SIEMENS

SINUMERIK

SINUMERIK 840D sl/ SINUMERIK 828D Preparación del trabajo

Manual de programación

Prólogo

Programación flexible de CN 1

Administración de programas y ficheros 2

Zonas protegidas 3

Órdenes de desplazamiento especiales 4

Transformadas de coordenadas (FRAMES) 5

Transformadas 6

Correcciones de herramientas 7

Comportamiento de contorneado 8

Acoplamientos de ejes 9

Acciones síncronas a desplazamiento 10

Vaivén 11

Troquelado y punzonado 12

Rectificado 13

Funciones adicionales 14

Programas de desbaste propios 15

Tablas 16

Anexo A

Válidas para

Control
SINUMERIK 840D sl/840DE sl
SINUMERIK 828D

Software	*Versión*
Software de sistema NCU para 840D sl/840DE sl	2.6
Software de sistema NCU para 828D	2.6

06/2009
6FC5398-2BP20-0EA0

Notas jurídicas

Filosofía en la señalización de advertencias y peligros

Este manual contiene las informaciones necesarias para la seguridad personal así como para la prevención de daños materiales. Las informaciones para su seguridad personal están resaltadas con un triángulo de advertencia; las informaciones para evitar únicamente daños materiales no llevan dicho triángulo. De acuerdo al grado de peligro las consignas se representan, de mayor a menor peligro, como sigue.

⚠PELIGRO

Significa que, si no se adoptan las medidas preventivas adecuadas **se producirá** la muerte, o bien lesiones corporales graves.

⚠ADVERTENCIA

Significa que, si no se adoptan las medidas preventivas adecuadas **puede producirse** la muerte o bien lesiones corporales graves.

⚠PRECAUCIÓN

Con triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales.

PRECAUCIÓN

Sin triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse daños materiales.

ATENCIÓN

Significa que puede producirse un resultado o estado no deseado si no se respeta la consigna de seguridad correspondiente.

Si se dan varios niveles de peligro se usa siempre la consigna de seguridad más estricta en cada caso. Si en una consigna de seguridad con triángulo de advertencia se alarma de posibles daños personales, la misma consigna puede contener también una advertencia sobre posibles daños materiales.

Personal cualificado

El equipo/sistema correspondiente sólo deberá instalarse y operarse respetando lo especificado en este documento. Sólo está autorizado a intervenir en este equipo el **personal cualificado**. En el sentido del manual se trata de personas que disponen de los conocimientos técnicos necesarios para poner en funcionamiento, conectar a tierra y marcar los aparatos, sistemas y circuitos de acuerdo con las normas estándar de seguridad.

Uso previsto o de los productos de Siemens

Considere lo siguiente:

⚠ADVERTENCIA

Los productos de Siemens sólo deberán usarse para los casos de aplicación previstos en el catálogo y la documentación técnica asociada. De usarse productos y componentes de terceros, éstos deberán haber sido recomendados u homologados por Siemens. El funcionamiento correcto y seguro de los productos exige que su transporte, almacenamiento, instalación, montaje, manejo y mantenimiento hayan sido realizados de forma correcta. Es preciso respetar las condiciones ambientales permitidas. También deberán seguirse las indicaciones y advertencias que figuran en la documentación asociada.

Marcas registradas

Todos los nombres marcados con ® son marcas registradas de Siemens AG. Los restantes nombres y designaciones contenidos en el presente documento pueden ser marcas registradas cuya utilización por terceros para sus propios fines puede violar los derechos de sus titulares.

Exención de responsabilidad

Hemos comprobado la concordancia del contenido de esta publicación con el hardware y el software descritos. Sin embargo, como es imposible excluir desviaciones, no podemos hacernos responsable de la plena concordancia. El contenido de esta publicación se revisa periódicamente; si es necesario, las posibles las correcciones se incluyen en la siguiente edición.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
ALEMANIA

Referencia del documento: 6FC5398-2BP20-0EA0
℗ 06/2009

Copyright © Siemens AG 2009.
Sujeto a cambios sin previo aviso

Prólogo

Documentación SINUMERIK

La documentación SINUMERIK se estructura en 3 categorías:

- Documentación general
- Documentación para el usuario
- Documentación para el fabricante/servicio

El enlace http://www.siemens.com/motioncontrol/docu le facilita información relativa a los siguientes temas:

- Pedidos de documentación
  Aquí se adjunta una lista actual de impresos.
- Descarga de documentación
  Enlaces para descarga de archivos desde Service & Support (soporte y servicio técnico).
- Búsqueda de documentación online
  Información sobre DOConCD y acceso directo a la lista de impresos en DOConWEB.
- Para diseñar de manera personalizada documentación basada en los contenidos propios de Siemens con My Documentation Manager (MDM), ver http://www.siemens.com/mdm.

  My Documentation Manager le ofrece una serie de funciones que le permitirá elaborar la documentación de la máquina de forma personalizada.
- Formación y FAQs

  Una información más detallada sobre la oferta de formación y las FAQ (preguntas frecuentes) navegando por la página.

Destinatarios

La presente documentación está destinada a:

- Programadores
- Proyectistas

Finalidad

El Manual de programación capacita a los destinatarios para diseñar, escribir y probar programas e interfaces de software y para resolver errores.

Alcance estándar

Las presentes instrucciones de programación contienen una descripción de la funcionalidad estándar. Los suplementos o las modificaciones realizados por el fabricante de la máquina son documentadas por el mismo.

En el control pueden ejecutarse otras funciones adicionales no descritas en la presente documentación. Sin embargo, no se pueden reclamar por derecho estas funciones en nuevos suministros o en intervenciones de mantenimiento.

Asimismo, por razones de claridad expositiva, esta documentación no detalla toda la información relativa a las variantes completas del producto descrito ni tampoco puede considerar todos los casos imaginables de instalación, de explotación ni de mantenimiento.

Technical Support

Si desea hacer algún tipo de consulta, diríjase a la siguiente hotline:

	Europa/África
Teléfono	+49 180 5050 - 222
Fax	+49 180 5050 - 223
0,14 €/minuto llamando desde la red de telefonía fija de Alemania, la tarifa de telefonía móvil puede diferir.	
Internet	http://www.siemens.de/automation/support-request

	América
Teléfono	+1 423 262 2522
Fax	+1 423 262 2200
E-mail:	mailto:techsupport.sea@siemens.com

	Asia/Pacífico
Teléfono	+86 1064 757575
Fax	+86 1064 747474
E-Mail	mailto:support.asia.automation@siemens.com

Nota

Los números de teléfono específicos de cada país para el asesoramiento técnico se encuentran en Internet:
http://www.automation.siemens.com/partner

Consultas con respecto a la documentación

Para cualquier consulta con respecto a la documentación (sugerencias, correcciones), sírvase enviar un fax o un e-mail a la siguiente dirección:

Fax:	+49 9131- 98 2176
Correo electrónico:	mailto:docu.motioncontrol@siemens.com

En el anexo de este documento encontrará una plantilla de fax.

Dirección de Internet para SINUMERIK

http://www.siemens.com/sinumerik

Manual de programación "Fundamentos" y "Preparación del trabajo"

Las descripciones sobre la programación CN están divididas en dos manuales:

1. **Fundamentos**

   El manual de programación "Fundamentos" está destinado al maquinista cualificado y presupone los correspondientes conocimientos en mecanizados de taladrado, fresado y torneado. Mediante ejemplos de programación sencillos se explican también los comandos e instrucciones conocidos según DIN 66025.

2. **Preparación del trabajo**

   El manual de programación "Preparación del trabajo" está destinado al técnico ya familiarizado con todas las posibilidades de programación. El control SINUMERIK permite, con un lenguaje de programación especial, la programación de un programa de pieza complejo (p. ej.: superficies de forma libre, coordinación de canales, etc.) y facilita la laboriosa programación del técnico.

Disponibilidad de los elementos de lenguaje CN descritos

Todos los elementos de lenguaje CN descritos en el presente documento están disponibles para SINUMERIK 840D sl. Consulte la disponibilidad para SINUMERIK 828D en la columna "828D" de la "Lista de instrucciones (Página 753)".

Sumario

Prólogo **3**

1 Programación flexible de CN **15**

1.1 Variables 15
1.1.1 Información general acerca de las variables 15
1.1.2 Variables del sistema 16
1.1.3 Variables de usuario predefinidas: Parámetros de cálculo R 19
1.1.4 Variables de usuario predefinidas: Variables de lincado 21
1.1.5 Definición de variables de usuario (DEF) 24
1.1.6 Redefinición de variables de sistema, variables de usuario e instrucciones de programación del CN (REDEF) 31
1.1.7 Atributo: Valor de inicialización 34
1.1.8 Atributo: Valores límite (LLI, ULI) 37
1.1.9 Atributo: Unidad física (PHU) 39
1.1.10 Atributo: Derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) 41
1.1.11 Vista general de atributos definibles y redefinibles 46
1.1.12 Definición e inicialización de variables de matriz (DEF, SET, REP) 47
1.1.13 Tipos de datos 54
1.2 Programación indirecta 55
1.2.1 Programación indirecta de códigos G 58
1.2.2 Programación indirecta de atributos de posición (GP) 59
1.2.3 Programación indirecta de líneas de programa de pieza (EXECSTRING) 62
1.3 Funciones de cálculo 63
1.4 Operaciones de comparación y operaciones lógicas 66
1.5 Corrección de precisión en caso de errores de comparación (TRUNC) 68
1.6 Mínimo, máximo y rango de variables (MINVAL, MAXVAL, BOUND) 70
1.7 Prioridad de los operaciones 72
1.8 Conversiones de tipos posibles 73
1.9 Operaciones con cadenas de caracteres 74
1.9.1 Conversión de tipos a STRING (AXSTRING) 74
1.9.2 Conversión de tipos de STRING (NUMBER, ISNUMBER, AXNAME) 75
1.9.3 Concatenación de cadenas (<<) 77
1.9.4 Conversión minúsculas/mayúsculas (TOLOWER, TOUPPER) 78
1.9.5 Definir la longitud de una cadena (STRLEN) 79
1.9.6 Búsqueda de caracteres/cadena en una cadena (INDEX, RINDEX, MINDEX, MATCH) 80
1.9.7 Selección de una cadena parcial (SUBSTR) 82
1.9.8 Selección de un carácter único (STRINGVAR, STRINGFELD) 83
1.10 SSSaltos y derivaciones del programa 85
1.10.1 Salto atrás hasta el inicio del programa (GOTOS) 85
1.10.2 Saltos de programa a marcas de salto (GOTOB, GOTOF, GOTO, GOTOC) 86
1.10.3 Bifurcación de programa (CASE ... OF ... DEFAULT ...) 89
1.11 Repetición de secciones de programa (REPEAT, REPEATB, ENDLABEL, P) 92
1.12 Estructuras de control 100
1.12.1 Bucle de programa con alternativa (IF, ELSE, ENDIF) 101
1.12.2 Bucle de programa sin fin (LOOP, ENDLOOP) 103

1.12.3 Bucle contador (FOR ... TO ..., ENDFOR) ... 104
1.12.4 Bucle de programa con la condición en el inicio del bucle (WHILE, ENDWHILE) ... 106
1.12.5 Bucle de programa con la condición al final del bucle (REPEAT, UNTIL) ... 107
1.12.6 Ejemplo de programa con estructuras de control imbricadas ... 108
1.13 Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) ... 109
1.14 Rutina de interrupción (ASUP) ... 115
1.14.1 Función de una rutina de interrupción ... 115
1.14.2 Creación de una rutina de interrupción ... 116
1.14.3 Asignar e iniciar una rutina de interrupción (SETINT, PRIO, BLSYNC) ... 117
1.14.4 Desactivación/reactivación de la asignación de una rutina de interrupción (DISABLE, ENABLE) ... 119
1.14.5 Borrado de la asignación de una rutina de interrupción (CLRINT) ... 120
1.14.6 Retirada rápida del contorno (SETINT LIFTFAST, ALF) ... 121
1.14.7 Sentido de desplazamiento en la retirada rápida del contorno ... 124
1.14.8 Desarrollo del movimiento en rutinas de interrupción ... 127
1.15 Intercambio de ejes, intercambio de cabezales (RELEASE, GET, GETD) ... 128
1.16 Transferir el eje a otro canal (AXTOCHAN) ... 133
1.17 Activar los datos de máquina (NEWCONF) ... 135
1.18 Escribir fichero (WRITE) ... 136
1.19 Borrar fichero (DELETE) ... 139
1.20 Leer líneas en fichero (READ) ... 141
1.21 Comprobar la presencia de un fichero (ISFILE) ... 144
1.22 Leer información de fichero (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) ... 146
1.23 Cálculo de la suma de control mediante una matriz (CHECKSUM) ... 149
1.24 Redondeo (ROUNDUP) ... 151
1.25 Uso de subprogramas ... 152
1.25.1 Generalidades ... 152
1.25.1.1 Subprograma ... 152
1.25.1.2 Nombres de subprogramas ... 153
1.25.1.3 Imbricación de subprogramas ... 154
1.25.1.4 Ruta de búsqueda ... 155
1.25.1.5 Parámetros formales y actuales ... 156
1.25.1.6 Transferencia de parámetros ... 157
1.25.2 Definición de un subprograma ... 159
1.25.2.1 Subprograma sin transferencia de parámetros ... 159
1.25.2.2 Subprograma con transferencia de parámetros Call-By-Value (PROC) ... 160
1.25.2.3 Subprograma con transferencia de parámetros Call-By-Reference (PROC, VAR) ... 162
1.25.2.4 Almacenamiento de las funciones G modales (SAVE) ... 164
1.25.2.5 Suprimir la ejecución de secuencia individual (SBLOF, SBLON) ... 166
1.25.2.6 Suprimir la visualización de secuencia actual (DISPLOF, DISPLON, ACTBLOCNO) ... 172
1.25.2.7 Identificación de subprogramas con preparación (PREPRO) ... 175
1.25.2.8 Salto atrás al subprograma M17 ... 176
1.25.2.9 Salto atrás al subprograma RET ... 177
1.25.2.10 Salto atrás parametrizable al subprograma (RET ...) ... 178
1.25.3 Llamada de un subprograma ... 185
1.25.3.1 Llamada de subprograma sin transferencia de parámetros ... 185
1.25.3.2 Llamada de subprograma con transferencia de parámetros (EXTERN) ... 187
1.25.3.3 Cantidad de repeticiones de programa (P) ... 189
1.25.3.4 Llamada de subprograma modal (MCALL) ... 191
1.25.3.5 Llamada de subprograma indirecta (CALL) ... 193

1.25.3.6 Llamada de programa indirecta con indicación de la sección de programa que se va a ejecutar (CALL BLOCK ... TO ...) ... 194
1.25.3.7 Llamada indirecta a un programa creado en lenguaje ISO (ISOCALL) ... 196
1.25.3.8 Llamada de subprograma con ruta de acceso y parámetros (PCALL) ... 197
1.25.3.9 Ampliación de la ruta de búsqueda en la llamada de subprograma (CALLPATH) ... 198
1.25.3.10 Ejecutar subprograma externo (EXTCALL) ... 200
1.25.4 Ciclos ... 204
1.25.4.1 Ciclos: Parametrizar ciclos de usuario ... 204
1.26 Técnica de macros (DEFINE ... AS) ... 208
2 Administración de programas y ficheros ... 211
2.1 Memoria de programas ... 211
2.2 Memoria de trabajo (CHANDATA, COMPLETE, INITIAL) ... 216
2.3 Instrucción de estructuración en el editor Step (SEFORM) ... 219
3 Zonas protegidas ... 221
3.1 Definición de las zonas protegidas (CPROTDEF, NPROTDEF) ... 221
3.2 Activar/desactivar zonas protegidas (CPROT, NPROT) ... 225
3.3 Comprobación con respecto a vulneraciones de zonas protegidas, limitación del campo de trabajo y límites de software (CALCPOSI) ... 229
4 Órdenes de desplazamiento especiales ... 237
4.1 Desplazamiento a posiciones codificadas (CAC, CIC, CDC, CACP, CACN) ... 237
4.2 Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) ... 239
4.3 Conjunto spline (SPLINEPATH) ... 252
4.4 Compresión de secuencias CN (COMPON, COMPCURV, COMPCAD, COMPOF) ... 254
4.5 Interpolación polinómica (POLY, POLYPATH, PO, PL) ... 257
4.6 Referencia de trayectoria ajustable (SPATH, UPATH) ... 264
4.7 Medida con palpador de contacto (MEAS, MEAW) ... 267
4.8 Función de medida ampliada (MEASA, MEAWA, MEAC) (opcional) ... 270
4.9 Funciones especiales para el usuario de OEM (OEMIPO1, OEMIPO2, G810 bis G829) ... 281
4.10 Reducción del avance con deceleración en los dos vértices (FENDNORM, G62, G621) ... 282
4.11 Criterio de fin del movimiento programable (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) ... 283
4.12 Secuencia de parámetros servo programable (SCPARA) ... 287
5 Transformadas de coordenadas (FRAMES) ... 289
5.1 Transformada de coordenadas a través de variables frame ... 289
5.1.1 Variables frame predefinidas ($P_BFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME) ... 291
5.2 Asignar valores a variables frame/frames ... 297
5.2.1 Asignar valores directos (valor de eje, ángulo, escala) ... 297
5.2.2 Leer y modificar componentes de frame (TR, FI, RT, SC, MI) ... 300
5.2.3 Combinación de frames completos ... 302
5.2.4 Definición de frames nuevos (DEF FRAME) ... 303
5.3 Decalaje basto y fino (CFINE, CTRANS) ... 304
5.4 Decalaje de origen externo ... 306
5.5 Decalaje de Preset (PRESETON) ... 307

5.6 Cálculo de frame a partir de 3 puntos de medida en el espacio (MEAFRAME) ... 309
5.7 Frames globales NCU ... 313
5.7.1 Frames específicos del canal ($P_CHBFR, $P_UBFR) ... 314
5.7.2 Frames activos en el canal ... 315

6 Transformadas ... 321

6.1 Programación general de los tipos de transformada ... 321
6.1.1 Movimientos de orientación en las transformadas ... 324
6.1.2 Resumen de la transformada de orientación TRAORI ... 328
6.2 Transformada de tres, cuatro y cinco ejes (TRAORI) ... 330
6.2.1 Relaciones generales, cabezal de herramienta cardánico ... 330
6.2.2 Transformada de tres, cuatro y cinco ejes (TRAORI) ... 333
6.2.3 Variantes de la programación de la orientación y el estado inicial (ORIRESET) ... 334
6.2.4 Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) ... 337
6.2.5 Fresado frontal (fresado 3D A4, B4, C4, A5, B5, C5) ... 345
6.2.6 Referencia de los ejes de orientación (ORIWKS, ORIMKS) ... 347
6.2.7 Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) ... 349
6.2.8 Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) ... 352
6.2.9 Especificación de orientación de dos puntos de contacto (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=) ... 356
6.3 Polinomios de orientación (PO[ángulo], PO[coordenada]) ... 358
6.4 Giros de la orientación de la herramienta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) ... 360
6.5 Orientaciones relativas a la trayectoria ... 363
6.5.1 Tipos de orientación relativos a la trayectoria ... 363
6.5.2 Giro relativo a la trayectoria de la orientación de herramienta (ORIPATH, ORIPATHS, ángulo de giro) ... 365
6.5.3 Interpolación relativa a la trayectoria del giro de herramienta (ORIROTC, THETA) ... 366
6.5.4 Alisamiento del desarrollo de la orientación (ORIPATHS A8=, B8=, C8=) ... 369
6.6 Compresión de la orientación (COMPON, COMPCURV, COMPCAD) ... 370
6.7 Alisado del desarrollo de la orientación (ORISON, ORISOF) ... 374
6.8 Transformación cinemática ... 376
6.8.1 Fresado en piezas torneadas (TRANSMIT) ... 376
6.8.2 Transformada de la envolvente del cilindro (TRACYL) ... 379
6.8.3 Eje inclinado (TRAANG) ... 389
6.8.4 Programar eje inclinado (G05, G07) ... 393
6.9 Desplazamiento PTP cartesiano ... 395
6.9.1 PTP con TRANSMIT ... 401
6.10 Limitaciones en la selección de una transformada ... 405
6.11 Cancelar transformada (TRAFOOF) ... 407
6.12 Transformadas concatenadas (TRACON, TRAFOOF) ... 408

7 Correcciones de herramientas ... 411

7.1 Memoria de corrección ... 411
7.2 Correcciones aditivas ... 415
7.2.1 Seleccionar correcciones aditivas (DL) ... 415
7.2.2 Definir valores de desgaste y de ajuste ($TC_SCPxy[t,d], $TC_ECPxy[t,d]) ... 417
7.2.3 Borrar correcciones aditivas (DELDL) ... 418
7.3 Acciones especiales de corrección de herramienta ... 419

7.3.1 Simetrizar longitudes de herramienta 421
7.3.2 Evaluación de signos de desgaste 422
7.3.3 Sistema de coordenadas del mecanizado activo (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)....423
7.3.4 Longitud de herramienta y cambio de plano....426
7.4 Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) 427
7.5 Activación correcciones de herramienta 3D (CUT3DC..., CUT3DF...)....433
7.5.1 Activar correcciones de herramientas 3D (CUT3DC, CUT3DF, CUT3DFS, CUT3DFF, ISD) 433
7.5.2 Corrección de herramienta 3D: Fresado periférico, fresado frontal 435
7.5.3 Corrección de herramienta 3D: Formas y datos de herramientas para el fresado frontal 437
7.5.4 Corrección de herramienta 3D: Corrección sobre la trayectoria, curvatura de la trayectoria, profundidad de penetración (CUT3DC, ISD) 438
7.5.5 Corrección de herramienta 3D: Esquinas interiores/exteriores y método de punto de intersección (G450/G451)....441
7.5.6 Corrección de herramienta 3D: Fresado periférico 3D con superficies de limitación....443
7.5.7 Corrección de herramienta 3D: Consideración de una superficie de limitación (CUT3DCC, CUT3DCCD)....443
7.6 Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)....447
7.7 Asignación libre de números D, número de filo....454
7.7.1 Asignación libre de números D, número de filo (dirección CE)....454
7.7.2 Asignación libre de números D: Comprobar números D (CHKDNO)....455
7.7.3 Asignación libre de números D: Cambiar nombre de números D (GETDNO, SETDNO) 456
7.7.4 Asignación libre de números D: Determinar el número T para el número D predeterminado (GETACTTD)....457
7.7.5 Asignación libre de números D: Fijación como no válidos de los números D (DZERO)....457
7.8 Cinemática del portaherramientas....458
7.9 Corrección longitudinal de herramienta para portaherramientas orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) 464
7.10 Corrección de longitud de herramienta online (TOFFON, TOFFOF) 468
7.11 Modificación de datos de filos en herramientas giratorias (CUTMOD) 471
8 Comportamiento de contorneado 477
8.1 Control tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) 477
8.2 Variación del avance (FNORM, FLIN, FCUB, FPO)....484
8.3 Ejecución del programa con memoria de pretratamiento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) 490
8.4 Secciones de programa con posibilidad de interrupción condicionada (DELAYFSTON, DELAYFSTOF) 493
8.5 Evitar punto de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK) 499
8.6 Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN)....502
8.7 Influencia en el guiado de movimiento 512
8.7.1 Corrección porcentual de sacudidas (JERKLIM)....512
8.7.2 Corrección porcentual de la velocidad (VELOLIM)....513
8.7.3 Ejemplo de programa para JERKLIM y VELOLIM....514
8.8 Tolerancia de orientación/contorno programable (CTOL, OTOL, ATOL)....515

9 Acoplamientos de ejes 521
9.1 Arrastre de ejes (TRAILON, TRAILOF) 521
9.2 Tablas de levas (CTAB) 526
9.2.1 Definición de tablas de levas (CTABDEF, CATBEND) 527
9.2.2 Comprobación de la existencia de una tabla de levas (CTABEXISTS) 535
9.2.3 Borrado de tablas de levas (CTABDEL) 535
9.2.4 Bloqueo de las tablas de levas contra borrado y sobrescritura (CTABLOCK, CTABUNLOCK) 537
9.2.5 Tablas de levas: Determinación de las propiedades de la tabla (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) 539
9.2.6 Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) 540
9.2.7 Tablas de levas: Comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) 546
9.3 Acoplamiento entre ejes de valores maestros (LEADON, LEADOF) 548
9.4 Reductor electrónico (EG) 554
9.4.1 Definición del reductor electrónico (EGDEF) 554
9.4.2 Activación del reductor electrónico (EGON, EGONSYN, EGONSYNE) 556
9.4.3 Desactivación del reductor electrónico (EGOFS, EGOFC) 559
9.4.4 Borrar la definición de un reductor electrónico (EGDEL) 561
9.4.5 Avance por vuelta (G95)/Reductor electrónico (FPR) 561
9.5 Cabezal síncrono 562
9.5.1 Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) 563
9.6 Conjunto maestro-esclavo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) 576
10 Acciones síncronas a desplazamiento 581
10.1 Fundamentos 581
10.1.1 Ámbito de validez y secuencia de ejecución (ID, IDS) 583
10.1.2 Comprobación cíclica de la condición (WHEN, WHENEVER, FROM, EVERY) 585
10.1.3 Acciones (DO) 587
10.2 Operadores para condiciones y acciones 589
10.3 Variables de proceso principal para acciones síncronas 591
10.3.1 Variables del sistema 591
10.3.2 Conversión implícita de tipos 593
10.3.3 Valores de las variables GUD 594
10.3.4 Identificador de eje por defecto (NO_AXIS) 596
10.3.5 Meta para acciones síncronas ($AC_META[n]) 597
10.3.6 Parámetro de acciones síncronas ($AC_PARAM[n]) 598
10.3.7 Parámetro de cálculo ($R[n]) 599
10.3.8 Leer y escribir datos de máquina y de operador CN 600
10.3.9 Variables de temporizador ($AC_Timer[n]) 601
10.3.10 Variables FIFO ($AC_FIFO1[n] ... $AC_FIFO10[n]) 602
10.3.11 Información acerca de tipos de secuencia en el interpolador ($AC_BLOCKTYPE, $AC_BLOCKTYPEINFO, $AC_SPLITBLOCK) 604
10.4 Acciones en instrucciones síncronas 607
10.4.1 Lista de las posibles acciones en acciones síncronas 607
10.4.2 Emisión de funciones auxiliares 609
10.4.3 Activar bloqueo de lectura (RDISABLE) 610
10.4.4 Desactivar la parada de decodificación previa (STOPREOF) 611
10.4.5 Borrado de trayecto residual (DELDTG) 612

10.4.6 Definición de polinomios (FCTDEF) 614
10.4.7 Función de sincronización (SYNFCT) 616
10.4.8 Regulación de distancia con corrección limitada ($AA_OFF_MODE) 620
10.4.9 Corrector de herramienta online (FTOC) 622
10.4.10 Corrección longitudinal de herramienta Online ($AA_TOFF) 625
10.4.11 Desplazamientos de posicionado 627
10.4.12 Posicionar eje (POS) 627
10.4.13 Posición en el margen de referencia definido (POSRANGE) 629
10.4.14 Arrancar/parar eje (MOV) 630
10.4.15 Intercambio de eje (RELEASE, GET) 631
10.4.16 Avance por eje (FA) 635
10.4.17 Finales de carrera software 635
10.4.18 Coordinación de ejes 636
10.4.19 Poner valor real (PRESETON) 637
10.4.20 Movimientos de cabezales 638
10.4.21 Arrastre de ejes (TRAILON, TRAILOF) 639
10.4.22 Acoplamiento de valores maestros (LEADON, LEADOF) 641
10.4.23 Medir (MEAWA, MEAC) 644
10.4.24 Inicialización de las variables de matriz (SET, REP) 645
10.4.25 Definir/borrar metas de espera (SETM, CLEARM) 646
10.4.26 Reacciones a errores (SETAL) 646
10.4.27 Desplazamiento a tope fijo (FXS, FXST, FXSW, FOCON, FOCOF) 647
10.4.28 Determinación del ángulo de tangente de trayectoria en acciones síncronas 650
10.4.29 Determinación de la corrección actual 650
10.4.30 Evaluación de la utilización a través de tiempo necesario para las acciones síncronas 651
10.5 Ciclos tecnológicos 653
10.5.1 Variable de contexto ($P_TECCYCLE) 656
10.5.2 Parámetro Call-By-Value 657
10.5.3 Inicialización de parámetros por defecto 658
10.5.4 Control de la ejecución de ciclos tecnológicos (ICYCOF, ICYCON) 658
10.5.5 Concatenaciones en cascada de ciclos tecnológicos 659
10.5.6 Ciclos tecnológicos en acciones síncronas por secuencias 660
10.5.7 Estructuras de control (IF) 660
10.5.8 Instrucciones de salto (GOTO, GOTOF, GOTOB) 661
10.5.9 Bloquear, habilitar, resetear (LOCK, UNLOCK, RESET) 662
10.6 Borrar acción síncrona (CANCEL) 664
10.7 Comportamiento de control en determinados estados operativos 665
11 Vaivén 669
11.1 Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) 669
11.2 Vaivén controlado mediante acciones síncronas (OSCILL) 675
12 Troquelado y punzonado 685
12.1 Activación, desactivación 685
12.1.1 Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) 685
12.2 División automática en segmentos 691
12.2.1 División en ejes de contorneado 694
12.2.2 División en ejes individuales 696
13 Rectificado 699
13.1 Vigilancia de herramienta específica de rectificado desde el programa de pieza (TMON, TMOF) 699

14 Funciones adicionales 701
14.1 Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) 701
14.2 Ejes geométricos conmutables (GEOAX) 704
14.3 Contenedor de ejes (AXCTSWE, AXCTSWED) 709
14.4 Comprobar el repertorio del lenguaje CN disponible (STRINGIS) 713
14.5 Leer llamada de función ISVAR y datos de máquina con índice Array 720
14.6 Aprendizaje de características de compensación (QECLRNON, QECLRNOF) 722
14.7 Llamar interactivamente la ventana del programa de pieza (MMC) 724
14.8 Tiempo de ejecución de programa/contador de piezas 725
14.8.1 Tiempo de ejecución de programa/contador de piezas (lista) 725
14.8.2 Tiempo de ejecución del programa 725
14.8.3 Contador de piezas 729
14.9 Alarmas (SETAL) 731
15 Programas de desbaste propios 733
15.1 Funciones de apoyo para el desbaste 733
15.2 Creación de tabla para el contorno (CONTPRON) 734
15.3 Creación de tabla para el contorno codificada (CONTDCON) 742
15.4 Determinación del punto de intersección entre dos elementos de contorno (INTERSEC) 747
15.5 Retirada de los elementos de contorno de una tabla secuencia a secuencia (EXECTAB) 749
15.6 Cálculo de datos de circunferencia (CALCDAT) 750
15.7 Desactivación de la preparación del contorno (EXECUTE) 752
16 Tablas 753
16.1 Lista de instrucciones 753
A Anexo 825
A.1 Lista de las abreviaturas 825
A.2 Su opinión sobre la documentación 830
A.3 Vista general de la documentación 832
Glosario 833
Index 855

Programación flexible de CN 1

1.1 Variables

1.1.1 Información general acerca de las variables

Con la utilización de variables, especialmente en combinación con las funciones de cálculo y las estructuras de control, pueden confeccionarse programas de pieza y ciclos con una flexibilidad extrema. Para ello el sistema proporciona tres tipos distintos de variables:

- Variables del sistema

  Las variables de sistema son variables definidas en el sistema y puestas a disposición del usuario con un significado especificado fijo. El software del sistema también las lee y escribe. Ejemplo: Dato de Máquina (DM)

  El sistema especifica en la máxima medida y de forma fija a través de las propiedades el significado de una variable de sistema. No obstante, el usuario puede adaptar una pequeña parte de las propiedades redefiniéndolas. Ver "Redefinición de variables de sistema, variables de usuario e instrucciones de programación del CN (REDEF) (Página 31)".

- Variables de usuario

  Las variables de usuario son variables cuyo significado es desconocido para el sistema y que éste tampoco puede evaluar. El usuario es el único que establece el significado.

  Las variables de usuario se dividen en:

  – Variables de usuario predefinidas

    Las variables de usuario predefinidas son variables ya definidas en el sistema cuyo número sólo debe ser parametrizado por el usuario mediante datos de máquina específicos. El usuario puede adaptar en la máxima medida las propiedades de estas variables. Ver "Redefinición de variables de sistema, variables de usuario e instrucciones de programación del CN (REDEF) (Página 31)".

  – Variables definidas por el usuario

    Las variables definidas por el usuario son variables que define exclusivamente el usuario y creadas por el sistema en el momento de la ejecución. El usuario es el único que establece su número, tipo de datos, visibilidad y todas las demás propiedades.

    Ver "Definición de variables de usuario (DEF) (Página 24)".

Consulte también

Variables del sistema (Página 16)

Variables de usuario predefinidas: Parámetros de cálculo R (Página 19)

Variables de usuario predefinidas: Variables de lincado (Página 21)

1.1.2 Variables del sistema

Las variables de sistema son variables predefinidas en el sistema que, en programas de piezas y ciclos, ofrecen acceso a la parametrización actual del control y a estados de máquina, control y proceso.

Variables de decodificación previa

Se denominan variables de decodificación previa aquellas variables de sistema que se leen y se escriben en el contexto de la decodificación previa, es decir, en el momento de la interpretación de la secuencia del programa de pieza en la que está programada la variable del sistema. Las variables de decodificación previa no activan ninguna parada de decodificación previa.

Variables de proceso principal

Se denominan variables de proceso principal aquellas variables de sistema que se leen o se escriben en el contexto del proceso principal, es decir, en el momento de la ejecución de la secuencia del programa de pieza en la que está programada la variable del sistema. Las variables de proceso principal son:

- Variables de sistema que pueden programarse en acciones síncronas (lectura/escritura)
- Variables de sistema que se pueden programar en el programa de pieza y que disparan una parada de decodificación previa (lectura/escritura)
- Variables de sistema que pueden programarse en el programa de pieza y cuyo valor se determina en la decodificación previa, pero no se escribe hasta el proceso principal (de forma sincronizada con el proceso principal: sólo escritura)

Sistemática de prefijos

Para identificar las variables de sistema de una forma especial, normalmente se añade al nombre un prefijo que se compone del carácter $ seguido de una o dos letras y un guión bajo:

$ + 1ª letra	**Significado: Tipo de datos**
Variables de sistema que se leen/escriben en la decodificación previa	
$M	Datos de máquina[1)]
$S	Datos de operador, zonas protegidas[1)]
$T	Datos de gestión de herramientas
$P	Datos programados
$C	Variables de ciclos de envolvente ISO
$O	Datos de opciones
R	Parámetros R (parámetros de cálculo)[2)]2
Variables de sistema que se leen/escriben en el proceso principal	
$$M	Datos de máquina[1)]
$$S	Datos de operador [1)]
$A	Datos actuales de proceso principal
$V	Datos servo
$R	Parámetros R (parámetros de cálculo) [2)]
1) Al utilizar datos de máquina y de operador en el programa de pieza/ciclo como variable de decodificación previa, se escribe el prefijo con un carácter $. Si se utilizan en acciones síncronas como variables de proceso principal, el prefijo se escribe con dos caracteres $. 2) Al utilizar un parámetro R en el programa de pieza/ciclo como variable de decodificación previa, no se escribe ningún prefijo, p. ej. R10. Si se utilizan en una acción síncrona como variable de proceso principal, se escribe un carácter $ como prefijo, p. ej. $R10.	

2ª letra	**Significado: Visibilidad**
N	Variable global de NCK (**N**CK)
C	Variable específica de canal (**C**hannel)
A	Variable específica de eje (**A**xis)

Limitaciones

Excepciones en la sistemática de prefijos
Las siguientes variables de sistema difieren de la sistemática de prefijos mencionada anteriormente:

- $TC_...: La 2ª letra C aquí no hace referencia a variables de sistema específicas de canal sino específicas del portaherramientas (TC = Tool Carrier)
- $P_ ...: Variables de sistema específicas del canal

Utilización de datos de máquina y de operador en acciones síncronas
Al utilizar datos de máquina y de operador en acciones síncronas, con el prefijo se puede determinar si el dato de máquina o de operador se lee/escribe de forma sincronizada con la decodificación previa o el proceso principal.

Si el dato permanece invariable durante el mecanizado, se puede leer de forma sincronizada con la decodificación previa. Para ello, el prefijo del dato de máquina o de operador se escribe con un carácter $:

Código de programa

```
ID=1 WHENEVER G710 $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]—6 DO $AA_OVR[X]=0
```

Si el dato varía durante el mecanizado, se debe leer/escribir de forma sincronizada con el proceso principal. Para ello, el prefijo del dato de máquina o de operador se escribe con dos caracteres $:

Código de programa

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

Nota

Escritura de datos de máquina

Al escribir un dato de máquina o de operador hay que prestar atención a que el nivel de acceso activo al ejecutar el programa de pieza/ciclo permita el acceso de escritura y que el efecto del dato sea "IMMEDIATE".

Bibliografía

La lista de las propiedades de todas las variables de sistema se encuentra en:

/PGA1/ Manual de listas de variables de sistema

Consulte también

Tipos de datos (Página 54)

1.1.3 Variables de usuario predefinidas: Parámetros de cálculo R

Función

Los parámetros de cálculo o los parámetros R son una variable de usuario predefinida con el nombre R, definida como matriz del tipo de datos `REAL`. Por motivos históricos, junto con la notación con índice de matriz p. ej. `R[10]` también se permite la notación sin índice de matriz p. ej. `R10` para los parámetros R.

Si se utilizan en acciones síncronas, debe anteponerse el carácter $, p. ej. `$R10`.

Sintaxis

En la utilización como variable de decodificación previa:

```
R<n>
R[<Expresión>]
```

En la utilización como variable de proceso principal:

```
$R<n>
$R[<Expresión>]
```

Descripción

`R`: Identificador en la utilización como variable de decodificación previa p. ej. en el programa de pieza

`$R`: Identificador en la utilización como variable de proceso principal p. ej. en acciones síncronas

Tipo: REAL

Rango de valores: En notación no exponencial:

± (0.000 0001 ... 9999 9999)

Nota:
Se permite un máximo de 8 decimales.

En notación exponencial:

± ($1*10^{-300}$ ... $1*10^{+300}$)

Atención:

- Notación: <Mantisa>EX<Exponente> p. ej. 8.2EX-3
- Se permite un máximo de 10 caracteres incluidos signo y punto decimal.

`<n>`: Número del parámetro R

Tipo: INT

Rango de valores: 0 - MAX_INDEX

Nota
MAX_INDEX se obtiene a partir de la cantidad parametrizada de parámetros R:
MAX_INDEX = (DM28050 $MN_MM_NUM_R_PARAM) - 1

`<Expresión>`: Índice de matriz

Como índice de matriz puede indicarse cualquier expresión mientras el resultado de la expresión pueda convertirse en el tipo de datos `INT` (`INT`, `REAL`, `BOOL`, `CHAR`)

Ejemplo

Asignaciones a parámetros R y utilización de parámetros R en funciones matemáticas:

```
Código de programa          Comentarios
R0=3.5678                   ;   Asignación en la decodificación previa
R[1]=-37.3                  ;   Asignación en la decodificación previa
R3=-7                       ;   Asignación en la decodificación previa
$R4=-0.1EX-5                ;   Asignación en el proceso principal:
                                R4 = -0.1 * 10^-5
$R[6]=1.874EX8              ;   Asignación en el proceso principal:
                                R6 = 1.874 * 10^8
R7=SIN(25.3)                ;   Asignación en la decodificación previa

R[R2]=R10                   ;   Direccionamiento indirecto mediante parámetros R
R[(R1+R2)*R3]=5             ;   Direccionamiento indirecto mediante expresión
                                matem.

X=(R1+R2)                   ;   Desplazar eje X a la posición resultante de la suma
                                de R1 y R2
Z=SQRT(R1*R1+R2*R2)         ;   Desplazar eje Z a la posición raíz cuadrada
                                (R1^2 + R2^2)
```

1.1.4 Variables de usuario predefinidas: Variables de lincado

Función

En el marco de la función "Lincado NCU", las variables de lincado permiten el intercambio cíclico de datos entre NCU interconectadas en una red. En ese caso posibilitan un acceso específico del formato de datos a la memoria de variables de lincado. El usuario/fabricante de la máquina establece de forma específica para la instalación la memoria de variables de lincado, tanto en lo que se refiere a tamaño como a estructura de datos.

Las variables de lincado son variables de usuario globales del sistema que todas las NCU del conjunto de lincado pueden leer y escribir en programas de pieza y ciclos durante la comunicación de lincado configurada. Al contrario que las variables globales del usuario (GUD), las variables de lincado también pueden utilizarse en acciones síncronas.

En instalaciones sin lincado NCU activo, las variables de lincado pueden utilizarse de forma local para el control, junto con las variables globales del usuario (GUD), como variables globales del usuario adicionales.

Sintaxis

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

Descripción

`$A_DLB`: Variable de lincado para el formato de datos BYTE (1 byte)

Tipo de dato: UINT

Rango de valores: 0 ... 255

`$A_DLW`: Variable de lincado para el formato de datos WORD (2 bytes)

Tipo de dato: INT

Rango de valores: -32768 ... 32767

`$A_DLD`: Variable de lincado para el formato de datos DWORD (4 bytes)

Tipo de dato: INT

Rango de valores: -2147483648 ... 2147483647

`$A_DLR`: Variable de lincado para el formato de datos REAL (8 bytes)

Tipo de dato: REAL

Rango de valores: $\pm(2{,}2*10^{-308} \ldots 1{,}8*10^{+308})$

`<Índice>`: Índice de direcciones en bytes, calculado desde el principio de la memoria de variables de lincado

Tipo de dato: INT

Rango de valores: 0 - MAX_INDEX

Nota

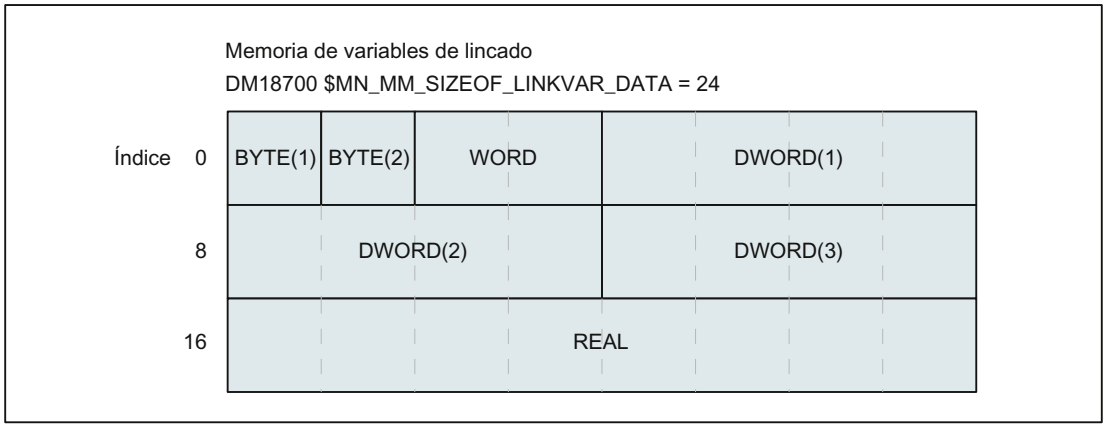
- MAX_INDEX se obtiene a partir del tamaño parametrizado de la memoria de variables de lincado:
  MAX_INDEX = (DM18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1
- Solamente pueden programarse índices para que los bytes direccionados en la memoria de variables de lincado se encuentren en un límite de formato de datos ⇒
  Índice = n * bytes, con n = 0, 1, 2, ...
  - `$A_DLB[i]`: i = 0, 1, 2, ...
  - `$A_DLW[i]`: i = 0, 2, 4, ...
  - `$A_DLD[i]`: i = 0, 4, 8, ...
  - `$A_DLR[i]`: i = 0, 8, 16, ...

Ejemplo

En la instalación de automatización hay 2 NCU (NCU1 y NCU2). A la NCU1 está conectado el eje de máquina AX2 que la NCU2 desplaza como eje lincado.

La NCU1 escribe cíclicamente el valor real de intensidad ($VA_CURR) del eje AX2 en la memoria de variables de lincado. La NCU2 lee cíclicamente el valor real de intensidad transmitido por comunicación de lincado e indica la alarma 61000 al sobrepasar el límite.

La estructura de datos en la memoria de variables de lincado se representa en la siguiente figura. El valor real de intensidad se transmite mediante el valor REAL.



NCU1

En una acción síncrona estática, la NCU1 escribe cíclicamente en el ciclo IPO el valor real de intensidad del eje AX2 mediante la variable de lincado $A_DLR[16] en la memoria de variables de lincado.

Código de programa

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

NCU2

En una acción síncrona estática, la NCU2 lee cíclicamente en el ciclo IPO el valor real de intensidad del eje AX2 mediante la variable de lincado $A_DLR[16] de la memoria de variables de lincado. Si el valor real de intensidad es mayor que 23.0 A, se indica la alarma 61000.

Código de programa

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

1.1.5 Definición de variables de usuario (DEF)

Función

Con el comando `DEF` se pueden definir variables propias y ocuparlas con valores. Para diferenciarlas de las variables del sistema, estas variables se denominan variables definidas por el usuario o variables de usuario (User Data).

De acuerdo con el ámbito de vigencia, es decir, el ámbito en el que la variable es visible, existen las siguientes categorías de variables de usuario:

- Variables locales de usuario (LUD)

  Las variables locales de usuario (LUD) son variables que están definidas en un programa de pieza que no es el programa principal en el momento de la ejecución. Se crean al llamar al programa de pieza y se borran al finalizar el programa de pieza o bien con un reset de CN. Sólo puede tenerse acceso a las LUD dentro del programa de pieza donde están definidas.

- Variables de usuario globales de programa (PUD)

  Las variables de usuario globales de programa (PUD) son variables que están definidas en un programa de pieza utilizado como programa principal. Se crean con el inicio del programa de pieza y se borran con el fin del programa de pieza o con un reset de CN. Puede tenerse acceso a las PUD en el programa principal y en todos los subprogramas.

- Variables globales de usuario (GUD)

  Las variables globales de usuario (GUD) son variables globales de CN o del canal que están definidas en un bloque de datos (SGUD, MGUD, UGUD, GUD4 ... GUD9) y que se conservan incluso después de un Power On. Puede tenerse acceso a las GUD en todos los programas de pieza.

Las variables de usuario deben haberse definido antes de su utilización (lectura/escritura). Se tienen que observar las siguientes reglas:

- Las GUD deben definirse en un fichero de definición p. ej. _N_DEF_DIR/_M_SGUD_DEF.
- Las PUD y las LUD deben definirse en la parte de definición de un programa de pieza.
- Los datos deben definirse en una secuencia propia.
- Por cada definición de datos solamente se puede utilizar un tipo de datos.
- Por cada definición de datos pueden definirse varias variables del mismo tipo de datos.

Sintaxis

```
DEF <Ámbito> <Tipo> <Parada_DP> <Instante_inic> <Unidad_fís>
<Valores límite> <Derechos de acceso>
<Nombre>[<Valor_1>,<Valor_2>,<Valor_3>]=<Valor_inic>
```

Descripción

DEF:	Comando para la definición de variables de usuario GUD, PUD, LUD	
<Ámbito>:	Ámbito de vigencia, sólo relevante para las GUD:	
	NCK:	Variable de usuario global de CN
	CHAN:	Variable de usuario global del canal
<Tipo>:	Tipo de dato:	
	INT:	Valor entero con signo
	REAL:	Número real (LONG REAL según IEEE)
	BOOL:	Valor lógico TRUE (1)/FALSE (0)
	CHAR:	Caracteres ASCII
	STRING[<LongitudMáx>]:	Cadena de caracteres de una longitud definida
	AXIS:	Identificador de cabezal/eje
	FRAME:	Datos geométricos para una transformación de coordenadas estática
	Ver "Tipos de datos (Página 54)"	
<Parada_DP>:	Parada de decodificación previa, sólo relevante para las GUD (opcional)	
	SYNR:	Parada de decodificación previa durante la lectura
	SYNW:	Parada de decodificación previa durante la escritura
	SYNRW:	Parada de decodificación previa durante la lectura/escritura
<Instante_inic>:	Instante en el que se reinicializa la variable (opcional)	
	INIPO:	PowerOn
	INIRE:	Fin del programa principal, reset de CN o PowerOn
	INICF:	NewConfig o fin del programa principal, reset de CN o PowerOn
	PRLOC:	Fin del programa principal, reset de CN tras una modificación local o PowerOn
	Ver "Atributo: Valor de inicialización (Página 34)"	
<Unidad_fís>:	Unidad física (opcional)	
	PHU <Unidad>:	
	Ver "Atributo: Unidad física (PHU) (Página 39)"	

<Valores límite>:	Valor límite inferior y superior (opcional) LLI <Valor límite>: Valor límite inferior (lower limit) ULI <Valor límite>: Valor límite superior (upper limit) Ver "Atributo: Valores límite (LLI, ULI) (Página 37)"
<Derechos de acceso>:	Derechos de acceso para la lectura/escritura de GUD mediante el programa de pieza o BTSS (opcional) APRP <Nivel de protección>: Lectura: Programa de pieza APWP <Nivel de protección>: Escritura: Programa de pieza APRB <Nivel de protección>: Lectura: BTSS APWB <Nivel de protección>: Escritura: BTSS Nivel de protección: Rango de valores: 0 ... 7 Ver "Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)"
<Nombre>:	Nombre de la variable **Nota** • 31 caracteres como máximo • Los dos primeros caracteres deben de ser una letra y/o un guión bajo. • El carácter $ está reservado para las variables de sistema y no debe utilizarse.
[<Valor_1>, <Valor_2>, <Valor_3>]:	Indicación de los tamaños de matriz para variables de matriz de 1 hasta 3 dimensiones como máximo (opcional)
<Valor_inic>:	Valor de inicialización (opcional) Ver "Atributo: Valor de inicialización (Página 34)" Para la inicialización de variables de matriz: Ver "Definición e inicialización de variables de matriz (DEF, SET, REP) (Página 47)"

Ejemplos

Ejemplo 1: Definiciones de variables de usuario en el bloque de datos para el fabricante de la máquina

Código de programa

```
%_N_MGUD_DEF                                          ; Bloque GUD: Fabricante de la máquina
$PATH=/_N_DEF_DIR
DEF CHAN REAL PHU 24 LLI 0 ULI 10 INTENSIDAD_1, INTENSIDAD_2
;  Descripción
;  Definición de dos GUD: INTENSIDAD_1, INTENSIDAD_2
;  Ámbito de vigencia: Todo el canal
;  Tipo de dato: REAL
;  Parada DP: no programada => valor por defecto = sin parada DP
;  Unidad fís.: 24 = [A]
;  Valores límite: Low = 0.0, High = 10.0
;  Derechos de acceso: no programados => valor por defecto = 7 = posición del interruptor de llave 0
;  Valor de inicialización: no programado => valor por defecto = 0.0

DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 TIEMPO_1=12, TIEMPO_2=45
;  Descripción
;  Definición de dos GUD: TIEMPO_1, TIEMPO_2
;  Ámbito de vigencia: Todo el NCK
;  Tipo de dato: REAL
;  Parada DP: no programada => valor por defecto = sin parada DP
;  Unidad fís.: 13 = [s]
;  Valores límite: Low = 10.0, High = no programado => límite superior de la zona de definición
;  Derechos de acceso:
;     Programa de pieza: escritura/lectura= 3 = usuario final
;     BTSS: escritura = 0 = Siemens, lectura= 3 = usuario final
;  Valor de inicialización: TIEMPO_1 = 12.0, TIEMPO_2 = 45.0

DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER"
;  Descripción
;  Definición de una GUD: GUD5_NAME
;  Ámbito de vigencia: Todo el NCK
;  Tipo de dato: STRING, máx. 5 caracteres
;  Parada DP: no programada => valor por defecto = sin parada DP
;  Unidad fís.: no programada => valor por defecto = 0 = ninguna unidad fís.
;  Valores límite: no programados => límites de la zona de definición: Low = 0, High = 255
;  Derechos de acceso:
;     Programa de pieza: escritura/lectura= 3 = usuario final
;     BTSS: escritura = 0 = Siemens, lectura= 3 = usuario final
;  Valor de inicialización: "COUNTER"
M30
```

Ejemplo 2: Variables de usuario locales y globales de programa (LUD/PUD)

Código de programa	**Comentarios**
PROC MAIN	; Programa principal
DEF INT VAR1	; Definición PUD
...	
SUB2	; Llamada a un subprograma
...	
M30	

Código de programa	**Comentarios**
PROC SUB2	; Subprograma SUB2
DEF INT VAR2	; DEFINICIÓN LUD
...	
IF (VAR1==1)	; Leer PUD
VAR1=VAR1+1	; Leer y escribir PUD
VAR2=1	; Escribir LUD
ENDIF	
SUB3	; Llamada a un subprograma
...	
M17	

Código de programa	**Comentarios**
PROC SUB3	; Subprograma SUB3
...	
IF (VAR1==1)	; Leer PUD
VAR1=VAR1+1	; Leer y escribir PUD
VAR2=1	; Error: LUD de SUB2 desconocido
ENDIF	
...	
M17	

Ejemplo 3: Definición y utilización de variables de usuario del tipo de datos `AXIS`

```
Código de programa                          Comentarios
DEF AXIS ABSCISA                        ;   1er eje geométrico
DEF AXIS CABEZAL                        ;   Cabezal
...
IF ISAXIS(1) == FALSE GOTOF CONTINUAR
 ABSCISA = $P_AXN1
CONTINUAR:
...
CABEZAL=(S1)                                1er cabezal
OVRA[CABEZAL]=80                        ;   Corrección del cabezal = 80%
CABEZAL=(S3)                                3er cabezal
```

Limitaciones

Variables globales de usuario (GUD)

En el contexto de la definición de variables globales de usuario (GUD) deben tenerse en cuenta los siguientes datos de máquina:

N.º	Identificador: $MN_	Significado
11140	GUD_AREA_ SAVE_TAB	Almacenamiento adicional para bloques GUD
18118	MM_NUM_GUD_MODULES	Número de ficheros GUD en el sistema de ficheros activo
18120	MM_NUM_GUD_NAMES_NCK	Número de nombres de GUD globales
18130	MM_NUM_GUD_NAMES_CHAN	Número de nombres de GUD específ. del canal
18140	MM_NUM_GUD_NAMES_AXIS	Número de nombres de GUD específ. del eje
18150	MM_GUD_VALUES_MEM	Espacio en memoria para valores de GUD globales
18660	MM_NUM_SYNACT_GUD_REAL	Número de GUD configurables del tipo de datos `REAL`
18661	MM_NUM_SYNACT_GUD_INT	Número de GUD configurables del tipo de datos `INT`
18662	MM_NUM_SYNACT_GUD_BOOL	Número de GUD configurables del tipo de datos `BOOL`
18663	MM_NUM_SYNACT_GUD_AXIS	Número de GUD configurables del tipo de datos `AXIS`
18664	MM_NUM_SYNACT_GUD_CHAR	Número de GUD configurables del tipo de datos `CHAR`
18665	MM_NUM_SYNACT_GUD_STRING	Número de GUD configurables del tipo de datos `STRING`

Variables de usuario globales de programa (PUD)

ATENCIÓN
Visibilidad de las variables de usuario locales de programa (PUD) Las variables de usuario locales de programa (PUD) definidas en el programa principal solamente estarán visibles en los subprogramas si está ajustado el siguiente dato de máquina: DM11120 $MN_LUD_EXTENDED_SCOPE = 1 Con DM11120 = 0 las variables de usuario locales de programa definidas en el programa principal sólo están visibles en el programa principal.

Utilización en todos los canales de una variable de usuario global de NCK del tipo de datos AXIS

Una variable de usuario global de NCK del tipo de datos `AXIS` que se ha inicializado con un identificador de eje en la definición en el bloque de datos solamente se podrá utilizar en diferentes canales del CN si el eje tiene el mismo número de eje de canal en esos canales.

Si no es así, debe cargarse la variable al principio del programa de pieza o bien, como en el siguiente ejemplo, debe utilizarse la función AXNAME(...) (ver "Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 701)").

```
Código de programa                          Comentarios
DEF NCK STRING[5] EJE="X"                   ; Definición en el bloque de datos

N100 AX[AXNAME(EJE)]=111 G00                ; Utilización en el programa de
                                              pieza
```

Consulte también

Redefinición de variables de sistema, variables de usuario e instrucciones de programación del CN (REDEF) (Página 31)

1.1.6 Redefinición de variables de sistema, variables de usuario e instrucciones de programación del CN (REDEF)

Función

Con el comando REDEF se pueden modificar los atributos de variables de sistema, variables de usuario e instrucciones de programación del CN. El requisito básico para la redefinición es que ésta se ejecute después de la correspondiente definición.

En una redefinición no se pueden modificar varios atributos simultáneamente. Para cada atributo que se desee modificar debe programarse una instrucción REDEF propia.

Si se programan varias modificaciones de atributos a la vez, se activa siempre la última modificación.

Atributos redefinibles

Ver "Vista general de atributos definibles y redefinibles (Página 46)"

Variables locales de usuario (PUD/LUD)

No pueden realizarse redefiniciones para variables locales de usuario (PUD/LUD).

Sintaxis

```
REDEF <Nombre> <Parada_DP>
REDEF <Nombre> <Unidad_fís.>
REDEF <Nombre> <Valores límite>
REDEF <Nombre> <Derechos de acceso>
REDEF <Nombre> <Instante_inic>
REDEF <Nombre> <Instante_inic> <Valor_inic>
```

Descripción

REDEF:	Comando para redefinir un atributo determinado de variables de sistema, variables de usuario e instrucciones de programación del CN	
<Nombre>:	Nombre de una variable ya definida o de una instrucción de programación del CN	
<Parada_DP>:	Parada de decodificación previa	
	SYNR:	Parada de decodificación previa durante la lectura
	SYNW:	Parada de decodificación previa durante la escritura
	SYNRW:	Parada de decodificación previa durante la lectura/escritura

`<Unidad_fís>`: Unidad física

`PHU` <Unidad>:

Ver "Atributo: Unidad física (PHU) (Página 39)"

Nota:
No redefinible para:

- Variables del sistema
- Datos globales del usuario (GUD)
- Tipos de datos: `BOOL`, `AXIS`, `STRING`, `FRAME`

`<Valores límite>`: Valores límite inferior y/o superior

`LLI` <Valor límite>: Valor límite inferior (lower limit)

`ULI` <Valor límite>: Valor límite superior (upper limit)

Ver "Atributo: Valores límite (LLI, ULI) (Página 37)"

Nota:
No redefinible para:

- Variables del sistema
- Datos globales del usuario (GUD)
- Tipos de datos: `BOOL`, `AXIS`, `STRING`, `FRAME`

`<Derechos de acceso>`: Derechos de acceso para la lectura/escritura mediante el programa de pieza o BTSS

`APX` <Nivel de protección>: Ejecución: Elemento de lenguaje CN

`APRP` <Nivel de protección>: Lectura: Programa de pieza

`APWP` <Nivel de protección>: Escritura: Programa de pieza

`APRB` <Nivel de protección>: Lectura: BTSS

`APWB` <Nivel de protección>: Escritura: BTSS

Nivel de protección: Rango de valores: 0 ... 7

Ver "Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)"

`<Instante_inic>`: Instante en el que se reinicializa la variable

`INIPO`: PowerOn

`INIRE`: Fin del programa principal, reset de CN o PowerOn

`INICF`: NewConfig o fin del programa principal, reset de CN o PowerOn

`PRLOC`: Fin del programa principal, reset de CN tras una modificación local o PowerOn

Ver "Atributo: Valor de inicialización (Página 34)"

<Valor_inic>: Valor de inicialización

Al redefinir el valor de inicialización también se debe indicar siempre el instante de la inicialización (ver <Instante_inic>).

Ver "Atributo: Valor de inicialización (Página 34)"

Para la inicialización de variables de matriz:

Ver "Definición e inicialización de variables de matriz (DEF, SET, REP) (Página 47)"

Nota:
No redefinible para:

- Variables de sistema, exceptuando datos de operador

Ejemplo

Redefiniciones de la variable de sistema $TC_DPC1 en el bloque de datos para fabricante de máquina

Código de programa

```
%_N_MGUD_DEF                                          ; Bloque GUD: Fabricante de la máquina
$PATH=/_N_DEF_DIR
REDEF $TC_DPC1 APWB 2 APWP 3
REDEF $TC_DPC1 PHU 21
REDEF $TC_DPC1 LLI 0 ULI 200
REDEF $TC_DPC1 INIPO (100, 101, 102, 103)
; Descripción
; Derecho de acceso de escritura: BTSS = nivel de protección 2, programa de pieza = nivel de
; protección 3
; Nota
; Si se utilizan ficheros ACCESS, la redefinición de los derechos de acceso debe desplazarse de
  _N_MGUD_DEF a _N_MACCESS_DEF
; Unidad física = [%]
; Valores límite: inferior = 0, superior = 200
; La variable de matriz se inicializa con los cuatro valores al realizar un PowerOn
M30
```

Limitaciones

Granularidad

Una redefinición hace referencia siempre a toda la variable identificada unívocamente mediante un nombre. No es posible, p. ej., asignar diferentes valores de atributo para elementos de matriz individuales en variables de matriz.

Consulte también

Definición de variables de usuario (DEF) (Página 24)

1.1.7 Atributo: Valor de inicialización

Definición (DEF) de variables de usuario

En la definición puede especificarse un valor de inicialización para las siguientes variables de usuario:

- Variables globales del usuario (GUD)
- Variables de usuario globales de programa (PUD)
- Variables locales de usuario (LUD)

Redefinición (REDEF) de variables de sistema y de usuario

En la redefinición puede especificarse un valor de inicialización para las siguientes variables:

- Datos de sistema
  - Datos del operador
- Datos de usuario
  - Parámetros R
  - Variables de acción síncrona ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD de acción síncrona (SYG_xy[ ], con x=R, I, B, A, C, S e y=S, M, U, 4, ..., 9)
  - Parámetros EPS
  - OEM de datos de herramienta
  - OEM de datos de almacén
  - Variables globales del usuario (GUD)

Instante de la reinicialización

En la redefinición se puede indicar el instante en el que se debe reinicializar la variable, es decir, el instante en el que se debe ajustar otra vez al valor de inicialización.

- `INIPO` (PowerOn)

  La variable se reinicializa con PowerOn.

- `INIRE` (Reset)

  La variable se reinicializa con reset de CN, reset GMO, fin del programa de pieza (`M02`/`M30`) o PowerOn.

- `INICF` (NewConfig)

  Con una solicitud NewConfig, la variable se reinicializa mediante HMI, un comando de programa de pieza `NEWCONFIG` o un reset de CN, reset de GMO, fin del programa de pieza (M02/M30) o PowerOn.

- PRLOC (modificación local de programa)

  Entonces la variable sólo se reinicializará con un reset de CN, un reset GMO o con el fin del programa de pieza (M02/M30) si se ha modificado en el marco del programa de pieza actual.

  El atributo PRLOC sólo se puede utilizar en relación con los datos de operador programables (ver la tabla siguiente).

Tabla 1- 1 Datos de operador programables

Número	Identificador	Comando G[1)]
42000	$SC_THREAD_START_ANGLE	SF
42010	$SC_THREAD_RAMP_DISP	DITS/DITE
42400	$SA_PUNCH_DWELLTIME	PDELAYON
42800	$SA_SPIND_ASSIGN_TAB	SETMS
43210	$SA_SPIND_MIN_VELO_G25	G25
43220	$SA_SPIND_MAX_VELO_G26	G26
43230	$SA_SPIND_MAX_VELO_LIMS	LIMS
43300	$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
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) Con este comando G se activa el dato de operador		

Limitaciones

Valor de inicialización: variables globales del usuario (GUD)

- Para que una variable global del usuario (GUD) pueda restablecerse a su valor de inicialización asignado en la definición o en la redefinición mediante `INIRE` (reset) o `INICF` (NewConfig), debe estar ajustado el siguiente dato de máquina:

  DM11270 $MN_DEFAULT_VALUES_MEM_MASK, BIT0 = 1
  Si el dato de máquina no está ajustado, las variables globales del usuario (GUD) se ajustan al valor de inicialización implícito del tipo de datos pertinente.
- Para las variables globales del usuario (GUD) con el ámbito de vigencia `NCK`, solamente se puede especificar `INIPO` (PowerOn) como instante de la inicialización.
- Para las variables globales del usuario (GUD) con el ámbito de vigencia `CHAN`, además de `INIPO` (PowerOn) se puede especificar `INIRE` (reset) o `INICF` (NewConfig) como instante de la inicialización.
- En las variables globales del usuario (GUD) con el ámbito de vigencia `CHAN` y el instante de la inicialización `INIRE` (reset) o `INICF` (NewConfig), en el caso de un reset de CN, reset de GMO y NewConfig las variables se reinicializan sólo en aquellos canales en los que se han activado los sucesos mencionados.

Valor de inicialización: tipo de datos FRAME

Para variables del tipo de datos `FRAME` no se puede indicar ningún valor de inicialización. Las variables del tipo de datos `FRAME` se inicializan siempre implícitamente con el frame por defecto.

Valor de inicialización: tipo de datos CHAR

Para las variables del tipo de datos `CHAR`, en lugar del código ASCII (0...255) también se puede programar el carácter ASCII correspondiente entre comillas, p. ej. "A".

Valor de inicialización: tipo de datos STRING

En variables del tipo de datos `STRING` la cadena de caracteres debe escribirse entre comillas, p. ej.: ...= "MÁQUINA_1".

Valor de inicialización: tipo de datos AXIS

Para variables del tipo de datos `AXIS`, en el caso de la dirección extendida el identificador de eje debe escribirse entre paréntesis, p. ej.: ...=(X3).

Valor de inicialización: variable del sistema

Para la variable del sistema no se pueden especificar valores de inicialización específicos del usuario mediante una redefinición. Los valores de inicialización de las variables del sistema están especificados en el sistema de forma fija. No obstante, mediante la redefinición se puede modificar el instante (`INIRE`, `INICF`) en el que se reinicializa la variable del sistema.

Valor de inicialización implícito: tipo de datos AXIS

Para variables del tipo de datos `AXIS` se utiliza el siguiente valor de inicialización implícito:

- Datos de sistema: "primer eje geométrico"
- GUD de acción síncrona (nombre: SYG_A*), PUD, LUD:
  identificador de eje del dato de máquina: DM20082
  $MC_AXCONF_CHANAX_DEFAULT_NAME

Valor de inicialización implícito: datos de herramienta y de almacén

Para los datos de herramienta y de almacén, los valores de inicialización pueden especificarse mediante el siguiente dato de máquina: DM17520
$MN_TOOL_DEFAULT_DATA_MASK

ATENCIÓN
Sincronización La sincronización de sucesos que desencadenan la reinicialización de una variable global con la lectura de esta variable en otro lugar es responsabilidad exclusiva del usuario/ fabricante de la máquina.

1.1.8 Atributo: Valores límite (LLI, ULI)

Los valores límite superior e inferior de la zona de definición sólo pueden especificarse para los siguientes tipos de datos:

- INT
- REAL
- CHAR

Definición (DEF) de variables de usuario: valores límite y valores de inicialización implícitos

Si al definir una variable de usuario de uno de los tipos de datos antes mencionados no se define ningún valor de inicialización explícito, la variable se ajusta al valor de inicialización implícito del tipo de datos:

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

Si el valor de inicialización implícito está fuera de la zona de definición establecida mediante los valores límite programados, la variable se inicializa con el valor límite que está más próximo al valor de inicialización implícito:

- Valor de inicialización implícito < valor límite inferior (LLI) ⇒ valor de inicialización = valor límite inferior
- Valor de inicialización implícito > valor límite superior (ULI) ⇒ valor de inicialización = valor límite superior

Ejemplos:

```
Código del programa          Comentarios
DEF REAL GUD1                ;   Valor límite inferior = límite de la zona de
                             ;   definición
                             ;   Valor límite superior = límite de la zona de
                             ;   definición
                                 Valor de inicialización =
                                 valor de inicialización implícito = 0.0
DEF REAL LLI 5.0 GUD2        ;   Valor límite inferior = 10.0
                             ;   Valor límite superior = límite de la zona de
                             ;   definición
                                 Valor de inicialización = 10.0
DEF REAL ULI –5 GUD3         ;   Valor límite inferior = límite de la zona de
                             ;   definición
                             ;   Valor límite superior = -10.0
                                 Valor de inicialización = -10.0
```

Redefinición (REDEF) de variables de usuario: Valores límite y valores reales actuales

Si al redefinir los valores límite de una variable de usuario éstos se modifican de tal manera que el valor real actual se sitúa fuera de la zona nueva de definición, se produce una alarma y los valores límite no se adoptan.

Nota

Redefinición (REDEF) de variables de usuario

Al redefinir los valores límite de una variable de usuario, hay que presentar atención a que los siguientes valores se modifiquen con coherencia:

- Valores límite
- Valor real
- Valor de inicialización al efectuar la redefinición y la reinicialización automática debido a INIPO, INIRE o INICF

1.1.9 Atributo: Unidad física (PHU)

Una unidad física solamente puede programarse para variables de los siguientes tipos de datos:

- INT
- REAL

Unidades físicas programables (PHU)

La unidad física se especifica como un número en coma fija.

Se pueden programar las siguientes unidades físicas:

<Unidad>	Descripción	Unidad física
0	Ninguna unidad física	-
1	Posición lineal o angular 1) 2)	[mm], [pulgadas], [grados]
2	Posición lineal 2)	[mm], [pulgadas]
3	Posición angular	[grados]
4	Velocidad lineal o angular 1) 2)	[mm/min], [pulgadas/min], [rpm]
5	Velocidad lineal 2)	[mm/min]
6	Velocidad angular	[rpm]
7	Aceleración lineal o angular 1) 2)	[m/s^2], [pulgadas/s^2], [r/s^2]
8	Aceleración lineal 2)	[m/s^2], [pulgadas/s^2]
9	Aceleración angular	[r/s^2]
10	Jerk lineal o angular 1) 2)	[m/s^3], [pulgadas/s^3], [r/s^3]
11	Jerk lineal 2)	[m/s^3], [pulgadas/s^3]
12	Jerk angular	[r/s^3]
13	Tiempo	[s]
14	Ganancia del regulador de posición	[16.667/s]
15	Avance por vuelta 2)	[mm/vuelta], [pulgadas/vuelta]
16	Compensación de temperatura 1) 2)	[mm], [pulgadas]
18	Fuerza	[N]
19	Masa	[kg]
20	Momento de inercia 3)	[kgm^2]
21	Porcentaje	[%]
22	Frecuencia	[Hz]
23	Tensión	[V]
24	Intensidad	[A]
25	Temperatura	[°C]
26	Ángulo	[grados]
27	Kv	[1000/min]
28	Posición lineal o angular 3)	[mm], [pulgadas], [grados]
29	Velocidad de corte 2)	[m/min], [pies/min]
30	Velocidad periférica 2)	[m/s], [pies/s]
31	Resistencia	[Ohm]

<Unidad>	Descripción	Unidad física
32	Inductancia	[mH]
33	Par 3)	[Nm]
34	Constante de par 3)	[Nm/A]
35	Ganancia del regulador de intensidad	[V/A]
36	Ganancia del regulador de velocidad 3)	[Nm/(rad*s)]
37	Velocidad de giro	[rpm]
42	Potencia	[kW]
43	Corriente, baja	[µA]
46	Par, bajo 3)	[µNm]
48	Por mil	-
49	-	[Hz/s]
65	Caudal	[l/min]
66	Presión	[bar]
67	Volumen 3)	[cm^3]
68	Ganancia del proceso 3)	[mm/(V*min)]
69	Ganancia del proceso del regulador de fuerza	[N/V]
155	Paso de rosca 3)	[mm/vuelta], [pulgadas/vuelta]
156	Cambio del paso de rosca 3)	[mm/vuelta / vuelta], [pulgadas/vuelta / vuelta]
1) La unidad física depende del tipo de eje: eje lineal o giratorio.		
2) La variable se convierte automáticamente al sistema de medida actual del CN (pulgadas/métrico).		
3) La variable **no** se convierte automáticamente al sistema de medida actual del CN (pulgadas/métrico). La conversión es responsabilidad exclusiva del usuario/fabricante de la máquina.		

Nota

Rebase de niveles por conversión de formato

El formato de almacenamiento interno para todas las variables de usuario (GUD/PUD/LUD) con unidades físicas con indicación de longitud es métrico. Una utilización excesiva de estas variables en el proceso principal del NCK, p. ej. en acciones síncronas, puede provocar un rebase del tiempo de cálculo del nivel de interpolador, alarma 4240, durante la conmutación del sistema de medida.

ATENCIÓN

Compatibilidad de unidades

Durante la utilización de variables (asignación, comparación, cálculo, etc.) no se comprueba la compatibilidad de las unidades implicadas. Si en algún caso resulta necesario realizar una conversión, ésta será responsabilidad exclusiva del usuario/fabricante de la máquina.

1.1.10 Atributo: Derechos de acceso (APR, APW, APRP, APWP, APRB, APWB)

Con los derechos de acceso se corresponden los siguientes niveles de protección que deben indicarse durante la programación:

Derecho de acceso	Nivel de protección
Contraseña del sistema	0
Contraseña del fabricante de la máquina	1
Contraseña de servicio técnico	2
Contraseña del usuario final	3
Interruptor de llave posición 3	4
Interruptor de llave posición 2	5
Interruptor de llave posición 1	6
Interruptor de llave posición 0	7

Definición (DEF) de variables de usuario

Pueden definirse derechos de acceso (APR.../APW...) para las siguientes variables:

- Datos globales del usuario (GUD)

Redefinición (REDEF) de variables de sistema y de usuario

Pueden redefinirse derechos de acceso (APR.../APW...) para las siguientes variables:

- Datos de sistema
  - Datos de máquina
  - Datos del operador
  - FRAME
  - Datos de proceso
  - Compensación de error de paso de husillo (EEC)
  - Compensación de la flexión (CEC)
  - Compensación del error de cuadrante (QEC)
  - Datos de almacén
  - Datos de herramienta
  - Zonas protegidas
  - Portaherramientas orientables
  - Cadenas cinemáticas
  - Zonas de protección 3D
  - Limitación del campo de trabajo
  - Datos de herramienta ISO

- Datos de usuario
  - Parámetros R
  - Variables de acción síncrona ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD de acción síncrona (SYG_xy[ ], con x=R, I, B, A, C, S e y=S, M, U, 4, ..., 9)
  - Parámetros EPS
  - OEM de datos de herramienta
  - OEM de datos de almacén
  - Variables globales del usuario (GUD)

Nota

Durante la redefinición el derecho de acceso puede asignarse libremente a una variable situada entre el nivel de protección más bajo, 7, y el nivel de protección propio, p. ej. 1 (fabricante de la máquina).

Redefinición (REDEF) de instrucciones de programación CN

El derecho de acceso o de ejecución (APX) puede redefinirse para las siguientes instrucciones de programación CN:

- Funciones G/condiciones de desplazamiento

  Bibliografía:

  /PG/ Manual de programación, Fundamentos; capítulo: Funciones G/condiciones de desplazamiento

- Funciones predefinidas

  Bibliografía:

  /PG/ Manual de programación, Fundamentos; capítulo: Funciones predefinidas

- Llamadas de subprograma predefinidas

  Bibliografía:

  /PG/ Manual de programación, Fundamentos; capítulo: Llamadas de subprograma predefinidas

- Instrucción DO para acciones síncronas
- Identificador de programa de ciclos

  El ciclo debe estar guardado en un directorio de ciclos y contener una instrucción PROC.

Derechos de acceso relativos a programas de pieza y ciclos (**APRP**, **APWP**)

Los diferentes derechos de acceso tienen los siguientes efectos para el acceso en un programa de pieza o ciclo:

- `APRP 0`/`APWP 0`
  - Al ejecutar el programa de pieza debe estar definida la contraseña de sistema.
  - El ciclo debe estar guardado en el directorio _N_CST_DIR (Sistema).
  - Para el directorio _N_CST_DIR, el derecho de ejecución debe estar ajustado a Sistema en el DM11160 $MN_ACCESS_EXEC_CST.
- `APRP 1`/`APWP 1` o bien `APRP 2`/`APWP 2`
  - Al ejecutar el programa de pieza debe estar definida la contraseña del fabricante de la máquina o de servicio técnico.
  - El ciclo debe estar guardado en el directorio _N_CMA_DIR (fabricante de la máquina) o _N_CST_DIR.
  - Para los directorios _N_CMA_DIR o _N_CST_DIR, los derechos de ejecución deben estar ajustados como mínimo a Fabricante de la máquina en los datos de máquina DM11161 $MN_ACCESS_EXEC_CMA o DM11160 $MN_ACCESS_EXEC_CST.
- `APRP 3`/`APWP 3`
  - Al ejecutar el programa de pieza debe estar definida la contraseña del usuario final.
  - El ciclo debe estar guardado en el directorio _N_CUS_DIR (Usuario), _N_CMA_DIR o _N_CST_DIR.
  - Para los directorios _N_CUS_DIR, _N_CMA_DIR o _N_CST_DIR, los derechos de ejecución deben estar ajustados como mínimo a Usuario final en los datos de máquina DM11162 $MN_ACCESS_EXEC_CUS, DM11161 $MN_ACCESS_EXEC_CMA o DM11160 $MN_ACCESS_EXEC_CST.
- `APRP 4...7`/`APWP 4...7`
  - Al ejecutar el programa de pieza debe estar ajustada la posición 3 ... 0 del interruptor de llave.
  - El ciclo debe estar guardado en el directorio _N_CUS_DIR, _N_CMA_DIR o _N_CST_DIR.
  - Para los directorios _N_CUS_DIR, _N_CMA_DIR o _N_CST_DIR, los derechos de ejecución deben estar ajustados como mínimo a la correspondiente posición del interruptor de llave en los datos de máquina DM11162 $MN_ACCESS_EXEC_CUS, DM11161 $MN_ACCESS_EXEC_CMA o DM11160 $MN_ACCESS_EXEC_CST.

Derechos de acceso relativos a BTSS (**APRB**, **APWB**)

Los derechos de acceso (APRB, APWB) limitan el acceso a las variables de sistema y de usuario mediante BTSS para todos los componentes de sistema (HMI, PLC, ordenadores externos, servicios EPS, etc.) en igual medida.

Nota

Derechos de acceso locales de HMI

Si se realizan modificaciones de derechos de acceso de datos de sistema hay que vigilar que estas modificaciones sean coherentes con los derechos de acceso definidos mediante mecanismos de HMI.

Atributos de acceso **APR/APW**

Los atributos de acceso APR/APW tienen las siguientes correspondencias:

```
APR 0...7          ⇒   APRP 0...7
APW 0...7          ⇒   APWP 0...7
APR 10...17        ⇒   APRB 0...7
APW 10...17        ⇒   APWB 0...7
```

Nota

Limitaciones relativas a APR

Al utilizar APR deben observarse las siguientes limitaciones:

- Una limitación de los derechos de acceso para la **lectura** de variables de sistema y de usuario en programas de pieza/ciclos mediante APR 0...7 no tiene ninguna repercusión.
- Una limitación de los derechos de acceso para la lectura de variables de sistema y de usuario a través de BTSS mediante APR 10...17 sólo se tiene en cuenta en caso de un acceso desde HMI. El acceso desde otros componentes (PLC, ordenadores externos, servicios EPS, etc.) no sufre limitaciones.

Ajustar los derechos de acceso mediante ficheros ACCESS

Al utilizar ficheros ACCESS para la asignación de derechos de acceso, las redefiniciones de derechos de acceso para datos de sistema, datos de usuario e instrucciones de programación CN ya sólo pueden programarse en estos ficheros ACCESS. Los datos globales del usuario (GUD) son una excepción. Para éstos, en caso de que sea necesario, la redefinición de los derechos de acceso deberá seguirse programando en los correspondientes ficheros de definición.

Para un derecho de acceso homogéneo, los datos de máquina para los derechos de ejecución y el derecho de acceso de los correspondientes directorios deben adaptarse de una forma coherente.

Se obtiene la siguiente forma de proceder básica:

- Creación de los ficheros de definición necesarios:
  - _N_DEF_DIR/_N_SACCESS_DEF
  - _N_DEF_DIR/_N_MACCESS_DEF
  - _N_DEF_DIR/_N_UACCESS_DEF
- Parametrización del derecho de escritura para los ficheros de definición al valor necesario para la redefinición:
  - DM11170 $MN_ACCESS_WRITE_SACCESS
  - DM11171 $MN_ACCESS_WRITE_MACCESS
  - DM11172 $MN_ACCESS_WRITE_UACCESS
- Para los accesos a los elementos protegidos desde los ciclos, deben adaptarse los derechos de ejecución y de escritura de los directorios de ciclos _N_CST_DIR, _N_CMA_DIR y _N_CST_DIR:

  Derechos de ejecución
  - DM11160 $MN_ACCESS_EXEC_CST
  - DM11161 $MN_ACCESS_EXEC_CMA
  - DM11162 $MN_ACCESS_EXEC_CUS

  Derechos de escritura
  - DM11165 $MN_ACCESS_WRITE_CST
  - DM11166 $MN_ACCESS_WRITE_CMA
  - DM11167 MN_ACCESS_WRITE_CUS

  El derecho de ejecución debe ajustarse como mínimo al mismo nivel que el nivel de protección máximo del elemento utilizado.

  El derecho de escritura debe ajustarse como mínimo al mismo nivel de protección que el derecho de ejecución.
- Los derechos de escritura de los directorios de ciclos locales de HMI deben ajustarse al mismo nivel de protección que los de los directorios de ciclos locales de CN.

  Bibliografía

  /BAD/ Manual del usuario HMI Advanced,

  Capítulo: Campo de manejo Servicios > Gestión de datos > Modificar propiedades

Llamadas de subprograma en ficheros ACCESS

Para la siguiente estructuración de la protección de acceso pueden llamarse también subprogramas (identificación SPF o MPF) en los ficheros ACCESS. Los subprogramas heredan entonces los derechos de ejecución del fichero ACCESS desde el que se llama.

Nota

En los ficheros ACCESS sólo pueden redefinirse los derechos de acceso. Todos los demás atributos se deben seguir redefiniendo o programando en los correspondientes ficheros de definición.

1.1.11 Vista general de atributos definibles y redefinibles

Las siguientes tablas muestran los atributos que se pueden definir (DEF) y/o redefinir (REDEF) y para qué tipos de datos.

Datos de sistema

Tipo de datos	Valor inic.	Valores límite	Unidad fís.	Derechos de acceso
Datos de máquina	---	---	---	REDEF
Datos del operador	REDEF	---	---	REDEF
Datos de FRAME	---	---	---	REDEF
Datos de proceso	---	---	---	REDEF
Comp. de error de paso de husillo (EEC)	---	---	---	REDEF
Compensación de la flexión (CEC)	---	---	---	REDEF
Compensación del error de cuadrante (QEC)	---	---	---	REDEF
Datos de almacén	---	---	---	REDEF
Datos de herramienta	---	---	---	REDEF
Zonas protegidas	---	---	---	REDEF
Portaherramientas orientables	---	---	---	REDEF
Cadenas cinemáticas	---	---	---	REDEF
Zonas de protección 3D	---	---	---	REDEF
Limitación del campo de trabajo	---	---	---	REDEF
Datos de herramienta ISO	---	---	---	REDEF

Datos de usuario

Tipo de datos	Valor inic.	Valores límite	Unidad fís.	Derechos de acceso
Parámetros R	REDEF	REDEF	REDEF	REDEF
Variable de acción síncrona ($AC_...)	REDEF	REDEF	REDEF	REDEF
GUD de acción síncrona (SYG_...)	REDEF	REDEF	REDEF	REDEF
Parámetros EPS	REDEF	REDEF	REDEF	REDEF
OEM de datos de herramienta	REDEF	REDEF	REDEF	REDEF
OEM de datos de almacén	REDEF	REDEF	REDEF	REDEF
Variables globales del usuario (GUD)	DEF/REDEF	DEF	DEF	DEF/REDEF
Variables locales del usuario (PUD/LUD)	DEF	DEF	DEF	---

1.1.12 Definición e inicialización de variables de matriz (DEF, SET, REP)

Función

Una variable de usuario puede definirse como una matriz (array) de 1 a 3 dimensiones como máximo:

- 1 dimensión: `DEF <Tipo de datos> <Nombre de la variable>[<n>]`
- 2 dimensiones: `DEF <Tipo de datos> <Nombre de la variable>[<n>,<m>]`
- 3 dimensiones: `DEF <Tipo de datos> <Nombre de la variable>[<n>,<m>,<o>]`

Nota

Las variables de usuario del tipo de datos STRING pueden definirse como una matriz de 2 dimensiones como máximo.

Tipos de datos

Las variables de usuario pueden definirse como matrices para los siguientes tipos de datos: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Asignación de valores a elementos de matriz

Las asignaciones de valores a elementos de matriz pueden realizarse en los momentos siguientes:

- en la definición de matrices (valores de inicialización);
- durante la ejecución del programa.

En este caso, los valores se pueden asignar:

- indicando explícitamente un elemento de matriz;
- indicando explícitamente un elemento de matriz como elemento inicial e indicando una lista de valores (`SET`);
- indicando explícitamente un elemento de matriz como elemento inicial e indicando un valor y su frecuencia de repetición (`REP`).

Nota

No se pueden asignar valores de inicialización a variables de usuario del tipo de datos FRAME.

Sintaxis (`DEF`)

```
DEF <Tipo de datos> <Nombre de la variable>[<n>,<m>,<o>]
DEF  STRING[<Longitud del string>] <Nombre de la variable>[<n>,<m>]
```

Sintaxis (`DEF...=SET...`)

Utilización de una lista de valores:

- En la definición:

```
DEF <Tipo de datos> <Nombre de la variable>[<n>,<m>,<o>]=
SET(<Valor1>,<Valor2>,...)
```

con el mismo significado que:

```
DEF <Tipo de datos> <Nombre de la variable>[<n>,<m>,<o>]=
(<Valor1>,<Valor2>,...)
```

Nota

En la inicialización mediante una lista de valores la indicación de `SET` es opcional.

- En una asignación de valores:

```
<Nombre de la variable>[<n>,<m>,<o>]=SET(<VALOR1>,<Valor2>,...)
```

Sintaxis (`DEF...= REP...`)

Utilización de un valor con repetición

- En la definición:

```
DEF <Tipo de datos> <Nombre de la variable>[<n>,<m>,<o>]=
REP(<Valor>)
```

```
DEF <Tipo de datos> <Nombre de la variable>[<n>,<m>,<o>]=
REP(<Valor>,<Cantidad de elementos de la matriz>)
```

- En una asignación de valores:

```
<Nombre de la variable>[<n>,<m>,<o>]=REP(<Valor>)
<Nombre de la variable>[<n>,<m>,<o>]=REP(<Valor>,<Cantidad de
elementos de la matriz>)
```

Descripción

`DEF`:	Comando para la definición de variables
`<Tipo de datos>`:	Tipo de datos de las variables Rango de valores: • En variables del sistema: BOOL, CHAR, INT, REAL, STRING, AXIS • En variables GUD o LUD: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME
`<Longitud del string>`:	Cantidad máxima de caracteres en el tipo de datos STRING

<Nombre de la variable>:	Nombre de variable
[<n>,<m>,<o>]:	Tamaños o índices de matriz
<n>:	Tamaño o índice de matriz para la 1ª dimensión Tipo: INT (en variables del sistema también AXIS) Rango de valores: Tamaño máx. de matriz: 65535 Índice de matriz: 0 ≤ n ≤ 65534
<m>:	Tamaño o índice de matriz para la 2ª dimensión Tipo: INT (en variables del sistema también AXIS) Rango de valores: Tamaño máx. de matriz: 65535 Índice de matriz: 0 ≤ m ≤ 65534
<o>:	Tamaño o índice de matriz para la 3ª dimensión Tipo: INT (en variables del sistema también AXIS) Rango de valores: Tamaño máx. de matriz: 65535 Índice de matriz: 0 ≤ o ≤ 65534
SET:	Asignación de valores mediante la lista de valores indicada
(<Valor1>,<Valor2>,...):	Lista de valores
REP:	Asignación de valores mediante el <Valor> indicado
<Valor>:	Valor con el que deben describirse los elementos de matriz durante la inicialización con REP.
<Cantidad de elementos de la matriz>:	Cantidad de elementos de la matriz que deben escribirse con el <Valor> indicado. Para el resto de elementos de la matriz, se aplica lo siguiente en función del momento: • Inicialización en la definición de matrices: → Los elementos de matriz restantes se escriben con cero • Asignación durante la ejecución del programa: → Los valores actuales de los elementos de matriz se mantienen invariables. Si el parámetro no está programado, todos los elementos de matriz se escriben con el <Valor>. Si el parámetro es igual a cero, en función del momento se aplica lo siguiente: • Inicialización en la definición de matrices: → Todos los elementos se preasignarán en cero • Asignación durante la ejecución del programa: → Los valores actuales de los elementos de matriz se mantienen invariables.

Índice de matriz

El orden implícito de los elementos de matriz, p. ej., en una asignación de valores mediante SET o REP, se realiza con la iteración del índice de matriz de derecha a izquierda.

Ejemplo: Inicialización de una matriz de 3 dimensiones con 24 elementos:

```
DEF INT MATRIZ[2,3,4] = REP(1,24)
  MATRIZ[0,0,0] = 1    1er elemento de matriz
  MATRIZ[0,0,1] = 1    2º elemento de matriz
  MATRIZ[0,0,2] = 1    3er elemento de matriz
  MATRIZ[0,0,3] = 1    4º elemento de matriz
  ...
  MATRIZ[0,1,0] = 1    5º elemento de matriz
  MATRIZ[0,1,1] = 1    6º elemento de matriz
  ...
  MATRIZ[0,2,3] = 1    12º elemento de matriz
  MATRIZ[1,0,0] = 1    13º elemento de matriz
  MATRIZ[1,0,1] = 1    14º elemento de matriz
  ...
  MATRIZ[1,2,3] = 1    24º elemento de matriz
```

correspondientemente:

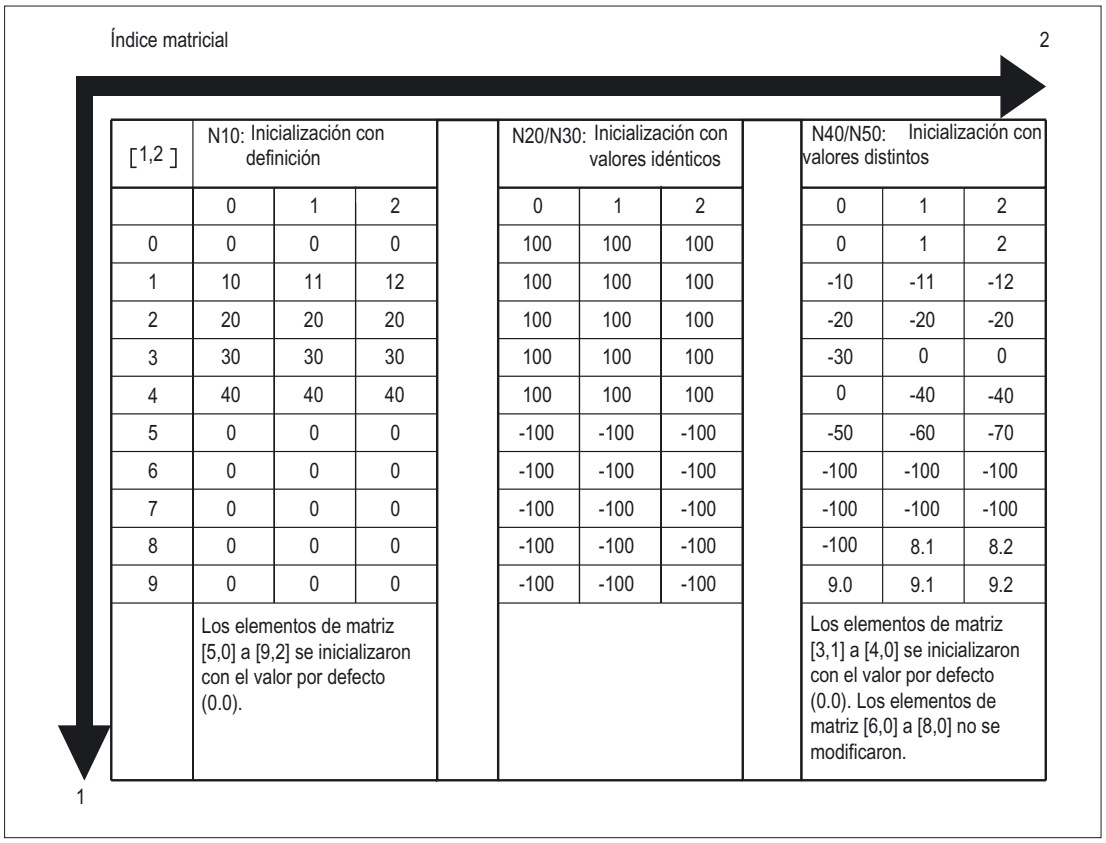
```
FOR n=0 TO 1
  FOR m=0 TO 2
     FOR o=0 TO 3
       MATRIZ[n,m,o] = 1
     ENDFOR
  ENDFOR
ENDFOR
```

Ejemplo: Inicialización de todos los elementos de matriz

Para la ocupación actual, ver figura.

Código del programa

```
N10 DEF REAL MATRIZ1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)
N20 MATRIZ1[0,0] = REP(100)
N30 MATRIZ1[5,0] = REP(-100)
N40 MATRIZ1[0,0]=SET(0,1,2,-10,-11,-12,-20,-20,-20,-30, , , ,-40,-40,-50,-60,-70)
N50 MATRIZ1[8,1]=SET(8.1,8.2,9.0,9.1,9.2)
```



[1,2]	N10: Inicialización con definición			N20/N30: Inicialización con valores idénticos			N40/N50: Inicialización con valores distintos		
	0	1	2	0	1	2	0	1	2
0	0	0	0	100	100	100	0	1	2
1	10	11	12	100	100	100	-10	-11	-12
2	20	20	20	100	100	100	-20	-20	-20
3	30	30	30	100	100	100	-30	0	0
4	40	40	40	100	100	100	0	-40	-40
5	0	0	0	-100	-100	-100	-50	-60	-70
6	0	0	0	-100	-100	-100	-100	-100	-100
7	0	0	0	-100	-100	-100	-100	-100	-100
8	0	0	0	-100	-100	-100	-100	8.1	8.2
9	0	0	0	-100	-100	-100	9.0	9.1	9.2
	Los elementos de matriz [5,0] a [9,2] se inicializaron con el valor por defecto (0.0).						Los elementos de matriz [3,1] a [4,0] se inicializaron con el valor por defecto (0.0). Los elementos de matriz [6,0] a [8,0] no se modificaron.		

Información adicional (SET)

Inicialización en la definición

- Empezando por el 1er elemento de matriz, se inicializan tantos elementos de matriz con los valores de la lista de valores como elementos programados en la lista de valores.
- Los elementos de matriz sin valores indicados explícitamente en la lista de valores (huecos en la lista de valores) se ocupan con 0.
- Las variables del tipo de datos AXIS no permiten huecos en la lista de valores.
- Si la lista de valores contiene más valores que elementos de matriz definidos, se muestra una alarma.

Asignación de valores en la ejecución del programa

Durante la asignación de valores en la ejecución del programa, se aplican las reglas descritas anteriormente en la definición: Adicionalmente existen las posibilidades siguientes:

- Como elementos de la lista de valores también están permitidas expresiones:
- La asignación de valores empieza con el índice de matriz programado. De esta manera, determinados elementos de la matriz se pueden inicializar con un valor.

Ejemplo:

Código del programa	Comentarios
`DEF INT MATRIZ[5,5]`	`; Definición de matrices`
`MATRIZ[0,0]=SET(1,2,3,4,5)`	`; Asignación de valores a los primeros 5 elementos de matriz [0,0] - [0,4]`
`MATRIZ[0,0]=SET(1,2, , ,5)`	`; Asignación de valores con hueco a los primeros 5 elementos de matriz [0,0] - [0,4], elementos de matriz [0,2] y [0,3] = 0`
`MATRIZ[2,3]=SET(VARIABLE,4*5.6)`	`; Asignación de valores con variable y expresión a partir del índice de matriz [2,3]: [2,3] = VARIABLE [2,4] = 4 * 5.6 = 22.4`

Información adicional (REP)

Inicialización en la definición

- Todas las cantidades o las cantidades indicadas opcionalmente de elementos de matriz se inicializan con el valor (constante) indicado.
- Las variables del tipo de datos FRAME no pueden inicializarse.

Ejemplo:

Código de programa	Comentarios
`DEF REAL varName[10]=REP(3.5,4)`	`; Inicializar la definición de matrices y elementos de matriz de [0] a [3] con el valor 3,5`

Asignación de valores en la ejecución del programa

Durante la asignación de valores en la ejecución del programa, se aplican las reglas descritas anteriormente en la definición: Adicionalmente existen las posibilidades siguientes:

- Como elementos de la lista de valores también están permitidas expresiones:
- La asignación de valores empieza con el índice de matriz programado. De esta manera, determinados elementos de la matriz se pueden inicializar con un valor.

Ejemplos:

Código de programa	**Comentarios**
`DEF REAL varName[10]`	`; Definición de matrices`
`varName[5]=REP(4.5,3)`	`; Elementos de matriz de [5] a [7] = 4,5`
`R10=REP(2.4,3)`	`; Parámetros R de R10 a R12 = 2,4`
`DEF FRAME FRM[10]`	`; Definición de matrices`
`FRM[5] = REP(CTRANS (X,5))`	`; Elementos de matriz de [5] a [9] = CTRANS(X,5)`

Información adicional (general)

Asignaciones de valores a datos de máquina axiales

Los datos de máquina axiales tienen en principio un índice de matriz del tipo de datos `AXIS`. En asignaciones de valores a un dato de máquina axial mediante `SET` o `REP`, este índice de matriz se ignora o no se ejecuta.

Ejemplo: Asignación de valores al dato de máquina DM36200 $MA_AX_VELO_LIMIT

```
$MA_AX_VELO_LIMIT[1, AX1] = SET(1.1, 2.2, 3.3)
```

Corresponde a:

```
$MA_AX_VELO_LIMIT[1,AX1]=1.1
$MA_AX_VELO_LIMIT[2,AX1]=2.2
$MA_AX_VELO_LIMIT[3,AX1]=3.3
```

ATENCIÓN

Asignaciones de valores a datos de máquina axiales

En asignaciones de valores a datos de máquina axiales mediante `SET` o `REP`, el índice de matriz del tipo de datos `AXIS` se ignora o no se ejecuta.

Memoria necesaria

Tipo de datos	Capacidad de memoria requerida por elemento
BOOL	1 byte
CHAR	1 byte
INT	4 bytes
REAL	8 bytes
STRING	(Longitud del string + 1) bytes
FRAME	~ 400 bytes, en función del número de ejes
AXIS	4 bytes

1.1.13 Tipos de datos

En el CN están disponibles los siguientes tipos de datos:

Tipo de datos	Descripción	Rango
INT	Valor entero con signo	-2147483648 ... +2147483647
REAL	Número real (LONG REAL según IEEE)	±(~2,2*10^{-308} … ~1,8*10^{+308})
BOOL	Valores binarios TRUE (1) y FALSE (0)	1, 0
CHAR	Caracteres ASCII	Código ASCII 0 … 255
STRING	Cadena de caracteres de una longitud definida	Máximo 200 caracteres (sin caracteres especiales)
AXIS	Identificador de cabezal/eje	Identificador de eje de canal
FRAME	Datos geométricos para una transformación de coordenadas estática (desplazar, girar, escalar, invertir en simetría especular)	---

Conversiones de tipos de datos implícitas

Son posibles las siguientes conversiones de tipos de datos y se realizan implícitamente en asignaciones y transferencias de parámetros:

de ↓/ a →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x

x: posible sin limitaciones

o: posible pérdida de datos por rebase del rango de valores ⇒ alarma;
redondeo: valor decimal ≥ 0,5 ⇒ redondear al alza, valor decimal < 0,5 ⇒ redondear a la baja

&: valor ≠ 0 ⇒ TRUE, valor == 0 ⇒ FALSE

1.2 Programación indirecta

Función

En la programación indirecta de direcciones, la dirección ampliada (índice) se sustituye por una variable del tipo adecuado.

Nota

La programación indirecta de direcciones no es posible para:

- N (Número de secuencia)
- L (subprograma)
- Direcciones ajustables
  (p. ej. X[1] en lugar de X1 no está permitido)

Sintaxis

```
<DIRECCIÓN>[<Índice>]
```

Descripción

`<DIRECCIÓN>[...]`:	Dirección fija con ampliación (índice)
`<Índice>`:	Variable, p. ej. para número de cabezal, eje...

Ejemplos

Ejemplo 1: Programación indirecta de un número de cabezal

Programación directa:

Código del programa	Comentarios
`S1=300`	`; Velocidad de 300 rpm para el cabezal con el número 1.`

Programación indirecta:

Código del programa	Comentarios
`DEF INT N_CABEZAL=1`	`; Definición de las variables del tipo INT y asignación de valores.`
`S[SPINU]=300`	`; Velocidad de 300 rpm para el cabezal cuyo número está guardado en la variable SPINU (en este ejemplo, el cabezal con el número 1).`

Ejemplo 2: Programación indirecta de un eje

Programación directa:

Código del programa	Comentarios
FA[U]=300	; Avance 300 para el eje "U".

Programación indirecta:

Código del programa	Comentarios
DEF AXIS AXVAR2=U	; Definición de una variable del tipo AXIS y asignación de valores.
FA[AXVAR2]=300	; Avance 300 para el eje cuyo nombre de dirección está guardado en la variable con el nombre AXVAR2.

Ejemplo 3: Programación indirecta de un eje

Programación directa:

Programación	Comentarios
$AA_MM[X]	; Leer medida del palpador (MKS) del eje "X".

Programación indirecta:

Código del programa	Comentarios
DEF AXIS AXVAR3=X	; Definición de una variable del tipo AXIS y asignación de valores.
$AA_MM[AXVAR3]	; Leer medida del palpador (MKS) del eje cuyo nombre está guardado en la variable AXVAR3.

Ejemplo 4: Programación indirecta de un eje

Programación directa:

Código del programa
X1=100 X2=200

Programación indirecta:

Código del programa	Comentarios
DEF AXIS AXVAR1 AXVAR2	; Definición de dos variables del tipo AXIS.
AXVAR1=(X1) AXVAR2=(X2)	; Asignación de los nombres de eje.
AX[AXVAR1]=100 AX[AXVAR2]=200	; Desplazamiento de los ejes cuyos nombres de dirección están guardados en las variables con los nombres AXVAR1 y AXVAR2.

Ejemplo 5: Programación indirecta de un eje

Programación directa:

Código del programa
G2 X100 I20

Programación indirecta:

Código del programa	Comentarios
DEF AXIS AXVAR1=X	; Definición de una variable del tipo AXIS y asignación de valores.
G2 X100 IP[AXVAR1]=20	; Programación indirecta de la indicación de centro para el eje cuyo nombre de dirección está guardado en la variable con el nombre AXVAR1

Ejemplo 6: Programación indirecta de elementos de matriz

Programación directa:

Código del programa	Comentarios
DEF INT MATRIZ1[4,5]	; Definición de la matriz 1.

Programación indirecta:

Código del programa	Comentarios
DEFINE DIM1 AS 4	; Al definir las dimensiones de la matriz debe indicarse el tamaño de la matriz como valor fijo.
DEFINE DIM2 AS 5	
DEF INT MATRIZ[DIM1,DIM2]	
MATRIZ[DIM1-1,DIM2-1]=5	

Ejemplo 7: Llamada indirecta de subprograma

Código del programa	Comentarios
CALL "L" << R10	; Llamada del programa cuyo número se encuentra en R10 (concatenación).

1.2.1 Programación indirecta de códigos G

Función

La programación indirecta de códigos G permite una programación de ciclos efectiva.

Sintaxis

```
G[<Grupo>]=<Número>
```

Descripción

`G[...]`:	Comando G con ampliación (índice)
`<Grupo>`:	Parámetro de índice: Grupo de funciones G Tipo: INT **Nota:** Sólo se pueden programar indirectamente los grupos de funciones G modales. Los grupos de funciones G por secuencias se rechazarán con una alarma.
`<Número>`:	Variable para el número de código G Tipo: INT o REAL **Nota:** En la programación indirecta de códigos G no se admiten funciones aritméticas. Un cálculo necesario del número de código G se tiene que realizar en una línea propia del programa de pieza antes de la programación indirecta de códigos G.

Ejemplos

Ejemplo 1: Decalaje de origen ajustable (grupo de funciones G 8)

```
Código del programa                    Comentarios
N1010 DEF INT INT_VAR
N1020 INT_VAR = 2
...
N1090 G[8]=INT_VAR G1 X0 Y0            ; G54
N1100 INT_VAR=INT_VAR+1                ; Cálculo de códigos G
N1110 G[8]=INT_VAR G1 X0 Y0            ; G55
```

Ejemplo 2: Selección del plano (grupo de funciones G 6)

Código del programa	Comentarios
N2010 R10=$P_GG[6]	; Leer función G activa del grupo de funciones G 6
...	
N2090 G[6]=R10	

Bibliografía

Encontrará información acerca de los grupos de funciones G en:
Manual de programación, Fundamentos; capítulo "Lista de las funciones G/Condiciones de recorrido"

1.2.2 Programación indirecta de atributos de posición (GP)

Función

Los atributos de posición, como p. ej. la programación incremental o absoluta de la posición del eje, pueden programarse indirectamente como variables junto con la palabra reservada `GP`.

Aplicación

La programación indirecta de atributos de posición se utiliza en **ciclos de sustitución**, ya que aquí se obtiene la siguiente ventaja con respecto a la programación de atributos de posición como palabra reservada (p. ej. `IC`, `AC`):

Gracias a la programación indirecta como variable, no se necesita **ninguna** instrucción `CASE` que se derive hacia todos los atributos de posición posibles.

Sintaxis

```
<COMANDO DE POSICIONAMIENTO>[<Eje/cabezal>]=
GP(<Posición>,<Atributo de posición>)
<Eje/cabezal>=GP(<Posición>,<Atributo de posición>)
```

Descripción

<COMANDO DE POSICIONAMIENTO>[]:	Los siguientes comandos de posicionamiento pueden programarse junto con la palabra reservada GP: POS, POSA,SPOS, SPOSA También son posibles: • todos los identificadores de eje/cabezal existentes en el canal: <Eje/cabezal> • identificador de eje/cabezal variable AX
<Eje/cabezal>:	Eje/cabezal que debe posicionarse
GP():	Palabra reservada para el posicionamiento
<Posición>:	Parámetro 1 Posición de eje/cabezal como constante o variable
<Atributo de posición>:	Parámetro 2 Atributo de posición (p. ej. modo de aproximación a la posición) como variable (p. ej. $P_SUB_SPOSMODE) o como palabra reservada (IC, AC...)

Los valores suministrados por las variables tienen el siguiente significado:

Valor	Descripción	Admisible en:
0	Ninguna modificación del atributo de posición	
1	AC	POS, POSA,SPOS, SPOSA,AX, dirección de eje
2	IC	POS, POSA,SPOS, SPOSA,AX, dirección de eje
3	DC	POS, POSA,SPOS, SPOSA,AX, dirección de eje
4	ACP	POS, POSA,SPOS, SPOSA,AX, dirección de eje
5	ACN	POS, POSA,SPOS, SPOSA,AX, dirección de eje
6	OC	-
7	PC	-
8	DAC	POS, POSA,AX, dirección de eje
9	DIC	POS, POSA,AX, dirección de eje
10	RAC	POS, POSA,AX, dirección de eje
11	RIC	POS, POSA,AX, dirección de eje
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

Ejemplo

En caso de acoplamiento síncrono activo de cabezales entre el cabezal maestro S1 y el cabezal esclavo S2, se llamará al siguiente ciclo de sustitución para el posicionamiento de los cabezales mediante el comando SPOS en el programa principal.

El posicionamiento se realiza mediante la instrucción en N2230:
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)

La posición que se debe alcanzar se lee de la variable del sistema $P_SUB_SPOSIT, el modo de aproximación a la posición se lee de la variable del sistema $P_SUB_SPOSMODE.

```
Código del programa                                    Comentarios
N1000 PROC LANG_SUB DISPLOF SBLOF
...
N2100 IF($P_SUB_AXFCT==2)
N2110                                                  ; Sustitución del comando SPOS/SPOSA/M19
                                                         durante un acoplamiento síncrono activo de
                                                         cabezales
N2185 DELAYFSTON                                       ; Inicio del área Stop-Delay
N2190 COUPOF(S2,S1)                                    ; Desactivar acoplamiento síncrono de
                                                         cabezales
N2200                                                  ; Posicionar cabezal maestro y cabezal
                                                         esclavo
N2210 IF($P_SUB_SPOS==TRUE) OR ($P_SUB_SPOSA==TRUE)
N2220                                                  ; Posicionar cabezal con SPOS:
N2230    SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
         SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
N2250 ELSE
N2260                                                  ; Posicionar cabezal con M19:
N2270 M1=19 M2=19                                      ; Posicionar cabezal maestro y cabezal
                                                         esclavo
N2280 ENDIF
N2285 DELAYFSTOF                                       ; Fin del área Stop-Delay
N2290 COUPON(S2,S1)                                    ; Activar acoplamiento síncrono de cabezales
N2410 ELSE
N2420                                                  ; Consulta de otras sustituciones
...
N3300 ENDIF
...
N9999 RET
```

Limitaciones

- La programación indirecta de atributos de posición no es posible en acciones síncronas.

Bibliografía

Manual de funciones básicas; BAG/GMO, canal, modo de programa, comportamiento Reset (K1), apartado: Sustitución de funciones CN mediante subprogramas

1.2.3 Programación indirecta de líneas de programa de pieza (EXECSTRING)

Función

Con el comando de programa de pieza EXECSTRING se transfiere como parámetro un string que contiene la línea del programa de pieza que se tiene que ejecutar efectivamente.

Sintaxis

```
EXECSTRING (<Variable string>)
```

Parámetros

EXECSTRING:	Transferencia de una variable string con la línea del programa de pieza a ejecutar
<Variable string>:	Parámetro con línea del programa de pieza a ejecutar efectivamente

Nota

Se pueden consignar todas las construcciones de programa de pieza que se pueden programar en la parte de programa de un programa de pieza. De esta manera se excluyen las instrucciones PROC y DEF, y en general la utilización en ficheros INI y DEF.

Ejemplo

Líneas indirectas de programa de pieza

Código del programa	Comentarios
N100 DEF STRING[100] BLOCK	; Variable string para el registro de las líneas de programa de pieza
N110 DEF STRING[10] MFCT1="M7"	
N200 EXECSTRING(MFCT1 << "M4711")	; Ejecutar línea de programa de pieza "M7 M4711"
N300 R10=1	
N310 BLOCK="M3"	
N320 IF(R10)	
N330 BLOCK = BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(BLOCK)	; Ejecutar línea de programa de pieza "M3 M4711"

1.3 Funciones de cálculo

Función

Las funciones de cálculo se utilizan generalmente para parámetros R y variables (o constantes y funciones) del tipo REAL. Los tipos de variables INT y CHAR también se pueden utilizar para dicho propósito.

Operador/función de cálculo	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
	Atención: (tipo INT)**/**(tipo INT)**=**(tipo REAL); ejemplo: 3/4 = 0.75
DIV	División, para variables del tipo INT y REAL
	Atención: (tipo INT)DIV(tipo INT)**=**(tipo INT); ejemplo: 3 DIV 4 = 0
MOD	División módulo (solamente para tipo INT) da como resultado el resto de una división INT
	Ejemplo: 3 MOD 4 = 3
:	Operador de concatenación (solamente para variables de tipo FRAME)
Sin()	Seno
COS( )	Coseno
TAN( )	Tangente
ASIN( )	Arcoseno
ACOS( )	Arcocoseno
ATAN2(,)	Arcotangente2
SQRT( )	Raíz cuadrada
ABS( )	Valor absoluto
POT( )	2º valor al cuadrado
TRUNC( )	Parte entera
	Precisiones en comandos de comparación ajustables con TRUNC (ver "Corrección de precisión en caso de errores de comparación (TRUNC) (Página 68)")
ROUND()	Redondeo a un entero
LN( )	Logaritmo neperiano (logaritmo natural)
EXP( )	Exponencial
MINVAL( )	Valor menor de dos variables
	(ver "Mínimo, máximo y rango de variables (MINVAL, MAXVAL, BOUND) (Página 70)")

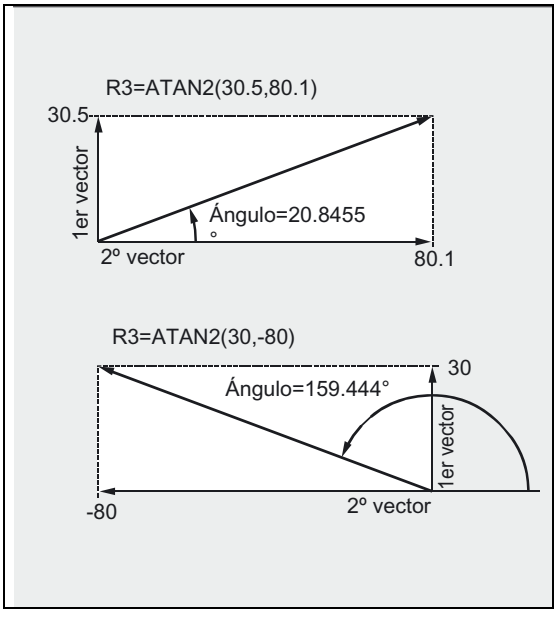
`MAXVAL( )`	Valor mayor de dos variables (ver "Mínimo, máximo y rango de variables (MINVAL, MAXVAL, BOUND) (Página 70)")
`BOUND( )`	Valor de variable que está en un rango de valores definido (ver "Mínimo, máximo y rango de variables (MINVAL, MAXVAL, BOUND) (Página 70)")
`CTRANS()`	Decalaje
`CROT ()`	Rotación
`CSCALE ()`	Factor de escala
`CMIRROR ()`	Simetría

Programación

En funciones de cálculo se utiliza la notación matemática estándar. Las prioridades para la ejecución de dichas operaciones se indican mediante paréntesis. Los ángulos utilizados para cálculos trigonométricos y sus funciones inversas se toman en grados (ángulo recto = 90°).

Ejemplos

Ejemplo 1: ATAN2



La función de cálculo ATAN2 calcula el ángulo respecto al origen del vector formado por dos componentes vectoriales orientadas a lo largo de los ejes de coordenadas.

El resultado se encuentra dentro del rango de los cuatro cuadrantes (-180° < 0 < +180°).

La referencia angular siempre se basa en el 2º valor en la dirección positiva.

Ejemplo 2: Inicialización de todos los elementos de matriz

Código del programa	**Comentarios**
R1=R1+1	; Nuevo valor R1 = antiguo valor R1 +1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; Multiplicaciones y divisiones se evalúan antes que sumas y restas.
R14=(R1+R2)*R3	; Primero se realizan las operaciones entre paréntesis.
R15=SQRT(POT(R1)+POT(R2))	; Primero se evalúan los paréntesis más internos: R15 = raíz cuadrada de (R1+R2)
RESFRAME=FRAME1:FRAME2 FRAME3=CTRANS(…):CROT(…)	; El operador de concatenación combina los frames en un frame resultante o asigna los valores a los componentes del frame

1.4 Operaciones de comparación y operaciones lógicas

Función

Las **operaciones de comparación** pueden utilizarse, p. ej., para formular una condición de salto. También se puede realizar la comparación de expresiones complejas.

Las operaciones de comparación se utilizan para variables del tipo `CHAR, INT, REAL` y `BOOL`. En variables del tipo `CHAR` se realiza una comparación de los valores codificados. Para variables del tipo `STRING`, `AXIS` y `FRAME` se pueden realizar comparaciones del tipo: **==** y **<>**, que también pueden utilizarse en acciones síncronas para operaciones con variables del tipo `STRING`.

El resultado de las operaciones de comparación siempre es del tipo `BOOL`.

Los**operadores lógicos** se utilizan para combinar valores lógicos.

Las operaciones lógicas sólo pueden aplicarse a variables del tipo `BOOL`. Mediante una conversión interna de tipos de variables también se pueden utilizar con los tipos de datos `CHAR, INT` y `REAL`.

En operaciones lógicas (booleanas) se debe tener en cuenta para los tipos de datos `BOOL`, `CHAR,INT` y `REAL`:

- 0 significa: FALSE
- Distinto de 0 significa: TRUE

Operadores lógicos binarios

Con las variables del tipo `CHAR` e `INT` se pueden realizar también operaciones lógicas binarias. En caso necesario, se realiza una conversión automática de los tipos de variables.

Programación

Operador de comparación	Descripción
==	Igual que
<>	Distinto
>	Mayor que
<	Menor que
>=	Mayor o igual
<=	Menor o igual

Operador lógico	Descripción
`AND`	Y
`OR`	O
`NOT`	Negación
`XOR`	O exclusiva

Operador lógico bit a bit	Descripción
B_AND	Y binario
B_OR	O binario
B_NOT	Negación binaria
B_XOR	O exclusivo binario

Nota

Se pueden utilizar paréntesis con expresiones aritméticas para definir la secuencia de ejecución de todos los operadores; esto permite anular las reglas normales de prioridad.

Nota

Entre los operadores y operandos lógicos deben insertarse espacios.

Nota

El operador B_NOT se refiere sólo a un operando. Éste se encuentra detrás del operador.

Ejemplos

Ejemplo 1: Operadores de comparación

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

o

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

El resultado de la comparación R10>=100 se almacena temporalmente en el parámetro R11.

Ejemplo 2: Operadores lógicos

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

o

```
IF NOT R10 GOTOB INICIO
```

NOT hace referencia sólo a un operando.

Ejemplo 3: Operadores lógicos binarios

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

1.5 Corrección de precisión en caso de errores de comparación (TRUNC)

Función

El comando TRUNC recorta el operando multiplicado con un factor de precisión.

Precisión ajustable en comandos de comparación

Los datos de telegrama de pieza del tipo REAL se representan a nivel interno en el formato IEEE con 64 bits. Debido a esta forma de representación, los números decimales se reflejan de forma imprecisa, lo cual puede producir resultados inesperados en una comparación con valores calculatorios ideales.

Igualdad relativa

Para evitar que las imprecisiones causadas por la forma de representación alteren el flujo del programa, la comprobación en los comandos de comparación no se efectúa con vistas a la igualdad absoluta, sino a la igualdad relativa.

Sintaxis

Corrección de precisión en caso de errores de comparación

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

Descripción

`TRUNC`: Supresión de decimales

Igualdad relativa tenida en cuenta de 10^{-12} con

- Igualdad: (==)
- Desigualdad: (<>)
- Mayor-igual: (>=)
- Menor-igual: (<=)
- Mayor/menor: (><) con igualdad absoluta
- Mayor: (>)
- Menor: (<)

Compatibilidad

Por motivos de compatibilidad, puede desactivarse la comprobación de la igualdad relativa con (>) y (<) definiendo el dato de máquina DM10280 $MN_ PROG_FUNCTION_MASK bit0 = 1.

Nota

Por las razones citadas, las comparaciones con datos del tipo REAL muestran generalmente una cierta imprecisión. En caso de desviaciones inaceptables se tiene que recurrir al cálculo de ENTEROS, multiplicando el operando con un factor de precisión y recortándolo después con TRUNC.

Acciones síncronas

El comportamiento descrito de los comandos de comparación también se aplica en acciones síncronas.

Ejemplos

Ejemplo 1: Consideraciones de precisión

```
Código del programa                                  Comentarios
N40 R1=61.01 R2=61.02 R3=0.01                        ; Asignación de los valores
                                                       iniciales
N41 IF ABS(R2-R1) > R3 GOTOF ERROR                   ; El salto se ejecutaría hasta
                                                       ahora
N42 M30                                              ; Fin del programa
N43 ERROR: SETAL(66000)                              ;
R1=61.01 R2=61.02 R3=0.01                            ; Asignación de los valores
                                                       iniciales
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000)                ; Corrección de precisión
R13=TRUNC(R3*1000)
IF ABS(R12-R11) > R13 GOTOF ERROR                    ; El salto ya no se ejecuta
M30                                                  ; Fin del programa
ERRORES: SETAL(66000)                                ;
```

Ejemplo 2: Formar y evaluar el cociente de ambos operandos

```
Código del programa                                  Comentarios
R1=61.01 R2=61.02 R3=0.01                            ; Asignación de los valores
                                                       iniciales
IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF ERROR           ; El salto no se ejecuta
M30                                                  ; Fin del programa
ERRORES: SETAL(66000)                                ;
```

1.6 Mínimo, máximo y rango de variables (MINVAL, MAXVAL, BOUND)

Función

Con los comandos MINVAL y MAXVAL pueden compararse entre sí los valores de dos variables. Como resultado se devuelve el valor menor (con MINVAL) o el valor mayor (con MAXVAL).

Con el comando BOUND se puede comprobar si el valor de una variable de prueba se encuentra dentro de un rango de valores definido.

Sintaxis

```
<Valor menor>=MINVAL(<Variable1>,<Variable2>)
<Valor mayor>=MAXVAL(<Variable1>,<Variable2>)
<Valor de retorno>=<BOUND>(<Mínimo>,<Máximo>,<Variable de prueba>)
```

Descripción

MINVAL:	Determina el valor **menor** de dos variables (<Variable1>, <Variable2>)
<Valor menor>:	Variable de resultado para el comando MINVAL Se ajusta al valor de variable menor.
MAXVAL:	Determina el valor **mayor** de dos variables (<Variable1>, <Variable2>)
<Valor mayor>:	Variable de resultado para el comando MAXVAL Se ajusta al valor de variable mayor.
BOUND:	Comprueba si una variable (<Variable de prueba>) está situada dentro de un rango de valores definido.
<Mínimo>:	Variable que define el valor mínimo del rango de valores
<Máximo>:	Variable que define el valor máximo del rango de valores
<Valor de retorno>:	Variable de resultado para el comando BOUND Si el valor de las variables de prueba está situado dentro del rango de valores definido, la variable de resultado se ajustará al valor de las variables de prueba. Si el valor de las variables de prueba es mayor que el valor máximo, la variable de resultado se ajustará al valor máximo de la zona de definición. Si el valor de las variables de prueba es menor que el valor mínimo, la variable de resultado se ajustará al valor mínimo de la zona de definición.

Nota

MINVAL, MAXVAL y BOUND se pueden programar también en acciones síncronas.

Nota

Comportamiento en caso de igualdad

En caso de igualdad, con MINVAL/MAXVAL se suministra este mismo valor. Con BOUND se devolverá el valor de las variables que se deben comprobar.

Ejemplo

```
Código del programa                                  Comentarios
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar
rValMin=MINVAL(rVar1,rVar2)                          ; rValMin toma el valor 10.5.
rValMax=MAXVAL(rVar1,rVar2)                          ; rValMax toma el valor 33.7.
rVar3=19.7
rRetVar=BOUND(rVar1,rVar2,rVar3)                     ; rVar3 está situado dentro de los límites, rRetVar
                                                       toma el valor 19.7.
rVar3=1.8
rRetVar=BOUND(rVar1,rVar2,rVar3)                     ; rVar3 está situado por debajo del límite mínimo,
                                                       rRetVar toma el valor 10.5.


rVar3=45.2
rRetVar=BOUND(rVar1,rVar2,rVar3)                     ; rVar3 está situado por encima del límite máximo,
                                                       rRetVar toma el valor 33.7.
```

1.7 Prioridad de los operaciones

Función

Cada operador tiene una prioridad asignada. Al evaluar una expresión aritmética se realizan primeramente las operaciones con mayor orden de prioridad. En el caso de que se encuentren varias operaciones con el mismo orden de prioridad, éstas se ejecutan secuencialmente de izquierda a derecha.

Se pueden utilizar paréntesis con expresiones aritméticas para definir la secuencia de ejecución de todos los operadores; esto permite anular las reglas normales de prioridad.

Prioridades de los operadores

De mayor a menor

1.	NOT, B_NOT	Negación, negación binaria
2.	*, /, DIV, MOD	Multiplicación, división
3.	+, –	Suma, resta
4.	B_AND	Y binario
5.	B_XOR	O exclusivo binario
6.	B_OR	O binario
7.	AND	Y
8.	XOR	O exclusivo
9.	OR	O
10.	<<	Concatenación de caracteres, resultado tipo STRING
11.	==, <>, >, <, >=, <=	Operadores de comparación

Nota

El operador de concatenación ":" para frames no debe aparecer con otros operadores en una misma expresión. Por lo tanto no es necesario definir un nivel de prioridad para dicho operador.

Ejemplo de instrucción If:

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

1.8 Conversiones de tipos posibles

Función

Conversión de tipos con asignación

Los valores numéricos constantes, variables o expresiones asignadas a una variable deben de ser compatibles con el tipo de la variable. En el caso de que esto sea así, la variable toma automáticamente el valor asignado.

Posibles conversiones de tipos

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

Aclaraciones

* En la conversión de tipos de REAL a INT se redondea hacia arriba con una fracción de >=0.5; de lo contrario, se redondea hacia abajo (ver función ROUND)

1) El valor <> 0 equivale a TRUE, el valor == 0 equivale a FALSE

2) Si el valor se sitúa en el margen numérico admisible

3) Con sólo 1 carácter

4) Longitud del string 0 = >FALSE, de lo contrario TRUE

Nota

Si a la hora de convertir un valor, dicho valor es mayor que el rango permitido, automáticamente se genera un mensaje de error.

Si se mezclan varios tipos de variables en una expresión, automáticamente se realiza una conversión de tipo. Las conversiones de tipos también son posibles en las acciones síncronas, ver apartado "Acciones síncronas de desplazamiento, conversión implícita de tipos".

1.9 Operaciones con cadenas de caracteres

Operaciones de cadena

Además de las clásicas operaciones "Asignación" y "Comparación", son posibles las siguientes operaciones de cadena:

- Conversión de tipos a STRING (AXSTRING)
- Conversión de tipos STRING (NUMBER, ISNUMBER, AXNAME)
- Concatenación de cadenas (<<)
- Conversión minúsculas/mayúsculas (TOLOWER, TOUPPER)
- Definición de la longitud de una cadena (STRLEN)
- Búsqueda de caracteres/cadena en una cadena (INDEX, RINDEX, MINDEX, MATCH)
- Selección de una cadena parcial (SUBSTR)
- Selección de un carácter único (STRINGVAR, STRINGFELD)

Significado especial del carácter 0

El carácter 0 se interpreta internamente como delimitador final de una cadena. Si se sustituye un carácter por 0, se acorta la cadena.

Ejemplo:

```
Código del programa                              Comentarios
DEF STRING[20] STRG="Eje . parado"
STRG[6]="X"
MSG(STRG)                                        ; Emite el aviso "Eje X parado".
STRG[6]=0
MSG(STRG)                                        ; Emite el aviso "Eje".
```

1.9.1 Conversión de tipos a STRING (AXSTRING)

Función

La función "Conversión de tipo a STRING" permite utilizar variables de distintos tipos como componente de un aviso (MSG).

Tienen lugar al utilizar el operador << implícitamente para los tipos de datos INT, REAL, CHAR y BOOL (ver "Concatenación de cadenas (<<) (Página 77)").

Un valor INT se convertirá en una forma legible normal. Para los valores REAL se introducirán hasta 10 caracteres asociados a las correspondientes posiciones decimales.

El comando `AXSTRING` permite convertir variables del tipo AXIS a STRING.

Sintaxis

```
<STRING_ERG> = << <Cualquier tipo>
<STRING_ERG> = AXSTRING(<Identificador de eje>)
```

Descripción

<STRING_ERG>:	Variable para el resultado de la conversión de tipo Tipo: STRING
<Cualquier tipo>:	Tipos de variables INT, REAL, CHAR, STRING y BOOL
AXSTRING:	El comando AXSTRING proporciona el identificador de eje indicado como cadena.
<Identificador de eje>:	Variable para el identificador de eje Tipo: AXIS

Nota

Las variables FRAME no se pueden convertir.

Ejemplos

Ejemplo 1:

```
MSG("Posición:"<<$AA_IM[X])
```

Ejemplo 2: AXSTRING

Código del programa	Comentarios
DEF STRING[32] STRING_ERG	
STRING_ERG=AXSTRING(X)	; STRING_ERG == "X"

1.9.2 Conversión de tipos de STRING (NUMBER, ISNUMBER, AXNAME)

Función

El comando NUMBER convierte STRING a REAL. Es posible comprobar la convertibilidad con el comando ISNUMBER.

El comando AXNAME convierte una cadena en el tipo de datos AXIS.

Sintaxis

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

Descripción

NUMBER: El comando NUMBER devuelve la cifra representada por el <String> como valor REAL.

<String>: Variable del tipo STRING que debe convertirse

<REAL_ERG>: Variable para el resultado de la conversión de tipo con NUMBER

Tipo: REAL

ISNUMBER: El comando ISNUMBER permite verificar si el <String> puede convertirse en un número válido.

<BOOL_ERG>: Variable para el resultado de la consulta con ISNUMBER

Tipo: BOOL

Valor: TRUE — ISNUMBER proporciona el valor TRUE cuando el <String> representa un número REAL de acuerdo a las reglas del idioma.

FALSE — Si ISNUMBER proporciona el valor FALSE, con la llamada de NUMBER con el mismo <String> se activa una alarma.

AXNAME: El comando AXNAME convierte el <String> indicado en un identificador de eje.

Nota:
Si el <String> no puede asignarse a ningún identificador de eje configurado, se desencadenará una alarma.

<AXIS_ERG>: Variable para el resultado de la conversión de tipo con AXNAME

Tipo: AXIS

Ejemplo

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

1.9.3 Concatenación de cadenas (<<)

Función

La función "Concatenación de cadenas" permite formar una cadena a partir de distintos componentes.

La concatenación se efectúa mediante el operador "<<". Este operador genera para todas las combinaciones de los tipos de variables básicas CHAR, BOOL, INT, REAL y STRING una variable del tipo de destino STRING. En el caso de que sea necesario realizar una conversión de tipo, ello se realizará siguiendo las reglas prefijadas.

Sintaxis

```
<Cualquier tipo> << <Cualquier tipo>
```

Descripción

`<Cualquier tipo>`:	Variable del tipo CHAR, BOOL, INT, REAL o STRING
`<<` :	Operador para la concatenación de variables (<Cualquier tipo>) para forma una cadena de caracteres compuesta (tipo STRING). Este operador también está disponible solo como variante "unaria". De este modo es posible ejecutar una conversión de tipo explícita a STRING (no para FRAME y AXIS): `<< <Cualquier tipo>`

Por ejemplo, es posible componer de este modo un aviso o un comando a partir de listas de textos y añadir parámetros (p. ej., un nombre de bloque):

```
MSG(STRG_TAB[LOAD_IDX]<<BAUSTEIN_NAME)
```

PRECAUCIÓN
Los resultados intermedios de la concatenación de cadenas no deben superar la longitud máxima de cadena.

Nota

Los tipos FRAME y AXIS no pueden utilizarse junto con el operador "<<".

Ejemplos

Ejemplo 1: Concatenación de cadenas

```
Código del programa                                    Comentarios
DEF INT IDX=2
DEF REAL VALUE=9.654
DEF STRING[20] STRG="INDEX:2"
IF STRG=="Índice:"<<IDX GOTOF NO_MSG
MSG("Índice:"<<IDX<<"/valor:"<<VALUE)                  ; Visualización:
                                                         "Índice:2/valor:9.654"
NO_MSG:
```

Ejemplo 2: Conversión de tipos explícita con <<

```
Código del programa        Comentarios
DEF REAL VALUE=3.5
<<VALUE                    ; La variable indicada del tipo REAL se convierte al tipo
                             STRING.
```

1.9.4 Conversión minúsculas/mayúsculas (TOLOWER, TOUPPER)

Función

La función "Conversión minúsculas/mayúsculas" permite convertir todas las letras de una cadena de caracteres para que se representen uniformemente.

Sintaxis

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

Descripción

TOUPPER:	El comando TOUPPER permite convertir todas las letras de una cadena de caracteres en letras **mayúsculas**.
TOLOWER:	El comando TOLOWER permite convertir todas las letras de una cadena de caracteres en letras **minúsculas**.
<String>:	Cadena de caracteres que debe convertirse Tipo: STRING
<STRING_ERG>:	Variable para el resultado de la conversión Tipo: STRING

Ejemplo

Como también existe la posibilidad de que el usuario introduzca datos a través de la interfaz de usuario, se puede obtener una representación uniforme utilizando letras minúsculas o mayúsculas:

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

1.9.5 Definir la longitud de una cadena (STRLEN)

Función

El comando STRLEN permite definir la longitud de una cadena de caracteres.

Sintaxis

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

Descripción

STRLEN:	El comando STRLEN define la longitud de la cadena de caracteres indicada. Se devuelve el número de caracteres que, contados desde el principio de la cadena de caracteres, no son caracteres 0.
<String>:	Cadena de caracteres cuya longitud debe determinarse Tipo: STRING
<INT_ERG>:	Variable para el resultado de la determinación Tipo: INT

Ejemplo

La función en combinación con el acceso a un carácter individual permite determinar el final de una cadena de caracteres:

```
Código del programa
IF (STRLEN(BAUSTEIN_NAME)>10) GOTOF ERROR
```

1.9.6 Búsqueda de caracteres/cadena en una cadena (INDEX, RINDEX, MINDEX, MATCH)

Función

Esta funcionalidad permite buscar caracteres individuales, así como cadenas de caracteres dentro de un string. El resultado de la función indica en qué posición del string en el que se realiza la búsqueda se encuentra el carácter/la cadena de caracteres que se desea localizar.

Sintaxis

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

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

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

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

Semántica

Funciones de búsqueda: Devuelven la posición de la cadena (primer parámetro) en la que la búsqueda ha tenido resultados. Si no se pueden encontrar el carácter o la cadena, el valor devuelto será -1. El primer carácter tiene la posición 0.

Descripción

`INDEX`:	Busca el carácter indicado como segundo parámetro (desde el principio) en el primer parámetro.
`RINDEX`:	Busca el carácter indicado como segundo parámetro (desde el final) en el primer parámetro.
`MINDEX`:	Equivale a la función INDEX, excepto en que se emite una lista de caracteres (como cadena) de la que se devuelve el índice del primer carácter encontrado.
`MATCH`:	Busca una cadena dentro de una cadena.

De este modo es posible dividir las cadenas según criterios determinados, como en posiciones con espacios o por el carácter separador de ruta ("/").

Ejemplo

División de una entrada en un nombre de ruta y de bloque

Código del programa	**Comentarios**
DEF INT PFADIDX, PROGIDX	
DEF STRING[26] ENTRADA	
DEF INT LISTIDX	
ENTRADA = "/_N_MPF_DIR/_N_EXECUTE_MPF"	
LISTIDX = MINDEX (ENTRADA, "M,N,O,P") + 1	; En LISTIDX se devuelve 3 como valor, puesto que "N" es el primer carácter del parámetro ENTRADA de la lista de selección (desde el principio).
PFADIDX = INDEX (ENTRADA, "/") +1	; Por tanto: PFADIDX = 1
PROGIDX = RINDEX (ENTRADA, "/") +1	; Por tanto: PROGIDX = 12
	Con la función SUBSTR introducida en la siguiente sección es posible dividir la variable ENTRADA en los componentes ;"Ruta";y "Bloque":
VARIABLE = SUBSTR (ENTRADA, PFADIDX, PROGIDX-PFADIDX-1)	; proporciona "_N_MPF_DIR"
VARIABLE = SUBSTR (ENTRADA, PROGIDX)	; proporciona "_N_EXECUTE_MPF"

1.9.7 Selección de una cadena parcial (SUBSTR)

Función

Con esta función se puede extraer una cadena parcial de caracteres de un string. Para ello se indica el índice para la posición del primer carácter dentro del string, así como eventualmente la cantidad de caracteres que se desean extraer. En el caso de que esta última no se indique, se toma por defecto el resto de la cadena a partir de la posición inicial indicada.

Sintaxis

`STRING_ERG = SUBSTR (STRING,INT)` ; **tipo de resultado: INT**

`STRING_ERG = SUBSTR(STRING,INT, INT)` ; **tipo de resultado: INT**

Semántica

En el primer caso, la cadena parcial se devuelve hasta el final de la cadena a partir de la posición definida por el segundo parámetro.

En el segundo caso, la cadena de resultado se limita a la longitud máxima definida por el tercer parámetro.

Si la posición inicial se encuentra detrás del final de la cadena, se devuelve la cadena vacía (" ").

Si la posición inicial o la longitud son negativas, se desencadena una alarma.

Ejemplo

Código del programa	**Comentarios**
`DEF STRING[29] RESULT`	
`RESULT = SUBSTR ("ACUSE:10 a 99", 10, 2)`	`; Por tanto: RESULT == "10"`

1.9.8 Selección de un carácter único (STRINGVAR, STRINGFELD)

Función

Esta función permite seleccionar un determinado carácter dentro de un string. Con ello no sólo es posible leer, sino también escribir un carácter dentro de una cadena.

Sintaxis

CHAR_ERG = STRINGVAR [IDX] ; **tipo de resultado: CHAR**

CHAR_ERG = STRINGFELD [IDX_FELD, IDX_CHAR] ; **tipo de resultado: CHAR**

Semántica

El carácter indicado se lee/escribe dentro de la cadena de caracteres dada, atendiendo a la posición indicada. Si se indica un valor de posición negativo o mayor que la longitud del string, el sistema emite una alarma.

Ejemplo para avisos:

Introducción del nombre de un eje en una cadena de caracteres predefinida.

```
Código del programa                                          Comentarios
DEF STRING [50] AVISO = "El eje n ha alcanzado su posición"
AVISO [6] = "X"
MSG (AVISO)                                                  ; Emite el aviso "El eje X ha
                                                               alcanzado la posición"
```

Parámetros

El acceso a caracteres individuales solamente se puede realizar dentro de variables definidas por el usuario (datos LUD, GUD y PUD).

Además, dentro de llamadas a subprogramas este tipo de acceso solamente es válido para parámetros de tipo "Call-By-Value".

Ejemplos

Ejemplo 1: Acceso a un carácter individual en un dato de sistema, de máquina...

```
Código del programa                                  Comentarios
DEF STRING [50] STRG
DEF CHAR ACUSE
…
STRG = $P_MMCA
ACUSE = STRG [0]                                     ;  Evaluación de los componentes de
                                                        acuse
```

Ejemplo 2: Acceso a un carácter individual en un parámetro Call-By-Reference

Código del programa	**Comentarios**
`DEF STRING [50] STRG`	
`DEF CHAR CHR1`	
`EXTERN UP_CALL (VAR CHAR1)`	`;  Parámetro Call-By-Reference`
…	
`CHR1 = STRG [5]`	
`UP_CALL (CHR1)`	`;  Call-By-Reference`
`STRG [5] = CHR1`	

1.10 Saltos y derivaciones del programa

1.10.1 Salto atrás hasta el inicio del programa (GOTOS)

Función

El comando GOTOS permite saltar atrás hasta el principio de un programa principal o subprograma para repetir dicho programa.

Mediante datos de máquina puede configurarse que, con cada salto atrás hasta el inicio del programa:

- el tiempo de ejecución del programa se ajuste a "0";
- el cómputo de piezas se incremente el valor "1".

Sintaxis

GOTOS

Descripción

GOTOS: Instrucción de salto hasta el inicio del programa.

La ejecución se controla mediante la señal de interfaz CN/PLC.

DB21, ... DBX384.0 (Controlar bifurcación de programa)

Valor:	Significado:
0	Sin salto atrás hasta el inicio del programa. La ejecución del programa continúa con la secuencia del programa de pieza que sigue a GOTOS.
1	Salto atrás hasta el inicio del programa. Se repite el programa de pieza.

Limitaciones

- GOTOS desencadena internamente una STOPRE (parada de decodificación previa).
- En un programa de pieza con definiciones de datos (variables LUD) con GOTOS se salta a la primera secuencia de programa que sigue a la sección de definición, es decir, las definiciones de datos no se ejecutan de nuevo. Con ello, las variables definidas mantienen el valor alcanzado en la secuencia GOTOS y no se restablecen al valor estándar programado en la sección de definición.
- El comando GOTOS no está disponible en las acciones síncronas ni en los ciclos tecnológicos.

Ejemplo

Código del programa	Comentarios
N10 ...	; Inicio del programa
...	
N90 GOTOS	; Salto al inicio del programa
...	

1.10.2 Saltos de programa a marcas de salto (GOTOB, GOTOF, GOTO, GOTOC)

Función

Es posible fijar marcas de salto (etiquetas) en un programa a las que se puede saltar desde otros puntos del mismo programa con los comandos GOTOF, GOTOB, GOTO oGOTOC. La ejecución del programa continúa con la instrucción inmediatamente posterior a la marca de salto. De este modo es posible realizar derivaciones dentro del programa.

Además de las marcas de salto, también son posibles números de secuencia principal y de secuencia auxiliar como destinos del salto.

Si, antes de la instrucción de salto, hay formulada una condición para el salto (IF ...), el salto del programa tendrá lugar sólo si se cumple la condición para el salto.

Sintaxis

```
GOTOB <Destino del salto>
IF <Condición para el salto> = TRUE GOTOB <Destino del salto>
GOTOF <Destino del salto>
IF <Condición para el salto> = TRUE GOTOF <Destino del salto>
GOTO <Destino del salto>
IF <Condición para el salto> = TRUE GOTO <Destino del salto>
GOTOC <Destino del salto>
IF <Condición para el salto> = TRUE GOTOC <Destino del salto>
```

Descripción

GOTOB:	Instrucción de salto con destino de salto hacia el principio del programa.
GOTOF:	Instrucción de salto con destino de salto hacia el final del programa.
GOTO:	Instrucción de salto con búsqueda de destino de salto. La búsqueda se realiza en primer lugar hacia el final del programa y, a continuación, hacia el principio del programa.

GOTOC:	Efecto idéntico que con GOTO con la diferencia de que se suprime la alarma 14080 "Destino del salto no encontrado". Esto significa que, la ejecución del programa no se interrumpe en caso de que la búsqueda del destino de salto no dé ningún resultado, sino que continúa con la siguiente línea de programa del comando GOTOC.	
<Destino del salto>:	Parámetro de destino de salto Posibles indicaciones:	
	<Marca de salto>:	El destino del salto es la marca de salto definida en el programa con un nombre definido por el usuario: <Marca de salto>:
	<Número de secuencia>:	El destino del salto es un número de secuencia principal o auxiliar (p. ej.: 200, N300)
	Variable del tipo STRING:	Variable Destino del salto. La variable corresponde a una marca de salto o un número de secuencia.
IF:	Palabra reservada para formular la condición de salto. La condición para el salto permite utilizar todos los operadores de comparación y lógicos (resultado: TRUE o FALSE). El salto del programa se realiza si el resultado de la operación es TRUE.	

Nota

Marcas de salto (etiquetas)

Las marcas de salto se encuentran siempre al principio de la secuencia. Cuando existe un número de programa, la marca de salto se encuentra inmediatamente a continuación del número de la secuencia.

Para la denominación de marcas de salto deben respetarse las siguientes reglas:

- Número de caracteres:
  - como mínimo 2
  - como máximo 32
- Caracteres permitidos:
  - Letras
  - Números
  - Caracteres de subrayado
- Los dos primeros caracteres tienen que ser letras o bien el carácter de subrayado.
- Al final del nombre de la marca de salto se deben introducir dos puntos (":").

Limitaciones

- El destino del salto solamente puede ser una secuencia cuya marca de salto o número de secuencia se encuentre **dentro** de dicho programa.
- La instrucción de salto sin condición de salto debe programarse en una secuencia separada. Esta limitación no se aplica en el caso de instrucciones de salto con condiciones de salto. Aquí pueden formularse varias instrucciones de salto en una secuencia.
- En programas con instrucciones de salto sin condiciones de salto, el final del programa `M2`/`M30` no debe encontrarse forzosamente al final del programa.

Ejemplos

Ejemplo 1: Saltos a marcas de salto

```
Código del programa              Comentarios
N10…
N20 GOTOF Label_1                ; Salto hacia el final del programa, hasta la marca
                                   de salto "Label_1".
N30 …
N40 Label_0: R1=R2+R3            ; Marca de salto "Label_0" fijada.
N50 …
N60 Label_1:                     ; Marca de salto "Label_1" fijada.
N70 …
N80 GOTOB Label_0                ; Salto hacia el inicio del programa, hasta la
                                   marca de salto "Label_0".
N90 …
```

Ejemplo 2: Salto indirecto a número de secuencia

```
Código del programa              Comentarios
N5 R10=100
N10 GOTOF "N"<<R10               ; Salto a la secuencia cuyo número figura en R10.
...
N90 ...
N100 ...                         ; Destino del salto
N110 ...
...
```

Ejemplo 3: Salto a un destino de salto variable

```
Código del programa              Comentarios
DEF STRING[20] DESTINO
DESTINO = "Marca2"
GOTOF DESTINO                    ; Salto hacia el final del programa, hasta el
                                   destino de salto variable DESTINO.
Marca1: T="Broca1"
...
Marca2: T="Broca2"               ; Destino del salto
...
```

Ejemplo 4: Salto con condición de salto

```
Código del programa                              Comentarios
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20          ; Asignación de los valores
                                                   iniciales.
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6      ; Marca de salto LA1 fijada.
N42 R1=R1+R3 R4=R4-1
N43 IF R4>0 GOTOB LA1                            ; Si se cumple la condición de
                                                   salto, el salto se efectúa hacia
                                                   el inicio del programa, hasta la
                                                   marca de salto LA1.
N44 M30                                          ; Fin del programa
```

1.10.3 Bifurcación de programa (CASE ... OF ... DEFAULT ...)

Función

La función CASE ofrece la posibilidad de comprobar el valor actual (tipo: INT) de una variable o una función de cálculo, así como de saltar a diferentes lugares del programa en función del resultado.

Sintaxis

```
CASE(<expresión>) OF <Constante_1> GOTOF <Destino del salto_1>
<Constante_2> GOTOF <Destino del salto_2> ... DEFAULT GOTOF <Destino
del salto_n>
```

Descripción

CASE:	Instrucción de salto
<Expresión>:	Variable o función de cálculo
OF:	Palabra reservada para formular las derivaciones de programa condicionadas
<Constante_1>:	Primer valor constante indicado para la variable o la función de cálculo Tipo: INT
<Constante_2>:	Segundo valor constante indicado para la variable o la función de cálculo Tipo: INT
DEFAULT:	En el caso de que la variable o la función de cálculo no tomen ninguno de los valores constantes indicados, se puede definir un destino de salto con la instrucción DEFAULT. **Nota:** Si no está programada la instrucción DEFAULT, el destino de salto en estos casos será la secuencia posterior a la instrucción CASE.
GOTOF:	Instrucción de salto con destino de salto hacia el final del programa. En lugar de GOTOF pueden programarse también todos los demás comandos GOTO (ver el tema "Saltos de programa a marcas de salto").
<Destino del salto_1>:	Se deriva a este destino de salto cuando el valor de las variables o de la función de cálculo corresponde a la primera constante indicada. El destino de salto se puede indicar como sigue: <Marca de salto>: El destino del salto es la marca de salto definida en el programa con un nombre definido por el usuario: <Marca de salto>: <Número de secuencia>: El destino del salto es un número de secuencia principal o auxiliar (p. ej.: 200, N300) Variable del tipo STRING: Variable Destino del salto. La variable corresponde a una marca de salto o un número de secuencia.
<Destino del salto_2>:	Se deriva a este destino de salto cuando el valor de las variables o de la función de cálculo corresponde a la segunda constante indicada.
<Destino del salto_n>:	Se deriva a este destino de salto cuando el valor de las variables no adopta ninguno de los valores constantes indicados.

Ejemplo

Código del programa

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

La instrucción CASE de N30 define las siguientes opciones de derivación de programa:

1. Si el valor de la función de cálculo VAR1+VAR2-VAR3 = 7, salta a la secuencia con la definición de marcas de salto "Label_1" (→ N40).
2. Si el valor de la función de cálculo VAR1+VAR2-VAR3 = 9, salta a la secuencia con la definición de marcas de salto "Label_2" (→ N50).
3. Si el valor de la función de cálculo VAR1+VAR2-VAR3 no es ni 7 ni 9, salta a la secuencia con la definición de marcas de salto "Label_3" (→ N60).

1.11 Repetición de secciones de programa (REPEAT, REPEATB, ENDLABEL, P)

Función

La función de repetición de secciones de programa permite repetir, en los puntos y formas que se deseen, partes ya escritas.

Las líneas o áreas del programa que deben repetirse se identifican con marcas de salto (etiquetas).

Nota

Marcas de salto (etiquetas)

Las marcas de salto se encuentran siempre al principio de la secuencia. Cuando existe un número de programa, la marca de salto se encuentra inmediatamente a continuación del número de la secuencia.

Para la denominación de marcas de salto deben respetarse las siguientes reglas:

- Número de caracteres:
  - como mínimo 2
  - como máximo 32
- Caracteres permitidos:
  - Letras
  - Números
  - Caracteres de subrayado
- Los dos primeros caracteres tienen que ser letras o bien el carácter de subrayado.
- Al final del nombre de la marca de salto se deben introducir dos puntos (":").

Sintaxis

1. Repetir una sola línea de programa:

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

2. Repetir una área del programa situada entre la marca de salto y la instrucción REPEAT:

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

3. Repetir una área situada entre dos marcas de salto:

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

Nota

No es posible acotar la instrucción `REPEAT` con las dos marcas de salto. Si la `<Marca de salto inicial>` se encuentra delante de la instrucción `REPEAT` y no se llega a la `<Marca de salto final>` antes de la instrucción `REPEAT`, se realiza la repetición entre la `<Marca de salto inicial>` y la instrucción `REPEAT`.

4. Repetir una área situada entre la marca de salto y ENDLABEL:

```
<Marca de salto>: ...
...
LABELFINAL: ...
...
REPEAT <Marca de salto> P=<n>
...
```

Nota

No es posible acotar la instrucción `REPEAT` con la `<Marca de salto>` y `ENDLABEL`. Si la `<Marca de salto>` se encuentra delante de la instrucción `REPEAT` y no se llega a `ENDLABEL` antes de la instrucción `REPEAT`, se realiza la repetición entre la `<Marca de salto>` y la instrucción `REPEAT`.

Descripción

REPEATB:	Comando para repetir una línea de programa
REPEAT:	Comando para repetir una área de programa
<Marca de salto>:	La <Marca de salto> identifica: • la línea de programa que debe repetirse (con REPEATB) o bien • el principio del área de programa que debe repetirse (con REPEAT) La línea de programa identificada con la <Marca de salto> puede estar delante o detrás de la instrucción REPEAT/REPEATB. La búsqueda comienza en sentido al inicio del programa. Si no se encuentra la marca de salto en esta dirección, se busca en dirección al final de programa. **Excepción:** si el área de programa situada entre la marca de salto y la instrucción REPEAT debe repetirse (ver punto 2 del apartado Sintaxis), la línea de programa identificada con la <Marca de salto> debe situarse **delante** de la instrucción REPEAT, ya que en este caso **sólo** se busca en dirección al inicio del programa. Si la línea con la <Marca de salto> tiene más instrucciones, éstas se vuelven a ejecutar en cada repetición.
ENDLABEL:	Palabra reservada que marca el final de una área de programa que debe repetirse. Si la línea con ENDLABEL tiene más instrucciones, éstas se vuelven a ejecutar en cada repetición. ENDLABEL puede utilizarse varias veces en el programa.
P:	Dirección para indicar el número de repeticiones
<n>:	Número de repeticiones de secciones de programa Tipo: INT La sección de programa que debe repetirse se repite <n> veces. Tras la última repetición, el programa continúa ejecutándose a partir de la línea siguiente a la línea REPEAT/REPEATB. **Nota:** Si no está indicado ningún P=<n>, la sección de programa que debe repetirse se repite exactamente una vez.

Ejemplos

Ejemplo 1: Repetir una sola línea de programa

```
Código del programa                    Comentarios
N10 POSITION1: X10 Y20
N20 POSITION2: CYCLE(0,,9,8)           ; Ciclo de posicionado
N30 ...
N40 REPEATB POSITION1 P=5              ; Ejecutar cinco veces la SECUENCIA N10.
N50 REPEATB POSITION2                  ; Ejecutar una vez la secuencia N20.
N60 ...
N70 M30
```

Ejemplo 2: Repetir una área del programa situada entre la marca de salto y la instrucción REPEAT

```
Código del programa          Comentarios
N5 R10=15
N10 Begin: R10=R10+1         ; Anchura
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         ; Ejecutar cuatro veces el área comprendida entre N10 y
                               N70.
N90 Z10
N100 M30
```

Ejemplo 3: Repetir una área situada entre dos marcas de salto

```
Código del programa              Comentarios
N5 R10=15
N10 Begin: R10=R10+1             ; Anchura
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        ; Ejecutar tres veces el área comprendida entre N10
                                   y N70.
N110 Z10
N120 M30
```

Ejemplo 4: Repetir una área situada entre la marca de salto y ENDLABEL

```
Código del programa          Comentarios
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X10
N40 Y10
N50 BEGIN2:
N60 X20
N70 Y30
N80 ENDLABEL: Z10
N90 X0 Y0 Z0
N100 Z-10
N110 BEGIN3: X20
N120 Y30
N130 REPEAT BEGIN3 P=3       ; Ejecutar tres veces el área comprendida entre N110 y
                               N120.
N140 REPEAT BEGIN2 P=2       ; Ejecutar dos veces el área comprendida entre N50 y
                               N80.
N150 M100
N160 REPEAT BEGIN1 P=2       ; Ejecutar dos veces el área comprendida entre N20 y
                               N80.
N170 Z10
N180 X0 Y0
N190 M30
```

Ejemplo 5: Fresado, mecanizar posición de taladrado con diferentes tecnologías

```
Código del programa          Comentarios
N10 CENTRADORA()             ; Cargar centradora.
N20 POS_1:                   ; Posiciones de taladrado 1
N30 X1 Y1
N40 X2
N50 Y2
N60 X3 Y3
N70 ENDLABEL:
N80 POS_2:                   ; Posiciones de taladrado 2
N90 X10 Y5
N100 X9 Y-5
N110 X3 Y3
N120 ENDLABEL:
N130 BROCA()                 ; Cambiar broca y ciclo de taladrado.
N140 MACHO(6)                ; Cargar macho de roscar M6 y ciclo de roscado.
N150 REPEAT POS_1            ; Repetir una vez la sección del programa comprendida
                               entre POS_1 y ENDLABEL.
N160 BROCA()                 ; Cambiar broca y ciclo de taladrado.
N170 MACHO(8)                ; Cargar macho de roscar M8 y ciclo de roscado.
N180 REPEAT POS_2            ; Repetir una vez la sección del programa comprendida
                               entre POS_2 y ENDLABEL.
N190 M30
```

Información adicional

- Las repeticiones de secciones de programa pueden imbricarse. Cada llamada ocupa un nivel de subprograma.
- Si durante la ejecución de una repetición de sección hay programada `M17` o `RET`, se interrumpe dicha repetición. En tal caso, el programa continúa ejecutándose a partir de la secuencia que sigue a la línea `REPEAT`.
- En la indicación actual del programa, la repetición de la sección de programa se indica como nivel de subprograma propio.
- Si durante la ejecución de una sección del programa se activa una interrupción de nivel, el programa continúa ejecutándose tras la instrucción de llamada de la sección del programa.

  Ejemplo:

```
Código del programa              Comentarios
N5 R10=15
N10 BEGIN: R10=R10+1             ; Anchura
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10                        ; Interrupción de nivel
N50 X=-R10
N60 Y=-R10
N70 END: Z10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3
N120 Z10                         ; Continuar con la ejecución del programa.
N130 M30
```

- Es posible combinar estructuras de control y repeticiones de secciones del programa. Sin embargo, no puede haber solapamientos. Una repetición de sección de programa conviene que esté situada en una ramificación de estructura de control o una estructura de control dentro de una repetición de sección de programa.

- Si se mezclan saltos y repeticiones de secciones de programa, las secuencias se ejecutan de forma secuencial pura. Por ejemplo, si se salta desde una repetición de sección de programa, la ejecución continúa hasta que se encuentre el fin de sección programado.

  Ejemplo:

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

Nota

La instrucción `REPEAT` se debería situar detrás de las secuencias de desplazamiento.

1.12 Estructuras de control

Función

De forma estándar, el control numérico ejecuta las secuencias de CN en el orden programado.

Es posible variar este orden mediante la programación de bloques y bucles de programa alternativos. La programación de estas estructuras de control se realiza con los elementos de estructuras de control (palabras reservadas) `IF...ELSE`, `LOOP`, `FOR`, `WHILE` y `REPEAT`.

⚠ PRECAUCIÓN

Solamente se pueden utilizar estructuras de control dentro del bloque de programa que contiene las instrucciones. Definiciones en la parte cabecera del programa no pueden ejecutarse condicional o repetidamente.

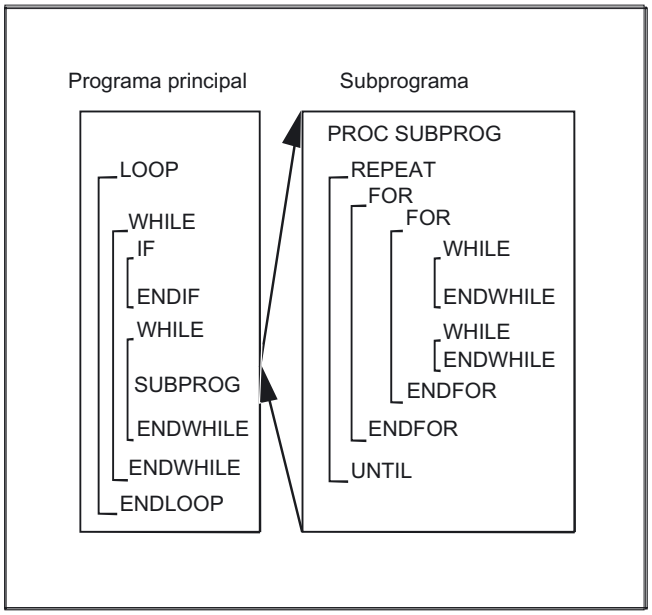
No se deberán solapar con macros las palabras reservadas para estructuras de control ni tampoco las metas de los saltos. Al definir las macros no se comprueba este punto.

Efecto

Las estructuras de control son válidas de forma local dentro de un mismo programa.

Profundidad de imbricado

Dentro de cada llamada a un subprograma es posible realizar un nivel de imbricación de hasta 16 estructuras de control.



Comportamiento en cuanto a tiempo de ejecución

El tiempo necesario para la ejecución de un determinado programa en el modo Intérprete, el estándar, se puede acortar sensiblemente mediante la utilización de saltos en vez de estructuras de control.

Al utilizar ciclos precompilados, no hay diferencia entre saltos y estructuras de control.

Limitaciones

- Las secuencias que contengan elementos de estructuras de control no pueden ser opcionales.
- Las marcas de salto (etiquetas) no están permitidas en las secuencias con elementos de estructuras de control.
- Las estructuras de control se ejecutan de forma interpretativa. El sistema, al reconocer el final de un bucle, buscará automáticamente el inicio del bucle, atendiendo a las estructuras de control halladas. Por ello, en el modo Intérprete no se realiza una comprobación completa de la estructura de bloques de un programa.
- Se aconseja no utilizar de forma mixta saltos y estructuras de control.
- Durante el preprocesamiento previo de ciclos se puede comprobar la imbricación correcta de estructuras de control.

1.12.1 Bucle de programa con alternativa (IF, ELSE, ENDIF)

Función

Una construcción con `IF` y `ELSE` se utiliza cuando el bucle de programa debe contener un bloque de programa alternativo: Si se cumple la condición `IF`, se ejecuta el bloque de programa que sigue a `IF`. Si **no** se cumple la condición `IF`, se ejecuta el bloque de programa alternativo que sigue a `ELSE`.

Nota

Si no es necesaria una alternativa, también puede programarse un bucle IF sin la instrucción `ELSE` ni el bloque de programa que sigue a `ELSE`.

Sintaxis

```
IF <Condición>
...
ELSE
...
ENDIF
```

Descripción

IF:	Inicia el bucle IF.
ELSE:	Inicia el bloque de programa alternativo.
ENDIF:	Marca el final del bucle IF y salta atrás hasta el inicio del bucle.
<Condición>:	Condición para decidir qué bloque de programa debe ejecutarse.

Ejemplo

Subprograma de cambio de herramienta

Código del programa	**Comentarios**
PROC L6	; Rutina de cambio de herramienta
N500 DEF INT TNR_AKTUELL	; Variable para número T activo
N510 DEF INT TNR_VORWAHL	; Variable para número T predefinido
	; Determinar la herramienta actual
N520 STOPRE	
N530 IF $P_ISTEST	; En el modo de test del programa...
N540 TNR_AKTUELL = $P_TOOLNO	; ... se lee la herramienta "actual" del contexto de programa.
N550 ELSE	; De lo contrario...
N560 TNR_AKTUELL = $TC_MPP6[9998,1]	; ... se lee la herramienta del cabezal.
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	; Leer el número T de la herramienta predefinida del cabezal.
N590 IF TNR_AKTUELL <> TNR_VORWAHL	; Si la herramienta predefinida no es todavía la herramienta actual...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	; ... Posicionamiento en el punto de cambio de herramienta...
N610 M206	; ... y ejecución del cambio de herramienta.
N620 ENDIF	
N630 M17	

1.12.2 Bucle de programa sin fin (LOOP, ENDLOOP)

Función

El bucle sin fin se utiliza para programas que se deban ejecutar constantemente. Al final del bucle siempre se realiza un salto hacia atrás hasta el principio del bucle.

Sintaxis

```
LOOP
...
ENDLOOP
```

Descripción

LOOP:	Inicia el bucle sin fin.
ENDLOOP:	Marca el final del bucle y salta atrás hasta el inicio del bucle.

Ejemplo

Código del programa

```
...
LOOP
MSG("Sin filos de herramienta activos")
M0
STOPRE
ENDLOOP
...
```

1.12.3 Bucle contador (FOR ... TO ..., ENDFOR)

Función

El bucle contador se utiliza cuando se debe realizar una determinada cantidad de pasadas.

Sintaxis

```
FOR <Variable> = <Valor inicial> TO <Valor final>
...
ENDFOR
```

Descripción

FOR:	Inicia el bucle contador.
ENDFOR:	Marca el final del bucle y realiza un salto atrás hasta el inicio del bucle, mientras no se haya alcanzado el valor final del recuento.
<Variable>:	Variable de contador que se incrementa en cada pasada el valor "1", desde su valor inicial hasta su valor final. Tipo INT o REAL **Nota:** El tipo REAL se toma cuando se programan p. ej. parámetros R para un bucle contador. Si la variable de contador es del tipo REAL su valor se redondea a un valor entero.
<Valor inicial>:	Valor inicial del recuento Condición: El valor inicial deberá ser menor que el valor final.
<Valor final>:	Valor final del recuento

Ejemplos

Ejemplo 1: Variable INTEGER o parámetro R como variable de contador

Variable INTEGER como variable de contador:

```
Código del programa                 Comentarios
DEF INT iVARIABLE1
R10=R12-R20*R1 R11=6
FOR iVARIABLE1 = R10 TO R11         ; Variable de contador = variable INTEGER
 R20=R21*R22+R33
ENDFOR
M30
```

Parámetro R como variable de contador:

```
Código del programa                 Comentarios
R11=6
FOR R10=R12-R20*R1 TO R11           ; Variable de contador = parámetro R (variable
                                      real)
 R20=R21*R22+R33
ENDFOR
M30
```

Ejemplo 2: Mecanizado de una cantidad de piezas determinada

```
Código del programa                 Comentarios
DEF INT NÚMERO PIEZAS               ; Variable definida del tipo INT con el nombre
                                      "NÚMERO PIEZAS".
FOR NÚMERO PIEZAS = 0 TO 100        ; Inicia el bucle contador. La variable "NÚMERO
                                      PIEZAS" se incrementa desde el valor inicial
                                      "0" hasta el valor final "100".
G01 …
ENDFOR                              ; Fin del bucle contador.
M30
```

1.12.4 Bucle de programa con la condición en el inicio del bucle (WHILE, ENDWHILE)

Función

En un bucle WHILE, la condición se encuentra en el inicio del bucle. Mientras se cumpla la condición se ejecutará el bucle WHILE.

Sintaxis

```
WHILE <Condición>
...
ENDWHILE
```

Descripción

WHILE:	Inicia el bucle de programa.
ENDWHILE:	Marca el final del bucle y salta atrás hasta el inicio del bucle.
<Condición>:	Condición que debe cumplirse para que se ejecute el bucle WHILE.

Ejemplo

```
Código del programa                          Comentarios
...
WHILE $AA_IW[EJE TALADRADO] > -10            ; Llamada del bucle WHILE con la siguiente
                                               condición: la consigna WKS para el eje de
                                               taladrado debe ser mayor que -10.
G1 G91 F250 AX[EJE TALADRADO] = -1
ENDWHILE
...
```

1.12.5 Bucle de programa con la condición al final del bucle (REPEAT, UNTIL)

Función

En un bucle REPEAT, la condición se encuentra al final del bucle. El bucle REPEAT se ejecuta una primera vez y se repite la ejecución hasta que la condición indicada al final se cumpla.

Sintaxis

```
REPEAT
...
UNTIL <Condición>
```

Descripción

REPEAT:	Inicia el bucle de programa.
UNTIL:	Marca el final del bucle y salta atrás hasta el inicio del bucle.
<Condición>:	Condición que debe cumplirse para que deje de ejecutarse el bucle REPEAT.

Ejemplo

```
Código del programa          Comentarios
...
REPEAT                       ; Llamada del bucle REPEAT.
...
UNTIL ...                    ; Comprobación de si se cumple la condición.
...
```

1.12.6 Ejemplo de programa con estructuras de control imbricadas

```
Código del programa                                  Comentarios
LOOP
 IF NOT $P_SEARCH                                    ; No hay búsqueda de secuencia
 G01 G90 X0 Z10 F1000
 WHILE $AA_IM[X] <= 100
 G1 G91 X10 F500                                     ; Figura de taladrado
 Z-F100
 Z5
 ENDWHILE
 Z10
 ELSE
 MSG("Durante la búsqueda de secuencia no se
taladra")
 ENDIF
 $A_OUT[1] = 1                                       ; Siguiente chapa a taladrar
 G4 F2
ENDLOOP
M30
```

1.13 Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

Función

Canales

Un canal puede ejecutar su propio programa, independientemente de otros canales. Así, los ejes y cabezales asignados temporalmente a dicho canal pueden controlarse a través del programa.

Durante la puesta en servicio se pueden configurar dos o más canales para el control.

Coordinación de programas

Si en la fabricación de una pieza participan varios canales, puede resultar necesario sincronizar las ejecuciones de programas.

Para tal coordinación de programas se dispone de instrucciones especiales (comandos). Estas van integradas en una secuencia propia.

Nota

La coordinación de programas también es posible en el canal propio.

Instrucciones para la coordinación de programas

- Definición absoluta de ruta

	La ruta absoluta se forma según las reglas siguientes:
INIT (n,"/_HUGO_DIR/_N_nombre_MPF") o	- Directorio actual/_N_nombre_MPF "Directorio actual" significa el directorio de piezas seleccionado o el directorio estándar /_N_MPF_DIR.
INIT (n,"/_N_MPF_DIR/_N_nombre_MPF")	- Selección de un determinado programa para la ejecución en un determinado canal: - n: Número del canal, valor según la configuración del control Nombre de programa completo
Ejemplo: INIT(2,"/_N_WKS_DIR/_DIAMANT_MPF") G01F0.1 START INIT (2,"/_N_WKS_DIR/_N_UNTER_1_SPF")	**Hasta SW 3:** Entre un comando **init** (sin sincronización) y una **Marcha CN** debe haber por lo menos una secuencia ejecutable. Para llamadas de subprograma se debe completar "_SPF" en la indicación de ruta.

- Definición relativa de ruta

Ejemplo:	Para la introducción relativa de ruta rigen las mismas reglas que para las llamadas de subprogramas.
INIT(2,"DIAMANT")	
INIT(3,"UNTER_1_SPF")	Para llamadas de subprograma se debe completar "_SPF" en el nombre de programa.

Parámetros

Para el intercambio de datos entre los programas se pueden utilizar las variables disponibles conjuntamente para los canales (variables NCK globales específicas). Por lo demás, los programas se crean por separado para cada canal.

`INIT(n, indicación de ruta, modo de acuse)`	Instrucción para la ejecución por un canal. Selección de un programa determinado con indicación de ruta absoluta o relativa.
`START (n, n)`	Arranque de los programas seleccionados en los otros canales. n,n: Enumeración de los números de canal: valor según la configuración del control
`WAITM (número de meta, n, n, ...)`	Activar la meta "Número de meta" en el propio canal. Terminar la secuencia anterior con parada precisa. Esperar a las metas o lábels con el mismo "Número de meta" en los canales indicados "n" (no hay que introducir el canal propio). Se borra la meta tras la sincronización. Simultáneamente se puede aplicar un máximo de 10 metas (lábels) por canal.
`WAITMC (número de meta, n, n,`	Activar la meta "Número de meta" en el propio canal. La parada precisa sólo se activa si los otros canales no han alcanzado aún la meta. Esperar a la meta con el mismo "Número de meta" en los canales indicados "n" (no hay que introducir el canal propio). Tan pronto se alcance la meta "Número de meta" en los canales indicados, continuar la ejecución sin finalizar la parada precisa.

`WAITE (n, n, ...)`	Esperar el fin de programa de los canales indicados (no indicar el canal propio). Ejemplo: programación de un tiempo de espera después del comando de arranque.
	`N30 START(2)`
	`N31 G4 F0.01`
	`N40 WAITE(2)`

SETM (número de meta, número de meta,	Aplicar las metas "Número de meta" en el propio canal, sin influir la ejecución actual. SETM () permanece vigente tras RESET y NC-START.
CLEARM (número de meta, número de meta,	Borrar las metas "Número de meta" en el canal propio, sin influir en la ejecución actual. Todas las metas del canal se pueden borrar con CLEARM(). CLEARM (0) borra la meta "0". CLEARM() permanece vigente tras RESET y NC-START.
n	Número o nombre de canal correspondiente

Nota

Todas los comandos anteriores deben encontrarse en secuencias independientes.

La cantidad de metas depende de la CPU utilizada.

Números de canal

Para los canales que se desea coordinar pueden indicarse hasta 10 números de canal (valor entero).

Nombres de canal

Los nombres de canal deben convertirse en números mediante una variable (ver apartado "Variable y parámetro de cálculo"), o bien pueden programarse los nombres de canal definidos mediante $MC_CHAN_NAME en lugar de números de canal (identificador o palabra reservada). Los nombres definidos deben corresponder a las convenciones del lenguaje CN (es decir, los dos primeros caracteres deben ser letras o caracteres de subrayado).

⚠ PRECAUCIÓN

La asignación de números debe protegerse contra modificaciones involuntarias.

Los nombres no deben existir previamente en el CN con otro significado, p. ej. como palabra reservada, instrucción de programación (comando), nombre de eje, etc.

SETM() y CLEARM()

`SETM()` y `CLEARM()` se pueden programar también desde una acción síncrona. Ver el apartado "Activar/borrar metas de espera: SETM CLEARM"

Ejemplo

El canal con nombre "MÁQUINA" debe recibir el número de canal 1,

El canal con nombre "CARGADOR" debe recibir el número de canal 2:

```
DEF INT MÁQUINA=1, CARGADOR=2
```

A las variables se les asigna el mismo nombre de los canales.

Así, se tiene, por ejemplo, la siguiente instrucción `START`:

```
START(MÁQUINA)
```

Ejemplo: Coordinación de programas

Canal 1:

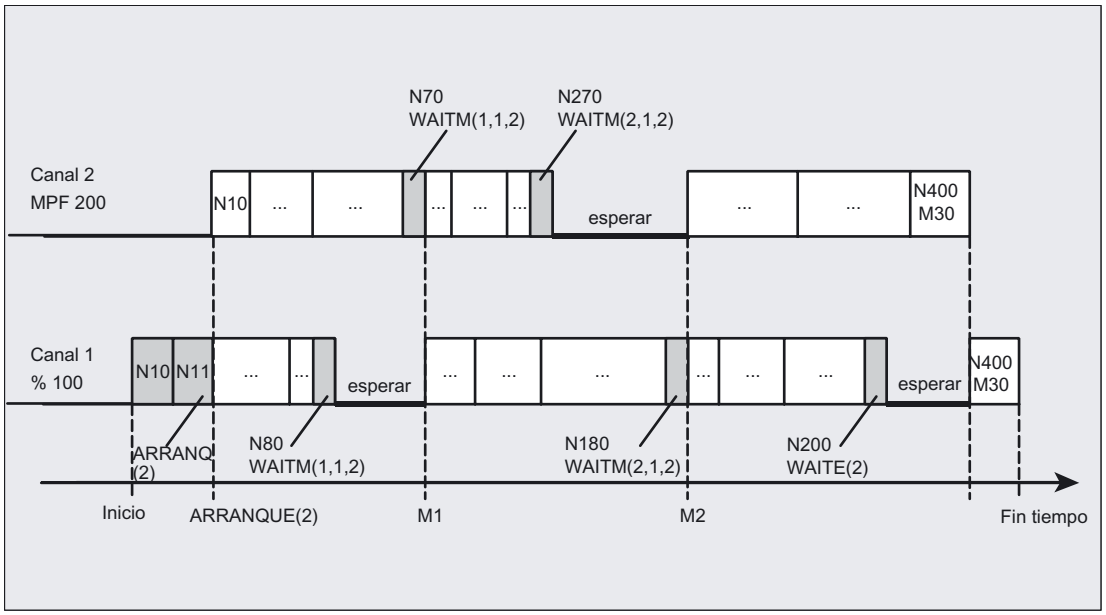
_N_MPF100_MPF

```
Código del programa          Comentarios
N10 INIT(2,"MPF200")
N11 START(2)                 ; Ejecutar en el canal 2
...
N80 WAITM(1,1,2)             ; Esperar la meta WAIT 1 en el canal 1 y en
                               el canal 2, ejecución subsiguiente en canal 1
...
N180 WAITM(2,1,2)            ; Esperar la meta WAIT 2 en el canal 1 y en
                               el canal 2, ejecución subsiguiente en canal 1
...
N200 WAITE(2)                ; Esperar el fin de programa del canal 2
N201 M30                     ; Fin del programa canal 1, final total
...
```

Canal 2:

_N_MPF200_MPF

```
Código del programa    Comentarios
;$PATH=/_N_MPF_DIR
                       ; Ejecutar en el canal 2
N70 WAITM(1,1,2)       ; Esperar la meta WAIT 1 en el canal 1 y en el canal 2,
                         ejecución subsiguiente en canal 1
...
N270 WAITM(2,1,2)      ; Esperar la meta WAIT 2 en el canal 1 y en el canal 2,
                         ejecución subsiguiente en canal 2
...
N400 M30               ; Fin de programa del canal 2
```



Ejemplo: Programa de pieza

Código del programa
`N10 INIT(2,"/_N_WKS_DIR/_N_EJE1_WPD/_N_MECANIZ1_MPF")`

Ejemplo: Comando INIT con ruta de acceso relativa

En el canal 1 está seleccionado el programa `/_N_MPF_DIR/_N_MAIN_MPF`

Código del programa	**Comentarios**
`N10 INIT(2,"MYPROG")`	`; Seleccionar el programa /_N_MPF_DIR/_N_MYPROG_MPF en el canal 2`

Ejemplo: Nombre y número de canal con variable entera

```
$MC_CHAN_NAME[0]= "CHAN_X" ;Nombre del 1er canal
$MC_CHAN_NAME[1]= "CHAN_Y" ;nombre del 2º canal
```

Código del programa	Comentarios
START(1, 2)	; Ejecutar el arranque en el 1er y 2º canal

De forma análoga, programación con los identificadores de canal:

Código del programa	Comentarios
START(CHAN_X, CHAN_Y)	; Ejecutar el arranque en el 1er y 2º canal
	; Los identificadores canal_X y canal_Y representan internamente a los números de canal 1 y 2 mediante el dato de máquina $MC_CHAN_NAME. Por consiguiente, también ejecutan un arranque en el 1er y 2º canal.

Programación con variable entera:

Código del programa	Comentarios
DEF INT chanNo1, chanNo2)	; Definir el número de canal
chanNo1=CHAN_X chanNo2=CHAN_Y	
START(chanNo1, chanNo2)	

1.14 Rutina de interrupción (ASUP)

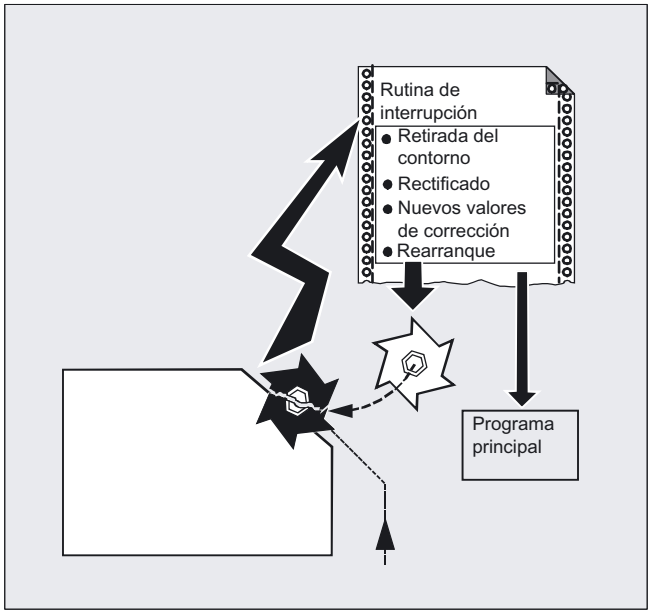
1.14.1 Función de una rutina de interrupción

Nota

Los términos "Subprograma asíncrono (ASUP)" y "Rutina de interrupción", que aparecen alternándose en la siguiente descripción, designan la misma funcionalidad.

Función

La función de una rutina de interrupción se ilustra mediante un ejemplo típico:



Durante el mecanizado se detecta rotura de herramienta. En este instante se desencadena una señal que para el proceso de mecanizado y simultáneamente inicia un subprograma, denominado rutina de interrupción. Dicha rutina contiene todas las instrucciones necesarias que se deben ejecutar en este caso.

Ejecutado el subprograma (y con ello restablecida la disponibilidad operacional), el control numérico retorna al programa principal y continúa el mecanizado – dependiendo del comando `REPOS` – en el punto donde tuvo lugar la interrupción (ver "Reposicionamiento en el contorno (Página 502)").

⚠ PRECAUCIÓN

Si no hay ningún comando `REPOS` programado en el subprograma, se colocará en el punto final de la secuencia que sigue a la secuencia interrumpida.

Bibliografía

Manual de funciones, Funciones básicas; BAG/GMO, canal, modo de programa, comportamiento Reset (K1), capítulo: "Subprogramas asíncronos (ASUP), rutinas de interrupción"

1.14.2 Creación de una rutina de interrupción

Creación de una rutina de interrupción como subprograma

La rutina de interrupción se define como un subprograma en la cabecera de definición.

Ejemplo:

Código del programa	**Comentarios**
PROC RETIR_Z	; Nombre del programa "ABHEB_Z"
N10 ...	; A continuación siguen las secuencias CN.
...	
N50 M17	; Para concluir, final del programa y regreso al programa principal.

Almacenamiento de las funciones G modales (SAVE)

La rutina de interrupción puede identificarse durante la definición con SAVE.

El atributo SAVE hace que se guarden las funciones G modales activas antes de la llamada de la rutina de interrupción y se reactiven de nuevo antes de finalizar dicha rutina (ver "Subprogramas con mecanismo SAVE (SAVE) (Página 164)").

De esta forma es posible continuar con la ejecución en el punto de interrupción tras finalizar la rutina de interrupción.

Ejemplo:

Código del programa
PROC RETIR_Z SAVE
N10 ...
...
N50 M17

Asignación de otras rutinas de interrupción (SETINT)

Dentro de una rutina de interrupción se pueden programar instrucciones SETINT (ver "Asignación y arranque de una rutina de interrupción (SETINT)" (Página 117)) y activar de este modo más rutinas de interrupción. Su ejecución se activa sólo con las correspondientes señales de entrada.

Bibliografía

Información adicional para la creación de subprogramas en el apartado "Técnica de subprogramas, macros".

1.14.3 Asignar e iniciar una rutina de interrupción (SETINT, PRIO, BLSYNC)

Función

El control dispone de señales (entrada 1…8), que provocan la interrupción del programa en curso y pueden iniciar la correspondiente rutina de interrupción.

Con el comando `SETINT` del programa de pieza se puede asignar qué entrada debe arrancar qué programa.

Si en el programa de pieza existen varias instrucciones `SETINT` y pueden por ello producirse varias señales al mismo tiempo, es preciso asignar valores de prioridad a las rutinas de interrupción que definan el orden de ejecución: `PRIO=<Valor>`

Si se produce la activación de nuevas señales durante la ejecución de una rutina de interrupción, las rutinas con mayor prioridad interrumpen la ejecución de la rutina actual.

Sintaxis

```
SETINT(<n>) PRIO=<Valor> <NOMBRE>
SETINT(<n>) PRIO=<Valor> <NAME> BLSYNC
SETINT(<n>) PRIO=<Valor> <NOMBRE> LIFTFAST
```

Descripción

`SETINT(<n>)`:	Comando: asignar la entrada <n> a una rutina de interrupción. La rutina de interrupción asignada arranca cuando se conecta la entrada <n>.	
	Nota: si se asigna una rutina nueva a una entrada ya ocupada, la asignación anterior dejará de ser efectiva automáticamente.	
<n>:	Parámetros: número de entrada	
	Tipo:	INT
	Rango de valores:	1 ... 8
`PRIO= :`	Comando: Definición de la prioridad	
`<Valor>`:	Valor de prioridad	
	Tipo:	INT
	Rango de valores:	1 ... 128
		La prioridad 1 corresponde a la máxima prioridad.
`<NOMBRE>`:	Nombre del subprograma (rutina de interrupción) que debe ejecutarse.	

BLSYNC:	Si la instrucción SETINT se programa junto con BLSYNC, al llegar la señal de interrupción la secuencia de programa en curso se sigue ejecutando y sólo después se inicia la rutina de interrupción.
LIFTFAST:	Si la instrucción SETINT se programa junto con LIFTFAST, al llegar la señal de interrupción antes de que se inicie la rutina de interrupción, se realiza una "retirada rápida de la herramienta del contorno" (ver "Retirada rápida del contorno (SETINT LIFTFAST, ALF) (Página 121)").

Ejemplos

Ejemplo 1: Asignar rutinas de interrupción y definir prioridad

```
Código del programa                    Comentarios
...
N20 SETINT(3) PRIO=1 ABHEB_Z           ; Cuando se conecta la entrada 3, debe arran-
                                         carse la rutina de interrupción "ABHEB_Z".
N30 SETINT(2) PRIO=2 ABHEB_X           ; Cuando se conecta la entrada 2, debe arran-
                                         carse la rutina de interrupción "ABHEB_X".
...
```

Las rutinas de interrupción se ejecutan una tras otra siguiendo el orden de los valores de prioridad cuando las entradas están presentes al mismo tiempo: en primer lugar "ABHEB_Z", a continuación "ABHEB_X".

Ejemplo 2: Reasignar la rutina de interrupción

```
Código de programa                     Comentarios
...
N20 SETINT(3) PRIO=2 ABHEB_Z           ; Cuando se conecta la entrada 3, debe arran-
                                         carse la rutina de interrupción "ABHEB_Z".
…
N120 SETINT(3) PRIO=1 RETIR_X          ; Se asigna una nueva rutina de interrupción a
                                         la entrada 3: en lugar de "ABHEB_Z", debe
                                         arrancar "ABHEB_X" cuando se conecta la
                                         entrada 3.
```

1.14.4 Desactivación/reactivación de la asignación de una rutina de interrupción (DISABLE, ENABLE)

Función

Una instrucción `SETINT` puede desactivarse con `DISABLE` y volver a activarse con `ENABLE` sin que se pierda la asignación entrada → rutina de interrupción.

Sintaxis

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

Descripción

`DISABLE(<n>)`:	Comando: **desactivar** la asignación de rutinas de interrupción de la entrada <n>
`ENABLE(<n>)`:	Comando: **reactivar** la asignación de rutinas de interrupción de la entrada <n>
<n>:	Parámetros: número de entrada
	Tipo: INT
	Rango de valores: 1 ... 8

Ejemplo

```
Código del programa                    Comentarios
...
N20 SETINT(3) PRIO=1 ABHEB_Z           ; Cuando se conecta la entrada 3, debe arran-
                                         carse la rutina de interrupción "ABHEB_Z".
...
N90 DISABLE(3)                         ; Se desactiva la instrucción SETINT de N20.
...
N130 ENABLE(3)                         ; Se reactiva la instrucción SETINT de N20.
...
```

1.14.5 Borrado de la asignación de una rutina de interrupción (CLRINT)

Función

Una asignación entrada → rutina de interrupción definida con `SETINT` puede borrarse con `CLRINT`.

Sintaxis

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

Descripción

`CLRINT(<n>)`:	Comando: borrar la asignación de rutinas de interrupción de la entrada <n>	
`<n>`:	Parámetros: número de entrada	
	Tipo:	INT
	Rango de valores:	1 ... 8

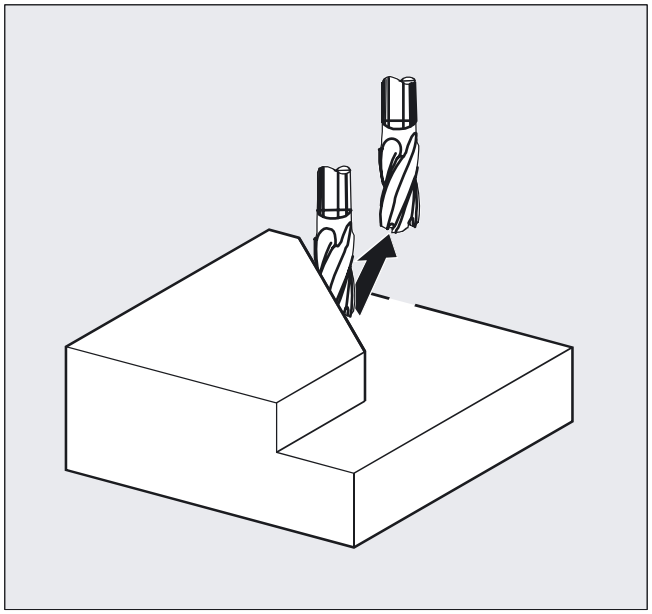
Ejemplo

Código del programa	**Comentarios**
...	
N20 SETINT(3) PRIO=2 ABHEB_Z	;
...	
N50 CLRINT(3)	; Se ha borrado la asignación entre la entrada "3" y la rutina de interrupción "ABHEB_Z".
...	

1.14.6 Retirada rápida del contorno (SETINT LIFTFAST, ALF)

Función

En una instrucción `SETINT` con `LIFTFAST` la herramienta se retira mediante la retirada rápida del contorno de la pieza al conectar la entrada.



El proceso subsiguiente depende de si la instrucción `SETINT` contiene una rutina de interrupción además de `LIFTFAST`:

Con rutina de interrupción:	**Después** de la retirada rápida, se ejecuta la rutina de interrupción.
Sin rutina de interrupción:	La ejecución se para con alarma después de la retirada rápida.

Sintaxis

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

Descripción

SETINT(<n>):	Comando: asignar la entrada <n> a una rutina de interrupción. La rutina de interrupción asignada arranca cuando se conecta la entrada <n>.
<n>:	Parámetros: número de entrada
	Tipo: INT
	Rango de valores: 1 ... 8
PRIO= :	Definición de la prioridad
<Valor>:	Valor de prioridad
	Rango de valores: 1 ... 128
	La prioridad 1 corresponde a la máxima prioridad.
<NOMBRE>:	Nombre del subprograma (rutina de interrupción) que debe ejecutarse.
LIFTFAST:	Comando: retirada rápida del contorno
ALF=… :	Comando: Dirección de desplazamiento programable (contenida en la secuencia de desplazamiento)
	En relación con las opciones de programación con ALF ver el tema "Sentido de desplazamiento en la retirada rápida del contorno (Página 124)".

Limitaciones

Comportamiento con frame activo con simetría especular

Antes de determinarse la dirección de retirada, se comprueba si está activo algún frame con simetría especular. En ese caso se intercambian a izquierda y derecha con relación a la dirección de la tangente, en la dirección de retirada. Los componentes en el sentido de la herramienta no se simetrizan. Este comportamiento se activa con el ajuste de DM:

DM21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

Ejemplo

Una herramienta rota debe sustituirse automáticamente por otra herramienta idéntica. El mecanizado debe continuar con la nueva herramienta.

Programa principal:

Programa principal	**Comentarios**
N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST	; Si se conecta la entrada 1, la herramienta se retira inmediatamente mediante retirada rápida (nº de código 7 para la corrección de radio de herramienta G41) del contorno. A continuación se ejecuta la rutina de interrupción "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	

Subprograma:

Subprograma	**Comentarios**
PROC CAMBIO_H SAVE	; Subprograma con almacenamiento del estado operativo actual
N10 G0 Z100 M5	; Desplazamiento a posición de cambio de herramienta, parada del cabezal
N20 T11 M6 D1 G41	; Cambio de herramienta
N30 REPOSL RMB M3	; Rearranque del contorno y salto atrás hasta el programa principal (se programa en una secuencia)

1.14.7 Sentido de desplazamiento en la retirada rápida del contorno

Movimiento de retroceso

El plano del movimiento de retroceso se define mediante el siguiente código G:

- `LFTXT`

  El plano del movimiento de retroceso se define a partir de la tangente de trayectoria y de la dirección de herramienta (ajuste estándar).

- `LFWP`

  El plano del movimiento de retroceso es el plano de trabajo activo que se selecciona con los códigos G `G17`, `G18` o `G19`. La dirección del movimiento de retirada es independiente de la tangente a la trayectoria. Esto permite programar movimientos de retirada rápida que discurren en paralelo a un eje.

- `LFPOS`

  Retirada del eje declarado con `POLFMASK`/`POLFMLIN` a la posición de eje absoluta programada con `POLF`.

  `ALF` no influye en la dirección de retirada para varios ejes, como tampoco para varios ejes en relación lineal.

  Bibliografía:
  Manual de programación, Fundamentos; capítulo: "Retirada rápida para tallado de roscas"

Sentido de desplazamiento programable (ALF=…)

La dirección se programa en el plano de retirada con `ALF` en pasos discretos de 45 grados.

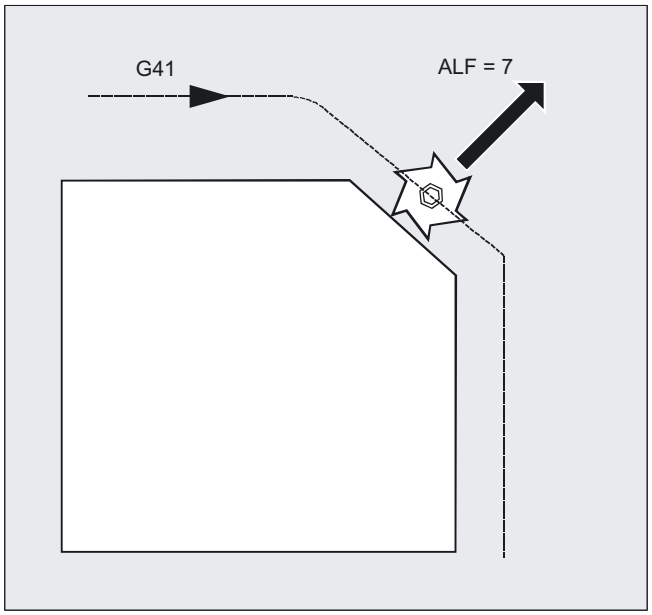
Los posibles sentidos de desplazamiento para estos casos se definen en el control numérico mediante códigos numéricos especiales; que a su vez sirven para poder llamarlos.

Ejemplo:

Código del programa

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

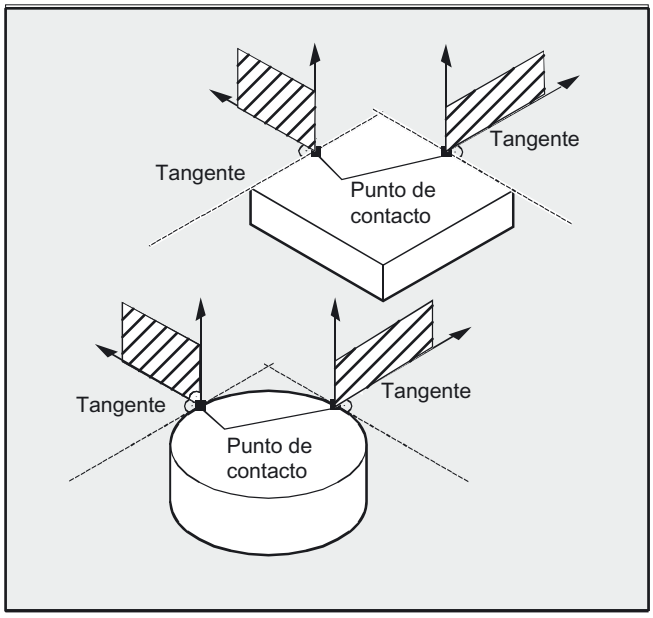
La herramienta se retira con `G41` activo (dirección de mecanizado a la izquierda del contorno) en dirección perpendicular al contorno.



Plano de referencia para la descripción de los sentidos de desplazamiento con LFTXT

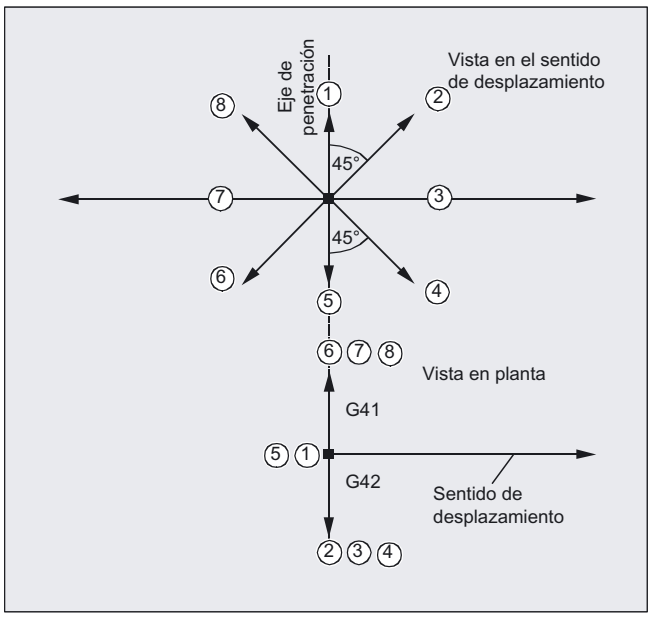
En el punto de contacto entre la herramienta y el contorno programado se define un plano de trabajo que sirve como referencia para definir los códigos numéricos de los movimientos de retirada.

El plano de referencia queda definido por el eje longitudinal de la herramienta (eje de penetración) y un vector perpendicular a dicho eje y a su vez perpendicular a la tangente al contorno por el punto de contacto.



Números de código con sentidos de desplazamiento con LFTXT

A partir del plano de referencia, encontrará en la siguiente figura los números de código con sentidos de desplazamiento.



Para `ALF=1` la retirada se define en la dirección de la herramienta.

Con `ALF=0` está desactivada la función "Retirada rápida".

⚠ PRECAUCIÓN

Si la corrección de radio de herramienta está activada:

- los códigos 2, 3, 4 con `G41` y
- los códigos 6, 7, 8 con `G42`

no deben utilizarse ya que, en estos casos, la herramienta se desplazaría hacia el contorno y colisionaría con la pieza.

Números de código con sentidos de desplazamiento con LFWP

Con LFWP, la dirección en el plano de mecanizado se asigna de la siguiente manera:

- G17: Plano X/Y

  ALF=1: Retirada en dirección X

  ALF=3: Retirada en dirección Y

- G18: Plano Z/X

  ALF=1: Retirada en dirección Z

  ALF=3: Retirada en dirección X

- G19: Plano Y/Z

  ALF=1: Retirada en dirección Y

  ALF=3: Retirada en dirección Z

1.14.8 Desarrollo del movimiento en rutinas de interrupción

Rutina de interrupción sin LIFTFAST

Los desplazamientos de eje se frenan en la trayectoria hasta alcanzar la parada. A continuación arranca la rutina de interrupción.

La posición de parada se almacena como posición de interrupción y se alcanza al final de la rutina de interrupción en REPOS con RMI

Rutina de interrupción con LIFTFAST

Los desplazamientos de eje se frenan en la trayectoria. Al mismo tiempo, el desplazamiento LIFTFAST se ejecuta como desplazamiento superpuesto. Si el desplazamiento en la trayectoria y el desplazamiento LIFTFAST se han parado, se inicia la rutina de interrupción.

Como posición de interrupción se almacena la posición en el contorno en la que se ha iniciado el movimiento LIFTFAST y, con ello, abandonado la trayectoria.

La rutina de interrupción se comporta con LIFTFAST y ALF=0 de forma idéntica que la rutina de interrupción sin LIFTFAST.

Nota

El valor de retirada de los ejes geométricos en la retirada rápida del contorno puede ajustarse mediante un dato de máquina.

1.15 Intercambio de ejes, intercambio de cabezales (RELEASE, GET, GETD)

Función

Uno o varios ejes o cabezales se pueden interpolar siempre en un solo canal. Si un eje ha de trabajar alternadamente en dos canales diferentes (p. ej., intercambiador de palets), deberá liberarse primero en el canal actual y ser aceptado luego en el otro canal. El eje se intercambia entre los canales.

Ampliaciones del intercambio de ejes

Un eje/cabezal se puede intercambiar con parada de decodificación previa y sincronización entre el preprocesamiento y la marcha principal o, como alternativa, también sin parada de decodificación. Además, el intercambio de ejes también es posible a través de:

- Giro del contenedor de ejes AXCTSWE o AXCTWED con `GET/GETD` implícito.
- Frame con rotación si el eje en cuestión está enlazado de esta forma con otros ejes.
- Acciones síncronas, ver acciones síncronas de desplazamientos, "Intercambio de ejes `RELEASE`, `GET`".

Fabricante de la máquina

Preste atención a las indicaciones del fabricante de la máquina. Para el intercambio de ejes, un eje tiene que estar definido unívocamente en todos los canales a través de datos de máquina configurables, y el comportamiento de intercambio también puede ajustarse mediante la modificación de datos de máquina.

Sintaxis

```
RELEASE (nombre de eje, nombre de eje, ...) o RELEASE (S1)
GET (nombre de eje, nombre de eje, ...) o GET (S2)
GETD(nombre de eje, nombre de eje, ...) o GETD(S3)
```

Con el comando GETD (GET Directly) se toma directamente un eje desde otro canal. Es decir, para este GETD no hay que programar el correspondiente comando RELEASE en otro canal. No obstante, esto significa que se precisa entonces establecer otra comunicación de canal (p. ej., metas de espera).

Descripción

`RELEASE (nombre eje, nombre eje, …)`:	Libera los ejes
`GET (nombre eje, nombre eje, …)`:	Aplica los ejes
`GETD (nombre eje, nombre eje, …)`:	Aplica directamente los ejes
`Nombre eje`:	Asignación de ejes en el sistema: AX1, AX2, … o indicación de los nombres de eje de máquina
`RELEASE(S1)` :	Libera los cabezales S1, S2, ...
`GET(S2)` :	Aplica los cabezales S1, S2, ...
`GETD(S3)` :	Aplicación directa de los cabezales S1, S2, ...

Solicitud GET sin parada de decodificación previa

Si, después de una solicitud GET **sin** parada de decodificación previa, se vuelve a liberar el eje con `RELEASE(eje)` o `WAITP(eje)`, un `GET` posterior produce un `GET` **con** parada de decodificación.

⚠ PRECAUCIÓN

Un eje o cabezal aplicado con el comando GET permanece asignado a este canal aún tras un RESET por tecla o por programa.

Al arrancar nuevamente el programa, los ejes o cabezales cambiados se deben asignar a través del programa en caso de que el eje se requiera en su canal básico.

Con POWER ON se asigna el eje al canal ajustado en los datos de máquina.

Ejemplos

Ejemplo 1: Intercambio de ejes entre dos canales

De 6 ejes se utilizan en el canal 1 para el mecanizado: 1er, 2º, 3er y 4º eje.
El 5º y 6º eje se utilizan en el canal 2 para el cambio de pieza.

El eje 2 debe poder intercambiarse entre ambos canales y estar asignado al canal 1 tras POWER ON.

Programa "MAIN" en el canal 1:

```
Código del programa              Comentarios
INIT (2,"CAMBIAR2")              ; Seleccionar el programa CAMBIAR2 en el canal 2.
N… START (2)                     ; Iniciar el programa en el canal 2.
N… GET (AX2)                     ; Aceptar el eje AX2.
...
N… RELEASE (AX2)                 ; Liberar el eje AX2.
N... WAITM(1,1,2)                ; Esperar la meta WAIT en los canales 1 y 2 para
                                   efectuar la sincronización en ambos canales.
...                              ; Continuación del proceso tras intercambio de eje.
N… M30
```

Programa "CAMBIAR2" en el canal 2:

```
Programación                     Comentarios
N… RELEASE (AX2)
N160 WAITM(1,1,2)                ; Esperar la meta WAIT en los canales 1 y 2 para
                                   efectuar la sincronización en ambos canales.
N150 GET(AX2)                    ; Aceptar el eje AX2.
...                              ; Continuación del proceso tras intercambio de eje.
N… M30
```

Ejemplo 2: Intercambio de ejes sin sincronización

Si el eje no tiene que ser sincronizado, GET no genera parada de decodificación previa.

```
Programación                     Comentarios
N01 G0 X0
N02 RELEASE(AX5)
N03 G64 X10
N04 X20
N05 GET(AX5)                     ; Si no se requiere una sincronización, esta
                                   secuencia no será ejecutable.
N06 G01 F5000                    ; Secuencia no ejecutable.
N07 X20                          ; Secuencia no ejecutable, posición X como en N04.
N08 X30                          ; Primera secuencia ejecutable tras N05.
...
```

Ejemplo 3: Activación de un intercambio de ejes sin parada de decodificación previa

Requisito: El intercambio de ejes sin parada de decodificación previa debe configurarse a través de un dato de máquina.

```
Programación                    Comentarios
N010 M4 S100
N011 G4 F2
N020 M5
N021 SPOS=0
N022 POS[B]=1
N023 WAITP(B)                   ; El eje B pasa a ser el eje neutro.
N030 X1 F10
N031 X100 F500
N032 X200
N040 M3 S500                    ; El eje no inicia ninguna parada de decodificación
                                  previa/REORG.
N041 G4 F2
N050 M5
N099 M30
```

Si el cabezal o el eje B se desplaza inmediatamente después de la secuencia N023 como **eje PLC**, p. ej., a 180 grados y de vuelta a 1 grado, este eje se vuelve a convertir en eje neutro y no produce ninguna parada de decodificación previa en la secuencia `N40`.

Requisito

Condiciones necesarias para el intercambio de ejes

- El eje tiene que estar definido a través de datos de máquina en todos los canales que utilizarán el eje.
- A través del dato de máquina específico del **eje**se tiene que definir a qué canal se asignará el eje después de POWER ON.

Descripción

Liberar eje: RELEASE

Al liberar el eje se ha de tener en cuenta lo siguiente:

1. El eje no puede estar activo en ninguna transformada.
2. En los acoplamientos de ejes (control tangencial) se deben desbloquear todos los ejes del conjunto.
3. Un eje de posicionado concurrente no se puede intercambiar en este estado.
4. En caso de un eje maestro Gantry se intercambian también todos los ejes subsiguientes.
5. En acoplamientos de ejes (arrastre, acoplamiento de valor maestro, reductor electrónico) sólo se puede desbloquear el eje maestro del conjunto.

Aceptar eje: GET

Con este comando tiene lugar el intercambio de ejes propiamente dicho. La responsabilidad sobre el eje reside completamente en el canal donde fue programado el comando.

Efectos de GET:

Intercambio de eje con sincronización:

Un eje tendrá que ser sincronizado siempre que haya sido asignado entretanto a otro canal o al PLC y la sincronización no haya sido efectuada antes de GET, a través de "WAITP", G74 o por borrado de trayecto residual.

- Se ejecuta una parada de decodificación previa (como en STOPRE).
- El mecanizado se interrumpe hasta que haya concluido completamente el intercambio.

GET automático"

Si un eje está disponible por principio en un canal, pero no existe actualmente como "eje de canal", se ejecuta automáticamente el comando GET. Si los ejes ya están sincronizados, no se genera ninguna parada de decodificación previa.

Ajuste modificable del comportamiento de intercambio

El momento de entrega de ejes se puede ajustar como sigue a través de un dato de máquina:

- El intercambio automático de ejes también se produce entre dos canales si el eje ha sido colocado mediante WAITP en un estado neutro (comportamiento como antes)
- Al hacer la solicitud de giro del contenedor de ejes, todos los ejes del contenedor que pueden asignarse al canal que realiza la ejecución pueden recogerse en el canal mediante un comando implícito GET o GETD. Sólo se permite un intercambio de ejes subsiguiente una vez que ha finalizado el giro del contenedor de ejes.
- Se comprueba después de una secuencia intermedia intercalada en la marcha principal si es necesaria o no una reorganización. Sólo si los estados de ejes de dicha secuencia **no** coinciden con los estados de ejes actuales es necesario proceder a una reorganización.
- En lugar de una secuencia GET con parada de decodificación previa y sincronización entre el preprocesamiento y la marcha principal, un intercambio de ejes también se puede realizar sin parada de decodificación previa. Entonces sólo se genera una secuencia intermedia con la solicitud GET. En la marcha principal se comprueba, en la ejecución de esta secuencia, si los estados del eje en la secuencia coinciden con los estados actuales del eje.

Para más información sobre la funcionalidad de un intercambio de ejes o cabezales, ver /FB2/ Manual de funciones de ampliación, BAG, canales, intercambio de ejes (K5).

1.16 Transferir el eje a otro canal (AXTOCHAN)

Función

Con la instrucción de programación (comando) `AXTOCHAN` se puede solicitar un eje para transferirlo a otro canal. El eje se puede traer al correspondiente canal desde el programa de pieza CN, así como desde una acción síncrona.

Sintaxis

```
AXTOCHAN(nombre de eje,nombre de canal[,nombre de eje,número de
canal[,...]])
```

Descripción

`AXTOCHAN`:	Solicitar un eje para un determinado canal
`Nombre eje`:	Asignación de ejes en el sistema: X, Y,… o indicación de los nombres de eje de máquina afectados. El canal que realiza la ejecución no necesita ser el canal propio ni tampoco el canal que posee actualmente el derecho de interpolación para el eje
`Número de canal`:	Número del canal al que debe asignarse el eje

Nota

Eje de posicionado concurrente y eje controlado exclusivamente por el PLC

Un eje PLC no puede cambiar de canal como eje de posicionado concurrente. Un eje controlado exclusivamente por el PLC no se puede asignar al programa CN.

Bibliografía
Manual de funciones de ampliación; Ejes de posicionado (P2)

Ejemplo

AXTOCHAN en el programa CN

Los ejes X e Y son conocidos en el 1er y en el 2º canal. Actualmente, el canal 1 tiene el derecho de interpolación y en el canal 1 se inicia el siguiente programa:

```
Código del programa          Comentarios
N110 AXTOCHAN(Y,2)           ; Desplazar eje Y al 2º canal.
N111 M0
N120 AXTOCHAN(Y,1)           ; Recuperar el eje Y (neutro).
N121 M0
N130 AXTOCHAN(Y,2,X,2)       ; Desplazar el eje Y y el eje X al 2º canal (ejes
                               neutros).
N131 M0
N140 AXTOCHAN(Y,2)           ; Desplazar eje Y al 2º canal (programa CN).
N141 M0
```

Información adicional

AXTOCHAN en el programa CN

En este caso, sólo se ejecuta un GET si se realiza una solicitud del eje para el programa CN en el canal propio, esperando así también el cambio de estado efectivo. Si el eje se solicita para otro canal o se tiene que convertir en el eje neutro en el canal propio, entonces sólo se envía la correspondiente solicitud.

AXTOCHAN desde una acción síncrona

Si se solicita un eje para el canal propio, AXTOCHAN se refleja en un GET de una acción síncrona desde una acción síncrona. En este caso, el eje se convierte en eje neutro en la primera solicitud para el canal propio. Con la segunda solicitud, el eje se asigna al programa CN de forma análoga a la petición GET en el programa CN. Sobre la solicitud GET desde una acción síncrona, ver el apartado "Acciones síncronas a desplazamientos".

1.17 Activar los datos de máquina (NEWCONF)

Función

El comando NEWCONF permite activar todos los datos de máquina del escalón de activación "NEW_CONFIG". Esta función también se puede activar en la interfaz hombre-máquina (HMI), accionando el pulsador de menú "Activar DM".

Durante la ejecución de la función "NEWCONF" se efectúa una parada de decodificación previa implícita, es decir, se interrumpe el movimiento interpolado.

Sintaxis

```
NEWCONF
```

Descripción

NEWCONF: Comando para activar todos los datos de máquina del escalón de activación "NEW_CONFIG"

Ejecutar NEWCONF desde el programa de pieza en varios canales

Si se modifican datos de máquina axiales desde el programa de pieza y se activan después con NEWCONF, el comando NEWCONF sólo activa los datos de máquina que producen modificaciones para el canal del programa de pieza.

Nota

Para asegurar que se activan todas las modificaciones, el comando NEWCONF se tiene que ejecutar en cada canal en el que se calculan actualmente los ejes o funciones afectados por los datos de máquina modificados.

En el caso de NEWCONF no se activa ningún dato de máquina axial.

Para ejes controlados por el PLC debe realizarse un RESET por eje.

Ejemplo

Fresado: mecanizar posición de taladrado con diferentes tecnologías

```
Código del programa                    Comentarios
N10 $MA_CONTOUR_TOL[AX]=1.0            ; Modificar dato de máquina.
N20 NEWCONF                            ; Activar datos de máquina.
...
```

1.18 Escribir fichero (WRITE)

Función

Con el comando `WRITE` pueden escribirse secuencias/datos del programa de pieza al final del fichero indicado (fichero de protocolo) o del programa de pieza que actualmente se está ejecutando. Las secuencias o los datos se insertan al final de fichero, es decir, tras `M30`.

Nota

Si no existe en el CN, se crea el fichero que se quiere escribir con el comando `WRITE`.

Su ubicación es la memoria CN estática. En SINUMERIK 840D sl es la tarjeta CompactFlash. Frente a SINUMERIK 840D, el tiempo de ejecución del comando `WRITE` aumenta aprox. 75 ms.

Si en el disco duro existe un fichero del mismo nombre, se sobrescribe el mismo tras cerrar el fichero (en el CN). Remedio: Modificar el nombre en el CN en el campo de manejo "Servicios" con el pulsador de menú "Propiedades".

Requisito

El nivel de protección ajustado actualmente debe ser igual o superior al derecho WRITE del fichero. Si no es el caso, se deniega el acceso con aviso de error (valor de retorno de las variables de error = 13).

Sintaxis

```
DEF INT <Error>
WRITE(<Error>,"<Nombre de fichero>","<Secuencia/datos>")
```

Descripción

`WRITE`:	Comando para insertar una secuencia o datos al final del fichero indicado		
`<Error>`:	Variable para la devolución del valor erróneo		
	Tipo:	INT	
	Valor:	0	sin errores
		1	Ruta no autorizada
		2	Ruta no encontrada
		3	Fichero no encontrado
		4	Tipo de fichero incorrecto
		10	Fichero lleno
		11	El fichero está siendo utilizado
		12	No hay recursos libres
		13	Acceso no autorizado
		20	Otros errores

`<Nombre de fichero>`:

Nombre del fichero en el que se deben insertar la secuencia indicada o los datos indicados

Tipo: STRING

Al indicar el nombre de fichero deben tenerse en cuenta los siguiente puntos:

- El nombre de fichero indicado no debe contener espacios en blanco ni caracteres de control (caracteres con código ASCII ≤ 32), ya que, de lo contrario, se interrumpe el comando `WRITE` con el código de error 1 "Ruta no autorizada".
- El nombre del fichero se puede indicar con la ruta y la identificación del fichero:
  - Indicación de rutas

    Las indicaciones de rutas han de ser absolutas, es decir, deben comenzar con "/".

    Sin indicación de ruta, el fichero se guarda en el directorio actual (= directorio del programa seleccionado).
  - Identificación de fichero

    Si el nombre del fichero no contiene ninguna identificación de dominio ("_N_"), se completará de forma correspondiente.

    Si el cuarto carácter del nombre del fichero empezando por el final es un guión bajo "_", los siguientes tres caracteres se interpretarán como identificación de fichero. Para poder utilizar el mismo nombre de fichero en todos los comandos de fichero, p. ej. mediante una variable del tipo STRING, solamente deben utilizarse las identificaciones de fichero _SPF y _MPF.

    Si no se ha indicado ninguna identificación "_MPF" o "_SPF", se asigna automáticamente _MPF.
- La longitud del nombre del fichero puede tener, como máximo, 32 bytes; la longitud de la indicación de ruta, como máximo, 128 bytes.

Ejemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

`<Secuencia/datos>`:

Secuencia o datos que se deben insertar en el fichero indicado.

Tipo: STRING

Nota:
internamente se añade LF, es decir, la cadena de caracteres se alarga 1 carácter.

Limitaciones

- **Tamaño máximo de fichero (→ fabricante de la máquina)**

  El tamaño máximo posible de ficheros de protocolo se ajusta con el dato de máquina:

  DM11420 $MN_LEN_PROTOCOL_FILE

  El tamaño máximo de fichero es válido para todos los ficheros creados con el comando `WRITE`. Si se supera, se emite un aviso de error y la secuencia o los datos no se guardan. Si la memoria tiene suficiente capacidad, se puede crear un nuevo fichero.

Ejemplos

Ejemplo 1: Comando WRITE sin indicación de ruta absoluta

```
Código de programa                                          Comentarios
N10 DEF INT ERROR                                           ; Definición de las variables
                                                              de error.
N20 WRITE(ERROR,"TEST1","PROTOCOLO DEL 7.2.97")             ; Escribir el texto "PROTOCOLO
                                                              DEL 7.2.97" en el fichero
                                                              _N_TEST1_MPF.
N30 IF ERROR                                                ; Evaluación de errores.
N40 MSG ("Error en comando WRITE:" << ERROR)
N50 M0
N60 ENDIF
...
```

Ejemplo 2: Comando WRITE con indicación de ruta absoluta

```
Código de programa
...
WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","PROTOCOLO DEL 7.2.97")
...
```

1.19 Borrar fichero (DELETE)

Función

Con el comando DELETE se pueden borrar todos los ficheros, independientemente de si han sido creados o no con el comando WRITE. También los ficheros creados con un nivel de acceso superior se pueden borrar con DELETE.

Sintaxis

```
DEF INT <Error>
DELETE(<Error>,"<Nombre de fichero>")
```

Descripción

DELETE:	Comando para borrar el fichero indicado		
<Error>:	Variable para la devolución del valor erróneo		
	Tipo:	INT	
	Valor:	0	Sin errores
		1	Ruta no autorizada
		2	Ruta no encontrada
		3	Fichero no encontrado
		4	Tipo de fichero incorrecto
		11	El fichero está siendo utilizado
		12	No hay recursos libres
		20	Otros errores
<Nombre de fichero>:	Nombre del fichero que se desea borrar		
	Tipo:	STRING	

Al indicar el nombre de fichero deben tenerse en cuenta los siguiente puntos:

- El nombre de fichero indicado no debe contener espacios en blanco ni caracteres de control (caracteres con código ASCII ≤ 32), ya que, de lo contrario, se interrumpe el comando DELETE con el código de error 1 "Ruta no autorizada".
- El nombre del fichero se puede indicar con la ruta y la identificación del fichero:
  - Indicación de rutas

    Las indicaciones de rutas han de ser absolutas, es decir, deben comenzar con "/".

    Sin indicación de ruta, el fichero se busca en el directorio actual (= directorio del programa seleccionado).
  - Identificación de fichero

    Si el nombre del fichero no contiene ninguna identificación de dominio ("_N_"), se completará de forma correspondiente.

    Si el cuarto carácter del nombre del fichero empezando por el final es un guión bajo "_", los siguientes tres caracteres se interpretarán como identificación de fichero. Para poder utilizar el mismo nombre de fichero en todos los comandos de fichero, p. ej. mediante una variable del tipo STRING, solamente deben utilizarse las identificaciones de fichero _SPF y _MPF.

    Si no se ha indicado ninguna identificación "_MPF" o "_SPF", se asigna automáticamente _MPF.
- La longitud del nombre del fichero puede tener, como máximo, 32 bytes; la longitud de la indicación de ruta, como máximo, 128 bytes.

Ejemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

Ejemplo

Código de programa	Comentarios
N10 DEF INT ERROR	; Definición de las variables de error.
N15 STOPRE	; Parada de decodificación previa.
N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")	; Borrar el fichero TEST1 en el directorio del subprograma.
N30 IF ERROR	; Evaluación de errores.
N40 MSG("Error en comando DELETE:" <<ERROR)	
N50 M0	
N60 ENDIF	

1.20 Leer líneas en fichero (READ)

Función

El comando `READ` lee una o varias líneas del fichero indicado y guarda la información leída en una matriz del tipo STRING. Cada línea leída ocupa un elemento de matriz en este campo.

Nota

El fichero se debe encontrar en la memoria de usuario estática del NCK (sistema de ficheros pasivo).

Requisito

El nivel de protección ajustado actualmente debe ser igual o superior al derecho READ del fichero. Si no es el caso, se deniega el acceso con aviso de error (valor de retorno de las variables de error = 13).

Sintaxis

```
DEF INT <Error>
DEF STRING[<Longitud del string>] <Resultado>[<n>,<m>]
READ(<Error>,"<Nombre de fichero>",<Línea inicial>,<Número de
líneas>,<Resultado>)
```

Descripción

`READ`: Comando para leer líneas del fichero indicado y para guardar estas líneas en un elemento de matriz.

`<Error>`: Variable para la devolución del valor erróneo (parámetro Call-By-Reference)

Tipo: INT

Valor:		
	0	Sin errores
	1	Ruta no autorizada
	2	Ruta no encontrada
	3	Fichero no encontrado
	4	Tipo de fichero incorrecto
	13	Derechos de acceso insuficientes
	21	Línea no presente (parámetro `<Línea inicial>` o `<Número de líneas>` mayor que el número de líneas en el fichero indicado).
	22	La longitud de matriz de las variables de resultado (`<Resultado>`) es demasiado pequeña.
	23	Rango de líneas demasiado grande (se ha seleccionado un parámetro `<Número de líneas>` tan grande que se lee más allá del final del fichero).

`<Nombre de fichero>`:

Nombre del fichero que se va a leer (parámetro Call-By-Value)

Tipo: STRING

Al indicar el nombre de fichero deben tenerse en cuenta los siguiente puntos:

- El nombre de fichero indicado no debe contener espacios en blanco ni caracteres de control (caracteres con código ASCII ≤ 32), ya que, de lo contrario, se interrumpe el comando `READ` con el código de error 1 "Ruta no autorizada".
- El nombre del fichero se puede indicar con la ruta y la identificación del fichero:
  - Indicación de rutas

    Las indicaciones de rutas han de ser absolutas, es decir, deben comenzar con "/".

    Sin indicación de ruta, el fichero se busca en el directorio actual (= directorio del programa seleccionado).
  - Identificación de fichero

    Si el nombre del fichero no contiene ninguna identificación de dominio ("_N_"), se completará de forma correspondiente.

    Si el cuarto carácter del nombre del fichero empezando por el final es un guión bajo "_", los siguientes tres caracteres se interpretarán como identificación de fichero. Para poder utilizar el mismo nombre de fichero en todos los comandos de fichero, p. ej. mediante una variable del tipo STRING, solamente deben utilizarse las identificaciones de fichero _SPF y _MPF.

    Si no se ha indicado ninguna identificación "_MPF" o "_SPF", se asigna automáticamente _MPF.
- La longitud del nombre del fichero puede tener, como máximo, 32 bytes; la longitud de la indicación de ruta, como máximo, 128 bytes.

Ejemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

`<Línea inicial>`:

Línea inicial del área del fichero que se va a leer (parámetro Call-By-Value)

Tipo: INT

Valor:	0	Se lee la cantidad de líneas indicada con el parámetro `<Número de líneas>` antes del final del fichero.
	1 … n	Número de la primera línea que se va a leer.

`<Número de líneas>`:

Número de líneas que se van a leer (parámetro Call-By-Value)

Tipo: INT

<Resultado>: Variable de resultado (parámetro Call-By-Reference)

Elemento de matriz en el que se guarda el texto leído.

Tipo: STRING (longitud máx.: 255)

Si en el parámetro <Número de líneas> están indicadas menos líneas que el tamaño [<n>,<m>] de la matriz de las variables de resultado, el resto de elementos de matriz no se modificará.

La terminación de una línea con el carácter de control "LF" (Line Feed) o "CR LF" (Carriage Return Line Feed) **no** se guarda en la variable de resultado.

Las líneas leídas se cortan cuando la línea es más larga que la longitud de cadena definida. No se señaliza ningún aviso de error.

Nota

No se pueden leer ficheros binarios. Se emite el error "Tipo de fichero incorrecto" (valor de retorno de las variables de error = 4). Los siguientes tipos de fichero no se pueden leer: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Ejemplo

```
Código de programa                                                  Comentarios
N10 DEF INT ERROR                                                   ; Definición de las variables de error.
N20 DEF STRING[255] RESULT[5]                                       ; Definición de las variables de
                                                                      resultado.
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF",1,5,RESULT)            ; Nombre del fichero con identificación
                                                                      de dominio y de fichero y ruta de
                                                                      acceso.
N40 IF ERROR <>0                                                    ; Evaluación de errores.
N50 MSG("ERROR"<<ERROR<<"CON COMANDO READ")
N60 M0
N70 ENDIF
...
```

1.21 Comprobar la presencia de un fichero (ISFILE)

Función

Con el comando `ISFILE` se verifica si un fichero está presente en la memoria de usuario estática del NCK (sistema de ficheros pasivo).

Sintaxis

```
<Resultado>=ISFILE("<Nombre de fichero>")
```

Descripción

`ISFILE`: Comando para verificar si el fichero indicado existe en el sistema de ficheros pasivo.

`<Nombre de fichero>`: Nombre del fichero cuya presencia en el sistema de ficheros pasivo debe verificarse.

Tipo: STRING

Al indicar el nombre de fichero deben tenerse en cuenta los siguiente puntos:

- El nombre de fichero indicado no debe contener espacios en blanco ni caracteres de control (caracteres con código ASCII ≤ 32).
- El nombre del fichero se puede indicar con la ruta y la identificación del fichero:
  - Indicación de rutas

    Las indicaciones de rutas han de ser absolutas, es decir, deben comenzar con "/".

    Sin indicación de ruta, el fichero se busca en el directorio actual (= directorio del programa seleccionado).
  - Identificación de fichero

    Si el nombre del fichero no contiene ninguna identificación de dominio ("_N_"), se completará de forma correspondiente.

    Si el cuarto carácter del nombre del fichero empezando por el final es un guión bajo "_", los siguientes tres caracteres se interpretarán como identificación de fichero. Para poder utilizar el mismo nombre de fichero en todos los comandos de fichero, p. ej. mediante una variable del tipo STRING, solamente deben utilizarse las identificaciones de fichero _SPF y _MPF.

    Si no se ha indicado ninguna identificación "_MPF" o "_SPF", se asigna automáticamente _MPF.
- La longitud del nombre del fichero puede tener, como máximo, 32 bytes; la longitud de la indicación de ruta, como máximo, 128 bytes.

Ejemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

<Resultado>: Variable de resultado para el registro del resultado de prueba

Tipo:	BOOL	
Valor:	TRUE	Fichero presente
	FALSE	Fichero no presente

Ejemplo

Código de programa	**Comentarios**
N10 DEF BOOL RESULT	; Definición de las variables de resultado.
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(RESULT==FALSE)	
N40 MSG("FICHERO NO PRESENTE")	
N50 M0	
N60 ENDIF	
...	

o:

Código de programa	**Comentarios**
N10 DEF BOOL RESULT	; Definición de las variables de resultado.
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(NOT ISFILE("TESTFILE"))	
N40 MSG("FICHERO NO PRESENTE")	
N50 M0	
N60 ENDIF	
...	

1.22 Leer información de fichero (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Función

Mediante los comandos FILEDATE, FILETIME, FILESIZE, FILESTAT y FILEINFO se pueden leer determinadas informaciones de fichero como la fecha o la hora del último acceso de escritura, el tamaño de fichero actual, el estado del fichero o la suma de estas informaciones.

Nota

El fichero se debe encontrar en la memoria de usuario estática del NCK (sistema de ficheros pasivo).

Requisito

El nivel de protección ajustado actualmente debe ser igual o superior al derecho Show del directorio superior. Si no es el caso, se deniega el acceso con aviso de error (valor de retorno de las variables de error = 13).

Sintaxis

```
DEF INT <Error>
DEF STRING[<Longitud de cadena>] <Resultado>
FILE....(<Error>,"<Nombre de fichero>",<Resultado>)
```

Descripción

FILEDATE:	El comando FILEDATE suministra la **fecha** del último acceso de escritura al fichero indicado.
FILETIME:	El comando FILETIME suministra la **hora** del último acceso de escritura al fichero indicado.
FILESIZE:	El comando FILESIZE suministra el **tamaño actual** del fichero indicado.
FILESTAT:	El comando FILESTAT suministra el **estado** del fichero indicado relativo a los derechos de lectura, escritura y ejecución.
FILEINFO:	El comando FILEINFO suministra la **suma de las informaciones de fichero** para el fichero indicado que se pueden leer mediante FILEDATE, FILETIME, FILESIZE y FILESTAT.

`<Error>`: Variable para la devolución del valor erróneo (parámetro Call-By-Reference)

Tipo: INT

Valor:
- 0 Sin errores
- 1 Ruta no autorizada
- 2 Ruta no encontrada
- 3 Fichero no encontrado
- 4 Tipo de fichero incorrecto
- 13 Derechos de acceso insuficientes
- 22 La longitud de cadena de las variables de resultado (`<Resultado>`) es demasiado pequeña.

`<Nombre de fichero>`: Nombre del fichero del que deben leerse las informaciones de fichero.

Tipo: STRING

Al indicar el nombre de fichero deben tenerse en cuenta los siguiente puntos:

- El nombre de fichero indicado no debe contener espacios en blanco ni caracteres de control (caracteres con código ASCII ≤ 32), ya que, de lo contrario, se interrumpe el comando `FILE...` con el código de error 1 "Ruta no autorizada".
- El nombre del fichero se puede indicar con la ruta y la identificación del fichero:
  - Indicación de rutas

    Las indicaciones de rutas han de ser absolutas, es decir, deben comenzar con "/".

    Sin indicación de ruta, el fichero se busca en el directorio actual (= directorio del programa seleccionado).
  - Identificación de fichero

    Si el nombre del fichero no contiene ninguna identificación de dominio ("_N_"), se completará de forma correspondiente.

    Si el cuarto carácter del nombre del fichero empezando por el final es un guión bajo "_", los siguientes tres caracteres se interpretarán como identificación de fichero. Para poder utilizar el mismo nombre de fichero en todos los comandos de fichero, p. ej. mediante una variable del tipo STRING, solamente deben utilizarse las identificaciones de fichero _SPF y _MPF.

    Si no se ha indicado ninguna identificación "_MPF" o "_SPF", se asigna automáticamente _MPF.
- La longitud del nombre del fichero puede tener, como máximo, 32 bytes; la longitud de la indicación de ruta, como máximo, 128 bytes.

Ejemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

`<Resultado>`: Variable de resultado (parámetro Call-By-Reference)

Variable en la que se guarda la información de fichero solicitada.

Tipo: STRING Con: `FILEDATE`
Formato: "dd.mm.aa"
⇒ la longitud de cadena debe ser **8**.
`FILETIME`
Formato: "hh:mm.ss"
⇒ la longitud de cadena debe ser **8**.
`FILESTAT`
Formato: "rwxsd"
(r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete)
⇒ la longitud de cadena debe ser **5**.
`FILEINFO`
Formato: "rwxsd nnnnnnnn dd.mm.aa hh:mm:ss"
⇒ la longitud de cadena debe ser **32**.

INT Con: `FILESIZE`
El tamaño del fichero se indica en bytes.

Ejemplo

Código del programa	**Comentarios**
`N10 DEF INT ERROR`	`; Definición de las variables de error.`
`N20 STRING[32] RESULT`	`; Definición de las variables de resultado.`
`N30 FILEINFO(ERROR,"/_N_MPF_DIR/_N_TESTFILE_MPF",RESULT)`	`; Nombre del fichero con identificación de dominio y de fichero y ruta de acceso.`
`N40 IF ERROR <>0`	`; Evaluación de errores`
`N50 MSG("ERROR"<<ERROR<<"CON COMANDO FILEINFO")`	
`N60 M0`	
`N70 ENDIF`	
`...`	

En la variable de resultado RESULT, este ejemplo podría suministrar, p. ej., el siguiente resultado:

"77777 12345678 26.05.00 13:51:30"

1.23 Cálculo de la suma de control mediante una matriz (CHECKSUM)

Función

Con el comando `CHECKSUM` la suma de control puede calcularse mediante una matriz. Comparando esta suma de control con el resultado de un cálculo de suma de control anterior se puede constatar si los datos de la matriz se han modificado.

Aplicación

Comprobación si, en el desbaste, se ha modificado el contorno de entrada.

Sintaxis

```
DEF INT <Error>
DEF STRING[<Longitud de cadena>] <Suma de control>
DEF ... <Matriz>[<n>,<m>,<o>]
<Error>=CHECKSUM(<Suma de control>,"<Matriz>"[,<Columna
inicial>,<Columna final>])
```

Descripción

`CHECKSUM`: Comando para el cálculo de la suma de control mediante una matriz

`<Error>`: Variable para la devolución del valor erróneo

Tipo: INT

Valor:

0	Sin errores
1	Símbolo no encontrado
2	Ninguna matriz
3	Índice 1 demasiado grande
4	Índice 2 demasiado grande
5	Tipo de datos inválido
10	Rebose de la suma de chequeo

`<Suma de control>`: Variable de resultado para el registro del resultado del cálculo de la suma de control (parámetro Call-By-Reference)

Tipo: STRING

Longitud de cadena necesaria: 16

La suma de control se representa como string de 16 cifras hexadecimales. No obstante, no se indican caracteres de formato.

Ejemplo: "A6FC3404E534047C"

<Matriz>:	Nombre de la matriz mediante la cual debe realizarse la suma de control (parámetro Call-By-Value) Tipo: STRING Longitud máx. de cadena: 32 Las matrices admisibles tienen de 1 a 3 dimensiones y son de los tipos: BOOL, CHAR, INT, REAL, STRING **Nota:** No se permiten matrices de datos de máquina.
<Columna inicial>:	Número de la columna inicial de la matriz para el cálculo de la suma de control (parámetro opcional)
<Columna final>:	Número de la columna final de la matriz para el cálculo de la suma de control (parámetro opcional)

Nota

Los parámetros <Columna inicial> y <Columna final> son opcionales. Si no se indica ningún índice de columna, se forma la suma de control de la matriz completa.

El resultado de la suma de control siempre es unívoco. En caso de modificaciones de un elemento de matriz se obtiene también un string de resultado diferente.

Ejemplo

```
Código del programa                                  Comentarios
N10 DEF INT ERROR                                    ; Definición de las variables de
                                                       error.
N20 DEF STRING[16] MY_CHECKSUM                       ; Definición de las variables de
                                                       resultado.
N30 DEF INT MY_VAR[4,4]                              ; Definición de matrices.
N40 MY_VAR=...
N50 ERROR=CHECKSUM(MY_CHECKSUM,"MY_VAR",0,2)
...
```

En la variable de resultado MY_CHECKSUM, este ejemplo podría suministrar, p. ej., el siguiente resultado:

"A6FC3404E534047C"

1.24 Redondeo (ROUNDUP)

Función

La función "ROUNDUP" permite redondear valores de entrada del tipo REAL (valor fraccionario con punto decimal) al siguiente valor entero superior.

Sintaxis

```
ROUNDUP(<Valor>)
```

Descripción

`ROUNDUP`:	Comando para redondear un valor de entrada
`<Valor>`:	Valor de entrada del tipo REAL

Nota

Los valores de entrada del tipo INTEGER (un número entero) se devuelven sin modificar.

Ejemplos

Ejemplo 1: Distintos valores de entrada y sus resultados de redondeo

Ejemplo	Resultado de redondeo
`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

Ejemplo 2: ROUNDUP en el programa CN

Código del programa

```
N10 X=ROUNDUP(3.5) Y=ROUNDUP(R2+2)
N15 R2=ROUNDUP($AA_IM[Y])
N20 WHEN X=100 DO Y=ROUNDUP($AA_IM[X])
...
```

1.25 Uso de subprogramas

1.25.1 Generalidades

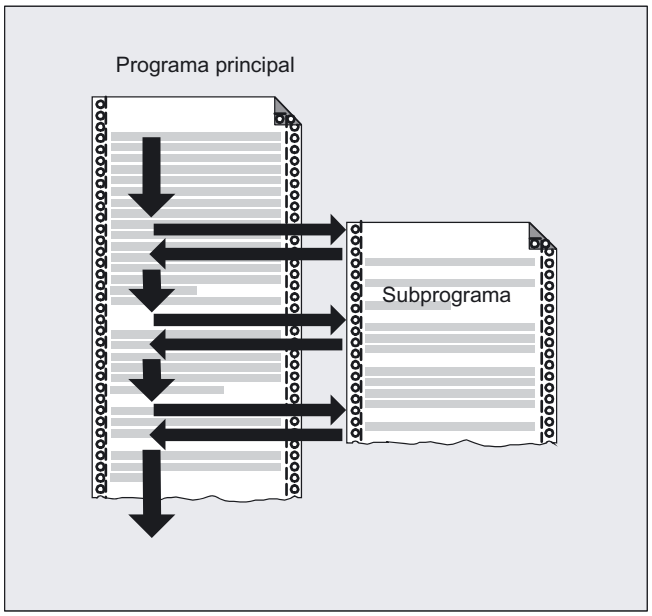
1.25.1.1 Subprograma

Función

El nombre "Subprograma" procede del tiempo en que los programas de pieza estaban divididos de una forma fija en programas principales y subprogramas. Los programas principales eran los programas de pieza que se seleccionaban en el control para la ejecución y luego se iniciaban. Los subprogramas eran programas de pieza que se llamaban desde el programa principal.

Esta división fija ya no existe con el lenguaje CN actual de SINUMERIK. En principio, cualquier programa de pieza puede seleccionarse e iniciarse como programa principal o llamarse como subprograma desde otro programa de pieza.

Con ello, en adelante se denomina subprograma aquel programa de pieza que se llama desde otro programa de pieza.



Aplicación

Como en todos los lenguajes de programación de alto nivel, en el lenguaje CN también se utilizan subprogramas para encapsular secciones de programa que se utilizan varias veces en programas autónomos cerrados en sí mismos.

Los subprogramas ofrecen las siguientes ventajas:

- Mejoran la claridad y legibilidad de los programas.
- Aumentan la calidad reutilizando secciones de programa ya comprobadas.
- Ofrecen la posibilidad de adquirir librerías de proceso específicas.
- Ahorran espacio en memoria.

1.25.1.2 Nombres de subprogramas

Reglas para la denominación

Al poner nombres a los subprogramas deben tenerse en cuenta las siguientes reglas:

- Los dos primeros caracteres deben ser letras (A - Z, a - z).
- Los siguientes caracteres pueden ser letras, cifras (0 - 9) y guiones bajos ("_") combinados libremente.
- Se pueden utilizar 31 caracteres como máximo.

Nota

En el lenguaje CN de SINUMERIK **no** se diferencia entre mayúsculas y minúsculas.

Extensiones del nombre del programa

El nombre de programa asignado durante la creación del programa se amplía en el interior del control con un prefijo y una extensión:

- Prefijo: _N_
- Extensión:
  - Programas principales: _MPF
  - Subprogramas: _SPF

Utilización del nombre de programa

Al utilizar el nombre de programa, p. ej. en la llamada de un subprograma, son posibles todas las combinaciones de prefijos, nombres de programas y extensiones.

Ejemplo:

El subprograma con el nombre "SUB_PROG" puede iniciarse con las siguientes llamadas:

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

Nota

Programas y subprogramas con el mismo nombre

Si hay programas principales (.MPF) y subprogramas (.SPF) con el mismo nombre, al utilizar el nombre de programa en el programa de pieza debe indicarse la extensión correspondiente para identificar unívocamente el programa.

1.25.1.3 Imbricación de subprogramas

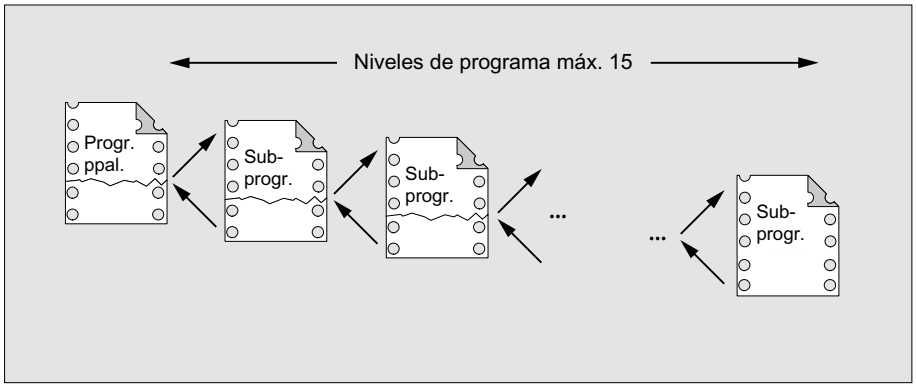
Un programa principal puede llamar a subprogramas que, a su vez, pueden llamar a otros subprogramas. Con ello, las ejecuciones de los programas están imbricadas entre sí. En ese caso, cada programa se ejecuta en un nivel de programa propio.

Profundidad de imbricado

El lenguaje CN ofrece hoy en día 16 niveles de programas. El programa principal siempre se ejecuta en el nivel de programa 0, que es el más alto. Un subprograma se ejecuta siempre en el nivel de programa inmediatamente inferior que sigue a la llamada. Así pues, el nivel de programa 1 es el primer nivel de subprograma.

División de los niveles de programa:

- Nivel de programa 0: nivel de programa principal
- Nivel de programa 1 - 15: nivel de subprograma 1 - 15



Rutinas de interrupción (ASUP)

Si en el contexto de una rutina de interrupción se llama a un subprograma, éste no se ejecuta en el nivel de programa (n) actual activo en el canal, sino también en el nivel de programa inmediatamente inferior (n+1). Para que esto también sea posible en el nivel de programa más bajo, están disponibles 2 niveles de programa adicionales (16 y 17) en combinación con las rutinas de interrupción.

Si se necesitan más de 2 niveles de programa, esto debe tenerse en cuenta explícitamente en la estructuración del programa de pieza ejecutado en el canal. Es decir, el número máximo de niveles de programa que se pueden utilizar es aquél que deje niveles de programa suficientes para la ejecución de rutinas de interrupción.

Si la ejecución de rutinas de interrupción necesita, p. ej. 4 niveles de programa, el programa de pieza debe estructurarse de tal manera que ocupe como máximo el nivel de programa 13. Si se produce entonces una interrupción, ésta tiene a su disposición los 4 niveles de programa necesarios (del 14 al 17).

Ciclos de Siemens

Los ciclos de Siemens necesitan 3 niveles de programa. En consecuencia, un ciclo de Siemens debe llamarse a más tardar en:

- Ejecución del programa de pieza: nivel de programa 12
- Rutina de interrupción: nivel de programa 14

1.25.1.4 Ruta de búsqueda

Al llamar a un subprograma sin indicación de ruta, el control realiza la búsqueda en los siguientes directorios en el orden indicado:

Orden	Directorio	Descripción
1.	Directorio actual	Directorio del programa desde el que se realiza la llamada
2.	/_N_SPF_DIR /	Directorio global de subprogramas
3.	/_N_CUS_DIR /	Ciclos de usuario
4.	/_N_CMA_DIR /	Ciclos del fabricante
5.	/_N_CST_DIR /	Ciclos estándar

1.25.1.5 Parámetros formales y actuales

Se habla de parámetros formales y actuales en relación con la definición y la llamada de subprogramas con transferencia de parámetros.

Parámetro formal

En la definición de un subprograma, los parámetros que deben transferirse al subprograma, los llamados parámetros formales, deben definirse con tipo y nombre de parámetro.

Por consiguiente, los parámetros formales definen la interfaz del subprograma.

Ejemplo:

```
Código de programa                    Comentarios
PROC CONTORNO (REAL X, REAL Y)        ; Parámetro formal: X e Y, ambos del tipo REAL
N20 X1=X Y1=Y                         ; Desplazamiento del eje X1 a la posición X y
                                        del eje Y1 a la posición Y
...
N100 RET
```

Parámetro actual

Al llamar a un subprograma, deben transferirse al subprograma valores o variables absolutos, los llamados parámetros actuales.

Así, al efectuar la llamada los parámetros actuales rellenan la interfaz del subprograma con valores actuales.

Ejemplo:

```
Código de programa                    Comentarios
N10 DEF REAL ANCHO                    ; Definición de variables
N20 ANCHO=20.0                        ; Asignación de variables
N30 CONTORNO(5.5, ANCHO)              ; Llamada de subprograma con parámetros
                                        actuales: 5.5 y ANCHO
...
N100 M30
```

1.25.1.6 Transferencia de parámetros

Definición de un subprograma con transferencia de parámetros

Un subprograma con transferencia de parámetros se define con la palabra reservada `PROC` y una lista completa de todos los parámetros que espera el subprograma.

Transferencia incompleta de parámetros

Al llamar el subprograma no siempre es necesario transferir explícitamente todos los parámetros definidos en la interfaz del subprograma. Si se omite un parámetro, para este parámetro se transferirá el valor estándar "0".

No obstante, para la identificación unívoca del orden de los parámetros deben indicarse siempre las comas como caracteres separadores de los parámetros. El último parámetro es una excepción. Si éste se omite al efectuar la llamada, también puede omitirse la última coma.

Ejemplo:

Subprograma:

```
Código de programa                          Comentarios
PROC SUB_PROG (REAL X, REAL Y, REAL Z)      ;  Parámetros formales: X, Y y Z
...
N100 RET
```

Programa principal:

```
Código de programa              Comentarios
PROC MAIN_PROG
...
N30 SUB_PROG(1.0,2.0,3.0)       ; Llamada de subprograma con transferencia de
                                  parámetros completa:
                                  X=1.0, Y=2.0, Z=3.0
…
N100 M30
```

Ejemplos de la llamada de subprograma en `N30` con una transferencia de parámetros incompleta:

```
N30 SUB_PROG( ,2.0,3.0)          ; X=0.0, Y=2.0, Z=3.0
N30 SUB_PROG(1.0, ,3.0)          ; X=1.0, Y=0.0, Z=3.0
N30 SUB_PROG(1.0,2.0)            ; X=1.0, Y=2.0, Z=0.0
N30 SUB_PROG( , ,3.0)            ; X=0.0, Y=0.0, Z=3.0
N30 SUB_PROG( , , )              ; X=0.0, Y=0.0, Z=0.0
```

PRECAUCIÓN
Transferencia de parámetros Call-By-Reference Los parámetros transferidos mediante Call-By-Reference no deben omitirse al llamar al subprograma.

PRECAUCIÓN
Tipo de datos AXIS Los parámetros del tipo de datos AXIS no deben omitirse al llamar al subprograma.

Comprobación de los parámetros de transferencia

Mediante la variable de sistema $P_SUBPAR [n] con n = 1, 2, ... en el subprograma puede comprobarse si un parámetro se ha transferido explícitamente o si se ha omitido. El índice n se refiere al orden de los parámetros formales. El índice n = 1 se refiere al 1er parámetro formal; el índice n = 2, al 2º parámetro formal, etc.

El siguiente segmento de programa muestra, a modo de ejemplo para el 1er parámetro formal, la manera en que puede realizarse una comprobación:

```
Programación                                Comentarios
PROC SUB_PROG (REAL X, REAL Y, REAL Z)      ; Parámetros formales: X, Y y Z
N20 IF $P_SUBPAR[1]==TRUE                   ; Comprobación del 1er parámetro formal
                                              X.
...                                         ; Estas acciones se ejecutan cuando el
                                              parámetro formal X se ha transferido
                                              explícitamente.
N40 ELSE
...                                         ; Estas acciones se ejecutan cuando el
                                              parámetro formal X no se ha
                                              transferido.
N60 ENDIF
...                                         ; Acciones generales
N100 RET
```

1.25.2 Definición de un subprograma

1.25.2.1 Subprograma sin transferencia de parámetros

Función

Al definir subprogramas sin transferencia de parámetros, se puede omitir la línea de definición al principio del programa.

Sintaxis

```
[PROC <Nombre de programa>]
...
```

Descripción

PROC:	Instrucción de definición al principio de un programa
<Nombre de programa>:	Nombre del programa

Ejemplo

Ejemplo 1: Subprograma con instrucción PROC

Código de programa	**Comentarios**
PROC SUB_PROG	; Línea de definición
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; Salto atrás al subprograma

Ejemplo 2: Subprograma sin instrucción PROC

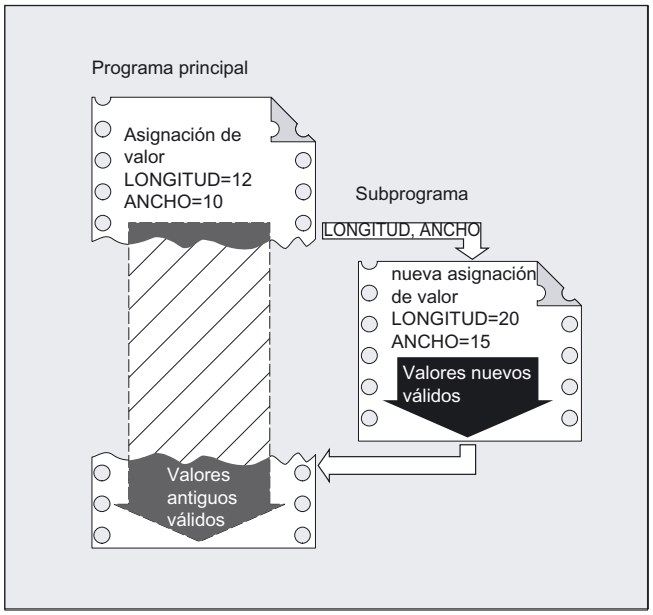
Código de programa	**Comentarios**
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; Salto atrás al subprograma

1.25.2.2 Subprograma con transferencia de parámetros Call-By-Value (PROC)

Función

Un subprograma con transferencia de parámetros Call-By-Value se define con la palabra reservada `PROC`, seguida del nombre del programa y una lista completa de todos los parámetros que espera el subprograma con tipo y nombre. La instrucción de definición debe estar en la primera línea de programa.

La transferencia de parámetros Call-By-Value no tiene repercusiones sobre el programa desde el que se realiza la llamada. Dicho programa transfiere al subprograma sólo los valores de los parámetros actuales.



Nota

Puede transferirse un máximo de 127 parámetros.

Sintaxis

```
PROC <Nombre de programa> (<Tipo de parámetro> <Nombre de
parámetro>, ...)
```

Descripción

`PROC`:	Instrucción de definición al principio de un programa
`<Nombre de programa>`:	Nombre del programa
`<Tipo de parámetro>`:	Tipo de datos del parámetro (p. ej. REAL, INT, BOOL)
`<Nombre del parámetro>`:	Nombre del parámetro

ATENCIÓN
El nombre de programa indicado después de la palabra reservada PROC debe coincidir con el nombre de programa asignado a la interfaz de usuario.

Ejemplo

Definición de un subprograma con 2 parámetros del tipo REAL:

Código de programa	**Comentarios**
`PROC SUB_PROG (REAL LONGITUD, REAL ANCHO)`	`; Parámetro 1: Tipo: REAL, nombre: LONGITUD` `Parámetro 2: Tipo: REAL, nombre: ANCHO`
`...`	
`N100 RET`	`; Salto atrás al subprograma`

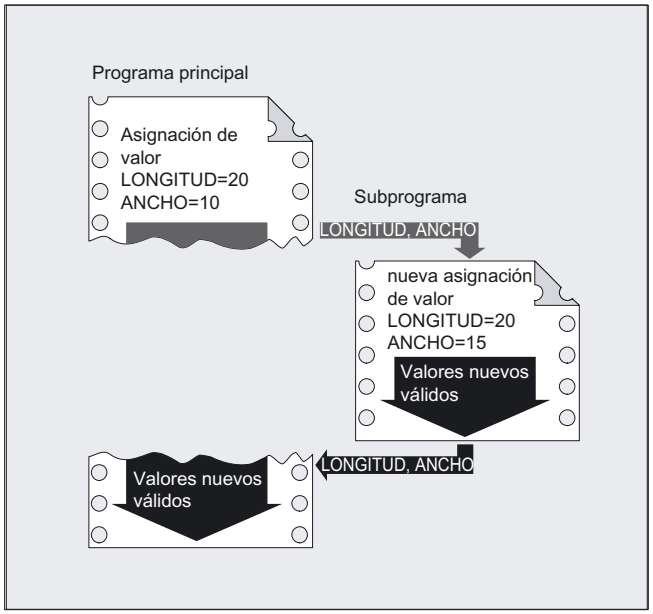
1.25.2.3 Subprograma con transferencia de parámetros Call-By-Reference (PROC, VAR)

Función

Un subprograma con transferencia de parámetros Call-By-Reference se define con la palabra reservada `PROC`, seguida del nombre del programa y una lista completa de todos los parámetros que espera el subprograma con la palabra reservada `VAR`, tipo y nombre. La instrucción de definición debe estar en la primera línea de programa.

En la transferencia de parámetros Call-By-Reference también pueden transferirse referencias a matrices.

La transferencia de parámetros Call-By-Reference no tiene repercusiones sobre el programa desde el que se realiza la llamada. Dicho programa transfiere al subprograma una referencia al parámetro actual y permite así que el subprograma acceda directamente a la correspondiente variable.



Nota

Puede transferirse un máximo de 127 parámetros.

Nota

Una transferencia de parámetros Call-By-Reference sólo será necesaria si la variable transferida se ha definido en el programa desde el que se realiza la llamada (LUD). No es necesario transferir variables globales de canal y globales de CN, ya que a estas variables también se puede acceder directamente desde el subprograma.

Sintaxis

```
PROC <Nombre de programa> (VAR <Tipo de parámetro> <Nombre de
parámetro>, ...)
PROC <Nombre de programa> (VAR <Tipo de matriz> <Nombre de matriz>
[<m>,<n>,<o>], ...)
```

Descripción

PROC:	Instrucción de definición al principio de un programa
VAR:	Palabra reservada para la transferencia de parámetros por referencia
<Nombre de programa>:	Nombre del programa
<Tipo de parámetro>:	Tipo de datos del parámetro (p. ej. REAL, INT, BOOL)
<Nombre del parámetro>:	Nombre del parámetro
<Tipo de matriz>:	Tipo de datos de los elementos de matriz (p. ej. REAL, INT, BOOL)
<Nombre de matriz>:	Nombre de la matriz
[<m>,<n>,<o>]:	Tamaño de matriz Actualmente se permiten matrices de 3 dimensiones como máximo: <m>: Tamaño de la matriz para 1ª dimensión <n>: Tamaño de la matriz para 2ª dimensión <o>: Tamaño de la matriz para 3ª dimensión

ATENCIÓN

El nombre de programa indicado después de la palabra reservada PROC debe coincidir con el nombre de programa asignado a la interfaz de usuario.

Nota

Con matrices de longitud indefinida como parámetros formales pueden ejecutarse matrices de subprogramas de longitud variable. Para ello no se indica la longitud de la 1ª dimensión como parámetro formal en la definición, p. ej., de una matriz bidimensional. Sin embargo, es preciso escribir la coma.

Ejemplo: PROC <Nombre de programa> (VAR REAL MATRIZ[,5])

Ejemplo

Definición de un subprograma con 2 parámetros como referencia al tipo REAL:

Código de programa	**Comentarios**
`PROC SUB_PROG(VAR REAL LONGITUD, VAR REAL ANCHO)`	`; Parámetro 1: Referencia al tipo: REAL, nombre: LONGITUD` `Parámetro 2: Referencia al tipo: REAL, nombre: ANCHO`
`...`	
`N100 RET`	

1.25.2.4 Almacenamiento de las funciones G modales (SAVE)

Función

El atributo `SAVE` hace que se guarden las funciones G modales activas antes de llamar al subprograma y se reactiven de nuevo antes de finalizar el subprograma.

PRECAUCIÓN

Modo de contorneado ininterrumpido

`SAVE` no debe utilizarse cuando está activo un modo de contorneado que no debería ser interrumpido por la llamada de un subprograma.

Sintaxis

```
PROC <Nombre del subprograma> SAVE
```

Descripción

`SAVE`: Almacenamiento de las funciones G modales antes de llamar al subprograma y restablecimiento tras finalizar el subprograma

Ejemplo

En el subprograma CONTORNO actúa la función G modal G91 (acotado incremental). En el programa principal actúa la función G modal G90 (acotado absoluto). Mediante la definición de subprograma con SAVE actúa de nuevo G90 tras finalizar el subprograma en el programa principal.

Definición de subprograma:

Código del programa	**Comentarios**
PROC CONTORNO (VALOR REAL1) SAVE	; Definición de subprograma con parámetro SAVE
N10 G91 …	; Función G modal G91: Acotado incremental
N100 M17	; Fin de subprograma

Programa principal:

Código del programa	**Comentarios**
N10 G0 X… Y… G90	; Función G modal G90: Acotado absoluto
N20 …	
...	
N50 CONTORNO (12.4)	; Llamada a un subprograma
N60 X… Y…	; Función G modal G90 reactivada mediante SAVE

Limitaciones

Frames

El comportamiento de frames en relación con subprogramas con el atributo SAVE depende del tipo de frame y puede ajustarse mediante los datos de máquina.

Bibliografía

Manual de funciones, Funciones básicas; Ejes, sistemas de coordenadas, frames (K2), capítulo: "Retorno a subprograma con SAVE"

1.25.2.5 Suprimir la ejecución de secuencia individual (SBLOF, SBLON)

Función

Supresión de secuencia individual para todo el programa

Los programas identificados con SBLOF se ejecutan por completo como una secuencia en el caso de ejecución de secuencia única, es decir, la ejecución de secuencia única se suprime para todo el programa.

SBLOF se encuentra en la línea PROC y es válido hasta el final o la interrupción de un subprograma. Con el comando de salto atrás se decide si se parará al final de un subprograma o no:

Salto atrás con M17:	Parada al final del subprograma
Salto atrás con RET:	Sin parada al final del subprograma

Supresión de secuencia individual dentro del programa

SBLOF debe encontrarse solo en la secuencia. La secuencia individual se desactiva desde esta secuencia hasta:

- el siguiente SBLON

  o

- el final del nivel de subprograma activo.

Sintaxis

Supresión de secuencia individual para todo el programa:
```
PROC ... SBLOF
```

Supresión de secuencia individual dentro del programa:

```
SBLOF
...
SBLON
```

Descripción

PROC:	Primera instrucción de un programa
SBLOF:	Comando para desactivar la ejecución de secuencia individual SBLOF puede encontrarse en una secuencia PROC o estar solo en la secuencia.
SBLON:	Comando para activar la ejecución de secuencia individual SBLON debe encontrarse en una secuencia propia.

Limitaciones

- **Supresión de secuencia individual y visualización de secuencia**

  La visualización de secuencia actual puede suprimirse en ciclos/subprogramas con DISPLOF. Si se programa DISPLOF junto con SBLOF, al efectuar paradas de secuencia individual dentro del ciclo/subprograma, se mostrará la llamada del ciclo/subprograma como antes.

- **Supresión de secuencia individual en ASUP de sistema o ASUP de usuario**

  Si la parada de secuencia individual se suprime en el ASUP de sistema o usuario mediante los ajustes del dato de máquina DM10702 $MN_IGNORE_SINGLEBLOCK_MASK (Bit0 = 1 o Bit1 = 1), puede activarse de nuevo programando SBLON en el ASUP.

  Si la parada de secuencia individual se suprime en el ASUP de usuario mediante el ajuste del dato de máquina DM20117 $MC_IGNORE_SINGLEBLOCK_ASUP, **no** puede activarse de nuevo programando SBLON en el ASUP.

- **Particularidades de la supresión de secuencia individual en diferentes tipos de ejecución de secuencia individual**

  En caso de ejecución de secuencia individual activa SBL2 (parada después de cada secuencia del programa de pieza) **no** se efectúa la parada en la secuencia SBLON si el bit 12 está fijado en "1" en DM10702 $MN_IGNORE_SINGLEBLOCK_MASK (evitar parada de secuencia individual).

  En caso de ejecución de secuencia individual activa SBL3 (parada después de cada secuencia del programa de pieza también en el ciclo), se suprime el comando SBLOF.

Ejemplos

Ejemplo 1: Supresión de secuencia individual dentro de un programa

```
Código del programa          Comentarios
N10 G1 X100 F1000
N20 SBLOF                    ; Desactivar Secuencia a secuencia
N30 Y20
N40 M100
N50 R10=90
N60 SBLON                    ; Volver a activar Secuencia a secuencia
N70 M110
N80 ...
```

El área entre N20 y N60 se ejecuta en el modo secuencia a secuencia como un solo paso.

Ejemplo 2: El ciclo tiene que actuar para el usuario como un comando

Programa principal:

```
Código del programa
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30
```

Ciclo CYCLE1:

```
Código del programa                          Comentarios
N100 PROC CYCLE1 DISPLOF SBLOF               ; Supresión de secuencia individual
N110 R10=3*SIN(R20)+5
N120 IF (R11 <= 0)
N130 SETAL(61000)
N140 ENDIF
N150 G1 G91 Z=R10 F=R11
N160 M17
```

El ciclo CYCLE1 se ejecuta con ejecución de secuencia individual activa, es decir, para la ejecución del ciclo CYCLE1 debe pulsarse una vez la tecla de arranque.

Ejemplo 3: Un ASUP iniciado por el PLC para activar el decalaje de origen y correcciones de herramienta modificados no debe ser visible.

```
Código del programa
N100 PROC NV SBLOF DISPLOF
N110 CASE $P_UIFRNUM OF     0 GOTOF _G500
                            1 GOTOF _G54
                            2 GOTOF _G55
                            3 GOTOF _G56
                            4 GOTOF _G57
                            DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET
```

Ejemplo 4: No se efectúa la parada con DM10702 bit 12 = 1

Situación inicial:

- La ejecución de secuencia individual está activa.
- DM10702 $MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

Programa principal:

Código del programa	Comentarios
N10 G0 X0	; Parada en esta línea del programa de pieza.
N20 X10	; Parada en esta línea del programa de pieza.
N30 CYCLE	; Secuencia de desplazamiento generada por el ciclo.
N50 G90 X20	; Parada en esta línea del programa de pieza.
M30	

Ciclo CYCLE:

Código del programa	Comentarios
PROC CYCLE SBLOF	; Suprimir parada de secuencia individual
N100 R0 = 1	
N110 SBLON	; No se efectúa la parada en esta línea del programa de pieza debido a DM10702 bit 12=1.
N120 X1	; La parada se efectúa en esta línea del programa de pieza.
N140 SBLOF	
N150 R0 = 2	
RET	

Ejemplo 5: Supresión de secuencia individual con imbricación del programa

Situación inicial:

La ejecución de secuencia individual está activa.

Imbricación del programa:

```
Código del programa                          Comentarios
N10 X0 F1000                                 ; La parada se efectúa en esta secuencia.
N20 UP1(0)
      PROC UP1(INT _NR) SBLOF                ; Suprimir parada de secuencia
                                               individual.
      N100 X10
      N110 UP2(0)
            PROC UP2(INT _NR)
            N200 X20
            N210 SBLON                       ; Activar parada de secuencia individual.
            N220 X22                         ; La parada se efectúa en esta secuencia.
            N230 UP3(0)
                  PROC UP3(INT _NR)
                  N300 SBLOF                 ; Suprimir parada de secuencia
                                               individual.
                  N305 X30
                  N310 SBLON                 ; Activar parada de secuencia individual.
                  N320 X32                   ; La parada se efectúa en esta secuencia.
                  N330 SBLOF                 ; Suprimir parada de secuencia
                                               individual.
                  N340 X34
                  N350 M17                   ; SBLOF está activo.
            N240 X24                         ; La parada se efectúa en esta secuencia.
                                               SBLON está activo.
            N250 M17                         ; La parada se efectúa en esta secuencia.
                                               SBLON está activo.
      N120 X12
      N130 M17                               ; La parada se efectúa en esta secuencia
                                               de salto atrás. SBLOF de la instrucción
                                               PROC está activo.
N30 X0                                       ; La parada se efectúa en esta secuencia.
N40 M30                                      ; La parada se efectúa en esta secuencia.
```

Información adicional

Bloqueo de Secuencia a secuencia para subprogramas asíncronos

Para ejecutar un ASUP en un paso en una secuencia individual debe programarse en el ASUP una instrucción PROC con SBLOF. Lo mismo se aplica para la función "ASUP editable de sistema" (DM11610 $MN_ASUP_EDITABLE).

Ejemplo de una ASUP editable de sistema:

```
Código del programa                    Comentarios
N10 PROC ASUP1 SBLOF DISPLOF
N20 IF $AC_ASUP=='H200'
N30 RET                                ; Sin REPOS con cambio de modo.
N40 ELSE
N50 REPOSA                             ; REPOS en todos los demás casos.
N60 ENDIF
```

Influencia del programa en Secuencia a secuencia

En la ejecución de secuencia individual, el usuario puede ejecutar el programa de pieza secuencia a secuencia. Se dispone de los siguientes tipos de ajuste:

- SBL1: Secuencia a secuencia IPO con parada después de cada secuencia de función de máquina.
- SBL2: Secuencia a secuencia con parada después de cada secuencia.
- SBL3: Parada en el ciclo (con la selección de SBL3 se suprime el comando SBLOF).

Supresión de secuencia individual con imbricación del programa

Si en un subprograma se programa SBLOF en la instrucción PROC, se parará en el salto atrás del subprograma con M17. De este modo se evita que ya se ejecute la siguiente secuencia en el programa desde el cual se efectúa la llamada. Si en un subprograma se activa una supresión de secuencia individual conSBLOF, sin SBLOF en la instrucción PROC, se parará sólo después de la siguiente secuencia de función de máquina del programa desde el que se efectúa la llamada. Si esto no se desea, es necesario programar de nuevo SBLON en el subprograma antes del salto atrás (M17). En un salto atrás con RET no se parará en un programa de nivel superior.

1.25.2.6 Suprimir la visualización de secuencia actual (DISPLOF, DISPLON, ACTBLOCNO)

Función

La secuencia de programa actual se muestra de manera estándar en la visualización de secuencia. En ciclos o subprogramas puede suprimirse la visualización de la secuencia actual como el comando DISPLOF. En vez de la secuencia actual se visualizará en ese caso la llamada del ciclo o del subprograma. Con el comando DISPLON puede volverse a eliminar la supresión de la visualización de secuencia.

DISPLOF o DISPLON se programa en la línea de programa con la instrucción PROC y actúa para el subprograma entero e implícitamente para todos los subprogramas a los que llama este subprograma y que no contienen el comando DISPLON o DISPLOF. Este comportamiento también se da para los ASUP.

Sintaxis

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

Descripción

DISPLOF: Comando para suprimir la visualización de secuencia actual.

Colocación: Al final de la línea del programa con la instrucción PROC

Eficacia: hasta el salto atrás desde el subprograma o el final del programa.

Nota:
Si desde el subprograma se llama a otros subprogramas con el comando DISPLOF, también se suprimirá en ellos la visualización de secuencia actual siempre que ahí no esté explícitamente programado DISPLON.

DISPLON: Comando para anular la supresión de la visualización de secuencia actual

Colocación: Al final de la línea del programa con la instrucción PROC

Eficacia: hasta el salto atrás desde el subprograma o el final del programa.

Nota:
Si desde el subprograma se llama a otros subprogramas con el comando DISPLON, también se visualizará en ellos la secuencia actual de programa siempre que ahí no esté explícitamente programado DISPLOF.

ACTBLOCNO: DISPLOF junto con el atributo ACTBLOCNO hace que, en caso de alarma, se emita el número de la secuencia actual en la que se ha producido la alarma. Esto también se aplica cuando en un nivel de programa inferior sólo está programado DISPLOF.

Con DISPLOF sin ACTBLOCNO, por el contrario, se muestra el número de secuencia de la llamada a ciclo o subprograma del último nivel de programa no identificado con DISPLOF.

Ejemplos

Ejemplo 1: Suprimir la visualización de secuencia actual en el ciclo

```
Código del programa                                          Comentarios
PROC CYCLE (AXIS TOMOV, REAL POSITION) SAVE DISPLOF          ; Suprimir la visualización de
                                                               secuencia actual. En su lugar, se
                                                               debe mostrar la llamada de ciclo,
                                                               p. ej.: CYCLE(X,100.0)
DEF REAL DIFF                                                ; Contenido del ciclo
G01 ...
...
RET                                                          ; Salto atrás a subprograma. En la
                                                               visualización de secuencia se
                                                               muestra la secuencia que sigue a la
                                                               llamada de ciclo.
```

Ejemplo 2: Visualización de secuencia con emisión de alarma

Subprograma SUBPROG1 (con ACTBLOCNO):

```
Código del programa                    Comentarios
PROC SUBPROG1 DISPLOF ACTBLOCNO
N8000 R10 = R33 + R44
...
N9040 R10 = 66 X100                    ; Disparar alarma 12080
...
N10000 M17
```

Subprograma SUBPROG2 (sin ACTBLOCNO):

```
Código del programa          Comentarios
PROC SUBPROG2 DISPLOF
N5000 R10 = R33 + R44
...
N6040 R10 = 66 X100          ; Disparar alarma 12080
...
N7000 M17
```

Programa principal:

```
Código del programa          Comentarios
N1000 G0 X0 Y0 Z0
N1010 ...
...
N2050 SUBPROG1               ; Emisión de alarma = "12080 canal K1 secuencia N9040
                               error de sintaxis en texto R10="
N2060 ...
N2350 SUBPROG2               ; Emisión de alarma = "12080 canal K1 secuencia N2350
                               error de sintaxis en texto R10="
...
N3000 M30
```

Ejemplo 3: Anular la supresión de la visualización de secuencia actual

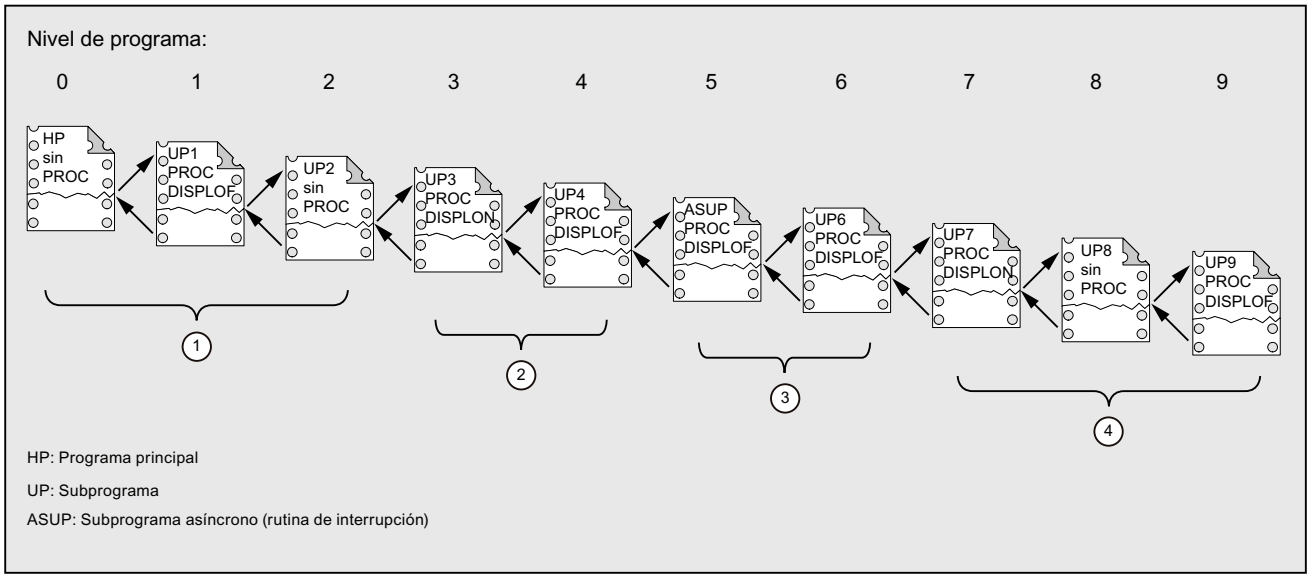
Subprograma SUB1 con supresión:

```
Código de programa           Comentarios
PROC SUB1 DISPLOF            ; Suprimir la visualización de secuencia actual en el
                               subprograma SUB1. En lugar de ello, la secuencia
                               debe visualizarse con la llamada SUB1.
...
N300 SUB2                    ; Llamar a subprograma SUB2.
...
N500 M17
```

Subprograma SUB2 sin supresión:

```
Código de programa           Comentarios
PROC SUB2 DISPLON            ; Anular la supresión de la visualización de
                               secuencia actual en el subprograma SUB2.
...
N200 M17                     ; Salto atrás al subprograma SUB1. En SUB1 se vuelve
                               a suprimir la visualización de secuencia actual.
```

Ejemplo 4: Comportamiento de la visualización con diferentes combinaciones DISPLON/DISPLOF



① En la visualización de secuencia actual, se muestran las líneas de programa de pieza desde el nivel de programa 0.
② En la visualización de secuencia actual, se muestran las líneas de programa de pieza desde el nivel de programa 3.
③ En la visualización de secuencia actual, se muestran las líneas de programa de pieza desde el nivel de programa 3.
④ En la visualización de secuencia actual, se muestran las líneas de programa de pieza desde el nivel de programa 7/8.

1.25.2.7 Identificación de subprogramas con preparación (PREPRO)

Función

Con la palabra reservada `PREPRO` se pueden identificar todos los ficheros en el arranque al final de la línea de instrucciones `PROC`.

Nota

Este tipo de preparación de programa depende del correspondiente dato de máquina ajustado. Tener en cuenta las indicaciones del fabricante de la máquina.

Bibliografía:
Manual de funciones especiales, Preprocesamiento (V2)

Sintaxis

```
PROC … PREPRO
```

Descripción

`PREPRO`: Palabra reservada para identificar todos los ficheros preparados en el arranque de los programas CN que se encuentran en directorios de ciclos

Lectura de subprogramas con preparación y llamada de subprogramas

Tanto en el arranque de subprogramas preparados con parámetros como en la llamada de subprogramas, los directorios de ciclos se tratan con el mismo orden:

1. _N_CUS_DIR ciclos de usuario
2. _N_CMA_DIR ciclos de fabricante
3. _N_CST_DIR ciclos estándar

En el caso de programas CN con el mismo nombre y distinta versión, se activa la instrucción `PROC` encontrada en primer lugar y se pasa por alto la otra instrucción `PROC` sin producirse un aviso de alarma.

1.25.2.8 Salto atrás al subprograma M17

Función

Al final de un subprograma se encuentra el comando de salto atrás `M17` (o bien el comando de fin del programa de pieza `M30`). Provoca el salto atrás al programa desde el que se realiza la llamada en la secuencia del programa de pieza después de la llamada del subprograma.

Nota

`M17` y `M30` se tratan de una forma equivalente en el lenguaje CN.

Sintaxis

```
PROC <Nombre de programa>
...
M17/M30
```

Limitaciones

Efecto del salto atrás al subprograma en el modo de contorneado

Si `M17` (o `M30`) se encuentra solo en la secuencia del programa de pieza, con él se interrumpe un modo de contorneado activo en el canal.

Para impedir que el modo de contorneado se interrumpa, `M17` (o `M30`) debe escribirse también en la última secuencia de desplazamiento. Adicionalmente debe estar ajustado a "0" el siguiente dato de máquina:

DM20800 $MC_SPF_END_TO_VDI = 0 (ninguna salida M30/M17 a la interfaz CN/PLC)

Ejemplo

1. Subprograma con M17 en la secuencia propia

```
Código de programa                Comentarios
N10 G64 F2000 G91 X10 Y10
N20 X10 Z10
N30 M17                           ; Salto atrás con interrupción del modo de
                                    contorneado.
```

2. Subprograma con M17 en la última secuencia de desplazamiento

```
Código de programa                Comentarios
N10 G64 F2000 G91 X10 Y10
N20 X10 Z10 M17                   ; Salto atrás sin interrupción del modo de
                                    contorneado.
```

1.25.2.9 Salto atrás al subprograma RET

Función

En lugar del comando de salto atrás M17, en el subprograma se puede utilizar también el comando RET. RET se debe programar en una secuencia de programa de pieza propia. Al igual que M17, RET provoca el salto atrás al programa desde el que se realiza la llamada en la secuencia del programa de pieza después de la llamada del subprograma.

Nota

Mediante la programación de parámetros se puede modificar el comportamiento del salto atrás de RET (ver "Salto atrás parametrizable al subprograma (RET ...) (Página 178)").

Aplicación

La instrucción RET debe utilizarse cuando un modo de contorneado G64 (G641 ... G645) no deba interrumpirse por el retorno.

Requisito

Un requisito para la utilización de RET es que el subprograma no posea ningún atributo SAVE.

Sintaxis

```
PROC <Nombre de programa>
...
RET
```

Ejemplo

Programa principal:

Código de programa	Comentarios
PROC MAIN_PROGRAM	; Inicio del programa
...	
N50 SUB_PROG	; Llamada a un subprograma: SUB_PROG
N60 ...	
...	
N100 M30	; Fin del programa

Subprograma:

Código de programa	Comentarios
PROC SUB_PROG	
...	
N100 RET	; El salto atrás se produce en la secuencia N60 en el programa principal.

1.25.2.10 Salto atrás parametrizable al subprograma (RET ...)

Función

Por lo general, desde un subprograma con un fin de subprograma `RET` o `M17`, se regresa al programa desde el que se ha llamado al subprograma y se continúa la ejecución con la línea de programa que sigue a la de la llamada al subprograma.

Además, hay también casos de aplicación en los que la ejecución del programa debe continuar en otro punto, p. ej.:

- Continuación de la ejecución del programa tras llamar a los ciclos de desbaste en modo de dialecto ISO (después de la descripción de contorno).
- Salto atrás al programa principal desde cualquier nivel de subprograma (también después de ASUP) en el tratamiento de errores.
- Salto atrás de varios niveles de programa para aplicaciones especiales en ciclos de compilación y en el modo de dialecto ISO.

En estos casos, el comando `RET` se programa junto con los "parámetros de salto atrás".

Sintaxis

```
RET("<Secuencia meta>")
RET("<Secuencia meta>", <Secuencia tras secuencia meta>)
RET("<Secuencia meta>"),
<Secuencia tras secuencia meta>,<Cantidad niveles salto atrás>)
RET("<Secuencia meta>",<Cantidad niveles salto atrás>)
RET("<Secuencia meta>"),
<Secuencia tras secuencia meta>,<Cantidad niveles salto atrás>,
<Salto atrás hasta inicio programa>)
RET( ,
,<Cantidad niveles salto atrás>,<Salto atrás hasta inicio programa>)
```

Descripción

RET:	Fin del subprograma (utilización en lugar de M17)
<Secuencia meta>:	Parámetro de salto atrás 1
	Indica la secuencia en la que debe continuar la ejecución del programa como destino del salto.
	Si está programado el parámetro de salto atrás 3, el destino del salto se encuentra en el programa desde el que se ha llamado al subprograma actual.
	Posibles indicaciones:
	"<Número de secuencia>" — Número de la secuencia meta
	"<Marca de salto>" — Marca de salto que debe fijarse en la secuencia meta.
	"<Cadena de caracteres>" — Cadena de caracteres que debe estar declarada en el programa (p. ej., nombre de programa o de variable).
	Para la programación de la cadena de caracteres en la secuencia meta, son válidas las siguientes reglas:
	• **Espacio al final** (a diferencia de la marca de salto, que se identifica por un ":" al final).
	• **Antes de la cadena de caracteres** sólo deben fijarse un número de secuencia y/o una marca de salto, pero **ningún comando de programa**.

`<Secuencia tras secuencia meta>`:

Parámetro de salto atrás 2

Se refiere al parámetro de salto atrás 1.

Tipo: INT

Valor:
- 0: El salto atrás se efectúa en la secuencia indicada con el parámetro de salto atrás 1.
- > 0: El salto atrás se efectúa en la secuencia que sigue a la indicada con el parámetro de salto atrás 1.

`<Cantidad niveles salto atrás>`:

Parámetro de salto atrás 3

Indica la cantidad de niveles que deben saltarse hacia atrás para llegar al nivel de programa en el que debe continuar la ejecución del programa.

Tipo: INT

Valor:
- 1: El programa continúa en el "nivel de programa actual - 1" (como `RET` sin parámetros).
- 2: El programa continúa en el "nivel de programa actual - 2", es decir, se salta un nivel hacia atrás.
- 3: El programa continúa en el "nivel de programa actual - 3", es decir, se saltan dos niveles hacia atrás.
- ...

Margen de valores: 1 ... 15

`<Salto atrás hasta inicio programa>`

Parámetro de salto atrás 4

Tipo: BOOL

Valor:
- 1: Si el salto atrás tiene lugar en el programa principal y allí hay activo un **modo de dialecto ISO**, se deriva al inicio del programa.

Nota

En el caso de salto atrás al subprograma con una cadena de caracteres como indicación para la búsqueda de secuencia meta, se busca siempre primero una marca de salto en el programa desde el que se llama.

Si debe definirse unívocamente un destino de salto con una cadena de caracteres, ésta no debe coincidir con el nombre de una marca de salto ya que, de lo contrario, el salto atrás al subprograma se ejecutará siempre en la marca de salto y no en la cadena de caracteres (ver ejemplo 2).

Limitaciones

Al saltar atrás varios niveles de programa se evalúan las instrucciones SAVE de los distintos niveles de programa.

Si en un salto atrás de varios niveles de programa hay activo un subprograma modal y se ha programado en uno de los subprogramas saltados el comando de cancelación MCALL para el subprograma modal, éste seguirá activo.

> **⚠ PRECAUCIÓN**
>
> El programador debe asegurarse de que, al saltar hacia atrás varios niveles de programa, se continúe con los ajustes modales correctos. Para ello debe, p. ej., programarse una secuencia principal adecuada.

Ejemplos

Ejemplo 1: Reposicionamiento en el programa principal tras un procesamiento ASUP

```
Programación                                  Comentarios
N10010 CALL "UP1"                             ; Nivel de programa 0 (programa
                                                principal)
  N11000 PROC UP1                             ; Nivel de programa 1
  N11010 CALL "UP2"
    N12000 PROC UP2                           ; Nivel de programa 2
    ...
      N19000 PROC ASUP                        ; Nivel de programa 3 (procesamiento
                                                ASUP)
      ...
      N19100 RET("N10900", ,$P_STACK)         ; Salto atrás al subprograma
N10900                                        ; Reposicionamiento en el programa
                                                principal.
N10910 MCALL                                  ; Desactivar subprograma modal.
N10920 G0 G60 G40 M5                          ; Corregir otros ajustes modales.
```

Ejemplo 2: Cadena de caracteres (<String>) como indicación para la búsqueda de secuencia meta

Programa principal:

```
Código de programa                    Comentarios
PROC MAIN_PROGRAM
N1000 DEF INT iVar1=1, iVar2=4
N1010 ...
N1200 subProg1                        ; Llamada del subprograma "subProg1"
N1210 M2 S1000 X10 F1000
N1220 ......
N1400 subProg2                        ; Llamada del subprograma "subProg2"
N1410 M3 S500 Y20
N1420 ..
N1500 lab1: iVar1=R10*44
N1510 F500 X5
N1520 ...
N1550 subprog1: G1 X30                ; "subProg1" está definido aquí como marca de
                                        salto.
N1560 ...
N1600 subProg3                          Llamada del subprograma "subProg3"
N1610 ...
N1900 M30
```

Subprograma subProg1:

```
Código de programa            Comentarios
PROC subProg1
N2000 R10=R20+100
N2010 ...
N2200 RET("subProg2")         ; Salto atrás al programa principal a la secuencia
                                N1400
```

Subprograma subProg2:

```
Código de programa            Comentarios
PROC subProg2
N2000 R10=R20+100
N2010 ...
N2200 RET("iVar1")            ; Salto atrás al programa principal a la secuencia
                                N1500
```

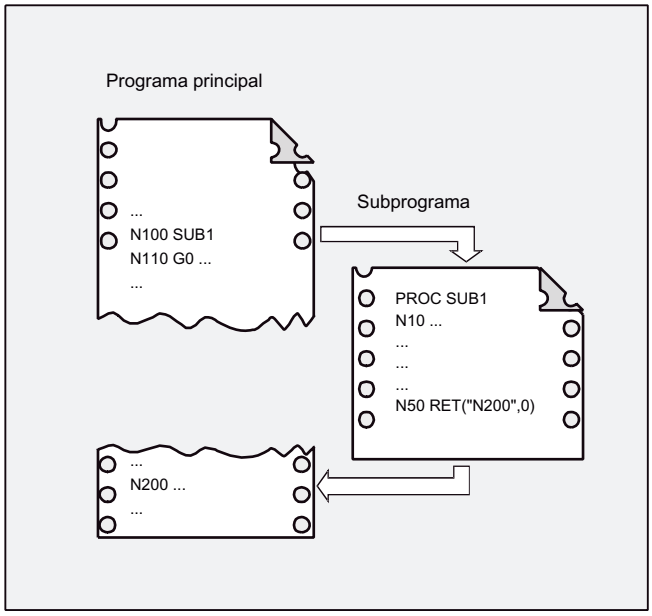
Subprograma subProg3:

Código de programa	**Comentarios**
`PROC subProg3`	
`N2000 R10=R20+100`	
`N2010 ...`	
`N2200 RET("subProg1")`	`; Salto atrás al programa principal a la secuencia N1550`

Información adicional

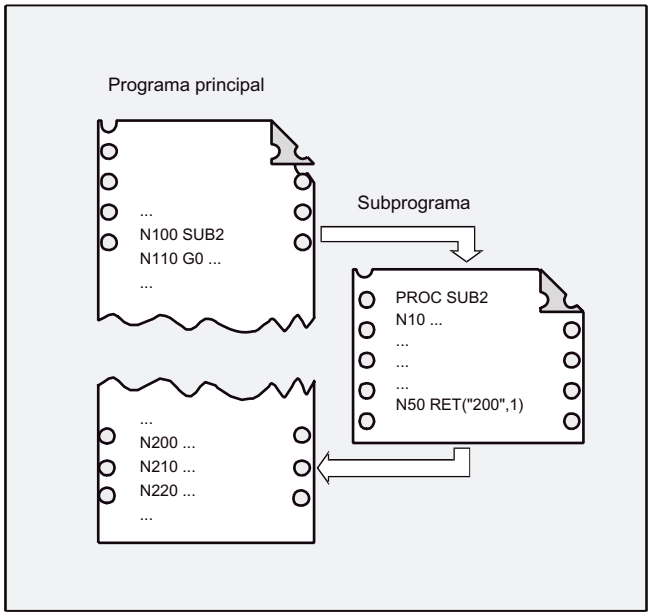
Los siguientes gráficos muestran los distintos efectos de los parámetros de salto atrás 1 a 3.

1er parámetro de salto atrás 1 = "N200", parámetro de salto atrás 2 = 0



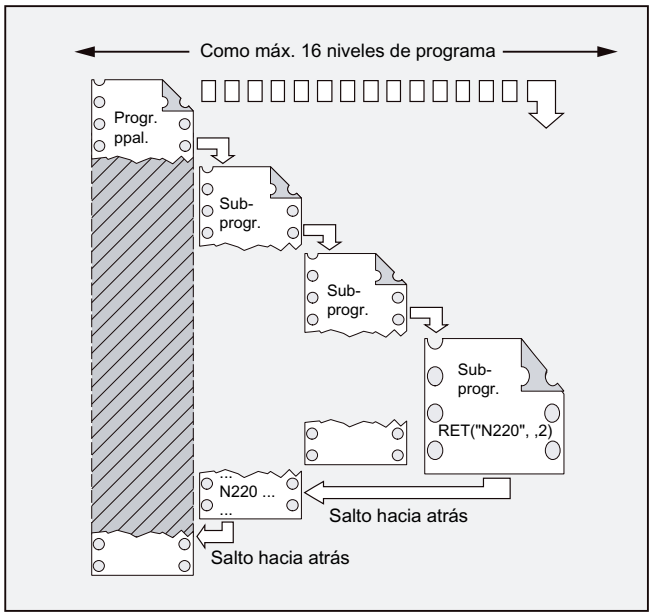
Después del comando `RET`, continúa la ejecución del programa con la secuencia `N200` en el programa principal.

2º parámetro de salto atrás 1 = "N200", parámetro de salto atrás 2 = 1



Después del comando `RET`, continúa la ejecución del programa con la secuencia (`N210`) que sigue a la secuencia `N200` en el programa principal.

3er parámetro de salto atrás 1 = "N220", parámetro de salto atrás 3 = 2



Después del comando `RET`, se salta atrás dos niveles de programa hacia atrás y continúa la ejecución del programa con la secuencia `N220`.

1.25.3 Llamada de un subprograma

1.25.3.1 Llamada de subprograma sin transferencia de parámetros

Función

La llamada de un subprograma se realiza bien con la dirección L y el número de subprograma, bien indicando el nombre de programa.

También se puede llamar a un programa principal como si se tratase de un subprograma. El fin de programa `M2` o `M30`, definido en el programa principal, se evalúa en este caso como `M17` (fin de programa con salto atrás al programa desde el que se realiza la llamada).

Nota

También se puede arrancar un subprograma como si fuese un programa principal.

Estrategia de búsqueda del control numérico:

¿Hay *_MPF?

¿Hay *_SPF?

De ello resulta: en el caso de que el nombre del subprograma llamado sea idéntico al nombre del programa principal desde donde se realiza la llamada, se abre de nuevo el programa desde el que se realiza la llamada. Este efecto, generalmente no deseado, se deberá evitar mediante la asignación unívoca de nombres tanto a programas como a subprogramas.

Nota

Los subprogramas que no requieren transferencia de parámetros también pueden llamarse desde un fichero de inicialización.

Sintaxis

```
L<Número>/<Nombre de programa>
```

Nota

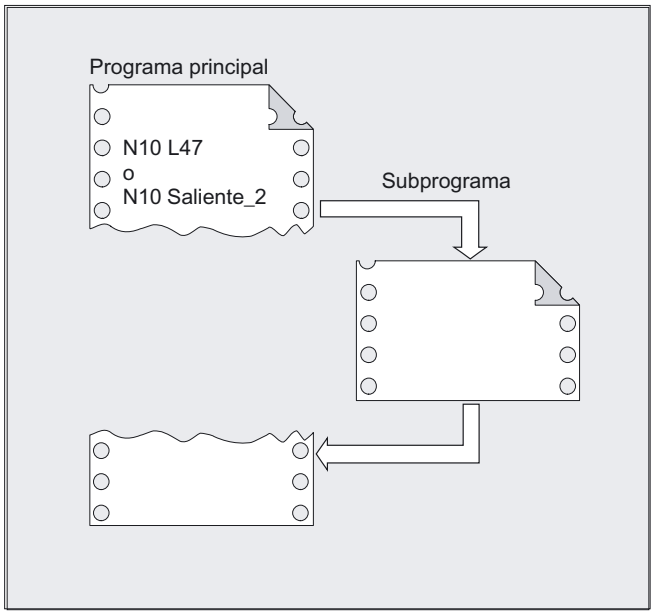
La llamada de un subprograma se tiene que programar siempre en una secuencia CN propia.

Descripción

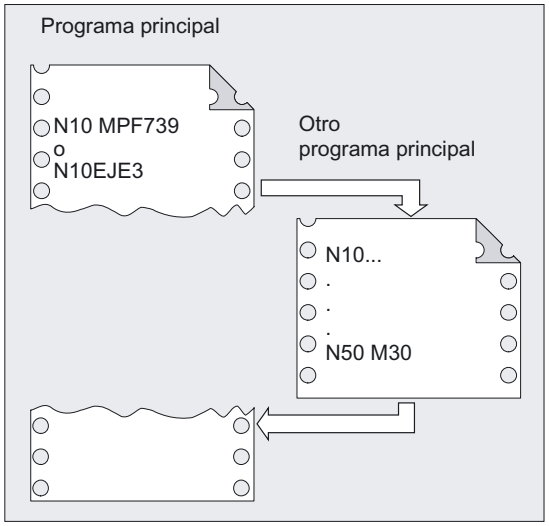
`L`:	Dirección para la llamada del subprograma
`<Número>`:	Número del subprograma Tipo: INT Valor: Como máximo 7 números decimales **Atención:** Los ceros a la izquierda tienen significado al dar nombre (⇒ L123, L0123 y L00123 son tres subprogramas diferentes).
`<Nombre de programa>`:	Nombre del subprograma (o del programa principal)

Ejemplos

Ejemplo 1: Llamada de un subprograma sin transferencia de parámetros



Ejemplo 2: Llamada de un programa principal como subprograma



1.25.3.2 Llamada de subprograma con transferencia de parámetros (EXTERN)

Función

En la llamada de subprograma con transferencia de parámetros se pueden transferir directamente variables o valores (no en parámetros `VAR`).

Los subprogramas con transferencia de parámetros se deben declarar con `EXTERN` antes de la llamada en el programa principal (p. ej., al inicio del programa). Para ello se indican el nombre del subprograma y los tipos de variables en el orden en que se transfieren.

⚠ PRECAUCIÓN

Tanto los tipos de variables como el orden de transferencia deben coincidir con las definiciones establecidas con `PROC` en el subprograma. Los nombres de los parámetros pueden ser distintos en el programa principal y en el subprograma.

Sintaxis

```
EXTERN <Nombre del programa>(<Tipo Par1>,<Tipo Par2>,<Tipo Par3>)
...
<Nombre del programa>(<Valor Par1>,<Valor Par2>,<Valor Par3>)
```

⚠ PRECAUCIÓN

La llamada de subprograma se tiene que programar en una secuencia CN propia.

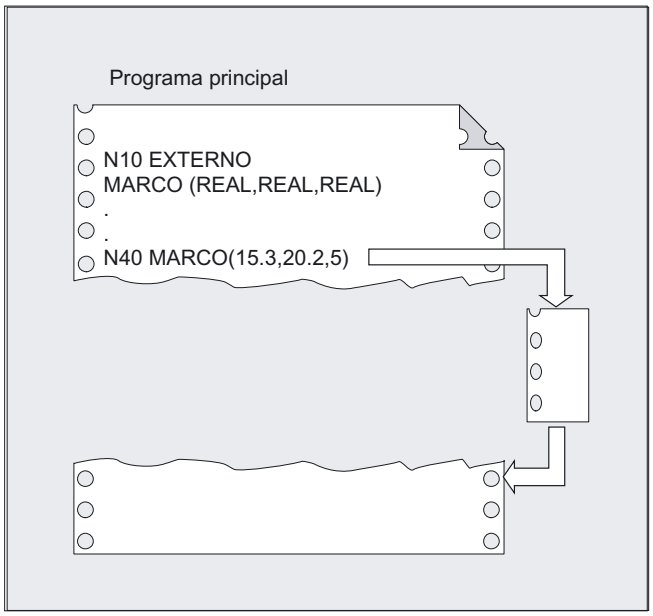
Descripción

<Nombre de programa>:	Nombre del subprograma
EXTERN:	Palabra reservada para declarar un subprograma con transferencia de parámetros **Nota:** EXTERN sólo debe indicarse cuando el subprograma está en el directorio de piezas o en el directorio global de subprogramas. Los ciclos no se declaran con EXTERN.
<Tipo Par1>,<Tipo Par2>,<Tipo Par3>:	Tipos de variable de los parámetros que deben transferirse en el orden de transmisión
<Valor Par1>,<Valor Par2>,<Valor Par3>:	Valores de variable para los parámetros que deben transferirse

Ejemplos

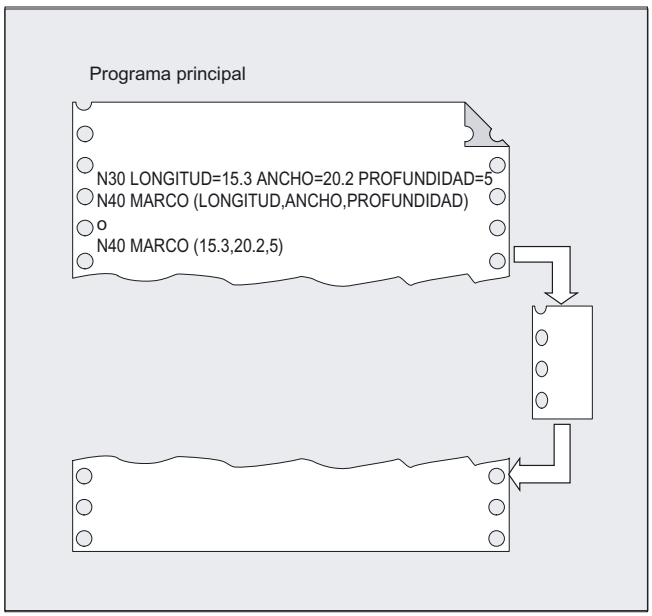
Ejemplo 1: Llamada de subprograma con declaración anterior

Código del programa	Comentarios
N10 MARCO EXTERNO(REAL,REAL,REAL)	; Indicación del subprograma
...	
N40 MARCO(15.3,20.2,5)	; Llamada del subprograma con transferencia de parámetros.



Ejemplo 2: Llamada de subprograma sin declaración

```
Código del programa                                  Comentarios
N10 DEF REAL LARGO, ANCHO, PROFUNDIDAD
N20 …
N30 LONGITUD=15.3 ANCHO=20.2 PROFUNDIDAD=5
N40 MARCO(LONGITUD,ANCHO,PROFUNDIDAD)        ; o: N40 MARCO(15.3,20.2,5)
```



1.25.3.3 Cantidad de repeticiones de programa (P)

Función

Si se debe ejecutar un subprograma varias veces seguidas, en la secuencia de llamada del subprograma se puede programar el número de repeticiones deseado con la dirección `P`.

⚠PRECAUCIÓN

Llamada de subprograma con repetición de programa y transferencia de parámetros

Los parámetros se transfieren sólo en el momento de la llamada al programa o en la primera pasada. Para el resto de pasadas no se alteran los parámetros. Si se desea modificar los parámetros en las repeticiones de programa es necesario realizar las definiciones correspondientes en el subprograma.

Sintaxis

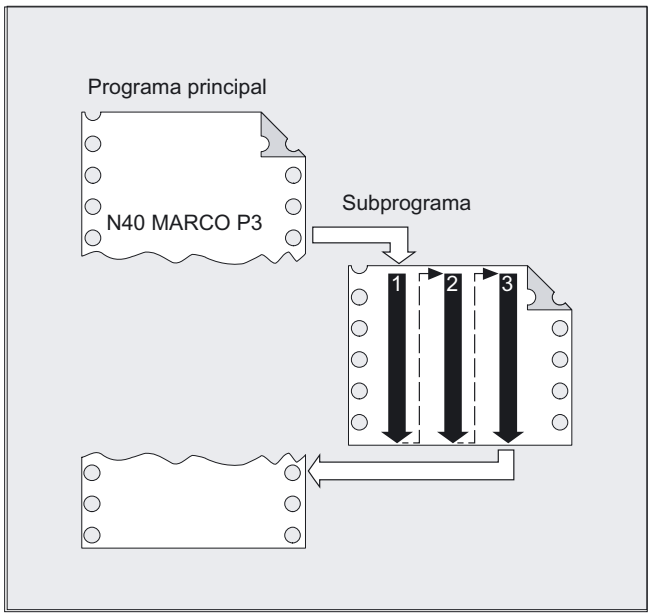
```
<Nombre de programa> P<Valor>
```

Descripción

`<Nombre de programa>`:	Llamada a un subprograma	
`P`:	Dirección para la programación de repeticiones de programa	
`<Valor>`:	Cantidad de repeticiones de programa	
	Tipo:	INT
	Rango de valores:	1 … 9999 (sin signo)

Ejemplo

```
Código del programa  Comentarios
...
N40 MARCO P3         ; El subprograma MARCO debe ejecutarse tres veces seguidas.
...
```



1.25.3.4 Llamada de subprograma modal (MCALL)

Función

En una llamada de subprograma modal con MCALL se llama automáticamente al subprograma después de cada secuencia con desplazamiento de trayectoria y dicho subprograma se ejecuta. De esta forma se automatiza la llamada de subprogramas que deben ejecutarse en distintas posiciones de pieza (p. ej., para la fabricación de figuras de taladrado).

La desconexión de la función se realiza con MCALL sin llamada de subprograma o mediante la programación de una nueva llamada de subprograma modal para un nuevo subprograma.

⚠ PRECAUCIÓN

En una ejecución de programa sólo puede ser efectiva al mismo tiempo una llamada MCALL. Los parámetros sólo se transmiten una vez en una llamada MCALL.

El programa modal se llama también en las siguientes situaciones, aunque no esté programado ningún movimiento:

- En la programación de las direcciones S y F si está activo G0 o G1.
- Si G0/G1 se ha programado solo en la secuencia o con otros códigos G.

Sintaxis

```
MCALL <Nombre de programa>
```

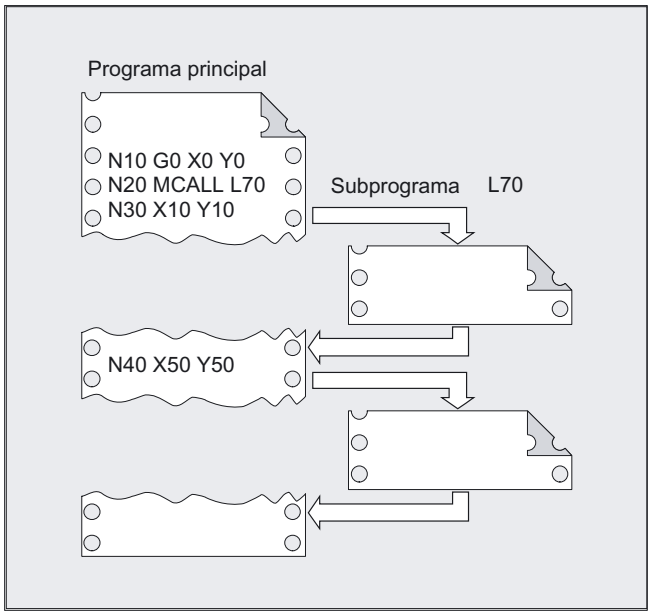
Descripción

MCALL:	Comando para la llamada de subprograma modal
<Nombre de programa>:	Nombre del subprograma

Ejemplos

Ejemplo 1:

Código de programa	Comentarios
N10 G0 X0 Y0	
N20 MCALL L70	; Llamada de subprograma modal.
N30 X10 Y10	; Se realiza el desplazamiento a la posición programada y, a continuación, se ejecuta el subprograma L70.
N40 X50 Y50	; Se realiza el desplazamiento a la posición programada y, a continuación, se ejecuta el subprograma L70.



Ejemplo 2:

Código del programa
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

En este ejemplo, las secuencias CN siguientes con ejes de contorneado programados se encuentran en el subprograma L80. L70 se llama a través de L80.

1.25.3.5 Llamada de subprograma indirecta (CALL)

Función

En dependencia de las condiciones dadas se pueden llamar en un mismo punto diferentes subprogramas. A tal efecto, se memoriza el nombre del subprograma en una variable del tipo STRING. La llamada de subprograma se efectúa con `CALL` y el nombre de la variable.

> **⚠ PRECAUCIÓN**
>
> La llamada indirecta de subprograma sólo resulta posible para subprogramas sin transferencia de parámetros. Para una llamada directa a un subprograma, el nombre debe consignarse en una constante STRING.

Sintaxis

```
CALL <nombre programa>
```

Descripción

`CALL`:	Comando para la llamada de subprograma indirecta
`<Nombre de programa>`:	Nombre del subprograma (variable o constante) Tipo: STRING

Ejemplo

Llamada directa con constante STRING:

```
Código de programa                                          Comentarios
…
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PIEZA1_SPF"             ; Llamar directamente a
                                                              subprograma PIEZA1 con
                                                              CALL.
…
```

Llamada indirecta mediante variable:

```
Código de programa                                          Comentarios
…
DEF STRING[100] PROGNAME                                    ; Definir variable.
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PIEZA1_SPF"         ; Asignar subprograma PIEZA1
                                                              a la variable PROGNAME.
CALL PROGNAME                                               ; Llamar indirectamente
                                                              subprograma PIEZA1
                                                              mediante CALL y la
                                                              variable PROGNAME.
…
```

1.25.3.6 Llamada de programa indirecta con indicación de la sección de programa que se va a ejecutar (CALL BLOCK ... TO ...)

Función

Con `CALL` y la combinación de palabras reservadas `BLOCK ... TO` se llama indirectamente a un subprograma y se ejecuta la sección del programa identificada con la marca inicial y la marca final.

Sintaxis

```
CALL <>Nombre de programa> BLOCK <Marca inicial> TO <Marca final>
CALL BLOCK <Marca inicial> TO <Marca final>
```

Descripción

`CALL`:	Comando para la llamada de subprograma indirecta
`<Nombre de programa>`:	Nombre del subprograma (variable o constante) que contiene la sección del programa que debe ejecutarse (**indicación opcional**). Tipo: STRING **Nota:** Si no hay programado ningún `<Nombre de programa>`, se busca y ejecuta la sección del programa identificada con `<Marca inicial>` y `<Marca final>` en el programa actual.
`BLOCK ... TO ...` :	Combinación de palabras reservadas para la ejecución indirecta de secciones del programa
`<Marca inicial>`:	Variable que remite al inicio de la parte del programa que debe ejecutarse. Tipo: STRING
`<Marca final>`:	Variable que remite al final de la parte del programa que debe ejecutarse. Tipo: STRING

Ejemplo

Programa principal:

```
Código de programa                                  Comentarios
...
DEF STRING[20] STARTLABEL, ENDLABEL                 ; Definición de variables para la
                                                      marca inicial y la final.
STARTLABEL="LABEL_1"
ENDLABEL="LABEL_2"
...
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL        ; Llamada de subprograma indirecta
                                                      e identificación de la sección de
                                                      programa que va a ejecutarse.
...
```

Subprograma:

```
Código del programa        Comentarios
PROC CONTUR_1 ...
LABEL_1                    ; Marca inicial: inicio de la ejecución de la sección de
                             programa
N1000 G1 ...
...
LABEL_2                    ; Marca final: fin de la ejecución de la sección de
                             programa
...
```

1.25.3.7 Llamada indirecta a un programa creado en lenguaje ISO (ISOCALL)

Función

Con la llamada de programa indirecta `ISOCALL` puede llamarse a un programa programado en lenguaje ISO. Con ello se activa el modo ISO ajustado en los datos de máquina. Al final del programa el modo de procesamiento original vuelve a ser efectivo. Si no hay ningún modo ISO ajustado en los datos de máquina, la llamada del subprograma se efectúa en modo Siemens.

Para más información sobre el modo ISO, ver:
Bibliografía:
Descripción de funciones Dialectos ISO

Sintaxis

```
ISOCALL <nombre programa>
```

Descripción

`ISOCALL`:	Palabra reservada para la llamada de subprograma indirecta con la que se activa el modo ISO ajustado en los datos de máquina.
`<Nombre de programa>`:	Nombre del programa programado en un lenguaje ISO (variable o constante del tipo STRING)

Ejemplo: Llamada de un contorno con programación de ciclos desde el modo ISO

Código del programa	**Comentarios**
`0122_SPF`	`; Descripción de contorno en el modo ISO`
`N1010 G1 X10 Z20`	
`N1020 X30 R5`	
`N1030 Z50 C10`	
`N1040 X50`	
`N1050 M99`	
`N0010 DEF STRING[5] PROGNAME = "0122"`	`; Programa de pieza Siemens (ciclo)`
`...`	
`N2000 R11 = $AA_IW[X]`	
`N2010 ISOCALL PROGNAME`	
`N2020 R10 = R10+1`	`; Ejecutar programa 0122.spf en el modo ISO`
`...`	
`N2400 M30`	

1.25.3.8 Llamada de subprograma con ruta de acceso y parámetros (PCALL)

Función

Con PCALL pueden llamarse subprogramas con indicación de ruta absoluta y transferencia de parámetros.

Sintaxis

```
PCALL <Ruta/nombre de programa>(<Parámetro 1>,…,<Parámetro n>)
```

Descripción

PCALL:	Palabra reservada para llamada de subprograma con ruta de acceso absoluta.
<Ruta/nombre de programa>:	Definición absoluta de ruta comenzando por "/", incluso nombre de subprograma. Si no se ha indica ninguna ruta absoluta, PCALL se comporta como una llamada de subprograma estándar con identificador de programa. El identificador de programa se indica sin prólogo _N_ ni extensión. Si el nombre del programa debe programarse con prólogo y extensión, debe declararse explícitamente con prólogo y extensión con el comando EXTERN.
<Parámetro 1>, ...,:	Parámetro actual según la instrucción PROC del subprograma.

Ejemplo

Código del programa
PCALL/_N_WKS_DIR/_N_WELLE_WPD/WELLE(Parámetro 1, parámetro 2,…)

1.25.3.9 Ampliación de la ruta de búsqueda en la llamada de subprograma (CALLPATH)

Función

El comando `CALLPATH` permite ampliar la ruta de búsqueda en las llamadas de subprograma.

Así pueden llamarse también subprogramas desde un directorio de pieza no seleccionado sin indicar el nombre de ruta absoluto y completo del subprograma.

La ampliación de la ruta de búsqueda tiene lugar antes de la entrada de ciclos de usuario (_N_CUS_DIR).

Los siguientes sucesos permiten anular la selección de la ampliación de la ruta de búsqueda:

- `CALLPATH` con espacio
- `CALLPATH` sin parámetros
- Fin del programa de pieza
- Reset

Sintaxis

```
CALLPATH("<Nombre de ruta>")
```

Descripción

`CALLPATH`:	Palabra reservada para la ampliación de la ruta de búsqueda programable. Se programa en una línea propia del programa de pieza.
`<Nombre de ruta>`:	Constante o variable del tipo STRING. Contiene la indicación de ruta absoluta de un directorio con la que debe ampliarse la ruta de búsqueda. La indicación de ruta comienza por "/". La ruta debe indicarse completa con prefijos y sufijos. La longitud máxima de la ruta es de 128 bytes. Si el <Nombre de ruta> contiene un espacio o se llama a `CALLPATH` sin parámetro, se restablece la instrucción de ruta de búsqueda.

Ejemplo

Código de programa

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

Con ello se ajusta la siguiente ruta de búsqueda (el punto 5 es nuevo):

1. Directorio raíz actual/identificador de subprograma
2. Directorio raíz actual/identificador de subprograma_SPF
3. Directorio raíz actual/identificador de subprograma_MPF
4. /_N_SPF_DIR/identificador de subprograma_SPF
5. **/_N_WKS_DIR/_N_MYWPD/identificador de subprograma_SPF**
6. /N_CUS_DIR/_N_MYWPD/identificador de subprograma_SPF
7. /_N_CMA_DIR/identificador de subprograma_SPF
8. /_N_CST_DIR/identificador de subprograma_SPF

Limitaciones

- CALLPATH verifica si el nombre de la ruta programado realmente existe. En caso de fallo, la ejecución del programa de pieza se interrumpe con la alarma de secuencia a corregir 14009.
- CALLPATH puede programarse también en ficheros INI. Es efectivo mientras dura la ejecución del fichero INI (fichero WPD-INI o programa de inicialización para datos CN activos, p. ej., frames en el 1.er canal _N_CH1_UFR_INI). Después de esto se restablece la ruta de búsqueda.

1.25.3.10 Ejecutar subprograma externo (EXTCALL)

Función

Con EXTCALL puede recargar un programa de HMI en el modo "Ejecución de externo". Se pueden recargar y ejecutar todos los programas alcanzables a través de la estructura de directorios del HMI.

Sintaxis

```
EXTCALL("<Ruta/Nombre de programa>")
```

Descripción

EXTCALL:	Comando para llamar a un subprograma externo
<Ruta/nombre de programa>:	Constante/variable del tipo STRING Se puede indicar una ruta absoluta (o una ruta relativa) o un nombre de programa. El nombre de programa se indica con o sin prólogo _N_ y sin extensión. Se puede añadir una extensión al nombre de programa con el carácter <_>. Ejemplo: "/_N_WKS_DIR/_N_EJE_WPD/_N_EJE_SPF" o "EJE"

Nota

Los subprogramas externos no deben contener instrucciones de salto como GOTOF, GOTOB, CASE, FOR, LOOP, WHILE o REPEAT.

Existe la posibilidad de construcciones IF-ELSE-ENDIF.

Son posibles las llamadas de subprograma y las llamadas EXTCALL imbricadas.

RESET, POWER ON

Con RESET y POWER ON se cancelan las llamadas de subprograma externas y se borran las correspondientes memorias de recarga.

Un subprograma seleccionado para "Ejecución de externo" sigue seleccionado para "Ejecución de externo" después de RESET/fin de programa de pieza. La selección se pierde con POWER ON.

Ejemplos

1. Ejecución desde el disco duro local

Sistemas: SINUMERIK solution line/powerline con HMI Advanced

El programa principal "_N_MAIN_MPF" se encuentra en la memoria CN y está seleccionado para la ejecución:

Código del programa

```
N010 PROC MAIN
N020 ...
N030 EXTCALL ("DESBASTE")
N040 ...
N050 M30
```

El subprograma que se desea recargar "_N_DESBASTE_SPF" se encuentra en el disco duro local en el directorio "_N_WKS_DIR/_N_WST1".

La ruta al subprograma está predefinida en SD42700:

SD42700 $SC_EXT_PROG_PATH = "_N_WKS_DIR/_N_WST1"

Código del programa

```
N010 PROC DESBASTE
N020 G1 F1000
N030 X= ... Y= ... Z= ...
N040 ...
...
...
N999999 M17
```

2. Ejecución desde la unidad de red

Sistemas: SINUMERIK solution line/powerline con HMI sl/HMI Advanced/HMI Embedded

El programa que se desea recargar "Contorno2.spf" se encuentra en la unidad de red en el directorio "\\R4711\Piezas".

Código del programa

```
...
N... EXTCALL("\\R4711\Piezas\Contorno2.spf")
...
```

Indicación de una ruta de programa externa

La ruta al directorio del subprograma externo puede predefinirse con el dato de operador:

SD42700 $SC_EXT_PROG_PATH

Junto con la ruta o identificador del subprograma indicado en la llamada `EXTCALL` resulta la ruta completa del programa que se desea llamar.

Efectos

Llamada EXTCALL con definición de ruta absoluta

Si el subprograma existe en la ruta indicada, se ejecuta después de la llamada `EXTCALL`. Si no existe, se interrumpe la ejecución del programa.

Llamada EXTCALL con definición de ruta relativa/sin definición de ruta

En una llamada `EXTCALL` con definición de ruta relativa o sin definición de ruta se buscan las memorias de programas existentes según el siguiente modelo:

- Si hay definición de ruta predefinida en SD42700, se busca primero partiendo de esta ruta la indicación en la llamada `EXTCALL` (nombre de programa, en su caso, con definición de ruta relativa). La ruta absoluta resulta entonces de la concatenación de caracteres de:
  - la definición de ruta predefinida en SD42700
  - el signo "/" como signo de separación
  - la ruta de acceso del subprograma o el identificador indicado en `EXTCALL`
- Si el subprograma llamado no se encuentra en la ruta de acceso predefinida, se busca seguidamente la indicación en la llamada `EXTCALL` dentro de los directorios de la memoria de usuario.
- Si el subprograma llamado no se encuentra en la memoria de programas actualmente examinada (p.ej. tarjeta CompactFlash), se examina la siguiente memoria de programas (p.ej. unidad de red) según el punto 1 y 2.
- La búsqueda finaliza cuando el subprograma se haya encontrado por primera vez. Si la búsqueda no da ningún resultado se cancela el programa.

Nota

SINUMERIK powerline con HMI Embedded

Con SINUMERIK powerline con HMI Embedded se tiene que indicar siempre una ruta absoluta.

Memoria de programas externa

Dependiendo del sistema (SINUMERIK solution line/powerline), de la interfaz de usuario existente (HMI sl/HMI Advanced/HMI Embedded) y de las opciones adquiridas, pueden encontrarse memorias de programas externas en los siguientes soportes de datos:

- Tarjeta CompactFlash
- Unidad de red
- Unidad USB
- Disco duro local

Nota

Ejecución de externo a través de interfaz USB en SINUMERIK solution line

Si deben transferirse programas externos desde una unidad USB (FlashDrive USB) externa a través de la interfaz USB, sólo debe utilizarse la interfaz de X203 con el nombre "TCU_1".

ATENCIÓN

- Las FlashDrives USB no son adecuadas como soportes de memoria persistentes.
- No se recomienda el uso de FlashDrives USB para "Ejecución de externo".
  Motivos:
  - Las FlashDrives USB pueden desprenderse o puede haber problemas de contacto durante su funcionamiento. A consecuencia de ello, se interrumpiría el proceso.
  - Las FlashDrives USB pueden romperse con un golpe y dañar el panel de operador.

Nota

Ejecución de externo a través de interfaz V24 en SINUMERIK powerline

Con HMI Embedded pueden transferirse al CN programas externos con el pulsador de menú "Ejecución de externo" a través de la interfaz V24.

Memoria de recarga ajustable (búfer FIFO)

Para la ejecución de un programa en el modo "Ejecución de externo" (programa principal o subprograma) se precisa una memoria de recarga en el NCK. El tamaño de la memoria de recarga está definido en 30 Kbytes y sólo el fabricante de la máquina puede modificarlo según las necesidades, al igual que otros datos de máquina relevantes para la memoria.

Para todos los programas (programas principales o subprogramas) que se ejecutan simultáneamente en el modo "Ejecución de externo" se tiene que ajustar una memoria de recarga.

Fabricante de la máquina

Diríjase al fabricante de la máquina cuando deba ampliar el tamaño y la cantidad de memorias de recarga.

Para más información sobre "Ejecución de externo", ver:
Bibliografía:
Manual de funciones, Funciones básicas; BAG/GMO, canal, modo de programa (K1)

Visualización de secuencia, secuencia a secuencia y comportamiento en parada CN

Al ejecutar desde el disco duro, así como en EXTCALL, sólo es posible la visualización de 3 secuencias de HMI Advanced "Ejecución de programa" Este ajuste se mantiene durante la secuencia a secuencia o el estado Parada CN.

1.25.4 Ciclos

1.25.4.1 Ciclos: Parametrizar ciclos de usuario

Función

Los ficheros cov.com y uc.com permiten parametrizar ciclos propios.

El fichero cov.com se suministra con los ciclos estándar y se tiene que ampliar en consecuencia. El fichero uc.com debe ser creado por el mismo usuario.

Ambos ficheros se tienen que cargar en el sistema de ficheros pasivo al directorio "Ciclos de usuario" (o dotar de la correspondiente ruta de acceso):

```
;$PATH=/_N_CUS_DIR
```

en el programa

Ficheros y rutas

`cov.com_COM`	Vista general de los ciclos
`uc.com`	Descripción de la llamada de ciclos

Adaptación cov.com – Vista general de los ciclos

El fichero cov.com suministrado con los ciclos estándar tiene la siguiente estructura:

`%_N_COV_COM`	Nombre de fichero
`;$PATH=/_N_CST_DIR`	Ruta de acceso
`;Vxxx 11.12.95 Sca Vista general de ciclos`	Línea de comentario
`C1(CYCLE81) Mandrinar, centrar (puntear)`	Llamada para el primer ciclo
`C2(CYCLE82) Mandrinar, avellanar`	Llamada para el 2º ciclo
`...`	
`C24(CYCLE98) Concatenación de roscas`	Llamada para el último ciclo
`M17`	Fin del fichero

Sintaxis

Para cada nuevo ciclo añadido se tiene que insertar una línea con la siguiente sintaxis:

```
C<Número> (<Nombre de ciclo>) Texto de comentario
```

Número: cualquier número entero que no se haya utilizado hasta entonces en el fichero;

Nombre de ciclo: el nombre de programa del ciclo a incorporar

Texto de comentario: un texto de comentario opcional para el ciclo

Ejemplo:

```
C25 (MI_CICLO_1) Ciclo_usuario_1
C26 (CICLO ESPECIAL)
```

Ejemplo: fichero uc.com (descripción de ciclos de usuario)

La aclaración se realiza en base a la continuación del ejemplo:
Para los dos ciclos siguientes se quiere crear una nueva parametrización de ciclo:

```
Programación          Comentarios
PROC MI_CICLO_1 (REAL PAR1, INT PAR2, CHAR PAR3, STRING[10] PAR4)
El ciclo tiene los siguientes parámetros de transferencia:
PAR1:                 ; Valor real en un rango de -1000.001 <= PAR2 <= 123.456,
                        ajuste previo 100
PAR2:                 ; Valor entero positivo entre 0 <= PAR3 <= 999999, asignación
                        previa 0
PAR3:                 ; 1 carácter ASCII
PAR4:                 ; String con la longitud 10 para un nombre de subprograma
...
M17                   ;
```

```
Programación          Comentarios
PROC CICLO ESPECIAL (REAL VALOR1, INT VALOR2)
El ciclo tiene los siguientes parámetros de transferencia:
VALOR1:               ; Valor real sin limitación del rango de valores ni ajuste
                        previo
VALOR2:               ; Valor entero sin limitación del rango de valores ni ajuste
                        previo
...
M17                   ;
```

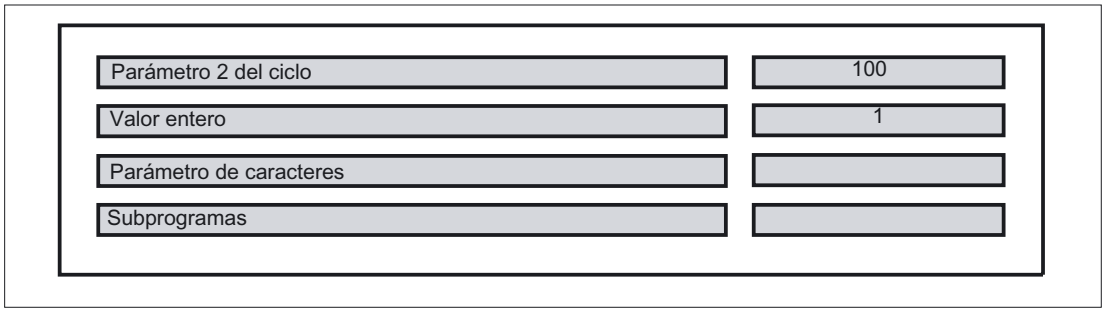
Fichero correspondiente uc.com:

Programación

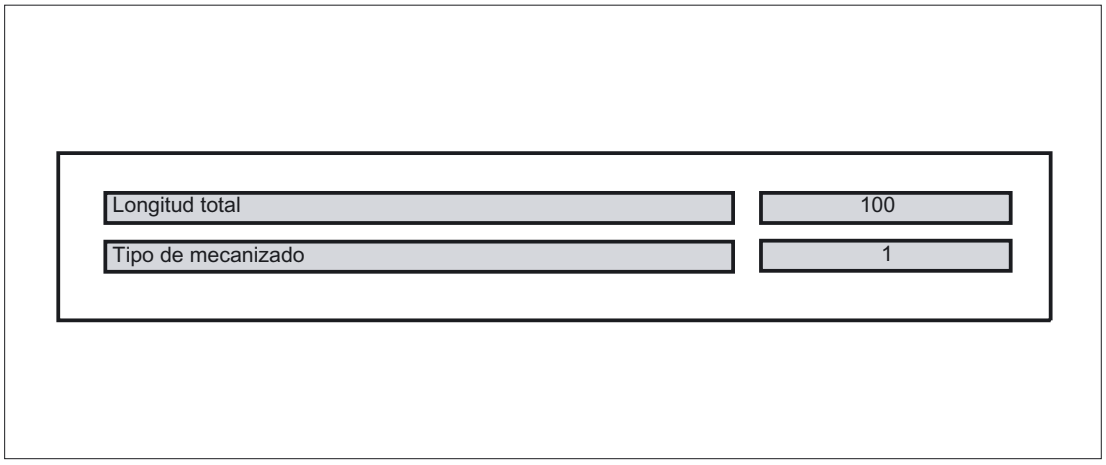
```
%_N_UC_COM
;$PATH=/_N_CUS_DIR
//C25 (MI_CICLO_1) Ciclo_usuario_1
(R/-1000.001 123.456 / 100 /Parámetro_2 del ciclo)
(I/0 999999 / 1 / valor entero)
(C//"A" / parámetro de caracteres)
(S///nombre subprograma)
//C26(CICLO ESPECIAL)
(R///Longitud total)
(I/*123456/3/tipo de mecanizado)
M17
```

Ejemplo: ambos ciclos

Máscara de visualización para el ciclo `MI_CICLO_1`



Máscara de visualización para el ciclo `CICLO ESPECIAL`



Descripción de sintaxis para el fichero uc.com - Descripción de ciclos de usuario

Línea de encabezamiento por ciclo:
como en el fichero cov.com con "//" previo

```
//C <Número> (<Nombre de ciclo>) Texto de comentario
```

Ejemplo:

```
//C25 (MI_CICLO_1) Ciclo_usuario_
```

Línea para la descripción por parámetros:

```
(<Identificador de tipo de datos>/<Valor mínimo> <Valor máximo>
/ <Valor de ajuste previo>/<Comentario>)
```

Identificación del tipo de fichero:

R	Real
I	Entero
C	Carácter (1 carácter)
S	String

Valor mínimo, valor máximo (se puede suprimir)

Límites del valor a introducir que se comprueban en la entrada; no se pueden introducir valores que se sitúan fuera de este margen. Se pueden indicar valores de enumeración que se pueden manejar con la tecla de conmutación. Éstos se cuentan en orden ascendente, empezando por "*"; en este caso no se admiten otros valores.

Ejemplo:

```
(I/*123456/1/tipo de mecanizado)
```

En los tipos String y Carácter no existen límites.

Valor de ajuste previo (se puede suprimir)

El valor que existe como ajuste previo en la correspondiente máscara al llamar al ciclo; se puede modificar mediante una operación.

Comentarios

Texto de máx. 50 caracteres que se muestra en la máscara de llamada para el ciclo delante del campo de entrada para el parámetro.

1.26 Técnica de macros (DEFINE ... AS)

> **⚠ PRECAUCIÓN**
>
> ¡Con la técnica de macros, el lenguaje de programación del control se puede ver modificado fuertemente! Por esta razón, proceda con la máxima precaución al utilizar la técnica de macros.

Función

Se denomina macto a una recopilación de instrucciones individuales en una nueva instrucción global con una denominación propia. También las funciones G, M y H o los nombres de subprograma L se pueden crear como macros. Al llamar la macro en la ejecución del programa, las instrucciones programadas bajo el nombre de la macro se ejecutan sucesivamente.

Aplicación

Las secuencias de instrucciones que se repiten se programan sólo una vez como macro en un módulo de macro propio (fichero de macros) o una vez al inicio del programa. Entonces, la macro se puede llamar y ejecutar en cualquier programa principal o subprograma.

Activación

Para poder utilizar las macros de un fichero de macros en el programa CN, el fichero de macros debe cargarse en el CN.

Sintaxis

Definición de macro:
```
DEFINE <Nombre de la macro> AS <Instrucción 1> <Instrucción 2> ...
```
Llamada en el programa CN:
```
<Nombre de la macro>
```

Descripción

`DEFINE ... AS` :	Combinación de palabras reservadas para definir una macro
`<Nombre de la macro>`:	Nombre de la macro Como nombres de macro sólo se admiten identificadores. Con el nombre de la macro ésta se llama desde el programa CN.
`<Instrucción>`:	Instrucción de programa que debe incluir la macro.

Reglas para la definición de la macro

- En la macro pueden definirse tantos identificadores, funciones G, M, H y nombres de programa L como se desee.
- También es posible definir las macros en el programa CN.
- Las macros de función G sólo pueden definirse en el módulo de macro de manera global para todo el control.
- Las funciones H y L deben programarse con 2 dígitos.
- Se pueden programar funciones M y G de 3 dígitos.

⚠PRECAUCIÓN

Las palabras reservadas y los nombres reservados no deben sobredefinirse con macros.

Limitaciones

La imbricación de macros no es posible.

Ejemplos

Ejemplo 1: Definición de macro al inicio del programa

Código del programa	Comentarios
DEFINE LINIE AS G1 G94 F300	; Definición de macro
...	
...	
N70 LÍNEA X10 Y20	; Llamada de macro
...	

Ejemplo 2: Definiciones de macro en un fichero de macros

Código del programa	Comentarios
DEFINE M6 AS L6	; En el cambio de herramienta se llama a un subprograma que asume la transferencia de datos necesaria. En el subprograma se emite la función M de cambio de herramienta propiamente dicha (p. ej. M106).
DEFINE G81 AS DRILL(81)	; Emulación de la función G DIN.
DEFINE G33 AS M333 G333	; En el roscado se solicita la sincronización con el PLC. La función G original G33 ha sido renombrada por DM en G333; la programación permanece idéntica para el usuario.

Ejemplo 3: Fichero de macros externo

Después de leer el fichero de macros externo en el control es preciso cargarlo en el CN. Tan sólo después pueden utilizarse las macros en el programa CN.

Código del programa	**Comentarios**
%_N_UMAC_DEF	
;$PATH=/_N_DEF_DIR	; Macros personalizadas
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; Cabezal derecho, refrigerante CON
DEFINE M14 AS M4 M7	; Cabezal izquierdo, refrigerante CON
DEFINE M15 AS M5 M9	; Parada del cabezal, refrigerante DES
DEFINE M6 AS L6	; Llamada del programa de cambio de herramienta
DEFINE G80 AS MCALL	; Cancelación ciclo de taladrado
M30	

Administración de programas y ficheros 2

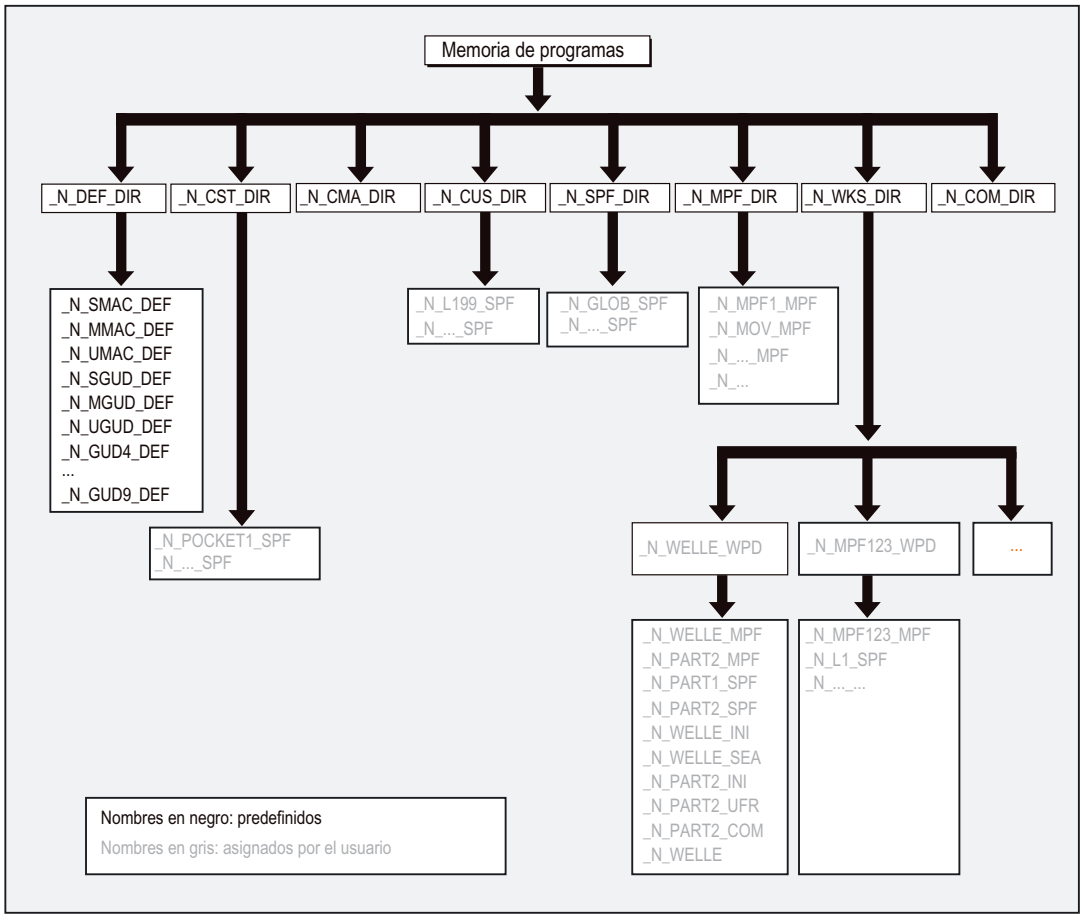
2.1 Memoria de programas

Función

En la memoria de programas se guardan ficheros y programas (p. ej. programas principales y subprogramas, definiciones de macro) de modo persistente (→ sistema pasivo de ficheros).

Bibliografía:
Manual de funciones de ampliación; Configuración de memoria (S7)

Además, ciertos tipos de fichero pueden guardarse aquí para transferirlos en caso necesario (p. ej., al mecanizar una pieza determinada) a la memoria de trabajo (p. ej., para inicialización).



Directorios estándar

Como estándar existen los siguientes directorios:

Directorio	Índice
_N_DEF_DIR	Bloques de datos y módulos de macro
_N_CST_DIR	Ciclos estándar
_N_CMA_DIR	Ciclos del fabricante
_N_CUS_DIR	Ciclos de usuario
_N_WKS_DIR	Piezas
_N_SPF_DIR	Subprogramas globales
_N_MPF_DIR	Programas principales
_N_COM_DIR	Comentarios

Tipos de fichero

En la memoria de programas se pueden almacenar los siguientes tipos de fichero:

Tipo de fichero	Descripción
name_MPF	Programa principal
name_SPF	Subprograma
name_TEA	Dato de Máquina (DM)
name_SEA	Datos del operador
name_TOA	Correcciones de herramientas
name_UFR	Decalajes de origen/frames
name_INI	Fichero de inicialización
name_GUD	Datos globales del usuario
name_RPA	Parámetros R
name_COM	Comentarios
name_DEF	Definición para datos de usuario globales y macros

Directorio raíz de piezas (_N_WKS_DIR)

El directorio raíz de piezas está creado por defecto con el nombre `_N_WKS_DIR` en la memoria de programas de pieza. En el directorio raíz de piezas se encuentran los correspondientes directorios de piezas de todas las piezas programadas.

Directorios raíz de piezas (..._WPD)

Para facilitar el manejo de datos y programas, se pueden agrupar determinados datos y programas o bien se pueden almacenar en directorios de pieza individuales.

Un directorio de pieza contiene todos los ficheros necesarios para mecanizar dicha pieza. Estos pueden ser programas principales, subprogramas, ficheros de inicialización y ficheros de comentario.

Los programas de inicialización se ejecutan una vez después de seleccionar el programa con el primer inicio de programa de pieza (según el dato de máquina DM11280 $MN_WPD_INI_MODE).

Ejemplo:

El directorio de pieza _N_EJE_WPD creado para la pieza EJE contiene los siguientes ficheros:

Archivo	Descripción
_N_EJE_MPF	Programa principal
_N_PART2_MPF	Programa principal
_N_PART1_SPF	Subprograma
_N_PART2_SPF	Subprograma
_N_EJE_INI	Programa general de inicialización de datos para la pieza
_N_EJE_SEA	Programa de inicialización para los datos de operador
_N_PART2_INI	Programa general de inicialización para los datos del programa Part 2
_N_PART2_UFR	Programa de inicialización para los datos del frame asociados al programa Part 2
_N_EJE_COM	Fichero de comentarios

Almacenar directorios de programas de pieza en un PC externo

A continuación, se describe el procedimiento para poder utilizar un PC externo como soporte de datos. En el manual de manejo se encuentran las instrucciones necesarias para el manejo de programas y ficheros (del PC al control).

Creación de un directorio de piezas con indicación de ruta ($PATH=…)

Se debe de introducir la ruta de destino en la segunda línea de un fichero con la instrucción $PATH=…. A continuación, el fichero se almacena en la ruta indicada.

Ejemplo:

Código del programa

```
%_N_EJE_MPF
;$PATH=/_N_WKS_DIR/_N_EJE_WPD
N10 G0 X… Z…
...
M2
```

El fichero _N_EJE_MPF se almacena en el directorio /_N_WKS_DIR/_N_EJE_WPD.

Creación de un directorio de piezas sin indicación de ruta

Si falta la indicación de ruta, se almacenan los ficheros con la extensión _SPF en el directorio /_N_SPF_DIR, los ficheros con la terminación _INI en la memoria de trabajo, y todos los demás ficheros en el directorio /_N_MPF_DIR.

Ejemplo:

Código del programa

```
%_N_EJE_SPF
...
M17
```

El fichero _N_EJE_SPF se almacena en el directorio /_N_SPF_DIR.

Selección del programa de pieza para el mecanizado

Es posible asociar un directorio para la ejecución en un cierto canal. Si en dicho directorio se encuentra un programa principal **con el mismo nombre** o un único programa principal (_MPF), éste queda automáticamente seleccionado para su ejecución.

Bibliografía:
/BAD/ Manual de manejo HMI Advanced; apartado "Lista de tareas" así como "Seleccionar programa para ejecución"

Rutas de búsqueda en la llamada a subprogramas

Si al realizar la llamada a un subprograma (o un fichero de inicialización) no se indica explícitamente en el programa de pieza la ruta de llamada, se busca el programa solicitado obedeciendo a una ruta de búsqueda prefijada.

Llamada de subprograma con definición absoluta de ruta

Ejemplo:

Código del programa

```
...
CALL"/_N_CST_DIR/_N_CYCLE1_SPF"
...
```

Llamada de subprograma sin definición absoluta de ruta

Generalmente se suelen realizar llamadas a subprogramas sin indicar la ruta.

Ejemplo:

Código del programa

```
...
CYCLE1
...
```

Los directorios se examinan en busca del programa llamado en el siguiente orden:

Nº	Directorio	Descripción
1	Directorio raíz actual / *nombre*	Directorio raíz de piezas o directorio estándar _N_MPF_DIR
2	Directorio raíz actual / *nombre_SPF*	
3	Directorio raíz actual / *nombre_MPF*	
4	/_N_SPF_DIR / *nombre_SPF*	Subprogramas globales
5	/_N_CUS_DIR / *nombre_SPF*	Ciclos de usuario
6	/_N_CMA_DIR / *nombre_SPF*	Ciclos del fabricante
7	/_N_CST_DIR / *nombre_SPF*	Ciclos estándar

Programación de rutas de búsqueda en la llamada de subprogramas (CALLPATH)

La ruta de búsqueda en la llamada de subprograma se puede ampliar con el comando de programa de pieza CALLPATH.

Ejemplo:

Código del programa

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

La ruta de búsqueda se guarda antes de la posición 5 (Ciclo de usuario) según la programación indicada.

Para más información sobre la ruta de búsqueda programable en llamadas de subprograma con CALLPATH, ver apartado "Ampliación de la ruta de búsqueda en llamadas de subprograma con CALLPATH".

2.2 Memoria de trabajo (CHANDATA, COMPLETE, INITIAL)

Función

La memoria de trabajo contiene los datos actuales del sistema y el usuario con los que se utiliza el control (sistema de ficheros activo), p. ej.:

- Datos de máquina activos
- Datos de corrección de herramienta
- Decalajes de origen
- ...

Programas de inicialización

Los programas de inicialización realizan la asignación previa (inicial) de datos en la memoria de trabajo. Para ello se pueden utilizar los siguientes tipos de fichero:

Tipo de fichero	Descripción
name_TEA	Dato de máquina (DM)
name_SEA	Datos del operador
name_TOA	Correcciones de herramientas
name_UFR	Decalajes de origen/frames
name_INI	Fichero de inicialización
name_GUD	Datos globales del usuario
name_RPA	Parámetros R

Encontrará información sobre todos los tipos de ficheros en el manual de manejo de la interfaz de usuario.

Área de datos

Los datos se pueden asociar a distintas áreas para las cuales sean válidos. Por ejemplo, un control puede disponer de varios canales o, habitualmente, también de varios ejes.

Existen las siguientes áreas de datos:

identificación	Áreas de datos
NCK	Datos específicos de NCK
CH<n>	Datos específicos de canal (<n> indica el número de canal)
AX<n>	Datos específicos de eje (<n> indica el número del eje de máquina)
TO	Datos de herramienta
COMPLETE	Todos los datos

Generar el programa de inicialización en un PC externo

Utilizando identificadores de área y de tipo de datos, se pueden definir las áreas que deben considerarse como una unidad al almacenar los datos.

_N_AX5_TEA_INI	Datos de máquina para el eje 5
_N_CH2_UFR_INI	Frames del canal no 2
_N_COMPLETE_TEA_INI	Todos los datos de máquina

Tras realizar la puesta en marcha del control numérico quedan definidos una serie de datos que permiten el correcto funcionamiento del control.

Forma de proceder en controles de varios canales (CHANDATA)

CHANDATA(<Número de canal>) para varios canales sólo se permite en el fichero _N_INITIAL_INI. Es el fichero de puesta en marcha con el que se inicializan todos los datos del control.

```
%_N_INITIAL_INI
CHANDATA(1)
                                    ; Asignación de ejes de máquina al canal 1:
$MC_AXCONF_MACHAX_USED[0] = 1.
$MC_AXCONF_MACHAX_USED[1] = 2.
$MC_AXCONF_MACHAX_USED[2] = 3.
CHANDATA(2)
                                    ; Asignación de ejes de máquina al canal 2:
$MC_AXCONF_MACHAX_USED[0] = 4.
$MC_AXCONF_MACHAX_USED[1] = 5.
CHANDATA(1)
                                    ; Datos de máquina axiales:
                                    ; Ventana de parada exacta basta:
$MA_STOP_LIMIT_COARSE[AX1]=0.2      ; Eje 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2      ; Eje 2
                                    ; Ventana de parada exacta fina:
$MA_STOP_LIMIT_FINE[AX1]=0.01       ; Eje 1
$MA_STOP_LIMIT_FINE[AX1]=0.01       ; Eje 2
```

⚠PRECAUCIÓN

Instrucción CHANDATA

En el programa de pieza, la instrucción `CHANDATA` puede definirse sólo para el canal en el que se ejecuta el programa CN, es decir, la instrucción puede utilizarse para evitar que los programas CN se ejecuten en un canal no previsto.

En caso de error, se interrumpe la ejecución del programa.

Nota

Los ficheros INI de listas de trabajos no reciben ninguna instrucción `CHANDATA`.

Guardar programas de inicialización (COMPLETE, INITIAL)

Los archivos de la memoria de trabajo pueden guardarse en un PC externo y leerse de nuevo desde allí.

- Los datos se guardan con `COMPLETE`.
- Con `INITIAL` se crea, a través de todas las áreas, un fichero INI (_N_INITIAL_INI).

Leer programas de inicialización

ATENCIÓN

Si se lee el fichero con el nombre "INITIAL_INI", todos los datos no alimentados en el fichero se inicializan con datos estándar. Tan sólo se exceptúan los datos de máquina. Por tanto, se alimentan **datos de operador, datos de herramienta, DO, valores GUD...** con datos estándar (normalmente "CERO").

Para leer datos de máquina individuales es adecuado, p. ej., el fichero COMPLETE_TEA_INI. En este fichero, el control sólo espera datos de máquina. De esta forma no se modifican las demás áreas de datos en este caso.

Cargar programas de inicialización

Cuando solamente se utilizan datos por un canal, se pueden utilizar programas del tipo "INI" como programas de pieza y por lo tanto se pueden seleccionar y ejecutar. Así también se pueden inicializar datos de forma controlada por programa.

2.3 Instrucción de estructuración en el editor Step (SEFORM)

Función

La instrucción de estructuración SEFORM se evalúa en el editor Step (ayuda de programación basada en editor) para generar la vista de pasos para HMI Advanced. La vista de pasos mejora la legibilidad del subprograma CN.

Sintaxis

```
SEFORM(<Nombre de sección>,<Plano>,<Botón>)
```

Descripción

SEFORM()	Llamada de función de la instrucción de estructuración con los parámetros <Nombre de sección>, <Plano> y <Botón>		
<Nombre de sección>	Identificador del paso de trabajo		
	Tipo:	STRING	
<Plano>	Índice para el plano principal o el subplano		
	Tipo:	INT	
	Valor:	0	Plano principal
		1, ..., <n>	subplano 1, ... , subplano <n>
<Botón>	Nombre del icono que se mostrará para esta sección.		
	Tipo:	STRING	

Nota

Las instrucciones SEFORM se generan en el editor Step.

La cadena de caracteres transmitida con el parámetro <Nombre de sección> se guarda de forma sincronizada con la marcha principal en la variable BTSS, de modo análogo a lo que sucede con la instrucciónMSG. La información se conserva hasta que se sobreescribe con la siguiente instrucción SEFORM. Con Reset y Fin del programa de pieza se borra el contenido.

Durante la ejecución del programa de pieza, el NCK verifica los parámetros <Plano> y <Botón>, pero no los procesa.

Bibliografía

Para más información sobre la ayuda de programación basada en editor, ver el manual del usuario de HMI Advanced

Zonas protegidas

3

3.1 Definición de las zonas protegidas (CPROTDEF, NPROTDEF)

Función

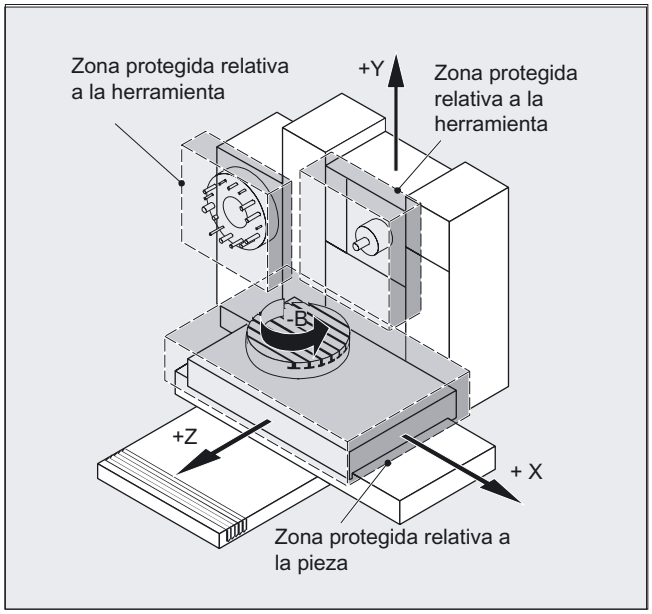
Las zonas protegidas permiten proteger distintos elementos en la máquina, el equipamiento, así como la pieza contra movimientos incorrectos.

Zonas protegidas **relativas a la herramienta**:

Para elementos pertenecientes a la herramienta (p. ej.: herramienta, portaherramientas).

Zonas protegidas **relativas a la pieza**:

Para elementos pertenecientes a la pieza (p. ej.: partes de la pieza, mesa de sujeción, garras de sujeción, mandril del cabezal, contrapunto).



Sintaxis

```
DEF INT NOT_USED
CPROTDEF(<n>,<t>,<applim>,<applus>,<appminus>)
NPROTDEF(<n>,<t>,<applim>,<applus>,<appminus>)
EXECUTE(NOT_USED)
```

Descripción

`DEF INT NOT_USED`:	Variable local, definir tipo de datos INTEGER (ver capítulo "Acciones síncronas a desplazamientos")
`CPROTDEF`:	Definir zonas protegidas específicas del canal (sólo para NCU 572/573)
`NPROTDEF`:	Definir zonas protegidas específicas de la máquina
`EXECUTE`:	Terminar definición
`<n>`:	Número de la zona protegida definida
`<t>`:	Tipo de zona protegida TRUE: Zona protegida relativa a la **herramienta** FALSE: Zona protegida relativa a la **pieza**
`<applim>`:	Tipo de limitación en la 3ª dimensión 0: Sin limitaciones 1: Limitación en dirección positiva 2: Limitación en dirección negativa 3: Limitación en dirección positiva y negativa
`<applus>`:	Valor de la limitación en dirección positiva de la 3ª dimensión
`<appminus>`:	Valor de la limitación en dirección negativa de la 3ª dimensión
`NOT_USED`:	La variable de error no surte efecto en zonas protegidas con EXECUTE

Otras informaciones

Definición de zonas protegidas

La definición de zonas protegidas comprende:

- `CPROTDEF` para zonas protegidas específicas del canal
- `NPROTDEF` para zonas protegidas específicas de la máquina
- Descripción del contorno de la zona protegida
- Fin de la definición con `EXECUTE`

Al activar la zona protegida en el programa de pieza CN puede efectuar un desplazamiento relativo del punto de referencia de la zona protegida.

Punto de referencia de la descripción del contorno

Las zonas protegidas relativas a la pieza se definen en el sistema de coordenadas básico.

Las zonas protegidas relativas a la herramienta se indican con relación al punto de referencia del portaherramientas F.

Descripción de contornos de zonas protegidas

El contorno de las zonas protegidas se indica con máx. 11 movimientos de desplazamiento en el plano seleccionado. El primer movimiento de desplazamiento corresponde al movimiento de posicionamiento en el contorno. Como zona protegida se considera entonces el área situada a la izquierda del contorno. Los movimientos de desplazamiento situados entre `CPROTDEF` o `NPROTDEF` y `EXECUTE` no se ejecutan, sino que definen la zona protegida.

Plano de trabajo

El plano deseado se selecciona antes de `CPROTDEF` o `NPROTDEF` con `G17`, `G18`, `G19` y no se debe modificar antes de `EXECUTE`. No se permite una programación de la aplicada entre `CPROTDEF` o `NPROTDEF` y `EXECUTE`.

Elementos de contorno

Se admiten:

- `G0`, `G1` para elementos de contorno rectos
- `G2` para segmentos circulares en sentido horario (únicamente para zonas protegidas relativas a la pieza)
- `G3` para segmentos circulares en sentido antihorario

Nota

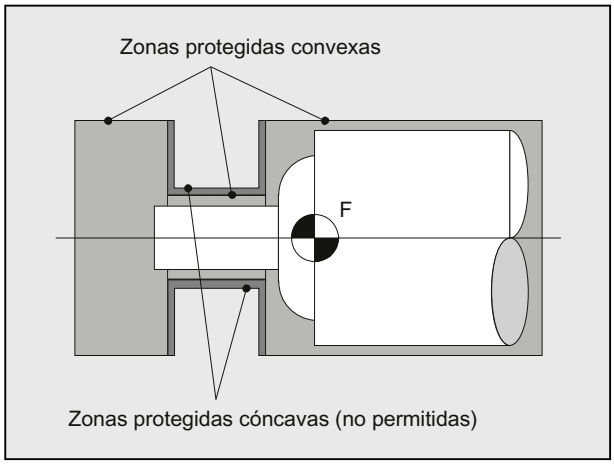
Si una circunferencia tiene que describir la zona protegida, se tiene que dividir en dos. No se admite la secuencia `G2`, `G3` o `G3`, `G2`. En su caso, se tiene que insertar una breve secuencia `G1`.

El último punto de la descripción de contorno tiene que coincidir con el primer punto.

Las zonas protegidas exteriores (solo posibles en zonas protegidas relativas a la pieza) se tienen que definir en sentido horario.

En zonas protegidas rotacionalmente simétricas (p. ej.: mandril del cabezal) se tiene que describir el contorno completo (¡no solo hasta el centro de giro!).

Las zonas protegidas relativas a la herramienta siempre tienen que ser convexas. Si se desea una zona protegida cóncava, se tiene que dividir en varias zonas protegidas convexas.



Condiciones marco

Durante la definición de las zonas protegidas no deben estar activos:

- Ninguna corrección del radio de fresa o de filo
- Ninguna transformada
- Ningún frame

Tampoco deben estar programados una búsqueda del punto de referencia (G74), un desplazamiento a punto fijo (G75), una parada de búsqueda de secuencia o un fin de programa.

3.2 Activar/desactivar zonas protegidas (CPROT, NPROT)

Función

Activar o preactivar zonas protegidas para la vigilancia de colisión o desactivar zonas protegidas activas.

El número máximo de zonas protegidas activas simultáneamente en un canal se establece a través de un dato de máquina.

Si no está activa ninguna zona protegida relativa a la pieza, la trayectoria de la herramienta se comprueba con las zonas protegidas relativas a la pieza.

Nota

Si no está activa ninguna zona protegida relativa a la pieza, no tiene lugar ninguna vigilancia de zonas protegidas.

Sintaxis

```
CPROT(<n>,<state>,<xMov>,<yMov>,<zMov>)
NPROT(<n>,<state>,<xMov>,<yMov>,<zMov>)
```

Significado

`CPROT`:	Llamada a zona protegida específica del canal (sólo para NCU 572/573)
`NPROT`:	Llamada a zona protegida específica de la máquina
`<n>`:	Número de la zona protegida
`<state>`:	Indicación del estado 0: Desactivar zona protegida 1: Preactivar zona protegida 2: Activar zona protegida 3: Preactivar zona protegida con parada condicionada
`<xMov>,<yMov>,<zMov>`:	Desplazar una zona protegida definida en los ejes geométricos

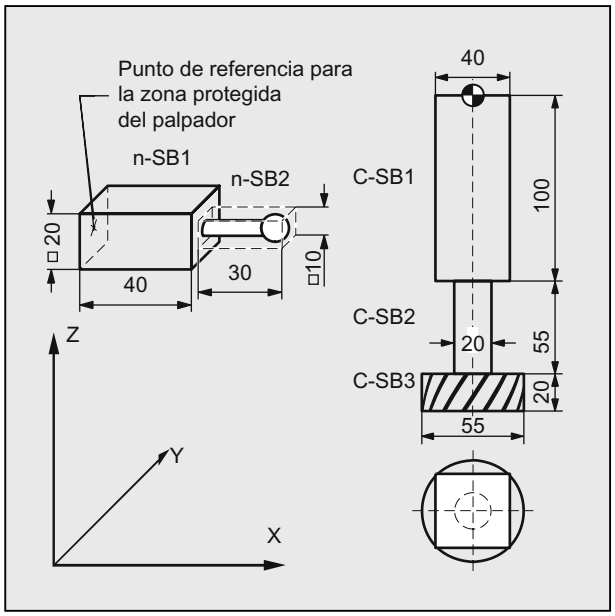
Ejemplo

Para una fresadora se quiere vigilar una posible colisión de la fresa con el palpador. La posición del palpador se indicará en la activación mediante un decalaje. Para este fin se definen las siguientes zonas protegidas:

- Una zona protegida específica de la máquina y otra específica de la pieza para el soporte del palpador (n-SB1) y para el palpador mismo (n-SB2).
- Una zona protegida específica del canal y otra específica de la herramienta para el portafresa (c-SB1), el vástago de la fresa (c-SB2) y la fresa misma (c-SB3).

La orientación de todas las zonas protegidas corresponde a la dirección Z.

La posición del punto de referencia del palpador en la activación debe ser X = -120, Y = 60 y Z = 80.



Código del programa	**Comentarios**
DEF INT SCHUTZB	; Definición de una variable auxiliar
Definición de las zonas protegidas G17	; Ajuste de la orientación
NPROTDEF(1,FALSE,3,10,–10)G01 X0 Y–10	; Zona protegida n–SB1
X40	
Y10	
X0	
Y–10	
EXECUTE(SCHUTZB)	
NPROTDEF(2,FALSE,3,5,–5)	; Zona protegida n–SB2
G01 X40 Y–5	
X70	
Y5	
X40	
Y–5	
EXECUTE(SCHUTZB)	
CPROTDEF(1,TRUE,3,0,–100)	; Zona protegida c–SB1
G01 X–20 Y–20	
X20	
Y20	
X–20	
Y–20	
EXECUTE(SCHUTZB)	
CPROTDEF(2,TRUE,3,–100,–150)	; Zona protegida c–SB2
G01 X0 Y–10	
G03 X0 Y10 J10	
X0 Y–10 J–10	
EXECUTE(SCHUTZB)	
CPROTDEF(3,TRUE,3,–150,–170)	; Zona protegida c–SB3
G01 X0 Y–27,5	
G03 X0 Y27,5 J27,5	
X0 Y27,5 J–27,5	
EXECUTE(SCHUTZB)	
Activación de las zonas protegidas:	
NPROT(1,2,–120,60,80)	; Activar zona protegida n–SB1 con decalaje
NPROT(2,2,–120,60,80)	; Activar zona protegida n–SB2 con decalaje
CPROT(1,2,0,0,0)	; Activar zona protegida c–SB1 con decalaje
CPROT(2,2,0,0,0)	; Activar zona protegida c–SB2 con decalaje
CPROT(3,2,0,0,0)	; Activar zona protegida c–SB3 con decalaje

Otras informaciones

Estado de activación (<state>)

- **<state>=2**

  Generalmente, una zona protegida se activa en el programa de pieza con el estado = 2.

  El estado siempre es específico del canal, también en zonas protegidas relativas a la máquina.

- **<state>=1**

  Si se ha previsto que una zona protegida pueda activarse mediante el programa de usuario del PLC, la preactivación necesaria al efecto tiene lugar mediante el estado = 1.

- **<state>=3**

  En la preactivación con parada condicionada, por norma la detención no se produce antes de una zona protegida preactivada vulnerada. La parada se produce únicamente cuando se ha activado la zona protegida. Esto permite un mecanizado ininterrumpido si las zonas protegidas solo se activan en casos especiales. Hay que tener en cuenta que, debido a la rampa de frenado, en ocasiones puede pasarse por una zona protegida, en caso de que dicha zona se haya activado inmediatamente antes de la pasada de prueba.

  La preactivación con parada condicionada se produce con el estado = 3.

- **<state>=0**

  La desactivación y, en consecuencia, la desconexión de las zonas protegidas tiene lugar mediante el estado = 0. Para ello no es necesario ningún desplazamiento.

Desplazamiento de zonas protegidas en la (pre-)activación

El desplazamiento se puede realizar en 1, 2 o 3 dimensiones. La indicación del desplazamiento se refiere:

- Al origen de máquina en zonas protegidas específicas de la pieza.
- Al punto de referencia del portaherramientas F en zonas protegidas específicas de la herramienta.

Estado después del arranque

Las zonas protegidas ya pueden estar activadas después del arranque y del posterior posicionamiento en el punto de referencia. Para este fin tiene que estar ajustada la variable de sistema $SN_PA_ACTIV_IMMED [<n>] o bien $SC_PA_ACTIV_IMMED[<n>] = TRUE. Se activan siempre con el estado = 2 y no tienen ningún desplazamiento.

Activación múltiple de zonas protegidas

Una zona protegida puede estar activa simultáneamente en varios canales (p. ej., pinola con dos carros opuestos). La vigilancia de las zonas protegidas sólo tiene lugar si están referenciados todos los ejes geométricos.

Se aplica:

- La zona protegida no se puede activar varias veces en un canal con decalajes distintos.
- Las zonas protegidas relativas a la máquina tienen que mostrar la misma orientación en ambos canales.

3.3 Comprobación con respecto a vulneraciones de zonas protegidas, limitación del campo de trabajo y límites de software (CALCPOSI)

Función

La función CALCPOSI sirve para comprobar si, partiendo de un punto inicial definido, los ejes geométricos pueden ejecutar un recorrido especificado sin vulnerar los límites de ejes (límites de software), las limitaciones del campo de trabajo o las zonas protegidas.

En caso de que no se pudiera ejecutar el recorrido especificado, se devuelve el máximo valor admisible.

La función CALCPOSI es un subprograma predefinido. Se tiene que encontrar por sí sola en una secuencia.

Sintaxis

```
Estado=CALCPOSI(_STARTPOS, _MOVDIST, _DLIMIT, _MAXDIST, _BASE_SYS,
_TESTLIM)
```

Descripción

`Estado`	0: Función correcta, puede ejecutarse por completo el recorrido especificado. –: En _DLIMIT existe por lo menos una componente negativa –: Se ha producido un fallo en un cálculo de transformación Si no se puede ejecutar por completo el recorrido especificado, se devuelve un valor positivo en codificación decimal: **Unidades (tipo de límite violado):** 1: El trayecto está limitado por límites de software. 2: El trayecto está limitado por limitación de la zona de trabajo. 3: El trayecto está limitado por zonas protegidas. Si se han violado varios límites al mismo tiempo (p. ej. límites de software y zonas protegidas), en las unidades se indica el límite al que se debe la restricción más estricta del trayecto especificado.

Decenas

10:

El valor inicial viola el límite

20:

La recta especificada viola el límite. Este valor se devuelve también cuando el punto final en sí no viola ningún límite, pero en el trayecto del punto inicial al punto final se produciría la violación de un valor límite (p. ej. al atravesar una zona protegida, límites de software curvados en el WKS en caso de transformadas no lineales, p. ej. Transmit).

Centenas

100:

Se ha violado el valor límite positivo (sólo si la unidad es 1 ó 2, es decir, en presencia de límites de software o límites de zona de trabajo)

100:

Se ha violado una zona protegida del NCK (sólo si la unidad es 3).

200:

Se ha violado el valor límite negativo (sólo si la unidad es 1 ó 2, es decir, en presencia de límites de software o límites de zona de trabajo)

200:

Se ha violado una zona protegida específica de canal (sólo si la unidad es 3).

Millares

1000:

Factor por el que se multiplica el número del eje que viola el límite (sólo si la unidad es 1 ó 2, en presencia de límites de software o límites de zona de trabajo).

La numeración de los ejes empieza por 1 y hace referencia, en caso de violación de límites de software (unidad = 1), a los ejes de máquina, y en caso de violación de limitación de zona de trabajo (unidad = 2), a los ejes geométricos.

1000:

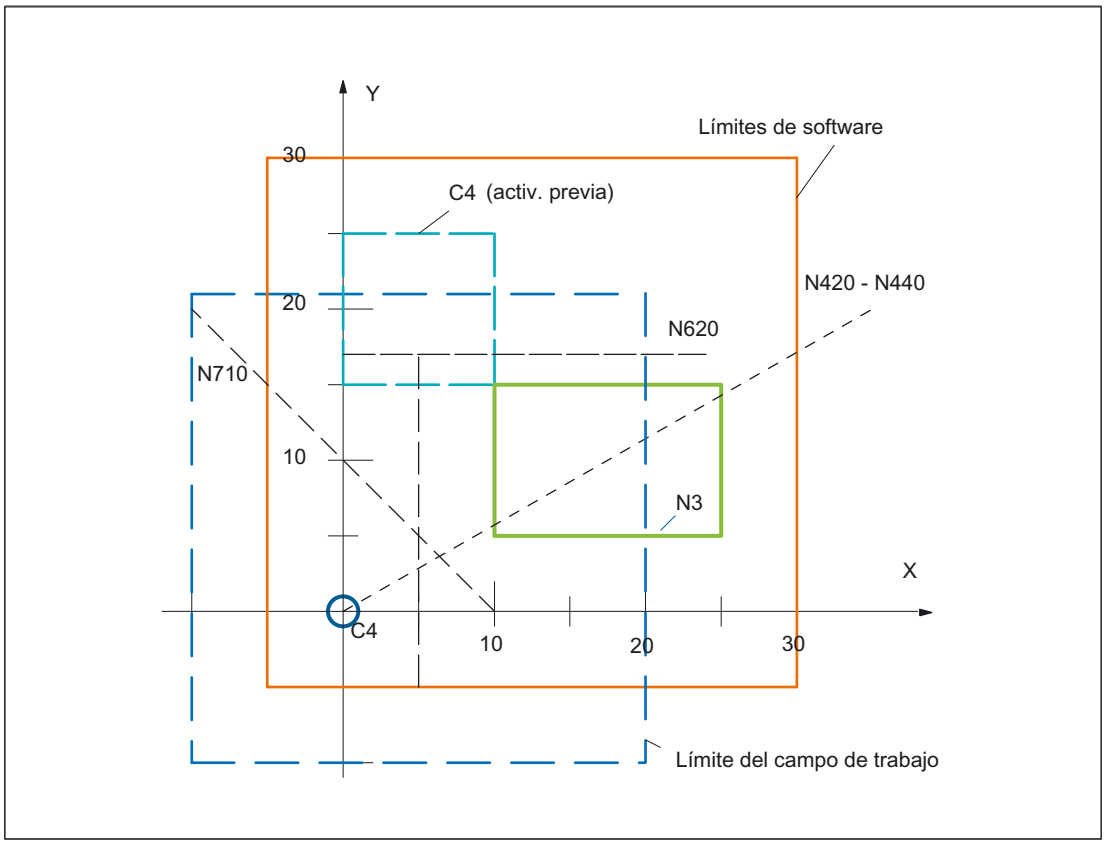
Factor por el que se multiplica el número de la zona protegida violada (sólo si la unidad es 3).

Si se han violado varias zonas protegidas, en las centenas y millares se indica la zona protegida a la que se debe la restricción más estricta del trayecto especificado.

_STARTPOS	Valor inicial de abscisas [0], ordenadas [1] y aplicadas [2] en la (WKS)
_MOVEDIST	Valor incremental de abscisas [0], ordenadas [1] y aplicadas [2]
_DLIMIT	[0] - [2]: Distancias mínimas asignadas a los ejes geométricos. [3]: Distancia mínima asignada a un eje de máquina lineal en caso de transformada no lineal, si no es posible asignar de manera unívoca un eje geométrico. [4]: Distancia mínima asignada a un eje de máquina giratorio en caso de transformada no lineal, si no es posible asignar de manera unívoca un eje geométrico. Sólo para transformadas especiales, en las que deben vigilarse límites de software.
_MAXDIST	Campo [0] - [2] para valor de retorno. Recorrido incremental en los tres ejes geométricos, sin que ningún límite de eje vulnere la distancia mínima especificada en los ejes de máquina afectados. Si el trayecto no está limitado, el contenido de ese parámetro de retorno es igual al contenido de _MOVDIST.
_BASE_SYS	FALSE o parámetro no especificado: Al valorar los datos de posición y de longitud, se evalúa el código G del grupo 13 (G70, G71, G700, G710; pulgadas/-métrico). Con G70 activo y sistema base métrico (o con G71 activo y pulgadas), las variables WKS de sistema $AA_IW[X] y $AA_MW[X] se devuelven en el sistema base, y en caso necesario deben convertirse para que la función CALCPOSI pueda utilizarlas. TRUE: Al valorar los datos de posición y longitud, se utiliza siempre el sistema base del control, independientemente del valor del código G activo del grupo 13.
_TESTLIM	Limitaciones que deben comprobarse (en codificación binaria): 1: Vigilancia de límites de software 2: Vigilancia de límites de zona de trabajo 3: Vigilancia de zonas protegidas activadas 4: Vigilancia de zonas protegidas preactivadas Combinaciones por medio de adición de valores. Ajuste estándar: 15; comprobar todos.

Ejemplo

En el ejemplo (ver figura) se representan en la X los límites de software y los límites de zona de trabajo. Se agregan además tres zonas protegidas: las dos específicas de canal, C2 y C4, y la N3 del NCK. C2 es una zona protegida circular activa, relativa a la herramienta, de 2 mm de radio. C4 es una zona protegida cuadrada, preactivada y relativa a la herramienta, de 10 mm de lado, mientras que N3 es una zona protegida rectangular activa de 10 y 15 mm de lado. En el siguiente CN se definen en primer lugar las zonas protegidas y los límites de zona de trabajo tal como se muestran en el croquis, y a continuación se llama la función CALCPOSI con distintas parametrizaciones. Los resultados de las distintas llamadas a CALCPOSI se resumen en la tabla situada al final del ejemplo.



Código del programa	**Comentarios**
N10 def real _STARTPOS[3]	
N20 def real _MOVDIST[3]	
N30 def real _DLIMIT[5]	
N40 def real _MAXDIST[3]	
N50 def int _SB	
N60 def int _STATUS	
N70 cprotdef(2, true, 0)	; Zona protegida relativa a la herramienta
N80 g17 g1 x–y0	
N90 g3 i2 x2	
N100 i–x–	
N110 execute(_SB)	
N120 cprotdef(4, false, 0)	; Zona protegida relativa a la pieza
N130 g17 g1 x0 y15	
N140 x10	
N150 y25	
N160 x0	
N170 y15	
N180 execute(_SB)	
N190 nprotdef(3, false, 0)	; Zona protegida relativa a la máquina
N200 g17 g1 x10 y5	
N210 x25	
N220 y15	
N230 x10	
N240 y5	
N250 execute(_SB)	
N260 cprot(2,2,0, 0, 0)	; Activar o preactivar zonas protegidas
N270 cprot(4,1,0, 0, 0)	
N280 nprot(3,2,0, 0, 0)	
N290 g25 XX=–YY=–	; Definir limitaciones de zona de trabajo
N300 g26 xx= 20 yy= 21	
N310 _STARTPOS[0] = 0.	
N320 _STARTPOS[1] = 0.	
N330 _STARTPOS[2] = 0.	
N340 _MOVDIST[0] = 35.	
N350 _MOVDIST[1] = 20.	
N360 _MOVDIST[2] = 0.	
N370 _DLIMIT[0] = 0.	
N380 _DLIMIT[1] = 0.	
N390 _DLIMIT[2] = 0.	
N400 _DLIMIT[3] = 0.	
N410 _DLIMIT[4] = 0.	

Código del programa	**Comentarios**
;Diversas llamadas a funciones	; Otro punto inicial
N420 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST)	
N430 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,3)	
N440 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,1)	
N450 _STARTPOS[0] = 5.	; Otro destino
N460 _STARTPOS[1] = 17.	
N470 _STARTPOS[2] = 0.	
N480 _MOVDIST[0] = 0.	
N490 _MOVDIST[1] =–.	
N500 _MOVDIST[2] = 0.	
;Diversas llamadas a funciones	
N510 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,14)	
N520 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 6)	
N530 _DLIMIT[1] = 2.	
N540 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 6)	
N550 _STARTPOS[0] = 27.	
N560 _STARTPOS[1] = 17.1	
N570 _STARTPOS[2] = 0.	
N580 _MOVDIST[0] =–.	
N590 _MOVDIST[1] = 0.	
N600 _MOVDIST[2] = 0.	
N610 _DLIMIT[3] = 2.	
N620 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 12)	
N630 _STARTPOS[0] = 0.	
N640 _STARTPOS[1] = 0.	
N650 _STARTPOS[2] = 0.	
N660 _MOVDIST[0] = 0.	
N670 _MOVDIST[1] = 30.	
N680 _MOVDIST[2] = 0.	
N690 trans x10	
N700 arot z45	
N710 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST)	
N720 M30	

Resultados de las comprobaciones del ejemplo:

Nº sec. N...	_STATUS	_MAXDIST [0] (= X)	_MAXDIST [1] (= Y)	Notas
420	3123	8.040	4.594	Se viola la zona protegida N3.
430	1122	20.000	11.429	No se vigilan zonas protegidas; se viola la limitación de zona de trabajo.
440	1121	30.000	17.143	Sólo está activa la vigilancia de los límites de software.
510	4213	0.000	0.000	El punto inicial viola la zona protegida C4
520	0000	0.000	–.000	No se vigila la zona protegida preactivada C4. Puede ejecutarse por completo el recorrido especificado.
540	2222	0.000	–.000	Debido a _DLIMIT[1]=2, el recorrido está restringido por una limitación de zona de trabajo.
620	4223	–.000	0.000	Distancia total respecto a C4: 4 mm (debido a C2 y _DLIMIT[3]). La distancia de 0,1 mm entre C2 y N3 no produce limitación del recorrido.
710	1221	0.000	21.213	Frame activo con traslación y rotación. El recorrido admisible en _MOVDIST es válido en el sistema de coordenadas desplazado y girado (WKS).

Casos especiales y otros detalles

Todos los datos de recorrido se indican siempre en medida de radio, también con un eje transversal con código G activo “DIAMON”. Si no se puede ejecutar por completo el recorrido de uno de los ejes afectados, se reducen en el valor de retorno _MAXDIST también los recorridos de los demás ejes, de modo que el punto final resultante se sitúa en la trayectoria especificada.

Se permite que, para uno o varios de los ejes afectados, no se definan límites de software o limitaciones del campo de trabajo o zonas protegidas. Todos los límites se vigilan únicamente si los ejes afectados están referenciados. Eventuales ejes giratorios afectados sólo se vigilan si no son ejes de valor módulo.

La vigilancia de los limites de software y de las limitaciones del campo de trabajo depende, igual que en el desplazamiento normal, de ajustes activos (señales de interfaz para la selección de los límites de software 1 ó 2, GWALIMON/WALIMOF, datos de operador para la activación individual de los límites del campo de trabajo y la especificación si el radio de la herramienta activa se tiene en cuenta o no en la vigilancia de las limitaciones del campo de trabajo).

En determinadas transformaciones cinemáticas (p. ej.: TRANSMIT), la posición de los ejes de máquina no se puede determinar claramente a partir de las posiciones en el sistema de coordenadas de pieza (WKS) (ambigüedad). En el desplazamiento normal, el carácter unívoco resulta generalmente del historial y de la condición de que a un movimiento continuo en el WKS tiene que corresponder un movimiento continuo de los ejes de máquina. Por esta razón, en la vigilancia de los límites de software con la ayuda de la función CALCPOSI se utiliza en estos casos la posición actual de la máquina para resolver la ambigüedad. Por esta razón, puede ser necesario programar antes de CALCPOSI un **STOPRE** para poder suministrar las posiciones válidas de los ejes de máquina a la función.

No puede asegurarse que, en un movimiento en la trayectoria definida, se cumpla en todos los puntos la distancia especificada en _DLIMIT[3] frente a las zonas protegidas. Para ello, al realizar la prolongación del punto final retornado en _MOVDIST en esta distancia no puede vulnerarse ninguna zona protegida. Sin embargo, la recta puede transcurrir a una distancia cualquiera de la zona protegida.

Nota

Los detalles de los límites del campo de trabajo se encuentran en
/PG/ Manual de programación, fundamentos,

y los detalles de los límites de software, en
/FB1/ Manual de funciones básicas, Vigilancias de ejes, Zonas protegidas (A3).

Órdenes de desplazamiento especiales 4

4.1 Desplazamiento a posiciones codificadas (CAC, CIC, CDC, CACP, CACN)

Función

Los siguientes comandos permiten desplazar ejes lineales y giratorios mediante números de posición a posiciones de eje fijas especificadas en tablas de datos de máquina. Este tipo de programación se denomina "desplazamiento a posiciones codificadas".

Sintaxis

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

Descripción

`CAC(<n>)`	Desplazar a la posición codificada desde el número de posición n
`CIC(<n>)`	Desplazar a la posición codificada n posiciones hacia delante (+) o hacia atrás (–n), partiendo del número de posición actual
`CDC(<n>)`	Desplazar a la posición codificada desde el número de posición n por el trayecto más corto (sólo para ejes giratorios)
`CACP(<n>)`	Desplazar a la posición codificada desde el número de posición n en la dirección positiva (sólo para ejes giratorios)
`CACN(<n>)`	Desplazar a la posición codificada desde el número de posición n en la dirección negativa (sólo para ejes giratorios)
`<n>`	Número de posición en la tabla de datos de máquina Margen de valores: 0, 1, … (número máx. de puestos en la tabla - 1)

Ejemplo: Desplazamiento a posiciones codificadas de un eje de posicionado

Código de programación	Comentarios
N10 FA[B]=300	; Avance para eje de posicionado B
N20 POS[B]=CAC(10)	; Desplazar a la posición codificada desde el número de posición 10
N30 POS[B]=CIC(-4)	; Desplazar a la posición codificada desde el "número de posición actual" - 4

Bibliografía

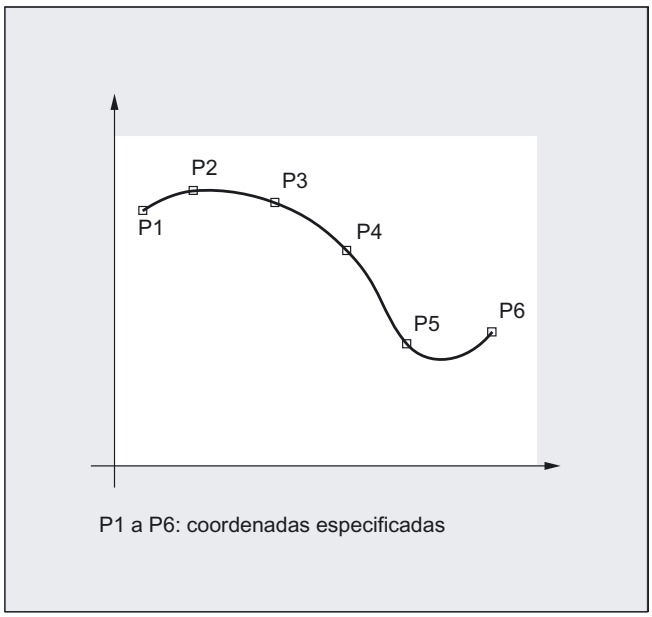
- Manual de funciones de ampliación; ejes de división (T1)
- Manual de funciones Acciones síncronas

4.2 Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Función

Los contornos de curvatura irregular de las piezas no se pueden describir analíticamente con exactitud. Por eso ese tipo de contornos se describen de modo aproximado por medio de un número limitado de puntos de interpolación, p. ej. para la digitalización de superficies. A fin de generar la superficie digitalizada en una pieza, deben asociarse los puntos de interpolación para formar una descripción de contorno. Para lograrlo se usa la interpolación spline.

Un spline define una curva compuesta por polinomios de 2º o 3er grado. Las propiedades de los puntos de interpolación de un spline pueden definirse **dependiendo del tipo de spline utilizado**.



P1 a P6: coordenadas especificadas

En SINUMERIK solution line están disponibles los siguientes tipos de splines:

- A-Spline
- B-Spline
- Spline C

Sintaxis

General:

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

Programable adicionalmente para splines B:

```
PW=<n>
SD=2
PL=<Valor>
```

Programable adicionalmente para splines A y C:

```
BAUTO/BNAT/BTAN
```

```
EAUTO/ENAT/ETAN
```

Descripción

Tipo de interpolación spline:

`ASPLINE`	Comando para activar la interpolación de splines A
`BSPLINE`	Comando para activar la interpolación de splines B
`CSPLINE`	Comando para activar la interpolación de splines C
	Los comandos `ASPLINE`, `BSPLINE` y `CSPLINE` son de tipo modal y pertenecen al grupo de los comandos de desplazamiento.

Puntos de interpolación o puntos de control:

`X... Y... Z...` `A... B... C...`	Posiciones en coordenadas cartesianas

Peso del punto (sólo para splines B):

`PW`	El comando `PW` permite programar un "peso del punto" para cada punto de interpolación.		
`<n>`	"Peso del punto"		
	Rango de valores:	0 ≤ n ≤ 3	
	Paso:	0.0001	
	Efecto:	n > 1	El punto de control atrae más a la curva.
		n < 1	El punto de control atrae menos a la curva.

Grado del spline (sólo para splines B):

`SD`	De forma estándar se utilizan polinomios de tercer grado. Sin embargo, programando `SD=2` puede utilizarse también un polinomio de 2º grado.

Distancia entre nodos (sólo para splines B):

`PL` Las distancias a los nodos se calculan adecuadamente a nivel interno. Pero el control también puede procesar distancias entre nodos especificadas, que se indican como "longitud de intervalo de parámetros" con el comando `PL`.

`<Valor>` Longitud de intervalo de parámetros

Rango de valores: como la cota

Comportamiento de transición al inicio de la curva spline (sólo para splines A o C):

`BAUTO` No se especifica ningún valor para el comportamiento de transición. El inicio se obtiene de la posición del primer punto.

`BNAT` Curvatura = 0 en puntos inicial y final

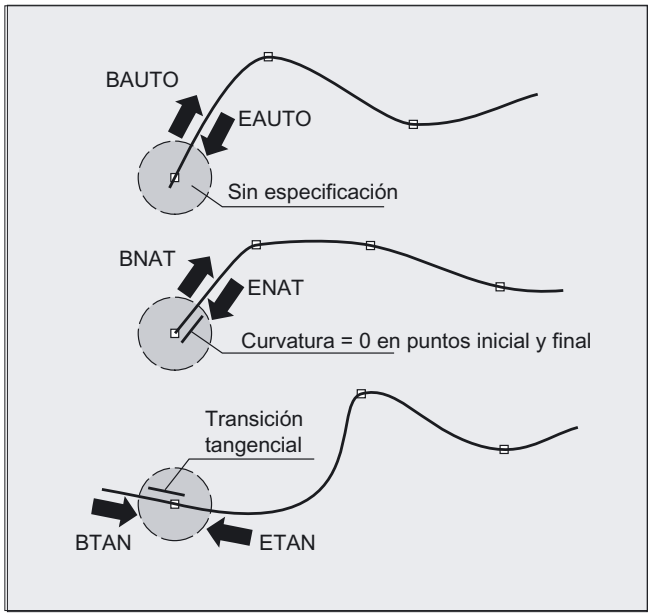
`BTAN` Transición tangencial a la secuencia anterior (posición de borrado)

Comportamiento de transición al final de la curva spline (sólo para splines A o C):

`EAUTO` No se especifica ningún valor para el comportamiento de transición. El final se obtiene de la posición del último punto.

`ENAT` Curvatura cero

`ETAN` Transición tangencial a la secuencia anterior (posición de borrado)



Nota

El comportamiento de transición programable no afecta al spline B. En los puntos inicial y final la interpolación B-spline es siempre tangencial al polígono de referencia.

Condiciones

- Puede utilizarse la corrección de radio de herramienta.
- La vigilancia de colisión se efectúa en la proyección sobre el plano.

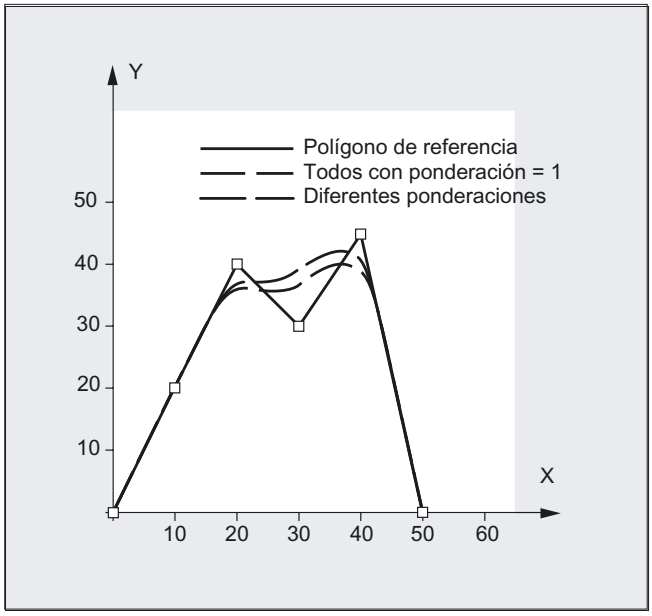
Ejemplos

Ejemplo 1: Spline B

```
Código de programa 1 (todos los pesos 1)
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

```
Código de programa 2 (diversos pesos)
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20 PW=2
N40 X20 Y40
N50 X30 Y30 PW=0.5
N60 X40 Y45
N70 X50 Y0
```

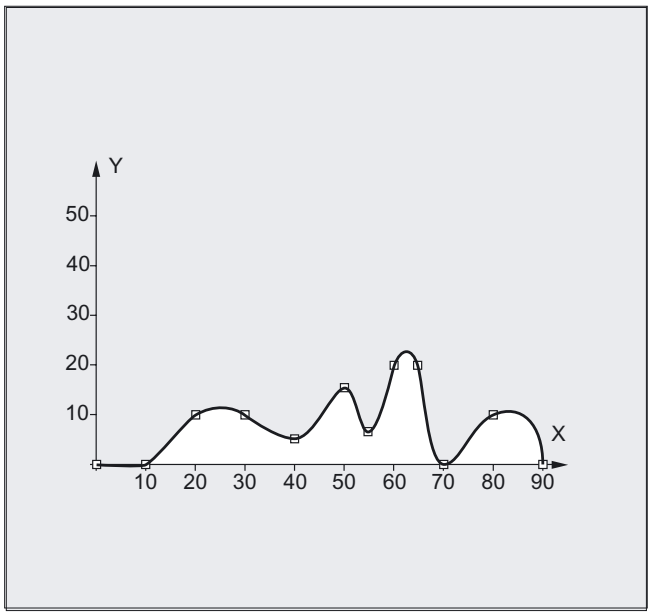
Código de programa 3 (polígono de referencia)	**Comentarios**
N10 G1 X0 Y0 F300 G64	
N20	; no procede
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	



Ejemplo 2: Spline C con radio de curvatura cero en puntos inicial y final

Código del programa

```
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
```



Ejemplo 3: Interpolación spline (spline A) y transformación de coordenadas (ROT)

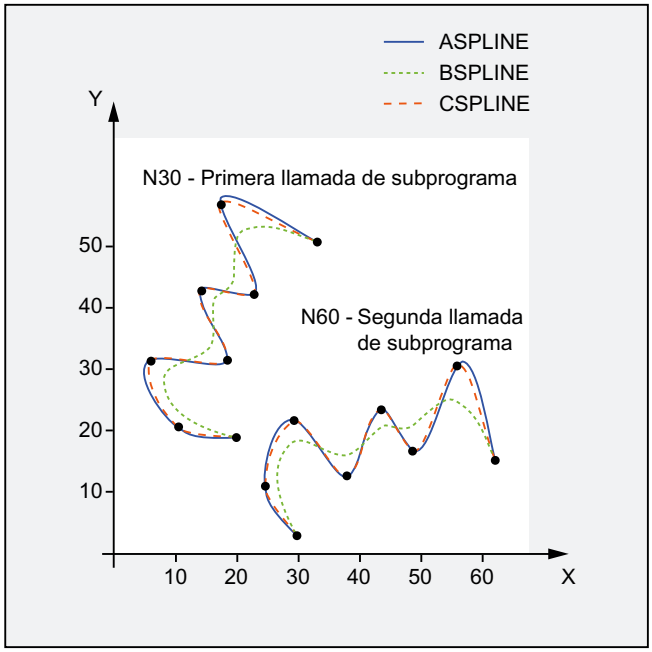
Programa principal:

Código del programa	**Comentarios**
N10 G00 X20 Y18 F300 G64	; Desplazar al punto inicial.
N20 ASPLINE	; Activar el tipo de interpolación de spline A.
N30 CONTORNO	; Primera llamada al subprograma.
N40 ROT Z-45	; Transformación de coordenadas: Giro de -45° de WKS alrededor del eje Z.
N50 G00 X20 Y18	; Desplazar al punto inicial del contorno.
N60 CONTORNO	; Segunda llamada al subprograma.
N70 M30	; Fin del programa

Subprograma "Contorno" (contiene las coordenadas de los puntos de interpolación):

Código del programa
N10 X20 Y18
N20 X10 Y21
N30 X6 Y31
N40 X18 Y31
N50 X13 Y43
N60 X22 Y42
N70 X16 Y58
N80 X33 Y51
N90 M1

En la figura siguiente se muestran, además de la curva spline resultante del ejemplo de programación (ASPLINE), las curvas spline que se habrían obtenido en caso de activar una interpolación de spline B o C (BSPLINE, CSPLINE):



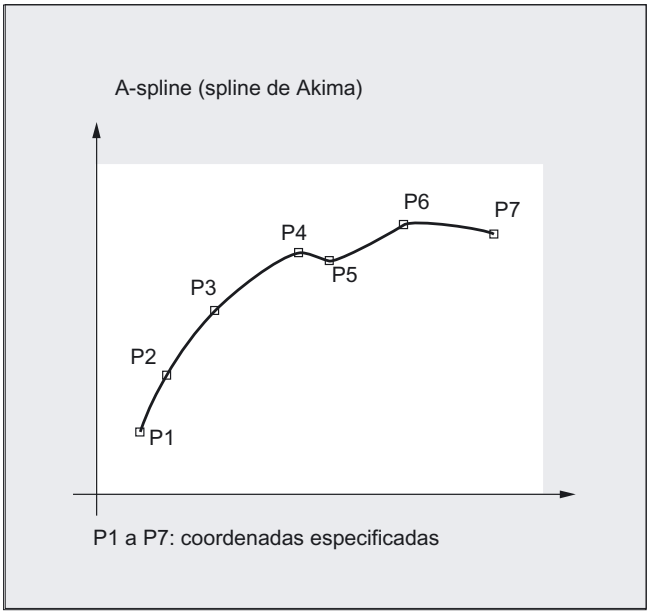
Información adicional

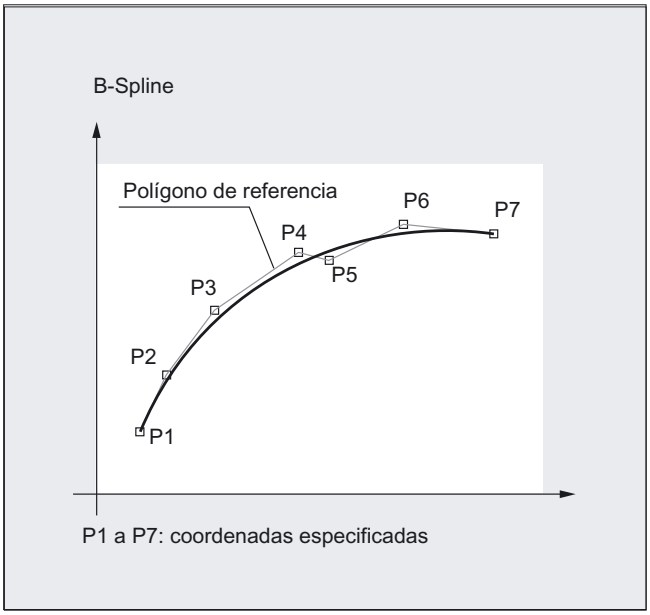
Ventajas de la interpolación spline

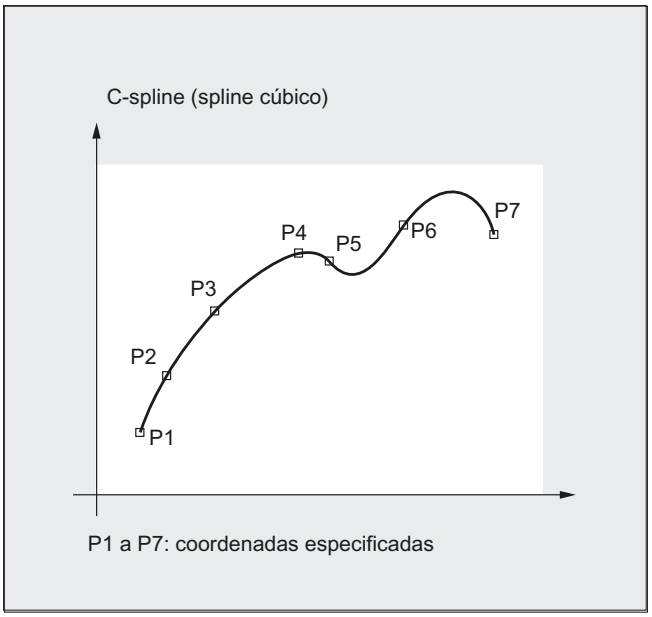
Utilizando la interpolación spline se obtienen las siguientes ventajas respecto a la utilización de secuencias de rectas `G01`:

- Menor número de secuencias de programa de pieza necesarias para la descripción del contorno
- Trazado de las curvas en la transición entre secuencias de programa de pieza suave y sin sobrecargas mecánicas

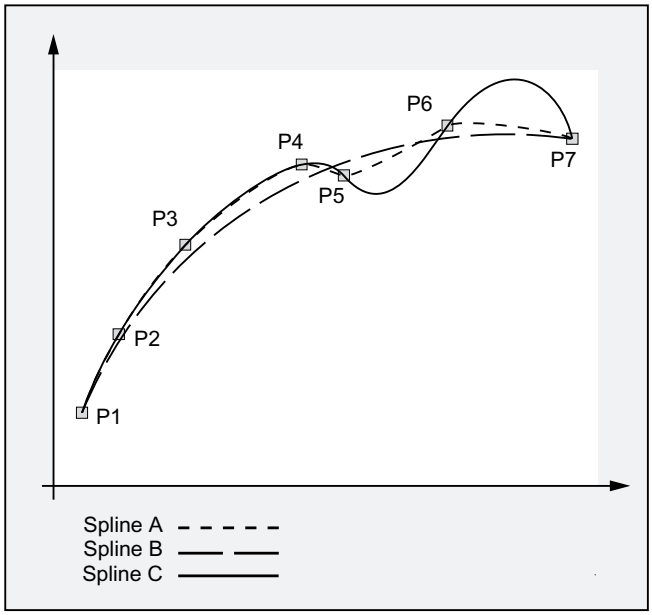
Características y aplicación de los diferentes tipos de splines

Tipo de spline	Características y aplicación
Spline A	 **Características:** • Pasa exactamente por entre los puntos de interpolación especificados. • El trazado de la curva es continuo en la tangente pero no en la curvatura. • Apenas genera oscilaciones no deseadas. • El área de influencia de las modificaciones de los puntos de interpolación es local, es decir: la modificación de uno de los puntos afecta como máximo a 6 puntos adyacentes. **Aplicación:** El spline A es idóneo para la interpolación de trazados de curva con grandes cambios de paso (p. ej. trazados curvos en forma de escalera).

Tipo de spline	Características y aplicación
Spline B	 **Características:** • No pasa por entre los puntos de interpolación especificados, sino sólo cerca de ellos. La curva es atraída por los puntos de interpolación. Otra manera de modificar el trazado de la curva es ponderando los puntos de interpolación. • El trazado de la curva es continuo en la tangente y en la curvatura. • No genera oscilaciones no deseadas. • El área de influencia de las modificaciones de los puntos de interpolación es local, es decir: la modificación de uno de los puntos afecta como máximo a 6 puntos adyacentes. **Aplicación:** La función principal del spline B es servir de interfaz con los sistemas CAD.

Tipo de spline	Características y aplicación
Spline C	 **Características:** • Pasa exactamente por entre los puntos de interpolación especificados. • El trazado de la curva es continuo en la tangente y en la curvatura. • Suele generar oscilaciones no deseadas, sobre todo en los puntos con grandes cambios de paso. • El área de influencia de las modificaciones de los puntos de interpolación es global; es decir, la modificación de un punto de interpolación afecta a todo el trazado de la curva. **Aplicación:** El spline C resulta de utilidad cuando los puntos de interpolación se encuentran en una curva previamente analizada (círculo, parábola, hipérbola)

Comparación de los tres tipos de interpolación spline para los mismos puntos predefinidos



Cantidad mínima de secuencias spline

Los códigos G `ASPLINE`, `BSPLINE B` y `CSPLINE` unen los puntos finales de la secuencia con splines. Para ello se deben de preprocesar una serie de secuencias (puntos finales) y calcularlos simultáneamente. El tamaño por defecto del búfer para el cálculo es de 10 secuencias. No toda información de secuencia es un punto final de spline. Sin embargo, el control precisa para 10 secuencias una cantidad determinada de secuencias de punto final de spline:

Tipo de spline	Cantidad mínima de secuencias spline
A-spline:	Por lo menos **4** secuencias de cada 10 deben ser secuencias spline. No cuentan secuencias con comentarios y cálculo de parámetros.
B-spline:	Por lo menos **6** secuencias de cada 10 deben ser secuencias spline. No cuentan secuencias con comentarios y cálculo de parámetros.
C-spline:	La cantidad mínima necesaria de secuencias spline resulta de la siguiente suma: valor de DM20160 $MC_CUBIC_SPLINE_BLOCKS + 1 En el DM20160 se indica la cantidad de puntos mediante la que se calcula la sección spline. El ajuste estándar es 8. Por tanto, de cada 10 secuencias, en el caso estándar, deben ser secuencias spline como mínimo **9**.

Nota

Si se rebasa por efecto el valor de tolerancia, se genera una alarma. También se emite una alarma si un eje que forma parte de la interpolación spline se programa simultáneamente como un eje de posicionado.

Resumen de secuencias spline breves

En la interpolación spline pueden aparecer secuencias spline breves que llevan a una reducción innecesaria de la velocidad de contorneado. Con la función "Resumen de secuencias spline breves" se pueden resumir estas secuencias de modo que la longitud de secuencia resultante sea suficientemente grande y no conduzca a una disminución de la velocidad de contorneado.

La función se activa mediante el dato de máquina específico del canal:

DM20488 $MC_SPLINE_MODE (ajuste de interpolación spline)

Bibliografía:
Manual de funciones básicas; Trabajo con control de contorneado, Parada precisa, Look Ahead (B1),
apartado: Resumen de secuencias spline breves

4.3 Conjunto spline (SPLINEPATH)

Función

Para seleccionar los ejes que se interpolarán en el conjunto spline se usa el comando `SPLINEPATH`. En una interpolación spline pueden intervenir hasta 8 ejes de contorneado.

Nota

Si no se programa explícitamente el comando `SPLINEPATH`, se desplazarán en forma de conjunto spline los tres primeros ejes del canal.

Sintaxis

El conjunto spline se define mediante una secuencia independiente:

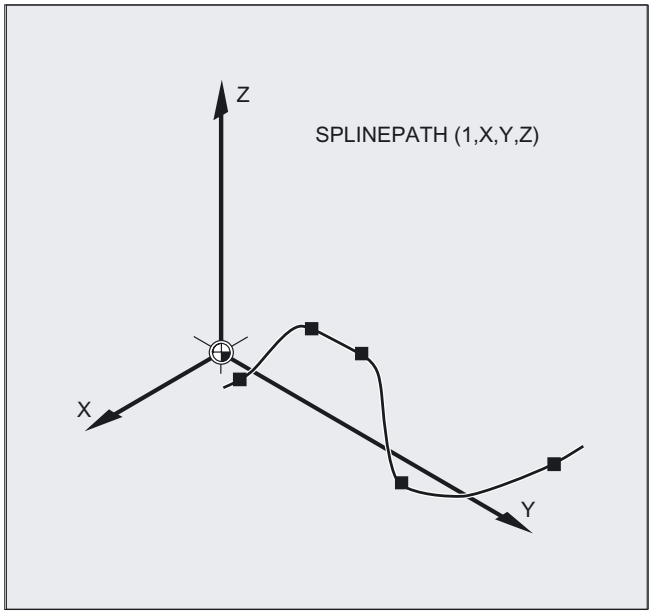
```
SPLINEPATH(n,X,Y,Z,…)
```

Descripción

`SPLINEPATH`	Comando para definir un conjunto spline
`n`	=1 (valor fijo)
`X,Y,Z,…`	Identificadores de los ejes de contorneado que se interpolarán en el conjunto spline

Ejemplo: Conjunto spline con tres ejes de contorneado

Código del programa	**Comentarios**
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH(1,X,Y,Z)	; Conjunto spline
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	; Spline C
N14 X30 Y40 Z50 A60 B70	; Puntos de interpolación
…	
N100 G1 X… Y…	; Cancelar interpolación spline



4.4 Compresión de secuencias CN (COMPON, COMPCURV, COMPCAD, COMPOF)

Función

Los sistemas CAD/CAM suelen suministrar secuencias lineales que cumplen la precisión parametrizada. Ello conlleva para contornos complejos una considerable cantidad de datos y, eventualmente, cortas secciones de trayectoria. Estas secciones de trayectoria cortas limitan la velocidad de ejecución.

Utilizando una función de compresión, se obtiene una aproximación mediante secuencias polinómicas al contorno especificado con secuencias lineales. De ello resultan las ventajas siguientes:

- Menor número de secuencias de programa de pieza necesarias para la descripción del contorno de la pieza
- Transiciones continuas entre secuencias
- Incremento de las velocidades de contorneado máximas posibles

Se ofrecen las siguientes funciones de compresor:

- COMPON

  Las transiciones entre secuencias sólo son continuas en lo que se refiere a la **velocidad**, ya que la aceleración de los ejes afectados puede realizar saltos en dichas transiciones.

- COMPCURV

  Los saltos entre transiciones se producen con **aceleración continua**. De esta manera se garantiza una evolución lisa tanto de la velocidad como también de la aceleración de todos los ejes en las transiciones entre secuencias.

- COMPCAD

  La compresión requiere mucho tiempo de cálculo y espacio en la memoria, pero está optimizada en cuanto a calidad de superficie y velocidad. Se recomienda utilizar COMPCAD sólo en caso de que el programa CAD/CAM no disponga de suficientes recursos para optimizar la superficie.

La función de compresor se finaliza con el comando `COMPOF`.

Sintaxis

```
COMPON
COMPCURV
COMPCAD
COMPOF
```

Descripción

COMPON:	Comando para activar la función de compresor COMPON. Eficacia: modal
COMPCURV:	Comando para activar la función de compresor COMPCURV. Eficacia: modal
COMPCAD:	Comando para activar la función de compresor COMPCAD. Eficacia: modal
COMPOF :	Comando para desactivar la función del compresor actualmente activa.

Nota

Para una mejora aún mayor de la calidad de superficie, pueden utilizarse las funciones de matado de esquinas G642 y limitación de tirones SOFT. Estos comandos deben escribirse al inicio del programa.

Limitaciones

- La compresión de secuencias CN sólo se utiliza para secuencias lineales (G1).
- Se comprimen sólo las secuencias que corresponden a una sintaxis sencilla:

  N... G1X... Y... Z... F... ;comentario

  Todas las demás secuencias se ejecutan sin cambio alguno (sin compresión).
- También se comprimen secuencias de desplazamiento con direcciones ampliadas como C=100 o A=AC(100).
- No es necesario programar directamente los valores de posición, ya que también pueden indicarse indirectamente por medio de parametrizaciones, como p. ej. X=R1*(R2+R3).
- Si está disponible la opción "Transformada de orientación", pueden comprimirse también secuencias CN en las que la orientación de la herramienta (y en algunos casos también el giro de la herramienta) está programada por medio de vectores de dirección (ver "Compresión de la orientación (Página 370)").
- El proceso de compresión se interrumpe cuando se ejecuta cualquier otra instrucción CN, p. ej. una emisión de funciones auxiliares.

Ejemplos

Ejemplo 1: COMPON

```
Código del programa                    Comentarios
N10 COMPON                             ; Activa la función de compresor COMPON.
N11 G1 X0.37 Y2.9 F600                 ; G1 antes de punto final y avance.
N12 X16.87 Y–.698
N13 X16.865 Y–.72
N14 X16.91 Y–.799
…
N1037 COMPOF                           ; Desactiva la función de compresor.
…
```

Ejemplo 2: COMPCAD

```
Código del programa                    Comentarios
G00 X30 Y6 Z40
G1 F10000 G642                         ; Activa la función de matado de esquinas G642.
SOFT                                   ; Activa la limitación de tirones SOFT.
COMPCAD                                ; Activa la función de compresor COMPCAD.
STOPFIFO
N24050 Z32.499
N24051 X41.365 Z32.500
N24052 X43.115 Z32.497
N24053 X43.365 Z32.477
N24054 X43.556 Z32.449
N24055 X43.818 Z32.387
N24056 X44.076 Z32.300
…
COMPOF                                 ; Desactiva la función de compresor.
G00 Z50
M30
```

Bibliografía

Manual de funciones básicas; Trabajo con control de contorneado, Parada precisa, Look Ahead (B1),
apartado: "Compresión de secuencias CN"

4.5 Interpolación polinómica (POLY, POLYPATH, PO, PL)

Función

La interpolación polinómica (POLY) no es un tipo de interpolación spline propiamente dicha. El primer lugar, está prevista como interfaz para la programación de curvas spline generadas a nivel externo. Las secciones spline se pueden programar directamente.

Este tipo de interpolación descarga al control numérico del cálculo de los coeficientes polinómicos. Esta función es especialmente apropiada cuando los coeficientes polinómicos se introducen al control numérico directamente desde un sistema CAD o bien desde un postprocesador.

Sintaxis

Polinomio de 3er grado:

```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

o ampliación a polinomios de 5º grado y nueva sintaxis polinómica:

```
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")
```

Descripción

`POLY`	Activar la interpolación polinómica con una secuencia con POLY.
`POLYPATH`	Interpolación polinómica seleccionable para los dos grupos de ejes AXIS o VECT
`PO[Identificador de eje/variable]`	Puntos finales y coeficientes de polinomio
`X, Y, Z`	Identificador de eje
`xe, ye, ze`	Indicar la posición final para el eje correspondiente; margen de valores como en cota
`a2, a3, a4, a5`	Los coeficientes a_2, a_3, a_4 y a_5 se escriben con su valor; margen de valores como en la cota. Se puede prescindir en cada caso del último coeficiente si tiene el valor cero.
`PL`	Longitud del intervalo de parámetros en el que se definen los polinomios (rango de definición de la función f(p)). El intervalo empieza siempre con 0, p puede tomar valores entre 0 y PL. Margen de valores teórico para PL: 0,0001 … 99 999,9999 **Nota:** El valor de PL es válido para la secuencia en la que se encuentra. Si no se ha programado ningún PL, entonces PL=1.

Activar/desactivar POLY

La interpolación polinómica pertenece al primer grupo G, junto con G0, G1, G2, G3, spline A, spline B y spline C. Cuando está activa no es necesario programar la sintaxis polinómica: Los ejes que están programados sólo con su nombre y punto final, se desplazan al punto final de modo lineal. Si todos los ejes están programados de este modo, el control se comporta como con G1.

La interpolación polinómica se desactiva mediante otro comando del grupo G (p. ej. G0 o G1).

Coeficiente de polinomio

El valor de `PO (PO[]=)` o `...=PO(...)` indica todos los coeficientes de polinomio correspondientes a un eje. De acuerdo con el grado de polinomio, se especifican varios valores separados por comas. Dentro de una secuencia puede haber diferentes grados de polinomio, correspondientes a diferentes ejes.

Nueva sintaxis polinómica con PO: La sintaxis anterior sigue siendo válida.

Llamada al subprograma POLYPATH

Con `POLYPATH` se puede especificar selectivamente la interpolación polinómica para los siguientes grupos de ejes:

- `POLYPATH("AXES")`

  Todos los ejes de contorneado y ejes adicionales.

- `POLYPATH("VECT")`

  Ejes de orientación (para transformada de orientación).

Por defecto los polinomios programados se interpolan también como polinomio para los dos grupos de ejes.

Ejemplos:

```
POLYPATH("VECT")
```

Para la interpolación polinómica se seleccionan sólo los ejes de orientación; todos los demás ejes se desplazan de modo lineal.

```
POLYPATH( )
```

Desactiva la interpolación polinómica para todos los ejes

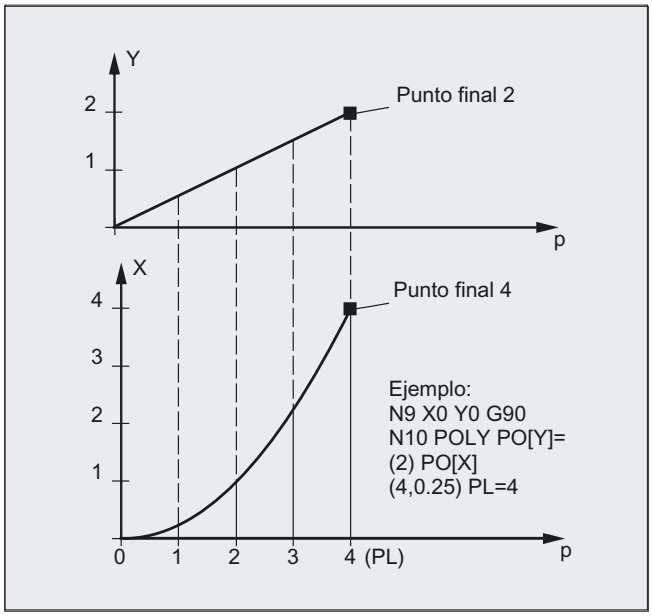
Ejemplo

```
Código del programa                                          Comentarios
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          ; Activa la interpolación polinómica
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                              ; Datos mixtos para los ejes
N27 PO[X]=(10,2.5) PO[Y]=(2.3)                               ; No hay PL programado; PL=1
N30 G1 X… Y… Z.                                              ; Desactiva la interpolación polinómica
…
```

Ejemplo: Sintaxis polinómica válida con PO

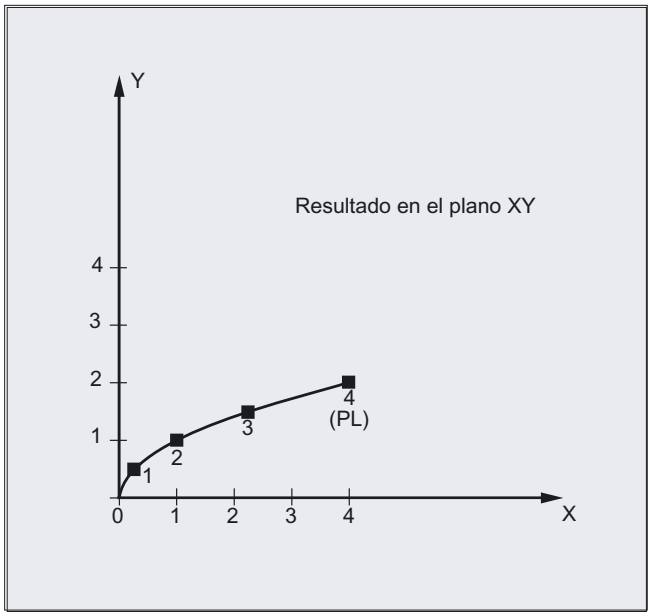
La sintaxis polinómica anterior sigue siendo válida	Nueva sintaxis polinómica (a partir de SW 6)
PO[Identificador de eje]=(.. , ..)	Identificador de eje=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[variable]=IC(.. , ..)	variable=PO IC(.. , ..)

Ejemplo: Curva en el plano X/Y



Código del programa

```
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4
```



Descripción

El control está en condiciones de ejecutar curvas (trayectorias) en las cuales cada eje de contorneado seleccionado sigue a una función polinómica de quinto grado como máximo.

La expresión general para un polinomio de tercer grado es:

$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3$

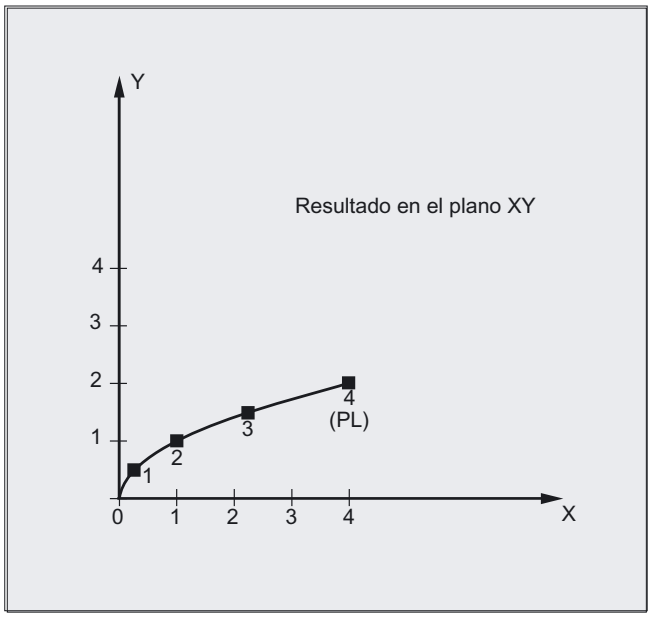
o

$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$

Significan:

a_n: coeficientes constantes

p: Parámetros



Se puede generar una gran variedad de curvas tales como funciones lineales, parabólicas, exponenciales mediante la asignación de valores determinados a dichos coeficientes.

Al definir los coeficientes $a_2 = a_3 = 0$ o bien $a_2 = a_3 = a_4 = a_5 = 0$ se obtiene, p. ej., una recta con:

$f(p) = a_0 + a_1 p$

Se debe considerar:

a_0 = posición de eje para el final de la secuencia

p = PL

$a_1 = (x_E - a_0 - a_2 * p^2 - a_3 * p^3) / p$

Es posible programar polinomios **sin** que esté activo el código G `POLY`. Pero en tal caso no se interpolan los polinomios programados, sino que se produce un desplazamiento lineal al punto final programado para cada eje (`G1`). La interpolación polinómica se activa programando `POLY`.

Además, cuando está activo el código G `POLY`, con el subprograma predefinido `POLYPATH(...)` se pueden seleccionar los ejes que se van a interpolar con polinomio.

Función especial: Denominador polinómico

`PO[ ]=(…)` permite programar un denominador polinómico común para ejes geométricos sin especificar el nombre del eje; esto quiere decir que el desplazamiento de ejes geométricos se interpola como el cociente entre dos polinomios.

De este modo se pueden mecanizar de forma exacta, p. ej., secciones cónicas (círculos, elipses, parábolas, hipérbolas).

Ejemplo:

Código del programa	**Comentarios**
`POLY G90 X10 Y0 F100`	`; Los ejes geométricos se desplazan de forma lineal a la posición X10 Y0.`
`PO[X]=(0,–) PO[Y]=(10) PO[]=(2,1)`	`; Los ejes geométricos se desplazan a lo largo del cuadrante de circunferencia a X0 Y10.`

El coeficiente constante (a_0) del denominador polinómico siempre se toma como 1; el punto final especificado es independiente de las funciones `G90`/`G91`.

El resultado del ejemplo anterior es el siguiente:

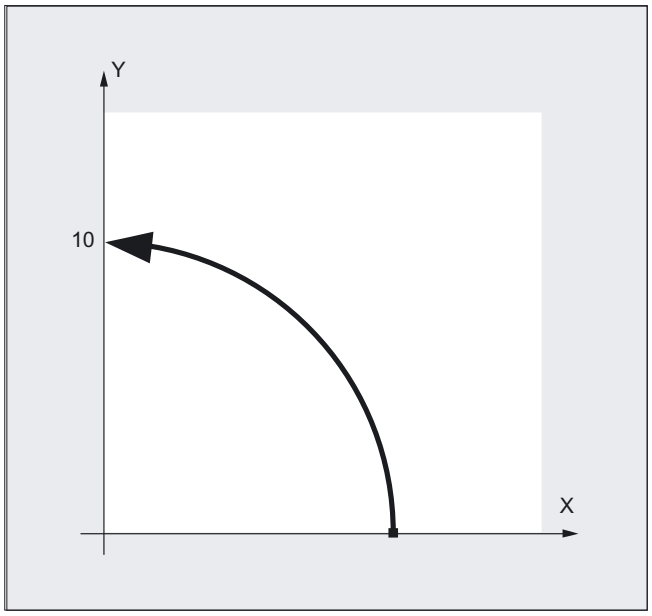
$X(p) = 10(1) / (1+p^2)$ y

$Y(p) = 20p / (1+p^2)$

con $0 \leq p \leq 1$

Los siguientes valores intermedios se generan a base del punto inicial programado, del punto final y de los coeficientes a_2 y PL=1:

Contador (X) =	$10+0*p-p^2$
Contador (Y) =	$0+20*p+0*p^2$
Denominador =	$1+2*p+1*p^2$



Cuando se tiene una interpolación polinómica activa, la programación de un denominador polinómico con ceros en el intervalo `[0,PL]` se rechaza, y se presenta una alarma. El denominador polinómico carece de efecto en desplazamientos de ejes adicionales.

Nota

La corrección del radio de herramienta también se puede activar para interpolaciones polinómicas mediante `G41` y `G42`, y se utiliza de la misma manera que para interpolaciones lineales o circulares.

4.6 Referencia de trayectoria ajustable (SPATH, UPATH)

Función

Durante la interpolación polinómica, el usuario puede seleccionar dos tipos diferentes de relaciones entre los ejes FGROUP, que determinan la velocidad, y los restantes ejes de contorneado: Estos últimos pueden desplazarse de modo síncrono a la trayectoria de los ejes FGROUP o de modo síncrono al parámetro de curva.

Por eso los ejes no pertenecientes a FGROUP pueden seguir la trayectoria de dos maneras:

- de modo síncrono al recorrido S (SPATH)

  o

- de modo síncrono al parámetro de curva U de los ejes FGROUP (UPATH)

Ambas modalidades de interpolación de trayectoria se utilizan en diversas aplicaciones y pueden conmutarse mediante los códigos G SPATH y UPATH.

Sintaxis

```
SPATH
UPATH
```

Descripción

SPATH	La trayectoria de referencia para los ejes FGROUP es la longitud de un arco
UPATH	La trayectoria de referencia para los ejes FGROUP es una curva de parámetros
FGROUP	Determinación de los ejes con avance de contorneado

SPATH, UPATH

Por medio de uno de los dos códigos G (SPATH,UPATH) se puede seleccionar y programar el comportamiento deseado.

Los comandos son de tipo modal. En caso de SPATH activo se guían los ejes de forma síncrona respecto al trayecto; en caso de UPATH activo, de forma síncrona al parámetro de curva.

UPATH y SPATH determinan también la relación del polinomio de palabra F (FPOLY, FCUB, FLIN) con el desplazamiento de trayectoria.

Activación FGROUP

La referencia a la trayectoria para los ejes no pertenecientes a FGROUP se ajusta mediante los comandos SPATH y UPATH contenidos en el grupo 45 de códigos G.

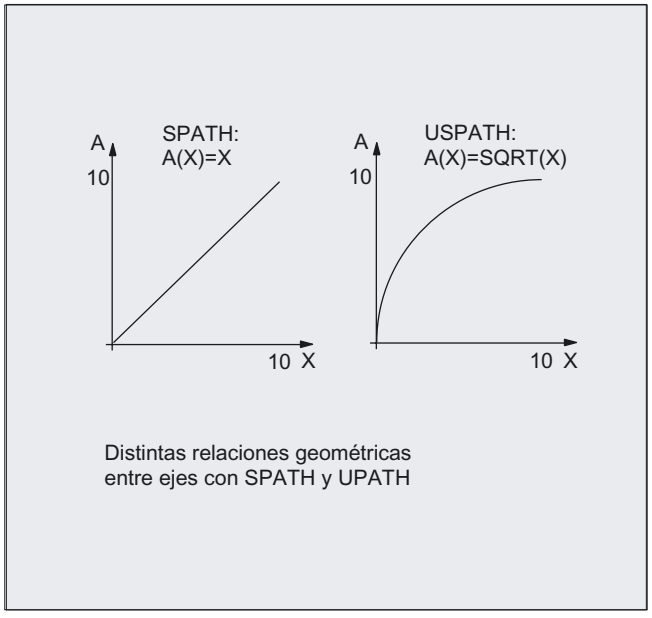
Ejemplo 1

En el siguiente ejemplo se matan las esquinas de un cuadrado de 20 mm de longitud de canto con G643. Las desviaciones máximas del contorno exacto para cada eje se definen mediante los datos de máquina DM33100 $MA_COMPRESS_POS_TOL[...].

```
Código del programa      Comentarios
N10 G1 X… Y… Z… F500
N20 G643                 ; Matado de esquinas interno a secuencia con G643
N30 XO Y0
N40 X20 Y0               ; mm de longitud de canto para los ejes
N50 X20 Y20
N60 X0 Y20
N70 X0 Y0
N100 M30
```

Ejemplo 2

El siguiente ejemplo aclara la divergencia entre las dos clases de guiado de movimiento. Se supone que en ambos casos se encuentra activo el ajuste por defecto FGROUP(X,Y,Z).



Distintas relaciones geométricas entre ejes con SPATH y UPATH

```
Programación
N10 G1 X0 A0 F1000 SPATH
N20 POLY PO[X]=(10,10) A10
```

o bien:

Programación

```
N10 G1 X0 F1000 UPATH
N20 POLY PO[X]=(10,10) A10
```

En la secuencia N20, el recorrido S de los ejes FGROUP depende del cuadrado del parámetro de curva U. Por ello a lo largo del recorrido de X se obtienen diferentes posiciones del eje síncrono A, según si está activo SPATH o UPATH.

Limitaciones

La trayectoria ajustada carece de significado en los siguientes casos

- interpolación lineal y circular,
- en secuencias de roscado y
- cuando todos los ejes de contorneado están incluidos en FGROUP

Descripción

Durante la interpolación polinómica - refiriéndose siempre a la interpolación polinómica

- en el sentido más estricto (POLY),
- todas las clases de interpolación spline (ASPLINE, BSPLINE, CSPLINE) y
- interpolación lineal con compresor (COMPON, COMPCURV),

- están predefinidas las posiciones de todos los ejes de contorneado i por polinomios pi(U). El parámetro de curva U se desplaza dentro de una secuencia CN desde 0 hasta 1, por tanto, está normalizado.

El comando en lenguaje de programación FGROUP permite seleccionar dentro de los ejes de contorneado aquellos ejes a los que se debe referir el avance de contorneado programado. No obstante, una interpolación con velocidad constante en el trayectoria S de estos ejes significa durante la interpolación polinómica una modificación no constante del parámetro de curva U.

Comportamiento del control con Reset y datos de máquina/opciones

Después del reset está activo el código G determinado por el DM 20150: GCODE_RESET_VALUES [44] (45º grupo de códigos G).

El valor de borrado para el tipo de matado de esquinas se define con DM 20150: GCODE_RESET_VALUES [9] (10º grupo de códigos G).

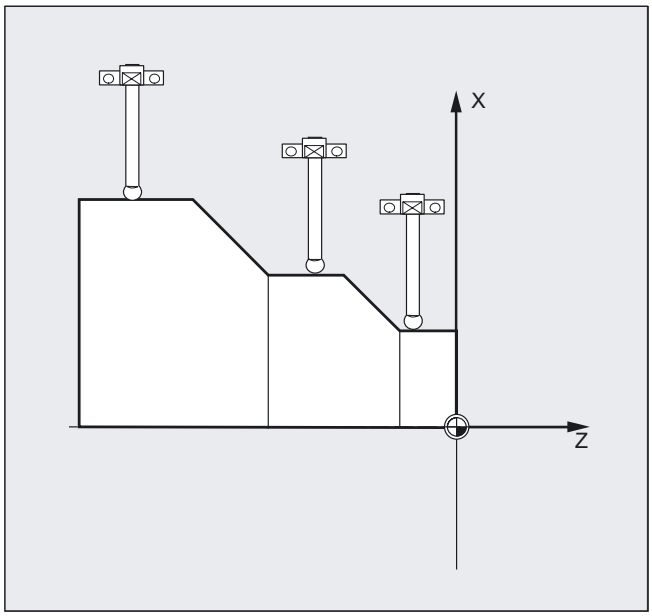
El valor activo tras reset para el grupo de códigos G se define con el dato de máquina DM 20150: GCODE_RESET_VALUES [44]. Para mantener la compatibilidad con instalaciones existentes se preajusta como valor por defecto SPATH.

Los datos de máquina específicos de eje DM 33100: COMPRESS_POS_TOL tienen un significado ampliado: contienen las tolerancias para la función de compresor y el matado de esquinas con G642.

4.7 Medida con palpador de contacto (MEAS, MEAW)

Función

Con la función "Medida con palpador de contacto" se realiza el desplazamiento a las posiciones reales en la pieza y con el flanco de conmutación del palpador se miden las posiciones para todos los ejes programados en la secuencia de medición y se escriben en la correspondiente celda de memoria para cada eje.



Programar secuencias de medición

Para la programación de la función están disponibles los dos comandos siguientes:

- `MEAS`

  Con el comando `MEAS` se borra el trayecto residual entre la posición real y la posición teórica.

- `MEAW`

  Para los procesos de medición en los que se debe alcanzar siempre la posición programada, se utiliza el comando `MEAW`.

`MEAS` y `MEAW` son válidos en una sola secuencia y se programan juntos con comandos de desplazamiento. Los avances y los tipos de interpolación (G0, G1, ...), así como la cantidad de ejes, deben estar adaptados en cada caso al problema de medición en cuestión.

Leer resultados de medición

Los resultados de la medición para los ejes capturados con palpador están disponibles en las siguientes variables:

- `$AA_MM[<Eje>]`
  Resultados de medición en el sistema de coordenadas de máquina
- `$AA_MW[<Eje>]`
  Resultados de medición en el sistema de coordenadas de pieza

Durante la lectura de estas variables no se genera internamente ninguna parada de decodificación previa.

Nota

Es necesario programar dicha parada en un lugar adecuado del programa CN utilizando `STOPRE`. De lo contrario, los valores leídos serían erróneos.

Sintaxis

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

Descripción

`MEAS`	Comando: Medición **con** borrado de trayecto residual Eficacia: secuencia a secuencia
`MEAW`	Comando: Medición **sin** borrar trayecto residual Eficacia: secuencia a secuencia
`<TE>`	Suceso de conmutación para activar la medición Tipo: INT Rango de valores: -2, -1, 1, 2 **Nota:** existen 2 palpadores como máximo (según configuración). Significado: (+)1 Flanco ascendente del palpador 1 (en entrada 1) -1 Flanco descendente del palpador 1 (en entrada 1) (+)2 Flanco ascendente del palpador 2 (en entrada 2) -2 Flanco descendente del palpador 2 (en entrada 2) **Nota:** Existen 2 palpadores como máximo (según configuración).
`G...`	Tipo de interpolación, p. ej. `G0`, `G1`, `G2` o `G3`
`X... Y... Z...`	Punto final en coordenadas cartesianas

Ejemplo

```
Código de programa                          Comentarios
N10 MEAS=1 G1 F1000 X100 Y730 Z40           ; Secuencia de medición con palpador por
                                              la primera entrada e interpolación
                                              lineal. Se realiza una parada de
                                              decodificación previa de forma
                                              automática.
...
```

Información adicional

Estado de la orden de medición

En caso de que el programa exija evaluar si el palpador ha conmutado o no, puede consultarse la variable de estado `$AC_MEA[n]` (n = número del palpador):

Valor	Descripción
0	Orden de medición no realizada
1	Orden de medición realizada correctamente (el palpador ha conmutado)

Nota

Si el palpador se ha deflectado en el programa, la variable toma el valor 1. Al iniciar una secuencia de medición, la variable toma automáticamente el valor del estado inicial del palpador.

Registro de medidas

Se registran las posiciones de todos los ejes de contorneado y de posicionado desplazados de la secuencia (número máximo de ejes según configuración del control). Con `MEAS` se frena el desplazamiento de modo definido cuando el palpador conmuta.

Nota

Si en una secuencia de medición está programado un eje geométrico, se guardan las medidas de todos los ejes geométricos actuales.

Si en una secuencia de medición está programado un eje que participa en una transformada, se guardan las medidas de todos los ejes que participan en esa transformada.

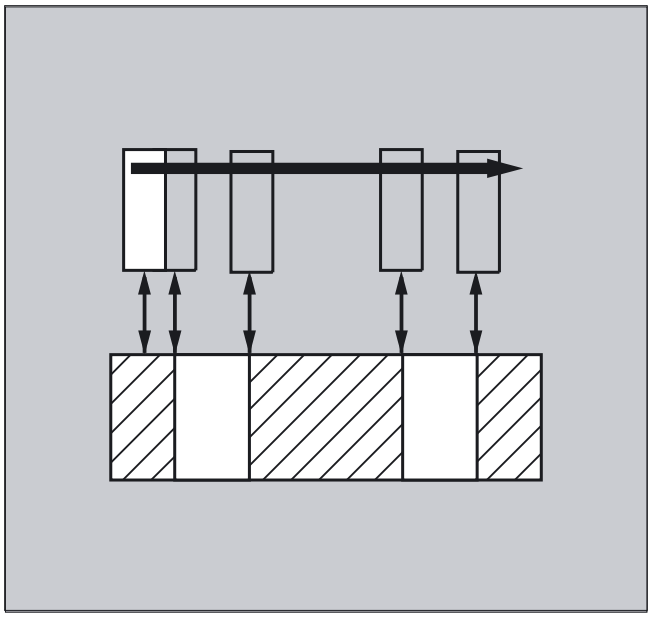
4.8 Función de medida ampliada (MEASA, MEAWA, MEAC) (opcional)

Función

En la medición axial se pueden utilizar varios palpadores y varios sistemas de medida.

Con el comando `MEASA` o `MEAWA` se capturan para cada eje programado hasta cuatro valores por medición, que se guardan en variables de sistema en función del suceso de conmutación.

Con el comando `MEAC` se pueden ejecutar órdenes de medición continuas. En este caso, los resultados de medición se almacenan en variables FIFO. `MEAC` también admite sólo un máximo de cuatro valores por medición:



Leer resultados de medición

Los resultados de medición están disponibles bajo las siguientes variables:

- $AA_MM1...4[<Eje>]

  Resultados de medición en el sistema de coordenadas de máquina
- $AA_MW1...4[<Eje>]

  Resultados de medición en el sistema de coordenadas de pieza

Sintaxis

```
MEASA[<Eje>]=(<Modo>,<TE1>,...,<TE4>)
MEAWA[<Eje>]=(<Modo>,<TE1>,...,<TE4>)
MEAC[<Eje>]=(<Modo>,<Memoria de medición>,<TE1>,...,<TE4>)
```

Nota

MEASA y MEAWA son válidos en una sola secuencia y se pueden programar juntos en una secuencia. Por el contrario, si MEASA/MEAWA se programa en una misma secuencia con MEAS/MEAW, se emite un aviso de error.

Descripción

MEASA
Comando: Medición axial con borrado de trayecto residual
Eficacia: secuencia a secuencia

MEAWA
Comando: Medición axial sin borrado de trayecto residual
Eficacia: secuencia a secuencia

MEAC
Comando: Medición axial continua sin borrado de trayecto residual
Eficacia: secuencia a secuencia

<Eje>
Nombre del deje de canal utilizado para la medición

<Modo>
Número de dos dígitos para indicar el modo de operación (modo de medición y sistema de medida)

Modo de medición (década de unidades):

0 Interrumpir la orden de medición.

1 Hasta 4 sucesos de conmutación diferentes activables simultáneamente.

2 Hasta 4 sucesos de conmutación activables sucesivamente.

3 Hasta 4 sucesos de conmutación activables sucesivamente, pero sin vigilancia del suceso de conmutación 1 en START (se suprimen las alarmas 21700/21703).

Nota:
Este modo no es posible con MEAC.

Sistema de medida (década de decenas):

0 (o sin datos) Sistema de medida activo

1 Sistema de medida 1

2 Sistema de medida 2

3 Ambos sistemas de medida

<TE>	Suceso de conmutación para activar la medición Tipo: INT Rango de valores: -2, -1, 1, 2 Significado: (+)1 Flanco ascendente del palpador 1 -1 Flanco descendente del palpador 1 (+)2 Flanco ascendente del palpador 2 -2 Flanco descendente del palpador 2
<Memoria de medición>	Número del FIFO (memoria cíclica)

Ejemplos

Ejemplo 1: Medición axial con borrado de trayecto residual en el modo 1 (evaluación en orden cronológico)

a) con 1 sistema de medida

```
Código del programa                              Comentarios
...
N100 MEASA[X]=(1,1,-1) G01 X100 F100             ; Medición en modo 1 con el sistema
                                                   de medida activo. Esperar la señal
                                                   de medición con flanco ascendente/
                                                   descendente del palpador 1 en el
                                                   recorrido hacia X=100.
N110 STOPRE                                      ; Parada de decodificación previa
N120 IF $AC_MEA[1]==FALSE GOTOF FIN              ; Comprobar si la medición se ha
                                                   realizado correctamente.
N130 R10=$AA_MM1[X]                              ; Guardar la medida correspondiente
                                                   al primer evento de disparo
                                                   programado (flanco ascendente).
N140 R11=$AA_MM2[X]                              ; Guardar la medida correspondiente
                                                   al segundo evento de disparo
                                                   programado (flanco descendente).
N150 FIN:
```

b) con 2 sistemas de medida

Código del programa	Comentarios
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; Medición en modo 1 con ambos sistemas de medida. Esperar la señal de medición con flanco ascendente/descendente del palpador 1 en el recorrido hacia X=100.
N210 STOPRE	; Parada de decodificación previa
N220 IF $AC_MEA[1]==FALSE GOTOF FIN	; Comprobar si la medición se ha realizado correctamente.
N230 R10=$AA_MM1[X]	; Guardar la medida del sistema de medida 1 con flanco ascendente.
N240 R11=$AA_MM2[X]	; Guardar la medida del sistema de medida 2 con flanco ascendente.
N250 R12=$AA_MM3[X]	; Guardar la medida del sistema de medida 1 con flanco descendente.
N260 R13=$AA_MM4[X]	; Guardar la medida del sistema de medida 2 con flanco descendente.
N270 FIN:	

Ejemplo 2: Medición axial con borrado de trayecto residual en el modo 2 (evaluación en orden programado)

```
Código del programa                              Comentarios
...
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100     ; Medición en modo 2 con el sistema de medida activo.
                                                  Esperar la señal de medición en el orden: flanco
                                                  ascendente del palpador 1; flanco descendente del
                                                  palpador 1, flanco ascendente del palpador 2, flanco
                                                  descendente del palpador 2 en el recorrido hacia
                                                  X=100.
N110 STOPRE                                   ; Parada de decodificación previa
N120 IF $AC_MEA[1]==FALSE GOTOF PALPADOR2     ; Comprobar si la medición se ha realizado correcta-
                                                  mente con el palpador 1.
N130 R10=$AA_MM1[X]                           ; Guardar la medida correspondiente al primer suceso de
                                                  conmutación programado (flanco ascendente del
                                                  palpador 1).
N140 R11=$AA_MM2[X]                           ; Guardar la medida correspondiente al segundo suceso
                                                  de conmutación programado (flanco ascendente del
                                                  palpador 1).
N150 PALPADOR2:
N160 IF $AC_MEA[2]==FALSE GOTOF FIN           ; Comprobar si la medición se ha realizado
                                                  correctamente con el palpador 2.
N170 R12=$AA_MM3[X]                           ; Guardar la medida correspondiente al tercer suceso de
                                                  conmutación programado (flanco ascendente del
                                                  palpador 2).
N180 R13=$AA_MM4[X]                           ; Guardar la medida correspondiente al cuarto suceso de
                                                  conmutación programado (flanco ascendente del
                                                  palpador 2).
N190 FIN:
```

Ejemplo 3: Medición continua axial en el modo 1 (evaluación en orden cronológico)

a) Medición de hasta 100 valores medidos

```
Código del programa                              Comentarios
...
N110 DEF REAL VALOR MEDIDO[100]
N120 DEF INT Bucle=0
N130 MEAC[X]=(1,1,-1) G01 X1000 F100             ; Medición en modo 1 con el sistema de medida activo,
                                                   guardar las medidas bajo $AC_FIFO1, esperar la señal
                                                   de medición con flanco descendente del palpador 1 en
                                                   el recorrido hacia X=1000.
N135 STPRE
N140 MEAC[X]=(0)                                 ; Interrumpir la medición una vez alcanzada la
                                                   posición del eje.
N150 R1=$AC_FIFO1[4]                             ; Guardar el número de valores medidos acumulados en
                                                   el parámetro R1.
N160 FOR Bucle=0 TO R1-1
N170 MEDIDA[Bucle]=$AC_FIFO1[0]                  ; Leer las medidas en el $AC_FIFO1 y guardarlas.
N180 ENDFOR
```

b) Medición con borrado de trayecto residual al cabo de 10 valores medidos

```
Código del programa                                          Comentarios
...
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)           ; Borrar trayecto
                                                               residual.
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500
N30 MEAC[X]=(0)
N40 R1=$AC_FIFO1[4]                                          ; Cantidad de medidas.
...
```

Información adicional

Tarea de medición

La programación de una orden de medición se puede realizar en el programa de pieza o desde una acción síncrona (ver capítulo "Acciones síncronas a desplazamientos"). Por cada eje sólo puede estar activa una orden de medición a la vez.

Nota

El avance se tiene que adaptar al problema de medición en cuestión.

Con `MEASA` y `MEAWA`, sólo se pueden garantizar resultados correctos con avances en los cuales no se produzca más de un suceso de conmutación igual ni más de 4 distintos por ciclo de regulador de posición.

En la medición continua con `MEAC`, la relación entre el ciclo de interpolación y el ciclo de regulador de posición no debe ser superior a 8:1.

Suceso de conmutación

Un suceso de conmutación se compone del número de palpador y del criterio de activación (flanco ascendente o descendente) de la señal de medición.

Para cada medición se pueden procesar hasta 4 sucesos de conmutación de los palpadores activados, es decir, hasta dos palpadores con hasta dos flancos de medición cada uno. El orden del procesamiento, así como el número máximo de sucesos de conmutación, depende del modo elegido.

Nota

Para el modo de medición 1 se aplica: Un mismo suceso de conmutación sólo debe programarse una vez en una orden de medición.

Modo de operación

Con el primer número (década de decenas) del modo de operación se selecciona el sistema de medida deseado. Si sólo existe un sistema de medida y se programa el segundo, se utiliza automáticamente el existente.

Con el segundo número (década de unidades) se selecciona el modo de medición deseado. Con ello, el proceso de medida se adapta a las posibilidades del correspondiente control:

- **Modo 1**

  La evaluación de los sucesos de conmutación tiene lugar en el orden cronológico de su aparición. En este modo, en caso de uso de módulos de seis ejes sólo se puede programar un suceso de conmutación; en caso de indicación de varios sucesos de conmutación se pasa automáticamente al modo 2 (sin mensaje).

- **Modo 2**

  La evaluación de los sucesos de conmutación tiene lugar en el orden programado.

- **Modo 3**

  La evaluación de los sucesos de conmutación tiene lugar en el orden programado, pero sin vigilancia del suceso de conmutación 1 en START.

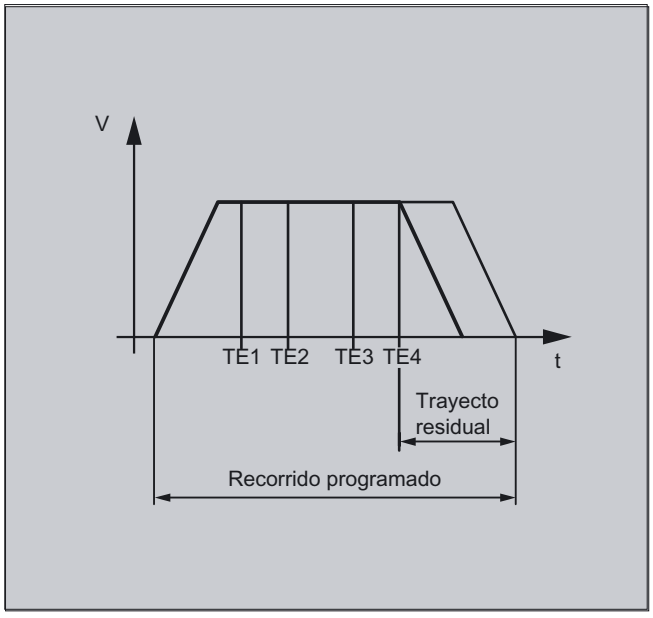
Nota

En caso de uso de 2 sistemas de medida sólo se pueden programar dos sucesos de conmutación.

Medición con y sin borrado de trayecto residual

En caso de programación de `MEASA`, el borrado de trayecto residual sólo se realiza una vez que se han capturado todos los valores medidos exigidos.

Para los procesos de medición especiales en los que siempre se debe alcanzar la posición final programada, se utiliza `MEAWA`.



Nota

MEASA no se puede programar en acciones síncronas. Alternativamente, MEAWAcon borrado de trayecto residual se puede programar como acción síncrona.

Si la orden de medición con MEAWA se inicia desde las acciones síncronas, los valores medidos sólo están disponibles en el sistema de coordenadas de máquina.

Resultados de medición para MEASA, MEAWA

Los resultados de medición están disponibles bajo las siguientes variables de sistema:

- En el sistema de coordenadas de máquina:

$AA_MM1[<Eje>]	Medida del sistema de medida programado con el suceso de conmutación 1
...	...
$AA_MM4[<Eje>]	Medida del sistema de medida programado con el suceso de conmutación 4

- En el sistema de coordenadas de pieza:

$AA_WM1[<Eje>]	Medida del sistema de medida programado con el suceso de conmutación 1
...	...
$AA_WM4[<Eje>]	Medida del sistema de medida programado con el suceso de conmutación 4

Nota

Durante la lectura de estas variables no se genera internamente ninguna parada de decodificación previa. Es necesario programar dicha parada en un lugar adecuado utilizando STOPRE. De lo contrario, los valores leídos serían erróneos.

Ejes geométricos/transformadas

Para iniciar la medición axial para un eje de geometría, la misma orden de medición se tiene que programar de forma explícita para todos los ejes geométricos restantes. Lo mismo rige para los ejes que participan en una transformada.

Ejemplo:

```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

o bien,

```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

Tarea de medición con 2 sistemas de medida

Si se ejecuta una orden de medición con dos sistemas de medida, cada uno de los dos posibles sucesos de conmutación es registrado por ambos sistemas de medida de los ejes en cuestión. De este modo queda establecida la asignación de las variables reservadas.

`$AA_MM1[<Eje>]`	o bien	`$AA_MW1[<Eje>]`	Medida del sistema de medida 1 con el suceso de conmutación 1
`$AA_MM2[<Eje>]`	o bien	`$AA_MW2[<Eje>]`	Medida del sistema de medida 2 con el suceso de conmutación 1
`$AA_MM3[<Eje>]`	o bien	`$AA_MW3[<Eje>]`	Medida del sistema de medida 1 con el suceso de conmutación 2
`$AA_MM4[<Eje>]`	o bien	`$AA_MW4[<Eje>]`	Medida del sistema de medida 2 con el suceso de conmutación 2

Estado del palpador

El estado del palpador está disponible bajo las siguientes variables de sistema:

$A_PROBE[<n>]

<n>=palpador

Valor	Descripción
1	Palpador deflectado
0	Palpador no deflectado

Estado de la orden de medición con MEASA, MEAWA

Si el programa exige una evaluación, el estado de la orden de medición se puede consultar a través de $AC_MEA[n], con n = número del palpador. En cuanto se hayan producido todos los sucesos de conmutación programados en una secuencia para el palpador <n>, esta variable suministra el valor 1. Por lo demás, el valor es 0.

Nota

Si la medición se inicia desde acciones síncronas, $AC_MEA ya no se actualiza. En este caso se tienen que consultar nuevas señales de estado PLC DB31, … DBX62.3 o la variable equivalente $AA_MEAACT[<Eje>].

Significado:

$AA_MEAACT==1: Medición activa

$AA_MEAACT==0: Medición no activa

Medición continua (MEAC)

Con MEAC, los valores medidos están disponibles en el sistema de coordenadas de maquina y se guardan en la memoria FIFO[n] indicada (memoria cíclica). Si se han configurado dos palpadores para la medición, los valores medidos del segundo palpador se guardan por separado en la memoria FIFO[n+1] configurada adicionalmente (ajustable con DM).

La memoria FIFO es una memoria cíclica en la cual se introducen valores medidos según el principio de ciclo en variables $AC_FIFO; ver capítulo "Acciones síncronas a desplazamientos".

Nota

El contenido FIFO sólo se puede leer una vez de la memoria cíclica. Para el uso múltiple de los datos de medición, éstos se tienen que guardar en los datos de usuario.

Si el número de valores medidos para la memoria FIFO sobrepasa el máximo establecido en el dato de máquina, la medición se termina automáticamente.

Una medición infinita se puede realizar mediante la lectura cíclica de valores medidos. La lectura se tiene que realizar, por lo menos, con la misma frecuencia que la entrada de nuevos valores medidos.

Detección de programaciones erróneas

Las siguientes programaciones erróneas se detectan y se señalizan con un error:

- MEASA/MEAWA programado en una misma secuencia con MEAS/MEAW

  Ejemplo:

  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```

- MEASA/MEAWA con número de parámetros <2 ó >5

  Ejemplo:

  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```

- MEASA/MEAWA con suceso de conmutación distinto a 1/ -1/ 2/ -2

  Ejemplo:

  ```
  N01 MEASA[B]=(1,1,3) B100
  ```

- MEASA/MEAWA con modo incorrecto

  Ejemplo:

  ```
  N01 MEAWA[B]=(4,1) B100
  ```

- MEASA/MEAWA con doble programación del suceso de conmutación

  Ejemplo:

  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```

- MEASA/MEAWA y falta de eje geométrico

  Ejemplo:

  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;Eje geométrico X/Y/Z
  ```

- Orden de medición no uniforme en ejes geométricos

  Ejemplo:

  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

4.9 Funciones especiales para el usuario de OEM (OEMIPO1, OEMIPO2, G810 bis G829)

Funcionamiento

Direcciones OEM

El usuario de OEM determina el significado de las direcciones OEM. La funcionalidad se aporta a través de ciclos de compilación. 5 direcciones OEM están reservadas. Los identificadores de dirección son ajustables. Las direcciones OEM se admiten en cada secuencia.

Parámetros

Grupos G reservados

Grupo 1 con `OEMIPO1`, `OEMIPO2`

El usuario de OEM puede definir dos nombres adicionales de las funciones `OEMIPO1`, `OEMIPO2`. Esta funcionalidad se aporta a través de ciclos de compilación y está reservada al usuario OEM.

- Grupo 31 con **G810 a G819**
- Grupo 32 con **G820 a G829**

Para el usuario OEM están reservados dos grupos G con 10 funciones G OEM cada uno.
Esto permite llevar las funciones introducidas por el usuario OEM para su uso al exterior.

Funciones y subprogramas

Adicionalmente, los usuarios OEM pueden crear también funciones y subprogramas predefinidos con transferencia de parámetros.

4.10 Reducción del avance con deceleración en los dos vértices (FENDNORM, G62, G621)

Función

En la deceleración automática en los dos vértices, el avance se reduce en forma de campana poco antes de la esquina en cuestión. Además, la medida del comportamiento de herramienta relevante para el mecanizado se puede parametrizar a través de datos de operador. Se trata de:

- Inicio y fin de la reducción del avance
- Corrección con la cual se reduce el avance
- Detección de la esquina relevante

Se consideran esquinas relevantes aquellas cuyo ángulo interior es menor que la esquina parametrizada a través del dato de operador.

Con el valor por defecto `FENDNORM` se desactiva la función de la corrección automática de esquinas.

Bibliografía:
/FBFA/ Descripción de funciones Dialectos ISO

Sintaxis

```
FENDNORM
G62 G41
G621
```

Descripción

`FENDNORM`	Deceleración automática en los dos vértices DES
`G62`	Deceleración en los dos vértices en esquinas interiores con la corrección del radio de corte/herramienta activa
`G621`	Deceleración en los dos vértices en todas las esquinas con la corrección del radio de herramienta activa

G62 sólo actúa en las esquinas interiores con

- corrección del radio de herramienta activa `G41`, `G42` y
- modo de contorneado activo `G64`, `G641`

El posicionado en la esquina en cuestión se realiza con el avance reducido resultante de:

F * (Corrección para la reducción del avance) * Corrección de avance

La máxima reducción de avance posible se alcanza exactamente cuando la herramienta tiene que realizar, con relación a la trayectoria del centro, el cambio de dirección en la esquina en cuestión.

G621 actúa por analogía a G62 en cada esquina de los ejes definidos por `FGROUP`.

4.11 Criterio de fin del movimiento programable (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Función

De un modo similar al criterio de cambio de secuencia con la interpolación de la trayectoria (G601, G602 y G603), el criterio de fin del movimiento en la interpolación de eje individual se puede programar en un programa de pieza o en acciones síncronas para ejes de comando/PLC.

Dependiendo del criterio de fin del movimiento que se haya establecido, se finalizarán con mayor o menor rapidez las secuencias de programa de pieza o las secuencias de ciclo tecnológico con desplazamientos del eje individual. Lo mismo se aplica con respecto al PLC a través de FC15/16/18.

Sintaxis

```
FINEA[<Eje>]
COARSEA[<Eje>]
IPOENDA[<Eje>]
IPOBRKA(<Eje>[,<Instante>])
ADISPOSA(<Eje>[,<Modo>,<Tamaño ventana>])
```

Descripción

FINEA:	Fin del movimiento al alcanzar la "Parada precisa fina"	
COARSEA:	Fin del movimiento al alcanzar la "Parada precisa basta"	
IPOENDA:	Fin del movimiento al alcanzar la "Parada interpolador"	
IPOBRKA:	Es posible el cambio de secuencia en la rampa de frenado	
ADISPOSA:	Tamaño de la ventana de tolerancia para el criterio de fin del movimiento	
<Eje>:	Nombre de eje de canal (X, Y, ...)	
<Instante>:	Instante del cambio de secuencia, referido a la rampa de frenado en %	
<Modo>:	Modo	
	Tipo: INT	
	Rango de valores:	0 Ventana de tolerancia no activa
		1 Ventana de tolerancia con relación a la posición teórica
		2 Ventana de tolerancia con relación a la posición real
<Tamaño ventana>:	Tamaño de la ventana de tolerancia	
	Este valor se introduce de forma sincronizada con la marcha principal en el dato de operador DO43610 $SA_ADISPOSA_VALUE.	
	Tipo: REAL	

Ejemplos

Ejemplo 1: Fin del movimiento al alcanzar Parada interpolador

```
Código del programa                                                    Comentarios
...
N110 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 IPOENDA[X]                    ; Desplazamiento a la posición
                                                                         X100 con una velocidad de
                                                                         contorneado de 1000 rpm con
                                                                         un valor de aceleración del
                                                                         90% y el fin del movimiento
                                                                         al alcanzar Parada
                                                                         interpolador.
...
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140 IPOENDA[X]      ; Desplazamiento a la posición
                                                                         X50 si la entrada 1 está
                                                                         activa, con una velocidad de
                                                                         contorneado de 2000 rpm y un
                                                                         valor de aceleración del
                                                                         140% y el fin del movimiento
                                                                         al alcanzar Parada
                                                                         interpolador.
...
```

Ejemplo 2: Criterio de cambio de secuencia rampa de frenado en el programa de pieza

```
Código del programa        Comentarios
                           ; Ajuste por defecto activo
N40 POS[X]=100             ; El cambio de secuencia tiene lugar cuando el eje X ha alcanzado la
                             posición 100 y la parada precisa fina.
N20 IPOBRKA(X,100)         ; Activar criterio de cambio de secuencia rampa de frenado.
N30 POS[X]=200             ; El cambio de secuencia se efectúa en cuanto el eje X empieza a frenar.
N40 POS[X]=250             ; el eje X no frena en la posición 200, sino que continúa hasta la
                             posición 250; en cuanto el eje X empieza a frenar, se produce el cambio
                             de secuencia.
N50 POS[X]=0               ; El eje X frena y vuelve a la posición 0; el cambio de secuencia se
                             produce en la posición 0 y la parada precisa fina.
N60 X10 F100
N70 M30
...
```

Ejemplo 3: Criterio de cambio de secuencia rampa de frenado en acciones síncronas

```
Código del programa          Comentarios
                             ; En el ciclo tecnológico:
FINEA                        ; Criterio de fin de movimiento Parada precisa fina.
POS[X]=100                   ; El cambio de secuencia del ciclo tecnológico tiene lugar cuando el eje X
                               ha alcanzado la posición 100 y la parada precisa fina.
IPOBRKA(X,100)               ; Activar criterio de cambio de secuencia rampa de frenado.
POS[X]=100                   ; POS[X]=100; el cambio de secuencia de ciclo tecnológico se efectúa en
                               cuanto el eje X empieza a frenar.
POS[X]=250                   ; El eje X no frena en la posición 200, sino que continúa hasta la
                               posición 250; en cuanto el eje X empieza a frenar, se produce el cambio
                               de secuencia en el ciclo tecnológico.
POS[X]=250                   ; El eje X frena y vuelve a la posición 0; el cambio de secuencia se
                               produce en la posición 0 y la parada precisa fina.
M17
```

Información adicional

Leer criterio de fin del movimiento

El criterio de fin del movimiento ajustado se puede consultar con la variable de sistema $AA_MOTEND[<Eje>]:

Valor	Descripción
1	Fin del movimiento con "Parada precisa fina"
2	Fin del movimiento con "Parada precisa basta"
3	Fin del movimiento con "Parada IPO"
4	Criterio de cambio de secuencia rampa de frenado del movimiento de eje
5	Cambio de secuencia en la rampa de frenado con ventana de tolerancia con respecto a la "posición teórica".
6	Cambio de secuencia en la rampa de frenado con ventana de tolerancia con respecto a la "posición real".

Nota

Después de RESET se conservará el último valor programado.

Criterio de cambio de secuencia en la rampa de frenado

El valor en porcentaje se introduce en el dato de operador de forma sincronizada con el proceso principal:

DO43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Si no se indica ningún valor, se activa el valor actual de dicho dato de operador.

Se puede ajustar un margen de 0% a 100%.

Ventana de tolerancia adicional para IPOBRKA

Además del criterio de cambio de secuencia existente en la rampa de frenado se puede seleccionar un criterio de cambio de secuencia adicional "Ventana de tolerancia". El desbloqueo se produce sólo cuando el eje:

- ha alcanzado, como hasta ahora, el porcentaje predefinido de su rampa de frenado

  y

- su posición real o teórica actual no está alejada en más de una tolerancia de la posición final del eje en la secuencia

Bibliografía

Para más información sobre el criterio de cambio de secuencia de los ejes de posicionado, ver:

- Manual de funciones de ampliación; Ejes de posicionado (P2)
- Manual de programación, Fundamentos; capítulo "Ajuste de avance"

4.12 Secuencia de parámetros servo programable (SCPARA)

Función

Con SCPARA la secuencia de parámetros (compuesta de DM) se puede programar en el programa de pieza y en acciones síncronas (hasta ahora, sólo a través de PLC).

DB3n DBB9 Bit3

Para evitar que se produzcan conflictos entre PLC y NCK, se define un bit adicional en la interfaz PLC –> NCK:

DB3n DBB9 Bit3 ”Predefinición de secuencia de parámetros por SCPARA bloqueada".

Si está bloqueada la predefinición de secuencias de parámetros para SCPARA no se produce ningún aviso de error si se programa a pesar de todo.

Sintaxis

```
SCPARA[<Eje>]=<Valor>
```

Descripción

SCPARA	Definir secuencia de parámetros
<Eje>	Nombre de eje de canal (X, Y, ...)
<Valor>	Secuencia de parámetros deseada (1<= valor <=6)

Nota

La secuencia de parámetros actual se puede consultar con la variable de sistema $AA_SCPAR[<Eje>].

Con G33, G331 o G332, el control elige la secuencia de parámetros más adecuada.

Si se debe **cambiar** la **secuencia de parámetros servo** en un programa de pieza o en una acción síncrona y en el PLC, debe ampliarse el programa de usuario de PLC.

Bibliografía:
/FB1/ Manual de funciones básicas; Avances (V1),
apartado "Influencia del avance".

Ejemplo

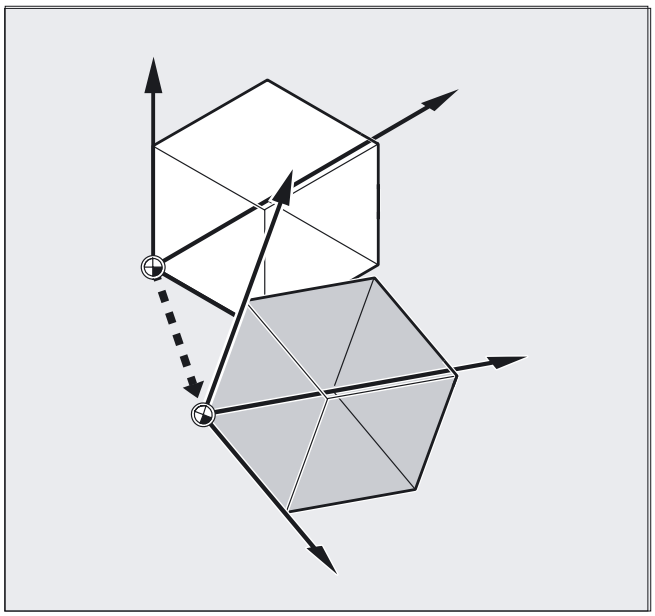
```
Código del programa    Comentarios
...
N110 SCPARA[X]= 3      ; Se selecciona la 3ª secuencia de parámetros para el eje X.
...
```

Transformadas de coordenadas (FRAMES) 5

5.1 Transformada de coordenadas a través de variables frame

Función

Además de las posibilidades de programación descritas en el manual de programación "Fundamentos", es posible estipular también los sistemas de coordenadas a través de variables frame predefinidas.



Se han definido los siguientes sistemas de coordenadas:

MKS: Sistema de coordenadas de máquina

BKS: Sistema de coordenadas básico

BNS: Sistema de coordenadas de origen básico

ENS: Sistema de coordenadas de origen ajustable

WKS: Sistema de coordenadas de pieza

¿Qué es una variable frame predefinida?

Las variables frame predefinidas son palabras reservadas que ya están determinadas con su correspondiente efecto en el uso del lenguaje del control y se pueden procesar en el programa CN.

Posibles variables frame:

- Frames básicos (decalaje básico)
- Frames ajustables
- Frame programable

Leer asignaciones de valores y valores reales

Interrelación variable frame/frame

Una transformada de coordenadas se puede activar asignando valores frame a una variable frame.

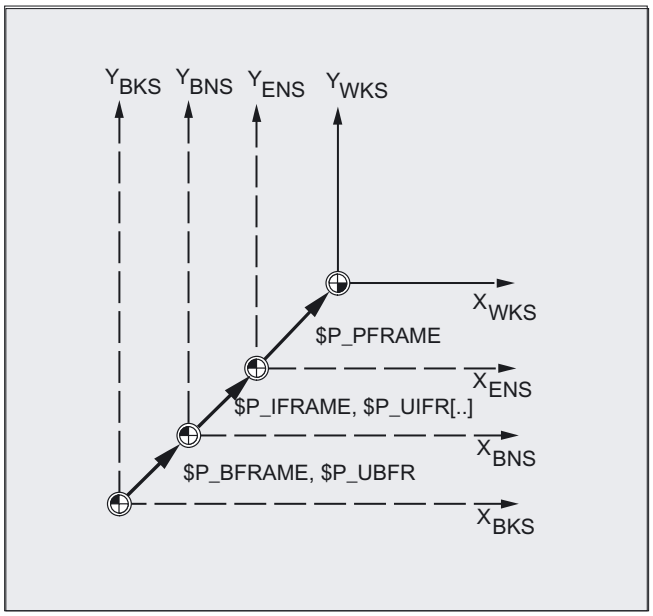
Ejemplo: `$P_PFRAME=CTRANS(X,10)`

Variable frame:

`$P_PFRAME` significa: frame programable actual.

Frame:

`CTRANS(X,10)` significa: decalaje de origen programable del eje X en 10 mm.



Lectura de valores reales

Mediante una variable predefinida en el programa de pieza se pueden leer los valores reales actuales de los sistemas de coordenadas:

$AA_IM[eje]: Lectura del valor real en el MKS

$AA_IB[eje]: Lectura del valor real en el BKS

$AA_IBN[eje:] Lectura del valor real en el BNS

$AA_IEN[eje]: Lectura del valor real en el ENS

$AA_IW[eje]: Lectura del valor real en el WKS

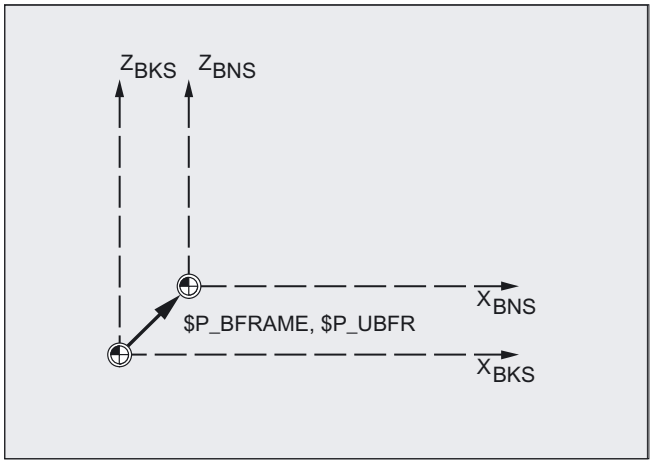
5.1.1 Variables frame predefinidas ($P_BFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)

$P_BFRAME

Variable frame básico actual, que establece la relación entre el sistema de coordenadas básico (BKS) y el sistema de origen básico (BNS).

Si el frame básico descrito mediante $P_UBFR debe actuar inmediatamente en el programa, se debe:

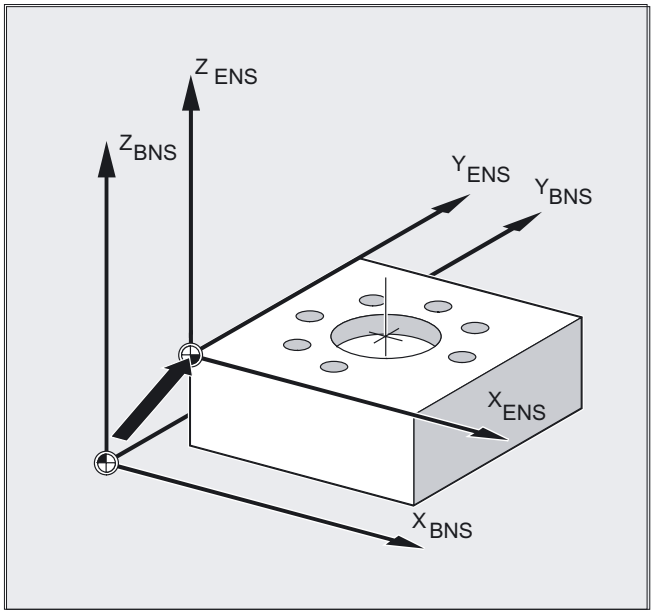
- Programar un `G500`, `G54...G599` o
- Escribir `$P_UBFR` en `$P_BFRAME`.



$P_IFRAME

Variable frame ajustable actual que establece la relación entre el sistema de origen básico (BNS) y el sistema de origen ajustable (ENS).

- `$P_IFRAME` corresponde a `$P_UIFR[$P_IFRNUM]`
- `$P_IFRAME` contiene después de la programación (p. ej., de G54) la traslación, rotación, escala y simetría especular definida por `G54`.

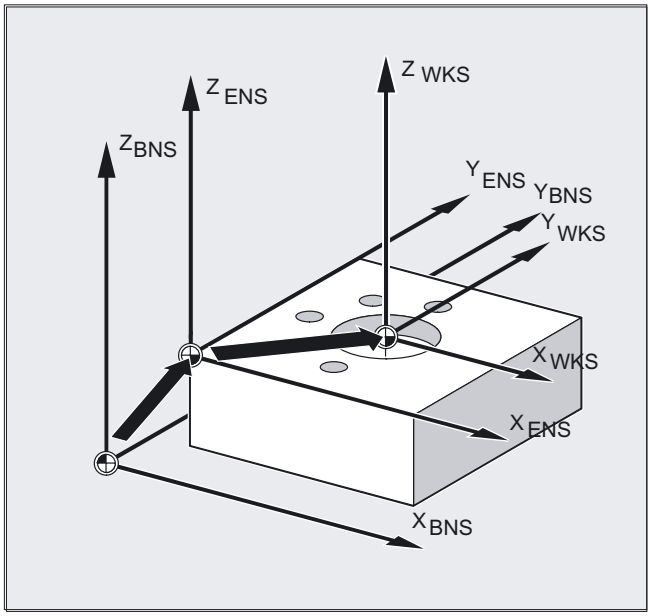


$P_PFRAME

Variable frame programable actual que establece la relación entre el sistema de origen ajustable (ENS) y el sistema de coordenadas de pieza (WKS).

`$P_PFRAME` contiene el frame resultante

- **de la programación** de `TRANS/ATRANS`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR` o
- **de la asignación** de `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` al FRAME programable



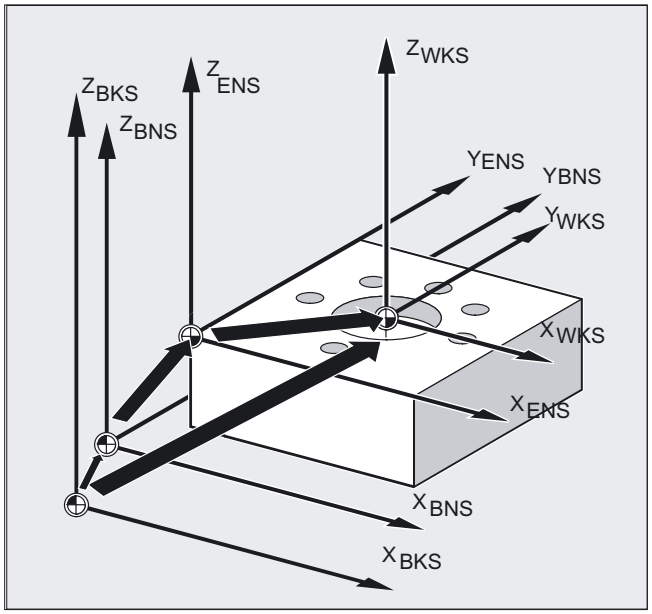
$P_ACTFRAME

Frame total actual resultante de la concatenación

- de la variable de frame básico actual `$P_BFRAME`,
- la variable de frame ajustable actual `$P_IFRAME` con frames de sistema y
- la variable de frame programable actual `$P_IFRAME` con frames de sistema-

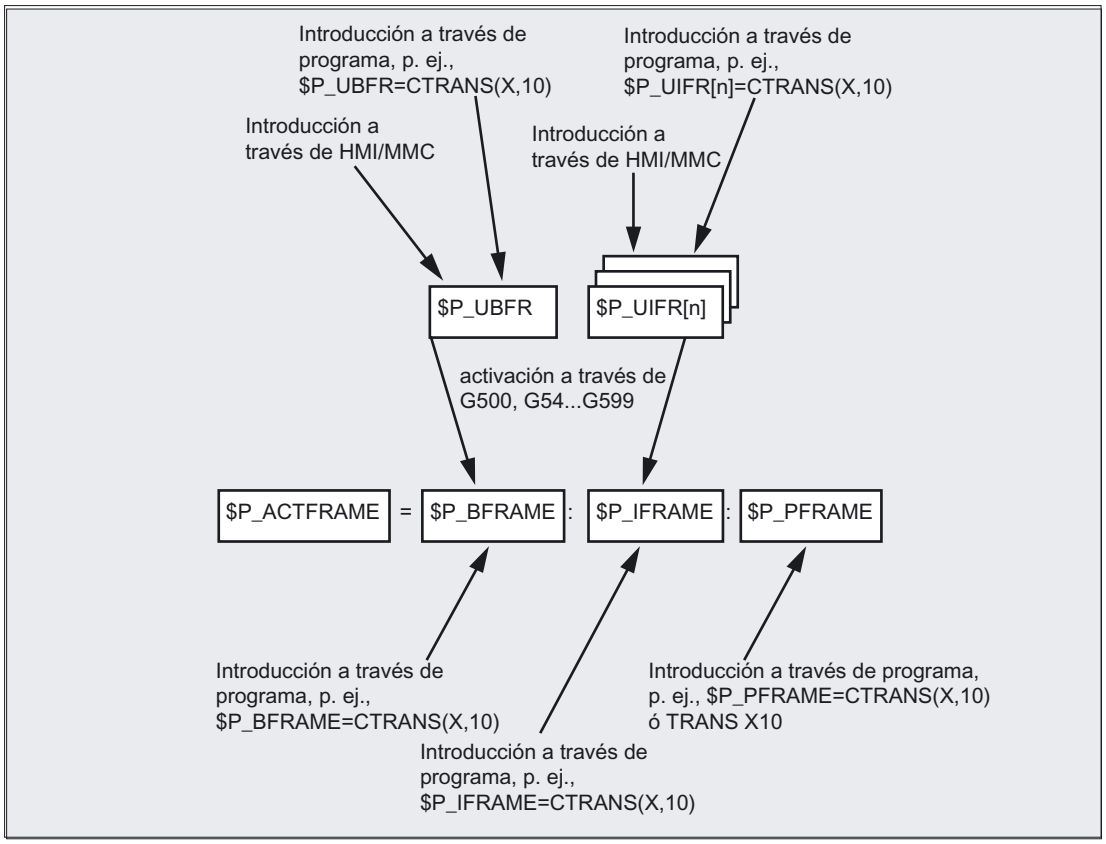
Frames de sistema, ver capítulo "Frames activos en el canal"

`$P_ACTFRAME` describe el origen de pieza actualmente válido.



Si se modifican `$P_BFRAME`, `$P_IFRAME` o `$P_PFRAME` , `$P_ACTFRAME` se vuelve a calcular.

`$P_ACTFRAME` se corresponde con `$P_BFRAME`:`$P_IFRAME`:`$P_PFRAME`



El frame básico y el frame ajustable actúan tras el Reset, cuando el DM 20110 RESET_MODE_MASK ha sido ajustado de la siguiente manera:

Bit0=1, bit14=1 --> `$P_UBFR` (frame básico) activo

Bit0=1, bit5=1 --> `$P_UIFR[$P_UIFRNUM]` (frame ajust.) activo

Frames ajustables predefinidos $P_UBFR[n]

Con $P_UBFR se programa el frame básico, el cual no es simultáneamente activo en el programa de pieza. El frame básico escrito con $P_UBFR se tiene en cuenta, cuando:

- Se han activado Reset y los bits 0 y 14 del DM RESET_MODE_MASK
- Se han ejecutado las instrucciones `G500, G54...G599`

Frames ajustables predefinidos $P_UIFR[n]

Mediante la variable frame predefinida $P_UIFR[n] se pueden leer o escribir los decalajes de origen ajustables `G54` a `G599` desde el programa de pieza.

Estas variables representan en la estructura una matriz unidimensional del tipo FRAME con el nombre `$P_UIFR[n]` .

Asignación a los comandos G

De forma estándar se dispone de 5 frames ajustables `$P_UIFR[0]... $P_UIFR[4]` o de 5 comandos G equivalentes –`G500` y `G54` a `G57`– bajo cuyas direcciones pueden almacenarse valores.

`$P_IFRAME=$P_UIFR[0]` se corresponde con `G500`

`$P_IFRAME=$P_UIFR[1]` se corresponde con `G54`

`$P_IFRAME=$P_UIFR[2]` se corresponde con `G55`

`$P_IFRAME=$P_UIFR[3]` se corresponde con `G56`

`$P_IFRAME=$P_UIFR[4]` se corresponde con `G57`

Con datos de máquina se puede modificar la cantidad de frames:

`$P_IFRAME=$P_UIFR[5]` se corresponde con `G505`

`... ... ...`

`$P_IFRAME=$P_UIFR[99]` se corresponde con `G599`

Nota

Así se puede generar un total de 100 sistemas de coordenadas que se pueden llamar en los diferentes programas, p. ej., como origen para dispositivos distintos.

⚠ PRECAUCIÓN

La programación de variables frame y de frames requiere una secuencia CN propia en el programa CN. **Excepción:** Programación de un frame ajustable con `G54`, `G55`, `...`

5.2 Asignar valores a variables frame/frames

5.2.1 Asignar valores directos (valor de eje, ángulo, escala)

Función

Se pueden asignar valores directamente en el programa CN a frames o variables frame.

Sintaxis

```
$P_PFRAME=CTRANS (X, Valor eje, Y, Valor eje, Z, Valor eje, …)
$P_PFRAME=CROT (X, Ángulo, Y, Ángulo, Z, Ángulo, …)
$P_UIFR[..]=CROT (X, ángulo, Y, ángulo, Z, ángulo, …)
$P_PFRAME=CSCALE (X, Escala, Y, Escala, Z, Escala, …)
$P_PFRAME=CMIRROR (X, Y, Z)
```

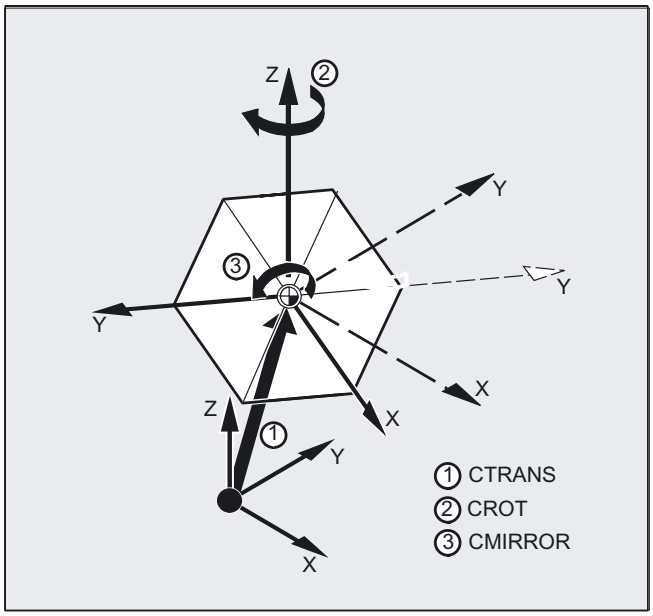
La programación de $P_BFRAME se realiza de forma análoga a $P_PFRAME.

Descripción

`CTRANS`	Desplazamiento en los ejes indicados
`CROT`	Rotación alrededor del eje indicado
`CSCALE`	Cambio de escala en los ejes indicados
`CMIRROR`	Inversión de sentido de los ejes indicados
`X Y Z`	Valores de decalaje a lo largo de los ejes geométricos indicados
`Valor de eje`	Asignar valor de eje al decalaje
`Ángulo`	Asignar ángulo de giro alrededor de los ejes indicados
`Escala`	Modificar escala

Ejemplo

Mediante la asignación de valores al frame programable actual se activan la translación, rotación y simetría (imagen especular).




```
N10 $P_PFRAME=CTRANS(X,10,Y,20,Z,5):CROT(Z,45):CMIRROR(Y)
```

Preasignación de los componentes de rotación de frames con otros valores

Con CROT, preasignar valores a los tres componentes de UIFR

Código del programa	**Comentarios**
`$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)`	
`N100 $P_UIFR[5, y, rt]=0`	
`N100 $P_UIFR[5, x, rt]=0`	
`N100 $P_UIFR[5, z, rt]=0`	

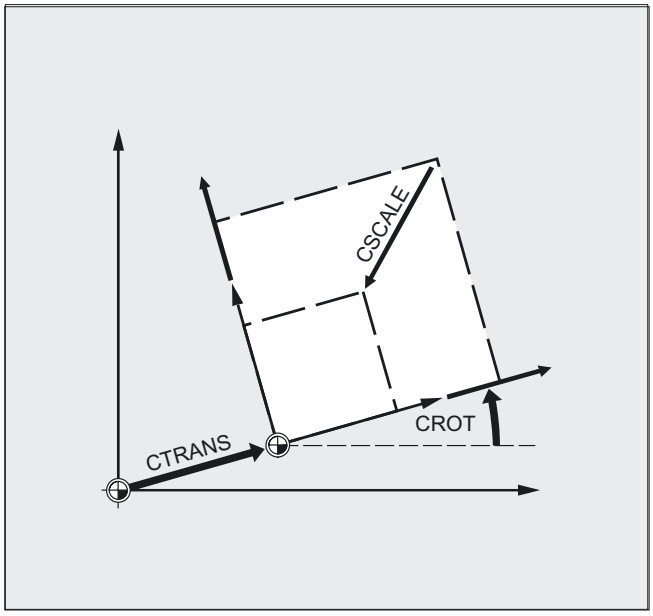
Descripción

Se pueden programar consecutivamente varias operaciones de cálculo.

Ejemplo:

```
$P_PFRAME=CTRANS(...):CROT(...):CSCALE...
```

Tenga en cuenta que los comandos se tienen que conectar con el operador de concatenación "dos puntos" (…):(…). Con ello, en primer lugar, se enlazan entre sí los comandos y, en segundo lugar, se ejecutan aditivamente en el orden programado.



Nota

Los valores programados con los comandos indicados se asignan a los frames y se memorizan.

Los valores sólo se activan cuando se asignan al frame de una variable frame activa `$P_BFRAME` o `$P_PFRAME`.

5.2.2 Leer y modificar componentes de frame (TR, FI, RT, SC, MI)

Función

Tiene la posibilidad de acceder a datos **individuales** de un frame, p. ej., a un determinado valor de decalaje o ángulo de giro. Estos valores se pueden modificar o asignar a otra variable.

Sintaxis

`R10=$P_UIFR[$P_UIFNUM,X,RT]`	El ángulo de giro RT alrededor del eje X desde el decalaje de origen ajustable actualmente válido $P_UIFRNUM se debe asignar a la variable R10.
`R12=$P_UIFR[25,Z,TR]`	El valor de decalaje TR en Z del registro de datos del frame ajustado núm. 25 se debe asignar a la variable R12.
`R15=$P_PFRAME[Y,TR]`	El valor de decalaje TR en Y del frame programable actual se debe asignar a la variable R15.
`$P_PFRAME[X,TR]=25`	El valor de decalaje TR en X del frame programable actual debe modificarse. A partir de ahora es X25.

Descripción

`$P_UIFRNUM`	`Con esta variable se establece automáticamente la relación con el decalaje de origen ajustable actualmente válido.`
`P_UIFR[n,…,…]`	`Especificando el número de frame n se accede al frame ajustable número n.`
	`Especificación del componente que se desea leer o modificar:`
`TR`	`TR Translation`
`FI`	`FI Translation Fine`
`RT`	`RT Rotation`
`SC`	`SC Scale cambio de escala`
`MI`	`MI Simetría especular`
`X Y Z`	`Además (ver ejemplos) se indica el eje correspondiente X, Y, Z.`

Rango de valores para rotación RT

Giro alrededor del 1er eje geométrico:	-180° a +180°
Giro alrededor del 2º eje geométrico:	-90° a +90°
Giro alrededor del 2º eje geométrico:	-180° a +180°

Descripción

Llamar un frame

Indicando la variable de sistema $P_UIFRNUM se puede acceder directamente al decalaje de origen ajustado actualmente con $P_UIFR o G54, G55, …
($P_UIFRNUM contiene el número del frame actualmente ajustado).

Todos los demás frames ajustables almacenados $P_UIFR se llaman indicando el correspondiente número $P_UIFR[n].

Para variables frame predefinidas y frames de definición propia se indica el nombre, p. ej., $P_IFRAME.

Llamar datos

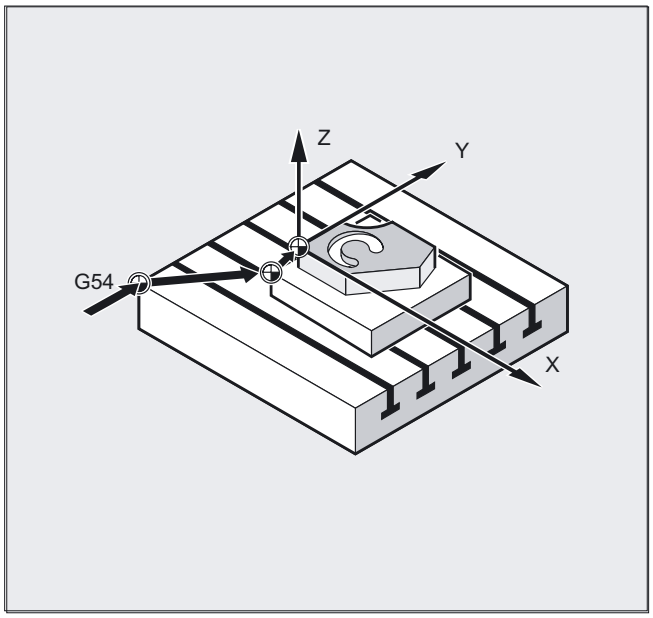
Entre corchetes se encuentran el nombre del eje y el componente de frame del valor al cual quiere acceder o que quiere modificar, p. ej., [X, RT] o [Z, MI].

5.2.3 Combinación de frames completos

Función

En el programa CN es posible asignar un frame completo a otro frame o concatenar frames.

Las concatenaciones de frames se prestan, por ejemplo, para definir varias piezas dispuestas sobre una paleta y que deben mecanizarse en un proceso de fabricación.



Para definir las operaciones de paleta, los componentes de frame podrían contener, p. ej., sólo determinados valores parciales, con cuya concatenación se generan diferentes decalajes de origen de las piezas.

Sintaxis

Asignar frames

```
DEF FRAME AJUSTE1                    Al frame programable actual se asignan
AJUSTE1=CTRANS(X,10)                 los valores del frame de definición
$P_PFRAME=AJUSTE1                    propia AJUSTE1.
DEF FRAME AJUSTE4                    El frame programable actual se memoriza
AJUSTE4=$P_PFRAME                    de forma intermedia y, en caso necesario,
$P_PFRAME=AJUSTE4                    se vuelve a almacenar.
```

Cadenas de frames

Los frames se concatenan en el orden programado y los componentes de frame (decalajes, rotaciones, etc.) se ejecutan aditivamente uno tras otro.

$P_IFRAME=$P_UIFR[15]:$P_UIFR[16]	$P_UIFR[15] contiene, p. ej., datos para decalajes de origen. A continuación, se procesan – sobre esta base – los datos de $P_UIFR[16] (p. ej., datos para rotaciones).
$P_UIFR[3]=$P_UIFR[4]:$P_UIFR[5]	El frame ajustable 3 se genera concatenando los frames ajustables 4 y 5.

Nota

Tenga en cuenta que los frames se tienen que conectar mediante el operador de concatenación "dos puntos" **:**.

5.2.4 Definición de frames nuevos (DEF FRAME)

Función

Además de los frames predefinidos y ajustables anteriormente descritos, existe la posibilidad de crear nuevos frames. En este caso se trata de variables del tipo FRAME que se pueden definir con cualquier nombre.

Con las funciones CTRANS, CROT, CSCALE, CMIRROR se pueden asignar valores a los frames en el programa CN.

Sintaxis

```
DEF FRAME PALETTE1
PALETTE1=CTRANS(…):CROT(…)…
```

Descripción

`DEF FRAME`	Generar nuevos frames.
`PALETTE1`	Nombre del nuevo frame
`=CTRANS(...):` `CROT(...)...`	Asignar valores a las funciones posibles

5.3 Decalaje basto y fino (CFINE, CTRANS)

Función

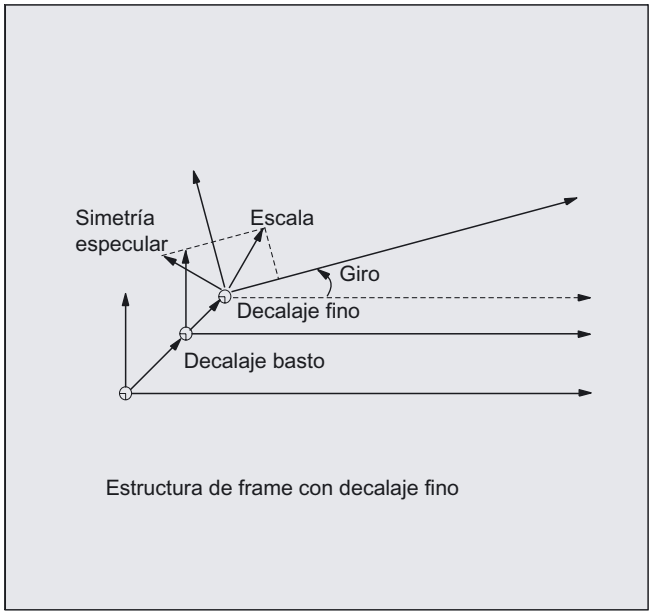
Decalaje fino

Con el comando `CFINE(X, ...,Y ...)` se puede programar un decalaje fino del frame básico y de todos los frames ajustables.

El decalaje fino sólo es posible con DM18600 $MN_MM_FRAME_FINE_TRANS=1.

Decalaje basto

Con `CTRANS(...)` se define el decalaje basto.



Estructura de frame con decalaje fino

Los decalajes aproximado y fino se suman para obtener el decalaje total.

Sintaxis

```
$P_UBFR=CTRANS(x, 10) : CFINE(x, 0.1)
: CROT(x, 45)
```
;Concatenación de decalaje,
;decalaje fino y rotación

```
$P_UIFR[1]=CFINE(x, 0.5 y, 1.0, z,
0.I)
```
;el frame completo se sobrescribe con
;CFINE incl. decalaje basto

Para acceder a los componentes individuales del decalaje fino es necesario definir la componente FI (Translation Fine).

```
DEF REAL FINEX                          ;Definición de la variable FINEX
FINEX=$P_UIFR[$P_UIFNUM, x, FI]         ;Emisión del decalaje fino
                                        ;a través de la variable FINEX
FINEX=$P_UIFR[3, x, FI]$P               ;Emisión del decalaje fino del eje x
                                        ;en el tercer frame a través de la
                                        ;variable FINEX
```

Descripción

`CFINE(x, valor, y, valor, z, valor)`	Decalaje fino para varios ejes. Decalaje aditivo (translation).
`CTRANS(x, valor, y, valor, z, valor)`	Decalaje basto para varios ejes. Decalaje absoluto (translation).
`x y z`	Decalaje de origen de los ejes (máx. 8)
`Valor`	Parte de traslación

Fabricante de la máquina

Con DM18600 $MN_ MM_FRAME_FINE_TRANS puede configurarse el decalaje fino en las siguientes variantes:

0:
El decalaje fino no se puede introducir o programar. G58 y G59 no son posibles.

1:
El decalaje fino para frames ajustables, frames básicos, frames programables, G58 y G59 se puede introducir o programar.

Descripción

Un decalaje fino modificado a través de una operación HMI sólo se activa tras la activación del correspondiente frame; es decir, que la activación se realiza a través de G500, G54...G599. Un decalaje fino activado de un frame permanece activo mientras el frame esté activo.

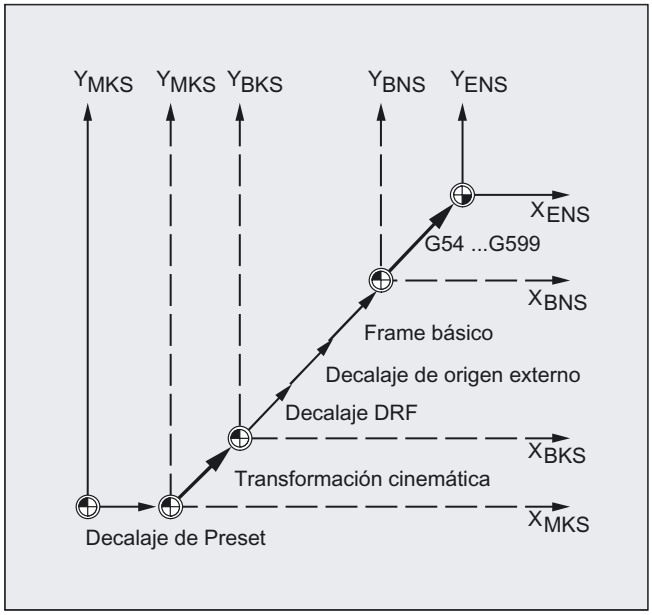
El frame programable no tiene componente de decalaje fino. Si al frame programable se asigna un frame con decalaje fino, entonces su decalaje total resulta de la suma de los decalajes fino y basto. Al leer los frames programables, el decalaje fino vale siempre cero.

5.4 Decalaje de origen externo

Funcionamiento

Este decalaje ofrece otra oportunidad para desplazar el origen entre el sistema de coordenadas básico y el referido a la pieza.

Para el decalaje de origen externo se pueden programar sólo decalajes lineales.



Programación

La programación de los valores de decalaje, $AA_ETRANS tiene lugar asignando las variables del sistema correspondientes a los canales.

Asignar valor de decalaje

```
$AA_ETRANS[eje]=RI
```

RI es la variable de cálculo del tipo REAL que contiene el nuevo valor.

Generalmente, el decalaje externo no está definido en el programa de pieza, sino que es activado por el PLC.

Nota

El valor escrito en el programa de pieza sólo se activa cuando se haya aplicado la señal correspondiente en la interfaz VDI
(interfaz NCU-PLC).

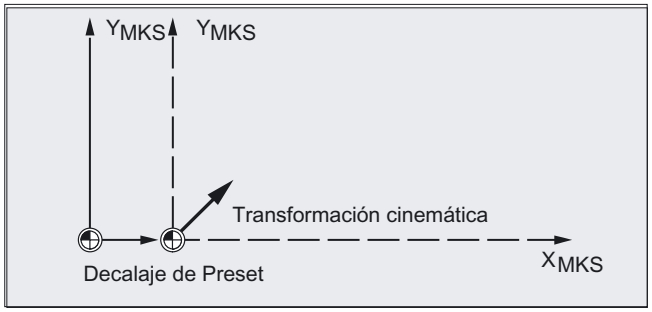
5.5 Decalaje de Preset (PRESETON)

Función

Para aplicaciones especiales puede resultar necesario asignar un nuevo valor real programado a uno o varios de los ejes en la posición actual (en reposo).

⚠ PRECAUCIÓN

Con la función PRESETON pierde su vigencia el punto de referencia. Por consiguiente, esta función se debe utilizar solo en ejes que no requieran punto de referencia. Si se desea restablecer el sistema original, utilizar la función G74 para buscar los puntos de referencia; ver apartado "Administración de programas y ficheros".



Sintaxis

```
PRESETON(eje, valor, ...)
```

Descripción

`PRESETON`	Poner valor real
`Eje`	Definición del eje de máquina
`Valor`	Nuevo valor real a tomar para el eje indicado

Nota

La definición de valores reales con acciones síncronas sólo se debería realizar con la palabra reservada "WHEN" o "EVERY".

Ejemplo

Los valores reales se asignan en el sistema de coordenadas referido a la máquina; los valores están referidos a los ejes de la máquina.

```
N10 G0 A760
N20 PRESETON(A1,60)
```

El eje A se desplaza a la posición 760. Al eje de máquina A1 se le asigna el nuevo valor real 60 en la posición 760. A partir de entonces, el posicionamiento tiene lugar en el nuevo sistema del valor real.

5.6 Cálculo de frame a partir de 3 puntos de medida en el espacio (MEAFRAME)

Función

MEAFRAME constituye una ampliación del lenguaje del 840D para el apoyo de los ciclos de medida.

La función MEAFRAME calcula el frame a partir de los tres puntos ideales y de los tres puntos asociados medidos.

Cuando se posiciona una pieza para su mecanizado, su posición respecto al sistema de coordenadas cartesiano de la máquina puede estar caracterizada por desplazamientos y giros. Para un exacto mecanizado o medición es necesario o bien un ajuste físico, operación costosa, o modificar las instrucciones de movimiento del programa de pieza.

Un frame puede definirse palpando tres puntos en el espacio de los que se conocen sus posiciones ideales. Se palpa con un sensor de contacto u óptico que toca agujeros o esferas de medición fijados con precisión en la placa soporte.

Sintaxis

```
MEAFRAME IDEAL_POINT,MEAS_POINT,FIT_QUALITY)
```

Descripción

`MEAFRAME`	Cálculo del frame a partir de 3 puntos medidos en el espacio	
`IDEAL_POINT`	Campo real dim. que incluye las tres coordenadas de los puntos ideales	
`MEAS_POINT`	Campo real dim. incluye las tres coordenadas de los puntos medidos	
`FIT_QUALITY`	Variable REAL	que permite ofrecer las siguientes informaciones:
	-1:	Los puntos ideales se sitúan prácticamente en una recta. No ha sido posible calcular el frame. La variable frame devuelta contiene un frame neutro.
	-2:	Los puntos de medida se sitúan prácticamente en una recta. No ha sido posible calcular el frame. La variable frame devuelta contiene un frame neutro.
	-4:	El cálculo de la matriz de rotación fracasa por otra razón.
	Valor positivo:	Suma de distorsiones (distancias entre los puntos) que se necesitan para trasladar el triángulo medido congruente con el triángulo ideal.

Nota

Calidad de la medición

Para poder asignar las coordenadas medidas a las ideales a través de una rotación/translación combinada, el triángulo abierto desde los puntos de medida tiene que ser congruente con el triángulo ideal. Esto se consigue con un algoritmo de compensación que minimiza la suma de los cuadrados de las desviaciones que trasladan el triángulo medido al triángulo ideal.

La distorsión efectiva necesaria de los puntos de medida puede considerarse un indicador de la calidad de la medición; por ello lo entrega también `MEAFRAME` en calidad de variable adicional.

Nota

El frame generado por `MEAFRAME` se puede transformar, con la función `ADDFRAME,` en otro frame en la cadena de frames.
Ver ejemplo: Concatenación de frames "Concatenación con ADDFRAME".

Para más información sobre los parámetros de `ADDFRAME(FRAME, STRING)` ver /FB1/ Manual de funciones básicas; Ejes, Sistemas de coordenadas, Frames (K2), apartado "Concatenación de FRAMES".

Ejemplo

Código del programa	Comentarios
	; Programa de pieza 1
DEF FRAME CORR_FRAME	

Definir puntos de medida

Programación	Comentarios
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)	; para prueba
DEF REAL FIT_QUALITY = 0	
DEF REAL ROT_FRAME_LIMIT = 5	; Permite un giro máx. de 5 grados de la posición de la pieza
DEF REAL FIT_QUALITY_LIMIT = 3	; Permite un decalaje máx. de 3 mm entre el triángulo ideal y el medido
DEF REAL SHOW_MCS_POS1[3]	
DEF REAL SHOW_MCS_POS2[3]	
DEF REAL SHOW_MCS_POS3[3]	

```
Código del programa                                  Comentarios
N100 G01 G90 F5000
N110 X0 Y0 Z0
N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS
_POINT,FIT_QUALITY)

N230 IF FIT_QUALITY < 0
SETAL(65000)
GOTOF NO_FRAME
ENDIF
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT
SETAL(65010)
GOTOF NO_FRAME
ENDIF
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT      ; Limitación del 1er ángulo RPY
SETAL(65020)
GOTOF NO_FRAME
ENDIF
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT      ; Limitación del 2º ángulo RPY
SETAL(65021)
GOTOF NO_FRAME
ENDIF
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT      ; Limitación del 3er ángulo RPY
SETAL(65022)
GOTOF NO_FRAME
ENDIF
N300 $P_IFRAME=CORR_FRAME                       ; Activar frame palpado con un
                                                  frame definible
                                                ; Comprobar el frame posicionando
                                                  los ejes geométricos en los
                                                  puntos ideales
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1]
Z=IDEAL_POINT[0,2]
N410 SHOW_MCS_POS1[0]=$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]
```

Código del programa	Comentarios
N700 G500	; Desactivar frame ajustable porque está ajustado por defecto con frame cero (no se ha registrado ningún valor).
No_FRAME	; Desactivar frame ajustable porque está ajustado por defecto con frame cero (no se ha registrado ningún valor)
M0	
M30	

Ejemplo: Concatenación de frames

Concatenación de MEAFRAME para correcciones

La función `MEAFRAME( )` suministra un frame de corrección. Si este frame de corrección se concatena con el frame ajustable `$P_UIFR[1]` que estaba activo en el momento de la llamada a la función, p. ej., `G54`, se obtiene un frame ajustable para conversiones adicionales para el desplazamiento o la ejecución.

Concatenación con ADDFRAME

Si este frame de corrección tiene que actuar en la cadena de frames o en otro punto, o si antes de frame ajustable existen otros frames activos, se puede utilizar la función `ADDFRAME( )` para la concatenación en uno de los frames básicos de canal o un frame de sistema.

En los frames no deben estar activos:

- Simetría especular con `MIRROR`
- Escala con `SCALE`

Los parámetros de entrada para valores teóricos y reales son las coordenadas de pieza. En el sistema básico del control, dichas coordenadas se tienen que indicar siempre:

- en forma métrica en pulgadas `(G71/G70)` y como
- medida relativa al radio `(DIAMOF)`.

5.7 Frames globales NCU

Función

Los frames globales NCU sólo existen una vez por cada NCU para todos los canales. Los frames globales NCU se pueden escribir y leer desde todos los canales. La activación de los frames globales NCU tiene lugar en el correspondiente canal.

Con frames globales se pueden añadir decalajes, escalas y simetría a **ejes de canal y de máquina**.

Relaciones geométricas y cadenas de frames

Para los frames globales no existe ninguna relación geométrica entre los ejes. Por ello no se pueden ejecutar rotaciones y programación de descriptores de ejes geométricos.

- Las rotaciones no se pueden aplicar a frames globales. La programación de una rotación se rechaza con la alarma: "18310 Canal %1 Secuencia %2 Frame: Rotación no permitida".
- La concatenación de frames globales y frames específicos del canal es posible. El frame resultante contiene todas las porciones de frame, incluyendo las rotaciones para todos los ejes. La asignación de un frame con porciones de rotación a un frame global se rechaza con la alarma "Frame: Rotación no permitida".

Frames globales NCU

Frames básicos globales NCU $P_NCBFR[n]

Se pueden configurar hasta 8 frames básicos globales NCU.

Al mismo tiempo pueden existir frames básicos específicos del canal.

Los frames globales se pueden escribir y leer desde todos los canales de una NCU. Al escribir en frames globales el usuario debe preocuparse por la coordinación de canales. Ello se puede realizar p. ej. con metas de espera `(WAITMC)`.

Fabricante de la máquina

La cantidad de frames básicos globales se configura mediante datos de máquina, ver /FB1/ Manual de funciones básicas; Ejes, Sistemas de coordenadas, Frames (K2).

Frames ajustables globales NCU $P_UIFR[n]

Todos los frames ajustables `G500, G54...G599` se pueden configurar como frames globales NCU o específicos para el canal.

Fabricante de la máquina

Se pueden reconfigurar todos los frames ajustables mediante un dato de máquina $MN_MM_NUM_GLOBAL_USER_FRAMES en frames globales.

Como descriptores del eje para los comandos de programa frame se pueden utilizar descriptores de canal y descriptores de eje de máquina. La programación de descriptores de eje geométrico se rechaza con una alarma.

5.7.1 Frames específicos del canal ($P_CHBFR, $P_UBFR)

Funcionamiento

Los frames y los frames básicos se pueden escribir y leer:

- a través del programa de pieza
- a través de BTSS

por intervención del operador, p. ej., HMI Advanced, y por el PLC.

El decalaje fino es posible también para los frames globales. La supresión de frames globales se realiza de igual manera que en el caso de frames específicos del canal a través de `G53`, `G153`, `SUPA` y `G500`.

Fabricante de la máquina

Con el MD28081 MM_NUM_BASE_FRAMES se puede configurar el número de frames básicos en el canal. La configuración básica está definida de modo que existe al menos un frame básico por canal. Como máximo son posibles 8 frames básicos por cada canal. Además de los 8 frames básicos en el canal puede haber 8 frames básicos globales NCU.

Frames específicos del canal

$P_CHBFR[n]

Mediante la variable de sistema `$P_CHBFR[n]` se pueden leer y escribir los frames básicos. Al escribir un frame básico no se activa el frame básico total concatenado, sino que la activación se realiza sólo con la ejecución de una instrucción `G500`, `G54...G599`. La variable sirve en primer lugar como memoria para los procesos de escritura en el frame básico de HMI o PLC. Las variables de tipo frame se aseguran con la protección de datos.

Primer frame básico en el canal

Una escritura en la variable predefinida `$P_UBFR` no activa al mismo tiempo el frame básico con el índice de matriz 0, sino que la activación sólo se produce con la ejecución de una instrucción `G500`, `G54...G599`. La variable también puede ser escrita y leída directamente desde el programa.

$P_UBFR

`$P_UBFR` es idéntico a `$P_CHBFR[0]`. Como estándar existe siempre un frame básico en el canal, de modo que la variable de sistema es compatible con estados más antiguos. Si no existe ninguna variable frame predefinida específica del canal, al escribir o leer se emite la alarma "Frame: Instrucción no permitida".

5.7.2 Frames activos en el canal

Funcionamiento

Frames activos en el canal son introducidos desde el programa de pieza a través de las variables de sistema correspondientes a dichos frames. Esto incluye los frames de sistema. A través de estas variables de sistema se puede leer y escribir en el programa de pieza el frame de sistema actual.

Frames activos en el canal actuales

Vista general

Frames de sistema actuales	para:
$P_PARTFRAME	TCARR y PAROT
$P_SETFRAME	Poner valor real y Aproximación con contacto
$P_EXTFRAME	Decalaje de origen externo
$P_NCBFRAME[n]	Frames básicos globales NCU actuales
$P_CHBFRAME[n]	Frames básicos de canal actuales
$P_BFRAME	Primer frame básico actual en el canal
$P_ACTBFRAME	Frame básico total
$P_CHBFRMASK y $P_NCBFRMASK	Frame básico total
$P_IFRAME	Frame ajustable actual
Frames de sistema actuales	para:
$P_TOOLFRAME	TOROT y TOFRAME
$P_WPFRAME	Puntos de referencia de pieza
$P_TRAFRAME	Transformadas
$P_PFRAME	Frame programable actual
Frame de sistema actual	para:
$P_CYCFRAME	Ciclos
P_ACTFRAME	Frame total actual
Concatenación de FRAMES	El frame actual se compone del frame básico total

$P_NCBFRAME[n] Frames básicos globales NCU actuales

A través de la variable de sistema `$P_NCBFRAME[n]` se pueden leer y escribir los elementos de matriz de frames básicos globales actuales. El frame básico global resultante se incorpora a través del proceso de escritura en el canal.

El frame modificado solamente se activa en el canal, en el que el frame ha sido programado. Para modificar el frame para todos los canales de una NCU, se tienen que escribir al mismo tiempo $P_NCBFR[n] y $P_NCBFRAME[n]. Entonces los demás canales deben activar todavía la variable frame, p. ej., con G54. Al escribir una variable frame predefinida se calcula de nuevo la variable frame predefinida total.

$P_CHBFRAME[n] Frames básicos de canal actuales

A través de la variable de sistema `$P_CHBFRAME[n]` se pueden leer y escribir los elementos de matriz de frames básicos de canal actuales. La variable frame predefinida total resultante se tiene en cuenta en el canal con el proceso de escritura. Al escribir una variable frame predefinida se calcula de nuevo la variable frame predefinida total.

$P_BFRAME Primer frame básico actual en el canal

A través de la variable frame predefinida `$P_BFRAME` se puede leer y escribir en el programa de pieza el frame básico actual con el índice de matriz 0 que es válido en el canal. La variable frame predefinida escrita se tiene en cuenta a efecto inmediato.

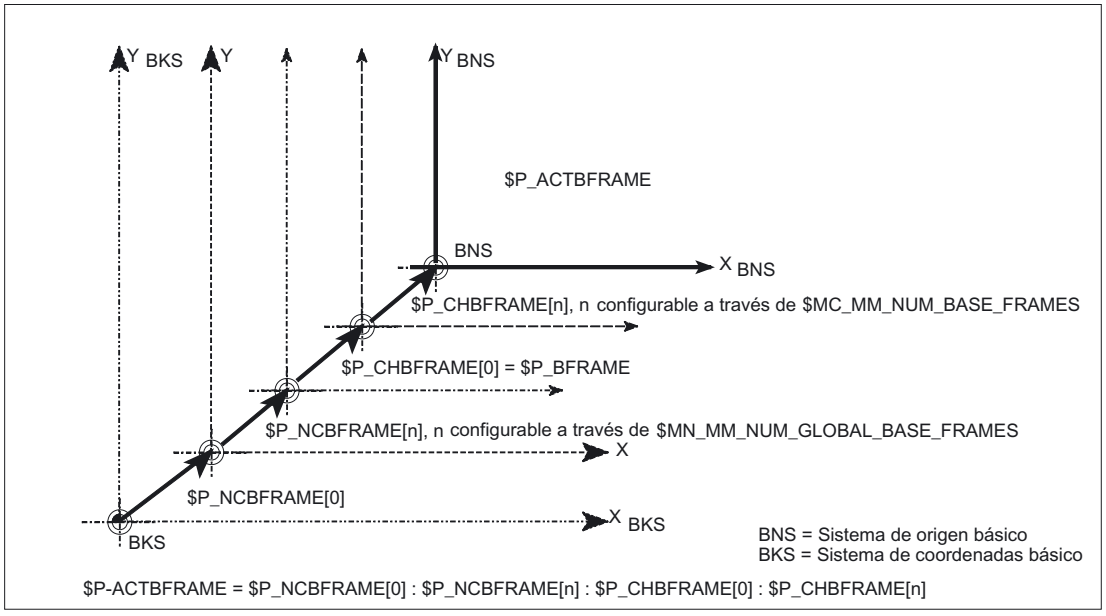
`$P_BFRAME` es idéntico a `$P_CHBFRAME[0]`. Como estándar, la variable de sistema tiene siempre un valor válido. Si no existe ninguna variable frame predefinida específica del canal, al escribir o leer se emite la alarma "Frame: Instrucción no permitida".

$P_ACTBFRAME Frame básico total

La variable `$P_ACTFRAME` determina el frame básico total concatenado. La variable sólo puede ser leída.

`$P_ACTFRAME` equivale a

`$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n].`



$P_CHBFRMASK y $P_NCBFRMASK Frame básico total

A través de la variable de sistema `$P_CHBFRMASK` y `$P_NCBFRMASK` K, el usuario puede elegir qué frame básico quiere incluir en el cálculo del frame básico "total". Las variables sólo pueden ser programadas en el programa y leídas vía la BTSS. El valor de las variables se interpreta como máscara de bits e indica qué elemento de matriz de variable frame predefinida de `$P_ACTFRAME` se tiene en cuenta para el cálculo.

Con `$P_CHBFRMASK` se puede predefinir cuáles son las variables frame predefinidas específicas del canal, y con `$P_NCBFRMASK` cuáles son las variables frame predefinidas NCU globales que se tendrán en cuenta.

Con la programación de las variables se calcula de nuevo la variable frame predefinida total y la variable frame total. Tras el Reset y con el ajuste base, los valores son los siguientes:

`$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK` y

`$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK`.

P. ej.,

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]
```

$P_IFRAME Frame ajustable actual

A través de la variable frame predefinida `$P_IFRAME` se puede leer y escribir en el programa de pieza el frame ajustable actual válido en el canal. La variable frame ajustable escrita se tiene en cuenta a efecto inmediato.

En caso de variables frame NCU globales ajustables, la variable frame modificada sólo actúa en el canal en el cual ha sido programado el frame. Con el fin de modificar el frame para todos los canales de una NCU, se tienen que escribir al mismo tiempo `$P_UIFR[n]` y `$P_IFRAME`. Entonces los demás canales deben activar todavía el correspondiente frame, p. ej., con G54.

$P_PFRAME Frame programable actual

`$P_PFRAME` es el frame programable que resulta de la programación de `TRANS/ATRANS`, `G58/G59`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR` o de la asignación de `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` al FRAME programable.

Variable frame programable actual que establece la relación entre el

- Sistema de origen ajustable (ENS) y el
- Sistema de coordenadas de pieza (WKS).

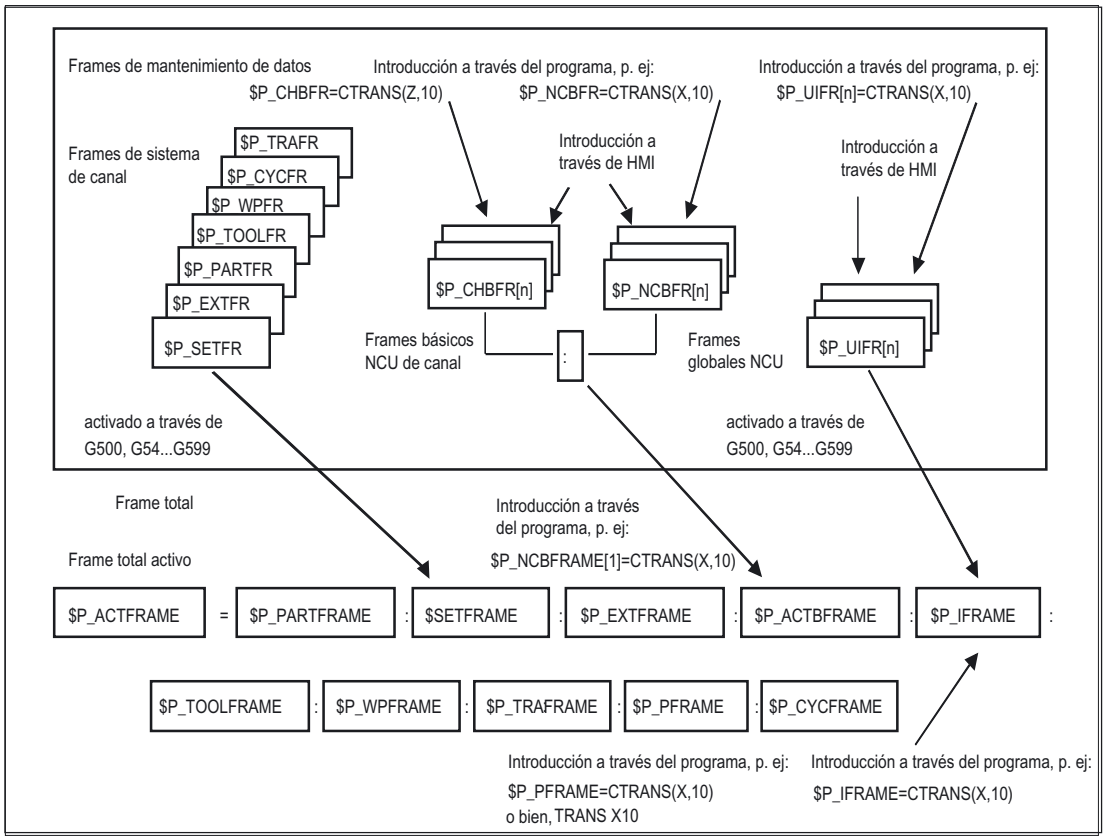
P_ACTFRAME Frame total actual

El frame total actual resultante $P_ACTFRAME se obtiene entonces como concatenación de todos los frames básicos, del frame ajustable actual y del frame programable. La variable frame actual se actualiza siempre, cuando se modifica una proporción de variable frame.

`$P_ACTFRAME` equivale a

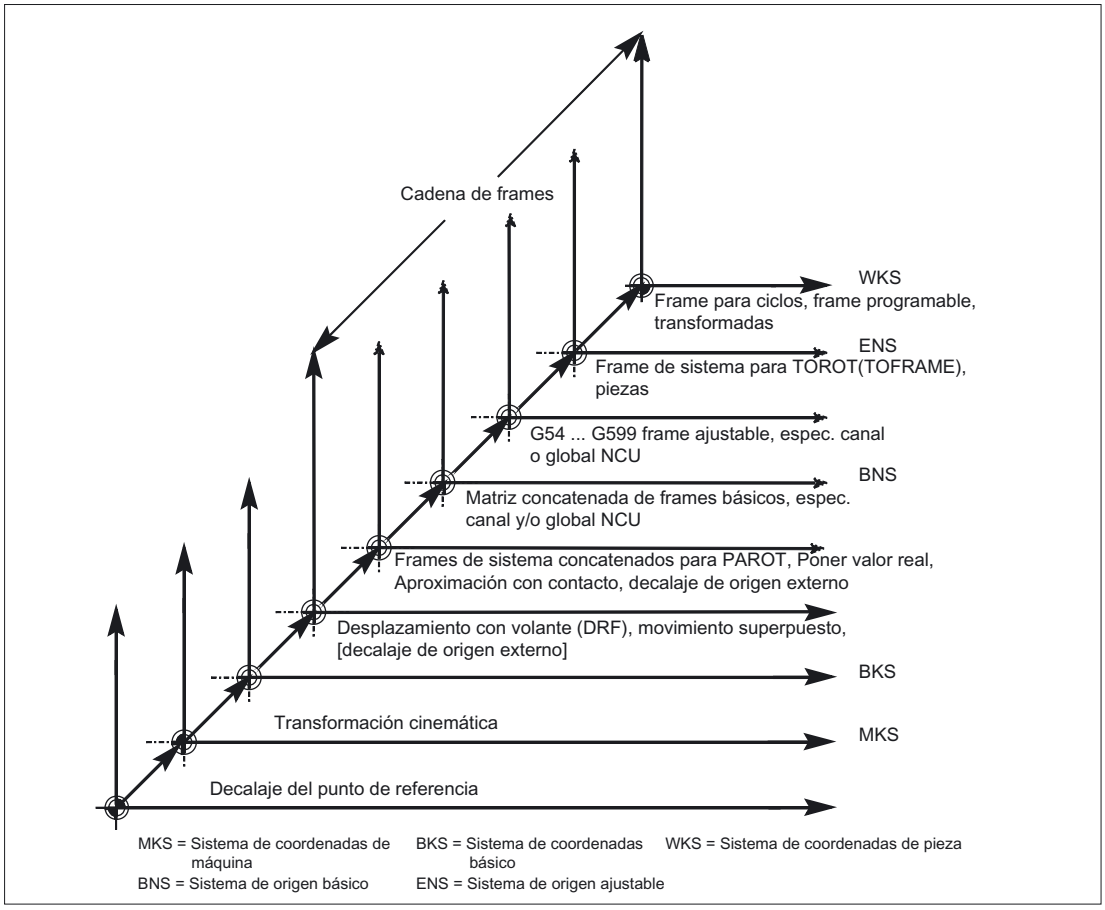
`$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :`

`$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME`



Concatenación de frames

El frame actual se compone del frame básico total, del frame ajustable, del frame de sistema y del frame programable según el frame total actual arriba indicado.



Transformadas

6

6.1 Programación general de los tipos de transformada

Función general

Para adaptar el control a distintas cinemáticas de máquina existe la selección de programar tipos de transformada con parámetros apropiados. A través de estos parámetros se pueden acordar, para la transformada seleccionada, tanto la orientación de la herramienta en el espacio como también los movimientos de orientación de los ejes giratorios.

En las transformadas de tres, cuatro y cinco ejes, los datos de posición programados se refieren siempre a la punta de la herramienta que sigue en posición ortogonal a la superficie de mecanizado situada en el espacio. Las coordenadas cartesianas se convierten del sistema de coordenadas básico al sistema de coordenadas de máquina y se refieren a los ejes geométricos. Éstos describen el punto de trabajo. Ejes giratorios virtuales describen las orientaciones de la herramienta en el espacio y se programan con TRAORI.

En la transformada cinemática se puede programar posiciones en el sistema de coordenadas cartesiano. El control transforma los movimientos de desplazamiento programados con TRANSMIT, TRACYL y TRAANG del sistema de coordenadas cartesiano a los movimientos de desplazamiento de los ejes de máquina reales.

Programación

Transformadas de tres, cuatro y cinco ejes TRAORI

La transformada de orientación acordada se activa con el comando TRAORI y los tres parámetros posibles para número de transformada, vector de orientación y correcciones de eje giratorio.

```
TRAORI(número de transformada, vector de orientación y correcciones de eje giratorio)
```

Transformadas cinemáticas

A las transformadas cinemáticas pertenecen las transformadas acordadas
`TRANSMIT(número de transformada)`

```
TRACYL(diámetro de trabajo, número de transformada)
TRAANG(ángulo del eje inclinado, número de transformada)
```

Desactivar transformada activa

Con `TRAFOOF` se puede desactivar la transformada actualmente activa.

Transformada de orientación

Transformadas de tres, cuatro y cinco ejes TRAORI

Para el mecanizado óptimo de superficies conformadas en el espacio en el área de trabajo de la máquina, las máquinas herramienta necesitan ejes adicionales a los tres ejes lineales X, Y y Z. Los ejes adicionales describen la orientación en el espacio y se denominarán a continuación como ejes de orientación. Están disponibles como ejes giratorios en cuatro tipos de maquinas con distintas cinemáticas.

1. Cabezal orientable de dos ejes, p. ej., cabezal tipo cardán con un eje giratorio paralelo a un eje lineal con mesa de herramienta fija.
2. Mesa giratoria de dos ejes, p. ej., cabezal orientable fijo con mesa de herramientas giratoria alrededor de dos ejes.
3. Cabezal orientable de un eje y mesa giratoria de un eje, p. ej., un cabezal orientable con herramienta girada con mesa de herramienta giratoria alrededor de un eje.
4. Cabezal orientable de dos ejes y mesa giratoria de un eje, p. ej., con mesa de herramienta giratoria alrededor de un eje y un cabezal orientable con herramienta que alrededor de sí misma.

Las **transformadas de 3 y 4 ejes** son formas especiales de la transformada de 5 ejes y se programan de forma análoga a las transformadas de 5 ejes.

La "**transformada genérica de 3/4/5/6 ejes**" cubre con su volumen de funciones para ejes giratorios dispuestos en ángulo recto las transformadas para el cabezal portafresas cardánico y se puede activar, como cualquier otra transformada de orientación, también para estos cuatro tipos de máquina con TRAORI. En la transformada genérica de 5/6 ejes, la orientación de herramienta tiene un tercer grado de libertad en el cual la herramienta se puede girar alrededor de su propio eje, en posición libre en el espacio frente a la dirección de la herramienta.

Bibliografía: /FB3/ Manual de funciones especiales; Transformada de 3 a 5 ejes (F2)

Estado inicial de la orientación de herramienta independiente de la cinemática

ORIRESET

Si hay una transformada de orientación activa con TRAORI, pueden indicarse con ORIRESET los estados iniciales de hasta 3 ejes de orientación con parámetros opcionales A, B y C. La asignación del orden de los parámetros programados a los ejes giratorios se realiza según el orden de los ejes de orientación establecido mediante la transformada. La programación de ORIRESET(A, B, C) produce un desplazamiento lineal y síncrono de los ejes de orientación desde su posición instantánea hasta la posición de estado inicial indicada.

Transformadas cinemáticas

TRANSMIT y TRACYL

Para fresados en tornos se puede programar, para la transformada acordada,

1. el mecanizado mediante herramienta motorizada en superficies de refrentado con TRANSMIT o
2. el mecanizado de ranuras con cualquier orientación en piezas cilíndricas con TRACYL.

TRAANG

Si el eje de penetración tiene que poder pasar también en posición inclinada (p. ej., para la tecnología Rectificación) se puede programar con TRAANG un ángulo parametrizable para la transformada acordada.

Desplazamiento PTP cartesiano

La transformada cinemática comprende también el "Desplazamiento PTP cartesiano" en el cual se pueden programar hasta 8 distintas posiciones de articulación STAT=. Las posiciones se programan en el sistema de coordenadas cartesiano, realizándose el desplazamiento de la máquina en coordenadas de máquina.

Bibliografía:
/FB2/ Manual de funciones de ampliación; Transformada cinemática (M1)

Transformadas concatenadas

Se pueden conectar dos transformadas en serie. En la segunda transformada concatenada de esta manera se adoptan las partes de movimiento para los ejes de la primera transformada.

Para la primera transformada son posibles:

- Transformada de orientación TRAORI
- Transformada polar TRANSMIT
- Transformada cilíndrica TRACYL
- Transformada eje inclinado TRAANG

La segunda transformada tiene que ser Eje inclinado TRAANG

6.1.1 Movimientos de orientación en las transformadas

Movimientos de desplazamiento y movimientos de orientación

Los movimientos de desplazamiento de las orientaciones programables dependen principalmente del tipo de máquina. En la transformada de tres, cuatro y cinco ejes con TRAORI, los ejes rotatorios o los ejes lineales basculables describen los movimientos de orientación de la herramienta.

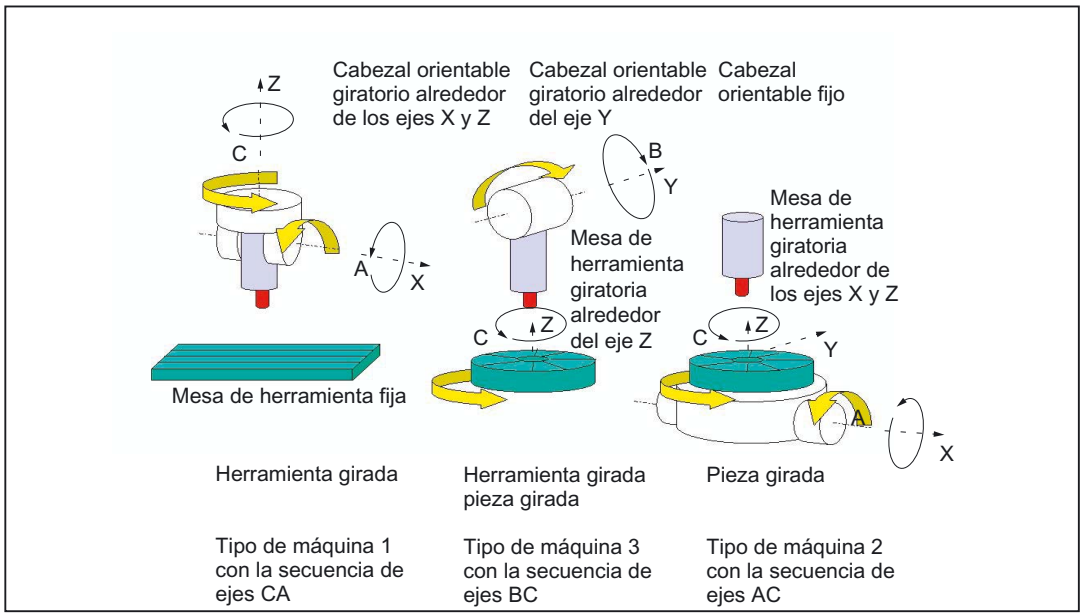
Modificaciones de las posiciones de los ejes giratorios que participan en la transformada de orientación producen movimientos de compensación de los demás ejes de máquina. La posición de la punta de la herramienta permanece inalterada.

Según la aplicación, los movimientos de orientación de la herramienta se pueden programar a través de los descriptores de eje giratorio A…, B…, C… de los ejes virtuales, indicando ángulos de Euler o RPY o vectores de dirección o normales a la superficie, vectores normalizados para el eje giratorio de un cono o para la orientación intermedia en una superficie de cono.

En la transformada cinemática con TRANSMIT, TRACYL y TRAANG, el control transforma los movimientos de desplazamiento programados del sistema de coordenadas cartesiano a los movimientos de desplazamiento de los ejes de máquina reales.

Cinemática de máquina en transformada de tres, cuatro y cinco ejes TRAORI

La herramienta o la mesa de herramienta pueden ser giratorias con hasta dos ejes giratorios. También es posible una combinación de un cabezal orientable de un eje y una mesa giratoria.



Tipo de máquina	Programación de la orientación
Transformada de tres ejes tipos de máquina 1 y 2	Programación de la orientación de herramienta únicamente en el plano perpendicular al eje rotatorio. Existen **dos** ejes de translación (ejes lineales) y **un** eje rotatorio (eje giratorio)
Transformada de cuatro ejes tipos de máquina 1 y 2	Programación de la orientación de herramienta únicamente en el plano perpendicular al eje rotatorio. Existen **tres** ejes de translación (ejes lineales) y **un** eje rotatorio (eje giratorio)
Transformada de cinco ejes tipos de máquina 3 Cabezal orientable de un eje y mesa giratoria de un eje	Programación de la transformada de orientación Cinemática con **tres** ejes lineales y **dos** ejes giratorios ortogonales. Los ejes giratorios son paralelos a dos de los tres ejes lineales. El primer eje giratorio es movido por dos ejes lineales cartesianos. Gira el tercer eje lineal con la herramienta. El segundo eje giratorio gira la pieza.

Transformadas genéricas de 5/6 ejes

Tipo de máquina	Programación de la transformada de orientación
Transformada genérica de cinco/seis ejes tipos de máquina 4 Cabezal orientable de dos ejes con herramienta giratoria sobre sí misma y mesa giratoria de un eje	Programación de la transformada de orientación Cinemática con **tres** ejes lineales y **tres** ejes giratorios ortogonales. Los ejes giratorios son paralelos a dos de los tres ejes lineales. El primer eje giratorio es movido por dos ejes lineales cartesianos. Gira el tercer eje lineal con la herramienta. El segundo eje giratorio gira la pieza. La orientación básica de la herramienta puede programarse mediante un giro adicional sobre sí misma con el ángulo de giro THETA.

Al llamar a la "transformada genérica de tres, cuatro y cinco/seis ejes" se puede consignar adicionalmente la orientación básica de la herramienta. Ya no se aplican las limitaciones con respecto a las direcciones de los ejes giratorios. Si los ejes giratorios no son exactamente perpendiculares entre sí o los ejes giratorios existentes no son exactamente paralelos a los ejes lineales, la "transformada genérica de cinco/seis ejes" puede suministrar mejores resultados para la orientación de herramienta.

Transformadas cinemáticas TRANSMIT, TRACYL y TRAANG

Para fresados en tornos o en un eje que se puede pasar en posición inclinada en la rectificación, se aplican, en función de la transformada, las siguientes disposiciones de ejes en el caso estándar:

TRANSMIT	Activación de la transformada polar
Mecanizado mediante herramienta motorizada en superficies de refrentado	un eje giratorio un eje de penetración perpendicular al eje giratorio un eje longitudinal paralelo al eje giratorio

TRACYL	Activación de la transformada de la envolvente del cilindro
Mecanizado de ranuras con cualquier orientación en piezas cilíndricas	un eje giratorio un eje de penetración perpendicular al eje giratorio un eje longitudinal paralelo al eje giratorio

TRAANG	Activación de la transformada Eje inclinado
Mecanizado con eje de penetración inclinado	un eje giratorio un eje de penetración con un ángulo parametrizable un eje longitudinal paralelo al eje giratorio

Desplazamiento PTP cartesiano

El movimiento de la máquina se realiza en coordenadas de máquina y se programa con:

TRAORI	Activación de la transformada
PTP Desplazamiento Punto a punto	Desplazamiento a la posición en el sistema de coordenadas cartesiano (MKS)
CP	Movimiento interpolado de los ejes cartesianos en el (BKS)
STAT	La posición de las articulaciones depende de la transformada
TU	En qué ángulo se desplazan los ejes por el trayecto más corto

Desplazamiento PTP en transformada genérica de 5/6 ejes

El movimiento de la máquina se realiza en coordenadas de máquina y la orientación de herramienta puede programarse tanto con posiciones de ejes giratorios como con los vectores de Euler o los ángulos RPY, independientes de la cinemática, o con los vectores de dirección.

Además, son posibles la interpolación de ejes giratorios, la interpolación de vectores con interpolación circular de gran radio o la interpolación del vector de orientación sobre una superficie de cono.

Ejemplo: Transformada de tres a cinco ejes con un cabezal de fresado cardánico

La máquina herramienta tiene al menos 5 ejes, de ellos:

- Tres ejes de translación para movimientos en línea recta que desplazan el punto de trabajo a cualquier posición en el área de trabajo.
- Dos ejes rotatorios dispuestos en un ángulo configurable (habitualmente 45 grados) permiten a la herramienta ocupar orientaciones en el espacio que, en la disposición en 45 grados, se limitan a una semiesfera.

6.1.2 Resumen de la transformada de orientación TRAORI

Posibles tipos de programación en relación con TRAORI

Tipo de máquina	Programación con transformada TRAORI activa
Tipos de máquina 1, 2 ó 3, cabezal orientable de dos ejes o mesa giratoria de dos ejes o una combinación de un cabezal orientable y una mesa giratoria de un eje.	La secuencia de ejes de los ejes de orientación y la dirección de orientación de la herramienta se pueden configurar **relativas a la máquina** a través de datos de máquina de forma dependiente de la cinemática de la máquina o **relativas a la pieza** con orientación programable de forma independiente de la cinemática de la máquina El sentido de giro de los ejes de orientación en el sistema de referencia se programa con: - ORIMKS sistema de referencia = sistema de coordenadas de máquina - ORIWKS sistema de referencia = sistema de coordenadas de pieza El ajuste básico es ORIWKS. Programación de los ejes de orientación con: A, B, C indicación directa de las posiciones de ejes A2, B2, C2 programación de ángulos de ejes virtuales mediante - ORIEULER a través de ángulos de Euler (estándar) - ORIRPY a través de ángulos RPY - ORIVIRT1 a través de ejes de orientación virtuales de 1ª definición - ORIVIRT2 a través de ejes de orientación virtuales de 2ª definición con distinción del tipo de interpolación: **Interpolación lineal** - ORIAXES de ejes de orientación o ejes de máquina **Interpolación circular de gran radio**(interpolación del vector de orientación) - ORIVECT de ejes de orientación Programación de los ejes de orientación por indicación A3, B3, C3 de las componentes vectoriales (normal de dirección/superficie) Programación de la orientación de herramienta resultante A4, B4, C4 del vector normal de superficie al inicio de la secuencia A5, B5, C5 del vector normal de superficie al final de la secuencia LEAD ángulo de avance para la orientación de la herramienta TILT ángulo lateral para la orientación de la herramienta

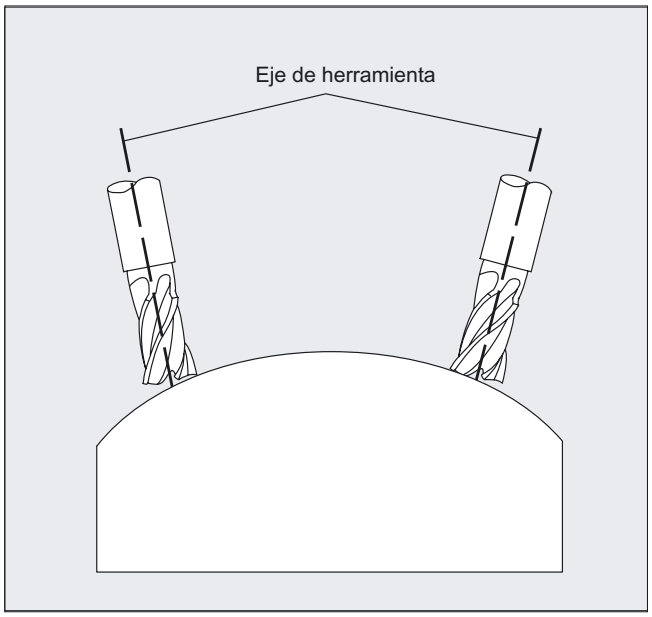
Tipo de máquina	Programación con transformada TRAORI activa
	Interpolación del vector de orientación en una superficie de cono Cambios de orientación en una superficie de cono situada **libremente en el espacio** por interpolación: - ORIPLANE en el plano (interpolación circular de gran radio) - ORICONCW en una superficie de cono en sentido horario - ORICONCCW en una superficie de cono en sentido antihorario A6, B6, C6 vector de dirección (eje de giro de cono) -OICONIO Interpolación en una superficie de cono con: A7, B7, C7 vectores intermedios (orientación inicial y final) o - ORICONTO en una superficie de cono, transición tangencial Cambios de orientación relativos**a una trayectoria** con - ORICURVE especificación del desplazamiento de dos puntos de contacto a través de PO[XH]=(xe, x2, x3, x4, x5) polinomios de orientación de hasta 5º grado PO[YH]=(ye, y2, y3, y4, y5) polinomios de orientación de hasta 5º grado PO[ZH]=(ze, z2, z3, z4, z5) polinomios de orientación de hasta 5º grado - ORIPATHS Alisado del desarrollo de la orientación con A8, B8, C8 fase de reorientación de la herramienta corresponde a: dirección y longitud del trayecto de la herramienta en el movimiento de levantamiento
Tipos de máquina 1 y 3 Otros tipos de máquina con giro adicional de la herramienta sobre sí misma requieren un tercer eje giratorio Transformada de orientación, como p. ej. transformada genérica de 6 ejes. Giros del vector de orientación.	Programación **de los giros** de la orientación de herramienta con LEAD ángulo de avance Ángulo relativo al vector normal a la superficie PO[PHI] Programación de un polinomio de hasta 5º grado TILT ángulo lateral giro alrededor de la trayectoria tangente (dirección Z) PO[PSI] Programación de un polinomio de hasta 5º grado THETA Ángulo de giro (giro alrededor de la dirección de la herramienta en Z) THETA= valor que se alcanza al final de la secuencia THETA=AC(...) Conmutar secuencia a secuencia a acotado absoluto THETA=IC(...) Conmutar secuencia a secuencia a acotado incremental THETA=Θ_e Interpolar ángulo programado G90/G91 PO[THT]=(..) Programación de un polinomio de hasta 5º grado Programación del vector de giro - ORIROTA Giro absoluto - ORIROTR vector de giro relativo - ORIROTT vector de giro tangencial
Orientación relativa a la trayectoria para cambios de orientación relativos a la trayectoria o giro del vector de giro tangencialmente a la trayectoria	Cambios de orientación **relativo a la trayectoria** con - ORIPATH orientación de herramienta relativa a la trayectoria - ORIPATHS adicionalmente con acodamiento en el desarrollo de la orientación Programación del vector de giro - ORIROTC vector de giro tangencial, giro frente a la trayectoria tangente

6.2 Transformada de tres, cuatro y cinco ejes (TRAORI)

6.2.1 Relaciones generales, cabezal de herramienta cardánico

Función

Para conseguir condiciones de corte óptimas en el mecanizado de superficies curvadas en el espacio, el ángulo de ataque de la herramienta se tiene que poder modificar.

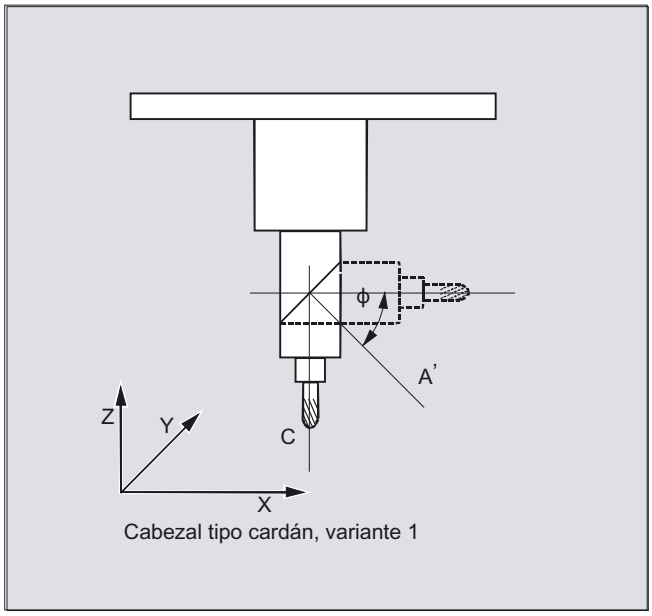


La configuración mecánica de la máquina así como sus parámetros elementales quedan almacenados en los datos de máquina de ejes.

Transformada de 5 ejes

Cabezal tipo cardan

En este caso, tres ejes lineales (X, Y, Z) y dos ejes de orientación (C, A) definen el punto de trabajo de la herramienta así como el ángulo de ataque. Uno de los dos ejes de orientación (en el ejemplo de la figura A') se define como eje inclinado, en la mayoría de los casos a 45°.

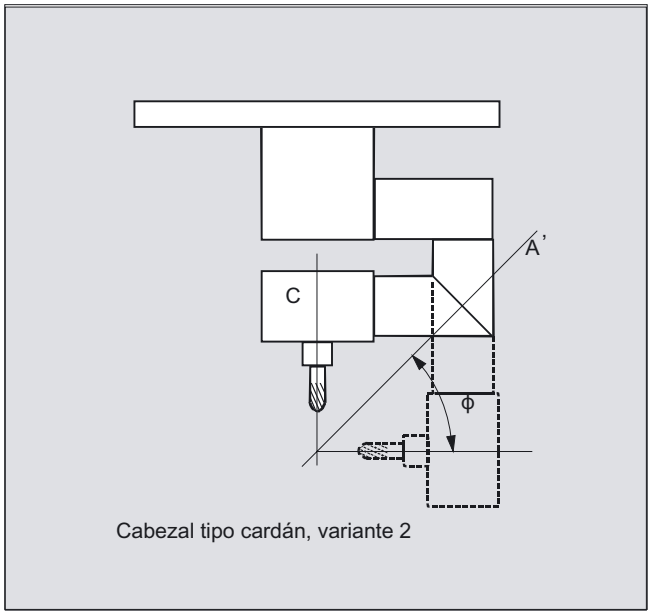


Cabezal tipo cardán, variante 1

En los ejemplos de las figuras se observa la asignación a la cinemática de la máquina para los ejes CA en el ejemplo del cabezal tipo cardán.

Fabricante de la máquina

La secuencia de movimiento para los ejes de orientación y la dirección para la orientación de la herramienta se pueden definir dependiendo de la cinemática de la máquina mediante datos de máquina.



Cabezal tipo cardán, variante 2

En este ejemplo, A' se sitúa en el ángulo φ frente al eje X

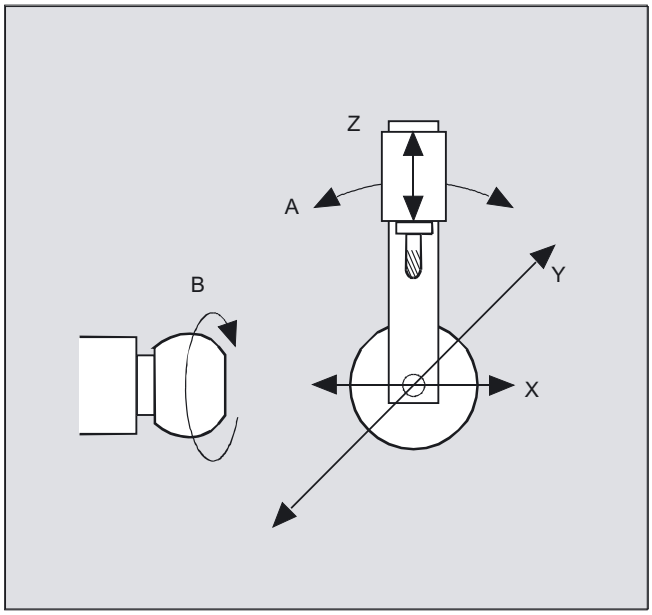
A nivel general rigen las siguientes relaciones:

A' se sitúa en el ángulo φ frente al	Eje X
B' se sitúa en el ángulo φ frente al	Eje Y
C' se sitúa en el ángulo φ frente al	Eje Z

El ángulo φ se puede parametrizar mediante datos de máquina en el rango entre 0° y +89°.

Con eje lineal basculable

En este caso se trata de una configuración en la cual se mueven tanto la herramienta como la pieza. La cinemática se compone de tres ejes lineales (X, Y, Z) y dos ejes giratorios a 90°. El primer eje giratorio se desplaza mediante el movimiento cruzado de dos ejes lineales; la herramienta se encuentra paralela al tercer eje lineal. El segundo eje giratorio gira la herramienta. El tercer eje lineal (eje basculante) se encuentra en el plano perpendicular formado por los otros dos ejes lineales.



La secuencia de movimiento para los ejes rotatorios y la dirección para la orientación de la herramienta se pueden definir dependiendo de la cinemática de la máquina mediante datos de máquina.

Se tienen las siguientes relaciones:

Ejes:	Secuencia de ejes:
1er eje giratorio	A A B B C C
2º eje giratorio	B C A C A B
Eje lineal basculando	Z Y Z X Y X

Para más explicaciones relacionadas con las secuencias de ejes configurables para la dirección de orientación de la herramienta, ver

Bibliografía: /FB3/ Manual de funciones especiales; Transformadas de 3 a 5 ejes (F2), apartado Cabezal portafresas cardánico, "Parametrización".

6.2.2 Transformada de tres, cuatro y cinco ejes (TRAORI)

Función

El usuario puede configurar dos o tres ejes de traslación y un eje de rotación. Las transformadas requieren que el eje de rotación sea ortogonal al plano de orientación.

La orientación de la herramienta únicamente es posible en el plano perpendicular al eje rotatorio. La transformada admite los tipos de máquina con herramienta móvil y pieza móvil.

La configuración y programación de las transformadas de tres y cuatro ejes son análogas a las de cinco ejes.

Bibliografía:
/FB3/ Manual de funciones especiales; Transformadas de 3 a 5 ejes (F2)

Programación

```
TRAORI(n)
o bien,
TRAORI(n,X,Y,Z,A,B)
o bien
TRAFOOF
```

Parámetros

`TRAORI`:	Activa la primera transformada de orientación declarada
`TRAORI(n)`:	Activa la transformada de orientación declarada con n
`n`:	Número de la transformada (n = 1 ó 2), TRAORI(1) corresponde a transformada de orientación CON
`X,Y,Z`:	Componente del vector de orientación en la que apunta la herramienta.
`A,B`:	Offset programable para ejes giratorios
`TRAFOOF`:	Desactivar transformada

Orientación de la herramienta

Según la dirección elegida para la orientación de la herramienta, en el programa CN debe ajustarse el plano de trabajo activo (G17, G18, G19) de tal modo que la corrección de la longitud de la herramienta actúe en esa dirección.

Nota

Tras activar la transformada, los datos de posición (X, Y, Z) se refieren siempre a la punta de la herramienta. Una modificación de las posiciones de los ejes giratorios que participan en la transformada produce los movimientos de compensación en los demás ejes de la máquina que son necesarios para mantener inalterable la posición de la punta de la herramienta.

La transformada de orientación siempre se ajusta desde la punta de la herramienta hasta el portaherramientas.

Ejemplo: Transformadas genéricas

La orientación básica de la herramienta apunta hacia:

`TRAORI(1,0,0,1)` dirección Z

`TRAORI(1,0,1,0)` dirección Y

`TRAORI(1,0,1,1)` dirección Y/Z (corresponde a la posición -45°)

Offset para ejes de orientación

Al activar la transformada de orientación puede programarse directamente un offset adicional para ejes de orientación.

Es posible omitir parámetros siempre que se siga el orden correcto en la programación.

Ejemplo

`TRAORI(, , , ,A,B)` si se tiene que introducir un único offset.

Como alternativa a la programación directa, puede aplicarse automáticamente este offset adicional para ejes de orientación desde el decalaje de origen activo en ese momento. Esta aplicación se configura a través de datos de máquina.

6.2.3 Variantes de la programación de la orientación y el estado inicial (ORIRESET)

Programación de la orientación de herramienta con TRAORI

Además de los ejes lineales X, Y, Z, en combinación con una transformada de orientación programable TRAORI pueden programarse posiciones de eje o ejes virtuales con ángulos o componentes vectoriales mediante los identificadores de ejes giratorios A..., B..., C.... Para ejes de orientación y de máquina existen distintos tipos de interpolación. Independientemente de los polinomios de orientación PO[ángulo] y de eje PO[eje] activos en cada momento, pueden estar programados distintos tipos de polinomio, p. ej.: G1, G2, G3, CIP o POLY.

La modificación de la orientación de la herramienta también se puede programar con vectores de orientación. La orientación final de cada secuencia se puede realizar mediante programación directa del vector o mediante la programación de las posiciones de ejes giratorios.

Nota

Variantes de la programación de la orientación en transformadas de tres a cinco ejes

En la transformada de tres a cinco ejes, las variantes

1. A, B, C indicación directa de las posiciones de ejes
2. A2, B2, C2 programación de ángulos de ejes virtuales mediante ángulos de Euler o RPY
3. A3 ,B3, C3 indicación de los componentes vectoriales
4. LEAD, TILT indicación de los ángulos de avance y laterales con relación a la trayectoria y la superficie
5. A4, B4, C4 y A5, B5, C5 vector normal a la superficie al inicio y al final de la secuencia
6. A6, B6, C6 y A7, B7, C7 interpolación del vector de orientación en una superficie de cono
7. A8, B8, C8 reorientación de la herramienta, dirección y longitud de trayectoria del movimiento de levantamiento

se excluyen mutuamente.

La programación mixta de valores se evita mediante avisos de alarma.

Estado inicial de la orientación de herramienta ORIRESET

Mediante la programación de ORIRESET(A, B, C) los ejes de orientación se desplazan de forma lineal y síncrona desde su posición instantánea hasta la posición de estado inicial indicada.

Si no se programa ninguna posición de estado inicial para un eje, se utiliza la posición definida en el correspondiente dato de máquina $MC_TRAFO5_ROT_AX_OFFSET_1/2. Los frames eventualmente activos de los ejes giratorios no se tienen en cuenta.

Nota

Sólo cuando está activa una transformada de orientación con TRAORI(...) se puede programar un estado inicial de la orientación de herramienta independientemente de la cinemática con ORIRESET(...) sin la alarma 14101.

Ejemplos

```
1er ejemplo para cinemática de máquina CA (nombres de eje de canal C, A)
ORIRESET(90, 45)       ;C a 90 grados, A a 45 grados
ORIRESET(, 30)         ;C a $MC_TRAFO5_ROT_AX_OFFSET_1/2[0], A a 30 grados
ORIRESET( )            ;C a $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                       ;A a $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]
2º ejemplo para cinemática de máquina CAC (nombres de eje de canal C, A, B)
ORIRESET(90, 45, 90)   ;C a 90 grados, A a 45 grados, B a 90 grados
ORIRESET( )            ;C a $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                       ;A a $MC_TRAFO5_ROT_AX_OFFSET_1/2[1],
                       ;B a $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]
```

Programación de los giros LEAD, TILT y THETA

Los giros de la orientación de herramienta se programan en la transformada de tres a cinco ejes con el ángulo de avance LEAD y el ángulo lateral TILT.

En una transformada con un tercer eje giratorio se permiten tanto para la orientación con componentes vectoriales como con indicación de los ángulos LEAD, TILT programaciones adicionales de C2 (giros del vector de orientación).

Con un tercer eje giratorio adicional, el giro de la herramienta sobre sí misma se puede programar con el ángulo de giro THETA.

6.2.4 Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT)

Función

Existen las siguientes posibilidades para programar la orientación de la herramienta:

1. Programación directa de las trayectorias de los ejes giratorios. El cambio en la orientación siempre se tiene en cuenta en el sistema de coordenadas base o en el sistema de coordenadas asociado a la máquina. Los ejes de orientación se desplazan como ejes síncronos.
2. Programación mediante ángulos de Euler o ángulos RPY según definiciones de ángulos mediante `A2, B2, C2`.
3. Programación del vector de dirección a través de `A3, B3, C3`. El vector de dirección apunta desde la punta de la herramienta en dirección al alojamiento de la herramienta.
4. Programación del vector normal a la superficie al inicio de la secuencia con `A4, B4, C4` y al final de la secuencia con `A5, B5, C5` (fresado frontal).
5. Programación a través del ángulo de avance `LEAD` y Ángulo lateral `TILT`.
6. Programación del eje giratorio del cono como vector normalizado a través de `A6, B6, C6` o de la orientación intermedia en la superficie del cono a través de `A7, B7, C7`, ver apartado "Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONxx)".
7. Programación de la reorientación, dirección y longitud del trayecto de la herramienta durante el movimiento de levantamiento mediante `A8, B8, C8`,
ver apartado "Alisado del desarrollo de la orientación (ORIPATHS A8=, B8=, C8=)"

Nota

En los tres casos, la programación de la orientación de la herramienta solamente es válida cuando está activada una transformada de orientación.

Ventaja: Estos programas son portables a cualquier cinemática de máquina.

Definición de la orientación de herramienta mediante código G

Nota

Fabricante de la máquina

Se pueden utilizar datos de máquina para seleccionar el cambio entre ángulos de Euler o ángulos RPY. Con los correspondientes ajustes de datos de máquina, es posible realizar un cambio tanto dependiente como independientemente del código G del grupo 50. Se puede escoger entre las siguientes posibilidades de ajuste:

1. Si ambos datos de máquina para la definición de los ejes y del ángulo de orientación mediante código G están puestos a cero:
   Los ángulos programados con `A2, B2, C2` se interpretan como ángulos de Euler o ángulos RPY **dependiendo del dato de máquina** Definición de ángulo de la programación de la orientación.
2. Cuando el dato de máquina para la definición de los ejes de orientación mediante código G está definido en uno, se efectúa un cambio
   dependiendo del código G activo del grupo 50:
   Los ángulos programados con`A2, B2, C2` se interpretan según uno de los códigos G activos `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS` y `ORIPY2`. Los valores programados con los ejes de orientación se interpretan también como ángulos de orientación según el código G activo del grupo 50.
3. Cuando el dato de máquina para la definición del ángulo de orientación mediante código G está definido en uno y el dato de máquina para la definición de los ejes de orientación mediante código G está definido en cero, se produce un cambio
   independiente del código G activo del grupo 50:
   Los ángulos programados con `A2, B2, C2` se interpretan según uno de los códigos G activos `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2 ORIAXPOS` y `ORIPY2`. Los valores programados con los ejes de orientación se interpretan siempre como posiciones de ejes giratorios independientemente del código G activo del grupo 50.

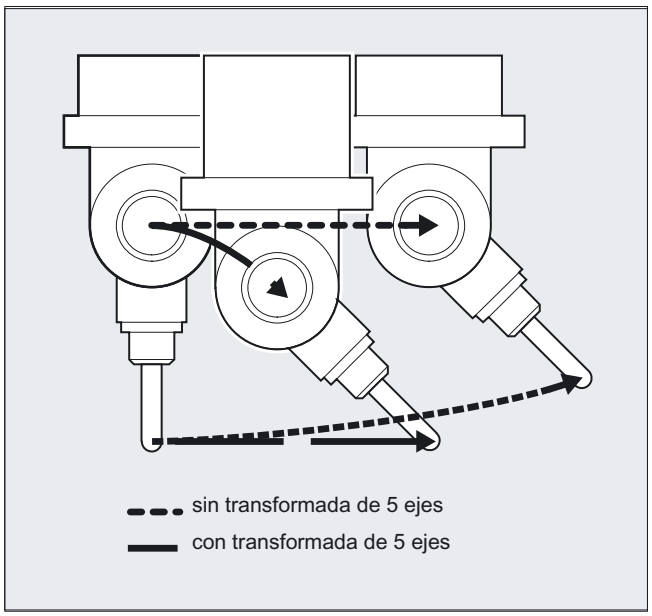
Programación

`G1 X Y Z A B C`	Programación del movimiento de los ejes giratorios
`G1 X Y Z A2= B2= C2=`	Programación con ángulos de Euler
`G1 X Y Z A3== B3== C3==`	Programación del vector de dirección
`G1 X Y Z A4== B4== C4==`	Programación del vector normal a la superficie al inicio de la secuencia
`G1 X Y Z A5== B5== C5==`	Programación del vector normal a la superficie al fin de la secuencia
`LEAD=`	Ángulo de avance para la programación de la orientación de la herramienta
`TILT=`	Ángulo lateral para la programación de la orientación de la herramienta

Parámetros

G....	Indicación del tipo de movimiento de los ejes giratorios
X Y Z	Indicación de los ejes lineales
A B C	Indicación de las posiciones de eje de máquina de los ejes giratorios
A2 B2 C2	Programación de ángulos (ángulos eulerianos o RPY) de ejes virtuales o de orientación
A3 B3 C3	Indicación de los componentes vectoriales del vector de dirección
A4 B4 C4	Indicación (p. ej.: en el fresado frontal) del componente del vector normal a la superficie al inicio de la secuencia
A5 B5 C5	Indicación (p. ej.: en el fresado frontal) del componente del vector normal a la superficie al final de la secuencia
LEAD	Ángulo relativo al vector normal a la superficie en el plano tendido por la tangente de trayectoria y el vector normal a la superficie
TILT	Ángulo en el plano, vertical respecto a la tangente de trayectoria, relativo respecto al vector normal a la superficie

Ejemplo: Confrontación sin y con transformada de 5 ejes



Descripción

Generalmente los programas de 5 ejes se generan en un sistema CAD/CAM (no se suelen introducir directamente desde el teclado del control numérico). Por lo tanto, la siguiente descripción está destinada sobre todo a los programadores de posprocesadores.

El tipo de programación de la orientación se determina en el código G del grupo 50:

`ORIEULER` mediante ángulo de Euler
`ORIRPY` mediante ángulo RPY (secuencia de giro ZYX)
`ORIVIRT1` mediante ejes de orientación virtuales (definición 1)
`ORIVIRT2` mediante ejes de orientación virtuales (definición 2)
`ORIAXPOS` mediante ejes de orientación virtuales con posiciones de ejes giratorios
`ORIPY2` mediante ángulos RPY (secuencia de giro XYZ)

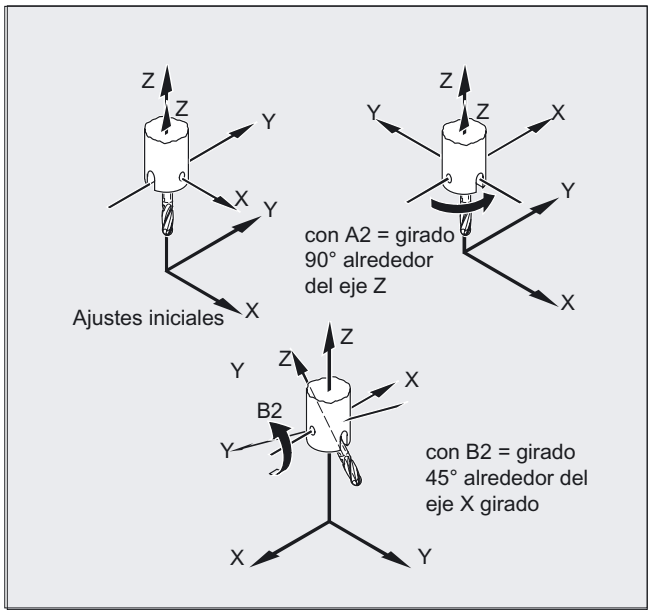
Fabricante de la máquina

El fabricante de la máquina puede definir distintas variantes mediante datos de máquina. Preste atención a las indicaciones del fabricante de la máquina.

Programación con ángulos de Euler ORIEULER

Los valores programados para la orientación mediante las direcciones `A2, B2, C2` se interpretan como ángulos de Euler (en grados).

El vector de orientación se genera girando un vector en primer lugar en la dirección Z, después con `A2` alrededor del eje Z, a continuación con `B2` alrededor del nuevo eje X, y por último con `C2` alrededor del nuevo eje Z.



En este caso, el valor de `C2` (giro alrededor del nuevo eje Z) es irrelevante y no necesita ser programado.

Programación en ángulos RPY ORIRPY

Los valores de orientación programados mediante `A2, B2, C2` se interpretan como si fuesen ángulos RPY en grados.

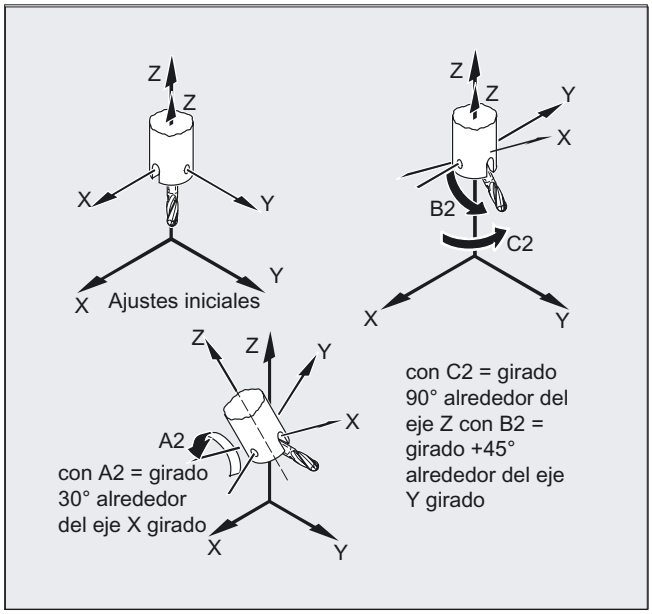
Nota

A diferencia de la programación con ángulos de Euler, en este caso influyen los tres valores en el vector de orientación.

Fabricante de la máquina

Para la definición de ángulos con ángulos de orientación RPY es válido para los ejes de orientación con $MC_ORI_DEF_WITH_G_CODE = 0

El vector de orientación resulta girando un vector en la dirección Z, primero con C2 alrededor del eje Z, después con B2 alrededor del nuevo eje Y y por último con A2 alrededor del nuevo eje X.

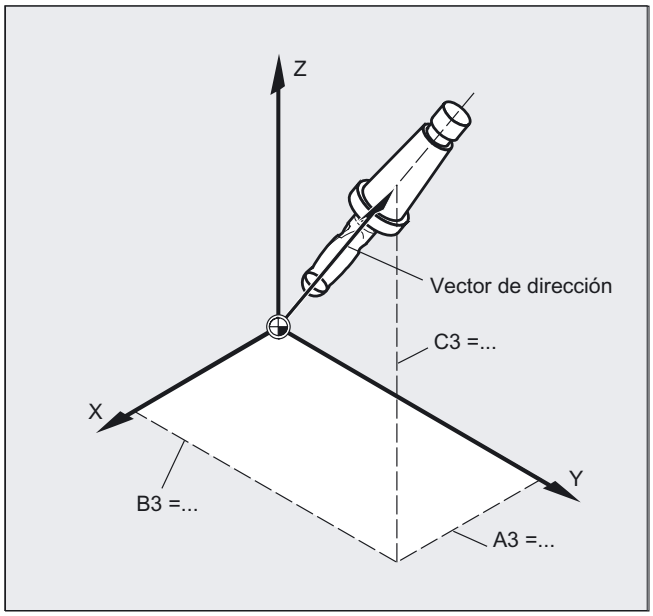


Si el dato de máquina para la definición de los ejes de orientación mediante código G es $MC_ORI_DEF_WITH_G_CODE = 1, entonces es válido:
El vector de orientación se genera girando un vector en primer lugar en la dirección Z, después con A2 alrededor del eje Z, a continuación con B2 alrededor del nuevo eje Y, y por último con C2 alrededor del nuevo eje X.

Programación del vector de dirección

Los componentes del vector de dirección se programan mediante `A3, B3, C3`. El vector indica la dirección de retirada de la herramienta, la longitud del vector carece por lo tanto de importancia.

Los componentes del vector no programados toman por defecto el valor 0.



Programación de la orientación de herramienta con LEAD= y TILT=

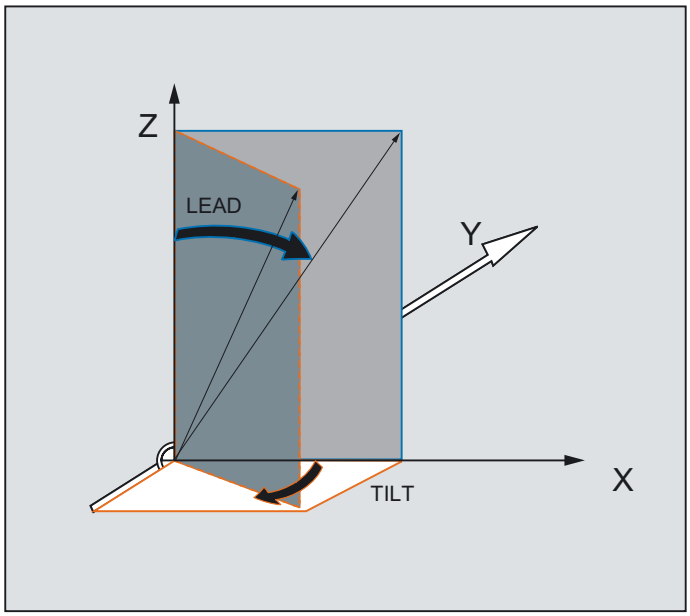
La orientación de herramienta resultante se averigua a partir de:

- Trayectoria tangente
- Vector normal a la superficie
  al inicio de la secuencia `A4, B4, C4` y al final de la secuencia `A5, B6, C5`
- Ángulo de avance `LEAD`
  en el plano abierto por la tangente a la trayectoria y el vector normal a la superficie
- Ángulo lateral `TILT` al final de la secuencia
  perpendicular a la tangente a la trayectoria y relativo al vector normal a la superficie

Comportamiento en esquinas interiores (con corrección de herramienta en 3D)

Cuando se acorta la secuencia en una esquina interior, la orientación de herramienta resultante se alcanza también al fin de secuencia.

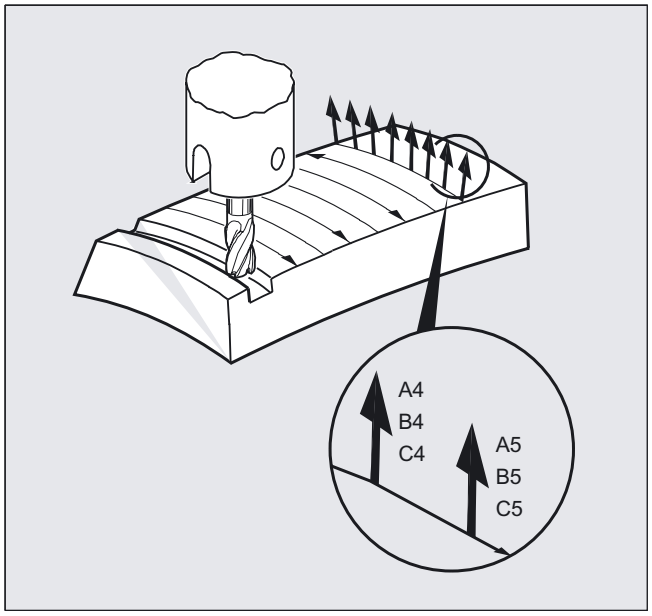
Definición de la orientación de herramienta con LEAD= y TILT=



6.2.5 Fresado frontal (fresado 3D A4, B4, C4, A5, B5, C5)

Funcionamiento

El fresado frontal sirve para mecanizar superficies con cualquier curvatura.



Para este tipo de fresado 3D se necesita una descripción línea a línea de las trayectorias 3D sobre la superficie de la pieza.

Los cálculos se realizan generalmente en sistemas CAM teniendo en cuenta la forma y las dimensiones de la herramienta. Las secuencias CN calculadas se cargan en el control numérico a través de posprocesadores.

Programación de la curvatura de la trayectoria

Descripción de las superficies

La descripción de la curvatura de la trayectoria se realiza mediante los vectores normales a la superficie con los siguientes componentes:

`A4, B4, C4` vector inicial al principio de la secuencia

`A5, B5, C5` vector final al final de la secuencia

Si en una secuencia solamente hay un vector inicial, el vector normal a la superficie permanece constante a lo largo de toda la secuencia. Si en una secuencia solamente se indica el vector final, se realiza una gran interpolación circular entre el valor final de la secuencia anterior y el valor final programado para la secuencia actual.

Si se programan los vectores inicial y final, entonces se realiza una interpolación circular mayor entre dichos vectores a lo largo de la secuencia. De esta manera es posible realizar desplazamientos suaves y uniformes a lo largo de la trayectoria generada en una secuencia.

En el ajuste básico los vectores normales a la superficie están orientados en la dirección Z, independientemente del plano activado (G17 hasta G19).

La longitud del vector carece de significado.

Los componentes no programados de los vectores tomarán el valor cero.

Cuando se encuentra activo ORIWKS, ver el apartado "Referencia de los ejes de orientación (ORIWKS, ORIMKS)"), los vectores normales a la superficie están referidos al frame activo y giran conjuntamente con las rotaciones asociadas al frame.

Fabricante de la máquina

El vector normal a la superficie deberá de encontrarse perpendicular a la tangente de trayectoria dentro de unos límites definidos mediante datos de máquina; en caso contrario se emite una alarma.

6.2.6 Referencia de los ejes de orientación (ORIWKS, ORIMKS)

Función

En la programación de la orientación en el sistema de coordenadas de pieza a través de

- ángulos eulerianos o RPY o
- el vector de orientación

el movimiento de giro se puede ajustar mediante ORIMKS/ORIWKS.

Nota

Fabricante de la máquina

El tipo de interpolación para la orientación se define con el dato de máquina:

DM21104 $MC_ORI_IPO_WITH_G_CODE

= FALSE: la referencia son las funciones G ORIWKS y ORIMKS

= TRUE: la referencia son las funciones G del grupo 51 (ORIAXES, ORIVECT, ORIPLANE...)

Sintaxis

```
ORIMKS=...
ORIWKS=...
```

Descripción

ORIMKS	Giro en el sistema de coordenadas de máquina
ORIWKS	Giro en el sistema de coordenadas de pieza

Nota

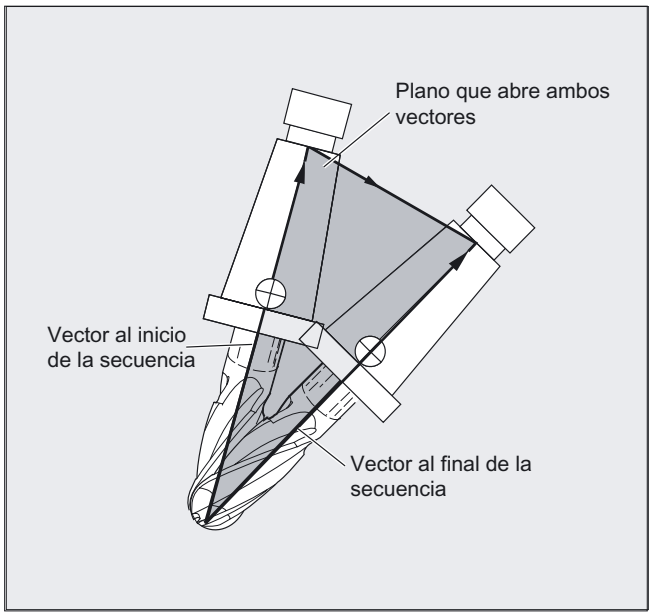
ORIWKS es el ajuste por defecto. Si a la hora de realizar una programación con cinco ejes no se ha definido para qué máquina se va a procesar el programa, se debe seleccionar ORIWKS. Qué movimientos sean realmente realizados por la máquina depende de la cinemática de ésta.

Con ORIMKS se pueden programar desplazamientos de máquina reales, por ejemplo, para evitar colisiones con dispositivos o similares.

Descripción

Con ORIMKS, el movimiento ejecutado por la herramienta **depende** de la cinemática de la máquina. Los cambios de orientación de la herramienta sin desplazamiento de la punta de la misma se realizan mediante una interpolación lineal entre las posiciones de los ejes giratorios.

Con ORIWKS, el desplazamiento de la herramienta es **independiente** de la cinemática de la máquina. Un cambio en la orientación de la herramienta sin desplazamiento de la punta de la misma provoca un desplazamiento de la herramienta en el plano creado entre los vectores inicial y final.



Posiciones singulares

Nota

ORIWKS

Los desplazamientos para la orientación de la herramienta en la zona de posición singular en la máquina de cinco ejes requieren grandes desplazamientos de los ejes de máquina. (P. ej.: con un cabezal orientable con eje C como eje giratorio y eje A como eje de orientación, todas las posiciones con A=0 son singulares).

Fabricante de la máquina

Para no sobrecargar los ejes de máquina, se produce una disminución sustancial de la velocidad contorneo en la proximidad de los puntos singulares.

Con los datos de máquina

`$MC_TRAFO5_NON_POLE_LIMIT`

`$MC_TRAFO5_POLE_LIMIT`

es posible parametrizar la transformada de forma que los desplazamientos para la orientación en las proximidades del polo se ejecuten atravesando el mismo, posibilitando así un mecanizado rápido.

Las posiciones singulares sólo se tratan con el DM `$MC_TRAFO5_POLE_LIMIT`.

Bibliografía:
/FB3/ Manual de funciones especiales; Transformada de 3 a 5 ejes (F2),
apartado "Posiciones singulares y su tratamiento".

6.2.7 Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

Función

La función Ejes de orientación describe la orientación de la herramienta en el espacio y se consigue mediante la programación de las correcciones para los ejes giratorios. Un tercer grado de libertad adicional se consigue mediante el giro adicional de la herramienta alrededor de sí misma. Esta orientación de la herramienta se realiza libremente en el espacio a través de un tercer eje giratorio y exige la transformada de seis ejes. El giro propio de la herramienta alrededor de sí misma se define en función del tipo de interpolación de los vectores de giro con el ángulo de giro THETA; ver el apartado "Giros de la orientación de la herramienta (ORIROTA/TR/TT, ORIROTC, THETA)".

Programación

Los ejes de orientación se programan a través de los descriptores de ejes A2, B2, C2.

`N... ORIAXES u ORIVECT`	Interpolación lineal o interpolación
`N... G1 X Y Z A B C`	circular de gran radio
o bien,	o bien,
`N... ORIPLANE`	Interpolación de orientación del plano
o bien,	o bien,
`N... ORIEULER u ORIRPY o bien ORIRPY2`	Ángulo de orientación Euler/RPY
`N... G1 X Y Z A2= B2= C2=`	Programación de ángulos de ejes virtuales
o bien,	o bien,
`N... ORIVIRT1 u ORIVIRT2`	Ejes de orientación virtuales definición
`N... G1 X Y Z A3= B3= C3=`	1 ó 2 Programación de vectores de dirección

Para cambios de orientación a lo largo de una superficie de cono situada en el espacio se pueden programar otras correcciones de eje giratorio de los ejes de orientación, ver el apartado "Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONxx).

Parámetros

ORIAXES	Interpolación lineal de los ejes de máquina o ejes de orientación
ORIVECT	Interpolación circular de gran radio (idéntico a ORIPLANE)
ORIMKS	Giro en el sistema de coordenadas de máquina
ORIWKS	Giro en el sistema de coordenadas de pieza
	Descripción, ver aptdo. Giros de la orientación de la herramienta
A= B= C=	Programación de la posición de eje de la máquina
ORIEULER	Programación de la orientación mediante ángulos eulerianos
ORIRPY	Programación de la orientación con ángulos RPY. La secuencia de giro es XYZ, donde: A2 es el ángulo de giro alrededor de X B2 es el ángulo de giro alrededor de Y C2 es el ángulo de giro alrededor de Z
ORIRPY2	Programación de la orientación con ángulos RPY. La secuencia de giro es ZYX, donde: A2 es el ángulo de giro alrededor de Z B2 es el ángulo de giro alrededor de Y C2 es el ángulo de giro alrededor de X
A2= B2= C2=	Programación de ángulos de ejes virtuales
ORIVIRT1 ORIVIRT2	Programación de la orientación de la herramienta con ejes de orientación virtuales
	(definición 1), definición según DM $MC_ORIAX_TURN_TAB_1
	(definición 2), definición según DM $MC_ORIAX_TURN_TAB_2
A3= B3= C3=	Programación del vector de dirección del eje de dirección

Descripción

Fabricante de la máquina

Con el DM $MC_ORI_DEF_WITH_G_CODE se establece el modo en que se definirán los ángulos programados `A2, B2, C2`:

La definición se efectúa de acuerdo con el DM $MC_ORIENTATION_IS_EULER (estándar) o de acuerdo con el grupo G 50 (`ORIEULER`, `ORIRPY`, `ORIVIRT1`, `ORIVIRT2`).

Con el DM $MC_ORI_IPO_WITH_G_CODE se define el tipo de interpolación activo: `ORIWKS/ORIMKS` u `ORIAXES/ORIVECT`.

Modo de operación JOG

En este modo de operación siempre se interpolan linealmente los ángulos de orientación. Durante el desplazamiento continuo e incremental mediante las teclas de desplazamiento sólo puede desplazarse un eje de orientación. Con los volantes electrónicos se pueden desplazar los ejes de orientación simultáneamente.

Para el desplazamiento manual de ejes de orientación actúa el interruptor de corrección del avance específico del canal, o bien, el interruptor de corrección del rápido en caso de superposición del rápido.

Con los siguientes datos de máquina se puede predefinir una velocidad separada:

$MC_JOG_VELO_RAPID_GEO

$MC_JOG_VELO_GEO

$MC_JOG_VELO_RAPID_ORI

$MC_JOG_VELO_ORI

Nota

SINUMERIK 840D con "Paquete de transformación de manejo (Handling)"

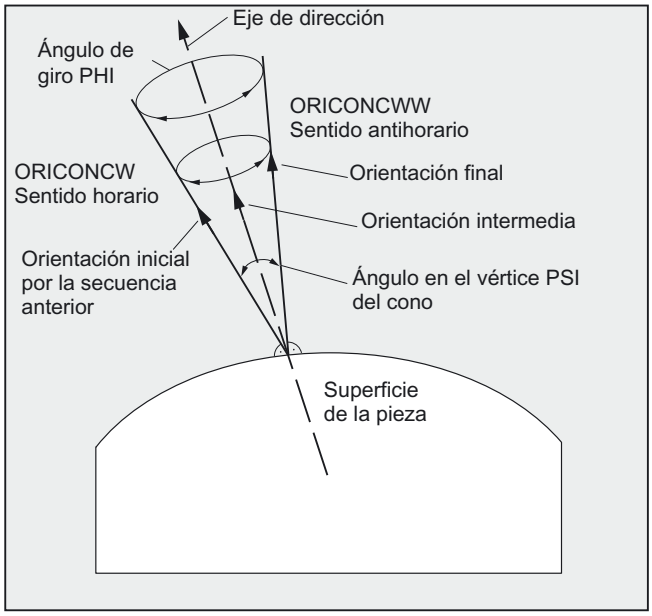
Con la función "Desplazamiento manual cartesiano" se puede regular por separado en el modo JOG la traslación de ejes geométricos en los sistemas de referencia MKS, WKS y TKS.

Bibliografía:
/FB2/ Manual de funciones de ampliación; Transformada cinemática (M1)

6.2.8 Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

Función

Con la orientación ampliada es posible ejecutar cambios de orientación a lo largo de una superficie de cono situada en el espacio. La interpolación del vector de orientación en una superficie de cono se realiza con los comandos modales ORICONxx. Para la interpolación en un plano, la orientación final se puede programar con ORIPLANE. A nivel general, la orientación inicial es definida por las secuencias anteriores.



Programación

La orientación final se define indicando la programación de ángulos en ángulos Euler o RPY con A2, B2, C2 o mediante la programación de las posiciones de eje giratorio A, B, C. Para los ejes de orientación a lo largo de la superficie de cono se necesitan datos de programación adicionales:

- Eje giratorio del cono como vector con `A6, B6, C6`
- Ángulo en el vértice `PSI` con el identificador `NUT`
- Orientación intermedia en la superficie de cono con `A7, B7, C7`

Nota

Programación del vector de dirección A6, B6, C6 para el eje de giro del cono

No es absolutamente necesario programar una orientación final. Si no se indica ninguna orientación final, se interpola una superficie de cono completa con 360 grados.

Programación del ángulo en el vértice del cono con NUT=ángulo

Es absolutamente necesario indicar una orientación final.

De este modo no se puede interpolar una superficie de cono completa con 360 grados.

Programación de la orientación intermedia A7, B7, C7 en la superficie del cono

Es absolutamente necesario indicar una orientación final. El cambio de orientación y el sentido de giro se definen de forma unívoca por los tres sectores Orientación inicial, final e intermedia. Los tres vectores tienen que ser distintos. Si la orientación intermedia programada es paralela a la orientación inicial o final, se ejecuta una interpolación circular de gran radio lineal de la orientación en el plano abierto por los vectores inicial y final.

Interpolación de orientación ampliada en una superficie de cono

```
N... ORICONCW u ORICONCCW

N... A6= B6= C6= A3= B3= C3=
o bien
N... ORICONTO
N... G1 X Y Z A6= B6= C6=

o bien
N... ORICONIO
N... G1 X Y Z A7= B7= C7=
N... PO[PHI]=(a2, a3, a4, a5)
N... PO[PSI]=(b2, b3, b4, b5)
```

Interpolación en una superficie de cono con

vector de dirección en sentido horario/ antihorario del cono y orientación final o

transición tangencial e indicación de la orientación final

o

indicación de la orientación final y una orientación intermedia en la superficie del cono con

polinomios para el ángulo de giro y polinomios para el ángulo en el vértice

Parámetros

ORIPLANE	Interpolación en el plano (interpolación circular de gran radio)
ORICONCW	Interpolación en una superficie de cono en sentido horario
ORICONCCW	Interpolación en una superficie de cono en sentido antihorario
ORICONTO	Interpolación en una superficie envolvente de cono con transición tangencial
A6= B6= C6=	Programación del eje giratorio del cono (vector normalizado)
NUT=ángulo	Ángulo en el vértice del cono en grados
NUT=+179	Ángulo de desplazamiento menor o igual a 180 grados
NUT=-181	Ángulo de desplazamiento mayor o igual a 180 grados
ORICONIO	Interpolación en una superficie de cono
A7= B7= C7=	Orientación intermedia (programación como vector normalizado)
PHI	Ángulo de giro de la orientación alrededor del eje de dirección del cono
PSI	Ángulo en el vértice del cono
Polinomios posibles PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	Además de los correspondientes ángulos también se pueden programar polinomios de máx. 5º grado

Ejemplo de distintos cambios de orientación

```
…
N10 G1 X0 Y0 F5000
N20 TRAORI(1)                          ; Transformada de orientación CON.
N30 ORIVECT                            ; Interpolar orientación de herramienta
                                         como vector.
…                                      ; Orientación de herramienta en el plano.
N40 ORIPLANE                           ; Seleccionar interpolación circular de
                                         gran radio.
N50 A3=0 B3=0 C3=1
N60 A3=0 B3=1 C3=1                     ; Orientación en el plano Y/Z girada
                                         45 grados; a final de secuencia se
                                         alcanza la orientación (0,1/√2,1/√2).
…
N70 ORICONCW                           ; Programación de orientación en la
                                         superficie de cono:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1      ; El vector de orientación se interpola
                                         en una superficie de cono con la
                                         dirección (0,0,1) hasta la orientación
                                         (1/√2,0,1/√2) en sentido horario; el
                                         ángulo de giro es de 270 grados.
N90 A6=0 B6=0 C6=1                     ; La orientación de herramienta describe
                                         una vuelta completa en la misma
                                         superficie de cono.
```

Descripción

Para describir cambios de orientación en una superficie de cono situada libremente en el espacio, se tiene que conocer el vector alrededor del cual se quiere girar la orientación de herramienta. Además se tiene que especificar la orientación inicial y final. La orientación estándar resulta de la secuencia anterior y la orientación final se tiene que programar o definir a través de otras condiciones.

Programación en el plano ORIPLANE equivale a ORIVECT

La programación de la interpolación circular de gran radio junto con polinomios angulares corresponde a la interpolación lineal y de polinomio de contornos. La orientación de herramienta se interpola un plano abierto por la orientación inicial y final. Si se programan adicionalmente polinomios, el vector de orientación también se puede volcar fuera del plano.

Programación de círculos en un plano G2/G3, CIP y CT

La orientación ampliada corresponde a la interpolación de círculos en un plano. Acerca de las posibilidades de programación correspondientes con indicación del centro o del radio como G2/G3, círculo con punto intermedio CIP arcos tangenciales CT, ver

Bibliografía: Manual de programación Fundamentos, "Programar instrucciones de desplazamiento".

Programación de la orientación de la herramienta

Interpolación del vector de orientación en una superficie de cono ORICONxx

Para la interpolación de orientaciones en una superficie de cono se pueden seleccionar cuatro tipos de interpolación distintos del grupo de códigos G 51:

1. Interpolación en una superficie de cono en sentido horario `ORICONCW` con indicación de la orientación final y la dirección del cono o del ángulo en vértice. El vector de dirección se programa con los descriptores `A6, B6, C6` y el ángulo en vértice del cono con el descriptor NUT= margen de valores en un intervalo de 0 a 180 grados.
2. Interpolación en una superficie de cono en sentido antihorario `ORICONCWW` con indicación de la orientación final y la dirección del cono o del ángulo en vértice. El vector de dirección se programa con los descriptores `A6, B6, C6` y el ángulo en vértice del cono con el descriptor NUT= margen de valores en un intervalo de 0 a 180 grados.
3. Interpolación en una superficie de cono `ORICONIO` con indicación de la orientación final y una orientación intermedia que se programa con los descriptores `A7, B7, C7`.
4. Interpolación en una superficie de cono `ORICONTO` con transición tangencial e indicación de la orientación final. El vector de dirección se programa con los descriptores `A6, B6, C6`.

6.2.9 Especificación de orientación de dos puntos de contacto (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Función

Programación del cambio de orientación a través de la segunda curva espacial ORICURVE

Otra posibilidad de programar cambios de orientación consiste en programar, además de la punta de herramienta a lo largo de una curva espacial, también el desplazamiento de un segundo punto de contacto de la herramienta con `ORICURVE`. Esto permite definir de forma unívoca los cambios de orientación de la herramienta, como en la programación del mismo vector de herramienta.

Fabricante de la máquina

Observe las indicaciones del fabricante de la máquina acerca de descriptores de eje ajustables a través de datos de máquina para la programación de la 2ª trayectoria de orientación de la herramienta.

Programación

En este tipo de interpolación se pueden programar, para las dos curvas espaciales, puntos con `G1` o polinomios con `POLY`. No se admiten círculos ni evolutas. Adicionalmente se puede activar una interpolación spline con `BSPLINE` y la función "Resumen de secuencias spline breves".

Bibliografía:
/FB1/ Manual de funciones básicas; Trabajo con control de contorneado, Parada precisa, Look Ahead (B1), apartado: Resumen de secuencias spline breves

No se permiten los demás tipos de spline `ASPLINE` y `CSPLINE`, así como la activación de un compresor con `COMPON`, `COMPCURV` o `COMPCAD`.

El desplazamiento de los dos puntos de contacto de la herramienta se puede especificar en la programación de los polinomios de orientación para coordenadas de máx. 5º grado.

Interpolación de orientación ampliada con curva espacial adicional y polinomios para coordenadas

```
N... ORICURVE
N… PO[XH]=(xe, x2, x3, x4, x5)
N… PO[YH]=(ye, y2, y3, y4, y5)
N… PO[ZH]=(ze, z2, z3, z4, z5)
```

Indicación del desplazamiento del segundo punto de contacto de la herramienta y polinomios adicionales de las coordenadas correspondientes

Parámetros

ORICURVE	Interpolación de la orientación con especificación del movimiento de dos puntos de contacto de la herramienta.
XH YH ZH	Descriptor de las coordenadas del segundo punto de contacto de la herramienta del contorno adicional como curva espacial
Polinomios posibles PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)	Además del correspondiente punto final se pueden programar curvas espaciales adicionales con polinomios.
xe, ye, ze	Puntos finales de la curva espacial
xi, yi, zi	Coeficientes de los polinomios de máx. 5º grado

Nota

Descriptor XH YH ZH para la programación de una 2ª trayectoria de orientación

Los descriptores se tienen que elegir de modo que no se produzca ningún conflicto con otros descriptores de los ejes lineales

ejes X Y Z

y ejes giratorios como

A2 B2 C2 ángulo de Euler o RPY

A3 B3 C3 vectores de dirección

A4 B4 C4 ó A5 B5 C5 vectores normales a la superficie

A6 B6 C6 vectores de giro o A7 B7 C7 coordenadas de punto intermedio

u otros parámetros de interpolación.

6.3 Polinomios de orientación (PO[ángulo], PO[coordenada])

Función

Independientemente de la interpolación de polinomio del grupo de códigos 1 que se encuentra activa en este momento, se pueden programar distintos tipos de polinomios de orientación de máx. 5º grado en una transformada de tres a cinco ejes.

1. Polinomios para **ángulo:** Ángulo de avance LEAD, ángulo lateral TILT relativo al plano abierto por la orientación inicial y final.
2. Polinomios para **coordenadas:** XH, YH, ZH de la segunda curva espacial para la orientación de un punto de referencia en la herramienta.

En una transformada de seis ejes el posible programar, para la orientación de herramienta, adicionalmente el giro del vector de giro THT con polinomios de máx. 5º grado para giros de la misma herramienta.

Sintaxis

Polinomios de orientación del tipo 1 para **ángulo**

`N… PO[PHI]=(a2, a3, a4, a5)`	Transformadas de tres a cinco ejes
`N… PO[PSI]=(b2, b3, b4, b5)`	Transformadas de tres a cinco ejes

Polinomios de orientación del tipo 2 para **coordenadas**

`N… PO[XH]=(xe, x2, x3, x4, x5)`
`N… PO[YH]=(ye, y2, y3, y4, y5)`
`N… PO[ZH]=(ze, z2, z3, z4, z5)`
	Descriptor para las coordenadas de la segunda trayectoria de orientación para la orientación de herramienta

Adicionalmente, se puede programar en ambos casos un polinomio para el **giro** en la transformada de seis ejes con

`N… PO[THT]=(c2, c3, c4, c5)`	Interpolación relativa a la trayectoria del giro
o	
`N… PO[THT]=(d2, d3, d4, d5)`	Interpolación absoluta, relativa y tangencial para el cambio de orientación

del vector de orientación. Esto es posible si la transformada soporta un vector de giro con un decalaje programable e interpolable con el ángulo de giro THETA.

Significado

PO[PHI]	Ángulo en el plano entre la orientación inicial y final
PO[PSI]	Ángulo del plano entre orientación inicial y final
PO[THT]	Ángulo de giro del vector en código G del grupo 54 programado con THETA
PHI	Ángulo de avance LEAD
PSI	Ángulo lateral TILT
THETA	Giro alrededor de la dirección de herramienta en Z
PO[XH]	Coordenada X del punto de referencia en la herramienta
PO[YH]	Coordenada Y del punto de referencia en la herramienta
PO[ZH]	Coordenada Z del punto de referencia en la herramienta

Descripción

No se pueden programar polinomios de orientación

- si las interpolaciones spline ASPLINE, BSPLINE, CSPLINE están activas.
  Polinomios del tipo 1 para ángulos de orientación son posibles para cada tipo de interpolación, excepto spline, es decir, en la interpolación lineal con marcha rápida G00 o con avance G01 en la interpolación de polinomios con POLY y en la interpolación circular y de evoluta con G02, G03, CIP, CT, INVCW y INCCCW.
  En cambio, los polinomios del tipo 2 para coordenadas de orientación sólo son posibles si está activa la interpolación lineal con marcha rápida G00 o con avance G01 o la interpolación de polinomios con POLY.
- si la orientación se interpola mediante interpolación de ejes ORIAXES. En este caso se pueden programar directamente polinomios con PO[A] y PO[B] para los ejes de orientación A y B.

Polinomios de orientación del tipo 1 con ORIVECT, ORIPLANE y ORICONxx

En la interpolación circular de gran radio y la interpolación de superficie de cono con ORIVECT, ORIPLANE y ORICONxx sólo son posibles polinomios de interpolación del tipo 1.

Polinomios de orientación del tipo 2 con ORICURVE

Si la interpolación con curva espacial adicional ORICURVE está activa, se interpolan los componentes cartesianos del vector de orientación y sólo son posibles polinomios de interpolación del tipo 2.

6.4 Giros de la orientación de la herramienta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)

Función

Si, en tipos de máquina con herramienta móvil, se tiene que poder cambiar también la orientación de la herramienta, se programa cada secuencia con una orientación final. En función de la cinemática de la máquina se pueden programar el sentido de orientación de los ejes de orientación o el sentido de giro del vector de orientación THETA. Para estos vectores de giro se pueden programar distintos tipos de interpolación:

- ORIROTA: Ángulo de rotación frente a un sentido de giro absoluto especificado.
- ORIROTR: Ángulo de rotación relativo al plano entre la orientación inicial y final.
- ORIROTT: Ángulo de rotación relativa a la modificación del vector de orientación.
- ORIROTC: Ángulo de giro tangencial a la trayectoria tangente

Sintaxis

Sólo si está activo el tipo de interpolación `ORIROTA`, el ángulo de giro o el vector de giro se puede programar como sigue en los cuatro tipos posibles:

1. Directamente las posiciones de eje giratorio `A, B, C`
2. Ángulo de Euler (en grados) a través de `A2, B2, C2`
3. Ángulo RPY (en grados) a través de `A2, B2, C2`
4. Vector de dirección a través de `A3, B3, C3` (ángulo de giro mediante `THETA=<valor>`).

Si `ORIOTR` u `ORIOTT` están activos, el ángulo de giro ya sólo se puede programar directamente con `THETA`.

Un giro también se puede programar por sí solo en una secuencia sin que se produzca un cambio de orientación. Entonces,`ORIROTR` y `ORIROTT` no tienen ningún significado. En este caso, el ángulo de giro se interpreta siempre con relación a la dirección absoluta (`ORIROTA`).

`N... ORIROTA`
`N... ORIROTR`
`N... ORIROTT`
`N... ORIROTC`
	Definir interpolación del vector de giro

`N... A3= B3= C3= THETA=valor`	Definir el giro del vector de orientación
`N... PO[THT]=(`d_2`, `d_3`, `d_4`, `d_5`)`	Interpolar ángulo de giro con polinomio de 5º grado

Descripción

ORIROTA	Ángulo de rotación frente a un sentido de giro absoluto especificado
ORIROTR	Ángulo de rotación relativo al plano entre la orientación inicial y final
ORIROTT	Ángulo de giro como vector de giro tangencial relativo al cambio de orientación
ORIROTC	Ángulo de giro como vector de giro tangencial relativo a la tangente de trayectoria
THETA	Giro del vector de orientación
THETA=valor	Ángulo de rotación en grados que se alcanza al final de la secuencia
THETA=Θe	Ángulo de rotación con el ángulo final Θ_e del vector de giro
THETA=AC(…)	Conmutación por secuencias al acotado absoluto
THETA=AC(…)	Conmutación por secuencias al acotado incremental
Θe	Ángulo final del vector de giro, tanto absoluto con G90 como relativo con G91 (acotado incremental), está activo
PO[THT]=(...)	Polinomio para el ángulo de giro

Ejemplo: giros de las orientaciones

Código del programa	**Comentarios**
N10 TRAORI	; Activar transformada de orientación
N20 G1 X0 Y0 Z0 F5000	; Orientación de la herramienta
N30 A3=0 B3=0 C3=1 THETA=0	; en dirección Z con el ángulo de giro 0
N40 A3=1 B3=0 C3=0 THETA=90	; en dirección X con giro de 90 grados
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	; Orientación
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; en dirección Y con giro de 180 grados
N70 ORIROTT	; se mantiene constante y giro de 90 grados
N80 A3=1 B3=0 C3=0 THETA=30	; Ángulo de giro relativo al cambio de orientación
	; Vector de giro en un ángulo de 30 grados al plano X-Y

En la interpolación de la secuencia
N40, el ángulo de giro se interpola del valor inicial de 0 grados al valor final de 90 grados de forma lineal. En la secuencia N50, el ángulo de giro cambia de 90 grados a 180 grados según la parábola $\Theta(u) = +90u^2$. En N60 también se puede ejecutar un giro sin que se produzca un cambio de orientación.

En N80, la orientación de herramienta gira de la dirección Y a la dirección X. El cambio de orientación se sitúa en el plano X-Y y el vector de giro forma un ángulo de 30 grados frente a dicho plano.

Descripción

ORIROTA

El ángulo de rotación `THETA` se interpola con relación a una dirección absoluta establecida en el espacio. El sentido de giro básico se establece a través de datos de máquina.

ORIROTR

El ángulo de rotación `THETA` se interpreta con relación al plano abierto por la orientación inicial y final.

ORIROTT

El ángulo de rotación `THETA` se interpreta con relación al cambio de orientación. Para `THETA=0`, el vector de giro se interpola tangencialmente al cambio de orientación y se distingue únicamente de `ORIROTR` si, para la orientación, se ha programado al menos un polinomio para el "ángulo de inclinación PSI“. De este modo se obtiene un cambio de orientación que no se desarrolla en el plano. Mediante un ángulo de giro `THETA` programado adicionalmente se puede, por ejemplo, interpolar el vector de giro de tal modo que forma siempre un determinado valor para el cambio de orientación.

ORIROTC

El vector de giro se interpola relativamente a la tangente de trayectoria con una corrección programable a través del ángulo `THETA`. Para el ángulo de decalaje también se puede programar un polinomio `PO[THT]=(c2, c3, c4, c5)` de máx. 5º grado.

6.5 Orientaciones relativas a la trayectoria

6.5.1 Tipos de orientación relativos a la trayectoria

Función

Con esta función ampliada, la orientación relativa no se alcanza sólo al final de la secuencia, sino a lo largo de toda la trayectoria. La orientación alcanzada en la secuencia anterior se traslada mediante interpolación circular de gran radio a la orientación final programada. Básicamente existen dos posibilidades de programar la orientación deseada relativa a la trayectoria:

1. La orientación de herramienta, así como el giro de la herramienta se interpolan con ORIPATH, ORPATHTS con relación a la trayectoria.
2. El vector de orientación se programa e interpola como de costumbre. Con ORIROTC, el giro del vector de orientación se aplica relativo a la tangente de trayectoria.

Sintaxis

El tipo de interpolación de la orientación y del giro de la herramienta se programa con:

`N... ORIPATH`	Orientación relativa a la trayectoria
`N... ORIPATHS`	Orientación relativa a la trayectoria con alisado del desarrollo de la orientación
`N... ORIROTC`	Interpolación del vector de giro relativa a la trayectoria

Un acodamiento de la orientación producida por un ángulo en el desarrollo de la trayectoria se puede alisar con `ORIPATHS`. La dirección y la longitud del trayecto del movimiento de levantamiento se programan a través del vector con los componentes `A8=X, B8=Y C8=Z` .

Con `ORIPATH/ORIPATHS` se pueden programar distintas referencias a la tangente de trayectoria a través de los tres ángulos

- `LEAD`= indicación ángulo de avance relativo a la trayectoria y la superficie
- `TILT`= indicación ángulo lateral relativo a la trayectoria y la superficie
- `THETA`= ángulo de giro

para toda la trayectoria.

Para el ángulo de giro `THETA` se pueden programar con `PO[THT]=(...)` polinomios adicionales de máx. 5º grado.

Nota

Fabricante de la máquina

Preste atención a las indicaciones del fabricante de la máquina. A través de datos de máquina y de operador configurables se pueden realizar otros ajustes para el tipo de orientación relativo a la trayectoria. Para más explicaciones, ver

Bibliografía:
/FB3/ Manual de funciones especiales; Transformada de 3 a 5 ejes (F2),
apartado "Orientación"

Descripción

La interpolación de los ángulos `LEAD` y `TILT` se puede ajustar de distintas maneras a través de datos de máquina:

- La referencia de la orientación de herramienta programada con `LEAD` y `TILT` se mantiene a lo largo de toda la secuencia.
- Ángulo de avance `LEAD`: Giro en la dirección perpendicular a la tangente y al vector normal `TILT`: Giro de la orientación alrededor del vector normal.
- Ángulo de avance `LEAD`: Giro en la dirección perpendicular a la tangente y al vector normal ángulo lateral `TILT`: Giro de la orientación alrededor de la dirección de la tangente de trayectoria.
- Ángulo de giro `THETA`: Giro de la herramienta alrededor de sí misma con un tercer eje giratorio adicional como eje de orientación en la transformada de seis ejes.

Nota

Orientación relativa a la trayectoria no permitida con OSC, OSS, OSSE, OSD, OST

La interpolación de orientación relativa a la trayectoria `ORIPATH` u `ORIPATHS` y `ORIOTC` no se puede programar junto con el alisamiento del desarrollo de la orientación con uno de los códigos G del grupo 34. Para este fin, OSOF tiene que estar activo.

6.5.2 Giro relativo a la trayectoria de la orientación de herramienta (ORIPATH, ORIPATHS, ángulo de giro)

Función

En una transformada de seis ejes, para una orientación libre de la herramienta en el espacio, la herramienta también se puede girar alrededor de sí misma con un tercer eje giratorio. En caso de giro relativo a la trayectoria de la orientación de herramienta con ORIPATH u ORIPATHS, el giro adicional se puede programar a través del ángulo de giro THETA. Como alternativa se pueden programar los ángulos LEAD y TILT a través de un sector en el plano perpendicular a la dirección de herramienta.

Fabricante de la máquina

Preste atención a las indicaciones del fabricante de la máquina. A través de un dato de máquina se puede ajustar la interpolación de los ángulos LEAD y TILT de distintas maneras.

Sintaxis

Giro de la orientación de la herramienta y de la herramienta

El tipo de orientación de herramienta relativo a la trayectoria se activa con ORIPATH o ORIPATHS.

`N... ORIPATH`	Activar tipo de orientación relativa a la trayectoria
`N... ORIPATHS`	Activar tipo de orientación relativa a la trayectoria con alisamiento del desarrollo de la orientación

Activación de los tres ángulos posibles con efecto de giro:

`N... LEAD=`	Ángulo para la orientación programada con relación al vector normal a la superficie
`N... TILT=`	Ángulo para la orientación programada en el plano, vertical respecto a la tangente de trayectoria, relativa respecto al vector normal a la superficie
`N... THETA=`	Ángulo de giro relativo al cambio de orientación alrededor de la dirección de herramienta del tercer eje giratorio

Los valores de los ángulos al final de la secuencia se programan con `LEAD=valor`, `TILT=valor` o `THETA=valor`. Además de los ángulos constantes se pueden programar para los tres ángulos unos polinomios de máx.5º grado.

`N... PO[PHI]=(a2, a3, a4, a5)`	Polinomio para el ángulo de avance LEAD
`N... PO[PSI]=(b2, b3, b4, b5)`	Polinomio para el ángulo lateral TILT
`N... PO[THT]=(d2, d3, d4, d5)`	Polinomio para el ángulo de giro THETA

En la programación se pueden prescindir de los coeficientes de polinomio mayores iguales a cero. Ejemplo: `PO[PHI]=a2` produce para el ángulo de avance `LEAD` una parábola.

Descripción

Orientación de herramienta relativa a la trayectoria

`ORIPATH`	Orientación de la herramienta con relación a la trayectoria
`ORIPATHS`	Orientación de la herramienta con relación a la trayectoria; se suaviza un acodamiento en el desarrollo de la orientación
`LEAD`	Ángulo relativo al vector normal a la superficie en el plano tendido por la tangente de trayectoria y el vector normal a la superficie
`TILT`	Giro de la orientación en dirección Z o giro alrededor de la tangente de trayectoria
`THETA`	Giro alrededor de la dirección de herramienta hacia Z
`PO[PHI]`	Polinomio de orientación para el ángulo de avance LEAD
`PO[PSI]`	Polinomio de orientación para el ángulo lateral TILT
`PO[THT]`	Polinomio de orientación para el ángulo de giro THETA

Nota

Ángulo de giro THETA

Para el giro de la herramienta alrededor de sí misma con el tercer eje giratorio como eje de orientación se precisa una transformada de seis ejes.

6.5.3 Interpolación relativa a la trayectoria del giro de herramienta (ORIROTC, THETA)

Función

Interpolación con vectores de giro

Para el giro programado con ORIROTC de la herramienta con relación a la tangente de trayectoria, el vector de giro también se puede interpolar con un decalaje programable a través del ángulo de giro THETA. Para el ángulo de decalaje se puede programar con PO[THT] un polinomio de máx. 5º grado.

Sintaxis

`N... ORIROTC`	Aplicación de un giro de la herramienta con relación a la tangente de trayectoria
`N... A3= B3= C3= THETA=valor`	Definir el giro del vector de orientación
`N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)`	Interpolar ángulo de decalaje con polinomio de máx. 5º grado

Un giro también se puede programar por sí solo en una secuencia sin que se produzca un cambio de orientación.

Descripción

Interpolación relativa a la trayectoria del giro de la herramienta en la transformada de seis ejes

`ORIROTC`	Aplicar vector de giro tangencial a la trayectoria tangente
`THETA=valor`	Ángulo de rotación en grados que se alcanza al final de la secuencia
`THETA=θe`	Ángulo de giro con ángulo final Θ_e del vector de giro
`THETA=AC(…)`	Conmutación por secuencias al acotado absoluto
`THETA=IC(…)`	Conmutación por secuencias al acotado incremental
`PO[THT]=(c2, c3, c4, c5)`	Interpolar ángulo de decalaje con polinomio de 5º grado

Nota

Interpolación del vector de giro ORIROTC

Si, en contra de la dirección de orientación de la herramienta, también se tiene que aplicar el giro de la herramienta con relación a la tangente de trayectoria, esto sólo es posible con una transformada de seis ejes.

Con ORIROTC activo

No se puede programar el vector de giro ORIROTA. En caso de una programación se emite la ALARMA 14128 "Programación absoluta del giro de herramienta con ORIROTC" activo.

Dirección de orientación de la herramienta en transformada de tres a cinco ejes

Según lo acostumbrado en la transformada de tres a cinco ejes, la dirección de orientación de la herramienta se puede programar a través de ángulos de Euler o RPY o del vector de dirección. Además, son posibles cambios de orientación de la herramienta en el espacio mediante la programación de la interpolación circular de gran radio ORIVECT, la interpolación lineal de los ejes de orientación ORIAXES, todas las interpolaciones en una superficie de cono ORICONxx, así como la interpolación adicionalmente a la curva espacial con dos puntos de contacto de la herramienta ORICURVE.

`G....`	Indicación del tipo de movimiento de los ejes giratorios
`X Y Z`	Indicación de los ejes lineales
`ORIAXES`	Interpolación lineal de los ejes de máquina o ejes de orientación
`ORIVECT`	Interpolación circular de gran radio (idéntico a ORIPLANE)
`ORIMKS`	Giro en el sistema de coordenadas de máquina
`ORIWKS`	Giro en el sistema de coordenadas de pieza
	Descripción, ver aptdo. Giros de la orientación de la herramienta
`A= B= C=`	Programación de la posición de eje de la máquina
`ORIEULER`	Programación de la orientación mediante ángulos eulerianos

`ORIRPY`	Programación de la orientación de la herramienta con ángulos RPY
`A2= B2= C2=`	Programación de ángulos de ejes virtuales
`ORIVIRT1` `ORIVIRT2`	Programación de la orientación de la herramienta con ejes de orientación virtuales (definición 1), definición según DM $MC_ORIAX_TURN_TAB_1 (definición 2), definición según DM $MC_ORIAX_TURN_TAB_2
`A3= B3= C3=`	Programación del vector de dirección del eje de dirección
`ORIPLANE`	Interpolación en el plano (interpolación circular de gran radio)
`ORICONCW`	Interpolación en una superficie de cono en sentido horario
`ORICONCCW`	Interpolación en una superficie de cono en sentido antihorario
`ORICONTO`	Interpolación en una superficie envolvente de cono con transición tangencial
`A6= B6= C6=`	Programación del eje giratorio del cono (vector normalizado)
`NUT=ángulo`	Ángulo en el vértice del cono en grados
`NUT=+179`	Ángulo de desplazamiento menor o igual a 180 grados
`NUT=-181`	Ángulo de desplazamiento mayor o igual a 180 grados
`ORICONIO`	Interpolación en una superficie de cono
`A7= B7= C7=`	Orientación intermedia (programación como vector normalizado)
`ORICURVE` `XH YH ZH, p. ej.: con polinomio PO[XH]=(xe, x2, x3, x4, x5)`	Interpolación de la orientación con especificación del movimiento de dos puntos de contacto de la herramienta. Además del correspondiente punto final se pueden programar polinomios de curvas de espacio adicionales.

Nota

Si la orientación de herramienta con ORIAXES activos se interpola a través de los ejes de orientación, la aplicación relativa a la trayectoria del ángulo de giro sólo se cumple al final de la secuencia.

6.5.4 Alisamiento del desarrollo de la orientación (ORIPATHS A8=, B8=, C8=)

Función

En cambios de orientación con aceleración continua en el contorno no se desean interrupciones de los movimientos interpolados que se pueden producir especialmente en una esquina del contorno. El acodamiento resultante en el desarrollo de la orientación se puede alisar insertando una secuencia intermedia propia. Entonces, el cambio de orientación se realiza con aceleración continua si, durante la reorientación, también está activo ORIPATHS. En esta fase se puede ejecutar un movimiento de levantamiento de la herramienta.

Fabricante de la máquina

Observe las indicaciones del fabricante de la máquina con respecto a eventuales datos de máquina y de operador predefinidos con los cuales se activa esta función.

A través de un dato de máquina se puede ajustar cómo se interpreta el vector de levantamiento:

1. En el sistema de coordenadas de herramienta se define la coordenada Z por la dirección de la herramienta.
2. En el sistema de coordenadas de pieza se define la coordenada Z por el plano activo.

Para más explicaciones relacionadas con la función "Orientación relativa a la trayectoria", ver
Bibliografía: /FB3/ Manual de funciones especiales; Transformada de 3 a 5 ejes (F2)

Sintaxis

Para orientaciones de herramienta continuas con relación a la trayectoria completa se precisan datos de programación adicionales en una esquina del contorno. La dirección y la longitud del trayecto del este movimiento se programan a través del vector con los componentes A8=X, B8=Y C8=Z.

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Descripción

`ORIPATHS`	Orientación de la herramienta con relación a la trayectoria; se suaviza un acodamiento en el desarrollo de la orientación.
`A8= B8= C8=`	Componentes del vector para dirección y longitud del trayecto
`X, Y, Z`	Movimiento de levantamiento en la dirección de la herramienta

Nota

Programación del vector de dirección A8, B8, C8

Si la longitud de este vector es igual a cero, no se produce ningún movimiento de levantamiento.

ORIPATHS

La orientación de herramienta relativa a la trayectoria se activa con ORIPATHS. De lo contrario, la orientación se traslada mediante interpolación circular de gran radio lineal de la orientación inicial a la orientación final.

6.6 Compresión de la orientación (COMPON, COMPCURV, COMPCAD)

Función

Los programas CN en los que está activa una transformada de orientación (TRAORI) y la orientación está programada por medio de vectores de dirección se pueden comprimir, a condición de que se cumplan las tolerancias especificadas.

Nota

El movimiento de orientación sólo se comprime con interpolación circular de gran radio activa, por lo cual depende del código G para la interpolación de orientación. Al igual que la longitud máxima del trayecto y una tolerancia admisible para cada eje o para el avance sobre la trayectoria, se puede ajustar para la función de compresor a través de datos de máquina. Preste atención a las indicaciones del fabricante de la máquina.

Programación

Orientación de herramienta

Si está activa una transformada de orientación (TRAORI), en las máquinas de 5 ejes la orientación de la herramienta se puede programar de la siguiente manera (independientemente de la cinemática):

- Programación del **vector** de dirección mediante:

  `A3=<...> B3=<...> C3=<...>`

- Programación de los **ángulos** de Euler o los **ángulos** RPY mediante:

  `A2=<...> B2=<...> C2=<...>`

Giro de la herramienta

Para las máquinas de **6 ejes**, se puede programar, además de la orientación de la herramienta, el giro de ésta.

Para programar el ángulo de giro se utiliza:

`THETA=<...>`

Ver "Giros de la orientación de la herramienta (Página 360)".

Nota

Las secuencias CN que contienen adicionalmente un giro sólo se pueden comprimir si el ángulo de giro se modifica de modo **lineal**. Por lo tanto, no debe estar programado ningún polinomio con `PO[THT]=(...)` para el ángulo de giro.

Forma general de una secuencia CN comprimible

En consecuencia, la forma general de una secuencia CN comprimible puede ser como la siguiente:

```
N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>
```

o bien

```
N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

Nota

Los valores de posición se pueden indicar de forma directa (p. ej. X90) o de forma indirecta a través de parametrizaciones (p. ej. X=R1*(R2+R3)).

Programación de la orientación de la herramienta por medio de posiciones de ejes giratorios

La orientación de la herramienta se puede indicar también mediante posiciones de eje giratorio, p. ej. en la forma:

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

En ese caso, la compresión se realiza de dos maneras diferentes, según si se lleva a cabo o no una interpolación circular de gran radio. Si no hay interpolación circular de gran radio, el cambio de orientación comprimido se expresa de la manera habitual, mediante polinomios axiales correspondientes a los ejes giratorios.

Precisión del contorno

Según cuál sea el modo de comprensión ajustado (DM20482 $MC_COMPRESSOR_MODE), para la compresión se aplicarán a los ejes geométricos y a los ejes de orientación las tolerancias específicas de eje configuradas (DM33100 $MA_COMPRESS_POS_TOL) o las siguientes tolerancias específicas de canal, ajustables mediante datos de operador:

DO42475 $SC_COMPRESS_CONTUR_TOL (desviación máxima del contorno)

DO42476 $SC_COMPRESS_ORI_TOL (desviación angular máxima para la orientación de la herramienta)

DO42477 $SC_COMPRESS_ORI_ROT_TOL desviación angular máxima para el ángulo de giro de la herramienta (sólo disponible en máquinas de 6 ejes)

Bibliografía:
Manual de funciones básicas; Transformada de 3 a 5 ejes (F2),
apartado: "Compresión de la orientación"

Activación/desactivación

Las funciones del compresor se activan mediante los códigos G modales `COMPON`, `COMPCURV` y `COMPCAD`.

La función de compresor se finaliza con el comando `COMPOF`.

Ver "Compresión de secuencias CC (COMPON, COMPCURV, COMPCAD) (Página 254)".

Nota

Los movimientos de orientación sólo se comprimen si está activa la interpolación circular de gran radio (es decir, el cambio de la orientación de la herramienta tiene lugar en el plano abierto por la orientación inicial y final).

También se lleva a cabo una interpolación circular de gran radio en las siguientes circunstancias:

- DM21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  ORIWKS está activo y
  la orientación está programada mediante vectores (con A3, B3, C3 o A2, B2, C2).
- DM21104 $MC_ORI_IPO_WITH_G_CODE = 1 y
  ORIVECT u ORIPLANE activos.

  La dirección de la herramienta puede estar programada como vector de dirección o con posiciones de eje giratorio. Si está activo uno de los códigos G `ORICONxx` u `ORICURVE` o hay polinomios programados para los ángulos de orientación (`PO[PHI]` y `PO[PSI]`), no se lleva a cabo la interpolación circular de gran radio.

Ejemplo

En el siguiente ejemplo de programa se comprime un círculo aproximado con un trazo polinomial. La orientación de herramienta se mueve de forma síncrona en una superficie de cono. Aunque los cambios de orientación sucesivos programados se producen de forma discontinua, la función del compresor genera un desarrollo liso de la orientación.

```
Programación                              Comentarios
DEF INT CANTIDAD=60
DEF REAL RADIO=20
DEF INT COUNTER
DEF REAL ÁNGULO
N10 G1 X0 Y0 F5000 G64

$SC_COMPRESS_CONTUR_TOL=0.05              ; Desviación máxima del contorno = 0.05 mm
$SC_COMPRESS_ORI_TOL=5                    ; Desviación máxima de la orientación
                                            = 5 grados


TRAORI
COMPCURV
                                          ; Se ejecuta un círculo formado por polí-
                                            gonos. La orientación se mueve en un cono
                                            alrededor del eje Z con un ángulo en el
                                            vértice de 45 grados.
N100 X0 Y0 A3=0 B3=-1 C3=1
N110 FOR COUNTER=0 TO CANTIDAD
N120 ÁNGULO=360*COUNTER/CANTIDAD
N130 X=RADIO*cos(ÁNGULO) Y=RADIO*sin(ÁNGULO)
A3=sin(ÁNGULO) B3=-cos(ÁNGULO) C3=1
N140 ENDFOR
```

6.7 Alisado del desarrollo de la orientación (ORISON, ORISOF)

Función

Con la función "Alisado del desarrollo de la orientación (ORISON)", las variaciones en la orientación pueden alisarse durante varias secuencias. De esta manera, se consigue un desarrollo liso tanto de la orientación como del contorno.

Requisito

La función "Alisado del desarrollo de la orientación (ORISON)" sólo está disponible en sistemas con transformadas de 5 y 6 ejes.

Sintaxis

```
ORISON
...
ORISOF
```

Descripción

`ORISON`:	Alisado del desarrollo de la orientación CON	
	Eficacia:	modal
`ORISOF`:	Alisado del desarrollo de la orientación DES	
	Eficacia:	modal

Datos del operador

El alisado del desarrollo de la orientación se realiza respetando lo siguiente:

- una tolerancia máxima especificada (desviación angular máxima de la orientación de la herramienta en grados)

  y

- una trayectoria máxima especificada.

Estas especificaciones se definen mediante datos del operador:

- DO42678 $SC_ORISON_TOL (tolerancia para el alisado del desarrollo de la orientación)
- DO42680 O$SC_ORISON_DIST (trayectoria para el alisado del desarrollo de la orientación)

Ejemplo

```
Código del programa                Comentarios
...
TRAORI()                           ; Activación de la transformada de orientación.
ORISON                             ; Activación del alisado de la orientación.
$SC_ORISON_TOL=1.0                 ; Tolerancia del alisado de la orientación
                                     = 1,0 grados.
G91
X10 A3=1 B3=0 C3=1
X10 A3=–1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=–1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=–1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=–1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=–1 B3=0 C3=1
...
ORISOF                             ; Desactivación del alisado de la orientación.
...
```

La orientación gira 90 grados en el plano XZ de -45 a +45 grados. Con el alisado del desarrollo de la orientación, la orientación ya no alcanza los valores angulares máximos de --45 o +45 grados.

Información adicional

Cantidad de secuencias

El alisado del desarrollo de la orientación se realiza a lo largo de una cantidad configurada de secuencias almacenada en el dato de máquina DM28590 $MC_MM_ORISON_BLOCKS.

Nota

Si se activa el alisado del desarrollo de la orientación con ORISON sin que se haya configurado memoria de secuencias suficiente para ello (DM28590 < 4), se produce entonces un aviso de alarma y la función no se puede ejecutar.

Longitud máxima de recorrido de secuencia

El desarrollo de la orientación se alisa solamente en aquellas secuencias cuyo recorrido es menor que la longitud de recorrido de secuencia máxima configurada (DM20178 $MC_ORISON_BLOCK_PATH_LIMIT). Las secuencias con recorridos más largos interrumpen el alisado y se ejecutan como se ha programado.

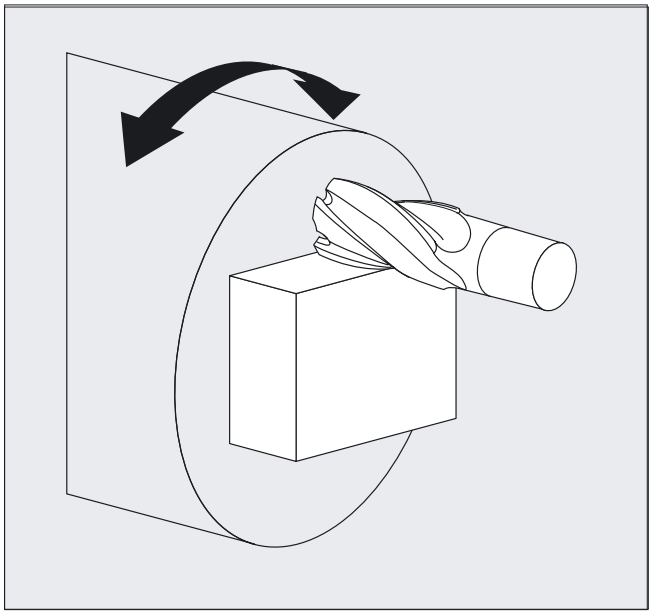
6.8 Transformación cinemática

6.8.1 Fresado en piezas torneadas (TRANSMIT)

Función

TRANSMIT tiene la siguiente funcionalidad:

- Mecanizado (taladros, contorneados) mediante herramienta motorizada en superficies de refrentado en piezas en proceso de torneado.
- Para la programación de estos mecanizados se puede utilizar un sistema de coordenadas cartesiano.
- El control convierte las trayectorias programadas del sistema de coordenadas cartesiano en los desplazamientos de los ejes reales de máquina (en el caso estándar).
  - Eje giratorio
  - Eje de penetración perpendicular al eje giratorio
  - Eje longitudinal paralelo al eje giratorio
  - Los ejes lineales se posicionan perpendicularmente entre sí.
- Es posible el decalaje de origen del centro de la herramienta respecto al centro del eje de giro.
- El control de velocidad tiene en cuenta las limitaciones definidas para los desplazamientos giratorios.



TRANSMIT Tipos de transformada

Para mecanizados TRANSMIT existen dos versiones ajustables:

- `TRANSMIT` en el caso estándar con (TRAFO_TYPE_n = 256)
- `TRANSMIT` con eje lineal Y adicional (TRAFO_TYPE_n = 257)

El tipo de transformada ampliada 257 se puede utilizar, por ejemplo, para compensar correcciones de sujeción de una herramienta con eje Y real.

Sintaxis

```
TRANSMIT o TRANSMIT(n)
TRAFOOF
```

Eje giratorio

El eje giratorio no se puede programar, dado que está ocupado por un eje geométrico, con lo cual no se puede programar directamente como eje de canal.

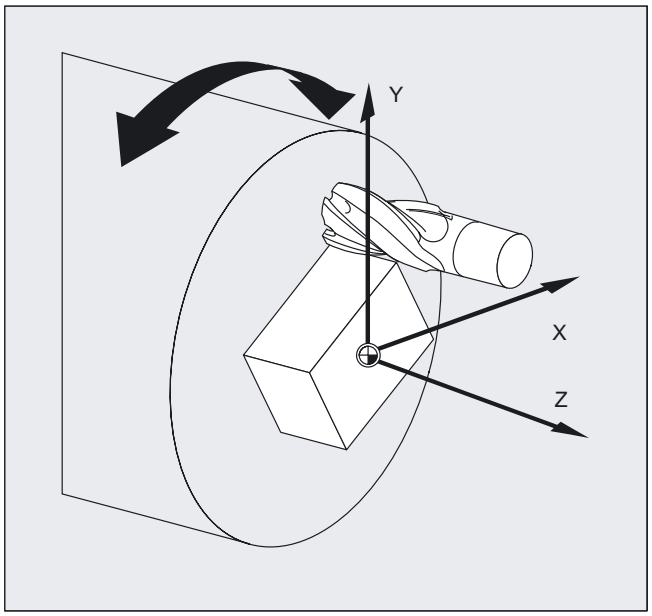
Descripción

`TRANSMIT`:	Activa la primera función TRANSMIT declarada. Esta función se denomina también transformada polar.
`TRANSMIT(n)`:	Activación de la función TRANSMIT (n) declarada; n no debe ser > 2 (TRANSMIT(1) es lo mismo que TRANSMIT).
`TRAFOOF`:	Desactiva una transformada activa
`OFFN`:	Offset contorno normal: Distancia entre el mecanizado frontal y el contorno de referencia programado

Nota

Una transformada `TRANSMIT` activa también se desactiva cuando en el mismo canal se activa alguna de las demás transformadas (p. ej.: `TRACYL`, `TRAANG`, `TRAORI`).

Ejemplo



Código del programa	Comentarios
N10 T1 D1 G54 G17 G90 F5000 G94	; Selección de herramienta
N20 G0 X20 Z10 SPOS=45	; Desplazamiento a la posición inicial
N30 TRANSMIT	; Activación de la función TRANSMIT
N40 ROT RPL=—45	; Ajustar el frame
N50 ATRANS X—2 Y10	
N60 G1 X10 Y—10 G41 OFFN=1OFFN	; Desbastar cuadrado; creces 1 mm
N70 X—10	
N80 Y10	
N90 X10	
N100 Y—10	
N110 G0 Z20 G40 OFFN=0	; Cambio de herramienta
N120 T2 D1 X15 Y—15	
N130 Z10 G41	
N140 G1 X10 Y—10	; Acabar cuadrado
N150 X—10	
N160 Y10	
N170 X10	
N180 Y—10	
N190 Z20 G40	; Deseleccionar frame
N200 TRANS	
N210 TRAFOOF	
N220 G0 X20 Z10 SPOS=45	; Desplazamiento a la posición inicial
N230 M30	

Descripción

Polo

Para el paso por el polo existen dos posibilidades:

- Desplazamiento del eje lineal por sí solo
- Desplazamiento al polo con giro del eje giratorio en el polo y retirada del polo

La selección tiene lugar a través de DM 24911 y 24951.

TRANSMIT con eje lineal Y adicional (tipo de transformada 257):

En una máquina con un eje lineal adicional, esta variante de la transformada polar aprovecha la redundancia para ejecutar una corrección de herramienta mejorada. Para el segundo eje lineal se aplica entonces:

- una zona de trabajo más pequeña y
- el hecho de que el segundo eje lineal no se debería utilizar para la ejecución del programa de pieza.

Para el programa de pieza y la asignación de los correspondientes ejes en el BKS o MKS se presuponen determinados ajustes de los datos de máquina; ver

Bibliografía

/FB2/ Manual de funciones de ampliación; Transformadas cinemáticas (M1)

6.8.2 Transformada de la envolvente del cilindro (TRACYL)

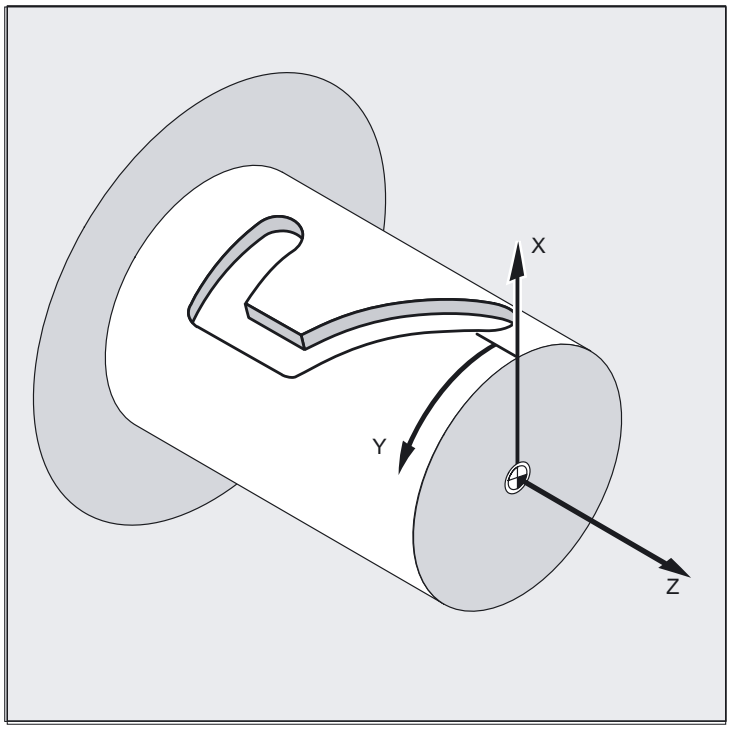
Función

La transformada cilíndrica TRACYL permite realizar las siguientes funciones:

Mecanizado de:

- Ranuras longitudinales en cuerpos cilíndricos
- Ranuras transversales en cuerpos cilíndricos,
- Ranuras con cualquier orientación en piezas cilíndricas

La geometría de las ranuras se programa tomando como referencia la superficie desarrollada del cilindro.



TRACYL Tipos de transformada

La transformada cilíndrica existe en tres versiones:

- `TRACYL` sin corrección de la pared de ranura: (TRAFO_TYPE_n=512)
- `TRACYL` con corrección de la pared de ranura: (TRAFO_TYPE_n=513)
- `TRACYL` con eje lineal adicional y con corrección de la pared de ranura: (TRAFO_TYPE_n=514)
  La corrección de la pared de ranura se parametriza con `TRACYL` a través del tercer parámetro.

En caso de transformada cilíndrica curva con corrección de la pared de ranura, el eje utilizado para la corrección se debería encontrar en (y=0) para que la ranura sea ejecutada en posición centrada frente a la línea central de la ranura programada.

Uso de los ejes

Los siguientes ejes no se pueden utilizar como eje de posicionado o eje de vaivén:

- El eje geométrico en el sentido circunferencial de la envolvente del cilindro (eje Y)
- El eje lineal adicional en la corrección de la pared de ranura (eje Z)

Sintaxis

```
TRACYL(d) o TRACYL(d, n) o
```

para el tipo de transformada 514

```
TRACYL(d, n, corrección de pared de ranura)
TRAFOOF
```

Eje giratorio

El eje giratorio no se puede programar, ya que está ocupado por un eje geométrico y por lo tanto no es posible programarlo directamente como eje de canal.

Descripción

`TRACYL (d)`	Activa la primera función TRACYL acordada en los datos de máquina de canal. d es el Parámetro para el diámetro de trabajo.
`TRACYL (d, n)`	Activa la n función TRACYL acordada en los datos de máquina de canal. n puede ser máx. 2, TRACYL(d,1) equivale a TRACYL(d).
`D`	Valor para el diámetro de trabajo. El diámetro de trabajo es la doble distancia entre la punta de la herramienta y el centro de giro. Este diámetro se tiene que indicar siempre y debe ser mayor que 1.
`n`	2º parámetro opcional para el registro TRACYL 1 (preseleccionado) o 2ª
`corrección de pared de ranura`	3er parámetro opcional cuyo valor se preselecciona para TRACYL de los modos de datos de máquina. Margen de valores: 0: Tipo de transformada 514 sin corrección de la pared de ranura como antes 1: Tipo de transformada 514 con corrección de la pared de ranura
`TRAFOOF`	Transformada DES (BKS y MKS vuelven a ser idénticos).
`OFFN`	Offset contorno normal: Distancia entre la pared de la ranura y el contorno de referencia programado.

Nota

Una transformada `TRACYL` activa también se desactiva cuando en el mismo canal se activa alguna de las demás transformadas (p. ej.: `TRANSMIT`, `TRAANG`, `TRAORI`).

Ejemplo: Definición de la herramienta

El siguiente ejemplo sirve para comprobar la parametrización de la transformada cilíndrica TRACYL:

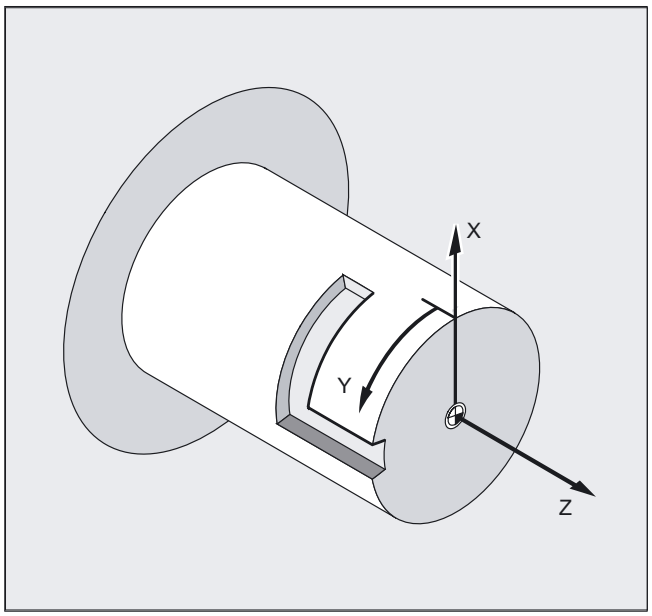
Código del programa	**Comentarios**	
Parámetros de herramienta (DP)	Descripción	Comentario
$TC_DP1[1,1]=120	Tipo de herramienta	Fresa
$TC_DP2[1,1] = 0	Posición del filo	Sólo para herramientas de torneado

Código del programa	**Comentarios**	
Geometría	Corrección longitudinal	
$TC_DP3[1,1]=8.	Vector de corrección longitudinal	Cálculo según tipo
$TC_DP4[1,1]=9.		y plano
$TC_DP5[1,1]=7.		

Código del programa	**Comentarios**	
Geometría	Radio	
$TC_DP6[1,1]=6.	Radio	Radio de la herramienta
$TC_DP7[1,1]=0	Ancho b para sierra, radio de redondeo para fresas	
$TC_DP8[1,1]=0	Alero k	Sólo para sierras
$TC_DP9[1,1]=0		
$TC_DP10[1,1]=0		
$TC_DP11[1,1]=0	Angulo para fresa cónica	

Código del programa	**Comentarios**	
Desgaste	Corrección longitudinal y radial	
$TC_DP12[1,1]=0	Los parámetros restantes hasta $TC_DP24=0	Acotado básico/Adaptador

Ejemplo: Mecanizado de una ranura angular



Activar transformada de evolvente de cilindro:

Código del programa	Comentarios
N10 T1 D1 G54 G90 F5000 G94	; Selección de herramienta, compensación de sujeción
N20 SPOS=0	; Desplazamiento a la posición inicial
N30 G0 X25 Y0 Z105 CC=200	
N40 TRACYL (40)	; Activar transformada cilíndrica curva
N50 G19	; selección del plano

Mecanizado de una ranura angular:

Código del programa	Comentarios
N60 G1 X20	; Posicionar herramienta en el fondo de la ranura
N70 OFFN=12	; Distancia de la pared de la ranura 12 mm respecto a la línea central de la ranura
N80 G1 Z100 G42	; Desplazamiento a la pared derecha de la ranura
N90 G1 Z50	; Segmento de la ranura paralelo al eje de cilindro
N100 G1 Y10	; Segmento de la ranura paralelo a la circunferencia
N110 OFFN=4 G42	; Desplazamiento a la pared izquierda de la ranura; definir distancia de la pared de la ranura en 4 mm respecto a la línea central de la ranura
N120 G1 Y70	; Segmento de la ranura paralelo a la circunferencia
N130 G1 Z100	; Segmento de la ranura paralelo al eje de cilindro

Código del programa	Comentarios
N140 G1 Z105 G40	; Retirada de la pared de la ranura
N150 G1 X25	; Retirar
N160 TRAFOOF	
N170 G0 X25 Y0 Z105 CC=200	; Desplazamiento a la posición inicial
N180 M30	

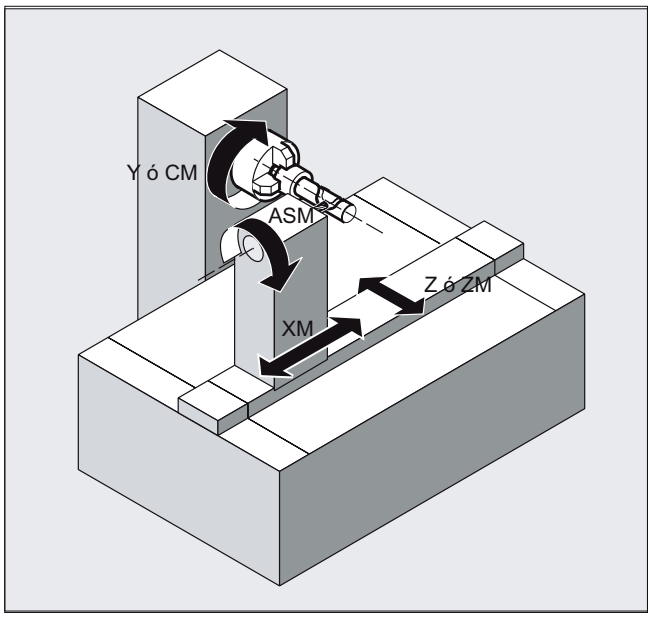
Descripción

Sin corrección de la pared de ranura (tipo de transformada 512):

El control transforma los movimientos de desplazamiento programados del sistema de coordenadas del cilindro a los movimientos de desplazamiento de los ejes de máquina reales.

- Eje giratorio
- Eje de penetración perpendicular al eje giratorio
- Eje longitudinal paralelo al eje giratorio

Los ejes lineales se posicionan perpendicularmente entre sí. El eje de penetración se corta con el eje giratorio.

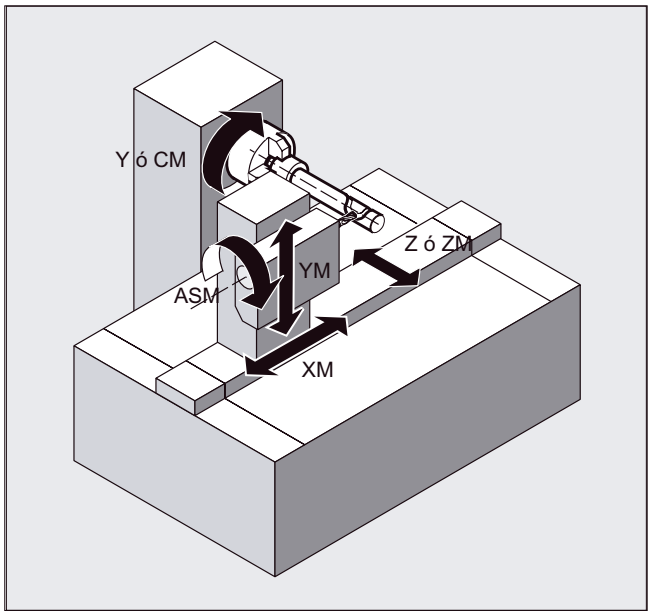


Con corrección de la pared de ranura (tipo de transformada 513):

Cinemática como arriba, pero adicionalmente con eje longitudinal paralelo a la dirección circunferencial

Los ejes lineales se posicionan perpendicularmente entre sí.

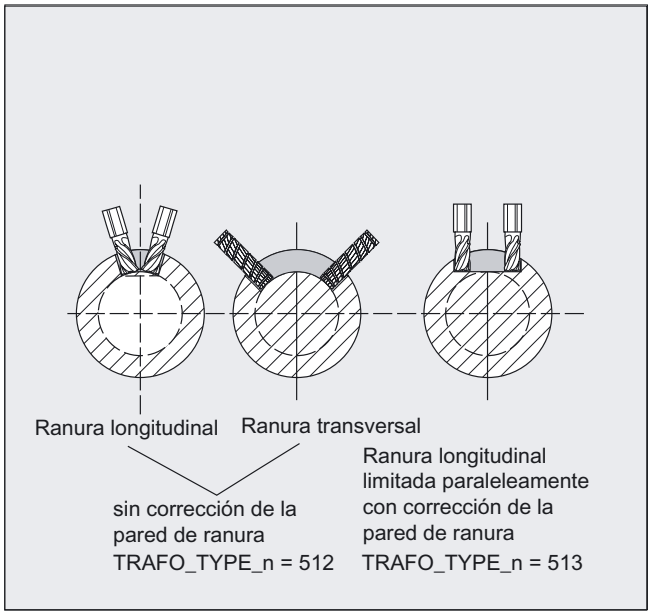
El control de velocidad tiene en cuenta las limitaciones definidas para los desplazamientos giratorios.



Sección de ranura

Con la configuración de ejes 1, las ranuras longitudinales al eje giratorio sólo están limitadas paralelamente si el ancho de la ranura corresponde exactamente al radio de la herramienta.

Las ranuras paralelas a la circunferencia (ranuras transversales) no son paralelas al principio y al final.



Con eje lineal adicional y con corrección de la pared de ranura (tipo de transformada 514):

En una máquina con un eje lineal adicional, esta variante de transformada aprovecha la redundancia para ejecutar una corrección de herramienta mejorada. Para el segundo eje lineal se aplica entonces:

- una zona de trabajo más pequeña y
- el hecho de que el segundo eje lineal no se debería utilizar para la ejecución del programa de pieza.

Para el programa de pieza y la asignación de los correspondientes ejes en el BKS o MKS se presuponen determinados ajustes de los datos de máquina; ver

Bibliografía

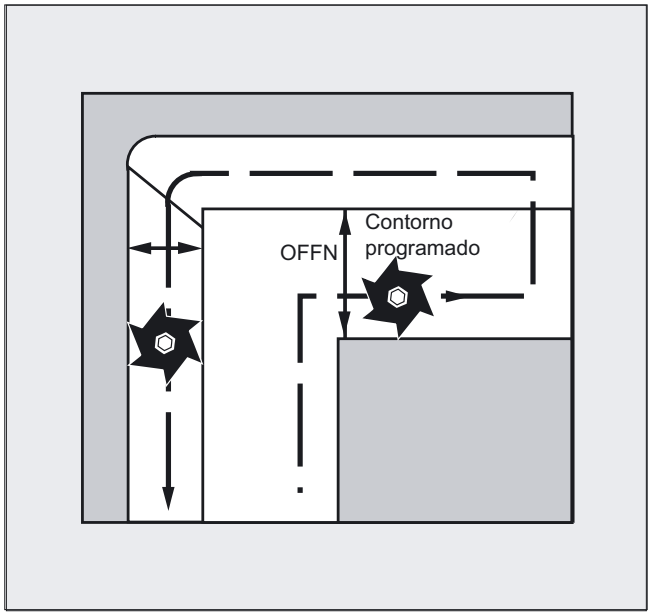
/FB2/ Manual de funciones de ampliación; Transformadas cinemáticas (M1)

Corrección contorno-normal OFFN (tipo de transformada 513)

Para fresar ranuras con `TRACYL`, se programa en el

- programa de pieza la línea del centro de la ranura,
- a través de `OFFN` **a media anchura de ranura**.

OFFN se activa por primera vez con la corrección del radio de herramienta seleccionada para evitar daños en la pared de la ranura. **Además, se debería aplicar OFFN>=radio de la herramienta para excluir daños en la pared opuesta de la ranura.**



Un programa de pieza para el fresado de una ranura se compone generalmente de los siguientes pasos:

1. Seleccionar herramienta
2. Seleccionar TRACYL
3. Seleccionar el decalaje de coordenadas adecuado (FRAME)
4. Posicionado
5. Programar OFFN
6. Seleccionar corrección de radio de herramienta
7. Secuencia de posicionado (entrada de la corrección de radio de herramienta y posicionado en la pared de ranura)
8. Contorno de la línea central de la ranura
9. Cancelar corrección de radio de herramienta
10. Secuencia de retirada (salida de la corrección de radio de herramienta y retirada de la pared de ranura)
11. Posicionado
12. TRAFOOF
13. Volver a seleccionar el decalaje de coordenadas (FRAME) original

Particularidades

- Selección de la corrección de radio de herramienta

  La corrección de radio de herramienta no se programa con respecto a la pared de la ranura, sino con relación a la línea del centro de la ranura programada. Para que la herramienta se desplace a la izquierda de la pared de la ranura, se introduce G42 (en lugar de G41). Esto se evita introduciendo en `OFFN` el ancho de ranura con un signo negativo.

- `OFFN` con `TRACYL` actúa de forma diferente que sin `TRACYL`. Dado que `OFFN` se incluye también sin `TRACYL` con la corrección de radio de herramienta activa, `OFFN` se debería volver a poner a cero después de `TRAFOOF`.

- La modificación de `OFFN` dentro del programa de pieza es posible. De este modo, la línea de centro de la ranura se podría desplazar del centro (ver figura).

- Ranuras guía:

  Con `TRACYL` no se genera la misma ranura con ranuras guía que si hubiera sido fabricada con una herramienta cuyo diámetro corresponde al ancho de la ranura. Por principio, no es posible crear con una herramienta cilíndrica más pequeña la misma geometría de pared de ranura que con una más grande. `TRACYL` reduce el error al mínimo. Para evitar problemas de precisión, el radio de la herramienta sólo debería ser ligeramente menor que la mitad del ancho de la ranura.

Nota

OFFN y corrección del radio de herramienta

Con TRAFO_TYPE_n = 512, el valor bajo `OFFN` actúa como creces para la corrección del radio de herramienta.

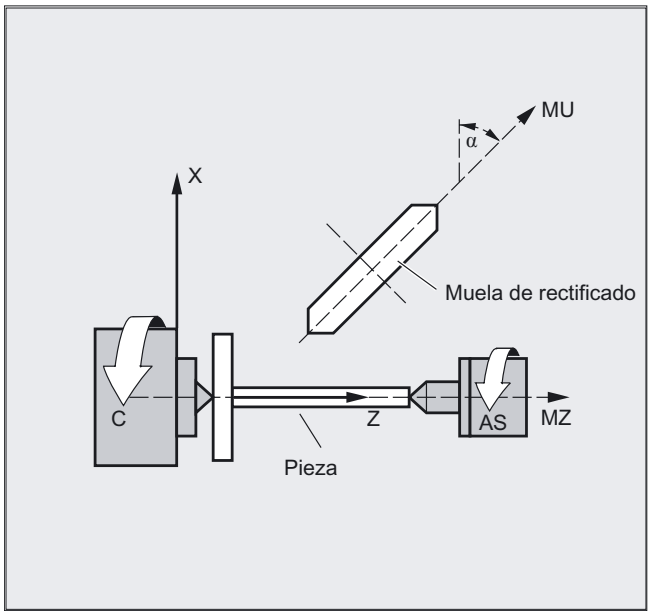
Con TRAFO_TYPE_n = 513 se programa en `OFFN` la mitad del ancho de la ranura. El contorno se recorre con OFFN-WRK.

6.8.3 Eje inclinado (TRAANG)

Función

La función Eje inclinado está prevista para la tecnología Rectificado y tiene las siguientes funcionalidades:

- Mecanizado con eje de penetración inclinado
- Para la programación se puede utilizar un sistema de coordenadas cartesiano.
- El control convierte las trayectorias programadas del sistema de coordenadas cartesiano en los desplazamientos de los ejes reales de máquina (en el caso estándar): eje de penetración inclinado.



Sintaxis

```
TRAANG(α) o TRAANG(α, n)
TRAFOOF
```

Descripción

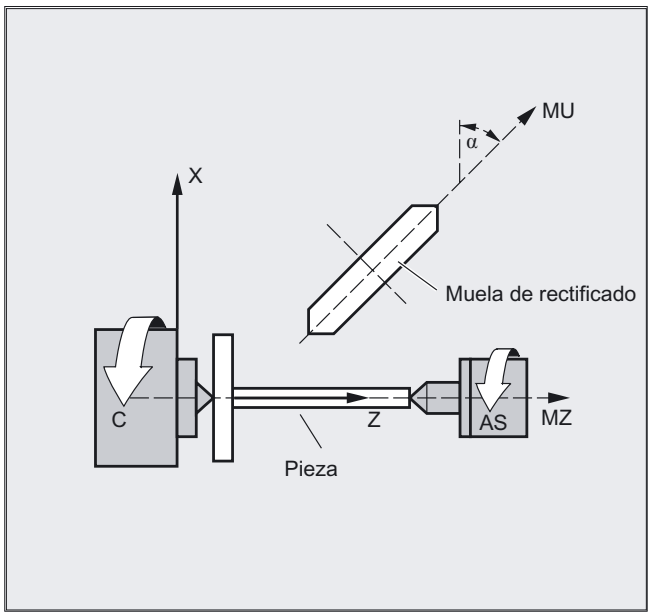
`TRAANG( ) o TRAANG( ,n)`	Activar transformada con la parametrización de la selección anterior.
`TRAANG(α)`	Activa la primera transformada de eje inclinado acordada
`TRAANG(α,n)`	Activa la n transformada de eje inclinado. n puede ser máx. 2. TRAANG(α,1) equivale a TRAANG(α).
`αA`	Ángulo del eje inclinado Los valores admisibles para α son: -90 grados < α < + 90 grados
`TRAFOOF`	Desactivar transformada
`n`	Número de transformadas acordadas

Omitir ángulo α o cero

Si se omite el ángulo α (p. ej.: `TRAANG()`, `TRAANG(,n)`), la transformada se activa con la parametrización de la selección anterior. En la primera selección se aplica la asignación previa según los datos de máquina.

Un ángulo α = 0 (p. ej.: `TRAANG(0)`, `TRAANG(0,n)`) es una parametrización válida y ya no equivale a la omisión del parámetro en las versiones anteriores.

Ejemplo



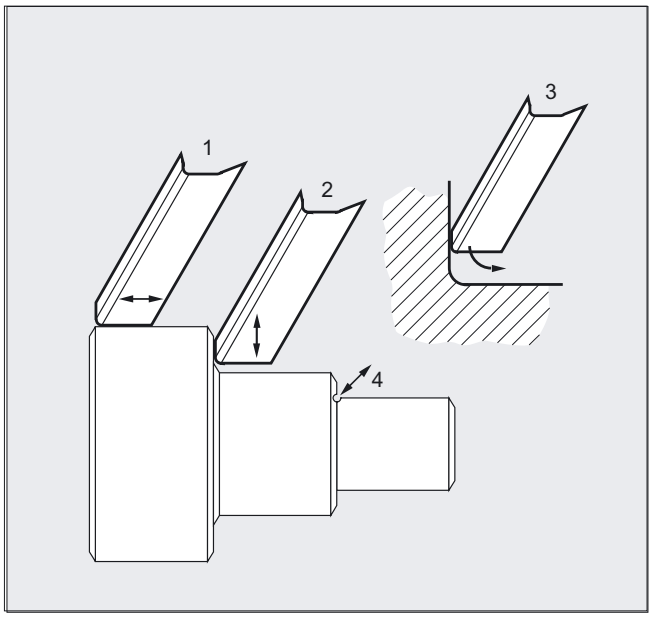
Código del programa	**Comentarios**
N10 G0 G90 Z0 MU=10 G54 F5000 ->	; Selección de herramienta, compensación de sujeción, selección del plano
-> G18 G64 T1 D1	
N20 TRAANG(45)	; Activar transformada eje inclinado
N30 G0 Z10 X5	; Desplazamiento a la posición inicial
N40 WAITP(Z)	; Habilitar ejes para ejecutar vaivén
N50 OSP[Z]=10 OSP2[Z]=5 OST1[Z]=—2 ->	; Ejecutar vaivén hasta alcanzar la cota (para el vaivén, ver el apartado "Vaivén")
-> OST2[Z]=—2 FA[Z]=5000	
N60 OS[Z]=1	
N70 POS[X]=4.5 FA[X]=50	
N80 OS[Z]=0	
N90 WAITP(Z)	; Habilitar ejes de vaivén como ejes de posicionado
N100 TRAFOOF	; Desactivar transformada
N110 G0 Z10 MU=10	; Retirar
N120 M30	;

-> se programa en una misma secuencia

Descripción

Los siguientes mecanizados son posibles:

1. Rectificado longitudinal
2. Rectificado plano
3. Rectificado de un determinado contorno
4. Rectificado oblicuo de ranuras



Fabricante de la máquina

Los siguientes ajustes se establecen a través de un dato de máquina:

- el ángulo entre un eje de máquina y el eje inclinado,
- la posición del origen de herramienta con relación al origen del sistema de coordenadas acordado en la función "Eje inclinado",
- la reserva de velocidad que se mantiene a disposición en el eje paralelo para el movimiento de compensación,
- la reserva de aceleración de eje que se mantiene a disposición en el eje paralelo para el movimiento de compensación.

Configuración de ejes

Para poder programar en el sistema de coordenadas cartesiano, se tiene que comunicar al control la relación entre este sistema de coordenadas y los ejes de máquina que existen efectivamente (MU, MZ):

- Denominación de los ejes geométricos
- Asignación de los ejes geométricos a ejes de canal
  - Caso general (eje inclinado no activo)
  - Eje inclinado activo
- Asignación de los ejes de canal a los números de eje de máquina
- Identificación de los cabezales
- Asignación de nombres de eje de máquina

Con excepción de "Eje inclinado activo", el procedimiento corresponde al procedimiento en la configuración de ejes normal.

6.8.4 Programar eje inclinado (G05, G07)

Función

En el modo JOG, la muela se puede mover, a elección, de forma cartesiana o en dirección al eje inclinado (la indicación sigue siendo cartesiana). Sólo se mueve el eje U real; la indicación del eje Z se actualiza.

Los decalajes REPOS se tienen que ejecutar en el modo Jog a nivel cartesiano.

El paso de la limitación cartesiana del campo de trabajo se vigila en el modo JOG con el "desplazamiento PTP" activo; el eje en cuestión se frena antes. Si el "desplazamiento PTP" no está activo, el eje se puede desplazar exactamente hasta la limitación de la zona de trabajo.

Bibliografía

/FB2/ Manual de funciones de ampliación; Transformada cinemática (M1)

Sintaxis

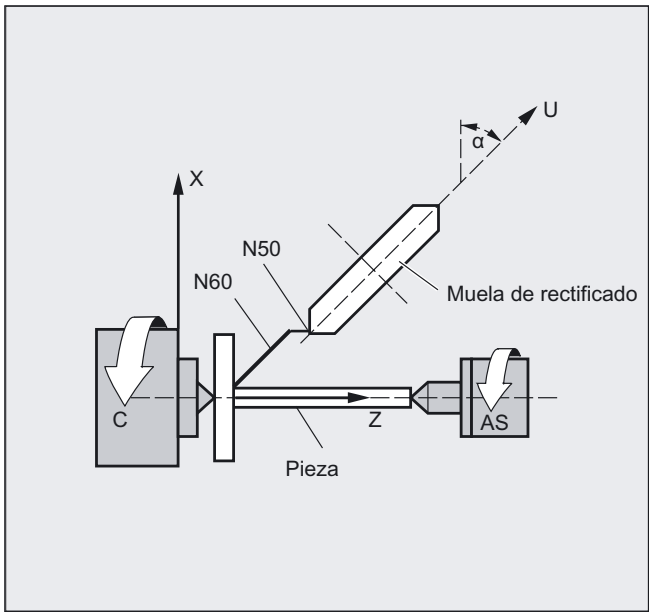
```
G07
G05
```

Los comandos `G07/G05` sirven para facilitar la programación del eje inclinado. Las posiciones se pueden programar y visualizar en el sistema de coordenadas cartesiano. La corrección de herramienta y el decalaje de origen se incluyen a nivel cartesiano. Tras la programación del ángulo para el eje inclinado en el programa CN se puede efectuar el desplazamiento a la posición inicial (`G07`) y después el entallado oblicuo (`G05`).

Descripción

G07	Desplazamiento a la posición inicial
G05	Activar penetración oblicua

Ejemplo



Programación	Comentarios
N.. G18	; Programar el ángulo para el eje inclinado
N50 G07 X70 Z40 F4000	; Desplazamiento a la posición inicial
N60 G05 X70 F100	; Penetración inclinada
N70 ...	;

6.9 Desplazamiento PTP cartesiano

Función

Esta función permite programar una posición en un sistema de coordenadas cartesiano, mientras que el movimiento de la máquina se realiza en coordenadas de máquina. Esta función se puede utilizar, por ejemplo, para cambiar la posición de la articulación, cuando el movimiento pasa por un punto singular.

Nota

Esta función sólo resulta razonable en combinación con una transformada activa. Además, el "Desplazamiento PTP" sólo se admite en combinación con G0 y G1.

Sintaxis

```
N... TRAORI
N... STAT='B10' TU='B100' PTP
N... CP
```

Desplazamiento PTP en transformada genérica de 5/6 ejes

Si se activa con PTP un desplazamiento punto a punto en el sistema de coordenadas de la máquina (ORIMKS) para una transformada genérica activa de 5/6 ejes, entonces la orientación de la herramienta puede programarse tanto con las posiciones de los ejes giratorios

```
N... G1 X Y Z A B C
```

como con los vectores de Euler o los ángulos RPY, independientes de la cinemática

```
N... ORIEULER u ORIRPY
N... G1 X Y Z A2 B2 C2
```

o con los vectores de dirección

```
N... G1 X Y Z A3 B3 C3
```

Además, puede estar activa tanto la interpolación de ejes giratorios como la interpolación vectorial con interpolación circular de gran radio ORIVECT o la interpolación del vector de orientación en una superficie de cono ORICONxx.

Ambigüedades de la orientación con vectores

En la programación de la orientación con vectores existe ambigüedad en las posibles posiciones de los ejes giratorios. Las posiciones de los ejes giratorios que deben alcanzarse pueden seleccionarse mediante la programación de STAT = <...>. En caso de que

se programe STAT = 0 (esto corresponde al ajuste por defecto),
se efectuará el desplazamiento hacia las posiciones que presenten la menor distancia hasta las posiciones iniciales. Si

se programa STAT = 1,
se efectuará el desplazamiento hacia las posiciones que presenten la mayor distancia hasta las posiciones iniciales.

Descripción

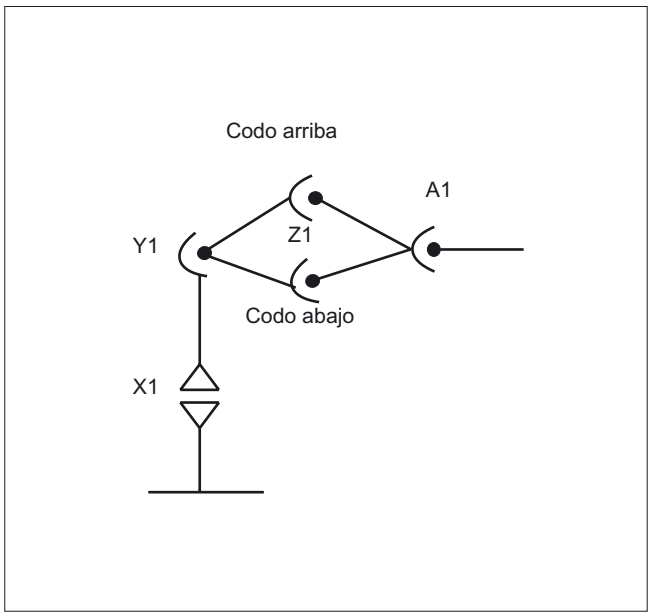
Los comandos `PTP` y `CP` son modalmente activos. `CP` es el ajuste por defecto.

Mientras que la programación del valor STAT es válida modalmente, la programación de TU = <...> es activa secuencia a secuencia.

Otra diferencia es que la programación de un valor STAT sólo tiene efecto con la interpolación vectorial, mientras que la programación de TU también se evalúa en el caso de interpolación activa de ejes giratorios.

`PTP`	**P**oint **t**o **P**oint (desplazamiento punto a punto) El movimiento se ejecuta como movimiento síncrono al eje; el eje más lento que participa en el movimiento es el determinante de velocidad
`CP`	**C**ontinuous **P**ath (movimiento de contorneo) El movimiento se ejecuta como desplazamiento de contorneado cartesiano.
`STAT=`	Posición de las articulaciones; el valor depende de la transformada
`TU=`	La información TURN actúa sólo en la secuencia. De esta manera son posibles desplazamientos unívocos a ángulos de eje entre -360 grados y +360 grados

Ejemplo




```
N10 G0 X0 Y-30 Z60 A-30 F10000
```
Posición inicial
-> Codo arriba

```
N20 TRAORI(1)
```
Transformada activada

```
N30 X1000 Y0 Z400 A0
N40 X1000 Z500 A0 STAT='B10'
TU='B100' PTP
```
Reorientación sin transformada
-> Codo abajo

```
N50 X1200 Z400 CP
```
Transformada de nuevo activa

```
N60 X1000 Z500 A20
N70 M30
```

Ejemplo de desplazamiento PTP en transformada genérica de 5 ejes

Supuesto: Se tiene una cinemática CA rectangular.

Código del programa	Comentarios
TRAORI	; Transformada de cinemática CA activada
PTP	; Conectar desplazamiento PTP
N10 A3 = 0 B3 = 0 C3 = 1	; Posiciones de ejes giratorios C = 0 A = 0
N20 A3 = 1 B3 = 0 C3 = 1	; Posiciones de ejes giratorios C = 90 A = 45
N30 A3 = 1 B3 = 0 C3 = 0	; Posiciones de ejes giratorios C = 90 A = 90
N40 A3 = 1 B3 = 0 C3 = 1 STAT = 1	; Posiciones de ejes giratorios C = 270 A = -45

Seleccionar posición de desplazamiento unívoca para la posición de los ejes giratorios:

De esta forma, en la secuencia N40 los ejes giratorios se desplazan mediante la programación de `STAT = 1` por el trayecto más largo desde su punto de inicio (C=90, A=90) hasta el punto final (C=270, A=–45), contrariamente al caso de `STAT = 0`, en el que se desplazarían por el trayecto mas corto hasta el punto final (C=90, A=45).

Descripción

La conmutación entre el desplazamiento cartesiano y el desplazamiento de los ejes de máquina se efectúa con los comandos `PTP` y `CP`.

Desplazamiento PTP en transformada genérica de 5/6 ejes

En el desplazamiento PTP, si sólo se modifica la orientación, el TCP no permanece en general estacionario, al contrario de lo que ocurre en la transformada de 5/6 ejes. Las posiciones finales transformadas de todos los ejes de transformación (3 ejes lineales y hasta 3 ejes giratorios) se alcanzan de forma lineal, sin que la transformada siga activa en sentido estricto.

El desplazamiento PTP se desactiva mediante programación del código G modal `CP`.

Las diferentes transformadas están incluidas en el documento:
/FB3/ Manual de funciones especiales; Manejo del paquete de transformadas (TE4).

Programación de la posición (STAT=)

La mera especificación de posición por coordenadas cartesianas y la orientación de la herramienta no define unívocamente la posición de la máquina. Según de qué cinemática se trata, existen hasta 8 posiciones de las articulaciones diferentes o diferenciadoras. Por tanto son específicas de la transformada. Para convertir de forma unívoca una posición cartesiana en los ángulos de eje, se debe indicar la posición de las articulaciones con el comando `STAT=`. El comando "`STAT`" contiene como valor binario un bit para cada una de las posiciones posibles.

Para los bits de posición a programar en "`STAT`", ver:
/FB2/ Manual de funciones de ampliación; Transformada cinemática (M1), apartado "Desplazamiento PTP cartesiano".

Programación de los ángulos de eje (TU=)

Para ejecutar posicionamientos unívocos en ángulos de eje < ±360 grados se debe programar esta información con el comando "TU=".

Los ejes se desplazan por el trayecto más corto:

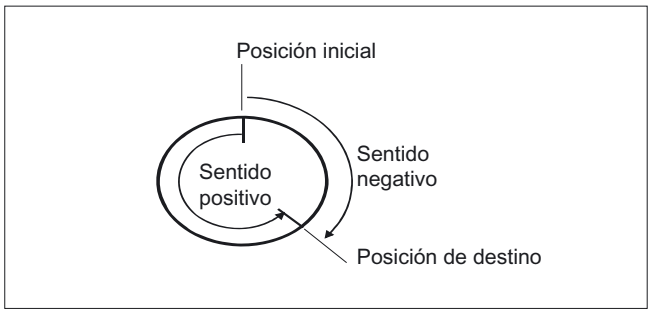
- cuando no se programa en ninguna posición TU
- para ejes con una zona de desplazamiento > ±360 grados.

Ejemplo:

A la posición de destino indicada en la figura se puede ir en sentido negativo o positivo. En la dirección A1 se programa el sentido.

A1=225°, TU=bit 0, → sentido positivo

A1=−135°, TU=bit 1, → sentido negativo



Ejemplo de evaluación de TU para transformada genérica de 5/6 ejes y posiciones de destino

Para cada eje que se ve afectado por la transformada, la variable TU contiene un bit que muestra el sentido de desplazamiento. La asignación de los bits de TU corresponde a la vista de eje de canal de los ejes giratorios. La información de TU sólo se evalúa para los 3 ejes giratorios posibles como máximo que se ven afectados por la transformada:

Bit0: eje 1, bit TU = 0: 0 grados <= ángulo del eje giratorio < 360 grados

Bit1: eje 2, bit TU = 1: –360 grados < ángulo del eje giratorio < 0 grados

La posición inicial de un eje giratorio es C = 0; mediante la programación de C = 270 el eje giratorio se desplaza hasta las siguientes posiciones de destino:

C = 270: bit TU 0, sentido de giro positivo

C = –90: bit TU 1, sentido de giro negativo

Comportamiento posterior

Cambio del modo de operación

La función "Desplazamiento cartesiano PTP" sólo resulta razonable en los modos AUTO y MDA. Al cambiar el modo de operación a JOG se mantiene el ajuste actual.

Si se ha ajustado el código G `PTP`, se desplazan los ejes en el MKS. Si se ha ajustado el código `PTP`, se desplazan los ejes en el WKS.

Power On/RESET

Tras Power On o tras RESET, el ajuste depende del dato de máquina `$MC_GCODE_REST_VALUES[48]`. De forma estándar está ajustado el modo de desplazamiento "`CP`".

REPOS

Si durante la secuencia interrumpida estaba ajustada la función "Desplazamiento cartesiano PTP", también se reposiciona con `PTP`.

Movimientos superpuestos

El decalaje DRF o el decalaje de origen externo sólo son posibles de forma limitada para el desplazamiento cartesiano PTP. Al cambiar de un movimiento PTP a un movimiento CP no deben existir superposiciones en el BKS.

Transición suave (matado de esquinas) entre movimientos CP y PTP

Entre las secuencias se puede incluir una transición suave programable con `G641`.

El tamaño de la zona de matado es el trayecto en mm o pulgadas a partir del cual, o bien hasta el cual se suaviza la transición entre secuencias. El tamaño se debe indicar como sigue:

- Para secuencia G0 con `ADISPOS`
- Para todos los demás comandos de desplazamiento con `ADIS`

El cálculo del trayecto corresponde a la consideración de las direcciones F para aquellas secuencias que no sean G0. El avance se mantiene en los ejes indicados en `FGROUP(...)`.

Cálculo del avance

Para secuencias CP se utilizan los ejes cartesianos del sistema de coordenadas básico para el cálculo.

Para secuencias PTP se utilizan para el cálculo los ejes correspondientes del sistema de coordenadas de máquina.

6.9.1 PTP con TRANSMIT

Función

Con PTP en TRANSMIT se pueden posicionar secuencias G0 y G1 optimizadas en el tiempo. En lugar de desplazar linealmente los ejes del sistema de coordenadas básico (CP), se desplazan linealmente los ejes de máquina (PTP). En consecuencia, el desplazamiento de los ejes de máquina en la proximidad del polo tiene el efecto de que el punto final de la secuencia se alcanza considerablemente antes.

El programa de pieza se sigue escribiendo en el sistema de coordenadas cartesiano y todos los decalajes de coordenadas, giros y programaciones de frames conservan su validez. La simulación en HMI se muestra igualmente en el sistema de coordenadas cartesiano.

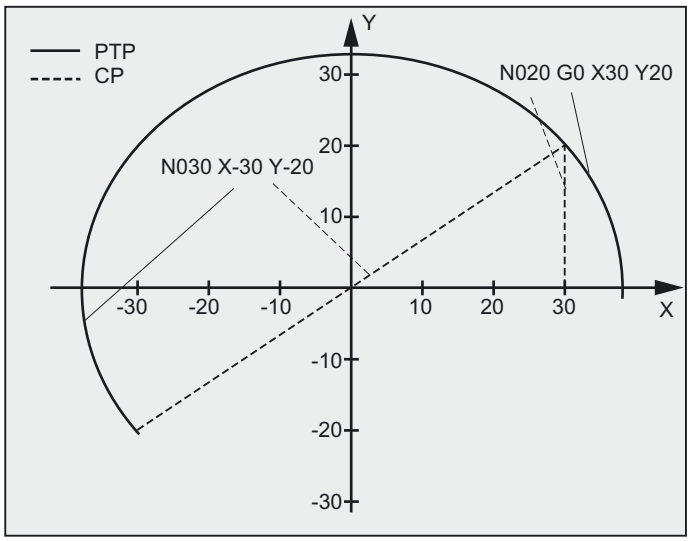
Sintaxis

```
N... TRANSMIT
N... PTPG0
N... G0 ...
...
N... G1 ...
```

Descripción

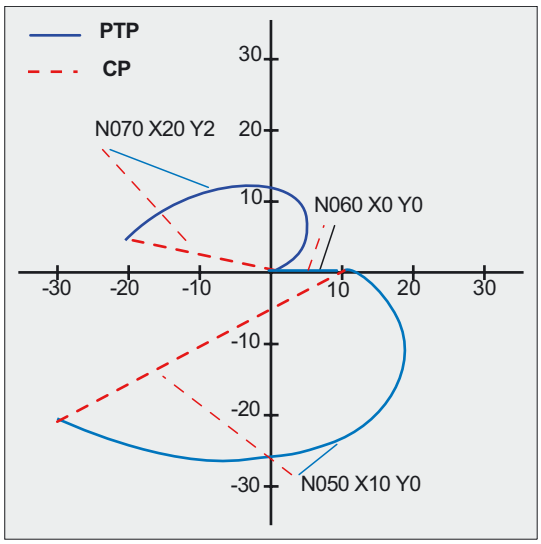
`TRANSMIT`	Activa la primera función TRANSMIT declarada (ver el apartado "Fresado en piezas torneadas: TRANSMIT")
`PTPG0`	**P**oint **t**o **P**oint **G0** (desplazamiento punto a punto automático para cada secuencia G0; a continuación, nuevo ajuste a CP) Dado que STAT y TU son modales, siempre se aplica el último valor programado.
`PTP`	**P**oint **t**o **P**oint (desplazamiento punto a punto) Para TRANSMIT, PTP significa que, en el sistema de coordenadas cartesiano, el movimiento se realiza en espirales de Arquímedes, esquivando el polo o retirándose de él. Los desplazamientos de la herramienta resultantes se desarrollan de forma claramente distinta a CP y se representan en los correspondientes ejemplos de programación.
`STAT=`	Resolución de la ambigüedad con respecto al polo.
`TU=`	TU no es relevante en PTP con TRANSMIT

Ejemplo: Esquivar el polo con PTP y TRANSMIT



Código del programa	**Comentarios**
N001 G0 X30 Z0 F10000 T1 D1 G90	; Posición inicial acotado absoluto
N002 SPOS=0	
N003 TRANSMIT	; Transformada TRANSMIT
N010 PTPG0	; Para cada secuencia G0 automáti-camente PTP y luego de nuevo CP
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Ejemplo: Retirada del polo con PTP y TRANSMIT



Programación	**Comentarios**
N001 G0 X90 Z0 F10000 T1 D1 G90	; Posición inicial
N002 SPOS=0	
N003 TRANSMIT	; Transformada TRANSMIT
N010 PTPG0	; Para cada secuencia G0 automáticamente PTP y luego de nuevo CP
N020 G0 X90 Y60	
N030 X-90 Y-60	
N040 X-30 Y-20	
N050 X10 Y0	
N060 X0 Y0	
N070 X-20 Y2	
N170 G1 X0 Y0	
N160 X10 Y0	
N150 X-30 Y-20	
M30	

Descripción

PTP y PTPG0

PTPG0 se considera en todas las transformaciones que pueden ejecutar PTP. En todos los demás casos, PTPG0 no tiene relevancia.

Las secuencias G0 se ejecutan en el modo CP.

La selección de PTP y PTPG0 se realiza en el programa de pieza o mediante la cancelación de CP en el dato de máquina $MC_GCODE_RESET_VALUES[48].

⚠ PRECAUCIÓN

Limitaciones

Con respecto a los desplazamientos de la herramienta y la colisión se aplican varias limitaciones y determinadas exclusiones de funciones, tales como:

Con PTP no debe estar activa ninguna corrección del radio de herramienta (WRK).

Con PTPG0, con la corrección del radio de herramienta (WRK) activa, se aplica CP.

Con PTP, la aproximación/retirada suave no es posible.

Con PTPG0, con la aproximación/retirada suave, se aplica CP.

Con PTP no se pueden realizar ciclos de desbaste (CONTPRON, CONTDCON).

Con PTPG0, se aplica en los ciclos de desbaste (CONTPRON, CONTDCON) CP.

Chaflán (CHF, CHR) y redondeo (RND, RNDM) se ignoran.

El compresor no es compatible con PTP y se cancela automáticamente en secuencias PTP.

Una superposición de ejes en la interpolación no debe cambiar durante la parte de PTP.

Con G643 se conmuta automáticamente después del matado de esquinas con precisión por eje G642.

Cuando PTP está activo, los ejes de la transformada no pueden ser, al mismo tiempo, ejes de posicionado.

Bibliografía:

/FB2/ Manual de funciones de ampliación; Transformada cinemática (M1), apartado "Desplazamiento PTP cartesiano"

PTP con TRACON:

PTP también se puede utilizar con TRACON si soporta la primera transformada concatenada PTP.

Significado de STAT= y TU= con TRANSMIT

Si el eje giratorio tiene que girar en 180 grados o el contorno en CP tiene que pasar por el polo, los ejes giratorios pueden, en función del dato de máquina $MC_TRANSMIT_POLE_SIDE_FIX_1/2 [48], girar en -/+ 180 grados o desplazarse en sentido horario o antihorario. Asimismo se puede ajustar si se pasa por el polo o se gira alrededor del mismo.

6.10 Limitaciones en la selección de una transformada

Funcionamiento

La selección de transformadas es posible a través del programa de pieza o MDA. Se deberá de tener en cuenta:

- Una secuencia intermedia de desplazamiento no se inserta (chaflanes/radios).
- Una sucesión de secuencias spline tiene que estar terminada; de lo contrario, aparece un mensaje.
- La corrección de herramienta fina tiene que estar cancelada (FTOCOF); de lo contrario, aparece un mensaje.
- La corrección de radio de herramienta tiene que estar cancelada (G40); de lo contrario, aparece un mensaje.
- Una corrección de longitud de herramienta activada es adoptada por el control a la transformada.
- El frame actual activo antes de la transformada es cancelado por el control.
- Una limitación del campo de trabajo activa es cancelada por el control para los ejes afectadas por la transformada (equivale a WALIMOF).
- La vigilancia de la zona protegida se cancela.
- El modo de contorneado y el matado de esquinas se interrumpen.
- Todos los ejes indicados en el dato de máquina tienen que estar sincronizados con relación a la secuencia.
- En los ejes invertidos se deshace la inversión; de lo contrario, aparece un mensaje.
- En ejes dependientes se emite un mensaje.

Cambio de herramienta

Un cambio de herramienta sólo se admite si la corrección del radio de herramienta está cancelada.

Un cambio de la corrección de longitud de herramienta y la selección/cancelación de la corrección del radio de herramienta no se deben programar en la misma secuencia.

Cambio de frame

Se admiten todas las instrucciones que se refieren únicamente en el sistema de coordenadas básico (FRAME, corrección del radio de herramienta). Sin embargo, a diferencia de la transformada activa, un cambio de frame con G91 (medida incremental) no se trata por separado. El incremento a ejecutar se evalúa en el sistema de coordenadas de pieza del nuevo frame, independientemente del frame que actuaba en la secuencia anterior.

Exclusiones

Los ejes afectados por la transformada no se pueden utilizar:

- como eje Preset (alarma),
- para el desplazamiento a punto fijo (alarma),
- para el referenciado (alarma),

6.11 Cancelar transformada (TRAFOOF)

Función

Con el comando `TRAFOOF` se desactivan todas las transformadas y todos los frames activos.

Nota

Los frames que se necesitan a continuación se tienen que activar mediante una nueva programación.

Se deberá de tener en cuenta:

Para la cancelación de las transformadas se aplican las mismas limitaciones que para la selección (ver capítulo anterior "Límites en la selección de una transformada").

Sintaxis

```
TRAFOOF
```

Descripción

`TRAFOOF`	Comando para desactivar todos los frames y transformadas activos

6.12 Transformadas concatenadas (TRACON, TRAFOOF)

Función

Se pueden concatenar **dos** transformadas, de modo que las partes de movimiento para los ejes de la primera transformada sean los datos de entrada para la segunda transformada concatenada. Las partes de movimiento de la segunda transformada actúan en los datos de máquina.

La cadena puede abarcar **dos** transformadas.

Nota

Una herramienta siempre se asigna a la primera transformada de una cadena. La siguiente transformada se comporta entonces como si la longitud de herramienta activa fuera cero. Sólo actúan las longitudes básicas de una herramienta (_BASE_TOOL_) ajustadas a través de datos de máquina para la primera transformada de la cadena.

Fabricante de la máquina

Observe las indicaciones del fabricante de la máquina acerca de eventuales transformadas predefinidas por datos de máquina.

Las transformadas y transformadas concatenadas con opciones. El catálogo informa sobre la disponibilidad de determinadas transformadas en la cadena en determinados controles.

Aplicaciones

- Rectificado de contornos programados como envolvente de un desarrollo de cilindro (TRACYL) con una muela inclinada, p. ej., afilado de herramientas.
- Mecanizado fino de un contorno no redondo generado con TRANSMIT con una muela inclinada.

Sintaxis

`TRACON(trf,par)`	Se activa una transformada concatenada.
`TRAFOOF`	

Descripción

`TRACON`	Se activa la transformada concatenada. Otra transformada activada previamente es desactivada implícitamente por TRACON().
`TRAFOOF`	La última transformada activada (concatenada) se desactiva.
`trf`	Número de la transformada concatenada: 0 ó 1 para la primera/única transformada concatenada. Si no se ha programado nada en este punto, equivale a la indicación del valor 0 ó 1; es decir, que se activa la primera/única transformada. 2 para la segunda transformada concatenada. (Valores distintos a 0 - 2 generan una alarma de error).
`par`	Uno o varios parámetros separados por comas para las transformadas en la concatenación que esperan parámetros, p. ej., ángulo del eje inclinado. Si no se definen parámetros, se aplican los ajustes previos o los últimos parámetros utilizados. Colocando comas, se tiene que asegurar que los parámetros indicados se evalúan en el orden en el cual se esperan si tienen que actuar ajustes previos para parámetros anteriores. En particular, en caso de indicación de al menos un parámetro, éste tiene que ser precedido por una coma aunque no es necesario indicar trf, o sea, por ejemplo, TRACON(, 3.7).

Requisito

La **segunda** transformada tiene que ser **"Eje inclinado"** (`TRAANG`). Para la primera transformada son posibles:

- Transformadas de orientación (`TRAORI`), incluido cabezal portafresas cardánico
- `TRANSMIT`
- `TRACYL`
- `TRAANG`

El requisito para el uso del comando de activación para una transformada concatenada es que las transformadas individuales a concatenar y la transformada concatenada a activar están definidas por datos de máquina.

Los límites y los casos especiales indicados en las descripciones individuales se tienen que observar también para el uso dentro de una concatenación.

La información sobre la configuración de los datos de máquina de las transformadas se encuentra en:

/FB2/ Manual de funciones de ampliación; Transformadas cinemáticas (M1) y

/FB3/ Manual de funciones especiales; Transformadas de 3 a 5 ejes (F2).

Correcciones de herramientas 7

7.1 Memoria de corrección

Función

Estructura de la memoria de correctores

Cada campo de datos se puede activar con un número T y D (con excepción del "Número D plano"), y contiene los campos con la información asociada a la herramienta (información geométrica, tipo de herramienta, etc.).

Estructura plana de números D

La "Estructura plana de números D" se utiliza cuando la gestión de herramientas se realiza fuera del NCK. En este caso, se depositan los números D con los correspondientes juegos de datos de corrección sin asignación a las herramientas.

En el programa de pieza se puede seguir programando T. Esta T carece de referencia con el número D programado.

Datos de filo del usuario

Mediante los datos de máquina se pueden configurar los datos de filo de usuario. Preste atención a las indicaciones del fabricante de la máquina.

Parámetros de herramienta

Nota

Valores individuales en la memoria de corrección

Los valores individuales de la memoria de corrección (P1 a P25) se pueden leer y escribir por programa a través de variables del sistema. Todos los demás parámetros están reservados.

Los parámetros de herramienta $TC_DP6 a $TC_DP8, $TC_DP10 y $TC_DP11, así como $TC_DP15 a $TC_DP17, $TC_DP19 y $TC_DP20 cambian de significado en función del tipo de herramienta.

[1]Es válido también como fresa para el fresado frontal en 3D
[2]En tipo de herramienta Sierra
[3]reservado: No utilizado por SINUMERIK 840D

Número del parámetro de herramientas (DP)	Significado de las variables de sistema	Observación
$TC_DP1	Tipo de herramienta	Vista de conjunto: ver lista
$TC_DP2	Posición del filo	Sólo para herramientas de torneado
Geometría	**Corrección longitudinal**	
$TC_DP3	Longitud 1	Cálculo según
$TC_DP4	Longitud 2	Tipo y plano
$TC_DP5	Longitud 3	
Geometría	**Radio**	
$TC_DP6[1] $TC_DP6[2]	Radio 1/longitud 1 diámetro d	Herramienta de fresado/ torneado/rectificado Sierra
$TC_DP7[1] $TC_DP7[2]	Longitud 2/radio de redondeo de fresas cónicas Anchura de la ranura b radio de redondeo	Fresas Sierra
$TC_DP8[1] $TC_DP8[2]	Radio de redondeo 1 para fresas Alero k	Fresas Sierra
$TC_DP9[1,3]	Radio de redondeo 2	reservado
$TC_DP10[1]	Ángulo 1 superficie frontal de la herramienta	Fresas cónicas
$TC_DP11[1]	Ángulo 2 eje longitudinal de la herramienta	Fresas cónicas
Desgaste	**Corrección longitudinal y radial**	
$TC_DP12	Longitud 1	
$TC_DP13	Longitud 2	
$TC_DP14	Longitud 3	
$TC_DP15[1] $TC_DP15[2]	Radio 1/longitud 1 diámetro d	Herramienta de fresado/torneado/rectificado Sierra
$TC_DP16[1] $TC_DP16[3]	Longitud 2/radio de redondeo de fresas cónicas Anchura de la ranura b radio de redondeo	Fresas Sierra para ranurar
$TC_DP17[1] $TC_DP17[2]	Radio de redondeo 1 para fresas Alero k	Fresas/fresas frontales en 3D Sierra
$TC_DP18[1,3]	Radio de redondeo 2	reservado
$TC_DP19[1]	Ángulo 1 superficie frontal de la herramienta	Fresas cónicas
$TC_DP20[1]	Ángulo 2 eje longitudinal de la herramienta	Fresas cónicas
Acotado básico/ Adaptador	**Correcciones longitudinales**	
$TC_DP21	Longitud 1	
$TC_DP22	Longitud 2	
$TC_DP23	Longitud 3	
Tecnología		
$TC_DP24	Ángulo de despulla	Sólo para herramientas de torneado
$TC_DP25		reservado

Notas

Para los datos geométricos (p. ej.: longitud 1 o radio) existen varios campos. El tamaño de la herramienta es el resultado aditivo de varios de estos campos (p. ej.: longitud total 1, radio total).

A los valores de corrección que no sean necesarios se les debe dar el valor 0.

Parámetros de herramienta $TC-DP1 a $TC-DP23 con herramientas de contorno

Nota

Los parámetros de herramientas que no están representados en la tabla, como p. ej. $TC_DP7, no se evalúan, es decir, su contenido carece de importancia.

Número del parámetro de herramientas (DP)	Descripción	Filos Dn		Observación
$TC_DP1	Tipo de herramienta			400 a 599
$TC_DP2	Posición del filo			
Geometría	**Corrección longitudinal**			
$TC_DP3	Longitud 1			
$TC_DP4	Longitud 2			
$TC_DP5	Longitud 3			
Geometría	**Radio**			
$TC_DP6	Radio			
Geometría	**Ángulo límite**			
$TC_DP10	Ángulo límite mínimo			
$TC_DP11	Ángulo límite máximo			
Desgaste	**Corrección longitudinal y radial**			
$TC_DP12	Desgaste longitud 1			
$TC_DP13	Desgaste longitud 2			
$TC_DP14	Desgaste longitud 3			
$TC_DP15	Desgaste radio			
Desgaste	**Ángulo límite**			
$TC_DP19	Desgaste ángulo límite mínimo			
$TC_DP20	Desgaste ángulo límite máximo			
Acotado básico/ Adaptador	**Correcciones longitudinales**			
$TC_DP21	Longitud 1			
$TC_DP22	Longitud 2			
$TC_DP23	Longitud 3			

Valor básico y valor de desgaste

Las magnitudes resultantes proceden de la correspondiente suma del valor básico más el valor de desgaste (p. ej. $TC_DP6 + $TC_DP15 para el radio). Para la longitud de herramienta de los primeros filos se añade además el acotado básico ($TC_DP21 – $TC_DP23). Adicionalmente, a estas longitudes de herramienta les afectan todas las demás magnitudes que también pueden influir en la longitud efectiva de la herramienta en el caso de una herramienta convencional (adaptador, portaherramientas orientable, datos de operador).

Ángulos límite 1 y 2

Los ángulos límite 1 o 2 se refieren al vector que va desde el centro hasta el punto de referencia del filo y se miden en sentido antihorario.

7.2 Correcciones aditivas

7.2.1 Seleccionar correcciones aditivas (DL)

Funcionamiento

Las correcciones aditivas pueden considerarse como correcciones de proceso programables durante la ejecución. Se refieren a los datos geométricos de un filo y forman parte, por lo tanto, de los datos del filo de la herramienta.

Los datos de una corrección aditiva se activan a través de un número DL (DL: location dependent; correcciones relativas al lugar de uso) y se introducen a través de la interfaz de usuario.

Aplicación

Las correcciones aditivas sirven para compensar errores dimensionales debidos a las peculiaridades del lugar de uso.

Sintaxis

```
DL=<Número>
```

Descripción

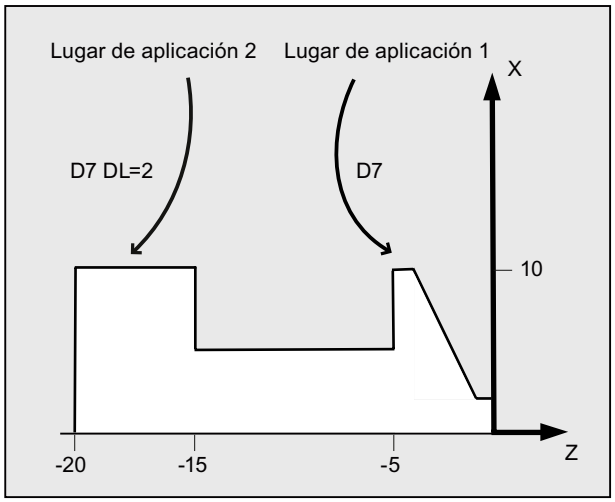
`DL`	Comando para activar una corrección aditiva
`<Número>`	El parámetro `<Número>` especifica el juego de datos de corrección de herramienta aditivo que se desea activar.

Nota

La cantidad y la activación de las correcciones aditivas se definen mediante datos de máquina (→ ¡Observar las indicaciones del constructor de la máquina!).

Ejemplo:

Se utiliza el mismo filo para 2 asientos de cojinete:



Código del programa	**Comentarios**
N110 T7 D7	; La torreta revólver se posiciona en el puesto 7. D7 y DL=1 se activan y se aplican en la secuencia siguiente.
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; Junto a D7 se activa DL=2 y se aplica en la secuencia siguiente.
N150 G1 Z-21	
N160 G0 X200 Z200	; Posicionamiento en el punto de cambio de herramienta.
...	

7.2.2 Definir valores de desgaste y de ajuste ($TC_SCPxy[t,d], $TC_ECPxy[t,d])

Función

Los valores de desgaste y de ajuste se pueden leer y escribir mediante variables de sistema. Para ello se orienta la lógica a la de las correspondientes variables de sistema para herramientas y filos.

Variables del sistema

Variable del sistema	Descripción
$TC_SCPxy[<t>,<d>]	Valores de desgaste asignados a los parámetros geométricos correspondientes mediante xy, donde "x" es el número del valor del desgaste e "y" la referencia a los parámetros geométricos.
$TC_ECPxy[<t>,<d>]	Valores de ajuste que estén asignados a los parámetros geométricos mediante xy, donde "x" es el número del valor del ajuste e "y" la referencia a los parámetros geométricos.
<t>: Número T de la herramienta <d>: Número D del filo de la herramienta	

Nota

Los valores definidos para el desgaste y el ajuste se suman a los parámetros geométricos y al resto de los parámetros de corrección (números D).

Ejemplo

El valor de desgaste de la longitud 1 se fija a 1.0 para el filo <d> de la herramienta <t>.

Parámetros: $TC_DP**3** (Longitud 1, en herramientas de tornear)

Valores de desgaste: $TC_SCP1**3** hasta $TC_SCP6**3**

Valores de ajuste: $TC_ECP1**3** hasta $TC_ECP6**3**

$TC_SCP4**3** [<t>,<d>] = 1.0

7.2.3 Borrar correcciones aditivas (DELDL)

Función

Con el comando `DELDL` se borran las correcciones aditivas del filo de una herramienta (libera memoria). Para ello se borran tanto los valores definidos para el desgaste como los de ajuste.

Sintaxis

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<Estado>=DELDL[<t>,<d>]
```

Descripción

`DELDL`	Comando para borrar correcciones aditivas
`<t>`	Número T de la herramienta
`<d>`	Número D del filo de la herramienta
`DELDL[<t>,<d>]`	Se borran todas las correcciones aditivas del filo `<d>` de la herramienta `<t>`.
`DELDL[<t>]`	Se borran todas las correcciones aditivas de todos los filos de la herramienta `<t>`.
`DELDL`	Se borran todas las correcciones aditivas de los filos de todas las herramientas de la unidad TO (para el canal en el que se ha programado el comando).
`<Estado>`	Estado de borrado Valor: Significado: 0 El borrado se ha ejecutado con éxito. - No se ha ejecutado el borrado (si la parametrización designa exactamente un filo), o el borrado no se ha ejecutado por completo (si la parametrización designa varios filos).

Nota

Los valores de ajuste y de desgaste de las herramientas activas no se pueden borrar (análogamente al comportamiento de borrado de correcciones `D` o de datos de herramienta).

7.3 Acciones especiales de corrección de herramienta

Función

Los datos de operador DO42900 a DO42960 permiten controlar la evaluación del signo de la longitud de herramienta y el desgaste.

Ello también es válido para el comportamiento de las componentes del desgaste al simetrizar ejes geométricos o al cambiar el plano de trabajo y para la compensación de la temperatura en la dirección de la herramienta.

Valores de desgaste

Cuando a partir de ahora se haga referencia a valores de desgaste, se entenderá respectivamente la suma de los valores de desgaste propiamente ($TC_DP12 a $TC_DP20) y las correcciones de suma con los valores de desgaste ($SCPX3 a $SCPX11) y de ajuste ($ECPX3 a $ECPX11).

Más información sobre las correcciones de suma en:
Bibliografía:
Manual de funciones Gestión de herramientas

Datos de operador

Dato de operador	Descripción
DO42900 $SC_MIRROR_TOOL_LENGTH	Simetrizar componentes longitudinales de herramientas y componentes del acotado básico.
DO42910 $SC_MIRROR_TOOL_WEAR	Simetrizar los valores de desgaste de las componentes longitudinales de herramienta.
DO42920 $SC_WEAR_SIGN_CUTPOS	Valoración del signo de las componentes de desgaste dependiendo de la posición del filo.
DO42930 $SC_WEAR_SIGN	Invierte el signo de las dimensiones de desgaste.
DO42935 $SC_WEAR_TRANSFORM	Transformación de los valores de desgaste.
DO42940 $SC_TOOL_LENGTH_CONST	Asignación de las componentes longitudinales de herramienta a los ejes geométricos.
DO42950 $SC_TOOL_LENGTH_TYPE	Asignación de las componentes longitudinales de herramienta independientemente del tipo de herramienta.
DO42960 $SC_TOOL_TEMP_COMP	Valor de compensación de temperatura en la dirección de la herramienta. También está activo si existe una orientación de herramienta

Bibliografía

Manual de funciones básicas; Corrección de herramientas (W1)

Información adicional

Validez de los datos de operador modificados

La nueva valoración de los datos de operador tras una modificación de los componentes de la herramienta se realiza la próxima vez que se seleccione el filo de la herramienta. Si la herramienta para la que se desean modificar los datos esta siendo activada y se pretende validar la modificación, entonces se debe volver a seleccionar la herramienta.

Ello también es válido para el caso en que se modifique la longitud resultante de la herramienta porque se haya cambiado el estado de simetría de un eje. La herramienta se debe volver a seleccionar tras el comando de simetrización; de esta manera, serán válidos las componentes longitudinales modificadas de la herramienta.

Portaherramientas orientable y nuevos datos de operador

Los datos de operador DO42900 a DO42940 no afectan a las componentes de un hipotético portaherramientas orientable activo. Para el cálculo con portaherramientas orientable siempre se tiene en cuenta la herramienta con su longitud total (longitud de herramienta + desgaste + dimensión base). En el cálculo de la longitud total resultante se tienen en cuenta todas las modificaciones generadas mediante datos de operador; es decir, los vectores del portaherramientas orientable son independientes del plano de trabajo.

Nota

Al utilizar un portaherramientas orientable es frecuente definir todas las herramientas para un sistema base no simetrizado, incluso aquellas que sólo se vayan a utilizar para el mecanizado con simetría especular. Para el mecanizado con ejes simetrizados se gira el portaherramientas de tal manera que se describa correctamente la posición de la herramienta. Todos los componentes longitudinales de herramienta actúan automáticamente en la dirección correcta, de modo que no es necesario controlar la valoración de los componentes individuales mediante datos de operador dependiendo del estado de la simetría especular de cada eje.

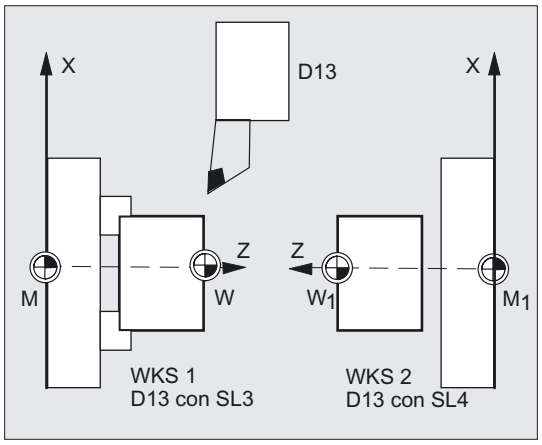
Otras posibilidades de aplicación

Puede tener sentido utilizar la funcionalidad del portaherramientas orientable cuando no sea posible girar la herramienta físicamente en la máquina, aunque se pueda fijar la herramienta con diferentes orientaciones fijas. El acotado de la herramienta se puede realizar de forma unitaria tomando una orientación base y obtener las dimensiones relevantes para el mecanizado mediante el giro de un portaherramientas virtual.

7.3.1 Simetrizar longitudes de herramienta

Función

Con datos de operador DO42900 $SC_MIRROR_TOOL_LENGTH y DO42910 $SC_MIRROR_TOOL_WEAR distintos a cero, las componentes de longitud de herramientas y las componentes del acotado básico se pueden simetrizar con valores de desgaste de los correspondientes ejes.



DO42900 $SC_MIRROR_TOOL_LENGTH

Dato de operador **distinto** de cero:

Invirtiendo el signo se simetrizan los componentes longitudinales de la herramienta ($TC_DP3, $TC_DP4 y $TC_DP5) y los componentes de las dimensiones base ($TC_DP21, $TC_DP22 y $TC_DP23) cuyos ejes han sido simetrizados.

Los valores de desgaste **no** se simetrizan. Si se tienen que simetrizar también estos valores, se debe activar el DO42910 $SC_MIRROR_TOOL_WEAR.

DO42910 $SC_MIRROR_TOOL_WEAR

Dato de operador **distinto** de cero:

Invirtiendo el signo se simetrizan los valores de desgaste de las componentes de longitud de la herramienta cuyos ejes han sido simetrizados.

7.3.2 Evaluación de signos de desgaste

Función

Con los datos de operador DO42920 $SC_WEAR_SIGN_CUTPOS y DO42930 $SC_WEAR_SIGN distintos de cero, se puede invertir la evaluación de signos de las componentes de desgaste.

DO42920 $SC_WEAR_SIGN_CUTPOS

Dato de operador **distinto** de cero:

En herramientas con posición orientada del filo (herramientas de torneado y rectificado, tipos de herramientas 400), la valoración del signo de las componentes del desgaste en el plano de mecanizado depende de la posición del filo. Para herramientas sin posición fija para el corte, este DO carece de sentido.

En la siguiente tabla se marcan con una X las dimensiones cuyos signos se invierten mediante el DO42920 (distinto de 0):

Posición del filo	Longitud 1	Longitud 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Nota

Las valoraciones del signo con los DO42920 y DO42910 son independientes entre sí. Si el signo de una medida se modifica mediante los dos datos de operador, el resultado es que no se altera el signo de la medida.

DO42930 $SC_WEAR_SIGN

Dato de operador **distinto** de cero:

Se invierte el signo de todas las medidas de desgaste. Actúa tanto para la longitud de la herramienta como para el resto de dimensiones como el radio de la herramienta, radio de redondeo, etc.

Si se introduce un valor de desgaste positivo, entonces la herramienta se hace mas "corta" y más "fina", ver el apartado "Corrección de herramienta -Tratamiento especial", Validez de los datos de operador modificados".

7.3.3 Sistema de coordenadas del mecanizado activo (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

Función

En función de la cinemática de la máquina o de la existencia de un portaherramientas orientable, los valores de desgaste medidos en uno de estos sistemas de coordenadas se trasladan a un sistema de coordenadas apropiado o se transforman.

Sistemas de coordenadas del mecanizado activo

De los siguientes sistemas de coordenadas pueden surgir offsets de la longitud de herramienta que incluyen el componente longitudinal de herramienta "Desgaste" a través del correspondiente código G del grupo 56 en una herramienta activa.

- Sistema de coordenadas de máquina (MKS)
- Sistema de coordenadas básico (BKS)
- Sistema de coordenadas de pieza (WKS)
- Sistema de coordenadas de herramienta (TCS)
- Sistema de coordenadas de herramienta de la transformación cinemática (KCS)

Sintaxis

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

Descripción

TOWSTD	Valor de posición preferencial para correcciones en la longitud de la herramienta Valor de desgaste
TOWMCS	Correcciones en la longitud de la herramienta en el MKS
TOWWCS	Correcciones en la longitud de la herramienta en el WKS
TOWBCS	Correcciones en la longitud de la herramienta en el BKS
TOWTCS	Correcciones de la longitud de la herramienta en el punto de referencia del portaherramientas (portaherramientas orientable)
TOWKCS	Correcciones de la longitud de la herramienta del cabezal de herramientas (transformación cinemática)

Información adicional

Características distintivas

En la siguiente tabla se representan las principales características de distinción:

Código G	Valor de desgaste	Portaherramientas orientable activo
TOWSTD	Valor preferencial, longitud de la herramienta	Los valores de desgaste están sujetos al giro.
TOWMCS	Valor de desgaste en el MKS. TOWMCS es idéntico a TOWSTD si no está activo ningún portaherramientas orientable.	Sólo gira el vector de la longitud de herramienta resultante, sin tener en cuenta el desgaste.
TOWWCS	El valor de desgaste se convierte en el WKS al MKS.	El vector de herramienta se calcula como en TOWMCS, sin tener en cuenta el desgaste.
TOWBCS	El valor de desgaste se convierte en el BKS al MKS.	El vector de herramienta se calcula como en TOWMCS, sin tener en cuenta el desgaste.
TOWTCS	El valor de desgaste se convierte en el sistema de coordenadas de herramienta al MKS.	El vector de herramienta se calcula como en TOWMCS, sin tener en cuenta el desgaste.

TOWWCS, TOWBCS, TOWTCS: El vector de desgaste se suma al vector de herramienta.

Transformación lineal

La longitud de herramienta en el MKS sólo se puede definir de forma racional si el MKS resulta del BKS mediante una transformación lineal.

Transformación no lineal

Si, p. ej., con TRANSMIT, está activa una transformada no lineal, se utiliza automáticamente el BKS si se indica el MKS como sistema de coordenadas deseado.

Sin transformación cinemática ni portaherramientas orientable

Si no está activa una transformación cinemática ni un portaherramientas orientable, coinciden los cuatro sistemas de coordenadas, con excepción de WKS. De este modo, sólo el sistema de coordenadas de pieza se distingue de los demás. Dado que sólo se necesitan evaluar longitudes de herramienta, las traslaciones entre los sistemas de coordenadas carecen de importancia.

Bibliografía:
Información adicional sobre la corrección de herramientas en:
Manual de funciones básicas; Corrección de herramientas (W1)

Inclusión de los valores de desgaste

El dato de operador **DO42935 $SC_WEAR_TRANSFORM** establece cuál de los tres componentes de desgaste

- Desgaste
- Correcciones de sumas finas
- Correcciones de suma gruesas

será sometido a giro por una transformada de adaptador o un portaherramientas orientable cuando esté activo uno de los siguientes códigos G:

- `TOWSTD` Ajuste inicial

  para correcciones de la longitud de la herramienta
- `TOWMCS` Valores de desgaste

  en el sistema de coordenadas de máquina (MKS)
- `TOWWCS` Valores de desgaste

  en el sistema de coordenadas de pieza (WKS)
- `TOWBCS` Valores de desgaste (BKS)

  en el sistema de coordenadas básico
- `TOWTCS` Valores de desgaste en el sistema de coordenadas de herramienta en el alojamiento del portaherramientas (T referencia de portaherramientas)
- `TOWKCS` Valores de desgaste en el sistema de coordenadas del cabezal de herramientas con transformación cinética

Nota

En la evaluación de los distintos componentes de desgaste (asignación a los ejes geométricos, evaluación de signos) influyen

- el plano activo
- la transformación del adaptador
- los siguientes datos de operador:
  - DO42910 $SC_MIRROW_TOOL_WEAR
  - DO42920 $SC_WEAR_SIGN_CUTPOS
  - DO42930 $SC_WEAR_SIGN
  - DO42940 $SC_TOOL_LENGTH_CONST
  - DO42950 $SC_TOOL_LENGTH_TYPE

7.3.4 Longitud de herramienta y cambio de plano

Función

Con los datos de operador DO42940 $SC_TOOL_LENGTH_CONST distintos de cero, las componentes de longitud de herramienta como longitud, desgaste y acotado básico, se pueden asignar a los ejes geométricos para herramientas de tornear y rectificar en un cambio de plano.

DO42940 $SC_TOOL_LENGTH_CONST

Dato de operador **distinto** de cero:

No se modifica la asignación de las componentes de longitud de la herramienta (longitud, desgaste y acotado básico) a los ejes geométricos (`G17` - `G19`) al cambiar el plano de trabajo.

La siguiente tabla muestra la asignación de los componentes longitudinales a los ejes geométricos para herramientas de torneado y rectificado (herramientas del tipo 400 a 599):

Contenido	Longitud 1	Longitud 2	Longitud 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Todos los valores distintos de 0 que no sean iguales a alguno de los seis valores presentados se evalúan como el valor 18.

La siguiente tabla muestra la asignación de los componentes longitudinales a los ejes geométricos para el resto de herramientas (hta. del tipo < 400 ó> 599):

Plano de mecanizado	Longitud 1	Longitud 2	Longitud 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Todos los valores distintos de 0 que no sean iguales a alguno de los seis valores presentados se evalúan como el valor 17.

Nota

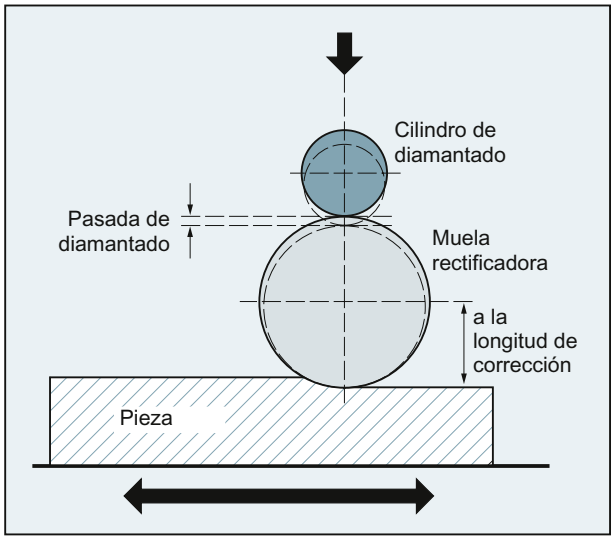
En la presentación de las tablas, se parte de la premisa de que los ejes geométricos 1 a 3 están identificados con los identificadores X, Y, Z. La asignación de una corrección a un eje viene especificada por el número secuencial del eje y no por el identificador del eje.

7.4 Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF)

Función

Con la función "Corrección de herramienta online" activa, en herramientas de rectificado se incluye inmediatamente en el cálculo una corrección longitudinal de herramienta resultado del mecanizado.

Un ejemplo de aplicación es el diamantado de muela, en el que la muela de rectificado es diamantada paralelamente al mecanizado:



La corrección longitudinal de herramienta puede modificarse desde el propio canal de trabajo, o en paralelo desde otro canal (p. ej., canal de diamantado).

Para escribir correctores de herramienta online se utilizan diferentes funciones según el momento deseado del proceso de diamantado:

- Escritura continua por secuencias (`PUTFTOCF`)

  Con `PUTFTOCF` el proceso de diamantado transcurre simultáneamente al mecanizado.

  La corrección de herramienta se modifica en el canal de mecanizado de forma constante siguiendo una función polinómica de 1er, 2º o 3er grado, que debe de ser definida previamente mediante la función `FCTDEF`.

  `PUTFTOCF` está siempre activo por secuencias, es decir, actúa en la secuencia de desplazamiento subsiguiente.

- Escritura continua modal: `ID=1 DO FTOC` (ver "Corrector de herramienta online (FTOC) (Página 622)")

- Escritura discreta (`PUTFTOC`)

  Con `PUTFTOC` el proceso de diamantado no transcurre simultáneamente al mecanizado desde un canal paralelo. El valor de corrección indicado con `PUTFTOC` pasa a estar activo inmediatamente en el canal de destino.

Nota

Los correctores de herramienta online solamente se pueden utilizar con herramientas de rectificado.

Sintaxis

Activar/desactivar el corrector de herramienta online en el canal de destino:

```
FTOCON
...
FTOCOF
```

Escritura del corrector de herramienta:

- Continua por secuencias:

```
FCTDEF(<Función>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
PUTFTOCF(<Función>,<Valor de ref.>,<Parámetro de herramienta>,<Canal>,<Cabezal>)
...
```

- Discreta:

```
PUTFTOC(<Valor de corrección>,<Parámetro de herramienta>,<Canal>,<Cabezal>)
...
```

Descripción

FTOCON:	Activar corrector de herramienta online FTOCON se debe programar en el canal en el que se desea activar el corrector de herramienta online.
FTOCOF:	Cancelar corrector de herramienta online Con FTOCOF se deja de realizar la corrección ; no obstante, el valor completo escrito con PUTFTOC se corrige en los datos de corrección específicos del filo. **Nota:** Para desactivar definitivamente el corrector de herramienta online, tras FTOCOF debe realizarse otra selección/cancelación de la herramienta (T...).

`FCTDEF`: Con `FCTDEF` se define la función polinómica para `PUTFTOCF`.

Parámetros:

`<Función>`:	Número de la función polinómica Tipo: INT
`<LLimit>`:	Límite inferior Tipo: REAL
`<ULimit>`:	Límite superior Tipo: REAL
`<a0> ... <a3>`:	Coeficientes de la función polinómica Tipo: REAL

`PUTFTOCF`: Llamar a la función "Escritura continua por secuencias del corrector de herramienta online"

Parámetros:

`<Función>`:	Número de la función polinómica Tipo: INT **Nota:** Debe coincidir con lo indicado en `FCTDEF`.
`<Valor de ref.>`:	Variable de valor de referencia del que debe deducirse la corrección (p. ej., para modificar el valor real). Tipo: VAR REAL
`<Parámetro de herramienta>`:	Número del parámetro de desgaste (longitud 1, 2 ó 3) en el que debe sumarse el valor de corrección. Tipo: INT
`<Canal>`:	Número del canal en el que debe activarse el corrector de herramienta online. Tipo: INT **Nota:** La indicación sólo es necesaria si la corrección no debe activarse en el canal activo.
`<Cabezal>`:	Número del cabezal para el que debe activarse el corrector de herramienta online. Tipo: INT **Nota:** La indicación sólo es necesaria si, en lugar de la herramienta que se encuentra activa y en uso, se debe corregir una muela rectificadora no activa.

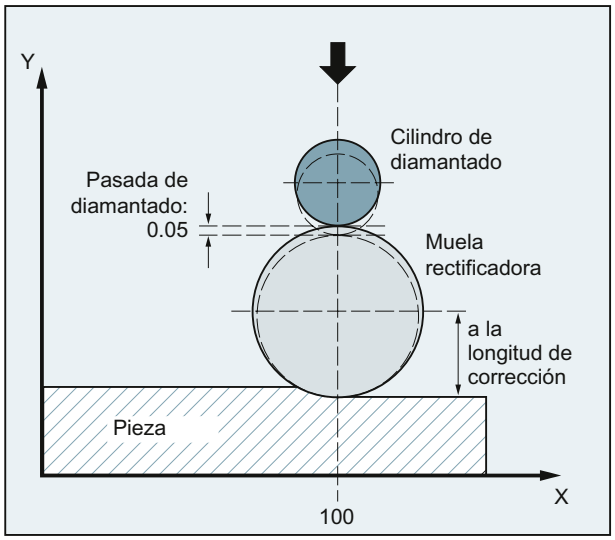
PUTFTOC:	Llamar a la función "Escritura discreta de corrector de herramienta online"	
	Parámetros:	
	<Valor de corrección>:	Valor de corrección que debe sumarse en el parámetro de desgaste. Tipo: REAL
	<Parámetro de herramienta>:	Ver PUTFTOCF
	<Canal>:	Número del canal en el que debe activarse el corrector de herramienta online. Tipo: INT
	<Cabezal>:	Ver PUTFTOCF

Ejemplo

Rectificadora plana con:

- Y: Eje de penetración para muela rectificadora
- V: Eje de penetración para el cilindro de diamantado
- Canal de mecanizado: En el canal 1 con los ejes X, Z, Y
- Canal de diamantado: En el canal 2 con el eje V

Después de comenzar con el desplazamiento de rectificado, en X100 se debe diamantar la muela y reducir su radio en 0,05. El valor del diamantado debe activarse con "Escritura continua del corrector de herramienta online" en la herramienta de rectificado.



Programa de trabajo en el canal 1:

```
Código del programa                 Comentarios
…
N110 G1 G18 F10 G90                 ; Ajuste inicial.
N120 T1 D1                          ; Seleccionar herramienta actual.
N130 S100 M3 X100                   ; Activar cabezal, desplazamiento a posición
                                      inicial.
N140 INIT(2,"DIAMANTE","S")         ; Selección del programa de diamantado por el
                                      canal 2.
N150 START(2)                       ; Arranque del programa de diamantado por el
                                      canal 2.
N160 X200                           ; Desplazamiento a la posición de destino.
N170 FTOCON                         ; Activar corrección Online.
N… G1 X100                          ; Continuar el mecanizado.
N… M30
```

Programa de diamantado en el canal 2:

```
Código del programa                       Comentarios
…
N40 FCTDEF(1,–1000,1000,–$AA_IW[V],1)     ; Definir función: Recta con
                                            pendiente = 1.
N50 PUTFTOCF(1,$AA_IW[V],3,1)             ; Los correctores de herramienta
                                            online escriben de modo continuo: a
                                            partir del desplazamiento del eje V,
                                            se corrige la longitud 3 de la muela
                                            de rectificado actual en el canal 1.
N60 V–0.05 G1 F0.01 G91                   ; Movimiento de penetración para el
                                            diamantado; PUTFTOCF está activo
                                            sólo en esta secuencia.
…
N… M30
```

Información adicional

Indicaciones generales para correctores de herramienta online

En la escritura continua (por ciclo IPO), tras la activación de la función de evaluación se suma cada modificación en la memoria de desgaste (para impedir saltos de valor teórico).

En los dos casos se aplica lo siguiente: El corrector de herramienta online puede activarse por cada canal para cada cabezal y ser efectivo con los parámetros de desgaste de longitud 1, 2 **o bien** 3.

Las longitudes se asignan a los ejes geométricos en función del plano de trabajo actual.

El cabezal se asigna a la herramienta mediante los datos de herramienta en `GWPSON` o `TMON` siempre que no se trate de la muela de rectificado activa.

Siempre se corrige el parámetro de desgaste para el lado de la muela actual, o bien el lado izquierdo de la muela en el caso de herramientas que no estén activas.

Nota

En caso de que se pretenda realizar la misma corrección para varias caras de muela, se deben cargar los valores de forma automática para la segunda cara utilizando reglas de concatenación.

Si se especifican correcciones online para un canal de mecanizado, no se deben modificar los valores de desgaste para la herramienta actual en dicho canal desde el programa de trabajo o mediante manejo.

El corrector de herramienta online también se tiene en cuenta para la velocidad periférica constante de la muela (`SUG`) y para la vigilancia de herramienta (`TMON`).

7.5 Activación correcciones de herramienta 3D (CUT3DC..., CUT3DF...)

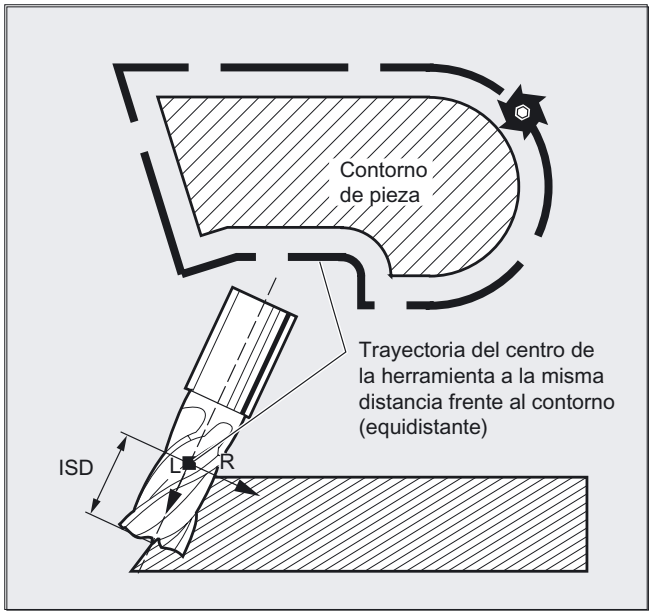
7.5.1 Activar correcciones de herramientas 3D (CUT3DC, CUT3DF, CUT3DFS, CUT3DFF, ISD)

Función

Para la corrección del radio de la herramienta para herramientas cilíndricas se tiene en cuenta la orientación variable de la herramienta.

Para la activación de la corrección del radio de herramienta en 3D se utilizan las mismas instrucciones de programación comandos que para la corrección del radio de herramienta 2D. Con `G41`/`G42` se especifica la corrección izquierda/derecha en la dirección del desplazamiento. El método de aproximación es siempre `NORM`. La corrección de radio de herramienta 3D sólo actúa si está activa la transformada de 5 ejes.

La corrección del radio de herramienta en 3D también se puede definir como corrección 5D, ya que en este caso se dispone de 5 grados de libertad para posicionar la herramienta.



Diferencia entre corrección del radio de la herramienta 2 1/2D y 3D

En la corrección del radio de herramienta 3D se puede cambiar la orientación de la herramienta. En la corrección de radio de herramienta 2 1/2D sólo se calcula con una herramienta con orientación constante.

Sintaxis

```
CUT3DC
CUT3DFS
CUT3DFF
CUT3DF
ISD=<Valor>
```

Descripción

`CUT3DC`	Activación de los correctores del radio de herramienta 3D para fresado periférico
`CUT3DFS`	Corrector de herramienta D para fresados frontales con orientación constante. La orientación de la herramienta se define mediante `G17` - `G19` y no depende de los frames.
`CUT3DFF`	Corrección de herramienta D para fresados frontales con orientación constante. La orientación de la herramienta es la dirección definida mediante `G17` - `G19` y en algunos casos girada por un frame.
`CUT3DF`	Corrector de herramienta D para fresados frontales con cambio de orientación. (Solamente para transformada de 5 ejes activa).
`G40 X... Y... Z...`	Para la desactivación: Secuencia lineal `G0`/`G1` con ejes geométricos
`ISD`	Profundidad penetración

Nota

Los comandos son válidos de forma modal y pertenecen al mismo grupo que `CUT2D` y `CUT2DF`. La desactivación se produce con el primer desplazamiento que se realice en el plano de trabajo actual. Esto se aplica siempre para `G40` y es independiente del comando `CUT`.

Se permiten utilizar secuencias intermedias con la corrección del radio de herramienta 3D activa. Valen las definiciones de la corrección de radio de herramienta 2 1/2D.

Condiciones

- **G450/G451 y DISC**

  En los vértices exteriores se inserta siempre una secuencia circular. `G450/G451` son irrelevantes.
  El comando `DISC` no se evalúa.

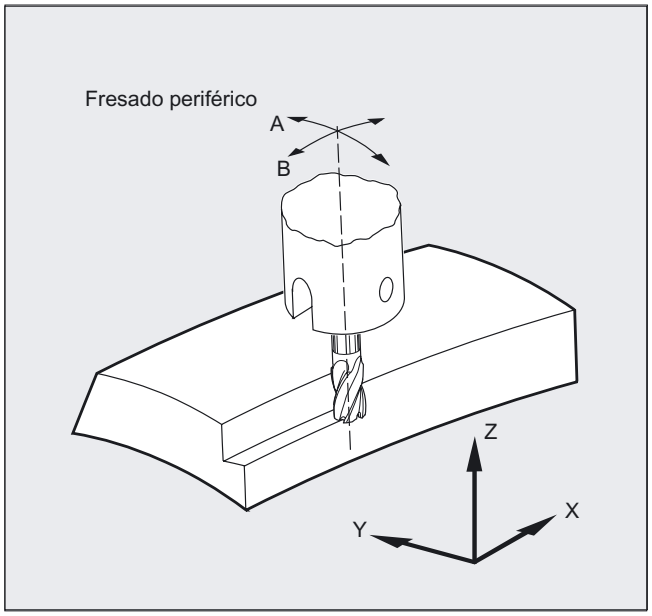
Ejemplo

```
Código del programa              Comentarios
N10 A0 B0 X0 Y0 Z0 F5000
N20 T1 D1                        ; Llamada a herramienta, llamada a los valores de
                                   corrección de herramienta.
N30 TRAORI(1)                    ; Seleccionar la transformada
N40 CUT3DC                       ; Seleccionar la corrección de radio de herramienta
                                   3D
N50 G42 X10 Y10                  ; Seleccionar la corrección de radio de herramienta
N60 X60
N70 …
```

7.5.2 Corrección de herramienta 3D: Fresado periférico, fresado frontal

Fresado periférico

La variante del fresado periférico aquí utilizada se realiza definiendo una trayectoria y la orientación de la misma. En este tipo de mecanizado la forma de la herramienta carece de significado para la trayectoria. Sólo cuenta el radio en el punto de contacto de la herramienta.

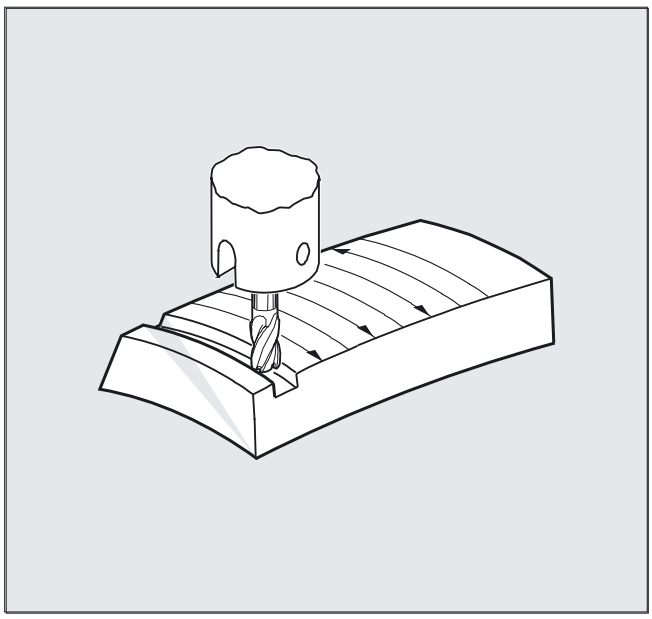


Nota

La función de corrección de radio de herramienta 3D se limita a las herramientas cilíndricas.

Fresado frontal

Para este tipo de fresado 3D se requiere la descripción por líneas de las trayectorias 3D sobre la superficie de la pieza. Los cálculos se realizan teniendo en cuenta la forma y las dimensiones de la herramienta, generalmente en sistemas CAM. El postprocesador escribe en el programa de pieza, además de las secuencias CN, las orientaciones de la herramienta (si está activa la transformada de 5 ejes) y el código G para la corrección de herramienta 3D deseada. Esto ofrece al operador de la máquina la posibilidad de utilizar herramientas ligeramente más pequeñas, apartándose de la herramienta utilizada para el cálculo de las trayectorias de CN.



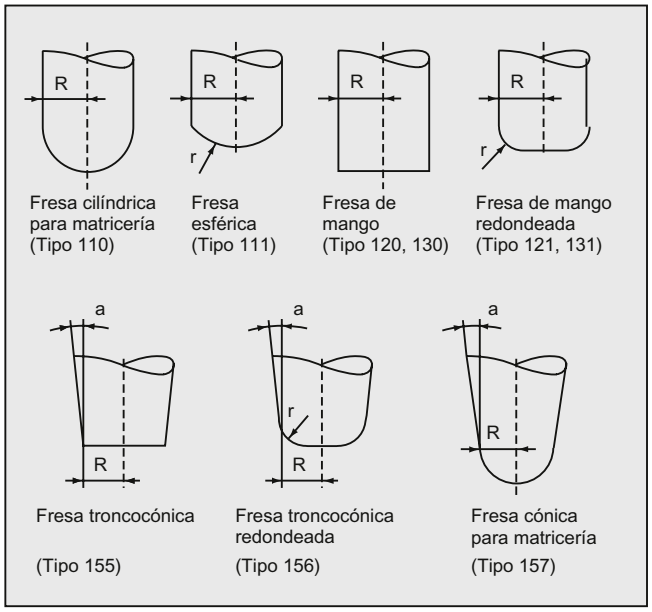
Ejemplo:

Las secuencias CN se han calculado con una fresa de 10 mm. En este caso también se podría mecanizar con un diámetro de fresa de 9,9 mm, aunque esto comportaría una alteración del perfil de rugosidad.

7.5.3 Corrección de herramienta 3D: Formas y datos de herramientas para el fresado frontal

Formas de fresa, datos de herramienta

A continuación se enumeran las formas de herramienta y valores límite de los datos de herramienta posibles para las fresas frontales. La forma del mango de la herramienta no se tiene en cuenta. Los tipos de herramienta 120 y 156 tienen idéntico efecto.



Si el número de tipo indicado en el programa CN difiere del que se muestra en la figura, el sistema utilizará automáticamente el tipo de herramienta 110 (fresa cilíndrica para matricería). Cuando se exceden los valores límites para los datos de la herramienta se emitirá una alarma.

Tipos de fresas	Nº de tipo	R	r	a
Fresa cilíndrica para matricería	110	> 0	-	-
Fresa esférica	111	> 0	> R	-
Fresa normal, fresa angular	120, 130	> 0	-	-
Fresa normal, fresa angular redondeada	121, 131	> r	> 0	-
Fresa de cono truncado	155	> 0	-	> 0
Fresa de cono truncado con redondeado	156	> 0	> 0	> 0
Fresa cónica para matricería	157	> 0	-	> 0

R = Radio del vástago (radio de la herramienta)

r = Radio de redondeo

a = Ángulo entre el eje longitudinal de la herramienta y el extremo superior de la superficie del toro

\- = No se evalúa

Datos de herramienta	Parámetros de herramienta	
Medidas de herramienta	Geometría	Desgaste
R	$TC_DP6	$TC_DP15
r	$TC_DP7	$TC_DP16
a	$TC_DP11	$TC_DP20

Corrección longitudinal de herramienta

Para la corrección de la longitud se toma como punto de referencia la punta de la herramienta (punto de intersección entre el eje longitudinal y la superficie).

Corrección de herramienta 3D, cambio de herramienta

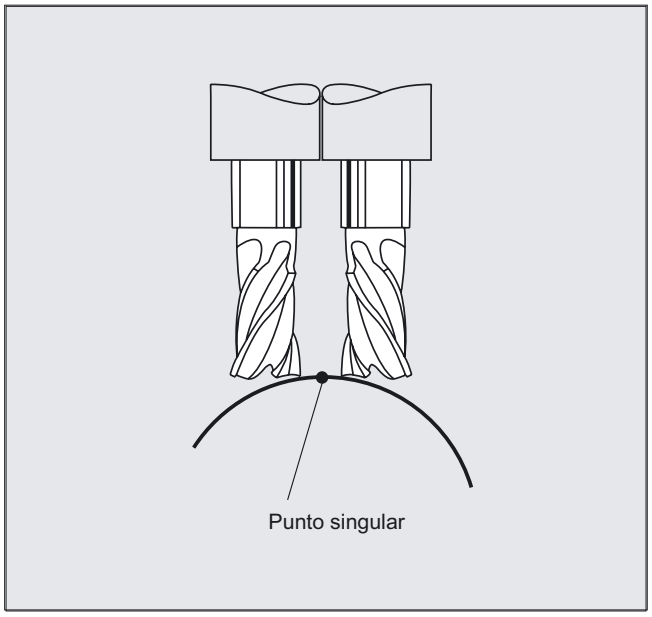
Para especificar una nueva herramienta con diferente geometría (R, r, a) u otra forma diferente, debe programarse necesariamente `G41` o `G42` (transición de `G40` a `G41` o `G42`, nueva programación de `G41` o `G42`). Los demás datos de herramienta, como p. ej. las longitudes, no están sometidos a esta regla, de modo que pueden activarse dichas herramientas sin necesidad de reprogramar `G41` o `G42`.

7.5.4 Corrección de herramienta 3D: Corrección sobre la trayectoria, curvatura de la trayectoria, profundidad de penetración (CUT3DC, ISD)

Función

Corrección sobre la trayectoria

Para fresados frontales se debe de tener en cuenta el caso de salto del punto de contacto entre herramienta y pieza. Como en este ejemplo de mecanizado de una superficie convexa con herramienta vertical. El ejemplo mostrado en la figura se puede considerar como un caso extremo.



El control vigila la aparición de este caso extremo, de forma que en base al ajuste de los ángulos entre la herramienta y los vectores perpendiculares a la superficie se detectan saltos en el punto de mecanizado. En estos puntos el control numérico inserta secuencias lineales de manera que se puedan realizar los desplazamientos.

Para el cálculo de las secuencias lineales se deben de introducir en datos de máquina rangos permitidos para los valores del ángulo de alabeo. En el caso de que se superen los valores límite indicados en los datos de máquina para dichos ángulos, el sistema emite una alarma.

Curvatura de la trayectoria

La curvatura de la trayectoria no se vigila. Es aconsejable utilizar únicamente herramientas que puedan trabajar sin producir errores en el contorno.

Profundidad de penetración (ISD)

La profundidad de penetración ISD sólo se evalúa estando activa la corrección de radio de herramienta 3D.

Con el comando `ISD` (Insertion Depth) se programa la profundidad de penetración de la herramienta durante el fresado periférico. Esto hace posible modificar la posición del punto de mecanizado en la envolvente de la herramienta.

Sintaxis

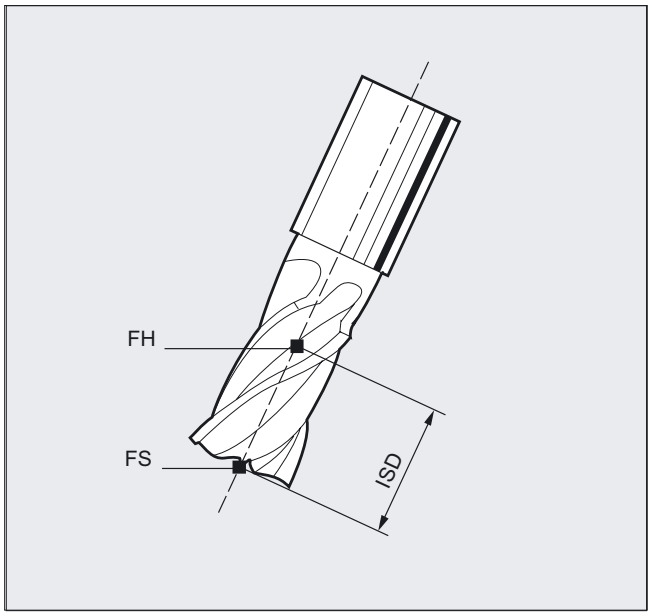
Corrección de herramienta 3D fresado periférico
```
CUT3DC
ISD=<Valor>
```

Descripción

`CUT3DC`	Activar la corrección de herramienta 3D para el fresado periférico, p. ej. para el fresado de cajas con paredes laterales inclinadas.
`ISD`	Con el comando `ISD` se indica la distancia (`<Valor>`) entre la punta de la fresa (FS) y el punto auxiliar de la fresa (FH).

Punto auxiliar de la fresa

El punto auxiliar de la fresa (FH) se genera mediante la proyección del punto de mecanizado programado sobre el eje de la herramienta.



Información adicional

Fresado de cajas con paredes laterales inclinadas para el fresado periférico con CUT3DC

En esta corrección de radio de herramienta 3D se compensa una desviación del radio de la fresa, penetrando en dirección de la normal de superficie de la superficie mecanizada. Sin embargo, el plano en el que se halla el lado frontal de la fresa no se modifica si la profundidad de penetración `ISD` no cambia. En tal caso, una fresa que p. ej. tuviera un radio más pequeño que una herramienta normalizada, no alcanzaría el fondo de la caja, que constituye también la superficie de limitación. Para la penetración automática de la herramienta, el control tiene que conocer esta superficie de limitación, ver apartado "Fresados periféricos 3D con superficies de limitación".

Información adicional sobre la vigilancia de colisión en:
Bibliografía:
Manual de programación, Fundamentos; apartado "Correcciones de herramienta".

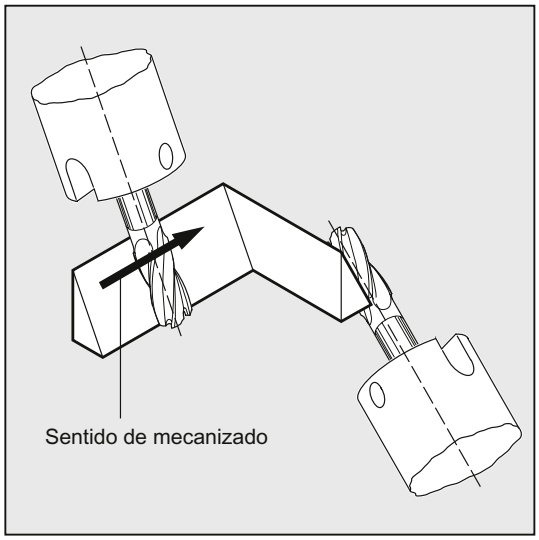
7.5.5 Corrección de herramienta 3D: Esquinas interiores/exteriores y método de punto de intersección (G450/G451)

Función

Esquinas interiores/exteriores

Las esquinas exteriores e interiores tienen tratamientos diferentes. La designación de una esquina interior o exterior depende de la orientación de la herramienta.

Al realizar cambios en la orientación de la herramienta en una esquina, puede suceder que durante el mecanizado cambie el tipo de esquina. En este caso se interrumpe el mecanizado con un aviso de error.



Sintaxis

```
G450
G451
```

Descripción

G450	Circunferencia de transición (la herramienta pasa por las esquinas de la pieza siguiendo una trayectoria circular)
G451	Punto de intersección de las equidistantes (la herramienta sale en el ángulo de la pieza)

Información adicional

Método de punto de intersección para contorno en 3D

Ahora, en el fresado periférico 3D se evalúa en las esquinas exteriores el código G `G450/G451`, de modo que es posible posicionar en el punto de intersección de las curvas de offset. Hasta SW 4 se ha insertado siempre un círculo en las esquinas exteriores. El método de punto de intersección es particularmente ventajoso en típicos programas 3D generados con CAD. Estos se componen a menudo de secuencias cortas de rectas (para la aproximación de curvas lisas), en las cuales las transiciones entre las secuencias contiguas son prácticamente tangenciales.

En la corrección del radio de la herramienta del lado exterior del contorno, hasta la fecha se han insertado, en principio, circunferencias para esquivar las esquinas exteriores. Como estas secuencias en caso de transiciones tangenciales se convierten en muy cortas, se producen unas interrupciones de velocidad no deseadas.

En estos casos las dos curvas afectadas se prolongan análogamente a la corrección de radio 2 ½ D y se efectúa el posicionamiento en el punto de intersección de las dos curvas prolongadas.

El punto de intersección se determina, prolongando las curvas offset de las dos secuencias afectadas y determinando su punto de intersección en el plano vertical respecto a la orientación de herramienta en la esquina. Si no existe dicho punto de intersección, la esquina recibirá el mismo tratamiento que hasta el momento, es decir, se insertará un círculo.

Información adicional sobre el método de punto de intersección:
Bibliografía:
Manual de funciones especiales; Corrección de radio de herramienta 3D (W5)

7.5.6 Corrección de herramienta 3D: Fresado periférico 3D con superficies de limitación

Adaptación del fresado periférico 3D a las particularidades de los programas CAD

Por regla general, los programas CN generados por sistemas CAD aproximan la trayectoria de centro de las herramientas normalizadas con un gran número de secuencias lineales cortas. Para que las secuencias de muchos contornos de pieza generadas de tal modo simulen de la manera más precisa posible el contorno original, es necesario llevar a cabo una serie de adaptaciones en el programa de pieza.

Es necesario sustituir algunos datos importantes que, pese a ser necesarios para una óptima corrección, ya no están disponibles en el programa de pieza. A continuación se muestran métodos habituales para compensar transiciones críticas, ya sea directamente en el programa de pieza o determinando el contorno real (p. ej. mediante la penetración de la herramienta).

Aplicaciones

Además de los casos de aplicación típicos, en los que la trayectoria del centro se describe con una herramienta real, en lugar de la herramienta normalizada, se tratan también las herramientas cilíndricas con corrección de herramienta 3D. En este caso, la trayectoria programada hace referencia al contorno de la superficie de mecanizado. La superficie de limitación aplicable en este caso es independiente de la herramienta. Como en la corrección de radio de herramienta convencional, para el cálculo del offset vertical con la superficie de limitación se toma en consideración el radio total.

7.5.7 Corrección de herramienta 3D: Consideración de una superficie de limitación (CUT3DCC, CUT3DCCD)

Función

Fresado periférico 3D con herramientas reales

En el fresado periférico 3D con modificación continua o constante de la orientación de la herramienta se programa frecuentemente la trayectoria del centro de la herramienta para una herramienta normalizada definida. Dado que, en la práctica, ocurre a menudo que no se dispone de las herramientas normalizadas adecuadas, se puede emplear una herramienta que no difiera excesivamente de una herramienta normalizada.

Con `CUT3DCCD` se considera para una herramienta diferencial real una superficie de limitación que describiría la herramienta normalizada programada. El programa CN describe la trayectoria del centro de la herramienta normalizada.

Con `CUT3DCC` se considera, en caso de utilizar herramientas cilíndricas, una superficie de limitación que habría alcanzado la herramienta normalizada programada. El programa CN describe el contorno en la superficie de mecanizado.

Sintaxis

```
CUT3DCCD
CUT3DCC
```

Descripción

`CUT3DCCD`	Activar la corrección de herramienta 3D para fresados periféricos con superficies de limitación con herramienta diferencial en la trayectoria del centro de la herramienta: Penetración a la superficie de limitación.
`CUT3DCC`	Activar la corrección de herramienta 3D para fresado periférico con superficies de limitación con corrección de radio 3D: Contorno en la superficie de mecanizado

Nota

Corrección del radio de herramienta con G41, G42

Para la corrección del radio de herramienta con `G41`, `G42` con `CUT3DCCD` o `CUT3DCC` activo tiene que existir la opción "Transformada de orientación".

Herramientas normalizadas con redondeo de esquinas

El redondeo de esquinas de la herramienta normalizada se describe con el parámetro de herramienta `$TC_DP7`. Del parámetro de herramienta `$TC_DP16` resulta la desviación del redondeo de esquinas de la herramienta real frente a la herramienta normalizada.

Ejemplo:

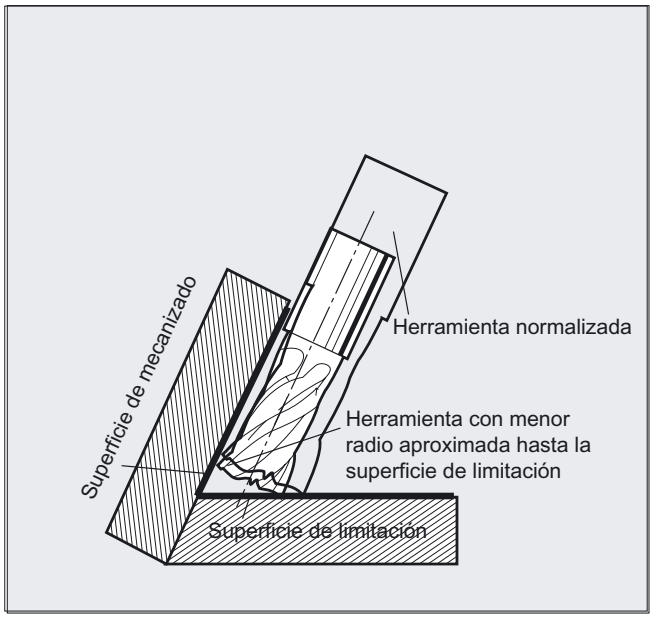
Dimensiones de herramienta para una fresa tórica con radio reducido frente a la herramienta normalizada.

Tipo de herramienta	R = Radio del vástago	r = Radio de redondeo
Herramienta normalizada con redondeo de esquinas	R = $TC_DP6	r = $TC_DP7
Herramienta real con redondeo de esquinas: tipos de herramienta, fresas tóricas 121 y 131 (fresas de mango)	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16
En este ejemplo son negativos tanto $TC_DP15 + OFFN como $TC_DP16. Se evalúa el tipo de herramienta ($TC_DP1).		
Sólo se admiten tipos de fresa con mango cilíndrico (fresa cilíndrica o de mango), así como fresas tóricas (tipo 121 y 131) y, en caso extremo, la fresa cilíndrica de matrices (tipo 110).	Con estos tipos de fresa admitidos, el radio de redondeo r es igual al radio del vástago R. Todos los demás tipos de herramienta admitidos se interpretan como fresas cilíndricas; no se evalúa una eventual medida indicada para el redondeo de esquinas.	
Se admiten todos los tipos de herramienta de los números 1 - 399, con excepción de los números 111 y 155 a 157.		

Información adicional

Trayectoria del centro de la herramienta con penetración hasta la superficie de limitación CUT3DCCD

Si se utiliza una herramienta con un radio inferior al de la herramienta normalizada, la fresa que penetra en dirección longitudinal se sigue desplazando hasta que vuelve a tocar el fondo de la caja. De este modo, la esquina formada por las superficies de mecanizado y de limitación se vacía tanto como lo permita la herramienta. Se trata de un método de mecanización mixto que combina el fresado periférico y el fresado frontal. De modo análogo a la herramienta con radio reducido, en el caso de una herramienta con el radio aumentado, la penetración se produce en la dirección opuesta.



A diferencia de todas las demás correcciones de herramienta del grupo del código G 22, un parámetro de herramienta `$TC_DP6` especificado para `CUT3DCCD` no influye en el radio de la herramienta y por lo tanto tampoco en la corrección resultante.

El offset de corrección resulta de la suma de:

- Valor de desgaste del radio de la herramienta (parámetro de herramienta `$TC_DP15`)
- y un offset de herramienta programado para el cálculo del offset vertical con la superficie de limitación `OFFN`.

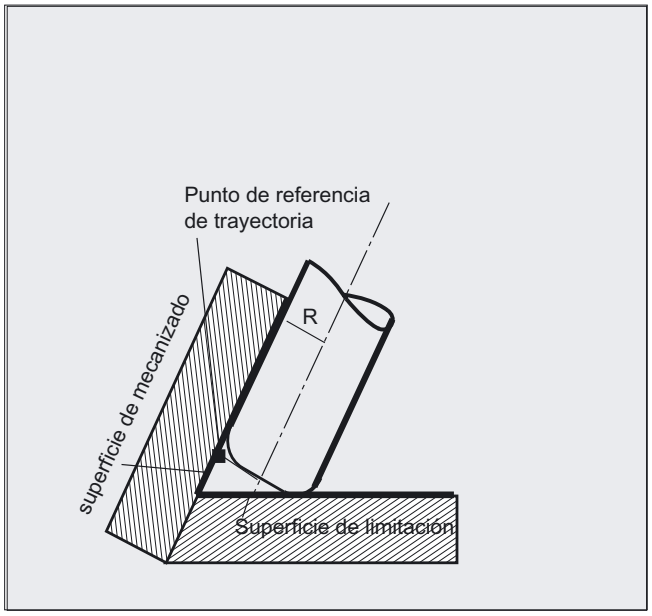
No se puede determinar a partir del programa de pieza generado si la superficie a mecanizar se halla a la izquierda o a la derecha de la trayectoria. Por ello se asume un radio positivo y un valor de desgaste negativo de la herramienta original. Un valor de desgaste negativo genera siempre una herramienta con diámetro reducido.

Uso de herramientas cilíndricas

Para el uso de herramientas cilíndricas, sólo es necesaria la penetración en caso de que la superficie de mecanizado y la de limitación formen un ángulo agudo (inferior a 90 grados). Si se usan fresas tóricas (cilindro con redondeo de esquinas), tanto con ángulo agudo como con ángulo obtuso será necesaria una penetración en la dirección longitudinal de la herramienta.

Corrección de radio 3D con CUT3DCC, contorno en la superficie de mecanizado

Si `CUT3DCC` está activo con una fresa tórica, la trayectoria programada hace referencia a una fresa cilíndrica ficticia del mismo diámetro. En la figura siguiente se muestra el punto de referencia de la trayectoria resultante utilizando una fresa tórica.



Es admisible que el ángulo entre las superficies de mecanizado y de limitación se transforme de agudo a obtuso o viceversa incluso dentro de una misma secuencia.

La herramienta real utilizada puede ser mayor o menor que la herramienta normalizada. Sin embargo, el radio de redondeo resultante no debe ser negativo, y debe mantenerse el signo del radio de herramienta resultante.

Con `CUT3DCC`, el programa de pieza CN hace referencia al contorno en la superficie de mecanizado. Como sucede con la corrección de radio de herramienta convencional, se toma en consideración el radio total, compuesto por la suma de:

- el radio de la herramienta (parámetro de herramienta `$TC_DP6`)
- el valor de desgaste (parámetro de herramienta `$TC_DP15`)
- y un offset de herramienta programado para el cálculo del offset vertical con la superficie de limitación `OFFN`.

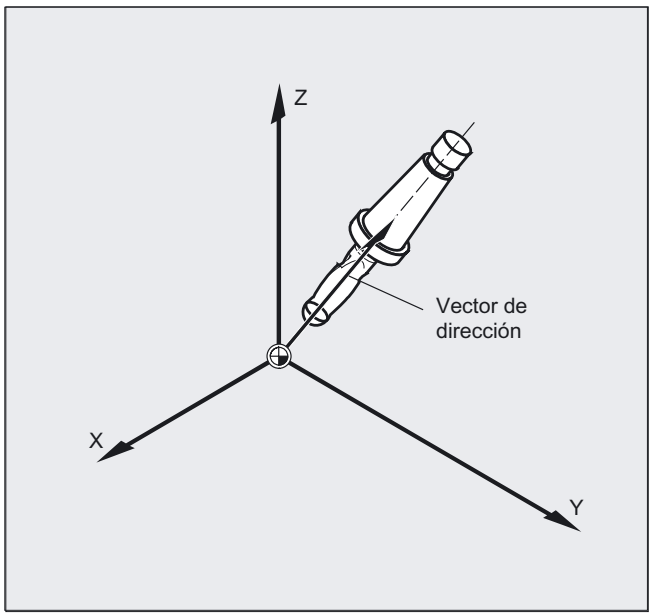
La posición de la superficie de limitación se determina a partir de la diferencia entre ambos valores:

- Dimensiones de la herramienta normalizada
- Radio de la herramienta (parámetro de herramienta `$TC_DP6`)

7.6 Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Función

Se entiende por orientación de la herramienta la alineación geométrica de la herramienta en el espacio. En una máquina de mecanizado de 5 ejes es posible definir la orientación de la herramienta mediante comandos de programa.



Los movimientos de transición de secuencias de la orientación activados con `OSD` y `OST` se producen de forma diferente según el tipo de interpolación para la orientación de la herramienta.

En el caso de interpolación activa vectorial, el desarrollo alisado de la orientación se interpola también por medio de interpolación vectorial. Por el contrario, en el caso de interpolación activa de ejes giratorios, la orientación se alisa directamente mediante movimientos de los ejes giratorios.

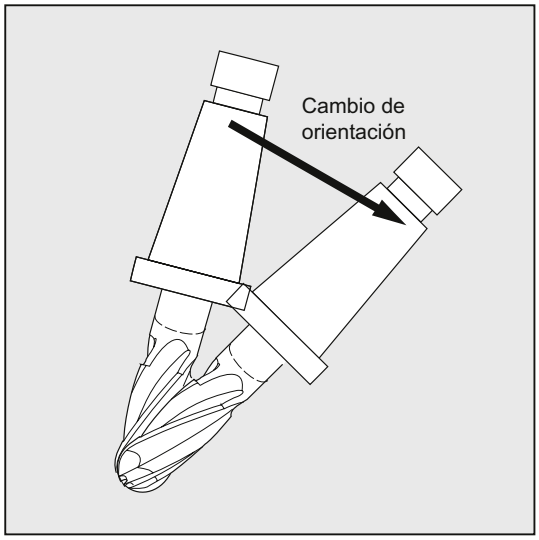
Programación

Programar el cambio de orientación:

Se puede programar un cambio en la orientación de la herramienta mediante:

- Programación directa de los ejes giratorios `A`, `B`, `C` (interpolación de ejes giratorios)
- Ángulo euleriano o RPY
- Vector de dirección (interpolación vectorial a través de la indicación de `A3` o `B3` o `C3`)
- `LEAD/TILT` (fresado frontal)

El sistema de coordenadas de referencia puede ser el sistema de coordenadas de máquina (ORIMKS) o bien el sistema de coordenadas actualmente asociado a la pieza (ORIWKS).



Programar la orientación de la herramienta:

Comando	Descripción
ORIC:	Orientación y movimiento interpolado paralelos
ORID:	Orientación y movimiento interpolado sucesivos
OSOF:	Sin alisado de la orientación
OSC:	Orientación constante
OSS:	Alisado de la orientación sólo al principio de la secuencia
OSSE:	Alisado de la orientación al principio y al final de la secuencia
ORIS:	Velocidad del cambio de orientación con el alisado de orientación activado en grados por mm (válido para OSS y OSSE)
OSD:	Matado de esquinas de la orientación mediante la especificación de la longitud de matado de esquinas con el dato de operador: DO42674 $SC_ORI_SMOOTH_DIST
OST:	Matado de esquinas de la orientación mediante la especificación de la tolerancia angular en grados en la interpolación vectorial con el dato de operador: DO42676 $SC_ORI_SMOOTH_TOL En el caso de interpolación de ejes giratorios, la tolerancia definida se toma como desviación máxima de los ejes de orientación.

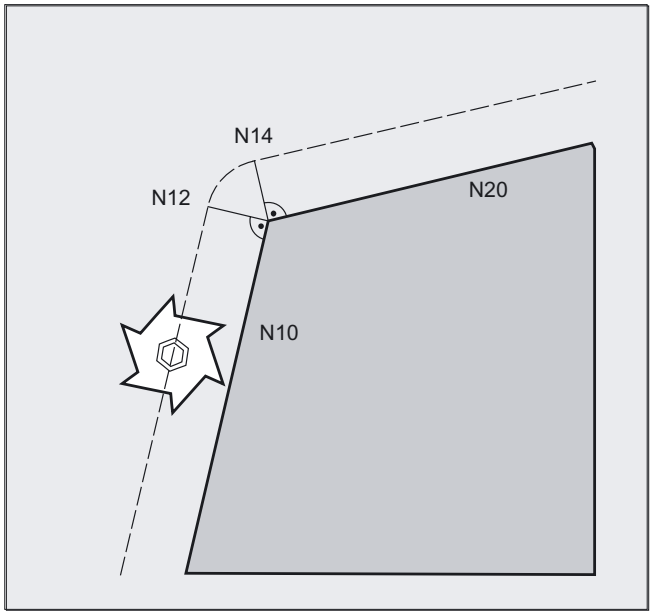
Nota

Todos los comandos para el matado de esquinas de la orientación de herramienta (OSOF, OSC, OSS, OSSE, OSD y OST) están reunidos en el grupo de funciones G 34. Son modalmente activos, es decir, sólo puede estar activo uno de estos comandos cada vez.

Ejemplos

Ejemplo 1: ORIC

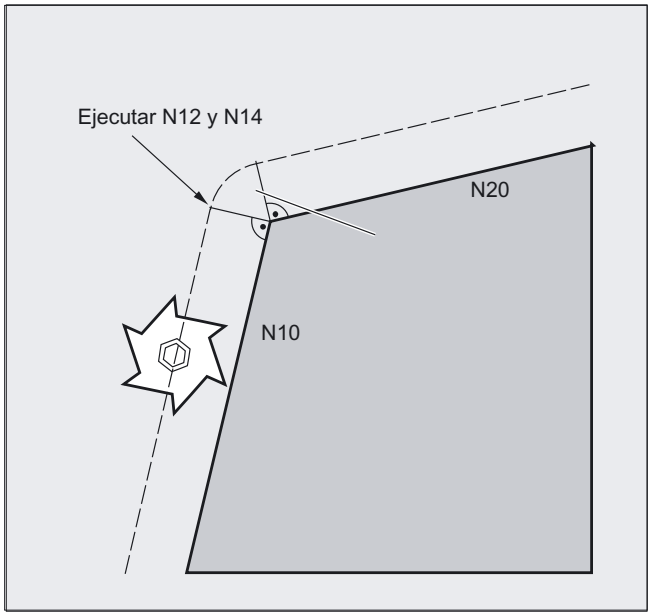
Si entre las secuencias de desplazamiento `N10` y `N20` se han programado dos o más secuencias con cambios de orientación (p. ej. `A2=... B2=... C2=...`) y está activo `ORIC`, la secuencia circular insertada se dividirá entre dichas secuencias intermedias en función del valor de los cambios de ángulo.



Código del programa	Comentarios
`ORIC`	
`N8 A2=… B2=… C2=…`	
`N10 X… Y… Z…`	
`N12 C2=… B2=…`	`; La secuencia circular que se inserta en la esquina`
`N14 C2=… B2=…`	`exterior se reparte entre N12 y N14, en función de la modificación de la orientación. El movimiento circular y el cambio de orientación se ejecutan paralelamente.`
`N20 X =…Y=… Z=… G1 F200`	

Ejemplo 2: ORID

Cuando ORID está activo, todas las secuencias entre las dos secuencias de desplazamiento se ejecutan al final de la primera secuencia de desplazamiento. La secuencia circular con orientación constante se ejecuta inmediatamente antes de la segunda secuencia de desplazamiento.



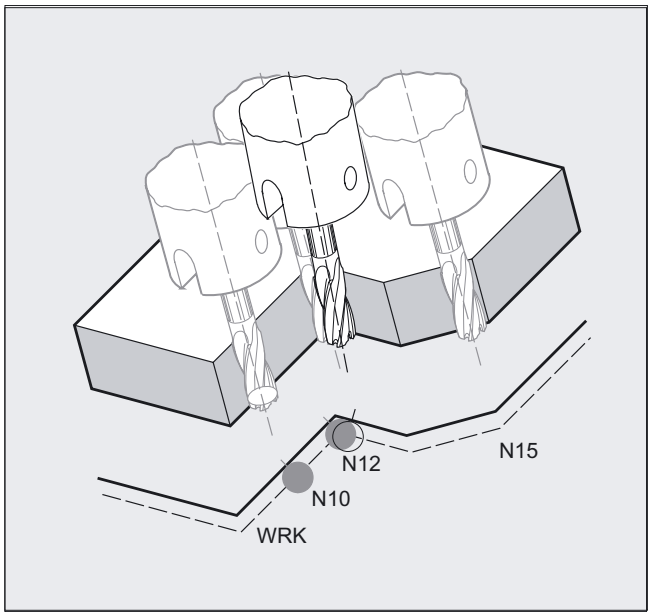
Código del programa	**Comentarios**
ORID	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 A2=… B2=… C2=…	; Las secuencias N12 y N14 se ejecutan al final de N10. A continuación se ejecuta la secuencia circular con la orientación actual.
N14 M20	; Funciones auxiliares, etc.
N20 X… Y… Z…	

Nota

El tipo del cambio de orientación en una esquina exterior viene determinado por el comando de programa activo en la primera secuencia de desplazamiento de una esquina exterior.

Sin cambio de orientación: Si la orientación no se modifica en el límite de la secuencia, la sección de la herramienta es un círculo que toca ambos contornos.

Ejemplo 3: Cambio de la orientación en una esquina interior



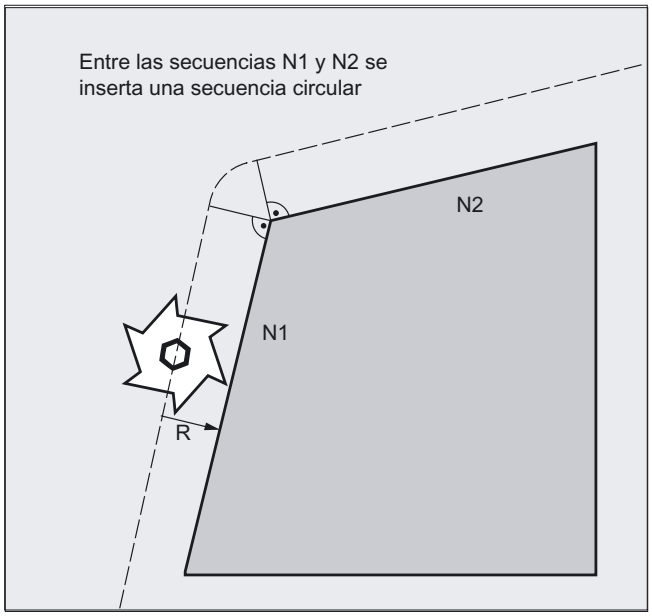
Código del programa
ORIC
N10 X …Y… Z… G1 F500
N12 X …Y… Z… A2=… B2=… C2=…
N15 X …Y… Z… A2=… B2=… C2=…

Información adicional

Comportamiento en esquinas exteriores

En una esquina exterior se inserta siempre una secuencia circular con el radio de la fresa.

Con los comandos `ORIC` y `ORID` se puede definir si los cambios de orientación programados entre las secuencias `N1` y `N2` se ejecutan antes del inicio de la secuencia circular insertada o simultáneamente con ésta.



Si es necesario efectuar un cambio de orientación en esquinas exteriores, éste se puede realizar paralelamente a la interpolación o por separado con el movimiento interpolado.

Con `ORID` se ejecutan primero las secuencias insertadas sin desplazamiento de trayectoria. La secuencia circular se inserta inmediatamente delante de la segunda de las dos secuencias de desplazamiento que forman la esquina.

Si, en una esquina exterior, se han insertado varias secuencias de orientación y seleccionado `ORIC`, el movimiento circular se distribuye entre las distintas secuencias insertadas en función de las magnitudes del cambio de orientación.

Matado de esquinas de la orientación con OSD u OST

En el matado de esquinas con G642, la desviación máxima para los ejes de contorno y los ejes de orientación no puede ser muy diferente. La tolerancia menor de ambos determina la forma del movimiento de matado de esquinas o la tolerancia angular para alisar con relativa intensidad el desarrollo de la orientación, sin que ello implique desviaciones importantes del contorno.

Mediante la activación de OSD y OST es posible alisar "generosamente" desviaciones muy pequeñas del desarrollo de la orientación sin desviaciones graves de contorno, con una longitud de matado de esquinas y una tolerancia angular definidas.

Nota

A diferencia del matado de esquinas del contorno (y del desarrollo de la orientación) con G642, en la transición de secuencias con OSD u OST no se genera ninguna secuencia propia, sino que el movimiento de transición de secuencias se agrega directamente a las secuencias originales programadas.

Con OSD u OST no puede hacerse un matado de esquinas en ninguna transición de secuencias en la que tenga lugar un cambio del tipo de interpolación (vector → eje giratorio, eje giratorio → vector) para la orientación de herramienta. En estas transiciones de secuencias pueden matarse las esquinas, en caso necesario, con las funciones habituales de matado G641, G642 o G643.

7.7 Asignación libre de números D, número de filo

7.7.1 Asignación libre de números D, número de filo (dirección CE)

Número D

Los números D se pueden utilizar como números de corrección. Adicionalmente, se puede direccionar a través de la dirección CE el número del filo. Mediante la variable de sistema $TC_DPCE se puede describir el número de filo.

Ajuste previo: Nº corrección == nº de filo

A través de datos de máquina se define la cantidad máxima de números D (números de filo) y la cantidad máxima de filos por herramienta (→ fabricante de la máquina). Los siguientes comandos sólo son útiles si la cantidad máxima de filos (DM18105) se ha definido mayor que la cantidad de filos por herramienta (DM18106). Préstese atención a las indicaciones del fabricante de la máquina herramienta.

Nota

Además de la asignación relativa, los números D también se pueden asignar como números D "planos" o "absolutos" (1-32000) sin relación con un número T (dentro de la función "Estructura de números D plana").

Bibliografía

Manual de funciones básicas; Corrección de herramientas (W1)

7.7.2 Asignación libre de números D: Comprobar números D (CHKDNO)

Función

El comando CKKDNO permite comprobar si los números D existentes se han asignado de manera unívoca. Los números D de todas las herramientas definidas en una unidad TO sólo deben aparecer una vez. No se tienen en cuenta las herramientas de recambio.

Sintaxis

```
state=CHKDNO(Tno1,Tno2,Dno)
```

Descripción

state	=TRUE:	Los números D han sido asignados de forma unívoca para el rango comprobado.
	=FALSE:	Se ha producido una colisión de números D, o la parametrización es inválida. A través de Tno1, Tno2 y Dno se transfieren los parámetros que han producido la colisión. Estos datos se pueden evaluar en el programa de pieza.
CHKDNO(Tno1,Tno2)	Se comprueban todos los números D de las herramientas indicadas.	
CHKDNO(Tno1)	Se comprueban todos los números D de Tno1 frente a todas las demás herramientas.	
CHKDNO	Se comprueban todos los números D de todas las herramientas frente a todas las demás herramientas.	

7.7.3 Asignación libre de números D: Cambiar nombre de números D (GETDNO, SETDNO)

Función

Los números D se tienen que asignar de forma unívoca. Dos filos distintos de una herramienta no pueden tener el mismo número D.

GETDNO

Este comando suministra el número D de un determinado filo (ce) de una herramienta con el número T "t". Si no existe ningún número D para los parámetros introducidos, se ajusta d=0. Si el número D es inválido, se devuelve un valor superior a 32000.

SETDNO

Con este comando se asigna el valor d del número D a un filo CE de la herramienta t.
A través de state se devuelve el resultado de esta instrucción (TRUE o FALSE). Si no existe ningún registro para los parámetros introducidos, se devuelve FALSE. Los errores de sintaxis generan una alarma. El número D no se puede poner explícitamente a 0.

Sintaxis

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

Descripción

`d`	Número D del filo de la herramienta
`t`	Número T de la herramienta
`ce`	Número de filo (número CE) de la herramienta
`state`	Indica si el comando se ha podido ejecutar sin errores (TRUE o FALSE).

Ejemplo: Renombrar un número D

Programación	**Comentarios**
`$TC_DP2[1,2] = 120`	`;`
`$TC_DP3[1,2] = 5.5`	`;`
`$TC_DPCE[1,2] = 3`	`;  Número del filo CE`
`...`	`;`
`N10 def int NºDantiguo, NºDnuevo = 17`	`;`
`N20 NºDantiguo = GETDNO(1,3)`	`;`
`N30 SETDNO(1,3,NºDnuevo)`	`;`

De este modo, se asigna al filo CE=3 el nuevo valor D de 17. Entonces, los datos de este filo se activan a través del número D 17, tanto a través de las variables de sistema como también en la programación con la dirección CN.

7.7.4 Asignación libre de números D: Determinar el número T para el número D predeterminado (GETACTTD)

Función

Con el comando GETACTTD se determina el número T correspondiente a un número D absoluto. No tiene lugar ninguna comprobación del carácter unívoco. Si existen varios números D iguales dentro de una unidad TO, se devolverá el número T de la primera herramienta encontrada. Si se utilizan números D "planos", el uso del comando no tiene sentido, ya que en este caso siempre se devuelve el valor "1" (sin números T en el mantenimiento de datos).

Sintaxis

```
Estado=GETACTTD(Tnr,Dnr)
```

Descripción

Dnr	Número D para el cual se tiene que buscar el número T.	
Tnr	Número T encontrado	
status	Valor:	Significado:
	0	Se ha encontrado el número T. Tnr contiene el valor del número T.
	-1	Para el número D indicado no existe ningún número T; Tnr=0.
	-2	El número D no es absoluto. Tnr obtiene el valor de la primera herramienta encontrada que contiene el número D con el valor Dnr.
	-5	La función no pudo ejecutarse por otro motivo.

7.7.5 Asignación libre de números D: Fijación como no válidos de los números D (DZERO)

Función

El comando DZERO sirve de refuerzo durante el Reset. Los registros de corrección marcados de esta manera ya no son comprobados por el comando CHKDNO. Para que vuelvan a ser accesibles, el número D debe fijarse de nuevo con SETDNO.

Sintaxis

```
DZERO
```

Descripción

DZERO	Marca todos los números D de la unidad TO como no válidos.

7.8 Cinemática del portaherramientas

Requisitos

Un portaherramientas sólo puede orientar una herramienta en todas las direcciones posibles en el espacio si

- existen dos ejes giratorios v_1 y v_2.
- los ejes giratorios están situados en posición perpendicular entre ellos.
- el eje longitudinal de la herramienta se encuentra en posición perpendicular frente al segundo eje giratorio v_2.

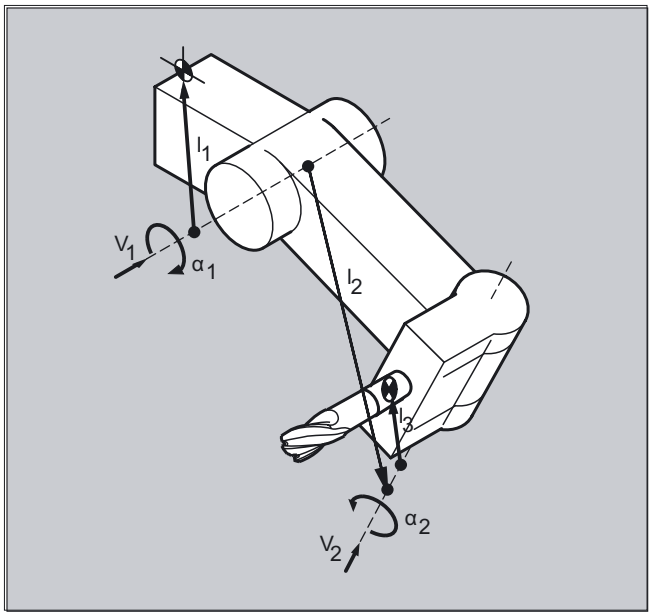
Adicionalmente, se aplica el siguiente requisito para máquinas en las cuales se tienen que poder ajustar todas las orientaciones posibles:

- la orientación de la herramienta tiene que ser perpendicular al primer eje giratorio v_1.

Función

La cinemática de portaherramientas con máx. dos ejes giratorios v_1 o v_2 se describe a través de las 17 variables de sistema `$TC_CARR1[m]` a `$TC_CARR17[m]`. La descripción del portaherramientas se compone de:

- la distancia vectorial del primer eje giratorio al punto de referencia del portaherramientas I_1, la distancia vectorial entre el primer y el segundo eje giratorio I_2, la distancia vectorial entre el segundo eje giratorio y el punto de referencia de la herramienta I_3.
- los vectores de dirección de ambos ejes giratorios v_1, v_2.
- los ángulos de giro α_1, α_2 alrededor de los dos ejes. Los ángulos de giro se cuentan en positivo en sentido horario, con vista en la dirección de los vectores de eje giratorio.



Para máquinas con **cinemática simple** (tanto la herramienta como también la pieza se pueden girar), las variables de sistema se amplían con las entradas

- `$TC_CARR18[m]` a `$TC_CARR23[m]` .

Parámetros

Función de las variables de sistema para portaherramientas orientables			
Denominación	**Componente x**	**Componente y**	**Componente z**
l_1 Vector de corrección	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
l_2 Vector de corrección	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
v_1 Eje giratorio	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
v_2 Eje giratorio	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
α_1 Ángulo de rotación α_2 Ángulo de rotación	$TC_CARR13[m] $TC_CARR14[m]		
l_3 Vector de corrección	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]

Ampliaciones de las variables de sistema para portaherramientas orientables			
Denominación	**Componente x**	**Componente y**	**Componente z**
l_4 Vector de corrección	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
Identificador de eje Eje giratorio v_1 Eje giratorio v_2	Denominador de eje de los ejes giratorios v_1 y v_2 (la ocupación previa es cero) $TC_CARR21[m] $TC_CARR22[m]		
Tipo de cinemática **T**ool **P**art **M**ixed mode	$TC_CARR23[m]		
	Tipo de cinemática T ->	Tipo de cinemática P ->	Tipo de cinemática M
	Sólo la herramienta es giratoria (ocupación previa)	Sólo la pieza es giratoria	La pieza y la herramienta son giratorias
Corrección del eje giratorio v_1 eje giratorio v_2	Ángulo en grados de los ejes giratorios v_1 y v_2 al ocupar la posición preferencial $TC_CARR24[m] $TC_CARR25[m]		
Offset de ángulo del eje giratorio v_1 eje giratorio v_2	Corrección del dentado Hirth en grados de los ejes giratorios v_1 y v_2 $TC_CARR26[m] $TC_CARR27[m]		
Incremento de ángulo v_1 eje giratorio v_2 eje giratorio	Incremento del dentado Hirth en grados de los ejes giratorios v_1 y v_2 $TC_CARR28[m] $TC_CARR29[m]		
Posición Mín. eje giratorio v_1 eje giratorio v_2	Límite de software para la posición mínima de los ejes giratorios v_1 y v_2 $TC_CARR30[m] $TC_CARR31[m]		
Posición Máx. eje giratorio v_1 eje giratorio v_2	Límite de software para la posición máxima de los ejes giratorios v_1 y v_2 $TC_CARR32[m] $TC_CARR33[m]		

Ampliaciones de las variables de sistema para portaherramientas orientables			
Nombre del porta-herramientas	En lugar de un número se puede asignar un nombre al portaherramientas. $TC_CARR34[m]		
Usuario: Nombre de eje 1 Nombre de eje 2 Identificación **Posición**	Uso previsto por el usuario dentro de los ciclos de medida. $TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
Decalaje fino	Parámetros que se pueden sumar **a los valores en los parámetros base**.		
l_1 Vector de corrección	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
l_2 Vector de corrección	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
l_3 Vector de corrección	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
l_4 Vector de corrección	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
v_1 Eje giratorio	$TC_CARR64[m]		
v_2 Eje giratorio	$TC_CARR65[m]		

Nota

Explicaciones sobre los parámetros

Con "m" se indica el número del portaherramientas a describir.

`$TC_CARR47` a `$TC_CARR54`, así como `$TC_CARR61` a `$TC_CARR63` no están definidos y producen una alarma en caso de un intento de acceso de lectura o escritura.

Los puntos iniciales y finales de los vectores de distancia en los ejes se pueden elegir libremente. Los ángulos de giro α_1, α_2 alrededor de los dos ejes se definen en el estado básico del portaherramientas con 0°. De este modo, la cinemática de un portaherramientas se puede describir con un número ilimitado de posibilidades.

Los portaherramientas con un único eje o sin eje giratorio se pueden describir mediante la puesta a cero de los vectores de dirección en uno o ambos ejes giratorios.
En un portaherramientas sin eje giratorio, los vectores de distancia actúan como correcciones de herramienta adicionales en cuyos componentes no se influye al conmutar los planos de mecanizado (`G17` a `G19`).

Ampliaciones de los parámetros

Parámetros de los ejes giratorios

Las variables de sistema se ampliaron con las entradas $TC_CARR24[m] a $TC_CARR33[m] y se describen como sigue:

La **corrección** de los ejes giratorios v_1, v_2	Modificación de la posición del eje giratorio v_1 o v_2 en la posición preferencial del portaherramientas orientable.
Offset de ángulo/incremento de ángulo ejes giratorios v_1, v_2	Offset o incremento de ángulo del dentado Hirth de los ejes giratorios v_1 y v_2. El ángulo programado o calculado se redondea al siguiente valor que se obtiene con un n entero de phi = s + n * d.
Posición mínima y máxima ejes giratorios v_1, v_2	La posición mínima/máxima del eje giratorio, Ángulo límite (límite de software) del eje giratorio v1 y v2.

Parámetros para el usuario

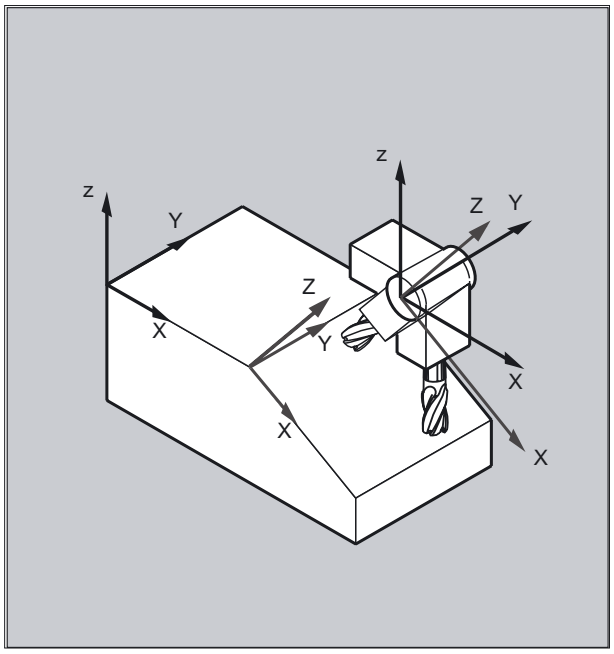
$TC_CARR34 a $TC_CARR40 contienen parámetros disponibles para el usuario que, de manera estándar no se evalúan o no tienen ninguna importancia dentro de NCK hasta el SW 6.4.

Parámetros del decalaje fino

$TC_CARR41 a $TC_CARR65 contienen parámetros de decalaje fino que pueden añadirse a los valores de los parámetros básicos. El valor de decalaje fino asignado a un parámetro básico se obtiene sumando el valor 40 al número de parámetro.

Ejemplo

El portaherramientas utilizado en el siguiente ejemplo se puede describir completamente con un giro alrededor del eje Y.



Código del programa	**Comentarios**
N10 $TC_CARR8[1]=1	; Definición de la componente Y del primer eje giratorio del portaherramientas 1.
N20 $TC_DP1[1,1]=120	; Definición de una fresa normal.
N30 $TC_DP3[1,1]=20	; Definición de una fresa normal con una longitud de 20 mm.
N40 $TC_DP6[1,1]=5	; Definición de una fresa normal con un radio de 5 mm.
N50 ROT Y37	; Definición de frame con giro de 37º alrededor del eje Y.
N60 X0 Y0 Z0 F10000	; Desplazamiento a la posición inicial.
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; Corrección del radio, ajustar corrección longitudinal de herramienta en el frame girado, portaherramientas 1, seleccionar herramienta 1.
N80 X40	; Realizar el mecanizado con un giro de 37º.
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Información adicional

Cinemática simple

Para máquinas con cinemática simple (tanto la herramienta como también la pieza se pueden girar), las variables de sistema se amplían con las entradas `$TC_CARR18[m]` a `$TC_CARR18[m]` y se describen como sigue:

La mesa de herramientas giratoria, compuesta de:

- La distancia vectorial del segundo eje giratorio v_2 al punto de referencia de una mesa de herramientas giratoria I_4 del tercer eje giratorio.

Los ejes giratorios compuestos de:

- Los dos identificadores de canal para la referencia de los ejes giratorios v_1 y v_2, a cuya posición se tendrá acceso, dado el caso, durante la determinación de la orientación del portaherramientas orientable.

El tipo de cinemática con uno de los valores T, P o M:

- Tipo de cinemática T: Sólo la herramienta es giratoria.
- Tipo de cinemática P: Sólo la pieza es giratoria.
- Tipo de cinemática M: La herramienta y la pieza son giratorias.

Borrar los datos del portaherramientas

Con `$TC_CARR1[0]=0` pueden borrarse los datos de todos los juegos de datos de portaherramientas.

El tipo de cinemática `$TC_CARR23[T]=T` debe definirse con una de las tres letras mayúsculas o minúsculas permitidas (T,P,M) y, por este motivo, no debe borrarse.

Modificar los datos del portaherramientas

Cada uno de los valores descritos se puede modificar asignando un nuevo valor en el programa de pieza. Cualquier otro carácter que no sea T, P o M conlleva una alarma si se intenta activar el portaherramientas orientable.

Lectura de los datos de portaherramientas

Cada uno de los valores descritos se puede leer asignándolo a una variable en el programa de pieza.

Decalajes finos

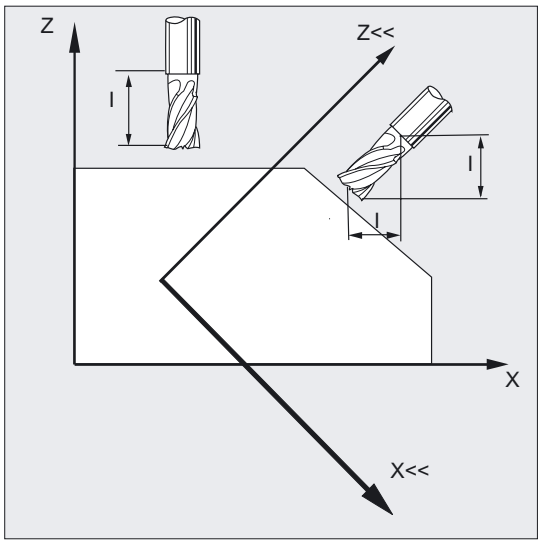
Un valor de decalaje fino no permitido se detecta por primera vez cuando un portaherramientas orientable que contiene uno de estos valores se activa y al mismo tiempo el dato de operador DO42974 $SC_TOCARR_FINE_CORRECTION = TRUE.

La magnitud del decalaje fino admisible queda limitada a través de datos de máquina a un valor máximo admisible.

7.9 Corrección longitudinal de herramienta para portaherramientas orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Función

Cuando cambia la orientación espacial de la herramienta, también se modifican los componentes longitudinales de la misma.



Después de un Reset (p. ej.: tras un ajuste manual o un cambio del portaherramientas con una orientación definida y fija en el espacio) se deberán de recalcular de nuevo los componentes longitudinales de la herramienta. Esto se realiza con los comandos de desplazamiento `TCOABS` y `TCOFR`.

Con un portaherramientas orientable de un frame activo es posible determinar, en la selección de la herramienta, con `TCOFRZ`, `TCOFRY` y `TCOFRX`, la dirección en la que ha de apuntar la herramienta.

Sintaxis

```
TCARR=[<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

Descripción

TCARR=[<m>]:	Solicitar portaherramientas con el número "m"
TCOABS:	Calcular las componentes longitudinales de la herramienta a partir de la orientación actual del portaherramientas.
TCOFR:	Determinar las componentes longitudinales de la herramienta a partir de la orientación del frame activo
TCOFRZ:	Portaherramientas orientable del frame activo cuya herramienta apunta en la dirección Z
TCOFRY:	Portaherramientas orientable del frame activo cuya herramienta apunta en la dirección Y
TCOFRX:	Portaherramientas orientable del frame activo cuya herramienta apunta en la dirección X

Información adicional

Corrección longitudinal de herramienta a partir de la orientación del portaherramientas (TCOABS)

TCOABS calcula la corrección longitudinal de la herramienta a partir de los ángulos de orientación actuales del portaherramientas (que se encuentran en las variables del sistema $TC_CARR13 y $TC_CARR14).

Para la definición de la cinemática del portaherramientas con variables de sistema, ver "Cinemática del portaherramientas (Página 458)".

Para recalcular la corrección longitudinal de herramienta al producirse un cambio de frame, se deberá volver a seleccionar de nuevo la herramienta.

Dirección de la herramienta del frame activo

El portaherramientas orientable se puede ajustar de modo que la herramienta apunte en las siguientes direcciones:

- con TCOFR o TCOFRZ en dirección Z
- con TCOFRY en dirección Y
- con TCOFRX en dirección X

El conmutar entre TCOFR y TCOABS provoca el recálculo de la corrección longitudinal de herramienta.

Solicitar portaherramientas (TCARR)

Con TCARR se solicitan los datos geométricos asociados al número de portaherramientas m (memoria de correcciones).

Con m=0 se deselecciona el portaherramientas activo.

Los datos geométricos del portaherramientas solamente se activan tras la llamada a la herramienta. La herramienta seleccionada permanece activa hasta que se realice un cambio de portaherramientas.

Los datos geométricos del portaherramientas actual se pueden definir desde el programa de pieza mediante las correspondientes variables del sistema.

Nuevo cálculo de la corrección longitudinal de herramienta (TCOABS) al producirse un cambio de frame

Para recalcular la corrección longitudinal de herramienta al producirse un cambio de frame, se deberá volver a seleccionar de nuevo la herramienta.

Nota

La orientación de la herramienta se deberá adaptar al frame activo de forma manual.

Para el cálculo de la corrección longitudinal de la herramienta también se determina en un paso intermedio el ángulo de giro del portaherramientas. Como para portaherramientas con dos ejes giratorios generalmente existen dos pares de ángulos mediante los cuales se puede realizar la adaptación de la orientación de la herramienta al frame activo, los valores angulares de giro memorizados en las variables del sistema deberán de corresponder aproximadamente a los ángulos de giro ajustados mecánicamente.

Nota

Orientación de la herramienta

El control no puede verificar si los ángulos de giro calculados mediante la orientación del frame se pueden ajustar en la máquina o no.

Si los ejes giratorios del portaherramientas tienen una disposición constructiva que impide alcanzar la orientación calculada por el frame, entonces se emite una alarma.

La combinación de la corrección fina de herramienta y las funciones para la corrección longitudinal de herramienta en portaherramientas desplazables, no está permitida. Si se llama a las dos funciones simultáneamente, se emite un mensaje de error.

TOFRAME permite definir un frame en base a la dirección de la orientación del portaherramientas seleccionado. Para más información, ver el capítulo "Frames".

Cuando está activa la transformada de orientación (transformada de 3, 4, 5 ejes), es posible seleccionar un portaherramientas con una orientación que se desvíe de la posición cero sin que se señalice una alarma.

Parámetros de transferencia de ciclos estándar y de medida

Para los parámetros de transferencia de ciclos estándar y de medida rigen unos rangos de valores definidos.

En valores angulares, el rango de valores está definido como sigue:

- Giro alrededor del 1er eje geométrico: -180 grados a +180 grados
- Giro alrededor del 2º eje geométrico: -90 grados a +90 grados
- Giro alrededor del 3er eje geométrico: -180 grados a +180 grados

Ver capítulo Frames, "Rotación programable (ROT, AROT, RPL)".

Nota

En la transferencia de valores angulares a un ciclo estándar o de medida se deberá observar lo siguiente:

¡Los valores menores a la precisión de cálculo del CN se tienen que redondear a cero!

La precisión de cálculo del CN para posiciones angulares está definida en el dato de máquina:

DM10210 $MN_INT_INCR_PER_DEG

7.10 Corrección de longitud de herramienta online (TOFFON, TOFFOF)

Función

A través de la variable de sistema $AA_TOFF[ ], las longitudes de herramienta efectivas según las tres direcciones de herramienta se pueden superponer de forma tridimensional en tiempo real.

Cómo índice se utilizan los tres identificadores de eje geométrico. De este modo, se define el número de direcciones de corrección a través de los ejes geométricos activos al mismo tiempo.

Todas las correcciones pueden estar activas a la vez.

La función Corrección de longitud de herramienta online se puede utilizar en:

- Transformada de orientación TRAORI
- Portaherramientas orientable TCARR

Fabricante de la máquina

La corrección de longitud de herramienta online es una **opción** que se tiene que desbloquear previamente. Esta función sólo tiene sentido en combinación con una transformada de orientación activa o un portaherramientas orientable activo.

Sintaxis

```
N.. TRAORI
N.. TOFFON(X,25)
```

`N.. WHEN TRUE DO $AA_TOFF[dirección de la herramienta]` en acciones síncronas

Para más explicaciones relacionadas con la programación de la corrección longitudinal de herramienta online en acciones síncronas a desplazamientos, ver apartado "Acciones en acciones síncronas".

Descripción

`TOFFON`	**T**ool **Off**set **ON** (activar corrección de longitud de herramienta online) En la activación se puede indicar, para el correspondiente sentido de corrección, un valor de corrección que se aplica inmediatamente.
`TOFFOF`	**T**ool **Off**set **OF** (poner a cero corrección de longitud de herramienta online) Los correspondientes valores de corrección se ponen a cero y se inicia una parada de decodificación previa.
`X, Y, Z,`	Sentido de corrección para el valor de corrección indicado con TOFFON

Ejemplo: Selección de la corrección de longitud de herramienta

Código del programa	Comentarios
DM21190 $MC_TOFF_MODE =1	; Se posiciona en valores absolutos
DM21194 $MC_TOFF_VELO[0] =1000	
DM21196 $MC_TOFF_VELO[1] =1000	
DM21194 $MC_TOFF_VELO[2] =1000	
DM21196 $MC_TOFF_ACCEL[0] =1	
DM21196 $MC_TOFF_ACCEL[1] =1	
DM21196 $MC_TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	; Transformada activada
N20 TOFFON(Z)	; Activación de la corrección longitudinal de herramienta online para la dirección de herramienta Z
N30 WHEN TRUE DO $AA_TOFF[Z] = 10 G4 F5	; Para la dirección de herramienta Z se interpola una corrección longitudinal de herramienta de 10
...	
N100 XOFFSET = $AA_TOFF_VAL[X] N120 TOFFON(X, -XOFFSET) G4 F5	; Asignar corrección actual en dirección X para la dirección de herramienta X, la corrección longitudinal de herramienta vuelve a 0

Ejemplo: Cancelación de la corrección de longitud de herramienta

Código del programa	Comentarios
N10 TRAORI(1)	; Transformada activada
N20 TOFFON(X)	; Activación de la dirección de herramienta Z
N30 WHEN TRUE DO $AA_TOFF[X] = 10 G4 F5	; Para la dirección de herramienta X se interpola una corrección longitudinal de herramienta de 10
...	
N80 TOFFOF(X)	; El decalaje de posición de la dirección de herramienta X se borra: ...$AA_TOFF[X] = 0 No se desplaza ningún eje a la posición actual en el WKS se suma el offset de posición conforme a la orientación actual

Descripción

Preparación de secuencias

En la preparación de secuencias en el preprocesamiento se considera también la corrección de longitud de herramienta actual que se encuentra activa en la marcha principal. Para poder aprovechar al máximo las máximas velocidades de eje admisibles, es necesario detener la preparación de secuencias con una parada de decodificación previa STOPRE mientras se establece una corrección de herramienta.

En el momento del preprocesamiento, la corrección de herramienta también se conoce siempre cuando las correcciones de longitud de herramienta ya no se modifican después del inicio del programa, o si, después de una modificación de las correcciones de longitud de herramienta, se han ejecutado más secuencias de las que puede alojar la memoria IPO entre el preprocesamiento y la marcha principal.

Variable $AA_TOFF_PREP_DIFF

La medida para la diferencia entre la corrección actual, activa en el interpolador, y la corrección activa en el momento de la preparación de la secuencia se puede consultar en la variable `$AA_TOFF_PREP_DIFF[ ]`.

Ajuste de datos de máquina y datos de operador

Para la corrección de longitud de herramienta online se dispone de los siguientes datos de máquina:

- `DM 20610: ADD_MOVE_ACCEL_RESERVE` Reserva de aceleración para movimiento sobrememorizado.
- `DM 21190: TOFF_MODE` El contenido de la variable de sistema `$AA_TOFF[ ]` se ejecuta como valor absoluto o se integra.
- `DM 21194: TOFF_VELO` Velocidad de la corrección de longitud de herramienta online.
- `DM 21196: TOFF_ACCEL` Aceleración de la corrección de longitud de herramienta online.
- Dato de operador para la definición de valores límite
  `DO 42970: TOFF_LIMIT` Límite superior del valor de corrección de longitud de herramienta.

Bibliografía: /FB3/ Manual de funciones especiales; Transformadas de 3 a 5 ejes (F2).

7.11 Modificación de datos de filos en herramientas giratorias (CUTMOD)

Función

Con la función "Modificación de datos de filos en herramientas giratorias", las condiciones geométricas que varían con el giro de herramientas (mayoritariamente herramientas giratorias, pero también brocas y fresas) en relación a la pieza mecanizada pueden tenerse en cuenta para la corrección de herramientas.

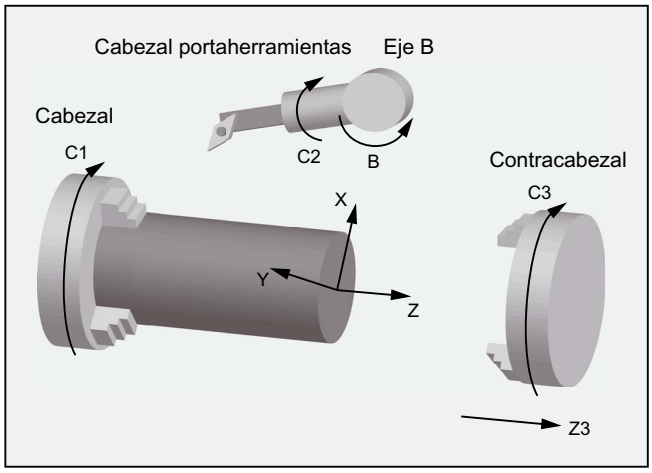


Figura 7-1 Herramienta giratoria en un torno

El giro actual de la herramienta siempre se determina a partir de un portaherramientas orientable actual y activo (ver "Corrección longitudinal de herramienta para portaherramientas orientables (Página 464)").

La función se activa con el comando `CUTMOD`.

Sintaxis

```
CUTMOD=<Valor>
```

Descripción

CUTMOD	Comando para activar la función "Modificación de datos de filos en herramientas giratorias"	
<Valor>	Pueden asignarse los valores siguientes al comando CUTMOD:	
	0	La función está desactivada. Los valores suministrados por las variables del sistema $P_AD... equivalen a los parámetros de herramienta correspondientes.
	> 0	La función se activa si un portaherramientas orientable con el número indicado está activo; es decir, que la activación va ligada a un portaherramientas orientable determinado. Los valores suministrados por las variables del sistema $P_AD... también se han modificado respecto a los parámetros de herramienta correspondientes en función del giro activo. La desactivación del portaherramientas orientable designado desactiva la función temporalmente; la activación de otro portaherramientas orientable la desactiva de forma permanente. En el primer caso, la función se activa de nuevo al seleccionar otra vez el mismo portaherramientas orientable; en el segundo caso, es necesaria una nueva selección, incluso si más adelante el portaherramientas orientable se activa de nuevo con el número indicado. La función no se ve afectada por el Reset.
	-1	La función siempre se activa cuando un portaherramientas orientable está activo. Al cambiar de portaherramientas o al cancelarlo para seleccionarlo de nuevo más adelante, no es necesario volver a fijar CUTMOD.
	-2	La función siempre se activa cuando está activo un portaherramientas orientable cuyo número es igual al del portaherramientas orientable activo actualmente. Si no hay ningún portaherramientas orientable activo, esto equivale a CUTMOD=0. Si hay un portaherramientas orientable activo, éste equivale a la indicación inmediata del número de portaherramientas actual.
	< -2	Los valores inferiores a 2 no se tienen en cuenta, es decir, que este caso se trata como si CUTMOD no estuviera programado. **Nota:** Este rango de valores no debe utilizarse, ya que está reservado para posibles ampliaciones posteriores.

Nota

DO42984 $SC_CUTDIRMOD

La función activable mediante el comando CUTMOD sustituye a la función activable mediante el dato de operador DO42984 $SC_CUTDIRMOD. Sin embargo, esta función sigue estando disponible como antes. De todas maneras, dado que no es conveniente utilizar ambas funciones de forma paralela, sólo puede activarse cuando CUTMOD es igual a cero.

Ejemplo

El siguiente ejemplo hace referencia a una herramienta con la posición de filo 3 y a un portaherramientas orientable que puede girar la herramienta alrededor del eje B.

Los valores numéricos de los comentarios indican las posiciones de fin de secuencia respectivas en coordenadas de máquina (MKS) en el orden X, Y, Z.

```
Código del programa                                          Comentarios
N10 $TC_DP1[1,1]=500
N20 $TC_DP2[1,1]=3                                           ; Posición del filo
N30 $TC_DP3[1,1]=12
N40 $TC_DP4[1,1]=1
N50 $TC_DP6[1,1]=6
N60 $TC_DP10[1,1]=110                                        ; Ángulo de soporte
N70 $TC_DP11[1,1]=3                                          ; Sentido de corte
N80 $TC_DP24[1,1]=25                                         ; Ángulo de despulla

N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0             ; Eje B
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0 $TC_CARR12[2]=1         ; Eje C
N110 $TC_CARR13[2]=0
N120 $TC_CARR14[2]=0
N130 $TC_CARR21[2]=X
N140 $TC_CARR22[2]=X
N150 $TC_CARR23[2]="M"

N160 TCOABS CUTMOD=0
N170 G18 T1 D1 TCARR=2                                         X          Y          Z
N180 X0 Y0 Z0 F10000                                         ; 12.000     0.000      1.000

N190 $TC_CARR13[2]=30
N200 TCARR=2
N210 X0 Y0 Z0                                                ; 10.892     0.000      -5.134
N220 G42 Z–10                                                ; 8.696      0.000      –17.330
N230 Z–20                                                    ; 8.696      0.000      –21.330
N240 X10                                                     ; 12.696     0.000      –21.330
N250 G40 X20 Z0                                              ; 30.892     0.000      –5.134

N260 CUTMOD=2 X0 Y0 Z0                                       ; 8.696      0.000      –7.330
N270 G42 Z–10                                                ; 8.696      0.000      –17.330
N280 Z–20                                                    ; 8.696      0.000      –21.330
N290 X10                                                     ; 12.696     0.000      –21.330
N300 G40 X20 Z0                                              ; 28.696     0.000      –7.330

N310 M30
```

Aclaraciones:

En la secuencia N180 se selecciona en primer lugar la herramienta en CUTMOD=0 y en un portaherramientas orientable sin girar. Dado que todos los vectores offset del portaherramientas orientable son 0, se alcanza la posición que corresponde a la longitud de la herramienta indicada en $TC_DP3[1,1] y $TC_DP4[1,1].

En la secuencia N200 se activa el portaherramientas orientable con un giro de 30º alrededor del eje B. Dado que la posición del filo no se modifica debido a CUTMOD=0, el antiguo punto de referencia del filo sigue siendo la referencia. Por esta razón se alcanza la posición en la secuencia N210, posición que mantiene el antiguo punto de referencia del filo en el origen (es decir, que el vector (1, 12) se gira 30º en el plano Z/X).

En la secuencia N260, a diferencia de la secuencia N200 CUTMOD=2, es efectivo. A causa del giro del portaherramientas orientable, la posición del filo modificada es 8. Con ello también resultan posiciones de eje divergentes.

En las secuencias N220 o N270 se activa respectivamente la corrección de radio de herramienta (WRK). La posición del filo diferente en ambas partes de programa no tiene ninguna influencia sobre las posiciones finales de las secuencias donde la WRK está activa; por lo tanto, las posiciones correspondientes son idénticas. De nuevo en las secuencias de desactivación N260 o N300 vuelven a tener efecto las posiciones del filo diferentes.

Información adicional

Eficacia de los datos de filos modificados

La posición de filo modificada y el punto de referencia del filo modificado serán efectivos de inmediato tras la programación, incluso para una herramienta ya activa. Para ello no se precisa una nueva selección de herramienta.

Influencia del plano de trabajo activo

Para la determinación de una posición de filo, sentido de corte y ángulo de soporte o de despulla modificados, la referencia es el filo en el plano activo correspondiente (G17 - G19).

Sin embargo, si el dato de operador DO42940 $SC_TOOL_LENGTH_CONST (cambio de las componentes de longitud de herramienta al cambiar nivel) contiene un valor válido que no sea cero (más o menos 17, 18 ó 19), su contenido determina el plano donde se tienen en cuenta las dimensiones relevantes.

Variables del sistema

Se ofrecen las siguientes variables del sistema:

Variables del sistema	Descripción
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	Suministra el ángulo (no redondeado) en el plano de mecanizado activo que se ha tomado para la modificación de los datos del filo (posición del filo, sentido de corte, ángulo de despulla y ángulo de soporte) en las funciones activadas con CUTMOD o $SC_CUTDIRMOD. $P_CUTMOD_ANG hace referencia al estado actual en el avance; $AC_CUTMOD_ANG, a la secuencia de ciclo principal actual.
$P_CUTMOD / $AC_CUTMOD	Lee el valor válido actual que se ha programado en último lugar con el comando CUTMOD (número del portaherramientas para el cual debe activarse la modificación de los datos del filo). Si el último valor CUTMOD programado = -2 (activación con el portaherramientas orientable activo actualmente), no se devuelve el valor -2 en $P_CUTMOD, sino el número del portaherramientas orientable activo en el momento de la programación. $P_CUTMOD hace referencia al estado actual en el avance; $AC_CUTMOD, a la secuencia de ciclo principal actual.
$P_CUT_INV / $AC_CUT_INV	Suministra el valor TRUE si la herramienta está girada de manera que deba invertirse el sentido de giro del cabezal. Además, en la secuencia a la que se refiere la operación de lectura en cuestión, deben cumplirse las cuatro condiciones siguientes: 1. Hay una muela o herramienta de tornear activa (tipos de herramienta 400 a 599 y/o DO42950 $SC_TOOL_LENGTH_TYPE = 2). 2. La influencia del filo se ha activado con la instrucción de programación CUTMOD. 3. Hay un portaherramientas orientable activo que se ha designado con el valor numérico de CUTMOD. 4. El portaherramientas orientable gira la herramienta alrededor de un eje en el plano de mecanizado (normalmente el eje C), de manera que los valores normales resultantes del filo de herramienta estén girados más de 90º (normalmente 180º) respecto a la posición de partida. Si al menos una de las cuatro condiciones mencionadas no se cumple, el contenido de la variable es FALSE. Para herramientas cuya posición del filo no esté definida, el valor de la variable es siempre FALSE. $P_CUT_INV hace referencia al estado actual en el avance y $AC_CUT_INV a la secuencia de ciclo principal actual.

Todas las variables de proceso principal ($AC_CUTMOD_ANG, $AC_CUTMOD y $AC_CUT_INV) pueden leerse en acciones síncronas. Un acceso de lectura del avance genera una parada de decodificación previa.

Datos del filo modificados:

Si hay un giro de herramienta activo, los datos modificados se pondrán a disposición en las variables del sistema siguientes:

Variable de sistema	Descripción
$P_AD[2]	Posición del filo
$P_AD[10]	Ángulo de soporte
$P_AD[11]	Sentido de corte
$P_AD[24]	Ángulo de despulla

Nota

Los datos están modificados respecto a los parámetros de herramienta correspondientes ($TC_DP2[..., ...] etc.) siempre que la función "Modificación de datos de filos en herramientas giratorias" se haya activado con el comando `CUTMOD` y esté activo un portaherramientas orientable que realice un giro de herramienta.

Bibliografía

Encontrará más información sobre la función "Modificación de datos de filos en herramientas giratorias" en:

Manual de funciones básicas; Corrección de herramientas (W1)

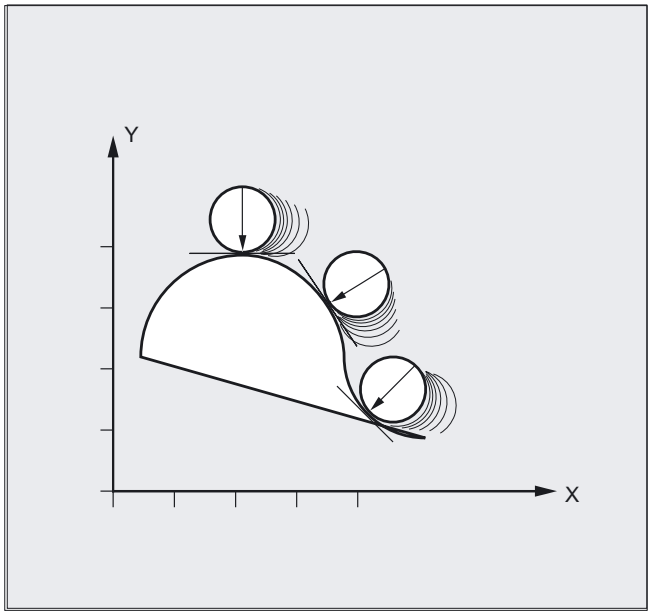
Comportamiento de contorneado

8

8.1 Control tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL)

Función

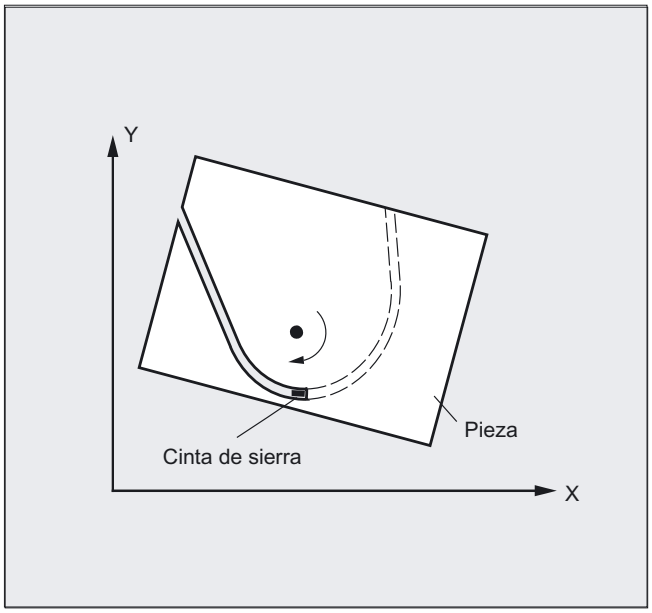
El eje esclavo se arrastra según la tangente por la trayectoria definida por los ejes maestros. De este modo, es posible ajustar una herramienta paralelamente al contorno. Mediante el ángulo programado en la instrucción `TANGON`, la herramienta puede aplicarse con relación a la tangente.



Rango de aplicación

El control tangencial se puede utilizar en aplicaciones tales como:

- Control tangencial de una herramienta giratoria en operaciones de punzonado.
- Seguimiento de la orientación de la pieza en una sierra de cinta (ver figura)
- Posicionado de una herramienta de diamantado en una muela
- Posicionado de un disco de corte en máquinas de transformación de vidrio o papel.
- Alimentación tangencial de la varilla en máquinas de soldadura de 5 ejes.



Sintaxis

```
TANG (Feje,Leje1,Leje2,Acoplamiento,KS,Opt)
TANGON (Feje,Ángulo, Dist, TolAng)
TANGOF (Feje)
TLIFT (Feje)
TANGDEL (Feje)
```

Programaciones simplificadas:

Un factor de acoplamiento de 1 no se necesita programar explícitamente.

`TANG(C, X, Y, 1, "B", "P")` se puede anotar de forma abreviada como `TANG(C, X, Y, , , , "P")`. Como hasta ahora, en lugar de `TANG(C, X, Y, 1, "B", "S")` se puede escribir `TANG(C, X, Y)`.

La instrucción TLIFT(…) se emite después de la asignación de ejes con TANG(…). Ejemplo:

```
TANG(C,X,Y...)
TLIFT(C)
```

Desactivación de TLIFT

Para este fin se repite la asignación de ejes `TANG(...)` sin `TLIFT(...)` posterior.

TANGDEL Borrar definición de un seguimiento tangencial

Un seguimiento tangencial definido por el usuario se tiene que borrar si se quiere definir un nuevo seguimiento tangencial con el mismo eje esclavo en la llamada de preparación `TANG`. El borrado sólo es posible si el acoplamiento está desactivado con `TANGOF(Feje)`.

Descripción

`TANG`	Instrucción preparatoria para la definición de un control tangencial; ajuste previo: 1 **TANG(C,X,Y,1,"B") significa:** El eje giratorio C sigue los ejes geométricos X e Y. Desactivar TLIFT
`TANGON`	Activar control tangencial indicando el eje esclavo y el ángulo de decalaje deseado del eje esclavo; en su caso, recorrido de matado de esquinas, desviación angular. **TANGON(C,90) significa:** El eje C es un eje esclavo. Para cada movimiento de los ejes de contorneado se gira para quedar perpendicular a la tangente a la trayectoria.
`TANGOF`	Desactivación del control tangencial indicando el eje esclavo. Para desactivar el eje esclavo del control tangencial se indica: **TANGOF(C)**
`TLIFT`	Insertar secuencia intermedia en las esquinas del contorno.
`TANGDEL`	Borrar definición de un seguimiento tangencial. **Ejemplo: TANGDEL(Feje)**
`Feje`	Eje esclavo: Eje giratorio adicional con seguimiento tangencial
`Leje1, Leje2`	Ejes maestro: ejes de contorneo a partir de los cuales se determina la tangente para el seguimiento
`Acoplamiento`	Factor de acoplamiento: relación entre la variación angular de la tangente y el eje arrastrado. Indicación opcional; ajuste previo: 1
`KS`	Caracteres para la identificación del sistema de coordenadas "B" = sistema de coordenadas básico; indicación opcional, ajuste previo "W" = sistema de coordenadas de pieza no disponible
`Opt`	Optimización: "S" estándar, valor por defecto "P" ajuste automático del desarrollo en el tiempo del eje tangencial y del contorno
`Ángulo`	Ángulo de decalaje para el eje esclavo
`Dist`	Recorrido de matado de esquinas del eje esclavo, necesario con Opt “P”
`TolAng`	Tolerancia angular del eje esclavo (opcional), evaluación sólo con Opt = “P”

Posibilidad de optimización Opt, Dist y TolAng

Con `Opt="P"` se considera la dinámica del eje esclavo en la limitación de los ejes maestro. Se recomienda sobre todo en caso de uso de transformaciones cinemáticas.

Los parámetros (`Dist` und `TolAng`) limitan de forma controlada el error entre el eje esclavo y la tangente de los ejes maestro.

Ejemplo: Cambio de plano

```
Código del programa          Comentarios
N10 TANG(A, X, Y,1)          ; 1ª definición del seguimiento tangencial.
N20 TANGON(A)                ; Activación del acoplamiento.
N30 X10 Y20                  ; Radio
...
N80 TANGOF(A)                ; Desactivación del 1er acoplamiento.
N90 TANGDEL(A)               ; Borrado de la 1ª definición.
...
TANG(A, X, Z)                ; 2ª definición del seguimiento tangencial.
TANGON(A)                    ; Activación del nuevo acoplamiento
...
N200 M30
```

Ejemplo: Conmutación de eje geométrico y TANGDEL

No se genera ninguna alarma.

```
Código del programa          Comentarios
N10 GEOAX(2,Y1)              ; Y1 es eje geométrico 2.
N20 TANG(A, X, Y)
N30 TANGON(A, 90)
N40 G2 F8000 X0 Y0 I0 J50
N50 TANGOF(A)                ; Desactivación seguimiento con Y1.
N60 TANGDEL(A)               ; Borrado de la 1ª definición.
N70 GEOAX(2, Y2)             ; Y2 es nuevo eje geométrico 2
N80 TANG(A, X, Y)            ; 2ª definición del seguimiento tangencial.
N90 TANGON(A, 90)            ; Activación del seguimiento con la 2ª definición.
...
```

Ejemplo: Seguimiento tangencial con optimización automática

Optimización automática mediante Dist y tolerancia angular.

```
Código del programa                                  Comentarios
N80 G0 C0                                            ; Y1 es eje geométrico 2.
N100 F=50000
N110 G1 X1000 Y500
N120 TRAORI                                          ; Matado de esquinas con
                                                       tolerancia por eje.
N130 G642
N171 TRANS X–Y–                                      ; Optimización automática de la
                                                       velocidad de trayectoria.
N180 TANG(C,X,Y, 1,,"P")                             ; Recorrido de matado de
                                                       esquinas 5 mm
N190 TANGON(C, 0, 5.0, 2.0)                          ; Tolerancia angular 2 grados
N210 G1 X1310 Y500                                   ; Activación del seguimiento con
                                                       la 2ª definición.
N215 G1 X1420 Y500
N220 G3 X1500 Y580 I=AC(1420) J=AC(580)
N230 G1 X1500 Y760
N240 G3 X1360 Y900 I=AC(1360) J=AC(760)
N250 G1 X1000 Y900
N280 TANGOF(C)
N290 TRAFOOF
N300 M02
```

Definir eje esclavo y maestro

TANG se utiliza para definir los ejes maestro y esclavo.

El factor de acoplamiento define la relación entre el cambio de ángulo de la tangente y del eje esclavo. Dicho valor es generalmente 1 (valor estándar).

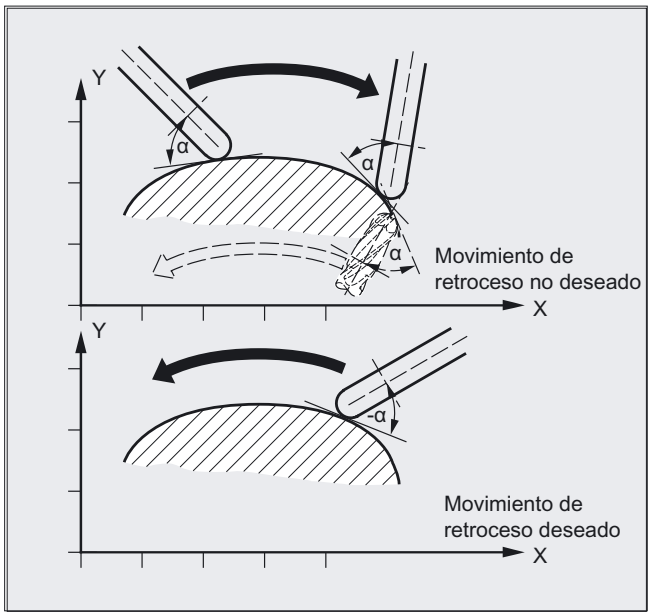
Ángulo límite por limitación del campo de trabajo

En desplazamientos de ida y vuelta a lo largo de la trayectoria, en los puntos de inversión la tangente cambia de pendiente bruscamente en 180°, lo que implica que el eje esclavo realiza también un giro de 180° de forma brusca.

Generalmente, este comportamiento no es conveniente: el movimiento de vuelta se tiene que ejecutar en el mismo ángulo de decalaje negativo que el movimiento de ida.

Para ello se limita la zona de trabajo del eje esclavo (`G25`, `G26`). La limitación de la zona de trabajo debe estar activa (`WALIMON`) en el momento de la inversión de la trayectoria.

Si el ángulo de decalaje para el eje esclavo se encuentra fuera de la limitación de la zona de trabajo, entonces el control numérico intenta volver a la zona de trabajo permitida cambiando el signo del ángulo de decalaje.



Inserción de secuencia intermedia en las esquinas del contorno, TLIFT

La tangente al contorno cambia bruscamente en las esquinas y por lo tanto el valor de consigna del eje en seguimiento también sufre un salto. El eje intenta generalmente corregir dicho salto de consigna con su máxima velocidad. Por ello se produce una diferencia con el ajuste tangencial deseado en el tramo del contorno que sucede a la esquina. Cuando esto no es aceptable por razones tecnológicas, se puede indicar al control numérico, mediante la instrucción `TLIFT`, que se detenga en las esquinas y que genere automáticamente una secuencia intermedia para el eje arrastrado para realizar un giro siguiendo la nueva dirección de la tangente.

El giro se realiza con el eje de contorneado programado si el eje esclavo se ha utilizado una vez como eje de contorneado. Mediante la función `TFGREF[ax] = 0.001` se puede conseguir aquí una velocidad máxima del eje esclavo.

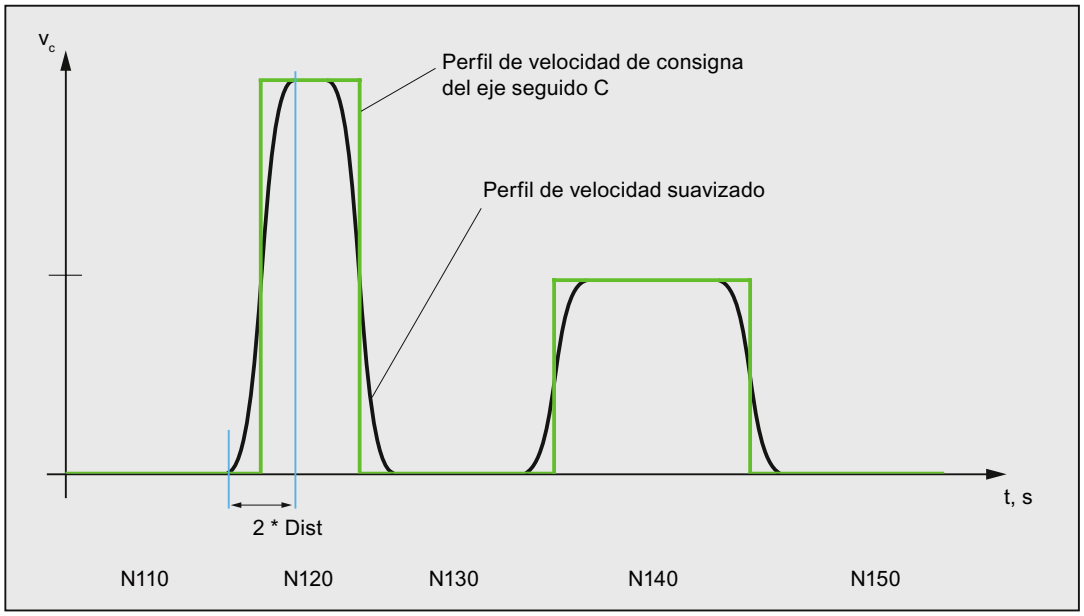
Si, hasta este momento, el eje esclavo no se ha desplazado como eje de contorneado, este eje se desplaza como eje de posicionado. Entonces, la velocidad depende de la velocidad de posicionado consignada en el dato de máquina.

El giro se realiza con la máxima velocidad del eje en seguimiento.

Posibilidad de optimización

Saltos de velocidad del eje esclavo como consecuencia de saltos en el contorno de eje maestro se redondean o alisan con (Dist y TolAng).

El eje esclavo se guía por anticipado (ver diagrama) para mantener la desviación lo más reducida posible.



Definir variación angular

La variación angular a partir de la cual se introduce una secuencia intermedia automática se define a través del dato de máquina `$MA_EPS_TLIFT_TANG_STEP`.

Influencia en transformadas

La posición del eje giratorio en seguimiento puede ser el valor de entrada para una transformada.

Posicionado explícito del eje esclavo

Si se posiciona de forma explícita un eje esclavo en seguimiento (de los maestros), dicha posición se suma al ángulo de decalaje preprogramado.

Se admiten todas las definiciones de recorrido: movimientos de ejes de contorneado y de posicionado.

Estado del acoplamiento

Desde el programa de pieza se puede consultar el estado del acoplamiento por medio de la siguiente variable del sistema:

`$AA_COUP_ACT[Eje]`

0: ningún acoplamiento activo

1,2,3: Seguimiento tangencial activo

8.2 Variación del avance (FNORM, FLIN, FCUB, FPO)

Función

Para poder flexibilizar la introducción de la ley de variación del avance, se ha ampliado la programación del mismo (siguiendo la norma DIN 66025) mediante leyes de variación lineal y cúbica.

La ley de variación cúbica se puede programar directamente, o mediante un spline interpolatorio. De esta forma se pueden conseguir unas velocidades de avance bastante uniformes cuando se tienen que mecanizar contornos muy angulosos.

Estas leyes de variación de velocidad permiten realizar cambios de aceleración evitando sacudidas bruscas, consiguiendo por lo tanto un acabado superficial más uniforme.

Sintaxis

```
F… FNORM
F… FLIN
F… FCUB
F=FPO(…,…,…)
```

Descripción

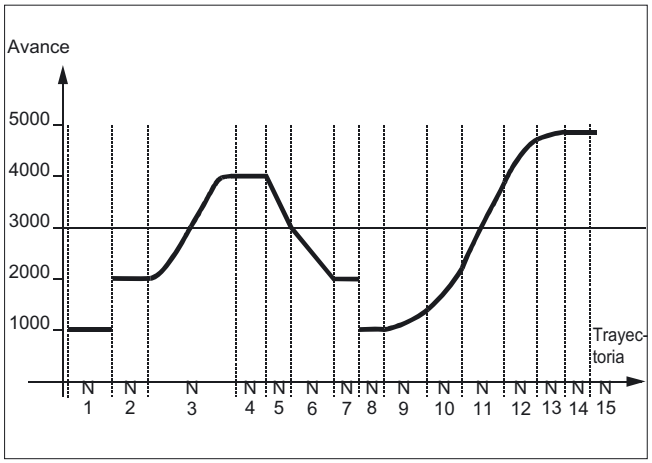
`FNORM`	Ajuste básico. El valor para el avance se define conjuntamente con el desplazamiento, generalmente en la misma secuencia, y después de esto es válido de forma modal.
`FLIN`	Perfil de velocidad de contorneado **lineal:** El valor del avance varía de forma lineal desde la posición inicial de la secuencia hasta la posición final indicada en la secuencia y después de esto es válido de forma modal. Este comportamiento se puede combinar con G93 y G94.
`FCUB`	Perfil de velocidad de contorneado **cúbico:** Los valores F programados secuencia a secuencia se asocian, con referencia al punto final de la secuencia, mediante una interpolación polinómica de tercer orden (spline). El spline se inicia y finaliza tangencialmente al dato de avance anterior y posterior y actúa con G93 y G94. Si se omite la dirección F en una secuencia, entonces se toma el último valor F programado.
`F=FPO…`	Perfil de velocidad de contorneado **mediante un polinomio:** La dirección F describe la variación del avance siguiendo una ley polinómica desde el valor actual de avance hasta el final de la secuencia. A continuación, el valor final se toma como modal.

Optimización del avance en tramos de contornos muy angulosos

El avance polinómico `F=FPO` y el avance spline `FCUB` deben ejecutarse siempre con velocidad de corte constante `CFC`. De esta forma se consigue un perfil de velocidades bastante uniforme y prácticamente sin aceleraciones bruscas.

Ejemplo: Diferentes perfiles de avance

En este ejemplo se puede ver la programación de los diferentes perfiles para el avance así como su representación gráfica.

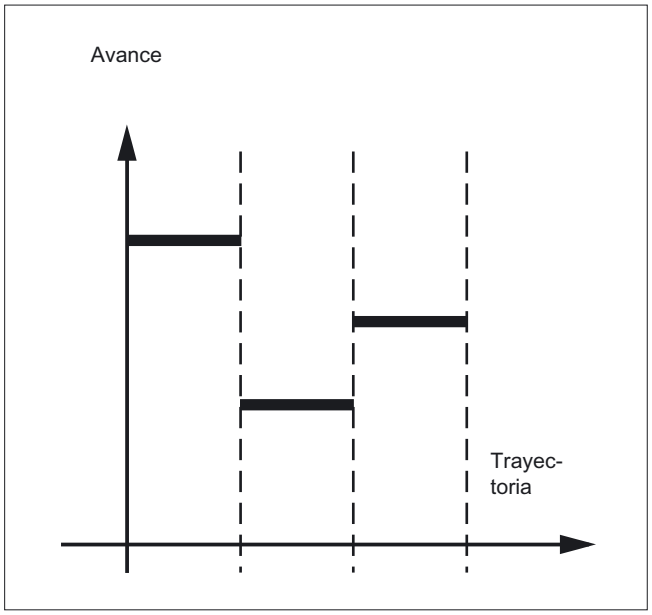


Código del programa	**Comentarios**
N1 F1000 FNORM G1 X8 G91 G64	; Perfil de avance constante, acotado incremental
N2 F2000 X7	; Modificación brusca de la velocidad consigna
N3 F=FPO(4000, 6000, -4000)	; Perfil de avance mediante polinomio con avance 4000 al final de la secuencia
N4 X6	; El avance por polinomio 4000 vale como valor modal
N5 F3000 FLIN X5	; Perfil de avance lineal
N6 F2000 X8	; Perfil de avance lineal
N7 X5	Perfil de avance lineal, vale como valor modal
N8 F1000 FNORM X5	; Perfil de avance constante con modificación brusca de la aceleración
N9 F1400 FCUB X8	; Los siguientes valores F programados por secuencias se unirán con splines
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; Desactivar perfil de spline
N14 FNORM X5	
N15 X20	

FNORM

La dirección F describe el valor para avance constante según la norma DIN 66025.

Más información al respecto se puede encontrar en el manual de programación “Fundamentos”.

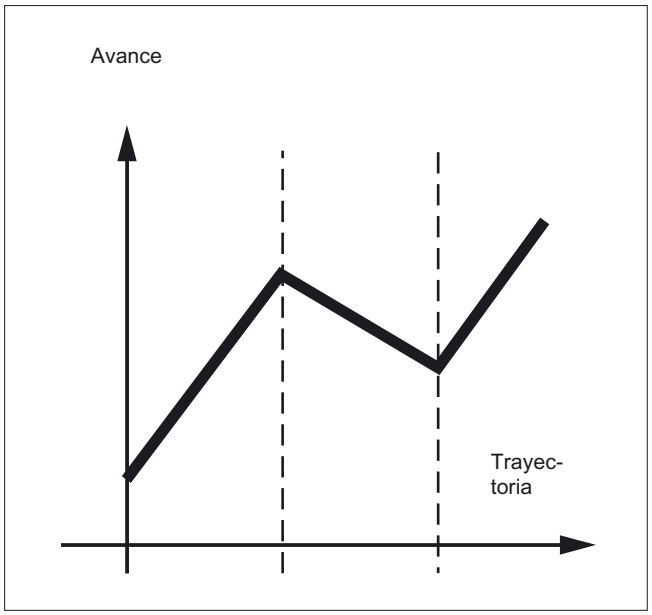


FLIN

El avance varía de forma lineal a lo largo de la trayectoria desde el valor del avance actual hasta el valor F programado.

Ejemplo:

```
N30 F1400 FLIN X50
```

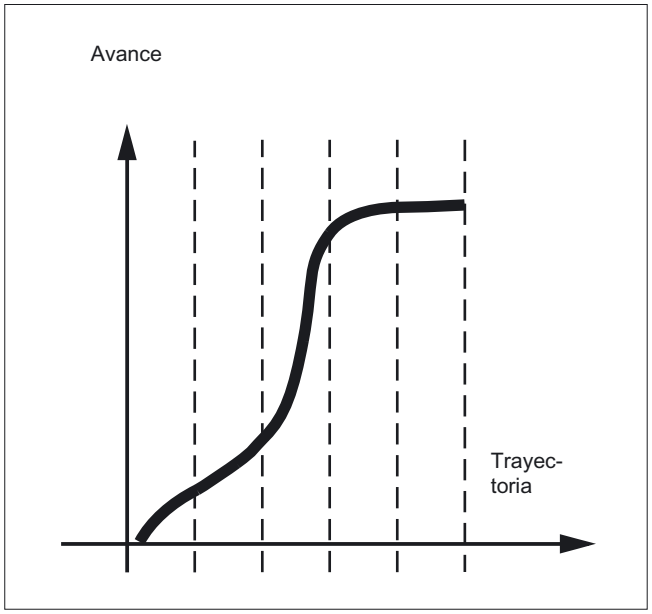


FCUB

El avance varía de forma cúbica a lo largo de la trayectoria programada desde el valor inicial de avance hasta el valor final programado. El control numérico une mediante splines los valores de avance programados en la secuencias activas con FCUB. Los valores de avance se toman como puntos de interpolación para el cálculo de los splines.

Ejemplo:

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```



F=FPO(…,…,…)

La variación del avance se programa directamente mediante un polinomio. La introducción de los coeficientes polinómicos se realiza de forma análoga a la interpolación polinómica.

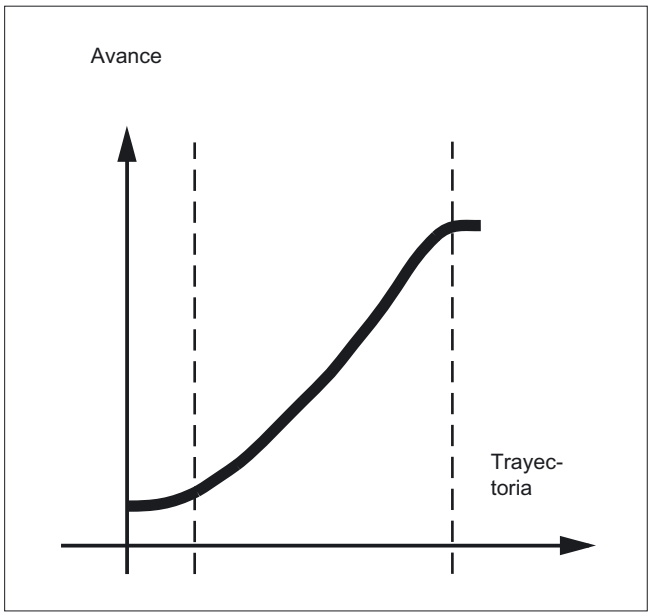
Ejemplo:

```
F=FPO(endfeed, quadf, cubf)
```

endfeed, quadf y cubf son variables con valores previamente definidos.

endfeed:	Avance al final de la secuencia
quadf:	Coeficiente polinómico cuadrado
cubf:	Coeficiente polinómico cúbico

Cuando se tiene la función `FCUB` activada, el spline que une los avances al principio y al final de la secuencia se concatena de forma tangencial con el desarrollo definido con `FPO`.



Limitaciones

Independientemente de la variación del avance programado, son válidas las funciones para la programación de la evolución de la trayectoria.

La variación de la velocidad programada es válida de forma absoluta independientemente de `G90` o `G91`.

La variación de la velocidad programada FLIN y FCUB actúa con

`G93` y `G94`.

`FLIN` y `FCUB` **no actúan con**

`G95`, `G96/G961` y `G97/G971`.

Compresor activo COMPON

Si está activo el compresor `COMPON` se juntan varias secuencias formando un segmento spline:

FNORM:

Para el segmento spline es válida la palabra F de la última secuencia asociada.

FLIN:

Para el segmento spline es válida la palabra F de la última secuencia asociada.
El valor F programado es válido hasta el final del segmento spline y evoluciona de forma lineal.

FCUB:

El spline de avances generado se desvía como máximo del valor final programado en la cantidad indicada en el dato de máquina `C $MC_COMPRESS_VELO_TOL`.

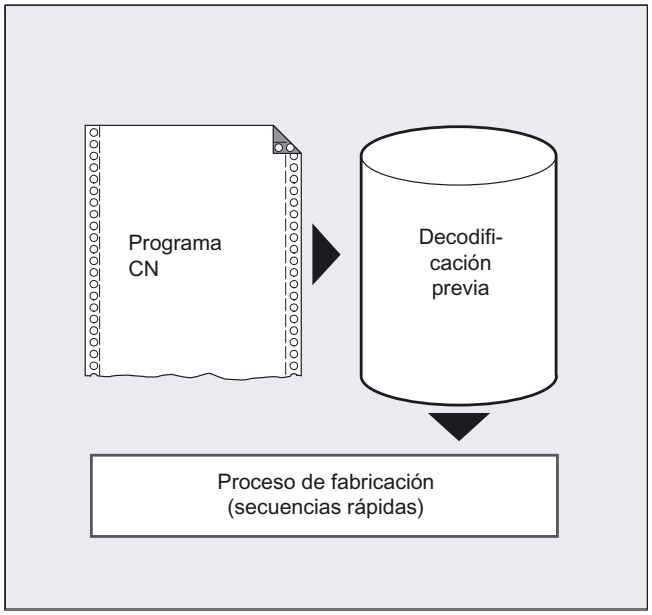
F=FPO(…,…,…)

Estas secuencias no se comprimen.

8.3 Ejecución del programa con memoria de pretratamiento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

Función

Dependiendo de su configuración, el control numérico dispone de una memoria (búfer) de pretratamiento (con decodificación previa) de una cierta capacidad. En ella se memorizan - antes de la ejecución del programa - secuencias ya preparadas. De allí salen como secuencias rápidas para el proceso de fabricación. Esto permite recorrer trayectos cortos a altas velocidades. Siempre que el tiempo residual del control numérico así lo permita, se llena la memoria de pretratamiento.



Identificación del segmento de ejecución

El segmento de ejecución que debe guardarse en la memoria de pretratamiento se identifica al principio con `STOPFIFO` y al final con `STARTFIFO` en el programa de pieza. La ejecución de las secuencias preparadas y guardadas empieza sólo después del comando `STARTFIFO` o cuando la memoria de pretratamiento está llena.

Control automático de la memoria de pretratamiento

El control automático de la memoria de pretratamiento se llama con el comando `FIFOCTRL`. `FIFOCTRL` actúa primero exactamente igual que `STOPFIFO`. En cada programación se espera hasta que la memoria de pretratamiento está llena, y entonces empieza la ejecución. Pero en este caso, el comportamiento es diferente al vaciar la memoria de pretratamiento: con `FIFOCTRL`, a partir de un nivel de llenado de 2/3 la velocidad de contorneado se reduce cada vez más para impedir un vaciado completo y un frenado hasta la parada.

Parada de decodificación previa

El procesamiento y el almacenamiento intermedio de secuencias se detienen si el comando STOPRE está programado en la secuencia. La secuencia subsiguiente sólo se ejecuta cuando se hayan ejecutado completamente todas las secuencias previamente preparadas y memorizadas. La secuencia anterior se detiene en la parada precisa (como G9).

Sintaxis

Tabla 8- 1 Identificar el segmento de ejecución:

```
STOPFIFO
...
STARTFIFO
```

Tabla 8- 2 Control automático de la memoria de pretratamiento:

```
...
FIFOCTRL
...
```

Tabla 8- 3 Parada de decodificación previa:

```
...
STOPRE
...
```

Nota

Los comandos STOPFIFO, STARTFIFO, FIFOCTRL y STOPRE deben programarse en la secuencia propia.

Descripción

STOPFIFO:	STOPFIFO identifica el principio de un segmento de ejecución que debe guardarse en la memoria de pretratamiento. Con STOPFIFO se detiene la ejecución y se llena la memoria de pretratamiento hasta que: • se detectan STARTFIFO o STOPRE o bien, • la memoria de pretratamiento está llena o bien, • se alcanza el final del programa.
STARTFIFO:	Con STARTFIFO empieza la ejecución rápida del segmento de ejecución; paralelamente, se llena la memoria de pretratamiento
FIFOCTRL:	Activación del control automático de la memoria de pretratamiento
STOPRE:	Parar decodificación previa

Nota

El llenado de la memoria de pretratamiento no se ejecuta o se interrumpe si el segmento de ejecución contiene comandos que obligan a un servicio sin respaldo de búfer (búsquedas de punto de referencia, funciones de medida...).

Nota

Cuando se accede a los datos de estado de la máquina ($SA...), el control genera una parada interna de decodificación previa.

⚠ PRECAUCIÓN

Cuando está activa la corrección de herramienta o en caso de interpolaciones spline no se debería programar STOPRE, ya que en este caso se interrumpirían series de secuencias conexas.

Ejemplo: Parar decodificación previa

```
Código de programa                          Comentarios
...
N30 MEAW=1 G1 F1000 X100 Y100 Z50           ; Secuencia de medición con palpador por
                                              la primera entrada e interpolación
                                              lineal.
N40 STOPRE                                  ; Parada de decodificación previa.
...
```

8.4 Secciones de programa con posibilidad de interrupción condicionada (DELAYFSTON, DELAYFSTOF)

Función

Las secciones de programa con posibilidad de interrupción condicionada se denominan áreas Stop-Delay. Dentro de determinadas secciones de programa no se debe **parar** ni modificar el **avance**. Básicamente, las secciones de programa cortas que se utilizan, p. ej., para la ejecución de una rosca, deben protegerse frente a casi todos los sucesos de parada. Una eventual parada sólo actúa una vez que se haya finalizado la ejecución de la sección de programa.

Sintaxis

```
DELAYFSTON
DELAYFSTOF
```

Los comandos se encuentran por sí solos en una línea del programa de pieza.

Ambos comandos se admiten únicamente en programas de pieza, no en acciones síncronas.

Descripción

`DELAYFSTON`	Definir el inicio de una área donde se retrasen las paradas "suaves" hasta alcanzar el fin del área Stop-Delay.
`DELAYFSTOF`	Definir el fin de una área Stop-Delay

Nota

En el dato de máquina DM11550 $MN_STOP_MODE_MASK Bit 0 = 0 (valor por defecto) se define implícitamente una área Stop-Delay si `G331/G332` está activo y se ha programado un desplazamiento de trayectoria o `G4`.

Ejemplo: Sucesos de parada

En el área Stop-Delay no se tiene en cuenta una modificación del **avance y del bloqueo de avance**. Tienen efecto tras el área Stop-Delay.

Los sucesos de parada se dividen en:

Sucesos de parada "suaves"	Reacción: delayed
Sucesos de parada "duros"	Reacción: immediate

Selección de algunos sucesos de parada que producen, al menos, una parada de corta duración:

Nombre del suceso	Reacción	Parámetros de interrupción
RESET	immediate	NST: DB21,… DBX7.7 y DB11, … DBX20.7
PROG_END	Alarma 16954	Prog. CN: M30
INTERRUPT	delayed	NST: FC-9 y ASUP DB10, ... DBB1
SINGLEBLOCKSTOP	delayed	Modo activado secuencia a secuencia en el área Stop-Delay: El CN se detiene al final de la primera secuencia fuera del área Stop-Delay. Modo secuencia a secuencia ya está seleccionado antes del área Stop-Delay: NST: "Parada CN en el límite de secuencia" DB21, ... DBX7.2
STOPPROG	delayed	NST: DB21,… DBX7.3 y DB11, … DBX20.5
PROG_STOP	Alarma 16954	Prog. CN: M0 y M1
WAITM	Alarma 16954	Prog. CN: WAITM
WAITE	Alarma 16954	Prog. CN: WAITE
STOP_ALARM	immediate	Alarma: Configuración de alarma STOPBYALARM
RETREAT_MOVE_THREAD	Alarma 16954	Prog. CN: Alarma 16954 con LFON (Stop & Fastlift no son posibles en G33)
WAITMC	Alarma 16954	Prog. CN: WAITMC
NEWCONF_PREP_STOP	Alarma 16954	Prog. CN: NEWCONF
SYSTEM_SHUTDOWN	immediate	Cierre del sistema con 840Di
ESR	delayed	Parada y retirada ampliada
EXT_ZERO_POINT	delayed	Decalaje de origen externo
STOPRUN	Alarma 16955	BTSS: PI "_N_FINDST" STOPRUN

Explicación de las reacciones

immediate (suceso de parada "duro")	Para inmediatamente, también en el área Stop-Delay
delayed (suceso de parada "suave")	La parada (incluso de corta duración) sólo se produce después del área Stop-Delay.
Alarma 16954	El programa se cancela porque se han utilizado instrucciones de programación no autorizadas en el área Stop-Delay.
Alarma 16955	El programa continúa; en el área Stop-Delay se ha producido una acción no permitida.
Alarma 16957	No se ha podido activar el área de programa (área Stop-Delay) delimitada por DELAYFSTON y DELAYFSTOF. De este modo, cualquier parada en el área actúa inmediatamente y sin retardo.

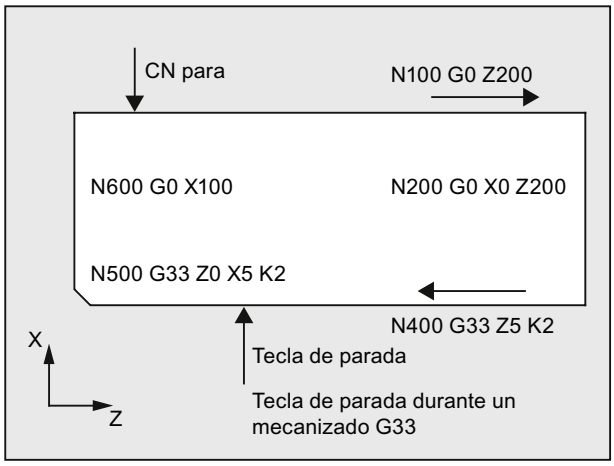
Para un resumen de otras reacciones a sucesos de parada, ver:
Bibliografía:
Manual de funciones básicas; BAG, canal, programación, (K1), apartado "Influencia y efecto sobre sucesos de parada"

Ejemplo: Imbricación de áreas Stop-Delay en dos niveles de programa

```
Código del programa          Comentarios
N10010 DELAYFSTON()          ; Secuencias con N10xxx nivel de programa 1.
N10020 R1 = R1 + 1
N10030 G4 F1                 ; Empieza el área Stop-Delay.
...
N10040 Subprograma 2
...
...                          ; Interpretación del subprograma 2.
N20010 DELAYFSTON()          ; Sin efecto, inicio repetido, 2º nivel.
...
N20020 DELAYFSTOF()          ; Sin efecto, fin en otro nivel.
N20030 RET
N10050 DELAYFSTOF()          ; Fin del área Stop-Delay en el mismo nivel.
...
N10060 R2 = R2 + 2
N10070 G4 F1                 ; Finaliza el área Stop-Delay. A partir de ahora, las
                               paradas tienen efecto inmediato.
```

Ejemplo: Extracto del programa

En un bucle se repite el siguiente bloque de programa:



En la figura se puede ver que el usuario pulsa "Parada" en el área Stop-Delay; el CN inicia el frenado fuera del área Stop-Delay, es decir, en la secuencia `N100`. De este modo, el CN se detiene en la zona delantera de `N100`.

Código del programa	**Comentarios**
`...`	
`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`	

Detalles sobre la búsqueda de secuencias de tipo SERUPRO y avances en combinación con `G331/G332` Avance en roscado con macho sin mandril de compensación ver:
Bibliografía:
Manual de funciones básicas; BAG, canal, programación (K1)
Manual de funciones básicas; Avances (V1)

Ventajas del área Stop-Delay

Una sección de programa se ejecuta sin caída de velocidad.

Si, después de la parada, el usuario cancela el programa con Reset, la secuencia de programa cancelada se sitúa después del área protegida. Entonces, esta secuencia de programa es apta como meta de búsqueda para una búsqueda posterior.

Mientras se ejecuta un área Stop-Delay, no se detienen los siguientes ejes de marcha principal:

- ejes de comando y
- ejes de posicionado que se desplazan con `POSA`.

El comando de programa de pieza `G4` se admite en el área Stop-Delay; en cambio, no se permiten otros comandos de programa de pieza que producen una parada temporal (p. ej., `WAITM`).

Al igual que un movimiento interpolado, `G4` activa un área Stop-Delay o mantiene su vigencia.

Ejemplo: Intervenciones de avance

Si el override se reduce antes del área Stop-Delay al 6%, se activa el override en el área Stop-Delay.

Si el override se reduce dentro del área Stop-Delay del 100% al 6%, el área Stop-Delay se termina con el 100% y se continúa después con el 6%.

El bloqueo de avance no actúa en el área Stop-Delay; la parada sólo se produce al abandonar el área Stop-Delay.

Solapado/imbricación

Si se solapan dos áreas Stop-Delay, una de los comandos en lenguaje de programación y la otra del dato de máquina `DM 11550: STOP_MODE_MASK`, se forma la máxima área Stop-Delay posible.

Los siguientes puntos regulan la coordinación de los comandos en lenguaje de programación `DELAYFSTON` y `DELAYFSTOF` con imbricaciones y el fin del subprograma:

1. Al final del subprograma en el cual se llamó a `DELAYFSTON`, se activa implícitamente `DELAYFSTOF`.
2. `DELAYFSTON` El área Stop-Delay queda sin efecto.
3. Si el subprograma 1 llama en un área Stop-Delay al subprograma 2, el subprograma 2 completo es un área Stop-Delay. En particular, `DELAYFSTOF` queda sin efecto en el subprograma 2.

Nota

REPOSA es un fin de subprograma y `DELAYFSTON` se cancela en todo caso.

Si un suceso de parada "duro" se encuentra con el "área Stop-Delay", se cancela el "área Stop-Delay" completo. Es decir, si se produce en esta sección de programa cualquier otra parada, se detiene inmediatamente. Tan sólo una nueva programación (nuevo `DELAYFSTON`) inicia una nueva área Stop-Delay.

¡Si la tecla Parar se pulsa antes del área Stop-Delay y el NCK tiene que entrar para el frenado en el área Stop-Delay, el NCK se detiene en el área Stop-Delay y éste queda cancelado!

¡Si se entra en un área Stop-Delay con corrección 0%, **no** se acepta el área Stop-Delay!

Esto es válido para todos los sucesos de parada "suaves".

Con `STOPALL` se puede frenar en el área Stop-Delay. Sin embargo, con un `STOPALL`, se activan inmediatamente todos los demás sucesos de parada que habían quedado retardados hasta entonces.

Variables del sistema

Un área Stop-Delay se puede detectar con `$P_DELAYFST` en el programa de pieza. Si el bit 0 de la variable de sistema está ajustado a 1, la ejecución del programa de pieza se encuentra en este momento en un área Stop-Delay.

Un área Stop-Delay se puede detectar con `$AC_DELAYFST` en acciones síncronas. Si el bit 0 de la variable de sistema está ajustado a 1, la ejecución del programa de pieza se encuentra en este momento en un área Stop-Delay.

Compatibilidad

El preajuste del dato de máquina `DM 11550: STOP_MODE_MASK` Bit 0 = 0 produce un área Stop-Delay implícita durante el grupo de códigos G `G331/G332` y si está programado un movimiento interpolado o `G4`.

Bit 0 = 1 posibilita la parada durante el grupo de códigos G `G331/G332` y si está programado un movimiento interpolado o `G4` (comportamiento hasta SW 6). Para la definición de un área Stop-Delay se tienen que utilizar las instrucciones `DELAYFSTON/DELAYFSTOF`.

8.5 Evitar punto de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK)

Función

Para determinadas situaciones mecánicas complicadas en la máquina es necesario evitar la búsqueda de secuencia con SERUPRO.

Con un puntero de interrupción programable existe una posibilidad de intervención, posicionándose en la "búsqueda en el punto de interrupción" delante del punto no apto para la búsqueda.

También se pueden definir zonas no aptas para la búsqueda en áreas del programa de pieza donde el NCK no puede entrar todavía. Con la interrupción de programa, el NCK indica la última secuencia procesada en la que puede realizarse una búsqueda mediante la interfaz de usuario HMI.

Sintaxis

```
IPTRLOCK
IPTRUNLOCK
```

Los comandos están solos en una línea del programa de pieza y permiten un puntero de interrupción programable

Descripción

`IPTRLOCK`	Inicio de la sección de programa no apta para la búsqueda
`IPTRUNLOCK`	Fin de la sección de programa no apta para la búsqueda

Ambos comandos se admiten únicamente en programas de pieza pero no en acciones síncronas.

Ejemplo

Imbricación de secciones de programa no aptas para la búsqueda en dos niveles de programa con IPTRUNLOCK implícito. El IPTRUNLOCK implícito en el subprograma 1 termina el área no apta para la búsqueda.

```
Código del programa              Comentarios
N10010 IPTRLOCK()
N10020 R1 = R1 + 1
N10030 G4 F1                     ; Secuencia de parada, empieza la sección de
                                   programa no apta para la búsqueda.
...
N10040 Subprograma 2
...                              ; Interpretación del subprograma 2.
N20010 IPTRLOCK ()               ; Sin efecto, inicio repetido.
...
N20020 IPTRUNLOCK ()             ; Sin efecto, fin en otro nivel.
N20030 RET
...
N10060 R2 = R2 + 2
N10070 RET                       ; Fin de la sección de programa no apta para la
                                   búsqueda.
N100 G4 F2                       ; Se prosigue con el programa principal.
```

Entonces, una interrupción en 100 vuelve a suministrar el puntero de interrupción.

Registrar y buscar áreas no aptas para la búsqueda

Las secciones de programa no aptas para la búsqueda se identifican con las instrucciones de programación IPTRLOCK e IPTRUNLOCK.

El comando IPTRLOCK congela el puntero de interrupción en una secuencia individual ejecutable en la marcha principal (SBL1). En lo sucesivo, esta secuencia se denomina secuencia de parada. Si se produce después de IPTRLOCK una cancelación del programa, se puede buscar esta llamada secuencia de parada en la interfaz hombre-máquina (HMI).

Reposicionar en la secuencia actual

El puntero de interrupción se posiciona con IPTRUNLOCK, para la sección de programa posterior, en la secuencia actual respecto al punto de interrupción.

Una vez que se haya encontrado una meta de búsqueda, se puede repetir con la misma secuencia de parada una nueva meta de búsqueda.

Un puntero de interrupción editado por el usuario se tiene que volver a quitar vía HMI.

Reglas en imbricaciones

Los siguientes puntos regulan la coordinación de las instrucciones de programación `IPTRLOCK` y `IPTRUNLOCK` con imbricaciones y el fin del subprograma:

1. Al final del subprograma en el cual se llamó a `IPTRLOCK`, se activa implícitamente `IPTRUNLOCK`.
2. `IPTRLOCK` en una zona no apta para la búsqueda queda sin efecto.
3. Si el subprograma 1 llama en una zona no apta para la búsqueda al subprograma 2, el subprograma 2 completo queda no apto para la búsqueda. En particular, `IPTRUNLOCK` queda sin efecto en el subprograma 2.

Para más información al respecto, ver
/FB1/ Manual de funciones básicas; BAG, canal, programación (K1).

Variable de sistema

Un área no apta para la búsqueda se puede detectar con `$P_IPTRLOCK` en el programa de pieza.

Puntero de interrupción automático

La función Puntero de interrupción automático define automáticamente un tipo de acoplamiento previamente determinado como no apto para la búsqueda. Mediante un dato de máquina se activa para

- el reductor electrónico con `EGON`
- el acoplamiento entre ejes de valores maestros con `LEADON`

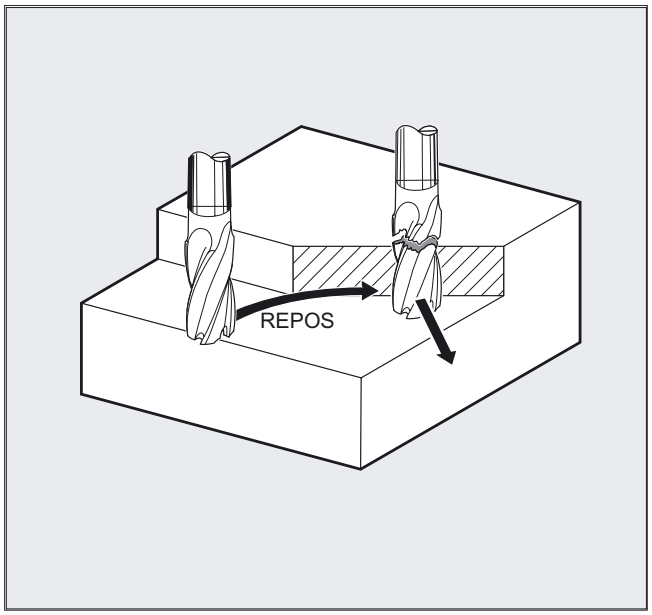
el puntero de interrupción automático. Si se solapan los punteros de interrupción programado y automático (activable a través de un dato de máquina) se forma el área no apta para la búsqueda más amplia posible.

8.6 Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN)

Función

Si el programa en curso se interrumpe durante su ejecución y se desplaza la herramienta - p. ej., debido a una rotura de herramienta o para efectuar una medición - es posible reposicionar en un punto seleccionable del contorno de forma controlada por el programa.

El comando REPOS tiene el mismo efecto que un retorno a subprograma (p. ej. mediante M17). En este caso, no se ejecutan secuencias subsiguientes en la rutina de interrupción.



Para más detalles sobre la interrupción del programa, ver también la sección "Programación CN flexible", apartado "Rutina de interrupción" en este manual de programación.

Sintaxis

```
REPOSA RMI DISPR=…
REPOSA RMB
REPOSA RME
REPOSA RMN
REPOSL RMI DISPR=…
REPOSL RMB
REPOSL RME
REPOSL RMN
REPOSQ RMI DISPR=… DISR=…
REPOSQ RMB DISR=…
REPOSQ RME DISR=…
REPOSQA DISR=…
REPOSH RMI DISPR=… DISR=…
REPOSH RMB DISR=…
REPOSH RME DISR=…
REPOSHA DISR=…
```

Descripción

Trayecto de posicionado

`REPOSA`	Posicionado en una recta con todos los ejes
`REPOSL`	Posicionado en una recta
`REPOSQ DISR=…`	Posicionado en un cuadrante de circunferencia con radio DISR
`REPOSQA DISR=…`	Posicionado en todos los ejes en un cuadrante circunferencia con de radio DISR
`REPOSH DISR=…`	Posicionado en un semicírculo de diámetro DISR
`REPOSHA DISR=…`	Posicionado en todos los ejes en un semicírculo de radio DISR

Punto de reposicionamiento

`RMI`	Posicionado en el punto de interrupción
`RMI DISPR=…`	Punto de entrada a una distancia DISR en mm/pulg antes del punto de interrupción
`RMB`	Posicionado en el inicio de secuencia
`RME`	Posicionado en el fin de secuencia
`RME DISPR=…`	Posicionado en el fin de secuencia a una distancia DISPR del punto final
`RMN`	Posicionado en el punto de trayectoria más cercano
`A0 B0 C0`	Ejes en los cuales hay que efectuar el posicionado

Ejemplo: Posicionado en una recta, REPOSA, REPOSL

La herramienta se aproxima al punto de reposicionamiento siguiendo directamente una línea recta.

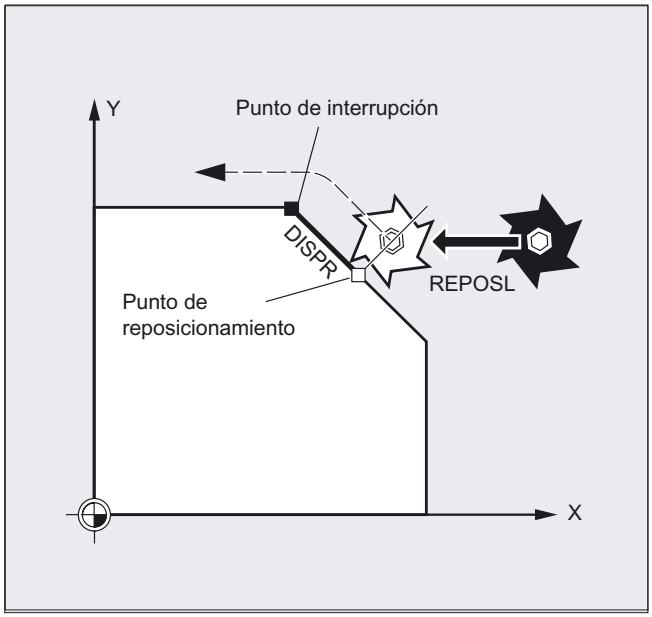
Con REPOSA se desplazan automáticamente todos los ejes. REPOSL permite indicar los ejes a desplazar.

Ejemplo:

```
REPOSL RMI DISPR=6 F400
```

o

```
REPOSA RMI DISPR=6 F400
```

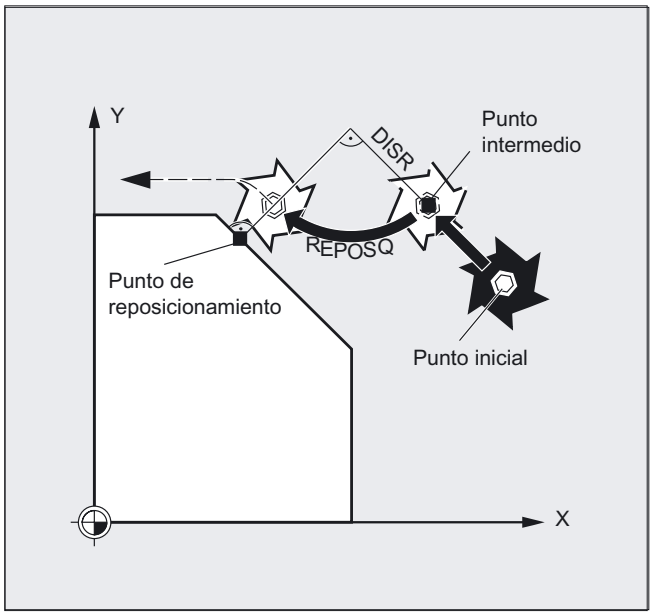


Ejemplo: Aproximación en cuadrante de circunferencia, REPOSQ, REPOSQA

La herramienta se aproxima al punto de reposicionamiento siguiendo un cuadrante de radio `DISR=...`. El control calcula automáticamente el punto intermedio requerido entre los puntos de arranque y reposicionamiento.

Ejemplo:

```
REPOSQ RMI DISR=10 F400
```

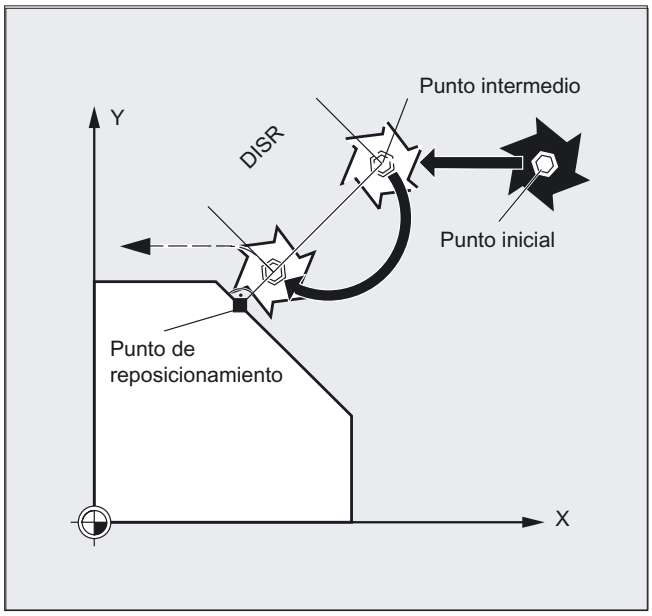


Ejemplo: Aproximación de herramienta en semicircunferencia, REPOSH, REPOSHA

La herramienta se aproxima al punto de reposicionamiento siguiendo un semicírculo de diámetro `DISR=...`. El control calcula automáticamente el punto intermedio requerido entre los puntos de arranque y reposicionamiento.

Ejemplo:

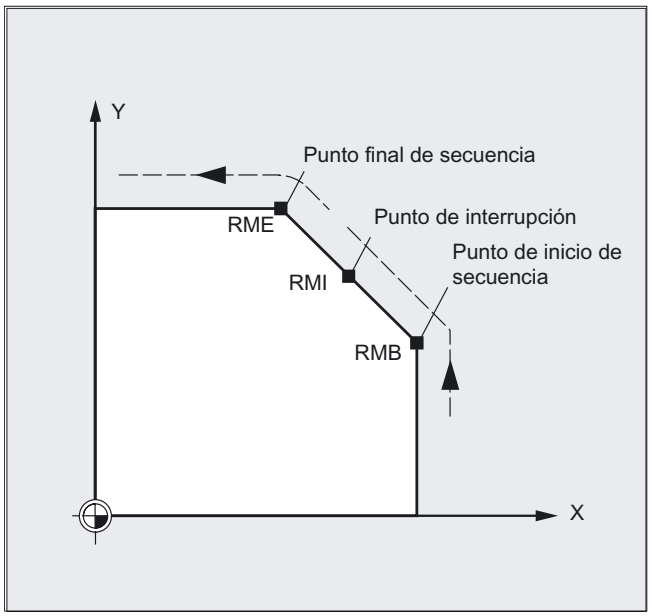
```
REPOSH RMI DISR=20 F400
```



Definir punto de reposicionamiento (no para SERUPRO Aproximación con RMN)

Con respecto a la secuencia de control numérico en la que se interrumpió la ejecución del programa, se puede elegir entre tres puntos de reposicionamiento:

- RMI, punto de interrupción
- RMB, punto de inicio de secuencia o bien último punto final
- RME, punto de fin de secuencia



Con `RMI DISPR=...` o con `RME DISPR=...` se define un punto de reposicionamiento que se encuentra delante del punto de la interrupción o el punto final de la secuencia.

Con `DISPR=...` se define un trayecto de contorno en mm/pulg., en el cual el punto de reposicionamiento queda localizado **antes** del punto de interrupción o del punto final. Como máximo, este punto puede estar localizado - también para valores mayores - en el punto de inicio de secuencia.

Si no se define ningún `DISPR=...`, rige `DISPR=0` y por tanto, el punto de la interrupción (con `RMI`) o el punto final de la secuencia (con `RME`).

Signo DISPR

Se evalúa el signo de `DISPR`. En caso de signo positivo, el comportamiento es como hasta ahora.

En caso de signo negativo se vuelve a posicionar detrás del punto de la interrupción o en `RMB` detrás del punto inicial.

La distancia entre punto de la interrupción-punto de reposicionamiento resulta de la magnitud de `DISPR`. Como máximo, este punto puede estar localizado - también para valores de magnitud mayor - en el punto final de secuencia.

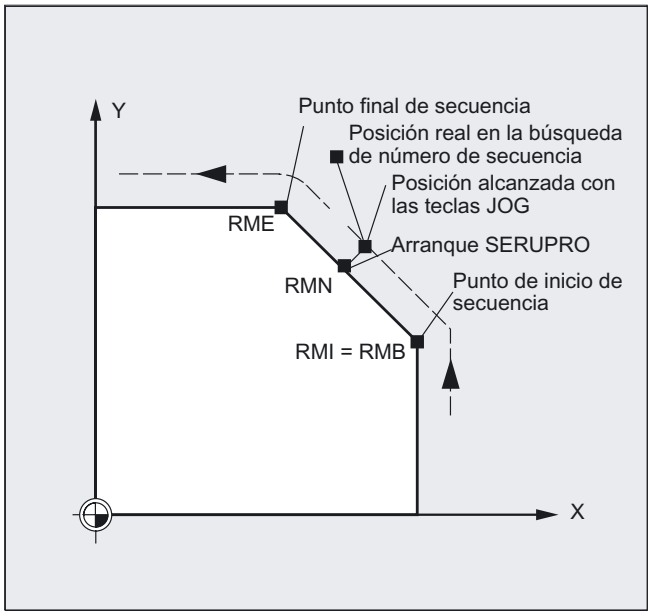
Ejemplo de aplicación:

Con un sensor se detecta la aproximación a una garra de sujeción. Se activa un `ASUP`, con el cual se esquiva la garra de sujeción.

A continuación se reposiciona con el `DISPR` negativo en un punto detrás de la garra de sujeción y se continua el programa.

SERUPRO-Aproximación con RMN

Si, en la ejecución, se fuerza en cualquier punto una cancelación, se posiciona con SERUPRO-Aproximación con `RMN` por la vía más corta en el punto de interrupción para ejecutar después solamente el recorrido restante. Para este fin, el usuario inicia un proceso SERUPRO con respecto a la secuencia de interrupción y posiciona con las teclas JOG delante del punto defectuoso de la secuencia de destino.



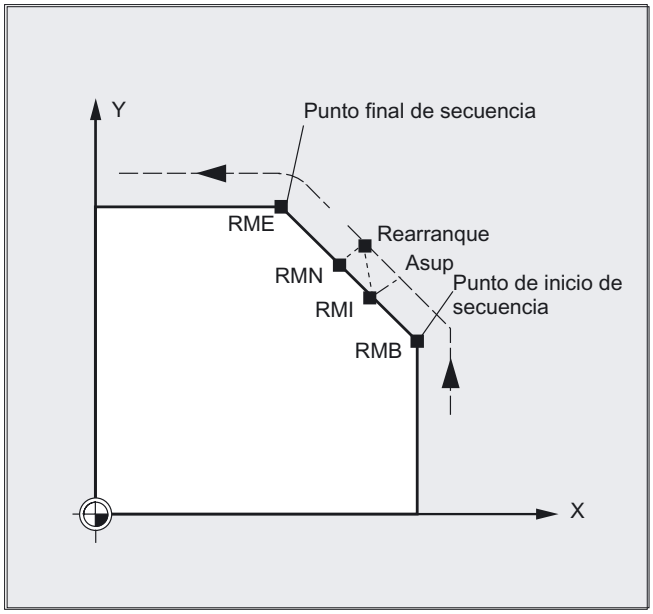
Nota

SERUPRO

Para `SERUPRO`, `RMI` y `RMB` son idénticos. `RMN` no se limita únicamente en `SERUPRO`, sino que tiene validez general.

Posicionado desde el punto de trayectoria más cercano RMN

En el momento de interpretación de `REPOSA`, la secuencia de reposicionamiento después de una interrupción no se vuelve a iniciar por completo con `RMN`, sino que sólo se ejecuta el recorrido restante. Se posiciona en el punto de trayectoria mas cercano de la secuencia interrumpida.



Estado para el modo REPOS válido

El modo REPOS válido de la secuencia interrumpida se puede leer a través de acciones síncronas con la variable `$AC_REPOS_PATH_MODE`:

0: Posicionado no definido

1 `RMB`: Posicionado al principio

2 `RMI`: Posicionado en el punto de interrupción

3 `RME`: Posicionado al final de la secuencia

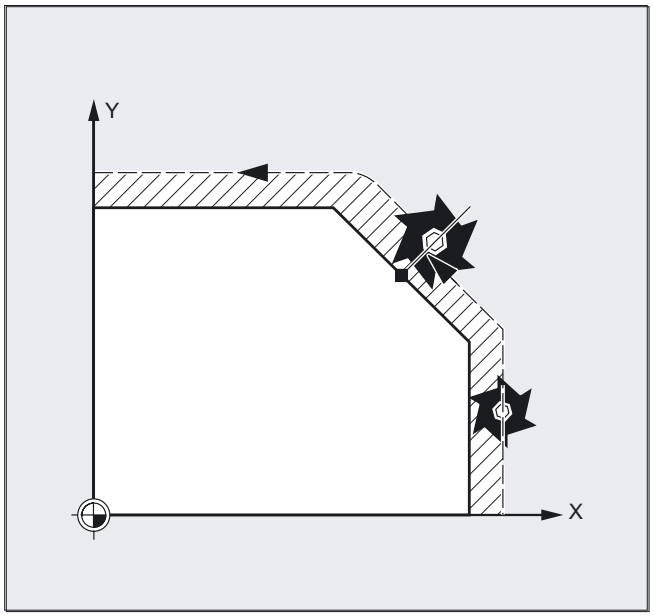
4 `RMN`: Posicionado en el punto de trayectoria más cercano de la secuencia interrumpida.

Aproximación con nueva herramienta

Si la ejecución del programa fue interrumpida por rotura de herramienta:

La programación de un nuevo número D permite continuar el programa a partir del punto de reposicionamiento, utilizando los valores de corrección de herramienta modificados.

Si se modifican los valores de corrección de herramienta es probable que no se pueda reposicionar en el punto de interrupción. En este caso se reposiciona en el punto más cercano a dicho punto en el nuevo contorno (eventualmente modificado con DISPR).



Reposicionamiento en el contorno

El movimiento con el cual la herramienta se aproxima al contorno se puede programar. Las direcciones de los ejes a desplazar se definirán con el valor cero.

Los comandos `REPOSA`, `REPOSQA` y `REPOSHA` permiten reposicionar automáticamente todos los ejes, sin necesidad de indicarlos.

Si se programan `REPOSL`, `REPOSQ` y `REPOSH`, todos los ejes arrancan automáticamente, aunque no se hayan indicado en el comando. Todos los demás ejes se deben definir en el comando.

Para los movimientos circulares REPOSH y REPOSQ se aplica:

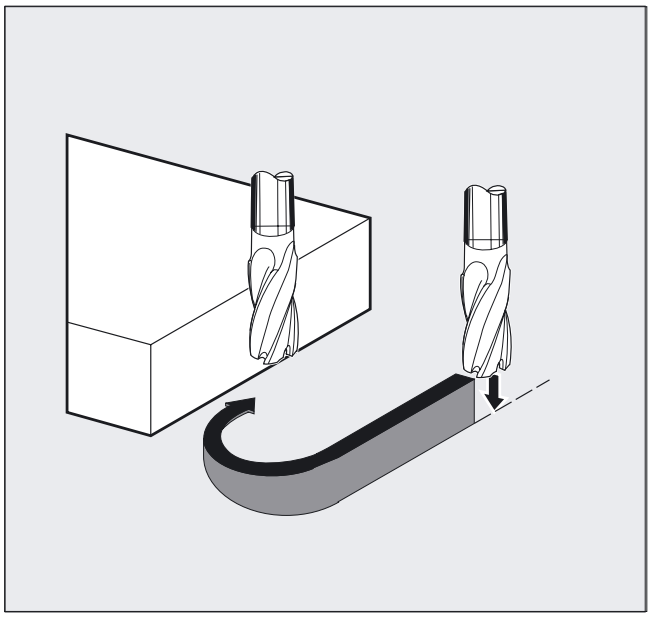
El círculo se recorre en el plano de trabajo indicado `G17` a `G19`.

Si en la secuencia de posicionado se indica el tercer eje geométrico (sentido de penetración), entonces el punto de reposicionamiento se posiciona en una espiral, en caso de que la posición de herramienta no coincida con la posición programada en el sentido de penetración.

En los siguientes casos se conmuta automáticamente a la

aproximación lineal `REPOSL`:

- No se indicó ningún valor para `DISR`.
- No está definido el sentido de aproximación (interrupción de programa en una secuencia sin información de desplazamiento).
- Sentido de aproximación perpendicular al plano actual de trabajo.



8.7 Influencia en el guiado de movimiento

8.7.1 Corrección porcentual de sacudidas (JERKLIM)

Función

En secciones críticas del programa de pieza puede llegar a ser necesario limitar el tirón por debajo del valor máximo permitido para minimizar, p. ej., el esfuerzo de la máquina. El modo de aceleración SOFT tiene que estar activo. La función sólo actúa en ejes de contorneado.

Sintaxis

```
JERKLIM[<Eje>]=<Valor>
```

Descripción

`JERKLIM`	Comando para corrección de tirón
`<Eje>`	Eje de la máquina cuyo valor límite de tirón debe ajustarse.
`<Valor>`	Modificación en porcentaje del tirón máximo permitido respecto al valor ajustado en el dato de máquina para el eje. Rango de valores: 1 ... 200 El valor 100 no tiene ninguna influencia en el tirón. Este ajuste es efectivo tras un Reset y un arranque de programa de pieza.

Ejemplo

En los modos de operación AUTOMÁTICO se restringe el valor límite del tirón para el eje programado al porcentaje indicado del valor límite de tirón consignado en el dato de máquina:

```
N60 JERKLIM[X]=75
```

Significa: El desplazamiento del carro en la dirección X debe acelerarse/retrasarse con sólo un 75% del tirón permitido para el eje.

8.7.2 Corrección porcentual de la velocidad (VELOLIM)

Función

En secciones críticas del programa de pieza puede llegar a ser necesario limitar la velocidad por debajo del valor máximo permitido, p. ej., para minimizar el esfuerzo de la máquina o mejorar la calidad del procesamiento. La función actúa únicamente en ejes de contorneado y de posicionado.

Sintaxis

```
VELOLIM[<Eje>]=<Valor>
```

Descripción

`VELOLIM`	Comando de corrección de la velocidad
`<Eje>`	Eje de la máquina cuyo límite de velocidad debe ajustarse.
`<Valor>`	Modificación en porcentaje de la velocidad máxima permitida respecto al valor ajustado en el dato de máquina para el eje. Rango de valores: 1 ... 100 El valor 100 no tiene ninguna influencia en la velocidad. Este ajuste es efectivo tras un Reset y un arranque de programa de pieza.

Ejemplo

En los modos de operación AUTOMÁTICO se restringe el límite de velocidad para el eje programado al porcentaje indicado del límite de velocidad consignado en el dato de máquina:

```
N70 VELOLIM[X]=80
```

Significa: El desplazamiento del carro en la dirección X debe realizarse con sólo un 80% de la velocidad permitida para el eje.

8.7.3 Ejemplo de programa para JERKLIM y VELOLIM

El siguiente programa representa un ejemplo de aplicación para la limitación porcentual del tirón y de la velocidad:

```
Código del programa                    Comentarios
N1000 G0 X0 Y0 F10000 SOFT G64
N1100 G1 X20 RNDM=5 ACC[X]=20
ACC[Y]=30
N1200 G1 Y20 VELOLIM[X]=5              ; El desplazamiento del carro en la direc-
                                         ción X debe realizarse con máx. 5% de la
                                         velocidad permitida para el eje.
JERKLIM[Y]=200                         ; El desplazamiento del carro en la direc-
                                         ción Y puede acelerarse/retrasarse con
                                         máx. el 200% del tirón permitido para el
                                         eje.
N1300 G1 X0 JERKLIM[X]=2               ; El desplazamiento del carro en la direc-
                                         ción X debe acelerarse/retrasarse con máx.
                                         el 2% del tirón permitido para el eje.
N1400 G1 Y0
M30
```

8.8 Tolerancia de orientación/contorno programable (CTOL, OTOL, ATOL)

Función

Con los comandos `CTOL`, `OTOL` y `ATOL` pueden adaptarse las tolerancias de ejecución definidas con datos de máquina y de operador para las funciones de compresor (COMPON, COMPCURV, COMPCAD), los tipos de matado de esquinas G642, G643, G645, OST y el alisado de la orientación ORISON en el programa CN.

Los valores programados se aplican hasta que se reprograman o se borran mediante la asignación de un valor negativo. Además se borran al final del programa, con un reset de canal, un reset GMO, un reset NCK (arranque en caliente) y tras un Power On (arranque en frío). Después del borrado, vuelven a tener validez los valores de los datos de máquina y de operador.

Sintaxis

```
CTOL=<Valor>
OTOL=<Valor>
ATOL[<Eje>]=<Valor>
```

Descripción

`CTOL` Comando para programar la **tolerancia de contorno**

`CTOL` es válido para:

- todas las funciones de compresor
- todos los tipos de matado de esquinas excepto G641 y G644

`<Valor>`: El valor para la tolerancia de contorno es una indicación de longitud.

Tipo: REAL

Unidad: Pulgadas/mm (en función del ajuste actual del acotado)

`OTOL` Comando para programar la **tolerancia de orientación**

`OTOL` es válido para:

- todas las funciones de compresor
- el alisado de la orientación ORISON
- todos los tipos de matado de esquinas excepto G641, G644, OSD

`<Valor>`: El valor para la tolerancia de la orientación es una especificación de ángulo.

Tipo: REAL

Unidad: grados

`ATOL` Comando para programar una **tolerancia específica de eje**

`ATOL` es válido para:

- todas las funciones de compresor
- el alisado de la orientación ORISON
- todos los tipos de matado de esquinas excepto G641, G644, OSD

`<Eje>`: Nombre del eje para el que debe programarse una tolerancia de eje

`<Valor>`: El valor de la tolerancia de eje es una indicación de longitud o una especificación de ángulo, en función del tipo de eje (lineal o giratorio).

Tipo:	REAL	
Unidad:	para ejes lineales:	Pulgadas/mm (en función del ajuste actual del acotado)
	para ejes giratorios:	Grados

Nota

`CTOL` y `OTOL` tienen prioridad por delante de `ATOL`.

Limitaciones

Frames de escala

Los frames de escala actúan sobre las tolerancias programadas de la misma forma que sobre las posiciones de eje, es decir, la tolerancia relativa se mantiene igual.

Ejemplo

Código de programa	**Comentarios**
COMPCAD G645 G1 F10000	; Activar la función de compresor COMPCAD.
X... Y... Z...	; Aquí actúan los datos de máquina y de operador.
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; A partir de aquí actúa una tolerancia de contorno de 0,02 mm.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
ASCALE X0.25 Y0.25 Z0.25	; A partir de aquí actúa una tolerancia de contorno de 0,005 mm.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=–1	; A partir de aquí vuelven a actuar los datos de máquina y de operador.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Información adicional

Leer valores de tolerancia

Para otros casos de aplicación o para el diagnóstico, las tolerancias válidas actualmente para las funciones de compresor (COMPON, COMPCURV, COMPCAD), los tipos de matado de esquinas G642, G643, G645, OST y el alisado de la orientación ORISON se pueden leer a través de variables de sistema independientemente del tipo de realización.

- En acciones síncronas o con parada de decodificación previa en el programa de pieza, a través de las variables del sistema:

$AC_CTOL	Tolerancia de contorno que estaba activa en el momento de preparar la secuencia del proceso principal actual En caso de que no esté activa ninguna tolerancia de contorno, `$AC_CTOL` suministra la raíz de la suma de los cuadrados de las tolerancias de los ejes geométricos.
$AC_OTOL	Tolerancia de orientación que estaba activa en el momento de preparar la secuencia del proceso principal actual En caso de que no esté activa ninguna tolerancia de orientación, `$AC_OTOL` suministra la raíz de la suma de los cuadrados de las tolerancias de los ejes de orientación durante una transformada de orientación activa; de lo contrario, suministra el valor "-1".
$AA_ATOL[<Eje>]	Tolerancia de eje que estaba activa en el momento de preparar la secuencia del proceso principal actual En caso de que esté activa una tolerancia de contorno, `$AA_ATOL[<Eje geométrico>]` suministra la tolerancia de contorno dividida entre la raíz del número de ejes geométricos. En caso de que estén activas una tolerancia de orientación y una transformada de orientación, `$AA_ATOL[<Eje de orientación>]` suministra la tolerancia de orientación dividida entre la raíz del número de ejes de orientación.

Nota

Si no se han programado valores de tolerancia, las variables $A no estarán lo suficientemente diferenciadas para distinguir las tolerancias que puedan ser diferentes de cada una de las funciones, ya que solamente pueden mencionar un valor.

Estos casos pueden darse cuando los datos de máquina y los de operador ajustan diferentes tolerancias para funciones de compresor, matado de esquinas y alisado de orientación. Las variables suministran entonces el valor mayor que aparece con las funciones que están activas en ese momento.

Si están activas, p. ej., una función de compresor con tolerancia de orientación 0,1° y un alisado de orientación ORISON con 1°, la variable $AC_OTOL suministra el valor "1". Si el alisado de orientación se desactiva, ya sólo se lee el valor "0.1".

- Sin la parada de decodificación previa en el programa de pieza mediante variantes de sistema:

$P_CTOL	Tolerancia de contorno programada
$P_OTOL	Tolerancia de orientación programada
$PA_ATOL	Tolerancia de eje programada

Nota

Si no están programados valores de tolerancia, las variables $P suministran el valor "-1".

Acoplamientos de ejes

9

9.1 Arrastre de ejes (TRAILON, TRAILOF)

Función

Cuando se desplaza un eje definido como maestro, los ejes arrastrados (= ejes esclavos) asignados al conjunto de ejes arrastrados se desplazan siguiendo los recorridos descritos por el eje maestro, teniendo en cuenta el factor de acoplamiento.

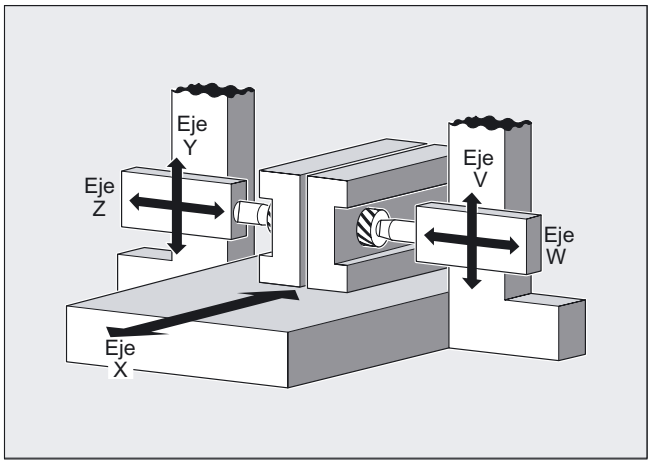
El eje maestro y los esclavos o arrastrados componen un conjunto de ejes maestro-esclavos.

Rango de aplicación

- Desplazamiento de un eje mediante un eje ficticio. El eje maestro es un eje ficticio, el eje arrastrado es un eje real. De esta manera se puede desplazar el eje real teniendo en cuenta el factor de acoplamiento.
- Mecanizado de dos caras con 2 conjuntos de ejes maestro-esclavos:

  1er eje maestro Y, eje arrastrado V

  2º eje maestro Z, eje arrastrado W



Sintaxis

```
TRAILON(<Eje esclavo>,<Eje maestro>,<Factor de acoplamiento>)
TRAILOF(<Eje esclavo>,<Eje maestro>,<Eje maestro 2>)
TRAILOF(<Eje esclavo>)
```

Descripción

`TRAILON`	Comando de activación y definición de un conjunto de ejes Eficacia: modal
`<Eje esclavo>`	Parámetro 1: Nombre del eje arrastrado (esclavo) **Nota:** Un eje arrastrado puede actuar también como eje maestro para otros ejes arrastrados. De esta forma se pueden definir diferentes configuraciones para conjuntos de ejes maestro-esclavos.
`<Eje maestro>`	Parámetro 2: Nombre del eje maestro.
`<Factor de acoplamiento>`	Parámetro 3: Factor de acoplamiento El factor de acoplamiento define la relación entre los trayectos del eje arrastrado y el eje maestro: <Factor de acoplamiento> = trayecto del eje arrastrado/trayecto del eje maestro Tipo: REAL Ajuste previo: 1 La introducción de valores negativos provoca que los desplazamientos entre el eje maestro y arrastrado sean en sentidos opuestos. Si al programar dicha función se omite el factor de acoplamiento, éste toma por defecto el valor 1.

`TRAILOF`	Comando de desactivación de un conjunto de ejes
	Eficacia: Modal
	La función `TRAILOF` con 2 parámetros desactiva sólo el acoplamiento al eje maestro indicado:
	`TRAILOF(<Eje esclavo>,<Eje maestro>)`
	Si un eje arrastrado posee 2 ejes maestros, para desactivar los dos acoplamientos puede ejecutarse la función `TRAILOF` con 3 parámetros:
	`TRAILOF(<Eje esclavo>,<Eje maestro>,<Eje maestro 2>)`
	El mismo resultado se obtiene con la programación de `TRAILOF` sin indicar un eje maestro:
	`TRAILOF(<Eje esclavo>)`

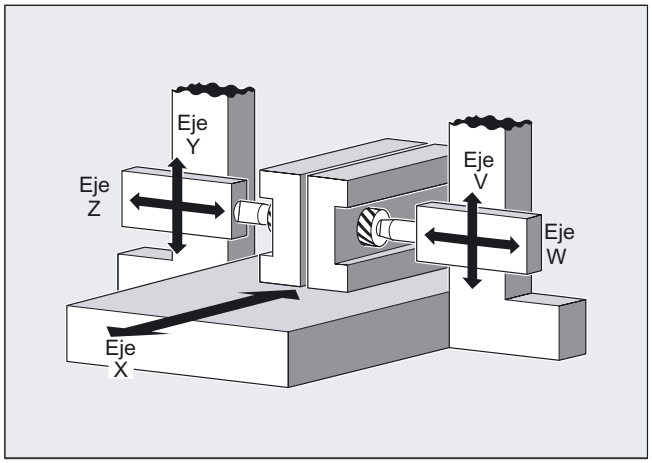
Nota

La función “eje arrastrado” siempre se activa en el sistema de coordenadas básico.

La cantidad máxima de conjuntos de ejes maestro-esclavos activables simultáneamente está limitada únicamente por las posibilidades de combinación de los ejes de la máquina.

Ejemplo

La pieza se debe de mecanizar por las dos caras simultáneamente tal como indica la configuración de la figura. Para esto se definen 2 conjuntos de ejes maestro-esclavos.



Código del programa	**Comentarios**
…	
N100 TRAILON(V,Y)	; Activación del 1er conjunto de ejes maestro-esclavos
N110 TRAILON(W,Z,–1)	; Activación del 2º conjunto de ejes maestro-esclavos Factor de acoplamiento negativo: el eje arrastrado se desplaza en la dirección contraria al eje maestro.
N120 G0 Z10	; Penetración de los ejes Z y W en sentidos opuestos.
N130 G0 Y20	; Penetración de los ejes Y y V en el mismo sentido.
…	
N200 G1 Y22 V25 F200	; Superposición de un desplazamiento del eje arrastrado V dependiente e independiente.
…	
TRAILOF(V,Y)	; Desactivación del 1er conjunto de ejes maestro-esclavos.
TRAILOF(W,Z)	; Desactivación del 2º conjunto de ejes maestro-esclavos.

Información adicional

Tipos de eje

Un conjunto de ejes maestro-esclavos puede estar formado por cualquier combinación de ejes lineales y giratorios. Como eje maestro también se puede definir un eje ficticio.

Ejes arrastrados

Un eje arrastrado puede estar asignado como máximo a dos ejes maestros. La asignación se puede realizar en diferentes conjuntos de ejes maestro-esclavos.

Un eje arrastrado puede programarse con todos los comandos de desplazamiento disponibles (`G0`, `G1`, `G2`, `G3`...). Adicionalmente a las trayectorias definidas de forma independiente, el eje arrastrado también puede efectuar desplazamientos que se derivan de los desplazamientos de su eje maestro por medio de los factores de acoplamiento.

Limitación de la dinámica

La limitación de la dinámica depende del tipo de activación del conjunto de ejes:

- Activación en el programa de pieza

  Si la activación se efectúa en el programa de pieza y todos los ejes maestros actúan como ejes de programa en el canal que se va a activar, durante el desplazamiento de los ejes maestros se tiene en cuenta la dinámica de todos los ejes arrastrados para no sobrecargar ninguno de estos ejes.

  Si la activación se lleva a cabo en el programa de pieza con ejes maestros que no están activos como ejes de programa en el canal que se va a activar ($AA_TYP ≠ 1), durante el desplazamiento de los ejes maestros no se tiene en cuenta la dinámica del eje arrastrado. Esto puede provocar una sobrecarga en los ejes arrastrados con una dinámica menor que la necesaria para el acoplamiento.

- Activación en acción síncrona

  Si la activación tiene lugar en una acción síncrona, durante el desplazamiento de los ejes maestros no se tiene en cuenta la dinámica de los ejes arrastrados. Esto puede provocar una sobrecarga en los ejes arrastrados con una dinámica menor que la necesaria para el acoplamiento.

⚠ PRECAUCIÓN

Si se activa un conjunto de ejes maestro-esclavos

- en acciones síncronas
- dentro del programa de pieza con ejes maestros que no actúan como ejes de programa en el canal del eje arrastrado,

es responsabilidad del usuario/fabricante de la máquina adoptar las medidas oportunas para que el desplazamiento del eje maestro no provoque una sobrecarga de los ejes arrastrados.

Estado de acoplamiento

El estado de acoplamiento de un eje puede consultarse en el programa de pieza mediante estas variables de sistema:

$AA_COUP_ACT[<Eje>]

Valor	Descripción
0	Ningún acoplamiento activo
8	Arrastre de ejes activo

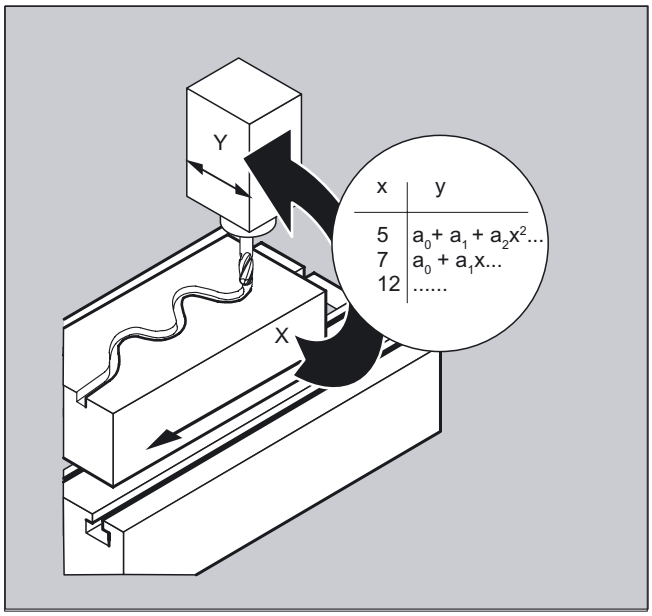
9.2 Tablas de levas (CTAB)

Función

Las tablas de levas sirven para programar relaciones de posición y velocidad entre dos ejes (eje maestro y esclavo). Las tablas de levas se definen en el programa de pieza.

Aplicación

Las tablas de levas sustituyen a los perfiles de levas mecánicos. La tabla de levas constituye la base para el acoplamiento entre ejes de valores maestros, creando la relación funcional entre los valores maestros y de seguimiento: Con la correspondiente programación, el control numérico calcula, a partir de las posiciones de eje maestro y arrastrado asignadas entre sí, un polinomio que permite simular el disco de leva.

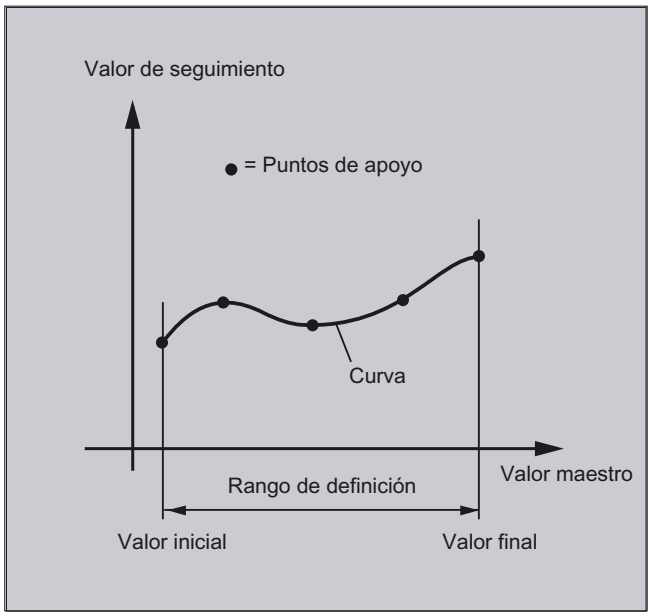


9.2.1 Definición de tablas de levas (CTABDEF, CATBEND)

Función

Una tabla de levas es un programa de pieza o una sección del programa de pieza que comienza con el comando `CTABDEF` y termina con `CTABEND`.

Por medio de comandos de desplazamiento, en esta sección del programa de pieza se asignan a las diferentes posiciones del eje maestro posiciones unívocas del eje esclavo, las cuales sirven como puntos de apoyo para calcular una curva en forma de polinomio de 5º grado como máximo.



Requisito

Para definir tablas de levas debe reservarse espacio en la memoria configurando el DM correspondiente (→ fabricante de la máquina).

Sintaxis

```
CTABDEF(<Eje esclavo>,<Eje
maestro>,<n>,<Periodicidad>[,<Ubicación>])
...
CTABEND
```

Descripción

`CTABDEF( )`	Inicio de la definición de la tabla de levas
`CTABEND`	Fin de la definición de la tabla de levas
`<Eje esclavo>`	Eje cuyo desplazamiento debe calcularse mediante la tabla de levas
`<Eje maestro>`	Eje que proporciona los valores maestros para calcular el desplazamiento del eje esclavo
`<n>`	Número (**ID**) de la tabla de levas El número de una tabla de levas es unívoco e independiente de la ubicación. En la memoria CN estática y dinámica no puede haber tablas con el mismo número.
`<Periodicidad>`	Periodicidad de la tabla `0` La tabla no es periódica (se ejecuta una sola vez, también en el caso de los ejes giratorios) `1` La tabla es periódica en relación con el eje maestro `2` La tabla es periódica en relación con el eje maestro y el eje esclavo
`<Ubicación>`	Indicación de la ubicación (opcional) `"SRAM"` La tabla de levas se crea en la memoria CN **estática**. `"DRAM"` La tabla de levas se crea en la memoria CN **dinámica**. **Nota:** Si no se programa ningún valor para este parámetro, se utiliza la ubicación estándar ajustada con DM20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.

Nota

Sobrescribir

Las tablas de levas se sobrescriben en cuanto se utiliza su número (`<n>`) para definir una nueva tabla (excepción: cuando la tabla de levas está activa en un acoplamiento de ejes o bloqueada con `CTABLOCK`). **¡Al sobrescribir no se emite ningún aviso de advertencia!**

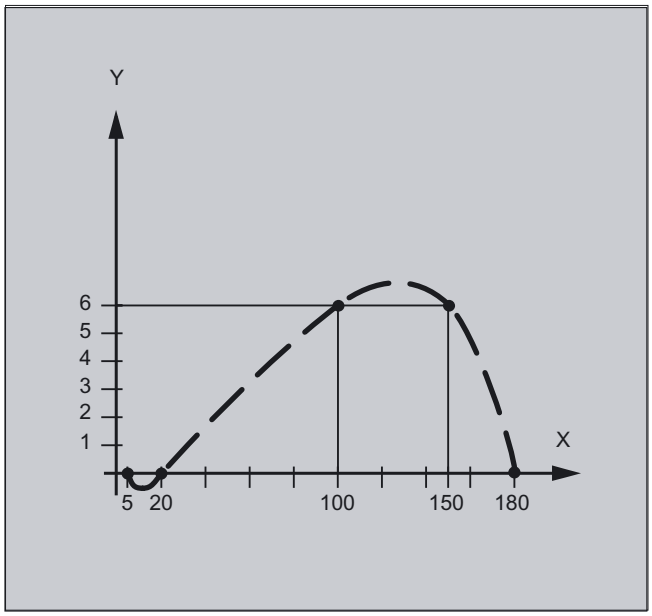
Ejemplos

Ejemplo 1: Sección del programa como definición de la tabla de levas

Se desea utilizar una sección de programa, sin modificación alguna, para definir una nueva tabla de levas. El comando de parada de decodificación previa STOPRE contenido en dicha sección puede conservarse y se activa nuevamente en cuanto la sección del programa deja de utilizarse para la definición de la tabla y se han borrado CTABDEF y CTABEND.

Código del programa	**Comentarios**
…	
CTABDEF(Y,X,1,1)	; Definición de una tabla de levas.
…	
IF NOT ($P_CTABDEF)	
STOPRE	
ENDIF	
…	
CTABEND	

Ejemplo 2: Definición de una tabla de levas no periódica



Código del programa	**Comentarios**
N100 CTABDEF(Y,X,3,0)	; Inicio de la definición de una tabla de levas no periódica con el número 3.
N110 X0 Y0	; El 1er comando de desplazamiento establece valores iniciales y el 1er punto de apoyo: Valor maestro: 0, valor esclavo: 0
N120 X20 Y0	; 2º punto de apoyo: Valor maestro: 0…20, valor esclavo: valor inicial …0
N130 X100 Y6	; 3er punto de apoyo: Valor maestro: 20…100, valor esclavo: 0…6
N140 X150 Y6	; 4º punto de apoyo: Valor maestro: 100…150, valor esclavo: 6…6
N150 X180 Y0	; 5º punto de apoyo: Valor maestro: 150…180, valor esclavo: 6…0
N200 CTABEND	; Fin de la definición. La representación interna de la tabla de levas adopta la forma de un polinomio de 5º grado como máximo. El cálculo de la curva con los puntos de apoyo indicados depende del tipo de interpolación seleccionada modalmente (interpolación circular, lineal, spline). El estado del programa de pieza se restablece antes del inicio de la definición.

Ejemplo 3: Definición de una tabla de levas periódica

Definición de una tabla de levas periódica con el número 2, rango de valores maestros de 0 a 360, desplazamiento de eje arrastrado desde 0 hasta 45 y de regreso a 0:

```
Código del programa                                    Comentarios
N10 DEF REAL DEPPOS
N20 DEF REAL GRADIENT
N30 CTABDEF(Y,X,2,1)                                   ; Inicio de la definición.
N40 G1 X=0 Y=0
N50 POLY
N60 PO[X]=(45.0)
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-90)
N80 PO[X]=(270.0)
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,90)
N100 PO[X]=(360.0)
N110 CTABEND                                           ; Fin de la definición.
;Test de la curva mediante acoplamiento de Y en X:
N120 G1 F1000 X0
N130 LEADON(Y,X,2)
N140 X360
N150 X0
N160 LEADOF(Y,X)

N170 DEPPOS=CTAB(75.0,2,GRADIENT)                      ; Lectura de la función de tabla
                                                         para el valor maestro 75.0.
N180 G0 X75 Y=DEPPOS                                   ; Posicionamiento de eje maestro y
                                                         esclavo.
;Tras la activación del acoplamiento no se requiere ninguna sincronización del eje
esclavo.
N190 LEADON(Y,X,2)
N200 G1 X110 F1000
N210 LEADOF(Y,X)
N220 M30
```

Información adicional

Valor inicial y final de la tabla de levas

Como valor inicial para el comienzo de la zona de definición de la tabla de levas se toma la primera indicación correspondiente a posiciones de ejes coherentes (el primer comando de desplazamiento) dentro de la definición de la tabla. El valor final de la zona de definición de dicha tabla queda determinado correspondientemente por el último comando de desplazamiento.

Repertorio de lenguaje disponible

En la definición de la tabla de levas se pueden utilizar todas las instrucciones del lenguaje de programación CN.

Nota

No se admiten las siguientes indicaciones en las definiciones de tablas de levas:

- Parada de decodificación previa
- Saltos en el desplazamiento del eje maestro (p. ej., en el cambio de transformadas)
- Comando de desplazamiento sólo para el eje arrastrado
- Inversión de desplazamiento del eje maestro; es decir, la posición del eje maestro debe ser siempre unívoca
- Instrucciones `CTABDEF` y `CTABEND` en diferentes niveles del programa

Efecto de las instrucciones modales

Todas las instrucciones modalmente activas programadas dentro de la definición de la tabla de levas pierden su vigencia al concluir la definición de la tabla. Por consiguiente, el estado del programa de pieza en el cual se define la tabla de levas es el mismo antes y después de la definición de la tabla.

Asignaciones a parámetros R

Las asignaciones a parámetros R dentro de la definición de la tabla se resetean tras CTABEND.

Ejemplo:

```
Código del programa                Comentarios
...
R10=5 R11=20                       ; R10=5
...
CTABDEF
G1 X=10 Y=20 F1000
R10=R11+5                          ; R10=25
X=R10
CTABEND
...                                ; R10=5
```

Activación de ASPLINE, BSPLINE, CSPLINE

Si dentro de una definición de tabla de levas CTABDEF ... CTABEND se activa un ASPLINE, BSPLINE o CSPLINE, debería programarse al menos un punto inicial antes de esta activación de spline. Se debería evitar una activación inmediata después de CTABDEF, dado que, de lo contrario, el spline depende de la posición actual del eje antes de la definición de la tabla de levas.

Ejemplo:

```
Código del programa
...
CTABDEF(Y,X,1,0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND
```

Aplicación múltiple de tablas de levas

La relación funcional entre los ejes maestro y esclavo calculada mediante la tabla de levas se conserva bajo el número de tabla seleccionado tras finalizar el programa de pieza y tras POWER OFF, siempre que la tabla esté guardada en la memoria CN estática (SRAM).

Las tablas creadas en la memoria dinámica (DRAM) se borran con POWER ON y posiblemente se tengan que volver a crear.

Las tablas de levas ya creadas se pueden utilizar para cualquier tipo de combinación de ejes maestro y esclavo, sin importar qué ejes hayan sido utilizados para crear dichas tablas.

Sobrescribir tablas de levas

Las tablas de levas se sobrescriben tan pronto como se utiliza su número en una nueva definición de tabla.

Excepción: cuando una tabla de levas está activa en un acoplamiento de ejes o bloqueada con `CTABLOCK`.

Nota

¡Al sobrescribir las tablas de levas no se emite ningún aviso de advertencia!

¿Definición de tabla de levas activa?

La variable de sistema `$P_CTABDEF` permite consultar en cualquier momento desde el programa de pieza si está activa una definición de tabla de levas.

Cancelación de la definición de la tabla de levas

La sección de programa de pieza se puede utilizar nuevamente como programa de pieza real si se excluyen las instrucciones de definición de la tabla de levas.

Carga de tablas de levas mediante "Ejecución de externo"

En la ejecución externa de tablas de levas, el tamaño del búfer de recarga (DRAM) debe seleccionarse a través de DM18360 $MN_MM_EXT_PROG_BUFFER_SIZE de modo que toda la definición de la tabla de levas pueda almacenarse simultáneamente en el búfer de recarga. De lo contrario, la ejecución del programa de pieza se cancela con una alarma.

Saltos del eje esclavo

En función del ajuste en el dato de máquina
DM20900 $MC_CTAB_ENABLE_NO_LEADMOTION,
pueden tolerarse saltos del eje esclavo en caso de falta de movimiento del eje maestro.

9.2.2 Comprobación de la existencia de una tabla de levas (CTABEXISTS)

Función

El comando CTABEXISTS permite comprobar si la memoria CN contiene un número concreto de tabla de levas.

Sintaxis

```
CTABEXISTS(<n>)
```

Descripción

CTABEXISTS	Comprueba si la tabla de levas con el número <n> está presente en la memoria CN estática o dinámica 0 La tabla no existe 1 La tabla existe
<n>	Número (ID) de la tabla de levas

9.2.3 Borrado de tablas de levas (CTABDEL)

Función

Con la función CTABDEL se pueden borrar las tablas de levas.

Nota

Las tablas de levas que están activas en un acoplamiento de ejes no se pueden borrar.

Sintaxis

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<Ubicación>)
CTABDEL ()
CTABDEL(,,<Ubicación>)
```

Descripción

`CTABDEL`	Comando para borrar tablas de levas
`<n>`	Número (**ID**) de la tabla de levas que desea borrarse
	Al borrar un área de tablas de levas `CTABDEL(<n>,<m>)`, `<n>` indica el número de la primera tabla de levas del área.
`<m>`	Al borrar un área de tablas de levas `CTABDEL(<n>,<m>)`, `<m>` indica el número de la última tabla de levas del área.
	¡`<m>` debe ser mayor que `<n>`!
`<Ubicación>`	Indicación de la ubicación (opcional)
	Al efectuar el borrado **sin** especificar la ubicación, las tablas de levas indicadas se borran de la memoria CN estática y dinámica.
	Al efectuar el borrado **con** especificación de la ubicación, de las tablas de levas indicadas se borran sólo las que están presentes en la memoria indicada. Las demás se conservan.
	`"SRAM"` Borrado en la memoria CN **estática**
	`"DRAM"` Borrado en la memoria CN **dinámica**

Si se programa `CTABDEL` sin indicar la tabla de levas que debe borrarse, se borran **todas** las tablas de levas o bien todas las que se encuentran en la memoria indicada:

`CTABDEL ()`	Borra todas las tablas de levas de la memoria CN estática y dinámica
`CTABDEL(,,"SRAM")`	Borra todas las tablas de levas de la memoria CN estática
`CTABDEL(,,"DRAM")`	Borra todas las tablas de levas de la memoria CN dinámica

Nota

Si a la hora de efectuar un borrado múltiple `CTABDEL(<n>,<m>)` o `CTABDEL()` está activa en un acoplamiento al menos una de las tablas de levas que desean borrarse, el comando de borrado no se ejecuta, es decir, no se borra **ninguna** de las tablas de levas especificadas.

9.2.4 Bloqueo de las tablas de levas contra borrado y sobrescritura (CTABLOCK, CTABUNLOCK)

Función

Las tablas de levas pueden protegerse contra el borrado y la sobrescritura accidentales activando bloqueos. Los bloqueos activados pueden anularse en cualquier momento.

Sintaxis

Activar un bloqueo:

```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<Ubicación>)
CTABLOCK()
CTABLOCK(,,<Ubicación>)
```

Anular un bloqueo:

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<Ubicación>)
CTABUNLOCK()
CTABUNLOCK(,,<Ubicación>)
```

Descripción

CTABLOCK	Comando para **activar** un bloqueo contra borrado/sobrescritura
CTABUNLOCK	Comando para **anular** un bloqueo contra borrado/sobrescritura CTABUNLOCK vuelve a habilitar las tablas de levas bloqueadas con CTABLOCK. Las tablas de levas que actúan en un acoplamiento activo permanecen bloqueadas y no pueden borrarse. El bloqueo con CTABLOCK queda anulado en cuanto el bloqueo ejercido por el acoplamiento activo se anula con la desactivación de éste. De este modo se puede borrar esta tabla. No es necesario volver a ejecutar CTABUNLOCK.
<n>	Número (**ID**) de la tabla de levas que desea bloquearse/desbloquearse Al bloquear/desbloquear una área de tablas de levas CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>), <n> indica el número de la primera tabla de levas del área.
<m>	Al bloquear/desbloquear una área de tablas de levas CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>), <m> indica el número de la última tabla de levas del área. ¡<m> debe ser mayor que <n>!

`<Ubicación>`	Indicación de la ubicación (opcional) Al activar/anular un bloqueo **sin** especificar la ubicación, las tablas de levas indicadas se bloquean/desbloquean en la memoria CN estática y dinámica. Al activar/anular un bloqueo **con** especificación de la ubicación, de las tablas de levas indicadas se bloquean/desbloquean sólo las que están presentes en la memoria indicada. Las demás no se bloquean/desbloquean. `"SRAM"` Activar/anular bloqueo en la memoria CN **estática** `"DRAM"` Activar/anular bloqueo en la memoria CN **dinámica**

Si se programa `CTABLOCK`/`CTABUNLOCK` sin indicar la tabla de levas que debe bloquearse/desbloquearse, se bloquean/desbloquean **todas** las tablas de levas o bien todas las que se encuentran en la memoria indicada:

`CTABLOCK()`	Bloquea todas las tablas de levas en la memoria CN estática y dinámica
`CTABLOCK(,,"SRAM")`	Bloquea todas las tablas de levas en la memoria CN estática
`CTABLOCK(,,"DRAM")`	Bloquea todas las tablas de levas en la memoria CN dinámica
`CTABUNLOCK()`	Desbloquea todas las tablas de levas en la memoria CN estática y dinámica
`CTABUNLOCK(,,"SRAM")`	Desbloquea todas las tablas de levas en la memoria CN estática
`CTABUNLOCK(,,"DRAM")`	Desbloquea todas las tablas de levas en la memoria CN dinámica

9.2.5 Tablas de levas: Determinación de las propiedades de la tabla (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Función

Estos comandos permiten consultar propiedades importantes de una tabla de levas (número de tabla, estado de bloqueo, ubicación, periodicidad).

Sintaxis

```
CTABID(<p>)
CTABID(<p>,<Ubicación>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

Descripción

`CTABID`

Proporciona el **número de tabla** consignado en la memoria indicada como tabla de levas número <p>.

Ejemplo:

`CTABID(1,"SRAM")` proporciona el número de la primera tabla de levas en la memoria CN estática. En este caso, la primera tabla de levas se corresponde con la tabla con el número más alto.

Nota:
Si entre dos llamadas consecutivas de `CTABID` se modifica el orden de las tablas de levas en la memoria, p. ej. borrando tablas de levas con `CTABDEL`, una función `CTABID(<p>,...)` con el mismo número <p> podría suministrar una tabla de levas diferente de la anterior.

`CTABISLOCK`

Devuelve el **estado de bloqueo** de la tabla de levas con el número <n>:

0	La tabla no está bloqueada
1	La tabla está bloqueada con `CTABLOCK`
2	La tabla está bloqueada por un acoplamiento activo
3	La tabla está bloqueada con `CTABLOCK` y por un acoplamiento activo
-1	La tabla no existe

`CTABMEMTYP`

Devuelve la **ubicación** de la tabla de levas con el número <n>:

0	Tabla en la memoria CN estática
1	Tabla en la memoria CN dinámica
-1	La tabla no existe

CTABPERIOD	Devuelve la **periodicidad** de la tabla de levas con el número <n>: 0 La tabla no es periódica 1 La tabla es periódica en relación con el eje maestro 2 La tabla es periódica en relación con los ejes maestro y esclavo -1 La tabla no existe
<p>	Número de entrada en la memoria
<n>	Número (**ID**) de la tabla de levas
<Ubicación>	Indicación de la ubicación (opcional) "SRAM" Memoria CN **estática** "DRAM" Memoria CN **dinámica** **Nota:** Si no se programa ningún valor para este parámetro, se utiliza la ubicación estándar ajustada con DM20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.

9.2.6 Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Función

En el programa de pieza pueden leerse los siguientes valores de las tablas de levas:

- Valores de eje esclavo y maestro al principio y al final de una tabla de levas
- Valores de eje esclavo al principio y al final de un segmento de curva
- Valor de eje esclavo en relación con un valor de eje maestro
- Valor de eje maestro en relación con un valor de eje esclavo
- Valor mínimo y máximo del eje esclavo
  - en toda la zona de definición de la tabla de levas

    o bien,
  - en un intervalo definido de la tabla de levas

Sintaxis

```
CTABTSV(<n>,<Gradiente>[,<Eje esclavo>])
CTABTEV(<n>,<Gradiente>[,<Eje esclavo>])
CTABTSP(<n>,<Gradiente>[,<Eje maestro>])
CTABTEP(<n>,<Gradiente>[,<Eje maestro>])
CTABSSV(<Valor maestro>,<n>,<Gradiente>[,<Eje esclavo>])
CTABSEV(<Valor maestro>,<n>,<Gradiente>[,<Eje esclavo>])
CTAB(<Valor maestro>,<n>,<Gradiente>[,<Eje esclavo>,<Eje maestro>]
CTABINV(<Valor esclavo>,<Valor aproximado>,<n>,<Gradiente>[,<Eje
esclavo>,<Eje maestro>]
CTABTMIN(<n>[,<Eje esclavo>])
CTABTMAX(<n>[,<Eje esclavo>])
CTABTMIN(<n>,<a>,<b>[,<Eje esclavo>,<Eje maestro>])
CTABTMAX(<n>,<a>,<b>[,<Eje esclavo>,<Eje maestro>])
```

Descripción

`CTABTSV`	Leer el valor de eje esclavo al **principio** de la tabla de levas nº `<n>`
`CTABTEV`	Leer el valor de eje esclavo al **final** de la tabla de levas nº `<n>`
`CTABTSP`	Leer el valor de eje maestro al **principio** de la tabla de levas nº `<n>`
`CTABTEP`	Leer el valor de eje maestro al **final** de la tabla de levas nº `<n>`
`CTABSSV`	Leer el valor de eje esclavo al **principio** del segmento de curva correspondiente al valor de eje maestro indicado (`<Valor maestro>`)
`CTABSEV`	Leer el valor de eje esclavo al **final** del segmento de curva correspondiente al valor de eje maestro indicado (`<Valor maestro>`)
`CTAB`	Leer el valor de eje esclavo en relación con el valor de eje maestro indicado (`<Valor maestro>`)
`CTABINV`	Leer el valor de eje maestro en relación con el valor de eje esclavo indicado (`<Valor esclavo>`)
`CTABTMIN`	Calcular el **valor mínimo** del eje esclavo: • en toda la zona de definición de la tabla de levas o bien, • en un intervalo definido `<a>` ... `<b>`
`CTABTMAX`	Calcular el **valor máximo** del eje esclavo: • en toda la zona de definición de la tabla de levas o bien, • en un intervalo definido `<a>` ... `<b>`
`<n>`	Número (**ID**) de la tabla de levas
`<Gradiente>`	El parámetro `<Gradiente>` devuelve el **paso** de la función de la tabla de levas respecto a la posición calculada
`<Eje esclavo>`	Eje cuyo desplazamiento debe calcularse mediante la tabla de levas (opcional)

<Eje maestro>	Eje que proporciona los valores maestros para calcular el desplazamiento del eje esclavo (opcional)
<Valor esclavo>	Valor de eje maestro para leer el valor de eje maestro correspondiente con CTABINV
<Valor maestro>	Valor de eje maestro: • para leer el valor de eje esclavo correspondiente con CTAB o bien, • para seleccionar el segmento de curva con CTABSSV/CTABSEV
<Valor aproximado>	No es necesario que la asignación de un valor de eje maestro a un valor de eje esclavo con CTABINV sea siempre unívoca. Por tanto, CTABINV requiere como parámetro un valor aproximado para el valor de eje maestro esperado.
<a>	Límite **inferior** del intervalo de valores maestros con CTABTMIN/CTABTMAX
<b>	Límite **superior** del intervalo de valores maestros con CTABTMIN/CTABTMAX **Nota:** El intervalo de valores maestros <a> ... <b> debe estar dentro de la zona de definición de la tabla de levas.

Ejemplos

Ejemplo 1:

Cálculo de los valores de eje esclavo y maestro al principio y al final de la tabla de levas, así como del valor mínimo y máximo del eje esclavo en toda la zona de definición de la tabla de levas.

```
Código de programa                          Comentarios
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL STARTPARA
N40 DEF REAL ENDPARA
N50 DEF REAL MINVAL
N60 DEF REAL MAXVAL
N70 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                       ; Inicio de la definición de tabla
N110 X0 Y10                                 ; Posición inicial del 1er segmento de tabla
N120 X30 Y40                                ; Posición final del 1er segmento de la tabla = posición
                                              inicial del 2º segmento de la tabla
N130 X60 Y5                                 ; Posición final del 2º segmento de la tabla = ...
N140 X70 Y30
N150 X80 Y20
N160 CTABEND                                ; Fin de la definición de la tabla.
...
N200 STARTPOS=CTABTSV(1,GRADIENT)           ; Valor de eje esclavo al principio de la tabla de levas = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)             ; Valor de eje esclavo al final de la tabla de levas = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)          ; Valor de eje maestro al principio de la tabla de levas = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)            ; Valor de eje maestro al final de la tabla de levas = 80
N240 MINVAL=CTABTMIN(1)                     ; Valor mínimo del eje esclavo con Y = 5
N250 MAXVAL=CTABTMAX(1)                     ; Valor máximo del eje esclavo con Y = 40
```

Ejemplo 2:

Cálculo de los valores de eje esclavo al principio y al final del segmento de curva correspondiente al valor de eje maestro X = 30.

```
Código de programa                          Comentarios
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                       ; Inicio de la definición de tabla.
N110 X0 Y0                                  ; Posición inicial del 1er segmento de tabla
N120 X20 Y10                                ; Posición final del 1er segmento de la tabla =
                                              posición inicial del 2º segmento de la tabla
N130 X40 Y40                                  Posición final del 2º segmento de la tabla = ...
N140 X60 Y10
N150 X80 Y0
N160 CTABEND                                ; Fin de la definición de la tabla.
...
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)      ; Posición inicial Y en el 2º segmento = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)        ; Posición final Y en el 2º segmento = 40
```

Información adicional

Uso en acciones síncronas

Todos los comandos para leer valores de las tablas de levas pueden utilizarse también en acciones síncronas (ver el apartado "Acciones síncronas a desplazamientos").

A la hora de emplear los comandos `CTABINV`, `CTABTMIN` y `CTABTMAX` es preciso asegurarse de que:

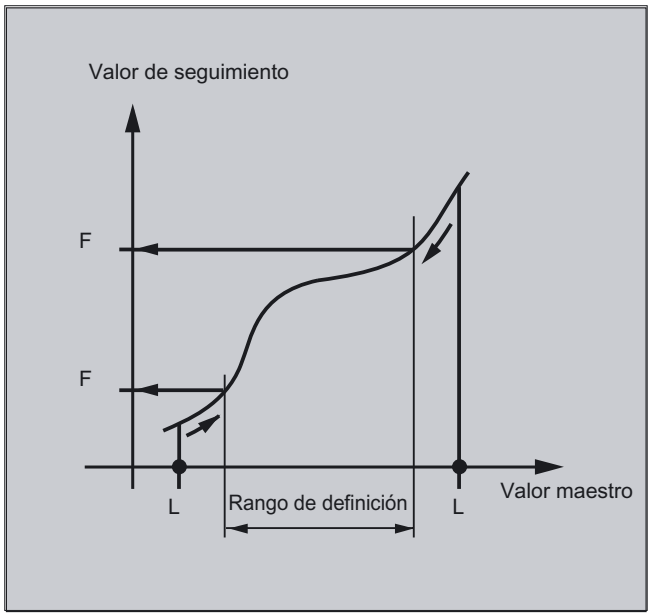
- el rendimiento de CN sea suficiente en el momento de ejecución

  o bien,

- se consulte el número de segmentos de la tabla de levas antes de la llamada para poder subdividir la tabla en cuestión si es necesario

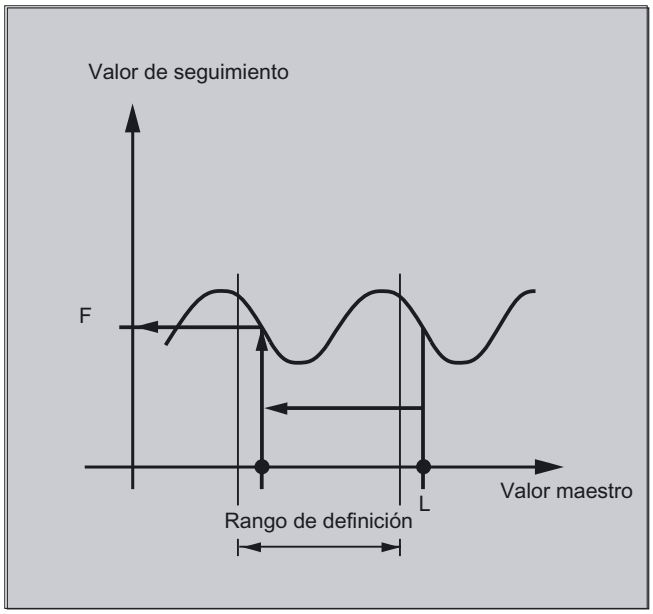
CTAB en tablas de levas no periódicas

Si el <Valor maestro> indicado está fuera de la zona de definición, se emite como valor esclavo el límite superior o inferior:



CTAB en tablas de levas periódicas

Si el <Valor maestro> indicado está fuera de la zona de definición, se evalúa el valor maestro Módulo de la zona de definición y se emite el valor esclavo correspondiente:



Valor aproximado para CTABINV

El comando `CTABINV` requiere un valor aproximado para el valor maestro esperado. `CTABINV` devuelve el valor maestro más cercano al valor aproximado. El valor aproximado puede ser, por ejemplo, el valor maestro del ciclo de interpolación anterior.

Paso de la función de la tabla de levas

La emisión del paso (`<Gradiente>`) permite calcular la velocidad del eje maestro o del eje esclavo en la posición correspondiente.

Indicación del eje maestro o del eje esclavo

La indicación opcional del eje maestro y/o del eje esclavo es importante en el caso de que dichos ejes hayan sido configurados con diferentes unidades de longitud.

CTABSSV, CTABSEV

En los siguientes casos, los comandos `CTABSSV` y `CTABSEV` **no** son aptos para consultar segmentos programados:

- Se han programado círculos o evolventes.
- El achaflanado o redondeo con`CHF`/`RND` está activo.
- El matado de esquinas con `G643` está activo.
- La compresión de secuencias CN con `COMPON`/`COMPCURV`/`COMPCAD` está activa.

9.2.7 Tablas de levas: Comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Función

Estos comandos ofrecen al programador la posibilidad de obtener información actual sobre la ocupación de los recursos para tablas de levas, segmentos de tablas y polinomios.

Sintaxis

```
CTABNO
CTABNOMEM(<Ubicación>)
CTABFNO(<Ubicación>)
CTABSEGID(<n>,<Ubicación>)
CTABSEG(<Ubicación>,<Tipo de segmento>)
CTABFSEG(<Ubicación>,<Tipo de segmento>)
CTABMSEG(<Ubicación>,<Tipo de segmento>)
CTABPOLID(<n>)
CTABPOL(<Ubicación>)
CTABFPOL(<Ubicación>)
CTABMPOL(<Ubicación>)
```

Descripción

CTABNO	Calcular el número total de tablas de levas **definidas** (en la memoria CN estática y dinámica)
CTABNOMEM	Calcular el número de tablas de levas **definidas** en la <Ubicación> indicada
CTABFNO	Calcular el número de tablas de levas **aún posibles** en la <Ubicación> indicada
CTABSEGID	Calcular el número de segmentos de curva del <Tipo de segmento> indicado utilizados por la tabla de levas con el número <n>
CTABSEG	Calcular el número de segmentos de curva **utilizados** del <Tipo de segmento> indicado en la <Ubicación> especificada
CTABFSEG	Calcular el número de segmentos de curva **aún posibles** del <Tipo de segmento> indicado en la <Ubicación> especificada
CTABMSEG	Calcular el número **máximo posible** de segmentos de curva del <Tipo de segmento> indicado en la <Ubicación> especificada
CTABPOLID	Calcular el número de polinomios de leva utilizados por la tabla de levas con el número <n>
CTABPOL	Calcular el número de polinomios de leva **utilizados** en la <Ubicación> indicada
CTABFPOL	Calcular el número de polinomios de leva **aún posibles** en la <Ubicación> indicada
CTABMPOL	Calcular el número **máximo posible** de polinomios de leva en la <Ubicación> indicada
<n>	Número (**ID**) de la tabla de levas
<Ubicación>	Indicación de la ubicación (opcional) "SRAM" Memoria CN **estática** "DRAM" Memoria CN **dinámica** **Nota:** Si no se programa ningún valor para este parámetro, se utiliza la ubicación estándar ajustada con DM20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.
<Tipo de segmento>	Indicación del tipo de segmento (opcional) "L" Segmentos lineales "P" Segmentos de polinomio **Nota:** Si no se programa ningún valor para este parámetro, se emite la suma de los segmentos lineales y de polinomio.

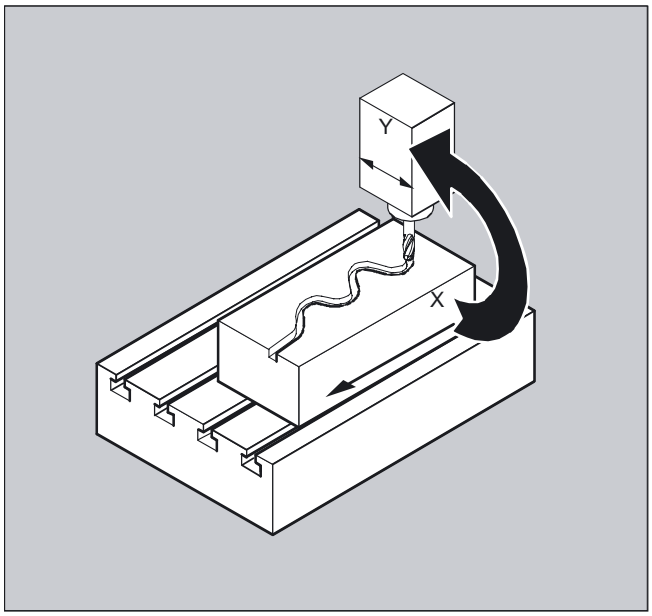
9.3 Acoplamiento entre ejes de valores maestros (LEADON, LEADOF)

Nota

¡Esta función no está disponible para SINUMERIK 828D!

Función

En el acoplamiento entre ejes de valores maestros se desplazan sincrónicamente un eje maestro y un eje arrastrado. En este caso, la posición correspondiente del eje arrastrado está asignada unívocamente, a través de una tabla de levas o del polinomio calculado a partir de dicha tabla, a una posición eventualmente simulada del eje maestro.



Eje maestro es el eje que suministra los valores iniciales para la tabla de levas. **Eje arrastrado** es el eje que asume las posiciones calculadas a través de la tabla de levas.

Acoplamiento de valores reales y de consigna

Como valores maestros, es decir, los valores iniciales para determinar el posicionamiento del eje arrastrado, se pueden utilizar:

- Valores reales de la posición del eje maestro: acoplamiento de valores reales
- Valores de consigna de la posición del eje maestro: acoplamiento de valores de consigna

El acoplamiento de valores maestros está referido siempre al sistema de coordenadas básico.

Para la creación de tablas de levas, ver el apartado "Tablas de levas".
Para el acoplamiento de valores de control, ver /FB/, M3, Arrastre, Acoplamiento de valores de control.

Sintaxis

```
LEADON(Feje,Leje,n)
LEADOF(Feje,Leje)
```

o desconexión sin indicar el eje maestro:
```
LEADOF(Feje)
```

El acoplamiento de valores maestros se puede activar y desactivar tanto desde el programa de piezas como también durante el desplazamiento de acciones síncronas (apartado "Acciones síncronas a desplazamientos").

Descripción

`LEADON`	Activar acoplamiento de valores maestros
`LEADOF`	Desactivar acoplamiento de valores maestros
`Feje`	Eje arrastrado o esclavo
`Leje`	Eje maestro
`n`	Número de la tabla de levas
`$SA_LEAD_TYPE`	Conmutación entre acoplamiento de valores de consigna y de valores reales

Desactivar el acoplamiento de valores maestros, LEADOF

Con la desactivación del acoplamiento de los valores maestros, el eje arrastrado vuelve a ser el eje de comando normal.

Acoplamiento entre ejes de valores maestros y diferentes estados operativos, RESET

Los acoplamientos de valores maestros se desactivan con RESET, dependiendo del ajuste en el dato de máquina.

Ejemplo: Acoplamiento de valores maestros desde acción síncrona

En una línea de prensas se desea sustituir un acoplamiento mecánico convencional entre un eje maestro (eje de estampado) y los ejes de un sistema de transferencia, compuesto de ejes de transferencia y auxiliares, por un sistema de acoplamiento electrónico.

Demuestra la forma de sustituir un sistema de transferencia mecánico por uno electrónico en una línea de prensas. Las operaciones de acoplamiento y desacoplamiento han sido concebidas como **acciones síncronas estáticas**.

El eje maestro LW (eje de estampado) manda de forma definida los ejes de transferencia y auxiliares, los ejes arrastrados, a través de tablas de levas.

Ejes esclavos

Eje longitudinal o de avance (X)
Eje transversal o de cierre (YL)
Eje de elevación (ZL)
Eje auxiliar, avance de rodillos (U)
Eje auxiliar, cabeza de ajuste (V)
Eje auxiliar, engrase (W)

Acciones

En las acciones síncronas se presentan, p. ej., las siguientes acciones:

- Acoplamiento `LEADON (eje arrastrado, eje maestro, número de tabla de levas)`
- Desacoplamiento, `LEADOF (eje arrastrado, eje maestro)`
- Poner valor real, `PRESETON (eje, valor)`
- Definir metas, `$AC_META[i]= valor`
- Modo de acoplamiento: valor maestro real/virtual
- Aproximación a posiciones de eje, `POS[eje]=valor`

Condiciones

Como condiciones se evalúan entradas digitales rápidas, variables de tipo real `$AC_META` y comparaciones de posición enlazadas con el operando lógico AND.

Nota

En el ejemplo siguiente se utilizaron cambios de línea, sangrado y **negrita**exclusivamente para mejorar la legibilidad de la programación. En el control numérico todo el contenido bajo un número de línea contiene una sola línea.

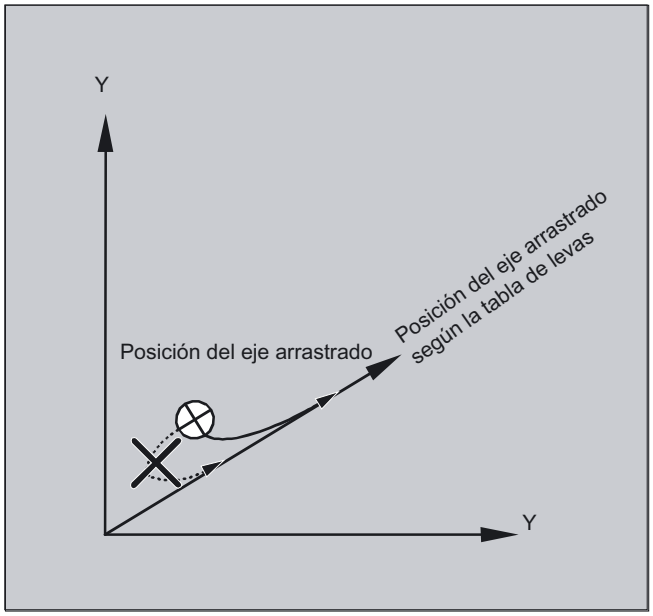
Código del programa	**Comentarios**
	; Define todas las acciones síncronas estáticas.
	; ****Resetear meta
N2 $AC_META[0]=0 $AC_META[1]=0 $AC_META[2]=0 $AC_META[3]=0 $AC_META[4]=0 $AC_META[5]=0 $AC_META[6]=0 $AC_META[7]=0	
	; **** E1 0=>1 Acoplamiento de transferencia CON
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_META[0]==0) DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) $AC_META[0]=1	
	; **** E1 0=>1 Acoplamiento Avance de rodillos CON
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_META[5]==0) DO LEADON(U,LW,4) PRESETON(U,0) $AC_META[5]=1	
	; **** E1 0->1 Acoplamiento Cabeza de ajuste CON
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_META[6]==0) DO LEADON(V,LW,4) PRESETON(V,0) $AC_META[6]=1	
	; **** E1 0->1 Acoplamiento Engrase CON
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_META[7]==0) DO LEADON(W,LW,4) PRESETON(W,0) $AC_META[7]=1	
	; **** E2 0=>1 Acoplamiento DES
N30 IDS=3 EVERY ($A_IN[2]==1) DO LEADOF(X,LW) LEADOF(YL,LW) LEADOF(ZL,LW) LEADOF(U,LW) LEADOF(V,LW) LEADOF(W,LW) $AC_META[0]=0 $AC_META[1]=0 $AC_META[3]=0 $AC_META[4]=0 $AC_META[5]=0 $AC_META[6]=0 $AC_META[7]=0	
....	
N110 G04 F01	
N120 M30	

Descripción

Para el acoplamiento de valores maestros se requiere la sincronización de los ejes maestro y arrastrado. Esta sincronización sólo puede lograrse si, al activar dicho acoplamiento, el eje arrastrado se encuentra dentro de la zona de tolerancia del desarrollo de la curva calculado en la tabla de levas.

La zona de tolerancia para la posición del eje arrastrado está definida en el dato de máquina `DM 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE`.

Si el eje arrastrado no se encuentra todavía en la posición correspondiente al activarse el acoplamiento de valores maestros, se establece la marcha síncrona de forma automática tan pronto el valor de posicionamiento calculado para el eje arrastrado se aproxima a la posición real de dicho eje. Durante la sincronización, el eje arrastrado se desplaza en el sentido definido por la velocidad de consigna de dicho eje (calculado en base a la velocidad del eje maestro y la tabla de levas CTAB).

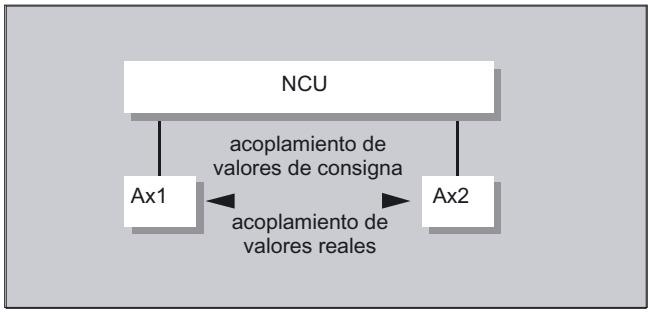


Sin marcha síncrona

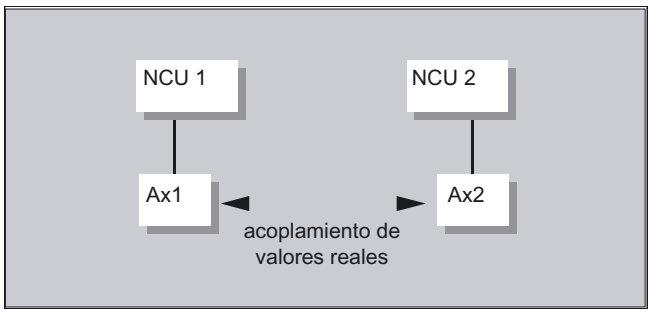
Si al activar el acoplamiento de valores maestros la posición de consigna del eje arrastrado difiere de la posición actual de dicho eje, no se establece la marcha síncrona.

Acoplamiento de valores reales y de valores de consigna

El acoplamiento de valores de consigna proporciona, en comparación con el de valores reales, un mejor sincronismo entre los ejes maestro y arrastrado, por lo que está ajustado de modo estándar.



El acoplamiento de valores de consigna sólo es posible cuando el eje maestro y el arrastrado son interpolados con la misma NCU. En caso de eje maestro externo, el eje arrastrado solo se puede acoplar al eje maestro a través de valores reales.



La **conmutación** es posible a través del dato de operador `$SA_LEAD_TYPE`.

La conmutación entre acoplamiento de valores reales y de consigna se debe realizar siempre estando parado el eje arrastrado. En efecto, sólo se resincroniza en una parada o tras una conmutación.

Ejemplo de aplicación

Los valores reales no se pueden leer correctamente en caso de grandes vibraciones de la máquina. Por consiguiente, al utilizar el acoplamiento de valores maestros durante la transferencia de prensa, en operaciones de grandes vibraciones puede resultar necesario conmutar del acoplamiento los valores reales al de valores de consigna.

Simulación de valores maestros en el acoplamiento de valores de consigna

El interpolador para el eje maestro se puede separar del servo a través de los datos de máquina. Por consiguiente, en el acoplamiento de valores de consigna se pueden generar valores de consigna sin que el eje maestro se haya desplazado realmente.

Los valores maestros generados con el acoplamiento de valores de consigna se leen, para su aplicación (p. ej., en acciones síncronas) de las variables siguientes:

- $AA_LEAD_P	Valor maestro posición
- $AA_LEAD_V	Valor maestro velocidad

Crear valores maestros

Los valores maestros se pueden generar opcionalmente con otros procedimientos programados. Los valores maestros generados de esta manera se escriben en la variable

- $AA_LEAD_SP	Valor maestro posición
- $AA_LEAD_SV	Valor maestro velocidad

y se leen desde allí. Para utilizar estas variables es necesario definir correspondientemente el dato de operador `$SA_LEAD_TYPE = 2`.

Estado del acoplamiento

Desde el programa de pieza se puede consultar el estado del acoplamiento mediante las variables del sistema siguientes:

`$AA_COUP_ACT[eje]`
0: ningún acoplamiento activo
16: acoplamiento de valores maestro activos

Gestión de estados en acciones síncronas

Los procesos de conmutación y de acoplamiento se gestionan a través de variables de tiempo real:

`$AC_MARKER[i] = n`
gestiona con:
i Número de meta (lábel)
n Valor de estado

9.4 Reductor electrónico (EG)

Funcionamiento

Con la función “Reductor electrónico” es posible controlar el movimiento de un **eje esclavo** según una secuencia de desplazamiento lineal dependiente de hasta cinco **ejes maestros**. El factor de acoplamiento define las relaciones entre los ejes maestro y el eje esclavo para cada eje maestro.

La proporción de movimiento del eje esclavo se forma mediante adición en base a las proporciones de movimiento del eje maestro multiplicado por los correspondientes factores de acoplamiento. Al activar un conjunto de ejes EG se puede forzar la sincronización del eje esclavo a una posición definida. Un conjunto de reductor se puede:

- Definir
- Activar
- Desactivar
- Borrar

El movimiento del eje esclavo se puede derivar, a elección, de los:

- Valores de consigna de los ejes maestros
- Valores reales de los ejes maestros

Como ampliación se pueden realizar también relaciones no lineales entre los ejes maestro y el eje esclavo a través de **tablas de levas** (ver cap. Comportamiento de contorneado). Los reductores electrónicos se pueden conectar en cascada; es decir, que el eje esclavo de un reductor electrónico puede actuar como eje maestro para otro reductor electrónico.

9.4.1 Definición del reductor electrónico (EGDEF)

Función

Un conjunto de ejes EG se define mediante la indicación del eje esclavo, y por lo menos uno y como máximo cinco ejes maestros con el correspondiente tipo de acoplamiento.

Requisito

Requisito para la definición de un conjunto de ejes EG:

No debe haberse definido ningún acoplamiento de eje para el eje esclavo (si la hay, se debe borrar dicha definición previamente con `EGDEL`).

Sintaxis

```
EGDEF(Eje esclavo,Eje maestro1,Tipo de acoplamiento1,Eje
maestro2,Tipo de acoplamiento2,...)
```

Descripción

`EGDEF`	Definición de un reductor electrónico
`Eje arrastrado o esclavo`	Eje influido por los ejes maestro
`Eje maestro1 ,..., Eje maestro5`	Ejes que tienen influencia en el eje esclavo
`Tipo de acoplamiento1 ,..., Tipo de acoplamiento5`	Tipo de acoplamiento El tipo de acoplamiento no ha de ser igual para todos los ejes maestro y ha de indicarse individualmente para cada uno de los ejes maestro.

Valor:	Significado:
0	El eje esclavo recibe la influencia del **valor real** del eje maestro correspondiente.
1	El eje esclavo recibe la influencia del **valor de consigna** del eje maestro correspondiente.

Nota

Los factores de acoplamiento se preasignan con cero mediante la definición del conjunto de acoplamientos EG.

Nota

`EGDEF` activa la parada de decodificación previa. La definición de reductor con `EGDEF` también se tiene que utilizar de forma invariable si, en sistemas actúan uno o varios ejes maestro a través de la **tabla de levas** en el eje esclavo.

Ejemplo

Código del programa	**Comentarios**
`EGDEF(C,B,1,Z,1,Y,1)`	`; Definición de un conjunto de ejes EG. Los ejes maestros B, Z, Y tienen influencia en el eje esclavo C a través del valor de consigna.`

9.4.2 Activación del reductor electrónico (EGON, EGONSYN, EGONSYNE)

Función

Para la activación de un conjunto de ejes EG existen 3 variantes.

Sintaxis

Variante 1:

El conjunto de ejes EG se activa de manera selectiva sin sincronización con:

```
EGON(FA,"Modo de cambio de
secuencia",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)
```

Variante 2:

El conjunto de ejes EG se activa de manera selectiva con sincronización con:

```
EGONSYN(FA,"Modo de cambio de
secuencia",SynPosFA,[,LAi,SynPosLAi,Zi,Ni])
```

Variante 3:

El conjunto de ejes EG se activa de manera selectiva con sincronización y el modo de aproximación se predetermina con:

```
EGONSYNE(FA,"Modo de cambio de secuencia",SynPosFA,Modo de
aproximación[,LAi,SynPosLAi,Zi,Ni])
```

Descripción

Variante 1:

`FA`	Eje arrastrado o esclavo	
`Modo de cambio de secuencia`	Se pueden utilizar los siguientes modos:	
	`"NOC"`	El cambio de secuencia se lleva a cabo de inmediato
	`"FINE"`	El cambio de secuencia se lleva a cabo con "Marcha síncrona fina"
	`"COARSE"`	El cambio de secuencia se lleva a cabo con "Marcha síncrona basta"
	`"IPOSTOP"`	El cambio de secuencia se lleva a cabo con la sincronización del valor de consigna
`LA1, ... LA5`	Ejes maestros	
`Z1, ... Z5`	Numerador para el factor de acoplamiento i	
`N1, ... N5`	Denominador para el factor de acoplamiento i	
	Factor de acoplamiento i = numerador i / denominador i	

Sólo deben programarse los ejes maestros que se hayan especificado antes con `EGDEF`. Como mínimo, debe programarse un eje maestro.

Variante 2:

`FA`	Eje arrastrado o esclavo	
`Modo de cambio de secuencia`	Pueden utilizarse los modos siguientes:	
	`"NOC"`	El cambio de secuencia se lleva a cabo de inmediato
	`"FINE"`	El cambio de secuencia se lleva a cabo con "Marcha síncrona fina"
	`"COARSE"`	El cambio de secuencia se lleva a cabo con "Marcha síncrona basta"
	`"IPOSTOP"`	El cambio de secuencia se lleva a cabo con la sincronización del valor de consigna
`[,LAi,SynPosLAi,Zi,Ni]`	(no escribir los corchetes)	
	Mín. 1, máx. 5 secuencias de:	
`LA1, ... LA5`	Ejes maestros	
`SynPosLAi`	Posición síncrona para el eje maestro i.	
`Z1, ... Z5`	Numerador para el factor de acoplamiento i	
`N1, ... N5`	Denominador para el factor de acoplamiento i	
	Factor de acoplamiento i = Numerador i/Denominador i	

Sólo deben programarse ejes maestros que se hayan especificado antes con `EGDEF`. Con las "Posiciones síncronas" programadas para el eje esclavo (`SynPosFA`) y para los ejes maestro (`SynPosLA`) se definen las posiciones en las que el conjunto de acoplamientos es considerado *síncrono*. Si el reductor electrónico no se encuentra en estado síncrono al ser activado, se desplaza el eje esclavo a su posición síncrona definida.

Variante 3:

Los parámetros equivalen a los de la variante 2, además de:

`Modo de aproximación`	Pueden utilizarse los modos siguientes:	
	`"NTGT"`	Desplazamiento optimizado en el tiempo al próximo hueco entre dientes
	`"NTGP"`	Desplazamiento optimizado en el recorrido al próximo hueco entre dientes
	`"ACN"`	Desplazamiento de eje giratorio en sentido de giro negativo, absoluto
	`"ACP"`	Desplazamiento de eje giratorio en sentido de giro positivo, absoluto
	`"DCT"`	Con optimización del tiempo para la posición síncrona programada
	`"DCP"`	Con optimización del recorrido para la posición síncrona programada

La variante 3 sólo actúa en ejes esclavo de valor módulo acoplados a ejes maestro de valor módulo. La optimización en el tiempo tiene en cuenta los límites de velocidad del eje esclavo.

Información adicional

Descripción de las variantes de activación

Variante 1:

Las posiciones de los ejes maestro y del eje esclavo en el momento de la activación se memorizan como "posiciones síncronas". Las "posiciones síncronas" se pueden leer con las variables de sistema `$AA_EG_SYN`.

Variante 2:

Cuando los ejes módulo están en un conjunto de acoplamiento, sus valores se reducen en el módulo. De esta manera, se garantiza que se realice la aproximación a la siguiente posición síncrona posible (la *sincronización relativa*: p. ej., el siguiente hueco entre dientes). Cuando no se ha emitido para el eje esclavo la "Señal de desbloqueo superposición del eje esclavo" interfaz DB(30 +número de eje), DBX 26, bit 4, no se realiza el desplazamiento a la posición síncrona. En vez de ello, se para el programa en la secuencia EGONSYN y se activa la alarma autoborrante 16771 hasta que se activa la señal mencionada.

Variante 3:

La distancia entre dientes (grados) resulta de: 360 * Zi/Ni. En caso de que el eje esclavo esté parado en el momento de la llamada, el eje optimizado en el recorrido se comporta exactamente igual que el optimizado en el tiempo.

Si el eje esclavo ya está en marcha, se sincroniza con `NTGP`, independientemente de la velocidad actual del eje esclavo, en el siguiente hueco entre dientes. Si el eje esclavo ya está en marcha, se sincroniza con `NTGT`, en función de la velocidad actual del eje esclavo, en el siguiente hueco entre dientes. Para este fin, el eje también se frena si es necesario.

Tablas de levas

En caso de utilizar una **tabla de levas** para alguno de los ejes maestros, es obligatorio:

Ni	Poner a 0 el denominador del factor de acoplamiento de los acoplamientos lineales. (El denominador 0 no estaría permitido para acoplamientos lineales). El denominador cero indica para el control que
Zi	se tiene que interpretar como número de la tabla de levas a utilizar. La tabla de levas con el número indicado ya tiene que estar definida en el momento de la activación.
LAi	La especificación del eje maestro equivale a la especificación del eje maestro en caso de acoplamiento con factor de acoplamiento (acoplamiento lineal).

Para más información sobre el uso de tablas de levas y la conexión en cascada de reductores electrónicos y su sincronización, ver:
Bibliografía:
Manual de funciones especiales; Acoplamientos de ejes y ESR (M3), apartado "Acoplamiento de valores de control".

Comportamiento del reductor electrónico con Power On, RESET, cambio del modo de operación, búsqueda

- Tras Power On no hay **ningún** acoplamiento activo.
- Los acoplamientos activos permanecen a pesar de RESET y cambio del modo de operación.
- En caso de búsqueda de secuencias no se ejecutan ni se recogen, sino que se pasan por alto las órdenes para conmutar, borrar o definir el reductor electrónico.

Variables de sistema del reductor electrónico

Con las variables de sistema del reductor electrónico, el programa de pieza puede averiguar y si fuera necesario, reaccionar frente a estados actuales de un conjunto de ejes EG.

Las variables del sistema del reductor electrónico se identifican de la siguiente manera:

$AA_EG_ ...

o

$VA_EG_ ...

Bibliografía:
Manual de las variables del sistema

9.4.3 Desactivación del reductor electrónico (EGOFS, EGOFC)

Función

Para la desactivación de un conjunto activo de ejes EG existen 3 variantes.

Programación

Variante 1:

Sintaxis	Descripción
`EGOFS(eje esclavo)`	Se desconecta el reductor electrónico. El eje esclavo se frena hasta detenerlo. La llamada provoca una parada de decodificación previa.

Variante 2:

Sintaxis	Descripción
`EGOFS(Eje esclavo,Eje maestro1,…,Eje maestro5)`	Esta parametrización del comando permite **selectivamente** eliminar la influencia que ejercen los ejes maestro individuales sobre el movimiento del eje esclavo.

Debe indicarse, por lo menos, un eje maestro. Se desconecta de forma intencionada la influencia de los ejes maestro sobre el eje esclavo. La llamada provoca una parada de decodificación previa. Si aún permanecen ejes maestros activos, el eje esclavo sigue el desplazamiento bajo su influencia. Una vez eliminadas todas las influencias de ejes maestros de esta manera, se frena el eje esclavo hasta detenerlo.

Variante 3:

Sintaxis	Descripción
`EGOFC(Cabezal esclavo1)`	Se desconecta el reductor electrónico. El cabezal esclavo sigue su marcha con la velocidad de giro/velocidad actuales en el momento de desconexión. La llamada provoca una parada de decodificación previa.

Nota

Esta variante sólo está permitida para cabezales.

9.4.4 Borrar la definición de un reductor electrónico (EGDEL)

Función

Un conjunto de ejes EG debe estar desactivado antes de que pueda borrarse su definición.

Programación

Sintaxis	Descripción
`EGDEL(eje esclavo)`	Se borra la definición de acoplamiento del conjunto de ejes. Hasta alcanzar el número máximo de conjuntos de ejes activados simultáneamente pueden definirse nuevamente otros conjuntos de ejes con `EGDEF`. La llamada provoca una parada de decodificación previa.

9.4.5 Avance por vuelta (G95)/Reductor electrónico (FPR)

Función

Con el comando `FPR` también puede indicarse el eje esclavo de un reductor electrónico como eje determinante del avance por vuelta. En tal caso rige el siguiente comportamiento:

- El avance depende de la velocidad nominal del eje esclavo del reductor electrónico.
- La velocidad de consigna se calcula en base a las velocidades de los cabezales maestro y los ejes maestros tipo módulo (que no son ejes de contorneado) y sus factores de acoplamiento asignados.
- No se tienen en cuenta componentes de velocidad de ejes maestro lineales o bien no módulo y movimientos superpuestos del eje esclavo.

9.5 Cabezal síncrono

Función

En modo de cabezal síncrono siempre tenemos un cabezal maestro (CM) y un cabezal esclavo (CE). De esta forma se define el **Par de cabezales síncronos**. El cabezal esclavo sigue los movimientos del cabezal maestro desde el momento en que se activa un acoplamiento (modo síncrono) y atendiendo al tipo de acoplamiento definido mediante los parámetros.

El par de cabezales síncronos se puede definir de forma permanente para una determinada máquina mediante los datos de máquina específicos de canal, o bien se puede definir por el usuario desde el programa de pieza. Se pueden definir hasta 2 pares de cabezales síncronos simultáneamente para un mismo canal.

Desde el programa de pieza, el acoplamiento se puede:

- Definir o modificar
- Conectar
- Desconectar
- Borrar

Además, dependiendo de la versión de software, se puede:

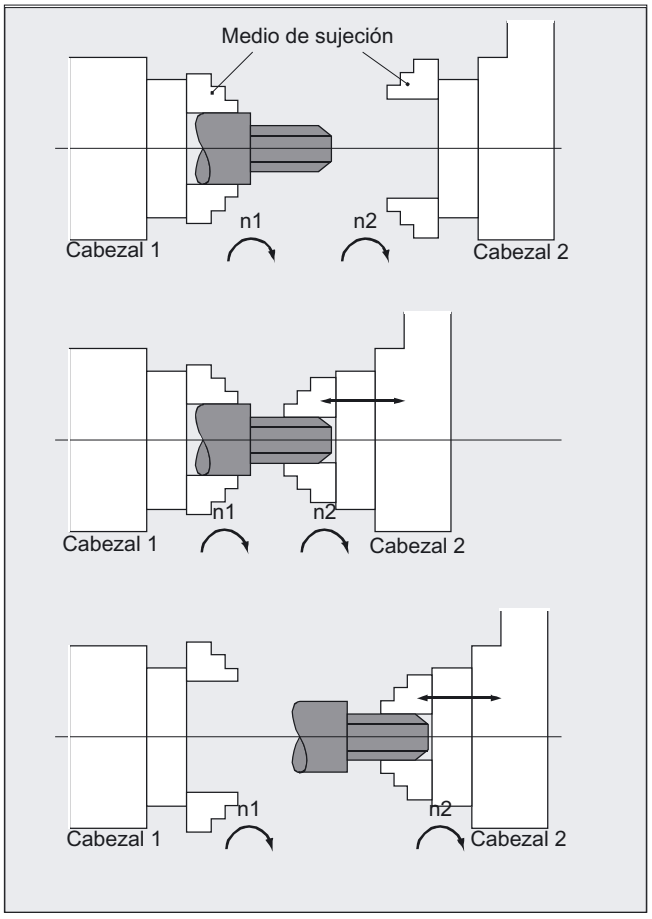
- Esperar a que se cumplan las condiciones para la sincronización
- Modificar el comportamiento para el cambio de secuencia
- Seleccionar el modo de acoplamiento como acoplamiento de valores de consigna o de valores reales, o bien definir el decalaje angular entre los cabezales maestro y esclavo
- Aplicar una programación anterior del cabezal esclavo al activar el acoplamiento
- Corregir una desviación de sincronización medida o ya conocida

9.5.1 Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

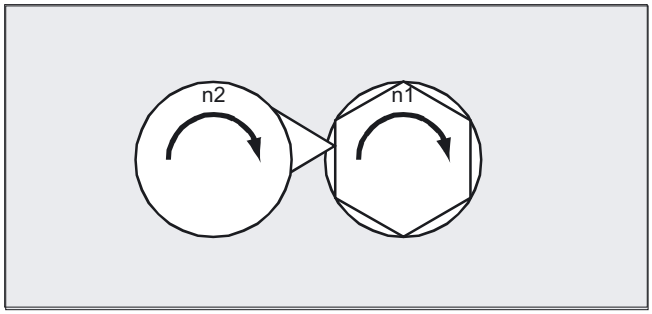
Función

La función de cabezal síncrono permite el desplazamiento síncrono de dos cabezales (cabezal esclavo, FS, y cabezal maestro, LS), p. ej. para la transferencia al vuelo de la pieza. La función ofrece los siguientes modos:

- Sincronismo de la velocidad de giro ($n_{FS} = n_{LS}$)
- Sincronismo de posición ($\phi_{FS} = \phi_{LS}$)
- Sincronismo de posición con un decalaje angular adicional ($\phi_{FS} = \phi_{LS}$)



Definiendo una relación de transformación distinta de 1 entre el cabezal maestro y el cabezal esclavo también se pueden mecanizar poliedros (torneado de polígonos).



Sintaxis

```
COUPDEF(FS,LS,ÜFS,ÜLS,cambio de secuencia,tipo de acoplamiento)
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,cambio de secuencia,LS,cambio de secuencia)
```

Nota

Notación abreviada

En las instrucciones `COUPOF`, `COUPOFS`, `COUPRES` y `COUPDEL` se puede utilizar una notación abreviada sin indicar el cabezal maestro.

Descripción

COUPDEF	Definición/modificación del acoplamiento específica del usuario
COUPON	Activación de un acoplamiento. El cabezal esclavo se sincroniza con el cabezal maestro partiendo de la velocidad de giro actual.
COUPONC	Aplicación del acoplamiento con la programación anterior de M3 S... o M4 S... durante la activación. Se aplica inmediatamente una velocidad de giro diferencial del cabezal esclavo.
COUPOF	Desactivación de un acoplamiento. • Con cambio de secuencia inmediato: COUPOF(S2,S1) • Cambio de secuencia tras sobrepasar las posiciones de desactivación POSFS o POSLS: COUPOF(S2,S1,POSFS) COUPOF(S2,S1,POSFS,POSLS)
COUPOFS	Desactivación de un acoplamiento con parada del cabezal esclavo. Cambio de secuencia con la máxima rapidez con cambio de secuencia inmediato: COUPOFS(S2,S1) Cambio de secuencia tras sobrepasar la posición de desconexión: COUPOFS(S2,S1,POSFS)
COUPRES	Puesta a cero de los parámetros de un acoplamiento a los DM y DO configurados
COUPDEL	Borrado de un acoplamiento definido por el usuario
WAITC	Esperar a que se cumplan las condiciones de sincronización (se anulan los NOC a IPO en caso de cambio de secuencia)
FS	Nombre del cabezal esclavo

Parámetros opcionales:

LS	Nombre del cabezal maestro	
	Indicación con número de cabezal: p. ej. S2, S1	
ÜFS, ÜLS	Relación de transformación entre el cabezal esclavo y el cabezal maestro.	
	ÜFS = numerador, ÜLS = denominador	
	Ajuste previo: ÜFS/ÜLS = 1.0; indicación del denominador opcional	
Cambio de secuencia	Comportamiento en cambio de secuencia	
	El cambio de secuencia se realiza:	
	"NOC"	inmediatamente
	"FINE"	al alcanzar "marcha síncrona fina"
	"COARSE"	al alcanzar "marcha síncrona basta"
	"IPOSTOP"	al alcanzar IPOSTOP, es decir, tras la marcha síncrona del valor de consigna (ajuste previo)
	El comportamiento en cambio de secuencia es modal.	
Modo de acoplamiento	Modo de acoplamiento: Acoplamiento entre cabezal esclavo y cabezal maestro	
	"DV"	Acoplamiento de valor de consigna (tipo de acoplamiento por defecto)
	"AV"	acoplamiento de valores reales
	"VV"	Acoplamiento de velocidad
	El tipo de acoplamiento es válido de forma modal.	
POSFS	Decalaje angular entre los cabezales maestro y esclavo	
	Rango de valores:	0°… 359,999°
POSFS,POSLS	Posiciones de desconexión de cabezal esclavo y maestro	
	"El cambio de secuencia se habilita después de sobrepasar POS_{FS}, POS_{LS}"	
	Rango de valores:	0°… 359,999°

Ejemplos

Ejemplo 1: Programación de cabezales maestro y esclavo

```
Programación                              Comentarios
                                          ; Cabezal guía = Cabezal maestro = Cabezal 1
                                          ; Cabezal esclavo = Cabezal 2
N05 M3 S3000 M2=4 S2=500                  ; El cabezal maestro gira a 3000 rpm,
                                          ; el cabezal esclavo gira a 500 rpm
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")         ; Def. del acoplamiento (también puede configurarse)
...
N70 SPCON                                 ; Tomar cabezal maestro en regulación de posición
                                            (acoplamiento de valores de consigna)
N75 SPCON(2)                              ; Tomar cabezal esclavo en regulación de posición
N80 COUPON(S2,S1,45)                      ; Acoplamiento al vuelo con offset de 45 grados
...
N200 FA[S2]=100                           ; Velocidad de posicionado =  100 grados/min
N205 SPOS[2]=IC(-90)                      ; Superponer 90 grados en sentido negativo al valor de
                                            posición
N210 WAITC(S2,"Fine")                     ; Esperar a la sincronización "fina"
N212 G1 X... Y... F...                    ; Mecanizado
...
N215 SPOS[2]=IC(180)                      ; Superponer 180 grados en sentido positivo al valor de
                                            posición
N220 G4 S50                               ; Tiempo de espera = 50 vueltas del cabezal maestro
N225 FA[S2]=0                             ; Activar velocidad configurada (DM)
N230 SPOS[2]=IC(-7200)                    ; 20 vueltas Con velocidad configurada en
                                          ; sentido negativo
...
N350 COUPOF(S2,S1)                        ; Desacoplar al vuelo, S=S2=3000
N355 SPOSA[2]=0                           ; Parar FS a cero grados
N360 G0 X0 Y0                             ;
N365 WAITS(2)                             ; Esperar a cabezal 2
N370 M5                                   ; Parar FS
N375 M30
```

Ejemplo 2: Programación de una velocidad de giro diferencial

```
Programación                    Comentarios
                                ; Cabezal guía = Cabezal maestro = Cabezal 1
                                ; Cabezal esclavo = Cabezal 2
N01 M3 S500                     ; El cabezal maestro gira a 500 rpm
N02 M2=3 S2=300                 ; El cabezal esclavo gira a 300 rpm
...
N10 G4 F1                       ; Tiempo de espera del cabezal maestro
N15 COUPDEF (S2, S1, -1)        ; Factor de acoplamiento con relación de transformación -1:1
N20 COUPON (S2, S1)             ; Activar acoplamiento. La velocidad de giro del cabezal esclavo se
                                ; obtiene a partir de la velocidad de giro del cabezal maestro y el
                                ; factor de acoplamiento
...
N26 M2=3 S2=100                 ; Programación de una velocidad de giro diferencial
```

Ejemplo 3: Ejemplos de la aplicación de un movimiento para la velocidad de giro diferencial

1. Activación del acoplamiento con la programación anterior del cabezal esclavo con COUPON

```
Programación                    Comentarios
                                ; Cabezal guía = Cabezal maestro = Cabezal 1
                                ; Cabezal esclavo = Cabezal 2
N05 M3 S100 M2=3 S2=200         ; El cabezal maestro gira a 100 rpm; el cabezal esclavo gira a 200 rpm
N10 G4 F5                       ; Tiempo de espera = 5 segundos del cabezal maestro
N15 COUPDEF(S2,S1,1)            ; La relación de transformación del cabezal esclavo con respecto al
                                  cabezal maestro es 1,0 (ajuste por defecto)
N20 COUPON(S2,S1)               ; Acoplamiento al vuelo al eje maestro
N10 G4 F5                       ; El cabezal esclavo gira a 100 rpm
```

2. Activación del acoplamiento con la programación anterior del cabezal esclavo con COUPONC

```
Programación                    Comentarios
                                ; Cabezal guía = Cabezal maestro = Cabezal 1
                                ; Cabezal esclavo = Cabezal 2
N05 M3 S100 M2=3 S2=200         ; El cabezal maestro gira a 100 rpm; el cabezal esclavo gira a 200 rpm
N10 G4 F5                       ; Tiempo de espera = 5 segundos del cabezal maestro
N15 COUPDEF(S2,S1,1)            ; La relación de transformación del cabezal esclavo con respecto al
                                  cabezal maestro es 1,0 (ajuste por defecto)
N20 COUPONC(S2,S1)              ; Acoplamiento al vuelo al cabezal maestro y aplicación de la
                                  velocidad de giro anterior a S2
N10 G4 F5                       ; S2 gira a 100 rpm + 200 rpm = 300 rpm
```

3. Activación del acoplamiento con el cabezal esclavo parado con COUPON

```
Programación                    Comentarios
                                ; Cabezal guía = Cabezal maestro = Cabezal 1
                                ; Cabezal esclavo = Cabezal 2
N05 SPOS=10 SPOS[2]=20          ; Cabezal esclavo S2 en modo de posicionamiento
N15 COUPDEF(S2,S1,1)            ; La relación de transformación del cabezal esclavo con respecto al
                                  cabezal maestro es 1,0 (ajuste por defecto)
N20 COUPON(S2,S1)               ; Acoplamiento al vuelo al eje maestro
N10 G4 F1                       ; Se cierra el acoplamiento, S2 permanece a 20 grados
```

4. Activación del acoplamiento con el cabezal esclavo parado con COUPONC

Nota

Modo de posicionamiento o de eje

Si el cabezal esclavo se encuentra en el modo de posicionamiento o en el modo de eje antes del acoplamiento, se comporta del mismo modo con COUPON(FS,LS) que con COUPONC(FS,LS).

ATENCIÓN

Cabezal maestro y modo Eje

Si el cabezal maestro se encuentra en el modo Eje antes de la definición del acoplamiento, tras la activación de éste también actúa el límite de velocidad del dato de máquina:

DM32000 $MA_MAX_AX_VELO (velocidad de eje máxima)

Para evitar este comportamiento, el eje debe pasar al modo Cabezal (M3 S... o M4 S...) antes de la definición del acoplamiento.

Definición de un par de cabezales síncronos

Acoplamiento configurado:

En el acoplamiento configurado, el cabezal maestro y el cabezal esclavo se definen a través de datos de máquina. Los cabezales configurados no pueden modificarse en el programa de pieza. El acoplamiento puede parametrizarse con COUPDEF en el programa de pieza (siempre que no exista protección contra escritura).

Acoplamiento definido por el usuario:

Con COUPDEF puede definirse o modificarse un acoplamiento en el programa de pieza. Si ya hay un acoplamiento activo, éste debe borrarse primero con COUPDEL antes de definir uno nuevo.

Definición de un acoplamiento COUPDEF

Un acoplamiento queda totalmente definido por:

```
COUPDEF(FS, LS, ÜFS, ÜLS, comportamiento en cambio de secuencia, tipo
de acoplamiento)
```

Cabezal esclavo (FS) y cabezal maestro (LS)

Los nombres de eje para el cabezal esclavo (`FS`) y el cabezal maestro (`LS`) se utilizan para definir un acoplamiento de forma unívoca. Los nombres de eje deben programarse con la instrucción `COUPDEF`. Los otros parámetros de acoplamiento son modalmente activos y sólo deben programarse cuando sea necesario modificarlos.

Ejemplo:

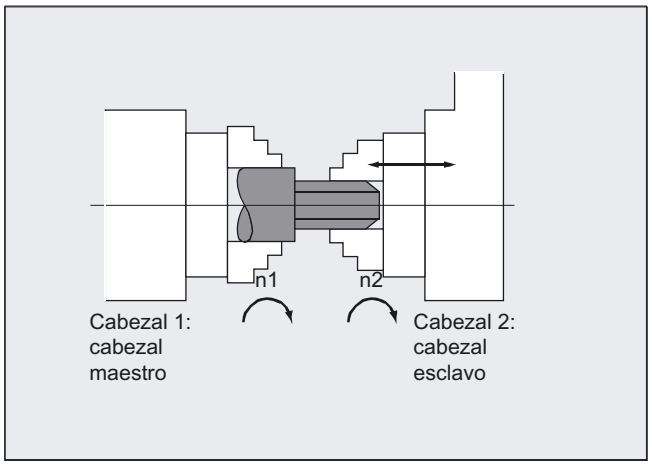
```
COUPDEF(S2, S1)
```

Relación de transformación $\ddot{u}_{FS}/\ddot{u}_{LS}$

La relación de transformación se indica como relación de la velocidad de giro entre el cabezal esclavo (numerador) y el cabezal maestro (denominador). El numerador debe programarse. Si no se programa ningún denominador, éste será 1.0.

Ejemplo: Cabezal esclavo `S2` y cabezal maestro `S1`, relación de transformación = 1/4 = 0.25.

```
COUPDEF (S2, S1, 1.0, 4.0)
```



Nota

La relación de transformación también puede modificarse cuando el acoplamiento está activado y los cabezales girando.

Comportamiento de cambio de secuencia NOC, FINE, COARSE, IPOSTOP

En la programación del comportamiento en cambio de secuencia se puede utilizar la siguiente notación abreviada:

- NO: inmediato (ajuste previo)
- FI: al alcanzar "marcha síncrona fina"
- CO: al alcanzar "marcha síncrona basta"
- IP: al alcanzar IPOSTOP, es decir, tras la marcha síncrona del valor de consigna

Modo de acoplamiento DV, AV

⚠ PRECAUCIÓN
El tipo de acoplamiento solamente se puede modificar cuando se encuentre desactivado.

Activación del modo síncrono COUPON, POS_{FS}

- Activación del acoplamiento con cualquier ángulo de decalaje entre LS y FS:
  - `COUPON(S2, S1)`
  - `COUPON(S2, S1, POSFS)`
  - `COUPON(S2)`
- Activación del acoplamiento con decalaje angular POS_{FS}

  Para el acoplamiento con sincronización de posición en piezas perfiladas.
  POS_{FS} se refiere a la posición 0° del cabezal maestro en sentido de giro positivo
  Margen de valores POS_{FS}: 0°… 359,999°:

  - `COUPON(S2, S1, 30)`

  Se puede utilizar este método para cambiar el ángulo de decalaje incluso cuando el acoplamiento ya se encuentre activado.

Posicionado del cabezal esclavo

Una vez activado el acoplamiento síncrono entre dos cabezales, también se puede realizar un posicionado del cabezal esclavo de ±180° alrededor de la posición de acoplamiento.

Posicionado SPOS

El cabezal esclavo se puede interpolar con `SPOS=...`

Ejemplo:

```
SPOS[2] = IC(-90)
```

Para más información sobre SPOS:

Bibliografía:
Manual de programación, Fundamentos.

Velocidad de giro diferencial M3 S... o M4 S...

La velocidad de giro diferencial se debe a la superposición con signo de dos fuentes de velocidad y se vuelve a programar para el cabezal esclavo, p. ej., con Sn=... o Mn=3, Mn=4 en el modo de control de velocidad en lazo abierto durante un acoplamiento de cabezal síncrono activo. Este componente de la velocidad se deriva del cabezal maestro a través del factor de acoplamiento y se suma al cabezal esclavo con el signo adecuado.

Nota

Con el sentido de giro M3 o M4 también se debe programar de nuevo la velocidad S...; de lo contrario, una alarma avisa de que falta dicha programación.

Para más información sobre la velocidad de giro diferencial, ver

Bibliografía: /FB2/ Manual de funciones de ampliación; Cabezal síncrono (S3)

Velocidad de giro diferencial con COUPONC

Aplicación de un movimiento a la velocidad de giro diferencial

Al activar un acoplamiento de cabezales síncronos con `COUPONC` se superpone la velocidad de giro del cabezal esclavo actualmente activa (`M3 S...` o `M4 S...`).

Nota

Habilitación de la superposición

La superposición de la velocidad de giro de un cabezal (`M3 S...` o `M4 S...`) por medio de un acoplamiento de cabezales síncronos `COUPONC` sólo tiene efecto cuando dicha superposición está habilitada.

Limitación de la dinámica del cabezal maestro

La dinámica del cabezal maestro debe limitarse hasta tal punto que no se sobrepasen los límites de dinámica del cabezal esclavo al producirse una superposición de éste.

Velocidad, aceleración: FA, ACC, OVRA, VELOLIMA

La velocidad y la aceleración axiales de un cabezal esclavo pueden programarse con:

- `FA[SPI(Sn)]` o `FA[Sn]` (velocidad axial)
- `ACC[SPI(Sn)]` o `ACC[Sn]` (aceleración axial)
- `OVRA[SPI(Sn)]` o `OVRA[Sn]` (corrección axial)
- `VELOLIMA[SPI(Sn)]` o `VELOLIMA[Sn]` (aumento o reducción de la velocidad axial)

Con n = 1, 2, 3... (número de cabezal de los cabezales esclavos)

Bibliografía:
Manual de programación, Fundamentos

Nota

Componente de la aceleración JERKLIMA[Sn]

La programación de un aumento o reducción de la velocidad axial no está activa actualmente en los cabezales.

Para más información sobre la configuración de la dinámica axial:

Bibliografía:
/FB2/ Manual de funciones de ampliación; Ejes giratorios (R2)

Comportamiento en cambio de secuencia programable WAITC

`WAITC` permite predefinir el comportamiento en cambio de secuencia, p. ej. después de modificar parámetros de acoplamiento o procesos de posicionado, con diferentes condiciones de marcha síncrona (basta, fina, `IPOSTOP`). Si no se indican condiciones de marcha síncrona, se aplica el comportamiento en cambio de secuencia definido en `COUPDEF`.

Ejemplo:

Esperar hasta que se cumplan las condiciones de marcha síncrona conforme a `COUPDEF`

```
WAITC( )
```

Esperar hasta que se cumpla la condición de marcha síncrona `FINE` en el cabezal esclavo S2 y `COARSE` en el cabezal esclavo S4:

```
WAITC(S2, "FINE", S4, "COARSE")
```

Desactivación del acoplamiento COUPOF

Con `COUPOF` se puede predefinir el comportamiento de desactivación del acoplamiento:

- Desactivación del acoplamiento con cambio de secuencia inmediato:
  - `COUPOF(S2,S1)` (indicando el cabezal maestro)
  - `COUPOF(S2)` (sin indicar el cabezal maestro)
- Desactivación del acoplamiento tras sobrepasar ciertas posiciones de desactivación. El cambio de secuencia tiene lugar después de sobrepasar las posiciones de desactivación.
  - `COUPOF(S2, S1, 150)` (posición de desactivación de FS: 150°)
  - `COUPOF(S2, S1, 150, 30)` (posición de desactivación de FS: 150°, de LS: 30°)

Desactivación del acoplamiento con parada del cabezal esclavo COUPOFS

Con COUPOFS se puede predefinir el comportamiento de desactivación del acoplamiento con parada del cabezal esclavo:

- Desactivación del acoplamiento con parada del cabezal esclavo y cambio de secuencia inmediato:
  - `COUPOFS(S2,S1)` (indicando el cabezal maestro)
  - `COUPOFS(S2)` (sin indicar el cabezal maestro)
- Desactivación del acoplamiento tras sobrepasar ciertas posiciones de desactivación con parada del cabezal esclavo. El cambio de secuencia tiene lugar después de sobrepasar las posiciones de desactivación.
  - `COUPOFS(S2, S1, 150)` (posición de desactivación de FS: 150°)

Borrado de acoplamientos COUPDEL

Mediante COUPDEL se borra el acoplamiento:

- `COUPDEL(S2, S1)` (indicando el cabezal maestro)
- `COUPDEL(S2)` (sin indicar el cabezal maestro)

Puesta a cero de los parámetros de acoplamiento COUPRES

COUPRES se utiliza para activar los valores del acoplamiento parametrizados en los datos de máquina y del operador:

- `COUPRES(S2, S1)` (indicando el cabezal maestro)
- `COUPRES(S2)` (sin indicar el cabezal maestro)

Variables del sistema

Estado de acoplamiento actual del cabezal esclavo

El estado de acoplamiento actual de un cabezal esclavo puede leerse por medio de la siguiente variable de sistema:

`$AA_COUP_ACT[FS]`

Valor	Descripción
0	Ningún acoplamiento activo
4	Acoplamiento de cabezal síncrono activo

Nota
Otros valores de las variables de sistema se refieren al modo Eje

Bibliografía
/LGA1/ Manual de las variables del sistema

Decalaje angular actual

El decalaje angular actual de un cabezal esclavo en relación con el cabezal maestro puede leerse mediante la siguiente variable de sistema:

- `$AA_COUP_OFFS[FS]` (decalaje angular del valor de consigna)
- `$VA_COUP_OFFS[FS]` (decalaje angular del valor real)

Nota

Tras anular la liberación del regulador con un acoplamiento activado y en modo de seguimiento, al volver a liberar el regulador, puede que el ángulo de decalaje entre los dos cabezales se haya modificado. En este caso es posible leer el offset de posición modificado y, si fuese necesario, corregirlo desde el programa de pieza.

9.6 Conjunto maestro-esclavo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

Función

El acoplamiento de maestro/esclavo antes de SW 6.4 permite el acoplamiento de los ejes esclavo a su eje maestro únicamente con los ejes afectados en estado parado.

La ampliación del estado de SW 6.5 permite el acoplamiento y la separación de cabezales de velocidad de giro controlada **en rotación** y la configuración dinámica.

Sintaxis

`MASLON(Slv1,Slv2,..., )`	
`MASLOF(Slv1,Slv2,..., )`	
`MASLDEF(Slv1,Slv2,..., eje maestro)`	Ampliación para la configuración dinámica
`MASLDEL(Slv1,Slv2,..., )`	Ampliación para configuración dinámica
`MASLOFS(Slv1, Slv2, ..., )`	Ampliación para cabezal esclavo

Nota

Con `MASLOF/MASLOFS` se suprime la parada de decodificación previa implícita. Debido a la falta de parada de decodificación previa, las variables de sistema $P no suministran valores actuales para los ejes esclavo hasta el momento de una nueva programación.

Descripción

Generalidades

`MASLON`	Activar un acoplamiento temporal.
`MASLOF`	Separar un acoplamiento activo. En cabezales se tienen que considerar las ampliaciones.

Ampliación configuración dinámica

MASLDEF	Crear/modificar acoplamiento según definición del usuario a través de datos de máquina o desde el programa de pieza.
MASLOFS	Separar el acoplamiento de forma análoga a MASLOF y frenar automáticamente el cabezal esclavo.
MASLDEL	Separar conjunto de ejes maestro-esclavo y borrar definición del conjunto.
Slv1, Slv2, ...	Ejes esclavo guiados por un eje maestro.
Eje maestro	Eje que guía ejes esclavo definidos en un conjunto de maestro-esclavo.

Ejemplos

Ejemplo 1: Configuración dinámica de un acoplamiento maestro-esclavo

Configuración dinámica de un acoplamiento de maestro/esclavo desde el programa de pieza.

El eje relevante después de un giro del contenedor de ejes se tiene que convertir en eje maestro.

```
Código del programa  Comentarios
MASLDEF(AUX,S3)      ; S3 maestro para AUX
MASLON(AUX)          ; Acoplamiento CON para AUX
M3=3 S3=4000         ; Sentido de giro horario
MASLDEL(AUX)         ; Borrar configuración y separar el acoplamiento
AXCTSWE(CT1)         ; Giro del contenedor
```

Ejemplos

Ejemplo 2: Acoplamiento de valores reales de un eje esclavo

Acoplamiento de valores reales de un eje esclavo en el mismo valor que el eje maestro mediante PRESETON.

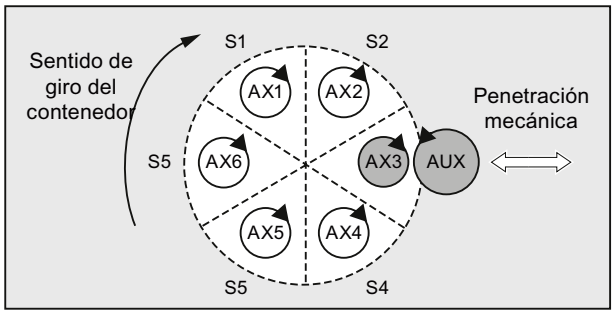
En un acoplamiento permanente de maestro-esclavo se tiene que modificar el valor real en el eje ESCLAVO mediante PRESETON.

Código del programa	**Comentarios**
N37262 $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; Desactivar por unos instantes el acoplamiento permanente.
N37263 NEWCONF	
N37264 STOPRE	
MASLOF(Y1)	; Acoplamiento temporal desactivado.
N5 PRESETON(Y1,0,Z1,0,B1,0,C1,0,U1,0)	; Fijar valor real de los ejes esclavo no referenciados, dado que éstos están activados con Power On.
N37262 $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; Activar acoplamiento permanente.
N37263 NEWCONF	

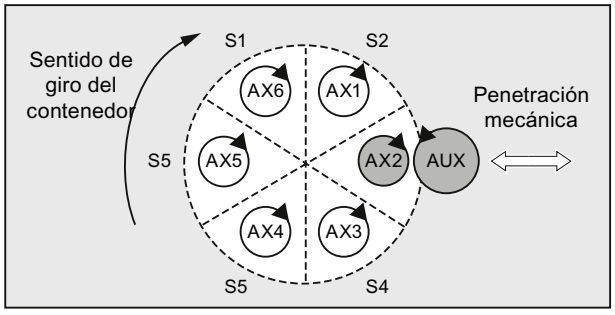
Ejemplo 3: Secuencia de acoplamiento Posición 3/Contenedor CT1

Para que, después del giro de contenedor, se pueda establecer el acoplamiento con otro cabezal, se tiene que separar primero el acoplamiento anterior, borrar la configuración y configurar el nuevo acoplamiento.

Situación inicial:



Después del giro en una ranura:



Bibliografía:

Manual de funciones de ampliación; Varios paneles de operador y NCU (B3), apartado: "Contenedores de ejes"

Información adicional

Generalidades

MASLOF	En el caso de los cabezales que se encuentran en el modo de control de velocidad de giro, esta instrucción se ejecutará inmediatamente. Los cabezales esclavos que giran es este momento mantienen su velocidad de giro hasta que se programe una nueva velocidad de giro.

Ampliación de la configuración dinámica

MASLDEF	Definición de un conjunto maestro-esclavo a partir del programa de pieza. Antes, la definición se realizaba únicamente a través de datos de máquina.
MASLDEL	La instrucción cancela la asignación de los ejes esclavos al eje maestro y al mismo tiempo separa, de forma análoga a MASLOF, el acoplamiento. Las definiciones maestro-esclavo acordadas en los datos de máquina se mantienen.
MASLOFS	MASLOFS puede utilizarse para frenar automáticamente los cabezales esclavos en el momento de la separación del acoplamiento. En el caso de los ejes y los cabezales en modo de posicionamiento, el acoplamiento sólo se cerrará y separará en reposo.

Nota

Para el eje esclavo, el valor real puede sincronizarse mediante PRESETON en el mismo valor que el eje maestro. Para ello, el acoplamiento maestro-esclavo permanente debe desactivarse por unos instantes para fijar el valor real del eje esclavo no referenciado con Power On en el valor del eje maestro. Después se restablece el acoplamiento permanente.

El acoplamiento maestro-esclavo permanente se activa con el ajuste de DM37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1 y no tiene efecto alguno sobre las instrucciones de programación del acoplamiento temporal.

Comportamiento de acoplamiento de los cabezales

En el caso de los cabezales en el modo de control de velocidad de giro, el comportamiento de acoplamiento de MASLON, MASLOF, MASLOFS y MASLDEL se especifica explícitamente mediante el dato de máquina DM37263 $MA_MS_SPIND_COUPLING_MODE.

En el ajuste estándar con DM37263 = 0, el acoplamiento y la separación de los ejes esclavos se realizan únicamente cuando lo ejes implicados están en reposo. MASLOFS equivale a MASLOF.

En el caso de DM37263 = 1, la instrucción de acoplamiento es inmediata y, por lo tanto, se ejecuta en movimiento. El acoplamiento se cierra de inmediato con MASLON y se separa de inmediato con MASLOFS o MASLOF. Los cabezales esclavos que estén girando en ese momento se frenan automáticamente mediante MASLOFS y mantienen, mediante MASLOF, sus velocidades de giro hasta una nueva programación de velocidad de giro.

Acciones síncronas a desplazamiento 10

10.1 Fundamentos

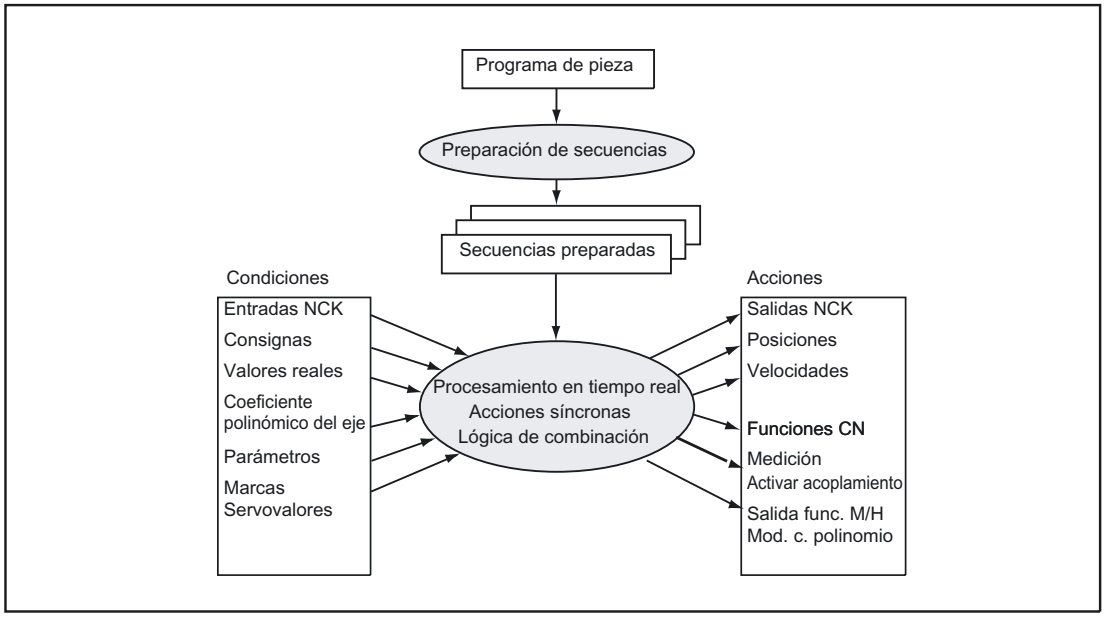
Función

Las acciones síncronas ofrecen la posibilidad de ejecutar acciones sincronizadas con las secuencias de mecanizado.

El momento de ejecución de las acciones puede definirse mediante condiciones. Las condiciones se vigilan en la cadencia de interpolación. Por consiguiente, las acciones constituyen una reacción a sucesos de tiempo real; su ejecución no está restringida a los límites de secuencia.

Además, las acciones síncronas contienen información sobre su vida útil y sobre la frecuencia de consulta de las variables de proceso principal programadas y, por lo tanto, sobre la frecuencia de ejecución de las acciones que se desean activar. Por consiguiente, una acción sólo se puede activar una sola vez o en cambio de forma cíclica (siguiendo la cadencia o ciclo de interpolación).

Aplicaciones posibles

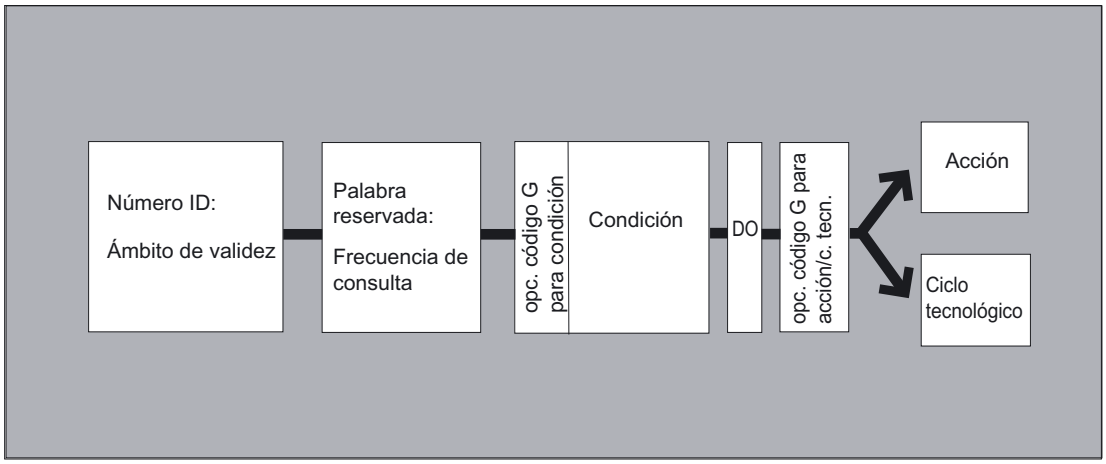


- Optimización de aplicaciones de duración crítica (p. ej.: cambio de herramienta)
- Reacción rápida a sucesos externos
- Programar regulaciones AC
- Materializar funciones de seguridad
- ....

Programación

Una acción síncrona va sola en una secuencia y actúa a partir de la siguiente secuencia ejecutable de una función de máquina (p. ej., desplazamiento con `G0`, `G1`, `G2`, `G3`).

Las acciones síncronas se componen de hasta 5 elementos de mando con funciones diferentes:



Sintaxis

```
DO <Acción1> <Acción2> …
<PALABRA RESERVADA> <Condición> DO <Acción1> <Acción2> …
ID=<n> <PALABRA RESERVADA> <Condición> DO <Acción1> <Acción2> …
IDS=<n> <PALABRA RESERVADA> <Condición> DO <Acción1> <Acción2> …
```

Significado

`DO`	Instrucción para la activación de las acciones programadas Sólo es efectivo si se cumple la `<Condición>` (si está programada) → Ver "Acciones"

`<Acción1>`
`<Acción2>`
…
	Acciones por activar
	Ejemplos:
	• Asignar variable
	• Iniciar ciclo tecnológico

`<PALABRA RESERVADA>`	Mediante la palabra reservada (`WHEN`, `WHENEVER`, `FROM` o `EVERY`) se define la comprobación cíclica de la `<Condición>` de una acción síncrona. → Ver "Comprobación cíclica de la condición"
`<Condición>`	Lógica para variables de proceso principal La condición se comprueba en el ciclo IPO.

`ID=<n>`
o
`IDS=<n>`
	Número de identificación
	Con el número de identificación se especifican el ámbito de validez y la posición dentro de la secuencia de ejecución.
	→ Ver "Ámbito de validez y secuencia de ejecución"

Coordinación de acciones síncronas/ciclos tecnológicos

Para la coordinación de acciones síncronas/ciclos tecnológicos están disponibles los siguientes comandos:

Comando	Descripción
CANCEL(<n>)	Borrar acciones síncronas → Ver "Borrar acción síncrona"
LOCK(<n>)	Bloquear acciones síncronas
UNLOCK(<n>)	Habilitar acciones síncronas
RESET	Poner a cero ciclo tecnológico En relación con LOCK, UNLOCK y RESET: → ver "Bloquear, habilitar, resetear"

Ejemplo

Código del programa	Comentarios
WHEN $AA_IW[Q1]>5 DO M172 H510	; Si el valor real del eje Q1 sobrepasa 5 mm, se emiten las funciones auxiliares M172 y H510 a la interconexión del PLC.

10.1.1 Ámbito de validez y secuencia de ejecución (ID, IDS)

Función

Ámbito de validez

El ámbito de validez de una acción síncrona se especifica mediante la identificación ID o IDS:

Sin ID modal:	Acción síncrona activa secuencia a secuencia en el modo Automático
ID:	Acción síncrona modal en el modo automático hasta el fin de programa
IDS:	Acción síncrona estática, modal en todos los modos de operación, incluso tras el fin de programa

Aplicaciones

- Bucles adaptativos en modo JOG
- Lógica para Safety Integrated
- Funciones de vigilancia, reacciones a estados de máquina en todos los modos de operación

Secuencia de ejecución

Las acciones síncronas modales y estáticas se ejecutan en el orden definido por su número `ID` o `IDS` (`ID=<n>` o `IDS=<n>`) en el ciclo de interpolación.

Las acciones síncronas vigentes secuencialmente (sin número ID) se procesan en el orden programado, tras la ejecución de las acciones síncronas modales.

Nota

Mediante ajustes en los datos de máquina, las acciones síncronas modales pueden protegerse contra modificaciones o borrado (→ ¡Fabricante de la máquina!).

Programación

Sintaxis	Descripción
Sin ID modal	La acción síncrona sólo es activa en **modo automático**. Tiene efecto sólo para la siguiente secuencia ejecutable (secuencia con instrucción de desplazamiento u otra acción de máquina), es decir, tiene vigencia **por secuencia**. Ejemplo: `WHEN $A_IN[3]==TRUE DO $A_OUTA[4]=10`
`ID=<n> ...`	La acción síncrona tiene un efecto **modal** en las secuencias siguientes y se puede desactivar con `CANCEL(<n>)` o sobrescribir programando una nueva acción síncrona con el mismo ID. Las acciones síncronas activas en la secuencia `M30` retardan el fin de programa. Las acciones síncronas ID sólo son efectivas en el **modo automático**. Margen de valores para <n>: 1 ... 255 Ejemplo: `ID=2 EVERY $A_IN[1]==1 DO POS[X]=0`
`IDS=<n>`	Las acciones síncronas estáticas actúan de forma **modal en todos los modos de operación**. También permanecen activas tras el fin de programa y pueden activarse directamente tras Power On con un ASUP. Gracias a ello, es posible activar acciones ejecutables en el CN independientemente del modo seleccionado. Margen de valores para <n>: 1 ... 255 Ejemplo: `IDS=1 EVERY $A_IN[1]==1 DO POS[X]=100`

10.1.2 Comprobación cíclica de la condición (WHEN, WHENEVER, FROM, EVERY)

Función

La comprobación cíclica de la condición de una acción síncrona se define mediante una palabra reservada. Si no hay programada ninguna palabra reservada, las acciones de la acción síncrona se ejecutan en cada ciclo IPO.

Palabras reservadas

Sin palabra reservada	La ejecución de la acción no depende de ninguna condición. La acción se ejecuta cíclicamente en la cadencia de interpolación.
`WHEN`	La condición se consulta en cada cadencia de interpolación hasta que se cumpla una vez; la acción correspondiente se ejecuta entonces exactamente una vez.
`WHENEVER`	La condición se comprueba cíclicamente en cada cadencia de interpolación. La acción correspondiente se ejecuta en cada cadencia de interpolación mientras se siga cumpliendo la condición.
`FROM`	La condición se consulta en cada cadencia de interpolación hasta que se cumpla una vez. La acción se ejecuta entonces mientras esté activa la instrucción síncrona, incluso aunque ya no se cumpla la condición.
`EVERY`	La condición se consulta en cada cadencia de interpolación. La acción se ejecuta siempre una vez al cumplirse la condición. Control por flanco: La acción se vuelve a ejecutar cuando la condición cambia del estado FALSE a TRUE.

Variables de proceso principal

Las variables utilizadas se evalúan en el ciclo de interpolación. Las variables de proceso principal en acciones síncronas no activan ninguna parada de decodificación previa.

Evaluación:

Si en un programa de pieza hay variables de proceso principal (p. ej., valor real, posición de una entrada o salida digital, etc.), se interrumpe el avance hasta que se haya ejecutado la secuencia anterior y se disponga de los valores correspondientes a las variables de proceso principal.

Ejemplos

Ejemplo 1: Sin palabra reservada

Código del programa	Comentarios
`DO $A_OUTA[1]=$AA_IN[X]`	`; Emisión de valor real a la salida analógica.`

Ejemplo 2: WHENEVER

Código del programa	Comentarios
`WHENEVER $AA_IM[X] > 10.5*SIN(45) DO …`	`; Comparación con la expresión calculada en el avance.`
`WHENEVER $AA_IM[X] > $AA_IM[X1] DO …`	`; Comparación con otras variables de proceso principal.`
`WHENEVER ($A_IN[1]==1) OR ($A_IN[3]==0) DO ...`	`; Dos comparaciones combinadas lógicamente entre sí.`

Ejemplo 3: EVERY

Código del programa	Comentarios
`ID=1 EVERY $AA_IM[B]>75 DO POS[U]=IC(10) FA[U]=900`	`; Siempre que el valor real del eje B en el MKS (sistema de coordenadas de máquina) sea mayor que 75, debe reposicionarse el eje U con un avance por eje de 10.`

Información adicional

Condición

La condición representa una expresión lógica que puede estar construida arbitrariamente a partir de operadores booleanos (lógicos). Las expresiones lógicas deben indicarse siempre entre corchetes.

La condición se comprueba en la cadencia de interpolación.

Antes de la condición se puede indicar un código G. De esta manera se consiguen, independientemente del estado de programa de pieza activado, unos ajustes definidos para la evaluación de la condición y la acción/el ciclo tecnológico a ejecutar. Se requiere desconectar el entorno del programa de las acciones síncronas porque éstas deben ejecutar en un momento cualquiera sus acciones en un estado inicial definido debido a las condiciones de activación cumplidas.

Casos de aplicación

Definición de sistemas de unidades para la evaluación de las condiciones y activación mediante los códigos G `G70, G71, G700, G710`.

Un código G indicado para la condición es válido para la evaluación de la condición y para la activación, cuando no se ha indicado un propio código G para la misma.

Por cada parte con condición sólo debe programarse **un código G** del grupo de códigos G.

Condiciones posibles

- Comparación de variables de proceso principal (entradas/salidas analógicas/digitales, etc.)
- Operación lógica de resultados de comparación
- Cálculo de expresiones de tiempo real
- Tiempo/distancia del inicio de secuencia
- Distancia del fin de secuencia
- Valores medidos, resultados de medición
- Servovalores
- Velocidades, estado de ejes

10.1.3 Acciones (DO)

Función

En las acciones síncronas se pueden programar una o varias acciones. Todas las acciones programadas en una secuencia se activan en la misma cadencia de interpolación.

Sintaxis

```
DO <Acción1> <Acción2> …
```

Descripción

`DO`	Inicia una acción o un ciclo tecnológico si se cumple la condición.
`<Acción>`	Acción que se activa al cumplirse la condición; p. ej. asignar variable, activar acoplamiento de ejes, definir salidas NCK, emitir funciones M, S y H, especificar código G programado...

Los códigos G se pueden programar en acciones síncronas para acciones/ciclos tecnológicos. Eventualmente, este código G predetermina para todas las acciones de la secuencia y los ciclos tecnológicos un código G diferente al que establece la condición. Si los ciclos tecnológicos se encuentran en la parte de acción, la modalidad del código G sigue siendo válida para todas las acciones posteriores hasta el próximo código G, incluso tras terminar el ciclo tecnológico.

Por cada parte de acción sólo debe programarse **un código G** del grupo de códigos G (`G70`, `G71`, `G700`, `G710`).

Ejemplo: Acción síncrona con dos acciones

Código del programa	**Comentarios**
`WHEN $AA_IM[Y]>=35.7 DO M135 $AC_PARAM=50`	`; Si se cumple la condición, se emite M135 al PLC y se ajusta a 50% la corrección del avance (override).`

10.2 Operadores para condiciones y acciones

Comparaciones (==, <>, <, >, <=, >=)	En las condiciones se pueden comparar variables o expresiones parciales. El resultado es siempre del tipo BOOL. Se admiten todos los operandos de comparación usuales.
Operadores lógicos (NOT, AND, OR, XOR)	Las variables, constantes o comparaciones se pueden combinar con los operadores lógicos conocidos.
Operadores bit a bit (B_NOT, B_AND, B_OR, B_XOR)	Son posibles los operadores bit a bit B_NOT, B_AND, B_OR, B_XOR.
Operandos aritméticos (+, -, *, /, DIV, MOD)	Las variables de proceso principal pueden combinarse entre sí o con constantes mediante los operandos aritméticos.
Funciones matemáticas (SIN, COS, TAN, ASIN, ACOS, ABS, TRUNC, ROUND, LN, EXP, ATAN2, POT, SQRT, CTAB, CTABINV).	En las variables de tipo REAL se pueden utilizar funciones matemáticas.
Indexación	La indexación es posible con expresiones de proceso principal.

Ejemplo

- **Operandos aritméticos combinados**

Las operaciones de multiplicación y división se ejecutan antes que las de adición y sustracción; las expresiones se pueden incluir entre paréntesis. Los operandos DIV y MOD también son lícitos para el tipo de datos REAL.

Programación	**Comentarios**
DO $AC_PARAM[3] = $A_INA[1]-$AA_IM[Z1]	; Sustracción de dos variables de proceso principal
WHENEVER $AA_IM[x2] < $AA_IM[x1]-1.9 DO $A_OUT[5] = 1	; Sustracción de una constante a variables
DO $AC_PARAM[3] = $INA[1]-4*SIN(45.7 $P_EP[Y])*R4	; Expresión constante, calculada en la decodificación previa

- **Funciones matemáticas**

Programación	**Comentarios**
DO $AC_PARAM[3] = COS($AC_PARAM[1])	;
	;

- **Expresiones de tiempo real**

Programación	**Comentarios**
ID=1 WHENEVER ($AA_IM[Y]>30) AND ($AA_IM[Y]<40) DO $AA_OVR[S1]=80	; Selección de una ventana de posicionamiento
ID=67 DO $A_OUT[1]=$A_IN[2] XOR $AN_META[1]	; Evaluar 2 señales lógicas
ID=89 DO $A_OUT[4]=$A_IN[1] OR ($AA_IM[Y]>10)	; Emitir el resultado de una comparación

- **Variable de proceso principal indexada**

Programación	**Comentarios**
WHEN…DO $AC_PARAM[$AC_META[1]] = 3	;
No es admisible	;
$AC_PARAM[1] = $P_EP[$AC_META]	;

10.3 Variables de proceso principal para acciones síncronas

10.3.1 Variables del sistema

Función

Con ayuda de las variables de sistema se pueden escribir y leer datos del CN. Las variables de sistema se distinguen en variables de decodificación previa y variables de proceso principal. Las variables de decodificación previa siempre se ejecutan en un momento previo. Las variables de proceso principal siempre calculan su valor en relación con el estado actual del proceso principal.

Denominación

El nombre de las variables de sistema comienza normalmente con el carácter $:

Variables de decodificación previa:

$M...	Datos de máquina
$S...	Datos de operador, zonas protegidas
$T...	Datos de gestión de herramientas
$P...	Valores programados, datos de decodificación previa
$C...	Variables de ciclos de envolvente ISO
$O...	Datos de opciones
R ...	Parámetros R

Variables de proceso principal:

$$A...	Datos actuales de proceso principal
$$V...	Datos servo
$R...	Parámetros R

Una 2ª letra describe las posibilidades de acceso a la variable:

N...	Valor NCK global (valor general)
C...	Valor específico del canal
A...	Valor específico del eje

La segunda letra suele utilizarse sólo para variables de proceso principal. Las variables de decodificación previa, como p. ej. $P_, se representan normalmente sin una 2ª letra.

Al prefijo ($ seguido de una o dos letras) le sigue siempre un carácter de subrayado y el subsiguiente nombre de variable (casi siempre una abreviatura o denominación en inglés).

Tipos de datos

Las variables de proceso principal pueden tener los siguientes tipos de datos:

INT	Integer para valores enteros con signo
REAL	Real para números racionales quebrados
BOOL	TRUE y FALSE lógicos
CHAR	Caracteres ASCII
STRING	String con caracteres alfanuméricos
AXIS	Direcciones de ejes y cabezales

Las variables de decodificación previa pueden tener adicionalmente los tipos de datos siguientes:

FRAME	Transformadas de coordenadas

Campos de variables

Las variables del sistema pueden crearse como matrices de 1 a 3 dimensiones.

Se admiten los siguientes tipos de datos: BOOL, CHAR, INT, REAL, STRING, AXIS

El tipo de datos del índice puede ser INT y AXIS, y éstos pueden ordenarse como se quiera.

Las variables STRING sólo pueden definirse con 2 dimensiones.

Ejemplos de definiciones de matrices:

```
DEF BOOL $AA_NEWVAR[x,y,2]
DEF CHAR $AC_NEWVAR[2,2,2]
DEF INT $AC_NEWVAR[2,10,3]
DEF REAL $AA_VECTOR[x,y,z]
DEF STRING $AC_NEWSTRING[3,3]
DEF AXIS $AA_NEWAX[x,3,y]
```

Nota

La visualización de variables del sistema de 3 dimensiones no tiene limitación, siempre que haya una variable BTSS para las variables del sistema.

10.3.2 Conversión implícita de tipos

Función

En las asignaciones de valores y las transferencias de parámetros pueden asignarse o transferirse variables de tipos de datos diferentes.

La conversión implícita de tipos activa una conversión interna de tipos de valores.

Posibles conversiones de tipos

a	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
de							
REAL	sí	sí*	sí[1])	–	–	–	–
INT	sí	sí	sí[1])	–	–	–	–
BOOL	sí	sí	sí	–	–	–	–

Aclaraciones

* En la conversión de tipos de REAL a INT se redondea hacia arriba con una fracción de >= 0,5; de lo contrario, se redondea hacia abajo (ver función ROUND).
Si se rebasan los valores, se emite una alarma.

1) El valor <> 0 equivale a TRUE, el valor == 0 equivale a FALSE

Resultados

```
Conversión de tipos de REAL o INTEGER a BOOL
Resultado BOOL = TRUE          si el valor de REAL o INTEGER es distinto de cero
Resultado BOOL = FALSE         si el valor de REAL o INTEGER es igual a cero
Conversión de tipos de BOOL a REAL o INTEGER
Resultado REAL = TRUE          si el valor de BOOL = TRUE es (1)
Resultado INTEGER = TRUE       si el valor de BOOL = TRUE es (1)
Conversión de tipos de BOOL a REAL o INTEGER
Resultado REAL = FALSE         si el valor de BOOL = FALSE es (0)
Resultado INTEGER = FALSE      si el valor de BOOL = FALSE es (0)
```

Ejemplos de conversiones implícitas de tipos

```
Conversión de tipos de INTEGER a BOOL
$AC_META[1]=561
ID=1 WHEN $A_IN[1] == TRUE DO $A_OUT[0]=$AC_META[1]

Conversión de tipos de REAL a BOOL
R401 = 100.542
WHEN $A_IN[0] == TRUE DO $A_OUT[2]=$R401

Conversión de tipos de BOOL a INTEGER
ID=1 WHEN $A_IN[2] == TRUE DO $AC_META[4] = $A_OUT[1]]

Conversión de tipos de BOOL a REAL
R401 = 100.542
WHEN $A_IN[3] == TRUE DO $R10 = $A_OUT[3]
```

10.3.3 Valores de las variables GUD

Variables GUD compatibles con acciones síncronas

Además de variables de sistema específicas, en las acciones síncronas pueden utilizarse variables de usuario de acción síncrona globales predefinidas (GUD de acción síncrona). El número de GUD de acción síncrona disponibles para el usuario se parametriza específicamente para cada tipo de datos y de acceso mediante los siguientes datos de máquina:

- DM18660 $MM_NUM_SYNACT_GUD_**REAL**[<x>] = <Cantidad>
- DM18661 $MM_NUM_SYNACT_GUD_**INT**[<x>] = <Cantidad>
- DM18662 $MM_NUM_SYNACT_GUD_**BOOL**[<x>] = <Cantidad>
- DM18663 $MM_NUM_SYNACT_GUD_**AXIS**[<x>] = <Cantidad>
- DM18664 $MM_NUM_SYNACT_GUD_**CHAR**[<x>] = <Cantidad>
- DM18665 $MM_NUM_SYNACT_GUD_**STRING**[<x>] = <Cantidad>

A través del índice <x> se indica el bloque de datos (derechos de acceso), y a través del valor <Cantidad>, el número de GUD de acción síncrona del tipo de datos correspondiente (REAL, INT...). Además, en el respectivo bloque de datos se crea para cada tipo de datos una variable de campo unidimensional cuyo nombre presenta la siguiente estructura: SYG_<Tipo de datos><Derecho de acceso>[<Índice>]:

Índice <x>		Tipo de datos (DM18660 ... DM18665)					
	Bloque	**REAL**	**INT**	**BOOL**	**AXIS**	**CHAR**	**STRING**
0	SGUD	SYG_RS[i]	SYG_IS[i]	SYG_BS[i]	SYG_AS[i]	SYG_CS[i]	SYG_SS[i]
1	MGUD	SYG_RM[i]	SYG_IM[i]	SYG_BM[i]	SYG_AM[i]	SYG_CM[i]	SYG_SM[i]
2	UGUD	SYG_RU[i]	SYG_IU[i]	SYG_BU[i]	SYG_AU[i]	SYG_CU[i]	SYG_SU[i]
3	GUD4	SYG_R4[i]	SYG_I4[i]	SYG_B4[i]	SYG_A4[i]	SYG_C4[i]	SYG_S4[i]
4	GUD5	SYG_R5[i]	SYG_I5[i]	SYG_B5[i]	SYG_A5[i]	SYG_C5[i]	SYG_S5[i]
5	GUD6	SYG_R6[i]	SYG_I6[i]	SYG_B6[i]	SYG_A6[i]	SYG_C6[i]	SYG_S6[i]
6	GUD7	SYG_R7[i]	SYG_I7[i]	SYG_B7[i]	SYG_A7[i]	SYG_C7[i]	SYG_S7[i]
7	GUD8	SYG_R8[i]	SYG_I8[i]	SYG_B8[i]	SYG_A8[i]	SYG_C8[i]	SYG_S8[i]
8	GUD9	SYG_R9[i]	SYG_I9[i]	SYG_B9[i]	SYG_A9[i]	SYG_C9[i]	SYG_S9[i]
Con i = 0 hasta (<Cantidad> - 1) Bloque: _N_DEF_DIR/_N_ ... _DEF, p. ej. para SGUD ⇒ _N_DEF_DIR/_N_**SGUD**_DEF							

Características

Las GUD de acción síncrona tienen las siguientes características:

- Las GUD de acción síncrona pueden leerse y escribirse en acciones síncronas y programas de pieza/ciclos.
- Se puede acceder a GUD de acción síncrona vía BTSS.
- Las GUD de acción síncrona se visualizan en la interfaz de usuario de HMI, en el campo de manejo "Parámetros".
- Las GUD de acción síncrona pueden utilizarse en HMI en el asistente, en la vista de variables y en el protocolo de variables.
- El tamaño de la matriz de las GUD de acción síncrona del tipo STRING está fijado en 32 (31 caracteres + \0).
- Incluso aunque no se hayan creado manualmente ficheros de definición para datos de usuario globales (GUD), las GUD de acción síncrona definidas pueden leerse por medio de datos de máquina en el bloque GUD correspondiente desde HMI.

ATENCIÓN

Sólo pueden definirse variables de usuario (GUD, PUD, LUD) con el mismo nombre que las GUD de acción síncrona (`DEF ... SYG_xy`) si no se han parametrizado GUD de acción síncrona con el mismo nombre (DM18660-DM18665). Estas GUD definidas por el usuario **no** pueden utilizarse en acciones síncronas.

Derechos de acceso

Los derechos de acceso definidos en un fichero de definición GUD siguen siendo válidos y se refieren únicamente a las variables GUD definidas en dicho fichero.

Comportamiento de borrado

Si vuelve a activarse el contenido de un determinado fichero de definición GUD, se borra primero el antiguo bloque de datos GUD del sistema de ficheros activo. Durante esta operación también se resetean las GUD de acción síncrona configuradas. Esta operación también puede realizarse a través del HMI en el campo de manejo "Servicios" > "Definir y activar datos de usuario (GUD)".

10.3.4 Identificador de eje por defecto (NO_AXIS)

Función

Las variables o parámetros del tipo AXIS que no se hayan inicializado con un valor, se pueden definir con identificadores de eje por defecto. Las variables de eje no definidas se inicializan con este valor por defecto.

Los nombres de eje válidos no inicializados pueden reconocerse en acciones síncronas mediante una consulta de la variable "NO_AXIS". A los identificadores de eje no inicializados se les asigna el identificador de eje por defecto configurado mediante un dato de máquina.

Fabricante de la máquina

Al menos un identificador de eje válido y disponible debe estar definido y preasignado mediante datos de máquina. También se pueden preasignar todos los identificadores de eje válidos disponibles. Preste atención a las indicaciones del fabricante de la máquina.

Nota

Durante la definición, se asigna automáticamente a las nuevas variables el valor consignado en el dato de máquina para nombres de eje por defecto.

Para más información sobre una definición válida mediante un dato de máquina, ver:
Bibliografía:
/FBSY/ Manual de funciones Acciones síncronas

Sintaxis

```
PROC UP(AXIS PAR1=NO_AXIS, AXIS PAR2=NO_AXIS)
IF PAR1 <>NO_AXIS...
```

Descripción

`PROC`	Definición de subprograma
`UP`	Nombre de subprograma para reconocimiento
`PARn`	Parámetro n
`NO_AXIS`	Inicialización del parámetro formal con un identificador de eje por defecto

Ejemplo: Definición de una variable de eje en el programa principal

```
Código del programa
DEF AXIS AXVAR
UP( , AXVAR)
```

10.3.5 Meta para acciones síncronas ($AC_META[n])

Función

La variable de matriz $AC_MARKER[n] puede leerse y escribirse en las acciones síncronas. Estas variables pueden encontrarse en la memoria del sistema de ficheros activo o pasivo.

Variable de acción síncrona: Tipo de datos INT

$AC_MARKER[n]	Marcas/contadores específicos del canal del tipo de datos INTEGER
$MC_MM_NUM_AC_MARKER	Dato de máquina para ajustar el número de marcas específicas de canal para acciones síncronas de desplazamiento
n	Índice de matriz de las variables 0-n

Ejemplo de lectura y escritura de una variable de marca

```
Código del programa
WHEN ... DO $AC_MARKER[0] = 2
WHEN ... DO $AC_MARKER[0] = 3
WHENEVER $AC_MARKER[0] == 3 DO $AC_OVR=50
```

10.3.6 Parámetro de acciones síncronas ($AC_PARAM[n])

Función

Los parámetros de acciones síncronas $AC_PARAM[n] se utilizan para efectuar cálculos y como memoria intermedia en las acciones síncronas. Estas variables pueden encontrarse en la memoria del sistema de ficheros activo o pasivo.

Variable de sincronización: Tipo de datos REAL

Los parámetros están disponibles con el mismo nombre una vez por cada canal.

`$AC_PARAM[n]`	Variable de cálculo para las acciones síncronas de desplazamiento (REAL)
`$MC_MM_NUM_AC_PARAM`	Dato de máquina para ajustar el número de parámetros para las acciones síncronas de desplazamiento hasta un máximo de 20.000.
`n`	Índice de matriz del parámetro 0n

Ejemplos de parámetros para acciones síncronas $AC_PARAM[n]

Código del programa

```
$AC_PARAM[0]=1.5
$AC_META[0]=1
ID=1 WHEN $AA_IW[X]>100 DO $AC_PARAM[1]=$AA_IW[X]
ID=2 WHEN $AA_IW[X]>100 DO $AC_MARKER[1]=$AC_MARKER[2]
```

10.3.7 Parámetro de cálculo ($R[n])

Función

Esta variable de matriz estática sirve para realizar cálculos en el programa de pieza y en las acciones síncronas.

Sintaxis

Programación en el programa de pieza:
```
REAL R[n]
REAL Rn
```
Programación en las acciones síncronas:
```
REAL $R[n]
REAL $Rn
```

Parámetro de cálculo

La utilización de parámetros de cálculo permite:

- Memorizar valores que se desean conservar tras el fin del programa, reposición del CN y Power On
- Indicar valores memorizados en la imagen de parámetros R.

Ejemplos

Código del programa	Comentarios
WHEN $AA_IM[X]>=40.5 DO $R10=$AA_MM[Y]	; Utilización de R10 en acción síncrona.
G01 X500 Y70 F1000	
STOPRE	; Parada de decodificación previa
IF R10>20	; Evaluación de las variables de cálculo.

Código del programa
SYG_AS[2]=X
SYG_IS[1]=1
WHEN $AA_IM[SGY_AS[2]]>10 DO $R3=$AA_EG_DENOM[SYG_AS[1]], SYG_AS[2]]
WHEN $AA_IM[SGY_AS[2]]>12 DO $AA_SCTRACE[SYG_AS[2]]=1
SYG_AS[1]=X
SYG_IS[0]=1
WHEN $AA_IM[SGY_AS[1]]>10 DO $R3=$$MA_POSCTRL_GAIN[SYG_IS[0]],SYG_AS[1]]
WHEN $AA_IM[SGY_AS[1]]>10 DO $R3=$$MA_POSCTRL_GAIN[SYG_AS[1]]
WHEN $AA_IM[SGY_AS[1]]>15 DO $$MA_POSCTRL_GAIN[SYG_AS[0]], SYG_AS[1]]=$R3

10.3.8 Leer y escribir datos de máquina y de operador CN

Función

También es posible leer y escribir datos de máquina y de operador CN desde las acciones síncronas. Al leer y escribir elementos de matriz de datos de máquina puede omitirse un índice durante la programación. Si esto ocurre en el programa de pieza, al realizar la lectura se lee el **primer** elemento de matriz y, al escribir, se describen con el valor todos los elementos de la matriz.

En este caso, en las acciones síncronas sólo se lee o escribe el **primer** elemento.

Definición

DM, DO con

`$:` lectura del valor en el momento de la interpretación de las acciones síncronas

`$$:` lectura del valor en proceso principal

Lectura de los valores DM y DO en preprocesamiento

Estos datos se direccionan desde la acción síncrona con el carácter $ y se evalúan en preprocesamiento.

```
ID=2 WHENEVER $AA_IM[z]<$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
;Aquí se activa la zona de inversión 2 para vaivén, supuesto como invariable.
```

Lectura de los valores DM y DO en proceso principal

Estos datos se direccionan desde la acción síncrona con los caracteres $$ y se evalúan en proceso principal.

```
ID=1 WHENEVER $AA_IM[z]<$$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
;Aquí se parte del supuesto de que la posición de inversión se podría modificar
mediante una intervención durante la ejecución.
```

Escritura de los valores DM y DO en proceso principal

El derecho de acceso actualmente ajustado debe permitir el acceso a escritura. Para mayor información sobre el efecto en todos los DM y DO se puede consultar la **bibliografía:** /LIS/, Listas (libro 1).

Los `DM` y los `DO` que deben escribirse se direccionan anteponiendo `$$`.

Ejemplo

Código del programa	Comentarios
ID=1 WHEN $AA_IW[X]>10 DO $$SN_SW_CAM_PLUS_POS_TAB_1[0]=20 $$SN_SW_CAM_MINUS_POS_TAB_1[0]=30	; Modificación de la posición de conmutación de levas de software. Nota: Las posiciones de conmutación se han de modificar 2 a 3 cadencias IPO antes de alcanzar la posición correspondiente.

10.3.9 Variables de temporizador ($AC_Timer[n])

Función

La variable de sistema $AC_TIMER[n] permite arrancar las acciones al transcurrir los tiempos de espera definidos.

Variable de tipo temporizador: Tipo de datos REAL

$AC_TIMER[n]	Temporizador específico de canal del tipo de datos REAL
s	Unidad en segundos
n	Índice de la variable tipo temporizador

Activar temporizador

El contaje de una variable de tipo temporizador se inicia con la asignación del valor:
$AC_TIMER[n] = value

n:	Número de las variables de tiempo
value:	Valor inicial (normalmente "0")

Parar temporizador

El contaje de una variable de tipo temporizador se detiene mediante la asignación de un valor negativo:
$AC_TIMER[n]=-1

Leer temporizador

El valor de tiempo actual se puede leer estando las variables de temporizador en marcha o parado. Tras el paro de las variables de temporizador mediante la asignación de -1, se conserva y se puede continuar leyendo el último valor de tiempo actual.

Ejemplo

Emisión de un valor real mediante salida analógica 500 ms después de reconocer una entrada digital:

```
Código del programa                                                    Comentarios
WHEN $A_IN[1]==1 DO $AC_TIMER[1]=0                                     ; Resetear y arrancar
WHEN $AC_TIMER[1]>=0.5 DO $A_OUTA[3]=$AA_IM[X] $AC_TIMER[1]=-1           temporizador
```

10.3.10 Variables FIFO ($AC_FIFO1[n] ... $AC_FIFO10[n])

Función

Para memorizar sucesiones de datos asociados se dispone de 10 variables FIFO (memoria cíclica).
Tipo de datos: REAL

Aplicación:

- Medición cíclica
- Mecanizado continuo

Se tiene acceso de lectura y de escritura para cada elemento.

Variable FIFO

La cantidad de variables FIFO disponibles se especifica a través del dato de máquina DM28260 $MC_NUM_AC_FIFO.

El número de valores registrables en una variable FIFO queda definido por el dato de máquina DM28264 $MC_LEN_AC_FIFO. Todas las variables FIFO tienen la misma longitud.

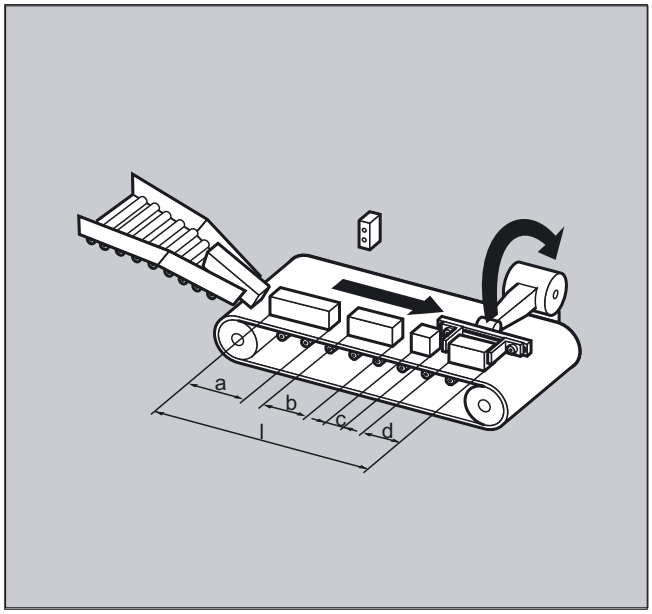
La suma de todos los elementos FIFO sólo se forma si está definido el bit 0 en DM28266 $MC_MODE_AC_FIFO.

Los índices **0 a 5** tienen significado especial:

Index	Descripción	
0	Al escribir:	El nuevo valor se guarda en FIFO.
	Al leer:	El elemento más antiguo se lee y se elimina de FIFO.
1	Acceso al elemento más antiguo guardado	
2	Acceso al elemento más reciente guardado	
3	Suma de todos los elementos FIFO	
4	Número de elementos disponibles en FIFO Se tiene acceso de lectura y de escritura a todos los elementos FIFO. Las variables FIFO se anulan poniendo a cero el número de elementos, p. ej., para la primera variable FIFO: $AC_FIFO1[4]=0	
5	Índice de escritura actual relativo al inicio de FIFO	
6 hasta $n_{máx}$	Acceso al enésimo elemento FIFO	

Ejemplo: Memoria cíclica

En un proceso productivo se utiliza una cinta transportadora para transportar productos de diferente longitud (a, b, c, d). Por consiguiente, en la cinta transportadora con la longitud de transporte se transportan simultáneamente, dependiendo de las respectivas longitudes, diferentes cantidades de productos. Para una velocidad de transporte constante, hay que adaptar la toma de productos de la cinta a los tiempos variables de llegada de los productos.



Código del programa	**Comentarios**
`DEF REAL ZWI=2.5`	`; Distancia constante entre los productos depositados en la cinta`
`DEF REAL TOTAL=270`	`; Distancia entre la posición de medida de longitud y la posición de toma`
`EVERY $A_IN[1]==1 DO $AC_FIFO[4]=0`	`; Al inicio del proceso, resetear FIFO.`
`EVERY $A_IN[2]==1 DO $AC_TIMER[0]=0`	`; Si un producto interrumpe la barrera fotoeléctrica, iniciar la medición.`
`EVERY $A_IN[2]==0 DO $AC_FIFO[0]=$AC_TIMER[0]*$AA_VACTM[B]`	`; Cuando se libera la barrera foto-eléctrica, debe calcularse la longitud del producto a partir del tiempo medido y la velocidad de transporte, y guardarse en FIFO.`
`EVERY $AC_FIFO1[3]+$AC_FIFO1[4]*ZWI>=GESAMT DO POS[Y]=-30 $R1=$AC_FIFO1[0]`	`; En cuanto la suma de todas las longitudes de producto y distan-cias entre productos sea mayor o igual a la longitud entre la posi-ción de carga y descarga, retirar el producto en la posición de des-carga de la cinta transportadora y leer su correspondiente longitud de FIFO.`

10.3.11 Información acerca de tipos de secuencia en el interpolador ($AC_BLOCKTYPE, $AC_BLOCKTYPEINFO, $AC_SPLITBLOCK)

Función

Para acciones síncronas se dispone de las siguientes variables de sistema para obtener información sobre una secuencia actual en la marcha principal:

- $AC_BLOCKTYPE
- $AC_BLOCKTYPEINFO
- $AC_SPLITBLOCK

Variables Blocktype y Blocktypeinfo

$AC_BLOCKTYPE		$AC_BLOCKTYPEINFO				
Valor:		Valor:				
0	no igual a 0	T	H	Z	E	Significado:
Secuencia original	Secuencia intermedia					Actuador para la secuencia intermedia
	1	1	0	0	0	Secuencia generada a nivel interno; sin más información
	2	2	0	0	1	Achaflanar/redondear: Línea recta
	2	2	0	0	2	Achaflanar/redondear: Circunferencia
	3	3	0	0	1	WAB: Arranque con recta
	3	3	0	0	2	WAB: Arranque con cuadrante
	3	3	0	0	3	WAB: Arranque con semicírculo
						Corrección de herramientas:
	4	4	0	0	1	Secuencia de posicionamiento después de STOPRE
	4	4	0	0	2	Secuencias de conexión si no se encuentra el punto de intersección
	4	4	0	0	3	Círculo en forma de punto en esquinas interiores (sólo con TRACYL)
	4	4	0	0	4	Circunferencia de rodeo (p. ej., sección cónica) en esquinas exteriores
	4	4	0	0	5	Secuencias de posicionamiento en la supresión de correcciones
	4	4	0	0	6	Secuencias de posicionamiento con nueva activación de corrección de radio de herramienta
	4	4	0	0	7	División de la secuencia debido a una curvatura excesiva
	4	4	0	0	8	Secuencias de compensación en el fresado frontal 3D (vector de herramienta || vector de superficie)

$AC_BLOCKTYPE		$AC_BLOCKTYPEINFO				
Valor:		Valor:				
0	no igual a 0	T	H	Z	E	Significado:
Secuencia original	Secuencia intermedia					Actuador para la secuencia intermedia
						Matado de esquinas con:
	5	5	0	0	1	G641
	5	5	0	0	2	G642
	5	5	0	0	3	G643
	5	5	0	0	4	G644
						Secuencia TLIFT con:
	6	6	0	0	1	Movimiento lineal del eje tangencial y sin movimiento de retirada
	6	6	0	0	2	Movimiento no lineal del eje tangencial (polinomio) y sin movimiento de retirada
	6	6	0	0	3	Movimiento de retirada, movimiento del eje tangencial y movimiento de retirada se inician simultáneamente
	6	6	0	0	4	Movimiento de retirada, eje tangencial sólo se inicia al alcanzar una determinada posición de retirada
						División de la trayectoria:
	7	7	0	0	1	División de la trayectoria programada sin troquelado/punzonado activo
	7	7	0	0	2	División de la trayectoria programada con troquelado/punzonado activo
	7	7	0	0	3	División de la trayectoria generada automáticamente a nivel interno
						Ciclos de compilación:
	8	ID aplicación				ID de la aplicación de ciclos de compilación que ha generado la secuencia

T: Millares
H: Centenas
Z: Decenas
E: unidades

Nota

$AC_BLOCKTYPEINFO también contiene siempre en los millares (M) el valor del tipo de bloque para el caso de que exista una secuencia intermedia. En $AC_BLOCKTYPE distinto a 0 no se adoptan los millares.

$AC_SPLITBLOCK	
Valor:	Significado:
0	Secuencia programada inalterada (una secuencia generada por el compresor se trata igualmente como secuencia programada)
1	Existe una secuencia generada a nivel interno o una secuencia original acortada
3	Existe la última secuencia en una cadena de secuencias generadas a nivel interno o secuencias originales acortadas

Ejemplo: Contaje de secuencias de matado de esquinas

```
Código del programa      Comentarios
$AC_META[0]=0
$AC_META[1]=0
$AC_META[2]=0
...
                         ; Definición de acciones síncronas con las cuales
                           se cuentan secuencias de matado de esquinas.
                         ; Todas las secuencias de matado de esquinas cuentan en $AC_META[0]:
ID=1 WHENEVER ($AC_TIMEC==0) AND ($AC_BLOCKTYPE==5) DO $AC_META[0]=$AC_META[0]+1
...
                         ; Las secuencias de matado de esquinas generadas con G641 cuentan en
                           $AC_META[1]:
ID=2 WHENEVER ($AC_TIMEC==0) AND ($AC_BLOCKTYPEINFO==5001) DO $AC_META[1]=$AC_META[1]+1
                         ; Las secuencias de matado de esquinas generadas con G642 cuentan en
                           $AC_META[2]:
ID=3 WHENEVER ($AC_TIMEC==0) AND ($AC_BLOCKTYPEINFO==5002) DO $AC_META[2]=$AC_META[2]+1
...
```

10.4 Acciones en instrucciones síncronas

10.4.1 Lista de las posibles acciones en acciones síncronas

Las acciones de las acciones síncronas se componen de asignación de valores, llamadas de función o parámetro, palabras reservadas o ciclos tecnológicos. Es posible ejecutar acciones complejas mediante operadores.

Posibles aplicaciones son:

- Cálculo de expresiones complejas en ciclo IPO
- Desplazamientos de eje y mandos del cabezal
- Modificar y evaluar Online los datos del operador de las acciones síncronas (p. ej. emitir posiciones y tiempos de levas de seguridad al PLC o a la periferia CN)
- Emisión de funciones auxiliares en el PLC
- Configurar funciones de seguridad adicionales
- Configurar movimiento superpuesto, corrección de herramienta Online y regulación de distancia
- Ejecución de acciones en todos los modos de operación
- Influencia en acciones síncronas desde el PLC
- Ejecución de ciclos tecnológicos
- Emisión de señales analógicas y digitales
- Registro del rendimiento de las acciones síncronas en la cadencia de interpolación y el tiempo de cálculo del regulador de posición para una evaluación de la ocupación
- Posibilidades de diagnóstico en la interfaz hombre-máquina

Acción síncrona	Descripción
DO $V…= DO $A...=	Asignar (valores servo) Asignar variable (variable de proceso principal)
DO $AC…[n]= DO $AC_MARKER[n]= DO $AC_PARAM[n]=	Variable de proceso principal especial Leer o escribir meta para acciones síncronas Leer o escribir parámetro para acciones síncronas
DO $R[n]=	Leer o escribir la variable de cálculo
DO $DM...= DO $$DO...=	Lectura del valor del DM en el momento de interpolación Escritura del valor del DO en el proceso principal
DO $AC_TIMER[n]=valor inicial	Timer (temporizador)
DO $AC_FIFO1[n] …FIFO10[n]=	Variable FIFO
DO $AC_BLOCKTYPE= DO $AC_BLOCKTYPEINFO= DO $AC_SPLITBLOCK=	Interpretar la secuencia actual (variables de proceso principal)
DO M, S y H (p. ej.: M07)	Emisión de funciones auxiliares M, S y H
DO RDISABLE	Activar bloqueo de lectura

Acción síncrona	Descripción
DO STOPREOF	Suprimir parada de decodificación previa
DO DELDTG	Borrado de trayecto residual rápido sin parar decodificación previa
FTCDEF(polin., LL, UL, coefic.) DO SYNFCT(polin., Output, Input)	Definición de polinomios Activación de funciones síncronas: regulación AC
DO FTOC	Corrector de herramienta Online
DO G70/G71/G700/G710	Especificar sistema de medida para tareas de posicionamiento (cotas en pulgadas o métrico)
DO POS[Eje]= / DO MOV[Eje]= DO SPOS[Cabezal]=	Arrancar/posicionar/parar ejes de comando Arrancar/posicionar/parar cabezales
DO MOV[eje]=valor	Arrancar/parar desplazamientos continuos de un eje de comando
DO POS[eje]= FA [eje]=	Avance por eje FA
ID=1 ... DO POS[Eje]= FA [Eje]= ID=2 ... DO POS[Eje]= $AA_IM[Eje] FA [Eje]=	Posicionamiento desde acciones síncronas
DO PRESETON (eje, valor)	Poner valor real (PRESET desde acciones síncronas)
ID=1 EVERY $A_IN[1]=1 DO M3 S… ID=2 EVERY $A_IN[2]=1 DO SPOS=	Arrancar/posicionar/parar cabezales
DO TRAILON(FA,LA,Factor de acoplamiento) DO LEADON(FA,LA,NRCTAB,OVW)	Activar arrastre Activar acoplamiento de valores maestros
DO MEAWA(Eje)= DO MEAC(Eje)=	Activar medición axial Activar medición continua
DO [Matriz n, m]=SET(Valor, Valor, ...) DO [Matriz n, m]=REP(Valor, Valor, ...)	Inicialización de variables de matriz con listas de valores Inicialización de variables de matriz con los mismos valores
DO SETM(número de meta) DO CLEARM(número de meta)	Activar metas de espera Borrar metas de espera
DO SETAL(nº alarma)	Activar alarma de ciclo (función de seguridad adicional)
DO FXS[Eje]= DO FXST[Eje]= DO FXSW[Eje]= DO FOCON[Eje]= DO FOCOF[Eje]=	Seleccionar desplazamiento a tope fijo Modificar par de apriete Modificar ventana de vigilancia Activar desplazamiento con par/fuerza limitados (modal) FOC Desactivar desplazamiento con par/fuerza limitados (acción síncrona referida a la secuencia)
ID=2 EVERY $AC_BLOCKTYPE==0 DO $R1=$AC_TANEB	Ángulo entre la tangente de trayectoria en el punto final de la secuencia actual y la tangente de trayectoria en el punto inicial de la siguiente secuencia programada

Acción síncrona	Descripción
DO $AA_OVR=	Corrección por eje
DO $AC_OVR=	Corrección del avance
DO $AA_PLC_OVR	La corrección por eje especificada por el PLC
DO $AC_PLC_OVR	La corrección del avance especificada por el PLC
DO $AA_TOTAL_OVR	La corrección por eje resultante
DO $AC_TOTAL_OVR	La corrección del avance resultante
$AN_IPO_ACT_LOAD=	Tiempo de cálculo IPO actual
$AN_IPO_MAX_LOAD=	Tiempo de cálculo IPO más largo
$AN_IPO_MIN_LOAD=	Tiempo de cálculo IPO más corto
$AN_IPO_LOAD_PERCENT=	Tiempo de cálculo IPO actual en relación con el ciclo IPO
$AN_SYNC_ACT_LOAD=	Tiempo de cálculo actual para acción síncrona a través de todos los canales
$AN_SYNC_MAX_LOAD=	Tiempo de cálculo más largo para acción síncrona a través de todos los canales
$AN_SYNC_TO_IPO=	Porcentaje del total de la acción síncrona
DO TECCYCLE	Ejecución del ciclo tecnológico
DO LOCK(n, n, ...)	Bloquear
DO UNLOCK(n, n, ...)	Desconectar y aislar de alimentación
DO RESET(n, n, ...)	RESET de un ciclo tecnológico
CANCEL(n, n, ...)	Borrar acciones síncronas modales con la denominación ID(S) en el programa de pieza

10.4.2 Emisión de funciones auxiliares

Función

Momento de emisión

La emisión de las funciones auxiliares tiene lugar en la acción síncrona justo en el momento de emisión de la acción. El momento de la emisión definido a través del dato de máquina no está activo.

El momento de emisión de indica cuando se cumple el requisito.

Ejemplo:

Activar refrigeración (taladrina) en una determinada posición del eje:

```
WHEN $AA_IM[X]>=15 DO M07 POS[X]=20 FA[X]=250
```

Palabras reservadas permitidas en acciones síncronas activas por secuencia (sin ID modal)

Las funciones auxiliares en acciones síncronas activas por secuencia (sin ID modal) sólo pueden programarse con las palabras reservadas `WHEN` o `EVERY`.

Nota

Las siguientes funciones auxiliares no están permitidas desde una acción síncrona:

- `M0`, `M1`, `M2`, `M17`, `M30`: parada/fin del programa (`M2`, `M17` y `M30` posibles en ciclo tecnológico)
- `M6` o funciones M para el cambio de herramienta ajustadas mediante datos de máquina

Ejemplo

Código del programa	Comentarios
`WHEN $AA_IW[Q1]>5 DO M172 H510`	`; Si el valor real del eje Q1 sobrepasa los 5 mm, deben emitirse las funciones auxiliares M172 y H510 al PLC.`

10.4.3 Activar bloqueo de lectura (RDISABLE)

Función

Si se cumple la condición, con RDISABLE se detiene la ejecución de la secuencia en el programa principal. Las acciones síncronas al desplazamiento programadas se siguen ejecutando y se preparan las secuencias subsiguientes.

En el modo de contorneado se genera siempre una parada exacta al inicio de la secuencia con RDISABLE, independientemente de que RDISABLE esté activo o no.

Ejemplo

Arrancar el programa en la cadencia de interpolación en función de las entradas externas.

Código del programa	Comentarios
`...`	
`WHENEVER $A_INA[2]<7000 DO RDISABLE`	`; Si la tensión en la entrada 2 es menor de 7 V, parar entonces la ejecución del programa (1000 = 1 V).`
`N10 G1 X10`	`; Si la condición se cumple, el bloqueo de lectura actúa al final de N10`
`N20 G1 X10 Y20`	
`...`	

10.4.4 Desactivar la parada de decodificación previa (STOPREOF)

Función

En caso de parada de decodificación previa STOPRE programada explícitamente, o activada implícitamente por una acción síncrona activa, STOPREOF desactiva esta parada tras la próxima secuencia de mecanizado, tan pronto se haya cumplido la condición.

Nota

STOPREOF se tiene que programar secuencia a secuencia (sin número ID) y con la palabra reservada `WHEN` .

Ejemplo

Bifurcación rápida de programa al fin de secuencia.

Código del programa	**Comentarios**
`WHEN $AC_DTEB<5 DO STOPREOF`	`; Si la distancia respecto al fin de secuencia es menor a 5 mm, debe suprimirse la parada de decodificación previa.`
`G01 X100`	`; Tras la ejecución de la interpolación lineal, se suprime la parada de decodificación previa.`
`IF $A_INA[7]>500 GOTOF META1=X100`	`; Si se supera la tensión de 5 V en la entrada 7, saltar a la etiqueta 1.`

10.4.5 Borrado de trayecto residual (DELDTG)

Función

Según la condición programada es posible activar el borrado de trayecto residual para la trayectoria y para los ejes indicados.

Es posible efectuar:

- Borrado rápido, previamente preparado, de trayecto residual
- Borrado de trayecto residual sin preparación

El borrado rápido, preparado, de trayecto residual con DELDTG permite reaccionar rápidamente a un suceso de activación y, por consiguiente, se utiliza en aplicaciones de tiempo crítico, p. ej., cuando:

- El tiempo entre el borrado de trayecto residual y el arranque de la secuencia siguiente debe ser muy corto.
- Es muy probable que se cumpla la condición requerida para el borrado de trayecto residual.

Nota

La designación de eje que aparece entre paréntesis detrás de DELDTG sólo es válida para un eje de posicionamiento.

Sintaxis

Borrado de trayecto residual para la trayectoria
```
DO DELDTG
```
Borrado de trayecto residual por eje
```
DO DELDTG(eje1) DELDTG(eje2) ...
```

Ejemplo Borrado de trayecto residual rápido trayectoria

Código del programa	**Comentarios**
`WHEN $A_IN[1]==1 DO DELDTG`	
`N100 G01 X100 Y100 F1000`	`; Si está activa la entrada, se interrumpe el desplazamiento`
`N110 G01 X…`	
`IF $AA_DELT>50…`	

Ejemplo Borrado de trayecto residual rápido por eje

```
Código del programa                                  Comentarios
Cancelación de un desplazamiento de
posicionado:
ID=1 WHEN $A_IN[1]==1 DO MOV[V]=3 FA[V]=700      ;  Arrancar eje
WHEN $A_IN[2]==1 DO DELDTG(V)                    ;  Borrado de trayecto residual,
                                                    parar eje con MOV=0
Borrar trayecto residual en función de la
tensión de entrada:
WHEN $A_INA[5]>8000 DO DELDTG(X1)                ;  En cuanto se sobrepase una
                                                    tensión de 8 V en la entrada 5,
                                                    borrar trayecto residual del eje
                                                    X1. El movimiento interpolado
                                                    continúa.
POS[X1]=100 FA[X1]=10 G1 Z100 F1000
```

Información adicional

Finalizada la secuencia de desplazamiento en la cual fue activado el borrado de trayecto residual preparado, se activa implícitamente una parada de decodificación previa.

Así, el contorneado o bien los desplazamientos de ejes de posicionado se pueden interrumpir o parar al final de las secuencias con borrado rápido de trayecto residual.

Nota

Borrado preparado de trayecto residual:

- No se puede utilizar cuando está activa la corrección de radio de herramienta.
- Sólo puede ser programado en acciones síncronas activas secuencia a secuencia (sin número ID).

10.4.6 Definición de polinomios (FCTDEF)

Función

Con FCTDEF se pueden definir polinomios de 3er grado en la forma $y=a_0+a_1x+a_2x^2+a_3x^3$. Estos polinomios son utilizados por la corrección de herramienta online FTOC y la función de evaluación SYNFCT.

Sintaxis

```
FCTDEF(Número polinomio,LLIMIT,ULIMIT,a0,a1,a2,a3)
```

Descripción

`Número polinomio`	Número del polinomio de tercer grado
`LLIMIT`	Límite inferior para el valor de función
`ULIMIT`	Límite superior para el valor de función
`a0, a1, a2, a3`	Coeficientes del polinomio

También es posible acceder a estos valores a través de la variable de sistema

`$AC_FCTLL[n]`	Límite inferior para el valor de función
`$AC_FCTUL[n]`	Límite superior para el valor de función
`$AC_FCT0[n]`	a_0
`$AC_FCT1[n]`	a_1
`$AC_FCT2[n]`	a_2
`$AC_FCT3[n]`	a_3

Nota

Escribir variables de sistema

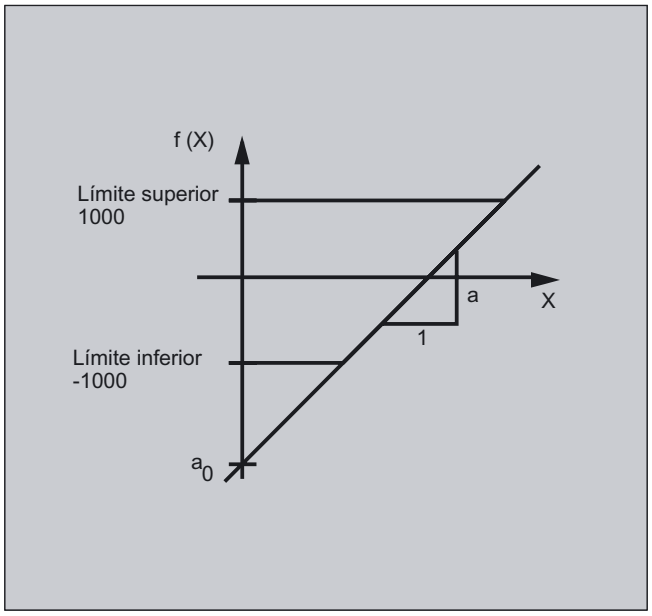
- Las variables de sistema se pueden escribir desde el programa de pieza o desde una acción síncrona. Al escribir desde el programa de pieza es preciso programar STOPRE para escritura síncrona a la secuencia.
- Las variables de sistema `$AC_FCTLL[n]`, `$AC_FCTUL[n]`, `$AC_FCT0[n]` a `$AC_FCTn[n]` se pueden modificar desde acciones síncronas

En la escritura desde acciones síncronas se activan de inmediato los coeficientes de polinomios y los límites de funciones.

Ejemplo: Polinomio para sección recta

Con el límite superior 1000, límite inferior -1000, sección de ordenadas a_0=$AA_IM[X] y pendiente 1, la definición del polinomio es:

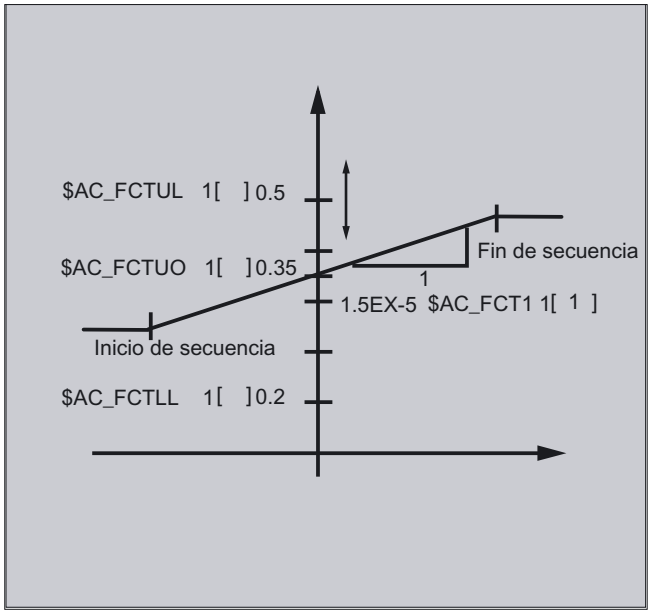
```
FCTDEF(1, -1000,1000,$AA_IM[X],1)
```



Ejemplo: Control de potencia del láser

Una de las aplicaciones posibles de la definición de polinomios es el control de potencia del láser.

Control de potencia del láser significa:
Poder influenciar una salida analógica en función, p. ej., de la velocidad en la trayectoria.



```
Código del programa                                  Comentarios
$AC_FCTLL[1]=0.2                                     ; Definición de los coeficientes
                                                       del polinomio
$AC_FCTUL[1]=0.5
$AC_FCT0[1]=0.35
$AC_FCT1[1]=1.5EX-5
STOPRE
ID=1 DO $AC_FCTUL[1]=$A_INA[2]*0.1 +0.35             ; Modificar límite superior Online.
ID=2 DO SYNFCT(1,$A_OUTA[1],$AC_VACTW)               ; en función de la velocidad de
                                                       contorneado (consignada en
                                                       $AC_VACTW) se efectúa el control
                                                       de potencia del láser a través de
                                                       la salida analógica 1
```

Nota

El polinomio anteriormente definido se utiliza con la función SYNFCT.

10.4.7 Función de sincronización (SYNFCT)

Función

SYNFCT calcula el valor de salida de un polinomio de tercer grado ponderado con las variables de entrada. El resultado se encuentra en las variables de salida y se limita por arriba y por abajo.

La función de evaluación se utiliza en:

- con regulación AC (control adaptativo),
- con control de potencia del láser,
- con control anticipativo de posición.

Sintaxis

```
SYNFCT (número de polinomio, salida de variable de proceso
principal, entrada de variable de proceso principal)
```

Descripción

Como variables de proceso principal se pueden elegir variables que entran al proceso:

- Con influencia aditiva
- Con influencia mutiplicativa
- Como decalaje de posición
- Directamente

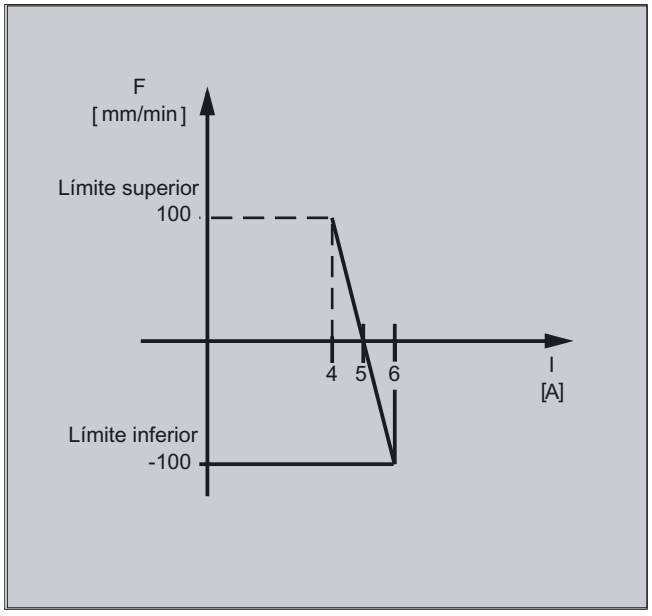
`DO SYNFCT`	Activación de la función de evaluación
`Número polinomio`	Polinomio definido con FCTDEF (ver subapartado "Definición de polinomios")
`Salida de variable de proceso principal`	Escribir variable de proceso principal
`Entrada de variable de proceso principal`	Leer variable de proceso principal

Ejemplo: Regulación adaptativa AC (aditiva)

Influencia aditiva sobre la velocidad de avance programada

Se desea efectuar un control aditivo de la velocidad de avance programada mediante la corriente del eje X (eje de penetración):

La velocidad de avance debe variar en +/- 100†mm/min, la corriente oscila +/-1A alrededor del punto de trabajo, 5A.



1. Definición de polinomios

Determinación de los coeficientes

$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$

a_1 = -100mm/1 min A

a_0 = -(-100)*5 =500

$a_2 = a_3$ = 0 (sin término cuadrado ni cúbico)

Límite superior = 100

Límite inferior = -100

De ello resulta:

```
FCTDEF(1,-100,100,500,-100,0,0)
```

2. Activar la regulación AC

```
ID=1 DO SYNFCT(1,$AC_VC,$AA_LOAD[x])
```

;Leer carga de eje actual (% corriente de accionamiento máx.) a través de `$AA_LOAD[x]`,
;calcular corrección de la velocidad de avance de trayectoria utilizando el polinomio anteriormente definido.

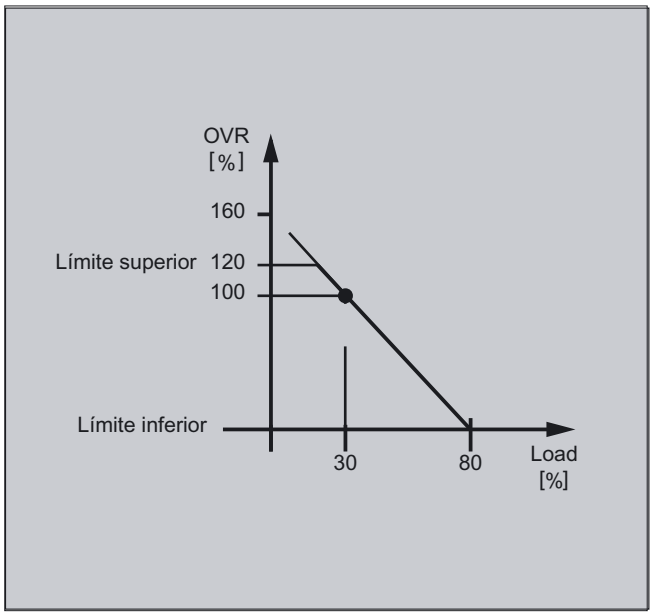
Ejemplo: Regulación adaptativa AC (multiplicativa)

Influencia multiplicativa sobre la velocidad de avance programada

Se desea efectuar una adaptación multiplicativa de la velocidad de avance programada, sin que dicha velocidad supere - dependiendo de la carga del accionamiento - ciertos límites:

- Para una carga del accionamiento del 80%, debe parar el avance: corrección = 0.
- Para una carga del accionamiento del 30%, se puede desplazar con la velocidad de avance programada: corrección = 100%.

La velocidad de avance se puede sobrepasar en máx. 20 %: corrección máx. = 120%



1. Definición de polinomio

Determinación de los coeficientes

$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$

$a_1 = -100\%/(80-30)\% = -2$

$a_0 = 100 + (2*30) = 160$

$a_2 = a_3 = 0$ (sin término cuadrado ni cúbico)

Límite superior = 120

Límite inferior = 0

De ello resulta:

```
FCTDEF(2,0,120,160,-2,0,0)
```

2. Activar la regulación AC

```
ID=1 DO SYNFCT(2,$AC_OVR,$AA_LOAD[x])
```

;Leer carga de eje actual (% corriente de accionamiento máx.) a través de `$AA_LOAD[x]` ,
;calcular corrección del avance utilizando el polinomio anteriormente definido.

10.4.8 Regulación de distancia con corrección limitada ($AA_OFF_MODE)

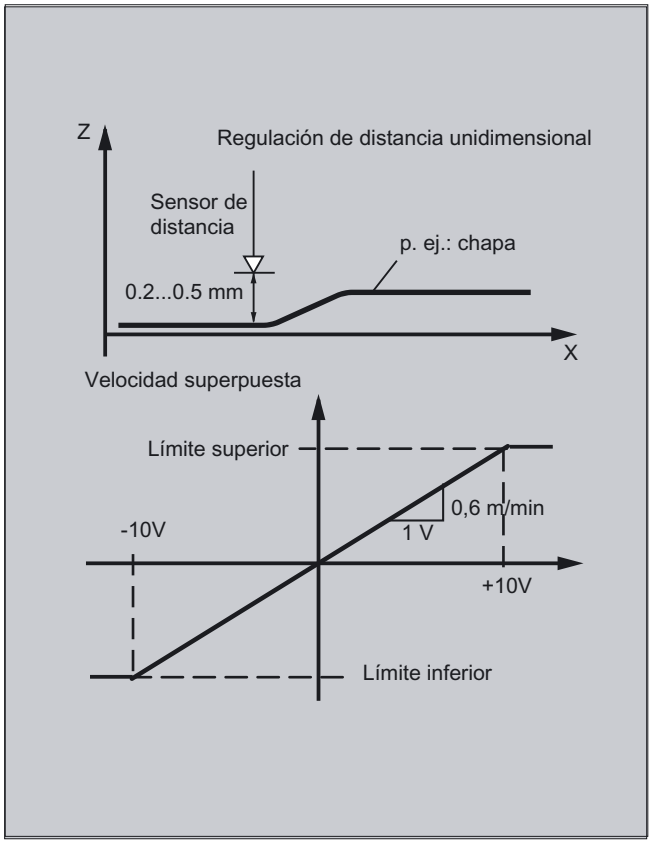
Nota

¡Esta función no está disponible para SINUMERIK 828D!

Función

El cálculo integrador de los valores de distancia se realiza con comprobación del margen límite:

$AA_OFF_MODE = 1



ATENCIÓN
La ganancia del lazo de regulación superior depende del ajuste del ciclo IPO. Remedio: Leer e incluir DM para el ciclo IPO.

Nota

Limitación de la velocidad del interpolador superpuesto mediante DM 32020: JOG_VELO en ciclo IPO 12 ms. Fórmula para la velocidad:

$$\frac{0.120mm}{0.6ms} / mV = 0.6\frac{m}{\min} / V$$

Ejemplo

Subprograma "AON": Regulación de distancia CON

Código del programa	Comentarios
PROC AON	
$AA_OFF_LIMIT[Z]=1	; Definir valor límite.
FCTDEF(1, -10, +10, 0, 0.6, 0.12)	; Definición de polinomio
ID=1 DO SYNFCT(1,$AA_OFF[Z],$A_INA[3])	; Regulación de distancia activa.
ID=2 WHENEVER $AA_OFF_LIMIT[Z]<>0 DO $AA_OVR[X] = 0	; Si se supera el margen límite, bloquear eje X.
RET	
ENDPROC.	

Subprograma "AOFF": Regulación de distancia DES

Código del programa	Comentarios
PROC AOFF	
CANCEL(1)	; Acción síncrona borrar regulación de distancia
CANCEL(2)	; Borrar comprobación del margen límite
RET	
ENDPROC.	

Programa principal "MAIN"

Código del programa	Comentarios
AON	; Regulación de distancia CON
...	
G1 X100 F1000	
AOFF	; Regulación de distancia DES
M30	

Información adicional

Decalaje de posición en el sistema de coordenadas básico

La variable de sistema $AA_OFF[Eje] permite superponer un desplazamiento a cada eje del canal. Actúa como decalaje de posición en el sistema de coordenadas básico.

El decalaje de posición así programado se superpone de inmediato al eje correspondiente, independientemente de que se haya programado o no, desplazamiento para el eje.

Limitar salida de variable de proceso principal:

Se puede limitar el valor que debe corregirse absolutamente (salida de variable de proceso principal) al valor memorizado en el dato de operador DO43350 $SA_AA_OFF_LIMIT.

A través del dato de máquina DM36750 $MA_AA_OFF_MODE se especifica el tipo de superposición de la distancia:

Valor	Descripción
0	Evaluación proporcional
1	Evaluación integradora

La variable de sistema $AA_OFF_LIMIT[Eje] permite consultar, dependiendo de la dirección, si el valor de corrección está dentro del margen límite. Esta variable del sistema se puede consultar desde acciones síncronas y, al alcanzarse un valor límite, permite p. ej. parar el eje o activar la alarma.

0: Valor de corrección fuera del margen límite
1 Límite del valor de corrección en sentido positivo alcanzado
-1: Límite del valor de corrección en sentido negativo alcanzado

10.4.9 Corrector de herramienta online (FTOC)

Función

La función `FTOC` permite un desplazamiento superpuesto para un eje geométrico según un polinomio programado con `FCTDEF` en función de un punto de referencia, que puede ser, p. ej., el valor real de un eje.

El coeficiente a_0 de la definición de función `FCTDEF(...)` se evalúa con `FTOC`. Los límites superior e inferior dependen de a_0.

Con `FTOC` pueden programarse como acciones síncronas regulaciones de distancia o correcciones de herramienta online modales.

La función se utiliza en el mecanizado de la pieza y en el diamantado de la muela rectificadora en el mismo canal o en diferentes canales (canal de mecanizado y canal de diamantado).

Las limitaciones y definiciones para el diamantado de muelas rectificadoras se aplican a `FTOC` de la misma forma que la corrección de herramienta online con `PUTFTOCF` (ver "Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 427)").

Sintaxis

```
FCTDEF(<Función>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
FTOC(<Función>,<Referencia>,<Parámetro de herramienta>,<Canal>,<Cabezal>)
...
```

Descripción

`FCTDEF`: Con `FCTDEF` se define la función polinómica para `FTOC`.

Parámetros:

`<Función>`: Número de la función polinómica
Tipo: INT
Rango de valores: 1 ... 3

`<LLimit>`: Límite inferior
Tipo: REAL

`<ULimit>`: Límite superior
Tipo: REAL

`<a0> ... <a3>`: Coeficientes de la función polinómica
Tipo: REAL

`DO FTOC`: Ejecutar la función de escritura continua modal del corrector de herramienta online

Parámetros:

`<Función>`: Número de la función polinómica
Tipo: INT
Rango de valores: 1 ... 3
Nota:
Debe coincidir con lo indicado en `FCTDEF`.

`<Valor de ref.>`: Variable de proceso principal para la cual ha de calcularse un valor de función a través de la función polinómica definida con `FCTDEF`.
Tipo: VAR REAL

`<Parámetro de herramienta>`: Número del parámetro de desgaste (longitud 1, 2 ó 3) en el que debe sumarse el valor de corrección.
Tipo: INT

`<Canal>`: Número del canal en el que debe activarse el corrector de herramienta online.
Tipo: INT
Nota:
La indicación sólo es necesaria si la corrección no debe activarse en el canal activo.

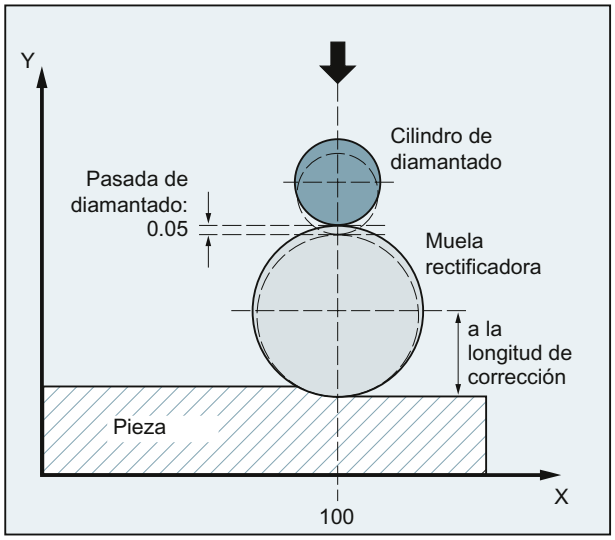
`<Cabezal>`: Número del cabezal para el que debe activarse el corrector de herramienta online.
Tipo: INT
Nota:
La indicación sólo es necesaria si, en lugar de la herramienta que se encuentra activa y en uso, se debe corregir una muela rectificadora no activa.

Nota

En el canal de destino debe estar activada la función FTOCON.

Ejemplo

Debe corregirse la longitud de la muela rectificadora activa que está en funcionamiento.



Código del programa	Comentarios
FCTDEF(1,-1000,1000,-$AA_IW[V],1)	; Definir función.
ID=1 DO FTOC(1,$AA_IW[V],3,1)	; Seleccionar corrección de herramienta Online: El valor real del eje V es el valor inicial del polinomio 1. En el canal 1, el resultado se suma como valor de corrección a la longitud 3 de la muela activa.
WAITM(1,1,2)	; Sincronización con canal de mecanizado.
G1 V-0.05 F0.01 G91	; Movimiento de penetración para diamantar.
G1 V-0.05 F0.02	
...	
CANCEL(1)	; Desactivar corrección online.
...	

10.4.10 Corrección longitudinal de herramienta Online ($AA_TOFF)

Función

A través de la variable de sistema $AA_TOFF[ ], las longitudes de herramienta efectivas según las tres direcciones de herramienta se pueden superponer de forma tridimensional en tiempo real.

Cómo índice se utilizan los tres identificadores de eje geométrico. De este modo, se define el número de direcciones de corrección a través de los ejes geométricos activos al mismo tiempo.

Todas las correcciones pueden estar activas a la vez.

Sintaxis

```
N.. TRAORI
N.. TOFFON(X, 25)
N.. WHEN TRUE DO $AA_TOFF[X]
N.. TOFFON(Y, 25)
N.. WHEN TRUE DO $AA_TOFF[Y]
N.. TOFFON(Z, 25)
N.. WHEN TRUE DO $AA_TOFF[Z]
```

Descripción

`TOFFON`	**T**ool **Off**set **ON** (activar corrección longitudinal de herramienta online) En la activación se puede indicar, para el correspondiente sentido de corrección, un valor de corrección que se aplica inmediatamente.
`TOFFOF`	**T**ool **Off**set **OF** (poner a cero corrección longitudinal de herramienta online) Los correspondientes valores de corrección se ponen a cero y se inicia una parada de decodificación previa.
`X, Y, Z`	Sentido de corrección para el valor de corrección indicado con TOFFON

`$AA_TOFF[X]=Valor`
`$AA_TOFF[Y]=Valor`
`$AA_TOFF[Z]=Valor`
	Superposición en dirección X
	Superposición en dirección Y
	Superposición en dirección Z

Ejemplos

Ejemplo 1: Activación de la corrección longitudinal de herramienta

Código del programa	**Comentarios**
N10 TRAORI(1)	; Transformada activada.
N20 TOFFON(Z)	; Activación de la corrección longitudinal de herramienta Online para la dirección de herramienta Z.
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; Para la dirección de herramienta Z se interpola una corrección longitudinal de herramienta de 10.
N40 TOFFON(X)	; Activación de la corrección longitudinal de herramienta Online para la dirección de herramienta X.
N50 ID=1 DO $AA_TOFF[X]=$AA_IW[X2] G4 F5	; Para la dirección de herramienta X se ejecuta una corrección en función de la posición del eje X2.
...	
	; Asignar corrección actual en dirección X. Para la dirección de herramienta X, la corrección longitudinal de herramienta vuelve a 0:
N100 XOFFSET=$AA_TOFF_VAL[X] N120 TOFFON(X, -XOFFSET) G4 F5	

Ejemplo 2: Cancelación de la corrección longitudinal de herramienta

Código del programa	**Comentarios**
N10 TRAORI(1)	; Transformada activada.
N20 TOFFON(X)	; Activación de la dirección de herramienta Z.
N30 WHEN TRUE DO $AA_TOFF[X] = 10 G4 F5	; para la dirección de herramienta X se interpola una corrección longitudinal de herramienta de 10.
...	
N80 TOFFOF(X)	; El decalaje de posición de la dirección de herramienta X se borra: ...$AA_TOFF[X]=0 No se desplaza ningún eje; a la posición actual en el WKS se suma el offset de posición n conforme a la orientación actual.

10.4.11 Desplazamientos de posicionado

Funcionamiento

Desde las acciones síncronas es posible posicionar los ejes asíncronamente al programa de pieza. La programación de los ejes de posicionado desde acciones síncronas se recomienda para procesos cíclicos o controlados principalmente por sucesos. Los ejes programados desde acciones síncronas se denominan **ejes de comando**.

Programación

Bibliografía:
/PG/ Manual de programación, Fundamentos; apartado "Definición de recorridos"
/FBSY/ Descripción de las funciones de acciones síncronas; "Arrancar ejes de comando"

Parámetros

El sistema de unidades para tareas de posicionamiento en acciones síncronas se define con los códigos G `G70`/`G71`/`G700`/`G710`.

Programando las Funciones G en la acción síncrona puede definirse la evaluación INCH/METRIC para la acción síncrona independientemente del contexto del programa de pieza.

10.4.12 Posicionar eje (POS)

Función

El desplazamiento de ejes de posicionado, al contrario que la programación desde el programa de pieza, no influye de ningún modo sobre la ejecución del programa de pieza.

Sintaxis

```
POS[Eje]=Valor
```

Descripción

`DO POS`	Arrancar/posicionar eje de comando
`Eje`	Nombre del eje a desplazar
`Valor`	Indicación del valor a recorrer (según el modo de desplazamiento)

Ejemplos

Ejemplo 1:

Código del programa	Comentarios
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100	; Desplazar el eje U según el modo de desplazamiento, incrementalmente en 100 (pulg/mm) o a la posición 100 (pulg/mm) desde el origen del control.
	; Desplazar eje U el valor calculado a partir de variables de proceso principal:
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=$AA_MW[V]-$AA_IM[W]+13.5	

Ejemplo 2:

El entorno del programa influye en el recorrido de posicionamiento del eje de posicionado (ninguna función G en la parte de la acción síncrona):

Código del programa	Comentarios
N100 R1=0	
N110 G0 X0 Z0	
N120 WAITP(X)	
N130 ID=1 WHENEVER $R==1 DO POS[X]=10	
N140 R1=1	
N150 G71 Z10 F10	; Z=10 mm X=10 mm
N160 G70 Z10 F10	; Z=254 mm X=254 mm
N170 G71 Z10 F10	; Z=10 mm X=10 mm
N180 M30	

G71 determina, como parte de la acción síncrona, el recorrido del eje de posicionado de forma unívoca (métrica), independientemente del entorno del programa:

Código del programa	Comentarios
N100 R1=0	
N110 G0 X0 Z0	
N120 WAITP(X)	
N130 ID=1 WHENEVER $R==1 DO G71 POS[X]=10	
N140 R1=1	
N150 G71 Z10 F10	; Z=10 mm X=10 mm
N160 G70 Z10 F10	; Z=254 mm X=10 mm (X posicionado siempre en 10 mm)
N170 G71 Z10 F10	; Z=10 mm X=10 mm
N180 M30	

Si el desplazamiento de eje no debe arrancar al inicio de la secuencia, desde una acción síncrona es posible detener en 0 la corrección del eje hasta el momento de arranque deseado:

```
Código del programa                                    Comentarios
WHENEVER $A_IN[1]==0 DO $AA_OVR[W]=0 G01 X10 Y25 F750 POS[W]=1500 FA=1000
                                                       ; El eje de posicionado se detiene mientras la entrada
                                                         digital 1=0.
```

10.4.13 Posición en el margen de referencia definido (POSRANGE)

Función

Con la función POSRANGE( ) se puede determinar si la posición teórica interpolada actual de un eje se encuentra en una ventana en una posición de referencia especificada. Los datos de posición pueden referirse a sistemas de coordenadas definibles.

Si se solicita la posición real de un eje de valor módulo se tiene en cuenta la corrección del módulo.

Nota

La función puede solicitarse solamente desde la acción síncrona. Si se solicita desde el programa de pieza, se emite la alarma 14091 %1 Secuencia %2 Función no permitida, índice: %3 con el índice 5.

Sintaxis

```
BOOL POSRANGE(eje, Refpos, Winlimit, [Coord])
```

Descripción

`BOOL POSRANGE`	La posición actual del eje de comando se encuentra en la ventana de la posición de referencia especificada.
`AXIS <eje>`	Identificador de eje de máquina, de canal o geométrico
`REAL Refpos`	Posición de referencia en el sistema de coordenadas Coord
`REAL Winlimit`	Valor que genera el límite para la ventana de posición
`INT Coord`	El MKS está activo opcionalmente. Son posibles: 0 para MKS (sistema de coordenadas de máquina) 1 para BKS (sistema básico de coordenadas) 2 para ENS (sistema de punto cero ajustable) 3 para WKS (sistema de coordenadas de pieza)

Valor de función

Posición teórica actual en función de la indicación de posición del sistema de coordenadas especificado.

`Valor de función: TRUE`	si Refpos(Coord) - abs(Winlimit) ≤ Actpos(Coord) ≤ Refpos(Coord) + abs(Winlimit)
`Valor de función: FALSE`	en el resto de casos

10.4.14 Arrancar/parar eje (MOV)

Función

Con MOV[Eje] = valor se puede arrancar un eje de comando sin necesidad de indicar la posición final. El eje correspondiente se desplaza en el sentido programado hasta que se determine otro movimiento a través de una nueva orden de desplazamiento o de posicionado o se pare el eje con una orden de parada.

Sintaxis

```
MOV[Eje] = Valor
```

Descripción

`DO MOV`	Iniciar desplazamiento de ejes de comando
`Eje`	Nombre del eje que se desea arrancar
`Valor`	Orden de arranque para desplazamiento/parada El signo determina el sentido de desplazamiento Tipo de datos del valor: INTEGER.
`Valor >0 (generalmente +1)`	Sentido positivo
`Valor <0 (generalmente -1)`	Sentido negativo
`Valor ==0`	Parar desplazamiento del eje

Nota

Si se programa `MOV[eje] = 0` para detener un eje indexado, entonces el eje se para en la próxima posición (próximo índice).

Ejemplo

Código del programa	**Comentarios**
`... DO MOV[U]=0`	`; El eje U se para`

10.4.15 Intercambio de eje (RELEASE, GET)

Función

Para cambiar la herramienta, pueden solicitarse los ejes de comando correspondientes como acción de una acción síncrona con GET(eje). El tipo de eje asignado a este canal y el derecho de interpolación relacionado con él en este momento, puede solicitarse mediante la variable de sistema $AA_AXCHANGE_TYP. En función del estado real y del canal que posee el derecho de interpolación actual de este eje, son posibles diversos procesos.

Si se ha completado el cambio de herramienta, puede liberarse este eje de comando para el canal como acción de una acción síncrona mediante RELEASE(eje).

Fabricante de la máquina

El eje en cuestión tiene que estar asignado al canal a través de datos de máquina. ¡Observar las indicaciones del fabricante de la máquina!

Sintaxis

`GET(eje[,eje{,...}])` solicitar eje

`RELAESE(eje[,eje{,...}])` liberar eje

Descripción

`DO RELEASE`	Liberar eje como eje neutro
`DO GET`	Llevar eje para el intercambio de ejes
`Eje`	Nombre del eje que debe arrancarse

Ejemplo: Desarrollo del programa para un intercambio de ejes de dos canales

El eje Z es conocido en el primer y en el segundo canal.

Desarrollo del programa en el primer canal:

Código del programa	Comentarios
WHEN TRUE DO RELEASE(Z)	; El eje Z se convierte en eje neutro
WHENEVER($AA_TYP[Z]==1) DO RDISABLE	; Bloqueo de lectura mientras el eje Z es el eje de programa
N110 G4 F0.1	
WHEN TRUE DO GET(Z)	; El eje Z se vuelve a convertir en eje de programa CN
WHENEVER($AA_TYP[Z]<>1) DO RDISABLE	; Bloqueo de lectura hasta que el eje Z sea el eje de programa
N120 G4 F0.1	
WHEN TRUE DO RELEASE(Z)	; El eje Z se convierte en eje neutro
WHENEVER($AA_TYP[Z]==1) DO RDISABLE	; Bloqueo de lectura mientras el eje Z es el eje de programa
N130 G4 F0.1	;
N140 START(2)	; Iniciar el 2.º canal

Desarrollo del programa en el segundo canal:

Código del programa	Comentarios
WHEN TRUE DO GET(Z)	; Llevar el eje Z al 2º canal
WHENEVER($AA_TYP[Z]==0) DO RDISABLE	; Bloqueo de lectura mientras el eje Z se encuentre en otro canal
N210 G4 F0.1	
WHEN TRUE DO GET(Z)	; El eje Z se convierte en eje de programa CN
WHENEVER($AA_TYP[Z]<>1) DO RDISABLE	; Bloqueo de lectura hasta que el eje Z sea el eje de programa
N220 G4 F0.1	
WHEN TRUE DO RELEASE(Z)	; Eje Z en el 2º canal eje neutro
WHENEVER($AA_TYP[Z]==1) DO RDISABLE	; Bloqueo de lectura mientras el eje Z es el eje de programa
N230 G4 F0.1	
N250 WAITM(10, 1, 2)	; Sincronizar con el canal 1

Desarrollo posterior del programa en el primer canal:

Código del programa	Comentarios
`N150 WAIM(10, 1, 2)`	`; Sincronizar con el canal 2`
`WHEN TRUE DO GET(Z)`	`; Llevar el eje Z a este canal`
`WHENEVER($AA_TYP[Z]==0) DO RDISABLE`	`; Bloqueo de lectura mientras el eje Z se encuentre en otro canal`
`N160 G4 F0.1`	
`N199 WAITE(2)`	
`N999 M30`	`; Esperar el fin del programa en el canal 2`

Ejemplo: Intercambio de ejes en el ciclo tecnológico

El eje U ($MA_AUTO_GET_TYPE=2) es conocido en el primer y en el segundo canal; actualmente, el canal 1 tiene el derecho de interpolación. En el canal 2 se inicia el siguiente ciclo tecnológico:

Código del programa	Comentarios
`GET(U)`	`; Llevar el eje U al canal`
`POS[U]=100`	`; El eje U se tiene que desplazar a la posición 100`

La línea del desplazamiento del ejes de comando POS[U] sólo se ejecuta cuando el eje U ha sido llevado al canal 2.

Ejecución

El eje solicitado en el momento de activación de la acción `GET(eje)` se puede leer, con respecto al tipo de eje, para un intercambio de ejes con la variable de sistema ($AA_AXCHANGE_TYP[<eje>]:

- 0: Eje asignado al programa CN
- 1: Eje asignado al PLC o activo como eje de comando o eje de vaivén
- 2: Otro canal tiene el derecho de interpolación
- 3: El eje es el eje neutro
- 4: El eje neutro está controlado por el PLC
- 5: Otro canal tiene el derecho de interpolación; el eje está solicitado para el programa CN
- 6: Otro canal tiene el derecho de interpolación; el eje está solicitado como eje neutro
- 7: El eje del PLC o activo como eje de comando o de vaivén; el eje está solicitado para el programa CN
- 8: El eje del PLC o activo como eje de comando o de vaivén; el eje está solicitado como eje neutro

Limitaciones

El eje en cuestión tiene que estar asignado al canal a través de datos de máquina.

Un eje controlado exclusivamente por el PLC no se puede asignar al programa CN

Bibliografía:
/FB2/ Manual de funciones de ampliación; Ejes de posicionado (P2)

Solicitar eje de otro canal con la acción GET

Si, en el momento de la activación de la acción `GET`, **otro canal tiene el derecho de escritura** (derecho de interpolación) para el eje ($AA_AXCHANGE_TYP[<eje>] == 2), el eje se solicita mediante intercambio de ejes de este canal ($AA_AXCHANGE_TYP[<eje>]==6) y se asigna lo antes posible al canal solicitante.

Entonces toma el estado de eje neutro ($AA_AXCHANGE_TYP[<eje>]==3).

No tiene lugar ninguna reorganización en el canal solicitante.

Asignación **como eje del programa CN con reorganización:**

Si el eje ya se ha solicitado en el momento de activación de la acción `GET` como eje neutro ($AA_AXCHANGE_TYP[<eje>]==6), el eje se solicita para el programa CN ($AA_AXCHANGE_TYP[<eje>]==5) y se asigna lo antes posible al programa CN del canal ($AA_AXCHANGE_TYP[<eje>]==0).

Eje ya asignado al canal solicitado

Asignación **como eje del programa CN con reorganización:**

Si, en el momento de la activación, el eje solicitado ya se ha asignado al canal solicitante, y en el estado Eje neutro – no controlado por el PLC – ($AA_AXCHANGE_TYP[<eje>]==3), se asigna al programa CN ($AA_AXCHANGE_TYP[<eje>]==0).

El eje en el estado de eje neutro está controlado por el PLC

Si, en el estado Eje neutro, el eje está controlado por el PLC ($AA_AXCHANGE_TYP[<eje>]==4), el eje se solicita como eje neutro ($AA_AXCHANGE_TYP[<eje>] == 8); en este caso, el eje es dependiente del bit 0 en el dato de máquina DM 10722: AXCHANGE_MASK bloqueado para el intercambio automático entre canales (bit 0 == 0). Esto corresponde a ($AA_AXCHANGE_STAT[<eje>] == 1).

El eje está activo como eje de comando neutro o eje de vaivén o está asignado al PLC

Si el eje está activo como eje de comando o de vaivén o asignado al PLC para el desplazamiento, eje PLC == eje de posicionado concurrente, ($AA_AXCHANGE_TYP[<eje>]==1), el eje se solicita como eje neutro ($AA_AXCHANGE_TYP[<eje>] == 8); en este caso, el eje es dependiente del bit 0 en el dato de máquina DM 10722: AXCHANGE_MASK bloqueado para el intercambio automático entre canales (bit 0 == 0). Esto corresponde a ($AA_AXCHANGE_STAT[<eje>] == 1).

Una nueva acción GET solicita entonces el eje para el programa CN ($AA_AXCHANGE_TYP[<eje>] pasa a == 7).

El eje ya está asignado al programa CN

Si el eje ya está asignado al programa CN del canal ($AA_AXCHANGE_TYP[<eje>]==0) o se solicita dicha asignación, p. ej., intercambio de ejes iniciado por el programa CN ($AA_AXCHANGE_TYP[<eje>]==5 ó $AA_AXCHANGE_TYP[<eje>] == 7), no se produce ningún cambio de estado.

10.4.16 Avance por eje (FA)

Función

El avance específico para ejes de comando es modalmente activo.

Sintaxis

```
FA[<Eje>]=<Valor>
```

Ejemplo

Código del programa	**Comentarios**
`ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 FA[U]=990`	`; Definir valor de avance fijo.`
	`; Formar valor de avance a partir de variables de proceso principal:`
`ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 FA[U]=$AA_VACTM[W]+100`	

10.4.17 Finales de carrera software

Funcionamiento

La limitación de zona de trabajo programada con G25/G26 se toma en cuenta, según los datos de ajuste $SA_WORKAREA_PLUS_ENABLE, para los ejes de comando.

La activación y desactivación de la limitación de zona de trabajo a través de las funciones G WALIMON/WALIMOF en el programa de pieza no surte efecto sobre los ejes de comando.

10.4.18 Coordinación de ejes

Función

Normalmente, un eje se desplaza desde el programa de pieza o como eje de posicionado desde la acción síncrona.

Si, no obstante, dicho eje se ha de desplazar alternadamente desde el programa de pieza, como eje de contorneado o de posicionado, y desde acciones síncronas, entonces se efectúa una transferencia coordinada entre ambos desplazamientos de eje.

Si a continuación se desplaza un eje de comando desde el programa de pieza, entonces hay que reorganizar el pretratamiento. Esto, a su vez, implica la interrupción de la ejecución del programa de pieza, similar a una parada de decodificación previa.

Ejemplo: Desplazar el eje X, a elección, desde el programa de pieza y desde acciones síncronas

Código del programa	**Comentarios**
N10 G01 **X100** Y200 F1000	; Eje X programado en el programa de pieza
…	
N20 ID=1 WHEN $A_IN[1]==1 DO **POS[X]=150** FA[X]=200	; Arrancar posicionado desde acción síncrona cuando esté activa la entrada digital
…	
CANCEL(1)	; Cancelar la acción síncrona
…	
N100 G01 **X240** Y200 F1000	; X se convierte en eje de contorneado; antes del movimiento existe un tiempo de espera debido a la transferencia de ejes si la entrada digital era 1 y X se posicionó desde una acción síncrona.

Ejemplo: Modificar orden de desplazamiento para el mismo eje

Código del programa	**Comentarios**
ID=1 EVERY $A_IN[1]>=1 DO POS[V]=100 FA[V]=560	; Arrancar posicionado desde acción síncrona cuando la entrada digital >= 1
ID=2 EVERY $A_IN[2]>=1 DO POS[V]=$AA_IM[V] FA[V]=790	; El eje sigue, se define la 2ª entrada; es decir, con dos acciones síncronas activas al mismo tiempo, la posición final y el avance para el eje V se siguen de forma fluida en un movimiento en curso.

10.4.19 Poner valor real (PRESETON)

Función

Con la ejecución de PRESETON (eje, valor) no se modifica la posición actual del eje, se le asigna un nuevo valor.

PRESETON es ejecutable desde acciones síncronas para:

- Ejes giratorios con módulo que fueron activados desde el programa de pieza
- Todos los ejes de comando activados desde la acción síncrona

Sintaxis

```
DO PRESETON (eje, valor)
```

Descripción

`DO PRESETON`	Poner valor real en acciones síncronas
`Eje`	Eje cuyo origen del control se quiere modificar
`Valor`	Valor al que se modifica el origen del control

Limitaciones para ejes

`PRESETON` no puede utilizarse para ejes participantes en la transformada.

El mismo eje sólo se puede desplazar con desfase temporal desde el programa de pieza o desde una acción síncrona. Por consiguiente, al programar un eje desde un programa de pieza se pueden presentar tiempos de espera si dicho eje ha sido programado previamente en una acción síncrona.

Si dicho eje se utiliza alternadamente, entonces se realiza una transferencia coordinada entre ambos desplazamientos. Para ello hay que interrumpir la ejecución del programa de pieza.

Ejemplo

Desplazar el origen del control de un eje

Código del programa	**Comentarios**
`WHEN $AA_IM[a] >= 89.5 DO PRESETON(a4,10.5)`	`; Desplazar el origen del control del eje a en 10,5 unidades de longitud (pulgadas o mm) en el sentido positivo del eje`

10.4.20 Movimientos de cabezales

Función

Desde las acciones síncronas es posible posicionar los cabezales de forma asíncrona al programa de pieza. Este tipo de programación es recomendable para operaciones o procesos cíclicos, controlados esencialmente por sucesos.

Si desde varias acciones síncronas simultáneamente activas se emiten diversas órdenes para un cabezal, entonces se ejecuta la última emitida.

Ejemplo Arrancar/parar/posicionar cabezal

Código del programa	Comentarios
ID=1 EVERY $A_IN[1]==1 DO M3 S1000	; Ajustar el sentido y la velocidad de giro
ID=2 EVERY $A_IN[2]==1 DO SPOS=270	; Posicionar cabezal

Ejemplo sentido de giro, ajustar velocidad de giro/ posicionar cabezal

Código del programa	Comentarios
ID=1 EVERY $A_IN[1]==1 DO M3 S300	; Ajustar el sentido y la velocidad de giro
ID=2 EVERY $A_IN[2]==1 DO M4 S500	; Especificar un nuevo sentido de giro y una nueva velocidad de giro
ID=3 EVERY $A_IN[3]==1 DO S1000	; Especificar una nueva velocidad de giro
ID=4 EVERY ($A_IN[4]==1) AND ($A_IN[1]==0) DO SPOS=0	; Posicionar cabezal

10.4.21 Arrastre de ejes (TRAILON, TRAILOF)

Función

Al activar el acoplamiento desde la acción síncrona puede estar en desplazamiento el eje maestro. En este caso, el eje esclavo se acelera hasta la velocidad nominal. La posición del eje maestro en el instante de sincronización de la velocidad constituye la posición de arranque para el arrastre. La funcionalidad se describe en el apartado "Influenciando la trayectoria".

Sintaxis

Activar arrastre

```
DO TRAILON(eje esclavo, eje maestro, factor de acoplamiento)
```

Desactivar arrastre

```
DO TRAILOF (eje esclavo, eje maestro, eje maestro 2)
```

Descripción

Activar arrastre asíncrono:

`... DO TRAILON(FA, LA, Kf)`	con: FA: Eje esclavo LA: Eje maestro Kf: Factor de acoplamiento

Desactivar arrastre asíncrono:

`... DO TRAILOF(FA, LA, LA2)`	con: FA: Eje esclavo LA: Eje maestro, opcional LA2: Eje maestro 2, opcional
`... DO TRAILOF(FA)`	Se desactivan todos los acoplamientos del eje esclavo.

Ejemplo

Código del programa		Comentarios
`$A_IN[1]==0 DO TRAILON(Y,V,1)`	;	Activar el 1er conjunto maestro-esclavos cuando la entrada digital sea 1
`$A_IN[2]==0 DO TRAILON(Z,W,-1)`	;	Activación del 2º conjunto maestro-esclavos
`G0 Z10`	;	Penetración de los ejes Z y W en sentidos opuestos
`G0 Y20`	;	Penetración de los ejes Y y V en el mismo sentido
`...`		
`G1 Y22 V25`	;	Superposición de un desplazamiento del eje arrastrado "V" dependiente e independiente
`...`		
`TRAILOF(Y,V)`	;	Desactivación del 1er conjunto maestro-esclavos
`TRAILOF(Z,W)`	;	Desactivación del 2º conjunto maestro-esclavos

Ejemplo: prevención de conflictos con TRAILOF

Si se pretende volver a liberar un eje acoplado para el acceso como eje de canal, se tiene que llamar previamente la función `TRAILOF`. Se tiene que asegurar que `TRAILOF` está ejecutada antes de que el canal solicite el eje en cuestión. Éste no es el caso en el siguiente ejemplo:

```
…
N50 WHEN TRUE DO TRAILOF(Y,X)
N60 Y100
…
```

En este caso, el eje no se libera a tiempo porque la acción síncrona que actúa secuencia a secuencia con `TRAILOF` se activa de forma síncrona con `N60` (ver el apartado Acciones síncronas a desplazamientos, "Estructura, principios generales").
Para evitar situaciones de conflicto, se debería

proceder de la siguiente manera:

```
…
N50 WHEN TRUE DO TRAILOF(Y,X)
N55 WAITP(Y)
N60 Y100
```

10.4.22 Acoplamiento de valores maestros (LEADON, LEADOF)

Nota

¡Esta función no está disponible para SINUMERIK 828D!

Función

El acoplamiento entre ejes de valores maestros se puede programar sin limitación alguna en acciones síncronas. La modificación de una tabla de levas con un acoplamiento existente sin resincronización previa sólo es posible, de forma opcional, en acciones síncronas.

Sintaxis

Activar acoplamiento de valores maestros

```
DO LEADON(eje esclavo, eje maestro, tabla de levas nº, OVW)
```

Desactivar acoplamiento de valores maestros

```
DO LEADOF(eje esclavo, eje maestro, eje maestro 2)
```

Descripción

`Activar acoplamiento entre ejes de valores maestros:`	
`...DO LEADON(FA, LA, NR, OVW)`	donde: FA: Eje esclavo LA: Eje maestro NR: Número de la tabla de levas almacenada OVW: Permitir la sobrescritura de un acoplamiento existente con la tabla de levas modificada
`Desactivar acoplamiento entre ejes de valores maestros:`	
`...DO LEADOF(FA, LA)`	con: FA: Eje esclavo LA: Eje maestro, opcional
`... DO LEADOF(FA)`	Forma abreviada sin indicación del eje maestro

Liberar el acceso mediante acciones síncronas RELEASE

La habilitación de un eje a acoplar para el acceso mediante acción síncrona implica la llamada previa de la función RELEASE del eje esclavo a acoplar.

Ejemplo:

```
RELEASE(XKAN)
ID=1 every SR1==1 to LEADON(CACH,XKAN,1)
```

OVW=0 (valor por defecto)

Sin una nueva sincronización no se puede especificar una nueva tabla de levas para un acoplamiento existente. Una modificación de la tabla de levas exige la desconexión previa del acoplamiento existente y una nueva conexión con el número de tabla de levas modificado. Esto produce una nueva sincronización del acoplamiento.

Modificación de la tabla de levas con acoplamiento existente con OVW=1

Con `OVW=1` se puede especificar una nueva tabla de levas para un acoplamiento existente. No tiene lugar ninguna nueva sincronización. El eje esclavo trata de seguir con la mayor rapidez posible al valor de posición definido por la nueva tabla de levas.

Ejemplo: Corte al vuelo

Un perfil extruido, que pasa continuamente por la zona de trabajo de un dispositivo tronzador, debe ser dividido en secciones de igual longitud.

Eje X: Eje en el cual se mueve el perfil extruido, sistema de coordenadas de pieza
Eje X1: Eje de máquina del perfil extruido, MKS
Eje Y: Eje en el cual el dispositivo tronzador "sigue" al perfil extruido

Se presupone que tanto el posicionado de la herramienta tronzadora como su mando son controlados por el PLC. Para determinar el sincronismo entre el perfil y la herramienta tronzadora se pueden evaluar las señales de la interconexión del PLC.

Acciones
Activar acoplamiento, `LEADON`
Desactivar acoplamiento, `LEADOF`
Poner valor real, `PRESETON`

Código del programa	**Comentarios**
N100 R3=1500	; Longitud de la pieza que debe tronzarse
N200 R2=100000 R13=R2/300	
N300 R4=100000	
N400 R6=30	; Posición inicial del eje Y
N500 R1=1	; Condición de arranque para eje de cinta
N600 LEADOF(Y,X)	; Borrar un posible acoplamiento
N700 CTABDEF(Y,X,1,0)	; Definición de tabla
N800 X=30 Y=30	; Pares de valores
N900 X=R13 Y=R13	
N1000 X=2*R13 Y=30	
N1100 CTABEND	; Fin de la definición de tabla
N1200 PRESETON(X1,0)	; PRESET para empezar
N1300 Y=R6 G0	; Posición inicial del eje Y, el eje es lineal
N1400 ID=1 WHENEVER $AA_IW[X]>$R3 DO PESETON(X1,0)	; PRESET después de longitud R3, nuevo inicio después del tronzado
N1500 RELEASE(Y)	
N1800 ID=6 EVERY $AA_IM[X]<10 DO LEADON(Y,X,1)	; Acoplar Y a X a través de la tabla 1 si X < 10
N1900 ID=10 EVERY $AA_IM[X]>$R3-30 DO EADOF(Y,X)	; Desacoplar > 30 de la longitud de tronzado transportada
N2000 WAITP(X)	
N2100 ID=7 WHEN $R1==1 DO MOV[X]=1 FA[X]=$R4	; Mover continuamente el eje del perfil extruido
N2200 M30	

10.4.23 Medir (MEAWA, MEAC)

Función

A diferencia de una aplicación en secuencias de desplazamiento del programa de pieza, la función de medición se puede activar y desactivar discrecionalmente desde acciones síncronas.

Más información sobre la medición en Órdenes de desplazamiento especiales "Función de medida ampliada

Sintaxis

Medición axial sin borrado de trayecto residual

```
MEAWA[eje]=(Modo, suceso de conmutación_1, ..._4)
```

Medición continua sin borrado de trayecto residual

```
MEAC[eje]=(Modo, memoria de medición, suceso de conmutación_1,
..._4))
```

Descripción

Código del programa		**Comentarios**	
DO MEAWA	;	Activar medición axial	
DO MEAC	;	Activar medición continua	
Eje	;	Nombre del eje para el cual se mide	
Modo	;	Indicación del dígito de **decenas** 0: sistema de medida activo Número de sistemas de medida (según el modo) 1: 1er sistema de medida 2: 2º sistema de medida 3: ambos sistemas de medida	Indicación del dígito de **unidades** 0: Cancelar orden de medición hasta 4 sucesos de conmutación activables 1: simultáneamente 2: de forma sucesiva 3: como 2, pero sin vigilancia del suceso de conmutación 1 en el arranque
Suceso de conmutación_1 a _4	;	: flanco ascendente, palpador 1 -1: flanco descendente, palpador 1 opcional 2: flanco ascendente, palpador 2 opcional -2: flanco descendente, palpador 2 opcional	
Memoria de medición	;	Número de la memoria cíclica FIFO	

10.4.24 Inicialización de las variables de matriz (SET, REP)

Función

En las acciones síncronas, las variables de matriz pueden inicializarse o describirse con determinados valores.

Nota

Sólo son posibles las variables que pueden describirse en las acciones síncronas. Con ello, no pueden inicializarse los datos de máquina. Las variables de eje no pueden especificarse con el valor NO_AXIS.

Sintaxis

```
DO MATRIZ[n,m]=SET(<Valor1>,<Valor2>,...)
DO MATRIZ[n,m]=REP(<Valor>)
```

Descripción

`MATRIZ[n,m]`	Índices de matriz programados La inicialización comienza por los índices de la matriz programados. En las matrices de 2 dimensiones se incrementa en primer lugar el segundo índice. Si se trata de un índice de eje éste no se ejecuta.
`SET(<Valor1>,<Valor2>,...)`	Inicialización con listas de valores La matriz está descrita desde el índice de la matriz programada con los parámetros de `SET`. Se asignarán tantos elementos de la matriz como valores hayan sido programados. Si se programan más valores que la cantidad de elementos de la matriz restantes, se emitirá una alarma
`REP(<Valor>)`	Inicialización con los mismos valores La matriz está descrita repetidamente desde el índice de la matriz programada hasta el final de la matriz con el parámetro (`<Valor>`) de `REP`.

Ejemplo

Código del programa	**Comentarios**
`WHEN TRUE DO SYG_IS[0]=REP(0)`	`; Resultado:`
`WHEN TRUE DO SYG_IS[1]=SET(3,4,5)`	`; SYG_IS[0]=0`
	`SYG_IS[1]=3`
	`SYG_IS[2]=4`
	`SYG_IS[3]=5`
	`SYG_IS[4]=0`

10.4.25 Definir/borrar metas de espera (SETM, CLEARM)

Función

En acciones síncronas se pueden activar o borrar metas de espera para, p. ej., coordinar entre sí los canales.

Sintaxis

```
DO SETM(<Número de meta>)
DO CLEARM(<Número de meta>)
```

Descripción

`SETM`	Comando para definir la meta de espera para el canal El comando `SETM` puede escribirse en el programa de pieza y en la parte de una acción síncrona. Define la meta (`<Número de meta>`) para el canal en el cual se ejecuta el comando.
`CLEARM`	Comando para borrar la meta de espera para el canal El comando `CLEARM` puede escribirse en el programa de pieza y en la parte de una acción síncrona. Borra la meta (`<Número de meta>`) para el canal en el cual se ejecuta el comando.
`<Número de meta>`	Meta de espera

10.4.26 Reacciones a errores (SETAL)

Función

Las reacciones a errores pueden programarse con acciones síncronas. Para ello se consultan variables de estado y se inician las acciones correspondientes.

Reacciones posibles a estados de error:

- Parar eje (Corrección=0)
- Activar alarma

  Con `SETAL` pueden activarse alarmas de ciclos a partir de acciones síncronas.
- Activar salida
- Total de acciones posibles en acciones síncronas

Sintaxis

Activar alarma de ciclos:

```
DO SETAL(<Número de alarma>)
```

Descripción

SETAL	Comando para activar una alarma de ciclos	
<Número de alarma>	Número de la alarma	
	Margen de alarma de ciclo para el usuario:	65000 a 69999

Ejemplo

Código del programa	**Comentarios**
ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO $AA_OVR[X2]=0	; Si la distancia de seguridad entre los ejes X1 y X2 es demasiado pequeña, detener el eje X2.
ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO SETAL(65000)	; Si la distancia de seguridad entre los ejes X1 y X2 es demasiado pequeña, activar la alarma 65000.

10.4.27 Desplazamiento a tope fijo (FXS, FXST, FXSW, FOCON, FOCOF)

Función

Los comandos para la función "Desplazamiento a tope fijo" se programarán con los comandos de programa de pieza FXS, FXST y FXSW en acciones síncronas/ciclos tecnológicos.

La activación puede tener lugar sin movimiento; el par se limita inmediatamente. Tan pronto como el eje se mueve hacia el lado del valor nominal, se controla en cuanto al tope.

Desplazamiento con par/fuerza limitados (FOC)

Mediante acciones síncronas, la función permite modificar el par o la fuerza en todo momento, y puede activarse de forma modal o referida a la secuencia.

Sintaxis

```
FXS[<Eje>]
FXST[<Eje>]
FXSW[<Eje>]
FOCON[<Eje>]
FOCOF[<Eje>]
```

Descripción

FXS	Selección sólo en sistemas con accionamientos digitales (VSA, HSA, HLA)
FXST	Modificación del par de bloqueo FXST
FXSW	Modificación de la ventana de vigilancia FXSW
FOCON	Activación de la limitación modal del par o de la fuerza
FOCOF	Desactivar la limitación del par o de la fuerza
<Eje>	Identificador de eje Los valores admisibles son • Identificador de eje geométrico • Identificador de eje de canal • Identificador de eje de máquina

Nota

La selección sólo debe realizarse una vez

Ejemplos

Ejemplo 1: Desplazamiento a tope fijo (FXS), accionado mediante una acción síncrona

```
Código del programa                                                    Comentarios
Eje Y:                                                                 ; Acciones síncronas estáticas
Activar:
N10 IDS=1 WHENEVER (($R1==1) AND $AA_FXS[y]==0)) D $R1=0 FXS[Y]=1 FXST[Y]=10 FA[Y]=200 POS[Y]=150
                                                                       ; Mediante la definición de $R1=1, se
                                                                         activa FXS para el eje Y, el par efec-
                                                                         tivo se reduce a un 10% y se inicia un
                                                                         desplazamiento en dirección al tope.
N11 IDS=2 WHENEVER ($AA_FXS[Y]==4) DO FXST[Y]=30                       ; En el momento en el que se reconoce el
                                                                         tope ($AA_FXS[Y]==4), el par se incre-
                                                                         menta al 30%.
N12 IDS=3 WHENEVER ($AA_FXS[Y]==1) DO FXST[Y]=$R0                      ; Tras alcanzar el tope, el par se con-
                                                                         trola en función de R0.
N13 IDS=4 WHENEVER (($R3==1) AND $AA_FXS[Y]==1)) DO FXS[Y]=0 FA[Y]=1000 POS[Y]=0
                                                                       ; Cancelar en función de R3 y retirar.
N20 FXS[Y]=0 G0 G90 X0 Y0                                              ; Desarrollo normal del programa:
N30 RELEASE(Y)                                                         ; Habilitar el eje Y para el movimiento en
                                                                         acción síncrona.
N40 G1 F1000 X100                                                      ; Movimiento de otro eje.
N50 ...
N60 GET(Y)                                                             ; Incluir el eje Y de nuevo en el conjunto
```

Ejemplo 2: Activación de la limitación de par/fuerza (FOC)

Código del programa	Comentarios
N10 FOCON[X]	; Activación modal de la limitación.
N20 X100 Y200 FXST[X]=15	; X se desplaza con un par reducido (15%).
N30 FXST[X]=75 X20	; Modificación del par a 75%, X se desplaza con este par limitado.
N40 FOCOF[X]	; Desactivar la limitación de par.

Información adicional

Selección múltiple

Si debido a un error de programación se vuelve a activar la función tras la activación (`FXS[<Eje>] = 1`), se dispara la siguiente alarma:

Alarma 20092 "Todavía está activa marcha a tope"

Una programación que solicita o bien $AA_FXS[ ] o bien una marca propia (aquí R1) en la condición impide una activación múltiple de la función "Fragmento de programa de pieza":

```
Código del programa
N10 R1=0
N20 IDS=1 WHENEVER ($R1==0 AND
$AA_IW[AX3] > 7) DO R1=1 FXST[AX1]=12
```

Acciones síncronas referidas a secuencias

Mediante la programación de una acción síncrona referida a secuencia, es posible activar el desplazamiento a tope fijo durante un desplazamiento de aproximación.

Ejemplo:

Código del programa	Comentarios
N10 G0 G90 X0 Y0	
N20 WHEN $AA_IW[X] > 17 DO FXS[X]=1	; Si X alcanza una posición superior a 17 mm, se activa FXS.
N30 G1 F200 X100 Y110	

Acciones síncronas estáticas y referidas a secuencias

En las acciones síncronas estáticas y referidas a secuencias pueden utilizarse los mismos comandos `FXS`, `FXST` y `FXSW` que en el transcurso normal de programa de pieza. Los valores que se asignan pueden ser fruto de un cálculo.

10.4.28 Determinación del ángulo de tangente de trayectoria en acciones síncronas

Función

La variable de sistema legible en acciones síncronas $AC_TANEB (**T**angent **AN**gel at **E**nd of **B**lock) determina el ángulo entre la tangente de trayectoria en el punto final de la secuencia actual y la tangente de trayectoria en el punto inicial de la siguiente secuencia programada.

Parámetros

El ángulo de tangente se emite siempre como valor positivo en un rango de 0.0 a 180.0 grados. Si no existe ninguna secuencia posterior en el proceso principal, se emite el ángulo de -180,0 grados.

La variable de sistema `$AC_TANEB` no se debería leer para secuencias generadas por el sistema (secuencias intermedias). Para distinguir si se trata de una secuencia programada (secuencia principal) sirve la variable de sistema `$AC_BLOCKTYPE`.

Ejemplo

```
ID=2 EVERY $AC_BLOCKTYPE==0 DO $SR1 = $AC_TANEB
```

10.4.29 Determinación de la corrección actual

Funcionamiento

La corrección actual

(parte del CN) se puede leer y escribir en acciones síncronas con las variables de sistema:

`$AA_OVR` corrección por eje

`$AC_OVR` corrección de la trayectoria

en acciones síncronas.

La corrección especificada por el PLC se pone a disposición de las acciones síncronas para lectura en las variables de sistema:

`$AA_PLC_OVR` corrección por eje

`$AC_PLC_OVR` corrección de la trayectoria

para la lectura.

La corrección resultante

se pone a disposición de las acciones síncronas para lectura en las variables de sistema:

`$AA_TOTAL_OVR` corrección por eje

`$AC_TOTAL_OVR` corrección de la trayectoria

para la lectura.

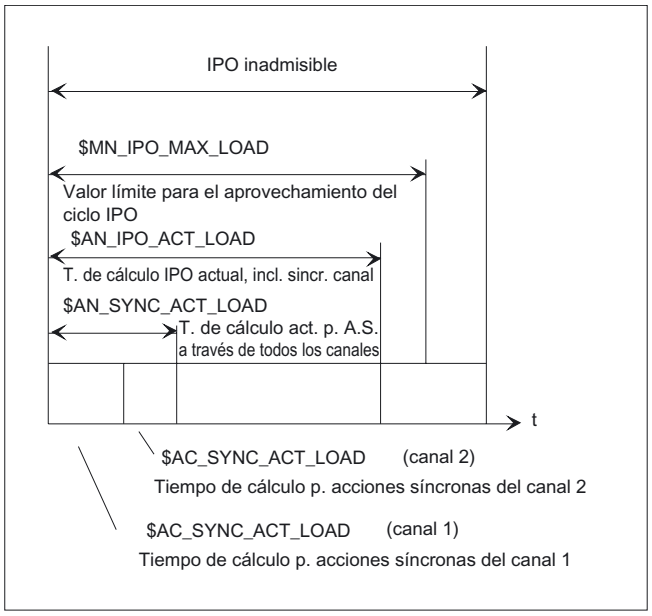
La corrección resultante se calcula como:

```
$AA_OVR * $AA_PLC_OVR ó
$AC_OVR * $AC_PLC_OVR
```

10.4.30 Evaluación de la utilización a través de tiempo necesario para las acciones síncronas

Función

En un ciclo de interpolación, el CN tiene que interpretar acciones síncronas, así como calcular movimientos, etc. Con las variables de sistema que se presentan a continuación, las acciones síncronas se pueden informar sobre las proporciones de tiempo actuales de las acciones síncronas en el ciclo de interpolación y el tiempo de cálculo de los reguladores de posición.



Descripción

Las variables sólo tienen valores válidos si el dato de máquina `$MN_IPO_MAX_LOAD` es mayor que 0. De lo contrario, las variables indican siempre, tanto para SINUMERIK powerline como para sistemas solution line, el tiempo de cálculo neto en el que ya no se tienen en cuenta las interrupciones generadas por HMI. El tiempo de cálculo neto resulta:

- del tiempo de acción síncrona,
- del tiempo de regulación de posición y
- del tiempo de cálculo IPO sin interrupciones debidas a HMI

Las variables de sistema contienen siempre los valores del ciclo IPO **anterior**

`$AN_IPO_ACT_LOAD`	Tiempo de cálculo IPO actual (incl. acciones síncronas de todos los canales)
`$AN_IPO_MAX_LOAD`	Tiempo de cálculo IPO más largo (incl. acciones síncronas de todos los canales)
`$AN_IPO_MIN_LOAD`	Tiempo de cálculo IPO más corto (incl. acciones síncronas de todos los canales)
`$AN_IPO_LOAD_PERCENT`	Tiempo de cálculo IPO actual en relación con el ciclo IPO (%)
`$AN_SYNC_ACT_LOAD`	Tiempo de cálculo actual para las acciones síncronas en todos los canales
`$AN_SYNC_MAX_LOAD`	Tiempo de cálculo más largo para las acciones síncronas en todos los canales
`$AN_SYNC_TO_IPO`	Porcentaje del total de acciones síncronas en el tiempo de cálculo IPO total (en todos los canales)
`$AC_SYNC_ACT_LOAD`	Tiempo de cálculo actual para acciones síncronas en el canal
`$AC_SYNC_MAX_LOAD`	Tiempo de cálculo más largo para acciones síncronas en el canal
`$AC_SYNC_AVERAGE_LOAD`	Tiempo de cálculo medio para acciones síncronas en el canal
`$AN_SERVO_ACT_LOAD`	Tiempo de cálculo actual del regulador de posición
`$AN_SERVO_MAX_LOAD`	Tiempo de cálculo más largo del regulador de posición
`$AN_SERVO_MIN_LOAD`	Tiempo de cálculo más corto del regulador de posición

Variable de la señalización de sobrecarga:

A través del dato de máquina `$MN_IPO_MAX_LOAD` se ajusta a partir de qué tiempo de cálculo IPO neto (en % del ciclo IPO) la variable de sistema `$AN_IPO_LOAD_LIMIT` se debe poner en TRUE. Cuando la carga actual vuelve a pasar por debajo de este límite, la variable vuelve a ponerse en FALSE. Si el dato de máquina es 0, toda la función de diagnóstico está desactivada.

La evaluación de `$AN_IPO_LOAD_LIMIT` permite al usuario definir una estrategia propia para evitar un rebose de planos.

10.5 Ciclos tecnológicos

Función

Como acción en instrucciones síncronas se pueden activar programas que, no obstante, sólo pueden incluir funciones admisibles también como acciones en instrucciones síncronas. Los programas así configurados se denominan ciclos tecnológicos.

Los ciclos tecnológicos se memorizan como subprogramas en el control numérico.

En un canal se pueden procesar simultáneamente varios ciclos tecnológicos o acciones.

Programación

Para la programación de ciclos tecnológicos rigen las siguientes reglas:

- El fin de programa se programa con `M02`/`M17`/`M30`/`RET`.
- Dentro de un nivel del programa, todas las acciones indicadas en `ICYCOF` sin ciclos de espera pueden ejecutarse en un ciclo.
- Pueden solicitarse hasta 8 ciclos tecnológicos consecutivos por cada acción síncrona.
- Los ciclos tecnológicos también son posibles en acciones síncronas por secuencias.
- Pueden programarse tanto estructuras de control `IF` como instrucciones de salto `GOTO`, `GOTOF` y `GOTOB`.
- Para secuencias con instrucciones `DEF` y `DEFINE` rige:
  - Las instrucciones `DEF` y `DEFINE` no se tienen en cuenta en los ciclos tecnológicos.
  - Emiten un aviso de alarma si la sintaxis no es correcta o no está completa.
  - Pueden pasarse por alto sin que se emita una alarma ni ser creadas por sí mismas.
  - Son tenidas en cuenta por completo como ciclo de programa de pieza con asignaciones de valores.

Transferencia de parámetros

Es posible la transferencia de parámetros a ciclos tecnológicos. Deben tenerse en cuenta tanto tipos de datos sencillos, que se transmiten como parámetros formales "Call-By-Value", como ajustes estándares que se activan al solicitar ciclos tecnológicos. Se trata de:

- Valores estándar programados cuando no hay ningún parámetro de transferencia programado.
- Asignar valores iniciales a parámetros estándar.
- Parámetros actuales no inicializados que se transmiten con un valor estándar.

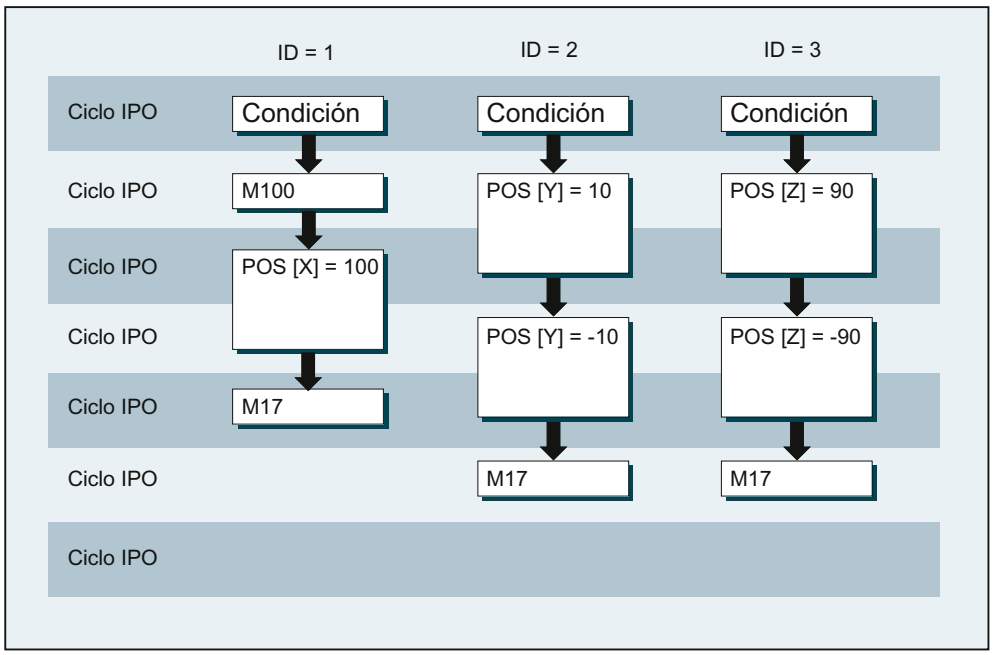
Ejecución

Los ciclos tecnológicos se arrancan tan pronto se cumplan sus condiciones. Cada línea del ciclo tecnológico se ejecuta en un ciclo IPO independiente. Para los ejes de posicionado se precisan para ello varias cadencias IPO. Otras funciones se ejecutan en una cadencia. En el ciclo tecnológico se lleva a cabo de forma secuencial la ejecución de las secuencias.

Si en la misma cadencia, es decir, ciclo de interpolación, se llaman acciones que se excluyen entre sí, entonces se activa aquella acción que haya sido llamada desde la acción síncrona con el número ID más alto.

Ejemplos

Ejemplo 1: Mediante la definición de entradas digitales se inician programas de ejes



Programa principal:

Código del programa	**Comentarios**
`ID=1 EVERY $A_IN[1]==1 DO Eje_X`	`; Si la entrada 1 es 1, se inicia el programa de eje Eje_X.`
`ID=2 EVERY $A_IN[2]==1 DO Eje _Y`	`; Si la entrada 2 es 1, se inicia el programa de eje Eje_Y.`
`ID=3 EVERY $A_IN[3]==1 DO Eje_Z`	`; Si la entrada 3 es 1, se inicia el programa de eje Eje_Z.`
`M30`	

Programa de eje Eje_X:

```
Código del programa
M100
POS[X]=100 FA[X]=300
M17
```

Programa de eje Eje_Y:

```
Código del programa
POS[Y]=10 FA[Y]=200
POS [Y] = -10
M17
```

Programa de eje Eje_Z:

```
Código de programa
POS[Z]=90 FA[Z]=250
POS[Z]=-90
M17
```

Ejemplo 2: Distintas secuencias de programa en el ciclo tecnológico

```
Código de programa
PROC CYCLE
N10 DEF REAL VALOR=12.3
N15 DEFINE ABC AS G01
```

Ambas secuencias se pasan por alto sin emitir una alarma ni crear las variables o la macro.

```
Código del programa
PROC CYCLE
N10 DEF REAL
N15 DEFINE ABC G01
```

Ambas secuencias provocan una alarma CN porque la sintaxis no está escrita correctamente.

```
Código de programa
PROC CYCLE
N10 DEF AXIS EJE1=XX2
```

Si no se conoce el eje XX2, se emite la alarma 12080. De lo contrario, se pasa por alto la secuencia sin emitir ninguna alarma ni crear las variables.

Código de programa

```
PROC CYCLE
N10 DEF AXIS EJE1
N15 G01 X100 F1000
N20 DEF REAL VALOR1
```

La secuencia `N20` provoca siempre la alarma 14500, ya que el 1.er comando de programa no puede ir seguido de una instrucción de definición.

10.5.1 Variable de contexto ($P_TECCYCLE)

Función

Con ayuda de las variables $P_TECCYCLE pueden dividirse los programas en programas de acción síncrona y programas de decodificación previa. De esta forma es posible ejecutar de forma alternativa secuencias o secuencias de programa cuya sintaxis es correcta también como ciclos de programa de pieza.

Interpretar variables de contexto

La variable de sistema $P_TECCYCLE permite controlar interpretaciones de partes del programa específicas del contexto en ciclos tecnológicos:

```
IF $P_TECCYCLE==TRUE
...                        ; Secuencia de programa para el ciclo tecnológico en la
                             acción síncrona.
ELSE
...                        ; Secuencia de programa para el ciclo de programa de
                             pieza.
ENDIF
```

Nota

Una secuencia con una sintaxis de programa errónea o no permitida, así como con asignaciones de valor desconocidas conlleva también la emisión de una alarma en el ciclo de programa de pieza.

Ejemplo

Secuencia de programa con solicitud de $P_TECCYCLE en el ciclo tecnológico:

```
Código de programa              Comentarios
PROC CYCLE
N10 DEF REAL VALOR1             ; Se pasa por alto en el ciclo tecnológico.
N15 G01 X100 F1000
N20 IF $P_TECCYCLE==TRUE
...                             ; Secuencia de programa para el ciclo tecnológico (sin la variable
                                  VALOR1).
N30 ELSE
...                             ; Secuencia de programa para el ciclo de programa de pieza (la variable
                                  VALOR1 está presente).
N40 ENDIF
```

10.5.2 Parámetro Call-By-Value

Función

Los ciclos tecnológicos pueden definirse con parámetros Call-By-Value. Los parámetros pueden ser tipos de datos sencillos como INT, REAL, CHAR, STRING, AXIS y BOOL.

Nota

Los parámetros formales que se transfieren a Call-By-Value no pueden ser matrices.

Los parámetros actuales también pueden estar formados por parámetros por defecto (ver "Inicialización de parámetros por defecto (Página 658)").

Sintaxis

```
ID=1 WHEN $AA_IW[X]>50 DO TEC(IVAL,RVAL,,SVAL,AVAL)
```

En el caso de los parámetros actuales no inicializados, se transfiere un valor por defecto:

```
ID=1 WHEN $AA_IW[X]>50 DO TEC(IVAL,RVAL,,SYG_SS[0],AVAL)
```

10.5.3 Inicialización de parámetros por defecto

Función

A los parámetros por defecto se les puede asignar también un valor inicial en la instrucción PROC.

Sintaxis

Asignar parámetros por defecto en el ciclo tecnológico:

```
PROC TEC (INT IVAL=1, REAL RVAL=1.0, CHAR CVAL='A', STRING[10]
SVAL="ABC", AXIS AVAL=X, BOOL BVAL=TRUE)
```

Si un parámetro actual está formado por un parámetro por defecto, se transfiere el valor inicial de la instrucción PROC. Esto es válido tanto en el programa de pieza como en las acciones síncronas.

Ejemplo

Código del programa	**Comentarios**
`TEC (IVAL, RVAL, SVAL, AVAL)`	`;  En el caso de CVAL y BVAL rige el valor inicial`

10.5.4 Control de la ejecución de ciclos tecnológicos (ICYCOF, ICYCON)

Función

Para controlar la ejecución en el tiempo de ciclos tecnológicos se utilizan las instrucciones de programación ICYCOF e ICYCON.

Con ICYCOF se ejecutan todas las secuencias de un ciclo tecnológico en una única cadencia de interpolación. Todas las acciones cuya ejecución requiera varias cadencias ocasionan en ICYCIF procesos de ejecución paralelos.

Aplicación

En el caso de ICYCON, los desplazamientos de los ejes de comando pueden ocasionar un retraso en la ejecución de un ciclo tecnológico. Si se quiere evitar esta situación, ICYCOF permite ejecutar todas las acciones en una cadencia de interpolación sin tiempos de espera.

Sintaxis

Para la ejecución cíclica de ciclos tecnológicos se aplica:

`ICYCON` cada secuencia de un ciclo tecnológico se ejecuta según ICYCON en un ciclo IPO separado

`ICYCOF` todas las secuencias subsiguientes de un ciclo tecnológico se ejecutan según ICYCOF en un ciclo IPO

Nota

Las dos instrucciones de programación ICYCON e ICYCOF sólo actúan en el nivel de programa. En el programa de pieza ambas instrucciones se ignoran sin reacción alguna.

Ejemplo de modo de ejecución ICYCOF

Código del programa	**Comentarios**
Ciclo IPO	; PROC TECHNOCYC
1.	; $R1=1
2.25	; POS[X]=100
26.	; ICYCOF
26.	; $R1=2
26.	; $R2=$R1+1
26.	; POS[X]=110
26.	; $R3=3
26.	; RET

10.5.5 Concatenaciones en cascada de ciclos tecnológicos

Función

Se pueden ejecutar hasta 8 ciclos tecnológicos conectados en serie. De esta forma, pueden programarse varios ciclos tecnológicos en una acción síncrona.

Sintaxis

```
ID=1 WHEN $AA_IW[X]>50 DO TEC1($R1) TEC2 TEC3(X)
```

Secuencia de ejecución

Los ciclos tecnológicos son ejecutados consecutivamente (en cascada), de izquierda a derecha, según la programación arriba indicada. Si un ciclo debe ejecutarse en modo ICYCON, éste retrasará las ejecuciones subsiguientes. Si se emite una alarma, se interrumpen todas las acciones subsiguientes.

10.5.6 Ciclos tecnológicos en acciones síncronas por secuencias

Función

Los ciclos tecnológicos también son posibles en acciones síncronas por secuencias.

Si el tiempo de ejecución de un ciclo tecnológico es superior al tiempo de ejecución de su correspondiente secuencia, se interrumpirá el ciclo tecnológico cuando se cambie de secuencia.

Nota

Un ciclo tecnológico no impide el cambio de secuencia.

10.5.7 Estructuras de control (IF)

Función

Para crear ramas el orden de ejecución de los ciclos tecnológicos pueden utilizarse estructuras de control IF en las acciones síncronas.

Sintaxis

```
IF <Condición>
$R1=1
[ELSE] opcional
$R1=0
ENDIF
```

10.5.8 Instrucciones de salto (GOTO, GOTOF, GOTOB)

Función

En los ciclos tecnológicos son posibles las instrucciones de salto GOTO, GOTOF y GOTOB. Las labels especificadas deben estar disponibles en el subprograma para que no se emita ninguna alarma.

Nota

Las labels y los números de secuencia deben ser siempre constantes.

Sintaxis

Saltos incondicionales

`GOTO` label, número de secuencia

`GOTOF` label, número de secuencia

`GOTOB` label, número de secuencia

Instrucciones de salto y destinos de salto

`GOTO`	El salto se efectúa en primer lugar hacia delante y, a continuación, hacia atrás
`GOTOF`	Salto hacia delante
`GOTOB`	Salto hacia atrás
`Label:`	Meta de salto
`Número de secuencia`	Destino de salto a esta secuencia
`N100`	El número de secuencia es una secuencia auxiliar
`:100`	El número de secuencia es una secuencia principal

10.5.9 Bloquear, habilitar, resetear (LOCK, UNLOCK, RESET)

Función

La ejecución de un ciclo tecnológico puede bloquearse, habilitarse de nuevo o resetearse mediante otra acción síncrona modal.

Sintaxis

```
LOCK(<n1>,<n2>,...)
UNLOCK(<n1>,<n2>,...)
RESET(<n1>,<n2>,...)
```

Descripción

`LOCK`	Comando de bloqueo de acciones síncronas La acción activa se interrumpirá.
`UNLOCK`	Comando de habilitación de acciones síncronas
`RESET`	Comando para resetear ciclos tecnológicos
`<n1>,<n2>,...`	Números de identificación de las acciones síncronas o ciclos tecnológicos que deben bloquearse, habilitarse o resetearse.

Bloqueo de acciones síncronas

Las acciones síncronas modales con los números ID **<n> = 1 ... 64** pueden ser bloqueadas por el PLC. En este caso no se evalúa la condición correspondiente y se bloquea la ejecución de la función en el NCK.

Con una señal de la interconexión del PLC se pueden bloquear globalmente todas las acciones síncronas.

Nota

Una acción síncrona programada actúa por defecto y se puede proteger contra la sobrescritura/bloqueo mediante un dato de máquina.

El cliente final no debería influir en las acciones síncronas definidas por el fabricante de la máquina.

Ejemplos

Ejemplo 1: Bloquear acciones síncronas (LOCK)

```
Código del programa
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
```

Ejemplo 2: Habilitar acciones síncronas (UNLOCK)

```
Código del programa
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
...
N250 ID=3 WHENEVER $A_IN[3]==1 DO UNLOCK(1)
```

Ejemplo 3: Interrumpir ciclo tecnológico (RESET)

```
Código del programa
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO RESET(1)
```

10.6 Borrar acción síncrona (CANCEL)

Función

Con el comando CANCEL puede interrumpirse (borrarse) una acción síncrona activa modalmente o estáticamente desde el programa de pieza.

Si se interrumpe una acción síncrona, estando activo el desplazamiento de eje de posicionado por ella activado, se concluye dicho desplazamiento. Si se quiere evitar esta situación, el desplazamiento de eje puede frenarse utilizando el borrado de trayecto residual axial antes del comando CANCEL.

Sintaxis

```
CANCEL(<n1>,<n2>,...)
```

Descripción

CANCEL:	Comando para borrar acciones síncronas programadas
<n1>,<n2>,...:	Números de identificación de las acciones síncronas por borrar **Nota:** Si no se indica ningún número de identificación, se borran **todas** las acciones síncronas modales/estáticas.

Ejemplos

Ejemplo 1: Cancelación de una acción síncrona

Código del programa	Comentarios
N100 ID=2 WHENEVER $A_IN[1]==1 DO M130	
...	
N200 CANCEL(2)	; Borra la acción síncrona modal nº 2.

Ejemplo 2: Borrado de trayecto residual antes de cancelar la acción síncrona

Código del programa	Comentarios
N100 ID=17 EVERY $A_IN[3]==1 DO POS[X]=15 FA[X]=1500	; Iniciar desplazamiento de eje de posicionado.
...	
N190 WHEN ... DO DELDTG(X)	; Finalizar desplazamiento de eje de posicionado.
N200 CANCEL(17)	; Borra la acción síncrona modal nº 17.

10.7 Comportamiento de control en determinados estados operativos

POWER ON

Con POWER ON no existen, en principio, acciones síncronas activas. Las acciones síncronas estáticas se pueden activar con un subprograma asíncrono (`ASUP`) activado desde el PLC.

Cambio del modo de operación

Las acciones síncronas activadas con la palabra reservada `IDS` permanecen activas después de cambiar el modo de operación. Todas las demás acciones síncronas se inactivan al cambiar el modo de operación (p. ej.: posicionar ejes) y se activan nuevamente cuando se reposiciona y se vuelve a conmutar al modo automático.

RESET

Con RESET CN finalizan todas las acciones síncronas modales y activas por secuencias. Las acciones síncronas estáticas permanecen activas. Desde ellas se pueden activar nuevas acciones. Si en el momento del RESET hay activo un desplazamiento de eje de comando, éste quedará interrumpido. Las acciones síncronas del tipo WHEN ya ejecutadas dejan de procesarse tras el RESET.

Comportamiento tras RESET		
Acción síncrona/ Ciclo tecnológico	**modal/por secuencia**	**estática-o (IDS)**
	Se interrumpe la acción activa; se borran las acciones síncronas.	Se interrumpe la acción activa; se pone a cero el ciclo tecnológico.
Eje/ cabezal de posicionado	Se interrumpe el movimiento.	Se interrumpe el movimiento.
Cabezal con regulación de velocidad	$MA_SPIND_ACTIVE_AFTER_RESET==1: Cabezal permanece activo $MA_SPIND_ACTIVE_AFTER_RESET==0: El cabezal se para.	
Acoplamiento de valores maestros	$MC_RESET_MODE_MASK, Bit13 == 1: El acoplamiento de valores maestros permanece activo. $MC_RESET_MODE_MASK, Bit13 == 0: El acoplamiento de valor maestro se anula	
Operaciones de medición	Se interrumpen las operaciones de medición iniciadas desde acciones síncronas.	Se interrumpen las operaciones de medición iniciadas desde acciones síncronas estáticas.

Parada CN

Las acciones síncronas **estáticas** permanecen activas con Parada CN. Los desplazamientos activados desde acciones síncronas estáticas no se interrumpen. Las acciones síncronas de **programa local** pertenecientes a la secuencia activa permanecen activas; los movimientos iniciados desde ellas se cancelan.

Fin del programa

El fin del programa y la acción síncrona no se influyen mutuamente. Las acciones síncronas en curso se concluyen aunque haya finalizado el programa. Las acciones síncronas activadas en la secuencia M30 permanecen activas en dicha secuencia. Si se quiere evitar esta situación, debe cancelarse la acción síncrona antes del fin del programa con `CANCEL`.

Comportamiento tras fin de programa		
Acción síncrona/ Ciclo tecnológico	**modal/por secuencia → se cancelan**	**estática-o (IDS) → se mantienen**
Eje/ cabezal de posicionado	Se retarda M30 hasta que se pare el eje/cabezal	Continúa el desplazamiento.
Cabezal con regulación de velocidad	Fin del programa: $MA_SPIND_ACTIVE_AFTER_RESET==1: El cabezal permanece activo $MA_SPIND_ACTIVE_AFTER_RESET==0: El cabezal se para El cabezal permanece activo durante el cambio de modo de operación.	El cabezal permanece activo.
Acoplamiento de valores maestros	$MC_RESET_MODE_MASK, Bit13 == 1: El acoplamiento de valores maestros permanece activo. $MC_RESET_MODE_MASK, Bit13 == 0: El acoplamiento de valor maestro se anula	El acoplamiento iniciado desde una acción síncrona estática se conserva.
Operaciones de medición	Se interrumpen las operaciones de medición iniciadas desde acciones síncronas.	Las operaciones de medición iniciadas desde acciones síncronas estáticas permanecen activas.

Búsqueda de secuencia

Durante la búsqueda de secuencias, se acumulan las acciones síncronas y se evalúan con Marcha CN, iniciando en su caso las correspondientes acciones. Las acciones síncronas estáticas permanecen activas también durante la búsqueda de secuencias. Si durante la búsqueda de secuencias se localizan coeficientes de polinomio programados con `FCTDEF`, entonces se activan éstos directamente.

Interrupción del programa por un subprograma asíncrono ASUP

Inicio ASUP:

Las acciones síncronas de desplazamiento modales y estáticas permanecen activas y tienen vigencia también en el subprograma asíncrono.

Fin ASUP:

Si el subprograma asíncrono no continúa con REPOS, las acciones síncronas a desplazamientos modales y estáticas modificadas en el subprograma asíncrono siguen activas en el programa principal.

Reposicionamiento (REPOS)

Después del reposicionamiento (REPOS) se vuelven a activar las acciones síncronas activas en la secuencia interrumpida. Tras el reposicionado REPOS, las acciones síncronas modales modificadas desde el subprograma asíncrono ya no están activas en la ejecución de la secuencia restante.

Los coeficientes de polinomio programados con `FCTDEF` no se ven afectados por subprogramas asíncronos ni REPOS. Independientemente de dónde hayan sido programados, pueden utilizarse en todo momento tanto en el subprograma asíncrono como en el programa principal incluso tras la ejecución de REPOS.

Comportamiento en caso de alarma

Los desplazamientos de eje y de cabezal iniciados mediante acciones síncronas se frenan si hay activa una alarma con parada de movimiento. Todas las demás acciones (p. ej. activación de una salida) se siguen ejecutando.

Si una acción síncrona dispara una alarma, se interrumpe el procesamiento y dejan de ejecutarse las acciones subsiguientes a dicha acción síncrona. Si la acción síncrona está modalmente activa, no continuará su procesamiento en el siguiente ciclo de interpolación. Por tanto, la alarma se emite sólo una vez. Todas las demás acciones síncronas siguen procesándose.

Las alarmas que tienen una parada de intérprete como reacción sólo se activan una vez que se han ejecutado las secuencias predecodificadas.

Si un ciclo tecnológico dispara una alarma con parada de movimiento, el ciclo deja de procesarse.

Vaivén 11

11.1 Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Función

Un eje de vaivén u oscilación se desplaza alternativamente entre los dos puntos de inversión del sentido de movimiento 1 y 2 con un determinado avance hasta que se desactiva dicho tipo de movimiento (vaivén u oscilación).

El resto de los ejes se pueden interpolar a discreción mientras se realiza el desplazamiento de vaivén. Mediante contorneado o bien mediante un posicionado por eje se puede conseguir una penetración continua. En este caso, no existe **ninguna relación** entre el movimiento de vaivén y el movimiento de penetración.

Características del vaivén asíncrono

- El vaivén asíncrono actúa de forma específica por eje, superando los límites de secuencias.
- Desde el programa de pieza se asegura la activación sincronizada con la secuencia del movimiento de vaivén.
- No se pueden interpolar varios ejes a la vez ni superponer trayectos de vaivén.

Programación

A través de los siguientes comandos es posible la activación e influenciación del vaivén asíncrono correspondiente a la ejecución del programa CN desde el programa de pieza.

Los valores programados se introducen de forma síncrona al mecanizado principal en el bloque de datos de operador correspondiente y permanecen activos hasta que se vuelven a modificar.

Sintaxis

```
OSP1[<Eje>]=<Valor> OSP2[<Eje>]=<Valor>
OST1[<Eje>]=<Valor> OST2[<Eje>]=<Valor>
FA[<Eje>]=<Valor>
OSCTRL[<Eje>]=(<Opción de activación>,<Opción de desactivación>)
OSNSC[<Eje>]=<Valor>
OSE[<Eje>]=<Valor>
OSB[<Eje>]=<Valor>
OS[<Eje>]=1
OS[<Eje>]=0
```

Descripción

`<Eje>` Nombre del eje de vaivén

`OS` Activación/desactivación del vaivén

Valor: 1 Activación del vaivén (**Con**)

0 **Des**activación del vaivén

`OSP1` Definición de la posición del punto de inversión 1

`OSP2` Definición de la posición del punto de inversión 2

Nota:
Si se ha activado un desplazamiento incremental, la posición se calcula de manera incremental respecto a la última posición de inversión correspondiente programada en el programa CN.

`OST1` Definición del tiempo de parada en [s] en el punto de inversión 1

`OST2` Definición del tiempo de parada en [s] en el punto de inversión 2

`<Valor>`:
- -2 La interpolación continúa sin esperar la parada exacta fina.
- -1 Espera hasta parada exacta basta.
- 0 Espera hasta parada exacta fina.
- >0 Espera hasta parada precisa fina y, a continuación, durante el tiempo de parada indicado

  Nota:
  La unidad del tiempo de parada es la misma que cuando se programa dicho tiempo mediante `G4`.

`FA` Definición de la velocidad de avance

El valor de la velocidad de avance se define mediante la velocidad de avance definida para el eje de posicionado. Si no se ha definido ningún valor de avance, entonces se toma por defecto el valor introducido en datos de máquina.

OSCTRL Indicación de las opciones de activación y desactivación

Los valores optativos 0-3 codifican el comportamiento en los puntos de inversión durante la desactivación. Puede seleccionarse una de las variantes 0-3. El resto de ajustes pueden combinarse a voluntad con la variante seleccionada. Se concatenan varias opciones mediante el carácter más (+)

<Valor>: 0 Parada en el siguiente punto de inversión al desactivar al vaivén (ajuste previo)

Nota:
Sólo es posible reseteando los valores 1 y 2.

1 Cuando se desactiva el vaivén, parada en el punto de inversión 1.

2 Cuando se desactiva el vaivén, parada en el punto de inversión 2.

3 Cuando se desactiva el vaivén, no desplazarse hacia el punto de inversión si no se han programado carreras de afinado.

4 Aproximación a la posición final tras el afinado

8 Si se interrumpe el vaivén por medio del borrado de trayecto residual, a continuación deben ejecutarse carreras de afinado y, en caso necesario, efectuar una aproximación a la posición final.

16 Si se interrumpe el vaivén por medio del borrado de trayecto residual, debe efectuarse una aproximación a la respectiva posición de inversión igual que en la desactivación.

32 El nuevo valor del avance se activa sólo tras el siguiente punto de inversión.

64 FA igual a 0, FA = 0: Corrección de la trayectoria activa

FA distinto a 0, FA <> 0: Corrección de la velocidad activa

128 Para eje giratorio DC (por el camino más corto).

256 La carrera de afinado se ejecuta como carrera doble (por defecto); 1 = la carrera de afinado se ejecuta como carrera individual.

512 Primero, desplazamiento a la posición inicial

OSNSC Definición del número de carreras de afinado

OSE Definición de la posición final (en el WKS) hacia la que debe efectuarse la aproximación tras la desactivación del vaivén

Nota:
Al programar OSE, se activa implícitamente la opción 4 para OSCTRL.

OSB Definición de la posición inicial (en el WKS) hacia la que debe efectuarse la aproximación tras la activación del vaivén

La aproximación a la posición inicial se realiza antes del punto de inversión 1. Si la posición inicial coincide con la posición de inversión 1, a continuación se realiza la aproximación a la posición de inversión 2. Al alcanzar la posición inicial no se activa el tiempo de parada, ni siquiera si dicha posición coincide con la posición de inversión 1; en vez de ello, se espera a que se produzca una parada precisa fina. Se cumple una condición de parada precisa configurada.

Nota:
Para que se efectúe la aproximación a la posición inicial debe estar activado el dato de operador DO43770 $SA_OSCILL_CTRL_MASK bit 9.

Ejemplos

Ejemplo 1: El eje de vaivén debe oscilar entre dos puntos de inversión

El eje de vaivén Z debe oscilar entre las posiciones 10 y 100. La aproximación al punto de inversión 1 debe realizarse con parada precisa fina, y al punto de inversión 2, con parada precisa basta. El avance para el eje de vaivén debe ser 250. Al final del mecanizado deben realizarse 3 carreras de afinado y el eje de vaivén debe colocarse en la posición final 200. El avance para el eje de penetración debe ser 1, y el final de la penetración en dirección X debe haberse alcanzado en la posición 15.

Código del programa	**Comentarios**
WAITP(X,Y,Z)	; Posición inicial.
G0 X100 Y100 Z100	; Conmutación al modo Eje de posicionado.
WAITP(X,Z)	
OSP1[Z]=10 OSP2[Z]=100	; Punto de inversión 1, punto de inversión 2.
OSE[Z]=200	; Posición final.
OST1[Z]=0 OST2[Z]=–1	; Tiempo de parada en U1: Parada precisa fina
	; Tiempo de parada en U2: parada exacta basta
FA[Z]=250 FA[X]=1	; Avance del eje de vaivén, avance del eje de penetración.
OSCTRL[Z]=(4,0)	; Opciones de activación.
OSNSC[Z]=3	; 3 carreras de afinado.
OS[Z]=1	; Arranque del vaivén.
WHEN $A_IN[3]==TRUE DO DELDTG(X)	; Borrado de trayecto residual.
POS[X]=15	; Posición inicial del eje X
POS[X]=50	Posición final del eje X.
OS[Z]=0	; Parada del vaivén.
M30	

Nota

La sucesión de comandos `OSP1[Z]=...` a `OSNCS[Z]=...` también puede programarse en una secuencia.

Ejemplo 2: Vaivén con modificación online de la posición de inversión

Los datos de operador necesarios para la definición del movimiento de vaivén asíncrono se pueden ajustar en el programa de pieza.

Si los datos de operador se indican directamente en el programa de pieza, entonces las modificaciones son válidas ya durante el preprocesado. Se puede lograr un comportamiento síncrono mediante una parada de decodificación previa (STOPRE).

```
Código del programa                                                    Comentarios
$SA_OSCILL_REVERSE_POS1[Z]=-10
$SA_OSCILL_REVERSE_POS2[Z]=10
G0 X0 Z0
WAITP(Z)
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0      ; Cuando el valor real del
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0        eje de vaivén sobrepasa
                                                                             el punto de inversión, se
                                                                             detiene el eje de
                                                                             penetración.
OS[Z]=1 FA[X]=1000 POS[X]=40                                           ; Activación del vaivén.
OS[Z]=0                                                                ; Desactivación del vaivén.
M30
```

Información adicional

Eje de vaivén

Para el eje de vaivén rige:

- Cualquier eje puede ser utilizado como eje de vaivén.
- Pueden estar activos varios ejes de vaivén a la vez (máximo: número de ejes de posicionado).
- Para el eje de vaivén siempre está activa la interpolación lineal G1 independientemente del comando G válido actualmente en el programa.

El eje de vaivén puede:

- Ser un eje de entrada para la transformada dinámica.
- Ser un eje maestro en configuraciones tipo Gantry y ejes arrastrados.
- Desplazarse:
  - sin limitación de tirones (BRISK)

    o
  - con limitación de tirones (SOFT)

    o bien,
  - con rampa de aceleración por tramos (como ejes de posicionado)

Punto de inversión en vaivén

A la hora de fijar los puntos de inversión del desplazamiento se deben de tener en cuenta los decalajes de origen actualmente activados:

- Introducción en valor absoluto

  `OSP1[Z]=<Valor>`

  Punto de inversión = Suma de los decalajes + valor programado

- Introducción incremental

  `OSP1[Z]=IC(<Valor>)`

  Posición Punto de inversión = Punto de inversión 1 + valor programado

Ejemplo:

Código de programa

```
N10 OSP1[Z]=100 OSP2[Z]=110
...
...
N40 OSP1[Z]=IC(3)
```

WAITP

En el caso de que deba realizarse un vaivén con un eje geométrico, éste debe habilitarse para el vaivén mediante la función `WAITP`.

Cuando finaliza el vaivén, `WAITP` permite transformar de nuevo el eje de vaivén en un eje de posicionado, de forma que éste pueda volver a ser utilizado normalmente.

Vaivén con acciones síncronas a desplazamientos y tiempos de parada

Tras expirar los tiempos de parada ajustados, en caso de vaivén se realiza el cambio de secuencia interno (el cual es visible en los nuevos trayectos residuales de los ejes). Al cambiar de secuencia se verifica la función de desactivación. Durante esta operación se determina la función de desactivación según la posición de control ajustada para la secuencia de movimientos (`OSCTRL`). *Este comportamiento en el tiempo se puede influir con la corrección del avance.*

Bajo ciertas circunstancias se ejecuta a continuación una carrera de vaivén antes de iniciar las carreras de afinado o de efectuar el desplazamiento a la posición final. *Parece como si se modificara el comportamiento de desactivación. Sin embargo, no es así.*

11.2 Vaivén controlado mediante acciones síncronas (OSCILL)

Función

En este tipo de vaivén solamente se permite realizar una penetración en los puntos de inversión o bien dentro del área definida para la inversión del sentido de desplazamiento.

Según las necesidades, el movimiento de vaivén puede

- continuarse o
- detenerse durante la penetración hasta que ésta se haya ejecutado por completo.

Sintaxis

1. **Definición de los parámetros para el vaivén**
2. **Definición de acciones síncronas a desplazamientos**
3. **Asignación de ejes, definición de la penetración**

Significado

`OSP1[<eje de vaivén>]=`	Posición del punto de inversión 1
`OSP2[<eje de vaivén>]=`	Posición del punto de inversión 2
`OST1[<eje de vaivén>]=`	Tiempo de parada en el punto de inversión 1 en segundos
`OST2[<eje de vaivén>]=`	Tiempo de parada en el punto de inversión 2 en segundos
`FA[<eje de vaivén>]=`	Avance para el eje de vaivén
`OSCTRL[<eje de vaivén>]=`	Activación/desactivación de las opciones
`OSNSC[<eje de vaivén>]=`	Número de carreras de afinado
`OSE[<eje de vaivén>]=`	Posición final
`WAITP(<eje de vaivén>)`	Liberación del vaivén para el eje

Asignación de ejes, penetración

```
OSCILL[<eje de vaivén>]=(<eje de penetración 1>,<eje de
penetración 2>,<eje de penetración 3>)
POSP[<eje de penetración>]=(<posición final>,<penetración
parcial>,<modo>)
```

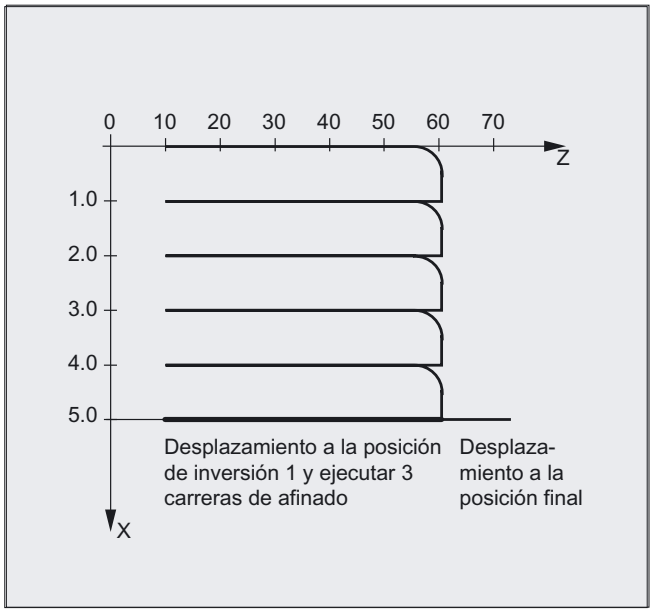
OSCILL:	Asignación de los ejes de penetración a los ejes para el vaivén
POSP:	Definición de las penetraciones parciales/total (ver cap. Administración de programas y ficheros)
Posición final:	Posición final para el eje de penetración tras haber realizado todas las penetraciones parciales.
Penetración parcial:	Valor de la penetración parcial en el punto de inversión/zona de inversión
Modo:	Partición de la penetración total en penetraciones parciales
	= dos pasos residuales del mismo tamaño (estándar);
	= mismo tamaño para todas las penetraciones parciales

Acciones síncronas a desplazamiento

WHEN… … DO	Si..., entonces...
WHENEVER … DO	Siempre que..., entonces...

Ejemplo

En el punto de inversión 1 no se debe realizar penetración. En el punto de inversión 2 se debe realizar la penetración cuando el eje de vaivén se encuentre a una distancia ii2 del punto de inversión 2. El eje de vaivén no deberá esperar en dicho punto a que finalice el desplazamiento de penetración parcial. El eje Z es el eje de vaivén y el eje X el eje de penetración.



1. Parámetros para el vaivén

Código del programa		Comentarios
DEF INT ii2	;	Definición de la variable para la zona de inversión 2.
OSP1[Z]=10 OSP2[Z]=60	;	Definición de los puntos de inversión 1 y 2.
OST1[Z]=0 OST2[Z]=0	;	Punto de inversión 1: parada exacta fina Punto de inversión 2: Parada precisa fina
FA[Z]=150 FA[X]=0.5	;	Avance para el eje de vaivén Z, avance para el eje de penetración X.
OSCTRL[Z]=(2+8+16,1)	;	Desactivar el movimiento de vaivén en el punto de inversión 2; afinar (apagado de chispas) tras borrado del trayecto residual y desplazar a posición final; tras borrado del trayecto residual, desplazar a la posición de inversión correspondiente.
OSNC[Z]=3	;	Carreras de afinado
OSE[Z]=70	;	Posición final = 70
ii2=2	;	Definición del valor para la zona de inversión.
WAITP(Z)	;	Liberar vaivén para el eje Z.

2. Acciones síncronas a desplazamientos

Código del programa	Comentarios
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z]DO -> -> $AA_OVR[X]=0 $AC_MARKER[0]=0	; Siempre que la posición actual del eje de vaivén Z en MKS sea más pequeña que el inicio de la zona de inversión 2, fijar la corrección axial del eje de penetración X a 0% y la marca con el índice 0 al valor 0.
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0	; Siempre que la posición actual del eje de vaivén Z en MKS sea igual de grande que la posición de inversión 2, fijar la corrección axial del eje de vaivén Z a 0%.
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0]=1	; Siempre que el trayecto residual sea igual a la penetración parcial, fijar la marca con índice 0 al valor 1.
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100	; Siempre que la marca sea igual al índice 0, fijar la corrección axial del eje de penetración X a 0%; de esta manera se evita una penetración demasiado temprana (el eje de vaivén Z aún no ha salido de la zona de inversión 2, el eje de penetración X ya está preparado para una nueva penetración); fijar la corrección axial del eje de vaivén Z a 100% (de esta manera se cancela la 2ª acción síncrona).

-> se debe programar en una secuencia

3. Iniciar vaivén

Código del programa	Comentarios
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; Iniciar los ejes El eje X se asigna al eje de vaivén Z como eje de penetración. El eje X debe desplazarse hasta la posición final 5 en pasos de 1.
M30	; Fin del programa

Descripción

1. **Definición de los parámetros para el vaivén**
   Los parámetros para el vaivén se deben definir antes de la secuencia de desplazamiento que contiene la asignación del eje de vaivén al eje de penetración así como el ajuste de la penetración (ver "Vaivén asíncrono").
2. **Definir acciones síncronas de movimiento**
   A través de condiciones síncronas se realizan las siguientes operaciones:
   Suprimir la penetración hasta que el eje de vaivén se encuentre dentro de una zona de inversión (ii1, ii2) o bien en un punto de inversión (U1, U2).
   Parar el movimiento de vaivén durante la penetración en el punto de inversión.
   Iniciar el movimiento de vaivén al finalizar una penetración parcial. Determinar **el inicio de la siguiente penetración parcial**.
3. **Definición de la asignación del eje de vaivén y eje de penetración** así como **de la penetración total y parcial**.

Definición de los parámetros para el vaivén

Asignación del eje de vaivén y del eje de penetración: OSCILL

```
OSCILL[Eje de vaivén] = (Eje de penetración1, Eje de penetración2,
Eje de penetración3)
```

El comando `OSCILL` permite definir la asignación de los ejes y el inicio del movimiento de vaivén.

Como máximo se pueden asignar 3 ejes de penetración a un eje de vaivén.

Nota

Antes del inicio de un movimiento de vaivén deben definirse las condiciones síncronas para el comportamiento de los ejes.

Definir penetraciones: POSP

```
POSP[Eje de penetración] = (Pos. final, Longitud parcial, Modo)
```

El comando `POSP` permite definir en el CN la siguiente información:

- La penetración total (respecto a la posición final)
- El tamaño de la penetración parcial correspondiente en el punto de inversión o bien en la zona de inversión
- El comportamiento de la penetración parcial al alcanzar la posición final (mediante el modo)

Modo = 0	La distancia restante hasta el punto final se divide en dos partes iguales para la penetración parcial (ajuste estándar).
Modo = 1	Las penetraciones parciales se definen del mismo tamaño. Se calculan a partir de la penetración total.

Definir acciones síncronas a desplazamientos

La siguiente lista de acciones síncronas a desplazamientos se utilizan para movimientos de vaivén en general.

Se ofrecen soluciones ejemplares para requisitos individuales que servirán como módulos para la creación de movimientos de vaivén específicos del usuario.

Nota

En casos individuales se pueden programar las condiciones síncronas de forma diferente.

Palabras reservadas

`WHEN … DO …`	si... entonces...
`WHENEVER … DO`	siempre que... entonces...

Funciones

Mediante los comandos descritos detalladamente más adelante se pueden realizar las siguientes funciones:

1. Penetración en el punto de inversión.
2. Penetración en la zona de inversión.
3. Penetración en ambos puntos de inversión.
4. Parada del movimiento de vaivén en el punto de inversión.
5. Rearranque del movimiento de vaivén.
6. No arrancar la penetración parcial demasiado pronto.

Para todas las acciones ejemplares anteriormente descritas se supone:

- Punto de inversión 1 < punto de inversión 2
- Z = eje de vaivén
- X = eje de penetración

Nota

Para explicaciones más detalladas, ver el capítulo Acciones síncronas a desplazamientos.

Definición de la asignación del eje de vaivén y eje de penetración así como de la penetración total y parcial

Penetración en la zona de inversión

El movimiento de penetración se deberá realizar en la zona de inversión antes de que se alcance el punto de inversión.

Esta acción síncrona evita que se realice el movimiento de penetración hasta que el eje de vaivén se encuentre en la zona de inversión.

A partir de los supuestos anteriores (ver arriba) se deducen las siguientes instrucciones:

Zona de inversión 1:

```
WHENEVER
$AA_IM[Z]>$SA_OSCILL_
RESERVE_POS1[Z]+ii1 –
DO $AA_OVR[X] = 0
```

Siempre que la posición actual del eje de vaivén en MKS sea más grande que el inicio de la zona de inversión 1, fijar la corrección axial del eje de penetración a 0%.

Zona de inversión 2:

```
WHENEVER
$AA_IM[Z]<$SA_OSCILL_
RESERVE_POS2[Z]+ii2 –
DO $AA_OVR[X] = 0
```

Siempre que la posición actual del eje de vaivén en MKS sea más pequeña que el inicio de la zona de inversión 2, fijar la corrección axial del eje de penetración a 0%.

Penetración en el punto de inversión

Hasta que el eje de vaivén no alcance el punto de inversión no se realiza ningún desplazamiento del eje de penetración.

A partir de los supuestos anteriores (ver arriba) se deducen las siguientes instrucciones:

Zona de inversión 1:

```
WHENEVER
$AA_IM[Z]<>$SA_OSCILL
 RESERVE_POS1[Z] DO
$AA_OVR[X] = 0 →
→ $AA_OVR[Z] = 100
```

Siempre que la posición actual del eje de vaivén Z en MKS sea más grande o más pequeña que la posición del punto de inversión 1, fijar la corrección axial del eje de penetración X a 0% y la corrección axial del eje de vaivén Z en 100%.

Zona de inversión 2:

Para el punto de inversión 2:

```
WHENEVER
$AA_IM[Z]<>$SA_OSCILL
 RESERVE_POS2[Z] DO
$AA_OVR[X] = 0 →
→ $AA_OVR[Z] = 100
```

Siempre que la posición actual del eje de vaivén Zu en MKS sea más grande o más pequeña que la posición del punto de inversión 2, fijar la corrección axial del eje de penetración X a 0% y la corrección axial del eje de vaivén Z en 100%.

Parada del movimiento de vaivén en el punto de inversión

El eje de vaivén se detiene en el punto de inversión y, simultáneamente, comienza el desplazamiento de penetración. El desplazamiento de vaivén continúa cuando haya finalizado completamente el desplazamiento de penetración.

Esta acción síncrona se puede utilizar simultáneamente para arrancar el desplazamiento de penetración, en el caso de que éste hubiera sido detenido por una acción síncrona anterior aún activa.

A partir de los supuestos anteriores (ver arriba) se deducen las siguientes instrucciones:

Zona de inversión 1:

```
WHENEVER
$SA_IM[Z]==$SA_OSCIL
L_RESERVE_POS1[Z] DO
$AA_OVR[X] = 0 →
→ $AA_OVR[Z] = 100
```

Siempre que la posición actual del eje de vaivén en MKS sea igual que la posición de inversión 1, fijar la corrección axial del eje de vaivén a 0% y la corrección axial del eje de penetración en 100%.

Zona de inversión 2:

```
WHENEVER
$SA_IM[Z]==$SA_OSCIL
L_RESERVE_POS2[Z] DO
$AA_OVR[X] = 0 →
→ $AA_OVR[Z] = 100
```

Siempre que la posición actual del eje de vaivén Zu en MKS sea igual que la posición de inversión 2, fijar la corrección axial del eje de vaivén X a 0% y la corrección axial del eje de penetración en 100%.

Evaluación online del punto de inversión

Cuando a la derecha de la comparación se encuentra una variable de proceso principal precedida por los caracteres `$$`, entonces se realiza la evaluación y comparación de las dos variables en cada tiempo de ciclo IPO.

Nota

Para más información al respecto ver capítulo "Acciones síncronas a desplazamientos".

Rearranque del movimiento de vaivén

Esta acción síncrona se utiliza para rearrancar el desplazamiento de vaivén cuando se ha realizado una penetración parcial.

A partir de los supuestos anteriores (ver arriba) se deducen las siguientes instrucciones:

```
WHENEVER
$AA_DTEPW[X]==0 DO
$AA_OVR[Z] = 100
```

Siempre que el trayecto residual para la penetración parcial del eje de penetración X en WKS sea igual a cero, fijar la corrección axial del eje de vaivén a 100%.

Siguiente penetración parcial

Cuando se ha finalizado la penetración, debe evitarse un arranque prematuro de la siguiente penetración parcial.

Para ello se utiliza la marca específica de canal (`$AC_MARKER[Index]`) que se activa al final de la penetración parcial (trayecto residual ≡ 0) y que se desactiva al abandonar la zona de inversión. Entonces se inhibe el siguiente desplazamiento de penetración mediante una acción síncrona.

Las suposiciones anteriores (ver arriba) dan como resultado, p. ej., las siguientes instrucciones para el punto de inversión 1:

1. Activación de la marca:

```
WHENEVER
$AA_DTEPW[X]==0 DO
$AC_MARKER[1]=1
```

Siempre que el trayecto residual para la penetración parcial del eje de penetración X en WKS sea igual a cero, fijar la marca con índice 1 a 1.

2. Desactivación de la marca:

```
WHENEVER $AA_IM[Z]<>
$SA_OSCILL_RESERVE_P
OS1[Z] DO
$AC_MARKER[1] = 0
```

Siempre que la posición actual del eje de vaivén Z en MKS sea más grande o más pequeña que la posición del punto de inversión 1, fijar la marca 1 a 0.

3. Inhibición de la penetración:

```
WHENEVER
$AC_MARKER[1]==1 DO
$AA_OVR[X]=0
```

Siempre que la marca sea igual a 1, fijar la corrección axial del eje de penetración X a 0%.

Troquelado y punzonado

12

12.1 Activación, desactivación

12.1.1 Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Función

Activación/desactivación del troquelado o el punzonado

PON y SON permiten activar la función de troquelado o punzonado. SPOF finaliza todas las funciones específicas de troquelado y punzonado. Los comandos modales PON y SON se excluyen mutuamente, es decir que PON desactiva a SON y viceversa.

Troquelado/punzonado controlado a nivel de servo

Las funciones SONS y PONS activan también las funciones de troquelado y punzonado.

Al contrario que el mando de carrera en plano de interpolación activo con SON/PON, en estas funciones el mando de señal del disparo de carrera se realiza a nivel servo. De este modo, se consiguen mayores frecuencias de punzonado, así como un mayor rendimiento para el troquelado.

Durante la evaluación de las señales se bloquean todas las funciones que puedan provocar un cambio de posición de los ejes de punzonado o troquelado (p. ej., desplazamiento con volante, modificaciones de frames vía PLC, funciones de medida).

Troquelado con retardo

PDELAYON genera una salida con retraso del punzón. El comando modal realiza una preparación y generalmente se escribe delante de PON. Tras PDELAYOF se prosigue con el troquelado con normalidad.

Nota

La temporización de retardo se ajusta en el dato de operador DO42400 $SC_PUNCH_DWELLTIME.

Aceleración dependiente del recorrido

Con PUNCHACC puede establecerse una característica de aceleración que defina diferentes aceleraciones en función de la distancia entre agujeros.

Segunda interfaz de troquelado

Las máquinas que deban utilizar alternativamente una segunda interfaz de troquelado (segunda unidad de troquelado o un medio equiparable), pueden cambiar a un segundo par de entradas y salidas digitales rápidas del control (par E/S). En ambas interfaces de troquelado pueden usarse todas las funciones de troquelado/punzonado. La conmutación entre la primera y la segunda interfaz de troquelado se lleva a cabo por medio de los comandos `SPIF1` y `SPIF2`.

Nota

Requisito: debe haberse definido mediante los datos de máquina un segundo par de E/S para las funciones de troquelado (→ ver datos del fabricante de la máquina).

Sintaxis

```
PON G... X... Y... Z...
SON G... X... Y... Z...
SONS G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYON
PDELAYOF
PUNCHACC(<Smin>,<Amin>,<Smax>,<Amax>)
SPIF1/SPIF2
SPOF
```

Descripción

`PON`	Activar el troquelado
`SON`	Activar el punzonado
`PONS`	Activar el troquelado controlado a nivel de servo
`SONS`	Activar el punzonado controlado a nivel de servo
`SPOF`	Desactivar el troquelado/punzonado
`PDELAYON`	Activar el troquelado con retardo
`PDELAYOF`	Desactivar el troquelado con retardo
`PUNCHACC`	Activar la aceleración dependiente del recorrido
	Parámetros:
	`<Smin>` Menor distancia entre agujeros
	`<Amin>` Aceleración inicial
	`<Amin>` puede ser más grande que `<Amax>`.
	`<Smax>` Mayor distancia entre agujeros
	`<Amax>` Aceleración final
	`<Amax>` puede ser mayor que `<Amin>`.
`SPIF1`	Activar la **primera** interfaz de troquelado
	El mando de carrera tiene lugar a través del primer par de E/S rápidas.

SPIF2 Activar la **segunda** interfaz de troquelado

El mando de carrera tiene lugar a través del segundo par de E/S rápidas.

Nota:
Después de un RESET o de un arranque del control siempre está activa la primera interfaz de troquelado. Si sólo se utiliza una interfaz de troquelado, no es necesario programarla.

Ejemplos

Ejemplo 1: Activación del punzonado

```
Código de programa              Comentarios
...
N70 X50 SPOF                    ; Posicionamiento sin disparo de troquelado.
N80 X100 SON                    ; Activación del punzonado, disparo de una carrera
                                  antes del movimiento (X = 50) y al acabar el
                                  movimiento programado (X = 100).
...
```

Ejemplo 2: Troquelado con retardo

```
Código de programa              Comentarios
...
N170 PDELAYON X100 SPOF         ; Posicionamiento sin disparo de troquelado,
                                  activación del disparo retardado de troquelado.
N180 X800 PON                   ; Activación del troquelado. Tras alcanzar la posi-
                                  ción final, la carrera de troquelado se realiza con
                                  retraso.
N190 PDELAYOF X700              ; Desactivación del troquelado con retardo, disparo
                                  de troquelado normal al acabar el movimiento
                                  programado.
...
```

Ejemplo 3: Troquelado con dos interfaces de troquelado

```
Código de programa              Comentarios
...
N170 SPIF1 X100 PON             ; Al final de la secuencia se produce un disparo de
                                  carrera en la primera salida rápida. La señal
                                  "Carrera activa" se vigila en la primera entrada.
N180 X800 SPIF2                 ; El segundo disparo de carrera se produce en la
                                  segunda salida rápida. La señal "Carrera activa" se
                                  vigila en la segunda entrada.
N190 SPIF1 X700                 ; El mando de carrera para todas las demás carreras
                                  tiene lugar con la primera interfaz.
...
```

Información adicional

Punzonado y troquelado controlados a nivel de servo (PONS/SONS)

El troquelado y el punzonado controlados a nivel de servo no se pueden realizar simultáneamente en varios canales. PONS y SONS sólo pueden activarse en un canal a la vez.

Aceleración dependiente del recorrido (PUNCHACC)

Ejemplo:
```
PUNCHACC(2,50,10,100)
```

Distancias entre agujeros inferiores a 2 mm:

El desplazamiento se efectúa con una aceleración del 50 % de la aceleración máxima.

Distancias entre agujeros de 2 mm a 10 mm:

La aceleración aumenta proporcionalmente a la distancia al 100 %.

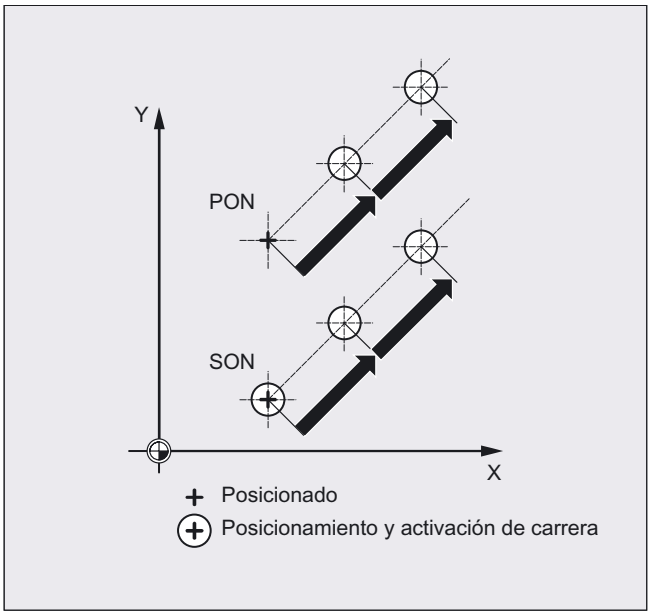
Distancias entre agujeros superiores a 10 mm:

Desplazamiento con una aceleración del 100 %.

Activación de la primera carrera del punzón

La activación de la primera carrera del punzón tras activar la función se realiza para los procesos de punzonado y troquelado en distinto momento.

- PON/PONS:
  - Todas las carreras – también en la primera secuencia después de la activación – se ejecutan al final de la secuencia.
- SON/SONS:
  - La primera carrera para el punzón se realiza tras activar el punzonado, ya al principio de la secuencia.
  - El resto de carreras se realizan al final de las secuencias.



Punzonado y troquelado en el sitio

La activación del punzón se realiza sólo cuando la secuencia contenga información sobre el desplazamiento para los ejes de punzonado o troquelado (ejes del plano activo).

Sin embargo, para disparar una carrera en el mismo sitio, debe programarse uno de los ejes de troquelado/punzonado con un recorrido 0.

Trabajos con herramientas giratorias

Nota

Se puede utilizar el control tangencial para conseguir que una herramienta giratoria se posicione de forma tangencial a la trayectoria programada.

Uso de comandos M

Gracias a la técnica de macros sigue siendo posible, como hasta ahora, utilizar funciones M especiales en lugar de instrucciones de programación (compatibilidad). En este caso se aplican las siguientes correspondencias con los sistemas antiguos:

```
M20, M23   ≙  SPOF
M22        ≙  SON
M25        ≙  PON
M26        ≙  PDELAYON
```

Ejemplo de fichero de macros:

Código de programa	**Comentarios**
DEFINE M25 AS PON	; Troquelado CON
DEFINE M125 AS PONS	; Troquelado controlado a nivel de servo CON
DEFINE M22 AS SON	; Punzonado CON
DEFINE M122 AS SONS	; Punzonado controlado a nivel de servo CON
DEFINE M26 AS PDELAYON	; Troquelado con retardo CON
DEFINE M20 AS SPOF	; Troquelado, punzonado DES
DEFINE M23 AS SPOF	; Troquelado, punzonado DES

Ejemplo de programación:

Código de programa	**Comentarios**
...	
N100 X100 M20	; Posicionamiento sin disparo de troquelado.
N110 X120 M22	; Activación del punzonado, disparo de carrera antes y después del movimiento.
N120 X150 Y150 M25	; Activación del troquelado, disparo de carrera al terminar el movimiento.
...	

12.2 División automática en segmentos

Función

Subdivisión en segmentos

Con el punzonado o el troquelado activados, SPP y también SPN ocasionan una división del desplazamiento total programado para los ejes de trayectoria en una cantidad de segmentos de igual longitud (división equidistante de la trayectoria). De manera interna, cada segmento corresponde a una secuencia.

Cantidad de carreras

En troquelado, la primera carrera del punzón se realiza al final del primer segmento, al contrario que en punzonado, donde la primera carrera del punzón se realiza al inicio del primer segmento. A lo largo de todo el desplazamiento se obtienen las siguientes cifras:

Troquelado: Cantidad de carreras = Cantidad de segmentos

Punzonado: Cantidad de carreras = Cantidad de segmentos + 1

Funciones auxiliares

Las funciones auxiliares se ejecutan en la primera de las secuencias generadas.

Sintaxis

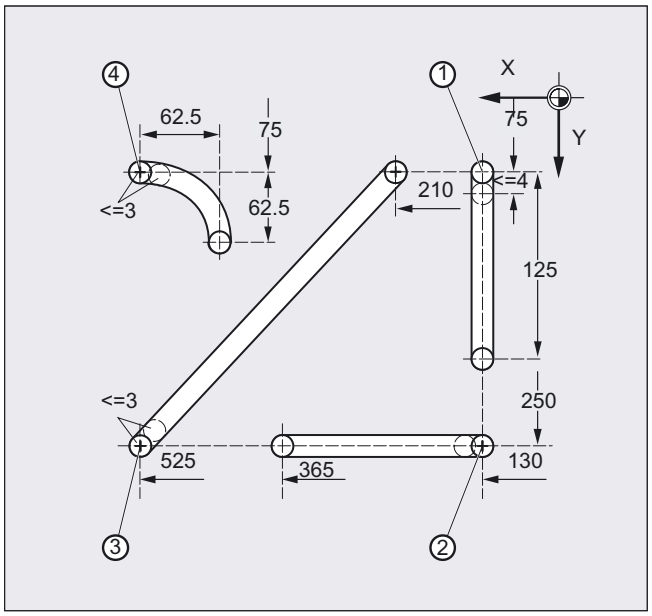
```
SPP=
SPN=
```

Descripción

SPP	Tamaño de los segmentos (distancia máxima entre punzonados), válido de forma modal
SPN	Cantidad de segmentos por secuencia; efectivos secuencia a secuencia

Ejemplo 1

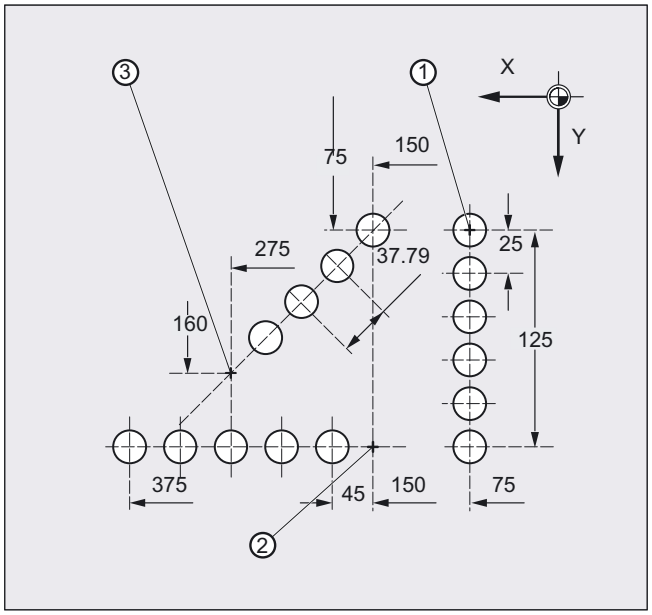
Los segmentos de punzonado programados deben dividirse automáticamente en segmentos de igual tamaño.



Código del programa	Comentarios
N100 G90 X130 Y75 F60 SPOF	; Posicionar en el punto de inicio 1
N110 G91 Y125 SPP=4 SON	; Punzonado activado; longitud de segmento máxima para segmentación automática del trayecto: 4 mm
N120 G90 Y250 SPOF	; Punzonado desactivado; posicionado en el punto inicial 2
N130 X365 SON	; Punzonado activado; longitud de segmento máxima para segmentación automática del trayecto: 4 mm
N140 X525 SPOF	; Punzonado desactivado; posicionado en el punto inicial 3
N150 X210 Y75 SPP=3 SON	; Punzonado activado; longitud de segmento máxima para segmentación automática del trayecto: 3 mm
N160 X525 SPOF	; Punzonado desactivado; posicionado en el punto inicial 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; Punzonado activado; longitud de segmento máxima para segmentación automática del trayecto: 3 mm
N180 G00 G90 Y300 SPOF	; Punzonado desactivado

Ejemplo 2

Para las filas de agujeros individuales debe realizarse una segmentación automática del trayecto. Para la división se indica respectivamente la longitud de segmento máxima (valor SPP).



Código del programa	**Comentarios**
N100 G90 X75 Y75 F60 PON	; Posicionar en el punto inicial 1; troquelado de agujero individual activado
N110 G91 Y125 SPP=25	; Longitud de segmento máxima para segmentación automática del trayecto: 25 mm
N120 G90 X150 SPOF	; Troquelado desactivado; posicionado en el punto inicial 2
N130 X375 SPP=45 PON	; Troquelado activado; longitud de segmento máxima para segmentación automática del trayecto: 45 mm
N140 X275 Y160 SPOF	; Troquelado desactivado; posicionado en el punto inicial 3
N150 X150 Y75 SPP=40 PON	; Troquelado activado; en lugar de la longitud de segmento programada de 40 mm ;se utiliza la longitud de segmento calculada de 37,79 mm.
N160 G00 Y300 SPOF	; Troquelado desactivado; posicionar

12.2.1 División en ejes de contorneado

Longitud de los segmentos SPP

Con `SPP` se indica la distancia máxima entre punzonados y, en consecuencia, la longitud máxima de los segmentos en los cuales se dividirá el recorrido total. La desactivación de la función se realiza mediante `SPOF` o bien `SPP=0`.

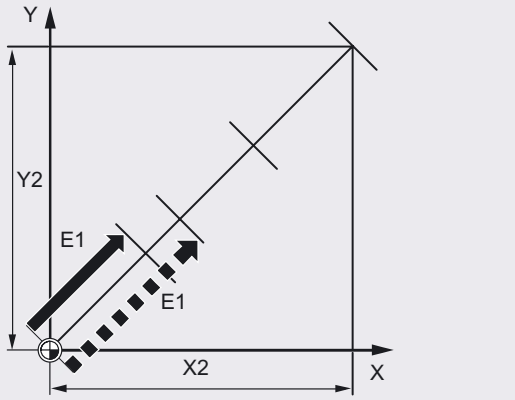
Ejemplo:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

El recorrido total de 10 mm se divide en 5 segmentos de 2 mm cada uno(`SPP=2`).

Nota

La división en segmentos con `SPP` se realiza siempre de forma equidistante: todos los segmentos tienen la misma longitud. Es decir, solamente es válido el valor de longitud programado para los segmentos cuando el cociente entre el trayecto total programado y el valor `SPP` es un número entero. En caso contrario se reduce internamente el número de segmentos, de manera que dicho cociente resulte un número entero.



X2/Y2 Recorrido programado (secuencia de punzonado o troquelado)
E1 Longitud de segmento programada
E1' Longitud de segmento redondeada automáticamente

Ejemplo:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

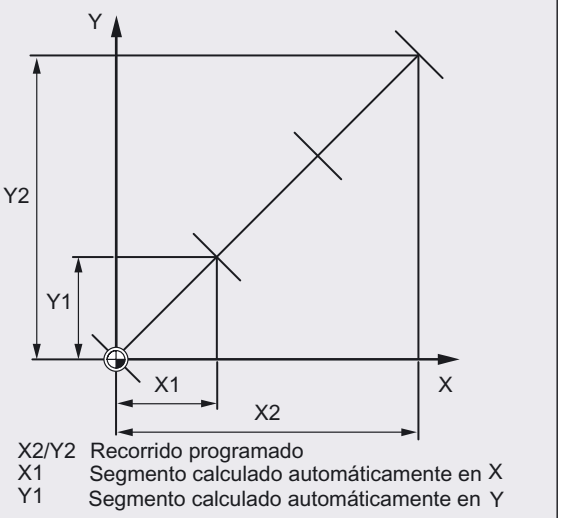
En un trayecto de 15 mm con una longitud de segmentos de 3,5 mm, el cociente es de (4,28). Como consecuencia se realiza una reducción de forma automática del valor SPP hasta el cociente de un número entero. En este caso, se obtiene una longitud para los segmentos de 3 mm.

Número de segmentos SPN

Con la función SPP se define el número de segmentos en los que se desea dividir la trayectoria programada. La longitud de los segmentos se calcula de forma automática. La función SPP es válida de forma secuencial, por lo cual se debe activar antes de troquelar o punzonar con PON o SON.

Aplicación de SPP y SPN en una misma secuencia

En el caso de que se programe en una misma secuencia la longitud de los segmentos (SPP) y la cantidad de los mismos (SPN), se tendrá en cuenta para dicha secuencia la función SPN y, para el resto de secuencias, la función SPP. Si se encuentra activa la función SPP antes de programar SPN, tras ejecutar dicha secuencia, vuelve a ser válida SPN.



Nota

Si las funciones de punzonado y troquelado están incluidas dentro de la funcionalidad básica del control, es posible programar dichas funciones de división en segmentos con SPN o SPP independientemente de la utilización de dichas tecnologías.

12.2.2 División en ejes individuales

Si se han definido como ejes de punzonado/troquelado ejes individuales adicionalmente a los ejes de la trayectoria, también se puede utilizar para éstos la división automática en segmentos.

Comportamiento de ejes individuales con la función SPP

La longitud programada para los segmentos (`SPP`) se refiere a los ejes de la trayectoria. Por lo tanto, se ignora la función SPP en una secuencia en la que, adicionalmente al desplazamiento del eje individual y el valor SPP, no se haya programado ningún eje de la trayectoria.

Si se programan ejes de contorneado conjuntamente con ejes individuales en una misma secuencia, el comportamiento del eje individual se determina por el ajuste realizado en los correspondientes datos de máquina.

1. Ajuste por defecto

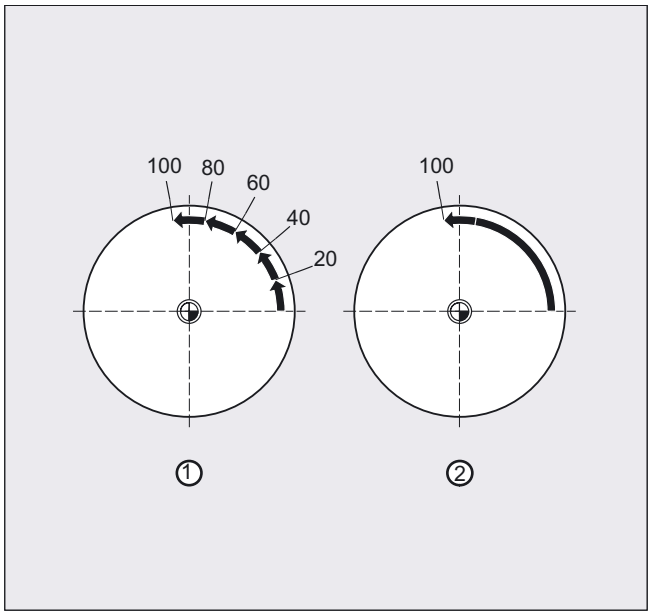
   La trayectoria del eje individual se divide uniformemente mediante las secuencias intermedias generadas por la función `SPP`.

Ejemplo:

```
N10 G1 SON X10 A0
N20 SPP=3 X25 A100
```

Debido a que los segmentos se han definido de 3 mm, se realizarán en total 5 secuencias a lo largo del desplazamiento total de 15 mm en el eje X (eje de trayectoria).

El eje A realiza así un giro de 20° por secuencia.



1. Eje individual sin división en segmentos

   El eje individual realiza su recorrido total en la primera secuencia generada.

2. División en segmentos diferente

   El comportamiento del eje individual depende de la interpolación de los ejes de contorneado:

- Interpolación circular: con división de la trayectoria
- Interpolación lineal: sin división de la trayectoria.

Comportamiento con SPN

El número de segmentos programado también es válido cuando no se ha programado simultáneamente un eje de contorneado.

Requisito: el eje individual se ha definido como eje de punzonado/troquelado.

Rectificado

13

13.1 Vigilancia de herramienta específica de rectificado desde el programa de pieza (TMON, TMOF)

Funcionamiento

El comando TMON se utiliza para activar la vigilancia de geometría y velocidad para muelas (herramientas tipo 400 - 499) desde el programa de pieza CN. La vigilancia permanece activa hasta que se desactiva desde el programa de pieza mediante el comando TMOF.

Nota

¡Observar las indicaciones del fabricante de la máquina!

Requisito

Los parámetros de herramienta específicos de rectificado $TC_TPG1 a $TC_TPG9 deben estar fijados.

Sintaxis

```
TMON(<Nº T>)
TMOF(<Nº T>)
```

Descripción

TMON	Comando para **Con**ectar la vigilancia de herramienta específica de rectificado
TMOF	Comando para **Des**conectar la vigilancia de herramienta específica de rectificado
<Nº T>	Indicación del número T **Nota:** Sólo es necesario cuando no está activa la herramienta con este número T.
TMOF(0)	Desactivar vigilancia para todas las herramientas

Información adicional

Parámetros de herramienta específicos de rectificado

Parámetros	Significado	Tipo de datos
$TC_TPG1	Número de cabezal	INT
$TC_TPG2	Regla de concatenación Los parámetros se mantienen idénticos automáticamente para las caras izquierda y derecha de la muela.	INT
$TC_TPG3	Radio mínimo de la muela	REAL
$TC_TPG4	Ancho mínimo de la muela	REAL
$TC_TPG5	Ancho actual de la muela	REAL
$TC_TPG6	Velocidad de giro máxima	REAL
$TC_TPG7	Velocidad periférica máxima	REAL
$TC_TPG8	Ángulo para la muela inclinada	REAL
$TC_TPG9	Número de parámetro para el cálculo del radio	INT

Bibliografía:
Manual de funciones básicas; Corrección de herramientas (W1)

Activar la vigilancia de herramienta a través de la selección de herramienta

La vigilancia de la herramienta se activa de forma implícita con tipos de herramienta impares (tipos impares comprendidos entre 400 y 499).

Solamente se puede activar **una** vigilancia de forma simultánea por cada cabezal.

Vigilancia de la geometría

Se vigilan el radio y ancho actuales de la muela actual.

La vigilancia de la velocidad de consigna respecto al rebase de límites se realiza de forma cíclica teniendo en cuenta la corrección para el cabezal.

Como velocidad de giro límite se toma el menor valor, resultante de la comparación entre la velocidad máxima y la velocidad calculada a partir de la máxima velocidad periférica de la muela y del radio actual de la muela.

Mecanizado sin números T y D

Mediante dato de máquina pueden ajustarse un número **T** estándar y un número **D** estándar

que ya no deben programarse y que se activan tras Power On/Reset.

Ejemplo: Rectificado con la misma muela.

A través del dato de máquina se puede ajustar que la herramienta activa permanezca con Reset (véase"Asignación libre de números D, número de filo (Página 454)").

Funciones adicionales 14

14.1 Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

Función

AXNAME se utiliza, p. ej., para la creación de ciclos de validez general, cuando los nombres de los ejes se desconocen.

AX se utiliza para la programación indirecta de ejes geométricos y síncronos. El identificador de eje se guarda en una variable del tipo AXIS o se suministra mediante un comando como AXNAME o SPI.

SPI se utiliza cuando se programan funciones de eje para un cabezal, p. ej. un cabezal síncrono.

AXTOSPI se utiliza para convertir un identificador de eje en un índice de cabezal (función inversa de SPI).

AXSTRING se utiliza para convertir un identificador de eje (tipo de datos AXIS) en un string (función inversa de AXNAME).

ISAXIS se aplica a los ciclos de propósito general para garantizar que se dispone de determinados ejes geométricos y evitar que se interrumpa con error la llamada siguiente de $P_AXNX.

MODAXVAL se utiliza para determinar la posición módulo en ejes giratorios con módulo.

Sintaxis

```
AXNAME("String")
AX[AXNAME("String")]
SPI(n)
```

AXTOSPI(A), AXTOSPI(B) o AXTOSPI(C)

```
AXSTRING(SPI(n))
ISAXIS(<Número de eje geométrico>)
<Posición módulo>=MODAXVAL(<Eje>,<Posición de eje>)
```

Descripción

AXNAME	Conversión de una cadena de caracteres en identificador de eje; la cadena de caracteres debe estar constituida por un nombre de un eje válido.
AX	Identificador de eje variable
SPI	Conversión de un número de cabezal en identificador de eje; el parámetro de transferencia debe ser un número de cabezal válido.
n	Número de cabezal
AXTOSPI	Convierte un identificador de eje en un índice de cabezal del tipo Integer. AXTOSPI es la función inversa de SPI.
X, Y, Z	Identificador de eje del tipo AXIS como variable o constante
AXSTRING	Se emite la cadena con el número de cabezal correspondiente.
ISAXIS	Comprobar si existe el eje geométrico introducido.
MODAXVAL	Determina la posición módulo en ejes giratorios con módulo; esta posición equivale al resto de módulo respecto a la zona de módulo parametrizada (entre 0 y 360 grados en el ajuste estándar; mediante DM30340 MODULO_RANGE_START y DM30330 $MA_MODULO_RANGE pueden modificarse el inicio y el tamaño de la zona de módulo).

Nota

Ampliaciones SPI

La función de eje SPI(n) también puede utilizarse para la lectura y la escritura de componentes frame. Ello permite, por ejemplo, escribir frames con la sintaxis `$P_PFRAME[SPI(1),TR]=2.22`.

Mediante la programación adicional de posiciones de eje a través de la dirección `AX[SPI(1)]=<Posición de eje>` puede desplazarse un eje. El requisito para ello es que el cabezal se encuentre en modo de posicionamiento o de eje.

Ejemplos

Ejemplo 1: AXNAME, AX, ISAXIS

```
Código del programa                          Comentarios
OVRA[AXNAME("eje refrentado")]=10            ; Corrección para el eje transversal
AX[AXNAME("eje refrentado")]=50.2            ; Posición final para el eje
                                               transversal
OVRA[SPI(1)]=70                              ; Corrección para cabezal 1
AX[SPI(1)]=180                               ; Posición final para cabezal 1
IF ISAXIS(1) == FALSE GOTOF CONTINUAR        ; ¿Existe la abscisa?
AX[$P_AXN1]=100                              ; Desplazar en abscisas
CONTINUAR:
```

Ejemplo 2: AXSTRING

En la programación con AXSTRING[SPI(n)] ya no se emite el índice del eje asignado al cabezal como número de cabezal, sino la cadena `"Sn"`.

Código del programa	Comentarios
`AXSTRING[SPI(2)]`	`; Se emite la cadena "S2".`

Ejemplo 3: MODAXVAL

Debe determinarse la posición de módulo del eje giratorio A con módulo.

El valor inicial para el cálculo es la posición de eje 372.55.

La zona de módulo parametrizada va de los 0 hasta los 360 grados:

DM30340 MODULO_RANGE_START = 0

DM30330 $MA_MODULO_RANGE = 360

Código del programa	Comentarios
`R10=MODAXVAL(A,372.55)`	`; Posición módulo calculada R10 = 12.55.`

Ejemplo 4: MODAXVAL

Cuando el identificador de eje programado no se refiere a un eje giratorio con módulo, entonces se devuelve el valor a convertir (<Posición de eje>) sin modificar.

Código del programa	Comentarios
`R11=MODAXVAL(X,372.55)`	`; X es el eje lineal; R11 = 372.55.`

14.2 Ejes geométricos conmutables (GEOAX)

Función

Con la función "Ejes geométricos conmutables" se puede modificar el conjunto de ejes geométricos configurado mediante datos de máquina desde el programa de pieza. Así, un eje de canal definido como eje adicional síncrono puede sustituir a cualquier eje geométrico.

Sintaxis

```
GEOAX(<n>,<Eje de canal>,<n>,<Eje de canal>,<n>,<Eje de canal>)
GEOAX()
```

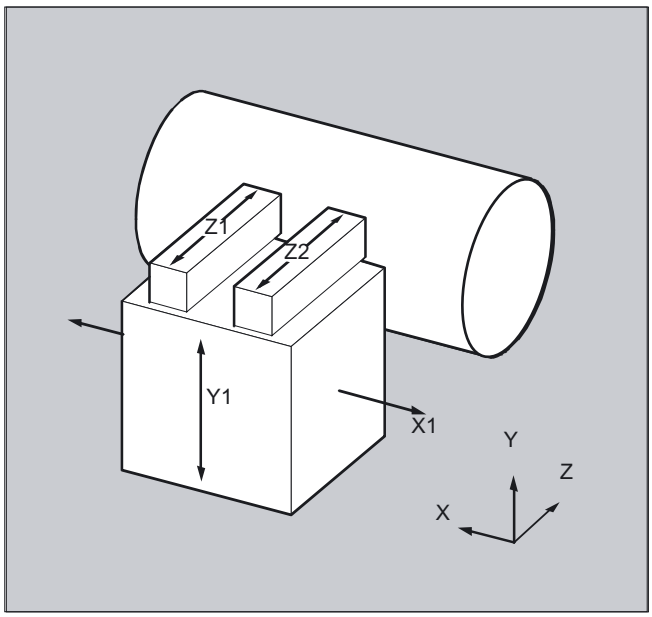
Descripción

`GEOAX(...)`	Comando para conmutar los ejes geométricos **Nota:** `GEOAX()` sin indicación de parámetros llama la configuración básica de los ejes geométricos.
`<n>`	Con este parámetro se indica el número del eje geométrico al que debe asignarse el eje de canal indicado a continuación. Margen de valores: 1, 2 ó 3 **Nota:** Con **<n>**=0 puede suprimirse sin sustitución el eje de canal indicado a continuación del conjunto de ejes geométricos.
`<Eje de canal>`	Con este parámetro se indica el nombre del eje de canal que debe incluirse en el conjunto de ejes geométricos.

Ejemplos

Ejemplo 1: Activar dos ejes alternativamente como eje geométrico

Un carro de herramienta se puede desplazar a través de los ejes de canal X1, Y1, Z1, Z2:



Los ejes geométricos están configurados de tal manera que, tras la activación, primero es efectivo Z1 como 3er eje geométrico con el nombre de eje geométrico "Z", que forma junto con X1 e Y1 el conjunto de ejes geométricos.

En el programa de pieza deben entrar ahora en acción los ejes Z1 y Z2 alternativamente como eje geométrico Z:

```
Código del programa      Comentarios
...
N100 GEOAX(3,Z2)         ; Como 3er eje geométrico (Z) actúa el eje de canal Z2.
N110 G1 ...
N120 GEOAX(3,Z1)         ; Como 3er eje geométrico (Z) actúa el eje de canal Z1.
...
```

Ejemplo 2: Conmutación de los ejes geométricos en 6 ejes de canal

Una máquina tiene 6 ejes de canal con los nombres XX, YY, ZZ, U, V y W.

El ajuste inicial de la configuración de ejes geométricos a través de datos de máquina es el siguiente:

Eje de canal XX = 1er eje geométrico (eje X)

Eje de canal YY = 2º eje geométrico (eje Y)

Eje de canal ZZ = 3er eje geométrico (eje Z)

Código del programa	**Comentarios**
N10 GEOAX()	; La configuración básica de los ejes geométricos es efectiva.
N20 G0 X0 Y0 Z0 U0 V0 W0	; Todos los ejes en Rápido en la posición 0.
N30 GEOAX(1,U,2,V,3,W)	; El eje de canal U será el primer eje geométrico (X), el V el segundo (Y) y el W el tercero (Z).
N40 GEOAX(1,XX,3,ZZ)	; El eje de canal XX será el primer eje geométrico (X) y el ZZ será el tercero (Z). El eje de canal V se mantiene como segundo eje geométrico (Y).
N50 G17 G2 X20 I10 F1000	; Círculo en el plano X/Y. Se desplazan los ejes de canal XX y V.
N60 GEOAX(2,W)	; El eje de canal W se convierte en segundo eje geométrico (Y).
N80 G17 G2 X20 I10 F1000	; Círculo en el plano X/Y. Se desplazan los ejes de canal XX y W.
N90 GEOAX()	; Vuelta al estado inicial.
N100 GEOAX(1,U,2,V,3,W)	; El eje de canal U será el primer eje geométrico (X), el V el segundo (Y) y el W el tercero (Z).
N110 G1 X10 Y10 Z10 XX=25	; Los ejes de canal U, V y W se desplazan respectivamente a la posición 10. XX como eje adicional se desplaza a la posición 25.
N120 GEOAX(0,V)	; V se elimina del conjunto de ejes geométricos. U y W siguen siendo el primer (X) y tercer eje geométrico (Z). El segundo eje geométrico (Y) se mantiene sin ocupar.
N130 GEOAX(1,U,2,V,3,W)	; El eje de canal U sigue siendo el primer eje geométrico (X), el V será el segundo (Y) y el W sigue siendo el tercero (Z).
N140 GEOAX(3,V)	; V se convierte en el tercer eje geométrico (Z), mientras que W se sobrescribe y por lo tanto se elimina del conjunto de ejes geométricos. El segundo eje geométrico (Y) sigue estando desocupado.

Nota

Configuración de ejes

La asignación entre los ejes geométricos, adicionales, de canal y de máquina, así como la especificación de los nombres de los distintos tipos de eje se determina mediante los datos de máquina siguientes:

DM20050 $MC_AXCONF_GEOAX_ASIGN_TAB (asignación de eje geométrico a eje de canal)

DM20060 $MC_AXCONF_GEOAX_NAME_TAB (nombre de eje geométrico en el canal)

DM20070 $MC_AXCONF_MACHAX_USED (número de eje de máquina válido en el canal)

DM20080 $MC_AXCONF_CHANAX_NAME_TAB (nombre de eje de canal en el canal)

DM10000 $MN_AXCONF_MACHAX_NAME_TAB (nombre del eje de máquina)

DM35000 $MA_SPIND_ASSIGN_TO_MACHAX (asignación del cabezal al eje de máquina)

Bibliografía:
Manual de funciones básicas; Ejes, sistemas de coordenadas, frames (K2)

Restricciones

- La conmutación de los ejes geométricos no es posible con:
  - Transformada activa
  - Interpolación spline activa
  - Corrección de radio de herramienta activa
  - Corrección fina de herramienta activa
- Si el eje geométrico y el eje de canal tienen el mismo nombre, no es posible ningún cambio del correspondiente eje geométrico.
- Ninguno de los ejes implicados en la conmutación debe participar en una acción que pueda durar más allá de los límites de secuencia, como pueden hacerlo, p. ej., los ejes de posicionado del tipo A o los ejes esclavos.
- Con el comando `GEOAX` pueden sustituirse sólo los ejes geométricos ya existentes en el momento de la activación (es decir, no pueden definirse ejes nuevos).
- Un cambio de eje con `GEOAX` durante la preparación de la tabla para el contorno (`CONTPRON`, `CONTDCON`) provoca una alarma.

Condiciones

Estado del eje tras la sustitución

Un eje sustituido mediante la conmutación en el conjunto de ejes geométricos puede programarse como eje adicional tras la operación de conmutación mediante su nombre de eje de canal.

Frames, zonas protegidas, limitaciones de zona de trabajo

Con la conmutación de los ejes geométricos se suprimen todos los frames, zonas protegidas y limitaciones de zona de trabajo.

Coordenadas polares

Un cambio de los ejes geométricos con `GEOAX`, de forma análoga a un cambio de nivel con `G17-G19`, define las coordenadas polares modales al valor 0.

DRF, DO

Un posible decalaje con volante (DRF) o un decalaje de origen externo (DO) sigue siendo efectivo tras la conmutación.

Configuración básica de los ejes geométricos

El comando `GEOAX()` llama la configuración básica del conjunto de ejes geométricos.

Tras POWER ON y al conmutar al modo de operación "Posicionar referencia" se vuelve automáticamente a la configuración básica.

Corrección longitudinal de herramienta

Una corrección longitudinal de herramienta activa sigue siendo efectiva tras la operación de conmutación. Sin embargo, para los ejes geométricos nuevos o cambiados de posición vale como aún no aplicada. En el primer comando de movimiento para estos ejes geométricos, el trayecto resultante se compone por lo tanto de la suma de la corrección longitudinal de herramienta y del trayecto programado.

Los ejes geométricos que mantienen su posición en el conjunto de ejes tras una conmutación, también mantienen su estado en lo que a la corrección longitudinal de herramienta se refiere.

Configuración de eje geométrico durante una transformada activa

La configuración de ejes geométricos válida en una transformada activa (especificada mediante datos de máquina) no puede modificarse mediante la función "Ejes geométricos conmutables".

Si es necesario modificar la configuración de ejes geométricos en el contexto de las transformadas, esto sólo es posible mediante otra transformada.

Una configuración de ejes geométricos modificada mediante `GEOAX` se borra al activar una transformada.

Si los ajustes de los datos de máquina para la transformada y para la conmutación de ejes geométricos se contradicen, tienen preferencia los ajustes en la transformada.

Ejemplo:

Una transformada está activa. Según los datos de máquina, la transformada debe mantenerse a pesar de un RESET, pero, al mismo tiempo, durante un RESET debe generarse la configuración básica de los ejes geométricos. En este caso, se mantiene la configuración de ejes geométricos que se especificó con la transformada.

14.3 Contenedor de ejes (AXCTSWE, AXCTSWED)

Función

En máquinas cíclicas giratorias o multihusillo, los ejes que portan la pieza se mueven desde una unidad de mecanizado hasta la siguiente. Si las unidades de mecanizado están subordinadas a diferentes canales NCU, en caso de cambio de estación o de posición, los ejes que llevan la pieza deben ser reasignados dinámicamente al correspondiente canal NCU. Para esta finalidad sirve el "Contenedor de ejes".

En un determinado instante sólo está activo un eje/cabezal con la pieza en la unidad de mecanizado local. El contenedor de ejes agrupa las posibilidades de conexión con todos los ejes/cabezales de amarre de piezas, de las cuales siempre sólo justamente una está activada para la unidad de mecanizado.

El cambio de los ejes utilizables definidos en un contenedor de ejes se efectúa por desplazamiento de las entradas en el contenedor de ejes ("Giro del contenedor de ejes") con un paso definible mediante dato de operador (cantidad de slots).

La llamada del giro del contenedor de ejes del programa de pieza se efectúa con el comando `AXCTSWE` o `AXCTSWED`.

Sintaxis

```
AXCTSWE(<Contenedor de ejes>)
AXCTSWED(<Contenedor de ejes>)
```

Descripción

`AXCTSWE`
Comando para girar un contenedor de ejes

Cuando el control haya recibido las señales de habilitación de todos los canales para los ejes del contenedor, se lleva a cabo el giro de contenedor con el paso específico de contenedor consignado en el DO41700 $SN_AXCT_SWWIDTH[<Número de contenedor>].

`AXCTSWED`
Comando para girar un contenedor de ejes bajo el único efecto del canal activo (¡variante de comando para la puesta en marcha!)

Nota:
Los ejes introducidos en el contenedor sólo se habilitarán cuando los otros canales que tienen ejes en el contenedor se encuentren en el estado Reset.

`<Contenedor de ejes>`
Identificador del contenedor de ejes que debe cambiarse.

Posibles indicaciones:

`CT<Número de contenedor>`	A la combinación de letras CT se le añade el número del contenedor de ejes. Ejemplo: `CT3`
`<Nombre del contenedor>`	Nombre del contenedor de ejes individual ajustado con DM12750 $MN_AXCT_NAME_TAB. Ejemplo: `A_CONT3`

Información adicional

Contenedor de ejes

Con los contenedores de ejes se pueden realizar las siguientes asignaciones.

- Ejes locales y/o
- Ejes lincados o Link

Los contenedores de ejes con ejes de enlace (link) son un recurso de orden supraordenado a la NCU (NCU global) que es coordinado por el control. Son posibles contenedores de ejes en los que se administren exclusivamente ejes locales.

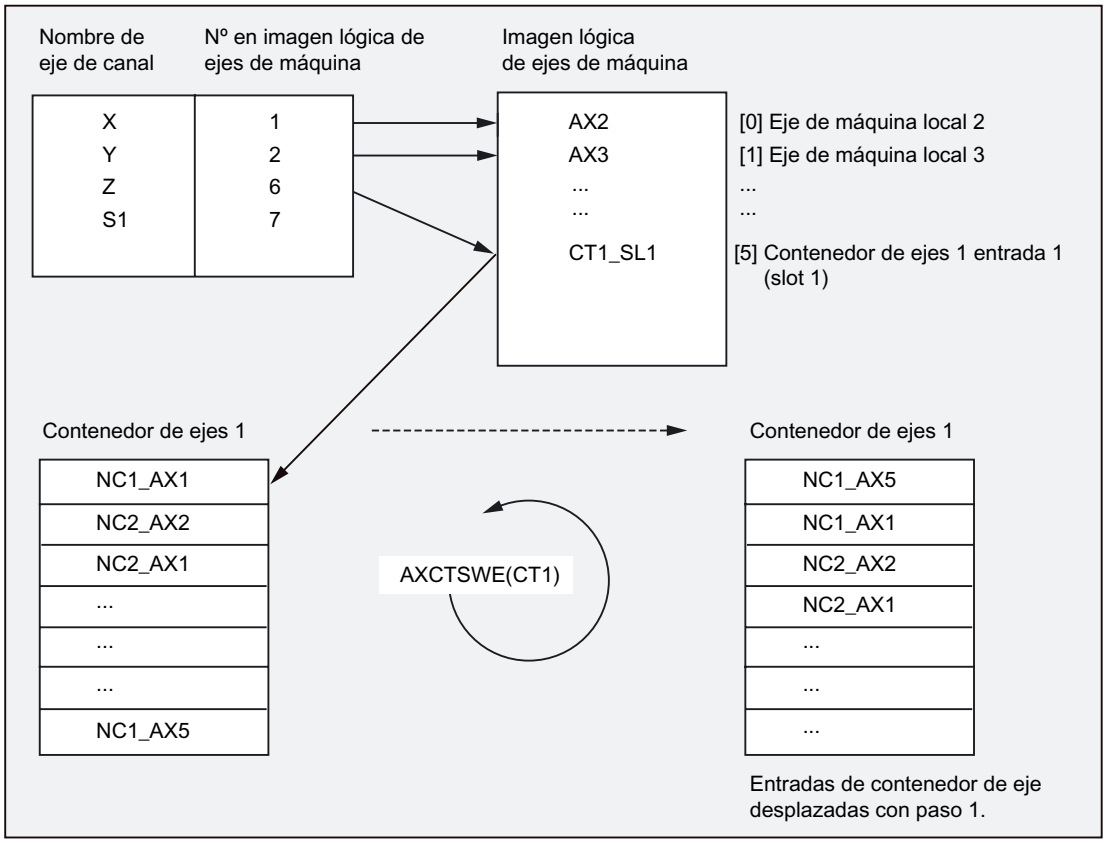
Bibliografía:
Encontrará información detallada sobre la configuración de contenedores de ejes en:
Manual de funciones de ampliación; Varios paneles de operador en varias NCU, Sistemas descentralizados (B3)

Criterios para la habilitación o el desbloqueo

AXCTSWE( )

Cualquier canal cuyos ejes queden registrados en el contenedor indicado emite la señal de habilitación para un giro de contenedor (enable) cuando ha terminado la ejecución de la posición/estación. Cuando el control haya recibido las señales de habilitación de **todos** los canales para los ejes del contenedor, se lleva a cabo el giro de contenedor con el paso específico de contenedor consignado en el DO41700 $SN_AXCT_SWWIDTH[<Número de contenedor>].

Ejemplo:



Tras la rotación en 1 del contenedor de ejes, el eje de canal Z queda asignado en la NCU 1 al eje AX5 en lugar del eje AX1en la NCU1.

AXCTSWED( )

La variante de comando `AXCTSWED()` se puede utilizar para simplificar la puesta en marcha. El contenedor de ejes realiza el giro con el solo efecto del canal activo en el paso específico de contenedor consignado en el DO41700 $SN_AXCT_SWWIDTH[<Número de contenedor>]. Esta llamada sólo se debe utilizar cuando los demás canales que tengan ejes en el contenedor se encuentren en el estado **Reset**.

Efecto

La nueva asignación de ejes tras un giro del contenedor de ejes afecta a todas las NCU cuyos canales estén relacionados, vía la imagen lógica de ejes de máquina, con el contenedor de ejes girado.

Giro del contenedor de ejes con GET/GETD implícito

Al habilitar un giro de contenedor de ejes, todos los ejes de contenedor de ejes asignados al canal se asignan al canal con `GET` o `GETD`. Una nueva transferencia de los ejes sólo se permite después del giro del contenedor de ejes.

Nota

Este comportamiento puede ajustarse en datos de máquina. Preste atención a las indicaciones del fabricante de la máquina.

Nota

El giro del contenedor de ejes con `GET`/`GETD` implícito **no** se puede utilizar para un eje que se encuentra en el estado Eje de marcha principal (p. ej. para un eje PLC), dado que dicho eje debería abandonar entonces este estado para el giro del contenedor de ejes.

14.4 Comprobar el repertorio del lenguaje CN disponible (STRINGIS)

Función

La disponibilidad real y las propiedades específicas de programación del repertorio del lenguaje CN generado con un SINUMERIK 840D sl, incluidas las definiciones activas de GUD/macros, así como los programas de ciclos instalados y activos, pueden comprobarse a través del comando `STRINGIS`. Así es posible, p. ej., verificar la efectividad de las funciones no activadas ya desde el principio de una interpretación de programa.

Los valores de retorno se emiten codificados por posiciones mediante la interfaz de usuario HMI y contienen información básica e información detallada codificada por posiciones adicional.

Sintaxis

```
STRINGIS(STRING name) = valor de retorno codificado por posiciones
```

En la configuración actual, el `(STRING name)` que debe comprobarse:

se identifica básicamente como desconocido con **000**.

se identifica básicamente con **100** como instrucción de programación CN, pero no es programable.

Todas las instrucciones de programación CN programables, que se encuentran activas como opción o función,

se identifican con **2xx**. Puede encontrarse información detallada al respecto en los márgenes de valores.

Descripción

El modo de proceder y con qué instrucciones de programación CN debe procederse, viene determinado por el fabricante de la máquina mediante un dato de máquina.

Si se programan instrucciones de programación cuya función no está activa o se desconoce en el entorno actual, se emite un aviso de alarma. ¡Observar en este caso las indicaciones del fabricante de la máquina!

`STRINGIS`	Comprueba el repertorio del lenguaje CN disponible y, de forma especial, los nombres de ciclo CN, variables de usuario, macros y nombres de label pertenecientes a este comando: si existen, son válidos, están definidos o están activos. El comando de lenguaje CN `STRINGIS` es una variable del tipo Integer.
`Especial para STRINGIS`	**Nombres de ciclo CN** (de un ciclo activo) **Valores de las variables GUD** **Variables LUD** **Macros** **Nombres de label**
`STRING name`	Identificador de variable del repertorio del lenguaje CN que debe comprobarse y parámetros de transferencia de los valores identificados del tipo STRING.

La instrucción de programación ISVAR representa un subconjunto del comando STRINGIS y puede seguir aplicándose para determinadas comprobaciones.

Repertorio del lenguaje CN

Para SINUMERIK powerline siguen siendo vigentes todas las instrucciones de programación disponibles y, también especialmente, las no necesarias pero activas. El repertorio del lenguaje que debe comprobarse en SINUMERIK solution line depende del dato de máquina preconfigurado y comprende, o bien todas las opciones conocidas o especialmente sólo las habilitadas, o bien las funciones activas en el repertorio del lenguaje CN actual.

Repertorio del lenguaje CN

Al repertorio del lenguaje CN pertenecen:

Códigos G todos los grupos de código G existentes, como p. ej., G0, G1, G2, INVCW, POLY, ROT, KONT, SOFT, CUT2D, CDON, RMB, SPATH

Direcciones DIN o CN como, p. ej., ADIS, RNDM, SPN, SR, MEAS

Funciones del lenguaje CN como, p. ej., subprogramas predefinidos
TANG(Feje1..n, Leje1..n, factor de acoplamiento).

Procedimientos del lenguaje CN (procedimientos predefinidos **con** valor de retorno), como p. ej., llamada de subprograma con transferencia de parámetros GETMDACT.

Procedimientos del lenguaje CN (procedimientos predefinidos **sin** valor de retorno), como p. ej., desconexión de la supresión de secuencia individual SBLOF.

Palabras reservadas CN como, p. ej., ACN, ACP, AP, RP, DEFINE, SETMS

Datos de máquina $MN generales, $MA de ejes, $MC específicos de canal, así como todos los datos de operador $S... y de opción $O....

Variable de sistema CN $ en el programa de pieza y en las acciones síncronas, así como **parámetros de cálculo CN** R.

Valores de retorno

Información básica STRINGIS	El valor de retorno está codificado por posiciones. Las informaciones básicas contenidas se dividen en y; la información detalla disponible, en x.
Codificación por posiciones:	**Resultado de la comprobación** de si en la configuración actual:
000	El NCK desconoce el nombre de STRING (STRING name).
100	El STRING name es una instrucción de programación pero **no es programable**, es decir, esta función es inactiva.
2xx	El STRING name es una instrucción de programación **programable**, es decir, esta función está activa.
y00	No es posible una asignación
y01 a y11	Se conocen los márgenes de valores de las informaciones detalladas.
400	Direcciones CN que no tienen xx=01 ni xx=10 y que no son un código G G o un parámetro de cálculo R, ver nota **(1)**.

Nota

Si en una comprobación con STRINGIS **no** se obtuviera otra codificación de posiciones, la instrucción de programación CN correspondiente se contaría como programable y se aplicaría la codificación por posiciones 2xx.

Márgenes de valores 2xx de las informaciones detalladas

Información detallada	Significado del resultado de la comprobación:
200	No es posible ninguna interpretación
201	Una dirección DIN o una dirección CN está definida, esto significa que, a partir del nombre se han identificado las letras de dirección, ver nota **(1)**
202	Los códigos G de los grupos de código G existentes se han identificado.
203	Se dispone de las funciones de lenguaje CN con valor de retorno y transferencia de parámetros.
204	Se dispone de los procedimientos de lenguaje CN sin valor de retorno y con transferencia de parámetros.
205	Se dispone de palabras reservadas CN.
206	Se dispone de datos de máquina generales, de ejes o específicos de canal (`$M...`), datos de operador (`$S...`) o datos de opciones(`$O...`).
207	Se dispone de variables de usuario, como p. ej., la variable de sistema CN, que comienzan por `$...` o de parámetros de cálculo que comienzan por `R`.
208	Los nombres de ciclos en NCK han sido cargados y se han activado programas de ciclos, ver nota **(2)**.
209	Se ha identificado el nombre definido de las variables de usuario globales (variables GUD) y la variable GUD activa.
210	Se han encontrado los nombres de macro con los nombres definidos en los ficheros de definición de macros, así como macros activadas, ver nota **(3)**.
211	De variables de usuario locales (variables LUD) cuyo nombre está contenido en un programa actual.

Nota

Notas en relación con algunos valores de retorno

(1) Como direcciones DIN se identifican direcciones normalizadas fijas. Para las direcciones CN con identificadores ajustables rigen las siguientes especificaciones para ejes geométricos:

`A, B, C` para ejes giratorios especificados, `E` está reservada para ampliaciones y las letras `I, J, K, Q, U, V, W, X, Y, Z` para ejes lineales especificados.

Los identificadores de eje pueden programarse con una ampliación de dirección y escribirse con, por ejemplo, `201 = STRINGIS("A1")` para la comprobación.

Las siguientes direcciones no pueden escribirse con una ampliación de dirección para la comprobación y suministran siempre el valor fijo 400.

Ejemplo `400 = STRINGIS("D")` o la indicación de la ampliación de dirección con `0 = STRINGIS("M02")` genera `400 = STRINGIS("M")`.

(2) Los nombres de los parámetros de ciclo no pueden comprobarse con STRINGIS.

(3) Las letras de dirección CN definidas como macro `G, H, L y M` se identifican como macro.

Direcciones CN válidas sin ampliación de dirección con el valor fijo 400

Son válidas la direcciones CN `D, F, G, H, R y L, M, N, O, P, S, T`. Siendo

```
400                 D corrección de herramienta, número de corte (función D)
                    F avance (función F)
                    G código G (en este caso no se trata de la función G)
                    H función auxiliar (función H)
                    R parámetro de sistema, así como:
                    L llamada de subprograma, M función adicional, N secuencia
                    auxiliar,
                    O libre para ampliaciones,
                    P número de ejecuciones del programa,
                    S velocidad de giro del cabezal (función S),
                    T número de herramienta (función T).
```

Ejemplo de función auxiliar T programable

```
Programación                                    Comentarios
T está definida como función auxiliar y siempre puede programarse
400 = STRINGIS("T")                             ;
0 = STRINGIS("T3")                                 ;Valor de retorno sin ampliación
                                                   de dirección
                                                   ;Valor de retorno con ampliación
                                                   de dirección
```

Ejemplos de otras comprobaciones del repertorio del lenguaje CN 2xx programable

Programación	**Comentarios**
X está definido como eje	; Eje es un eje lineal X
201 = STRINGIS("X")	; Valor de retorno del eje lineal X
201 = STRINGIS("X1")	Valor de retorno del eje lineal X1
A2 es una dirección CN con ampliación	Dirección CN A2 con ampliación
201 = STRINGIS("A")	Valor de retorno con dirección CN A
201 = STRINGIS("A2")	con dirección CN A2 ampliada
INVCW es un código G definido	INVCW es una interpolación de evoluta de código G en sentido horario.
202 = STRINGIS("INVCW")	Valor de retorno del código G denominado
GETMDACT es una función de lenguaje CN	La función de lenguaje CN GETMDACT está disponible.
203 = STRINGIS("GETMDACT")	GETMDACT es una función de lenguaje CN
DEFINE es una palabra reservada CN	Existe la palabra reservada DEFINE como identificador de macros.
205 = STRINGIS("DEFINE")	DEFINE está disponible como palabra reservada
$MC_GCODES_RESET_VALUES es un dato de máquina específico de canal	Existe el dato de máquina $MC_GCODE_RESET_VALUES.
206 = STRINGIS("$MC_GCODE_RESET_VALUES")	$MC_GCODE_RESET_VALUES ha sido identificado como dato de máquina
$TC_DP3 es una variable del sistema para una componente longitudinal de herramienta	Existe la variable de sistema CN $TC_DP3 para componente longitudinal de herramienta.
207 = STRINGIS("$TC_DP3")	$TC_DP3 identificado como variable del sistema.
$TC_TP4 es una variable del sistema para un tamaño de herramienta	Existe la variable de sistema CN $TC_TP4 para el tamaño de herramienta.
207 = STRINGIS("$TC_TP4")	$TC_TP4 identificado como variable del sistema.
$TC_MPP4 es una variable del sistema para el estado del puesto en el almacén	Comprobación de la gestión del almacén en
207 = STRINGIS("$TC_MPP4")	Gestión del almacén activa
0 = STRINGIS("$TC_MPP4")	La gestión del almacén no está disponible (4)
MACHINERY_NAME está definida como variable GUD	Variable global de usuario está definida como MACHINERY_NAME.
209 = STRINGIS("MACHINERY_NAME")	MACHINERY_NAME encontrada como GUD
LONGMACRO está definida como macro	El nombre de la macro es LONGMACRO

Programación	**Comentarios**
210 = STRINGIS("LONGMACRO")	Macro identificada como LONGMACRO
MYVAR está definida como variable LUD	La variable local de usuario ha sido denominada como MYVAR
211 = STRINGIS("MYVAR")	La variable LUD está contenida como nombre MYVAR en el programa actual
X, Y, Z es un comando desconocido en CN	X,Y,Z es un comando de lenguaje desconocido y tampoco es un nombre de GUD/macro/ciclo
0 = STRINGIS("XYZ")	STRING name X, Y, Z es desconocido

(4) Para los parámetros de sistema de la gestión del almacén es válida especialmente la característica en caso de que la función no esté activa. Entonces, STRINGIS proporciona siempre el valor de resultado = 0 para la configuración del repertorio del lenguaje CN, independientemente del valor definido del dato de máquina.

14.5 Leer llamada de función ISVAR y datos de máquina con índice Array

Función

El comando ISVAR es una función en el sentido del lenguaje CN con un:

- Valor de función del tipo BOOL
- Parámetro de transferencia del tipo STRING

El comando ISVAR suministra TRUE si el parámetro de transferencia contiene una variable conocida en el CN (dato máquina, dato de operador, variable de sistema, variables generales como GUD).

Sintaxis

```
ISVAR(<Identificador de variable>)
ISVAR(<Identificador>,[<Valor>,<Valor>])
```

Descripción

`<Identificador de variable>`	El parámetro de transferencia del tipo string puede carecer de dimensión, ser unidimensional o bidimensional.
`<Identificador>`	Identificador de una variable conocida de CN con o sin índice Array como dato máquina, dato de operador, variable de sistema o variable general. **Ampliación:** En datos de máquina generales y específicos de cada canal, el primer elemento del Array se lee incluso si falta el índice.
`<Valor>`	Valor de función del tipo BOOL

Comprobaciones

Conforme al parámetro de transferencia se realizan las siguientes comprobaciones:

- Si existe el identificador
- Si se trata de una matriz uni o bidimensional
- Si se admite un índice Array

Sólo si todas estas comprobaciones son positivas, se devuelve TRUE. Si sólo una comprobación no se cumple o si se produce un error de sintaxis, se emite FALSE. Las variables de eje se aceptan como índice para los nombres de eje, pero no se comprueban más detalladamente.

Ampliación: Leer datos de máquina y datos de operador Array sin índice.

Si falta el índice de datos de máquina **generales o específicos del canal**, **ya no** se emite la alarma 12400 "Canal % 1 Secuencia % 2 Matriz % 3 Elemento no existe".

Asimismo, se tiene que programar **al menos el índice del eje** en datos de máquina **específicos del eje**. De lo contrario, se emite la alarma 12400.

Ejemplo: Llamada de función ISVAR

```
Código del programa                                    Comentarios
DEF INT VAR1
DEF BOOL IS_VAR=FALSE                                  ; El parámetro de transferencia es una
                                                         variable general
N10 IS_VAR=ISVAR("VAR1")                               ; IS_VAR es TRUE en este caso
DEF REAL VARARRAY[10,10]
DEF BOOL IS_VAR=FALSE                                  ; Distintas variantes de sintaxis
N20 IS_VAR=ISVAR("VARARRAY[,]")                        ; IS_VAR es TRUE con un Array bidimensional
N30 IS_VAR=ISVAR("VARARRAY")                           ; IS_VAR es TRUE, la variable existe
N40 IS_VAR=ISVAR("VARARRAY[8,11]")                     ; IS_VAR es FALSE, el índice Array no se
                                                         admite
N50 IS_VAR=ISVAR("VARARRAY[8,8")                       ; IS_VAR es FALSE, error de sintaxis debido
                                                         a la falta de "]"
N60 IS_VAR=ISVAR("VARARRAY[,8]")                       ; IS_VAR es TRUE, el índice Array se admite
N70 IS_VAR=ISVAR("VARARRAY[8,]")                       ; IS_VAR es TRUE
DEF BOOL IS_VAR=FALSE                                  ; El parámetro de transferencia es un dato
                                                         de máquina
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"          ; IS_VAR es TRUE
DEF BOOL IS_VAR=FALSE                                  ; El parámetro de transferencia es una
                                                         variable del sistema
N10 IS_VAR=ISVAR("$P_EP")                              ; IS_VAR es TRUE en este caso
N10 IS_VAR=ISVAR("$P_EP[X]")                           ; IS_VAR es TRUE en este caso
```

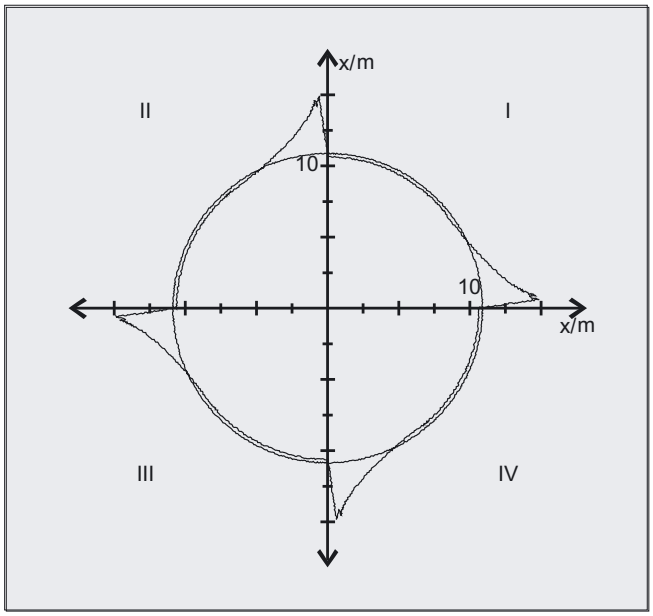
Ejemplo: Leer datos de máquina Array con y sin índice

El primer elemento se lee en
`R1=$MC_EXTERN_GCODE_RESET_VALUES`
esto equivale, como hasta ahora, a
`R1=$MC_EXTERN_GCODE_RESET_VALUES[0]`
o se lee el primer elemento
`R1=$MA_POSTCTRL_GAIN[X1]`
esto equivale como hasta ahora
`R1=$MA_POSTCTRL_GAIN[0, X1]`
También se lee el primer elemento en acciones síncronas con
`WHEN TRUE DO $R1 = $MC_EXTERN_GCODE_RESET_VALUES`
esto equivale como hasta ahora
`WHEN TRUE DO $R1 = $MC_EXTERN_GCODE_RESET_VALUES[0]`
y, hasta este momento, **no se leyó** con la alarma 12400.
Se sigue emitiendo la alarma 12400 con
`R1=$MA_POSTCTRL_GAIN`

14.6 Aprendizaje de características de compensación (QECLRNON, QECLRNOF)

Función

La compensación del error de cuadrante (QFK) reduce errores en el contorno generados por efectos mecánicos no lineales (p. ej., rozamiento, juego) o torsión. Basados en una red neuronal, los valores óptimos de compensación los puede adaptar el control numérico durante una fase de aprendizaje que permita determinar la característica de compensación de forma automática. El aprendizaje se puede realizar de forma simultánea para hasta 4 ejes.



Sintaxis

```
QECLRNON
QECLRNOF
```

Activar el proceso de aprendizaje: QECLRNON

El proceso de aprendizaje en sí se activa en el programa CN con el comando `QECLRNON`, indicando los ejes:

```
QECLRNON (X1, Y1, Z1, Q)
```

Sólo cuando este comando esté activo se modificarán las características.

Desactivar aprendizaje: QECLRNOF

Una vez concluidos los movimientos de aprendizaje de los ejes deseados, el proceso de aprendizaje se desactivará con `QECLRNOF` para todos los ejes al mismo tiempo.

Descripción

`QECLRNON (Eje.1,…4)`	Activar función "Aprendizaje compensación de errores de cuadrante"
`QECLRNO`	Desactivar función "Aprendizaje compensación de errores de cuadrante"
`QECLRN.SPF`	Ciclo de aprendizaje
`QECDAT.MPF`	Programa CN de muestra para la ocupación de las variables del sistema y para la parametrización del ciclo de aprendizaje
`QECTEST.MPF`	Programa CN de muestra para test de circularidad

Descripción

Los desplazamientos de los ejes requeridos por el proceso de aprendizaje se generan mediante un programa de pieza. Dichos movimientos se memorizan en el programa en forma de un ciclo de aprendizaje.

Fase inicial de aprendizaje

Para el primer aprendizaje en la puesta en marcha, el disquete del programa básico PLC contiene programas CN de muestra para el aprendizaje para los movimientos de aprendizaje, así como para la asignación de las variables de sistema QFK.

Optimización

Esta función permite optimizar posteriormente la curva característica aprendida. Los datos almacenados previamente se toman como base para la optimización. Para el proceso de optimización se debe realizar una adaptación del programa maestro a los requisitos del proceso.

Para la optimización se deben modificar algunos de los parámetros incluidos en el ciclo de aprendizaje (QECLRN.SPF):

- ”Modo aprendizaje” = poner a 1
- Reducción del “número de pasadas de aprendizaje”
- Se deben definir los límites de trabajo y se debe activar el “aprendizaje modular”

14.7 Llamar interactivamente la ventana del programa de pieza (MMC)

Función

A través del comando MMC se pueden mostrar desde el programa de pieza en el HMI ventanas de diálogo (pantallas de diálogo) definidas por el usuario.

El aspecto de las ventanas de diálogo es definido a través de una configuración puramente textual (fichero COM en el directorio de ciclos); el software de sistema HMI permanece invariable.

Las ventanas de diálogo definidas por el usuario no se pueden llamar simultáneamente en varios canales.

Sintaxis

```
MMC(CYCLES,PICTURE_ON,T_SK.COM,BILD,MGUD.DEF,BILD_3.AWB,TEST_1,A1","
S")
```

Descripción

MMC	Llamar la ventana de diálogo de forma interactiva desde el programa de pieza en el HMI.
CYCLES	Campo de manejo en el cual se ejecutan los diálogos de usuario configurados.
PICTURE_ON o PICTURE_OFF	Comando: selección o deselección de pantallas
T_SK.COM	Fichero Com: Nombre del fichero de pantallas de diálogo (ciclos del usuario). Aquí está definido el aspecto de las pantallas de diálogo. En la pantalla de diálogo se pueden visualizar variables de usuario y/o textos de comentario.
FIG.	Nombre de la pantalla de diálogo: Las distintas pantallas se seleccionan mediante el nombre de pantalla de diálogo.
MGUD.DEF	Fichero de definición de datos del usuario al cual se accede en la lectura / escritura de variables.
BILD_3.AWB	Fichero de gráficos
TEST_1	Tiempo de visualización o variable de acuse
A1	Variables de texto...",
"S"	Modo de acuse: sincrónico, acuse de recibo a través del pulsador de menú "OK"

Bibliografía

Encontrará información detallada sobre la programación del comando MMC (incl. ejemplos de programación) en el manual de puesta en servicio.

14.8 Tiempo de ejecución de programa/contador de piezas

14.8.1 Tiempo de ejecución de programa/contador de piezas (lista)

Para ayudar al operador de máquinas herramienta, se proporciona información sobre el tiempo de ejecución del programa y la cantidad de piezas.

Esta información puede procesarse como variables del sistema en el programa CN o PLC. Al mismo tiempo, esta información está disponible para la visualización en la interfaz de usuario.

14.8.2 Tiempo de ejecución del programa

Función

La función "Tiempo de ejecución del programa" proporciona temporizadores internos de CN para la vigilancia de procesos tecnológicos que pueden leerse mediante variables del sistema específicas de canal y de CN en el programa de pieza y en acciones síncronas.

El disparador de la medición de tiempo de ejecución ($AC_PROG_NET_TIME_TRIGGER) es la única variable del sistema de la función que puede escribirse y sirve para la medición selectiva de partes del programa. Es decir que, mediante la descripción del disparador en el programa CN, la medición de tiempo puede activarse y desactivarse.

Variable de sistema	Descripción	Actividad
Específico de CN		
$AN_SETUP_TIME	Tiempo en minutos desde el último arranque del control con valores estándar ("arranque en frío") Se resetea automáticamente a "0" con cada arranque del control con valores estándar.	Siempre activo
$AN_POWER ON_TIME	Tiempo en minutos desde el último arranque normal del control ("arranque en caliente") Se resetea automáticamente a "0" con cada arranque normal del control.	

Variable de sistema	Descripción	Actividad
Específico de canal		
$AC_OPERATING_TIME	Tiempo de ejecución total en segundos de programas CN en el modo automático El valor se resetea automáticamente a "0" con cada arranque del control.	• Activación con DM27860 • Sólo modo de operación AUTOMÁTICO
$AC_CYCLE_TIME	Tiempo de ejecución en segundos del programa CN seleccionado El valor se resetea automáticamente a "0" con el inicio de un nuevo programa CN. A través de DM27860 puede ajustarse que este borrado se produzca también al saltar al principio del programa con `GOTOS` o al iniciar ASUP y PROG_EVENT.	
$AC_CUTTING_TIME	Tiempo de mecanizado en segundos Se mide el tiempo de ejecución de los ejes de contorneado (al menos uno debe estar activo) sin activar el desplazamiento en rápido en todos los programas CN entre Marcha CN y Fin de programa/Reset CN. La medición se interrumpe adicionalmente cuando está activo el tiempo de espera. El valor se resetea automáticamente a "0" con cada arranque del control con valores estándar.	
$AC_ACT_PROG_NET_TIME	Tiempo de ejecución neto actual en segundos del programa CN actual Se resetea automáticamente a "0" con el inicio de un programa CN.	• Siempre activo • Sólo modo de operación AUTOMÁTICO
$AC_OLD_PROG_NET_TIME	Tiempo neto de ejecución del programa finalizado correctamente con `M30` en segundos	
$AC_OLD_PROG_NET_TIME_COUNT	Modificaciones en $AC_OLD_PROG_NET_TIME Tras POWER ON se encuentra $AC_OLD_PROG_NET_TIME_COUNT en "0". $AC_OLD_PROG_NET_TIME_COUNT se incrementa siempre que el control $AC_OLD_PROG_NET_TIME se escribe de nuevo.	

Variable de sistema	Descripción		Actividad
$AC_PROG_NET_TIME_TRIGGER	Disparador para la medición de tiempo de ejecución:		Sólo modo de operación AUTOMÁTICO
	0	Estado neutro El disparador no está activo.	
	1	Salir Finaliza la medición y copia el valor de $AC_ACT_PROG_NET_TIME en $AC_OLD_PROG_NET_TIME. $AC_ACT_PROG_NET_TIME se fija en "0" y después continúa.	
	2	Inicio Inicia la medición y setea $AC_ACT_PROG_NET_TIME en "0". $AC_OLD_PROG_NET_TIME no se modifica.	
	3	Parada Para la medición. No modifica $AC_OLD_PROG_NET_TIME y mantiene $AC_ACT_PROG_NET_TIME constante hasta el momento de proseguir.	
	4	Continuar Continuar la medición, es decir que se reanuda una medición parada anteriormente. $AC_ACT_PROG_NET_TIME continúa. $AC_OLD_PROG_NET_TIME no se modifica.	
¡Mediante POWER ON, todas las variables del sistema vuelven a "0"!			

Nota

La conexión de los temporizadores activables y el comportamiento de las mediciones de tiempo activas en ciertas situaciones (p. ej. cuando está activo el avance de recorrido de prueba o en una prueba del programa) se efectúa mediante el dato de máquina DM27860 $MC_PROCESSTIMER_MODE (→ fabricante de la máquina).

Bibliografía:
Manual de funciones, Funciones básicas; BAG/GMO, canal, modo de programa, comportamiento Reset (K1), capítulo: Tiempo de ejecución del programa

Nota

Tiempo residual de una pieza

Si se producen piezas iguales sucesivamente, a partir de los valores de temporización:

- tiempo de mecanizado de la última pieza producida (ver $AC_OLD_PROG_NET_TIME)

y

- tiempo de mecanizado actual (ver $AC_ACT_PROG_NET_TIME)

puede calcularse el tiempo residual de una pieza.

El tiempo residual se muestra en la interfaz de usuario junto con el tiempo de mecanizado actual.

ATENCIÓN
Uso de STOPRE Las variables del sistema $AC_OLD_PROG_NET_TIME y $AC_OLD_PROG_NET_TIME_CTR no generan ninguna parada de decodificación previa implícita. Esto no es ningún problema si se usa en el programa de pieza cuando el valor de las variables del sistema proviene de la ejecución de programa anterior. Sin embargo, cuando el disparador de la medición del tiempo de ejecución ($AC_PROG_NET_TIME_TRIGGER) se escribe con alta frecuencia y a causa de ello $AC_OLD_PROG_NET_TIME cambia muy a menudo, debe utilizarse un STOPRE explícito en el programa de pieza.

Limitaciones

- **Búsqueda de secuencia**

  En la búsqueda de secuencia no se calculan tiempos de ejecución del programa.

- **REPOS**

  La duración de un proceso REPOS se suma al tiempo de mecanizado actual ($AC_ACT_PROG_NET_TIME).

Ejemplos

Ejemplo 1: Medir la duración de "mySubProgrammA"

Código del programa

```
...
N50 DO $AC_PROG_NET_TIME_TRIGGER=2
N60 FOR ii= 0 TO 300
N70 mySubProgrammA
N80 DO $AC_PROG_NET_TIME_TRIGGER=1
N95 ENDFOR
N97 mySubProgrammB
N98 M30
```

Cuando el programa ha procesado la línea N80, en $AC_OLD_PROG_NET_TIME aparece el tiempo de ejecución neto de "mySubProgrammA".

El valor de $AC_OLD_PROG_NET_TIME:

- se mantiene más allá de M30.
- se actualiza tras cada rectificado.

Ejemplo 2: Medir la duración de "mySubProgrammA" y "mySubProgrammC"

Código del programa

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
```

14.8.3 Contador de piezas

Función

La función "Contador de piezas" ofrece diversos contadores que pueden utilizarse, sobre todo, para contar piezas dentro del control.

Los contadores existen como variables de sistema específicas del canal con acceso de escritura y lectura, en la gama de valores de 0 a 999 999 999.

Variable del sistema	Descripción
$AC_REQUIRED_PARTS	Número de piezas que deben fabricarse (cantidad teórica de piezas) En este contador puede definirse el número de piezas que, al ser alcanzado, pone a "0" el número real de piezas ($AC_ACTUAL_PARTS).
$AC_TOTAL_PARTS	Número total de piezas fabricadas (cantidad real total de piezas) Este contador indica el número total de piezas fabricadas desde el momento de arranque. El valor sólo se resetea automáticamente a "0" con un arranque del control con valores estándar.
$AC_ACTUAL_PARTS	Número de piezas fabricadas (cantidad real de piezas) En este contador se registra el número total de piezas fabricadas desde el momento del arranque. Al alcanzar el número teórico de piezas ($AC_REQUIRED_PARTS), el contador se pone a "0" automáticamente (siempre que $AC_REQUIRED_PARTS > 0).
$AC_SPECIAL_PARTS	Número de piezas contadas por el usuario Este contador permite al usuario contar las piezas según su propia definición. Puede definirse que se emita una alarma al alcanzar el número teórico de piezas ($AC_REQUIRED_PARTS). El mismo usuario tiene que realizar la puesta a cero del contador.

Nota

Todos los contadores de piezas se ponen a "0" en un arranque del control con valores estándar y pueden leerse y escribirse independientemente de su activación.

Nota

A través de los datos de máquina específicos del canal se puede influir en la activación de contadores, el momento de la puesta a cero y el algoritmo de recuento.

Nota

Contador de piezas con comando M definido por el usuario

A través de los datos de máquina puede definirse que los impulsos de contaje para los diferentes contadores de pieza se disparen mediante comandos M definidos por el usuario en vez de hacerlo mediante el fin del programa `M2`/`M30`.

Bibliografía

Para más información sobre la función "Contador de piezas":

- Manual de funciones, Funciones básicas; BAG/GMO, canal, modo de programa, comportamiento Reset (K1), capítulo: Contador de piezas

14.9 Alarmas (SETAL)

Función

Es posible fijar alarmas en un programa CN. Las alarmas se visualizan en la interfaz de usuario dentro de un campo especial. Una alarma provoca una reacción en el control numérico que depende de la categoría a la que ha sido asociada.

Bibliografía:
Para más información sobre las reacciones de alarma, ver el Manual de puesta en marcha.

Sintaxis

```
SETAL(<Número de alarma>)
SETAL(<Número de alarma>,<Cadena de caracteres>)
```

Descripción

`SETAL`	Palabra reservada para programar una alarma. `SETAL` se debe programar en una secuencia CN propia.
`<Número de alarma>`	Una variable del tipo INT. Contiene el número de alarma. El rango de alarmas válido es de 60000 a 69999; el rango desde 60000 a 64999 está reservado para las alarmas de los ciclos de SIEMENS, mientras que el rango de 65000 a 69999 queda a disposición del usuario.
`<Cadena de caracteres>`	Durante la programación de alarmas de ciclos de usuario se puede introducir además una cadena de caracteres con un máximo de 4 parámetros. En estos parámetros se pueden definir textos de usuario variables. Sin embargo, también están disponibles los siguientes parámetros predefinidos:

Parámetros	Descripción
%1	Número de canal
%2	Número de secuencia, lábel
%3	Índice de texto para alarmas de ciclo
%4	Parámetro de alarma adicional

Nota

Los textos de alarma deben configurarse en la interfaz de usuario.

Ejemplo

Código del programa	Comentarios
...	
N100 SETAL (65000)	; Activar la alarma nº 65000
...	

Programas de desbaste propios 15

15.1 Funciones de apoyo para el desbaste

Funciones

Para el desbaste se ofrecen ciclos de ejecución terminados. Además tiene la posibilidad de crear programas de desbaste propios con las funciones indicadas a continuación:

- Creación de tabla para el contorno (CONTPRON)
- Creación de tabla para el contorno codificada (CONTDCON)
- Desactivación de la preparación del contorno (EXECUTE)
- Determinación del punto de intersección entre dos elementos de contorno (INTERSEC)

  (Sólo para tablas elaboradas mediante CONTPRON).
- Ejecución de los elementos de contorno de una tabla secuencia a secuencia (EXECTAB)

  (Sólo para tablas elaboradas mediante CONTPRON).
- Cálculo de los datos de circunferencia (CALCDAT)

Nota

Puede utilizar estas funciones de forma universal y no sólo para el desbaste.

Requisitos

Antes de la llamada a las funciones CONTPRON o CONTDCON debe:

- Realizarse un desplazamiento a un punto inicial que permita un mecanizado sin colisiones.
- Estar desactivada la compensación del radio de filo mediante `G40`.

15.2 Creación de tabla para el contorno (CONTPRON)

Función

El comando CONTPRON activa la preparación del contorno. Las secuencias CN que se van a llamar a continuación no se ejecutan, sino que se dividen en desplazamientos individuales y se almacenan en la tabla para el contorno. Cada uno de los elementos del contorno corresponde a una fila en un campo bidimensional de la tabla de contornos. Se devuelve el número de puntos de destalonado (cambio de sentido en eje x) calculados.

Sintaxis

Activar la preparación del contorno:

```
CONTPRON(<Tabla para el contorno>,<Tipo de
mecanizado>,<Destalonados>,
<Sentido de mecanizado>)
```

Desactivación de la preparación del contorno y conmutación al modo de ejecución normal de programa:

```
EXECUTE(<ERROR>)
```

Ver "Desactivación de la preparación del contorno (EXECUTE)"

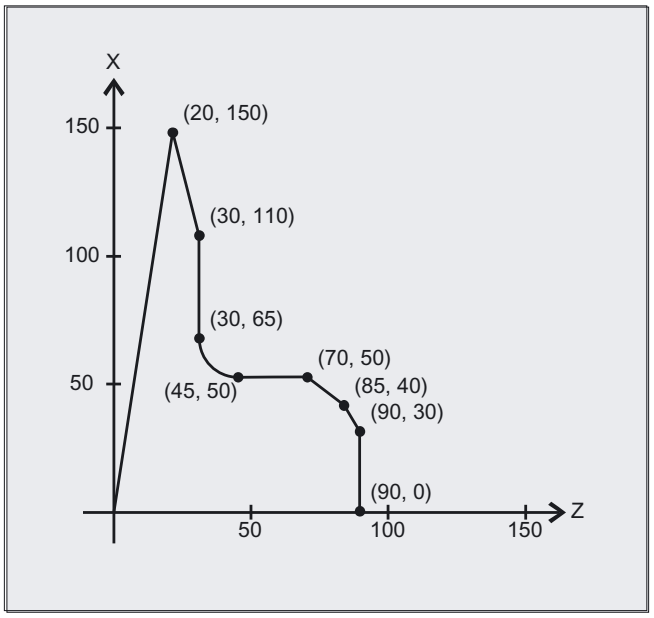
Descripción

CONTPRON	Comando para activar la preparación del contorno para crear una tabla para el contorno		
<Tabla para el contorno>	Nombre de la tabla para el contorno		
<Tipo de mecanizado>	Parámetros del tipo de mecanizado		
	Tipo:	CHAR	
	Valor:	"G"	Torneado longitudinal: Mecanizado interior
		"L"	Torneado longitudinal: Mecanizado exterior
		"N"	Refrentado: Mecanizado interior
		"P"	Refrentado: Mecanizado exterior
<Destalonados>	Variable de resultado para el número de elementos de destalonado presentes		
	Tipo:	INT	
<Sentido de mecanizado>	Parámetros del sentido de mecanizado		
	Tipo:	INT	
	Valor:	0	Preparación del contorno hacia delante (valor estándar)
		1	Preparación del contorno en ambos sentidos

Ejemplo 1

Creación de una tabla para el contorno con:

- nombres "KTAB",
- 30 elementos de contorno como máximo (círculos, rectas),
- una variable para el número de elementos de destalonado presentes,
- una variable para avisos de error.



Programa CN:

```
Código del programa                          Comentarios
N10 DEF REAL KTAB[30,11]                     ; Tabla para el contorno con nombres KTAB
                                               y 30 elementos de contorno como máximo,
                                               valor de parámetro 11 (número de colum-
                                               nas de la tabla) es un valor fijo.
N20 DEF INT CORTES                           ; Variable para el número de elementos de
                                               destalonado con nombre CORTES.
N30 DEF INT ERROR                            ; Variable para señalización de error
                                               (0=ningún error, 1=error).
N40 G18
N50 CONTPRON(KTAB,"G",CORTES)                ; Activar la preparación del contorno.
N60 G1 X150 Z20                              ; N60 a N120: descripción del contorno
N70 X110 Z30
N80 X50 RND=15
N90 Z70
N100 X40 Z85
N110 X30 Z90
N120 X0
N130 EXECUTE(ERROR)                          ; Finalización de la introducción de datos
                                               en la tabla para el contorno, conmuta-
                                               ción a la ejecución normal del programa
N140 …                                       ; Continuación del procesamiento de la
                                               tabla.
```

Tabla para el contorno KTAB:

Index Línea	Columna									
(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

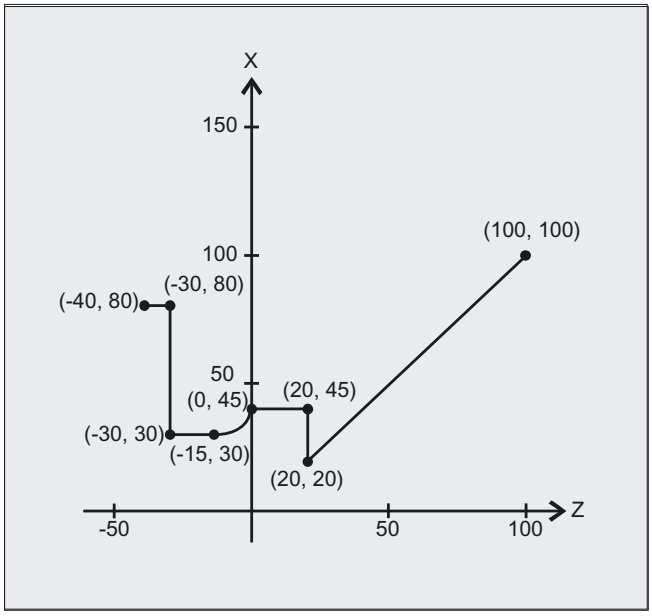
Explicación del contenido de las columnas:

(0)	Puntero en el siguiente elemento de contorno (en el número de línea del mismo)
(1)	Puntero en el elemento de contorno anterior
(2)	Codificación del modo de contorno para el movimiento Posibles valores para X = abc a = 10^2 G90 = 0 G91 = 1 b = 10^1 G70 = 0 G71 = 1 c = 10^0 G0 = 0 G1 = 1 G2 = 2 G3 = 3
(3), (4)	Punto inicial de los elementos de contorno (3) = Abscisa, (4) = Ordenada en el plano actual
(5), (6)	Punto final de los elementos de contorno (5) = Abscisa, (6) = Ordenada en el plano actual
(7)	Indicador máx/mín: caracteriza los máximos y mínimos locales en el contorno
(8)	Valor máximo entre el elemento de contorno y la abscisa (en el mecanizado longitudinal), o la ordenada (en el refrentado). El ángulo depende del modo de mecanizado programado.
(9), (10)	Coordenadas del punto central de un elemento del contorno en el caso de que sea un arco de circunferencia. (9) = Abscisa, (10) = Ordenada

Ejemplo 2

Creación de una tabla para el contorno con

- Nombres KTAB,
- 92 elementos de contorno como máximo (círculos, rectas),
- Modo de operación: torneado longitudinal, mecanizado exterior,
- Preparación hacia delante y atrás.



Programa CN:

Código del programa	**Comentarios**
N10 DEF REAL KTAB[92,11]	; Tabla para el contorno con nombres KTAB y como máx. 92 elementos de contorno; el valor de parámetro 11 es un valor fijo.
N20 DEF CHAR BT="L"	; Modo de operación para CONTPRON: torneado longitudinal, mecanizado exterior,
N30 DEF INT HE=0	; Cantidad de elementos de destalonado=0
N40 DEF INT MODE=1	; Preparación hacia delante y atrás.
N50 DEF INT ERR=0	; Señalización de error
...	
N100 G18 X100 Z100 F1000	

```
Código del programa                                  Comentarios
N105 CONTPRON(KTAB,BT,HE,MODE)                       ; Activar la preparación del
                                                       contorno.
N110 G1 G90 Z20 X20
N120 X45
N130 Z0
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)
N150 G1 Z-30
N160 X80
N170 Z-40
N180 EXECUTE(ERR)                                    ; Finalización de la introducción de
                                                       datos en la tabla para el
                                                       contorno, conmutación a la
                                                       ejecución normal del programa
...
```

Tabla para el contorno KTAB:

Tras finalizar la preparación del contorno, éste se encuentra disponible en ambos sentidos.

Index	Columna										
Línea	(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

Explicación del contenido de las columnas y de las notas de las filas 0, 1, 6, 8, 83, 85 y 91

Son válidas las explicaciones del contenido de las columnas citadas en el ejemplo 1.

Siempre en la línea de tabla 0:

1) Elemento anterior: A la línea n se le asigna el extremo de contorno hacia delante

2) Elemento posterior: La línea n es el final de la tabla para el contorno

En cada caso una vez dentro de los elementos de contorno hacia delante:

3) Elemento anterior: Comienzo de contorno (hacia delante)

4) Elemento posterior: Fin de contorno (hacia delante)

Siempre en la línea del final de la tabla para el contorno (hacia delante) +1:

5) Elemento anterior: Cantidad de puntos de destalonado hacia delante

6) Elemento posterior: Cantidad de puntos de destalonado hacia atrás

En cada caso una vez dentro de los elementos de contorno hacia atrás:

7) Elemento posterior: Extremo de contorno (hacia atrás)

8) Elemento anterior: Comienzo de contorno (hacia atrás)

Siempre en la última línea de tabla:

9) Elemento anterior: La línea n es el comienzo de la tabla para el contorno (hacia atrás)

10) Elemento posterior: La línea n contiene el comienzo del contorno (hacia atrás)

Información adicional

Órdenes de desplazamiento permitidas, sistema de coordenadas

Para la programación de contornos se admiten los siguientes comandos G:

- Grupo G 1: G0, G1, G2, G3

También son posibles:

- Redondeo y chaflán,
- Programación del círculo mediante CIP y CT

Las funciones Spline, Polinomio y Rosca producen errores.

No se pueden realizar cambios en el sistema de coordenadas mediante la activación de un frame entre las instrucciones CONTPRON y EXECUTE. Lo mismo rige para un cambio entre G70 y G71 o entre G700 y G710.

Cualquier cambio de los ejes geométricos con GEOAX durante la preparación de la tabla para el contorno origina una alarma.

Elementos de destalonado

La descripción del contorno para el cálculo de los puntos de destalonado (cambio de sentido en eje X) se puede realizar en un subprograma o bien en secuencias individuales.

Desbaste independiente de la dirección de contorno programada

La preparación del contorno con CONTPRON se ha ampliado de forma que, tras llamarla, la tabla para el contorno estará disponible independientemente del sentido programado.

15.3 Creación de tabla para el contorno codificada (CONTDCON)

Función

Cuando la preparación del contorno se activa con CONTDCON, las secuencias CN siguientes se almacenan de forma ventajosa para la memoria en una tabla para el contorno de 6 columnas. Cada uno de los elementos del contorno corresponde a una fila en la tabla de contornos. Conociendo las reglas de codificación abajo indicadas, se pueden confeccionar programas en código DIN, p. ej., para ciclos a partir de las líneas de la tabla. En la línea de la tabla con el número 0 se almacenan los datos del punto de partida.

Sintaxis

Activar la preparación del contorno:
```
CONTDCON(<Tabla para el contorno>,<Sentido de mecanizado>)
```

Desactivación de la preparación del contorno y conmutación al modo de ejecución normal de programa:
```
EXECUTE(<ERROR>)
```

Ver "Desactivación de la preparación del contorno (EXECUTE)"

Descripción

CONTDCON	Comando para activar la preparación del contorno para crear una tabla para el contorno codificada		
<Tabla para el contorno>	Nombre de la tabla para el contorno		
<Sentido de mecanizado>	Parámetro para el sentido de mecanizado		
	Tipo:	INT	
	Valor:	0	Preparación del contorno según el orden de las secuencias del contorno (valor estándar)
		1	inadmisible

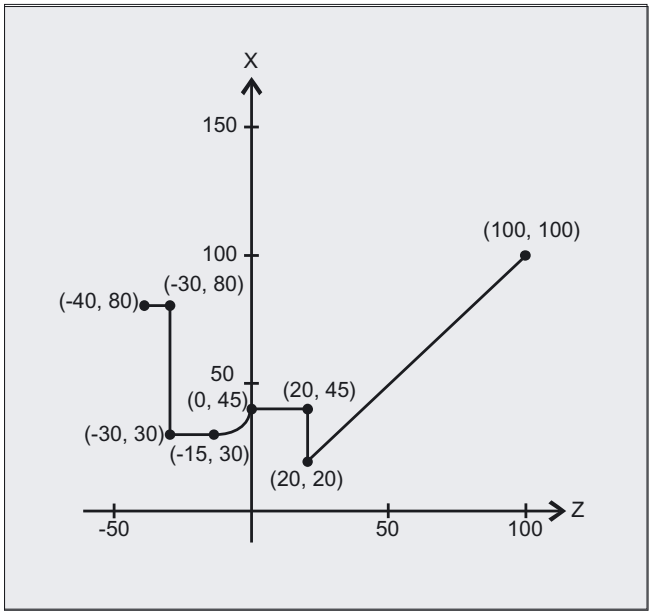
Nota

Los códigos G admitidos para CONTDCON en la parte de programa a tabular son de más volumen que para CONTPRON. Además se guardarán también los avances y el tipo de avance por cada sección del contorno.

Ejemplo

Creación de una tabla para el contorno con:

- Nombres "KTAB",
- Elementos de contorno (círculos, rectas)
- Modo de operación: Rotación
- Dirección de mecanizado: Adelante



Programa CN:

Código del programa	**Comentarios**
N10 DEF REAL KTAB[9,6]	; Tabla para el contorno con nombres KTAB y 9 líneas de tabla. Éstas permiten 8 secuencias de contornos. El valor de parámetro 6 (número de columnas de la tabla) es un valor fijo.
N20 DEF INT MODE = 0	; Variable para el sentido de mecanizado. Valor por defecto 0: sólo en el sentido de contorno programado.
N30 DEF INT ERROR = 0	; Variable para señalización de error.
...	
N100 G18 G64 G90 G94 G710	

```
Código del programa                          Comentarios
N101 G1 Z100 X100 F1000
N105 CONTDCON (KTAB, MODE)                   ; Llamada de la preparación del contorno
                                               (MODE se puede omitir).
N110 G1 Z20 X20 F200                         ; Descripción del contorno.
N120 G9 X45 F300
N130 Z0 F400
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)F100
N150 G64 Z-30 F600
N160 X80 F700
N170 Z-40 F800
N180 EXECUTE(ERROR)                          ; Finalización de la introducción de
                                               datos en la tabla para el contorno,
                                               conmutación a la ejecución normal del
                                               programa
...
```

Tabla para el contorno KTAB:

	Índice de columna					
	0	1	2	3	4	5
Índice de líneas	Modo de contorno	Punto final, abscisa	Punto final, ordenada	Centro, abscisa	Centro, ordenada	Avance
0	30	100	100	0	0	7
1	11031	20	20	0	0	200
2	111031	20	45	0	0	300
3	11031	0	45	0	0	400
4	11032	-15	30	-15	45	100
5	11031	-30	30	0	0	600
6	11031	-30	80	0	0	700
7	11031	-40	80	0	0	800
8	0	0	0	0	0	0

Explicación del contenido de las columnas:

Línea 0: Codificaciones para el **punto inicial:**

Columna 0:	10^0 (unidades): G0 = 0
	10^1 (decenas): G70 = 0, G71 = 1, G700 = 2, G710 = 3
Columna 1:	Punto inicial, abscisa
Columna 2:	Punto inicial ordenada
Columna 3-4:	0
Columna 5:	Índice de línea de la última sección del contorno en la tabla

Líneas 1-n: Entradas de las **secciones del contorno**

Columna 0:	10^0 (unidades): G0 = 0, G1 = 1, G2 = 2, G3 = 3
	10^1 (decenas): G70 = 0, G71 = 1, G700 = 2, G710 = 3
	10^2 (centenas): G90 = 0, G91 = 1
	10^3 (millares): G93 = 0, G94 = 1, G95 = 2, G96 = 3
	10^4 (decenas de millar): G60 = 0, G44 = 1, G641 = 2, G642 = 3
	10^5 (centenas de millar): G9 = 1
Columna 1:	Punto final, abscisa
Columna 2:	Punto final ordenada
Columna 3:	Centro abscisa en la interpolación circular
Columna 4:	Centro ordenada en la interpolación circular
Columna 5:	Avance

Información adicional

Órdenes de desplazamiento permitidas, sistema de coordenadas

Para la programación de contornos se admiten los siguientes grupos y comandos G:

Grupo G 1:	G0, G1, G2, G3
Grupo G 10:	G60, G64, G641, G642
Grupo G 11:	G9
Grupo G 13:	G70, G71, G700, G710
Grupo G 14:	G90, G91
Grupo G 15:	G93, G94, G95, G96, G961

También son posibles:

- Redondeo y chaflán,
- Programación del círculo mediante CIP y CT

Las funciones Spline, Polinomio y Rosca producen errores.

No se pueden realizar cambios en el sistema de coordenadas mediante la activación de un frame entre las instrucciones CONTDCON y EXECUTE. Lo mismo rige para un cambio entre G70 y G71 o entre G700 y G710.

Cualquier cambio de los ejes geométricos con GEOAX durante la preparación de la tabla para el contorno origina una alarma.

Sentido de mecanizado

La tabla para el contorno generada con CONTDCON está pensada para el desbaste en la dirección programada del contorno.

15.4 Determinación del punto de intersección entre dos elementos de contorno (INTERSEC)

Función

INTERSEC calcula el punto de intersección o corte de dos elementos normalizados del contorno, obtenidos de tablas para el contorno generadas mediante CONTPRON.

Sintaxis

```
<Estado>=INTERSEC(<Tabla para el contorno_1>[<Elemento de
contorno_1>],
<Tabla para el contorno_2>[<Elemento de contorno_2>],<Punto de
intersección>,<Tipo de mecanizado>)
```

Descripción

INTERSEC	Palabra reservada para determinar el punto de intersección de dos elementos de contorno obtenidos de tablas para el contorno generadas mediante CONTPRON.		
<Estado>	Variable para el estado del punto de intersección		
	Tipo:	BOOL	
	Valor:	TRUE	Se ha encontrado el punto de intersección
		FALSE	no se ha encontrado punto de intersección
<Tabla para el contorno_1>	Nombre de la primera tabla para el contorno		
<Elemento de contorno_1>	Número del elemento de contorno de la primera tabla para el contorno		
<Tabla para el contorno_2>	Nombre de la segunda tabla para el contorno		
<Elemento de contorno_2>	Número del elemento de contorno de la segunda tabla para el contorno		
<Punto de intersección>	Coordenadas del punto de intersección en el plano activo (G17/G18/G19)		
	Tipo:	REAL	
<Tipo de mecanizado>	Parámetros del tipo de mecanizado		
	Tipo:	INT	
	Valor:	0	Cálculo del punto de intersección en el plano activo con el parámetro 2 (valor estándar)
		1	Cálculo del punto de intersección independiente del plano transferido

Nota

Tenga en cuenta que las variables deben estar definidas antes de su utilización.

La transferencia de los contornos requiere el cumplimiento de los valores definidos con CONTPRON:

Parámetros	Descripción
2	Codificación del modo de contorno para el movimiento
3	Punto inicial de contorno, abscisa
4	Punto inicial de contorno, ordenada
5	Punto final de contorno, abscisa
6	Punto final de contorno, ordenada
9	Coordenada central de la abscisa (sólo en contornos circulares)
10	Coordenada central de la ordenada (sólo en contornos circulares)

Ejemplo

Cálculo del punto de corte del elemento del contorno 3 de la tabla TABNAME1 y del elemento del contorno 7 de la tabla TABNAME2. Las coordenadas del punto de intersección en el plano activo se almacenan en la variable ISCOORD (1er elemento = abscisa, 2º elemento = ordenada). En el caso de no existir un punto de intersección se realiza un salto a NOCORTA (no existe punto de intersección).

```
Código del programa                                                    Comentarios
DEF REAL TABNAME1[12,11]                                               ; Tabla para el contorno 1
DEF REAL TABNAME2[10,11]                                               ; Tabla para el contorno 2
DEF REAL ISCOORD [2]                                                   ; Variable para las coordenadas
                                                                         del punto de intersección.
DEF BOOL ISPOINT                                                       ; Variable para el estado del
                                                                         punto de corte.
DEF INT MODE                                                           ; Variable para el tipo de
                                                                         mecanizado.
…
MODE=1                                                                 ; Cálculo con independencia del
                                                                         plano activo.
N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7],ISCOORD,MODE)             ; Llamada del punto de
                                                                         intersección de los elementos
                                                                         de contorno.
N20 IF ISPOINT==FALSE GOTOF NOCORTE                                    ; Salto a NOCORTA.
…
```

15.5 Retirada de los elementos de contorno de una tabla secuencia a secuencia (EXECTAB)

Función

El comando EXECTAB permite retirar secuencia a secuencia los elementos de contorno de una tabla creada, p. ej., con el comando CONTPRON.

Sintaxis

```
EXECTAB(<Tabla para el contorno>[<Elemento de contorno>])
```

Descripción

EXECTAB	Comando para retirar un elemento de contorno
<Tabla para el contorno>	Nombre de la tabla para el contorno
<Elemento de contorno>	Número del elemento de contorno

Ejemplo

Los elementos de contorno 0 a 2 de la tabla KTAB deben retirarse secuencia a secuencia.

Código del programa	**Comentarios**
N10 EXECTAB(KTAB[0])	; El elemento 0 de la tabla KTAB se desplaza.
N20 EXECTAB(KTAB[1])	; El elemento 1 de la tabla KTAB se desplaza.
N30 EXECTAB(KTAB[2])	; El elemento 2 de la tabla KTAB se desplaza.

15.6 Cálculo de datos de circunferencia (CALCDAT)

Función

El comando `CALCDAT` permite calcular el radio y las coordenadas del centro de una circunferencia a partir de tres o cuatro puntos conocidos de ésta. Los puntos indicados deben de ser diferentes. Para el cálculo utilizando 4 puntos que no se encuentran exactamente sobre la circunferencia se utiliza un valor medio para el cálculo del centro y radio de la circunferencia.

Sintaxis

```
<Estado>=CALCDAT(<Puntos de
circunferencia>[<Cantidad>,<Tipo>],<Cantidad>,<Resultado>)
```

Descripción

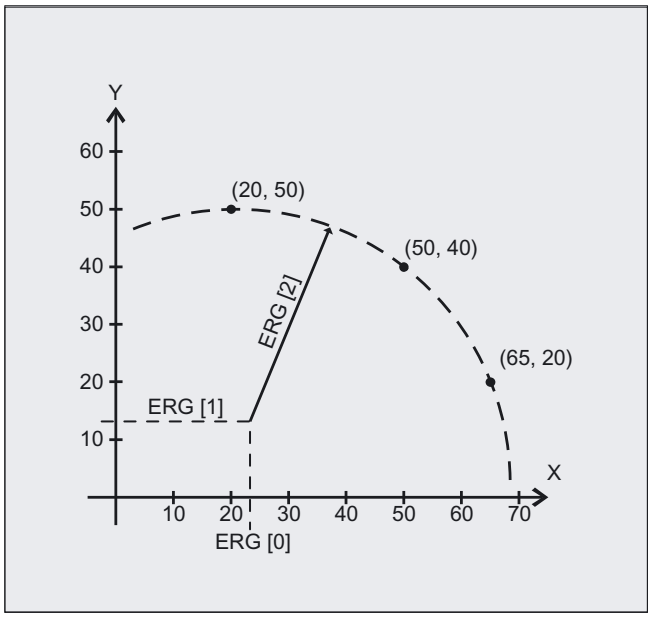
`CALCDAT`	Comando para calcular el radio y las coordenadas del centro de una circunferencia a partir de 3 ó 4 puntos.		
`<Estado>`	Variable para el estado del cálculo de circunferencia		
	Tipo:	BOOL	
	Valor:	TRUE	Los puntos indicados se encuentran en una circunferencia.
		FALSE	Los puntos indicados **no** se encuentran en una circunferencia.
`<Puntos de circunferencia>[]`	Variable para indicar los puntos de circunferencia con los parámetros:		
	`<Cantidad>`	Cantidad de puntos de circunferencia (3 ó 4)	
	`<Tipo>`	Tipo del dato de coordenadas, p. ej., 2 para coordenadas de 2 puntos	
`<Cantidad>`	Parámetro para la cantidad de puntos utilizados para el cálculo (3 ó 4)		
`<Resultado>[3]`	Variable para el resultado:		
	Indicación de coordenadas de centro de una circunferencia y radio		
	0	Coordenadas del centro de una circunferencia: valor de abscisas	
	1	Coordenadas del centro de una circunferencia: valor de ordenadas	
	2	Radio	

Nota

Tenga en cuenta que las variables deben estar definidas antes de su utilización.

Ejemplo

A partir de tres puntos debe determinarse si éstos se encuentran en un segmento circular.



Código del programa	**Comentarios**
N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)	; Variable para la indicación de puntos de circunferencia
N20 DEF REAL RESULT[3]	; Variable para resultado
N30 DEF BOOL STATUS	; Variable para el estado
N40 STATUS=CALCDAT(PKT,3,RESULT)	; Llamada de los datos de circunferencia determinados.
N50 IF STATUS == FALSE GOTOF ERROR	; Salto a error

15.7 Desactivación de la preparación del contorno (EXECUTE)

Función

El comando `EXECUTE` desactiva la preparación del contorno y conmuta al mismo tiempo al modo de ejecución normal de programa.

Sintaxis

```
EXECUTE(<ERROR>)
```

Descripción

`EXECUTE`	Comando para finalizar la preparación del contorno
`<ERROR>`	Variable para señalización de error Tipo: INT El valor de las variables indica si el contorno se ha preparado sin errores: 0 Fallo 1 Sin errores

Ejemplo

Código del programa

```
...
N30 CONTPRON(...)
N40 G1 X... Z...
...
N100 EXECUTE(...)
...
```

Tablas

16

16.1 Lista de instrucciones

Leyenda:

1) Referencia al documento que contiene la descripción detallada de la instrucción:

PG	Manual de programación Fundamentos
PGA	Manual de programación Preparación del trabajo
BHD	Manual del usuario HMI sl Torneado
BHF	Manual del usuario HMI sl Fresado
FB1 ( )	Manual de funciones básicas (con la abreviatura alfanumérica de la descripción de la respectiva función entre paréntesis)
FB2 ( )	Manual de funciones de ampliación (con la abreviatura alfanumérica de la descripción de la respectiva función entre paréntesis)
FB3 ( )	Manual de funciones especiales (con la abreviatura alfanumérica de la descripción de la respectiva función entre paréntesis)
FBSI	Manual de funciones Safety Integrated
FBSY	Manual de funciones Acciones síncronas
FBW	Manual de funciones Gestión de herramientas

2) Efecto de la instrucción:

m	modal
s	secuencia a secuencia

3) Disponibilidad en SINUMERIK 828D (D = Torneado, F = Fresado):

●	Estándar
○	Opción
-	No disponible

4) Ajuste por defecto al principio del programa (así viene de fábrica el control numérico, si no hay nada diferente programado)

Instrucción	Significado	Consultar la descripción en 1)	W 2)	828D 3) PPU260/261 - - - PPU280/281			
				D - - - - -	F - - - - -	D - - - - -	F
:	Número de secuencia principal CN, fin de marcas de salto, operador de concatenación	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
*	Operador de multiplicación	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
+	Operador de suma	*PGA* Funciones de cálculo (Página 63)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
-	Operador de resta	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
<	Operador de comparación, menor que	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
<<	Operador de concatenación para cadenas de caracteres	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
<=	Operador de comparación, menor o igual que	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
=	Operador de asignación	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
>=	Operador de comparación, mayor o igual que	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
/	Operador de división	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
/0 … … /7	La secuencia se omite (1er plano opcional) La secuencia se omite (8º plano opcional)	*PG*		● ○	● ○	● ○	● ○
A	Nombre de eje	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	m/s	●	●	●	●
A2	Orient. herramienta: Ángulo euleriano o RPY	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	s	●	●	●	●
A3	Orient. herramienta: Componente vectorial normal a la dirección/ superficie	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	s	●	●	●	●
A4	Orient. herramienta: Vector normal a la superficie para el principio de la secuencia	*PGA* Fresado frontal (fresado 3D A4, B4, C4, A5, B5, C5) (Página 345)	s	●	●	●	●
A5	Orient. herramienta: Vector normal a la superficie para el final de la secuencia	*PGA* Fresado frontal (fresado 3D A4, B4, C4, A5, B5, C5) (Página 345)	s	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
ABS	Valor absoluto	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
AC	Acotado absoluto de coordenadas/posiciones	*PG*	s	●	●	●	●
ACC	Influencia en la aceleración axial actual	*PG*	m	●	●	●	●
ACCLIMA	Influencia en la aceleración axial máxima actual	*PG*	m	●	●	●	●
ACN	Acotado absoluto para ejes giratorios, aproximación a la posición en dirección negativa	*PG*	s	●	●	●	●
ACOS	Arcocoseno (función trigon.)	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
ACP	Acotado absoluto para ejes giratorios, aproximación a la posición en dirección positiva	*PG*	s	●	●	●	●
ACTBLOCNO	Emisión del número de secuencia actual de una secuencia de alarma, incluso aunque la función "Suprimir la visualización de secuencia actual" (DISPLOF) esté activa.	*PGA* Suprimir la visualización de secuencia actual (DISPLOF, DISPLON, ACTBLOCNO) (Página 172)		●	●	●	●
ADDFRAME	Inclusión y, en su caso, activación de un frame medido	*PGA, FB1(K2)* Cálculo de frame a partir de 3 puntos de medida en el espacio (MEAFRAME) (Página 309)		●	●	●	●
ADIS	Distancia de matado para funciones de contorneado G1, G2, G3, ...	*PG*	m	●	●	●	●
ADISPOS	Distancia de redondeo (matado de esquinas) para desplazamiento rápido G0	*PG*	m	●	●	●	●
ADISPOSA	Magnitud de la ventana de tolerancia para IPOBRKA	*PGA* Criterio de fin del movimiento programable (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 283)	m	●	●	●	●
ALF	Ángulo de retirada rápida	*PGA* Retirada rápida del contorno (SETINT LIFTFAST, ALF) (Página 121)	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
AMIRROR	Simetría programable	*PG*	s	•	•	•	•
AND	Y lógico	*PGA* Operaciones de comparación y operaciones lógicas (Página 66)		•	•	•	•
ANG	Ángulo de sucesión de contorno	*PG*	s	•	•	•	•
AP	Ángulo polar	*PG*	m/s	•	•	•	•
APR	Lectura/visualización de la protección de acceso	*PGA* Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APRB	Lectura de derechos de acceso, BTSS	*PGA* Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APRP	Lectura de derechos de acceso, programa de pieza	*PGA* Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APW	Escritura de la protección de acceso	*PGA* Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APWB	Escritura de derechos de acceso, BTSS	*PGA* Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APWP	Escritura de derechos de acceso, programa de pieza	*PGA* Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APX	Definición de la protección de acceso para la ejecución del elemento de lenguaje indicado	*PGA* Atributo: derechos de acceso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
AR	Ángulo en el vértice	*PG*	m/s	•	•	•	•
AROT	Rotación programable	*PG*	s	•	•	•	•
AROTS	Rotaciones de frames programables con ángulos espaciales	*PG*	s	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
AS	Definición de macro	*PGA* Técnica de macros (DEFINE ... AS) (Página 208)		●	●	●	●
ASCALE	Escala programable	*PG*	s	●	●	●	●
ASIN	Función de cálculo, arcoseno	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
ASPLINE	Spline de Akima	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
ATAN2	Arcotangente 2	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
ATOL	Tolerancia específica del eje para funciones de compresor, alisado de la orientación y tipos de matado de esquinas	*PGA* Tolerancia de orientación/ contorno programable (CTOL, OTOL, ATOL) (Página 515)		-	●	-	●
ATRANS	Decalaje aditivo programable	*PG*	s	●	●	●	●
AX	Identificador de eje variable	*PGA* Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 701)	m/s	●	●	●	●
AXCSWAP	Activar contenedor de ejes	*PGA* Contenedor de ejes (AXCTSWE, AXCTSWED) (Página 709)		-	-	-	-
AXCTSWE	Activar contenedor de ejes	*PGA* Contenedor de ejes (AXCTSWE, AXCTSWED) (Página 709)		-	-	-	-
AXCTSWED	Rotación del contenedor de ejes	*PGA* Contenedor de ejes (AXCTSWE, AXCTSWED) (Página 709)		-	-	-	-
AXIS	Identificador de eje, dirección de eje	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
AXNAME	Conversión de la cadena de caracteres introducida en un identificador de eje	*PGA* Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 701)		•	•	•	•
AXSTRING	Conversión de la cadena de caracteres en un número de cabezal	*PGA* Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 701)		•	•	•	•
AXTOCHAN	Solicitar un eje para un determinado canal. Es posible desde el programa de CN y desde una acción síncrona.	*PGA* Transferir el eje a otro canal (AXTOCHAN) (Página 133)		•	•	•	•
AXTOSPI	Conversión del identificador de eje en un índice de cabezal	*PGA* Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 701)		•	•	•	•
B	Nombre de eje	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	m/s	•	•	•	•
B2	Orient. herramienta: Ángulo euleriano o RPY	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	s	•	•	•	•
B3	Orient. herramienta: Componente vectorial normal a la dirección/ superficie	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	s	•	•	•	•
B4	Orient. herramienta: Vector normal a la superficie para el principio de la secuencia	*PGA* Fresado frontal (fresado 3D A4, B4, C4, A5, B5, C5) (Página 345)	s	•	•	•	•
B5	Orient. herramienta: Vector normal a la superficie para el final de la secuencia	*PGA* Fresado frontal (fresado 3D A4, B4, C4, A5, B5, C5) (Página 345)	s	•	•	•	•
B_AND	Y binario	*PGA* Operaciones de comparación y operaciones lógicas (Página 66)		•	•	•	•
B_OR	O binario	*PGA* Operaciones de comparación y operaciones lógicas (Página 66)		•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
B_NOT	Negación binaria	*PGA* Operaciones de comparación y operaciones lógicas (Página 66)		●	●	●	●
B_XOR	O exclusivo binario	*PGA* Operaciones de comparación y operaciones lógicas (Página 66)		●	●	●	●
BAUTO	Definición del primer segmento spline mediante los 3 puntos siguientes	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
BLOCK	Define junto con la palabra reservada TO la parte ejecutable del programa en una ejecución indirecta del subprograma	*PGA* Llamada de programa indirecta con indicación de la sección de programa que se va a ejecutar (CALL BLOCK ... TO ...) (Página 194)		●	●	●	●
BLSYNC	Ejecución de la rutina de interrupción debe primero comenzar con el siguiente cambio de secuencia	*PGA* Asignar e iniciar una rutina de interrupción (SETINT, PRIO, BLSYNC) (Página 117)		●	●	●	●
BNAT [4]	Transición natural a la primera secuencia spline	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
BOOL	Tipo de dato: Valor binario TRUE/FALSE o bien 1/0	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
BOUND	Comprueba si el valor se encuentra dentro del margen de valores definido. En caso de igualdad se devuelve el valor de comprobación.	*PGA* Mínimo, máximo y rango de variables (MINVAL, MAXVAL, BOUND) (Página 70)		●	●	●	●
BRISK [4]	Aceleración de contorneado de forma escalonada	*PG*	m	●	●	●	●
BRISKA	Activar la aceleración en escalón en contorneo para los ejes programados	*PG*		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
BSPLINE	B-Spline	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
BTAN	Transición tangencial a la primera secuencia spline	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
C	Nombre de eje	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	m/s	●	●	●	●
C2	Orient. herramienta: Ángulo euleriano o RPY	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	s	●	●	●	●
C3	Orient. herramienta: Componente vectorial dirección/normal a la superficie	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	s	●	●	●	●
C4	Orient. herramienta: Vector normal a la superficie para el principio de la secuencia	*PGA* Fresado frontal (fresado 3D A4, B4, C4, A5, B5, C5) (Página 345)	s	●	●	●	●
C5	Orient. herramienta: Vector normal a la superficie para el final de la secuencia	*PGA* Fresado frontal (fresado 3D A4, B4, C4, A5, B5, C5) (Página 345)	s	●	●	●	●
CAC	Desplazamiento a una posición absoluta	*PGA* Desplazamiento a posiciones codificadas (CAC, CIC, CDC, CACP, CACN) (Página 237)		●	●	●	●
CACN	Se efectúa una aproximación absoluta en dirección negativa al valor consignado en la tabla	*PGA* Desplazamiento a posiciones codificadas (CAC, CIC, CDC, CACP, CACN) (Página 237)		●	●	●	●
CACP	Se efectúa una aproximación absoluta en dirección positiva al valor consignado en la tabla	*PGA* Desplazamiento a posiciones codificadas (CAC, CIC, CDC, CACP, CACN) (Página 237)		●	●	●	●
CALCDAT	Calcula el radio y el centro de una circunferencia a partir de 3 ó 4 puntos	*PGA* Cálculo de datos de circunferencia (CALCDAT) (Página 750)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CALCPOSI	Comprobación con respecto a vulneraciones de zonas protegidas, limitación del campo de trabajo y límites de software	*PGA* Comprobación con respecto a vulneraciones de zonas protegidas, limitación del campo de trabajo y límites de software (CALCPOSI) (Página 229)		●	●	●	●
CALL	Llamada indirecta de subprograma	*PGA* Llamada de subprograma indirecta (CALL) (Página 193)		●	●	●	●
CALLPATH	Ruta de búsqueda programable en llamadas de subprograma	*PGA* Ampliación de la ruta de búsqueda en la llamada de subprograma (CALLPATH) (Página 198)		●	●	●	●
CANCEL	Interrumpir la acción síncrona modal	*PGA* Borrar acción síncrona (CANCEL) (Página 664)		●	●	●	●
CASE	Bifurcación de programa condicionada	*PGA* Bifurcación de programa (CASE ... OF ... DEFAULT ...) (Página 89)		●	●	●	●
CDC	Desplazamiento directo a una posición	*PGA* Desplazamiento a posiciones codificadas (CAC, CIC, CDC, CACP, CACN) (Página 237)		●	●	●	●
CDOF [4]	Vigilancia de colisión DES	*PG*	m	●	●	●	●
CDOF2	Vigilancia de colisión DES, en el fresado periférico 3D	*PG*	m	●	●	●	●
CDON	Vigilancia de colisión CON	*PG*	m	●	●	●	●
CFC [4]	Avance constante en el contorno	*PG*	m	●	●	●	●
CFIN	Avance constante sólo con curvatura interna, no con curvatura externa	*PG*	m	●	●	●	●
CFINE	Asignación del decalaje fino a una variable FRAME	*PGA* Decalaje basto y fino (CFINE, CTRANS) (Página 304)		●	●	●	●
CFTCP	Avance constante en el punto de referencia del filo de herramienta, trayectoria del centro	*PG*	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CHAN	Especificación del ámbito de vigencia de datos	*PGA* Definición de variables de usuario (DEF) (Página 24)		•	•	•	•
CHANDATA	Ajustar el número de canal para accesos a datos de canal	*PGA* Memoria de trabajo (CHANDATA, COMPLETE, INITIAL) (Página 216)		•	•	•	•
CHAR	Tipo de dato: Caracteres ASCII	*PGA* Definición de variables de usuario (DEF) (Página 24)		•	•	•	•
CHECKSUM	Forma la suma de chequeo en una matriz como STRING con una longitud definida	*PGA* Cálculo de la suma de control mediante una matriz (CHECKSUM) (Página 149)		•	•	•	•
CHF	Chaflán; Valor = Longitud del chaflán	*PG*	s	•	•	•	•
CHKDM	Comprobación de la univocidad dentro de un almacén	*FBW*		•	•	•	•
CHKDNO	Prueba de unicidad de los números D	*PGA* Asignación libre de números D: Comprobar números D (CHKDNO) (Página 455)		•	•	•	•
CHR	Chaflán; Valor = longitud del chaflán en el sentido de desplazamiento	*PG*		•	•	•	•
CIC	Desplazamiento a una posición incremental	*PGA* Desplazamiento a posiciones codificadas (CAC, CIC, CDC, CACP, CACN) (Página 237)		•	•	•	•
CIP	Interpolación circular a través de punto intermedio	*PG*	m	•	•	•	•
CLEARM	Resetear una/varias metas para la coordinación de canales	*PGA* Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 109)		-	-	-	-
CLRINT	Interrupción de la selección	*PGA* Borrado de la asignación de una rutina de interrupción (CLRINT) (Página 120)		•	•	•	•
CMIRROR	Simetría respecto a un eje del sistema de coordenadas	*PGA* Funciones de cálculo (Página 63)		•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
COARSEA	Fin de movimiento al alcanzar la "Parada precisa BASTA"	*PGA* Criterio de fin del movimiento programable (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 283)	m	●	●	●	●
COMPCAD	Compresor CON: Calidad de superficie optimizada en programas CAD	*PGA* Compresión de secuencias CN (COMPON, COMPCURV, COMPCAD, COMPOF) (Página 254)	m	-	○	-	○
COMPCURV	Compresor CON: Polinomios de curvatura continua	*PGA* Compresión de secuencias CN (COMPON, COMPCURV, COMPCAD, COMPOF) (Página 254)	m	-	○	-	○
COMPLETE	Instrucción de control para lectura y escritura de datos	PGA Memoria de trabajo (CHANDATA, COMPLETE, INITIAL) (Página 216)		●	●	●	●
COMPOF [4]	Compresor DES	*PGA* Compresión de secuencias CN (COMPON, COMPCURV, COMPCAD, COMPOF) (Página 254)	m	-	○	-	○
COMPON	Compresor CON	*PGA* Compresión de secuencias CN (COMPON, COMPCURV, COMPCAD, COMPOF) (Página 254)		-	○	-	○
CONTDCON	Codificación de contornos en forma de tabla CON	*PGA* Creación de tabla para el contorno codificada (CONTDCON) (Página 742)		●	●	●	●
CONTPRON	Activar preparación de referencia	*PGA* Creación de tabla para el contorno (CONTPRON) (Página 734)		●	●	●	●
CORROF	Todos los movimientos superpuestos activos se cancelan.	*PG*		●	●	●	●
COS	Coseno (función trigon.)	*PGA* Funciones de cálculo (Página 63)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
COUPDEF	Definición del conjunto de reductores electrónicos/ conjunto de cabezales síncronos	*PGA* Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 563)		○	-	○	-
COUPDEL	Borrado de un conjunto de reductores electrónicos	*PGA* Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 563)		○	-	○	-
COUPOF	Conjunto de reductores electrónicos/par de cabezales síncronos CON	*PGA* Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 563)		○	-	○	-
COUPOFS	Desactivación de un conjunto de reductores electrónicos/par de cabezales síncronos con parada del cabezal esclavo	*PGA* Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 563)		○	-	○	-
COUPON	Conjunto de reductores electrónicos/par de cabezales síncronos CON	*PGA* Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 563)		○	-	○	-
COUPONC	Activación de un conjunto de reductores electrónicos/par de cabezales síncronos con aplicación de la programación anterior	*PGA* Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 563)		○	-	○	-
COUPRES	Reseteo de un conjunto de reductores electrónicos	*PGA* Cabezal síncrono: Programación (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 563)		○	-	○	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CP	Movimiento interpolado	*PGA* Desplazamiento PTP cartesiano (Página 395)	m	●	●	●	●
CPRECOF [4]	Precisión de contorno programable DES	*PG*	m	●	●	●	●
CPRECON	Precisión de contorno programable CON	*PG*	m	●	●	●	●
CPROT	Zona protegida específica de canal on/off	*PGA* Activar/desactivar zonas protegidas (CPROT, NPROT) (Página 225)		●	●	●	●
CPROTDEF	Definición de una zona protegida específica del canal	*PGA* Definición de las zonas protegidas (CPROTDEF, NPROTDEF) (Página 221)		●	●	●	●
CR	Radio del círculo	*PG*	s	●	●	●	●
CROT	Giro del sistema de coordenadas actual	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
CROTS	Rotaciones de frames programables con ángulos espaciales (rotación en los ejes indicados)	*PG*	s	●	●	●	●
CRPL	Rotación de frame en un plano cualquiera	*FB1(K2)*		●	●	●	●
CSCALE	Factor de escala para varios ejes	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
CSPLINE	Spline cúbico	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
CT	Círculo con transición tangencial	*PG*	m	●	●	●	●
CTAB	Averigua posición del eje esclavo a partir de la posición del eje maestro de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CTABDEF	Activación definición de tabla	*PGA* Definición de tablas de levas (CTABDEF, CATBEND) (Página 527)		-	-	-	-
CTABDEL	Borrar tabla de levas	*PGA* Borrado de tablas de levas (CTABDEL) (Página 535)		-	-	-	-
CTABEND	Desactivación definición de tabla	*PGA* Definición de tablas de levas (CTABDEF, CATBEND) (Página 527)		-	-	-	-
CTABEXISTS	Comprueba la tabla de levas con el número n	*PGA* Comprobación de la existencia de una tabla de levas (CTABEXISTS) (Página 535)		-	-	-	-
CTABFNO	Número de tablas de levas todavía posibles en la memoria	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABFPOL	Número de polinomios posibles en la memoria	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABFSEG	Número de segmentos de curvas todavía posibles en la memoria	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABID	Indica el número de la enésima tabla de levas	*PGA* Tablas de levas: determinación de las propiedades de la tabla (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) (Página 539)		-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CTABINV	Averigua posición del eje maestro a partir de la posición del eje esclavo de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-
CTABISLOCK	Devuelve el estado de bloqueo de la tabla de levas con el número n	*PGA* Tablas de levas: determinación de las propiedades de la tabla (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) (Página 539)		-	-	-	-
CTABLOCK	Borrar y sobrescribir, bloquear	*PGA* Bloqueo de las tablas de levas contra borrado y sobrescritura (CTABLOCK, CTABUNLOCK) (Página 537)		-	-	-	-
CTABMEMTYP	Devuelve la memoria en la cual se ha creado la tabla de levas con el número n	*PGA* Tablas de levas: determinación de las propiedades de la tabla (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) (Página 539)		-	-	-	-
CTABMPOL	Número máximo de polinomios posibles en la memoria	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABMSEG	Número máximo de segmentos de curvas posibles en la memoria	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABNO	Número de tablas de levas definidas en la memoria SRAM o DRAM	*FB3(M3)*		-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CTABNOMEM	Número de tablas de levas definidas en la memoria SRAM o DRAM	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABPERIOD	Devuelve la periodicidad de la tabla de levas con el número n	*PGA* Tablas de levas: determinación de las propiedades de la tabla (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) (Página 539)		-	-	-	-
CTABPOL	Número de polinomios ya utilizados en la memoria	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABPOLID	Número de polinomios de leva utilizados por la tabla de levas con el número n	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABSEG	Número de segmentos de curva ya utilizados en la memoria	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CTABSEGID	Número de segmentos de curva utilizados por la tabla de levas con el número n	*PGA* Tablas de levas: comprobación del uso de los recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 546)		-	-	-	-
CTABSEV	Suministra el valor final del eje esclavo de un segmento de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-
CTABSSV	Suministra el valor inicial del eje esclavo de un segmento de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-
CTABTEP	Suministra el valor del eje maestro al final de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-
CTABTEV	Suministra el valor del eje esclavo al final de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-
CTABTMAX	Suministra el valor máximo del eje esclavo de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CTABTMIN	Suministra el valor mínimo del eje esclavo de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-
CTABTSP	Suministra el valor del eje maestro al inicio de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-
CTABTSV	Suministra el valor del eje esclavo al inicio de la tabla de levas	*PGA* Lectura de valores de las tablas de levas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 540)		-	-	-	-
CTABUNLOCK	Cancelación del bloqueo de borrado y sobrescritura	*PGA* Bloqueo de las tablas de levas contra borrado y sobrescritura (CTABLOCK, CTABUNLOCK) (Página 537)		-	-	-	-
CTOL	Tolerancia de contorno para funciones de compresor, alisado de la orientación y tipos de matado de esquinas	*PGA* Tolerancia de orientación/contorno programable (CTOL, OTOL, ATOL) (Página 515)		-	○	-	○
CTRANS	Decalaje de origen para varios ejes	*PGA* Decalaje basto y fino (CFINE, CTRANS) (Página 304)		●	●	●	●
CUT2D [4]	Corrección de herramienta 2D	*PG*	m	●	●	●	●
CUT2DF	Corrección de herramienta 2D: la corrección de herramienta es relativa al frame actual (plano inclinado).	*PG*	m	●	●	●	●
CUT3DC	Corrección de herramienta 3D fresado periférico	*PGA* Activación correcciones de herramienta 3D (CUT3DC..., CUT3DF...) (Página 433)	m	-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
CUT3DCC	Corrección de herramienta 3D, fresado periférico con superficies límite	*PGA* Corrección de herramienta 3D: Consideración de una superficie de limitación (CUT3DCC, CUT3DCCD) (Página 443)	m	-	-	-	-
CUT3DCCD	Corrección de herramienta 3D, fresado periférico con superficies límite y herramienta diferencial	*PGA* Corrección de herramienta 3D: Consideración de una superficie de limitación (CUT3DCC, CUT3DCCD) (Página 443)	m	-	-	-	-
CUT3DF	Corrección de herramienta 3D, fresado frontal	*PGA* Activación correcciones de herramienta 3D (CUT3DC..., CUT3DF...) (Página 433)	m	-	-	-	-
CUT3DFF	Corrección de herramienta 3D, fresado frontal con orientación de herramienta constante dependiente del frame activo	*PGA* Activación correcciones de herramienta 3D (CUT3DC..., CUT3DF...) (Página 433)	m	-	-	-	-
CUT3DFS	Corrección de herramienta 3D, fresado frontal con orientación de herramienta constante independiente del frame activo	*PGA* Activación correcciones de herramienta 3D (CUT3DC..., CUT3DF...) (Página 433)	m	-	-	-	-
CUTCONOF [4]	Corrección del radio constante DES	*PG*	m	●	●	●	●
CUTCONON	Corrección del radio constante CON	*PG*	m	●	●	●	●
CUTMOD	Activación de la función "modificación de los datos de corrección en herramientas girables"	*PGA* Modificación de datos de filos en herramientas giratorias (CUTMOD) (Página 471)		●	●	●	●
CYCLE...	Ciclos de medida	*BHD/BHF*					
D	Número de corrección de herramienta	*PG*		●	●	●	●
D0	Con D0 se invalidan las correcciones para la herramienta	*PG*		●	●	●	●
DAC	Programación por diámetros específica del eje, por secuencias, absoluta	*PG*	s	●	●	●	●
DC	Acotado absoluto para ejes giratorios, aproximación directa a la posición	*PG*	s	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
DEF	Definición de variables	*PGA* Definición de variables de usuario (DEF) (Página 24)		•	•	•	•
DEFINE	Palabra reservada para definiciones de macros	*PGA* Técnica de macros (DEFINE ... AS) (Página 208)		•	•	•	•
DEFAULT	Rama de la bifurcación CASE	*PGA* Bifurcación de programa (CASE ... OF ... DEFAULT ...) (Página 89)		•	•	•	•
DELAYFSTON	Definir el inicio de un área Stop-Delay	*PGA* Bifurcación de programa (CASE ... OF ... DEFAULT ...) (Página 89)	m	•	•	•	•
DELAYFSTOF	Definir el fin de una área Stop-Delay	*PGA* Secciones de programa con posibilidad de interrupción condicionada (DELAYFSTON, DELAYFSTOF) (Página 493)	m	•	•	•	•
DELDL	Borrar correcciones aditivas	*PGA* Borrar correcciones aditivas (DELDL) (Página 418)		•	•	•	•
DELDTG	Borrado de trayecto residual	*PGA* Borrado de trayecto residual (DELDTG) (Página 612)		•	•	•	•
DELETE	Borrar el fichero indicado. El nombre del fichero se puede indicar con la ruta y la identificación del fichero.	*PGA* Borrar fichero (DELETE) (Página 139)		•	•	•	•
DELTOOLENV	Borrar bloques de datos para la descripción de entornos de herramienta	*FB1(W1)*		•	•	•	•
DIACYCOFA	Programación por diámetros específica del eje, modal: DES en ciclos	*FB1(P1)*	m	•	•	•	•
DIAM90	Programación por diámetros para G90, programación por radios para G91	*PGA*	m	•	•	•	•
DIAM90A	Programación por diámetros específica del eje, modal, para G90 y AC, Programación por radios para G91 e IC	*PG*	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
DIAMCHAN	Aplicación de todos los ejes del DM Funciones de eje en el estado de canal de la programación por diámetros	*PG*		•	•	•	•
DIAMCHANA	Aplicación estado del canal de la programación por diámetros	*PG*		•	•	•	•
DIAMCYCOF	Programación por diámetros específica del canal: DES en ciclos	*FB1(P1)*	m	•	•	•	•
DIAMOF [4]	Programación por diámetros: DES Ajuste inicial: ver fabricante de la máquina	*PG*	m	•	•	•	•
DIAMOFA	Programación por diámetros específica del eje, modal: DES Ajuste inicial: ver fabricante de la máquina	*PG*	m	•	•	•	•
DIAMON	Programación por diámetros: CON	*PG*	m	•	•	•	•
DIAMONA	Programación por diámetros específica del eje, modal: CON Habilitación: ver fabricante de la máquina	*PG*	m	•	•	•	•
DIC	Programación por diámetros específica del eje, por secuencias, incremental	*PG*	s	•	•	•	•
DILF	Distancia de retirada (longitud)	*PG*	m	•	•	•	•
DISABLE	Desactivada la interrupción	*PGA* Desactivación/reactivación de la asignación de una rutina de interrupción (DISABLE, ENABLE) (Página 119)		•	•	•	•
DISC	Rebase circunferencia de transición; corrección de radio de herramienta	*PG*	m	•	•	•	•
DISCL	Distancia entre el punto final del movimiento de penetración rápido y el plano de mecanizado	*PG*		•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
DISPLOF	Suprimir la visualización de secuencia actual	*PGA* Suprimir la visualización de secuencia actual (DISPLOF, DISPLON, ACTBLOCNO) (Página 172)		•	•	•	•
DISPLON	Anular la supresión de la visualización de secuencia actual	*PGA* Suprimir la visualización de secuencia actual (DISPLOF, DISPLON, ACTBLOCNO) (Página 172)		•	•	•	•
DISPR	Diferencia trayectoria reposicionamiento	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	s	•	•	•	•
DISR	Distancia reposicio-namiento	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	s	•	•	•	•
DITE	Trayecto de salida de rosca	*PG*	m	•	•	•	•
DITS	Trayecto de entrada en rosca	*PG*	m	•	•	•	•
DIV	División entera	*PGA* Funciones de cálculo (Página 63)		•	•	•	•
DL	Seleccionar una corrección de herramienta aditiva dependiente de la posición (DL, suma de correcciones de preparación)	*PGA* Seleccionar correcciones aditivas (DL) (Página 415)	m	-	-	-	-
DO	Palabra reservada para acción síncrona, dispara la acción cuando se cumple la condición	*PGA* Acciones (DO) (Página 587)		•	•	•	•
DRFOF	Desactivación de los desplazamientos con volante (DRF)	*PG*	m	•	•	•	•
DRIVE	Aceleración de contorneado dependiente de la velocidad	*PG*	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
DRIVEA	Activar aceleración con perfil discontinuo por tramos para los ejes programados	*PG*		●	●	●	●
DYNFINISH	Dinámica para acabado fino	*PG*	m	●	●	●	●
DYNNORM	Dinámica normal	*PG*	m	●	●	●	●
DYNPOS	Dinámica para el modo de posicionado, roscado de taladros	*PG*	m	●	●	●	●
DYNROUGH	Dinámica para desbaste	*PG*	m	●	●	●	●
DYNSEMIFIN	Dinámica para acabado	*PG*	m	●	●	●	●
DZERO	Marca todos los números D de la unidad TO como no válidos	*PGA* Asignación libre de números D: Fijación como no válidos de los números D (DZERO) (Página 457)		●	●	●	●
EAUTO	Definición del último segmento spline mediante los 3 últimos puntos	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
EGDEF	Definición de un reductor electrónico	*PGA* Definición del reductor electrónico (EGDEF) (Página 554)		-	-	-	-
EGDEL	Borrar la definición de acoplamiento para el eje esclavo	*PGA* Borrar la definición de un reductor electrónico (EGDEL) (Página 561)		-	-	-	-
EGOFC	Desactivación continua del reductor electrónico	*PGA* Desactivación del reductor electrónico (EGOFS, EGOFC) (Página 559)		-	-	-	-
EGOFS	Desactivación selectiva del reductor electrónico	*PGA* Desactivación del reductor electrónico (EGOFS, EGOFC) (Página 559)		-	-	-	-
EGON	Activación del reductor electrónico	*PGA* Desactivación del reductor electrónico (EGOFS, EGOFC) (Página 559)		-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
EGONSYN	Activación del reductor electrónico	*PGA* Desactivación del reductor electrónico (EGOFS, EGOFC) (Página 559)		-	-	-	-
EGONSYNE	Activación del reductor electrónico, con definición del modo de aproximación	*PGA* Desactivación del reductor electrónico (EGOFS, EGOFC) (Página 559)		-	-	-	-
ELSE	Bifurcación de programa si no se cumple la condición IF	*PGA* Bucle de programa con alternativa (IF, ELSE, ENDIF) (Página 101)		●	●	●	●
ENABLE	Interrupción activada	*PGA* Desactivación/reactivación de la asignación de una rutina de interrupción (DISABLE, ENABLE) (Página 119)		●	●	●	●
ENAT [4]	Transición de curva natural a la siguiente secuencia de desplazamiento	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
ENDFOR	Línea final del bucle contador FOR	*PGA* Bucle contador (FOR ... TO ..., ENDFOR) (Página 104)		●	●	●	●
ENDIF	Línea final de la bifurcación IF	*PGA* Bucle de programa con alternativa (IF, ELSE, ENDIF) (Página 101)		●	●	●	●
ENDLABEL	Marca final para repeticiones del programa de pieza mediante REPEAT	*PGA, FB1(K1)* Repetición de secciones de programa (REPEAT, REPEATB, ENDLABEL, P) (Página 92)		●	●	●	●
ENDLOOP	Línea final del bucle de programa sin fin LOOP	*PGA* Bucle de programa sin fin (LOOP, ENDLOOP) (Página 103)		●	●	●	●
ENDPROC	Línea final de un programa con la línea inicial PROC			●	●	●	●
ENDWHILE	Línea final del bucle WHILE	*PGA* Bucle de programa con la condición en el inicio del bucle (WHILE, ENDWHILE) (Página 106)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
ETAN	Transición de curva tangencial a la próxima secuencia de desplazamiento para inicio spline	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	m	-	○	-	○
EVERY	Ejecutar una acción síncrona al pasar la condición de FALSE a TRUE	*PGA* Comprobación cíclica de la condición (WHEN, WHENEVER, FROM, EVERY) (Página 585)		●	●	●	●
EX	Palabra reservada para la asignación de valores en notación exponencial	*PGA* Variables de usuario predefinidas: Parámetros de cálculo R (Página 19)		●	●	●	●
EXECSTRING	Transferencia de una variable string con la línea del programa de pieza que debe ejecutarse	*PGA* Programación indirecta de líneas de programa de pieza (EXECSTRING) (Página 62)		●	●	●	●
EXECTAB	Ejecutar un elemento de una tabla de movimientos	*PGA* Programación indirecta de líneas de programa de pieza (EXECSTRING) (Página 62)		●	●	●	●
EXECUTE	Activación de la ejecución del programa	*PGA* Desactivación de la preparación del contorno (EXECUTE) (Página 752)		●	●	●	●
EXP	Función exponencial ex	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
EXTCALL	Ejecutar subprograma externo	*PGA* Ejecutar subprograma externo (EXTCALL) (Página 200)		●	●	●	●
EXTERN	Declaración de un subprograma con transferencia de parámetros	*PGA* Llamada de subprograma sin transferencia de parámetros (Página 185)		●	●	●	●
F	Valor de avance (en combinación con G4, el tiempo de espera se programa también con F)	*PG*		●	●	●	●
FA	Avance por eje	*PG*	m	●	●	●	●
FAD	Avance de penetración para aproximación y retirada suaves	*PG*		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
FALSE	Constante lógica: falso	*PGA* Definición de variables de usuario (DEF) (Página 24)		•	•	•	•
FB	Avance por secuencia	*PG*		•	•	•	•
FCTDEF	Definición de la función polinomio	*PGA* Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 427)		-	-	-	-
FCUB	Avance variable según spline cúbico	*PGA* Variación del avance (FNORM, FLIN, FCUB, FPO) (Página 484)	m	•	•	•	•
FD	Avance de contorneado para corrección por volante	*PG*	s	•	•	•	•
FDA	Avance axial para corrección por volante	*PG*	s	•	•	•	•
FENDNORM	Deceleración en los dos vértices DES	*PGA* Reducción del avance con deceleración en los dos vértices (FENDNORM, G62, G621) (Página 282)	m	•	•	•	•
FFWOF [4]	Mando anticipativo DES	*PG*	m	•	•	•	•
FFWON	Mando anticipativo CON	*PG*	m	•	•	•	•
FGREF	Radio de referencia en ejes giratorios o factores de referencia de trayectoria en ejes de orientación (interpolación vectorial)	*PG*	m	•	•	•	•
FGROUP	Determinación de los ejes con avance de contorneado	*PG*		•	•	•	•
FI	Parámetros para el acceso a datos frame: Decalaje fino	*PGA* Leer y modificar componentes de frame (TR, FI, RT, SC, MI) (Página 300)		•	•	•	•
FIFOCTRL	Control de la memoria (búfer) de pretratamiento	*PGA* Ejecución del programa con memoria de pretratamiento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) (Página 490)	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
FILEDATE	Indica la fecha del último acceso de escritura al fichero	*PGA* Leer información de fichero (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 146)		●	●	●	●
FILEINFO	Indica la suma de FILEDATE, FILESIZE, FILESTAT y FILETIME en total	*PGA* Leer información de fichero (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 146)		●	●	●	●
FILESIZE	Indica el tamaño actual del fichero	*PGA* Leer información de fichero (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 146)		●	●	●	●
FILESTAT	Indica el estado del fichero por lo que respecta a derechos de lectura, escritura, ejecución, visualización y borrado (rwxsd)	*PGA* Leer información de fichero (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 146)		●	●	●	●
FILETIME	Indica la hora del último acceso de escritura al fichero	*PGA* Leer información de fichero (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 146)		●	●	●	●
FINEA	Fin de movimiento al alcanzar la "Parada precisa fina"	*PGA* Criterio de fin del movimiento programable (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 283)	m	●	●	●	●
FL	Velocidad límite para ejes síncronos	*PG* Leer y modificar componentes de frame (TR, FI, RT, SC, MI) (Página 300)	m	●	●	●	●
FLIN	Avance lineal variable	*PGA* Variación del avance (FNORM, FLIN, FCUB, FPO) (Página 484)	m	●	●	●	●
FMA	Varios avances axiales	*PG*	m	-	-	-	-
FNORM [4]	Avance normal según DIN66025	*PGA* Variación del avance (FNORM, FLIN, FCUB, FPO) (Página 484)	m	●	●	●	●
FOCOF	Desactivar desplazamiento con par/fuerza limitado	*PGA* Desplazamiento a tope fijo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 647)	m	○	-	○	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
FOCON	Activar desplazamiento con par/fuerza limitado	*PGA* Desplazamiento a tope fijo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 647)	m	○	-	○	-
FOR	Bucle contador con número fijo de pasadas	*PGA* Bucle contador (FOR ... TO ..., ENDFOR) (Página 104)		●	●	●	●
FP	Punto fijo: Número del punto fijo que se debe alcanzar	*PG*	s	●	●	●	●
FPO	Variación del avance programada mediante un polinomio	*PGA* Variación del avance (FNORM, FLIN, FCUB, FPO) (Página 484)		-	-	-	-
FPR	Identificación eje giratorio	*PG*		●	●	●	●
FPRAOF	Desactivar avance por vuelta	*PG*		●	●	●	●
FPRAON	Activar avance por vuelta	*PG*		●	●	●	●
FRAME	Tipo de datos para la determinación de sistemas de coordenadas	*PGA* Definición de frames nuevos (DEF FRAME) (Página 303)		●	●	●	●
FRC	Avance para el radio y chaflán	*PG*	s	●	●	●	●
FRCM	Avance modal para radio y chaflán	*PG*	m	●	●	●	●
FROM	La acción se ejecuta si la condición se cumple una vez y mientras la acción síncrona esté activa	*PGA* Comprobación cíclica de la condición (WHEN, WHENEVER, FROM, EVERY) (Página 585)		●	●	●	●
FTOC	Modificación de la corrección de herramienta fina	*PG* Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 427)		●	●	●	●
FTOCOF [4]	Corrección de herramienta fina activa online DES	*PGA* Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 427)	m	●	●	●	●
FTOCON	Corrección de herramienta fina activa online CON	*PGA* Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 427)	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
FXS	Desplazamiento a tope fijo	*PG* Desplazamiento a tope fijo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 647)	m	●	●	●	●
FXST	Límite de par para desplazamiento a tope fijo	*PG* Desplazamiento a tope fijo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 647)	m	●	●	●	●
FXSW	Ventana de vigilancia para desplazamiento a tope fijo	*PG* Desplazamiento a tope fijo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 647)		●	●	●	●
FZ	Avance por diente	*PG*	m	●	●	●	●
G0	Interpolación lineal con velocidad de desplazamiento rápido (movimiento en desplazamiento rápido)	*PG*	m	●	●	●	●
G1 [4]	Interpolación lineal con avance (interpolación lineal)	*PG*	m	●	●	●	●
G2	Interpolación circular en sentido horario	*PG*	m	●	●	●	●
G3	Interpolación circular en sentido antihorario	*PG*	m	●	●	●	●
G4	Tiempo de espera, temporizado	*PG*	s	●	●	●	●
G5	Rectificado oblicuo de ranuras	*PGA* Eje inclinado (TRAANG) (Página 389)	s	●	●	●	●
G7	Movimiento de compensación en el rectificado oblicuo de ranuras	*PGA* Eje inclinado (TRAANG) (Página 389)	s	●	●	●	●
G9	Parada precisa reducción de velocidad	*PG*	s	●	●	●	●
G17 [4]	Selección del plano de trabajo X/Y	*PG*	m	●	●	●	●
G18	Selección del plano de trabajo Z/X	*PG*	m	●	●	●	●
G19	Selección del plano de trabajo Y/Z	*PG*	m	●	●	●	●
G25	Limitación inferior del campo de trabajo	*PG*	s	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
G26	Limitación superior del campo de trabajo	*PG*	s	•	•	•	•
G33	Tallado de roscas con paso constante	*PG*	m	•	•	•	•
G34	Tallado de roscas con paso linealmente creciente	*PG*	m	•	•	•	•
G35	Tallado de roscas con paso linealmente decreciente	*PG*	m	•	•	•	•
G40 [4]	Corrección radio herramienta DES	*PG*	m	•	•	•	•
G41	Corrección del radio de la herramienta a la izquierda del contorno	*PG*	m	•	•	•	•
G42	Corrección del radio de la herramienta a la derecha del contorno	*PG*	m	•	•	•	•
G53	Supresión del decalaje de origen actual (por secuencia)	*PG*	s	•	•	•	•
G54	1er decalaje de origen ajustable	*PG*	m	•	•	•	•
G55	2º decalaje de origen ajustable	*PG*	m	•	•	•	•
G56	3er decalaje de origen ajustable	*PG*	m	•	•	•	•
G57	4º decalaje de origen ajustable	*PG*	m	•	•	•	•
G58	Decalaje de origen absoluto axial programable, decalaje basto	*PG*	s	•	•	•	•
G59	Decalaje de origen aditivo axial programable, decalaje fino	*PG*	s	•	•	•	•
G60 [4]	Parada precisa reducción de velocidad	*PG*	m	•	•	•	•
G62	Deceleración en los dos vértices en esquinas interiores con corrección del radio de herramienta activa (G41, G42)	*PGA* Reducción del avance con deceleración en los dos vértices (FENDNORM, G62, G621) (Página 282)	m	•	•	•	•
G63	Roscado de taladros con mandril de compensación	*PG*	s	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
G64	Modo de contorneado	*PG*	m	●	●	●	●
G70	Acotado en pulgadas para datos geométricos (longitudes)	*PG*	m	●	●	●	●
G71 [4]	Acotado métrico para datos geométricos (longitudes)	*PG*	m	●	●	●	●
G74	Búsqueda de punto de referencia	*PG*	s	●	●	●	●
G75	Desplazamiento a punto fijo	*PG*	s	●	●	●	●
G90 [4]	Acotado absoluto	*PG*	m/s	●	●	●	●
G91	Acotado incremental	*PG*	m/s	●	●	●	●
G93	Avance inverso al tiempo en rpm	*PG*	m	●	●	●	●
G94 [4]	Avance lineal F en mm/min o pulgadas/min y grados/min	*PG*	m	●	●	●	●
G95	Avance por vuelta F en mm/vuelta o pulgadas/ vuelta	*PG*	m	●	●	●	●
G96	Velocidad de corte constante (como con G95) CON	*PG*	m	●	●	●	●
G97	Velocidad de corte constante (como con G95) DES	*PG*	m	●	●	●	●
G110	Programación de polo relativa a la última posición nominal programada	*PG*	s	●	●	●	●
G111	Programación de polo relativa al origen del sistema actual de coordenadas de pieza	*PG*	s	●	●	●	●
G112	Programación de polo relativa al último polo válido	*PG*	s	●	●	●	●
G140 [4]	Dirección para aprox./ retirada suaves del contorno definida mediante G41/G42	*PG*	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
G141	Dirección de aprox./ retirada suaves a la izquierda del contorno	*PG*	m	•	•	•	•
G142	Dirección de aprox./ retirada suaves a la derecha del contorno	*PG*	m	•	•	•	•
G143	Dirección de aprox./ retirada suaves del contorno dependiente de la tangente	*PG*	m	•	•	•	•
G147	Aproximación suave siguiendo una recta	*PG*	s	•	•	•	•
G148	Retirada suave siguiendo una recta	*PG*	s	•	•	•	•
G153	Supresión de frames actuales, incluido el frame básico	*PG*	s	•	•	•	•
G247	Aproximación suave siguiendo un cuadrante de circunferencia	*PG*	s	•	•	•	•
G248	Retirada suave siguiendo un cuadrante de circunferencia	*PG*	s	•	•	•	•
G290	Conmutar al modo SINUMERIK CON	*FBW*	m	•	•	•	•
G291	Conmutar al modo ISO2/3 CON	*FBW*	m	•	•	•	•
G331	Roscado de taladros sin mandril de compensación, paso positivo, sentido de giro a derechas	*PG*	m	•	•	•	•
G332	Roscado de taladros sin mandril de compensación, paso negativo, sentido de giro a izquierdas	*PG*	m	•	•	•	•
G340 [4]	Secuencia de desplazamiento en el espacio (simultáneamente en profundidad y en el plano (hélice))	*PG*	m	•	•	•	•
G341	Primero penetrar en el eje perpendicular (z); después desplazamiento en el plano	*PG*	m	•	•	•	•
G347	Aproximación suave siguiendo una semicircunferencia	*PG*	s	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
G348	Retirada suave siguiendo una semicircunferencia	*PG*	s	●	●	●	●
G450 [4]	Circunferencia de transición	*PG*	m	●	●	●	●
G451	Intersección de equidistantes	*PG*	m	●	●	●	●
G460 [4]	Activación de la vigilancia de colisión para la secuencia de aproximación y retirada	*PG*	m	●	●	●	●
G461	Inserción de un círculo en la secuencia de corrección del radio de herramienta	*PG*	m	●	●	●	●
G462	Inserción de una recta en la secuencia de corrección del radio de herramienta	*PG*	m	●	●	●	●
G500 [4]	Desactivación de todos los frames ajustables; frames básicos activos	*PG*	m	●	●	●	●
G505 ... G599	5 ... 99º decalaje de origen ajustable	*PG*	m	●	●	●	●
G601 [4]	Cambio de secuencia con parada precisa fina	*PG*	m	●	●	●	●
G602	Cambio de secuencia con parada precisa basta	*PG*	m	●	●	●	●
G603	Cambio de secuencia para interpolador - fin de secuencia	*PG*	m	●	●	●	●
G621	Deceleración en los dos vértices en todas las esquinas	*PGA* Reducción del avance con deceleración en los dos vértices (FENDNORM, G62, G621) (Página 282)	m	●	●	●	●
G641	Modo de contorneado con matado de esquinas según criterio de trayectoria (= distancia de matado programable)	*PG*	m	●	●	●	●
G642	Modo de contorneado con matado de esquinas respetando las tolerancias definidas	*PG*	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
G643	Modo de contorneado con matado de esquinas respetando las tolerancias definidas (internas de la secuencia)	*PG*	m	•	•	•	•
G644	Modo de contorneado con matado de esquinas con la máxima dinámica posible	*PG*	m	•	•	•	•
G645	Modo de contorneado con matado de esquinas y transiciones de secuencia tangenciales respetando las tolerancias definidas	*PG*	m	•	•	•	•
G700	Acotado en pulgadas para datos geométricos y tecnológicos (longitudes, avance)	*PG*	m	•	•	•	•
G710 [4]	Acotado métrico para datos geométricos y tecnológicos (longitudes, avance)	*PG*	m	•	•	•	•
G751	Aproximación a un punto fijo a través de un punto intermedio	PG	s	•	•	•	•
G810 [4], ..., G819	Grupo G reservado para el usuario OEM	*PGA* Funciones especiales para el usuario de OEM (OEMIPO1, OEMIPO2, G810 bis G829) (Página 281)		•	•	•	•
G820 [4], ..., G829	Grupo G reservado para el usuario OEM	*PGA* Funciones especiales para el usuario de OEM (OEMIPO1, OEMIPO2, G810 bis G829) (Página 281)		•	•	•	•
G931	Especificación del avance mediante tiempo de desplazamiento		m	•	•	•	•
G942	Avance lineal y velocidad de corte constante o congelar velocidad de giro del cabezal		m	•	•	•	•
G952	Avance por vuelta y velocidad de corte constante o congelar velocidad de giro del cabezal		m	•	•	•	•
G961	Velocidad de corte constante y avance lineal	*PG*	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
G962	Avance lineal o avance por vuelta y velocidad de corte constante	*PG*	m	●	●	●	●
G971	Congelar velocidad de giro del cabezal y avance lineal	*PG*	m	●	●	●	●
G972	Avance lineal o avance por vuelta y congelar velocidad de giro del cabezal constante	*PG*	m	●	●	●	●
G973	Avance por vuelta sin limitación de revoluciones del cabezal	*PG*	m	●	●	●	●
GEOAX	Asignar a los ejes geométricos 1 - 3 nuevos ejes del canal	*PGA* Ejes geométricos conmutables (GEOAX) (Página 704)		●	●	●	●
GET	Cambiar el eje habilitado entre canales	*PGA* Intercambio de ejes, intercambio de cabezales (RELEASE, GET, GETD) (Página 128)		●	●	●	●
GETACTT	Define la herramienta activa a partir de un grupo de herramientas con el mismo nombre	*FBW*		●	●	●	●
GETACTTD	Define el número T correspondiente en relación con un número D absoluto	*PGA* Asignación libre de números D: Determinar el número T para el número D predeterminado (GETACTTD) (Página 457)		●	●	●	●
GETD	Cambiar el eje directamente entre canales	*PGA* Intercambio de ejes, intercambio de cabezales (RELEASE, GET, GETD) (Página 128)		●	●	●	●
GETDNO	Indica el número D de un filo (CE) de una herramienta (T)	*PGA* Asignación libre de números D: Cambiar nombre de números D (GETDNO, SETDNO) (Página 456)		●	●	●	●
GETEXET	Lectura del número T cambiado	*FBW*		●	●	●	●
GETFREELOC	Buscar un puesto vacío en los almacenes para una determinada herramienta	*FBW*		●	●	●	●
GETSELT	Suministrar número T seleccionado	*FBW*		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
GETT	Determinar número T de un nombre de herramienta	*FBW*		•	•	•	•
GETTCOR	Leer longitudes de herramienta o componentes longitudinales de herramienta	*FB1(W1)*		•	•	•	•
GETTENV	Leer números T, D y DL	*FB1(W1)*		•	•	•	•
GOTO	Instrucción de salto primero hacia delante y después hacia atrás (dirección primero hacia el fin del programa y después hacia el inicio del programa)	*PGA* Saltos de programa a marcas de salto (GOTOB, GOTOF, GOTO, GOTOC) (Página 86)		•	•	•	•
GOTOB	Instrucción de salto hacia atrás (en dirección al inicio del programa)	*PGA* Saltos de programa a marcas de salto (GOTOB, GOTOF, GOTO, GOTOC) (Página 86)		•	•	•	•
GOTOC	Como GOTO, pero suprimiendo la alarma 14080 "Destino del salto no encontrado"	*PGA* Saltos de programa a marcas de salto (GOTOB, GOTOF, GOTO, GOTOC) (Página 86)		•	•	•	•
GOTOF	Instrucción de salto hacia delante (en dirección al final del programa)	*PGA* Saltos de programa a marcas de salto (GOTOB, GOTOF, GOTO, GOTOC) (Página 86)		•	•	•	•
GOTOS	Salto hacia atrás al inicio del programa	*PGA* Salto atrás hasta el inicio del programa (GOTOS) (Página 85)		•	•	•	•
GP	Palabra reservada para la programación indirecta de atributos de posición	*PGA* Programación indirecta de atributos de posición (GP) (Página 59)		•	•	•	•
GWPSOF	Desactivar la velocidad periférica de muela constante (SUG)	*PG*	s	•	•	•	•
GWPSON	Activar la velocidad periférica de muela constante (SUG)	*PG*	s	•	•	•	•
H...	Emisión de funciones auxiliares al PLC	*PG/FB1(H2)*		•	•	•	•
HOLES1	Ciclo de figuras de taladros, taladros en línea	*BHD/BHF*		•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
HOLES2	Ciclo de figuras de taladros, taladros en círculo	*BHD/BHF*		●	●	●	●
I	Parámetro de interpolación	*PG*	s	●	●	●	●
I1	Coordenada del punto intermedio	*PG*	s	●	●	●	●
IC	Acotado incremental	*PG*	s	●	●	●	●
ICYCOF	Ejecución de todas las secuencias de un ciclo tecnológico según ICYCOF en un ciclo IPO	*PGA* Control de la ejecución de ciclos tecnológicos (ICYCOF, ICYCON) (Página 658)		●	●	●	●
ICYCON	Ejecutar cada secuencia de un ciclo tecnológico según ICYCON en un ciclo IPO independiente	*PGA* Control de la ejecución de ciclos tecnológicos (ICYCOF, ICYCON) (Página 658)		●	●	●	●
ID	Identifica acciones síncronas modales	*PGA* Ámbito de validez y secuencia de ejecución (ID, IDS) (Página 583)	m	●	●	●	●
IDS	Identifica acciones síncronas estáticas modales	*PGA* Ámbito de validez y secuencia de ejecución (ID, IDS) (Página 583)		●	●	●	●
IF	Introducción de un salto condicionado en el programa de pieza/ciclo tecnológico	*PGA* Bucle de programa con alternativa (IF, ELSE, ENDIF) (Página 101)		●	●	●	●
INDEX	Definir un índice de un carácter en la cadena de caracteres introducida	*PGA* Búsqueda de caracteres/cadena en una cadena (INDEX, RINDEX, MINDEX, MATCH) (Página 80)		●	●	●	●
INIPO	Inicialización de las variables con PowerOn	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
INIRE	Inicialización de las variables con Reset	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
INICF	Inicialización de las variables con NewConfig	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
INIT	Selección de un determinado programa CN para la ejecución en un determinado canal	*PGA* Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 109)		-	-	-	-
INITIAL	Creación de un fichero INI en todas las áreas	*PGA* Memoria de trabajo (CHANDATA, COMPLETE, INITIAL) (Página 216)		●	●	●	●
INT	Tipo de dato: Valor entero con signo	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
INTERSEC	Calcular el punto de intersección entre dos elementos de contorno	*PGA* Determinación del punto de intersección entre dos elementos de contorno (INTERSEC) (Página 747)		●	●	●	●
INVCCW	Desplazar evoluta en sentido antihorario	*PG*	m	-	-	-	-
INVCW	Desplazar evoluta en sentido horario	*PG*	m	-	-	-	-
INVFRAME	Calcular el frame inverso a partir de un frame	*FB1(K2)*		●	●	●	●
IP	Parámetro de interpolación variable	*PGA* Programación indirecta (Página 55)		●	●	●	●
IPOBRKA	Criterio de desplazamiento desde el punto de aplicación de la rampa de frenado	*PGA* Criterio de fin del movimiento programable (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 283)	m	●	●	●	●
IPOENDA	Fin de movimiento al alcanzar la "Parada IPO"	*PGA* Criterio de fin del movimiento programable (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 283)	m	●	●	●	●
IPTRLOCK	Congelar inicio de la sección de programa sin posibilidad de búsqueda en la siguiente secuencia de función de la máquina.	*PGA* Evitar punto de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK) (Página 499)	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
IPTRUNLOCK	Ajustar fin de la sección de programa sin posibilidad de búsqueda a la secuencia actual en el momento de la interrupción	*PGA* Evitar punto de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK) (Página 499)	m	●	●	●	●
ISAXIS	Comprobar si el eje geométrico indicado como parámetro es 1	*PGA* Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 701)		●	●	●	●
ISD	Profundidad de penetración	*PGA* Activar correcciones de herramientas 3D (CUT3DC, CUT3DF, CUT3DFS, CUT3DFF, ISD) (Página 433)	m	●	●	●	●
ISFILE	Comprobar si existe un fichero en la memoria de aplicación NCK	*PGA* Comprobar la presencia de un fichero (ISFILE) (Página 144)		●	●	●	●
ISNUMBER	Comprobar si es posible convertir la cadena de caracteres introducida en un valor numérico	*PGA* Conversión de tipos de STRING (NUMBER, ISNUMBER, AXNAME) (Página 75)		●	●	●	●
ISOCALL	Llamada indirecta de un programa programado en lenguaje ISO	*PGA* Llamada indirecta a un programa creado en lenguaje ISO (ISOCALL) (Página 196)		●	●	●	●
ISVAR	Comprobar si el parámetro de transferencia contiene una variable conocida al CN	*PGA* Leer llamada de función ISVAR y datos de máquina con índice Array (Página 720)		●	●	●	●
J	Parámetro de interpolación	*PG*	s	●	●	●	●
J1	Coordenada del punto intermedio	*PG*	s	●	●	●	●
JERKA	Activar características de aceleración ajustadas en los datos de máquina para los ejes programados			●	●	●	●
JERKLIM	Reducción o aumento del jerk axial máximo	*PGA* Corrección porcentual de sacudidas (JERKLIM) (Página 512)	m	●	●	●	●
JERKLIMA	Reducción o aumento del jerk axial máximo	*PG*	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
K	Parámetro de interpolación	*PG*	s	●	●	●	●
K1	Coordenada del punto intermedio	*PG*	s	●	●	●	●
KONT	Rodear el contorno en la corrección de herramienta	*PG*	m	●	●	●	●
KONTC	Aproximación/retirada con polinomio de curvatura continua	*PG*	m	●	●	●	●
KONTT	Aproximación/retirada con polinomio de tangente continua	*PG*	m	●	●	●	●
L	Número de subprograma	*PGA* Llamada de subprograma sin transferencia de parámetros (Página 185)	s	●	●	●	●
LEAD	Ángulo de avance 1. Orientación de la herramienta 2. Polinomio de orientación	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	m	● -	● -	● -	● -
LEADOF	Acoplamiento de valores maestros DES	*PGA* Acoplamiento entre ejes de valores maestros (LEADON, LEADOF) (Página 548)		-	-	-	-
LEADON	Acoplamiento de valores maestros CON	*PGA* Acoplamiento entre ejes de valores maestros (LEADON, LEADOF) (Página 548)		-	-	-	-
LENTOAX	Proporciona información acerca de la asignación de las longitudes L1, L2 y L3 de la herramienta activa a los ejes de abscisas, ordenadas y aplicadas	*FB1(W1)*		●	●	●	●
LFOF [4]	Retirada rápida para el tallado de roscas DES	*PG*	m	●	●	●	●
LFON	Retirada rápida para el tallado de roscas CON	*PG*	m	●	●	●	●
LFPOS	Retirada del eje declarado con POLFMASK o POLFMLIN a la posición de eje absoluta programada con POLF	*PG*	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
LFTXT	El plano del movimiento de retroceso en la retirada rápida se determina a partir de la tangente de trayectoria y de la orientación de herramienta actual	*PG*	m	●	●	●	●
LFWP	El plano del movimiento de retroceso en la retirada rápida se determina mediante el plano de trabajo actual (G17/G18/G19)	*PG*	m	●	●	●	●
LIFTFAST	Retirada rápida	*PG* Retirada rápida del contorno (SETINT LIFTFAST, ALF) (Página 121)		●	●	●	●
LIMS	Limitación de la velocidad de giro con G96/G961 y G97	*PG*	m	●	●	●	●
LLI	Límite inferior de las variables	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
LN	Logaritmo neperiano (logaritmo natural)	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
LOCK	Bloquear acción síncrona con ID (parar ciclo tecnológico)	*PGA* Bloquear, habilitar, resetear (LOCK, UNLOCK, RESET) (Página 662)		●	●	●	●
LONGHOLE	Ciclo de figuras de fresado, agujeros rasgados en círculo	*BHD/BHF*		-	-	-	-
LOOP	Introducción de un bucle sin fin	*PGA* Bucle de programa sin fin (LOOP, ENDLOOP) (Página 103)		●	●	●	●
M0	Parada programada	*PG*		●	●	●	●
M1	Parada opcional	*PG*		●	●	●	●
M2	Fin de programa principal con reseteo al principio del programa	*PG*		●	●	●	●
M3	Giro a la derecha para cabezal maestro	*PG*		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
M4	Giro a la izquierda para cabezal maestro	*PG*		•	•	•	•
M5	Parada para cabezal maestro	*PG*		•	•	•	•
M6	Cambio de herramienta	*PG*		•	•	•	•
M17	Fin de subprograma	*PG*		•	•	•	•
M19	Programaciones de cabezal recopiladas con SSL	*PG*		•	•	•	•
M30	Fin de programa, mismo efecto que M2	*PG*		•	•	•	•
M40	Cambio automático de gama o escalón de reducción	*PG*		•	•	•	•
M41... M45	Nivel de reducción 1, ..., 5	*PG*		•	•	•	•
M70	Transición a modo Eje	*PG*		•	•	•	•
MASLDEF	Definir conjunto de ejes maestro/esclavo	*PGA* Conjunto maestro-esclavo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 576)		•	•	•	•
MASLDEL	Separar conjunto de ejes maestro/esclavo y borrar definición del conjunto	*PGA* Conjunto maestro-esclavo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 576)		•	•	•	•
MASLOF	Desconexión de un acoplamiento temporal	*PGA* Conjunto maestro-esclavo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 576)		•	•	•	•
MASLOFS	Desconexión de un acoplamiento temporal con parada automática del eje esclavo	*PGA* Conjunto maestro-esclavo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 576)		•	•	•	•
MASLON	Conexión de un acoplamiento temporal	*PGA* Conjunto maestro-esclavo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 576)		•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
MATCH	Búsqueda de una cadena de caracteres en el string	*PGA* Búsqueda de caracteres/cadena en una cadena (INDEX, RINDEX, MINDEX, MATCH) (Página 80)		●	●	●	●
MAXVAL	Valor más grande de dos variables (función aritm.)	*PGA* Mínimo, máximo y rango de variables (MINVAL, MAXVAL, BOUND) (Página 70)		●	●	●	●
MCALL	Llamada de subprograma modal	*PGA* Llamada de subprograma modal (MCALL) (Página 191)		●	●	●	●
MEAC	Medición continua sin borrado de trayecto residual	*PGA* Función de medida ampliada (MEASA, MEAWA, MEAC) (opcional) (Página 270)	s	-	-	-	-
MEAFRAME	Cálculo de frame a partir de puntos de medida	*PGA* Cálculo de frame a partir de 3 puntos de medida en el espacio (MEAFRAME) (Página 309)		●	●	●	●
MEAS	Medida con palpador de contacto	*PGA* Medida con palpador de contacto (MEAS, MEAW) (Página 267)	s	●	●	●	●
MEASA	Medición con borrado de trayecto residual	*PGA* Función de medida ampliada (MEASA, MEAWA, MEAC) (opcional) (Página 270)	s	-	-	-	-
MEASURE	Método de cálculo para la medición de pieza y de herramienta	*FB2(M5)* Medida con palpador de contacto (MEAS, MEAW) (Página 267)		●	●	●	●
MEAW	Medición con palpador de contacto sin borrado de trayecto residual	*PGA* Medida con palpador de contacto (MEAS, MEAW) (Página 267)	s	●	●	●	●
MEAWA	Medición sin borrado de trayecto residual	*PGA* Función de medida ampliada (MEASA, MEAWA, MEAC) (opcional) (Página 270)	s	-	-	-	-
MI	Acceso a datos de frame: Simetría especular	*PGA* Leer y modificar componentes de frame (TR, FI, RT, SC, MI) (Página 300)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
MINDEX	Definir un índice de un carácter en la cadena de caracteres introducida	*PGA* Búsqueda de caracteres/cadena en una cadena (INDEX, RINDEX, MINDEX, MATCH) (Página 80)		•	•	•	•
MINVAL	Valor más pequeño de dos variables (función aritm.)	*PGA* Mínimo, máximo y rango de variables (MINVAL, MAXVAL, BOUND) (Página 70)		•	•	•	•
MIRROR	Simetría programable	*PGA*	s	•	•	•	•
MMC	Llamar la ventana de diálogo de forma interactiva desde el programa de pieza en el HMI	*PGA* Llamar interactivamente la ventana del programa de pieza (MMC) (Página 724)		•	•	•	•
MOD	División de módulo	*PGA* Funciones de cálculo (Página 63)		•	•	•	•
MODAXVAL	Calcular la posición de módulo de un eje giratorio con módulo	*PGA* Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 701)		•	•	•	•
MOV	Iniciar eje de posicionado	*PGA* Arrancar/parar eje (MOV) (Página 630)		•	•	•	•
MSG	Avisos programables	*PG*	m	•	•	•	•
MVTOOL	Instrucción de programación para el desplazamiento de una herramienta	*FBW*		•	•	•	•
N	Número de secuencia auxiliar CN	*PG*		•	•	•	•
NCK	Especificación del ámbito de vigencia de datos	*PGA* Definición de variables de usuario (DEF) (Página 24)		•	•	•	•
NEWCONF	Aceptar los datos de máquina modificados (equivale a "activar los datos de máquina")	*PGA* Activar los datos de máquina (NEWCONF) (Página 135)		•	•	•	•
NEWT	Crear nueva herramienta	*PGA* Leer y modificar componentes de frame (TR, FI, RT, SC, MI) (Página 300)		•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
NORM [4]	Ajuste normal en el punto inicial; final durante la corrección de herramienta	*PG*	m	●	●	●	●
NOT	NO lógica (negación)	*PGA* Operaciones de comparación y operaciones lógicas (Página 66)		●	●	●	●
NPROT	Zona protegida específica de máquina ON/OFF	*PGA* Activar/desactivar zonas protegidas (CPROT, NPROT) (Página 225)		●	●	●	●
NPROTDEF	Definición de una zona protegida específica de la máquina	*PGA* Definición de las zonas protegidas (CPROTDEF, NPROTDEF) (Página 221)		●	●	●	●
NUMBER	Conversión de la cadena de caracteres introducida en un número	*PGA* Conversión de tipos de STRING (NUMBER, ISNUMBER, AXNAME) (Página 75)		●	●	●	●
OEMIPO1	Interpolación OEM 1	*PGA* Funciones especiales para el usuario de OEM (OEMIPO1, OEMIPO2, G810 bis G829) (Página 281)	m	●	●	●	●
OEMIPO2	Interpolación OEM 2	*PGA* Funciones especiales para el usuario de OEM (OEMIPO1, OEMIPO2, G810 bis G829) (Página 281)	m	●	●	●	●
OF	Palabra reservada de la bifurcación CASE	*PGA* Bifurcación de programa (CASE ... OF ... DEFAULT ...) (Página 89)		●	●	●	●
OFFN	Demasía/creces para el contorno programado	*PG*	m	●	●	●	●
OMA1	Dirección OEM 1		m	●	●	●	●
OMA2	Dirección OEM 2		m	●	●	●	●
OMA3	Dirección OEM 3		m	●	●	●	●
OMA4	Dirección OEM 4		m	●	●	●	●
OMA5	Dirección OEM 5		m	●	●	●	●
OR	Operador lógico, combinación O	*PGA* Operaciones de comparación y operaciones lógicas (Página 66)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
ORIAXES	Interpolación lineal de los ejes de máquina o ejes de orientación	*PGA* Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 349)	m	•	•	•	•
ORIAXPOS	Ángulo de orientación a través de ejes de orientación virtuales con posiciones de eje giratorio		m	•	•	•	•
ORIC [4]	Los cambios de orientación en esquinas exteriores se superponen a la secuencia circular a insertar	*PGA* Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 447)	m	•	•	•	•
ORICONCCW	Interpolación en una superficie envolvente de círculo en sentido antihorario	*PGA/FB3(F3)* Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 352)	m	•	•	•	•
ORICONCW	Interpolación en una superficie envolvente de círculo en sentido horario	*PGA/FB3(F4)* Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 352)	m	•	•	•	•
ORICONIO	Interpolación en una superficie envolvente de círculo con indicación de una orientación intermedia	*PGA/FB3(F4)* Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 352)	m	•	•	•	•
ORICONTO	Interpolación en una superficie envolvente de círculo en la transición tangencial (indicación de la orientación final)	*PGA/FB3(F5)* Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 352)	m	•	•	•	•
ORICURVE	Interpolación de la orientación con especificación del movimiento de dos puntos de contacto de la herramienta	*PGA/FB3(F6)* Especificación de orientación de dos puntos de contacto (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=) (Página 356)	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
ORID	Los cambios de orientación se ejecutan antes de la secuencia circular	*PGA* Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 447)	m	•	•	•	•
ORIEULER	Ángulo de orientación mediante ángulos eulerianos	*PGA* Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 349)	m	•	•	•	•
ORIMKS	Orientación de herramienta en el sistema de coordenadas de máquina	*PGA* Referencia de los ejes de orientación (ORIWKS, ORIMKS) (Página 347)	m	•	•	•	•
ORIPATH	Orientación de la herramienta con relación a la trayectoria	*PGA* Giro relativo a la trayectoria de la orientación de herramienta (ORIPATH, ORIPATHS, ángulo de giro) (Página 365)	m	•	•	•	•
ORIPATHS	Orientación de la herramienta con relación a la trayectoria; se suaviza un acodamiento en el desarrollo de la orientación	*PGA* Giro relativo a la trayectoria de la orientación de herramienta (ORIPATH, ORIPATHS, ángulo de giro) (Página 365)	m	•	•	•	•
ORIPLANE	Interpolación en un plano (corresponde a ORIVECT) interpolación circular de gran radio	*PGA* Programación de la orientación a lo largo de una superficie de cono (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 352)	m	•	•	•	•
ORIRESET	Estado inicial de la orientación de herramienta con hasta 3 ejes de orientación	*PGA* Variantes de la programación de la orientación y el estado inicial (ORIRESET) (Página 334)		•	•	•	•
ORIROTA	Ángulo de rotación frente a un sentido de giro absoluto especificado	*PGA* Giros de la orientación de la herramienta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 360)	m	•	•	•	•
ORIROTC	Vector de giro tangencial a la trayectoria tangente	*PGA* Giros de la orientación de la herramienta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 360)	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
ORIROTR	Ángulo de rotación relativo al plano entre la orientación inicial y final	*PGA* Giros de la orientación de la herramienta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 360)	m	•	•	•	•
ORIROTT	Ángulo de rotación relativo a la modificación del vector de orientación	*PGA* Giros de la orientación de la herramienta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 360)	m	•	•	•	•
ORIRPY	Ángulo de orientación mediante ángulos RPY (XYZ)	*PGA* Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 349)	m	•	•	•	•
ORIRPY2	Ángulo de orientación mediante ángulos RPY (ZYX)	*PGA* Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 349)	m	•	•	•	•
ORIS	Cambio de orientación	*PGA* Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 447)	m	•	•	•	•
ORISOF [4]	Alisado del desarrollo de la orientación DES	*PGA* Alisado del desarrollo de la orientación (ORISON, ORISOF) (Página 374)	m	•	•	•	•
ORISON	Alisado del desarrollo de la orientación CON	*PGA* Alisado del desarrollo de la orientación (ORISON, ORISOF) (Página 374)	m	•	•	•	•
ORIVECT	Interpolación circular de gran radio (idéntico a ORIPLANE)	*PGA* Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 349)	m	•	•	•	•
ORIVIRT1	Ángulo de orientación mediante ejes de orientación virtuales (definición 1)	*PGA* Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 349)	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
ORIVIRT2	Ángulo de orientación mediante ejes de orientación virtuales (definición 1)	*PGA* Programación de los ejes de orientación (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 349)	m	●	●	●	●
ORIWKS [4]	Orientación de herramienta en el sistema de coordenadas de pieza	*PGA* Referencia de los ejes de orientación (ORIWKS, ORIMKS) (Página 347)	m	●	●	●	●
OS	Activar/desactivar vaivén (oscilación)	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)		-	-	-	-
OSB	Vaivén: punto inicial	*FB2(P5)*	m	-	-	-	-
OSC	Alisado de la orientación de herramienta constante	*PGA* Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 447)	m	●	●	●	●
OSCILL	Axis: 1 - 3 ejes de penetración	*PGA* Vaivén controlado mediante acciones síncronas (OSCILL) (Página 675)	m	-	-	-	-
OSCTRL	Opciones de vaivén	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)	m	-	-	-	-
OSD	Matado de esquinas de la orientación de herramienta con especificación de la longitud de matado de esquina con DO	*PGA* Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 447)	m	●	●	●	●
OSE	Punto final vaivén	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)	m	-	-	-	-
OSNSC	Vaivén: Número de ciclos de afinado	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)	m	-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
OSOF [4]	Alisado orientación de herramienta DES	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)	m	●	●	●	●
OSP1	Vaivén: Punto de inversión izquierdo	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)	m	-	-	-	-
OSP2	Vaivén: Punto de inversión derecho	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)	m	-	-	-	-
OSS	Alisado de la orientación de herramienta al final de la secuencia	*PGA* Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 447)	m	●	●	●	●
OSSE	Alisado de la orientación de herramienta al principio y al final de la secuencia	*PGA* Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 447)	m	●	●	●	●
OST	Matado de esquinas de la orientación de herramienta con especificación de la tolerancia angular en grados con el DO (desviación máxima del desarrollo de la orientación programado)	*PGA* Orientación de herramienta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 447)	m	●	●	●	●
OST1	Vaivén: punto de parada en el punto de inversión izquierdo	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)	m	-	-	-	-
OST2	Vaivén: punto de parada en el punto de inversión derecho	*PGA* Vaivén asíncrono (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 669)	m	-	-	-	-
OTOL	Tolerancia de orientación para funciones de compresor, alisado de la orientación y tipos de matado de esquinas	*PGA* Tolerancia de orientación/ contorno programable (CTOL, OTOL, ATOL) (Página 515)		-	●	-	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
OVR	Corrección de la velocidad de giro	*PGA*	m	●	●	●	●
OVRA	Corrección axial de la velocidad de giro	*PGA*	m	●	●	●	●
OVRRAP	Corrección en rápido	*PGA*	m	●	●	●	●
P	Número de pasadas del subprograma	*PGA* Cantidad de repeticiones de programa (P) (Página 189)		●	●	●	●
PAROT	Alinear sistema de coordenadas de pieza en la pieza	*PG*	m	●	●	●	●
PAROTOF	Desactivar la rotación de frame relativa a la pieza	*PG*	m	●	●	●	●
PCALL	Subprogramas con ruta de acceso absoluta y transferencia de parámetros	*PGA* Llamada de subprograma con ruta de acceso y parámetros (PCALL) (Página 197)		●	●	●	●
PDELAYOF	Retardo en el troquelado DES	*PGA* Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 685)	m	-	-	-	-
PDELAYON [4]	Retardo en el troquelado CON	*PGA* Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 685)	m	-	-	-	-
PHU	Unidad física de una variable	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
PL	1. B-spline: Distancia entre nodos 2. Interpolación polinómica: Duración del intervalo de parámetros en la interpolación polinómica	*PGA* 1. Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239) 2. Interpolación polinómica (POLY, POLYPATH, PO, PL) (Página 257)	s	- -	○ -	- -	○ -
PM	Por minuto	*PG*		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
PO	Coeficiente de polinomio en la interpolación polinómica	*PGA* Interpolación polinómica (POLY, POLYPATH, PO, PL) (Página 257)	s	-	-	-	-
POCKET3	Ciclo de fresado, caja rectangular (cualquier fresa)	*BHD/BHF*		●	●	●	●
POCKET4	Ciclo de fresado, caja circular (cualquier fresa)	*BHD/BHF*		●	●	●	●
POLF	Posición de retirada LIFTFAST	*PG/PGA*	m	●	●	●	●
POLFA	Iniciar posición de retirada de ejes individuales con $AA_ESR_TRIGGER	*PG*	m	●	●	●	●
POLFMASK	Liberar ejes para la retirada sin relación entre los ejes	*PG*	m	●	●	●	●
POLFMLIN	Liberar ejes para la retirada con relación lineal entre los ejes	*PG*	m	●	●	●	●
POLY	Interpolación de polinomios	*PGA* Interpolación polinómica (POLY, POLYPATH, PO, PL) (Página 257)	m	-	-	-	-
POLYPATH	Interpolación polinómica seleccionable para los grupos de ejes AXIS o VECT	*PGA* Interpolación polinómica (POLY, POLYPATH, PO, PL) (Página 257)	m	-	-	-	-
PON	Troquelado CON	*PGA* Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 685)	m	-	-	-	-
PONS	Troquelado CON en el ciclo IPO	*PGA* Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 685)	m	-	-	-	-
POS	Posicionar eje	*PG*		●	●	●	●
POSA	Posicionar eje incluso tras el final de la secuencia	*PG*		●	●	●	●
POSM	Posicionar almacén	*FBW*		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
POSP	Posicionado en tramos (vaivén)	*PG*		●	●	●	●
POSRANGE	Calcular si la posición teórica interpolada actualmente de un eje se encuentra en una ventana en torno a una posición de referencia predefinida	*PGA* Posición en el margen de referencia definido (POSRANGE) (Página 629)		●	●	●	●
POT	Cuadrado (función aritmética)	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
PR	Por vuelta	*PG*		●	●	●	●
PREPRO	Identificación de subprogramas con preparación	*PGA* Identificación de subprogramas con preparación (PREPRO) (Página 175)		●	●	●	●
PRESETON	Ajuste de valores reales para ejes programados	*PGA* Decalaje de Preset (PRESETON) (Página 307)		●	●	●	●
PRIO	Palabra reservada para definir la prioridad en el tratamiento de interrupciones	*PGA* Asignar e iniciar una rutina de interrupción (SETINT, PRIO, BLSYNC) (Página 117)		●	●	●	●
PROC	Primera instrucción de un programa	*PGA* Llamada de subprograma con ruta de acceso y parámetros (PCALL) (Página 197)		●	●	●	●
PTP	Desplazamiento punto a punto	*PGA* Desplazamiento PTP cartesiano (Página 395)	m	●	●	●	●
PTPG0	Desplazamiento punto a punto sólo con G0; si no, CP	*PGA* PTP con TRANSMIT (Página 401)	m	●	●	●	●
PUNCHACC	Aceleración dependiente del recorrido durante el punzonado	*PGA* Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 685)		-	-	-	-
PUTFTOC	Corrección de herramienta fina para diamantado paralelo	*PGA* Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 427)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
PUTFTOCF	Corrección de herramienta fina dependiente de una función definida con FCTDEF para diamantado paralelo	*PGA* Corrección de herramienta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 427)		●	●	●	●
PW	B-spline, peso del punto	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	s	-	○	-	○
QECLRNOF	Aprendizaje de la compensación de errores de cuadrante DES	*PGA* Aprendizaje de características de compensación (QECLRNON, QECLRNOF) (Página 722)		●	●	●	●
QECLRNON	Aprendizaje de la compensación de errores de cuadrante CON	*PGA* Aprendizaje de características de compensación (QECLRNON, QECLRNOF) (Página 722)		●	●	●	●
QU	Emisión rápida de funciones (auxiliares) adicionales	*PG*		●	●	●	●
R...	Parámetro de cálculo también como identificador de direcciones ajustable y con extensión numérica	*PGA* Variables de usuario predefinidas: Parámetros de cálculo R (Página 19)		●	●	●	●
RAC	Programación por radios específica del eje, por secuencias, absoluta	*PG*	s	●	●	●	●
RDISABLE	Bloqueo de lectura	*PGA* Activar bloqueo de lectura (RDISABLE) (Página 610)		●	●	●	●
READ	Lee una o varias líneas en el fichero indicado y guarda información leída en la matriz	*PGA* Leer líneas en fichero (READ) (Página 141)		●	●	●	●
REAL	Tipo de dato: Variable de coma flotante con signo (números reales)	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
REDEF	Ajuste para datos máquina, elementos de lenguaje CN y variables de sistema en los que se visualizan grupos de usuarios	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
RELEASE	Habilitar ejes de máquina para el intercambio	*PGA* Intercambio de ejes, intercambio de cabezales (RELEASE, GET, GETD) (Página 128)		●	●	●	●
REP	Palabra reservada para inicializar todos los elementos de una matriz con el mismo valor	*PGA* Definición e inicialización de variables de matriz (DEF, SET, REP) (Página 47)		●	●	●	●
REPEAT	Repetición de un bucle de programa	*PGA* Repetición de secciones de programa (REPEAT, REPEATB, ENDLABEL, P) (Página 92)		●	●	●	●
REPEATB	Repetición de una línea de programa	*PGA* Repetición de secciones de programa (REPEAT, REPEATB, ENDLABEL, P) (Página 92)		●	●	●	●
REPOSA	Reposicionamiento en el contorno lineal con todos los ejes	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	s	●	●	●	●
REPOSH	Reposicionamiento en el contorno en semicírculo	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	s	●	●	●	●
REPOSHA	Reposicionamiento en el contorno con todos los ejes; ejes geométricos en semicírculo	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	s	●	●	●	●
REPOSL	Reposicionamiento en el contorno lineal	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	s	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
REPOSQ	Reposicionamiento en el contorno en cuadrante	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	s	•	•	•	•
REPOSQA	Reposicionamiento en el contorno lineal con todos los ejes; ejes geométricos en cuadrante	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	s	•	•	•	•
RESET	Resetear ciclo tecnológico	*PGA* Bloquear, habilitar, resetear (LOCK, UNLOCK, RESET) (Página 662)		•	•	•	•
RESETMON	Instrucción de programación para la activación de valores de consigna	*FBW*		•	•	•	•
RET	Fin de subprograma	*PGA* Salto atrás parametrizable al subprograma (RET ...) (Página 178)		•	•	•	•
RIC	Programación por radios específica del eje, por secuencias, incremental	*PG* Intercambio de ejes, intercambio de cabezales (RELEASE, GET, GETD) (Página 128)	s	•	•	•	•
RINDEX	Definir un índice de un carácter en la cadena de caracteres introducida	*PGA* Búsqueda de caracteres/cadena en una cadena (INDEX, RINDEX, MINDEX, MATCH) (Página 80)		•	•	•	•
RMB	Reposicionamiento en el punto inicial de la secuencia	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	m	•	•	•	•
RME	Reposicionamiento en el punto final de la secuencia	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
RMI [4]	Reposicionamiento en el punto de interrupción	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	m	●	●	●	●
RMN	Reposicionamiento en el punto de trayectoria más cercano	*PGA* Rearranque en contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 502)	m	●	●	●	●
RND	Redondear esquina de contorno	*PG*	s	●	●	●	●
RNDM	Redondeo modal	*PG*	m	●	●	●	●
ROT	Rotación programable	*PG*	s	●	●	●	●
ROTS	Rotaciones de frame programables con ángulos espaciales	*PG*	s	●	●	●	●
ROUND	Redondeo de decimales	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
ROUNDUP	Redondeo al alza de un valor de entrada	*PGA* Redondeo (ROUNDUP) (Página 151)		●	●	●	●
RP	Radio polar	*PG*	m/s	●	●	●	●
RPL	Rotación en el plano	*PG*	s	●	●	●	●
RT	Parámetros para el acceso a datos frame: Rotación	*PGA* Leer y modificar componentes de frame (TR, FI, RT, SC, MI) (Página 300)		●	●	●	●
RTLIOF	G0 sin interpolación lineal (interpolación de ejes individuales)	*PG*	m	●	●	●	●
RTLION	G0 con interpolación lineal	*PG*	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
S	Velocidad de giro del cabezal (otro significado con G4, G96/G961)	*PG*	m/s	●	●	●	●
SAVE	Atributo para salvaguardar información de llamadas de subprograma	*PGA* Almacenamiento de las funciones G modales (SAVE) (Página 164)		●	●	●	●
SBLOF	Supresión de secuencia individual	*PGA* Suprimir la ejecución de secuencia individual (SBLOF, SBLON) (Página 166)		●	●	●	●
SBLON	Cancelación de la supresión de secuencia individual	*PGA* Suprimir la ejecución de secuencia individual (SBLOF, SBLON) (Página 166)		●	●	●	●
SC	Parámetros para el acceso a datos frame: Escala	*PGA* Leer y modificar componentes de frame (TR, FI, RT, SC, MI) (Página 300)		●	●	●	●
SCALE	Escala programable	*PG*	s	●	●	●	●
SCC	Asignación selectiva de un eje de refrentado a G96/G961/G962. Los identificadores de eje pueden ser ejes geométricos, ejes de canal o de máquina.	*PG*		●	●	●	●
SCPARA	Programación de la secuencia de parámetros servo	*PGA* Secuencia de parámetros servo programable (SCPARA) (Página 287)		●	●	●	●
SD	Grado del spline	*PGA* Interpolación spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 239)	s	-	○	-	○
SEFORM	Instrucción de estructuración en el editor Step para generar la vista de pasos para HMI Advanced	*PGA* Instrucción de estructuración en el editor Step (SEFORM) (Página 219)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
SET	Palabra reservada para inicializar todos los elementos de una matriz con valores listados	*PGA* Definición e inicialización de variables de matriz (DEF, SET, REP) (Página 47)		●	●	●	●
SETAL	Activar alarma	*PGA* Alarmas (SETAL) (Página 731)		●	●	●	●
SETDNO	Asignación del número D del filo (CE) de una herramienta (T)	*PGA* Asignación libre de números D: Cambiar nombre de números D (GETDNO, SETDNO) (Página 456)		●	●	●	●
SETINT	Determinación de qué rutina de interrupción se debe activar cuando aparece una entrada NCK	*PGA* Asignar e iniciar una rutina de interrupción (SETINT, PRIO, BLSYNC) (Página 117)		●	●	●	●
SETM	Activación de metas en el canal propio	PGA Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 109)		-	-	-	-
SETMS	Retornar al cabezal maestro especificado en los datos de máquina			●	●	●	●
SETMS(n)	Cabezal n debe actuar como cabezal maestro	*PG*		●	●	●	●
SETMTH	Definir número de porta-herramientas maestro	*FBW*		●	●	●	●
SETPIECE	Define el número de piezas para todas las herramientas asignadas al cabezal	*FBW*		●	●	●	●
SETTA	Definir herramienta como activa en conjunto de desgaste	*FBW*		●	●	●	●
SETTCOR	Modificación de componentes de herramienta teniendo en cuenta todas las limitaciones	*FB1(W1)*		●	●	●	●
SETTIA	Definir herramienta como inactiva en conjunto de desgaste	*FBW*		●	●	●	●
SF	Decalaje del punto inicial para el tallado de roscas	*PG*	m	●	●	●	●
SIN	Seno (función trigon.)	*PGA* Funciones de cálculo (Página 63)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
SIRELAY	Activación de las funciones de seguridad parametrizadas con SIRELIN, SIRELOUT y SIRELTIME	*FBSI*		•	•	•	•
SIRELIN	Inicialización de las magnitudes de entrada del bloque de función	*FBSI*		•	•	•	•
SIRELOUT	Inicialización de las magnitudes de salida del bloque de función	*FBSI*		•	•	•	•
SIRELTIME	Inicialización del temporizador del bloque de función	*FBSI*		•	•	•	•
SLOT1	Ciclo de figuras de fresado, ranuras en círculo	*BHD/BHF*		•	•	•	•
SLOT2	Ciclo de figuras de fresado, ranura circular	*BHD/BHF*		•	•	•	•
SOFT	Aceleración de contorneado con limitación de tirones	*PG*	m	•	•	•	•
SOFTA	Activar la aceleración suave de los ejes programados	*PG*		•	•	•	•
SON	Punzonado CON	*PGA* Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 685)	m	-	-	-	-
SONS	Punzonado CON en el ciclo IPO	*PGA* Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 685)	m	-	-	-	-
SPATH [4]	La trayectoria de referencia para los ejes FGROUP es la longitud de un arco	*PGA* Referencia de trayectoria ajustable (SPATH, UPATH) (Página 264)	m	•	•	•	•
SPCOF	Conmutar cabezal maestro o cabezales de la regulación de posición a regulación de velocidad	*PG*	m	•	•	•	•
SPCON	Conmutar cabezal maestro o cabezales de regulación de velocidad a regulación de posición	*PGA*	m	•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
SPI	Conversión del número de cabezal en un identificador de eje	*PGA* Funciones de eje (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 701)		●	●	●	●
SPIF1 [4]	Entradas/salidas NCK rápidas para troquelado/ punzonado, byte 1	*FB2(N4)*	m	-	-	-	-
SPIF2	Entradas/salidas NCK rápidas para troquelado/ punzonado, byte 2	*FB2(N4)*	m	-	-	-	-
SPLINEPATH	Determinar conjunto spline	*PGA* Conjunto spline (SPLINEPATH) (Página 252)		-	○	-	○
SPN	Cantidad de trayectos parciales por secuencia	*PGA* División automática en segmentos (Página 691)	s	-	-	-	-
SPOF [4]	Carrera DES, troquelado, punzonado DES	*PGA* Punzonado y troquelado activado o desactivado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 685)	m	-	-	-	-
SPOS	Posición del cabezal	*PG*	m	●	●	●	●
SPOSA	Pos. del cabezal tras el final de la secuencia	*PG*	m	●	●	●	●
SPP	Longitud de un trayecto parcial	*PGA* División automática en segmentos (Página 691)	m	-	-	-	-
SQRT	Raíz cuadrada (función aritmética) (square root)	*PGA* Funciones de cálculo (Página 63)		●	●	●	●
SR	Distancia de retirada de vaivén para acción síncrona	*PG*	s	-	-	-	-
SRA	Distancia de retirada de vaivén en entrada externa axial para acción síncrona	*PG*	m	-	-	-	-
ST	Tiempo de afinado de vaivén para acción síncrona	*PG*	s	-	-	-	-
STA	Tiempo de afinado de vaivén axial para acción síncrona	*PG*	m	-	-	-	-

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
START	Arranque de los programas seleccionados simultáneamente por varios canales desde el programa actualmente en ejecución	*PGA* Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 109)		-	-	-	-
STARTFIFO [4]	Mecanizado; simultáneamente llenado del búfer de pretratamiento	*PGA* Ejecución del programa con memoria de pretratamiento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) (Página 490)	m	●	●	●	●
STAT	Posición de las articulaciones	*PGA* Desplazamiento PTP cartesiano (Página 395)	s	●	●	●	●
STOPFIFO	Parada del mecanizado; llenado del búfer de pretratamiento hasta que se reconozca STARTFIFO; búfer de pretratamiento lleno o fin de programa	*PGA* Ejecución del programa con memoria de pretratamiento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) (Página 490)	m	●	●	●	●
STOPRE	Parada de decodificación previa hasta que se hayan ejecutado todas las secuencias preparadas del proceso principal	*PGA* Ejecución del programa con memoria de pretratamiento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) (Página 490)		●	●	●	●
STOPREOF	Suprimir parada de decodificación previa	*PGA* Desactivar la parada de decodificación previa (STOPREOF) (Página 611)		●	●	●	●
STRING	Tipo de dato: Cadena de caracteres	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
STRINGFELD	Selección de un carácter individual de la matriz de strings programada	*PGA* Selección de un carácter único (STRINGVAR, STRINGFELD) (Página 83)		●	●	●	●
STRINGIS	Comprueba para el repertorio del lenguaje CN disponible y, de forma especial para este comando, si existen, son válidos, están definidos o activos nombres de ciclo CN, variables de usuario, macros y nombres de label.	*PGA* Comprobar el repertorio del lenguaje CN disponible (STRINGIS) (Página 713)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
STRINGVAR	Selección de un carácter individual del string programado	*PGA* Selección de un carácter único (STRINGVAR, STRINGFELD) (Página 83)		-	-	-	-
STRLEN	Definir la longitud de un string	*PGA* Definir la longitud de una cadena (STRLEN) (Página 79)		•	•	•	•
SUBSTR	Definir un índice de un carácter en la cadena de caracteres introducida	*PGA* Selección de una cadena parcial (SUBSTR) (Página 82)		•	•	•	•
SUPA	Supresión del decalaje de origen actual, incluidos los decalajes programados, frames de sistema, decalajes con volante (DRF), decalaje de origen externo y desplazamiento superpuesto	*PG*	s	•	•	•	•
SVC	Velocidad de corte de la herramienta	*PG*	m	•	•	•	•
SYNFCT	Evaluación de un polinomio dependiente de una condición en la acción síncrona a desplazamientos	*PGA* Función de sincronización (SYNFCT) (Página 616)		•	•	•	•
SYNR	La lectura de la variable se realiza de forma síncrona, es decir, en el momento de ejecución	*PGA* Definición de variables de usuario (DEF) (Página 24)		•	•	•	•
SYNRW	La lectura y la escritura de la variable se realizan de forma síncrona, es decir, en el momento de ejecución	*PGA* Definición de variables de usuario (DEF) (Página 24)		•	•	•	•
SYNW	La escritura de la variable se realiza de forma síncrona, es decir, en el momento de ejecución	*PGA* Definición de variables de usuario (DEF) (Página 24)		•	•	•	•
T	Llamar herramienta (cambiar sólo si se especifica en dato de máquina; de lo contrario, se requiere el comando M6)	*PG*		•	•	•	•
TAN	Tangente (función trigon.)	*PGA* Funciones de cálculo (Página 63)		•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
TANG	Definición del conjunto de ejes, arrastre tangencial	*PGA* Control tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 477)		-	-	-	-
TANGDEL	Borra la definición del conjunto de ejes, arrastre tangencial	*PGA* Control tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 477)		-	-	-	-
TANGOF	Arrastre tangencial DES	*PGA* Control tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 477)		-	-	-	-
TANGON	Arrastre tangencial CON	*PGA* Control tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 477)		-	-	-	-
TCA	Selección/cambio de herramienta independiente del estado de la herramienta	*FBW*		●	●	●	●
TCARR	Solicitar portaherramientas (número "m")	*PGA* Corrección longitudinal de herramienta para portaherramientas orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 464)		-	●	-	●
TCI	Cambio de herramienta desde la memoria intermedia en el almacén	*FBW*		●	●	●	●
TCOABS [4]	Calcular las componentes longitudinales de la herramienta a partir de la orientación actual de la herramienta	*PGA* Corrección longitudinal de herramienta para portaherramientas orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 464)	m	-	●	-	●
TCOFR	Determinar las componentes longitudinales de la herramienta a partir de la orientación del frame activo	*PGA* Corrección longitudinal de herramienta para portaherramientas orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 464)	m	-	●	-	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
TCOFRX	Determinar la orientación de la herramienta de un frame activo en la selección de herramientas; la herramienta apunta en dirección X	*PGA* Corrección longitudinal de herramienta para portaherramientas orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 464)	m	-	●	-	●
TCOFRY	Determinar la orientación de la herramienta de un frame activo en la selección de herramientas; la herramienta apunta en dirección Y	*PGA* Corrección longitudinal de herramienta para portaherramientas orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 464)	m	-	●	-	●
TCOFRZ	Determinar la orientación de la herramienta de un frame activo en la selección de herramientas; la herramienta apunta en dirección Z	*PGA* Corrección longitudinal de herramienta para portaherramientas orientables (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 464)	m	-	●	-	●
THETA	Ángulo de giro	*PGA* Giros de la orientación de la herramienta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 360)	s	●	●	●	●
TILT	Ángulo lateral	*PGA* Programación de la orientación de herramienta (A..., B..., C..., LEAD, TILT) (Página 337)	m	●	●	●	●
TLIFT	En el control tangencial, inserción de una secuencia intermedia en las esquinas del contorno	*PGA* Control tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 477)		-	-	-	-
TMOF	Desactivar vigilancia de herramienta	*PGA* Vigilancia de herramienta específica de rectificado desde el programa de pieza (TMON, TMOF) (Página 699)		●	●	●	●
TMON	Activar vigilancia de herramienta	*PGA* Vigilancia de herramienta específica de rectificado desde el programa de pieza (TMON, TMOF) (Página 699)		●	●	●	●
TO	Designa el valor final de un bucle contador FOR	*PGA* Bucle contador (FOR ... TO ..., ENDFOR) (Página 104)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
TOFF	Offset de longitud de herramienta en la dirección de la componente longitudinal de herramienta que actúa paralelamente al eje geométrico indicado en el índice.	*PG*	m	•	•	•	•
TOFFL	Offset de longitud de herramienta en la dirección de la componente longitudinal de herramienta L1, L2 ó L3	*PG*	m	•	•	•	•
TOFFOF	Reset corrección de longitud de herramienta online	*PGA* Corrección de longitud de herramienta online (TOFFON, TOFFOF) (Página 468)		•	•	•	•
TOFFON	Activación de la corrección longitudinal de herramienta online	*PGA* Corrección de longitud de herramienta online (TOFFON, TOFFOF) (Página 468)		•	•	•	•
TOFFR	Offset del radio de herramienta	*PG*	m	•	•	•	•
TOFRAME	Orientar el eje Z del sistema de coordenadas de pieza paralelamente a la orientación de la herramienta mediante rotación del frame	*PG*	m	•	•	•	•
TOFRAMEX	Orientar el eje X del sistema de coordenadas de pieza paralelamente a la orientación de la herramienta mediante rotación del frame	*PG*	m	•	•	•	•
TOFRAMEY	Orientar el eje Y del sistema de coordenadas de pieza paralelamente a la orientación de la herramienta mediante rotación del frame	*PG*	m	•	•	•	•
TOFRAMEZ	Como TOFRAME	*PG*	m	•	•	•	•
TOLOWER	Conversión de las letras de una cadena de caracteres en letras minúsculas	*PGA* Conversión minúsculas/ mayúsculas (TOLOWER, TOUPPER) (Página 78)		•	•	•	•

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
TOOLENV	Guardar todos los estados actuales importantes para la evaluación de los datos de herramienta guardados en la memoria	*FB1(W1)*		●	●	●	●
TOROT	Orientar el eje Z del sistema de coordenadas de pieza paralelamente a la orientación de la herramienta mediante rotación del frame	*PG*	m	●	●	●	●
TOROTOF	Rotaciones de frame en dirección a la herramienta DES	*PG*	m	●	●	●	●
TOROTX	Orientar el eje X del sistema de coordenadas de pieza paralelamente a la orientación de la herramienta mediante rotación del frame	*PG*	m	●	●	●	●
TOROTY	Orientar el eje Y del sistema de coordenadas de pieza paralelamente a la orientación de la herramienta mediante rotación del frame	*PG*	m	●	●	●	●
TOROTZ	Como TOROT	*PG*	m	●	●	●	●
TOUPPER	Conversión de las letras de una cadena de caracteres en letras mayúsculas	*PGA* Conversión minúsculas/ mayúsculas (TOLOWER, TOUPPER) (Página 78)		●	●	●	●
TOWBCS	Valores de desgaste en el sistema de coordenadas básico BKS	*PGA* Sistema de coordenadas del mecanizado activo (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 423)	m	-	●	-	●
TOWKCS	Valores de desgaste en el sistema de coordenadas del cabezal de herramienta con transformación cinética (se distingue del MKS por el giro de la herramienta)	*PGA* Sistema de coordenadas del mecanizado activo (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 423)	m	-	●	-	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
TOWMCS	Valores de desgaste en el sistema de coordenadas de máquina (MKS)	*PGA* Sistema de coordenadas del mecanizado activo (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 423)	m	-	●	-	●
TOWSTD	Ajuste básico para correcciones en la longitud de la herramienta	*PGA* Sistema de coordenadas del mecanizado activo (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 423)	m	-	●	-	●
TOWTCS	Valores de desgaste en el sistema de coordenadas de herramienta (punto de referencia del portaherramientas T en el alojamiento del portaherramientas)	*PGA* Sistema de coordenadas del mecanizado activo (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 423)	m	-	●	-	●
TOWWCS	Valores de desgaste en el sistema de coordenadas de pieza (WKS)	*PGA* Sistema de coordenadas del mecanizado activo (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 423)	m	-	●	-	●
TR	Componente de decalaje de una variable frame	*PGA* Leer y modificar componentes de frame (TR, FI, RT, SC, MI) (Página 300)		●	●	●	●
TRAANG	Transformada eje inclinado	*PGA* Eje inclinado (TRAANG) (Página 389)		-	-	○	-
TRACON	Transformada en cascada	*PGA* Transformadas concatenadas (TRACON, TRAFOOF) (Página 408)		-	-	○	-
TRACYL	Cilindro: Transformada de superficie envolvente	*PGA* Transformada de la envolvente del cilindro (TRACYL) (Página 379)		○	○	○	○
TRAFOOF	Desactivación de las transformadas activas en el canal	*PGA* Transformadas concatenadas (TRACON, TRAFOOF) (Página 408)		●	●	●	●
TRAILOF	Arrastre sincronizado de eje DES	*PGA* Arrastre de ejes (TRAILON, TRAILOF) (Página 521)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
TRAILON	Arrastre sincronizado de eje CON	*PGA* Arrastre de ejes (TRAILON, TRAILOF) (Página 521)		●	●	●	●
TRANS	Decalaje programable	*PG*	s	●	●	●	●
TRANSMIT	Transformada polar (mecanizado de superficies frontales)	*PGA* Fresado en piezas torneadas (TRANSMIT) (Página 376)		○	○	○	○
TRAORI	Transformada de 4, 5 ejes, transformada genérica	*PGA* Transformada de tres, cuatro y cinco ejes (TRAORI) (Página 333)		-	●	-	●
TRUE	Constante lógica: Verdadero	*PGA* Definición de variables de usuario (DEF) (Página 24)		●	●	●	●
TRUNC	Supresión de decimales	*PGA* Corrección de precisión en caso de errores de comparación (TRUNC) (Página 68)		●	●	●	●
TU	Ángulo del eje	*PGA* Desplazamiento PTP cartesiano (Página 395)	s	●	●	●	●
TURN	Nº de vueltas para hélices o espirales	*PG*	s	●	●	●	●
ULI	Límite superior de las variables	*PGA* Atributo: Valores límite (LLI, ULI) (Página 37)		●	●	●	●
UNLOCK	Liberar acción síncrona con ID (continuar ciclo tecnológico)	*PGA* Bloquear, habilitar, resetear (LOCK, UNLOCK, RESET) (Página 662)		●	●	●	●
UNTIL	Condición para terminar un bucle REPEAT	*PGA* Bucle de programa con la condición en el inicio del bucle (WHILE, ENDWHILE) (Página 106)		●	●	●	●
UPATH	La trayectoria de referencia para los ejes FGROUP es una curva de parámetros	*PGA* Referencia de trayectoria ajustable (SPATH, UPATH) (Página 264)	m	●	●	●	●
VAR	Palabra reservada: modo de transferencia de parámetros	*PGA* Llamada de subprograma con transferencia de parámetros (EXTERN) (Página 187)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
VELOLIM	Reducción de un aumento de la velocidad axial máxima	*PGA* Corrección porcentual de la velocidad (VELOLIM) (Página 513)	m	●	●	●	●
VELOLIMA	Reducción o aumento de la velocidad axial máxima	*PG*	m	●	●	●	●
WAITC	Esperar hasta que se haya cumplido el criterio de cambio de secuencia de acoplamiento para los ejes/cabezales	*PGA* Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 109)		-	-	○	-
WAITE	Esperar el fin del programa en otro canal.	*PGA* Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 109)		-	-	-	-
WAITM	Esperar la meta en el canal indicado; terminar la secuencia anterior con parada precisa.	*PGA* Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 109)		-	-	-	-
WAITMC	Esperar meta en el canal indicado; parada precisa sólo si los otros canales no han alcanzado aún la meta.	*PGA* Coordinación de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 109)		-	-	-	-
WAITP	Esperar a fin de desplazamiento	*PG*		●	●	●	●
WAITS	Esperar a que se alcance la posición del cabezal	*PG*		●	●	●	●
WALCS0	Limitación del campo de trabajo WKS anulada	*PG*	m	●	●	●	●
WALCS1	Grupo de limitación del campo de trabajo WKS 1 activo	*PG*	m	●	●	●	●
WALCS2	Grupo de limitación del campo de trabajo WKS 2 activo	*PG*	m	●	●	●	●
WALCS3	Grupo de limitación del campo de trabajo WKS 3 activo	*PG*	m	●	●	●	●
WALCS4	Grupo de limitación del campo de trabajo WKS 4 activo	*PG*	m	●	●	●	●
WALCS5	Grupo de limitación del campo de trabajo WKS 5 activo	*PG*	m	●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
WALCS6	Grupo de limitación del campo de trabajo WKS 6 activo	*PG*	m	●	●	●	●
WALCS7	Grupo de limitación del campo de trabajo WKS 7 activo	*PG*	m	●	●	●	●
WALCS8	Grupo de limitación del campo de trabajo WKS 8 activo	*PG*	m	●	●	●	●
WALCS9	Grupo de limitación del campo de trabajo WKS 9 activo	*PG*	m	●	●	●	●
WALCS10	Grupo de limitación del campo de trabajo WKS 10 activo	*PG*	m	●	●	●	●
WALIMOF	Limitación del campo de trabajo BKS DES	*PG*	m	●	●	●	●
WALIMON [4]	Limitación del campo de trabajo BKS CON	*PG*	m	●	●	●	●
WHEN	La acción se ejecuta cíclicamente si se cumple la condición.	*PGA* Comprobación cíclica de la condición (WHEN, WHENEVER, FROM, EVERY) (Página 585)		●	●	●	●
WHENEVER	La acción se ejecuta una vez si se cumple la condición una vez.	*PGA* Comprobación cíclica de la condición (WHEN, WHENEVER, FROM, EVERY) (Página 585)		●	●	●	●
WHILE	Inicio del bucle de programa WHILE	*PGA* Bucle de programa con la condición en el inicio del bucle (WHILE, ENDWHILE) (Página 106)		●	●	●	●
WRITE	Escribir texto en el sistema de ficheros. Inserta una secuencia al final del fichero indicado.	*PGA* Escribir fichero (WRITE) (Página 136)		●	●	●	●
WRTPR	Retarda la tarea de mecanizado sin interrumpir el modo de contorneado			●	●	●	●
X	Nombre de eje	*PG*	m/s	●	●	●	●
XOR	O exclusivo lógico	*PGA* Operaciones de comparación y operaciones lógicas (Página 66)		●	●	●	●

Instrucción	Significado	Consultar la descripción en [1]	W [2]	828D [3] PPU260/261 - - - PPU280/281 D - - - - - F - - - - - D - - - - - F			
Y	Nombre de eje	*PG*	m/s	•	•	•	•
Z	Nombre de eje	*PG*	m/s	•	•	•	•

Anexo

A

A.1 Lista de las abreviaturas

A	Salida
AS	Sistema de automatización
ASCII	American Standard Code for Information Interchange: Código estándar americano para el intercambio de la información
ASIC	Application Specific Integrated Circuit: circuito integrado del usuario
ASUP	Subprograma asíncrono
AV	Preparación del trabajo
AWL	Lista de instrucciones
BA	Modo de operación
BAG	Grupo de modos de operación
BB	Preparado para el servicio
BCD	Binary Coded Decimals: decimales codificados en binario
BHG	Botonera manual
BIN	Ficheros binarios (**Bin**ary Files)
BIOS	Basic Input Output System
BKS	Sistema de coordenadas básico
BOF	Interfaz de usuario (hombre-máquina)
BOT	Boot Files: ficheros de arranque para SIMODRIVE 611 digital
BT	Panel de operador
BTSS	Interfaz de panel de operador
BuB, B&B	Manejo y visualización
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CN	Control numérico
CNC	Computerized Numerical Control: Control Numérico Computerizado
COM	Comunicación
CP	Communication Processor
CPU	Central Processing Unit: Unidad central de proceso
CR	Carriage Return
CRT	Cathode Ray Tube: tubo de imagen
CSB	Central Service Board: Tarjeta del PLC
CTS	Clear To Send: Código de "Preparado para enviar" en transmisiones de datos serie
CUTOM	Cutter radius compensation: Corrección de radio de herramienta
DAU	Convertidor digital-analógico
DB	Bloque de datos en el PLC
DBB	Byte de bloque de datos en el PLC

DBW	Palabra de bloque de datos en el PLC
DBX	Bit de bloque de datos en el PLC
DC	Direct Control: Desplazamiento del eje giratorio por la vía más corta a la posición absoluta dentro de una vuelta
DCD	Carrier Detect
DDE	Dynamic Data Exchange
DEE	Terminal de datos
DIN	Deutsche Industrie Norm: Norma Industrial Alemana
DIO	Data Input/Output: Señalización en la pantalla para la transmisión de datos
DIR	Directory: Directorio
DLL	Dynamic Link Library
DOE	Dispositivo de transferencia de datos
DOS	Disk Operating System
DPM	Dual Port Memory
DPR	RAM de doble acceso
DRAM	Dynamic Random Access Memory
DRF	Differential Resolver Function: Función de resolver diferencial (volante)
DRY	Dry Run: Avance de recorrido de prueba
DSB	Decoding Single Block: Decodificación secuencia a secuencia
DW	Palabra de datos
E	Entrada
E/R	Unidad de alimentación/realimentación (alimentación eléctrica) de SIMODRIVE 611 digital
E/S	Entrada/Salida
EIA-Code	Código para cinta perforada, la cantidad de perforaciones es siempre impar
ENC	Encoder: Captador de posición real
EPROM	Erasable Programmable Read Only Memory (memoria de lectura modificable y programable eléctricamente)
ERROR	Error from printer
FB	Módulo de funciones
FBS	Pantalla plana
FC	Function Call: Módulo de funciones en el PLC
FDB	Base de datos de productos/artículos
FDD	Floppy Disk Drive
FEPROM	Flash-EPROM: Memoria de lectura y escritura
FIFO	First In First Out: Tipo de almacenamiento en memoria sin direccionamiento, en el cual los datos son leídos en el mismo orden en el que fueron almacenados.
FIPO	Interpolador fino
FM	Módulo de función
FPU	Floating Point Unit: Unidad de coma flotante
FRA	Módulo Frame
FRAME	Registro (marco)
FRK	Corrección del radio de la fresa
FST	Feed Stop: Parada de avance

FUP	Esquema de funciones (método de programación para PLC)
GP	Programa básico
GUD	Global User Data: Datos globales del usuario
HD	Hard Disk: Disco duro
HEX	Abreviatura para número hexadecimal
HiFu	Función auxiliar
HMI	Human Machine Interface: Funcionalidad de manejo de SINUMERIK para manejo, programación y simulación.
HMS	Sistema de medida de alta resolución
HSA	Accionamiento de cabezal
HW	Hardware
IF	Desbloqueo de impulsos del módulo de accionamiento
IK (GD)	Comunicación implícita (datos globales)
IKA	Interpolative Compensation: Compensación interpolatoria
IM	Interface-Modul: Módulo de conexión
IMR	Interface-Modul Receive: Módulo de conexión para el servicio de recepción
IMS	Interface-Modul Send: Módulo de conexión para el servicio de transmisión
INC	Increment: Medida incremental
INI	Initializing Data: Datos de inicialización
IPO	Interpolador
ISA	International Standard Architecture
ISO	International Standard Organization
ISO-Code	Código para cinta perforada, la cantidad de perforaciones es siempre par
JOG	Jogging: Modo de ajuste
K1 .. K4	Canales 1 a 4
K-Bus	Bus de comunicación
KD	Giro de coordenadas
KOP	Esquema de contactos (método de programación para PLC)
$K_Ü$	Relación de transmisión
K_v	Factor de amplificación para el lazo de regulación (ganancia)
LCD	Liquid-Crystal Display: Pantalla de cristal líquido
LED	Light-Emitting Diode: Diodo emisor de luz
LF	Line Feed
LMS	Sistema de medida de la posición
LR	Regulador de posición
LUD	Local User Data
MB	Megabyte
MD	Dato de Máquina (DM)
MDA	Manual Data Automatic: Introducción de programa manual
MK	Circuito de medición
MKS	Sistema de coordenadas de máquina
MLFB	Código MLFB
MPF	Main Program File: Programa de pieza en el control numérico (programa principal)

MPI	Multi Port Interface: interfaz multipuntos
MS-	Microsoft (fabricante de software)
MSTT	Panel de mando de máquina
NCK	Numerical Control Kernel: Núcleo de control numérico para la preparación de secuencias, cálculo de los desplazamientos, etc.
NCU	Numerical Control Unit: Unidad de hardware del NCK
NRK	Denominación del sistema operativo del NCK
NST	Señal de interfaz
NURBS	Non-Uniform Rational B-Spline
NV	Decalaje del origen
OB	Módulo de organización en el PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Dispositivo de operación
OPI	Operation Panel Interface: Conexión para el panel de operador
OPT	Options: Opciones
OSI	Open Systems Interconnection: Normalización para la comunicación con ordenadores
P. e. m.	Puesta en marcha
P-Bus	Bus de periferia
PC	Ordenador personal
PCIN	Nombre del software para el intercambio de datos con el control
PCMCIA	Personal Computer Memory Card International Association: Normalización para tarjetas de memoria
PCU	PC Unit: PC-Box (caja central)
PG	Unidad de programación para PLC
PLC	Programmable Logic Control: autómata programable
PLC	PLC en alemán
POS	Posicionamiento
PU	Motor paso a paso
RAM	Random Access Memory: Memoria de programa para lectura y escritura
REF	Función Búsqueda del punto de referencia
REPOS	Función Reposicionar
RISC	Reduced Instruction Set Computer: Tipo de procesador con juego de instrucciones reducido y tiempos de elaboración muy cortos
ROV	Rapid Override: Corrección de entrada
RPA	R-Parameter Active: Área de memoria en NCK para R- NCK para números de parámetro R
RPY	Roll Pitch Yaw: Modo de giro de un sistema de coordenadas
RTS	Request To Send: Conexión de la unidad de transmisión, señal de control de interfaces serie de datos
SBL	Single Block: Secuencia a secuencia
SD	Dato del operador
SDB	Bloque de sistema
SEA	Setting Data Active: Identificación (tipo de fichero) para datos de operador
SFB	Módulo de funciones del sistema

SFC	System Function Call
SK	Pulsador de menú (Softkey)
SKP	Skip: Omitir secuencia
SPF	Sub Program File: Subprograma
SRAM	Memoria estática (con respaldo)
SRK	Corrección del radio del filo
SSFK	Corrección del error del paso de husillo
SSI	Serial Synchron Interface: Interfaz serie síncrona
SW	Software
SYF	System Files: Ficheros de sistema
TEA	Testing Data Active: Identificación para datos de máquina
TO	Tool Offset: Corrección de herramientas
TOA	Tool Offset Active: Identificación (tipo de fichero) para correcciones de herramienta
TRANSMIT	Transform Milling into Turning: Transformación del sistema de coordenadas en un torno para realizar operaciones de fresado
UFR	User Frame: Decalaje del origen
UP	Subprograma
V.24	Interfaz serie (Definición de las líneas de intercambio entre DEE y DÜE)
VSA	Accionamiento de avance
WKS	Sistema de coordenadas de pieza
WKZ	Herramienta
WLK	Corrección longitudinal de herramienta
WOP	Programación orientada al taller
WPD	Work Piece Directory: Directorio de piezas
WRK	Corrección de radio de herramienta
WZK	Corrección de herramientas
WZW	Cambio de herramienta
ZOA	Zero Offset Active: Identificación (tipo de fichero) para datos de decalaje de origen
µC	Microcontrolador

A.2 Su opinión sobre la documentación

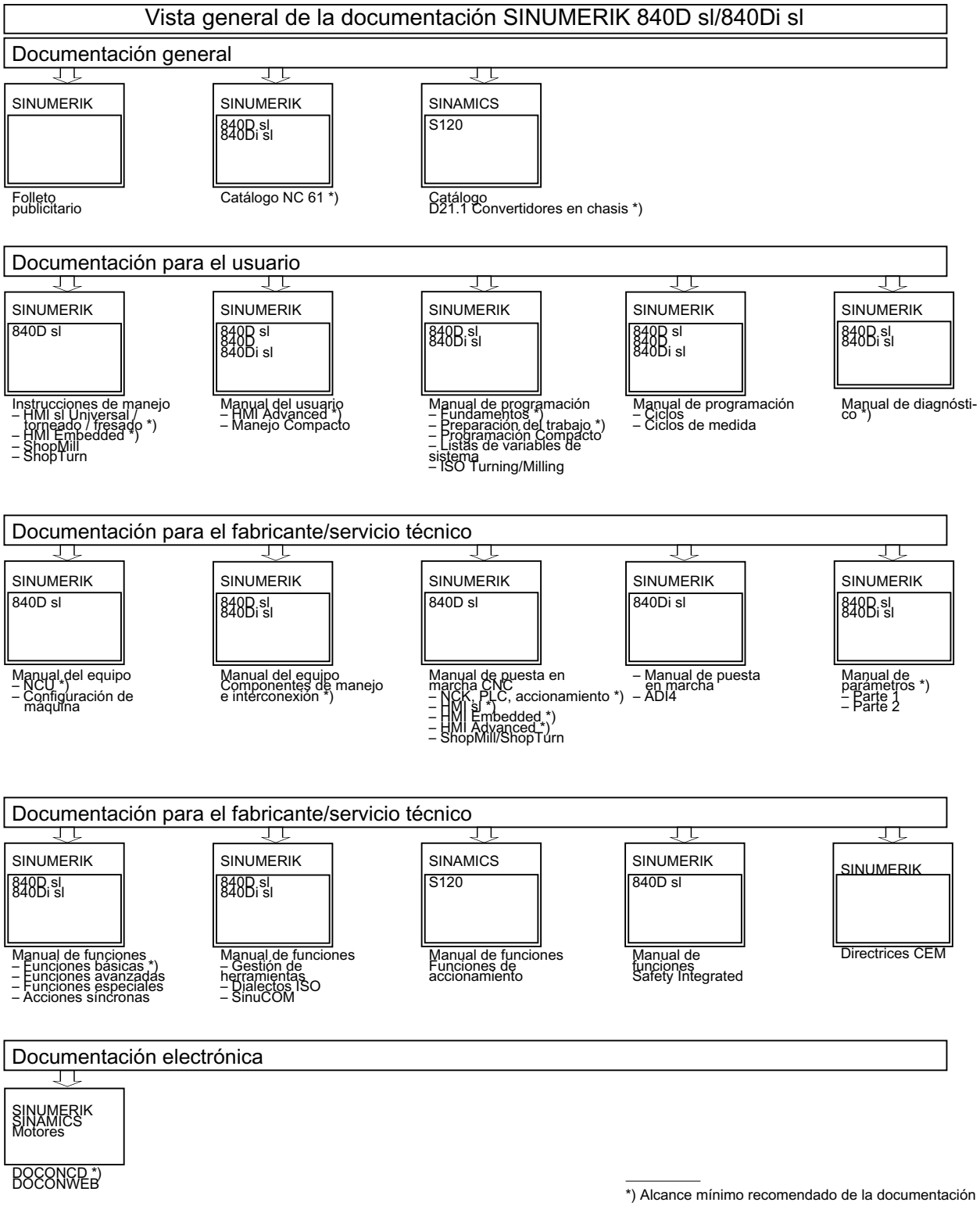
El presente documento está en continuo desarrollo en lo que se refiere a su calidad y facilidad de manejo. Agradeceríamos que nos enviaran sus observaciones y propuestas de mejora por correo electrónico o fax a:

Correo electrónico:	mailto:docu.motioncontrol@siemens.com
Fax:	+49 9131 - 98 2176 Rogamos utilicen la plantilla de fax del reverso.

A SIEMENS AG I DT MC MS1 Postfach 3180 D-91050 Erlangen Fax: +49 9131 - 98 2176 (Documentación)	Remitente Nombre: Empresa/Departamento Calle: C.P.: Localidad: Teléfono: / Telefax: /

Sugerencias y/o correcciones

A.3 Vista general de la documentación



*) Alcance mínimo recomendado de la documentación

Glosario

Accionamiento

El accionamiento es aquella unidad del CNC que regula la velocidad de giro y el par basándose en las especificaciones del CN.

Acciones síncronas

1. Emisión de funciones auxiliares

   Durante el mecanizado de la pieza, se pueden emitir desde el programa CNC funciones tecnológicas (→ funciones auxiliares) al PLC. Con estas funciones se controlan, por ejemplo, dispositivos auxiliares de la máquina herramienta como pinola, pinza, mandril de sujeción, etc.

2. Salida rápida de funciones de ayuda

   Para funciones de conmutación críticas en el tiempo, los tiempos de acuse de recibo para las → funciones auxiliares se pueden reducir al mínimo, evitando puntos de parada innecesarios en el proceso de mecanizado.

Aceleración con limitación de tirones

Para conseguir un comportamiento de aceleración óptimo en la máquina, protegiendo al mismo tiempo la mecánica, se puede conmutar en el programa de mecanizado entre aceleración brusca y aceleración continua (sin sacudidas).

Acotado absoluto

Indicación de la meta de desplazamiento de un eje mediante una cota referida al origen del sistema de coordenadas actualmente válido. Ver → Acotado incremental

Acotado incremental

También medida incremental: indicación de una meta de desplazamiento de un eje a través de un recorrido a ejecutar y una dirección con relación a un punto ya alcanzado. Ver → Acotado absoluto

Alarmas

Todos los → avisos y las alarmas se visualizan en el panel de operador textualmente, con fecha y hora y con el icono correspondiente para el criterio de borrado. La indicación se efectúa separadamente por alarmas y avisos.

1. Alarmas y avisos en el programa de pieza

   Las alarmas y los avisos se pueden visualizar directamente desde el programa de pieza en texto explícito.

2. Alarmas y avisos del PLC

   Las alarmas y los avisos de la máquina se pueden visualizar en texto explícito desde el programa del PLC. Para ello, no se precisan grupos adicionales de módulos funcionales.

Archivo

Copia de ficheros y/o directorios a un dispositivo de memoria **externo**.

Área TOA

El área TOA comprende todos los datos de herramienta y de almacén. Por defecto, este área coincide con el área → Canal por lo que respecta al alcance de los datos. No obstante, a través de los datos de máquina se puede definir que varios canales compartan una → unidad TOA de tal forma que estos canales tengan acceso a datos WZV comunes.

Arrancar

Carga del programa de sistema después de Power On.

Automático

Modo del control (modo de sucesión de secuencias según DIN): modo de operación en sistemas CN en el que se selecciona y ejecuta de forma continua un → programa de pieza.

Avance de contorneado

El avance de contorneado actúa sobre los → ejes de contorneado. Representa la suma geométrica de los avances de los → ejes geométricos afectados.

Avance inverso al tiempo

En SINUMERIK 840D, se puede programar, en lugar de la velocidad de avance para el desplazamiento del eje, el tiempo que debe ocupar la trayectoria de una secuencia (G93).

Avisos

Todos los avisos programados en el programa de pieza y las → alarmas detectadas por el sistema se muestran en el panel de operador textualmente, con fecha y hora y con el correspondiente icono para el criterio de borrado. La indicación se efectúa separadamente por alarmas y avisos.

Bloque

Se denominan módulos a todos los ficheros necesarios para la creación y el procesamiento del programa.

Bloque de datos

1. Unidad de datos del → PLC a la que tienen acceso los programas → HIGHSTEP.
2. Unidad de datos del → CN: Los bloques de datos contienen definiciones de datos para datos globales de usuario. Los datos pueden ser inicializados directamente en el momento de la definición.

Bloque de programa

Los bloques de programa contienen los programas principales y subprogramas de los → programas de piezas.

Borrado general

En el borrado general se eliminan las siguientes memorias de la → CPU:

- → Memoria de trabajo
- Área de escritura/lectura de la → memoria de carga
- → Memoria de sistema
- → Memoria Backup

Búsqueda de número de secuencia

Para comprobar programas de pieza o después de una cancelación del mecanizado, se puede seleccionar con la función "Búsqueda de secuencia" cualquier punto en el programa de pieza en el que se desee iniciar o continuar el mecanizado.

Cables de conexión

Los cables de conexión son cables de 2 hilos con 2 conectores, prefabricados o a fabricar por el mismo usuario. Estos cables de conexión conectan la → CPU a través de la → interfaz multipunto (MPI) con una → PG o con otras CPU.

Canal

Un canal se caracteriza porque puede ejecutar un → programa de pieza independientemente de otros canales. Un canal tiene el control exclusivo de los ejes y cabezales que le han sido asignados. Los ciclos de programa de pieza de distintos canales se pueden coordinar mediante → sincronización.

Canal de mecanizado

A través de una estructura de canales, los movimientos paralelos permiten reducir los tiempos no productivos, p. ej., desplazamiento de un pórtico de carga simultáneamente al mecanizado. En este contexto, un canal CNC se tiene que considerar como control CNC propio con decodificación, preparación de secuencias e interpolación.

Ciclos

Subprogramas protegidos para la ejecución de procesos de mecanizado repetidos en la → pieza.

Ciclos estándar

Para tareas de mecanizado recurrentes se dispone de ciclos estándar:

- para la tecnología Taladrar/Fresar
- para la tecnología Tornear

En el área de manejo "Programa" se listan en el menú "Ayuda de ciclos" los ciclos disponibles. Tras la selección del ciclo de mecanizado deseado se visualizan los parámetros necesarios para la asignación de valores en texto explícito.

Clave de programación

Caracteres y cadenas de caracteres que tienen un significado definido en el lenguaje de programación para → programas de pieza.

CN

Numerical Control: El control CN comprende todos los componentes del control de máquina herramienta. → NCK, → PLC, HMI, → COM.

Nota

Para los controles SINUMERIK 840D sería más correcto utilizar la denominación "control CNC": Computerized Numerical Control.

CNC

Ver → CN

Coincidencia previa

Cambio de secuencia cuando la trayectoria se acerca a un delta predefinido de la posición final.

COM

Componente del control CN para la ejecución y la coordinación de la comunicación.

Compensación de error de paso de husillo

Compensación a través del control de imprecisiones mecánicas de un husillo de bolas que participa en el avance mediante valores medidos consignados de las desviaciones.

Compensación de errores de paso de cuadrante

Los errores de contorno en transiciones de cuadrante producidos por cambios en las condiciones de fricción en guías se pueden eliminar en gran parte con la compensación de errores de cuadrante. La parametrización de la compensación de errores de cuadrante tiene lugar mediante un test circular.

Compensación del juego

Compensación de un juego mecánico de la máquina, p. ej., juego de inversión en husillos de bolas. La compensación del juego se puede introducir por separado para cada eje.

Compensación interpolatoria

Con la ayuda de la compensación interpolatoria se pueden compensar errores de paso del husillo debidos a la fabricación y errores del sistema de medida (SSFK, MSFK).

Contorno

Contorno de la → pieza

Contorno de la pieza

Contorno teórico de la → pieza que se debe ejecutar/mecanizar.

Contorno de pieza acabada

Contorno de la pieza mecanizada. Ver → Pieza en bruto

Control anticipativo dinámico

El control anticipativo dinámico en función de la aceleración permite prácticamente eliminar las imprecisiones del → contorno causadas por errores de seguimiento. De este modo se consigue, incluso con elevadas → velocidades de contorneado, una excelente precisión de mecanizado. El control anticipativo se puede seleccionar y cancelar específicamente para el eje a través del → programa de pieza.

Control de velocidad

Para poder alcanzar una velocidad aceptable en desplazamientos de valor muy reducido por secuencia, puede ajustarse una evaluación anticipativa para varias secuencias (→ Look Ahead).

Coordenadas polares

Sistema de coordenadas que define la posición de un punto en un plano mediante su distancia del origen y el ángulo que forma el vector de radio con un eje definido.

Corrección de herramientas

Consideración de las dimensiones de la herramienta durante el cálculo del trayecto.

Corrección de radio de herramienta

Para poder programar directamente el → contorno deseado de la pieza, el control tendrá que recorrer, teniendo en cuenta el radio de la herramienta utilizada, una trayectoria equidistante con respecto al contorno programado (G41/G42).

Corrección del avance

A la velocidad programada se superpone el ajuste de velocidad actual a través del → panel de mando de la máquina o desde el → PLC (0-200%). La velocidad de avance se puede corregir adicionalmente en el programa de mecanizado mediante un factor de porcentaje programable (1-200%).

Corrección del radio del filo

En la programación de un contorno se parte de una herramienta de punta. Dado que, en la práctica, esto no se puede realizar, se indica al control el radio de curvatura de la herramienta y éste lo tiene en cuenta. El centro de la curvatura se lleva, desplazado en el radio de curvatura, de forma equidistante alrededor del contorno.

Cotas métricas y en pulgadas

En el programa de pieza se pueden programar los valores de posiciones y pasos en pulgadas. Independientemente del acotado programable (G70/G71), el control es ajustado a un sistema básico.

CPU

Central Processing Unit, ver → autómata programable

Curvatura

La curvatura k de un contorno es la inversa del radio r del círculo tangente en un punto del contorno (k = 1/r).

Datos del operador

Datos que comunican las características de la máquina herramienta de una forma definida por el software del sistema al control CN.

Decalaje de origen externo

Decalaje de origen determinado por el → PLC.

Decalaje del origen

Definición de un nuevo punto de referencia para un sistema de coordenadas con relación a un origen existente y un → frame.

1. Configurable

   SINUMERIK 840D: se cuenta con un número definible de decalajes de origen ajustables para cada eje CNC. Los decalajes seleccionables a través de funciones G actúan de forma alternativa.

2. Externo

   Adicionalmente a todos los decalajes, que establecen la posición de origen de la pieza, por medio del volante o del PLC puede ser superpuesto un decalaje de origen externo (decalaje DRF).

3. Programables

   Con la instrucción `TRANS` se pueden programar decalajes de origen para todos los ejes de contorneado y de posicionado.

Definición de variables

Una definición de variables comprende la definición de un tipo de datos y un nombre de variable. Con el nombre de variable se puede activar el valor de la variable.

Desplazamiento a tope fijo

Las máquinas herramienta se pueden desplazar de forma definida a puntos fijos, tales como punto de cambio de herramienta, punto de carga, punto de cambio de palette, etc. Las coordenadas de dichos puntos están consignadas en el control. El control desplaza los ejes en cuestión, si es posible, en → rápido.

Diagnosis

1. Campo de manejo del control
2. *El control cuenta tanto con un programa de autodiagnóstico como con ayudas para la prueba utilizables en el servicio: indicaciones de estado, alarma y servicio.

Dirección

Una dirección es la identificación de un determinado operando o gama de operando (p. ej.: entrada, salida, etc).

DRF

Differential Resolver Function: Función CN que en combinación con un volante electrónico genera un decalaje de origen en Automático.

Editor

El editor permite crear, modificar, completar, agrupar e insertar programas/textos/secuencias de programas.

Editor de textos

Ver → Editor

Eje básico

Ejes cuyo valor teórico o real se utiliza para el cálculo de un valor de compensación.

Eje C

Eje alrededor del cual se ejecuta un movimiento giratorio controlado y un posicionamiento con el cabezal porta piezas.

Eje de compensación

Eje cuyo valor nominal o real se modifica con el valor de compensación.

Eje de contorneado

Ejes de contorneado son todos los ejes de mecanizado del → canal que son conducidos por el → interpolador de modo que arrancan, aceleran, paran y alcanzan el punto final simultáneamente.

Eje de posicionado

Eje que ejecuta un movimiento auxiliar en una máquina herramienta. (p. ej.: almacén de herramientas, transporte de paletas). Los ejes de posicionado son ejes que no interpolan con los → ejes de contorneado.

Eje de redondeo

Los ejes de redondeo producen un giro de la pieza o la herramienta a una posición angular correspondiente a una retícula de división. Al alcanzar una retícula, el eje de redondeo se encuentra "en posición".

Eje geométrico

Los ejes geométricos sirven para la descripción de un área bi o tridimensional en el sistema de coordenadas de pieza.

Eje giratorio

Los ejes giratorios producen un giro de la pieza o la herramienta a una posición angular definida.

Eje lineal

Al contrario que el eje giratorio, el eje lineal es un eje que describe una línea recta.

Eje maestro

El eje maestro es el → eje Gantry existente desde el punto de vista del operador y del programador y, por ello, capaz de ser influido como un eje CN normal.

Eje síncrono

El eje síncrono es el → eje Gantry cuya posición nominal se deriva siempre del desplazamiento del → eje maestro y, de esta forma, se desplaza de manera síncrona. Desde la perspectiva del operador y del programador, el eje síncrono "no existe".

Ejes

Los ejes CNC se clasifican de acuerdo con su funcionalidad en:

- Ejes: ejes de contorneado con interpolación
- Ejes auxiliares: ejes de aproximación y posicionamiento sin interpolación y con avance específico por eje. Los ejes auxiliares no participan en el mecanizado propiamente dicho (p. ej., alimentador de herramientas, almacén de herramientas).

Ejes de máquina

Ejes físicamente presentes en la máquina herramienta.

Ejes síncronos

Para realizar su desplazamiento, los ejes síncronos necesitan el mismo tiempo que los ejes geométricos para su trayectoria.

Entorno de usuario

La interfaz hombre-máquina es el medio de visualización de un control CNC en forma de una pantalla. Está diseñada con pulsadores de menú horizontales y verticales.

Entradas/salidas digitales rápidas

A través de las entradas digitales se pueden, por ejemplo, iniciar rutinas de programa rápidas (rutinas de interrupción). A través de las salidas CNC digitales se pueden activar funciones de maniobra rápidas controladas por el programa (SINUMERIK 840D).

Escala

Componente de un → frame que produce cambios de escala específicos del eje.

Fin de carrera de software

Los fines de carrera de software limitan el área de desplazamiento de un eje y evitan que el carro choque con los fines de carrera de hardware. Por cada eje se pueden definir 2 parejas de valores que se pueden activar por separado a través del → PLC.

Frame

Un frame viene a ser una regla de cálculo que transfiere un sistema de coordenadas cartesianas a otro sistema de coordenadas cartesianas. Un frame contiene los componentes → Decalaje de origen, → Rotación, → Escala, → Simetría especular.

Frames programables

Con los → frames programables se pueden definir de forma dinámica y durante la ejecución del programa de pieza nuevos orígenes del sistema de coordenadas. Se distingue entre la definición absoluta mediante un nuevo frame y la definición aditiva con relación a un origen existente.

funciones auxiliares

Las funciones auxiliares sirven para transferir → parámetros desde → programas de pieza al → PLC, donde provocan reacciones definidas por el fabricante de la máquina.

Funciones de seguridad

El control contiene vigilancias activas en permanencia que detectan los fallos en el → CNC, en el mando de interconexión (→ PLC) y en la máquina de forma tan temprana que se excluye en gran parte la posibilidad de daños en la pieza, la herramienta o la máquina. En caso de un fallo se interrumpe el proceso de mecanizado y los accionamientos se paran; la causa del fallo se memoriza y se visualiza como alarma. Al mismo tiempo se comunica al PLC que existe una alarma CNC pendiente.

Geometría

Descripción de una → pieza en el → sistema de coordenadas de pieza.

Gestión de programas de piezas

La gestión de programas de pieza se puede organizar por → piezas. El tamaño de la memoria de usuario determina el número de programas y datos a gestionar. Cada fichero (programas y datos) se puede dotar de un nombre con máx. 24 caracteres alfanuméricos.

Giro

Componente de un → frame que define un giro del sistema de coordenadas en un ángulo determinado.

Grupo de modos de operación

Los ejes y cabezales que tecnológicamente puedan ser agrupados se pueden asignar a un grupo de modos de operación (BAG). Los ejes y cabezales de un BAG se pueden controlar desde uno o varios → canales. Los canales del BAG siempre tienen asignado el mismo → modo de operación.

Herramienta

Elemento activo en la máquina herramienta que produce el mecanizado (p. ej.: herramienta de tornear, fresa, broca, rayo láser, etc.).

HIGHSTEP

Resumen de las posibilidades de programación para el → PLC del sistema AS300/AS400.

Identificador de eje

Los ejes se denominan X, Y, Z según DIN 66217 para un → sistema de coordenadas cartesiano dextrógiro.

Los → ejes giratorios que giran en torno a X, Y, Z llevan el identificador A, B, C. Los demás ejes paralelos a los señalados pueden llevar otros caracteres de dirección.

Identificadores

Las palabras según DIN 66025 se completan con indicadores (nombres) para variables (variables de cálculo, variables de sistema, variables de usuario), para subprogramas, para palabras reservadas y palabras con varias letras de dirección. Estos complementos se equiparan en su significado a las palabras cuando se trata de la estructura de la secuencia. Los descriptores tienen que ser unívocos. Un mismo descriptor no puede ser utilizado para diferentes objetos.

Interpolación circular

La → herramienta debe desplazarse entre unos puntos definidos del contorno con un avance definido en un círculo, mecanizando la pieza.

Interpolación de polinomios

Con la interpolación de polinomios se pueden generar las formas de curva más diversas, tales como **funciones lineales, parabólovas y exponenciales** (SINUMERIK 840D).

Interpolación helicoidal

La interpolación helicoidal es particularmente apta para la ejecución sencilla de roscas internas o externas con fresas perfiladas y el fresado de ranuras de lubricación.

La línea helicoidal se compone de dos movimientos:

- movimiento circular en un plano
- movimiento lineal perpendicular a dicho plano.

Interpolación lineal

La herramienta se desplaza en línea recta hasta el punto de destino mecanizando la pieza.

Interpolación spline

Con la interpolación spline, el control puede generar una curva lisa a partir de unos pocos puntos de interpolación predefinidos.

Interpolador

Unidad lógica del → NCK que, después de especificar posiciones de destino en el programa de pieza, establece valores intermedios para los movimientos que se deben realizar en los diferentes ejes.

Interruptor llave

El interruptor de llave en el → panel de mando de máquina posee 4 posiciones asignadas por el sistema operativo del control a funciones. Al interruptor le pertenecen tres llaves de distintos colores que se pueden retirar en las posiciones indicadas.

JOG

Modo de operación del control (modo de preparación): el modo JOG sirve para preparar la máquina. De esta forma los ejes individuales y los cabezales se pueden desplazar mediante los pulsadores de dirección. Otras funciones en el modo de operación JOG son la → búsqueda de punto de referencia, → Repos, así como → Preset (establecer el valor real).

KÜ

Relación de transmisión

KV

Factor de ganancia del lazo de regulación; parámetro de un lazo de regulación

Lenguaje de alto nivel CNC

El lenguaje de alto nivel ofrece: → Variable definida por el usuario, → Variable de sistema, → Macros.

Limitación del campo de trabajo programable

Limitación del área de movimiento de la herramienta a un área definida por limitaciones programadas.

Límite de parada precisa

Cuando todos los ejes de contorneado alcanzan su límite de parada precisa, el control se comporta como si hubiera alcanzado exactamente un punto de destino. Se produce un avance de secuencia del → programa de pieza.

Límite del campo de trabajo

El límite del campo de trabajo sirve para limitar el margen de desplazamiento de los ejes además de con los finales de carrera. Por cada eje puede haber dos valores para describir la zona de trabajo protegida.

Look ahead

Con la función **Look Ahead** se consigue, mediante el "control anticipativo" a lo largo de un número parametrizable de secuencias de desplazamiento, una velocidad de mecanizado óptima.

Masa

Como masa se considera la totalidad de los elementos inactivos de un utillaje que, incluso en caso de un fallo, no pueden tomar una tensión al contacto peligrosa.

MDA

Modo de operación del control: Manual Data Automatic. En el modo MDA se pueden introducir secuencias de programa o sucesiones de secuencias individuales que no guardan relación alguna con un programa principal o subprograma y que se pueden ejecutar inmediatamente a continuación mediante la tecla Cycle Start.

Mecanizado en planos oblicuos

Los mecanizados de taladrado y fresado en superficies de pieza que no se sitúan en los planos de coordenadas de la máquina se pueden ejecutar cómodamente con la función "Mecanizado en planos oblicuos".

Medida incremental

Indicación de la longitud del recorrido de desplazamiento mediante el número de incrementos (acotado incremental). El número de incrementos puede estar consignado como → dato de operador o se puede seleccionar con las correspondientes teclas 10, 100, 1.000, 10.000.

Memoria de carga

La memoria de carga es, en la CPU 314 del → PLC, igual a la → memoria de trabajo.

Memoria de corrección

Campo de datos del control en el que están guardados los datos de corrección (correctores) de herramienta.

Memoria de programa del PLC

SINUMERIK 840D: En la memoria PLC de usuario se guardan el programa PLC de usuario y los datos de usuario junto con el programa PLC básico.

Memoria de sistema

La memoria de sistema es una memoria en la CPU donde se guardan los siguientes datos:

- datos necesarios para el sistema operativo
- los operandos Tiempos, Contadores, Marcas

Memoria de trabajo

La memoria de trabajo es una memoria RAM en la → CPU a la cual accede el procesador durante la ejecución del programa al programa de usuario.

Memoria de usuario

Todos los programas y datos, tales como programas de pieza, subprogramas, comentarios, correcciones de herramienta, decalajes de origen/frames, así como datos de usuario de canal y de programa se pueden guardar n la memoria de usuario CNC común.

Modo de contorneado

El objetivo del modo de contorneado es evitar frenadas bruscas de los → ejes de contorneado en los finales de secuencia del programa de pieza y pasar a la siguiente secuencia, a ser posible, con la misma velocidad de contorneado.

Modo de operación

Proceso secuencial para el modo de operación de un control SINUMERIK. Los modos de operación definidos son → JOG, → MDA, → Automático.

Módulo periférico

Los módulos periféricos establecen la conexión entre la CPU y el proceso.

Los módulos periféricos son:

- → Módulos de entrada/salida digitales
- → Módulos de entrada/salida analógicas
- → Módulos de simulador

NCK

Numerical Control Kernel: componente del control CN, que ejecuta → programas de pieza y que principalmente coordina los procesos de desplazamiento para la máquina herramienta.

Nombre eje

Ver → Identificador de eje

NRK

Numeric Robotic Kernel (sistema operativo del → NCK)

NURBS

El guiado de movimiento y la interpolación de trayectoria internos del control se ejecutan sobre la base de NURBS (**N**on **U**niform **R**ational **B**-**S**plines). De este modo se dispone en SINUMERIK 840D, a nivel interno del control, de un procedimiento unitario para todas las interpolaciones.

OEM

Para fabricantes de máquina que quieren crear su propia interfaz hombre-máquina o introducir funciones específicas de la tecnología en el control, se han previsto espacios para soluciones individuales (aplicaciones OEM) para SINUMERIK 840D.

Origen de máquina

Punto fijo de la máquina herramienta del cual parten todos los sistemas de medida (derivados).

Origen de pieza

El origen de la pieza forma el punto inicial del → sistema de coordenadas de pieza. Queda definido por distancias frente al → origen de la máquina.

Override

Posibilidad de intervención manual o programable que permite al operador superponerse a avances o velocidades de giro programados para adaptarlos a una determinada pieza o un material.

Palabra de datos

Unidad de datos de dos bytes dentro de un → bloque de datos.

Palabras reservadas

Palabras con una notación definida que tienen un significado definido en el lenguaje de programación para → programas de pieza.

Panel de mando de máquina

Panel de mando de la máquina herramienta con los elementos de manejo teclas, interruptores giratorios, etc. y elementos de visualización sencillos como LED. Sirve para influir directamente en la máquina herramienta a través del PLC.

Parada de cabezal orientada

Parada del cabezal portapieza en una posición angular definida, p. ej., para realizar en un punto determinado un mecanizado adicional.

Parada precisa

Cuando está programada la instrucción de parada precisa, se busca la posición exacta especificada en una secuencia, a poca velocidad si resulta necesario. Para reducir el tiempo de aproximación se definen → límites de parada precisa para el rápido y el avance.

Parámetros R

Parámetro de cálculo; puede ser activado o consultado por el programador del → programa de pieza para cualquier finalidad en el programa.

Pieza

Pieza a ejecutar/mecanizar por la máquina herramienta.

Pieza en bruto

Pieza con la que se empieza el mecanizado de una pieza.

Pila tampón

La pila tampón garantiza que el → programa de usuario está consignado en la → CPU a prueba de alimentación, y las áreas de datos y marcas, tiempos y contadores definidos se mantienen de forma remanente.

PLC

Programmable **L**ogic **C**ontrol: → autómata programable. Componente del → control CN: mando de interconexión para el procesamiento de la lógica de control de la máquina herramienta.

PLC en alemán

Los autómatas programables (PLC) son controles electrónicos cuya función está almacenada como programa en el aparato de control. Por lo tanto, la estructura y el cableado del aparato no dependen de la función del control. El PLC tiene la estructura de un ordenador; se compone de una CPU (unidad central) con memoria, módulos de entrada/salida y un sistema de bus interno. Los periféricos y el lenguaje de programación están adaptados a las condiciones de la técnica de control.

Programa de pieza

Serie de instrucciones al control CN que producen en su totalidad la ejecución de una determinada → pieza. Asimismo, ejecución de un determinado mecanizado en una → pieza en bruto definida.

Programa de transmisión de datos PCIN

PCIN es un programa auxiliar para la transferencia y la recepción de datos de usuario CNC a través de la interfaz serie (p. ej., programas de pieza, correcciones de herramienta, etc.). El programa PCIN funciona bajo MS-DOS en PC industriales estándar.

Programa de usuario

Los programas de usuario para sistemas de automatización S7-300 se editan con el lenguaje de programación STEP 7. El programa de usuario es modular y consta de distintos bloques.

Los principales tipos de bloques son:

- Bloques de código

  Estos bloques contienen los comandos de STEP 7.

- Bloques de datos

  Estos bloques contienen constantes y variables para el programa STEP 7.

Programa principal

→ Programa de pieza identificado con un número o un identificador en el que se pueden abrir otros programas principales, subprogramas o → ciclos.

Programación del PLC

El PLC se programa con el software **STEP 7**. El software de programación STEP 7 se basa en el sistema operativo estándar **WINDOWS** y contiene las funciones de la programación STEP 5 con nuevos desarrollos innovadores.

Puerto serie V.24

En la PCU 20 hay un puerto serie V.24 (RS232) para la entrada/salida de datos; en la PCU 50/70 hay dos puertos V.24. A través de estos puertos se pueden cargar y guardar programas de mecanizado, así como datos del fabricante y del usuario.

Pulsador de menú (Softkey)

Tecla cuya rotulación se representa mediante un campo en pantalla que se adapta de forma dinámica a la situación de manejo actual. Las teclas de función de libre asignación (pulsadores de menú) se asignan a funciones definidas a nivel del software.

Punto de referencia

Punto de la máquina herramienta al cual se refiere el sistema de medida de los → ejes de máquina.

Punto fijo de la máquina

Punto definido de forma unívoca por la máquina herramienta, p. ej., punto de referencia de la máquina.

Rápido

La velocidad de desplazamiento más rápida de un eje. Se utiliza, p. ej. para aproximar la herramienta desde una posición de reposo al → contorno de la pieza o retirarla de éste. La velocidad en rápido se ajusta específicamente para cada máquina a través del dato de máquina.

Red

Una red es la conexión de varios S7-300 y otros aparatos terminales, p. ej., un PG, a través de → cables de conexión. A través de la red tiene lugar un intercambio de datos entre los aparatos conectados.

Retirada de herramienta orientada

`RETTOOL`: en caso de interrupciones del mecanizado (p. ej., por rotura de herramienta), la herramienta se puede retirar mediante un comando de programa con una orientación especificable y en un recorrido definido.

Retirada rápida del contorno

Cuando llega una interrupción, se puede iniciar, a través del programa de mecanizado CNC, un movimiento que permite la retirada rápida de la herramienta del contorno de pieza que se está mecanizando en este momento. Adicionalmente, se pueden parametrizar el ángulo de retirada y la magnitud del recorrido. Después de la retirada rápida se puede ajustar adicionalmente una rutina de interrupción (SINUMERIK 840D).

Roscado con macho sin mandril de compensación

Esta función permite taladrar roscas sin macho de compensación. Con el desplazamiento interpolado del cabezal como eje giratorio y del eje de taladrado, las roscas se cortan exactamente hasta la profundidad final de taladro, p. ej., taladros ciegos (requisito: servicio de eje del cabezal).

Rutina de interrupción

Las rutinas de interrupción son → subprogramas especiales que pueden ser iniciados por sucesos (señales externas) del proceso de mecanizado. Una secuencia del programa de pieza que se encuentra en ejecución se cancela y la posición de interrupción de los ejes se memoriza automáticamente.

Secuencia auxiliar

Secuencia iniciada por "N" con información para un paso de trabajo, p. ej., una indicación de posición.

Secuencia de programa de pieza

Parte de un → programa de pieza, delimitada por Line Feed. Se distingue entre → secuencias principales y → secuencias subordinadas.

Secuencia principal

Secuencia iniciada por ":" que contiene todos los datos para poder iniciar el ciclo de trabajo en un → programa de pieza.

Secuencias intermedias

Los movimientos de desplazamiento con → corrección de herramienta seleccionada (`G41`/`G42`) se pueden interrumpir con un número limitado de secuencias intermedias (secuencias sin movimientos de ejes en el plano de corrección), y se puede efectuar todavía el cálculo correcto de la corrección de herramienta. El número admisible de secuencias intermedias leídas de forma anticipada por el control se puede ajustar a través de parámetros de sistema.

Simetría especular

En la simetría especular se invierten los signos de los valores de coordenadas de un contorno frente a un eje. La simetría especular se puede aplicar en varios ejes a la vez.

Sincronización

Instrucciones en → programas de pieza para la coordinación de los procesos en distintos → canales en determinados puntos de mecanizado.

Sistema de acotado en pulgadas

Sistema de acotado que define distancias en "pulgadas" y fracciones de ellas.

Sistema de coordenadas

Ver → Sistema de coordenadas de máquina, → Sistema de coordenadas de pieza

Sistema de coordenadas básico

Sistema de coordenadas cartesiano; se refleja por transformación al sistema de coordenadas de máquina.

En el → programa de pieza, el programador utiliza nombres de eje del sistema de coordenadas básico. Si no está activa ninguna → transformación, existe paralelamente al → sistema de coordenadas de máquina. La diferencia reside en los → identificadores de eje.

Sistema de coordenadas de máquina

Sistema de coordenadas que está referido a los ejes de la máquina herramienta.

Sistema de coordenadas de pieza

El sistema de coordenadas de pieza tiene su posición inicial en el → origen de pieza. En la programación en el sistema de coordenadas de pieza, las medidas y las direcciones se refieren a este sistema.

Sistema de medida métrico

Sistema normalizado de unidades: para longitudes, p. ej., mm (milímetros), m (metros).

Spline C

El spline C es el más conocido y utilizado. Las transiciones en los puntos de interpolación son de tangente y curvatura continua. Se utilizan polinomios de tercer grado.

Subprograma

Serie de instrucciones de un → programa de pieza que se puede abrir repetidamente con distintos parámetros. La llamada al subprograma se realiza desde un programa principal. Cada subprograma se puede bloquear contra la emisión y visualización no autorizada.
→ Los ciclos son un tipo de subprogramas.

Subprograma asíncrono

Programa de pieza que se puede iniciar de forma asíncrona (independiente) al estado de programa actual mediante una señal de interrupción (p. ej., señal "Entrada CN rápida").

Tabla de compensación

Tabla de puntos de interpolación. Suministra para posiciones seleccionadas del eje básico los valores de compensación del eje de compensación.

Técnica de macros

Recopilación de una serie de instrucciones bajo un indicador. El indicador representa en el programa la serie de instrucciones reunidas.

Transformada

Decalaje de origen aditivo o absoluto de un eje.

Unidad TOA

Cada → área TOA puede contener varias unidades TOA. El número de unidades TOA posibles está limitado por la cantidad máxima de → canales activos. Una unidad TOA comprende exactamente un bloque de datos de herramienta y un bloque de datos de almacén. Además, puede contener también un bloque de datos de portaherramientas (opcional).

Valor de compensación

Diferencia entre la posición del eje medida por el captador de posición y el posición de eje deseado que se ha programado.

Variable de sistema

Variable que existe sin intervención del programador de un → programa de pieza. Queda definida por un tipo de datos y el nombre de variable que empieza por el carácter **$**. Ver → Variable definida por el usuario.

Variables definidas por el usuario

Los usuarios pueden convenir unas variables definidas por ellos para un uso cualquiera en el → programa de pieza o bloque de datos (datos globales de usuario). Una definición contiene la indicación del tipo de datos y el nombre de variable. Ver → Variable de sistema

Velocidad de contorneado

La máxima velocidad de contorneado programable depende de la precisión de entrada. Con una resolución de, por ejemplo, 0,1 mm, la máxima velocidad de contorneado programable es de 1000 m/min.

Velocidad de giro límite

Velocidad de giro máxima/mínima (del cabezal): especificando datos de máquina, del → PLC o → datos del operador se puede limitar la velocidad de giro máxima de un cabezal.

Velocidad de transmisión

Velocidad en la transferencia de datos (bits/s).

Vigilancia del contorno

Como cota para la precisión de contorno se vigila el error de seguimiento dentro de una banda de tolerancia definida. Un error de seguimiento inadmisiblemente elevado puede producirse, p. ej., por sobrecarga del accionamiento. En dicho caso, se emite una alarma y se paran los ejes.

WinSCP

WinSCP es un programa de código abierto de libre acceso para Windows utilizado para la transferencia de ficheros.

Zona de desplazamiento

La máxima zona de desplazamiento admisible en ejes lineales es de ± 9 décadas. El valor absoluto depende de la precisión de entrada y de regulación de posición seleccionada y del sistema de unidades (pulgadas o métrico).

Zona de trabajo

Zona tridimensional en la que puede entrar la punta de la herramienta debido al diseño de la máquina herramienta. Ver → Zona protegida

Zona protegida

Área tridimensional dentro del → área de trabajo en la cual no debe entrar la punta de la herramienta.

Index

$

$AA_ATOL, 518
$AA_COUP_ACT, 525, 553
$AA_LEAD_SP, 553
$AA_LEAD_SV, 553
$AA_MOTEND, 285
$AA_TOFF[ ], 625
$AC_ACT_PROG_NET_TIME, 726
$AC_ACTUAL_PARTS, 729
$AC_BLOCKTYPE, 604
$AC_BLOCKTYPEINFO, 604
$AC_CTOL, 518
$AC_CUT_INV, 475
$AC_CUTMOD, 475
$AC_CUTMOD_ANG, 475
$AC_CUTTING_TIME, 726
$AC_CYCLE_TIME, 726
$AC_FIFO1, 602
$AC_META, 597
$AC_OLD_PROG_NET_TIME, 726
$AC_OLD_PROG_NET_TIME_COUNT, 726
$AC_OPERATING_TIME, 726
$AC_OTOL, 518
$AC_PARAM, 598
$AC_PROG_NET_TIME_TRIGGER, 727
$AC_REQUIRED_PARTS, 729
$AC_SPECIAL_PARTS, 729
$AC_SPLITBLOCK, 604
$AC_TIMER, 601
$AC_TOTAL_PARTS, 729
$AN_POWER ON_TIME, 725
$AN_SETUP_TIME, 725
$MC_COMPESS_VELO_TOL, 489
$P_AD, 476
$P_CTOL, 519
$P_CUT_INV, 475
$P_CUTMOD, 475
$P_CUTMOD_ANG, 475
$P_OTOL, 519
$P_SUBPAR, 158
$P_TECCYCLE, 656
$PA_ATOL, 519
$R, 599
$Rn, 599
$SA_LEAD_TYPE, 552, 553
$SC_PA_ACTIV_IMMED, 228
$SN_PA_ACTIV_IMMED, 228
$TC_CARR1...14, 458
$TC_CARR18[m], 458, 463
$TC_DP1, 412
$TC_DP10, 412
$TC_DP11, 412
$TC_DP12, 412
$TC_DP13, 412
$TC_DP14, 412
$TC_DP15, 412
$TC_DP16, 412
$TC_DP17, 412
$TC_DP18, 412
$TC_DP19, 412
$TC_DP2, 412
$TC_DP20, 412
$TC_DP21, 412
$TC_DP22, 412
$TC_DP23, 412
$TC_DP24, 412
$TC_DP25, 412
$TC_DP3, 412
$TC_DP4, 412
$TC_DP5, 412
$TC_DP6, 412
$TC_DP7, 412
$TC_DP8, 412
$TC_DP9, 412
$TC_ECPxy, 417
$TC_SCPxy, 417
$TC_TPG1 ... 9, 699, 700

*

* (Función de cálculo), 63

/

/ (Función de cálculo), 63

:

+

+ (Función de cálculo), 63

<

< (Operador de comparación), 66
<<, 72
<< (operador de concatenación), 77
<= (Operador de comparación), 66
<> (Operador de comparación), 66

=

== (Operador de comparación), 66

>

> (Operador de comparación), 66
>= (Operador de comparación), 66

A

A1, A2, 458
A2, 337
A3, 337
A4, 337, 346
A5, 337, 346
A6, 352
A7, 352
ABS, 63
ACC, 569
Acción síncrona
  Acción, 587
  Ámbito de validez, 583
  Borrar, 664
  Cancelar, 664
  Condición, 585
  Elementos de mando (instrucción), 582
  Posicionar eje, 627
  Sintaxis, 582
Acciones síncronas
  Lista de las acciones, 607
  Variable de proceso principal, 591
  Variables de decodificación previa, 591
Acoplamiento, 477
acoplamiento de valores de consigna, 566
Acoplamiento de valores maestros
  Acoplamiento de valores reales y de consigna, 548
  Acoplamiento de valores reales y de valores de Consigna, 552
  Desde acciones síncronas estáticas, 549
  Sincronización eje maestro y esclavo, 551
Acoplamiento de valores reales, 566
Acoplamiento de velocidad, 566
Acoplamiento entre ejes de valores maestros, 548
ACOS, 63
ACTBLOCNO, 172
ACTFRAME, 291
Activación de carrera, 688
ADISPOSA, 283
Alarma, 731
  Comportamiento en acciones síncronas, 667
  Número, 731
Alarmas de ciclo, 731
ALF, 121, 124
Alisado
  del desarrollo de la orientación, 374
Alisamiento del desarrollo de la orientación, 366, 369
AND, 66
Ángulo de avance, 338
Ángulo de decalaje, 571
Ángulo de giro, 361
Ángulo de tangente de trayectoria, 650
Ángulo de torsión 1, 2, 458
Ángulo final, 361
Ángulo lateral, 338
APR, 41
APRB, 41
Aprendizaje de características de compensación, 722
APRP, 41
APW, 41
APWB, 41
APWP, 41
Archivo
  Información, 146
Arrancar/parar eje, 630
Arrastre de ejes, 521, 639
  Limitación de la dinámica, 524
AS, 208
ASIN, 63
ASPLINE, 239
ASUP, 115
ATAN2, 63
ATOL, 515
Atributos de posición
  Programar indirectamente, 59
AV, 563
Avance por eje, 635
AX, 701
AXCTSWE, 709
AXCTSWED, 709
AXIS, 24
AXNAME, 75, 701
AXSTRING, 701
AXTOCHAN, 133
AXTOSPI, 701

B

B_AND, 66
B_NOT, 66
B_OR, 66
B_XOR, 66
B2, 337
B3, 337
B4, 337, 346
B5, 337, 346
B6, 352
B7, 352
BAUTO, 239
BFRAME, 291
BLOCK, 194
Bloqueo de lectura, 610
BLSYNC, 117
BNAT, 239
BOOL, 24
Borrado de trayecto residual, 277, 612
Borrado de trayecto residual con preparación, 612
BOUND, 70
BSPLINE, 239
BTAN, 239
Bucle
  Bucle contador, 104
  Bucle final, 103
  Bucle IF, 101
  Bucle REPEAT, 107
  Bucle WHILE, 106
Bucle contador, 104
Bucle sin fin, 103

C

C2, 337
C3, 337
C4, 337, 346
C5, 337, 346
C6, 352
C7, 352
Cabezal
  Intercambio, 128
Cabezal síncrono, 562
  Definición de un par, 569
  Relación de transformación kT, 570
CAC, 237
CACN, 237
CACP, 237
CALCDAT, 750
Cálculo de frame
  MEAFRAME, 309
CALL, 193
CALLPATH, 198, 215
Cambio de estación/posición, 709
Campo
  Elemento, 47
CANCEL, 664
Carácter 0, 74
CASE, 89
CDC, 237
CFINE, 304
CIC, 237
Ciclos
  Parametrizar ciclos de usuario, 204
Ciclos de SIEMENS, 731
Ciclos tecnológicos
  Concatenaciones en cascada, 659
Ciclos tecnológicos, 653
  Controlar la ejecución cíclica con ICYCOF, 658
  Parámetros por defecto con valores iniciales, 658
Ciclos tecnológicos
  en acciones síncronas por secuencias, 660
Ciclos tecnológicos
  Estructuras de control IF, 660
Ciclos tecnológicos
  Instrucciones de salto GOTO, GOTOF y GOTOB, 661
Ciclos tecnológicos
  Saltos incondicionales, 661
Cinemática
  Simple, 463
Cinemática simple, 458
CLEARM, 109, 646
CLRINT, 120
CMIRROR, 63, 297
COARSE, 563
COARSEA, 283
Código G, 264
  Programar indirectamente, 58
Coeficiente de polinomio, 258
Comandos de programación
  Lista, 753
COMCAD, 254
COMPCAD, 370
COMPCURV, 254, 370
Compensación de errores de cuadrante
  Optimización, 723
Compensación de errores de paso de cuadrante
  Activar el proceso de aprendizaje, 722
  Desactivar proceso de aprendizaje, 722
COMPLETE, 216
COMPOF, 254, 370
COMPON, 254, 370, 489
Componente frame
  FI, 300

MI, 300
SC, 300
TR, 300
Componente frame RT, 300
Compresor, 254, 266
Compresor de secuencias CN, 254
con división de la trayectoria, 695
Concatenación
de cadenas, 77
concatenadas
Transformadas, 408
Conjunto de ejes, 521
Conjunto spline, 252
CONTDCON, 742
Contenedor de ejes, 709
Contorno
Codificación, 742
Preparación, 734
Rearranque, 502
Tabla, 734, 742
CONTPRON, 734
Control
estructuras, 100
Control de potencia del láser, 615
Control tangencial, 477
Coordinación de ejes, 636
Coordinación de programas
Nombres de canal, 111
Números de canal, 111
Corrección contorno-normal OFFN, 387
Corrección de herramienta 3D, 437
Corrección sobre la trayectoria, 438
Curvatura de la trayectoria, 439
Fresado periférico:con superficies de limitación, 443
Método de punto de intersección, 442
Orientación de la herramienta, 447
Profundidad de penetración, 439
Corrección de herramientas
Memoria de corrección, 411
Online, 427, 622
Sistema de coordenadas para valores de desgaste, 423
Corrección de los ejes giratorios, 461
Corrección de radio de herramienta
Deceleración en los dos vértices, 282
Fresado periférico 3D sin superficies de limitación, 443
Corrección de radio de herramienta 3D
Circunferencia de transición, 441
Esquinas interiores/exteriores, 441
Fresado frontal, 436
Fresado periférico, 435
Punto de intersección 3D de las equidistantes, 441
Corrección de radio de herramienta 3D, 433
Corrección longitudinal de herramienta online, 468, 625
COS, 63
COUPDEF, 563
COUPDEL, 563
COUPOF, 563
COUPOFS, 563
COUPON, 563
COUPONC, 563
COUPRES, 563
CP, 395
CPROT, 225
CPROTDEF, 221
Criterio de fin del movimiento
Programable, 283
CROT, 63, 297
CSCALE, 63, 297
CSPLINE, 239
CT, 709
CTAB, 540
CTABDEF, 527
CTABDEL, 535
CTABEND, 527
CTABEXISTS, 535
CTABFNO, 546
CTABFPOL, 546
CTABFSEG, 546
CTABID, 539
CTABINV, 540
CTABISLOCK, 539
CTABLOCK, 537
CTABMEMTYP, 539
CTABMPOL, 546
CTABMSEG, 546
CTABNO, 546
CTABNOMEM, 546
CTABPERIOD, 539
CTABPOL, 546
CTABPOLID, 546
CTABSEG, 546
CTABSEGID, 546
CTABSEV, 540
CTABSSV, 540
CTABTEP, 540
CTABTEV, 540
CTABTMAX, 540
CTABTMIN, 540
CTABTSP, 540
CTABTSV, 540
CTABUNLOCK, 537
CTOL, 515
CTRANS, 63, 297, 304

CUT3DC, 433, 438
CUT3DCC, 443
CUT3DCCD, 443
CUT3DF, 433
CUT3DFF, 433
CUT3DFS, 433
CUTMOD, 471

Ch

CHAN, 24
CHANDATA, 216
CHAR, 24
CHECKSUM, 149
CHKDNO, 455

D

Datos de circunferencia
    no Siemens, 750
De orientación
    Ejes, 352
    Interpolación, 354
Decalaje basto, 304
Decalaje de origen externo, 306
Decalaje de Preset, 307
Decalaje del origen
    Decalaje de origen externo, 306
    PRESETON, 307
Decalaje fino, 304
Deceleración en los dos vértices en esquinas interiores, 282
Deceleración en los dos vértices en todas las esquinas, 282
DEF, 24, 47, 653
DEFAULT, 89
DEFINE, 653
DEFINE ... AS, 208
Definición de matrices, 47
Definición de polinomio, 614
DELAYFSTOF, 493
DELAYFSTON, 493
DELDL, 418
DELDTG, 612
DELETE, 139
Denominador polinómico, 262
Desbaste, 733
Desplazamiento del eje individual, 696
Desplazamiento PTP cartesiano, 327
Desplazamientos de posicionado, 627
Direcciones
    Programar indirectamente, 55
Direcciones OEM, 281
DISABLE, 119
DISPLOF, 172
DISPLON, 172
Disponibilidad
    dependiente del sistema, 5
DISPR, 502
DIV, 63
División automática en segmentos, 691
División en ejes de contorneado, 694
DL, 415
DM20800, 176
DO, 587
DO42475, 371
DO42476, 371
DO42477, 371
DO42678, 374
DO42680, 374
DO42900, 421
DO42910, 421
DO42920, 422
DO42930, 422
DO42935, 425
DO42940, 426, 474
DO42984, 472
DV, 563

E

EAUTO, 239
EG
    Reductor electrónico, 554
EGDEF, 554
EGDEL, 561
EGOFC, 559
EGOFS, 559
EGON, 556
EGONSYN, 556
EGONSYNE, 556
Eje
    Aceptar directamente, 128
    Amarre, 709
    Arrastrado, 524
    Intercambio, 128
    Locales, 710
    Oblicuo (TRAANG), 389
Eje análogo, 620
Eje arrastrado o esclavo, 548
Eje de enlace, 710
Eje geométrico
    Conmutar, 704
Eje inclinado, TRAANG, 326

Eje maestro, 548
Ejes de comando, 627
Ejes de orientación, 337, 347, 349
Ejes FGROUP, 264
Ejes geométricos conmutables, 704
Ejes giratorios
  Vectores de dirección V1, V2, 458
  Vectores de distancia l1, l2, 458
Elementos de contorno
  Retirada, 749
ELSE, 101
ENABLE, 119
ENAT, 239
ENDFOR, 104
ENDIF, 101
ENDLABEL, 92
ENDLOOP, 103
ENDPROC., 621
ENDWHILE, 106
Estado de acoplamiento, 525, 553
Estado de la orden de medición, 279
Estado del palpador, 279
Estado inicial de la orientación de herramienta
ORIRESET, 335
ETAN, 239
Evaluación de la utilización, 651
EVERY, 585
EXECSTRING, 62
EXECTAB, 749
EXECUTE, 221, 752
EXP, 63
EXTCALL, 200
EXTERN, 187

F

F10, 221
F3, 722
FA, 563, 635
Factor de acoplamiento, 521
FALSE, 24
FCTDEF, 427, 614
FCUB, 484
Feje, 477, 548
FENDNORM, 282
FIFOCTRL, 490
FILEDATE, 146
FILEINFO, 146
FILESIZE, 146
FILESTAT, 146
FILETIME, 146
Finales de carrera software, 635
FINE, 563
FINEA, 283
FLIN, 484
FNORM, 484
FOCOF, 647
FOCON, 647
FOR, 104
Formas de fresa, 437
FPO, 484
FPR, 561
Frame
  Concatenación de frames, 319
  Llamar, 301
FRAME, 24
Frame ajustable actual, 317
Frame básico total, 316, 317
Frame programable actual, 317
Frame total actual, 318
Frames
  Asignar, 302
  Cadenas de frames, 303
Frames básicos de canal actuales, 316
Frames básicos globales NCU, 313
Frames básicos globales NCU actuales, 315
Frames de ajuste globales NCU, 313
Frames de sistema actuales, 315
Frames específicos del canal, 314
Fresa
  Punta (FS), 440
  Punto auxiliar (FH), 440
Fresado frontal, 433, 436
Fresado frontal 3D, 345
  Curvatura de la trayectoria a través de vectores normales a la superficie, 346
Fresado periférico, 434, 435
Fresado periférico (3D)
  con superficies de limitación, 443
Fresado periférico 3D con superficies de Limitación, 443
FROM, 585
FTOC, 622
FTOCOF, 427
FTOCON, 427
Función auxiliar, 609
Función de evaluación, 616
Función de medición ampliada, 395
Funciones auxiliares, 691
Funciones OEM, 281
FXS, 647
FXST, 647
FXSW, 647

G

G05, 394
G07, 394
G40, 433
G450, 441
G451, 441
G62, 282
G621, 282
G643, 265
GEOAX, 704
GET, 128
GET automático", 132
GETACTTD, 457
GETD, 128
GETDNO, 456
Giros del vector de orientación Programación con THETA, 360
GOTO, 86
GOTOB, 86
GOTOC, 86
GOTOF, 86
GOTOS, 85
GP, 59
GUD, 24, 212

H

Herramienta
  Corrección del radio, 419
  Corrección longitudinal, 464
  Correcciones, aditivas, 415
  de herramienta, 411
  Memoria de corrección, 411
  Orientación al producirse un cambio de frame, 466
  Orientación, alisado, 374
  Vigilancia, específica de rectificado, 699

I

I1,I2, 458
ICYCOF, 658
ICYCON, 658
ID, 583
Identificador de eje por defecto, 596
IDS, 583
IF, 86, 101
IFRAME, 291
II1,II2, 679
INDEX, 80
Índice de matriz, 50
INICF, 24
Inicialización
  de matrices, 47
  de variables de matriz, 645
INIPO, 24
INIRE, 24
INIT, 109
INITIAL, 216
INITIAL_INI, 216
Instrucción de salto
  CASE, 89
Instrucciones
  Lista, 753
INT, 24
Intercambio de eje, 133
  Aceptar eje, 132
  Ajuste modificable del comportamiento, 132
  Liberar eje, 131
  Requisitos, 131
  sin parada de decodificación previa, 132
  sin sincronización, 130
  Solicitar y liberar a través de acciones síncronas, 631
Interpolación de orientación, 368
Interpolación de polinomios, 257
  Denominador polinómico, 262
Interpolación del vector de giro, 360, 367
Interpolación lineal, 266
Interpolación spline, 266
INTERSEC, 747
Inversión
  Punto, 675
IPOBRKA, 283
IPOENDA, 283
IPOSTOP, 563
IPTRLOCK, 499
IPTRUNLOCK, 499
ISAXIS, 701
ISD, 433, 438
ISFILE, 144
ISNUMBER, 75
ISOCALL, 196
ISVAR, 720

J

JERKLIM, 512
Juego, 722

L

L..., 185
Label, 92

LEAD, 337
LEADOF, 548
Leje, 477, 548
LIFTFAST, 121
Límites en transformadas, 405
LN, 63
LOCK, 662
LOOP, 103
LUD, 24

LI

Llamada de subprograma con ruta de acceso y parámetros, 197
Llamada de subprograma modal, 191
LLI, 37
LLIMIT, 614

M

M, 460
M17, 176
M30, 176
Macro, 208
Marcha síncrona
  Fina, 566
  Somero, 566
MASLDEF, 576
MASLDEL, 576
MASLOF, 576
MASLOFS, 576
MASLON, 576
Matas de espera, 646
MATCH, 80
Matriz, 47
MAXVAL, 70
MCALL, 191
MEAC, 270
MEAFRAME, 309
MEAFRAME, 309
MEAFRAME, 313
MEAS, 267
MEASA, 270
MEAW, 267
MEAWA, 270
Medición, 644
Memoria
  de trabajo, 216
  Memoria de programas, 211
  Pretratamiento, 490
Memoria de corrección, 411
Memoria de programas, 211
  Directorios estándar, 212
  Tipos de fichero, 212
Memoria de trabajo, 216
  Área de datos, 216
MINDEX, 80
MINVAL, 70
MIRROR, 291
MMC, 724
MOD, 63
MODAXVAL, 701
Modo de acoplamiento, 566
Modo de operación
  en la medición, 276
MOV, 630
Movimiento de vaivén
  de inversión derecho, 679
  Penetración en el punto de inversión, 681
  Suprimir penetración, 679
  Zona de inversión, 679
Movimientos de cabezales, 638
MPF, 212, 722
MU, 392
MZ, 392

N

NCK, 24
NEWCONF, 135
NOC, 563
NOT, 66
NPROT, 225
NPROTDEF, 221
NUMBER, 75
Número D
  Asignación libre, 454
Número de identificación, 583
Número del filo, 454
Números D
  Cambiar nombre, 456
  Comprobar, 455
NUT=ángulo, 352

O

OEMIPO1/2, 281
OFFN, 376, 379
Offset de ángulo/incremento de ángulo de los ejes giratorios, 461
Operadores de comparación, 66
Operadores lógicos, 66
OR, 66
ORIAXES, 349, 366

ORIC, 447
ORICONCCW, 352, 366
ORICONCW, 352, 366
ORICONIO, 352, 366
ORICONTO, 352, 366
ORICURVE, 356, 366
ORID, 447
Orientación de la herramienta, 447
Orientación relativa a la trayectoria
  Giro de la orientación de herramienta, 366
  Giro del vector de orientación, 366
  Giros de la herramienta, 365
  Insertar secuencias intermedias, 369
ORIEULER, 349, 366
ORIMKS, 347, 349
ORIPATH, 365
ORIPATHS, 365, 369
ORIPLANE, 352, 366
ORIRESET(A, B, C), 334
ORIROTA, 360
ORIROTC, 360, 366
ORIROTR, 360
ORIROTT, 360
ORIRPY, 349, 366
ORIRPY2, 349
ORIS, 447
ORISOF, 374
ORISON, 374
ORIVECT, 349, 366
ORIVIRT1, 349, 366
ORIVIRT2, 349, 366
ORIWKS, 347, 349
OS, 669
OSB, 669
OSC, 447
OSCILL, 675, 679
OSCTRL, 669
OSD, 447
OSE, 669
OSNSC, 669
OSOF, 447
OSP1, 669
OSP2, 669
OSS, 447
OSSE, 447
OST, 447
OST1, 669
OST2, 669
OTOL, 515
OVRA, 569

P

P..., 189
Parada de decodificación previa, 611
Parámetro
  Actual, 156
  Formal, 156
Parámetro Call-By-Value
  para ciclos tecnológicos, 657
Parámetro de cálculo
  Número n, 19, 21
Parámetros
  Parámetros, 411
  Transferencia en llamada de subprograma, 157, 187
Parámetros de cálculo R, 19, 21
Parámetros para acciones síncronas, 598
Parámetros R, 599
PCALL, 197
PDELAYOF, 685
PDELAYON, 685
Penetración
  a profundidad de trabajo, 682
  Eje, 677
PFRAME, 291
PHI, 352, 359
PHU, 39
Pieza
  Contador, 729
Pieza
  Directorio raíz, 212
  Directorios, 213
PL, 239, 257
PO, 257
PO[PHI], 359, 365
PO[PHI]=(a2, a3, a4, a5), 352
PO[PSI], 359, 365
PO[PSI]=(b2, b3, b4, b5), 352
PO[THT], 359, 365
PO[XH], 359
PO[XH]=(xe, x2, x3, x4, x5), 356
PO[YH], 359
PO[YH]=(ye, y2, y3, y4, y5), 356
PO[ZH], 359
PO[ZH]=(ze, z2, z3, z4, z5), 356
Polinomio
  Interpolación, 266
Polinomio de palabra F, 264
POLY, 257
POLYPATH, 257
PON, 695
Poner valor real, 637
PONS, 685

Por eje
Avance, 635
Portaherramientas, 464
Borrar/modificar/leer datos, 463
Cinemática, 458
Orientable, 464
Portaherramientas orientables
Número del portaherramientas, 460
Variables del sistema, 459
Portaherramientas orientables, 458
POS, 627
POSFS, 563
Posición de desactivación, 573, 574
Posición mínima/posición máxima del eje giratorio, 461
Posicionado desde el punto de trayectoria más cercano, 509
Posicionar eje
Posición de referencia especificada, 629
Posiciones singulares, 349
POSP, 675
POSRANGE, 629
POT, 63
Preparación del contorno
Señalización de error, 752
PREPRO, 175
PRESETON, 307, 637
Pretratamiento
Memoria, 490
Primer frame básico actual en el canal, 316
Primer frame básico en el canal, 314
PRIO, 117, 121
PRLOC, 24
PROC, 160
Profundidad de imbricado
de estructuras de control, 100
Profundidad de penetración, 439
Profundidad de penetración (ISD), 433
Programa
Bifurcación, 89
de inicialización, 216
Memoria, 213
Repetición, 189
Saltos, 86
Tiempos de ejecución, 725
Programa de inicialización, 216
Programación de la orientación de la herramienta, 350, 367
Programación indirecta, 59
Códigos G, 58
Direcciones, 55
Programar eje inclinado
G05, G07, 393
Protegidas
Zonas, 221
PSI, 352, 359
PTP, 395, 401
PTP con TRANSMIT, 401
PTPG0, 401
PUD, 24
PUNCHACC, 685
Puntero de interrupción automático, 501
Punzonado, 685, 691
PUTFTOC, 427
PUTFTOCF, 427
PW, 239

Q

QECDAT, 722
QECLRN, 722
QECLRNOF, 722
QECLRNON, 722
QECTEST, 722
QFK, 722

R

R..., 19, 21
RDISABLE, 610
READ, 141
REAL, 24
REDEF, 31
redondear, 151
Reductor electrónico, 554
Referencia a la trayectoria
Ajustable, 264
Avance de contorneado, 266
Ejes de contorneado, 266
Grupo de códigos G, 264
Interpolación circular e interpolación lineal, 266
Limitaciones, 266
Parámetros de curva, 264
Secuencias de roscado, 266
Refpos, 629
Registrar y buscar áreas no aptas para la búsqueda, 500
Regulación AC, aditiva, 617
Regulación AC, multiplicativa, 619
RELEASE, 128
REP, 47, 645
REPEAT, 92, 107
REPEATB, 92
Repetición de partes del programa
con programación indirecta CALL, 194
REPOS, 115

REPOSA, 502
REPOSH, 502
REPOSHA, 502
Reposicionamiento en el contorno
  Aproximación con nueva herramienta, 510
  Punto de reposicionamiento, 507
REPOSL, 502
REPOSQ, 502
REPOSQA, 502
RESET, 662
RET, 177, 178
Retirada rápida del contorno, 121
RINDEX, 80
RMB, 502
RME, 502
RMI, 502
RMN, 502
ROUND, 63
ROUNDUP, 151
Rozamiento, 722
Ruta de acceso
  Absoluto, 109
  Relativo, 110
Ruta de búsqueda
  en llamada a subprograma, 155, 214
  Ruta de búsqueda programable, 198
Rutina de interrupción, 115
  Almacenamiento de las funciones G modales, 116
  Asignación y arranque, 117
  Desconexión/conexión, 119
  Dirección de desplazamiento programable, 122
  Movimiento de retroceso, 124
  Reasignar, 118
  Retirada rápida del contorno, 121
  Sentido de desplazamiento programable, 124
Rutinas de conversión, 593

S

S1, S2, 563
Salto
  al inicio del programa, 85
  Condición, 87
  Destino, 86
  Instrucción, 86
  Marca, 87, 92
SAVE, 164
SBLOF, 166
SBLON, 166
SC, 477
SCPARA, 287
SD, 239
Sección de programa
  Repetición, 92
Secuencia a secuencia
  Supresión, 166
Secuencia de parada, 500
Secuencia de parámetros servo
  Programable, 287
SEFORM, 219
Segmento, 691
Segmentos, 691
Seleccionar caracteres individuales, 83
SET, 47, 645
SETAL, 646, 731
SETDNO, 456
SETINT, 117
SETM, 109, 646
Simulación de valores maestros, 553
SIN, 63
Sincronismo de posición, 563
Sistema
  Disponibilidad dependiente de, 5
Sobreaceleración (tirón)
  Corrección, 512
SON, 685, 694, 695
SONS, 685
SPATH, 264
SPF, 212, 722
SPI, 701
SPIF1, 685
SPIF2, 685
Spline
  Interpolación, 239
  Tipos, 247
Spline A, 247
Spline B, 248
Spline C, 249
SPLINEPATH, 252
SPN, 691
SPOF, 685
SPOS, 563
SPP, 691
SQRT, 63
START, 109
STARTFIFO, 490
STAT, 395, 401
STOPFIFO, 490
STOPRE, 490
STOPREOF, 611
String
  Concatenación, 77
  Longitud, 79
  Operaciones, 74
STRING, 24

STRINGIS, 713
Direcciones CN, 717
STRINGVAR, 83
STRLEN, 79
Subprograma, 152
Llamada con transferencia de parámetros, 187
Llamada sin transferencia de parámetros, 185
Llamada, indirecta, 193
Llamada, modal, 191
Nombre, 153
Repetición, 189
Ruta de búsqueda programable, 198
Salto atrás parametrizable, 178
SUBSTR, 82
Suceso de conmutación
en la medición, 276
SYNFCT, 616
SYNR, 24
SYNRW, 24
SYNW, 24

T

TAN, 63
TANG, 477
TANGDEL, 477
TANGOF, 477
TANGON, 477
TCARR, 464
TCOABS, 464
TCOFR, 464
TCOFRX, 464
TCOFRY, 464
TCOFRZ, 464
THETA, 359, 360
Tiempo de ejecución
Comportamiento de estructuras de control, 101
Tiempo de mecanizado, 726
Tiempo necesario
Acciones síncronas, 651
Tiempo residual
de una pieza, 727
TILT, 337
Tipo de cinemática, 463
Tipo de cinemática M, 463
Tipo de cinemática P, 463
Tipo de cinemática T, 463
Tipos de transformada
Función general, 321
TLIFT, 477
TMOF, 699
TMON, 699
TOFFOF, 625
TOFFOF, 468
TOFFON, 625
TOFFON, 468
TOLOWER, 78
Tope fijo, 647
Torsión, 722
TOUPPER, 78
TOWBCS, 423
TOWKCS, 423
TOWMCS, 423
TOWSTD, 423
TOWTCS, 423
TOWWCS, 423
TRAANG, 389
TRACON, 408
TRACYL, 379, 386
TRAFOOF, 407
TRAILOF, 521
TRAILON, 521
Transformada
Eje oblicuo, 389
Transformada cilíndrica curva, 379, 380
Corrección contorno-normal OFFN, 387
Transformada cinemática TRANSMIT, TRACYL y TRAANG, 326
Transformada con eje lineal basculable, 332
Transformada de 5 ejes
Programación de la orientación de herramienta con LEAD y TILT, 344
Programación del vector de dirección, 343
Transformada de envolvente de cilindro, 326
Transformada de orientación TRAORI
Cinemática de máquina, 325
Movimientos de desplazamiento y movimientos de orientación, 324
Programación de la orientación, 334
Transformada genérica de 5/6 ejes, 326
Variantes de la programación de la orientación, 335
Transformada polar, 326
Transformada TRACYL, 381
Transformada TRANSMIT, 377
Transformada, cinco ejes
Programación a través de LEAD/TILT, 337
Programación con ángulos de Euler, 341
Programación de la curvatura de la trayectoria en vectores normales a la superficie, 345
Programación en ángulos RPY, 342
Transformadas
Estado inicial de la orientación de herramienta independiente de la cinemática, 322
Transformada de orientación, 322
Transformadas cinemáticas, 323

Transformadas concatenadas, 323
Transformadas de tres y cuatro ejes, 333
Transformadas de tres, cuatro y cinco ejes TRAORI, 322
TRANSMIT, 376, 379, 401
TRAORI, 330, 333
Troquelado, 685, 691
TRUE, 24
TRUNC, 63
TU, 395, 401

U

U1,U2, 679
uc.com, ciclos de usuario, 205
ULI, 37
ULIMIT, 614
UNLOCK, 662
UNTIL, 107
UPATH, 264

V

V1,V2, 458
Vaivén
Asíncrono, 669
Controlar a través de acciones síncronas, 675
Penetración parcial, 679
Vaivén asíncrono, 669
Vaivén síncrono, 675
Vaivén asíncrono, 669
Vaivén síncrono
Acciones síncronas, 680
Asignación del eje de vaivén y del eje de penetración, 679
Definir penetraciones, 679
Evaluación, ciclo IPO, 682
Parada en el punto de inversión, 682
Penetración a profundidad de trabajo, 681
Penetración en la zona de inversión, 681
Siguiente penetración parcial, 683
Valor de ajuste, 417
Valor de desgaste, 417
Valor maestro
Acoplamiento, 641
VAR, 162
Variable
Conversión de tipos, 73
Variable de marca, 597
Variable de tipo temporizador, 601
Variable FIFO, 602
Variable frame
Asignaciones a los comandos G G54 a G599, 296
Decalajes de origen G54 a G599, 295
Variables
Conversión de tipos, 74
Definición, 24
definidas por el usuario, 24
Nombre, 26, 31
Tipo, 24
Variables del sistema, 591
Variables frame, 289
Asignar valores, 297
Definición de frames nuevos, 303
Llamada a transformadas de coordenadas, 289
Variable frame predefinida, 291, 301
Variables GUD
Compatibles con acciones síncronas, 594
VELOLIM, 513
Vista general
Frames activos en el canal, 315
Visualización de secuencia, 196
Suprimir, 172

W

WAIT, 109
WAITC, 563
WAITE, 109
WAITM, 109
WAITMC, 109
WHEN, 585
WHEN-DO, 680
WHENEVER, 585
WHENEVER-DO, 680
WHILE, 106
Winlimit, 629
WRITE, 136

X

xe, ye, ze, 356
XH YH ZH, 356
xi, yi, zi, 356
XOR, 66