停止
SIM	單電源模組
SK	軟鍵
SKP	略過：用來略過工件程式單節的功能
SLM	同步線性馬達
SM	步進馬達
SMC	箱型感測器模組
SME	外部感測器模組
SMI	整合式感測器模組
SPF	Sub Routine File：副程式 (NC)
PLC	可程式邏輯控制器
SRAM	靜態 RAM（非揮發性）
TNRC	刀鼻半徑補正
SRM	同步旋轉馬達
LEC	導螺桿錯誤補正
SSI	Serial Synchronous Interface：序列同步介面
SSL	單節搜尋
STW	Control word

S	
GWPS	研磨轉輪的周邊速率
SW	軟體
SYF	System Files：系統檔案
SYNACT	同步動作：同步動作

T	
TB	端子板 (SINAMICS)
TCP	Tool Center Point：刀尖
TCP/IP	傳輸控制通訊協定 / 網際網路通訊協定
TCU	Thin Client Unit
TEA	啟用測試資料：機台參數識別碼
TIA	完全整合自動化
TM	端子模組 (SINAMICS)
TO	Tool Offset：刀具偏移量
TOA	刀具偏移量已啟用：刀具偏移量的識別碼（檔案類型）
TRANSMIT	將銑削轉換為車削：銑床上銑刀操作的座標轉換
TTL	電晶體 – 電晶體邏輯（介面類型）
TZ	技術循環

U	
UFR	使用者框架：工件偏移
SR	副程式
USB	通用序列匯流排
UPS	不斷電系統

V	
VDI	NC 和 PLC 之間的內部通訊介面
VDI	Verein Deutscher Ingenieure [德國工程師協會]
VDE	Verband Deutscher Elektrotechniker [德國電子工程師協會]
VI	電壓輸入

V	
VO	電壓輸出
FDD	進給驅動器

W	
SAR	平滑逼近與返回
WCS	工件座標系統
T	刀具
TLC	刀長補正
WOP	工作場導向式程式設計
WPD	Workpiece Directory：工件目錄
TRC	刀具半徑補正
T	刀具
TO	刀具偏移量
TM	刀具管理
TC	換刀

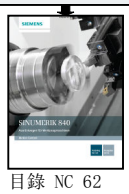
X	
XML	可擴展標示語言

Z	
WOA	工件偏移量：工件偏移量識別碼
ZSW	（驅動器）狀態字組

A.2 文件概觀

一般文件

廣告手冊
- SINUMERIK 840D sl
- SINUMERIK 828D
- SINUMERIK 828D 基本



目錄 NC 62
SINUMERIK 840D sl

目錄 NC 82
SINUMERIK 828D

目錄 PM 21
SIMOTION、SINAMICS S120

系統手冊
智慧機器
刀具操作

系統手冊
控制能源

使用者文件

操作手冊
- 通用
- 車削
- 銑削
- 研磨

編程手冊
- 基本知識
- 作業規劃
- 測量循環

編程手冊
- ISO 車削
- ISO 銑削

診斷手冊
警報

診斷手冊
警報

製造商 / 服務文件

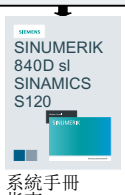
設備手冊
- NCU
- 操作元件及網路
- ADI4

設備手冊
調試手冊
維修手冊

調試手冊
- CNC：NCK、PLC、驅動器
- 基礎軟體與操作軟體

清單手冊
- 機械資料
- 介面訊號
- 變數

清單手冊
- 機械資料
- 介面訊號
- 參數
- 變數



系統手冊
指南
設定
機器

製造商 / 服務文件

功能手冊
- 基本功能
- 延伸功能
- 特殊功能
- 同步動作
- ISO 用語

功能手冊
刀具管理

功能手冊
Safety Integrated

功能手冊
Safety Integrated

設定手冊
- EMC 安裝指南
- 工業安全

資訊 / 訓練

訓練文件
- 簡單銑削 使用 ShopMill
- 簡單車削 使用 ShopTurn

手動
刀具與
模具製作

電子文件

DOConCD

Industry Online 支援 (SIOS)

Industry Mall

字彙表

C 曲線

C 曲線是最廣為人知及使用的曲線。插補點上的變化以切線及曲線持續進行。使用三元多項式。

C 軸

刀具主軸定義受控制旋轉和定位動作所環繞的軸。

CNC

請參閱 → NC

電腦數值控制：包含元件 → NC → 可程式邏輯控制器 (PLC)、人機介面 (HMI) → 組件物件模型 (COM)。

CNC

請參閱 → NC

電腦數值控制：包含元件 → NC → 可程式邏輯控制器 (PLC)、人機介面 (HMI) → 組件物件模型 (COM)。

COM

NC 的元件，用於通訊的完成和協調。

CPU

中央處理器，請參閱 → PLC

CU

轉換率

DRF

差動解算器功能：在自動模式中搭配電動手輪使用而產生增量工件偏移量的 NC 功能。

HIGHSTEP

AS300/AS400 系統 PLC 的程式設計選項摘要。

JOG（寸動進給）

控制系統的操作模式（設定模式）：可在寸動進給模式中設定機台。在寸動進給模式中，可使用方向鍵移動個別的軸和主軸。寸動進給模式中的其他功能包括：→ 原點復歸、→ 重新定位與→ 預設（設定實際值）。

KV

伺服器增益係數，一種在控制迴路中的控制變數。

MDI

控制系統的操作模式：手動資料輸入。在 MDI 模式中，個別程式單節或未參考主程式或子程式的單節序列，可以在啟動 NC 起始鍵之後立刻輸入並執行。

NC

CNC 的數值控制元件 →執行 → 工件程式，協調工具機的移動。

NRK

數值自動核心（NC 的操作系統）

NURBS

發生在控制中的動作控制和路徑插補是在 NURBS（**N**on **U**niform **R**ational **B**-**S**plines）基礎上執行的。對所有內部插補提供一致的程序。

OEM

針對想要在控制中建立自己的使用者介面或整合特定技術功能的機台製造商，已提供個別解決方案（OEM 應用程式）的範圍。

PCIN 資料傳輸程式

PCIN 是透過一個連續介面用於發送及接收 CNC 使用者資料（例如：工件程式、刀具偏移量等等）的工具程式。PCIN 程式可於標準工業電腦上以 MS-DOS 執行。

PLC

Programmable **L**ogic **C**ontroller： 可程式設計的邏輯控制器。 NC 元件： 可供處理加工刀具控制邏輯的可程式設計控制。

PLC 程式記憶體

SINUMERIK 840D sl： PLC 使用者程式、使用者資料及基礎 PLC 程式一併儲存於 PLC 使用者記憶體。

PLC 編程

PLC 是以 **STEP 7** 軟體進行程式設計。 STEP 7 程式設計軟體以 **WINDOWS** 標準操作系統為基礎，並包含創新加強的 STEP 5 程式設計功能。

R 參數

可在程式中由工件程式的程式設計師為任何目的設定或查詢的算術參數。

RS-232-C

資料輸入 / 輸出的序列介面。 可透過此介面載入和儲存加工程式、製造商和使用者資料。

TOA 元件

每個 TOA 區可以有超過一個 TOA 元件。 可能的 TOA 元件數量限於啟用中通道的最大數量。 TOA 元件包括一個刀具資料單節和一個刀庫資料單節。 此外，TOA 元件也可包含一個刀把資料單節（選配）。

TOA 區

TOA 區包含所有刀具和刀庫資料。 預設情況下，此區與資料存取相關的通道區會重疊。但是，加工資料可用來指定多通道分享一個 TOA 元件，因此，通道才能取得一般刀具管理資料。

WinSCP

WinSCP 為 Windows 用來傳送檔案之任意開放原始碼程式。

刀具

用來實施加工的加工刀具上的有效工件（如車削刀具、銑削刀具、鑽頭、雷射光束等）。

刀具半徑補正

為了直接將所需的工件輪廓程式設計，控制必須將基部重合路徑移動到程式設計輪廓，並考慮到正在使用刀具的半徑 (G41/G42)。

刀具偏移量

在計算路徑時考慮刀具尺寸。

刀鼻半徑補正

輪廓程式設計假設已指向刀具。 由於這不是實際情況，因此必須將所使用的刀具曲率半徑傳送到控制器，以便將其納入考量。 曲率中心必須和輪廓維持等距，以曲率半徑偏移。

子單節

前面加上「N」的單節含順序資訊（如位置資料）。

子程式

當工件程式嚴謹地區分為主與子程式時，「子程式」具有其原點。 在今日的 SINUMERIK NC 語言中已無嚴格的區別。 原則上，任何工件程式或循環皆可做為子程式在其他工件程式中被呼叫。 然後它將通過下一個程式層級 (x+1)（子程式層級 (x+1)）。

工件

要由加工刀具製作或加工的工件。

工件座標系統

工件座標系統具有在工件零點中的起點。 在工件座標系統中程式設計加工操作，尺寸和方向參考此系統。

工件偏移量

藉由參考現有零點和框架，為座標系統指定新的參考點。

1. 可設定
   每個 CNC 軸有可設定數量的可設定工件偏移量可以使用。以 G 指令選擇的偏移量會輪流生效。
2. 外部
   除了所有定義工件零點位置的偏移量之外，外部工件偏移量可藉由手輪（DRF 偏移量）或從 PLC 手動超調。
3. 可程式設計
   使用 TRANS 敘述來程式設計所有路徑和位置軸的工件偏移量。

工件程式

到 NC 的一系列敘述，會共同作用以產生特定的工件。 同樣地，此條件適用在執行已知原始工件上的特定加工操作。

工件程式單節

工件程式的工件，以換行作為區分。 可分成兩種類型： 主單節和子單節。

工件程式管理

工件程式管理可由工件組織而成。 使用者記憶體的大小會決定可管理的程式數量及資料量。每個檔案（程式及資料）名稱最多可含 24 個字母數字字元。

工件零點

工件零點為工件座標系統的起點。 依據到機械零點的距離而定義。

工件輪廓

設定要建立或加工的工件輪廓。

工作記憶體

工作記憶體為處理應用程式時，處理器於 CPU 中存取的 RAM。

工作區

刀尖根據加工刀具的實體設計而可移入的 3D 區域。請參閱 → 保護區。

工作區限制

有了工作區限制的輔助，除了限制開關以外，可進一步限制軸的移動範圍。每軸可用一對數值定義受保護的工作區。

中斷常式

中斷常式是特殊的子程式，可在加工處理中由事件（外部訊號）啟動。加工中的工件程式單節已中斷，並自動儲存中斷點的軸位置。

中繼單節

含選定刀具偏移量 (G41/G42) 的動作，可使用限量的中繼單節（偏移量平面中不含軸動作的單節）中斷，藉此仍可正確補正刀具偏移量。 控制器事先讀取的中繼單節許可數量，可在系統參數中設定。

公制測量系統

單位的標準化系統：以長度而言，如 mm（毫米）、m（公尺）。

反時限進給率

單節路徑移動所需時間亦可設定為軸動作，而不是進給速度 (G93)。

尺寸規格、公制和英制

在加工程式中，位置和螺距值可使用英制單位寫入程式。與可程式設計尺寸無關（G70/G71），控制設為基準系統。

手動超調

手動或可程式設計操作可能性，可讓使用者配合特定工件或材質而調整程式設計進給率或速度。

文字編輯器

請參閱 → 編輯器

主要單節

前面加上「:」的單節，其中包含用來啟動工件程式中操作順序的所有資訊。

主程式

當工件程式嚴謹地區分為主與子程式時，「主程式」具有其原點。在今日的 SINUMERIK NC 語言中已無嚴格的區別。原則上，通道中的任何工件程式皆可選擇及啟動。然後它將通過程式層級 0（主程式層級）。其他工件程式或循環可做為子程式在主程式中被呼叫。

主軸準停

於特定角度位置停止工件主軸，例如為了要在一特定位置上執行其他的加工。

主導軸

先導軸是操作員及程式設計師眼中的龍門軸，因此可如標準 NC 軸般受影響。

加工通道

藉由平行連續動作，通道結構可用來縮短停滯次數，例如加工時同時移動負載架。在此，CNC 通道必須視為分離於解碼、單節準備和插補的 CNC 控制系統。

可程式設計工作區限制

刀具到某一空間的動作空間限制是以程式設計限制而定義。

可程式設計框架

當正在執行工件程式時，可程式設計框架可供動態定義新的座標系統輸出點。 使用新框架和參考現有起點的附加定義，區分絕對定義。

可程式邏輯控制器

可程式設計邏輯控制器 (PLC) 為電子式控制器，其功能被儲存為控制元件中的程式。 這代表裝置的配置和佈線與控制器的功能無關。 可程式設計邏輯控制具有與電腦相同的結構；包括含記憶體的 CPU（中央模組）、輸入 / 輸出模組與內部匯流排系統。 週邊設備和程式設計語言符合控制技術的需求。

外部工件偏移量

由 → PLC 指定工件偏移量。

巨集技術

在單一識別碼下聚集一組敘述。識別碼代表程式中一組整合敘述。

同步

工件程式中，在某些加工點不同通道內順序協調的敘述。

同步化軸

同步軸是龍門軸，其設定位置不斷來自主導軸的動作，因此會與主導軸同步移動。從程式設計師和操作員眼中來看，同步軸「不存在」。

同步動作

1. 輔助功能輸出
   工件加工時，技術性功能（輔助功能）可由 CNC 程式輸出至 PLC。 例如，這些輔助功能用來控制加工刀具的附加設備，如套管、抓取器、固定夾頭等等。
2. 快速輔助功能輸出
   針對有時間性的切換功能，輔助功能的確認時間可以縮到最短，也可避免加工程序中不必要的暫停點。

同步軸

同步軸移動其路徑的時間和幾何軸移動路徑的時間相同。

多項式插補

多項式插補可以產生許多種的弧線，諸如**直線、拋物線、指數函數** (SINUMERIK 840D sl)。

安全功能

控制器配備永久啟用的監控功能，可即時偵測 CNC、PLC 和機台中的錯誤，大幅避免工件、刀具或機台損壞。 萬一發生錯誤，加工操作會中斷，驅動器也會停止。 記錄故障的原因，將以警報輸出。 於此同時，PLC 會收到觸發 CNC 警報的通知。

曲率

輪廓曲率 k 為輪廓點 (k = 1/r) 中巢狀環的反向半徑 r。

曲線插補

有了曲線插補，控制器可以從一組輪廓的幾個指定插補點，便能產生平滑的曲線特性。

自動

控制器的操作模式（根據 DIN 的單節順序操作）：NC 系統的操作模式連續選擇和執行子程式。

位址

位址是用於特定運算元或運算元範圍的識別碼，如輸入、輸出等。

快速移動

軸的最高移動速率。例如，用於當刀具從靜止位置逼近工件輪廓或將刀具從工件輪廓退刀時。快速移動速率是以機台專屬為基準，使用機械資料元件項目。

系統記憶體

系統記憶體是 CPU 內的記憶體，用來儲存以下資料：

- 操作系統所需的資料
- 運算元計時器、計數器、標記

系統變數

存在無工件程式的程式設計師輸入的變數。 以資料類型和變數名稱之前加上字元 **$** 而定義。請參閱 → 使用者定義變數。

使用者介面

使用者介面 (UI) 是 CNC 顯示的螢幕顯示媒介。具橫向和縱向軟鍵。

使用者定義變數

使用者可在工件程式或資料單節（全域使用者資料）中為任何目的登錄自己的變數。定義包含資料類型規格和變數名稱。請參閱 → 系統變數。

使用者記憶體

所有的程式和資料，如工件程式、子程式、註解、刀具偏移量和工作偏移量 / 框架，以及通道和程式使用者資料，皆可儲存在共用的 CNC 使用者記憶體中。

使用者程式

S7-300 自動裝置系統的使用者程式是使用程式設計語言 STEP 7 而建立的。使用者程式具有模組配置並含有個別單節。

基本單節類型為：

- 代碼單節
  這些單節含有 STEP 7 指令。
- 資料單節
  這些單節含 STEP 7 程式的常數和變數。

具震動限制之加速度

為了最佳化機台的加速度回應，並同時同步保護機械元件，可在加工程式中切換瞬間加速與持續（無晃動）加速。

刻度

框架元件，使用軸專屬的比例修改。

固定機台點

加工刀具唯一定義的點，例如加工參考點。

固定點逼近

加工刀具可使用定義方式逼近固定點，如刀具變更點、載入點、棘爪變更點等。這些點的座標會儲存在控制系統中。控制系統於可行的狀況下，便會快速移動→相關軸。

定位軸

在加工刀具上執行輔助動作的軸（如刀匣、棘爪傳輸）。 位置軸是不使用路徑軸插補的軸。

空白

加工前的工件原貌。

非同步子程式

可使用中斷訊號（如「Rapid NC input」（快速 NC 輸入）訊號）非同步啟動至（獨立的）目前程式狀態的工件程式。

保護區

工作區中刀具刀尖不可通過的 3D 區域。

按鍵開關

機台控制面板上的按鍵開關有四個位置，由控制器的操作系統指派功能。 按鍵開關有三種不同顏色的按鍵，可以按指定位置移除。

背隙補正

機械機台背隙的補正，例如，倒轉滾珠螺絲的背隙。可個別輸入各軸背隙補正。

英制測量系統

以英制和英制分數定義距離的測量系統。

負載記憶體

負載記憶體和 PLC CPU 314 的工作記憶體一樣。

限制速度

最高 / 最低（主軸）速度：可藉由指定機械資料、PLC 或設定資料來限制主軸的最高速度。

座標系統

請參閱 → 機台座標系統 → 工件座標系統

框架

框架是一種算術規則，可將一種直角座標系統轉換成其他的直角座標系統。框架包含以下元件：→ 工件偏移量、→ 旋轉、→ 比例、→ 鏡像。

訊息

在工件程式中程式設計的所有訊息及系統偵測到的警報，會以含有日期時間的純文字與刪除條件的對應符號顯示在操作員面板上。警報和訊息會分別顯示。

高速數位輸入 / 輸出

數位輸入可用於如啟動快速 CNC 程式（中斷常式）等方面。 可透過數位 CNC 輸出啟始高速、以程式驅動的開關功能

高階 CNC 語言

高階語言用於編寫 NC 程式 → 同步動作 → 技術循環。擁有以下內容：控制架構 → 使用者自訂變數 → 系統變數 → 巨集程式設計。

偏移記憶體

於控制系統中儲存含刀具偏移量的資料範圍。

動態前饋控制

使用動態、加速度相關的前饋控制，可實際消除由於下列錯誤所造成的輪廓不正確性。 即使使用高路徑速率，仍能產生極佳的加工正確性。 可透過工件程式，根據軸專屬基準選擇或取消選擇前饋控制。

參考點

機械軸參考測量系統的加工刀具位置。

啟動

開機後載入系統程式。

基本座標系統

藉由轉換而對應到機械座標系統上的直角座標系統。

程式設計師使用工件程式中基本座標系統的軸名稱。存在的基本座標系統平行於機械座標系統，如果不是則會啟用轉換。差異在於軸名稱中。

基本軸

其設定點或實際值位置構成補正值計算基準的軸。

從輪廓快速退刀

發生中斷時，可以透過 CNC 加工程式初始一個動作，使得刀具可以快速從目前加工的工件輪廓回退。 回退角度和回退距離也可以參數化。 可在快速退刀之後執行中斷常式。

接地

接地的目的在於萬一故障時，即使有危險的接點電壓，裝置的所有非啟用連結總零件也不會作用。

旋轉

框架元件定義環繞特定角度旋轉的座標系統。

旋轉軸

將工件或刀具旋轉套用到定義角度位置的旋轉軸。

移動範圍

線性軸的最大允許移動範圍為 ±90。 絕對值取決於選定輸入和位置控制解析度以及測量單位（英制或公制）。

設定資料

傳遞加工刀具的屬性到 NC 的資料，如系統軟體所定義。

軟鍵

名稱顯示在畫面區域上的按鍵。 軟鍵選項會動態配合操作狀況而顯示。 可在軟體中為可自由指派的功能鍵（軟鍵）指派定義的功能。

軟體限制開關

軟體限制開關會限制軸的移動範圍，避免硬體限制開關上的滑板瞬間停止。 可為每個軸指定兩個數值對，並使用 PLC 分別加以啟用。

通道

通道的特性是可以獨立處理其他通道的工件程式。通道以特定方式控制指派的軸和主軸。不同通道的工件程式執行可透過同步化而協調一致。

速率控制

為了使每個單節上極輕微動作達成可接受的移動率，可指定數個單節預期評估（預讀控制）。

連續路徑模式

連續路徑模式的目標在於避免實質減速工件程式單節邊界的路徑軸，並儘可能接近相同路徑速率變更下一個單節。

備用電池

備用電池可確保 CPU 中的使用者程式已儲存，因此不會受停電影響，且指定的資料區和位元記憶體、計時器和計數器能夠妥善保存。

備檔

從**外部**記憶體裝置上的檔案及 / 或目錄讀取。

單節

「Block」（單節）是為任何建立和處理程式需要的檔案提供的詞語。

單節搜尋

為了除錯目的或程式中止之後，可使用「Block search」（單節搜尋）功能選擇工件程式中要啟動或繼續的任何位置。

幾何

工件座標系統中的工件說明。

幾何軸

2 維或 3 維工件座標系統的幾何軸，在工件程式中已程式設計工件的幾何。

循環

受保護的子程式可用於工件上的重複加工操作。

插補補正

機台的機械偏差藉由差補補正功能來補償，例→ 導螺桿錯誤，懸垂，傾斜度和溫度補正。

插補器

NC 的邏輯元件可根據在工件程式中指定的結束位置，定義要在個別軸中執行動作的中間值。

無補正夾頭的攻牙

這項功能允許無補正夾頭的攻牙螺紋。藉由將主軸作為轉軸、鑽孔軸的插補法，螺紋可切削成精確的最終鑽孔深度，例如，用於閉孔螺紋（需求：軸操作中的主軸）。

硬體設定

SIMATIC S7 工具適用於 S7 專案內硬體元件之配置及參數設定

程式設計鍵

在工件程式的程式設計語言中已有定義的意義之字元和字元字串。

程式單節

程式單節包括工件程式的主程式和子程式。

程式層級

工件程式在通道中啟動並做為主程式在程式層級 0（主程式層級）執行。 在主程式中呼叫的任何工件程式均做為子程式在其自身的程式層級 1 ... n 執行。

絕對尺寸

軸動作的目的地是以參考目前有效座標系統原點的尺寸所定義的。請參閱 → 增量尺寸

診斷

1. 控制的操作區。
2. 控制具有自我診斷程式以及用來提供服務的測試功能：狀態、警報和服務畫面

象限錯誤補正

導引路徑上摩擦力改變所導致的象限變化上的輪廓錯誤，可利用象限錯誤補正完全排除。象限錯誤補正的參數化由電路測試執行。

軸

CNC 軸可根據其功能範圍細分為：

- 軸：插補路徑軸
- 輔助軸：含軸專屬進給率的非插補進給與位置軸。輔助軸與實際加工無關，例如刀具進給器、刀匣等。

軸名稱

為確保清楚的辨識，所有通道及控制系統的機械軸必須於通道及控制系統中指派獨特的名稱。其中幾何軸稱為 X、Y、Z。繞著幾何軸旋轉的旋轉軸稱為 A、B、C。

軸位址

請參閱 → 軸名稱

週邊模組

I/O 模組呈現 CPU 和處理之間的連結。

I/O 模組為：

- → 數位輸入 / 輸出模組
- → 類比輸入 / 輸出模組
- → 模擬模組

進給手動超調

程式設計速率是透過機台控制面板或從 PLC（0 到 200%）所作的目前速率設定而調整。也可以利用加工程式中的可程式設計百分比係數（1 至 200%）修正進給率。

傳輸率

資料傳輸率（位元 / 秒）。

傾斜表面加工

不在機台座標平面上的工件表面，可利用「傾斜表面加工」功能輕易地執行鑽孔銑削操作。

圓形插補

刀具在輪廓上的指定點之間以已知進給率呈圓弧狀移動，藉以加工工件。

極座標

一個座標系統根據點到原點的距離與由半徑向量和定義軸所形成的角度，定義點在平面上的位置。

補正表

含插補點的表格。在基準軸上的選定位置提供補正軸的補正值。

補正值

軸位置之間的差距由編碼器和所需的程式設計軸位置測量。

補正軸

含設定點或由補正值修改實際值的軸

資料字組

在一個資料單節中的雙位元資料單元。

資料區塊

1. HIGHSTEP 程式可存取的 PLC 資料元件。
2. NC 的資料元件：資料區塊包含全域使用者資料的資料定義。此資料在定義後可直接初始化。

路徑速率

最大可程式設計路徑速率取決於輸入解析度。例如，解析度 0.1 毫米時，最大可程式設計路徑速率為 1000 公尺 / 分鐘。

路徑軸

路徑軸包含由插補器以同時啟動、加速、停止和到達終點的方式所控制通道的所有機械軸。

路徑進給率

路徑進給率影響 → 路徑軸。代表相關幾何軸的進給率幾何總和。

預先一致

當路徑距離逼近到相當於端點位置的指定舊資料時，發生單節更改。

預讀控制

預讀控制功能藉由預測移動單節的可指派數目，以獲得最佳加工速度。

精加工工件輪廓

精加工工件輪廓。請參閱 → 原始工件。

精確停止

當程式設定精確停止陳述式時，會精確逼近在單節中指定的位置，且若有需要，會使用非常緩慢的速度。為了降低逼近時間，為快速移動和進給定義精確停止臨界值。

精確停止臨界值

當所有路徑軸到達其精確停止臨界值時，控制會將其視為已到達其精確目的地點而回應。產生工件程式前單節。

網路

網路係多重 S7-300 和其他終端裝置的連結，例如透過連接線的程式設計裝置。資料交換發生於連結裝置間的網路上。

輔助功能

輔助功能可使工件程式傳輸參數到 PLC，然後會觸發由機台製造商定義的反應。

增量

根據增量數指定移動路徑距離。 增量數可儲存為設定資料，或利用合適標示按鍵加以選擇（即 10、100、1000、10000）。

增量進給

增量尺寸：軸移動的目的地是由待涵蓋的距離以及參考已到達點的方向所定義。請參閱 → 絕對尺寸。

標準循環

標準循環是為經常重複的加工操作而提供：

- 用於鑽孔 / 銑削技術
- 用於車削技術

可用的循環列在「Program」（程式）操作區中的「Cycle support」（循環支援）功能表中。 一旦選擇了所需的加工循環，指定值所需的參數就會以純文字顯示。

模式

SINUMERIK 控制的操作概念。定義模式如下所示：→ 寸動進給、→ MDI、→ 自動。

模式群組

技術上相關的軸和主軸可合併成一個模式群組。可由一個或多個通道控制模式群組的軸 / 主軸。相同的模式類型通常指派至模式群組的通道。

線性插補

加工工件時，刀具會沿著直線移動到目的地點。

線性軸

與轉軸相反，線性軸用來定義直線。

編輯器

編輯器可供建立、編輯、延伸、聯結和匯入程式 / 文字 / 程式單節。

輪廓

工件的輪廓

輪廓監控

作為輪廓精確度的測量方法，可定義的允差頻帶中監控到下列錯誤。例如，高到令人無法接受的後續錯誤會造成驅動器過載。在這種情況下，會輸出警告並停止各軸。

導螺桿錯誤補正

補正進給中使用的導螺桿的機械性不精確。 控制器使用補正的儲存誤差值。

整體重置

萬一整體重置，則會刪除 CPU 的以下記憶體：

- → 工作記憶體
- 負載記憶體的讀取 / 寫入區域
- → 系統記憶體
- 備份記憶體

機台控制面板

加工刀具上的操作員面板，例如具按鍵、旋轉開關等操作元件，以及如 LED 的簡單指示燈。透過 PLC 用來直接影響加工刀具。

機械座標系統

與加工刀具軸相關的座標系統。

機械軸

加工刀具上實體存在的軸。

機械零點

可加以追蹤所有（衍生的）測量系統的加工刀具固定點。

磨圓軸

磨圓軸將工件或刀具旋轉至對應指標網格的角度位置。 當到達網格指標時，磨圓軸即「到位」。

螺旋插補

螺旋插補功能非常適用於使用成形銑刀加工內部和外部螺紋以及銑削潤滑溝槽。

螺旋有兩種動作方式：

- 在一平面上的螺旋動作
- 垂直於此平面的線性動作

轉換

軸的附加或絕對零點偏移量。

識別碼

依據 DIN 66025，字組是用變數（算術變數、系統變數、使用者變數）、子程式、關鍵字和含多重位址字母的字之識別碼（名稱）附加而成的。這些補充和單節格式相關字組具有相同的意義。識別碼必須為唯一。不可將相同的識別碼用在不同的物件上。

鏡射

鏡像反轉輪廓相對於軸的座標值符號。 可一次相對於多軸進行鏡射。

關鍵字

具有特定標記法、且在工件程式的程式設計語言中有已定義意義的字組。

警報

全部 → 訊息和警報會以含有日期時間的純文字與刪除條件的相應符號顯示在操作員面板上。警報和訊息會分別顯示。

1. 工件程式中的警報和訊息：
   可直接從工件程式以純文字顯示警報和訊息。
2. 從 PLC 的警報和訊息：
   可直接從 PLC 程式以純文字顯示警報和訊息。此用途不需要其他的功能單節套件。

驅動器

驅動器為 CNC 的元件，根據 NC 的設定執行速度和扭矩控制。

變數定義

變數定義包括指定資料類型和變數名稱。 變數名稱可用來存取變數值。

索引

-

- 嘗試切削添加結束 - GROUP_ADDEND
  外部程式設計, 914

$

$AA_ATOL, 600
$AA_COUP_ACT
  用於軸主動值耦合, 631
  耦合動作期間, 606
$AA_ESR_ENABLE, 729
$AA_LEAD_SP, 631
$AA_LEAD_SV, 631
$AC_ACT_PROG_NET_TIME, 716
$AC_ACTUAL_PARTS, 720
$AC_AXCTSWA, 700
$AC_AXCTSWE, 700
$AC_CTOL, 600
$AC_CUT_INV, 519
$AC_CUTMOD, 518
$AC_CUTMOD_ANG, 518
$AC_CUTMODK, 518
$AC_CUTTING_TIME, 716
$AC_CYCLE_TIME, 716
$AC_DELAYFST, 581
$AC_ESR_TRIGGER, 729
$AC_OLD_PROG_NET_TIME, 716
$AC_OLD_PROG_NET_TIME_COUNT, 716
$AC_OPERATING_TIME, 716
$AC_OTOL, 600
$AC_PROG_NET_TIME_TRIGGER, 717
$AC_REPOS_PATH_MODE, 590
$AC_REQUIRED_PARTS, 719
$AC_SMAXVELO, 595
$AC_SMAXVELO_INFO, 595
$AC_SPECIAL_PARTS, 720
$AC_TOTAL_PARTS, 719
$AC_TRAFO_CORR_ELEM_P, 446
$AC_TRAFO_CORR_ELEM_T, 446
$AC_TRAFO_ORIAX_LOC, 446
$AN_AXCTAS, 700
$AN_AXCTSWA, 700
$AN_ESR_TRIGGER, 729
$AN_LANGUAGE_ON_HMI, 1042
$AN_POWERON_TIME, 715
$AN_SETUP_TIME, 715
$NT_CLOSE_CHAIN_T, 446
$NT_CNTRL, 446
$NT_CORR_ELEM_P, 445
$NT_CORR_ELEM_T, 445
$NT_NAME, 437
$NT_ROT_AX_NAME, 514
$NT_TRAFO_INDEX, 437
$P_ACTBFRAME, 343
$P_AD, 518
$P_BFRAME, 342
$P_CHBFRAME, 342
$P_CHBFRMASK, 343
$P_CTOL, 600
$P_CUT_INV, 519
$P_CUTMOD, 518
$P_CUTMOD_ANG, 518
$P_CUTMOD_ERR, 520
$P_CUTMODK, 518
$P_DELAYFST, 581
$P_IFRAME, 344
$P_IS_EES_PATH, 244
$P_NCBFRAME, 342
$P_NCBFRMASK, 343
$P_ORI_DIFF, 509
$P_ORI_POS, 509
$P_ORI_SOL, 510
$P_ORI_STAT, 513
$P_OTOL, 600
$P_PATH, 243
$P_PFRAME, 344
$P_PROG, 244
$P_PROGPATH, 244
$P_SIM, 310
$P_STACK, 243
$P_SUBPAR, 179
$P_TOOLENV, 529
$P_TOOLENVN, 528
$PA_ATOL, 600
$SA_LEAD_TYPE, 630
$SC_CONTPREC, 574
$SC_MINFEED, 574
$SC_PA_ACTIV_IMMED, 257
$SN_PA_ACTIV_IMMED, 257
$TC_CARR1...14, 493
$TC_CARR18...23, 494
$TC_CARR18[m], 497
$TC_DP1 ... 25, 447
$TC_ECPxy, 452
$TC_SCPxy, 452

*

*（算術函數）, 81

/

/（算術函數）, 81

+

+（算術函數）, 81

<

<（比較運算子）, 83
<<（串鍊運算子）, 94
<=（關係運算子）, 83
<>（比較運算子）, 83

=

==（比較運算子）, 83

>

>（比較運算子）, 83
>=（關係運算子）, 83

0

0 字元, 91

A

ABS, 81
ACCLIMA, 569
ACOS, 81
ACTBLOCNO, 191
ACTFRAME, 317
ADISPOSA, 312
ALF
　從輪廓快速退刀, 146
AND, 83
APR, 49
APRB, 49
APRP, 49
APW, 49
APWB, 49
APWP, 49
AS, 230
ASIN, 81
ASPLINE, 273
ATAN2, 81
ATOL, 597
AV, 639
AX, 691
AXCTSWE, 699
AXCTSWEC, 699
AXCTSWED, 699
AXIS, 31
AXNAME, 92
AXSTRING, 691
AXTOCHAN, 155
AXTOSPI, 691
A 曲線, 280

B

B_AND, 83
B_NOT, 83
B_OR, 83
B_XOR, 83
BAUTO, 273
BFRAME, 317
BLOCK, 217
BLSYNC, 140
BNAT, 273
BOOL, 31
BOUND, 87
BRISK, 567
BRISKA, 567
BSPLINE, 273
BTAN, 273
B 曲線, 281

C

CAC, 271
CACN, 271
CACP, 271
CALL, 216
CALLPATH, 220
CASE, 114
CDC, 271
CFINE, 328
CHAN, 31
CHANDATA, 245
CHAR, 31

Check
結構, 124
CHKDNO, 489
CIC, 271
CLEARM, 132
CLRINT, 142
COARSE, 639
COARSEA, 312
COLLPAIR, 430
COMPCAD, 287
COMPCURV, 287
COMPLETE, 245
COMPOF, 287
COMPON, 287
COMPSURF, 287
CONTDCON, 753
CONTPRON, 746
CORRTRAFO, 439
COS, 81
COUPDEF, 639
COUPDEL, 639
COUPOF, 639
COUPOFS, 639
COUPON, 639
COUPONC, 639
COUPRES, 639
CP, 406
CPBC, 651
CPDEF, 650
CPDEL, 650
CPFMOF, 653
CPFMON, 653
CPFMSON, 652
CPFPOS + CPOF, 653
CPFPOS + CPON, 651
CPFRS, 651
CPLA, 650
CPLCTID, 651
CPLDEF, 650
CPLDEL, 650
CPLDEN, 651
CPLINSC, 655
CPLINTR, 655
CPLNUM, 651
CPLOF, 651
CPLON, 651
CPLOUTSC, 655
CPLOUTTR, 655
CPLPOS, 651
CPLSETVAL, 651
CPMALARM, 656
CPMBRAKE, 656
CPMPRT, 655
CPMRESET, 654
CPMSTART, 655
CPMVDI, 656
CPOF, 650
CPON, 650
CPRECOF, 574
CPRECON, 574
CPROT, 254
CPROTDEF, 249
CPSETTYPE, 656
CPSYNCOP, 655
CPSYNCOP2, 655
CPSYNCOV, 655
CPSYNFIP, 655
CPSYNFIP2, 655
CPSYNFIV, 655
CSPLINE, 273
CTAB, 619
CTABDEF, 608
CTABDEL, 615
CTABEND, 608
CTABEXISTS, 614
CTABFNO, 624
CTABFPOL, 624
CTABFSEG, 624
CTABID, 617
CTABINV, 619
CTABISLOCK, 617
CTABLOCK, 616
CTABMEMTYP, 617
CTABMPOL, 624
CTABMSEG, 624
CTABNO, 624
CTABNOMEM, 624
CTABPERIOD, 617
CTABPOL, 624
CTABPOLID, 624
CTABSEG, 624
CTABSEGID, 624
CTABSEV, 619
CTABSSV, 619
CTABTEP, 619
CTABTEV, 619
CTABTMAX, 619
CTABTMIN, 619
CTABTSP, 619
CTABTSV, 619
CTABUNLOCK, 616
CTOL, 597
CTRANS, 328
CUT3DC, 467

CUT3DCC, 477
CUT3DCCD, 477
CUT3DCD, 467
CUT3DF, 471
CUT3DFD, 471
CUT3DFF, 471
CUT3DFS, 471
CUTMOD, 515
CUTMODK, 515
CYCLE4071
外部程式設計, 893
CYCLE4072
外部程式設計, 895
CYCLE4073
外部程式設計, 898
CYCLE4074
外部程式設計, 899
CYCLE4075
外部程式設計, 903
CYCLE4077
外部程式設計, 905
CYCLE4078
外部程式設計, 908
CYCLE4079
外部程式設計, 910
CYCLE435 - 設定修整器座標系統
外部程式設計, 846
CYCLE495 - 形狀修整
外部程式設計, 847
CYCLE60－刻字
外部程式設計, 786
CYCLE61－平面銑削
外部程式設計, 788
CYCLE62－輪廓呼叫
外部程式設計, 791
CYCLE63－銑削輪廓挖槽
外部程式設計, 792
CYCLE64－預鑽孔輪廓挖槽
外部程式設計, 796
CYCLE70－螺紋銑削
外部程式設計, 797
CYCLE72－路徑銑削
外部程式設計, 800
CYCLE76－矩形島嶼
外部程式設計, 804
CYCLE77－圓弧島嶼
外部程式設計, 806
CYCLE78－鑽孔螺紋銑削
外部程式設計, 809
CYCLE79－多邊緣
外部程式設計, 812
CYCLE800 - 旋轉
外部程式設計, 849
CYCLE801－網格 / 框架位置樣式
外部程式設計, 853
CYCLE802－任意位置
外部程式設計, 855
CYCLE81－中心鑽
外部程式設計, 815
CYCLE82－鑽孔
外部程式設計, 816
CYCLE830 - 深孔鑽 2
外部程式設計, 859
CYCLE832 - 高速設定
外部程式設計, 865
CYCLE83－深孔鑽
外部程式設計, 819
CYCLE840 - 搭配補正夾頭進行攻牙
外部程式設計, 869
CYCLE84－不搭配補正夾頭進行攻牙
外部程式設計, 823
CYCLE85－鉸孔
外部程式設計, 827
CYCLE86－鏜孔
外部程式設計, 829
CYCLE899－銑削開孔槽
外部程式設計, 872
CYCLE92－切離
外部程式設計, 830
CYCLE930－槽
外部程式設計, 875
CYCLE940－底切
外部程式設計, 879
CYCLE951－材料移除
外部程式設計, 882
CYCLE952 - 輪廓切槽
外部程式設計, 885
CYCLE95－輪廓切削
外部程式設計, 832
CYCLE98－螺紋鍊
外部程式設計, 834
CYCLE99－螺紋車削
外部程式設計, 841
C 曲線, 282

D

DCI, 54
DCM, 54
DCU, 54
DEF, 31
DEFAULT, 114
DEFINE ... AS, 230

DELAYFSTOF, 579
DELAYFSTON, 579
DELDL, 453
DELETE, 163
DELOBJ, 421
DELTOOLENV, 526
DIN 子程式名稱, 242
DISABLE, 142
DISPLOF, 191
DISPLON, 191
DISPR, 584
DIV, 81
DL, 451
DO, 661
DRIVE, 567
DRIVEA, 567
DV, 639
DYNFINISH, 571
DYNNORM, 571
DYNPOS, 571
DYNROUGH, 571
DYNSEMIFIN, 571
D 數量
  Check, 489
  任意指派, 489
  重新命名, 490

E

EAUTO, 273
EES, 236
EES 標記, 238
EG
  電子齒輪, 632
EGDEF, 632
EGDEL, 637
EGOFC, 636
EGOFS, 636
EGON, 633
EGONSYN, 633
EGONSYNE, 633
ELSE, 125
ENABLE, 142
ENAT, 273
ENDFOR, 127
ENDIF, 125
ENDLABEL, 117
ENDLOOP, 127
ENDWHILE, 129
ESR, 728
ESRR, 735
ESRS, 734
ETAN, 273
EVERY, 661
EXECSTRING, 79
EXECTAB, 759
EXECUTE, 762
EXP, 81
EXTCALL
  僅適用 SINUMERIK 828D, 225
  僅適用 SINUMERIK 840D sl, 221
EXTCLOSE, 721
EXTERN, 210
EXTOPEN, 721

F

FALSE, 31
FCTDEF, 462
FCUB, 562
FENDNORM, 311
FFWOF, 573
FFWON, 573
FIFOCTRL, 576
FILEDATE, 170
FILEINFO, 170
FILESIZE, 170
FILESTAT, 170
FILETIME, 170
FINE, 639
FINEA, 312
FLIN, 562
FNORM, 562
FOR, 127
FPO, 562
FRAME, 31
FROM, 661
FTOCOF, 465
FTOCON, 465

G

G 碼
  間接程式設計, 74
G290, 742
G291, 742
G5, 401
G62, 311
G621, 311
G7, 401
G810 ... G819, 310
G820 ... G829, 310
GEOAX, 694

GET, 150
GETACTTD, 491
GETD, 150
GETDNO, 490
GETTCOR, 529
GETTENV, 527
GETVARAP, 66
GETVARDFT, 69
GETVARDIM, 69
GETVARLIM, 68
GETVARPHU, 66
GETVARTYP, 71
GOTO, 111
GOTOB, 111
GOTOC, 111
GOTOF, 111
GOTOS, 110
GP, 76
GROUP_ADDEND - 嘗試切削添加結束
    外部程式設計, 914
GROUP_BEGIN - 程式單節開始處
    外部程式設計, 913
GROUP_END - 程式單節結束
    外部程式設計, 914
GUD, 32
G 群組
    技術, 571

H

HOLES1－直線位置樣式
    外部程式設計, 766
HOLES2－圓位置樣式
    外部程式設計, 766

I

ID, 661
IDS, 661
IF, 125
IFRAME, 317
INDEX, 96
INICF, 31
INIPO, 31
INIRE, 31
INIT, 132
INITIAL, 245
INITIAL_INI, 245
INT, 31
INTERSEC, 757
IPOBRKA, 312
IPOENDA, 312
IPOSTOP, 639
IPTRLOCK, 582
IPTRUNLOCK, 582
ISAXIS, 691
ISFILE, 168
ISNUMBER, 92
ISOCALL, 218
ISVAR, 64

J

JERKLIM, 593
JERKLIMA, 569

L

L..., 208
LEAD, 362
LEADOF, 626
LEADON, 626
LENTOAX, 551
LIFTFAST, 144
LLI, 45
LN, 81
LONGHOLE - 細長孔洞
    外部程式設計, 783
LOOP, 127
LUD, 32

M

M17, 195
M30, 195
MASLDEF, 657
MASLDEL, 657
MASLOF, 657
MASLOFS, 657
MASLON, 657
MATCH, 96
MAXVAL, 87
MCALL, 214
MD10010, 132
MD10280, 132
MD15800, 28
MD18104, 526
MD18116, 526
MD18156, 28
MD20360, 534
MD24558, 536
MD24658, 536

MEAC, 300
MEAFRAME, 336
MEAS, 297
MEASA, 300
MEAW, 297
MEAWA, 300
MINDEX, 96
MINVAL, 87
MMC, 709
MOD, 81
MODAXVAL, 691
MPF, 234

N

NAMETOINT, 426
NCK, 31
NCK 標記, 238
NEWCONF, 157
NOC, 639
NOT, 83
NPROT, 254
NPROTDEF, 249
NUMBER, 92
NUT, 374

O

OEM 位址, 310
OEM 函數, 310
OEMIPO1/2, 310
OMA1 ... OMA5, 310
OR, 83
ORIAXES, 371
ORIC, 483
ORICONCCW, 374
ORICONCW, 374
ORICONIO, 374
ORICONTO, 374
ORICURVE, 377
ORID, 483
ORIEULER, 371
ORIMKS, 369
ORIPATH, 386
ORIPATHS, 386
ORIPLANE, 374
ORIRESET（A，B，C）, 360
ORIROTA, 382
ORIROTC
    用於刀具旋轉方向, 382
    插補刀具旋轉時, 388
ORIROTR, 382
ORIROTT, 382
ORIRPY, 371
ORIRPY2, 371
ORIS, 483
ORISOF, 395
ORISOLH, 505
ORISON, 395
ORIVECT, 371
ORIVIRT1, 371
ORIVIRT2, 371
ORIWKS, 369
OS, 663
OSB, 663
OSC, 483
OSCILL, 669
OSCTRL, 663
OSD, 483
OSE, 663
OSNSC, 663
OSOF, 483
OSP1, 663
OSP2, 663
OSS, 483
OSSE, 483
OST, 483
OST1, 663
OST2, 663
OTOL, 597

P

P..., 213
P_ACTFRAME, 344
PCALL, 219
PDELAYOF, 677
PDELAYON, 677
PFRAME, 317
PHI
    方向多項式, 381
    沿著錐形柱面的方向, 374
PHU, 46
PL
    多項式插補, 288
    曲線插補, 273
PO, 288
PO[PHI]
    方向多項式, 381
    用於刀具旋轉方向, 386
    沿著錐形柱面的方向, 374
PO[PSI]
    方向多項式, 381

用於刀具旋轉方向, 386
沿著錐形柱面的方向, 374
PO[THT]
方向多項式, 381
用於刀具旋轉方向, 386
PO[XH]
方向多項式, 381
用於兩接點的方向規格, 377
PO[YH]
方向多項式, 381
用於兩接點的方向規格, 377
PO[ZH]
方向多項式, 381
用於兩接點的方向規格, 377
POCKET3¡Ð¯x§ÎµÄ
外部程式設計, 769
POCKET4－圓弧腔
外部程式設計, 773
POLF
用於 NC 控制回退, 729
POLFA, 729
POLFMASK
用於 NC 控制回退, 729
POLFMLIN
用於 NC 控制回退, 729
POLY, 288
POLYPATH, 288
PON, 686
PONS, 677
POSFS, 639
POT, 81
PREPRO, 195
PRESETON, 332
PRESETONS, 334
PRIO, 140
PRLOC, 31
PROTA, 432
PROTD, 434
PROTS, 433
PSI
方向多項式, 381
沿著錐形柱面的方向, 374
PTP, 406
PTPG0, 406
PTPWOC, 406
PUD, 32
PUNCHACC, 677
PUTFTOC, 465
PUTFTOCF, 464
PW, 273

R

READ, 165
REAL, 31
REDEF, 37
RELEASE, 150
REP, 56
REPEAT, 117
REPEATB, 117
REPOSA, 584
REPOSH, 584
REPOSHA, 584
REPOSL, 584
REPOSQ, 584
REPOSQA, 584
RET, 196
RET（可參數化）, 198
RETB（可參數化）, 204
RG, 27
RINDEX, 96
RMBBL, 584
RMEBL, 584
RMIBL, 584
RMNBL, 584
ROUND, 81
ROUNDUP, 173
RPY, 365
Run MyScreens, 709

S

SAVE, 185
SBLOF, 186
SBLON, 186
SCPARA, 703
SD, 273
SD41610, 446
SD41611, 446
SD42475, 393
SD42476, 393
SD42477, 393
SD42900, 456
SD42910, 456
SD42920, 457
SD42930, 458
SD42935, 460
SD42940, 461
SD42984, 517
SET, 56
SETAL, 726

SETDNO, 490
SETINT, 140
SETM, 132
SETTCOR, 536
SIN, 81
SLOT1－長形槽
　外部程式設計, 777
SLOT2－環槽
　外部程式設計, 780
SOFT, 567
SOFTA, 567
SON, 677
SONS, 677
SPF, 234
SPI, 691
SPIF1, 677
SPIF2, 677
SPLINEPATH, 285
SPN, 683
SPOF, 677
SPP, 683
SPRINT, 100
SQRT, 81
START, 132
STARTFIFO, 576
STAT, 407
STOPFIFO, 576
STOPRE, 576
STRING, 31
STRINGIS, 705
STRLEN, 96
SUBSTR, 97
SYNR, 31
SYNRW, 31
SYNW, 31

T

TAN, 81
TANG, 555
TANGDEL, 560
TANGOF, 559
TANGON, 558
TCARR, 499
TCOABS, 499
TCOFR, 499
TCOFRX, 499
TCOFRY, 499
TCOFRZ, 499
THETA
　用於刀具旋轉方向, 382
　插補刀具旋轉時, 388
TILT, 362
TLIFT, 557
TMOF, 689
TMON, 689
TOFFOF, 502
TOFFON, 502
TOLOWER, 95
TOOLENV, 524
TOUPPER, 95
TOWBCS, 458
TOWKCS, 458
TOWMCS, 458
TOWSTD, 458
TOWTCS, 458
TOWWCS, 458
TRAANG
　使用可程式設計角度, 400
TRACON, 404
TRACYL, 397
TRAFOOF, 420
TRAFOON, 437
TRAILOF, 603
TRAILON, 603
TRANSMIT, 397
TRAORI, 359
TRUE, 31
TRUNC, 81
TU, 412

U

ULI, 45
UNTIL, 130

V

VELOLIM, 594
VELOLIMA, 569

W

WAITC, 639
WAITE, 132
WAITENC, 701
WAITM, 132
WAITMC, 132
WHEN, 661
WHEN-DO, 672
WHENEVER, 661
WHENEVER-DO, 672
WHILE, 129

WRITE, 158

X

XOR, 83

刀

刀刃數量, 489
刀把
    - 可定向, 499
    動力學, 493
刀具
    - 方向, 483
    -框架變更的方向, 501
    -參數, 447
刀具方向
    相對於路徑, 385
刀具半徑補正
    轉角減速, 311
刀具偏移量
    附加, 451
    偏移記憶體, 447
    磨損值的座標系統, 458
刀具鏈, 444

子

子程式
    可程式設計搜尋路徑, 220
    名稱, 175
    沒有參數傳輸的呼叫, 208
    具有參數傳輸的呼叫, 210
    呼叫，間接, 216
    呼叫，模態, 214
    重覆, 213
    傳回，可參數化 (RET...), 198
    傳回，可參數化 (RETB...), 204
    應用, 174

工

工件, 737
    主目錄, 234
    目錄, 235
    計數器, 719
工件鏈, 444
工作記憶體, 245

不

不區分大小寫, 237
不搭配補償夾頭進行攻牙－CYCLE84
    外部程式設計, 823

中

中心鑽－CYCLE81
    外部程式設計, 815
中斷常式
    可程式設計移動方向, 146
    回退移動, 146
    刪除, 142
    停用 / 啟動, 142
    從輪廓快速退刀, 144
    新指派, 141

內

內角的轉角減速, 311

分

分母多項式, 292

切

切片, 677
切離－CYCLE92
    外部程式設計, 830

尤

尤拉角, 365

方

方向向量, 365
方向向量 THETA, 382
方向程式設計, 371
方向軸, 371
方向轉換 TRAORI
    一般 5/6 軸轉換, 351
    方向移動, 349
    方向程式設計, 360
    方向程式設計變數, 360
    機台動態, 350

比

比較運算子, 83

主

主動值耦合
    先導軸與跟隨軸同步, 629
    實際值與設定點耦合, 629
主控值模擬, 630
主軸
    替換, 150
主導軸
    用於軸主動值耦合, 626

加

加速模式, 567

可

可切換幾何軸, 694
可用性
    與系統相關, 5

外

外部程式設計, 913
外部零點偏移量, 330

巨

巨集, 230

平

平面銑削, 368
平面銑削－CYCLE61
    外部程式設計, 788
平滑化
    方向特性, 395

目

目錄路徑, 240

任

任意位置－CYCLE802
    外部程式設計, 855

全

全域工件程式記憶體 (GDIR), 236

同

同步
    粗調, 642
    微調, 642
同步主軸
    配對定義, 644
    耦合, 639
同步震盪
    下一個部分進給, 675
    同步動作, 672
    定義進給, 672
    於反轉點範圍內進給, 673
    評估，IPO 循環, 674
    進給移動, 673
    震盪與進給軸的指派：, 671

回

回退
    NC 控制, 729
    自主驅動, 735

多

多項式係數, 289
多項式插補, 288
多邊緣－CYCLE79
    外部程式設計, 812

字

字串
    串鍊, 94
    長度, 96
    格式化, 100
    -運算, 91

曲

曲線群組, 285

自

自動中斷指示器, 583
自動路徑分段, 683

串

串鍊
    字串的, 94

位

位置同步, 639

刪

刪除剩餘距離, 305

含

含角度偏移之位置同步, 639
含迴轉線性軸之轉換, 358

形

形狀修整 - CYCLE495
    外部程式設計, 847

材

材料移除
    支援功能, 745
材料移除－CYCLE951
    外部程式設計, 882

沖

沖孔, 677

系

系統
    -相關系統中是否可用, 5
系統框架, 341
系統變數
    探針狀態, 307
    探針限制, 307

使

使用者 XML, 709

具

具有定向功能的刀把, 493
具有路徑規格和參數的子程式, 219

初

初始刀具方向設定 ORIRESET, 361
初始化
    陣列的, 56
初始化程式, 245

刻

刻字－CYCLE60
    外部程式設計, 786

取

取得及尋找無法追蹤的區段, 583

坯

坯料定義, 737

定

定位屬性
    間接程式設計, 76
定址, 238

底

底切－CYCLE940
外部程式設計, 879

所

所有角的轉角減速, 311

直

直角座標 PTP 移動, 406
直線位置模式－HOLES1
外部程式設計, 766

非

非同步振盪, 663

保

保護區, 249

相

相對於路徑的刀具方向, 385

計

計數迴圈, 127

振

振盪
同步震盪, 669
非同步, 663
非同步振盪, 663
透過同步動作控制, 669
部分進給, 671

栓

栓槽
插補, 273
類型, 280

框

框架
- 鏈結, 346
NCU 全域, 340
系統, 341
呼叫, 325
指定, 326
框架鏈結, 326
通道專屬, 341
框架元件
FI, 324
MI, 324
RT, 324
SC, 324
TR, 324
框架變數
呼叫座標轉換, 315
指派值, 322
預先定義的框架變數, 317

矩

矩形島嶼－CYCLE76
外部程式設計, 804
矩形腔－POCKET3
外部程式設計, 769

記

記憶體
使用, 245
程式, 233
預處理, 576

陣

陣列, 56
元件, 56
定義, 56
陣列索引, 59

高

高速加工設定－CYCLE832
外部程式設計, 865

偏

偏移記憶體, 447

停

停止
NC 控制, 733
自主驅動, 734

動

動力學
解析, 497
動力學類型, 497
動作結尾條件
可程式設計的, 312

參

參數
在子程式呼叫時傳輸, 178
形式, 177
供子程式呼叫所用的傳輸, 210
實際, 178
機台, 447

執

執行時期
檢查結構的回應, 125

巢

巢狀深度
檢查結構的, 124

從

從輪廓快速退刀, 144

斜

斜向直面進給研磨, 401

旋

旋轉
方向向量, 382
旋轉－CYCLE800
外部程式設計, 849
旋轉向量的插補, 382
旋轉軸
方向向量, 493
旋轉角度, 493
距離向量, 493

深

深孔鑽 2－CYCLE830
外部程式設計, 859
深孔鑽－CYCLE83
外部程式設計, 819

細

細長孔洞－細長孔洞
外部程式設計, 783

處

處理時間, 716
處理資料共享, 721

設

設定修整器座標系統 - CYCLE435
外部程式設計, 846
設定值, 452
設定點值耦合, 642

速

速度同步, 639
速度耦合, 642

連

連結
變數, 29

剩

剩餘時間
對於工件, 718

單

單一位置, 370
單一單節
停用, 186
單節顯示
停用, 191

幾

幾何軸
切換, 694

循

循環警報, 726

插

插入深度, 470

無

無條件進位, 173
無窮迴圈, 127

程

程式
分支, 114
初始化, 245
定址, 238
重覆, 213
記憶體, 235
執行時間, 715
跳躍, 111
程式記憶體
標準目錄, 234
檔案類型, 234
程式迴圈
IF 迴圈, 125
REPEAT 迴圈, 130
WHILE 迴圈, 129
計數迴圈, 127
迴圈結尾, 127
程式區段
-重覆, 117
程式區段重複
具有間接程式設計 CALL, 217
程式單節結束 - GROUP_END
外部程式設計, 914
程式單節開始處 - GROUP_BEGIN, 913

軸

軸
替換, 150
耦合動作, 605
軸主動值耦合, 626

量

量測作業狀態, 308

間

間接程式設計
G 碼的, 74
工件程式行, 79
位址的, 72
定位屬性, 76

傾

傾斜角轉換 (TRAANG)
使用可程式設計角度, 400

圓

圓位置模式－HOLES2
外部程式設計, 766
圓弧島嶼－CYCLE77
外部程式設計, 806
圓弧腔－POCKET4
外部程式設計, 773
圓弧資料
計算, 760
圓柱表面轉換, 351

搜

搜尋路徑
可程式設計搜尋路徑, 220
對子程式呼叫, 242

搭

搭配補償夾頭進行攻牙—CYCLE840
外部程式設計, 869

極

極轉換, 351

溝

溝槽－CYCLE930
外部程式設計, 875

資

資料層級, 54

跟

跟隨軸
用於軸主動值耦合, 626

路

路徑規格, 239
路徑銑削－CYCLE72
外部程式設計, 800

跳

跳躍
至程式開始處, 110
至跳躍標籤, 111
跳躍標記
程式區段重複, 117
跳躍標籤
程式跳躍, 112

零

零點偏移量
外部, 330

電

電子齒輪, 632

預

預處理
－記憶體, 576
預鑽挖槽輪廓－CYCLE64
外部程式設計, 796

實

實際值耦合, 642

磁

磁碟名稱, 239

算

算術參數
全域, 27
通道專屬, 26

網

網格 / 框架位置模式－CYCLE801
外部程式設計, 853

語

語言模式, 742

鉸

鉸孔－CYCLE85
外部程式設計, 827

銑

銑削刀尖, 470
銑削刀具加工點, 470
銑削刀具參考點, 470
銑削開孔槽－CYCLE899
　外部程式設計, 872
銑削輪廓挖槽－CYCLE63
　外部程式設計, 792

暫

暫停單節, 583

標

標籤, 117

線

線上刀長偏移量, 502

耦

耦合
　一般, 650
耦合狀態
　用於軸主動值耦合, 631
　耦合動作期間, 606
耦合係數, 603
耦合動作, 603
耦合動作軸, 605
耦合軸組合, 603
耦合類型, 642

輪

輪廓
　table, 746
　重新定位, 584
　-準備, 746
　-編碼, 753
輪廓元素
　移動, 759
輪廓切削－CYCLE95
　外部程式設計, 832
輪廓切槽－CYCLE952
　外部程式設計, 885
輪廓呼叫－CYCLE62
　外部程式設計, 791
輪廓準備
　錯誤反饋訊號, 762
輪廓準確度
　可程式設計的, 574

震

震動
　限制, 567
　偏移量, 593
震盪動作
　反轉範圍, 671
　反轉點, 671
　於反轉點的進給, 673

操

操作模式
　在量測過程中, 305

磨

磨損值, 452

輸

輸出
　至外部裝置/檔案, 721

檔

檔案
　-資訊, 170
檔案名稱, 241

環

環形槽－SLOT2
　外部程式設計, 780

縱

縱向溝槽－SLOT1
　外部程式設計, 777

螺

螺紋車削－CYCLE99
外部程式設計, 841
螺紋銑削－CYCLE70
外部程式設計, 797
螺紋鍊－CYCLE98
外部程式設計, 834

點

點對點移動, 406

簡

簡易 XML, 709

轉

轉換
三、四及五軸轉換, 359
不管動力學，初始刀具方向設定, 348
方向轉換, 347
序連, 404
動態轉換, 348
鏈結轉換, 349
轉換的限制, 419
轉換類型
一般功能, 347

鏜

鏜孔－CYCLE86
外部程式設計, 829

觸

觸發事件
在量測過程中, 304

警

警報
在 NC 程式中設定, 726

變

變數
使用者定義, 31
類型轉換, 90, 92

邏

邏輯運算子, 83

鑽

鑽孔－CYCLE82
外部程式設計, 816
鑽孔螺紋銑削－CYCLE78
外部程式設計, 809