SIEMENS

SINUMERIK

SINUMERIK 840D sl
NC 프로그래밍

프로그래밍 매뉴얼

소개 1

기본 안전 지침 2

기본 사항 3

작업 준비 4

테이블 5

부록 A

해당 제품

제어 시스템
SINUMERIK 840D sl / 840DE sl

CNC 소프트웨어 버전 4.95

07/2021
6FC5398-2EP40-0LA2

Промэлектроника, sales@prom-elec.com

법률상의 주의

경고사항

본 메뉴얼에는 여러분 자신의 안전과 재산 손실을 방지하기위해 여러분이 지켜야할 주의사항이 담겨있습니다. 여러분의 안전에 관련된 주의사항은 안전 경고 심볼로 강조되어있으며, 재산 손실에 관련된 주의사항은 안전 경고 심볼이 없습니다.

⚠ 위험
피하지 않으면 사망 또는 심각한 부상을 초래할 수 있는 절박한 위험 상황을 나타냅니다.

⚠ 경고
피하지 않으면 사망 또는 심각한 부상을 초래할 수 있는 잠재적인 위험 상황을 나타냅니다.

⚠ 주의
예방조치를 적절하게 취하지 않을 경우 경미한 인명 피해가 발생할 수 있음을 나타냅니다.

유의사항
예방조치를 적절하게 취하지 않을 경우 재산 피해가 발생할 수 있음을 나타냅니다.

여러 위험 수준이 적용될 때에는, 항상 가장 높은 레벨(낮은 번호)의 알림이 표시됩니다. 안전 경고 심볼이 인적 손실을 나타내는 경우, 재산 손실을 경고하는 또 다른 알림이 추가될 수도 있습니다.

자격을 가진 자

본서가 대상으로 하는 제품/시스템은 반드시 자격을 가진 자가 취급하는 것으로 하고, 각 조작 내용에 관련하는 문서,특히 안전상의 주의 및 경고가 준수되지 않으면 안됩니다. 자격을 가진 자란 훈련 내용 및 경험을 토대로 하면서 해당 제품/시스템의 취급에 동반하는 위험성을 인식하고, 발생할 수 있는 위해를 사전에 회피할 수 있는 자를 가리킵니다.

시멘스 제품의 올바른 사용을 위해

다음에 주의하십시오:

⚠ 경고
시멘스 제품은 카탈로그 및 부속의 기술 설명서의 지시에 따라 사용해 주십시오. 타사의 제품 또는 부품과 함께 사용하는 것은 당사의 권장 또는 허가가 있을 경우에 한합니다. 제품의 올바르고 안전한 사용을 위해 적절한 운반, 보관, 조립, 설치, 배선, 시동, 조작, 보수를 시행하고 있습니다. 사용할 때에는 허용된 범위를 꼭 지켜 주십시오. 부속의 기술 설명서에 기술되어있는 지시를 엄수해 주십시오.

상표

® 표시는 Siemens AG의 등록상표입니다. 본 문서의 기타 표시는 특정 목적으로 제삼자가 사용하는 경우, 지적 재산권을 해칠 수 있는 상표입니다.

책임의 포기

저희는 기술된 하드웨어와 소프트웨어가 본 메뉴얼의 내용물과 일치하는 것을 확인했습니다. 편차가 발생하는 것을 완전히 배제할 수는 없으므로, 완전히 동일하다고는 보장할 수 없습니다. 그렇지만, 메뉴얼의 데이터는 정기적으로 검토되며, 필요한 수정은 다음의 수정판에 반영됩니다. 품질 개선을 위한 의견은 환영합니다.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
독일

문서 주문번호: 6FC5398-2EP40-0LA2
Ⓟ 11/2021 변경될 수 있습니다

Copyright © Siemens AG 2019 - 2021.
모든 권리 보유

목차

1 소개 21

1.1 SINUMERIK 정보 21

1.2 문서 정보 21

1.3 인터넷에서 제공되는 문서 24

1.3.1 매뉴얼 개요 SINUMERIK 840D sl 24

1.3.2 문서 개요 SINUMERIK 조작반 구성요소 24

1.4 기술 문서에 대한 피드백 25

1.5 mySupport 문서 25

1.6 서비스 및 지원 26

1.7 중요 제품 정보 28

2 기본 안전 지침 29

2.1 일반 안전 지침 29

2.2 적용 예제의 보증 및 책임 29

2.3 보안 정보 29

3 기본 사항 31

3.1 기본적인 기하학 원리 31

3.1.1 공작물위치 31

3.1.1.1 포지셔닝의 참조 시스템 31

3.1.1.2 직교 좌표 31

3.1.1.3 극 좌표 34

3.1.1.4 절대 치수 35

3.1.1.5 증분 치수 37

3.1.2 작업 면 38

3.1.3 영점 및 원점 39

3.1.4 좌표계 41

3.1.4.1 기계 좌표계(MCS) 41

3.1.4.2 기본 좌표계 (BCS) 43

3.1.4.3 베이직 제로 시스템 (BZS) 45

3.1.4.4 셋터블 제로 시스템 (SZS) 45

3.1.4.5 공작물 좌표계 (WCS) 46

3.1.4.6 다양한 좌표계는 어떠한 관계인가? 47

3.2 NC 프로그램 작성의 기본 원칙 47

3.2.1 NC 프로그램의 이름 48

3.2.2 NC 프로그램의 구조와 내용 49

3.2.2.1 블록과 블록 구성 요소 49
3.2.2.2 블록 규칙 52
3.2.2.3 값 지정 53
3.2.2.4 주석 54
3.2.2.5 블럭 건너뛰기 54

3.3 NC 프로그램 생성 56
3.3.1 기본 절차 56
3.3.2 유효한 문자 57
3.3.3 프로그램 헤더 59
3.3.4 프로그램 예 60
3.3.4.1 예 1: 첫번째 프로그램 작성 단계 60
3.3.4.2 예 2: 선삭의 NC 프로그램 61
3.3.4.3 예 3: 밀링의 NC 프로그램 63

3.4 공구 교환 65
3.4.1 공구 관리가 활성화된 상태의 공구 교환 67
3.4.1.1 T=위치 기능을 사용한 공구 관리가 활성화된 상태의 공구 교환 67
3.4.1.2 M6을 사용한 공구 관리가 활성화된 상태의 공구 교환 69
3.4.2 공구 관리가 비활성화된 상태의 공구 교환 71
3.4.2.1 T 번호를 사용한 공구 관리가 비활성화된 상태의 공구 교환 71
3.4.2.2 M6을 사용한 공구 관리가 비활성화된 상태의 공구 교환 72
3.4.3 손상된 T 프로그래밍 동작 74

3.5 공구 옵셋 74
3.5.1 프로그래밍된 형상 및 공구 경로 74
3.5.2 공구 길이 보정 75
3.5.3 공구 반경 보정 76
3.5.4 공구 보정 메모리 76
3.5.5 공구 종류 78
3.5.5.1 공구 유형 번호 및 공구 그룹 78
3.5.5.2 밀링 공구 78
3.5.5.3 드릴 81
3.5.5.4 연마 공구 82
3.5.5.5 선삭 공구 84
3.5.5.6 특수 공구 86
3.5.6 공구 옵셋 활성화/비활성화(D, D0): 88
3.5.7 공구 옵셋 억제(SUPD) 91
3.5.8 프로그래밍이 가능한 공구 옵셋(TOFFL, TOFF, TOFFR, TOFFLR): 92

3.6 스핀들 운동 99
3.6.1 스핀들 속도 (S), 스핀들 회전 방향 (M3, M4, M5) 99
3.6.2 공구 절삭 속도(SVC) 103
3.6.3 주속 일정 제어 (G96/G961/G962, G97/G971/G972, G973, LIMS, SCC) 110
3.6.4 연삭 휠 원주 속도 일정 제어 (GWPSON, GWPSOF) 켜기/끄기: 115
3.6.5 프로그램 가능한 스핀들 속도 제한 (G25, G26) 116

3.7 속도 (피드) 제어 117

3.7.1 이송 속도 (G93, G94, G95, F, FGROUP, FL, FGREF) 117
3.7.2 포지셔닝 축 이송 (POS, POSA, POSP, FA, WAITP, WAITMC) 126
3.7.3 위치 제어 스핀들 모드 (SPCON, SPCOF) 130
3.7.4 스핀들 포지셔닝 (SPOS, SPOSA, M19, M70, WAITS) 131
3.7.5 포지셔닝 축/스핀들의 이송 속도 (FA, FPR, FPRAON, FPRAOF) 137
3.7.6 프로그램 가능한 이송 속도 오버라이드 (OVR, OVRRAP, OVRA) 141
3.7.7 프로그래밍이 가능한 가속 보정(ACC) 142
3.7.8 핸드휠 오버라이드의 주입속도(FD, FDA) 144
3.7.9 곡선 부분의 주입속도 최적화 (CFTCP, CFC, CFIN) 148
3.7.10 한 블록 내의 몇 가지 주입속도 (F, ST, SR, FMA, STA, SRA) 150
3.7.11 넌모달 이송 속도 (FB) 154
3.7.12 절삭날 이송 속도 (G95 FZ) 155

3.8 형상 설정 160
3.8.1 셋터블 워크 옵셋 (G54~G57, G505~G599, G53, G500, SUPA, G153) 160
3.8.2 좌표 설정 (G54~G57, G505~G599, G53, G500, SUPA, G153): 추가 정보 162
3.8.3 작업 평면 선택 (G17/G18/G19) 164
3.8.4 치수 166
3.8.4.1 절대치(G90, AC) 167
3.8.4.2 증분 치수(G91, IC) 169
3.8.4.3 선삭과 밀링의 절대와 증분 치수(G90/G91) 173
3.8.4.4 로터리 축의 절대 치수 (DC, ACP, ACN) 174
3.8.4.5 미터식/인치식 치수 시스템(G70/G71, G700/G710) 176
3.8.4.6 채널 관련 직경/반경 (DIAMON, DIAM90, DIAMOF, DIAMCYCOF) 프로그래밍 180
3.8.4.7 축 관련 직경/반경 프로그래밍 (DIAMONA, DIAM90A, DIAMOFA, DIACYCOFA, DIAMCHANA, DIAMCHAN, DAC, DIC, RAC, RIC) 183
3.8.5 선삭용 공작물 위치 187

3.9 이송 명령 189
3.9.1 이동 명령어의 일반 정보 189
3.9.2 직교 좌표의 이동 명령어(G0, G1, G2, G3, X..., Y..., Z...) 191
3.9.3 극 좌표의 이송 명령어 192
3.9.3.1 극 좌표의 기준점(G110, G111, G112) 192
3.9.3.2 극 좌표로의 이동 명령어(G0, G1, G2, G3, AP, RP) 194
3.9.4 급 이송 운동 198
3.9.4.1 급 이송 (G0) 활성화 198
3.9.4.2 급 이송 동작을 위한 선형 보간 켜기/끄기(RTLION, RTLIOF) 200
3.9.4.3 급이송 모션을 위한 허용 공차 조정(STOLF, CTOLG0, OTOLG0) 202
3.9.5 선형 보간(G1) 206
3.9.6 원호 보간 209
3.9.6.1 개요 209
3.9.6.2 중심점과 끝점의 원 보간(G2/G3, X... Y... Z..., I... J... K...) 210
3.9.6.3 반경과 종점의 원호 보간 (G2/G3, X... Y... Z..., CR) 212
3.9.6.4 틈 각도 및 종점 / 중간점을 가진 원호 보간(G2/G3, X... Y... Z... / I... J... K..., AR) 214
3.9.6.5 극 좌표로의 원 보간(G2/G3, AP, RP) 217
3.9.6.6 중간점과 끝점의 원 보간(CIP, X... Y... Z..., I1... J1... K1...) 219

3.9.6.7 접선 전이로의 원 보간 222
3.9.7 헬리컬 보간 (G2/G3, TURN) 226
3.9.8 내향 나선 보간(INVCW, INVCCW) 229
3.9.9 형상 정의 234
3.9.9.1 형상 정의 프로그래밍 234
3.9.9.2 형상 정의 직선 1개 235
3.9.9.3 형상 정의 두 개의 직선 237
3.9.9.4 형상 정의 직선 3개 241
3.9.9.5 형상 정의 각도로의 끝점 프로그래밍 244
3.9.10 나사 절삭 245
3.9.10.1 연속 리드를 사용하여 나사 절삭(G33, SF) 245
3.9.10.2 리드 증가 또는 감소의 나사 절삭(G34, G35) 252
3.9.10.3 G33, G34 및 G35의 프로그램된 런인 및 런아웃 경로(DITS, DITE): 253
3.9.10.4 나사 절삭 중 고속 후퇴 (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN) 256
3.9.10.5 볼록 나사 (G335, G336) 260
3.9.11 보정 척을 사용하지 않는 태핑 265
3.9.11.1 보정 척 및 후퇴 모션을 사용하지 않는 태핑(G331, G332) 265
3.9.11.2 예: G331 / G332를 사용한 태핑 267
3.9.11.3 예: 현 기어 단계의 프로그램된 천공 속도 출력 267
3.9.11.4 예: 두 번째 기어단계 데이터 블록 응용 사례 268
3.9.11.5 예: 속도는 프로그램 되지 않았고, 기어박스 단계는 모니터링 됩니다. 268
3.9.11.6 예: 기어박스 단계는 변경할 수 없고, 기어박스 단계 모니터링 됩니다. 268
3.9.11.7 예: SPOS를 사용하지 않는 프로그래밍 269
3.9.12 태핑 척을 사용한 태핑 270
3.9.12.1 보정 척 및 후퇴 모션을 갖는 태핑(G63) 270
3.9.13 모따기, 라운딩(CHF, CHR, RND, RNDM, FRC, FRCM) 271

3.10 공구 반경 보정 277
3.10.1 공구 반경 보정 (G40, G41, G42, OFFN) 277
3.10.2 형상 접근 및 후퇴 (NORM, KONT, KONTC, KONTT) 288
3.10.3 외부 코너(G450, G451, DISC)에서 보정 295
3.10.4 유연한 접근과 후퇴 298
3.10.4.1 접근 및 후퇴 (G140~G143, G147, G148, G247, G248, G347, G348, G340, G341, DISR, DISCL, DISRP, FAD, PM, PR) 298
3.10.4.2 연장 후퇴 방식을 이용한 접근과 후퇴 (G460, G461, G462) 311
3.10.5 충돌 탐지("병목 탐지") 활성화/비활성화(CDON, CDOF, CDOF2) 315
3.10.6 2 1/2 D 공구 옵셋 (CUT2D, CUT2DD, CUT2DF, CUT2DFD) 317
3.10.7 일정한 공구 반경 보정 유지(CUTCONON, CUTCONOF) 320
3.10.8 적절한 절삭날 위치의 공구 322

3.11 경로 동작 324
3.11.1 정위치 정지 (G60, G9, G601, G602, G603) 324
3.11.2 연속 경로 모드 (G64, G641, G642, G643, G644, G645, ADIS, ADISPOS) 327

3.12 좌표 변형(프레임) 337

3.12.1 프레임 337
3.12.2 프레임 지침 339
3.12.3 프로그래밍 워크 옵셋 (TRANS, ATRANS) 342
3.12.4 프로그래밍 워크 옵셋 (G58, G59) 346
3.12.5 프로그램 가능한 회전 (ROT, AROT, RPL) 348
3.12.6 입체각으로 프로그래밍 프레임 회전(ROTS, AROTS, CROTS) 355
3.12.7 프로그래밍 배율 계수 (SCALE, ASCALE) 360
3.12.8 프로그램 가능한 미러링 (MIRROR, AMIRROR) 363
3.12.9 공구 오리엔테이션에 따른 프레임 생성 (TOFRAME, TOROT, PAROT): 368
3.12.10 프레임 (G53, G153, SUPA, G500) 선택 해제 371
3.12.11 프로그래밍: Deselecting overlays axis-specifically (CORROF) 372
3.12.12 추가 워크 옵셋 선택해제(DRFROF) 375
3.12.13 특정 연삭 워크 옵셋 (GFRAME0, GFRAME1 ... GFRAME100) 377

3.13 보조 기능 출력 378
3.13.1 M 코드 381

3.14 보조 명령어 384
3.14.1 메시지 출력 (MSG) 384
3.14.2 OPI 변수 (WRTPR) 에 문자열 쓰기 386
3.14.3 작업 영역 제한 388
3.14.3.1 BCS (G25/G26, WALIMON, WALIMOF)에서의 작업 영역 제한 388
3.14.3.2 WCS/SZS (WALCS0 ... WALCS10)에서의 작업 영역 제한 391
3.14.4 원점 복귀(G74) 395
3.14.5 고정 지점 접근(G75) 396
3.14.6 고정 정지점으로 이동(FXS, FXST, FXSW) 400
3.14.7 드웰 시간 (G4) 405
3.14.8 내부 전처리 정지 407

3.15 기타 정보 408
3.15.1 축 408
3.15.1.1 메인 축/기하 축 409
3.15.1.2 특수 축 410
3.15.1.3 메인 스핀들, 마스터 스핀들 411
3.15.1.4 기계 축 411
3.15.1.5 채널 축 412
3.15.1.6 경로 축 412
3.15.1.7 위치지정 축 412
3.15.1.8 동기화 축 413
3.15.1.9 명령 축 414
3.15.1.10 PLC 축 414
3.15.1.11 링크 축 414
3.15.1.12 선도 링크 축 416
3.15.2 이송 명령부터 기계 이동까지 418
3.15.3 경로 계산 418
3.15.4 어드레스 420

3.15.5 이름 422
3.15.6 상수 423
3.15.7 연산자 및 산술 함수 425

4 작업 준비 429

4.1 플렉시블 NC 프로그래밍 429
4.1.1 변수 429
4.1.1.1 시스템 데이터 430
4.1.1.2 사전 정의된 사용자 변수: 채널별 산술 파라미터 (R) 433
4.1.1.3 사전 정의된 사용자 변수: 전역 산술 파라미터 (RG) 435
4.1.1.4 사전 정의된 사용자 변수: 링크 변수 436
4.1.1.5 사용자 변수 정의 (DEF) 439
4.1.1.6 시스템 데이터, 사용자 데이터 및 NC 명령어 (REDEF)의 재정의 446
4.1.1.7 속성: 초기화 값 450
4.1.1.8 속성: 한계값 (LLI, ULI) 453
4.1.1.9 속성: 물리 단위 (PHU) 455
4.1.1.10 속성: 사용 권한 (APR, APW, APRP, APWP, APRB, APWB) 457
4.1.1.11 정의 가능 속성 및 재정의 가능 속성 개요 462
4.1.1.12 배열 변수 정의 및 초기화 (DEF, SET, REP) 464
4.1.1.13 데이터 유형 470
4.1.1.14 변수 최소, 최대 및 범위 (MINVAL, MAXVAL 및 BOUND) 471
4.1.1.15 변수 가용성 검사 (ISVAR) 472
4.1.1.16 속성 값/데이터 유형 읽기(GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP) 474
4.1.1.17 가능한 유형 변환 481
4.1.2 간접 프로그래밍 482
4.1.2.1 어드레스 간접 프로그래밍 482
4.1.2.2 G 코드 간접 프로그래밍 484
4.1.2.3 위치 속성 간접 프로그래밍 (GP) 485
4.1.2.4 가공 프로그램 라인 간접 프로그래밍(EXECSTRING) 488
4.1.3 명령 489
4.1.3.1 산술 함수 489
4.1.3.2 비교 및 논리 연산자 492
4.1.3.3 연산 우선 순위 494
4.1.3.4 비교 오류에 대한 정밀도 수정(TRUNC) 494
4.1.3.5 반올림 (ROUNDUP) 496
4.1.4 문자열 연산 497
4.1.4.1 STRING으로 유형 변환 (AXSTRING) 498
4.1.4.2 STRING 유형 변환 (NUMBER, ISNUMBER, AXNAME) 499
4.1.4.3 문자열 연결 (<<) 500
4.1.4.4 소문자/대문자로 변환 (TOLOWER, TOUPPER) 502
4.1.4.5 문자열 길이 결정 (STRLEN) 502
4.1.4.6 문자열에서 문자/문자열 검색 (INDEX, RINDEX, MINDEX, MATCH) 503
4.1.4.7 하위 문자열 선택 (SUBSTR) 504
4.1.4.8 개별 문자 읽기/쓰기 505

4.1.4.9 문자열 포맷 (SPRINT) 507
4.1.5 프로그램 점프 및 분기 517
4.1.5.1 프로그램 시작으로 점프 (GOTOS) 517
4.1.5.2 점프 마커로 프로그램 점프 (GOTOB, GOTOF, GOTO, GOTOC) 519
4.1.5.3 프로그램 분기 (CASE ... OF ... DEFAULT ...) 522
4.1.6 프로그램 섹션 반복 (REPEAT, REPEATB, ENDLABEL, P) 524
4.1.7 점검 구조 531
4.1.7.1 조건문 및 분기 (IF, ELSE, ENDIF) 534
4.1.7.2 연속 프로그램 루프 (LOOP, ENDLOOP) 535
4.1.7.3 카운트 루프 (FOR ... TO ..., ENDFOR) 536
4.1.7.4 루프 시작 시 조건을 가진 프로그램 루프 (WHILE, ENDWHILE) 538
4.1.7.5 루프 끝에서 조건을 가진 프로그램 루프 (REPEAT, UNTIL) 538
4.1.7.6 중첩된 점검 구조를 가진 프로그램 예 539
4.1.8 채널 교차 프로그램 코디네이션(INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) 540
4.1.9 매크로 방법 (DEFINE ... AS) 546

4.2 서브프로그램 방법 549
4.2.1 일반 정보 549
4.2.1.1 서브프로그램 549
4.2.1.2 서브프로그램 이름 551
4.2.1.3 서브프로그램 중첩 552
4.2.1.4 검색 경로 554
4.2.1.5 유형 파라미터 및 실제 파라미터 554
4.2.1.6 파라미터 전송 555
4.2.2 서브프로그램 정의 556
4.2.2.1 파라미터 전송이 없는 서브프로그램 556
4.2.2.2 값별 호출 파라미터 전송을 포함한 서브프로그램 (PROC) 557
4.2.2.3 기준별 호출 파라미터 전송을 포함한 서브프로그램 (PROC, VAR) 559
4.2.2.4 모달 G 코드 저장 (SAVE) 562
4.2.2.5 싱글 블록 실행 억제 (SBLOF, SBLON) 563
4.2.2.6 현재 블록 디스플레이 억제 (DISPLOF, DISPLON, ACTBLOCNO) 568
4.2.2.7 준비 과정에서 서브프로그램 식별 (PREPRO) 572
4.2.2.8 서브프로그램 복귀 M17 572
4.2.2.9 RET 서브프로그램 복귀 574
4.2.2.10 파라미터 지정 가능한 서브프로그램 리턴 점프 (RET ...) 575
4.2.2.11 파라미터 지정이 가능한 서브프로그램 리턴 점프 (RETB ...) 582
4.2.3 서브프로그램 호출 587
4.2.3.1 파라미터 전송 없이 서브프로그램 호출 587
4.2.3.2 파라미터 전송을 포함한 서브프로그램 호출 (EXTERN) 590
4.2.3.3 프로그램 반복 횟수 (P) 592
4.2.3.4 모달 서브프로그램 호출 (MCALL) 594
4.2.3.5 간접 서브프로그램 호출 (CALL) 596
4.2.3.6 호출 프로그램 일부를 지정하여 서브프로그램 간접 호출 (CALL BLOCK ... TO ...) 597
4.2.3.7 ISO 언어로 프로그래밍된 프로그램의 간접 호출 (ISOCALL) 598

4.2.3.8 경로 지정 및 파라미터를 통한 서브프로그램 호출 (PCALL) 599
4.2.3.9 서브프로그램 호출에 대한 확장 검색 경로 (CALLPATH) 600
4.2.3.10 외부 서브루틴 실행(EXTCALL) 602

4.3 인터럽트 루틴 (ASUB) 606
4.3.1 인터럽트 루틴 기능 606
4.3.2 인터럽트 루틴 생성 607
4.3.3 인터럽트 루틴 지정 및 시작 (SETINT, PRIO, BLSYNC) 608
4.3.4 인터럽트 루틴 할당 비활성화/다시 활성화 (DISABLE, ENABLE) 611
4.3.5 인터럽트 루틴 할당 삭제 (CLRINT) 611
4.3.6 형상으로부터 빠른 후퇴 (SETINT LIFTFAST, ALF) 612
4.3.7 형상에서 고속 후퇴 시 이송 방향 615
4.3.8 인터럽트 루틴의 이동 순서 618

4.4 파일 및 프로그램 관리 619
4.4.1 프로그램 메모리 619
4.4.1.1 NCK 내 프로그램 메모리 619
4.4.1.2 외부 프로그램 메모리 622
4.4.1.3 프로그램 메모리 파일 주소 지정 623
4.4.1.4 서브프로그램 호출을 위한 경로 검색 628
4.4.1.5 경로와 파일 이름 조사 630
4.4.2 작업 메모리 (CHANDATA, COMPLETE, INITIAL) 632

4.5 파일 처리 635
4.5.1 파일 쓰기 (WRITE) 635
4.5.2 파일 삭제 (DELETE) 639
4.5.3 파일의 행 읽기 (READ) 640
4.5.4 파일의 존재 여부 점검 (ISFILE) 642
4.5.5 파일 정보 읽기 (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) 644

4.6 보호 영역 646
4.6.1 보호 영역 정의(CPROTDEF, NPROTDEF) 646
4.6.2 보호 영역 활성화/비활성화 (CPROT, NPROT) 650
4.6.3 보호 영역 위반, 작업 영역 제한 및 소프트웨어 리미트 스위치 점검 (CALCPOSI) 654

4.7 특수 이동 명령 666
4.7.1 코드 위치 복귀 (CAC, CIC, CDC, CACP, CACN) 666
4.7.2 스플라인 보간 (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) 667
4.7.3 스플라인 그룹 (SPLINEPATH) 679
4.7.4 NC 블록 압축 (COMPON, COMPCURV, COMPCAD, COMPOF) 활성화/비활성화 680
4.7.5 폴리노미얼 보간 (POLY, POLYPATH, PO, PL) 681
4.7.6 설정 가능한 경로 기준 (SPATH, UPATH) 687
4.7.7 채널별 측정 (MEAS, MEAW) 689
4.7.8 축별 측정(MEASA, MEAWA, MEAC)(옵션) 692
4.7.9 OEM 사용자를 위한 특수 기능 (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829) ... 702
4.7.10 코너 감속을 통한 이송 속도 감소 (FENDNORM, G62, G621) 703
4.7.11 프로그램 가능한 이동 종료 기준 (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) 704

4.8 좌표 변환 (프레임) 707
4.8.1 프레임 변수를 통한 좌표 변환 707
4.8.1.1 사전 정의된 프레임 변수 ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME) 709
4.8.2 프레임에 값 지정 713
4.8.2.1 직접 값 할당(축 값, 각도, 배율) 713
4.8.2.2 프레임 콤포넌트 읽기 및 변경 (TR, FI, RT, SC, MI) 715
4.8.2.3 프레임 계산 717
4.8.2.4 프레임 변수 정의 (DEF FRAME) 718
4.8.3 일반 옵셋 및 미세 옵셋 (CTRANS, CFINE) 719
4.8.4 외부 워크 옵셋 ($AA_ETRANS) 720
4.8.5 원점 복귀 상태를 삭제하고 실제 값 설정 (PRESETON) 722
4.8.6 원점 복귀 상태를 유지하면서 실제 값 설정 (PRESETONS) 723
4.8.7 공간의 세 개 측정 지점에서 프레임 계산 (MEAFRAME) 725
4.8.8 글로벌 프레임 728
4.8.8.1 채널 관련 프레임 ($P_CHBFR, $P_UBFR) 729
4.8.8.2 채널에서 프레임 활성화 730

4.9 변환 735
4.9.1 일반적인 변환 유형 프로그래밍 735
4.9.1.1 변환을 위한 방향 이동 738
4.9.1.2 방향 변환 개요 TRAORI 743
4.9.2 3축, 4축 및 5축 변환 (TRAORI) 746
4.9.2.1 유니버셜 공구 헤드의 일반적인 관계 746
4.9.2.2 3축, 4축 및 5축 변환 (TRAORI) 749
4.9.2.3 오리엔테이션 프로그래밍 변수 및 초기 설정 (ORIRESET) 750
4.9.2.4 공구 오리엔테이션 프로그래밍 (A..., B..., C..., LEAD, TILT) 752
4.9.2.5 페이스 밀링 (A4, B4, C4, A5, B5, C5) 758
4.9.2.6 오리엔테이션 축의 기준 (ORIWKS, ORIMKS): 759
4.9.2.7 오리엔테이션 축 프로그래밍 (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) 761
4.9.2.8 테이퍼의 주변 표면을 따라 오리엔테이션 프로그래밍 (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) 764
4.9.2.9 두 개의 접점 방향 지정 (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=) 767
4.9.3 방향 폴리노미얼 (PO[angle], PO[coordinate]) 769
4.9.4 공구 오리엔테이션 회전 (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) 771
4.9.5 경로 기준 방향 774
4.9.5.1 경로 기준 방향 유형 774
4.9.5.2 경로 기준 공구 오리엔테이션 회전 (ORIPATH, ORIPATHS, 회전 각도) 775
4.9.5.3 경로 기준으로 공구 회전 보간 (ORIROTC, THETA) 777
4.9.5.4 방향 특성 유연화 (ORIPATHS A8=, B8=, C8=) 779
4.9.6 방향 압축 (COMPON, COMPCURV, COMPCAD, COMPSURF) 780
4.9.7 오리엔테이션 특성 활성화/비활성화(ORISON, ORISOF) 783
4.9.8 운동학적 변형 784
4.9.8.1 단면 끝부분 변환 활성화 (TRANSMIT) 784
4.9.8.2 실린더 표면 변환을 활성화합니다. (TRACYL) 785

4.9.8.3 각도 프로그램이 가능한 사각 변환(TRAANG) 활성화 (TRAANG) 788
4.9.8.4 연마 기계에 대한 사각 플런지 절삭 (G5, G7) 790
4.9.9 연쇄 변환 (TRACON)을 활성화합니다. 792
4.9.10 직교 PTP 이송 793
4.9.10.1 직교 PTP 이송 (PTP, PTPG0, PTPWOC, CP)를 활성화/비활성화합니다. 793
4.9.10.2 조인트의 위치를 명시합니다 (STAT). 795
4.9.10.3 축 각도 (TU) 표시를 명시하십시오. 800
4.9.10.4 예 1: ROBX 변환을 갖는 6-축 로봇의 PTP 이송 803
4.9.10.5 예 2: 일반 5축 변환을 위한 PTP 이송 804
4.9.10.6 예 3: PTPG0 및 TRANSMIT 805
4.9.11 변환 선택 시 제한사항 807
4.9.12 변환 선택 해제 (TRAFOOF) 808

4.10 키네마틱 체인 808
4.10.1 콤포넌트 삭제 (DELOBJ) 808
4.10.2 이름을 이용한 인덱스 결정 (NAMETOINT) 813

4.11 키네마틱 체인의 충돌 방지 814
4.11.1 충돌 쌍 점검 (COLLPAIR) 815
4.11.2 충돌 방지용 기계 모델의 요청 재계산(PROTA) 815
4.11.3 보호 영역 상태 설정 (PROTS) 816
4.11.4 두 보호 영역의 간격 결정 (PROTD) 817

4.12 운동학적 체인을 이용한 변환 819
4.12.1 변환 활성화(TRAFOON) 819
4.12.2 기계 측정(CORRTRAFO) 후 오리엔테이션 변환 수정 820

4.13 공구 옵셋 828
4.13.1 옵셋 메모리 828
4.13.2 추가 옵셋 831
4.13.2.1 추가 옵셋 선택(DL) 831
4.13.2.2 마모값 및 설정값을 지정($TC_SCPxy[t,d], $TC_ECPxy[t,d]) 832
4.13.2.3 추가 옵셋 삭제(DELDL) 832
4.13.3 공구 옵셋 특수 처리 833
4.13.3.1 공구 길이 미러링 835
4.13.3.2 마모 기호 평가 836
4.13.3.3 활성화된 가공 작업의 좌표계(TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) 837
4.13.3.4 공구 길이 및 평면 변경 839
4.13.4 온라인 공구 옵셋 841
4.13.4.1 폴리노미널 기능 정의 (FCTDEF) 841
4.13.4.2 온라인 공구 옵셋 연속 작성 (PUTFTOCF) 843
4.13.4.3 온라인 공구 옵셋 작성, 불연속 (PUTFTOC) 844
4.13.4.4 온라인 공구 옵셋 활성화/비활성화 (FTOCON/FTOCOF) 844
4.13.5 3D 공구 반경 보정 845
4.13.5.1 3D 원주 밀링을 위한 3D 공구 반경 보정 선택(CUT3DC, CUT3DCD, ISD) 845
4.13.5.2 3D 페이스 밀링용 3D 공구 반경 보정 선택(CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD) 850

4.13.5.3 한계 표면을 고려한 3D 원주 밀링(CUT3DCC, CUT3DCCD) 856
4.13.6 공구 오리엔테이션 (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) 861
4.13.7 D 번호, 절삭날 번호의 자유 할당 867
4.13.7.1 D 번호, 절삭날 번호의 자유 할당 (CE 어드레스) 867
4.13.7.2 D 번호의 자유 할당: D 번호 확인(CHKDNO) 867
4.13.7.3 D 번호의 자유 할당: D 번호 이름 변경(GETDNO, SETDNO) 868
4.13.7.4 D 번호의 자유 할당: 지정된 D 번호의 T 번호를 결정(GETACTTD) 869
4.13.7.5 D 번호의 자유 할당: 무효화된 D 번호(DZERO) 870
4.13.8 공구 홀더 좌표계 870
4.13.9 방향 조정 가능한 공구 홀더에 대한 공구 길이 보정 (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) 877
4.13.10 기계 측정(CORRTC)에 따라 오리엔테이션 설정이 가능한 공구 캐리어 수정 882
4.13.11 온라인 공구 길이 보정(TOFFON, TOFFOF) 886
4.13.12 회전 공구에 대한 옵셋 데이터 수정 889
4.13.12.1 오리엔테이션 계산(ORISOLH) 889
4.13.12.2 회전식 공구에 대한 옵셋 데이터 수정 활성화(CUTMOD, CUTMODK) 902
4.13.13 공구 환경에 대한 작업 911
4.13.13.1 공구 환경 저장(TOOLENV) 911
4.13.13.2 도구 환경 삭제(DELTOOLENV) 914
4.13.13.3 T, D 및 DL 번호 읽기(GETTENV) 915
4.13.13.4 저장된 공구 환경에 대한 정보 읽기($P_TOOLENVN, ($P_TOOLENV) 916
4.13.13.5 공구 길이 및/또는 공구 길이 컴포넌트 읽기(GETTCOR) 917
4.13.13.6 공구 컴포넌트 변경(SETTCOR) 925
4.13.14 총 길이 L1, L2, L3의 좌표 축으로의 지정을 읽습니다(LENTOAX). 940

4.14 경로 이송 동작 943
4.14.1 이송 속도 특성 (FNORM, FLIN, FCUB, FPO) 943
4.14.2 가속 특성 948
4.14.2.1 가속 모드 (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA) 948
4.14.2.2 다음 축에 대한 가속의 영향(VELOLIMA, ACCLIMA, JERKLIMA) 951
4.14.2.3 기술별 동적 값의 활성화(DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH, DYNPREC) 952
4.14.3 피드포워드 제어 이송 (FFWON, FFWOF) 954
4.14.4 프로그램 가능한 형상/오리엔테이션 공차 (CTOL, OTOL, ATOL) 955
4.14.5 프로그래밍 형상 정확도 켜기/끄기(CPRECON, CPRECOF) 960
4.14.6 자동 필터 전환 활성화/비활성화(AFISON, AFISOF) 962
4.14.7 전처리 메모리를 사용한 프로그램 순서 (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) 965
4.14.8 정지 지연 섹션 정의(DELAYFSTON, DELAYFSTOF) 968
4.14.9 SERUPRO을 위한 프로그램 위치 방지 (IPTRLOCK, IPTRUNLOCK) 970
4.14.10 형상 리포지셔닝 (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL) 972
4.14.11 모션 제어에 영향 981
4.14.11.1 최대 축 속도 또는 스핀들 속도 조정(VELOLIM) 981
4.14.11.2 최대 축 저크 조정(JERKLIM) 983
4.14.11.3 최대 경로 속도 조정(FLIM) 984

4.14.11.4 최대 경로 가속 조정(PACCLIM) 986
4.14.12 커플링이 활성인 경우 블록 변경 동작 (CPBC) 987

4.15 축 기능 988
4.15.1 축 교체, 스핀들 교체 (RELEASE, GET, GETD) 988
4.15.2 다른 채널로 축 이송 (AXTOCHAN) 993
4.15.3 축 기능(AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) 995
4.15.4 대체 가능한 기하 축(GEOAX) 997
4.15.5 축 컨테이너 (AXCTSWE, AXCTSWED, AXCTSWEC) 1002
4.15.6 유효 축 위치 대기 (WAITENC) 1004
4.15.7 프로그램 가능한 파라미터 세트 전환 (SCPARA) 1005
4.15.8 조정 활성화/비활성화(CADAPTON, CADAPTOF) 1006
4.15.9 FIR 저크 필터를 동적 모드로 조정(CALCFIR) 1009
4.15.10 가공 프로그램에서 드라이브 파라미터 읽기/쓰기(DRVPRD, DRVPWR) 1011

4.16 축 커플링 1014
4.16.1 커플 모션 (TRAILON, TRAILOF) 1014
4.16.2 커브 테이블 (CTAB) 1019
4.16.2.1 커브 테이블 정의 (CTABDEF, CATBEND) 1019
4.16.2.2 커브 테이블의 존재 검사 (CTABEXISTS) 1025
4.16.2.3 커브 테이블 삭제 (CTABDEL) 1026
4.16.2.4 삭제 및 덮어쓰기 방지를 위한 커브 테이블 잠금 (CTABLOCK, CTABUNLOCK) 1027
4.16.2.5 커브 테이블: 테이블 속성 결정 (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) ... 1029
4.16.2.6 커브 테이블 값 읽기 (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) 1030
4.16.2.7 커브 테이블: 리소스 사용 확인 (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) 1035
4.16.3 축 마스터 값 커플링 (LEADON, LEADOF) 1037
4.16.4 전자 기어 (EG) 1043
4.16.4.1 전자 기어 정의(EGDEF) 1044
4.16.4.2 전자 기어박스 활성화(EGON, EGONSYN, EGONSYNE) 1045
4.16.4.3 전자 기어박스 활성화(EGOFS, EGOFC) 1048
4.16.4.4 전자 기어 정의 삭제(EGDEL) 1049
4.16.4.5 회전 이송 속도(G95)/전자 기어(FPR) 1050
4.16.5 동기 스핀들 1050
4.16.5.1 동기 스핀들: 프로그래밍(COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) 1051
4.16.6 일반 커플링(CP...) 1062
4.16.7 접선 제어 1072
4.16.7.1 커플링 정의 (TANG) 1072
4.16.7.2 중간 블록 생성 활성화 (TLIFT) 1075
4.16.7.3 커플링 활성화 (TANGON) 1076
4.16.7.4 커플링 비활성화 (TANGOF) 1077
4.16.7.5 커플링 삭제 (TANGDEL) 1078
4.16.8 마스터/슬레이브 커플링 (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) 1079

4.17 동기 동작 1083
4.17.1 동기 동작 정의 1083

4.18 오실레이션 1084
4.18.1 비동기 오실레이션 (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) 1084
4.18.2 동기 동작에 의한 오실레이션 제어 (OSCILL) 1090

4.19 펀칭 및 니블링 1098
4.19.1 활성화/비활성화 1098
4.19.1.1 펀칭 및 니블링 활성화/비활성화(SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) 1098
4.19.2 자동 경로 분할 1103
4.19.2.1 경로 축을 대상으로 경로 분할 1106
4.19.2.2 단일 축을 대상으로 경로 분할 1107

4.20 연삭 1108
4.20.1 연삭 전용 공구 모니터링 활성화/비활성화 (TMON, TMOF) 1108

4.21 확장 정지 및 후퇴 (ESR) 1110
4.21.1 NC 제어 ESR 1111
4.21.1.1 NC 제어 후퇴 (POLF, POLFA, POLFMASK, POLFMLIN) 1111
4.21.1.2 NC 제어 정지 1115
4.21.2 드라이브 통합 ESR 1116
4.21.2.1 드라이브 통합 정지 (ESRS) 설정 1116
4.21.2.2 드라이브 통합 후퇴 (ESRS) 설정 1117

4.22 프로그램 실행 시간/ 가공 카운터 1119
4.22.1 프로그램 실행 시간 1119
4.22.2 소재 계수기 1123

4.23 프로그램 시뮬레이션 1124
4.23.1 기능 1124
4.23.2 컴파일 싸이클과 연계한 프로그램 시뮬레이션 1125

4.24 추가 기능 1126
4.24.1 가공 데이터 활성화 (NEWCONF) 1126
4.24.2 NC 언어(STRINGIS)의 범위 확인 1127
4.24.3 가공 프로그램(MMC)에서 창 양방향 호출 1131
4.24.4 공정 데이터 공유 - 외부 장치/파일로 출력 (EXTOPEN, WRITE, EXTCLOSE) 1136
4.24.5 알람 (SETAL) 1141
4.24.6 소재를 정의하십시오(공작물). 1142
4.24.7 언어 전환 모드(G290, G291) 1148

4.25 사용자 선삭 프로그램 1149
4.25.1 스톡 제거 기능 지원 1149
4.25.2 형상 테이블 생성(CONTPRON) 1150
4.25.3 코드화 형상 테이블의 생성(CONTDCON) 1157
4.25.4 형상 요소 2개 사이의 교차점을 결정(INTERSEC) 1160
4.25.5 테이블의 형상 요소를 블록별로 실행(EXECTAB) 1162

4.25.6 원 데이터 계산(CALCDAT) 1163
4.25.7 형상 준비 종료(EXECUTE) 1165

4.26 외부에서 싸이클 프로그래밍 1166
4.26.1 공정 싸이클 1166
4.26.1.1 소개 1166
4.26.1.2 특정 기술 기반 개요 1168
4.26.1.3 HOLES1 - 행 위치 패턴 1170
4.26.1.4 HOLES2 - 원호 또는 피치원 위치 패턴 1171
4.26.1.5 POCKET3 - 사각 포켓 1174
4.26.1.6 POCKET4 - 원호 포켓 1178
4.26.1.7 SLOT1- 일반 슬롯 1182
4.26.1.8 SLOT2 - 원호 슬롯 1186
4.26.1.9 LONGHOLE - 연장된 홀 1189
4.26.1.10 CYCLE60 - 조각 1192
4.26.1.11 CYCLE61 - 페이스 밀링 1196
4.26.1.12 CYCLE62 - 형상 호출 1198
4.26.1.13 CYCLE63 – 형상 포켓 밀링 / 형상 포켓 잔삭 가공 / 형상 스피곳 밀링 / 형상 스피곳 잔삭 가공 1199
4.26.1.14 CYCLE64 - 형상 포켓 예비 드릴링 1203
4.26.1.15 CYCLE70 - 나사 밀링 1206
4.26.1.16 CYCLE72 - 경로 밀링 1208
4.26.1.17 CYCLE76 - 사각 스피곳 1212
4.26.1.18 CYCLE77 - 원형 스피곳 1216
4.26.1.19 CYCLE78 - 드릴 나사 밀링 1218
4.26.1.20 CYCLE79 - 다각형 1222
4.26.1.21 CYCLE81 - 드릴링, 센터링 1225
4.26.1.22 CYCLE82 - 드릴링, 카운터보링 1227
4.26.1.23 CYCLE83 - 심공 드릴링 1 1230
4.26.1.24 CYCLE84 - 동기 태핑 1234
4.26.1.25 CYCLE85 - 리밍 1238
4.26.1.26 CYCLE86 - 보링 1240
4.26.1.27 CYCLE92 - 절단 1242
4.26.1.28 CYCLE95 - 형상 절삭 1244
4.26.1.29 CYCLE98 - 나사 체인 1246
4.26.1.30 CYCLE99 - 나사 선삭 1253
4.26.1.31 CYCLE435 - 드레서 좌표계 설정 1258
4.26.1.32 CYCLE495 - 형상 트루잉 1259
4.26.1.33 CYCLE782 - 부하에 맞게 조정 1261
4.26.1.34 CYCLE800 – 스위블 평면 / 스위블 공구 / 정렬 공구 1265
4.26.1.35 CYCLE801 – 격자 또는 프레임 위치 패턴 1270
4.26.1.36 CYCLE802 - 임의 위치 1272
4.26.1.37 CYCLE806 - 보간 선삭 1276
4.26.1.38 CYCLE830 - 심공 드릴링 2 1277
4.26.1.39 CYCLE832 - 고속 가공 설정 1285

4.26.1.40 CYCLE840 - 태핑 척을 사용한 태핑 1289
4.26.1.41 CYCLE899 - 개방형 슬롯 1293
4.26.1.42 CYCLE930 - 홈 1296
4.26.1.43 CYCLE940 – 언더컷 형상 E 및 F / 언더컷 나사 1299
4.26.1.44 CYCLE951 - 선삭 1303
4.26.1.45 CYCLE952 – 칩 제거 / 잔여 칩 제거 / 플런지 절삭 / 잔여 플런지 절삭 / 플런지 선삭 / 잔여 플런지 선삭 1306
4.26.1.46 CYCLE4071 - 반전 지점에서 절입을 통한 세로 방향 연삭 1314
4.26.1.47 CYCLE4072 - 반전 지점에서의 절입을 통한 세로 방향 연삭과 취소 신호 1316
4.26.1.48 CYCLE4073 - 연속 절입을 통한 세로 방향 연삭 1320
4.26.1.49 CYCLE4074 - 연속 절입을 통한 세로 방향 연삭 및 취소 신호 1322
4.26.1.50 CYCLE4075 - 반전 지점에서 절입을 통한 표면 연삭 1326
4.26.1.51 CYCLE4077 - 반전 지점에서의 절입을 통한 표면 연삭과 취소 신호 1329
4.26.1.52 CYCLE4078 - 연속 절입을 통한 표면 연삭 1333
4.26.1.53 CYCLE4079 - 간헐적 절입을 통한 표면 연삭 1335
4.26.1.54 GROUP_BEGIN - 프로그램 블록 시작 1338
4.26.1.55 GROUP_END - 프로그램 블록 종료 1339
4.26.1.56 GROUP_ADDEND - 시험 절삭 추가 종료 1339
4.26.1.57 보충 조건 1340
4.26.2 측정 싸이클 1342
4.26.2.1 측정 싸이클 파라미터 개요 1342

5 테이블 1403

5.1 작업 1403

5.2 어드레스 1450
5.2.1 어드레스 문자 1450
5.2.2 고정 어드레스 1452
5.2.3 셋터블 어드레스 1458

5.3 G 명령 1466
5.3.1 G 코드 1466
5.3.2 G 그룹 1: 동작 명령어로 유효한 모달 명령어 1466
5.3.3 G 그룹 2: 넌모달로 적용되는 동작, 드웰 시간 1467
5.3.4 G 그룹 3: 프로그래밍 프레임, 작업 구역 제한과 극점 프로그래밍 1468
5.3.5 G 그룹 4: FIFO 1469
5.3.6 G 그룹 6: 평면 선택 1469
5.3.7 G 그룹 7: 공구 반경 보정 1469
5.3.8 G 그룹 8: 설정 가능한 워크 옵셋 1470
5.3.9 G 그룹 9: 프레임 및 공구 옵셋 억제 1470
5.3.10 G 그룹 10: 정위치 정지 / 연속 경로 모드 1471
5.3.11 G 그룹 11: 정위치 정지, 넌모달 1471
5.3.12 G 그룹 12: 정위치 정지의 블록 변경 기준(G60/G9) 1471
5.3.13 G 그룹 13: 인치/미터 시스템 1472
5.3.14 G 그룹 14: 절대/증분 시스템 1472
5.3.15 G 그룹 15: 이송 형식 1472

5.3.16 G 그룹 16: 커브 내부 및 외부의 피드 오버라이드 1473
5.3.17 G 그룹 17: 공구 옵셋의 접근 및 후퇴 동작 1473
5.3.18 G 그룹 18: 공구 옵셋 코너 동작 1474
5.3.19 G 그룹 19: 스플라인 시작점에서의 커브 트렌지션 1474
5.3.20 G 그룹 20: 스플라인 끝에서 커브 트렌지션 1474
5.3.21 G 그룹 21: 가속 특성 1475
5.3.22 G 그룹 22: 공구 옵셋 유형 1475
5.3.23 G 그룹 23: 내부 형상에서 충돌 모니터링 1476
5.3.24 G 그룹 24: 사전 제어 1476
5.3.25 G 그룹 25: 공구 오리엔테이션 기준 1476
5.3.26 G 그룹 26: REPOS 리포지셔닝 모드(모달) 1477
5.3.27 G 그룹 27: 외부 코너에서 오리엔테이션 변경을 위한 공구 옵셋 1477
5.3.28 G 그룹 28: 작업 영역 제한 1477
5.3.29 G 그룹 29: 반경/직경 프로그래밍 1478
5.3.30 G 그룹 30: NC 블록 압축 1478
5.3.31 G 그룹 31: OEM G 명령 1478
5.3.32 G 그룹 32: OEM G 명령 1479
5.3.33 G 그룹 33: 공구 옵셋 미세 조정 옵셋 1480
5.3.34 G 그룹 34: 공구 오리엔테이션 스무딩 1480
5.3.35 G 그룹 35: 펀칭 및 니블링 1480
5.3.36 G 그룹 36: 지연 펀칭 1481
5.3.37 G 그룹 37: 이송 특성 1481
5.3.38 G 그룹 38: 펀치/니블링을 위한 고속 입력/출력 지정 1481
5.3.39 G 그룹 39: 프로그램 형상 정확도 1481
5.3.40 G 그룹 40: 공구 반경 보상 상수 1482
5.3.41 G 그룹 41: 나사 절삭 중단 1482
5.3.42 G 그룹 42: 공구 케리어 1482
5.3.43 G 그룹 43: SAR 접근 방향 1483
5.3.44 G 그룹 44: SAR 경로 분할 1483
5.3.45 G 그룹 45: FGROUP 축의 경로 기준 1483
5.3.46 G 그룹 46: 고속 후퇴를 위한 면 선택 1484
5.3.47 G 그룹 47: 외부 NC 코드 모드로의 전환 1484
5.3.48 G 그룹 48: 공구 반경 보정을 통한 접근 및 후퇴 반응 1484
5.3.49 G 그룹 49: 지점 간 모션 1485
5.3.50 G 그룹 50: 오리엔테이션 프로그래밍 1485
5.3.51 G 그룹 51: 오리엔테이션 프로그래밍을 위한 보간 유형 1486
5.3.52 G 그룹 52: 공작물에 대한 프레임 회전 1486
5.3.53 G 그룹 53: 공구에 대한 프레임 회전 1487
5.3.54 G 그룹 54: 폴리노미얼 프로그래밍을 위한 벡터 회전 1487
5.3.55 G 그룹 55: 직선 보간 유무의 급이송 1488
5.3.56 G 그룹 56: 공구 마모 고려 1488
5.3.57 G 그룹 57: 코너 감속 1488
5.3.58 G 그룹 59: 경로 보간을 위한 동적 응답 모드 1489
5.3.59 G 그룹 60: 작업 영역 제한 1489
5.3.60 G 그룹 61: 공구 오리엔테이션 스무딩 1490

5.3.61 G 그룹 62: REPOS 리포지셔닝 모드(넌모달) 1490
5.3.62 G 그룹 64: 연삭 프레임 1490
5.4 사전 정의된 절차 1491
5.5 동기 동작을 위해 사전 정의된 절차 1522
5.6 사전 정의된 기능 1524
5.7 현재 HMI에서 설정된 언어 1541
A 부록 1543
A.1 약어 목록 1543
인덱스 1551

소개 1

1.1 SINUMERIK 정보

단순하고 표준화된 CNC 장비에서 프리미엄 모듈형 장비 설계에 이르기까지 SINUMERIK CNC는 모든 장비 유형에 적합한 솔루션을 제공합니다. 개별 부품이든 대량 생산이든, 단순한 공작물이든 복잡한 공작물이든 상관없이 SINUMERIK은 프로토타입 제작과 공구 설계에서 금형 제작과 대규모 연속 생산에 이르는 모든 생산 영역에 고도로 동적인 자동화 솔루션을 제공합니다.

웹 사이트에서 SINUMERIK (https://www.siemens.com/sinumerik)에 대한 더욱 자세한 정보를 제공합니다.

1.2 문서 정보

이 문서는 SINUMERIK 프로그래밍 매뉴얼 문서 카테고리의 일부입니다.

"NC 프로그래밍" 프로그래밍 매뉴얼

"NC 프로그래밍" 프로그래밍 매뉴얼에는 SINUMERIK 컨트롤러의 NC 기능 프로그래밍에 관한 모든 정보가 포함되어 있습니다.

대상 그룹은 프로그래머와 설정 엔지니어입니다.

대상 그룹은 이 프로그래밍 매뉴얼을 참조해 프로그램과 소프트웨어 사용자 인터페이스를 개발, 작성, 테스트 및 디버깅할 수 있습니다.

내용 개요

제목 페이지의 내용 개요에 따르면 "NC 프로그래밍" 프로그래밍 매뉴얼은 두 개의 주요 장으로 나뉩니다.

1. **기본 사항**
   "기본 사항" 장은 드릴링, 밀링 및 선삭 작업에 대해 적합한 전문 지식을 갖춘 숙련된 조작자용으로 제작되었습니다. 또한 DIN 66025에 따라 정의된 명령 및 명령문을 설명하기 위해 간단한 프로그래밍 예가 사용되었습니다.
2. **작업 준비**
   "작업 준비" 장은 심층적이고 폭넓은 프로그래밍 지식을 갖춘 기술자용으로 제작되었습니다. SINUMERIK 제어 시스템에서 특수 프로그래밍 언어를 사용하여 자유 형상 표면, 채널 좌표계 등의 여러 공작물을 쉽게 프로그램할 수 있으며 또한 프로그래밍을 하기 위한 복잡한 조작을 단순화할 수 있습니다.

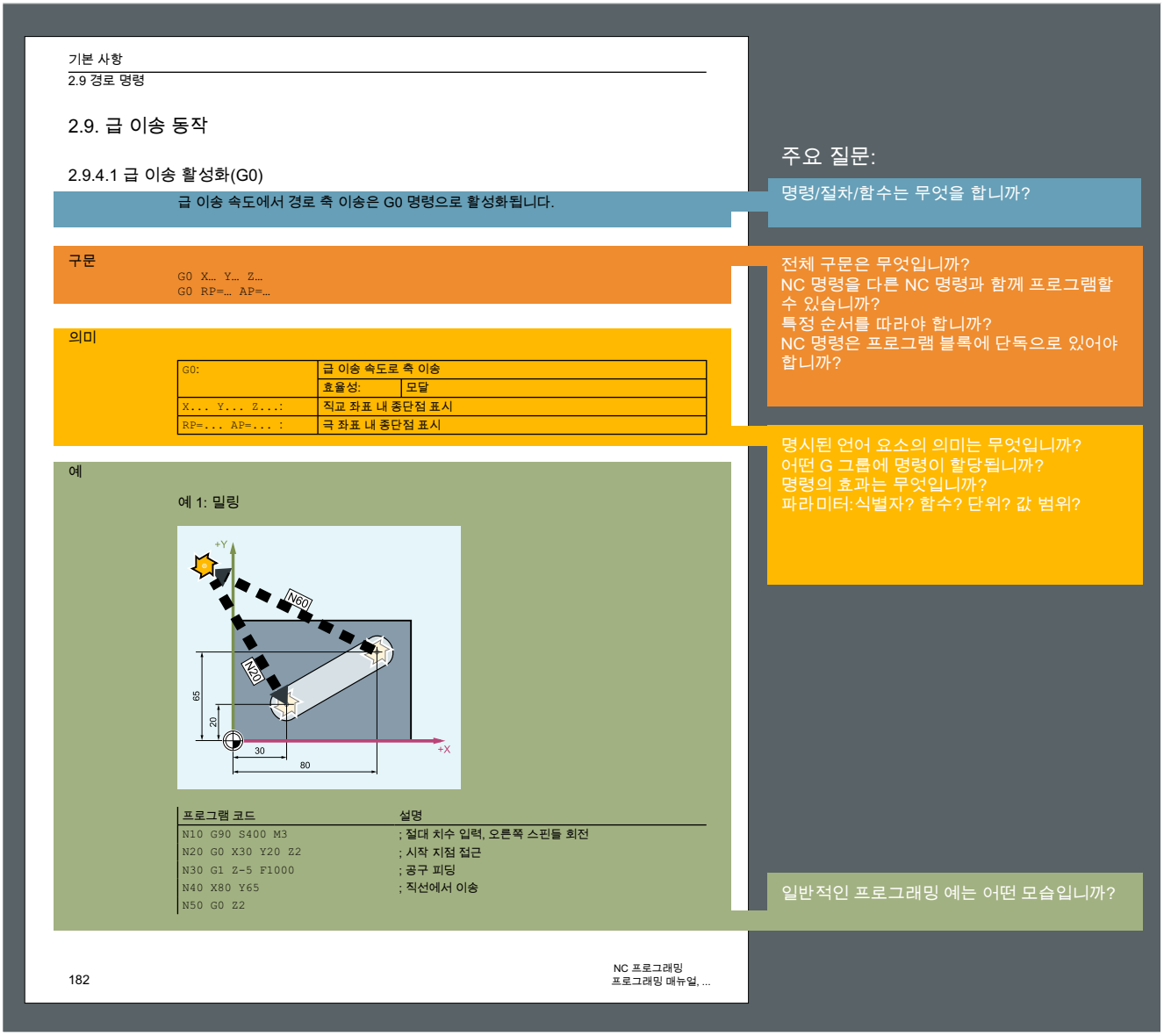
적용 대상

제목 페이지에는 이 프로그래밍 매뉴얼 버전이 어떤 SINUMERIK 제어 시스템과 어떤 소프트웨어 버전에 적용되는지 등 문서의 적용 대상에 대한 모든 정보도 포함됩니다.

장 구조

NC 언어 요소(G 명령, 절차, 함수 등)에 대한 설명은 통일된 방식으로 구성됩니다. 내용은 미리 정의된 주요 질문을 고려해 작성됩니다.

다음 그림은 예 장으로 이를 설명합니다.



예에 항상 나오는 구문, 의미 및 예 항목에 더해 다른 항목도 있을 수 있습니다.

추가 항목	사용
필수 조건	프로그래밍 함수의 적용을 위해 일부 필수 조건을 충족해야 하는 경우(예: 옵션의 라이선스).
유효성	프로그래밍 함수가 일부 조건에서만 유효한 경우(예: 특정 작동 모드에서만 유효).
제한 사항	프로그래밍 중에 다른 함수와의 상호작용을 고려해야 하는 경우.
추가 정보	함수의 프로그래밍을 이해하기 위해 추가 설명이 필요한 경우.

표준 범위

이 문서에서는 표준 버전의 기능에 대해서만 설명합니다. 실제로 제공되는 시스템의 기능 범위와 다를 수 있습니다. 제공된 드라이브 시스템의 기능만 주문 문서를 참조하십시오.

이 문서에서 설명하지 않는 시스템의 다른 기능을 실행할 수 있습니다. 하지만 이 조항이 새로운 시스템이나 서비스 수행 시 이러한 기능을 제공해야 한다는 것을 의미하지는 않습니다.

이 문서에 모든 제품 유형에 대한 상세 정보가 모두 포함되어 있지는 않습니다. 또한 이 문서는 가능한 모든 유형의 설치, 작동 및 서비스/유지보수 방법에 대해 설명하고 있지는 않습니다.

장비 제조업체는 제품의 추가 사항이나 변경 사항을 직접 문서화해야 합니다.

타사 웹 사이트

이 문서에 타사 웹 사이트의 하이퍼링크가 포함될 수 있습니다. Siemens는 타사 웹 사이트와 웹 사이트 콘텐츠에 대해 책임을 지지 않습니다. Siemens는 타사 웹 사이트에 표시되는 정보를 통제하지 않으며 제공된 콘텐츠와 정보에 대해 책임을 지지 않습니다. 사용자는 이러한 콘텐츠와 정보의 사용 위험을 감수해야 합니다.

1.3 인터넷에서 제공되는 문서

1.3.1 매뉴얼 개요 SINUMERIK 840D sl

840D sl 문서 개요 (https://support.industry.siemens.com/cs/ww/en/view/109766213)에서 SINUMERIK 840D sl의 기능에 관한 종합적인 문서를 제공합니다.

문서를 PDF 및 HTML5 형식으로 표시하거나 다운로드할 수 있습니다.

문서는 다음과 같은 카테고리로 나뉩니다.

- 사용자: 작동
- 사용자: 프로그래밍
- 제조업체/서비스: 기능
- 제조업체/서비스: 하드웨어
- 제조업체/서비스: 구성/설정
- 제조업체/서비스: Safety Integrated
- 제조업체/서비스: SINUMERIK Integrate/MindApp
- 정보 및 교육
- 제조업체/서비스: SINAMICS

1.3.2 문서 개요 SINUMERIK 조작반 구성요소

문서 개요 SINUMERIK 조작반 구성요소 (https://support.industry.siemens.com/cs/document/109783841/technische-dokumentation-zu-sinumerik-bedienkomponenten?dti=0&lc=en-WW)에서 SINUMERIK 조작반 구성요소에 관한 종합적인 문서를 제공합니다

문서를 PDF 및 HTML5 형식으로 표시하거나 다운로드할 수 있습니다.

문서는 다음과 같은 카테고리로 나뉩니다.

- 화면 조작반
- 기계 조작반
- 기계 푸시버튼 조작반
- 핸드헬드 장치/미니 핸드헬드 장치
- 추가 조작반 구성요소

SINUMERIK 개요 - 주제 페이지 (https://support.industry.siemens.com/cs/document/109766201/sinumerik-an-overview-of-the-most-important-documents-and-links?dti=0&lc=en-WW)에서 SINUMERIK에 관한 가장 중요한 문서, 항목 및 링크의 개요를 제공합니다.

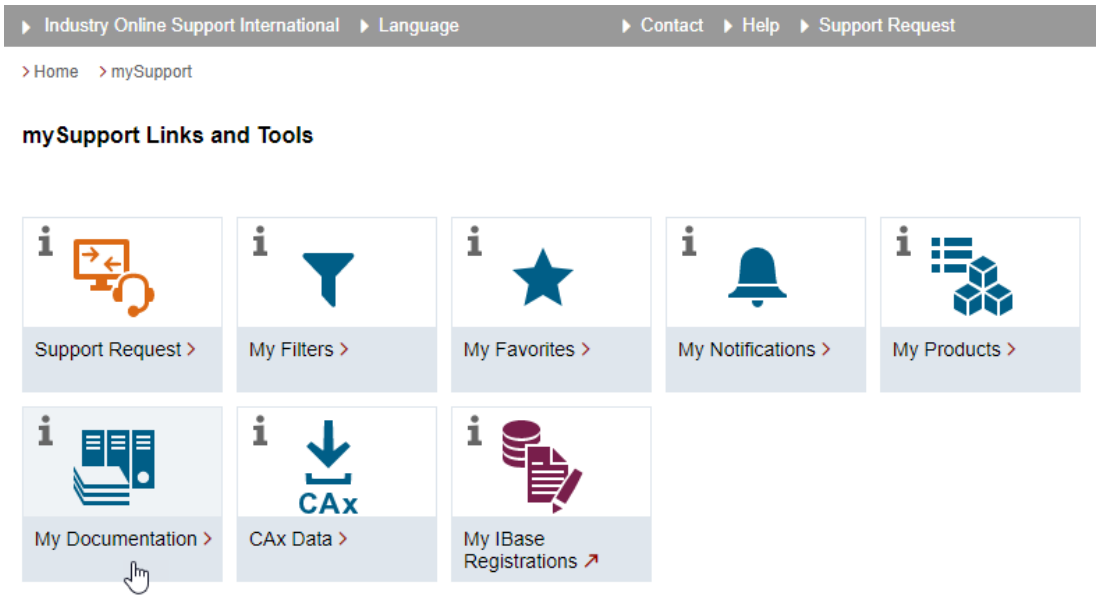
1.4 기술 문서에 대한 피드백

Siemens Industry Online Support에 게시되는 기술 문서에 대해 질문, 제안 또는 수정 사항이 있을 경우 항목 끝에 표시되는 "피드백 보내기" 링크를 사용하십시오.

1.5 mySupport 문서

"mySupport 문서" 웹 기반 시스템을 사용하면 Siemens 콘텐츠를 기반으로 자신만의 장비 문서를 편집할 수 있습니다.

애플리케이션을 시작하려면 mySupport 홈 페이지 (https://support.industry.siemens.com/cs/ww/en/my)에서 "내 문서" 타일을 클릭하십시오.



구성한 문서를 RTF, PDF 또는 XML 형식으로 내보낼 수 있습니다.

참고

mySupport 문서 애플리케이션을 지원하는 Siemens 콘텐츠에는 "구성" 링크가 있습니다.

1.6 서비스 및 지원

제품 지원

제품 정보는 인터넷에서 확인할 수 있습니다.

제품 지원 (https://support.industry.siemens.com/cs/ww/en/)

이 주소로 이동하면 다음이 제공됩니다.

- 최신 제품 정보(제품 소개)
- FAQ(자주 묻는 질문)
- 매뉴얼
- 다운로드
- 최신 제품 정보를 소개하는 뉴스레터
- 사용자와 전문가가 정보와 모범 사례를 공유하는 글로벌 포럼

- Siemens 데이터베이스의 현지 담당자 연락처(→ "연락처")
- 현장 서비스, 수리, 예비 부품 등에 관한 정보(→ "현장 서비스")

기술 지원

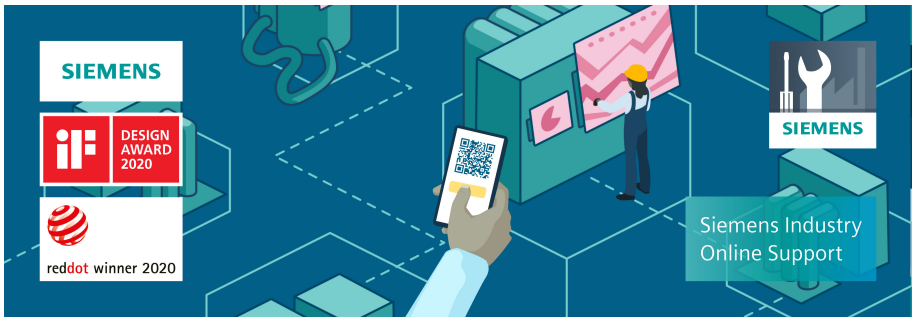
국가별 기술 지원 전화번호는 웹 사이트의 "연락처" 섹션에 있는 주소 (https://support.industry.siemens.com/cs/ww/en/sc/4868)에서 확인할 수 있습니다.

기술적인 질문이 있는 경우 "지원 요청"의 온라인 양식을 사용하십시오.

교육

다음 주소 (https://www.siemens.com/sitrain)에서 SITRAIN 관련 정보를 확인할 수 있습니다.
SITRAIN은 Siemens의 자동화 및 드라이브 제품, 시스템 및 솔루션에 대한 교육 과정을 제공합니다.

이동 중의 Siemens 지원



수상 경력이 있는 "Siemens Industry Online Support" 앱을 사용하면 언제 어디서나 Siemens Industry 제품에 관한 30만 개 이상의 문서에 액세스할 수 있습니다. 이 앱은 다음을 지원합니다.

- 프로젝트 구현 시 문제 해결
- 오류 발생 시 문제 해결
- 시스템 확장 또는 새 시스템 계획

또한 기술 포럼과 다른 전문가 문서에도 액세스할 수 있습니다.

- FAQ
- 적용 예
- 매뉴얼

- 인증서
- 제품 소개 등

"Siemens Industry Online Support" 앱은 Apple iOS 및 Android에서 사용할 수 있습니다.

명판의 데이터 매트릭스 코드

명판의 데이터 매트릭스 코드에는 특정 장치 데이터가 포함되어 있습니다. 이 코드를 스마트폰으로 읽으면 장치에 대한 기술 정보가 "Industry Online Support" 모바일 앱을 통해 표시됩니다.

1.7 중요 제품 정보

OpenSSL 사용

이 제품에는 다음과 같은 소프트웨어가 포함될 수 있습니다.

- OpenSSL 툴킷에서 사용하기 위해 OpenSSL 프로젝트에 의해 개발된 소프트웨어
- Eric Young이 개발한 암호화 소프트웨어
- Eric Young이 개발한 소프트웨어

자세한 정보는 인터넷에서 확인할 수 있습니다.

- OpenSSL (https://www.openssl.org)
- Cryptsoft (https://www.cryptsoft.com)

일반 데이터 보호 규정 준수

Siemens는 표준 데이터 보호 원칙, 특히 데이터 최소화 규칙(privacy by design)을 준수합니다.

이 제품의 경우 다음을 의미합니다.

이 제품은 개인 데이터는 처리 또는 저장하지 않고 기술 기능 데이터(예: 타임스탬프)만 처리 또는 저장합니다. 사용자가 이 데이터를 다른 데이터(예: 교대 계획)에 연결하거나 개인 관련 데이터를 동일한 데이터 매체(예: 하드 디스크)에 저장해 이 데이터를 개인화할 경우, 해당 데이터 보호 조항을 준수해야 합니다.

기본 안전 지침 2

2.1 일반 안전 지침

⚠ 경고

안전 지침과 잔류 위험 지침을 준수하지 않는 경우 생명 위험

관련 하드웨어 문서의 안전 지침과 잔류 위험 지침을 준수하지 않는 경우 심각한 부상을 당하거나 사망에 이를 수 있습니다.

- 하드웨어 문서에 기술된 안전 지침을 준수하십시오.
- 잔류 위험에 대해서는 위험 평가를 실시하십시오.

⚠ 경고

파라미터 설정의 오류나 변경으로 인한 장비 오작동

부정확하거나 잘못 수정된 파라미터로 인해 기계가 고장 나거나 부상이나 인명 피해가 발생할 수 있습니다.

- 무단 액세스로부터 파라미터를 보호하십시오.
- 오작동이 발생하면 적절한 조치(예: 비상 정지 또는 비상 차단)를 취해 대응하십시오.

2.2 적용 예제의 보증 및 책임

적용 예제는 구속력이 없으며, 구성, 장비 또는 발생할 수 있는 상황에 대해 완전하지 않습니다. 적용 예제는 특정 고객의 솔루션을 나타내지 않으며, 일반적인 작업에 지원을 제공하는 용도로만 이용됩니다.

설명된 제품을 정확하게 작동할 책임은 사용자 자신에게 있습니다. 적용 예제는 장비를 사용, 설치, 작동 및 유지보수할 때 안전한 취급에 대한 사용자의 책임을 면제하지 않습니다.

2.3 보안 정보

Siemens는 공장, 시스템, 기계류 및 네트워크의 안전한 작동을 지원하는 산업 보안 기능과 함께 제품 및 솔루션을 제공합니다.

공장, 시스템, 기계류 및 네트워크를 사이버 위협으로부터 보호하기 위해서는 통합적인 최첨단의 산업 보안 개념을 구현하고 지속적으로 유지 관리해야 합니다. Siemens의 제품과 솔루션은 이러한 전체적인 개념의 한 요소만을 구성합니다.

고객은 고객 소유의 공장, 시스템, 기계류 및 네트워크에 대한 무단 액세스를 방지할 책임이 있습니다. 시스템, 기계류 및 구성 요소는 필요한 범위 내에서, 그리고 적절한 보안 조치가 취해진 상태(예: 방화벽 및 네트워크 세그멘테이션 사용)에서만 회사 네트워크 또는 인터넷에 연결되어야 합니다.

이행될 수 있는 산업 보안 조치에 대한 자세한 내용은 https://www.siemens.com/industrialsecurity (https://www.siemens.com/industrialsecurity)를 참조하십시오.

Siemens의 제품과 솔루션은 지속적인 개발을 통해 보안을 더욱 강화하고 있습니다. Siemens는 제품 업데이트가 나오는 즉시 적용하고 항상 최신 제품 버전을 사용할 것을 강력히 권장합니다. 더 이상 지원되지 않는 제품 버전을 사용하거나, 최신 업데이트를 적용하지 않으면 사이버 위협에 노출될 가능성이 높아집니다.

제품 업데이트에 대한 정보를 제때 받아보려면 https://www.siemens.com/industrialsecurity (https://new.siemens.com/global/en/products/services/cert.html#Subscriptions)에서 Siemens 산업 보안 RSS 피드를 구독하십시오.

추가 정보는 인터넷에서 제공합니다:

산업 안전 환경설정 매뉴얼 (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ 경고

악의적인 소프트웨어 조작으로 인한 안전하지 않은 작동 상태

악의적인 소프트웨어 조작(예: 바이러스, 트로이 목마, 웜)은 시스템 운영 상태를 위험에 빠뜨려 사망, 심각한 부상, 재산상의 피해를 야기할 수 있습니다.

- 소프트웨어를 최신 상태로 유지하십시오.
- 설비 또는 기계의 자동화 및 드라이브 콤포넌트를 총체적인 첨단 산업 보안 개념에 통합시키십시오.
- 반드시 모든 설치 제품들을 총체적 산업 보안 개념에 포함시키십시오.
- 바이러스 검사 프로그램 같은 적절한 보호 조치를 통해 교환 가능 저장 매체에 저장된 파일을 악성 소프트웨어로부터 보호하십시오.
- 스타트업이 완료되면 모든 보안 관련 설정을 확인하십시오.

기본 사항

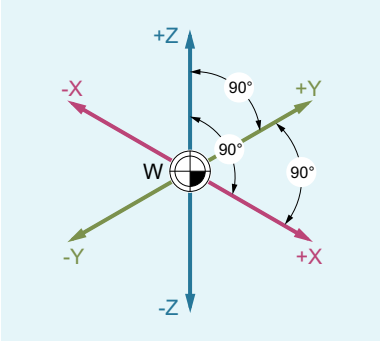
3

3.1 기본적인 기하학 원리

3.1.1 공작물위치

3.1.1.1 포지셔닝의 참조 시스템

기계 또는 제어 시스템이 NC 프로그램에서 명시한 위치에서 작동하려면 기계 축이 움직이는 방향으로 이송될 수 있는 기준 좌표계에 위치를 지정해야 합니다. DIN 66217에 따른 직교 좌표계 (CW 방향, 직각) 가 공작 기계의 공작물 좌표계로 사용됩니다.



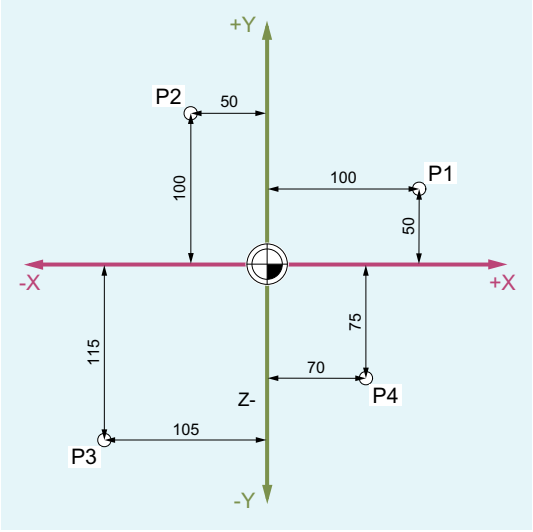
공작물 영점(W)은 공작물 좌표계의 원점입니다.

3.1.1.2 직교 좌표

좌표계의 축에는 수치가 부여됩니다. 이 방법으로 좌표계 내의 모든 지점을 명확하게 표시할 수 있습니다. 즉, 모든 공작물의 위치를 방향 (X, Y 및 Z) 과 3 개의 숫자 값으로 명확히 표시할 수 있습니다. 공작물 원점의 좌표는 항상 X0, Y0, Z0입니다.

직교좌표 양식의 포지셔닝

이해를 돕기 위해 다음 예에서 좌표계의 한 평면, X/Y 면만을 고려해보겠습니다.

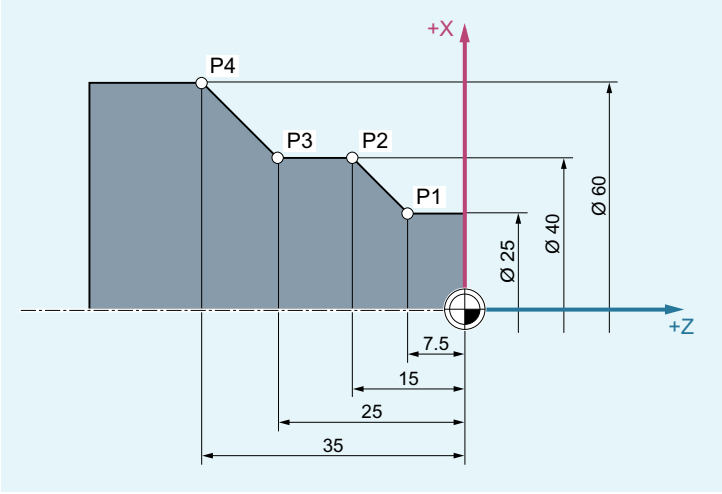


지점 P1에서 P4의 좌표는 다음과 같습니다.

위치	좌표
P1	X100 Y50
P2	X-50 Y100
P3	X-105 Y-115
P4	X70 Y-75

예: 선삭용 공작물 위치

선반에서는 형상을 표시하는데 한 면이면 충분합니다.

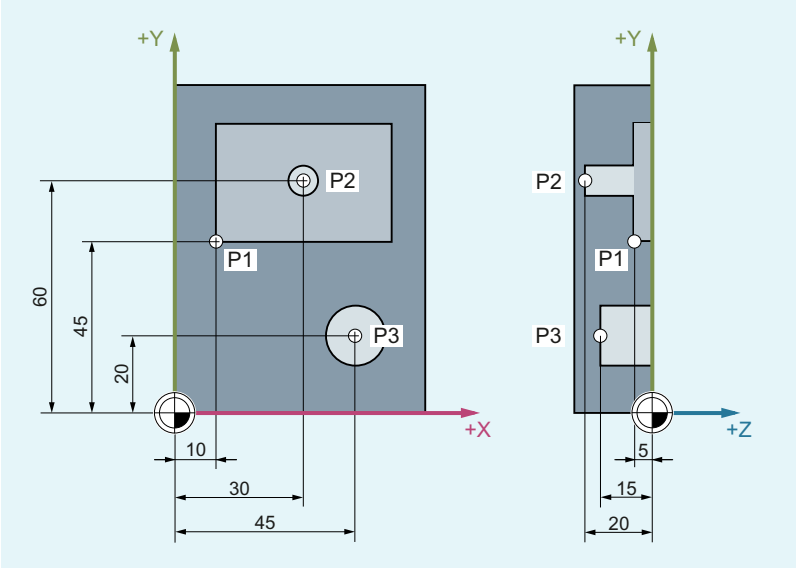


지점 P1에서 P4의 좌표는 다음과 같습니다.

위치	좌표
P1	X25 Z-7.5
P2	X40 Z-15
P3	X40 Z-25
P4	X60 Z-35

예: 밀링용 공작물 위치

밀링의 경우 세 번째 좌표(이 경우 Z)에도 수치 값이 반드시 부여되는 예와 같이 피드 깊이 역시 반드시 표시되어야 합니다.



점 P1에서 P3까지는 다음 좌표를 가집니다:

위치	좌표
P1	X10 Y45 Z-5
P2	X30 Y60 Z-20
P3	X45 Y20 Z-15

3.1.1.3 극 좌표

공작물 위치를 표시하는데 직교 좌표 대신에 극 좌표가 사용될 수 있습니다. 공작물이나 공작물 일부의 치수가 반경과 각도로 표시되었을 때 유용합니다. 치수가 시작되는 지점을 "극점"이라고 합니다.

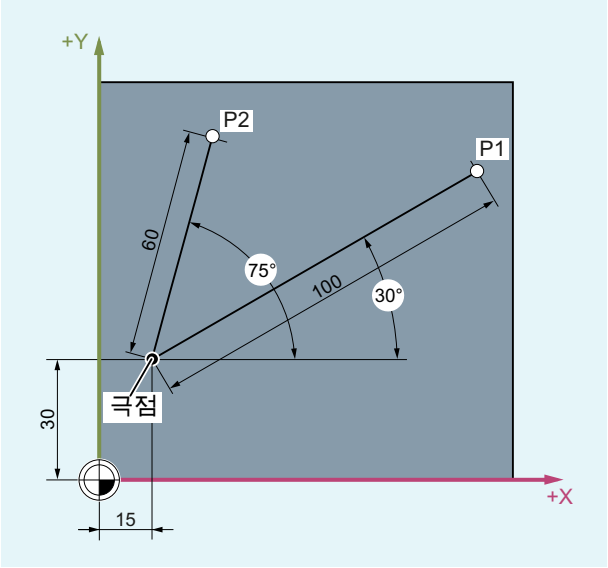
극 좌표 양식의 위치 지정

극 좌표는 **극점 반경**과 **극점 각도**로 구성됩니다.

극점 반경은 극점과 위치사이의 거리입니다.

극점 각도는 극점 반경과 공작물 평면에 있는 수평 축과의 각도입니다. 음의 극점 각도는 시계 방향에 존재하고 양의 극점 각도는 시계 반대 방향에 존재하게 됩니다.

예



극점을 기준으로 한 P1과 P2는 다음과 같이 표시됩니다:

위치	극 좌표
P1	RP=100 AP=30
P2	RP=60 AP=75
RP: 극점 반경 AP: 극점 각도	

3.1.1.4 절대 치수

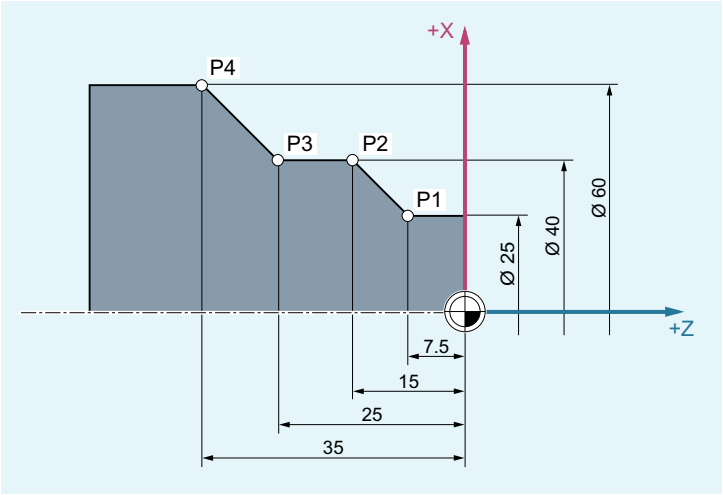
절대 치수로의 위치 지정

절대 좌표계를 사용할 경우에는 모든 위치 지정은 항상 현재 설정된 영점을 기준으로 합니다.

공구 이송에 적용됩니다, 이 의미는:

공구가 이동하게되는 위치.

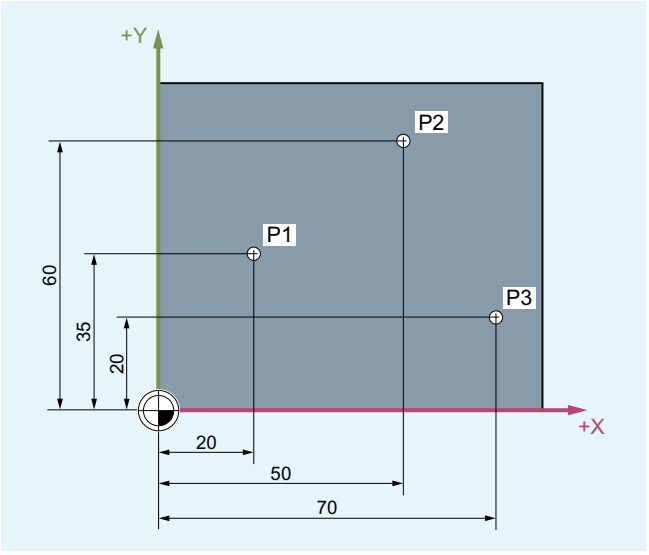
예: 선삭



절대 치수에서는 점 P1에서 P4까지에 다음 위치 지정이 나타납니다.

위치	절대 치수로의 위치 지정
P1	X25 Z-7.5
P2	X40 Z-15
P3	X40 Z-25
P4	X60 Z-35

예: 밀링



절대 치수에서는 점 P1에서 P3까지에 다음 위치 지정이 나타납니다.

위치	절대 치수로의 위치 지정
P1	X20 Y35
P2	X50 Y60
P3	X70 Y20

3.1.1.5 증분 치수

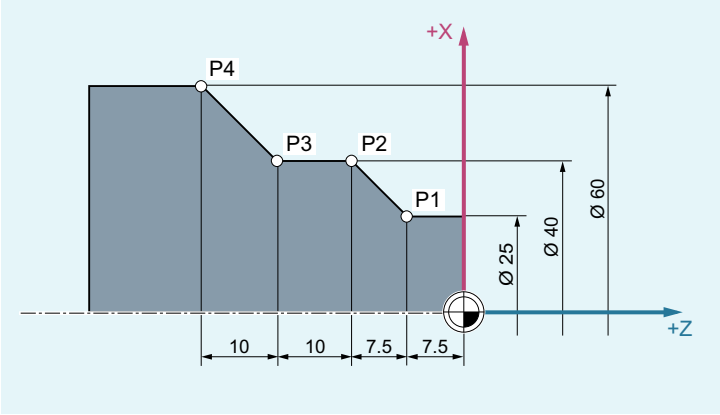
증분 치수를 사용한 포지셔닝

생산 도면에서 치수가 영점이 아니라 다른 공작물 점을 나타내는 경우가 있습니다. 이러한 치수는 변환하지 않아도 되므로 증분 치수로 지정할 수 있습니다. 증분 치수로 프로그램 할 경우, 이전에 프로그래밍된 좌표가 포지셔닝의 기준점이 됩니다.

공구 이동에 사용된 경우는 다음을 의미합니다:

증분 치수는 공구가 이동할 거리를 나타냅니다.

예: 선삭



증분 치수에서 P2부터 P4까지의 포지셔닝은 다음과 같습니다.

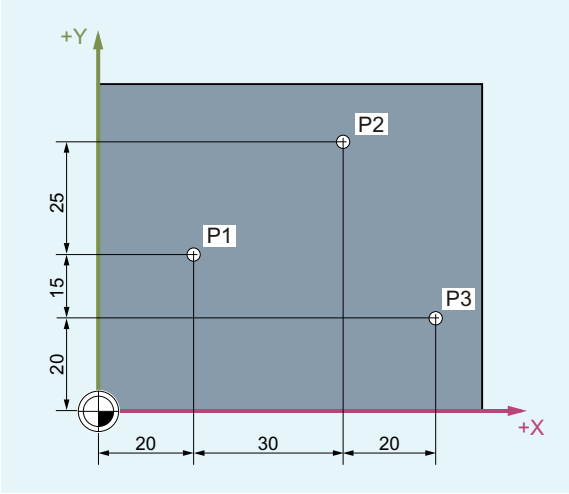
위치	증분치 좌표 지정	기준점:
P2	X15 Z-7.5	P1
P3	Z-10	P2
P4	X20 Z-10	P3

참고

DIAMOF 또는 DIAM90 (페이지 180) 이 활성된 상태에서 증분 치수 (G91) 는 반경으로 프로그램 됩니다.

예: 밀링

증분 치수에서의 P1부터 P3의 포지셔닝은 다음과 같습니다:



증분 치수에서 P1부터 P3까지의 포지셔닝은 다음과 같습니다.

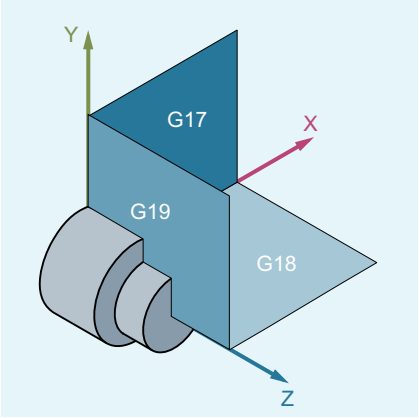
위치	증분치 좌표 지정	기준점:
P1	X20 Y35	원점
P2	X30 Y20	P1
P3	X20 Y-35	P2

3.1.2 작업 면

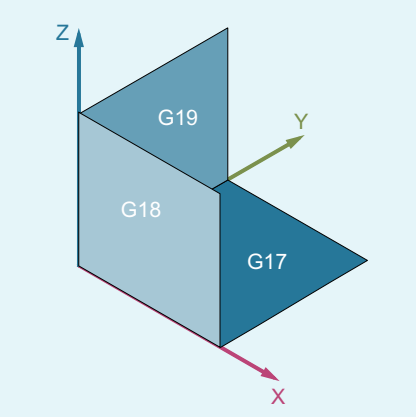
NC 프로그램에서는 가공 평면에 대한 정보가 필요합니다. 이 정보가 있을 때에만 제어 시스템에서 공구 옵셋 값을 올바르게 고려할 수 있습니다. 또한 특정 원호-경로 프로그래밍과 극좌표 프로그래밍에서도 작업 평면을 지정해야 합니다.

작업 평면은 2개의 좌표 축을 사용하는 베이스 직교 공작물 좌표계에서 지정합니다. 세 번째 좌표 축은 이 작업 평면에 직각이 되며 공구의 절입 방향을 결정합니다(예: 2D 가공).

선삭/밀링 가공 평면



선삭 가공 평면



밀링 가공 평면

가공 평면 활성화

작업 평면은 NC 프로그램에서 G 명령인 ? G17, G18, G19를 사용해 활성화합니다. 관계는 다음과 같이 정의합니다.

G 코드	가공 평면	가로 좌표	세로 좌표	직교 ≙ 절입 방향
G17	X/Y	X	Y	Z
G18	Z/X	Z	X	Y
G19	Y/Z	Y	Z	X

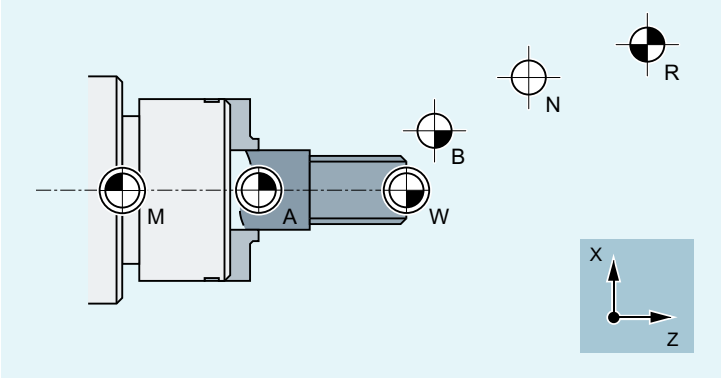
3.1.3 영점 및 원점

다양한 영점과 원점이 NC 프로그램에서 정의됩니다.

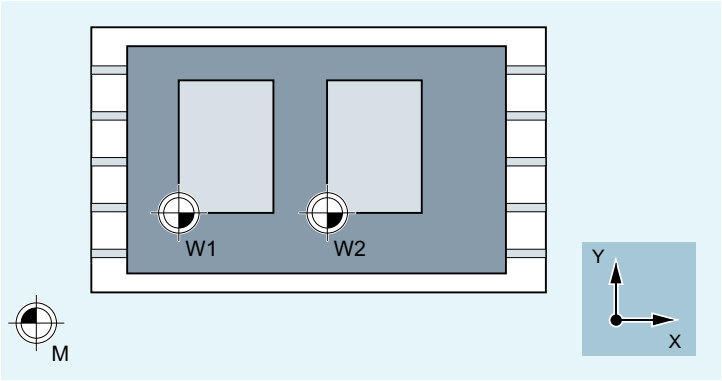
영점		
	M	기계 원점 기계 원점은 기계 좌표계 (MCS) 를 정의합니다. 다른 모든 원점은 기계 원점을 기준합니다.
	W	공작물 원점 = 프로그램 영점 공작물 원점은 기계 원점과 관련된 공작물 좌표계를 정의합니다.
	A	차단점 공작물 원점과 동일할 수 있습니다 (선반의 경우에만).

원점		
	R	원점 원점 도구 및 측정 시스템에 의해 정의된 위치 현재 지점의 축 위치를 원점으로 설정하기 위해서 반드시 기계 원점 **M**까지의 위치를 알아야 합니다.
	B	시작점 프로그램에 의하여 정의될 수 있습니다. 첫 번째 공구가 여기서 가공을 시작합니다.
	T	공구 홀더 원점 공구 홀더 상에 있습니다. 제어 시스템은 공구 길이를 입력하여 공구 팁과 공구 홀더 원점 사이의 거리를 계산합니다.
	N	공구 교환 위치

선삭의 영점 및 원점



밀링 영점



3.1.4 좌표계

다음 좌표계 간의 상이점

- 기계 영점 **M**의 기계 좌표계(MCS) (페이지 41)
- 기본 좌표계 (BCS) (페이지 43)
- 베이직 제로 시스템 (BZS) (페이지 45)
- 셋터블 제로 시스템 (SZS) (페이지 45)
- 공작물 영점 **W**의 공작물 좌표계 (WCS) (페이지 46)

3.1.4.1 기계 좌표계(MCS)

기계 좌표계는 물리적으로 존재하는 모든 축을 포함합니다.

기준점, 공구 및 팔레트 변경점(고정된 기계점)은 기계 좌표계에서 정의됩니다.

프로그래밍이 기계 좌표계에서 직접 수행되면(G 명령을 이용할 경우 가능) 기계의 물리적 축이 직접 반응합니다. 설치된 모든 공작물 clamp는 고려하지 않습니다.

참고

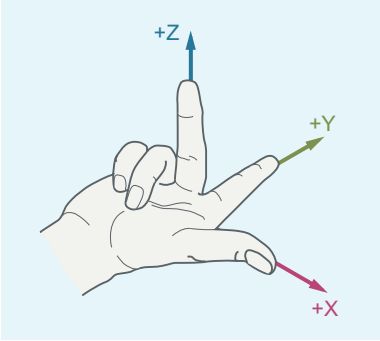
다양한 기계 좌표계가 사용된다면(예, 5-축 변형) 프로그래밍이 수행되는 좌표계에 기계 운동을 매핑하는 데 사용됩니다.

세 손가락 법칙

기계에 기준이 되는 좌표계의 방향은 각 기계 종류에 따라 다릅니다. 축 방향은 이른바 **오른** 손의 '세 손가락 법칙'이라 불리는 것을 따른다(DIN 66217에 의거).

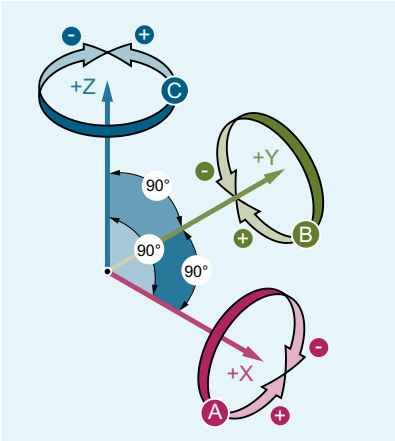
기계 정면에서 보았을 때 오른손의 가운데 손가락을 스핀들의 절입 방향과 반대 방향으로 둡니다. 따라서

- 엄지손가락 방향이 +X 방향이 됩니다
- 검지 손가락 방향이 +Y 방향이 됩니다
- 가운데 손가락 방향이 +Z 방향이 됩니다



좌표 축 X,Y, Z를 기준으로 회전하는 운동은 각각 A, B, C로 명명됩니다. 좌표 축 양의 방향에서 볼 때의 회전 운동 방향에서 회전 방향이 정해집니다.

회전 운동의 방향	회전 방향
CW 방향	양의
CCW 방향	음의

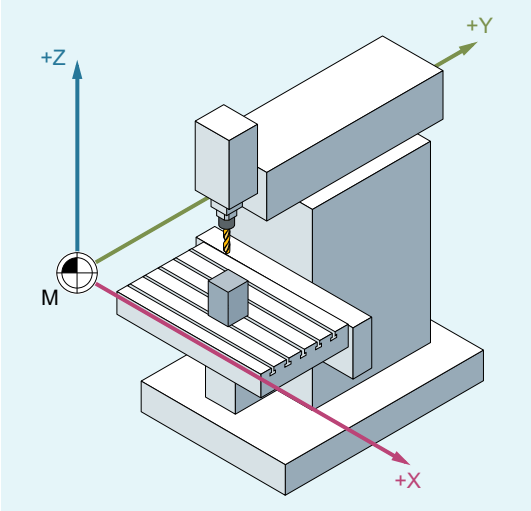


X, Y, Z 다른 축의 상단에 정렬된 수직 좌표 축

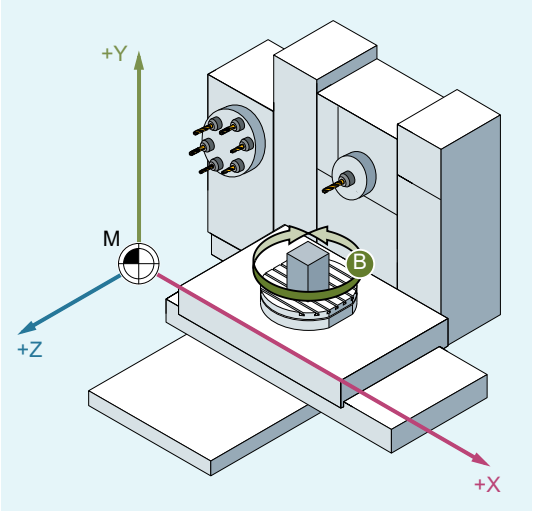
A, B, C 로터리 축: X, Y, Z 축 기준 회전

상이한 기계 종류의 좌표계 위치

'세 손가락 법칙'에서 도출된 좌표계 위치는 기계 종류에 따라 방향이 다를 수 있습니다(아래 예 참고).



수직 3축 밀링 기계



수평 4축 밀링 기계

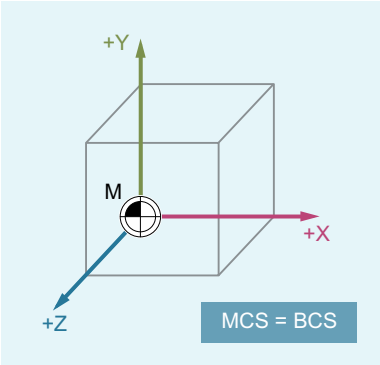
3.1.4.2 기본 좌표계 (BCS)

기본 좌표계 (BCS) 는 서로 수직을 이루는 3개의 축 (기하 축) 및 기하학적으로 보간되지 않은 기타 특수 축으로 구성됩니다.

키네마틱 변환이 불가능한 기계

키네마틱 변환(예: 5축 변환, TRANSMIT/TRACYL/TRAANG)을 실시하지 않은 채 BCS를 MCS로 매핑하면 BCS와 MCS가 동일하게 됩니다.

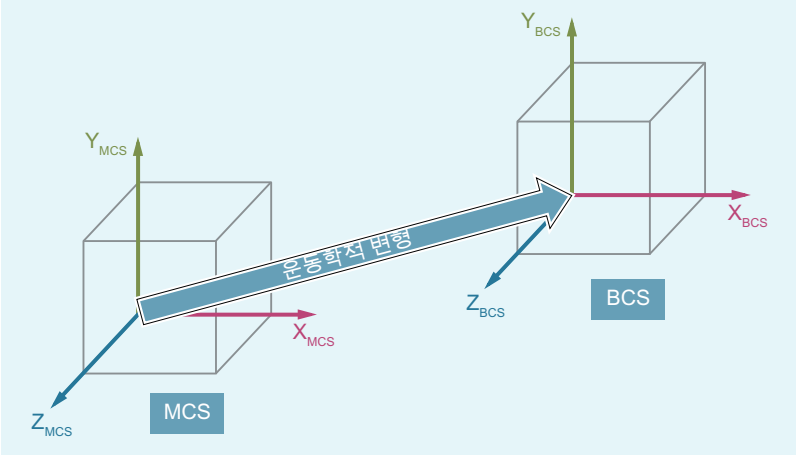
그런 기계에서는 기계 축 및 기하 축의 이름이 동일해도 무방합니다.



키네마틱 변환이 가능한 기계

키네마틱 변환(예: 5축 변환, TRANSMIT/TRACYL/TRAANG)을 실시한 상태에서 BCS를 MCS로 매핑하면 BCS와 MCS가 동일하지 않게 됩니다.

그런 기계에서는 기계 축 및 기하 축의 이름이 반드시 달라야 합니다.

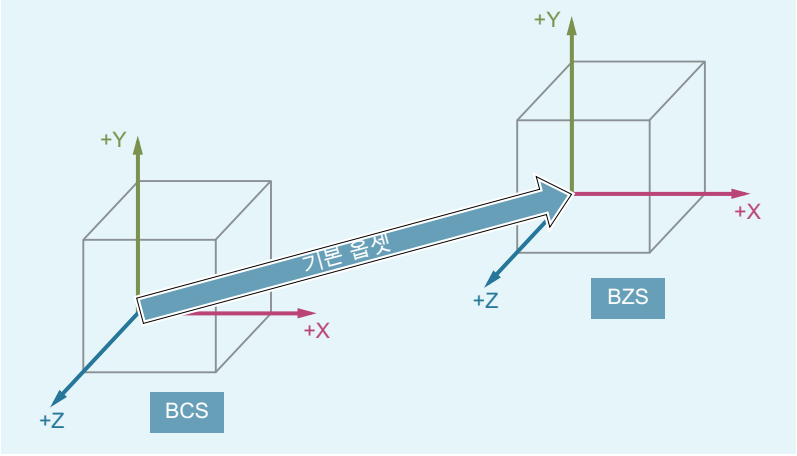


기계 구조

공작물은 항상 2차원 또는 3차원 직교 좌표계 (WCS)로 프로그래밍됩니다. 그러나 그런 공작물은 서로 수직을 이루지 않은 직선 축이나 로터리 축이 설치된 공작 기계에서 더 빈번하게 프로그래밍됩니다. 실제 기계 축 운동에서 공작물 좌표계(직사각형)에 프로그래밍된 좌표를 나타낼 때 키네마틱 변환이 사용됩니다.

3.1.4.3 베이직 제로 시스템 (BZS)

기본 제로 시스템(BZS)은 기본 옵셋을 포함한 기본 좌표계에서 파생된 시스템입니다.



기본 옵셋

기본 옵셋은 BCS 및 BZS 간의 좌표 변환을 나타냅니다. 예를 들어 팔레트 영점을 정의하는데 사용됩니다.

기본 옵셋은 다음을 포함합니다:

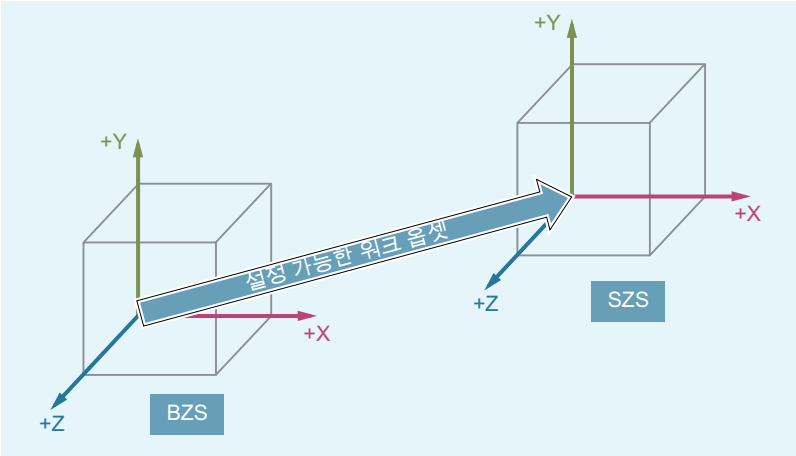
- 외부 워크 옵셋
- DRF 옵셋
- 오버레이 이동
- 연계된 시스템 프레임
- 연계된 기본 프레임

3.1.4.4 셋터블 제로 시스템 (SZS)

설정 가능한 워크 옵셋

'설정 가능한 제로 시스템'(SZS)은 설정 가능한 워크 옵셋을 통해 기본 제로 시스템(BZS)에서 도출한 시스템입니다.

설정 가능한 워크 옵셋은 G 명령어 G54~G57과 G505~G599을 이용해 NC 프로그램에서 활성화됩니다.



프로그램 좌표계(프레임)를 사용하지 않으면 '설정 가능한 제로 시스템'이 공작물 좌표계(WCS) 가 됩니다.

프로그램 좌표계 (프레임)

종종 처음 선택한 공작물 좌표계(또는 '설정 가능한 제로 시스템')를 다른 위치로 이동, 회전, 미러링 및/또는 배율 조정을 해야 하거나 하는 것이 좋습니다. 이것은 프로그램 좌표 변환 (프레임) (페이지 337) 을 사용하여 수행됩니다.

참고

프로그램 좌표 변환(프레임)의 기준은 '설정 가능한 제로 시스템' 입니다.

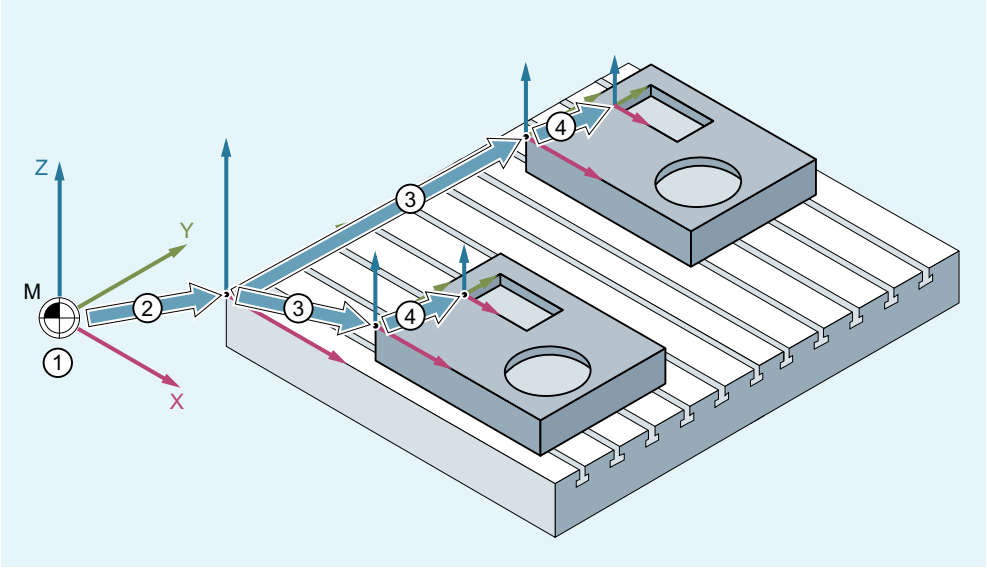
3.1.4.5 공작물 좌표계 (WCS)

공작물 형상이 공작물 좌표계(WCS)에 표시됩니다. 다시 말해 NC 프로그램의 데이터는 공작물 좌표계를 기준으로 합니다.

공작물 좌표계는 항상 직교 좌표계이며 특정 공작물에 부여됩니다.

3.1.4.6 다양한 좌표계는 어떠한 관계인가?

다음 예를 보면 다양한 좌표계의 관계를 이해하는 데 도움이 됩니다.



① 키네마틱 변환이 활성화되지 않았습니다. 즉, 기계 좌표계와 기본 좌표계가 동일합니다.

② 팔레트 제로점의 기본 제로 시스템(BZS)은 기본 옵셋에서 도출합니다.

③ 설정 가능한 작업 옵셋 G54 또는 G55는 공작물 1 또는 2의 '설정 가능한 제로 시스템'(SZS)을 지정합니다.

④ 공작물 좌표계(WCS)는 프로그래밍이 가능한 좌표 변환의 결과입니다.

3.2 NC 프로그램 작성의 기본 원칙

참고

DIN 66025 이 NC 프로그램 작성의 지침이 된다.

3.2.1 NC 프로그램의 이름

규칙

NC 프로그램을 생성할 때마다 프로그램 이름(명칭)을 지정해야 합니다. 프로그램 이름은 다음 규칙에 따라 자유롭게 선택할 수 있습니다.

- 허용되는 문자:
  - 문자: A ... Z, a ... z
  - 숫자: 0 ... 9
  - 밑줄: _
- 처음 2개의 문자는 반드시 2개의 글자이거나 밑줄 다음에 1개의 글자로 구성해야 합니다.

  참고

  이 조건이 만족되면 프로그램 이름을 지정하여 다른 프로그램에서 NC 프로그램을 서브프로그램으로 호출할 수 있습니다. 하지만 프로그램 이름이 숫자로 시작되면 CALL 명령문을 통해서만 서브프로그램을 호출할 수 있습니다.

- 최대 길이: 24문자

참고

대문자/소문자

SINUMERIK NC 언어는 대문자와 소문자를 구분하지 **않습니다**.

참고

허용되지 않는 프로그램 이름

Windows 응용 프로그램에서의 문제 방지를 위해 다음과 같은 프로그램 이름은 사용할 수 **없습니다**.

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

그 외 규칙은 '이름 (페이지 422)'을 참조하십시오.

시스템 내부 확장자

프로그램을 생성할 때 지정한 프로그램 이름을 제어 시스템에서 접두어 및 접미어를 추가하여 확장할 수 있습니다.

- 접두어: _N_
- 접미어:
  - 메인 프로그램: _MPF
  - 서브 프로그램: _SPF

천공 테이프 양식의 파일

외부에서 생성한 후 V.24 인터페이스를 통해 읽어오는 프로그램 파일은 반드시 천공 테이프 형식이어야 합니다.

천공 테이프 형식의 파일 이름에는 다음과 같은 추가 규칙이 적용됩니다.

- 첫 번째 문자: %
- 그 다음, 4문자 파일 확장자: _xxx

예:

- %_N_SHAFT123_MPF
- %Flange3_MPF

3.2.2 NC 프로그램의 구조와 내용

3.2.2.1 블록과 블록 구성 요소

블록

NC 프로그램은 NC 블록의 연속으로 구성됩니다. 공작물 가공 단계의 실행에 관한 정보가 각 블록에 포함되어 있습니다.

블록 구성요소

NC 블록은 다음 구성요소로 구성됩니다.

- DIN 66025에 따른 명령(성명)
- NC 고급 언어의 요소

DIN 66025에 따른 명령

DIN 66025에 의거한 명령은 어드레스 문자와 숫자 또는 산술값을 나타내는 숫자 시퀀스로 구성됩니다.

어드레스 문자(어드레스)

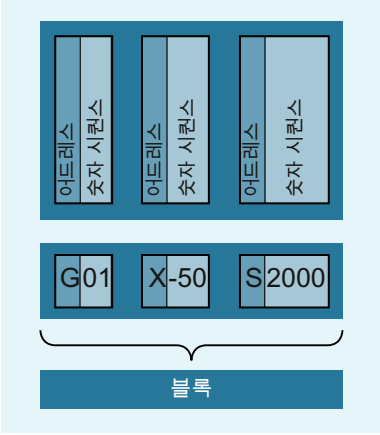
어드레스 문자(일반적으로 문자)는 명령어의 의미를 정의합니다.

예:

어드레스 문자	의미
G	G 명령(준비 기능)
X	X축의 위치 데이터
S	스핀들 속도

숫자 시퀀스

숫자 시퀀스는 어드레스 문자에 부여된 값입니다. 숫자 시퀀스는 신호 및 소수점을 포함할 수 있습니다. 신호는 항상 어드레스 문자와 숫자 시퀀스 사이에 나타납니다. 양의 기호(+)와 앞의 0은 명시하지 않아도 됩니다.



NC 고급 언어의 요소

현대 기계 공구의 복잡한 기계 가공 시퀀스 프로그램 작성에 DIN 66025에 의거한 명령어 설정이 더 이상 적합하지 않기 때문에 NC 고급 언어의 요소로 확장되었습니다.

이 기능에는 예를 들어 다음이 포함됩니다:

- NC 고급 언어의 명령어
  DIN 66025 에 의거한 명령어와는 대조적으로 NC 고급 언어의 명령어는 아래와 같은 몇 가지 어드레스로 구성됩니다:
  - 속도 오버라이드 OVR
  - 스핀들 포지셔닝 SPOS
- 식별자 (정의된 이름)는:
  - 시스템 변수
  - 사용자 정의 변수
  - 서브프로그램
  - 키워드
  - 분기 라벨
  - 매크로

  참고

  식별자는 반드시 고유한 것이어야 하고 다른 목표물에는 사용할 수 없습니다.
- 비교 연산자
- 논리 운전자
- 산술 함수
- 제어 구조

명령의 유효성

명령어는 모달 또는 넌모달 중 하나입니다.

- 모달
  모달 명령어는 다음 사항이 발생될 때까지 모든 후속 블록에서 적용됩니다.
  - 동일한 명령으로 새로운 값이 프로그래밍될 때까지
  - 해당 명령을 취소하는 명령이 프로그래밍될 때까지
- 넌모달
  넌모달 명령어는 프로그래밍된 블록에만 적용됩니다.

프로그램의 끝

실행 순서에서 마지막 블록에는 프로그램 끝을 나타내는 특수한 단어가 포함됩니다: M2, M17 또는 M30.

3.2.2.2 블록 규칙

블록 시작

NC 블록은 블록 시작에서 블록 번호로 식별될 수 있다. 이는 문자 "N"과 양의 정수로 구성됩니다. 예,
`N40`

블록 번호의 순서는 임의적이지만 내림차순이 권고됩니다.

참고

검색시 애매모호한 결과를 피하려면 프로그램 내에서 블록 번호는 각기 유일한 것이어야 합니다.

블록 끝

블록은 문자 LF (LINE FEED = 새 라인) 로 끝납니다.

참고

LF 문자는 입력할 필요가 없습니다. 라인 변경으로 자동으로 생성됩니다.

블록 길이

블록에는 최대 **512개의 문자**가 포함될 수 있습니다 (주석과 블록 끝 문자 LF 포함).

참고

보통 각기 66 개까지의 문자 블록 3 개가 화면의 현 블록 디스플레이에 디스플레이됩니다. 주석 역시 디스플레이됩니다. 메시지는 별도의 메시지 창에 디스플레이됩니다.

성명의 순서

가능한 가장 명백한 블록 구조를 유지하려면 블록 내의 성명은 다음 순서를 따라야 합니다.
`N... G... X... Y... Z... F... S... T... D... M... H...`

어드레스	의미
`N`	블록 번호의 어드레스
`G`	준비 기능
`X,Y,Z`	위치 데이터

F	이송 속도
S	속도
T	공구
D	공구 옵셋 번호
M	추가 기능
H	보조 기능

참고

특정 어드레스는 블록 내에서 반복적으로 사용될 수 있습니다. 예를 들면,

G..., M..., H...

3.2.2.3 값 지정

어드레스에 값이 지정될 수 있습니다. 다음과 같은 규칙이 적용됩니다:

- 다음의 경우에는 반드시 "=" 기호 하나가 어드레스와 값 사이에 삽입되어야 합니다.
  - 어드레스는 하나 이상의 문자로 구성됩니다.
  - 하나 이상의 상수가 값에 포함됩니다.

  어드레스가 단일 문자이고 값이 하나의 상수로만 구성된 경우 "=" 기호는 생략할 수 있습니다.
- 기호는 허용됩니다.
- 어드레스 문자 뒤에 분별자가 허용됩니다.

예:

X10	어드레스 X 에의 값 지정(10)에는 "="는 필요하지 않습니다.
X1=10	수치 확장(1)이 있는 어드레스 (X) 에의 값 지정(10)에는 "=" 는 필요합니다.
X=10*(5+SIN(37.5))	수치 확장을 사용한 값 지정에는 "="가 필요합니다.

참고

값을 가진 어드레스 문자와 수치 확장의 어드레스를 구분하기 위해 수치 확장 뒤에는 반드시 특수 문자기호 "=", "(", "[", ")", "]", "," 중 하나 또는 운영자가 따르게 됩니다.

3.2.2.4 주석

NC 프로그램을 쉽게 이해하기 위해 코멘트가 NC 블록에 추가될 수 있습니다.

코멘트는 블록 끝에 위치하고 NC 블록의 프로그램 부분과는 세미콜론(";")으로 구분됩니다.

예 1:

프로그램 코드	코멘트
N10 G1 F100 X10 Y20	; NC 블록을 설명하는 코멘트

예 2:

프로그램 코드	코멘트
N10	; G&S 사, 주문 번호 12A71
N20	; 1994년 11월 21일, TV 4 부서에서 H. Smith가 작성한 프로그램
N50	; 섹션 번호 12, 수중 펌프 유형 TP23A용 하우징

참고

프로그램이 가동 중일 때 코멘트는 보관되고 현 블록 디스플레이에 나타납니다.

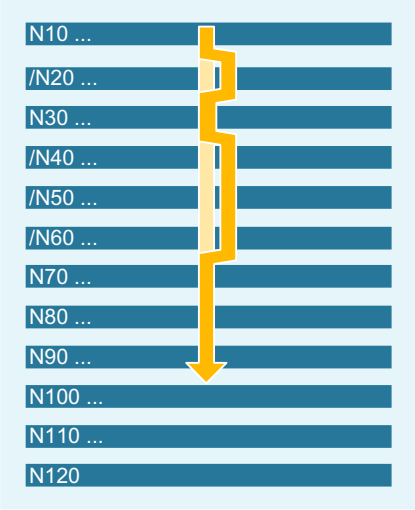
3.2.2.5 블럭 건너뛰기

프로그램이 실행될 때마다 실행되지 않는 NC 블록은 건너뛰고 처리하지 않을 수 있습니다. 이 기능은 테스트 및/또는 새 프로그램의 실행 시 사용됩니다.

블록 건너뛰기

건너뛸 블록의 경우 가공 프로그램에서 블록 번호 앞에 '/' 기호가 있습니다. 몇 개의 연속되는 블록 역시 건너뛸 수 있습니다. 건너뛴 블록의 명령문은 실행되지 않으며, 프로그램이 건너뛰지 않은 다음 블록을 계속 실행합니다.

예:



프로그램 코드	설명
N10 …	; 실행됨
/N20 ...	; 건너뜀
N30 …	; 실행됨
/N40 ...	; 건너뜀
/N50 …	; 건너뜀
/N60 …	; 건너뜀
N70 …	; 실행됨
...	

건너뛰기 레벨

블록에는 건너뛰기 레벨(최대 10)이 지정될 수 있고 사용자 인터페이스 또는 PLC 사용자 프로그램을 통해 활성화될 수 있습니다.

NC 프로그램에서 슬래시와 건너뛰기 레벨 번호를 사용해 지정합니다. 각 블록에는 하나의 건너뛰기 레벨만이 명시될 수 있습니다.

예:

프로그램 코드	설명
/ ...	; 블록을 건너뜀(건너뛰기 레벨 1)
/0 ...	; 블록을 건너뜀(건너뛰기 레벨 1)
/1 N010...	; 블록을 건너뜀(건너뛰기 레벨 2)
/2 N020...	; 블록을 건너뜀(건너뛰기 레벨 3)
...	
/7 N100...	; 블록을 건너뜀(건너뛰기 레벨 8)
/8 N080...	; 블록을 건너뜀(건너뛰기 레벨 9)
/9 N090...	; 블록을 건너뜀(건너뛰기 레벨 10)

참고

건너뛸 레벨은 제어 시스템이 STOP/리셋 상태일 때만 변경할 수 있습니다.

참고

사용할 수 있는 건너뛰기 레벨 번호는 디스플레이 머신 데이터에 따라 다릅니다.

참고

블록 건너뛰기는 블록 검색 중에도 활성 상태로 유지됩니다.

참고

프로그램 실행을 제어하기 위해 시스템과 사용자 변수 역시 조건적 점프에 사용될 수 있습니다.

3.3 NC 프로그램 생성

3.3.1 기본 절차

NC 언어로의 개별 운영 단계 프로그램 작성은 일반적으로 NC 프로그램 개발 작업의 작은 부분일 뿐입니다.

운영 단계의 기획과 준비가 실질적인 지시의 프로그램 작성보다 선행되어야 합니다. NC 프로그램을 구축하고 조직할 방법을 미리 더 정확하게 기획할수록 더욱 쉽고 빠르게 오류가 없는 완전한 프로그램을 만들 수 있게 됩니다. 깔끔하게 구축된 프로그램은 이후 변경이 필요할 때 특히 유익합니다.

모든 부분이 동일하지 않기 때문에 모든 프로그램을 동일한 방법으로 생성한다는 것은 생각할 수 없습니다. 그러나 다음 절차는 그 자체가 대부분의 경우에 적합하다고 판명되었습니다.

절차

1. **공작물 도면 준비**
   - 공작물 영점을 지정하시오
   - 좌표계를 도면으로 나타내시오.
   - 누락된 모든 좌표를 계산하십시오.
2. **기계 가공 시퀀스 지정**
   - 어떤 형상의 기계 가공을 위해 언제 어떤 공구를 사용할 것인가?
   - 어떤 순서로 공작물의 개별 요소를 기계 가공할 것인가?
   - 어떤 개별 요소가 반복되고(또한 회전될 수 있는지) 서브루틴에 보관되어야 하는가?
   - 현 공작물에 사용될 수 있는 형상 부분이 다른 가상 프로그램이나 서브루틴에 있는가?
   - 유용하거나 필요한(프레임 개념) 제로 옵셋, 회전, 대칭 및 눈금 측정은 어디 있는가?
3. **기계 기획 창출**
   모든 기계 가공 운영을 단계별로 지정하시오. 예,
   - 위치 확정의 급이송 운동
   - 공구 교환
   - 기계 가공 면 지정
   - 점검을 위한 후진
   - 스핀들 교체, 절삭유 켜고/끔
   - 공구 데이터 호출
   - 주입
   - 경로 보정
   - 형상에의 접근
   - 형상에서 후진
   - 등
4. **프로그램 작성 언어로의 기계 가공 단계 컴파일**
   - NC 블럭과 같이 각 개별 단계를 작성하시오.
5. **프로그램으로의 개별 단계 결합**

3.3.2 유효한 문자

NC 프로그램 작성에 다음 문자가 유효합니다.

- 대문자:
  A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W X, Y, Z
- 소문자:
  a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

- 숫자:
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- 특수 문자
  아래의 도표를 참조하세요.

특수 문자	의미
%	프로그램 시작 문자(외부 PC에 프로그램을 작성할 때만 사용됨)
(	변수 또는 표현을 괄호안에 넣을 때
)	변수 또는 표현을 괄호안에 넣을 때
[	어드레스 또는 색인을 괄호안에 넣을 때
]	어드레스 또는 색인을 괄호안에 넣을 때
<	보다 작음
>	보다 큼
:	기본 블록, 레이블의 끝, 연쇄 운영자
=	할당, 등식 연산자
/	나누기, 블록 숨기기
*	곱셈
+	덧셈
-	빼기, 빼기 기호
"	이중 따옴표, 문자 열의 식별자
'	단일 따옴표, 특수 숫자 값의 식별자: 16진법, 이진법
$	시스템 변수 식별자
s_	밑줄, 글자에 속함
?	예비용
!	예비용
.	소수점
,	쉼표, 변수 구분자
;	주석 시작
&	포맷 문자, 공백 문자와 동일한 효과
LF	블록 끝
탭 문자	구분자
소재	구분자(공백)

참고

글자 "O"와 숫자 "0"과의 구분을 주의하시오.

참고

대문자와 소문자는 구분하지 않습니다 (예외: 공구 호출).

참고

인쇄되지 않는 특수 문자는 공백으로 취급합니다.

3.3.3 프로그램 헤더

공작물 형상을 가공하기 위해 실제 모션 블록 앞에 위치하는 NC 블록을 프로그램 헤더라고 합니다.

프로그램 헤더에는 다음에 관한 정보/성명이 포함되어 있습니다:

- 공구 교환
- 공구 옵셋
- 스핀들 운동
- 주입 제어
- 기하 설정 (워크 옵셋, 작업 평면 선택)

선삭의 프로그램 헤더

다음 예에서 선삭용 NC 프로그램 헤더의 일반적인 구조를 알 수 있습니다.

프로그램 코드	코멘트
N10 G0 G153 X200 Z500 T0 D0	; 공구 터렛이 회전하기 전에 공구 홀더를 후퇴시킵니다.
N20 T5	; 공구 5로 회전시킵니다.
N30 D1	; 공구의 절삭날 데이터 세트를 활성화합니다.
N40 G96 S300 LIMS=3000 M4 M8	; 주속 일정 제어 속도 (Vc) = 300 m/min, 속도 제한 = 3000 rpm, 회전 방향 CCW, 냉각 작동.
N50 DIAMON	; X축이 직경으로 프로그램됩니다.
N60 G54 G18 G0 X82 Z0.2	; 워크 옵셋과 작업 평면을 호출하고 시작 위치로 접근합니다.
...	

밀링의 프로그램 헤더

다음 예에서 밀링용 NC 프로그램 헤더의 일반적인 구조를 알 수 있습니다.

프로그램 코드	코멘트
N10 T="SF12"	; 또는: T123
N20 M6	; 공구를 바꿉니다.
N30 D1	; 공구의 절삭날 데이터 세트를 활성화합니다.
N40 G54 G17	; 워크 옵셋과 작업 평면.
N50 G0 X0 Y0 Z2 S2000 M3 M8	; 공작물에 접근, 스핀들 및 냉각 작동.
...	

공구 방향/좌표 변형이 사용 중이라면 프로그램 시작할 때 아직 활성화된 모든 변형은 삭제되어야 합니다.

프로그램 코드	코멘트
N10 CYCLE800()	; 스위블된 평면의 재설정
N20 TRAFOOF	; TRAORI, TRANSMIT, TRACYL, ...의 재설정
...	

3.3.4 프로그램 예

3.3.4.1 예 1: 첫번째 프로그램 작성 단계

프로그램 예 1은 첫 번째 프로그래밍 단계를 시험하고 수행하는 데 사용됩니다.

작업 순서

1. 새 가상 프로그램 생성(이름)
2. 가공 프로그램 편집
3. 가공 프로그램 선택
4. 단일 블록 활성화
5. 가공 프로그램을 시작합니다.

참고

기계에서 프로그램이 실행되려면 머신 데이터가 적절하게 설정되어야 합니다(→ 기계 제조업체 데이터 참고).

참고

프로그램을 테스트할 때 알람이 울리면 먼저 리셋해야 합니다.

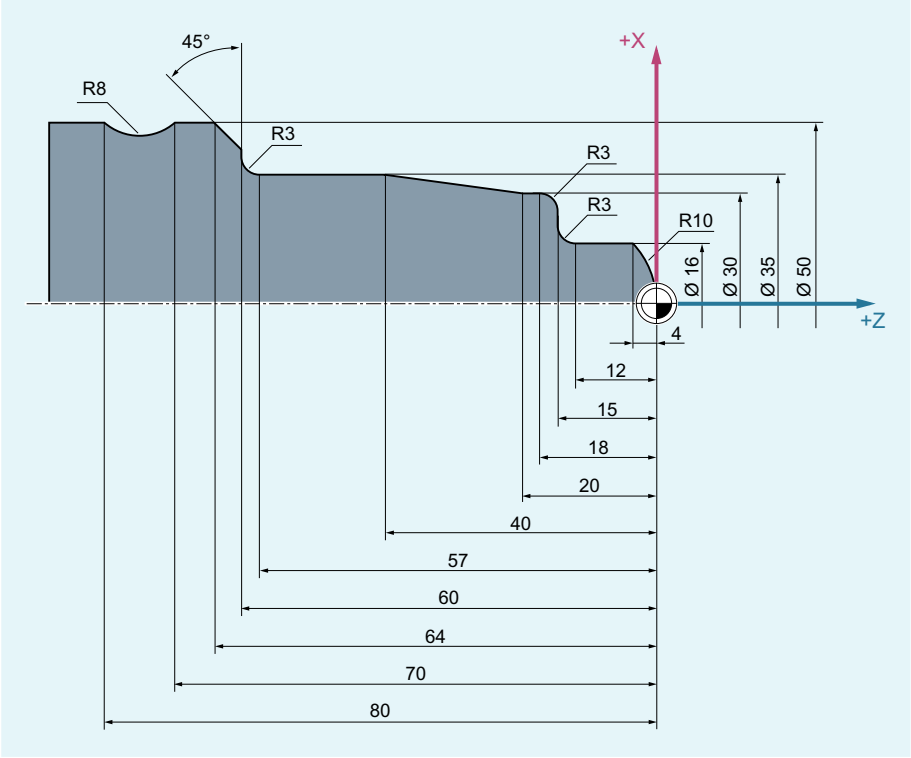
NC 프로그램

프로그램 코드	설명
N10 MSG('이는 나의 프로그램이다')	; 경고 라인에 'THIS IS MY NC PROGRAM'이라는 메시지가 표시됩니다.
N20 F200 S900 T1 D2 M3	; 이송 속도, 스핀들, 공구, 공구 옵셋, CW 방향 스핀들.
N30 G0 X100 Y100	; 급 이송을 통해 위치에 접근합니다.
N40 G1 X150	; 이송 속도로 직사각형, X축으로 직선.
N50 Y120	; Y축으로 직선.
N60 X100	; X축으로 직선.
N70 Y100	; Y축으로 직선.
N80 G0 X0 Y0	; 급 이송으로 후퇴.
N100 M30	; 블록 종료.

3.3.4.2 예 2: 선삭의 NC 프로그램

프로그램 예 2는 선반 위 공작물의 기계 가공에 대한 것입니다. 반경 프로그램 작성과 공구 반경 보상을 포함하고 있습니다.

공작물 치수 도면



NC 프로그램

프로그램 코드	설명
N5 G0 G53 X280 Z380 D0	; 시작점.
N10 TRANS X0 Z250	; 워크 옵셋.
N15 LIMS=4000	; 속도 제한 (G96).
N20 G96 S250 M3	; 일정 정삭 속도를 선택합니다.
N25 G90 T1 D1 M8	; 공구와 옵셋을 선택합니다.
N30 G0 G42 X-1.5 Z1	; 공구 반경 보정을 통해 공구를 설정합니다.
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	; 반경 10 회전.
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	; 반경 3 회전.
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	; 반경 3 회전.
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	; 반경 3 회전.
N85 G1 X46	

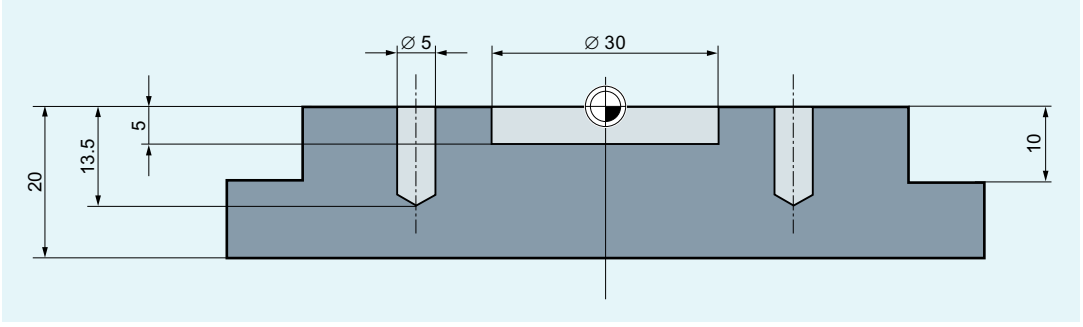
프로그램 코드	설명
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	; 공구 반경 보정을 선택 해제하고 공구 교체 위치에 접근합니다.
N100 T2 D2	; 공구를 호출하고 옵셋을 선택합니다.
N105 G96 S210 M3	; 일정 정삭 속도를 선택합니다.
N110 G0 G42 X50 Z-60 M8	; 공구 반경 보정을 통해 공구를 설정합니다.
N115 G1 Z-70 F0.12	; 직경 50 회전.
N120 G2 X50 Z-80 I6.245 K-5	; 반경 8 회전.
N125 G0 G40 X100 Z50 M9	; 공구를 후퇴시키고 공구 반경 보정을 선택 해제합니다.
N130 G0 G53 X280 Z380 D0 M5	; 공구 교체 위치에 접근합니다.
N135 M30	; 프로그램 종료

3.3.4.3 예 3: 밀링의 NC 프로그램

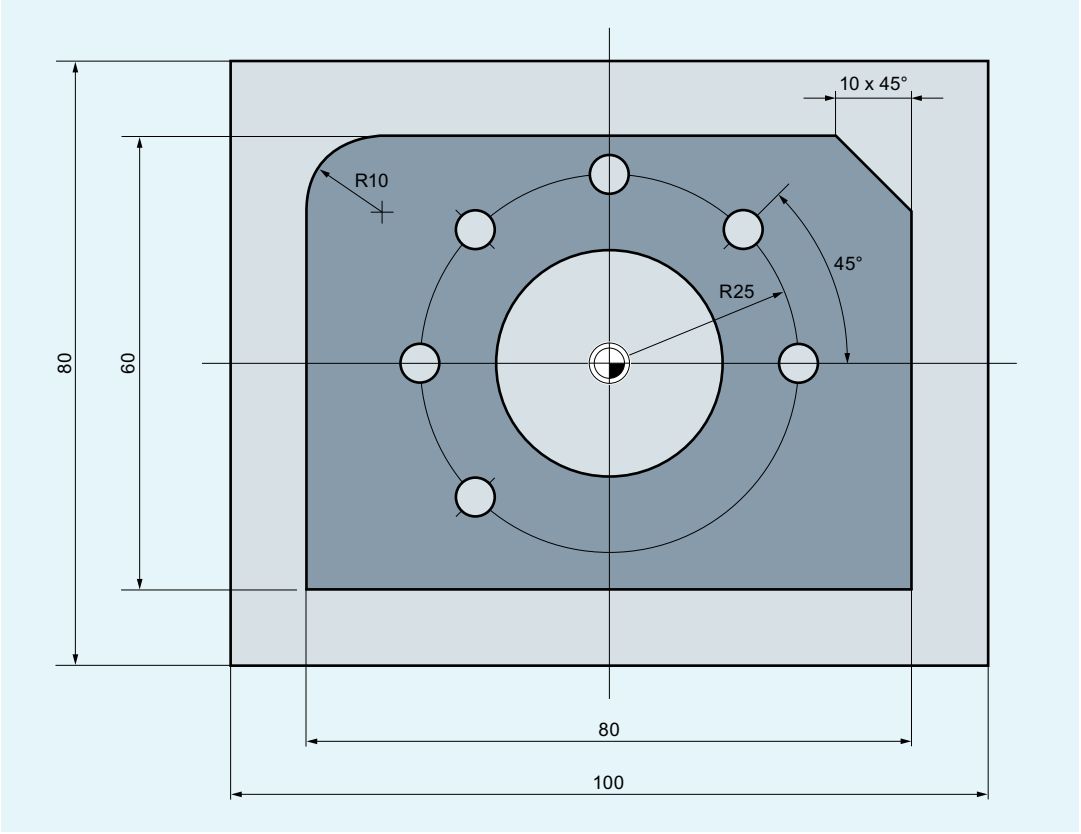
프로그램 예 3는 수직 밀링 기계 위의 공작물 기계 가공에 대한 것입니다. 표면과 측면은 물론 천공을 포함하고 있습니다.

공작물 치수 도면

측면 보기



평면 보기



NC 프로그램

프로그램 코드	설명
N10 T='PF60'	; PF60이라는 이름의 공구를 사전 선택.
N20 M6	; 공구를 스핀들에 장착합니다.
N30 S2000 M3 M8	; 속도, 회전 방향, 냉각 작동.
N40 G90 G64 G54 G17 G0 X-72 Y-72	; 형상 및 시작점 접근에 대한 기본 설정.
N50 G0 Z2	; 안전 공차에서의 z축
N60 G450 CFTCP	; G41/G42 활성화 시의 동작.
N70 G1 Z-10 F3000	; 밀링 공구에서 가공 깊이까지의 이송 속도 = 3000 mm/min
N80 G1 G41 X-40	; 밀링 공구 반경 보정 활성화.
N90 G1 X-40 Y30 RND=10 F1200	; 이송 속도 = 1200 mm/min로 형상으로 이동.
N100 G1 X40 Y30 CHR=10	
N110 G1 X40 Y-30	
N120 G1 X-41 Y-30	
N130 G1 G40 Y-72 F3000	; 밀링 공구 반경 보정 선택 해제.
N140 G0 Z200 M5 M9	; 밀링 공구의 후퇴, 스핀들 + 냉각 작동 해제.
N150 T='SF10'	; SF10이라는 이름의 공구를 사전 선택.

프로그램 코드	설명
N160 M6	; 공구를 스핀들에 장착합니다.
N170 S2800 M3 M8	; 속도, 회전 방향, 냉각 작동.
N180 G90 G64 G54 G17 G0 X0 Y0	; 형상 및 시작점 접근에 대한 기본 설정.
N190 G0 Z2	
N200 POCKET4(2,0,1,-5,15,0,0,0,0,0,800,1300,0,21,5,,,2,0.5)	; 포켓 밀링 주기 호출
N210 G0 Z200 M5 M9	; 밀링 공구의 후퇴, 스핀들 + 냉각 작동 해제.
N220 T='ZB6'	; 6 mm 중앙 드릴 호출
N230 M6	
N240 S5000 M3 M8	
N250 G90 G60 G54 G17 X25 Y0	; 정확한 포지셔닝을 위한 정위치 정지 G60.
N260 G0 Z2	
N270 MCALL CYCLE82(2,0,1,-2.6,,0)	; 드릴링 싸이클의 모달 호출.
N280 POSITION:	; 반복을 위한 점프 표시.
N290 HOLES2(0,0,25,0,45,6)	; 드릴 작업의 위치 패턴.
N300 ENDLABEL:	; 한계 지점 반복
N310 MCALL	; 모달 호출 리셋
N320 G0 Z200 M5 M9	
N330 T='SPB5'	; D 5 mm 드릴 호출
N340 M6	
N350 S2600 M3 M8	
N360 G90 G60 G54 G17 X25 Y0	
N370 MCALL CYCLE82(2,0,1,-13.5,,0)	; 드릴링 싸이클의 모달 호출.
N380 반복 위치	; 중앙으로부터 위치 설명 반복.
N390 MCALL	; 드릴링 싸이클 재설정.
N400 G0 Z200 M5 M9	
N410 M30	; 프로그램 종료.

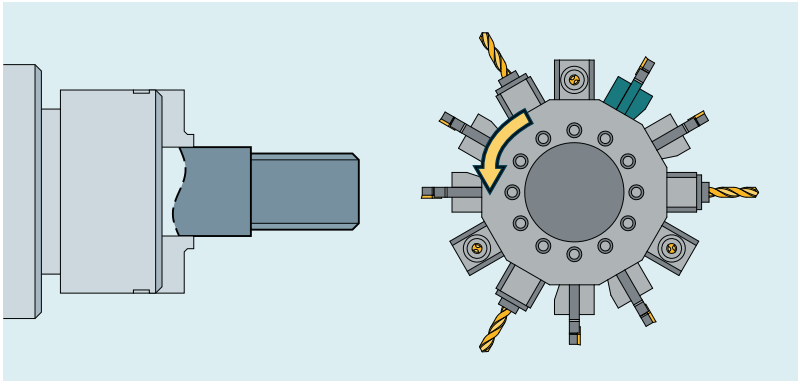
3.4 공구 교환

공구 교환 방법

참고

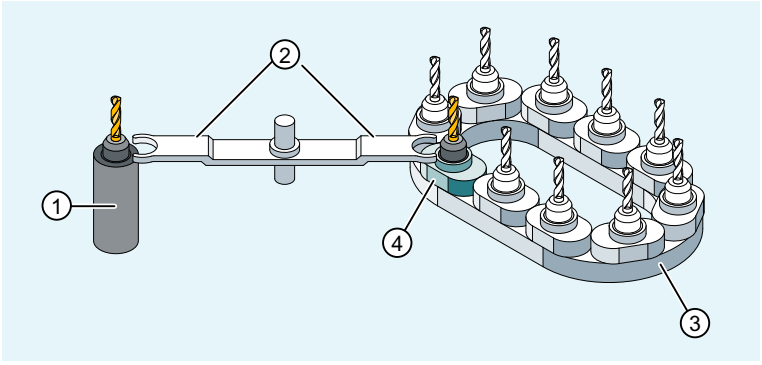
기계 OEM이 스타트업 중에 공구 교환의 유형을 지정합니다.

공구 터렛이 있는 선반의 공구 교환



선삭 기계의 터렛 매거진에서는 공구를 찾아 교환하는 공구 교환을 T 명령어로만 호출합니다.

체인, 회전 판 또는 평면 매거진이 있는 공작 기계의 공구 교환



① 스핀들
② 그리퍼
③ 매거진(본 자료에서는 체인 매거진)
④ 스핀들 위치 변경

체인, 회전 판 및 평면 매거진에서 공구 교환은 일반적으로 두 단계로 진행됩니다.

1. 공구는 매거진에서 T 명령어로 선택됩니다.
2. 이때 공구는 M 명령어로 스핀들에 적재됩니다.

공구 관리가 활성화/비활성화된 경우의 공구 교환 프로그래밍

공구 관리가 활성화/비활성화된 경우의 공구 교환 프로그래밍은 공구 관리가 비활성화된 상태의 공구 교환 프로그래밍과 다릅니다. 따라서 두 경우 모두 설명합니다.

→ 공구 관리가 활성화된 상태의 공구 교환 (페이지 67)

→ 공구 관리가 비활성화된 상태의 공구 교환 (페이지 71)

작업 평면 프로그래밍

공구를 교체하려면 적절한 가공 평면 (페이지 38)을 프로그래밍해야 합니다(최초 상태: G18). 공구 길이 보상이 맞는 축에 지정됨을 보장합니다.

공구 옵셋 활성화

공구 교환을 통해 D 번호 (페이지 88)에 저장된 공구 옵셋 값이 활성화됩니다.

3.4.1 공구 관리가 활성화된 상태의 공구 교환

공구 관리

"공구 관리" 기능은 주어진 시간에 올바른 공구가 올바른 위치에 있으며 공구에 지정된 데이타가 최신 상태임을 보장합니다. 또한 빠른 공구 교환을 가능하게 하고, 공구 수명을 모니터링하여 폐기를 줄이고 교체 공구를 고려하여 기계 가동 중단을 최소화합니다.

공구 이름

활성화된 공구 관리의 기계 공구에서 명확한 식별을 위해 공구의 이름과 번호가 지정되어야 합니다(예: '드릴', '3').

이때 공구 이름으로 공구가 호출될 수 있습니다. 예:

```
T='Drill'
```

참고

공구 이름에는 특수 문자기호가 들어가서는 안됩니다.

3.4.1.1 T=위치 기능을 사용한 공구 관리가 활성화된 상태의 공구 교환

T=위치 기능이 프로그램된 경우 직접 공구 교환이 일어납니다.

적용

터렛을 사용한 선반.

구문

공구 선택
```
T=<No>
T=<Name>
T<n>=<No>
T<n>=<Name>
```

공구 선택 해제
```
T0
```

의미

T=:	공구 교체 및 공구 옵셋 활성화를 포함하는 공구 선택 주소 다음과 같은 값을 지정할 수 있습니다.	
	<No>:	매거진 포켓 번호
	<name>:	공구 이름 **참고:** 공구 이름을 프로그래밍할 때 반드시 정확한 표기법 (대/소문자) 사용해야 합니다.
<n>:	주소 확장자로 스핀들 번호 **참고:** 스핀들 번호를 주소 확장자로 프로그래밍할 수 있는지 여부는 기계 설정에 의해 결정됩니다(→ 장비 제조업체의 사양 참조).	
T0:	공구 선택 해제(매거진 위치가 점유되지 않음)	

참고

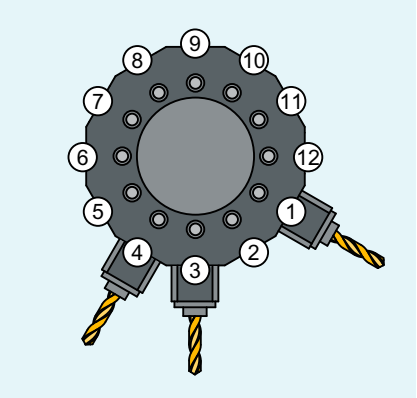
만약 선택된 매거진 포켓이 비어있을 경우, 해당 명령은 T0와 같이 동작합니다. 점유되지 않은 매거진 위치를 선택하여 빈 위치를 지정할 수 있습니다.

예

원형 매거진에는 다음과 같은 할당이 적용된 1~12 위치가 있습니다.

포켓	공구	내부 T 번호	상태
1	Drill, duplo no. = 1	T15	디스에이블
2	사용 안 함		
3	Drill, duplo no. = 2	T10	활성화됨

포켓	공구	내부 T 번호	상태
4	Drill, duplo no. = 3	T1	활성
5 ... 12	사용 안 함		



① ... 매거진/포켓 번호
⑫

다음과 같이 공구 호출을 프로그래밍 합니다

```
N10 T=1
```

다음과 같이 호출이 이루어 집니다.

1. 매거진 포켓 1에서 공구 식별자(공구 번호)가 결정됩니다.
2. 공구 관리에서 해당 공구가 차단되어 사용할 수 없다고 확인됩니다.
3. 검색 방법 설정에 따라 공구 이름이 'drill'인 공구(T='drill')를 검색합니다.
   '활성화된 공구를 찾거나, 다음으로 높은 듀플로 번호를 선택합니다.'
4. 그러면 다음과 같이 사용 가능한 공구가 발견됩니다.
   'Drill', duplo no. 3 (매거진 포켓 4)
   공구 선택 과정이 완료되고 공구 교체가 시작됩니다..

참고

검색 방법으로 '그룹에서 첫 번째로 사용 가능한 공구 선택'을 선택하면 그룹의 순서가 정의되지 않습니다. 이 경우 T10 **또는** T1이 선택됩니다.

검색 방법으로 "상태가 '활성'인 공구를 먼저 선택'을 선택하면 T1이 로드됩니다.

3.4.1.2 M6을 사용한 공구 관리가 활성화된 상태의 공구 교환

공구는 T 명령어가 프로그램 될 때 선택됩니다. 공구는 M6 와 함께 할 때에만 활성화됩니다 (공구 옵셋 포함).

적용

체인, 회전 판 또는 평면 매거진이 있는 공작 기계.

구문

공구 선택
```
T=<No>
T=<Name>
T<n>=<No>
T<n>=<Name>
```

공구 교체
```
M6
```

공구 선택 해제
```
T0
```

의미

T=:	공구 선택 주소 다음과 같은 값을 지정할 수 있습니다.	
	<No>:	매거진 포켓 번호
	<name>:	공구 이름 **참고:** 공구 이름을 프로그래밍할 때 반드시 정확한 표기법 (대/소문자) 사용해야 합니다.
<n>:	주소 확장자로 스핀들 번호 **참고:** 스핀들 번호를 주소 확장자로 프로그래밍할 수 있는지 여부는 기계 설정에 의해 결정됩니다(→ 장비 제조업체의 사양 참조).	
M6:	공구 교체 M 코드 (DIN 66025를 따름) M6에 의해 선택된 공구 (T...) 및 옵셋 (D...) 이 적용됩니다.	
T0:	공구 선택 해제(매거진 위치가 점유되지 않음)	

예

프로그램 코드	설명
N10 T=1 M6	; 매거진 1 위치에서 공구 장착.
N20 D1	; 공구 길이 보정 선택.

프로그램 코드	설명
N30 G1 X10 ...	; 공구 T=1로 가공.
...	
N70 T='Drill'	; 'Drill'이라는 이름의 공구를 사전 선택.
N80 ...	; 공구 T=1로 가공.
...	
N100 M6	; 공구 T="Drill" 장착.
N140 D1 G1 X10 ...	; 공구 T="Drill"로 가공.
...	

3.4.2 공구 관리가 비활성화된 상태의 공구 교환

3.4.2.1 T 번호를 사용한 공구 관리가 비활성화된 상태의 공구 교환

T 번호가 프로그램된 경우 직접 공구 교환이 일어납니다.

적용

터렛을 사용한 선반.

구문

공구 선택
T<No>
T=<No>
T<n>=<No>

공구 선택 해제
T0
T0=<No>

의미

T:	공구 교체 및 공구 옵셋 활성화를 포함하는 공구 선택 주소	
<No>:	공구 번호	
	값 범위:	0 ... 32000

<n>:	주소 확장자로 스핀들 번호 **참고:** 스핀들 번호를 주소 확장자로 프로그래밍할 수 있는지 여부는 기계 설정에 의해 결정됩니다(→ 장비 제조업체의 사양 참조).
T0:	활성 공구 선택 해제

예

프로그램 코드	설명
N10, T1, D1	; 공구 T1 장착 및 공구 옵셋 D1 활성화.
...	
N70 T0	; 공구 T1 선택 해제.
...	

3.4.2.2 M6을 사용한 공구 관리가 비활성화된 상태의 공구 교환

공구는 T 명령어가 프로그램 될 때 선택됩니다. 공구는 M6 와 함께 할 때에만 활성화됩니다 (공구 옵셋 포함).

적용

체인, 회전 판 또는 평면 매거진이 있는 공작 기계.

구문

공구 선택
T<No>
T=<No>
T<n>=<No>

공구 교체
M6

공구 선택 해제
T0
T0=<No>

의미

T:	공구 선택 주소	
<No>:	공구 번호	
	값 범위:	0 ... 32000
<n>:	주소 확장자로 스핀들 번호 **참고:** 스핀들 번호를 주소 확장자로 프로그래밍할 수 있는지 여부는 기계 설정에 의해 결정됩니다(→ 장비 제조업체의 사양 참조).	
M6:	공구 교체 M 코드 (DIN 66025를 따름) M6에 의해 선택된 공구 (T...) 및 옵셋 (D...) 이 적용됩니다.	
T0:	활성 공구 선택 해제	

예

프로그램 코드	**설명**
N10 T1 M6	; 공구 T1 장착.
N20 D1	; 공구 길이 보정 선택.
N30 G1 X10 ...	; T1으로 가공.
...	
N70 T5	; 공구 T5의 사전 선택.
N80 ...	; T1으로 가공.
...	
N100 M6	; 공구 T5 장착.
N110 D1 G1 X10 ...	; 공구 T5로 가공
...	

3.4.3 손상된 T 프로그래밍 동작

손상된 T 프로그래밍 동작은 기계 환경 설정에 따라 다릅니다.

MD22562 TOOL_CHANGE_ERROR_MODE		
Bit	**값**	**의미**
7	0	기본 세팅 T 프로그래밍으로 NC의 T 숫자 인식 여부에 대한 점검이 즉각 이뤄집니다. 그렇지않다면 경고가 촉발됩니다.
	1	프로그램 작성된 T 숫자는 다음의 D 선택으로만 점검됩니다. NC가 공구 숫자를 인식하지 못하면 D 선택 동안 경고가 발생합니다. 예를 들어 공구 프로그래밍 역시 위치 지정을 하려하고 공구 데이타가 반드시 유효할 필요가 없다면 그 반응은 바람직한 것이다(원형 매거진).

3.5 공구 옵셋

3.5.1 프로그래밍된 형상 및 공구 경로

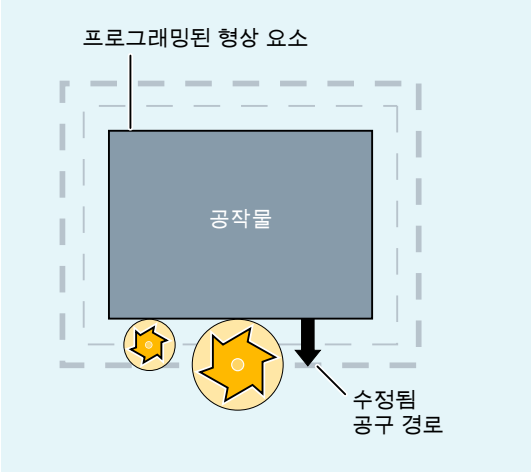
공작물 치수는 직접 프로그램 작성됩니다. 그러므로 밀링 공구 직경, 선삭 공구(시계 반대 방향/시계 방향 선삭 공구)의 절단 날 위치 및 공구 길이 같은 공구 데이터를 프로그램 생성 시 고려하지 않아도 됩니다.

제어 시스템에서 이송 경로를 수정합니다.

공작물 가공 시 공구 경로가 공구 형상에 따라 제어되어 프로그래밍된 형상을 임의의 공구를 사용하여 가공할 수 있습니다.

제어 시스템이 공구 경로를 계산하려면 공구 데이터가 반드시 제어 시스템의 공구 보정 메모리에 입력되어야 합니다. 필요한 공구 (T...)와 필요한 옵셋 데이터 기록(D...)만이 NC 프로그램을 통해 호출됩니다.

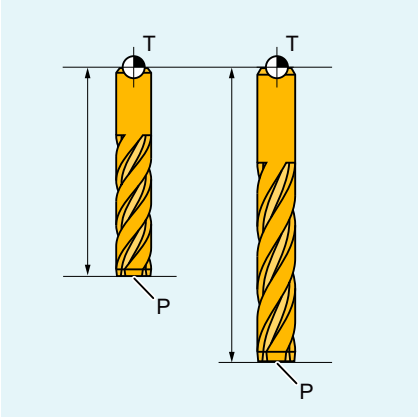
프로그램이 처리되는 동안 제어 시스템이 공구 보정 메모리에서 필요한 옵셋 데이터를 가져와서 각 공구에 대해 공구 경로를 개별적으로 보정합니다.

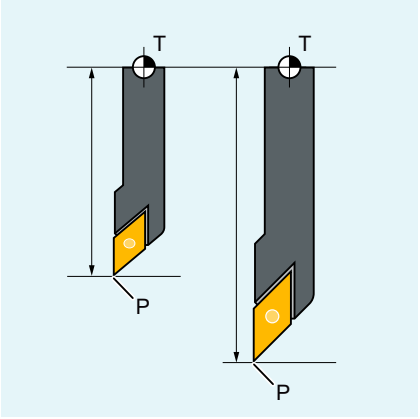


3.5.2 공구 길이 보정

공구 길이 보정은 사용된 공구 간의 길이 차이를 보정합니다.

공구 길이는 공구 캐리어 원점 및 공구 팁 간의 간격을 말합니다.





T 공구 캐리어 원점

P 공구 팁

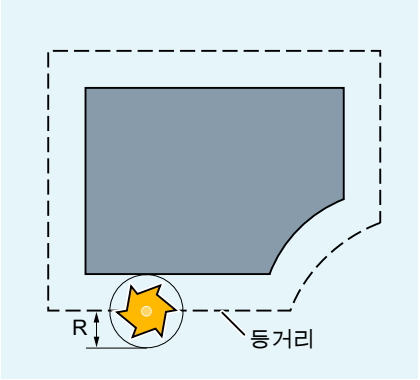
이 길이가 측정된 후 마모 보정값과 함께 시스템의 공구 보정 메모리에 입력됩니다. 이 데이터를 기준으로 이송 동작 및 절입 방향이 계산됩니다.

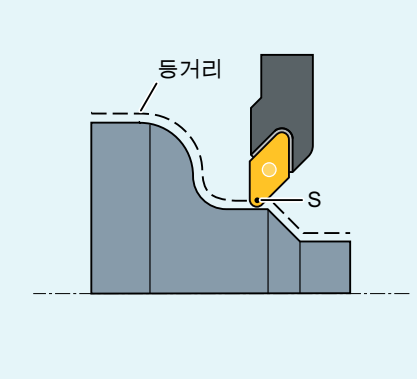
참고

공구 길이의 보정값은 공구의 공간적 방향에 따라 다릅니다.

3.5.3 공구 반경 보정

형상과 공구 경로가 동일하지 않습니다. 밀링 공구 또는 절삭날 중심점은 형상으로부터 늘 같은 거리에 있는 공구 반경에 맞는 경로(공구 중심점 경로)를 따라 이동합니다. 이를 위해서 프로그램을 실행하는 동안 제어기가 프로그램된 공구 중심점 경로를 활성화된 공구의 반경에 따라 이동시킵니다(공구 옵셋 메모리). 이렇게 해서 공구 절삭날이 정확히 프로그램된 형상을 따라 이동할 수 있습니다.





R 공구 반경

S 마지막 위치

공구 반경 보정은 "공구 반경 보정 (페이지 277)" 장에 상세하게 설명되어 있습니다.

다음 또한 참고:

2 1/2 D 공구 옵셋 (CUT2D, CUT2DD, CUT2DF, CUT2DFD) (페이지 317)

3.5.4 공구 보정 메모리

각 절삭날을 위한 제어 시스템의 공구 옵셋 메모리에 다음과 같은 데이터가 있어야 합니다.

- 공구 종류
- 절삭날 위치
- 공구 도형 변수 (길이, 반경)

이 데이터는 공구 변수(최대 25) 로 입력됩니다. 공구에 필요한 변수는 공구의 종류에 따라 다릅니다. 필요하지 않은 모든 공구 변수는 반드시 '0'으로 설정되어야 합니다(시스템의 기본 설정과 일치합니다)

참고

옵셋 메모리에 입력된 값은 각 공구 호출 시 처리에 포함됩니다.

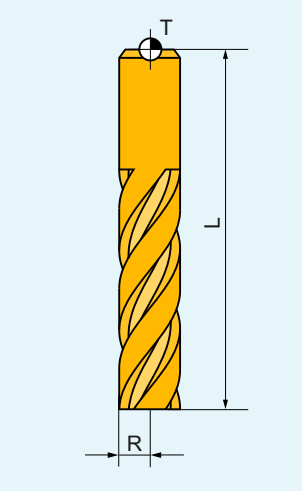
공구 종류

공구 유형(드릴, 밀링 또는 선삭 공구)은 어느 기하 데이터가 필요한지 어떻게 계산하는지 결정합니다.

절삭날 위치

절삭날 위치는 절삭날 중심점에 대한 공구 팁 P의 상대적 위치를 설명합니다. 선삭 공구 (페이지 84)(공구 유형 5xx)의 공구 반경 보정을 계산하려면 절삭날 위치와 절삭날 반경이 필요합니다.

공구 도형 변수 (길이, 반경)



T 공구 캐리어 원점

R 공구 반경

L 공구 길이

공구 도형 변수에는 몇 가지 구성 요소(도형, 마모)로 구성됩니다. 제어 시스템에서 이러한 구성 요소의 특정 치수 (예: 전체 길이 1, 총 반경) 를 계산합니다. 해당 전체 치수는 옵셋 메모리가 활성화될 때 적용됩니다.

이러한 값이 축에서 계산되는 방법은 공구 종류와 현재 면 (G17/G18/G19)을 통해 결정됩니다.

3.5.5 공구 종류

3.5.5.1 공구 유형 번호 및 공구 그룹

각 공구 종류에는 3 단위의 숫자가 할당됩니다. 다음 기술 중 하나 또는 공구 그룹에 공구를 할당할 때 첫 번째 숫자(백의 자리)를 사용합니다.

공구 종류	공구 그룹
1xy	밀링 공구 (페이지 78)
2xy	드릴 (페이지 81)
3xy	예비
4xy	연삭 공구 (페이지 82)
5xy	선삭 공구 (페이지 84)
6xy	예비
7xy	특수 공구 (페이지 86)

3.5.5.2 밀링 공구

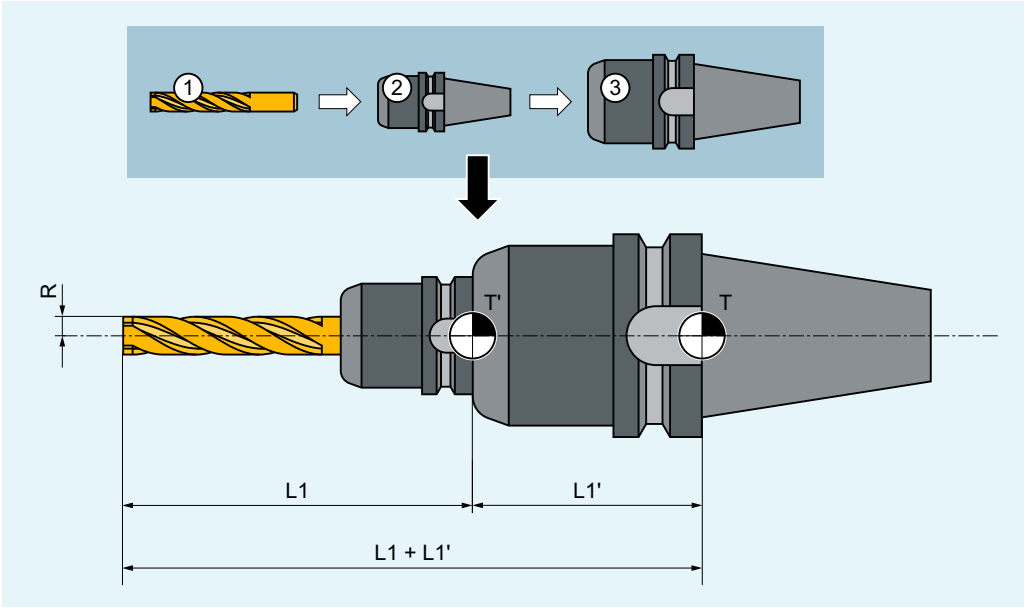
'밀링 공구' 그룹에는 다음 종류의 공구가 있습니다.

100	CLDATA (커터 위치 데이터)에 따른 밀링 공구
110	볼 엔드 밀
111	원통형 다이 싱크 밀링 커터
120	코너 라운딩이 없는 엔드 밀링 커터
121	코너 라운딩 엔드 밀
130	코너 라운딩이 없는 앵글 헤드 커터
131	코너 라운딩이 있는 앵글 헤드 밀
140	페이싱 공구

145	나사 커터
150	측면 밀링
151	톱
155	베벨 커터(코너 라운딩 없음)
156	코너 라운딩 베벨 커터
157	끝이 좁아지는 다이 밀링 공구
160	드릴과 나사 밀링 커터

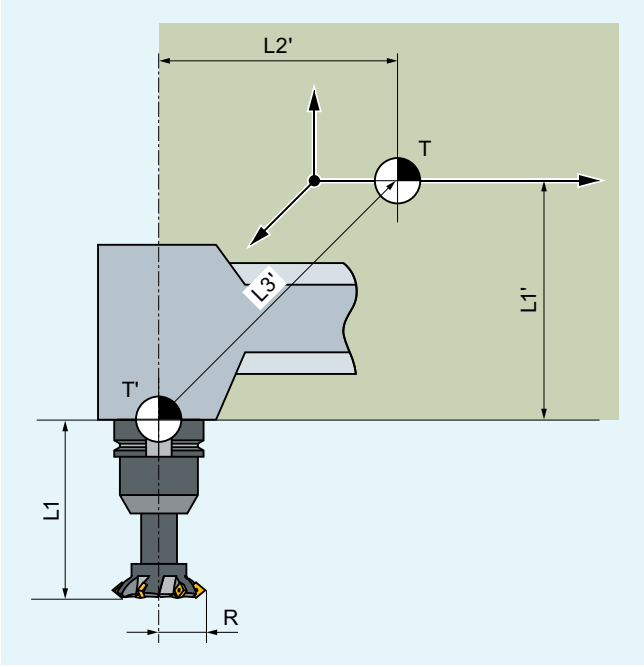
공구 파라미터

아래의 다이어그램에는 보정 메모리에 입력된 밀링 공구 파라미터의 개요가 나와 있습니다.



① 공구
② 공구 홀더
③ 공구 어댑터
T 어댑터 기준점(삽입된 공구 = 공구 캐리어 기준점)
T' 공구 캐리어 원점
L1 기하 - 길이 1
L1' 어댑터 치수 - 길이 1
L1 + L1' 총 길이 L1
R 반경

공구 파라미터	의미
$TC_DP1	공구 유형 1xy
$TC_DP3	기하 - 길이 1
$TC_DP6	기하 - 반경
$TC_DP21	어댑터 치수 - 길이 1
• 요구사항을 충족하는 마모 값입니다. • 다른 값은 0으로 설정해야 합니다.	



T 공구 캐리어 원점

T' 공구 캐리어 원점

L1 기하 - 길이 1

R 공구 반경

L1' 베이스 치수 - 길이 1

L2' 베이스 치수 - 길이 2

L3' 베이스 치수 - 길이 3

공구 파라미터	의미
$TC_DP1	공구 종류
$TC_DP3	기하 - 길이 1

공구 파라미터	의미
$TC_DP6	기하 - 반경
$TC_DP21	베이스 치수 - 길이 1
$TC_DP22	베이스 치수 - 길이 2
$TC_DP23	베이스 치수 - 길이 3
• 요구사항을 충족하는 마모 값입니다. • 다른 값은 0으로 설정해야 합니다.	

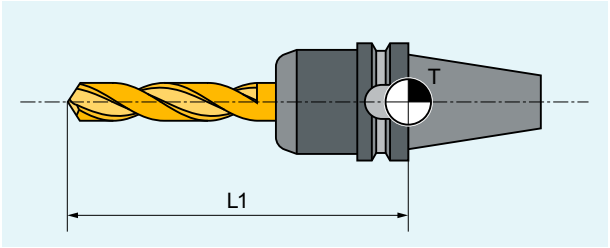
3.5.5.3 드릴

'드릴' 그룹에는 다음 종류의 공구가 있습니다.

번호	공구 종류
200	트위스트 드릴
205	드릴
210	보링 막대
220	중심 드릴
230	카운터싱크
231	카운터보어
240	일반 나사 탭
241	미세 나사 탭
242	Whitworth 나사 탭
250	리머

공구 파라미터

아래의 다이어그램에는 보정 메모리에 입력된 드릴 공구 파라미터의 개요가 나와 있습니다.



T 공구 캐리어 원점

L1 길이 1

공구 파라미터	의미
$TC_DP1	공구 종류
$TC_DP3	기하 - 길이 1
• 요구사항을 충족하는 마모 값입니다. • 다른 값은 0으로 설정해야 합니다.	

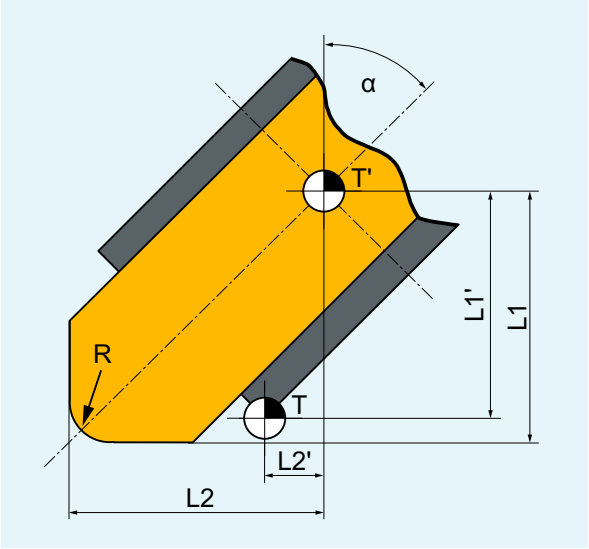
3.5.5.4 연마 공구

'연삭 공구' 그룹에는 다음 종류의 공구가 있습니다.

400	표면 연삭 휠
401	모니터링 기능이 있는 표면 연삭 휠
402	기본 치수가 없는 모니터링 기능이 없는 표면 연삭 휠(TOOLMAN)
403	기본 치수가 없는 모니터링 기능이 있는 연삭 휠 고정 속도 GWPS의 표면 연삭 휠
410	페이싱 휠
411	모니터링 기능이 있는 페이싱 휠(TOOLMAN)
412	모니터링 기능이 없는 페이싱 휠(TOOLMAN)
413	기본 치수가 없는 모니터링 기능이 있는 연삭 휠 고정 속도 GWPS의 페이싱 휠
490	드레서

공구 파라미터

아래의 다이어그램에는 보정 메모리에 입력된 연삭 공구 파라미터의 개요가 나와 있습니다.



T 공구 캐리어 원점

T' 공구 홀더 기준점

L1 기하 - 길이 1

L1' 베이스 치수 - 길이 1

L2 기하 - 길이 2

L2' 베이스 치수 - 길이 2

R 반경

α 경사 휠 각도

절삭날별 파라미터	의미
$TC_DP1	공구 유형 4xy
$TC_DP2	절삭날 위치
$TC_DP3	기하 길이 1
$TC_DP4	기하 길이 2
$TC_DP6	반경
$TC_DP21	베이스 치수 길이 1
$TC_DP22	베이스 치수 길이 2
• 요구사항을 충족하는 마모 값입니다. • 다른 값은 0으로 설정하십시오.	

공구 관련 변수	의미
$TC_TPG1	스핀들 번호
$TC_TPG2	연결 규칙 [1]
$TC_TPG3	최소 휠 반경
$TC_TPG4	최소 휠 너비
$TC_TPG5	실제 휠 너비
$TC_TPG6	최대 속도
$TC_TPG7	최대 원주 속도
$TC_TPG8	경사 휠 각도
$TC_TPG9	반경 계산을 위한 파라미터 번호
$TC_TPG_DRSPATH	드레싱 프로그램으로의 디렉토리 경로
$TC_TPG_DRSPROG	드레싱 프로그램 이름

[1] 왼쪽 및 오른쪽 공구 노즈 반경 보정을 위해 기하 길이 보정, 마모 및 베이스 치수를 연결할 수 있습니다. 왼쪽 절삭날에 대한 길이 보정이 변경되어 오른쪽 절삭날에 대한 값이 자동으로 입력됩니다(반대의 경우도 적용됨).

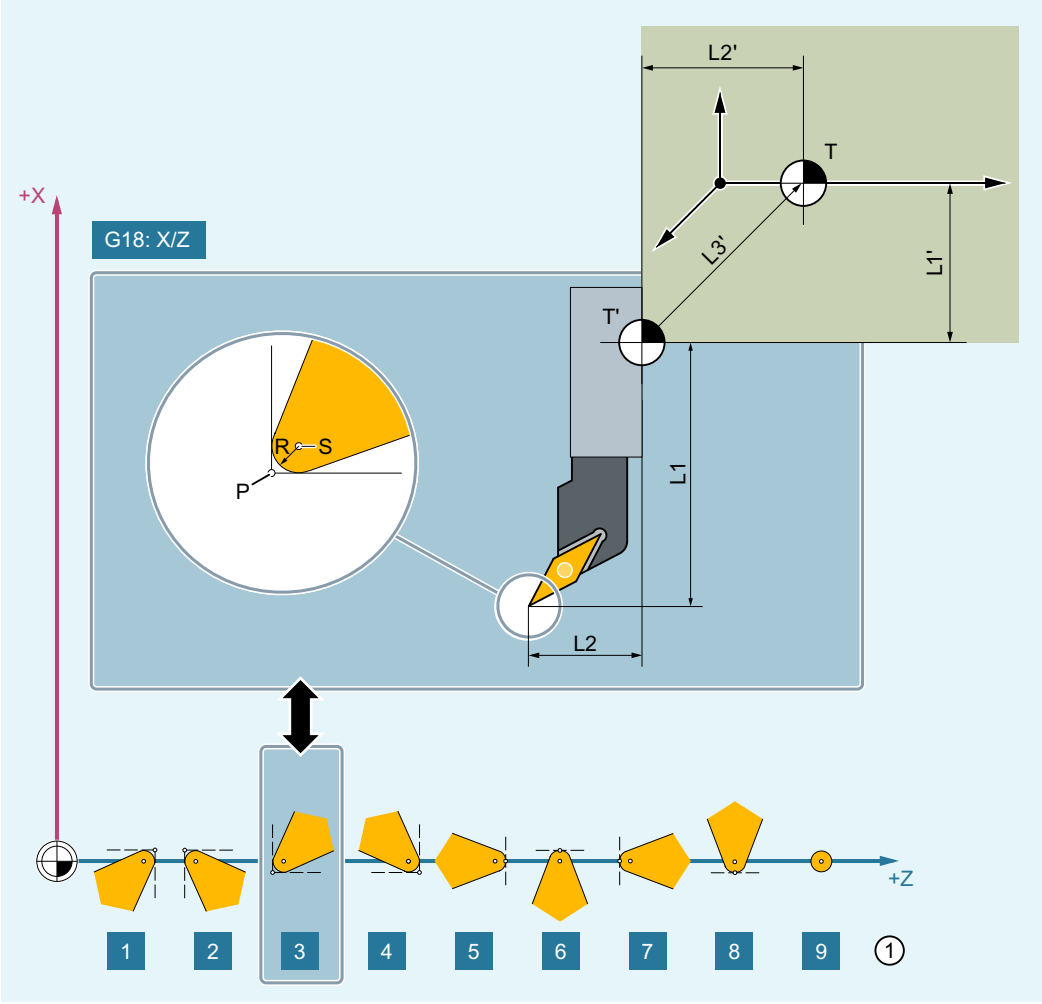
3.5.5.5 선삭 공구

'선삭 공구' 그룹에는 다음 종류의 공구가 있습니다.

500	황삭 공구
510	정삭 공구
520	플런지 커터
530	분리 공구
540	나사 가공 공구
550	버턴 공구/성형 공구(TOOLMAN)
560	회전 드릴 (ECOCUT)
580	절삭날 위치 변수로 탐색

공구 파라미터

아래의 다이어그램에는 보정 메모리에 입력된 선삭 공구 파라미터의 개요가 나와 있습니다.



① 선삭 센터 뒤의 가공을 위한 절삭날 위치(1~9)

P 공구 팁

S 마지막 위치

R 절삭날 반경

T 공구 캐리어 원점

T' 공구 홀더 기준점

L1 기하 - 길이 1

L2 기하 - 길이 2

L1' 베이스 치수 - 길이 1

L2' 베이스 치수 - 길이 2
L3' 베이스 치수 - 길이 3

공구 파라미터	의미
$TC_DP1	공구 종류
$TC_DP2	절삭날 위치
$TC_DP3	기하 - 길이 1
$TC_DP4	기하 - 길이 2
$TC_DP6	기하 - 반경(절삭날 반경)
$TC_DP21	베이스 치수 - 길이 1
$TC_DP22	베이스 치수 - 길이 2
$TC_DP23	베이스 치수 - 길이 3
• 요구사항을 충족하는 마모 값입니다. • 다른 값은 0으로 설정하십시오.	

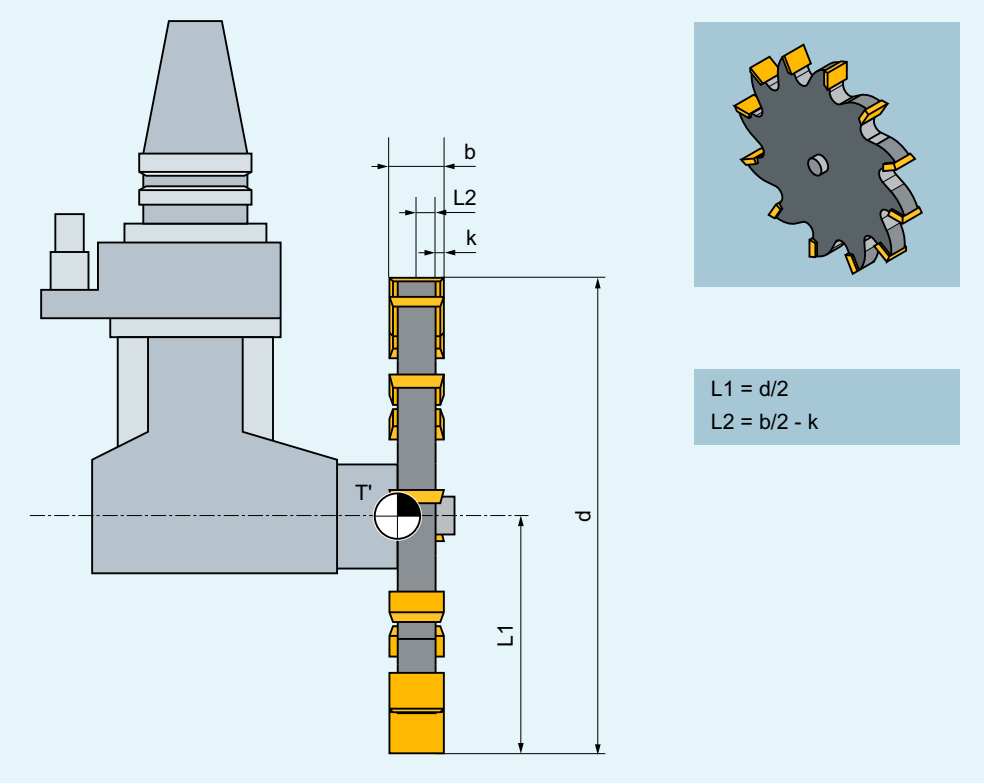
3.5.5.6 특수 공구

'특수 공구' 그룹에는 다음 종류의 공구가 있습니다.

700	슬로팅 톱
710	3D 프로브
711	날 탐색
712	모노 프로브
713	L 프로브
714	스타 프로브
725	계측 공구
730	중지
731	스핀들 슬리브
732	지지구

공구 파라미터

아래의 다이어그램에는 보정 메모리에 입력된 '슬로팅 톱' 공구 유형의 공구 파라미터에 대한 개요가 나와 있습니다.



T' 공구 캐리어 원점
L1 기하 - 길이 1
L2 기하 - 길이 2
d 직경
b 슬롯 너비
k 예상

공구 파라미터	의미
$TC_DP1	공구 종류
$TC_DP3	기하 - 길이 1
$TC_DP4	기하 - 길이 2
$TC_DP6	직경
$TC_DP7	슬롯 너비

공구 파라미터	의미
$TC_DP8	예상
$TC_DP21	베이스 치수 길이 1
$TC_DP22	베이스 치수 길이 2
$TC_DP23	베이스 치수 길이 3
• 요구사항을 충족하는 마모 값입니다. • 다른 값은 0으로 설정해야 합니다.	

3.5.6 공구 옵셋 활성화/비활성화(D, D0):

공구의 절삭날 1~8(활성화된 공구 관리의 경우 12)에 서로 다른 공구 옵셋 데이터 블록을 할당할 수 있습니다(예: 그루빙 공구의 좌측 및 우측 절삭날에 서로 다른 보정값 할당).

특수 절삭날의 옵셋 데이터(공구 길이 보정 데이터 포함)는 D 번호를 호출함으로써 활성화됩니다. D0을 프로그램 하여 공구 옵셋을 해제할 수 있습니다.

공구 반경 보정도 G41/G42를 통해 활성화되어야 합니다.

참고

공구 길이 보정은 D 번호가 프로그래밍되어 있는 경우 즉시 적용됩니다. D 번호가 프로그래밍되지 않은 경우 공구 교체 시 머신 데이터를 통해 기본으로 정의된 설정이 적용됩니다(→ 장비 제조업체의 사양 참조).

구문

```
D<No>
X... Y... Z...
G41/G42 X... Y... Z...
G40
D0
```

의미

D:	활성 공구의 옵셋 데이터 블록을 활성화시키는 주소 프로그램에 의해 공구 길이 보정이 적용되어야 할 축의 이송이 시작되면 공구 길이 보정이 적용됩니다. **주의:** 공구 교체 시 절삭날이 자동으로 활성화되도록 설정된 경우 D 번호를 프로그래밍하지 않고도 공구 길이 보정을 적용할 수 있습니다(→ 장비 제조업체의 사양 참조).
<No>:	활성화될 공구 옵셋 데이터 블록은 D 번호를 통해 지정됩니다. D 프로그래밍 유형은 장비 구성에 따라 다릅니다(→ 장비 제조업체가 제공한 정보 참조). 옵션은 다음과 같습니다. • D 번호 = 절삭날 번호 각 공구(TOOLMAN이 없는 경우 T<No>, TOOLMAN이 있는 경우 T="Name")마다 1부터 12까지 D 번호를 사용할 수 있습니다. 이들 D 번호는 공구 절삭날에 직접 지정됩니다. 각 D 번호(= 절삭날 번호)마다 옵셋 데이터 세트($TC_DPx[<t>,<d>])가 지정됩니다. • D 번호 자유 선택 공구의 절삭날 번호에 자유롭게 D 번호를 할당할 수 있습니다. 머신 데이터를 통해 사용 가능한 D 번호의 최대 개수가 정해집니다.
	값 범위: 0 ... 32000
D0:	활성 공구의 옵셋 데이터 블록 비활성화
G41:	형상의 **좌측** 가공 방향을 적용하여 공구 반경 보정을 활성화하는 명령
G42:	형상의 **우측** 가공 방향을 적용하여 공구 반경 보정을 활성화하는 명령
G40:	공구 반경 보정을 비활성화하는 명령

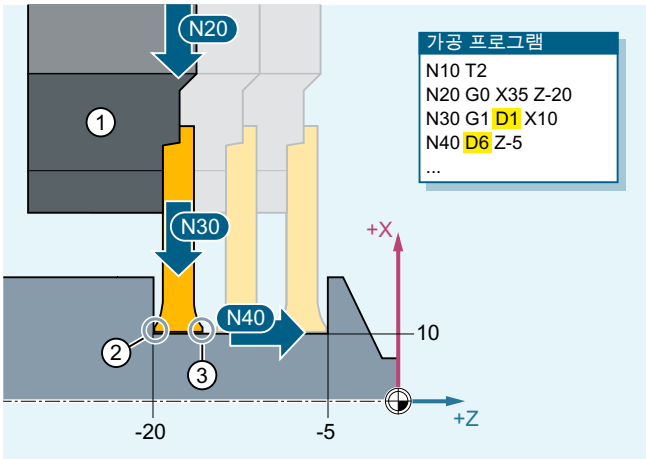
G40/G41/G42에 대한 **추가 정보**는 "공구 반경 보정 (페이지 277)"에서 확인할 수 있습니다.

예

예 1: T 명령을 사용한 공구 교체 (선삭)

프로그램 코드	설명
N10, T1, D1	; 공구 T1 장착 및 T1의 공구 옵셋 데이터 블록 D1 활성화.
N11 G0 X... Z...	; 공구 길이 보정이 적용됩니다.
N50, T4, D2	; 공구 T4 장착 및 T4의 공구 옵셋 데이터 블록 D2 활성화.
...	
N70 G0 Z... D1	; 공구 T4를 위한 다른 절삭날 D1을 활성화합니다.

예 2: 그루빙 공구의 좌측 및 우측 절삭날에 서로 다른 보정값 설정



① 플런지 선삭 공구(T2)
② 절삭날 D1
③ 절삭날 D6

다음 또한 참고:

공구 반경 보정 (페이지 76)

추가 정보

공구 옵셋 데이타의 변경

표준 설정에서 공구 옵셋 데이터의 변경은 그 다음 T 또는 D 번호가 프로그램 작성될 때 효과를 나타냅니다.

다음 기계 데이터는 입력된 공구 옵셋 데이터가 즉시 활성화되는 것을 지정하는데 사용될 수 있습니다.

MD9440 $MM_ACTIVATE_SEL_USER

⚠ 경고

충돌 위험

MD9440 가 설정되면 가공 프로그램 중단 동안 공구 옵셋의 변경으로 인한 공구 옵셋은 가공 프로그램이 계속될 때 적용됩니다.

3.5.7 공구 옵셋 억제(SUPD)

SUPD 명령을 사용해 블록별로 공구 옵셋을 억제할 수 있습니다.

D0으로 공구 옵셋 선택을 취소하면 공구 옵셋 데이터가 SUPD로 유지된다는 장점이 있습니다. 따라서 D 번호를 프로그래밍하여 공구 옵셋 데이터를 다시 활성화할 필요가 없습니다.

SUPD는 공구 길이 보정(예: 공구 길이 대신 영점을 계산에 포함해야 하는 가공 프로그램 블록)을 억제하는 데만 사용됩니다. 공구 반경 보정의 추가 억제를 위해 활성 G41/G42와 함께 사용하지 않는 것이 좋습니다.

구문

```
D<No.>
X... Y... Z...
X... Y... Z... SUPD
...
D0
```

의미

SUPD:	활성 블록에서 활성 공구의 옵셋 데이터 블록을 비활성화하는 명령	
	효율성:	넌모달

제한조건

- SUPD는 선형 블록에서만 사용해야 합니다.
- SUPD는 동기 동작에서 사용할 수 없습니다.
- SUPD는 다음 기능과 함께 사용할 수 없습니다.
  - 3D 페이스 밀링을 위한 3D 공구 반경 보정(CUT3DFxx)
  - 커브 테이블 (CTAB)
- SUPD를 활성 공구 반경 보정(G41/G42)과 함께 사용하는 것이 가능하지만, 권장하지 않습니다.
  그럼에도 불구하고 기능을 활성화하려면 다음 설정 데이터를 "0"으로 설정해야 합니다.
  SD42480 $SC_STOP_CUTCOM_STOPRE = 0
  이렇게 하면 프로그램이 활성 G41/G42로 중단되지 않습니다.

예

이송 시 공구 길이가 서브프로그램 SUB_SUP에서 억제되어야 합니다.

가공 프로그램

프로그램 코드	설명
...	
N300 $P_UIFR[1]=CTRANS(X,1000,Y,400,Z,-120)	
N310 T="BALL_D3"	
N320 M6	
N330 TRAFOOF	
N340 G54 G0 Z49 D1	
N350 G0 X1100 Y500 C0 A0	
N360 **SUB_SUP**	; 서브루틴 호출.
N370 G0 X1200	
N380 M30	

D0을 사용한 서브프로그램

프로그램 코드	설명
N10 PROC **SUB_SUP**	
N20 DEF INT NUMBER	
N30 NUMBER=$P_TOOL	
N40 G0 Z49 **D0**	; 공구 옵셋 선택 취소
N50 D=NUMBER	; 공구 옵셋 다시 선택.
N60 RET	

SUPD를 사용한 서브프로그램

프로그램 코드	설명
N10 PROC **SUB_SUP**	
N40 G0 Z49 **SUPD**	; 활성 블록에서 공구 옵셋 억제.
N60 RET	

3.5.8 프로그래밍이 가능한 공구 옵셋(TOFFL, TOFF, TOFFR, TOFFLR):

TOFFx 주소를 기반으로 NC 프로그램에서 보정 메모리에 저장된 공구 옵셋 데이터를 수정하지 않고 유효한 공구 길이 및 유효한 공구 반경을 수정할 수 있습니다.

프로그램 종료 시 해당 옵셋값들은 다시 삭제됩니다.

구문

공구 길이 옵셋

```
TOFFL=<value>
TOFFL[1]=<Value> TOFFL[2]=<Value> TOFFL[3]=<Value>
TOFF[<GeoAx>]=<Value>
```

공구 길이는 세가지 구성 요소 모두에서 동시에 변경될 수 있습니다. 하지만 TOFFL/ TOFFL[1..3] 그룹의 명령과 TOFF[<geometry axis>] 그룹의 명령을 한 블록에서 동시에 사용할 수 없습니다. 마찬가지로 TOFFL 및 TOFFL[1]을 동일한 블록에 동시에 쓸 수 없습니다.

3개의 공구 길이 구성 요소가 모두 한 블록에서 프로그래밍되지 않으면 프로그래밍되지 않은 요소는 변경되지 않습니다. 이 방법으로 블록 별로 다른 요소로 구성된 옵셋을 만들 수 있습니다. 하지만 이는 공구 구성요소가 TOFFL 또는 TOFF만으로 수정된 경우에만 적용됩니다. TOFFL에서 TOFF로 또는 그 반대로 프로그래밍 버전을 변경한 경우 이전에 프로그래밍된 공구 길이 옵셋이 삭제됩니다(예 3 참고).

공구 반경 옵셋

```
TOFFR=<value>
```

동시 공구 길이 옵셋 및 공구 반경 옵셋

```
TOFFLR=<Value>
```

의미

TOFFL:	유효 공구 길이 수정 TOFFL은 인덱스에 관계없이 프로그램할 수 있습니다.	
	TOFFL=...	프로그램된 옵셋 값은 보정 메모리에 보관된 공구 길이 구성요소 **L1**과 동일한 방향으로 적용됩니다. TOFFL 및 TOFFL[1] 지침은 영향이 동일합니다.
	TOFFL[1]=... TOFFL[2]=... TOFFL[3]=...	프로그래밍된 옵셋 값은 옵셋 메모리에 저장된 공구 길이 구성요소인 **L1**, **L2**, **L3**과 동일한 방향으로 적용됩니다.
	참고: 이러한 공구 길이 구성요소 값이 축에서 계산되는 방식은 공구 유형 및 현재 작업 평면(G17/G18/G19)에 따라 달라집니다.	
TOFF:	지정된 기하 축과 평행한 구성요소에서 공구 길이 수정 TOFF는 공구 길이 요소의 방향으로 적용되어 인덱스에서 명시된 기하 축에 평행한 비회전 공구(오리엔테이션 설정이 가능한 공구 캐리어 또는 오리엔테이션 변환)에 적용됩니다. **참고:** 프레임은 공구 길이 구성요소에 프로그래밍된 값을 할당하는 데 영향을 주지 않습니다. 이는 공작물 좌표계(WCS)가 공구 길이 구성 요소를 기하 축에 할당하는 데 사용되는 것이 아니라 기본 공구 위치의 공구 좌표계를 할당하는 데 사용된다는 뜻입니다.	
<GeoAx>:	기하 축의 식별자	
TOFFR:	유효 공구 반경 수정 TOFFR는 프로그래밍된 옵셋값에 의해 **공구 반경 보정이 동작중인 상태**로 적용 중인 공구 반경을 변경합니다.	
TOFFLR:	구성 요소 L1 **및** 유효 공구 반경의 유효 공구 길이 수정 **참고:** 코너 라운딩 기능(유형 111, 121, 131, 156)이 있는 공구의 경우 TOFFLR가 코너 반경을 보정합니다.	
<Value>:	옵셋 값	
	유형:	REAL

예

예 1: 양의 공구 길이 옵셋

사용 공구는 길이 L1 = 100 mm의 드릴입니다.

사용되는 작업 평면은 G17입니다. 즉, 드릴이 Z방향을 가리킵니다.

효과적인 드릴 길이는 1 mm 단위로 증가합니다. 다음 변수는 이 공구 길이 옵셋의 프로그래밍을 위해 사용할 수 있습니다.

- TOFFL=1
- TOFFL[1]=1
- TOFF[Z]=1

예 2: 음의 공구 길이 옵셋

사용 공구는 길이 L1 = 100 mm의 드릴입니다.

사용되는 작업 평면은 G18입니다. 즉, 드릴이 Y방향을 가리킵니다.

효과적인 드릴 길이는 1 mm 단위로 감소합니다. 다음 변수는 이 공구 길이 옵셋의 프로그래밍을 위해 사용할 수 있습니다.

- TOFFL=-1
- TOFFL[1]=-1
- TOFF[Y]=1

예 3: 버전을 TOFFL에서 TOFF로 프로그래밍할 때의 변화

사용 공구는 밀링 공구입니다. 사용되는 작업 평면은 G17입니다.

프로그램 코드	설명
N10 TOFFL[1]=3 TOFFL[3]=5	; 적용 옵셋: L1=3, L2=0, L3=5
N20 TOFFL[2]=4	; 적용 옵셋: L1=3, L2=4, L3=5
N30 TOFF[Z]=1.3	; 적용 옵셋: L1=0, L2=0, L3=1.3

예 4: 평면 변화 후의 옵셋 값 할당

프로그램 코드	설명
N10 $TC_DP1[1,1]=120	
N20 $TC_DP3[1,1]= 100	; 공구 길이 L1=100 mm.
N30 T1 D1 G17	
N40 **TOFF[Z]=1.0**	; Z 방향의 옵셋(G17에서 L1)
N50 G0 X0 Y0 Z0	; 기계 축 위치 X0 Y0 Z101
N60 G18 G0 X0 Y0 Z0	; 기계 축 위치 X0 Y100 Z1.
N70 G17	

프로그램 코드	설명
N80 **TOFFL=1.0**	; L1 방향의 옵셋(G17에서 Z)
N90 G0 X0 Y0 Z0	; 기계 축 위치 X0 Y0 Z101.
N100 G18 G0 X0 Y0 Z0	; 기계 축 위치 X0 Y101 Z0.

본 예에서 Z 축 방향의 1 mm 옵셋은 블록 N60에서 G18로 변경할 때 유지됩니다; Y 축 방향으로 적용되는 공구 길이는 변경되지 않은 채 100 mm로 유지됩니다.

하지만 블록 N100에서 프로그래밍에서 옵셋 공구 길이 L1에 할당되어 G18로 변경할 경우 옵셋이 Y축으로 적용되며, 이 길이 요소가 G18에서 Y축으로 적용됩니다.

예 5: 동시 공구 길이 옵셋 및 공구 반경 옵셋

코너 라운딩 기능(공구 유형 120)이 **없는** 엔드 밀링 커터

프로그램 코드	설명
...	
TOFFLR=0.1	; 적용 옵셋:
	; 공구 길이 옵셋(L1) = 5
	; 공구 반경 옵셋 = 5
...	

b, 코너 라운딩(공구 유형 121) 기능이 **있는** 엔드밀:

프로그램 코드	설명
...	
TOFFLR=0.1	; 적용 옵셋:
	; 공구 길이 옵셋(L1) = 0.1
	; 공구 반경 옵셋 = 0.1
	; 옵셋 코너 반경 = 0.1
...	

추가 정보

공구 길이 옵셋

프로그래밍 종류에 따라 보정 메모리에 보관된 공구 길이 요소 L1, L2 및 L3 (TOFFL) 또는 기하 축 (TOFF) 중 하나에 프로그래밍된 공구 길이 옵셋이 지정됩니다. 프로그래밍된 옵셋은 가공 평면 변경에 맞게 처리됩니다 (G17/G18/G19 ↔ G17/G18/G19):

- 공구 길이 요소에 옵셋 값이 할당되면, 프로그래밍된 옵셋이 적용되는 방향이 그에 따라 교체됩니다.
- 옵셋 값이 기하 축에 할당된 경우 가공 평면 변경이 좌표축과 연관된 할당에 영향을 주지 않습니다.

프로그래밍된 옵셋 값을 공구 길이 요소로 할당할 경우 다음의 설정 데이터 설정을 확인합니다.

SD42940 $SC_TOOL_LENGTH_CONST (작업 평면 변경 시 공구 길이 요소 변경).

SD42950 $SC_TOOL_LENGTH_TYPE (공구 종류와 관계 없이 공구 길이 옵셋 지정).

이 설정에 0이 아닌 유효 값이 있으면 이 값이 공구 데이터에 포함된 G그룹 6(평면 선택 G17/G18/G19) 또는 공구 유형($TC_DP1[<T no.>, <D no.>])에 우선하여 적용됩니다. 즉, 이 설정 데이터가 공구 길이 요소 L1~L3과 같은 방식으로 옵셋 평가에 영향을 미칩니다.

공구 반경 옵셋

TOFFR 어드레스의 효과는 OFFN (페이지 277) 어드레스와 동일합니다. 표면 곡선 변환(TRACYL) 및 슬롯 측면 보정에서만 차이가 납니다. 이 경우, 공구 반경은 음의 부호인 OFFN에 의해 영향을 받습니다만 TOFFR은 양의 부호입니다.

OFFN 과 TOFFR는 동시에 적용 가능합니다. 일반적으로 이때 추가적으로 적용됩니다 (슬롯 측면 보정 제외).

공구 교체

공구 교체(절삭날 교체) 중에 모든 옵셋 값이 유지됩니다. 즉, 새 공구(새 절삭날)에도 효과적입니다.

현 옵셋 값 판독의 시스템 변수

현재 효과적인 옵셋는 다음 시스템 변수로 판독될 수 있습니다.

시스템 변수	의미
$P_TOFFL [<n>] 0 ≤ n ≤ 3	컨텍스트 사전 처리에서 TOFFL (n = 0 의 경우) 또는 TOFFL[1...3] (n = 1, 2, 3 의 경우)의 현 옵셋 값 판독.
$P_TOFF [<GeoAx>]	컨텍스트 전처리에서 TOFF[<GeoAx>]의 현재 옵셋 값을 판독합니다.
$P_TOFFR	컨텍스트 사전처리에서 TOFFR의 현 옵셋 값 판독.
$P_TOFFCR	컨텍스트 전처리에서 코너 반경의 현재 옵셋 값을 판독합니다.
$AC_TOFFL[<n>] 0 ≤ n ≤ 3	주 가동 컨텍스트에서(동기화된 행동) TOFFL (n = 0 의 경우) 또는 TOFFL[1...3] (n = 1, 2, 3 의 경우)의 현 옵셋 값 판독.
$AC_TOFF[<GeoAx>]	메인 실행 컨텍스트(동기 동작)에서 TOFF[<GeoAx>]의 현재 옵셋 값을 판독합니다.

시스템 변수	의미
$AC_TOFFR	주 가동 컨텍스트에서(동기화된 행동) TOFFR의 현 옵셋 값 판독.
$AC_TOFFCR	메인 실행 컨텍스트(동기 동작)에서 코너 반경의 현재 옵셋 값을 판독합니다.

참고

시스템 변수 $AC_TOFFL, $AC_TOFF, AAC_TOFFR은 컨텍스트(NC 프로그램) 전처리로부터 판독할 때 자동 전처리 중단을 실행합니다.

적용

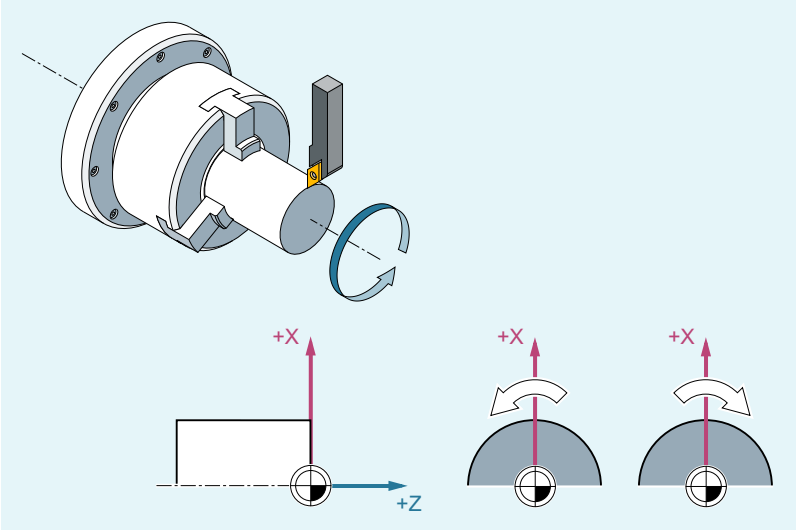
'프로그램 가능한 공구 옵셋' 기능은 코너 반경들이 종종 볼 끝이 아닌 볼 중심에의 CAM 시스템에서 계산되기 때문에 코너 반경의 볼 밀링과 밀링 공구에 특히 유용합니다. 그러나 일반적으로 공구 팁은 공구를 측정하여 보정 메모리에 공구 길이로 저장될 때 측정됩니다.

볼 밀을 이용한 3D 공구 반경 보정의 경우 공구 길이와 반경을 동일한 값으로 동시에 보정하는 것이 유리합니다. 이때 사용자는 TOFFLR 어드레스를 사용할 수 있습니다.

3.6 스핀들 운동

3.6.1 스핀들 속도 (S), 스핀들 회전 방향 (M3, M4, M5)

스핀들 속도 및 회전 방향 값에 따라 스핀들 회전 모션 설정이 달라지며 칩 제거 조건이 결정됩니다.



메인 스핀들 이외에도 다른 스핀들을 이용할 수 있습니다 (예: 서브 스핀들 또는 자동 공구). 규칙에 의해 메인 스핀들은 머신 데이터에서 마스터 스핀들로 지정됩니다. 이 지정 상태는 NC 명령어를 이용하여 변경할 수 있습니다.

구문

```
S... / S<n>=...
M3 / M<n>=3
M4 / M<n>=4
M5 / M<n>=5
```

SETMS(<n>)	
...	
SETMS	

의미

S...:	마스터 스핀들의 속도(rpm)
S<n>=...:	스핀들 <n>의 속도(rpm)
	참고: S0=...으로 지정된 속도가 마스터 스핀들에 적용됩니다.
M3:	마스터 스핀들의 시계 방향 회전
M<n>=3:	스핀들 <n>의 시계 방향 회전
M4:	마스터 스핀들의 시계 반대 방향 회전
M<n>=4:	스핀들 <n>의 시계 반대 방향 회전
M5:	마스터 스핀들의 정지
M<n>=5:	스핀들 <n>의 정지
SETMS(<n>):	스핀들 <n>을 마스터 스핀들로 설정
SETMS:	스핀들 번호를 지정하지 않고 SETMS을 프로그래밍하면 설정된 마스터 스핀들이 사용됩니다.

참고

NC 블록 당 최대 3개의 S 값을 프로그래밍할 수 있습니다. 예를 들면,

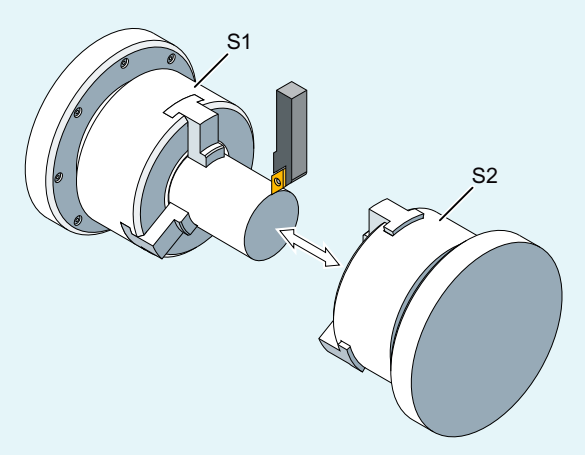
```
S... S2=... S3=...
```

참고

SETMS는 별도의 블록에 프로그램되어야 합니다.

예

S1은 주 스핀들이며 S2는 두 번째 스핀들입니다. 해당 소재는 양면으로 가공됩니다. 이렇게 하려면 작업을 단계별로 분할해야 합니다. 절단 지점에서, 공작물 절단 이후 두 번째 스핀들(S2)에서 가공이 이루어집니다. 이렇게 하려면 S2가 G95를 사용한 마스터 스핀들로 정의되어야 합니다.



프로그램 코드	설명
N10 S300 M3	; 마스터 스핀들의 회전 속도 및 회전 방향.
...	; 소재의 우측 면 가공.
N100 SETMS(2)	; S2를 마스터 스핀들로 지정.
N110 S400 G95 F…	; 새 마스터 스핀들의 속도.
...	; 소재의 좌측 면 가공.
N160 SETMS	; S1을 마스터 스핀들로 재 지정.

추가 정보

마스터 스핀들을 위한 S값 해석

G코드 그룹 1(모달식으로 유효한 동작 명령)에서 G331 또는 G332 기능이 활성화되면, 프로그램된 S값은 항상 rpm 단위의 속도로 해석됩니다. 그렇지 않은 경우, S값에 대한 해석은 G 코드 그룹 15(이송 속도 유형)에 의존하게 됩니다. G96, G961 또는 G962가 활성화되면 S 값은 m/min 단위 주속 일정 제어로 동작합니다. 그렇지 않은 경우 rpm 단위로 인식됩니다.

G96/G961/G962에서 G331/G332으로 변경하면 주속 일정 제어 속도 값이 0으로 설정되고, G331/G332에서 G331/G332 이외의 G 코드 그룹 1에 속한 기능으로 변경할 경우 속도 값이 0으로 설정됩니다. 필요한 경우 해당 S 값을 다시 프로그램해야 합니다.

M 명령어 (M3, M4, M5) 사전 설정

축 명령어의 블록에서 M3, M4, M5 코드는 이송 **전에** 프로그램되어야 합니다 (시스템의 기본 설정).

예제:

프로그램 코드	설명
N10 G1 F500 X70 Y20 S270 M3	; 스핀들이 270 rpm까지 램프업된 후 X 및 Y 방향에서 이동이 실행됩니다.
N100 G0 Z150 M5	; Z 방향으로 후퇴하기 전에 스핀들이 먼저 정지합니다.

참고

축 이송이 이루어져야 할 때 머신 데이터를 사용하여 스핀들이 지령치까지 도달한 경우 이송할 것인지 아니면 프로그램이 실행된 후 곧 바로 이송이 될 것인지 여부를 설정할 수 있습니다.

다중 스핀들로 작업

한 채널에서 동시에 5개의 스핀들 (마스터 스핀들 + 보조 스핀들 4개) 을 사용할 수 있습니다.

이 스핀들 중 하나는 머신 데이터에서 **마스터 스핀들**로 정의됩니다. 나사 절삭, 태핑, 회전 당 이송 속도, 드웰 시간과 같은 특수 기능이 이 스핀들에 적용됩니다. 나머지 스핀들 (예: 보조 스핀들 및 자동 공구) 의 경우, 속도 및 회전/스핀들 정지 방향에 맞는 번호를 지정해야 합니다.

예제:

프로그램 코드	설명
N10 S300 M3 S2=780 M2=4	; 마스터 스핀들: 300 rpm, CW 회전 2번째 스핀들: 780 rpm, CCW 회전

프로그램에 의한 마스터 스핀들 전환

SETMS (<n>) 명령어는 NC 프로그램에서 원하는 스핀들을 마스터 스핀들로 지정하는 데 사용됩니다. SETMS는 별도의 블록에 프로그램되어야 합니다.

예제:

프로그램 코드	설명
N10 SETMS (2)	; 스핀들 2가 마스터 스핀들이 됩니다.

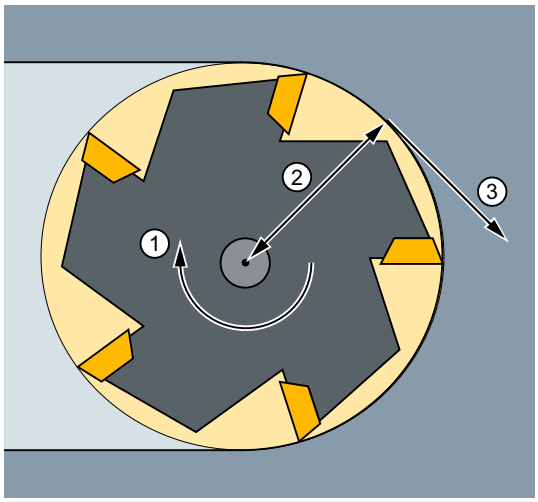
참고

S...으로 지정한 속도와 M3, M4, M5으로 프로그래밍한 코드가 새로 선언된 마스터 스핀들에 적용됩니다.

스핀들 번호를 지정하지 않고 SETMS를 프로그래밍 하면 머신 데이터에 설정된 마스터 스핀들이 사용됩니다.

3.6.2 공구 절삭 속도(SVC)

스핀들 속도 대신, 실제로는 보다 일반적으로 사용하는 공구 절삭 속도를 밀링 작업에 프로그래밍할 수 있습니다.



① 스핀들 속도
② 공구 반경
③ 공구 절삭 속도

제어 시스템은 활성 공구의 반경을 이용하여 프로그램된 공구 절삭 속도에서 적용할 스핀들 속도를 계산합니다.

$S = (SVC * 1000) / (R_T * 2\pi)$

여기서
- S: 스핀들 속도 (rpm)
- SVC: 공구 절삭 속도(단위: m/min 또는 ft/min)
- R_T: 사용 중인 공구의 반경 (mm)

활성 공구의 종류 ($TC_DP1) 는 고려되지 않습니다.

프로그래밍된 공구 절삭 속도는 경로 이송 속도 F 및 G 코드 그룹 15(이송 속도 유형)와 독립적입니다. 회전 방향과 스핀들 시작은 M3과 M4를 통해 구현되며 스핀들 정지는 M5를 통해 구현됩니다.

옵셋 메모리에서 공구 반경 데이터를 수정할 경우 수정된 내용은 다음 번에 공구 옵셋을 선택하거나 사용 중인 옵셋 데이터를 업데이트할 때 적용됩니다.

공구를 교체하거나 공구 옵셋 데이터 세트를 선택 또는 선택을 해제할 경우 적용되는 스핀들 속도가 다시 계산됩니다.

요구 사항

공구 절삭 속도를 프로그래밍하려면 다음이 요구됩니다.

- 회전 공구 (밀링 커터 또는 드릴 공구) 의 기하 비율
- 사용할 공구 옵셋 데이터 세트

구문

```
T... D... SVC[<n>]=<value>
...
S... M3/M4
```

의미

SVC:	공구 절삭 속도 프로그래밍을 위한 키워드	
[<n>]:	스핀들 번호 이 주소 확장은 프로그래밍된 절삭 속도가 적용될 스핀들을 지정합니다. 별도로 명기하지 않을 경우 마스터 스핀들에 절삭 속도가 적용됩니다. **참고:** 각 스핀들에 별도의 공구 절삭 속도가 사전 설정될 수 있습니다. **참고:** 마스터 스핀들에 활성 공구가 로드되어 있는 경우, 별도로 스핀들 번호를 지정하지 않고 SVC를 프로그램 해야 합니다. 마스터 스핀들을 변경한 경우 사용자가 그에 맞게 공구를 선택해야 합니다.	
<값>:	공구 절삭 속도 값	
	단위:	m/min (G71/G710의 경우) 또는 ft/min (G70/G700의 경우)

`T... D...:`	공구의 반지름은 SVC와 함께 블록에 지정되어 있어야 합니다. 따라서 공구 옵셋 데이터 블록을 포함한 해당 공구가 블록에서 선택 또는 활성화되어 있어야 합니다. 동일한 블록에서 프로그래밍을 할 때 SVC나 T/D의 선택 순서는 특별히 정해져 있지 않습니다.
`S... M3/M4:`	스핀들 속도 프로그래밍은 공구 절삭 속도의 선택 해제에 영향을 미칩니다. **참고:** SVC 프로그래밍과 S 프로그래밍 간 전환은 스핀들이 회전될 때를 포함하여 언제든지 가능합니다. 이 경우, 활성화되지 않은 값은 삭제됩니다.

참고

다음과 같은 스핀들 이송 속도에서의 이동이 활성화된 경우 SVC를 프로그래밍할 수 없습니다.

- 주속 일정 제어: G96/G961/G962 S... (페이지 110)
- 연삭 휠 원주 속도 일정 제어: SUG (페이지 115)
- 스핀들 포지셔닝: SPOS/SPOSA/M19 (페이지 131)
- ; 마스터 스핀들을 축 모드로 전환: M70 (페이지 131)

반대로, 이 기능 중 하나를 프로그래밍하면 SVC(공구 절삭 속도)의 선택 해제에 영향을 미칩니다.

참고

최대 공구 속도

시스템 변수 $TC_TP_MAX_VELO[<공구 번호>]를 이용하여 사전에 공구의 최대 속도 (스핀들 속도) 를 설정할 수 있습니다.
제한 속도를 지정하지 않은 경우 속도를 모니터링하지 않습니다.

참고

예를 들어 CAD 시스템에 공구 반경을 입력하고 공구 끝 반경에 대해 표준 공구와의 편차만을 입력해 넣은 경우, 이 CAD 시스템으로 생성한 '표준 공구'의 공구 경로는 SVC 프로그램에서 지원하지 않습니다.

예

다음 사항은 아래 모든 예제에 적용됩니다. 공구 캐리어 = 스핀들(표준 밀링용)

예 1: 밀링 커터 반경 6 mm

프로그램 코드	코멘트
N10 G0 X10 T1 D1	; 밀링 공구 선택, 예: $TC_DP6[1,1] = 6 (공구 반경 = 6 mm)
N20 SVC=100 M3	; 절삭 속도 = 100 m/min
	⇒ 그에 따른 스핀들 속도:
	S = (100 m/min * 1000) / (6.0 mm * 2 * 3.14) = 2653.93 rpm
N30 G1 X50 G95 FZ=0.03	; SVC 및 절삭날 이송 속도
...	

예 2: 동일한 블록의 공구 선택 및 SVC

프로그램 코드	코멘트
N10 G0 X20	
N20 T1 D1 SVC=100	; 블록에서 SVC와 함께 공구 및 옵셋 데이터 세트 선택 (지정된 시퀀스 없음).
N30 X30 M3	; CW 방향으로 스핀들 회전 시작 (절삭 속도 100m/min)
N40 G1 X20 F0.3 G95	; SVC 및 회전 당 이송 속도

예 3: 2개 스핀들의 절삭 속도 지정

프로그램 코드	코멘트
N10 SVC[3]=100 M6 T1 D1	
N20 SVC[5]=200	; 활성 공구 옵셋의 공구 반경은 양 스핀들이 서로 동일합니다. 스핀들 3 및 스핀들 5에 적용되는 속도는 서로 다릅니다.

예 4:

가정:

공구 캐리어에 의해 마스터 또는 공구 교환이 결정됩니다.

MD20124 $MC_TOOL_MANAGEMENT_TOOL CARRIER > 1

공구 교환이 이뤄지더라도 기존의 공구 옵셋은 유지됩니다. 새 공구에 대한 공구 옵셋은 D를 프로그램한 경우에만 적용됩니다.

MD20270 $MC_CUTTING_EDGE_DEFAULT = - 2

프로그램 코드	코멘트
N10 $TC_MPP1[9998,1]=2	; 매거진 위치가 공구 캐리어임
N11 $TC_MPP5[9998,1]=1	; 매거진 위치가 공구 캐리어 1임
N12 $TC_MPP_SP[9998,1]=3	; 공구 캐리어 1이 스핀들 3에 할당됨
N20 $TC_MPP1[9998,2]=2	; 매거진 위치가 공구 캐리어임
N21 $TC_MPP5[9998,2]=4	; 매거진 위치가 공구 캐리어 4임
N22 $TC_MPP_SP[9998,2]=6	; 공구 캐리어 4가 스핀들 6에 할당됨
N30 $TC_TP2[2]="WZ2"	

프로그램 코드	코멘트
N31 $TC_DP6[2,1]=5.0	; 반경 = T2 5.0 mm, 옵셋 D1
N40 $TC_TP2[8]="WZ8"	
N41 $TC_DP6[8,1]=9.0	; 반경 = 9.0 mm (T8), 옵셋 D1
N42 $TC_DP6[8,4]=7.0	; 반경 = 7.0 mm (T8), 옵셋 D4
...	
N100 SETMTH(1)	; 마스터 공구 캐리어 번호 지정
N110 T="WZ2" M6 D1	; 공구 T2가 로드되고 옵셋 D1이 활성화됩니다.
N120 G1 G94 F1000 M3=3 SVC=100	; S3 = (100 m/min * 1000) / (5.0 mm * 2 * 3.14) = 3184.71 rpm
N130 SETMTH(4)	; 마스터 공구 캐리어 번호 지정
N140 T="WZ8"	; T8="WZ8"
N150 M6	; M4=6.
	공구 'WZ8'은 마스터 공구 캐리어에 있지만 MD20270=-2이기 때문에 기존의 공구 옵셋이 계속 적용됩니다.
N160 SVC=50	; S3 = (50 m/min * 1000) / (5.0 mm * 2 * 3.14) = 1592.36 rpm
	공구 캐리어 1에 적용된 옵셋이 계속 적용되며 스핀들 3에 지정됩니다.
N170 D4	; 새 공구 'WZ8'의 옵셋 D4가 (공구 캐리어 4에서) 활성화됩니다.
N180 SVC=300	; S6 = (300 m/min * 1000) / (7.0 mm * 2 * 3.14) = 6824.39 rpm
	; 스핀들 6가 공구 캐리어 4에 할당됨.

예 5:

가정:

스핀들은 공구 캐리어이기도 합니다.

MD20124 $MC_TOOL_MANAGEMENT_TOOL CARRIER = 0

공구 교환 시 공구 옵셋 데이터 세트 D4가 자동 선택됩니다.

MD20270 $MC_CUTTING_EDGE_DEFAULT = 4

프로그램 코드	코멘트
N10 $TC_MPP1[9998,1]=2	; 매거진 위치가 공구 캐리어임
N11 $TC_MPP5[9998,1]=1	; 매거진 위치가 공구 캐리어 1(= 스핀들 1)
N20 $TC_MPP1[9998,2]=2	; 매거진 위치가 공구 캐리어임
N21 $TC_MPP5[9998,2]=3	; 매거진 위치가 공구 캐리어 3(= 스핀들 3)
N30 $TC_TP2[2]="WZ2"	
N31 $TC_DP6[2,1]=5.0	; 반경 = T2 5.0 mm, 옵셋 D1
N40 $TC_TP2[8]="WZ8"	
N41 $TC_DP6[8,1]=9.0	; 반경 = 9.0 mm (T8), 옵셋 D1
N42 $TC_DP6[8,4]=7.0	; 반경 = 7.0 mm (T8), 옵셋 D4
...	
N100 SETMS(1)	; 스핀들 1 = 마스터 스핀들
N110 T="WZ2" M6 D1	; 공구 T2가 로드되고 옵셋 D1이 활성화됩니다.
N120 G1 G94 F1000 M3 SVC=100	; S1 = (100 m/min * 1000) / (5.0 mm * 2 * 3.14) = 3184.71 rpm
N200 SETMS(3)	; 스핀들 3 = 마스터 스핀들

프로그램 코드	코멘트
N210 M4 SVC=150	; S3 = (150 m/min * 1000) / (5.0 mm * 2 * 3.14) = 4777.07 rpm
	T="WZ2"의 공구 옵셋 D1을 기준하여 S1이 이전 속도로 계속 회전합니다.
N220 T="WZ8"	; T8="WZ8"
N230 M4 SVC=200	; S3 = (200 m/min * 1000) / (5.0 mm * 2 * 3.14) = 6369.43 rpm
	T="WZ2"의 공구 옵셋 D1을 기준으로 합니다.
N240 M6	; M3=6
	공구 "WZ8"이 마스터 스핀들에 있으며, 새 공구의 공구 옵셋 D4가 활성화됩니다.
N250 SVC=50	; S3 = (50 m/min * 1000) / (7.0 mm * 2 * 3.14) = 1137.40 rpm
	마스터 스핀들의 옵셋 D4가 활성화됩니다.
N260 D1	; 새 공구 "WZ8"의 옵셋 D1이 활성화됩니다.
N270 SVC[1]=300	; S1 = (300 m/min * 1000) / (9.0 mm * 2 * 3.14) = 5307.86 rpm
	S3 = (50 m/min * 1000) / (9.0 mm * 2 * 3.14) = 884.64 rpm
...	

추가 정보

공구 반경

(활성 공구와 관련된) 다음과 같은 공구 옵셋 데이터가 공구 반경에 영향을 미칩니다.

- $TC_DP6 (반경 - 기하)
- $TC_DP15 (반경 - 마모)
- $TC_SCPx6 ($TC_DP6의 옵셋)
- $TC_ECPx6 ($TC_DP6의 옵셋)

다음 사항은 고려되지 않습니다.

- 온라인 반경 보정
- 프로그램된 형상의 허용치 (OFFN)

공구 반경 보정 (G41/G42)

공구 반경 보정 코드(G41/G42)와 SVC는 둘 다 공구 반경과 관련되지만 기능적 측면에서 서로 무관하고 상호 독립적입니다.

보정 척 (G331, G332) 이 없는 태핑

G331 또는 G332와 연계해서 SVC를 프로그래밍할 수도 있습니다.

동기 동작

동기 동작에는 SVC를 사용할 수 없습니다.

절삭 속도 및 스핀들 속도 프로그래밍 변수 읽기

스핀들의 절삭 속도 및 속도 프로그래밍 변수(스핀들 속도 S 또는 공구 절삭 속도 SVC)는 시스템 변수를 이용하여 읽을 수 있습니다.

- 가공 프로그램에서 시스템 변수를 통한 전처리 정지:

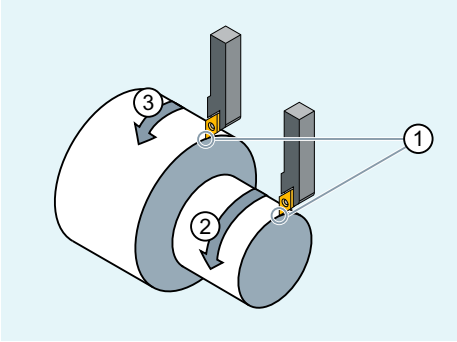
$AC_SVC[<n>]	번호가 <n>인 스핀들에 대해 현재 메인 실행 블록이 전처리될 때 적용되는 절삭 속도.
$AC_S_TYPE[<n>]	번호가 <n>인 스핀들에 대해 현재 메인 실행 블록이 전처리될 때 적용되는 스핀들 속도 프로그래밍 변수.
	값: **의미:**
	1 스핀들 속도 S (rpm)
	2 공구 절삭 속도 SVC(단위: m/min 또는 ft/min)

- 시스템 변수를 통한 가공 프로그램의 전처리 정지 안 함:

$P_SVC[<n>]	스핀들 <n>의 프로그래밍된 절삭 속도
$P_S_TYPE[<n>]	스핀들 <n>의 프로그래밍된 스핀들 속도 프로그래밍 변수
	값: **의미:**
	1 스핀들 속도 S (rpm)
	2 공구 절삭 속도 SVC(단위: m/min 또는 ft/min)

3.6.3 주속 일정 제어 (G96/G961/G962, G97/G971/G972, G973, LIMS, SCC)

'주속 일정 제어' 기능이 동작된 경우, 스핀들 속도가 공작물 직경에 따라 변경되어 절삭 속도 S(m/min 또는 ft/min 단위)가 공구 모서리에서 일정하게 유지됩니다.



① 주속 일정 제어
② 증가된 스핀들 속도
③ 감속된 스핀들 속도

이로 인한 장점은 다음과 같습니다.

- 표면 조도가 향상됩니다.
- 마모가 더 적은 공구로 가공

구문

마스터 스핀들에 대한 주속 일정 제어 활성화/비활성화:

```
G96/G961/G962 S...
...
G97/G971/G972/G973
```

마스터 스핀들의 속도 제한:
```
LIMS=<value>
LIMS[<spindle>]=<value>
```

G96/G961/G962의 다른 기준 축:
```
SCC[<axis>]
```

참고

SCC[<Axis>]를 G96/G961/G962와 함께 또는 개별적으로 프로그래밍할 수 있습니다.

의미

G96:	회전 당 이송 속도(G95 (페이지 117)관련) 및 주속 일정 제어 G96를 통해 G95가 자동으로 활성화됩니다. G95가 아직 활성화되지 않았다면 G96를 호출할 때 새로운 이송 속도 값 F...를 지정해야 합니다.	
G961:	직선 이송 속도(G94 (페이지 117)관련) 및 주속 일정 제어	
G962:	직선 또는 회전 당 이송 속도 및 주속 일정 제어	
S...:	G96, G961 또는 G962에 사용될 경우 S...는 스핀들 속도가 아니라 절삭 속도로 해석됩니다. 절삭 속도는 항상 마스터 스핀들에 적용됩니다.	
	단위:	m/min (G71/G710의 경우) 또는 ft/min (G70/G700의 경우)
	값 범위:	0.1m/min ~ 9999 9999.9m/min
G97:	회전 당 이송 속도 및 일정 스핀들 속도(주속 일정 제어 OFF)	
G971:	직선 이송 속도 및 일정 스핀들 속도(주속 일정 제어 OFF)	
G972:	직선 이송 속도 또는 회전 당 이송 속독 및 일정 스핀들 속도(주속 일정 제어 OFF)	
G973:	스핀들 속도 제한 없는 회전 당 이송 속도 및 일정 스핀들 속도(ISO 모드를 위한 LIMS 없는 G97)	
	참조: G97 (또는G971 ... G973) 이후에는 S...가 다시 rpm 단위의 스핀들 속도를 해석됩니다. 새 스핀들 속도가 지정되지 않은 경우 G96(각각 G961 또는G962)으로 설정된 마지막 속도가 유지됩니다.	
LIMS:	마스터 스핀들의 속도 제한 (G96/G961/G97이 동작된 경우에만 적용됨) 마스터 스핀들을 선택할 수 있는 기계의 경우, 한 블록에서 최대 4 개의 스핀들에 각기 다른 제한 속도 값을 지정할 수 있습니다.	
	<spindle>:	스핀들 번호
	<value>:	스핀들 속도 상한값 (rpm)
SCC:	G96/G961/G962 코드 중 하나가 활성화되면 SCC[<axis>]를 이용하여 원하는 기하 축을 기준 축으로 지정할 수 있습니다.	

참고

G96/G961/G962를 처음으로 선택한 경우, 주속 일정 제어 속도 S...를 반드시 입력해야 합니다. G96/G961/G962를 다시 선택하는 경우 입력은 선택 사항입니다.

참고

LIMS로 프로그래밍한 제한 속도는 G26으로 프로그래밍한 제한 속도나 설정 데이터에 지정된 제한 속도를 초과해서는 안 됩니다.

참고

G96/G961/G962를 위한 기준 축은 SCC[<axis>]이 프로그램 될 때 채널에 지정된 기하 축이어야 합니다. G96/G961/G962 기능 중 하나가 활성화될 때 SCC[<axis>]도 프로그램 될 수 있습니다.

예

예 1: 속도 제한을 이용한 주속 일정 제어 적용

프로그램 코드	설명
N10 SETMS (3)	
N20 G96 S100 LIMS=2500	; 주속 일정 제어 속도 = 100 m/min, 최대 속도 = 2500 rpm
...	
N60 G96 G90 X0 Z10 F8 S100 LIMS=444	; 최대 속도 = 444 rpm

예 2: 4개 스핀들에 대한 속도 제한 정의

스핀들 1 (마스터 스핀들) 과 스핀들 2, 3, 및 4의 제한 속도는 다음과 같이 지정합니다.

프로그램 코드
N10 LIMS=300 LIMS[2]=450 LIMS[3]=800 LIMS[4]=1500
...

예 3: X 축을 사용한 페이스 커팅 시 Y 축 지정

프로그램 코드	설명
N10 G18 LIMS=3000 T1 D1	; 최대 속도 3000 rpm
N20 G0 X100 Z200	
N30 Z100	
N40 G96 S20 M3	; 주속 일정 제어 속도 20 m/min은 X 축에 따라 달라짐.
N50 G0 X80	
N60 G1 F1.2 X34	;1.2 mm/rev로 X 축 방향 페이스 커팅.
N70 G0 G94 X100	
N80 Z80	
N100 T2 D1	
N110 G96 S40 SCC[Y]	; Y 축이 G96에 지정되고 G96 이 활성화됨 (싱글 블록에서 수행 가능). ; 주속 일정 제어 속도 40 m/min은 X 축에 따라 달라짐.
...	

프로그램 코드	설명
N140 Y30	
N150 G01 F1.2 Y=27	' Y에서의 플런지 절삭, 이송 속도 F = 1.2 mm/rev
N160 G97	; 주속 일정 제어 OFF
N170 G0 Y100	

추가 정보

스핀들 속도 계산

페이스 축 (반경) 의 SZS 위치는 프로그램된 절삭 속도에서 스핀들 속도를 계산하는 데 기초가 됩니다.

참고

WCS와 SZS 사이의 프레임 (예: SCALE, TRANS 또는 ROT와 같은 프로그래밍 프레임) 은 스핀들 속도의 계산 시 고려되며 속도 변경을 야기할 수 있습니다 (예를 들어, SCALE에서 유효 직경이 변경된 경우).

속도 제한 LIMS

직경이 크게 차이가 있는 공작물을 가공하는 경우 LIMS (최대 스핀들 속도) 로 스핀들의 제한 속도를 지정하는 것이 바람직합니다. 이것으로 작은 직경에서의 과도한 속도를 방지할 수 있습니다. LIMS는 G96, G961 및 G97가 활성화된 경우에만 적용됩니다. G971을 선택한 경우 LIMS가 적용되지 않습니다. 메인 실행으로 해당 블록을 로드할 때 프로그래밍된 모든 값은 설정 데이터로 이송됩니다.

참고

가공 프로그램에서 LIMS로 변경한 속도 한계가 설정 데이터에 적용되어 프로그램 종료 후에도 계속 유지됩니다.

하지만 프로그램 종료 후 LIMS로 변경한 속도 한계가 더 이상 적용되지 않게 하려면 다음 정의를 장비 제조업체의 GUD 블록에 삽입해야 합니다.

REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC

주속 일정 제어 해제 (G97/G971/G972/G973)

G97 (또는G971... G973) 이후에는 S...가 다시 rpm 단위의 스핀들 속도를 해석됩니다. 새 스핀들 속도가 지정되지 않은 경우 G96(각각 G961 또는G962)으로 설정된 마지막 속도가 유지됩니다.

G96/G961 기능 역시 G94 또는 G95를 이용하여 비활성화될 수 있습니다. 이 경우 최종 프로그래밍된 속도 S...가 후속 가공 작업에 사용됩니다.

먼저 G96을 프로그램 하지 않은 상태로 G97을 프로그램 할 수 있습니다. 그러면 이 기능은 G95와 동일한 효과를 가집니다. 그리고 IMS 또한 프로그램 할 수 있습니다.

G973을 사용하여 스핀들 속도 제한 없이 주속 일정 제어를 해제할 수 있습니다.

참고

반드시 수평축은 머신 데이터에 정의 되어야 합니다.

급 이송 G0

급 이송 G0을 사용하는 경우 속도 변화가 없습니다.

예외:

급 횡단으로 형상에 접근하고 차후 NC 블럭이 G1/G2/G3/... 등의 경로 명령어를 포함한다면, 그 다음 명령어를 위해 속도는 G0 접근 블럭에서 조정됩니다.

G96/G961/G962의 기준 축

G96/G961/G962 코드 중 하나가 활성화되면 SCC[<axis>]를 이용하여 원하는 기하 축을 기준 축으로 지정할 수 있습니다. 기준 축이 변할 경우 주속 일정 제어 속도의 TCP (공구 중심점) 기준 위치에 차례로 영향을 미치게 되어 설정된 가감속을 통하여 해당 스핀들 속도에 도달하게 됩니다.

할당된 채널 축 교환

G96/G961/G962의 기준 축 속성은 항상 기하 축에 부여됩니다. 할당된 채널 축에 관련된 축 교체의 경우 G96/G961/G962에 대한 기준 축 속성이 기존의 채널에 계속 유지됩니다.

기하 축을 교체하더라도 주속 일정 제어 속도에 기하 축을 지정하는 방법에는 영향을 미치지 않습니다. G96/G961/G962의 TCP 기준 위치가 기하 축 교체의 영향을 받는다면 스핀들은 램프를 통해 새 속도에 도달하게 됩니다.

기하 축 교체의 결과로 인해 새로운 채널 축이 할당되지 않는 경우(예:GEOAX(0,X)), 스핀들 속도는 G97에 따라 동결됩니다.

기준 축을 지정한 상태에서 기하 축을 교체한 예

프로그램 코드	설명
N05 G95 F0.1	
N10 GEOAX(1, X1)	;채널 축 X1이 첫 번째 기하 축이 됩니다.
N20 SCC[X]	;첫 번째 기하 축 (X) 가
	;G96/G961/G962의 기준 축이 됩니다.
N30 GEOAX(1, X2)	;채널 축 X2가 첫 번째 기하 축이 됩니다.
N40 G96 M3 S20	;G96의 기준 축이 채널 축 X2가 됩니다.

프로그램 코드	설명
N05 G95 F0.1	
N10 GEOAX(1, X1)	;채널 축 X1이 첫 번째 기하 축이 됩니다.
N20 SCC[X1]	;X1과 암시적으로 첫 번째 기하 축 (X) 이 G96/G961/G962의 기준 축이 됩니다.
N30 GEOAX(1, X2)	;채널 축 X2가 첫 번째 기하 축이 됩니다.
N40 G96 M3 S20	;G96의 기준 축은 X2 또는 X가 되며 알람은 출력되지 않습니다.

프로그램 코드	설명
N05 G95 F0.1	
N10 GEOAX(1, X2)	;채널 축 X2가 첫 번째 기하 축이 됩니다.
N20 SCC[X1]	;X1은 기하 축이 아니며 알람이 출력됩니다.

프로그램 코드	설명
N05 G0 Z50	
N10 X35 Y30	
N15 SCC[X]	;G96/G961/G962의 기준 축은 X입니다.
N20 G96 M3 S20	;10 mm/min 에서 주속 일정 제어 ON.
N25 G1 F1.5 X20	;1.5 mm/rev로 X 축 방향 페이스 커팅.
N30 G0 Z51	
N35 SCC[Y]	;G96의 기준 축은 Y이고 스핀들 속도 (Y30) 가 감소함.
N40 G1 F1.2 Y25	;1.2 mm/rev로 Y 축 방향 페이스 커팅.

3.6.4 연삭 휠 원주 속도 일정 제어 (GWPSON, GWPSOF) 켜기/끄기:

미리 규정된 절차 GWPSON(...) 그리고 GWPSOF(...)의 경우, 연삭 공구 (공구 유형: 400~499)에 대한 연삭 휠 원주 속도 일정 제어 (GWPS)가 켜지거나 꺼집니다.

구문

```
GWPSON(<TNo>)
S<n>=... :
...
GWPSOF(<TNo>)
```

의미

GWPSON(...):	연삭 휠 원주 속도 일정 제어 켜짐
GWPSOF(...):	연삭 휠 원주 속도 일정 제어 꺼짐
<TNo>:	T 번호 **참고:** 현재 사용되고 있는 활성 연삭 휠이 아니라 비활성 연삭 휠에 대해 연삭 휠 원주 속도 일정 제어가 켜지거나 꺼지는 경우에만 필요함.
S<n>=…:	스핀들 <n>의 연삭 휠 원주 속도 (m/s 또는 ft/s)
S0=... 또는 S...:	마스터 스핀들의 연삭 휠 원주 속도

상태 조회

다음 시스템 변수를 사용하여 가공 프로그램에서 GWPS가 특정 스핀들에 활성화되었는지 여부를 조회할 수 있습니다.

$P_GWPS[<n>] ; 위치 <n> = 스핀들 번호

값	의미
0 (= FALSE)	GWPS **해제**
1 (= TRUE)	GWPS **동작**

3.6.5 프로그램 가능한 스핀들 속도 제한 (G25, G26)

머신 데이터와 셋팅 데이터에 지정된 최소 및 최대 스핀들 속도는 가공 프로그램 명령어로 수정할 수 있습니다.

프로그래밍된 스핀들 속도 제한은 채널의 모든 스핀들에 사용 가능합니다.

구문

```
G25 S… S1=… S2=…
G26 S… S1=… S2=…
```

의미

G25: **하한** 스핀들 제한 속도

G26: **상한** 스핀들 제한 속도

S... S1=… S2=… : 최소 또는 최대 스핀들 속도

참고:

각 블록에 최대 3 개의 스핀들 속도 한계를 프로그래밍할 수 있습니다.

값 범위: 0.1~9999 9999.9rpm

참고

G25 또는 G26으로 프로그래밍된 스핀들 제한 속도는 셋팅 데이터의 제한 속도를 덮어쓰고 프로그램 종료 후에도 계속 유지됩니다.

하지만 프로그램 종료 후 G25/G26으로 변경한 속도 한계가 더 이상 적용되지 않게 하려면 다음 정의를 장비 제조업체의 GUD 블록에 삽입해야 합니다.

REDEF $SA_SPIND_MIN_VELO_G25 PRLOC

REDEF $SA_SPIND_MAX_VELO_G26 PRLOC

예

프로그램 코드	코멘트
N10 G26 S1400 S2=350 S3=600	; 마스터 스핀들, 스핀들 2 및 스핀들 3의 속도 상한.

3.7 속도 (피드) 제어

3.7.1 이송 속도 (G93, G94, G95, F, FGROUP, FL, FGREF)

이 명령어들은 NC 프로그램에서 가공 순서와 관련된 모든 축의 이송 속도를 설정하는 데 사용됩니다.

구문

```
G93
G94
G95
F<값>
FGROUP(<axis_1>,<axis_2>,...)
FGREF[<rotary axis>]=<reference radius>
FL[<axis>]=<value>
```

의미

G93:	경로 피드 유형: 역 시간 이송 속도 [rpm]
G94:	경로 피드 유형: 직선 이송 속도 [mm/min], [inch/min] 또는 [degrees/min]
G95:	경로 피드 유형: 회전 당 이송 속도 [mm/rev] 또는 [inch/rev] 회전 당 이송 속도는 마스터 스핀들이나 다른 스핀들 또는 로터리 축에서 계산할 수 있습니다.
F<값>	FGROUP으로 선택한 모든 경로 축의 경로 이송 속도.
FGROUP:	F로 프로그램한 경로 피드의 기준이 되는 경로 축 정의
FGREF:	FGREF를 사용하여 FGROUP에 지정된 각 로터리 축의 유효 반경 (<reference radius>) 을 프로그래밍합니다.
FL:	동기화/경로 축의 속도 한계 G94로 지정한 단위가 적용됩니다. 축 (채널 축, 기하 축 또는 오리엔테이션 축) 당 하나의 FL 값을 지정할 수 있습니다.
<axis>:	채널 축 이름, 유형: AXIS

예

예 1: FGROUP의 운영 모드

다음의 예제를 통해 FGROUP이 경로 및 경로 이송 속도에 미치는 영향을 확인할 수 있습니다. 변수 $AC_TIME에는 초 단위의 블록 시작 시간이 포함되어 있습니다. 이는 동기 동작에만 사용될 수 있습니다.

프로그램 코드	코멘트
N100 G0 X0 A0	
N110 FGROUP(X,A)	
N120 G91 G1 G710 F100	; 이송 속도= 100mm/min 또는100 °/min
N130 DO $R1=$AC_TIME	
N140 X10	; 이송 속도 = 100 mm/min, 경로 = 10 mm, R1 = 약 6초
N150 DO $R2=$AC_TIME	
N160 X10 A10	; 이송 속도 = 100 mm/min, 경로 = 14.14 mm, R2 = 약 8초
N170 DO $R3=$AC_TIME	
N180 A10	; 이송 속도 = 100 °/min, 경로 = 10°, R3 = 약 6초
N190 DO $R4=$AC_TIME	

프로그램 코드	코멘트
N200 X0.001 A10	; 이송 속도 = 100 mm/min, 경로 = 10 mm, R4 = 약 6초
N210 G700 F100	; 이송 속도= 2,540 mm/min 또는100 °/min
N220 DO $R5=$AC_TIME	
N230 X10	; 이송 속도 = 2,540 mm/min, 경로 = 254 mm, R5 = 약 6초
N240 DO $R6=$AC_TIME	
N250 X10 A10	; 이송 속도 = 2,540 mm/min, 경로 = 254.2 mm, R6 = 약 6초
N260 DO $R7=$AC_TIME	
N270 A10	; 이송 속도 = 100 °/min, 경로 = 10°, R7 = 약 6초
N280 DO $R8=$AC_TIME	
N290 X0.001 A10	; 이송 속도 = 2,540 mm/min, 경로 = 10 mm, R8 = 약 0.288초
N300 FGREF[A]=360/(2*$PI)	; 적용 반경을 통해 1° = 1인치 설정
N310 DO $R9=$AC_TIME	
N320 X0.001 A10	; 이송 속도 = 2,540 mm/min, 경로 = 254 mm, R9 = 약 6초
N330 M30	

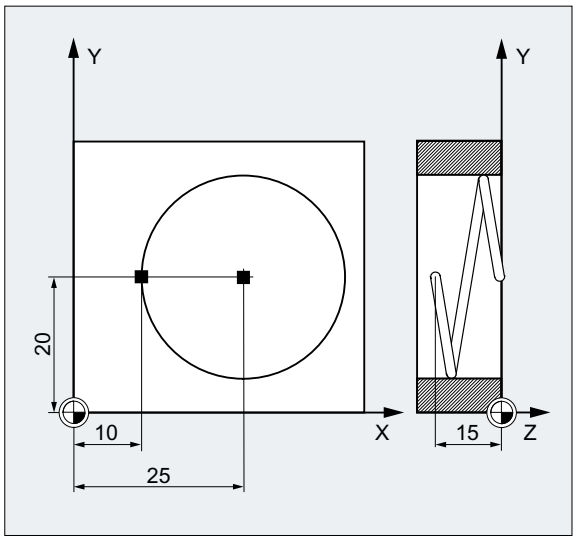
예 2: 제한 속도 FL로 동기 축 이송

동기 축 Z가 속도 한계에 도달하면 경로 축의 경로 속도는 감소합니다.

프로그램 코드
N10 G0 X0 Y0
N20 FGROUP(X)
N30 G1 X1000 Y1000 G94 F1000 FL[Y]=500
N40 Z-50

예 3: 나선형 보간

경로 축 X 및 Y는 프로그램 된 속도로 이송되며, 절입 축 Z는 해당 축들에 동기된 축입니다.

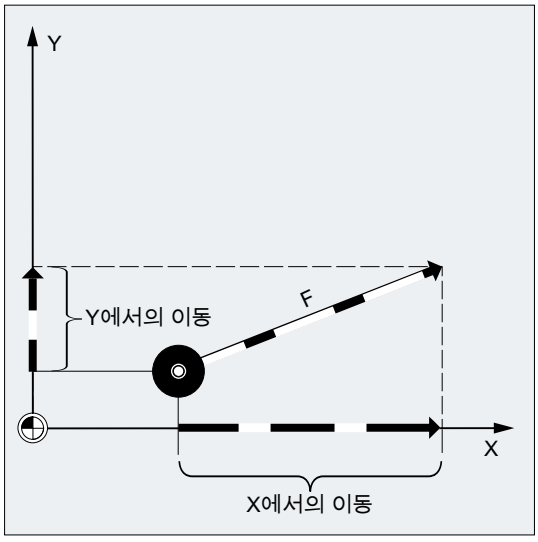


프로그램 코드	코멘트
N10 G17 G94 G1 Z0 F500	; 공구 절입.
N20 X10 Y20	; 시작 위치에 접근.
N25 FGROUP(X,Y)	; 축 X/Y는 경로 축이며 Z는 동기 축입니다.
N30 G2 X10 Y20 Z-15 I15 J0 F1000 FL[Z]=200	; 원호 경로에서 이송 속도는 1,000 mm/min이며 Z 방향의 이송은 동기화됩니다.
...	
N100 FL[Z]=$MA_AX_VELO_LIMIT[0,Z]	; 제한 속도는 MD를 통해 해제됩니다. MD 값을 확인.
N110 M30	; 프로그램 종료

추가 정보

경로 축의 이송 속도 (F)

일반적으로 경로 속도는 이송에 관여하는 전체 축의 개별 속도 요소로 구성되며, 커터의 중심점이나 선삭 공구 끝을 기준으로 합니다.



이송 속도는 어드레스 F에 지정됩니다. 머신 데이터의 디폴트 설정에 따라 G 명령어로 지정한 측정 단위가 mm 또는 인치로 결정됩니다.

NC 블록당 하나의 F 값을 지정할 수 있습니다. 이송 속도 단위는 G 명령어 G93/G94/G95를 사용하여 지정합니다. 이송 속도 F는 경로 축에서만 작용하고 새 이송 속도가 프로그래밍될 때까지 사용 중인 상태를 유지합니다. F 이후에 속도값을 입력할 수 있습니다.

예:

```
F100 또는 F 100

F.5

F=2*FEED
```

이송 속도 유형 (G93/G94/G95)

G 명령어 G93, G94 및 G95는 모달입니다. G93, G94 및 G95로 전환하는 경우 경로 이송속도 값을 다시 프로그래밍해야 합니다. 로터리 축을 사용하여 가공할 경우, 이송 속도를 °/min으로 지정할 수 있습니다.

인버스-타임 이송 속도 (G93)

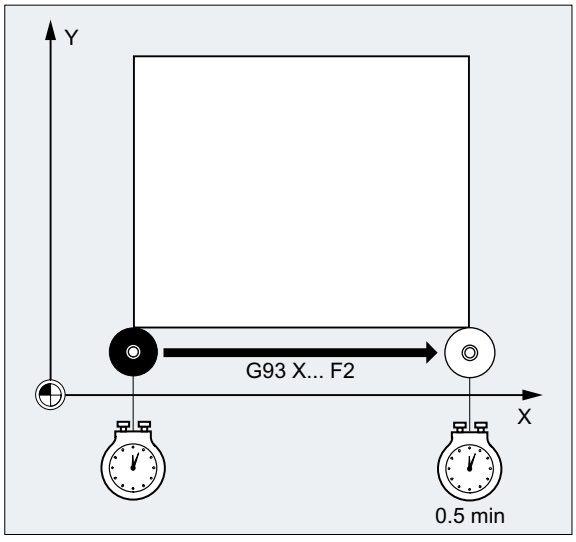
인버스-타임 이송 속도는 블록에서 모션 명령을 실행하기 위해 필요한 시간을 지정합니다.

단위: rpm

예:

```
N10 G93 G01 X100 F2
```

의미: 프로그램된 경로를 0.5분에 걸쳐 이동합니다.



참고

블록 간의 경로 길이가 크게 다른 경우 G93으로 각 블록의 F 값을 지정해야 합니다. 로터리 축을 사용하여 가공할 경우, 이송 속도를 °/min으로 지정할 수 있습니다.

동기 축 이송 속도

F 를 사용하여 프로그램 된 속도는 모든 경로 축에 적용되지만 동기 축에는 적용되지 않습니다. 동기 축은 경로 축과 같이 해당 경로에 동일한 시간을 사용하여 제어됩니다, 따라서 전체 축은 동시에 종점에 도달하게 됩니다.

동기 축의 속도 한계 (FL)

FL 명령어를 이용하여 동기 축의 속도 한계를 프로그래밍할 수 있습니다. FL을 프로그래밍하지 않은 경우 급 이송 속도가 적용됩니다. FL은 MD (MD36200 $MA_AX_VELO_LIMIT) 를 할당하여 해제됩니다.

경로 축을 동기 축 같이 이송 (FGROUP)

FGROUP을 이용하여 경로 축이 경로 이송 속도로 이송하도록 지정하거나 경로 축을 동기 축으로 지정할 수 있습니다. 예를 들어 헬리컬 보간에서 단지 두 개의 기하 축 X 및 Y가 프로그래밍된 이송 속도로 이송하도록 지정할 수 있습니다. 이 경우 절입 축 Z는 동기 축입니다.

예: FGROUP(X,Y)

FGROUP 변경

FGROUP으로 지정한 설정을 다음과 같은 방법으로 변경할 수 있습니다.

1. FGROUP를 다시 프로그래밍: 예: FGROUP(X,Y,Z)
2. 축을 지정하지 않은 상태로 FGROUP을 프로그래밍: FGROUP()
   FGROUP()에 따라 머신 데이터의 초기 설정이 다음과 같이 적용됩니다. 이제 기하 축이 다시 한 번 경로 축 그룹화에서 이송됩니다.

참고

FGROUP을 사용할 때 축 이름은 반드시 채널 축 이름을 사용해야 합니다.

이송 속도 F의 측정 단위

G700 및 G710의 기하학적 설정 외에도 이들 G 코드는 이송 속도 F의 측정 단위를 지정하는 데도 사용됩니다. 다시 말하면,

- G700의 경우: [inch/min]
- G710의 경우: [mm/min]

참고

G70/G71은 이송 속도 설정에 **전혀** 영향을 미치지 않습니다.

제한 속도 FL을 사용한 동기 축의 측정 단위

G 코드 G700/G710을 사용하여 F에 설정한 단위는 FL에도 적용됩니다.

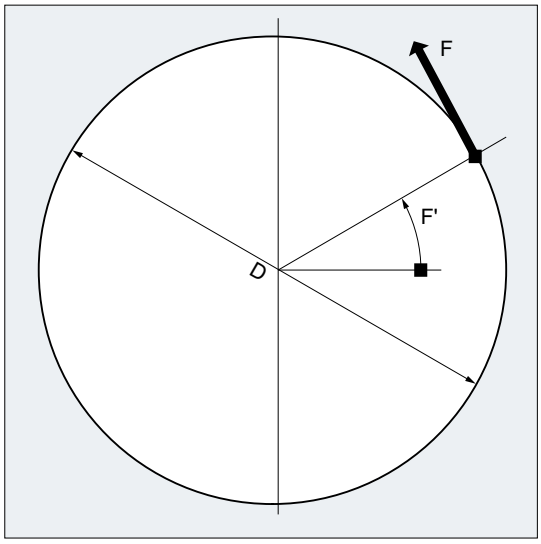
로터리 축 및 직선 축의 단위

FGROUP로 연결되어 한 경로를 함께 이송하는 직선 축과 로터리 축의 경우, 이송 속도는 직선 축의 단위로 해석됩니다. (G94/G95의 디폴트 설정 상태에 따라 단위가 mm/min 또는 inch/min와 mm/rev 또는 inch/rev로 결정됩니다.)

로터리 축에서 mm/min 또는 inch/min 단위의 탄젠트 속도는 다음 공식에 따라 계산됩니다:

F[mm/min] = F'[도/min] * π * D[mm]/360[도]

의미: F: 탄젠트 속도
F': 각 속도
π: 원호 상수
D: 직경



경로 속도 F 를 사용한 로터리 축 이송 (FGREF)

로터리 축에 의해 공구 또는 공작물 또는 양쪽 모두가 이동되는 가공에서 적용되는 가공 속도는 F 값을 기준으로 일반적으로 경로 속도로 인식됩니다. 이를 위해서는 관련된 모든 로터리 축의 적용 반경 (기준 반경) 을 지정해야 합니다.

기준 반경의 단위는 G70/G71/G700/G710 설정에 따라 달라집니다.

관련된 모든 축이 FGROUP 명령어에 포함돼야만 경로 이송 속도를 계산하는 데 반영됩니다.

FGREF가 프로그램 되지 않은 상태와의 호환성을 위해 전원 ON 및 RESET 시 1 degree = 1 mm 가 적용됩니다. 이는 FGREF=360 mm/(2π)=57.296mm의 기준 반경에 해당합니다.

참고

이 디폴트 설정은 현재 사용 중인 기본 시스템 (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) 및 현재 활성 상태인 G70/G71/G700/G710 설정과 무관합니다.

특수 상황:

프로그램 코드

```
N100 FGROUP(X,Y,Z,A)
N110 G1 G91 A10 F100
N120 G1 G91 A10 X0.0001 F100
```

이런 유형의 프로그램에서는 N110에 프로그래밍된 F 값이 로터리 축 이송 속도 (°/min) 로 평가되는 반면, N120의 이송 속도는 현재 활성 상태인 G70/G71/G700/G710 설정에 따라 100 inch/min 또는100 mm/min으로 평가됩니다.

유의사항
이송 속도 차이 블록에 로터리 축만 프로그래밍된 경우 FGREF도 평가됩니다. 이 경우에 반경 기준이 다음 FGREF 디폴트 설정과 일치하는 경우에 한하여 일반적인 F 값의 도/분 단위 해석 결과가 적용됩니다. • G71/G710의 경우: FGREF[A]=57.296 • G70/G700의 경우: FGREF[A]=57.296/25.4

기준 반경 읽기

로터리 축의 기준 반경 값은 시스템 변수를 사용하여 읽을 수 있습니다.

- 시스템 변수를 통한 가공 프로그램에서의 동기 동작 또는 전처리 정지:

  $AA_FGREF[<축>] 현재의 메인 실행 값

- 다음 시스템 변수를 통한 가공 프로그램의 전처리 정지 안 함:

  $PA_FGREF[<축>] 프로그래밍된 값

프로그래밍된 값이 없는 경우 두 변수에서 디폴트 설정인 360 mm/(2π) = 57.296 mm가 사용됩니다.

직선 축의 경우 두 변수의 값은 항상 1mm입니다.

속도에 영향을 미치는 경로 축 읽기

경로 보간과 관련된 축은 시스템 변수를 이용하여 읽을 수 있습니다.

- 시스템 변수를 통한 가공 프로그램에서의 동기 동작 또는 전처리 정지:

  $AA_FGROUP[<축>] 지정된 축이 기본 설정이나 FGROUP 프로그래밍을 통해 현재 메인 실행 레코드의 경로 속도에 영향을 미치는 경우 "1"의 값을 반환합니다. 그 밖의 경우에는 변수가 "0"의 값을 반환합니다.

  $AC_FGROUP_MASK 경로 속도에 영향을 미칠 수 있는 FGROUP로 프로그래밍된 채널 축의 비트 키를 반환합니다.

- 시스템 변수를 통한 가공 프로그램의 전처리 정지 안 함:

$PA_FGROUP[<축>]	지정된 축이 기본 설정이나 FGROUP 프로그램을 통해 경로 속도에 영향을 미치는 경우 "1"의 값을 반환합니다. 그 밖의 경우에는 변수가 "0"의 값을 반환합니다.
$P_FGROUP_MASK	경로 속도에 영향을 미칠 수 있는 FGROUP로 프로그래밍된 채널 축의 비트 키를 반환합니다.

FGREF를 사용한 오리엔테이션 축의 경로 기준 팩터

오리엔테이션 축과 관련한 FGREF[] 인자의 작동 모드는 로터리 축 또는 벡터 보간에 의해 공구 오리엔테이션의 변경이 실행되는가에 달려있습니다.

로터리 축 보간의 경우 로터리 축의 경우와 마찬가지로 오리엔테이션 축의 적절한 FGREF 인자는 개별적으로 축 경로의 기준 반경으로 계산됩니다.

벡터 보간의 경우 개별 FGREF 인자의 기하학적 평균값으로 계산되는 유효 FGREF 인자가 적용됩니다.

FGREF[적용] = [(FGREF[A] * FGREF[B]...)]의 n 제곱근

의미: A: 첫 번째 오리엔테이션 축 이름
B: 두 번째 오리엔테이션 축 이름
C: 세 번째 오리엔테이션 축 이름
n: 오리엔테이션 축 개수

예:

표준 5축 변환에는 두 개의 오리엔테이션 축이 있으므로 적용 팩터는 두 축 팩터의 곱의 제곱근이 됩니다.

FGREF[적용] = [(FGREF[A] * FGREF[B])]의 제곱근

참고

따라서 오리엔테이션 축 FGREF의 적용 팩터를 사용하여 프로그래밍된 경로 이송 속도의 기준이 될 공구의 원점을 지정할 수 있습니다.

3.7.2 포지셔닝 축 이송 (POS, POSA, POSP, FA, WAITP, WAITMC)

포지셔닝 축은 경로 축과는 달리 별도의 축 이송 속도를 통해 이송됩니다. 보간이 이루어지지는 않습니다. POS/POSA/POSP 명령어로 포지셔닝 축은 횡단되며 이동 시퀀스는 그때 조정됩니다.

다음이 포지셔닝 축의 일반적인 예입니다.

- 팔레트 피드 장비
- 측정 스테이션

WAITP를 사용하여 이전 NC 블록에서 POSA로 프로그래밍된 종점까지 도달할 때까지 NC 프로그램에서 축 위치를 확인할 수 있습니다.

WAITMC은 지정된 대기 라벨을 확인한 즉시 다음 NC 블록을 로드합니다.

구문

```
POS[<axis>]=<position>

POSA[<axis>]=<position>

POSP[<axis>]=(<end position>,<partial length>,<mode>)

FA[<axis>]=<value>

WAITP(<axis>), 별도의 NC 블록에서 프로그래밍

WAITMC(<wait marker>)
```

의미

POS/POSA:	포지셔닝 축을 지정된 위치로 이동 POS 및 POSA는 기능적인 측면에서 동일하지만 블록 변경 특성에서 다음과 같은 차이를 보입니다. • POS는 특정 위치에 도달할 때까지 NC 블록 실행을 유보합니다. • POSA는 특정 위치에 도달하지 않더라도 NC 블록을 실행합니다.		
	<axis>:	이송될 축의 이름 (채널 또는 기하 축 이름)	
	<position>:	도달해야 할 축 위치	
		유형:	REAL

POSP:	포지셔닝 축을 섹션에 지정된 끝점으로 이동		
	<end position>:	도달해야 할 축의 끝점	
	<partial length>:	섹션의 길이	
	<mode>:	접근 모드	
		= 0:	최종 두 개의 섹션에서, 종점까지의 잔여 경로가 사전에 동일하게 설정된 크기의 두 개의 잔여 섹션으로 나눠집니다.
		=1:	산정된 모든 부분 길이의 총계가 끝점까지의 경로와 정확히 일치하도록 부분 길이가 조정됩니다.
	참고: POSP는 구체적으로 오실레이션 모션을 프로그래밍 (페이지 1084)하는 데 사용됩니다.		
FA:	지정된 포지셔닝 축의 이송 속도		
	<axis>:	이송될 축의 이름 (채널 또는 기하 축 이름)	
	<Value>:	이송 속도	
		단위:	mm/min, inch/min 또는 °/min
	참고: 최대 5개의 FA 값을 NC 블록별로 프로그래밍할 수 있습니다.		
WAITP:	포지셔닝 축이 이송될 때까지 대기 POSA로 이전 NC 블록에 프로그래밍된 특정 포지셔닝 축이 미세 정위치 정지로 끝점에 도달할 때까지 후속 블록이 처리되지 않습니다.		
	<axis>:	WAITP 명령어가 적용될 축의 이름 (채널 또는 기하 축 이름)	
	참고: WAITP를 이용하여 축을 오실레이션 축으로 지정하거나 PLC를 통해 동일한 포지셔닝 축으로 이송할 수 있습니다.		
WAITMC:	지정된 대기 라벨을 수신할 때까지 대기 대기 라벨을 수신하면 즉시 다음 NC 블록을 로드합니다.		
	<wait marker>:	대기 라벨 번호	

⚠ 주의

POSA를 사용한 이송

다음 블록에 암시적으로 전처리 정지를 유발하는 명령이 있는 경우 이미 전처리되어 저장된 블록들이 모두 실행될 때까지 이 블록은 실행되지 않습니다. 이전 블록은 정위치 정지에서 멈춥니다 (G9와 같음).

예

예 1: POSA를 통한 이송 및 기계 상태 데이터 액세스

시스템은 기계 상태 데이터 ($A...)를 액세스하기 위해 내부 전처리 정지를 생성합니다. 저장된 모든 전처리 블록의 실행이 완료될 때까지 가공이 중단됩니다.

프로그램 코드	설명
N40 POSA[X]=100	
N50 IF $AA_IM[X]==R100 GOTOF LABEL1	; 기계 상태 데이터 액세스.
N60 G0 Y100	
N70 WAITP(X)	
N80 LABEL1:	
N...	

예 2: WAITP로 이동이 종료할 때까지 대기

팔레트 피드 장비

축 U: 팔레트 스토어
공작물 팔레트를 작업 영역으로 이송

축 V: 가공 지원을 위해 즉석 검사가 실시되는 측정 스테이션으로 라인 이송

프로그램 코드	설명
N10 FA[U]=100 FA[V]=100	; 개별 포지셔닝 축 U와 V의 축 관련 이송 속도 지정.
N20 POSA[V]=90 POSA[U]=100 G0 X50 Y70	; 포지셔닝 축 및 경로 축 이송.
N50 WAITP(U)	; 축 U가 N20에 프로그래밍된 종점에 도달할 때까지 프로그램 실행이 재개되지 않습니다.
…	

추가 정보

POSA를 사용한 이송

POSA는 블록 단계 활성화 또는 프로그램 실행에 영향을 미치지 않습니다. 후속 NC 블록이 실행되는 동안 종점까지 이송됩니다.

POS를 사용한 이송

POS에 프로그래밍된 모든 축이 끝점에 도달할 때까지 다음 블록이 실행되지 않습니다.

WAITP로 이동이 종료할 때까지 대기

WAITP 이후에는 NC 프로그램에 해당 축을 더 이상 지정할 수 없으며 이는 해당 축을 다시 프로그래밍할 때까지 적용됩니다. 이때 이축은 PLC를 통해 포지셔닝 축 또는 NC 프로그램/PLC 또는 HMI에서의 왕복 축으로 동작됩니다.

IPOBRKA 및 WAITMC를 통한 정지 동작 시 블록 변경

대기 라벨을 아직 수신하지 않았거나 다른 블록 종료 기준이 블록 변경을 방해하는 경우에만 축이 감속하게 됩니다. WAITMC 이후에 다른 블록 종료 기준이 블록 변경을 방해하지 않는 경우 축 이동이 즉시 시작됩니다.

3.7.3 위치 제어 스핀들 모드 (SPCON, SPCOF)

SPCON 또는 SPCOF 명령을 이용하는 스핀들의 위치 제어 작업은 명시적으로 활성화 또는 비활성화됩니다.

참고

SPCON 을 이용한 위치 제어 모드를 켜는 것은 최대 3개의 위치 제어 싸이클을 요구합니다.

구문

```
SPCON
SPCON(<n>)
SPCON(<n>,<m>,...)
SPCOF
SPCOF(<n>)
SPCOF(<n>,<m>,...)
```

의미

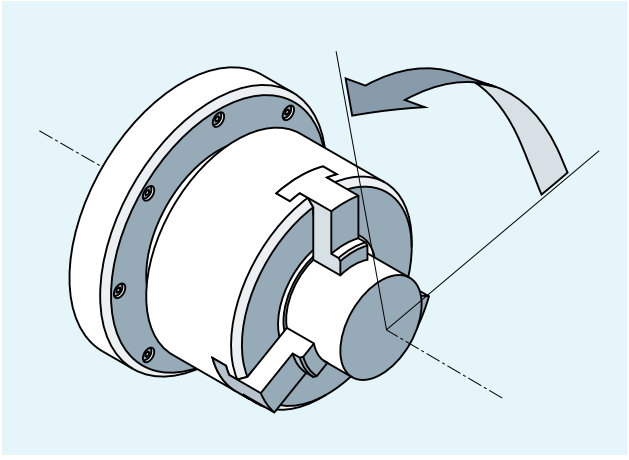
SPCON	위치 제어 모드 활성화 지정된 스핀들이 속도 제어에서 위치 제어로 전환됩니다. SPCON은 모달이고 SPCOF까지 유지됩니다.
SPCOF	위치 제어 모드 해제 지정된 스핀들이 위치 제어에서 속도 제어로 전환됩니다.
<n>,<m>,...	스핀들 번호 스핀들 번호를 지정하지 않음: 채널의 마스터 스핀들

참고

설정점 커플링이 적용된 동기 스핀들의 경우 리딩 스핀들이 SPCOF를 사용한 속도 제어 모드로 전환하면 안 됩니다.

3.7.4 스핀들 포지셔닝 (SPOS, SPOSA, M19, M70, WAITS)

SPOS, SPOSA 또는 M19를 사용하면, 예를 들어 공구 교환이 이뤄지는 동안 스핀들을 특정 각도 위치에 설정할 수 있습니다.



SPOS, SPOSA 및 M19는 다음 M3/M4/M5/M41~M45까지 스핀들을 위치 제어 모드로 일시적으로 전환시킵니다.

축 모드에서 포지셔닝

스핀들은 또한 머신 데이터에 지정된 주소에서 경로 축, 동기 축 또는 포지셔닝 축으로 동작할 수 있습니다. 해당 축 이름이 지정된 경우 스핀들은 축 모드로 동작합니다. M70은 스핀들을 축 모드로 직접 전환합니다.

포지셔닝 종료

FINEA, CORSEA, IPOENDA 또는 IPOBRKA를 사용하여 스핀들 포지셔닝 종료 기준을 프로그래밍할 수 있습니다.

전체 스핀들 또는 현재 블록에 프로그램된 전체 축의 모션 종료 기준 및 경로 보간에서 블록 변경 기준이 충족된 경우 프로그램은 다음 블록으로 진행합니다.

동기화

WAITS를 사용하여 스핀들 위치에 도달할 때까지 대기하게 함으로써 스핀들 이동을 동기화할 수 있습니다.

요구 사항

위치 지정할 스핀들은 반드시 위치 제어 모드에서 작동할 수 있어야 합니다.

구문

스핀들 포지셔닝:

`SPOS=<value>` / `SPOS[<n>]=<value>`

`SPOSA=<value>` / `SPOSA[<n>]=<value>`

`M19` / `M<n>=19`

스핀들을 축 모드로 전환:

`M70` / `M<n>=70`

모션 종료 기준 정의:

`FINEA`/`FINEA[S<n>]`

`COARSEA`/`COARSEA[S<n>]`

`IPOENDA`/`IPOENDA[S<n>]`

`IPOBRKA` / `IPOBRKA(<axis>[,<instant in time>])`; 별도의 NC 블록에서 프로그래밍.

스핀들 이동 동기화:

`WAITS`/`WAITS(<n>,<m>)` ; 별도의 NC 블록에서 프로그래밍.

의미

SPOS/SPOSA:	스핀들을 지정된 각도로 설정 SPOS 및 SPOSA는 기능적인 측면에서 동일하지만 블록 변경 특성에서 다음과 같은 차이가 있습니다. • SPOS의 경우 특정 위치에 도달해야만 NC 블록을 실행합니다. • SPOSA의 경우 특정 위치에 도달하지 않더라도 블록을 실행합니다.			
	<n>:	위치를 지정할 스핀들의 번호 스핀들 번호를 지정하지 않거나 "0"으로 지정한 경우 SPOS 또는 SPOSA는 마스터 스핀들에 적용됩니다.		
	<value>:	스핀들을 설정할 각도 위치		
		단위:	°	
		유형:	REAL	
		위치 접근 모드 프로그래밍에 사용할 수 있는 옵션은 다음과 같습니다.		
		=AC(<value>):	절대치	
			값 범위:	0 ... 359.9999
		=IC(<value>):	증분 치수	
			값 범위:	0 ... ±99 999.999
		=DC(<value>):	절대치로 직접 접근	
		=ACN(<value>):	절대 치수, 음의 방향으로 접근	
		=ACP(<value>):	절대 치수, 양의 방향으로 접근	
		=<value>:	DC(<value>)와 같음	
M<n>=19:	SD43250 $SA_M19_SPOSMODE에 의해 사전 설정된 위치 접근 모드를 이용하여 마스터 스핀들(M19 또는 M0=19)이나 번호가 <n>인 스핀들(M<n>=19)을 SD43240 $SA_M19_SPOS에 의해 사전 설정된 각도로 설정합니다. 위치에 도달할 때까지 NC 블록이 실행되지 않습니다.			

M<n>=70:	마스터 스핀들(M70 또는 M0=70)이나 번호가 <n>인 스핀들(M<n>=70)을 축 모드로 전환합니다. 정의된 위치로 접근하지 않습니다. 전환이 실행된 후 NC 블록이 활성화됩니다.		
FINEA:	"미세 정위치 정지"에 도달하면 모션 종료		
COARSEA:	"일반 정위치 정지"에 도달하면 이동이 끝남		
IPOENDA:	"보간자 정지"에 도달하면 모션 종료		
S<n>:	프로그래밍된 모션 종료 기준이 적용될 스핀들		
	<n>:	스핀들 번호	
	[S<n>]에 스핀들을 지정하지 않거나 스핀들 번호를 "0"으로 지정한 경우 프로그래밍된 모션 종료 기준은 마스터 스핀들에 적용됩니다.		
IPOBRKA:	정지 동작에서 블록 변경이 가능합니다.		
	<axis>:	채널 축 식별자	
	<시간 인스턴트>:	정지 동작을 기준으로 한 블록 변경 시간 인스턴트	
		단위:	%
		값 범위:	100 (정지 동작 적용 시점) ~0 (정지 동작 종료)
		<시간 인스턴트> 파라미터에 값을 지정하지 않은 경우 설정 데이터의 현재 값은 다음과 같이 적용됩니다. SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE **참고:** 시간 인스턴트가 "0"인 IBOBRKA는 IPOENDA와 동일합니다.	
WAITS:	지정된 스핀들의 동기화 명령어 SPOSA를 사용하여 이전 NC 블록에 지정된 스핀들이 미세 정위치 정지 지점에 도달할 때까지 후속 블록이 처리되지 않습니다.		
	M5 이후의 WAITS:	지정된 스핀들이 정지할 때까지 대기	
	M3/M4 이후의 WAITS:	지정된 스핀들이 지령치 속도에 도달할 때까지 대기	
	<n>,<m>:	동기화 명령어가 적용될 스핀들의 번호 스핀들 번호를 지정하지 않거나 "0"으로 지정한 경우 WAITS는 마스터 스핀들에 적용됩니다.	

참고

각각의 NC 블록에는 3개의 스핀들 위치를 사용할 수 있습니다.

참고

증분 치수 IC(<value>) 을 사용하면 스핀들 포지셔닝이 여러 번의 회전에 걸쳐 수행될 수 있습니다.

참고

SPOS에 앞서 SPCON으로 위치 제어가 활성화되면 SPCOF를 통해 위치 제어가 해제되기 전까지 계속 유지됩니다.

참고

시스템은 프로그램 순서에서 축 모드로의 트렌지션을 자동으로 감지합니다. 그러므로 가공 프로그램에서 M70을 프로그래밍할 필요가 없습니다. 하지만 가공 프로그램의 가독성 향상 등을 위해 M70을 프로그래밍할 수도 있습니다.

추가 정보

SPOSA로 포지셔닝

SPOSA에 의해 블록 인에이블 및 프로그램 실행에는 아무런 영향을 미치지 않습니다. 후속 NC 블록 실행 중에 스핀들 포지셔닝을 할 수 있습니다. 현재 블록에서 프로그램 된 모든 기능 (스핀들 제외) 들의 종료 기준에 도달한 경우 프로그램은 다음 블록으로 이동합니다. 스핀들 포지셔닝 작업은 몇 개의 블록에 걸쳐 프로그램 될 수 있습니다. (WAITS를 참조하시오.)

참고

암시적으로 전-처리를 중지시키는 명령어가 후속 블록에서 확인된 경우, 이 블록의 실행은 전체 포지셔닝 스핀들이 정지될 때 까지 지연됩니다.

SPOS/M19로 포지셔닝

블록에 프로그래밍된 모든 코드가 블록 종료 기준에 도달하고 (예: PLC이 모든 보조 코드를 인식하고 모든 축이 종점에 도달) 스핀들이 프로그래밍된 위치에 도달하면 블록 실행 조건이 충족됩니다.

이동 속도:

포지셔닝에 대한 지연 반응 및 속도는 머신 데이터에 저장됩니다. 설정된 값은 프로그래밍이나 동기 동작으로 수정할 수 있습니다. 다음을 참조하십시오.

- 포지셔닝 축/스핀들의 이송 속도 (FA, FPR, FPRAON, FPRAOF) (페이지 137)
- 프로그래밍이 가능한 가속 보정(ACC) (페이지 142)

스핀들 포지셔닝:

G90/G91 명령어는 여기에 적용되지 않으므로 상응하는 치수가 명시적으로 적용됩니다 (예: AC, IC, DC, ACN, ACP). 위치를 지정하지 않은 경우 DC를 기준으로 이송이 자동으로 이뤄집니다.

WAITS로 스핀들 이동 동기화

WAITS를 사용하여 이전 NC 블록에서 SPOSA로 프로그래밍된 하나 이상의 스핀들이 해당 위치에 도달할 때까지 NC 프로그램이 대기하는 지점을 확인할 수 있습니다.

예:

프로그램 코드	코멘트
N10 SPOSA[2]=180 SPOSA[3]=0	
...	
N40 WAITS(2,3)	; 스핀들 2 및 3 이 블록 N10에서 지정된 위치에 도달될 때까지 대기.

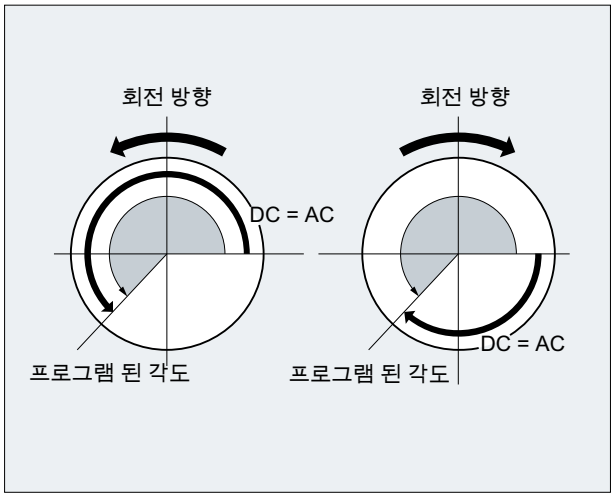
M5 이후에 WAITS를 사용하여 스핀들이 중지할 때까지 대기할 수 있습니다. M3/M4 이후에 WAITS를 사용하여 스핀들이 지정된 회전 속도/방향에 도달할 때까지 대기할 수 있습니다.

참고

스핀들이 아직 동기되지 않은 경우 머신 데이터에 설정된 것과 같이 (기본 설정) 양의 회전 방향이 적용됩니다.

회전(M3/M4) 중 스핀들 포지셔닝

M3 또는 M4가 활성화되면 스핀들은 프로그래밍된 값에서 정지하게 됩니다.



DC 및 AC 치수 사이에는 차이가 없습니다. 두 경우 모두 절대 끝 위치에 도달할 때까지 M3/M4에 의해 선택된 방향으로 계속 회전합니다. 필요한 경우 ACN 및 ACP를 통해 감속이 이뤄지고 적절한 접근 방향을 결정합니다. IC를 사용하면 스핀들이 현재 스핀들 위치에서 명시된 값을 향해 추가적으로 회전합니다.

정지 상태 (M5) 에서 스핀들 포지셔닝

프로그래밍된 거리만큼 정확하게 정지 상태 (M5) 에서 이송됩니다.

3.7.5 포지셔닝 축/스핀들의 이송 속도 (FA, FPR, FPRAON, FPRAOF)

또한 경로 축 및 동기 축 또는 개별 포지셔닝 축/스핀들에 대해 다른 로터리 축이나 스핀들에서 사용된 회전당 이송 속도를 사용할 수 있습니다.

공작물 이송 시스템, 공구 터렛 및 종단 지지대와 같은 포지셔닝 축은 경로 및 동기 축과 무관하게 이송됩니다. 따라서 각각의 포지셔닝 축에 별도의 이송 속도가 지정됩니다.

스핀들을 대상으로 별도의 축 이송 속도를 프로그래밍할 수도 있습니다.

구문

포지셔닝 축의 이송 속도:

```
FA[<axis>]=…
```

스핀들의 축 이송 속도:

```
FA[SPI(<n>)]=…
FA[S<n>]=…
```

경로 축/동기 축의 회전 당 이송 속도 도출:

```
FPR (<rotary axis>)
```

```
FPR(SPI(<n>))
FPR(S<n>)
```

포지셔닝 축/스핀들의 회전 당 이송 속도 도출:

```
FPRAON(<axis>,<rotary axis>)
FPRAON(<axis>,SPI(<n>))
FPRAON(<axis>,S<n>)
FPRAON(SPI(<n>),<rotary axis>)
FPRAON(S<n>,<rotary axis>)
FPRAON(SPI(<n>),SPI(<n>))
FPRAON(S<n>,S<n>)
FPRAOF(<axis>,SPI(<n>), etc.)
FPRAOF(<axis>,S<n>, etc.)
```

의미

FA[...]=...:	지정된 포지셔닝 축의 이송 속도 또는 지정된 스핀들의 포지셔닝 속도 (축 이송 속도)	
	단위:	mm/min, inch/min 또는 °/min
	값 범위:	... 999,999.999 mm/min, °/min ... 39 999.9999inch/min
FPR(...):	FPR을 사용하여 G95에 프로그래밍된 경로 축 및 동기 축의 회전 당 이송 속도를 도출할 로터리 축 (<rotary axis>) 또는 스핀들 (SPI(<n>)/S<n>) 을 확인할 수 있습니다.	
FPRAON(...):	포지셔닝 축 및 스핀들의 회전 이송 속도 도출 첫 번째 파라미터 (<축>/SPI(<n>)/S<n>) 는 회전 당 이송 속도로 이송될 포지셔닝 축 및 스핀들을 확인하는 데 사용됩니다. 두 번째 파라미터 (<rotary axis>/SPI(<n>)/S<n>) 는 회전 당 이송 속도를 도출할 로터리 축/스핀들을 확인하는 데 사용됩니다. **참고:** 이송 속도를 마스터 스핀들에서 도출한 경우에는 두 번째 파라미터는 생략할 수 있습니다.	

FPRAOF(...):	FPRAOF는 해당 축 또는 스핀들의 도출된 회전 당 이송 속도의 선택을 취소하는 데 사용됩니다.
<axis>:	축 이름 (포지셔닝 축 또는 기하 축)
SPI(<n>)/S<n>:	스핀들 이름 SPI(<n>)과 S<n>는 기능적인 측면에서 동일합니다.
	<n>: 스핀들 번호
	참고: SPI는 스핀들 번호를 축 이름으로 변환합니다. 전송 파라미터 (<n>) 에는 반드시 유효한 스핀들 번호가 포함돼야 합니다.

참고

프로그램 된 이송 속도 FA[...]는 모달입니다.

개별 NC 블록에서 포지셔닝 축 또는 스핀들에 최대 5개의 이송 속도를 프로그램 할 수 있습니다.

참고

도출된 이송 속도는 다음 공식에 의거 계산됩니다.

도출된 이송 속도 = 프로그래밍된 이송 속도 * 절대 마스터 이송 속도

예

예 1: 동기 스핀들 커플링

동기 스핀들 커플링을 통해 추종 스핀들의 포지셔닝 속도를 마스터 스핀들과 무관하게 프로그래밍할 수 있습니다 (예: 포지셔닝 작업에서).

프로그램 코드	코멘트
...	
FA[S2]=100	; 추종 스핀들 (스핀들 2) 의 포지셔닝 속도 = 100°/min
...	

예 2: 경로 축에 대해 도출된 회전 당 이송 속도

경로 축 X, Y는 반드시 로터리 축 A에서 도출된 회전 당 이송 속도로 이송되어야 합니다.

프로그램 코드
...
N40 FPR(A)
N50 G95 X50 Y50 F500

프로그램 코드
...

예 3: 마스터 스핀들의 회전 당 이송 속도 도출

프로그램 코드	코멘트
N30 FPRAON(S1,S2)	; 마스터 스핀들 (S1) 의 회전 당 이송 속도는 반드시 스핀들 2에서 도출해야 합니다.
N40 SPOS=150	; 마스터 스핀들을 포지셔닝합니다.
N50 FPRAOF(S1)	; 마스터 스핀들의 도출된 회전 당 이송 속도 선택을 취소합니다.

예 4: 포지셔닝 축의 회전 당 이송 속도 도출

프로그램 코드	코멘트
N30 FPRAON(X)	; 포지셔닝 축 X의 회전당 이송 속도는 반드시 마스터 스핀들로부터 도출되어야 합니다.
N40 POS[X]=50 FA[X]=500	; 포지셔닝 축은 500 mm/rev의 회전당 이송 속도로 이송합니다.
N50 FPRAOF(X)	

추가 정보

FA[...]

이송 속도 유형은 항상 G94입니다. G70/G71이 활성화된 경우 측정 단위는 머신 데이터의 디폴트 설정에 따라 미터/인치가 됩니다. G700/G710을 사용하여 프로그램의 측정 단위를 수정할 수 있습니다.

참고

FA를 프로그래밍하지 않은 경우 머신 데이터에 지정한 값이 적용됩니다.

FPR(...)

G95 명령어 (마스터 스핀들을 기준으로 하는 회전 당 이송 속도) 의 확장 기능인 FPR을 이용하여 선택된 모든 스핀들 또는 로터리 축에서 회전 당 이송 속도를 도출할 수 있습니다. G95 FPR (...) 은 경로 축 및 동기 축에 유효합니다.

FPR 명령으로 지정된 로터리 축/스핀들은 위치 제어로 동작하며 지령치 링크가 동작됩니다. 그렇지 않을 경우 실제값 링크가 동작합니다.

FPRAON(...)

FPRAON을 사용하여 다른 로터리 축이나 스핀들의 현재 이송 속도에서 포지셔닝 축 및 스핀들의 회전 당 이송 속도를 도출할 수 있습니다.

FPRAOF(...)

FPRAOF 명령어를 사용하면 하나 또는 여러 축/스핀들의 회전 당 이송 속도를 동시에 비활성화할 수 있습니다.

3.7.6 프로그램 가능한 이송 속도 오버라이드 (OVR, OVRRAP, OVRA)

NC 프로그램에서 경로/포지셔닝 축 및 스핀들의 속도를 수정할 수 있습니다.

구문

```
OVR=<value>
OVRRAP=<value>
OVRA[<axis>]=<value>
OVRA[SPI(<n>)]=<value>
OVRA[S<n>]=<value>
```

의미

OVR:	경로 이송 속도 F 수정	
OVRRAP:	급 이송 속도 수정	
OVRA:	포지셔닝 이송 속도 FA 또는 스핀들 속도 S 수정	
<axis>:	축 이름 (포지셔닝 축 또는 기하 축)	
SPI(<n>)/S<n>:	스핀들 이름 SPI(<n>) 과 S<n>는 기능적인 측면에서 동일합니다.	
	<n>:	스핀들 번호
	참고: SPI는 스핀들 번호를 축 이름으로 변환합니다. 전송 파라미터 (<n>) 에는 반드시 유효한 스핀들 번호가 포함돼야 합니다.	

<value>:	백분율 (%) 로 나타낸 이송 속도 수정 이 값은 기계 조작반에 설정된 피드 오버라이드를 기준으로 하거나 피드 오버라이드와 조합됩니다.	
	값 범위:	0 ... 200%, 필수
	참고: 경로 및 급 이송 오버라이드를 사용할 경우 머신 데이터에 설정된 최고 속도를 초과할 수 없습니다.	

3.7.7 프로그래밍이 가능한 가속 보정(ACC)

중요한 프로그램 부분에서는 예를 들어 기계 진동을 방지하기 위해 최고 값 아래로 가속을 제한하는 것이 필요할 수 있습니다.

프로그램 가능한 가속 오버라이드를 사용하여 NC 프로그램의 명령어로 각 경로 축 또는 스핀들의 가속도를 수정할 수 있습니다. 이 제한은 모든 종류의 보간에 적용됩니다. 머신 데이터에 정의된 값은 100% 가속도로 적용됩니다.

구문

```
ACC[<axis>]=<value>
ACC[SPI(<n>)]=<value>
ACC(S<n>)=<value>
```

해제:

```
ACC[...]=100
```

구문

ACC:	지정된 경로 축에 대한 가속도 변화 또는 지정된 스핀들에 대한 속도 변화	
<axis>:	경로 축의 채널 축 이름	
SPI(<n>)/S<n>:	스핀들 이름 SPI(<n>)과 S<n>는 기능적인 측면에서 동일합니다.	
	<n>:	스핀들 번호
	참고: SPI는 스핀들 번호를 축 이름으로 변환합니다. 전송 파라미터 (<n>) 에는 반드시 유효한 스핀들 번호가 포함돼야 합니다.	

<value>:	백분율 (%) 로 나타낸 가속도 변화 이 값은 기계 조작반에 설정된 피드 오버라이드를 기준으로 하거나 피드 오버라이드와 조합됩니다.	
	값 범위:	1~200%, 정수

참고

가속률이 더 높을 경우 제조업체에서 허용한 값을 초과할 수도 있으므로 주의하십시오.

예

프로그램 코드	코멘트
N50 ACC[X]=80	; X 방향의 축 슬라이드가 80% 가속도로만 이송됩니다.
N60 ACC[SPI(1)]=50	; 스핀들 1이 최고 가속도의 50%로만 가속 또는 감속됩니다.

추가 정보

ACC로 프로그래밍된 가속 오버라이드

ACC[...]로 프로그래밍된 가속 오버라이드는 시스템 변수 $AA_ACC에서와 같이 항상 출력이 고려됩니다. 가공 프로그램 및 동기 동작의 판독은 NC 처리 실행에서 여러 시간에 발생합니다.

가공 프로그램

그 사이에 동기 동작에 의해 ACC가 변경되지 않은 경우 가공 프로그램에 작성된 시스템 변수 $AA_ACC에서만 가공 프로그램에 작성된 값이 반영됩니다.

동기 동작

따라서 다음이 적용됩니다. 그 사이에 동기 동작에 의해 ACC가 변경되지 않은 경우 동기 동작에 기입된 시스템 변수 $AA_ACC에서만 동기 동작에 기입된 값이 반영됩니다.

지정된 가속도 동기 동작을 통해 변경될 수 있습니다.

예:

프로그램 코드
... N100 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140

실제 가속 값은 시스템 변수 $AA_ACC[<Axis>]를 사용하여 호출할 수 있습니다. 머신 데이터를 사용하여 마지막으로 설정된 ACC 값이 RESET/가공 프로그램 끝에 적용되어야 하는지 또는 100%로 설정되어야 하는지를 정할 수 있습니다.

3.7.8 핸드휠 오버라이드의 주입속도(FD, FDA)

FD 및 FDA 명령어를 사용하면 가공 프로그램이 실행되는 동안 핸드휠로 축을 이송할 수 있습니다. 그러면 경로 또는 속도 입력으로 평가된 핸드휠 펄스가 축 이송과 관련한 설정을 덮어쓰게 됩니다.

경로 축

경로 축의 경우, 프로그래밍된 경로 이송 속도를 덮어쓰기도 합니다. 채널의 첫 번째 기하 축으로 핸드휠을 평가합니다. 보간 싸이클별로 평가된 핸드휠 펄스는 회전 방향에 따라 달라지며, 오버레이 경고 속도에 해당합니다. 핸드휠 오버라이드를 통해 도달할 수 있는 경로 속도 한계값은 다음과 같습니다.

- 최소: 0
- 최대: 이송과 관련된 경로 축의 머신 데이터 한계값

참고

경로 이송 속도

경로 이송 속도 F와 핸드휠 이송 속도 FD는 같은 NC 블록에 프로그래밍할 수 없습니다.

포지셔닝 축

포지셔닝 축의 경우, 이송 경로 또는 속도를 축 관련 값으로 덮어쓸 수 있습니다. 축에 지정된 핸드휠이 산정됩니다.

- 경로 오버라이드
  회전 방향을 토대로 산정된 핸드휠 펄스는 이송할 축 경로와 일치합니다. 프로그래밍된 위치와 방향이 동일한 핸드휠 펄스만 산정됩니다.
- 속도 오버라이드
  보간 싸이클별로 평가되는 핸드휠 펄스는 회전 방향에 따라 달라지며, 오버레이된 축 속도에 해당합니다. 핸드휠 오버라이드를 통해 도달할 수 있는 경로 속도 한계값은 다음과 같습니다.
  - 최소: 0
  - 최대: 포지셔닝 축의 머신 데이터 한계값

구문

```
FD=<velocity>
FDA[<axis>]=<velocity>
```

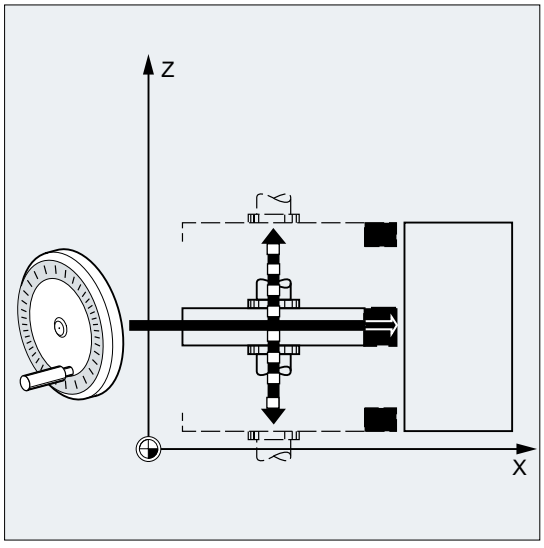
의미

FD=<velocity>:	핸드휠을 사용한 속도 오버라이드 실행 및 경로 이송 속도 <velocity>: • 값 = 0: 허용 안 됨! • 값 ≠ 0: 경로 속도
FDA[<axis>]=<velocity>:	축 이송 속도 <velocity>: • 값 = 0: 핸드휠로 경로 기본 설정 • 값 ≠ 0: 축 속도
<axis>:	포지셔닝 축의 축 이름

참고

FD와 FDA는 넌모달입니다.

예



경로 정의: Z 방향으로 왕복하는 연삭 휠은 핸드휠을 사용하여 X 방향으로 공작물에 이송됩니다.

스파크가 균일하게 발생할 때까지 운전자는 수동으로 이송을 계속할 수 있습니다. "잔여 거리 삭제"를 활성화하면 다음 NC 블록으로 전환하고 AUTOMATIC 모드로 가공이 계속됩니다.

추가 정보

속도 오버라이드를 사용한 경로 축 이송 (FD=<속도>)

경로 속도 오버라이드가 프로그래밍되는 가공 프로그램 블록은 반드시 다음 조건을 충족해야 합니다.

- 경로 명령어 G1, G2 또는 G3이 활성화돼야 합니다.
- 정위치 정지 명령어 G60이 활성화돼야 합니다.
- 직선 이송 속도 명령어 G94가 활성화돼야 합니다.

이송 속도 오버라이드

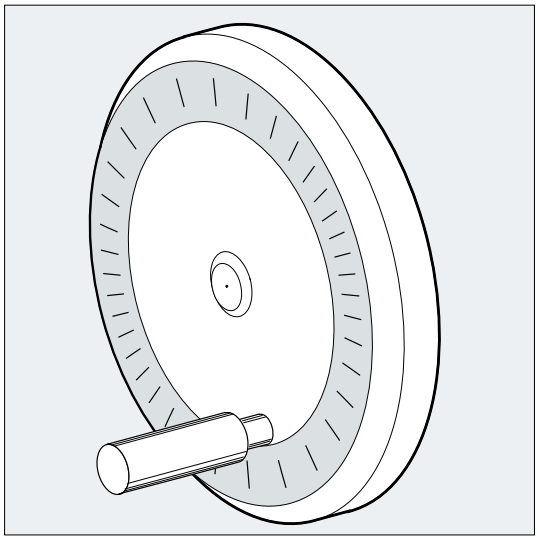
이송 속도 오버라이드는 프로그래밍된 경로 속도에만 영향을 미치며 핸드휠로 생성된 속도 구성요소에는 영향을 미치지 않습니다. (피드 오버라이드 = 0인 경우는 제외).

예:

프로그램 코드	설명
N10 X… Y… F500	; 이송 속도 = 500 mm/min
N20 X… Y… FD=700	; 이송 속도 = 700mm/min이며 핸드휠에 의한 속도 오버라이드. ; N20에서 500mm/min에서 700mm/min으로 가속. 핸드휠을 사용하여 회전 방향에 따라 0과 최대값 (머신 데이터) 사이로 속도를 변경할 수 있습니다.

경로 디폴트 설정과 포지셔닝 축 이송의 관계(FDA[<축>]=0)

FDA[<축>]=0이 설정된 NC 블록의 경우, 프로그램이 이송 동작을 유발하지 못하도록 피드가 0으로 설정됩니다. 대상 위치까지 프로그램된 이송은 핸드휠 회전에 의해서만 제어됩니다.



예:

프로그램 코드	설명
...	
N20 POS[V]=90 FDA[V]=0	; 목표 위치 = 90mm, 축 이송 속도 = 0mm/min 및 핸드휠로 경로 오버라이드. ; 블록 시작 시 축 V의 속도 = 0mm/min. ; 핸드휠 펄스를 사용하여 경로 및 속도 디폴트가 설정됩니다.

이동 방향, 이송 속도
핸드휠에 의해 해당 기호의 방향으로 설정된 경로로 축이 이동합니다. 회전 방향에 따라 전진 및 후진이 가능합니다. 핸드휠의 회전 속도가 빠를수록 이송 속도가 증가합니다.

이송 범위:
이송 범위는 시작 위치와 프로그래밍된 종점에 의해 제한됩니다.

속도 오버라이드와 포지셔닝 축 이송의 관계(FDA[<축>]=<속도>)

FDA[...]=...로 프로그래밍된 NC 블록에서, 최종 프로그래밍된 FA 값을 토대로 한 이송 속도는 FDA에 프로그래밍된 값으로 가속 또는 감속합니다. 현재 이송 속도 FDA에서 시작한 경우 핸드휠을 사용하면 프로그래밍된 이송 동작을 목표 위치까지 가속하거나 0으로 감속할 수 있습니다. 머신 데이터에 파라미터로 설정된 값은 최대 속도로 적용됩니다.

예:

프로그램 코드	설명
N10 POS[V]=… FA[V]=100	; 축 이송 속도 = 100mm/min

프로그램 코드	설명
N20 POS[V]=100 FAD[V]=200	; 축 목표 위치 = 100, 축 이송 속도 = 200mm/min ; 및 핸드휠을 이용한 속도 오버라이드. ; N20에서 100mm/min에서 200 mm/min으로 가속. 핸드휠을 사용하여 ; 회전 방향에 따라 0과 최대값 (머신 데이터) 사이로 속도를 ; 변경할 수 있습니다.

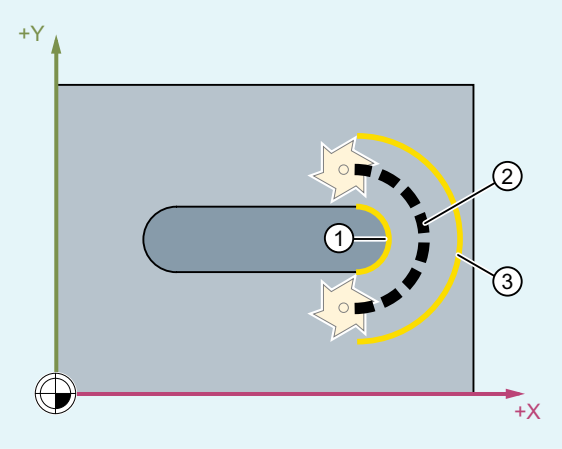
이송 범위:
이송 범위는 시작 위치와 프로그래밍된 종점에 의해 제한됩니다.

3.7.9 곡선 부분의 주입속도 최적화 (CFTCP, CFC, CFIN)

수정 모드 G41/G42가 활성화된 상태에서, 밀링 공구 반경의 프로그램된 주입속도는 먼저 밀링 공구 중심 경로를 기준으로 삼습니다("좌표 변형(프레임) (페이지 337)" 장 참조).

원호를 밀링하는 경우 특정 상황에서는 커터 이송 속도의 변화 정도가 중대한 영향을 미치게 되어 가공된 부품의 품질을 저하시킬 수 있습니다. (폴리노미얼 보간 및 스플라인 보간에도 동일한 원리가 적용됩니다.)

예: 작은 외부 반경을 큰 공구로 밀링. 밀링 공구의 외경이 이동해야 할 경로는 형상을 따라 이동하는 경로보다 상당히 길어지게 됩니다.



이런 이유로 형상 가공은 대단히 느린 이송 속도로 이뤄집니다. 부작용을 막으려면 곡선 형상에 맞게 이송 속도를 조정해야 합니다.

구문

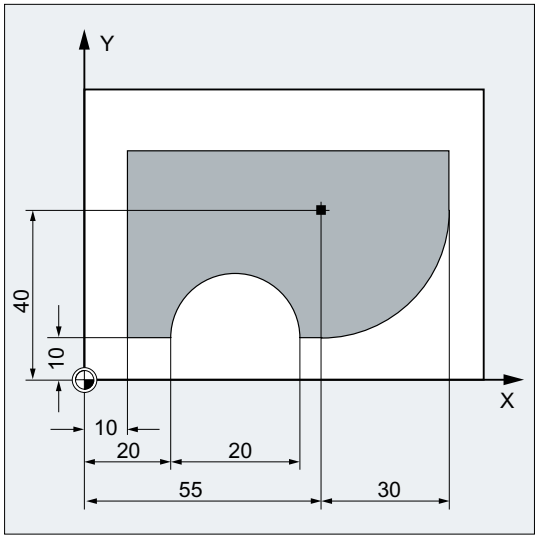
```
CFTCP
CFC
CFIN
```

의미

CFTCP:	밀링 커터 중심 경로의 일정한 이송 속도 제어 기능은 이송 속도를 일정하게 유지하고 이송 속도 오버라이드는 비활성화됩니다.
CFC:	형상에서 일정한 이송 속도 유지 (공구 절삭날) 이 기능은 디폴트로 사전 설정됩니다.
CFIN:	오목 형상에서 공구 절삭날이 일정한 이송 속도를 유지하고 그 밖의 경우에는 밀링 커터 중심 경로에서 일정한 이송 속도 유지 내경에서 이송 속도는 감소합니다.

예

이 예에서는 CFC 보정 이송 속도로 형상이 만들어집니다. 정삭 중, 베이스가 또한 CFIN으로 가공됩니다. 따라서 너무 높은 이송 속도에 의해 외부 반경에서 절삭 베이스가 손상되는 것이 방지됩니다.

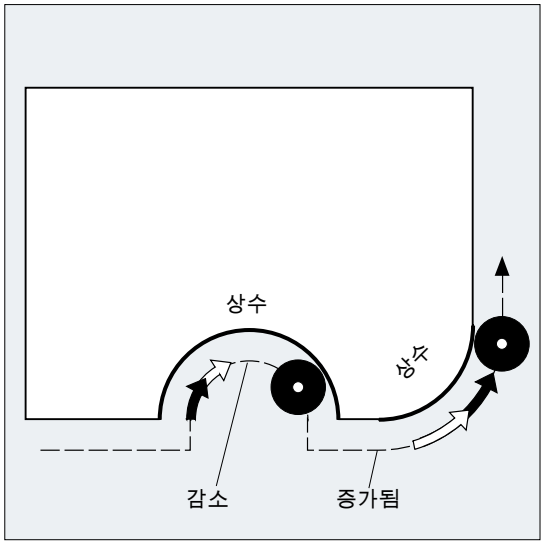


프로그램 코드	설명
N10 G17 G54 G64 T1 M6	
N20 S3000 M3 CFC F500 G41	
N30 G0 X-10	
N40 Y0 Z-10	; 첫 번째 절삭 깊이로 이송
N50 CONTOUR1	;서브프로그램 호출
N40 CFIN Z-25	; 두 번째 절삭 깊이로 이송
N50 CONTOUR1	;서브프로그램 호출

프로그램 코드	설명
N60 Y120	
N70 X200 M30	

추가 정보

CFC를 통한 형상과 함께 일정한 이송 속도



이송 속도는 내경에서는 감소하고 외경에서는 증가합니다. 이로 인해 공구 절삭날 및 형상 간에서 일정한 속도가 유지됩니다.

3.7.10 한 블록 내의 몇 가지 주입속도 (F, ST, SR, FMA, STA, SRA)

"한 블록에 복수의 이송 속도" 코드를 사용하여 외부 디지털 또는 아날로그 입력에 따라 NC 블록, 드웰 시간 또는 동시 후퇴 동작의 이송 속도 값을 여러 개 활성화할 수 있습니다.

구문

경로 모션
```
F=... F7=... F6=... F5=... F4=... F3=... F2=... ST=... SR=...
```

축 모션:
```
FA[<Ax>]=... FMA[7,<Ax>]=... FMA[6,<Ax>]=... FMA[5,<Ax>]=...
FMA[4,<Ax>]=... FMA[3,<Ax>]=... FMA[2,<Ax>]=... STA[<Ax>]=...
SRA[<Ax>]=...
```

의미

F=... :	경로 이송 속도는 어드레스 F에 프로그래밍되며 입력 신호가 없는 동안에 적용됩니다.	
	적용:	모달
F2=... to F7=... :	경로 이송 속도 외에도 최대 6개의 이송 속도를 블록에 추가로 프로그래밍할 수 있습니다. 숫자값은 변경 시 이송 속도를 활성화하는 입력 비트 번호를 나타냅니다.	
	적용:	넌모달
ST=... :	초 (s) 단위 축 드웰 시간 (연삭 기술의 경우: 스파크아웃 시간)	
	입력 비트:	1
	적용:	넌모달
SR=... :	후퇴 경로 후퇴 경로 단위는 현재 사용중인 측정 단위 (mm 또는 inch) 를 사용합니다.	
	입력 비트:	0
	적용:	넌모달
FA[<Ax>]=... :	축 이송 속도는 어드레스 FA에 프로그래밍되며 입력 신호가 없는 동안에 적용됩니다.	
	적용:	모달
FMA[2,<Ax>]=... to FMA[7,<Ax>]=... :	축 이송 속도 FA 외에 FMA를 사용하여 축당 최대 6개의 이송 속도를 블록에 추가로 프로그래밍할 수 있습니다. 첫 번째 파라미터는 입력의 비트 번호를 나타내며 두 번째 파라미터는 이송 속도가 적용될 축을 나타냅니다.	
	적용:	넌모달
STA[<Ax>]=...:	초 (s) 단위 축 드웰 시간 (연삭 공정에 사용: 스파크아웃 시간)	
	입력 비트:	1
	적용:	넌모달

SRA[<Ax>]=...:	축 후퇴 경로	
	입력 비트:	0
	적용:	넌모달
<Ax>:	이송 속도가 적용되는 축	

참고

신호 우선 순위

입력 비트 0 (I0) 부터 오름차순으로 신호를 스캔합니다. 따라서 후퇴 모션이 우선 순위가 가장 높고 이송 속도 F7이 우선 순위가 가장 낮습니다. 드웰 시간과 후퇴 모션은 F2~F7로 활성화한 이송 속도 모션을 중단시킵니다.

우선 순위가 가장 높은 순위가 실제 이송 속도를 결정합니다.

참고

이동할 거리 삭제

입력 비트 1이 드웰 시간에 대해 활성화되거나 비트 0이 복귀 경로에 대해 활성화되는 경우 경로 축 또는 개별 축의 잔여 이송 거리가 삭제되고 드웰 시간 또는 복귀가 시작됩니다.

참고

후퇴 경로

후퇴 경로 단위는 현재 사용중인 측정 단위 (mm 또는 inch) 를 사용합니다.

역 스트로크는 항상 현재 모션의 반대 방향에서 실행됩니다. SR/SRA는 항상 역 스트로크의 값만 프로그래밍합니다. 부호는 프로그래밍하지 않습니다.

참고

POSA 대신 POS

외부 입력 때문에 축에 대해 이송 속도, 드웰 시간 또는 복귀 경로를 프로그래밍하는 경우, 이 블록에서 이 축을 POSA 축 (여러 블록에 걸친 포지셔닝 축) 으로 프로그래밍할 수 없습니다.

참고

상태 조회

각종 축의 동기 명령에 대한 입력 상태를 조회할 수 있습니다.

참고

선행 제어

한 블록에 복수의 이송 속도에 대해 선행 제어도 활성화됩니다. 이처럼 현재 이송 속도는 선행 제어 값에 의해 제한될 수 있습니다.

예

예 1: 경로 모션

프로그램 코드	코멘트
G1 X48 F1000 F7=200 F6=50 F5=25 F4=5 ST=1.5 SR=0.5	; 경로 이송 속도 = 1000 ; 추가 경로 이송 속도 값: ; 200 (입력 비트 7) ; 50 (입력 비트 6) ; 25 (입력 비트 5) ; 5 (입력 비트 4) ; 드웰 시간 1.5초 ; 후퇴 0.5 mm

예 2: 축 모션

프로그램 코드	코멘트
POS[A]=300 FA[A]=800 FMA[7,A]=720 FMA[6,A]=640 FMA[5,A]=560 STA[A]=1.5 SRA[A]=0.5	; 축 A의 이송 속도 = 800 ; 축 A의 추가 이송 속도 값: 720 (입력 비트 7) ; 640 (입력 비트 6) ; 560 (입력 비트 5) ; 축 드웰 시간: 1.5 s ; 축 후퇴: 0.5 mm

예 3: 한 블록에 복수의 연산

프로그램 코드	코멘트
N20 T1 D1 F500 G0 X100	초기 설정
N25 G1 X105 F=20 F7=5 F3=2.5 F2=0.5 ST=1.5 SR=0.5	; F를 사용한 정상 이송 속도, ; F7을 사용한 황삭, ; F3을 사용한 정삭, ; F2를 사용한 스무딩 정삭, ; 드웰 시간 1.5초, ; 후퇴 경로 0.5mm
...	

3.7.11 넌모달 이송 속도 (FB)

"넌모달 이송 속도" 코드를 이용하여 싱글 블록의 이송 속도를 별도로 지정할 수 있습니다. 이 블록 후 이전 모달 이송 속도가 다시 활성화됩니다.

구문

```
FB=<value>
```

의미

FB:	현재 블록 전용 이송 속도
<VALUE>:	프로그래밍된 값은 반드시 0보다 커야 합니다. 값은 다음과 같은 활성 상태의 이송 속도 유형을 토대로 해석됩니다. • G94: mm/min 또는 deg/min 단위의 이송 속도 • G95: mm/rev 또는 inch/rev 단위의 이송 속도 • G96: 주속 일정 제어

참고

블록에 이송 모션이 프로그래밍되지 않은 경우 (예: 계산 블록), FB는 영향을 미치지 않습니다.

모따기/라운딩의 이송 속도가 명확하게 프로그래밍되지 않은 경우 FB 값 역시 이 블록의 모따기/라운딩 형상 구성 요소에 적용됩니다.

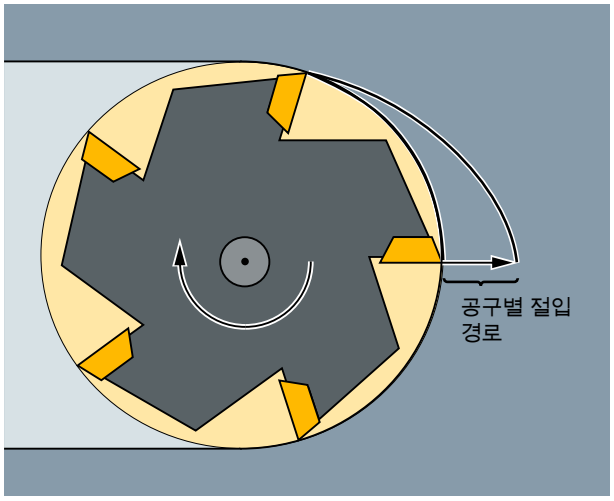
FLIN, FCUB 등의 이송 속도 보간도 제약 없이 가능합니다.

FB 및 FD (이송 속도 오버라이드로 핸드휠 이동) 또는 F (모달 경로 이송 속도) 를 동시에 프로그래밍하는 것은 **불가능**합니다.

예

프로그램 코드	코멘트
N10 G0 X0 Y0 G17 F100 G94	;초기 설정
N20 G1 X10	; 이송 속도 100 mm/min
N30 X20 FB=80	; 이송 속도 80 mm/min
N40 X30	; 이송 속도는 다시 100mm/min가 됩니다.
...	

3.7.12 절삭날 이송 속도 (G95 FZ)



시스템은 활성 상태의 공구 옵셋 데이터 블록과 관련된 $TC_DPNT (절삭날 수) 공구 파라미터를 사용하여 각 이송 블록의 회전 당 유효 이송 속도를 절삭날 이송 속도에서 산출합니다.

F = FZ * $TC_DPNT

여기서

F:	회전 당 이송 속도 (mm/rev 또는 inch/rev)
FZ:	절삭날 이송 속도 (mm/tooth 또는 inch/tooth)
$TC_DPNT:	시스템 변수 공구 파라미터: 회전 당 절삭날 수

활성 공구의 종류 ($TC_DP1) 는 고려되지 않습니다.

프로그래밍된 절삭날 이송 속도는 공구 교환 및 공구 옵셋 데이터 블록 선택/선택 취소의 영향을 받지 않으며 모달 형식으로 유지됩니다.

활성 상태의 공구 절삭날과 관련된 $TC_DPNT 공구 파라미터를 변경하면 공구 옵셋을 선택하거나 활성 상태의 옵셋 데이터를 업데이트한 다음부터 변경사항이 적용됩니다.

공구를 교환하거나 공구 옵셋 데이터 블록을 선택 또는 선택 취소하면 유효 회전당 이송 속도가 다시 계산됩니다.

참고

절삭날 이송 속도에는 경로만 반영됩니다. (특정 축의 프로그래밍은 불가능합니다.)

구문

```
G95 FZ...
```

의미

G95:	이송 속도 유형: 회전 당 이송 속도 (mm/rev 또는 inch/rev). (G700/G710에 따라 다름) G95의 경우 "이송 속도 (G93, G94, G95, F, FGROUP, FL, FGREF) (페이지 117)" 참조	
FZ:	절삭날 이송 속도	
	활성화:	G95 사용
	효율성:	모달
	단위:	mm/tooth 또는 inch/tooth (G700/G710에 따라 다름)

유의사항

공구 교환/마스터 스핀들 변경

사용자는 반드시 상응하는 프로그래밍 작업 (예: FZ 다시 프로그래밍) 을 통해 후속 공구 교환이나 마스터 스핀들 변경을 고려해야 합니다.

유의사항

공구 운전 미정의

경로 구조 (직선, 원호 등) 외에도 하향 밀링, 상향 밀링, 페이스 밀링, 원주 밀링 등과 같은 기술적 사항은 자동으로 반영되지 않습니다. 그러므로 절삭날 이송 속도를 프로그래밍할 때 이 요소들을 고려해야 합니다.

참고

G95 F... 또는 G95 FZ...로 전환

G95 F...(회전 이송 속도)와 G95 FZ...(공구 이송 속도) 간의 전환에 따라 비활성화된 이송 속도 값이 사안별로 삭제됩니다.

참고

FPR로 이송 속도 도출

회전 당 이송 속도의 경우와 마찬가지로 FPR을 사용하여 원하는 로터리 축이나 스핀들의 절삭날 이송 속도를 도출할 수 있습니다 ("포지셔닝 축/스핀들의 이송 속도 (FA, FPR, FPRAON, FPRAOF) (페이지 137)" 참조).

예

예 1: 절삭날이 5개인 밀링 커터($TC_DPNT = 5)

프로그램 코드	설명
N10 G0 X100 Y50	
N20 G1 G95 FZ=0.02	; 절삭날 이송 속도 0.02 mm/tooth
N30 T3 D1	; 공구를 로드하고 공구 옵셋 데이터 블록을 활성화합니다.
M40 M3 S200	; 스핀들 속도 200rpm
N50 X20	; 밀링 조건:
	FZ = 0.02mm/tooth
	적용되는 회전 당 이송 속도:
	F = 0.02mm/tooth * 5teeth/rev = 0.1mm/rev
	또는
	F = 0.1mm/rev * 200rpm = 20mm/min
...	

예 2: G95 F... 또는 G95 FZ...로 전환

프로그램 코드	설명
N10 G0 X100 Y50	
N20 G1 G95 F0.1	; 회전 당 이송 속도 0.1 mm/rev
N30 T1 M6	
N35 M3 S100 D1	
N40 X20	
N50 G0 X100 M5	
N60 M6 T3 D1	; 예를 들어 절삭날이 5개인 공구를 로드합니다 ($TC_DPNT = 5).
N70 X22 M3 S300	
N80 G1 X3 G95 FZ=0.02	; G95 F…를 G95 FZ…로 변경하고 0.02mm/tooth 상태의 절삭날 이송 속도를 활성화합니다.
...	

예 3: 스핀들의 절삭날 이송 속도 도출 (FBR)

프로그램 코드	설명
…	
N41 FPR(S4)	; 스핀들 4의 공구 (마스터 스핀들 아님).
N51 G95 X51 FZ=0.5	; 스핀들 S4에 근거한 절삭날 이송 속도 0.5 mm/tooth
...	

예 4: 후속 공구 교환

프로그램 코드	설명
N10 G0 X50 Y5	
N20 G1 G95 FZ=0.03	; 절삭날 이송 속도 0.03 mm/tooth
N30 M6 T11 D1	; 예를 들어 절삭날이 7개인 공구를 로드합니다 ($TC_DPNT = 7).

프로그램 코드	설명
N30 M3 S100	
N40 X30	; 적용되는 회전 당 이송 속도 0.21 mm/rev
N50 G0 X100 M5	
N60 M6 T33 D1	; 예를 들어 절삭날이 5개인 공구를 로드합니다 ($TC_DPNT = 5).
N70 X22 M3 S300	
N80 G1 X3	; 모달 절삭날 이송 속도 0.03 mm/tooth, 적용되는 회전 당 이송 속도 0.15 mm/rev
...	

예 5: 마스터 스핀들 변경

프로그램 코드	설명
N10 SETMS (1)	; 스핀들 1이 마스터 스핀들이 됩니다.
N20 T3 D3 M6	; 공구 3이 스핀들 1로 변경됩니다.
N30 S400 M3	; 스핀들 1 (결과적으로 T3) 의 속도 S400
N40 G95 G1 FZ0.03	; 절삭날 이송 속도 0.03 mm/tooth
N50 X50	; 경로 모션. 유효 이송 속도는 다음 조건에 따라 달라집니다. - 절삭날 이송 속도 FZ - 스핀들 1의 속도 - 활성 공구 T3의 절삭날 개수
N60 G0 X60	
...	
N100 SETMS(2)	; 스핀들 2가 마스터 스핀들이 됩니다.
N110 T1 D1 M6	; 공구 1이 스핀들 2로 변경됩니다.
N120 S500 M3	; 스핀들 2 (결과적으로 T1) 의 속도 S500
N130 G95 G1 FZ0.03 X20	; 경로 모션. 유효 이송 속도는 다음 조건에 따라 달라집니다. - 절삭날 이송 속도 FZ - 스핀들 2의 속도 - 활성 공구 T1의 절삭날 개수

참고

마스터 스핀들에서 교환한 후 (N100) 스핀들 2에 의해 활성화된 공구를 교체해야 합니다 (N110).

추가 정보

G93, G94 또는 G95로 전환

G95가 활성화되지 않은 경우 FZ를 프로그래밍할 수도 있지만 G95를 선택하면 FZ가 적용되지 않고 삭제됩니다 . 다시 말해서, G93, G94 또는 G95로 전환하면 F와 동일한 방식으로 FZ 값도 삭제됩니다.

G95 다시 선택

G95가 이미 활성화된 상태에서 G95를 선택해도 (F나 FZ로 전환하도록 프로그래밍하지 않은 한) 아무런 영향을 미치지 않습니다.

넌모달 이송 속도 (FB)

G95 FZ... (모달) 가 활성화된 경우 넌모달 이송 속도 FB...는 절삭날 이송 속도로 해석됩니다.

SAVE 메커니즘

SAVE 속성이 존재하는 서브프로그램에서는 (F와 동일한 방식으로) 서브프로그램 시작에 앞서 FZ가 값으로 기록됩니다.

한 블록에 복수의 이송 속도 값

절삭날 이송 속도와 관련해서는 "한 블록에 복수의 이송 속도 값" 코드를 사용할 수 없습니다.

동기 동작

FZ는 동기 동작에 프로그래밍할 수 없습니다.

절삭날 이송 속도 및 경로 이송 속도 형식 읽기

시스템 변수를 사용하여 절삭날 이송 속도 및 경로 이송 속도 유형을 읽을 수 있습니다.

- 다음 시스템 변수를 통한 가공 프로그램의 전처리 정지:

	$AC_FZ	현재 메인 실행 블록이 사전 처리될 때 적용되는 절삭날 당 이송 속도.	
	$AC_F_TYPE	현재 메인 실행 블록이 사전 처리될 때 적용되는 경로 이송 속도.	
		값:	**의미:**
		0	mm/min
		1	mm/rev.
		2	inch/min
		3	inch/rev.
		11	mm/tooth
		33	inch/tooth

- 다음 시스템 변수를 통한 가공 프로그램의 전처리 정지 안 함:

	$P_FZ	프로그래밍된 절삭날 이송 속도	
	$P_F_TYPE	프로그래밍된 경로 이송 속도 유형	
		값:	**의미:**
		0	mm/min
		1	mm/rev.
		2	inch/min
		3	inch/rev.
		11	mm/tooth
		33	inch/tooth

참고

G95가 활성화되지 않은 경우 $P_FZ 및 $AC_FZ 변수가 항상 0의 값을 반환합니다.

3.8 형상 설정

3.8.1 셋터블 워크 옵셋 (G54~G57, G505~G599, G53, G500, SUPA, G153)

G54~G57 및 G505~G599 명령은 사용자 인터페이스에서 설정한 기본 좌표계와 비교해 공작물 좌표계의 설정 가능한 워크 옵셋 값을 활성화합니다.

구문

스위치 ON:

```
G54
...
G57
G505
...
G599
```

스위치 OFF 또는 억제:

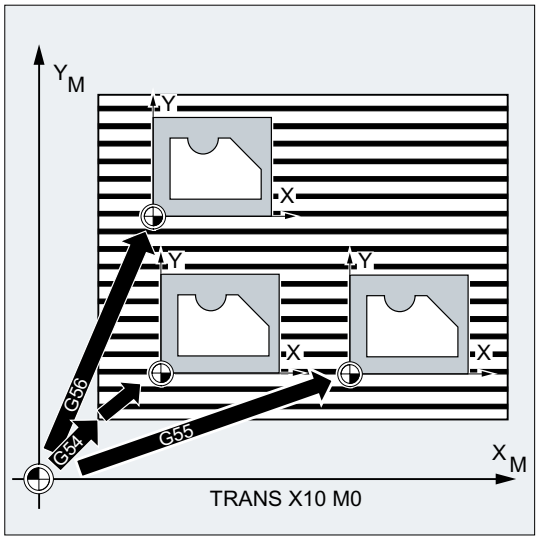
```
G500 / G53 / G153 / SUPA
```

의미

G54~G57:	첫 번째부터 네 번째 설정 가능한 워크 옵셋(WO) 호출	
G505~G599:	다섯 번째부터 99 번째 설정 가능한 워크 옵셋 호출	
G500:	설정 가능한 워크 옵셋 해제	
	G500=제로 프레임: (기본 설정: 옵셋, 회전, 미러 또는 스케일링 미포함)	다음 호출 때까지 설정 가능한 워크 옵셋 해제, 기본 프레임 활성화 ($P_ACTBFRAME)
	G500이 0이 아닌 경우:	첫 번째 설정 가능한 워크 옵셋 ($P_UIFR[0]) 및 전체 기본 프레임 활성화($P_ACTBFRAME) 또는 수정된 기본 프레임 활성화.
G53:	G53은 좌표 설정 및 프로그램 워크 옵셋을 넌모달로 억제합니다.	
G153:	G153 은 G53과 같은 효능을 가지며 전체 기본 프레임을 억제합니다.	
SUPA:	SUPA는 G153과 같은 효능을 가지며 또한 다음을 억제합니다: • 핸드휠 옵셋 (DRF) • 오버레이 이동 • 외부 워크 옵셋 • PRESET 옵셋	

예

워크 옵셋 값 G54~G56에 따라 팔레트에 정렬된 3개의 공작물이 순차적으로 가공됩니다. 가공 순서는 서브프로그램 L47에 프로그래밍합니다.



프로그램 코드	설명
N10 G0 G90 X10 Y10 F500 T1	; 접근
N20 G54 S1000 M3	; 첫 번째 WO 호출, 스핀들 CW 회전
N30 L47	; 서브프로그램으로 프로그램 통과
N40 G55 G0 Z200	; 두 번째 WO 호출, 급 이송으로 Z축 이동
N50 L47	; 서브프로그램으로 프로그램 통과
N60 G56	; 세 번째 WO 호출
N70 L47	; 서브프로그램으로 프로그램 통과
N80 G53 X200 Y300 M30	; 워크 옵셋 해제, 프로그램 끝

3.8.2 좌표 설정 (G54~G57, G505~G599, G53, G500, SUPA, G153): 추가 정보

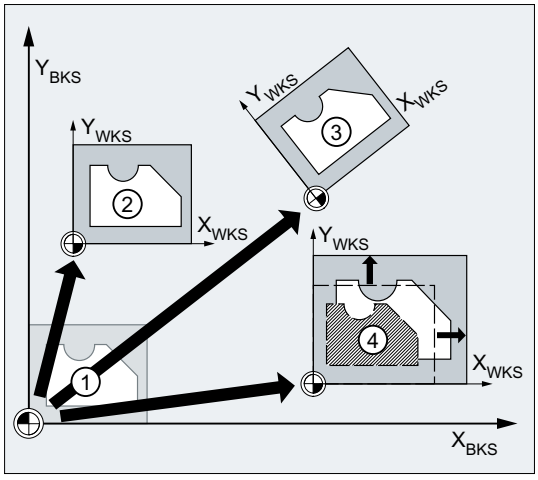
추가 정보

설정 가능한 워크 옵셋

원칙적으로 설정 가능한 워크 옵셋은 설정된 프레임 (페이지 337)입니다. 따라서 다음 콤포넌트와 프레임 값 역시 설정 가능한 워크 옵셋에 사용할 수 있습니다.

- 옵셋
- 회전

- 스케일링
- 배율



① BCS에서 초기 위치
② 옵셋
③ 옵셋 + 회전
④ 옵셋 + 스케일링

설정 가능한 워크 옵셋의 프레임 값은 사용자 인터페이스에서 입력합니다.

SINUMERIK Operate: '파라미터' > '워크 옵셋' > '세부사항' 작업 영역

파라미터로 설정된 파라미터 프레임 (G505~G599) 개수

사용자별 설정 가능한 워크 옵셋(G505~G599)의 개수는 다음 구문을 사용해 채널별로 설정할 수 있습니다.

MD28080 $MC_MM_NUM_USER_FRAMES = <개수>

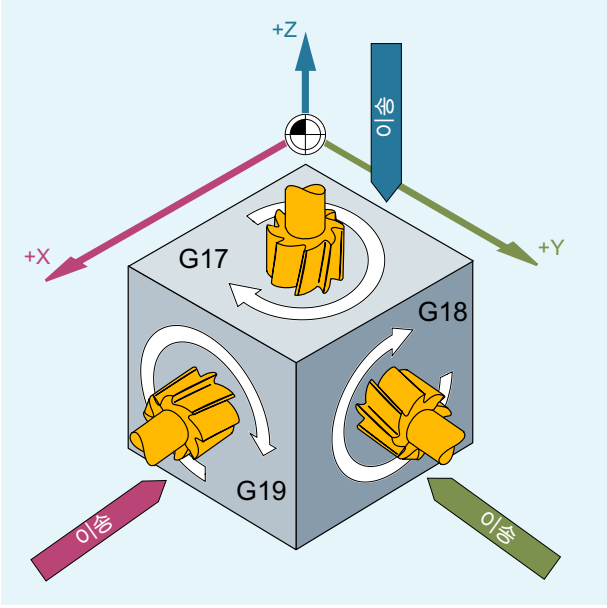
다음 또한 참고:

프로그래밍 워크 옵셋 (G58, G59) (페이지 346)

3.8.3 작업 평면 선택 (G17/G18/G19)

원하는 형상을 가공하기 위해 작업 평면을 정의할 경우 다음과 같은 기능도 함께 정의됩니다.

- 공구 반경 보정 평면
- 공구 종류에 따라 상이한 공구 길이 옵셋의 절입 방향
- 원호 보간의 평면



구문

```
G17/G18/G19 등
```

의미

G17:	X/Y 작업 평면 절입 방향 Z, 첫 번째 - 두 번째 기하 축의 평면 선택
G18:	X/Z 작업 평면 절입 방향 Y, 첫 번째 - 세 번째 기하 축의 평면 선택
G19:	Y/Z 작업 평면 절입 방향 X, 두 번째 - 세 번째 기하 축의 평면 선택

참고

기본 설정으로 밀링은 G17 (X/Y 평면), 선삭은 G18 (Z/X 평면) 이 정의됩니다.

공구 경로 보정 G41/G42 ("공구 반경 보정 (페이지 277)" 절 참조) 를 호출할 경우 제어 시스템이 공구 길이 및 반경을 보정할 수 있도록 반드시 작업 평면을 정의해야 합니다.

예

밀링의 "전통적" 접근 방식:

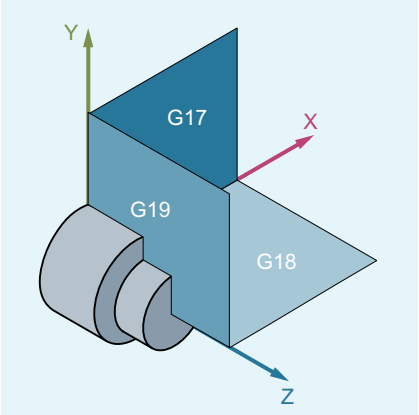
1. 작업 평면을 정의합니다 (밀링의 기본 G17).
2. 공구 종류 (T) 와 공구 옵셋 값 (D) 을 선택합니다.
3. 경로 보정 (G41) 을 선택합니다.
4. 이송 동작을 프로그램 합니다.

프로그램 코드	코멘트
N10 G17 T5 D8	; 작업 평면 X/Y의 호출, 공구 호출. Z 방향에서 공구 길이 옵셋 수행
N20 G1 G41 X10 Y30 Z-5 F500	; X/Y 평면에서 반경 보정이 적용됩니다.
N30 G2 X22.5 Y40 I50 J40	; X/Y 평면의 원호 보간/공구 반경 보정.

추가 정보

일반

프로그램을 시작할 때 작업 평면을 G17에서 G19까지 선택할 것을 권장합니다. 기본 설정으로 선삭 G18에는 Z/X 평면이 사전 설정되어 있습니다.

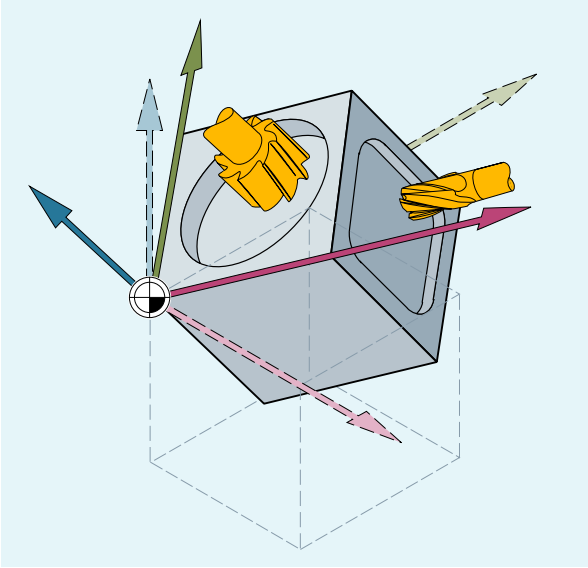


회전 방향을 계산하기 위해 제어 시스템에 작업 평면 지정이 필요합니다. 자세한 정보는 "원호 보간 (페이지 209)" 섹션에서 확인할 수 있습니다.

경사면 기계가공

경사 표면 좌표 축 위치 지정을 위해 ROT (페이지 348)로 좌표계를 회전하십시오. 작업 평면은 이에 따라 회전합니다.

경사면에의 공구 길이 보정



일반적인 규칙으로 공구 길이 보정은 항상 고정되어 회전하지 않는 작업 평면을 기준합니다.

참고

공구 길이 보정 구성 요소는 "방향 조정가능 공구 길이 보정"의 기능으로 회전된 작업 평면에 따라 계산될 수 있습니다.

보정 평면은 CUT2D, CUT2DF로 선택합니다. 자세한 정보와 사용 가능한 계산 방법에 관한 설명은 "공구 반경 보정 (페이지 277)" 장에서 확인할 수 있습니다.

제어 시스템은 작업 평면의 공간적 정의에 편리한 좌표 변환 기능을 제공합니다. 자세한 정보는 "좌표 변형(프레임) (페이지 337)" 섹션에서 확인할 수 있습니다.

3.8.4 치수

대부분의 NC 프로그램의 기초는 특정 치수의 공작물 도면입니다.

다음은 일반 치수입니다.

- 절대치 또는 증분치
- 밀리미터 또는 인치
- 반경 또는 직경(선삭의 경우)

특정 프로그래밍 명령어가 다양한 치수 옵션에 유효하여 치수 도면에서의 데이터는 NC 프로그램에 직접(변환 작업 없이) 이송될 수 있습니다.

3.8.4.1 절대치(G90, AC)

절대치를 사용하여 위치 지정은 항상 현재 사용중인 좌표계의 영점을 기준합니다. 즉 절대 위치는 공구가 이송될 위치를 프로그램 합니다.

모달 절대치

모달 절대치는 G90 명령어로 활성화됩니다. 일반적으로 이는 후속 NC 블록에서 프로그래밍된 모든 축에 적용됩니다.

넌모달 절대치

증분치 모드 (G91) 를 사용하여 개별 축의 넌모달 절대치를 설정하기 위해 AC 명령을 사용합니다.

참고

스핀들 포지셔닝 (SPOS, SPOSA) 및 보간 파라미터 (I, J, K) 에도 넌모달 절대치 (AC) 를 사용할 수 있습니다.

구문

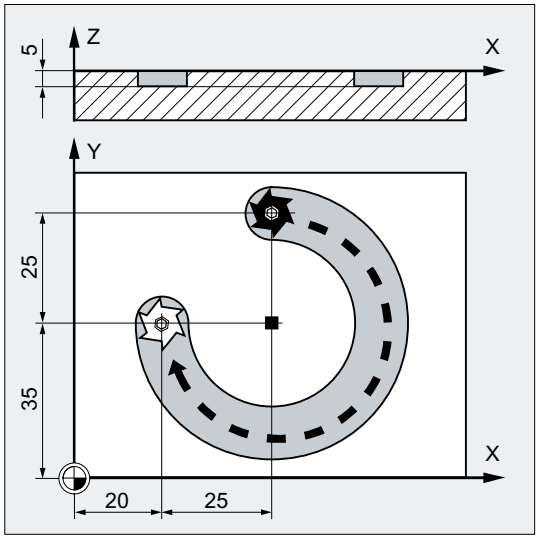
```
G90
<axis>=AC(<value>)
```

의미

G90:	모달 절대치 활성 명령
AC:	넌모달 절대치 활성 명령
<axis>:	이송될 축의 축 이름
<value>:	절대치로 이송될 축의 위치 지령치

예

예 1: 밀링

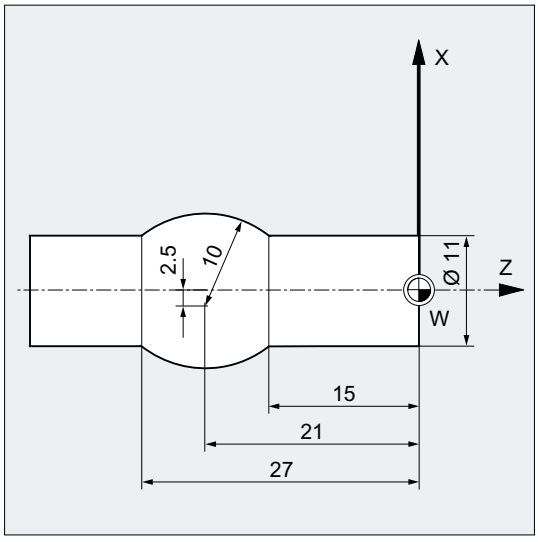


프로그램 코드	코멘트
N10 G90 G0 X45 Y60 Z2 T1 S2000 M3	; 절대 치수 입력, 급 이송으로 XYZ 위치로 이동, 공구 선택, CW 방향 스핀들 회전.
N20 G1 Z-5 F500	; 직선 보간, 공구 이송.
N30 G2 X20 Y35 I=AC(45) J=AC(35)	; CW 방향 원호 보간, 절대치 원호 종점 및 원호 중심점 지정
N40 G0 Z2	; 이송
N50 M30	; 블록 종료

참고

원호 중심점 I 및 J에 대한 자세한 정보는 "원호 보간" 단락을 참조 바랍니다.

예 2: 선삭



프로그램 코드	코멘트
N5 T1 D1 S2000 M3	; T1 공구 선택, CW 방향 스핀들 회전
N10 G0 G90 X11 Z1	; 절대치 입력, 급 이송으로 XZ 위치로 이동
N20 G1 Z-15 F0.2	; 직선 보간, 공구 이송.
N30 G3 X11 Z-27 I=AC(-5) K=AC(-21)	; CCW 방향 원호 보간, 절대치 원호 종점 및 원호 중심점 지정
N40 G1 Z-40	; 이송
N50 M30	; 블록 종료

참고

원호 중심점 I 및 J에 대한 자세한 정보는 "원호 보간" 단락을 참조 바랍니다.

다음 또한 참고:

선삭과 밀링의 절대와 증분 치수(G90/G91) (페이지 173)

3.8.4.2 증분 치수(G91, IC)

증분 치수로 위치를 지정할 경우 항상 이전의 마지막 지점을 기준으로 합니다. 즉, 증분 치수의 프로그래밍은 공구가 이송하게 될 거리의 차이를 나타냅니다.

모달 증분 치수

모달 증분 치수는 G91 명령어로 활성화됩니다. 일반적으로 이는 후속 NC 블록에서 프로그래밍된 모든 축에 적용됩니다.

넌모달 증분 치수

절대치 모드 (G90) 를 사용하여 개별 축의 넌모달 증분 치수를 설정하기 위해 IC 명령을 사용합니다.

참고

스핀들 포지셔닝 (SPOS, SPOSA) 및 보간 파라미터 (I, J, K) 에도 넌모달 증분 치수 (IC) 를 사용할 수 있습니다.

구문

```
G91
<axis>=IC(<value>)
```

의미

G91:	모달 증분 치수 활성 명령
IC:	넌모달 증분 치수 활성 명령
<axis>:	이송될 축의 축 이름
<value>:	증분 치수로 이송될 축의 위치 지령치

G91 확장

스크래칭 등의 특정 작업의 경우, 증분치로만 프로그램된 거리가 이송되어야 합니다. 활성 워크 옵셋 또는 공구 길이 옵셋은 이송되지 않습니다.

이 동작은 다음 셋팅 데이터를 통하여 활성 워크 옵셋 및 공구 길이 옵셋에 별도로 설정할 수 있습니다.

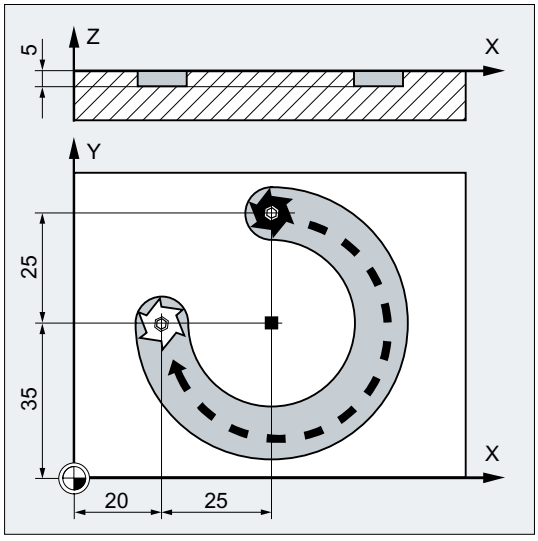
SD42440 $SC_FRAME_OFFSET_INCR_PROG (프레임 내 워크 옵셋)

SD42442 $SC_TOOL_OFFSET_INCR_PROG (공구 길이 옵셋)

값	의미
0	축의 증분치 프로그래밍 (증분 치수) 을 통한 워크 옵셋 또는 공구 길이 옵셋은 이송되지 **않습니다**.
1	축의 증분치 프로그래밍 (증분 치수) 을 통한 워크 옵셋 또는 공구 길이 옵셋이 이송됩니다.

예

예 1: 밀링

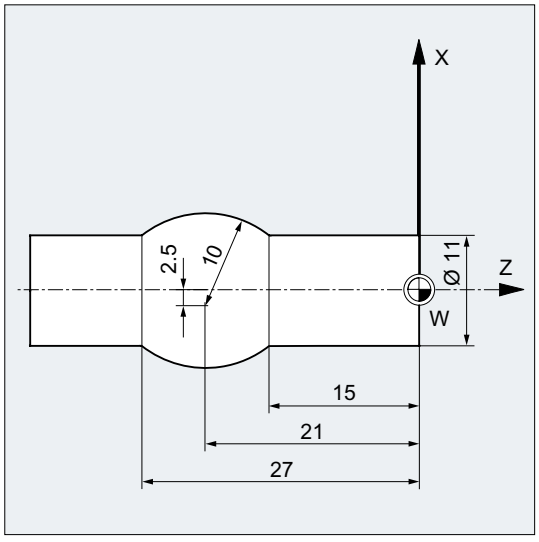


프로그램 코드	코멘트
N10 G90 G0 X45 Y60 Z2 T1 S2000 M3	; 절대 치수 입력, 급 이송으로 XYZ 위치로 이동, 공구 선택, CW 방향 스핀들 회전
N20 G1 Z-5 F500	; 직선 보간, 공구 이송.
N30 G2 X20 Y35 I0 J-25	; CW 방향 원호 보간, 절대치 원호 종점, 증분치 원호 중심점.
N40 G0 Z2	; 이송
N50 M30	; 블록 종료

참고

원호 중심점 I 및 J에 대한 자세한 정보는 "원호 보간" 단락을 참조 바랍니다.

예 2: 선삭



프로그램 코드	코멘트
N5 T1 D1 S2000 M3	; T1 공구 선택, CW 방향 스핀들 회전
N10 G0 G90 X11 Z1	; 절대 치수 입력, 급 이송으로 XZ 위치로 이동
N20 G1 Z-15 F0.2	; 직선 보간, 공구 이송.
N30 G3 X11 Z-27 I-8 K-6	; CCW 방향 원호 보간, 절대치 원호 종점, 증분치 원호 중심점
N40 G1 Z-40	; 이송
N50 M30	; 블록 종료

참고

원호 중심점 I 및 J에 대한 자세한 정보는 "원호 보간" 단락을 참조 바랍니다.

예 3: 활성 워크 옵셋을 이송하지 않는 증분 치수

설정

- G54는 X 축 방향으로 25의 옵셋 값을 포함합니다.
- SD42440 $SC_FRAME_OFFSET_INCR_PROG = 0

프로그램 코드	코멘트
N10 G90 G0 G54 X100	
N20 G1 G91 X10	; 증분 치수 활성화, X를 10 mm 이송 (워크 옵셋은 이송되지 않음).
N30 G90 X50	; 절대 치수 활성화, 위치 X75로 이송 (워크 옵셋만큼 이송).

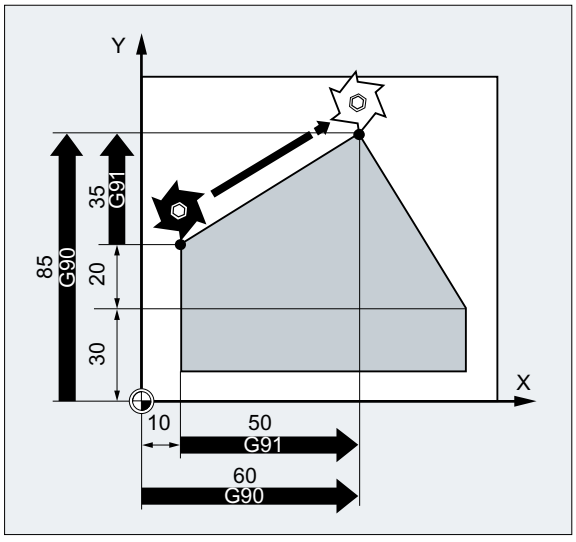
다음 또한 참고:

선삭과 밀링의 절대와 증분 치수(G90/G91) (페이지 173)

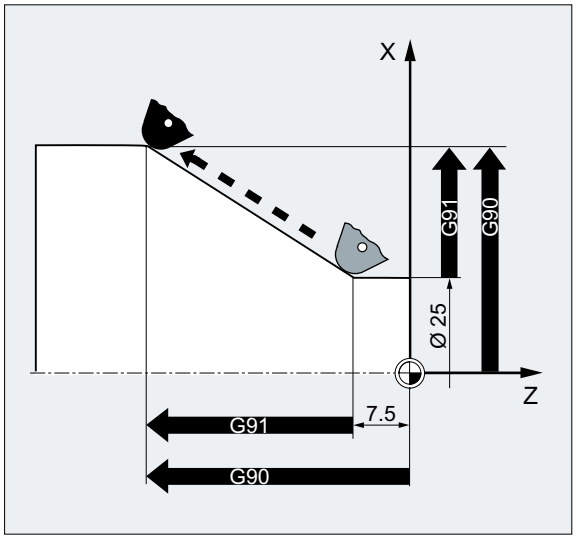
3.8.4.3 선삭과 밀링의 절대와 증분 치수(G90/G91)

다음 두 그림은 선삭과 밀링 기술을 사용한 절대 치수(G90) 또는 증분 치수 (G91)의 프로그래밍의 예를 보여준다.

밀링:



선삭:



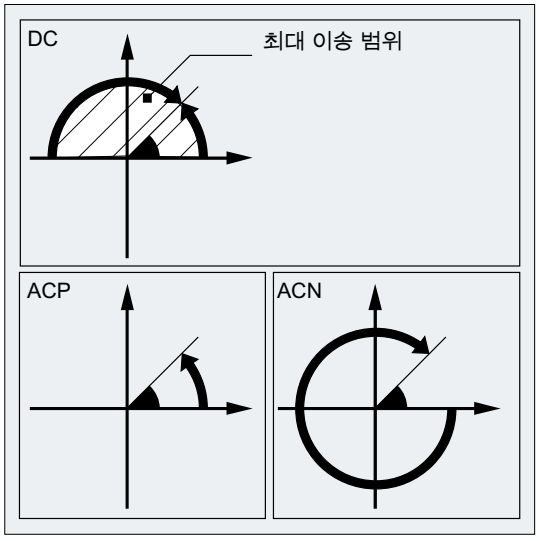
참고

재래식 선삭 기계에서는 직경 치수가 지준 치수에 적용되는 반면 가로 축의 증분 이송 블록을 반경 값으로 계산하는 것이 정상이었습니다. G90의 변환은 명령어 DIAMON, DIAMOF 또는 DIAM90을 사용하여 수행됩니다.

3.8.4.4 로터리 축의 절대 치수 (DC, ACP, ACN)

넌모달 및 G90/G91과는 별도로 DC, ACP 및 ACN 명령을 사용하여 절대 치수로 로터리 축 포지셔닝을 프로그래밍할 수 있습니다.

DC, ACP 및 ACN의 기본 접근 방식이 다릅니다.



구문

```
<Rotary axis>=DC(<Value>)
<Rotary axis>=ACP(<Value>)
<Rotary axis>=ACN(<Value>)
```

의미

<Rotary axis>:	이송될 로터리 축의 이름 (예: A, B 또는 C)
DC:	위치에 **직접** 접근하는 명령 로터리 축은 최단 경로를 통해 프로그래밍된 위치로 이동합니다. 로터리 축의 경우 최대 180° 범위 내에서 이송합니다.

ACP:	**양의** 방향으로 해당 위치로 이동하는 명령 로터리 축이 양의 축 회전 방향 (CCW 방향) 으로 프로그램된 위치까지 이동합니다.	
ACN:	**음의** 방향으로 해당 위치로 이동하는 명령 로터리 축은 음의 회전 방향 (CW 방향) 을 통해 프로그램된 위치로 이동합니다.	
<Value>:	절대 치수로 접근하게 될 로터리 축 위치	
	값 범위:	0 - 360 도

참고

양의 회전 방향 (CW 방향 또는 CCW 방향) 은 머신 데이터를 통해 설정됩니다.

참고

방향이 지정된 포지셔닝을 위해 머신 데이터를 통해 0~360°에서 이송 범위를 지정해야 합니다(ACP, ACN). 모듈로 로터리 축이 360° 이상 이송되게 하려면 블록에서 반드시 G91 또는 IC가 프로그램 되어야 합니다.

참고

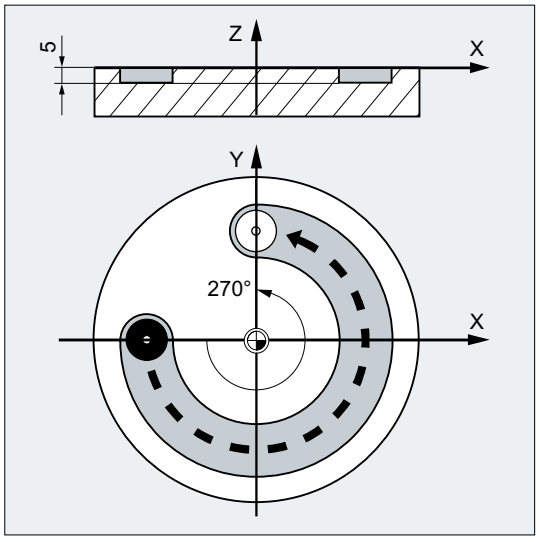
정지 상태에서 스핀들을 포지셔닝 (SPOS, SPOSA) 하고자 할 경우 DC, ACP 및 ACN 명령을 사용할 수 있습니다.

예: SPOS=DC(45)

예

로터리 테이블에서 밀링

공구는 정지 상태이며, 테이블이 **시계 방향**으로 270도까지 회전합니다. 원형 홈/슬롯이 가공됩니다.



프로그램 코드	설명
N10 SPOS=0	; 스핀들 위치 제어 모드.
N20 G90 G0 X-20 Y0 Z2 T1	; 절대 치수, 급 이송으로 공구 T1에서 이송
N30 G1 Z-5 F500	; 이송 속도로 공구 내림
N40 C=ACP(270)	; 원호 포켓을 가공하기 위해 로터리 테이블이 270° 회전.
N50 G0 Z2 M30	; 프로그램 끝까지 후퇴.

3.8.4.5 미터식/인치식 치수 시스템(G70/G71, G700/G710)

가공 프로그램 내 G그룹 13(장치의 인치/미터 시스템)의 명령어를 사용하면 미터 및 인치 단위 시스템을 서로 변경할 수 있습니다.

활성화

명령어 G700 및 G710이 사용 가능하려면 확장 단위 시스템 기능이 켜져 있어야 합니다 (MD10260 $MN_CONVERT_SCALING_SYSTEM = 1).

구문

```
G70
G71
G700
G710
```

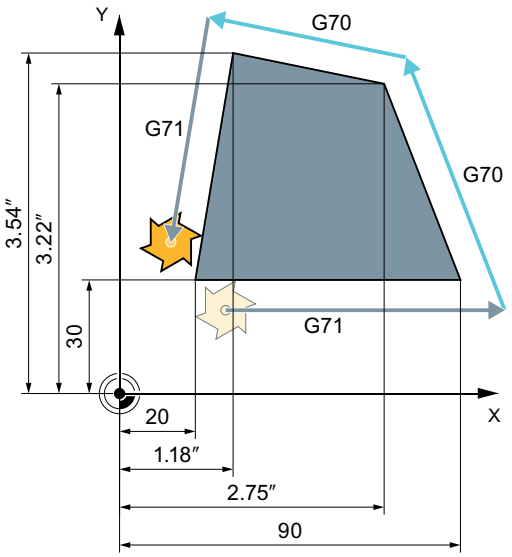
의미

G70:	인치 단위 시스템 활성화 **인치 단위 시스템**은 **기하 데이터를 길이 단위로** 읽고 쓰는 데 사용됩니다. **길이 단위의 기술적 데이터**(예: 이송 속도, 공구 옵셋, 조정 가능한 워크 옵셋, 머신 데이터 및 시스템 변수)는 **파라미터화 된 기본 시스템**을 사용하여 읽히고 쓰여집니다.	
	G 그룹:	13
	초기 설정:	MD20150 $MC_GCODE_RESET_VALUES을 통해 설정 가능
	효율성:	모달
G71:	미터 단위 시스템 활성화 **기하 데이터를 길이 단위**로 읽고 쓰는 데 **미터 단위 시스템**이 사용됩니다. **길이 단위의 기술적 데이터**(예: 이송 속도, 공구 옵셋, 조정 가능한 워크 옵셋, 머신 데이터 및 시스템 변수)는 **파라미터화 된 기본 시스템**을 사용하여 읽히고 쓰여집니다.	
	G 그룹:	13
	초기 설정:	MD20150 $MC_GCODE_RESET_VALUES을 통해 설정 가능
	효율성:	모달
G700:	인치 단위 시스템 활성화 모든 길이 단위의 기하 데이터와 기술적 데이터는 인치 단위 시스템을 사용하여 읽히고 쓰여집니다.	
	G 그룹:	13
	초기 설정:	MD20150 $MC_GCODE_RESET_VALUES을 통해 설정 가능
	효율성:	모달
G710:	미터 단위 시스템 활성화 모든 길이 단위의 기하 데이터와 기술적 데이터는 미터 단위 시스템을 사용하여 읽히고 쓰여집니다.	
	G 그룹:	13
	초기 설정:	MD20150 $MC_GCODE_RESET_VALUES을 통해 설정 가능
	효율성:	모달

유의사항
로터리 축의 축별 데이터 로터리 축의 축별 데이터는 파라미터화 된 기본 시스템을 사용하여 읽히고 쓰여집니다.

예제

기본 시스템은 미터 단위로 되어 있습니다(MD10240 $MN_SCALING_SYSTEM_IS_METRIC = 1). 하지만, 공작물 그림은 인치 단위의 치수로 되어 있습니다. 이것은 가공 프로그램에서 인치 단위 시스템이 선택되었기 때문입니다. 인치 단위 치수가 처리되고 나면 미터 단위 시스템을 다시 선택하도록 합니다.



프로그램 코드	코멘트
N10 G0 G90 X20 Y30 Z2 S2000 M3 T1	; X=20 mm, Y=30 mm, Z=2 mm, F=급 이송 mm/min
N20 G1 Z-5 F500	; Z=-5 mm, F=500 mm/min
N30 X90	; X=90 mm
N40 **G70** X2.75 Y3.22	; 프로그램 된 단위 시스템: 인치
	; X=2.75인치, Y=3.22인치, F=500mm/분
N50 X1.18 Y3.54	; X=1.18 인치, Y=3.54 인치, F=500 mm/min
N60 **G71** X20 Y30	; 프로그램 된 단위 시스템: 미터법
	; X=20mm, Y=30mm, F=500mm/분
N70 G0 Z2	; Z=2mm, F=급 이송 mm/분
N80 M30	; 프로그램 종료

추가 정보

G70/G71 및 G700/G710인 경우의 데이터 읽기 및 쓰기

데이터 영역	G70 / G71		G700 / G710	
	읽기	쓰기	읽기	쓰기
디스플레이, 소수점 이하 자리 (WCS)	P	P	P	P
디스플레이, 소수점 이하 자리 (MCS)	G	G	G	G
이송 속도	G	G	P	P
위치 데이터 X, Y, Z	P	P	P	P
보간 파라미터 I, J, K	P	P	P	P
원 반경(CR)	P	P	P	P
극 반경(RP)	P	P	P	P
나사 피치	P	P	P	P
프로그래밍 프레임	P	P	P	P
설정 가능한 프레임	G	G	P	P
기본 프레임	G	G	P	P
확장 워크 옵셋	G	G	P	P
축 사전 설정 옵셋	G	G	P	P
작업 영역 한계(G25/G26)	G	G	P	P
보호 영역	P	P	P	P
공구 옵셋	G	G	P	P
길이 관련 머신 데이터	G	G	P	P
길이 관련 설정 데이터	G	G	P	P
길이 관련 시스템 변수	G	G	P	P
여러 개의 GUD	G	G	G	G
여러 개의 LUD	G	G	G	G
여러 개의 PUD	G	G	G	G
R 파라미터	G	G	G	G
Siemens 싸이클	P	P	P	P

데이터 영역	G70 / G71		G700 / G710	
	읽기	쓰기	읽기	쓰기
조그/핸드휠 증분 팩터	G	G	G	G
P: 쓰기/읽기는 프로그래밍 된 단위 시스템에서 실시됩니다. G: 읽기/쓰기는 구성된 기본 시스템 내에서 실시됩니다.				

동기 동작

참고

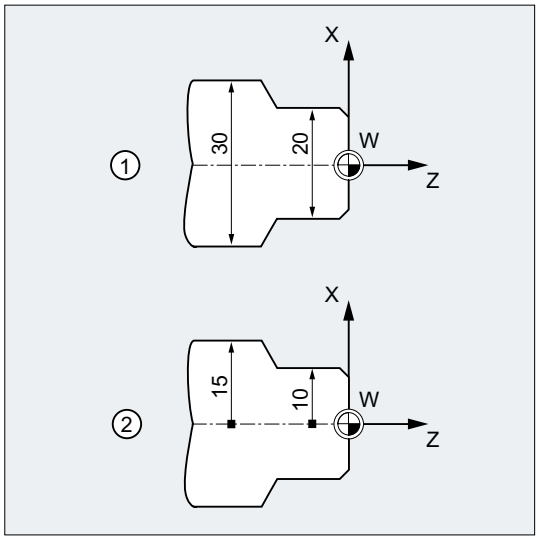
동기 동작에서의 위치 데이터 읽기

단위 시스템이 동기 동작에서 명확하게 프로그래밍 되지 않은 경우(조건 컴포넌트 및/또는 동작 컴포넌트), 동기 동작에서의 **길이 관련 위치 데이터** 는 항상 파라미터화 된 **기본 시스템** 에서 읽혀지게 됩니다.

추가 정보: 기능 매뉴얼, 동기 동작

3.8.4.6 채널 관련 직경/반경 (DIAMON, DIAM90, DIAMOF, DIAMCYCOF) 프로그래밍

선삭 시 **가로 축**의 치수는 다음과 같이 직경 (①) 또는 반경 (②) 으로 지정할 수 있습니다.



채널 관련 직경 또는 반경 프로그래밍을 모달 명령어 DIAMON, DIAM90, DIAMOF 및 DIAMCYCOF로 활성화하여 도면상의 데이터를 직접 (변환 없이) NC 프로그램에 전송할 수 있습니다.

참고

채널 관련 직경/반경 프로그래밍은 MD20100 $MC_DIAMETER_AX_DEF을 통해 가로 축으로 정의된 기하 축을 기준으로 합니다. (장비 제조 업체의 사양을 참조 바랍니다.)
MD20100을 통해 채널 당 하나의 가로축을 정의할 수 있습니다.

구문

```
DIAMON
DIAM90
DIAMOF
```

의미

DIAMON:	**독립된** 채널 관련 직경 프로그래밍을 활성화하는 명령어. DIAMON의 효과는 프로그래밍된 치수 모드 (절대 치수 G90 또는 증분 치수 G91) 의 영향을 받지 않습니다.	
	• G90의 경우:	직경치
	• G91의 경우:	직경치
DIAM90:	**종속적인** 채널 관련 직경 프로그래밍을 활성화하는 명령어. DIAM90의 효과는 프로그래밍된 치수 모드에 따라 다릅니다.	
	• G90의 경우:	직경치
	• G91의 경우:	반경치

DIAMOF:	채널 관련 직경 프로그래밍을 해제하는 명령어 채널 관련 반경 프로그래밍은 직경 프로그래밍을 비활성화한 경우에 적용됩니다. DIAMOF의 효과는 프로그래밍된 치수 모드의 영향을 받지 않습니다.	
	• G90의 경우:	반경치
	• G91의 경우:	반경치
DIAMCYCOF:	사이클 프로세스가 진행되는 동안 채널 관련 직경 프로그래밍 해제 명령어 이 방식은 항상 반경을 기준으로 사이클을 계산합니다. 이 그룹에서 활성화된 마지막 G 명령은 위치 표시기 및 기본 블록 표시기를 위해 활성화된 상태를 유지합니다.	

참고

DIAMON 또는 DIAM90을 통해 가로 축의 실제 값은 항상 직경으로 디스플레이됩니다. 또한 MEAS, MEAW, $P_EP[x] 와 $AA_IW[x]를 사용하는 공작물 좌표계에서 실제값을 판독할 때도 직경이 사용됩니다.

예

프로그램 코드	코멘트
N10 G0 X0 Z0	; 시작점으로 접근.
N20 DIAMOF	; 직경 프로그래밍 OFF.
N30 G1 X30 S2000 M03 F0.7	; X축 = 가로 축; 반경 프로그래밍 활성화: 반경 위치 X30으로 이송.
N40 DIAMON	; 가로축에 대해 직경 프로그래밍이 적용됩니다.
N50 G1 X70 Z-20	; 직경 위치 X70 및 Z-20으로 이송.
N60 Z-30	
N70 DIAM90	; 절대 치수 직경 프로그래밍 및 증분 치수 반경 프로그래밍.
N80 G91 X10 Z-20	; 증분 치수 활성화.
N90 G90 X10	; 절대 치수 활성화.
N100 M30	; 프로그램 종료

추가 정보

직경 값 (DIAMON/DIAM90)

직경치는 다음 데이터에 적용됩니다.

- 공작물 좌표계의 가로 축에 표시되는 실제 값
- JOG 모드: 증분 치수 및 수동 핸드휠 이동을 위한 증분
- 종점 위치 프로그래밍:
  AC 의 절대 치수를 사용하여 프로그래밍된경우의 G2/G3의 보간 파라미터 I, J, K,
  I, J, K가 증분 치수 (IC) 로 프로그래밍되었다면 항상 반경이 계산됩니다.
- 공작물 좌표계로 다음 항목에 대한 실제값 읽기
  MEAS, MEAW, $P_EP[X], $AA_IW[X]

3.8.4.7 축 관련 직경/반경 프로그래밍 (DIAMONA, DIAM90A, DIAMOFA, DIACYCOFA, DIAMCHANA, DIAMCHAN, DAC, DIC, RAC, RIC)

채널 관련 직경 프로그래밍에 추가한 축 관련 직경 프로그래밍 기능 때문에 모달 또는 넌모달 치수와 하나 또는 그 이상의 축에 대한 직경으로의 디스플레이가 가능해집니다.

참고

MD30460 $MA_BASE_FUNCTION_MASK을 통한 축 관련 직경 프로그래밍에서 추가적으로 축으로 허용된 축에만 축 관련 직경 프로그래밍이 가능합니다. (장비 제조업체 사양을 참조 바랍니다.)

구문

채널 내 다수의 가로 축에 대한 모달 축 관련 직경 프로그래밍:

```
DIAMONA[<axis>]
DIAM90A[<axis>]
DIAMOFA[<axis>]
DIACYCOFA[<axis>]
```

채널 관련 직경/반경 프로그래밍의 적용:

```
DIAMCHANA[<axis>]
DIAMCHAN
```

넌모달 축 관련 직경/반경 프로그래밍

```
<axis>=DAC(<value>)
<axis>=DIC(<value>)
<axis>=RAC(<value>)
<axis>=RIC(<value>)
```

의미

모달 축 관련 직경 프로그래밍		
DIAMONA:	**독립된** 축 관련 직경 프로그래밍을 활성화하는 명령어 DIAMONA의 효과는 프로그래밍된 치수 모드 (G90/G91 또는 AC/IC) 와는 무관합니다.	
	• G90, AC의 경우:	직경치
	• G91, IC의 경우:	직경치
DIAM90A:	**종속된** 축 관련 직경 프로그래밍을 활성화하는 명령어 DIAM90A의 효과는 프로그래밍된 치수 모드에 따라 다릅니다.	
	• G90, AC의 경우:	직경치
	• G91, IC의 경우:	반경치
DIAMOFA:	축 관련 직경 프로그래밍을 비활성화하는 명령어 축 관련 반경 프로그래밍은 직경 프로그래밍을 해제한 경우에 적용됩니다. DIAMOFA의 효과는 프로그래밍된 치수 모드와는 무관합니다.	
	• G90, AC의 경우:	반경치
	• G91, IC의 경우:	반경치
DIACYCOFA:	사이클 프로세스가 진행되는 동안 축 관련 직경 프로그래밍을 해제하는 명령어 이 방식은 항상 반경을 기준으로 사이클을 계산합니다. 이 그룹에서 활성화된 마지막 G 명령은 위치 표시기 및 기본 블록 표시기를 위해 활성화된 상태를 유지합니다.	
<axis>:	축 관련 직경 프로그래밍이 적용될 축 이름. 식별자로 허용된 축은 다음과 같습니다: • 기하/채널 축 이름 또는 • 기계 축 이름	
	값 범위:	지정된 축은 반드시 채널에 알려져야 합니다. 그 외의 조건: • 축은 반드시 MD30460 $MA_BASE_FUNCTION_MASK를 통하여 축 관련 직경 프로그래밍에 허용되어야 합니다. • 로터리 축은 가로 축으로 허용되지 않습니다.
채널 관련 직경/반경 프로그래밍의 수용		

DIAMCHANA:	**지정된 축**은 DIAMCHANA[<axis>] 명령어를 사용하여 직경/반경 프로그래밍의 채널 상태를 수용하고 채널 관련 직경/반경 프로그래밍으로 지정됩니다.
DIAMCHAN:	축 관련 직경 프로그래밍하도록 허용된 **모든**축은DIAMCHAN 명령어를 사용하여 직경/반경 프로그래밍의 채널 상태를 수용하고 채널 관련 직경/반경 프로그래밍으로 지정됩니다.
넌모달 축 관련 직경/반경 프로그래밍 가공 프로그램 및 동기 동작에서, 넌모달 축 지정 직경/반경 프로그램은 치수 형식을 직경 또는 반경 값으로 지정합니다. 직경/반경 프로그래밍의 모달 상태는 변하지 않고 남아 있게 됩니다.	
DAC:	DAC 명령 이후 지정된 축의 치수를 넌모달로 지정합니다: 절대치 직경
DIC:	DIC 명령 이후 지정된 축의 치수를 넌모달로 지정합니다: 증분치 직경
RAC:	RAC 명령 이후 지정된 축의 치수를 넌모달로 지정합니다: 절대치 반경
RIC:	RIC 명령 이후 지정된 축의 치수를 넌모달로 지정합니다: 증분치 반경

참고

DIAMONA[<axis>] 또는 DIAM90A[<axis>]를 통해 가로 축 실제 값은 항상 직경으로 디스플레이됩니다. 또한 MEAS, MEAW, $P_EP[x] 와 $AA_IW[x]를 사용하는 공작물 좌표계에서 실제값을 판독할 때도 직경이 사용됩니다.

참고

추가 가로 축 교체 동안 GET 요청에 의하여 다른 채널의 직경/반경/ 프로그래밍 상태가 RELEASE[<axis>]와 함께 수용됩니다.

예

예 1: 모달 축 관련 직경/반경 프로그래밍

X는 채널의 가로 축이며 축 관련 직경 프로그래밍은 Y에 허용됩니다.

프로그램 코드	코멘트
N10 G0 X0 Z0 DIAMON	; X 축에 대해 채널 지정 직경 프로그래밍 적용.

프로그램 코드	코멘트
N15 DIAMOF	; 채널 지정 직경 프로그래밍 OFF.
N20 DIAMONA[Y]	; Y 축에 대해 모달 축 지정 직경 프로그래밍 동작.
N25 X200 Y100	; X 축에 대해 반경 프로그래밍 적용.
N30 DIAMCHANA[Y]	; Y 축은 채널 지정 직경/반경 프로그래밍 상태를 수용하고 이것에 지정됩니다.
N35 X50 Y100	; X 및 Y 축에 대해 반경 프로그래밍 적용.
N40 DIAMON	; 채널 지정 직경 프로그래밍 ON.
N45 X50 Y100	; X 및 Y 축에 대해 직경 프로그래밍 적용.

예 2: 넌모달 축 관련 직경/반경 프로그래밍

X는 채널의 가로 축이며 축 관련 직경 프로그래밍은 Y에 허용됩니다.

프로그램 코드	코멘트
N10 DIAMON	; 채널 지정 직경 프로그래밍 ON.
N15 G0 G90 X20 Y40 DIAMONA[Y]	; Y 축에 대해 모달 축 지정 직경 프로그래밍 동작.
N20 G01 X=RIC(5)	; 이 블록에서 X 축에 적용되는 증분치 반경 치수.
N25 X=RAC(80)	; 이 블록에서 X 축에 적용되는 절대치 반경 치수.
N30 WHEN $SAA_IM[Y]> 50 DO POS[X]=RIC(1)	; X 축은 지령축입니다. 이 블록에서 X 축에 적용되는 증분치 반경 치수.
N40 WHEN $SAA_IM[Y]> 60 DO POS[X]=DAC(10)	; X 축은 지령축입니다. 이 블록에서 X 축에 적용되는 절대치 반경 치수.
N50 G4 F3	

추가 정보

직경 값 (DIAMONA/DIAM90A)

직경치는 다음 데이터에 적용됩니다.

- 공작물 좌표계의 가로 축에 표시되는 실제 값
- JOG 모드: 증분 치수 및 수동 핸드휠 이동을 위한 증분
- 종점 위치 프로그래밍:
  AC 의 절대 치수를 사용하여 프로그래밍된경우의 G2/G3 의 보간 파라미터 I, J, K, I, J, K 가 증분 치수 (IC) 로 프로그래밍되었다면 항상 반경이 계산됩니다.
- 공작물 좌표계로 다음 항목에 대한 실제값 읽기
  MEAS, MEAW, $P_EP[X], $AA_IW[X]

넌모달 축 관련 직경 프로그래밍 (DAC, DIC, RAC, RIC)

DAC, DIC, RAC, RIC 명령은 채널 지정 직경 프로그래밍 명령에 사용할 수 있습니다.

- 축 위치: X..., POS, POSA
- 왕복: OSP1, OSP2, OSS, OSE, POSP
- 보간 파라미터: I, J, K
- 형상 정의: 각도가 지정된 직선
- 고속 후퇴: POLF[AX]
- 공구 방향으로의 이송: MOVT
- 부드러운 접근 및 후퇴:
  G140 에서 G143, G147, G148, G247, G248, G347, G348, G340, G341까지

3.8.5 선삭용 공작물 위치

축 식별자

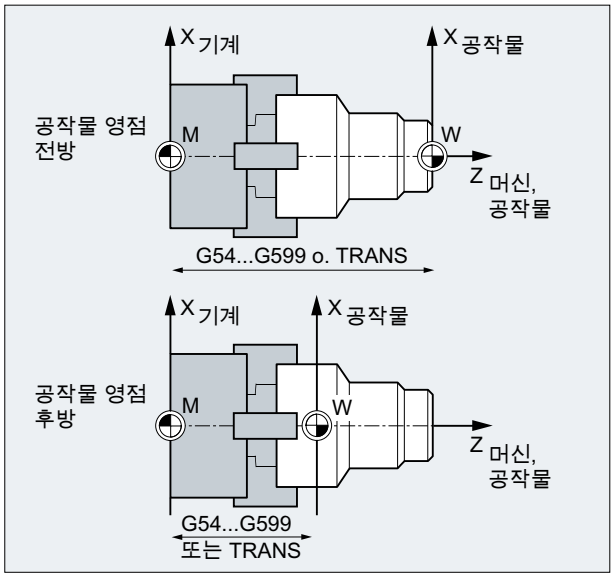
일반적으로 서로 직각인 두 도형 축이 호출됩니다:

세로 축	= Z 축 (가로 좌표)
가로 축	= X 축 (세로 좌표)

공작물 영점

기계 영점은 영구히 정의되는 반면 공작물 영점은 세로 축에서 임의로 선택될 수 있습니다. 일반적으로 공작물 영점은 공작물의 전면 또는 후면에 위치합니다.

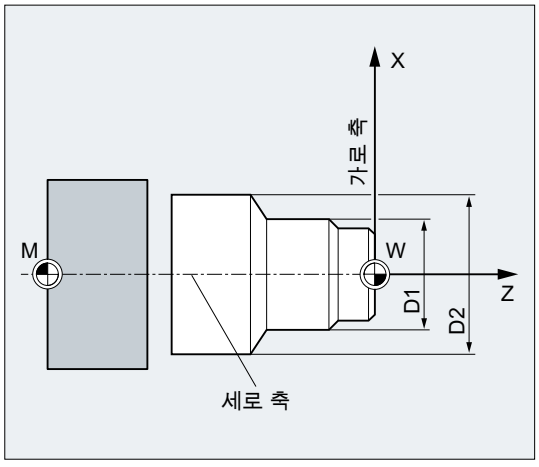
기계와 공작물의 두 영점은 선삭 중심에 위치합니다. 그러므로 X 축의 설정 가능한 옵셋은 영점이 됩니다.



M	기계 영점
W	공작물 영점
Z	세로 축
X	가로 축
G54 에서 G599 까지 또는 TRANS	공작물 영점 호출

가로 축

일반적으로 가로 축 치수는 직경 사양입니다(다른 축과 비교되는 이중 경로 치수):



가로 축이 되는 도형 축은 기계 데이터에서 정의됩니다(→기계 제조업체)

3.9 이송 명령

3.9.1 이동 명령어의 일반 정보

형상 요소

프로그램된 공작물 형상은 다음의 형상 요소로 만들어 질 수 있습니다.

- 직선
- 원호,
- 나선형 곡선(직선과 원호의 겹침을 통하여)

이동 명령어

형상 요소 생성에 다음의 이동 명령어가 유효합니다.

- 급이송 동작(G0)
- 선형 보간(G1)
- 시계 방향 원호 보간(G2)
- CCW 방향 원호 보간 (G3)

이동 명령은 모달식입니다.

목표 위치

동작 블록에는 이송될 축의 목표 위치가 포함되어 있습니다(경로 축, 동기화된 축, 위치 지정 축).

목표 위치는 직교 좌표 또는 극 좌표로 프로그램될 수 있습니다.

참고

축 주소는 블록 당 하나만 프로그램될 수 있습니다.

시작점 - 목표점

이송 동작은 항상 프로그램된 목표 위치에 도달한 최종 지점까지 입니다. 목표 위치는 이때 다음 이동 명령어의 시작점이 됩니다.

공작물 형상

유의사항
공구 운전 미정의 기계 가공 전 공작물은 반드시 그렇게 위치되어야 공구 또는 공작물이 손상되지 않습니다.

동작 블록이 연속으로 수행될 때 공작물이 생성됩니다:

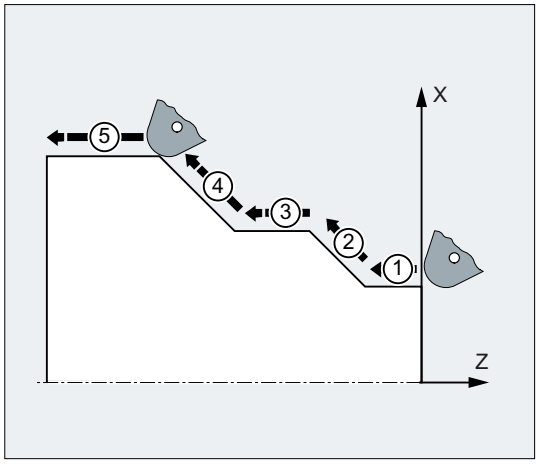


그림 3-1 선삭의 동작 블록

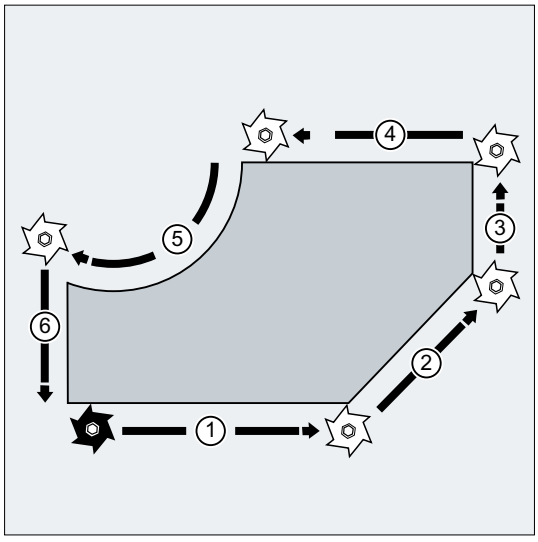


그림 3-2 밀링의 동작 블록

3.9.2 직교 좌표의 이동 명령어(G0, G1, G2, G3, X..., Y..., Z...)

직교 좌표를 사용하여 NC 블록에서 지정된 위치로 이동할 경우, 급 이송 G0, 직선 보간 G1 또는 원호 보간 G2/G3를 사용하여 이동할 수 있습니다.

구문

```
G0 X... Y... Z...
G1 X... Y... Z...
G2 X... Y... Z... ...
G3 X... Y... Z... ...
```

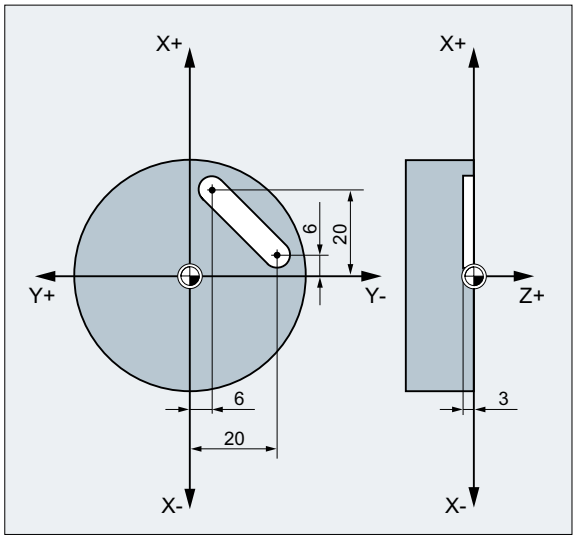
의미

G0:	급 이송 모션 활성화 명령
G1:	직선 보간 명령
G2:	CW 방향 원호 보간 명령
G3:	CCW 방향 원호 보간 명령
X...:	X 방향 대상 위치의 직교 좌표
Y...:	Y 방향 대상 위치의 직교 좌표
Z...:	Z 방향 대상 위치의 직교 좌표

참고

대상 위치 좌표 X..., Y..., Z...외에 원호 보간G2 / G3 역시 추가 데이터가 필요합니다 (예: 원호 중심점 좌표. "개요 (페이지 209)" 참조).

예



프로그램 코드	코멘트
N10 G17 S400 M3	; 작업 평면의 선택, CW 방향 스핀들 회전
N20 G0 X40 Y-6 Z2	; 급 이송으로 직교 좌표로 지정된 시작 위치로 이동
N30 G1 Z-3 F40	; 직선 보간 동작, 공구 절입
N40 X12 Y-20	; 직교 좌표로 지정된 종료 위치로 이동
N50 G0 Z100 M30	; 공구 교환을 위해 급 이송으로 후퇴

3.9.3 극 좌표의 이송 명령어

3.9.3.1 극 좌표의 기준점(G110, G111, G112)

치수가 시작되는 지점을 극점이라고 합니다.

극점은 직교 좌표 또는 극 좌표로 지정될 수 있습니다.

극 좌표의 원은 G110부터 G112까지의 명령어로 명확하게 정의됩니다. 절대 치수 또는 증분 치수는 아무런 영향을 미치지 않습니다.

구문

```
G110/G111/G112 X… Y… Z…
G110/G111/G112 AP=… RP=…
```

의미

G110 ...:	G110명령을 사용한 경우 후속 극 좌표는 **이전에 도달한 위치**를 기준으로 합니다.		
G111 ...:	G111명령을 사용한 경우 후속 극 좌표는 **현재 공작물 좌표계의 영점**을 기준으로 합니다.		
G112 ...:	G112명령을 사용한 경우 후속 극 좌표는 **최종적으로 사용된 극점**을 기준으로 합니다.		
	참고: 명령어 G110...G112는 반드시 별도의 NC 블록에서 프로그래밍되어야 합니다.		
X… Y… Z…:	직교 좌표에서 극점 지정		
AP=… RP=…:	극 좌표에서 극점 지정		
	AP=…:	극점 각도 극점 반경 및 작업 평면 상에서 수평축 (G17의 경우 X 축) 간의 각도. 양의 회전 방향은 CCW 방향으로 회전합니다.	
		값 범위:	± 0...360°
	RP=…:	극점 반경 [mm] 또는 [inch] 단위로 **항상 양의 절대치**로 지정합니다.	

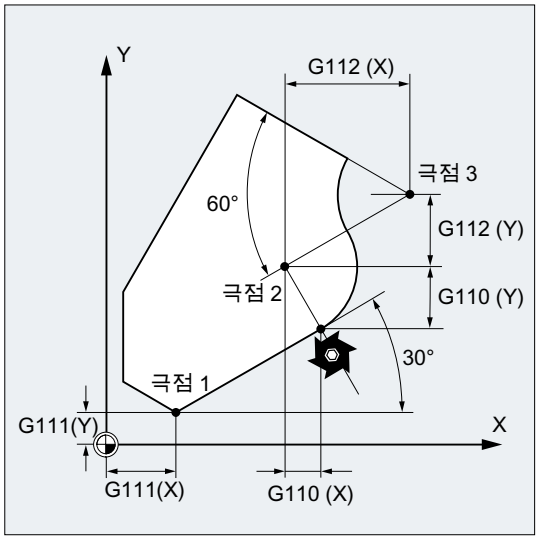
참고

NC 프로그램에서 극 좌표와 직교 좌표 치수 사이를 블록 별로 전환하는 것이 가능합니다. 직교 좌표 식별자 (X..., Y..., Z...) 를 사용하여 직교 좌표로 직접 복귀하는 것도 가능합니다. 게다가 정의된 극점은 프로그램 종료까지 유지됩니다.

참고

지정된 극점이 없다면 현 공작물 좌표계의 영점이 적용됩니다.

예

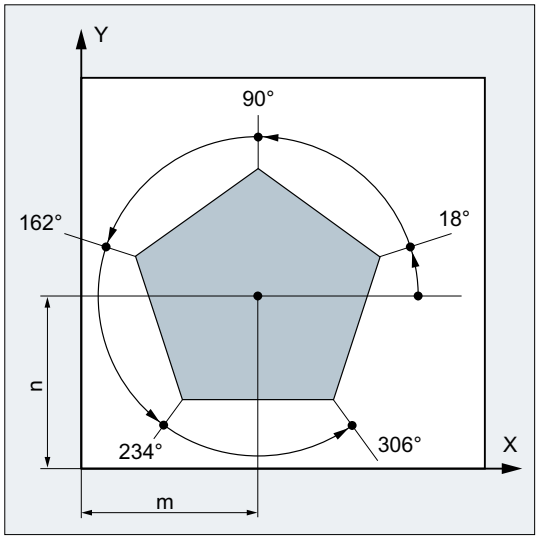


극점 1, 2, 3이 다음과 같이 지정됩니다.

- `G111 X… Y…`의 극점 1
- `G110 X… Y…`의 극점 2
- `G112 X… Y…`의 극점 3

3.9.3.2 극 좌표로의 이동 명령어(G0, G1, G2, G3, AP, RP)

극 좌표 이송 명령은 치수 또는 공작물의 일부가 중심점에서 측정되며 치수가 각도와 반경으로 지정되었을 때 유용합니다.



구문

```
G0/G1/G2/G3 AP=… RP=…
```

의미

G0:	급 이송 모션 활성화 명령	
G1:	직선 보간 명령	
G2:	CW 방향 원호 보간 명령	
G3:	CCW 방향 원호 보간 명령	
AP:	극점 각도 극점 반경 및 작업 평면 상에서 수평축 (G17의 경우 X 축) 간의 각도. 양의 회전 방향은 CCW 방향으로 회전합니다.	
	값 범위:	± 0...360°
	각도는 증분이나 절대 치수 중 하나로 지정될 수 있습니다.	
	AP=AC(...):	절대 치수 입력
	AP=IC(...):	증분치 입력 증분치 입력으로 최종 프로그래밍된 각도가 기준으로 적용됩니다.
	극점 각도는 새 극점이 정의되거나 작업 평면이 변경될 때까지 계속 유지됩니다.	
RP:	극점 반경 [mm] 또는 [inch] 단위로 **항상 양의 절대치**로 지정합니다. 극점 반경은 새 값이 입력될 때까지 유지됩니다.	

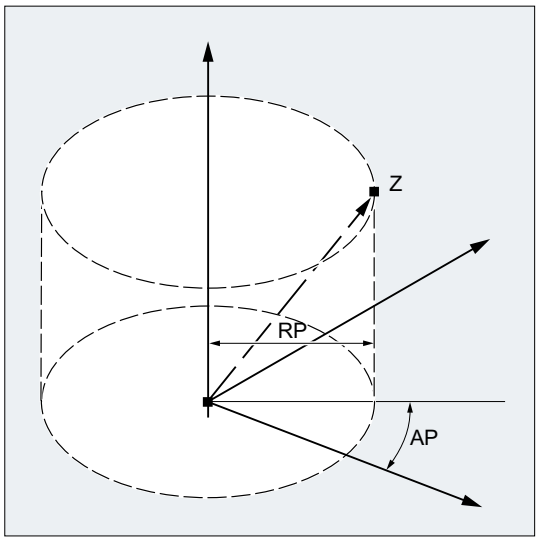
참고

극 좌표는 G110 ... G112에 의해 지정된 극점을 기준하고 G17에서 G19까지에 의해 선택된 작업 평면에 적용됩니다.

참고

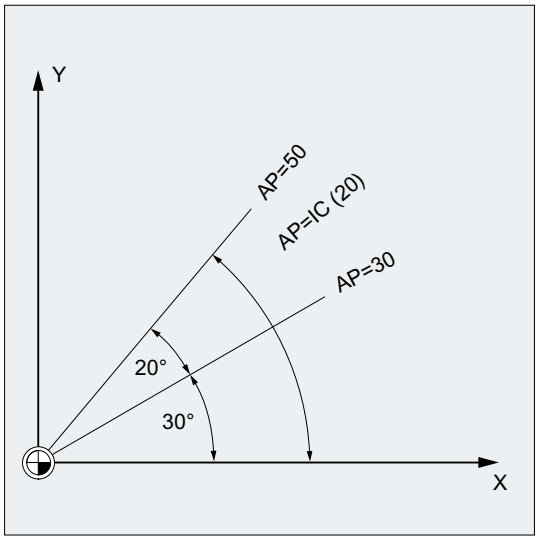
작업 평면과 수직을 이루고 있는 세 번째 기하 축은 직교 좌표에도 지정할 수 있습니다 (아래 다이어그램 참조). 이로써 원통형 좌표에 프로그래밍될 공간 파라미터가 가능해집니다.

예: G17 G0 AP… RP… Z…



보충 조건

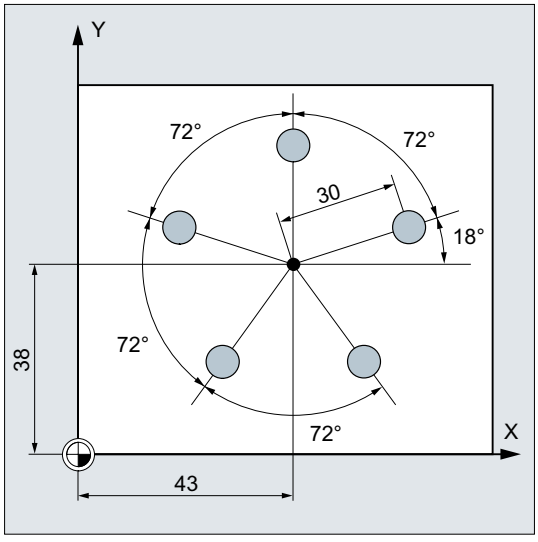
- 보간 파라미터, 축 어드레스 등과 같이 직교 좌표가 아닌 것을 NC블록에서 극 종점 좌표를 사용하여 선택된 작업 평면에 프로그래밍할 수 있습니다.
- 극점이 G110 ... G112로 정의되지 않았다면 현 공작물 좌표계의 영점은 자동으로 극점을 계산합니다.



- 극점 반경 RP=0
  극점 반경은 극점 평면에서 시작점 벡터 및 사용중인 극점 벡터 간의 사이의 거리로부터 계산됩니다. 계산된 극점 반경은 이때 모달로 저장됩니다.
  이는 선택된 극점 정의 (G110 ... G112) 와는 무관하게 적용됩니다. 지점이 동일하게 프로그래밍 된 경우 반경은 0 이 되며 알람 14095가 발생됩니다.
- 단지 극점 각도 AP가 프로그래밍되게 됩니다.
  극점 반경 RP가 현 블록에 프로그래밍되지 않고 극점 각도 AP만 프로그래밍되었다면 현재 위치와 공작물 좌표의 극점에 차이가 있을 때 이 차이는 극점 반경으로 사용되고 모달로 저장됩니다. 그 차이가 0 이라면 극점 좌표는 다시 지정되고 모달 극점 반경은 0로 남아 있게 됩니다.

예

드릴링 패턴 생성



홀들의 위치는 극 좌표로 지정됩니다.
각 홀들은 동일한 순서로 가공됩니다.
황삭 드릴링, 드릴링, 리밍...
가공 순서는 서브프로그램에 저장됩니다.

프로그램 코드	코멘트
N10 G17 G54	; 작업 평면 X/Y, 공작물 원점
N20 G111 X43 Y38	; 극점 지정.
N30 G0 RP=30 AP=18 Z5	; 시작점 접근, 원통형 좌표에 지정.
N40 L10	; 서브프로그램 호출.
N50 G91 AP=72	; 급 이송으로 다음 위치 접근, 증분 치수로의 극점 각도, 블록 N30에서의 극점 반경은 저장되어 남게되고 지정할 필요가 없습니다.
N60 L10	; 서브프로그램 호출.
N70 AP=IC(72)	.

프로그램 코드	코멘트
N80 L10	…
N90 AP=IC(72)	
N100 L10	…
N110 AP=IC(72)	
N120 L10	…
N130 G0 X300 Y200 Z100 M30	; 공구 후퇴, 프로그램 종료.

다음 또한 참고:

개요 (페이지 209)

3.9.4 급 이송 운동

3.9.4.1 급 이송 (G0) 활성화

급 이송 속도의 경로 축 이송이 G 명령 G0로 활성화됩니다.

구문

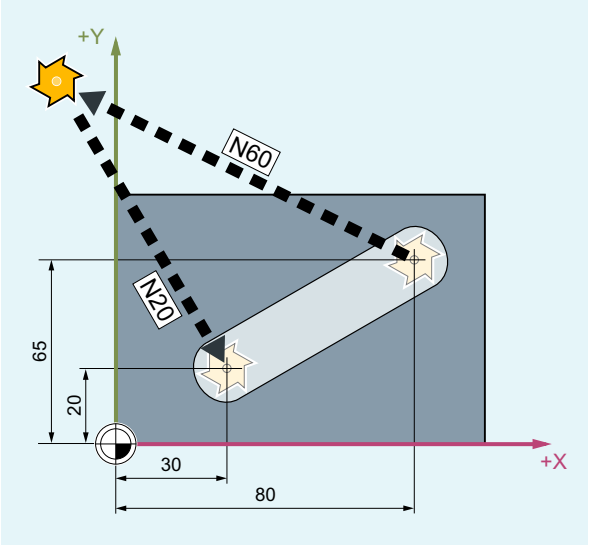
```
G0 X… Y… Z…
G0 RP=… AP=…
```

의미

G0:	축을 급 이송 속도로 이송	
	적용:	모달
X... Y... Z...:	직교 좌표 내 종단점을 표시	
RP=... AP=... :	극 좌표 내 종단점을 표시	

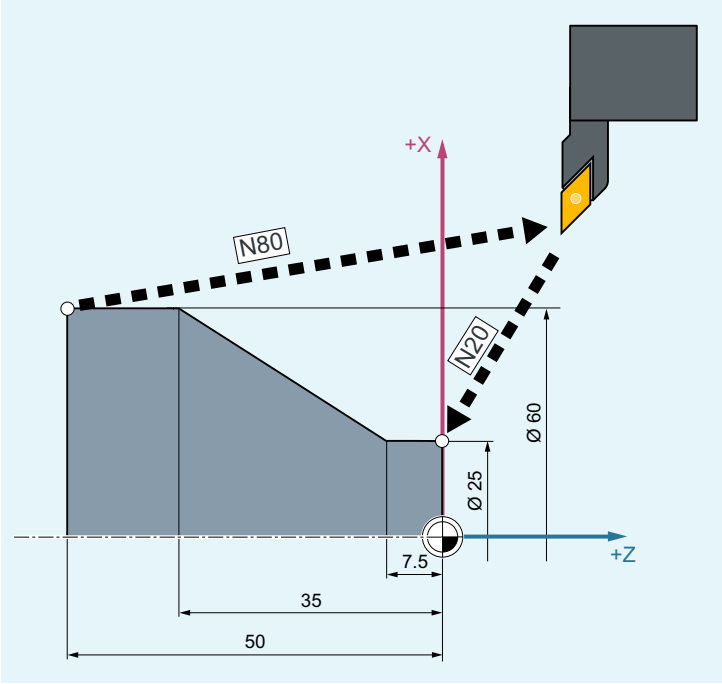
예

예 1: 밀링



프로그램 코드	코멘트
N10 G90 S400 M3	; 절대 치수 입력, CW 방향 스핀들 회전
N20 G0 X30 Y20 Z2	; 시작 위치에 접근
N30 G1 Z-5 F1000	; 공구 절입
N40 X80 Y65	; 직선을 따라 이송
N50 G0 Z2	
N60 G0 X-20 Y100 Z100 M30	; 프로그램 끝까지 공구 후퇴

예 2: 선삭



프로그램 코드	코멘트
N10 G90 S400 M3	; 절대 치수 입력, CW 방향 스핀들 회전
N20 G0 X25 Z5	; 시작 위치에 접근
N30 G1 G94 Z0 F1000	; 공구 절입
N40 G95 Z-7.5 F0.2	
N50 X60 Z-35	; 직선을 따라 이송
N60 Z-50	
N70 G0 X62	
N80 G0 X80 Z20 M30	; 프로그램 끝까지 공구 후퇴

3.9.4.2 급 이송 동작을 위한 선형 보간 켜기/끄기(RTLION, RTLIOF)

기본 설정(MD20730 $MC_G0_LINEAR_MODE)과 별개로, 급 이송 보간 반응은 G 그룹 55 명령어들을 이용해 가공 프로그램에서 설정할 수도 있습니다.

구문

```
RTLIOF
...
RTLION
```

의미

RTLIOF:	선형 보간 끄기 G 명령 ⇒ 급 이송 모드(G0)에서 **비 선형** 보간이 활성화됩니다. 모든 경로 축이 상호 독립적으로 종점에 도달합니다.	
	적용:	모달
RTLION:	선형 보간 켜기 G 명령 ⇒ 급 이송 모드(G0)에서 **선형** 보간이 활성화됩니다. 모든 경로 축이 동시에 종점에 도달합니다.	
	적용:	모달

참고

RTLIOF 사전 조건

RTLIOF **비 선형** 보간의 경우 다음 조건이 충족되어야 합니다.

- 활성인 변환 없음(TRAORI, TRANSMIT, etc.).
- G60 활성(블록 끝에서 정지).
- 활성인 컴프레서 없음(COMPOF).
- 활성인 공구 반경 보정 없음(G40).
- 선택된 형상 핸드휠 없음.
- 활성인 니블링 없음.

이 중 어느 한 조건이라도 충족되지 않으면 선형 보간이 RTLION으로 하는 것과 같습니다.

예제

프로그램 코드	**코멘트**
	; 선형 보간이 기본:
	; MD20730 $MC_GO_LINEAR_MODE == TRUE
...	
N30 **RTLIOF**	; 선형 보간 끔.
N40 G0 X0 Y10	; G0 블록이 비 선형 보간을 이용해 이송됩니다.
N50 G41 X20 Y20	; TRC 활성 ⇒ G0 블록이 선형 보간을 이용해 이송됩니다.
N60 G40 X30 Y30	; TRC 활성 안 됨 ⇒ G0 블록이 비 선형 보간을 이용해 이송됩니다.
N70 **RTLION**	; 선형 보간 켬.
...	

추가 정보

현재 보간 동작 읽기

현재 보간 동작은 $AA_G0MODE 시스템 변수들을 통해 읽을 수 있습니다.

3.9.4.3 급이송 모션을 위한 허용 공차 조정(STOLF, CTOLG0, OTOLG0)

머신 데이터를 사용하여 구성된 급이송 모션(G0 공차) 공차는 가공 프로그램에서 임시로 조정할 수 있습니다. 이렇게 하면 머신 데이터의 설정이 변경되지 않습니다. 채널 또는 프로그램 종료가 리셋되면 구성된 공차가 다시 적용됩니다.

요구 사항

G0 공차는 다음 조건을 준수하는 경우에만 유효합니다.

- 다음 함수 중 하나가 활성인 경우
  - 컴프레서 기능 COMP...
  - 스무딩 기능 G642 또는 G645
  - 오리엔테이션 스무딩 OST
  - 오리엔테이션 스무딩 ORISON
  - 경로 관련 오리엔테이션 스무딩 ORIPATH
- 가공 프로그램에서 연속 G0 블록이 2개 이상인 경우
  G0 블록이 1개인 경우 G0 공차가 적용되지 않습니다. 이는 비G0 모션에서 G0 모션으로, 또는 그 반대로 트렌지션할 때 항상 '**더 낮은 공차**', 즉 공작물 가공 공차가 적용되기 때문입니다.

구문

상대 G0 공차의 조정

```
STOLF=<Value>
```

절대 G0 공차의 조정

```
CTOLG0=<Value>
```

```
OTOLG0=<Value>
```

의미

STOLF:	급이송 모션을 위해 임시로 유효 공차 팩터를 프로그래밍하는 주소			
	<Value>:	G0 공차 팩터		
		유형:	REAL	
		값:	≥ 0:	공차 팩터 G0은 1.0보다 크거나 작을 수 있습니다. 팩터가 1.0(기본 설정값)이면 비급이송 모션과 같은 공차가 급이송 모션에 적용됩니다. 보통 공차 팩터는 1.0보다 크게 설정됩니다. 프로그래밍된 G0 공차 팩터는 갱신된 STOLF 프로그래밍으로 덮어쓰기 되거나, CTOLG0/OTOLG0 프로그래밍으로 대체되거나, 채널 또는 프로그램 재설정 종료로 삭제될 때까지 유효합니다.
			<0:	프로그래밍된 공차 팩터가 삭제됩니다. ⇒ 머신 데이터에서 사전 설정된 공차 값이 다시 적용됩니다.
CTOLG0:	급이송 모션을 위해 임시로 유효 형상 공차 팩터를 프로그래밍하는 주소			
	<Value>:	형상 공차의 절대치		
		유형:	REAL	
		값:	≥ 0:	형상 공차의 프로그래밍된 절대치는 갱신된 CTOLG0 프로그래밍으로 덮어쓰기 되거나, STOLF 프로그래밍으로 대체되거나, 채널 또는 프로그램 재설정 종료로 삭제될 때까지 유효합니다.
			<0:	프로그래밍된 공차값은 삭제됩니다. ⇒ 머신 데이터에서 사전 설정된 공차 값이 다시 적용됩니다.

OTOLG0:	급이송 모션을 위해 임시로 유효 오리엔테이션 공차 팩터를 프로그래밍하는 주소			
	<Value>:	오리엔테이션 공차의 절대치		
		유형:	REAL	
		값:	≥ 0:	오리엔테이션 공차의 프로그래밍된 절대치는 갱신된 OTOLG0 프로그래밍으로 덮어쓰기 되거나, STOLF 프로그래밍으로 대체되거나, 채널 또는 프로그램 재설정 종료로 삭제될 때까지 유효합니다.
			<0:	프로그래밍된 공차값은 삭제됩니다. ⇒ 머신 데이터에서 사전 설정된 공차 값이 다시 적용됩니다.

참고

항상 마지막으로 프로그래밍된 주소가 우선 적용됩니다(다음 예 참고).

- CTOLG0이 활성 STOLF로 프로그래밍되면 CTOLG0으로 프로그래밍된 공차 값이 적용되어 형상을 부드럽게 합니다.
- OTOLG0이 활성 STOLF로 프로그래밍되면 OTOLG0으로 프로그래밍 된 공차 값이 적용되어 오리엔테이션을 부드럽게 합니다.
- STO를 다시 프로그래밍하면 형상 및 오리엔테이션 공차에 대한 공차 팩터가 적용됩니다.

예

예 1: 상대 G0 공차의 조정

프로그램 코드	설명
COMPCAD G645 G1 F10000	; 컴프레서 기능 COMPCAD
X... Y... Z...	; 머신 데이터 및 설정 데이터가 여기에 적용됩니다.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
G0 X... Y... Z...	; 머신 데이터 $MC_G0_TOLERANCE_FACTOR (예: 3) 이 여기에 적용됩니다. 예:
	$MC_G0_TOLERANCE_FACTOR * $MA_COMPRESS_POS_TOL의 스무딩 공차
CTOL=0.02	
STOLF=4	
G1 X... Y... Z...	; 여기부터 형상 공차 0.02 mm가 적용됩니다.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	

프로그램 코드	설명
X... Y... Z...	; 여기부터 G0 공차 팩터로 4가 적용됩니다. 즉, 형상 공차 0.08 mm가 적용됩니다.
...	

예 2: 절대 G0 공차의 조정

머신 데이터에서 다음 절대 G0 공차를 사전 설정해야 합니다.

- G0 형상 공차: 0.1
- G0 오리엔테이션 공차: 1.0

가공 프로그램에서 이러한 공차를 임시로 조정해야 합니다.

프로그램 코드	설명
COMPCAD G645 G1 F10000	; 컴프레서 기능 COMPCAD
X... Y... Z...	; 구성된 공작물 가공 공차가 여기에 적용됩니다.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
G0 X... Y... Z...	; 구성된 절대 G0 공차가 여기에 적용됩니다.
CTOLG0=0.2 OTOLG0=2.0	; 절대 G0 공차 프로그래밍
G1 X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
X... Y... Z...	; 프로그래밍된 G0 공차가 여기에 적용됩니다.
...	

추가 정보

G0 공차 팩터 판독

급이송 모션에 대한 현재 활성 공차 팩터는 시스템 변수를 사용해 읽을 수 있습니다.

- 시스템 변수를 통한 가공 프로그램에서의 동기 동작 또는 전처리 정지:
  $AC_STOLF 활성 G0 공차 팩터
  실제 메인 실행 블록을 처리할 때 적용된 G0 공차 팩터

- 시스템 변수를 통한 가공 프로그램의 전처리 정지 안 함:
  $P_STOLF 프로그램된 G0 공차 팩터

STOLF가 있는 값이 활성 가공 프로그램에서 프로그래밍되지 않은 경우 이러한 두 시스템 변수는 머신 데이터에서 구성된 값을 반환합니다.

블록에 사용 중인 급 이송 (G0) 이 없으면 이 시스템 변수는 항상 값 1을 제공합니다.

절대 G0 공차 판독

급이송 모션에 대한 현재 활성 절대 공차는 시스템 변수를 사용해 읽을 수 있습니다.

- 시스템 변수를 통한 가공 프로그램에서의 동기 동작 또는 전처리 정지:

  $AC_CTOL_G0_ABS G0 모션을 위한 활성 형상 공차
  현재의 메인 실행 블록이 전처리되었을 때 활성화된 G0 형상 공차

  $AC_OTOL_ G0_ABS G0 모션을 위한 활성 오리엔테이션 공차
  현재의 메인 실행 블록이 전처리되었을 때 활성화된 G0 오리엔테이션 공차

- 시스템 변수를 통한 가공 프로그램의 전처리 정지 안 함:

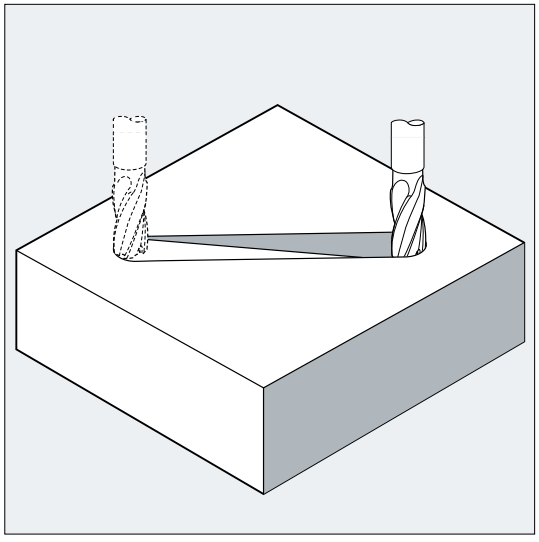
  $P_CTOL_ G0_ABS G0 모션을 위한 프로그래밍된 형상 공차

  $P_OTOL_ G0_ABS G0 모션을 위한 프로그래밍된 오리엔테이션 공차

CTOLG0 및 OTOLG0가 있는 절대 G0 공차가 활성 가공 프로그램에서 프로그래밍되지 않은 경우 이러한 두 시스템 변수는 머신 데이터에서 구성된 값을 반환합니다.

3.9.5 선형 보간(G1)

G1을 사용하여 공구가 축과 평행하게 기울어진 상태로 또는 직선으로 임의의 위치로 이동할 수 있습니다. 직선 보간을 사용하여 3D 표면, 홈, 등의 가공을 할 수 있습니다.



구문

```
G1 X… Y… Z … F…
G1 AP=… RP=… F…
```

의미

G1:	절삭 이송 (직선 보간)
X... Y... Z...:	직교 좌표의 종점
AP=...:	극 좌표의 종점, 이 경우 극점 각도
RP=...:	극 좌표의 종점, 이 경우 극점 반경
F...:	mm/분 단위 이송 속도. 공구는 현 시작점에서 프로그래밍된 종착점까지의 직선을 따라 이송 속도 F로 이동합니다. 직교 좌표 또는 극 좌표 내에 목적점을 입력할 수 있습니다. 공작물은 이 경로를 따라 가공됩니다. 예: G1 G94 X100 Y20 Z30 A40 F100 100 mm/min의 속도로 X, Y, Z 축의 종점으로 이송되며 로터리 축 A는 동기 축으로 이송이 이루어져 모든 4개의 축 이송이 동시에 완료됩니다.

참고

G1은 모달입니다.

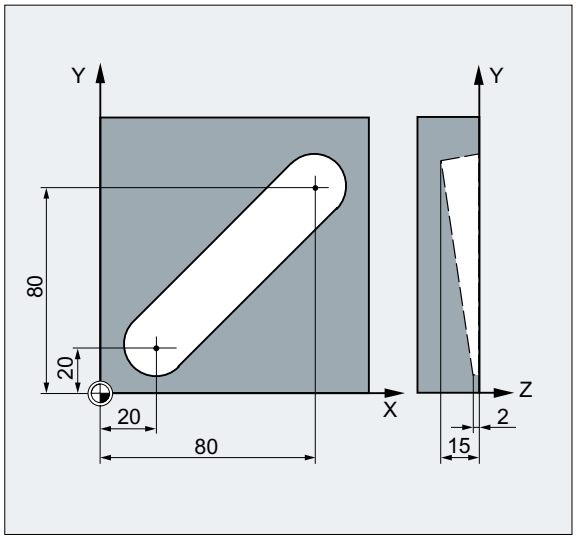
스핀들 속도 S와 스핀들 방향 M3/M4은 반드시 가공에 지정되어야 합니다.

경로 속도 F가 적용되는 축 그룹은 FGROUP으로 정의할 수 있습니다. 자세한 정보는 "경로 동작"을 참조 바랍니다.

예

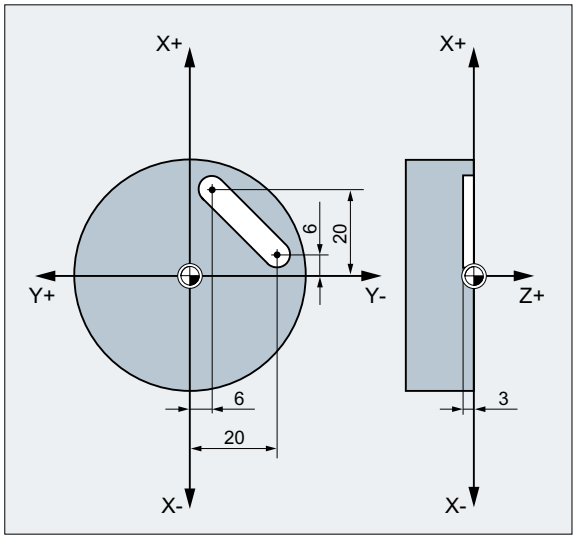
예 1: 홈 가공 (밀링)

공구는 시작점에서 X/Y방향으로 종점까지 이송됩니다. 동시에 Z 축 방향으로 절입이 이루어집니다.



프로그램 코드	코멘트
N10 G17 S400 M3	; 작업 평면의 선택, CW 방향 스핀들 회전
N20 G0 X20 Y20 Z2	; 시작 위치에 접근
N30 G1 Z-2 F40	; 공구 절입
N40 X80 Y80 Z-15	; 경사선 상의 이동
N50 G0 Z100 M30	; 공구 교환을 위한 후퇴

예 2: 홈 가공 (선삭)



프로그램 코드	코멘트
N10 G17 S400 M3	; 작업 평면의 선택, CW 방향 스핀들 회전

프로그램 코드	코멘트
N20 G0 X40 Y-6 Z2	; 시작 위치에 접근
N30 G1 Z-3 F40	; 공구 절입
N40 X12 Y-20	; 경사선 상의 이동
N50 G0 Z100 M30	; 공구 교환을 위한 후퇴

3.9.6 원호 보간

3.9.6.1 개요

원호 보간은 완전한 원이나 원호 가공을 가능케 해줍니다.

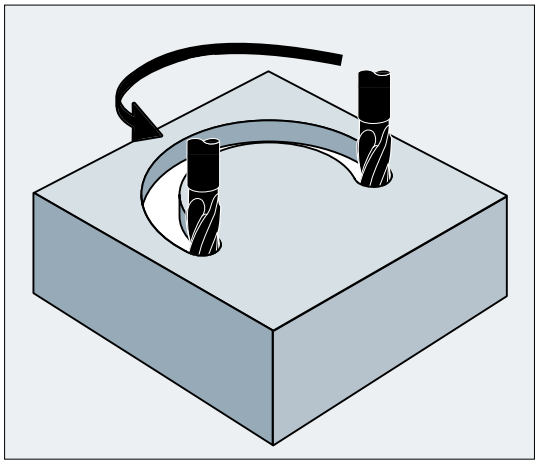


그림 3-3 적용 예: 원형 밀링

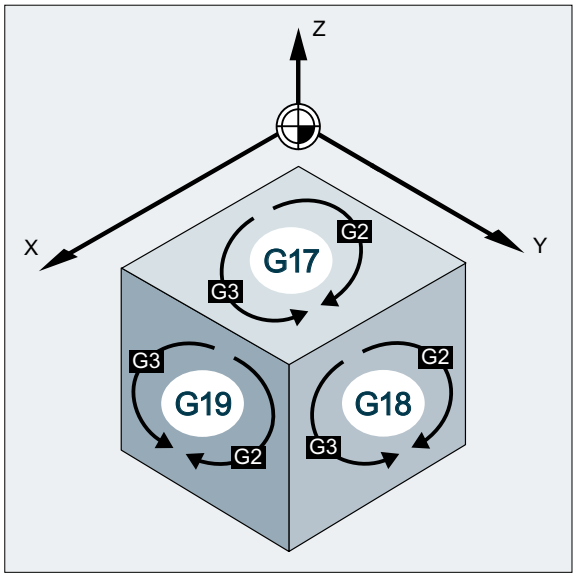
프로그래밍 옵션

제어 시스템은 원형 움직임 프로그래밍을 위한 다양한 옵션을 제공합니다. 이로써 사용자는 거의 모든 유형의 도면 치수를 직접 실행할 수 있습니다.

- 중심점과 끝점의 원 보간(G2/G3, X... Y... Z..., I... J... K...) (페이지 210)
- 반경과 종점의 원호 보간 (G2/G3, X... Y... Z..., CR) (페이지 212)
- 틈 각도 및 종점 / 중간점을 가진 원호 보간(G2/G3, X... Y... Z... / I... J... K..., AR) (페이지 214)
- 극 좌표로의 원 보간(G2/G3, AP, RP) (페이지 217)
- 중간점과 끝점의 원 보간(CIP, X... Y... Z..., I1... J1... K1...) (페이지 219)
- 접선 전이로의 원 보간 (페이지 222)

원호 보간을 위한 평면

원호의 회전 방향(G2 는 시계 방향 또는 G3 는 시계 반대 방향)을 계산함에 있어 제어는작업 평면 파라미터 (페이지 164) 를 필요로 합니다.



예외:

또한 선택된 작업 평면 밖에 원호를 생성할 수 있습니다(개방 각도가 지정된 경우 제외). 이 경우 프로그래머가 원호 종점으로 지정한 축 식별자가 원호 평면을 결정합니다.

3.9.6.2 중심점과 끝점의 원 보간(G2/G3, X... Y... Z..., I... J... K...)

보간을 위해 원호 형상 요소의 **중심점** 및 **종점** 을 사용하는 원호 보간 버전.

원호가 종점 없이 프로그래밍 된다면 결과는 완전한 원입니다.

구문

```
G2/G3 X… Y… Z… I… J… K…
G2/G3 X… Y… Z… I=AC(…) J=AC(…) K=(AC…)
```

의미

G2:	CW 방향 원호 보간	
	적용:	모달
G3:	CCW 방향 원호 보간	
	적용:	모달

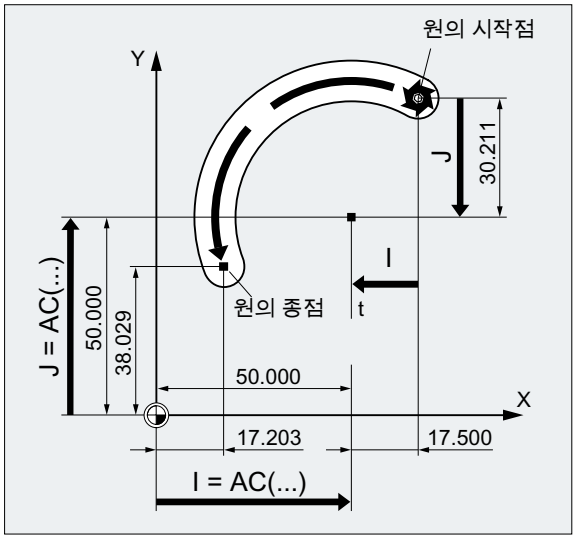
`X... Y... Z...`:	직교 좌표의 원호 종점 현재 유효한 치수 표기 설정 `G90`/`G91` 또는 `...=AC(...)` / `...=IC(...)` 에 따라 원호 종점 좌표는 절대 치수 또는 증분 치수로 해석됩니다.
`I... J... K...`:	원호 중심점 좌표를 X, Y, Z 방향으로 지정하기 위한 보간 파라미터 기본 설정 상 원호 중심점 좌표는 원호 시작점을 기준으로 한 증분 치수로 표시됩니다. 원호 중심점 좌표가 공작물 영점을 기준으로 한 절대 치수로 표시된다면 보간 파라미터 I, J, K는 반드시 다음과 같이 프로그래밍 되어야 합니다. `I=AC(...) J=AC(...) K=AC(...)` **참조** 값이 0인 보간 파라미터는 생략할 수 있지만, 연관된 2번째 파라미터는 반드시 항상 지정되어야 합니다.

참고

기본 설정 `G90`/`G91` 절대 또는 증분 치수는 원호 종점에 대해서만 유효합니다.

예

예 1: 밀링



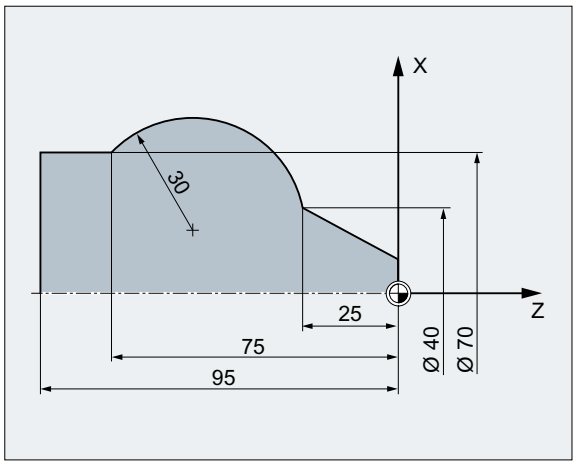
증분 치수를 이용한 중심점 데이터

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I-17.5 J-30.211 F500
```

절대 치수를 이용한 중심점 데이터

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I=AC(50) J=AC(50)
```

예 2: 선삭



증분 치수를 이용한 중심점 데이터

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I-3.335 K-29.25
N135 G1 Z-95
```

절대 치수를 이용한 중심점 데이터

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I=AC(33.33) K=AC(-54.25)
N135 G1 Z-95
```

3.9.6.3 반경과 종점의 원호 보간 (G2/G3, X... Y... Z..., CR)

보간을 위해 원호 형상 요소의 **반경** 및 **종점** 을 사용하는 원호 보간 버전.

참고

원(360 °를 이동)은 이 버전으로 프로그래밍 할 수 **없습니다**.

구문

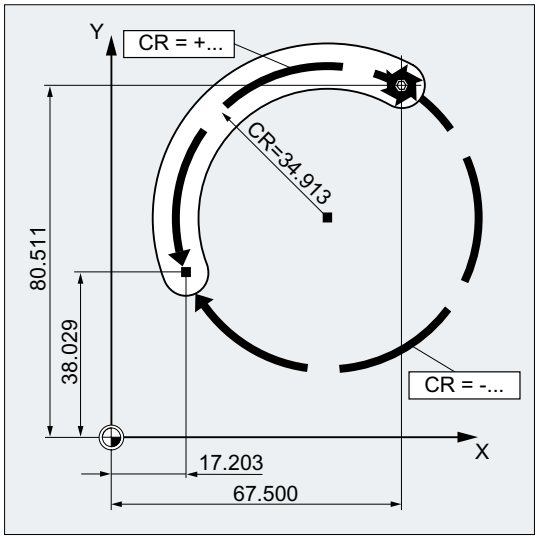
```
G2/G3 X… Y… Z… CR=±...
```

의미

G2:	CW 방향 원호 보간	
	적용:	모달
G3:	CCW 방향 원호 보간	
	적용:	모달
X... Y... Z...:	직교 좌표의 원호 종점 현재 유효한 치수 표기 설정 G90/G91 또는 ...=AC(...) / ...=IC(...) 에 따라 종점 좌표는 절대 치수 또는 증분 치수로 해석됩니다.	
CR=±...:	원호 반경 이 표시는 이송 각도가 180°보다 큰 지 아니면 작은 지를 알려줍니다. 양수 표시는 생략할 수 있습니다.	
	CR=+...:	이송 각도 ≤ 180°
	CR=-...:	이송 각도 > 180°
	참조 프로그램 가능한 반경의 최대 크기에 대해서는 별도의 제한이 없습니다.	

예

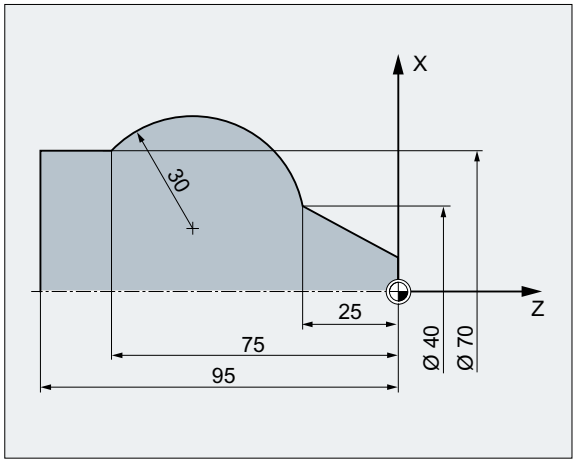
예 1: 밀링



프로그램 코드

```
N10 G0 X67.5 Y80.511
N20 G3 X17.203 Y38.029 CR=34.913 F500
...
```

예 2: 선삭



프로그램 코드

```
...
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 CR=30
N135 G1 Z-95
...
```

3.9.6.4 틈 각도 및 종점 / 중간점을 가진 원호 보간(G2/G3, X... Y... Z... / I... J... K..., AR)

보간을 위해 원호 형상 요소의 **틈 각도** 및 **중간점** 또는 **종점** 을 사용하는 원호 보간 버전.

참고

원(360 °를 이동)은 이 버전으로 프로그래밍 할 수 **없습니다**.

구문

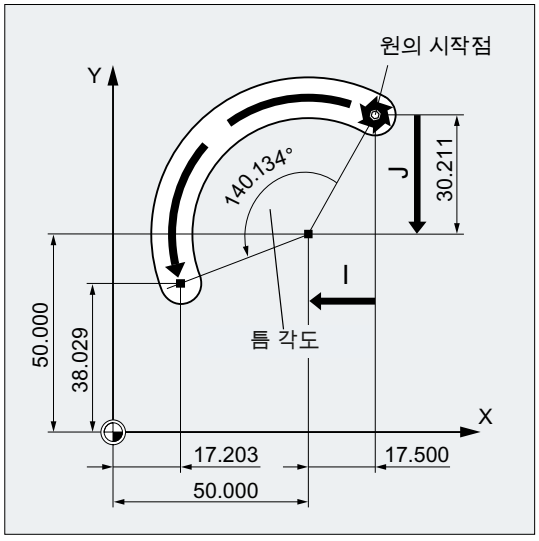
```
G2/G3 X… Y… Z… AR=...
G2/G3 I… J… K… AR=...
```

의미

G2:	CW 방향 원호 보간	
	적용:	모달
G3:	CCW 방향 원호 보간	
	적용:	모달
X... Y... Z...:	직교 좌표의 원호 종점 현재 유효한 치수 표기 설정 G90/G91 또는 ...=AC(...) / ...=IC(...) 에 따라 원호 종점 좌표는 절대 치수 또는 증분 치수로 해석됩니다.	
I... J... K...:	원호 중심점 좌표를 X, Y, Z 방향으로 지정하기 위한 보간 파라미터 기본 설정 상 원호 중심점 좌표는 원호 시작점을 기준으로 한 증분 치수로 표시됩니다. 원호 중심점 좌표가 공작물 영점을 기준으로 한 절대 치수로 표시된다면 보간 파라미터 I, J, K는 반드시 다음과 같이 프로그래밍 되어야 합니다. I=AC(...) J=AC(...) K=AC(...) **참조** 값이 0인 보간 파라미터는 생략할 수 있지만, 연관된 2번째 파라미터는 반드시 항상 지정되어야 합니다.	
AR=...:	틈 각도	
	값 범위:	0° ... 360°

예

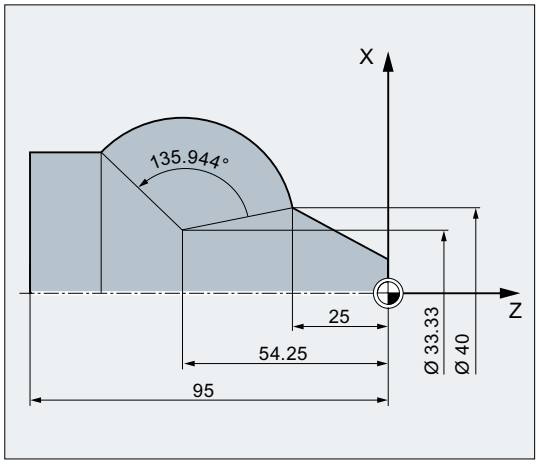
예 1: 밀링



프로그램 코드

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 AR=140.134 F500
N20 G3 I-17.5 J-30.211 AR=140.134 F500
```

예 2: 선삭



프로그램 코드

```
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 AR=135.944
```

프로그램 코드
N130 G3 I-3.335 K-29.25 AR=135.944
N130 G3 I=AC(33.33) K=AC(-54.25) AR=135.944
N135 G1 Z-95

3.9.6.5 극 좌표로의 원 보간(G2/G3, AP, RP)

보간을 위해 **극 좌표 내 원호 종점** 을 사용하는 원호 보간 버전.

다음과 같은 규칙이 적용됩니다:

- 극점은 원 중심에 오게 됩니다.
- 극점 반경은 원 반경과 일치합니다.

구문

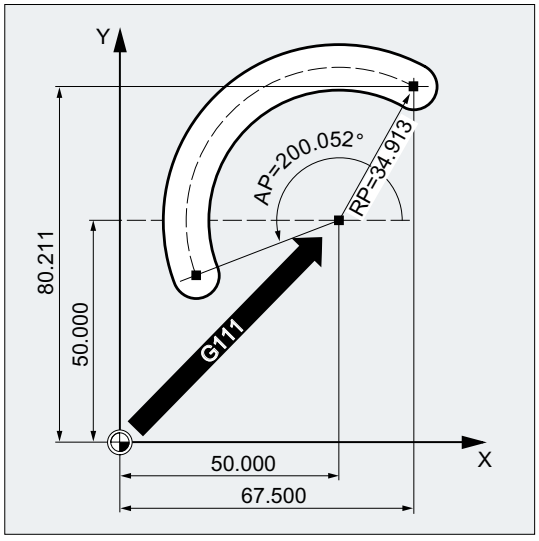
G2/G3 절대 압력=... 레시피 절차=...

의미

G2:	CW 방향 원호 보간	
	적용:	모달
G3:	CCW 방향 원호 보간	
	적용:	모달
절대 압력=... 레시피 절차=...:	직교 좌표 내 원호 종점.	
	절대 압력=...:	극점 각도
	레시피 절차=...:	극 반경 (≙ 원호 반경)

예

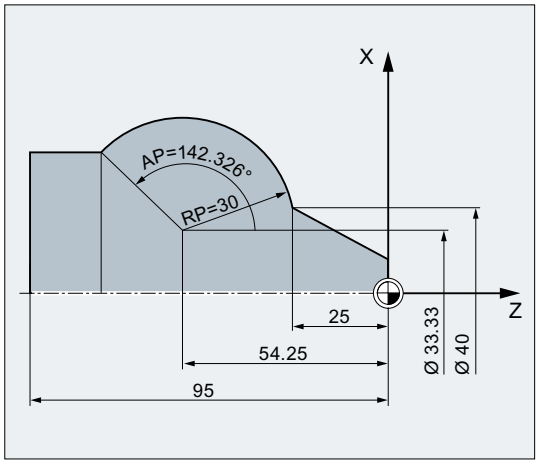
예 1: 밀링



프로그램 코드

```
N10 G0 X67.5 Y80.211
N20 G111 X50 Y50
N30 G3 RP=34.913 AP=200.052 F500
```

예 2: 선삭



프로그램 코드

```
N125 G1 X40 Z-25 F0.2
N130 G111 X33.33 Z-54.25
```

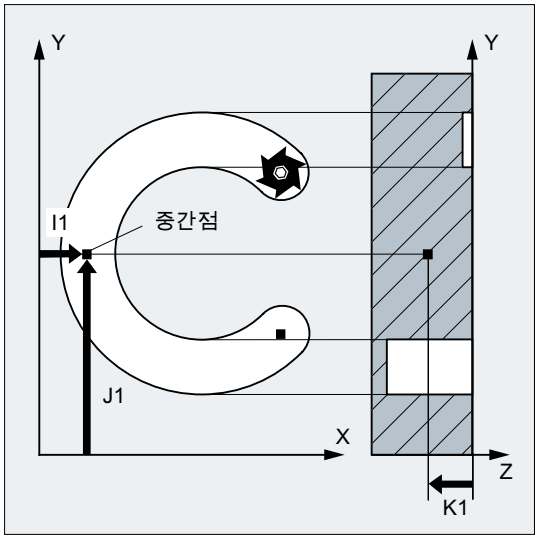
프로그램 코드
N135 G3 RP=30 AP=142.326
N140 G1 Z-95

3.9.6.6 중간점과 끝점의 원 보간(CIP, X... Y... Z..., I1... J1... K1...)

G 명령어 CIP 로 프로그래밍 된 원호 보간 버전은 공간 내 경사에 놓여 있는 원호의 보간을 허용합니다.

원호 모션은 원호 형상의 **중간점** 및 **종점** 에 의해 묘사됩니다.

이송 방향은 시작점 → 중간점 → 종점 순서로 결정됩니다.



구문

```
CIP X… Y… Z… I1=AC(…) J1=AC(…) K1=(AC…)
```

의미

CIP:	중간점을 통한 원호 보간	
	적용:	모달

X... Y... Z...:	직교 좌표의 원호 종점 현재 유효한 치수 표기 설정 G90/G91 또는 ...=AC(...) / ...=IC(...) 에 따라 원호 종점 좌표는 절대 치수 또는 증분 치수로 해석됩니다.
I1... J1... K1...:	원호 중간점 좌표를 X, Y, Z 방향으로 지정하기 위한 보간 파라미터 현재 유효한 치수 표기 설정 G90/G91 또는 ...=AC(...) / ...=IC(...) 에 따라 원호 중간점 좌표는 절대 치수 또는 증분 치수로 해석됩니다. **참조** 값이 0인 보간 파라미터는 생략할 수 있지만, 연관된 2번째 파라미터는 반드시 항상 지정되어야 합니다.

참고

기본 설정 G90/G91 (절대 또는 증분 치수)은 원호 중간점 및 종점을 사용한 경우에만 유효합니다.

증분 치수 G91 또는 ...=IC(...) 가 활성화된 상태에서는 원호 시작점이 중간점 및 종점에 대한 기준으로 사용됩니다.

참고

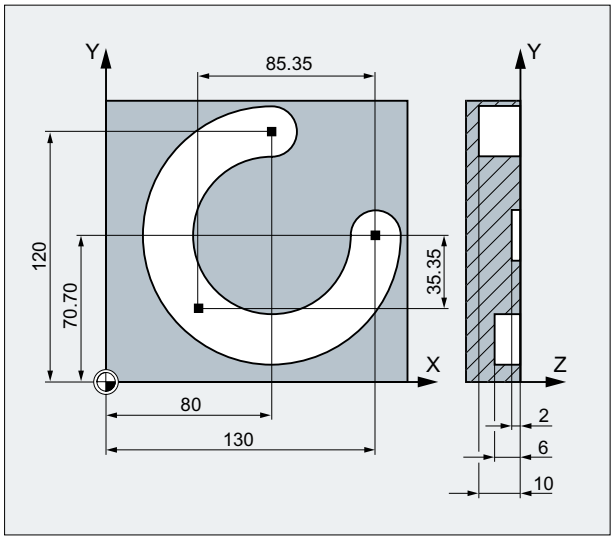
선삭 테크놀로지

가로 축에 대한 보간 파라미터의 직경 프로그래밍의 경우 원호 경로 프로그래밍에서 CIP에 의해 지원되지 않습니다. 그러므로 가로 축에 대한 보간 파라미터를 **반경**으로 프로그래밍해야 합니다.

예

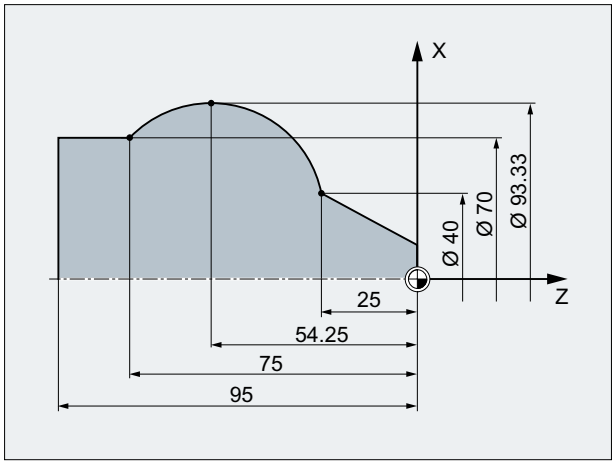
예 1: 밀링

경사진 원호 홈을 가공하려면 3개 보간 파라미터로 중간점을 지정하고 3개 좌표로 종점을 지정하여 원호를 지정합니다.



프로그램 코드	코멘트
N10 G0 G90 X130 Y70.70 S800 M3	; 시작점으로 접근.
N20 G17 G1 Z-2 F100	; 공구 절입.
N30 CIP X80 Y120 Z-10 I1=IC(-85.35) J1=IC(-35.35) K1=-6	; 원호 종점 및 경유 지점.
	; 세 기하 축 모두의 좌표.
N40 M30	; 프로그램 종료

예 2: 선삭



프로그램 코드	코멘트
...	
N125 G1 G90 X40 Z-25 F0.2	
N130 CIP X70 Z-75 **I1=IC(26.665)** K1=IC(-29.25)	; 이송 축에 대한 보간 파라미터 I1은 반경으로 프로그램해야 합니다.
; or	

프로그램 코드	코멘트
; N130 CIP X70 Z-75 **I1=46.665** K1=-54.25	
N135 G1 Z-95	

3.9.6.7 접선 전이로의 원 보간

G 명령어 CT 로 프로그램 된 원호 보간 버전은 이전에 프로그램 된 형상 요소에 접선으로 연결되는 원호 보간을 허용합니다.

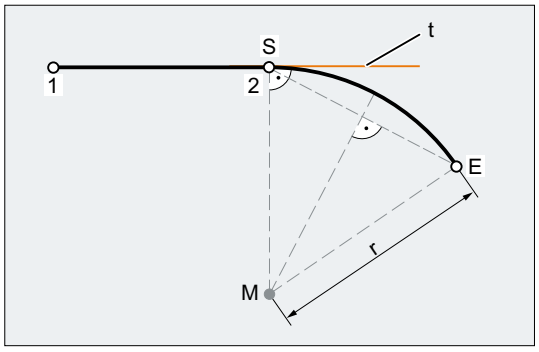
원호는 **시작점 및 종점**, 그리고 **시작점에서의 접선 방향** 에 의해 정의됩니다.

참고

시작점에서의 접선 방향.

CT 블록의 시작점에 있는 접선 방향은 이송 동작 마지막 블록의 프로그래밍된 끝 접선에서 결정됩니다.

이 블록과 현재 블록 간의 이송 정보가 없는 여러 개의 블록이 가능합니다.



S 시작점

E 종점

M 원호 중심

r 원호 반경

t 이송 움직움을 갖는 마지막 블록의 프로그램 된 형상의 끝 접선.

그림 3-4 원호 경로 S-E를 연결하는 직선 섹션 1-2에 대해 접선 형태

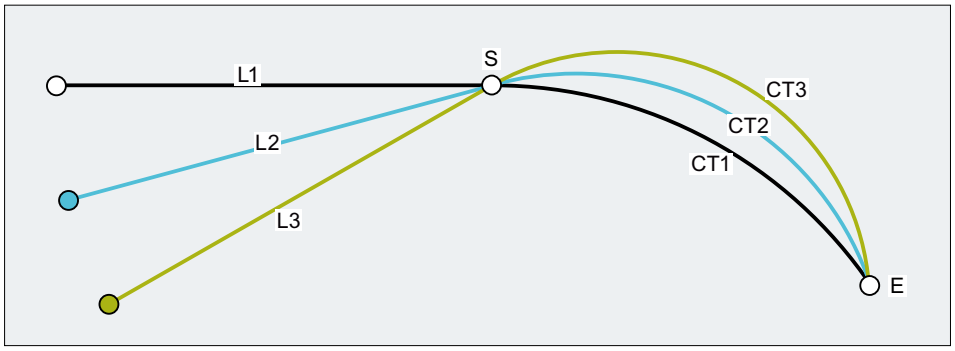


그림 3-5 원호 경로를 접선으로 연결하는 것은 이전 형상 요소에 달려 있습니다.

구문

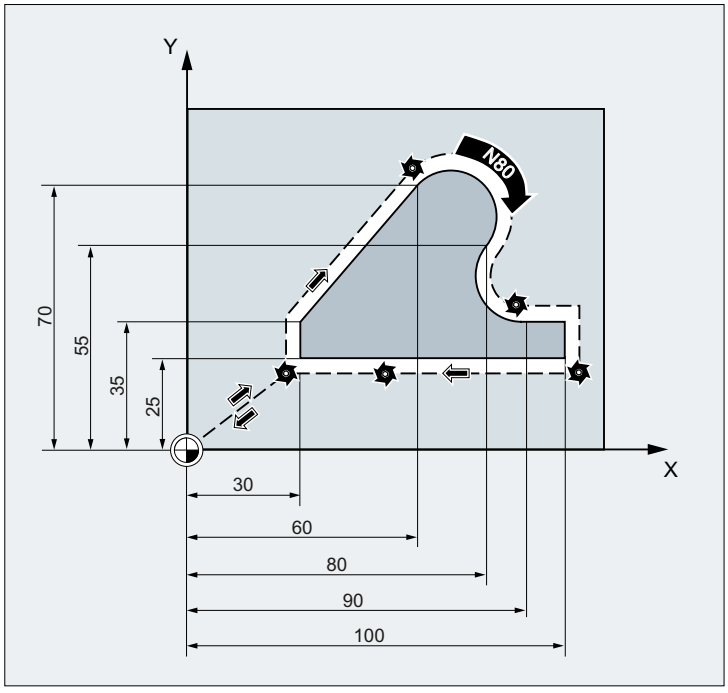
```
CT X... Y... Z...
```

의미

CT:	접선 트렌지션에 의한 원호 보간	
	적용:	모달
X... Y... Z...:	직교 좌표의 원호 종점 현재 유효한 치수 표기 설정 G90/G91 또는 ...=AC(...) / ...=IC(...) 에 따라 원호 종점 좌표는 절대 치수 또는 증분 치수로 해석됩니다.	

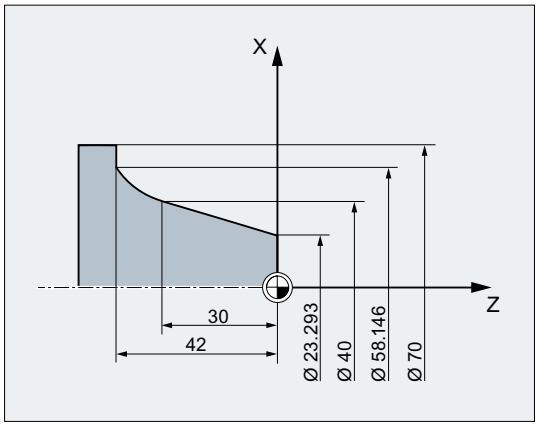
예

예 1: 밀링



프로그램 코드	코멘트
N10 G0 Z100	
N20 G17 T1 M6	
N30 G0 X0 Y0 Z2 M3 S300 D1	
N40 Z-5 F1000	; 공구 절입.
N50 G41 X30 Y25 G1 F1000	; 공구 반경 보정을 켬.
N60 Y35	; 형상 밀링.
N70 X60 Y70	
N80 CT X80 Y55	**; 접선 트렌지션을 사용한 원호 경로 프로그래밍.**
N90 X90 Y35	
N100 G1 X100	
N110 Y25	
N120 X30	
N130 G0 G40 X0 Y0	; 공구 반경 보정을 끔.
N140 Z100	; 공구 후퇴.
N140 M30	

예 2: 선삭



프로그램 코드	코멘트
...	
N110 G1 X23.293 Z0 F10	
N115 X40 Z-30 F0.2	
N120 CT X58.146 Z-42	**; 접선 트렌지션을 사용한 윤곽 경로 프로그래밍.**
N125 G1 X70	
...	

추가 정보

스플라인

스플라인의 경우 접선 방향은 마지막 두 점을 통과한 직선에 의해 정의됩니다. ENAT 또는 EAUTO가 활성 상태인 A 및 C 스플라인의 경우 일반적으로 이 방향은 스플라인 종점의 방향과 동일하지 않습니다.

B 스플라인은 항상 접선을 이루고 접선 방향은 A 또는 C 스플라인과 적용 중인 ETAN으로 정의됩니다.

프레임 변경

접선 및 CT 블록을 정의하는 블록 간에 프레임 변경이 발생된 경우, 접선은 항상 이 변경을 따릅니다.

제한 경우

시작 접선의 연장선이 종점을 통과하면 원 대신에 직선이 만들어 집니다(제한 경우: 무한대의 반경을 가진 원호). 이런 특수한 경우 TURN이 절대 프로그래밍되지 않거나 그 값은 TURN=0 일 것입니다.

참고

값이 이 제한 경우에 도달하려 할 때 무제한 반경의 원이 만들어 지고 0이 아닌 TURN을 이용한 가공은 일반적으로 소프트웨어 한계를 위반하게 되어 알람과 함께 취소됩니다.

원호 평면의 위치

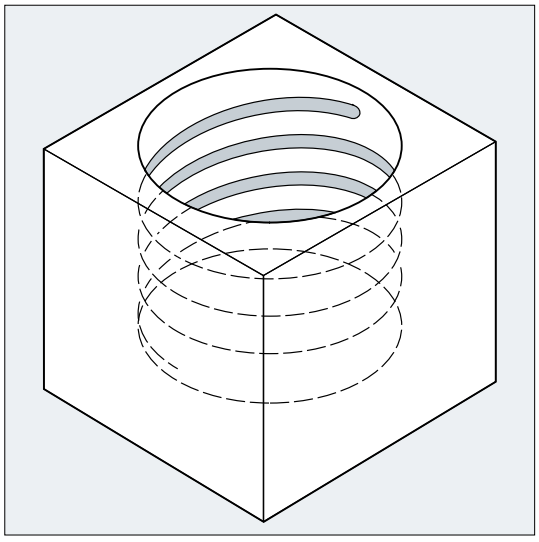
원호 평면의 위치는 동작 중인 작업 평면 (G17-G19) 에 따라 달라집니다.

이전 블록의 접선이 활성 평면에 있지 않게 된다면 활성 평면으로의 투영이 사용됩니다.

시작점과 종점이 적용중인 작업 평면에 직각이지 않을 경우, 원호 대신 나선이 만들어집니다.

3.9.7 헬리컬 보간 (G2/G3, TURN)

예를 들어 헬리컬 보간으로 나사산 또는 오일 홈의 제작이 가능해집니다.



헬리컬 보간으로 두 개의 동작이 겹쳐지고 평행하게 실행됩니다.

- 평면 원호 모션 위에
- 수직 직선 모션이 중첩됩니다.

구문

```
G2/G3 X… Y… Z… I… J… K… TURN=

G2/G3 X… Y… Z… I… J… K… TURN=

G2/G3 AR=… I… J… K… TURN=

G2/G3 AR=… X… Y… Z… TURN=

G2/G3 AP… RP=… TURN=
```

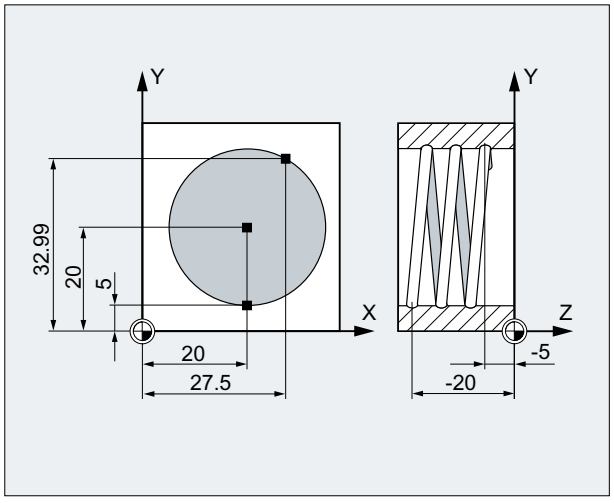
의미

G2:	CW 방향으로 원호 경로 상 이동
G3:	CCW 방향으로 원호 경로를 따라 이동
X Y Z:	직교 좌표의 종점
I J K:	직교 좌표의 원호 중심점
AR:	개방 각도
TURN=:	0 에서 999 까지 범위에서 추가적인 원호 패스 횟수
AP=:	극점 각도
RP=:	극점 반경

참고

G2 와 G3는 모달입니다.

예



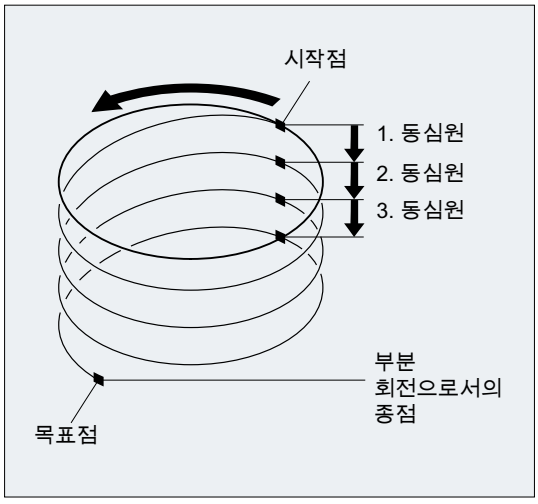
프로그램 코드	설명
N10 G17 G0 X27.5 Y32.99 Z3	; 시작 위치에 접근.
N20 G1 Z-5 F50	; 공구 절입.
N30 G3 X20 Y5 Z-20 I=AC(20) J=AC(20) TURN=2	; 나선 지정: 시작 위치 이후 완전한 원 2개를 실행하고 종점으로 이동합니다.
N40 M30	; 프로그램 종료

추가 정보

작동 순서

1. 시작점으로 이동
2. TURN=으로 프로그래밍된 완전한 원호를 생성.
3. 소재를 회전시켜 원호 종점으로 이동
4. 절입 깊이까지 2~3번 단계를 반복 실행

나선이 가공될 피치는 원호의 개수와 프로그램된 원호의 종점으로부터 계산됩니다. (절입 깊이에 도달될 때까지 실행됩니다.)



헬리컬 보간에서 종점 프로그래밍

보간 파라미터에 대한 세부 설명은 원호 보간을 참조하시오.

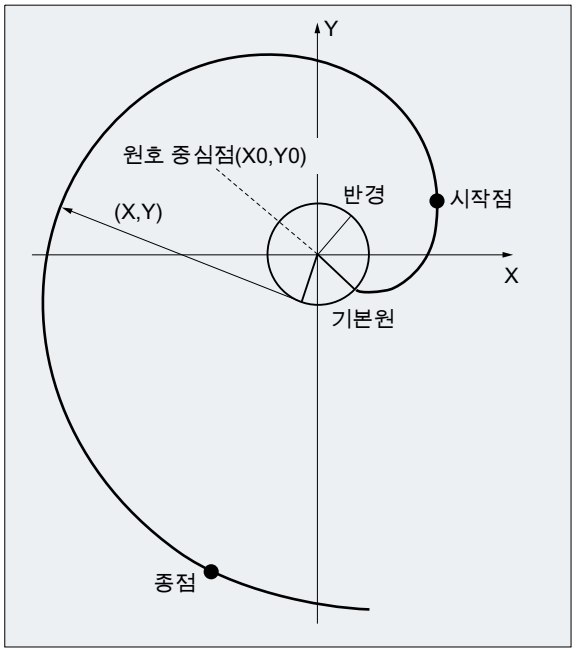
프로그래밍된 이송 속도

헬리컬 보간의 경우 프로그래밍된 이송 속도 오버라이드 (CFC) 를 지정하는 것이 좋습니다. FGROUP 를 이용하면 프로그래밍된 이송 속도로 이송될 축을 지정할 수 있습니다. 자세한 내용은 경로 동작 단락을 참조하십시오.

3.9.8 내향 나선 보간(INVCW, INVCCW)

원의 나선은 곡선 주위를 풀 때 이상적인 문자열의 끝이 차지하는 경로입니다.

나선형 보간으로 나선을 따른 경로 궤적이 가능합니다. 이는 기본 원호가 정의된 평면에서 실행되며 프로그램된 시작점에서 프로그램된 종점까지 실행됩니다.



종점은 두 가지 방법으로 프로그래밍될 수 있습니다:

1. 직교 좌표를 통한 직접 프로그래밍
2. 틈 각도 지정을 통한 간접 프로그래밍(또한 원호 경로 프로그래밍의 틈 각도 프로그래밍을 참고)

시작점과 종점이 기본 원호 평면에 있고 원의 헬리컬 보간과 유사하다면 공간에서 커브가 서로 겹치게 됩니다.

활성 평면에 직각인 경로를 추가적으로 지정하여 공간 상에서 나선이 이송될 수 있습니다 (원에 대한 헬리컬 보간과 비교 가능).

구문

```
INVCW X... Y... Z... I... J... K... CR=...
INVCCW X... Y... Z... I... J... K... CR=...
INVCW I... J... K... CR=... AR=...
INVCCW I... J... K... CR=... AR=...
```

의미

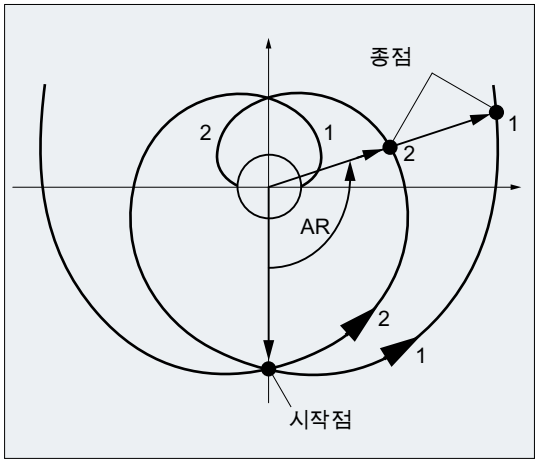
INVCW:	시계 방향으로 나선을 따라 이동 명령
INVCCW:	시계 반대 방향으로 나선을 따라 이동 명령
X... Y... Z...:	직교 좌표 종점 직접 프로그래밍

I... J... K...:	직교 좌표의 기본 원호 중심점을 지정하는 보간 파라미터 **참고:** 좌표 지정은 나선의 시작점을 기준합니다.	
CR=...:	기본 원호의 반경	
AR=...:	틈 각도(회전 각도) 지정을 통한 종점의 간접 프로그래밍 틈 각도의 시초는 원호 중심점에서 시작점까지의 선입니다.	
	AR > 0:	나선의 경로는 **기본 원호에서 멀어지는 쪽**으로 이동합니다.
	AR < 0:	나선의 경로는 **기본 원호를 향하여** 이동합니다. AR<0 의 경우 회전의 최대 각도는 종점이 항상 기본 원호의 바깥에 있어야 한다는 현실 때문에 제한됩니다.

틈 각도 지정을 통한 종점의 간접 프로그래밍

유의사항
틈 각도 미정의 틈 각도 AR 지정을 통해 종점을 간접 프로그래밍하는 경우 부호 변경이 다른 나선과 경로가 만들어지므로 각도의 부호를 반드시 계산해야 합니다.

이는 다음의 예로 예시됩니다:

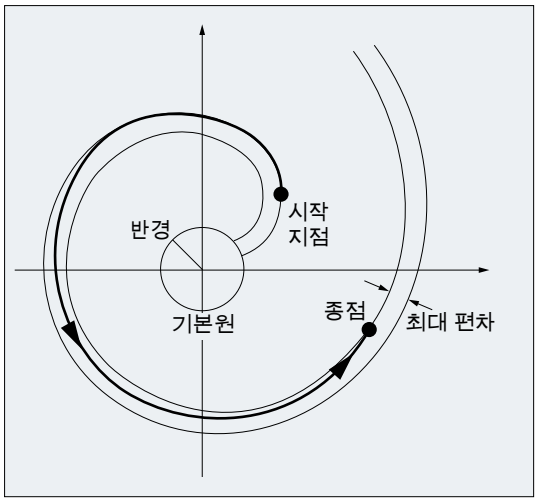


기본 원호의 반경 및 중심점과 방향의 시작점 지정(INVCW/INVCCW)은 나선 1과 2에 대해 동일합니다. 유일한 차이점은 틈 각도의 기호입니다.

- AR > 0 인 경우 경로는 나선 1에 있게 되고 종점 1에 접근하게 됩니다.
- AR < 0 인 경우 경로는 나선 2에 있게 되고 종점 2에 접근하게 됩니다.

보충 조건

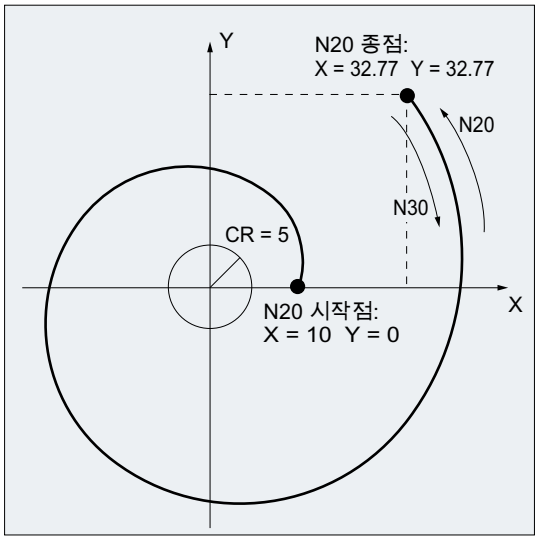
- 시작점과 종점 둘은 반드시 나선의 기본 원호 바깥 구역에 있어야 합니다 (I, J, K로 지정된 중심점 주위의 반경 CR의 원). 이 조건이 충족되지 않으면 알람이 생성되고 프로그램 처리가 취소됩니다.
- 종점 프로그래밍의 두 옵션(직교 좌표를 통한 직접 방식 또는 틈 각도의 지정을 통한 간접 방식)은 상호 배타적입니다. 당연히 두 프로그래밍 옵션 중 하나만이 한 블록에서 사용될 수 있습니다.
- 프로그래밍된 종점이 시작점과 기본 원호에서 정의한 나선에 정확하게 위치하지 않으면 시작과 종점에서 정의한 두 나선 사이에 보간이 발생합니다. (다음 그림을 참조하시오.)



종점의 최대 편차는 머신 데이터에 의해 결정됩니다 (→ 장비 제조업체). 방사형 방향으로 프로그래밍된 종점의 편차가 MD에 정의된 것보다 큰 경우 알람이 생성되고 프로그램 처리가 취소됩니다.

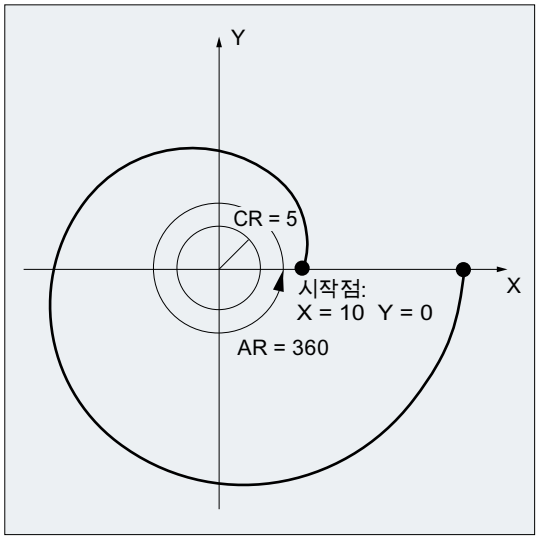
예

예 1: 시작점에서 프로그래밍된 종점까지 CCW 방향 나선을 따라 이송 후 다시 CW 방향 나선으로 복귀



프로그램 코드	설명
N10 G1 X10 Y0 F5000	; 시작점에 접근합니다.
N15 G17	; X/Y 평면을 가공 평면으로 선택
N20 INVCCW X32.77 Y32.77 CR=5 I-10 J0	; CCW 방향 나선, 직교 좌표의 종점.
N30 INVCW X10 Y0 CR=5 I-32.77 J-32.77	; CW 방향 나선. N20에서의 종점이 시작점이 되고 새 종점 위치는 N20부터 시작됩니다. 새 원호 중심점은 새 시작점을 기준으로 하고 이는 이전 원호의 중심점과 같게 됩니다.
...	

예 2: 틈 각도 지정을 통한 종점의 간접 프로그래밍을 이용한 시계 반대 방향 나선



프로그램 코드	설명
N10 G1 X10 Y0 F5000	; 시작점에 접근합니다.
N15 G17	; X/Y 평면을 가공 평면으로 선택
N20 INVCCW CR=5 I-10 J0 AR=360	; CCW 방향 나선으로 이동한 후 1회 완전 회전 (360°) 으로 기본 원호 (양의 각도 지정) 에서 후퇴.
...	

3.9.9 형상 정의

3.9.9.1 형상 정의 프로그래밍

기능

형상 정의 프로그램을 사용하여 단순한 형상을 빨리 입력할 수 있습니다.

형상 정의는 직교 좌표 및/또는 각도 (ANG 또는 ANG1 및 ANG2) 지정을 통해 1개, 2개 또는 3개 지점과 트렌지션 요소 모따기 또는 라운딩으로 프로그래밍합니다.

추가 축을 위한 주소 문자(단일 축 또는 가공 평면에 수직인 축), 보조 기능 사양, G 명령, 속도 등과 같은 임의적인 추가 NC 주소를 형상 정의를 설명하는 블록에서 사용할 수 있습니다.

참고

형상 계산기

형상 정의는 형상 계산기를 사용하여 쉽게 프로그래밍될 수 있습니다. 이를 통해, 단순하거나 또는 복잡한 공작물 형상을 프로그램 할 수 있으며 또한 프로그램 형상을 그래픽으로 표시할 수 있습니다. 형상 계산기로 프로그래밍된 형상은 가공 프로그램으로 전송됩니다.

추가 정보: 사용 매뉴얼

파라미터 설정

각도, 코너 라운딩 및 모따기는 머신 데이터를 통해 정의됩니다.

MD10652 $MN_CONTOUR_DEF_ANGLE_NAME (형상 정의의 각도 이름)

MD10654 $MN_RADIUS_NAME (형상 정의의 코너 라운딩 이름)

MD10656 $MN_CHAMFER_NAME (형상 정의의 모따기 이름)

참고

기계 제조업체의 사양을 참고하십시오.

3.9.9.2 형상 정의 직선 1개

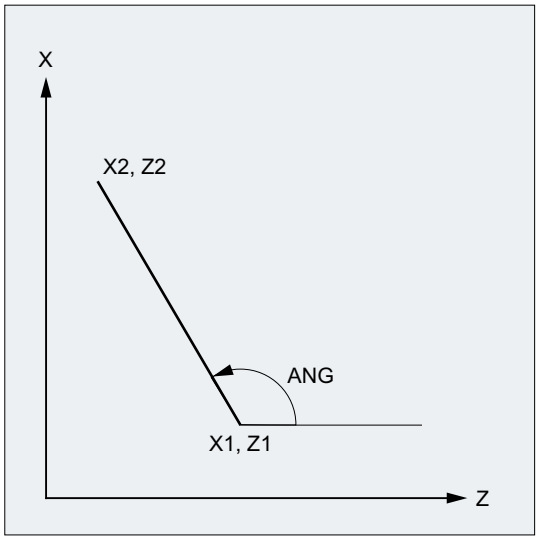
참고

아래 설명에 앞서 다음 사항을 가정합니다.

- G18이 활성 상태입니다 (⇒ 활성 작업 평면이 Z/X 평면입니다).
  (단, G17 또는 G19를 사용하면 제한 없이 형상 정의를 프로그래밍할 수 있습니다.)
- 각도, 반경 및 모따기에 다음과 같은 식별자가 지정되어 있습니다.
  - ANG (각도)
  - RND (반경)
  - CHR (모따기)

직선의 종점은 다음 지정으로 정의됩니다.

- 각도 ANG
- **하나**의 직교 종점 좌표 (X2 또는 Z2)



ANG: 직선의 각도

X1, Z1: 시작 좌표

X2, Z2: 직선의 종점 좌표

구문

```
X... ANG=...
Z... ANG=...
```

의미

X...:	X방향 종점 좌표
Z...:	Z방향 종점 좌표
ANG:	각도 프로그래밍의 식별자 지정된 값 (각도) 은 활성 작업 평면 (G18로 지정된 Z 축) 의 가로 좌표를 기준으로 합니다.

예

프로그램 코드	코멘트
N10 X5 Z70 F1000 G18	; 시작 위치에 접근
N20 X88.8 ANG=110	; 각도가 지정된 직선
N30 ...	

또는

프로그램 코드	코멘트
N10 X5 Z70 F1000 G18	; 시작 위치에 접근
N20 Z39.5 ANG=110	; 각도가 지정된 직선
N30 ...	

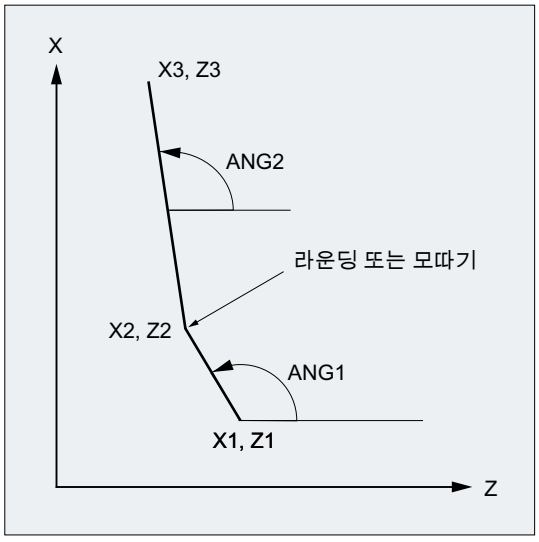
3.9.9.3 형상 정의 두 개의 직선

참고

아래 설명에 앞서 다음 사항을 가정합니다.

- G18이 활성 상태입니다 (⇒ 활성 작업 평면이 Z/X 평면입니다).
  (단, G17 또는 G19를 사용하면 제한 없이 형상 정의를 프로그래밍할 수 있습니다.)
- 각도, 반경 및 모따기에 다음과 같은 식별자가 지정되어 있습니다.
  - ANG (각도)
  - RND (반경)
  - CHR (모따기)

첫 번째 직선의 종점은 직교 좌표 또는 두 직선의 각도를 지정하여 프로그래밍될 수 있습니다. 두 번째 직선의 종점은 항상 반드시 직교 좌표로 프로그래밍되어야 합니다. 두 직선의 교차는 코너, 곡면 또는 모따기로 설계될 수 있습니다.



ANG1: 첫 번째 직선의 각도

ANG2: 두 번째 직선의 각도

X1, Z1: 첫 번째 직선의 시작 좌표

X2, Z2: 첫 번째 직선의 종점 좌표 또는
두 번째 직선의 시작 좌표

X3, Z3: 두 번째 직선의 종점 좌표

구문

각도를 지정하여 첫 번째 직선의 종점 프로그래밍

- 직선 간 트렌지션으로 코너:

```
ANG=...
X… Z… ANG=…
```

- 직선 간 트렌지션으로 라운딩:

```
ANG=… RND=...
X… Z… ANG=…
```

- 직선 간 트렌지션으로 모따기:

```
ANG=… CHR=...
X… Z… ANG=…
```

좌표를 지정하여 첫 번째 직선의 종점 프로그래밍

- 직선 간 트렌지션으로 코너:

```
X... Z...
X... Z...
```

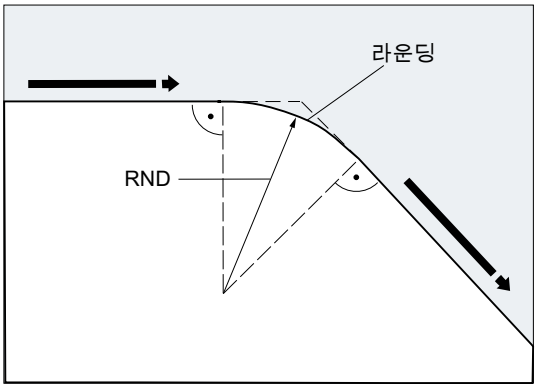
- 직선 간 트렌지션으로 라운딩:

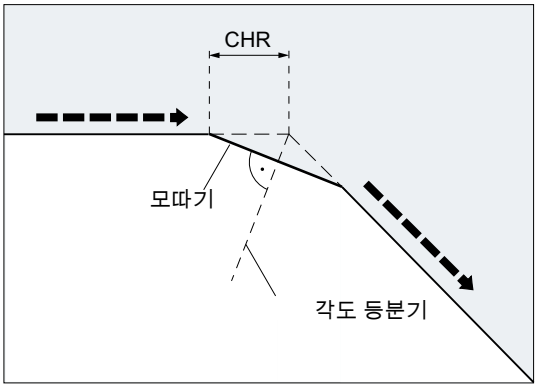
```
X... Z... RND=...
X... Z...
```

- 직선 간 트렌지션으로 모따기:

```
X... Z... CHR=...
X... Z...
```

의미

ANG=... :	각도 프로그래밍의 식별자 지정된 값 (각도) 은 활성 작업 평면 (G18로 지정된 Z 축) 의 가로 좌표를 기준으로 합니다.
RND=... :	라운딩 프로그래밍 식별자 지정된 값이 라운딩의 반경이 됩니다. 

CHR=… :	모따기 프로그래밍 식별자 지정된 값이 이동 방향에서 모따기 너비가 됩니다. 
X...:	X 방향 좌표
Z...:	Z 방향의 좌표

참고

모따기 또는 라운딩 프로그래밍에 대한 추가 정보는 "모따기, 라운딩 (CHF, CHR, RND, RNDM, FRC, FRCM) (페이지 271)"을 참조 바랍니다.

예

프로그램 코드	코멘트
N10 X10 Z80 F1000 G18	; 시작 위치에 접근.
N20 ANG=148.65 CHR=5.5	; 각도와 모따기 지정의 직선.
N30 X85 Z40 ANG=100	; 각도와 종점 지정의 직선.
N40	

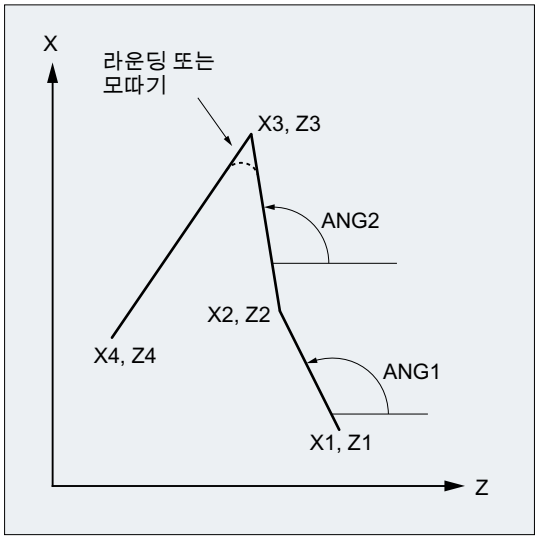
3.9.9.4 형상 정의 직선 3개

참고

아래 설명에 앞서 다음 사항을 가정합니다.

- G18이 활성 상태입니다 (⇒ 활성 작업 평면이 Z/X 평면입니다).
  (단, G17 또는 G19를 사용하면 제한 없이 형상 정의를 프로그래밍할 수 있습니다.)
- 각도, 반경 및 모따기에 다음과 같은 식별자가 지정되어 있습니다.
  - ANG (각도)
  - RND (반경)
  - CHR (모따기)

첫 번째 직선의 종점은 직교 좌표 또는 두 직선의 각도를 지정하여 프로그래밍될 수 있습니다. 두 번째와 세 번째 직선의 종점은 항상 반드시 직교 좌표로 프로그래밍되어야 합니다. 직선의 교차는 코너, 곡면 또는 모따기로 설계될 수 있습니다 .



ANG1: 첫 번째 직선의 각도

ANG2: 두 번째 직선의 각도

X1, Z1: 첫 번째 직선의 시작 좌표

X2, Z2: 첫 번째 직선의 종점 좌표 또는
두 번째 직선의 시작 좌표

X3, Z3: 두 번째 직선의 종점 좌표 또는
세 번째 직선의 시작 좌표

X4, Z4: 세 번째 직선의 종점 좌표

참고

세 점 형상 정의로 여기에 기술된 프로그래밍은 세 점 이상의 형상 정의로 임의 확장될 수 있습니다.

구문

각도를 지정하여 첫 번째 직선의 종점 프로그래밍

- 직선 간 트렌지션으로 코너:

```
ANG=...
X… Z… ANG=…
X… Z…
```

- 직선 간 트렌지션으로 라운딩:

```
ANG=… RND=...
X… Z… ANG=… RND=...
X… Z…
```

- 직선 간 트렌지션으로 모따기:

```
ANG=… CHR=...
X… Z… ANG=… CHR=...
X… Z…
```

좌표를 지정하여 첫 번째 직선의 종점 프로그래밍

- 직선 간 트렌지션으로 코너:

```
X… Z…
X… Z…
X… Z…
```

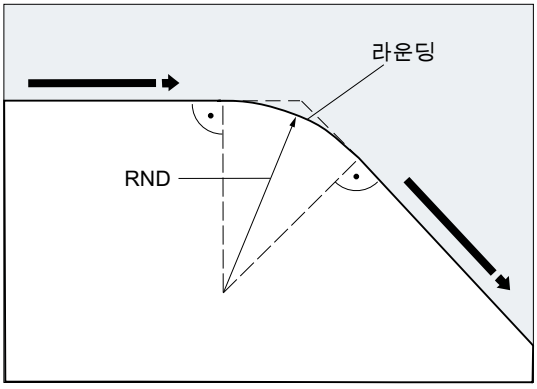
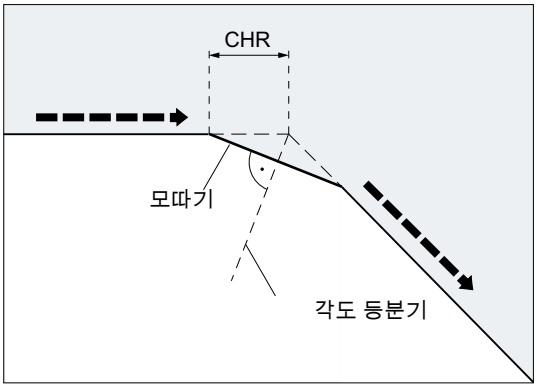
- 직선 간 트렌지션으로 라운딩:

```
X… Z… RND=...
X… Z… RND=...
X… Z…
```

- 직선 간 트렌지션으로 모따기:

```
X… Z… CHR=...
X… Z… CHR=...
X… Z…
```

의미

ANG=...:	각도 프로그래밍의 식별자 지정된 값 (각도) 은 활성 작업 평면 (G18로 지정된 Z 축) 의 가로 좌표를 기준으로 합니다.
RND=…:	라운딩 프로그래밍 식별자 지정된 값이 라운딩의 반경이 됩니다. 
CHR=…:	모따기 프로그래밍 식별자 지정된 값이 이동 방향에서 모따기 너비가 됩니다. 
X...:	X 방향 좌표
Z...:	Z 방향의 좌표

참고

모따기와 라운딩에 관한 자세한 정보는 " 모따기, 라운딩(CHF, CHR, RND, RNDM, FRC, FRCM) (페이지 271) "을 참조하십시오.

예

프로그램 코드	코멘트
N10 X10 Z100 F1000 G18	; 시작 위치에 접근
N20 ANG=140 CHR=7.5	; 각도와 모따기 지정의 직선.
N30 X80 Z70 ANG=95.824 RND=10	; 각도와 모따기 지정의 중간점까지의 직선.
N40 X70 Z50	; 종점까지의 직선.

3.9.9.5 형상 정의 각도로의 끝점 프로그래밍

기능

어드레스 레터 A가 NC 블록에 나타나면 활성 면의 축이 프로그래밍되지 않거나 하나 또는 두 축이 프로그래밍될 수 있습니다.

프로그래밍된 축의 수

- 활성 면의 **어떠한 축도** 프로그래밍되지 않았다면 이는 두 블록으로 구성된 형상 정의의 첫 또는 두 번째 블록 중 하나 입니다.
  만약 형상 정의의 두 번째 블록이라면 시작점과 종점이 활성 평면에서 동일하다는 의미가 됩니다. 형상 정의는 이때 활성 면에 직각인 동작일 뿐이다.
- 활성 면의 **정확히 하나의 축**이 프로그래밍되었다면 이 때 이는 각도와 프로그래밍된 직교 좌표를 통해 명확하게 정의될 수 있는 종점을 가진 단일 직선 또는 두 블록으로 구성된 형상 정의의 두 번째 블록 중 하나입니다. 두 번째 경우에서 누락 좌표는 최종 (모달) 도달한 위치와 동일하게 설정됩니다.
- 활성 면의 **두 축**이 프로그래밍되었다면 이는 두 블록으로 구성된 형상 정의의 두 번째 블록 입니다. 활성 면의 프로그래밍된 축이 없는 각도 프로그래밍의 블록이 현재 블록 앞에 없다면 이 블록은 허용되지 않습니다.

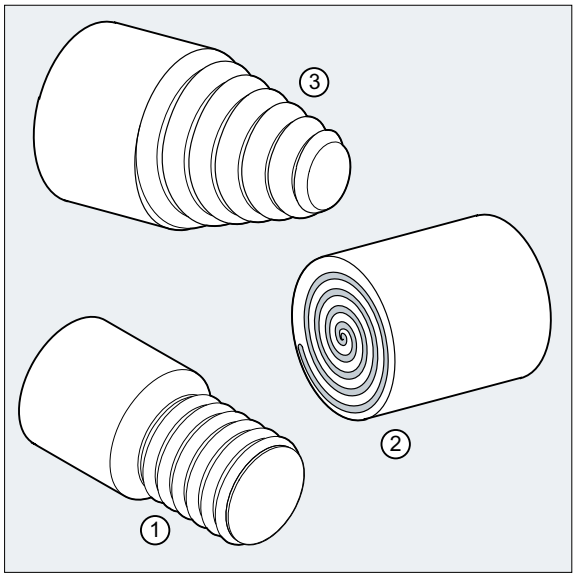
각도 A는 직선 또는 스플라인 보간에만 프로그래밍될 수 있습니다.

3.9.10 나사 절삭

3.9.10.1 연속 리드를 사용하여 나사 절삭(G33, SF)

연속 리드의 나사는 G33을 사용하여 가공될 수 있습니다:

- 실린더 나사 ①
- 단면 나사 ②
- 테이퍼 나사 ③

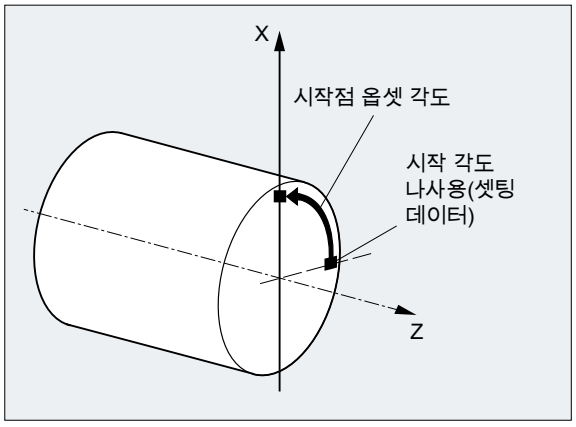


참고

G33을 사용한 나사 절삭에는 측정 시스템이 사용된 가변 속도 스핀들이 필요합니다.

다중 나사

다중 나사 (옵셋 절삭의 나사) 는 시작점 옵셋을 지정하여 가공할 수 있습니다. 프로그래밍은 G33블록 어드레스SF에서 수행됩니다.

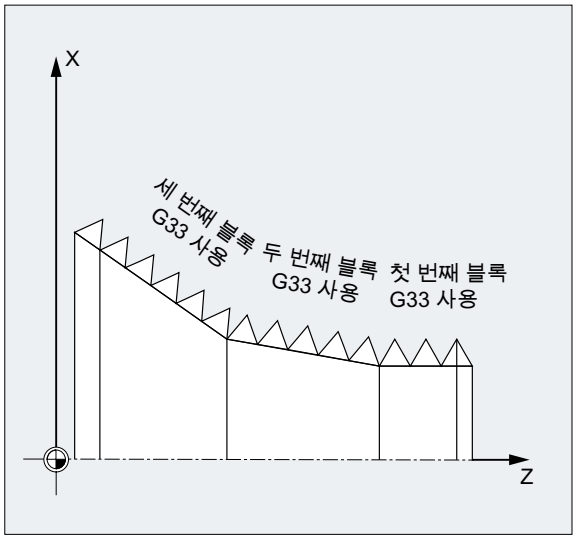


참고

시작점 옵셋이 명시되지 않으면 셋팅 데이터에서 정의한 "나사의 시작 각도"가 사용됩니다.

나사 체인

나사 체인은 연속으로 프로그래밍된 여러 개의 G33 블록으로 가공될 수 있습니다.



참고

연속 경로 모드 G64로 블록은 전면 감지 속도 제어에 연결되어 속도 점프가 발생하지 않습니다.

나사 회전의 방향

나사 회전 방향은 스핀들 회전 방향으로 결정됩니다.

- M3의 시계방향은 오른 나사를 만듭니다.
- M4가 CCW 방향이면 왼 나사가 만들어집니다.

구문

원통형 나사:
```
G33 Z… K…
G33 Z… K… SF=…
```

단면 나사:
```
G33 X… I…
G33 X… I… SF=…
```

테이퍼 나사:
```
G33 X… Z… K…
G33 X… Z… K… SF=…
G33 X… Z… I…
G33 X… Z… I… SF=…
```

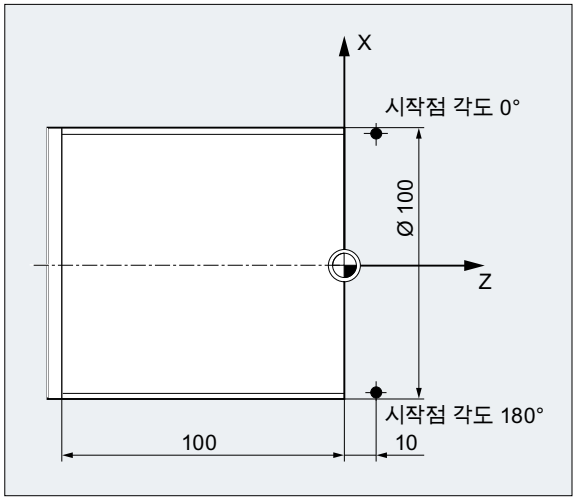
의미

G33:	연속 리드를 사용한 나사 절삭 명령어	
X... Y... Z...:	직교 좌표의 종점	
I...:	X 측 방향 나사 리드	
J...:	Y 측 방향 나사 리드	
K...:	Z 측 방향 나사 리드	
Z:	세로 축	
X:	가로 축	
Z... K...:	원통형 나사의 나사 길이와 리드	
X... I...:	단면 나사의 나사 직경과 나사 리드	
I... 또는 K...:	테이퍼 나사의 나사 리드 지정 (I... 또는 K...) 은 테이퍼 각도를 기준합니다:	
	< 45°:	나사 리드는 K...로 지정됩니다 (세로 방향의 나사 리드).
	> 45°:	나사 리드는 I..로 지정합니다 (가로 방향 나사 리드).
	= 45°:	나사 리드는 I... 또는 K...로 명시될 수 있습니다.

SF=... :	시작점 옵셋 (다중 나사에만 필요) 시작점 옵셋은 절대 각도 위치로 명시됩니다.	
	값 범위:	0.0000 to 359.999 도

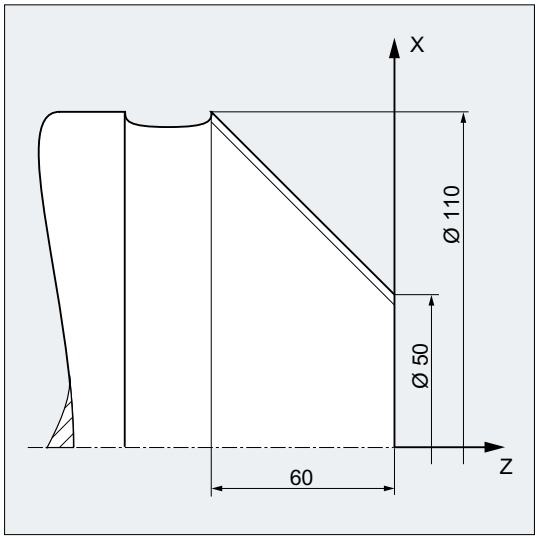
예

예 1: 180° 시작점 옵셋의 이중 원통형 나사



프로그램 코드	코멘트
N10 G1 G54 X99 Z10 S500 F100 M3	; 워크 옵셋, 시작점 접근, 스핀들 가동.
N20 G33 Z-100 K4	; 실린더 나사: z 방향 종점.
N30 G0 X102	; 시작 위치로 후퇴.
N40 G0 Z10	
N50 G1 X99	
N60 G33 Z-100 K4 SF=180	; 2차 절삭: 시작점 옵셋 180°.
N70 G0 X110	; 공구 후퇴.
N80 G0 Z10	
N90 M30	; 프로그램 종료

예 2: 45°보다 작은 각도의 테이퍼 나사

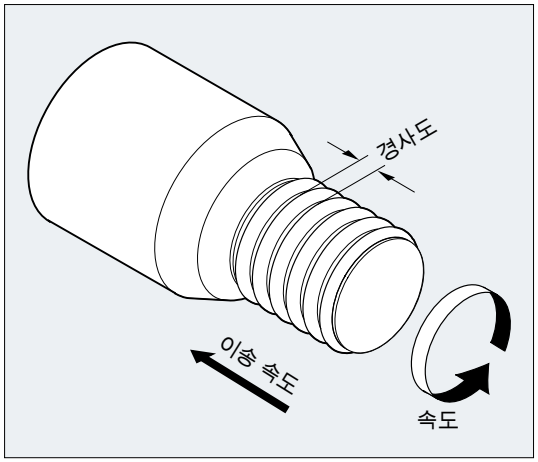


프로그램 코드	코멘트
N10 G1 X50 Z0 S500 F100 M3	; 시작점 접근, 스핀들 활성.
N20 G33 X110 Z-60 K4	; 테이퍼 나사: X와 Z에 종점, (각도가 45° 미만이므로) Z 방향의 K...로 나사 리드 지정
N30 G0 Z0 M30	; 프로그램 끝까지 후퇴.

추가 정보

G33의 나사 절삭 이송 속도

프로그래밍된 스핀들 속도와 나사 리드로부터 제어장치는 선삭 공구가 세로 축 그리고/또는 가로 축 방향으로 나사 길이 전체를 이송하는데 필요한 이송 속도를 계산합니다. 이송 속도F는 G33에 계산되지 않고 제어장치가 최고 축 속도 (급 이송) 에의 제한을 모니터합니다.



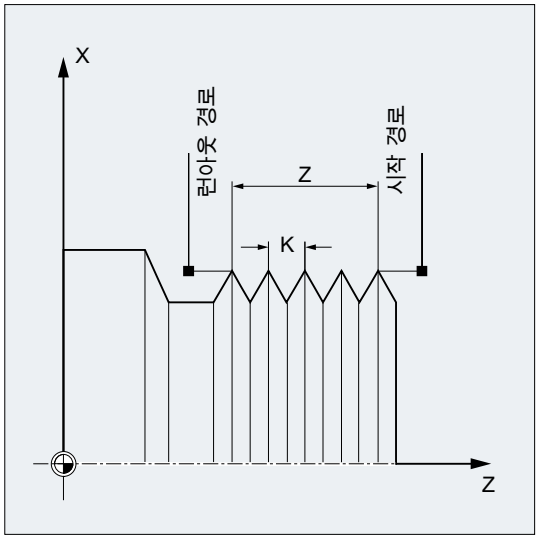
원통형 나사

원통형 나사는 아래로 표시됩니다:

- 나사 길이
- 나사 리드

나사 길이는 절대 또는 증분 치수를 사용하여 직교 좌표 X, Y 또는 Z 중 하나로 입력됩니다 (주로 Z 방향의 선삭 기계) 피드가 가속되거나 또는 감속되는 곳에 대한 런-인 및 런-아웃에 대한 공차 또한 만들어져야 합니다.

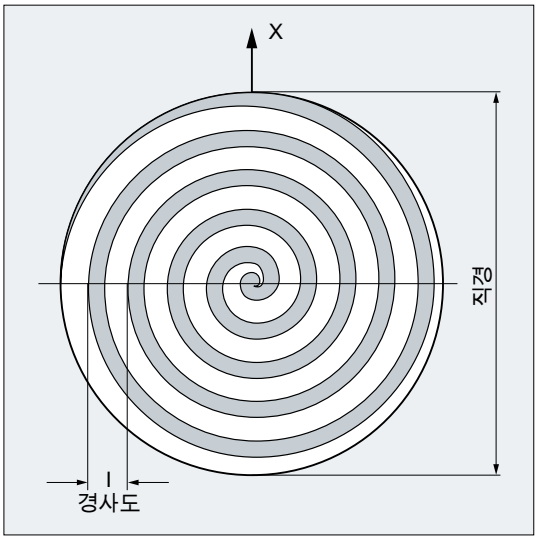
나사 리드는 어드레스 I, J, K에 입력됩니다 (선삭 기계에서는 주로 K를 사용).



단면 나사

단면 나사는 아래로 표시됩니다:

- 나사 직경 (주로 X 방향)
- 나사 리드 (주로 I를 사용)



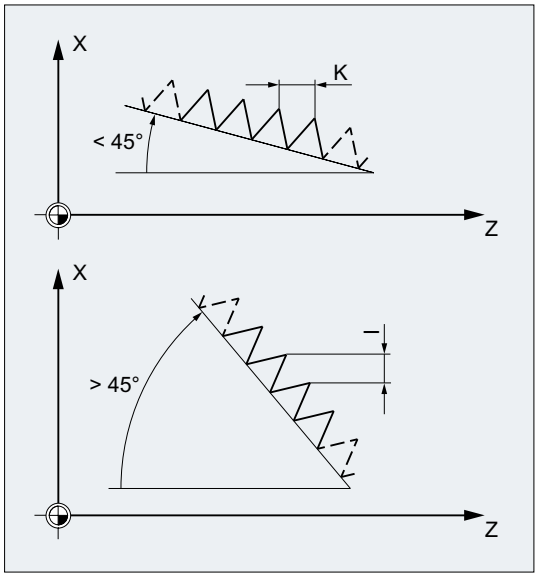
테이퍼 나사

테이퍼 나사는 아래로 표시됩니다:

- 세로 축과 가로 축 방향의 종점 (테이퍼 형상)
- 나사 리드

테이퍼 형상은 선삭 가공을 위해 우선적으로 X와 Z 방향으로 절대 또는 증분 치수로 직교 좌표 X, Y, Z에 입력됩니다. 피드가 가속되거나 또는 감속되는 곳에 대한 런-인 및 런-아웃에 대한 공차 또한 만들어져야 합니다.

나사의 지정은 테이퍼 각도에 따라 달라 집니다 (세로 축과 테이퍼 외부 사이의 각):



3.9.10.2 리드 증가 또는 감소의 나사 절삭(G34, G35)

G34 및 G35 명령을 통해 어드레스 F에서 나사 피치 변경을 프로그래밍하는 옵션을 통해 G33 기능이 확장되었습니다. G34를 사용하면 선형으로 증가하고, G35를 통해 나사 피치가 선형으로 감소합니다. 따라서 명령 G34 와 G35는 자체 태핑 나사의 가공에 사용될 수 있습니다.

구문

피치 증가 원통형 나사:
```
G34 Z… K… F...
```
피치 감소 원통형 나사:
```
G35 Z… K… F...
```
피치 증가 단면 나사:
```
G34 X… I… F...
```
피치 감소 단면 나사:
```
G35 X… I… F...
```
피치 감소 테이퍼 나사:
```
G34 X… Z… K… F...
G34 X… Z… I… F...
```
피치 감소 테이퍼 나사:
```
G35 X… Z… K… F...
G35 X… Z… I… F...
```

의미

G34:	선형 피치 **증가** 나사 절삭 명령
G35:	선형 피치 **감소** 나사 절삭 명령
X... Y... Z...:	직교 좌표의 종점
I...:	X방향의 나사 피치
J...:	Y방향의 나사 피치
K...:	Z방향의 나사 피치

F...:	나사 피치 변경 나사의 시작 및 최종 피치를 이미 알고 있는 경우 다음 공식에 따라 프로그래밍할 나사 피치 변경을 계산할 수 있습니다. $F = \frac{ke2 - ka2}{2 * IG}$ [mm/rev2] 계산식의 각 기호는 다음을 의미합니다.	
	k_e:	최종 피치 나사(축 목표점 좌표의 나사 피치) [mm/rev]
	k_a:	시작 나사 피치(I, J, 또는 K로 프로그래밍) [mm/rev]
	I_G:	나사 길이[mm]

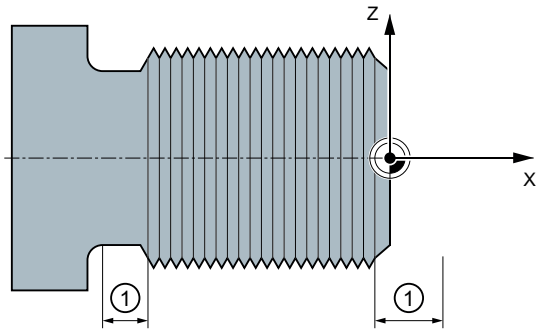
예

프로그램 코드	설명
N1608 M3 S10	; 스핀들 가동.
N1609 G0 G64 Z40 X216	; 시작점으로 접근.
N1610 G33 Z0 K100 SF=R14	; 고정 피치를 적용한 나사 절삭(100 mm/rev)
N1611 G35 Z-200 K100 F17.045455	; 피치 감소: 17.0454 mm/rev2
	블록 종단 피치: 50 mm/rev
N1612 G33 Z-240 K50	; 저크 없이 나사 블록 이송.
N1613 G0 X218	
N1614 G0 Z40	
N1615 M17	

3.9.10.3 G33, G34 및 G35의 프로그램된 런인 및 런아웃 경로(DITS, DITE):

나사의 런인과 런아웃 경로는 DITS와 DITE 주소를 이용하여 가공 프로그램에서 지정할 수 있습니다.

나사 축은 지정 축을 따라 가속되거나 제동됩니다.



① 가공 방향에 따른 런인/런아웃 경로

짧은 런인 경로

나사 런인의 이음고리 때문에 공구 시작 램프를 위한 공간이 거의 없습니다.
따라서 DITS로 런인을 더 짧게 지정해야 합니다.

짧은 런아웃 경로

나사 런아웃에서 나사 어깨 때문에 공구 제동 램프의 공간이 부족할 경우 공작물과 공구 절삭날이 서로 충돌할 위험이 있습니다. 이 경우, DITE로 감속 램프를 더 짧게 지정할 수 있습니다. 그러나 기계 시스템의 관성으로 인해 여전히 충돌이 발생할 수 있습니다.

해결책: 짧은 나사를 프로그래밍하고 스핀들 속도를 낮춤.

참고

DITE는 나사 끝에서 라운딩 공차로 사용됩니다. 이렇게 하면 축 이동에서 매끄러운 변경이 이루어집니다.

효과

프로그램된 런인과 런아웃 경로는 경로에서의 가속 속도만 높입니다. 두 경로 중 하나가 현재 가속의 나사 축 요구사항보다 크게 설정되면 나사 축이 최고 가속으로 가속하거나 감속합니다.

구문

```
DITS=<Value> DITE=<Value>
```

의미

DITS:	나사 런인 경로 지정
DITE:	나사 런아웃 경로 지정
<value>:	위치가 아닌 경로만 DITS 및 DITE를 통해 프로그래밍됩니다. 프로그램된 런인/런아웃 경로는 현재 치수 설정에 따라 처리됩니다(인치, 미터).

예

프로그램 코드	코멘트
...	
N40 G90 G0 Z100 X10 SOFT M3 S500	
N50 G33 Z50 K5 SF=180 DITS=1 DITE=3	; Z=53을 사용해 스무딩 시작.
N60 G0 X20	

추가 정보

SD42010 $SC_THREAD_RAMP_DISP

DITS 또는 DITE를 포함하는 블록이 메인 실행에 삽입되면 프로그램된 런인/런아웃 경로가 설정 데이터 SD42010 $SC_THREAD_RAMP_DISP로 전동됩니다:

- SD42010 $SC_THREAD_RAMP_DISP[**0**] = 프로그램된 DITS 값
- SD42010 $SC_THREAD_RAMP_DISP[**1**] = 프로그램된 DITE 값

런인/런아웃 경로가 이전에 또는 첫 번째 나사 블록에 프로그래밍되지 않은 경우, 설정 데이터의 현재 값이 사용됩니다.

채널/모드 그룹/프로그램 종료 리셋 이후 동작

DITS 또는 DITE가 덮어 쓴 SD 42010 값은 채널/모드 그룹/프로그램 종료 리셋 이후에도 여전히 유효합니다.

온간 시동 후 동작

온간 시동의 경우, 설정 데이터는 DITS 또는 DITE로 덮어 쓰기(표준 동작) 전에 유효했던 값으로 재설정됩니다.

하지만 온간 재시동 후 DITS 및 DITE로 프로그램된 값도 유효하려면, 설정 데이터 SD42010 $SC_THREAD_RAMP_DISP가 가공 데이터 MD10710 $MN_PROG_SD_RESET_SAVE_TAB에 반드시 포함되어 있어야 합니다.

MD10710 $MN_PROG_SD_RESET_SAVE_TAB[<n>] = 42010

런인 또는 런아웃 경로가 매우 짧은 경우 동작

런인 그리고/또는 런아웃 경로가 너무 짧다면 나사 축의 가속은 환경 설정된 값보다 높아집니다. 이 때문에 축에 가속 부하가 발생합니다.

이때 나사 런인에 대해 알람 22280 "프로그램된 런인 경로가 너무 짧습니다"가 발생합니다 (MD11411 $MN_ENABLE_ALARM_MASK에 따른 설정). 단지 알람만 발생되며, 가공 프로그램 실행에는 영향을 주지 않습니다.

3.9.10.4 나사 절삭 중 고속 후퇴 (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN)

다음과 같은 상황에서 '나사 절삭 중 고속 후퇴(G33)' 기능을 사용하여 심각한 손상 없이 나사 절삭을 중단할 수 있습니다.

- DB21 ... DBX7.3를 통해 시작된 NC 정지
- 알람을 통해 묵시적으로 시작된 NC 정지
- 고속 입력 전환 (페이지 612)

후퇴 이동은 다음을 통해 프로그래밍할 수 있습니다.

- 후퇴 경로 및 후퇴 방향 (상대적)
- 후퇴 위치 (절대적)

참고

NC 정지 신호

다음 NC 정지 신호는 나사 절삭 중 고속 후퇴를 트리거하지 않습니다.

- DB21, ... DBX3.4 (축 및 스핀들에서 NC 정지)
- DB21, ... DBX7.2 (블록 한계에서 NC 정지)

태핑

'고속 후퇴' 기능은 **태핑** (G331/G332) 에는 사용할 수 **없습니다**.

구문

고속 후퇴 인에이블, 후퇴 경로 및 후퇴 방향을 따라 후퇴:

```
G33 ... LFON DILF=<value> LFTXT/LFWP ALF=<value>
```

고속 후퇴 인에이블, 후퇴 위치에 따라 후퇴:

```
POLF[<axis identifier>]=<value> LFPOS
POLFMASK/POLFMLIN(<axis 1 name>,<axis 2 name>, etc.)
G33 ... LFON
```

나사 절삭 중 고속 후퇴 디스에이블:

LFOF

의미

LFON:	나사 절삭 중 고속 후퇴 (G33) 인에이블	
LFOF:	나사 절삭 중 고속 후퇴 (G33) 디스에이블	
DILF= :	후퇴 경로의 길이 지정	
	MD 구성 (MD21200 $MC_LIFTFAST_DIST) 과정에서 사전에 설정된 값은 DILF를 프로그래밍하여 가공 프로그램에서 수정할 수 있습니다. **참고:** 구성된 MD 값은 항상 NC 리셋 이후에 활성화됩니다.	
LFTXT LFWP:	후퇴 방향은 G 명령 LFTXT 및 LFWP를 ALF와 연계하여 제어할 수 있습니다.	
	LFTXT:	후퇴 이동이 실행되는 평면을 경로 접선과 공구 방향 (기본 설정) 을 기준으로 계산합니다.
	LFWP:	후퇴 이동이 실행되는 평면은 활성 작업 평면입니다.
ALF= :	후퇴 평면에서 ALF를 사용해 각도 증분치로 방향을 프로그래밍합니다.	
	LFTXT를 사용할 때는 ALF=1로 공구 후퇴 방향을 지정합니다. LFWP의 경우 가공 평면의 방향은 다음 할당에서 도출됩니다. • G17 (X/Y 평면) ALF=1 ; X 방향으로 후퇴 ALF=3 ; Y 방향으로 후퇴 • G18 (Z/X 평면) ALF=1 ; Z 방향으로 후퇴 ALF=3 ; X 방향으로 후퇴 • G19 (Y/Z 평면) ALF=1 ; Y 방향으로 후퇴 ALF=3 ; Z 방향으로 후퇴	
	ALF를 이용한 **프로그래밍 옵션에 대해 자세히 알아보려면** '형상에서 고속 후퇴 시 이송 방향 (페이지 615)'장을 참고하십시오.	
LFPOS:	POLFMASK 또는 POLFMLIN로 선언된 축을 POLF로 프로그램된 절대 축 위치로 후퇴시킵니다.	
POLFMASK:	축을 절대 위치까지 개별적으로 후퇴하도록 축들 (<axis 1 name>,<axis 1 name> 등) 을 분리합니다.	

POLFMLIN:	직선 관계에 있는 절대 위치로 후퇴하도록 축을 분리합니다. **참고:** 관련된 모든 축의 역학적 관계에 따라 도구가 인양 위치에 도달하기 전에 직선 관계가 형성되지 못할 수도 있습니다.	
POLF[]:	인덱스에서 기하 축 또는 기계 축의 절대 후퇴 위치를 지정합니다.	
	적용:	모달
	=<value>:	기하 축의 경우 지정된 값은 공작물 좌표계의 위치로 해석됩니다. 기계 축의 경우 지정된 값은 기계 좌표계의 위치로 해석됩니다. 또한 지정된 값을 증분 치수로 프로그래밍할 수도 있습니다. =IC<value>
<axis identifier>:	기하 축 또는 기계 축 이름	

참고

LFON 또는 LFOF는 언제라도 프로그래밍할 수 있으나 나사 절삭 (G33) 에만 사용됩니다.

참고

POLFMASK/POLFMLIN를 사용한 POLF는 나사 절삭 적용에 제한을 받지 않습니다.

예

예 1: 나사 절삭 중 고속 후퇴 인에이블

프로그램 코드	설명
N55 M3 S500 G90 G18	; 가공 평면 활성화
...	; 시작 위치에 접근
N65 MSG ("나사 절삭")	; 공구 절입
MM_THREAD:	
N67 $AC_LIFTFAST=0	; 나사 가공 시작 전에 리셋.
N68 G0 Z5	
N68 X10	

프로그램 코드	설명
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=7	; 나사 절삭 중 고속 후퇴 인에이블. 후퇴 경로 : 10mm 후퇴 평면: Z/X (G18에 의해) 후퇴 방향: -X (ALF=3: 후퇴 방향 +X)
N71 G33 Z55 X15	
N72 G1	; 나사 절삭 선택 취소.
N69 IF $AC_LIFTFAST GOTOB MM_THREAD	; 나사 절삭이 중단된 경우.
N90 MSG ("")	
...	
N70 M30	

예 2: 태핑 전에 고속 후퇴 해제.

프로그램 코드	설명
N55 M3 S500 G90 G0 X0 Z0	
...	
N87 MSG ("태핑")	
N88 LFOF	; 태핑 전 고속 후퇴 비활성화.
N89 CYCLE...	; G33으로의 태핑.
N90 MSG ("")	
...	
N99 M30	

예 3: 절대 후퇴 위치까지 고속 후퇴

최대 속도로 POLF[X]위치로 이동하거나 정지하는 경우 X의 경로 보간이 이루어지지 않습니다. 다른 축의 모션은 이전과 마찬가지로 프로그래밍된 형상이나 나사 리드 및 스핀들 속도에 의해 결정됩니다.

프로그램 코드	설명
N10 G0 G90 X200 Z0 S200 M3	
N20 G0 G90 X170	
N22 POLF[X]=210 LFPOS	
N23 POLFMASK(X)	; X 축 고속 후퇴 활성화 (인에이블).
N25 G33 X100 I10 LFON	
N30 X135 Z-45 K10	
N40 X155 Z-128 K10	
N50 X145 Z-168 K10	
N55 X210 I10	
N60 G0 Z0 LFOF	
N70 POLFMASK()	; 전체 축 도피 비활성화.

프로그램 코드	설명
M30	

3.9.10.5 볼록 나사 (G335, G336)

G 명령 G335 및 G336을 사용하여 볼록 나사(= 원통형과 반대)를 선삭할 수 있습니다. 자체 무게 때문에 기계 안으로 늘어지는 초대형 콤포넌트 가공에 적용됩니다. 패럭시얼 나사는 초대형 콤포넌트 가운데에 사용하기에는 너무 작습니다. 이 경우 볼록 나사로 보정할 수 있습니다.

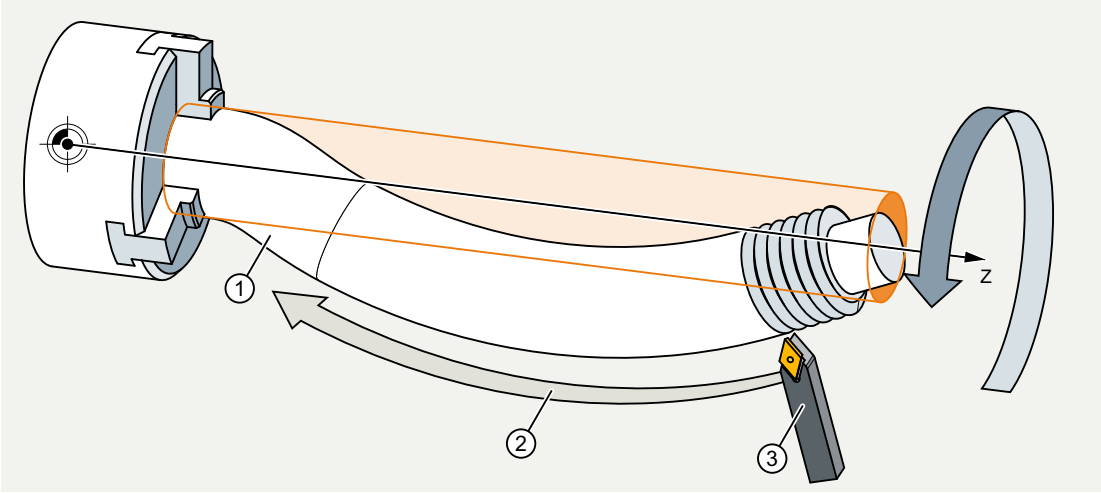


그림 3-6 볼록 나사 선삭

프로그래밍

볼록 나사 선삭은 G335 또는 G336으로 프로그래밍합니다.

G335:	**CW 방향**의 원호 공구 경로에서 볼록 나사 선삭
G336:	**CCW 방향**의 원호 공구 경로에서 볼록 나사 선삭

먼저 파라미터 I, J 및 K로 축 블록 종점 및 피치를 지정하여 직선 나사를 프로그래밍합니다 ("연속 리드를 사용하여 나사 절삭(G33, SF) (페이지 245)" 참조).

원호도 지정합니다. G2/G3의 경우 중심점, 반경, 틈 각도 또는 중간점을 지정하여 프로그래밍할 수 있습니다 ("원호 보간 (페이지 209)" 참조). 중심점을 프로그래밍하여 볼록 나사를 프로그래밍하는 경우 다음 사항을 고려해야 합니다. I, J 및 K는 나사 절삭에서 피치에 사용되

기 때문에 중심점을 프로그래밍할 때 원호 파라미터는 IR=..., JR=... 및 KR=...로 프로그래밍해야 합니다.

IR=...:	X 방향 원호 중심점의 직교 좌표
JR=...:	Y 방향 원호 중심점의 직교 좌표
KR=...:	Z 방향 원호 중심점의 직교 좌표

참고

IR, JR 및 KR은 머신 데이터 (MD10651 $MN_IPO_PARAM_THREAD_NAME_TAB) 를 통해 설정할 수 있는 볼록 나사에 대한 보간 파라미터 이름의 디폴트 값입니다.

장비 제조업체의 사양에서 디폴트 값과의 편차를 계산해야 합니다.

또는 시작점 옵셋 SF를 지정할 수도 있습니다 ("연속 리드를 사용하여 나사 절삭(G33, SF) (페이지 245)" 참조).

구문

볼록 나사 프로그래밍 구문의 기본 형식은 다음과 같습니다.

```
G335/G336 <축 목표점 좌표> <피치> <원호> [<시작점 옵셋>]
```

예

예 1: 종점 및 중심점 프로그래밍으로 CW 방향 볼록 나사 선삭

프로그램 코드	코멘트
N5 G0 G18 X50 Z50	; 시작점으로 접근.
N10 **G335 Z100** K=3.5 **KR=25 IR=-20** SF=90	; CW 방향으로 볼록 나사를 선삭합니다.

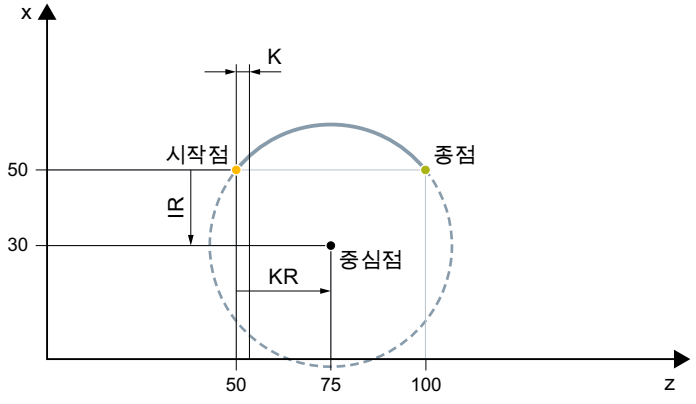


그림 3-7 종점 및 중심점 프로그래밍으로 CW 방향 볼록 나사 선삭

예 2: 종점 및 중심점 프로그래밍으로 CCW 방향 볼록 나사 선삭

프로그램 코드	코멘트
N5 G0 G18 X50 Z50	; 시작점으로 접근.
N10 **G336 Z100** K=3.5 **KR=25 IR=20** SF=90	; CCW 방향으로 볼록 나사를 선삭합니다.

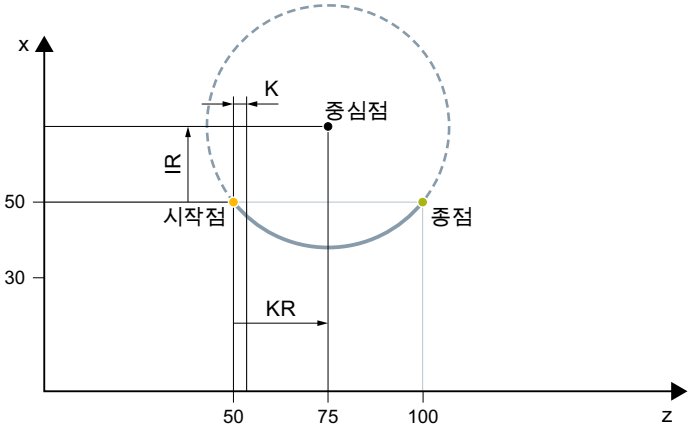


그림 3-8 종점 및 중심점 프로그래밍으로 CCW 방향 볼록 나사 선삭

예 3: 종점 및 반경 프로그래밍으로 CW 방향 볼록 나사 선삭

프로그램 코드
N5 G0 G18 X50 Z50
N10 **G335 Z100** K=3.5 **CR=32** SF=90

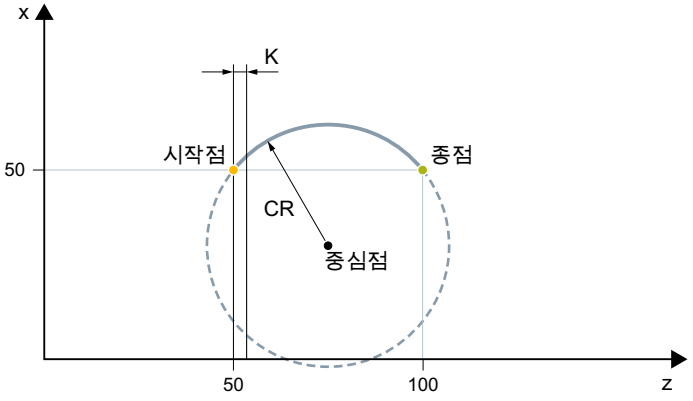


그림 3-9 종점 및 반경 프로그래밍으로 CW 방향 볼록 나사 선삭

예 4: 종점 및 틈 각도 프로그래밍으로 CW 방향 볼록 나사 선삭

프로그램 코드

```
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 AR=102.75 SF=90
```

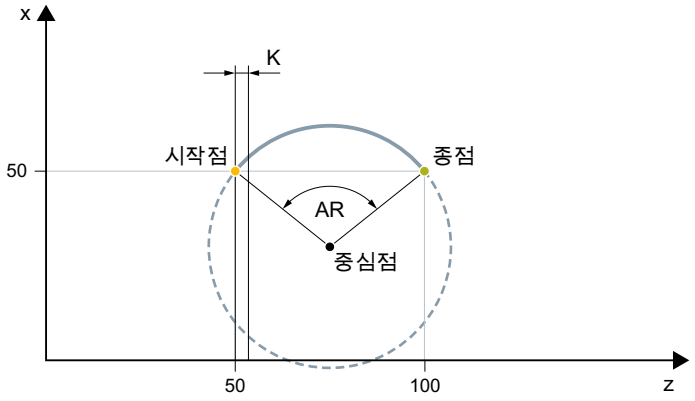


그림 3-10 종점 및 틈 각도 프로그래밍으로 CW 방향 볼록 나사 선삭

예 5: 중심점 및 틈 각도 프로그래밍으로 CW 방향 볼록 나사 선삭

프로그램 코드

```
N5 G0 G18 X50 Z50
N10 G335 K=3.5 KR=25 IR=-20 AR=102.75 SF=90
```

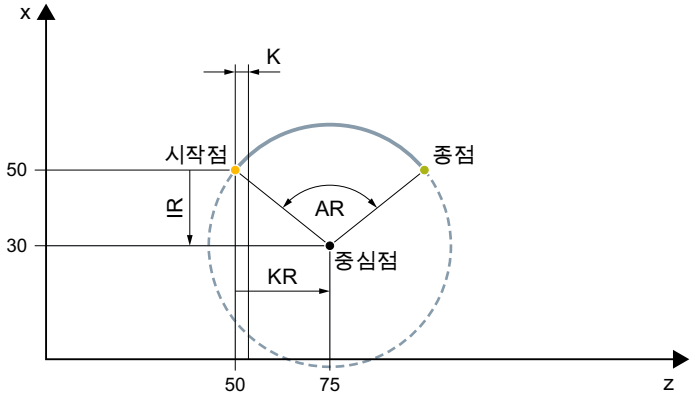


그림 3-11 중심점 및 틈 각도 프로그래밍으로 CW 방향 볼록 나사 선삭

예 6: 종점 및 중간점 프로그래밍으로 CW 방향 볼록 나사 선삭

프로그램 코드

```
N5 G0 G18 X50 Z50
```

프로그램 코드
N10 **G335 Z100** K=3.5 **I1=60 K1=64**

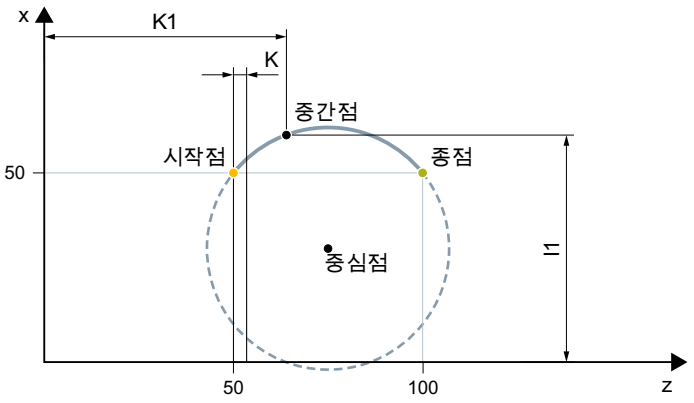
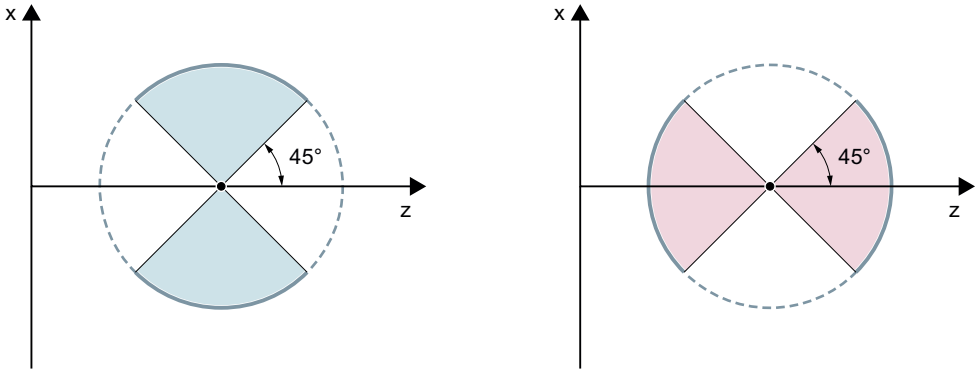


그림 3-12 종점 및 중간점 프로그래밍으로 CW 방향 볼록 나사 선삭

추가 정보

허용 원호 영역

G335/G336에 프로그램된 원호는 지정된 나사 메인 축 (I, J 또는 K) 이 전체 원호에서 메인 축이 원호를 공유하는 영역 내에 있어야 합니다.



Z 축에 허용 가능한 영역 (K로 프로그램된 피치) **X** 축에 허용 가능한 영역 (I로 프로그램된 피치)

다음 그림과 같이 나사 메인 축을 변경하는 것은 허용되지 **않습니다**.

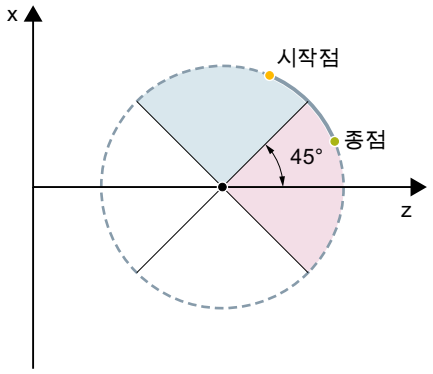


그림 3-13 볼록 나사: 허용되지 않는 영역

프레임

G335 및 G336으로 프레임을 활성화시킬 수도 있습니다. 하지만 허용 원호 영역은 베이직 좌표계 (BCS) 에서 유지되도록 해야 합니다.

원호 경로 프로그래밍의 보충 조건

G2/G3을 이용한 원호 경로 프로그래밍에 설명된 보충 조건이 G335/G336을 이용한 원호 경로 프로그래밍에 적용됩니다 ("원호 보간 (페이지 209)" 참조).

3.9.11 보정 척을 사용하지 않는 태핑

3.9.11.1 보정 척 및 후퇴 모션을 사용하지 않는 태핑(G331, G332)

보정 척을 사용하지 않는 태핑의 경우, G331 및 G332명령어를 사용하여 다음 이송 모션이 실행됩니다.

- G331: 나사 점의 끝부분까지 가는 태핑 방향에서의 태핑
- G332: 회전 역전의 자동 스핀들 방향을 갖는 태핑 블록 G331 까지의 후퇴 모션

구문

```
G331 <axis> <나사 피치>
G331 <axis> <나사 피치> S...
G332 <axis> <나사 피치>
```

의미

G331:	태핑 태핑된 구멍은 축의 이송 모션(드릴링 깊이) 및 나사 피치에 의해 정의됩니다.	
	효율성:	모달
G332:	태핑 시의 후퇴 모션 후퇴 모션은 태핑 시와 같은 피치를 반드시 가지고 있어야 합니다 (G331). 스핀들 회전 방향이 자동으로 바뀝니다.	
	효율성:	모달
<axis>:	나사 끝부분에 있는 기하 축(X, Y 또는 Z)의 이송 거리/위치, 예 Z50	
<나사 피치>:	나사 피치 I (X), J (Y) 또는 K (Z): • 양의 피치: 오른쪽 나사, 예 K1.25 • 음의 피치: 왼쪽 나사, 예 K-1.25	
	값 범위:	±0.001 to ±2000.00mm/회전
S...:	스핀들 속도 마지막 활성 스핀들 속도는 스핀들 속도가 명시되지 않은 경우 사용됩니다.	

참고

두 번째 기어단계 데이터 블록

첫 번째 기어단계 블록에서 속도 전환 임계치의 영향을 받지 않는 축 관련 머신 데이터를 도출할 수 있습니다. 이 데이터에 구성 가능한 전환 임계치 (최대 속도 및 최저 속도) 2개에 관한 두 번째 기어단계 데이터 블록을 사전에 설정하면 스핀들 속도와 모터 토크를 효과적으로 적용하고 가속 속도를 높일 수 있습니다. 머신 제조사의 명시 사항은 반드시 준수해야 합니다.

자세히 알아보려면 '축 및 스핀들' 기능 매뉴얼을 참고하십시오.

예

- 예: G331 / G332를 사용한 태핑 (페이지 267)
- 예: 현 기어 단계의 프로그램된 천공 속도 출력 (페이지 267)
- 예: 두 번째 기어단계 데이터 블록 응용 사례 (페이지 268)
- 예: 속도는 프로그램 되지 않았고, 기어박스 단계는 모니터링 됩니다. (페이지 268)

- 예: 기어박스 단계는 변경할 수 없고, 기어박스 단계 모니터링 됩니다. (페이지 268)
- 예: SPOS를 사용하지 않는 프로그래밍 (페이지 269)

3.9.11.2 예: G331 / G332를 사용한 태핑

프로그램 코드	코멘트
N10 SPOS[n]=0	; 스핀들: 위치 제어 모드
	; 시작 위치 0도
N20 G0 X0 Y0 Z2	; 축: 시작 위치로 접근
N30 G331 Z-50 K-4 S200	; Z에서의 태핑,
	; 피치 K-4 음성 =>
	; 스핀들 회전 방향: CCW 회전,
	; 스핀들 속도 200rpm
N40 G332 Z3 K-4	; Z에서의 후퇴 모션,
	; 피치 K-4 음성(시계 반대 방향),
	; 회전 역전의 자동 방향 =>
	; 시계 방향 스핀들 회전 방향
N50 G1 F1000 X100 Y100 Z100 S300 M3	; 스핀들 작동 내 스핀들

3.9.11.3 예: 현 기어 단계의 프로그램된 천공 속도 출력

프로그램 코드	코멘트
N05 M40 S500	; 프로그램 된 스핀들 속도: 500rpm =>
	; 기어박스 단계 1(20~1028rpm)
...	
N55 SPOS=0	; 스핀들 위치
N60 G331 Z-10 K5 S800	; 태핑
	; 스핀들 속도 800rpm => 기어박스 단계 1

M40로 프로그래밍된 스핀들 속도 S500의 적절한 기어단계는 첫 번째 기어단계 데이터 블록을 기준으로 결정됩니다. 프로그래밍된 드릴링 속도 S800이 현재 기어단계에서 출력되며 필요한 경우 기어단계의 최고 속도로 제한됩니다. SPOS실행 후 자동 기어 단계 변경은 불가능합니다. 기어단계를 자동으로 변경하려면 스핀들이 반드시 속도 제어 모드에 있어야 합니다.

참고

스핀들 속도 800rpm에서 기어박스 단계 2가 선택된 경우, 두 번째 기어단계 데이터 블록의 관련 머신 데이터에 최고 및 최저 속도의 전환 임계치가 반드시 구성되어야 합니다(아래 예제 참조).

3.9.11.4 예: 두 번째 기어단계 데이터 블록 응용 사례

최고 및 최저 속도에 대한 두 번째 기어단계 데이터 블록의 전환 임계치는 G331/G332와 활성 상태 마스터 스핀들의 S값 프로그래밍에 반영됩니다. 자동 M40 기어 단계 변경은 반드시 활성화되어야 합니다. 기어 단계는 상기 설명한 방법으로 결정되기 때문에 활성 기어 단계와 비교됩니다. 다른 점이 발견된다면 기어박스 단계가 변경됩니다.

프로그램 코드	코멘트
N05 M40 S500	; 프로그램 된 스핀들 속도: 500 rpm
...	
N50 G331 S800	; 마스터 스핀들: 기어박스 단계 2가 선택됩니다.
N55 SPOS=0	; 스핀들 위치
N60 G331 Z-10 K5	; 태핑
	; 두 번째 기어박스 단계 데이터 블록 2로부터의 스핀들 가속

3.9.11.5 예: 속도는 프로그램 되지 않았고, 기어박스 단계는 모니터링 됩니다.

G331 과 함께 두 번째 기어단계 데이터 블록을 사용할 때 속도가 프로그래밍 되지 않은 경우, 프로그래밍 된 마지막 속도는 나사 생산에 사용되게 됩니다. 기어단계는 변하지 않습니다. 그러나 이 경우 최종 프로그래밍된 속도가 사용 중인 기어단계의 사전 설정된 속도 범위 (최고 및 최저 속도 임계치로 지정) 안에 있는지 확인하기 위해 모니터링이 수행됩니다. 그렇지 않으면, 경고 16748이 출력됩니다.

프로그램 코드	코멘트
N05 M40 S800	; 프로그램 된 스핀들 속도: 800 rpm
...	
N55 SPOS=0	; 스핀들 위치
N60 G331 Z-10 K5	; 태핑
	; 스핀들 속도 모니터링, 800 rpm
	; 기어박스 단계 1가 활성 상태임.
	; 기어박스 단계 2는 활성 상태여야 함 => 알람 16748

3.9.11.6 예: 기어박스 단계는 변경할 수 없고, 기어박스 단계 모니터링 됩니다.

두 번째 기어단계 데이터 블록을 사용할 때 스핀들 속도가 G331 블록에 형상과 함께 프로그래밍된 경우 스핀들 속도가 사전 설정된 활성 기어단계의 속도 범위 (최대 및 최소 속도 임계치로 지정) 를 벗어나면 절입 축 및 스핀들의 경로 모션이 유지되지 않기 때문에 기어단계를 변경할 수 없습니다.

상기 예와 같이 속도와 기어박스 단계는 G331 블록에서 모니터링되고 필요한 경우 알람 16748이 발생됩니다.

프로그램 코드	코멘트
N05 M40 S500	; 프로그램 된 스핀들 속도: 500rpm =>
	; 기어박스 단계 1
...	
N55 SPOS=0	; 스핀들 위치
N60 G331 Z-10 K5 S800	; 태핑
	; 기어박스 단계는 변경될 수 없습니다,
	; 스핀들 속도 모니터링, 800 rpm
	; 기어박스 단계 데이터 세트 1과 함께: 기어박스 단계 2
	; 활성 상태여야 함 => 알람 16748

3.9.11.7 예: SPOS를 사용하지 않는 프로그래밍

프로그램 코드	코멘트
N05 M40 S500	; 프로그램 된 스핀들 속도: 500rpm =>
	; 기어박스 단계 1(20~1028rpm)
...	
N50 G331 S800	; 마스터 스핀들: 기어박스 단계 2가 선택됩니다.
N60 G331 Z-10 K5	; 태핑
	; 두 번째 기어박스 단계 데이터 블록 2로부터의 스핀들 가속

스핀들의 나사 보간이 현재 위치에서 시작됩니다. 현재 위치는 예를 들어 기어단계가 변경된 경우처럼 이전에 처리된 가공 프로그램 섹션에 따라 결정됩니다. 그러므로 나사 재가공이 불가능할 수도 있습니다.

참고

여러 개의 스핀들을 이용하여 가공하는 경우에는 공구 스핀들도 마스터 스핀들로 지정해야 한다는 점을 유의하십시오. SETMS (<스핀들 번호>) 를 프로그래밍하여 드릴 스핀들을 마스터 스핀들로 설정할 수 있습니다.

3.9.12 태핑 척을 사용한 태핑

3.9.12.1 보정 척 및 후퇴 모션을 갖는 태핑(G63)

보정 척이 있는 태핑의 경우, G63 및 명령어를 사용하여 다음 이송 모션이 실행됩니다.

- G63: 나사 점의 끝부분까지 가는 태핑 방향에서의 태핑
- G63: 회전 역전의 프로그램 된 스핀들 방향을 갖는 후퇴 모션

참고

G63블록 이후, 마지막으로 유효한 보간 유형 G0, G1, G2 는 활성화 됩니다.

구문

```
G63 <축> <회전 방향> <속도 <이송 속도>
```

의미

G63:	태핑 척을 사용한 태핑	
	적용:	넌모달
<축>:	나사 끝부분에 있는 기하 축(X, Y 또는 Z)의 이송 거리/위치, 예 Z50	
<회전 방향>:	스핀들 회전 방향: • M3: 시계 방향 회전, 오른쪽 나사 • M4: 시계 반대 방향 회전, 왼쪽 나사	
<속도>:	태핑 시의 최대 허용 가능 스핀들 속도, 예 S100	
<이송 속도>:	태핑 축 F의 이송 속도, F = 스핀들 속도 * 나사 피치	

예

M5 나사 태핑:

- 표준에 따른 스핀들 피치: 0.8 mm/rev
- 스핀들 속도 S: 200 rpm
- 이송 속도 F = 200 rpm * 0.8 mm/rev = 160 mm/min.

프로그램 코드	코멘트
N10 G1 X0 Y0 Z2 F1000 S200 M3	; 시작점 접근 ; 스핀들 시계 방향 회전, 200 rpm

프로그램 코드	코멘트
N20 G63 Z-50 F160	; 태핑 척을 사용한 태핑
	; 드릴링 깊이: 절대 Z=50mm
	; 이송 속도: 160 mm/min
N30 G63 Z3 M4	; 후퇴 이동: 절대 Z=3mm
	; 회전 역전 방향
	; 시계 반대 방향 회전을 갖는 스핀들, 200 rpm

3.9.13 모따기, 라운딩(CHF, CHR, RND, RNDM, FRC, FRCM)

사용 중인 작업 평면의 형상 코너를 라운딩 또는 모따기로 제작할 수 있습니다.

최적의 표면 품질을 위해 모따기/라운딩 이송 속도를 별도로 프로그래밍할 수 있습니다. 이송 속도를 지정하지 않은 경우 표준 경로 이송 속도 F가 적용됩니다.

"모달 라운딩" 코드로 복수의 형상 코너를 동일한 방법으로 하나씩 라운딩할 수 있습니다.

구문

형상 코너 모따기:

```
G... X... Z... CHR/CHF=<value> FRC/FRCM=<value>
G... X... Z...
```

형상 코너 라운딩:

```
G... X... Z... RND=<value> FRC=<value>
G... X... Z...
```

모달 라운딩:

```
G... X... Z... RNDM=<value> FRCM=<value>
...
RNDM=0
```

참고

모따기/라운딩과 관련한 기능 (이송 속도, 이송 속도 유형, M 명령어 등) 은 머신 데이터 MD20201 $MC_CHFRND_MODE_MASK (모따기/라운딩 특성) 의 비트 0의 설정에 따라 이전 또는 다음 블록에서 도출됩니다. 권장하는 설정은 이전 블록에서 도출된 설정 (비트 0 = 1) 입니다.

의미

CHF=… :	형상 코너 모따기	
	<value>:	모따기의 길이 (G70/G71과 일치하는 단위)
CHR=… :	형상 코너 모따기	
	<value>:	기존 모션 방향을 기준으로 한 모따기의 너비 (G70/G71과 일치하는 단위)
RND=… :	형상 코너 라운딩	
	<value>:	라운딩의 반경 (G70/G71과 일치하는 단위)
RNDM=… :	모달 라운딩 (복수의 형상 코너를 동일한 방법으로 하나씩 라운딩)	
	<value>:	라운딩 반경 (G70/G71과 일치하는 단위)
		RNDM=0로 모달 라운딩 비활성화
FRC=… :	모따기/라운딩에 사용되는 넌모달 이송 속도	
	<value>:	mm/min (G94를 사용할 때) 또는 mm/rev (G95를 사용할 때) 단위로 나타낸 이송 속도
FRCM=… :	모따기/라운딩의 모달 이송 속도	
	<value>:	mm/min (G94를 사용할 때) 또는 mm/rev (G95를 사용할 때) 단위로 나타낸 이송 속도
		FRCM=0은 모따기/라운딩의 모달 이송 속도를 비활성화하고 F에 프로그래밍된 이송 속도를 활성화함

참고

모따기/라운딩이 너무 높음

모따기 (CHF/CHR) 또는 라운딩 (RND/RNDM) 에 프로그래밍된 값이 관련 형상 요소에 비해 너무 크면 모따기 또는 라운딩은 자동으로 조정됩니다.

1. MD11411 $MN_ENABLE_ALARM_MASK 비트 4가 설정된 경우, 알람 10833 "모따기 또는 라운딩을 줄여야 합니다"가 출력됩니다 (알람 취소).
2. 모따기/라운딩이 형상 코너에 맞을 때까지 감소됩니다. 그 결과, 최소 한 블록을 모션 없이 진행합니다. 이 블록에서는 필요한 모션이 중지됩니다.

참고

모따기/라운딩 불가능

다음의 경우, 모따기/라운딩이 수행되지 않습니다.

- 평면에 직선 또는 원호 형상이 없는 경우
- 평면 외부에서 동작이 행해지는 경우
- 평면이 변경된 경우
- 이송 정보가 전혀 없는 머신 데이터 (예: 명령만 출력하는 머신 데이터) 에 지정된 블록 개수를 초과한 경우

참고

FRC/FRCM

G0를 사용해 모따기를 이송하면 FRC/FRCM가 적용되지 않습니다. 따라서 에러 메시지 없이 F 값에 따라 명령어를 프로그래밍할 수 있습니다.

모따기/라운딩이 블록에 프로그래밍되거나 RNDM이 활성화된 경우에만 FRC가 적용됩니다.

FRC는 현재 블록의 F 또는 FRCM 값을 덮어씁니다.

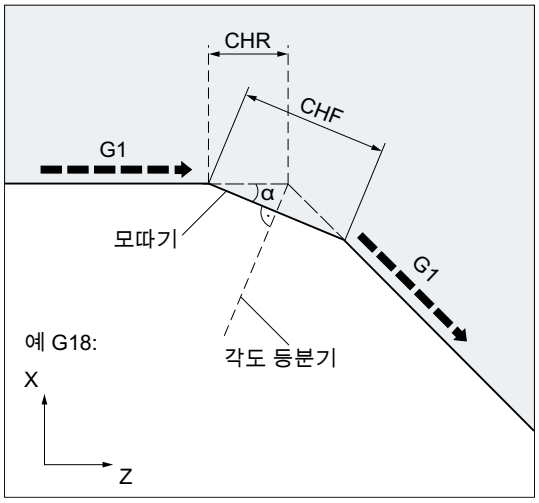
FRC에 프로그래밍된 이송 속도는 반드시 0보다 커야 합니다.

FRCM=0를 사용해 모따기/라운딩에서 F로 프로그래밍된 이송 속도를 활성화합니다.

FRCM을 프로그래밍한 경우 G94나 G95로 변경할 때 F 값을 처리하는 방법처럼 FRCM 값도 다시 프로그래밍해야 합니다. F만 다시 프로그래밍한 상태에서 변경 이전의 이송 속도 유형 FRCM이 0보다 큰 경우 에러 메시지가 출력됩니다.

예

예 1: 두 개 직선 사이 모따기



- MD20201 Bit 0=1 (이전 블록에서 도출).
- G71이 활성 상태입니다.
- 모션 방향에서 모따기의 너비 (CHR) 는 2mm 이고 이송 속도는100mm/min입니다.

다음 두 가지 방법으로 프로그래밍할 수 있습니다.

- CHR로 프로그래밍

프로그램 코드

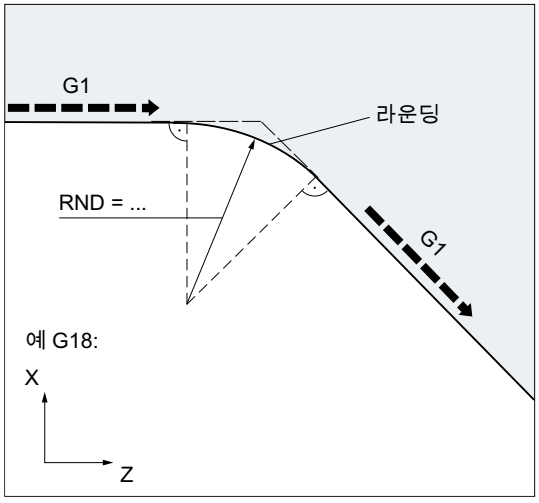
```
...
N30 G1 Z… CHR=2 FRC=100
N40 G1 X…
...
```

- CHF로 프로그래밍

프로그램 코드

```
...
N30 G1 Z… CHF=2(cosα*2) FRC=100
N40 G1 X…
...
```

예 2: 두 개 직선 사이 라운딩



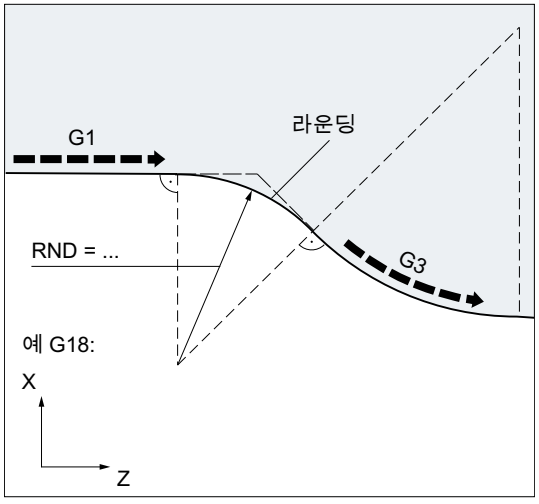
- MD20201 Bit 0=1 (이전 블록에서 도출).
- G71이 활성 상태입니다.
- 라운딩 반경은 2mm이고 이송 속도는 50mm/min입니다.

프로그램 코드

```
...
N30 G1 Z… RND=2 FRC=50
N40 G1 X…
...
```

예 3: 직선과 원호 사이 라운딩

RND 코드를 사용하면 접선 연결을 통해 임의로 조합된 직선 형상과 원호 형상 사이에 원호 형상 요소를 삽입할 수 있습니다.



- MD20201 Bit 0=1 (이전 블록에서 도출).
- G71이 활성 상태입니다.
- 라운딩 반경은 2mm이고 이송 속도는 50mm/min입니다.

프로그램 코드
...
N30 G1 Z… RND=2 FRC=50
N40 G3 X… Z… I… K…
...

예 4: 날카로운 공작물 모서리 연마를 위한 모달 라운딩

프로그램 코드	코멘트
...	
N30 G1 X… Z… RNDM=2 FRCM=50	; 모달 라운딩 활성화.
	라운딩의 반경: 2mm
	라운딩의 이송 속도: 50mm/min
N40...	
N120 RNDM=0	; 모달 라운딩 비활성화.
...	

예제 5: 다음 블록 또는 이전 블록의 기술 적용

- MD20201 Bit 0=0: 다음 블록에서 도출 (디폴트 설정)

프로그램 코드	코멘트
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; 모따기 N20~N30 (F=100 mm/min)
N30 Y10 CHF=4	; 모따기 N30~N40 (FRC=200 mm/min)
N40 X20 CHF=3 FRC=200	; 모따기 N40~N60 (FRCM=50 mm/min)
N50 RNDM=2 FRCM=50	
N60 Y20	; 모달 라운딩 N60~N70 (FRCM=50mm/min)
N70 X30	; 모달 라운딩 N70~N80 (FRCM=50mm/min)
N80 Y30 CHF=3 FRC=100	; 모따기 N80~N90 (FRC=100mm/min)
N90 X40	; 모달 라운딩 N90-N100 (F=100 mm/min) (FRCM 선택 해제)
N100 Y40 FRCM=0	; 모달 라운딩 N100~N120 (G95 FRC=1 mm/rev)
N110 S1000 M3	
N120 X50 G95 F3 FRC=1	
...	
M02	

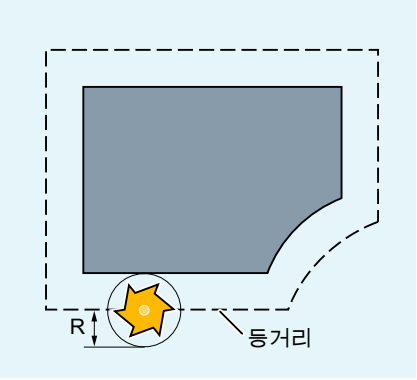
- MD20201 Bit 0 = 1: 이전 블록에서 도출 (권장 설정)

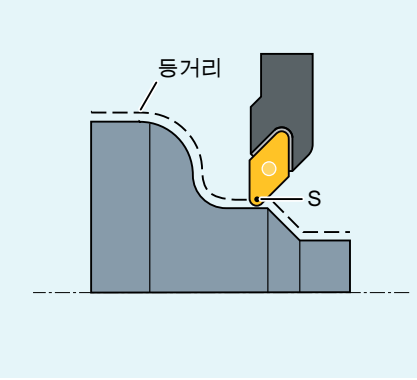
프로그램 코드	코멘트
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; 모따기 N20~N30 (F=100 mm/min)
N30 Y10 CHF=4 FRC=120	; 모따기 N30-N40 (FRC=120 mm/min)
N40 X20 CHF=3 FRC=200	; 모따기 N40~N60 (FRCM=200mm/min)
N50 RNDM=2 FRCM=50	
N60 Y20	; 모달 라운딩 N60~N70 (FRCM=50mm/min)
N70 X30	; 모달 라운딩 N70~N80 (FRCM=50mm/min)
N80 Y30 CHF=3 FRC=100	; 모따기 N80~N90 (FRC=100mm/min)
N90 X40	; 모달 라운딩 N90-N100 (FRCM=50mm/min)
N100 Y40 FRCM=0	; 모달 라운딩 N100-N120 (FRCM=100 mm/min)
N110 S1000 M3	
N120 X50 CHF=4 G95 F3 FRC=1	; 모따기 N120~N130 (G95 FRC=1 mm/rev)
N130 Y50	; 모달 라운딩 N130~N140 (FRCM=3 mm/min)
N140 X60	
...	
M02	

3.10 공구 반경 보정

3.10.1 공구 반경 보정 (G40, G41, G42, OFFN)

공구 반경 보정 (TRC) 이 활성화되면 자동적으로 시스템에서 여러 공구에 대한 등거리 공구 경로를 계산합니다.





구문

G0/G1 X... Y… Z... **G41**/**G42** [**OFFN=<값>**]	
...	
G40 X... Y… Z...	

의미

G41:	가공 방향을 형상의 **왼쪽**으로 하여 TRC를 활성화함.
G42:	가공 방향을 형상의 **오른쪽**으로 하여 TRC를 활성화함.
OFFN=<value>:	프로그래밍된 형상에 대한 허용치 (일반 형상 옵셋): (옵션). 예: 황삭을 위한 등거리 경로 생성
G40:	TRC 비활성화.

참고

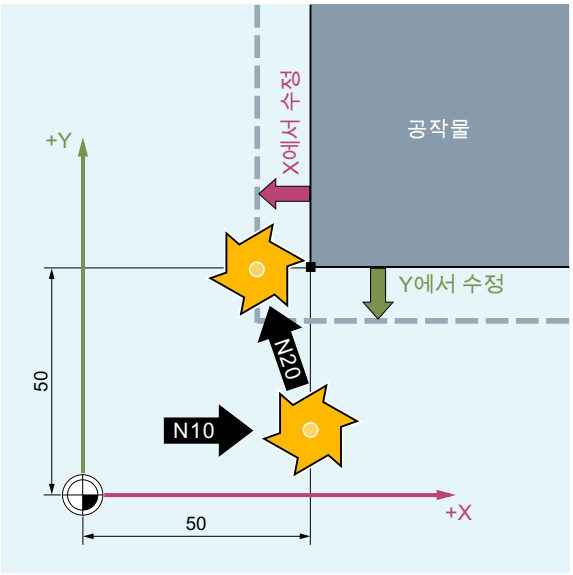
G40/G41/G42를 포함한 NC 블록에서는 G0 또는 G1이 활성 상태여야 하며, 선택한 작업 평면에서 적어도 하나의 축을 지정해야 합니다.

활성 상태에서 하나의 축만 지정하면 두 번째 축의 최종 위치가 자동으로 추가되고 **두** 축과 함께 이송됩니다.

두 축은 반드시 채널에서의 기하 축처럼 활성 상태여야 합니다. GEOAX 프로그래밍을 통해 축을 활성화합니다.

예

예 1: 밀링

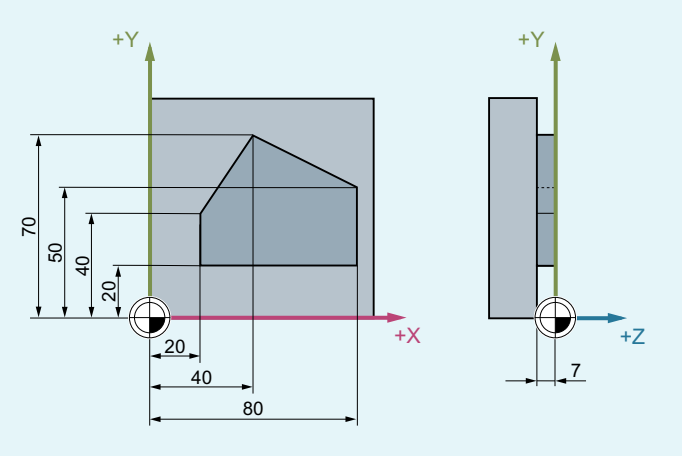


프로그램 코드	설명
N10 G0 X50 T1 D1	; 공구 길이 보정만 활성화됩니다. 보정 없이 X50에 접근합니다.
N20 G1 G41 Y50 F200	; 반경 보정이 활성화되고 보정을 통해 X50/Y50 지점에 접근합니다.
N30 Y100	
…	

예 2: 밀링을 사용한 "통상적인" 절차의 예

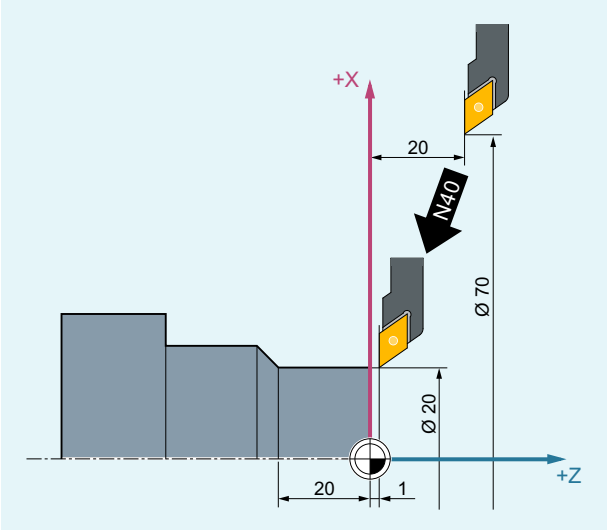
'통상적인' 절차:

1. 공구 호출.
2. 공구 교환
3. 작업 평면 및 공구 반경 보정 활성화



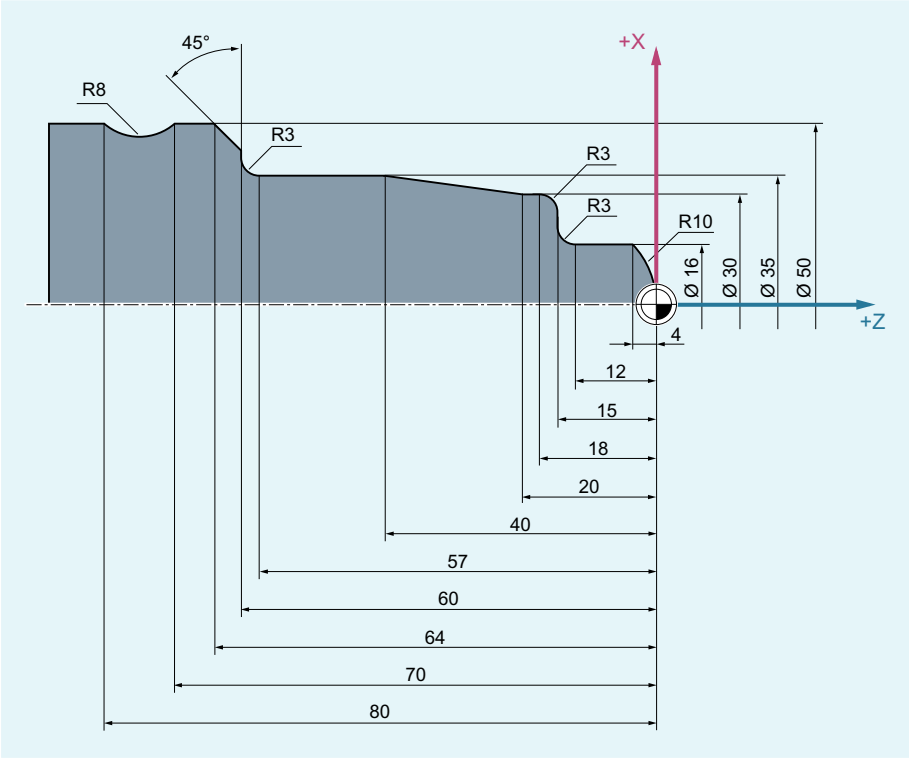
프로그램 코드	설명
N10 G0 Z100	; 공구 교환을 위한 후퇴.
N20 G17 T1 M6	; 공구 교환
N30 G0 X0 Y0 Z1 M3 S300 D1	; 공구 옵셋 값 호출, 길이 보정 선택.
N40 Z-7 F500	; 공구 절입.
N50 G41 X20 Y20	; 공구 반경 보정 활성화, 형상 좌측으로 가공.
N60 Y40	; 형상 밀링.
N70 X40 Y70	
N80 X80 Y50	
N90 Y20	
N100 X20	
N110 G40 G0 Z100 M30	; 프로그램 끝까지 공구 후퇴.

예 3: 선삭



프로그램 코드	설명
…	
N20 T1 D1	; 공구 길이 보정만 활성화됩니다.
N30 G0 X70 Z20	; 보정 없이 X70 Z20에 접근합니다.
N40 G42 X20 Z1	; 반경 보정이 활성화되며 보정을 통해 X20/Z1 지점에 접근합니다.
N50 G1 Z-20 F0.2	
…	

예 4: 선삭



프로그램 코드	설명
N5 G0 G53 X280 Z380 D0	; 시작점.
N10 TRANS X0 Z250	; 워크 옵셋.
N15 LIMS=4000	; 속도 제한 (G96).
N20 G96 S250 M3	; 주속 일정 제어 선택
N25 G90 T1 D1 M8	; 공구와 옵셋을 선택합니다.
N30 G0 G42 X-1.5 Z1	; 공구 반경 보정을 통해 공구를 설정합니다.
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	; 반경 10 회전.
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	; 반경 3 회전.
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	; 반경 3 회전.
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	; 반경 3 회전.
N85 G1 X46	
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	; 공구 반경 보정을 선택 해제하고 공구 교체 위치에 접근합니다.

프로그램 코드	설명
N100 T2 D2	; 공구를 호출하고 옵셋을 선택합니다.
N105 G96 S210 M3	; 일정 정삭 속도를 선택합니다.
N110 G0 G42 X50 Z-60 M8	; 공구 반경 보정을 통해 공구를 설정합니다.
N115 G1 Z-70 F0.12	; 직경 50 회전.
N120 G2 X50 Z-80 I6.245 K-5	; 반경 8 회전.
N125 G0 G40 X100 Z50 M9	; 공구를 후퇴시키고 공구 반경 보정을 선택 해제합니다.
N130 G0 G53 X280 Z380 D0 M5	; 공구 교체 위치에 접근합니다.
N135 M30	; 프로그램 종료.

추가 정보

공구 경로 계산

시스템에서 공구 경로를 계산하기 위해 다음의 정보가 필요합니다:

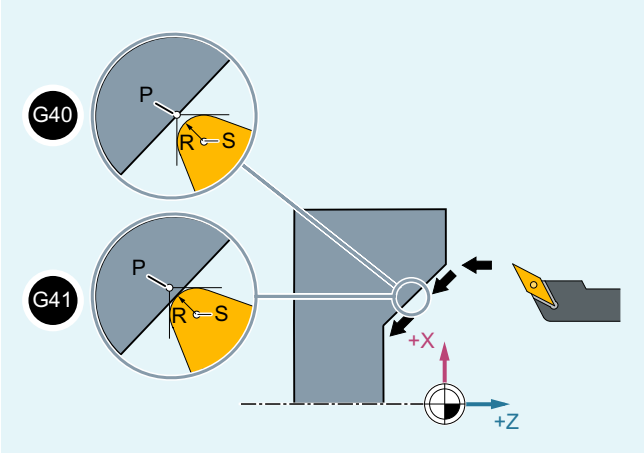
- 공구 번호 (T...), 절삭날 번호 (D...)
- 가공 방향 (G41/G42)
- 작업 평면 (G17/G18/G19)

공구 번호 (T...), 절삭날 번호 (D...)

공구 경로와 공작물 형상 간의 거리는 밀링 커터 반경 또는 절삭날 반경 그리고 절삭날 위치 설정을 통해 계산됩니다.

가공 방향 (G41/G42)

이 정보를 통해 시스템은 공구 경로를 이탈하는 방향을 탐지합니다.

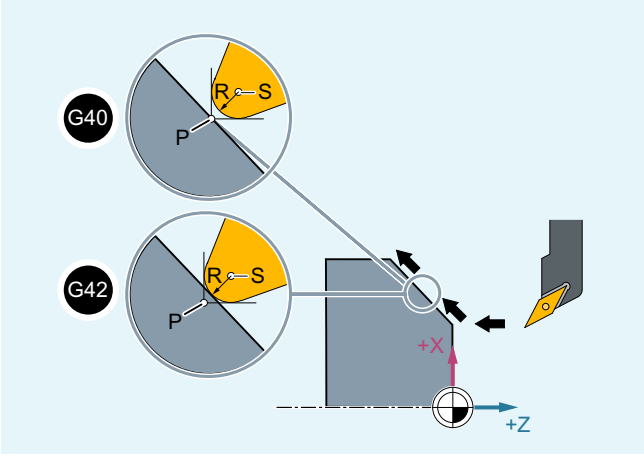


P 공구 팁

S 절삭날 중심점

R 절삭날 반경

그림 3-14 가공 방향이 형상의 왼쪽인 선삭 시 공구 반경 보정(G41)



P 공구 팁

S 절삭날 중심점

R 절삭날 반경

그림 3-15 가공 방향이 형상의 오른쪽인 선삭 시 공구 반경 보정(G42)

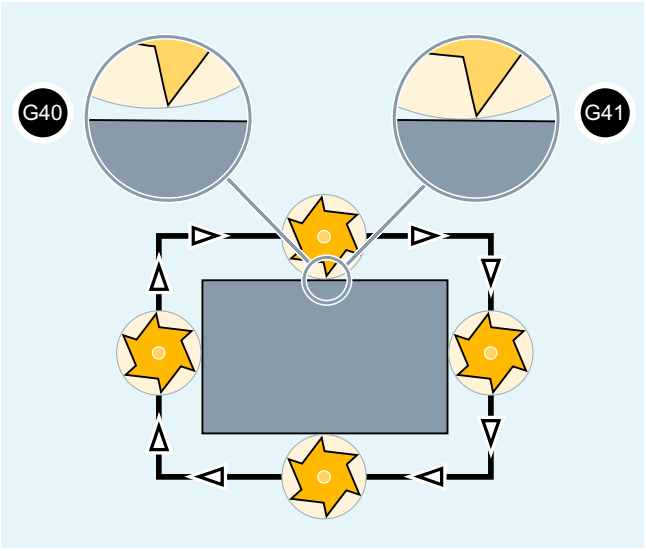


그림 3-16 가공 방향이 형상의 왼쪽인 밀링 시 공구 반경 보정(G41)

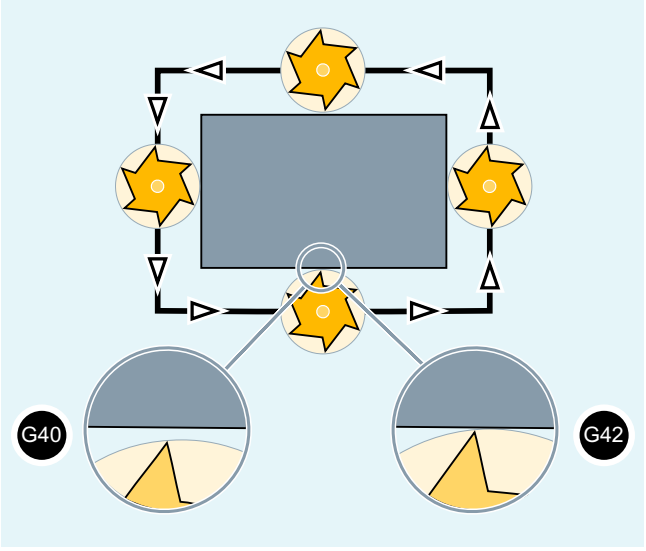


그림 3-17 가공 방향이 형상의 오른쪽인 밀링 시 공구 반경 보정(G42)

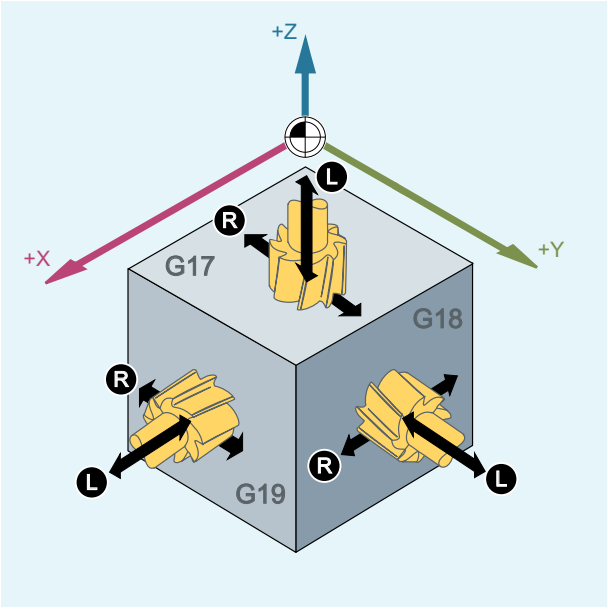
참고

음의 보정 값은 옵셋 측면 변경(G41 ↔ G42) 과 동일한 의미를 가집니다.

작업 평면 (G17/G18/G19)

제어 시스템은 이 정보를 통해 작업 평면을 탐지하여 보정되는 축 방향을 탐지합니다.

예: 밀링 공구



L 길이

R 반경

프로그램 코드	설명
...	
N10 G17 G41 …	; 공구 반경 보정은 X/Y 평면에서 수행되고 공구 길이 보정은 Z 축 방향에서 수행됩니다.
...	

참고

2축 기계에서 공구 반경 보정은 대개 G18을 통해 '실제' 평면에서만 가능합니다.

공구 길이 보정

공구 선택에서 직경 축에 지정된 마모 파라미터는 MD를 이용해 직경 값으로 정의될 수 있습니다. 이 지정은 면이 연속으로 변경될 때 자동으로 변경되지 않습니다. 이렇게 하려면 작업 평면이 변경된 후 해당 공구를 다시 선택해야 합니다.

폐쇄 형상인 경우 공구 반경 보정 동작

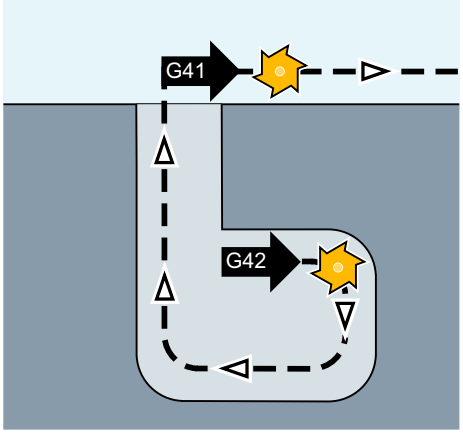
폐쇄 형상인 경우 공구 반경 보정 동작은 설정 날짜를 통해 설정할 수 있습니다:

SD42496 $SC_CUTCOM_CLSD_CONT

값	의미
FALSE	연속적인 2개의 원호 블록 또는 원호 블록 1개와 직선 블록 1개로 구성된 (사실상의) 폐쇄 형상을 옵셋할 때 내부에 2개의 교차점이 생기는 경우, 표준 절차에 따라 첫 번째 부분 형상의 블록 끝에 가장 가깝게 위치한 교차점이 선택됩니다. 첫 블록의 시작점과 두 번째 블록의 종점과의 간격이 유효 보정 반경의 10% 보다 작지만 1000 경로 증분 (소수점 이하 세 자리 수를 포함한 1 mm에 해당) 보다 크지 않다면 형상은 사실상 폐쇄된 것으로 간주됩니다.
TRUE	위에서 설명된 것과 동일한 상황에서 블록 시작에 더 가까운 첫 부분 형상에 위치한 교차점이 선택됩니다.

보정 방향 변경 (G41 ↔ G42)

중간에 G40이 없어도 보정 방향 변경 (G41 ↔ G42) 을 프로그래밍할 수 있습니다.



작업 평면 변경

G41/G42가 활성일 때 작업 평면(G17/G18/G19)을 변경하는 것은 **불가능**합니다.

공구 옵셋 데이터 블록 변경 (D...)

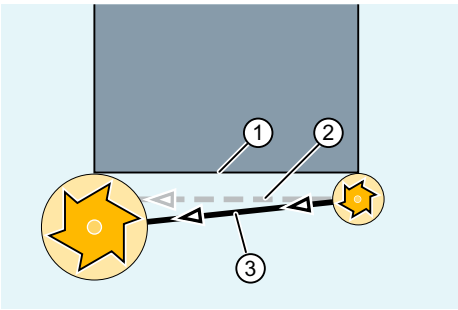
공구 옵셋 데이터 블록은 보정 모드에서 변경할 수 있습니다.

새 D 번호를 포함하는 블록에서와 마찬가지로 변경된 공구 반경은 이미 활성화 상태가 됩니다.

참고

반경 변경 또는 보정 이동은 전체 블록에 걸쳐 수행되고 프로그램된 종점에서 새 등거리에 도달할 뿐입니다.

직선 이동인 경우 공구는 시작점과 종점 사이의 경사진 경로를 따라 이동합니다.



① 프로그래밍된 형상 요소
② 활성 블록의 공구 옵셋 데이터 블록을 변경하지 않은 공구 경로
③ 활성 블록의 공구 옵셋 데이터 블록이 변경되어 그 결과로 공구 반경이 변경된 공구 경로

원호 보간은 나선 이동을 만들어 냅니다.

공구 반경 변경

시스템 변수 등을 사용하여 변경할 수 있습니다. 시퀀스는 공구 옵셋 데이터 블록을 변경하는 시퀀스 (D...) 와 동일합니다.

참고

나중에 T 또는 D를 프로그래밍해야만 수정된 값이 적용됩니다. 변경 내용은 다음 블록까지 적용되지 않습니다.

보정 모드

보정 모드는 특정한 개수의 연속 블록 또는 보정 평면에서 이송 명령이나 위치 데이터를 포함하지 않은 M 명령어를 통해서만 중단시킬 수 있습니다.

참고

연속 중단 블록의 최대 개수 또는 M 명령어는 머신 데이터를 통해 설정할 수 있습니다(장비 제조업체 정보 참조).

참고

경로 거리가 0인 블록 역시 중단으로 간주됩니다!

다음 또한 참고:

형상 접근 및 후퇴 (NORM, KONT, KONTC, KONTT) (페이지 288)

3.10.2 형상 접근 및 후퇴 (NORM, KONT, KONTC, KONTT)

요구 사항

KONTC 및 KONTT 명령은 시스템에서 '폴리노미얼 보간' 옵션이 사용된 경우에만 사용할 수 있습니다.

기능

공구 반경 보정이 활성 상태인 경우 (G41/G42), NORM, KONT, KONTC 또는 KONTT 명령을 사용하여 필요한 형상 특성 또는 소재 형상에서 공구의 접근 및 후퇴 경로를 조정할 수 있습니다.

KONTC 또는 KONTT는 세 축 모두에서 연속성 조건이 준수되도록 합니다. 따라서 옵셋 면에 수직인 경로 구성 요소를 동시에 프로그래밍할 수 있습니다.

구문

G41/G42 **NORM**/**KONT**/**KONTC**/**KONTT** X... Y... Z...	
...	
G40 X... Y... Z...	

의미

NORM:	직선으로의 직접 접근 및 직선에서의 직접 후퇴를 활성화합니다. 공구는 형상 지점과 수직이 되는 방향을 향합니다.
KONT:	프로그래밍된 코너 동작G450 또는 G451에 따라 시작점/종점 주위를 이동하는 접근/후퇴를 활성화합니다.
KONTC:	곡률이 일정한 접근/후퇴를 활성화합니다.
KONTT:	접선이 일정한 접근/후퇴를 활성화합니다.

참고

KONTC 및 KONTT 블록에서 G1 블록 만 오리지널 접근/후퇴 블록으로 사용 가능합니다. 시스템은 이들을 적절한 접근/후퇴 경로의 폴리노미얼로 교체합니다.

보충 조건

KONTT 및 KONTC는 공구 반경 보정의 3D 변수 (CUT3DC, CUT3DCC, CUT3DF) 에서는 사용할 수 없습니다. 만약 위의 명령어들이 프로그램된 경우 시스템에서는 별도의 에러 메시지 없이 내부적으로 NORM으로 전환합니다.

예

KONTC

완전한 원은 원 중심점에서 시작하여 접근됩니다. 접근 블록 종점에서의 커브 반경 및 방향이 그 다음 원의 값과 동일합니다. 접근/후퇴 블록에서 절입은 Z 축 방향으로 이루어집니다. 아래 그림은 공구 경로가 수직으로 투영된 모습입니다.

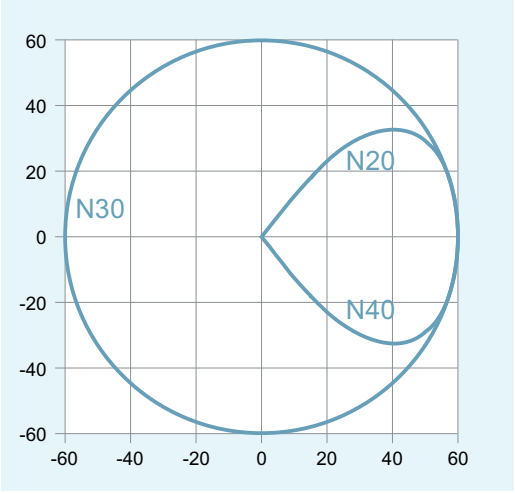


그림 3-18 수직 투영

연관된 NC 프로그램 세그먼트는 다음과 같습니다:

프로그램 코드	코멘트
$TC_DP1[1,1]=121	; 밀링 공구
$TC_DP6[1,1]=10	; 반경 10 mm
N10 G1 X0 Y0 Z60 G64 T1 D1 F10000	
N20 G41 KONTC X70 Y0 Z0	; 접근
N30 G2 I-70	; 완전한 원
N40 G40 G1 X0 Y0 Z60	, 후진
N50 M30	

곡률이 완전한 원의 원호 경로에 맞춰지는 동시에 Z60에서 원 Z0의 평면으로 이송이 이루어집니다.

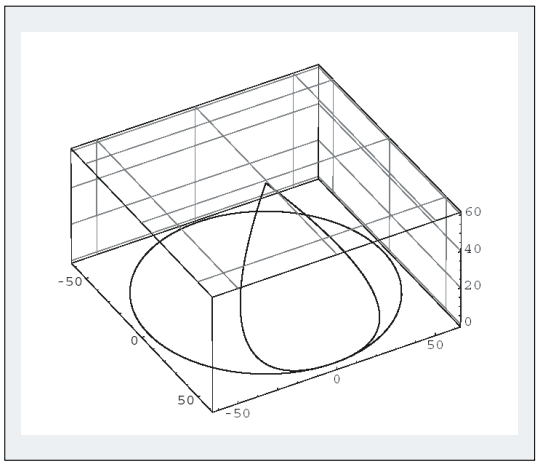
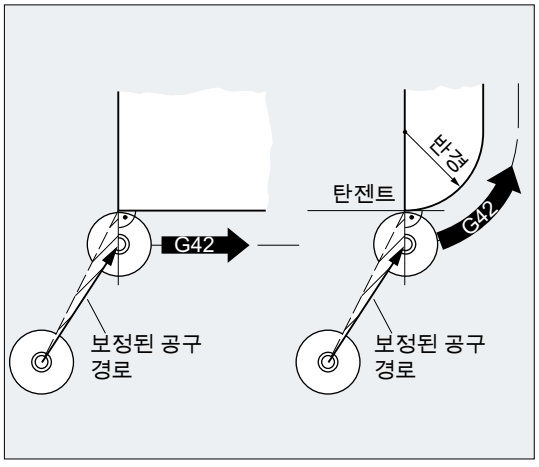


그림 3-19 3D 표시

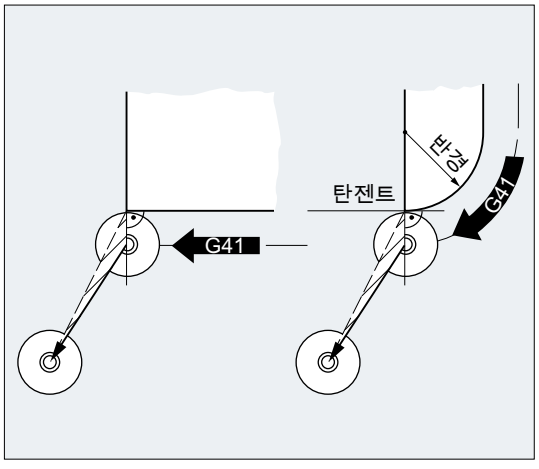
추가 정보

NORM을 이용한 접근/후퇴

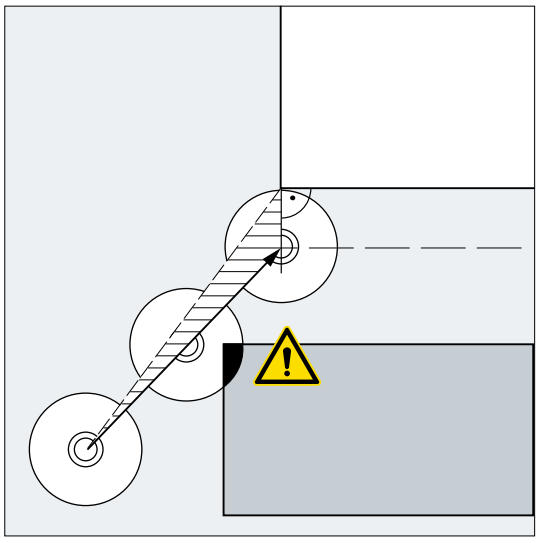
1. 접근:
   NORM을 활성화하면 이송 동작을 위해 프로그램된 사전 설정 접근 각도와 관계 없이 공구는 직선을 따라 보정 시작 위치로 직접 이동하고 시작점의 경로 접선과 수직으로 놓여집니다.



2. 후퇴:
   공구는 마지막으로 보정된 경로 종점과 수직을 이루며 이송 동작을 위해 프로그래밍된 사전 설정 접근 각도와 관계 없이 보정되지 않은 다음 위치 (예: 공구 교환 지점) 로 직선을 따라 직접 이동합니다.



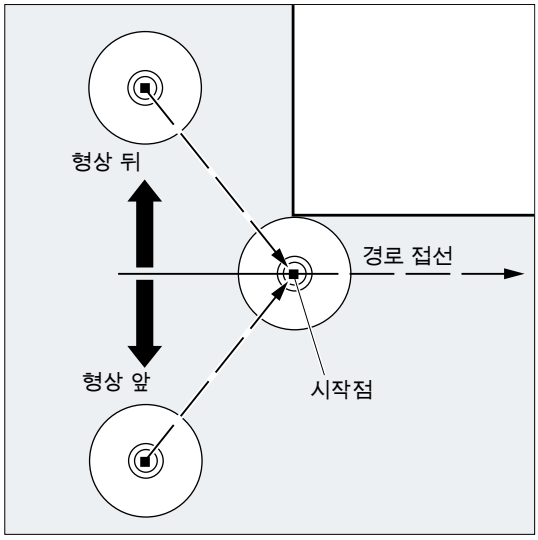
접근/후퇴 각도를 수정하면 충돌 위험이 발생합니다.



유의사항
충돌 위험 충돌 위험을 피하려면 프로그래밍 과정에서 수정된 접근/후퇴 각도를 고려해야 합니다.

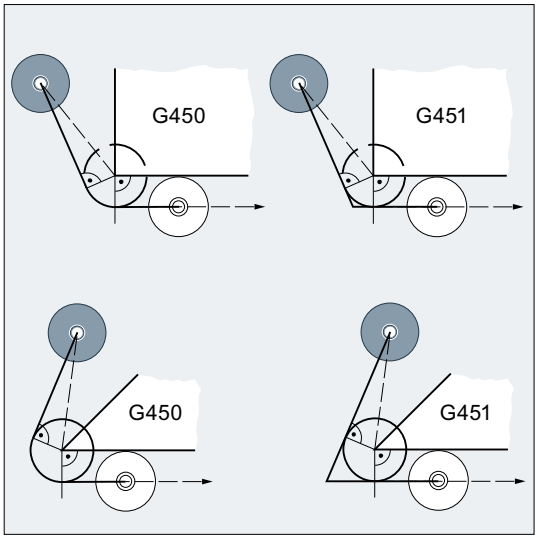
KONT를 이용한 접근/후퇴

접근하기 전에 공구를 형상의 **앞쪽** 또는 **뒤쪽**에 배치할 수 있습니다. 시작점의 경로 접선은 분리선 역할을 합니다.

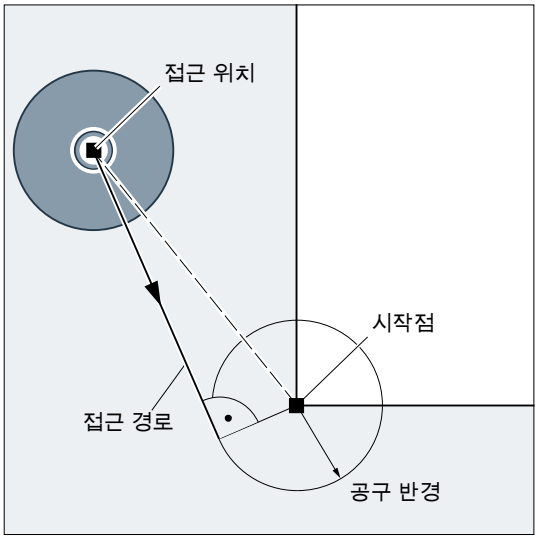


따라서 KONT를 사용한 접근/후퇴에 관련해서는 두 가지 상황을 구별해야 합니다.

1. 공구가 형상의 앞쪽에 있습니다.
   → 접근/후퇴 방식은 NORM의 경우와 동일합니다.
2. 공구가 형상의 뒤쪽에 있습니다.
   – 접근:
   공구는 프로그래밍된 코너 동작 (G450/G451) 에 따라 원호 경로를 따라 또는 등거리 경로의 교차점 위에서 시작점 주위를 이동합니다.
   G450/G451 명령은 현재 블록에서 다음 블록으로의 트렌지션에 적용됩니다.



두 경우 (G450/G451) 모두 다음과 같은 접근 경로가 생성됩니다.



보정되지 않은 접근 지점으로부터 직선이 그려집니다. 이 선은 원 반경이 공구 반경이 되는 원의 접선입니다. 원의 중심점은 시작점에 있게 됩니다.

   – 후퇴:
   후퇴도 접근과 동일한 방식이지만 순서가 반대입니다.

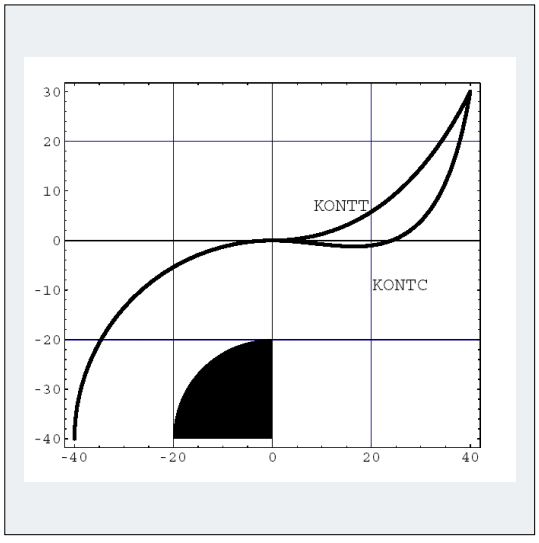
KONTC를 이용한 접근/후퇴

형상 지점은 일정한 커브를 통해 접근/후퇴합니다. 형상 지점에서 가속에의 점프가 없습니다. 시작점에서 형상 지점까지의 경로는 폴리노미널로 보간됩니다.

KONTT를 이용한 접근/후퇴

형상 지점은 일정한 접선으로 접근/후퇴합니다. 형상에서 가속 시 점프가 발생될 수 있습니다. 시작점에서 형상 지점까지의 경로는 폴리노미널로 보간됩니다.

KONTC와 KONTT의 차이점

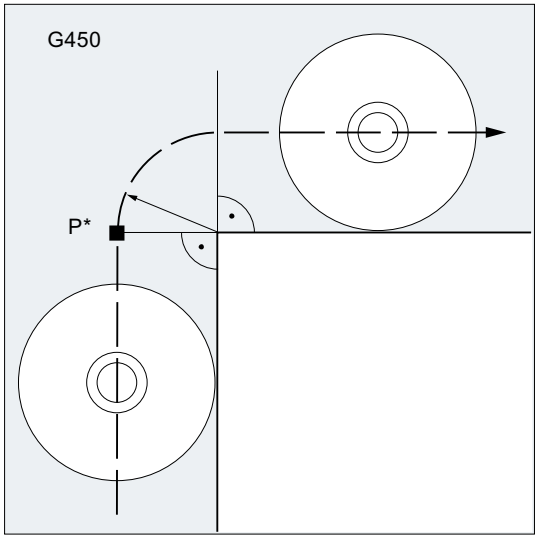


아래 그림은 KONTT와 KONTC 사이의 접근/후퇴 동작의 차이를 나타낸 것입니다. 중심점 X0 Y-40인 20 mm 반경의 원은 20mm 외부 반경의 공구로 보정됩니다. 그러므로 공구 중심점은 40mm 반경의 원호 경로를 따라 이동합니다. 접근 블록의 종점은 X40 Y30에 있게 됩니다. 제로 지점에서 원호 블록 및 후퇴 블록 간의 트렌지션이 이루어집니다. KONTC와 연관된 커브의 확장된 연속성 때문에 후퇴 블록은 먼저 음의 Y 축 방향으로 이동합니다. 이는 때때로 바람직하지 않습니다. 이 반응은 KONTT를 사용한 후퇴 블록에서는 발생하지 않습니다. 하지만, 이 블록으로 인해 블록 트렌지션에서 가속 스텝 변경이 발생합니다.

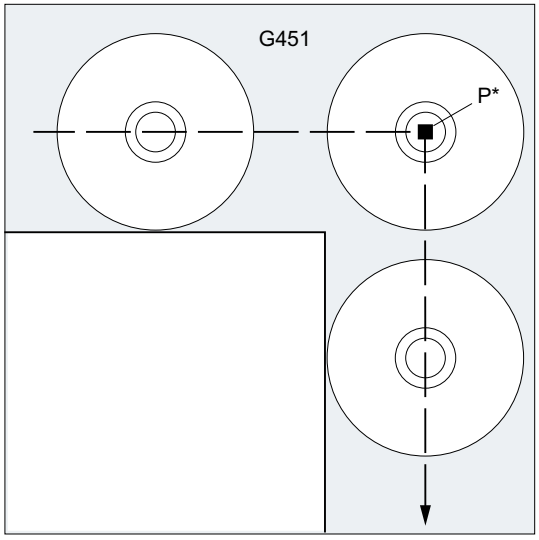
KONTT 또는 KONTC 블록이 후퇴 블록이 아닌 접근 블록인 경우 형상은 정확히 일치하지만 반대 방향으로 가공됩니다.

3.10.3 외부 코너(G450, G451, DISC)에서 보정

공구 반경 보정이 활성화된 상태에서 (G41/G42) G450 또는 G451 명령을 사용하여 외경 코너 주변을 이동할 때 보정된 공구 경로의 코스를 정의할 수 있습니다.



G450을 사용하면 공구 중심점은 공구 반경의 호를 가로지르는 공작물 코너의 주위를 이동합니다.



G451 을 사용하면 공구 중심점은 프로그램된 형상으로부터 공구 반경까지 등거리에 위치하는 두 등거리 교차점에 접근합니다. G451 은 원호 및 직선에만 적용됩니다.

참고

G450/G451을 사용하면 KONT가 활성 상태이고 접근점이 형상 뒤쪽에 있을 때의 접근 경로를 정의할 수 있습니다 ('형상 접근 및 후퇴 (NORM, KONT, KONTC, KONTT) (페이지 288)' 참조).

DISC 명령을 사용하면 G450을 통해 트렌지션 원을 변형시켜 더 예리한 형상 코너를 만들 수 있습니다.

구문

```
G450 [DISC=<value>]

G451
```

의미

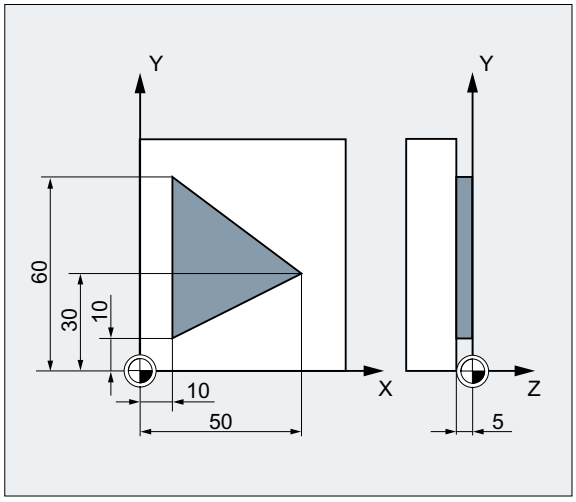
G450:	G450은 원호 경로에서 공작물 코너를 따라 이동하는 데 사용됩니다.			
DISC:	G450을 사용하여 원호 경로를 유연하게 프로그래밍합니다. (옵션)			
	<value>:	유형:	INT	
		값 범위:	0, 1, 2, ... 100	
		의미:	0	트렌지션 원
			100	등거리 경로의 교차 (이론상의 값)
G451:	G451은 공작물 코너의 경우 두 등거리 경로의 교차점에 접근하는 데 사용됩니다. 공구는 공작물 코너에서 뒤로 후퇴합니다.			

참고

DISC는 G450을 호출해야 적용되지만 G450 없이 이전 블록에 프로그래밍할 수 있습니다. 두 명령어는 모달입니다.

예

다음 예제에서 트렌지션 반경은 모든 외부 코너를 위해 프로그래밍됩니다(블록 N30에서의 코너 동작 프로그래밍에 해당). 이는 공구가 방향 변경할 때의 중지와 후퇴를 방지합니다.



프로그램 코드	코멘트
N10 G17 T1 G0 X35 Y0 Z0 F500	; 시작 조건.
N20 G1 Z-5	; 공구 절입.
N30 G41 KONT **G450** X10 Y10	; KONT 접근/후퇴 모드 및 **코너 동작 G450**을 사용하여 TRC **활성화**

프로그램 코드	코멘트
N40 Y60	; 형상 밀링.
N50 X50 Y30	
N60 X10 Y10	
N80 G40 X-20 Y50	; 보정 모드 비활성화, 트렌지션 원호에서 후퇴.
N90 G0 Y100	
N100 X200 M30	

추가 정보

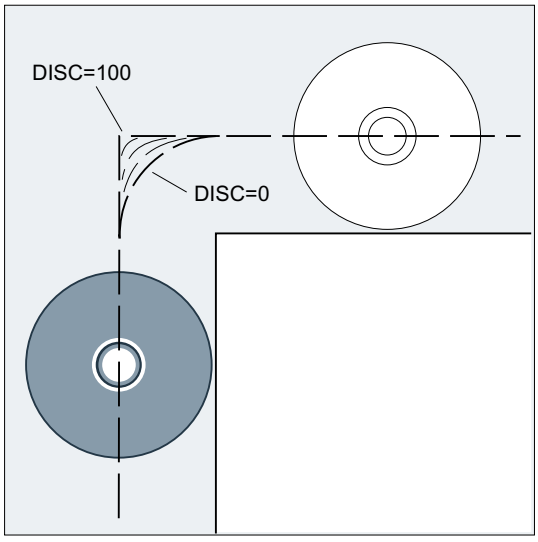
G450/G451

중간점 P*에서, 시스템이 절입 이동 또는 기능 전환 같은 작업을 실행합니다. 이러한 작업은 코너를 형성하는 두 블록 사이에 삽입된 블록에서 프로그래밍됩니다.

G450을 사용하면 트렌지션 원은 데이터와 관련된 그 다음 이송 명령에 귀속됩니다.

DISC

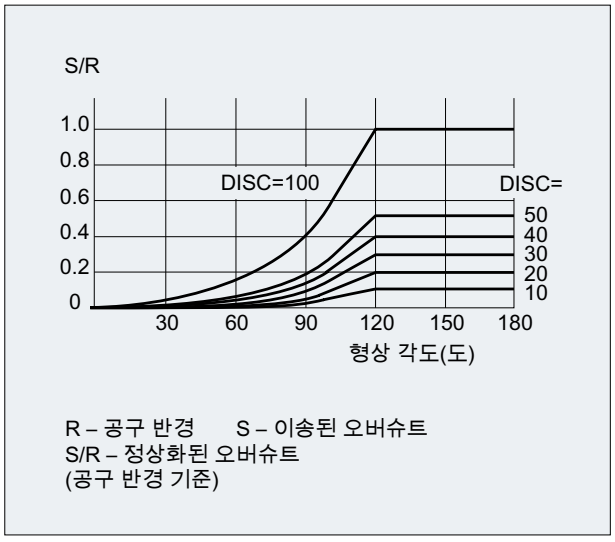
DISC 값이 0보다 크게 지정되면 중간 원은 확대된 높이로 나타나고 그 결과로 트렌지션 타원 또는 포물선이나 쌍곡선이 됩니다.



상한은 일반적으로 DISC=50 으로 머신 데이터에 정의됩니다.

이송 동작

G450을 활성화한 상태에서 형상 각도가 예각이고 DISC 값이 크면 공구는 코너에서 형상으로부터 들어 올려집니다. 형상 각도가 120° 이상인 경우에는 형상 주위로 일정하게 이동합니다.

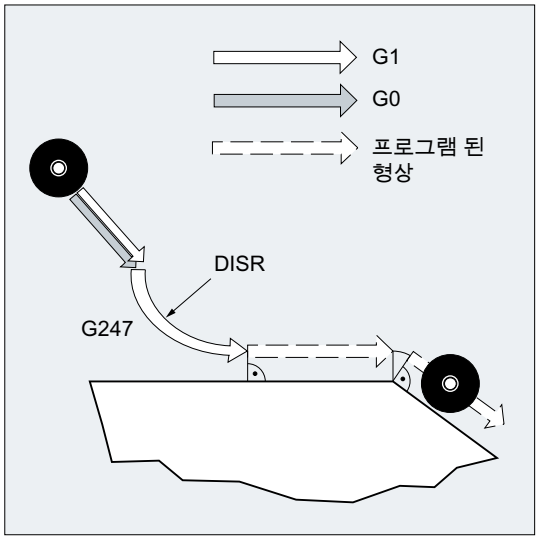


G451을 활성화한 상태에서 형상 각도가 예각이면 공구가 들어올려짐에 따라 잉여 비절삭 공구 경로가 생성됩니다. 이런 경우 파라미터가 트렌지션 원으로의 자동 트렌지션을 정의하기 위해 머신 데이터에서 사용될 수 있습니다.

3.10.4 유연한 접근과 후퇴

3.10.4.1 접근 및 후퇴 (G140~G143, G147, G148, G247, G248, G347, G348, G340, G341, DISR, DISCL, DISRP, FAD, PM, PR)

SAR (유연한 접근 및 후퇴) 기능은 시작점의 위치와는 무관하게 형상의 시작점으로 접선 접근할 수 있게 하기 위해 사용됩니다.



이 기능은 주로 공구 반경 보정과 함께 사용됩니다.

이 기능이 활성화되면 지정된 파라미터에 따라 추종 블록으로 트렌지션 (또는 후퇴 중에 이전 블록에서 트렌지션) 할 수 있도록 제어 시스템이 중간점을 계산합니다.

접근 이동은 최대 4개의 하위 이동으로 구성됩니다. 이동의 시작점은 P_0, 종점은 P_4라고 부릅니다. 이 두 지점 사이에 최대 3개의 중간점 P_1, P_2 및 P_3을 삽입할 수 있습니다. 지점 P_0, P_3 및 P_4는 항상 정의됩니다. 중간 경유 지점 P_1 과 P_2는 정의된 파라미터와 기하학적 조건에 따라 생략될 수 있습니다. 후퇴할 때는 역방향으로 지점을 통과합니다. 즉, P_4에서 시작해 P_0에서 이송을 끝냅니다.

구문

유연한 접근:

- 직선인 경우:
  ```
  G147 G340/G341 ... DISR=..., DISCL=..., DISRP=... FAD=...
  ```
- 4분원/반원인 경우:
  ```
  G247/G347 G340/G341 G140/G141/G142/G143 ... DISR=... DISCL=...
  DISRP=... FAD=...
  ```

유연한 후퇴:

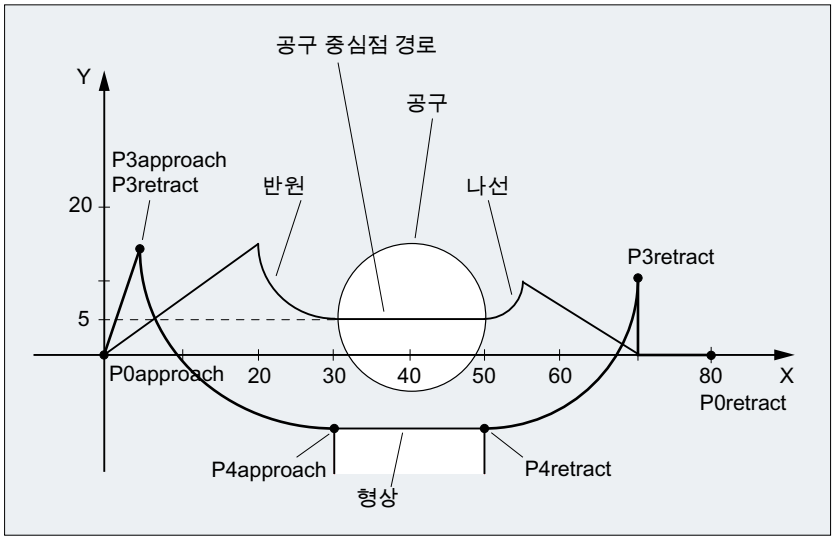
- 직선인 경우:
  ```
  G148 G340/G341 ... DISR=..., DISCL=..., DISRP=... FAD=...
  ```
- 4분원/반원인 경우:
  ```
  G248/G348 G340/G341 G140/G141/G142/G143 ... DISR=... DISCL=...
  DISRP=... FAD=...
  ```

의미

G147:	직선을 따라 접근
G148:	직선을 따라 후퇴
G247:	90°회전을 따라 접근
G248:	90°회전을 따라 후퇴
G347:	180°회전을 따라 접근
G348:	180°회전을 따라 후퇴
G340:	공간에서 접근 및 후퇴 (디폴트 설정)
G341:	평면에서 접근 및 후퇴

G140:	접근 방향과 후퇴 방향은 현재 보정 면에 따라 달라 집니다 (디폴트 설정).
G141:	좌측 접근 및 좌측 후퇴
G142:	우측 접근 및 우측 후퇴
G143:	접근과 후퇴 방향은 접선 방향에 대한 시작 또는 종점의 상대적인 위치에 따라 달라집니다.
DISR=...:	1. 직선을 따라 접근 및 후퇴 (G147/G148): 형상의 시작점에서 절삭날까지의 거리 2. 원을 따라 접근 및 후퇴 (G247, G347/G248, G348): 공구 중심점 경로의 반경 **주의:** 반원을 따라 REPOS하는 경우 DISR이 원호 직경이 됩니다.
DISCL=...:	고속 절입 모션의 경우 가공 평면에서 종점까지의 거리 DISCL=AC(...) 고속 절입 모션을 위해 종점 절대 위치 지정
DISCL=AC(...):	고속 절입 모션을 위해 종점 절대 위치 지정
DISRP:	가공 평면부터 지점 P1 (후퇴 평면) 까지의 거리
DISRP=AC(...):	지점 P1의 절대 위치 지정
FAD=...:	저속 피드 이송 속도 프로그램된 값은 활성 이송 속도 유형 (G그룹 15)에 따라 작용합니다.
FAD=PM(...):	프로그램된 값을 활성 이송 속도 유형과 관계 없이 G94처럼 직선 이송 속도로 해석합니다.
FAD=PR(...):	프로그램된 값을 활성 이송 속도 유형과 관계 없이 G95처럼 회전 당 이송 속도로 해석합니다.

예



- 유연한 접근 (블록 N20 동작)
- 90°회전을 따라 접근 (G247)
- 접근 방향이 프로그램되지 않았음, G140이 적용, 즉 TRC가 동작됩니다 (G41).
- 형상 옵셋 OFFN=5 (N10)
- 현재 공구 반경 = 10, 따라서 적용 보정 반경 TRC = 15, SAR 형상 반경 = 25, 공구 중심 경로의 반경이 DISR=10 과 동일합니다.
- Z 위치만이 N20에 프로그램되기 때문에 원의 종점은 N30에서 구해 집니다.
- 절입 이동
  - 급 이송으로 Z20 에서 Z7 로 이동합니다 (DISCL=AC(7)).
  - 그 후 FAD=200으로 Z0로 이동합니다.
  - X/Y 평면으로 이동 후 다음 블록부터 F1500이 적용됩니다. (다음 블록부터 이 속도가 적용되며 N30에서 동작중인 G0는 G1으로 변경됩니다, 그렇지 않을 경우 G0 속도를 사용하여 가공이 이루어집니다.)
- 유연한 후퇴 (블록 N60 동작)
- 90°회전 (G248) 및 나선형 (G340) 으로 후퇴합니다.
- FAD는 G340에 부적합하지 때문에 프로그램되지 않았습니다.

- 시작점에서 Z=2; DISCL=6 으로 인하여 종점에서 Z=8
- DISR=5 일 때 SAR 형상의 반경 = 20, 공구 중심점 경로의 반경 = 5

Z8 에서 Z20 으로 후퇴한 다음 X/Y 평면에 평행하게 X70 Y0 으로 이동합니다.

프로그램 코드	**코멘트**
`$TC_DP1[1,1]=120`	`; 공구 정의 T1/D1`
`$TC_DP6[1,1]=10`	`, 반경`
`N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN=5`	`; (P0 app)`
`N20 G41 G247 G341 Z0 DISCL=AC(7) DISR=10 F1500 FAD=200`	`; 접근 (P3 app)`
`N30 G1 X30 Y-10`	`; (P4 app)`
`N40 X40 Z2`	
`N50 X50`	`; (P4 ret)`
`N60 G248 G340 X70 Y0 Z20 DISCL=6 DISR=5 G40 F10000`	`; 후퇴 (P3 ret)`
`N70 X80 Y0`	`; (P0 ret)`
`N80 M30`	

추가 정보

접근과 후퇴 형상 선택

접근 및 후퇴 형상은 2번째 G그룹의 적절한 G 코드를 사용하여 선택합니다.

G147:	직선을 따라 접근
G247:	90°회전을 따라 접근
G347:	180°회전을 따라 접근
G148:	직선을 따라 후퇴
G248:	90°회전을 따라 후퇴
G348:	180°회전을 따라 후퇴

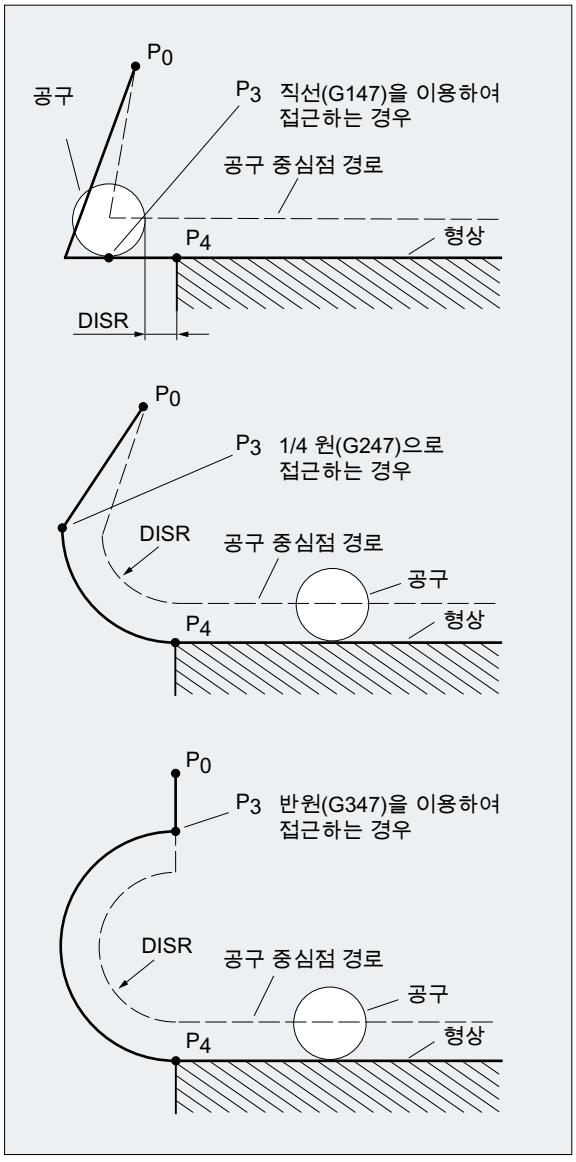


그림 3-20 접근 이동과 동시에 공구 반경 보정 활성화

접근 및 후퇴 방향 선택

공구 반경이 양수인 경우 접근 및 후퇴 방향은 공구 반경 보정 (G140, 디폴트 설정) 을 사용하여 결정합니다.

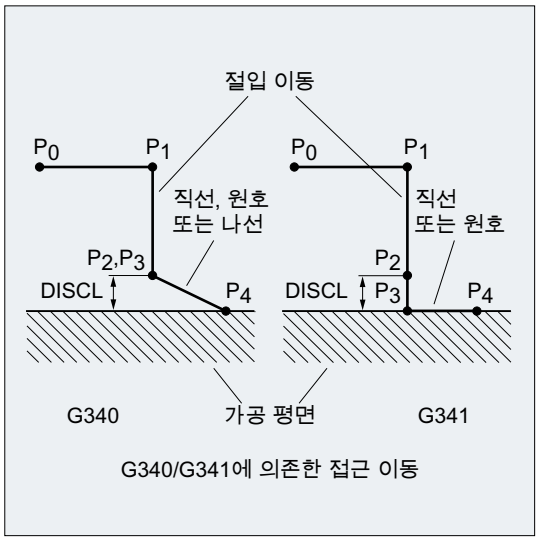
- G41 활성화 →좌측에서 접근
- G42 활성화 →우측에서 접근

G141, G142 및 G143 등의 다른 접근 방식을 사용할 수 있습니다.

해당 G 코드들은 4분원 또는 반원 모드 접근에서만 사용됩니다.

시작점과 종점 사이의 동작 단계 (G340 과 G341).

어떠한 경우든 이동은 1개 이상의 직선과 접근 형상을 결정하는 G 코드에 따라 추가 직선 1개, 4분원 1개 또는 반원 1개로 구성됩니다. 두 가지 경로 분할 방법은 아래 그림에 설명되어 있습니다.



G340/G341에 의존한 접근 이동

G340:	지점 P_0에서 지점 P_1까지 직선으로 접근. 파라미터 DISRP가 프로그래밍되지 않은 경우 이 직선은 가공 평면과 평행합니다. 지점 P_1에서 지점 P_3까지, 즉 DISCL 파라미터가 지정한 안전 거리만큼 가공 평면과 떨어진 지점까지 절입. 2번째 그룹의 G 코드에 의해 결정된 커브로 종점 P4로 접근(직선, 원호, 나선). G247 (4분원) 또는 G347 (반원) 이 활성이고 시작점 P_3이 종점 P_4에 의해 정의된 가공 평면 밖에 있는 경우 원호 대신 나선을 삽입합니다. 지점 P_2는 정의하지 않거나 P_3과 동일합니다. 원호 평면인지 나선 평면인지는 SAR 블록 (G17/G18/G19) 에서 활성인 평면에 의해 결정됩니다. 즉, 추종 블록은 접선 자체가 아니라 시작 접선을 투영시켜 원호를 정의합니다. 지점 P_0부터 P_3까지는 SAR 블록 이전까지 유효한 속도로 두 직선을 따라 이동합니다.
G341:	지점 P_0에서 지점 P_1까지 직선으로 접근. 파라미터 DISRP가 프로그래밍되지 않은 경우 이 직선은 가공 평면과 평행합니다. 지점 P_1에서 지점 P_2까지, 즉 DISCL 파라미터가 지정한 안전 거리만큼 가공 평면과 떨어진 지점까지 절입. P_2에서 P_3까지 가공 평면과 수직으로 절입. 2번째 그룹의 G 코드가 결정한 커브로 종점에 접근. G247 또는 G347의 경우 나선 대신 항상 원호를 삽입하기 때문에 P_3과 P_4가 가공 평면 내에 위치합니다.

활성 평면 G17/G18/G19의 위치 (원호 평면, 나선 평면, 활성 평면과 수직인 절입 모션) 를 포함하는 경우 모든 활성 회전 프레임이 적용됩니다.

직선 접근 시 길이 또는 원호 접근 시 반경 (DISR)

- 직선을 따라 접근/후퇴

  DISR은 형상 시작점부터의 절삭날 모서리 거리를 명시합니다. 즉 TRC가 활성 상태일 때의 직선 길이는 공구 반경과 DISR의 프로그램된 값을 더한 값입니다. 공구 반경은 양수인 경우에만 적용됩니다.
  계산된 직선 길이는 반드시 양수여야 합니다. 즉, DISR의 절대값이 공구 반경보다 작을 때만 음의 DISR 값이 허용됩니다.

- 원을 따라 접근/후퇴
  DISR은 공구 중심점 경로의 반경을 지정합니다. TRC가 활성화되면 프로그램된 반경의 공구 중심점 경로에 나타난 반경의 원이 만들어집니다.

가공 평면부터 지점 P2까지의 거리 (DISCL)

지점 P_2의 위치를 원호 평면과 수직인 축을 기준으로 한 절대값으로 지정하려면 이 값은 반드시 DISCL=AC(...) 형식으로 프로그래밍해야 합니다.

DISCL=0 에 다음이 적용됩니다:

- G340 사용: 이때 접근 동작 전체는 두 블록으로만 구성됩니다 (P_1, P_2 및 P_3 의 조합). 접근 형상은 P_1 에서 P_4까지에 의해 형성됩니다.
- G341 사용: 전체 접근 형상은 3 블록으로 구성됩니다 (P_2 및 P_3 의 조합). P_0 과 P_4가 동일 면에 있다면 두 블록만 합성됩니다 (P_1에서 P_3까지의 절입 이동은 생략됩니다).
- DISCL에 의해 정의된 지점이 P_1 and P_3사이에 있는지 확인하기 위해 모니터링 됩니다. 즉, 부호는 반드시 전체 동작에서 가공 평면과 수직인 컴포넌트의 부호와 동일해야 합니다.
- 방향 반전 시 머신 데이터 MD20204 $MC_SAR_CLEARANCE_TOLERANCE에서 정의한 공차가 허용됩니다.

가공 평면부터 지점 P1 (후퇴 평면) 까지의 거리 (DISRP)

지점 P_1의 위치를 가공 평면과 수직인 축을 기준으로 한 절대값으로 지정하려면 이 값은 반드시 DISRP=AC(...) 형식으로 프로그래밍해야 합니다.

이 파라미터를 프로그래밍하지 않으면 지점 P_1부터 가공 평면까지 거리는 지점 P_0부터 가공 평면까지 거리와 같아집니다. 다시 말해 P_0부터 P_1까지의 접근 직선이 가공 평면과 평행하게 됩니다.

시스템은 DISRP가 정의한 지점이 P_0과 P_2 사이에 있는지 점검합니다. 즉, 가공 평면과 수직인 요소가 포함된 모든 이동 (예: 절입 이동, P_3부터 P_4까지 접근 이동) 에서 이 요소의 부호와 정의된 지점의 부호가 일치해야 합니다. 방향은 변경할 수 없습니다. 이 조건을 위반하면 알람이 출력됩니다.

방향 반전 시 머신 데이터 MD20204 $MC_SAR_CLEARANCE_TOLERANCE에서 정의한 공차가 허용됩니다. P_1이 P_0과 P_2에 의해 정의된 범위를 벗어나지만 그 편차가 이 공차보다 작거나 같은 경우, P_1은 P_0 또는 P_2에 의해 정의된 평면에 있는 것으로 간주합니다.

종점 프로그래밍

일반적으로 종점은 X... 로 프로그램됩니다. Y... Z...

접근을 위한 형상 종점 프로그래밍은 후퇴를 위한 형상 종점 프로그래밍과 상당히 다릅니다. 따라서 여기서는 두 프로그래밍을 나누어 설명하도록 하겠습니다.

접근을 위한 형상 종점 P4 프로그래밍

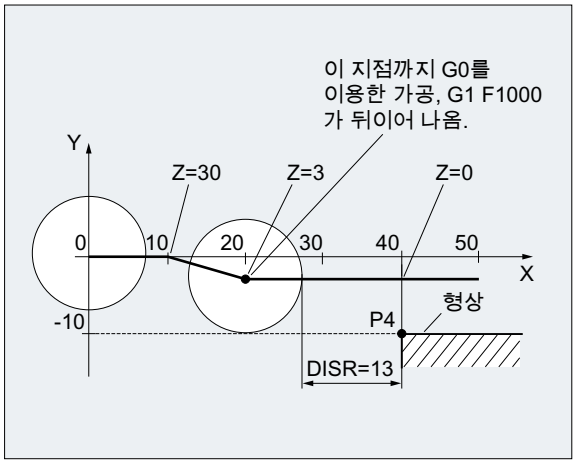
종점 P_4를 실제 SAR 블록에 프로그래밍할 수 있습니다. 또는 P_4를 다음 이송 블록의 종점으로 정의할 수 있습니다. 기하 축을 이동하지 않고도 SAR 블록과 다음 이송 블록 사이에 블록을 더 삽입할 수 있습니다.

예:

프로그램 코드	코멘트
$TC_DP1[1,1]=120	; 밀링 공구 T1/D1
$TC_DP6[1,1]=7	; 7 mm 반경의 공구
N10 G90 G0 X0 Y0 Z30 D1 T1	
N20 X10	
N30 G41 G147 DISCL=3 DISR=13 Z=0 F1000	
N40 G1 X40 Y-10	
N50 G1 X50	
...	

N30/N40은 다음으로 교체될 수 있습니다:
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000

또는
N30 G41 G147 DISCL=3 DISR=13 F1000
N40 G1 X40 Y-10 Z0



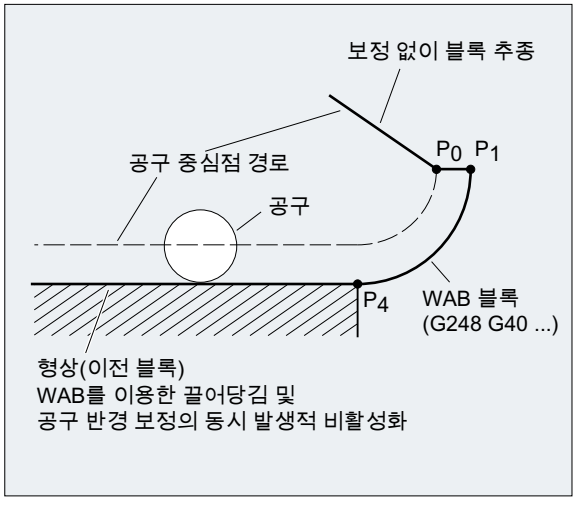
후퇴를 위한 형상 종점 P0 프로그래밍

후퇴의 경우 SAR 형상의 종점을 추종 블록에 프로그램할 수 없습니다. 다시 말해 프로그램된 축의 개수와 상관 없이 종점을 항상 SAR 블록에서 가져옵니다. 종점을 결정할 때, 다음 3가지 경우를 구별할 수 있습니다.

1. SAR 블록에 프로그램된 기하 축이 없는 경우. 이 경우 형상은 지점 P_1 (DISRP가 프로그램된 경우), 지점 P_2 (DISCL은 있지만 DISRP가 프로그램되지 않은 경우) 또는 지점 P_3 (DICLS와 DISRP가 모두 프로그램되지 않은 경우) 에서 끝납니다.
   가공 평면을 설명하는 축의 경우 후퇴 형상으로 위치를 결정합니다 (직선 또는 원호의 종점). 여기에 수직인 축 콤포넌트는 DISCL 또는 DISPR에 의해 정의됩니다. 이 경우 DISCL=0이고 DISRP=0이면 평면에서 모든 이동이 이루어집니다. 즉, P_0과 P_3이 일치합니다.
2. SAR 블록에 가공 평면과 수직인 축만 프로그램된 경우. 이 경우 형상은 지점 P_0에서 끝납니다. DISRP가 프로그램된 경우, 즉 지점 P_0과 P_1이 다른 경우 P_1에서 P_0까지의 직선이 가공 평면과 수직이 됩니다. 나머지 두 축의 위치는 1항의 경우와 동일한 방식으로 결정합니다.
3. 적어도 가공 평면의 1개 축이 프로그램 됩니다. 선행 블록의 마지막 위치에서 가공 평면의 2번째 축을 모달로 결정할 수 있습니다.

가공 평면과 수직인 축의 위치는 축이 프로그램되었는지 여부에 따라 1항 또는 2항의 설명에 따라 결정합니다. 이렇게 결정된 위치가 종점 P_0을 정의합니다. SAR 후퇴 블록을 사용하여 공구 반경 보정을 비활성화하려면 1과 2항의 경우 모두 후퇴 형상 끝에서 공구 반경 보정이 비활성화될 때 이동이 실행되지 않도록 가공 평면에 P_1부터 P_0 사이의 경로 요소를 추가로 삽입해야 합니다. 이 지점은 수정할 형상의 위치가 아닌 공구 중심점을 정의합니다. 3항의 경우 공구 반경 보정을 해제하기 위한 별도의 조치가 필요하지 않습니다. 프로그램된 지점 P_0이 이미 전체 형상의 끝에서 공구 중심점의 위치를 직접 정의하기 때문입니다.

공구 반경 보정을 동시에 비활성화하도록 가공 평면에 종점이 명시적으로 프로그램되지 않은 1항과 2항의 경우 아래 그림과 같이 동작합니다.

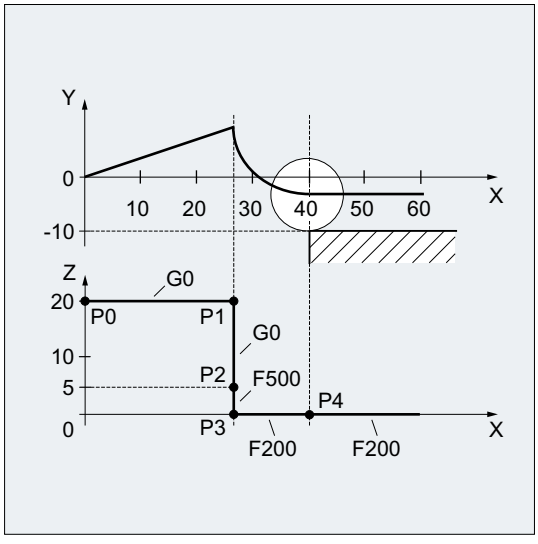


접근 및 후퇴 속도

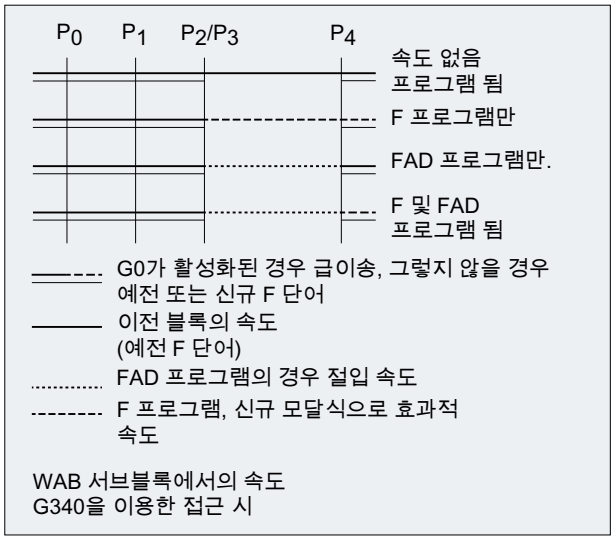
- 이전 블록 속도 (G0)
  P_0 에서 P_2까지의 모든 모션, 즉 가공 평면에 평행한 모션과 안전 공차까지의 절입 모션의 일부는 이 속도에서 실행됩니다.
- FAD를 이용한 프로그래밍
  이송 속도 지정
  - G341: P_2 에서 P_3까지 가공 평면과 수직으로 절입 이동
  - G340: 지점 P_2 또는 P_3에서 P_4까지 이동.
    SAR 블록에 속도를 정의하는 F 명령이 없는 경우, FAD를 프로그래밍하지 않으면 이 부분의 형상은 선행 블록에서 모달로 적용되는 속도로 이송됩니다.
- 프로그램된 이송 속도 F
  만일 FAD가 프로그램되지 않았다면 이 이송속도 값은 P_3 또는 P_2 를 기준으로 유효합니다.
  어떤 F 문자도 SAR 블록에서 프로그램되지 않았다면 이전 블록의 속도가 활성화됩니다.

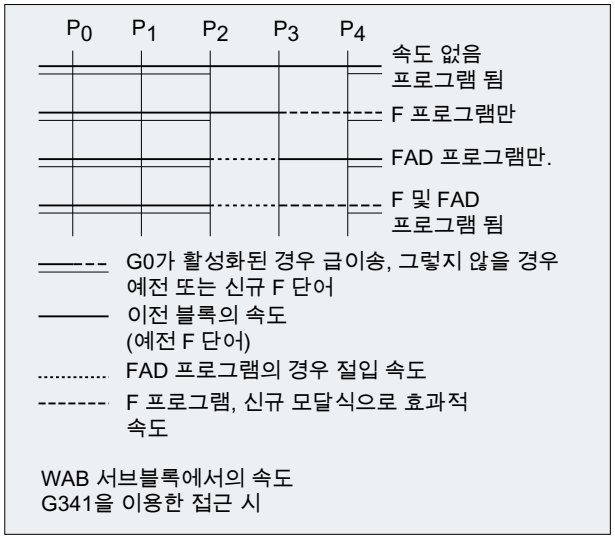
예:

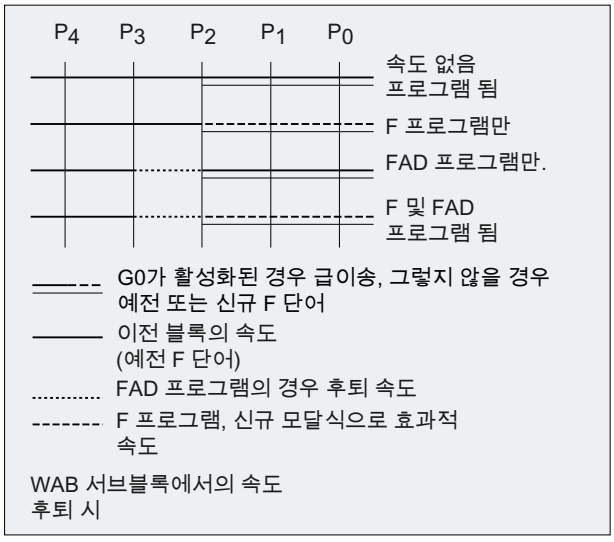
프로그램 코드	코멘트
`$TC_DP1[1,1]=120`	`; 밀링 공구 T1/D1`
`$TC_DP6[1,1]=7`	`; 7 mm 반경의 공구`
`N10 G90 G0 X0 Y0 Z20 D1 T1`	
`N20 G41 G341 G247 DISCL=AC(5) DISR=13 FAD 500 X40 Y-10 Z=0 F200`	
`N30 X50`	
`N40 X60`	
`...`	



후퇴할 때는 이전 블록에서 모달로 활성화된 이송 속도의 역할과 SAR 블록에서 프로그래밍된 이송 속도 값의 역할이 뒤바뀝니다. 즉, 실제 후퇴 형상은 이전 이송 속도 값으로 이송되고 F 워드로 프로그래밍된 새로운 속도가 P_2에서 P_0까지 적용됩니다.







읽기 위치

점 P_3 과 P_4는 접근 동안 시스템 변수로 WCS에서 판독됩니다.

- $P_APR: P 읽기
- $_3$ (최초 지점)
- $P_AEP: P 읽기
- $_4$ (형상 시작 지점)
- $P_APDV: $P_APR과 $P_AEP의 데이터가 유효한지 판독

3.10.4.2 연장 후퇴 방식을 이용한 접근과 후퇴 (G460, G461, G462)

어떤 특수 기하학적 상황에서 공구 반경 보정을 활성화 또는 비 활성화하기 위해 접근과 후퇴 블록의 활성화된 충돌 탐지를 사용한 이전의 이행과 비교되는 특별히 연장된 접근과 후퇴 방식이 요구됩니다. 예를 들어 충돌 탐지 기능 때문에 형상의 일부가 완전히 가공되지 않을 수도 있습니다. 다음 그림을 참조하십시오.

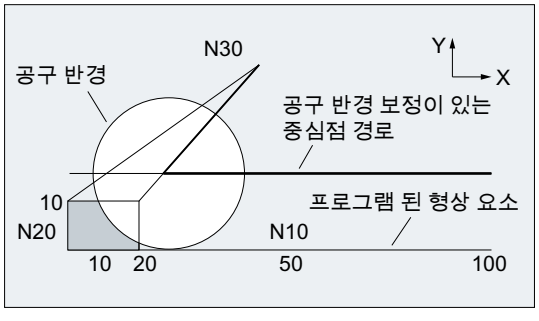


그림 3-21 G460을 사용한 후퇴 동작

구문

```
G460
G461
G462
```

의미

G460:	이전과 동일 (접근 및 후퇴 블록에 충돌 탐지 활성화)
G461:	TRC 블록에의 원 삽입, 중심점이 보정되지 않은 블록의 종점이 되는 교차와 반경이 공구 반경과 같은 교차가 불가능한 경우. 교차 지점까지, 가공은 형상 종점 주위의 **보조 원**을 사용하여 수행됩니다 (즉, 형상의 끝까지).
G462:	TRC 블록에의 원 삽입, 교차가 불가능한 경우; 해당 블록은 접선에 의해 연장됩니다 (디폴트 설정). 최종 형상 요소의 **연장**까지 가공됩니다 (즉, 형상의 끝 바로 전까지).

참고

접근 동작은 후퇴 동작에 대칭합니다.

접근/후퇴 동작은 접근/후퇴 블록 내 G 명령어의 상태에 의해 결정됩니다. 그러므로 접근 동작은 후퇴 동작과는 별개의 것으로 설정될 수 있습니다.

예

예 1: G460을 사용한 후퇴 동작

다음의 예제는 공구 반경 보정을 비활성화하는 상황에만 해당됩니다. 접근 동작은 정확히 동일합니다.

프로그램 코드	코멘트
G42 D1 T1	; 공구 반경 20 mm
...	
G1 X110 Y0	
N10 X0	
N20 Y10	
N30 G40 X50 Y50	

예 2: G461을 사용한 접근

프로그램 코드	코멘트
N10 $TC_DP1[1,1]=120	; 밀링 공구 종류
N20 $TC_DP6[1,1]=10	; 공구 반경
N30 X0 Y0 F10000 T1 D1	
N40 Y20	
N50 G42 X50 Y5 G461	
N60 Y0 F600	
N70 X30	
N80 X20 Y-5	
N90 X0 Y0 G40	
N100 M30	

추가 정보

G461

최종 TRC 블록과 선행 블록 사이에서 교차가 불가능하면 이 블록의 옵셋 곡선은 중심점이 보정되지 않은 블록의 종점에 놓이고 반경이 공구 반경과 동일한 원으로 연장됩니다.

시스템은 선행 블록 중 하나로 이 원을 절단하려 합니다.

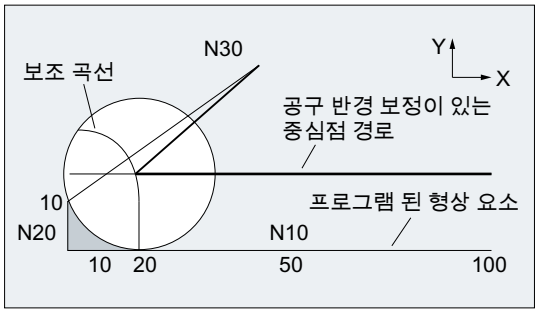


그림 3-22 G461을 사용한 후퇴 동작

충돌 모니터링 CDON, CDOF

CDOF 상태에서 (충돌 모니터링, CDON, CDOF 단락을 참조하시오.) 교차가 발견된 경우 검색은 취소됩니다. 즉, 시스템은 이전 블록에서의 추가 교차 존재 여부를 점검하지 않습니다.

CDON 이 활성화되면 첫 교차 발견 후에도 추가 교차 검색을 계속합니다.

이러한 방법으로 찾아진 교차 지점은 선행 블록의 새로운 종점과 비활성화 블록의 시작점이 됩니다. 삽입된 원은 교차를 계산하기 위해서만 사용되고 이송 동작은 만들지 않습니다.

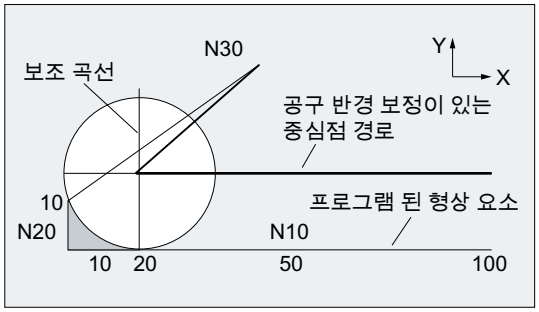
참고

교차가 발견되지 않으면 알람 10751 (충돌 위험) 이 발생됩니다.

G462

최종 TRC 블록과 선행 블록 사이에서 교차가 발견되지 않으면 G462 (초기 설정) 를 사용한 후퇴에서 공구 반경 보정된 최종 블록 (블록은 끝 정접에 의해 연장됨) 의 종점에 직선이 삽입됩니다.

이때 교차 검색은 G461의 절차와 동일합니다.



G462을 사용한 후퇴 동작 (예제 참조)

G462로 예제 프로그램에서 블록 N10과 N20으로 생성된 코너는 사용된 공구를 통해 실제적으로 가능한 전체 범위까지 가공되지는 않습니다. 그러나, 가공 형상 (프로그램된 형상과는 현저히 다름) 이 10mm보다 큰 y 값으로도 본 예제의 N20 좌측으로 넘어가지 않게 된다면 이 동작이 필요할 수 있습니다.

KONT를 사용한 코너 동작

KONT가 활성화되면 (시작 또는 종점에서 형상 주위를 이동) 이 동작은 종점이 형상의 전면 또는 뒷면에 위치하는 여부에 따라 달라집니다.

- **형상 앞쪽에 있는 종점**
  종점이 형상의 앞쪽에 있는 경우의 후퇴 동작은 NORM의 경우와 동일합니다. G451의 마지막 형상 블록이 직선 또는 원으로 확장되는 경우에도 이 속성은 달라지지 않습니다. 따라서 형상 종점 주변에서의 형상 침범을 방지하기 위한 추가 우회 전략은 필요하지 않습니다.
- **형상 뒤쪽에 있는 종점**
  종점이 형상 뒤쪽에 있다면 G450/G451에 따라 원이나 직선이 삽입됩니다. 이 경우 G460-462 는 효력을 발생하지 않습니다. 이 상황에서 최종 이송 블록에 선행 블록과의 교차가 없다면 삽입된 형상 구성 요소와 또는 우회 원 종점에서 프로그램된 종점까지의 직선과의 교차가 발생할 수 있습니다.
  삽입된 형상 구성 요소가 원 (G450) 이고 이것이 선행 블록과의 인터페이스를 형성한다면 이는 NORM 과 G461로 발생할 수 있는 인터페이스와 동일하게 됩니다. 그러나 일반적으로 원의 남은 부분은 여전히 이송되어야 합니다. 후퇴 블록의 직선 부분에 대해서는 교차에 대한 추가 계산이 필요하지 않습니다.
  두 번째 경우로는, 선행 블록과 삽입된 형상 요소 간에 인터페이스가 없는 경우에는 후퇴 직선과 선행 블록 사이의 교차가 이송됩니다.
  따라서 G460에서 벗어나는 동작은 NORM이 활성 상태이거나 KNOT의 동작이 기하학적으로 NORM의 동작과 동일한 경우에만 활성 G461 또는 G462를 통해 발생할 수 있습니다.

3.10.5 충돌 탐지("병목 탐지") 활성화/비활성화(CDON, CDOF, CDOF2)

NC 프로그램에서 공구 반경 보정이 활성 상태인 충돌 모니터링("병목 탐지")은 G 그룹 23의 명령으로 활성화 또는 비활성화됩니다.

구문

```
G41/G42 CDON
...
CDOF/CDOF2
```

의미

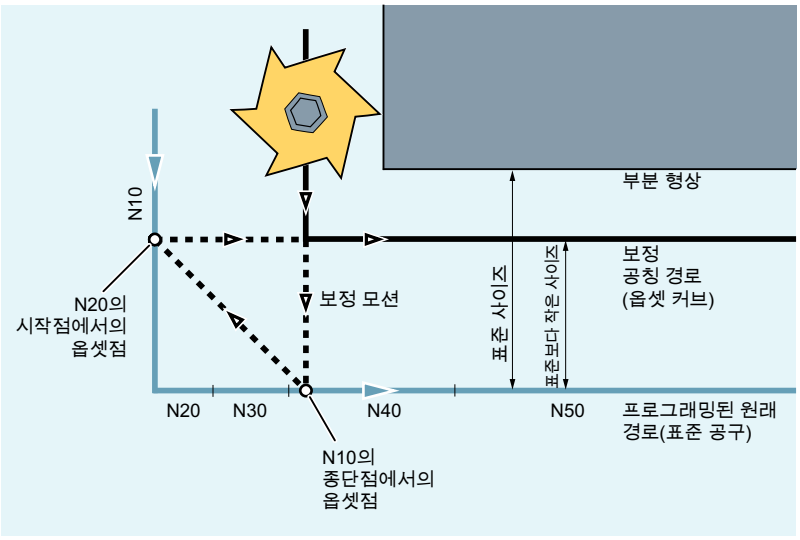
CDON:	충돌 탐지("병목 탐지") 활성화 CDON을 통해 비인접 또는 비연속 블록들의 공구 경로가 서로 교차하는지 여부와 관련하여 조절 가능한(MD20240) 개수의 블록을 점검할 수 있습니다. 선행 제어 기능으로 충돌을 미리 감지하여 시스템이 충돌을 방지할 수 있습니다.
CDOF:	충돌 감지("병목 탐지") 비활성화 CDOF로 현재 블록과의 공통 교차점을 찾기 위해 이전 이송 블록(안쪽 코너)에서 검색을 실행합니다. 필요한 경우 그보다 더 이전 블록까지 검색을 확장할 수 있습니다. 교차점을 발견하면 블록 검사가 중단됩니다. 바깥쪽 코너의 경우, 교차점은 항상 두개의 연속 블록 사이에서 찾을 수 있습니다. **참고:** CDOF는 NC 프로그램에서의 정보 부족 등으로 인해 발생할 수 있는 병목 탐지 오류를 방지하기 위해 사용할 수 있습니다.
CDOF2:	3D 원주 밀링의 충돌 탐지 비활성화 공구 옵셋 방향은 CDOF2를 사용하여 인접 블록 부분에서 결정됩니다. CDOF2는 3D 원주 밀링(CUT3Dx)에만 적용되며, 그 외 모든 종류의 가공(예: CUT2D 또는 CUT3DFD)에서 CDOF와 동일한 의미를 가집니다.

예제를 이용한 충돌 방지의 효과

NC 프로그램에서는 공구의 중심점을 기준으로 프로그램 합니다. 실제 사용되는 공구의 형상은 다음 그림에서는 기하학적인 관계를 예시하기 위해 과장되어 크게 나타나 있지만 실제로는 과소 규격입니다.

또한 그 형상은 예제에서 전체 3 개 블록만을 보여 줍니다.

MD20240 $MC_CUTCOM_MAXNUM_CHECK_BLOCKS = 3



교차는 두 블록 N10과 N40의 옵셋 곡선 사이만 존재하기 때문에 두 블록 N20과 N30은 생략됩니다. 예제에서 제어장치는 N40 블록에서 N10 이 완전히 처리되었는지 알 수 없습니다. 그러므로 단지 싱글 블록만이 생략될 수 있습니다.

활성 CDOF2를 사용하여 그림에 나타난 보정 동작이 실행되고 계속됩니다. 이 상황에서 CDOF 또는 CDON에 따라 알람이 발생됩니다.

3.10.6 2 1/2 D 공구 옵셋 (CUT2D, CUT2DD, CUT2DF, CUT2DFD)

경사 표면을 가공할 때 **공작물**이 회전하도록 하려면 공구 정렬이 아니라 2½ D 공구 반경 보정이 사용되어야 합니다. 이 기능은 CUT2D, CUT2DD, CUT2DF 또는 CUT2DFD와 같은 명령어를 사용하여 활성화됩니다.

공구 길이 옵셋

회전하지 않고 공간에 고정되어 있는 가공 평면에 대해 공구 길이 보정은 항상 고려됩니다.

형상 공구를 위한 2½ D 공구 반경 보정

CUT2D, CUT2DD, CUT2DF 또는 CUT2DFD 와 함께 2개의 명령어, 즉 G41 (형상 왼쪽 공구 반경 보정) 또는 G42 (형상 오른쪽 공구 반경 보정) 중 하나가 프로그램 된 경우, 형상 공구를 위한 2½ D 공구 반경 보정이 활성화됩니다.

참고

2½ D 공구 반경 보정이 활성화되지 않으면 형상 공구는 첫 번째 절삭날만을 사용하는 표준 공구처럼 작동합니다.

차등 공구에 대한 2½ 공구 반경 보정

차등 공구에 대한 2½ D 공구 반경 보정은 CUT2DD 또는 CUT2DFD 명령어를 사용하여 활성화됩니다. 프로그램된 형상이 차등 공구의 중심점 경로를 따르거나 가공에 차등 공구 외 다른 공구가 사용되는 경우 이 보정이 적용되어야 합니다. 2½ D 공구 반경 보정을 계산할 때는 활성 공구 ($TC_DP_15) 반경의 마모와 프로그램 되었을 공구 옵셋OFFN (페이지 277) 및 TOFFR (페이지 92)만 고려됩니다. 활성 공구의 기본 반경 ($TC_DP6) 는 고려되지 **않습니다**.

구문

```
CUT2D
CUT2DD
CUT2DF
CUT2DFD
```

의미

CUT2D:	2½ D 반경 보정 활성화
CUT2DD:	차등 공구에 대한 2½ D 반경 보정 활성화
CUT2DF:	2½ D 반경 보정 활성화, 현 프레임 및/또는 경사 면에 대한 공구 반경 보정
CUT2DFD:	2½ D 반경 보정 활성화, 현 프레임 및/또는 경사 면에 대한 공구 반경 보정

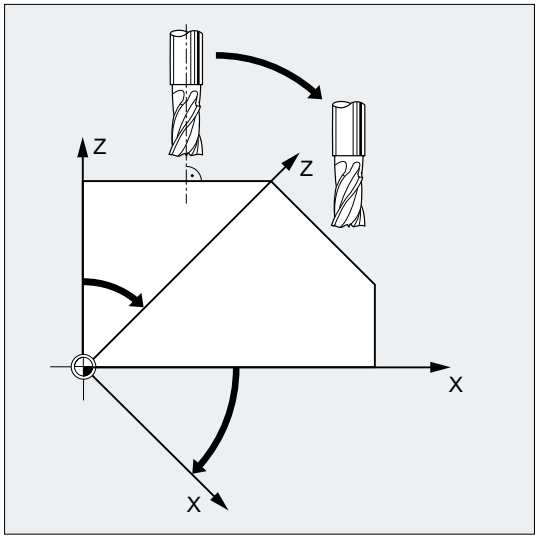
추가 정보

형상 공구

- 인에이블
  형상 공구를 위한 공구 반경 보정은 다음을 이용한 채널 고유의 기초 사항에 의해 이루어집니다:
  MD28290 $MC_MM_SHAPED_TOOLS_ENABLE
- 공구 종류
  형상 공구 유형은 다음을 이용한 채널 고유의 기초 사항에 의해 정의됩니다:
  MD20370 $MC_SHAPED_TOOL_TYPE_NO
- 절삭날
  각 형상 공구에 여러 개의 절삭날을 (D개) 순서에 관계 없이 할당할 수 있습니다. 절삭날 최대 개수는 다음을 사용하여 파라미터화할 수 있습니다:
  MD18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL
  공구를 활성활 때 선택되는 형상 공구의 첫 번째 절삭날. `T3 D5` 명령어를 사용하여 프로그램에서 세 번째 공구 (T3)의 다섯 번째 절삭날 (D5)이 활성화된 경우, D5 및 다음 절삭날이 일부분 또는 전체적으로 형상 공구를 정의합니다. D5 전에 위치한 절삭날은 무시됩니다.

수정 평면 (CUT2D, CUT2DD) 회전이 없는 경우의 2½ D 공구 반경 보정

회전을 포함하고 있는 프레임이 프로그래밍되면 `CUT2D` 또는 `CUT2DD`의 경우 공구 반경 보정 (수정 평면)이 이뤄지는 평면은 **동시에 회전하지 않습니다**. 공구 반경 보정은 **비회전** 가공 평면 (`G17`, `G18`, `G19`)에 준하여 고려됩니다. 공구 길이 보정은 수정 평면에 따라 반응합니다.

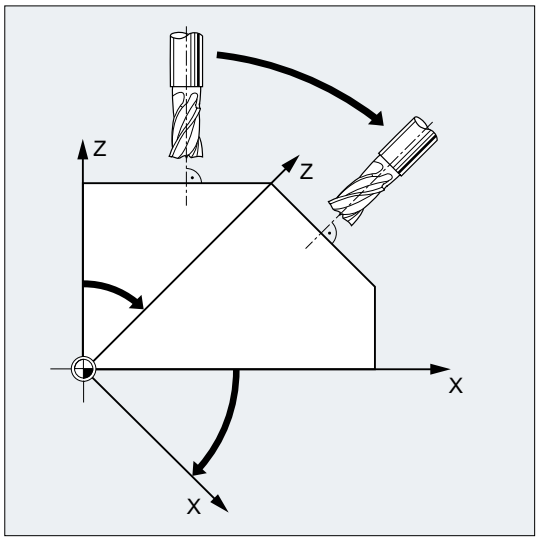


경사 표면을 가공할 경우, "방향을 지정할 수 있는 공구에 대한 공구 길이 보정"을 위한 함수에 기초하여 공구 옵셋이 적절히 정의되거나 계산되어야 합니다.

보정 평면 (CUT2DF, CUT2DFD)의 회전 시 2½ D 공구 반경 보정

회전을 포함하고 있는 프레임이 프로그래밍되면 CUT2DF 또는 CUT2DFD의 경우 공구 반경 보정 (수정 평면)이 이뤄지는 평면도 동시에 **회전합니다**. 공구 반경 보정은 **회전** 가공 평면 (G17, G18, G19)에 준하여 고려됩니다. 그러나 공구 길이 보정은 여전히 **비회전** 가공 평면에 준하여 적용됩니다.

요구사항: 기계에서 공구 방향은 반드시 회전되는 기계 평면에 수직으로 조정될 수 있어야 하고, 가공을 위해서는 그렇게 설정되어야 합니다.



참고

공구 길이 보정은 지속적으로 회전되지 않은 가공 평면에 적용됩니다.

자세히 알아보려면 '공구' 기능 매뉴얼을 참고하십시오.

3.10.7 일정한 공구 반경 보정 유지(CUTCONON, CUTCONOF)

"일정한 공구 반경 보정 유지" 기능은 공구 반경 보정 블록 수를 억제하여 프로그램된 것과 이전 블록에서 공구 반경 보정으로 설정되어 이동한 실 공구 중심 경로와의 차이가 보정으로 유지되게 하는데 사용됩니다. 반전 지점에서의 라인 밀링 동안 여러 개의 이송 블록이 필요하지만 공구 반경 보정으로 만들어진 형상을 원하지 않을 때에 이 방법을 사용할 수 있습니다. 이는 공구 반경 보정의 종류와는 독립적으로 사용될 수 있습니다 ($2^1/_2$D, 3D 페이스 밀링, 3D 원주 밀링).

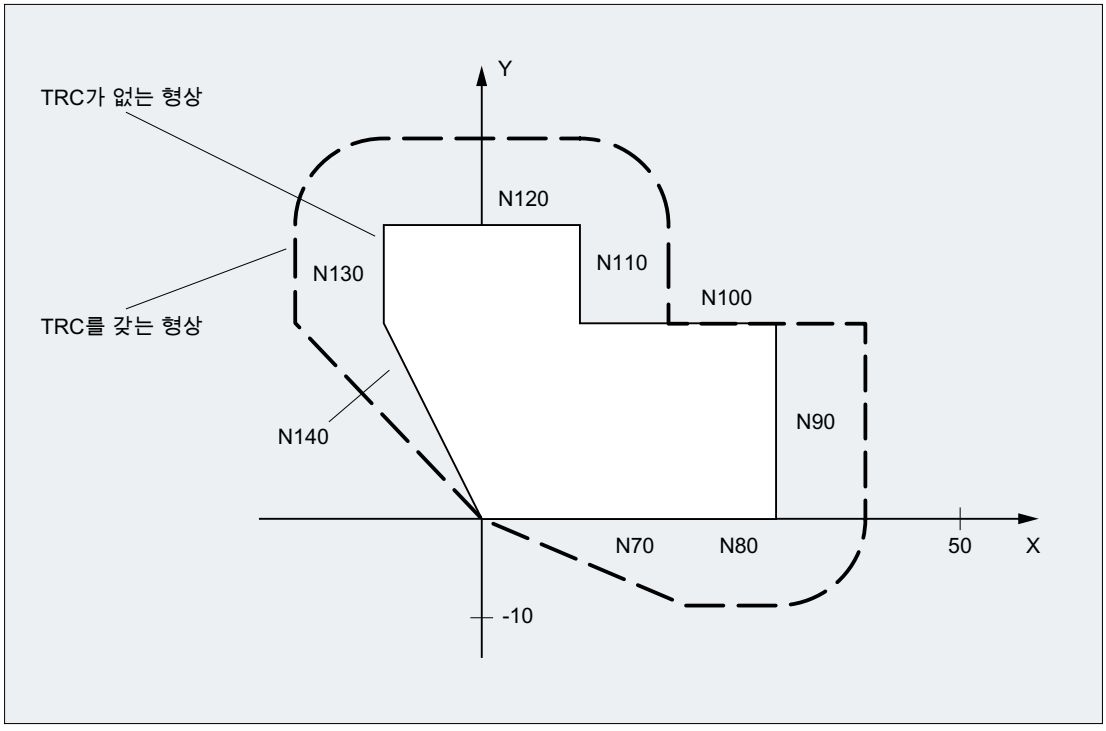
구문

```
CUTCONON
CUTCONOF
```

의미

CUTCONON:	"일정한 공구 반경 보정 유지" 기능을 활성화하는 명령어
CUTCONOF:	"일정한 공구 반경 보정 유지" 기능을 비활성화하는 명령어

예



프로그램 코드	코멘트
N10	; 공구 d1의 정의.
N20 $TC_DP1[1,1] = 110	, 종류
N30 $TC_DP6[1,1]= 10.	, 반경
N40	
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000	
N60	
N70 X20 G42 NORM	
N80 X30	

프로그램 코드	코멘트
N90 Y20	
N100 X10 CUTCONON	; 보정 억제의 활성화.
N110 Y30 KONT	; 필요하다면 보정 억제를 비활성화할 때 우회 원을 삽입하시오.
N120 X-10 CUTCONOF	
N130 Y20 NORM	; TRC를 비활성화할 때 우회 원 없음.
N140 X0 Y0 G40	
N150 M30	

그 외의 정보

공구 반경 보정은 보통 보정 억제 전에 활성화되어 있고 보정 억제가 다시 비활성화될 때 여전히 활성화됩니다. CUTCONON 이전 최종 이송 블럭에서 블럭 끝점에 있는 옵셋 지점이 접근됩니다. 옵셋 억제가 활성화되어 있는 모든 후속 블럭은 옵셋없이 이송됩니다. 그러나 이들은 최종 옵셋 블럭의 끝점에서 이의 옵셋까지의 벡터에 의해 옵셋됩니다. 이들 블럭은 모든 종류의 보간을 가질 수 있습니다(선형, 원형, 다면).

보정 억제의 비활성화 블럭 즉 CUTCONOF를 포함한 블럭은 정상으로 보정됩니다. 인느 시작점의 옵셋 지점에서 시작합니다. 하나의 선형 블럭이 이전 블럭의 끝점 사이에 삽입됩니다, 즉, 활성화된 CUTCONON을 사용한 최종 프로그램된 이송 블럭과 옵셋 지점 사이에 삽입된다는 것입니다.

보정 면에 수직인 원형 면(수직 원) 때문에 원형 블럭은 이들이 CUTCONON 프로그램되었지만 처리됩니다. 옵셋 억제의 이러한 보이지 않는 활성화는 옵셋 면에 이송 동작이 없고 그러한 원을 가지지 않은 첫 번째 이송 블럭에서는 자동으로 취소됩니다. 이런 의미의 수직 원은 원주 밀링동안에만 발생할 수 있습니다.

3.10.8 적절한 절삭날 위치의 공구

적절한 공구 지점 방향이 지정된 공구의 경우 (선삭 및 연삭 공구 - 공구 종류 400~599, "마모 평가 부호" 절 참조) G40에서 G41/G42로, 또는 그 반대로 변경하면 공구 교환으로 간주됩니다. 변환이 활성화되면 (예, TRANSMIT) 선-처리공정 중지 (디코딩 중지) 에 이르게 되어 원하는 가공 형상으로부터의 편차를 가져 올 수 있습니다.

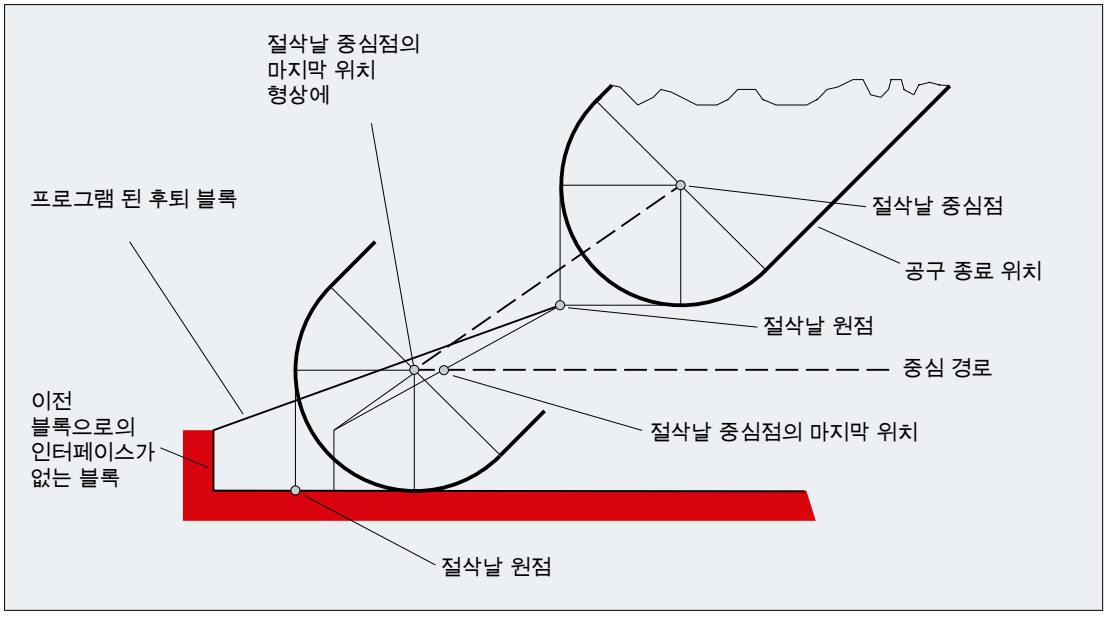
원래의 기능성은 아래와 관련하여 변경됩니다:

1. TRANSMIT에서 선-처리 공정 중지
2. KONT를 사용한 접근과 후퇴에서의 교차 지점 계산
3. 활성 공구 반경 보정으로 공구 교환
4. 변형에서 가변 공구 오리엔테이션의 공구 반경 보정

추가 정보

원래의 기능이 다음과 같이 수정되었습니다.

- G40 에서 G41/G42로 변경하거나 혹은 반대로 변경할 경우, 더 이상 공구 교환으로 처리되지 않습니다. 그러므로 TRANSMIT로서 선-처리공정 중지는 더 이상 발생하지 않습니다.
- 블록 시작 및 블록 끝에서 공구날 중심점 사이에 있는 직선이 접근과 후퇴 블록과의 교차 지점을 계산하는데 사용됩니다. 공구날 원점과 공구날 중심점 간의 차이는 이 이동에서 겹쳐집니다.
  KNOT를 사용한 접근과 후퇴에서 (공구는 형상 지점을 우회 이동합니다. 위의 하위 단락 "형상 접근과 후퇴"를 참조하십시오.) 겹침은 접근 또는 후퇴 동작의 직선 가공 블록에서 발생합니다. 그러므로 기하학적 조건은 적절한 공구 지점 방향의 유무와 관계없이 동일합니다. 이전 동작과의 편차는 접근 또는 후퇴 블록이 인접 이송 블록과 교차하지 않는 드문 상황에서만 발생합니다, 아래 그림을 참조하십시오.



- 원호 블록 및 분모 각도가 4보다 큰 공통 각도의 논리 폴리노미널을 포함한 이동 블록에서, 공구날 중심점과 공구날 원점과의 거리가 변경하는 곳의 경우 활성화된 공구 반경 보정을 사용한 공구 교환은 허용되지 않습니다 다른 종류의 보간을 사용하면 변형이 활성화될 때 변경하는 것이 가능해집니다 (예, TRANSMIT)
- 가변 공구 오리엔테이션의 공구 반경 보정의 경우 공구날 원점에서 공구날 중심점까지의 변환은 단순 워크 옵셋의 방법으로는 더 이상 수행되지 않습니다. 그러므로 적절한 공구 지점 방향의 공구는 3D 외면 밀링에 허용되지 않습니다. (알람이 발생됩니다.)

참고

어쨌던 적절한 공구 지점 방향 없이 정의된 공구 종류만 이 작업에 허용되기 때문에 이 주제는 페이스 밀링에 관해서는 부적절합니다. (명확하게 승인되지 않은 종류의 공구는 지정된 반경의 볼 엔드 밀링으로 처리됩니다. 공구 지점 방향 파라미터는 무시됩니다.)

3.11 경로 동작

3.11.1 정위치 정지 (G60, G9, G601, G602, G603)

정위치 정지 이송 모드에서 모든 경로 축과 모달로 이송되지 않는 이송 동작에 관련된 특수 축은 정지될 때까지 각 블록의 끝에서 감속됩니다.

날카로운 외부 코너가 가공되거나 내부 코너가 정확한 치수로 정삭될 때 정위치 정지가 사용됩니다.

정위치 정지로 코너 지점으로 정확하게 접근되는 방법과 다음 블록으로 트렌지션이 이행되는 때를 지정하게 됩니다.

- "미세 정위치 정지"
  이송에 포함된 전체 축에 대해, 이송 중 "미세 정위치 정지" 공차 내에 도달된 경우 블록 변경이 이루어집니다.
  "미세 정위치 정지"가 다음을 통해 설정됩니다: MD36010 $MA_STOP_LIMIT_FINE[<Axis>]
- "일반 정위치 정지"
  이송에 포함된 전체 축에 대해, 이송 중 "일반 정위치 정지" 공차 내에 도달된 경우 블록 변경이 이루어집니다.
  "일반 정위치 정지"는 다음을 통해 설정됩니다: MD36000 $MA_STOP_LIMIT_COARSE[<Axis>]
- "보간자 끝"
  이송에 포함된 전체 축에 대해, 이송 중 지령치가 0 으로 되는 경우 블록 변경이 이루어집니다. 실제 위치 또는 관련 축의 추종 오차는 계산되지 않습니다.

구문

```
G60 ...
G9 ...
G601/G602/G603 등
```

의미

G60:	**모달**정위치 정지를 활성화하는 명령어
G9:	**넌모달**정위치 정지를 활성화하는 명령어
G601:	정위치 정지 기준 '**미세 정위치 정지**'를 활성화하는 명령
G602:	정위치 정지 기준 '**일반 정위치 정지**'를 활성화하는 명령
G603:	정위치 정지 기준 '**보간자 끝**'을 활성화하는 명령

참고

정위치 정지 기준 (G601/G602/G603) 을 활성화하는 명령은 G60 또는 G9가 활성 상태인 경우에만 유효합니다.

예

프로그램 코드	코멘트
N5 G602	; 선택된 "일반 정위치 정지" 기준.
N10 G0 G60 Z...	; 정위치 정지 모달 활성화.
N20 X... Z...	; G60 은 계속 적용됩니다.
...	
N50 G1 G601	; 선택된 "미세 정위치 정지" 기준.
N80 G64 Z...	; 연속 경로 모드로 전환.
...	
N100 G0 G9	; 정위치 정지는 이 블록에서만 작동합니다.
N110 ...	; 연속 경로 모드 재 활성화.

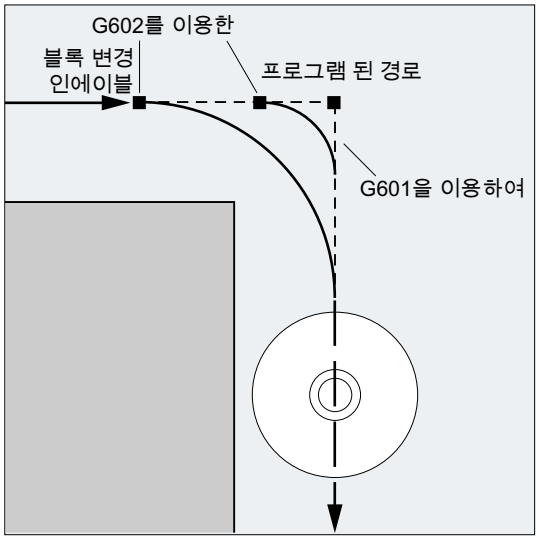
추가 정보

G60, G9

G9는 현재 블록에서 정위치 정지를 적용하고, G60은 현재 블록 및 전체 후속 블록에 정위치 정지가 적용됩니다.

연속 경로 모드 명령인 G64 또는 G641 - G645는 G60을 비활성화하는데 사용됩니다.

G601, G602



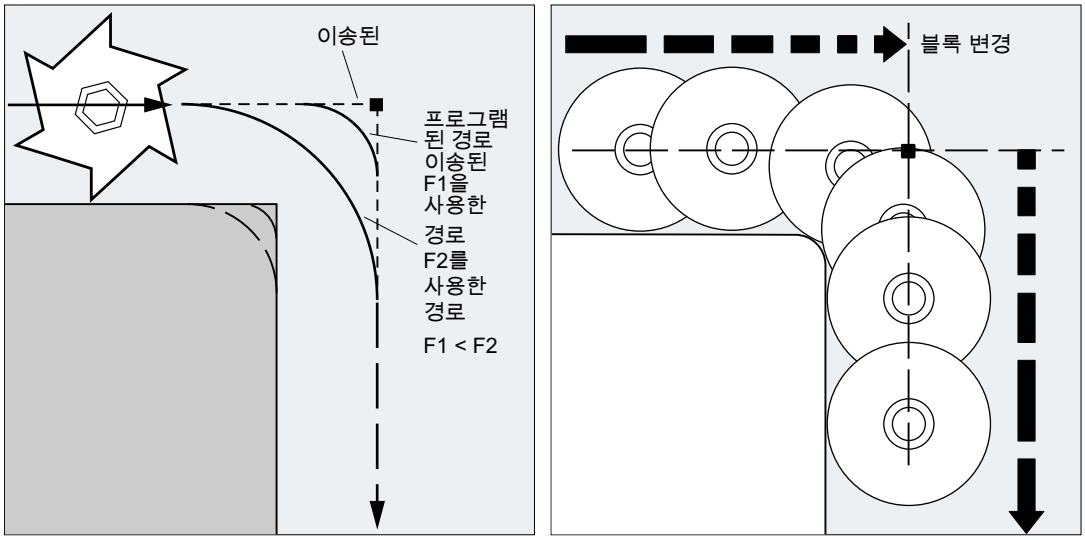
이동은 감속하고 코너 지점에서 잠시 정지합니다.

참고

정위치 정지 기준의 한계를 필요 이상으로 엄격하게 설정하지 마십시오. 한계가 엄격할수록 위치 지정과 목표 위치 접근에 더 긴 시간이 소요됩니다.

G603

해당 축의 지령 속도가 0이 된 경우 블록 변경이 이루어집니다. 이 시점에서 실제 값은 축과 축 속도의 역학적 반응에 따라 비례 인수만큼 뒤떨어지게 됩니다. 공작물 코너는 이때 라운딩될 수 있습니다.



환경 설정된 정위치 정지 기준

첫 번째 G그룹에서 G0 및 기타 명령에 대해 채널별 설정을 할 수 있으며, 이는 프로그래밍된 정위치 정지 기준과 달리 미리 설정된 기준을 자동으로 사용해야 한다는 뜻입니다(기계 제조업체에서 사양 참고).

3.11.2 연속 경로 모드 (G64, G641, G642, G643, G644, G645, ADIS, ADISPOS)

연속 경로 모드에서는 블록 끝의 경로 속도가 블록 변경 시 정위치 정지 기준을 만족하는 속도로 감속되지 않습니다. 사실 이 모드의 목표는 프로그램이 다음 블록으로 이동할 때 축 속도를 가능한 한 일정하게 유지하도록 블록 변경 지점에서 경로 축의 급격한 감속을 막는 것입니다. 이 목표를 달성하기 위해 연속 경로 모드를 선택하면 "선행 제어" 기능도 활성화됩니다.

스무딩과 함께 연속 경로 모드를 사용하면 프로그래밍된 형상의 국지적 변화로 인해 각이 진 블록 전이가 발생했을 때 더 쉽게 접선 형태 수정 및/또는 스무딩을 할 수 있습니다.

연속 경로 모드:

- 형상 라운딩
- 정위치 정지 기준을 준수하기 위해 필요한 제동 및 가속 프로세스를 제거함으로써 가공 시간 단축
- 보다 일정해진 속도로 절삭 조건 개선

다음의 경우 연속 경로 모드가 적합합니다.

- 형상을 가능한 한 빨리 이송해야 하는 경우 (예: 급 이송 시)
- 연속 형상을 얻기 위해 특정 공차 내에서 프로그래밍된 형상과 정확한 형상이 차이가 날 수 있는 경우

연속 경로 모드가 적합하지 않은 경우

- 형상을 정확하게 이송하려는 경우
- 전적으로 일정한 속도가 필요한 경우

참고

다음과 같이 전처리 정지를 암시적으로 일으키는 블록은 연속 경로 모드를 중단합니다.

- 특정 기계 상태 데이터 액세스 ($A...)
- 보조 기능 출력

구문

```
G64 ...
G641 ADIS=…
G641 ADISPOS=…
G642 ...
G643 ...
G644 ...
G645 ...
```

의미

G64:	과부하 팩터에 따라 속도가 줄어드는 연속 경로 모드
G641:	거리 기준에 따라 스무딩이 이루어지는 연속 경로 모드
ADIS=... :	경로 코드 G1, G2, G3 등에 대한 G641의 거리 기준
ADISPOS=... :	급 이송 G0에 대한 G641의 거리 기준

	거리 기준 (= 라운딩 공차) ADIS 또는 ADISPOS는 라운딩 블록이 끝나기 전에 포함할 수 있는 최대 거리 또는 라운딩 블록을 종료해야 하는 블록 끝부분 뒤의 거리를 나타냅니다. **참고:** ADIS/ADISPOS가 프로그램되지 않았다면 "0" 값이 적용되므로 이송 동작은 G64에 일치하게 됩니다. 짧은 이송 거리에서 라운딩 공차는 자동으로 감소 (36%까지) 합니다.
G642:	정해진 공차 안에서 스무딩하는 연속 경로 모드 이 모드에서는 정상 환경일 때 최대 허용 경로 편차 내에서 스무딩이 이루어집니다. 그러나 이와 같은 축 중심의 공차 대신 최대 형상 편차 (형상 공차) 또는 공구 오리엔테이션의 최대 각도 편차 (오리엔테이션 공차) 를 설정할 수 있습니다. **참고:** 형상 및 오리엔테이션 공차를 포함하는 확장 기능은 '폴리노미얼 보간' 옵션이 있는 시스템에서만 지원됩니다.
G643:	정해진 공차 안에서 스무딩하는 연속 경로 모드 (블록 내경) G643은 별개의 라운딩 블록을 생성하는 데 사용되지 않는다는 점에서 G642와 다르지만, 대신 축 관련 블록 내경 라운딩 동작이 삽입됩니다. 라운딩 공차는 축마다 다를 수 있습니다.
G644:	다이나믹 응답을 가능 최대치로 하여 스무딩하는 연속 경로 모드 **참고** G644는 활성화된 좌표계 변환에는 사용할 수 없습니다. 시스템은 G642로 내부적으로 전환합니다.
G645:	정해진 공차 안에서 스무딩 및 접선 블록 트렌지션을 포함하는 연속 경로 모드 G645는 코너에 대해 G642와 동일한 효과를 가집니다. G645를 사용하면 원래 형상의 곡률이 적어도 하나의 축에서 점프를 표시하는 경우 접선 블록 트렌지션에서만 라운딩 블록도 생성됩니다.

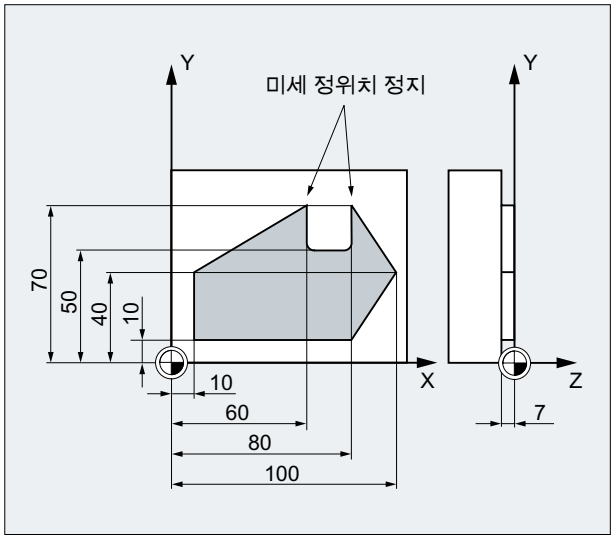
참고

라운딩은 스무딩 명령 (RND) 대용으로는 사용할 수 없습니다. 라운딩 구역 이내에 있는 형상의 외관과 관련하여 어떠한 가정도 해서는 안 됩니다. 라운딩의 종류는 공구 경로 속도와 같이, 역학적 조건에 따라 달라질 수 있습니다. 그러므로 형상에의 라운딩은 작은 ADIS 값으로만 사용 가능합니다. 정의된 형상이 코너에서 이송되는 경우에는 반드시 RND를 사용해야 합니다.

참고

G641, G642, G643, G644 또는 G645에 의해 개시된 라운딩 동작이 중단되면, 중단 지점보다는 원래 이송 블록의 시작 또는 종점 (REPOS 모드에 적합) 이 후속 위치 재지정 (REPOS) 에 사용되게 됩니다.

예



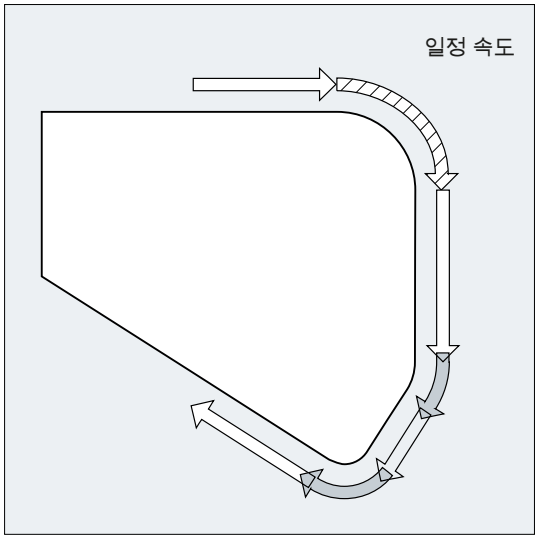
홈에서 두 개의 외경 코너는 정확하게 접근됩니다. 그렇지 않다면 가공은 연속 경로 모드에서 수행됩니다.

프로그램 코드	코멘트
N05DIAMOF	; 반경 치수
N10 G17 T1 G41 G0 X10 Y10 Z2 S300 M3	; 시작 위치로 접근, 스핀들 활성화, 경로 보정.
N20 G1 Z-7 F8000	; 공구 절입.
N30 G641 ADIS=0.5	; 형상 트렌지션은 스무딩됩니다.
N40 Y40	
N50 X60 Y70 G60 G601	; 미세 정위치 정지로의 위치 접근.
N60 Y50	
N70 X80	
N80 Y70	
N90 G641 ADIS=0.5 X100 Y40	; 형상 트렌지션은 스무딩됩니다.
N100 X80 Y10	
N110 X10	
N120 G40 G0 X-20	; 경로 보정 비활성화
N130 Z10 M30	; 프로그램 끝까지 공구 후퇴.

추가 정보

연속 이송 모드 G64

연속 이송 모드에서 공구는 가능한 일정한 경로 속도로 접선 형상 트렌지션을 가로 질러 이동합니다 (블록 경계에서 감속 없음). 선행 제어 감속은 정위치 정지를 통해 코너 및 블록 이전에 적용됩니다.



코너 역시 일정한 속도로 이송됩니다. 형상 오차를 최소화하기 위해 속도는 가속 한계와 과부하 팩터에 의거 감소됩니다.

참고

형상 트렌지션의 스무딩 범위는 이송 속도 및 과부하 팩터에 따라 달라집니다. 과부하 팩터는 MD32310 $MA_MAX_ACCEL_OVL_FACTOR에서 설정될 수 있습니다.

MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS를 설정하면 설정된 과부하 팩터와 관계없이 블록 트렌지션이 항상 라운딩됩니다.

경로 모션에서 원하지 않는 정지를 방지하기 위해 다음 지점을 주의해야 합니다 (릴리프 절삭):

- 이송 모션이 끝난 후 또는 다음 모션이 시작하기 전에 실행되는 보조 기능이 연속 경로 모드를 중단시킵니다 (예외: 신속 보조 기능).
- 포지셔닝 축은 항상 정위치 정지 원리, 미세 포지셔닝 창에 따라 이동합니다 (G601의 것처럼). NC 블록이 포지셔닝 축을 대기해야 할 경우, 연속 이송 모드는 경로 축에서 중단됩니다.

그러나 설명, 계산 블록 또는 서브프로그램 호출을 포함하는 중간 블록은 연속 이송 모드에 영향을 미치지 않습니다.

참고

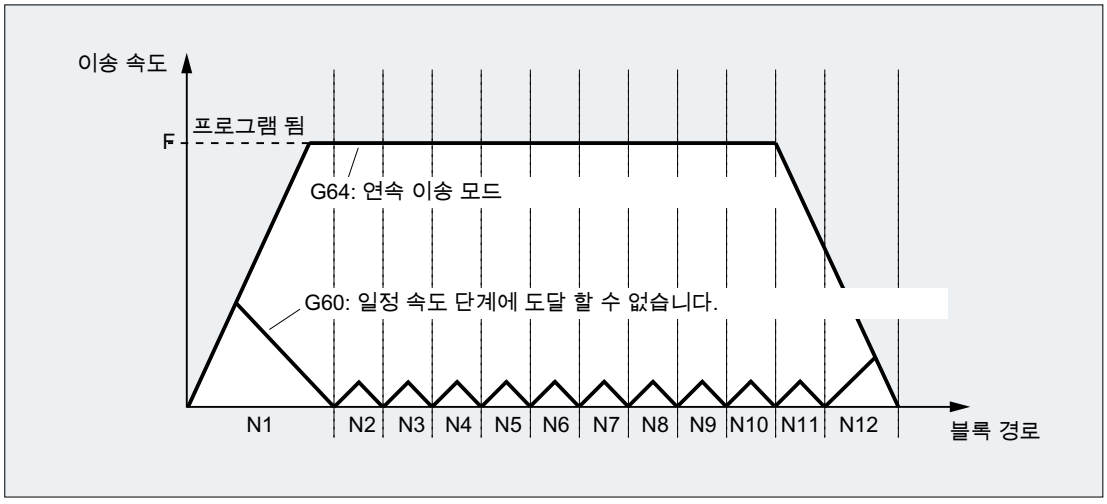
FGROUP이 모든 경로 축을 포함하지 않으면 FGROUP에서 제외된 축 때문에 블록 트렌지션의 속도에서 빈번한 단계 변화가 있게 되며 제어장치가 MD32300 $MA_MAX_AX_ACCEL 과 MD32310 $MA_MAX_ACCEL_OVL_FACTOR 에서 설정한 허용 가능한 값까지 이러한 변화를 제한합니다. 이러한 제동 작동은 경로 축 사이에 있는 특정 위치의 상호관계를 '스무딩'하는 기능의 적용을 통해 방지할 수 있습니다.

선행 제어 속도 예측 제어

연속 이송 모드에서는 제어 시스템이 여러 NC 블록의 속도 제어를 사전에 자동으로 결정합니다. 근접한 접선 트렌지션을 사용하여 다중 블록 전체에서 가속 및 감속할 수 있게 됩니다.

선행 제어는 고속으로 짧은 거리를 이동하는 모션으로 구성된 연속 이동 가공에 특히 적합합니다.

선행 제어 계산에 포함된 NC 블록의 개수는 머신 데이터를 통해 정의할 수 있습니다.



거리 기준에 따라 스무딩이 이루어지는 연속 이송 모드 (G641)

G641을 사용하여 제어장치는 형상 트렌지션에서 트렌지션 요소를 삽입합니다. 스무딩 공차 ADIS(또는 G0를 위한 ADISPOS)가 코너를 부드럽게 하는 라운딩의 최대 범위를 지정합니다. 이러한 스무딩 공차 내에서 컨트롤은 구성된 경로를 무시하고 동적으로 최적화 된 경로로 바꿀 수 있습니다.

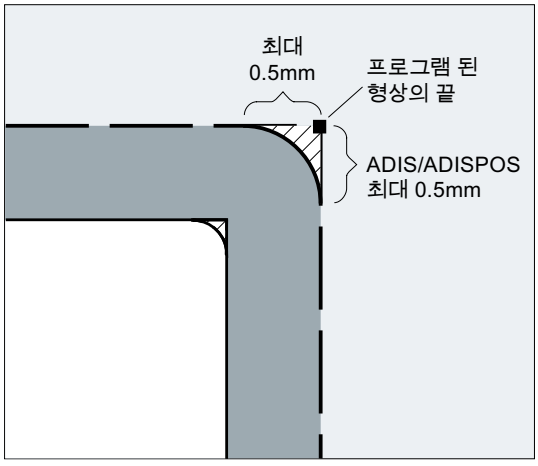
단점 : 단 하나의 ADIS 값이 모든 축에 유효합니다.

G641의 효능은 RNDM과 유사하지만 이는 작업 평면의 축에 제한되지 않습니다.

G64와 마찬가지로 G641도 선행 제어 속도 예측 제어를 통해 작동합니다. 각이 큰 굴곡의 스무딩 블록은 감소된 속도에서 접근됩니다.

예:

프로그램 코드	설명
`N10 G641 ADIS=0.5 G1 X... Y...`	`; 스무딩 블록은 프로그래밍된 블록 끝 이전의 0.5 mm 이내에서 시작하여 블록 끝 다음의 0.5 mm 이내에서 끝나야 합니다. 이 설정은 모달로 남아 있게 됩니다.`



참고

스무딩은 정의된 스무딩 (RND, RNDM, ASPLINE, BSPLINE, CSPLINE) 의 기능을 대체할 수 도 없으며 해서도 안 됩니다.

G642를 사용해 정밀 축으로 스무딩

G642의 경우에는 정해진 ADIS 범위 내에서 스무딩이 이루어지지 않지만, MD33100 $MA_COMPRESS_POS_TOL로 정의된 축 공차가 적용됩니다. 스무딩 공차는 모든 축의 최단 스무딩 경로를 기준으로 결정됩니다. 이 값은 스무딩 블록을 생성할 때 고려됩니다.

G643을 사용한 블록 내경 스무딩

G643을 사용하여 스무딩할 때 정밀 형상에서의 최대 편차는 머신 데이터 MD33100 $MA_COMPRESS_POS_TOL을 사용하는 각 축에 정의됩니다.

G643은 별개의 스무딩 블록을 생성하는 데 사용되지 않지만, 축별 블록 내경 스무딩 모션이 삽입됩니다. G643의 경우 각 축의 스무딩 공차가 다를 수 있습니다.

G642/G643을 사용하여 형상 및 오리엔테이션 공차로 스무딩

MD20480 $MC_SMOOTHING_MODE는 G642와 G643을 사용한 스무딩을 설정하는 데 사용되어 축별 공차 대신에 형상 공차와 오리엔테이션 공차가 적용될 수 있게 합니다.

형상 공차 및 오리엔테이션 공차는 채널 관련 설정 데이터에서 설정됩니다.

SD42465 $SC_SMOOTH_CONTUR_TOL (최대 형상 편차)

SD42466 $SC_SMOOTH_ORI_TOL (공구 오리엔테이션의 최대 각도 편차)

설정 데이터는 NC 프로그램에서 프로그래밍할 수 있습니다. 즉, 각 블록 트렌지션마다 서로 다르게 지정할 수 있습니다. 형상 공차와 공구 오리엔테이션 공차의 매우 상이한 지정은 G643을 사용할 때만 유효합니다.

참고

형상 및 오리엔테이션 공차를 포함하는 확장 기능은 '폴리노미얼 보간' 옵션이 있는 시스템에서만 지원됩니다.

참고

오리엔테이션 공차 내에서 스무딩하는 경우에는 오리엔테이션 변환이 활성화되어야 합니다.

G644에서 최대로 가능한 다이나믹 응답으로 스무딩

다이나믹 응답 가능 최대치를 이용한 스무딩은 MD20480 $MC_SMOOTHING_MODE로 수많은 위치에서 지정됩니다.

값	의미
0	최대 축 편차 지정: MD33100 $MA_COMPRESS_POS_TOL
1	다음과 같이 프로그래밍하여 최대 스무딩 공차 지정: ADIS=... 또는 ADISPOS=...

값	의미
2	스무딩 범위에서 발생하는 각 축의 최대 가용 빈도로 지정: MD32440 $MA_LOOKAH_FREQUENCY 스무딩 모션 중에 지정된 최대값을 초과하는 빈도가 발생하도록 스무딩 범위가 정의됩니다.
3	G644를 이용한 스무딩에서 공차 또는 스무딩 공차를 모니터링합니다. 각 축은 가능한 최대의 다이나믹 반응으로 코너 둘레를 이동합니다. SOFT를 사용하여 각 축의 최대 가속과 최대 저크를 유지합니다. BRISK 명령어를 사용하여 저크가 제한되지 않는 대신 각 축은 가능한 최대 가속으로 이동합니다.

G645를 사용한 접선 블록 트렌지션의 스무딩

G645를 사용하여 모든 관련 축의 가속이 점프 없이 유연하게 유지되고 원래 형상 (MD33120 $MA_PATH_TRANS_POS_TOL) 에서 파라미터로 지정된 최대 편차가 초과되지 않도록 스무딩 모션이 정의됩니다.

비 접선 각도 블록 트렌지션의 경우에는 스무딩 동작이 G642의 경우와 동일합니다.

중간 스무딩 블록 없음

다음의 경우에는 중간 스무딩 블록이 삽입되지 않습니다.

- 축이 두 블록 사이에서 중지합니다.
  이는 다음의 경우 발생합니다:
  - 모션 전에 그 다음 블록에 보조 기능 출력이 있는 경우.
  - 그 다음 블록에 경로 모션이 없는 경우.
  - 경로 축이 이전까지 포지셔닝 축인 상태로, 처음으로 다음 블록에서 경로 축으로 이동된 경우.
  - 포지셔닝 축이 이전까지 경로 축인 상태로, 처음으로 다음 블록에서 포지셔닝 축으로 이동된 경우.
  - 이전 블록에서는 기하 축을 이송하였으나, 다음 블록에서는 아닌 경우.
  - 다음 블록에서는 기하 축을 이송하였으나, 이전 블록에서는 아닌 경우.
  - 태핑 이전에 다음 블록에서는 준비기능으로 G33이 사용되었으나, 이전 블록에서는 아닌 경우.
  - BRISK 및 SOFT 간에 변경이 발생된 경우.
  - 변환에 관련된 축이 경로 모션으로 완전하게 할당되지 지정되지 않은 경우 (예, 오실레이션, 포지셔닝 축).
- 스무딩 블록에 의해 가공 프로그램 실행이 감속됩니다.
  이러한 현상은 다음과 같은 경우에 발생합니다.
  - 매우 짧은 블록들 사이에서.
    각 블록에 최소한 하나의 보간 싸이클이 필요한 경우, 추가된 중간 블록으로 인해 가공 시간이 두 배로 연장될 수 있습니다.
  - 블록 트렌지션 G64 (스무딩 없는 연속 이송 모드) 가 속도 감속 없이 이송될 수 있는 경우.
    스무딩으로 가공 시간이 증가합니다. 이는 허용된 과부하 팩터 (MD32310 $MA_MAX_ACCEL_OVL_FACTOR) 값이 블록 트렌지션의 라운딩 유무에 영향을 줍니다. 과부하 팩터는 G641/G642를 사용한 스무딩에서만 계산됩니다. 과부하 팩터는 G643을 사용한 스무딩에 영향을 주지 않습니다. (이 동작은 MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS를 TRUE로 설정하여 G641과 G642에 설정될 수 있습니다.)

- 스무딩이 파라미터 설정되지 않습니다.
  이는 다음의 경우 발생합니다:
  - G0 블록에 있는 G641의 경우, ADISPOS=0 (사전 할당!)
  - 비 G0 블록에 있는 G641의 경우, ADISPOS=0 (사전 할당!)
  - G641의 경우, G0에서 비G0으로 또는 비G0에서 G0으로 전환하려면 ADISPOS 및 ADIS의 값 중 낮은 값이 적용
  - G642/G643의 경우 전체 축별 공차가 0인 경우
- 이 블록에는 이동 동작이 없습니다 (제로 블록).
  이는 다음의 경우 발생합니다:
  - 동기 동작이 동작중인 경우.
    일반적으로 해석기로 제로 블록은 제거됩니다. 그러나 동기 동작이 활성화되면 이러한 제로 블록이 포함되고 실행됩니다. 따라서, 동작 중인 프로그램에 맞게 정위치 정지가 시작됩니다. 이로서 동기 동작이 또한 전환됩니다.
  - 제로 블록은 프로그램 점프에 의해 생성됩니다.

급 이송 G0 의 연속 이송 모드

급이송 모션의 경우 G60/G9 또는 G64/G641/G645 기능 중 하나를 지정해야 합니다. 그렇지 않으면 머신 데이터의 기본값이 사용됩니다.

3.12 좌표 변형(프레임)

3.12.1 프레임

프레임

프레임은 직교 좌표계를 다른 직교 좌표계로 변환하는 내포된 산술적 규칙입니다.

베이직 프레임 (베이직 옵셋)

베이직 프레임은 기본 좌표계 (BCS) 에서 베이직 제로 시스템 (BZS) 으로의 좌표 변환을 표시하고 설정 가능 프레임과 동일한 효과를 가집니다.

기본 좌표계 (BCS) (페이지 43)을 참조하십시오.

설정 가능 프레임

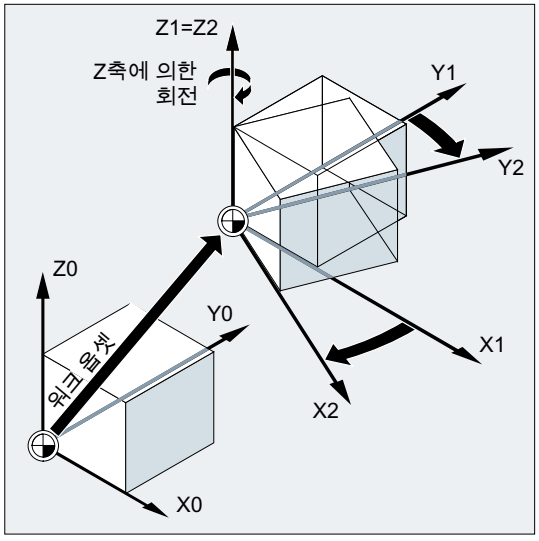
설정 가능 프레임이란 G54~G57 및 G505~G599 명령을 사용하여 NC 프로그램에서 호출할 수 있는 설정 가능 워크 옵셋을 말합니다. 옵셋 값은 사용자가 사전에 정의하고 컨트롤러의 워크 옵셋 메모리에 저장합니다. 이것은 공작물 제로 시스템 (SZS) 을 정의하는 데 사용됩니다.

참고:

- 셋터블 제로 시스템 (SZS) (페이지 45)
- 셋터블 워크 옵셋 (G54~G57, G505~G599, G53, G500, SUPA, G153) (페이지 160)

프로그래밍 프레임

종종 처음 선택한 공작물 좌표계(또는 '설정 가능한 제로 시스템')를 다른 위치로 이동, 회전, 미러링 및/또는 배율 조정을 해야 하거나 하는 것이 좋습니다. 이 경우 프로그래밍 프레임을 사용할 수 있습니다.



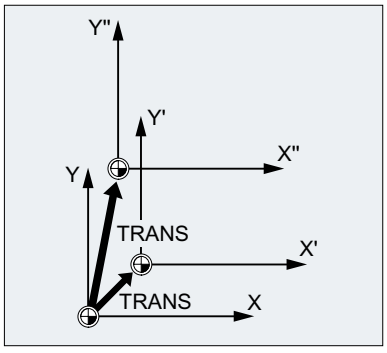
프레임 지침 (페이지 339)을 참조하십시오.

3.12.2 프레임 지침

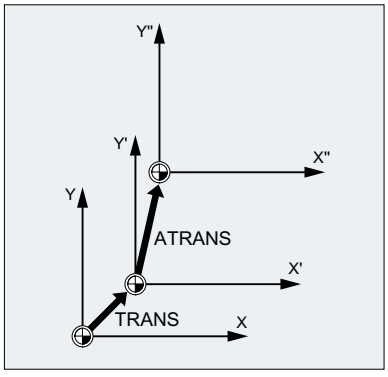
기능

프로그래밍 프레임에 대한 명령문은 현재 NC 프로그램에 적용됩니다. 이것은 추가 요소 또는 대체 요소의 역할을 합니다.

- 대체 명령문
  기존에 프로그래밍된 모든 프레임 명령문을 삭제합니다. 마지막에 호출된 셋터블 워크 옵셋 (G54~G57, G505~G599) 에 의해 원점이 결정됩니다.

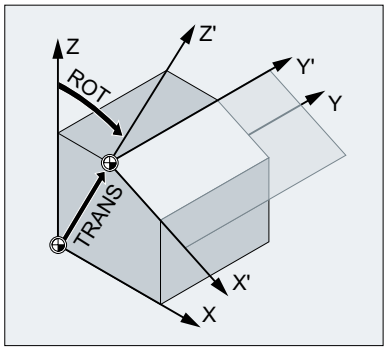


- 추가 명령문
  기존 프레임에 추가됩니다. 현재 설정된 공작물 원점 또는 프레임 명령문으로 프로그래밍한 마지막 공작물 원점에 의해 원점이 결정됩니다.



적용 사례

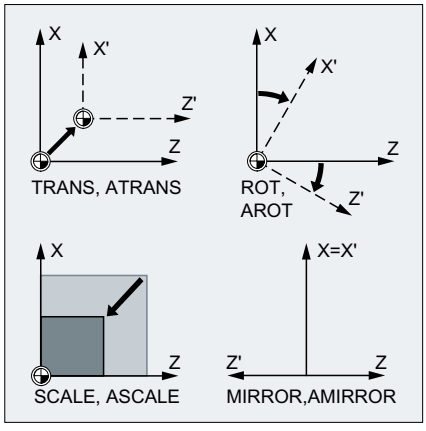
1. 공작물 좌표계 (WCS) 의 영점을 이동합니다.
2. 공작물 좌표계 (WCS) 를 회전시켜 평면이 원하는 작업 평면과 평행하게 만듭니다.

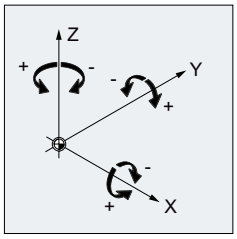


구문

대체 구문:	추가 구문
TRANS X… Y… Z…	ATRANS X… Y… Z…
ROT X… Y… Z…	AROT X… Y… Z…
ROT RPL=…	AROT RPL=…
ROTS/CROTS X... Y...	AROTS X... Y...
SCALE X… Y… Z…	ASCALE X… Y… Z…
MIRROR X0/Y0/Z0	AMIRROR X0/Y0/Z0

의미



TRANS/ATRANS:	지정된 기하 축 방향의 공작물 좌표계 옵셋		
ROT/AROT:	공작물 좌표계 회전: • 지정된 기하 축 주변에 개별 회전 연결 또는 • 현재 작업 평면 (G17/G18/G19) 의 각도 RPL=... 주변		
	회전 방향:		
	회전 순서:	RPY 표시법 사용:	Z, Y', X"
		Euler 각도 사용:	Z, X', Z"
	값 범위:	회전 각도는 다음 범위에서만 분명하게 정의됩니다.	
		RPY 표시법 사용:	-180 ≤ x ≤ 180
			-90 < y < 90
			-180 ≤ z ≤ 180
		Euler 각도 사용:	0 ≤ x < 180
			-180 ≤ y ≤ 180
			-180 ≤ z ≤ 180
ROTS/AROTS:	입체각 지정을 통한 공작물 좌표계 회전 공간에서의 평면 오리엔테이션은 2개의 입체각 지정을 통해 분명하게 정의됩니다. 따라서 최대 2개의 입체각을 프로그래밍할 수 있습니다. ROTS/AROTS X... Y... / Z... X... / Y... Z...		
CROTS:	CROTS는 ROTS와 동일한 방식으로 작동하지만 데이터베이스에서 유효한 프레임을 기준합니다.		
SCALE/ASCALE:	형상의 크기 증가/감소를 위해 지정된 기하 축 방향으로 스케일링		
MIRROR/AMIRROR:	지정된 기하 축을 미러링 (방향 변경) 하여 공작물 좌표계 미러링		
	값:	임의 선택 가능 (이 경우 "0")	

보충 조건

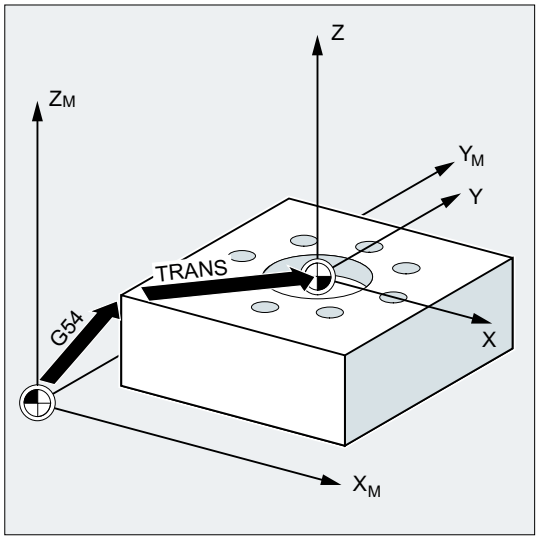
- 프레임 구문은 반드시 별도의 NC 블록에서 프로그램해야 합니다.
- 프레임 명령문은 필요에 따라 개별적으로 사용하거나 임의로 조합할 수 있습니다.
- 프레임 명령문은 프로그래밍된 순서대로 실행됩니다.
- 추가 명령문은 서브프로그램에서 빈번하게 사용됩니다. 서브프로그램에 SAVE 속성이 프로그램된 경우 서브프로그램을 종료해도 메인 프로그램에 정의된 기본 명령문은 삭제되지 않습니다.

3.12.3 프로그래밍 워크 옵셋 (TRANS, ATRANS)

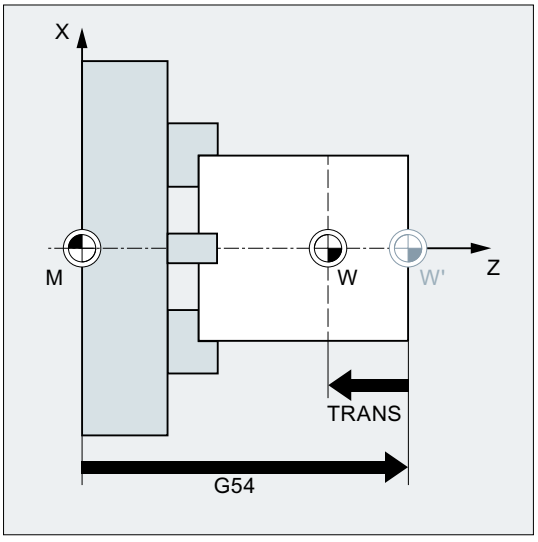
TRANS 명령은 설정 가능 워크 옵셋(G54 ... G57, G505 ... G599)을 적용하여 생성된 SZS를 절대적인 기준으로 WCS를 이동합니다.

ATRANS 명령은 TRANS로 생성된 WCS를 추가로 이동합니다.

밀링:



선삭:



구문

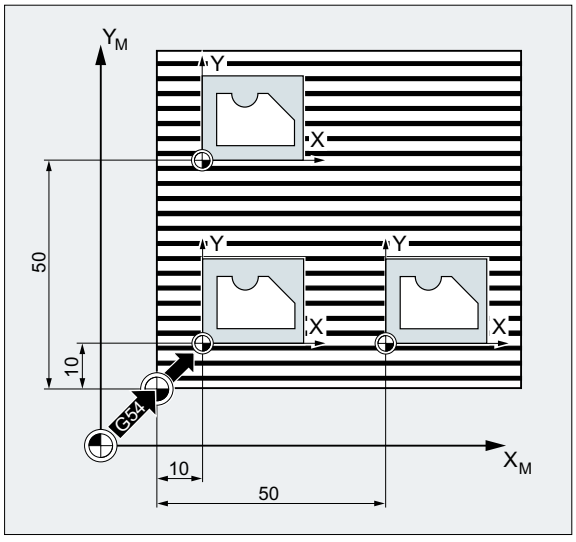
```
TRANS X... Y... Z...
ATRANS X... Y... Z...
```

의미

TRANS:	설정 가능 워크 옵셋(G54 ... G57, G505 ... G599)을 적용해 설정된 공작물 원점(SZS)을 기준으로 한 WCS의 절대치 옵셋.	
	블록에서만:	예
ATRANS:	TRANS로 설정한 공작물 원점을 기준으로 한 WCS의 추가 워크 옵셋	
	블록에서만:	예
X... Y... Z...:	지정된 기하 축 방향에 있는 옵셋 값	

예

예 1: 밀링



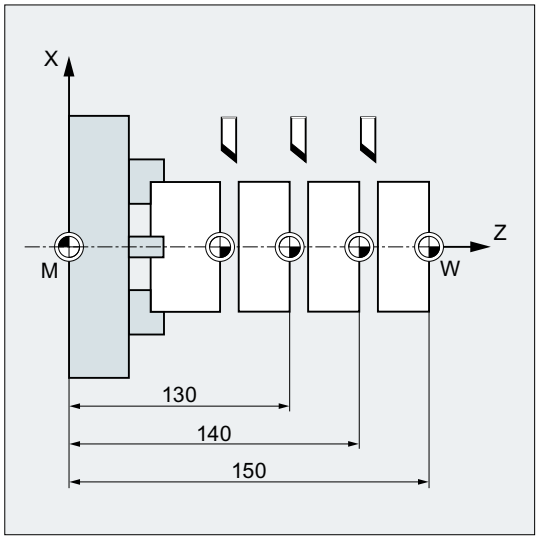
이 공작물의 경우 그림에 표시된 형태가 한 프로그램에서 반복됩니다.

이 형태에 대한 가공 순서는 서브프로그램에 저장됩니다.

워크 옵셋은 각 사례에서 필요한 공작물 원점을 설정하고 서브프로그램을 호출하는 데 사용됩니다.

프로그램 코드	설명
N10 G1 G54	; 작업 평면 X/Y, 공작물 원점
N20 G0 X0 Y0 Z2	; 시작점 접근
N30 TRANS X10 Y10	; 절대치 옵셋
N40 L10	; 서브프로그램 호출
N50 TRANS X50 Y10	; 절대치 옵셋
N60 L10	; 서브프로그램 호출
N70 M30	; 프로그램 종료

예 2: 선삭



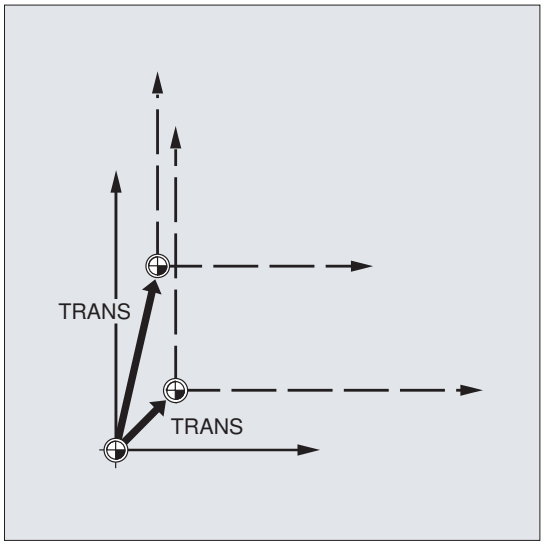
프로그램 코드	설명
...	
N10 TRANS X0 Z150	; 절대치 옵셋
N15 L20	;서브프로그램 호출
N20 TRANS X0 Z140 (또는 ATRANS Z-10)	; 절대치 옵셋
N25 L20	;서브프로그램 호출
N30 TRANS X0 Z130 (or ATRANS Z-10)	; 절대치 옵셋
N35 L20	;서브프로그램 호출
...	

추가 정보

TRANS X... Y... Z...

지정된 축 방향에서 프로그램된 옵셋 값을 통한 변환 (경로 축, 동기 축 및 포지셔닝 축) 최근 호출된 좌표 설정 (G54 ~ G57, G505 ~ G599) 에 의해 기준이 제시됩니다.

유의사항
오리지널 프레임 없음 TRANS 명령어는 이전에 활성화된 프로그램 프레임의 모든 프레임 구성 요소를 재설정합니다.

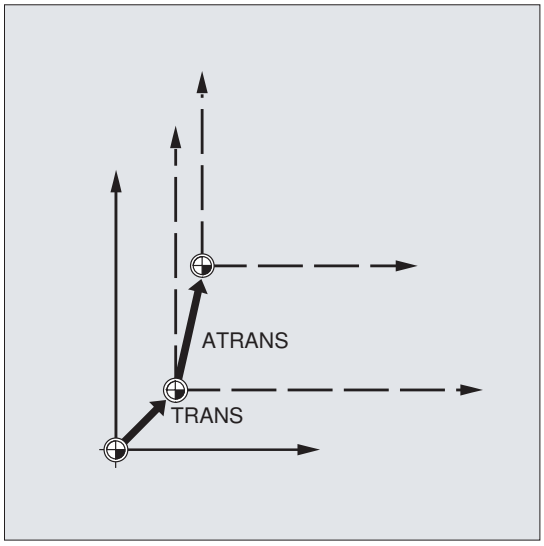


참고

ATRANS는 기존 프레임에 추가할 옵셋을 프로그래밍하는 데 사용할 수 있습니다.

ATRANS X... Y... Z...

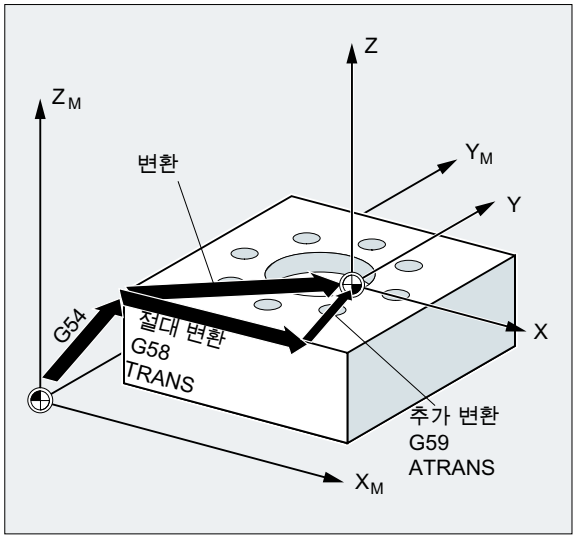
지정된 축 방향에서 프로그램된 옵셋 값을 통한 변환. 현재 설정되거나 또는 최종 프로그램된 영점이 기준으로 사용됩니다.



3.12.4 프로그래밍 워크 옵셋 (G58, G59)

G58 및 G59 기능을 사용하여 구체적인 축의 프로그래밍 가능 워크 옵셋(TRANS/ATRANS) (페이지 342)의 변환 요소를 대체할 수 있습니다.

- G58: 절대 변환 콤포넌트 (일반 옵셋)
- G59: 추가 변환 콤포넌트 (미세 옵셋)



요구 사항

G58 및 G59 코드는 미세 옵셋이 설정된 경우에만 사용할 수 있습니다 (MD24000 $MC_FRAME_ADD_COMPONENTS = 1).

구문

```
G58 <축_1><값_1> ... <축_3><값_3>
G59 <축_1><값_1> ... <축_3><값_3>
```

의미

G58:	G58은 지정된 축에서 프로그래밍 가능한 워크 옵셋의 절대 변환 구성요소를 대체하지만, 프로그래밍된 추가 옵셋은 유효한 상태를 유지합니다. 마지막 설정 가능한 워크 옵셋에 의해 원점이 결정됩니다 (G54 ... G57, G505 ... G599).
	블록에서만: 예

G59:	G59는 지정된 축에서 프로그래밍 가능한 워크 옵셋의 추가 변환 구성요소를 대체하지만, 프로그래밍된 절대 옵셋은 유효한 상태를 유지합니다.
	블록에서만: 예
<축_n>:	채널의 기하 축
<값_n>:	지정된 형상 축 방향으로의 옵셋 값

예

프로그램 코드	설명
...	
N50 TRANS X10 Y10 Z10	; 절대 변환 콤포넌트 X10 Y10 Z10
N60 ATRANS X5 Y5	; 추가 변환 콤포넌트 X5 Y5
	→ 옵셋 합계: X15 Y15 Z10
N70 G58 X20	; 절대 변환 콤포넌트 X20
	→ 옵셋 합계 X25 Y15 Z10
N80 G59 X10 Y10	; 추가 변환 콤포넌트 X10 Y10
	→ 옵셋 합계 X30 Y20 Z10
...	

추가 정보

절대 변환 콤포넌트 (**일반 옵셋**)은 다음 구문으로 수정합니다:

- TRANS
- G58
- CTRANS
- CFINE
- $P_PFRAME[X,TR]

추가 변환 콤포넌트 (**미세 옵셋**)은 다음 구문으로 수정합니다:

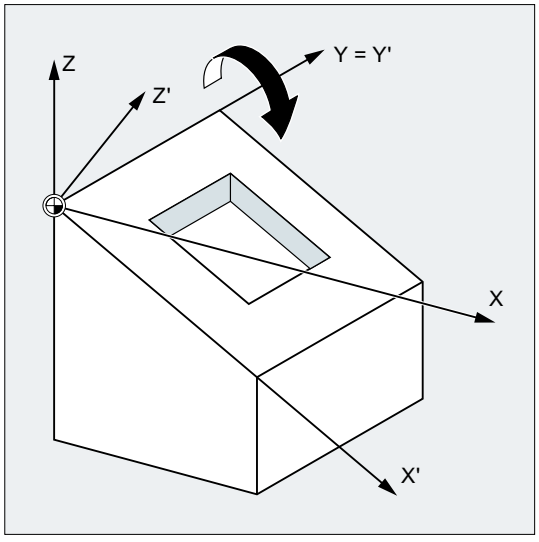
- ATRANS
- G59
- CTRANS
- CFINE
- $P_PFRAME[X,FI]

예

명령	일반 옵셋 V_C	미세 옵션 V_F
TRANS X10	$V_C = 10$	변경되지 않음
G58 X10	$V_C = 10$	변경되지 않음
$P_PFRAME[X,TR] = 10	$V_C = 10$	변경되지 않음
ATRANS X10	변경되지 않음	$V_F = V_F + 10$
G59 X10	변경되지 않음	$V_F = 10$
$P_PFRAME[X,FI] = 10	변경되지 않음	$V_F = 10$
CTRANS(X,10)	$V_C = 10$	$V_F = 0$
CTRANS()	$V_C = 0$	$V_F = 0$
CFINE(X,10)	$V_C = 0$	$V_F = 10$

3.12.5 프로그램 가능한 회전 (ROT, AROT, RPL)

ROT/AROT 구문으로 공작물 좌표계를 공간 상에서 회전시킬 수 있습니다. 이 구문은 전적으로 프로그래밍 프레임 $P_PFRAME을 기준으로 합니다.



구문

```
ROT <1st GeoAx><angle> <2nd GeoAx><angle> <3rd GeoAx><angle>
ROT RPL=<angle>
AROT <1st GeoAx><angle> <2nd GeoAx><angle> <3rd GeoAx><angle>
AROT RPL=<angle>
```

참고

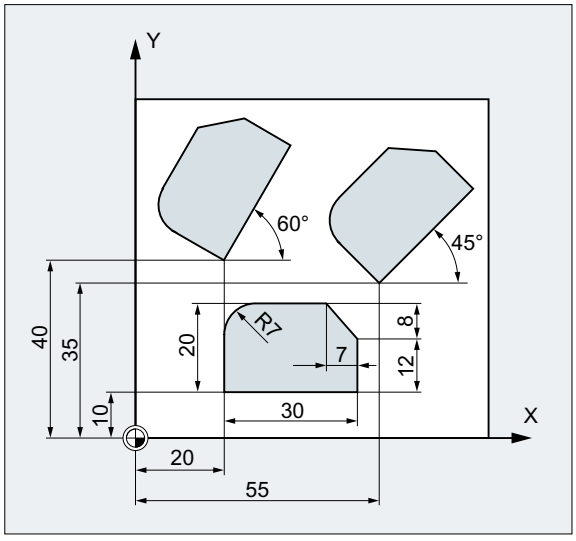
공작물 좌표계의 회전은 Euler 각도를 통해 수행됩니다.

의미

ROT:	절대치 회전	
	기준 프레임:	프로그래밍 프레임 $P_PFRAME
	원점:	G54 ... G57, G505 ... G599로 설정된 현재 공작물 좌표계의 영점
AROT:	추가 회전	
	기준 프레임:	프로그래밍 프레임 $P_PFRAME
	원점:	G54 ... G57, G505 ... G599로 설정된 현재 공작물 좌표계의 영점
<nth GeoAx>:	지정된 각도만큼 회전을 수행할 때 중심이 되는 n번째 기하 축의 이름. 프로그램되지 않은 기하 축인 경우 내부적으로 회전 각도를 0°로 설정합니다.	
RPL:	활성 평면 (G17, G18, G19) 과 수직인 기하 축을 중심으로 지정된 각도만큼 회전	
	기준 프레임:	프로그래밍 프레임 $P_PFRAME
	원점:	G54 ... G57, G505 ... G599로 설정된 현재 공작물 좌표계의 영점
<Angle>	° 단위 각도	
	값 범위:	-360° ≤ 각도 ≤ 360°

예

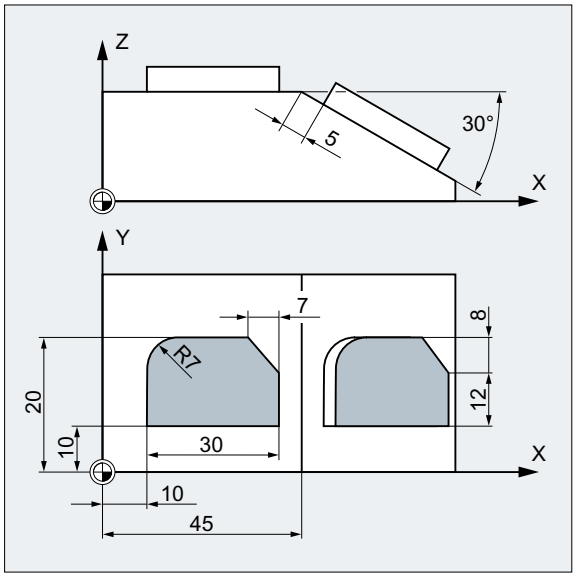
예 1: G17 평면에서 회전



이 공작물의 경우 그림에 표시된 형태가 한 프로그램에서 반복됩니다. 워크 옵셋에 추가적으로 해당 형상이 축과 평행하게 배치되지 않은 관계로 회전이 실행됩니다.

프로그램 코드	설명
N10 G17 G54	; 작업 평면 X/Y, 공작물 원점
N20 TRANS X20 Y10	; 절대치 옵셋
N30 L10	; 서브프로그램 호출
N40 TRANS X55 Y35	; 절대치 옵셋
N50 AROT RPL=45	; G17 평면과 수직인 Z 축을 중심으로 45° 추가 회전
N60 L10	; 서브프로그램 호출
N70 TRANS X20 Y40	; 절대치 옵셋 (이전 옵셋 전부를 재설정)
N80 AROT RPL=60	; G17 평면과 수직인 Z 축을 중심으로
	; 60° 추가 회전
N90 L10	; 서브프로그램 호출
N100 G0 X100 Y100	;후퇴
N110 M30	; 프로그램 종료

예 2: Y 축을 중심으로 공간 회전



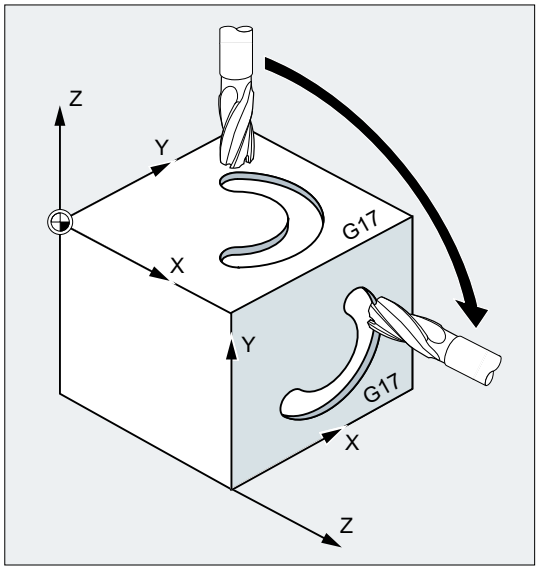
이 예에서 근축과 경사진 공작물 표면은 클램핑에서 가공됩니다.

조건:

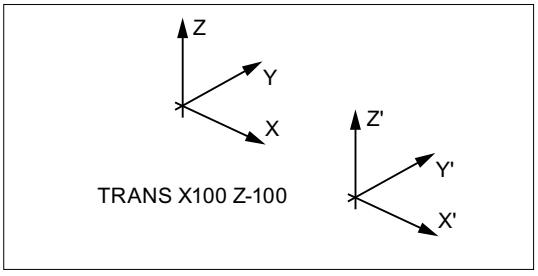
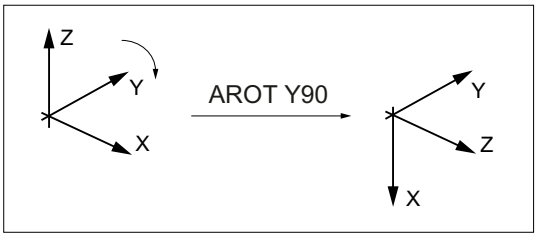
공구는 반드시 Z 축 방향으로 회전된 경사 표면에 수직으로 정렬되어야 합니다.

프로그램 코드	설명
N10 G17 G54	; 작업 평면 X/Y, 공작물 원점
N20 TRANS X10 Y10	; 절대치 옵셋
N30 L10	; 서브프로그램 호출
N40 ATRANS X35	; 추가 옵셋
N50 AROT Y30	; Y 축을 중심으로 추가 회전
N60 ATRANS X5	; 추가 옵셋
N70 L10	; 서브프로그램 호출
N80 G0 X300 Y100 M30	;후퇴, 프로그램 종료

예 3: 다중 단면 가공



이 예에서는 서브프로그램을 통해 서로 수직인 두 공작물 표면에 동일한 형태를 가공합니다. 우측 공작물 표면의 새 좌표계에서 절입 방향, 작업 평면 및 영점은 상단 표면을 기준으로 셋업되었습니다. 따라서 서브프로그램 수행에 필요한 조건이 그대로 적용됩니다. 작업 평면 G17, 좌표 평면 X/Y, 절입 방향 Z.

프로그램 코드	설명
N10 G17 G54	; 작업 평면 X/Y, 공작물 원점
N20 L10	; 서브프로그램 호출
N30 TRANS X100 Z-100	; WCS의 절대치 옵셋
	
N40 AROT Y90	; Y를 중심으로 WCS를 90° 추가 회전
	

프로그램 코드	설명
N50 AROT Z90	; Z를 중심으로 WCS를 90° 추가 회전
N60 L10	; 서브프로그램 호출
N70 G0 X300 Y100 M30	;후퇴, 프로그램 종료

추가 정보

활성 평면에서 회전

RPL=...를 사용하여 프로그래밍하면 WCS가 활성 평면과 수직인 축을 중심으로 회전합니다.

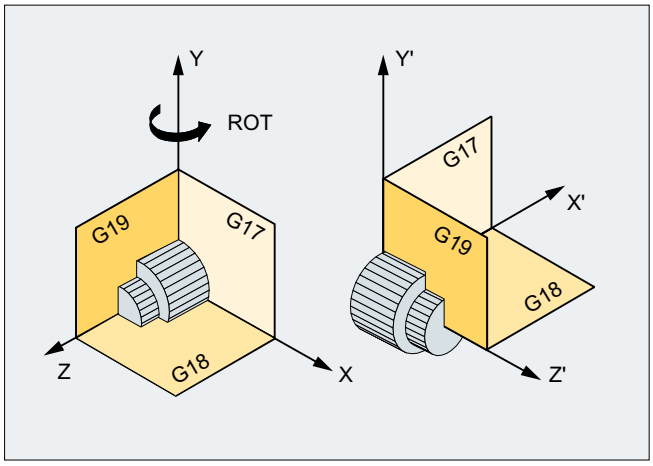


그림 3-23 Y 축 중심으로 회전 또는 G18 평면에서 회전

⚠ 경고

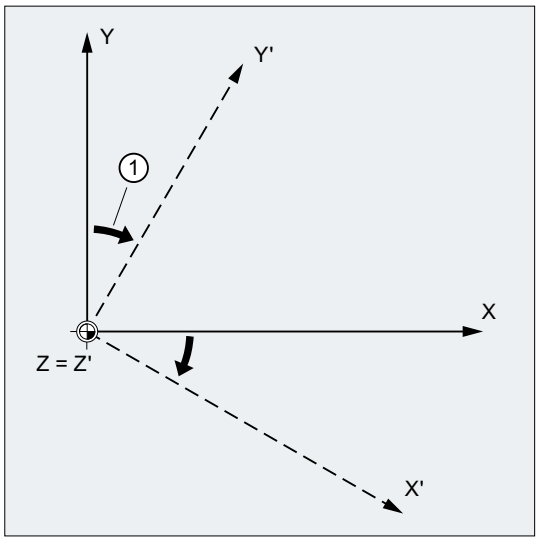
작업 평면 변경

회전 이후 평면을 변경 (G17, G18, G19) 하도록 프로그래밍한 경우 해당 축의 현재 회전 각도가 그대로 유지되어 새 평면에도 계속 적용됩니다. 따라서 평면을 변경하기 전에 현재 회전 각도를 0으로 리셋할 것을 적극 권장합니다.

- N100 ROT X0 Y0 Z0 ; 명시적 각도 프로그래밍
- N100 ROT ; 암시적 각도 프로그래밍

ROT X... Y... Z...를 이용한 절대 회전

지정된 축을 기준으로 프로그램된 회전 각도까지 WCS를 회전합니다.

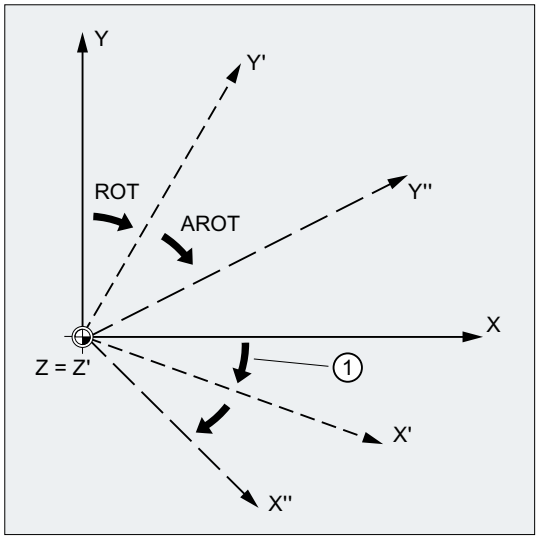


① 회전 각도

그림 3-24 Z 축을 중심으로 절대 회전

AROT X... Y... Z...를 이용한 추가 회전

지정된 축을 기준으로 프로그램된 회전 각도까지 WCS를 추가로 회전합니다.



① 회전 각도

그림 3-25 Z 축을 중심으로 절대 및 추가 회전

작업 평면 회전

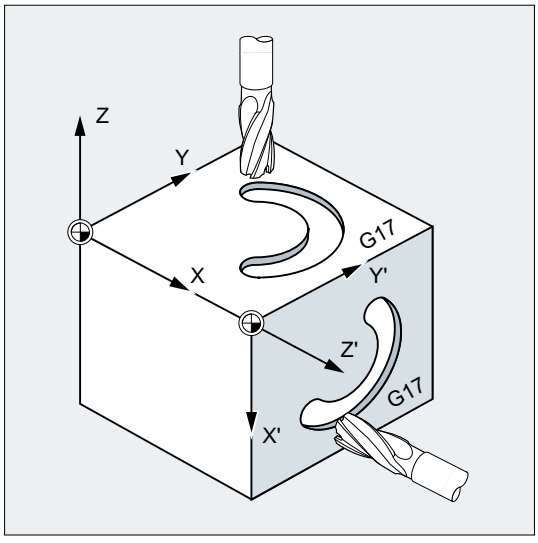
ROT/AROT를 사용하여 회전하면 작업 평면 (G17, G18, G19) 도 회전됩니다.

예: 작업 평면 G17

WCS가 공작물 상단 표면에 위치합니다. 옵셋과 회전을 이용하여 좌표계를 측면으로 이동시

킵니다. 작업 평면 G17 역시 회전합니다. 이 방법으로 X 및 Y를 통해 이송 모션을, Z를 통해 인피드를 G17 평면에 계속 프로그래밍할 수 있습니다.

요구 사항:
공구가 작업 평면과 수직이어야 하고, 절입 축 지점의 양의 방향이 공구 베이스 방향이어야 합니다.
CUT2DF를 지정하여 회전된 평면에서 공구 반경 보정을 활성화해야 합니다.



3.12.6 입체각으로 프로그래밍 프레임 회전(ROTS, AROTS, CROTS)

ROTS, AROTS 및 CROTS 구문을 사용하여 공작물 좌표계의 회전을 입체각으로 지정할 수 있습니다. 입체각은 아직 회전하지 않은 WCS의 메인 평면을 공간 상에서 회전시킨 평면의 교차점에 의해 생성되는 각도를 말합니다.

참고

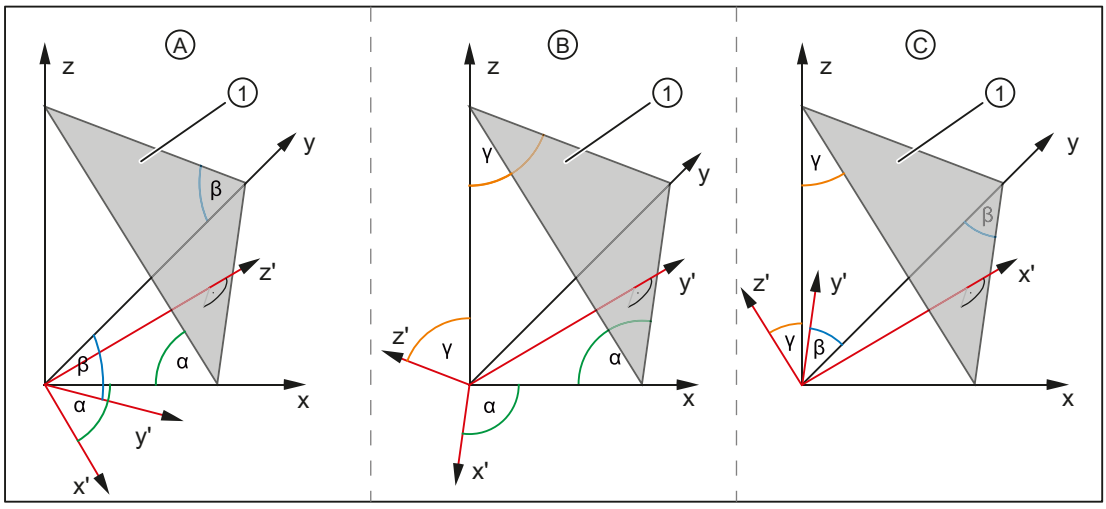
기하 축 이름

이후 설명을 위해 용어를 다음과 같이 정의합니다.

- 첫 번째 기하축: X
- 두 번째 기하 축: Y
- 세 번째 기하 축: Z

아래 그림과 같이 ROTS Xα Yβ를 프로그래밍하면 WCS의 G17 평면이 표시된 경사면과 평행하게 정렬됩니다. WCS 영점의 위치는 바뀌지 않습니다.

회전된 WCS의 오리엔테이션을 정의하여 첫 번째 회전된 축이 이 축과 원래 좌표계의 세 번째 축에 의해 생성된 평면에 위치할 수 있도록 합니다. 예의 경우: X'가 오리지널 X/Z 평면에 위치합니다.



① 경사 면

α, β, γ 입체각

A 경사면과 평행인 G17 평면의 정렬:

- 첫 번째 회전
  Y를 중심으로 각도 α만큼 X 회전 ⇒
  경사면과 평행인 X축
- 두 번째 회전
  X를 중심으로 각도 β만큼 Y 회전 ⇒
  경사면과 평행인 Y축
  ⇒ 경사면과 평행인 Z축
  ⇒ 경사면과 평행인 G17

B 경사면과 평행인 G18 평면의 정렬:

- 첫 번째 회전
  X를 중심으로 각도 γ만큼 Z 회전 ⇒
  경사면과 평행인 Z축
- 두 번째 회전
  Z를 중심으로 각도 α만큼 X 회전 ⇒
  경사면과 평행인 X축
  ⇒ 경사면과 평행인 Y축
  ⇒ 경사면과 평행인 G18

C 경사면과 평행인 G19 평면의 정렬:

- 첫 번째 회전
  Z를 중심으로 각도 β만큼 Y 회전 ⇒
  경사면과 평행인 Y축
- 두 번째 회전
  Y'를 중심으로 각도 γ만큼 Z 회전 ⇒
  경사면과 평행인 Z축
  ⇒ 경사면과 평행인 X축
  ⇒ 경사면과 평행인 G19

구문

요구 사항

공간에서 평면의 위치가 2개의 입체각을 통해 명확히 정의되어 있어야 합니다. 세 번째 입체각을 지정하면 평면이 "지나치게" 정의될 수 있습니다. 따라서 세 번째 입체각 지정은 허용되지 않습니다.

입체각이 1개만 프로그램된 경우 WCS의 회전이 `ROT` 및 `AROT` ("프로그램 가능한 회전 (ROT, AROT, RPL) (페이지 348)" 단원 참조) 와 동일해야 합니다.

프로그램된 두 축을 통해 `G17`, `G18`, `G19` 평면 정의에 따라 평면이 지정됩니다. 이에 따라 좌표 축의 순서 (평면의 첫 번째 축/두 번째 축) 또는 입체각을 통한 회전 순서가 결정됩니다.

평면	첫 번째 축	두 번째 축
G17	X	Y
G18	Z	X
G19	Y	Z

G17 평면 정렬 ⇒ X 및 Y에 대한 입체각

- 첫 번째 회전: Y를 중심으로 각도 α만큼 X 회전
- 두 번째 회전: X'를 중심으로 각도 β만큼 Y 회전
- 오리엔테이션: X'가 오리지널 Z/X 평면에 위치.

```
ROTS  X<α> Y<β>
AROTS X<α> Y<β>
CROTS X<α> Y<β>
```

G18 평면 정렬 ⇒ Z 및 X에 대한 입체각

- 첫 번째 회전: X를 중심으로 각도 γ만큼 Z 회전
- 두 번째 회전: Z'를 중심으로 각도 α만큼 X 회전
- 오리엔테이션: Z'가 오리지널 Y/Z 평면에 위치.

```
ROTS  Z<γ> X<α>
AROTS Z<γ> X<α>
CROTS Z<γ> X<α>
```

G19 평면 정렬 ⇒ Y 및 Z에 대한 입체각

- 첫 번째 회전: Z를 중심으로 각도 β만큼 Y 회전
- 두 번째 회전: Y'를 중심으로 각도 γ만큼 Z 회전
- 오리엔테이션: Y'가 오리지널 X/Z 평면에 위치.

```
ROTS  Y<β> Z<γ>
AROTS Y<β> Z<γ>
CROTS Y<β> Z<γ>
```

의미

ROTS:	입체각을 이용한 절대 프레임 회전, 기준 프레임: 프로그래밍 프레임 $P_PFRAME
AROTS:	입체각을 이용한 추가 프레임 회전, 기준 프레임: 프로그래밍 프레임 $P_PFRAME
CROTS:	입체각을 이용한 절대 프레임 회전, 기준 프레임: 프로그램된 $P_ ...
X, Y, Z:	기하 축 이름 (위의 참고 참조: 기하 축 이름)
A, β, γ:	각 기하 축에 대한 입체각: • α → X • β → Y • γ → Z

3.12.7 프로그래밍 배율 계수 (SCALE, ASCALE)

SCALE/ASCALE은 모든 경로 축, 동기 축 및 포지셔닝 축의 확대 또는 축소 스케일 팩터를 각 사례에서 지정된 축의 방향으로 프로그래밍하는 데 사용할 수 있습니다. 따라서 기하학적으로 유사한 형태 또는 상이한 수축 허용치를 프로그래밍 계산에 넣을 수 있습니다.

구문

```
SCALE X… Y… Z…
ASCALE X… Y… Z…
```

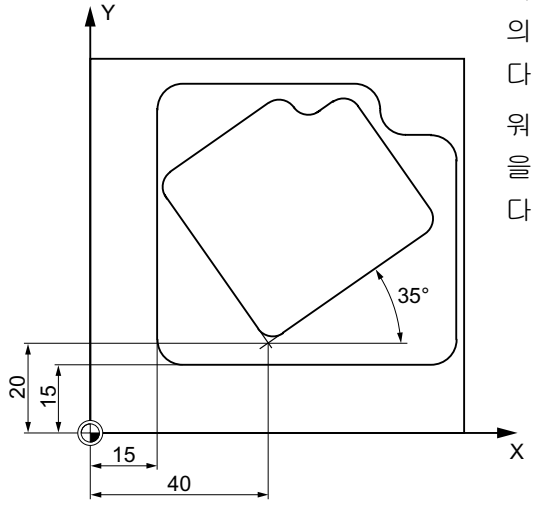
참고

각 프레임 작업을 서로 다른 NC 블록에 개별적으로 프로그래밍합니다.

의미

SCALE:	G54 ~ G57, G505 ~ G599로 설정된 현재 유효한 좌표계를 기준으로 한 배율 확대/수축 절대치
ASCALE:	현재 유효한 설정 좌표계 또는 프로그램된 좌표계를 기준으로 한 배율 확대/축소 추가치.
X… Y… Z…:	지정된 기하 축 방향의 배율 계수.

예



이 공작물에서 포켓이 두 번 발생하지만 하나의 포켓과 비교하여 크기 및 방향이 틀립니다. 가공 순서는 서브프로그램에 저장됩니다.

워크 옵셋 및 회전을 통해 필요한 공작물 원점을 설정하고 스케일링을 통해 형상을 축소한 다음 서브프로그램을 다시 호출합니다.

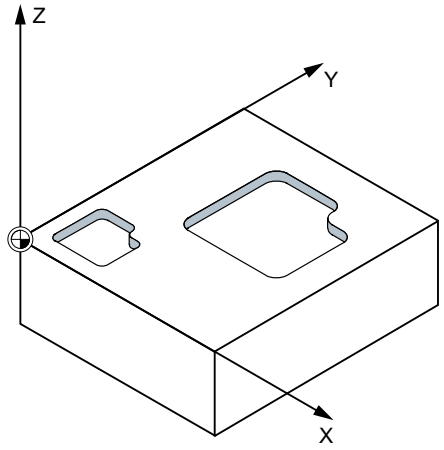
프로그램 코드	코멘트
N10 G17 G54	; 작업 평면 X/Y, 공작물 원점
N20 TRANS X15 Y15	; 절대치 옵셋
N30 L10	; 큰 포켓 가공
N40 TRANS X40 Y20	; 절대치 옵셋
N50 AROT RPL=35	; 평면에서 35° 회전
N60 ASCALE X0.7 Y0.7	; 작은 포켓의 배율 계수
N70 L10	; 작은 포켓 가공
N80G0 X300 Y100 M30	; 후퇴, 프로그램 종료

추가 정보

SCALE X... Y... Z...

각 축에 개별 스케일 팩터를 지정하여 형태가 축소되거나 확대됩니다. 스케일링은 G54 ~ G57, G505 ~ G599를 사용하여 설정된 공작물 좌표계를 기준으로 합니다.

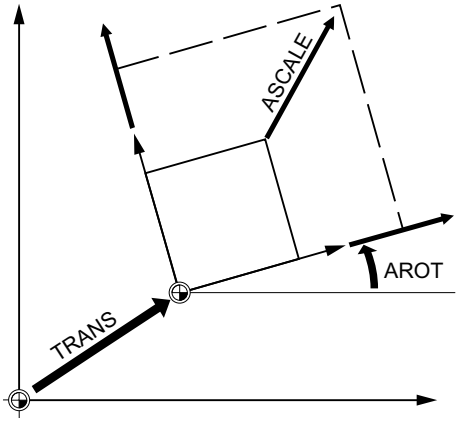
유의사항
오리지널 프레임 없음 SCALE 명령어는 이전에 활성화된 프로그램 프레임의 모든 프레임 요소를 리셋합니다.



ASCALE X... Y... Z...

ASCALE 명령은 기존의 프레임에 추가할 스케일 변경을 프로그래밍하는 데 사용됩니다. 이 경우 최종 유효한 스케일 팩터는 새 값으로 곱셈됩니다.

현재 설정되거나 또는 최종 프로그램된 좌표계가 스케일 변경에 대한 기준으로 사용됩니다.



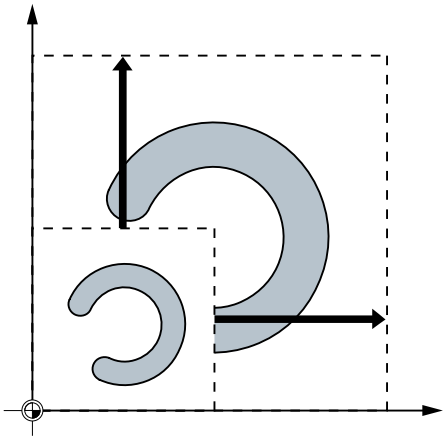
스케일링 및 옵셋

참고

SCALE 다음에 ATRANS로 옵셋이 프로그래밍되는 경우에는 옵셋 값도 스케일링됩니다.

다른 스케일 팩터

유의사항
충돌 위험 상이한 스케일 팩터를 사용할 때에는 각별히 주의하여야 합니다. 예를 들어, 원호 보간의 경우 반드시 동일한 스케일 팩터를 사용하여야 합니다.



참고

하지만 타원을 프로그래밍할 때는 다른 스케일 팩터를 사용할 수 있습니다.

3.12.8 프로그램 가능한 미러링 (MIRROR, AMIRROR)

MIRROR/AMIRROR는 좌표 축의 공작물 형태를 미러링하는 데 사용할 수 있습니다. 미러 호출 (예: 서브프로그램에서) 후에 프로그래밍된 모든 이송 동작은 미러링과 함께 실행됩니다.

구문

```
MIRROR X... Y... Z...
AMIRROR X... Y... Z...
```

참고

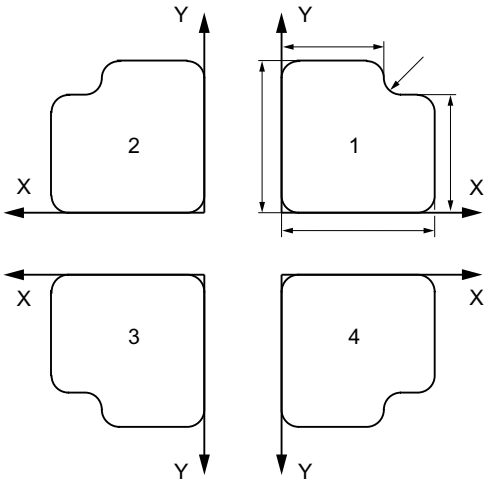
각 프레임 작업을 서로 다른 NC 블록에 개별적으로 프로그래밍합니다.

의미

MIRROR:	G54 ~ G57, G505 ~ G599로 설정된 현재 유효한 좌표계와 관련한 미러 절대치
AMIRROR:	현재 유효한 설정 좌표계 또는 프로그램된 좌표계를 기준으로 추가 미러 이미지.
X... Y... Z...:	방향이 변경될 기하 축. 여기에 지정된 값은 X0 Y0 Z0과 같이 자유롭게 선택할 수 있습니다.

예

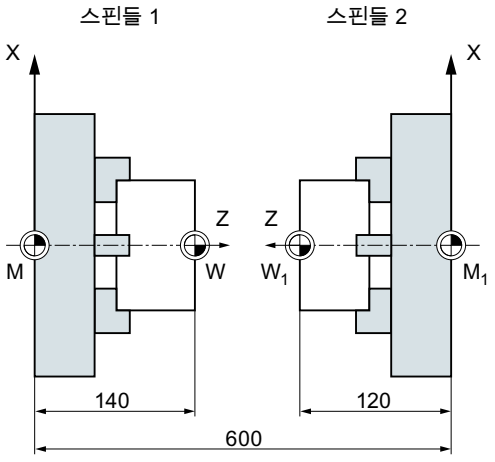
예 1: 밀링



여기에 표시된 형상은 서브프로그램으로서 한 번 프로그래밍된 것입니다. 나머지 3개 형상은 미러링을 통해 생성된 것입니다. 공작물 영점은 형상의 중심에 위치하게 됩니다.

프로그램 코드	코멘트
N10 G17 G54	; 작업 평면 X/Y, 공작물 원점
N20 L10	; 우측 상단에서 첫 번째 형상 가공
N30 MIRROR X0	; X축 미러링 (방향이 X에서 변경됨)
N40 L10	; 좌측 상단에서 두 번째 형상 가공
N50 AMIRROR Y0	; Y축 미러링 (방향이 Y에서 변경됨)
N60 L10	; 좌측 하단에서 세 번째 형상 가공
N70 MIRROR Y0	; MIRROR로 이전 프레임 리셋 Y축 미러링 (방향이 Y에서 변경됨)
N80 L10	; 우측 하단에서 네 번째 형상 가공
N90 MIRROR	; 미러링 비활성화
N100 G0 X300 Y100 M30	; 후퇴, 프로그램 종료

예 2: 선삭



실제 가공은 서브프로그램으로 보관되고 관련 스핀들에서의 실행은 미러링과 옵셋을 통해 이행됩니다.

프로그램 코드	설명
N10 TRANS X0 Z140	; W에 대한 워크 옵셋
...	; 스핀들 1로 첫 번째 측면 가공
N30 TRANS X0 Z600	; 스핀들 2에 대한 워크 옵셋
N40 AMIRROR Z0	; Z 축 미러링
N50 ATRANS Z120	; W1에 대한 워크 옵셋
...	; 스핀들 2로 두 번째 측면 가공

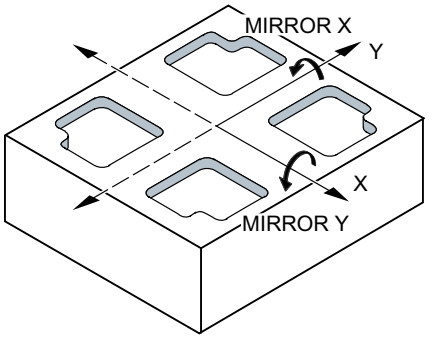
추가 정보

MIRROR X... Y... Z...

미러는 선택된 작업 평면에서 축 방향 변경을 사용하여 프로그램됩니다.

예: 작업 평면 G17 X/Y

미러 (Y축에서) 는 X에서의 방향 변경이 필요하며, 이에 따라 MIRROR X0을 사용하여 프로그래밍됩니다. 이 때 형상이 미러 축 Y의 반대 편에 미러됩니다.

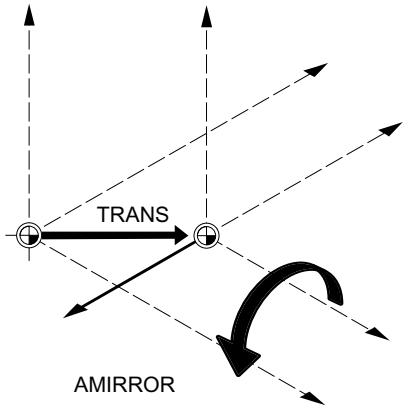


미러링은 G54 ~ G57, G505 ~ G599로 설정된 현재 유효한 좌표계와 관련하여 이행됩니다.

유의사항
오리지널 프레임 없음 MIRROR 명령어는 이전에 활성화된 프로그램 프레임의 모든 프레임 요소를 리셋합니다.

AMIRROR X... Y... Z...

기존 변환에 추가되어야 하는 미러 이미지는 AMIRROR를 사용하여 프로그래밍합니다. 현재 설정되거나 또는 최종 프로그램된 좌표계가 기준으로 사용됩니다.



미러링 해제

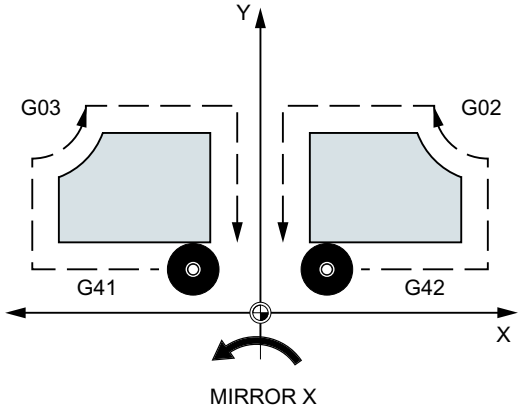
모든 축: MIRROR (축 파라미터 사용 안 함)

이전에 프로그래밍된 프레임의 모든 프레임 요소는 리셋됩니다.

공구 반경 보정

참고

미러 명령어를 사용하면 제어 시스템이 새로운 가공 방향에 따라 경로 보정 명령 (G41/G42 또는 G42/G41) 을 자동으로 변경합니다.



동일한 기능이 원호 회전 방향에도 적용됩니다 (G2/G3 또는 G3/G2).

참고

MIRROR 이후 AROT를 사용하여 추가 회전을 프로그래밍하는 경우 회전의 역방향 (양/음 또는 음/양) 을 사용해야 할 수도 있습니다. 형상 축에 대한 미러는 시스템에서 회전 및 머신 데이터에 지정된 미러 축에 대한 미러 (해당되는 경우) 로 자동 변환됩니다. 이는 셋터블 제로 옵셋에도 적용됩니다.

축 미러링

미러링할 축을 머신 데이터에 설정할 수 있습니다.

MD10610 $MN_MIRROR_REF_AX = <값>

값	의미
0	미러링은 프로그래밍된 축을 기준으로 합니다 (값의 부정).
1	기준 축은 X축입니다.
2	기준 축은 Y축입니다.
3	기준 축은 Z축입니다.

프로그래밍된 값 해석

머신 데이터는 프로그래밍된 값의 해석 방법을 지정하는 데 사용됩니다.

MD10612 $MN_MIRROR_TOGGLE = <값>

값	의미
0	프로그래밍된 축 값이 반영되지 않습니다.
1	프로그래밍된 축 값이 반영됩니다. • 프로그래밍된 축 값이 0이 아닌 경우 아직 미러링되지 않은 축이 미러링됩니다. • 프로그래밍된 축 값이 0이면 미러링이 비활성화됩니다.

3.12.9 공구 오리엔테이션에 따른 프레임 생성 (TOFRAME, TOROT, PAROT):

TOFRAME은 현재 공구 오리엔테이션과 일치하는 Z축을 가진 직사각형 프레임을 생성합니다. 따라서 사용자는 충돌의 염려 없이 Z 방향에서 공구를 후퇴시킬 수 있습니다 (예: 5축 프로그램에서 공구 파손 이후).

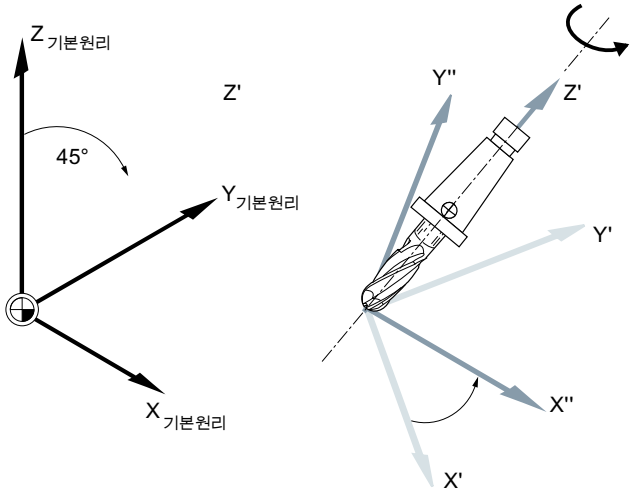
X축과 Y축의 위치는 머신 데이터 MD21110 $MC_X_AXES_IN_OLD_X_Z_PLANE (자동 프레임 정의된 좌표계) 의 설정에 따라 결정됩니다. 새 좌표계는 기계 운동학적으로 생성된 상태로 남거나 새로운 Z축을 중심으로 추가 회전하여 새로운 X축이 이전의 Z/X 면에 놓이게 됩니다 (장비 제조업체에서 제공한 사양서 참조).

오리엔테이션을 나타내는 결과 프레임이 프로그래밍 프레임 ($P_PFRAME) 의 시스템 변수에 기록됩니다.

TOROT만이 프로그래밍된 프레임에서 회전 구성 요소를 덮어씁니다. 다른 모든 구성 요소는 아무런 변경 없이 그대로 유지됩니다.

TOFRAME 및 TOROT은 G17 (작업 평면 X/Y) 이 일반적으로 활성 상태인 밀링 작업을 위한 것입니다. 그러나 선삭 작업 시 또는 G18이나 G19가 활성 상태일 때는 X축 또는 Y축이 공구 오리엔테이션과 일치하는 프레임이 필요합니다. 이 프레임은 TOFRAMEX/TOROTX 또는 TOFRAMEY/TOROTY 구문을 사용하여 프로그래밍합니다.

PAROT은 공작물에서 공작물 좌표계를 정렬합니다.



구문

```
TOFRAME/TOFRAMEZ/TOFRAMEY/TOFRAMEX
...
TOROTOF
```

```
TOROT/TOROTZ/TOROTY/TOROTX
...
TOROTOF
```

```
PAROT
...
PAROTOF
```

의미

TOFRAME:	프레임을 회전시켜 WCS의 Z축을 공구 오리엔테이션과 평행하게 정렬
TOFRAMEZ:	TOFRAME과 동일
TOFRAMEY:	프레임을 회전시켜 WCS의 Y축을 공구 오리엔테이션과 평행하게 정렬
TOFRAMEX:	프레임을 회전시켜 WCS의 X축을 공구 오리엔테이션과 평행하게 정렬
TOROT:	프레임을 회전시켜 WCS의 Z축을 공구 오리엔테이션과 평행하게 정렬 TOROT에 의해 정의된 회전은 TOFRAME으로 정의된 것과 동일합니다.

TOROTZ:	TOROT와 동일
TOROTY:	프레임을 회전시켜 WCS의 Y축을 공구 오리엔테이션과 평행하게 정렬
TOROTX:	프레임을 회전시켜 WCS의 X축을 공구 오리엔테이션과 평행하게 정렬
TOROTOF:	공구 방향과 평행한 오리엔테이션을 비활성화합니다.
PAROT:	프레임을 회전시켜 공작물 위에 공작물 좌표계를 정렬합니다. 활성 프레임의 변환, 스케일링 및 미러링은 그대로 유지됩니다.
PAROTOF:	PAROT로 활성화된 공작물 관련 프레임 회전은 PAROTOF로 비활성화됩니다.

참고

TOROT 구문을 사용하여 각 운동 유형에 맞게 오리엔테이션 가능한 활성 공구 홀더로 일관된 프로그래밍이 가능합니다.

회전 가능한 공구 홀더의 상황과 마찬가지로 PAROT는 작업 테이블의 회전을 활성화하는 데 사용할 수 있습니다. 프레임의 이러한 정의는 기계에 보정 이동이 수행되지 않는 방법으로 공작물 좌표계의 위치를 변경합니다. 오리엔테이션 조정이 가능한 공구 홀더가 활성화되어 있지 않으면 언어 명령어 PAROT는 거부되지 않습니다.

예

프로그램 코드	설명
N100 G0 G53 X100 Z100 D0	
N120 TOFRAME	
N140 G91 Z20	; TOFRAME가 계산에 포함되며, 프로그래밍된 모든 기하 축 이동은 새 좌표계를 기준으로 합니다.
N160 X50	
...	

추가 정보

축 방향 할당

TOFRAMEX, TOFRAMEY, TOROTX, TOROTY 구문 중 하나를 TOFRAME/TOFRAMEZ 또는 TOROT/TOROTZ 대신 프로그래밍한 경우에는 다음 표에 나오는 축 방향 구문이 적용됩니다.

구문	공구 방향 (직교)	두 번째 축 (가로축)	두 번째 축 (세로축)
TOFRAME / TOFRAMEZ / TOROT / TOROTZ	Z	X	Y
TOFRAMEY/TOROTY	Y	Z	X
TOFRAMEX/TOROTX	X	Y	Z

TOFRAME 또는 TOROT의 개별 시스템 프레임

TOFRAME 또는 TOROT로 만들어진 프레임을 개별 시스템 프레임$P_TOOLFRAME으로 작성할 수 있습니다. 이를 위해서는 머신 데이터 MD28082 $MC_MM_SYSTEM_FRAME_MASK에서 비트 3을 활성화해야 합니다. 프로그램 프레임은 변하지 않고 유지됩니다. 프로그램 프레임이 다른 곳에서 더 처리될 때 차이가 발생합니다.

3.12.10 프레임 (G53, G153, SUPA, G500) 선택 해제

공구 교환 위치로의 접근과 같은 특정 처리를 실행할 때 다양한 프레임 구성 요소가 각기 다르게 정의되고 억제되어야 합니다.

셋터블 프레임을 모달 또는 넌모달로 억제할 수 있습니다.

프로그래밍 프레임은 넌모달로 억제 또는 삭제할 수 있습니다.

구문

```
G53
G153
SUPA
G500
TRANS
ROT
SCALE
MIRROR
```

의미

G53:	모든 프로그래밍 프레임과 셋터블 프레임의 넌모달 억제
G153:	G153은 G53과 같은 효과를 가지며 전체 기본 프레임 ($P_ACTBFRAME) 을 억제함
SUPA:	SUPA는 G153과 같은 효능을 가지며 또한 다음을 억제합니다: • 핸드휠 옵셋 (DRF) • 오버레이 이동 • 외부 워크 옵셋 • PRESET 옵셋
G500:	만약 G500에 아무런 값이 설정되지 않은 경우 전체 셋터블 프레임
TRANS ROT SCALE MIRROR:	축 상세 정보가 없으면 프로그래밍 프레임이 삭제됩니다.

3.12.11 프로그래밍: Deselecting overlays axis-specifically (CORROF)

다음 축별 오버레이(오버라이드)가 CORROF 절차를 통해 삭제됩니다.

- 핸드휠 이송을 통해 설정된 추가 워크 옵셋(DRF 옵셋)
- $AA_OFF 시스템 변수를 통해 프로그램된 위치 옵셋

오버레이(오버라이드) 값 삭제를 통해 전처리 중지가 개시되고, 선택 해제된 오버레이 동작의 위치 구성요소가 기본 좌표계의 위치로 이송됩니다. 반면, 이송되는 축은 없습니다.

$AA_IM 시스템 변수를 통해 읽을 수 있는 위치 값(축의 현재 **MCS** 지령치)은 **기계**좌표계에서 **변경되지 않습니다.**

$AA_IW 시스템 변수를 통해 읽을 수 있는 위치 값(축의 현재 시스템 태그 **WCS** 설정점)은 선택 해제된 오버레이 동작 구성 요소를 포함하기 때문에 **공작물** 좌표계에서 **변경됩니다.**

참고

CORROF는 **NC 프로그램**에서 프로그래밍될 수 있습니다.
CORROF는 **동기화된 동작**으로 프로그래밍되어서는 **안** 됩니다.

구문

```
CORROF(<Axis>,"<String>"[,<Axis>,"<String>"])
```

의미

CORROF:	축의 다음 옵셋 및 오버레이 선택해제 절차: • DRF 옵셋 • 위치 옵셋($AA_OFF)	
	효율성:	모달
<Axis>:	축 이름 (채널 축, 기하 축 또는 기계 축 이름)	
	데이터 유형:	AXIS
<String>:	오버레이 유형 정의를 위한 문자열	
	데이터 유형:	BOOL
	값	**의미**
	DRF	DRF 옵셋
	AA_OFF	위치 옵셋($AA_OFF)

예

예 1: DRF 옵셋의 축별 선택해제 (1)

DRF 옵셋은 DRF 핸드휠 이송에 의해 X 축에서 생성됩니다. DRF 옵셋은 채널의 다른 축에는 적용되지 않습니다.

프로그램 코드	설명
N10 CORROF(X,"DRF")	; 여기서는 CORROF가 DRFOF와 같은 역할을 합니다.
...	

예 2: DRF 옵셋의 축별 선택해제 (2)

DRF 옵셋은 DRF 핸드휠 이송에 의해 X 축과 Y 축에서 생성됩니다. DRF 옵셋은 채널의 다른 축에는 적용되지 않습니다.

프로그램 코드	설명
; X 축의 DRF 옵셋만 선택해제되고 Y축의 DRF 옵셋은 유지됩니다.	
; DRFOF의 경우 두 옵셋 모두 선택해제되었을 것입니다.	
N10 CORROF(X,"DRF")	
...	

예 3: $AA_OFF 위치 옵셋의 축별 선택해제

프로그램 코드	설명
; 위치 옵셋이 10이면 X 축이 보간됩니다.	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; X축의 위치 옵셋이 선택해제됩니다: $AA_OFF[X]=0	
; X 축은 이송되지 않습니다.	
; X 축의 현재 위치에 위치 옵셋이 추가됩니다.	
N80 CORROF(X,"AA_OFF")	
...	

예 4: DRF 옵셋 및 $AA_OFF 위치 옵셋의 축별 선택해제(1)

DRF 옵셋은 DRF 핸드휠 이송에 의해 X 축에서 생성됩니다. DRF 옵셋은 채널의 다른 축에는 적용되지 않습니다.

프로그램 코드	설명
; 위치 옵셋이 10이면 X 축이 보간됩니다.	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; X 축의 DRF 옵셋과 위치 옵셋만 선택해제됩니다.	
; Y 축의 DRF 옵셋은 유지됩니다.	
N70 CORROF(X,"DRF",X,"AA_OFF")	
...	

예 5: DRF 옵셋 및 $AA_OFF 위치 옵셋의 축별 선택해제(2)

DRF 옵셋은 DRF 핸드휠 이송에 의해 X 축과 Y 축에서 생성됩니다. DRF 옵셋은 채널의 다른 축에는 적용되지 않습니다.

프로그램 코드	설명
; 위치 옵셋이 10이면 X 축이 보간됩니다.	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; Y축의 DRF 옵셋과 X축의 위치 옵셋이 선택해제됩니다.	
; X축의 DRF 옵셋은 유지됩니다.	
N70 CORROF(Y,"DRF",X,"AA_OFF")	
...	

추가 정보

$AA_OFF_VAL

$AA_OFF를 사용해 위치 옵셋을 선택해제하면, 해당 축의 시스템 변수 $AA_OFF_VAL(축 오버레이의 통합 거리)은 0이 됩니다.

JOG 모드에서 $AA_OFF

JOG 모드에서도 $AA_OFF가 변경된 경우, 머신 데이터 MD 36750 $MA_AA_OFF_MODE를 통해 이 기능이 활성화된 상태이면 위치 옵셋을 오버레이 이동으로 보간시킵니다.

동기 동작에서 $AA_OFF

CORROF(<axis>,"AA_OFF")를 사용해 위치 옵셋을 선택해제할 때 $AA_OFF(DO $AA_OFF[<axis>]=<value>)를 재설정하는 동기 동작이 활성화되면, $AA_OFF가 선택해제되지만 재설정되지는 않으며 알람 21660이 발생합니다. 그러나, 가령 CORROF 다음의 블록에서 동기 동작이 나중에 활성화되면 $AA_OFF는 설정대로 유지되고 위치 옵셋이 보간됩니다.

채널 축 자동 교체

다른 채널에서 활성인 축이 CORROF에 프로그래밍된 경우 축 교환을 통해 해당 축으로 불러오고(조건: MD30552 $MA_AUTO_GET_TYPE > 0) 위치 옵셋 및/또는 DRF 옵셋의 선택이 해제됩니다.

3.12.12 추가 워크 옵셋 선택해제(DRFROF)

핸드휠 이송을 통해 설정되는 추가 워크 옵셋(DRF 옵셋)은 DRFOF 절차를 통해 선택해제됩니다.

오버레이 값 선택해제를 통해 전처리 중지가 개시되고, 선택해제된 DRF 옵셋의 위치 구성 요소가 기본 좌표계의 위치로 이송되며 이 좌표계에서 축은 이송되지 않습니다. 시스템 변수 $AA_IM의 값(축의 현재 MCS 지령치)은 변경되지 않습니다. 시스템 변수 $AA_IW의 값(축의 현재 WCS 지령치)은 변경됩니다. $AA_IW 변수가 선택 해제된 오버레이 이동의 구성 요소를 포함하기 때문입니다.

구문

```
DRFOF
```

의미

DRFOF:	채널 내 모든 활성 축의 DRF 옵셋 선택해제 절차	
	효율성:	모달

예

예 1: DRF 옵셋의 축별 선택해제 (1)

DRF 옵셋은 DRF 핸드휠 이송에 의해 X 축에서 생성됩니다. DRF 옵셋은 채널의 다른 축에는 작동하지 않습니다.

프로그램 코드	코멘트
N10 CORROF(X,"DRF")	; 여기서는 CORROF가 DRFOF와 같은 역할을 합니다.
...	

예 2: DRF 옵셋의 축별 선택해제 (2)

DRF 옵셋은 DRF 핸드휠 이송에 의해 X 축과 Y 축에서 생성됩니다. DRF 옵셋은 채널의 다른 축에는 적용되지 않습니다.

프로그램 코드	코멘트
N10 CORROF(X,"DRF")	; X 축의 DRF 옵셋만 선택 해제되고 Y 축의 DRF 옵셋은 그대로 유지됩니다 (두 DRFOF 옵셋이 모두 선택 해제된 경우).
...	

예 3: DRF 옵셋 및 $AA_OFF 위치 옵셋의 축별 선택해제(1)

DRF 핸드휠 이송에 의해 X 축에서 DRF 옵셋이 생성됩니다. DRF 옵셋은 채널의 다른 축에는 적용되지 않습니다.

프로그램 코드	코멘트
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	; 위치 옵셋이 10이면 X 축이 보간됩니다.
...	
N70 CORROF(X,"DRF",X,"AA_OFF")	; X 축의 DRF 옵셋과 위치 옵셋만 선택 해제됩니다. Y 축의 DRF 옵셋은 그대로 유지됩니다.
...	

예 4: DRF 옵셋 및 $AA_OFF 위치 옵셋의 축별 선택해제(2)

DRF 핸드휠 이송에 의해 X 축과 Y 축에서 DRF 옵셋이 생성됩니다. DRF 옵셋은 채널의 다른 축에는 적용되지 않습니다.

프로그램 코드	코멘트
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	; 위치 옵셋이 10이면 X 축이 보간됩니다.
...	

프로그램 코드	코멘트
N70 CORROF(Y,"DRF",X,"AA_OFF")	; Y 축의 DRF 옵셋과 X 축의 위치 옵셋이 선택 해제됩니다. X 축의 DRF 옵셋은 그대로 유지됩니다.
...	

3.12.13 특정 연삭 워크 옵셋 (GFRAME0, GFRAME1 ... GFRAME100)

채널 내 연삭 프레임 활성화 명령어

GFRAME<n> 명령어의 프로그래밍은 $P_GFR[<n>] 데이터 관리의 관련 연삭 프레임이 채널 내에서 활성화되도록 해줍니다. 이는 활성 $P_GFRAME 연삭 프레임이 데이터 관리 $P_GFR[<n>] 연삭 프레임과 동일하도록 해줍니다:

GFRAME<n> ⇒ $P_GFRAME = $P_GFR[<n>]

명령	채널 내 활성 연삭 프레임
GFRAME0	$P_GFR[0] (**프레임 없음**)
GFRAME1	$P_GFR[1]
...	...
GFRAME100	$P_GFR[100]

구문

GFRAME<n>

의미

GFRAME<n>:	데이터 관리의 연삭 프레임 <n> 활성화	
	G 그룹:	64
	기본 위치:	MD20150 $MC_GCODE_RESET_VALUES[**63**]
	효율성:	모달
<n>:	연삭 프레임의 개수	
	값 범위:	0, 1, 2, ... 100

3.13 보조 기능 출력

기능

보조 기능 출력은 정보를 PLC로 전송함으로써 NC 프로그램이 언제 PLC가 기계 공구에서 세부적인 전환 작업을 수행하기를 원하는 지를 알려줍니다. 보조 기능은 해당 파라미터와 함께 PLC 인터페이스로 출력됩니다. 이송 값 및 신호는 PLC 사용자 프로그램에 의해 반드시 처리되어야 합니다.

보조 기능

다음 보조 기능이 PLC로 이송될 수 있습니다:

보조 기능	주소
공구 선택	T
공구 옵셋	D, DL
이송 속도	F/FA
스핀들 속도	S
M 코드	M
H 기능	H

각 기능 그룹이나 단일 기능의 경우 출력이 이송 동작 **이전**, **동시** 또는 **이후**실행될 지를 판정하는데 기계 데이터가 사용됩니다.

PLC는 다양한 방법으로 보조 기능 출력을 인식하도록 프로그램될 수 있습니다.

속성

보조 기능의 중요한속성은 다은의 전체보기 도표에 나와 있습니다:

기능	주소 확장		값			설명	블록 당 최대 수
	의미	범위	범위	종류	의미		
M	-	0 (함축적)	0 ... 99	INT	기능	주소 확장은 0에서 99까지 범위의 0 입니다. 어드레스 확장자가 없는 경우 필수: M0, M1, M2, M17, M30	5
	스핀들 번호	1 - 12	1 ... 99	INT	기능	어드레스 확장자로 스핀들 번호가 있는 M3, M4, M5, M19, M70 (예: `M2=5` 이면 스핀들 2를 정지). 스핀들 번호가 없으면 이 기능은 마스터 스핀들에 적용됩니다.	
	모두	0 - 99	100 ... 2147483647	INT	기능	사용자 M 기능*	
S	스핀들 번호	1 - 12	0 ... ± 1.8*10^{308}	REAL	속도	스핀들 번호가 없으면 이 기능은 마스터 스핀들에 적용됩니다.	3
H	모두	0 - 99	0 ... ± 2147483647 ± 1.8*10^{308}	INT REAL	모두	기능은 NC에서 아무런 영향이 없으며 PLC에서만 구현됩니다.*	3
T	스핀들 번호 (활성 공구 관리)	1 - 12	0 - 32000 (또는 활성 공구 관리의 공구 이름)	INT	공구 선택	공구 이름은 PLC 인터페이스에 전달되지 않습니다.	1
D	-	-	0 - 12	INT	공구 옵셋 선택	D0: 선택 취소 디폴트 설정: D1	1
DL	위치 의존 옵셋	1 - 6	0 ... ± 1.8*10^{308}	REAL	공구 미세 옵셋 선택	이전 선택 D 번호를 기준합니다.	1
F	-	-	0.001 - 999 999.999	REAL	경로 이송 속도		6

기능	주소 확장		값			설명	블록 당 최대 수
	의미	범위	범위	종류	의미		
FA	축 번호	1 - 31	0.001 - 999 999.999	REAL	축 이송 속도		
* 기계 제조업체가 기능의 의미를 정의합니다(**기계 제조업체 사양을 참조하시오**).							

추가 정보

NC 블록 당 기능 출력의 수

한 NC 블록에 10 개 기능 출력까지 프로그램할 수 있습니다. 보조 기능 역시 **동기화된 작용**의 작용 구성 요소에서 출력될 수 있습니다.

그룹화

설명된 기능은 함께 그룹화될 수 있습니다. 그룹 할당은 몇몇 M 명령어로 사전 정의됩니다. 행동 인식은 그룹화에 의해 정의될 수 있습니다.

고속 기능 출력(QU)

고속 출력으로 프로그램되지 않은 기능은 키워드QU를 사용하여 개별 출력의 고속 출력으로 정의될 수 있다. 기타 기능 인식(프로그램은 이동 인식을 대기합니다)을 대기하지 않고 프로그램은 계속 실행됩니다. 이는 이송 동작에의 불필요한 일시적 대기 지점과 중단을 방지하는 데 도움이 됩니다.

참고

"고속 기능 출력" 기능의 적절한 머신 데이터가 반드시 설정되어야 합니다(→ **기계 제조업체**).

이동 명령어의 기능 출력

정보의 이송과 적절한 반응 대기 시간이 소요되기 때문에 이송 동작에 영향을 줍니다.

블록 변경 지연 없는 고속 인식

블록 변경 동작은 머신 데이터의 영향을 받을 수 있습니다. "블록 변경 지연 없음" 설정이 선택되면 고속 보조 기능에 관련된 시스템 반응은 다음과 같습니다:

보조 기능 출력	응답
이동 **이전**	고속 보조 기능으로 블록 간의 블록 전환이 **방해 없이** 그리고 **속도의 감소 없이** 발생합니다. 보조 기능 출력은 블록의 첫 번째 보간 클럭 싸이클에서 발생합니다. 인식 지연 없이 그 다음 블록이 실행됩니다.
이동 **동안**	고속 보조 기능으로 블록 간의 블록 전환이 **방해 없이** 그리고 **속도의 감소 없이** 발생합니다. 보조 기능 출력이 블록 중간에 발생합니다. 인식 지연 없이 그 다음 블록이 실행됩니다.
이동 **이후**	블록 끝에서 이동이 중지합니다. 보조 기능 출력은 블록의 끝에서 발생합니다. 인식 지연 없이 그 다음 블록이 실행됩니다.

⚠ 주의

연속 경로 모드에서 기능 출력

이송 모션이 연속 경로 모드(G64/G641)를 방해하고 이전 블록을 위한 정확한 정지를 생성하기 **전**의 기능 출력

이송 동작이 연속 경로 모드(G64/G641)를 방해하고 현재 블록의 정확한 정지를 생생한 **후**의 기능 출력

중요: PLC로부터 대기 중인 인식 신호를 대기하면 경로 길이가 매우 짧은 블록에서 M 명령 순서의 예처럼 연속 경로 모드가 방해될 수 있습니다.

3.13.1 M 코드

M 코드는 "절삭유 ON/OFF"나 다른 기계 기능을 제어합니다.

구문

```
M<value>
M[<address extension>] = <value>
```

의미

M:	M 코드 번호	
<address extension>:	몇몇 M 코드는 확장된 어드레스가 적용됩니다 (예, 스핀들 기능에 스핀들 번호 지정).	
<value>:	특정 기계 기능에 M 코드 번호가 할당됩니다.	
	유형:	INT
	값 범위:	0 ... 2147483647 (최대 INT 값)

지정된 M 코드

프로그램 실행에 중요한 M 코드들이 표준으로 제공됩니다.

M 코드	의미
M0*	프로그램 정지
M1*	선택적 정지
M2*	프로그램 종료, 메인 프로그램 (M30)
M3	CW 방향 스핀들 회전
M4	스핀들 CCW 방향
M5	스핀들 정지
M6	공구 교환 (표준 설정)
M17*	서브프로그램 종료
M19	스핀들 포지셔닝
M30*	프로그램 종료, 메인 프로그램 (M2)
M40	기어 자동 변경
M41	기어 1단
M42	기어 2단
M43	기어 3단
M44	기어 4단
M45	기어 5단
M70	스핀들 축 모드 전환

참고

별표 (*) 표시의 기능에는 확장된 어드레스를 사용할 수 없습니다.

M0, M1, M2, M17 및 M30 코드는 항상 이송 동작 **이후에** 트리거됩니다.

M 코드는 장비 제조업체에서 지정합니다.

클램핑 기기 또는 추가 기계 기능의 활성화 및 비활성화 전환 기능의 예처럼, 모든 미 지정 M 코드 번호를 장비 제조업체가 사용할 수 있습니다.

참고

미 지정 M 코드 번호에 할당된 기능은 해당 기계의 고유 기능입니다. 따라서, M 코드는 기계에 따라 상이한 기능성을 가질 수 있습니다.

기계와 이의 기능에 유효한 M 코드는 장비 제조업체 사양을 참조 바랍니다.

예

예 1: 블록 내의 최대 M 코드 개수

프로그램 코드	코멘트
N10 S...	
N20 X... M3	; 축 이동이 포함된 블록의 M 코드,
	; X 축 이동 전에 스핀들이 가속됩니다.
N180 M789 M1767 M100 M102 M376	; 한 블록 내 최대 5 개의 M 코드 가능.

예 2: 고속 출력 M 코드

프로그램 코드	코멘트
N10 H=QU(735)	; H735를 고속 출력.
N10 G1 F300 X10 Y20 G64	
N20 X8 Y90 M=QU(7)	; M7을 고속 출력.

M7이 고속 출력으로 프로그래밍되어 연속 경로 모드 (G64) 를 방해하지 않습니다.

참고

다른 기능 출력과 특별한 경우에만 이 기능을 사용하십시오.

사전 정의된 M 명령어에 대한 추가 정보

프로그램 정지: M0

M0으로 NC 블록에서 기계가공이 정지합니다. 부스러기를 제거하고 측정 등을 할 수 있습니다.

프로그램 정지 1 - 선택적 정지: M1

M1은 다음과 같은 방법으로 설정할 수 있습니다.

- HMI / 대화 상자 "프로그램 제어"
  또는
- NC/PLC 인터페이스

NC의 프로그램 실행은 프로그램된 블록에 의해 정지됩니다.

프로그램된 정지 2 - 프로그램 실행 정지의 M1과 연관된 보조 기능

프로그램된 정지 2 는 HMI/대화 상자 "프로그램 제어"를 통해 설정될 수 있고 기계 가공될 부품의 끝 어느 시점에서도 기술적 시퀀스가 중지되도록 합니다. 이러한 방법으로 운전자가 부스러기 흐름 제거 같은 작업을 중지할 수 있습니다.

프로그램 종료: M2, M17, M30

프로그램은 M2, M17 또는 M30으로 종료됩니다. 서브프로그램과 같은 다른 프로그램에서 메인 프로그램을 호출하면 M2/M30가 M17과 동일한 효과를 가지며 그 반대의 경우, 즉 M17이 메인 프로그램에서 M2/M30과 동일한 효과를 가집니다.

스핀들 기능: M3, M4, M5, M19, M70

스핀들 번호 지정을 통한 확장 어드레스 표기법이 모든 스핀들에 적용됩니다.

예:

프로그램 코드	코멘트
M2=3	; 두 번째 스핀들의 CW 방향 스핀들 회전

어드레스 확장이 프로그램되지 않았다면 이 기능은 마스터 스핀들에 적용됩니다.

3.14 보조 명령어

3.14.1 메시지 출력 (MSG)

MSG() 명령문을 사용하여 가공 프로그램의 문자열을 메시지로 출력하여 작업자에게 보여줄 수 있습니다.

구문

```
MSG("<Message text>"[,<Execution>])
...
MSG ()
```

의미

MSG:	메시지 출력을 위해 사전 정의된 서브프로그램 호출		
<message text>:	메시지로 표시할 문자열		
	유형:	STRING	
	최대 길이:	124자, 1줄당 62자씩, 2줄까지 표시할 수 있습니다.	
	또한, 연결 연산자 '<<'를 사용하면 변수도 메시지 텍스트에 출력할 수 있습니다.		
<Execution>:	메시지가 작성되는 시간을 정의하는 파라미터 (옵션)		
	유형:	INT	
	값:	0 (기본 설정)	메시지를 작성하기 위한 전용 메인 실행 블록을 생성하지 않습니다. 메시지 작성은 실행 가능한 다음 NC 블록에서 수행됩니다. 실행 중인 연속 경로 모드는 중단되지 않습니다.
		1	메시지를 작성하기 위한 전용 메인 실행 블록을 생성합니다. 실행 중인 연속 경로 모드가 중단됩니다.
MSG():	메시지 텍스트 없이 MSG()를 프로그래밍하여 실제 메시지를 삭제할 수 있습니다. 삭제하지 않으면 다음 메시지가 나타날 때까지 디스플레이가 유지됩니다.		

참고

메시지를 사용자 인터페이스에서 사용되는 언어로 출력하려면 HMI에 설정된 언어에 대한 정보가 필요합니다. 이 정보는 가공 프로그램 및 동기 동작에서 시스템 변수 $AN_LANGUAGE_ON_HMI (페이지 1541)로 문의할 수 있습니다.

예

예 1: 메시지 출력/삭제

프로그램 코드	설명
N10 G91 G64 F100	; 연속 경로 모드
N20 X1 Y1	
N... X... Y...	
N20 MSG ("가공 부품 1")	; N30에서 메시지를 먼저 출력합니다.
	; 연속 경로 모드가 유지됩니다.
N30 X... Y...	
N... X... Y...	
N400 X1 Y1	
N410 MSG ("가공 부품2",1)	; N410에서 메시지를 출력합니다.
	; 연속 경로 모드가 중단됩니다.
N420 X1 Y1	
N... X... Y...	
N900 MSG ()	; 메시지를 삭제합니다.

예 2: 변수가 포함된 메시지 텍스트

프로그램 코드	설명
N10 R12=$AA_IW [X]	; R12에서 X 축의 실제 위치
N20 MSG ("Check position of X axis"<<R12<<)	; 변수 R12가 포함된 메시지를 출력합니다.
...	
N90 MSG ()	; N20에서 메시지를 삭제합니다.

3.14.2 OPI 변수 (WRTPR) 에 문자열 쓰기

WRTPR() 함수를 사용하면 가공 프로그램에서 OPI 변수 progProtText에 문자열을 쓸 수 있습니다.

구문

```
WRTPR(<String>[,<ExecTime>])
```

의미

WRTPR:	문자열 출력을 위한 기능 호출.		
<String>:	OPI 변수 progProtText에 쓰기가 가능한 문자열		
	유형:	STRING	
	최대 길이:	128 문자	
<ExecTime>:	문자열 쓰기에 걸리는 시간을 정의하는 옵션 파라미터		
	유형:	INT	
	값 범위:	0, 1	
		0 (디폴트 설정)	문자열을 쓰기 위한 전용 메인 실행 블록이 생성되지 않습니다. 대신 실행 가능한 다음 NC 블록에 문자열 쓰기를 실행합니다. 실행 중인 연속 경로 모드는 중단되지 않습니다.
		1	문자열을 쓰기 위한 전용 메인 실행 블록이 생성됩니다. 실행 중인 연속 경로 모드가 중단됩니다.

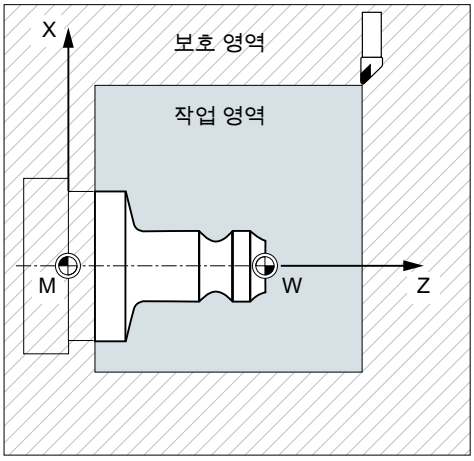
예

프로그램 코드	**코멘트**
N10 G91 G64 F100	; 연속 경로 모드
N20 X1 Y1	
N30 WRTPR("N30")	; 문자열 "N30"이 먼저 N40에 쓰여집니다.
	; 연속 경로 모드는 유지됩니다.
N40 X1 Y1	
N50 WRTPR("N50",1)	; 문자열 "N50"이 N50에 쓰여집니다.
	; 연속 경로 모드가 중단됩니다.
N60 X1 Y1	

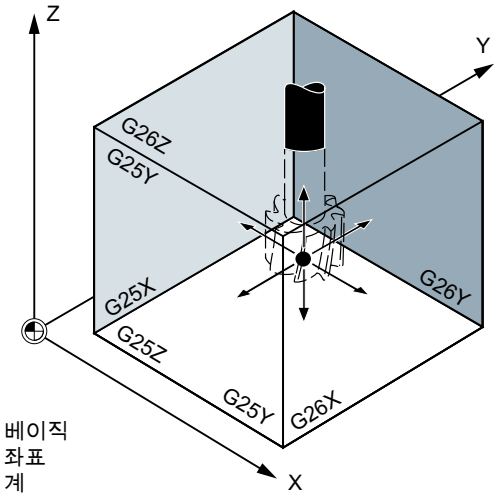
3.14.3 작업 영역 제한

3.14.3.1 BCS (G25/G26, WALIMON, WALIMOF)에서의 작업 영역 제한

G25/G26는 공구가 이송될 수 있는 작업 영역 (작업 공간) 을 제한합니다. G25/G26로 정의된 작업 영역 밖의 영역에서는 모든 공구의 동작이 금지됩니다.



개별 축의 좌표는 기본 좌표계로 적용됩니다.



유효한 모든 축의 작업 영역 제한은 반드시 WALIMON명령어로 프로그램되어야 합니다. WALIMOF명령어는 작업 영역 제한을 취소합니다. WALIMON 은 디폴트 설정입니다. 그러므로, 사전에 작업 영역 제한이 가능하게 되었을 때만 프로그램되어야 합니다.

구문

```
G25 X…Y…Z…
G26 X…Y…Z…
WALIMON
...
WALIMOF
```

의미

G25:	**하위** 작업 영역 제한 기본 좌표계에서 채널 축에 값을 지정합니다.
G26:	**상위** 작업 영역 제한 기본 좌표계에서 채널 축에 값을 지정합니다.
X… Y… Z…:	개별 채널 축에 대한 하위 또는 상위 작업 영역 제한 제한치는 기본 좌표계를 기준으로 입력합니다.
WALIMON:	모든 축의 작업 영역 제한 기능 **활성화**
WALIMOF:	모든 축의 작업 영역 제한 기능**취소**

G25/G26을 사용하는 프로그래밍에 추가하여, 값은 또한 축 지정 셋팅 데이터를 사용하여 값을 입력할 수 있습니다.

SD43420 $SA_WORKAREA_LIMIT_PLUS (작업 영역 제한 플러스)

SD43430 $SA_WORKAREA_LIMIT_MINUS (작업 영역 제한 마이너스)

SD43420 및 SD43430을 사용한 작업 영역 제한 활성화 및 비활성화는 축 셋팅 데이터에 따라 지정된 방향으로 즉각 실행됩니다.

SD43400 $SA_WORKAREA_PLUS_ENABLE (플러스 방향 작업 영역 제한)

SD43410 $SA_WORKAREA_MINUS_ENABLE (마이너스 방향 작업 영역 제한)

방향 활성화/비활성화를 통해 축의 작업 범위를 한 방향으로 제한할 수 있습니다.

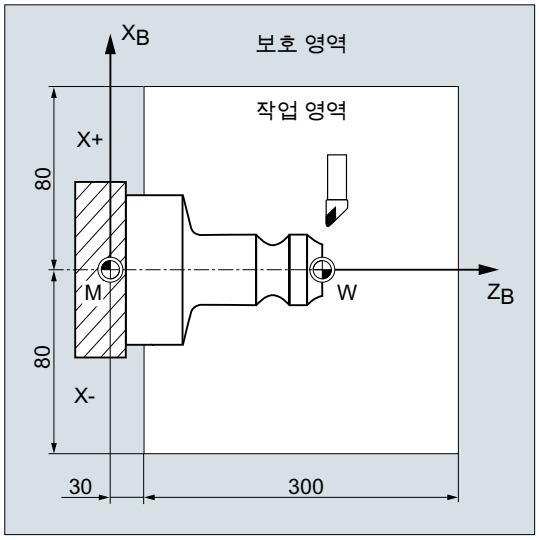
참고

G25/G26으로 프로그램된 작업 영역 제한이 우선권을 가지며, SD43420 과 SD43430이 입력된 값을 변경합니다

참고

G25/G26 은 어드레스 S"에서의 스핀들 속도 제한을 프로그램하는데 사용될 수 있습니다. 좀 더 상세한 정보는 "프로그램 가능한 스핀들 속도 제한 (G25, G26) (페이지 116)"를 참조하십시오.

예



작업 영역 제한 코드 G25/26으로 선반의 작업 영역을 제한하여 터렛, 측정 스테이션 같은 주변 디바이스와 장비를 보호할 수 있습니다.

디폴트 설정: WALIMON

프로그램 코드	설명
N10 G0 G90 F0.5 T1	
N20 G25 X-80 Z30	; 개별 좌표 축의 하한값 정의
N30 G26 X80 Z330	; 상한값 정의
N40 L22	; 절삭 프로그램
N50 G0 G90 Z102 T2	; 공구 교환 위치로 이동
N60 X0	
N70 WALIMOF	; 작업 영역 제한 비활성화
N80 G1 Z-2 F0.5	;드릴
N90 G0 Z200	;복귀
N100 WALIMON	; 작업 영역 제한 실행
N110 X70 M30	; 프로그램 종료

추가 정보

공구의 원점

공구 길이 옵셋이 활성이면 공구 팁을 원점으로 모니터링합니다. 그렇지 않으면 공구 홀더가 원점이 됩니다.

공구 반경 관련 사항이 별도로 활성화되어야 합니다. 이는 채널 지정 머신 데이터를 사용하여 이루어 집니다:

MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS

공구 원점이 작업 영역 제한으로 정의된 작업 영역 밖에 있거나 이 영역에서 나가면 프로그램 시퀀스는 중지됩니다.

참고

변환이 활성인 경우 적용된 공구 데이터 (공구 길이와 공구 반경) 가 설명한 동작과 다를 수 있습니다.

프로그램 가능한 작업 영역 제한, G25/G26

각 축에 상부 (G26) 와 하부 (G25) 작업 영역 제한이 정의될 수 있습니다. 이들 값은 즉시 유효해지고, RESET 및 전원을 켠 후에는 MD10710 $MD_PROG_SD_RESET_SAVE_TAB과 일치하게 됩니다.

참고

사전 정의된 CALCPOSI (페이지 654) 태그 기능을 사용해 예정된 경로의 작업 영역 제한 및/또는 보호 영역의 통과 여부를 점검할 수 있습니다.

3.14.3.2 WCS/SZS (WALCS0 ... WALCS10)에서의 작업 영역 제한

"WCS/ENS에서 작업 영역 제한"은 공작물 좌표계 (WCS) 또는 셋터블 제로 시스템 (SZS) 내에서 채널 축의 이송 범위를 특정 공작물에 대해 자유롭게 제한할 수 있도록 해줍니다. 재래식 선반에 주로 사용되는 기능입니다.

요구사항

채널 축이 원점 복귀된 상태여야 합니다.

작업 영역 제한 그룹

축 지정을 변경할 때, 예를 들어 변환 또는 활성 프레임을 ON/OFF 할 때 작업 영역 제한 그룹을 사용하면 모든 축 채널에 대해 축 작업 영역 제한을 다시 설정할 필요가 없습니다.

작업 영역 제한 그룹은 다음 데이터로 구성됩니다.

- 모든 채널 축의 작업 영역 제한
- 작업 영역 제한의 기준 좌표계

구문

```
...
$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<Value>
$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<Ax>]=<Value>
$P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<Ax>]=<Value>
$P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<Ax>]=<Value>
$P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<Ax>]=<Value>
...
WALCS<n>
...
WALCS0
```

의미

$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<Value>		
작업 영역 제한 그룹의 기준이 되는 좌표계		
<WALimNo>:	작업 영역 제한 그룹	
	유형:	INT
	값 범위:	0 (그룹 1) ... 9 (그룹 10)
<값>:	INT 유형의 값	
	1	공작물 좌표계 (WCS)
	3	셋터블 제로 시스템 (SZS)

$$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<Ax>]=<Value>		
지정된 채널 축의 **양의** 축 방향에서 작업 영역 제한 인에이블		
<WALimNo>:	작업 영역 제한 그룹	
	유형:	INT
	값 범위:	0 (그룹 1) ... 9 (그룹 10)
<Ax>:	채널 축 이름	
<값>:	BOOL 유형의 값	
	0 (FALSE)	비가공 상태
	1 (TRUE)	가공 상태

$P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<Ax>]=<Value>		
지정된 채널 축의 **음의** 축 방향에서 작업 영역 제한 인에이블		
<WALimNo>:	작업 영역 제한 그룹	
	유형:	INT
	값 범위:	0 (그룹 1) ... 9 (그룹 10)
<Ax>:	채널 축 이름	
<값>:	BOOL 유형의 값	
	0 (FALSE)	릴리즈 안됨
	1 (TRUE)	인에이블

$P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<Ax>]=<Value>		
지정된 채널 축의 **양의** 방향에서 작업 영역 제한		
<WALimNo>:	작업 영역 제한 그룹	
	유형:	INT
	값 범위:	0 (그룹 1) ... 9 (그룹 10)
<Ax>:	채널 축 이름	
<값>:	REAL 유형의 값	

$P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<Ax>]=<Value>		
지정된 채널 축의 **음의** 방향에서 작업 영역 제한		
<WALimNo>:	작업 영역 제한 그룹	
	유형:	INT
	값 범위:	0 (그룹 1) ... 9 (그룹 10)
<Ax>:	채널 축 이름	
<값>:	REAL 유형의 값	

WALCS<n>:	작업 영역 제한 그룹의 작업 영역 제한 활성화	
<n>:	작업 영역 제한 그룹의 번호	
	값 범위:	1 ... 10

WALCS0:	채널에 활성인 작업 영역 제한 비활성화

참고

실제 사용 가능한 작업 영역 제한 그룹의 수는 구성에 따라 달라집니다 (→ 장비 제조업체의 설명 참조).

예

채널 내에 3 개의 축이 정의됩니다: X, Y 및 Z

작업 영역 제한 그룹 번호 2 가 정의 및 활성화됩니다. 다음과 같이 축의 이동이 WCS 내에서 제한됩니다.

- X 축 방향 플러스: 10 mm
- X 축 방향 마이너스: 제한 없음
- Y 축 방향 플러스: 34 mm
- 마이너스 방향의 Y축: -25mm
- Z 축 방향 플러스: 제한 없음
- 마이너스 방향의 Z축: -600mm

프로그램 코드	코멘트
...	
N51 $P_WORKAREA_CS_COORD_SYSTEM[1]=1	; WCS에서 작업 영역 제한 그룹 2의 작업 영역 제한이 적용됩니다.
N60 $P_WORKAREA_CS_PLUS_ENABLE[1,X]=TRUE	
N61 $P_WORKAREA_CS_LIMIT_PLUS[1,X]=10	
N62 $P_WORKAREA_CS_MINUS_ENABLE[1,X]=FALSE	
N70 $P_WORKAREA_CS_PLUS_ENABLE[1,Y]=TRUE	
N73 $P_WORKAREA_CS_LIMIT_PLUS[1,Y]=34	
N72 $P_WORKAREA_CS_MINUS_ENABLE[1,Y]=TRUE	
N73 $P_WORKAREA_CS_LIMIT_PLUS[1,Y]=-25	
N80 $P_WORKAREA_CS_PLUS_ENABLE[1,Y]=TRUE	
N82 $P_WORKAREA_CS_MINUS_ENABLE[1,Y]=TRUE	
N83 $P_WORKAREA_CS_LIMIT_PLUS[1,Y]=-600	
...	
N90 WALCS2	; 작업 영역 제한 그룹 2를 활성화합니다.
...	

추가 정보

유효성

WALCS1-WALCS10을 사용한 작업 영역 제한은 WALIMON으로 작업 영역 제한과는 독립적으로 작용합니다. 두 기능을 동시에 사용하면 축 이동 시 먼저 도착하는 한계가 유효합니다.

공구에서의 원점

공구 데이터 (공구 길이와 공구 반경) 를 계산하고 작업 영역 제한을 모니터할 때, 공구에서의 원점은 WALIMON으로의 작업 영역 제한 동작과 일치하게 됩니다.

3.14.4 원점 복귀(G74)

기계에 전원 투입 시 (증분 측정 시스템 사용 경우) 모든 축은 원점 복귀 작업이 필요합니다. 이 때에는 프로그램 된 동작만 이송할 수 있습니다.

NC 프로그램에서는 G74를 사용하여 원점 복귀할 수 있습니다.

구문

```
G74 X1=0 Y1=0 Z1=0 A1=0 …; 별도의 NC 블록에 프로그래밍
```

의미

G74:	G 코드 호출 기준점 접근
X1=0 Y1=0 Z1=0 …:	**직선 축**에 지정된 기계 축 어드레스 X1, Y1, Z1 …을 원점으로 간주하고 이 지점으로 복귀합니다.
A1=0 B1=0 C1=0 …:	**로터리 축**에 지정된 기계 축 어드레스 A1, B1, C1 …을 원점으로 간주하고 이 지점으로 복귀합니다.

참고

G74로 원점 복귀하는 축에는 변환을 프로그래밍해서는 안됩니다.
변환은 TRAFOOF 명령으로 취소합니다.

예

측정 시스템이 변경되면 원점으로 복귀하고 공작물 원점이 초기화됩니다.

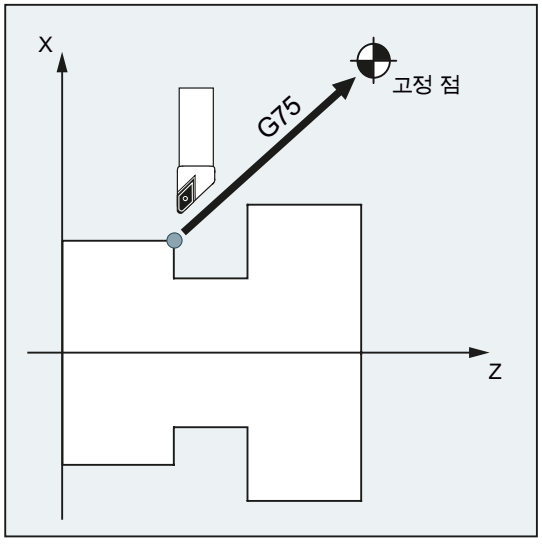
프로그램 코드	설명
N10 SPOS=0	; 위치 제어 상태의 스핀들
N20 G74 X1=0 Y1=0 Z1=0 C1=0	; 직선 축과 로터리 축의 원점 복귀
N30 G54	; 워크 옵셋
N40 L47	; 절삭 프로그램
N50 M30	; 프로그램 종료

3.14.5 고정 지점 접근(G75)

넌모달 명령어 G75를 사용해 축을 개별적으로 그리고 다른 축과 독립적으로 기계 공간 내의 고정 정지점(예: 공구 교환 지점, 로딩 지점, 팔레트 변경 지점) 등으로 이동할 수 있습니다.

고정 정지점은 머신 데이터에 저장된 기계 좌표계의 위치입니다 (MD30600 $MA_FIX_POINT_POS[n]). 각 축에 최대 4 개의 고정 정지점이 정의될 수 있습니다.

고정 정지점은 현재 공구 또는 공작물 위치와 무관하게 모든 NC 프로그램에서 접근할 수 있습니다. 축을 이동하기 전에 내부 전처리 정지가 실행됩니다.



요구사항

G75로 고정 정지점에 접근하려면 다음 요구 사항을 반드시 충족해야 합니다.

- 고정 정지점 좌표가 정확히 계산되고 머신 데이터에 저장되어 있어야 합니다.
- 고정 정지점이 유효한 이송 범위 내에 있어야 합니다(소프트웨어 리미트 제한 범위를 고려할 것).
- 이송될 축은 반드시 원점 복귀가 되어있어야 합니다.
- 모든 공구 반경 보정이 취소 상태여야 합니다.
- 좌표계 트렌스포메이션은 취소되어야 합니다.
- 이송할 축 중에 트렌스포메이션이 활성화된 축이 없어야 합니다.
- 이송할 축 중에 활성 커플링의 추종 축이 없어야 합니다.
- 이송할 축 중에 갠트리 그룹에 포함된 축이 없어야 합니다.
- 컴파일 사이클이 모션 구성 요소를 활성화하지 않아야 합니다.

구문

```
G75 <axis name><axis position> ... FP=<n>
```

의미

G75:	고정 정지점 접근		
<axis name>:	고정 정지점에 이송될 기계축의 이름 모든 축의 이름이 기입 가능합니다.		
<axis position>:	위치 값은 일반적으로 "0"으로 지정합니다.		
FP=:	접근하게 될 고정 정지점		
	<n>:	고정 정지점 번호	
		값 범위:	1, 2, 3, 4
	참고: FP=<n> 또는 고정 정지점 번호가 없거나 FP=0이 프로그래밍된 경우 FP=1로 해석되거나 고정 정지점 1 위치에 접근합니다.		

참고

하나의 G75 블록에 복수의 축을 프로그래밍할 수 있습니다. 이 경우 축들은 동시에 지정된 고정 정지점으로 이송됩니다.

참고

어드레스 FP의 값은 MD30610 $MA_NUM_FIX_POINT_POS에 지정된 고정 정지점 번호보다 커서는 안됩니다.

예

공구 교환의 경우 X 축 (= AX1) 및 Z 축 (= AX3) 이 고정 기계축 위치 1 (X = 151.6, Z = -17.3) 로 이동해야 합니다.

머신 데이터:

- MD30600 $MA_FIX_POINT_POS[AX1,0] = 151.6
- MD30600 $MA_FIX_POINT[AX3,0] = -17.3

NC 프로그램:

프로그램 코드	코멘트
…	
N100 G55	; 셋터블 워크 옵셋을 활성화합니다.
N110 X10 Y30 Z40	, WCS의 위치로 접근합니다.
N120 G75 X0 Z0 FP=1 M0	, 151.6으로 X축이 이동하고
	, MCS에서 17.3으로 Z축이 이동합니다.
	, 각 축은 최대 속도로 이동합니다.
	이 블록을 활성화하려면 추가 이동을 수행할 수 없습니다.
	정지를 여기에 입력하여
	끝 위치에 도달하면,
	추가 모션이 발생하지 않습니다.
N130 X10 Y30 Z40	N110 위치에 다시 접근합니다.
	; 워크 옵셋을 재활성화합니다.
…	

참고

"메거진에 있는 공구 관리" 기능이 활성인 경우 보조 기능 T... 또는 M... (보통 M6) 만으로는 G75 모션 종료 지점에서 블록 변경 금지를 실행시킬 수 없습니다.

이유: "메거진을 사용해 공구 관리"가 활성인 경우 공구 교환을 위한 보조 기능이 PLC에 출력되지 않습니다.

추가 정보

G75

축은 급 이송으로 이송됩니다. 모션은 내부적으로 "SUPA" (모든 프레임 억제) 및 "G0 RTLIOF" (단일 축 보간을 적용한 급 이송) 기능을 사용해 매핑합니다.

"RTLIOF" (단일 축 보간) 조건이 충족되지 않으면, 경로에 있는 고정 정지점에 접근합니다.

고정 정지점에 도달하면 "미세 정위치 정지" 공차 범위 내에서 축이 정지 상태가 됩니다.

G75에 대해 파라미터 설정이 가능한 다이나믹 반응

고정 정지점 위치로 이동(G75)하는 데 필요한 다이나믹 반응을 다음 머신 데이터를 사용해 설정할 수 있습니다.

MD18960 $MN_POS_DYN_MODE (포지셔닝 축 다이나믹 반응 유형)

축 추가 이동

G75 블록이 해석되는 시점에 다음과 같은 추가 축 이동을 고려합니다.

- 외부 워크 옵셋
- DRF
- 동기화 옵셋 ($AA_OFF)

이 다음부터 G75 블록이 이송을 종료할 때까지 축 추가 이동을 변경할 수 없습니다.

G75 블록 해석 후의 추가 모션이 고정 정지점에 대한 접근을 옵셋합니다.

보간이 실행되는 지점에 관계 없이 다음 추가 이동은 고려되지 않으며 그에 따라 목표 위치를 보정합니다.

- 온라인 공구 옵셋
- BCS 및 기계 좌표계의 컴파일 싸이클에서 추가 이동

활성 프레임

모든 활성 프레임은 무시됩니다. 이송은 기계 좌표계에서 수행됩니다.

공작물 좌표계/SZS의 작업 영역 제한

좌표계별 작업 영역 제한(WALCS0 ... WALCS10)은 G75가 포함된 블록에서는 효과적이지 않습니다. 목표점은 다음 블록의 시작점으로 모니터됩니다.

POSA/SPOSA로 축/스핀들 이동

프로그래밍된 축/스핀들이 이전에 POSA 또는 SPOSA에 의해 이송된 경우 고정 정지점에 접근하기 전에 이 이동부터 완료됩니다.

G75 블록에서의 스핀들 기능

스핀들이 '고정점 접근'에서 제외되면 추가 스핀들 기능(예: SPOS/SPOSA으로 포지셔닝)을 G75 블록에서 프로그래밍할 수 있습니다.

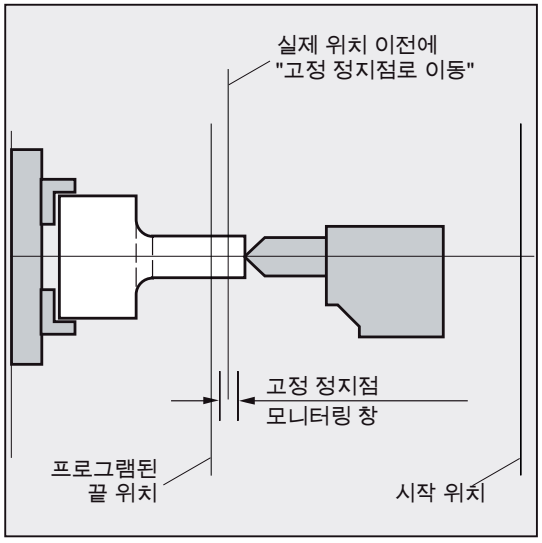
모듈로 축

모듈로 축의 경우 최단 거리를 따라 고정 정지점에 접근합니다.

3.14.6 고정 정지점으로 이동(FXS, FXST, FXSW)

기능

"고정 정지점으로 이동" 기능은 심압대, 퀼과 그리퍼 및 클램핑 공작물에 정의된 힘을 설정하는 데 사용할 수 있습니다. 이 기능은 기계 원점 복귀에도 사용할 수 있습니다.



토크가 충분히 감소된 경우에는 프로브를 연결하지 않고 단순 측정 작업을 수행할 수도 있습니다. "고정 정지점으로의 이동" 기능은 축 이송 능력으로 축과 스핀들에 구현될 수 있습니다.

구문

```
FXS[<axis>]=…
FXST[<axis>]=…
FXSW[<axis>]=…
FXS[<axis>]=… FXST[<axis>]=…
FXS[<axis>]=… FXST[<axis>]=… FXSW[<axis>]=…
```

의미

FXS:	"고정 정지점으로의 이동" 기능 활성화와 비활성화 명령어
	FXS[<axis>]=1: 기능 활성화
	FXS=[<axis>]=0: 기능 비활성화
FXST:	클램핑 토크 설정의 선택적 명령어
	최대 드라이브 토크의 %로 지정됨
FXSW:	고정 정지점 모니터링 창의 폭 설정에 대한 선택적 명령어
	mm, 인치 또는 도 단위로 지정됨
<axis>:	기계 축 이름
	기계 축 (X1, Y1, Z1, 등) 이 프로그램됩니다.

참고

명령어FXS, FXST와 FXSW은 모달입니다.

FXST 와 FXSW의 프로그래밍은 선택적입니다: 파라미터가 지정되지 않으면 최종 프로그램된 값이나 해당 머신 데이터에 설정된 값이 적용됩니다.

고정 정지점으로 이동 활성화: FXS[<axis>] = 1

종착점까지의 이동은 경로 또는 포지셔닝 축 이동으로 설명될 수 있습니다. 포지셔닝 축으로 본 기능이 블록 경계를 가로 질러 수행될 수 있습니다.

고정 정지점으로 이동은 여러 축에 대해 동시에 그리고 다른 축의 이동에 평행하게 수행될 수 있습니다. 고정 정지점은 시작 및 끝 위치 사이에 있어야 합니다.

유의사항
충돌 위험 "고정 정지점으로 이동" 기능이 이미 축/스핀들에 대해 활성화된 경우 축에 대해 새 위치를 프로그램하는 것은 허용되지 않습니다. 이 기능을 선택하기 전에 스핀들을 위치 제어 모드로 전환해야 합니다.

예:

프로그램 코드	설명
X250 Y100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2	; 축 X1은 대상 위치 X=250 mm 까지 이송 속도 F100 (선택적 지정) 으로 이동합니다. 클램핑 토크는 최고 드라이브 토크의 12.3%이고 2mm 내에서 모니터링이 수행됩니다.
...	

고정 정지점으로 이동 취소: FXS[<axis>] = 0

기능 선택 취소는 사전 처리 중지를 트리거합니다.

FXS[<axis>]=0의 블록은 이송 동작을 가지게 됩니다.

유의사항

충돌 위험

후퇴 위치로의 이송 동작은 고정 정지점으로부터 이동되어야 하며 그렇지 않으면 정지나 기계가 손상될 수 있습니다.

후퇴 위치에 도달된 경우에는 블록이 변경됩니다. 후퇴 위치를 지정하지 않은 경우 토크 제한이 비활성화되는 즉시 블록이 변경됩니다.

예:

프로그램 코드	설명
X200 Y400 G01 G94 F2000 FXS[X1]=0	; X1축이 고정 정지점에서 위치 X= 200 mm로 후퇴합니다. 다른 모든 파라미터는 옵션입니다.
...	

클램핑 토크 (FXST) 와 모니터링 창 (FXSW)

프로그램된 토크 한계FXST는 시작 블록부터 유효합니다. 즉, 고정 정지점에는 감소된 토크로 접근합니다. FXST와 FXSW는 언제라도 가공 프로그램에서 프로그램되고 변경될 수 있습니다. 동일 블록 내의 이송 전에 변경 사항이 적용됩니다.

유의사항
충돌 위험 새로운 고정 정지점 모니터링 창을 프로그래밍하면 축이 재프로그래밍 이전에 움직였을 경우 창 너비 뿐만 아니라 창 중심의 원점에도 변화가 생기게 됩니다. 창이 변경되었을 때 기계 축의 실제 위치는 새 창 중심점입니다. 고정 정지점에서의 이탈 만이 고정 정지점 모니터링이 반응을 초래하도록 창이 반드시 선택되어야 합니다.

추가 정보

상승 램프

토크 한계 설정에 대한 갑작스런 변화를 막기 위해 MD에 새 토크 제한의 상승 램프율을 정의할 수 있습니다 (예: 퀄 삽입).

경고 억제

고정 정지점 경고는 머신 데이터 항목의 경고를 마스킹하고 NEW_CONF로 새 MD설정을 활성화한 가공 프로그램에 의해 적용이 억제될 수 있습니다.

활성화

고정 정지점으로의 이동 명령어는 동기화된 작용이나 기술 싸이클로부터 요청될 수 있습니다. 이들은 동작을 야기시키지 않고 활성화될 수 있으며 토크는 즉작적으로 제한됩니다. 설정점을 통해 축이 이동한 순간부터 제한 정지 모니터가 활성화됩니다.

동기화된 작용으로부터 활성화

예:

예견된 이벤트($R1)가 발생하고 고정 정지점으로의 이동이 아직 가동하지 않으면 FXS가 축 Y에 활성화되어야 합니다. 토크는 정 토크 값의 10%에 반드시 일치해야 합니다. 모니터링 창의 폭은 기본 값으로 설정됩니다.

프로그램 코드

```
N10 IDS=1 WHENEVER (($R1=1) AND ($AA_FXS[Y]==0)) DO $R1=0 FXS[Y]=1
FXST[Y]=10
```

정상 가공 프로그램은 $R1 이 원하는 시점에서 설정되는 것을 반드시 보장해야 합니다.

동기화된 작용으로부터 비활성화

예:

예견된 이벤트($R3)가 발생하고 위상 "한계 정지 접촉됐음" (시스템 변수 $AA_FXS)에 도달되면 FXS가 반드시 선택되어야 합니다.

프로그램 코드

```
IDS=4 WHENEVER (($R3==1) AND ($AA_FXS[Y]==1)) DO FXS[Y]=0
FA[Y]=1000 POS[Y]=0
```

고정 정지점에 도달

고정 정지점에 도달한 경우:

- 이동할 거리가 삭제되고 설정 위치가 추적됩니다.
- 드라이브 토크는 프로그램 된 한계값 FXSW 까지 증가하며 그 이후에는 일정하게 유지됩니다.
- 고정 정지점 모니터링은 지정된 창의 폭 이내에서 활성화됩니다.

보충 조건

- 이동할 거리 삭제하고 측정
  "이동할 거리를 삭제하고 측정" (MEAS 명령) 과 "고정 정지점으로 이동"은 한 블록에 동시에 프로그램할 수 없습니다.
  예외: 하나의 기능은 경로 축에, 다른 하나는 위치지정 축에 작용하고 또는 양쪽 모두 위치지정에 작용합니다.
- 형상 모니터링
  "고정 정지점으로 이동"이 활성화된 동안에는 형상 모니터링은 수행되지 않습니다.
- 포지셔닝 축
  위치지정 축을 사용한 "고정 정지점으로의 이동"의 경우 고정 정지점 이동과 무관하게 블록 변경이 수행됩니다.
- 링크와 콘테이너 축
  고정 정지점으로의 이동 역시 링크와 콘테이너 축에 허용됩니다.
  할당된 기계 축의 위상은 콘테이너 회전을 넘어서까지 유지됩니다. 이는 또한 FOCON의 모달식 토크 제한에 적용됩니다.

- 고정 정지점으로 이동은 가능하지 **않습니다**:
  - 갠트리 축으로
  - PLC에서 배타적으로 제어되는 경쟁적 포지셔닝 축의 경우(FXS 는 반드시 NC 프로그램에서 선택되어야 합니다).
- 토크 한계가 너무 많이 감소되는 경우에는 축이 지정한 지령치를 뒤따를 수 없습니다. 그리고 나면 위치 컨트롤러가 토크 한계로 이동되고 형상 편차가 증가합니다. 이러한 작동 상태에서는 토크 제한 증가로 인해 갑작스런 저크(jerk) 이동이 발생할 수도 있습니다. 축이 설정점을 따를 수 있는 것을 보장하려면 형상 편차가 무제한 토크의 편차보다 크지 않은 것을 확실하게 하기 위해 형상 편차를 점검하시오.

3.14.7 드웰 시간 (G4)

시간 (드웰 시간)은 명령어 G4를 이용하여 메인 실행에서 블록이 실행되자마자 만료되는 블록에서 프로그래밍됩니다. 다음 블록으로의 블록 변경은 시간이 완전히 만료되는 순간 수행됩니다.

참고

G4는 연속 경로 모드를 중단합니다.

구문

```
G4 F<Time>
G4 S<NumSpi>
G4 S<n> = <NumSpi>
```

의미

G4:	드웰 시간을 활성화합니다.	
	블록에서만:	예
F<Time>:	드웰 시간 <Time>이 어드레스 F에 초 단위로 명시됩니다.	
S<NumSpi>:	드웰 시간이 주소 S에 현재의 메인 스핀들과 관련한 스핀들 회전 수 <NumSpi>로 프로그래밍됩니다.	
S<n>=NumSpi>:	드웰 시간이 주소 S에 주소 확장자 <n>으로 주소가 된 스핀들과 관련한 스핀들 회전 수 <NumSpi>로 프로그래밍됩니다.	

참고

드웰 블록 G4에 명시된 시간에 대한 주소 F 및 S는 프로그램의 이송 속도F...및 스핀들 속도S...에 영향을 미치지 않습니다.

보충 조건

동기 동작

드웰 시간을 가진 다음 블록이 동기 동작이 수행되는 동작 블록이 될 수 있도록 두 개의 동기 동작은 하나의 프로그램에 프로그래밍됩니다. 하나의 동기 동작은 모달 동기 동작입니다. 다른 동기 동작은 넌모달 동기 동작입니다. 넌모달 동기 동작이 모달 동기 동작에 영향을 주기 위한 것이라면, 예를 들어 **적어도 두 번의 보간 싸이클** 동안 실행될 수 있도록 UNLOCK을 이용하여 이를 해제하도록 합니다. G4 F<보간기_싸이클 * 2>는 반드시 유효한 드웰 시간과 같도록 제공되어야 합니다.

유효 드웰 시간은 기계 데이터 MD10280 $MN_PROG_FUNCTION_MASK, Bit 4 = <value> 상의 설정값에 의거합니다.

값	의미
0	유효 드웰 시간은 프로그래밍된 드웰 시간과 같습니다.
1	유효 드웰 시간은 보간기 싸이클 다음 숫자 중 가장 큰 배수로 반올림되어 프로그래밍된 드웰 시간과 같습니다.

프로그램 예:

- MD10071 $MN_IPO_CYCLE_TIME == 8 ms
- MD10280 $MN_PROG_FUNCTION_MASK, Bit 4 = 1

프로그램 코드	주석
N10 WHEN TRUE DO LOCK(1)	; 넌모달 동기 동작(SynAct): 모달 동기 동작(SynAct)
	; 잠금. ID=1
N20 G4 F2	; N10의 동기 동작(SyncAct)을 위한 동작 블록
N30 WHEN TRUE DO UNLOCK(1)	; 넌모달 동기 동작(SynAct): 모달 동기 동작(SynAct)
	; 해제. ID=1
N40 ID=1 WHENEVER TRUE DO $R0=1 RDISABLE	; 모달 동기 동작(SynAct) ID=1
	; R 파라미터 R0=1
	; 리드인 디스에이블 설정

프로그램 코드	주석
N50 G4 F0.012	; N40 및 N50의 동기 동작(SyncAct)을 위한 동작 블록
	; 아래 "설명" 문구를 참조하십시오.
N60 G4 F10	

설명

필요한 행동은 N30의 모달 동기 동작이 N40의 ID=1 모달 동기 동작의 활성 잠금 (LOCK)을 취소하여 R 파라미터가 N50에 씌어지고, 리드인 디스에이블이 활성화되도록 하는 것입니다. 이러한 행동은 활성 드웰 시간이 적어도 두 번의 보간 싸이클 정도 되는 경우에만 달성됩니다.

활성 드웰 시간은 MD10280 $MN_PROG_FUNCTION_MASK, Bit 4에서의 프로그래밍된 드웰 시간, 보간 싸이클 그리고 설정에서 나옵니다. 활성 드웰 시간이 적어도 두 번의 보간 싸이클 정도 되도록 하기 위해서는 아래 드웰 시간이 반드시 프로그래밍되어야 합니다:

- Bit 4 == 0: 프로그래밍된 드웰 시간 ≥ 2 * 보간기 싸이클
- Bit 4 == 1: 프로그래밍된 드웰 시간 ≥ 1.5 * 보간기 싸이클

활성 드웰 시간이 두 번의 보간 싸이클보다 짧은 경우, R 파라미터 쓰기와 리드인 디스에이블은 블록 N60까지 실행되지 않습니다.

예

프로그램 코드	설명
N10 G1 F200 Z-5 S300 M3	;피드 F; 스핀들 속도 S
N20 G4 F3	; 드웰 시간: 3 s
N30 X40 Y10	
N40 G4 S30	; 스핀들 30회전 드웰 (S=300 rpm에서 100% 속도 오버라이드는 t = 0.1 min 의미).
N50 X...	; N10에 프로그래밍된 이송 속도 및 스핀들 속도가 계속 적용됩니다.

3.14.8 내부 전처리 정지

기능

시스템은 기계 상태 데이터 ($SA...) 를 액세스하기 위해 내부 전처리 정지를 생성합니다. 전처리 후 저장된 모든 블록이 완전히 실행되기 전까지는 후속 블록이 실행되지 않습니다. 이전 블록은 정위치 정지에서 멈춥니다 (G9 와 같음).

예제

프로그램 코드	설명
...	
N40 POSA[X]=100	
N50 IF $AA_IM[X]==R100 GOTOF MARKE1	; 기계 상태 데이터 ($SA...) 를 액세스하기 위해 제어기가 내부 전처리 정지를 생성합니다.
N60 G0 Y100	
N70 WAITP(X)	
N80 LABEL1:	
...	

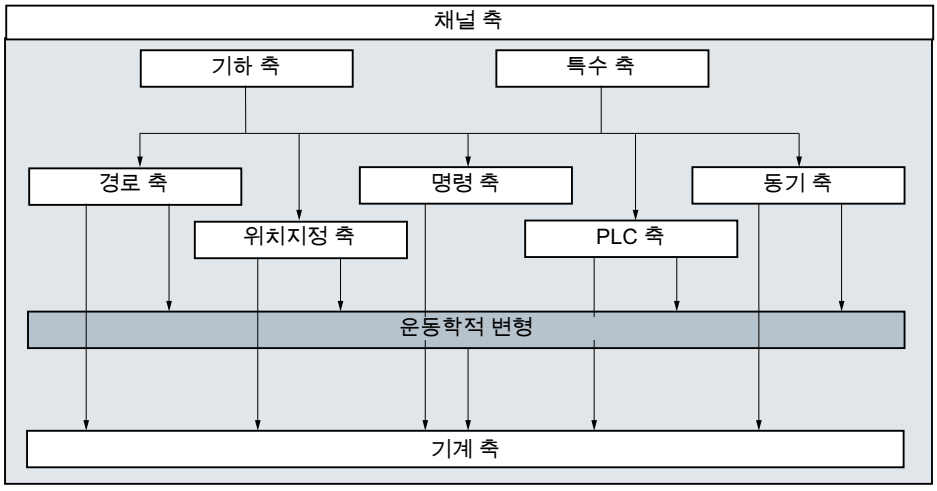
3.15 기타 정보

3.15.1 축

축 종류

프로그래밍할 때 축의 유형은 다음과 같습니다:

- 메인 축/기하 축
- 특수 축
- 메인 스핀들, 마스터 스핀들
- 기계 축
- 채널 축
- 경로 축
- 포지셔닝 축
- 동기 축
- 명령 축
- PLC 축/경쟁 포지셔닝 축
- 링크 축
- 리드 링크 축



3.15.1.1 메인 축/기하 축

메인 축은 우측 각, 오른손 좌표계를 정의합니다. 공구 이동은 이 좌표계로 프로그램됩니다.

NC 에서 메인 축은 기하 축이라 불립니다. 이 용어는 또한 본 프로그래밍 안내서에서도 사용됩니다.

기하 축 교체 기능

"기하 축 교체" 기능 (기능 매뉴얼) 으로 가공 프로그램에서 머신 데이터를 사용해 설정한 기하 축 그룹화를 변경할 수 있습니다. 어떤 기하 축이든 특수 동기 축으로 정의된 채널 축으로 교체할 수 있습니다.

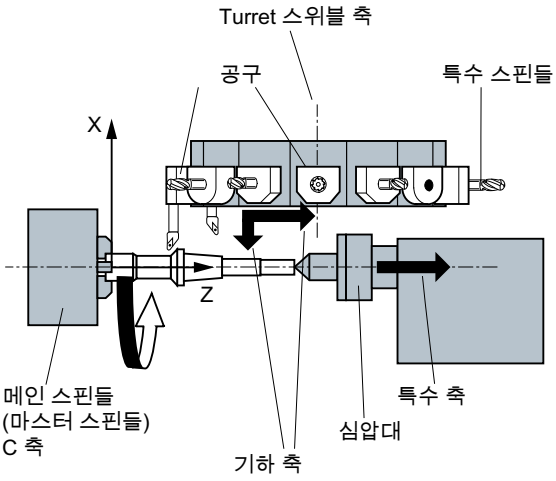
축 이름

기하 축의 이름/식별자는 다음과 같은 머신 데이터를 사용하여 정의할 수 있습니다.

MD20060 $MC_AXCONF_GEOAX_NAME_TAB(채널에서 기하 축 이름)

선반에 대한 표준 식별자:

1. 기하 축: X
2. 기하 축: Z



밀링 기계에 대한 표준 식별자:

1. 기하 축: X
2. 기하 축: Y
3. 기하 축: Z

추가 정보

프레임과 공작물 기하 (형상) 프로그래밍에 최대 3개의 기하 축이 사용됩니다.

기준이 가능하다면 기하와 채널 축의 식별자는 동일할 수 있습니다.

기하 축과 채널 축의 이름은 모든 채널에서 동일해야 합니다. 즉, 프로그램이 어느 채널에서도 실행될 수 있다는 뜻입니다.

3.15.1.2 특수 축

기하 축과는 반대로, 특수 축 사이에는 기하학적 관계는 정의되지 않습니다.

전형적인 특수 축은 다음과 같습니다.

- 공구 리볼버 축
- 스위블 테이블 축
- 스위블 헤드 축
- 로더 축

축 이름

원형 매거진이 있는 선삭 기계의 경우:

- 리볼버 위치 U
- 심압대 V

프로그래밍 예제

프로그램 코드	설명
N10 G1 X100 Y20 Z30 A40 F300	; 경로 축 이동
N20 POS[U]=10POS[X]=20 FA[U]=200 FA[X]=350	; 포지셔닝 축 이동
N30 G1 X500 Y80 POS[U]=150FA[U]=300 F550	; 경로 및 포지셔닝 축
N40 G74 X1=0 Z1=0	; 원점 복귀

3.15.1.3 메인 스핀들, 마스터 스핀들

기계 구조적으로 메인 스핀들을 결정합니다. 이 스핀들은 머신 데이터에서 보통 마스터 스핀들로 선언됩니다.

스핀들 지정은 프로그램 명령어 SETMS (<스핀들 번호>) 로 변경할 수 있습니다. SETMS를 사용해 스핀들 번호를 지정하지 않고 머신 데이터에 정의된 마스터 스핀들로 다시 전환할 수 있습니다.

마스터 스핀들은 나사 절삭과 같은 기능을 지원합니다.

스핀들 이름

S 또는 S0

3.15.1.4 기계 축

기계 축은 기계에 존재하는 물리적인 축입니다.

경로 축 또는 추가 축의 프로그램된 모션은 해당 채널에 활성화되어 있는 변형 (TRANSMIT, TRACYL, 또는 TRAORI) 때문에 여러 머신 축에서 작동할 수 있습니다.

기계 축은 특별한 경우에만 프로그램에 직접 언급됩니다 (예: 원점 또는 고정 정지점 접근을 위해).

축 이름

기계 축의 이름/식별자는 다음과 같은 NC별 머신 데이터를 사용하여 정의할 수 있습니다.

MD10000 $MN_AXCONF_MACHAX_NAME_TAB(기계 축 이름)

디폴트 설정: X1, Y1, Z1, A1, B1, C1, U1, V1

또한, 기계 축에는 머신 데이터에 설정된 이름과는 별개로 항상 사용할 수 있는 고정된 축 식별자가 있습니다.

AX1, AX2, ..., AX<n>

3.15.1.5 채널 축

채널에 할당되는 모든 기하 축, 특수 축, 기계 축을 채널 축이라고 합니다.

축 이름

기하 축과 특수 축의 채널별 이름/식별자는 다음과 같은 머신 데이터를 사용하여 정의할 수 있습니다.

MD20080 $MC_AXCONF_CHANAX_NAME_TAB (채널 축 이름)

디폴트 설정: X, Y, Z, A, B, C, U, V

기하 축 또는 특수 축이 채널 내에서 어떤 기계 축에 에뮬레이션되는지에 관해서는 다음 머신 데이터를 사용하여 정의합니다.

MD20070 $MC_AXCONF_MACHAX_USED (사용된 기계 축)

3.15.1.6 경로 축

경로 축은 경로를 정의하고 공간에서 공구의 이동을 정의합니다.

이 경로에 프로그램된 이송이 활성화됩니다. 이 경로에는 프로그램된 이송 속도가 사용됩니다. 이러한 축들을 기하 축이라고 합니다.

그러나, 디폴트 설정이 경로 축을 정의하고 속도를 결정합니다.

경로 축은 NC 프로그램에서 FGROUP으로 지정할 수 있습니다.

FGROUP에 대한 자세한 내용은 "이송 속도 (G93, G94, G95, F, FGROUP, FL, FGREF) (페이지 117)"를 참조하십시오.

3.15.1.7 위치지정 축

포지셔닝 축은 별도로 보간됩니다. 즉, 각 포지셔닝 축은 자체적인 축 보간자와 이송 속도를 갖습니다. 포지셔닝 축은 경로 축과 같이 보간되지 않습니다.

포지셔닝 축은 NC 프로그램 또는 PLC에 의해 이송됩니다. 축이 NC 프로그램과 PLC에 의해 동시에 이송하게 되면 오류 메시지가 나타납니다.

일반적인 포지셔닝 축:

- 공작물을 기계 방향으로 이동시키는 로더
- 공작물을 기계로부터 멀리 이동시키는 로더
- 공구 매거진/터렛

유형

블록 끝 또는 몇 블록에 걸친 동기화를 사용하여 포지셔닝 축 간의 구분이 만들어 집니다.

POS 축

이 블록에서 프로그램된 모든 축과 포지셔닝 축이 각기 프로그램된 종점에 도달했을 때 블록의 에서 블록 변경이 발생합니다.

POSA 축

이들 포지셔닝 축의 이동은 수 개의 블록에 걸쳐 연장될 수 있습니다.

POSP 축

끝 위치에 도달하려는 이들 포지셔닝 축 이동이 이 부분에서 발생합니다.

참고

포지셔닝 축은 특별한 POS/POSA 식별자 없이 이송되면 축이 동기화됩니다.

경로 축의 연속 이송 모드 (G64) 는 포지셔닝 축 (PODS) 가 경로 축 이전에 최종 위치에 도달할 때만 가능해집니다.

POS/POSA로 프로그래밍한 경로 축은 이 블록이 진행되는 동안 경로 축 그룹화에서 제거됩니다.

POS, POSA 및 POSP에 대한 자세한 내용은 "포지셔닝 축 이송 (POS, POSA, POSP, FA, WAITP, WAITMC) (페이지 126)"를 참조하십시오.

3.15.1.8 동기화 축

동기 축은 경로에 동기화되어 시작 위치에서 프로그램된 끝 위치로 이송하게 됩니다.

F에 프로그래밍된 이동 속도는 동기 축을 제외하고 블록 내에 프로그래밍된 모든 경로 축에 적용됩니다. 동기 축은 이송하는데 경로 축과 동일한 시간을 소요합니다.

동기 축은 동기화되어 경로 보간으로 이송되는 회전 축이 될 수 있습니다.

3.15.1.9 명령 축

명령 축은 이벤트 (명령) 에 반응하여 동기화 작용에서 시작됩니다. 이들은 가공 프로그램과 완전하게 동기화되어 포지셔닝되고, 시작 또는 정지됩니다. 가공 프로그램과 동기 동작으로 축을 동시에 이동할 수는 없습니다.

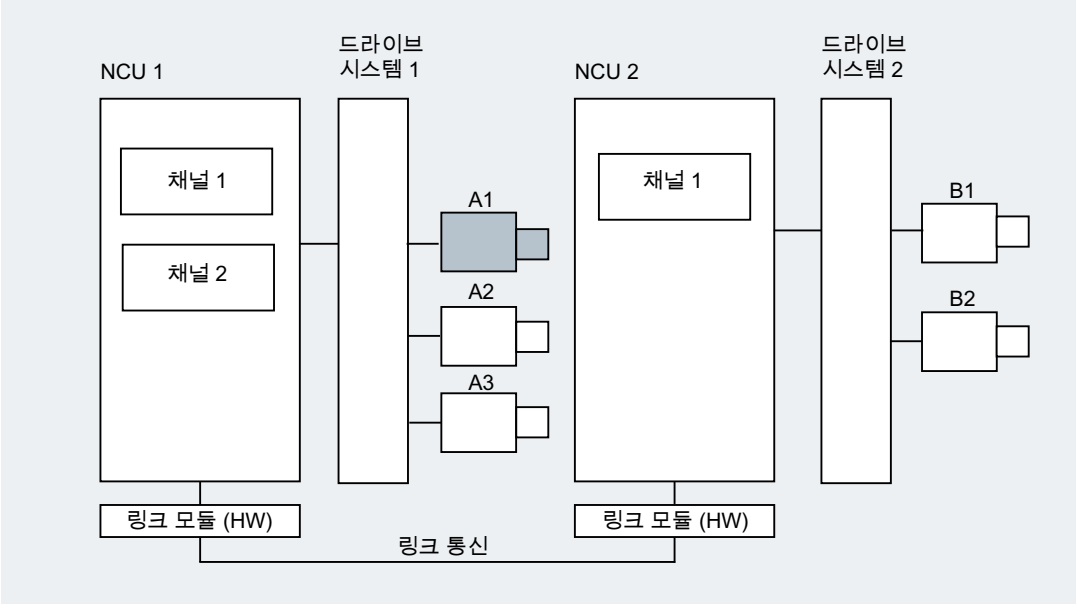
명령 축은 별도로 보간됩니다. 즉, 각 명령 축은 자체적인 축 보간자와 이송 속도를 갖습니다.

3.15.1.10 PLC 축

PLC 축은 기본 프로그램에 있는 특수 기능 블록을 통하여 PLC에 의해 이송되며 이들의 이동은 다른 모든 축에 동기화될 수 있습니다. 이송 이동은 경로 및 동기 이동과는 별도로 진행됩니다.

3.15.1.11 링크 축

링크 축은 다른 NCU에 물리적으로 연결되고 이들의 위치가 NCU에서 제어되는 축입니다. 링크 축은 **다른** NCU의 채널에 동적으로 할당될 수 있습니다. 링크 축은 특정 NCU의 관점에서 로칼 축이 아닙니다.



축 컨테이너 개념은 NCU 할당의 동적인 수정에 사용됩니다. 가공 프로그램에서 GET 및 RELEASE로 축을 교체하는 방법은 링크 축에는 사용할 수 **없습니다**.

추가 정보

요구 사항

- 참여하는 NCUs, NCU1 및 NCU2 는 반드시 링크 모듈을 통해 고속 통신의 수단으로 연결되어야 합니다.
- 축은 반드시 머신 데이터에 의해 적절하게 설정되어야 합니다.
- "링크 축" 옵션이 반드시 설치되어야 합니다.

설명

축이 물리적으로 드라이브에 연결되는 NCU 에 위치 컨트롤러가 주어지게 됩니다. 이 NCU에는 또한 연관된 축 VDI 인터페이스가 포함되어 있습니다. 링크 축의 위치 설정점이 다른 NCU에 생성되고 NCU 링크를 통해 통신됩니다.

링크 통신은 반드시 보간자와 위치 컨트롤러 또는 PLC 인터페이스 간의 상호 작용을 보장해야 합니다. 보간으로 계산된 설정점은 반드시 홈 NCU에 있는 위치 제어 고리로 연결되어야 하고, 그 역 역시 그렇게 되어 실제 값은 반드시 그 곳에서 다시 보간자로 복귀하게 됩니다.

축 컨테이너

축 컨테이너는 로컬 축 그리고/또는 링크 축이 채널에 할당되는 원형 버퍼 데이터 구조입니다. 원형 버퍼의 입력은 **주기적으로 전환**될 수 있습니다.

로컬 축이나 링크 축에의 직접 기준에 추가하여 논리적 기계 축 영상의 링크 축 환경 설정 역시 축 컨테이너에의 기준이 허용되어야 합니다. 이 기준 유형은 다음으로 구성됩니다.

- 컨테이너 번호 **및**
- 슬롯 (컨테이너 내의 원형 버퍼 포켓)

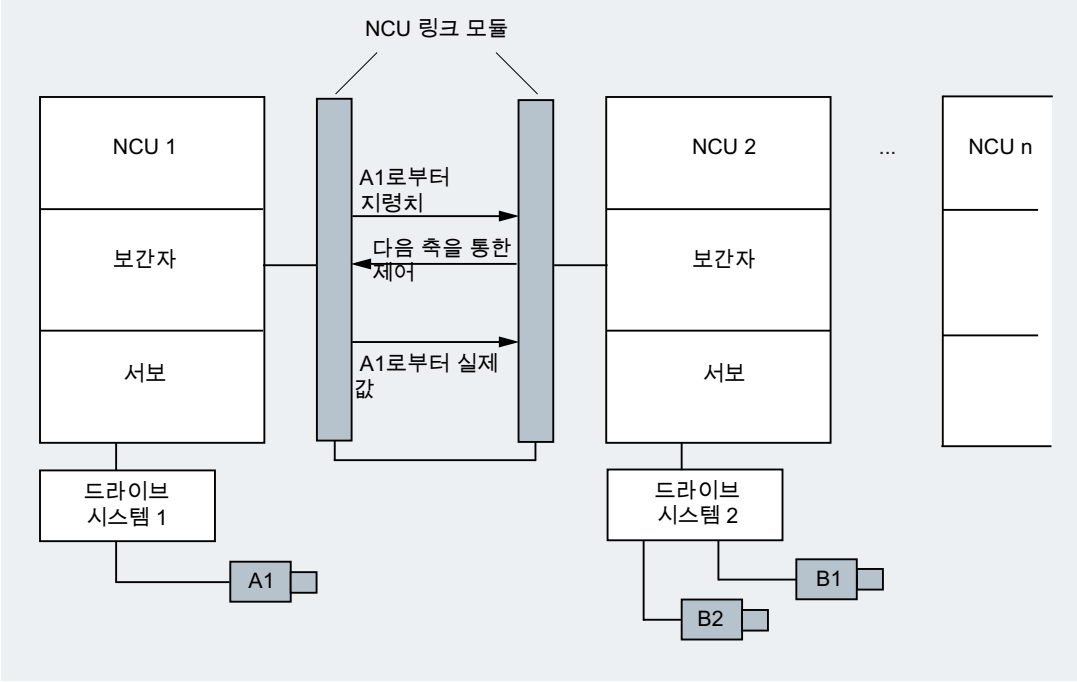
원형 버퍼 위치의 입력에는 다음이 포함되어 있습니다:

- 로컬 축 **또는**
- 링크 축

축 컨테이너 입력에는 개별 NCU의 관점에서 로컬 축 또는 링크 축이 포함됩니다. 개별 NCU의 논리 기계 축 이미지 (MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB) 에 포함되는 항목은 고정되어 있습니다.

3.15.1.12 선도 링크 축

리드 링크 축은 하나의 NCU에 의해 보간되고 하나 또는 몇 개의 NCU에 의하여 제어 종속 축의 마스터 축으로 활용되는 것입니다.



축 위치 제어기 경고는 리드 링크 축을 통해 영향을 받은 축에 연결되는 다른 모든 NCU에 발송됩니다.

리드 링크 축에 의존하는 NCU는 이와의 후속 커플링 관계성을 활용할 수 있습니다:

- 마스터 값 (설정점, 실 마스터 값, 모의된 마스터 값)
- 커플 모션
- 접선 보정
- 전자 기어 (ELG)
- 동기 스핀들

프로그래밍

마스터 NCU:

마스터 값 축에 물리적으로 할당된 NCU만이 이 축의 이동 동작을 프로그램할 수 있습니다. 이동 프로그램은 절대 어떠한 특수 기능이나 작업을 포함해서는 안됩니다.

종속 축의 NCU:

종속 축의 NCU에의 이동 프로그램은 절대 리드 링크 축 (마스터 값 축) 의 어떠한 이동 명령어를 포함해서는 안됩니다. 이 규칙의 모든 위반은 경고를 촉발합니다.

리드 링크 축은 채널 축 이름을 통해 일상적 방법으로 지정됩니다. 리드 링크 축의 상태는 선택된 시스템 변수의 수단으로 액세스될 수 있습니다.

추가 정보

요구 사항

- 관련 NCU, 즉 NCU1 ~ NCU<n> (<n>은 최대 8)이 고속 통신을 위해 링크 모듈을 통해 상호 연결되어 있어야 합니다.
  참고 자료:
  장치 매뉴얼 환경설정 NCU
- 축은 반드시 머신 데이터에 의해 적절하게 설정되어야 합니다.
- "링크 축" 옵션이 반드시 설치되어야 합니다.
- 참여하는 모든 NCU에 대해 동일한 보간 클럭 싸이클을 구성해야 합니다.

제한사항

- 리드 링크 축인 마스터 축은 링크 축이 될 수 없습니다. 다시 말해 홈 NCU 외의 다른 NCU에 의해 이송될 수 없습니다.
- 리드 링크 축인 마스터 축은 컨테이너 축이 될 수 없습니다. 다시 말해 다른 NCU들이 주소를 다르게 지정할 수 없습니다.
- 리드 링크 축은 갠트리 그룹핑에서 리딩 축으로 프로그램될 수 없습니다.
- 리드 링크 축과의 커플링은 종속적으로 될 수 없습니다.
- 축 대체는 리드 링크 축의 홈 NCU 이내에서만 이행될 수 있습니다.

시스템 변수

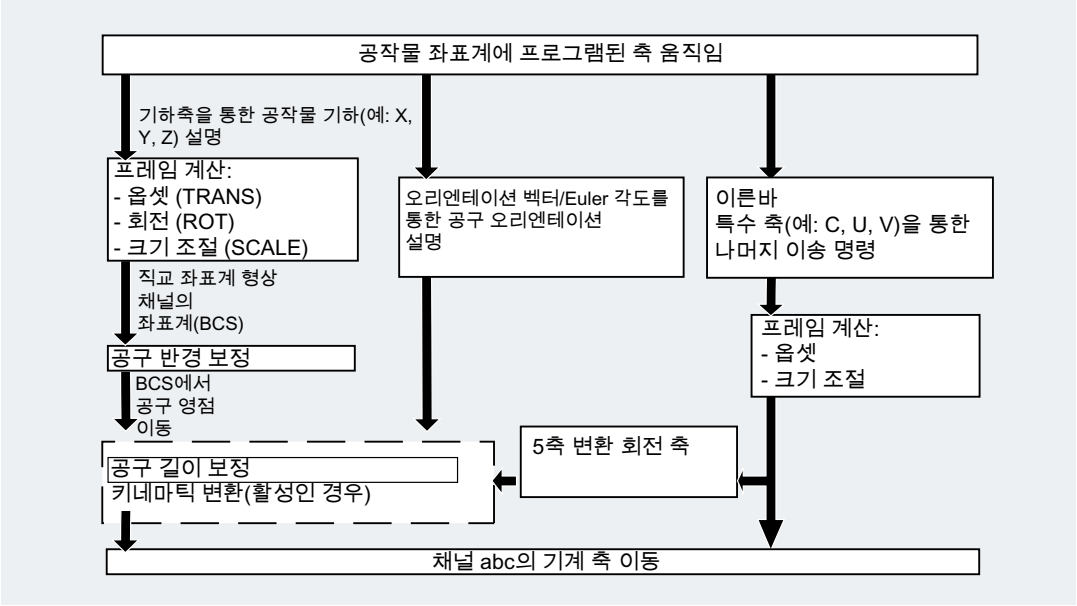
다음의 시스템 변수가 리드 링크 축의 채널 축 이름과 연결되어 사용될 수 있습니다:

시스템 변수	의미
$AA_LEAD_SP	가상 마스터 값 - 위치
$AA_LEAD_SV	가상 마스터 값 - 속도

마스터 축의 홈 NCU에의해 이들 시스템 변수가 업데이트되면 새 값 역시 종속 축을 이 마스터 축의 기능으로 제어하려는 다른 NCU로 전송됩니다.

3.15.2 이송 명령부터 기계 이동까지

다음 그림은 프로그래밍된 축 이동 (이송 명령) 과 그 결과인 기계 이동의 관계를 나타낸 것입니다.

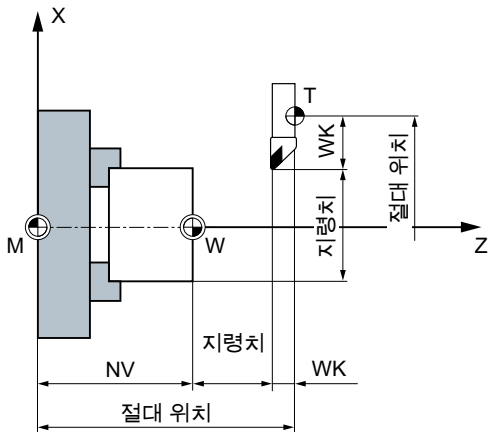


3.15.3 경로 계산

경로 계산으로 모든 옵셋과 보상을 계산하여 블록 내 이동할 거리가 결정됩니다.

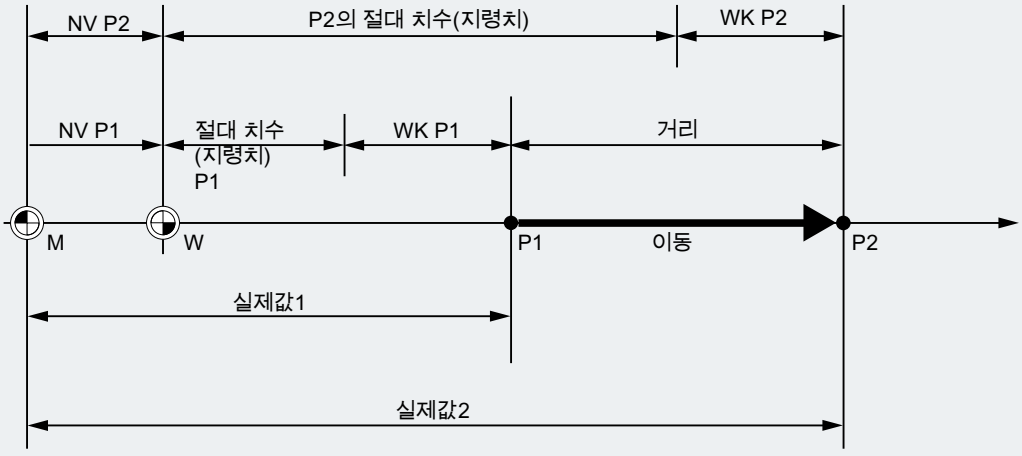
일반적으로:

경로 =
지령치 - 실제 값 + 제로 옵셋(ZO) + 공구 옵셋(TO)



새 프로그램 블록에서 새 워크 옵셋과 새 공구 옵셋이 프로그램되면 다음이 적용됩니다:

- 절대 값으로:
  경로 = (절대 치수 P2 - 절대 치수 P1) + (WO P2 - WO P1) +
  (TO P2 - TO P1).
- 증분 값으로:
  경로 = 증분 치수 + (WO P2 - WO P1) +
  (TO P2 - TO P1).



3.15.4 어드레스

고정 어드레스

이 어드레스는 어드레스 문자가 변하지 않으며 영구적으로 설정됩니다.

목록은 표 "고정 어드레스 (페이지 1452)"에서 확인할 수 있습니다.

설정 가능한 어드레스

기계 제조업체에서 머신 데이터를 통해 이들 주소의 다른 이름을 할당할 수 있습니다.

참고

설정 가능한 어드레스는 제어 시스템 내에서 고유해야 합니다. 즉, 동일한 어드레스 이름이 다른 어드레스 유형(축 값 및 종점, 공구 오리엔테이션, 보간 파라미터 등)에 사용되어서는 안 됩니다.

목록은 표 "셋터블 어드레스 (페이지 1458)"에서 확인할 수 있습니다.

모달/넌모달 주소

모달식 주소는 동일 주소에 새 값이 프로그램되기 전까지 프로그램된 값(모든 후속 블록에서)으로 유효하게 유지됩니다.

비-모달식 주소는 프로그램된 블록에만 적용됩니다.

예:

프로그램 코드	설명
N10 G01 F500 X10	
N20 X10	; 새 이송 속도가 입력될 때까지 N10의 이송 속도 F가 계속 적용됩니다.

축 확장의 주소

축 확장의 주소에서는 주소 후 사각 괄호 안에 축 이름이 삽입됩니다. 축 이름이 축에 할당됩니다.

예:

프로그램 코드	설명
FA[U]=400	; U 축의 축 이송 속도.

표 "고정 어드레스 (페이지 1452)" 참조.

확장 어드레스 표기

확장 주소 표기로 많은 수의 축과 스핀들이 시스템 내에 조직됩니다.

확장된 주소는 많은 수의 확장과 문자 기호"="가 할당된 산술적 표현으로 구성됩니다. 숫자적 확장은 하나 또는 두 단위이며 항상 양의 정수 입니다.

확장 주소 표기는 다음의 직접 주소에만 허용됩니다.

어드레스	의미
X, Y, Z, ...	축 주소
I, J, K	보간 파라미터
S	스핀들 속도
SPOS, SPOSA	스핀들 위치
M	특수 기능
H	보조 기능
T	공구 번호
F	이송 속도

예:

프로그램 코드	설명
`X7`	`; 7은 값이며 "=" 이 필요하지 않지만 "=" 문자를 사용할 수도 있습니다.`
`X4=20`	`; 축 X4; "="이 필요합니다.`
`CR=7.3`	`; 2문자; "="가 필요합니다.`
`S1=470`	`; 첫 번째 스핀들 속도: 470 rpm`
`M3=5`	`; 세 번째 스핀들 정지`

숫자 확장은 주소 M, H, S 그리고 SPOS 및 SPOSA의 변수와 대체될 수 있습니다. 변수 식별자는 사각 괄호안에 들어가게 됩니다.

예:

프로그램 코드	설명
`S[SPINU]=470`	`; SPINU 변수에 번호가 저장된 스핀들의 속도.`
`M[SPINU]=3`	`; SPINU 변수에 번호가 저장된 스핀들을 CW 방향 회전.`
`T[SPINU]=7`	`; SPINU 변수에 번호가 저장된 스핀들의 공구 선택.`

3.15.5 이름

DIN 66025 에 따른 명령에는 NC 고급 언어로 네임드 오브젝트 등을 추가합니다.

네임드 오브젝트의 예:

- 시스템 변수
- 사용자 정의 변수
- 축/스핀들
- 서브프로그램
- 키워드
- 분기 라벨
- 매크로

참고

식별자는 고유해야 합니다. 다른 개체와 동일한 식별자를 사용할 수 **없습니다**.

이름 지정 규칙

이름은 다음 규칙에 따라 자유롭게 선택할 수 있습니다.

- 허용되는 문자:
  - 문자: A ... Z, a ... z
  - 숫자: 0 ... 9
  - 밑줄: _
- 처음 2자는 글자 또는 밑줄이어야 합니다.
- 최대 길이:
  - 프로그램 이름 (페이지 48): 24문자
  - 축 이름: 8 문자
  - 변수 이름: 31 문자

참고

예약된 키워드를 식별자로 사용해서는 안됩니다.

싸이클

이름 충돌을 방지하기 위해 사용자 싸이클의 이름을 지정할 때 다음 규칙을 준수하기 바랍니다.

문자열	예약된 이름
• CYCLE • CUST_ • GROUP_ • _ • S_ • E_ • F_	SIEMENS 싸이클
• CCS_	SIEMENS 컴파일 싸이클
• CC_	사용자 컴파일 싸이클

사용자 싸이클

사용자 싸이클의 이름은 U_로 시작하는 것이 좋습니다.

변수

변수 이름 지정에 관한 정보는 다음 장에서 확인할 수 있습니다.

- 시스템 데이터 (페이지 430)
- 사용자 변수 정의 (DEF) (페이지 439)

3.15.6 상수

상수 (일반)

상수는 프로그램 실행 중에 값이 바뀌지 않는 데이터를 말합니다. 예를 들어 어드레스에 지정된 값이 상수에 해당합니다.

10진수 상수

10진수 상수의 숫자 값은 10진법으로 표시됩니다.

INTEGER 상수

INTEGER 상수는 정수 값, 즉 소수점이 없고 부호는 있거나 없을 수 있는 일련의 숫자입니다.

예:

X10	어드레스 X에 값 +10 지정
X-35	어드레스 X에 값 -35 지정
X0	어드레스 X에 값 0 지정 **참고:** X0을 X로 대체할 수 없습니다.

REAL 상수

REAL 상수는 소수점이 있고 부호와 지수는 있거나 없을 수 있는 일련의 숫자를 말합니다.

예:

X10.25	어드레스 X에 값 +10.25 할당
X-10.25	어드레스 X에 값 -10.25 할당
X0.25	어드레스 X에 값 +0.25 할당
X.25	선도 "0" 이 없는 어드레스 X에 값 +0.25 할당
X=-.1EX-3	어드레스 X에 값 $-0.1*10^{-3}$ 할당

참고

십진 단위표가 허용되는 어드레스에서는 어드레스에 실제 제공된 것 보다 더 많은 십진 단위가 지정된다면 제공된 단위의 수에 맞추기 위해 반올림하게 됩니다.

16진법 상수

상수는 16진법 상수, 즉 16을 밑으로 하는 숫자로 해석될 수도 있습니다. 16진법에서 문자 A~F는 10~15를 의미합니다.

16진수는 작은 따옴표 안에 들어가고 16진법 표기 값 앞에 문자 'H"로 시작합니다. 문자와 숫자 사이에 구분자가 허용됩니다.

예:

프로그램 코드	코멘트
$MC_TOOL_MANAGEMENT_MASK='H7F'	; 16진수 상수를 지정하면 비트 0~7이 머신 데이터에 설정됩니다.

참고

문자의 최대 개수는 정수 데이터 유형의 값 범위로 제한됩니다.

이진 상수

상수는 또한 이진법 포맷으로 해석될 수 있습니다. 이 경우 숫자 "0"과 "1"만이 사용됩니다.

이진수는 작은 따옴표 안에 들어가고 이진법 표기 값 앞에 문자 "B"로 시작합니다. 숫자 사이에 구분자가 허용됩니다.

예:

프로그램 코드	설명
`$MN_AUXFU_GROUP_SPEC='B10000001'`	; 이진 상수를 지정하면 비트 0과 비트 7이 머신 데이터에 설정됩니다.

참고

문자의 최대 개수는 정수 데이터 유형의 값 범위로 제한됩니다.

3.15.7 연산자 및 산술 함수

연산자

산술 연산자

REAL 및 INT 형식의 시스템 변수는 다음 연산자에 의해 연결될 수 있습니다.

연산자	의미
+	덧셈
-	뺄셈
*	곱셈
/	• 동기 동작의 나눗셈: INT / INT ⇒ **INT** • 동기 동작의 나눗셈 및 ITOR() 함수를 사용한 REAL 결과: ITOR(INT) / ITOR(INT) ⇒ **REAL** • NC 프로그램의 나눗셈: INT / INT ⇒ **REAL**

연산자	의미
DIV	정수 나눗셈: INT / INT ⇒ **INT**
MOD	모듈로 나눗셈 (유형 INT에만 해당) 은 INT 나눗셈의 나머지를 제공합니다. 예: 3 MOD 4 = 3

참고

같은 형식의 변수만 이 연산자에 의해 연결될 수 있습니다.

관계 연산자

연산자	의미
==	같음
>	같지 않음
<	보다 작음
>	보다 큼
<=	보다 작거나 같음
>=	보다 크거나 같음

부울 운전자

연산자	의미
NOT	NOT
AND	AND
OR	OR
XOR	익스클루시브 OR

비트 논리 연산자

연산자	의미
B_OR	비트 단위 OR
B_AND	비트 단위 AND
B_XOR	비트 단위 익스클루시브 OR
B_NOT	비트 단위 부정

연산자 우선 순위

동기 동작에서 연산자의 실행 우선 순위는 다음과 같습니다(가장 높은 우선 순위: 1):

우선 순위	연산자	의미
1	NOT, B_NOT	부정, 비트 단위 부정
2	*, /, DIV, MOD	곱셈, 나눗셈
3	+, -	덧셈, 뺄셈
4	B_AND	비트 단위 AND
5	B_XOR	비트 단위 익스클루시브 OR
6	B_OR	비트 단위 OR
7	AND	AND
8	XOR	익스클루시브 OR
9	OR	OR
10	<<	문자열 연결, 결과 유형 STRING
11	==, <>, <, >, >=, <=	관계 연산자

참고

표현식에서 여러 연산자를 사용할 때는 괄호"(...)"를 설정해 각 연산자의 우선 순위를 명확히 지정하는 것이 좋습니다.

여러 연산자를 사용하는 표현식의 예:

프로그램 코드

```
... WHEN ($AA_IM[X] > VALUE) AND ($AA_IM[Y] > VALUE1) DO ...
```

산술 함수

연산자	의미
SIN()	사인
COS()	코사인
TAN()	탄젠트
ASIN()	아크 사인
ACOS()	아크 코사인
ATAN2()	아크 탄젠트 2
SQRT()	제곱근

연산자	의미
ABS()	절대치
POT()	2승 (제곱)
TRUNC()	정수 비교 명령의 정확도는 TRUNC를 사용하여 설정할 수 있습니다.
ROUND()	정수로 반올림
LN()	자연 로그
EXP()	지수 함수

지수 연동

"Array of ..." 형식의 시스템 변수의 지수는 시스템 변수가 될 수 있습니다. 또한 지수는 보간 클록 싸이클의 메인 실행에서 평가됩니다.

예

프로그램 코드

```
... WHEN … DO $AC_PARAM[$AC_MARKER[1]]=3
```

제한사항

- 지수를 추가 시스템 변수와 중첩하는 것은 허용되지 않습니다.
- 지수는 전처리 변수를 통해 형성되면 안 됩니다. 따라서$P_EP가 전처리 변수: $AC_PARAM[1] = $P_EP[$AC_MARKER[0]]이기 때문에 허용되지 **않습니다**.

다음 또한 참고:

산술 함수 (페이지 489)

명령 (페이지 489)

작업 준비 4

4.1 플렉시블 NC 프로그래밍

4.1.1 변수

시스템 데이터 및 사용자 데이터 영역의 변수를 사용하면 특히 산술 함수 및 점검 구조와 관련하여 매우 유연한 NC 프로그램 및 싸이클을 작성할 수 있습니다.

⚠ 경고

변경된 변수로 인한 물질적 피해 및 부상

NC 프로그램에서 변수 사용 시 기계 작업자 또는 해당 접근 권한이 있는 미승인 인력이 변수를 변경할 수 있기 때문에 프로그램 실행에 영향을 줄 수 있습니다. 이로 인해 물질적 피해 및 부상이 발생할 수 있습니다.

- 변수 변경으로 인해 프로그램 실행에 부정적 영향이 발생하는 것을 방지하려면 NC 프로그램에서 적절한 데이터 점검("입력 검증")을 제공해야 합니다.

- 시스템 데이터
  시스템 데이터는 시스템 내에 미리 정의된 변수를 포함하고 있습니다. 이러한 변수들은 정의된 의미를 가지고 있습니다. 이 변수들은 시스템 소프트웨어에 의해 주로 사용됩니다. 사용자는 이러한 변수들을 NC 프로그램과 싸이클에서 읽거나 쓸 수 있습니다. 예: 기계 데이터, 설정 데이터, 시스템 변수.
  시스템 데이터의 의미가 고정되어 있다고 할지라도, 사용자는 특정 범위 내에서 재정의를 통해 특성을 수정할 수 있습니다.
  "시스템 데이터, 사용자 데이터 및 NC 명령어 (REDEF)의 재정의 (페이지 446)"를 참조하십시오.
- 사용자 데이터
  사용자 데이터는 사용자가 정의한 변수들을 포함하고 있으며, 그 의미는 전적으로 사용자가 정의할 수 있습니다. 시스템은 이를 평가하지 않습니다.
  사용자 데이터는 다음으로 나뉩니다:
  - 사전 정의된 사용자 변수
    사전 정의된 사용자 변수는 시스템에서 이미 정의된 변수이며, 그 번호가 기계 데이터 내에 파라미터화되어 있습니다. 사용자는 이러한 변수의 속성을 변경할 수 있습니다.
    "시스템 데이터, 사용자 데이터 및 NC 명령어 (REDEF)의 재정의 (페이지 446)"을 참조하십시오.
  - 사용자 정의 변수
    사용자 정의 변수란 전적으로 사용자에 의해 정의되며 런타임까지 시스템에서 생성되지 않는 변수를 말합니다. 이 변수의 번호, 데이터 유형, 가시성 및 기타 모든 속성은 전적으로 사용자에 의해 정의됩니다.
    "사용자 변수 정의 (DEF) (페이지 439)"를 참조하십시오.

4.1.1.1 시스템 데이터

시스템 데이터는 시스템에 사전 정의된 변수를 포함하며, 제어 시스템의 현재 파라미터 설정에 대한 액세스는 물론 가공 프로그램 및 싸이클의 기계, 제어 및 공정 상태에 대한 액세스를 제공합니다.

전처리 변수

전처리 변수는 전처리 과정 중, 즉 변수를 포함하는 블록이 해석되는 순간 읽히고 쓰여지는 시스템 데이터입니다. 전처리 변수는 전처리 정지를 실행시키지 않습니다.

메인 실행 변수

메인 실행 변수는 메인 실행 중, 즉 변수를 포함하는 블록이 실행되는 순간 읽히고 쓰여지는 시스템 데이터입니다. 다음과 같은 메인 실행 변수가 있습니다.

- 동기 동작에 프로그램 가능한 변수 (읽기/쓰기)
- NC프로그램에 프로그램 가능하며 전처리 정지를 실행시키는 변수 (읽기/쓰기)
- NC 프로그램에 프로그래밍할 수 있고 전처리 중에 값이 계산되지만 메인 실행 전에는 쓰여지지 않는 변수 (메인 실행 동기화: 쓰기 전용)

접두어 시스템

시스템 데이터를 다른 데이터와 구분하기 위해 이름 앞에 보통 $ 표시 다음에 한 개 또는 두 개의 문자 그리고 밑줄이 나오는 접두어가 붙습니다.

$ + 1. 문자	**의미: 데이터 유형**
전처리 데이터 (전처리 중 읽고/쓰여지는 시스템 데이터)	
$M	머신 데이터 [1]
$S	설정 데이터, 보호 영역 [1]
$T	공구 관리 데이터
$P	프로그램된 값
$C	ISO 인벨로프 싸이클의 싸이클 변수
$O	옵션 데이터
R	R-파라미터 (산술 파라미터) [2]
메인 실행 데이터 (메인 실행 중 읽고 쓰여지는 시스템 데이터)	
$$M	머신 데이터 [1]
$$S	설정 데이터 [1]
$A	현재 메인 실행 데이터
$V	위치 제어기 데이터
$R	R-파라미터 (산술 파라미터) [2]

[1] 머신 데이터와 설정 데이터를 전처리 변수와 메인 실행 변수 중 어떤 변수로 처리할지는 $ 문자가 1개인지 2개인지에 따라 결정됩니다. 적용 분야에 따라 표기 방법은 자유롭게 선택할 수 있습니다.

[2] R-파라미터가 가공 프로그램/싸이클에서 전처리 변수로 사용되는 경우 접두어를 생략합니다 (예: R10). R-파라미터가 동기 동작에서 메인 실행 변수로 사용되는 경우 1개의 $ 부호가 접두어로 붙습니다 (예: $R10).

두 번째 문자	의미: 가시성
N	NC 전역 변수(**N**C)
C	채널별 변수 (**C**hannel)
A	축별 변수 (**A**xis)

보충 조건

접두어 체계의 예외

다음 변수 체계는 위에 명시한 접두어 체계와 다릅니다.

- $TC_...: 여기서 두 번째 글자인 C는 채널별 시스템 변수가 아닌 공구 홀더별 시스템 변수(TC=공구 캐리어) 를 의미합니다.
- $P_ ...: 채널별 시스템 변수

동기 동작에서 머신 데이터와 설정 데이터 사용

머신 데이터와 설정 데이터를 동기 동작에서 사용할 때는 머신 데이터와 설정 데이터의 읽기/쓰기가 전처리 실행과 메인 실행 중 어디에 동기화시킬지 여부를 접두어를 사용하여 정의할 수 있습니다.

가공 중 데이터가 변경되지 않는 경우 전처리 실행과 동기적으로 읽을 수 있습니다. 이를 위해 머신 데이터 또는 설정 데이터에 1개의 $ 부호를 접두어로 붙입니다.

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

가공 중 데이터가 변경되는 경우 메인 실행과 동기적으로 읽거나 써야 합니다. 이를 위해 머신 데이터 또는 설정 데이터에 2개의 $ 부호를 접두어로 붙입니다.

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

참고

쓰기 머신 및 설정 데이터

머신 데이터 또는 설정 데이터 항목을 쓰기 위해서는 가공 프로그램/싸이클 실행 시 활성화된 사용 권한이 쓰기 액세스를 허용해야 하고 해당 데이터가 "즉시 적용" 효과를 갖도록 설정해야 합니다.

다음 또한 참고:

변수 (페이지 429)

4.1.1.2 사전 정의된 사용자 변수: 채널별 산술 파라미터 (R)

채널별 산술 파라미터 또는 R 파라미터는 REAL 데이터 유형의 배열을 의미하는 R이 표시된 사전 정의된 사용자 변수입니다. 역사적인 이유 때문에 R-파라미터는 배열 인덱스를 표기할 수도 있고 (예: `R[10]`) 배열 인덱스를 표기하지 않을 수도 (예: `R10`) 있습니다.

동기 동작을 사용하는 경우 R-파라미터에 $ 부호를 접두어로 붙여야 합니다 (예: `$R10`).

구문

전처리 변수로 사용되는 경우:

```
R<n>
R[<expression>]
```

메인 실행 변수로 사용되는 경우:

```
$R<n>
$R[<expression>]
```

의미

R:	가공 프로그램 등에서 전처리 변수로 사용됨을 표시하는 식별자	
$R:	동기 동작 등에서 메인 실행 변수로 사용됨을 표시하는 식별자	
	유형:	REAL
	값 범위:	비-지수 표기: ± (0.000 0001 ... 9999 9999) **참고:** 최대 8 개의 소수 자리 수가 허용됩니다.
		지수 표기: ± ($1*10^{-300}$ ... $1*10^{+300}$) **참고:** • 지수: <Mantissa>EX<Exponent> e.g. 8.2EX-3 • 최대 10 개의 문자가 허용됩니다 (부호와 소수점 포함).

<n>:	R 파라미터 번호	
	유형:	INT
	값 범위:	0 - MAX_INDEX **참고** MAX_INDEX는 설정된 R-파라미터의 개수에 의해 계산됩니다. MAX_INDEX = (MD28050 $MN_MM_NUM_R_PARAM) - 1
<expression>:	배열 인덱스 결과를 INT 데이터 유형 (INT, REAL, BOOL, CHAR) 으로 변환할 수 있는 표현식은 모두 배열 인덱스로 사용할 수 있습니다.	

예

산술 함수로 R-파라미터 지정 및 사용:

프로그램 코드	설명
R0=3.5678	; 전처리에 지정
R[1]=-37.3	; 전처리에 지정
R3=-7	; 전처리에 지정
$R4=-0.1EX-5	; 메인 프로그램 실행 시 지정: R4 = -0.1 * 10^-5
$R[6]=1.874EX8	; 메인 프로그램 실행 시 지정: R6 = 1.874 * 10^8
R7=SIN(25.3)	; 전처리에 지정
R[R2]=R10	; R-파라미터를 사용해 간접 어드레스 지정
R[(R1+R2)*R3]=5	; 산술 표현식을 사용해 간접 어드레스 지정
X=(R1+R2)	; 축 X를 R1과 R2의 합에서 나온 위치로 이송
Z=SQRT(R1*R1+R2*R2)	; 축 Z를 제곱근 위치로 이송 (R1^2 + R2^2)

다음 또한 참고:

변수 (페이지 429)

4.1.1.3 사전 정의된 사용자 변수: 전역 산술 파라미터 (RG)

기능

사용자는 채널별 R 파라미터 외에 전역 R 파라미터에도 액세스할 수 있습니다. 제어 유닛에 있는 이 파라미터를 모든 채널에서 읽거나 쓸 수 있습니다.

예를 들어 하나의 채널에서 다른 채널로 정보를 전송하기 위해 전역 R 파라미터가 사용됩니다. 다른 예는 스핀들에서 미가공 부품 과잉 등 모든 채널에서 평가해야 하는 전역 설정에도 이 파라미터를 사용합니다.

전역 R 파라미터는 전처리 과정에서 사용자 인터페이스나 NC 프로그램에서 읽고 씁니다. 동기 동작과 공정 싸이클은 사용할 수 없습니다.

참고

전역 R 파라미터를 읽고 쓸 때 채널 간 동기화는 **안됩니다**.

읽기 및 쓰기는 전처리 중에 수행되기 때문에 한 채널에서 쓴 값이 다른 채널에서 활성화되는 시점을 정의할 수 없습니다.

예:

채널 1에서 전역 R 파라미터를 이용해 루프를 루프 카운터로 실행합니다. 채널 2에서 이 전역 R 파라미터에 값을 씁니다. 이렇게 되면 채널 1에서 루프가 중단됩니다. 하지만 채널 1에서 전처리 중에 해석 가능한 모든 루프는 계속 실행됩니다. 루프 개수는 정의하지 않고 채널 로딩 등에 따라 달라집니다.

채널 간 동기화는 사용자가 WAIT 플래그 등을 사용해 애플리케이션으로 직접 구현해야 합니다.

구문

NC 프로그램에서 쓰기

```
RG[<n>]=<값>
RG[<표현식>]=<값>
```

NC 프로그램에서 읽기

```
R...=RG[<n>]
R...=RG[<표현식>]
```

의미

RG :	**전역** R 파라미터의 NC 주소의 디폴트 이름 **참조:** NC 주소의 이름은 MD15800 $MN_R_PARAM_NCK_NAME을 통해 설정할 수 있습니다.	
<n>:	전역 R 파라미터의 번호	
	유형:	INT
	값 범위:	0 ... MAX_INDEX **참고** MAX_INDEX는 설정된 전역 R 파라미터 개수에서 다음과 같이 계산됩니다. MAX_INDEX = (MD18156 $MN_MM_NUM_R_PARAM_NCK) - 1
<표현식>:	결과를 INT 데이터 유형 (INT, REAL, BOOL, CHAR) 으로 변환할 수 있는 표현식은 모두 배열 인덱스로 사용할 수 있습니다.	
<값>:	전역 R 파라미터의 값	
	유형:	REAL
	값 범위:	비-지수 표기 ± (0.000 0001 ... 9999 9999) **참고:** 최대 8개의 소수 자리 수가 허용됩니다.
		지수 표기: ± ($1*10^{-300}$ ... $1*10^{+300}$) **참고:** • 표기: <가수>EX<지수> 예: 8.2EX-3 • 부호와 소수점을 포함한 최대 10 문자가 허용됩니다.

4.1.1.4 사전 정의된 사용자 변수: 링크 변수

링크 변수는 네트워크를 통해 서로 링크되어 있는 NCU들이 싸이클 데이터를 교환하는 데 쓰이는 "NCU-링크" 기능에 사용할 수 있습니다. 링크 변수는 링크 변수 메모리에 대한 데이터-형식별 액세스를 가능하게 합니다. 링크 변수 메모리의 크기와 데이터 구조는 사용자/장비 제조업체가 시스템별로 정의합니다.

링크 변수는 링크 통신이 설정된 경우 링크에 연관된 모든 NCU가 가공 프로그램 및 싸이클에서 읽고 쓸 수 있는 시스템 전역 사용자 변수입니다. 전역 사용자 변수 (GUD) 와 달리 링크 변수는 동기 동작에도 사용할 수 있습니다.

활성 NCU 링크가 없는 시스템인 경우 제어 시스템에서 로컬로 전역 사용자 변수 (GUD) 와 함께 추가적인 전역 사용자 변수로 링크 변수를 사용할 수 있습니다.

구문

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

의미

$A_DLB:	BYTE 데이터 형식을 위한 링크 변수 (1 바이트)	
	데이터 유형:	UINT
	값 범위:	0 ... 255
$A_DLW:	WORD 데이터 형식을 위한 링크 변수 (2 바이트)	
	데이터 유형:	INT
	값 범위:	-32768 ... 32767
$A_DLD:	DWORD 데이터 형식을 위한 링크 변수 (4 바이트)	
	데이터 유형:	INT
	값 범위:	-2147483648 ... 2147483647
$A_DLR:	REAL 데이터 형식을 위한 링크 변수 (8 바이트)	
	데이터 유형:	REAL
	값 범위:	$\pm(2.2*10^{-308} \ldots 1.8*10^{+308})$

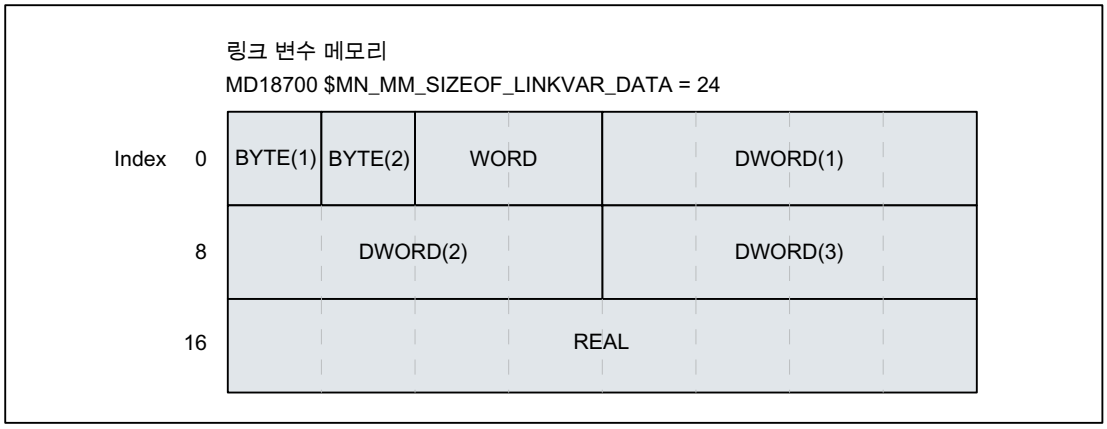
<index>:	바이트 단위 어드레스 인덱스 (링크 변수 메모리의 시작에서 계산)	
	데이터 유형:	INT
	값 범위:	0 - MAX_INDEX **참고** • MAX_INDEX는 파라미터화된 링크 변수 메모리의 크기에서 계산됩니다. MAX_INDEX = (MD18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1 • 인덱스만 프로그래밍이 가능하기 때문에 링크 변수 메모리에 지정된 바이트는 데이터 형식 한계 상에 위치합니다. ⇒ 인덱스 = n * 바이트. 여기서 n = 0, 1, 2, ... – $A_DLB[i]: i = 0, 1, 2, ... – $A_DLW[i]: i = 0, 2, 4, ... – $A_DLD[i]: i = 0, 4, 8, ... – $A_DLR[i]: i = 0, 8, 16, ...

예

자동화 시스템당 2개의 NCU (NCU 1 및 NCU 2) 가 있습니다. 기계 축 AX2는 NCU1에 연결되어 있습니다. AX2는 NCU2의 링크 축으로 취급되어 이송됩니다.

NCU1은 주기적으로 축 AX2의 실제 현재 값 ($VA_CURR) 을 링크 변수 메모리에 기록합니다. NCU2는 링크 통신을 통해 전송되는 실제 현재 값을 주기적으로 읽고 이 값이 한계값을 초과하는 경우 알람 61000을 표시합니다.

다음 그림은 링크 변수 메모리의 데이터 구조를 나타낸 것입니다. 실제 현재 값은 REAL 값 형식으로 전송됩니다.



NCU1

NCU1은 스태틱 동기 동작의 보간 싸이클에서 링크 변수 $A_DLR[16]를 사용하여 축 AX2의 실제 현재 값을 주기적으로 링크 변수 메모리에 씁니다.

프로그램 코드

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

NCU2

NCU2는 스태틱 동기 동작의 보간 싸이클에서 링크 변수 $A_DLR[16]를 사용하여 축 AX2의 실제 현재 값을 주기적으로 링크 변수 메모리에서 읽습니다. 읽은 실제 현재 값이 23.0 A보다 큰 경우 알람 61000이 표시됩니다.

프로그램 코드

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

4.1.1.5 사용자 변수 정의 (DEF)

DEF 명령어를 사용하여 특정 사용자 변수나 사용자 변수(사용자 데이터)를 정의할 수 있고, 변수에 값을 할당할 수 있습니다.

유효성 범위, 즉 변수가 표시되는 범위에 따라 사용자 변수는 다음과 같은 범주로 나뉩니다.

- 로컬 사용자 변수 (LUD)
  로컬 사용자 변수 (LUD)는 실행 시점을 기준으로 메인 프로그램이 아닌 NC 프로그램에서 정의된 변수입니다. LUD는 NC 프로그램이 호출될 때 만들어지고, 프로그램 리셋의 종료와 함께 혹은 제어 시스템에 전원이 켜지는 다음 번에 삭제됩니다. 로컬 사용자 변수는 해당 변수가 정의된 NC 프로그램 내에서만 액세스할 수 있습니다.
- 프로그램 전역 사용자 변수 (PUD)
  프로그램 전역 사용자 변수 (PUD) 는 메인 프로그램으로 사용되는 NC 프로그램에서 정의된 사용자 변수입니다. LUD는 NC 프로그램이 호출될 때 만들어지고, 프로그램 리셋의 종료와 함께 혹은 제어 시스템에 전원이 켜지는 다음 번에 삭제됩니다. PUD는 메인 프로그램과 메인 프로그램의 모든 하부 프로그램에서 액세스할 수 있습니다.

  참고

  프로그램 전역 사용자 변수(PUD)의 사용 가능성

  메인 프로그램에 정의된 프로그램 전역 사용자 변수(PUD)는 다음 머신 데이터가 설정된 경우에만 서브프로그램에서 사용할 수 있습니다.

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  MD11120 = 0이면 메인 프로그램에 정의된 프로그램 전역 사용자 변수(PUD)는 메인 프로그램에서만 사용할 수 있습니다.

- 전역 사용자 변수 (GUD)
  전역 사용자 변수(GUD)는 데이터 블록(SGUD, MGUD, UGUD, GUD4에서 GUD9까지)에서 정의되는 NC 또는 채널-전역 변수이고, 프로그램 리셋 종료 이후 혹은 제어 시스템에 전원이 켜지는 다음 번에 동일하게 유지됩니다. GUD는 모든 NC 프로그램에서 액세스될 수 있습니다.

사용자 변수를 사용하기 위해서는 먼저 정의 (읽기/쓰기) 를 해야 합니다. 사용자 변수를 정의할 때 다음 규칙을 준수해야 합니다.

- GUD는 반드시 정의 파일, 즉 _N_DEF_DIR/_N_MGUD_DEF로 정의되어야 합니다.
- PUD 및 LUD는 NC 프로그램의 정의 섹션에 정의해야 합니다.
- 데이터는 지정된 블록에 정의해야 합니다.
- 각 데이터 정의에 단 하나의 데이터 유형만 사용할 수 있습니다.
- 각 데이터 정의에 데이터 유형이 동일한 변수를 여러 개 정의할 수 있습니다.

구문

LUD 및 PUD

```
DEF <type> <phys_unit> <limit values> <name>[<value_1>,
<value_2>, <value_3>]=<init_value>
```

GUD

```
DEF <range> <pp_stop> <access_rights> <data class> <type>
<phys_unit> <limit values> <name>[<value_1>, <value_2>,
<value_3>]=<init_value>
```

의미

DEF:	사용자 변수 GUD, PUD 및 LUD를 정의하는 명령어.	
<range>:	유효 범위 (GUD만 해당):	
	NC:	NC 전역 사용자 변수
	CHAN:	채널 전역 사용자 변수
<PP_stop>:	전처리 정지 (GUD에만 해당하는 선택사항)	
	SYNR:	읽는 동안 전처리 정지
	SYNW:	쓰는 동안 전처리 정지
	SYNRW:	읽거나 쓰는 동안 전처리 정지
<access rights>:	NC 프로그램 또는 OPI를 통한 GUD 읽기/쓰기 보호 수준 (선택사항)	
	APRP <protection level>:	읽기: NC 프로그램
	APWP <protection level>:	쓰기: NC 프로그램
	APRB <protection level>:	읽기: OPI
	APWB <protection level>:	쓰기: OPI
	<protection level>:	값 범위: 0 ... 7
	"속성: 사용 권한 (APR, APW, APRP, APWP, APRB, APWB) (페이지 457)"를 참조하십시오.	

<데이터 클래스>:	데이터 클래스 지정 (SINUMERIK 828D만 해당)	
	DCM:	데이터 클래스 M (= 제조업체)
	DCI:	데이터 클래스 I (= 개별)
	DCU:	데이터 클래스 U (= 사용자)
<type>:	데이터 유형:	
	INT:	부호가 있는 정수
	REAL:	실수 (LONG REAL~IEEE)
	BOOL:	참 값 TRUE (1) / FALSE (0)
	CHAR:	ASCII 문자
	STRING[<MaxLength>]:	길이가 정의된 문자열
	AXIS:	축 이름/스핀들 이름
	FRAME:	정적 좌표 변환을 위한 기하 데이터
	"데이터 유형 (페이지 470)"를 참조하십시오.	
<phys_unit>:	물리 단위 (선택사항)	
	PHU <unit>:	물리 단위
	"속성: 물리 단위 (PHU) (페이지 455)"를 참조하십시오.	
<limit values>:	상한값/하한값 (선택사항)	
	LLI <limit value>:	하한값
	ULI <limit value>:	상한값
	"속성: 한계값 (LLI, ULI) (페이지 453)"를 참조하십시오.	
<name>:	변수 이름 **참고** • 최대 31개 문자 • 첫 두 문자는 글자 및/또는 밑줄이어야 합니다. • $ 부호는 시스템 변수를 위한 것이므로 사용하지 않도록 합니다.	

[<value_1>, <value_2>, <value_3>]:	1차원부터 최대 3차원 배열 변수의 배열 크기 지정 (선택사항) 배열 변수 초기화에 대한 설명은 "배열 변수 정의 및 초기화 (DEF, SET, REP) (페이지 464)"를 참조하십시오.
<init_value>:	초기화 값 (선택사항) "속성: 초기화 값 (페이지 450)"를 참조하십시오. 배열 변수 초기화에 대한 설명은 "배열 변수 정의 및 초기화 (DEF, SET, REP) (페이지 464)"를 참조하십시오.

예

예 1: 데이터 블록에 장비 제조업체의 사용자 변수 정의

프로그램 코드	설명
%_N_MGUD_DEF	; GUD 블록: 장비 제조업체
$PATH=/_N_DEF_DIR	
DEF CHAN REAL PHU 24 LLI 0 ULI 10 CURRENT_1, CURRENT_2	
;설명	
;2개의 GUD 항목 정의: CURRENT_1, CURRENT_2	
;유효 범위: 채널 전체	
;데이터 유형: REAL	
PP 정지: 프로그래밍 안됨 => 디폴트 값 = PP 정지 없음	
; 물리 단위: 24 = [A]	
;한계값: 하한값 = 0.0, 상한값 = 10.0	
;사용 권한: 프로그래밍 안됨 => 디폴트 값 = 7 = 키 스위치 위치 0	
;초기화 값: 프로그래밍 안됨 => 디폴트 값 = 0.0	
DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 TIME_1=12, TIME_2=45	
;설명	
;2개의 GUD 항목 정의: TIME_1, TIME_2	
;유효 범위: NC 전체	
;데이터 유형: REAL	
PP 정지: 프로그래밍 안됨 => 디폴트 값 = PP 정지 없음	
; 물리 단위: 13 = [s]	
;한계값: 하한값 = 10.0, 상한값 = 프로그래밍 안됨 => 정의 범위 상한값	
;사용 권한:	
; NC 프로그램: 읽기/쓰기 = 3 = 사용자	
; OPI: 쓰기 = 0 = Siemens, 읽기 = 3 = 사용자	
;초기화 값: TIME_1 = 12.0, TIME_2 = 45.0	
DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER"	
;설명	
; 1개의 GUD 항목 정의: GUD5_NAME	

프로그램 코드	설명
;유효 범위: NC 전체	
;데이터 유형: 문자열 (최대 5자)	
PP 정지: 프로그래밍 안됨 => 디폴트 값 = PP 정지 없음	
; 물리 단위: 프로그래밍 안됨 => 디폴트 값 = 0 = 물리 단위 없음	
;한계값: 프로그래밍 안됨 => 정의 범위 한계: 하한값 = 0, 상한값 = 255	
;사용 권한:	
; NC 프로그램: 읽기/쓰기 = 3 = 사용자	
; OPI: 쓰기 = 0 = Siemens, 읽기 = 3 = 사용자	
;초기화 값: "COUNTER"	
M30	

예 2: 프로그램 전역 사용자 변수 및 로컬 사용자 변수 (PUD/LUD)

프로그램 코드	설명
PROC MAIN	; 메인 프로그램
DEF INT VAR1	; PUD 정의
...	
SUB2	;서브프로그램 호출
...	
M30	

프로그램 코드	설명
PROC SUB2	; 서브프로그램 SUB2
DEF INT VAR2	; LUD 정의
...	
IF (VAR1==1)	; PUD 읽기
VAR1=VAR1+1	;PUD 읽기 및 쓰기
VAR2=1	;LUD 쓰기
ENDIF	
SUB3	;서브프로그램 호출
...	
M17	

프로그램 코드	설명
PROC SUB3	; 서브프로그램 SUB3
...	
IF (VAR1==1)	; PUD 읽기
VAR1=VAR1+1	;PUD 읽기 및 쓰기
VAR2=1	; 에러: 알 수 없는 SUB2의 LUD
ENDIF	
...	
M17	

예 3: AXIS 데이터 유형의 사용자 변수 정의 및 사용

프로그램 코드	설명
DEF AXIS ABSCISSA	; 첫 번째 기하 축
DEF AXIS SPINDLE	; 스핀들
...	
IF ISAXIS(1) == FALSE GOTOF CONTINUE	
ABSCISSA = $P_AXN1	
CONTINUE:	
...	
SPINDLE=(S1)	; 첫 번째 스핀들
OVRA[SPINDLE]=80	; 스핀들 오버라이드 = 80%
SPINDLE=(S3)	; 세 번째 스핀들

보충 조건

전역 사용자 변수 (GUD)

전역 사용자 변수 정의 시 다음의 머신 데이터를 고려해야 합니다.

번호	식별자: $MN_	의미
11140	GUD_AREA_ SAVE_TAB	GUD 블록을 위한 추가 저장
18118 [1)]	MM_NUM_GUD_MODULES	활성 파일 시스템 내 GUD 파일의 개수
18120 [1)]	MM_NUM_GUD_NAMES_NCK	전역 GUD 이름의 개수
18130 [1)]	MM_NUM_GUD_NAMES_CHAN	채널 GUD 이름의 개수
18150 [1)]	MM_GUD_VALUES_MEM	전역 GUD 값의 메모리 위치
18660 [1)]	MM_NUM_SYNACT_GUD_REAL	REAL 데이터 유형의 설정 가능한 GUD 개수
18661 [1)]	MM_NUM_SYNACT_GUD_INT	INT 데이터 유형의 설정 가능한 GUD 개수
18662 [1)]	MM_NUM_SYNACT_GUD_BOOL	BOOL 데이터 유형의 설정 가능한 GUD 개수
18663 [1)]	MM_NUM_SYNACT_GUD_AXIS	AXIS 데이터 유형의 설정 가능한 GUD 개수
18664 [1)]	MM_NUM_SYNACT_GUD_CHAR	CHAR 데이터 유형의 설정 가능한 GUD 개수
18665 [1)]	MM_NUM_SYNACT_GUD_STRING	STRING 데이터 유형의 설정 가능한 GUD 개수

[1)] SINUMERIK 828D의 경우 MD만 읽을 수 있습니다!

AXIS 데이터 유형의 NC 전역 사용자 변수를 여러 채널에 사용

축 식별자를 갖는 데이터 블록에서의 정의 중 초기화된 AXIS 데이터 유형의 NC 전역 사용자 변수는 해당 축이 다른 여러 NC 채널에서 동일한 채널 축 번호를 갖고 있는 경우에만 이러한 다른 NC 채널에서 사용될 수 있습니다.

그렇지 않은 경우 해당 변수는 NC 프로그램의 시작에서 로드되거나 다음 예의 경우처럼 AXNAME(...) 기능 ("축 기능(AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (페이지 995)" 참조) 을 사용해야 합니다.

프로그램 코드	설명
DEF NCK STRING[5] ACHSE="X"	;데이터 블록에 정의
...	
N100 AX[AXNAME(AXIS)]=111 G00	; NC 프로그램에서 사용

4.1.1.6 시스템 데이터, 사용자 데이터 및 NC 명령어 (REDEF)의 재정의

REDEF 명령어는 시스템 데이터, 사용자 데이터 및 NC 명령어의 속성을 변경합니다. 재 정의를 위해서는 기본적으로 해당 정의보다 재 정의 날짜가 더 늦어야 합니다.

재 정의 중에 여러 속성을 동시에 변경할 수 없습니다. 변경할 속성마다 별도의 REDEF 명령을 프로그래밍해야 합니다.

여러 속성을 동시에 변경하도록 프로그래밍한 경우 항상 마지막 변경이 활성화됩니다.

속성 값 리셋

REDEF를 사용하여 설정한 사용 권한 및 초기화 시간 변경에 대한 속성을 변수 이름 또는 NC 언어 명령 앞에 REDEF를 다시 프로그래밍하여 기본 값으로 리셋할 수 있습니다.

- 사용 권한: 사용 권한 7
- 초기화 시간: 현재 값 초기화 또는 유지 안함

재 정의 가능한 속성

"정의 가능 속성 및 재정의 가능 속성 개요 (페이지 462)"을 참조하십시오.

로컬 사용자 변수 (PUD/LUD)

로컬 사용자 변수 (PUD/LUD) 는 재 정의가 허용되지 않습니다.

구문

```
REDEF <name> <PP_stop>
REDEF <name> <phys_unit>
REDEF <name> <limit_values>
```

```
REDEF <name> <access_rights>
REDEF <name> <init_time>
REDEF <name> <init_time> <init_value>
REDEF <name> <data class>
REDEF <name>
```

의미

REDEF:	특정 속성 재정의 명령 또는 시스템 변수, 사용자 변수 및 NC 언어 명령의 "사용 권한" 및/또는 "초기화 시간" 속성 리셋 명령	
<name>:	사전 정의된 변수 또는 NC 언어 명령의 이름	
<PP stop>:	전처리 정지	
	SYNR:	읽는 동안 전처리 정지
	SYNW:	쓰는 동안 전처리 정지
	SYNRW:	읽거나 쓰는 동안 전처리 정지
<phys_unit>:	물리 단위	
	PHU <unit>:	물리 단위
	"속성: 물리 단위 (PHU) (페이지 455)"을 참조하십시오. **참고** 다음은 재정의할 수 없습니다. • 시스템 변수 • 데이터 유형에 대한 전역 사용자 데이터 (GUD): BOOL, AXIS, STRING, FRAME	
<limit values>:	상한값/하한값	
	LLI <limit value>:	하한값
	ULI <limit value>:	상한값
	"속성: 한계값 (LLI, ULI) (페이지 453)"을 참조하십시오. **참고** 다음은 재정의할 수 없습니다. • 시스템 변수 • 데이터 유형에 대한 전역 사용자 데이터 (GUD): BOOL, AXIS, STRING, FRAME	

<access rights>:	가공 프로그램 또는 OPI를 통한 읽기/쓰기 사용 권한	
	APX <protection level>:	실행: NC 언어 요소
	APRP <protection level>:	읽기: 가공 프로그램
	APWP <protection level>:	쓰기: 가공 프로그램
	APRB <protection level>:	읽기: OPI
	APWB <protection level>:	쓰기: OPI
	<protection level>:	값 범위: 0 ... 7
	"속성: 사용 권한 (APR, APW, APRP, APWP, APRB, APWB) (페이지 457)"을 참조하십시오.	
<init_time>:	변수가 다시 초기화되는 시점	
	INIPO:	전원 ON
	INIRE:	메인 프로그램 종료, NC 리셋 또는 전원 ON
	INICF:	NEWCONF 또는 메인 프로그램 종료, NC 리셋 또는 전원 ON
	PRLOC:	메인 프로그램 종료, 로컬 변경 후 NC 리셋 또는 전원 ON
	"속성: 초기화 값 (페이지 450)"을 참조하십시오.	
<init_value>:	초기화 값 초기화 값을 재 정의할 때 항상 초기화 시간을 함께 지정해야 합니다 (<init_time> 참조). "속성: 초기화 값 (페이지 450)"을 참조하십시오. 배열 변수들의 초기화에 대해서는 "배열 변수 정의 및 초기화 (DEF, SET, REP) (페이지 464)"를 참조하십시오. **참고** 설정 데이터를 제외한 시스템 변수는 재정의할 수 없습니다.	
<data class>:	데이터 클래스 지정 (SINUMERIK 828D만 해당)	
	DCM:	데이터 클래스 M (= 제조업체)
	DCI:	데이터 클래스 I (= 개별)
	DCU:	데이터 클래스 U (= 사용자)

예

기계 제조업체를 위한 데이터 블록에 시스템 변수 $TC_DPCx 재정의

프로그램 코드

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

N100 관 쓰기 액세스: OPI = 사용 권한 2, 가공 프로그램 = 사용 권한 3
련:

N200 관 물리 단위 [%]
련:

N300 관 하한값 = 0, 상한값 = 200
련:

N400 관 배열 변수는 POWER ON 시 4개의 값으로 초기화합니다.
련:

N800 / "사용 권한" 및/또는 "초기화 시간" 속성 값 리셋
N900 관
련

참고

액세스 파일 사용

액세스 파일이 사용되면 사용 권한에 대한 재정의가 _N_MGUD_DEF에서 _N_MACCESS_DEF로 이동해야 합니다.

보충 조건

입도

재 정의는 항상 그 이름으로 식별되는 전체 변수에 적용됩니다. 예를 들어 배열 변수를 사용해서 개별 배열 요소에 다른 속성을 지정할 수는 없습니다.

4.1.1.7 속성: 초기화 값

사용자 변수의 정의 (DEF)

정의 과정에서 다음의 사용자 변수에 초기화 값을 미리 지정할 수 있습니다.

- 전역 사용자 변수 (GUD)
- 프로그램 전역 사용자 변수 (PUD)
- 로컬 사용자 변수 (LUD)

시스템 변수 및 사용자 변수의 재 정의 (REDEF)

재정의 과정에서 다음의 변수에 초기화 값을 미리 지정할 수 있습니다.

- 시스템 데이터
  - 셋팅 데이터
- 사용자 데이터
  - R 파라미터
  - 동기 동작 변수 ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - 동기 동작 GUD (SYG_xy[ ], 여기서 x=R, I, B, A, C, S, y=S, M, U, 4~9)
  - EPS 파라미터
  - 공구 데이터 OEM
  - 메거진 데이터 OEM
  - 전역 사용자 변수 (GUD)

재초기화 시간

재정의 과정에서 변수를 재초기화, 즉 초기화 값으로 리셋해야 하는 시점을 지정할 수 있습니다.

- INIPO(POWER ON)
  전원 ON 시 변수를 다시 초기화합니다.
- INIRE (리셋)
  NC 리셋, 모드 그룹 리셋, 가공 프로그램 종료 (M02/M30) 또는 전원 ON 시 변수를 다시 초기화합니다.

- INICF (NEWCONF)
  "머신 데이터 활성 설정" 기능의 경우, 변수는 HMI, 가공 프로그램 명령 NEWCONF 또는 NC 리셋, 모드 그룹 리셋, 가공 프로그램 종료(M02 / M30) 또는 전원 ON을 통해 다시 초기화 됩니다.
- PRLOC (프로그램 로컬 변경)
  해당 변수는 현재 가공 프로그램에서 변경된 경우 NC 리셋, 모드 그룹 리셋 또는 가공 프로그램의 끝 (M02/M30) 에서만 다시 초기화 됩니다.
  PRLOC 속성은 프로그램 가능한 셋팅 데이터를 통해서만 변경이 가능합니다 (아래 표 참조).

도표 4-1 프로그램 가능한 셋팅 데이터

번호	식별자	G 명령어 [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

번호	식별자	G 명령어 1)
43790	$SA_OSCILL_START_POS	OSB
1) 이 G 명령은 셋팅 데이터를 지정합니다.		

보충 조건

초기화 값: 전역 사용자 변수 (GUD)

- NC 유효 범위의 전역 사용자 변수(GUD)인 경우 INIPO (전원 ON)만 초기화 시간으로 정의할 수 있습니다.
- INIPO (전원 ON)이외에도 CHAN 유효 범위의 전역 사용자 변수(GUD)인 경우, INIRE (리셋) 또는 INICF (NewConfig) 를 초기화 시간으로 정의할 수 있습니다.
- INIRE (리셋) 또는 INICF (NEWCONF) 초기화 시간 그리고 CHAN 유효범위를 가진 전역 사용자 변수(GUD)의 경우, NC 리셋, 모드 그룹 리셋 및 "머신 데이터 활성"이 촉발된 채널에서만 변수가 다시 초기화 됩니다.

초기화 값: FRAME 데이터 유형

FRAME 데이터 유형의 변수인 경우 초기화 값을 지정할 수 없습니다. FRAME 데이터 유형의 변수는 암시적으로, 항상 디폴트 프레임에 의해 초기화됩니다.

초기화 값: CHAR 데이터 유형

CHAR 데이터 유형의 변수인 경우, ASCII 코드(0...255) 대신 해당하는 ASCII 문자를 큰 따옴표 안에 프로그램 할 수 있습니다(예: "A").

초기화 값: STRING 데이터 유형

STRING 데이터 유형의 변수인 경우 문자열은 큰 따옴표 안에 있어야 합니다 (예: ...= "MACHINE_1").

초기화 값: AXIS 데이터 유형

AXIS 데이터 유형의 변수인 경우 확장 어드레스를 지정할 때 축 이름은 괄호 안에 있어야 합니다 (예: ...=(X3)).

초기화 값: 시스템 변수

시스템 변수의 경우 재 정의를 통해 사용자 지정 초기화 값을 정의할 수 없습니다. 시스템 변수의 초기화 값은 시스템이 지정하기 때문에 변경할 수 없습니다. 하지만 재 정의를 통해 시스템 변수가 다시 초기화되는 시점 (INIRE, INICF) 은 변경할 수 있습니다.

암시적 초기화 값: AXIS 데이터 유형

AXIS 데이터 유형의 변수에는 다음과 같은 암시적 초기화 값이 사용됩니다.

- 시스템 데이터: "첫 번째 기하 축"
- 동기 동작 GUD (표기: SYG_A*), PUD, LUD:
  머신 데이터의 축 이름: MD20082 $MC_AXCONF_CHANAX_DEFAULT_NAME

암시적 초기화 값: 공구 및 메거진 데이터

공구 및 메거진 데이터의 초기화 값은 다음의 머신 데이터를 사용해 정의할 수 있습니다.
MD17520 $MN_TOOL_DEFAULT_DATA_MASK

참고

동기화

전역 변수를 다른 위치에서 읽어올 때 전역 변수의 재초기화를 실행하는 이벤트를 동기화하는 것은 전적으로 사용자/장비 제조업체의 책임입니다.

다음 또한 참고:

변수 (페이지 429)

4.1.1.8 속성: 한계값 (LLI, ULI)

정의 범위의 상한값 및 하한값은 다음의 데이터 유형에 대해서만 정의할 수 있습니다.

- INT
- REAL
- CHAR

사용자 변수의 정의 (DEF) 한계값 및 암시적 초기화 값

위와 같은 데이터 유형의 사용자 변수를 정의할 때 명시적 초기화 값이 정의되어 있지 않은 경우, 변수는 해당 데이터 유형의 암시적 초기화 값으로 설정됩니다.

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

암시적 초기화 값이 프로그래밍한 한계값의 정의 범위를 벗어나는 경우, 해당 변수는 암시적 초기화 값에 가장 가까운 한계값으로 초기화됩니다.

- 암시적 초기화 값 < 하한값 (LLI) ⇒
  초기화 값 = 하한값
- 암시적 초기화 값 > 상한값 (ULI) ⇒
  초기화 값 = 상한값

예:

프로그램 코드	코멘트
DEF REAL GUD1	; 하한값 = 정의 범위 한계 ; 상한값 = 정의 범위 한계 ; 프로그래밍된 초기화 값 없음 ; => 암시적 초기화 값 = 0.0
DEF REAL LLI 5.0 GUD2	; 하한값 = 5.0 ; 상한값 = 정의 범위 한계 ; => 초기화 값 = 5.0
DEF REAL ULI -5 GUD3	; 하한값 = 정의 범위 한계 ; 상한값 = -5.0 ; => 초기화 값 = -5.0

사용자 변수의 재 정의 (REDEF) 한계값 및 현재 실제 값

사용자 변수의 한계값을 다시 정의할 때 현재 실제 값이 새 정의 범위를 벗어나게 정의하면 알람이 발생하고 한계값이 거부됩니다.

참고

사용자 변수의 재 정의 (REDEF)

사용자 변수의 한계값을 다시 정의할 때 반드시 다음 값이 일관되게 변경되도록 주의해야 합니다.

- 한계값
- 실제값
- INIPO, INIRE 또는 INICF를 기준으로 한 재 정의 및 자동 재초기화의 초기화 값

다음 또한 참고:

변수 (페이지 429)

4.1.1.9 속성: 물리 단위 (PHU)

다음과 같은 데이터 유형의 변수만 물리 단위로 지정할 수 있습니다.

- INT
- REAL

프로그램 가능한 물리 단위 (PHU)

물리 단위는 고정된 자연수로 정의합니다. `PHU <unit>`

다음과 같은 물리 단위를 프로그래밍할 수 있습니다.

<unit>	의미	물리 단위
0	물리 단위 아님	-
1	직선위치 또는 각위치 1)2)	[mm], [inch], [°]
2	직선 위치2)	[mm], [inch]
3	각위치	[°]
4	직선속도 또는 각속도 1)2)	[mm/min], [inch/min], [rpm]
5	직선 속도2)	[mm/min]
6	각속도	[rpm]
7	직선가속 또는 각가속 1)2)	[m/s^2], [$inch/s^2$], [rev/s^2]
8	직선 가속2)	[m/s^2], [$inch/s^2$]
9	각가속	[rev/s^2]
10	직선저크 또는 각저크 1)2)	[m/s^3], [$inch/s^3$], [rev/s^3]
11	직선 저크2)	[m/s^3], [$inch/s^3$]
12	각저크	[rev/s^3]
13	시간	[s]
14	위치 제어기 게인	[16.667/s]
15	회전당 이송 속도 2)	[mm/rev], [inch/rev]
16	온도 보정 1)2)	[mm], [inch]
18	힘	[N]
19	질량	[kg]
20	관성 모멘트 3)	[kgm^2]
21	퍼센트	[%]
22	주파수	[Hz]

<unit>	의미	물리 단위
23	전압	[V]
24	전류	[A]
25	온도	[°C]
26	각도	[°]
27	KV	[1000/min]
28	직선위치 또는 각위치 3)	[mm], [inch], [°]
29	절삭 속도 2)	[m/min], [feet/min]
30	원주 속도 2)	[m/s], [feet/s]
31	저항	[ohm]
32	유도 계수	[mH]
33	토크 3)	[Nm]
34	토크 상수 3)	[Nm/A]
35	전류 제어기 게인	[V/A]
36	속도 제어기 게인 3)	[Nm/(rad*s)]
37	속도	[rpm]
42	전원	[kW]
43	미세 전류	[µA]
46	미세 토크 3)	[µNm]
48	밀 당	-
49	-	[Hz/s]
65	유속	[l/min]
66	압력	[bar]
67	부피 3	[cm^3]
68	시스템 게인 3)	[mm/(V*min)]
69	힘 제어기로 시스템 게인	[N/V]
155	나사 리드 3)	[mm/rev], [inch/rev]
156	나사 리드 변경 3)	[mm/rev / rev], [inch/rev / rev]
1) 물리 단위는 축의 유형, 즉 직선 축인지 로터리 축인지에 따라 달라집니다.		

<unit>	의미	물리 단위
2) 단위 체계 전환 G70/G71 (인치/미터) 길이와 관련된 시스템 변수 및 사용자 변수에 읽기/쓰기 작업을 하기 위해 베이직 시스템 (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) 을 G70/G71로 전환한 후에는 값이 변환되지 **않습니다** (실제 값, 디폴트 값, 한계값). G700/G710 (인치/미터) 길이와 관련된 시스템 변수 및 사용자 변수에 읽기/쓰기 작업을 하기 위해 베이직 시스템 (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) 을 G700/G710로 전환한 후에 값이 **변환됩니다** (실제 값, 디폴트 값, 한계값).		
3) 변수가 현재 NC 측정 시스템 (인치/미터) 에 따라 자동으로 변환되지 **않습니다**. 측정 방식 전환은 전적으로 사용자/장비 제조업체의 책임입니다.		

참고

형식 변환에 따른 레벨 오버플로우

길이를 표시하는 물리 단위를 포함하는 모든 사용자 변수 (GUD/PUD/LUD) 는 내부적으로 미터 형식으로 저장됩니다. NCK 메인 실행 (예: 동기 동작) 에 이런 유형의 변수를 지나치게 사용하면 측정 시스템 변환으로 인해 보간 단계에서 CPU 타임 오버플로우가 발생하고 알람 4240이 출력됩니다.

참고

단위 호환성

변수 (지정, 비교, 계산 등) 를 사용할 때 관련 단위의 호환성은 점검하지 않습니다. 변환이 필요한 경우 호환 여부는 전적으로 사용자/장비 제조업체의 책임입니다.

다음 또한 참고:

변수 (페이지 429)

4.1.1.10 속성: 사용 권한 (APR, APW, APRP, APWP, APRB, APWB)

명칭

액세스 속성 AP... 명칭은 다음으로 구성됩니다.

1. A: 액세스(**A**ccess)
2. P: **보호**
3. R / W: 읽기(**R**ead) / 쓰기(**W**rite)
4. P / O: **P**rogram / **B**TSS (OPI)

액세스 권한 / 액세스 수준

다음의 액세스 수준은 프로그래밍할 때 지정되어야 하며 그에 해당되는 사용 권한을 가집니다.

사용 권한	사용 권한 레벨
시스템 암호	0
장비 제조업체 암호	1
서비스 암호	2
사용자 암호	3
키 스위치 위치 3	4
키 스위치 위치 2	5
키 스위치 위치 1	6
키 스위치 위치 0	7

사용자 데이터의 정의 (DEF)

다음의 변수에 대해 사용 권한 (APR.../APW...)을 정의할 수 있습니다:

- 전역 사용자 데이터 (GUD)

시스템 및 사용자 데이터의 재정의 (REDEF)

다음의 변수에 대해 사용 권한 (APR.../APW...)을 재정의할 수 있습니다:

- 시스템 데이터
  - 머신 데이터

    참고

    머신 데이터의 읽기 권한에 대한 재정의

    머신 데이터 읽기 관련 보호 레벨은 가공 프로그램 및 OPI를 위한 APR 키워드를 사용해야 설정할 수 있습니다.

    키워드 APRP 및 APRB는 읽기 권한의 재정의에 의해 지원되지 않으며, 알람 12490 'Access right APRP/APRB <protection level> was not set'가 표시됩니다.
  - 설정 데이터
  - 시스템 변수
  - 공정 데이터
  - 매거진 데이터
  - 공구 데이터
- 사용자 데이터
  - R 파라미터
  - 동기 동작 변수 ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - 동기 동작 GUD (SYG_xy[ ], 여기서 x=R, I, B, A, C, S, y=S, M, U, 4~9)
  - EPS 파라미터
  - 공구 데이터 OEM
  - 매거진 데이터 OEM
  - 전역 사용자 변수 (GUD)

    참고

    해당 변수를 재 정의할 때 최저 사용 권한 레벨 7부터 전용 사용 권한 레벨 1 (장비 제조업체) 사이의 사용 권한을 자유롭게 변수에 지정할 수 있습니다.

NC 언어 명령 재 정의 (REDEF)

다음의 NC 언어 명령에 대해 사용 권한 또는 실행 권한 (APX) 을 재 정의할 수 있습니다.

- G 명령어/준비 기능 (페이지 189)
- 미리 정의된 함수 (페이지 1524)
- 사전 정의된 서브프로그램 호출

- 동기 동작으로 수행하는 DO 작업
- 싸이클 프로그램 식별자
  싸이클은 싸이클 디렉토리 중 하나에 저장되어야 하며 PROC 연산을 포함해야 합니다.

NC 프로그램 및 싸이클과 관련된 사용 권한 (APRP, APWP)

다양한 사용 권한은 NC 프로그램이나 싸이클로부터의 액세스와 관련하여 다음을 가능하도록 해줍니다.

- APRP 0/APWP 0
  - NC 프로그램 처리 중에 시스템 암호를 설정해야 합니다.
  - 해당 싸이클은 _N_CST_DIR 디렉토리 (시스템) 에 저장되어야 합니다.
  - _N_CST_DIR 디렉토리의 경우, MD11160 $MN_ACCESS_EXEC_CST에서 시스템 레벨로 사용 권한이 설정되어야 합니다.
- APRP 1/APWP 1 or APRP 2/APWP 2
  - NC 프로그램 처리 중에 장비 제조업체 암호 또는 서비스 암호를 설정해야 합니다.
  - 해당 싸이클은 _N_CMA_DIR (장비 제조업체) 또는 _N_CST_DIR 디렉토리에 저장되어야 합니다.
  - _N_CMA_DIR 또는 _N_CST_DIR 디렉토리의 경우, MD11161 $MN_ACCESS_EXEC_CMA 또는 MD11160 $MN_ACCESS_EXEC_CST에서 최소한 장비 제조업체 레벨로 사용 권한이 설정되어야 합니다.
- APRP 3/APWP 3
  - NC 프로그램 실행 중 사용자 비밀번호를 반드시 설정해야 합니다.
  - 해당 싸이클은 _N_CUS_DIR (사용자), _N_CMA_DIR 또는 _N_CST_DIR 디렉토리에 저장해야 합니다.
  - _N_CUS_DIR, _N_CMA_DIR 또는 _N_CST_DIR 디렉토리의 경우, MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA 또는 MD11160 $MN_ACCESS_EXEC_CST에서 최소한 사용자 레벨로 사용 권한이 설정되어야 합니다.
- APRP 4...7 / APWP 4...7
  - NC 프로그램 처리 중에 키 스위치를 3 ... 0으로 설정해야 합니다.
  - 해당 싸이클은 _N_CUS_DIR, _N_CMA_DIR 또는 _N_CST_DIR 디렉토리에 저장해야 합니다.
  - _N_CUS_DIR, _N_CMA_DIR 또는 _N_CST_DIR 디렉토리의 경우, MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA 또는 MD11160 $MN_ACCESS_EXEC_CST에서 최소한 관련 키 스위치 위치에 맞는 사용 권한이 설정되어야 합니다.

OPI (APRB, APWB) 와 관련된 사용 권한

사용 권한 (APRB, APWB) 은 OPI는 물론 모든 시스템 구성 요소 (HMI, PLC, 외부 컴퓨터, EPS 서비스 등) 를 통한 시스템 및 사용자 변수 액세스를 제한합니다.

참고

로컬 HMI 사용 권한

시스템 데이터 상의 사용 권한을 변경할 때 변경 사항이 반드시 HMI 메커니즘을 통해 정의한 사용 권한과 일치하도록 해야 합니다.

APR/APW 액세스 속성

호환성을 위해 속성 APR 및 APW 가 암시적으로 속성 APRP / APRB 및 APWP / APWB에 매핑됩니다.

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

ACCESS 파일을 사용한 사용 권한

액세스 권한을 할당하기 위해 액세스 파일을 사용하는 경우, 시스템 데이터, 사용자 데이터 및 NC 언어 명령어에 대한 액세스 권한을 액세스 파일에서 반드시 재정의해야 합니다. 전역 사용자 데이터 (GUD) 는 예외입니다. 이 데이터의 경우 액세스 권한은 해당 정의 파일 *_DEF 에서 재정의해야 합니다.

지속적인 액세스 보호를 위해서는 해당 디렉토리의 실행 권한 및 액세스 보호 관련 머신 데이터를 지속적으로 수정해야 합니다.

원칙적으로 액세스 보호 절차는 다음과 같습니다.

1. 필요한 정의 파일 생성:
   - _N_DEF_DIR/_N_SACCESS_DEF
   - _N_DEF_DIR/_N_MACCESS_DEF
   - _N_DEF_DIR/_N_UACCESS_DEF
2. 재 정의를 위해 필요한 값에 정의 파일에 대한 쓰기 권한 설정
   - MD11170 $MN_ACCESS_WRITE_SACCESS = <protection level>
   - MD11171 $MN_ACCESS_WRITE_MACCESS = <protection level>
   - MD11172 $MN_ACCESS_WRITE_MACCESS = <protection level>

3. 싸이클에서 보호 요소에 액세스하기 위해서는 싸이클 디렉토리 _N_CST_DIR, _N_CMA_DIR 및 _N_CST_DIR의 실행 권한 및 쓰기 권한을 수정해야 합니다.
   실행 권한
   - MD11160 $MN_ACCESS_WRITE_MACCESS = <protection level>
   - MD11161 $MN_ACCESS_WRITE_MACCESS = <protection level>
   - MD11162 $MN_ACCESS_EXEC_CUS = <protection level>

   쓰기 권한
   - MD11165 $MN_ACCESS_WRITE_MACCESS = <protection level>
   - MD11166 $MN_ACCESS_WRITE_MACCESS = <protection level>
   - MD11167 $MN_ACCESS_EXEC_CUS = <protection level>

   실행 권한은 적어도 사용되는 요소의 최고 사용 권한과 동일한 레벨의 사용 권한으로 설정되어야 합니다.
   쓰기 권한은 적어도 실행 권한과 동일한 레벨의 사용 권한으로 설정되어야 합니다.
4. 로컬 HMI 싸이클 디렉토리의 쓰기 권한은 로컬 NC 싸이클 디렉토리와 동일한 레벨의 사용 권한으로 설정되어야 합니다.

ACCESS 파일에 서브프로그램 호출

액세스 보호를 추가적으로 설정하기 위해 ACCESS 파일에 서브프로그램 (SPF 또는 MPF 식별자) 을 호출할 수 있습니다. 서브프로그램은 호출하는 ACCESS 파일의 실행 권한을 그대로 물려 받습니다.

참고

ACCESS 파일에서는 사용 권한만 다시 정의할 수 있습니다. 다른 모든 속성은 계속해서 해당 정의 파일에서 프로그래밍하거나 다시 정의해야 합니다.

다음 또한 참고:

변수 (페이지 429)

4.1.1.11 정의 가능 속성 및 재정의 가능 속성 개요

다음 표는 어떤 데이터 유형의 어떤 속성을 정의 (DEF) 및/또는 재정의 (REDEF) 할 수 있는지 보여줍니다.

시스템 데이터

데이터 유형	초기값	한계값	물리 단위	사용 권한	데이터 클래스 (828D에만 해당)
머신 데이터	---	---	---	REDEF	REDEF
셋팅 데이터	REDEF	---	---	REDEF	---
FRAME 데이터	---	---	---	REDEF	---
공정 데이터	---	---	---	REDEF	---
리드스크류 에러 보정 (EEC)	---	---	---	REDEF	---
처짐 보정 (CEC)	---	---	---	REDEF	---
상한 돌기 보정 (QEC)	---	---	---	REDEF	---
매거진 데이터	---	---	---	REDEF	---
공구 데이터	---	---	---	REDEF	---
보호 영역	---	---	---	REDEF	---
오리엔테이션 조정 가능한 공구 홀더	---	---	---	REDEF	---
운동학적 체인	---	---	---	REDEF	---
3D 보호 영역	---	---	---	REDEF	---
작업 영역 제한	---	---	---	REDEF	---

사용자 데이터

데이터 유형	초기값	한계값	물리 단위	사용 권한	데이터 클래스
R 파라미터	REDEF	REDEF	REDEF	REDEF	---
동기 동작 변수 ($AC_...)	REDEF	REDEF	REDEF	REDEF	---
동기 동작 GUD (SYG_...)	REDEF	REDEF	REDEF	REDEF	---
EPS 파라미터	REDEF	REDEF	REDEF	REDEF	---
공구 데이터 OEM	REDEF	REDEF	REDEF	REDEF	---
매거진 데이터 OEM	REDEF	REDEF	REDEF	REDEF	---
전역 사용자 변수 (GUD)	DEF/REDEF	DEF	DEF	DEF/REDEF	DEF/REDEF
로컬 사용자 변수 (PUD/LUD)	DEF	DEF	DEF	---	---

다음 또한 참고:

변수 (페이지 429)

4.1.1.12 배열 변수 정의 및 초기화 (DEF, SET, REP)

사용자 변수는 1차원 배열부터 최대 3차원 배열로 정의할 수 있습니다.

- 1차원: `DEF <data type> <variable name>[<n>]`
- 2차원: `DEF <data type> <variable name>[<n>,<m>]`
- 3차원: `DEF <data type> <variable name>[<n>,<m>,<o>]`

참고

STRING 데이터 유형의 사용자 변수는 최대 2차원 배열로 정의할 수 있습니다.

데이터 유형

다음과 같은 데이터 유형의 사용자 변수를 배열로 정의할 수 있습니다. BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

배열 요소에 값 지정

다음과 같은 시점에 배열 요소에 값을 지정할 수 있습니다.

- 배열 정의 중 (초기화 값)
- 프로그램 실행 중

값은 다음과 같은 방법으로 지정할 수 있습니다.

- 배열 요소를 명시적으로 지정
- 배열 요소를 시작 요소로 명시적으로 지정하고 값 목록 (`SET`) 지정
- 배열 요소를 시작 요소로 명시적으로 지정하고 값과 해당 값의 반복 주기를 지정 (`REP`)

참고

FRAME 데이터 유형의 사용자 변수는 초기화 값으로 지정할 수 없습니다.

구문 (DEF)

```
DEF <data type> <variable name>[<n>,<m>,<o>]
DEF  STRING[<string length>] <variable name>[<n>,<m>]
```

구문 (DEF...=SET...)

값 목록 사용:

- 정의 중:
  ```
  DEF <data type> <variable
  name>[<n>,<m>,<o>]=SET(<value1>,<value2>, ...)
  ```
  해당값:
  ```
  DEF <data type> <variable
  name>[<n>,<m>,<o>]=(<value1>,<value2>, ...)
  ```

  참고

  값 목록을 통해 초기화하는 경우 SET을 지정할 필요가 없습니다.

- 값 지정 중:
  ```
  <variable name>[<n>,<m>,<o>]=SET(<VALUE1>,<value2>, ...)
  ```

구문 (DEF...=REP...)

값을 반복 사용

- 정의 중:
  ```
  DEF <data type> <variable name>[<n>,<m>,<o>]=REP(<value>)
  DEF <data type> <variable name>[<n>,<m>,<o>]=REP(<value>,
  <number_array_elements>)
  ```
- 값 지정 중:
  ```
  <variable name>[<n>,<m>,<o>]=REP(<value>)
  DEF <data type> <variable
  name>[<n>,<m>,<o>]=REP(<value>,<number_array_elements>)
  ```

의미

DEF:	변수를 정의하기 위한 명령
<data type>:	변수의 데이터 유형
	값 범위: • 시스템 변수의 경우: BOOL, CHAR, INT, REAL, STRING, AXIS • GUD 또는 LUD 변수의 경우: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME
<string length>:	STRING 데이터 유형의 문자에 대한 최대값

<variable name>:	변수 이름.	
[<n>,<m>,<o>]:	배열 크기 또는 배열 인덱스	
<n>:	1차원의 배열 크기 또는 배열 인덱스	
	유형:	INT (시스템 변수 및 AXIS의 경우)
	값 범위:	최대 배열 크기: 65535 배열 인덱스: 0 ≤ n ≤ 65534
<m>:	2차원의 배열 크기 또는 배열 인덱스	
	유형:	INT (시스템 변수 및 AXIS의 경우)
	값 범위:	최대 배열 크기: 65535 배열 인덱스: 0 ≤ m ≤ 65534
<o>:	3차원의 배열 크기 또는 배열 인덱스	
	유형:	INT (시스템 변수 및 AXIS의 경우)
	값 범위:	최대 배열 크기: 65535 배열 인덱스: 0 ≤ o ≤ 65534
SET:	지정된 값 목록을 이용해 값 지정	
(<value1>,<value2>, ...):	값 목록	
REP:	지정된 <value>를 이용해 값 지정	
<value>:	값, REP로 초기화하는 경우 배열 요소 값이 작성됩니다.	
<number_array_elements>:	지정된 <value>이 쓰여져야 할 배열 요소의 수 나머지 배열 요소에는 시점에 따라 다음이 적용됩니다. • 배열 정의 시 초기화: → 나머지 배열 요소에 0이 쓰여집니다. • 프로그램 실행 중 지정: → 배열 요소의 실제값이 그대로 유지됩니다. 이 파라미터를 프로그래밍하지 않을 경우 모든 배열 요소에 <value>가 쓰여집니다. 파라미터가 0인 경우 시점에 따라 다음이 적용됩니다. • 배열 정의 시 초기화: → 모든 요소가 미리 지정된 0입니다. • 프로그램 실행 중 지정: → 배열 요소의 실제값이 그대로 유지됩니다.	

배열 인덱스

배열 요소의 암시적인 순서는 배열 인덱스의 반복 때문에 왼쪽부터 오른쪽 순서로 지정됩니다 (예: SET 또는 REP를 사용해 값을 지정하는 경우).

예: 3차원 배열을 24개 배열 요소로 초기화:

```
DEF INT FELD[2,3,4] = REP(1,24)
  FELD[0,0,0] = 1    1번째 배열 요소
  FELD[0,0,1] = 1    2번째 배열 요소
  FELD[0,0,2] = 1    3번째 배열 요소
  FELD[0,0,3] = 1    4번째 배열 요소
  ...
  FELD[0,1,0] = 1    5번째 배열 요소
  FELD[0,1,1] = 1    6번째 배열 요소
  ...
  FELD[0,2,3] = 1    12번째 배열 요소
  FELD[1,0,0] = 1    13번째 배열 요소
  FELD[1,0,1] = 1    14번째 배열 요소
  ...
  FELD[1,2,3] = 1    24번째 배열 요소
```

해당값:

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

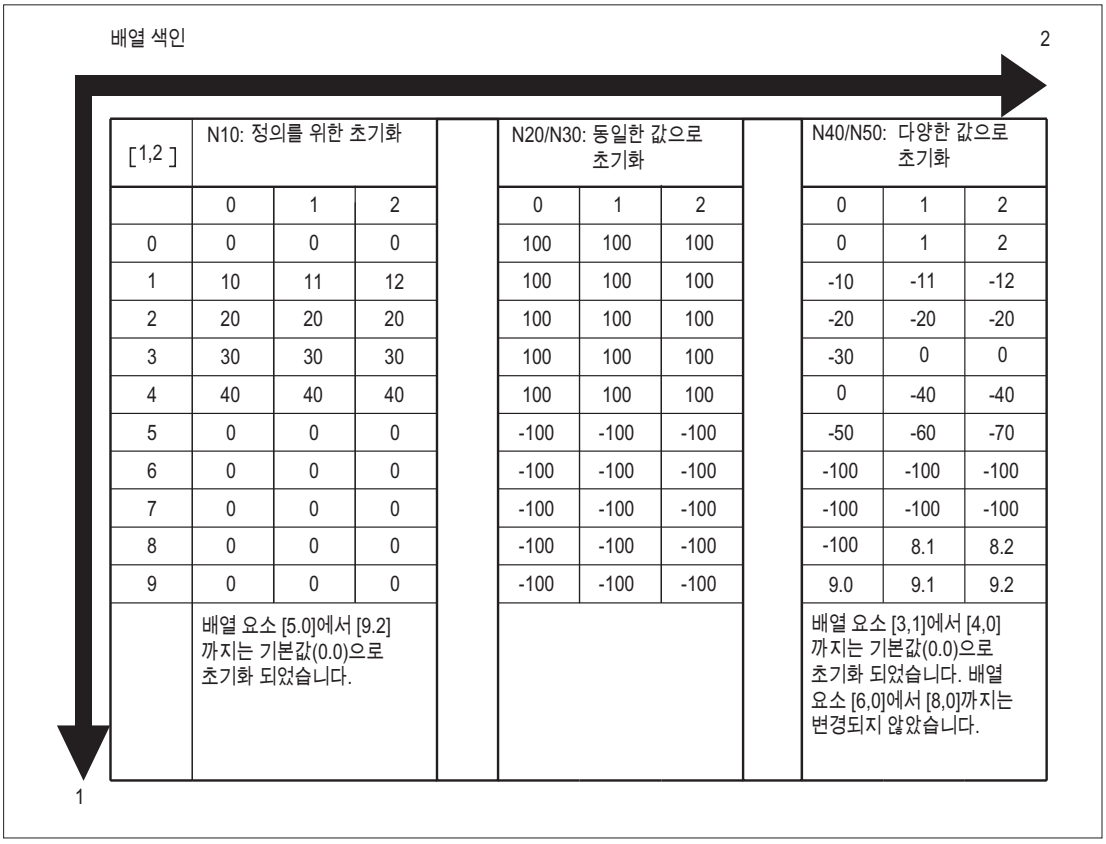
예: 전체 변수 배열 초기화

실제 지정의 경우 다이어그램을 참조하십시오.

프로그램 코드
N10 DEF REAL FELD1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)

프로그램 코드

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



[1,2]	N10: 정의를 위한 초기화		
	0	1	2
0	0	0	0
1	10	11	12
2	20	20	20
3	30	30	30
4	40	40	40
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
	배열 요소 [5.0]에서 [9.2] 까지는 기본값(0.0)으로 초기화 되었습니다.		

N20/N30: 동일한 값으로 초기화		
0	1	2
100	100	100
100	100	100
100	100	100
100	100	100
100	100	100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100
-100	-100	-100

N40/N50: 다양한 값으로 초기화		
0	1	2
0	1	2
-10	-11	-12
-20	-20	-20
-30	0	0
0	-40	-40
-50	-60	-70
-100	-100	-100
-100	-100	-100
-100	8.1	8.2
9.0	9.1	9.2
배열 요소 [3,1]에서 [4,0] 까지는 기본값(0.0)으로 초기화 되었습니다. 배열 요소 [6,0]에서 [8,0]까지는 변경되지 않았습니다.		

다음 또한 참고:

배열 변수 정의 및 초기화 (DEF, SET, REP) (페이지 464)

변수 (페이지 429)

추가 정보(SET)

정의 초기화

- 첫 번째 배열 요소부터 시작해 값 목록에 프로그래밍된 배열 요소의 수만큼 값 목록에 있는 값을 지정합니다.
- 값 목록에 명시적으로 선언된 값이 없는 경우 배열 요소에 값 0이 지정됩니다 (값 목록에서의 갭).

- AXIS 데이터 유형의 변수인 경우 값 목록에서의 갭이 허용되지 없습니다.
- 값 목록에 정의된 배열 요소보다 더 많은 값이 포함된 경우 알람이 표시됩니다.

프로그램 실행 시 값 지정

프로그램 실행 시 값을 지정하는 경우 앞서 설명한 정의 관련 규칙이 적용됩니다. 또한 다음 옵션이 지원됩니다.

- 표현식도 값 목록에서 요소로 허용됩니다.
- 값 지정은 프로그래밍된 배열 인덱스부터 시작합니다. 값을 하위 배열에 선택적으로 지정할 수 있습니다.

예:

프로그램 코드	설명
`DEF INT ARRAY[5,5]`	`; 배열 정의`
`ARRAY[0,0]=SET(1,2,3,4,5)`	`; 처음 5개 배열 요소 [0,0]~[0,4]에 값 지정`
`ARRAY[0,0]=SET(1,2, , ,5)`	`; 처음 5개 배열 요소 [0,0]~[0,4]에 갭을 포함한 값을 지정, 배열 요소 [0,2] 및 [0,3] = 0`

`ARRAY[2,3]=SET(VARIABLE,4*5.6)`	`; 배열 인덱스 [2,3]부터 변수 및 표현식을 포함한 값 지정:`
	`[2,3] = VARIABLE`
	`[2,4] = 4 * 5.6 = 22.4`

추가 정보 (REP)

정의 초기화

- 모든 배열 요소 또는 선택적으로 지정한 배열 요소는 지정된 값 (상수) 으로 초기화합니다.
- FRAME 데이터 유형의 변수는 초기화할 수 없습니다.

예:

프로그램 코드	설명
`DEF REAL varName[10]=REP(3.5,4)`	`; 배열 정의 및 배열 요소 [0]~[3]을 값 3.5로 초기화.`

프로그램 실행 시 값 지정

프로그램 실행 시 값을 지정하는 경우 앞서 설명한 정의 관련 규칙이 적용됩니다. 또한 다음 옵션이 지원됩니다.

- 표현식도 값 목록에서 요소로 허용됩니다.
- 값 지정은 프로그래밍된 배열 인덱스부터 시작합니다. 값을 하위 배열에 선택적으로 지정할 수 있습니다.

예:

프로그램 코드	설명
`DEF REAL varName[10]`	`; 배열 정의`
`varName[5]=REP(4.5,3)`	`; 배열 요소 [5]~[7] = 4.5`
`R10=REP(2.4,3)`	`; R-파라미터 R10~R12 = 2.4`
`DEF FRAME FRM[10]`	`; 배열 정의`
`FRM[5]=REP(CTRANS(X,5))`	`; 배열 요소 [5]~[9] = CTRANS(X,5)`

다음 또한 참고:

배열 변수 정의 및 초기화 (DEF, SET, REP) (페이지 464)

4.1.1.13 데이터 유형

NC에는 다음과 같은 데이터 유형을 사용할 수 있습니다.

데이터 유형	의미	값 범위
INT	부호가 있는 정수	-2147483648 ... +2147483647
REAL	실수 (LONG REAL ~ IEEE)	±(~2,2*10^{-308} ... ~1,8*10^{+308})
BOOL	참 값 TRUE (1) / FALSE (0)	1, 0
CHAR	ASCII 문자	ASCII 코드 0 ~ 255
STRING	정의된 길이의 문자열	최대 200 문자기호 (특수 문자기호가 아님)
AXIS	축 이름/스핀들 이름	채널 축 이름
FRAME	스태틱 좌표 변환을 위한 기하 파라미터 (변환, 회전, 스케일링, 미러링)	---

암시적 데이터 유형 변환

다음과 같은 데이터 유형 변환이 가능하며 변환은 지정 및 파라미터 전송 중 암시적으로 실행됩니다.

↓에서 →로 변환	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&

↓에서 →로 변환	REAL	INT	BOOL
BOOL	x	x	x
x: 제한 없이 사용 가능 o: 값 범위를 초과한 데이터 유실 가능 ⇒ 알람; 반올림: 소수 자리 값 ≥ 0.5 ⇒ 반올림, 소수 자리 값 < 0.5 ⇒ 버림 &: 값 ≠ 0 ⇒ TRUE, 값 == 0 ⇒ FALSE			

다음 또한 참고:

변수 (페이지 429)

4.1.1.14 변수 최소, 최대 및 범위 (MINVAL, MAXVAL 및 BOUND)

MINVAL 및 MAXVAL 명령은 2개 변수의 값을 비교합니다. MINVAL의 경우 더 작은 값이, MAXVAL의 경우 더 큰 값이 결과로 출력됩니다.

BOUND 명령은 테스트 변수의 값이 정의된 값 범위 내에 있는지 테스트합니다.

구문

```
<smaller value>=MINVAL(<variable1>,<variable2>)
<larger value>=MAXVAL(<variable1>,<variable2>)
<return value>=<BOUND>(<minimum>,<maximum>,<test variable>)
```

의미

MINVAL:	2개의 변수 (<variable1>, <variable2>) 중 **더 작은 값**을 가져옵니다.
<smaller value>:	MINVAL 명령의 결과 변수 더 작은 변수 값으로 설정합니다.
MAXVAL:	2개의 변수 (<variable1>, <variable2>) 중 **더 큰 값**을 가져옵니다.
<larger value>:	MAXVAL 명령의 결과 변수 더 큰 변수 값으로 설정합니다.
BOUND:	변수 (<test variable>) 가 정의된 값 범위 내에 있는지 테스트합니다.
<minimum>:	값 범위의 최소값을 정의하는 변수.

<maximum>:	값 범위의 최대값을 정의하는 변수.
<return value>:	BOUND 명령의 결과 변수 테스트 변수의 값이 정의된 값 범위 내에 있는 경우 테스트 변수의 값이 결과 변수로 설정됩니다. 테스트 변수의 값이 정의된 값 범위보다 클 경우 정의된 값 범위의 최대값이 결과 변수로 설정됩니다. 테스트 변수의 값이 정의된 값 범위보다 작을 경우 정의된 값 범위의 최소값이 결과 변수로 설정됩니다.

참고

MINVAL, MAXVAL 및 BOUND는 동기 동작에도 프로그래밍할 수 있습니다.

참고

값이 같은 경우, 동작

두 값이 같은 경우 MINVAL/MAXVAL이 모두 이 값으로 설정됩니다. BOUND의 경우 테스트할 변수의 값이 다시 반환됩니다.

예

프로그램 코드	코멘트
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar	
rValMin=MINVAL(rVar1,rVar2)	; rValMin을 값 10.5로 설정합니다.
rValMax=MAXVAL(rVar1,rVar2)	; rValMax을 값 33.7로 설정합니다.
rVar3=19.7	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3이 한계 내에 있습니다. rRetVar이 19.7로 설정됩니다.
rVar3=1.8	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3이 최소값보다 작습니다. rRetVar이 10.5로 설정됩니다.
rVar3=45.2	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3이 최대값보다 큽니다. rRetVar이 33.7로 설정됩니다.

4.1.1.15 변수 가용성 검사 (ISVAR)

사전 정의된 ISVAR 기능을 사용하여 시스템/사용자 변수(예: 머신 데이터, 설정 데이터, 시스템 변수, GUD와 같은 일반 변수)를 NC에서 인식하는지 여부를 확인할 수 있습니다.

변수

조회할 변수는 다음 구조를 가집니다.

무차원 변수:	<Variable>
배열 인덱스가 없는 1차원 변수:	<Variable>[ ]
배열 인덱스 n이 있는 1차원 변수:	<Variable>[<n>]
배열 인덱스가 없는 2차원 변수:	<Variable>[,]
배열 인덱스 n과 m이 있는 2차원 변수:	<Variable>[<n>,<m>]

구문

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

의미

<result>:	반환값		
	데이터 유형:	BOOL	
	값 범위:	1	사용 가능한 변수
		0	알 수 없는 변수
ISVAR:	시스템/사용자 변수를 NC에서 인식하는지 여부 검사.		
<variable>:	시스템/사용자 변수의 이름		
	데이터 유형:	STRING	
<n>:	1차원 배열 인덱스(**옵션**)		
	데이터 유형:	INT	
<m>:	2차원 배열 인덱스(**옵션**)		
	데이터 유형:	INT	

전송 파라미터를 기준으로 다음 검사를 수행합니다.

- 이름이 인식됩니까?
- 변수가 배열입니까?
- 1차원 배열입니까, 아니면 2차원 배열입니까?
- 각 배열 인덱스가 허용 범위에 있습니까?

모든 검사 결과가 긍정인 경우에만 TRUE (1)가 반환됩니다.

검사 결과가 부정이거나 구문 오류가 발생하면 FALSE (0)가 반환됩니다.

예

프로그램 코드	코멘트
DEF INT VAR1	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VAR1")	; IS_VAR이 TRUE입니다.

프로그램 코드	코멘트
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VARARRAY[,]")	; IS_VAR이 TRUE이고 2차원 배열입니다.
N20 IS_VAR=ISVAR("VARARRAY")	; IS_VAR이 TRUE이고 변수가 존재합니다.
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	; IS_VAR이 FALSE이고 허용되지 않는 배열 인덱스입니다.
N40 IS_VAR=ISVAR("VARARRAY[8,8")	; IS_VAR이 FALSE이고 "]"이 누락되었습니다 (구문 에러).
N50 IS_VAR=ISVAR("VARARRAY[,8]")	; IS_VAR이 TRUE이고 허용되는 배열 인덱스입니다.
N60 IS_VAR=ISVAR("VARARRAY[8,]")	; IS_VAR이 TRUE이고 허용되는 배열 인덱스입니다.

프로그램 코드	코멘트
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"	; 전송 파라미터가 머신 데이터 항목이고 IS_VAR이 TRUE입니다.

프로그램 코드	코멘트
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("$P_EP")	; IS_VAR이 TRUE입니다.
N20 IS_VAR=ISVAR("$P_EP[X]")	; IS_VAR이 TRUE입니다.

4.1.1.16 속성 값/데이터 유형 읽기(GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM, GETVARDFT, GETVARTYP)

시스템/사용자 변수의 속성 값은 GETVARPHU, GETVARAP, GETVARLIM, GETVARDIM 및 GETVARDFT 기능, 시스템/사용자 변수의 데이터 유형은 GETVARTYP 기능을 이용하여 읽을 수 있습니다.

물리 단위 읽기

구문:
<Result>=GETVARPHU(<name>)

의미:

<result>:	물리 단위의 숫자 값		
	데이터 유형:	INT	
	값 범위:	"속성: 물리 단위 (PHU) (페이지 455)"의 표 참조	
		에러가 있는 경우	
		- 2	지정된 변수 이름이 시스템 파라미터 또는 사용자 변수에 지정되지 않았습니다.
GETVARPHU:	시스템/사용자 변수의 물리 단위 읽기		
<name>:	시스템/사용자 변수의 이름		
	데이터 유형:	STRING	

예:

NC에는 다음 GUD 변수가 포함되어 있습니다.

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

프로그램 코드	설명
DEF INT result=0	
result=GETVARPHU("electric")	; GUD 변수의 물리 단위를 결정합니다.
IF (result < 0) GOTOF error	

값 42가 결과로 반환됩니다. 이 값은 물리 단위 [kW]에 해당합니다.

참고

GETVARPHU를 사용하여 예를 들어 변수 지정 a = b에서 두 변수가 지정된 물리 단위를 사용하는지 여부를 검사할 수 있습니다.

읽기 사용 권한

구문:
<Result>=GETVARAP(<name>,<access>)

의미:

<result>:	지정된 <access>에 대한 사용 권한		
	데이터 유형:	INT	
	값 범위:	0 ... 7	"속성: 사용 권한 (APR, APW, APRP, APWP, APRB, APWB) (페이지 457)"을 참조하십시오.
		에러가 있는 경우	
		- 1	쓸 수 없습니다(액세스 유형 "WP" 및 "WB"만 해당)
		- 2	지정된 변수 이름이 시스템 파라미터 또는 사용자 변수에 지정되지 않았습니다.
		- 3	파라미터 <access> 값이 잘못되었습니다.
GETVARAP:	시스템/사용자 변수의 사용 권한 읽기		
<name>:	시스템/사용자 변수의 이름		
	데이터 유형:	STRING	
<access>:	액세스 유형		
	데이터 유형:	STRING	
	값 범위:	"RP"	가공 프로그램을 통해 읽기
		"WP"	가공 프로그램을 통해 쓰기
		"RB"	OPI를 통해 읽기
		"WB"	OPI를 통해 쓰기

예:

프로그램 코드	설명
DEF INT result=0	
result=GETVARAP("$TC_MAP8","WB")	; OPI를 통해 쓰기를 기준으로 시스템 파라미터 "magazine position"에 대한 사용 권한을 지정합니다.
IF (result < 0) GOTOF error	

값 7이 결과로 반환됩니다. 이 값은 키 스위치 위치 0 (= 액세스 보호 안함) 에 해당합니다.

참고

GETVARAP를 사용하여 예를 들어 애플리케이션의 사용 권한을 검사하는 검사 프로그램을 구현할 수 있습니다.

한계값 읽기

구문:

```
<Status>=GETVARLIM(<name>,<limit value>,<result>)
```

의미:

<Status>:	기능 상태		
	데이터 유형:	INT	
	값 범위:	1	OK
		-1	정의된 한계값이 없습니다. (AXIS, STRING 및 FRAME 유형 변수의 경우)
		-2	지정된 변수 이름이 시스템 파라미터 또는 사용자 변수에 지정되지 않았습니다.
		-3	파라미터 <limit value> 값이 잘못되었습니다.
GETVARLIM:	시스템/사용자 변수의 하한값/상한값 읽기		
<name>:	시스템/사용자 변수의 이름		
	데이터 유형:	STRING	
<limit value>:	어떤 한계값을 읽어올지 지정합니다.		
	데이터 유형:	CHAR	
	값 범위:	"L"	하한값
		"U":	상한값
<result>:	한계값 반환		
	데이터 유형:	VAR REAL	

예:

프로그램 코드	설명
DEF INT state=0	
DEF REAL result=0	
state=GETVARLIM("$MA_MAX_AX_VELO","L",result)	; MD32000 $MA_MAX_AX_VELO의 하한값을 결정합니다.
IF (result < 0) GOTOF error	

치수 값 읽기

구문:
```
<Result>=GETVARDIM(<Name>,<Index>)
```

의미:

<result>:	파라미터 <Index>에 의해 지정된 치수 크기		
	데이터 유형:	INT	
GETVARDIM:	시스템/사용자 변수 필드의 첫 번째, 두 번째 또는 세 번째 치수의 크기 읽기		
<name>:	시스템/사용자 변수의 이름		
	데이터 유형:	STRING	
<index>:	배열 인덱스		
	데이터 유형:	INT	
	값 범위:	1 ... 3	
		1	필드의 첫 번째 치수의 인덱스
		2	필드의 두 번째 치수의 인덱스
		3	필드의 세 번째 치수의 인덱스

예:

프로그램 코드	설명
N5 DEF REAL myReal[5,4]	
N10 R1=GETVARDIM("myReal",1)	; 필드의 첫 번째 치수의 크기를 결정합니다.
	; 결과: R1 = 5
N15 R2=GETVARDIM("myReal",2)	; 필드의 두 번째 치수의 크기를 결정합니다.
	; 결과: R2 = 4

디폴트 값 읽기

구문:
```
<Status>=GETVARDFT(<Name>,<Result>[,<Index_1>,<Index_2>,
<Index_3>])
```

의미:

<Status>:	기능 상태		
	데이터 유형:	INT	
	값 범위:	1	OK
		-1	사용 가능한 기본값이 없습니다 (예: 결과 변수의 데이터 유형 정의가 시스템/사용자 변수의 데이터 유형과 일치하지 않기 때문).
		-2	지정된 변수 이름이 시스템 파라미터 또는 사용자 변수에 지정되지 않았습니다.
		-3	파라미터 <Index_1> 값이 잘못되었습니다. 치수가 1보다 작습니다(⇒ 스칼라 변수 이외에 필드 없음).
		-4	파라미터 <Index_2> 값이 잘못되었습니다.
		-5	파라미터 <Index_3> 값이 잘못되었습니다.
GETVARDFT:	시스템/사용자 변수의 디폴트 값 읽기		
<name>:	시스템/사용자 변수의 이름		
	데이터 유형:	STRING	
<result>:	디폴트 값 반환		
	데이터 유형:	VAR REAL (INT, REAL, BOOL 및 AXIS 유형 변수의 디폴트 값을 읽는 경우)	
		VAR STRING (STRING 및 CHAR 유형 변수의 디폴트 값을 읽는 경우)	
		VAR FRAME (FRAME 유형 변수의 디폴트 값을 읽는 경우)	
<index_1>:	필드의 첫 번째 치수의 인덱스(옵션)		
	데이터 유형:	INT	
	프로그래밍되지 않음 = 0		
<index_2>:	필드의 두 번째 치수의 인덱스(옵션)		
	데이터 유형:	INT	
	프로그래밍되지 않음 = 0		
<index_3>:	필드의 세 번째 치수의 인덱스(옵션)		
	데이터 유형:	INT	
	프로그래밍되지 않음 = 0		

예:

프로그램 코드	설명
DEF INT state=0	
DEF REAL resultR=0	; INT, REAL, BOOL 및 AXIS 유형의 디폴트 값을 수용하는 변수
DEF FRAME resultF=0	; FRAME 유형의 디폴트 값을 수용하는 변수
IF (GETVARTYP("$MA_MAX_AX_VELO") <> 4) GOTOF 에러	
state=GETVARDFT("$MA_MAX_AX_VELO",resultR, AXTOINT(X))	; "X" 축의 디폴트 값을 결정합니다.
IF (result < 0) GOTOF error	
IF (GETVARTYP("$TC_TP8") <> 3) GOTOF error	
state=GETVARDFT("$TC_TP8", resultR)	
IF (GETVARTYP("$P_UBFR") <> 7) GOTOF error	
state=GETVARDFT("$P_UBFR", resultF)	

데이터 유형 읽기

구문:
<Result>=GETVARTYP(<name>)

의미:

<result>:	지정된 시스템/사용자 변수의 데이터 유형		
	데이터 유형:	INT	
	값 범위:	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		에러가 있는 경우	
		< 0	지정된 변수 이름이 시스템 파라미터 또는 사용자 변수에 지정되지 않았습니다.
GETVARTYP:	시스템/사용자 변수의 데이터 유형 읽기		

<name>:	시스템/사용자 변수의 이름	
	데이터 유형:	STRING

예:

프로그램 코드	설명
DEF INT result=0	
DEF STRING name="R"	
result=GETVARTYP(name)	; R 파라미터의 유형을 확인합니다.
IF (result < 0) GOTOF error	

값 4가 결과로 반환됩니다. 이 값은 REAL 데이터 유형에 해당합니다.

4.1.1.17 가능한 유형 변환

일정한 숫자 값, 변수 또는 변수에 할당된 표현식은 변수 유형과 비교 가능해야 합니다. 따라서 값이 할당되면 유형이 자동으로 변환됩니다.

가능한 유형 변환

to	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
변경 전							
REAL	예	예*	예[1]	예*	–	–	–
INT	예	예	예[1]	예[2]	–	–	–
BOOL	예	예	예	예	예	–	–
CHAR	예	예	예[1]	예	예	–	–
STRING	–	–	예[4]	예[3]	예	–	–
AXIS	–	–	–	–	–	예	–
FRAME	–	–	–	–	–	–	예

설명

* REAL에서 INT 유형으로 변환 시 >=0.5의 분수 값은 반올림되고 그 이외 값은 반내림됩니다(ROUND 함수 참조).

[1] 값 <> 0은 TRUE에 해당하고 값 == 0은 FALSE에 해당

[2] 값이 허용 가능한 범위 내에 있는 경우

[3] 1 문자인 경우

[4] 문자열 길이 0 = >FALSE, 그렇지 않은 경우 TRUE

참고

변환으로 인해 대상 범위보다 큰 값이 산출된 경우 오류 메시지가 표시됩니다.

표현식에 혼합된 유형이 있는 경우 유형 변환이 자동으로 이루어집니다. 또한 동기식 작동에서도 유형 변환이 가능합니다. "이동-동기식 작동, 함축된 유형 변환" 장을 참조하십시오.

4.1.2 간접 프로그래밍

4.1.2.1 어드레스 간접 프로그래밍

간접적으로 어드레스 (페이지 1450)를 프로그래밍하는 경우 확장된 어드레스(<Index>)가 적절한 유형의 변수로 교체됩니다.

참고

다음의 경우 어드레스 간접 프로그래밍을 사용할 수 없습니다.

- N (블록 번호)
- L (서브프로그램)
- 설정 가능한 어드레스
  (예를 들어 X1 대신 X[1]은 허용되지 않습니다.)

구문

```
<ADDRESS>[<Index>]
```

의미

<ADDRESS>[...]:	확장자 (인덱스) 를 가진 고정 어드레스
<index>:	변수 (예: 스핀들 번호, 축, ...)

예

예 1: 스핀들 번호 간접 프로그래밍

직접 프로그래밍:

프로그램 코드	설명
S1=300	; 스핀들 번호 1의 속도 300(rpm)

간접 프로그래밍:

프로그램 코드	설명
DEF INT SPINU=1	; 변수, INT 유형 및 값 지정 정의.
S[SPINU]=300	; SPINU 변수에 번호가 저장된 스핀들의 속도 300 rpm (예 1의 경우 번호가 1인 스핀들).

예 2: 축 간접 프로그래밍

직접 프로그래밍:

프로그램 코드	설명
FA[U]=300	; 축 "U"의 이송 속도 300

간접 프로그래밍:

프로그램 코드	설명
DEF AXIS AXVAR2=U	; 변수, AXIS 유형 및 값 지정 정의.
FA[AXVAR2]=300	; AXVAR2 변수에 어드레스 이름이 저장된 축의 이송 속도 300.

예 3: 축 간접 프로그래밍

직접 프로그래밍:

프로그램 코드	설명
$AA_MM[X]	; "X" 축의 프로브 측정 값 (MCS) 을 읽습니다.

간접 프로그래밍:

프로그램 코드	설명
DEF AXIS AXVAR3=X	; 변수, AXIS 유형 및 값 지정 정의.
$AA_MM[AXVAR3]	; 변수 AXVAR3에 이름이 저장된 프로브 측정 값 (MCS) 을 읽습니다.

예 4: 축 간접 프로그래밍

직접 프로그래밍:

프로그램 코드
X1=100 X2=200

간접 프로그래밍:

프로그램 코드	설명
DEF AXIS AXVAR1 AXVAR2	; AXIS 유형의 변수 2개 정의.
AXVAR1=(X1) AXVAR2=(X2)	; 축 이름 지정.
AX[AXVAR1]=100 AX[AXVAR2]=200	; AXVAR1 및 AXVAR2 변수에 어드레스 이름이 저장된 축 이송

예 5: 축 간접 프로그래밍

직접 프로그래밍:

프로그램 코드
G2 X100 I20

간접 프로그래밍:

프로그램 코드	설명
DEF AXIS AXVAR1=X	; 변수, AXIS 유형 및 값 지정 정의.
G2 X100 IP[AXVAR1]=20	; AXVAR1 변수에 어드레스 이름이 저장된 축의 중심점 데이터 간접 프로그래밍.

예 6: 배열 요소 간접 프로그래밍

직접 프로그래밍:

프로그램 코드	설명
DEF INT ARRAY1[4,5]	; 배열 1 정의

간접 프로그래밍:

프로그램 코드	설명
DEFINE DIM1 AS 4	; 배열 치수의 경우 배열 크기를 고정값으로 지정해야 합니다.
DEFINE DIM2 AS 5	
DEF INT ARRAY[DIM1,DIM2]	
ARRAY[DIM1-1,DIM2-1]=5	

예 7: 간접 서브프로그램 호출

프로그램 코드	설명
CALL "L" << R10	; R10에 지정된 번호를 가진 프로그램 호출 (문자열 캐스케이딩).

4.1.2.2 G 코드 간접 프로그래밍

G 명령 (페이지 1466)의 간접 프로그래밍을 통해 싸이클을 효과적으로 프로그래밍할 수 있습니다.

구문

```
G[<group>]=<number>
```

의미

G[...]:	확장자 (인덱스) 를 가진 G 명령	
<group>:	인덱스 파라미터: G 그룹	
	유형:	INT
<number>:	G 코드 번호에 대한 변수	
	유형:	INT 또는 REAL

참고

일반적으로 구문을 결정하지 않는 G 코드만 간접적으로 프로그래밍할 수 있습니다.

구문을 결정하는 G 코드 중 가운데 G 코드 그룹 1만 프로그래밍이 가능합니다.
G 코드 그룹 2, 3 및 4에서 구문을 결정하는 G 코드는 프로그래밍이 불가능합니다.

참고

G 코드 간접 프로그래밍에서 산술 함수는 허용되지 않습니다. G 코드 값을 계산해야 하는 경우 G 코드 간접 프로그래밍을 수행하기 전에 개별 가공 프로그램 라인에서 계산해야 합니다.

예

예 1: 조정 가능한 워크 옵셋 (G 코드 그룹 8)

프로그램 코드	설명
N1010 DEF INT INT_VAR	
N1020 INT_VAR=2	
...	
N1090 G[8]=INT_VAR G1 X0 Y0	;G54
N1100 INT_VAR=INT_VAR+1	; G 코드 계산
N1110 G[8]=INT_VAR G1 X0 Y0	;G55

예 2: 레벨 선택(G 그룹 6)

프로그램 코드	설명
N2010 R10=$P_GG[6]	; G 코드 그룹 6의 활성 G 코드 읽기
...	
N2090 G[6]=R10	

4.1.2.3 위치 속성 간접 프로그래밍 (GP)

축 위치의 증분치 또는 절대치 프로그래밍과 같은 위치 속성을 키워드 GP를 사용하여 변수로 간접 프로그래밍할 수 있습니다.

적용

위치 속성 간접 프로그래밍이 **교체 싸이클**에서 사용되며, 이 경우처럼 위치 속성을 키워드(예: IC, AC) 로 프로그래밍하면 다음과 같은 이점이 있습니다.

변수로 간접 프로그래밍할 경우 CASE 명령이 필요하지 **않습니다**. 그렇지 않으면 가능한 모든 위치 속성에 대해 분기됩니다.

구문

```
<POSITIONING COMMAND>[<축/스핀들>]=
GP(<위치>,<위치 속성)
<axis/spindle>=BP(<position>,<position attribute)
```

의미

<POSITIONING COMMAND>[]:	다음 포지셔닝 명령을 키워드 GP와 함께 프로그래밍할 수 있습니다. POS, POSA, SPOS, SPOSA 또한 다음도 가능합니다. • 해당 채널에 전체 축 및 스핀들 이름이 있습니다. <axis/spindle> • 변수 축/스핀들 식별자 AX
<axis/spindle>:	포지셔닝할 축/스핀들
GP():	포지셔닝을 위한 키워드
<position>:	파라미터 1 상수 또는 변수인 축/스핀들 위치
<position attribute>:	파라미터 2 변수 (예: $P_SUB_SPOSMODE) 또는 키워드(IC, AC, ...)인 위치 속성(예: 위치 접근 모드)

변수에서 제공된 값은 다음과 같은 의미를 갖습니다.

값	의미	허용 가능한 값:
0	위치 속성에 대해 변경 사항 없음	
1	AC	POS, POSA,SPOS, SPOSA,AX, 축 어드레스
2	IC	POS, POSA,SPOS, SPOSA,AX, 축 어드레스
3	DC	POS, POSA,SPOS, SPOSA,AX, 축 어드레스
4	ACP	POS, POSA,SPOS, SPOSA,AX, 축 어드레스

값	의미	허용 가능한 값:
5	ACN	POS, POSA,SPOS, SPOSA,AX, 축 어드레스
6	OC	-
7	PC	-
8	DAC	POS, POSA,AX, 축 어드레스
9	DIC	POS, POSA,AX, 축 어드레스
10	RAC	POS, POSA,AX, 축 어드레스
11	RIC	POS, POSA,AX, 축 어드레스
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

예

1차 스핀들 S1 및 2차 스핀들 S2 사이 활성 동기 스핀들 커플링의 경우, 스핀들 위치를 지정하기 위한 다음 교체 싸이클이 메인 프로그램의 SPOS 명령을 사용하여 호출됩니다.

포지셔닝은 N2230의 구문을 사용해 실행됩니다.

```
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
```

복귀할 위치는 시스템 변수 $P_SUB_SPOSIT로부터 판독되고 위치 복귀 모드는 시스템 변수 $P_SUB_SPOSMODE로부터 판독됩니다.

프로그램 코드	설명
N1000 PROC LANG_SUB DISPLOF SBLOF	
...	
N2100 IF($P_SUB_AXFCT==2)	
N2110	; 활성 동기 스핀들 커플링에서 SPOS/SPOSA/M19 명령 교체
N2185 DELAYFSTON	; 정지 지연 영역의 시작
N2190 COUPOF(S2,S1)	; 동기 스핀들 커플링 비활성화
N2200	; 리딩 스핀들 및 추종 스핀들 포지셔닝
N2210 IF($P_SUB_SPOS==TRUE) OR ($P_SUB_SPOSA==TRUE)	
N2220	; SPOS를 사용하여 스핀들 포지셔닝:
N2230 SPOS[1]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)	

프로그램 코드	설명
SPOS[2]=GP($P_SUB_SPOSIT, $P_SUB_SPOSMODE)	
N2250 ELSE	
N2260	; M19를 사용하여 스핀들 포지셔닝:
N2270 M1=19 M2=19	; 리딩 스핀들 및 추종 스핀들 포지셔닝
N2280 ENDIF	
N2285 DELAYFSTOF	; 정지 지연 영역의 끝
N2290 COUPON(S2,S1)	; 동기 스핀들 커플링 활성화
N2410 ELSE	
N2420	; 추가 교체에 대한 쿼리
...	
N3300 ENDIF	
...	
N9999 RET	

보충 조건

위치 속성 간접 프로그래밍은 동기 동작에서 수행할 수 없습니다.

4.1.2.4 가공 프로그램 라인 간접 프로그래밍(EXECSTRING)

가공 프로그램 명령어 EXECSTRING을 사용하면 기존에 생성한 문자열 변수를 가공 프로그램 라인으로 실행할 수 있습니다.

구문

EXECSTRING은 개별 가공 프로그램 라인에서 프로그래밍됩니다.
EXECSTRING (<string_variable>)

의미

EXECSTRING:	문자열 변수를 가공 프로그램 라인으로 실행시키는 명령어
<string variable>:	실행할 실제 가공 프로그램 라인이 포함된 STRING 변수를 입력하십시오.

참고

EXECSTRING을 사용하면, 제어 구조 (페이지 531)를 제외하고 **가공 프로그램의 프로그램 섹션**에서 프로그램할 수 있는 모든 가공 프로그램 구문을 추출할 수 있습니다. 다시 말해 PROC 및 DEF 구문이 제외된다는 의미이며 INI 및 DEF 파일에서도 일반적으로 제외됩니다.

예

프로그램 코드	코멘트
N100 DEF STRING[100] MY_BLOCK	; 실행할 가공 프로그램 라인을 적용할 문자열 변수 정의.
N110 DEF STRING[10] MFCT1="M7"	
...	
N200 EXECSTRING(MFCT1 << "M4711")	; 가공 프로그램 라인 "M7 M4711" 실행.
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF(R10)	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(MY_BLOCK)	; 가공 프로그램 라인 "M3 M7" 실행.

4.1.3 명령

4.1.3.1 산술 함수

연산자/산술 함수	의미
+	덧셈
–	뺄셈
*	곱셈
/ [1]	나눗셈 [1]
DIV [1]	정수 나눗셈 [1]
MOD [1]	나머지 (정수 나눗셈의 나머지를 제공함) [1]
:	체인 연산자 (FRAME 변수)
SIN( )	사인
COS( )	코사인
TAN( )	접선
ASIN( )	아크 사인
ACOS( )	아크 코사인
ATAN2(,) [1]	아크 탄젠트2 [1]
SQRT( )	제곱근

ABS( )	절대치
POT( )	2승 (제곱)
TRUNC( )	정수 비교 명령의 정확도는 TRUNC를 사용하여 설정할 수 있습니다 ("비교 오류에 대한 정밀도 수정(TRUNC) (페이지 494)" 참조).
ROUND( )	정수로 반올림
LN( )	자연 로그
EXP( )	지수 함수
MINVAL ( )	두 변수 중 낮은 값 ("변수 최소, 최대 및 범위 (MINVAL, MAXVAL 및 BOUND) (페이지 471)" 참조)
MAXVAL ( )	두 변수 중 높은 값 ("변수 최소, 최대 및 범위 (MINVAL, MAXVAL 및 BOUND) (페이지 471)" 참조)
BOUND ( )	정의된 값 범위 내에서 변수 값 ("변수 최소, 최대 및 범위 (MINVAL, MAXVAL 및 BOUND) (페이지 471)" 참조)
CTRANS( )	옵셋
CROT ( )	회전
CSCALE( )	축척 변경
CMIRROR( )	미러링
1) "예" 단원 참조	

프로그래밍

산술 함수에는 일반적인 수학적 표기법이 사용됩니다. 실행 우선 순위는 괄호 안에 표시됩니다. 각도는 삼각 함수 및 역 함수로 지정됩니다 (직각 = 90°).

예

나눗셈: /

(유형 REAL) = 유형 INT 또는 유형 REAL) / (유형 INT 또는 유형 REAL);

예: 3 / 4 = 0.75

정수 나눗셈: DIV

(유형 INT) = (유형 INT 또는 REAL) / (유형 INT 또는 REAL);

예: 7 DIV 4.1 = 1

나머지 (정수 나눗셈의 나머지를 제공함): MOD

(유형 REAL) = (유형 INT 또는 REAL) MOD (유형 INT 또는 REAL);

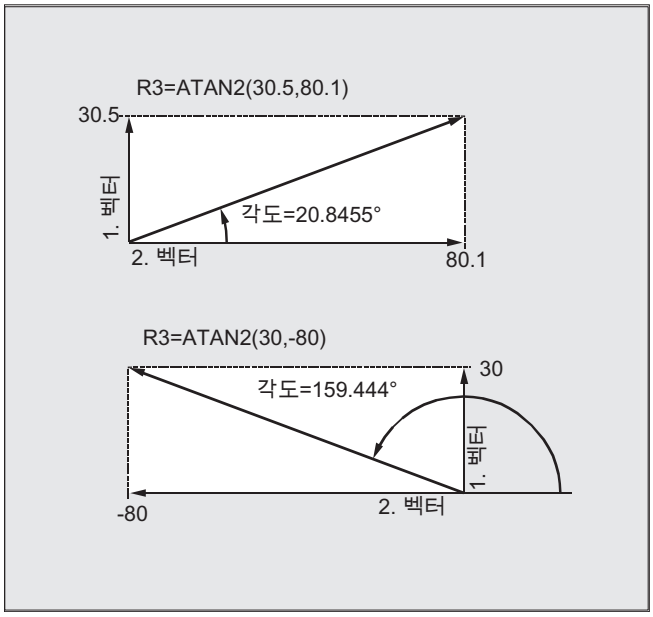
예: 7 MOD 4.1 = 2.9

아크 탄젠트 2: ATAN2

산술 함수 ATAN2는 서로 직각을 이루는 두 개의 벡터에서 총 벡터의 각도를 계산합니다.

결과는 네 개의 4분원 중 하나에 위치합니다 (-180° < 0 < +180°).

항상 각도 참조는 양의 방향으로 두 번째 값에 기반합니다.



프로그래밍 예제

프로그램 코드	코멘트
R1=R1+1	; 신규 R1 = 이전 R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; 곱셈 또는 나눗셈이 덧셈 또는 뺄셈보다 우선합니다.
R14=(R1+R2)*R3	; 식과 괄호를 먼저 계산합니다.
R15=SQRT(POT(R1)+POT(R2))	; 안쪽의 괄호를 먼저 해결합니다.
	R15 = ((R1^2 + R2^2))의 제곱근
RESFRAME=FRAME1:FRAME2	; 체인 연산자를 통한 FRAME 로직 연산
FRAME3=CTRANS(…):CROT(…)	FRAME 콤포넌트에 값 지정

4.1.3.2 비교 및 논리 연산자

비교 연산은 예를 들어 점프 조건을 공식으로 작성하는 데 사용할 수 있습니다. 복잡한 표현식도 비교할 수 있습니다.

비교 연산은 CHAR, INT, REAL 및 BOOL 유형의 변수에 적용 가능합니다. 코드 값을 CHAR 유형과 비교합니다.
STRING, AXIS 및 FRAME에는 == 및 <>가 허용됩니다. 동기 동작을 포함하여 STRING 유형 연산에서 사용할 수 있습니다.

비교 연산의 결과는 항상 BOOL 유형입니다.

논리 연산은 참/거짓 값을 연결하는 데 사용됩니다.

논리 연산은 유형 BOOL 변수에만 적용할 수 있습니다. 그러나 또한 내부 유형 변환을 통해 CHAR, INT 및 REAL 데이터 유형에도 적용할 수 있습니다.

논리 (Bool) 연산의 경우 BOOL, CHAR, INT 및 REAL 데이터 유형에 다음 규칙이 적용됩니다.

- 0: FALSE
- 0 이외 값: TRUE

비트 단위 논리 연산자

또한 논리 연산을 단일 비트의 CHAR 및 INT 유형에 적용할 수 있습니다. 유형 변환은 자동으로 수행됩니다.

프로그래밍

관계 연산자	의미
==	같음
<>	같지 않음
>	보다 큼
<	보다 작음
>=	보다 크거나 같음
<=	보다 작거나 같음

논리 연산자	의미
AND	AND
또는	또는

NOT	부정
XOR	익스클루시브 OR

비트 단위 논리 연산자	의미
B_AND	비트 단위 AND
B_OR	비트 단위 OR
B_NOT	비트 단위 부정
B_XOR	비트 단위 익스클루시브 OR

참고

산술 표현식에서 모든 연산자의 실행 순서는 괄호로 지정하여 일반 우선 순위 규칙을 재지정할 수 있습니다.

참고

부울 피연산자와 연산자 사이에 공백이 있어야 합니다.

참고

연산자 B_NOT만 단일 피연산자를 의미합니다. 이 피연산자는 연산자 뒤에 위치합니다.

예제

예 1: 비교 연산자

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

또는

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

R10>=100 비교 결과가 먼저 R11에 버퍼링됩니다.

예 2: 논리 연산자

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

또는

```
IF NOT R10 GOTOB START
```

NOT만 단일 피연산자를 의미합니다.

예 3: 비트 단위 논리 연산자

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

4.1.3.3 연산 우선 순위

각 연산자에 우선 순위를 지정합니다. 표현식을 평가할 때 항상 최상위 우선 순위를 가진 연산자가 먼저 적용됩니다. 여러 연산자가 동일한 우선 순위를 가진 경우 왼쪽부터 오른쪽으로 평가가 이루어집니다.

산술 표현식에서 모든 연산자의 실행 순서는 괄호로 지정하여 일반 우선 순위 규칙을 재지정할 수 있습니다.

연산자 순서

최상위부터 최하위 우선 순위

1.	NOT, B_NOT	부정, 비트 단위 부정
2.	*, /, DIV, MOD	곱셈, 나눗셈
3.	+, –	덧셈, 뺄셈
4.	B_AND	비트 단위 AND
5.	B_XOR	비트 단위 익스클루시브 OR
6.	B_OR	비트 단위 OR
7.	AND	AND
8.	XOR	익스클루시브 OR
9.	또는	또는
10.	<<	문자열 연결, 결과 유형 STRING
11.	==, <>, >, <, >=, <=	비교 연산자

참고

프레임의 연결 연산자 ":"는 다른 연산자와 동일한 표현식에서 사용할 수 없습니다. 따라서 이 연산자의 경우 우선 순위가 필요 없습니다.

예: IF 문

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

4.1.3.4 비교 오류에 대한 정밀도 수정(TRUNC)

TRUNC 명령을 수행하면 정밀도 인수에 의해 곱해진 피연산자를 자릅니다.

비교 명령에 대해 설정 가능한 정밀도

REAL 유형의 프로그램 데이터는 내부적으로 64비트의 IEEE 형식으로 표시됩니다. 이 표시 형식을 통해 십진수 값이 부정확하게 표시될 수 있으며 이로 인해 이론적으로 계산된 값과 비교할 때 예기치 않은 결과가 나올 수 있습니다.

상대 등식

프로그램 흐름을 간섭하는 표시 형식으로 인해 발생하는 부정확성을 방지하기 위해 비교 명령은 절대 등식이 아닌 상대 등식을 사용하여 비교합니다.

구문

비교 에러에 대한 정밀도 수정

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

의미

TRUNC:	십진수 자리를 자릅니다.

다음을 고려한 10^{-12}의 상대 등식

- 등식: (==)
- 부등식: (<>)
- 보다 크거나 같음: (>=)
- 보다 작거나 같음: (<=)
- 보다 크거나/작음: 절대 품질을 가진 (><)
- 보다 큼: (>)
- 보다 작음
  : (<)

호환성

호환성의 이유로 (>) 및 (<) 의 상대 품질 확인은 머신 데이터 MD10280 $MN_PROG_FUNCTION_MASK Bit0 = 1을 설정하여 비활성화할 수 있습니다.

참고

유형 REAL 데이터와의 비교는 위에 설명한 이유로 인해 특정 부정확도가 발생됩니다. 적용할 수 없는 편차가 발생할 경우 피연산자를 정밀도 인자와 곱하고 TRUNC로 자르는 INTEGER 계산을 사용합니다.

동기 동작

비교 명령에 대해 설명한 응답을 동기 동작에도 적용합니다.

예

예 1: 정확도 계산

프로그램 코드	설명
N40 R1=61.01 R2=61.02 R3=0.01	;초기값 지정
N41 IF ABS(R2-R1) > R3 GOTOF ERROR	; 지금까지 점프가 실행되었을 수 있음
N42 M30	; 프로그램 종료
N43 ERROR: SETAL(66000)	
R1=61.01 R2=61.02 R3=0.01	;초기값 지정
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000)	; 정확도 수정
R13=TRUNC(R3*1000)	
IF ABS(R12-R11) > R13 GOTOF ERROR	; 점프가 더 이상 실행되지 않음
M30	; 프로그램 종료
ERROR: SETAL(66000)	

예 2: 두 피연산자의 몫 계산 및 평가

프로그램 코드	설명
R1=61.01 R2=61.02 R3=0.01	;초기값 지정
IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF ERROR	; 점프가 실행되지 않음
M30	; 프로그램 종료
ERROR: SETAL(66000)	

4.1.3.5 반올림 (ROUNDUP)

실수 (REAL) 유형의 입력 값 (소수점을 가진 실수) 은 "ROUNDUP" 기능을 사용하여 한 단계 높은 정수로 반올림할 수 있습니다.

구문

```
ROUNDUP(<value>)
```

의미

ROUNDUP:	입력 값을 반올림하기 위한 명령
<value>:	유형 입력 값

참고

정수 (INTEGER) 유형의 입력값은 변경되지 않은 상태로 리턴됩니다.

예

예 1: 다양한 입력 값 및 반올림 결과

예	반올림 결과
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

예 2: NC 프로그램에서 ROUNDUP

프로그램 코드

```
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])
...
```

4.1.4 문자열 연산

문자열 연산

"할당" 및 "비교" 이외에 다음과 같은 문자열 동작을 사용할 수 있습니다

- STRING으로 유형 변환 (AXSTRING) (페이지 498)
- STRING 유형 변환 (NUMBER, ISNUMBER, AXNAME) (페이지 499)
- 문자열 연결 (<<) (페이지 500)
- 소문자/대문자로 변환 (TOLOWER, TOUPPER) (페이지 502)
- 문자열 길이 결정 (STRLEN) (페이지 502)

- 문자열에서 문자/문자열 검색 (INDEX, RINDEX, MINDEX, MATCH) (페이지 503)
- 하위 문자열 선택 (SUBSTR) (페이지 504)
- 개별 문자 읽기/쓰기 (페이지 505)
- 문자열 포맷 (SPRINT) (페이지 507)

0 문자의 특별한 의미

내부적으로 0 문자는 문자열의 끝 식별자로 해석됩니다. 특정 문자를 0으로 바꿀 경우 문자열이 잘립니다.

예:

프로그램 코드	코멘트
`DEF STRING[20] STRG="axis . stationary"`	
`STRG[6]="X"`	
`MSG(STRG)`	`; 메시지 "axis X stationary"가 표시됩니다.`
`STRG[6]=0`	
`MSG(STRG)`	`; 메시지 "axis"가 표시됩니다.`

4.1.4.1 STRING으로 유형 변환 (AXSTRING)

"STRING으로 유형 변환" 기능을 사용하여 다른 유형의 변수를 메시지 (MSG) 명령의 콤포넌트로 사용할 수 있습니다.

연산자 <<를 사용하면 데이터 유형 INT, REAL, CHAR 및 BOOL에 대해 암시적으로 유형 변환이 수행됩니다 ("문자열 연결 (<<) (페이지 500)" 참조).

INT 값은 일반 읽기 가능한 형식으로 변환됩니다. REAL 값은 최대 10개의 십진수 자리로 변환됩니다.

Type AXIS 변수는 `AXSTRING` 명령을 사용하여 STRING으로 변환할 수 있습니다.

구문

```
<STRING_RES> = << <any_type>
<STRING_RES> = AXSTRING(<축 이름>)
```

의미

`<STRING_RES>`:	유형 변환 결과에 대한 변수	
	유형:	STRING

<모든_유형>:	변수 유형 INT, REAL, CHAR, STRING 및 BOOL	
AXSTRING:	AXSTRING 명령을 실행하면 지정된 축 식별자가 문자열로 제공됩니다.	
<축 이름>:	축 식별자의 변수	
	유형:	AXIS

참고

FRAME 변수는 변환할 수 없습니다.

4.1.4.2 STRING 유형 변환 (NUMBER, ISNUMBER, AXNAME)

NUMBER 명령을 사용하여 STRING에서 REAL로 변환할 수 있습니다. 변환 가능 여부는 ISNUMBER 명령을 사용하여 확인할 수 있습니다.

문자열은 AXNAME 명령을 사용하여 축 데이터 유형으로 변환합니다.

구문

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

의미

NUMBER:	NUMBER 명령을 실행하면 <string>으로 표현된 값을 REAL 값으로 반환합니다.	
<string>:	변환할 STRING 유형 변수	
<REAL_RES>:	NUMBER를 사용한 유형 변환 결과에 대한 변수	
	유형:	REAL
ISNUMBER:	ISNUMBER 명령은 <문자열>을 유효한 숫자로 변환할 수 있는지 여부를 검사합니다.	

<BOOL_RES>:	ISNUMBER를 사용한 조회 결과에 대한 변수		
	유형:	BOOL	
	값:	TRUE	ISNUMBER는 <string>이 언어 규칙을 준수하여 유효한 REAL 값을 표시하는 경우 TRUE 값을 반환합니다.
		FALSE	ISNUMBER는 동일한 <string>으로 NUMBER를 호출할 경우 FALSE 값을 반환하고 알람이 실행됩니다.
AXNAME:	AXNAME 명령을 실행하면 지정된 <string>이 축 식별자로 변환됩니다. **참고:** 구성된 축 이름을 <문자열>에 지정할 수 없는 경우 알람이 발생합니다.		
<AXIS_RES>:	AXNAME를 사용한 유형 변환 결과에 대한 변수		
	유형:	AXIS	

예

프로그램 코드	코멘트
DEF BOOL BOOL_RES	
DEF REAL REAL_RES	
DEF AXIS AXIS_RES	
REAL_RES == 1234.9876Ex-7	; BOOL_RES == TRUE
BOOL_RES=ISNUMBER("1234XYZ")	; BOOL_RES == FALSE
REAL_RES=NUMBER("1234.9876Ex-7")	; REAL_RES == 1234.9876Ex-7
AXIS_RES=AXNAME("X")	; AXIS_RES == X

4.1.4.3 문자열 연결 (<<)

"문자열 연결" 기능을 사용하여 개별 요소로부터 문자열을 구성할 수 있습니다.

연결은 연산자 "<<"를 사용하여 수행됩니다. 이 연산자에는 기본 유형 CHAR, BOOL, INT, REAL 및 STRING의 모든 조합에 대한 대상 유형으로 STRING이 포함되어 있습니다. 필요한 경우 변환이 기존 규칙에 따라 수행됩니다.

구문

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

의미

<모든_유형>:	변수, 유형 CHAR, BOOL, INT, REAL 또는 STRING
<< :	문자열 (유형 STRING) 을 구성하기 위한 체인 변수 (<any_type>) 에 대한 연산자입니다. 또한 이 연산자는 소위 "unary" 변수로 독립적으로 사용할 수 있습니다. 이 연산자는 STRING (FRAME 및 AXIS의 경우 해당 없음) 에 대한 명시적 유형 변환기에 사용할 수 있습니다. << <any_type>

예를 들어, 문자 목록 및 파라미터로부터 구성될 수 있는 메시지 또는 명령을 삽입할 수 있습니다 (예를 들어 블록 이름).

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

참고

문자열 연결의 중간 결과가 최대 문자열 길이를 초과하지 않아야 합니다.

참고

FRAME 및 AXIS 유형을 연산자 "<<"와 함께 사용할 수 없습니다.

예

예 1: 문자열 연결

프로그램 코드	코멘트
DEF INT IDX=2	
DEF REAL VALUE=9.654	
DEF STRING[20] STRG="INDEX:2"	
IF STRG=="Index:"<<IDX GOTOF NO_MSG	
MSG("Index:"<<IDX<<"/value:"<<VALUE)	; 표시: "Index:2/value:9.654"
NO_MSG:	

예 2: <<을 이용한 명시적 유형 변환

프로그램 코드	코멘트
DEF REAL VALUE=3.5	
<<VALUE	; 지정된 REAL 유형 변수가 STRING 유형으로 변환됩니다.

4.1.4.4 소문자/대문자로 변환 (TOLOWER, TOUPPER)

"소문자/대문자로 변환" 기능을 통해 문자열의 모든 문자를 표준 표기로 변환할 수 있습니다.

구문

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

의미

TOUPPER:	TOUPPER 명령은 문자열의 모든 문자를 **대문자**로 변환합니다.	
TOLOWER:	TOLOWER 명령은 문자열의 모든 문자를 **소문자**로 변환합니다.	
<string>:	변환할 문자열	
	유형:	STRING
<STRING_RES>:	변환 결과에 대한 변수	
	유형:	STRING

예

사용자가 운영자 인터페이스에서 입력할 수 있기 때문에 입력 내용에 표준 대문자 또는 소문자 구분 규칙이 적용될 수 있습니다.

프로그램 코드

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

4.1.4.5 문자열 길이 결정 (STRLEN)

STRLEN 명령은 문자열의 길이를 결정합니다.

구문

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

의미

STRLEN:	STRLEN 명령은 지정된 문자열 길이를 결정합니다. 0이 아닌 문자 수가 문자열 시작부터 계산하여 반환됩니다.	
<string>:	길이를 결정할 문자열	
	유형:	STRING
<INT_RES>:	결정 결과에 대한 변수	
	유형:	INT

예

단일 문자 액세스와 함께 이 함수를 사용하여 문자열 길이를 결정할 수 있습니다.

프로그램 코드

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

4.1.4.6 문자열에서 문자/문자열 검색 (INDEX, RINDEX, MINDEX, MATCH)

이 기능을 통해 문자열 내에서 단일 문자 또는 문자열을 검색합니다. 기능 결과에서 검색된 문자열에 문자/문자열이 배치되는 위치를 지정합니다.

구문

INT_RES=INDEX(STRING,CHAR) ; **결과 유형: INT**

INT_RES=RINDEX(STRING,CHAR) ; **결과 유형: INT**

INT_RES=MINDEX(STRING,STRING) ; **결과 유형: INT**

INT_RES=MINDEX(STRING,STRING) ; **결과 유형: INT**

기호론

검색 기능: 검색에 성공한 문자열의 위치 (첫 번째 파라미터) 를 제공합니다. 문자/문자열을 찾을 수 없는 경우 값 -1이 반환됩니다. 첫 번째 문자의 위치는 0입니다.

의미

INDEX:	첫 번째 파라미터에서, 두 번째 파라미터로 지정된 문자를 시작부터 검색합니다.
RINDEX:	첫 번째 파라미터에서, 두 번째 파라미터로 지정된 문자를 끝에서 검색합니다.
MINDEX:	첫 번째로 찾은 문자의 인덱스가 반환되는 문자 목록이 문자열로 전달되는 경우를 제외하고 INDEX 함수에 해당합니다.
MATCH:	문자열에서 문자열을 검색합니다.

이를 통해 예를 들어 비어 있거나 경로 구분자 ("/") 가 있는 위치처럼 특정 기준에 따라 문자열을 분할할 수 있습니다.

예

입력을 경로 및 블록 이름으로 분할

프로그램 코드	코멘트
DEF INT PFADIDX, PROGIDX	
DEF STRING[26] INPUT	
DEF INT LISTIDX	
INPUT = "/_N_MPF_DIR/_N_EXECUTE_MPF"	
LISTIDX = MINDEX (INPUT, "M,N,O,P") + 1	; "N"은 선택 목록의 처음부터 시작해 파라미터 INPUT의 첫 번째 문자이기 때문에 LISTIDX에 반환되는 값은 3입니다.
PFADIDX = INDEX (INPUT, "/") +1	; 따라서 다음 규칙이 적용됩니다. PFADIDX = 1
PROGIDX = RINDEX (INPUT, "/") +1	; 따라서 다음 규칙이 적용됩니다. PROGIDX = 12
	; 다음 단원에 설명된 SUBSTR 기능을 사용하여 변수 INPUT을 "경로" 및 "모듈" 콤포넌트로 분할할 수 있습니다.
VARIABLE = SUBSTR (INPUT, PFADIDX, PROGIDX-PFADIDX-1)	"_N_MPF_DIR"을 반환합니다.
VARIABLE = SUBSTR (INPUT, PROGIDX)	; "_N_EXECUTE_MPF"를 반환합니다.

4.1.4.7 하위 문자열 선택 (SUBSTR)

문자열 내의 특정 부분을 SUBSTRING 기능을 사용하여 읽을 수 있습니다.

구문

<STRING_RES>=SUBSTR(<문자열>,<인덱스>,<길이>)

<STRING_RES>=SUBSTR(<문자열>,<인덱스>)

의미

SUBSTR:	이 기능은 <string>에서 <index>부터 지정된 <length>까지의 하위 문자열을 반환합니다. <length> 파라미터가 지정되어 있지 않으면 <index>부터 문자열 끝 사이의 하위 문자열을 반환합니다.
<index>:	문자열 내에서 하위 문자열의 시작 위치. 이 시작 위치가 문자열 끝보다 뒤에 있으면 빈 문자열 (" ") 이 반환됩니다. 문자열의 첫 번째 문자: Index = 0 값 범위: 0 ... (문자열 길이 - 1)
<length>:	하위 문자열의 길이. 길이를 너무 길게 지정하면 문자열 끝까지만 하위 문자열이 반환됩니다. 값 범위: 1 ... (문자열 길이 - 1)

예

프로그램 코드	코멘트
DEF STRING[29] RES	
; 1	
; 01234567890123456789	
RES = SUBSTR("QUITTUNG: 10 to 99", 10, 2)	; RES == "10"
RES = SUBSTR("QUITTUNG: 10 to 99", 10)	; RES == "10 to 99"

4.1.4.8 개별 문자 읽기/쓰기

문자열 내의 개별 문자를 읽고 쓸 수 있습니다.

이 때 다음 보충 조건을 준수해야 합니다.

- 사용자 정의 변수만 가능하고 시스템 변수는 불가능합니다.
- 문자열의 개별 문자는 서브프로그램 호출 시 "값별 호출"로만 전송됩니다.

구문

```
<Character>=<string>[<index>]
<Character>=<string_array>[<array_index>,<index>]
<String>[<index>]=<character>
<String_array>[<array_index>,<index>]=<character>
```

의미

<string>:	모든 문자열
<character>:	CHAR 유형의 변수
<index>:	문자열 내에서 문자의 위치. 문자열의 첫 번째 문자: Index = 0 값 범위: 0 ... (문자열 길이 - 1)

예

예 1: 변수 메시지

프로그램 코드	코멘트
; 0123456789	
DEF STRING [50] MESSAGE = "Axis n has reached position"	
MESSAGE [6] = "X"	
MSG (MESSAGE)	; "축 X가 위치에 도달했습니다."

예 2: 시스템 변수 평가

프로그램 코드	코멘트
DEF STRING[50] STRG	; 시스템 변수 버퍼
...	
STRG = $P_MMCA	; 시스템 변수 로드
IF STRG[0] == "E" GOTO ...	; 시스템 변수 평가

예 3: "값별 호출" 및 "기준별 호출" 파라미터 전송

프로그램 코드	코멘트
; 0123456	
DEF STRING[50] STRG = "Axis X"	
DEF CHAR CHR	
...	
EXTERN UP_VAL(ACHSE)	; "값별 호출" 파라미터로 서브프로그램 정의
EXTERN UP_REF(VAR ACHSE)	; "기준별 호출" 파라미터로 서브프로그램 정의
...	
UP_VAL(STRG[6])	; "값별" 파라미터 전송
...	
CHR = STRG[6]	; 버퍼
UP_REF(CHR)	; "기준별" 파라미터 전송

4.1.4.9 문자열 포맷 (SPRINT)

사전 정의된 SPRINT 기능을 사용하여 문자열을 포맷하고 외부 디바이스에 출력할 수 있습니다 ("공정 데이터 공유 - 외부 장치/파일로 출력 (EXTOPEN, WRITE, EXTCLOSE) (페이지 1136)" 참조).

구문

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

의미

SPRINT:	STRING 유형의 값을 제공하는 사전 정의된 기능에 대한 식별자
"<Format_String>":	고정 요소 및 가변 요소를 포함한 문자열. 가변 요소는 포맷 제어 문자 % 및 그 뒤에 나오는 포맷 설명을 사용하여 정의됩니다.
< value_1>,< value_2>,…,< value_n>:	상수 또는 NC 변수 형식의 값. 이 값은 n번째 포맷 제어 문자 %의 위치에 삽입되며 <format_string>에 나오는 포맷 설명에 해당합니다.
"<result_string>":	포맷된 문자열 (최대 400바이트)

사용 가능한 포맷 설명

%B:	변환할 값이 다음과 같은 경우 "TRUE" 문자열로 변환:
	• 0이 아닙니다.
	• 빈 문자열이 아닙니다 (문자열 값의 경우).
	변환할 값이 다음과 같은 경우 "FALSE" 문자열로 변환:
	• 0입니다.
	• 빈 문자열입니다.
	예:
	`N10 DEF BOOL BOOL_VAR=1`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)`
	결과: 문자열 "CONTENT OF BOOL_VAR:TRUE"가 RESULT 문자열 변수에 쓰여집니다.

%C:	ASCII 문자로 변환
	예:
	`N10 DEF CHAR CHAR_VAR="X"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C",CHAR_VAR)`
	결과: 문자열 "CONTENT OF CHAR_VAR:X"가 RESULT 문자열 변수에 쓰여집니다.

%D:	정수 값 (INTEGER) 을 포함한 문자열로 변환
	예:
	`N10 DEF INT INT_VAR=123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D",INT_VAR)`
	결과: 문자열 "CONTENT OF INT_VAR:123"이 RESULT 문자열 변수에 쓰여집니다.

%<m>D:	정수 값 (INTEGER) 을 포함한 문자열로 변환 문자열의 최소 길이는 <m>글자입니다. 누락된 자리는 공백으로 채워지고 왼쪽으로 정렬됩니다.
	예:
	`N10 DEF INT INT_VAR=-123`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D",INT_VAR)`
	결과: 문자열 "CONTENT OF INT_VAR:xx-123"이 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)

%F:	소수 부분이 6자리인 소수를 포함한 문자열로 변환. 적절한 경우에는 소수 부분이 반올림되거나 0으로 채워집니다. **예:** N10 DEF REAL REAL_VAR=-1.2341234EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR) 결과: 문자열 "CONTENT OF REAL_VAR: -1234.123400"이 문자열 변수 RESULT에 쓰여집니다.
%<m>F:	소수 부분이 6자리이고 총 길이가 <m>자 이상인 소수를 포함한 문자열로 변환. 적절한 경우에는 소수 부분이 반올림되거나 0으로 채워집니다. 누락된 글자는 공백을 사용하여 총 길이 <m>까지 채워지며, 좌측으로 정렬됩니다. **예:** N10 DEF REAL REAL_VAR=-1.23412345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR) 결과: 문자열 "CONTENT OF REAL_VAR: xxx-1234.123457"이 문자열 변수 RESULT에 쓰여집니다 (여기서 "x"는 공백 플레이스홀더).
%.<n>F:	소수 부분이 <n>자리인 소수를 포함한 문자열로 변환. 적절한 경우에는 소수 부분이 반올림되거나 0으로 채워집니다. **예:** N10 DEF REAL REAL_VAR=-1.2345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR) 결과: 문자열 "CONTENT OF REAL_VAR:-1234.568"이 RESULT 문자열 변수에 쓰여집니다.
%<m>.<n>F:	소수 부분이 <n>자리이고 총 길이가 <m>자 이상인 소수를 포함한 문자열로 변환. 적절한 경우에는 소수 부분이 반올림되거나 0으로 채워집니다. 누락된 글자는 공백을 사용하여 총 길이 <m>까지 채워지며, 좌측으로 정렬됩니다. **예:** N10 DEF REAL REAL_VAR=-1.2341234567890EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR) 결과: 문자열 "CONTENT OF REAL_VAR:xx-1234.12"가 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)

%E:	지수로 표현되는 소수를 포함한 문자열로 변환. 가수는 정수 부분 1자리와 소수 부분 6자리로 정규화되어 저장됩니다. 적절한 경우에는 소수 부분이 반올림되거나 0으로 채워집니다. 지수는 키워드 "EX"로 시작됩니다. 그 다음에는 부호 ("+" 또는 "-") 및 2자리 수 또는 3자리 수가 옵니다.
	예:
	`N10 DEF REAL REAL_VAR=-1234.567890`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:-1.234568EX+03"이 RESULT 문자열 변수에 쓰여집니다.

%<m>E:	지수로 표현되고 총 길이가 <m>자 이상인 소수를 포함한 문자열로 변환. 누락된 글자는 공백으로 채워지고 왼쪽으로 정렬됩니다. 가수는 정수 부분 1자리와 소수 부분 6자리로 정규화되어 저장됩니다. 적절한 경우에는 소수 부분이 반올림되거나 0으로 채워집니다. 지수는 키워드 "EX"로 시작됩니다. 그 다음에는 부호 ("+" 또는 "-") 및 2자리 수 또는 3자리 수가 옵니다.
	예:
	`N10 DEF REAL REAL_VAR=-1234.5`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03"이 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)

%.<n>E:	지수로 표현되는 소수를 포함한 문자열로 변환. 가수는 정수 부분 1자리와 소수 부분 <n>자리로 정규화되어 저장됩니다. 적절한 경우에는 소수 부분이 반올림되거나 0으로 채워집니다. 지수는 키워드 "EX"로 시작됩니다. 그 다음에는 부호 ("+" 또는 "-") 및 2자리 수 또는 3자리 수가 옵니다.
	예:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:-1.23EX+03"이 RESULT 문자열 변수에 쓰여집니다.

%<m>.<n>E:	지수로 표현되고 총 길이가 <m>자 이상인 소수를 포함한 문자열로 변환. 누락된 글자는 공백으로 채워지고 왼쪽으로 정렬됩니다. 가수는 정수 부분 1자리와 소수 부분 <n>자리로 정규화되어 저장됩니다. 적절한 경우에는 소수 부분이 반올림되거나 0으로 채워집니다. 지수는 키워드 "EX"로 시작됩니다. 그 다음에는 부호 ("+" 또는 "-") 및 2자리 수 또는 3자리 수가 옵니다.
	예:
	`N10 DEF REAL REAL_VAR=-1234.5678`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:xx-1.23EX+03"이 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)

%G:	값 범위에 따라 소수 또는 지수로 표현되는 소수를 포함한 문자열로 변환. 표현할 절대값이 1.0EX-04보다 작거나 1.0EX+06보다 크거나 같으면 지수 표기법이 선택되고, 그렇지 않으면 소수 표기법이 선택됩니다. 최대 6개 유효 자릿수가 표시되거나, 필요한 경우에는 반올림됩니다.
	소수 표기의 예:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:0.000123457"이 RESULT 문자열 변수에 쓰여집니다.
	지수 표기의 예:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:1.23457EX+06"이 RESULT 문자열 변수에 쓰여집니다.

%<m>G:	값 범위에 따라 소수 또는 지수로 표현되는 소수를 포함한 문자열로 변환(예: %G). 문자열의 최소 길이는 <m>글자입니다. 누락된 글자는 공백으로 채워지고 왼쪽으로 정렬됩니다.
	소수 표기의 예:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:xxxx0.000123457"이 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)
	지수 표기의 예:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+06`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:xxx1.23457EX+06"이 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)

%.<n>G:	값 범위에 따라 소수 또는 지수로 표현되는 소수를 포함한 문자열로 변환. 최대 <n>개 유효 자릿수가 표시되거나, 필요한 경우에는 반올림됩니다. 표현할 절대값이 1.0EX-04보다 작거나 1.0EX(+<n>)보다 크거나 같으면 지수 표기법이 선택되고, 그렇지 않으면 소수 표기법이 선택됩니다.
	소수 표기의 예:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:0.000123"이 RESULT 문자열 변수에 쓰여집니다.
	지수 표기의 예:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+03`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:1.23EX+03"이 RESULT 문자열 변수에 쓰여집니다.

%<m>.<n>G:	값 범위에 따라 소수 또는 지수로 표현되는 소수를 포함한 문자열로 변환(예: %.<n>G). 문자열의 최소 길이는 <m>글자입니다. 누락된 글자는 공백으로 채워지고 왼쪽으로 정렬됩니다.
	소수 표기의 예:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX-04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:xxx0.0001235"가 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)
	지수 표기의 예:
	`N10 DEF REAL REAL_VAR=1.234567890123456EX+04`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)`
	결과: 문자열 "CONTENT OF REAL_VAR:xx1.235EX+06"이 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)

%.<n>P:	REAL 값을 <n>자리의 소수 부분을 고려한 INTEGER 값으로 변환. INTEGER 값은 32비트 이진수로 출력됩니다. 변환할 값을 32비트로 표현할 수 없는 경우에는 처리가 중단되고 알람이 표시됩니다.
	포맷 구문 %.<n>P를 사용해 생성된 바이트 시퀀스도 이진수 0을 포함할 수 있습니다. 따라서 이 방법으로 생성되는 전체 문자열은 더 이상 NC 데이터 유형 STRING의 규칙을 따르지 않습니다. 결과적으로 이러한 문자열은 변수 유형 STRING에 저장할 수도 없고 NC 언어의 문자열 명령을 사용하여 처리할 수도 없습니다. 유일한 용도는 파라미터를 WRITE 명령으로 전송하여 적절한 외부 장치에서 출력하는 것입니다 (다음 예 참조).
	<Format_String>이 %P 유형의 포맷 설명을 포함하게 되면 %.<n>P를 사용하여 생성된 이진수를 제외한 전체 문자열은 MD10750 $MN_SPRINT_FORMAT_P_CODE에 따라 ASCII 문자 코드인 ISO(DIN6024) 또는 EIA(RS244)의 형식에 맞게 출력됩니다. 변환할 수 없는 문자가 프로그래밍되면 처리가 중단되고 알람이 표시됩니다.
	예:
	`N10 DEF REAL REAL_VAR=123.45`
	`N20 DEF INT ERROR`
	`N30 DEF STRING[20] EXT_DEVICE="/ext/dev/1"`
	`...`
	`N100 EXTOPEN(ERROR,EXT_DEVICE)`
	`N110 IF ERROR <> 0`
	`... ; 에러 처리`
	`N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR)`
	`N210 IF ERROR <> 0`
	`… ; 에러 처리`
	결과: 문자열 "INTEGER BINARY CODED: 'H0001E23A'"가 출력 장치 /ext/dev/1로 전송됩니다. 16진수 값 0x0001E23A는 십진수 값 123450에 해당합니다.

%<m>.<n>P:	머신 데이터 MD10751 $MN_SPRINT_FORMAT_P_DECIMAL의 설정에 해당하는 REAL 값을 다음 숫자를 포함하는 문자열로 변환
	• <m> + <n> 자리의 정수 **또는**
	• 최대 <m>자리의 정수 부분과 <n>자리의 소수 부분으로 이루어진 소수.
	포맷 설명 %.<n>P의 경우와 마찬가지로 전체 문자열이 MD10750 $MN_SPRINT_FORMAT_P_CODE에 의해 정의된 문자 코드로 저장됩니다.
	MD10751 = 0인 경우의 변환:
	REAL 값은 <m> + <n> 자리의 정수를 포함한 문자열로 변환됩니다. 필요한 경우 소수 부분은 <n>자리로 반올림되거나 0으로 채워집니다. 누락된 정수 자리는 공백으로 채워집니다. 마이너스 부호가 첨부되며 왼쪽으로 정렬됩니다. 플러스 부호 대신 공백이 입력됩니다.
	예:
	`N10 DEF REAL REAL_VAR=-123.45`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)`
	결과: 문자열 "PUNCHED TAPE FORMAT:-xx123450"이 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)
	MD10751 = 1인 경우의 변환:
	REAL 값은 최대 <m>자리의 정수 부분과 <n>자리의 소수 부분을 갖는 소수를 포함한 문자열로 변환됩니다. 필요한 경우에는 정수 부분이 잘리고 소수 부분이 반올림되거나 0으로 채워집니다. <n>이 0과 같으면 소수점이 생략됩니다.
	예:
	`N10 DEF REAL REAL_VAR1=-123.45`
	`N20 DEF REAL REAL_VAR2=123.45`
	`N30 DEF STRING[80] RESULT`
	`N40 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P VAR2:%2.0P", REAL_VAR1,REAL_VAR2)`
	결과: 문자열 "PUNCHED TAPE FORMAT:-123.450 VAR2:23"이 RESULT 문자열 변수에 쓰여집니다.

%S:	문자열 삽입
	예:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)`
	결과: 문자열 "CONTENT OF STRING_VAR:ABCDEFG"가 RESULT 문자열 변수에 쓰여집니다.

%<m>S:	최소 <m>글자를 포함한 문자열 삽입. 누락된 자리는 공백으로 채워집니다.
	예:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)`
	결과: 문자열 "CONTENT OF STRING_VAR:xxxABCDEFG"가 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)

%.<n>S:	첫 번째 글자부터 시작하여 문자열의 <n>글자 삽입
	예:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR)`
	결과: 문자열 "CONTENT OF STRING_VAR:ABC"가 RESULT 문자열 변수에 쓰여집니다.

%<m>.<n>S:	첫 번째 글자부터 시작하여 문자열의 <n>글자 삽입 생성된 문자열의 총 길이는 <m>글자 이상입니다. 누락된 자리는 공백으로 채워집니다.
	예:
	`N10 DEF STRING[16] STRING_VAR="ABCDEFG"`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR)`
	결과: 문자열 "CONTENT OF STRING_VAR:xxxxxABCDE"가 RESULT 문자열 변수에 쓰여집니다. (예의 "x"는 공백을 나타냅니다.)

%X:	INTEGER 값을 16진수 표기법을 사용한 문자열로 변환
	예:
	`N10 DEF INT INT_VAR='HA5B8'`
	`N20 DEF STRING[80] RESULT`
	`N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR)`
	결과: 문자열 "INTEGER HEXADECIMAL:A5B8"이 RESULT 문자열 변수에 쓰여집니다.

참고

식별자와 키워드에서 대문자와 소문자를 구분하지 않는 NC 언어의 속성도 포맷 설명에 적용됩니다. 결과적으로 대문자와 소문자 중 어느 것을 사용하더라도 기능 차이 없이 프로그램할 수 있습니다.

조합 옵션

다음 테이블은 NC 데이터 유형과 조합할 수 있는 포맷 설명에 대한 정보를 제공합니다. 암시적 데이터 유형 변환 규칙이 적용됩니다 ("데이터 유형 (페이지 470)" 참조).

	NC 데이터 유형						
	BOOL	**CHAR**	**INT**	**REAL**	**STRING**	**AXIS**	**FRAME**
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

참고

테이블은 NC 데이터 유형 AXIS 및 FRAME을 SPRINT 기능에서 직접 사용할 수 없음을 나타냅니다. 그러나 다음은 가능합니다.

- AXIS 데이터 유형을 AXSTRING 기능을 사용하여 문자열로 변환한 후 SPRINT로 처리할 수 있습니다.
- 프레임 콤포넌트에 액세스할 때마다 FRAME 데이터 유형의 개별 값을 읽을 수 있습니다. 결과적으로 REAL 데이터 유형을 이용하여 SPRINT로 처리할 수 있습니다.

4.1.5 프로그램 점프 및 분기

4.1.5.1 프로그램 시작으로 점프 (GOTOS)

프로그램을 반복하기 위해 GOTOS 명령을 사용하여 메인 프로그램 또는 서브프로그램의 시작으로 점프할 수 있습니다.

머신 데이터를 사용하여 프로그램 시작으로 리턴 점프하도록 설정할 수 있습니다.

- 프로그램 런타임을 "0"으로 설정합니다.
- 공작물 계산은 값 "1" 단위로 증분됩니다.

구문

```
GOTOS
```

의미

GOTOS:	대상 위치가 프로그램 시작인 점프 구문.	
	실행은 NC/PLC 인터페이스 신호를 통해 제어됩니다. DB21에서 DBX384.0로 (제어 프로그램 분기)	
	값:	의미:
	0	프로그램 시작으로 리턴 점프하지 않습니다. GOTOS 이후 다음 가공 프로그램 블록에서 프로그램이 다시 실행됩니다.
	1	프로그램 시작으로 리턴 점프합니다. 가공 프로그램이 반복됩니다.

제한 사항

- GOTOS는 내부적으로 STOPRE(전처리 정지)를 실행합니다.
- 데이터 정의(로컬 사용자 변수(LUD))가 포함된 가공 프로그램의 경우 GOTOS를 통해 정의 섹션 이후 첫 번째 프로그램 블록으로 점프합니다. 즉, 데이터 정의는 다시 실행되지 않습니다. 따라서 정의된 변수가 GOTOS 블록에서 도달된 값을 유지하고 정의 섹션에서 프로그램된 기본값으로 리셋되지 않습니다.
- "외부 소스에서 실행" 기능을 통해 외부 프로그램 메모리에서 점프 명령으로 프로그램을 실행할 경우 점프 대상이 리로드 메모리 안에 있어야 합니다.
  이 조건은 프로그램 시작에 대한 점프 명령(GOTOS)과 관련하여 특히 문제가 되는데, 이는 프로그램이 일반적으로 너무 커서 리로드 메모리에 완전히 맞지 않기 때문입니다. 처음 리로드하면 프로그램 시작이 리로드 메모리에서 제거됩니다. 프로그램 시작에 대한 점프 명령을 실행하면 이 기능이 더 이상 점프 대상을 찾을 수 없습니다. 프로그램이 중단되고 알람 14000이 출력됩니다.

  참고

  프로그램된 점프 명령에 대한 제한 없이 외부 프로그램을 실행하려면 "외부 소스에서 실행" 기능 대신 "외부 저장 장치에서 실행(EES)" 옵션을 사용하는 것이 좋습니다.

- GOTOS 명령은 동기 동작 및 고정 싸이클에서 사용할 수 없습니다.

예

프로그램 코드	설명
N10 ...	; 프로그램 시작.
...	
N90 GOTOS	; 프로그램 시작으로 점프.
...	

4.1.5.2 점프 마커로 프로그램 점프 (GOTOB, GOTOF, GOTO, GOTOC)

점프 마커(라벨)는 명령어 GOTOF, GOTOB, GOTO 또는 GOTOC를 사용하여 동일한 프로그램 내에서 다른 위치로 점프할 수 있도록 프로그램에서 설정할 수 있습니다. 대상 마커 바로 다음 구문에서 프로그램 실행이 재개됩니다. 즉, 프로그램 내에서 분기를 실현할 수 있습니다.

점프 마커 이외에 메인 및 서브 블록 번호를 점프 대상으로 사용할 수 있습니다.

점프 구문 앞에 점프 조건(IF...)을 지정하면 점프 조건을 충족하는 경우에만 프로그램 점프가 실행됩니다.

구문

```
GOTOB <jump destination>
IF <jump condition> == TRUE GOTOB <jump destination>

GOTOF <jump destination>
IF <jump condition> == TRUE GOTOF <jump destination>

GOTO <jump destination>
IF <jump condition> == TRUE GOTO <jump destination>

GOTOC <jump destination>
IF <jump condition> == TRUE GOTOC <jump destination>
```

의미

GOTOB:	점프 대상을 프로그램 시작으로 지정하는 점프 문구
GOTOF:	점프 대상을 프로그램 종료로 지정하는 점프 문구
GOTO:	점프 대상을 검색하는 점프 구문. 검색은 먼저 프로그램 진행 방향으로 수행된 다음 프로그램 진행 반대 방향으로 수행됩니다.
GOTOC:	알람 14080 "점프 대상을 찾을 수 없습니다"가 발생되지 않는다는 점을 제외하고 GOTO와 동일한 효과. 따라서 점프 대상 검색이 실패할 경우 프로그램 실행이 중단되지 않지만 GOTOC 명령에 따라 프로그램 행을 계속 수행합니다.

<jump destination>:	점프 대상 파라미터 가능한 데이터는 다음과 같습니다.	
	<점프 마커 (라벨):	점프 대상은 사용자 정의 이름을 가진 프로그램에서 설정된 점프 마커 (라벨) 입니다: 점프 마커 (라벨)
	<block number>:	점프 대상이 메인 블록 또는 서브 블록 번호 (예: 200, N300)
	STRING 유형 변수:	변수 점프 대상. 변수는 점프 마커 또는 블록 번호를 의미합니다.
IF:	점프 조건을 작성하기 위한 키워드. 점프 조건은 모든 비교 및 논리 연산을 허용합니다 (결과: TRUE 또는 FALSE). 프로그램 점프는 이 연산 결과가 TRUE인 경우 실행됩니다.	

참고

점프 마커

점프 마커는 항상 블록 시작에 위치합니다. 프로그램 번호가 있는 경우 점프 마커는 블록 번호 바로 다음에 위치합니다.

점프 마커 이름을 지정할 때 다음 규칙이 적용됩니다.

- 문자 개수:
  - 최소 2
  - 최대 32
- 허용되는 문자는 다음과 같습니다.
  - 문자
  - 숫자
  - 밑줄
- 처음 2자는 글자 또는 밑줄이어야 합니다.
- 점프 마커 이름 뒤에 콜론 (":") 을 붙입니다.

제한 사항

- 점프 대상은 프로그램 **내에** 위치한 점프 마커 또는 블록 번호가 있는 블록만 가능합니다.
- 점프 조건이 없는 점프 구문은 별도 블록에 프로그래밍해야 합니다. 이 규칙은 점프 조건이 있는 점프 구문에는 적용되지 않습니다. 이 경우 여러 점프 구문을 하나의 블록에 구성할 수 있습니다.

- 점프 조건이 없는 점프 명령이 있는 프로그램의 경우 프로그램 M2/M30의 끝이 반드시 프로그램 끝에 있을 필요가 없습니다.
- "외부 소스에서 실행" 기능을 통해 외부 프로그램 메모리에서 점프 명령으로 프로그램을 실행할 경우 점프 대상이 리로드 메모리 안에 있어야 합니다. 그렇지 않으면 점프 대상을 찾을 수 없어 프로그램이 중단되고 알람 14000이 출력됩니다.

참고

프로그램된 점프 명령에 대한 제한 없이 외부 프로그램을 실행하려면 "외부 소스에서 실행" 기능 대신 "외부 저장 장치에서 실행(EES)" 옵션을 사용하는 것이 좋습니다.

예

예 1: 점프 마커 (라벨)로 점프

프로그램 코드	설명
N10 …	
N20 GOTOF Label_1	; 프로그램 종료로 점프
	; 점프 마커 "라벨 _1".
N30 …	
N40 Label_0: R1=R2+R3	; 점프 마커 "라벨 _0" 설정.
N50 …	
N60 Label_1:	; 점프 마커 "라벨 _1" 설정.
N70 …	
N80 GOTOB Label_0	; 프로그램 시작으로 점프
	; 점프 마커 "라벨 _0".
N90 …	

예 2: 블록 번호로 간접 점프

프로그램 코드	설명
IF <condition> == TRUE	
R10=100	; 점프 대상 할당
ELSE	
R10=110	; 점프 대상 할당
ENDIF	
; R10에 위치한 블록 번호를 가진 블록에 대한 프로그램 종료로 점프	
N10 GOTOF "N"<<R10	
...	
N90 ...	
N100 ...	; 점프 대상
N110 ...	
...	

예 3: 변수 점프 대상으로 점프

프로그램 코드	설명
DEF STRING[20] DESTINATION	
IF <condition> == TRUE	
DESTINATION = "Label1"	; 점프 대상 할당
ELSE	
DESTINATION = "Label2"	; 점프 대상 할당
ENDIF	
; 변수 점프 대상 "Content of DESTINATION"에 대한 프로그램 종료로 점프	
GOTOF DESTINATION	
라벨1: T="Drill1"	; 점프 대상 1
...	
라벨2: T="Drill2"	; 점프 대상 2
...	

예 4: 점프 조건에 따라 점프

프로그램 코드	설명
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; 초기값 할당
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; 점프 라벨LA1
N42 R1=R1+R3 R4=R4-1	
; IF 점프 조건 == TRUE	
; 그러고 나서 점프 라벨 LA1에 대한 프로그램 시작으로 점프	
N43 IF R4>0 GOTOB LA1	
N44 M30	; 프로그램 종료

4.1.5.3 프로그램 분기 (CASE ... OF ... DEFAULT ...)

CASE 기능은 사용해 변수 또는 산술 함수의 실제 값 (유형: INT) 을 검사하고 그 결과에 따라 프로그램의 다른 위치로 점프할 수 있습니다.

구문

```
CASE(<expression>) OF <constant_1> GOTOF <jump destination_1>
<constant_2> GOTOF <jump destination_2> ... DEFAULT GOTOF <jump
destination_n>
```

의미

CASE:	점프 명령
<expression>:	변수 또는 산술 함수
OF:	조건 부 프로그램 분기를 구성하기 위한 키워드

<constant_1>:	변수 또는 산술 함수에 지정된 첫 번째 상수 값	
	유형:	INT
<constant_2>:	변수 또는 산술 함수에 대해 지정된 두 번째 상수 값	
	유형:	INT
DEFAULT:	변수 또는 산술 함수가 지정된 상수 값을 적용하지 않는 경우 DEFAULT 구문을 사용하여 점프 대상을 결정할 수 있습니다. **참고:** DEFAULT 구문이 프로그래밍되지 않은 경우 CASE 구문 다음에 오는 블록이 점프 대상입니다.	
GOTOF:	점프 대상을 프로그램 종료로 지정하는 점프 문구 GOTOF 대신 다른 모든 GOTO 명령을 프로그래밍할 수 있습니다 (제목 "프로그램이 점프 마커로 점프" 참조).	
<jump destination_1>:	변수 또는 산술 함수 값이 첫 번째 지정된 상수에 해당하는 경우 이 점프 대상에 대해 분기가 적용됩니다. 점프 대상은 다음과 같이 지정할 수 있습니다.	
	<점프 마커 (라벨):	점프 대상은 사용자 정의 이름을 가진 프로그램에서 설정된 점프 마커 (라벨) 입니다: 점프 마커 (라벨)
	<block number>:	점프 대상이 메인 블록 또는 서브 블록 번호 (예: 200, N300)
	STRING 유형 변수:	변수 점프 대상. 변수는 점프 마커 또는 블록 번호를 의미합니다.
<jump destination_2>:	변수 또는 산술 함수 값이 두 번째 지정된 상수에 해당하는 경우 이 점프 대상에 대해 분기가 적용됩니다.	
<jump destination_n>:	변수 값으로 지정된 상수 값을 적용하지 않는 경우 이 점프 대상에 대해 분기가 적용됩니다.	

제한 사항

외부 소스에서 처리

"외부 소스에서 실행" 기능을 통해 외부 프로그램 메모리에서 CASE 함수로 프로그램을 실행할 경우 점프 대상이 리로드 메모리 안에 있어야 합니다. 그렇지 않으면 점프 대상을 찾을 수 없어 프로그램이 중단되고 알람 14000이 출력됩니다.

참고

프로그램된 점프 명령에 대한 제한 없이 외부 프로그램을 실행하려면 "외부 소스에서 실행" 기능 대신 "외부 저장 장치에서 실행(EES)" 옵션을 사용하는 것이 좋습니다.

예

프로그램 코드

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

N30의 CASE 구문은 다음과 같이 프로그램 분기 방법을 정의합니다.

1. 산술 함수의 값이 VAR1+VAR2-VAR3 = 7이면 점프 마커 정의 "Label_1" (→ N40) 을 가진 블록으로 점프합니다.
2. 산술 함수의 값이 VAR1+VAR2-VAR3 = 9이면 점프 마커 정의 "Label_2" (→ N50) 를 가진 블록으로 점프합니다.
3. 산술 함수의 값이 VAR1+VAR2-VAR3 = 7 또는 9가 아니면 점프 마커 정의 "Label_3" (→ N60) 을 가진 블록으로 점프합니다.

4.1.6 프로그램 섹션 반복 (REPEAT, REPEATB, ENDLABEL, P)

프로그램 섹션 반복을 통해 프로그램 내에서 기존 프로그램 섹션을 특정 순서대로 반복할 수 있습니다.

반복할 프로그램 라인 또는 프로그램 섹션을 점프 마커 (라벨) 로 표시합니다.

참고

점프 마커 (라벨)

점프 마커는 항상 블록 시작에 위치합니다. 프로그램 번호가 있는 경우 점프 마커는 블록 번호 바로 다음에 위치합니다.

점프 마커 이름을 지정할 때 다음 규칙이 적용됩니다.

- 문자 개수:
  - 최소 2
  - 최대 32
- 허용되는 문자는 다음과 같습니다.
  - 문자
  - 숫자
  - 밑줄
- 처음 2자는 글자 또는 밑줄이어야 합니다.
- 점프 마커 이름 뒤에 콜론 (":") 을 붙입니다.

구문

1. 개별 프로그램 라인 반복:

```
<jump marker>: ...
...
REPEATB <jump marker> P=<n>
...
```

2. 점프 마커와 REPEAT 구문 사이의 프로그램 섹션 반복:

```
<점프 마커>: ...
...
REPEAT <jump marker> P=<n>
...
```

3. 두 점프 마커 사이의 섹션 반복:

```
<start jump marker>: ...
...
<end jump marker>: ...
...
REPEAT <start jump marker> <end jump marker> P=<n>
...
```

참고

REPEAT 명령문과 2개의 점프 마커를 괄호로 함께 묶을 수 없습니다. <start jump marker>는 REPEAT 명령문 앞에 있지만 <end jump marker>가 REPEAT 명령문 앞에 없으면 <start jump marker>와 REPEAT 명령문 사이의 섹션이 반복됩니다.

4. 점프 마커와 ENDLABEL 사이의 섹션 반복:

```
<점프 마커>: ...
...
ENDLABEL: ...
...
REPEAT <점프 마커> P=<n>
...
```

참고

REPEAT 명령문을 <jump marker> 및 ENDLABEL과 함께 괄호 안에 묶을 수 없습니다. <jump marker>가 REPEAT 명령문 앞에 있고 ENDLABEL이 REPEAT 명령문 앞에 없으면 <jump marker>와 REPEAT 명령문 사이의 섹션이 반복됩니다.

의미

REPEATB:	프로그램 라인 반복을 위한 명령
REPEAT:	프로그램 섹션 반복을 위한 명령

<jump marker>:	<jump marker>는 다음을 표시합니다. • 반복할 프로그램 라인 (REPEATB의 경우) 또는 • 반복할 프로그램 섹션의 시작 부분 (REPEAT의 경우)
	<jump marker>로 표시된 프로그램 라인은 REPEAT/REPEATB 명령문의 앞 또는 뒤에 위치할 수 있습니다. 처음에 프로그램 시작 방향으로 검색이 시작됩니다. 이 방향에서 점프 마커를 찾을 수 없는 경우 프로그램 끝 방향으로 검색이 계속됩니다. **예외:** 점프 마커와 REPEAT 구문 사이의 프로그램 섹션을 반복해야 하는 경우 (구문 아래 2항 참조), <점프 마커>로 표시된 프로그램 라인이 반드시 REPEAT 구문 **앞에** 위치해야 합니다. 검색이 프로그램 시작 **방향으로만** 진행되기 때문입니다. <jump marker>로 표시된 라인에 추가적인 연산이 있는 경우 라인이 반복될 때마다 이 연산도 다시 실행됩니다.
ENDLABEL:	반복할 프로그램 섹션의 끝을 표시하는 키워드. ENDLABEL로 표시된 라인에 추가적인 연산이 있는 경우 라인이 반복될 때마다 이 연산도 다시 실행됩니다. ENDLABEL은 프로그램에서 1회 이상 사용할 수 있습니다.
P:	반복 횟수를 지정하는 어드레스
<n>:	프로그램 섹션 반복 횟수
	유형: INT
	반복 대상인 프로그램 섹션을 <n>회 반복합니다. 마지막 반복 후 프로그램은 REPEAT/REPEATB 라인 다음 라인부터 다시 시작됩니다. **참고:** P=<n>에 반복 횟수가 지정되지 않은 경우 프로그램 섹션은 단 1회만 반복됩니다.

예

예 1: 개별 프로그램 라인 반복

프로그램 코드	코멘트
N10 POSITION1: X10 Y20	
N20 POSITION2: CYCLE(0,,9,8)	;위치 싸이클
N30	
N40 REPEATB POSITION1 P=5	; BLOCK N10을 5회 실행.
N50 REPEATB POSITION2	; 블록 N20을 1회 실행.

프로그램 코드	코멘트
N60 ...	
N70 M30	

예 2: 점프 마커와 REPEAT 구문 사이의 프로그램 섹션 반복:

프로그램 코드	코멘트
N5 R10=15	
N10 Begin: R10=R10+1	;너비
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	; N10부터 N70까지의 섹션 4회 실행.
N90 Z10	
N100 M30	

예 3: 두 점프 마커 사이의 섹션 반복

프로그램 코드	코멘트
N5 R10=15	
N10 Begin: R10=R10+1	;너비
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	; N10부터 N70까지의 섹션 3회 실행.
N110 Z10	
N120 M30	

예 4: 점프 마커와 ENDLABEL 사이의 섹션 반복

프로그램 코드	코멘트
N10 G1 F300 Z-10	
N20 BEGIN1:	
N30 X10	
N40 Y10	
N50 BEGIN2:	
N60 X20	
N70 Y30	

프로그램 코드	코멘트
N80 ENDLABEL: Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3: X20	
N120 Y30	
N130 REPEAT BEGIN3 P=3	; N110부터 N120까지의 섹션 3회 실행.
N140 REPEAT BEGIN2 P=2	; N50부터 N80까지의 섹션 2회 실행.
N150 M100	
N160 REPEAT BEGIN1 P=2	; N20부터 N80까지의 섹션 2회 실행.
N170 Z10	
N180 X0 Y0	
N190 M30	

예 5: 밀링에서 테크놀로지에 따른 기계 드릴 위치

프로그램 코드	코멘트
N10 CENTER DRILL()	; 센터링 드릴 로드.
N20 POS_1:	;드릴링 위치 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL:	
N80 POS_2:	;드릴링 위치 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	
N130 DRILL()	; 드릴 및 드릴링 싸이클 변경.
N140 THREAD(6)	; 탭 M6 및 나사 가공 싸이클 로드.
N150 REPEAT POS_1	; POS_1부터 ENDLABEL까지 프로그램 섹션 1회 반복.
N160 DRILL()	; 드릴 및 드릴링 싸이클 변경.
N170 THREAD(8)	; 탭 M8 및 나사 가공 싸이클 로드.
N180 REPEAT POS_2	; POS_2부터 ENDLABEL까지 프로그램 섹션 1회 반복.
N190 M30	

추가 정보

중첩

프로그램 섹션 반복을 중첩할 수 있습니다. 각 호출에서 서브프로그램 레벨을 사용합니다.

M17 / RET

프로그램 섹션 반복을 처리하는 중에 M17 또는 RET이 프로그래밍되면 반복이 중단됩니다. 해당 프로그램은 REPEAT 라인 다음 블록에서 다시 시작됩니다.

프로그램 디스플레이

현재 프로그램 디스플레이에서 프로그램 섹션 반복은 개별 서브프로그램 레벨로 표시됩니다.

레벨 취소

프로그램 섹션 반복 동안 레벨이 취소되면 프로그램은 프로그램 섹션 반복 호출 이후 지점에서 재개됩니다.

예:

프로그램 코드	설명
N5 R10=15	
N10 BEGIN: R10=R10+1	;너비
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	; 인터럽트 레벨
N50 X=-R10	
N60 Y=-R10	
N70 END: Z10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	
N120 Z10	; 프로그램 실행 재개.
N130 M30	

점검 구조 및 프로그램 섹션 반복

점검 구조 및 프로그램 섹션 반복을 조합하여 사용할 수 있습니다. 그러나 서로 간에 겹치지 않아야 합니다. 프로그램 섹션 반복이 점검 구조 분기 내에 나타나거나 점검 구조가 프로그램 섹션 반복 내에 나타나야 합니다.

점프 및 프로그램 섹션 반복

점프 및 프로그램 섹션 반복이 혼합된 경우 블록은 정확하게 순서대로 실행됩니다. 예를 들어 점프가 프로그램 섹션 반복에서 수행되는 경우 프로그램 섹션의 프로그램된 끝이 검색될 때까지 프로세스가 계속됩니다.

예:

프로그램 코드
N10 G1 F300 Z-10
N20 BEGIN1:

프로그램 코드
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

참고

REPEAT 문은 이송 블록 다음에 위치해야 합니다.

외부 소스에서 실행 및 프로그램 섹션 반복

프로그램 섹션 반복이 있는 외부 프로그램의 경우 반복할 프로그램 라인(REPEATB) 또는 반복할 프로그램 섹션의 시작(REPEAT)이 리로드 메모리 안에 있어야 합니다. 그렇지 않으면 점프 대상을 찾을 수 없어 프로그램이 중단되고 알람 14000이 출력됩니다.

참고

프로그램 섹션 반복에 대한 제한 없이 외부 프로그램을 실행하려면 "외부 소스에서 실행" 기능 대신 "외부 저장 장치에서 실행(EES)" 기능을 사용하는 것이 좋습니다.

4.1.7 점검 구조

시스템은 프로그램 순서대로 NC 블록을 처리합니다.

이 순서는 대체 프로그램 블록 및 프로그램 루프 프로그래밍에 따라 다를 수 있습니다. 이 점검 구조는 키워드 IF, ELSE, ENDIF, LOOP, FOR, WHILE 및 REPEAT를 사용하여 프로그래밍합니다.

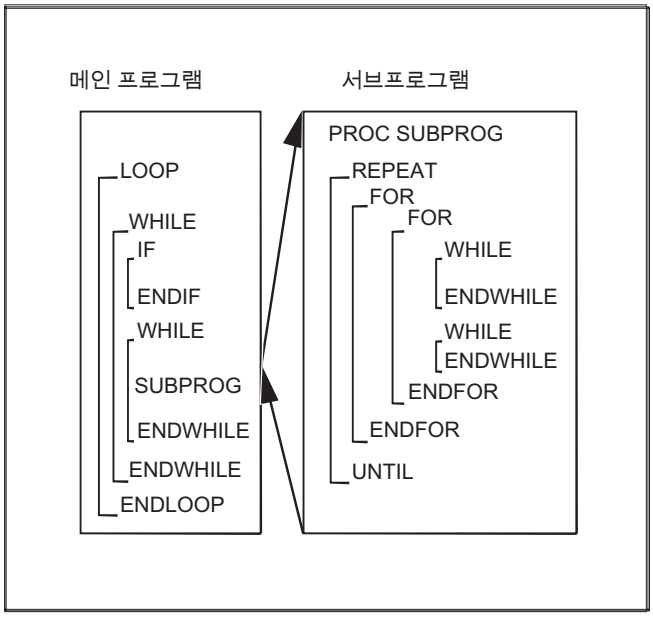
유의사항
프로그래밍 에러 점검 구조는 프로그램의 명령문 섹션에만 삽입될 수 있습니다. 프로그램 헤더 정의는 조건적 또는 반복적으로 실행될 수 없습니다. 점검 구조의 키워드 또는 점프 대상에 매크로를 중첩시킬 수 없습니다. 매크로를 정의할 경우 이러한 점검이 수행되지 않습니다.

유효성

점검 구조는 프로그램 전체를 대상으로 할 수 없습니다.

중첩 깊이

최대 16개의 점검 구조에 대한 중첩 깊이를 각 서브프로그램 레벨에서 설정할 수 있습니다.



런타임 응답

해석 모드 (표준으로 활성) 에서 점검 구조에서 가져올 수 있는 것보다 많은 프로그램 분기를 사용하여, 프로그램 처리 시간을 상당히 효율적으로 줄일 수 있습니다.

미리 컴파일된 싸이클 내에서 프로그램 분기 및 점검 구조 사이에 차이가 없습니다.

프로그램 루프 중 현재 블록 디스플레이

프로그램 루프에서 선택된 블록만 실행되는 경우 프로그램 루프 **이전의** 마지막 메인 실행 블록이 현재 블록 디스플레이에 표시됩니다.

선택된 블록을 처리한 후 진단 등의 목적으로 현재 블록 디스플레이에 표시하려면 디코딩 싱글 블록 SBL2를 활성화해야 합니다.

메인 실행 블록 없는 연삭

프로그램 루프 안에서 메인 실행 블록이 프로그램 되지 않았다면, 루프는 루프 조건이 만족될 때까지 전처리됩니다.

결과적으로 높은 수준의 활용이 일어나고 이는 디스플레이에 부정적 영향을 미칠 수 있습니다.

STOPRE 명령 또는 0초의 드웰 타임 G04가 대응 조치의 일환으로 루프 **내** 에 삽입될 수 있습니다.

제한 사항

블록 디스플레이

점검 구조 요소를 가진 블록을 억제할 수 없습니다.

점검 구조 및 프로그램 점프

점검 구조 요소 블록에서 점프 마커 (라벨) 는 허용되지 않습니다.

참고

일반적으로 점검 구조와 프로그램 분기를 혼합하여 사용하지 않는 것이 바람직합니다.

외부 소스에서 처리

점검 구조가 있는 외부 프로그램의 경우 루프의 시작이 리로드 메모리 안에 있어야 합니다. 그렇지 않으면 점프 대상을 찾을 수 없어 프로그램이 중단되고 알람 14000이 출력됩니다.

참고

점검 구조에 대한 제한 없이 외부 프로그램을 실행하려면 "외부 소스에서 실행" 기능 대신 "외부 저장 장치에서 실행(EES)" 기능을 사용하는 것이 좋습니다.

해석 모드

점검 구조는 분석적으로 처리됩니다. 루프 끝이 감지되면 루프 시작을 검색하여 프로세스에 있는 점검 구조를 찾을 수 있습니다. 이러한 이유로 프로그램의 블록 구조는 해석 모드에서 완전히 점검되지 않습니다.

싸이클의 사전 처리

점검을 통해 싸이클 사전 처리 시, 점검 구조가 중첩되었는지 확인할 수 있습니다.

4.1.7.1 조건문 및 분기 (IF, ELSE, ENDIF)

조건문: IF - program block - ENDIF

조건문을 사용하면 IF와 ENDIF 사이의 프로그램 블록은 지정된 조건을 충족하는 경우에만 실행됩니다.

분기: IF - program block_1 - ELSE - program block_2 - ENDIF

분기를 사용하면 항상 두 프로그램 블록 중 하나가 실행됩니다.

조건을 충족하면 IF와 ELSE 사이의 program block_1이 실행됩니다.

조건을 **충족하지 못하면** ELSE와 ENDIF 사이의 program block_2가 실행됩니다.

참고

동기 동작의 ELSE

키워드 ELSE는 동기 동작에서도 프로그래밍할 수 있습니다. 따라서 조건이 충족되지 않으면 실행될 동작으로 동기 동작을 확장할 수 있습니다.

구문

조건문

```
IF <condition>
   프로그램 블록                              ; 실행 조건: <Condition> == TRUE
ENDIF
```

분기

```
IF <condition>
   Program block_1                           ; 실행 조건: <Condition> == TRUE
ELSE
   Program block_2                           ; 실행 조건: <Condition> == FALSE
ENDIF
```

의미

IF:	조건문 또는 분기를 소개합니다.
ELSE:	대체 프로그램 블록을 소개합니다.
ENDIF:	조건문 또는 분기의 끝을 나타냅니다.
<condition>:	TRUE 또는 FALSE로 평가되는 논리 표현식.

예: 공구 교환 서브프로그램

프로그램 코드	설명
PROC L6	공구 교환 루틴
N500 DEF INT TNR_AKTUELL	활성 T 번호에 대한 변수
N510 DEF INT TNR_VORWAHL	사전 선택된 T 번호에 대한 변수
	현재 공구 확인
N520 STOPRE	
N530 IF $P_ISTEST	프로그램 테스트 모드에서...
N540 TNR_AKTUELL = $P_TOOLNO	... '현재' 공구가 프로그램 컨텍스트에서 판독됩니다.
N550 ELSE	그렇지 않으면...
N560 TNR_AKTUELL = $TC_MPP6[9998,1]	... 스핀들 공구를 판독합니다.
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	스핀들에서 미리 선택한 공구의 T 번호가 판독됩니다.
N590 IF TNR_AKTUELL <> TNR_VORWAHL	미리 선택한 공구가 계속 현재 공구가 아닌 경우...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	... 공구 교환 위치로 이동...
N610 M206	... 공구를 교체합니다.
N620 ENDIF	
N630 M17	

4.1.7.2 연속 프로그램 루프 (LOOP, ENDLOOP)

연속 루프는 연속 프로그램에서 사용됩니다. 루프의 끝에는 항상 시작으로 돌아가는 분기가 있습니다.

구문

```
LOOP
...
ENDLOOP
```

의미

LOOP:	연속 루프를 시작합니다.
ENDLOOP:	루프 끝을 표시하고 루프 시작으로 리턴 점프합니다.

예

프로그램 코드

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

4.1.7.3 카운트 루프 (FOR ... TO ..., ENDFOR)

카운트 루프는 고정된 실행 횟수만큼 작업을 반복해야 하는 경우 사용됩니다.

구문

```
FOR <variable> = <initial value> TO <end value>
...
ENDFOR
```

의미

FOR:	카운트 루프를 시작합니다.	
ENDFOR:	카운트 최종 값에 도달할 때까지 루프 끝을 표시하고 루프 시작으로 리턴 점프합니다.	
<variable>:	카운트 변수이며 처음부터 끝 값까지 증분되며 실행 시마다 "1" 값 단위로 증가합니다.	
	유형	INT 또는 REAL **참고:** REAL 유형은 예를 들어 카운트 루프에 R 파라미터가 프로그래밍된 경우 사용됩니다. 카운트 변수가 REAL 유형인 경우 해당 값은 정수로 반올림됩니다.

<initial value>:	카운트 초기 값 조건: 시작값은 반드시 최종값보다 작아야 합니다.
<full-scale value>:	카운트 최종 값

예

예 1: INTEGER 변수 또는 R 파라미터를 카운트 변수로 사용

카운터 변수인 INTEGER 변수:

프로그램 코드	코멘트
DEF INT iVARIABLE1	
R10=R12-R20*R1 R11=6	
FOR iVARIABLE1 = R10 TO R11	; 카운트 변수 = INTEGER 변수
R20=R21*R22+R33	
ENDFOR	
M30	

카운터 변수인 R 파라미터:

프로그램 코드	코멘트
R11=6	
FOR R10=R12-R20*R1 TO R11	; 카운터 변수 = R 파라미터 (Real 변수)
R20=R21*R22+R33	
ENDFOR	
M30	

예 2: 지정된 수량의 부품 생산

프로그램 코드	코멘트
DEF INT WKPCCOUNT	; "WKPCCOUNT" 이름을 가진 유형 INT 변수를 정의합니다.
FOR WKPCCOUNT = 0 TO 100	; 카운트 루프를 시작합니다. "WKPCCOUNT" 변수는 초기값 "0"부터 최종값 "100"까지 증분합니다.
G01 …	
ENDFOR	; 카운트 루프 종료
M30	

4.1.7.4 루프 시작 시 조건을 가진 프로그램 루프 (WHILE, ENDWHILE)

WHILE 루프의 경우 조건은 루프 시작에 위치합니다. 조건을 충족하는 한 WHILE 루프가 실행됩니다.

구문

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

의미

WHILE:	프로그램 루프를 시작합니다.
ENDWHILE:	루프 끝을 표시하고 루프 시작으로 리턴 점프합니다.
<condition>:	WHILE 루프가 실행되도록 조건을 충족해야 합니다.

예

프로그램 코드	코멘트
...	
WHILE $AA_IW[DRILL_AXIS] > -10	; 다음 조건에 따라 WHILE 루프를 호출합니다. 드릴링 축에 대해 실제 WCS 지령치는 -10 이상이어야 합니다.
G1 G91 F250 AX[DRILL_AXIS] = -1	
ENDWHILE	
...	

4.1.7.5 루프 끝에서 조건을 가진 프로그램 루프 (REPEAT, UNTIL)

REPEAT 루프의 경우 조건은 루프 끝에 위치합니다. REPEAT 루프는 한번만 실행되고 조건을 충족할 때까지 계속 반복합니다.

구문

```
REPEAT
...
UNTIL <significance>
```

의미

REPEAT:	프로그램 루프를 시작합니다.
UNTIL:	루프 끝을 표시하고 루프 시작으로 리턴 점프합니다.
<condition>:	REPEAT 루프가 더 이상 실행되지 않도록 충족해야 하는 조건입니다.

예

프로그램 코드	코멘트
...	
REPEAT	; REPEAT 루프를 호출합니다.
...	
UNTIL ...	; 조건을 충족하는지 검사합니다.
...	

4.1.7.6 중첩된 점검 구조를 가진 프로그램 예

프로그램 코드	코멘트
LOOP	
IF NOT $P_SEARCH	; IF 블록 탐색 안함
G1 G90 X0 Z10 F1000	
WHILE $AA_IM[X] <= 100	; WHILE (지령치 X 축 <= 100)
G1 G91 X10 F500	; 드릴링 패턴
Z-5 F100	
Z5	
ENDWHILE	
ELSE	; ELSE 블록 탐색
MSG("No drilling during block search")	
ENDIF	; ENDIF
$A_OUT[1] = 1	; 다음 드릴링 플레이트
G4 F2	
ENDLOOP	
M30	

4.1.8 채널 교차 프로그램 코디네이션(INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

원칙적으로 NC의 채널은 그 안에서 시작된 프로그램을 해당 모드 그룹의 다른 채널들과 독립적으로 실행할 수 있습니다. 그러나 어떤 공작물의 가공에 모드 그룹의 여러 채널 내 여러 프로그램이 관련된 경우에는 다음 코디네이션 명령을 사용하여 여러 채널의 프로그램 순서들을 조율해야 합니다.

요구사항

프로그램 코디네이션에 관련된 모든 채널은 **같은** 모드 그룹에 속해야 합니다.

MD10010 $MC_ASSIGN_CHAN_TO_MODE_GROUP[<Channel>] = <Mode group number>

채널 번호 대신 채널 이름

채널 번호 대신 MD20000 $MC_CHAN_NAME[<Channel index>]에 입력된 채널 이름은 사전 정의된 프로그램 코디네이션 절차의 파라미터로 사용할 수 있습니다. NC 프로그램에서의 채널 이름 사용이 먼저 활성화되어야 합니다.

MD10280 $MN_PROG_FUNCTION_MASK, bit 1 = TRUE

참고

명령 간 최소 거리

명령 INIT, START, WAITE, WAITM, SETM, CLEARM과 명령 WAITMC 사이에 최소 2개의 이송 블록 거리를 유지해야 합니다. WAITMC는 실행 블록이지만 최적화를 위해 이전 블록으로 이동된 후 블록으로서 삭제됩니다. 예를 들어 SETM은 실행 블록이 아니며 다음 블록으로 이동되기 때문에 두 명령 사이에 하나의 블록 거리가 있으면 두 명령 모두 중간 블록에 있게 됩니다. 하나의 블록만 가능하기 때문에 WAITMC의 경우 최적화는 하나의 블록 거리로 수행되지 않습니다.

이로 인해 프로그램이 정지되고 잠깐 동안 처리가 중단됩니다.

구문

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

의미

INIT():	지정된 채널에서 실행될 NC 프로그램을 선택하기 위해 사전 정의된 절차
START():	해당 채널에서 선택된 프로그램을 시작하기 위한 사전 정의 절차
WAITM():	지정된 채널에서 대기 마커에 도달할 때까지 대기하는 사전 정의 절차 지정된 대기 마커는 같은 채널에서 WAITM에 의해 설정됩니다. 이전 블록은 정위치 정지로 종료됩니다. 대기 마커는 동기화 후 삭제됩니다. 각 채널에 최대 10개의 마커를 동시에 지정할 수 있습니다.
WAITE():	하나 이상의 채널에서 프로그램 끝까지 대기하도록 사전 정의된 절차
WAITMC(): [1)]	지정된 채널에서 대기 마커에 도달할 때까지 대기하는 사전 정의 절차 WAITM과 달리 축 제동의 정위치 정지는 다른 채널이 아직 대기 마커에 도달하지 않은 경우에만 개시됩니다.
SETM(): [1)]	채널 코디네이션을 위해 하나 이상의 대기 마커를 설정하는 사전 정의된 절차 자체 채널 내의 처리는 이 명령의 영향을 받지 않습니다. SETM은 채널 리셋과 NC 시작 이후에 계속 유효합니다.
CLEARM(): [1)]	채널 코디네이션을 위해 하나 이상의 대기 마커를 삭제하는 사전 정의된 절차 자체 채널 내의 처리는 이 명령의 영향을 받지 않습니다. CLEARM()은 채널의 모든 대기 마커를 삭제합니다. CLEARM(0)은 대기 마커 "0"만 삭제합니다. CLEARM은 채널 리셋과 NC 시작 이후에 계속 유효합니다.
<ChanNo>:	채널 번호 자체 채널의 번호는 지정할 필요가 없습니다.
	유형: INT
<Prog>:	절대 또는 상대 경로 지정(**옵션**) + 프로그램 이름
	유형: STRING
	경로 데이터 관련 정보는 프로그래밍 매뉴얼 NC 프로그래밍의 "프로그램 메모리 파일 주소 지정" (페이지 623) 장을 참조하십시오.

<AckMode>:	승인 모드 (**옵션**)		
	유형:	CHAR	
	값:	"N"	승인 없음 명령 전송 후 프로그램 실행이 계속 진행됩니다. 명령을 성공적으로 실행할 수 없어도 발송자에게 통보되지 않습니다.
		"S"	동기 승인 수신 콤포넌트가 명령을 승인할 때까지 프로그램 실행이 중지됩니다. 승인 결과가 긍정이면 그 다음 명령이 실행됩니다. 승인 결과가 부정이면 에러 메시지가 출력됩니다.
<MarkNo>:	대기 라벨 번호 **참고** 다중 채널 시스템에서는 최대 100개의 대기 마커가 사용 가능합니다(대기 마커 0 ... 99). 단일 채널 시스템에서는 대기 마커 0만 사용할 수 있습니다.		
1) 사용자별 채널 통신 및 코디네이션의 경우, SETM/CLEARM을 이용하여(또한 조건부 대기 명령 WAITMC를 사용하지 않고) 대기 마커를 적용할 수 있습니다. 대기 마커는 채널 리셋 및 NC 시작 후에도 자신의 값을 유지합니다.			

예

MD20000의 채널 이름을 이용한 START

- 파라미터 지정

```
MD10280 $MN_PROG_FUNCTION_MASK, bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "MACHINING"; 채널 1의 이름
$MC_CHAN_NAME[ 1 ] = "INFEED"; 채널 2의 이름
```

- 프로그래밍

프로그램 코드	설명
START(MACHINING)	; 채널 1의 시작
START(INFEED)	; 채널 2의 시작

로컬 "채널 이름"과 사용자 변수를 사용하여 시작

프로그램 코드	설명
DEF INT MACHINE = 1	; 채널 1에 대한 사용자 변수의 정의
DEF INT LOADER = 2	; 채널 2에 대한 사용자 변수의 정의
...	
START(MACHINE)	; 채널 1의 시작

프로그램 코드	설명
START(LOADER)	; 채널 2의 시작

로컬 "채널 이름", 사용자 변수 및 파라미터가 지정된 채널 이름을 사용하여 시작

프로그램 코드	설명
DEF INT chanNo1	; 채널 1에 대한 사용자 변수의 정의
DEF INT chanNo2	; 채널 2에 대한 사용자 변수의 정의
chanNo1 = CHAN_1	; 채널 1을 채널 이름으로 지정
chanNo2 = CHAN_2	; 채널 2를 채널 이름으로 지정
...	
START(ChanNo1)	; 채널 1의 시작
START(ChanNo2)	; 채널 2의 시작

절대 경로가 지정된 INIT 명령

프로그램 선택 /채널 2의 _N_MPF_DIR/_N_ABSPAN1_MPF.

프로그램 코드

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

프로그램 이름이 지정된 INIT 명령

이름이 "MYPROG"인 프로그램 선택. 제어 시스템이 검색 경로를 사용해 프로그램을 검색합니다.

프로그램 코드

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

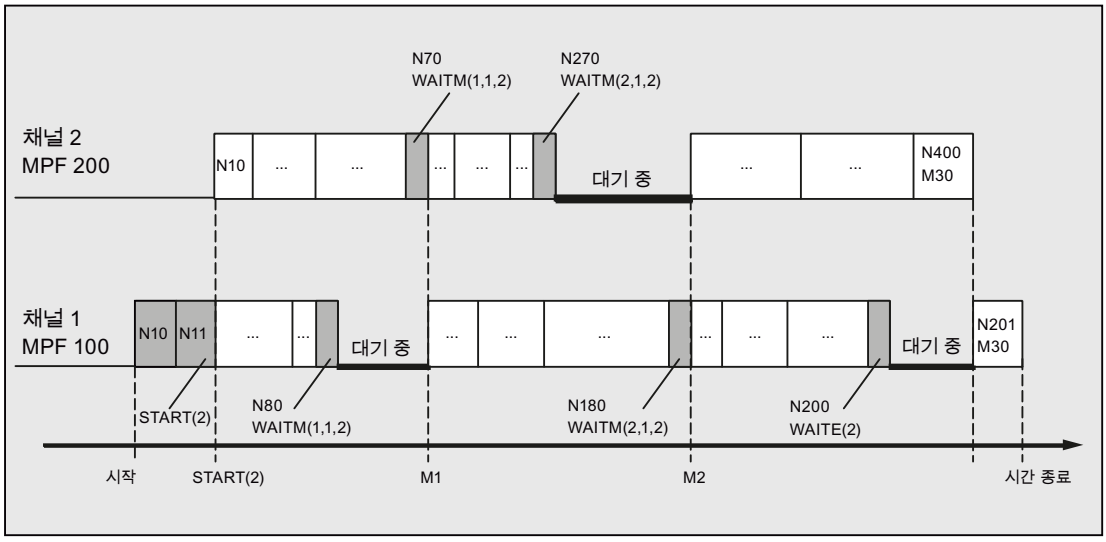
WAITM으로 프로그램 코디네이션

- 채널 1: 프로그램 /_N_MPF_DIR/_N_MPF100_MPF가 이미 선택되어 시작되었습니다.

프로그램 코드	설명
	; 프로그램 MPF100
N10 INIT(2,"MPF200","N")	; 프로그램 MPF200 선택, 채널 2
N11 START(2)	; 채널 2의 시작
...	
N80 WAITM(1,1,2)	; 채널 1과 2에서 WAIT 마커 1을 대기
N81 ...	; 채널 1, N81과 채널 2, N71이 ; 동시에 시작됩니다.
...	
N180 WAITM(2,1,2)	; 채널 1과 2에서 WAIT 마커 2를 대기
N181 ...	; 채널 1, N181과 채널 2, N271이 ; 동시에 시작됩니다.
...	
N200 WAITE(2)	; 채널 2에서 프로그램 종료 대기
N201 ...	; N201은 프로그램이 종료될 때까지 시작되지 않습니다. ; 채널 2에서 MPF200 시작됨.
N201 M30	; 프로그램 채널 1의 종료

- 채널 2: 채널 1에서, 프로그램 MPF200_MPF가 선택되어 블록 N10과 N20을 사용하여 채널 2에서 시작됩니다.

프로그램 코드	설명
;$PATH=/_N_MPF_DIR	; 프로그램 MPF200
...	
N70 WAITM(1,1,2)	채널 1과 2에서 WAIT 마커 1을 대기
N71 ...	; 채널 1, N81과 채널 2, N71이 ; 동시에 시작됩니다.
...	
N270 WAITM(2,1,2)	채널 1과 2에서 WAIT 마커 2를 대기
N271 ...	; 채널 1, N181과 채널 2, N271이 ; 동시에 시작됩니다.
...	
N400 M30	프로그램 채널 2의 종료



보충 조건

WAIT 마커 이후 후속 블록 실행의 비동기 시작

WAIT 마커를 이용한 채널 코디네이션의 경우, 후속 블록 실행이 비동기적으로 시작할 수 있습니다. 이 같은 동작은 공통 WAIT 마커에 도달하기 직전에 동기화될 채널들 중 하나에서 동작이 일어나는 경우에 발생하는데, 그 결과 이동할 거리 삭제에서 암묵적인 재배치(REPOSA)가 일어납니다.

가정: 채널 1과 2에서의 현재 축 지정

- 채널 1: 축 X1 및 U
- 채널 2: 축 X2

도표 4-2 채널 1과 2의 시간 순서

채널 1	채널 2	설명
. . .	. . .	채널 1과 2에서의 임의 처리
N100 WAITM(20,1,2)		채널 1: WAIT 마커에 도달하고 채널 2와의 동기화를 대기함

채널 1	채널 2	설명
GETD(U) 처리의 시작: • *축 교환* • *이동할 거리 삭제* • *REPOSA* *종료*	N200 GETD(U)	채널 2: 채널 1에서 축 U를 요청함 채널 1: 백그라운드에서 GET(U) 처리
	N210 WAITM(20,1,2)	채널 2: WAIT 마커에 도달. ⇒ 이렇게 되면 채널 1 및 2의 동기화가 완료됩니다.
	N220 G0 X2=100	채널 2: N220의 처리 시작
N110 G0 X1=100		채널 1: N110의 처리 시작이 지연됨

4.1.9 매크로 방법 (DEFINE ... AS)

유의사항
매크로 기술은 프로그래밍의 복잡성을 높입니다. 매크로를 사용하면 제어 시스템의 프로그래밍 언어가 대폭 변경될 수 있기 때문에 매크로 테크놀로지는 신중하게 사용해야 합니다.

매크로는 여러 명령문을 하나의 시퀀스로 묶어 고유한 이름을 할당한 것입니다. 프로그램 실행 동안 매크로를 호출하면 프로그램 이름 하에 프로그래밍된 명령문이 차례로 실행됩니다.

유효성 범위(즉, 매크로 정의가 활성화되는 범위)에 따라, 다음과 같은 매크로 범주가 존재합니다.

- 로컬 매크로
  로컬 매크로는 실행 시점을 기준으로 메인 프로그램이 아닌 NC 프로그램의 시작 지점에서 정의된 매크로입니다. LUD는 NC 프로그램이 호출될 때 만들어지고, 프로그램 리셋의 종료와 함께 혹은 제어 시스템에 전원이 켜지는 다음 번에 삭제됩니다. 로컬 매크로는 해당 매크로가 정의된 NC 프로그램 내에서만 액세스할 수 있습니다.
- 프로그램-전역 매크로
  프로그램-전역 매크로는 메인 프로그램으로 사용되는 NC 프로그램의 시작 지점에서 정의되는 매크로입니다. LUD는 NC 프로그램이 호출될 때 만들어지고, 프로그램 리셋의 종료와 함께 혹은 제어 시스템에 전원이 켜지는 다음 번에 삭제됩니다. 프로그램-전역 매크로는 메인 프로그램 및 모든 서브 프로그램에서 접근할 수 있습니다.

  참고

  프로그램-전역 매크로의 사용 가능성

  메인 프로그램에 정의된 프로그램-전역 매크로는 다음 머신 데이터가 설정된 경우 서브 프로그램에서만 사용할 수 있습니다.

  MD11120 $MN_LUD_EXTENDED_SCOPE = 1

  MD11120 = 0이면 메인 프로그램에 정의된 프로그램-전역 매크로는 메인 프로그램에서만 사용할 수 있습니다.

- 전역 매크로
  전역 매크로는 정의 파일(매크로 파일)에서 정의되는 NC 또는 채널-전역 매크로인데, 프로그램 리셋 종료 이후 혹은 제어 시스템에 전원이 켜지는 다음 번에 동일하게 유지됩니다. 전역 매크로는 아무 메인 프로그램 또는 서브프로그램에서 호출해서 실행할 수 있습니다.

  참고

  NC 프로그램에서 **외부** 매크로 파일의 매크로를 사용하려면 매크로 파일을 NC로 다운로드해야 합니다.

매크로는 사용하기 전에 반드시 정의해야 합니다. 사용자 변수를 정의할 때 다음 규칙을 준수해야 합니다.

- 식별자, G, M, H 기능 및 L 서브 프로그램 이름을 매크로에서 정의할 수 있습니다.
- 매크로는 프로그램 시작 시점에서 정의되거나 전용 정의 파일(매크로 파일)에서 정의될 수 있습니다.
- 로컬 및 프로그램-전역 매크로는 프로그램 시작 시점에 정의됩니다.

- 전역 매크로는 반드시 매크로 파일, 즉 ._N_DEF_DIR/_N_UMAC_DEF로 정의되어야 합니다.
- G 코드 매크로는 오직 전역 매크로로만 정의될 수 있습니다.
- H 및 L 코드는 2자리로 프로그래밍할 수 있습니다.
- M 및 G 코드는 3개의 숫자를 사용하여 프로그래밍할 수 있습니다.

참고

키워드 및 예약된 이름은 매크로로 덮어쓰기 할 수 없습니다. 이것은 GOTO 명령 내의 모든 점프 목적지와 FOR, WHILE, LOOP, REPEAT와 같은 프로그램 루프의 키워드에도 적용됩니다.

구문

매크로 정의:
```
DEFINE <Macro_name> AS <Operation_1> <Operation_2> ...
```
NC 프로그램에서의 호출:
```
<Macro_name>
```

의미

DEFINE ... AS :	매크로를 정의하기 위한 키워드 조합
<Macro_name>:	매크로 이름 식별자만 매크로 이름으로 허용됩니다. 매크로는 매크로 이름을 사용하여 NC 프로그램에서 호출됩니다.
<Operation_1>:	매크로에서의 첫 번째 프로그래밍 지시
<Operation_2>:	매크로에서의 두 번째 프로그래밍 지시

예

예 1: 프로그램 시작에 매크로 정의

프로그램 코드	설명
DEFINE LINE AS G1 G94 F300	; 매크로 정의
...	
N70 LINE X10 Y20	; 매크로 호출
...	

예 2: 매크로 파일에 매크로 정의

프로그램 코드	설명
DEFINE M6 AS L6	; 공구 교환 시 데이터 전송 처리에 필요한 서브프로그램이 호출됩니다. 실제 공구 변경 M 함수는 서브프로그램에 출력됩니다 (예: M106).
DEFINE G81 AS DRILL(81)	; DIN-G 코드 에뮬레이션.
DEFINE G33 AS M333 G333	; 나사 절삭 동안 PLC에서 동기화가 요청됩니다. 원래 G 코드 G33은 머신 데이터에 의해 G333으로 이름이 변경되었으며, 이에 따라 사용자의 프로그래밍 방법은 동일합니다.

예 3: 외부 매크로 파일

외부 매크로 파일을 제어 시스템으로 읽어온 후 매크로 파일을 NC에 다운로드해야 합니다. NC 프로그램에서만 매크로를 사용할 수 있습니다.

프로그램 코드	설명
%_N_UMAC_DEF	
;$PATH=/_N_DEF_DIR	; 고객별 매크로
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; CW 방향 스핀들 회전, 절삭유 ON
DEFINE M14 AS M4 M7	; CCW 방향 스핀들 회전, 절삭유 ON
DEFINE M15 AS M5 M9	; 스핀들 정지, 절삭유 OFF
DEFINE M6 AS L6	; 공구 교환 프로그램 호출
DEFINE G80 AS MCALL	; 드릴링 싸이클 선택 취소
M30	

4.2 서브프로그램 방법

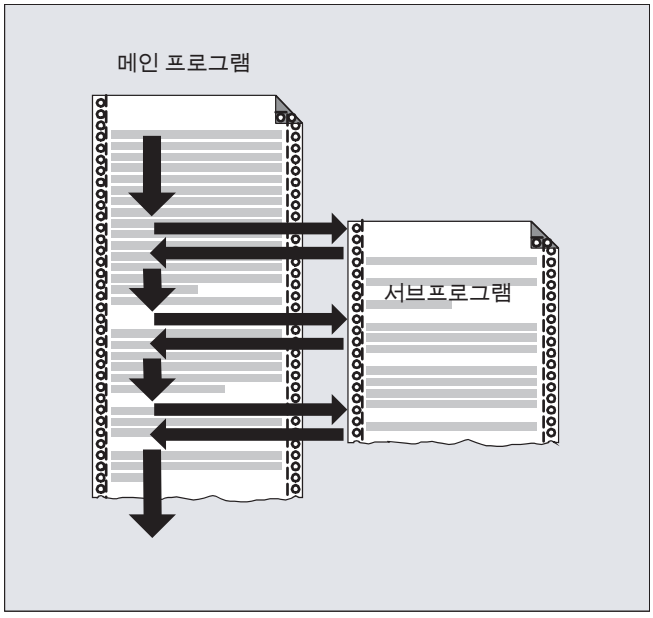
4.2.1 일반 정보

4.2.1.1 서브프로그램

"서브프로그램"은 가공 프로그램을 메인 프로그램과 서브프로그램으로 나누던 시절에 만들어진 용어입니다. 그 당시 메인 프로그램은 제어 시스템 상에서의 처리를 위해 선택하고 실행시키는 가공 프로그램이었습니다. 서브프로그램은 메인 프로그램 내에서 호출하는 가공 프로그램이었습니다.

오늘날의 SINUMERIK NC 언어에서는 이처럼 엄격하게 두 프로그램을 구분하지 않습니다. 원칙적으로 각 가공 프로그램은 메인 프로그램으로 선택이 가능하며 다른 메인 프로그램에서 서브프로그램으로 실행 또는 호출할 수도 있습니다.

따라서 다른 가공 프로그램에서 호출한 가공 프로그램을 서브프로그램이라고 부를 수 있습니다.



적용

모든 고급 프로그래밍 언어가 그렇듯이 NC 언어에서도 서브프로그램은 1회 이상 사용된 프로그램 섹션을 독립적인 자립식 프로그램으로 바꿉니다.

서브프로그램을 사용하면 다음과 같은 이점이 있습니다.

- 프로그램을 읽기 쉽고 간단하게 할 수 있습니다.
- 검증된 프로그램 일부를 재사용함으로써 품질을 향상할 수 있습니다
- 특정 가공 라이브러리 생성 기능을 제공합니다.
- 메모리 공간 절약

4.2.1.2 서브프로그램 이름

이름 지정 규칙

서브프로그램 이름은 다음 규칙에 따라 자유롭게 선택할 수 있습니다.

- 허용되는 문자:
  - 문자: A ... Z, a ... z
  - 숫자: 0 ... 9
  - 밑줄: _
- 처음 2개의 문자는 반드시 2개의 글자이거나 밑줄 다음에 1개의 글자로 구성해야 합니다.

  참고

  이 조건이 만족되면 프로그램 이름을 지정하여 다른 프로그램에서 NC 프로그램을 서브프로그램으로 호출할 수 있습니다. 하지만 프로그램 이름이 숫자로 시작되면 CALL 명령문을 통해서만 서브프로그램을 호출할 수 있습니다.

- 최대 길이: 24문자

참고

대문자/소문자

SINUMERIK NC 언어는 대문자와 소문자를 구분하지 **않습니다**.

참고

허용되지 않는 프로그램 이름

Windows 응용 프로그램에서의 문제 방지를 위해 다음과 같은 프로그램 이름은 사용할 수 **없습니다**.

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

시스템 내부 확장자

서브프로그램을 생성할 때 지정한 프로그램 이름을 제어 시스템에서 접두어 및 접미어를 추가하여 확장할 수 있습니다.

- 접두어: `_N_`
- 접미어: `_SPF`

프로그램 이름 사용

서브프로그램 호출 컨텍스트에서 프로그램 이름을 사용할 때 접두어, 프로그램 이름 및 접미어를 어떤 식으로든 조합할 수 있습니다.

예:

다음 식별자를 사용해 프로그램 이름이 SUB_PROG 인 서브프로그램을 시작할 수 있습니다.

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

이름이 같은 메인 프로그램과 서브프로그램

프로그램 이름이 동일한 메인 프로그램 (.MPF) 과 서브프로그램 (.SPF) 이 존재하는 경우, NC 프로그램에서 이 프로그램 이름을 사용할 때 적절한 파일 확장자를 지정하여 프로그램을 고유하게 식별할 수 있도록 해야 합니다. 그렇지 않으면 지정된 이름을 가진 프로그램 중 검색 경로에서 처음 발견된 프로그램이 사용됩니다.

파일 이름 및 프로그램 이름

이 파일에 포함된 파일 이름과 프로그램 이름은 항상 동일해야 합니다.

따라서 하나의 파일에는 항상 하나의 프로그램이 포함됩니다.

4.2.1.3 서브프로그램 중첩

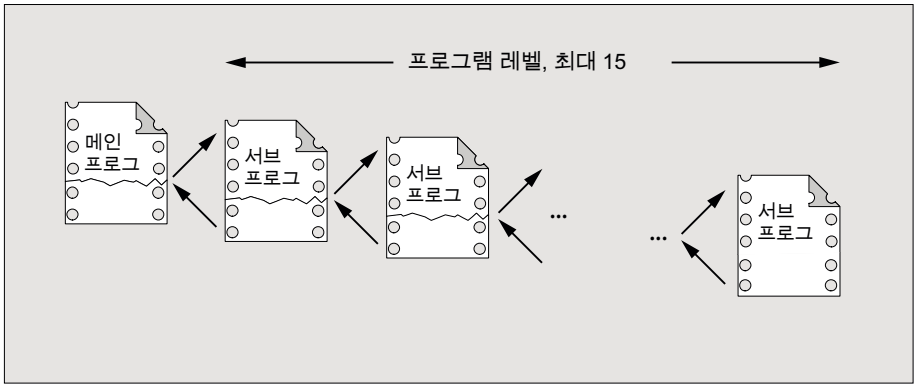
메인 프로그램은 서브프로그램을 호출할 수 있고, 호출된 서브프로그램은 다시 더 많은 서브프로그램을 호출할 수 있습니다. 따라서 여러 프로그램의 순서가 중첩됩니다. 각 프로그램은 전용 프로그램 레벨에서 실행됩니다.

중첩 깊이

NC 언어는 현재 16개의 프로그램 레벨을 제공합니다. 메인 프로그램은 항상 최상위 프로그램 레벨인 0에서 실행됩니다. 서브프로그램은 호출한 프로그램의 레벨 바로 아래 프로그램 레벨에서 실행됩니다. 따라서 첫 번째 서브프로그램의 레벨은 프로그램 레벨 1이 됩니다.

프로그램 레벨 구분:

- 프로그램 레벨 0: 메인 프로그램 레벨
- 프로그램 레벨 1 ~ 15: 서브프로그램 레벨 1 ~ 15



인터럽트 루틴 (ASUB)

서브프로그램이 인터럽트 루틴 컨텍스트에서 호출된 경우, 해당 서브프로그램은 채널 내에 현재 활성화된 프로그램 레벨 (n) 에서 실행되지 않고 그 아래 프로그램 레벨 (n+1) 에서 실행됩니다. 가장 낮은 프로그램 레벨에서도 이런 식으로 실행될 가능성이 있기 때문에 인터럽트 루틴에서는 2개의 프로그램 레벨 (레벨 16 및 17) 을 추가로 사용할 수 있습니다.

2개 이상의 프로그램 레벨이 필요한 경우 채널에서 실행되는 가공 프로그램을 설정할 때 명시적으로 지정해야 합니다. 즉, 지정된 최대 프로그램 레벨까지만 사용할 수 있도록 해서 인터럽트 처리에 사용할 프로그램 레벨을 충분히 남겨두어야 합니다.

예를 들어 인터럽트 처리에 4개의 프로그램 레벨이 필요한 경우 최대 프로그램 레벨 13까지 사용할 수 있도록 가공 프로그램을 설정해야 합니다. 그래야만 인터럽트 발생 시 인터럽트 처리에 필요한 4개 프로그램 레벨 (레벨 14~17) 을 가공 프로그램이 사용할 수 있습니다.

Siemens 사이클

Siemens 사이클은 3개의 프로그램 레벨을 필요로 합니다. 따라서 Siemens 사이클은 마지막으로 호출되어야 합니다.

- 가공 프로그램 처리: 프로그램 레벨 12
- 인터럽트 루틴: 프로그램 레벨 14

4.2.1.4 검색 경로

경로 정보가 없는 서브프로그램을 호출하면 제어 시스템이 사전 정의된 검색 시퀀스에 따라 사용 가능한 프로그램 메모리를 검색합니다 ("서브프로그램 호출을 위한 경로 검색 (페이지 628)" 참조).

4.2.1.5 유형 파라미터 및 실제 파라미터

유형 파라미터 및 실제 파라미터는 파라미터 전송이 포함된 서브프로그램의 정의 및 호출과 관련된 파라미터입니다.

유형 파라미터

서브프로그램을 정의할 때 서브프로그램으로 전송될 파라미터 (유형 파라미터) 를 해당 파라미터의 유형 및 파라미터 이름과 함께 정의해야 합니다.

즉, 유형 파라미터가 서브프로그램의 인터페이스를 정의하게 됩니다.

예제:

프로그램 코드	설명
`PROC CONTOUR (REAL X, REAL Y)`	`; 유형 파라미터: X 및 Y는 실수 형식`
`N20 X1=X Y1=Y`	`; 축 X1을 X 위치로, 축 Y1을 Y 위치로 이송`
`...`	
`N100 RET`	

실제 파라미터

서브프로그램을 호출할 때 절대값 혹은 절대 변수 (실제 파라미터) 를 해당 서브프로그램에 전송해야 합니다.

그에 따라 실제 파라미터는 서브프로그램이 호출될 때 해당 서브프로그램의 인터페이스에 가장 최근의 값을 지정합니다.

예제:

프로그램 코드	설명
`N10 DEF REAL WIDTH`	`; 변수 정의`
`N20 WIDTH=20.0`	`; 변수 지정`
`N30 CONTOUR(5.5, WIDTH)`	`; 실제 파라미터를 사용해 서브프로그램 호출: 5.5 및 WIDTH`
`...`	
`N100 M30`	

4.2.1.6 파라미터 전송

파라미터 전송이 포함된 서브프로그램 정의

파라미터 전송이 포함된 서브프로그램은 PROC 키워드와 서브프로그램에서 요구하는 모든 파라미터가 포함된 완전한 파라미터 목록을 사용해 정의합니다.

불완전한 파라미터 전송

서브프로그램이 호출된 경우 해당 서브프로그램 인터페이스에 정의된 모든 파라미터를 명시적으로 전송할 필요는 없습니다. 전송에서 제외된 파라미터 대신 기본값 "0"이 전송됩니다.

단, 파라미터 순서는 분명하게 식별이 가능해야 하기 때문에 쉼표를 파라미터 구분자로 항상 포함시켜야 합니다. 마지막 파라미터는 예외입니다. 마지막 파라미터가 호출에서 제외되면 마지막 쉼표 역시 전송되지 않습니다.

예제:

서브프로그램:

프로그램 코드	코멘트
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; 유형 파라미터: X, Y 및 Z
...	
N100 RET	

메인 프로그램:

프로그램 코드	설명
PROC MAIN_PROG	
...	
N30 SUB_PROG(1.0,2.0,3.0)	; 전체 파라미터 전송을 통해 서브프로그램 호출: X=1.0, Y=2.0, Z=3.0
…	
N100 M30	

N30에서 불완전한 파라미터 전송을 포함한 서브프로그램을 호출한 예:

```
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
```

유의사항
기준별 호출 파라미터 전송 기준별 호출을 통해 전송되는 파라미터는 서브프로그램 호출 시 함께 전송되어야 합니다.

유의사항
AXIS 데이터 유형 AXIS 데이터 유형의 파라미터는 서브프로그램 호출 시 함께 전송되어야 합니다.

전송 파라미터 점검

시스템 변수 $P_SUBPAR [n] (여기서 n = 1, 2,...) 을 사용해 파라미터가 명시적으로 전송되었는지 아니면 서브프로그램에 남겨졌는지 점검할 수 있습니다. 인덱스 n은 유형 파라미터의 순서를 나타냅니다. 인덱스 n = 1은 첫 번째 유형 파라미터를, 인덱스 n = 2는 두 번째 유형 파라미터를 나타내며 그 다음도 이런 식으로 계속됩니다.

다음 예는 프로그램의 일부를 발췌한 것으로 첫 번째 유형 파라미터를 기준으로 어떻게 점검이 실행되는지 보여줍니다.

프로그래밍	코멘트
`PROC SUB_PROG (REAL X, REAL Y, REAL Z)`	`; 유형 파라미터: X, Y 및 Z`
`N20 IF $P_SUBPAR[1]==TRUE`	`; 첫 번째 유형 파라미터 X를 검사합니다.`
`...`	`; 유형 파라미터 X가 명시적으로 전송된 경우 이 작업들이 실행됩니다.`
`N40 ELSE`	
`...`	`; 유형 파라미터 X가 전송되지 않은 경우 이 작업들이 실행됩니다.`
`N60 ENDIF`	
`...`	`; 일반 작업`
`N100 RET`	

4.2.2 서브프로그램 정의

4.2.2.1 파라미터 전송이 없는 서브프로그램

파라미터 전송이 없는 서브프로그램을 정의할 때는 프로그램의 시작에서 정의 라인을 생략할 수 있습니다.

구문

```
[PROC <프로그램 이름>]
...
```

의미

PROC:	프로그램 시작에 정의 연산
<프로그램 이름>:	프로그램 이름

예제

예제 1: PROC 연산을 포함한 서브프로그램

프로그램 코드	설명
PROC SUB_PROG	; 정의 라인
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; 서브프로그램 리턴

예제 2: PROC 연산이 없는 서브프로그램

프로그램 코드	설명
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; 서브프로그램 리턴

다음 또한 참고:

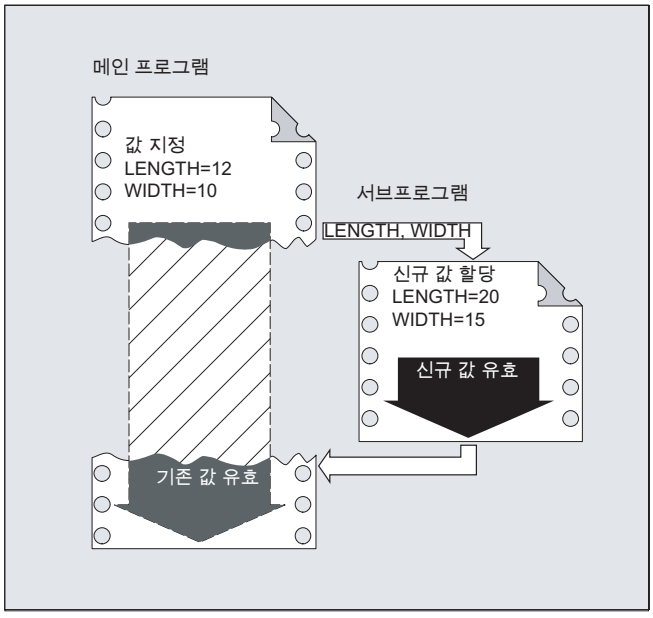
파라미터 전송 없이 서브프로그램 호출 (페이지 587)

4.2.2.2 값별 호출 파라미터 전송을 포함한 서브프로그램 (PROC)

값별 호출 파라미터 전송이 포함된 서브프로그램은 프로그램의 이름 그리고 유형 및 이름이 포함된 완전한 파라미터 목록이 뒤따르는 PROC 키워드를 사용하여 정의됩니다. 정의 연산은 첫 번째 프로그램 라인에 위치해야 합니다.

값별 호출

호출 프로그램은 값별 호출 파라미터 전송을 통해 변수 값만을 서브프로그램으로 전송합니다. 따라서 서브프로그램에는 변수에 대한 직접적인 액세스가 주어지지 않습니다. 이런 식으로 파라미터 값이 바뀔 때 서브프로그램에 보이는 값만 수정됩니다. 호출 프로그램에서 정의된 변수 값은 바뀌지 않습니다. 결과적으로, 값별 호출 파라미터 전송은 호출 프로그램에 영향을 미치지 않습니다.



구문

```
PROC <program name> (<parameter type> <parameter
name>=<init_value>, ...)
```

참고

최대 127개의 파라미터가 전송됩니다.

의미

PROC:	프로그램 시작에 정의 연산
<program name>:	프로그램 이름
<parameter type>:	파라미터의 데이터 유형 (예: REAL, INT, BOOL)

<parameter name>:	파라미터 이름
<init_value>:	파라미터 초기화를 위한 **옵션** 값 서브프로그램을 호출할 때 파라미터를 지정하지 않으면 파라미터에 초기화 값이 지정됩니다.

예

예 1

REAL 유형의 파라미터 3개에 디폴트 값을 적용하여 서브프로그램 SUB_PROG 정의:

프로그램 코드

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

예 2

다양한 호출 버전

프로그램 코드

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; 호출 변용
  SUB_PROG
  SUB_PROG(PAR_1, PAR_2, PAR_3)
  SUB_PROG(PAR_1)
  SUB_PROG(PAR_1, , PAR_3)
  SUB_PROG( , , PAR_3)
N100 RET
```

다음 또한 참고:

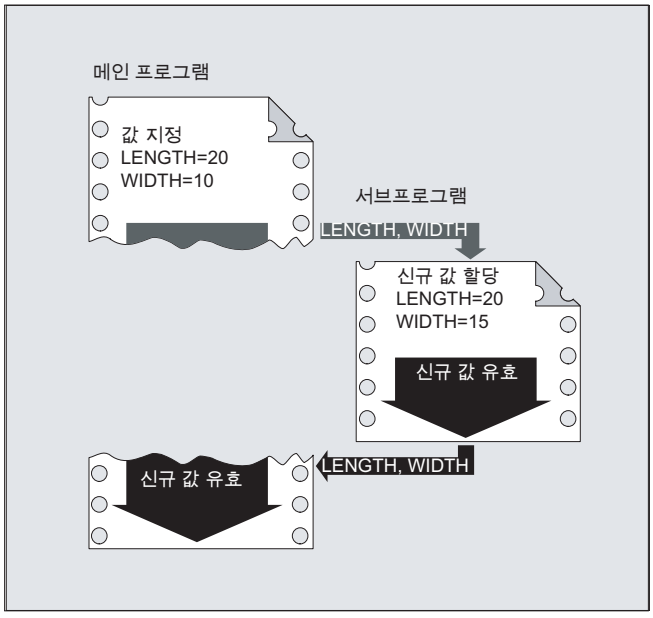
파라미터 전송을 포함한 서브프로그램 호출 (EXTERN) (페이지 590)

4.2.2.3 기준별 호출 파라미터 전송을 포함한 서브프로그램 (PROC, VAR)

기준별 호출 파라미터 전송이 포함된 서브프로그램은 PROC 키워드, 뒤이어 프로그램 이름 그리고 VAR 키워드, 유형 및 이름이 포함된 모든 파라미터의 전체 목록에 의해 정의됩니다. 정의 연산은 첫 번째 프로그램 라인에 위치해야 합니다. 파라미터와 마찬가지로 배열 기준도 전송할 수 있습니다.

기준별 호출

호출 프로그램은 기준별 호출 파라미터 전송에 따른 변수 값이 아니라 변수에 대한 기준 (포인터)를 전송합니다. 이는 서브프로그램에 변수에 대한 직접적인 액세스를 제공합니다. 이런 식으로 파라미터 값이 바뀔 때 서브프로그램에 보이는 값이 수정될 뿐 아니라 호출 프로그램에 정의된 변수의 값도 수정됩니다. 따라서 기준별 호출 파라미터 전송은 호출 프로그램에 영향을 미치고, 심지어 서브프로그램이 끝난 뒤에도 영향을 줍니다.



참고

전송된 변수가 호출 프로그램 내부 (LUD)에 정의되어 있는 경우에만 기준별 호출 파라미터 전송이 필요합니다. 채널 전역 변수 또는 NC 전역 변수는 서브프로그램 내에서 직접 액세스할 수 없기 때문에 전송할 필요가 없습니다.

구문

```
PROC <program name> (VAR <parameter type> <parameter name>, etc.)
PROC <program name> (VAR <array type> <array name>,
[<m>,<n>,<o>], etc.)
```

참고

최대 127개의 파라미터가 전송됩니다.

의미

PROC:	프로그램 시작에 정의 연산	
VAR:	기준별 파라미터 전송 키워드	
<program name>:	프로그램 이름	
<parameter type>:	파라미터의 데이터 유형 (예: REAL, INT, BOOL)	
<parameter name>:	파라미터 이름	
<array type>:	배열 요소의 데이터 유형 (예: REAL, INT, BOOL)	
<array name>:	배열 이름	
[<m>,<n>,<o>]:	배열 크기 현재 최대 3차원 배열까지 사용 가능	
	<m>:	1차원 배열 크기
	<n>:	2차원 배열 크기
	<o>:	3차원 배열 크기

참고

- PROC 키워드 다음에 지정된 프로그램 이름은 사용자 인터페이스 상에서 지정된 프로그램 이름과 일치해야 합니다.
- 정의되지 않은 배열 길이의 배열이 있는 경우 서브프로그램은 변수 길이의 배열을 유형 파라미터로 처리할 수 있습니다. 예를 들어 2차원 배열을 유형 파라미터로 정의하는 경우 1차원 배열의 길이가 지정되지 않습니다. 이 경우 쉼표를 대신 표시해야 합니다.
  예: `PROC <program name> (VAR REAL ARRAY[,5])`

예

REAL 유형의 파라미터 2개를 이용한 서브프로그램 정의:

프로그램 코드

```
; 파라미터 1: 유형 기준: REAL, 이름: LENGTH
; 파라미터 2: 유형 기준: REAL, 이름: WIDTH
PROC SUB_PROG(VAR REAL LENGTH, VAR REAL WIDTH)
```

다음 또한 참고:

파라미터 전송을 포함한 서브프로그램 호출 (EXTERN) (페이지 590)

4.2.2.4 모달 G 코드 저장 (SAVE)

SAVE 속성은 서브프로그램 호출 이전에 활성 모달 G 코드가 저장되고 서브프로그램 종료 후 다시 활성화 된다는 것을 의미합니다.

유의사항
연속 경로 모드 중단 연속 경로 모드를 실행 중인 상태에서 SAVE 속성으로 서브프로그램을 호출하면 서브프로그램 끝에서 연속 경로 모드가 중단됩니다 (리턴 점프).

구문

```
PROC <subprogram name> SAVE
```

의미

SAVE:	서브프로그램 호출 이전에 모달 G 코드를 저장하고 서브프로그램 종료 후 복원합니다.

예

CONTOUR 서브루틴에서는 모달 G코드 G91 증분 치수가 적용됩니다. 모달 G 코드 G90 은 메인 프로그램에서 유효합니다(절대 치수). SAVE를 정의한 서브프로그램으로 인해 서브프로그램 종료 후 메인 프로그램에서 G90이 다시 적용됩니다.

서브프로그램 정의:

프로그램 코드	설명
`PROC CONTOUR (REAL VALUE1) SAVE`	`; SAVE 파라미터를 통한 서브프로그램 정의`
`N10 G91 ...`	`; 모달 G 코드 G91: 증분치`
`N100 M17`	`; 서브프로그램 종료`

메인 프로그램:

프로그램 코드	설명
`N10 G0 X... Y... G90`	`; 모달 G 코드 G90: 절대치`
`N20 ...`	
`...`	
`N50 CONTOUR (12.4)`	`;서브프로그램 호출`
`N60 X... Y...`	`; SAVE를 사용하여 모달 G 코드 G90 재활성화`

보충 조건

프레임

SAVE 특성을 통한 서브프로그램과 관련된 프레임 동작은 프레임 시간에 따라 다르며 머신 데이터를 사용하여 설정할 수 있습니다.

4.2.2.5 싱글 블록 실행 억제 (SBLOF, SBLON)

사용자는 활성 싱글 블록 가공으로도 중단 없이 완전히 또는 부분적으로 NC 프로그램을 처리할 수 있습니다. 싱글 블록 가공은 SBLOF 명령을 통해 억제되고 SBLON 명령을 통해 다시 활성화됩니다.

전체 NC 프로그램의 싱글 블록 가공 억제

메인 프로그램의 첫 번째 라인(PROC ...)에 싱글 블록 가공(SBLOF)의 비활성화가 프로그래밍된 경우 NC 프로그램이 종료될 때까지 또는 NC 프로그램이 취소될 때까지 유효합니다. 그런 다음 싱글 블록 모드에서 NC 프로그램이 중지되지 않고 실행됩니다.

서브프로그램의 첫 번째 라인(PROC ...)에 싱글 블록 가공(SBLOF)의 비활성화가 프로그래밍된 경우 NC 프로그램이 종료될 때까지 또는 NC 프로그램이 취소될 때까지 유효합니다. 프로그래밍된 리턴 명령을 사용해 서브프로그램 종료 시 중단할지 여부를 결정합니다.

- M17로 리턴: 서브프로그램 종료 시 중단
- RET로 리턴: 서브프로그램 종료 시 중단 **없음**

NC 프로그램 내에서 싱글 블록 가공 억제

NC 프로그램 내 블록에 싱글 블록 가공(SBLOF)의 비활성화가 프로그래밍된 경우 이 블록에서 다음 프로그래밍된 싱글 블록 가공(SBLON)의 활성화까지 또는 활성 서브프로그램 레벨의 종료 시 싱글 블록 가공이 비활성화됩니다.

구문

전체 NC 프로그램의 싱글 블록 가공 억제:

```
PROC ... SBLOF
...
```

NC 프로그램 내에서 싱글 블록 가공 억제:

```
...
SBLOF
...
SBLON
...
```

의미

PROC:	프로그램의 첫 번째 실행	
SBLOF:	사전 정의된 싱글 블록 가공 비활성화 절차	
	블록에서만:	예, PROC 블록에서 가능
	효율성:	모달
SBLON:	사전 정의된 싱글 블록 가공 활성화 절차	
	블록에서만:	예
	효율성:	모달

특수한 측면

- **싱글 블록 가공이 억제된 블록 표시**
  현재 블록 표시는 서브프로그램에서 DISPLOF를 사용해 억제될 수 있습니다. 서브프로그램 내 싱글 블록 정지를 위해 DISPLOF가 SBLOF와 함께 프로그래밍된 경우 서브프로그램 호출이 표시됩니다.
- **비동기 서브프로그램(ASUB)으로 싱글 블록 가공 억제**
  활성 싱글 블록 가공으로 ASUB를 한 번에 실행하려면 PROC 구문을 SBLOF를 사용해 ASUB에 프로그래밍해야 합니다. 또한 이 작업은 작동 "편집 가능한 시스템 ASUB"(MD11610 $MN_ASUP_EDITABLE)에도 적용됩니다.
  PROC 라인에 SBLOF를 프로그래밍하거나 머신 데이터 MD10702 $MN_IGNORE_SINGLEBLOCK_MASK(bit0 = 1 또는 bit1 = 1)의 설정을 사용하여 시스템 또는 사용자 ASUB에서 싱글 블록 정지가 억제된 경우 ASUB에 SBLON을 프로그래밍하여 싱글 블록 정지를 다시 활성화할 수 있습니다.
  사용자 ASUB에서 머신 데이터 MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP의 설정을 사용하여 싱글 블록 정지를 억제한 경우 ASUB에서 SBLON을 프로그래밍하여 싱글 블록 정지를 다시 활성화할 수 **없습니다**.

- **다양한 싱글 블록 가공 유형의 특수 기능**
  - "SB2: Arithmetic block" AND MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, Bit 12 = 1:
    → 프로그램이 SBLON 블록에서 정지되지 않습니다.
  - "SB3: single block fine":
    → SBLOF 명령이 억제됩니다.
- **중첩된 프로그램에서 싱글 블록 가공 억제**
  SBLOF가 서브프로그램의 PROC 구문에 프로그래밍된 경우 M17로 서브프로그램 리턴 시 실행이 정지됩니다. 이를 통해 호출 프로그램의 다음 블록이 실행되는 것을 방지합니다. SBLOF가 있는 서브프로그램에서 PROC 구문에 SBLOF 없이 싱글 블록 가공의 억제가 활성화된 경우 호출 프로그램의 다음 가공 펑션 블록 이후에 실행이 정지됩니다. 이러한 작동을 원하지 않는 경우 리콜(M17) 이전에 서브프로그램에서 SBLON을 프로그래밍해야 합니다. RET로 더 높은 레벨의 프로그램으로 리턴하기 위해 실행이 정지되지 않습니다.

예

예 1: NC 프로그램 내에서 싱글 블록 가공 억제

초기 상황: 싱글 블록 가공이 활성화됩니다.

프로그램 코드	설명
N10 G1 X100 F1000	
N20 **SBLOF**	; 싱글 블록 가공 비활성화
N30 Y20	
N40 M100	
N50 R10=90	
N60 **SBLON**	; 싱글 블록 가공 재활성화
N70 M110	
N80 ...	

N20과 N60 사이의 영역이 싱글 블록 모드에서 한 번에 실행됩니다.

예 2: 싸이클을 사용자를 위한 하나의 명령으로 실행

초기 상황: 싱글 블록 가공이 활성화됩니다.

메인 프로그램:

프로그램 코드
...
N100 G1 X10 G90 F200
N120 X-4 Y6
N130 **CYCLE1**
N140 G1 X0

프로그램 코드
N150 M30

싸이클 CYCLE1:

프로그램 코드	설명
N100 **PROC** CYCLE1 DISPLOF **SBLOF**	; 전체 프로그램의 싱글 블록 가공 억제.
N110 R10=3*SIN(R20)+5	
N120 IF (R11 <= 0)	
N130 SETAL(61000)	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

싸이클 CYCLE1은 활성 싱글 블록 가공으로 처리됩니다. 활성 싱글 블록 가공을 위해 CYCLE1을 처리합니다. CYCLE1을 처리하려면 시작 키를 눌러야 합니다.

예 3: 수정된 워크 옵셋 및 공구 옵셋을 활성화하기 위해 PLC에 의해 시작된 ASUB는 보이지 않게 실행되어야 합니다.

프로그램 코드	설명
N100 **PROC** NV **SBLOF DISPLOF**	; 싱글 블록 가공 및 블록 표시 억제.
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	

예 4: 서브프로그램에서 특정 정지

초기 상황:

- 싱글 블록 가공이 활성화됩니다.
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, Bit12 = 1

메인 프로그램:

프로그램 코드	설명
N10 G0 X0	; 싱글 블록 정지
N20 X10	; 싱글 블록 정지
N30 **CYCLE**	; 싸이클에 의해 생성된 이송 블록.
N50 G90 X20	; 싱글 블록 정지
M30	

싸이클 CYCLE:

프로그램 코드	설명
PROC CYCLE **SBLOF**	; 싱글 블록 가공 억제
N100 R0=1	
N110 **SBLON**	; MD10702 bit12 = 1이어서 싱글 블록 정지 없음
N120 X1	; 싱글 블록 정지
N140 **SBLOF**	
N150 R0=2	
RET	

예 5: 프로그램 중첩으로 싱글 블록 가공 억제

초기 상황:

- 싱글 블록 가공 유형 2가 활성화됩니다.
- SBLON 블록에서 프로그램 실행이 중단되면 안 됩니다 (MD10702 $MN_IGNORE_SINGLEBLOCK_MASK, Bit12 = 1).

프로그램 코드	설명
N10 X0 F1000	; 싱글 블록 정지
N20 UP1(0)	
PROC UP1(INT _NR) **SBLOF**	; UP1의 싱글 블록 가공 비활성화
N100 X10	
N110 UP2(0)	
PROC UP2(INT _NR)	
N200 X20	
N210 **SBLON**	; 싱글 블록 가공 활성화
N220 X22	; 싱글 블록 정지
N230 UP3(0)	
PROC UP3(INT _NR)	
N300 **SBLOF**	; 싱글 블록 가공 비활성화
N305 X30	
N310 **SBLON**	; 싱글 블록 가공 활성화
N320 X32	; 싱글 블록 정지
N330 **SBLOF**	; 싱글 블록 가공 비활성화

프로그램 코드	설명
N340 X34	
N350 M17	; 싱글 블록 정지(M17)
N240 X24	; 싱글 블록 정지(N210)
N250 M17	; 싱글 블록 정지(M17)
N120 X12	
N130 M17	; 싱글 블록 정지(M17)
N30 X0	; 싱글 블록 정지
N40 M30	; 싱글 블록 정지

4.2.2.6 현재 블록 디스플레이 억제 (DISPLOF, DISPLON, ACTBLOCNO)

현재 프로그램 블록이 블록 디스플레이에 표준으로 표시됩니다. DISPLOF 명령을 사용해 싸이클 및 서브프로그램에서 해당 블록의 디스플레이를 억제할 수 있습니다. 현재 블록 대신 싸이클 또는 서브프로그램의 호출이 표시됩니다. DISPLON 명령은 블록 디스플레이 억제를 취소합니다.

DISPLOF 및 DISPLON은 PROC 연산으로 프로그램 라인에 프로그래밍합니다. 또한 두 명령은 서브프로그램 전체는 물론 암시적으로 해당 서브프로그램이 호출한 다른 서브프로그램 중 DISPLON 또는 DISPLOF 명령을 포함하지 않은 모든 서브프로그램에도 적용됩니다. 이 사항은 모든 ASUB에 적용됩니다.

구문

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

의미

DISPLOF:	현재 블록 디스플레이를 억제하기 위한 명령.	
	위치:	PROC 연산이 포함된 프로그램 라인의 끝
	적용:	서브프로그램 혹은 프로그램 끝에서 리턴 점프까지
	참고: DISPLOF 명령을 사용해 서브프로그램에서 다른 서브프로그램을 호출하는 경우, 호출된 서브프로그램 중 DISPLON이 명시적으로 프로그래밍된 경우를 제외하고 모든 서브프로그램에서 현재 블록 디스플레이가 억제됩니다.	

DISPLON:	해당 블록 디스플레이 억제를 취소하는 명령어.	
	위치:	PROC 연산이 포함된 프로그램 라인의 끝
	적용:	서브프로그램 혹은 프로그램 끝에서 리턴 점프까지
	참고: DISPLON 명령을 사용하여 서브프로그램에서 다른 서브프로그램을 호출하는 경우, 호출된 서브프로그램 중 DISPLOF가 명시적으로 프로그래밍되어 있는 경우를 제외하고 모든 서브프로그램에 현재 블록이 표시됩니다.	
ACTBLOCNO:	DISPLOF에 ACTBLOCNO 속성이 포함되어 있으면 알람이 발생하는 경우 해당 알람이 발생한 실제 블록의 번호가 출력됩니다. 하위 프로그램 레벨에 DISPLOF만 프로그래밍된 경우에도 이 규칙이 적용됩니다. 반면에 ACTBLOCNO가 없는 DISPLOF의 경우 DISPLOF에서 지정하지 않은 마지막 프로그램 레벨의 싸이클 또는 서브프로그램 호출에 대한 블록 번호가 표시됩니다.	

예

예 1: 싸이클에서 현재 블록 디스플레이 억제

프로그램 코드	코멘트
PROC CYCLE (AXIS TOMOV, REAL POSITION) SAVE DISPLOF	; 현재 블록 디스플레이를 억제하고 대신 싸이클 호출을 표시합니다. 예: CYCLE(X,100.0)
DEF REAL DIFF	;싸이클 내용
G01 ...	
...	
RET	; 서브프로그램 리턴 점프. 싸이클 호출 다음 블록이 블록 디스플레이에 표시됩니다.

예 2: 알람 출력 시 블록 디스플레이

서브프로그램 SUBPROG1 (ACTBLOCNO 사용):

프로그램 코드	코멘트
PROC SUBPROG1 DISPLOF ACTBLOCNO	
N8000 R10 = R33 + R44	
...	
N9040 R10 = 66 X100	; 알람 12080 트리거
...	
N10000 M17	

서브프로그램 SUBPROG2 (ACTBLOCNO 사용 안 함):

프로그램 코드	코멘트
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; 알람 12080 트리거
...	
N7000 M17	

메인 프로그램:

프로그램 코드	코멘트
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; 알람 출력 = "12080 채널 K1 블록 N9040 구문 오류, 텍스트 R10="
N2060 ...	
N2350 SUBPROG2	; 알람 출력 = "12080 채널 K1 블록 N2350 구문 오류, 텍스트 R10="
...	
N3000 M30	

예 3: 현재 블록 디스플레이 억제 취소

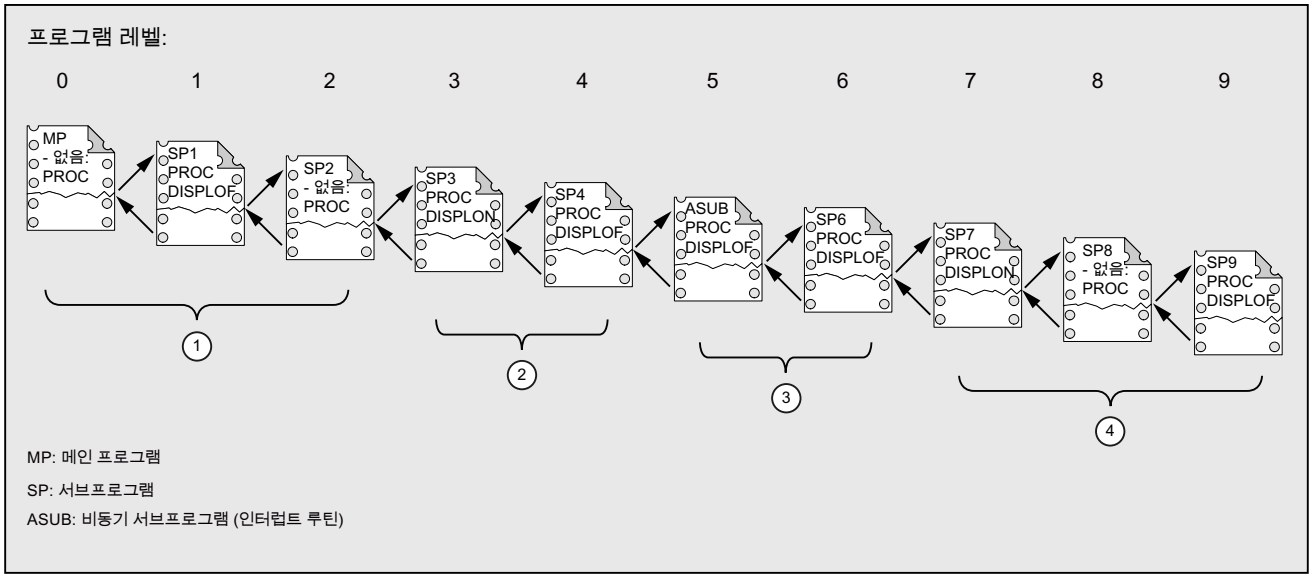
억제가 포함된 서브프로그램 SUB1:

프로그램 코드	코멘트
PROC SUB1 DISPLOF	; SUB1 서브프로그램에서 현재 블록 디스플레이를 억제합니다. 하지만 SUB1이 호출되면 현재 블록이 표시됩니다.
...	
N300 SUB2	; 서브프로그램 SUB2를 호출합니다.
...	
N500 M17	

억제가 없는 서브프로그램 SUB2:

프로그램 코드	코멘트
PROC SUB2 DISPLON	; 서브프로그램 SUB2에서 현재 블록 디스플레이 억제를 취소합니다.
...	
N200 M17	; 서브프로그램 SUB1로 돌아갑니다. SUB1에 현재 블록 디스플레이 억제가 복구됩니다.

예 4: 다양한 DISPLON/DISPLOF 조합에 대한 디스플레이 응답



① 프로그램 레벨 0의 가공 프로그램 라인은 현재 블록 디스플레이에 표시됩니다.

② 프로그램 레벨 3의 가공 프로그램 라인은 현재 블록 디스플레이에 표시됩니다.

③ 프로그램 레벨 3의 가공 프로그램 라인은 현재 블록 디스플레이에 표시됩니다.

④ 프로그램 레벨 7/8의 가공 프로그램 라인은 현재 블록 디스플레이에 표시됩니다.

4.2.2.7 준비 과정에서 서브프로그램 식별 (PREPRO)

모든 파일은 전원을 켜는 동안 PROC 연산 끝에 있는 PREPRO 키워드를 통해 식별할 수 있습니다.

참고

이 유형의 프로그램 전처리는 머신 데이터 MD10700 $MN_PREPROCESSING_LEVEL 설정에 따라 달라집니다. 제조업체 설명서를 참조하십시오.

구문

```
PROC … PREPRO
```

의미

PREPRO:	전원을 켜는 동안 준비된 싸이클 디렉토리에 저장된 NC 프로그램의 모든 파일을 식별하기 위한 키워드

서브프로그램 호출과 준비를 통한 서브프로그램 읽기

싸이클 디렉토리는 전원을 켜는 동안 및 서브프로그램을 호출하는 동안 파라미터로 전처리한 서브프로그램에 대해 동일한 순서로 처리됩니다.

1. _N_CUS_DIR 사용자 싸이클
2. _N_CMA_DIR 제조업체 싸이클
3. _N_CST_DIR 표준 싸이클

동일한 이름을 가진 NC 프로그램이 서로 다른 특성을 가진 경우, 검색된 첫 번째 PROC 연산이 활성화되고 다른 PROC 연산은 알람 메시지 없이 무시됩니다.

4.2.2.8 서브프로그램 복귀 M17

복귀 명령 M17 (또는 가공 프로그램 종료 명령 M30) 은 서브프로그램의 끝에 위치합니다. 이 명령은 서브프로그램 호출 시 가공 프로그램 블록에 있는 호출 프로그램으로 복귀하도록 처리합니다.

참고

M17 및 M30은 NC 언어에서 동일하게 취급됩니다.

구문

```
PROC <program name>
...
M17/M30
```

보충 조건

서브프로그램 복귀가 연속 이송 모드에 미치는 영향

M17 (또는 M30) 이 가공 프로그램 블록에 독립적으로 존재하는 경우 채널 내에서 사용 중인 연속 이송 모드가 인터럽트됩니다.

연속 이송 모드가 인터럽트되는 것을 방지하려면 M17 (또는 M30) 을 마지막 이송 블록에 포함시켜야 합니다. 또한 다음 머신 데이터를 "0"으로 설정해야 합니다.

MD20800 $MC_SPF_END_TO_VDI = 0 (NC/PLC 인터페이스에 M30/M17 출력 안 함)

예

1. 개별 블록에 M17이 포함된 서브프로그램

프로그램 코드	코멘트
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10	
N30 M17	; 연속 이송 모드를 인터럽트하고 리턴 점프.

2. 마지막 이송 블록에 M17이 포함된 서브프로그램

프로그램 코드	코멘트
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	; 연속 이송 모드를 인터럽트하지 않고 리턴 점프.

4.2.2.9 RET 서브프로그램 복귀

서브프로그램에서 RET 명령을 M17 리턴 점프 명령 대신 사용할 수도 있습니다. RET는 개별 가공 프로그램 블록에 프로그래밍되어야 합니다. M17과 마찬가지로 RET는 서브프로그램을 호출한 다음 가공 프로그램 블록에서 호출한 프로그램으로 복귀합니다.

참고

파라미터를 사용하여 RET의 리턴 점프 방식을 변경할 수 있습니다 ("파라미터 지정 가능한 서브프로그램 리턴 점프 (RET ...) (페이지 575)" 참조).

용도

리턴 점프가 발생 되더라도 G64 연속 경로 모드 (G641 ~ G645) 를 중단하지 않으려면 RET 명령을 사용해야 합니다.

요구 사항

RET 명령어는 SAVE 속성으로 정의되지 않은 서브프로그램에만 사용할 수 있습니다.

구문

```
PROC <program name>
...
RET
```

예

메인 프로그램:

프로그램 코드	코멘트
PROC MAIN_PROGRAM	; 프로그램 시작
...	
N50 SUB_PROG	; 서브프로그램 호출: SUB_PROG
N60 ...	
...	
N100 M30	; 프로그램 종료

서브프로그램:

프로그램 코드	코멘트
PROC SUB_PROG	
...	
N100 RET	; 메인 프로그램 내 블록 N60으로 리턴 점프.

4.2.2.10 파라미터 지정 가능한 서브프로그램 리턴 점프 (RET ...)

일반적으로, 리턴 점프는 RET 명령을 사용하여 서브프로그램으로부터 호출 프로그램으로 실행됩니다. 그러면 서브프로그램 호출 다음의 프로그램 라인에서 처리가 계속 진행됩니다. 다른 위치에서 프로그램을 계속 처리해야 할 경우, 다음과 같은 옵션이 있습니다.

- ISO 언어 모드에서 스톡 제거 싸이클을 호출한 후 (형상 설명 이후) 프로그램 실행을 재개하는 경우.
- 에러 처리를 위해 서브프로그램 레벨에서 메인 프로그램으로 복귀하는 경우 (ASUB 이후 포함).
- 컴파일 싸이클 및 ISO 언어 모드에서 특정 적용 분야에 대해 여러 프로그램 레벨을 통한 리턴 점프

이를 위해, RET 명령에 추가 파라미터를 프로그램해야 합니다.

검색 방향

파라미터 <target block>을 지정하면, 리턴 점프가 먼저 호출 블록 이후의 블록으로 실행됩니다. 그 다음, 리턴 점프가 실행된 프로그램의 **종료** 방향으로 대상에 대한 검색이 진행됩니다. 이 검색에 성공하지 못한 경우, 프로그램의 시작을 향하여 검색이 진행됩니다.

구문

```
RET("<target block>")
RET("<target block>",<block after target block>)
RET("<target block>",<block after target block> <number of return
jump levels>)
RET("<target block>", ,<number of return jump levels>)
RET("<target block>",<block after target block>,<number of return
jump levels>,
<return jump to the beginning of the program>)
RET( , ,<number of return jump levels>,<return jump to the
beginning of the program>)
```

의미

RET:	서브프로그램 종료	
<target block>:	파라미터에서 프로그램 실행을 재개해야 하는 블록이 점프 대상으로 지정됩니다. 파라미터 <number of return jump levels>가 프로그래밍되지 않으면 점프 대상이 현재 서브프로그램이 호출되었던 프로그램이 됩니다. 가능한 데이터는 다음과 같습니다.	
	<block number>	대상 블록 번호. 처음에 리턴 점프가 실행되었던 프로그램에서 프로그램의 끝 방향으로 블록 번호에 대한 검색이 실행됩니다.
	<jump marker>	리턴 점프가 실행되는 프로그램에 있어야 하는 점프 마커. 처음에 리턴 점프가 실행되었던 프로그램에서 프로그램의 끝 방향으로 점프 마커에 대한 검색이 실행됩니다.
	<character string>	리턴 점프가 실행되는 프로그램에 있어야 하는 문자열 (예: 프로그램 또는 변수 이름). 처음에 리턴 점프가 실행되었던 프로그램에서 프로그램의 끝 방향으로 문자열에 대한 검색이 실행됩니다. 다음 규칙은 문자열을 프로그래밍할 때 적용됩니다. • **끝에 공란**(끝에 ":"을 붙이는 점프 마커와는 다름). • **문자열 앞**에만 하나의 블록 번호 및/또는 점프 마커를 설정할 수 있으며, **프로그램 명령이 없습니다**.

<block after target block>:	이 파라미터는 프로그램 처리가 <target block> 파라미터에 명시된 블록에서 계속되어야 하는지, 그 다음 블록에서 계속되어야 하는지 지정합니다.		
	유형:	INT	
	값:	0	<target block> 파라미터에 지정된 블록으로 리턴 점프가 실행됩니다.
		> 0	<target block> 파라미터에 지정된 블록 다음 블록으로 리턴 점프가 실행됩니다.
<리턴 점프 레벨 개수>:	이 파라미터는 리턴 점프 후 대상 블록을 검색하고 프로그램 처리를 계속할 프로그램 레벨의 번호를 지정합니다.		
	유형:	INT	
	값:	1	The program is resumed at the 'current program level -1' (just like RET without parameter).
		2	The program is resumed at the 'current program level -2', i.e. one level is skipped.
		3	The program is resumed at the 'current program level -3', i.e. two levels are skipped.
		...	
	값 범위:	1 ... 15	
<프로그램 시작으로 리턴 점프>:	메인 프로그램으로 리턴 점프 시 프로그램이 프로그램 시작 지점에서 계속되어야 하는지를 이 파라미터가 활성 **ISO 모드**에서 지정합니다.		
	유형:	BOOL	
	값:	1	메인 프로그램으로 리턴 점프되고 거기에서 **ISO 언어 모드**가 활성화되면 프로그램은 프로그램의 시작 위치로 분기됩니다.

참고

대상 블록 탐색을 지정하기 위해 문자열을 사용하여 서브프로그램 리턴 점프를 하는 경우 항상 호출 프로그램의 점프 마커부터 탐색을 시작합니다.

문자열을 사용하여 점프 대상을 고유하게 정의해야 하는 경우 문자열이 점프 마커의 이름과 일치하지 않아야 합니다. 일치할 경우 서브프로그램이 항상 문자열이 아닌 점프 마커로 리턴 점프됩니다(예 2 참조).

제한 사항

여러 프로그램 레벨에 걸친 리턴 점프

여러 프로그램 레벨을 통해 리턴 점프하는 경우 개별 프로그램 레벨에 대해 SAVE 명령이 평가됩니다.

여러 프로그램 레벨에 걸쳐 리턴 점프를 하는 경우 모달 서브프로그램이 활성 상태이고 건너뛴 프로그램 중 하나에서 선택 취소 명령 MCALL이 모달 서브프로그램에 프로그래밍되면 모달 서브프로그램은 활성 상태를 유지합니다.

유의사항
프로그래밍 에러 여러 프로그램 레벨을 통과하는 리턴 점프의 경우 필요한 모달 설정에 따라 처리가 계속되도록 하는 것은 사용자의 책임입니다. 이 작업은 적절한 메인 블록을 프로그래밍하여 수행할 수 있습니다.

외부 소스에서 처리

"외부 소스에서 실행" 기능을 통해 외부 프로그램 메모리에서 메인 프로그램과 서브 프로그램을 실행할 경우 RET(...)로 프로그램한 리턴 대상이 리로드 메모리 안에 있어야 합니다. 그렇지 않으면 점프 대상을 찾을 수 없어 프로그램이 중단되고 알람 14000이 출력됩니다.

참고

프로그램된 점프 명령에 대한 제한 없이 외부 프로그램을 실행하려면 "외부 소스에서 실행" 기능 대신 "외부 저장 장치에서 실행(EES)" 옵션을 사용하는 것이 좋습니다.

예

예 1: ASUB 실행 이후 메인 프로그램에서 재개

프로그래밍	설명
N10010 CALL "UP1"	;프로그램 레벨 0 (메인 프로그램)
N11000 PROC UP1	; 프로그램 레벨 1
N11010 CALL "UP2"	
N12000 PROC UP2	; 프로그램 레벨 2
...	
N19000 PROC ASUP	; 프로그램 레벨 3 (ASUB 실행)
...	
N19100 RET("N10900", ,$P_STACK)	; 서브프로그램이 메인 프로그램으로 리턴 점프
	; $P_STACK: 실제 프로그램 레벨
N10900	; 메인 프로그램 내 대상 블록
N10910 MCALL	; 모달 서브루틴 호출 비활성화
N10920 G0 G60 G40 M5	; 추가 모달 설정 초기화

예 2: 대상 블록 탐색을 지정하기 위한 문자열 (string>)

메인 프로그램:

프로그램 코드	설명
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; 서브프로그램 "subProg1" 호출
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; 서브프로그램 "subProg2" 호출
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1: G1 X30	; 여기서 "subProg1"은 점프 마커로 정의됩니다.
N1560 ...	
N1600 subProg3	; 서브프로그램 "subProg3" 호출
N1610 ...	
N1900 M30	

서브프로그램 subProg1:

프로그램 코드	설명
PROC subProg1	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; 블록 N1400에서 메인 프로그램으로 리턴 점프

서브프로그램 subProg2:

프로그램 코드	설명
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; 블록 N1500에서 메인 프로그램으로 리턴 점프

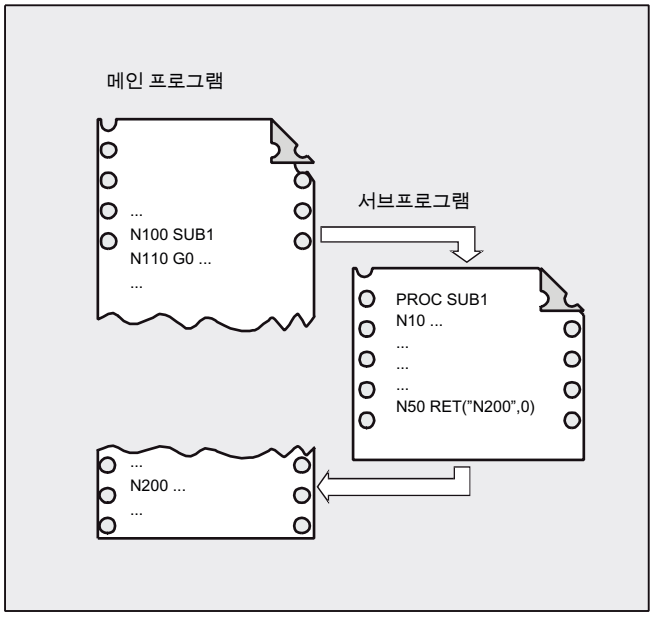
서브프로그램 subProg3:

프로그램 코드	설명
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; 블록 N1550에서 메인 프로그램으로 리턴 점프

추가 정보

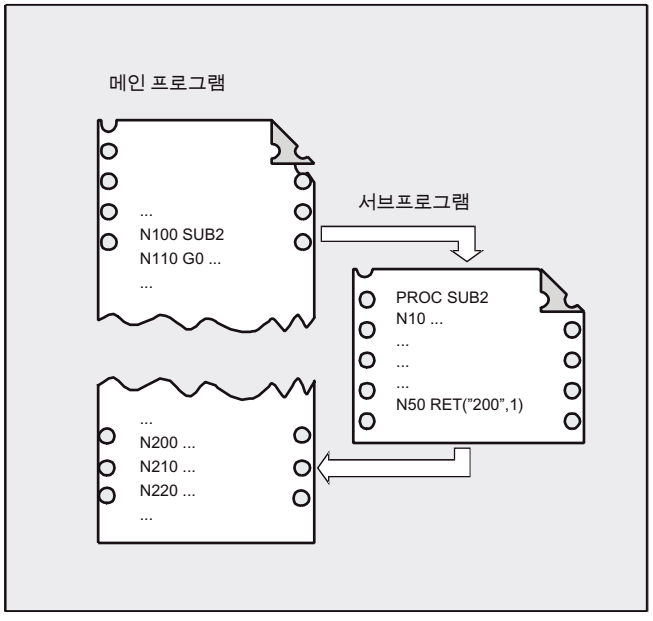
다음 그림은 리턴 점프 파라미터의 여러 가지 효과를 보여줍니다.

1. <target block> = "N200", <block after target block> = 0



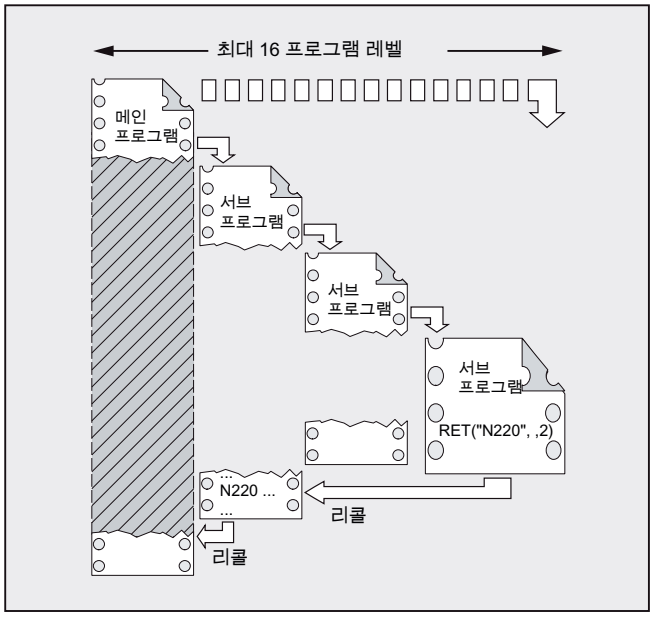
RET 명령 이후 프로그램 실행은 메인 프로그램의 블록 N200에서 계속됩니다.

2. <target block> = "N200", <block after target block> = 1



RET 명령 이후 프로그램 실행은 메인 프로그램의 블록 N200 다음에 오는 블록 (N210) 에서 계속됩니다.

3. <target block> = "N220", <number of return jump levels> = 2



RET 명령 이후 두 개의 프로그램 레벨이 점프되고 프로그램 실행은 블록 N220에서 계속됩니다.

4.2.2.11 파라미터 지정이 가능한 서브프로그램 리턴 점프 (RETB ...)

일반적으로, 리턴 점프는 RETB 명령을 사용하여 서브프로그램으로부터 호출 프로그램으로 실행됩니다. 그러면 서브프로그램 호출 다음의 프로그램 라인에서 처리가 계속 진행됩니다. 다른 위치에서 프로그램을 계속 처리해야 할 경우, 다음과 같은 옵션이 있습니다.

- ISO 언어 모드에서 스톡 제거 싸이클을 호출한 후 (형상 설명 이후) 프로그램 실행을 재개하는 경우.
- 에러 처리를 위해 서브프로그램 레벨에서 메인 프로그램으로 복귀하는 경우 (ASUB 이후 포함).
- 컴파일 싸이클 및 ISO 언어 모드에서 특정 적용 분야에 대해 여러 프로그램 레벨을 통한 리턴 점프

이를 위해, RETB 명령에 추가 파라미터를 프로그램해야 합니다.

검색 방향

파라미터 <target block>을 지정하면, 리턴 점프가 먼저 호출 블록 이후의 블록으로 실행됩니다. 그런 다음 리턴 점프가 실행된 프로그램의 **시작** 방향으로 대상에 대한 검색이 진행됩니다. 검색이 실패하면 프로그램 끝 방향으로 검색이 다시 진행됩니다.

구문

```
RETB("<target block>")
RETB("<target block>",<block after target block>)
RETB("<target block>",<block after target block> <number of
return jump levels>)
RETB("<target block>", ,<number of return jump levels>)
RETB("<target block>",<block after target block>,<number of
return jump levels>,
<return jump to the beginning of the program>)
RETB( , ,<number of return jump levels>,<return jump to the
beginning of the program>)
```

의미

RETB:	서브프로그램 종료	
<target block>:	파라미터에서 프로그램 실행을 재개해야 하는 블록이 점프 대상으로 지정됩니다. 파라미터 <number of return jump levels>가 프로그래밍되지 않으면 점프 대상이 현재 서브프로그램이 호출되었던 프로그램이 됩니다. 가능한 데이터는 다음과 같습니다.	
	<block number>	대상 블록 번호. 처음에 리턴 점프가 실행되었던 프로그램에서 프로그램의 시작 방향으로 블록 번호에 대한 검색이 실행됩니다.
	<jump marker>	리턴 점프가 실행되는 프로그램에 있어야 하는 점프 마커. 처음에 리턴 점프가 실행되었던 프로그램에서 프로그램의 시작 방향으로 점프 마커에 대한 검색이 실현됩니다.
	<character string>	리턴 점프가 실행되는 프로그램에 있어야 하는 문자열 (예: 프로그램 또는 변수 이름). 처음에 리턴 점프가 실행되었던 프로그램에서 프로그램의 시작 방향으로 문자열에 대한 검색이 실현됩니다. 다음 규칙은 문자열을 프로그래밍할 때 적용됩니다. • **끝에 공란**(끝에 ":"을 붙이는 점프 마커와는 다름). • **문자열 앞**에만 하나의 블록 번호 및/또는 점프 마커를 설정할 수 있으며, **프로그램 명령이 없습니다**.

<block after target block>:	이 파라미터는 프로그램 처리가 <target block> 파라미터에 명시된 블록에서 계속되어야 하는지, 그 다음 블록에서 계속되어야 하는지 지정합니다.		
	유형:	INT	
	값:	0	<target block> 파라미터에 지정된 블록으로 리턴 점프가 실행됩니다.
		> 0	<target block> 파라미터에 지정된 블록 다음 블록으로 리턴 점프가 실행됩니다.
<리턴 점프 레벨 개수>:	이 파라미터는 리턴 점프 후 대상 블록을 검색하고 프로그램 처리를 계속할 프로그램 레벨의 번호를 지정합니다.		
	유형:	INT	
	값:	1	The program is resumed at the 'current program level -1' (just like RET without parameter).
		2	The program is resumed at the 'current program level -2', i.e. one level is skipped.
		3	The program is resumed at the 'current program level -3', i.e. two levels are skipped.
		...	
	값 범위:	1 ... 15	
<프로그램 시작으로 리턴 점프>:	메인 프로그램으로 리턴 점프 시 프로그램이 프로그램 시작 지점에서 계속되어야 하는지를 이 파라미터가 활성 **ISO 모드**에서 지정합니다.		
	유형:	BOOL	
	값:	1	메인 프로그램으로 리턴 점프되고 거기에서 **ISO 언어 모드**가 활성화되면 프로그램은 프로그램의 시작 위치로 분기됩니다.

참고

대상 블록 탐색을 지정하기 위해 문자열을 사용하여 서브프로그램 리턴 점프를 하는 경우 항상 호출 프로그램의 점프 마커부터 탐색을 시작합니다.

문자열을 사용하여 점프 대상을 고유하게 정의해야 하는 경우 문자열이 점프 마커의 이름과 일치하지 않아야 합니다. 일치할 경우 서브프로그램이 항상 문자열이 아닌 점프 마커로 리턴 점프됩니다(예 2 참조).

제한 사항

여러 프로그램 레벨에 걸친 리턴 점프

여러 프로그램 레벨을 통해 리턴 점프하는 경우 개별 프로그램 레벨에 대해 SAVE 명령이 평가됩니다.

여러 프로그램 레벨에 걸쳐 리턴 점프를 하는 경우 모달 서브프로그램이 활성 상태이고 건너뛴 프로그램 중 하나에서 선택 취소 명령 MCALL이 모달 서브프로그램에 프로그래밍되면 모달 서브프로그램은 활성 상태를 유지합니다.

유의사항
프로그래밍 에러 여러 프로그램 레벨을 통과하는 리턴 점프의 경우 필요한 모달 설정에 따라 처리가 계속되도록 하는 것은 사용자의 책임입니다. 이 작업은 적절한 메인 블록을 프로그래밍하여 수행할 수 있습니다.

외부 소스에서 처리

"외부 소스에서 실행" 기능을 통해 외부 프로그램 메모리에서 메인 프로그램과 서브 프로그램을 실행할 경우 RETB(...)로 프로그램한 리턴 대상이 리로드 메모리 안에 있어야 합니다. 그렇지 않으면 점프 대상을 찾을 수 없어 프로그램이 중단되고 알람 14000이 출력됩니다.

참고

프로그램된 점프 명령에 대한 제한 없이 외부 프로그램을 실행하려면 "외부 소스에서 실행" 기능 대신 "외부 저장 장치에서 실행(EES)" 옵션을 사용하는 것이 좋습니다.

예

프로그램 코드	설명
EXAMPLE.MPF	
…	

프로그램 코드	설명
N3000 START_CYC(parm1, param2, …)	
N3010 TECH_CYC1(param1, param2, …)	
N3020 TECH_CYC2(param1, param2, …)	
N3030 TECH_CYC3(param1, param2, …)	
N3040 END_CYC(param1, param2, …)	
N3040 **END_CYC**(param1, param2, …)	
N3050 …	
N4500 START_CYC(param11, param12, …)	
N4510 …	
N4590 END_CYC(param11, param12, ..)	
N5000 …	
…	
N6000 M30	

프로그램 코드	설명
PROC END_CYC(…)	; 메인 프로그램에서 호출, 라인 N3040
N10000 …	
N15000 if status == 1	
N15010 RETB("START_CYC")	; 호출 프로그램 EXAMPLE.MPF로 리턴 점프
	; 문자열 "START_CYC" 검색
	; 검색 방향: 프로그램 시작을 향해 역방향
	; 라인 N3000에서 프로그램 처리가 계속됨
N15020 endif	
N15030 if status == 0	
N15040 RET	; 호출 프로그램 EXAMPLE.MPF로 리턴 점프
	; 라인 N3050에서 프로그램 처리가 계속됨
N15050 endif	
N16000 RET("START_CYC")	; 호출 프로그램 EXAMPLE.MPF로 리턴 점프
	; 문자열 "START_CYC" 검색
	; 검색 방향: 프로그램 끝을 향해 정방향
	; 라인 N4500에서 프로그램 처리가 계속됨
N17060 RETB	; 호출 프로그램 EXAMPLE.MPF로 리턴 점프
	; 라인 N3050에서 프로그램 처리가 계속됨
	; 파라미터가 없는 RETB는 RET와 동일함

4.2.3 서브프로그램 호출

4.2.3.1 파라미터 전송 없이 서브프로그램 호출

서브프로그램은 어드레스 L 및 서브프로그램 번호를 사용해 호출하거나 프로그램 이름을 지정해 호출합니다.

메인 프로그램을 서브프로그램으로 호출할 수도 있습니다. 이 경우 메인 프로그램에 설정된 프로그램 종료 M2 또는 M30이 M17 (프로그램 종료 시 호출 프로그램으로 복귀) 로 평가됩니다.

참고

따라서 서브프로그램을 메인 프로그램처럼 시작할 수도 있습니다.

시스템의 탐색 방식:

*_MPF 포함?

*_SPF 포함?

호출할 서브프로그램의 이름이 메인 프로그램의 이름과 동일한 경우 호출을 실행한 메인 프로그램이 다시 호출됩니다. 이는 보통의 경우 바람직하지 않은 현상이므로 서브프로그램 및 메인 프로그램에 고유한 이름을 지정하여 이런 현상을 방지해야 합니다.

참고

파라미터 전송을 요구하지 않는 서브프로그램은 초기화 파일에서 호출할 수도 있습니다.

구문

```
L<number>/<program name>
```

참고

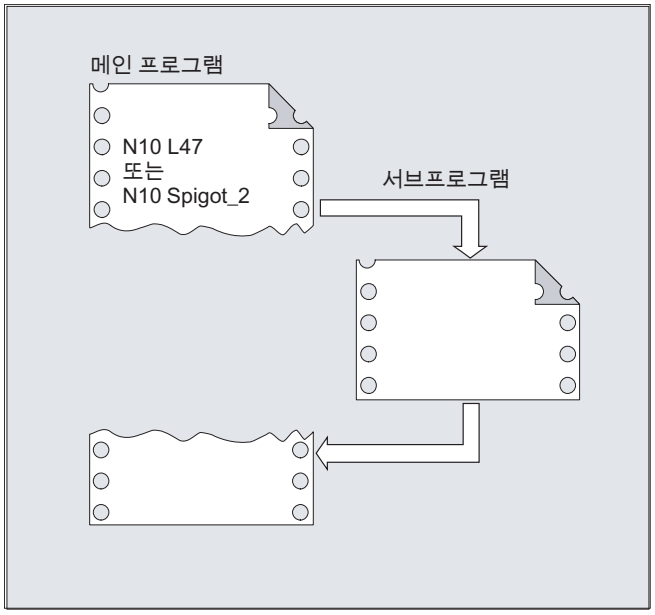
서브프로그램 호출은 항상 개별 NC 블록에 프로그래밍해야 합니다.

의미

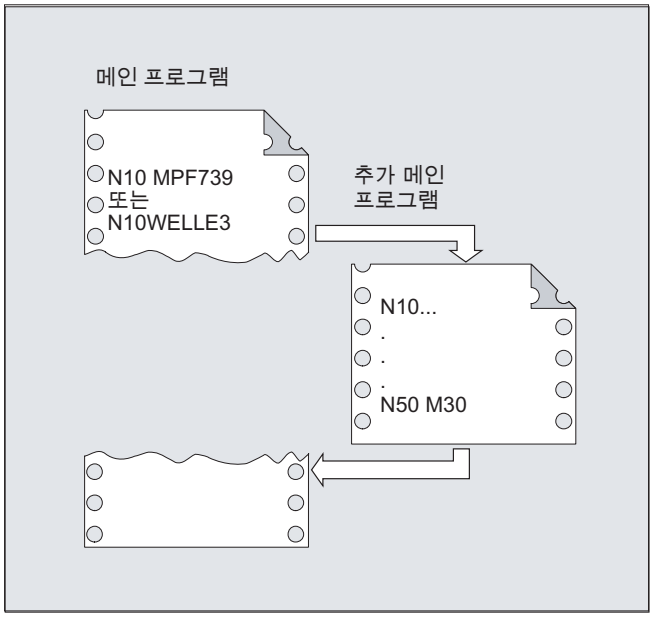
L:	서브프로그램 호출 어드레스	
<number>:	서브프로그램 이름	
	유형:	INT
	값:	소수점 최대 7개 **주의:** 선행 0은 이름에서 중요합니다 (⇒ L123, L0123 및 L00123은 3개의 서로 다른 서브프로그램).
<program name>:	서브프로그램 이름 (또는 메인 프로그램 이름)	

예

예 1: 파라미터 전송 없이 서브프로그램 호출



예 2: 메인 프로그램을 서브프로그램으로 호출



다음 또한 참고:

파라미터 전송이 없는 서브프로그램 (페이지 556)

4.2.3.2 파라미터 전송을 포함한 서브프로그램 호출 (EXTERN)

파라미터 전송을 포함한 서브프로그램이 호출되는 경우 변수 또는 값을 직접 전송할 수 있습니다 (VAR 파라미터 제외).

파라미터 전송이 포함된 서브프로그램은 메인 프로그램에 호출하기 전에 (예를 들어 프로그램 시작에) EXTERNAL로 메인 프로그램에 선언해야 합니다. 서브프로그램의 이름 및 변수 유형은 전송되는 순서대로 지정됩니다.

유의사항
불일치 위험 변수 유형 및 전송 순서는 모두 서브프로그램의 PROC로 선언된 정의와 일치해야 합니다. 파라미터 이름은 메인 프로그램과 서브프로그램에서 서로 다를 수 있습니다.

구문

```
EXTERNAL <program name>(<type_Par1>,<type_Par2>,<type_Par3>)
...
<program name>(<value_Par1>,<value_Par2>,<value_Par3>)
```

참고

서브프로그램 호출은 항상 개별 NC 블록에 프로그래밍해야 합니다.

의미

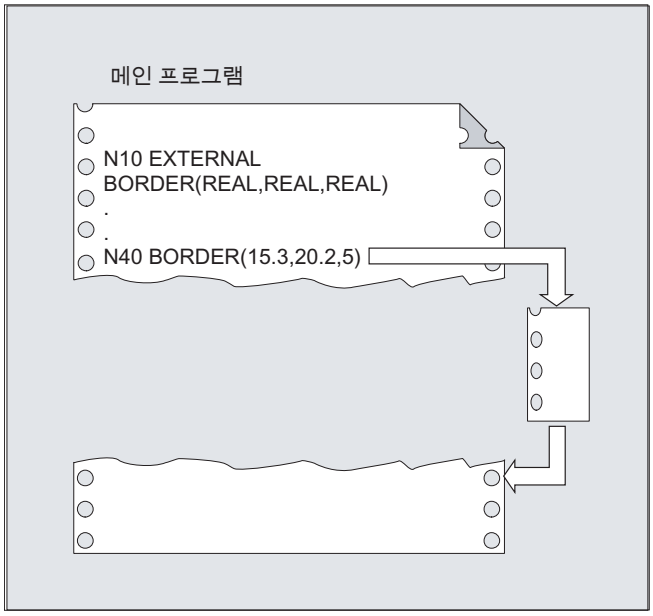
<program name>:	서브프로그램 이름
EXTERNAL:	파라미터 전송을 통해 서브프로그램을 선언할 키워드입니다. **참고:** 서브프로그램이 공작물 또는 전역 서브프로그램 디렉토리에 있는 경우에만 EXTERNAL을 지정해야 합니다. 싸이클은 EXTERNAL로 선언할 필요가 없습니다.

<type_par1>,<type_par2>,<type_par3>:	전송 순서대로 전송할 파라미터의 변수 유형
<value_par1>,<value_par2>,<value_par3>:	전송할 파라미터의 변수 값

예

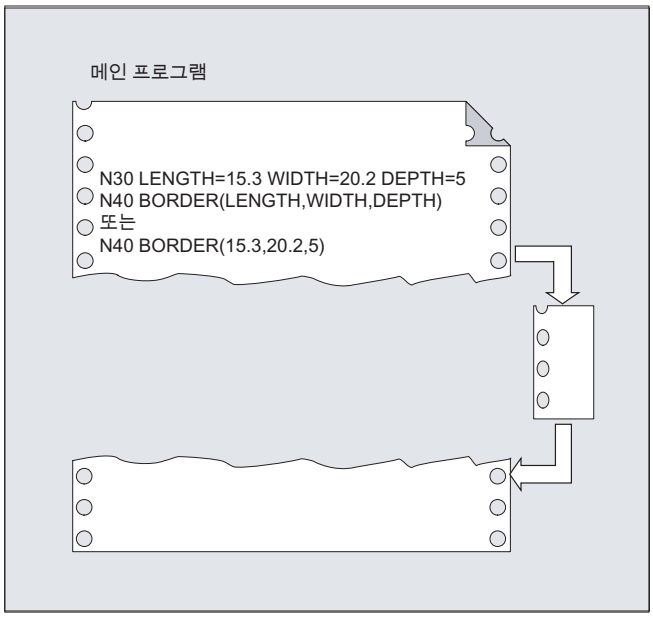
예 1: 선언 후 서브프로그램 호출

프로그램 코드	코멘트
N10 EXTERNAL BORDERS(REAL,REAL,REAL)	; 서브프로그램을 지정합니다.
...	
N40 BORDER(15.3,20.2,5)	; 파라미터 전송을 통해 서브프로그램을 호출합니다.



예 2: 선언 없이 서브프로그램 호출

프로그램 코드	코멘트
N10 DEF REAL LENGTH, WIDTH, DEPTH	
N20…	
N30 LENGTH=15.3 WIDTH=20.2 DEPTH=5	
N40 BORDER(LENGTH,WIDTH,DEPTH)	; 또는 N40 BORDER(15.3,20.2,5)



다음 또한 참고:

값별 호출 파라미터 전송을 포함한 서브프로그램 (PROC) (페이지 557)

기준별 호출 파라미터 전송을 포함한 서브프로그램 (PROC, VAR) (페이지 559)

4.2.3.3 프로그램 반복 횟수 (P)

서브프로그램을 연속적으로 여러 번 실행해야 하는 경우 서브프로그램 호출을 통해 블록의 어드레스 P에서 원하는 프로그램 반복 횟수를 입력할 수 있습니다.

⚠ 주의

프로그램 반복 및 파라미터 전송을 통한 서브프로그램 호출

예를 들어 파라미터는 첫 번째 실행에서와 같이 프로그램을 호출하는 경우에만 전송됩니다. 나머지 반복 동안 파라미터는 변경되지 않습니다. 프로그램을 반복하는 동안 파라미터를 반복하려면 서브프로그램에 적절한 조건을 만들어야 합니다.

구문

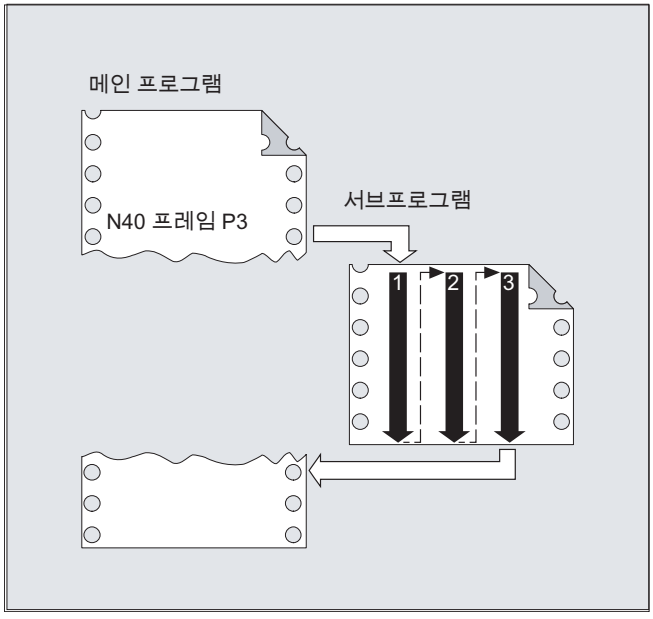
```
<program name> P<value>
```

의미

<program name>:	서브프로그램 호출	
P:	프로그램 반복을 프로그래밍할 어드레스	
<value>:	프로그램 반복 횟수	
	유형:	INT
	값 범위:	1 ... 9999 (부호 없음)

예

프로그램 코드	코멘트
...	
N40 FRAME P3	; BORDER 서브프로그램은 순서대로 3회 실행됩니다.
...	



4.2.3.4 모달 서브프로그램 호출 (MCALL)

모달 서브프로그램 호출 MCALL <program name>을 수행할 경우 지정된 서브프로그램이 즉시 호출되지 않습니다. 그 대신, 경로 모션으로 각 블록을 이송한 후에 가공 프로그램에서 호출이 실행됩니다. 이는 또한 여러 프로그램 레벨에 걸쳐 실행됩니다.

참고

프로그램이 실행될 때 각 경우에서 오직 마지막 모달 서브프로그램 호출 MCALL <program name>만이 적용됩니다. 현재의 모달 서브프로그램 호출이 그 때까지 활성화되어 있었던 모달 서브프로그램 호출을 대체합니다.

파라미터를 서브프로그램으로 전송할 경우 파라미터는 호출

MCALL(<ProgName>(Par1, Par2, ...))을 통해서만 전송됩니다.

참고

서브프로그램 시작 시 데이터 정의 영역은 프로그래밍된 MCALL 호출로 블록을 실행할 때만 한 번 실행됩니다. 다음 서브프로그램 호출의 경우 이송 블록 다음에 데이터 정의 영역이 더 이상 실행되지 않습니다. 즉, 첫 번째 호출 이후에 로컬 사용자 변수(LUD)의 정의에 지정된 값을 더 이상 사용할 수 없지만, 싸이클에서 마지막으로 기록된 값은 사용할 수 있습니다. 첫 번째 호출 이후에 변수는 다시 생성되지 않지만 이전 호출의 마지막 값은 유지됩니다. 이러한 반응을 해결하려면 LUD 정의와 값 지정을 서로 분리하고 값 지정을 데이터 정의 영역 다음의 전용 블록에 프로그래밍하는 것이 좋습니다.

유의사항
경로 모션 없이 모달 서브프로그램 호출
다음 경우에도 모달 서브프로그램은 경로 모션 프로그래밍 없이 호출됩니다: • G0 또는 G1이 활성화된 경우 주소 S 또는 F를 프로그래밍할 때 • 블록에서 G0 또는 G1 을 단독으로 프로그래밍하거나 추가 G 명령과 함께 프로그래밍할 때.

구문

```
MCALL <program name>
...
MCALL
```

의미

MCALL <program name>:	"모달 서브프로그램 호출" 기능 활성화
<program name>:	서브프로그램 이름
MCALL:	프로그램 이름을 지정하지 않고 MCALL을 사용하여 "모달 서브프로그램 호출" 기능을 비활성화합니다.

한계 조건

ASUB

가공 프로그램 처리가 ASUB에 의해 인터럽트되면 ("인터럽트 루틴 (ASUB) (페이지 606)" 장 참조), 이 ASUB에서는 어떤 모달 서브프로그램 호출도 실행되지 않습니다.

ASUB를 "Reset" 채널 상태에서 시작할 경우, 모달 서브프로그램 호출과 관련하여 정상적인 가공 프로그램과 똑같이 동작합니다.

공구 교환 싸이클

공구 교환 싸이클 시 "모달 서브프로그램 호출" 기능을 선택 해제하면, 블록 검색 이후에도 공구 교환 싸이클이 검색 ASUB를 통해, 또는 수동으로 오버스토어(overstore)를 통해 암묵적으로 호출됩니다. 이 경우, "모달 서브프로그램 호출" 기능의 선택을 해제하면 안 됩니다. 해제할 경우 검색 결과가 왜곡됩니다. 그러므로 공구 교환 싸이클의 "모달 서브프로그램 호출" 기능 선택 해제를 다음과 같이 프로그래밍할 것을 권장합니다.

프로그램 코드	설명
...	
IF $AC_ASUB == 0	; 검색 ASUB 또는 오버스토어(overstore)를 통해 호출이 수행되지 않습니다.
MCALL	; "모달 서브프로그램 호출" 기능을 비활성화합니다.
ENDIF	
...	

예

예 1

프로그램 코드	설명
N10 G0 X0 Y0	
N20 MCALL L70	; L70에 대한 모달 서브프로그램 호출을 활성화합니다.
N30 X10 Y10	; X10 Y10에 접근한 후, L70을 호출합니다.

프로그램 코드	설명
N40 X20 Y20	; X20 Y20에 접근한 후, L70을 호출합니다.
...	
N100 MCALL	; "모달 서브프로그램 호출" 기능을 비활성화합니다.
N110 X0 Y0	; X0 Y0에 접근하고, L70을 호출하지 **않습니다**.

예 2

프로그램 코드
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

이 예제에서 프로그래밍된 경로 축을 가진 다음 NC 블록이 서브프로그램 L80에 위치합니다. L70은 L80에 의해 호출됩니다.

4.2.3.5 간접 서브프로그램 호출 (CALL)

프로그램의 특정 지점에서 사용 중인 조건에 따라 서로 다른 서브프로그램을 호출할 수 있습니다. 서브프로그램의 이름은 STRING 유형의 변수에 저장됩니다. 서브프로그램 호출은 CALL 및 변수 이름을 통해 실행됩니다.

참고

간접 서브프로그램 호출은 파라미터 전송 없는 서브프로그램의 경우에만 가능합니다. 직접 서브프로그램 호출의 경우 STRING 상수에 이름을 저장합니다.

구문

```
CALL <program name>
```

의미

CALL:	간접 서브프로그램 호출을 위한 명령어.	
<program name>:	서브프로그램 이름 (변수 또는 상수)	
	유형:	STRING

예

STRING 상수로 직접 호출:

프로그램 코드	코멘트
…	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PART1_SPF"	; CALL을 통해 서브프로그램 PART1 직접 호출.
…	

변수를 통한 간접 호출:

프로그램 코드	코멘트
…	
DEF STRING[100] PROGNAME	: 변수를 정의합니다.
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PART1_SPF"	; 서브프로그램 PART1을 PROGNAME 변수에 지정합니다.
CALL PROGNAME	; CALL 및 PROGNAME 변수를 통해 서브프로그램 PART1을 간접 호출합니다.
…	

4.2.3.6 호출 프로그램 일부를 지정하여 서브프로그램 간접 호출 (CALL BLOCK ... TO ...)

CALL 및 키워드 조합 BLOCK ... TO를 사용하여 서브프로그램을 간접 호출하고 시작 라벨 및 종료 라벨로 지정한 프로그램 섹션을 실행할 수 있습니다.

구문

```
CALL <program name> BLOCK <start label> TO <end label>
CALL BLOCK <start label> TO <end label>
```

의미

CALL:	간접 서브프로그램 호출을 위한 명령어.	
<program name>:	실행할 프로그램 섹션이 포함된 서브프로그램 이름 (변수 또는 상수) (**지정**은 **옵션**).	
	유형:	STRING
	참고: <program name>이 프로그래밍되지 않은 경우 현재 프로그램에서 <start label> 및 <end label>로 지정한 프로그램 섹션을 검색해 실행합니다.	

BLOCK ... TO ... :	프로그램 섹션 간접 실행을 위한 키워드 조합	
<start label>:	실행할 프로그램 섹션의 시작을 의미하는 변수.	
	유형:	STRING
<end label>:	실행할 프로그램 섹션의 종료를 의미하는 변수.	
	유형:	STRING

예

메인 프로그램:

프로그램 코드	코멘트
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; 시작 라벨 및 종료 라벨의 변수 정의.
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	; 간접 서브프로그램 호출 및 호출 프로그램 섹션을 표시하는 식별자.
...	

서브프로그램:

프로그램 코드	코멘트
PROC CONTUR_1 ...	
LABEL_1	; 시작 라벨: 프로그램 섹션 실행 시작.
N1000 G1 ...	
...	
LABEL_2	; 종료 라벨: 프로그램 섹션 실행 종료.
...	

4.2.3.7 ISO 언어로 프로그래밍된 프로그램의 간접 호출 (ISOCALL)

ISO 언어로 프로그래밍된 프로그램은 간접 프로그램 호출 ISOCALL을 사용하여 호출할 수 있습니다. 머신 데이터에서 설정된 ISO 모드가 활성화됩니다. 원래 실행 모드는 프로그램 종료 시 다시 적용됩니다. ISO 모드를 머신 데이터에서 설정하지 않으면 서브프로그램이 Siemens 모드에서 호출됩니다.

ISO 모드에 대한 **추가 정보**: 기능 매뉴얼 ISO 언어 모드

구문

```
ISOCALL <program_name>
```

의미

ISOCALL:	머신 데이터에서 설정된 ISO 모드가 활성화되는 간접 서브프로그램 호출 키워드.
<program name>:	ISO 언어로 프로그래밍된 프로그램 이름 (변수 또는 상수, 유형 STRING)

예: ISO 모드에서 프로그래밍된 싸이클을 통한 형상 호출

프로그램 코드	설명
0122_SPF	; ISO 모드에서 형상 설명
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME = "0122"	; Siemens 가공 프로그램 (싸이클)
...	
N2000 R11 = $AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	; ISO 모드에서 프로그램 0122.spf 실행
...	
N2400 M30	

4.2.3.8 경로 지정 및 파라미터를 통한 서브프로그램 호출 (PCALL)

PCALL을 사용하여 절대 경로 및 파라미터 전송을 통해 서브프로그램을 호출할 수 있습니다.

구문

```
PCALL <path/program name>(<parameter 1>,…,<parameter n>)
```

의미

PCALL:	절대 경로 이름을 통한 서브프로그램 호출을 위한 키워드
<path/program name>:	서브프로그램 이름을 포함한 절대 경로 데이터. 경로 데이터에 관한 규칙은 "프로그램 메모리 파일 주소 지정 (페이지 623)" 참조. 절대 경로 이름을 지정하지 않을 경우 PCALL은 프로그램 식별자를 통한 표준 서브프로그램 호출과 유사하게 동작합니다. 접두어와 파일 식별자 없이 프로그램 이름이 지정됩니다. 프로그램 이름을 접두어와 파일 식별자와 함께 프로그래밍할 경우, EXTERN 명령을 사용하여 프로그램 이름을 접두어 및 파일 식별자와 함께 명시적으로 선언해야 합니다.
<parameter 1>, ...:	서브프로그램의 PROC 연산 결과를 따르는 실제 파라미터.

예

프로그램 코드
PCALL/_N_WKS_DIR/_N_SHAFT_WPD/SHAFT(parameter1,parameter2,…)

4.2.3.9 서브프로그램 호출에 대한 확장 검색 경로 (CALLPATH)

서브프로그램 호출에 대한 검색 경로는 CALLPATH 명령을 사용하여 확장할 수 있습니다. 따라서 서브프로그램의 전체 절대 경로 이름을 지정할 필요 없이 선택하지 않은 공작물 디렉토리에서도 서브프로그램을 호출할 수 있습니다.

외부 프로그램 메모리에 있는 다른 디렉토리를 글로벌 서브루틴 저장에 사용하는 경우, EES 모드 "GDIR 없는 EES"에서 또 다른 적용 옵션이 제공됩니다. 이 경우, CALLPATH를 사용하면 이 서브프로그램 디렉토리로 검색 경로를 확장할 수 있습니다.

검색 경로 확장은 사용자 싸이클 항목 앞에 적용됩니다 (_N_CUS_DIR).

다음 이벤트가 발생할 경우 검색 경로 확장이 다시 선택 취소됩니다.

- 공란을 포함한 CALLPATH
- 파라미터가 없는 CALLPATH
- 가공 프로그램의 끝
- 리셋

구문

```
CALLPATH("<path name>")
```

의미

CALLPATH:	프로그램 가능한 검색 경로 확장 키워드. 별도의 가공 프로그램 라인에 프로그램됩니다.
<path name>:	상수 또는 STRING 유형의 변수. 검색 경로를 확장하기 위한 디렉토리의 절대 경로 이름이 포함되어 있습니다. 경로 데이터에 관한 규칙은 "프로그램 메모리 파일 주소 지정 (페이지 623)" 참조.

예

특정 공작물 디렉토리로 검색 경로를 확장해야 할 경우:

프로그램 코드

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

다음 검색 경로가 설정됩니다 (위치 5.는 새로운 경로):

1. 실제 디렉토리/*이름*
2. 실제 디렉토리/*이름_SPF*
3. 실제 디렉토리/*이름_MPF*
4. //NC:/_N_SPF_DIR / *이름_SPF*
5. **/_N_WKS_DIR/_N_MYWPD_WPD/이름_SPF**
6. /N_CUS_DIR/*이름_SPF*
7. /_N_CMA_DIR/*name_SPF*
8. /_N_CST_DIR/*name_SPF*

보충 조건

- CALLPATH는 프로그래밍된 경로 이름이 실제 존재하는지 여부를 확인합니다. 에러가 발생하면 수정 블록 알람 14009와 함께 가공 프로그램 실행이 중단됩니다.
- CALLPATH는 INI 파일에 프로그래밍할 수도 있습니다. INI 파일을 처리하는 데 걸리는 시간 동안에만 유효합니다 (WPD-INI 파일 또는 NC 활성 데이터의 초기화 프로그램, 예를 들어 첫 번째 채널 _N_CH1_UFR_INI의 프레임). 검색 경로가 다시 리셋됩니다.

4.2.3.10 외부 서브루틴 실행(EXTCALL)

EXTCALL 명령을 사용하여 외부 메모리에서 가공 프로그램을 로드하여 실행할 수 있습니다.

사용 가능한 외부 메모리는 다음과 같습니다.

- 로컬 드라이브
- 네트워크 드라이브
- USB 드라이브

참고

화면 조작반 전면 또는 TCU의 USB 인터페이스만 외부 프로그램을 USB 드라이브에서 처리하기 위한 인터페이스로 사용할 수 있습니다.

유의사항
USB 드라이브에 의한 공구/공작물 손상 외부 서브프로그램을 실행할 때 USB 드라이브는 사용하지 않는 것이 좋습니다. 서브프로그램 실행 중에 접촉 불량, 고장, 트리거로 인한 실행 취소가 발생하는 경우 또는 실수로 드라이브를 제거하는 경우 USB 드라이브와의 통신이 중단되면서 가공이 즉시 중단됩니다. 이로 인해 공구 및/또는 공작물이 손상될 수 있습니다.

외부 프로그램 경로 기본 설정

외부 프로그램 디렉토리의 경로는 설정 데이터를 사용하여 미리 설정할 수 있습니다.

SD42700 $SC_EXT_PROG_PATH

이 설정과 EXTCALL 호출에 지정된 프로그램 경로 및 식별자가 호출할 서브프로그램의 전체 경로를 지정합니다.

참고

프로그램 경로가 EXTCALL 호출을 통해서만 지정되는 경우 SD42700이 반드시 비어 있어야 합니다.

참고

파라미터

외부 프로그램이 호출되면 어떤 파라미터로 여기로 전송될 수 없습니다.

구문

```
EXTCALL("<Path/><Program name>")
```

의미

EXTCALL:	외부 서브프로그램 호출을 위한 명령	
"<Path/><Program name>":	STRING 유형의 상수/변수	
	<Path/>:	절대 또는 상대 경로 지정 (**옵션**)
	<Program name>:	프로그램 이름은 접두어 "_N_" 없이 지정합니다. "_" 또는 "." 문자를 사용하여 프로그램 이름에 파일 확장자 ("MPF", "SPF") 를 추가할 수 있습니다 (**옵션**). 예: "SHAFT" "SHAFT_SPF" "SHAFT.SPF"

경로 지정: 단축 명칭

경로를 지정할 때 다음과 같은 단축 명칭을 사용할 수 있습니다.

- 로컬 드라이브: "**LOCAL_DRIVE:**"
- 메모리 카드: "**CF_CARD:**"
- USB 드라이브 (화면 조작반 전면): "**USB:**"

축약어 CF_CARD: 및 LOCAL_DRIVE:는 서로 번갈아 가며 사용이 가능합니다.

예

로컬 드라이브에서 실행

"MAIN.MPF" 메인 프로그램이 NC 메모리에 저장되고 실행을 위해 선택됩니다.

서브프로그램 "SP_1"

외부 서브프로그램 "SP_1.SPF" 또는 "SP_1.MPF"는 로컬 드라이브의 "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD" 디렉토리에 있습니다.

외부 프로그램 디렉토리의 경로는 다음 설정 데이터로 설정합니다.

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

참고

외부 서브프로그램 호출을 위한 경로 지정:

- 기본 설정을 사용하지 않는 경우: "LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1"
- 기본 설정을 사용하는 경우: "SP_1"

서브프로그램 "SP_2"

외부 서브프로그램 "SP_2.SPF" 또는 "SP_2.MPF"는 USB 드라이브의 WKS.DIR /WST1.WPD 디렉토리에 있습니다. 외부 프로그램 디렉토리의 기본 경로 설정은 서브프로그램 "SP_1"의 경로 지정에 사용되고 메인 프로그램에 다시 작성되지 않습니다. 따라서 서브프로그램 "SP_2"를 호출할 때는 전체 경로를 지정해야 합니다.

메인 프로그램 "MAIN"

프로그램 코드

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

추가 정보

절대 경로 이름을 사용한 EXTCALL 호출

지정된 경로에 서브프로그램이 존재할 경우 EXTCALL 호출로 실행할 수 있습니다. 지정된 경로에 서브프로그램이 존재하지 않는 경우 EXTCALL 호출을 하면 프로그램 실행이 중단됩니다.

상대 경로 이름 또는 경로 이름 없이 EXTCALL 호출

상대 경로 이름 또는 경로 이름 없이 EXTCALL 호출을 실행하면 다음과 같이 사용 가능한 프로그램 메모리를 검색합니다.

1. 경로 이름이 SD42700 $SC_EXT_PROG_PATH에 사전 설정된 경우 EXTCALL 호출에 지정된 데이터 (프로그램 이름 또는 상대 경로 이름) 를 이 경로에서 제일 먼저 검색합니다. 절대 경로는 다음의 문자로 구성됩니다.
   - SD42700 $SC_EXT_PROG_PATH에 설정된 기본 경로 지정
   - 구분자 "/"
   - EXTCALL 명령의 경로 지정 및 서브프로그램 이름
2. 1.에서 서브프로그램을 찾지 못하면 사용자 메모리의 디렉토리를 검색합니다.

서브프로그램을 찾으면 검색이 종료됩니다. 지정한 서브프로그램을 찾지 못하면 EXTCALL 호출 프로그램 실행이 중단됩니다.

조정 가능한 리로드 메모리 (FIFO 버퍼)

외부 서브프로그램을 실행하려면 리로드 메모리가 필요합니다. 리로드 메모리의 크기는 30 KB로 사전 설정되어 있으며, 장비 제조업체만 변경할 수 있습니다(MD18360 MM_EXT_PROG_BUFFER_SIZE 사용).

참고

점프 명령을 이용한 서브프로그램

점프 명령(GOTOF, GOTOB, CASE, FOR, LOOP, WHILE, REPEAT, IF, ELSE, ENDIF 및 기타)을 포함하는 외부 서브프로그램의 경우 점프 대상이 리로드 메모리 안에 있어야 합니다.

이 조건은 프로그램 시작에 대한 점프 명령(GOTOS)과 관련하여 특히 문제가 되는데, 이는 프로그램이 일반적으로 너무 커서 리로드 메모리에 완전히 맞지 않기 때문입니다. 처음 리로드하면 프로그램 시작이 리로드 메모리에서 제거됩니다. 프로그램 시작에 대한 점프 명령을 실행하면 이 기능이 더 이상 점프 대상을 찾을 수 없습니다. 프로그램이 중단되고 알람 14000이 출력됩니다.

프로그램된 점프 명령에 대한 제한 없이 외부 서브프로그램을 실행하려면 "외부 서브프로그램에서 실행(EXTCALL)" 기능 대신 "외부 저장 장치에서 실행(EES)" 기능을 사용하는 것이 좋습니다.

참고

ShopMill/ShopTurn 프로그램

파일 끝에 추가된 형상 설명은 ShopMill 및 ShopTurn 프로그램을 읽기 전용 메모리에 완전히 저장해야 한다는 뜻입니다.

여러 외부 서브프로그램을 동시에 실행하려면 별도의 리로드 메모리가 필요합니다.

리셋 / 프로그램 종료 / 전원 ON

리셋이나 전원 ON을 하면 외부 서브프로그램 호출이 중단되고 관련 로드 메모리가 삭제됩니다.

"외부 실행"으로 선택된 프로그램은 리셋/프로그램 종료 이후에도 "외부 실행" 선택 상태로 유지됩니다. 외부 프로그램 메모리를 계속 사용할 수 있는 한, 동작은 내부에서 선택된 프로그램과 다르지 않습니다.

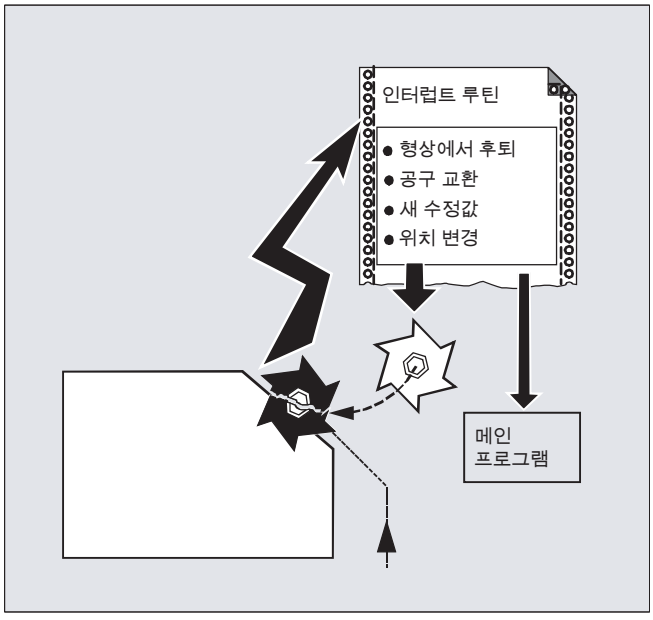
4.3 인터럽트 루틴 (ASUB)

4.3.1 인터럽트 루틴 기능

참고

"비동기 서브프로그램 (ASUB)" 및 "인터럽트 루틴"이란 용어는 여기서 동일한 기능을 설명하는 것이며 서로 번갈아 사용됩니다.

전형적인 예는 인터럽트 루틴의 기능에 대해 명확하게 설명해야 합니다.



가공하는 동안 공구가 파손되었습니다. 이로 인해 현재 가공 프로세스를 중단하는 동시에 인터럽트 루틴이라는 서브프로그램을 시작한다는 신호가 실행됩니다. 인터럽트 루틴에는 이 경우에 실행해야 할 모든 명령문이 포함되어 있습니다.

인터럽트 루틴이 실행을 완료하고 기계가 작동을 계속할 준비를 마치면 제어 시스템이 메인 프로그램으로 다시 이동하여 REPOS 명령에 따라 중단된 지점에서 가공을 계속합니다('형상 리포지셔닝 (페이지 972)' 참고).

⚠ 주의

충돌 위험

REPOS 명령이 서브프로그램에 프로그래밍되지 않은 경우 제어 시스템은 중단된 블록과 이어지는 블록 끝 지점으로 이동합니다.

4.3.2 인터럽트 루틴 생성

인터럽트 루틴을 서브프로그램으로 생성

인터럽트 루틴은 정의 시 서브프로그램으로 식별됩니다.

예:

프로그램 코드	설명
PROC LIFT_Z	; 프로그램 이름 "ABHEB_Z"
N10 ...	; NC 블록이 이어서 나옵니다.
...	
N50 M17	; 마지막으로 프로그램을 종료하고 메인 프로그램으로 돌아갑니다.

모달 G 코드 저장(SAVE)

인터럽트 루틴은 SAVE를 정의하여 지정할 수 있습니다.

SAVE 속성은 인터럽트 루틴을 호출하기 전에 활성 모달 G 코드를 저장하고 인터럽트 루틴을 종료한 후 다시 활성화합니다(' SAVE 메커니즘을 사용하는 서브프로그램(SAVE) (페이지 562) ' 참조).

따라서 인터럽트 루틴을 완료한 후 중단 지점에서 프로세스를 다시 시작할 수 있습니다.

예:

프로그램 코드
PROC LIFT_Z SAVE

프로그램 코드
N10 ... ... N50 M17

추가 인터럽트 루틴 할당 (SETINT)

SETINT 구문을 인터럽트 루틴 내에서 프로그래밍 ('인터럽트 루틴 지정 및 시작 (SETINT)' (페이지 608) 참조) 하고 추가 인터럽트 루틴을 활성화할 수 있습니다. 이 작업은 입력을 통해 실행됩니다.

4.3.3 인터럽트 루틴 지정 및 시작 (SETINT, PRIO, BLSYNC)

제어 시스템은 인터럽트 (1 ... 8) 을 시작하는 몇 가지 빠른 입력 (입력 1 ... 8) 기능을 제공합니다. 각 인터럽트마다 SETINT 명령을 사용하여 우선 순위와 인터럽트 루틴을 할당할 수 있습니다. 빠른 입력을 설정하여 인터럽트가 시작되면, 해당 채널의 처리가 중단되고 인터럽트 루틴이 시작됩니다.

인터럽트 우선 순위

가공 프로그램에서 몇몇 입력에 인터럽트가 할당되면, 각 인터럽트에 서로 다른 우선 순위가 지정되어야 합니다.

1 ... 128 사이의 우선 순위가 인터럽트에 할당될 수 있습니다. 우선 순위 1은 최상위 128은 최하위 우선 순위에 해당합니다.

구문

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

의미

SETINT(<n>):	입력 <n>에 인터럽트 루틴 <Name>이 지정됩니다. 입력 <n> == 1이 감지되자마자 지정된 인터럽트 루틴이 시작됩니다. **참고:** 이미 프로그램된 입력 <n>에 다른 인터럽트 루틴이 지정되면, 이전 지정 값이 더 이상 적용되지 않습니다.	
<n>:	입력 번호	
	유형:	INT
	값 범위:	1 ... 8
PRIO= :	인터럽트의 우선 순위 (옵션)	
<value>:	우선 순위 값 (옵션)	
	유형:	INT
	값 범위:	1 ... 128 (1 ⇒ 가장 높은 우선 순위)
<NAME>:	인터럽트 루틴의 이름 (서브프로그램)	
BLSYNC:	BLSYNC는 인터럽트가 시작된 경우 실제 블록이 완성될 때까지 시스템이 기다리게 합니다. 그 이후에야 인터럽트 루틴이 실행됩니다. (옵션)	
LIFTFAST:	LIFTFAST는 인터럽트가 시작된 경우 먼저 고속 후퇴를 시행합니다 ("형상으로부터 빠른 후퇴 (SETINT LIFTFAST, ALF) (페이지 612)" 장 참조). 그 이후에야 인터럽트 루틴이 실행됩니다. (옵션)	

보충 조건

인터럽트 규칙

1. 즉시 실행될 수 없거나 현재 이미 처리되고 있는 모든 인터럽트에 대해 추가 인터럽트 요청이 저장됩니다. 이 인터럽트에 대한 다른 모든 인터럽트 요청은 사라집니다.
2. 현재 인터럽트가 처리되고 있는데 우선 순위가 더 높은 인터럽트가 추가로 시작되면, 우선 순위가 높은 인터럽트가 우선 순위가 낮은 인터럽트를 중단시킵니다. 우선 순위가 높은 인터럽트가 완료되고 나면 우선 순위가 낮은 인터럽트가 계속 실행됩니다. 우선 순위가 높은 인터럽트가 처리되고 있는 동안에 우선 순위가 더 낮은 인터럽트에 대한 추가 요청이 수신되면, 하나의 요청이 저장됩니다. 그리고 다른 모든 요청들은 사라집니다.
3. 현재 인터럽트가 처리되고 있는데 우선 순위가 더 높은 인터럽트가 추가로 시작되면, 우선 순위가 높은 인터럽트가 우선 순위가 낮은 인터럽트를 중단시킵니다. 우선 순위가 더 높은 인터럽트가 처리됩니다. 우선 순위가 높은 인터럽트가 시작된 경우, 현재 인터럽트는 중단되고 우선 순위가 더 높은 인터럽트가 처리됩니다. 최대 6개의 활성 인터럽트 레벨을 사용할 수 있습니다. 현재 하나의 인터럽트 레벨이 처리 중이라면, 나머지 다섯 개의 인터럽트 레벨은 대기합니다. 각 활성 인터럽트 레벨마다 최대 하나의 추가 인터럽트 요청이 저장됩니다. 다른 모든 인터럽트 요청은 사라집니다. 추가 인터럽트 레벨 (인터럽트 레벨 ≥ 7) 에 대해 요청되는 인터럽트 요청 역시 사라집니다.

예

예 1: 인터럽트 루틴 할당 및 우선 순위 정의

프로그램 코드	코멘트
...	
N20 SETINT(3) PRIO=1 ABHEB_Z	; 입력 3 == 1이면, 인터럽트 루틴 "ABHEB_Z"를 시작함
N30 SETINT(2) PRIO=2 ABHEB_X	; 입력 2 == 1이면, 인터럽트 루틴 "ABHEB_X"를 시작함
...	

입력을 동시에 사용할 수 있는 경우 (동시에 활성화되는 경우) 인터럽트 루틴은 우선 순위 값의 순서에 따라 실행됩니다. 먼저 "ABHEB_Z"가 실행된 다음 "ABHEB_X"가 실행됩니다.

예 2: 새로운 인터럽트 루틴 할당

프로그램 코드	코멘트
...	
N20 SETINT(3) PRIO=2 ABHEB_Z	; 입력 3 == 1이면, 인터럽트 루틴 **"ABHEB_Z"**를 시작함
...	
N80 SETINT(3) PRIO=1 ABHEB_X	; 입력 3 == 1이면, 인터럽트 루틴 **"ABHEB_X"**를 시작함
...	

4.3.4 인터럽트 루틴 할당 비활성화/다시 활성화 (DISABLE, ENABLE)

SETINT 구문은 DISABLE을 통해 비활성화할 수 있고 입력 → 인터럽트 루틴 지정을 유지하면서 ENABLE을 통해 다시 활성화할 수 있습니다.

구문

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

의미

DISABLE(<n>):	명령: 입력 <n>의 인터럽트 루틴 할당 **비활성화**	
ENABLE(<n>):	명령: 입력 <n>의 인터럽트 루틴 할당 **다시 활성화**	
<n>:	파라미터: 인터럽트 신호 번호	
	유형:	INT
	값 범위:	1 ... 32

예

프로그램 코드	코멘트
N20 SETINT(3) PRIO=1 ABHEB_Z	; 입력 3 스위치가 켜지면 인터럽트가 발생합니다.
	; 루틴 "ABHEB_Z"가 시작되어야 합니다.
...	
N90 DISABLE(3)	; N20에서 SETINT 구문이 비활성화됩니다.
...	
N130 ENABLE(3)	; N20에서 SETINT 구문이 재활성화됩니다.
...	

4.3.5 인터럽트 루틴 할당 삭제 (CLRINT)

NC 프로그램 (ASUP) 을 위해 SETINT로 정의된 인터럽트 신호 할당은 CLRINT를 이용하여 삭제할 수 있습니다.

구문

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

의미

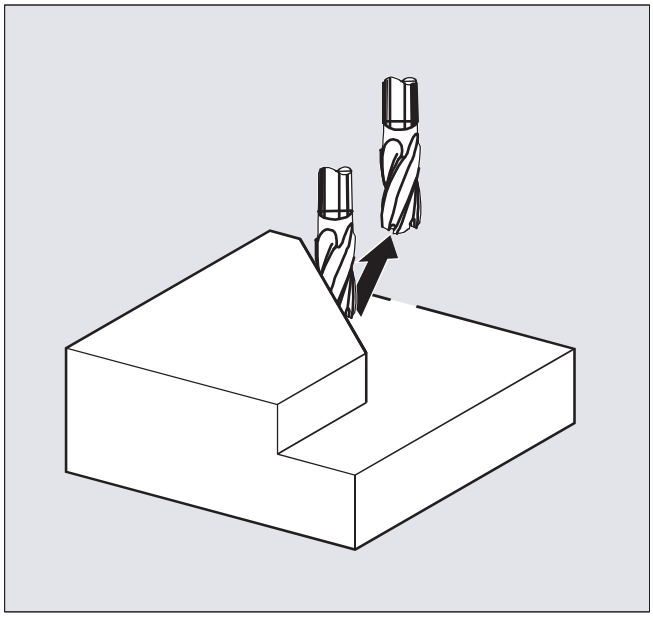
CLRINT(<n>):	명령: SETINT <n>으로 정의된 NC 프로그램 (ASUP)에 대한 인터럽트 신호의 할당을 삭제합니다.	
<n>:	파라미터: 인터럽트 신호 번호	
	유형:	INT
	값 범위:	1 ... 32

예

프로그램 코드	코멘트
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	; 입력 "3"과 인터럽트 루틴 "ABHEB_Z" 간의 지정이 삭제됩니다.

4.3.6 형상으로부터 빠른 후퇴 (SETINT LIFTFAST, ALF)

LIFTFAST가 포함된 SETINT 명령의 경우 입력이 활성화되면 고속 후퇴를 사용하여 공구가 공작물 형상으로부터 빠르게 후퇴합니다.



추가 시퀀스는 SETINT 명령에 LIFTFAST 이외의 인터럽트 루틴이 포함되었는지 여부에 따라 다릅니다.

인터럽트 루틴이 포함된 경우: 고속 후퇴 **이후**에 인터럽트 루틴이 실행됩니다.

인터럽트 루틴이 없는 경우: 고속 후퇴 이후 가공이 중단되고 알람이 출력됩니다.

구문

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

의미

SETINT(<n>):	명령: 입력 <n>을 인터럽트 루틴에 할당합니다. 할당된 인터럽트 루틴은 입력 <n>이 ON될 때 시작됩니다.	
<n>:	파라미터: 입력 번호	
	유형:	INT
	값 범위:	1 ... 8
PRIO= :	우선 순위 정의	
<value>:	우선 순위 값	
	값 범위:	1 ... 128
	우선 순위 1은 최상위 우선 순위에 해당합니다.	
<NAME>:	실행할 서브프로그램 (인터럽트 루틴) 의 이름.	
LIFTFAST:	명령: 형상에서 고속 후퇴	
ALF=… :	명령: 프로그램 가능한 이송 방향 (이동 블록) ALF 프로그래밍에 대한 자세한 설명은 "형상에서 고속 후퇴 시 이송 방향 (페이지 615)"을 참조하십시오.	

보충 조건

미러링을 통한 활성 프레임 동작

후퇴 방향을 결정할 때 미러를 가진 프레임이 활성 상태인지 여부를 점검합니다. 이 경우 후퇴 방향에 대해, 오른쪽 및 왼쪽이 서로 교체됩니다. 공구 방향의 방향 구성 요소는 미러링되지 않습니다. 이 동작은 MD 설정으로 활성화합니다.

MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

예

파손된 공구는 자동으로 대체 공구로 교체되어야 합니다. 그런 다음 새 공구로 가공이 계속됩니다.

메인 프로그램:

메인 프로그램	코멘트
N10 SETINT(1) PRIO=1 W_WECHS LIFTFAST	; 입력 1이 ON되면 공구는 형상에서 고속 후퇴로 즉시 후퇴합니다 (공구 반경 보정 G41의 경우 코드 번호 7). 그런 다음 인터럽트 루틴 "W_WECHS"가 실행됩니다.
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	

서브프로그램:

서브프로그램	코멘트
PROC W_CHANGE SAVE	; 실제 작동 상태가 저장되는 서브프로그램
N10 G0 Z100 M5	; 공구 교환 위치, 스핀들 정지
N20 T11 M6 D1 G41	; 공구 교환
N30 REPOSL RMBBL M3	; 형상에 리포지셔닝하고 메인 프로그램으로 리턴 점프 (이 작업은 한 블록에 프로그래밍)

4.3.7 형상에서 고속 후퇴 시 이송 방향

후퇴 이동

다음 G 코드는 후퇴 이동 평면을 정의합니다.

- LFTXT
  후퇴 이동 평면은 경로 접선 및 공구 방향에 의해 정의됩니다 (기본 설정).
- LFWP
  후퇴 이동 평면은 G 코드 G17, G18 또는 G19를 사용하여 선택된 활성 작업 평면입니다. 후퇴 이동 방향은 접선 경로에 종속되지 않습니다. 따라서 축과 평행한 방향으로 고속 후퇴하도록 프로그래밍할 수 있습니다.
- LFPOS
  POLF로 프로그래밍한 절대 축 위치로 POLFMASK/POLFMLIN를 이용해 축의 후퇴를 선언합니다.
  ALF는 직선 시스템 내에 여러 축이 있는 경우 축의 후퇴 방향에 영향을 주지 않습니다.

프로그램 가능한 이송 방향 (ALF=...)

방향은 후퇴 이동 평면에서 ALF를 통해 45도 단위씩 프로그래밍됩니다.

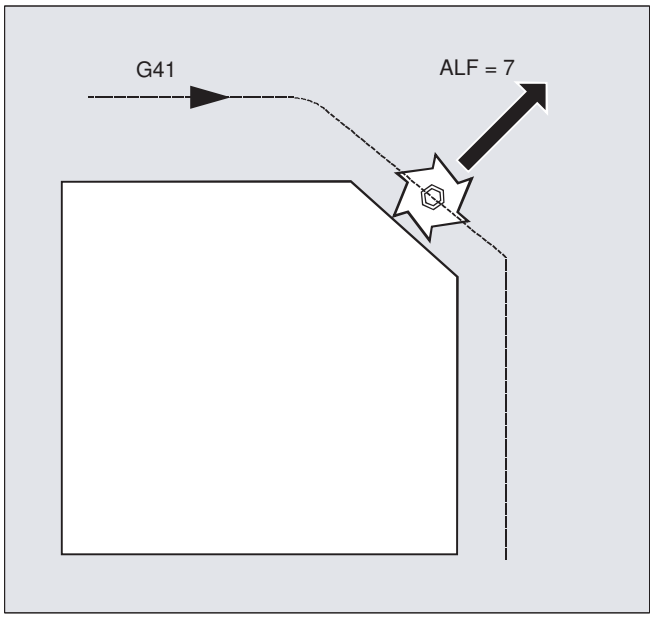
가능한 이송 방향은 제어 시스템의 특수 코드 번호로 저장되며 이 번호를 사용하여 호출할 수 있습니다.

예:

프로그램 코드

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

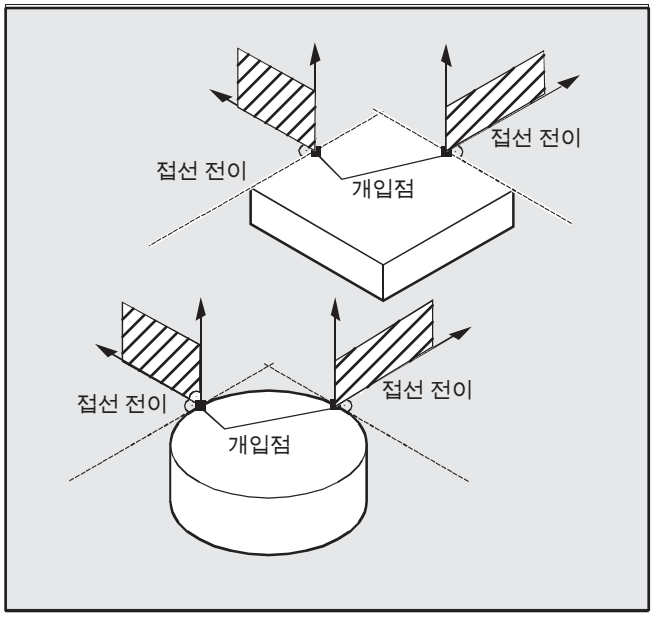
G41이 활성화된 상태에서 (가공 방향이 형상 왼쪽 방향) 공구는 형상으로부터 수직 방향으로 후퇴합니다.



LFTXT의 이송 방향을 정의하기 위한 기준 평면

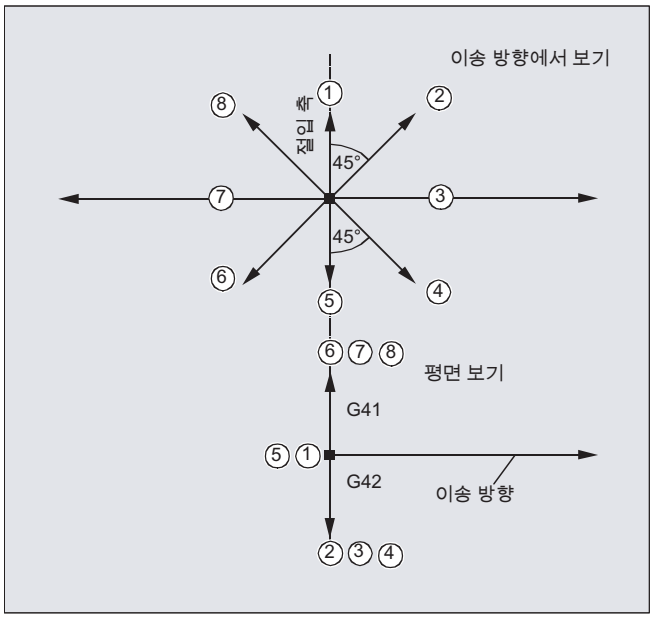
공구가 프로그래밍된 형상에 적용하는 시점에서 공구는 해당 코드 번호를 통해 후퇴 이동을 지정하기 위한 기준으로 사용되는 평면에 클램프됩니다.

기준 평면은 수직 공구 축 (절입 방향) 및 이 축과 수직 위치에 있는 벡터로부터 파생되며 공구 적용 지점의 접선 방향과 수직을 이룹니다.



LFTXT의 이송 방향이 적용된 코드 번호

기준 평면부터 시작하여 다음 그림에서 이송 방향을 가진 코드 번호를 찾습니다.



공구 후퇴 방향은 ALF=1에 의해 정의됩니다.

"빠른 후퇴" 동작은 ALF=0에 의해 비활성화됩니다.

⚠ 주의

충돌 위험

공구 반경 보정이 활성화될 경우:

- G41 코드 2, 3, 4
- G42 코드 6, 7, 8

이런 경우 공구가 형상으로 이동하여 공작물과 충돌하기 때문에 사용하지 **않아야 합니다**.

LFWP의 이송 방향이 적용된 코드 번호

LFWP의 경우 가공 평면의 방향은 다음 할당에서 도출됩니다.

- G17: X/Y 평면
  ALF=1: X 방향으로 후퇴
  ALF=3: Y 방향으로 후퇴
- G18: Z/X 평면
  ALF=1: Z 방향으로 후퇴
  ALF=3: X 방향으로 후퇴
- G19: Y/Z 평면
  ALF=1: Y 방향으로 후퇴
  ALF=3: Z 방향으로 후퇴

다음 또한 참고:

나사 절삭 중 고속 후퇴 (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN) (페이지 256)

4.3.8 인터럽트 루틴의 이동 순서

LIFTFAST 없이 인터럽트 루틴

축 이동은 경로 아래를 따라 제동되어 정지합니다(0 속도). 그런 다음 인터럽트 루틴이 시작됩니다.

정지 위치는 인터럽트 위치로 저장되고 인터럽트 루틴 종료 시 RMIBL을 사용하여 REPOS를 위해 접근합니다.

LIFTFAST에 의한 인터럽트 루틴

축 이동은 경로를 따라 제동됩니다. LIFTFAST 이동은 중복된 이동으로 동시에 실행됩니다. 경로 이동 및 LIFTFAST 이동이 정지하면(0 속도) 인터럽트 루틴이 시작됩니다.

형상 위치는 LIFTFAST 이동이 시작되고 그에 따라 경로만 남겨진 인터럽트 위치로 저장됩니다.

LIFTFAST 및 ALF=0에 의한 인터럽트 루틴은 LIFTFAST 없는 인터럽트 루틴과 동일한 방법으로 정밀하게 동작합니다.

참고

형상으로부터 신속하게 후퇴할 때 형상 축을 이동하는 절대값은 가공 데이터를 사용하여 설정할 수 있습니다.

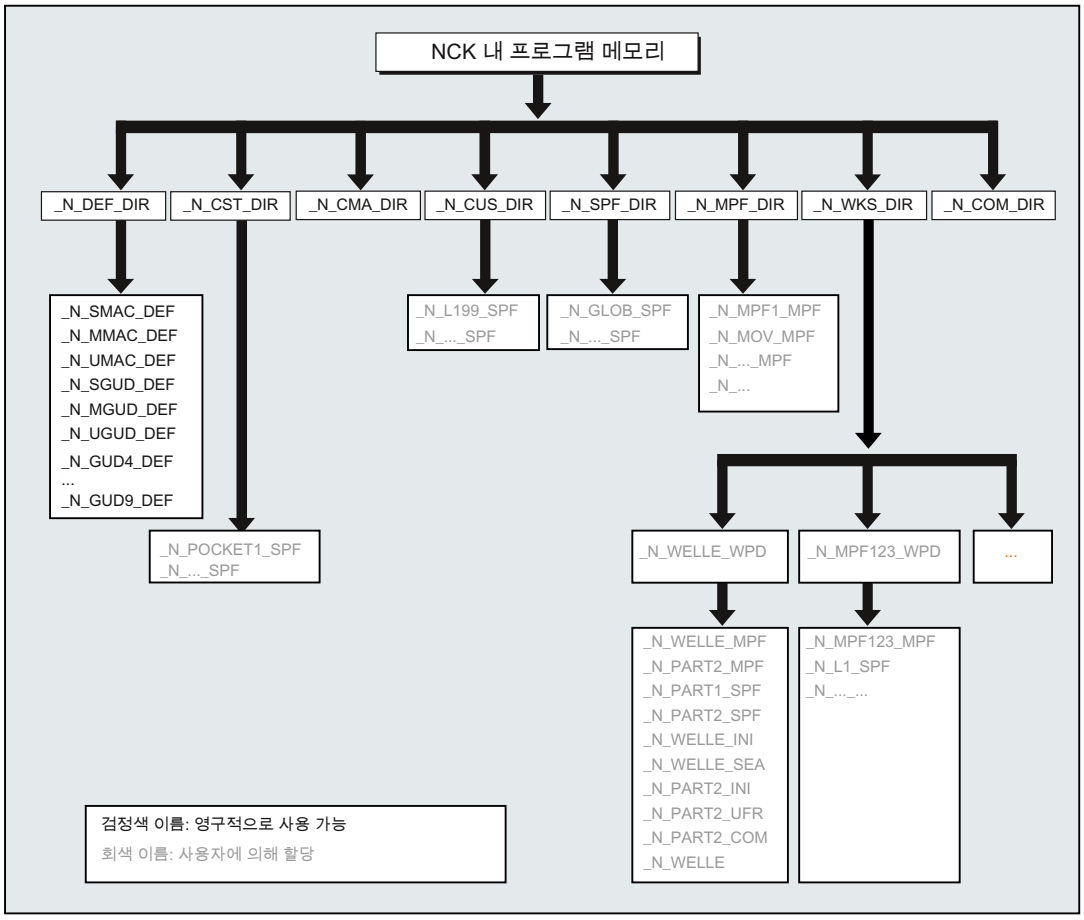
4.4 파일 및 프로그램 관리

4.4.1 프로그램 메모리

4.4.1.1 NCK 내 프로그램 메모리

파일 및 프로그램 (예: 메인 프로그램 및 서브프로그램, 매크로 정의) 은 비활성 프로그램 메모리에 저장됩니다 (→ 패시브 파일 시스템).

또한 다양한 파일 유형을 일시적으로 이 메모리에 저장하고 필요에 따라 작업 메모리로 전송할 수 있습니다 (예: 특정 공작물 가공 시 초기화를 위해 전송).



표준 디렉토리

다음과 같은 표준 디렉토리를 사용할 수 있습니다.

디렉토리	내용
_N_DEF_DIR	데이터 모듈 및 매크로 모듈
_N_CST_DIR	표준 싸이클
_N_CMA_DIR	제조업체 싸이클
_N_CUS_DIR	사용자 싸이클
_N_WKS_DIR	공작물
_N_SPF_DIR	글로벌 서브프로그램
_N_MPF_DIR	메인 프로그램
_N_COM_DIR	설명

파일 유형

메인 메모리에 다음과 같은 파일 유형을 저장할 수 있습니다:

파일 유형	설명
<name>_MPF	메인 프로그램
<name>_SPF	서브프로그램
<name>_TEA	머신 데이터
<name>_SEA	설정 데이터
<name>_TOA	공구 옵셋
<name>_UFR	워크 옵셋/FRAME
<name>_INI	초기화 파일
<name>_GUD	전역 사용자 데이터
<name>_RPA	R 파라미터
<name>_COM	설명
<name>_DEF	글로벌 사용자 데이터 및 매크로 정의

공작물 메인 디렉토리 (_N_WKS_DIR)

공작물 메인 디렉토리의 이름은 프로그램 메모리의 표준 설정에서 _N_WKS_DIR입니다. 공작물 메인 디렉토리에는 프로그래밍된 공작물의 모든 공작물 디렉토리가 포함되어 있습니다.

공작물 디렉토리 (..._WPD)

공작물 디렉토리에는 공작물을 가공하는 데 필요한 모든 파일이 포함되어 있습니다. 이러한 파일로 메인 프로그램, 서브프로그램, 초기화 프로그램 및 설명 파일이 포함될 수 있습니다.

처음으로 가공 프로그램을 시작하면 선택한 프로그램에 따라 머신 데이터 MD11280 $MN_WPD_INI_MODE에 의해 초기화 프로그램이 한 번 실행됩니다.

예:

SHAFT 공작물을 위해 생성된 공작물 디렉토리 _N_SHAFT_WPD에 다음 파일이 포함되어 있습니다.

파일	설명
_N_SHAFT_MPF	메인 프로그램
_N_PART2_MPF	메인 프로그램
_N_PART1_SPF	서브프로그램
_N_PART2_SPF	서브프로그램
_N_SHAFT_INI	공작물 데이터에 대한 일반 초기화 프로그램
_N_SHAFT_SEA	데이터 초기화 프로그램 설정
_N_PART2_INI	가공 2 프로그램 데이터에 대한 일반 초기화 프로그램
_N_PART2_UFR	가공 2 프로그램 데이터에 대한 초기화 프로그램
_N_SHAFT_COM	설명 파일

NC가 가공을 위해 직접 요구하지는 않는 공작물 디렉토리에도 데이터를 저장할 수 있습니다. ASCII 파일에 더하여 이 설명 파일은 JPG 형식의 이미지 또는 PDF 형식의 설명과 같은 이진 파일일 수 있습니다. NC가 이러한 파일들을 이진 파일로 해석하도록 하기 위해서는, NC에서 파일 확장자가 반드시 알려져야 합니다(MD17000 $MN_ EXTENSIONS_OF_BIN_FILES을 통한 스타트업 중 설정, 다음 파일 확장자는 기본 설정 내에 존재함). 파일 확장자 JPG, GIF, PNG, BMP, PDF, ICO 및 HTM는 기본 설정에 사전 설정되어 있습니다.)

가공할 공작물 선택

채널에서 실행할 공작물 디렉토리를 선택할 수 있습니다. **동일한 이름**을 가진 메인 프로그램 또는 단일 메인 프로그램(_MPF)을 이 디렉토리에 저장하면 자동으로 실행할 디렉토리가 선택됩니다.

4.4.1.2 외부 프로그램 메모리

NC의 패시브 파일 시스템뿐 아니라 기계에서 외부 프로그램 메모리도 기계에서 사용할 수 있습니다(예: 로컬 드라이브나 네트워크 드라이브 상에서 사용).

"외부로부터 실행" 또는 "EES (외부 저장 장치로부터 실행)" 기능을 사용하여, 외부 프로그램 메모리에서 **직접** 가공 프로그램을 실행할 수 있습니다.

추가 정보: 기능 매뉴얼 기본 기능

글로벌 가공 프로그램 메모리 (GDIR)

드라이브를 선언할 때 드라이브 중 하나를 글로벌 가공 프로그램 메모리 (GDIR) 로 지정할 수 있습니다.

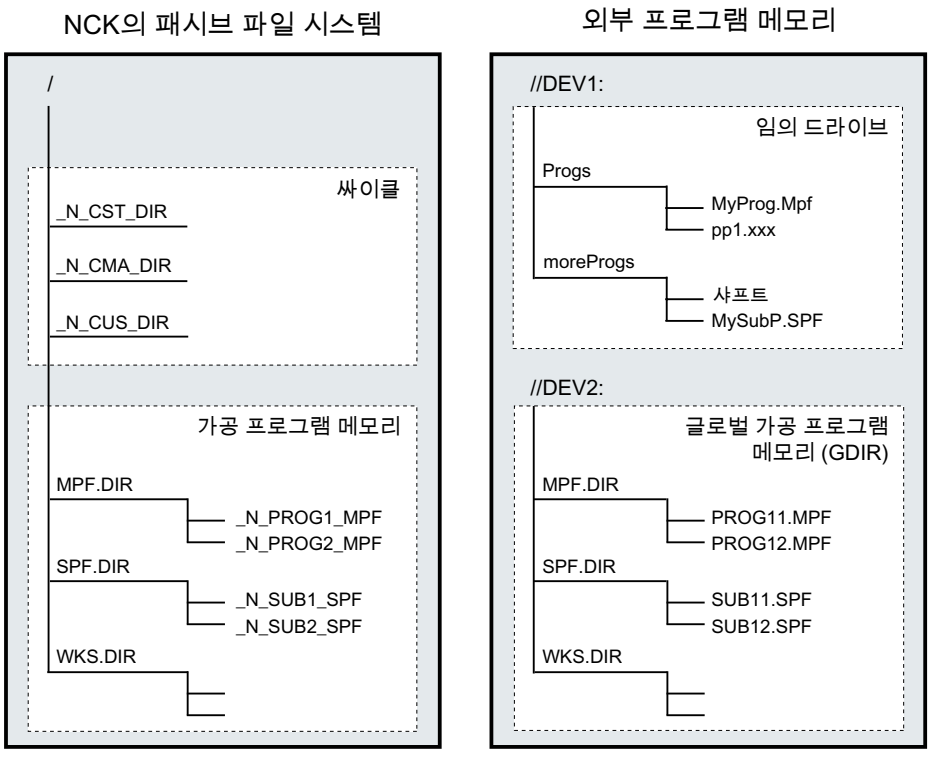
시스템이 자동으로 MPF.DIR, SPF.DIR 및 WKS.DIR 디렉토리를 드라이브에 생성합니다. 이 세 디렉토리가 GDIR이 됩니다.

GDIR은 EES 기능에만 사용됩니다. 드라이브 구성에 따라 GDIR이 NC 가공 프로그램 메모리를 대체하거나 확장합니다. EES 작업을 위해 GDIR을 반드시 생성해야 하는 것은 아닙니다.

GDIR의 디렉토리 및 파일은 패시브 파일 시스템과 같은 방식으로 가공 프로그램에서 지정할 수 있습니다. 따라서 NC 프로그램과 경로 세부 사항이 호환되어 패시브 파일 시스템에서 GDIR로 전송할 수 있습니다. GDIR의 SPF.DIR 디렉토리는 서브프로그램 검색 경로에 포함되어 있습니다.

프로그램 구조

다음 그림은 외부 프로그램 메모리의 프로그램 구조입니다.



대소문자 구분 파일 시스템

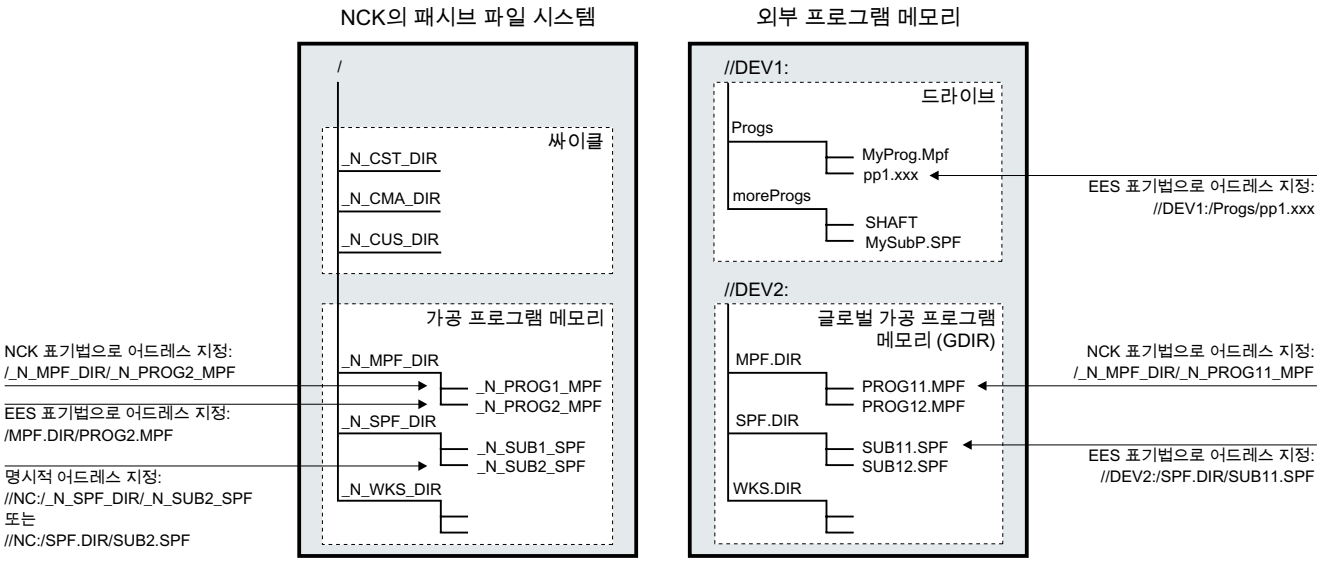
참고

파일 어드레스를 지정 ("프로그램 메모리 파일 주소 지정 (페이지 623)" 참조) 할 때 대소문자로 인한 문제를 방지하기 위해 외부 프로그램 메모리에는 **대소문자 구분** 파일 시스템을 사용해야 합니다.

4.4.1.3 프로그램 메모리 파일 주소 지정

파일 처리 명령 (예: WRITE, DELETE, READ, ISFILE, FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) 을 사용하여 어드레스가 지정되는 프로그램 메모리 내 파일은 파일 이름과 절대 경로를 사용하거나 파일 이름만을 사용하여 참조합니다. 후자의 경우, 선택한 프로그램의 경로가 파일 경로로 사용됩니다.

NC/EES 표기법으로 어드레스 지정



패시브 파일 시스템의 파일 어드레스 지정

패시브 파일 시스템의 파일은 일반적으로 드라이브 이름을 명시하지 않고 **NCK 표기법** (디렉토리와 파일 이름이 도메인 이름 "_N_"으로 시작하고 파일 식별자의 구분자로 "_" 사용)에 따라 어드레스를 지정합니다. 하지만 **EES 표기법** (도메인 이름 "_N_"을 사용하지 않고 디렉토리/파일 확장자의 구분자로 "." 사용) 으로 어드레스를 지정할 수도 있습니다.

예:

- NC 표기법: "/_N_SPF_DIR/_N_SUB1_SPF"
- EES 표기법: "/SPF.DIR/SUB1.SPF"

참고

패시브 파일 시스템의 파일에 대한 EES 표기법에서의 어드레시 지정 계획은 다음 규칙에 따라 내부에서 NC 표기법으로 변환됩니다.

- 디렉토리 및 파일 이름에 도메인 이름 "_N_"이 추가됩니다.
- 디렉토리 또는 파일 이름의 끝에서 네 번째 문자가 마침표 (".") 이면 밑줄 ("_") 로 변환됩니다.

패시브 파일 시스템은 사전 정의된 드라이브 이름 "//NC"을 사용하여 명시적으로 주소가 지정될 수도 있습니다.

예:

- NC 표기법: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- EES 표기법: "//NC:/SPF.DIR/SUB1.SPF"

외부 프로그램 메모리의 파일 어드레스 지정

GDIR로 기록되지 않은 외부 프로그램 메모리의 파일은 EES 표기법에 따라 어드레스를 지정해야 합니다. 드라이브 이름(예: "//DEV1:")은 어드레스 지정 경로의 시작 부분에 명시되어야 합니다. /user/sinumerik/hmi/cfg/logdrive.ini에 설정된 모든 심볼릭 디바이스 이름이 허용됩니다.

예:

- EES 표기법: "//DEV1:/MyProgDir/pp1.xxx"
- NC 표기법: 허용되지 않음

글로벌 가공 프로그램 메모리 (GDIR) 의 파일 어드레스 지정

GDIR 파일의 어드레스를 지정할 때 EES 표기법으로 경로를 지정하는 방법 외에 NC 표기법으로 경로를 지정할 수 있습니다.

예:

- EES 표기법: "//DEV2:/MPF.DIR/PROG11.MPF"
- NC 표기법: "/_N_MPF_DIR/_N_PROG11_MPF"

참고

NC 표기법으로 어드레스가 지정된 GDIR의 파일은 다음 규칙에 따라 내부에서 EES 표기법으로 변환됩니다.

- 디렉토리와 파일 이름에서 도메인 이름 "_N_"이 제거됩니다.
- 디렉토리 또는 파일 이름의 끝에서 네 번째 문자가 밑줄 ("_") 이면 마침표 (".") 로 변환됩니다.

경로 지정 규칙

전체 경로는 드라이브 이름, 디렉토리 경로 및 파일 이름으로 구성됩니다.

드라이브 이름

드라이브 이름 지정에는 다음 규칙이 적용됩니다.

- /user/sinumerik/hmi/cfg/logdrive.ini에 설정된 모든 심볼릭 디바이스 이름이 허용됩니다.
- "//"로 시작하고 그 뒤에 최소 하나의 문자 또는 숫자가 붙습니다.
- 그 다음 문자들은 알파벳, 숫자, "_" 및 공백을 어떤 식으로든 조합할 수 있습니다.
- 이름은 문자나 숫자로 끝나고 그 뒤에 ":"가 붙습니다.
- 그 밖의 특수 문자는 허용되지 않습니다.

참고

드라이브 이름 "//NC:"는 패시브 파일 시스템에 사전 정의된 이름입니다.

예:

- 외부 프로그램 메모리:
  - //Drive1:
  - //Drive_1:
  - //Drive 1:
  - //A B:
  - //1 B C 2:

디렉토리 경로

디렉토리 경로 지정에는 다음 규칙이 적용됩니다.

- 디렉토리 경로의 시작과 끝에 "/"이 있어야 하고 개별 경로 섹션의 구분자로 "/"을 사용합니다.

  참고

  디렉토리 경로에 이중 슬래시 ("//") 를 사용할 수 **없습니다**!

- 디렉토리 이름:
  - 디렉토리 이름은 문자 또는 숫자로 시작해야 합니다. NC 표기법으로 어드레스를 지정하는 경우에만 디렉토리 이름을 도메인 이름 "_N_"으로 시작할 수 있습니다.
  - 그 다음 문자들은 알파벳, 숫자, "_"를 어떤 식으로든 조합할 수 있습니다.

    참고

    외부 프로그램 메모리의 디렉토리 이름에는 공백도 허용됩니다. 단, 외부 프로그램 메모리가 글로벌 가공 프로그램 메모리 (GDIR) 로 생성된 경우는 제외합니다.

  - 그 밖의 특수 문자는 허용되지 않습니다.
- 디렉토리 확장자:
  - 디렉토리 확장자는 3개 문자/숫자로 구성되어야 합니다.
  - 디렉토리 이름은 NC 표기법의 경우 "_", EES 표기법의 경우 "."으로 구분합니다.

  참고

  패시브 파일 시스템의 디렉토리 확장자는 _DIR과 _WPD 뿐입니다.

예:

- 패시브 파일 시스템 또는 GDIR:
  - NC 표기법: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
  - EES 표기법: WKS.DIR/MYPROGS.WPD/...
- 외부 프로그램 메모리:
  - /abc
  - /ab_c.def
  - /ab c1.def
  - /a b c .d11
  - /abc.def/ghi.klm

파일 이름

파일 이름에는 다음 규칙이 적용됩니다.

- NC 표기법으로 어드레스를 지정하는 경우에만 파일 이름을 도메인 이름 "_N_"으로 시작할 수 있습니다.
- 다음 2개의 문자는 반드시 2개의 글자이거나 밑줄 다음에 1개의 글자로 구성해야 합니다.

  참고

  이 조건이 만족되면 프로그램 이름을 지정하여 다른 프로그램에서 NC 프로그램을 서브프로그램으로 호출할 수 있습니다. 하지만 프로그램 이름이 숫자로 시작되면 CALL 명령문을 통해서만 서브프로그램을 호출할 수 있습니다.

- 그 다음 문자들은 알파벳, 숫자, "_"를 어떤 식으로든 조합할 수 있습니다.
- 파일 확장자:
  - 파일 확장자는 3개 문자/숫자로 구성되어야 합니다.

    참고

    패시브 파일 시스템에 허용되는 파일 확장자는 "NCK 내 프로그램 메모리 (페이지 619)"를 참조하십시오.

  - 디렉토리 이름은 NC 표기법의 경우 "_", EES 표기법의 경우 "."으로 구분합니다.

예:

- 패시브 파일 시스템 또는 GDIR:
  - NC 표기법: _N_SUB1_SPF
  - EES 표기법: SUB1.SPF
- 외부 프로그램 메모리:
  - Part 1
  - _Part1
  - Part_1.spf
  - Part1.mpf

DIN 서브프로그램 이름

DIN 서브프로그램 이름에는 다음 규칙이 적용됩니다.

- 첫 번째 문자는 알파벳 "L"이어야 합니다.
- 그 다음 문자는 숫자 (최소 1개의 숫자) 입니다.
- 파일 확장자:
  - 파일 확장자는 3개 문자로 구성되어야 합니다.
  - 디렉토리 이름은 NC 표기법의 경우 "_", EES 표기법의 경우 "."으로 구분합니다.

예:

- L123
- L1_SPF(NC 표기법) 또는 L1.SPF(EES 표기법)

최대 경로 길이

드라이브 이름 및 디렉토리 경로 지정에 최대 128 byte를 사용할 수 있습니다. 파일 이름의 최대 길이는 31 byte입니다. 전체 경로의 최대 길이는 159 byte입니다.

4.4.1.4 서브프로그램 호출을 위한 경로 검색

경로 데이터가 없는 서브프로그램 호출의 경우, 고정된 검색 경로를 처리하여 절대 경로를 결정합니다.

이 경우, 다음 순서에 따라 프로그램 메모리에서 검색이 실행됩니다.

		디렉토리	설명
1		현재 디렉토리/*이름*	현재 디렉토리는 프로그램이 선택된 디렉토리입니다. 실제 디렉토리: • NC 가공 프로그램 메모리 또는 글로벌 가공 프로그램 메모리의 공작물 디렉토리 또는 표준 디렉토리 _N_MPF_DIR 또는 • 외부 프로그램 메모리의 임의의 디렉토리
2		현재 디렉토리 / *name_SPF*	
3		현재 디렉토리 / *name_MPF*	
4	a	//NC:/_N_SPF_DIR / *이름_SPF*	NC 가공 프로그램 메모리 내 서브프로그램 디렉토리
	b	//DEV2:/_N_SPF_DIR / 이름_SPF [1]	글로벌 가공 프로그램 메모리 내 서브프로그램 디렉토리 **참고:** 이 검색 단계는 글로벌 가공 프로그램 메모리가 생성되지 않았거나 프로그램이 NC 가공 프로그램 메모리에서 선택된 경우에는 실행되지 않습니다.
5		CALLPATH로 프로그램된 경로 확장자 검색 ("서브프로그램 호출에 대한 확장 검색 경로 (CALLPATH) (페이지 600)" 참조). **참고:** 이 검색 단계는 CALLPATH가 프로그램되지 않은 경우에는 실행되지 않습니다.	
6		/_N_CUS_DIR/*name_SPF*	사용자 싸이클 디렉토리
7		/_N_CMA_DIR/*name_SPF*	제조업체 싸이클 디렉토리
8		/_N_CST_DIR/*name_SPF*	표준 싸이클 디렉토리

[1] //DEV2:" 예를 들어 글로벌 가공 프로그램 메모리가 생성된 드라이브를 의미합니다.

검색에는 다음 규칙이 적용됩니다.

- 경로 검색은 각 개별 서브프로그램 호출마다 수행됩니다. 즉, 더 높은 레벨의 프로그램이 어디에 위치해 있느냐와 무관하다는 뜻입니다.
- 디렉토리에 따라 다른 파일 유형이 적용됩니다.
- 검색은 낮은 레벨 즉 내포된 디렉토리들이 아니라 한 디렉토리 내에서 이루어집니다.

4.4.1.5 경로와 파일 이름 조사

가공 프로그램에서 판독할 수 있는 다음과 같은 시스템 변수들을 사용하여 NC 프로그램의 경로와 파일 이름을 조사할 수 있습니다.

시스템 변수	유형	의미
$P_STACK	INT	현재 NC 프로그램이 실행되는 프로그램 레벨을 제시합니다.
$P_PATH[<n>]	STRING	필드 인덱스 <n>을 사용하여 선택된 프로그램 레벨에서 처리되는 NC 프로그램의 경로를 제시합니다. 예: $P_PATH[0]는 메인 프로그램에 대한 경로를 제시합니다. 예: "/_N_WKS_DIR/_N_WELLE_WPD/". $P_PATH[$P_STACK - 1]은 호출 프로그램의 경로를 제시합니다. 경로가 NC의 패시브 파일 시스템 또는 글로벌 가공 프로그램 메모리 (GDIR) 에 저장된 NC 프로그램을 가리킬 경우, 그 경로는 NC 표기법으로 제시됩니다. 경로가 글로벌 가공 프로그램 메모리 이외의 외부 프로그램 메모리에 의해 실행되는 NC 프로그램을 가리킬 경우, $P_PATH가 그 경로를 EES 표기법으로 제시합니다.
$P_PROG[<n>]	STRING	필드 인덱스 <n>을 사용하여 선택된 프로그램 레벨에서 처리되는 NC 프로그램의 이름을 제시합니다. NC 프로그램이 NC의 패시브 파일 시스템 또는 글로벌 가공 프로그램 메모리에 저장된 경우, 그 프로그램 이름은 NC 표기법으로 제시됩니다. NC 프로그램이 글로벌 가공 프로그램 메모리 이외의 외부 드라이브에 의해 실행될 경우, $P_PROG는 그 이름을 EES 표기법으로 제시합니다.
$P_PROGPATH	STRING	현재 처리되고 있는 NC 프로그램의 경로를 제시합니다. $P_PROGPATH를 호출하는 방법은 $P_PATH[$P_STACK]과 동일합니다.

시스템 변수	유형	의미	
$P_IS_EES_PATH[<n>]	BOOL	$P_PATH[<n>]에 의해 제시된 경로 또는 $P_PROG[<n>]에 의해 제시된 프로그램 이름이 NC 표기법 또는 EES 표기법에 해당하는지 조사합니다.	
		= FALSE	$P_PATH[<n>] 및 $P_PROG[<n>]은 NC 표기법을 제공합니다. 이는 모든 식별자가 접두어 "_N_"를 갖는다는 것을 의미합니다. 파일 식별자를 위한 분리 부호는 "_"입니다. 예: • NC 표기법의 경로: "/_N_WKS_DIR/_N_MYWPD_WPD/" • NC 표기법의 프로그램 이름:"_N_MYPROG_MPF" NC 표기법으로 표기된 경로는 NC의 패시브 파일 시스템을 가리킬 수도 있고 글로벌 가공 프로그램 메모리를 가리킬 수도 있습니다.
		= TRUE	$P_PATH[<n>] 및 $P_PROG[<n>]는 EES 표기법을 제공합니다. 이는 식별자가"_N_" 접두어를 가지지 않는다는 것을 의미합니다. 파일 식별자를 위한 분리 부호는 "."입니다. 예: • EES 표기법의 경로: "//DEV1:/WKS.DIR/MYWPD.WPD/" • EES 표기법의 프로그램 이름: "MYPROG.MPF"

<n>: 인덱스 <n>은 경로 정보를 판독해야 하는 프로그램 레벨을 정의합니다 (값의 범위: 0 ... 17)

참고

EES 모드에서, 글로벌 가공 프로그램 메모리 (GDIR) 외부에서는 시스템 변수 $P_PROG, $P_PATH, $P_PROGPATH 경로 이름이 EES 표기법으로 제시됩니다. EES 모드에서는 이들 경로 이름을 평가하고 처리하는 사용자 프로그램을 EES 표기법으로 표기된 경로 이름도 처리할 수 있도록 확장해야 합니다.

4.4.2 작업 메모리 (CHANDATA, COMPLETE, INITIAL)

기능

작업 메모리에는 제어 시스템에서 작동되는 현재 시스템 및 사용자 데이터가 저장되어 있습니다 (활성 파일 시스템). 예:

- 활성 머신 데이터
- 공구 옵셋 데이터
- 워크 옵셋
- ...

초기화 프로그램

작업 메모리 데이터를 초기화하는 프로그램입니다. 다음 파일 유형을 사용할 수 있습니다.

파일 유형	설명
name_TEA	머신 데이터
name_SEA	셋팅 데이터
name_TOA	공구 옵셋
name_UFR	워크 옵셋/프레임
name_INI	초기화 파일
name_GUD	전역 사용자 데이터
name_RPA	R-파라미터

데이터 영역

적용할 데이터를 서로 다른 영역에 구성할 수 있습니다. 예를 들어 한 제어 시스템이 여러 개의 채널을 갖거나 흔히 그렇듯이 여러 개의 축을 가질 수도 있습니다.

다음과 같은 데이터 영역을 사용할 수 있습니다.

식별자	데이터 영역
NC	NC별 데이터
CH<n>	채널 관련 데이터 (<n>은 채널 이름)
AX<n>	축 관련 데이터 (<n>은 가공 축 번호)

식별자	데이터 영역
TO	공구 데이터
COMPLETE	모든 데이터

외부 PC에서 초기화 프로그램 생성

데이터 영역 식별자 및 데이터 유형 식별자를 사용하여 영역을 지정할 수 있습니다. 데이터를 저장할 때 이 데이터 영역들은 하나의 장치로 취급됩니다.

_N_AX5_TEA_INI	축 5의 머신 데이터
_N_CH2_UFR_INI	채널 2의 프레임
_N_COMPLETE_TEA_INI	모든 머신 데이터

제어 시스템을 처음 기동하면 제어 시스템이 정상 작동할 수 있도록 데이터 세트가 자동으로 로드됩니다.

다중 채널 제어 절차 (CHANDATA)

여러 채널에 대한 CHANDATA(<channel number>)는 파일 _N_INITIAL_INI에서만 허용됩니다. 이 파일은 제어 시스템의 모든 데이터를 초기화하는 테스트 파일입니다.

프로그램 코드	코멘트
%_N_INITIAL_INI	
CHANDATA(1)	
	; 기계 축 지정, 채널 1:
$MC_AXCONF_MACHAX_USED[0]=1	
$MC_AXCONF_MACHAX_USED[1]=2	
$MC_AXCONF_MACHAX_USED[2]=3	
CHANDATA(2)	
	; 기계 축 지정, 채널 2:
$MC_AXCONF_MACHAX_USED[0]=4	
$MC_AXCONF_MACHAX_USED[1]=5	
CHANDATA(1)	
	; 축 머신 데이터:
	; 일반 정위치 정지 창:
$MA_STOP_LIMIT_COARSE[AX1]=0.2	; 축 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2	; 축 2
	; 미세 정위치 정지 창:
$MA_STOP_LIMIT_FINE[AX1]=0.01	; 축 1
$MA_STOP_LIMIT_FINE[AX1]=0.01	; 축 2

유의사항
CHANDATA 문 가공 프로그램에서 CHANDATA 명령문은 NC 프로그램이 실행되는 채널에만 설정할 수 있습니다. 즉, 이 명령문을 사용하여 NC 프로그램이 잘못된 채널에서 실행되지 않도록 보호할 수 있습니다. 오류가 발생할 경우 진행 중인 프로그램이 중단됩니다.

참고

작업 목록의 INI 파일에는 CHANDATA 명령문이 없습니다.

초기화 프로그램 저장 (COMPLETE, INITIAL)

작업 메모리의 파일을 외부 PC에 저장한 다음 읽을 수 있습니다.

- 파일은 COMPLETE를 통해 저장됩니다.
- INITIAL을 사용하여 모든 영역에서 INI 파일 (_N_INITIAL_INI) 을 생성합니다.

초기화 프로그램 읽기

유의사항
데이터 손실 "INITIAL_INI" 이름을 가진 파일을 리드인했을 때 이 파일에서 제공하지 않는 모든 데이터는 표준 데이터로 초기화됩니다. 단, 머신 데이터는 예외입니다. 따라서 **설정 데이터, 공구 데이터, ZO, GUD 값, ...**은 표준 데이터와 함께 제공됩니다 (일반적으로 "ZERO").

예를 들어 개별 머신 데이터를 읽는 경우 COMPLETE_TEA_INI 파일이 적합합니다. 제어 시스템은 이 파일에서 머신 데이터만 필요로 합니다. 이 경우에서 다른 데이터 영역은 영향을 받지 않기 때문입니다.

초기화 프로그램 로딩

또한 한 채널의 데이터만 사용하는 경우 INI 프로그램을 선택하여 가공 프로그램으로 호출할 수 있습니다. 다시 말해 프로그램별로 데이터를 초기화할 수도 있습니다.

4.5 파일 처리

4.5.1 파일 쓰기 (WRITE)

WRITE 명령을 사용하여 NC 프로그램의 설정/데이터를 패시브 파일 시스템의 파일 (로그 파일) 끝에, 또는 외장 프로그램 메모리에 기록할 수 있습니다. 이것은 현재 실행 중인 프로그램일 수도 있습니다.

참고

프로그램 메모리에 그러한 파일이 존재하지 않는 경우 해당 파일이 생성되며, WRITE 명령을 사용하여 그 파일에 기록을 할 수 있습니다.

요구 사항

현재 설정된 보호 레벨은 파일의 WRITE 권한과 동일하거나 커야 합니다. 그렇지 않을 경우 에러 메시지 (에러 변수의 반환값 = 13) 와 함께 액세스가 거부됩니다.

구문

```
DEF INT <error>
...
WRITE(<error>,"<file name>"/"<ExtG>","<set/data>")
```

의미

WRITE:	지정된 파일의 끝에 블록 또는 데이터를 추가하는 명령어.		
<error>:	**파라미터 1:** 에러 값을 반환하기 위한 변수		
	유형:	INT	
	값:	0	에러 없음
		1	허용되지 않는 경로
		2	경로를 찾을 수 없습니다.
		3	파일을 찾을 수 없습니다.
		4	잘못된 파일 유형
		10	파일이 가득 참
		11	파일 사용 중
		12	사용 가능한 리소스 없음
		13	액세스 권한 없음
		14	출력 디바이스에서 누락되거나 실패한 EXTOPEN
		15	외부 장치로 쓸 때 발생한 에러
		16	외부 경로가 잘못되었습니다.
<file name>:	**파라미터 2:** 지정된 블록 또는 데이터가 추가될 파일의 이름.		
	유형:	STRING	
	실제 파일 이름 앞에 절대 경로를 지정할 수 있습니다. 지정된 경로가 없는 경우 현재 디렉토리 (선택한 프로그램의 디렉토리) 에서 해당 파일을 검색합니다. 경로 데이터에 관한 규칙은 "프로그램 메모리 파일 주소 지정 (페이지 623)" 참조.		
<ExtG>:	"Process DataShare" 코드를 사용하여 데이터를 외부 장치/파일로 출력하려는 경우, 파일 이름 대신 열고자 하는 외부 장치/파일의 심볼 이름을 지정해야 합니다.		
	유형:	STRING	
	자세한 내용은 "공정 데이터 공유 - 외부 장치/파일로 출력 (EXTOPEN, WRITE, EXTCLOSE) (페이지 1136)"을 참조하십시오. **참고:** 식별자는 EXTOPEN 명령에서 지정한 식별자와 동일해야 합니다.		
<block/ data>:	**파라미터 3:** 지정된 파일에 추가할 블록 또는 데이터.		
	유형:	STRING	

참고

패시브 파일 시스템 또는 외부 프로그램 메모리에 기록을 할 때 WRITE 명령이 출력 문자열 끝에 암시적으로 "LF" 문자 (LINE FEED = 새 라인) 를 삽입합니다.

이 같은 동작은 "Process DataShare" 코드를 사용하여 외부 장치/파일로 출력할 경우에는 적용되지 않습니다. "LF"도 출력하는 경우에는 출력 문자열에 명시적으로 지정해야 합니다.

예 3: 암묵적/명시적 "LF"를 참조하십시오!

보충 조건

- **최대 파일 크기 (→ 장비 제조업체)**
  패시브 파일 시스템에서 가능한 로그 파일의 최대 파일 크기는 머신 데이터로 설정합니다.
  MD11420 $MN_LEN_PROTOCOL_FILE
  최대 파일 길이는 패시브 파일 시스템에서 WRITE 명령을 사용해 생성한 모든 파일에 적용됩니다. 최대 파일 길이를 초과하면 에러 메시지가 출력되고 해당 블록 또는 데이터가 저장되지 않습니다. 사용 가능한 메모리가 충분할 경우 새 파일을 생성할 수 있습니다.

예

예 1: 절대 경로 데이터가 없는 패시브 파일 시스템에 WRITE 명령 적용

프로그램 코드	코멘트
N10 DEF INT ERROR	;에러 변수 정의.
N20 WRITE(ERROR,"PROT","LOG FROM 7.2.97")	; 파일 _N_PROT_MPF에 "LOG FROM 7.2.97"라는 문구를 기록함.
N30 IF ERROR	;에러 평가.
N40 MSG ("Error with WRITE command:" <<ERROR)	
N50 M0	
N60 ENDIF	
...	

예 2: 절대 경로 데이터를 포함한 패시브 파일 시스템에 WRITE 명령 적용

프로그램 코드
...
WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","LOG FROM 7.2.97")
...

예 3: 암묵적/명시적 "LF"

a) 암시적으로 생성된 "LF"를 패시브 파일 시스템에 쓰기

프로그램 코드

```
...
N110 DEF INT ERROR
N120 WRITE(ERROR,"/_N_MPF_DIR/_N_MYPROTFILE_MPF","MY_STRING")
N130 WRITE(ERROR,"/_N_MPF_DIR/_N_MYPROTFILE_MPF","MY_STRING")
N140 M30
```

출력 결과:

MY_STRING

MY_STRING

b) 암시적으로 생성된 "LF" 없이 외부 파일에 쓰기

프로그램 코드

```
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING")
N250 WRITE(ERROR,DEV_1,"MY_STRING")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30
```

출력 결과:

MY_STRINGMY_STRING

c) 명시적으로 생성된 "LF"를 외부 파일에 쓰기

a에서와 동일한 결과를 얻으려면 다음을 프로그래밍해야 합니다.

프로그램 코드

```
...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")
```

프로그램 코드

```
N250 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30
```

출력 결과:

MY_STRING

MY_STRING

4.5.2 파일 삭제 (DELETE)

DELETE 명령은 WRITE 명령을 이용해 생성했는지 여부에 관계 없이 모든 파일을 삭제합니다. 또한 더 높은 레벨의 액세스 권한을 사용하여 생성된 파일도 DELETE를 이용해 삭제할 수 있습니다.

구문

```
DEF INT <error>
DELETE(<error>,"<file name>")
```

의미

DELETE:	지정된 파일을 삭제하는 명령어.		
<error>:	에러 값을 반환하기 위한 변수.		
	유형.	INT	
	값:	0	에러 없음
		1	허용되지 않는 경로입니다.
		2	경로를 찾을 수 없습니다.
		3	파일을 찾을 수 없습니다.
		4	잘못된 파일 유형
		11	파일 사용 중
		12	사용 가능한 리소스 없음
		20	기타 에러

<file name>:	삭제할 파일 이름	
	유형:	STRING
	실제 파일 이름 앞에 절대 경로를 지정할 수 있습니다. 지정된 경로가 없는 경우 현재 디렉토리 (선택한 프로그램의 디렉토리) 에서 해당 파일을 검색합니다. 경로 데이터에 관한 규칙은 "프로그램 메모리 파일 주소 지정 (페이지 623)" 참조.	

예

프로그램 코드	코멘트
N10 DEF INT ERROR	;에러 변수 정의.
N15 STOPRE	; 전처리 정지.
N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")	; 서브프로그램 디렉토리에서 파일 TEST1 을 삭제합니다.
N30 IF ERROR	;에러 평가.
N40 MSG("error for DELETE command:" <<ERROR)	
N50 M0	
N60 ENDIF	

4.5.3 파일의 행 읽기 (READ)

READ 명령은 지정된 파일에서 하나 또는 여러 라인을 읽고 STRING 유형의 배열에 읽은 정보를 저장합니다. 이 배열에서 각각의 읽은 행에 배열 요소가 포함됩니다.

요구 사항

현재 설정된 보호 레벨은 파일의 READ 권한과 동일하거나 커야 합니다. 그렇지 않을 경우 에러 메시지 (에러 변수의 반환값 = 13) 와 함께 액세스가 거부됩니다.

구문

```
DEF INT <error>
DEF STRING[<string length>] <result>[<n>,<m>]
READ(<error>,"<file name>",<start line>,<number of
lines>,<result>)
```

의미

READ:	지정된 파일에서 라인을 읽고 해당 라인을 변수 배열에 저장하도록 하는 명령어.		
<error>:	에러 값 리턴을 위한 변수 (기준별 호출 파라미터)		
	유형.	INT	
	값:	0	에러 없음
		1	허용되지 않는 경로입니다.
		2	경로를 찾을 수 없습니다.
		3	파일을 찾을 수 없습니다.
		4	잘못된 파일 유형
		11	파일 사용 중
		13	액세스 권한이 부족합니다.
		21	라인이 존재하지 않음 (<start line> 또는 <number of lines> 파라미터가 지정된 파일의 라인 수를 초과한 경우).
		22	결과 변수 (<result>) 의 필드 길이가 너무 작음.
		23	라인의 범위가 너무 큼 (선택한 <number of lines> 파라미터가 너무 커서 읽기가 파일 끝을 초과하여 진행).
<file name>:	읽을 파일의 이름 (값별 호출 파라미터)		
	유형:	STRING	
	실제 파일 이름 앞에 절대 경로를 지정할 수 있습니다. 지정된 경로가 없는 경우 현재 디렉토리 (선택한 프로그램의 디렉토리) 에서 해당 파일을 검색합니다. 경로 데이터에 관한 규칙은 "프로그램 메모리 파일 주소 지정 (페이지 623)" 참조.		
<start line>:	읽을 파일 섹션의 시작 라인 (값별 호출 파라미터)		
	유형:	INT	
	값:	0	<number of lines> 파라미터가 지정한 수의 라인을 파일 끝 전까지 읽습니다.
		1 ~ n	읽을 첫 번째 라인의 번호.
<number of lines>:	읽을 라인의 개수 (값별 호출 파라미터)		
	유형:	INT	

<result>:	결과 변수 (기준별 호출 파라미터) 읽은 텍스트가 저장된 변수 배열.	
	유형:	STRING (최대 길이: 255)
	<number of lines> 파라미터가 지정한 라인의 개수가 결과 변수의 배열 크기 [<n>,<m>]보다 작은 경우 나머지 배열 요소는 수정되지 않습니다. 제어 문자 "LF" (줄 바꿈) 또는 "CR LF" (캐리지 리턴 줄 바꿈)를 사용해 라인을 종료하면 결과 변수에 저장되지 **않습니다**. 읽은 라인이 지정된 문자열 길이보다 길면 초과한 부분이 잘립니다. 에러 메시지가 출력되지 않습니다.	

참고

이진 파일은 읽어올 수 없습니다. 읽기를 시도하면 "잘못된 데이터 유형" 에러가 출력됩니다 (에러 변수의 반환값 = 4). 다음 유형의 파일을 읽을 수 없습니다. _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

예

프로그램 코드	코멘트
N10 DEF INT ERROR	; 에러 변수 정의.
N20 DEF STRING[255] RESULT[5]	; 결과 변수 정의.
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF",1,5,RESULT)	;도메인 및 파일 식별자가 있는 파일 이름 및 경로 이름.
N40 IF ERROR <>0	;에러 평가.
N50 MSG("ERROR"<<ERROR<<"ON READ COMMAND")	
N60 M0	
N70 ENDIF	
...	

4.5.4 파일의 존재 여부 점검 (ISFILE)

ISFILE명령은 파일이 프로그램 메모리 내에 존재하는지 점검합니다.

구문

<결과>=ISFILE("<파일 이름>")

의미

ISFILE:	파일의 가용성을 점검하는 명령		
<파일 이름>:	가용성을 확인할 파일의 이름.		
	유형:	STRING	
	실제 파일 이름 앞에 절대 경로를 지정할 수 있습니다. 지정된 경로가 없는 경우 현재 디렉토리 (선택한 프로그램의 디렉토리) 에서 해당 파일을 검색합니다. 경로 데이터에 관한 규칙은 "프로그램 메모리 파일 주소 지정 (페이지 623)" 참조.		
<결과>:	점검 결과를 지정할 결과 변수.		
	유형.	BOOL	
	값:	TRUE	파일이 있음
		FALSE	파일이 없음

예

예 1

프로그램 코드	코멘트
N10 DEF BOOL RESULT	; 결과 변수 정의.
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(RESULT==FALSE)	
N40 MSG("FILE DOES NOT EXIST")	
N50 M0	
N60 ENDIF	
...	

예 2

프로그램 코드	코멘트
N10 DEF BOOL RESULT	; 결과 변수 정의.
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(NOT ISFILE("TESTFILE"))	
N40 MSG("FILE DOES NOT EXIST")	
N50 M0	
N60 ENDIF	
...	

4.5.5 파일 정보 읽기 (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

FILEDATE, FILETIME, FILESIZE, FILESTAT 및 FILEINFO 명령은 최종 쓰기 액세스 날짜 및 시간, 현재 파일 크기, 파일 상태 등 특정 파일 정보를 읽어옵니다.

요구 사항

현재 설정된 보호 레벨은 파일의 상위 디렉토리의 표시 권한과 동일하거나 커야 합니다. 그렇지 않을 경우 에러 메시지 (에러 변수의 반환값 = 13) 와 함께 액세스가 거부됩니다.

구문

```
FILE....(<오류>,"<파일 이름>",<결과>)
```

의미

FILEDATE:	파일에 마지막으로 쓰기 액세스한 **날짜**를 반환합니다.
FILETIME:	파일에 마지막으로 쓰기 액세스한 **시간**을 반환합니다.
FILESIZE:	파일의 **현재 크기**를 반환합니다.
FILESTAT:	다음 **권한**을 기준으로 파일의 **상태**를 반환합니다. • 읽기(r: read) • 쓰기(w: write) • 실행(x: execute) • 보이기(s: show) • 삭제(d: delete) **참고:** 이들 보호 레벨은 패시브 파일 시스템에만 국한되는 속성입니다. 따라서 외부 프로그램 메모리에 액세스할 때 FILESTAT은 디폴트 사용권한 (77777) 을 제공합니다.
FILEINFO:	파일의 경우, FILEDATE, FILETIME, FILESIZE 및 FILESTAT을 통해 읽어낼 수 있는 **정보의 총합**을 제공합니다.

<오류>:	에러 값 리턴을 위한 변수 (기준별 호출 파라미터)			
	유형.	VAR INT		
	값:	0	에러 없음	
		1	허용되지 않는 경로입니다.	
		2	경로를 찾을 수 없습니다.	
		3	파일을 찾을 수 없습니다.	
		4	잘못된 파일 유형	
		13	액세스 권한이 부족합니다.	
		22	결과 변수 (<result>) 의 문자열 길이가 너무 작습니다.	
<file name>:	파일 정보를 읽어올 파일의 이름			
	유형:	CHAR[160]		
	실제 파일 이름 앞에 절대 경로를 지정할 수 있습니다. 지정된 경로가 없는 경우 현재 디렉토리 (선택한 프로그램의 디렉토리) 에서 해당 파일을 검색합니다. 경로 데이터에 관한 규칙은 "프로그램 메모리 파일 주소 지정 (페이지 623)" 참조.			
<result>:	결과 변수(Call-By-Reference 파라미터) 요청한 파일 정보를 저장하는 변수.			
	유형:	VAR CHAR[8]	at	FILEDATE 형식: "날짜.월.연도"
		VAR CHAR[8]	at	FILETIME 형식: "시간.분.초"
		VAR INT	at	FILESIZE 파일 크기는 바이트로 출력됩니다.
		VAR CHAR[5]	at	FILESTAT 형식: "rwxsd" (r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete)
		VAR CHAR[32]	at	FILEINFO 형식: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"

예

프로그램 코드	코멘트
N10 DEF INT ERROR	; 에러 변수 정의.
N20 STRING[32] RESULT	; 결과 변수 정의.
N30 FILEINFO(ERROR,"/_N_MPF_DIR/ _N_TESTFILE_MPF",RESULT)	; 도메인을 갖는 파일 이름, 파일 ID 및 경로 데이터.
N40 IF ERROR <> 0	; 에러 평가
N50 MSG("ERROR"<<ERROR<<"FOR FILE INFORMATION COMMAND")	
N60 M0	
N70 ENDIF	
...	

결과 RESULT 변수에서, 예는 다음 결과를 제공할 수 있습니다.

"77777 12345678 26.05.00 13:51:30"

4.6 보호 영역

4.6.1 보호 영역 정의(CPROTDEF, NPROTDEF)

머신 요소를 충돌로부터 보호하는 보호 영역은 블록 내 가공 프로그램에서 정의됩니다. 이는 다음 요소를 포함합니다.

1. 가공 평면 정의
   실제 보호 영역 정의 이전에 보호 영역의 형상 설명이 참조하는 가공 평면을 반드시 선택해야 합니다.
2. 정의 시작
   특정 NC 명령어에 따라 채널별 또는 머신별 보호 영역이 만들어집니다.
3. 보호 영역의 형상 설명
   보호 영역의 형상은 이송 모션으로 정의됩니다. 이들은 실행되지 않고 이전 또는 다음의 기하학적 설명과도 관련이 없습니다. 이들은 보호 영역만 정의합니다.
4. 정의 종료

구문

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/... X/Y/Z...
...
```

```
EXECUTE(<Var>)
```

의미

DEF INT <Var>:	로컬 도움말 변수의 정의, INTEGER 데이터 유형		
<Var>:	Help 변수의 이름		
G17/G18/G19:	가공 평면 **참조:** 정의가 끝나기 전에 가공 평면을 변경하는 것은 허용되지 않습니다. 직교 좌표를 프로그래밍하는 것은 정의 시작 및 끝 사이에는 허용되지 않습니다.		
CPROTDEF():	**채널**별 보호 영역을 정의하기 위해 사전 정의된 절차		
NPROTDEF():	**머신**별 보호 영역을 정의하기 위해 사전 정의된 절차		
<n>:	정의된 보호 영역 수		
	데이터 유형:	INT	
<t>:	보호 영역의 유형		
	데이터 유형:	BOOL	
	값:	TRUE	**공구** 보호 영역
		FALSE	**공작물** 보호 영역
<applim>:	3차원에서의 제한 유형		
	데이터 유형:	INT	
	값:	0	제한 없음
		1	플러스 방향으로 제한
		2	마이너스 방향으로 제한
		3	플러스 방향 및 마이너스 방향으로 제한
<AppPlus>:	양의 방향 및 3차원에서의 제한 값		
	데이터 유형:	REAL	

<appminus>:	마이너스 방향 및 3차원에서의 제한 값
	데이터 유형: REAL
G0/G1/... X/Y/Z... ...:	보호 영역의 형상은 선택한 가공 평면에서 최대 11개의 이송 이동으로 명시됩니다. 첫 번째 이송은 형상 이동입니다. 형상 설명의 마지막 점이 형상 설명의 첫 번째 점과 일치해야 합니다. 유효한 보호 영역은 형상의 왼쪽 영역입니다. • 내부 보호 영역 내부 보호 영역의 형상은 반드시 시계 반대 방향으로 설명되어야 합니다. • 외부 보호 영역(공작물 보호 영역에 대해서만 허용) 외부 보호 영역의 형상은 반드시 시계 방향으로 설명되어야 합니다. 다음 형상 요소가 허용됩니다. • 직선 형상 요소의 경우 G0, G1 • 시계 방향의 원호 세그먼트를 위한 G2 공작물 보호 영역에 대해서만 허용됨. 볼록해야 하므로 공구 보호 영역에 대해서는 허용되지 않음. • CCW 방향 원호 세그먼트의 경우 G3 **참조:** 보호 영역은 완전한 원으로는 설명될 수 없습니다. 완전한 원은 반드시 2개의 반원으로 나뉘어져야 합니다. **참조:** 시퀀스 G2 → G3 또는 G3 → G2 는 허용되지 않습니다. 짧은 G1 블록은 반드시 두 개의 반원 사이에 삽입되어야 합니다.
EXECUTE(<Var>):	정의의 끝을 표시하는 미리 정해진 절차 EXECUTE 를 사용하면 스위치가 정상 프로그램 처리로 되돌아 갑니다.

예제

"보호 영역 활성화/비활성화 (CPROT, NPROT) (페이지 650)" 아래 예를 참조하십시오.

추가 정보

머신별 보호 영역

머신별 보호 영역 또는 형상이 기하 축을 사용하여 정의됩니다. 즉 채널의 기본 좌표계(BCS)에 참조됩니다. 머신별 보호 영역이 활성화되는 모든 채널에서 정확한 보호 영역 모니터링이 발생할 수 있도록 하기 위해서는, 모든 관련 채널의 기본 좌표계(BCS)가 반드시 동일해야 합니다.

- 기계 영점을 기준으로 한 좌표 기원의 위치
- 좌표 축의 방향

형상 설명 기준점

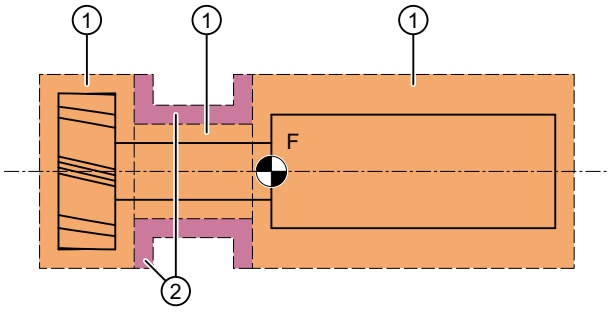
- 공구 보호 영역
  공구보호 영역의 좌표는 반드시 **공구 홀더 기준점 F**를 기준으로 한 절대 값으로 명시되어야 합니다.
- 공작물 보호 영역
  공작물 보호 영역의 좌표는 반드시 **기본 좌표계(BCS)**의 영점을 기준으로 한 절대 값으로 명시되어야 합니다.

회전 중심에서 대칭되는 보호 영역

축 또는 회전 중심에서 대칭되는 보호 영역(예: 스핀들 척)의 경우 회전 중심까지가 아닌 형상 전체를 반드시 묘사해야 합니다.

공구 보호 영역

공구 보호 영역은 항상 볼록해야 합니다. 오목한 보호 영역을 원하는 경우 여러 개의 볼록한 보호 영역으로 나누어야 합니다.



① 볼록 보호 영역
② 오목 보호 영역(**허용되지 않음**)
F 공구 홀더 원점

일반 조건

보호 영역 정의 중 다음 기능은 반드시 활성 상태가 아니거나 사용되지 않아야 합니다.

- 공구 반경 보정(커터 반경 보정, 공구 노즈 반경 보정)
- 변환
- 원점 접근(G74
- 고정 정지점 접근(G75)
- 드웰 시간 (G4)
- 블록 탐색 정지(STOPRE)
- 프로그램 종료(M17, M30)
- M 기능: M0, M1, M2

4.6.2 보호 영역 활성화/비활성화 (CPROT, NPROT)

이전에 가공 프로그램에서 정의된 보호 영역은 언제라도 활성화될 수 있습니다. 또는 PLC 사용자 프로그램에 의한 다음 활성화를 위해 미리 활성화될 수도 있습니다. 활성 보호 영역은 언제라도 비활성화될 수 있습니다.

활성화하거나 미리 활성화하는 경우, 보호 영역의 원점을 상대적으로 움직이는 것도 가능합니다.

참고

보호 영역이 활성화된 채널의 모든 기하 축 참조가 끝난 이후에만 보호 영역이 감안됩니다.

참고

보호 영역 모니터링

공구 보호 영역이 활성화되지 않은 상태라면, 공작물 보호 영역을 기준으로 공구 경로를 검사합니다.

공작물 방향 보호 영역이 활성화된 상태라면, 보호 영역 모니터링은 없습니다.

구문

```
CPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
NPROT(<n>,<Status>,<XMov>,<YMov>,<ZMov>)
```

의미

CPROT:	**채널**별 보호 영역을 활성화하기 위해 사전 정의된 절차		
NPROT:	**머신**별 보호 영역을 활성화하기 위해 사전 정의된 절차		
<n>:	보호 영역의 개수		
	데이터 유형:	INT	
<Status>:	채널별 활성화 상태는 파라미터를 사용하여 설정됩니다.		
	데이터 유형:	INT	
	값:	0	보호 영역 비활성화
		1	보호 영역 사전 활성화
		2	보호 영역 활성화
		3	조건부 정지를 통한 보호 영역 사전 활성화
<xMov>,<yMov>,<zMov>:	X/Y/Z 방향의 추가 옵셋값 옵셋을 1차원, 2차원 또는 3차원에서 수행할 수 있습니다. 옵셋값은 다음을 참조합니다. • **공작물** 보호 영역에 대한 기계 영점 • **공구**별 보호 영역을 위한 공구 캐리어 기준점 F		
	데이터 유형:	REAL	

예

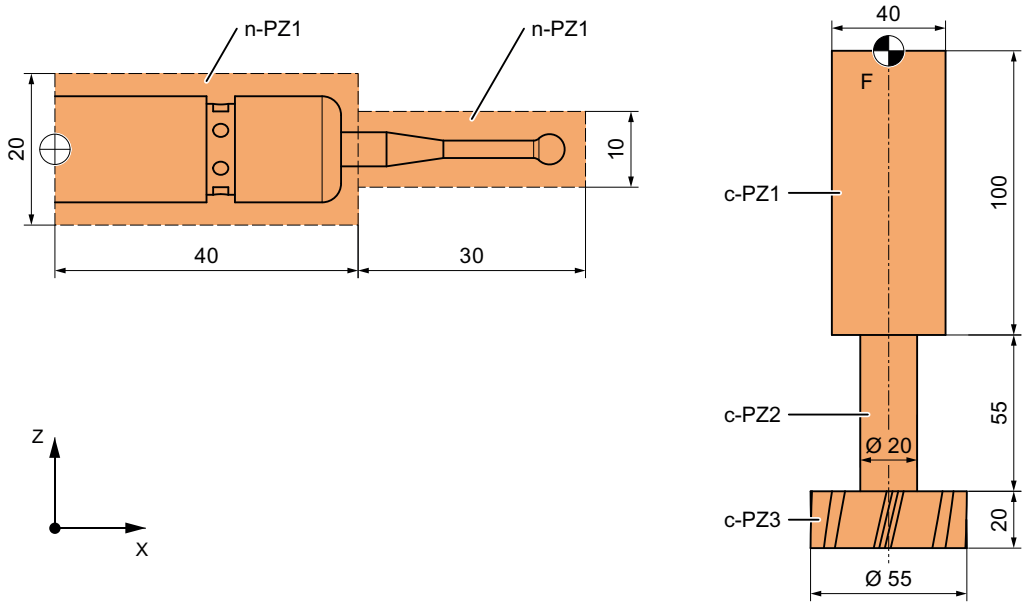
측정 프로브를 통해 밀링 커터의 충돌 가능성이 밀링 기계에서 모니터링되어야 합니다. 측정 프로브의 위치는 기능 작동 시 옵셋에 의해 정의되어야 합니다.

이를 위해 다음과 같은 보호 영역이 정의됩니다.

- 측정 프로브 홀더(n-PZ1) 및 측정 프로브 자체(n-PZ2)에 대한 머신별 및 공작물 관련 보호 영역.
- 밀링 커터 홀더(c-PZ1), 커터 생크(c-PZ2) 및 밀링 커터 자체(c-PZ3)에 대한 채널별 및 공구 관련 보호 영역.

모든 보호 영역의 오리엔테이션은 Z 방향입니다.

기능 작동 시 측정 프로브의 원점 위치는 X = -120, Y = 60 및 Z = 80이어야 합니다.



① 프로브 보호 영역의 이름

F 공구 홀더 원점

프로그램 코드	코멘트
DEF INT PROTZONE	; 도움말 변수 정의
G17	; 가공 평면 XY
; 보호 영역 정의:	
NPROTDEF(1,FALSE,3,10,-10)	; 보호 영역 n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE(PROTZONE)	
NPROTDEF(2,FALSE,3,5,-5)	; 보호 영역 n-PZ2
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE(PROTZONE)	
CPROTDEF(1,TRUE,3,0,-100)	; 보호 영역 c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	

프로그램 코드	코멘트
Y-20	
EXECUTE(PROTZONE)	
CPROTDEF(2,TRUE,3,-100,-150)	; 보호 영역 c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE(PROTZONE)	
CPROTDEF(3,TRUE,3,-150,-170)	; 보호 영역 c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE(PROTZONE)	
; 보호 영역 활성화:	
NPROT(1,2,-120,60,80)	; 옵셋을 통한 보호 영역 n-PZ1 활성화
NPROT(2.2,-120,60,80)	; 옵셋을 통한 보호 영역 n-PZ2 활성화
CPROT(1,2,0,0,0)	; 보호 영역 c-PZ1 활성화
CPROT(2,2,0,0,0)	; 보호 영역 c-PZ2 활성화
CPROT(3,2,0,0,0)	; 보호 영역 c-PZ3 활성화

추가 정보

제어 시스템에 전원이 공급된 이후의 활성화 상태

제어 시스템에 전원이 공급되고 축을 참조하고 난 후 보호 영역은 이미 활성화되어 있습니다. 이 경우, 보호 영역의 경우 다음 시스템 변수가 TRUE로 설정됩니다.

- $SN_PA_ACTIV_IMMED[<n>](머신별 보호 영역) 또는
- $SC_PA_ACTIV_IMMED[<n>](채널별 보호 영역)
  지수 "<n>"은 보호 영역의 개수와 일치합니다. 0 = 1. 보호 영역

보호 영역은 상태 = 2, 옵셋은 없는 상태에서 활성화됩니다.

여러 번의 보호 영역 활성화

머신별 보호 영역은 동시에 여러 채널에서 활성화할 수 있습니다(예: 두 개의 반대 측면이 있는 테일스톡의 보호 영역). 모든 형상 축이 원점 복귀된 경우에만 보호 영역이 모니터링됩니다.

단일 채널에서의 여러 다른 옵셋으로는 보호 영역을 동시에 활성화 할 수 없습니다.

공구 반경 보정이 활성인 경우 보호 영역 모니터링

공구 반경 보정이 활성인 경우 공구 반경 보정의 평면이 보호 영역 정의 평면과 일치하는 경우에만 보호 영역을 모니터링할 수 있습니다.

4.6.3 보호 영역 위반, 작업 영역 제한 및 소프트웨어 리미트 스위치 점검 (CALCPOSI)

기능

공작물 좌표계(WCS)에서 CALCPOSI 기능은 **기하 축** 이 활성 한계를 위반하지 않으면서도 특정 거리를 이동할 수 있는지 시작점에서부터 점검합니다. 한계 때문에 지정된 거리까지 완전히 이송할 수 없는 경우 양의 십진수 상태 값과 최대 이송 거리가 반환됩니다.

정의

```
INT CALCPOSI(VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

구문

```
<Status> = CALCPOSI(VAR <Start>, VAR <Dist>, VAR <Limit>, VAR
<MaxDist>, <MeasSys>, <TestLim>)
```

의미

CALCPOSI(...):	기하 축 관련 한계 위반을 시험하기 위한 사전 정의된 기능	
	전처리 정지:	아니오
	블록에서만:	예

<status>: (부분 1)	기능 반환값. 음수는 에러 상태를 의미합니다.		
	데이터 유형:		INT
	값 범위:		-8 ≤ x ≤ 100000
	값:	0	지정된 거리를 완전히 이송할 수 있습니다.
		-1	최소 하나의 콤포넌트가 <Limit>에서 음수.
		-2	변환 계산 에러. 예: 이송 거리가 특이점 (singularity) 을 통과하여 축 위치를 정의할 수 없습니다.
		-3	지정된 이송 거리 <Dist>와 최대 이송 거리 <MaxDist>가 일차 종속적입니다. **참고** <TestLim>에서 bit 4 == 1인 경우에만 발생할 수 있습니다.
		-4	<Dist>에 지정된 이송 거리의 제한 표면에 대한 사영 값이 제로 벡터이거나 이송 거리가 위반한 제한 표면과 직각입니다. **참고** <TestLim>에서 bit 5 == 1인 경우에만 발생할 수 있습니다.
		-5	<TestLim>에서 bit 4 == 1 AND bit 5 == 1
		-6	이송 한계 점검 시 고려해야 하는 기계축 중 최소 하나가 참조되지 않았습니다.
		-7	충돌 방지 기능: 운동학적 체인 또는 보호 영역이 잘못 정의되었습니다.
		-8	충돌 방지 기능: 메모리가 부족하여 이 명령을 실행할 수 없습니다.

<status>: (부분 2)	일 단위 숫자		
	참고 여러 한계를 동시에 위반한 경우 지정된 이송 거리에 대한 제한이 가장 큰 한계가 표시됩니다.		
	값:	1	소프트웨어 리미트 스위치가 이송 거리를 제한합니다.
		2	작업 영역 제한이 이송 거리를 제한합니다.
		3	보호 영역이 이송 거리를 제한합니다.
		4	충돌 방지 기능: 보호 영역이 이송 경로를 제한합니다.
	10자리 숫자		
	값:	1x	초기값이 제한을 위반합니다.
		2x	지정된 직선이 제한을 위반합니다. 종점이 한계 자체를 위반하지 않지만 시작점부터 종점까지의 경로에서 한계값을 위반하는 경우에도 이 값이 반환됩니다(예: 보호 영역을 통과하는 경우, 비선형 변형을 위한 WCS 내 곡선 소프트웨어 리미트 스위치, 전송).
<status>: (부분 3)	100자리 숫자		
	값:	1xx	AND 일 단위 숫자 == 1 또는 2: 양의 한계값을 위반했습니다.
			AND 일 단위 숫자 == 3 [1]: NC 보호 영역을 위반했습니다.
		2xx	AND 일 단위 숫자 == 1 또는 2: 음의 한계값을 위반했습니다.
			AND 일 단위 숫자 == 3 [1]: 채널별 보호 영역이 위반되었습니다.
<status>: (부분 4)	1000자리 숫자		
	값:	1xxx	AND 일 단위 숫자 == 1 또는 2: 한계를 위반한 축의 번호에 곱해지는 인수. 축 번호는 1부터 시작합니다. 참조: • 소프트웨어 한계 스위치: 기계 축 • 작업 영역 제한: 기하 축 AND 일 단위 숫자 == 3 [1]: 위반한 보호 영역의 번호에 곱해지는 인수.

<status>: (부분 5)	십만 단위 숫자		
	값:	0xxxxx	십만 단위 숫자 == 0: <dist> 는 바뀌지 않습니다.
		1xxxxx	제한 표면에서 추가 이동 방향을 정의하는 방향 벡터가 <Dist>에 반환됩니다. 다음 보충 조건을 충족하는 경우에만 발생할 수 있습니다. • 시작점이 아닌 곳에서 소프트웨어 리미트 스위치 또는 작업 영역 제한 위반 • 변환이 **비활성** • <TestID>, bit 4 또는 bit 5 == 1
<Start>:	시작 위치와 벡터로 지정: • <Start> [0]: 첫 번째 기하 축 • <Start> [1]: 두 번째 기하 축 • <Start> [2]: 세 번째 기하 축		
	파라미터 유형:		입력
	데이터 유형:		VAR REAL [3]
	값 범위:		-최대. 실제 값 ≤ x[n] ≤ +max. 실제 값

<Dist>:	벡터로 지정.	
	입력: 증분 이송 거리 • <Dist> [0]: 첫 번째 기하 축 • <Dist> [1]: 두 번째 기하 축 • <Dist> [2]: 세 번째 기하 축	
	출력(<Status>가 십만 단위 숫자로 설정된 경우만 해당):	
	<Dist>에는 WCS에서 추가 이송 방향을 지정하는 단위 벡터 **v**가 출력됩니다.	
	사례 1: <TestID>, bit 4 == 1인 경우 벡터 **v** 형성 입력 벡터 <Dist> 및 <MaxDist>가 이동 평면을 지나갑니다. 이 평면을 위반한 제한 표면이 가로지릅니다. 두 평면의 교차 라인이 벡터 **v**의 방향을 정의합니다. 입력 벡터 <MaxDist>와 **v** 사이의 각도가 90°를 넘지 않도록 오리엔테이션(부호)을 선택합니다.	
	사례 2: <TestID>, bit 5 == 1인 경우 벡터 **v** 형성 벡터 **v** 는 <Dist>에 출력된 이송 벡터의 제한 표면에 대한 사영 방향에서 단위 벡터입니다. 제한 표면에 대한 이송 벡터의 사영값이 제로 벡터인 경우 에러가 반환됩니다.	
	파라미터 유형:	입력/출력
	데이터 유형:	VAR REAL [3]
	값 범위:	-최대. 실제 값 ≤ x[n] ≤ +max. 실제 값

<Limit>:	길이 5인 배열로 지정. • <Limit> [0 - 2]: 기하 축과 한계 사이의 최소 간격 – <Limit> [0]: 첫 번째 기하 축 – <Limit> [1]: 두 번째 기하 축 – <Limit> [2]: 세 번째 기하 축 다음 방법으로 최소 간격을 유지합니다. – 작업 영역 제한: 제한 없음 – 소프트웨어 한계 스위치: 활성인 변환이 없거나 기하 축을 직선 축에 명확히 지정할 수 있는 변환 (예: 5축 변환) 이 활성인 경우. • <Limit> [3]: 비선형 변환이기 때문에 기하 축을 지정할 수 없는 직선 기계 축에 대한 최소 간격을 지정합니다. 또한 이 값은 일반 보호 영역 및 충돌 방지 보호 영역 모니터링을 위한 한계값으로도 사용됩니다. • <Limit> [4]: 비선형 변환이기 때문에 기하 축을 지정할 수 없는 로터리 기계 축에 최소 간격을 지정합니다. **참고** 이 값은 특수 변환을 위한 소프트웨어 리미트 스위치 모니터링에만 활성화됩니다.	
	파라미터 유형:	입력
	데이터 유형:	VAR REAL [5]
	값 범위:	-최대. 실제 값 ≤ x[n] ≤ +max. 실제 값
<MaxDist>:	지정된 축 한계 최소 간격을 관련 기계 축에서 위반하지 않는 증분 이송 거리에 대한 벡터 지정. • <Dist> [0]: 첫 번째 기하 축 • <Dist> [1]: 두 번째 기하 축 • <Dist> [2]: 세 번째 기하 축 이송 거리가 제한되지 않을 경우 이 반환 파라미터의 내용은 <Dist>의 내용과 동일합니다.	
	<TestID>의 경우, Bit 4 == 1: <Dist> 및 <MaxDist> <MaxDist> 및 <Dist>는 이동 평면을 지나가는 벡터를 입력 값으로 갖습니다. 두 벡터는 서로 일차 종속적이어야 합니다. <MaxDist>의 절대치는 임의로 지정됩니다. 이동 방향 계산에 대해서는 <Dist>의 설명을 참조하십시오.	
	파라미터 유형:	Output
	데이터 유형:	VAR REAL [3]
	값 범위:	-최대. 실제 값 ≤ x[n] ≤ +max. 실제 값

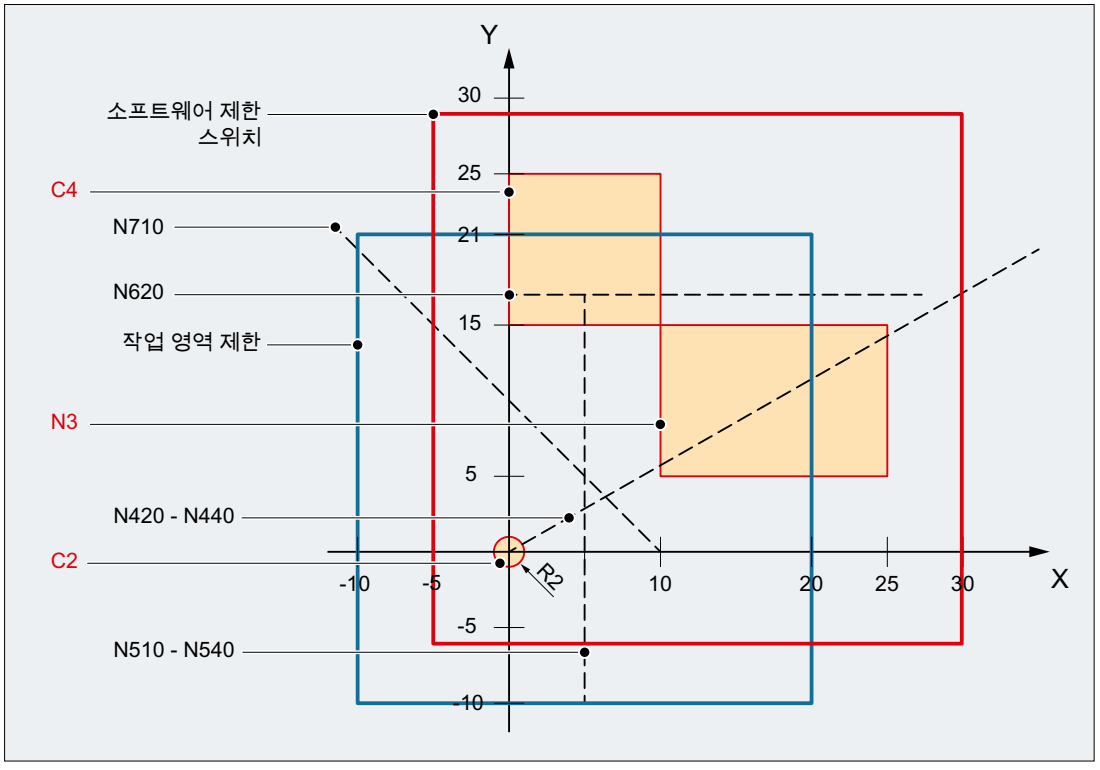
<MeasSys>:	위치 및 거리 지정에 사용되는 측정 시스템 (인치/미터) (**옵션**)		
	데이터 유형:		BOOL
	값:	FALSE (디폴트)	G그룹 13의 현재 활성 G 코드 (G70, G71, G700, G710)에 해당하는 장치 시스템 **참고** G70이 활성이고 베이직 시스템이 미터인 경우(또는 G71이 활성이고 베이직 시스템이 인치인 경우) 시스템 변수 $AA_IW 및 $AA_MW가 베이직 시스템에 제공되고, 이 변수를 사용하는 경우 CALCPOSI에 맞게 변환해야 합니다.
		TRUE	설정된 베이직 시스템에 따른 장치 시스템: MD52806 $MN_ISO_SCALING_SYSTEM

<TestLim>:	모니터링할 한계를 비트 코드로 선택 (**옵션**)		
	데이터 유형:		INT
	디폴트 값:		비트 0, 1, 2, 3, 6, 7 == 1 (207)
	비트	**십진**	**의미**
	0	1	소프트웨어 리미트 스위치
	1	2	작업 영역 제한
	2	4	활성화된 일반 보호 영역
	3	8	사전 활성화된 일반 보호 영역
	4	16	<Dist>에서 소프트웨어 리미트 스위치 또는 작업 영역 제한을 위반한 경우 **사례 1**처럼 이송 방향을 반환합니다 (위 설명 참조).
	5	32	<Dist>에서 소프트웨어 리미트 스위치 또는 작업 영역 제한을 위반한 경우 **사례 2**처럼 이송 방향을 반환합니다 (위 설명 참조).
	6	64	활성화된 충돌 방지 보호 영역
	7	128	사전 활성화된 충돌 방지 보호 영역
	8	256	활성화된 충돌 방지 보호 영역과 사전 활성화된 충돌 방지 보호 영역 쌍

[1] 여러 보호 영역을 위반한 경우 지정된 이송 거리에 대한 제한이 가장 큰 보호 영역이 반환됩니다.

예

제한조건



예제의 경우 X-Y 평면의 활성 소프트웨어 리미트 스위치 및 작업 영역 제한과 다음의 세 보호 영역이 표시됩니다:

- C2: 공구 관련, 채널 보호 영역, 활성, 원호, 반경 = 2 mm
- C4: 공작물 관련, 채널 보호 영역, 사전 활성, 정사각형, 변 길이 = 10 mm
- N3: 기계별 보호 영역, 활성, 직사각형, 변 길이 = 10 mm x 15 mm

NC 프로그램

먼저 보호 영역과 작업 영역 제한을 NC 프로그램에 정의합니다. 그런 다음 파라미터를 다르게 설정하여 CALCPOSI() 기능을 호출합니다.

프로그램 코드

```
N10 DEF REAL _START[3]
N20 DEF REAL _DIST[3]
N30 DEF REAL _LIMIT[5]
N40 DEF REAL _MAXDIST[3]
N50 DEF INT _PA
N60 DEF INT _STATUS
```

프로그램 코드

```
: 공구 관련 보호 영역C2
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
; 공작물 관련 보호 영역 C4
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
; 기계별 보호 영역 N3
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
; 보호 영역 활성화 또는 사전 활성화
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
; 작업 영역 제한 정의
N290 G25 XX=-10 YY=-10
N300 G26 XX=20 YY=21
N310 _START[0] = 0.
N320 _START[1] = 0.
N330 _START[2] = 0.
N340 _DIST[0] = 35.
N350 _DIST[1] = 20.
N360 _DIST[2] = 0.
N370 _LIMIT[0] = 0.
N380 _LIMIT[1] = 0.
N390 _LIMIT[2] = 0.
N400 _LIMIT[3] = 0.
N410 _LIMIT[4] = 0.
N420 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)
N430 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,3)
N440 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,1)
N450 _START[0] = 5.
N460 _START[1] = 17.
N470 _START[2] = 0.
```

프로그램 코드

```
N480 _DIST[0] = 0.
N490 _DIST[1] = -27.
N500 _DIST[2] = 0.
N510 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,14)
N520 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N530 _LIMIT[1] = 2.
N540 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N550 _START[0] = 27.
N560 _START[1] = 17.1
N570 _START[2] =  0.
N580 _DIST[0] =-27.
N590 _DIST[1] =  0.
N600 _DIST[2] =  0.
N610 _LIMIT[3] = 2.
N620 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,12)
N630 _START[0] = 0.
N640 _START[1] = 0.
N650 _START[2] = 0.
N660 _DIST[0] = 0.
N670 _DIST[1] = 30.
N680 _DIST[2] =  0.
N690 TRANS X10
N700 AROT Z45
N710 _STATUS = CALCPOSI(_START,_DIST, _LIMIT, _MAXDIST)
; N690 및 N700에서 프레임을 다시 삭제
N720 TRANS
N730 _START[0] =  0.
N740 _START[1] = 10.
N750 _START[2] =  0.
; vectors_DIST 및 _MAXDIST가 이동 평면을 정의합니다.
N760 _DIST[0] = 30.
N770 _DIST[1] = 30.
N780 _DIST[2] =  0.
N790 _MAXDIST[0] = 1.
N800 _MAXDIST[1] = 0.
N810 _MAXDIST[2] = 1.
N820 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,17)
N830 M30
```

CALCPOSI()의 결과

N...	<status>	<MaxDist>[0] ≙ X	<MaxDist>[1] ≙ Y	추가 설명
420	3123	8.040	4.594	N3을 위반합니다.
430	1122	20.000	11.429	보호 영역 모니터링이 없으며 작업 영역 제한이 위반되었습니다.
440	1121	30.000	17.143	소프트웨어 제한 모니터링만 여전히 활성화된 상태입니다.
510	4213	0.000	0.000	시작점이 C4를 위반합니다.
520	0000	0.000	-27.000	사전 활성화된 C4를 모니터링하지 않습니다. 지정된 거리를 완전히 이송할 수 있습니다.
540	2222	0.000	-25.000	_DLIMIT[1] = 2이기 때문에 이송 거리가 작업 영역 제한에 의해 제한됩니다.
620	4223	-13.000	0.000	C4에 대한 여유간격은 C2 및 _DLIMIT[3]으로 인해 총 4mm입니다. C2와 N3 사이의 여유간격 0.1mm는 이송 거리 제한을 초래하지 않습니다.
710	1221	0.000	21.213	변환과 회전이 활성인 프레임. _DIST에서 허용 가능한 이송 거리는 WCS를 이동 및 회전시켜 적용합니다.
820	102121	18.000	18.000	Y 축의 소프트웨어 리미트 스위치를 위반합니다. 추가 이송 방향의 계산은 <_TESTLIM> = 17를 이용하여 요청됩니다. 이 방향은 _DIST (0.707, 0.0, 0.707)에 있습니다. 이 값은 <_STATUS>가 십만 단위 숫자로 설정되어 있기 때문에 유효합니다.

추가 정보

"참조된" 축 상태

CALCPOSI()에 적용되는 모든 기계 축은 호밍시켜야 합니다.

원호 관련 거리 지정

모든 원호 관련 거리 지정은 **항상** 반경 지정으로 해석됩니다. 특히 직경 프로그래밍 (DIAMON/DIAM90) 이 활성화된 가로 축의 경우 이 점을 반드시 고려해야 합니다.

이송 거리 감소

지정된 축의 이송 거리가 제한되는 경우 <MaxDist> 반환값으로 출력되는 다른 축들의 이송 거리 역시 그에 비례하여 감소합니다. 따라서 종점은 여전히 지정된 경로 상에 있게 됩니다.

로터리 축

로터리 축은 모듈로 로터리 축이 아닌 경우에만 모니터링됩니다.

하나 이상의 관련 축에 대해 소프트웨어 리미트 스위치, 작업 영역 제한 또는 보호 영역이 정의되지 않을 수 있습니다.

소프트웨어 리미트 스위치 및 작업 영역 제한 상태

소프트웨어 리미트 스위치와 작업 영역 제한은 CALCPOSI() 실행 중에 활성인 경우에만 적용됩니다. 이들의 상태는 다음과 같은 방법으로 변경할 수 있습니다.

- 머신 데이터: MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS
- 설정 데이터: $AC_WORKAREA_CS_...
- NC/PLC 인터페이스 신호 DB31, ... DBX12.2 / 3
- 명령: WALIMON / WALIMOF

소프트웨어 리미트 스위치 및 변환

CALCPOSI() 를 사용할 때 이송 거리의 일부 위치가 불확실하기 때문에 다양한 좌표계 변환 (예: TRANSMIT) 중 기하 축의 위치(WCS)에서 기계 축의 위치 (MCS)를 정확히 알기 어려운 경우가 있습니다. 일반적인 이송 동작 시 보통 WCS에서의 연속 모션은 MCS에서의 연속 모션과 일치해야 한다는 조건과 이전 이력에 의해 위치의 고유성이 확보됩니다. 따라서 소프트웨어 리미트 스위치를 모니터링할 때 CALCPOSI() 가 실행되는 시점의 기계 위치를 사용하면 정확한 위치를 알 수 있습니다.

참고

전처리 정지

CALCPOSI() 를 변환과 함께 사용하는 경우 기계 축 위치 동기화를 위해 CALCPOSI() 전에 전처리와 전처리 정지 (STOPRE)를 프로그래밍할 책임은 전적으로 사용자에게 있습니다.

보호 영역 간격 및 일반 보호 영역

일반 보호 영역의 경우 지정된 경로를 따라 이송 동작을 수행하는 동안 모든 보호 영역에서 파라미터 <Limit>[3] 에 설정된 안전 거리가 지켜진다는 보장은 **없습니다**. <Dist>에 반

환된 종점이 이송 방향으로 안전 거리만큼 확장될 때 어떤 보호 영역도 위반하지 않는다는 것만 보장됩니다. 단, 직선은 보호 영역에 거의 근접하여 통과할 수 있습니다.

보호 영역 간격 및 충돌 방지 보호 영역

충돌 방지 보호 영역의 경우 지정된 경로를 따라 이송 동작을 수행하는 동안 모든 보호 영역에서 파라미터 <Limit>[3]에 설정된 안전 거리가 지켜집니다.

파라미터 <Limit>[3]에 지정된 안전 거리는 다음 경우에 한해 적용됩니다.

<Limit>[3] > (MD10619 $MN_COLLISION_TOLERANCE)

비트 4가 파라미터 <TestLim>(진행중인 이송 방향의 계산)에 설정된다면, <DIST>에 수신된 방향 벡터는 십만 단위 숫자가 기능 반환 값(<status>)에 설정된 경우에만 유효합니다. 보호 영역이 위반되었거나 변환이 활성화된 상태여서 방향을 결정할 수 없는 경우에는 <DIST> 에서의 입력값은 변하지 않습니다. 추가 에러 메시지가 출력되지 않습니다.

4.7 특수 이동 명령

4.7.1 코드 위치 복귀 (CAC, CIC, CDC, CACP, CACN)

경로 명령을 사용하여 '코딩된 위치로 접근'하면 위치 번호를 지정하여 머신 데이터 테이블에 저장된 고정 축 위치로 선형 및 로터리 축을 이송할 수 있습니다.

구문

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

의미

CAC(<n>):	위치 수 n에서 코드 위치로 복귀합니다.
CIC(<n>):	실제 위치 수부터 시작하여 코드 위치를 이전(+n) 또는 뒤로(-n) n 위치로 복귀합니다.
CDC(<n>):	최단 경로를 따라 위치 번호 n에서 해당 위치로 이동 (로터리 축에만 해당).
CACP(<n>):	양의 방향으로 위치 번호 n에서 코드 위치로 이동 (로터리 축에만 해당).

CACN(<n>):	음의 방향으로 위치 번호 n에서 코드 위치로 이동 (로터리 축에만 해당).
<n>:	머신 데이터 테이블 내의 위치 수 값 범위: 0, 1, ... (테이블 위치 최대값 - 1)

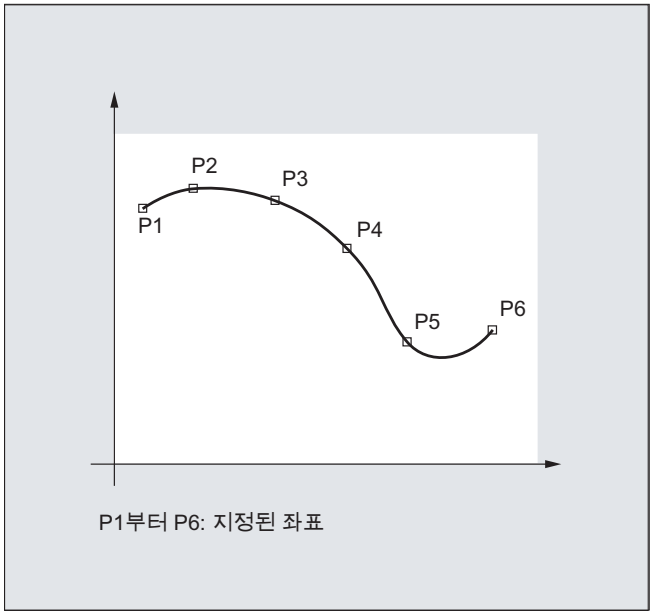
예: 위치 지정 축의 코드 위치 복귀

프로그램 코드	설명
N10 FA[B]=300	; 포지셔닝 축 B의 이송 속도
N20 POS[B]=CAC(10)	; 위치 번호 10에서 코드 위치로 이동
N30 POS[B]=CIC(-4)	; "현재 위치 번호" - 4에서 코드 위치로 이동

4.7.2 스플라인 보간 (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

랜덤 곡선 공작물 형상은 분석 양식으로 정밀하게 정의할 수 없습니다. 이러한 유형의 형상은 예를 들어 표면을 디지털화할 때 곡선 위의 몇몇 점을 사용하여 근사치로 표시하기 때문입니다. 공작물의 디지털화된 표면을 생성하려면 곡선을 따라 점을 연결하여 형상을 정의해야 합니다. 스플라인 보간은 이 작업을 허용합니다.

스플라인은 2차 또는 3차 폴리노미얼을 통해 형성되는 곡선을 정의합니다. 스플라인 곡선을 따라 형성된 점의 특성은 **사용 중인 스플라인 유형에 따라** 정의할 수 있습니다.



P1부터 P6: 지정된 좌표

SINUMERIK 솔루션 라인의 경우 다음 스플라인 유형을 사용할 수 있습니다:

- A 스플라인
- B 스플라인
- C 스플라인

구문

일반:

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

B 스플라인의 경우 다음을 추가로 프로그래밍할 수 있습니다:

```
PW=<n>
SD=2
PL=<value>
```

A 및 C 스플라인의 경우 다음을 추가로 프로그래밍할 수 있습니다:

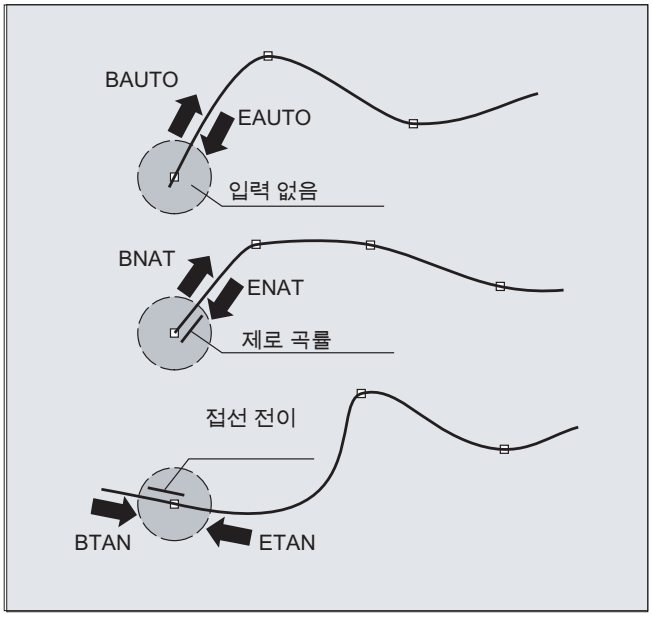
BAUTO / BNAT / BTAN

EAUTO / ENAT / ETAN

의미

스플라인 보간 유형:	
ASPLINE:	A 스플라인 보간을 활성화하기 위한 명령
BSPLINE:	B 스플라인 보간을 활성화하기 위한 명령
CSPLINE:	C 스플라인 보간을 활성화하기 위한 명령
	ASPLINE, BSPLINE 및 CSPLINE 명령은 모달식으로 적용되며 이동 명령 그룹에 속합니다.
곡선을 따라 점 및 확인 점:	
X... Y... Z... A... B... C...	직교 좌표계에서의 위치
점 가중치(B 스플라인만 해당):	
PW:	PW 명령을 사용하면 소위 곡선을 따라 모든 점에 대해 "점 가중치"를 프로그래밍할 수 있습니다.

<n>:	"점 가중치"		
	값 범위:	0 ≤ n ≤ 3	
	증분:	0.0001	
	결과:	n > 1	확인 점은 곡선을 상당히 많이 끌어당깁니다.
		n < 1	확인 점은 곡선을 상당히 적게 끌어당깁니다.
스플라인 차수(B 스플라인만 해당):			
SD:	3차 다각형이 표준으로 사용되지만 SD=2를 프로그래밍하면 2차 다각형도 사용할 수 있습니다.		
노드 사이 거리(B 스플라인만 해당):			
PL :	노드 사이 거리는 내부적으로 적절하게 계산됩니다. 또한 제어 시스템은 PL 명령을 사용하여 소위 파라미터-간격-길이에서 지정된 사전 정의된 노드 간극을 가공할 수 있습니다.		
<값>:	파라미터 간격 길이		
	값 범위:	경로 치수	
스플라인 곡선 시작 위치에서 진행 동작(A 또는 C 스플라인만 해당):			
BAUTO:	진행 동작을 지정하지 않습니다. 시작은 첫 번째 점의 위치에 의해 결정됩니다.		
BNAT:	제로 곡률		
BTAN:	이전 블록으로 접선 트렌지션 (위치 삭제)		
스플라인 곡선 끝 위치에서 진행 동작(A 또는 C 스플라인만 해당):			
EAUTO:	진행 동작을 지정하지 않습니다. 끝은 마지막 점의 위치에 의해 결정됩니다.		
ENAT:	제로 곡률		

ETAN:	이전 블록으로 접선 트렌지션 (위치 삭제)
	

참고

프로그램 가능한 진행 동작은 B 스플라인에는 영향을 주지 않습니다. B 스플라인은 항상 시작점 및 끝점에서 확인 다각형에 대해 접선 방향입니다.

보충 조건

- 공구 반경 보정을 사용할 수 있습니다.
- 평면 돌출부에서 충돌 모니터링이 실행됩니다.

예

예 1: B 스플라인

프로그램 코드 1(모두 가중치 1)

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
```

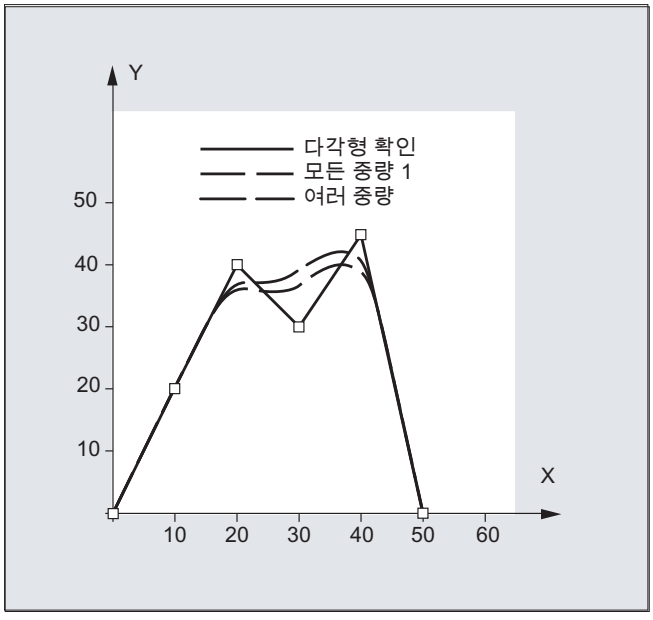
프로그램 코드 1(모두 가중치 1)

```
N70 X50 Y0
```

프로그램 코드 2(가중치가 서로 다름)

```
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
```

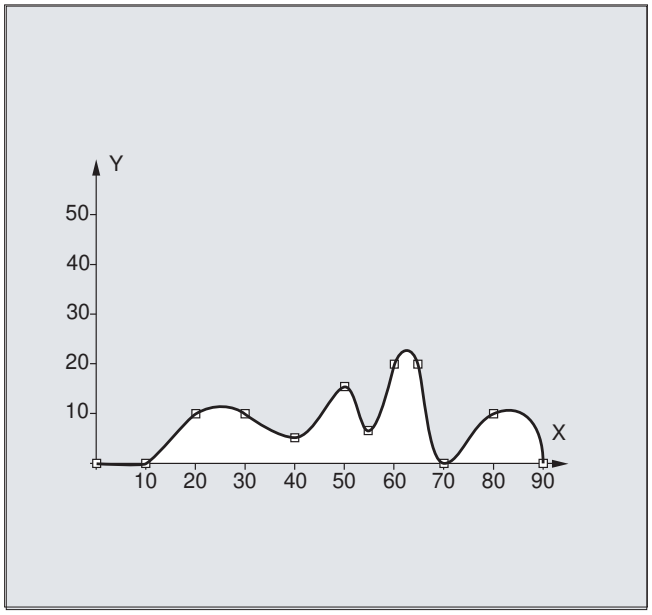
프로그램 코드 3(확인 다각형)	코멘트
N10 G1 X0 Y0 F300 G64	
N20	; n.a.
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	



예 2: C 스플라인, 시작 및 끝 위치에서 제로 곡률

프로그램 코드

```
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
```



예 3: 스플라인 보간 (A 스플라인) 및 좌표 변환 (ROT)

메인 프로그램:

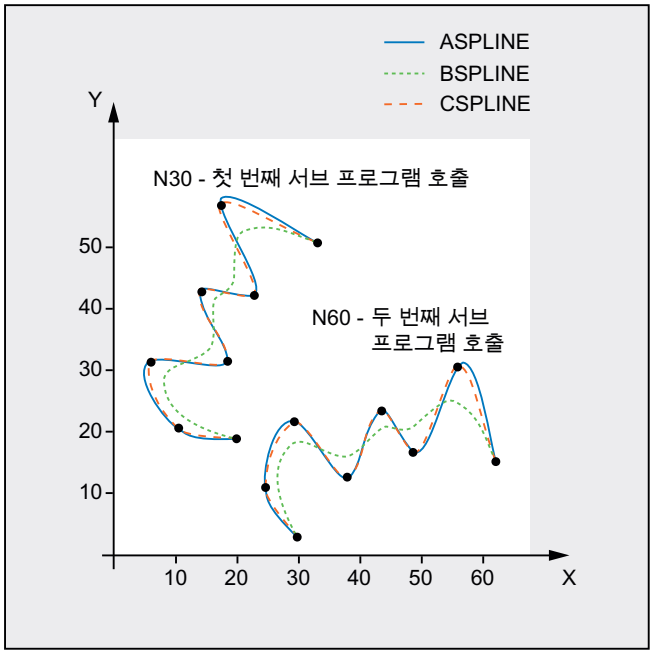
프로그램 코드	코멘트
N10 G00 X20 Y18 F300 G64	; 시작점으로 접근.
N20 ASPLINE	; 보간 유형 A 스플라인 활성화.
N30 CONTOUR	; 첫 번째 서브프로그램 호출.

프로그램 코드	코멘트
N40 ROT Z-45	; 좌표 변환: Z축 주위를 -45° 각도로 WCS 회전.
N50 G00 X20 Y18	; 형상 시작점으로 복귀.
N60 CONTOUR	; 두 번째 서브프로그램 호출.
N70 M30	; 프로그램 종료

서브프로그램 "형상"(곡선을 따라 점에 대한 좌표 포함):

프로그램 코드
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

스플라인 곡선 이외에 예제 프로그램 (ASPLINE)을 실행하면, 다음 그림과 같이 B 또는 C 스플라인 보간(BSPLINE, CSPLINE)을 활성화할 때 가져온 스플라인 곡선도 포함됩니다.



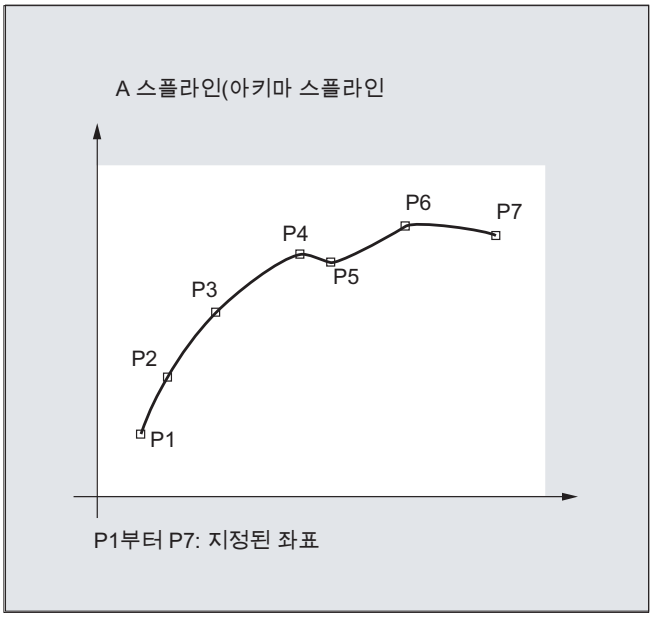
추가 정보

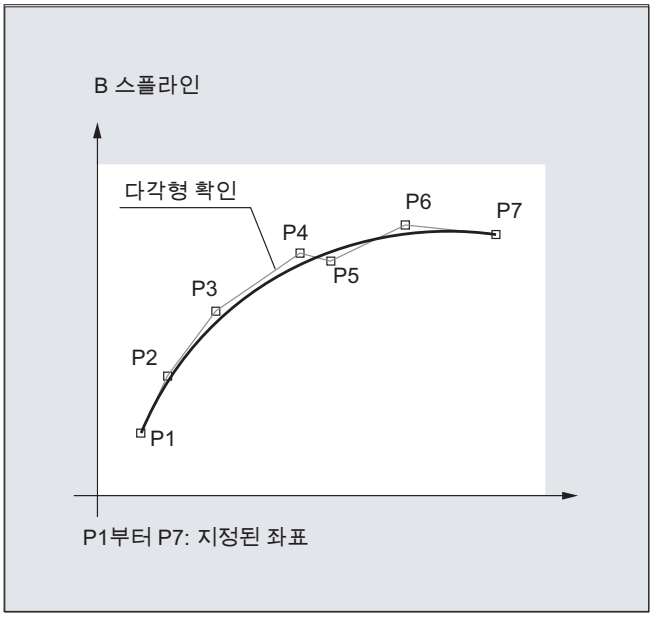
스플라인 보간의 이점

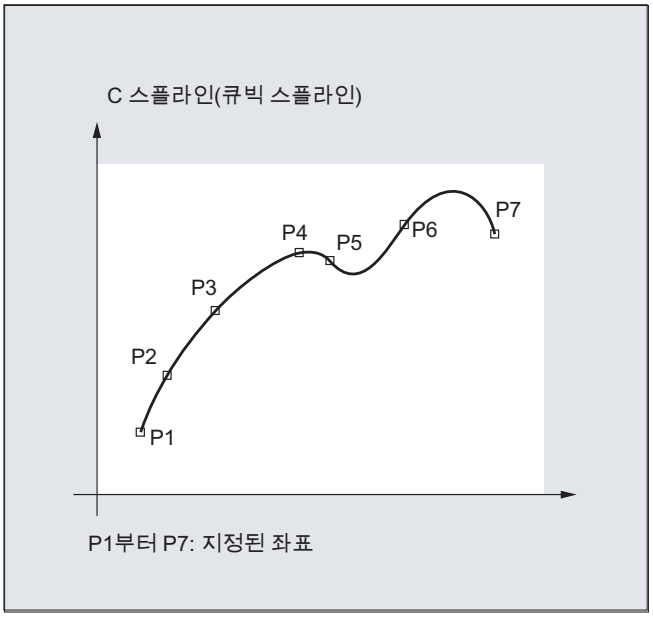
스플라인 보간을 사용하면 직선 블록 G01과 달리 다음과 같은 장점이 있습니다.

- 형상을 설명하는 데 필요한 가공 프로그램 블록 수 감소.
- 가공 프로그램 블록 간의 이동 시 기계 시스템의 응력을 줄이는 부드러운 곡선 특성.

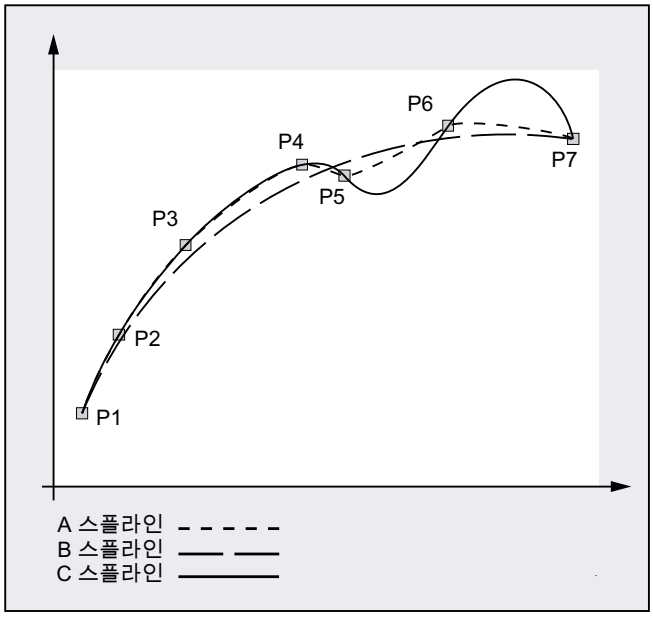
속성 및 다양한 스플라인 유형 사용

스플라인 유형	속성 및 사용
A 스플라인	 **속성:** • 곡선을 따라 지정된 중간 지점들을 정확하게 통과합니다. • 곡선 특성은 접선이지만 연속 곡률이 없습니다. • 원치 않는 오실레이션이 거의 발생하지 않습니다. • 곡선을 따라 중간 지점에 대한 변경에 영향을 받는 영역은 제한적입니다. 즉, 곡선을 따라 중간 지점에 대한 변경은 최대 6개의 인접한 중간 지점에만 영향을 줍니다. **적용:** A 스플라인은 특히 기울기가 상당히 변경되는 보간 곡선에 적합합니다(예를 들어 계단식 곡선 및 특성).

스플라인 유형	속성 및 사용
B 스플라인	 **속성:** • 곡선을 따라 지정된 중간 지점을 통해 진행하지 말고 근접한 위치에서 진행하십시오. 중간 지점은 곡선을 끌어 당기지 않습니다. 곡선 특성은 인자를 사용하여 중간 지점에 가중치를 부과하는 방법으로 추가로 영향을 줄 수 있습니다. • 곡선 특성은 연속 곡률을 통해 접선 방향입니다. • 원하지 않는 오실레이션을 생성하지 않습니다. • 곡선을 따라 중간 지점에 대한 변경에 영향을 받는 영역은 제한적입니다. 즉, 곡선을 따라 중간 지점에 대한 변경은 최대 6개의 인접한 중간 지점에만 영향을 줍니다. **적용:** B 스플라인은 주로 CAD 시스템에 대한 인터페이스로 사용됩니다.

스플라인 유형	속성 및 사용
C 스플라인	 **속성:** • 곡선을 따라 지정된 중간 지점들을 정확하게 통과합니다. • 곡선 특성은 연속 곡률을 통해 접선 방향입니다. • 특히 기울기가 상당히 변경되는 위치에서 원하지 않는 오실레이션이 자주 생성됩니다. • 중간 지점에 대한 변경에 대해 전체 영역이 영향을 받습니다. 즉, 중간 지점이 변경되면 전체 곡선 특성에 영향을 줍니다. **적용:** 중간 지점이 분석을 통해 정의된 곡선 (원호, 포물선, 쌍곡선) 위에 위치한 경우 C 스플라인을 사용할 수 있습니다.

동일한 보간 지점을 가진 세 가지 스플라인 유형의 비교



최소 스플라인 블록 수

G 코드 ASPLINE, BSPLINE 및 CSPLINE은 블록 끝점을 스플라인과 연결합니다. 이를 위해 일련의 블록(끝점)이 동시에 계산되어야 합니다. 계산을 위한 버퍼 크기의 기본값은 10개 블록입니다. 모든 블록 정보가 스플라인 종점은 아닙니다. 하지만 제어 시스템은 10개 블록 당 특정 수의 스플라인 종점 블록을 필요로 합니다.

스플라인 유형	최소 스플라인 블록 수
A 스플라인:	10개 블록 중 최소한 **4**개 블록이 스플라인 블록이어야 합니다. 여기서 설명 블록 또는 파라미터 계산은 제외됩니다.
B 스플라인:	10개 블록 중 최소한 **6**개 블록이 스플라인 블록이어야 합니다. 여기서 설명 블록 또는 파라미터 계산은 제외됩니다.
C 스플라인:	필요한 최소 스플라인 블록 수는 다음 합계와 같습니다. MD20160 $MC_CUBIC_SPLINE_BLOCKS + 1의 값 스플라인 세그먼트를 계산하기 위한 지점 개수를 MD20160에 입력합니다. 기본 설정은 8%입니다. 10개 블록 중 최소한 **9** 개 블록이 스플라인 블록이어야 합니다.

참고

허용 오차 값이 더 작고 스플라인의 관련 축 중 하나가 위치 지정 축으로 프로그래밍된 경우 알람이 출력됩니다.

짧은 스플라인 블록 조합

스플라인 보간을 통해 짧은 스플라인 블록을 만들 수 있으며 이로 인해 불필요한 경로 속도를 줄입니다. "짧은 스플라인 블록 조합" 기능을 통해 결과 블록 길이가 충분하도록 이러한 블록을 조합할 수 있으며 경로 속도가 줄어 들지 않습니다.

이 기능은 채널 관련 머신 데이터를 통해 활성화됩니다.

MD20488 $MC_SPLINE_MODE(스플라인 보간 설정).

추가 정보: 기능 매뉴얼 기본 기능

4.7.3 스플라인 그룹 (SPLINEPATH)

스플라인 그룹에서 보간할 축은 SPLINEPATH 명령을 사용하여 선택합니다. 스플라인 보간 그룹화에 최대 8개의 축을 사용할 수 있습니다.

참고

SPLINEPATH가 명시적으로 프로그래밍되지 않을 경우 채널의 세 개 축 중 첫 번째 축이 스플라인 그룹으로 이송됩니다.

구문

스플라인 그룹이 개별 블록에 정의됩니다:

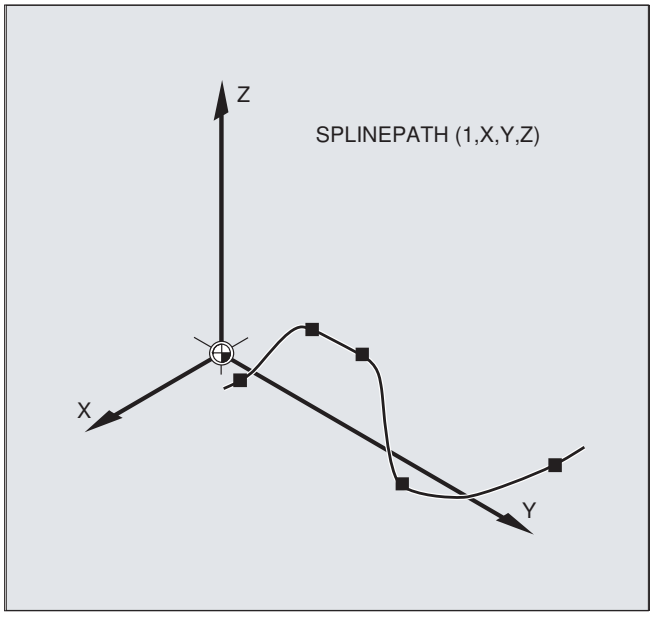
```
SPLINEPATH(n,X,Y,Z,…)
```

의미

SPLINEPATH:	스플라인 그룹을 정의하기 위한 명령
n:	=1 (고정값)
X,Y,Z,… :	스플라인 그룹에서 보간할 경로 축의 식별자

예: 세 개의 경로 축을 가진 스플라인 그룹

프로그램 코드	코멘트
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH(1,X,Y,Z)	; 스플라인 그룹
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	; C 스플라인
N14 X30 Y40 Z50 A60 B70	; 중간점
…	
N100 G1 X… Y…	; 스플라인 보간 선택 취소



4.7.4 NC 블록 압축 (COMPON, COMPCURV, COMPCAD, COMPOF) 활성화/비활성화

선형 블록을 압축하는 기능 (파라미터화에 의존하는, 원형 및/또는 급이송 블록도 포함)은 G 그룹 30의 G 명령어를 사용하여 활성화/비활성화됩니다. 명령은 모달식입니다.

구문

```
COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF
```

의미

COMPON:	컴프레서 기능 COMPON 활성화
COMPCURV:	컴프레서 기능 COMPCURV 활성화
COMPCAD:	컴프레서 기능 COMPCAD 활성화
COMPSURF:	컴프레서 기능 COMPSURF 활성화
COMPOF:	현재 활성 상태인 컴프레서 기능 비활성화

참고

스무딩 기능 G642 및 저크 제한 SOFT는 표면 품질을 추가로 향상시킵니다. 이러한 명령은 프로그램 시작 시 기록해야 합니다.

예: COMPCAD

프로그램 코드	설명
N10 G00 X30 Y6 Z40	
N20 G1 F10000 G642	; 활성화: 스무딩 기능 G642
N30 SOFT	; 활성화: 저크 한계 SOFT
N40 **COMPCAD**	; 활성화: 컴프레서 함수 COMPCAD
N50 FIFOCTRL	
N24050 Z32.499	; 첫 번째 이송 블록
N24051 X41.365 Z32.500	; 두 번째 이송 블록
...	
N99999 X... Z...	; 마지막 이송 블록
COMPOF	; 컴프레서 기능 꺼짐.
...	

4.7.5 폴리노미얼 보간 (POLY, POLYPATH, PO, PL)

실제로 스플라인 보간이 아닌 폴리노미얼 보간 (PLLY) 이 사용됩니다. 폴리노미얼 보간의 주요 목적은 스플라인 섹션을 직접 프로그래밍할 수 있도록 외부에서 생성된 스플라인 곡선을 프로그래밍하기 위한 인터페이스 역할을 수행하는 것입니다.

이 보간 모드는 폴리노미얼 계수를 계산하는 NC 작업의 부담을 덜어 줍니다. 계수가 CAD 시스템 또는 다음 프로세서에 의해 직접 제공되는 경우 최적화될 수 있습니다.

구문

3차 폴리노미얼:

```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

5차 폴리노미얼 및 새로운 폴리노미얼 구문:

```
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")
```

참고

NC 블록에 프로그래밍된 축과 폴리노미얼 계수의 합은 블록당 허용된 최대 축 수를 초과하지 않아야 합니다.

의미

`POLY` :	POLY가 포함된 블록을 통한 폴리노미얼 보간 활성화.
`POLYPATH` :	AXIS 축 그룹 그리고 VECT 축 그룹 모두에 폴리노미얼 보간 선택 가능
`PO[축 이름/변수]` :	끝점 및 폴리노미얼 계수
`X, Y, Z`:	축 이름
`xe, ye, ze` :	특정 축의 끝 위치 지정; 경로 차원에 대한 값 범위
`a2, a3, a4, a5` :	계수 a_2, a_3, a_4 및 a_5는 경로 차원에 대한 값 범위에서 해당 값이 기록됩니다. 각 경우에서 마지막 계수가 0인 경우 생략될 수 있습니다.
`PL` :	폴리노미얼이 정의된 파라미터 간격 길이(함수 f(p)의 정의 범위). 간격은 항상 0부터 시작하며 p는 0-PL 사이의 값으로 간주할 수 있습니다. PL의 이론 값 범위: 0.0001 ... 99 999.9999 **참고:** PL 값은 이 값이 위치한 블록에 적용됩니다. PL을 프로그래밍하지 않을 경우 PL=1이 적용됩니다.

폴리노미얼 보간 활성화/비활성화

`POLX` G 명령을 사용하여 가공 프로그램에서 폴리노미얼 보간을 활성화합니다.

POLY G 명령은 G0, G1, G2, G3, ASPLINE, BSPLINE 및 CSPLINE과 함께 그룹 1에 속합니다.

이름과 종점만 프로그래밍된 축 (예: X10) 은 직선으로 움직입니다. NC 블록의 모든 축이 이렇게 프로그래밍되면 제어 시스템은 G1의 경우와 동일하게 동작합니다.

첫 번째 G 그룹 (G0, G1) 에 다른 명령을 프로그래밍하면 폴리노미얼 보간이 암시적으로 다시 비활성화됩니다.

폴리노미얼 계수

PO 값 (PO[]=) 또는 ...=PO(...) 는 축의 모든 폴리노미얼 계수를 지정합니다. 값이 여러 개인 경우 폴리노미얼 차수에 해당하는 쉼표로 구분되어 지정됩니다. 한 블록 내에서 여러 축에 대해 서로 다른 폴리노미얼 차수가 가능합니다.

POLYPATH 서브프로그램

POLYPATH(...) 를 사용하면 특정 축 그룹에 대한 폴리노미얼 보간을 선택적으로 해제할 수 있습니다.

경로 축 및 보조 축만 해제하는 경우:	POLYPATH("AXES")
오리엔테이션 축만 해제하는 경우: (오리엔테이션 변환을 적용하여 이동할 때)	POLYPATH("VECT")

해제되지 않은 축은 직선으로 움직입니다.

폴리노미얼 보간은 기본적으로 축 그룹 모두에 활성화되어 있습니다.

POLYPATH( ) 파라미터없이 프로그래밍하면 모든 축에 대한 폴리노미얼 보간을 비활성화할 수 있습니다.

예

프로그램 코드	코멘트
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	; 폴리노미얼 보간 ON
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	; 축의 데이터 혼합
N27 PO[X]=(10,2.5) PO[Y]=(2.3)	; 프로그램된 PL 없음, PL=1 적용
N30 G1 X… Y… Z.	; 폴리노미얼 보간 OFF

프로그램 코드	코멘트
…	

예: 새로운 폴리노미얼 구문

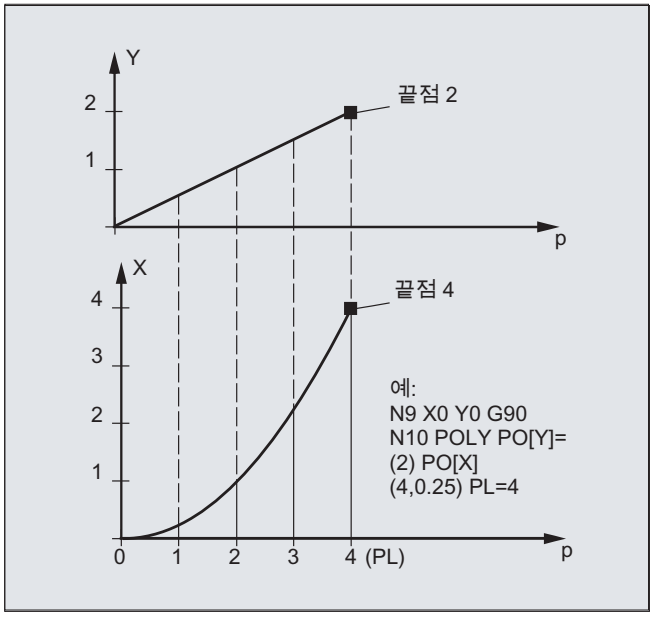
아직 유효한 폴리노미얼 구문	새로운 폴리노미얼 구문
PO[axis identifier]=(.. , ..)	Axis identifier=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[variable]=IC(.. , ..)	variable=PO IC(.. , ..)

예: X/Y 평면의 곡선.

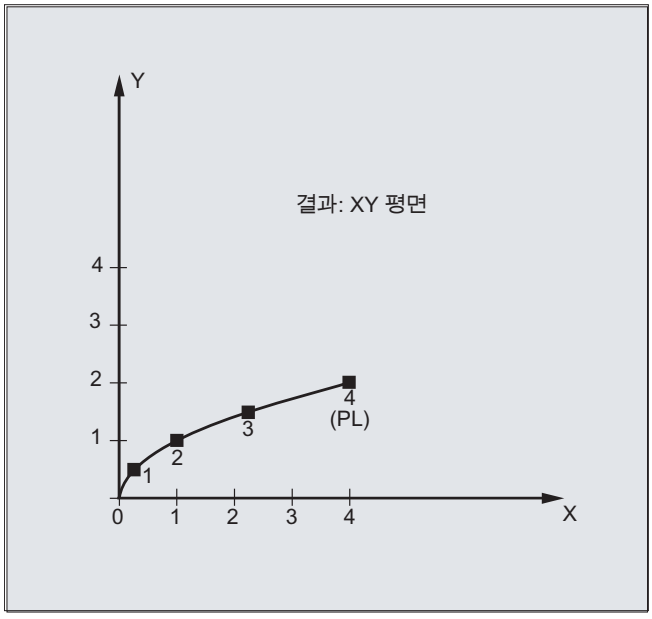
프로그래밍

프로그램 코드
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

곡선 X(p) 및 Y(p)의 형태



XY 평면의 곡선 형태



설명

폴리노미얼 함수를 표현하는 방정식은 일반적으로 다음과 같습니다.

$$f(p)= a_0 + a_1p + a_2p^2 + . . . + a_np^n$$

여기 a_i: 상수 계수 (i = 0, 1, ..., n)
서 p: 파라미터

제어 시스템에 최대 5차 폴리노미얼까지 프로그래밍할 수 있습니다.

$$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

특정 값을 이러한 계수에 할당하면 선, 포물선 및 제곱 함수와 같은 다양한 곡선 형태를 만들 수 있습니다.

직선은 $a_2 = a_3 = a_4 = a_5 = 0$으로 만듭니다.

$$f(p) = a_0 + a_1p$$

다음 내용이 그대로 적용됩니다.

$a_{0:}$ 선행 블록의 끝에서 축 위치

$p = PL$

$a_1 = (x_E - a_0 - a_2*p^2 - a_3*p^3)/p$

G 명령 POLY를 사용하면 폴리노미얼 보간을 활성화하지 **않고** 폴리노미얼을 프로그램할 수 있습니다. 이 경우 프로그래밍된 폴리노미얼은 보간되지 않는 대신 축에 프로그래밍된 종점은 직선으로 복귀합니다 (G1). 프로그래밍된 폴리노미얼은 가공 프로그램 (POLY) 에서 폴리노미얼 보간을 명시적으로 활성화한 후에만 이동됩니다.

특징: 분모 폴리노미얼

PO[]=(...) 명령을 사용하여 축 이름을 지정하지 않고 기하 축의 공통 분모 폴리노미얼을 프로그래밍할 수 있습니다. 예를 들어 형상 축은 두 폴리노미얼의 몫으로 보간됩니다.

이 프로그래밍 옵션을 통해 원뿔 (원호, 타원, 포물선, 쌍곡선) 과 같은 형태를 정확하게 표현할 수 있습니다.

예:

프로그램 코드	코멘트
`POLY G90 X10 Y0 F100`	`; 기하 축이 X10 Y0 위치로 직선으로 이송됩니다.`
`PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)`	`; 기하 축이 4분원을 따라 X0 Y10으로 이송됩니다.`

분모 폴리노미얼의 상수 계수(a_0)는 항상 1로 추정됩니다. 프로그램 된 종점은 G90 / G91 으로부터 독립되어 있습니다.

다음과 같이 프로그래밍된 값에서 X(p) 및 Y(p)를 계산합니다.

$X(p) = (10 - 10 * p^2) / (1 + p^2)$

$Y(p) = 20 * p / (1 + p^2)$

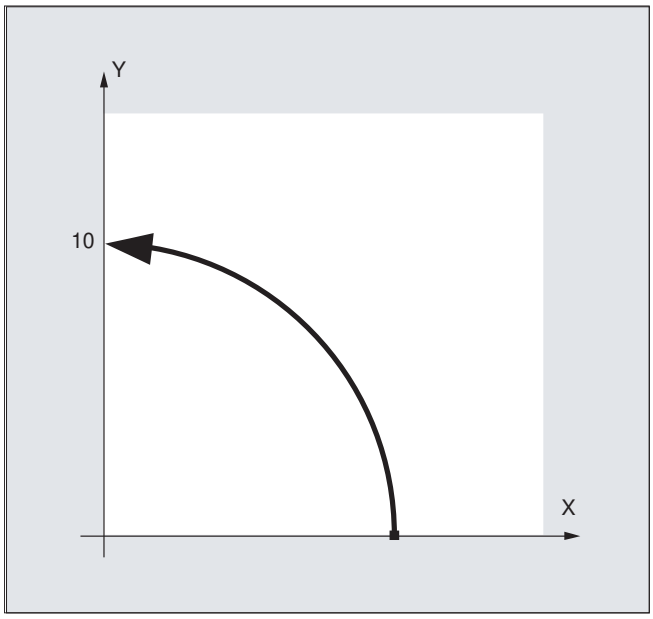
$0 \le p \le 1$

프로그래밍된 시작점, 종점, 계수 a_2 및 PL=1을 적용한 중간 값은 다음과 같습니다.

분자 (X) = $10 + 0 * p - 10 * p^2$

분자 (Y) = $0 + 20 * p + 0 * p^2$

분모 = $1 + p^2$



폴리노미얼 보간이 활성화되고 분모 폴리노미얼이 간격 [0,PL] 내에서 0으로 프로그래밍된 경우 이 조건은 거부되고 알람이 출력됩니다. 분모 폴리노미얼은 특수 축의 이동에 영향을 주지 않습니다.

참고

공구 반경 보정은 폴리노미얼 보간과 함께 G41, G42를 통해 활성화할 수 있으며 선형 또는 원호 보간 모드와 동일한 방법으로 적용할 수 있습니다.

4.7.6 설정 가능한 경로 기준 (SPATH, UPATH)

폴리노미얼 보간(POLY, ASPLINE, BSPLINE, CSPLINE, COMP...)의 경우 경로 축 i의 위치가 폴리노미얼 $p_i(U)$로 지정됩니다. 커브 파라미터 U는 NC 블록 내에서 0에서 1로 이동합니다.

FGROUP은 경로 이송 속도 F가 적용되는 축 (FGROUP 축) 을 선택합니다. FGROUP 축의 경로 S에서 일정 속도로 보간한다는 것은 폴리노미얼 보간 중에 보통 커브 파라미터 U가 일정하지 않게 변한다는 뜻입니다. 따라서 FGROUP에 속하지 않은 축이 FGROUP 축을 따르게 하는 방법은 두 가지가 있습니다.

- 경로 S와 동기화 (SPATH)
- 커브 파라미터 U와 동기화 (UPATH)

구문

```
SPATH
UPATH
```

의미

SPATH:	FGROUP에 속하지 않은 축을 경로 S에 따라 이송
UPATH:	FGROUP에 속하지 않은 축을 커브 파라미터 U에 따라 이송

참고

또한 UPATH 및 SPATH는 경로 이동을 통해 F 워드 폴리노미얼의 상호 관계를 정의합니다 (FPOLY, FCUB, FLIN).

한계 조건

SPATH와 UPATH는 다음의 경우 아무 의미가 없습니다.

- 직선 보간 (G1)
- 원호 보간(G2, G3)
- 나사 블록 (G33, G34, G35, G33x, G63)
- 모든 경로 축이 FGROUP에 속한 경우

예

다음 예에서는 두 가지 이동 제어 유형 간의 차이점을 보여 줍니다.

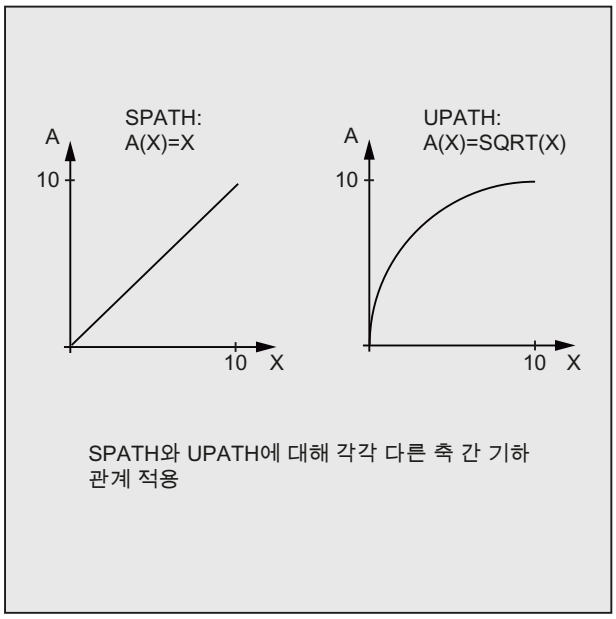
프로그램 코드

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 SPATH                ; SPATH
N20 POLY PO[X]=(10,10) A10
```

프로그램 코드

```
N10 FGROUP(X,Y,Z)
N15 G1 X0 A0 F1000 UPATH                ; UPATH
N20 POLY PO[X]=(10,10) A10
```

두 프로그램 섹션 모두에서 N20의 FGROUP 축 경로 S는 곡선 파라미터 U의 제곱에 따라 달라집니다. 따라서 SPATH와 UPATH 중 어느 것이 활성인지에 따라 경로 X를 따라 이동하는 동기 축 A의 위치가 달라집니다.



SPATH와 UPATH에 대해 각각 다른 축 간 기하 관계 적용

추가 정보

리셋 및 기계/옵션 데이터에 대한 제어 동작

MD20150 $MC_GCODE_RESET_VALUES[44]로 정의된 G 코드는 리셋 후에 적용됩니다(45번째 G 코드 그룹).

스무딩 유형의 초기 상태는 MD20150 $MC_GCODE_RESET_VALUES[9](10번째 G 코드 그룹)에 의해 정의됩니다.

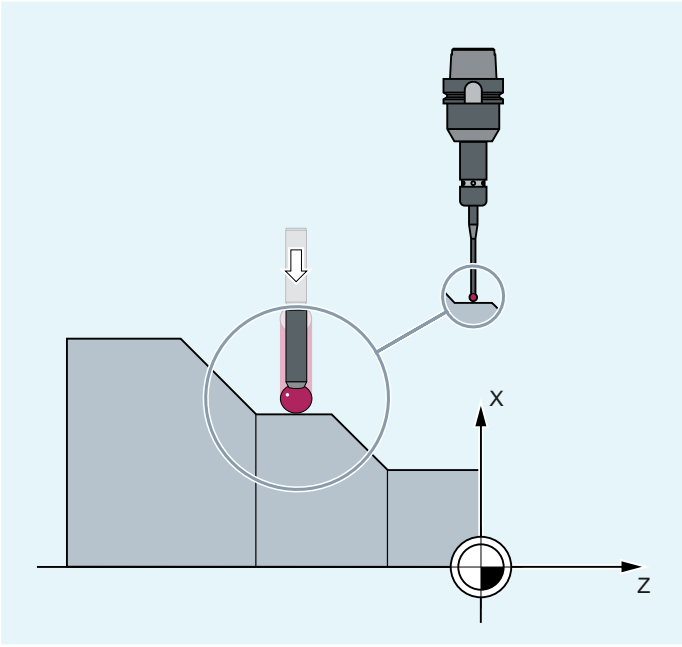
축 머신 데이터 MD33100 $MA_COMPRESS_POS_TOL[<n>]은 확장된 의미를 지니고 있습니다. 이 데이터는 컴프레서 기능 및 G642을 사용한 스무딩에 대한 공차를 포함합니다.

4.7.7 채널별 측정 (MEAS, MEAW)

채널별 측정의 경우, NC 채널의 측정 프로세스는 항상 해당 채널에서 실행 중인 가공 프로그램으로부터 활성화됩니다. **하나의** 트리거 이벤트(프로브의 상승 또는 하강 에지)와 이동할 거리를 삭제하는(MEAS) 또는 이동할 거리를 삭제하지 않는(MEAW) **하나의** 측정 모드가 측정 블록에 프로그램됩니다. 그러면 측정 블록에 프로그램된 **모든** 축이 측정 프로세스에 참여합니다.

측정 블록이 활성화되는 즉시 프로브가 공작물로 이동됩니다. 프로브의 전환 에지에서는 측정 블록에 프로그래밍된 모든 축의 위치가 측정되어 각 축의 해당 메모리 셀에 기록됩니다.

측정 결과는 가공 프로그램에서 또는 기계와 공작물 좌표계 양쪽 모두에서 동기 동작으로 읽을 수 있습니다.



구문

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

의미

MEAS:	이동할 거리 삭제**하고** 측정	
	효율성:	넌모달
MEAW:	이동할 거리를 삭제하지 **않고** 측정	
	효율성:	넌모달
	용도:	프로그램된 위치에 접근하는 측정 작업

<TE>:	측정을 시작하는 트리거 이벤트		
	유형:	INT	
	값 범위:		-2, -1, 1, 2
	값:	(+)1	프로브 1의 상승 에지(측정 입력 1)
		-1	프로브 1의 하강 에지(측정 입력 1)
		(+)2	프로브 2의 상승 에지(측정 입력 2)
		-2	프로브 2의 하강 에지(측정 입력 2)
	참고: 최대 2개의 프로브가 있습니다(구성 레벨에 따름).		
G...:	보간 유형 (예: G0, G1, G2 또는 G3)		
X... Y... Z...:	직교 좌표의 종점		

참고

MEAS 및 MEAW는 넌모달 명령으로서 모션 작업과 함께 프로그래밍됩니다. 축 개수, 이송 속도 및 보간 유형 (G0, G1 등) 을 측정 작업에 맞게 선택해야 합니다.

예

프로그램 코드	설명
... N10 MEAS=1 G1 F1000 X100 Y730 Z40 ...	; 첫 번째 측정 입력 및 직선 보간에서 프로브를 통한 측정 블록. 전처리 정지가 자동으로 생성됩니다.

추가 정보

상태 조회

프로그램에서 평가가 요구되는 경우, 프로브가 편향되었는지 아니면 전환되었는지 등 다음 시스템 변수를 통해 상태를 조회할 수 있습니다:

시스템 변수	의미	데이터 유형	값	
$A_PROBE[<n>]	프로브 상태	INT	0	프로브 비편향
			1	프로브 편향

시스템 변수	의미	데이터 유형	값	
$AC_MEA[<n>]	프로브의 전환 상태 $AC_MEA[<n>]는 측정 시작 시 자동으로 리셋됩니다.	INT	0	프로브가 전환되지 않았습니다.
			1	프로브가 전환되었습니다.

<n> = 프로브 수

측정값 읽기

채널별 측정의 경우, 블록의 모든 이송 경로 및 포지셔닝 축에 대한 위치가 판독됩니다(최대 축 수는 제어 구성에 따라 다름). MEAS의 경우 프로브가 실행된 후 정의된 방법으로 이동이 제동됩니다.

참고

측정 블록에 기하 축이 프로그래밍된 경우 모든 현재 기하 축에 대한 측정값이 저장됩니다. 변환에 참여하는 축이 측정 블록에 프로그래밍될 경우 이 변환에 참여하는 모든 축에 대한 측정값이 기록됩니다.

측정 결과 읽기

프로브의 축 측정값은 가공 프로그램과 동기 동작에서 다음 시스템 변수를 통해 읽을 수 있습니다.

시스템 변수	의미
$AA_MM[<Axis>]	기계 좌표계에서의 프로브 측정값
$AA_MW[<Axis>]	공작물 좌표계에서의 프로브 측정값

4.7.8 축별 측정(MEASA, MEAWA, MEAC)(옵션)

축별 측정의 경우, 가공 프로그램 **또는** 동기 동작에서 적용될 수 있습니다. 축에 대해 두 가지 측정 시스템이 제공되는 경우, 측정에 두 가지 모두 사용할 수 있습니다.

다음과 같은 측정 방법을 사용할 수 있습니다.

- 이동할 거리를 삭제하고 측정(MEASA)
- 이동할 거리를 삭제하지 않고 측정(MEAWA)
- 이동할 거리를 삭제하지 않고 연속 측정 (MEAC)

축에 MEASA 또는 MEAWA를 프로그래밍하면 측정할 때마다 최대 4개의 측정값을 확보하고 트리거 이벤트에 따라 시스템 변수에 저장합니다.

MEAC를 이용한 연속 측정의 경우, 측정 결과는 FIFO 변수에 저장됩니다.

구문

```
MEASA[<axis>]=(<mode>,<TE1>,...,<TE4>)
MEAWA[<axis>]=(<mode>,<TE1>,...,<TE4>)
MEAC[<Axis>]=(<Mode>,<MeasMem>,<TE1>,...,<TE4>)
```

참고

MEASA 및 MEAWA는 넌모달이므로 한 블록에 함께 프로그래밍할 수 있습니다. 그러나 MEASA/MEAWA를 동일한 블록에 MEAS/MEAW와 함께 프로그래밍하면 에러 메시지가 출력됩니다.

의미

MEASA:	이동할 거리를 삭제한 축별 측정	
	효율성:	넌모달
MEAWA:	이동할 거리를 삭제하지 않은 축별 측정	
	효율성:	넌모달
MEAC:	이동할 거리를 삭제하지 않은 축별 연속 측정	
	효율성:	넌모달
<Axis>:	측정에 사용된 채널 축 이름	

<Mode>:	운전 모드를 나타내는 두 자리(xx) 수(측정 모드 및 측정 시스템)	
	일의 자릿수: 측정 모드 트리거 이벤트를 시간 순서로, 아니면 프로그램된 순서로 실행할지 지정합니다.	
	x**0**	측정 작업 취소
	x**1**	최대 4개의 **서로 다른** 트리거 이벤트를 동시에 실행할 수 있습니다.
	x**2**	최대 4개의 서로 다른 트리거 이벤트를 연속적으로 실행할 수 있습니다.
	x**3**	최대 4개의 서로 다른 트리거 이벤트를 연속적으로 실행할 수 있지만 시작할 때 트리거 이벤트 1을 모니터링하지 않습니다 (알람 21700/21703 억제). **참고:** MEAC는 이 모드를 지원하지 않습니다.
	십의 자릿수: 측정 시스템 측정을 수행할 측정 시스템을 지정합니다.	
	0x (또는 지정 없음)	활성 측정 시스템
	1x	측정 시스템 1
	2x	측정 시스템 2
	3x	두 측정 시스템
<TE>:	측정을 시작하는 트리거 이벤트	
	유형:	INT
	값 범위:	-2, -1, 1, 2
	(+)1	프로브 1의 상승 에지
	-1	프로브 1의 하강 에지
	(+)2	프로브 2의 상승 에지
	-2	프로브 2의 하강 에지
	참고: 두 측정 시스템을 이용해 측정 프로세스를 실행하는 경우 최대 2개의 트리거 이벤트를 프로그램할 수 있습니다(상승 또는 하강 에지). 트리거 이벤트 둘 다에 대해 두 측정 시스템 모두의 측정값이 얻어집니다.	
<MeasMem>:	FIFO 수 (원형 버퍼)	

예

예 1: 모드 1에서 이동할 거리를 삭제한 축별 측정(시간 순서대로 평가)

a) 1개 측정 시스템 사용 측정

프로그램 코드	설명
...	
N100 **MEASA[X]=(1,1,-1)** G01 X100 F100	; 활성 측정 시스템으로 모드 1에서 측정. 이송 경로의 프로브 1에서 X=100까지의 상승/하강 에지에서 측정 신호 대기.
N110 IF $AC_MEA[1]==FALSE GOTOF END	; 측정이 성공했는지 확인합니다.
N120 R10=$AA_MM1[X]	; 첫 번째 프로그래밍된 트리거 이벤트 (상승 에지)에서 가져온 측정값을 저장합니다.
N130 R11=$AA_MM2[X]	' 두 번째 프로그래밍된 트리거 이벤트 (하강 에지)에서 가져온 측정값을 저장합니다.
N140 END:	

b) 2개 측정 시스템 사용 측정

프로그램 코드	설명
...	
N200 **MEASA[X]=(31,1,-1)** G01 X100 F100	; 2개 측정 시스템으로 모드 1에서 측정. 이송 경로의 프로브 1에서 X=100까지의 상승/하강 에지에서 측정 신호 대기.
N210 IF $AC_MEA[1]==FALSE GOTOF END	; 측정이 성공했는지 확인합니다.
N220 R10=$AA_MM1[X]	; 상승 에지에서 측정 시스템 1의 측정값을 저장합니다.
N230 R11=$AA_MM2[X]	; 상승 에지에서 측정 시스템 2의 측정값을 저장합니다.
N240 R12=$AA_MM3[X]	; 하강 에지에서 측정 시스템 1의 측정값을 저장합니다.
N250 R13=$AA_MM4[X]	; 하강 에지에서 측정 시스템 2의 측정값을 저장합니다.
N260 END:	

예 2: 모드 2에서 이동할 거리를 삭제한 축별 측정(시간 순서대로 평가)

프로그램 코드	설명
...	
N100 **MEASA[X]=(2,1,-1,2,-2)** G01 X100 F100	; 활성 측정 시스템으로 모드 2에서 측정. X=100으로 경로를 이송하면서 프로브 1 상승 에지, 프로브 1 하강 에지, 프로브 2 상승 에지, 프로브 2 하강 에지의 순서로 측정 신호를 기다림
N110 IF $AC_MEA[1]==FALSE GOTOF PROBE2	; 프로브 1의 측정이 성공했는지 점검합니다.

프로그램 코드	설명
N120 R10=$AA_MM1[X]	; 첫 번째 프로그래밍된 트리거 이벤트 (프로브 1 상승 에지) 에서 가져온 측정값을 저장합니다.
N130 R11=$AA_MM2[X]	; 두 번째 프로그래밍된 트리거 이벤트 (프로브 1 상승 에지) 에서 가져온 측정값을 저장합니다.
N140, PROBE2:	
N150 IF $AC_MEA[2]==FALSE GOTOF END	; 프로브 2의 측정이 성공했는지 점검합니다.
N160 R12=$AA_MM3[X]	; 세 번째 프로그래밍된 트리거 이벤트 (프로브 2 상승 에지) 에서 가져온 측정값을 저장합니다.
N170 R13=$AA_MM4[X]	; 네 번째 프로그래밍된 트리거 이벤트 (프로브 2 상승 에지) 에서 가져온 측정값을 저장합니다.
N180 END:	

예 3: 모드 1에서 축별 연속 측정(시간 순서대로 평가)

a) 최대 100개의 측정값 측정

프로그램 코드	설명
...	
N110 DEF REAL MEASVALUE[100]	
N120 DEF INT loop=0	
N130 **MEAC[X]=(1,1,-1)** G01 X1000 F100	활성 측정 시스템의 모드 1에서 측정, $AC_FIFO1에 측정값 저장, 프로브 1 이송 경로에서 X=1000까지 하강 에지의 측정 시스템 대기.
N135 STOPRE	
N140 MEAC[X]=(0)	; 축 위치에 도달할 경우 측정을 종료합니다.
N150 R1=$AC_FIFO1[4]	; 파라미터 R1에 누적된 측정값 개수를 저장합니다.
N160 FOR loop=0 TO R1-1	
N170 MEASURED VALUE[loop]=$AC_FIFO1[0]	; $AC_FIFO1에서 측정값을 읽어와 저장합니다.
N180 ENDFOR	

b) 10개 측정값 이후 이동할 거리를 삭제하면서 측정

프로그램 코드	설명
...	
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; 이동할 거리 삭제.
N20 **MEAC[x]=(1,1,1,-1)** G01 X100 F500	
N30 MEAC [X]=(0)	
N40 R1 = $AC_FIFO1[4]	; 측정값의 개수.
...	

c) 프로브 2개를 이용해 공구 상승/하강 측면 측정

프로그램 코드	설명
...	
N110 DEF REAL MEASVALUE[16]	
N120 DEF INT loop=0	
N130 **MEAC[X]=(1,1,-1,2)** G01 X100 F100	; 모드 1에서 활성 측정 시스템으로 측정, $AC_FIFO1에 측정값 저장, X=100까지 경로를 이송하며 프로브 1 하강 에지와 프로브 2의 상승 에지에서 측정 신호 대기.
N140 STOPRE	; 전처리 정지.
N150 MEAC[X]=(0)	; 축 위치에 도달할 경우 측정을 종료합니다.
N160 R1=$AC_FIFO1[4]	; 파라미터 R1에 누적된 측정값 개수를 저장합니다.
N170 FOR loop=0 TO R1-1	
N180 MEASURED VALUE[loop]=$AC_FIFO1[0]	; $AC_FIFO1에서 측정값을 읽어와 저장합니다.
N190 ENDFOR	

추가 정보

측정 작업

가공 프로그램에 또는 동기 동작으로 측정 작업을 프로그래밍할 수 있습니다. 각 축에 대해 주어진 시간에 하나의 측정 작업만 실행할 수 있습니다.

참고

이송 속도는 특정 측정 작업에 맞도록 조정해야 합니다.

MEASA 및 MEAWA의 경우 각 위치 제어 싸이클에서 동일한 유형의 트리거 이벤트가 1개, 서로 다른 유형의 트리거 이벤트가 4개 이하인 이송 속도에서만 정확한 결과를 얻을 수 있습니다.

MEAC를 통한 연속 측정의 경우 보간 싸이클 및 위치 제어 싸이클 간의 비율은 1:8을 초과하지 않아야 합니다.

트리거 이벤트

트리거 이벤트는 프로브 수 및 측정 신호의 트리거 기준 (상승 또는 하강 에지) 으로 구성됩니다.

지정된 프로브에 대해 최대 4개의 트리거 이벤트를 각 측정마다 처리할 수 있습니다. 즉, 각각 2개의 측정 신호 에지를 가진 최대 2개의 프로브가 지정됩니다. 트리거 이벤트의 처리 순서 및 최대 개수는 선택한 모드에 따라 다릅니다.

두 측정 시스템을 이용해 측정 작업을 수행하는 경우 최대 2개의 트리거 이벤트를 프로그램할 수 있습니다(상승 또는 하강 에지). 트리거 이벤트 둘 다에 대해 두 프로브 모두의 측정값이 얻어집니다.

PROFIBUS 텔레그램 391(PROFIBUS 통신의 기본 설정)을 사용하면 각 트리거 이벤트와 위치 제어 싸이클에 하나의 측정값만 가능합니다.

MEAC의 경우 트리거 이벤트당 측정 값의 개수는 작동 모드 1 MEAC의 경우 PROFIBUS 텔레그램 395를 사용하여 각 트리거 이벤트 및 위치 컨트롤러 클럭 싸이클마다 상승용 측정값 총 8개 그리고 하강용 측정값 총 8개까지 증가할 수 있습니다.

- 1개의 프로브: 상승 에지 8개 측정값 및 하강 에지 8개 측정값
- 2개의 프로브: 각 프로브별로 상승 에지 4개 측정값 및 하강 에지 4개 측정값

이는 PROFIBUS 텔레그램 395를 이용하여 더 높은 이송 속도 또는 더 높은 속도에 도달할 수 있다는 것을 의미합니다.

운전 모드

운전 모드의 첫 번째 숫자 (십진수의 첫째 자릿수) 는 필수 측정 시스템을 선택합니다. 하나의 측정 시스템만 설치되었지만 두 번째가 프로그래밍된 경우 설치된 시스템이 자동으로 선택됩니다.

두 번째 숫자(일의 자릿수)는 원하는 측정 모드를 선택합니다. 따라서 측정 과정은 관련 제어 시스템에서 지원하는 옵션에 맞추어집니다.

- **모드 1**
  트리거 이벤트는 발생된 시간 순서대로 평가됩니다.
- **모드 2**
  트리거 이벤트는 프로그래밍된 순서대로 평가됩니다.
- **모드 3**
  트리거 이벤트는 프로그래밍된 순서대로 평가되지만 시작 시 트리거 이벤트 1을 모니터링하지 않습니다.

이동할 거리 삭제/삭제 없이 측정

명령 MEASA를 프로그래밍하면 이동할 거리는 모든 필요한 측정값을 얻을 때까지 삭제되지 않습니다.

MEAWA는 프로그래밍된 위치를 항상 접근해야 하는 특별한 측정 작업에 사용됩니다.

참고

MEASA는 동기 동작에서 프로그래밍할 수 없습니다. 대안으로 MEAWA와 이동할 거리 삭제를 동기 동작으로 프로그래밍할 수 있습니다.

MEAWA를 통한 측정 작업을 동기 동작에서 시작하는 경우 측정값은 기계 좌표계에서만 사용할 수 있습니다.

기하 축/변환

형상 축에 대해 축 측정을 시작하는 경우 나머지 모든 형상 축에 대해 동일한 측정 작업을 프로그래밍해야 합니다. 변환에 관련된 축에 대해 동일하게 적용됩니다.

예:

```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

또는

```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

상태 조회

프로그램에서 평가가 요구되는 경우, 프로브가 편향되었는지 아니면 전환되었는지 등 다음 시스템 변수를 통해 상태를 조회할 수 있습니다:

시스템 변수	의미	데이터 유형	값	
$A_PROBE[<n>]	프로브 상태	INT	0	프로브 비편향
			1	프로브 편향
$AC_MEA[<n>]	프로브의 전환 상태 $AC_MEA[<n>]는 측정 시작 시 자동으로 리셋됩니다.	INT	0	프로브가 전환되지 않았습니다.
			1	프로브가 전환되었습니다(측정 블록에 프로그램된 모든 트리거 이벤트가 발생했음).

<n> = 프로브 수

참고

측정이 동기 동작에서 시작될 경우 $AC_MEA는 더 이상 업데이트되지 않습니다. 이 경우, NC/PLC 인터페이스 신호 DB31, ... DBX62.3 또는 해당 변수 $AA_MEAACT[<axis>]를 조회해야 합니다.

$AA_MEAACT==1: 측정 적용

$AA_MEAACT==0: 측정이 활성화되지 않음

프로브 제한

NC 프로그램 또는 동기 동작에서 PROFIBUS 텔레그램 395 사용 시 시스템 변수 $A_PROBE_LIMITED를 통해 프로브 제한 상태를 읽을 수 있습니다.

$A_PROBE_LIMITED[<n>] == 0: 프로브 제한 비활성/리셋

$A_PROBE_LIMITED[<n>] == 1: 프로브 제한 활성

<n> = 프로브 번호

MEASA/MEAWA의 측정 결과

프로브의 측정값은 가공 프로그램과 동기 동작에서 다음 시스템 변수를 통해 읽을 수 있습니다.

시스템 변수	의미
$AA_MM1[<Axis>] ... $AA_MM4[<Axis>]	기계 좌표계에서 트리거 이벤트 1에 대한 프로브 측정값 ... 기계 좌표계에서 트리거 이벤트 4에 대한 프로브 측정값
$AA_MW1[<Axis>] ... $AA_MW4[<Axis>]	공작물 좌표계에서 트리거 이벤트 1에 대한 프로브 측정값 ... 공작물 좌표계에서 트리거 이벤트 4에 대한 프로브 측정값

<Axis> = 측정 축

두 개의 측정 시스템에 의해 측정 작업이 실행되는 경우 두 측정 시스템 모두에서 가능한 각 트리거 이벤트를 가져옵니다. 그런 다음, 시스템 변수 지정은 다음과 같습니다.

$AA_MM1[<Axis>]	또는	$AA_MW1[<Axis>]	트리거 이벤트 1의 측정 시스템 1에서 측정한 값
$AA_MM2[<Axis>]	또는	$AA_MW2[<Axis>]	트리거 이벤트 1의 측정 시스템 2에서 측정한 값
$AA_MM3[<Axis>]	또는	$AA_MW3[<Axis>]	트리거 이벤트 2의 측정 시스템 1에서 측정한 값
$AA_MM4[<Axis>]	또는	$AA_MW4[<Axis>]	트리거 이벤트 2의 측정 시스템 2에서 측정한 값

연속 측정 (MEAC)

MEAC의 측정값은 기계 좌표계에서 사용할 수 있으며 프로그래밍된 FIFO[n] 메모리(원형 버퍼) 에 저장됩니다. 측정을 위해 2개의 프로브가 구성된 경우 두 번째 프로브의 측정값은 이러한 용도로 특별히 구성된 FIFO[n+1] 메모리에 별도로 저장됩니다(머신 데이터에 정의).

FIFO 메모리는 순환 원리에 따라 측정값을 $AC_FIFO 변수에 기록하는 원형 버퍼입니다.

참고

FIFO 내용은 원형 버퍼에서 한 번만 읽을 수 있습니다. 측정 데이터를 여러 번 사용해야 하는 경우 사용자 데이터에 버퍼링해야 합니다.

FIFO 메모리의 측정값 수가 머신 데이터에 정의된 최대값을 초과할 경우 측정이 자동으로 종료됩니다.

측정값을 주기적으로 읽어 연속 측정 프로세스를 실행할 수 있습니다. 이 경우 새 측정값이 입력되는 횟수와 최소한 동일하게 데이터를 읽어야 합니다.

추가 정보: 기능 매뉴얼 동기 동작

프로그래밍 에러 방지

다음 프로그래밍 에러가 감지되어 표시됩니다.

- MEASA/MEAWA가 동일한 블록에 MEAS/MEAW와 함께 프로그래밍되는 경우
  예:
  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA의 파라미터 개수가 2보다 작거나 5보다 큰 경우
  예:
  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```
- MEASA/MEAWA의 트리거 이벤트가 1/ -1/ 2/ -2가 아닌 경우
  예:
  ```
  N01 MEASA[B]=(1,1,3) B100
  ```
- MEASA/MEAWA가 잘못된 모드를 가진 경우
  예:
  ```
  N01 MEAWA[B]=(4,1) B100
  ```
- MEASA/MEAWA의 트리거 이벤트가 두 번 프로그래밍된 경우
  예:
  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```
- MEASA/MEAWA에서 기하 축이 누락된 경우
  예:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100; GEO 축 X/Y/Z
  ```
- 측정 작업이 기하 축과 일치하지 않는 경우
  예:
  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

다음 또한 참고:

동기 동작 (페이지 1083)

4.7.9 OEM 사용자를 위한 특수 기능 (OMA1 ... OMA5, OEMIPO1, OEMIPO2, G810 ... G829)

OEM 어드레스

OEM 주소의 의미는 OEM 사용자에 의해 결정됩니다. 해당 기능은 컴파일 싸이클을 통해 조합됩니다. 예비용으로 5개의 OEM 어드레스 (OMA1 ... OMA5) 가 있습니다. 주소 식별자는 설정 가능합니다. 블록에서 OEM 주소를 프로그래밍할 수 있습니다.

유보된 G 코드 호출

OEM 사용자를 위해서는 다음과 같은 G 코드 호출이 예비용으로 제공됩니다.

- OEMIPO1, OEMIPO2 (G 코드 그룹 1)
- G810 ... G819 (G 코드 그룹 31)
- G820 ... G829 (G 코드 그룹 32)

해당 기능은 컴파일 싸이클을 통해 조합됩니다.

기능 및 서브프로그램

또한 OEM 사용자는 파라미터 전송을 통해 사전 정의된 기능 및 서브프로그램을 설정할 수 있습니다.

참고

공작물 시뮬레이션

SW 4.4까지는 컴파일 싸이클이 지원되지 않으며, SW 4.4 이상에서는 일부 컴파일 싸이클 (CC) 만 공작물 시뮬레이션을 위해 지원됩니다.

지원되지 않는 컴파일 싸이클 (OMA1 ... OMA5, OEMIPO1/2, G810 ... G829, 해당 절차 및 기능) 의 가공 프로그램에 사용되는 언어 명령 - 따라서 개별적으로 처리하지 않아도 알람 메시지 생성 및 시뮬레이션 취소가 수행됩니다.

해결: 가공 프로그램 ($P_SIM query) 에서 누락된 CC 고유 언어 요소를 개별적으로 처리하십시오.
예:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ; CC 시뮬레이션에서는 활성화되지 않음
ELSE
N5 X300 OMA1=10
ENDIF
```

4.7.10 코너 감속을 통한 이송 속도 감소 (FENDNORM, G62, G621)

자동 코너 감속을 사용하여 코너에 도달하기 전에 종 모양의 곡선을 따라 이송 속도가 감소합니다. 또한 세팅 데이터를 통한 가공 관련 공구 동작의 범위를 파라메타로 지정할 수 있습니다.

- 피드 감소의 시작 및 끝
- 피드가 감소되는 상태에서 오버라이드
- 관련 코너의 삭제

관련 코너는 이러한 코너의 내부 각도가 세팅 데이터에서 파라메타로 지정된 코너보다 작습니다.

기본값 FENDNORM은 자동 코너 오버라이드의 기능을 비활성화합니다.

구문

```
FENDNORM
G62 G41
G621
```

의미

FENDNORM:	자동 코너 감속 OFF
G62:	공구 반경 보정이 활성 상태일 때 **내부 코너에서** 코너 감속
G621:	공구 반경 보정이 활성 상태일 때 **모든 코너에서** 코너 감속

G62은 활성 공구 반경 보정 G41/G42 및 활성 연속 경로 모드 G64/G641을 통해 내부 코너에서만 작동합니다.

코너는 감소된 이송 속도로 접근하며 다음과 같이 계산됩니다.

F * (피드 감소에 대한 오버라이드) * 피드 오버라이드

중심 경로를 기준으로 코너에서 공구의 방향을 변경할 정확한 지점에서 최대 가능 피드 감소에 도달한다.

G621은 FGROUP으로 정의된 축의 각 코너에서 G62에 동일하게 작용합니다.

4.7.11 프로그램 가능한 이동 종료 기준 (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

경로 보간 (G601, G602 및 G603) 의 블록 변경 기준과 마찬가지로, 가공 프로그램 또는 명령/PLC 축의 동기 동작에 싱글 블록 보간의 모션 종료 기준을 프로그래밍할 수 있습니다.

설정된 모션 종료 기준은 싱글 축 이동이 포함된 가공 프로그램 블록과 공정 싸이클 블록을 완료하는 속도에 영향을 미칩니다. FC15/16/18을 통해 PLC에 동일하게 적용됩니다.

구문

```
FINEA[<axis>]
COARSEA[<axis>]
IPOENDA[<axis>]
IPOBRKA(<축>[,<시간 인스턴트>])
ADISPOSA[<축>]=(<모드>,<범위 크기>)
```

의미

FINEA:	이동 종료 기준: "미세 정위치 정지"		
	적용:	모달	
COARSEA:	이동 종료 기준: "일반 정위치 정지"		
	적용:	모달	
IPOENDA:	모션 종료 기준: "보간자 정지"		
	적용:	모달	
IPOBRKA:	블록 변경 기준: 정지 동작		
	적용:	모달	
ADISPOSA:	이동 종료 기준에 대한 공차 범위		
	적용:	모달	
<axis>:	채널 축 이름 (X, Y, ...)		
<instant in time>:	블록 변경 시간, 정지 동작을 %로 참조: • 100% = 정지 동작 시작 • 0% = 정지 동작 종료, IPOENDA와 동일한 중요도		
	유형:	REAL	
<mode>:	공차 범위 참조		
	값 범위:	0	허용 오차 창 비활성
		1	설정 위치에 따라 허용 오차 창
		2	실제 위치에 따라 허용 오차 창
	유형:	INT	
<window size>:	허용 오차 창의 크기		
	유형:	REAL	

예

예 1: 이동 종료 기준: "보간자 정지"

프로그램 코드

```
; 포지셔닝 축 X를 100, 속도 200 m/min, 가속도 90%로 이송,
; 이동 종료 기준: 보간자 정지
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]
```

프로그램 코드
; 동기 동작: ; ALWAYS IF: 입력 1을 설정 ; THEN 포지셔닝 축 X를 50, 속도 200 m/min, 가속도 140% 이송, ; 이동 종료 기준: 보간자 정지 N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140 IPOENDA[X]

예 2: 블록 변경 기준: "정지 동작"

프로그램 코드	코멘트
	; 디폴트 설정 적용
N40 POS[X]=100	; X에서 위치 100으로 포지셔닝 모션
	블록 변경 기준: 미세 정위치 정지
N20 IPOBRKA(X,100)	; 블록 변경 기준: "정지 동작",
	100% = 정지 동작 시작
N30 POS[X]=200	; X축이 제동되기 시작하는 즉시 블록을 변경합니다.
N40 POS[X]=250	; X축이 위치 200에서 제동되지 않고 위치 250까지 계속 이송됩니다.
	축이 제동을 시작하자마자 블록이 변경됩니다.
N50 POS[X]=0	; X축이 제동되어 위치 0으로 돌아갑니다.
	위치 0에서 "미세 정위치 정지"로 블록 변경이 이루어집니다.
N60 X10 F100	; X축은 경로 축으로 위치 10으로 이송됩니다.

추가 정보

이동 종료 기준에 대한 시스템 변수

효과적인 이동 종료 기준은 시스템 변수 $AA_MOTEND를 사용하여 읽을 수 있습니다.

블록 변경 기준: "정지 동작" (IPOBRKA)

블록 변경 기준 "정지 동작"을 활성화할 때 옵션 블록 변경 시간 인스턴트 값이 프로그래밍되는 경우, 그 값은 다음 포지셔닝 모션에 적용되며 메인 실행에 동기화된 설정 데이터로 쓰여집니다. 블록 변경 시간 인스턴트 값을 지정하지 않으면 설정 데이터의 실제 값이 사용됩니다.

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

해당 축에 대해 축 모션 종료 기준(FINEA, COARSEA , IPOENDA)을 프로그래밍하면 해당 축에 대해 IPOBRKA가 비활성화됩니다.

추가 블록 변경 기준: "공차 범위" (ADISPOSA)

ADISPOSA를 사용하면 블록 끝 (실제 위치 또는 지령치 위치로서) 에 있는 공차 범위를 추가 블록 변경 기준으로 정의할 수 있습니다. 그런 경우에는 블록 변경을 위한 두 가지 조건을 만족해야 합니다.

- 블록 변경 기준: "정지 동작"
- 블록 변경 기준: "공차 범위"

4.8 좌표 변환 (프레임)

4.8.1 프레임 변수를 통한 좌표 변환

ROT, AROT, SCALE 같은 프레임 명령 (페이지 339)에 더해 프레임 변수 $P_...FR(데이터 관리 프레임) 및 $P_...FRAME(활성 프레임)을 사용하여 공작물 좌표계(WCS)를 변환할 수도 있습니다.

다음 다이어그램은 프레임 변수 구조의 개요를 보여줍니다.

- 데이터 관리 프레임
- 활성 프레임
- 활성 전체 프레임: 모든 활성 프레임 연결
- NCU 글로벌 프레임
- 채널 프레임

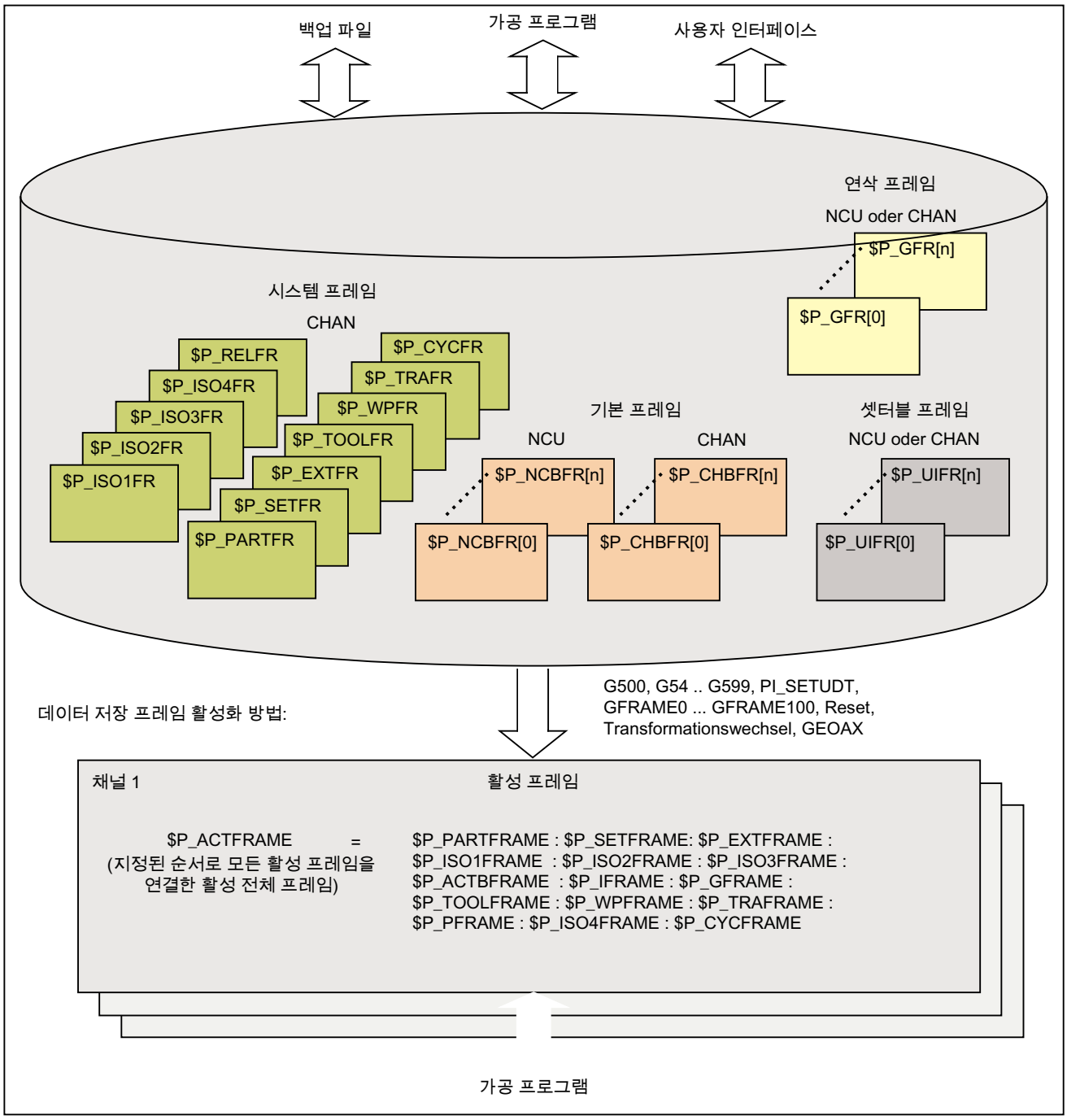


그림 4-1 프레임 변수 개요

4.8.1.1 사전 정의된 프레임 변수 ($P_CHBFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)

활성: 채널 베이스 프레임 $P_CHBFRAME[<n>] ($P_BFRAME)

참고

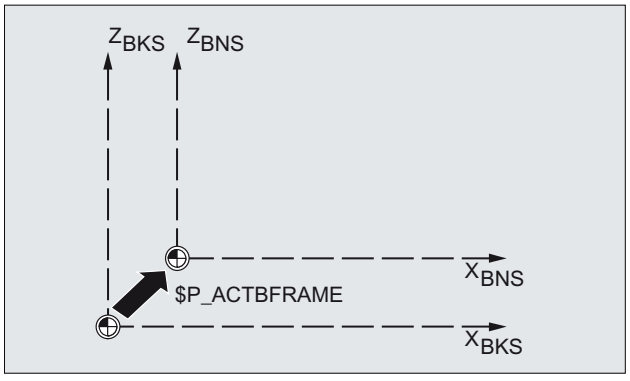
현재 베이스 프레임 $P_BFRAME과 데이터 저장 베이스 프레임 $P_UBFR는 호환성을 위해 유지됩니다.

- $P_BFRAME ≙ $P_CHBFRAME[0]
- $P_UBFR ≙ $P_CHBFR[0].

프레임 변수$P_CHBFRAME[<n>]은 베이직 좌표계 (BCS) 와 베이직 원점 좌표계 (BOS) 사이의 기준을 정의합니다.

현재 채널 베이스 프레임 $P_CHBFRAME[<n>]을 NC 프로그램에서 즉시 활성화해야 하는 경우 다음과 같은 방법을 사용할 수 있습니다.

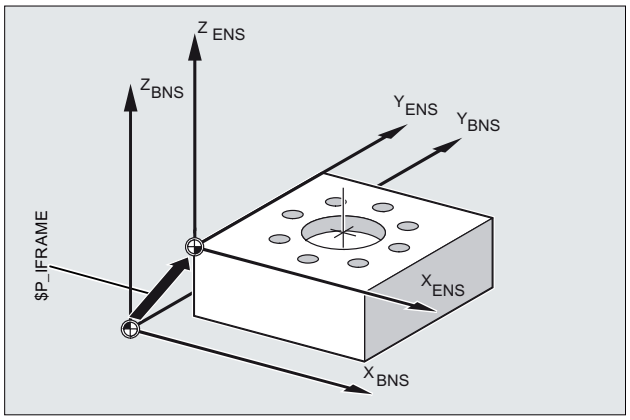
- 명령:
  - G500 (모든 셋터블 프레임 비활성화, 베이스 프레임은 계속 활성)
  - G54~G599 (셋터블 워크 옵셋)
- 데이터 저장소의 채널 베이스 프레임을 현재 채널 베이스 프레임에 지정:
  $P_CHBFRAME[<n>] = $P_CHBFR[<m>]



활성: 채널 셋터블 프레임 $P_IFRAME

프레임 변수 $P_IFRAME은 베이직 원점 좌표계 (BOS) 와 셋터블 제로 시스템 (SZS) 간의 기준을 정의합니다.

- $P_IFRAME은 $P_UIFR[$P_IFRNUM]에 해당합니다.
- 예를 들어 G54를 프로그래밍하면 G54에 의해 정의된 변환, 회전, 스케일링 및 미러링이 $P_IFRAME에 저장됩니다.

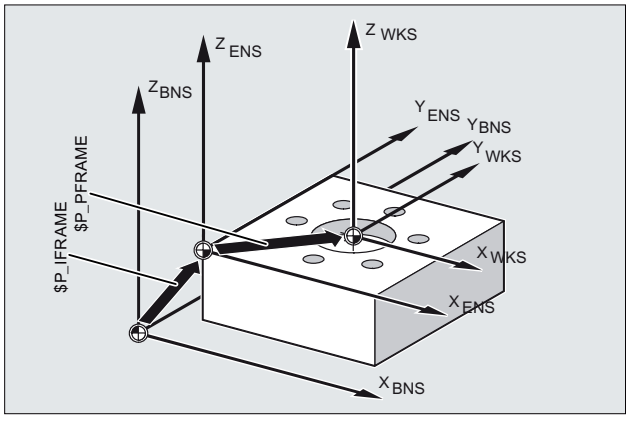


활성: 채널 프로그래밍 프레임 $P_PFRAME

$P_PFRAME 프레임 변수는 셋터블 제로 시스템 (SZS) 과 공작물 좌표계 (WCS) 간의 기준을 정의합니다.

$P_PFRAME 은 다음 방법으로 지정된 프레임을 포함해야 합니다.

- TRANS/ATRANS, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR의 **프로그래밍** 또는
- CTRANS, CROT, CMIRROR 및 CSCALE을 프로그래밍된 FRAME에 **지정**



활성: 전체 프레임 $P_ACTFRAME

채널에 활성인 전체 프레임은 채널에서 작동 중인 모든 채널을 연결한 결과입니다.

```
$P_ACTFRAME =   $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :
                $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :
                $P_ACTBFRAME : $P_IFRAME : $P_GFRAME :
                $P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME :
                $P_PFRAME    : $P_ISO4FRAME : $P_CYCFRAME
```

$P_ACTFRAME은 현재 유효한 공작물 좌표계를 설명합니다.

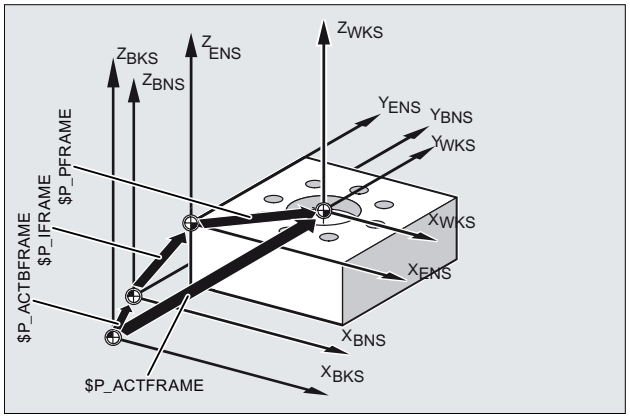
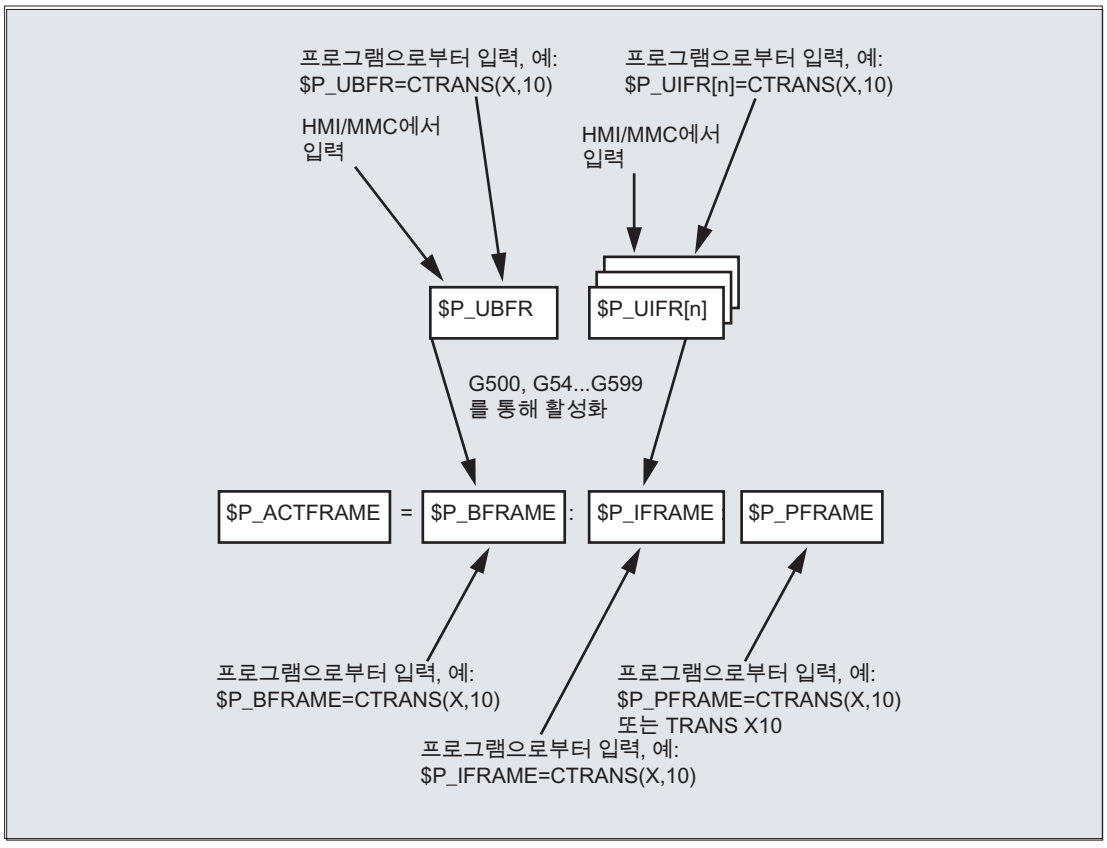


그림 4-2 프레임 변수 $P_ACTFRAME

프레임 $P_BFRAME / $P_CHBFRAME[<n>], $P_IFRAME 또는 $P_PFRAME 중 하나가 변경되면 현재 전체 프레임 $P_ACTFRAME이 다시 계산됩니다.



베이직 프레임 및 셋터블 프레임은 MD 20110 RESET_MODE_MASK를 다음과 같인 설정할 경우 리셋 이후 적용됩니다.

Bit0=1, bit14=1 --> `$P_UBFR` (베이직 프레임) acts

Bit0=1, bit5=1 --> `$P_UIFR [$P_UIFRNUM]` (셋터블 프레임) acts

데이터 저장소: 채널 베이스 프레임 $P_CHBFR[<n>]

프레임 변수 $P_CHBFR[<n>]은 데이터 저장소에서 베이스 프레임을 읽거나 씁니다. 데이터 저장 프레임은 채널에 쓰여지는 즉시 활성화되지 않습니다. 쓰여진 프레임은 다음에 의해 활성화됩니다.

- 채널 리셋과 MD20110 $MC_RESET_MODE_MASK, Bit0 == 1 및 Bit14 == 1
- 명령 `G500`, `G54` ... `G57`, `G505` ... `G599` (베이스 프레임 활성화/비활성화 후 현재 전체 프레임 재계산)

데이터 저장소: 채널 셋터블 프레임 $P_UIFR[<n>]

프레임 변수 $P_UIFR[<n>]은 데이터 저장소에서 셋터블 베이스 프레임을 읽거나 씁니다. 이 프레임은 채널에 쓰여지는 즉시 활성화되지 않습니다. 채널에 쓰여진 프레임은 다음에 의해 계산됩니다.

- G500 명령 (모든 셋터블 프레임 또는 워크 옵셋 비활성화)
- G54 ... G57, G505 ... G599 명령 (특정 셋터블 프레임 또는 워크 옵셋 활성화)

활성 셋터블 프레임	데이터 저장 프레임	(명령과 동일)
$P_IFRAME =	$P_UIFR[0]	G500
	$P_UIFR[1]	G54
	$P_UIFR[2]	G55
	$P_UIFR[3]	G56
	$P_UIFR[4]	G57
	$P_UIFR[5]	G505
	$P_UIFR[6]	G506
	...	...
	$P_UIFR[99]	G599

4.8.2 프레임에 값 지정

4.8.2.1 직접 값 할당(축 값, 각도, 배율)

NC 프로그램의 프레임 또는 프레임 변수에 값을 직접 할당할 수 있습니다.

구문

구문

```
$P_PFRAME = CTRANS(X, <옵셋 값>, Y, <옵셋 값>, Z, <옵셋 값>, ...)
$P_PFRAME = ROT(X, <각도>, Y, <각도>, Z, <각도>, ...)
$P_UIFR[..] = CROT(X, <각도>, Y, <각도>, Z, <각도>,  ...
$P_PFRAME = CSCALE(X, <배율>, Y, <배율>, Z, <배율>, ...)
$P_PFRAME = CMIRROR(X, Y, Z)
```

$P_CHBFRAME[<n>] 구문은 $P_PFRAME과 동일합니다.

의미

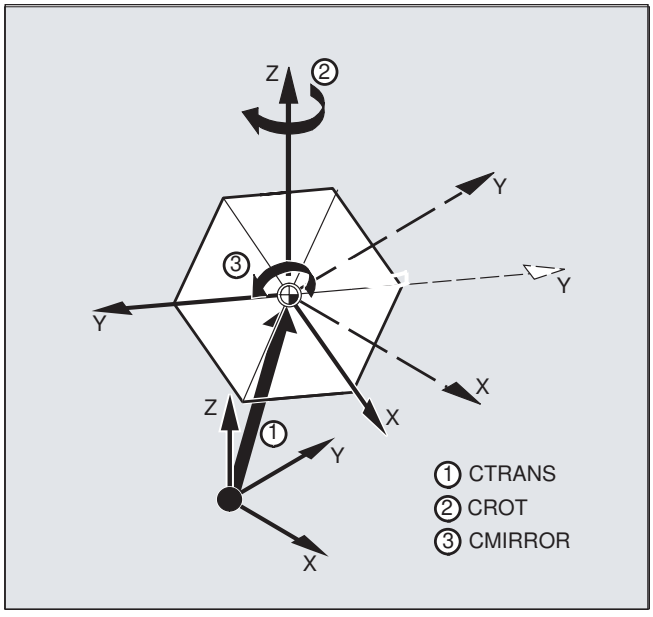
CTRANS:	지정된 축의 변환
CROT:	지정된 축 주위 회전
CSCALE:	지정된 축에 대한 배율 변경
CMIRROR:	지정된 축의 방향 전환
X, Y, Z:	지정된 형상 축의 방향으로 옵셋 값
<옵셋 값>:	옵셋 값
<각도>:	회전 각도
<배율>:	배율 값

예

현재 프로그래밍 프레임의 프레임 콤포넌트에 값 지정

현재 프로그래밍 프레임의 변환, 회전 및 미러링 프레임 콤포넌트에 값 지정:

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) : CROT(Z,45) : CMIRROR(Y)
```



프레임이 회전 콤포넌트 쓰기

셋터블 데이터 저장 프레임 $P_UIFR의 회전 콤포넌트에 CROT으로 세 축의 값 지정:

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

또는 데이터 저장 프레임의 회전 콤포넌트에 포함된 축에 개별 값을 직접 지정:

```
$P_UIFR[5, Y, RT]=0
$P_UIFR[5, X, RT]=0
$P_UIFR[5, Z, RT]=0
```

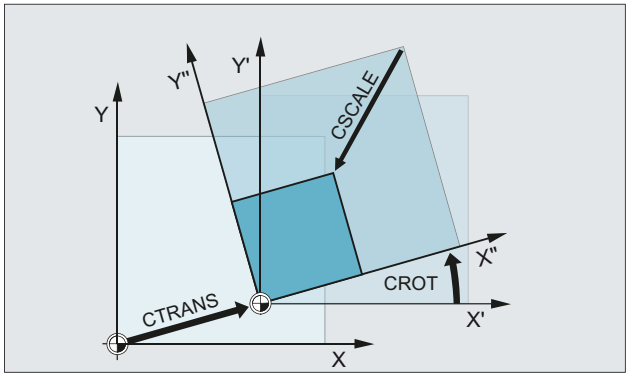
설명

연결 연산자 **:**은 한 프레임의 여러 연산을 조합합니다. 조합된 연산은 왼쪽부터 오른쪽으로 이어서 처리됩니다.

예

$P_PFRAME에 결합된 옵셋, 회전 및 스케일링 연산:

```
$P_PFRAME = CTRANS(...) : CROT(...) : CSCALE...
```



4.8.2.2 프레임 콤포넌트 읽기 및 변경 (TR, FI, RT, SC, MI)

이 기능을 통해 예를 들어 특정 옵셋 값, 회전 각도 등과 같은 프레임의 **개별** 데이터에 액세스할 수 있습니다. 이러한 값을 수정하거나 다른 변수에 할당할 수 있습니다.

구문

`R10=$P_UIFR[$P_UIFNUM,X,RT]`	현재 유효한 설정 가능한 워크 옵셋 $P_UIFRNUM의 X축 주위 회전 각도 RT를 변수 R10에 할당합니다.
`R12=$P_UIFR[25,Z,TR]`	셋터블 프레임 번호 25의 데이터 세트에서 Z 방향의 옵셋 값 TR을 변수 R12에 할당합니다.

R15=$P_PFRAME[Y,TR]	현재 프로그래밍 프레임의 Y에서 옵셋 값 TR을 변수 R15에 할당합니다.
$P_PFRAME[X,TR] = 25	현재 프로그래밍 프레임의 X에서 옵셋 값 TR을 수정합니다. X25가 즉시 적용됩니다.

의미

$P_UIFRNUM:	이 명령으로 현재 유효한 설정 가능한 워크 옵셋에 대한 기준이 자동으로 구성됩니다.
P_UIFR[n,…,…]:	셋터블 프레임 수 n에 액세스할 프레임 수 n을 지정합니다.
	읽거나 수정할 구성 요소를 지정합니다:
TR:	TR 변환
FI:	FI 미세 변환
RT:	RT 회전
SC:	SC 배율 수정
MI:	MI 미러링
X, Y, Z:	해당 축 X, Y, Z도 지정됩니다(예제 참조).

RT 회전 값 범위

1차 기하축을 중심으로 회전: -180° ~ +180°
2차 기하축을 중심으로 회전: -90° ~ +90°
3차 기하축을 중심으로 회전: -180° ~ +180°

설명

프레임 호출

시스템 변수 $P_UIFRNUM을 지정하여 $P_UIFR 또는 G54, G55, ...으로 설정한 현재 워크 옵셋에 액세스할 수 있습니다
($P_UIFRNUM은 현재 설정된 프레임 개수).

다른 모든 저장된 설정 가능한 $P_UIFR 프레임은 해당 번호 $P_UIFR[n]을 지정하여 호출합니다.

사전 정의된 프레임 변수 및 사용자 정의 프레임의 경우 예를 들어 $P_IFRAME으로 이름을 지정합니다.

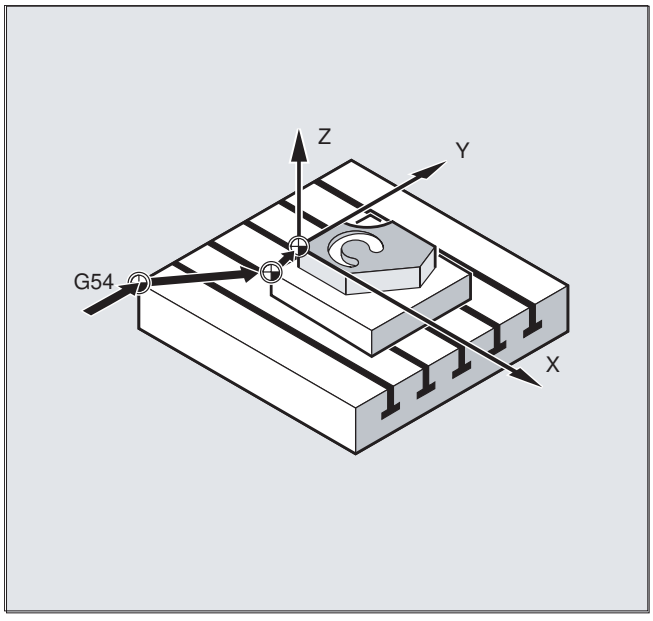
데이터 호출

액세스 또는 수정할 값의 축 이름 및 프레임 콤포넌트를 꺽쇠괄호 안에 입력합니다 (예: [X, RT] 또는 [Z, MI]).

4.8.2.3 프레임 계산

한 프레임을 다른 프레임에 지정하거나 NC 프로그램에서 여러 프레임을 서로 연결할 수 있습니다.

프레임 연결은 동일한 프로세스에서 가공되도록 하나의 팰릿에 정렬된 여러 공작물을 설명할 때 적합한 기능입니다.



프레임 구성 요소에는 팰릿 작업 설명을 위해 중간 값만 포함될 수 있습니다. 이러한 구성 요소는 다양한 공작물 제로 생성을 위해 연결됩니다.

예

할당

프로그램 코드	설명
DEF FRAME SETTING_1	; 로컬 프레임 변수 정의
SETTING_1 = CTRANS(X,10)	; 기능 결과를 프레임 변수에 할당
$P_PFRAME = SETTING_1	; 프레임 변수를 현재 프레임에 할당
DEF FRAME SETTING_4	; 로컬 프레임 변수 정의
SETTING_4 = $P_PFRAME	; 현재 프레임을 프레임 변수에 버퍼링
...	

프로그램 코드	설명
$P_PFRAME = SETTING_4	; 프레임 변수에서 현재 프레임 가져오기

연결

연산자 :은 프레임을 프로그램된 순서대로 서로 연결시킵니다. 옵셋 및 회전과 같은 프레임 콤포넌트는 순서대로 실행됩니다.

프로그램 코드	설명
$P_IFRAME = $P_UIFR[15] : $P_UIFR[16]	; 2개의 설정 가능 데이터 저장 프레임을 ; 연결한 결과 프레임을 ; 활성 설정 가능 전체 프레임에 할당 ; 적용 예: ; $P_UIFR[15]: 옵셋 ; $P_UIFR[16]: 회전
$P_UIFR[3] = $P_UIFR[4] : $P_UIFR[5]	; 2개의 설정 가능 데이터 저장 프레임을 ; 연결한 결과 프레임을 ; 다른 데이터 저장 프레임에 할당

4.8.2.4 프레임 변수 정의 (DEF FRAME)

사전 정의된 프레임 변수 외에 사용자 프레임 변수도 정의할 수 있습니다. 사용자 정의 프레임 변수는 FRAME 유형의 사용자 변수입니다. 프레임의 이름은 사용자 변수 규칙에 따라 자유롭게 할당할 수 있습니다.

CTRANS, CROT, CSCALE 및 CMIRROR 기능은 사용자 정의 프레임 변수에 값을 할당합니다.

구문

```
DEF FRAME <이름>
```

의미

DEF FRAME:	FRAME 유형의 사용자 변수 정의.
<이름>:	프레임 변수의 이름

예

프레임 변수 "PALETTE"를 정의하고 옵셋 및 회전 값 할당:

프로그램 코드	설명
DEF FRAME PALETTE	; 프레임 변수 PALETTE 정의

프로그램 코드	설명
PALETTE = CTRANS(...) : CROT(...)	; 프레임 연결의 결과를 프레임 변수 PALETTE의 ; 옵셋 및 회전에 할당

4.8.3 일반 옵셋 및 미세 옵셋 (CTRANS, CFINE)

미세 옵셋

미세 옵셋 CFINE(...)은 다음 프레임에 적용할 수 있습니다.

- 셋터블 프레임: $P_UIFR 또는 $P_IFRAME
- 베이직 프레임: $P_NCBFR[<n>], $P_CHBFR[<n>], $P_CHBFRAMES[<n>] 또는 $P_ACTBFRAME
- 프로그래밍 프레임: $P_PFRAME

프레임의 미세 옵셋은 CFINE(...) 명령으로 프로그래밍합니다.

일반 옵셋

일반 옵셋 CTRANS(...)는 다음 프레임에 적용할 수 있습니다.

옵셋 합계

옵셋 합계는 일반 옵셋과 미세 옵셋을 더한 결과입니다.

머신 데이터

미세 옵셋 인에이블

미세 옵셋은 다음 머신 데이터로 인에이블합니다.

MD18600 $MN_MM_FRAME_FINE_TRANS = 1

구문

미세 옵셋

- 전체 프레임
  - <프레임> = CFINE(<K_1>,<값>)
  - <프레임> = CFINE(<K_1>,<값>, <K_2>, <값>)
  - <프레임> = CFINE(<K_1>,<값>, <K_2>, <값>, <K_3>, <값>)
- 프레임 콤포넌트
  - <프레임>[<n>, <K_1>, FI] = <값>

일반 옵셋

- 전체 프레임
  - <프레임> = CTRANS(<K_1>,<값>)
  - <프레임> = CTRANS(<K_1>,<값>, <K_2>,<값>)
  - <프레임> = CTRANS(<K_1>,<값>, <K_2>,<값>, <K_3>,<값>)
- 프레임 콤포넌트
  - <프레임>[<n>,<K_1>,TR] = <값>

프로그래밍 프레임 $P_PFRAME의 경우:

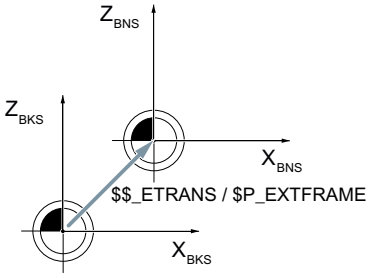
- TRANS <K_1> <값>
- TRANS <K_1> <값> <K_2> <값>
- TRANS <K_1> <값> <K_2> <값> <K_3> <값>

의미

<프레임>:	데이터 저장 프레임 $P_UIFR[<n>] (예: 셋터블 프레임)
CFINE:	미세 옵셋, 추가 옵셋.
CTRANS:	일반 옵셋, 절대치 옵셋.
TRANS:	프로그래밍 프레임만 해당: 일반 옵셋, 절대치 옵셋.
<K_n>:	좌표 축 X, Y, Z
<값>:	옵셋 값

4.8.4 외부 워크 옵셋 ($AA_ETRANS)

외부 워크 옵셋은 베이스 좌표계 (BCS) 와 베이직 원점 좌표계 (BOS) 간의 직선 옵셋입니다.



$AA_ETRANS를 이용한 외부 워크 옵셋은 머신 데이터 파라미터 설정에 따라 두 가지 방식으로 적용됩니다.

1. NC/PLC 인터페이스 신호가 활성화된 후 시스템 변수 $AA_ETRANS가 직접 옵셋 값으로 적용됩니다.
2. NC/PLC 인터페이스 신호가 활성화된 후 시스템 변수가 $AA_ETRANS가 활성 시스템 프레임 $P:EXTFRAME과 데이터 저장 프레임 $P_EXTFR로 전송됩니다. 그런 다음 활성 전체 프레임을 다시 계산합니다.

머신 데이터

시스템 변수 $AA_ETRANS는 다음 머신 데이터가 선택한 두 가지 절차 중 하나에 따라 적용됩니다.

MD28082 $MC_MM_SYSTEM_FRAME_MASK,Bit1 = <값>

<값>	의미
0	기능: PLC, HMI 또는 NC 프로그램이 직접 작성한 $AA_ETRANS[]. 다음 가능 이송 블록에서 $AA_ETRANS[<axis>]를 위한 제로 옵셋 후퇴 인에이블: DB31, ... DBX3.0
1	기능: 활성 시스템 프레임 $P:EXTFRAME과 데이터 저장 프레임 $P_EXTFR 활성화 다음에 의한 $AA_ETRANS[<axis>]를 위한 제로 옵셋 후퇴 인에이블: DB31, ... DBX3.0. 채널에서 다음 작업이 수행됩니다. • 채널의 모든 이송 동작 정지 (명령 및 PLC 축 제외) • 전처리 정지 후 재조직 (STOPRE) • 활성 프레임 일반 옵셋 $P_EXTFRAME[<축>] = $AA_ETRANS[<축>] • 데이터 저장 프레임의 일반 옵셋 $P_EXTFR[<축>] = $AA_ETRANS[<축>] • 활성 전체 프레임 $P_ACTFRAME 재계산 • 프로그램된 축에서 옵셋 후퇴 • 인터럽트된 이송 모션 또는 NC 프로그램 계속 실행

프로그래밍

- 구문

  ```
  $AA_ETRANS[<축>] = <값>
  ```

- 의미

$AA_ETRANS:	외부 워크 옵셋 버퍼링을 위한 시스템 변수
<축>:	채널 축
<값>:	옵셋 값

NC/PLC 인터페이스 신호

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<축>] = $P_EXTFR[<축>] = $AA_ETRANS[<축>]

4.8.5 원점 복귀 상태를 삭제하고 실제 값 설정 (PRESETON)

PRESETON() 과정에서 기계 좌표계 (MCS)로 하나 이상의 축에 새 실제 값을 설정합니다. 축의 MCS 워크 옵셋과 같은 기능을 합니다. 축은 이송되지 않습니다.

PRESETON은 동기화와 함께 전처리 정지를 시작합니다. 실제 위치는 정지 상태에서만 축에 할당됩니다.

PRESETON의 축이 채널에 할당되지 않으면 축 교체 동작의 축별 설정에 따라 다음 단계가 달라집니다.

MD30552 $MA_AUTO_GET_TYPE

원점 복귀 상태

기계 좌표계에 새 실제 값을 설정하면 기계 축의 원점 복귀 상태가 리셋됩니다.

DB31, ... DBX60.4/.5 = 0 (원점 복귀/동기 측정 시스템 1/2)

따라서 PRESETON은 원점이 필요하지 않은 축에만 사용할 것을 권장합니다.

원래 기계 좌표계를 복구하려면 예를 들어 가공 프로그램에서 원점 복귀 (G74) 를 활성화하여 기계 축의 측정 시스템을 다시 원점 복귀시켜야 합니다.

⚠ 주의

원점 복귀 상태 삭제

PRESETON을 사용해 기계 좌표계(MCS)에 새 실제 값을 설정하면 기계 축의 원점 복귀 상태가 '원점 복귀/동기화 안 됨'으로 리셋됩니다.

프로그래밍

구문

```
PRESETON(<axis_1>, <value_1> [, <axis_2>, <value_2>, ...
<axis_8>, <value_8>])
```

의미

PRESETON:	원점 복귀 상태를 삭제하고 실제 값 설정	
	전처리 정지:	예
	블록에서만:	예

<axis_x>:	기계 축 이름	
	유형:	AXIS
	값 범위:	채널에 정의된 기계 축 이름
<value_x>:	기계 좌표계(MCS)에서 기계 축의 새 실제 값 현재 활성 측정 시스템(인치/미터)으로 입력 활성 직경 프로그래밍(DIAMON)을 고려	
	유형:	REAL

추가 정보

NC 프로그램의 PRESETON

NC 프로그램의 PRESETON에 대한 자세한 설명은 기능 매뉴얼 기본 기능을 참고하십시오.

동기 동작의 PRESETON

동기 동작의 PRESETON에 대한 자세한 설명은 기능 매뉴얼 동기 동작에 나와 있습니다.

4.8.6 원점 복귀 상태를 유지하면서 실제 값 설정 (PRESETONS)

PRESETONS() 과정에서 기계 좌표계(MCS)로 하나 이상의 축에 새 실제 값을 설정합니다. 축의 MCS 워크 옵셋과 같은 기능을 합니다. 축은 이송되지 않습니다.

PRESETONS은 동기화와 함께 전처리 정지를 시작합니다. 실제 위치는 정지 상태에서만 축에 할당됩니다.

PRESETONS의 축이 채널에 할당되지 않으면 축 교체 작동의 축별 설정에 따라 다음 단계가 달라집니다.

MD30552 $MA_AUTO_GET_TYPE

원점 복귀 상태

PRESETONS를 사용해 기계 좌표계(MCS)에 새 실제 값을 설정하면 기계 축의 원점 복귀 상태가 변경되지 **않습니다**.

사전 조건

- **엔코더 유형**
  PRESETONS는 다음과 같은 엔코더 유형의 활성 측정 시스템에서만 사용할 수 있습니다.
  - MD30240 $MA_ENC_TYPE[<measuring system>] = 0 (시뮬레이션 엔코더)
  - MD30240 $MA_ENC_TYPE[<measuring system>] = 1 (원본 신호 엔코더)
- **원점 복귀 모드**
  PRESETONS는 다음과 같은 원점 복귀 유형의 활성 측정 시스템에서만 사용할 수 있습니다.
  - MD34200 $MA_ENC_REFP_MODE[<measuring system>] = 0 (원점 복귀 불가능)
  - MD34200 $MA_ENC_REFP_MODE[<measuring system>] = 1 (증분, 로터리 또는 직선 측정 시스템의 경우 원점 복귀: 엔코더 트랙의 제로 펄스)

프로그래밍

구문

```
PRESETONS(<axis_1>, <value_1> [, <axis_2>, <value_2>, ...
<axis_8>, <value_8>])
```

의미

PRESETONS:	원점 복귀 상태를 유지하면서 실제 값 설정	
	전처리 정지:	예
	블록에서만:	예
<axis_x>:	기계 축 이름	
	유형:	AXIS
	값 범위:	채널에 정의된 기계 축 이름
<value_x>:	기계 좌표계(MCS)에서 기계 축의 새 현재 실제 값 활성 측정 시스템 (인치/미터) 으로 입력 활성 직경 프로그래밍(DIAMON)을 고려	
	유형:	REAL

추가 정보

NC 프로그램의 PRESETON

NC 프로그램의 PRESETON에 대한 자세한 설명은 기능 매뉴얼 기본 기능을 참고하십시오.

동기 동작의 PRESETON

동기 동작의 PRESETON에 대한 자세한 설명은 기능 매뉴얼 동기 동작에 나와 있습니다.

4.8.7 공간의 세 개 측정 지점에서 프레임 계산 (MEAFRAME)

MEAFRAME은 측정 싸이클을 지원하는 기능입니다. 이 기능은 3개의 이상적인 지점과 해당 측정 지점에서 프레임을 계산합니다.

가공을 위해 공작물을 포지셔닝하면 일반적으로 공작물의 이상적인 위치를 참조하여 공작물의 직교 기계 좌표계 기준 위치를 이동 및 회전시킵니다. 정확한 가공 또는 측정을 위해 비용이 드는 물리적 부품 조정이 필요하거나 가공 프로그램에 정의된 이동을 변경해야 합니다.

이상 위치가 알려진 공간에서 세 개 지점을 샘플링하여 프레임을 정의할 수 있습니다. 터치 프로브 또는 광학 센서는 지지 플레이트 또는 프로브 볼에 고정된 특수 구멍을 정밀하게 접촉하는 샘플링에 사용됩니다.

구문

```
MEAFRAME(<ideal points>,<measuring points>,<quality>)
```

의미

MEAFRAME:	기능 호출		
<ideal points>:	2-dim. 이상적인 지점의 세 좌표가 포함된 REAL 배열		
<measuring points>:	2-dim. 측정 지점의 세 좌표가 포함된 REAL 배열		
<quality>:	프레임 계산 품질에 대한 정보를 반환하기 위한 변수		
	유형:	VAR REAL	
	값:	-1	이상 지점은 거의 직선 위에 있습니다. 프레임은 계산할 수 없습니다. 반환된 프레임 변수에는 중립 프레임이 포함됩니다.
		-2	측정 지점은 거의 직선 위에 있습니다. 프레임은 계산할 수 없습니다. 반환된 프레임 변수에는 중립 프레임이 포함됩니다.
		-4	회전 매트릭스 계산은 다른 이유로 실패했습니다.
		≥ 0.0	측정된 삼각형을 이상적인 삼각형에 맞게 변환하는 데 필요한 비틀림 합계 (지점 간의 거리)

참고

측정 품질

측정된 좌표를 회전 및 변환을 사용하여 이상적인 좌표로 매핑하려면 측정 지점에 의해 형성된 삼각형을 이상적인 삼각형과 일치해야 합니다. 이 작업은 측정된 삼각형을 이상적인 삼각형으로 다시 구성하는 데 필요한 제곱 편차 합계를 최소화하는 보정 알고리즘을 통해 수행됩니다.

유효 비틀림을 사용하여 측정 품질을 판단할 수 있기 때문에 MEAFRAME은 유효 비틀림을 추가 변수로 반환합니다.

참고

MEAFRAME으로 생성한 프레임을 ADDFRAME을 사용하여 프레임 체인의 다른 프레임으로 변환할 수 있습니다 ("ADDFRAME으로 연결" 예 참조).

예

예 1:

가공 프로그램 1:

프로그램 코드
...
DEF FRAME CORR_FRAME

측정 지점 설정:

프로그램 코드	설명
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)	; 테스트용.
DEF REAL FIT_QUALITY=0	
DEF REAL ROT_FRAME_LIMIT=5	; 부품 위치를 최대 5°까지 회전하도록 허용합니다.
DEF REAL FIT_QUALITY_LIMIT=3	; 이상 및 측정 삼각형 간에 최대 3 mm 옵셋이 허용됩니다.
DEF REAL SHOW_MCS_POS1[3]	
DEF REAL SHOW_MCS_POS2[3]	
DEF REAL SHOW_MCS_POS3[3]	

프로그램 코드	설명
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	; 첫 번째 RPY 각도 제한.
SETAL(65020)	
GOTOF NO_FRAME	
ENDIF	
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT	; 두 번째 RPY 각도 제한.
SETAL(65021)	
GOTOF NO_FRAME	
ENDIF	
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT	; 세 번째 RPY 각도 제한.
SETAL(65022)	
GOTOF NO_FRAME	
ENDIF	
N300 $P_IFRAME=CORR_FRAME	; 설정 가능한 프레임으로 샘플 프레임 활성화. ; 기하 축을 이상적인 지점에 포지셔닝하여 프레임 검사.
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=$AA_IM[X]	
N420 SHOW_MCS_POS1[1]=$AA_IM[Y]	
N430 SHOW_MCS_POS1[2]=$AA_IM[Z]	
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2]	
N510 SHOW_MCS_POS2[0]=$AA_IM[X]	
N520 SHOW_MCS_POS2[1]=$AA_IM[Y]	
N530 SHOW_MCS_POS2[2]=$AA_IM[Z]	
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2]	
N610 SHOW_MCS_POS3[0]=$AA_IM[X]	
N620 SHOW_MCS_POS3[1]=$AA_IM[Y]	
N630 SHOW_MCS_POS3[2]=$AA_IM[Z]	
N700 G500	; 설정 가능한 프레임을 제로 프레임으로 비활성화(입력 값 없음, 사전 시정).

프로그램 코드	설명
No_FRAME	; 설정 가능한 프레임을 제로 프레임으로 사전 지정하여 비활성화(입력 값 없음).
M0	
M30	

예 2: 프레임 연결

옵셋을 위한 MEAFRAME 연결

MEAFRAME 기능은 옵셋 프레임을 반환합니다. 이 옵셋 프레임이 G54와 같은 코드를 호출할 때 활성화된 설정 가능한 프레임 $P_UIFR[1]과 연결된 경우, 이송 또는 가공을 위해 추가 변환을 할 수 있도록 설정 가능한 프레임이 제공됩니다.

ADDFRAME으로 연결

프레임 체인에 포함된 이 옵셋 프레임을 다른 위치에 적용하거나 설정 가능한 프레임 전에 다른 프레임을 먼저 활성화하려면 ADDFRAME 기능을 사용하여 채널 기본 프레임 중 하나 또는 시스템 프레임에 연결할 수 있습니다.

다음은 프레임에서 활성화하지 않아야 합니다.

- MIRROR를 통한 미러링
- SCALE을 통한 배율 지정

지령치 및 실제 값에 대한 입력 파라미터는 공작물 좌표계입니다. 이 좌표는 반드시 미터 또는 인치 단위 (G71/G70) 로 지정해야 하고 제어 시스템의 기본 시스템의 반경 (DIAMOF) 을 기준으로 해야 합니다.

ADDFRAME에 대해 **자세히 알아보려면** 기능 매뉴얼 기본 기능을 참고하십시오.

4.8.8 글로벌 프레임

각 제어 시스템의 모든 채널에 대해 글로벌 프레임 1세트만 사용됩니다. 모든 채널에서 글로벌 프레임을 읽고 쓸 수 있습니다. 각 채널에서 글로벌 프레임이 활성화됩니다.

옵셋이 있는 **채널 축 및 기계 축**은 글로벌 프레임을 통해 배율을 지정하고 미러링할 수 있습니다.

형상 관계 및 프레임 체인

글로벌 프레임의 경우 축 간에 형상 관계가 없습니다. 따라서 회전을 수행하거나 형상 축 식별자를 프로그래밍할 수 없습니다.

글로벌 프레임에서 회전을 사용할 수 없습니다. 알람 18310 "Channel %1 Block %2 Frame: 회전이 허용되지 않습니다."를 통해 회전 프로그래밍이 거부됩니다.

글로벌 프레임 및 채널 관련 프레임을 연결할 수 있습니다. 결과 프레임에는 모든 축의 회전을 포함하여 모든 프레임 콤포넌트가 포함되어 있습니다. 글로벌 프레임에 회전 구성 요소를 가진 프레임 할당은 알람 "프레임: 회전이 허용되지 않습니다."를 통해 거부됩니다.

글로벌 프레임

글로벌 기본 프레임 $P_NCBFR[n]

최대 8개의 글로벌 기본 프레임을 구성할 수 있습니다.

채널 기본 프레임도 사용할 수 있습니다.

제어 시스템의 모든 채널에서 글로벌 프레임을 읽고 쓸 수 있습니다. 글로벌 프레임을 쓸 때 사용자는 채널 코디네이션을 확인해야 합니다. 이 확인 작업은 예를 들어 대기 라벨 (WAITMC) 을 사용하여 수행할 수 있습니다.

참고

장비 제조업체

글로벌 기본 프레임의 개수는 머신 데이터에 설정합니다.

추가 정보: 기능 매뉴얼 기본 기능

설정 가능한 프레임($P_UIFR[n])

설정 가능한 모든 프레임 G500, G54...G599을 전역적으로 또는 채널별로 구성할 수 있습니다.

참고

장비 제조업체

모든 설정 가능한 프레임은 머신 데이터
MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES를 사용하여 글로벌 프레임으로 재설정할 수 있습니다.

채널 축 식별자 및 기계 축 식별자는 프레임 프로그램 명령에서 축 식별자를 사용할 수 있습니다. 기하 식별자 프로그래밍은 알람과 함께 거부됩니다.

4.8.8.1 채널 관련 프레임 ($P_CHBFR, $P_UBFR)

설정 가능한 프레임 또는 기본 프레임은 가공 프로그램, PLC 또는 작업자가 OPI를 통해 읽고 쓸 수 있습니다.

미세 옵셋도 글로벌 프레임에 사용할 수 있습니다. G53, G153, SUPA, G500을 통해 채널별 프레임과 동일한 방식으로 글로벌 프레임이 억제됩니다.

장비 제조업체

채널의 기본 프레임 개수는 머신 데이터 MD28081 $MC_MM_NUM_BASE_FRAMES를 통해 설정할 수 있습니다. 표준 구성은 채널당 최소 하나의 기본 프레임에 대해 설계됩니다. 채널당 최대 8개의 기본 프레임이 지원됩니다. 8개의 기본 프레임 외에 8개의 NCU 글로벌 기본 프레임을 채널에 추가할 수 있습니다.

채널별 프레임

$P_CHBFR[n]

시스템 변수 $P_CHBFR[n]를 사용하여 기본 프레임을 읽고 쓸 수 있습니다. 기본 프레임을 작성하는 경우 G500, G54...G599 구문이 실행될 때까지 연결된 전체 기본 프레임이 활성화되지 않습니다. 변수는 주로 HMI 또는 PLC에서 기본 프레임에 대한 쓰기 작업을 저장하는 데 사용됩니다. 이러한 프레임 변수는 데이터 백업에 의해 저장됩니다.

채널에서 첫 번째 기본 프레임

배열 인덱스가 0인 기본 프레임은 사전 정의된 $P_UBFR 변수에 작성할 때 동시에 활성화되는 것이 아니라, G500, G54...G599 구문이 실행될 때만 활성화됩니다. 또한 프로그램에서 변수를 읽고 쓸 수 있습니다.

$P_UBFR

$P_UBFR은 $P_CHBFR[0]과 동일합니다. 기본적으로 채널에는 항상 하나의 기본 프레임만 존재하기 때문에 시스템 변수는 이전 버전과 호환됩니다. 채널 기본 프레임이 없는 경우 읽기/쓰기를 수행할 때 "프레임: 허용되지 않는 구문" 알람이 출력됩니다.

4.8.8.2 채널에서 프레임 활성화

채널에 활성화된 프레임은 이 프레임과 관련된 시스템 변수를 통해 가공 프로그램에서 입력합니다. 여기에는 시스템 프레임이 포함됩니다. 현재 시스템 프레임은 가공 프로그램에서 이 시스템 변수를 통해 읽고 쓸 수 있습니다.

채널에 현재 활성화된 프레임

개요

현재 시스템 프레임	다음의 경우:
$P_PARTFRAME	TCARR 및 PAROT

$P_SETFRAME	사전 설정된 실제값 메모리 및 스크래칭
$P_EXTFRAME	외부 워크 옵셋
$P_NCBFRAME[n]	현재 글로벌 기본 프레임
$P_CHBFRAME[n]	현재 채널 기본 프레임
$P_BFRAME	현재 1 채널의 기본 프레임
$P_ACTBFRAME	전체 기본 프레임
$P_CHBFRMASK 및 $P_NCBFRMASK	전체 기본 프레임
$P_IFRAME	현재 설정 가능한 프레임
현재 시스템 프레임	다음의 경우:
$P_TOOLFRAME	TOROT 및 TOFRAME
$P_WPFRAME	공작물 원점
$P_TRAFRAME	변환
$P_PFRAME	현재 프로그래밍 프레임
현재 시스템 프레임	다음의 경우:
$P_CYCFRAME	싸이클
P_ACTFRAME	현재 전체 프레임
FRAME 체인	현재 프레임은 전체 기본 프레임으로 구성됩니다.

$P_NCBFRAME [n] 현재 글로벌 기본 프레임

시스템 변수 $P_NCBFRAME[n]을 사용하여 현재 글로벌 기본 프레임 필드 요소를 읽고 쓸 수 있습니다. 결과 전체 기본 프레임은 채널에서 쓰기 프로세스를 통해 계산됩니다.

수정된 프레임은 프레임이 프로그래밍된 채널에서만 활성화됩니다. 제어의 모든 채널에 대해 프레임을 수정하려면 $P_NCBFR [n] 및 $P_NCBFRAME [n]을 동시에 작성해야 합니다. 다른 채널은 예를 들어 G54 등으로 프레임을 활성화해야 합니다. 기본 프레임을 쓸 때마다 전체 기본 프레임이 다시 계산됩니다.

$P_CHBFRAME[n] 현재 채널 기본 프레임

시스템 변수 $P_CHBFRAME[n]을 사용하여 현재 채널 기본 프레임 필드 요소를 읽고 쓸 수 있습니다. 그 결과인 전체 기본 프레임은 채널에서 쓰기 프로세스를 통해 계산됩니다. 기본 프레임을 쓸 때마다 전체 기본 프레임이 다시 계산됩니다.

채널의 $P_BFRAME 현재 첫 번째 기본 프레임

사전 정의된 프레임 변수 $P_BFRAME을 사용하여 채널에서 유효한 배열 인덱스가 0인 현재 기본 프레임을 가공 프로그램에서 읽고 쓸 수 있습니다. 기록된 기본 프레임이 즉시 계산에 포함됩니다.

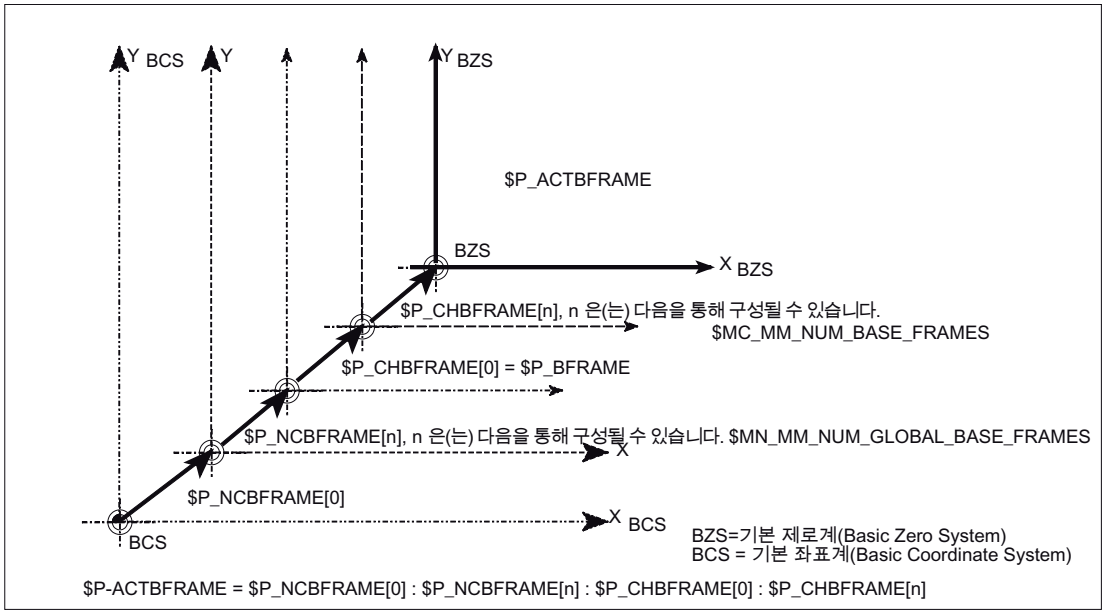
$P_BFRAME은 $P_CHBFRAME[0]과 동일합니다. 시스템 변수에는 항상 유효한 기본값이 포함됩니다. 채널 기본 프레임이 없는 경우 읽기/쓰기를 수행할 때 "프레임: 허용되지 않는 구문" 알람이 출력됩니다.

$P_ACTBFRAME 전체 기본 프레임

$P_ACTFRAME 변수는 연결된 전체 기본 프레임을 결정합니다. 변수는 읽기 전용입니다.

$P_ACTFRAME은 다음에 해당합니다.

$P_NCBFRAME[0] : ... : $P_NCBFRAME[n] : $P_CHBFRAME[0] : ... : $P_CHBFRAME[n].



$P_CHBFRMASK 및 $P_NCBFRMASK 전체 기본 프레임

사용자는 시스템 변수 $P_CHBFRMASK 및 $P_NCBFRMASK를 사용하여 "전체" 기본 프레임 계산에 어떤 프레임을 포함시킬지 선택할 수 있습니다. 변수는 프로그램에 프로그래밍하고 OPI를 통해서만 읽을 수 있습니다. 변수 값은 비트 마스크로 해석되며 계산에 포함될 $P_ACTFRAME 의 기본 프레임 필드 요소를 지정합니다.

$P_CHBFRMASK를 사용하여 채널별 기본 프레임을 지정하고, $P_NCBFRMASK를 이용해 글로벌 기본 프레임을 계산에 포함할 수 있습니다.

전체 기본 프레임과 전체 프레임은 변수를 프로그래밍하여 다시 계산합니다. 기본 설정으로 리셋한 후 $P_CHBFRMASK와 $P_NCBFRMASK의 값은 다음과 같습니다.

$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK

$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK

예:

```
$P_NCBFRMASK = 'H81' ;$P_NCBFRAME[0] : $P_NCBFRAME[7]
$P_CHBFRMASK = 'H11' ;$P_CHBFRAME[0] : $P_CHBFRAME[4]
```

$P_IFRAME 현재 설정 가능한 프레임

사전 정의된 프레임 변수 $P_IFRAME을 사용하여 채널에서 유효한 현재 설정 가능한 프레임을 가공 프로그램에서 읽고 쓸 수 있습니다. 기록된 설정 가능한 프레임이 즉시 계산에 포함됩니다.

글로벌 설정 가능한 프레임의 경우 프레임이 프로그래밍된 채널에서만 수정된 프레임이 작동합니다. 제어의 모든 채널에 대해 프레임을 수정하려면 $P_UIFR[n] and $P_IFRAME을 동시에 작성해야 합니다. 나머지 채널들은 예를 들어 G54 등을 통해 해당 프레임을 활성화해야 합니다.

$P_PFRAME 현재 프로그래밍 프레임

$P_PFRAME은 TRANS/ATRANS, G58/G59, ROT/AROT, SCALE/ASCALE, MIRROR/AMIRROR를 프로그래밍하거나 CTRANS, CROT, CMIRROR, CSCALE을 프로그래밍 프레임에 지정한 결과인 프로그래밍 프레임입니다.

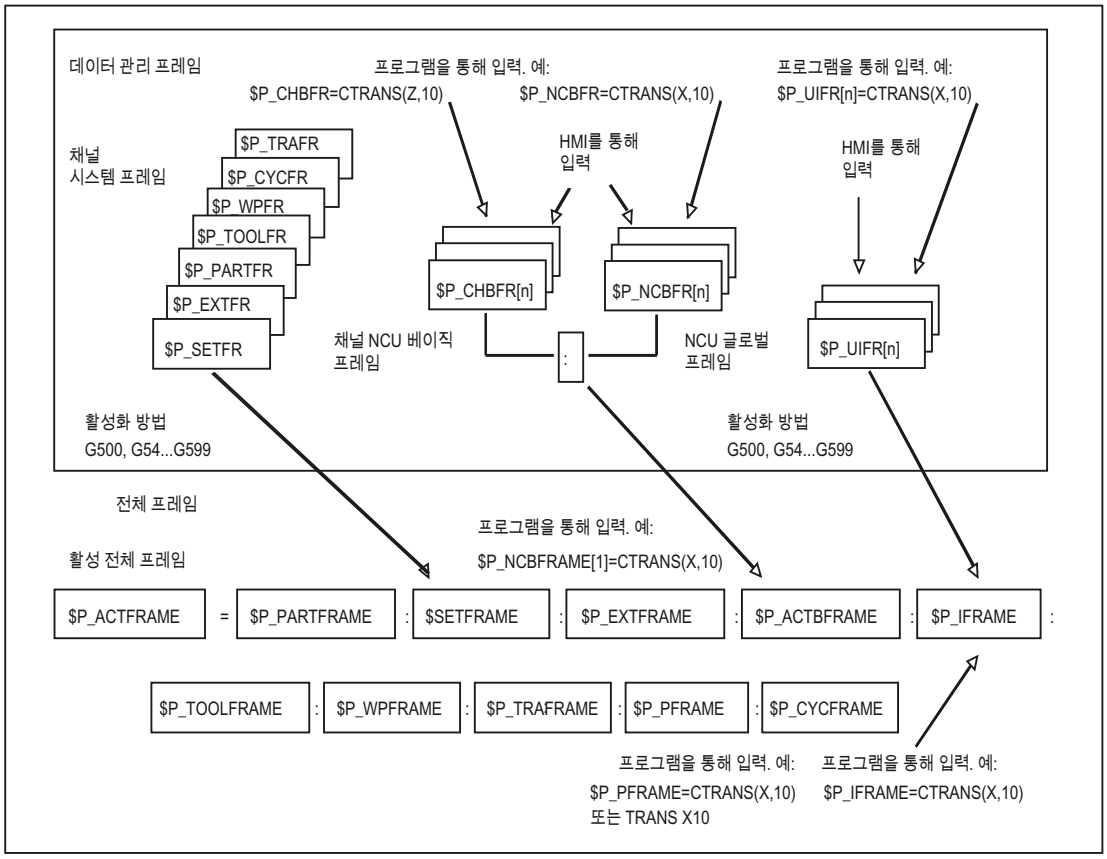
설정 가능한 제로 시스템(SZS)과 공작물 좌표계(WCS) 사이에서 기준이 되는 현재 프로그래밍 프레임 변수입니다.

P_ACTFRAME 현재 전체 프레임

결과 현재 전체 프레임 $P_ACTFRAME은 모든 기본 프레임, 현재 설정 가능한 프레임 및 프로그래밍 프레임의 체인입니다. 프레임 콤포넌트가 변경될 때마다 항상 현재 프레임이 업데이트됩니다.

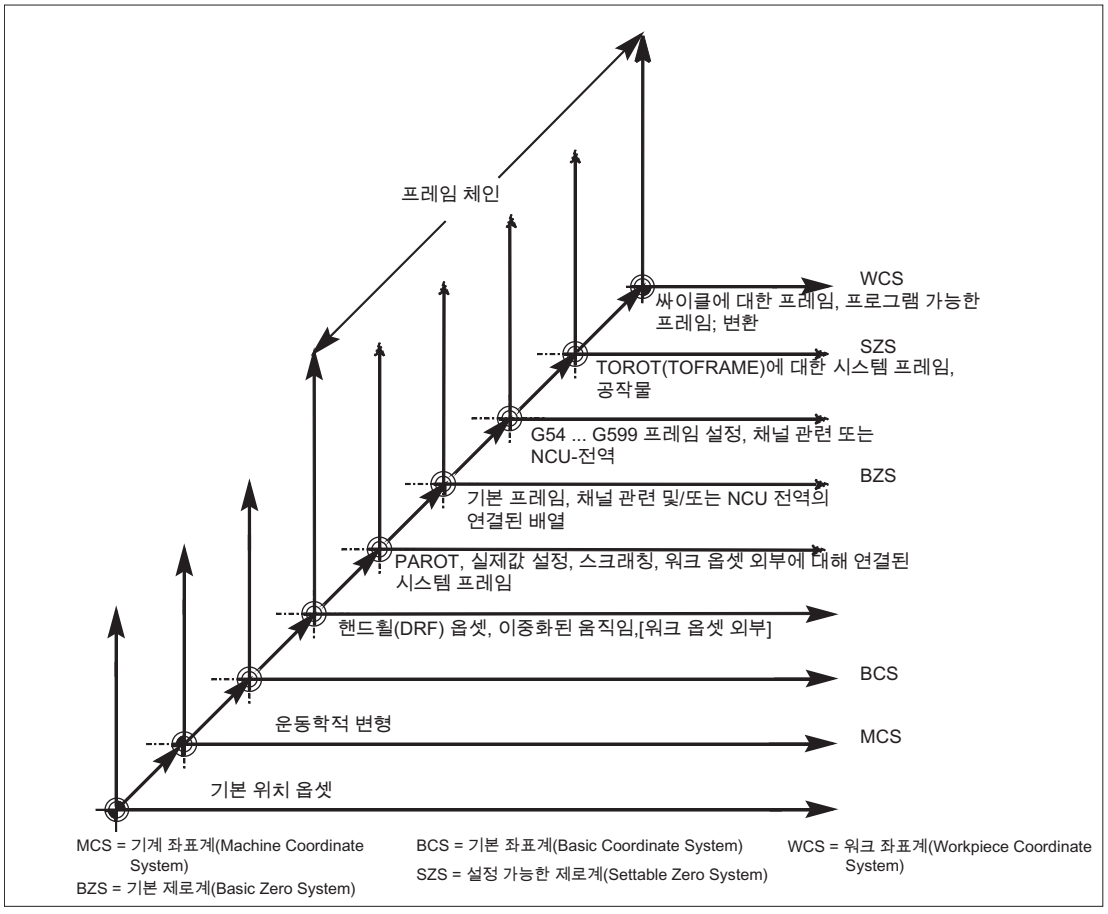
$P_ACTFRAME은 다음에 해당합니다.

```
$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :
$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME
```


데이터 관리 프레임
프로그램을 통해 입력. 예:
$P_CHBFR=CTRANS(Z,10)
$P_NCBFR=CTRANS(X,10)
프로그램을 통해 입력. 예:
$P_UIFR[n]=CTRANS(X,10)
채널
시스템 프레임
$P_TRAFR
$P_CYCFR
$P_WPFR
$P_TOOLFR
$P_PARTFR
$P_EXTFR
$P_SETFR
HMI를 통해
입력
HMI를 통해
입력
$P_CHBFR[n]
$P_NCBFR[n]
$P_UIFR[n]
채널 NCU 베이직
프레임
:
NCU 글로벌
프레임
활성화 방법
G500, G54...G599
활성화 방법
G500, G54...G599
전체 프레임
프로그램을 통해 입력. 예:
$P_NCBFRAME[1]=CTRANS(X,10)
활성 전체 프레임
$P_ACTFRAME
=
$P_PARTFRAME
:
$SETFRAME
:
$P_EXTFRAME
:
$P_ACTBFRAME
:
$P_IFRAME
:
$P_TOOLFRAME
:
$P_WPFRAME
:
$P_TRAFRAME
:
$P_PFRAME
:
$P_CYCFRAME
프로그램을 통해 입력. 예:
$P_PFRAME=CTRANS(X,10)
또는 TRANS X10
프로그램을 통해 입력. 예:
$P_IFRAME=CTRANS(X,10)

FRAME 체인

현재 프레임은 위에서 언급한 현재 전체 프레임에 따라 전체 기본 프레임, 설정 가능한 프레임, 시스템 프레임 및 프로그래밍 프레임으로 구성됩니다.



4.9 변환

4.9.1 일반적인 변환 유형 프로그래밍

변환은 다양한 기계 키네마틱스에 대한 제어를 조정합니다. 변환은 NC 프로그램의 기능 호출에 의해 활성화됩니다. 해당 파라미터를 사용하여 공간 상의 공구 오리엔테이션 및 로터리 축의 오리엔테이션 이동 모두 정의할 수 있습니다.

오리엔테이션 변환의 경우(3축, 4축 및 5축 변환), 프로그래밍된 위치 데이터는 항상 공구의 팁을 참조하며, 이 공구 팁은 공간 상의 가공 표면에 대해 직각 방향으로 추적됩니다. 직교 좌

표는 기본 좌표계에서 기계 좌표계로 변환되어 형상 축과 관련됩니다. 이것은 작동 지점을 설명합니다. 가상의 로터리 축은 공간 상의 공구 오리엔테이션을 설명합니다.

키네마틱 변환의 경우(TRANSMIT, TRACYL, TRAANG) 직교 좌표계에서 위치를 프로그래밍할 수 있습니다. 제어 시스템이 직교 좌표계의 프로그램된 이송을 실제 기계 축의 이송으로 변환합니다.

프로그래밍

오리엔테이션 변환 활성화

```
TRAORI(<변환 수>, <오리엔테이션 벡터>, <로터리 축 옵셋>)
```

키네마틱 변환 활성화

```
TRANSMIT(<변환 수>)
```

```
TRACYL(<작업 직경>,<변환 수>)
```

```
TRAANG(<경사축 각도>,<변환 수>)
```

활성 변환 비활성화

```
TRAFOOF
```

오리엔테이션 변환

3축, 4축 및 5축 변환 (TRAORI)

기계의 작동 영역에서 공간 상에 구성되는 표면 가공의 최적화를 위해 기계 공구는 세 가지 직선 축 X, Y 및 Z 이외에 다른 축이 필요합니다. 추가 축은 공간 상의 방향을 설명하며 이후 섹션에서 오리엔테이션 축이라고 합니다. 이러한 추가 축은 키네마틱이 서로 다른 네 가지 기계 유형에서 로터리 축으로 사용할 수 있습니다.

1. 2축 스위블 헤드
   예: 고정 공구 테이블을 가지고 있으며 직선 축과 평행한 1개의 로터리 축을 가진 카르단(Cardanic) 공구 헤드.
2. 2축 로터리 테이블
   예: 두 축을 중심으로 회전하는 공구 테이블을 가진 고정 스위블 헤드
3. 단일축 스위블 헤드 및 단일축 로터리 테이블
   예: 한 축을 중심으로 회전하는 공구 테이블과 회전 공구가 있는 로터리 스위블 헤드
4. 2축 스위블 헤드 및 단일축 로터리 테이블
   예: 한 축을 중심으로 회전할 수 있는 공구 테이블과 스스로 회전하는 공구를 가진 로터리 스위블 헤드

3축 및 4축 변환은 5축 변환의 특별한 유형이며 5축 변환과 동일한 방법으로 프로그래밍됩니다.

일반 3/4/5/6축 변환의 기능상 범위는 다른 모든 오리엔테이션 변환과 마찬가지로, 직교 로터리 축에 대한 변환 및 카르단 밀링 헤드에 대한 변환 모두에 적합하며 TRAORI를 통해 이러한 네 가지 기계 유형에 대해 활성화할 수도 있습니다. 일반 5/6축 변환의 경우 공구 오리엔테이션은 추가로 세 번째 자유도가 있기 때문에 공구 방향에 상대적인 공간에서 공구 축을 중심으로 공구를 회전할 수 있습니다.

추가 정보: 기능 매뉴얼 변환

운동성과 관계 없이 초기 공구 방향 설정

ORIRESET

오리엔테이션 변환이 TRAORI를 이용해 활성화된 경우 ORIRESET을 사용하여 옵션 파라미터 A, B, C를 사용하여 최대 3개 오리엔테이션 축에 대한 초기 설정을 지정할 수 있습니다. 프로그래밍된 파라미터는 변환에 의해 지정된 오리엔테이션 축 순서에 따라 로터리 축에 할당됩니다. ORIRESET(A, B, C)을 프로그래밍하면 오리엔테이션 축이 선형적이고 동기식으로 현재 위치에서 지정된 초기 설정 위치로 이동합니다.

운동학적 변환

TRANSMIT 및 TRACYL

선삭 기계에서의 밀링의 경우 동의된 변환에 대해 다음 가공 유형을 활성화할 수 있습니다.

- TRANSMIT를 통해 선삭 가공 클램프에서 표면 가공
- TRACYL을 이용해 원통형 본체에서 임의적 흐름의 홈 가공

TRAANG

절입 축의 경사 절입 설정이 필요한 경우(예: 연삭 기술) TRAANG을 사용하여 선언된 변환에 대한 구성 가능한 각도를 프로그래밍할 수 있습니다.

직교 PTP 이송

키네마틱 변환에는 최대 8개의 서로 다른 접점(STAT=...)를 프로그래밍할 수 있는 "직교 PTP 이동"도 포함됩니다. 위치는 직교 좌표계에 프로그래밍되지만 기계 좌표에서 기계 이동이 발생합니다.

추가 정보: 기능 매뉴얼 변환

연결된 변환

두 개의 변환을 차례대로 전환할 수 있습니다. 여기서 두 번째로 연결된 변환의 경우 축의 이동 부분을 첫 번째 변환에서 가져옵니다.

첫 번째 변환은 다음과 같습니다.

- 오리엔테이션 변환 TRAORI
- 극좌표 변환 TRANSMIT
- 실린더 변환 TRACYL
- 경사 축 변환 TRAANG

두 번째 변환은 경사 축 TRAANG이어야 합니다.

4.9.1.1 변환을 위한 방향 이동

이송 및 오리엔테이션 이동

프로그래밍된 오리엔테이션으로 이송은 주로 기계 유형에 의해 결정됩니다. TRAORI를 통한 3축, 4축 및 5축 유형 변환의 경우 로터리 축 또는 피봇 직선 축은 공구의 오리엔테이션 이동을 설명합니다.

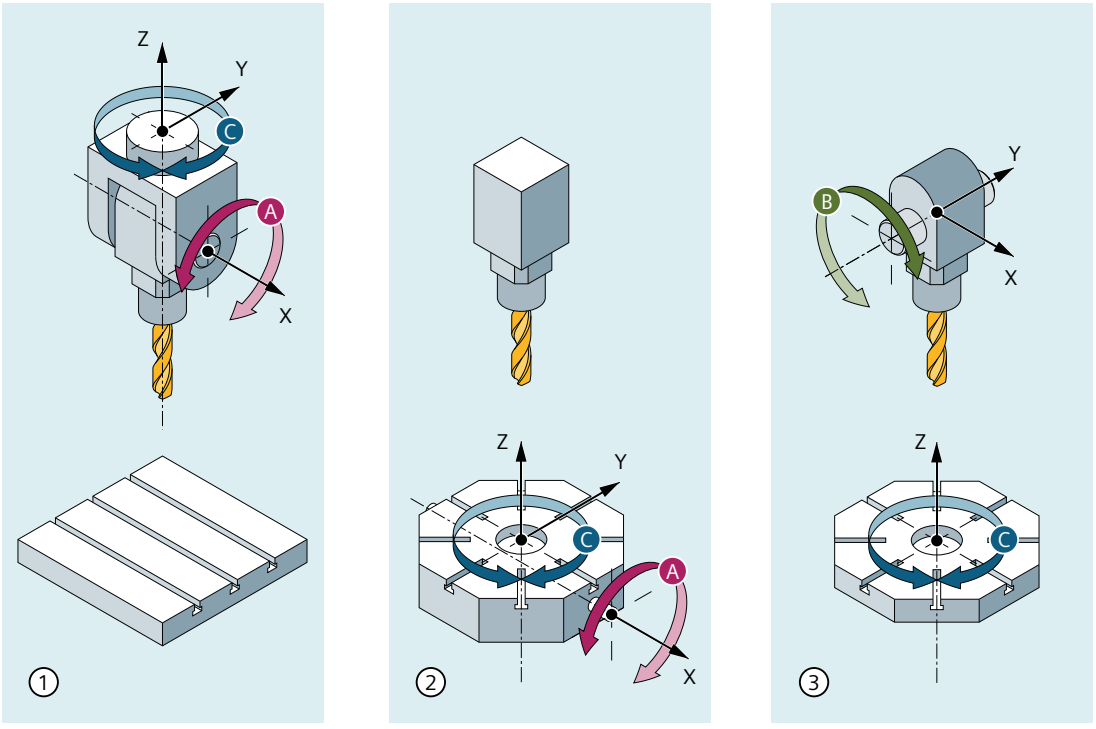
오리엔테이션을 변환할 로터리 축의 위치를 변경하면 나머지 기계 축에 대해 이동 보정이 유도됩니다. 공구 팁의 위치는 고정되어 있습니다.

공구의 오리엔테이션 이동은 Euler 또는 RPY 각도, 방향 또는 표면 법선 벡터, 테이퍼의 회전 축 또는 테이퍼 주변 표면의 중간 오리엔테이션에 대해 법선 방향 벡터를 입력하여 해당 분야에 적절한 수직 축의 로터리 축 식별자 A..., B..., C...를 사용하여 프로그래밍할 수 있습니다.

TRANSMIT, TRACYL 및 TRAANG을 통한 좌표계 변환의 경우 컨트롤러가 프로그래밍된 직교 좌표계 이송을 실제 기계 축의 이송에 매핑합니다.

3축, 4축 및 5축 변환에 대한 기계 운동 TRAORI

최대 2개의 로터리 축을 사용하여 공구 또는 공구 테이블을 회전할 수 있습니다. 스위블 헤드 및 로터리 테이블 (각각의 경우 단일 축) 조합도 가능합니다.



① 축 순서 CA의 기계 유형 1
- 공구 홀더, X축과 Z축에 대해 회전 가능
- 공작물 테이블 고정됨

② 축 순서 AC의 기계 유형 2
- 공구 홀더 고정됨
- 공작물 테이블, X축과 Z축에 대해 회전 가능

③ 축 순서 BC의 기계 유형 3
- 공구 홀더, Y축에 대해 회전 가능
- 공작물 테이블, Z축에 대해 회전 가능

기계 유형	오리엔테이션 프로그래밍
3축 변환 기계 유형 1 및 2	로터리 축과 수직 방향인 평면에서만 공구 오리엔테이션 프로그래밍이 가능합니다. **2개**의 이송 축 (직선 축) 및 **1개**의 회전 축 (로터리 축)
4축 변환 기계 유형 1 및 2	로터리 축과 수직 방향인 평면에서만 공구 오리엔테이션 프로그래밍이 가능합니다. **3개**의 이송 축 (직선 축) 및 **1개**의 회전 축 (로터리 축)
5축 변환 기계 유형 3 단일 축 스위블 헤드 및 단일 축 로터리 테이블	오리엔테이션 변환 프로그래밍. **3개**의 직선 축 및 **2개**의 직교 로터리 축을 가진 구조. 로터리 축은 3개의 직선 축 중 2개와 평행합니다. 첫 번째 로터리 축은 2개의 직교 직선 축에 의해 이동됩니다. 이 축은 공구와 함께 세 번째 직선 축을 회전시킵니다. 두 번째 로터리 축은 공작물을 회전합니다.

일반 5/6축 변환

기계 유형	오리엔테이션 변환 프로그래밍
일반 5/6축 변환 기계 유형 4 공구 자체 및 단일 축 로터리 테이블 주위를 회전하는 공구를 가진 2축 스위블 헤드	오리엔테이션 변환 프로그래밍. **3개**의 직선 축 및 **3개**의 직교 로터리 축을 가진 구조. 로터리 축은 3개의 직선 축 중 2개와 평행합니다. 첫 번째 로터리 축은 2개의 직교 직선 축에 의해 이동됩니다. 이 축은 공구와 함께 세 번째 직선 축을 회전시킵니다. 두 번째 로터리 축은 공작물을 회전합니다. 또한 기본 공구 오리엔테이션도 THETA 로터리 각도로 공구 자체 주위를 회전하는 추가 공구 회전을 통해 프로그래밍할 수 있습니다.

"일반 3축, 4축 및 5/6축 변환"을 호출하면 공구의 기본 오리엔테이션도 전달할 수 있습니다. 로터리 축 방향 측면에서 제한은 더 이상 적용되지 않습니다. 로터리 축이 서로 간에 정확하게 수직을 이루지 않거나 기존 로터리 축이 직선 축과 정확하게 평행하지 않을 경우 "일반 5/6축 변환"이 공구 오리엔테이션 측면에서 최상의 결과를 제공할 수 있습니다.

좌표계 변환 TRANSMIT, TRACYL 및 TRAANG

선반에서 밀링 또는 연삭 가공 동안 경사 절입을 위해 설정 가능한 축의 경우 기본적으로 다음 축 정렬이 선언된 변환에 따라 적용됩니다.

TRANSMIT	**극점 변환 활성화**
선삭 가공 클램프에서 표면 가공	로터리 축 회전 축과 수직인 절입 축 회전 축과 평행한 세로 축

TRACYL	**원통 표면 변환 활성화**
원통형 본체의 경로를 따라 홈 가공	로터리 축 회전 축과 수직인 절입 축 회전 축과 평행한 세로 축

TRAANG	**절입 축 변환 활성화**
사면 절입 축을 사용하여 가공	로터리 축 각도 설정이 가능한 절입 축 회전 축과 평행한 수평 축

직교 PTP 이송

기계가 기계 좌표에서 이동하고 다음을 통해 프로그래밍됩니다:

TRAORI	**변환 활성화**
PTP 포인트 투 포인트 이동	직교 좌표계 (MCS) 로 위치 이동
CP	BCS에서 직교 축의 경로 이동
STAT	연결 조인트의 위치는 변환에 종속됩니다.
TU	축이 최단 경로를 통해 이송되는 각도

일반 5/6축 변환을 통한 PTP 이송

기계가 기계 좌표 및 공구 오리엔테이션을 사용하여 이동하고 여기서 이동은 운동 또는 방향 벡터와 관계 없이 라운드 축 위치 및 Euler 및/또는 RPY 각도 벡터를 사용하여 프로그래밍할 수 있습니다.

이러한 경우 라운드 축 보간, 대형 원 보간을 통한 벡터 보간 또는 테이퍼의 주변 표면에 대한 오리엔테이션 벡터 보간을 수행할 수 있습니다.

예: 범용 밀링 헤드에 대한 3축 - 5축 변환

최소 5개의 축으로 구성된 기계:

- 작업 영역에서 현재 위치에서 임의의 위치로 3개의 이송 축이 직선으로 이동합니다.
- 설정 가능한 각도 (일반적으로 45도) 로 정렬된 2개의 로터리 스위블 축을 통해 공구는 45도로 설정된 공간 상의 위치로 회전할 수 있습니다.

4.9.1.2 방향 변환 개요 TRAORI

TRAORI와 함께 사용 가능한 프로그래밍 유형

기계 유형	활성 변환 TRAORI를 통한 프로그래밍
기계 유형 1, 2 또는 3 2축 스위블 헤드 또는 2축 로타리 테이블 또는 단일 축 스위블 헤드와 단일 축 로타리 테이블의 조합.	방향 축의 축 순서 및 공구의 방향은 기계 운동 또는 기계 운동과 관계 없이 프로그램 가능한 방향으로 **공작물 관련** 기준에 따라 기계 데이터를 사용하여 **기계 관련** 기준으로 구성할 수 있습니다. 기준 시스템의 방향 축에 대한 회전 방향은 다음을 통해 프로그래밍됩니다: - ORIMKS 기준 시스템 = 기계 좌표계 - ORIWKS 기준 시스템 = 공작물 좌표계 기본 설정은 ORIWKS입니다. 다음을 사용하여 방향 축 프로그래밍: 기계 축 위치 A, B, C는 Euler 각도(표준)를 통해 - ORIEULER에 의해 A2, B2, C2 각도 프로그래밍 가상 축을 향함. - RPY 각도를 통한 ORIRPY, - 가상 방향 축을 통한 ORIVIRT1, 첫 번째 정의. - 가상 방향 축을 통한 ORIVIRT2, 두 번째 정의. 보간 유형 간 차이점: **선형 보간** - 방향 축 또는 기계 축의 ORIAXES **대형 반경 원 보간**(방향 벡터의 보간) - 방향 축에서 ORIVECT, 벡터 구성 요소(방향/표면 법선)의 A3, B3, C3을 지정하여 방향 축 프로그래밍, 결과 공구 방향 프로그램, 블록 시작부에서 벡터 표면 법선의 A4, B4, C4, 블록 종단부에서 표면에 수직인 벡터의 A5, B5, C5, 공구 방향에 대한 LEAD 선행 각도, 공구 방향에 대한 TILT 경사 각도

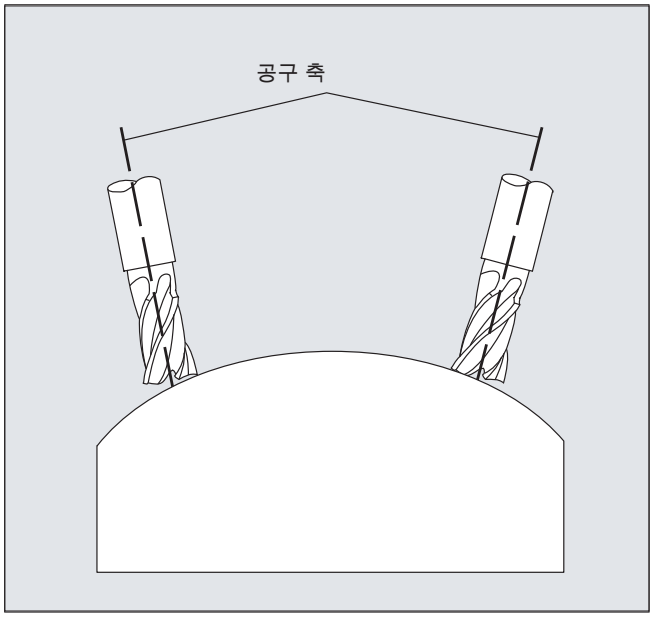
기계 유형	활성 변환 TRAORI를 통한 프로그래밍
	테이퍼 주변 표면에 대한 **방향 벡터의 보간**, 보간을 사용하여 **공간 상의 모든 위치에서** 테이퍼 주변 표면에 대한 방향 변경: - 평면에서 ORIPLANE(대형 반경 원 보간) - 시계 방향으로 테이퍼 주변 표면에 대한 ORICONCW, - 시계 반대 방향으로 테이퍼 주변 표면에 대한 ORICONCCW, A6, B6, C6 방향 벡터(테이퍼 회전 축) - 다음을 사용하여 테이퍼 주변 표면에 대한 OICONIO 보간: A7, B7, C7 중간 벡터(초기 및 최종 방향) 또는 - 테이퍼의 주변 표면에 대한 ORICONTO, 법선 방향 변환 다음을 통해 **경과** 관련 방향 변경 - 최대 5차의 PO[XH]=(xe, x2, x3, x4, x5) 방향 다항식을 사용하여 2 개의 접점 이동에 대한 ORICURVE 지정 최대 5차의 PO[YH]=(ye, y2, y3, y4, y5) 방향 다항식 최대 5차의 PO[ZH]=(ze, z2, z3, z4, z5) 방향 다항식 - 다음에 해당하는 공구의 A8, B8, C8 방향 재지정 단계를 통한 방향 특성의 ORIPATHS 유연화: 후퇴 이동 동안 공구의 방향 및 경로 길이

기계 유형	**활성 변환 TRAORI를 통한 프로그래밍**
기계 유형 1 및 3 공구 자체 주위를 추가로 회전하는 공구의 기타 기계 유형에는 세 번째 로타리 축이 필요합니다. 예를 들어 일반 6축 변환과 같은 방향 변환. 방향 벡터의 회전.	LEAD 각도를 가진 공구 방향에 대한 **회전** 프로그래밍, 각도는 표면 법선 벡터 기준 최대 5차 다항식의 PO[PHI] 프로그래밍, 경로 법선(Z 방향)에 대한 TILT 각도 회전, 최대 5차 다항식의 PO[PSI] 프로그래밍, 회전 THETA 각도(Z에서 공구 방향에 대한 회전) THETA= 블록 끝에 도달한 값 THETA=AC(...) 차원에 대한 절대 비모달식 전환 THETA=IC(...) 연결 차원에 대한 비모달식 전환, THETA=Θ_e는 프로그래밍된 각도 G90/G91 보간, PO[THT]=(..) 최대 5차 다항식의 프로그래밍, 회전 벡터 프로그래밍 - ORIROTA 회전, 절대값 - ORIROTR 상대 회전 벡터 - ORIROTT 법선 회전 벡터
방향 경로 기준 방향은 경로에 대해 법선 방향인 로타리 벡터의 경로 또는 회전을 기준으로 변경됩니다.	다음을 사용하여 **경로 기준** 방향 변경 - ORIPATH 경로 기준 공구 방향 - ORIPATHS 또한 회전 벡터의 방향 특성 프로그래밍에 blip가 있는 경우 - ORIROTC 법선 회전 벡터, 경로 법선 방향 회전

4.9.2 3축, 4축 및 5축 변환 (TRAORI)

4.9.2.1 유니버셜 공구 헤드의 일반적인 관계

3D 곡면을 가공할 때 최적의 절삭 조건을 구현하기 위해서는 공구의 설정 각도를 다양하게 바꿀 수 있어야 합니다.



이를 실행하기 위한 기계 설계가 축 데이터에 저장됩니다.

5축 변환

Cardanic 공구 헤드

세 개의 직선 축(X, Y, Z) 및 두 개의 오리엔테이션 축(C, A)이 여기서 공구의 설정 각도 및 작동 지점을 정의합니다. 두 개의 오리엔테이션 축 중 하나는 예 A처럼 경사축(대부분의 경우 45°)으로 생성됩니다.

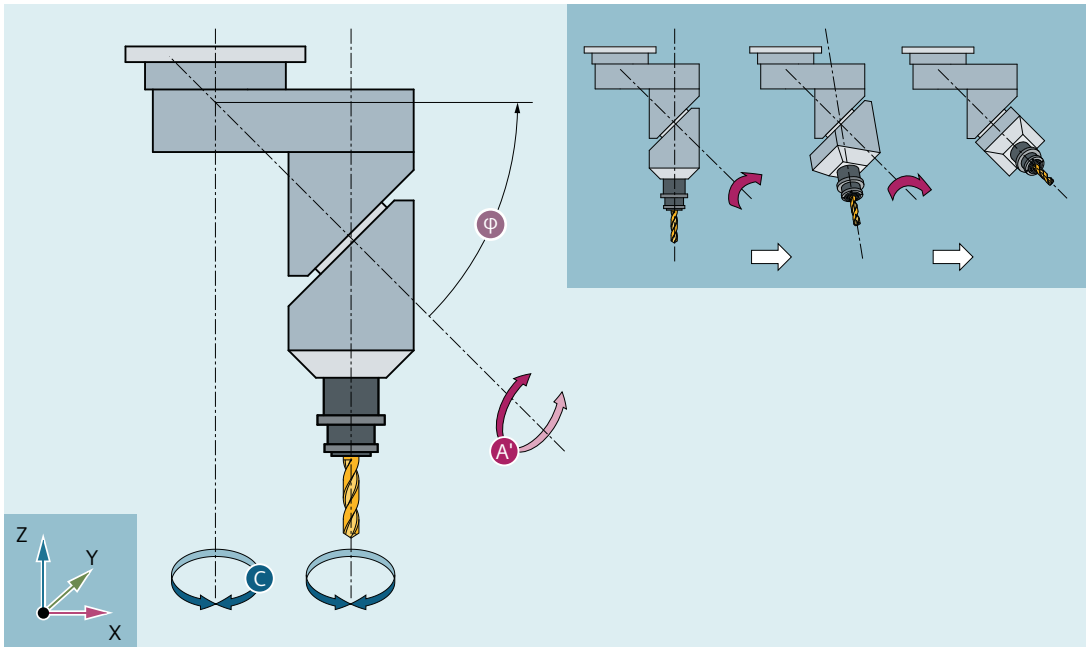
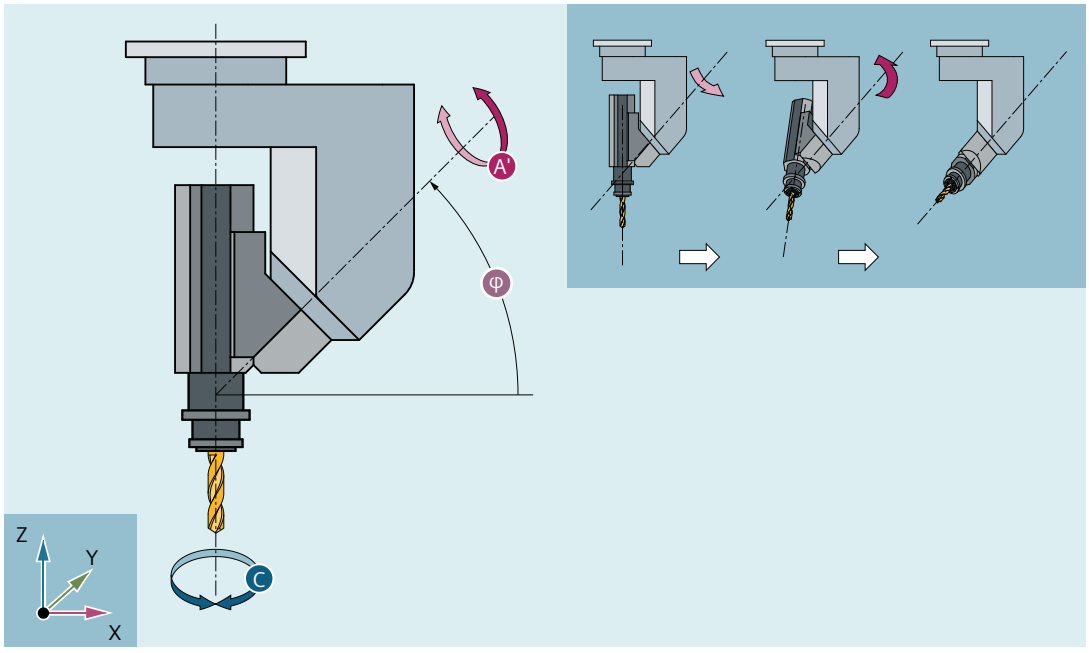


그림 4-3 Cardanic 공구 헤드 변용 1

여기 표시된 예에서 Cardanic 공구 헤드를 통한 CA 기계 운동을 그림으로 표현한 정렬을 확인할 수 있습니다.

오리엔테이션 축의 축 순서 및 공구의 방향은 기계 운동에 따라 기계 데이터를 사용하여 구성할 수 있습니다(→ 장비 제조업체).



이 예에서 A'는 X축을 기준으로 φ 각도 아래에 위치합니다.

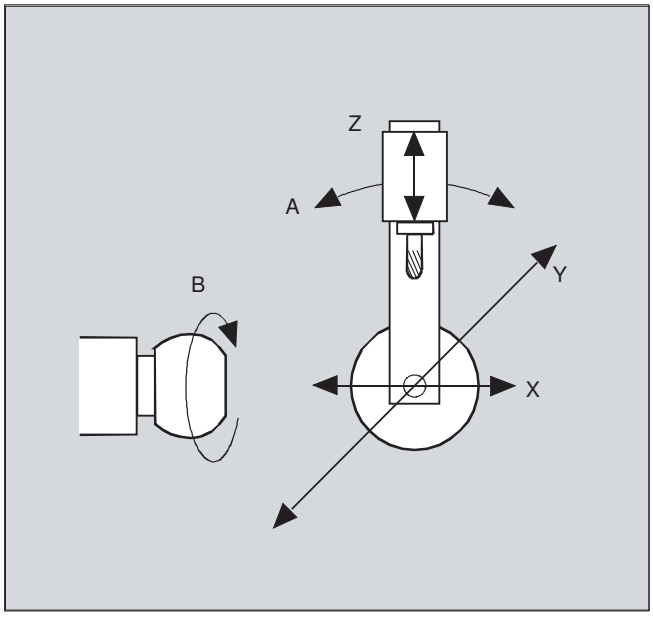
일반적으로 다음과 같은 가능한 관계가 적용됩니다:

A'는 다음 축을 기준으로 φ 각도 아래 위치 X 축
B'는 다음 축을 기준으로 φ 각도 아래 위치 Y 축
C'는 다음 축을 기준으로 φ 각도 아래 위치 Z 축

φ 각도는 기계 데이터를 사용하여 0° ~ +89° 범위에서 구성할 수 있습니다.

직선 축 회전

이 동작은 이동하는 공작물 및 이동하는 공구를 정렬하는 것입니다. 운동 구조는 세 개의 직선 축(X, Y, Z) 및 두 개의 직교 방향으로 정렬된 회전 축으로 구성됩니다. 예를 들어 첫 번째 로터리 축은 두 개의 직선 축의 복합 슬라이드 위에서 이동하고 공구는 세 번째 직선 축에 평행하게 위치합니다. 두 번째 로터리 축은 공작물을 회전합니다. 세 번째 직선 축(회전 축)은 복합 슬라이드 평면에 위치합니다.



로터리 축의 축 순서 및 공구의 방향은 기계 운동에 따라 기계 데이터를 사용하여 구성할 수 있습니다.

다음 관계가 가능합니다:

축:	**축 순서:**
첫 번째 로터리 축	A A B B C C
두 번째 로터리 축	B C A C A B
회전한 직선 축	Z Y Z X Y X

공구의 오리엔테이션 방향에 대한 구성 가능한 축 순서에 대해 **자세히 알아보려면** 기능 매뉴얼 변환을 참고하십시오.

4.9.2.2 3축, 4축 및 5축 변환 (TRAORI)

사용자가 2개 또는 3개의 변환 축과 2개의 로터리 축을 구성할 수 있습니다. 변환에서 로터리 축은 오리엔테이션 평면에 대해 수직인 것으로 간주합니다.

공구의 오리엔테이션은 로터리 축과 평행인 평면에서만 가능합니다. 변환은 이동 가능한 공구 및 이동 가능한 공작물을 가진 기계 유형을 지원합니다.

3축 및 4축 변환은 5축 변환과 동일한 방법으로 구성 및 프로그래밍됩니다.

구문

```
TRAORI(<n>)
TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)
TRAFOOF
```

의미

TRAORI:	첫 번째 지정된 오리엔테이션 변환을 실행합니다.	
TRAORI(<n>):	n에 의해 지정된 오리엔테이션 변환을 실행합니다.	
<n>:	변환의 번호	
	값:	1 또는 2
	예: TRAORI(1)이 오리엔테이션 변환 1을 활성화합니다.	
<X>,<Y>,<Z>:	공구가 향하고 있는 오리엔테이션 벡터의 구성 요소	
<A>,<B>:	로터리 축의 프로그램 옵셋	
TRAFOOF:	변환 비활성화	

공구 오리엔테이션

공구에 대해 선택한 오리엔테이션에 따라 공구 길이 옵셋이 공구 오리엔테이션에서 작동하도록 활성 작업 평면 (G17, G18, G19) 을 NC 프로그램에서 설정해야 합니다.

참고

변환이 동작되면 위치 데이터 (X, Y, Z)는 항상 공구의 팁을 기준으로 합니다. 변환에서 로터리 축의 위치를 변경하면 나머지 기계 축의 동작이 보정됩니다. 즉, 공구 팁의 위치는 변경되지 않습니다.

오리엔테이션 변환은 항상 공구 팁에서 공구 어댑터를 향합니다.

오리엔테이션 축 옵셋

오리엔테이션 변환이 활성화되면 오리엔테이션 축에 대해 직접 추가 옵셋을 프로그래밍할 수 있습니다.

정확한 순서대로 프로그래밍할 경우 파라미터를 생략할 수 있습니다.

예:

`TRAORI(, , , ,A,B)` ; 옵셋을 하나만 입력하는 경우

직접 프로그래밍에 대한 대안으로 오리엔테이션 축의 추가 옵셋도 현재 활성 워크 옵셋에서 자동으로 전송할 수 있습니다. 이송은 머신 데이터에서 구성됩니다.

예

```
TRAORI(1,0,0,1)          ; 공구의 기본 오리엔테이션은 Z 방향입니다.
TRAORI(1,0,1,0)          ; 공구의 기본 오리엔테이션은 Z 방향입니다.
TRAORI(1,0,1,1)          ; 공구의 기본 오리엔테이션은 Y/Z 방향입니다 (-45°에 해당).
```

4.9.2.3 오리엔테이션 프로그래밍 변수 및 초기 설정 (ORIRESET)

TRAORI를 통한 공구 오리엔테이션 프로그래밍

프로그램 가능한 TRAORI 오리엔테이션 변환과 연계하여 직선 축 X, Y, Z뿐만 아니라 회전 축 식별자 A..., B..., C...회전 축을 사용하여 축 위치를 프로그래밍하거나 각도나 벡터 구성 요소를 가진 가상 축을 프로그래밍할 수 있습니다. 오리엔테이션 및 기계 축에 대해 다양한 보간 유형이 가능합니다. 어떤 PO[angle] 오리엔테이션 폴리노미얼 및 PO[axis] 축 폴리노미얼이 현재 활성 상태인지 여부와 관계 없이 다양한 유형의 폴리노미얼을 프로그래밍할 수 있습니다. 이는 G1, G2, G3, CIP 또는 POLY를 포함합니다.

일부 경우 공구 오리엔테이션 변경은 오리엔테이션 벡터를 사용하여 프로그래밍할 수도 있습니다. 이러한 경우 각 블록의 최종 오리엔테이션은 벡터에 대한 직접 프로그래밍 또는 로터리 축 위치 프로그래밍을 통해 설정할 수 있습니다.

3축 - 5축 변환에 대한 오리엔테이션 프로그래밍 변수

오리엔테이션 프로그래밍의 다음 버전은 서로 배타적입니다.

A, B, C	회전 축 위치의 직접 입력
A2, B2, C2	Euler 각도 또는 RPY 각도를 통한 가상 축의 각도 프로그래밍
A3, B3, C3	벡터 컴포넌트 명칭
LEAD, TILT	경로 및 표면을 기준으로 한 선도각 및 경사각에 대한 명시 사항
A4, B4, C4 A5, B5, C5	블록 시작점 및 블록 끝지점에서의 표면 법선 벡터
A6, B6, C6 A7, B7, C7	테이퍼 표면 변환 상의 오리엔테이션 벡터 보간.
A8, B8, C8	공구의 방향 재설정, 후퇴 이동의 방향 및 경로 길이

공구 오리엔테이션의 접근 초기 설정(ORIRESET)

ORIRESET(...) 을 통해 관련 기계 동역학의 오리엔테이션 축은 선형으로 그리고 동기화된 방식으로 현재 위치에서 프로그램 된 초기 상태 위치로 이동됩니다. 축에 대해 기본 위치가 프로그램 되지 않을 경우 관련 머신 데이터 $MC_TRAFO5_ROT_AX_OFFSET_1/2로부터의 위치가 사용됩니다.

회전 축의 활성 프레임은 무시됩니다.

기계 운동 CA(채널 축 이름 C, A)의 예제

명령	설명
ORIRESET(90, 45)	축 C: 90° 축 A: 45°
ORIRESET(, 30)	축 C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 축 A: 30°
ORIRESET( )	축 C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 축 A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

기계 운동 CAC(채널 축 이름 C, A, B)의 예제

명령	설명
ORIRESET(90, 45, 90)	축 C: 90° 축 A: 45° 축 B: 90°
ORIRESET( )	축 C: $MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 축 A: $MC_TRAFO5_ROT_AX_OFFSET_1/2[1] 축 B: $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]

참고

ORIRESET...) 을 사용한 공구 오리엔테이션의 최초 상태로의 이동은 활성 오리엔테이션 변환 TRAORI...) 의 경우에만 발생합니다.

LEAD, TILT 및 THETA 회전 프로그램

선도각 LEAD 및 경사각 TILT.

3축 ~ 5축 변환과 관련하여 공구 오리엔테이션 회전이 LEAD 및 TILT 각도를 통해 프로그래밍됩니다.

회전 각도 THETA

세 번째 회전 축 을 이용한 변환의 경우, 공구 주변의 회전은 벡터 컴포넌트를 갖는 오리엔테이션뿐만 아니라 각도 LEAD, TILT 프로그래밍을 위한 THETA 회전 각도로 프로그램 될 수 있습니다.

4.9.2.4 공구 오리엔테이션 프로그래밍 (A..., B..., C..., LEAD, TILT)

다음 옵션은 공구 오리엔테이션 프로그래밍하는 경우 사용할 수 있습니다:

1. 로터리 축의 이동 직접 프로그래밍. 오리엔테이션 변경은 항상 베이직 또는 기계 좌표계에서 수행됩니다. 오리엔테이션 축은 동기화된 축으로 이송됩니다.
2. A2, B2, C2를 사용하여 각도 정의에 따라 Euler 또는 RPY 각도에서 프로그래밍
3. A3, B3, C3을 사용하여 방향 벡터 프로그래밍. 방향 벡터의 방향은 공구 팁에서 공구 어댑터를 향합니다.
4. A4, B4, C4를 통해 블록 시작 및 A5, B5, C5 (페이스 밀링) 를 통해 블록 끝에서 표면 법선 벡터 프로그래밍.
5. 리드 각도 LEAD 및 틸트 각도 TILT를 사용한 프로그래밍

6. A6, B6, C6을 사용해 법선 벡터로 테이퍼의 로터리 축 프로그래밍 또는 A7, B7, C7을 사용해 테이퍼의 원주면에 중간 오리엔테이션 프로그래밍.
   "테이퍼의 원주면을 따라 오리엔테이션 프로그래밍 (ORIPLANE, ORICONxx)" 을 참조하십시오.
7. A8, B8, C8을 사용하여 후퇴 이동 동안 오리엔테이션 재지정, 방향 및 경로 길이 프로그래밍.
   "오리엔테이션 특성 스무딩 (ORIPATHS A8=, B8=, C8=)" 을 참조하십시오.

참고

모든 경우 오리엔테이션 프로그래밍은 오리엔테이션 변환을 수행이 동작중인 경우에 한해서 허용됩니다.

이점: 어떤 형태의 기계라도 이 프로그램을 전송할 수 있습니다.

G 코드를 통한 공구 오리엔테이션 정의

참고

장비 제조업체

머신 데이터를 사용하여 Euler 또는 RPY 각도 간에 전환할 수 있습니다. 이에 따라 머신 데이터를 설정한 경우, 그룹 50의 활성 G 코드에 따라 또는 이 코드와 관계 없이 전환이 가능합니다. 다음과 같은 설정 옵션을 선택할 수 있습니다.

1. 오리엔테이션 축 정의 및 오리엔테이션 각도 정의에 대한 머신 데이터가 G 코드를 통해 0으로 설정된 경우:
   A2, B2, C2를 사용하여 프로그래밍된 각도는 **머신 데이터에 따라 다릅니다**. 오리엔테이션 프로그래밍의 각도 정의는 Euler 또는 RPY 각도로 해석됩니다.
2. 오리엔테이션 축 정의를 위한 머신 데이터가 G 코드를 통해 1로 설정되면 전환은 그룹 50의 활성 G 코드에
   종속 됩니다.
   A2, B2, C2를 사용하여 프로그래밍된 각도는 활성 G 코드 ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS 및 ORIPY2 에 따라 해석됩니다. 또한 오리엔테이션 축으로 프로그래밍된 값은 그룹 50의 활성 G 코드에 따라 오리엔테이션 각도로 해석됩니다.
3. 오리엔테이션 각도 정의를 위한 머신 데이터가 G 코드를 통해 1로 설정되고 오리엔테이션 축 정의를 머신 데이터가 G 코드를 통해 0으로 설정되면 전환은 그룹 50의 활성 G 코드에
   종속되지 않습니다.
   A2, B2, C2 를 사용하여 프로그래밍된 각도는 활성 G 코드 ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2 ORIAXPOS 및 ORIPY2 중 하나에 따라 해석됩니다. 오리엔테이션 축으로 프로그래밍된 값은 그룹 50의 활성 G 코드와 관계 없이 항상 라운드 축으로 해석됩니다.

구문

회전 축 위치

```
G1 X<값> Y<값> Z<값> A<값> B<값> C<값>
```

Euler 각도

```
G1 X<값> Y<값> Z<값> A2<값> B2<값> C2<값>
```

방향 벡터

```
G1 X<값> Y<값> Z<값> A3<값> B3<값> C3<값>
```

블록 시작점에서의 표면 법선 벡터

```
G1 X<값> Y<값> Z<값> A4<값> B4<값> C4<값>
```

블록 끝지점에서의 표면 법선 벡터

```
G1 X<값> Y<값> Z<값> A5<값> B5<값> C5<값>
```

리드 각도

```
LEAD=<값>
```

틸트 각도

```
TILT=<값>
```

의미

G1:	직선 보간
X, Y, Z:	선형 축 위치
A, B, C:	회전 축 위치
A2=, B2=, C2=:	각도 프로그래밍(Euler 각도 또는 RPY 각도)
A3=, B3=, C3=:	WCS의 X, Y 및 Z 좌표에서의 방향성 벡터.
A4=, B4=, C4=:	WCS의 X, Y 및 Z 좌표 상의 블록 시작점에서의 표면 법선 벡터
A5=, B5=, C5=:	WCS의 X, Y 및 Z 좌표 상의 블록 끝지점에서의 표면 법선 벡터
LEAD= :	선도각 [1)]
TILT= :	경사각 [1)]
1) 각도 표시의 해석은 MD21094 $MC_ORIPATH_MODE에서의 설정에 달려 있습니다.	

추가 정보

일반적으로 5축 프로그램은 CAD/CAM 시스템에 의해 생성되고 제어 시스템에서 입력되지 않습니다. 따라서 사후 프로세서의 프로그램에서 주로 다음 설명을 따릅니다.

다음 명령어가 오리엔테이션 프로그래밍에 사용될 수 있습니다.

명령	의미
ORIEULER:	회전 시퀀스 ZX'Z"를 갖는 Euler 각도
ORIRPY:	회전 시퀀스 XY'Z"를 갖는 RPY 각도
ORIRPY2:	회전 시퀀스 ZY'X"를 갖는 RPY 각도
ORIVIRT1:	다음을 통한, 자유롭게 정의할 수 있는 회전 시퀀스를 갖는 가상 오리엔테이션 축: MD21120 $MC_ORIAX_TURN_TAB_1
ORIVIRT2:	다음을 통한, 자유롭게 정의할 수 있는 회전 시퀀스를 갖는 가상 오리엔테이션 축: MD21130 $MC_ORIAX_TURN_TAB_2
ORIAXPOS:	로터리 축 위치를 갖는 가상 오리엔테이션 축

참고

장비 제조업체는 기계 데이터를 사용하여 다양한 변용을 정의할 수 있습니다. 장비 제조업체의 설명서를 참조하십시오.

Euler 각도 ORIEULER 에서의 프로그래밍, 회전 시퀀스 Z X' Z"

A2, B2, C2 를 사용한 ORIEULER 오리엔테이션 프로그래밍하는 동안 프로그램 된 값은 Euler 각도(도)로 해석됩니다.

새 오리엔테이션 벡터는 원래 오리엔테이션 벡터의 다음 세 개 회전으로부터 발생합니다.

1. 좌표 축 Z에 대한 회전 축 A2 를 이용
2. 새 좌표 축 X'에 대한 회전 축 B2 를 이용
3. 좌표 축 Z"에 대한 회전 축 C2 를 이용

이 경우 C2의 값(새 Z 축 중심 회선)은 의미 없으며 프로그래밍할 필요가 없습니다.

RPY 각도 ORIRPY 에서의 프로그래밍, 회전 시퀀스 X Y' Z"

A2, B2, C2 를 사용한 ORIEULER 오리엔테이션 프로그래밍하는 동안 프로그램 된 값은 회전 시퀀스 X Y' Z"를 갖는 RPY 각도(도)로 해석됩니다.

참고

ORIEULER, ORIRPY 를 이용한 프로그래밍과는 반대로, 여기 세 가지 값은 모두 오리엔테이션 벡터에 영향을 미칩니다.

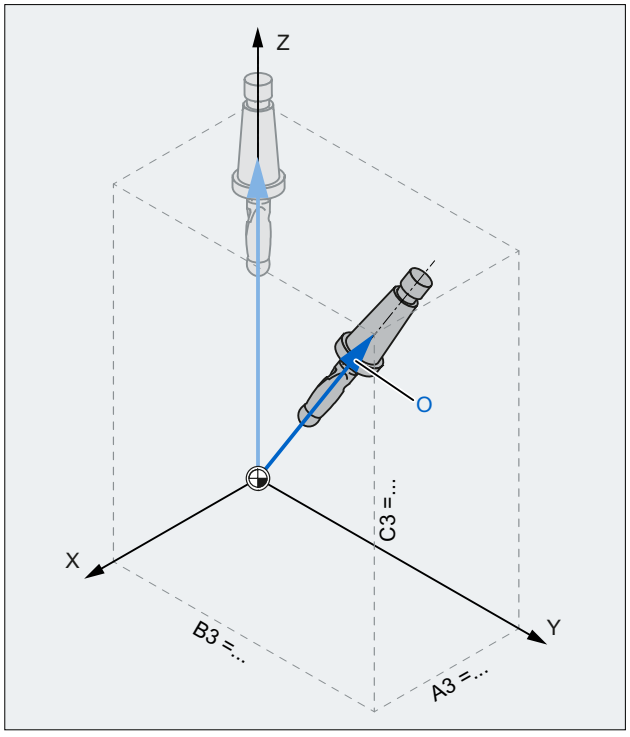
새 오리엔테이션 벡터는 원래 오리엔테이션 벡터의 다음 세 개 회전으로부터 발생합니다.

1. 좌표 축 X에 대한 회전 축 A2 를 이용
2. 새 좌표 축 Y'에 대한 회전 축 B2 를 이용
3. 좌표 축 Z"에 대한 회전 축 C2 를 이용

방향 벡터 프로그래밍

방향 벡터의 구성 요소는 A3, B3, C3를 사용하여 프로그램합니다. 벡터는 공구 어댑터 방향을 향하고 벡터 길이는 중요하지 않습니다.

프로그램되지 않은 벡터 구성 요소는 '0'으로 설정됩니다.



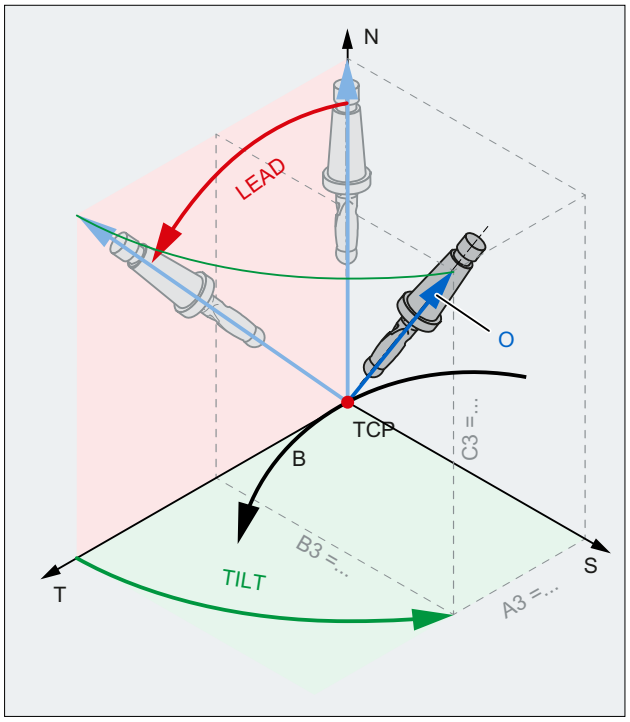
X, Y, Z	WCS의 좌표 축
A3, B3, C3	방향 벡터 컴포넌트
O	오리엔테이션 벡터

그림 4-4 방향 벡터 프로그래밍

LEAD 및 TILT를 통해 공구 방향 프로그램

다음으로부터 공구 방향이 결정됩니다:

- 경로 접선
- 곡면 법선 벡터
  블록 A4, B4, C4의 시작 및 블록A5, B5, C5의 끝
- 리드 각도LEAD
  경로 접선 및 표면 법선 벡터에 의해 정의된 평면에서의 각도
- 블록 끝에서의 틸트 각도TILT
  평면에서의 각도는 표면 법선 벡터를 기준으로 경로 접선에 대해 수직입니다.



T 경로 접선
S 경로 접선에 수직
N 곡면 법선
B 경로
TCP 공구 중심점
O 오리엔테이션 벡터

그림 4-5 LEAD TILT의 프로그래밍

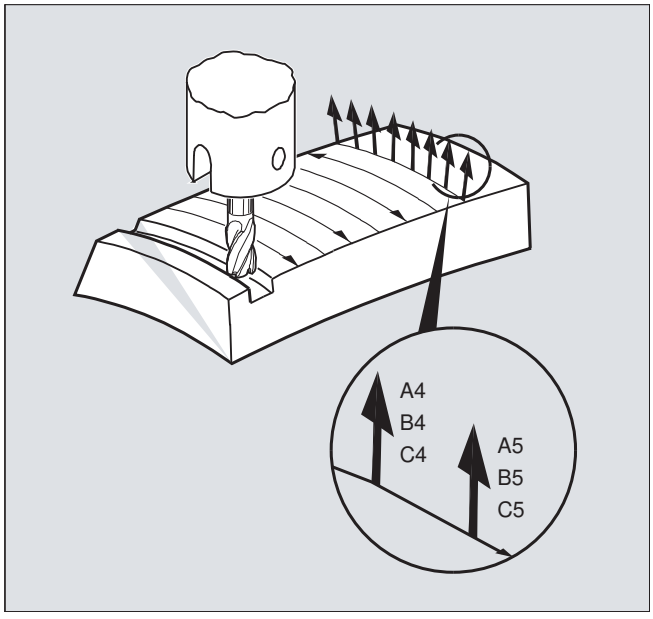
참고

3D 공구 옵셋이 있는 내부 코너에서의 동작

블록이 내부 모서리에서 짧아지는 경우, 프로그램 된 공구 방향은 블록 끝에서 여전히 잡힙니다.

4.9.2.5 페이스 밀링 (A4, B4, C4, A5, B5, C5)

페이스 밀링은 모든 종류의 곡면을 가공하는데 사용됩니다.



이런 유형의 3D 밀링을 위해서는 공작물 표면에서의 3D 경로를 라인 단위로 설명해야 합니다.

공구 형상 및 치수는 계산 시 고려되며 일반적으로 CAM에서 수행됩니다. 계산이 완료된 NC 블록은 포스트프로세서를 통해 제어 시스템으로 읽어옵니다.

경로 곡률 프로그램

표면 설명

경로 곡률은 다음 구성 요소를 통해 표면 법선 벡터에 의해 설명됩니다.

`A4, B4, C4` 블록 시작에서 벡터 시작

`A5, B5, C5` 블록 끝에서 벡터 종료

블록에 시작 벡터만 포함된 경우 표면 법선 벡터는 블록 전반에 걸쳐 일정합니다. 블록에 종료 벡터만 포함된 경우 대형 원호 보간을 통해 이전 블록의 종료 값에서 프로그래밍된 종료 값으로 보간을 진행합니다.

시작 벡터와 종료 벡터가 모두 프로그래밍된 경우 대형 원호 보간을 통해 두 방향 간의 보간을 실행합니다. 이를 통해 지속적으로 생성할 경로를 유연화할 수 있습니다.

초기 설정 시 활성 G17 - G19 레벨과 관계 없이 표면 법선 벡터는 Z 방향을 향합니다.

벡터 길이는 의미없습니다.

프로그램되지 않은 벡터 구성 요소는 0으로 설정됩니다.

ORIWKS가 활성화 ("오리엔테이션 축의 기준 (ORIWKS, ORIMKS): (페이지 759)" 참조) 되면 표면 법선 벡터는 활성 프레임을 기준으로 하고 프레임 회전을 통해 회전됩니다.

장비 제조업체

표면 법선 벡터는 기계 데이터를 통해 설정된 제한 값 이내에서 경로 접선와 수직을 이루어야 하며 그렇지 않을 경우 알람이 출력됩니다.

4.9.2.6 오리엔테이션 축의 기준 (ORIWKS, ORIMKS):

다음을 사용하여 공작물 좌표계에서 오리엔테이션 프로그래밍

- Euler 또는 RPY 각도 또는
- 오리엔테이션 벡터

회전 모션의 코스는 ORIMKS/ORIWKS를 사용하여 설정할 수 있습니다.

참고

장비 제조업체

오리엔테이션에 대한 보간 유형은 머신 데이터를 사용하여 지정합니다.
MD21104 $MC_ORI_IPO_WITH_G_CODE
= FALSE: G 코드 ORIWKS 및 ORIMKS가 기준을 제공합니다.
= TRUE: 기준은 51번째 그룹의 G 코드입니다(ORIAXES, ORIVECT, ORIPLANE, ...).

구문

```
ORIMKS=...
ORIWKS=...
```

의미

ORIMKS:	기계 좌표계에서 회전
ORIWKS:	공작물 좌표계에서 회전

참고

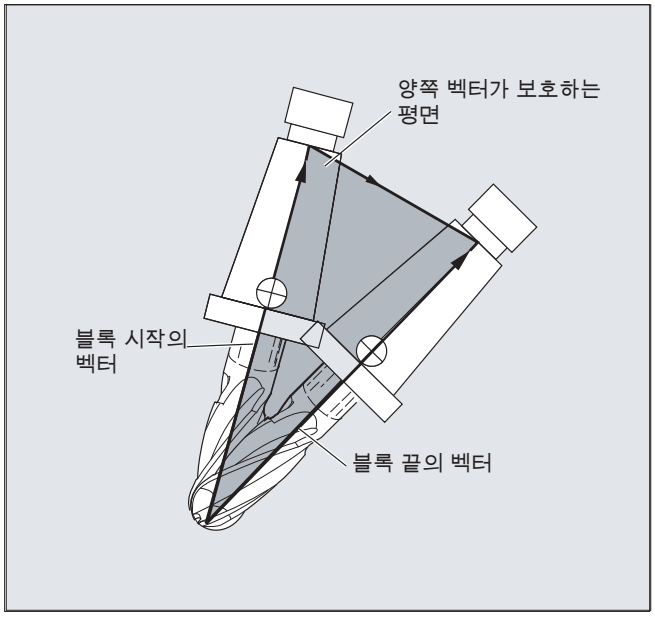
ORIWKS가 기본 설정입니다. 5축 프로그램의 경우 실행할 기계를 즉시 확인할 수 없을 때에는 항상 ORIWKS를 선택해야 합니다. 기계가 실제로 실행하는 이동은 기계 운동 구조에 따라 다릅니다.

ORIMKS는 기계의 실제 동작을 프로그래밍할 때 (장치 등과의 충돌을 방지하기 위해) 사용할 수 있습니다.

추가 정보

ORIMKS의 경우 공구에 의해 수행되는 이동은 기계 운동 구조에 따라 **다릅니다**. 공간 상의 고정 지점에서 공구 팁의 방향을 변경할 경우 선형 보간은 로터리 축 위치 사이에서 발생합니다.

ORIWKS의 경우 공구에 의해 수행되는 이동은 기계 운동 구조에 따라 **종속되지 않습니다**. 고정 툴 팁의 방향 변경을 통해 공구는 시작 및 종료 벡터에 의해 설정된 평면에서 이동합니다.



단일 위치

참고

ORIWKS

5축 기계의 단일 설정 영역에서 방향을 이동하려면 기계 축을 상당히 이동해야 합니다. (예를 들어 로터리 축으로 C 및 회전 축으로 A를 가진 로터리 스위블 헤드에서 A = 0의 모든 위치는 일치합니다.)

장비 제조업체

기계 축의 과부하를 방지하기 위해 속도 제어 시스템이 단일 위치 근처의 공구 경로 속도를 상당히 감소합니다.

머신 데이터

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

극점에 가까운 방향 이동은 극점을 향하고 급속한 가공이 가능하도록 변환을 파라미터화할 수 있습니다.

단일 위치는 MD $MC_TRAFO5_POLE_LIMIT에 의해서만 처리됩니다.

추가 정보: 기능 매뉴얼 변환

4.9.2.7 오리엔테이션 축 프로그래밍 (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

"오리엔테이션 축" 기능은 공간 상의 공구 오리엔테이션을 설명하고 로터리 축의 옵셋을 프로그래밍하여 수행됩니다. 또한 공구 자체를 회전하여 3도의 자유도를 확보할 수 있습니다. 이 경우 공구 방향은 6축 변환이 필요한 세 번째 로터리 축을 통해 공간에서 지정됩니다. 공구 자체 회전은 회전 벡터의 보간 유형에 따라 THETA 회전 각도를 사용하여 정의됩니다 ("공구 오리엔테이션 회전 (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (페이지 771)" 참조).

축 식별자 A2, B2 및 C2를 사용하여 오리엔테이션 축을 프로그램합니다.

구문

```
N... ORIAXES/ORIVECT                ; 직선 또는 대형 원호 보간
N... G1 X Y Z A B C

N... ORIPLANE                       ; 평면의 오리엔테이션 보간
```

```
N... ORIEULER/ORIRPY/ORIRPY2     : 오리엔테이션 각도 Euler/RPY 각도
N... G1 X Y Z A2= B2= C2=        ; 가상 축의 각도 프로그래밍

N... ORIVIRT1/ORIVIRT2           ; 가상 오리엔테이션 축 정의 1/2
N... G1 X Y Z A3= B3= C3=        ; 방향 벡터 프로그래밍
```

참고

오리엔테이션 축의 다른 로터리 축 옵셋을 프로그래밍하여 테이퍼 원주면에 따라 오리엔테이션을 변경할 수 있습니다 ("테이퍼의 주변 표면을 따라 오리엔테이션 프로그래밍 (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (페이지 764)" 참조).

의미

ORIAXES:	기계 또는 오리엔테이션 축의 직선 보간
ORIVECT:	대형 원호 보간 (ORIPLANE과 동일)
ORIMKS: ORIWKS:	기계 좌표계에서 회전 공작물 좌표계에서 회전 자세한 내용은 "오리엔테이션 축의 기준 (ORIWKS, ORIMKS): (페이지 759)"을 참조하십시오.
A= B= C=:	기계 축 위치 프로그램
ORIEULER:	Euler 각도를 이용한 오리엔테이션 프로그래밍
ORIRPY:	RPY 각도를 이용한 오리엔테이션 프로그래밍 방향 순서는 XYZ이고, • A2는 X를 기준으로 한 회전 각도입니다. • B2는 Y를 기준으로 한 회전 각도입니다. • C2는 Z를 기준으로 한 회전 각도입니다.
ORIRPY2:	RPY 각도를 이용한 오리엔테이션 프로그래밍 방향 순서는 ZYX이고, • A2는 Z를 기준으로 한 회전 각도입니다. • B2는 Y를 기준으로 한 회전 각도입니다. • C2는 X를 기준으로 한 회전 각도입니다.
A2= B2= C2=:	가상 축의 각도 프로그램

ORIVIRT1/ORIVIRT2:	가상 오리엔테이션 축을 사용한 오리엔테이션 프로그래밍 정의 1: MD21120 $MC_ORIAX_TURN_TAB_1에 따른 정의 정의 2: MD21130 $MC_ORIAX_TURN_TAB_2에 따른 정의
A3= B3= C3=:	방향 축의 방향 벡터 프로그램

추가 정보

장비 제조업체

MD21102 $MC_ORI_DEF_WITH_G_CODE는 프로그래밍된 각도 A2, B2 및 C2를 정의하는 방법을 지정합니다.

정의는 MD21100 $MC_ORIENTATION_IS_EULER (표준) 또는 G 그룹 50 (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2) 을 따릅니다.

MD21104 $MC_ORI_IPO_WITH_G_CODE는 활성화할 보간 모드 유형을 정의합니다. ORIWKS/ORIMKS 또는 ORIAXES/ORIVECT 중에서 활성화할 보간 모드 유형을 정의하는 데 사용됩니다.

JOG 모드

이 모드의 작업에서 방향 각도 보간은 항상 선형적입니다. 이송 키를 통해 연속 및 증분 이송 동안 하나의 오리엔테이션 축만 이송할 수 있습니다. 오리엔테이션 축은 핸드휠을 사용하여 동시에 이송할 수 있습니다.

오리엔테이션 축이 수동으로 이송될 때는 채널별 이송속도 오버라이드 스위치 또는 급이송 오버라이드 상의 급이송 오버라이드 스위치가 적용됩니다.

개별 속도 설정은 다음 머신 데이터를 통해 가능합니다:

MD21160 $MC_JOG_VELO_RAPID_GEO

MD21165 $MC_JOG_VELO_GEO

MD21150 $MC_JOG_VELO_RAPID_ORI

MD21155 $MC_JOG_VELO_ORI

참고

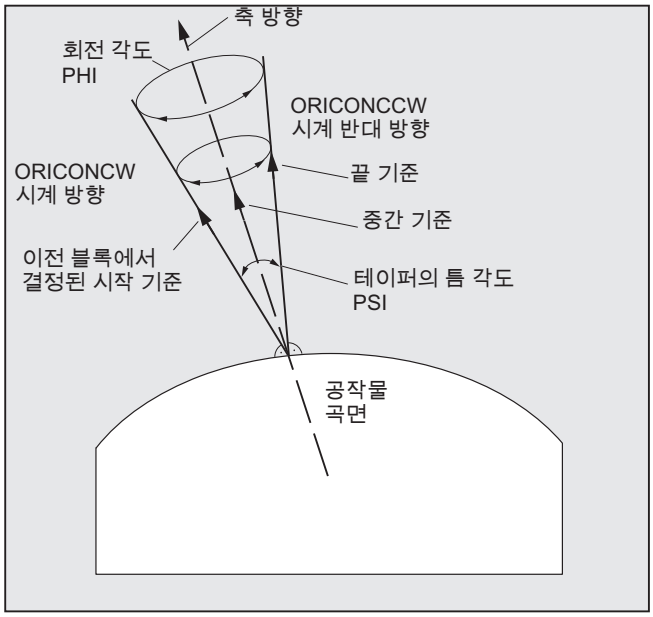
"변환 패키지 처리" 기능이 있는 SINUMERIK 840D sl

"Cartesian 수동 이송" 기능을 사용하면 JOG 모드에서 형상 축의 변환은 참조 시스템 MCS, WCS 및 TCS에서 서로 간에 개별적으로 설정할 수 있습니다.

추가 정보: 기능 매뉴얼 변환

4.9.2.8 테이퍼의 주변 표면을 따라 오리엔테이션 프로그래밍 (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

확장된 오리엔테이션을 통해 공간 상의 테이퍼의 주변 표면을 따라 오리엔테이션을 변경할 수 있습니다. 오리엔테이션 벡터는 ORICONxx 모달 명령을 사용하여 테이퍼의 주변 표면에서 보간됩니다. 공간에서 보간하는 경우 종료 오리엔테이션은 ORIPLANE을 통해 프로그래밍할 수 있습니다. 시작 오리엔테이션은 일반적으로 이전 블록에 의해 정의됩니다.



프로그래밍

종료 오리엔테이션은 A2, B2, C2를 사용하여 Euler 또는 RPY 각도 방식으로 각도 프로그램을 지정하거나 A, B, C를 사용하여 로터리 축 위치를 프로그래밍하여 정의됩니다. 테이퍼의 주변 표면을 따라 오리엔테이션 축에 대해 자세한 프로그램 세부 정보가 필요합니다.

- A6, B6, C6 벡터를 통한 테이퍼의 로터리 축
- 식별자 NUT를 가진 개방 각도 PSI
- A7, B7, C7를 통해 테이퍼를 벗어난 중간 오리엔테이션

참고

테이퍼의 로터리 축에 대한 방향 벡터 A6, B6, C6 프로그램

종료 오리엔테이션 프로그래밍은 경우에 따라 필요 없습니다. 종료 오리엔테이션을 지정하지 않으면 360도의 전체 외부 테이퍼가 보간됩니다.

NUT=angle로 테이퍼의 개방 각도 프로그램

종료 오리엔테이션을 지정해야 합니다.

360도의 전체 외부 테이퍼는 이 방법으로 보간할 수 없습니다.

테이퍼 외부에서 중간 오리엔테이션 A7, B7, C7 프로그램

종료 오리엔테이션을 지정해야 합니다. 오리엔테이션 및 회전 방향 변경은 세 가지 벡터인 시작 오리엔테이션, 종료 오리엔테이션 및 중간 오리엔테이션에 의해 고유하게 정의됩니다. 세 가지 모든 벡터가 서로 달라야 합니다. 프로그래밍된 중간 오리엔테이션이 시작 또는 종료 오리엔테이션과 평행할 경우 오리엔테이션을 위한 직선 대형 원호 보간을 시작 및 종료 벡터에 의해 정의된 평면에서 실행합니다.

테이퍼의 주변 표면에 대한 확장 오리엔테이션 보간

`N... ORICONCW 또는 ORICONCCW`	테이퍼 외부에 대한 **보간**
`N... A6= B6= C6= A3= B3= C3=`	테이퍼 CW 방향/CCW 방향의 방향 벡터
또는	및 종료 오리엔테이션 또는
`N... ORICONTO`	접선 트렌지션
`N... G1 X Y Z A6= B6= C6=`	및 종료 오리엔테이션 지정
또는	또는 다음을 사용하여
`N... ORICONIO`	종료 오리엔테이션 지정 및
`N... G1 X Y Z A7= B7= C7=`	테이퍼 외부에 중간 오리엔테이션 지정
`N... PO[PHI]=(a2, a3, a4, a5)`	회전 각도 폴리노미얼 및
`N... PO[PSI]=(b2, b3, b4, b5)`	틈 각도 폴리노미얼

파라미터

ORIPLANE:	평면에서 보간 (대형 원호 보간)
ORICONCW:	테이퍼의 주변 표면에서 시계 방향으로 보간됩니다.
ORICONCCW:	테이퍼의 원주면에서 CCW 방향으로 보간
ORICONTO:	테이퍼의 주변 표면에서 접선 트렌지션으로 보간
A6= B6= C6=:	테이퍼의 로터리 축 프로그램 (법선 벡터)
NUT=각도:	테이퍼의 개방 각도 (도)
NUT=+179:	180°보다 작거나 같은 각도로 이송
NUT=-181:	180도보다 크거나 같은 각도로 이송
ORICONIO:	테이퍼의 주변 표면에 대한 보간
A7= B7= C7=:	중간 오리엔테이션 (법선 벡터로 프로그래밍)
PHI:	테이퍼의 방향 축에 대한 회전 오리엔테이션 각도
PSI:	테이퍼의 개방 각도
가능한 폴리노미얼 PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	각도와 관계 없이 최대 5차 폴리노미얼까지 프로그래밍 할 수 있습니다.

예: 오리엔테이션 변경

프로그램 코드	코멘트
…	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	; 오리엔테이션 변환 ON
N30 ORIVECT	; 벡터로 공구 오리엔테이션 보간
…	; 평면에서 공구 오리엔테이션
N40 ORIPLANE	; 대형 원호 보간을 선택합니다.
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; Y/Z 평면에서 오리엔테이션은 45°로 회전합니다. 블록 끝에서 오리엔테이션 (0,1/√2, 1/√2)에 도달합니다.
…	
N70 ORICONCW	; 테이퍼 외부에 오리엔테이션 프로그래밍:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	오리엔테이션 벡터는 방향 (0,0,1) 의 테이퍼 외부에서 CW 방향으로 오리엔테이션 (1/√2,0,1/√2)까지 보간되고, 회전 각도는 270도입니다.
N90 A6=0 B6=0 C6=1	; 공구 오리엔테이션은 동일한 테이퍼의 외부에서 전체 회전을 통해 적용됩니다.

추가 정보

공간 상의 어디서나 테이퍼의 주변 표면을 따라 방향 변경에 대해 설명해야 하는 경우 회전해야 할 공구 방향에 대한 벡터를 알고 있어야 합니다. 또한 시작 및 종료 방향을 지정해야 합니다. 시작 방향은 이전 블록으로부터 제공되며 종료 방향은 다른 조건을 통해 프로그램 또는 정의해야 합니다.

ORIPLANE 평면에서의 프로그래밍은 ORIVECT에 해당

각도 다항식과 함께 대형 반경 원호 보간 프로그래밍은 형상에 대한 직선 및 폴리노미널 보간에 해당합니다. 공구 방향은 시작 및 종료 방향에 의해 정의된 평면에서 보간됩니다. 추가 다항식을 프로그래밍하는 경우 방향 벡터를 평면에서 벗어나 기울일 수 있습니다.

평면 G2/G3, CIP 및 CT에 원호 프로그래밍

확장 방향은 평면에서 원호 보간에 해당합니다. G2/G3, 중간 지점 CIP를 이용하는 원, 접선 원 CT 등의 중간 지점 데이터 또는 반경 데이터를 이용하는 원의 프로그램 옵션에 대해 자세히 알아보려면 '원호 보간 (페이지 209)'장을 참고하십시오.

오리엔테이션 프로그래밍

테이퍼의 주변 표면에서 방향 벡터 보간 ORICONxx

테이퍼의 주변 표면에 대한 오리엔테이션 보간을 위해 G 코드 그룹 51에서 네 가지 서로 다른 유형의 보간을 선택할 수 있습니다.

1. 종료 방향, 테이퍼 방향 또는 개방 각도 지정을 통해 시계 방향 ORICONCW으로 테이퍼 외부에서 보간. 방향 벡터는 식별자 A6, B6, C6 및 간격 0도 - 180도 사이에서 식별자 NUT= value 범위를 가진 테이퍼의 개방 각도를 통해 프로그래밍됩니다.
2. 종료 오리엔테이션 및 테이퍼 방향 또는 틈 각도의 지정으로 시계 반대 방향 ORICONCW의 테이퍼 외부에서 보간 방향 벡터는 식별자 A6, B6, C6 및 간격 0도 - 180도 사이에서 식별자 NUT= value 범위를 가진 테이퍼의 개방 각도를 통해 프로그래밍됩니다.
3. 종료 방향 및 중간 방향 지정을 통해 테이퍼 외부에서 보간 ORICONIO, 식별자 A7, B7, C7를 통해 프로그래밍됩니다.
4. 접선 전이 및 종료 방향 지정을 통해 테이퍼 외부에서 보간 ORICONTO. 방향 벡터는 식별자 A6, B6, C6을 통해 프로그래밍됩니다.

4.9.2.9 두 개의 접점 방향 지정 (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

공간 상의 두 번째 곡선을 사용하여 오리엔테이션 변경 프로그램

공간 상의 곡선을 따라 공구 팁을 사용하는 것 이외에 오리엔테이션 변경을 프로그래밍하는 다른 방법은 ORICURVE를 사용하여 공구의 두 번째 접점 이동을 프로그래밍하는 것입니다. 이 방법에서, 공구 오리엔테이션 변경은 공구 벡터를 프로그램할 때처럼 고유하게 정의할 수 있습니다.

장비 제조업체

공구의 두 번째 오리엔테이션 경로를 프로그래밍하는 경우, 머신 데이터를 통해 설정할 수 있는 축 식별자에 대한 장비 제조업체의 참고 사항을 참조하십시오.

프로그래밍

이 유형의 보간은 공간 상의 두 개 곡선에 대한 점 (G1 사용) 또는 폴리노미얼 (POLY 사용) 을 프로그래밍하는 데 사용할 수 있습니다. 원형 및 나선은 허용되지 않습니다. BSPLINE 스플라인 보간 및 '짧은 스플라인 블록 조합' 기능도 실행할 수 있습니다.

'짧은 스파인 블록 결합' 기능에 대해 **자세히 알아보려면** 기능 매뉴얼 기본 기능을 참고하십시오.

다른 스플라인 유형(ASPLINE 및 CSPLINE)과 COMP...를 사용한 컴프레서 활성화는 허용되지 않습니다.

공구의 두 개 접점 이동은 좌표에 대한 오리엔테이션 폴리노미얼로 프로그램할 때 5차 폴리노미얼로 미리 정의할 수 있습니다.

공간 상의 추가 곡선 및 좌표 폴리노미얼을 통한 확장 오리엔테이션 보간

```
N... ORICURVE                          ;공구의 두 번째 접점 이동 및 관련 좌표의 추가 폴리노미얼 지정
N... PO[XH]=(xe, x2, x3, x4, x5)
N... PO[YH]=(ye, y2, y3, y4, y5)
N... PO[ZH]=(ze, z2, z3, z4, z5)
```

파라미터

`ORICURVE`	공구의 두 접점 사이 이동을 지정하는 오리엔테이션 보간.
`XH YH ZH`	공간 상의 곡선으로 추가 형상의 공구에 대한 두 번째 접점의 좌표 식별자

`PO[XH]=(xe, x2, x3, x4, x5)`
`PO[YH]=(ye, y2, y3, y4, y5)`
`PO[ZH]=(ze, z2, z3, z4, z5)`
	가능한 폴리노미얼
	해당 종점을 사용하는 것 이외에 공간 상의 곡선은 폴리노미얼을 사용하여 프로그래밍할 수 있습니다.

`xe, ye, ze`	공간 상의 곡선에 대한 종점
`xi, yi, zi`	최대 5차 폴리노미얼의 계수

참고

두 번째 오리엔테이션 경로를 프로그램하기 위한 식별자 XH YH ZH

다른 식별자 또는 직선 축과 충돌하지 않도록 식별자를 선택해야 합니다.

X Y Z 축

및 다음과 같은 로터리 축

A2 B2 C2 Euler 각도 또는 RPY 각도

A3 B3 C3 방향 벡터

A4 B4 C4 또는 A5 B5 C5 표면 법선 벡터

A6 B6 C6 회전 벡터 또는 A7 B7 C7 중간 지점 좌표

또는 기타 보간 파라미터.

4.9.3 방향 폴리노미얼 (PO[angle], PO[coordinate])

현재 활성화된 G 코드 그룹 1의 폴리노미얼 보간과 관계 없이, 두 가지 서로 다른 유형의 오리엔테이션 폴리노미얼을 최대 5도까지 3축에서 5축까지의 변환에 대해 프로그래밍할 수 있습니다.

1. **각도**에 대한 폴리노미얼: 시작 및 종료 방향에 의해 정의된 평면을 기준으로 리드 각도 LEAD, 틸트 각도 TILT.
2. **좌표**에 대한 폴리노미얼: 공구의 기준점에서 공구 방향에 대해 공간 상의 두 번째 곡선의 XH, YH, ZH.

6축 변환을 통해 회전 벡터 THT의 회전은 공구 방향에 추가하여 공구 회전에 대해 최대 5차 폴리노미얼로 프로그래밍할 수 있습니다.

구문

유형 1 **각도**에 대한 방향 폴리노미얼

```
N... PO[PHI]=(a2, a3, a4, a5)
N... PO[PSI]=(b2, b3, b4, b5)
```

3축 ~ 5축 변환

유형 2 **좌표**에 대한 방향 폴리노미얼

`N… PO[XH]=(xe, x2, x3, x4, x5)`
`N… PO[YH]=(ye, y2, y3, y4, y5)`
`N… PO[ZH]=(ze, z2, z3, z4, z5)`
	공구 방향에 대해 두 번째 방향 경로의 좌표에 대한 식별자

두 가지 경우 모두, 6축 변환을 통해 다음을 사용하여 **회전**에 대한 폴리노미얼을 프로그래밍할 수 있습니다.

`N… PO[THT]=(c2, c3, c4, c5)`	경로를 기준으로 회전 보간
또는	
`N… PO[THT]=(d2, d3, d4, d5)`	방향 변경에 대한 절대, 상대 및 접선 보간

방향 벡터. 이것은 변환이 THETA 회전 각도를 사용하여 프로그램 및 보간할 수 있는 옵셋을 가진 회전 벡터를 지원하는 경우 가능합니다.

의미

`PO[PHI]`	시작 및 종료 방향 사이 평면에서의 각도
`PO[PSI]`	시작 및 종료 방향 사이 평면에서 방향의 기울기를 설명하는 각도
`PO[THT]`	THETA를 사용하여 프로그래밍된 그룹 54의 G 코드 중 하나에 대한 방향 벡터를 회전하여 생성된 회전 각도
`PHI`	리드 각도 LEAD
`PSI`	틸트 각도 TILT
`THETA`	Z에서 공구 방향에 대한 회전
`PO[XH]`	공구의 기준점에 대한 X 좌표
`PO[YH]`	공구의 기준점에 대한 Y 좌표
`PO[ZH]`	공구의 기준점에 대한 Z 좌표

추가 정보

다음과 같은 경우 방향 다항식을 프로그래밍할 수 없습니다:

- ASPLINE, BSPLINE, CSPLINE 스플라인 보간이 활성화된 경우.
  유형 1 오리엔테이션 각 폴리노미얼은 스플라인 보간을 제외하고 모든 유형의 보간에 허용됩니다. 즉, 급이송 G00 또는 이송 속도 G01을 통한 직선 보간,
  POLY를 사용한 폴리노미얼 보간 및
  G02, G03, CIP, CT, INVCW 및 INCCCW를 이용한 원호/나선 보간
  이 모두 허용됩니다.
  그러나 유형 2 오리엔테이션 좌표 폴리노미얼은 급이송 G00 또는
  이송 속도 G01을 통한 직선 보간이나
  POLY를 통한 폴리노미얼 보간이 활성화된 경우에만 가능합니다.
- 방향이 ORIAXES 축 보간을 사용하여 보간되는 경우. 이 경우 다항식은 방향 축 A 및 B에 대해 PO[A] and PO[B]를 통해 직접 프로그래밍할 수 있습니다.

ORIVECT, ORIPLANE 및 ORICONxx의 경우 유형 1 오리엔테이션 폴리노미얼

ORIVECT, ORIPLANE 및 ORICONxx를 이용한 대형 반경 원호 보간 및 테이퍼의 외부 보간에는 유형 1 오리엔테이션 폴리노미얼만 허용됩니다.

ORICURVE의 경우 유형 2 오리엔테이션 폴리노미얼

공간 간의 추가 곡선을 통한 보간 ORICURVE가 활성화되면 방향 벡터의 Cartesian 구성 요소가 보간되고 유형 2 방향 다항식만 가능합니다.

4.9.4 공구 오리엔테이션 회전 (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)

또한 이동 가능한 공구를 가진 기계 유형에서 공구 방향을 변경하려면 각 블록을 종료 방향으로 프로그래밍합니다. 기계 운동 구조에 따라 오리엔테이션 축의 방향 또는 방향 벡터 THETA 회전 방향을 프로그래밍할 수 있습니다. 이러한 회전 벡터에 대해 서로 다른 보간 유형을 프로그래밍할 수 있습니다:

- ORIROTA: 절대 회전 방향을 기준으로 한 회전 각도.
- ORIROTR: 시작 오리엔테이션과 종료 오리엔테이션 사이의 평면을 기준으로 한 회전 각도.
- ORIROTT: 방향 벡터의 변경을 기준으로 회전 각도.
- ORIROTC: 경로 접선에 대한 접선 회전 각도.

구문

보간 유형 ORIROTA가 활성화된 경우에만 회전 또는 회전 벡터의 각도를 다음 네 가지 모드에서 모두 프로그래밍할 수 있습니다.

1. 로터리 축 위치 `A, B, C`로 직접
2. `A2, B2, C2`를 통한 Euler 각도 (도)
3. `A2, B2, C2`를 통한 RPY 각도 (도)
4. `A3, B3, C3`를 통한 방향 벡터 (`THETA=<value>`로 회전 각도 지정)

ORIOTR 또는 ORIOTT가 활성인 경우에만 회전 각도를 `THETA`로 직접 프로그래밍할 수 있습니다.

또한 방향 변경 없이 개별 블록에서도 회전을 프로그래밍할 수 있습니다. 이 경우 ORIROTR 및 ORIROTT는 관계가 없고, 회전 각도는 항상 절대 방향 (ORIROTA) 을 기준으로 해석됩니다.

`N... ORIROTA`	회전 벡터 보간 정의
`N... ORIROTR`	
`N... ORIROTT`	
`N... ORIROTC`	
`N... A3= B3= C3= THETA=<value>`	방향 벡터 회전 정의
`N... PO[THT]=(`d_2`, `d_3`, `d_4`, `d_5`)`	5차 폴리노미얼을 통한 회전 각도 보간

의미

`ORIROTA:`	절대 회전 방향을 기준으로 한 회전 각도
`ORIROTR:`	시작 오리엔테이션과 종료 오리엔테이션 사이의 평면을 기준으로 한 회전 각도
`ORIROTT:`	방향 변경에 대한 접선 회전 벡터를 통한 회전 각도
`ORIROTC:`	경로 접선에 대한 접선 회전 벡터를 통한 회전 각도
`THETA:`	방향 벡터의 회전
`THETA=<value>:`	블록 끝에 도달한 회전 각도 (도)
`THETA=`Θe`:`	회전 벡터의 끝각 Θ_e를 통한 회전 각도
`THETA=AC(...):`	절대 치수로 넌모달 전환
`THETA=AC(...):`	증분 치수로 넌모달 전환

Θe:	G90의 절대값 및 G91(증분 치수)의 상대값 모두에 대한 회전 벡터의 끝각이 활성화됩니다.
PO[THT]=(....):	회전 각도에 대한 폴리노미얼

예: 오리엔테이션의 회전

프로그램 코드	코멘트
N10 TRAORI	; 오리엔테이션 변환 활성화.
N20 G1 X0 Y0 Z0 F5000	; 공구 오리엔테이션
N30 A3=0 B3=0 C3=1 THETA=0	; Z 방향에서 회전 각도 0
N40 A3=1 B3=0 C3=0 THETA=90	; X 방향에서 회전 각도 약 90°
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	;오리엔테이션
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; Y 방향에서 회전 각도 약 180°
N70 ORIROTT	; 90°로 회전을 일정하게 유지
N80 A3=1 B3=0 C3=0 THETA=30	;오리엔테이션 변경에 따른 회전 각도
	;X/Y 평면에 대해 30° 각도로 회전 벡터

블록 N40을 보간하는 경우 회전 각도가 초기 값 0°에서 최종 값 90°로 직선 보간됩니다. 블록 N50의 경우 회전 각도는 포물선 θ(u) = +90u2에 따라 90°에서 180°로 변경됩니다. N60에서 방향 변경 없이 회전을 실행할 수 있습니다.

N80에서 공구 방향은 Y 방향에서 X 방향으로 회전합니다. 방향 변경은 X/Y 평면에서 발생하고 회전 벡터는 이 평면에 대해 30도 각도를 설명합니다.

추가 정보

ORIROTA

회전 각도 THETA는 공간 상의 절대 방향을 기준으로 보간됩니다. 기본 회전 방향은 기계 데이터에서 정의됩니다.

ORIROTR

회전 각도 THETA는 시작 및 종료 방향에 의해 정의된 평면을 기준으로 보간됩니다.

ORIROTT

회전 각도 THETA는 방향 변경을 기준으로 보간됩니다. THETA=0의 경우 회전 벡터가 방향 변경에 대해 접선 방향으로 보간되며 최소 하나의 다항식이 방향에 대해 "틸트 각도 PSI"를 프로그래밍한 경우에만 ORIROTR과 다릅니다. 결과적으로 평면에서 실행되지 않는 방향으로 변경됩니다. 추가 회전 각도 THETA는 항상 방향 변경을 기준으로 특정 값을 산출하도록 회전 벡터를 보간하는 데 사용할 수 있습니다.

ORIROTC

회전 벡터는 THETA 각도를 사용하여 프로그래밍할 수 있는 옵센을 통해 경로 접선을 기준으로 보간됩니다. 또한 옵셋 각도에 대해 최대 5차 다항식 PO[THT]=(c2, c3, c4, c5)을 프로그래밍할 수 있습니다.

4.9.5 경로 기준 방향

4.9.5.1 경로 기준 방향 유형

이 확장 기능을 사용하여 블록 끝 뿐만 아니라 전체 경로 전반에 걸쳐 관련 방향을 설정할 수 있습니다. 이전 블록에서 확보한 오리엔테이션을 대형 원호 보간을 사용하여 프로그래밍된 종료 오리엔테이션으로 전송합니다. 기본적으로 경로를 기준으로 원하는 방향을 프로그래밍하는 두 가지 방법은 다음과 같습니다.

1. 공구 회전처럼, 공구 방향은 ORIPATH, ORPATHTS를 사용하여 경로를 기준으로 보간됩니다.
2. 방향 벡터는 일반적인 방법으로 프로그래밍 및 보간됩니다. 방향 벡터의 회전은 ORIROTC를 사용하여 경로 접선을 기준으로 실행됩니다.

구문

공구의 방향 및 회전 보간 유형은 다음을 사용하여 프로그래밍됩니다:

N... ORIPATH	경로 기준 방향
N... ORIPATHS	방향 유연화 특성을 통한 경로 기준 방향
N... ORIROTC	경로를 기준으로 회전 벡터 보간

경로 궤도의 코너에서 발생하는 오리엔테이션 블립은 ORIPATHS를 사용하여 스무딩할 수 있습니다. 리프트 모션의 방향 및 경로 길이는 구성 요소 A8=X, B8=Y C8=Z를 사용하여 벡터를 통해 프로그래밍됩니다.

ORIPATH/ORIPATHS는 세 개의 각도를 통해 경로 접선에 대한 다양한 기준을 프로그래밍하는 데 사용할 수 있습니다.

- LEAD= 경로 및 표면에 상대적인 리드 각도의 지정
- TILT= 경로 및 표면에 상대적인 기울기 각도의 지정
- THETA= 회전 각도

전체 경로의 경우. PO[THT]=(...)를 사용하는 THETA 회전 각도 이외에 최대 5도의 폴리노미얼을 프로그래밍할 수 있습니다.

참고

장비 제조업체

장비 제조업체의 설명서를 참조하십시오. 기타 설정은 구성 가능한 기계 및 설정 데이터를 통해 경로 기준 방향에 대해 설정할 수 있습니다.

추가 정보: 기능 매뉴얼 변환

의미

다양한 설정은 머신 데이터를 통해 보간 각도 LEAD 및 TILT에 대해 수행할 수 있습니다:

- LEAD 및 TILT를 사용하여 프로그래밍된 공구 방향 기준은 전체 블록에 대해 유지됩니다.
- 리드 각도 LEAD: 접선 및 법선 벡터 TILT 기준 수직 방향 회전: 법선 벡터 기준 방향 회전.
- 리드 각도 LEAD: 접선 및 법선 벡터 TILT 각도 기준 수직 방향 회전: 경로 접선 방향으로 방향 회전.
- 회전 각도 THETA=: 6축 변환에서 오리엔테이션 축 역할을 수행하는 추가적인 세 번째 로터리 축의 공구 회전.

참고

OSC, OSS, OSSE, OSD 및 OST와 연동하여 허용되지 않는 경로 기준 방향

경로 기준 오리엔테이션 보간, 즉 ORIPATH 또는 ORIPATHS 및 ORIOTC는 그룹 34의 G 코드를 통해 방향 특성 유연화와 연동하여 프로그래밍할 수 없습니다. 이를 위해서는 OSOF를 활성화해야 합니다.

4.9.5.2 경로 기준 공구 오리엔테이션 회전 (ORIPATH, ORIPATHS, 회전 각도)

6축 변환의 경우 공구는 공간 상에서 원하는 대로 공구 방향을 지정하도록 세 번째 축을 통해 공구를 회전할 수 있습니다. ORIPATH 또는 ORIPATHS를 사용하여 경로 기준 공구 오리엔테이션 회전의 경우 THETA 회전 각도를 통해 추가 회전을 프로그래밍할 수 있습니다. 또는 공구 방향에 수직인 평면에 위치한 벡터를 사용하여 LEAD 및 TILT 각도를 프로그래밍할 수 있습니다.

장비 제조업체

장비 제조업체의 설명서를 참조하십시오. LEAD 및 TILT 각도 보간은 머신 데이터를 사용하여 서로 다르게 설정할 수 있습니다.

구문

공구 오리엔테이션 및 공구 회전

경로 기준 공구 오리엔테이션 유형은 ORIPATH 또는 ORIPATHS를 사용하여 활성화됩니다.

`N... ORIPATH`	경로 기준 오리엔테이션 유형 활성화
`N... ORIPATHS`	오리엔테이션 유연화 특성을 통해 경로 기준 오리엔테이션 유형 활성화

회전 가능한 세 가지 각도 활성화:

`N... LEAD=`	표면 법선 벡터 기준 프로그래밍된 오리엔테이션의 각도
`N... TILT=`	평면에서의 프로그래밍된 오리엔테이션의 각도는 표면 법선 벡터를 기준으로 경로 접선에 대해 수직입니다.
`N... THETA=`	세 번째 로터리 축의 공구 방향에서 오리엔테이션 변경 기준 회전 각도

블록 끝에서 각도 값은 `LEAD=value`, `TILT=value` 또는 `THETA=value`를 사용하여 프로그래밍됩니다. 일정한 각도 이외에 세 가지 모든 각도에 대해 최대 5차 폴리노미얼을 프로그래밍할 수 있습니다.

`N... PO[PHI]=(a2, a3, a4, a5)`	리드 각도 LEAD에 대한 폴리노미얼
`N... PO[PSI]=(b2, b3, b4, b5)`	틸트 각도 TILT에 대한 폴리노미얼
`N... PO[THT]=(d2, d3, d4, d5)`	회전 각도 THETA에 대한 폴리노미얼

고차 폴리노미얼 계수가 0인 경우 프로그래밍할 때 생략할 수 있습니다. 예: `PO[PHI]=a2`의 결과는 `LEAD` 각도의 포물선입니다.

의미

경로 기준 공구 오리엔테이션

`ORIPATH:`	경로 기준 공구 오리엔테이션
`ORIPATHS:`	경로 기준 공구 오리엔테이션. 오리엔테이션 특성에서 소음이 감소됩니다.
`LEAD:`	경로 접선 및 표면 법선 벡터에 의해 정의된 평면에서 표면 법선 벡터 기준 각도

TILT:	Z 방향에서 오리엔테이션 회전 또는 경로 접선 기준 회전
THETA:	Z 방향으로의 공구 방향에 대한 회전
PO[PHI]:	LEAD 각도에 대한 방향 폴리노미얼
PO[PSI]:	TILT 각도에 대한 오리엔테이션 폴리노미얼
PO[THT]:	THETA 회전 각도에 대한 오리엔테이션 폴리노미얼

참고

회전 각도 THETA

자체적으로 오리엔테이션 축 역할을 수행하는 세 번째 로터리 축을 통해 공구를 회전시키려면 6축 변환이 필요합니다.

4.9.5.3 경로 기준으로 공구 회전 보간 (ORIROTC, THETA)

회전 벡터를 통한 보간

경로 접선을 기준으로 ORIROTC를 통해 프로그래밍된 공구 회전의 회전 벡터도 THETA 회전 각도를 사용하여 프로그래밍할 수 있는 옵셋을 통해 보간될 수 있습니다. 따라서 PO[THT]를 사용하여 옵셋 각도에 대해 최대 5차 폴리노미얼을 프로그래밍할 수 있습니다.

구문

`N... ORIROTC`	경로 접선 기준 공구 회전 실행
`N... A3= B3= C3= THETA=value`	방향 벡터 회전 정의
`N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)`	최대 5차 폴리노미얼을 통한 옵셋 각도 보간

또한 방향 변경 없이 개별 블록에서도 회전을 프로그래밍할 수 있습니다.

의미

6축 변환으로 경로 기준 공구 회전 보간

ORIROTC:	경로 접선 기준 접선 회전 벡터 실행
THETA=값:	블록 끝에 도달한 회전 각도 (도)
THETA=θe:	회전 벡터의 끝각 Θ_e를 통한 회전 각도
THETA=AC(…):	절대 치수로 넌모달 전환
THETA=IC(…):	증분 치수로 넌모달 전환
PO[THT]=(c2, c3, c4, c5):	5차 폴리노미얼을 통한 옵셋 각도 보간

참고

회전 벡터 보간 ORIROTC

공구 방향과 반대 방향으로 경로 접선 기준 공구 회전 실행은 6축 변환을 통해서만 가능합니다.

활성 ORIROTC 사용

회전 벡터 ORIROTA를 프로그래밍할 수 없습니다. 프로그램을 수행할 경우 알람 14128 "활성 ORIROTC를 통한 공구 회전 절대 프로그래밍"이 출력됩니다.

3축 ~ 5축 변환에 대해 공구 회전 방향

공구 방향은 Euler 각도, RPY 각도 또는 3축 ~ 5축 변환을 통한 방향 벡터를 사용하여 프로그래밍할 수 있습니다. 공간 상의 공구 오리엔테이션 변경 역시 대형 원호 보간 ORIVECT, 오리엔테이션 축 직선 보간 ORIAXES, 테이퍼의 원주면에서의 모든 보간 ORICONxx, 그리고 공구의 두 접점과 공간 상의 곡선을 이용한 보간 ORICURVE을 프로그래밍하여 수행할 수 있습니다.

G....:	로터리 축 이동 세부 정보
X, Y, Z:	직선 축 세부 정보
ORIAXES:	기계 또는 오리엔테이션 축의 직선 보간
ORIVECT:	대형 원호 보간 (ORIPLANE과 동일)
ORIMKS: ORIWKS:	기계 좌표계에서 회전 공작물 좌표계에서 회전 설명은 공구 오리엔테이션 회전 절을 참조하십시오.
A= B= C=:	기계 축 위치 프로그램

ORIEULER:	Euler 각도를 이용한 오리엔테이션 프로그래밍
ORIRPY:	RPY 각도를 이용한 오리엔테이션 프로그래밍
A2= B2= C2=:	가상 축의 각도 프로그램
ORIVIRT1: ORIVIRT2:	가상 오리엔테이션 축을 사용한 오리엔테이션 프로그래밍 (정의 1), MD $MC_ORIAX_TURN_TAB_1에 따라 정의 (정의 2), MD $MC_ORIAX_TURN_TAB_2에 따라 정의
A3= B3= C3=:	방향 축의 방향 벡터 프로그램
ORIPLANE:	평면에서 보간 (대형 원호 보간)
ORICONCW:	테이퍼 원주면에서 CW 방향으로 보간
ORICONCCW:	테이퍼의 원주면에서 CCW 방향으로 보간
ORICONTO:	테이퍼의 원주면에서 접선 트렌지션으로 보간
A6= B6= C6=:	테이퍼의 로터리 축 프로그램 (법선 벡터)
NUT=angle	테이퍼의 개방 각도 (도)
NUT=+179	180°보다 작거나 같은 각도로 이송
NUT=-181	180도보다 크거나 같은 각도로 이송
ORICONIO:	테이퍼의 주변 표면에 대한 보간
A7= B7= C7=:	중간 오리엔테이션 (법선 벡터로 프로그래밍)
ORICURVE XH YH ZH. 예: 폴리노미얼 PO[XH]=(xe, x2, x3, x4, x5) 사용	공구의 두 접점 사이 이동을 지정하는 오리엔테이션 보간. 끝점 이외에 추가 곡선 폴리노미얼도 프로그래밍할 수 있습니다.

참고

활성 ORIAXES로 공구 방향을 오리엔테이션 축을 통해 보간하면 회전 각도는 블록 끝에서 경로를 기준으로만 실행됩니다.

4.9.5.4 방향 특성 유연화 (ORIPATHS A8=, B8=, C8=)

형상에서 일정한 가속으로 발생하는 방향 변경으로 인해 특히 형상 모서리를 포함하여 경로 이동을 따라 원치 않는 중단이 발생할 수 있습니다. 방향 특성의 결과 blip는 개별 중간 블록을 삽입하여 유연화할 수 있습니다. 방향을 다시 지정하는 동안 ORIPATHS를 활성화하면 일정한 가속 상태에서 방향이 변경됩니다. 이 단계에서 공구를 후퇴할 수 있습니다.

장비 제조업체

이 기능을 활성화하는 데 사용된 미리 정의된 기계 및 설정 데이터에 대해서는 장비 제조업체의 참고 사항을 참조하십시오.

머신 데이터는 후퇴 벡터를 해석하는 방법을 설정하는 데 사용할 수 있습니다:

1. TCS에서 Z 좌표는 공구 방향에 의해 정의됩니다.
2. WCS에서 Z 좌표는 활성 평면에 의해 정의됩니다.

'경로에 상대적인 오리엔테이션' 기능에 대해 **자세히 알아보려면** 기능 매뉴얼 변환을 참고하십시오.

구문

전반적으로 경로 기준 일정한 공구 오리엔테이션에 대해 형상 모서리에서 자세한 프로그래밍 세부 정보가 필요합니다. 이 모션의 방향 및 경로 길이는 구성 요소 A8=X, B8=Y C8=Z를 사용하여 벡터를 통해 프로그래밍됩니다.

```
N... ORIPATHS A8=X B8=Y C8=Z
```

의미

ORIPATHS:	경로 기준 공구 오리엔테이션; 방향 특성의 blip가 유연화됩니다.
A8= B8= C8=:	방향 및 경로 길이에 대한 벡터 구성 요소
X, Y, Z:	공구 방향으로 후퇴 이동

참고

방향 벡터 A8, B8, C8 프로그래밍

이 벡터의 길이가 정확하게 0인 경우 후퇴 이동은 실행되지 않습니다.

ORIPATHS

경로 기준 공구 오리엔테이션은 ORIPATHS를 사용하여 활성화됩니다. 그렇지 않을 경우 직선 대형 원호 보간을 통해 시작 오리엔테이션에서 종료 오리엔테이션으로 오리엔테이션이 전송됩니다.

4.9.6 방향 압축 (COMPON, COMPCURV, COMPCAD, COMPSURF)

오리엔테이션 변환 (TRAORI) 이 활성화되고 공구 오리엔테이션이 프로그래밍된 (유형에 관계 없이) NC 프로그램이 지정된 제한 범위 내에 있는 경우 압축할 수 있습니다.

프로그래밍

공구 오리엔테이션

오리엔테이션 변환 (TRAORI) 이 활성화되면 5축 기계의 경우 공구 오리엔테이션을 다음과 같은 방법으로 프로그래밍할 수 있습니다(운동 구조와 관계 없음):

- 다음을 통해 방향 **벡터** 프로그래밍:
  `A3=<...> B3=<...> C3=<...>`
- 다음을 통해 Euler **각도** 또는 RPY-**각도** 프로그래밍:
  `A2=<...> B2=<...> C2=<...>`

공구의 회전

6-축 기계의 경우 공구 오리엔테이션 이외에 공구 회전을 프로그래밍할 수 있습니다.

회전 각도는 다음을 통해 프로그래밍됩니다:

```
THETA=<...>
```

"공구 오리엔테이션 회전 (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (페이지 771)"을 참조하십시오.

참고

회전 또한 프로그래밍되는 NC 블록은 회전 각도가 **선형**으로 바뀌는 경우에만 압축됩니다. 이는 폴리노미얼 PO[THT]=(...)가 회전 각도에 대해 프로그래밍되는 것이 허용되지 않는다는 뜻입니다.

압축 가능한 NC 블록의 일반 구조

압축 가능한 NC 블록의 일반 구조는 다음과 같습니다.

```
N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>
```

또는

```
N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

참고

위치 값은 직접 입력하거나(예: X90) 파라미터 설정을 통해 간접적으로(예: X=R1*(R2+R3)) 입력할 수 있습니다.

로터리 축 위치를 사용하여 공구 오리엔테이션 프로그래밍

또한 로터리 축 위치를 사용하여 공구 오리엔테이션을 지정할 수 있습니다. 예를 들면 다음 구조를 사용합니다:

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

이 경우 압축은 대형 반경 원호 보간의 실행 여부에 따라 두 가지 방법으로 실행됩니다. 대형 반경 원호 보간이 실행되지 않을 경우 방향에서 압축된 변경은 로터리 축의 축 폴리노미얼을 통해 일반적인 방법으로 표시됩니다.

형상 정확도

선택한 압축 모드(MD20482 $MC_COMPRESSOR_MODE)에 따라 설정 데이터를 사용하여 설정된 구성된 축 관련 허용 오차(MD33100 $MA_COMPRESS_POS_TOL) 또는 다음 채널 관련 허용 오차가 압축을 위한 형상 축 및 오리엔테이션 축에 적용됩니다:

SD42475 $SC_COMPRESS_CONTUR_TOL(최대 형상 편차)

SD42476 $SC_COMPRESS_ORI_TOL(공구 오리엔테이션에 대한 최대 각도 편차)

SD42477 $SC_COMPRESS_ORI_ROT_TOL(공구의 회전 각도에 대한 최대 각도 편차) (6축 기계에서만 사용 가능)

활성화/비활성화

압축 기능은 모달 G 명령어 COMPON, COMPCURV, COMPCAD 또는 COMPSURF에 의해 활성화됩니다.

COMPOF는 컴프레서 기능을 종료합니다.

"NC 블록 압축 (COMPON, COMPCURV, COMPCAD, COMPOF) 활성화/비활성화 (페이지 680)"을 참조하십시오.

참고

오리엔테이션 모션은 대형 반경 원호 보간이 활성화되는 경우에만 압축됩니다(예를 들어 공구 오리엔테이션은 시작 및 종료 방향에 의해 결정되는 평면에서 변경됨).

대형 반경 원호 보간은 다음 조건 하에서 실행됩니다.

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  ORIWKS가 활성화되고
  방향이 벡터로 프로그래밍됩니다 (A3, B3, C3 또는 A2, B2, C2 사용).
- MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 및
  ORIVECT 또는 ORIPLANE이 활성화됩니다.
  공구 오리엔테이션은 방향 벡터 또는 로터리 축 위치로 프로그래밍할 수 있습니다. G 명령어 ORICONxx또는 ORICURVE 중 하나가 활성화되거나 방향 각도 폴리노미얼(PO[PHI] 및 PO[PSI])가 프로그래밍되면 대형 원호 보간은 실행되지 않습니다.

예

아래 프로그램 예에서 다각형 정의에 의해 구성된 원이 압축됩니다. 동시에 공구 오리엔테이션이 테이퍼 외부에서 이동합니다. 프로그래밍된 오리엔테이션 변경이 순서대로 실행되지만 불안정하게 컴프레서 기능을 통해 유연한 오리엔테이션 모션이 생성됩니다.

프로그래밍	설명
DEF INT NUMBER=60	
DEF REAL RADIUS=20	
DEF INT COUNTER	
DEF REAL ANGLE	
N10 G1 X0 Y0 F5000 G64 //sort match rate lower first	
$SC_COMPRESS_CONTUR_TOL=0.05	; 형상의 최대 편차 = 0.05 mm
$SC_COMPRESS_ORI_TOL=5	; 오리엔테이션의 최대 편차 = 5°
TRAORI	
COMPCURV	; 이동은 다각형에서 생성된 원호를 그립니다. 방향은 Z 축을 따라 45도의 개방 각도로 테이퍼에서 이동합니다.
N100 X0 Y0 A3=0 B3=-1 C3=1	
N110 FOR COUNTER=0 TO NUMBER	
N120 ANGLE=360*COUNTER/NUMBER	
N130 X=RADIUS*cos(각도) Y=RADIUS*sin(각도) A3=sin(각도) B3=-cos(각도) C3=1	
N140 ENDFOR	

4.9.7 오리엔테이션 특성 활성화/비활성화(ORISON, ORISOF)

"오리엔테이션 특성의 스무딩"은 가공 프로그램에서 G그룹 61의 명령어을 사용하여 활성화/비활성화 됩니다. 명령은 모달식입니다.

전제 조건

- 5/6-축 변환
- 컴프레서 기능 COMPCAD이 활성화됩니다.

구문

```
ORISON
...
ORISOF
```

의미

ORISON:	오리엔테이션 특성 스무딩 활성화
ORISOF:	오리엔테이션 특성 스무딩 비활성화

예

프로그램 코드	코멘트
...	
TRAORI()	; 오리엔테이션 변환 활성화
COMPCAD	; COMPCAD 컴프레서 기능 활성화.
ORISON	; 오리엔테이션 스무딩 활성화.
$SC_ORISON_TOL=1.0	; 공구 오리엔테이션의 최대 각 편차 = 1.0도.
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	; 오리엔테이션 스무딩 비활성화
...	

오리엔테이션은 -45도에서 +45도까지 XZ 평면에서 90로 회전합니다. 오리엔테이션 특성 스무딩으로 인해 오리엔테이션은 더 이상 -45도나 +45도의 최대 각도 값에 도달할 수 없습니다.

4.9.8 운동학적 변형

4.9.8.1 단면 끝부분 변환 활성화 (TRANSMIT)

전면 변환 (TRANSMIT) 은 TRANSMIT 명령문을 사용하여 가공 프로그램 또는 동기 동작에서 실행됩니다.

구문

```
TRANSMIT
TRANSMIT(<n>)
```

의미

TRANSMIT:	첫 번째 TRANSMIT 데이터 세트로 TRANSMIT 활성화
TRANSMIT(n):	n번째 TRANSMIT 데이터 세트로 TRANSMIT 활성화

참고

채널에서 활성인 TRANSMIT 변환은 다음을 통해 활성화됩니다.

- 변환 비활성화: TRAFOOF
- 다른 변환의 활성화: 예: TRACYL, TRAANG, TRAORI

4.9.8.2 실린더 표면 변환을 활성화합니다. (TRACYL)

실린더 표면 변환 (TRACYL) 은 TRACYL 명령문을 사용하여 가공 프로그램 또는 동기 동작에서 실행됩니다.

구문

```
TRACYL(<d>)
TRACYL(<d>,<n>)
TRACYL(<d>,<n>,<k>)
```

의미

TRACYL(<d>):	첫 번째 TRACYL 데이터 세트와 직경 값 <d>로 TRACYL 활성화	
TRACYL (<d>,<n>):	<n> 번째 TRACYL 데이터 세트와 직경 값 <d>로 TRACYL 활성화	
<d>:	기준 또는 직경값 1보다 더 큰 값이어야 합니다.	
<n>:	TRACYL 데이터 세트 번호 (옵션)	
	값 범위:	1, 2

<k>:	파라미터 <k>는 변환 유형 514에만 해당됨	
	k = 0:	홈 측면 수정을 안하는 경우
	k = 1:	홈 측면 수정을 하는 경우
	파라미터가 지정되지 않으면 파라미터가 지정된 기본 위치가 적용됨: $MC_TRACYL_DEFAULT_MODE_<n> <n> = TRACYL 데이터 세트 번호	

참고

채널에서 활성인 TRACYL 변환은 다음을 통해 해제됩니다.

- 변환 비활성화: TRAFOOF
- 다른 변환의 활성화: 예: TRAANG, TRANSMIT, TRAORI

예

프로그램 코드	코멘트
... N40 TRACYL(40.) ...	; 첫 번째 TRACYL 데이터 세트와 직경 값 40mm로 TRACYL 활성화

추가 정보

프로그램 구조

TRACYL 변환 513 (홈 측면 옵셋이 적용된 TRACYL) 을 사용한 홈 밀링을 위한 가공 프로그램은 일반적으로 다음과 같은 단계들로 구성됩니다.

1. 공구를 선택합니다.
2. TRACYL을 선택합니다.
3. 적절한 좌표 옵셋 (프레임) 을 선택합니다.
4. 위치를 선정합니다.
5. OFFN을 프로그램합니다.
6. TRC를 선택합니다.
7. 블록에 접근합니다 (TRC 포지셔닝 및 홈 측면 접근).
8. 중심선 형상을 그루빙합니다.
9. TRC를 선택 해제합니다.
10. 블록을 후퇴시킵니다 (TRC 후퇴 및 홈 측면으로부터 멀리 이동).

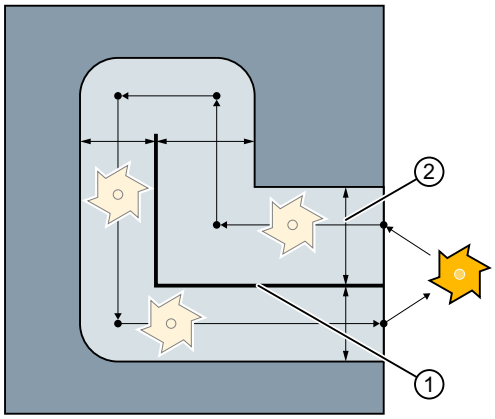
11. 위치를 선정합니다.
12. TRAFOOF.
13. 원래의 좌표 변환 (프레임) 을 다시 선택합니다.

형상 옵셋 (OFFN)

TRACYL 변환 513을 사용하여 홈을 밀링하기 위해, 홈의 중심선과 **홈 너비의 절반**이 OFFN 어드레스를 통해 가공 프로그램에서 프로그램됩니다.

홈 측면 손상을 방지하기 위해 OFFN은 공구 반경 보정이 활성화된 경우에만 실행됩니다.

가공 프로그램에서 OFFN을 변경할 수 있습니다. 이는 홈 중심선을 중심선이 중심에서부터 옵셋될 수 있도록 해줍니다.



① OFFN
② 프로그래밍된 경로

참고

반대편 홈 벽면에 대한 손상을 피하려면 OFFN이 공구 반경보다 크거나 같아야 합니다.

참고

TRACYL을 사용할 때와 TRACYL을 사용하지 않을 때 OFFN은 다르게 작동합니다. TRACYL을 사용하지 않더라도 TRC가 활성화된 경우 OFFN이 포함되어 있으므로 OFFN은 TRAFOOF 이후에 0으로 리셋되어야 합니다.

유의사항
OFFN의 효과는 변환 유형에 따라 달라집니다. TRACYL 변환 513 (홈 측면 옵셋을 통한 TRACYL) 의 경우, 홈 너비의 절반이 OFFN용으로 프로그램됩니다. TRACYL 변환 512 (홈 측면 옵셋을 통한 TRACYL) 의 경우, OFFN의 값이 TRC를 위한 여유로 작용합니다.

공구 반경 보정 (TRC)

TRACYL 변환 513의 경우, TRC는 홈 측면이 아니라 홈의 프로그램된 중심을 기준으로 프로그래밍됩니다. 공구가 홈 측면의 좌측으로 이동하게 하기 위해, 명령문 G41 대신 G42를 프로그래밍하거나 OFFN의 값에 음의 기호를 지정해야 합니다.

공구 직경

TRACYL과 홈의 너비보다 직경이 작은 공구를 사용할 경우, 홈의 너비와 동일한 직경의 공구를 사용할 때와 동일한 홈 측면 기하 형상이 생성되지 않습니다. 정밀도를 높이려면 홈 너비보다 약간 더 좁은 공구 지름을 선택하는 것이 좋습니다.

축 활용

참고

다음 축은 포지셔닝 축 또는 왕복 이동 축으로 사용할 수 없습니다:

- 실린더 표면의 주변 방향에 있는 기하 축 (Y 축).
- 홈 측면 보정에 대한 추가 직선 축 (Z 축)

4.9.8.3 각도 프로그램이 가능한 사각 변환(TRAANG) 활성화 (TRAANG)

각도 프로그램이 가능한 사각 변환은 TRAANG 명령문을 사용하여 가공 프로그램 또는 동기 동작에서 활성화됩니다.

키네마틱 체인 제약

머신 데이터를 이용한 파라미터 설정의 경우, 해당 좌표 축에 대한 경사 축의 각도 α는 머신 데이터를 통해 직교 좌표계에 정의됩니다 각도 α는 TRAANG(<α>)를 통해 호출 시 변경될 수 있습니다. 각도가 변경되면 키네마틱 모델도 변경되므로 다른 작업도 이로 인해 영향을 받을 수 있습니다. 그러므로 각도 지정은 각도가 키네마틱 체인에 기인한 각도와 동일한 경우에만 허용됩니다.

참고

각도 정의

각도는 경사 축이 좌표 축 주위 회전에 기인한 경우에만 정의됩니다. 따라서 좌표 축 중 하나와 평행하지 않고 주 평면에 놓여 있는 기하 축은 하나 뿐입니다.

구문

```
TRAANG
TRAANG()
TRAANG(, <n>)
TRAANG(<α>)
TRAANG(<α>,<n>)
```

의미

TRAANG: TRAANG():	첫 번째 TRAANG 데이터 세트 및 마지막 유효 각도 <α>로 TRAANG 활성화	
TRAANG(, <n>):	<n> 번째 TRAANG 데이터 세트 및 마지막 유효 각도 <α>로 TRAANG 활성화	
TRAANG(<α>):	첫 번째 TRAANG 데이터 세트 및 각도 <α>로 TRAANG 활성화	
TRAANG(<α>,<n>):	<n> 번째 TRAANG 데이터 세트 및 각도 <α>로 TRAANG 활성화	
<α>:	경사 축의 각도 (옵션)	
	값 범위:	-90° < α < + 90°
	머신 데이터에서 파라미터화 된 최초 상태는 각도가 명시되지 않은 경우 효과적입니다. MD2xxxx $MC_TRAANG_ANGLE_<n>	
<n>:	TRAANG 데이터 세트 번호 (옵션)	
	값 범위:	1, 2

참고

채널에서 활성화된 사각 변환 TRAANG 은 다음을 이용하여 비활성화됩니다:

- 변환 비활성화: TRAFOOF
- 다른 변환의 활성화: 예: TRACYL, TRANSMIT, TRAORI

예

프로그램 코드	설명
N20 TRAANG(45)	; 첫 번째 TRAANG 데이터 세트 및 각도 45°로 TRAANG 활성화

4.9.8.4 연마 기계에 대한 사각 플런지 절삭 (G5, G7)

G 명령어 G7 및 G5는 "경사축 (TRAANG)" 을 사용하는 연마 기계에서의 사면 플런지 절삭 프로그램을 간소화하여 플런지 절삭 과정에서 경사축만 이송할 수 있게 하는 데 사용합니다.

플런지 절삭 모션의 필수 종료 위치 프로그래밍은 X 및 Z에서만 이루어져야 합니다. G7의 경우 실제 X축 위치에서 시작하는 NC는 프로그래밍된 종료 위치와 경사 축의 각도 α를 계산 및 접근합니다.

시작 위치 계산은 2개의 직선이 만나는 지점에서 이루어집니다.

- 실제 X축 위치와 멀리 떨어지고 Z축에 대해 평행인 직선
- 프로그래밍된 종료 위치를 통해 경사 각에 대해 평행인 직선

후속 G5와 함께 경사 축은 프로그래밍된 종료 위치로 이송할 수 있습니다.

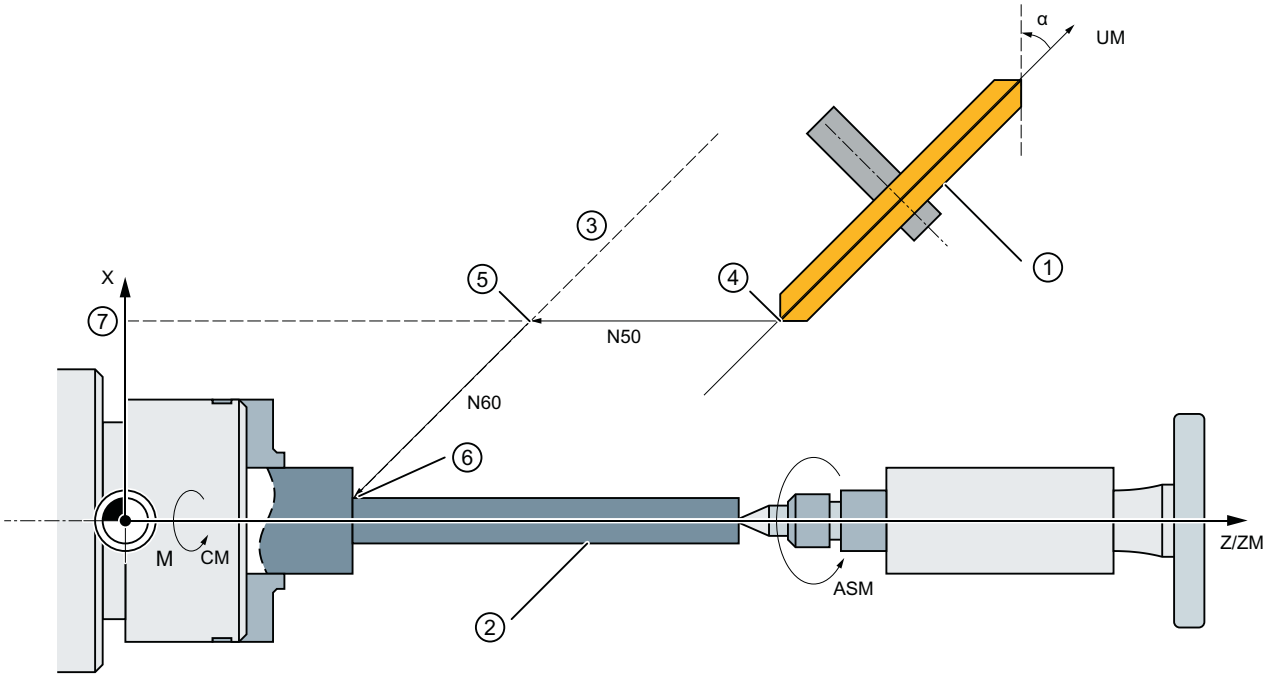
구문

```
G7 <Endpos_X> <Endpos_Z>
G5 <Endpos_X>
```

의미

G7:	사면 플런지 절삭 및 접근에 대한 시작 위치 계산.
G5:	프로그래밍된 종료 위치에 대한 경사축 이송
<Endpos_X>:	X축 끝 위치
<Endpos_Z>:	Z축 종료 위치

예



① 연삭 휠
② 공작물
③ 프로그래밍된 종료 위치를 통해 경사 각에 대해 평행
④ 시작 위치
⑤ 플런지 절삭: 시작 위치
⑥ 플런지 절삭: 종료 위치
⑦ 실제 X축 위치와 멀리 떨어지고 Z축에 대해 평행
X 기하 축
Z 기하 축
ZM 기계 축
UM 기계 축

그림 4-6 경사 축 프로그램

프로그램 코드	설명
N... G18	; XZ 평면 선택.
N40 TRAANG(45.0)	; TRAANG 변환 활성화, 각도 = 45°
N50 G7 X40 Z70 F4000	; 시작 위치 및 접근 계산
N60 G5 X40 F100	; 종료 위치로 경사축 이동.
N70 ...	

4.9.9 연쇄 변환 (TRACON)을 활성화합니다.

구성된 연쇄 변환은 TRAFOON (페이지 819) 또는 TRACON 연산을 통해 가공 프로그램 또는 동기 동작에서 활성화됩니다.

TRACON을 통해 키네마틱 체인으로 파라미터가 설정된 변환을 활성화하려면 시스템 변수 $NT_TRACON_CHAIN에서 이것을 파라미터 설정해야 합니다.

구문

```
TRACON(<Trafo_No>,<Par_1>,...,<Par_n>,<Par_n+1>)
...
TRAFOOF
```

의미

TRACON:	연쇄 변환 활성화 이전에 다른 변환을 실행한 경우 TRACON() 을 통해 암시적으로 비활성화됩니다.		
<Trafo_No>:	연쇄 변환의 번호		
	유형:	INT	
	값 범위:	0 ... 2	
	값:	0, 1	첫 번째/유일한 연쇄 변환
		2	두 번째 연쇄 변환
		명시되지 않음	0 또는 1과 동일한 의미
		참고: 값이 0, 1, 2가 아닌 경우 에러 알람이 발생합니다.	

<Par_1>, ..., <Par_n>, <Par_n+1>:	연쇄 변환의 파라미터	
	<Par_1>, ..., <Par_n>	체인의 첫 번째 변환 파라미터 실제 수는 변환 유형에 따라 달라집니다. • TRAORI: 파라미터 2개 • TRACYL: 파라미터 1~2개
	<Par_n+1>	체인의 두 번째 변환 파라미터 (TRAANG) ≙ 경사 축 각도
	파라미터를 설정하지 않을 경우 기본값 또는 마지막으로 사용한 파라미터가 적용됩니다. 이전 파라미터에 대해 기본 설정을 적용해야 하는 경우 지정된 파라미터가 필요한 순서대로 평가되도록 하려면 쉼표를 사용해야 합니다. 특히, <Trafo_No>를 지정하는 데 필요하지 않더라도 1개 이상의 파라미터 앞에 쉼표가 있어야 합니다(예: TRACON(, 3.7)).	
TRAFOOF:	마지막으로 활성화된 (연쇄된) 변환 비활성화	

예

프로그램 코드	설명
...	
N230 TRACON(1,45.)	; 첫 번째 연쇄 변환 활성화.
	; 이전에 활성화된 변환이 자동으로 선택 해제됩니다.
	; 경사 축 각도는 45°입니다.
...	
N330 TRACON(2,40.)	; 두 번째 연쇄 변환 활성화.
	; 경사 축 각도는 40°입니다.
...	
N380 TRAFOOF	; 두 번째 연쇄 변환 비활성화.
...	

4.9.10 직교 PTP 이송

4.9.10.1 직교 PTP 이송 (PTP, PTPG0, PTPWOC, CP)를 활성화/비활성화합니다.

직교 지점간 즉 PTP 이송이 NC 프로그램에서 G 그룹의 49개 명령어를 사용하여 활성화/비활성화됩니다.

명령은 모달식입니다. 디폴트 설정은 직교 경로 모션 (CP)을 이용한 이송입니다.

CP와 달리 활성 PTP 이송의 경우, 직교 목표점만 변환되고 기계 축들은 동시에 이송됩니다.

직교 목표점이 기계 축 값으로 유일하게 변환되도록 하기 위해서는 위치 및 각도 데이터에 더해 축 위치를 식별할 수 있는 정보가 필요합니다. 이 데이터는 조정 가능한 주소 STAT (페이지 795) 및 TU (페이지 800)에서 얻을 수 있습니다.

요구사항

변환 TRAORI_DYN, TRAORI_STAT, TRANSMIT_K, RCTRA 또는 ROBX가 활성 상태입니다(기존의 TRAORI와 TRANSMIT).

RCTRA 또는 ROBX는 키네마틱 체인에 사용할 수 없습니다.

구문

```
PTP / PTPG0 / PTPWOC
...
CP
```

의미

PTP:	지점 간 모션 PTP 활성화 G0 및 G1 블록에서의 프로그래밍된 직교 위치는 동기 축 모션과 함께 접근됩니다.
PTPG0:	지점 간 모션 PTP G0 활성화 G0 블록에서만 프로그래밍된 직교 위치가 동기 축 모션과 함께 접근됩니다. G1 블록에서는 CP 경로 모션에 대한 전환이 이뤄집니다.
PTPWOC:	포인트-투-포인트 이동 PTPWOC를 활성화합니다(오리엔테이션 변환이 활성화된 경우에만 가능). PTP와 같지만 회전 축 및 오리엔테이션 축 모션에 의해 발생하는 보정 모션이 없습니다.
CP:	지점간 모션 비활성화 및 경로 모션 CP 활성화 직교 경로 모션은 CP를 이용하여 실행됩니다.

참고

PTPWOC

PTPWOC를 RCTRA 또는 ROBX 변환과 함께 이용하는 것은 합당하지 않습니다. RCTRA 또는 ROBX는 키네마틱 체인에 사용할 수 없습니다.

예

참고:

- 예 1: ROBX 변환을 갖는 6-축 로봇의 PTP 이송 (페이지 803)
- 예 2: 일반 5축 변환을 위한 PTP 이송 (페이지 804)
- 예 3: PTPG0 및 TRANSMIT (페이지 805)

4.9.10.2 조인트의 위치를 명시합니다 (STAT).

같은 공구 방향에 대해 여러 조인트 위치가 가능하기 때문에 직교 좌표의 위치 데이터 및 공구 방향에 대한 명시로는 기계 위치를 정확하게 식별할 수 없습니다. 관련된 운동력에 따라 최대 8가지의 조인트 위치가 가능합니다. 이러한 각기 다른 조인트 위치는 특정 변환을 갖습니다.

모호한 부분을 피하기 위해서는 조인트 위치를 STAT 주소 아래 명시해야 합니다.

참고

제어 시스템은 PTP 모션에 대해서만 프로그래밍된 STAT 값을 고려합니다. 활성 변환으로 이송하는 동안 일반적으로 위치 변경이 불가능하기 때문에 CP 모션의 경우 STAT 값이 무시됩니다. 활성 CP를 통한 이송의 경우에는 목표점에 대한 위치가 시작점으로부터 나옵니다.

구문

```
STAT=<값>
```

의미

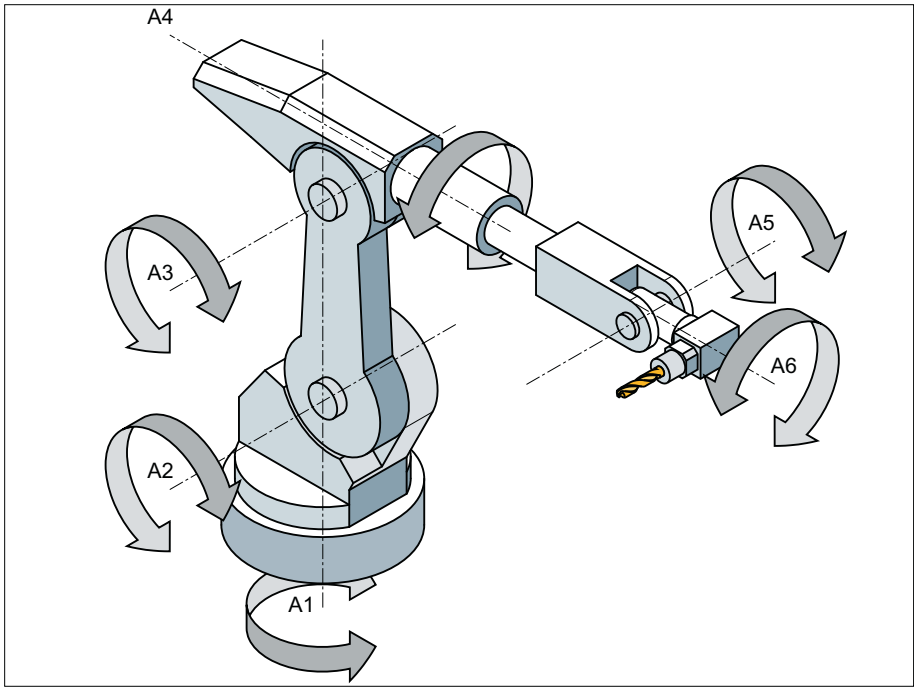
STAT:	조인트 위치를 명시하기 위한 조정 가능한 주소
<value>:	2진수 또는 10진수 가능한 위치 별로 1 비트씩을 포함하고 있습니다. 비트의 중요성은 특정 변환에 의해 정의됩니다.

STAT의 사용은 밀링 스핀들을 갖춘 6-축 연접 로봇 예를 통해 설명됩니다. 동역학 변환은 ROBX 로봇 변환을 사용하여 실현됩니다(사전 조건: 컴파일 싸이클 "RMCC/ROBX Transformation Extended Robotics"가 로딩되고 활성화됨).

참고

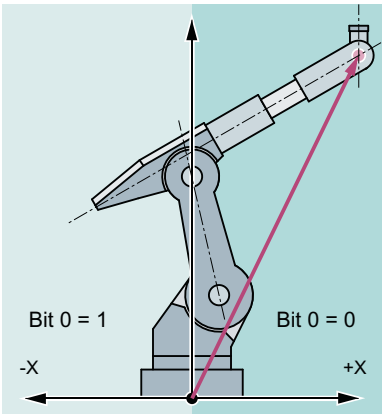
기존의 머신 데이터 파라미터화만 이용한 ROBX의 예

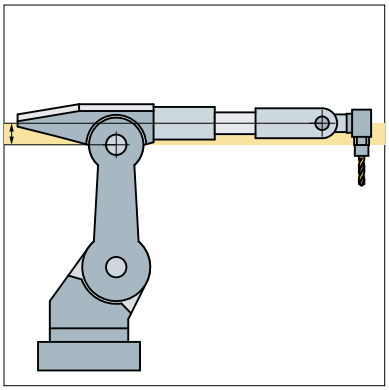
이 예는 키네마틱 체인이 아닌 머신 데이터 파라미터화만으로만 가능합니다.



축 A1, A2 및 A3은 연접 로봇의 메인 축입니다. 헤드 또는 손/손목 축으로도 지정된 축 A4, A5 및 A6는 메인 축과 함께 작업 공간에 위치해 있습니다. 손/손목 축의 추가 동작을 통해 특정 가공 작업의 필요에 따라 공간에서 밀링 스핀들이 방향을 전환할 수 있습니다. 다양한 연접 조인트 위치는 같은 공구 오리엔테이션을 달성할 수 있습니다.

가공에 필요한 연접 조인트 위치는 조정 가능한 STAT 주소의 비트 0 ... 2를 프로그래밍 함으로써 선택됩니다.

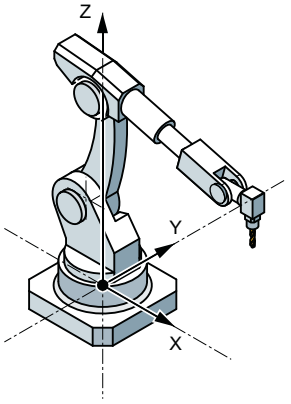
<table>
<tr><td rowspan="3">비트 0</td><td colspan="2">손/손목 축 (A4, A5, A6)의 교차점 위치</td></tr>
<tr><td>= 0</td><td>기본 범위(숄더 오른쪽)
손/손목 축 교차점의 X 값이 A1 좌표계 기준으로 양(+)인 경우, 로봇은 기본 범위 내에 있습니다.</td></tr>
<tr><td>=1</td><td>오버헤드 범위(숄더 왼쪽)
손/손목 축 교차점의 X 값이 A1 좌표계 기준으로 음(-)인 경우, 로봇은 오버헤드 범위에 있습니다.</td></tr>
<tr><td colspan="3"> 

예: 손/손목 축의 교차점이 기본 범위에 있습니다.</td></tr>
<tr><td rowspan="4">비트 1</td><td colspan="2">축 3의 위치
비트 1의 값이 변화하는 곳의 각도는 특정 로봇 유형에 따라 다릅니다. 다음은 축 3과 4가 교차하는 로봇에 적용됩니다.</td></tr>
<tr><td>= 0</td><td>A3 < 0° (엘보우 다운)</td></tr>
<tr><td>=1</td><td>A3 ≥ 0° (엘보우 업)</td></tr>
<tr><td colspan="2">참고:
축 3과 4 사이에 옵셋이 있는 로봇의 경우, 비트 1의 값이 변화하는 곳의 각도는 이 옵셋의 정도에 따라 다릅니다.</td><td> 
A3 및 A4 사이의 옵셋</td></tr>
<tr><td rowspan="3">비트 2</td><td colspan="2">축 5의 위치</td></tr>
<tr><td>= 0</td><td>A5 ≥0° (핸드 플립 없음)</td></tr>
<tr><td>=1</td><td>A5 <0° (핸드 플립)</td></tr>
</table>

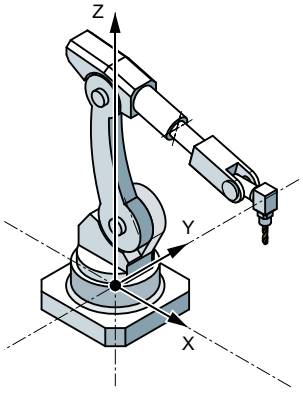
프로그램 예:

프로그램 코드	설명
...	
N14 T="T8MILLD20" D1	; $TC_DP3[1,1]=132.95
N16 ORIMKS	
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 **STAT=...** F2000	; STAT 값은 연접 조인트 위치를 정의합니다(아래 참조).
...	

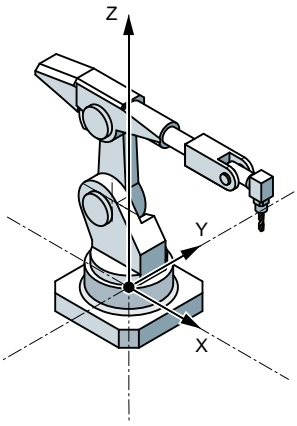
STAT=1 ('B001') → 숄더 왼쪽
→ 엘보우 다운
→ 핸드 플립 없음



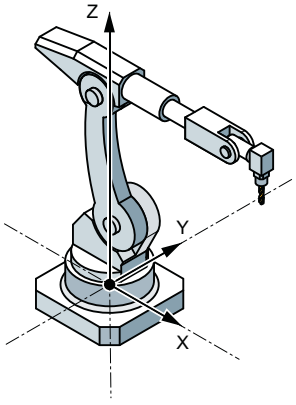
STAT=2 ('B010') → 숄더 오른쪽
→ 엘보우 업
→ 핸드 플립 없음



`STAT=5 ('B101')` → 숄더 왼쪽
→ 엘보우 다운
→ 핸드 플립



`STAT=6 ('B110')` → 숄더 오른쪽
→ 엘보우 업
→ 핸드 플립



TRANSMIT_K (기존 TRANSMIT)

TRANSMIT_K의 경우, 주소 STAT이 극점과 관련한 모호성을 해결하기 위해 사용됩니다.

회전 축이 180°로 회전하거나, CP의 경우 형상이 극점을 통과해야 하는 경우, 다음이 적용됩니다.

비트 0	$NT_POLE_SIDE_FIX[n] = 1 또는 2에 대해서만 적용 머신 데이터 파라미터화의 경우 $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 또는 2:	
	= 0	회전 축은 +180° 회전하며 시계 방향으로 회전합니다.
	=1	회전 축은 -180° 회전하며 시계 반대방향으로 회전합니다.

비트 1	$NT_POLE_SIDE_FIX[n] = 0에 대해서만 적용 머신 데이터 파라미터화의 경우 $MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0:	
	= 0	축은 극점을 통과합니다. 축은 회전하지 않습니다.
	=1	축은 극점 주변을 회전합니다. STAT의 비트 0에 관한 것입니다.

4.9.10.3 축 각도 (TU) 표시를 명시하십시오.

특수 이송 전략 없이 (예: 중간 지점) +180° 보다 크거나 -180° 보다 적은 축 각도에 회전 축이 접근하기 위해서는 회전 각도 표시가 조정 가능한 주소 TU 밑에 반드시 명시되어야 합니다.

참고

제어 기능은 PTP 모션에 대해 프로그래밍된 TU 값만 고려합니다. CP 모션은 무시됩니다.

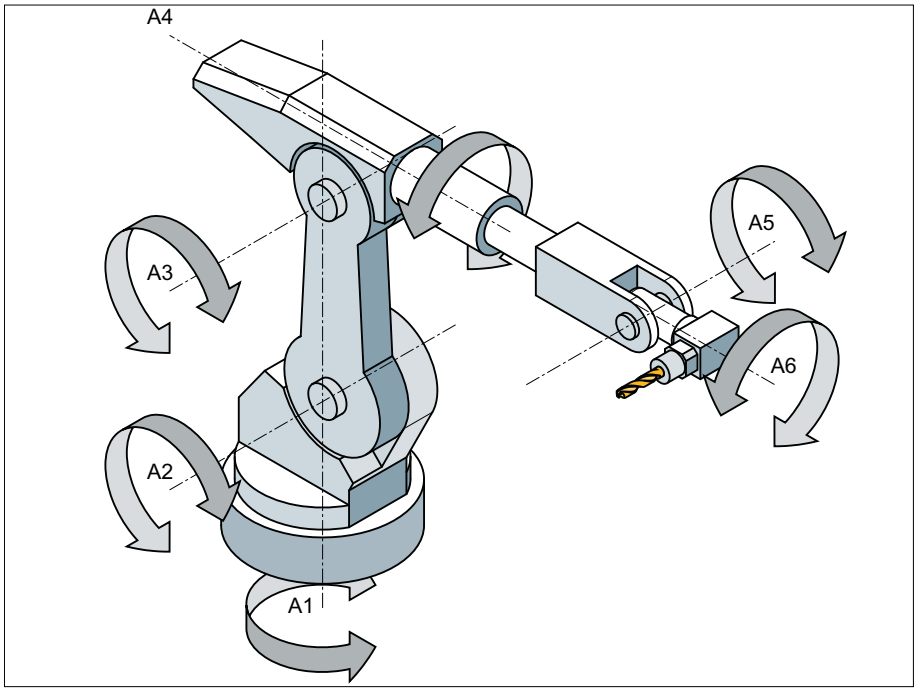
구문

```
TU=<Value>
```

의미

TU:	축 각도 표시를 명시하기 위한 조정 가능한 주소			
<value>:	2진수 또는 10진수 변환에 관여하는 각 축의 경우, 축 각도(θ)에 대한 표시를 알려주는 비트가 있으며, 결과적으로 이송 방향도 알려줍니다.			
	비트	= 0	축 각도 신호: +	축 각도 범위: $0° \le \theta < 360°$
		=1	축 각도 신호: -	축 각도 범위: $-360° < \theta < 0°$

예: 6-축 연접 로봇



비트	의미	값	축 각도 신호	축 각도
비트 0 [1]	A1의 축 각도를 위한 신호	= 0	+/-	≥ 0°
		=1	-	< 0°
비트 1 [1]	A2의 축 각도를 위한 신호	= 0	+/-	≥ 0°
		=1	-	< 0°
비트 2 [1]	A3의 축 각도를 위한 신호	= 0	+/-	≥ 0°
		=1	-	< 0°
비트 3 [1]	A4의 축 각도를 위한 신호	= 0	+/-	≥ 0°
		=1	-	< 0°
비트 4 [1]	A5의 축 각도를 위한 신호	= 0	+/-	≥ 0°
		=1	-	< 0°
비트 5 [1]	A6의 축 각도를 위한 신호	= 0	+/-	≥ 0°
		=1	-	< 0°

[1] 로봇 축의 채널 축 번호로부터 얻어진 실제 TU 바이트 번호! 예를 들어, 로봇 축(A1~A6)은 채널에서 첫 번째 축입니다. 결과적으로 TU 바이트 0 ... 5가 사용됩니다. 로봇 축의 또 다른 채널 축 지정과 관련, 로봇 축의 TU 바이트 번호가 이에 따라 변합니다(예: 로봇 축은 세 번째에서 8번째 채널 축인데, 이는 TU 바이트 2 ... 7이 로봇 축에 대해 사용되는 것임).

따라서 TU=19 (TU='B010011에 해당)는 다음을 의미합니다.

비트	값		축 각도
0	=1	⇒	$\theta_{A1} < 0°$
1	=1	⇒	$\theta_{A2} < 0°$
2	= 0	⇒	$\theta_{A3} \geq 0°$
3	= 0	⇒	$\theta_{A4} \geq 0°$
4	=1	⇒	$\theta_{A5} < 0°$
5	= 0	⇒	$\theta_{A6} \geq 0°$

참고

이송 범위가 >±360°인 축의 경우, 축은 항상 가장 짧은 경로를 따라 움직입니다. 왜냐하면 축 위치가 TU 정보에 의해 유일하게 명시될 수 없기 때문입니다.

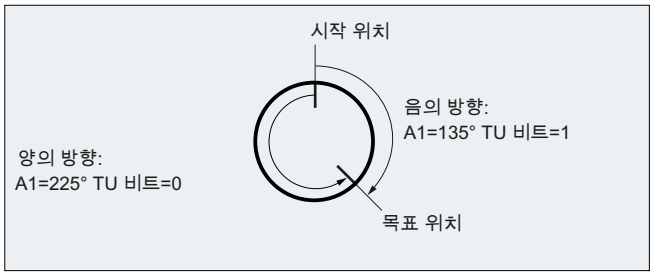
위치에 대해 프로그래밍된 TU가 없으면 MD30455 $MA_MISC_FUNCTION_MASK에 따라 짧거나 긴 경로가 이송됩니다.

TRANSMIT_K (기존 TRANSMIT)

활성 TRANSMIT를 이용한 PTP 이동인 경우, TU 주소는 무의미합니다.

예

그림에 표시된 나와 있는 회전 축 위치는 음 또는 양의 방향으로 접근할 수 있습니다. 각도 위치는 주소 A1 하에서 프로그래밍됩니다. 이송 방향은 TU가 명시될 때에만 절대적으로 명확해집니다.



4.9.10.4 예 1: ROBX 변환을 갖는 6-축 로봇의 PTP 이송

다음 적용 예에서는 직교 PTP 이송 및 관련 NC 명령어가 예의 형태로 보여집니다.

참고

기존의 머신 데이터 파라미터화만 이용

이 예는 키네마틱 체인에는 해당되지 않습니다.

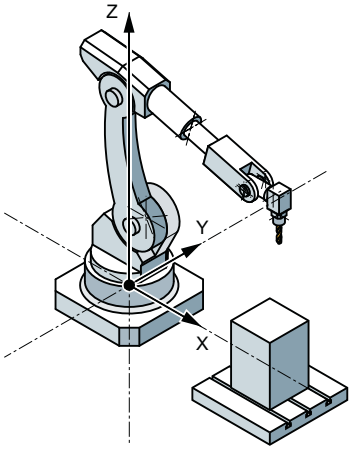


그림 4-7 밀링 스핀들을 가진 6-축 연접 로봇

```
N1 G90
N2 T="T8MILLD20" D1 M6
N3 TRAORI
;$P_UIFR[1]=CTRANS(X,1500,Y,0,Z,400):CROT(X,0,Y,0,Z,-90)
N4 G54
N5 M3 S20000
N6 ORIWKS
N7 ORIVIRT1
N8 CYCLE832(0.01,_FINISH,1)
;HOME
N9 TRAFOOF
N10 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N11 TRAORI
N12 G54
N13 G0 PTP X1369.2426 Y956.7528 Z502.5517 A=135.5761 B=-33.2223
C=161.1435 STAT='B010' TU='B001011'
N14 G0 X1355.1242 Y1014.9394 Z424.9695 A=135.8491 B=-33.1439
C=160.9941 STAT='B010' TU='B001011'
```

```
N15 G1 CP X1354.8361 Y1016.1269 Z423.3862 A=136.0635 B=-33.0819 C=160.8770
F1000
N16 G1 X1336.4283 Y1016.1269 Z426.6311 A=136.0484 B=-32.2151 C=160.9643
F2000
N17 G1 X1317.9831 Y1016.1269 Z429.6730 A=136.0175 B=-31.3394 C=161.0655
;HOME
N18 TRAFOOF
N19 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N20 M30
```

4.9.10.5 예 2: 일반 5축 변환을 위한 PTP 이송

가정: 기본으로 사용된 직각 CA 운동.

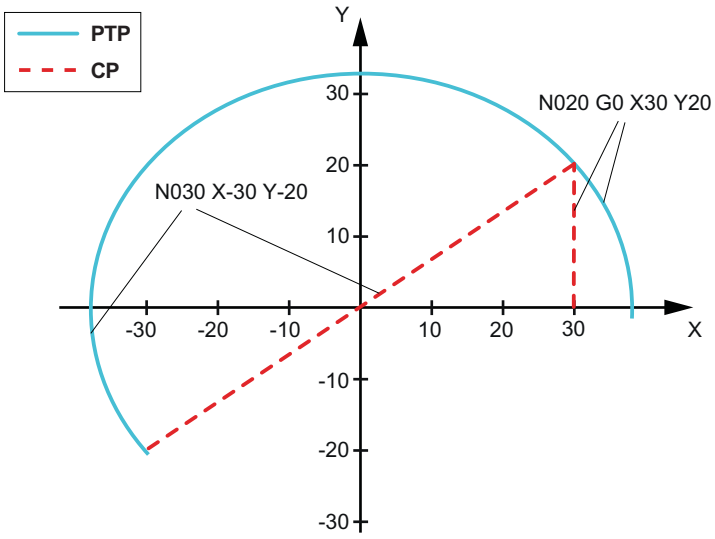
프로그램 코드	코멘트
TRAORI	; 변환 CA 운동 실행
PTP	; PTP 이송 활성화
N10 A3=0 B3=0 C3=1	; 회전 축 위치 C=0 A=0
N20 A3=1 B3=0 C3=1	; 회전 축 위치 C=90 A=45
N30 A3=1 B3=0 C3=0	; 회전 축 위치 C=90 A=90
N40 A3=1 B3=0 C3=1 **STAT=1**	; 회전 축 위치 C=270 A=-45

로터리 축 위치에 대한 명확한 접근 위치를 선택합니다:

STAT=1 프로그래밍의 결과로서, 블록 N40에서는 회전 축이 시작점 (C=90, A=90)에서 종료 지점 (C=270, A=–45)까지 더 먼 거리를 이동합니다. 반면 **STAT=0** 인 경우, 회전 축은 종료 지점 (C=90, A=45)까지 가장 짧은 경로를 따라 이동합니다.

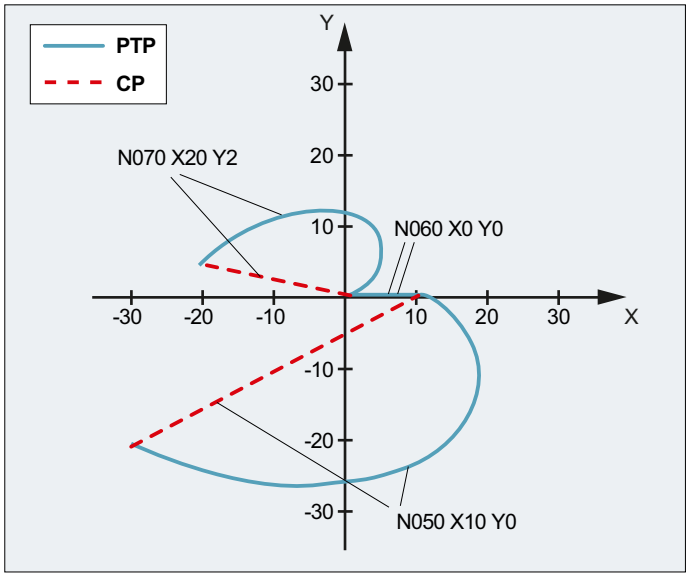
4.9.10.6 예 3: PTPG0 및 TRANSMIT

PTPG0 및 TRANSMIT를 사용한 극점 주변에서의 이송



프로그램 코드	코멘트
N001 G0 X30 Z0 F10000 T1 D1 G90	;초기 설정, 절대 치수
N002 SPOS=0	
N003 **TRANSMIT**	;TRANSMIT 변환
N010 **PTPG0**	; 각 G0 블록의 경우 PTP는 자동, 그러고 나서 다시 CP
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

PTPG0 및 TRANSMIT를 사용한 극점으로부터의 이송



프로그래밍	코멘트
N001 G0 X90 Z0 F10000 T1 D1 G90	;초기 설정
N002 SPOS=0	
N003 **TRANSMIT**	;TRANSMIT 변환
N010 **PTPG0**	; 각 G0 블록의 경우 PTP는 자동, 그러고 나서 다시 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	

4.9.11 변환 선택 시 제한사항

기능

가공 프로그램 또는 MDA를 통해 변환을 선택할 수 있습니다. 주:

- 중간 이동 블록이 삽입되지 않습니다(모따기/반경).
- 스플라인 블록 순서는 제외되어야 합니다. 그렇지 않을 경우 메시지가 표시됩니다.
- 미세 공구 보정은 선택 취소해야 합니다(FTOCOF). 그렇지 않을 경우 메시지가 표시됩니다.
- 공구 반경 보정은 선택 취소해야 합니다(G40). 그렇지 않을 경우 메시지가 표시됩니다.
- 활성화된 공구 길이 옵셋은 제어 시스템에 의해 변환에 포함됩니다.
- 제어 시스템은 변환하기 전에 현재 프레임 활성을 선택 취소합니다.
- 제어 시스템은 변환에 의해 영향을 받은 활성 작동 영역 제한을 선택 취소합니다 (WALIMOF).
- 보호 영역 모니터링이 선택 취소됩니다.
- 연속 경로 제어 및 라운딩이 중단됩니다.
- 기계 데이터에서 지정된 모든 축이 블록을 기준으로 동기화되어야 합니다.
- 교환된 축이 다시 교환됩니다. 그렇지 않을 경우 메시지가 표시됩니다.
- 메시지는 종속 축에 대해 출력됩니다.

공구 변경

공구는 공구 반경 보정 기능을 선택 취소할 경우에만 교환할 수 있습니다.

공구 길이 옵셋 및 공구 반경 보정 선택/선택 취소 변경은 동일한 블록에서 프로그래밍하지 않아야 합니다.

프레임 변경

기본 좌표계를 전적으로 참조하는 모든 명령이 허용됩니다(FRAME, 공구 반경 보정). 그러나 비활성 변환과 달리, G91을 통한 프레임 변경(증분 치수)은 개별적으로 처리되지 않습니다. 이송할 증분은 이전 블록에 적용된 프레임과 관계 없이 새 프레임의 공작물 좌표계에서 평가됩니다.

예외

변환이 적용된 축은 다음으로 사용할 수 없습니다:

- 미리 설정된 축(알람),
- 확인 점 복귀의 경우(알람),
- 원점 복귀의 경우(알람).

4.9.12 변환 선택 해제 (TRAFOOF)

사전 정의된 TRAFOOF 절차는 모든 활성 변환 및 프레임을 비활성화합니다.

참고

변환 선택을 취소하려면 선택 시와 같은 2차 조건들 (페이지 807)이 적용됩니다.
이후 필요한 프레임은 갱신된 프로그램에 의해 활성화되어야 합니다.

구문

```
...
TRAFOOF
```

의미

TRAFOOF:	모든 활성 변환/프레임 비활성화

4.10 키네마틱 체인

4.10.1 콤포넌트 삭제 (DELOBJ)

DELOBJ() 기능은 할당된 시스템 변수를 디폴트 값으로 재설정해서 구성 요소를 "삭제"합니다.

- 운동학적 체인의 요소
- 보호 영역, 보호 영역 요소 및 충돌 쌍
- 트렌스포메이션 데이터

구문

```
[<RetVal>=] DELOBJ(<CompType>[,,,<NoAlarm>)])
[<RetVal>=] DELOBJ(<CompType>,<Index1>[,,<NoAlarm>])
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][,<Index2>][,<NoAlarm>])
```

의미

DELOBJ:	좌표계 체인, 보호 영역, 보호 영역 요소, 충돌 페어 및 변환 데이터에서 요소 삭제

<CompType>:	삭제할 콤포넌트 유형
	데이터 유형: STRING
	값: "KIN_CHAIN_ELEM" **의미**: 모든 운동학적 요소의 시스템 변수: $NK_...
	값: "KIN_CHAIN_SWITCH" **의미**: 시스템 변수 $NK_SWITCH[<i>]
	값: "KIN_CHAIN_ALL" **의미**: 모든 운동학적 요소 및 스위치가 "KIN_CHAIN_ELEM" 및 "KIN_CHAIN_SWITCH"를 사용한 DELOBJ의 연속 콜과 같습니다.
	값: "PROT_AREA" **의미**: 보호 영역의 시스템 변수: • $NP_PROT_NAME • $NP_CHAIN_NAME • $NP_CHAIN_ELEM • $NP_1ST_PROT
	값: "PROT_AREA_ELEM" **의미**: 기계 보호 영역 및/또는 자동 공구 보호 영역의 보호 영역 요소 시스템 변수: • $NP_NAME • $NP_NEXT • $NP_NEXTP • $NP_COLOR • $NP_D_LEVEL • $NP_USAGE • $NP_TYPE • $NP_FILENAME • $NP_PARA • $NP_OFF • $NP_DIR • $NP_ANG
	값: "PROT_AREA_COLL_PAIRS" **의미**: 충돌 쌍의 시스템 변수: • $NP_COLL_PAIR • $NP_SAFETY_DIST

	값: "PROT_AREA_ALL" **의미**: 모든 보호 영역, 보호 영역 요소 및 충돌 쌍 (시스템 변수 $NP_...)이 "PROT_AREA," "PROT_AREA_ELEM," 및 "PROT_AREA_COLL_PAIRS"를 사용한 DELOBJ의 연속 콜과 같습니다.	
	값: "TRAFO_DATA" **의미**: 모든 변환의 시스템 변수 $NT_...	
<Index1>:	삭제할 첫 번째 콤포넌트의 인덱스 (**선택 사항**)	
	데이터 유형:	INT
	디폴트 값:	-1
	값 범위:	-1 ≤ x ≤ (구성된 콤포넌트 최대 개수 -1)
	값	**의미**
	0, 1, 2, ...	삭제할 콤포넌트의 인덱스.
	-1	명시된 유형의 모든 구성요소가 삭제됩니다. <인덱스2>는 평가되지 않습니다.
<Index2>:	삭제할 마지막 콤포넌트의 인덱스 (**선택 사항**) <Index2>가 프로그래밍되지 않은 경우 <Index1>에 명시된 콤포넌트의 시스템 변수만 삭제됩니다.	
	데이터 유형:	INT
	디폴트 값:	<Index1>에 명시된 콤포넌트의 시스템 변수만 삭제됩니다.
	값 범위:	<Index1> < x ≤ (구성된 콤포넌트 최대 개수 -1)
<NoAlarm>:	알람 억제 (**옵션**)	
	데이터 유형:	BOOL
	디폴트 값:	FALSE
	값	**의미**
	FALSE	에러 (<RetVal> < 0) 가 발생하면 프로그램 처리가 중단되고 알람이 표시됩니다..
	TRUE	에러가 발생해도 프로그램 처리가 중단되지 않고 알람도 표시되지 않습니다. 용도: 반환값에 따른 사용자별 반응 지정

<RetVal>:	기능 반환값	
	데이터 유형:	INT
	값 범위:	0, -1, -2, ... -7
	값	**의미**
	0	에러가 발생하지 않았습니다.
	-1	파라미터 없이 기능 호출. 적어도 <Comp Type> 파라미터는 지정해야 합니다.
	-2	<Comp Type>이 알 수 없는 콤포넌트를 지정했습니다.
	-3	<Index1> -1보다 작습니다.
	-4	<Index1>이 구성된 콤포넌트 개수보다 큽니다.
	-5	콤포넌트 **그룹** 삭제 시 <Index1>의 값이 -1이 아닙니다.
	-6	<Index2>가 <Index1>보다 적습니다.
	-7	<Index2>가 구성된 콤포넌트 개수보다 큽니다.
	-8	불충분한 쓰기 권한

4.10.2 이름을 이용한 인덱스 결정 (NAMETOINT)

사용자 특정 이름은 유형의 시스템 변수 배열에 입력됩니다STRING. 시스템 변수의 식별자 및 이름에 기초하여 NAMETOINT() 기능은 시스템 변수 배열 내 저장 시 사용되는 이름에 속하는 인덱스 값을 결정합니다.

구문

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

의미

NAMETOINT:	시스템 변수 인덱스 결정	
<SysVar>:	유형의 시스템 변수 배열 이름STRING	
	데이터 유형:	STRING
	값 범위:	유형의 모든 NC 시스템 변수 배열 이름STRING
<Name>:	시스템 변수 인덱스 결정 기준이 되는 문자열 또는 이름.	
	데이터 유형:	STRING

<NoAlarm>:	알람 억제 (**옵션**)	
	데이터 유형:	BOOL
	디폴트 값:	FALSE
	값	**의미**
	TRUE	에러가 발생해도 프로그램 처리가 중단되지 않고 알람도 표시되지 않습니다. 용도: 반환값에 따른 사용자별 반응 지정
	FALSE	에러 (<RetVal> < 0) 가 발생하면 프로그램 처리가 중단되고 알람이 표시됩니다.
<RetVal>:	시스템 변수 인덱스 또는 에러 메시지	
	데이터 유형:	INT
	값 범위:	-1 ≤ x ≤ (구성된 콤포넌트 최대 개수 -1)
	값	**의미**
	≥ 0	찾는 이름을 명시된 시스템 변수 인덱스 하에서 발견했습니다.
	-1	이름을 찾지 못했거나 에러가 발생했습니다.

예

프로그램 코드	**주석**
`DEF INT INDEX`	
`$NP_PROT_NAME[27]="Cover"`	
`...`	
`INDEX = NAMETOINT("$NP_PROT_NAME","Cover")`	`; INDEX == 27`

4.11 키네마틱 체인의 충돌 방지

참고

보호 영역

다음 장에 명시된 보호 영역은 '기하 기계 모델링' 기능을 말합니다.
이 기능에 대해 **알아보려면** 기능 매뉴얼 모니터링 및 보정을 참고하십시오.

4.11.1 충돌 쌍 점검 (COLLPAIR)

4.11.2 충돌 방지용 기계 모델의 요청 재계산(PROTA)

좌표 체인 $NK_...의 시스템 변수, 기하 머신 모델링 또는 충돌 방지 $NP_...가 가공 프로그램에 쓰여진 경우, 변화가 충돌 방지를 위한 NC 내부 기계 모델에서 실효성을 가질 수 있도록 PROTA 절차는 반드시 호출되어야 합니다.

구문

```
PROTA[(<Par>)]
```

의미

PROTA:	충돌 방지용 기계 모델의 요청 재계산(PROTA) • 전처리 정지를 트리거합니다. • 블록에 단독으로 있어야 합니다.		
<Par>:	파라미터(**옵션**)		
	데이터 유형:	STRING	
	값:	---	파라미터 없음 기계 모델은 재계산됩니다. 보호 영역의 상태는 그대로 유지됩니다.
		"R"	기계 모델은 재계산됩니다. 보호 영역은 $NP_INIT_STAT 에 의거하여 최초 상태로 설정됩니다.

보충 조건

시뮬레이션

PROTA 절차를 시뮬레이션(simNC)과 관련하여 가공 프로그램에 사용해서는 안됩니다.

예: 시뮬레이션이 활성화된 상태에서의 PROTA 호출 회피.

프로그램 코드	코멘트
...	
IF $P_SIM == FALSE	IF 시뮬레이션 비활성
PROTA	THEN 충돌 모델 재계산
ENDIF	; ENDIF

프로그램 코드	코멘트
...	

다음 또한 참고:

보호 영역 상태 설정 (PROTS) (페이지 816)

4.11.3 보호 영역 상태 설정 (PROTS)

PROTS() 절차는 보호 영역의 상태를 특정 값으로 설정합니다.

구문

```
PROTS(<State>[,<Name_1>,...,<Name_n>])
```

의미

PROTS:	보호 영역의 상태를 설정합니다.		
	• 블록에 단독으로 있어야 합니다.		
<State>:	지정된 보호 영역에 설정될 상태		
	데이터 유형:	CHAR	
	값:	"A" 또는 "a"	상태: 활성
		"I" 또는 "i"	상태: 비활성
		"P" 또는 "p"	상태: 사전 활성화되거나 PLC에 의해 제어됨 [1]
		"R" 또는 "r"	상태: 초기화 상태의 NC 내부 값 [2]

<Name_1> ... <Name_n>:	지정된 상태로 설정될 하나 또는 그 이상의 보호 영역 이름(**옵션**) 이름을 지정하지 않으면 정의된 모든 보호 영역에 지정된 상태가 설정됩니다.	
	데이터 유형:	STRING
	값 범위:	파라미터로 설정된 보호 영역 이름
	참고: 파라미터로 지정할 수 있는 최대 보호 영역 개수는 프로그램 라인당 최대 문자 개수에 따라 달라집니다.	
[1] 활성화/비활성화는 다음을 통해 실시됩니다: DB10.DBX234.0 - DBX241.7 [2] 상태가 초기화 상태의 NC 내부 값, 즉 시스템 변수 $NP_INIT_STAT 가 마지막 PROTA( ) (페이지 815) 호출 시 가지고 있던 값으로 설정됩니다.		

4.11.4 두 보호 영역의 간격 결정 (PROTD)

PROTD() 기능은 두 보호 영역 간의 간격을 계산합니다.

기능 속성:

- 간격 계산은 보호 영역 상태 (활성화, 비활성화, 사전 활성화) 와 관계 없이 수행됩니다.
- 두 보호 영역 사이의 간격을 계산하는 데는 $NP_USAGE = "C" or "A"로 표시된 보호 영역 요소만 사용됩니다. $NP_USAGE = "V"로 표시된 보호 영역의 요소는 검토되지 않습니다.
- 모든 요소가 $NP_USAGE = "V"로 표시된 보호 영역은 간격 계산에 사용될 수 없습니다.
- 이전 블록 끝에서 유효했던 위치를 적용하여 간격을 계산합니다.
- 메인 런 계산에 포함된 오버레이 (예: DRF 옵셋 또는 외부 워크 옵셋) 는 **해석 시간** 기능에 설정된 값을 이용한 간격 계산에 포함시킵니다.

참고

동기화

PROTD() 기능을 사용할 때 메인 런 및 전처리를 동기화하는 것은, 그리고 필요 시 STOPRE 전처리 정지와 동기화하는 것은 전적으로 사용자의 책임입니다.

충돌

지정된 두 보호 영역 사이에 충돌이 발생하면 기능은 0.0 간격을 반환합니다. 두 보호 영역이 서로 닿거나 교차하면 충돌이 발생합니다.

충돌 검사를 위한 안전 거리(MD10622 $MN_COLLISION_SAFETY_DIST)는 간격 계산에 포함되지 않습니다.

구문

```
[<RetVal> =] PROTD([<Name_1>],[<Name_2>],VAR <Vector>[,<System>])
```

의미

PROTD:	지정된 두 보호 영역 간의 간격을 계산합니다.		
	• 블록에 단독으로 있어야 합니다.		
<RetVal>:	기능 반환값: 두 보호 영역의 간격 절대치 또는 충돌이 있는 경우 0.0 (위의 충돌 설명 참조)		
	데이터 유형:	REAL	
	값 범위:	0.0 ≤ x ≤ +max. 실제 값	
<Name_1>, <Name_2>:	간격을 계산할 두 보호 영역의 이름(**옵션**)		
	데이터 유형:	STRING	
	값 범위:	파라미터로 설정된 보호 영역 이름	
	디폴트 값:	"" (빈 스트링) 보호 영역이 지정되지 않은 경우 PROTD 기능은 충돌 모델에서 활성화 및 사전 활성화된 모든 보호 영역에서 현재 최소 간격을 계산합니다.	
<Vector>:	반환값: 다음 항목과 함께 한 보호 영역 <Name_2>부터 다른 보호 영역 <Name_1> 까지의 3차원 간격 벡터: • <Vector>[0]: 절대 좌표계의 X 좌표 • <Vector>[1]: 절대 좌표계의 Y 좌표 • <Vector>[2]: 절대 좌표계의 Z 좌표 충돌의 경우: <Vector> == 0 벡터		
	데이터 유형:	VAR REAL [3]	
	값 범위:	<Vector> [n]: 0.0 ≤ x ≤ ±최대 REAL 값	
<System>:	간격 및 간격 벡터에 사용되는 측정 시스템 (인치/미터) (**옵션**)		
	데이터 유형:	BOOL	
	값:	FALSE (디폴트)	G 그룹 13의 현재 활성 G 코드 (G70, G71, G700, G710))에 해당하는 측정 시스템.
		TRUE	설정된 베이직 시스템에 해당하는 측정 시스템: MD52806 $MN_ISO_SCALING_SYSTEM

4.12 운동학적 체인을 이용한 변환

4.12.1 변환 활성화(TRAFOON)

키네마틱 체인을 이용해 정의된 변환은 사전 지정된 TRAFOON 절차로 활성화됩니다. 이 호출은 블록에 단독으로 있어야 합니다.

참고

또는 키네마틱 체인으로 정의된 변환은 TRAORI 이나 TRANSMIT 등 기존 NC 명령으로도 활성화할 수 있습니다. 이를 위해 0이 아닌 적절한 값이 $NT_TRAFO_INDEX 시스템 변수에 입력되어야 합니다.

$NT_TRAFO_INDEX에 관한 자세한 내용은 "시스템 변수 목록 매뉴얼"을 참조하십시오.

구문

```
TRAFOON(<Trafoname>,<Diameter>,<k>)
```

의미

TRAFOON:	키네마틱 체인을 이용해 지정된 변환 활성화 절차	
<Trafoname>:	변환 데이터 세트 이름	
	데이터 유형:	STRING
	값 범위:	$NK_NAME을 통해 지정된 모든 변환 데이터 세트 이름
	주의: 변환 데이트 세트의 이름은 고유해야 합니다. 이것은 $NT_NAME에서 한 번만 발생해야 합니다.	
<Diameter>:	기준 또는 작업 직경(TRACYL만 해당)	
	데이터 유형:	REAL
	값은 > 1여야 합니다.	

<k>:	홈 측면 옵셋의 사용을 지정합니다(TRACYL만 해당).		
	데이터 유형:	BOOL	
	값:	FALSE	홈 측면 옵셋이 없는 경우
		TRUE	홈 측면 옵셋이 있는 경우
	TRACYL 변환 유형 514(홈 측면 옵셋 프로그래밍 가능)에 해당합니다. <k>가 지정되지 않으면 $NT_CNTRL[<n>]에서 비트 10의 파라미터화된 설정이 적용됩니다.		

예

프로그램 코드	설명
`TRAFOON["Trans_1"]`	`Trans_1` 이름으로 변환을 활성화합니다.

4.12.2 기계 측정(CORRTRAFO) 후 오리엔테이션 변환 수정

키네마틱 체인을 통해 지정된 오리엔테이션 변환이 있는 기계의 경우, 사용자는 사전 지정된 CORRTRAFO 기능을 사용하여 기계 측정 후 기계의 키네마틱 모델에 있는 오리엔테이션 축의 옵셋 벡터나 방향 벡터를 수정할 수 있습니다.

참고

CORRTRAFO 기능으로 작성한 수정 값은 변환에 즉시 적용되지 않습니다. 변환 선택 해제, NEWCONF 및 변환 선택이 완료될 때까지 수정 값이 적용되지 않습니다.

구문

```
<Corr_Status> = CORRTRAFO(<Corr_Vect>, <Corr_Index>, <Corr_Mode>,
[ <No_Alarm>])
```

의미

CORRTRAFO:	기능 호출

<Corr_Status>:	기능 반환값		
	데이터 유형:	INT	
	값:	0	기능이 에러 없이 실행되었습니다.
		1	활성 상태의 변환이 없습니다.
		2	현재 활성 변환은 오리엔테이션 변환이 아닙니다.
		3	활성 오리엔테이션 변환이 키네마틱 체인으로 지정되지 않았습니다.
		10	<Corr_Index> 호출 파라미터가 음수입니다.
		11	<Corr_Mode> 호출 파라미터가 음수입니다.
		12	서브체인 섹션에 대해 유효하지 않은 참조(<Corr_Index>의 일의 자리) 값은 서브체인의 오리엔테이션 축 개수 이하여야 합니다.
		13	서브체인의 오리엔테이션 축에 대해 유효하지 않은 참조(<Corr_Index>의 일의 자리) 값은 서브체인의 오리엔테이션 축 개수 미만이어야 합니다.
		14	서브체인에 대한 유효하지 않은 참조(<Corr_Index> 십의 자리). 값이 0과 1만 허용됩니다(부품 또는 공구 체인 참조). 이 에러 번호는 <Corr_Index>이 참조하는 서브체인이 존재하지 않을 때 발생합니다.
		15	<Corr_Index> 파라미터로 참조되는 섹션에 수정 요소가 없습니다($NT_CORR_ELEM_P 또는 $NT_CORR_ELEM_T).
		20	유효하지 않은 수정 모드(<Corr_Mode>의 일의 자리) 값이 0과 1만 허용됩니다.
		21	유효하지 않은 수정 모드(<Corr_Mode>의 십 또는 백의 자리). 축 방향을 작성할 때 일의 자리만 0이 아닐 수 있습니다.
		30	<Corr_Mode>의 백의 자리가 유효하지 않습니다. 값이 0과 1만 허용됩니다.
		31	<Corr_Mode>의 천의 자리가 유효하지 않습니다. 값이 0과 1만 허용됩니다.
		40	축 방향으로 가져오는 방향 벡터는 제로 벡터입니다. 이 에러는 <Corr_Mode>의 천의 자리가 0일 때만 발생합니다. 이 파라미터의 천의 자리가 1이면(최대 수정 모니터링 비활성화됨), 제로 벡터도 작성할 수 있습니다.

		41	옵셋 벡터 수정과 관련하여, 현재 값과의 차이가 최소 한 좌표에서 설정 데이터 SD41610 $SN_CORR_TRAFO_LIN_MAX에 의해 지정된 최대 값을 초과합니다. <Corr_Vect> 파라미터는 오류 벡터에 의해 덮어쓰기됩니다. 알람과 함께 프로세싱이 중단될 때도 이것이 적용됩니다(<No_Alarm> 파라미터 참조). 수정 값이 허용 한계를 초과한 컴포넌트의 경우, 지정된 수정 값과 한계 값 간 차이는 수정 표시와 함께 에러 벡터에 있습니다. 한계를 초과하지 않은 컴포넌트의 내용은 0입니다.
		42	방향 벡터 수정과 관련하여 현재 방향 대비 각도 편차가 설정 데이터 SD41611 $SN_CORR_TRAFO_LIN_MAX에 의해 지정된 최대 값을 초과합니다.
		43	시스템 변수를 작성하려는 시도가 작성 권한 없음으로 인해 거부되었습니다.
<Corr_Vect>:	수정 벡터 수정 벡터 내용은 파라미터 <Corr_Index>와 <Corr_Mode>에 의해 결정됩니다. <Corr_Status> = 41이면, 벡터 내용이 덮어쓰기됩니다(상기 참조).		
	데이터 유형:	REAL	
<Corr_Index>:	수정 요소가 수정되는 섹션 / 방향 벡터가 수정되는 오리엔테이션 축의 인덱스		
	데이터 유형:	INT	
	<Corr_Index> 파라미터는 십진수로 코딩됩니다(일의 자리부터 십의 자리까지).		
	일의 자리:	서브체인에서 오리엔테이션 축 또는 섹션의 인덱스를 포함합니다.	
	십의 자리:	서브체인을 참조합니다.	
		0x	공작물 체인
		1x	공구 체인

<Corr_Mode>:	수정 모드		
	데이터 유형:	INT	
	<Corr_Index> 파라미터는 십진수로 코딩됩니다(일의 자리부터 천의 자리까지).		
	일의 자리:	어느 요소가 수정되는지 지정합니다.	
		xxx**0**	선형 옵셋 벡터 수정
		xxx**1**	오리엔테이션 축의 방향 벡터 수정
	십의 자리:	<Corr_Index> 내용이 참조하는 수정 요소의 수정 방식을 지정합니다.	
		xx**0**x	수정 벡터는 수정 요소에 즉시 작성됩니다. 이 방식은 해당 시스템 데이터 ($NK_OFF_DIR[<n>, ...])의 인덱스 <n>을 파악할 필요 없이 즉시 수정 요소를 작성하는 데 사용될 수 있습니다.
		xx**1**x	0일 경우, 전송된 수정값이 월드 좌표계에서 해석된다는 것이 차이입니다. 초기 상태의 키네마틱 체인(모든 오리엔테이션 축의 위치가 0)이 다른 회전을 포함하는 경우, 값 0과 1 사이의 차이가 항상 발생할 수 있습니다.
		xx**2**x	1일 경우, 수정값이 전체 섹션을 참조한다는 것이 차이입니다(예: 전체 섹션이 수정값에 의해 지정된 길이에 도달하도록 수정 요소에 값이 입력됩니다).
		주의: 값 1과 2는 오리엔테이션 축의 방향 벡터 작성 시 허용되지 않습니다.	
	백의 자리:	<Corr_Vect> 파라미터의 내용 해석 방식을 지정합니다.	
		x**0**xx	전송된 수정 벡터 <Corr_Vect>는 수정 요소의 전체 새 길이 또는 <Corr_Index>가 <Corr_Mode>의 십의 자리와 함께 참조하는 섹션을 포함합니다(절대 수정).
		x**1**xx	전송된 수정 벡터 <Corr_Vect>은 수정 요소의 현재 길이 대비 편차 또는 <Corr_Index>가 <Corr_Mode>의 십의 자리수와 함께 참조한 섹션만 포함합니다(증분 수정).

		주의: 오리엔테이션 축의 방향 벡터 수정의 경우, 백의 자리수는 0이어야 합니다.	
	천의 자리:	다음 최대값에 의한 수정 제한 여부를 지정합니다. • SD41610 $SN_CORR_TRAFO_LIN_MAX 또는 • SD41611 $SN_CORR_TRAFO_DIR_MAX	
		0xxx	최대 수정 모니터링이 활성 상태입니다.
		1xxx	최대 수정 모니터링이 비활성 상태입니다.
<No_Alarm>:	에러 발생 시 동작(반환값 > 0) (**옵션 사항**)		
	데이터 유형:	BOOL	
	값:	FALSE (기본 설정)	에러가 발생하면 프로그램 처리가 중단되고 알람 14103이 표시됩니다.
		TRUE	에러가 발생해도 프로그램 처리가 중단되지 않고 알람도 표시되지 않습니다. 적용: 반환값에 따른 사용자별 반응 지정

참고

기능 호출 시 에러가 발생하면 알람이 출력되거나 오류 번호가 반환되어(<No_Alarm> 파라미터 참조) 사용자가 에러 상태에 적절하게 대응할 수 있습니다. 에러의 원인은 알람 파라미터를 통해 자세히 설명됩니다. 알람 대신 반환되는 에러 번호는 알람 파라미터와 동일합니다.

CORRTRAFO 관련 추가 정보

오리엔테이션 변환이 있는 기계의 키네마틱 구조는 월드 좌표계의 영점에서 시작해 1개 또는 2개의 키네마틱 체인(서브체인)에 의해 설명됩니다. 2개의 체인 중 1개는 **공구 체인** 인데 공구의 기준점에서 끝나고, 나머지 체인인 **공작물 체인** 은 기본 좌표계의 영점에서 끝납니다.

CORRTRAFO 기능은 오리엔테이션 변환이 있는 기계의 축 방향과 레버 암 길이를 특수 수정 요소에 작성합니다. 키네마틱 체인은 $ NK_TYPE을 통해 정의되는 OFFSET 타입의 요소 등으로 설명됩니다.

섹션을 통한 CORRTRAFO 작업

2개의 서브체인은 최대 4개 섹션으로 나뉠 수 있습니다.

- 섹션 1은 체인의 시작점에서 시작하여 첫 번째 오리엔테이션 축에서 끝납니다.
- 섹션 2는 오리엔테이션 축 1과 2 사이의 섹션입니다.
- 섹션 3은 오리엔테이션 축 2과 3 사이의 섹션입니다.
- 섹션 4는 오리엔테이션 축 3과 공구나 공작물 체인의 끝 사이 섹션입니다.

각 섹션은 타입 OFFSET 또는 ROT_CONST의 상수 체인 요소를 포함할 수 있습니다.

다음 그림은 2개의 오리엔테이션 축이 있는 오리엔테이션 변환을 보여줍니다.

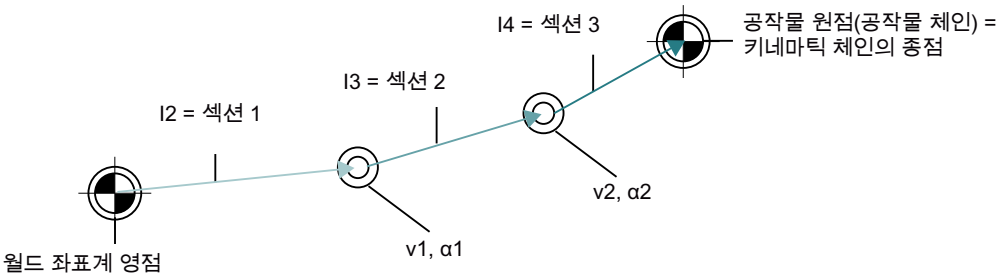


그림 4-8 CORRTRAFO 예

섹션이 명확하게 정의됩니다. 시작점에서 종료점까지 키네마틱 서브체인을 따라 이동하면 첫 번째 섹션의 인덱스는 0이고, 다음 섹션의 인덱스는 1이 되는 등 인덱스 번호가 순서대로 매겨집니다. 마지막 섹션의 인덱스는 항상 오리엔테이션 축의 개수와 같습니다.

수정 요소

$NT_CORR_ELEM_T[<n>, 0 ... 3] 또는 $NT_CORR_ELEM_P[<n>, 0 ... 3] 시스템 변수로 각 섹션에서 상수 키네마틱 체인 요소(유형 $NK_TYPE[<n>] = "OFFSET" 체인 요소)를 참조할 수 있습니다. 기계 측정 동안 파악된 수정값이 CORRTRAFO 기능을 통해 이 요소들에 작성됩니다.

변환 인덱스 = 1 예:

- $NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"; 섹션 1에서는 C축(오리엔테이션 축 1)의 옵셋이 수정 요소로 지정됩니다.
- $NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"; 섹션 2에서는 B축(오리엔테이션 축 2)의 옵셋이 수정 요소로 지정됩니다.
- $NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"; 섹션 3에서는 공구 기준점으로부터 B축의 옵셋이 수정 요소로 지정됩니다.

$NT_CORR_ELEM_T/P[<n>, 0 ... 3]에 있는 참조 시퀀스는 상기 설명된 섹션에 대응되어야 합니다(예: 1개의 체인 요소만 첫 번째 오리엔테이션 축 전에 있는 $NT_CORR_ELEM_T/P [<n>, 0]에 있을 수 있음).

CORRTRAFO 기능은 이러한 방식으로 기계 측정에 의해 파악된 값을 수정 요소에 작성합니다. 수정값 변경은 <Corr_Mode> 파라미터를 통해 CORRTRAFO에 정의됩니다.

체인 닫기

비트 7 또는 비트 8이 $NT_CNTRL[<n>] 시스템 변수에 설정된 경우, 체인의 종단점에서 기계의 영점까지 연결하는 추가 상수 체인 요소가 자동으로 공작물 체인 끝(비트 7) 또는 공구 체인의 시작점(비트 8) 전에 내부적으로 삽입됩니다("체인 닫기").

이렇게 자동 삽입된 요소는 외부에서 작성할 수 없고 읽기만 가능합니다 ($AC_TRAFO_CORR_ELEM_P/T 시스템 변수 참조).

공구 체인을 닫는 지점

$NT_CLOSE_CHAIN_T 시스템 변수가 비어 있지 않으면 공구 체인이 체인의 종단점에서 닫히지 않고 지정된 체인 요소의 종단점에서 닫힙니다. 이 지점 뒤에 있는 다른 체인 요소들은 변환이 활성화될 때 해당 워크 옵셋을 가져옵니다.

오리엔테이션 축의 인덱스

오리엔테이션 축들 간의 상수 옵셋 외에 오리엔테이션 축의 방향 벡터도 CORRTRAFO 기능으로 작성할 수 있습니다. 오리엔테이션 축의 인덱스는 키네마틱 서브체인을 따라 원점에서 종단점까지 이동할 때 발생하는 인덱스로 0부터 시작합니다. 따라서 오리엔테이션 축의 인덱스는 항상 앞 섹션의 인덱스와 동일합니다.

오리엔테이션 축의 인덱스는 $AC_TRAFO_ORIAX_LOC 시스템 변수로도 지정할 수 있습니다.

체인 요소의 최대 변경 허용치

체인 요소의 최대 변경 허용치는 2개의 설정 데이터로 제한할 수 있는데, 오리엔테이션 축의 옵셋 벡터에 대한 SD41610 $SN_CORR_TRAFO_LIN_MAX와 방향 벡터에 대한 SD41611 $SN_CORR_TRAFO_DIR_MAX입니다. SD41610 $SN_CORR_TRAFO_LIN_MAX는 각 개별 벡터 컴포넌트가 해당 기준값과 관련하여 변경될 수 있는 최대치를 지정합니다. SD41611 $SN_CORR_TRAFO_DIR_MAX는 축 벡터의 방향이 해당 기준값과 관련하여 변경될 수 있는 최대 각도를 지정합니다. 기준값은 항상 CORRTRAFO 호출 시 활성화되어 있는 변환에서 활성 상태인 해당 값입니다. 이것은 변경된 키네마틱 데이터 내용이 변환 활성화 후 CORRTRAFO 기능의 작동 방법에 아무 영향도 미치지 않을 수 있다는 것을 의미합니다.

4.13 공구 옵셋

4.13.1 옵셋 메모리

옵셋 메모리 구조

모든 데이터 필드는 T와 D 번호로 호출할 수 있고 공구의 기하학적 사양 뿐 아니라 공구 유형과 같은 추가 항목도 포함합니다.

사용자 절삭 에지 데이터

사용자 절삭 에지 데이터는 기계 데이터를 통해 구성할 수 있습니다. 장비 제조업체의 설명서를 참조하십시오.

공구 파라미터

참고

옵셋 메모리의 개별 값

옵셋 메모리 P1 - P25의 개별 값은 시스템 변수를 통해 프로그램에서 읽고 쓸 수 있습니다. 다른 모든 파라미터는 보관됩니다.

공구 파라미터 $TC_DP15 ~ $TC_DP17, $TC_DP19 및 $TC_DP20 뿐만 아니라 $TC_DP6 ~ $TC_DP8, $TC_DP10 및 $TC_DP11에는 공구 유형에 따라 다른 의미가 있습니다.

공구 파라미터 번호(DP)	시스템 변수 의미	참고
$TC_DP1	공구 종류	개요에 대해서는 목록 참조
$TC_DP2	절삭날 위치	선삭 공구에만 해당
기하	**길이 보정**	
$TC_DP3	길이 1	다음에 할당
$TC_DP4	길이 2	유형 및 레벨
$TC_DP5	길이 3	
기하	**반경**	
$TC_DP6 [1)]	반경 1/길이 1	밀링/선삭/연삭 공구
$TC_DP6 [2)]	직경 d	슬롯 톱
$TC_DP7 [1)]	길이 2/모서리 반경, 테이퍼 밀링 공구	밀링 공구
$TC_DP7 [2)]	슬롯 너비 b 모서리 반경	슬롯 톱

공구 파라미터 번호(DP)	시스템 변수 의미	참고
$TC_DP8 [1] $TC_DP8 [2]	밀링 공구의 라운딩 반경 1 돌출 길이 k	밀링 공구 슬롯 톱
$TC_DP9 [1] [3]	라운딩 반경 2	예비
$TC_DP10 [1]	각도 1 공구의 페이스 끝	테이퍼 밀링 공구
$TC_DP11 [1]	각도 2 공구 수직 방향 축	테이퍼 밀링 공구
마모량	**길이 및 반경 보정**	
$TC_DP12	길이 1	
$TC_DP13	길이 2	
$TC_DP14	길이 3	
$TC_DP15 [1] $TC_DP15 [2]	반경 1 / 길이 1 직경 d	밀링/선삭/연삭 공구 슬롯 톱
$TC_DP16 [1] $TC_DP16 [3]	길이 2/모서리 반경, 테이퍼 밀링 공구, 슬롯 너비 b 모서리 반경	밀링 공구 슬롯 톱
$TC_DP17 [1] $TC_DP17 [2]	밀링 공구의 라운딩 반경 1 돌출 길이 k	밀링/3D 페이스 밀링 슬롯 톱
$TC_DP18 [1] [3]	라운딩 반경 2	예비
$TC_DP19[1]	각도 1 공구의 페이스 끝	테이퍼 밀링 공구
$TC_DP20[1]	각도 2 공구 수직 방향 축	테이퍼 밀링 공구
공구 베이스 치수/어댑터	**길이 옵셋**	
$TC_DP21	길이 1	
$TC_DP22	길이 2	
$TC_DP23	길이 3	
기술		
$TC_DP24	공차 각도	선삭 공구에만 해당
$TC_DP25		예비

[1] 3D 페이스 밀링용 밀링 공구에도 적용

[2] 슬롯 톱 공구 종류에 해당

[3] 예비

추가 설명

여러 입력 구성 요소는 형상 변수에 대해 사용 가능합니다(예를 들어 길이 1 또는 반경). 이 콤포넌트들을 더하여 값 (예: 총 길이 1, 총 반경) 을 구하고 이 값을 다시 계산에 사용합니다.

필요 없는 옵셋 값은 0으로 지정되어야 합니다.

형상 공구의 공구 파라미터 $TC-DP1 ~ $TC-DP23

참고

$TC_DP7과 같이 테이블에 나열되지 않은 공구 파라미터는 평가되지 않습니다. 예를 들어 내용이 의미가 없습니다.

공구 파라미터 번호 (DP)	의미	절삭 Dn		참고
$TC_DP1	공구 종류			400~599
$TC_DP2	절삭날 위치			
기하	**길이 보정**			
$TC_DP3	길이 1			
$TC_DP4	길이 2			
$TC_DP5	길이 3			
기하	**반경**			
$TC_DP6	반경			
기하	**제한 각도**			
$TC_DP10	각도 최소값			
$TC_DP11	각도 최대값			
마모량	**길이 및 반경 보정**			
$TC_DP12	마모 길이 1			
$TC_DP13	마모 길이 2			
$TC_DP14	마모 길이 3			
$TC_DP15	마모 반경			
마모량	**제한 각도**			
$TC_DP19	마모 최소 제한 각도			
$TC_DP20	마모 최대 제한 각도			
공구 베이스 치수/어댑터	**길이 옵셋**			
$TC_DP21	길이 1			

공구 파라미터 번호 (DP)	의미	절삭 Dn		참고
$TC_DP22	길이 2			
$TC_DP23	길이 3			

기본값 및 마모 값

결과 값은 기본값 및 마모 값의 합계입니다(예를 들어 반경의 경우 $TC_DP6 + $TC_DP15). 기본 측정값($TC_DP21 – $TC_DP23)도 첫 번째 절삭 에지의 공구 길이에 추가됩니다. 표준 공구의 유효 공구 길이에 영향을 줄 수 있는 다른 모든 파라미터도 이 공구 길이에 영향을 줍니다(어댑터, 방향 공구 홀더, 설정 데이터).

제한 각도 1 및 2

제한 각도 1 및 2 각각은 절삭 에지 원점에 대한 절삭 에지 중심점의 벡터를 나타내며 시계 방향으로 카운트됩니다.

4.13.2 추가 옵셋

4.13.2.1 추가 옵셋 선택(DL)

추가 옵셋은 가공 과정에서 프로그램될 수 있는 공정 옵셋으로 볼 수도 있습니다. 이 옵셋은 절삭날의 기하학적 데이터를 참조하기 때문에 공구 절삭 데이터의 요소라고 할 수 있습니다.

추가 옵셋의 데이터는 DL 번호 (DL: 위치 종속, 사용 위치 기준 옵셋) 를 사용하여 지정하고 사용자 인터페이스를 통해 입력합니다.

적용

치수 오류가 발생하면 추가 옵셋을 사용하여 사용 위치를 보정해야 하는 경우가 발생합니다.

구문

```
DL=<number>
```

의미

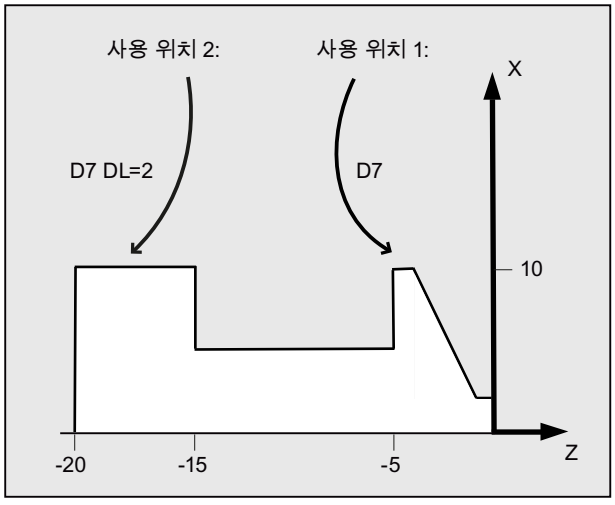
DL:	추가 옵셋 활성화 명령
<number>:	활성화할 추가 공구 옵셋 데이터는 <number> 파라미터를 사용하여 지정합니다.

참고

머신 데이터는 추가 옵셋의 번호를 정의하고 이를 활성화하는 데 이용됩니다(→ 기계 OEM의 데이터를 주의 깊게 관찰하십시오!).

예

2개의 베어링 시트에 동일한 절삭날 사용:



프로그램 코드	코멘트
N110 T7 D7	; 터렛을 위치 7에 포지셔닝합니다. D7과 DL=1은 다음 블록에서 활성화하고 이동됩니다.
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; D7 외에 DL=2가 활성화되며 다음 블록에서 이동합니다.
N150 G1 Z-21	
N160 G0 X200 Z200	; 공구 교환 위치에 접근합니다.
...	

4.13.2.2 마모값 및 설정값을 지정($TC_SCPxy[t,d], $TC_ECPxy[t,d])

4.13.2.3 추가 옵셋 삭제(DELDL)

DELDL 명령은 공구의 절삭날에 대한 추가 옵셋을 삭제합니다(메모리 공간 확보용). 정의된 마모값 및 설정값 모두가 삭제됩니다.

구문

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<Status>=DELDL[<t>,<d>]
```

의미

DELDL:	추가 옵셋 삭제 명령	
<t>:	공구의 T 번호	
<d>:	공구 절삭날의 D 번호	
DELDL[<t>,<d>]:	공구 <t> 절삭날 <d>의 추가 옵셋이 모두 삭제됩니다.	
DELDL[<t>]:	공구 <t> 절삭날의 추가 옵셋이 모두 삭제됩니다.	
DELDL:	TO 장치의 모든 공구 절삭날의 옵셋이 모두 삭제됩니다(명령이 프로그램되는 채널).	
<상태>:	삭제 상태	
	값:	의미:
	0	삭제가 성공적으로 완료되었습니다.
	-	옵셋이 아직 삭제되지 않았거나(파라미터 설정이 정확하게 단 하나의 공구 날을 지정하고 있는 경우) 또는 완전히 삭제되지 않았습니다(매개 변수 설정이 몇 개의 절삭날을 지정하고 있는 경우).

참고

활성화된 공구의 마모값 및 설정값은 삭제할 수 없습니다(D 또는 공구 데이터의 삭제 특성과 기본적으로 동일함).

4.13.3 공구 옵셋 특수 처리

공구 길이 및 마모에 대한 기호의 평가는 설정 데이터 SD42900에서 SD42960를 사용하여 제어할 수 있습니다.

이 사항은 기하축을 미러링하거나 가공 평면을 변경할 때 마모 요소의 특성 및 공구 방향으로 온도 보정을 할 때도 동일하게 적용됩니다.

마모값:

다음에서 마모값에 참조가 만들어진 경우, 이 참조는 실제 마모값의 합계($TC_DP12 to $TC_DP20) 및 마모값($SCPX3 to $SCPX11)과 설정값($ECPX3 to $ECPX11)을 사용한 합계 옵셋으로 받아들여야 합니다.

총계 옵셋에 대해 **알아보려면** 기능 매뉴얼 공구 관리를 참고하십시오.

설정 데이터

SD42900 $SC_MIRROR_TOOL_LENGTH	공구 길이 요소의 미러링 및 공구 베이스 치수의 요소
SD42910 $SC_MIRROR_TOOL_WEAR	공구 길이 요소의 마모값 미러링
SD42920 $SC_WEAR_SIGN_CUTPOS	마모 콤포넌트의 부호를 절삭날 위치 기능으로 평가.
SD42930 $SC_WEAR_SIGN	마모 치수의 기호를 전환하십시오.
SD42935 $SC_WEAR_TRANSFORM	마모값의 변환
SD42940 $SC_TOOL_LENGTH_CONST	공구 길이 요소의 기하축 할당
SD42950 $SC_TOOL_LENGTH_TYPE	종류와 상관없이 공구 길이 요소 할당
SD42960 $SC_TOOL_TEMP_COMP	공구 방향으로 온도 보정값. 또한 공구 방향이 프로그램된 경우에 효과적입니다.

추가 정보

수정된 셋팅 데이터의 활성화

위에서 설명한 셋팅 데이터가 수정된 경우 다음에 절삭날이 선택될 때까지 공구 콤포넌트를 다시 평가하지 않습니다. 이미 공구가 활성인 상태에서 이 공구의 데이터를 재평가하려면 공구를 다시 선택해야 합니다.

결과 공구 길이가 축의 미러링 상태 변경으로 인해 수정된 경우에도 동일한 사항이 적용됩니다. 미러링 명령을 사용한 이후에는 수정된 공구 길이 요소를 활성화시키기 위해 공구를 다시 선택해야만 합니다.

전향적 공구 홀더 및 새로운 세팅 데이터

셋팅 데이터 SD42900~SD42940은 오리엔테이션 기능이 있는 활성 공구 홀더의 콤포넌트에는 영향을 미치지 않습니다. 하지만 전향적 공구 홀더를 계산할 때는 항상 총 결과 길이(공구 길이 + 마모 + 공구 베이스 수치)를 고려합니다. 셋팅 데이터에 의해 수정된 모든 내용은 총

길이 계산에 포함됩니다. 즉, 오리엔테이션 가능한 공구 홀더의 벡터는 가공 평면과 관계가 없습니다.

참고

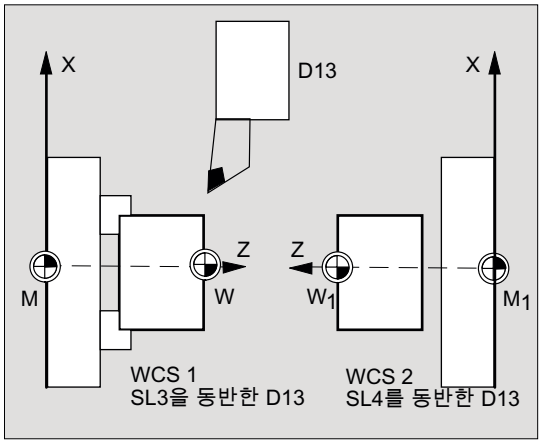
전향적 공구 홀더가 사용되었을 때, 비 미러 기본 시스템에 사용하는 모든 공구를 정의하는 것이 실용적일 때가 많습니다. 미러링 된 가공에만 사용될 때도 마찬가지입니다. 미러링된 축을 사용하여 가공할 때, 공구의 실제 위치가 정확하게 설명되어 공구 홀더가 그만큼 회전합니다. 그러면 모든 공구 길이 요소가 정확한 방향으로 자동으로 작동합니다. 이 때 개별 축의 미러링 상태에 따른 셋팅 데이터를 통해 개별 요소 평가를 제어하지 않아도 됩니다.

그 밖의 이용 옵션

전향적 공구 홀더 기능을 활용하면 기계의 선삭 공구의 물리적 옵션이 없는 경우에도 유용합니다. 아무리 다른 방향의 공구가 영구적으로 설치되어 있는 경우라 하더라도 마찬가지입니다. 공구 치수는 기본 오리엔테이션에서 획일적으로 수행될 수 있습니다. 이 때 가공에 대한 치수 관련성은 실제 공구 홀더의 회전에 따라 계산됩니다.

4.13.3.1 공구 길이 미러링

셋팅 데이터인 SD42900 $SC_MIRROR_TOOL_LENGTH 및
SD42910 $SC_MIRROR_TOOL_WEAR가 0으로 설정되어 있지 않은 경우, 공구 길이 요소 및 마모값 및 해당 축을 사용한 기본 치수의 요소를 미러링할 수 있습니다.



SD42900 $SC_MIRROR_TOOL_LENGTH

셋팅 데이터가 **0이 아님**:

공구 길이 요소($TC_DP3, $TC_DP4 and $TC_DP5) 및 기본 치수 요소($TC_DP21, $TC_DP22 and $TC_DP23)는 해당 축에 대해 미러링되지 않고, 기호 전환에 의해 미러링됩니다.

마모값이 미러링되지 **않습니다**. 마모 값도 미러링하려면 셋팅 데이터 SD42910 $SC_MIRROR_TOOL_WEAR를 설정해야 합니다.

SD42910 $SC_MIRROR_TOOL_WEAR

셋팅 데이터가 **0이 아님**:

공구 길이 요소의 마모값 - 해당 축은 미러링 됨 - 역시 기호 전환에 의해 미러링됩니다.

4.13.3.2 마모 기호 평가

셋팅 데이터 SD42920 $SC_WEAR_SIGN_CUTPOS and SD42930 $SC_WEAR_SIGN이 0으로 설정되지 않는 경우, 마모 요소의 기호 평가를 전환할 수 있습니다.

SD42920 $SC_WEAR_SIGN_CUTPOS

셋팅 데이터가 **0이 아님**:

절삭날 위치를 사용하는 공구의 경우 (선삭 및 연삭 공구, 공구 종류 400), 가공 평면에서 마모 콤포넌트의 부호 평가는 절삭날 위치에 따라 달라집니다. 이 셋팅 데이터는 절삭날 위치와 관련이 없는 공구 종류에는 아무 의미가 없습니다.

다음 표에서, SD42920(0이 아님)을 사용하여 전환된 기호를 사용하는 치수는 X를 사용하여 고안되었습니다.

절삭날 위치	길이 1	길이 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

참고

SD42920 및 SD42910을 사용한 기호 평가는 서로 연관성이 없습니다. 예를 들어 두 셋팅 데이터를 사용하여 치수의 부호를 변경해도 그 결과 부호는 변경되지 않습니다.

SD42930 $SC_WEAR_SIGN

셋팅 데이터가 0이 아님:

마모 치수의 기호를 모두 전환하십시오. 이는 공구 길이 및 공구 반경, 라운드 반경 등과 같은 기타 변수 모두에 영향을 줍니다.

공구 치수를 양수로 입력한 경우, 공구는 "짧고" "가늘어"집니다. "공구 옵셋, 특정 처리", "변경된 셋팅 데이터 활성화" 장을 참조하십시오.

4.13.3.3 활성화된 가공 작업의 좌표계(TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

기계의 구조 또는 오리엔테이션 가능한 공구 캐리어의 가용 여부에 따라, 좌표계 중 하나에서 측정된 마모값이 적절한 좌표계로 변환 또는 전환됩니다.

활성화된 가공 작업의 좌표계

다음 좌표계는 그룹 56의 해당 G 코드를 통해 공구 길이 마모 구성요소를 활성 공구에 통합시키는 공구 길이 옵셋을 생성합니다.

- 기계 좌표계(MCS)
- 기본 좌표계(BCS)
- 공작물 좌표계 (WCS)
- 공구 좌표계(Tool coordinate system: TCS)
- 구조 전환의 공구 좌표계(Tool coordinate system of kinematic transformation: KCS)

구문

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

의미

TOWSTD:	공구 길이 마모값에서 옵셋에 대한 설정값 시작하기
TOWMCS:	MCS에서 공구 길이의 옵셋
TOWWCS:	WCS에서 공구 길이의 옵셋
TOWBCS:	BCS에서 공구 길이의 옵셋

TOWTCS:	공구 캐리어 원점에서 공구 길이 옵셋(오리엔테이션 가능한 공구 캐리어)
TOWKCS:	공구 헤드의 공구 길이 보정(구조 전환)

추가 정보

특징적 기능들

가장 중요한 특징적 기능들은 다음 표에 나와 있습니다.

G 코드	마모값	활성화된 오리엔테이션 가능한 공구 캐리어
TOWSTD	초기값, 공구 길이	마모값은 회전 대상입니다.
TOWMCS	MCS에서 마모값. 오리엔테이션 가능한 공구 캐리어가 활성화되지 않은 경우, TOWMCS 는 TOWSTD 와 동일합니다.	마모는 상관 없이 결과 공구 길이의 벡터만을 회전시킵니다.
TOWWCS	마모값이 WCS에서 MCS로 변환되었습니다.	공구 벡터는 마모는 고려하지 않고 TOWMCS 에 대해서만 계산되었습니다.
TOWBCS	마모값이 BCS에서 MCS로 변환되었습니다.	공구 벡터는 마모는 고려하지 않고 TOWMCS 에 대해서만 계산되었습니다.
TOWTCS	마모값은 공작물 좌표계에서 MCS로 변환되었습니다.	공구 벡터는 마모는 고려하지 않고 TOWMCS 에 대해서만 계산되었습니다.

TOWWCS , TOWBCS, TOWTCS: 공구 벡터에 마모 벡터가 추가되었습니다.

선형 변환

MCS가 BSC에서 선형 변환에 의해 생성된 경우에만 MCS에서 공구 길이를 의미 있게 정의할 수 있습니다.

비선형 변환

예를 들어, TRANSMIT을 사용하는 비선형 변환이 활성화된 경우, 좌표계의 요청대로 MCS가 지정되었다면, BCS는 자동적으로 사용됩니다.

구조 전환 및 오리엔테이션 가능한 공구 캐리어 없음

구조 전환이나 오리엔테이션 가능한 공구 캐리어가 활성화되어있지 않은 경우, 나머지 4개의 좌표계가 모두(WCS에 대한 좌표계 제외) 결합됩니다. 그렇게 되면 WCS만 다른 좌표계와 달라집니다. 공구 길이만 평가되면 되기 때문에 좌표계간의 해석은 무의미합니다.

계산에 마모값 포함

설정 데이터 **SD42935 $SC_WEAR_TRANSFORM** 는 3개의 마모 요소를 정의합니다.

- 마모량
- 미세조정 옵셋 합계
- 일반 옵셋 합계

위 항목들은 다음의 G 코드 중 하나가 활성화되어 있는 경우, 어댑터 변환을 사용하거나 오리엔테이션 가능한 공구 캐리어를 사용하여 회전됩니다.

- TOWSTD
  기본 위치. 공구 길이 보정용
- TOWMCS
  기계 좌표계(MCS) 에서 마모 값
- TOWWCS
  공작물 좌표계(WCS)의 마모값
- TOWBCS
  기본 좌표계(BCS)의 마모값
- TOWTCS
  공구 캐리어 고정물에서 공구 좌표계에서의 마모값 (T 공구 캐리어 기준)
- TOWKCS
  구조 전환에 대한 공구 헤드의 좌표계에서 마모값

참고

개별 마모 구성요소 평가(기하축, 기호 평가에 할당)는 다음 요인의 영향을 받습니다.

- 활성 평면
- 어댑터 변환
- 설정 데이터:
  - SD42910 $SC_MIRROR_TOOL_WEAR
  - SD42920 $SC_WEAR_SIGN_CUTPOS
  - SD42930 $SC_WEAR_SIGN
  - SD42940 $SC_TOOL_LENGTH_CONST
  - SD42950 $SC_TOOL_LENGTH_TYPE

4.13.3.4 공구 길이 및 평면 변경

셋팅 데이터 SD42940 $SC_TOOL_LENGTH_CONST가 0으로 설정되지 않은 경우, 평면을 변경할 때 연삭 및 선삭 공구에 대해 공구 길이 요소 - 길이, 마모 및 기본 치수 등 -를 기하축에 할당할 수 있습니다.

SD42940 $SC_TOOL_LENGTH_CONST

셋팅 데이터가 0이 아님:

공구 길이 요소(길이, 마모 및 공구 기본 치수)에 기하축을 할당하면 가공 평면이 변경되어도 할당이 변경되지 않습니다(G17 - G19).

다음 표는 연삭 및 선삭 공구(공구 종류 400 - 599)에 대한 기하축에 공구 길이 요소를 할당하는 것을 보여줍니다.

내용	길이 1	길이 2	길이 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) 나열된 6개 값 중 하나와 동일하지 않은 0이 아닌 각 값은 18의 값으로 평가됩니다.

다음 표는 다른 모든 공구(공구 종류 400 초과 또는 599 미만)에 대한 기하축에 공구 길이 요소를 할당하는 것을 보여줍니다.

작업 평면	길이 1	길이 2	길이 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) 나열된 6개 값 중 하나와 동일하지 않은 0이 아닌 각 값은 17의 값으로 평가됩니다.

참고

표에 나온 설명에서는 기하 축 1 ~ 3의 이름을 X, Y, Z 축으로 간주합니다. 하지만 축 식별자가 아닌 축 순서가 축에 보정을 할당하는 데 있어 결정적인 요소입니다.

4.13.4 온라인 공구 옵셋

4.13.4.1 폴리노미널 기능 정의 (FCTDEF)

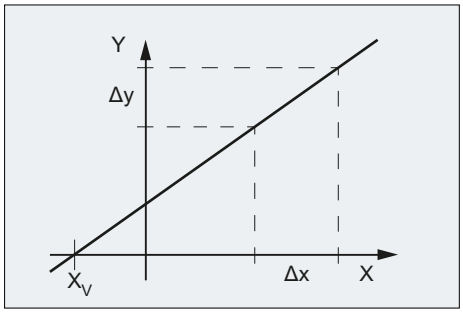
특정 드레싱 전략 (예: 드레싱 롤러)는 드레싱 롤러를 투입하는 것과 마찬가지로 연삭 휠 반경이 지속적으로 (선형으로) 감소된다는 사실에 의해 규정됩니다. 이 전략은 드레싱 롤러의 절입 그리고 각 길이의 마모 값을 기록하는 것 사이의 선형 함수를 필요로 합니다. 선형 함수는 3차 폴리노미널 함수까지를 위한 사전 정의된 절차FCTDEF(...)를 사용하여 정의됩니다.

직선 방정식

$y = f(x) = a_0 + a_1 * x_1$

a_1: $a_1 = \Delta x / \Delta y$인 경우 직선의 경사도

a_0: $a_0 = -a1 * X_V$인 경우 X축 상의 직선 이동



구문

```
FCTDEF(<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
```

의미

FCTDEF(...):	PUTFTOCF(...)를 위한 폴리노미널 함수 정의: $y = f(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3$	
<Func>:	함수 수	
	데이터 유형:	INT
	값 범위:	1, 2, 3
<LLimit>:	하한값	
	데이터 유형:	REAL
<ULimit>:	상한값	
	데이터 유형:	REAL

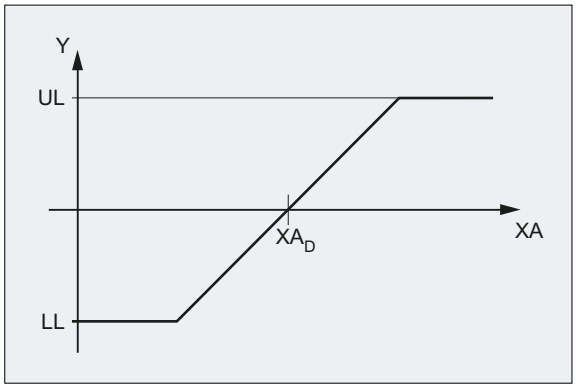
<a0>,<a1>,<a2>,<a3>:	폴리노미얼 기능 계수	
	데이터 유형:	REAL

예

정의

- 함수 번호: 1
- 하한 및 상한 제한값: -100, 100
- 특성 경사도: $a_1 = 1$
- 조작점은 반드시 특성의 중심에 위치해야 합니다. 함수가 NC 프로그램 상에 정의되는 순간에 WCS 상의 축 XA의 설정점 위치에 기반하여 특성은 반드시 음의 Y 방향으로 이동해야 합니다: $a_0 = -a_1 * XA_D = -1 * \$AA_IW$
- $a_2 = a_3 = 0$

특성



ul 상한값

LL 하한값

XA_D 함수가 NC 프로그램에 정의되는 순간의 축 XA 설정점

프로그래밍

프로그램 코드	주석
FCTDEF(1,-100,100,-$AA_IW[XA],1)	; 기능 정의

4.13.4.2 온라인 공구 옵셋 연속 작성 (PUTFTOCF)

FCTDEF(...) (페이지 841)를 통해 이전에 정의된 폴리노미널 함수에 기초하여 사전 정의된 절차 PUTFTOCF(...) 를 사용해 온라인 공구 옵셋이 실행됩니다.

참고

온라인 공구 옵셋은 동기 동작을 사용하여 실현될 수도 있습니다.
좀 더 상세한 정보를 보려면 "기능 매뉴얼 동기 동작"을 참조하십시오.

구문

```
PUTFTOCF(<Func>,<RefVal>,<ToolPar>,<Chan>,<Sp>)
```

의미

PUTFTOCF(...):	FCTDEF(...)를 통해 정의된 폴리노미널 함수를 사용하여 블록별로 연속해서 온라인 공구 옵셋을 작성합니다.	
<Func>:	FCTDEF(...)를 통한 함수 정의에 정의된 함수 번호	
	데이터 유형:	INT
	값 범위:	1, 2, 3
<RefVal>:	옵셋이 도출될 기준 값 (예: 축의 설정점)	
	데이터 유형:	VAR REAL
<ToolPar>:	옵셋 값이 포함될 마모 파라미터 (길이 1, 2 또는 3)의 번호	
	데이터 유형:	INT
<Chan>:	온라인 공구 옵셋이 적용될 채널 번호 **참고:** 옵셋이 활성 채널에서 효과를 미치지 못하는 경우에만 필요합니다.	
	데이터 유형:	INT
<Sp>:	온라인 공구 옵셋이 적용될 스핀들 번호 **참고:** 옵셋이 현재 사용되고 있는 활성 공구보다는 비활성 연삭 휠에 대해 적용되는 경우 필요합니다.	
	데이터 유형:	INT

4.13.4.3 온라인 공구 옵셋 작성, 불연속 (PUTFTOC)

기능

사전 정의된 절차PUTFTOC(...)를 사용해서 고정된 옵셋 값에 기초하여 온라인 공구 옵셋이 실행됩니다.

구문

```
PUTFTOC(<CorrVal>,<ToolPar>,<Chan>,<Sp>)
```

의미

PUTFTOC(...):	온라인 공구 옵셋 작성	
<CorrVal>:	마모 파라미터에 추가된 옵셋 값.	
	데이터 유형:	REAL
<ToolPar>:	옵셋 값이 포함될 마모 파라미터 (길이 1, 2 또는 3)의 번호	
	데이터 유형:	INT
<Chan>:	온라인 공구 옵셋이 적용될 채널 번호 **참고:** 옵셋이 활성 채널에서 효과를 미치지 못하는 경우에만 필요합니다.	
	데이터 유형:	INT
<Sp>:	온라인 공구 옵셋이 적용될 스핀들 번호 **참고:** 옵셋이 현재 사용되고 있는 활성 공구보다는 비활성 연삭 휠에 대해 적용되는 경우 필요합니다.	
	데이터 유형:	INT

4.13.4.4 온라인 공구 옵셋 활성화/비활성화 (FTOCON/FTOCOF)

온라인 공구 옵셋은 G 명령어 FTOCON 및 FTOCOF를 사용하여 활성화되거나 비활성화됩니다.

구문

```
FTOCON
```

```
...
FTOCOF
```

의미

FTOCON:	온라인 공구 옵셋 활성화 명령어는 반드시 온라인 공구 옵셋이 활성화되는 채널에 프로그래밍되어야 합니다.
FTOCOF:	온라인 공구 옵셋 비활성화 명령어는 반드시 온라인 공구 옵셋이 비활성화되는 채널에 프로그래밍되어야 합니다. **참고:** FTOCOF의 경우 축은 공구 옵셋을 위해 더 멀리 움직이지 않습니다. 그러나, PUTFTOC/PUTFTOCF를 사용하여 계산된 값은 절삭 특정 옵셋 데이터로 남습니다. 온라인 공구 옵셋을 최종적으로 비활성화하려면 공구 (T...)가 FTOCOF 이후 반드시 재선택/선택 취소되어야 합니다.

4.13.5 3D 공구 반경 보정

4.13.5.1 3D 원주 밀링을 위한 3D 공구 반경 보정 선택(CUT3DC, CUT3DCD, ISD)

한계 표면을 고려하지 않는 3D 원주 밀링용 3D 공구 반경 보정(3D TRC)은 모달로 적용되는 G 명령 CUT3DC 또는 CUT3DCD를 통해 선택됩니다.

실제 활성화는 G41이나 G42를 통해 수행됩니다. 공구 반경 보정은 G40로 비활성화됩니다.

구문

```
G41/G42 ORIC/ORID ISD=... CUT3DC/CUT3DCD CDOF2 X... Y... Z...
...
G40 X... Y... Z...
```

의미

CUT3DC:	원주 밀링을 위한 3D TRC(5축 변환이 활성인 경우만 해당)	
CUT3DFS	차동 공구에 대한 원주 밀링용 3D TRC(5축 변환이 활성화된 경우만 해당) 반경 차이는 공구 파라미터 $TC_DP15에 의해 지정됩니다.	
G41/G42 X... Y... Z...:	공구 반경 보정 활성화	
	G41:	형상 좌측 공구 반경 보정
	G42:	형상 우측 공구 반경 보정
	주의: 활성화는 반드시 선형 블록(G0/G1)에서 수행해야 합니다.	
CDOF2:	3D 원주 밀링의 충돌 탐지 비활성화	
ORIC/ORID:	바깥쪽 코너에서의 오리엔테이션 변경 동작은 G 명령 ORIC와 ORID를 통해 지정됩니다.	
	ORIC:	외부 코너에서의 오리엔테이션 변경은 삽입될 원호 블록과 겹쳐집니다.
	ORID:	외부 코너에서의 오리엔테이션 변경은 삽입될 원호 블록 전에 실행됩니다.
ISD=<Value>:	ISD주소로 공구의 삽입 깊이가 원주 밀링 및 **활성** 3D 공구 반경 보정에 대해 변경될 수 있습니다.	
	<Value>:	삽입 깊이의 길이
G40 X... Y... Z...:	공구 반경 보정 비활성화 **주의:** 비활성화는 반드시 기하축 이동을 통해 선형 블록(G0/G1)에서 수행해야 합니다.	

참고

3D TRC를 선택하기 위한 G 명령은 접근 블록에서 평가됩니다(예: 일반적으로 G41 또는 G42를 포함하는 블록).

G41 또는 G42는 보정 관련 기하 축에서 이송 동작 없이 블록으로 프로그래밍될 수 있습니다. 이 경우, 접근 블록은 이러한 블록 이후 첫 번째 이송 블록입니다.

활성 공구 반경 보정을 사용하는 3D TRC variant의 변경은 알람 없이 무시됩니다.

예

프로그램 코드	설명
	; 공구 D1 정의:
$TC_DP1[1,1]=120	; 타입(엔드 밀)
$TC_DP3[1,1]=20	; 길이 보정 벡터
$TC_DP6[1,1]=8	, 반경
N10 X0 Y0 Z0 T1 D1 F12000	; 공구 선택
N20 TRAORI(1)	; 변환 활성화.
N30 G42 ORIC ISD=10 **CUT3DC** G64 X30	; 3D 원주 밀링 활성화,
	; 바깥 코너에서 연속 오리엔테이션 변경,
	; 삽입 깊이: 10 mm
N40 ORIWKS A30 B15	; 축 위치 지정을 통한 코너에서의 오리엔테이션 변경.
N50 Y20 A3=1 C3=1	; 오리엔테이션 변경이 있는 이송 블록,
	; 방향 벡터가 있는 오리엔테이션 지정.
N60 X50 Y30	; 일정한 오리엔테이션이 있는 이송 블록.
N70 Y50 A3=0.5 B3=1 C3=5	; 오리엔테이션 변경이 있는 이송 블록.
N80 M63	; 이송 정보가 없는 블록.
N90 X0 ISD=20	; 삽입 깊이 변화가 있는 이송 블록.
N100 G40 Y0	; 공구 반경 보정 비활성화.
N110 M30	

추가 정보

경로 및 오리엔테이션

여기에서 사용하는 원주 밀링의 유형은 경로(가이드 라인)와 해당 오리엔테이션을 정의하여 구현됩니다. 이런 유형의 가공에서, 경로에 있는 공구의 형태는 관련이 없습니다. 공구 개입점의 반경만 결정적인 역할을 합니다.

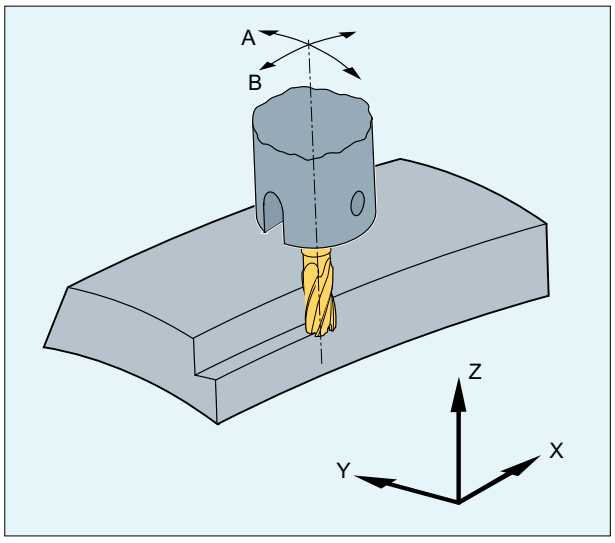


그림 4-9 원주 밀링

접근 동작

접근 동작은 항상 공구 반경 보정의 3D variant에 대해 NORM입니다.

바깥쪽 코너에서의 동작

그룹 18의 G 명령(코너 동작, 공구 옵셋)은 3D TRC를 이용한 원주 밀링을 위해 바깥쪽 코너에서 2½D TRC 조건과 같은 방식으로 평가됩니다.

- G450: 전이 원(공구는 원형 경로에서 둥근 공작물 코너를 이동)
  2½D TRC 솔루션과 달리, 바깥쪽 코너에 삽입된 형상 요소는 항상 반경이 0인 원으로, 여기에서는 공구 반경 보정이 다른 프로그램된 경로에서와 같은 방식으로 동작합니다. 원 대신 원뿔을 삽입할 수 없습니다. 이 경우, DISC 주소는 의미가 없으므로 평가되지 않습니다.
- G451: 등거리 경로의 교차(공구는 공작물 코너에서 물러남)
  교차는 2개의 관련된 블록의 옵셋 곡선의 확장 및 공구 방향에 대한 평면 수직의 코너에 있는 블록 2개의 교차 정의에 의해 결정됩니다.

교차 절차(G451)는 공구 오리엔테이션 변경을 포함하는 하나 이상의 블록이 관련 이송 블록들 사이에 삽입되었을 때 사용되지 않습니다. 이 경우, 원이 항상 코너에 삽입됩니다.

바깥쪽 코너에서의 오리엔테이션 변경 동작

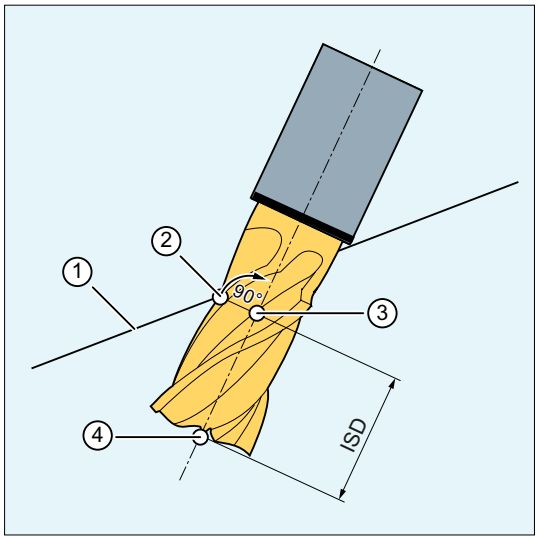
ORIC 및 ORID G 명령은 코너를 이루는 두 블록 사이에 프로그램된 오리엔테이션 변경을 삽입된 원호 블록(ORID)이 처리되기 전에 실행할지 아니면 동시에 실행할지 결정하는 데 사용됩니다.

삽입 깊이

밀링 공구의 삽입 깊이는 공구 팁에서부터 밀링 공구 원점까지의 거리입니다.

밀링 공구 원점은 공구의 세로 축에 대한 프로그램 된 경로 상의 밀링 공구 가공 지점의 수직 투사입니다.

공구 주변 표면 상의 가공 지점의 위치는 삽입 깊이로 설정됩니다.



① 프로그래밍된 경로
② 밀링 공구 가공 지점
③ 밀링 공구 원점
④ 밀링 공구 팁
ISD 삽입 깊이(InSertion Depth)

그림 4-10 삽입 깊이

차등 공구에 대한 공구 반경 보정

차동 공구에 대한 원주 밀링용 3D TRC는 CUT3DCD 명령을 통해 활성화됩니다. 프로그램된 형상이 표준 공구의 중심점 경로를 참조하고 차동 공구 외 다른 공구가 가공에 사용되는 경우 이것이 적용되어야 합니다. 3D 공구 반경 보정을 계산할 때는 활성 공구($TC_DP15) 반경의 마모값과 프로그래밍된 공구 옵셋 OFFN 및 TOFFR/TOFFLR만 고려됩니다. 활성 공구의 기본 반경 ($TC_DP6) 는 고려되지 **않습니다**.

CUT3DC를 사용하는 원주 밀링에 대해 경사면 벽을 사용하는 포켓 밀링

3D 공구 반경 보정에서, 밀링 반경의 편차는 가공하게 될 표면 법선을 향한 절입에 의해 보정됩니다. 밀링 공구 페이스가 위치해 있는 평면은 삽입 깊이 ISD가 동일하게 남아 있는 한 변경되지 않습니다. 예를 들어, 표준 공구 반경보다 작은 밀링 공구는 역시 한계 표면인 포켓 베이스에 도달하지 못할 것입니다. 자동 공구 절입을 하기 위해서는 제어 시스템이 한계 표면을 인지하고 있어야 합니다("한계 표면을 고려한 3D 원주 밀링(CUT3DCC, CUT3DCCD) (페이지 856)" 참조).

Advanced Surface / Top Surface

참고

공구 반지름 보정 CUT3DCD을 "Advanced Surface" 또는 "Top Surface" 옵션(라이선스 필요)과 함께 적용하는 경우, "Advanced Surface" / "Top Surface" 관련 권장 설정을 반드시 준수해야 합니다.

설정된 데이터 확인을 위해 특수 테스트 프로그램이 SIOS 포털에서 제공됩니다.

→ Advanced Surface / Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)의 테스트 프로그램

4.13.5.2 3D 페이스 밀링용 3D 공구 반경 보정 선택(CUT3DF, CUT3DFS, CUT3DFF, CUT3DFD)

3D 페이스 밀링용 3D 공구 반경 보정(3D TRC)은 모달 적용 G 명령 CUT3DF, CUT3DFS, CUT3DFF 또는 CUT3DFD로 선택됩니다.

실제 활성화는 G41이나 G42로 수행됩니다.

3D 페이스 밀링의 경우, 가공될 평면의 표면 법선이 정의되어 있어야 공구 반경 보정을 계산할 수 있습니다. 이것은 A4, B4, C4와 A5, B5, C5 주소를 통해 G41 또는 G42로 블록에서 수행되어야 합니다.

공구 반경 보정은 G40로 비활성화됩니다.

구문

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD X... Y... Z... A4=... B4=...
C4=... A5=... B5=... C5=...
...
G40 X... Y... Z...
```

의미

CUT3DFS:	일정한 오리엔테이션을 사용하는 페이스 밀링을 위한 3D TRC 공구 오리엔테이션은 G17 - G19에 의해 정의되고, 프레임의 영향은 받지 않습니다.
CUT3DFF:	일정한 오리엔테이션을 사용하는 페이스 밀링을 위한 3D TRC 공구 오리엔테이션은 G17 - G19에 의해 정의된 방향이고, 일부 경우 프레임에 의해 회전됩니다.
CUT3DF:	오리엔테이션 변경이 있는 페이스 밀링용 3D TRC(5축 변환만 해당).

CUT3DFD:	오리엔테이션이 변할 때 페이스 밀링용 차동 공구에 대한 3D TRC(활성 5축 변환만 해당) 반경 차이는 공구 파라미터 $TC_DP15에 의해 지정됩니다. **주의**: CUT3DFD는 "Smoothing of surface normals in 3D face milling(3D 페이스 밀링에서 표면 법선의 스무딩)"과 함께만 사용 가능합니다. 이는 CYCLE832(...)를 통해 "Top Surface" 기능(라이선스 필요)을 호출하여 활성화됩니다.	
G41/G42 X... Y... Z...:	공구 옵셋 활성화 G41와 G42를 이용한 동작은 3D 페이스 밀링에 동일합니다. **주의:** 활성화는 반드시 선형 블록(G0/G1)에서 수행해야 합니다.	
A4/5=... B4/5=... C4/5=...:	가공될 평면의 표면 법선 정의	
	A4=... B4=... C4=...:	블록 시작에 정의
	A5=... B5=... C5=...:	블록 끝에 정의
ORIC/ORID:	바깥쪽 코너에서의 오리엔테이션 변경 동작은 G 명령 ORIC와 ORID를 통해 지정됩니다.	
	ORIC:	외부 코너에서의 오리엔테이션 변경은 삽입될 원호 블록과 겹쳐집니다.
	ORID:	오리엔테이션 변경은 원호 블록 이전에 수행됩니다
G40 X... Y... Z...:	공구 반경 보정 비활성화 **주의:** 비활성화는 반드시 기하축 이동을 통해 선형 블록(G0/G1)에서 수행해야 합니다.	

참고

G41 또는 G42는 보정 관련 기하 축에서 이송 동작 없이 블록으로 프로그래밍될 수 있습니다. 이 경우, 접근 블록은 이러한 블록 이후 첫 번째 이송 블록입니다.

활성 공구 반경 보정을 사용하는 3D TRC variant의 변경은 알람 없이 무시됩니다.

예

예 1: CUT3DF를 이용한 3D 페이스 밀링

프로그램 코드	설명
N10	; 공구 D1 정의:
N20 $TC_DP1[1,1]=121	; 공구 타입(환상면 밀러)
N30 $TC_DP3[1,1]=20	; 길이 보정
N40 $TC_DP6[1,1]=5	, 반경
N50 $TC_DP7[1,1]=3	; 라운드 반경
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; 공구 선택
N90 TRAORI(1)	; 오리엔테이션 변환 선택.
N100 B4=-1 C4=1	; 평면 정의.
N110 **G41** ORID **CUT3DF** G64 X10 Y0 Z0	; 공구 옵셋 활성화
N120 X30	
N130 Y20 A4=1 C4=1	; 바깥 코너, 새 평면 정의.
N140 B3=1 C3=5	; ORID로 오리엔테이션 변경
N150 B3=1 C3=1	; ORID로 오리엔테이션 변경
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; ORIC로 오리엔테이션 변경
N180 A3=-1 C3=1	; ORIC로 오리엔테이션 변경
N190 Y-10 A4=-1 C4=3	; 새 평면 정의.
N200 X-20 Y-20 Z10	; 이전 블록의 안쪽 코너
N210 X-30 Y10 A4=1 C4=1	; 안쪽 코너, 새 평면 정의.
N220 A3=1 B3=0.5 C3=1.7	; ORIC로 오리엔테이션 변경
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; 오리엔테이션 변경.
N250 X0 Y30 C4=1	; 경로 이동, 새 평면, ; 상대 프로그래밍으로 오리엔테이션.
N260 BSPLINE X20 Z15	; 스플라인 시작, 오리엔테이션의 상대 프로그래밍
N270 X30 Y25 Z18	; 스플라인 동안 활성 상태 유지.
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; 나선, 축 프로그래밍으로 오리엔테이션.
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; 경로 이동, 오리엔테이션, 비상수 평면.
N330 X-20 G40	; 공구 반경 보정 비활성화.
N340 M30	

예 2: CUT3DFD로 CAD 시스템에서 생성된 NC 프로그램(섹션)

프로그램 코드	설명
N01 G710	

프로그램 코드	설명
N03 T="12"	
N06 S5305 M03	
N07 G642	
; 공구 길이를 고려하여 MCS에서의 시작 위치에 접근.	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; MCS에서의 포지셔닝 종료.	
;	
TRAORI(1)	; 오리엔테이션 변환 선택.
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; CYCLE832 호출: ; 형상 공차 = 0.01 mm, ; 프로세싱 유형: 스무딩이 있는 Top Surface ; 오리엔테이션 공차를 입력하여 정삭 ; 오리엔테이션 공차 = 1도
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; N09부터 N10까지 블록은 일정한 오리엔테이션으로 공작물에 접근하는 동작의 빠른 부분입니다.
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; N11부터 N21까지 블록에서, 접근 동작의 느린 부분이 실행됩니다. 공구가 이미 공작물에 가깝습니다. 또한 CYCLE832의 금형 가공 설정(예: COMPSURF)이 현재 활성화되어 있습니다(활성 G01로 인해). 금형 가공 동작을 위한 이러한 이른바 전환 단계 경로는 형상 공차의 약 1000배여야 합니다(이 예에서는 10 mm).
N12 X-247.69126 Y33.82182 Z24.78219 F1061.00000	
N13 X-247.76560 Y33.13299 Z21.73022	
N14 X-247.82755 Y32.55897 Z19.18691	
N15 X-247.87918 Y32.08062 Z17.06748	
N16 X-247.92220 Y31.68200 Z15.30129	
N17 X-247.95805 Y31.34981 Z13.82947 A3=-.216361686 B3=.934284391 C3=-.283373071	
N18 X-247.98792 Y31.07299 Z12.60295 A3=-.216360662 B3=.934280801 C3=-.283385691	
N19 X-248.01282 Y30.84230 Z11.58085 A3=-.216336015 B3=.934194446 C3=-.283689030	
N20 X-248.03357 Y30.65006 Z10.72910 A3=-.216233089 B3=.933833626 C3=-.284952647	
N21 X-248.05086 Y30.48986 Z10.01931 A5=-.060687572 B5=.974940255 C5=-.214029243 A3=-.215712821 B3=.932005189 C3=-.291263295	

프로그램 코드	설명
N22 **G41** X-248.06237 Y30.32400 Z9.36695 A5=-.060431854 B5=.973045457 C5=-.222554556 A3=-.214974689 B3=.929398552 C3=-.300007025 F1061.03295	; N22부터, 표면 법선이 처음으로 전체 블록에 걸쳐 완전히 정의됩니다(예: 이전 블록 N21에 표면 법선이 있기 때문에 블록 N22의 시작에 표면 법선이 있고 N22 블록의 끝에 표면 법선이 있음). 그러므로 G41/G42로 공구 옵셋을 켜기 위한 사전 조건이 충족됩니다.
N23 X-248.07130 Y30.15119 Z8.71082 A5=-.060165696 B5=.971048883 C5=-.231179920 A3=-.214177198 B3=.926684940 C3=-.308841625	
N24 X-248.07829 Y29.97126 Z8.05094 A5=-.059884286 B5=.968941717 C5=-.239928784 A3=-.213318480 B3=.923853466 C3=-.317789237	
N25 X-248.08317 Y29.78487 Z7.38844 A5=-.059584206 B5=.966718449 C5=-.248807482 A3=-.212397895 B3=.920898045 C3=-.326854594	
N26 X-248.08578 Y29.59254 Z6.72679 A5=-.059263963 B5=.964380907 C5=-.257793037 A3=-.211418355 B3=.917822366 C3=-.336012474	
...	

참고

CUT3DFD을 이용한 3D 페이스 밀링의 경우, G41/G42로 공구 옵셋을 활성화하려면 표면 법선을 정의해야 합니다. 표면 법선을 정의하지 않고 G41/G42를 프로그래밍하면 알람이 출력될 수 있습니다.

다음 또한 참고:

CYCLE832 - 고속 가공 설정 (페이지 1285)

추가 정보

3D 페이스 밀링

이런 유형의 3D 밀링을 위해서는 공작물 표면에서의 3D 경로를 라인 단위로 설명해야 합니다. 공구 형상 및 치수는 계산 시 고려되며 일반적으로 CAM에서 수행됩니다. 포스트 프로세서는 NC 블록 외에 가공 프로그램에 공구 오리엔테이션(활성 5축 변환용) 및 요구되는 3D 공구 옵셋 G 명령을 작성합니다. 즉, 기계 조작자가 NC 경로를 계산할 때 사용했던 공구보다 약간 작은 공구를 사용할 수도 있다는 의미입니다.

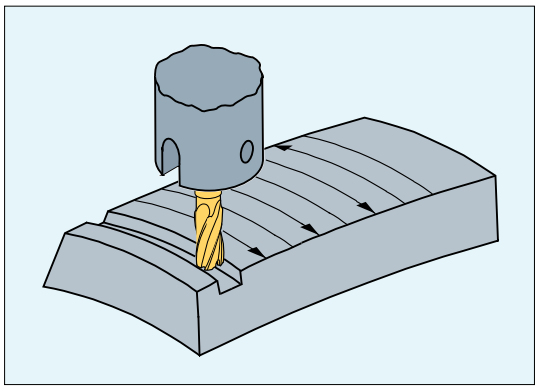


그림 4-11 페이스 밀링

접근 동작

접근 동작은 항상 공구 반경 보정의 3D variant에 대해 NORM입니다.

바깥쪽 코너에서의 동작

바깥쪽 코너는 페이스 밀링을 위해 반경이 0인 원으로 처리되고 원호 평면은 첫 번째 블록의 끝 접선에서 두 번째 블록의 시작 접선으로 연장됩니다. 이러한 방식으로 블록 트랜지션 동안 오리엔테이션을 변경할 수 있습니다. 그러므로 원은 항상 바깥쪽 코너에서 형상 요소로 삽입됩니다. 교차 절차는 페이스 밀링에 적용되지 않습니다.

바깥쪽 코너에서의 오리엔테이션 변경 동작

ORIC 및 ORID G 명령은 코너를 이루는 두 블록 사이에 프로그램된 오리엔테이션 변경을 삽입된 원호 블록(ORID)이 처리되기 전에 실행할지 아니면 동시에 실행할지 결정하는 데 사용됩니다.

차등 공구에 대한 공구 반경 보정

차동 공구를 참조하는 3D 공구 반경 보정은 CUT3DFD 명령을 사용해 선택됩니다. 프로그램된 형상이 표준 공구의 중심점 경로를 참조하고 차동 공구 외 다른 공구가 가공에 사용되는 경우 이것이 적용되어야 합니다. 3D 공구 반경 보정을 계산할 때는 활성 공구($TC_DP15) 반경의 마모값과 프로그래밍된 공구 옵셋 OFFN 및 TOFFR/TOFFLR만 고려됩니다. 활성 공구의 기본 반경 ($TC_DP6) 는 고려되지 **않습니다**.

CUT3DFD를 이용한 3D 페이스 밀링은 "Smoothing of surface normals in 3D face milling(3D 페이스 밀링에서 표면 법선의 스무딩)"과 함께만 가능합니다. 이는 CYCLE832(...)를 통해 "Top Surface" 기능(라이선스 필요)을 호출하여 활성화됩니다. 이 활성화는 G41/G42로 공구 옵셋을 활성화하기 **전에** 되어야 합니다. 공구 개입 직전이 아니라 그 이전의 경로 길이로, 형상 공차의 약 1000배에 해당합니다(예: 1000 x 0.01 mm = 10 mm). 비활성화는 역순으로 수행되어야 합니다. 먼저 G40으로 공구 옵셋을 끈 다음, 형상 공차의 약 1,000배에 해당하는 경로 길이 다음에 CUT2D(또는 이와 유사한 것)를 이용하여 비활성화하십시오.

"Smoothing of surface normals in 3D face milling"을 사용하려면 "Interpolation of surface normals via polynomials(다항식을 통한 표면 법선 보간)" 기능도 활성화되어 있어야 합니다.

MD28291 $MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

참고

"Top Surface"와 함께 CUT3DFD를 이용한 3D 페이스 밀링의 경우, "Top Surface" 관련 권장 설정을 반드시 따라야 합니다.

설정된 데이터 확인을 위해 특수 테스트 프로그램이 SIOS 포털에서 제공됩니다.

→ Top Surface의 테스트 프로그램 (https://support.industry.siemens.com/cs/ww/en/view/109738423)

다음 또한 참고:

CYCLE832 - 고속 가공 설정 (페이지 1285)

4.13.5.3 한계 표면을 고려한 3D 원주 밀링(CUT3DCC, CUT3DCCD)

공구 오리엔테이션의 연속적인 또는 일정한 변경이 있는 3D 원주 밀링에서, 공구 중심점 경로는 정의된 표준 공구에 대해 자주 프로그램됩니다. 실제 상황에서 적절한 표준 공구는 사용하지 못하는 경우가 많기 때문에 표준 공구(≤ 5%)에서 너무 많이 벗어나지 않은 공구를 사용할 수도 있습니다.

CUT3DCCD은 프로그램된 표준 공구가 정의한 실제 차등 공구에 대한 한계 표면을 고려합니다. NC 프로그램은 표준 공구의 중심점 경로를 정의합니다.

실린더 공구를 사용하는 CUT3DCC는 프로그램된 표준 공구가 도달했을 한계 표면을 고려합니다. 가공 표면의 형상을 정의하는 NC 프로그램

한계 표면의 표면 법선 벡터는 3D 페이스 밀링에 대해 A4, B4, C4와 A5, B5, C5로 지정됩니다.

구문

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X... Y... Z... A4=... B4=... C4=... A5=... B5=...
C5=...
...
G40 X... Y... Z...
```

의미

CUT3DCCD:	공구 중심점 경로에서 차동 공구를 사용하는 한계 표면을 고려하는 원주 밀링용 3D TRC: 한계 표면으로 절입	
CUT3DCC:	3D 반경 보정을 사용하는 한계 표면을 고려하는 원주 밀링용 3D TRC: 가공 표면의 형상	
G41/G42 X... Y... Z...:	공구 반경 보정 활성화	
	G41:	형상 좌측 공구 반경 보정
	G42:	형상 우측 공구 반경 보정
	주의: 활성화는 반드시 선형 블록(G0/G1)에서 수행해야 합니다.	
CDOF2:	3D 원주 밀링의 충돌 탐지 비활성화	
A4/5=... B4/5=... C4/5=...:	한계 표면의 표면 법선 정의	
	A4=... B4=... C4=...:	블록 시작에 정의
	A5=... B5=... C5=...:	블록 끝에 정의
G40 X... Y... Z...:	공구 반경 보정 비활성화 **주의:** 비활성화는 반드시 기하축 이동을 통해 선형 블록(G0/G1)에서 수행해야 합니다.	

참고

3D TRC를 선택하기 위한 G 명령은 접근 블록에서 평가됩니다(예: 일반적으로 G41 또는 G42를 포함하는 블록).

G41 또는 G42는 보정 관련 기하 축에서 이송 동작 없이 블록으로 프로그래밍될 수 있습니다. 이 경우, 접근 블록은 이러한 블록 이후 첫 번째 이송 블록입니다.

활성 공구 반경 보정을 사용하는 3D TRC variant의 변경은 알람 없이 무시됩니다.

예

프로그램 코드	설명
N10 $TC_DP1[1,1]=120	; 원통형 밀링 공구
N20 $TC_DP6[1,1]=10	
N30 $TC_DP15[1,1]=-3	
...	

프로그램 코드	설명
; 원통형 밀링 공구와 CUT3DCCD를 이용한 프로세싱	
N110 TRAORI	; 변환 활성화.
N120 A4=0 B4=0 C4=1	; 블록 시작에 한계 표면의 표면 법선 정의
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	
N140 X10 Y0 Z0 G41 **CUT3DCCD** CDOF2 G64	; 3D 원주 밀링 활성화, 한계 표면 고려 + 충돌 탐지 비활성화
N150 X20	
N160 X30 A45	; 둔각 ==> 절입 없음
N170 X40 A-45	; 예각 ==> 절입
N180 X55	
N190 Y10 Z10	; 공구 방향으로 이동.
N200 Y20	
N210 C45	; 오리엔테이션 순 변경.
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; 표면 변경.
N240 Y50 G40	; 공구 반경 보정 비활성화.
...	

추가 정보

공구 종류

공구 유형(공구 파라미터 $TC_DP1)이 평가됩니다. 원통형 샹크(실린더 또는 엔드 밀, 환상면 밀러와 제한된 경우 원통형 다이 밀)를 사용하는 밀링 공구만 허용됩니다. 111번 및 155~157번을 제외한 1~ 399번의 모든 공구 유형이 이에 해당됩니다.

코너 라운딩을 사용하는 표준 공구

표준 공구를 사용한 코너 라운딩은 공구 파라미터 $TC_DP7에 의해 정의됩니다. 공구 파라미터 $TC_DP16은 표준 공구와 비교했을 때 실제 공구의 코너 라운딩의 편차를 설명합니다.

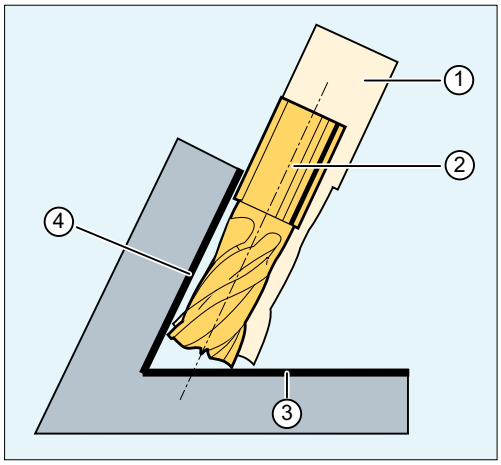
예: 표준 공구와 비교했을 때 반경이 감소한 환상면 밀러

공구 종류	샤프트 반경(R)	코너 반경(r)
코너 라운딩을 사용하는 표준 공구	R = $TC_DP6	r = $TC_DP7
코너 라운딩을 사용하는 실제 공구 공구 유형 121 및 131 환상면 밀러 (코너 라운딩이 있는 엔드 밀)	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16

이번 예에서는 $TC_DP15 + OFFN 및 $TC_DP16이 모두 음수입니다.

CUT3DCCD를 이용한 3D TRC: 한계 표면 까지 절입되는 공구 중심점 경로

적절한 표준 공구의 반경보다 작은 공구를 사용하는 경우, 가공은 밀링 공구를 사용하여 실행됩니다. 이 밀링 공구는 포켓의 바닥(베이스)에 닿을 때까지 수직 방향으로 절입됩니다. 이 공구는 가공 표면 및 한계 표면에 의해 형성된 코너에서 최대한 많은 재료를 제거합니다. 이 과정에는 원주 및 페이스 밀링을 혼합한 가공 형식이 포함됩니다. 반경이 감소된 공구와 비슷하게 반경이 증가된 공구에 대한 절입은 반대 방향으로 이루어집니다.



① 표준 공구
② 한계 표면까지 줄어든 반경으로 절입되는 공구
③ 한계 표면
④ 가공 단면

G 그룹 22의 다른 모든 공구 옵셋과 반대로, CUT3DCCD에 지정된 공구 파라미터 $TC_DP6는 공구 반경과 상관이 없고 그 결과로 나오는 보정에 영향을 주지 않습니다. 보정 옵셋은 한계 표면에 대한 수직 옵셋 계산을 위해 프로그래밍된 공구 옵셋 OFFN 과 공구 반경(공구 파라미터 $TC_DP15)의 마모값을 합한 결과입니다.

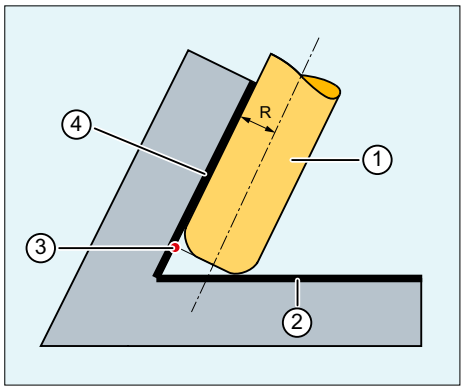
생성된 부품 프로그램에서는 가공될 표면이 경로의 오른쪽인지 왼쪽인지 지정하지 않습니다. 그렇기 때문에 반경이 양수 값이고 원래 공구의 마모값이 음수 값이라고 가정하겠습니다. 음수 마모값은 직경이 줄어든 공구를 나타냅니다.

원통형 공구 사용

원통형 공구를 사용하는 경우, 가공 표면 및 한계 표면이 정확한 각을 이룰 때(90도 이하)만 절입이 필요합니다. 환상면 밀러(코너 라운딩이 있는 엔드 밀)가 사용된 경우, 공구를 예각 및 둔각으로 절입할 때 수직 방향이 되어야 합니다.

CUT3DCC를 이용한 3D TRC: 가공 표면의 형상

환상면 밀러를 사용하여 CUT3DCC가 활성화된 경우, 프로그램된 경로는 직경이 동일한 가상의 원통형 밀링 공구를 참조합니다. 그에 따른 경로 기준점이 환상면 밀러에 대한 다음 다이어그램에 나와 있습니다.



① 환상면 밀러
② 한계 표면
③ 경로 기준점
④ 가공 단면
R 샤프트 반경 (공구 반경)

가공 표면과 한계 표면 사이의 각도는 동일한 블록 내라 할지라도 예각에서 둔각 또는 둔각에서 예각으로 변경될 수 있습니다.

이 공구는 실질적으로 표준 공구보다 크건 작건 상관 없이 사용됩니다. 하지만 결과 코너 반경은 음수가 되어야 하며, 결과 공구 반경 기호는 반드시 유지해야 합니다.

CUT3DCC에서, NC 부품 프로그램은 가공 표면의 형상을 참조합니다. 기존의 공구 반경 보정의 경우, 다음 컴포넌트로 구성된 총 공구 반경이 사용됩니다.

- 공구 반경 (공구 파라미터 $TC_DP6)
- 마모값 (공구 파라미터 $TC_DP15)
- 한계 표면에 대한 수직 옵셋 계산을 위해 프로그램밍되는 공구 옵셋 OFFN

한계 표면의 위치는 다음 차에 따라 정의됩니다.:

표준 공구의 수치 - 공구 반경(공구 파라미터 $TC_DP6)

Advanced Surface / Top Surface

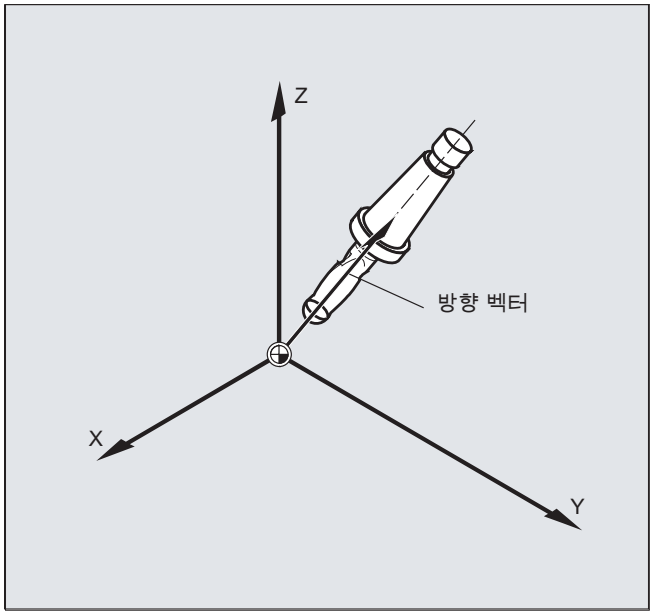
참고

공구 반지름 보정 CUT3DCC / CUT3DCCD를 "Advanced Surface" 또는 "Top Surface" 기능(라이선스 필요)과 함께 적용하는 경우, "Advanced Surface" / "Top Surface" 관련 권장 설정을 반드시 준수해야 합니다. 설정된 데이터 확인을 위해 특수 테스트 프로그램이 SIOS 포털에서 제공됩니다.

→ Advanced Surface / Top Surface의 테스트 프로그램 (https://support.industry.siemens.com/cs/ww/en/view/109738423)

4.13.6 공구 오리엔테이션 (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

공구 오리엔테이션이란 공간 내에서 공구의 기하학적 정렬을 말합니다. 5축 기계에서는 프로그램 명령을 사용하여 공구 오리엔테이션을 설정할 수 있습니다.



오리엔테이션 라운딩 이송은 OSD 및 OST로 활성화되며, 이 둘은 공구 오리엔테이션에 대한 보간 유형에 따라 다르게 형성됩니다.

벡터 보간이 활성화되어 있는 경우, 스무딩된 오리엔테이션 특성 역시 벡터 보간을 사용하여 보간할 수 있습니다. 반면, 로터리 축 보간이 활성화되어 있는 경우, 그 오리엔테이션은 로터리 축 움직임을 직접 사용하여 스무딩 됩니다.

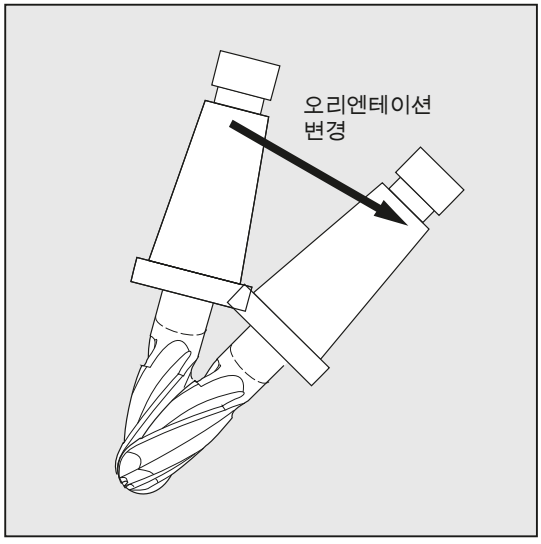
프로그래밍

오리엔테이션 변경 프로그래밍:

공구 오리엔테이션을 변경하도록 프로그램할 수 있는 방법

- 로터리 축 A, B, C를 직접 프로그래밍 (로터리 축 보간)
- 오일러 (Euler) 각도 또는 RPY 각도
- 방향 벡터 (A3, B3 또는 C3 지정을 통한 벡터 보간)
- LEAD/TILT (페이스 밀링)

기준 좌표계는 기계 좌표계 (ORIMKS) 또는 현재 공작물 좌표계 (ORIWKS) 중 하나가 될 수 있습니다.



공구 오리엔테이션 프로그래밍:

ORIC:	오리엔테이션 및 경로 이송이 평행
ORID:	오리엔테이션 및 경로 이송이 일관됨
OSOF:	오리엔테이션 스무딩 없음
OSC:	일관된 오리엔테이션
OSS:	블록 시작에서만 오리엔테이션 스무딩
OSSE:	블록의 시작 및 끝에서 오리엔테이션 스무딩
ORIS:	밀리미터 (mm) 당 각도 단위로 활성화된 오리엔테이션 스무딩을 사용한 오리엔테이션 변경 속도 (OSS 및 OSSE에 적용)

OSD:	셋팅 데이터로 스무딩 거리를 지정하여 오리엔테이션 스무딩: SD42674 $SC_ORI_SMOOTH_DIST
OST:	셋팅 데이터로 벡터 보간에 대한 각도 공차를 지정하여 오리엔테이션 스무딩: SD42676 $SC_ORI_SMOOTH_TOL 로터리 축 보간을 사용하여 지정되어 있는 공차가 오리엔테이션 축의 최대 편차가 되도록 하십시오.

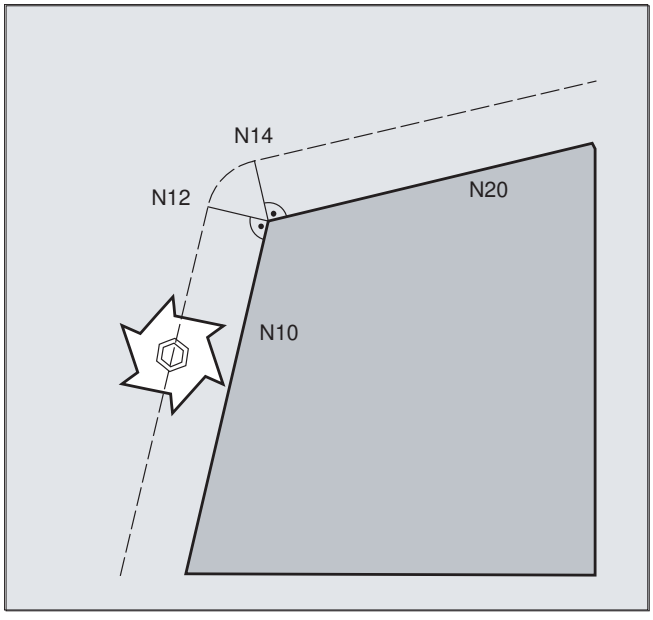
참고

공구 오리엔테이션 스무딩을 위한 명령(OSOF, OSC, OSS, OSSE, OSD, 및 OST)는 G 코드 그룹 34에 요약되어 있습니다. 이들은 모달식인데, 다시 말하자면 이러한 명령 중 하나만 한 시점에 유효하다는 것입니다.

예

예 1: ORIC

만일 오리엔테이션 변경이 있는 2개 또는 그 이상의 블록이 이송 블록 N10 및 N20 사이에 프로그램 되고(예: A2=... B2=... C2=...) 아울러 ORIC가 활성화된 경우라면, 삽입된 원호 블록은 각도 상의 절대 변화에 따라 이러한 중간 블록 사이에 배분됩니다.

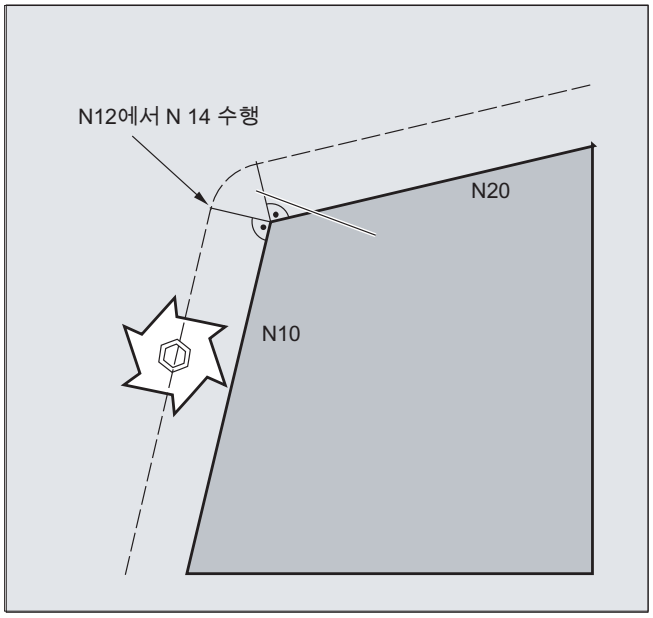


프로그램 코드	코멘트
ORIC	

프로그램 코드	코멘트
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 C2=… B2=…	; 외부 코너에 삽입된 원호 블록은 오리엔테이션 변경에 따라 N12 와 N14 사이에 분산됩니다. 원호 이송 및 오리엔테이션 변경은 평행하게 실행됩니다.
N14 C2=… B2=…	
N20 X =…Y=… Z=… G1 F200	

예 2: ORID

ORID가 활성화되면, 2개의 이송 블록 사이에 있는 모든 블록은 첫 번째 이송 블록의 끝에서 실행됩니다. 일정 오리엔테이션을 사용하는 원호 블록은 두 번째 이송 블록 직전에 실행됩니다.



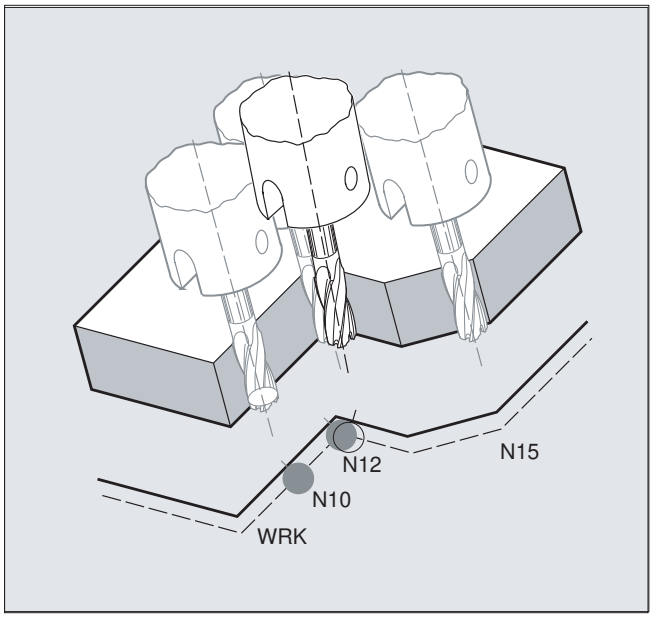
프로그램 코드	코멘트
ORID	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 A2=… B2=… C2=…	; N12 및 N14 블록이 N10 블록의 끝에서 실행됩니다. 그 이후 원호 블록은 실제 오리엔테이션으로 실행됩니다.
N14 M20	; 도움말 기능 등
N20 X… Y… Z…	

참고

외부 형상에서 오리엔테이션 변경에 사용되는 방법은 외부 코너의 첫 번째 이송 블록에서 활성화된 프로그램 명령을 사용하여 결정됩니다.

오리엔테이션이 변경되지 않는 경우: 블록 경계에서 오리엔테이션이 변경되지 않은 경우 공구의 교차 섹션은 양쪽 형상과 모두 접촉하는 원호가 됩니다.

예 3: 내부 코너에서 오리엔테이션 변경



프로그램 코드

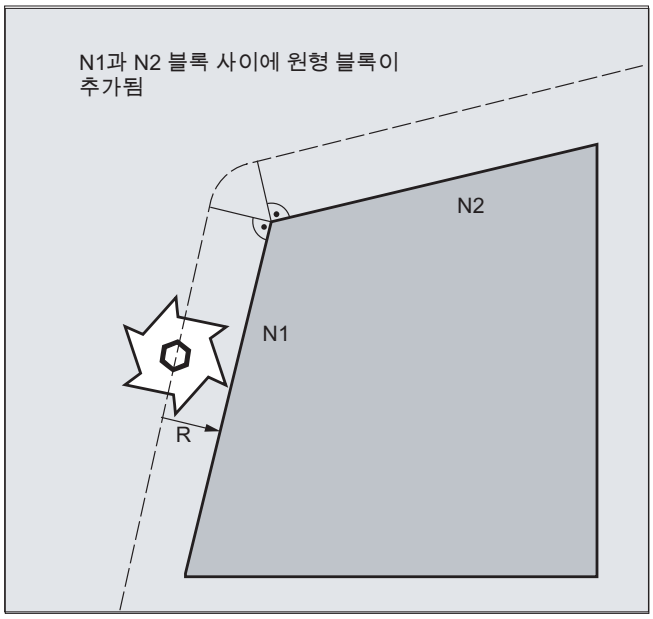
```
ORIC
N10 X …Y… Z… G1 F500
N12 X …Y… Z… A2=… B2=… C2=…
N15 X …Y… Z… A2=… B2=… C2=…
```

추가 정보

외부 코너에서의 동작

커터의 반경을 사용하는 원호 블록은 항상 외부 코너에 삽입됩니다.

ORIC 및 ORID 프로그램 명령은 N1 및 N2 블록 사이에 프로그램된 오리엔테이션을 변경할 것인지 결정하는 데 사용되며, 삽입된 원호 블록이 처리되기 전에 실행되거나 또는 동시에 실행됩니다.



외부 코너에서 오리엔테이션을 변경해야 하는 경우에는 경로 이동과 함께 별도로 실행되거나 또는 보간과 동시에 실행되도록 할 수 있습니다.

ORID가 프로그램된 경우, 삽입된 블록은 경로 이동 없이 먼저 실행됩니다. 코너를 생성하는 원호 블록은 두 개의 이송 블록 중 두 번째 블록 직전에 삽입됩니다.

몇 개의 오리엔테이션 블록이 외부 코너에 삽입되고, ORIC가 선택된 경우, 원호 이송은 오리엔테이션 변경의 절대값에 따라 개별적으로 삽입된 블록 사이로 배분됩니다.

OSD 또는 OST를 사용한 오리엔테이션 스무딩

G642를 사용하여 블렌딩할 때, 기하 축 및 오리엔테이션 축에 대한 최대 편차는 되도록 적어야 합니다. 공차 중 작은 쪽은 스무딩 모션 및/또는 각도 공차의 유형을 결정합니다. 따라서 높은 형상 편차를 수용하지 않고도 오리엔테이션 특성을 강하게 스무딩합니다.

심한 형상 편차 없이 지정된 스무딩 거리 및 각도 공차대로 오리엔테이션 특성에서 아주 조금만 벗어나게 스무딩하도록 OSD 및 OST를 활성화할 수 있습니다.

참고

G642를 사용한 형상 (및 오리엔테이션 특성) 의 라운딩 가공과는 달리, OSD 및/또는 OST를 사용하는 오리엔테이션을 라운딩할 때, 별도의 블록이 형성되지 않는 대신 라운딩 움직임이 프로그램된 원본 블록에 직접 추가됩니다.

OSD 및/또는 OST를 사용하는 블록 트렌지션은 공구 오리엔테이션에 대한 보간의 유형이 변경 (벡터 → 로터리 축, 로터리 축 → 벡터) 된 경우에는 라운딩이 될 수 없습니다. 이러한 블록 트렌지션은 필요한 경우, 표준 라운딩 기능 G641, G642 및 G643을 사용하여 라운딩될 수 있습니다.

4.13.7 D 번호, 절삭날 번호의 자유 할당

4.13.7.1 D 번호, 절삭날 번호의 자유 할당 (CE 어드레스)

D 번호

D 번호는 옵셋 번호로 사용할 수 있습니다. CE 주소를 통해 절삭날 번호의 주소를 지정할 수 있습니다. 절삭날 번호는 시스템 변수인 $TC_DPCE에 의해 작성될 수 있습니다.

기본 설정: 보정 번호 == 절삭날 번호

머신 데이터는 D 번호의 최대 번호(절삭날 번호) 및 공구 당 절삭날의 최대 개수(→ 장비 제조업체)를 정의하는 데 사용됩니다. 다음 명령은 최대 절삭 번호(MD18105)가 공구 당 절삭 번호(MD18106)보다 클 때만 사용됩니다. 장비 제조업체의 사양을 참조하십시오.

추가 정보

기능 매뉴얼 공구

4.13.7.2 D 번호의 자유 할당: D 번호 확인(CHKDNO)

CKKDNO 명령을 사용하면 기존 D 번호가 독자적으로 할당된 것인지 확인할 수 있습니다. 모든 공구의 D 번호는 TO 유닛 내에서 정의되며, 한 번만 나옵니다. 예비 공구는 허용하지 않습니다.

구문

```
state=CHKDNO(Tno1,Tno2,Dno)
```

의미

상태:	=TRUE:	D 번호는 확인된 영역에만 독자적으로 할당됩니다.
	= FALSE:	D 번호 충돌이 발생했거나 파라미터가 유효하지 않습니다. Tno1, Tno2 및 Dno는 충돌을 야기한 파라미터를 반환합니다. 이 데이터는 부품 프로그램에서 평가됩니다.
CHKDNO(Tno1,Tno2):	지정된 부품의 D 번호가 모두 확인되었습니다.	

CHKDNO(Tno1):	Tno1의 D 번호가 다른 공구와 비교하여 모두 확인되었습니다.
CHKDNO:	공구의 D 번호가 다른 공구와 비교하여 모두 확인되었습니다.

4.13.7.3 D 번호의 자유 할당: D 번호 이름 변경(GETDNO, SETDNO)

독자적인 D 번호를 할당하셔야만 합니다. 2개의 서로 다른 공구의 절삭날에 동일한 D 번호를 할당해서는 안 됩니다.

GETDNO

이 명령은 공구 번호가 t인 공구의 특정 절삭날 (ce) 의 D 번호를 반환합니다. 입력된 파라미터에 D 번호가 없으면 d=0으로 설정됩니다. D 번호가 유효하지 않는 경우에는 32000 이상의 값이 반환됩니다.

SETDNO

이 명령은 공구 t의 절삭날 (ce) 에 D 번호 값 d를 할당합니다. 이 구문의 결과는 상태 (TRUE 또는 FALSE) 로 반환됩니다. 지정된 파라미터에 대한 데이터 블록이 없는 경우에는 FALSE 값이 반환됩니다. 구문 오류가 발생하면 알람이 발생합니다. D 번호를 직접 0으로 설정해서는 안 됩니다.

구문

```
d = GETDNO (t,ce)

state = SETDNO (t,ce,d)
```

의미

d:	공구 날의 D 번호
t:	공구의 T 번호
ce:	공구의 절삭날 번호(CE 번호)
상태:	명령의 실행 여부를 나타냅니다(TRUE 또는 FALSE).

D 번호 이름 변경의 예제

프로그래밍	코멘트
$TC_DP2[1.2]=120	
$TC_DP3[1,2] = 5.5	
$TC_DPCE[1,2] = 3	; 절삭날 번호 CE
...	

프로그래밍	코멘트

```
N10 def int DNoOld, DNoNew = 17
N20 DNoOld = GETDNO(1,3)
N30 SETDNO(1,3,DNoNew)
```

새 D 값 17이 절삭날 CE=3에 할당됩니다. 이제 절삭날 데이터는 D 번호 17을 통해 어드레스를 지정합니다. 시스템 변수로 지정하거나 NC 어드레스를 프로그래밍하여 지정할 수 있습니다.

4.13.7.4 D 번호의 자유 할당: 지정된 D 번호의 T 번호를 결정(GETACTTD)

사전 정의된 GETACTTD 기능은 절대 D 번호와 관련된 T 번호를 결정합니다. 고유성은 검사하지 않습니다. TO 유닛 내에 있는 몇 개의 D 번호가 동일한 경우, 검색에서 발견된 첫 번째 공구의 T 번호가 반환됩니다.

구문

```
<Status>=GETACTTD(<TNo>,<DNo>)
```

의미

GETACTTD():	기능 호출	
<DNo>:	검색하게 될 T 번호에 대한 D 번호	
	데이터 유형:	INT
<TNo>:	발견된 T 번호	
	데이터 유형:	VAR INT

<status>:	결과		
	데이터 유형:	INT	
	값:	0	T 번호가 발견되었습니다. <Tno>는 T 번호의 값을 포함합니다.
		-1	지정된 D 번호에 대한 T 번호가 없습니다. <Tno>=0
		-2	D 번호가 절대적이지 않습니다. <Tno>는 <Dno> 값을 가진 D 번호를 포함하는 처음 발견된 공구의 값을 수신합니다.
		-5	다른 이유로는 이 기능을 실행할 수 없습니다.

4.13.7.5 D 번호의 자유 할당: 무효화된 D 번호(DZERO)

4.13.8 공구 홀더 좌표계

요구 사항

공구 홀더는 공간 내에서 다음과 같은 경우에만 사용 가능한 모든 방향에서 공구의 방향을 잡을 수 있습니다.

- 2개 로터리 축 V_1 및 V_2가 존재하는 경우
- 로터리 축들이 서로 직각인 경우
- 공구의 세로 축이 두 번째 로터리 축 V_2와 직각인 경우

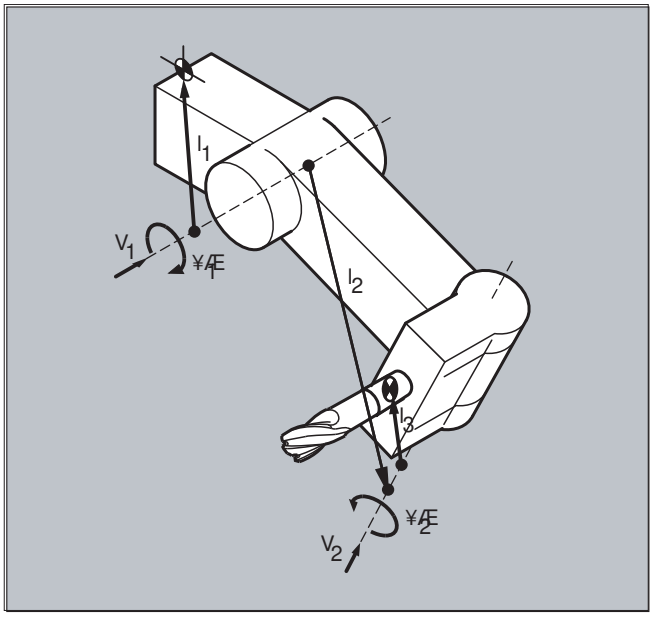
또한, 모든 사용 가능한 방향을 설정할 수 있는 경우 기계에 다음 요구 사항이 적용됩니다.

- 공구의 세로 축은 첫 번째 로터리 축 V_1과 직각이어야 합니다.

기능

최대 2개의 로터리 축 v_1 또는 v_2를 사용하는 공구 홀더 좌표계는 17개의 시스템 변수 $TC_CARR1[m]부터 $TC_CARR17[m]을 사용하여 정의합니다. 공구 홀더의 설명은 다음으로 구성되어 있습니다.

- 공구 홀더의 첫 번째 로터리 축부터 벡터 거리 I_1, 첫 번째 로터리 축부터 두 번째 로터리 축까지의 벡터 거리 I_2, 두 번째 로터리 축부터 공구 원점까지의 벡터 거리 I_3.
- 두 로터리 축 V_1 및 V_2의 방향 벡터.
- 두 축을 기준으로 한 회전 각도 α1, α2. 회전 각도는 로터리 축 벡터는 보기 방향, 즉 회전의 시계 방향인 양의 방향으로 계산됩니다.



고정 좌표계를 사용하는 기계 (공구와 부품 모두 회전 가능) 를 위해 $TC_CARR18[m]부터 $TC_CARR23[m]으로 시스템 변수를 확장했습니다.

파라미터

방향 조정이 가능한 공구 홀더에 대한 시스템 변수의 기능			
지정	**x 요소**	**y 요소**	**z 요소**
I_1 옵셋 벡터	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
I_2 옵셋 벡터	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
v_1 로터리 축	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
v_2 로터리 축	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]

방향 조정이 가능한 공구 홀더에 대한 시스템 변수의 기능			
α_1 회전 각도 α_2 회전 각도	\$TC_CARR13[m] \$TC_CARR14[m]		
l_3 옵셋 벡터	\$TC_CARR15[m]	\$TC_CARR16[m]	\$TC_CARR17[m]

방향 조정이 가능한 공구 홀더에 대한 시스템 변수의 확장			
지정	**x 요소**	**y 요소**	**z 요소**
l_4 옵셋 벡터	\$TC_CARR18[m]	\$TC_CARR19[m]	\$TC_CARR20[m]
축 이름 로터리 축 v_1 로터리 축 v_2	로터리 축 v_1 및 v_2의 축 이름 (0으로 초기화) \$TC_CARR21[m] \$TC_CARR22[m]		
운동학적 유형 **공**구 **부**품 **혼**합 모드	\$TC_CARR23[m]		
	운동학적 유형 T ->	운동학적 유형 P ->	운동학적 유형 M
	공구만 회전 가능(기본).	부품만 회전 가능	부품 및 공구 회전 가능
옵셋 로터리 축 v_1 로터리 축 v_2	초기 설정 적용 시 로터리 축 v_1 및 v_2의 각도 (° 단위) \$TC_CARR24[m] \$TC_CARR25[m]		
각도 옵셋 로터리 축 v_1 로터리 축 v_2	로터리 축 v_1 및 v_2에 대한 Hirth 절삭날 시스템의 옵셋 (° 단위) \$TC_CARR26[m] \$TC_CARR27[m]		
각도 증분 v_1 로터리 축 v_2 로터리 축	로터리 축 v_1 및 v_2대한 Hirth 절삭날 시스템의 옵셋 (° 단위) \$TC_CARR28[m] \$TC_CARR29[m]		
최소 위치 로터리 축 v_1 로터리 축 v_2	로터리 축 v_1 및 v_2의 최소 위치에 대한 프트웨어 한계 \$TC_CARR30[m] \$TC_CARR31[m]		
최대 위치 로터리 축 v_1 로터리 축 v_2	로터리 축 v_1 및 v_2의 최대 위치에 대한 소프트웨어 한계 \$TC_CARR32[m] \$TC_CARR33[m]		
공구 홀더 이름	공구 홀더에는 번호 대신 이름을 붙일 수 있습니다. \$TC_CARR34[m]		

방향 조정이 가능한 공구 홀더에 대한 시스템 변수의 확장			
사용자: 축 이름 1 축 이름 2 식별자 **위치**	사용자 측정 싸이클 $TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]에 의도적으로 사용		
	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
미세 **옵셋**	**기본 파라미터** 의 **값**에 추가될 수 있는 파라미터.		
l_1 옵셋 벡터	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
l_2 옵셋 벡터	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
l_3 옵셋 벡터	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
l_4 옵셋 벡터	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
v_1 로터리 축	$TC_CARR64[m]		
v_2 로터리 축	$TC_CARR65[m]		

참고

파라미터에 대한 설명

"m"은 프로그램 될 공구 홀더의 개수를 지정합니다.

$TC_CARR47부터 $TC_CARR54 및 $TC_CARR61부터 $TC_CARR63은 정의되지 않았고, 읽기 또는 쓰기 액세스가 시도되면 알람이 생성됩니다.

축에 있는 거리 벡터의 시작점/종점은 자유롭게 선택할 수 있습니다. 두 축을 중심으로 하는 회전 각도 α_1 및 α_2는 공구 홀더의 초기 상태에서 0°으로 정의됩니다. 이런 방법으로 공구 홀더의 운동학은 사용 가능한 어떤 번호에 대해서도 프로그램이 될 수 있습니다.

로터리 축이 없거나, 한 개만 있는 공구 홀더는 한 개 또는 양쪽의 로터리 축의 방향 벡터를 0으로 설정하여 설명할 수 있습니다.
로터리 축이 없는 공구 홀더의 경우 거리 벡터는 가공 평면 (G17~G19) 을 변경해도 영향을 받지 않는 콤포넌트를 가진 추가 공구 옵셋으로 작동합니다.

파라미터 확장자

로터리 축의 파라미터

시스템 변수는 엔트리인 $TC_CARR24[m]부터 $TC_CARR33[m]까지 및 다음에 나와 있는 사항에 의해 확장되었습니다.

로터리 축 v_1, v_2의 **옵셋**	방향이 정해진 공구 홀더의 초기 설정에 대한 로터리 축 v_1 또는 v_2의 위치가 변경됩니다.
로터리 축 v_1, v_2의 **각도 옵셋/각도 증분**	로터리 축 v_1 및 v_2의 Hirth 투스 시스템의 옵셋 또는 각도 증분 프로그램 또는 계산된 각도는 n이 정수일 때 phi = s + n * 계산에서 나온 결론인 다음 값까지 라운딩이 됩니다.
로터리 축 v_1, v_2의 **최소 위치 및 최대 위치**	로터리 축 v1 및 v2의 로터리 축 제한 각도(소프트웨어 제한)의 최소 및 최대 위치

사용자용 파라미터

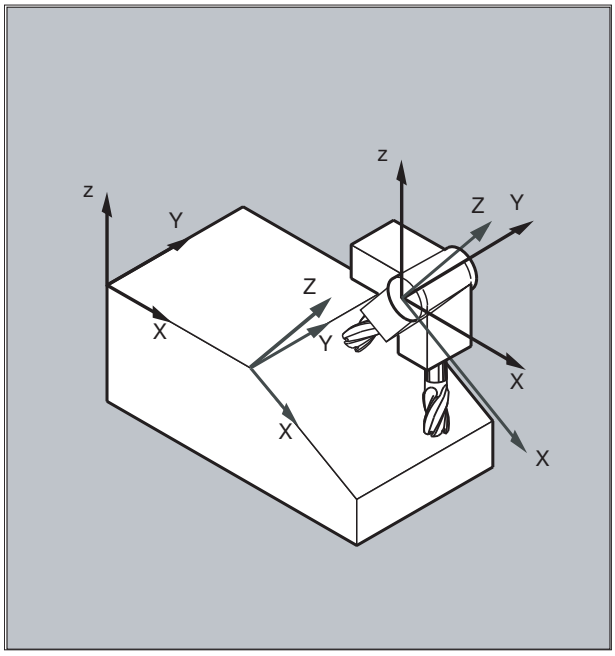
$TC_CARR34 to $TC_CARR40에는 파라미터가 포함되어 있습니다. 이 파라미터는 사용자가 자유롭게 사용할 수 있으며, 최대 SW 6.4까지 표준화되어 있습니다. NCK에서는 더 이상 평가되지 않으며 중요하지 않습니다.

미세 옵셋 파라미터

$TC_CARR41부터 $TC_CARR65에는 기본 파라미터에 있는 값에 추가할 수 있는 미세 옵셋 파라미터가 포함되어 있습니다. 기본 파라미터에 할당된 미세 옵셋 값은 파라미터 번호에 값 40이 추가되었을 때 얻어집니다.

예

다음 예제에서 사용되는 공구 홀더는 Y 축을 중심으로 한 회전에 의해 모두 설명될 수 있습니다.



프로그램 코드	코멘트
N10 $TC_CARR8[1]=1	; 공구 홀더 1의 첫 번째 로터리 축의 Y 콤포넌트에 대한 설명.
N20 $TC_DP1[1,1] = 120	; 샤프트 밀러에 대한 설명.
N30 $TC_DP3[1,1]=20	; 20 mm 길이의 샤프트 밀러에 대한 설명.
N40 $TC_DP6[1,1]=5	; 반경 5 mm의 샤프트 밀러에 대한 설명.
N50 ROT Y37	; Y 축을 중심으로 37° 회전하는 프레임 정의.
N60 X0 Y0 Z0 F10000	; 시작 위치에 접근.
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	회전된 프레임에서 반경 보정, 공구 길이를 설정한 뒤 공구 홀더 1, 공구 1을 선택
N80 X40	; 37° 회전시켜 가공 실행.
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

추가 정보

결정된 구조

결정된 구조를 사용하는 머신(회전 가능한 공작물뿐 아니라 공구도 포함)에 있어서, 시스템 변수는 다음을 포함하여 $TC_CARR18[m]에서 최대 $TC_CARR23[m]까지 항목에 의해 확장됩니다.

다음으로 구성된 회전 가능한 공구 테이블

- 두 번째 로터리 축 V_2와 회전 가능한 세 번째 로터리 축의 I_4인 공구 테이블의 원점 사이의 벡터 공차

다음으로 구성된 로터리 축

- 로터리 축 V_1 및 V_2의 참조에 관한 2개 채널 식별자로, 필요한 경우 전향적이 될 수 있는 공구 홀더의 방향을 결정하기 위해 그 위치로 액세스할 수 있음

T, P, M 값 중 하나를 사용하는 운동학적 유형

- 운동학적 유형 T: 공구만 회전 가능
- 운동학적 유형 P: 부품만 회전 가능
- 운동학적 유형 M: 공구 및 부품 회전 가능

공구 홀더 데이터 공차

공구 홀더 데이터 세트의 모든 데이터는 $TC_CARR1[0]=0를 사용하여 삭제할 수 있습니다.

키네마틱스 유형 $TC_CARR23[T]=T에는 3개의 사용 가능한 대문자 또는 소문자 (T, P, M) 중 하나가 할당되어야 하며 이런 이유로 삭제되어서는 안됩니다.

공구 홀더 데이터 변경

설명된 값은 각각 부품 프로그램에 새로운 값을 할당하여 수정될 수 있습니다. T, P, M 이외의 다른 문자를 사용하면 전향적이 될 수 있는 공구 홀더 활성화 시도를 할 때, 알람을 발생시킵니다.

공구 홀더 데이터 읽기

설명된 값은 각각의 값을 부품 프로그램에 있는 변수에 할당하여 읽혀질 수 있습니다.

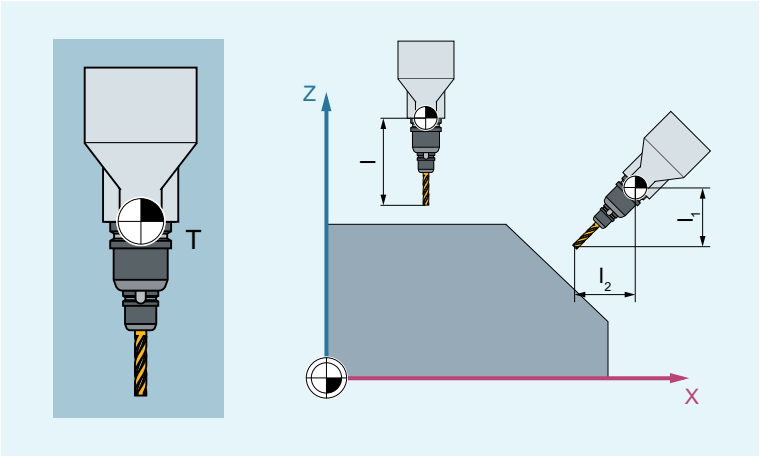
미세 옵셋

허용되지 않는 미세 옵셋값은 전향적이 될 수 있는 공구 홀더가 활성화된 경우에만 감지됩니다. 이 공구 홀더에는 이런 값이 포함되어 있으며, 동시에 셋팅 데이터는 SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE가 됩니다.

최대 허용 가능한 미세 옵셋은 머신 데이터 내의 허용 가능한 값으로 제한됩니다.

4.13.9 방향 조정 가능한 공구 홀더에 대한 공구 길이 보정 (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

공구 캐리어 및 공구의 공간 오리엔테이션이 바뀌면 공구 길이 요소도 바뀝니다.



T 공구 캐리어 원점

l_1, l_2 공구 길이 요소

수동 설정 또는 고정된 공간 오리엔테이션의 공구 캐리어를 변경하는 등의 방법으로 리셋한 후에는 공구 길이 요소도 다시 지정해야 합니다. 이를 위해 G 그룹 42 "공구 캐리어" 명령이 사용됩니다.

구문

```
TCARR=<m>
TCOABS
TCOFR
TCOFRZ/TCOFRY/TCOFRX
```

의미

요소	유형	의미	
TCARR=<m>	주소	공구 캐리어 요청	
		<m>	공구 캐리어 번호
TCOABS	G 명령어	현재 공구 캐리어의 오리엔테이션에서 공구 길이 요소 결정	
TCOFR	G 명령어	활성화된 프레임의 오리엔테이션에서 공구 길이 요소 결정	

요소	유형	의미	
TCOFRZ/TCOFRY/TCOFRX	G 명령어	TCOFRX/TCOFRY/TCOFRZ 명령은 공구가 해당 방향(X / Y / Z)으로 향한다고 가정하고 활성 프레임의 공구가 같은 방향으로 향하도록 오리 엔테이션 가능 공구 캐리어의 설정 각도를 계산합니다.	
		TCOFRZ	Z 방향으로 향하는 공구는 활성 프레임에서도 Z 방향으로 향하도록 정렬됩니다.
		TCOFRY	Y 방향으로 향하는 공구는 활성 프레임에서도 Y 방향으로 향하도록 정렬됩니다.
		TCOFRX	X 방향으로 향하는 공구는 활성 프레임에서도 X 방향으로 향하도록 정렬됩니다.

추가 정보

공구 캐리어 선택(TCARR)

공구 캐리어 데이터 블록은 자신의 형상을 설명하는 공구 캐리어에 연결됩니다. 공구가 공구 캐리어 선택 시점에 활성 상태이면 새로 선택한 공구 캐리어에 지정되고 공구 캐리어의 형상 데이터가 바로 적용됩니다. 활성 공구를 변경하거나 재프로그래밍하지 않아도 됩니다. 새 공구는 활성화되면 활성 공구 캐리어에 장착된 것처럼 취급됩니다.

TCARR=0이면 활성 공구 캐리어가 선택 해제됩니다.

참고

공구 캐리어의 현재 형상 데이터는 해당 시스템 변수를 통해 NC 프로그램에서 정의할 수도 있습니다("공구 캐리어 데이터 블록 액세스" 참조).

참고

제어 측면에서 공구 캐리어 번호 <m>과 공구 번호 <n>은 자유롭게 조합될 수 있습니다. 그러나 실제 적용에서 특정 조합은 가공 또는 기계적 이유로 인해 배제될 수 있습니다. 제어 시스템은 조합이 유효한지 확인하지 않습니다.

공구 캐리어 데이터 블록 액세스

NC 프로그램에서 공구 캐리어의 현재 형상 데이터에 액세스하는 방법은 다음과 같습니다.

- 쓰기:
  `$TC_CARR<n>[<m>]=<Value>`
- 읽기:
  `<Value>=$TC_CARR<n>[<m>]`
  (<Value>는 반드시 데이터 유형이 REAL인 변수여야 합니다.)

참조된 공구 캐리어가 정의되지 않은 경우 알람이 표시됩니다.

공구 캐리어 데이터 블록을 0으로 설정

모든 공구 캐리어 데이터 블록의 모든 값은 하나의 명령어를 사용해 가공 프로그램에서 삭제할 수 있습니다.

```
$TC_CARR1[0] = 0
```

개별 공구 캐리어 데이터 블록은 사전 정의된 프로시저 DELTC를 통해 선별적으로 삭제할 수 있습니다.

공구 캐리어 선택의 적용

공구 캐리어와 공구가 모두 활성화되면 공구 캐리어가 적용됩니다. 공구 캐리어 선택만으로는 효과가 없습니다.

공구 캐리어 선택의 적용은 TCARR과 함께 프로그래밍된 G 그룹 42(공구 캐리어)의 명령에 달려 있습니다.

이 그룹에서 G 명령어를 바꾸면 공구 캐리어 활성 시 공구 길이 요소가 다시 계산됩니다.

공구 캐리어 오리엔테이션에서 공구 길이 보정 계산(TCOABS)

TCOABS는 시스템 변수 $TC_CARR13 및 $TC_CARR14에 저장되어 있는 공구 캐리어의 현재 오리엔테이션 각도에서 공구 길이 보정을 계산합니다. 고려된 오리엔테이션은 활성 프레임의 오리엔테이션과 무관합니다.

시스템 변수가 있는 공구 캐리어 키네마틱의 정의는 NC 프로그래밍 매뉴얼의 "공구 캐리어 키네마틱 (페이지 870)" 장을 참조하십시오.

활성 프레임의 오리엔테이션에서 공구 길이 요소 결정(TCOFR)

TCOFR을 이용해 활성 프레임의 오리엔테이션으로부터 공구 캐리어의 오리엔테이션 각도를 결정할 수 있습니다. 그러나 공구 캐리어 데이터에 저장된 값은 변하지 않습니다. 이것들은 프레임에서 회전 각도를 계산할 때 발생할 수 있는 모호성을 해결하는 데도 사용됩니다 (다음 단락 참조).

모호성

두 개의 축을 이용하여, 프레임에 의해 정의된 특정 공구 오리엔테이션을 서로 다른 **두** 회전 각도 쌍으로 일반적으로 설정할 수 있습니다. 제어 시스템은 가능한 이 두 가지 옵션 중 회전 각도가 프로그래밍된 회전 각도에 최대한 가까운 것을 선택합니다.

참고

모호성이 발생할 수 있는 경우 일반적으로 프레임에서 예상되는 각도를 공구 캐리어 데이터에 저장해야 합니다.

활성 프레임의 공구 오리엔테이션(TCOFRZ/TCOFRY/TCOFRX)

TCOFRX/TCOFRY/TCOFRZ 명령은 공구 캐리어의 설정 각도를 계산하기 위해 활성 오리엔테이션과 상관 없이 X/Y/Z 방향으로 오리엔테이션을 가정합니다. 가공 평면 선택(G17/G18/G19)은 활성 프레임의 오리엔테이션에 영향을 주지 않습니다. 하지만 공구 길이 보정 계산에는 영향을 미칩니다.

프레임 변경

사용자는 공구 선택 후 프레임을 변경할 수 있습니다. 이것은 공구 길이 보정 요소에 어떤 영향도 미치지 않습니다.

공구 캐리어 데이터에 저장된 회전 각도는 프레임으로 정의된 회전 각도의 영향을 받지 않습니다. TCOFR에서 TCOABS로 변경하면 공구 캐리어 데이터의 원래(프로그래밍된) 회전 각도가 다시 활성화됩니다.

공구 길이 보정 재계산

프레임 변경 후 공구 길이 보정을 재계산하려면 공구를 반드시 다시 선택해야 합니다. 공구 길이 보정이 계산되면 공구 캐리어의 회전 각도가 중간 단계에서 계산됩니다. 두 개의 회전 축이 있는 공구 캐리어에는 일반적으로 공구 오리엔테이션을 활성 프레임에 적용할 수 있는 두 쌍의 회전 각도가 있으므로 공구 캐리어 데이터에 저장된 회전 각도 값은 최소한 기계적으로 설정된 회전 각도와 거의 일치해야 합니다.

참고

모호성이 발생할 수 있는 거의 모든 경우, 프레임에서 예상되는 대략적인 각도를 공구 캐리어 데이터에 저장해야 합니다.

참고

공구 오리엔테이션은 활성화된 프레임에 맞게 수동으로 조정되어야만 합니다.

참고

제어 시스템은 기계 상의 조정 가능성을 위해 프레임 오리엔테이션을 통해 계산된 회전 각도를 확인할 수 없습니다. 기계가 프레임 오리엔테이션에 의해 계산된 공구 오리엔테이션에 공구 캐리어의 회전 축이 도달할 수 없는 방식으로 설계된 경우, 알람이 출력됩니다.

참고

공구 정밀도 보정과 움직일 수 있는 공구 캐리어의 공구 길이 옵셋 기능을 조합하는 것은 허용되지 않습니다. 이 두 가지 기능을 동시에 호출하게 되면 오류 메시지가 발생됩니다.

참고

TOFRAME (페이지 368)을 통해 선택 공구 캐리어의 오리엔테이션 방향을 기준으로 프레임을 정의할 수 있습니다.

참고

오리엔테이션 변환이 활성 상태인 경우(3, 4 또는 5축 변환), 알람을 발생시키지 않으면서 오리엔테이션이 제로 위치에서 벗어나는 공구 캐리어를 선택할 수 있습니다.

참고

회전이 없는 활성 프레임에서 TCOFRX/TCOFRY/TCOFRZ를 사용하는 오리엔테이션은 공구 오리엔테이션이 이미 X/Y/Z 방향을 가리키고 있기 때문에 자명한 해가 됩니다.

Z 방향 오리엔테이션(TOFRAME)

G 그룹 53의 TOFRAME 명령(공구 관련 프레임 회전)으로 Z 방향이 선택 공구 캐리어의 오리엔테이션 방향과 평행하게 정렬되는 프레임을 정의할 수 있습니다.

활성인 공구 캐리어가 없거나 오리엔테이션 변경 없이 공구 캐리어가 활성인 경우, 새 프레임의 Z 방향은 다음과 같습니다.

- G17의 기존 Z 방향과 동일
- G18의 기존 Z 방향과 동일
- G19의 기존 Z 방향과 동일

표준 및 측정 싸이클에서 파라미터 전송

표준 및 측정 싸이클의 파라미터 전송에 대해서는 다음에 지정되어 있는 값 범위를 적용하십시오.

각도 값의 경우 다음 범위가 정의됩니다.

회전 중심	값 범위
첫 번째 기하 축	-180° ≤ 각도 ≤ +180°
두 번째 기하 축	-90° ≤ 각도 ≤ +90°
세 번째 기하 축	-180° ≤ 각도 ≤ +180°

참고

각도값을 표준 또는 측정 싸이클로 전송할 때는 다음을 주의 깊게 관찰하여야 합니다.

NC의 계산 정확도보다 작은 값은 0으로 반올림해야 합니다!

각도 위치에 대한 NC의 계산 결과는 머신 데이터에 정의되어 있습니다.

MD10210 $MN_INT_INCR_PER_DEG

4.13.10 기계 측정(CORRTC)에 따라 오리엔테이션 설정이 가능한 공구 캐리어 수정

공구 캐리어의 측정된 키네마틱 체인 요소는 CORRTC 기능을 이용해 특수 교정 요소에 작성할 수 있습니다.

참고

CORRTC 기능으로 작성한 수정 값은 공구 캐리어에 즉시 적용되지 않습니다. 수정 값은 공구 캐리어, NEWCONF의 선택을 취소하고 공구 캐리어를 선택한 후에만 적용됩니다.

구문

```
<_Corr_Status> = CORRTC(<_Corr_Vect>, <_Corr_Index>,
<_Corr_Mode>, [ <_No_Alarm>])
```

의미

CORRTC:	기능 호출		
<_Corr_Status>:	기능 반환값		
	데이터 유형:	INT	
	값:	0	기능이 에러 없이 실행되었습니다.
		1	공구 캐리어가 활성화되지 않습니다.
		2	활성 공구 캐리어가 키네마틱 체인으로 정의되지 않았습니다.
		10	<_Corr_Index> 호출 파라미터가 음수입니다.
		11	<_Corr_Mode> 호출 파라미터가 음수입니다.
		12	서브체인 섹션에 대한 참조가 올바르지 않음(_CORR_INDEX)
		13	_CORR_INDEX 파라미터($TC_CARR_CORR_ELEM)가 참조하는 섹션에서 정의된 수정 요소가 없습니다.
		20	<_CORR_MODE>의 백의 자리가 유효하지 않습니다. 값이 0과 1만 허용됩니다.
		21	<_CORRr_MODE>의 천의 자리가 유효하지 않습니다. 값이 0과 1만 허용됩니다.
		30	옵셋 벡터 수정과 관련하여, 현재 값과의 차이가 최소 한 좌표에서 설정 데이터 SD41612 $SN_CORR_TRAFO_LIN_MAX에 의해 지정된 최대 값을 초과합니다.
		31	시스템 변수를 작성하려는 시도가 작성 권한 없음으로 인해 거부되었습니다.
<_Corr_Vect>:	수정 벡터 수정 벡터 내용은 파라미터 <_Corr_Index>와 <_Corr_Mode>에 의해 정해집니다. If <_Corr_Status> = 30, 벡터의 내용이 덮어쓰기 됩니다(위의 내용 참고).		
	데이터 유형:	REAL	
<_Corr_Index>:	보정 요소의 방향 벡터가 보정될 섹션을 지정합니다.		
	데이터 유형:	INT	

<_Corr_Mode>:	수정 모드		
	데이터 유형:	INT	
	<Corr_Index> 파라미터는 십진수로 코딩됩니다(일의 자리부터 천의 자리까지):		
	일의 자리:	예비	
	십의 자리:	<_Corr_Index> 내용이 참조하는 수정 요소의 수정 방식을 지정합니다.	
		xx**0**x	수정 벡터는 수정 요소에 즉시 작성됩니다. 이 방식은 해당 시스템 데이터($NK_OFF_DIR[<n>, ...])의 인덱스 <n>을 파악할 필요 없이 즉시 수정 요소를 작성하는 데 사용될 수 있습니다.
		xx**1**x	0일 경우, 전송된 수정값이 월드 좌표계에서 해석된다는 것이 차이입니다. 초기 상태의 키네마틱 체인(모든 로터리 축의 위치가 0)이 다른 회전을 포함하는 경우 값 0과 1의 차이가 항상 발생할 수 있습니다.
		xx**2**x	1일 경우, 수정값이 전체 섹션을 참조한다는 것이 차이입니다(예: 전체 섹션이 수정값에 의해 지정된 길이에 도달하도록 수정 요소에 값이 입력됩니다).
	백의 자리:	<_Corr_Vect> 파라미터의 내용 해석 방식을 지정합니다.	
		x**0**xx	전송된 수정 벡터 <_Corr_Vect>는 수정 요소의 전체 새 길이 또는 <_Corr_Index>가 <_Corr_Mode>의 십의 자리와 함께 참조하는 섹션을 포함합니다(절대 수정).
		x**1**xx	전송된 수정 벡터 <_Corr_Vect>은 수정 요소의 현재 길이 대비 편차 또는 <_Corr_Index>가 <_Corr_Mode>의 십의 자리수와 함께 참조한 섹션만 포함합니다(증분 수정).
	천의 자리:	설정 데이터 $SN_CORR_TOCARR_LIN_MAX에 의해 최대 허용 수정이 제한되는지 여부를 결정합니다.	
		0xxx	기준값 모니터링이 활성화됩니다.
		1xxx	기준값 모니터링이 압축됩니다.

<_No_Alarm>:	에러 발생 시 응답(반환값 > 0) (**옵션 사항**)		
	데이터 유형:	BOOL	
	값:	FALSE (기본 설정)	에러가 발생하면 프로그램 실행이 정지되고 알람이 표시됩니다.
		TRUE	에러가 발생해도 프로그램 처리가 중단되지 않고 알람도 표시되지 않습니다. 적용: 반환값에 따른 사용자별 응답 지정

CORRTC에 대한 세부 정보

공구 캐리어의 키네마틱 구조는 관련 원점, 기계 영점 또는 공구 캐리어 원점에서 시작하는 1개(T형 및 P형) 또는 2개(M형) 키네마틱 체인(서브체인)으로 설명됩니다. 2개의 체인 중 1개는 공구 체인 인데 공구의 기준점에서 끝나고, 나머지 체인인 공작물 체인 은 기본 좌표계의 영점에서 끝납니다.

CORRTC 기능은 특수 교정 요소에 공구 캐리어가 있는 기계의 축 방향을 기록합니다. 키네마틱 체인은 $ NK_TYPE을 통해 정의되는 OFFSET 타입의 요소 등으로 설명됩니다.

섹션을 통한 CORRTC 작업

2개의 서브체인은 최대 4개 섹션으로 나뉠 수 있습니다.

- 섹션 1은 체인의 시작점에서 시작하여 첫 번째 로터리 축에서 끝납니다.
- 섹션 2는 로터리 축 1과 로터리 축 2 사이의 섹션입니다.
- 섹션 3은 로터리 축 2와 체인의 끝 부분 사이의 섹션입니다.

아래의 그림에는 로터리 축 2개가 있는 오리엔테이션 가능 공구 캐리어가 나와 있습니다.

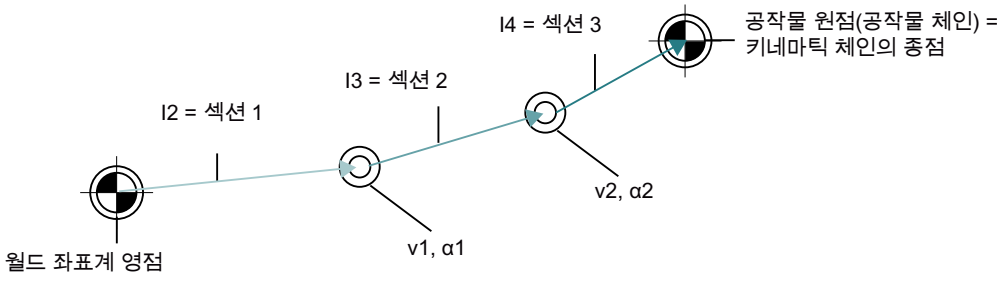


그림 4-12 CORRTC 예시

섹션이 명확하게 정의됩니다. 시작점에서 종료점까지 키네마틱 서브체인이 실행되면 첫 번째 섹션의 인덱스가 1이 되고 다음 섹션의 인덱스가 2가 되는 등 인덱스 번호가 순서대로 매겨집니다.

수정 요소

$TC_CARR_CORR_ELEM [, 0 ... 3] 시스템 변수를 사용하여 이러한 각 섹션에서 일정한 키네마틱 체인 요소($NK_TYPE [<n>] = 'OFFSET')를 참조할 수 있습니다. 기계 측정 중에 결정된 보정 값은 CORRTC 기능을 사용하여 지정된 요소에 작성됩니다.

$TC_CARR_CORR_ELEM [m, 0 ... 3]의 참조 시퀀스는 위에서 설명한 섹션과 일치해야 합니다. 즉, $TC_CARR_CORR_ELEM [m, 0]에는 옵셋 벡터 **I**1에 속하는 체인 요소 하나가 있을 수 있습니다.

기준 값은 항상 CORRTC가 호출될 때 공구 캐리어 활성화에 적용되는 값입니다. 공구 캐리어를 선택한 후 저장된 키네마틱 데이터의 변경된 내용은 CORRTC 기능의 작동 방법에 영향을 주지 않습니다.

4.13.11 온라인 공구 길이 보정(TOFFON, TOFFOF)

시스템 변수 $AA_TOFF[<n>]를 사용하면 실시간으로 3개의 공구 방향에 따른 유효 공구 길이를 3차원으로 오버레이할 수 있습니다.

3개의 기하 축 이름이 색인 <n>으로 사용됩니다. 따라서 활성 방향 옵셋의 번호는 동시에 활성 상태인 기하 축에 의해 결정됩니다.

모든 옵셋은 동시에 활성화할 수 있습니다.

온라인 공구 길이 옵셋 기능은 다음에 사용될 수 있습니다.

- 오리엔테이션 변환 TRAORI
- 오리엔테이션 공구 홀더 TCARR

참고

온라인 공구 길이 옵셋은 **옵션**이며, 사전에 활성화되어 있어야만 합니다. 이 기능은 오리엔테이션 변환 또는 전향적 공구 홀더의 활성화와 연결되어 실행되어야만 합니다.

구문

```
TRAORI
TOFFON(<compensation direction>[,<offset value>])
WHEN TRUE DO $AA_TOFF[<compensation direction>]          ; 동기 동작
```

```
...
TOFFOF(<compensation direction>)
```

의미

TOFFON:	온라인 공구 길이 옵셋 **활성화**	
	<compensation direction>:	온라인 공구 길이 옵션을 실행해야 하는 공구 방향 (X, Y, Z)
	<offset value>:	이를 활성화할 때, 옵셋 값은 보정에 상응하는 방향으로 지정될 수 있으며, 이는 곧바로 복구됩니다.
TOFFOF:	온라인 공구 길이 옵셋 **리셋** 지정된 보정 방향의 보정 값이 리셋되고 전처리 정지가 시작됩니다.	

예

예 1: 공구 길이 보정 선택

프로그램 코드	설명
MD21190 $MC_TOFF_MODE = 1	; 절대치에 도달.
MD21194 $MC_TOFF_VELO[0] =1000	
MD21196 $MC_TOFF_VELO[1] =1000	
MD21194 $MC_TOFF_VELO[2] =1000	
MD21196 $MC_TOFF_ACCEL[0] =1	
MD21196 $MC_TOFF_ACCEL[1] =1	
MD21196 $MC_TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	; 변환 활성화.
N20 TOFFON(Z)	; Z 공구 방향에 온라인 공구 길이 보정 활성화.
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; Z 공구 방향에 TLC 10 보간.
...	
N100 XOFFSET=$AA_TOFF_VAL[X]	; X 방향에서 실제 보정 할당.
N120 TOFFON(X,-XOFFSET) G4 F5	; X 공구 방향에서 TLC가 다시 0으로 감소.

예 2: 공구 길이 옵셋 선택 취소

프로그램 코드	설명
N10 TRAORI(1)	; 변환 활성화.

프로그램 코드	설명
N20 TOFFON(X)	; X 공구 방향에서 온라인 공구 길이 보정 활성화.
N30 WHEN TRUE DO $AA_TOFF[X] = 10 G4 F5	; X 공구 방향에 TLC 10 보간.
...	
N80 TOFFOF(X)	; X 공구 방향의 위치 옵셋 삭제: ...$AA_TOFF[X]=0 축이 이동하지 않습니다. 실제 오리엔테이션에 따라 공작물 좌표계에서 위치 옵셋이 실제 위치에 추가됩니다.

추가 정보

블록 준비

사전 처리에서 블록 준비를 하는 중, 현재 기본 실행에서 공구 길이 옵셋 활성화 역시 고려 사항에 들어갑니다. 최대 허용 축 속도를 최대한 활용할 수 있도록 하려면 공구 옵셋이 설정되는 동안 STOPRE 전처리 정지를 사용하여 블록 준비를 중지시켜야 합니다.

공구 옵셋은 IPO 버퍼가 런인과 기본 실행 사이에 적응될 수 있는 것보다는 프로그램 시작 이후 공구 길이 옵셋이 변경되지 않았거나 공구 길이 옵셋이 변경된 이후 더 많은 블록이 처리되는 런인 시간에 항상 알려집니다.

Variable $AA_TOFF_PREP_DIFF

현재 보간기에 활성인 보정과 블록 준비 중에 적용된 보정 간의 치수 차이는 변수 $AA_TOFF_PREP_DIFF[<n>]에 폴링할 수 있습니다.

머신 데이터 및 설정 데이터 조절

다음 시스템 데이터는 온라인 공구 길이 옵셋에 사용할 수 있습니다.

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE (오버레이 동작에 대한 가속 마진)
- MD21190 $MC_TOFF_MODE
  시스템 변수 $AA_TOFF[<n>]의 내용이 절대치로 옮겨지거나 통합됩니다.
- MD21194 $MC_TOFF_VELO (온라인 공구 길이 옵셋의 속도)
- MD21196 $MC_TOFF_ACCEL (온라인 공구 길이 옵셋의 가속)
- 한계값 사전 설정에 대한 설정 데이터
  :
  SD42970 $SC_TOFF_LIMIT (공구 길이 옵셋 값의 상한값)

4.13.12 회전 공구에 대한 옵셋 데이터 수정

4.13.12.1 오리엔테이션 계산(ORISOLH)

사전 지정된 ORISOLH 기능은 선삭 공구가 공작물에 대해 키네마틱 독립적인 지정 위치에 올 수 있도록 사용자가 기계의 로터리 축 위치를 설정하도록 지원합니다. 사전 조건은 키네마틱 체인으로 파라미터화된 6축 변환이 활성화되어 있는 것입니다.

2가지 기본적인 기능이 제공됩니다.

- 공구 조정
  β와 γ 각도는 지정됩니다. 이 기능은 필요한 3개의 오리엔테이션 축의 각도를 계산합니다.
- 직접 공구 조정
  두 번째 및 세 번째 오리엔테이션 축의 각도는 지정됩니다. 이 기능은 관련 β 및 γ 각도와 여기에 없는 첫 번째 오리엔테이션 각도를 계산합니다.

참고

오리엔테이션 축 순서

공작물부터 공구까지 기계의 구조를 설명하는 키네마틱 체인을 살펴보면 다음 사항이 6축 변환의 오리엔테이션 축 3개의 순서에 적용됩니다.

- **공작물**에 가장 가까운 오리엔테이션 축은 **첫 번째** 오리엔테이션 축입니다.
- **공구**에 가장 가까운 오리엔테이션 축은 **세 번째** 오리엔테이션 축입니다.

일반적으로 첫 번째 오리엔테이션 축은 스핀들이기 때문에 이 경우 해당 회전은 회전 프레임을 통해 구현됩니다.

구문

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

의미

ORISOLH:	기능 호출

<RetVal>:	기능 반환값		
	데이터 유형:	INT	
	값 범위:	0, -2, -3, ..., -17	
	값:	0	기능이 에러 없이 종료되었습니다.
		-2	유효한 변환(6축 오리엔테이션 변환)이 활성화되어 있지 않습니다.
		-3	첫 번째 파라미터(<Cntrl>)가 음수입니다.
		-4	첫 번째 파라미터(<Cntrl>)의 일의 자리가 유효하지 않습니다. 값이 0과 1만 허용됩니다.
		-5	첫 번째 파라미터(<Cntrl>)의 십의 자리가 유효하지 않습니다. 값이 0~3만 허용됩니다.
		-6	첫 번째 파라미터(<Cntrl>)의 백의 자리가 유효하지 않습니다. 값이 0과 1만 허용됩니다.
		-7	첫 번째 파라미터(<Cntrl>)의 천의 자리가 유효하지 않습니다. 값이 0~3만 허용됩니다.
		-8	γ 각도가 "Direct tool alignment(직접 공구 정렬)" 기능을 사용하기에 너무 큽니다.
		-9	지정된 축 위치 중 하나 이상이 "Direct tool alignment" 기능의 축 한계를 벗어납니다.
		-10	활성 공구가 없습니다.
		-11	요청된 오리엔테이션을 설정할 수 없습니다.
		-12	Hirth 조인트를 위한 자유 축 각도 조절이 첫 번째 또는 단일 솔루션에 가능하지 않습니다.
		-13	Hirth 조인트를 위한 자유 축 각도 조절이 두 번째 솔루션에 가능하지 않습니다.
		-14	Hirth 조인트를 위한 자유 축 각도 조절이 두 솔루션 중 어느 것에도 가능하지 않습니다.
		-15	첫 번째 오리엔테이션 축이 Hirth 축으로 파라미터 지정됩니다.

		-16	두 번째와 세 번째 로터리 축이 Hirth 축으로 파라미터 지정되었습니다. 두 축 중 하나만 Hirth 축이 될 수 있습니다.
		-17	지정된 축 위치 하나 이상이 "Swivel directly(직접 스위블)" 기능을 위한 관련 Hirth 조인트와 호환되지 않습니다.

<Cntrl>:	기능의 동작 제어.		
	데이터 유형:	INT	
	<Cntrl> 파라미터는 십진수로 코딩됩니다(일의 자리부터 천의 자리까지):		
	일의 자리:	일의 자리는 에러에 대한 반응을 제어합니다.	
		xxx**0**	에러(반환값 < 0)가 발생하면, 알람 14106이 출력되고 프로그램 처리가 중단됩니다. **주의:** 알람은 <Cntrl> 파라미터가 음수일 때 일의 자리값과 상관없이 출력되기도 합니다.
		xxx**1**	에러(반환값 < 0)가 발생하면 출력되는 알람이 **없습니다**. 사용자가 프로그램에서 적절하게 대응합니다.
	십의 자리:	Hirth 조인트가 있는 오리엔테이션 축이 존재할 때의 동작을 제어합니다. **주의:** 이 파라미터는 "Tool alignment(공구 정렬)" 기능을 위해서만 평가됩니다(예: 백의 자리값이 "0"일 때).	
		xx**0**x	축 위치는 가장 가까운 자리로 반올림됩니다.
		xx**1**x	축 위치는 β 각도와 프로그램된 해당 값 사이 차이가 최소가 되도록 반올림됩니다.
		xx**2**x	축 위치는 β 각도가 프로그램된 값보다 적은 가능한 최대 값이 되도록 반올림됩니다(β는 내림 처리).
		xx**3**x	축 위치는 β 각도가 프로그램된 값보다 큰 가능한 최소 값이 되도록 반올림됩니다(β는 올림 처리).
	백의 자리:	실행할 기능과 다음 두 파라미터 <W1>과 <W2>의 의미를 지정합니다.	
		x**0**xx	"Tool alignment(공구 조정)" 기능 <W1>과 <W2> 파라미터는 다음을 의미합니다. • <W1> = β • <W2> = γ 오리엔테이션 축의 관련 각도가 계산됩니다.
		x**1**xx	"Direct tool alignment(직접 공구 조정)" 기능 <W1>은 6축 변환의 두 번째 오리엔테이션 축의 위치 지정이고 <W2>는 세 번째 오리엔테이션 축의 위치 지정입니다. 첫 번째 오리엔테이션 축의 위치와 β 및 γ 각도는 두 위치 지정값과 호환되도록 지정됩니다.

			에러가 발생하지 않으면, 두 솔루션은 항상 $P_ORI_POS[<n>, <m>] 시스템 변수에 출력됩니다. 첫 번째 인덱스 <n> (0 또는 1)은 솔루션을 참조하고 두 번째 인덱스 <m> (0 ... 2)은 오리엔테이션 축을 참조합니다. • $P_ORI_POS[0/1, **0**]: 첫 번째 오리엔테이션 축의 위치 • $P_ORI_POS[0/1, **1**]: 각도 β • $P_ORI_POS[0/1, **2**]: 각도 γ 위치 지정 <W1>과 <W2>가 Hirth 조인트 또는 활성 소프트웨어 한계와 호환되는지 확인됩니다. 그렇지 않을 경우, 해당 오류 번호가 반환됩니다(<RetVal> 파라미터 참조). 각도 <W1> 및 <W2>가 임의로 선택되면 일반적으로 공구의 절삭날이 가공 평면에 있지 않습니다. 절삭날이 회전하여 가공 평면에서 벗어나는 각도 γ는 설정 데이터 SD42999 $SC_ORISOLH_INCLINE_TOL에 의해 정의되는 한계 값보다 커서는 안 됩니다.
	천의 자리:	백의 자리값이 "0"인 경우 솔루션의 어떤 위치를 수정할지 지정합니다(예: "Tool alignment" 기능).	
		0xxx	계산된 축 위치는 현재 기계 축 위치에 최대한 가까워야 합니다.
		1xxx	모듈로 축의 계산된 축 위치는 모듈로 범위의 중간에 최대한 가까워야 하고, 다른 축의 경우 0에 최대한 가까워야 합니다. 비모듈로 축의 경우, 이것은 축 위치가 -180° ... +180° 범위로 감소한다는 것을 의미합니다.
		2xxx	계산된 축 위치는 축 유형과 상관없이 -180° ... +180° 범위로 감소해야 합니다.
<W1>:	첫 번째 각도 의미는 <Cntrl> 파라미터의 백의 자리의 결과입니다.		
	데이터 유형:	REAL	
<W1>:	두 번째 각도 의미는 <Cntrl> 파라미터의 백의 자리의 결과입니다.		
	데이터 유형:	REAL	

참고

프로그램되지 않은 파라미터는 기본 값 "0"을 가집니다.

추가 정보

ORISOLH 기능 실행 시 추가 상태 정보와 함께 발견되는 솔루션 개수는 다음 시스템 변수를 통해 읽을 수 있습니다.

시스템 변수	의미		
$P_ORI_POS [<n>, <m>]	오리엔테이션 프로그래밍에서 산출된 오리엔테이션 축의 각도를 반환합니다.		
	<n>:	솔루션 인덱스	
		값 범위:	0, 1
	<m>:	오리엔테이션 축의 인덱스	
		값 범위:	0 ... 2
			오리엔테이션 축의 순서(1 ... 3)는 $NT_ROT_AX_NAME에 있는 축 정의를 참조합니다.
	ORISOLH 기능이 "Direct tool alignment(직접 공구 정렬)" 모드에서 호출되면, $P_ORI_POS[0/1, 1]와 P_ORI_POS[0/1, 2] 변수는 두 솔루션에 속한 두 각도 β와 γ 값을 포함합니다. $P_ORI_POS[<n>, <m>]에 입력된 첫 번째 솔루션(예: 인덱스 <n> = 0 이용)은 항상 요청된 오리엔테이션이 직접 접근될 때 컨트롤 시스템으로 선택된 솔루션입니다. 두 번째 인덱스 <m>은 오리엔테이션 축을 의미합니다(예: $NT_ROT_AX_NAME). $P_ORI_POS[<n>, <m>]에 입력된 축 위치는 $NK_OFF와 $NK_OFF_FINE에 입력된 옵셋을 고려하고, 이러한 축 각도는 다음 블록에서 추가 수정 없이 요청된 오리엔테이션을 설정할 때 사용될 수 있습니다. 로터리 축이 Hirth 축이면, 솔루션 위치는 Hirth 축의 나머지 부분의 가장 가까운 위치로 반올림됩니다. Hirth 조인트 연결 로터리 축의 경우, 정확한 솔루션의 축 위치들과 $P_ORI_DIFF 시스템 변수의 Hirth 증분에 조정된 솔루션의 축 위치들 간 차이를 읽을 수 있습니다.		

시스템 변수	의미	
$P_ORI_DIFF [<n>, <m>]	오리엔테이션 축의 정확한 위치들과 오리엔테이션 프로그래밍에서 나온 $P_ORI_POS에서 제공하는 위치들 간 차이를 반환합니다.	
	<n>: 솔루션 인덱스	
	값 범위:	0, 1
	<m>: 오리엔테이션 축의 인덱스	
	값 범위:	0 ... 2
		오리엔테이션 축의 순서(1 ... 3)는 $NT_ROT_AX_NAME에 있는 축 정의를 참조합니다.
	그 내용은 위치가 증분(Hirth 조인트)될 때만 0이 아닐 수 있습니다. 즉, 관련 축의 시스템 데이터 $NT_HIRTH_INCR은 이 축이 매뉴얼 로터리 축일 때 0이 아닙니다.	

시스템 변수	의미		
$P_ORI_SOL	둘 이상의 오리엔테이션 축이 있는 오리엔테이션 변환에서 축 각도가 특정 오리엔테이션에 이르도록 계산되는 경우, 일반적으로 여러 개의 솔루션이 있습니다. $P_ORI_SOL 시스템 변수는 유효한 솔루션의 개수와 추가 상태 정보를 포함합니다. $P_ORI_SOL 내용은 다음과 같이 코딩됩니다.		
	값 < 0	일반 오류 상태	
		-1	활성 변환을 위해 아직 계산된 솔루션이 없습니다 (ORISOLH의 호출 없음).
		-2	변환이 활성화되지 않았거나 활성 변환이 지정된 오리엔테이션 프로그래밍을 위해 위치를 제공할 수 있는 오리엔테이션 변환(6축 변환)이 아닙니다.
		-4	원하는 오리엔테이션을 현재 키네마틱으로 설정할 수 없습니다.
		-5	ORISOLH 기능이 "Direct tool alignment(직접 공구 정렬)" 모드에서 호출되었을 때 발견된 솔루션이 없습니다.
		-6	"Direct tool alignment(직접 공구 정렬)" 모드에서 ORISOLH 기능이 호출되었을 때 각도 γ가 너무 큽니다.
		-7	"Direct tool alignment(직접 공구 정렬)" 모드에서 ORISOLH 기능이 호출되었을 때 Hirth 조인트 때문에 설정할 수 없는 각도가 지정되었습니다.
		-8	첫 번째 오리엔테이션 축(프레임 축)은 Hirth 축으로 파라미터 지정되지 않아야 합니다.
		-9	두 번째와 세 번째 로터리 축이 Hirth 축으로 파라미터 지정되었습니다. 두 축 중 하나만 Hirth 축이 될 수 있습니다.
		-10	Hirth 조인트에 대한 솔루션 조정이 발견되지 않았습니다.
	값 > 0 일의 자리	축 한계와 에러 상태를 고려하지 않고 수학적으로 가능한 솔루션의 수	
		0	솔루션이 없습니다. 즉, 요청된 오리엔테이션을 설정할 수 없습니다. 이러한 경우에는 3가지 원인이 있을 수 있습니다. • 원칙적으로 요청된 오리엔테이션은 기계 좌표계로 인해 달성될 수 없습니다(오리엔테이션 축을 올바

시스템 변수	의미		
			른 각도로 맞출 수 없음). 오리엔테이션 축의 임의 이송 범위를 이용하더라도 마찬가지입니다. 이 경우, $P_ORI_SOL의 십의 자리와 백의 자리가 모두 0이고 오리엔테이션 축에 할당된 $P_ORI_STAT 상태 변수는 값이 "-4"입니다. • 계산된 솔루션은 축 한계를 벗어나기 때문에 달성될 수 없습니다. 축 한계 없이 결과로 도출되는 오리엔테이션 축의 위치는 $P_ORI_POS에서 읽을 수 있습니다. • 축 위치는 ORISOLH 기능이 "Direct tool alignment" 모드에서 호출되었을 때 지정되었습니다. 그러면 축의 위치가 계산될 첫 번째 오리엔테이션 축과 평행으로 정렬되는 공구의 오리엔테이션 벡터나 오리엔테이션 법선 벡터가 나옵니다. 이 축의 위치는 이 경우에 지정되지 않습니다.
		1	솔루션이 있습니다. 이러한 경우에는 3가지 원인이 있을 수 있습니다. • 축 한계를 고려하지 않더라도 지정된 오리엔테이션과 기계 좌표계를 기준으로 하나의 솔루션만 있습니다(수학적 관점에서는 2개의 동시 솔루션). 이러한 경우는 올바른 각도에 있지 않은 좌표계의 오리엔테이션 범위의 가장자리에서 발생합니다. $P_ORI_POS는 두 솔루션 모두(동일함) 포함합니다. • 축 한계 이탈로 인해 두 번째 솔루션이 유효하지 않기 때문에 하나의 솔루션만 있습니다. 유효한 솔루션은 항상 $P_ORI_POS에서 첫 번째 솔루션입니다. 축 한계를 고려하지 않을 때 나오는 두 번째 솔루션도 $P_ORI_POS에서 읽을 수 있습니다. • 이것은 "Direct tool alignment(직접 공구 정렬)" 모드에서 ORISOLH 기능이 호출될 때 일반적인 경우입니다. 두 오리엔테이션 축의 지정된 축 위치의 경우, 계산될 누락 오리엔테이션 축의 유효한 위치는 일반적으로 1개입니다.
		2	2개의 솔루션이 있습니다.
		8	무한개의 솔루션이 있습니다. 즉, 오리엔테이션 축의 위치(극축)가 임의적입니다. 그러나 다른 축의 2개의 가능한 솔루션에서는 축 한계 이탈로 인해 1개가 제외됩니다.

시스템 변수	의미		
		9	무한개의 솔루션이 있습니다. 즉, 오리엔테이션 축의 위치(극축)가 미정입니다. 미정 축은 백의 자리나 $P_ORI_STAT 시스템 변수로 파악할 수 있습니다.
	값 > 0 십의 자리	축 한계 이탈 비트 코딩 표시 오류의 정확한 원인은 $P_ORI_STAT 시스템 변수를 통해 파악할 수 있습니다.	
		비트 0 (값 10):	1개 이상의 솔루션에 대해 첫 번째 오리엔테이션 축의 1개 이상의 축 한계가 이탈되었습니다.
		비트 1 (값 20):	1개 이상의 솔루션에 대해 두 번째 오리엔테이션 축의 1개 이상의 축 한계가 이탈되었습니다.
		비트 2 (값 40):	1개 이상의 솔루션에 대해 세 번째 오리엔테이션 축의 1개 이상의 축 한계가 이탈되었습니다.
	값 > 0 백의 자리	미정 축 위치의 비트 코딩 표시(무한 개수의 솔루션이 있는 경우에만 발생, 예: 일의 자리가 "9"일 때).	
		비트 0 (값 100):	첫 번째 오리엔테이션 축의 위치가 정의되지 않았습니다.
		비트 1 (값 200):	첫 번째 오리엔테이션 축의 위치가 정의되지 않았습니다.
		비트 2 (값 400):	세 번째 오리엔테이션 축의 위치가 정의되지 않았습니다.
	첫 번째, 두 번째, 세 번째 오리엔테이션 축의 지정은 $NT_ROT_AX_NAME에 있는 축 정의를 참조합니다.		

시스템 변수	의미		
$P_ORI_STAT [<n>]	ORISOLH가 호출된 후 최대 세 오리엔테이션 축의 각각에 대한 상태를 반환합니다.		
	<n>:	오리엔테이션 축의 인덱스 ($NT_ROT_AX_NAME에 있는 관련 오리엔테이션 축의 인덱스에 해당)	
		값 범위:	0 ... 2
			오리엔테이션 축의 순서(1 ... 3)는 $NT_ROT_AX_NAME에 있는 축 정의를 참조합니다.
	$P_ORI_STAT의 내용은 다음과 같이 코딩됩니다.		
	값 < 0	일반 오류 상태	
		-1	상태가 정의되지 않았습니다(ORISOLH 호출 없음).
		-2	변환이 활성화되지 않았거나 활성 변환이 지정된 오리엔테이션 프로그래밍을 위해 위치를 제공할 수 있는 오리엔테이션 변환(6축 변환)이 아닙니다.
		-3	축이 활성 변환에 포함되어 있지 않습니다.
		-4	축의 임의 이송 범위를 이용하더라도 현재 좌표계에서 요청된 오리엔테이션을 달성할 수 없기 때문에 축의 위치를 계산할 수 없습니다.
		-5	축 위치는 ORISOLH 기능이 "Direct tool alignment" 모드에서 호출되었을 때 지정되었습니다. 그러면 축의 위치가 계산될 첫 번째 오리엔테이션 축과 평행으로 정렬되는 공구의 오리엔테이션 벡터나 오리엔테이션 법선 벡터가 나옵니다. 이 축의 위치는 이 경우에 지정되지 않습니다.
		-6	"Direct tool alignment(직접 공구 정렬)" 모드에서 ORISOLH 기능이 호출되었을 때 각도 γ가 너무 큽니다.
		-7	"Direct tool alignment(직접 공구 정렬)" 모드에서 ORISOLH 기능이 호출되었을 때 Hirth 조인트 때문에 설정할 수 없는 각도가 지정되었습니다.
		-8	첫 번째 오리엔테이션 축(프레임 축)은 Hirth 축으로 파라미터 지정되지 않아야 합니다.
		-9	두 번째와 세 번째 로터리 축이 Hirth 축으로 파라미터 지정되었습니다. 두 축 중 하나만 Hirth 축이 될 수 있습니다.

시스템 변수	의미		
		-10	Hirth 조인트에 대한 솔루션 조정이 발견되지 않았습니다.
	값 > 0 일의 자리	첫 번째 솔루션의 축 한계 이탈 비트 코딩 표시	
		비트 0 (값 1):	첫 번째 솔루션이 축 하한값을 이탈합니다.
		비트 1 (값 2):	첫 번째 솔루션이 축 상한값을 이탈합니다.
	값 > 0 십의 자리	두 번째 솔루션의 축 한계 이탈 비트 코딩 표시	
		비트 0 (값 10):	두 번째 솔루션이 축 하한값을 이탈합니다.
		비트 1 (값 20):	두 번째 솔루션이 축 상한값을 이탈합니다.
	값 > 0 백의 자리	미정의 축 위치 표시.	
		비트 0 (값 100):	오리엔테이션 축의 위치가 정의되지 않았습니다. 즉, 로터리 축의 각 임의 설정(극 위치)으로 요청된 오리엔테이션이 달성됩니다. 이 정보는 $P_ORI_SOL 시스템 변수에도 있습니다.
	축 한계 이탈을 나타내는 오류 번호 중 여러 개가 동시에 발생할 수 있습니다. 축 한계를 벗어나면 360°의 배수를 가감하여 허용 축 한계 내의 위치에 도달하려는 시도가 이루어집니다. 이것이 불가능하면 벗어난 값이 하한값인지 상한값인지 파악할 수 없습니다. 요청된 오리엔테이션에 대한 솔루션이 없으면($P_ORI_SOL = 0), 변환에서 오리엔테이션 축의 상태가 "0"입니다.		

참고

$NT_ROT_AX_NAME

이 시스템 변수는 오리엔테이션 설정에 사용된 최대 3개의 축을 가리킵니다. 키네마틱 변환으로 인한 오리엔테이션 이동을 수행해야 하는 기계 축(로터리 축)을 정의하는 체인 요소($NK_NAME)의 이름을 포함하고 있습니다. 최대 3개의 로터리 축이 이 시스템 변수에 포함되는 순서는 기계 좌표계에 대해 무관합니다. 이것은 키네마틱 체인의 구조에 기인하기 때문입니다. 그러나 다른 변수가 로터리 축에 액세스하는 순서를 정의하기 때문에 $NT_ROT_AX_NAME에서의 오리엔테이션 축 순서는 키네마틱 설명과 일치해야 합니다.

참고

상태 정보

오리엔테이션이 달성 불가능하다거나 관련 축 한계를 벗어난 경우에만 달성 가능하다는 등의 상태 정보는 NC 알람을 발생시키지 않습니다. 표시된 상태에 적합하게 대응하는 것은 사용자의 책임입니다.

4.13.12.2 회전식 공구에 대한 옵셋 데이터 수정 활성화(CUTMOD, CUTMODK)

회전식 공구의 옵셋 데이터 수정은 CUTMOD(오리엔테이션 가능한 공구 캐리어와 함께) 또는 CUTMODK 언어 명령(키네마틱 체인을 통해 지정된 오리엔테이션 변환의 경우)을 통해 NC 프로그램에서 활성화됩니다.

참고

오리엔테이션 가능 공구 캐리어와 키네마틱 체인을 통해 지정된 오리엔테이션 변환은 동시에 활성화될 수 없기 때문에 두 방식 사이에 충돌은 없습니다.

구문

```
CUTMOD = <Value>
```

또는

```
CUTMODK = <Command>
```

의미

CUTMOD:	오리엔테이션 가능 공구 캐리어와 함께 기능 호출

<Value>:	할당된 값		
	데이터 유형:	INT	
	값:	0	함수가 중지됩니다. 시스템 변수 $P_AD... 에서 공급된 값은 상응하는 공구 파라미터와 동일합니다.
		> 0	지정 번호의 오리엔테이션 가능 공구 캐리어가 활성화되면 이 기능이 활성화됩니다. 즉, 이 활성화가 특정 오리엔테이션 공구 캐리어에 연결되어 있습니다. 시스템 변수 $P_AD... 에서 공급되는 값은 활성화된 회전에 의해 정해지는 해당 공구 파라미터에 의해 수정될 수 있습니다. 지정된 오리엔테이션 가능 공구 캐리어가 비활성화되면 이 기능이 일시적으로 비활성화됩니다. 다른 오리엔테이션 가능 공구 캐리어가 활성화되면 이 기능이 영구적으로 비활성화됩니다. 첫 번째의 경우에서 동일한 오리엔테이션 가능 공구 캐리어를 다시 선택하면 기능이 다시 활성화되는 것도 이 때문입니다. 두 번째의 경우 새로 선택해야만 합니다. 이후 특정 번호의 오리엔테이션 가능 공구 캐리어가 다시 활성화된 경우라도 마찬가지입니다. 이 기능은 리셋의 영향을 받지 않습니다.
		-1	오리엔테이션 가능 공구 캐리어가 활성화되면 이 기능이 항상 활성화됩니다. 공구 캐리어를 변경하거나 이를 선택 취소하고 난 후 새로 선택을 하면 CUTMOD를 다시 설정하지 않아도 됩니다.
		-2	현재 활성화된 오리엔테이션 가능 공구 캐리어와 동일한 번호의 오리엔테이션 가능 공구 캐리어가 활성화되면 이 기능이 항상 활성화됩니다. 오리엔테이션 가능 공구 캐리어가 활성화되어 있지 않으면, CUTMOD=0과 같은 의미입니다. 오리엔테이션 가능 공구 캐리어가 활성화되어 있으면 실제 공구 캐리어 번호를 직접 지정하는 것과 같은 의미입니다.
		< -2	2보다 작은 값은 무시됩니다. 즉, 이 경우는 CUTMOD가 프로그래밍되지 않은 것처럼 취급됩니다.

			주의: 이 값 범위는 이후 확장을 대비한 범위이므로 사용해서는 안 됩니다.
CUTMODK:	키네마틱 체인을 통해 지정된 오리엔테이션 변환과 함께 기능 호출		
<Command>:	할당된 명령		
	데이터 유형:	STRING	
	값:	"NEW"	"옵셋 데이터 수정"과 관련하여 키네마틱 체인을 통해 지정된 활성 변환 상태, 변환 이름, 현재 형상 프레임이 저장됩니다. **주의:** 이 명령은 적합한 변환(TRAORI_DYN, TRAORI_STAT 또는 TRAANG_K)이 활성인 경우에만 허용됩니다.
		"OFF"	활성 "옵셋 데이터 수정"을 비활성화합니다. "NEW"를 이용해 이전에 저장된 데이터는 유지됩니다. **주의:** 이 명령 또한 CUTMODK가 활성 상태가 아닐 때 허용됩니다. 그래야 이것이 영향을 주지 않습니다. "옵셋 데이터 수정"을 위해 존재하는 데이터 세트는 유지됩니다.
		"ON"	이 명령으로 "옵셋 데이터 수정"이 "NEW"명령으로 이전에 저장되었던 데이터 세트와 함께 다시 활성화됩니다. 이 명령이 실행될 때 저장된 데이터 세트 이름의 변환이 활성 상태이면 "옵셋 데이터 수정"이 즉시 적용됩니다. 그렇지 않으면 활성화가 활성 변환이 활성화될 때까지 지연됩니다.
		"CLEAR"	"OFF" 명령과 마찬가지로 "옵셋 데이터 수정"을 비활성화하고 저장된 데이터 세트도 삭제합니다. **주의:** 이 명령 또한 CUTMODK가 활성 상태가 아닐 때 허용됩니다.

참고

SD42984 $SC_CUTDIRMOD

CUTMOD 또는 CUTMODK 명령은 설정 데이터 SD42984 $SC_CUTDIRMOD를 사용해 활성화될 수 있는 기능을 대체합니다. 하지만 이 기능은 변경되지 않은 채로 계속 사용 가능합니다. 그러나 두 기능을 동시에 사용하는 것은 타당하지 않기 때문에 CUTMOD가 0이고 CUTMODK가 제로 문자열인 경우에만 이것이 활성화될 수 있습니다.

추가 정보

수정된 옵셋 데이터 읽기

수정된 옵셋 데이터는 다음 시스템 변수와 OPI 변수에서 제공됩니다.

의미	시스템 변수	OPI 변수
절삭날 위치	$P_AD[2]	cuttEdgeParam2
홀더 각	$P_AD[10]	cuttEdgeParam10
절삭 방향	$P_AD[11]	cuttEdgeParam11
공차 각도	$P_AD[24]	cuttEdgeParam24

데이터는 "회전식 공구에 대한 옵셋 데이터 수정" 기능이 CUTMOD 또는 CUTMODK 명령으로 활성화되고 오리엔테이션 가능 공구 캐리어나 적합한 오리엔테이션 변환에 의해 공구가 회전되었을 때 항상 해당 공구 파라미터($TC_DP2[..., ...] 등)와 관련하여 수정됩니다.

기능 관련 추가 시스템 변수

시스템 변수	의미
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	공구가 활성 가공 평면에서 회전된 각도와 CUTMOD 및 CUTMODK 기능에 사용 가능한 수정된 절삭날 데이터를 반환합니다.
$P_CUTMOD / $AC_CUTMOD	CUTMOD 명령을 사용하여 마지막으로 프로그램된 현재 유효 값을 읽습니다(옵셋 데이터 수정이 활성화되어야 하는 공구 캐리어 번호). 마지막으로 프로그램된 값이 CUTMOD = -2(현재 활성인 오리엔테이션 가능 공구 캐리어로 활성화)이면, 값 "-2"가 시스템 변수로 반환되지 않고 프로그래밍 당시 활성이었던 오리엔테이션 가능 공구 캐리어 번호가 반환됩니다.

시스템 변수	의미
$P_CUTMODK / $AC_CUTMODK	"옵셋 데이터 수정"에 대한 현재 유효 데이터 세트 생성 시 사용된 변환 이름을 읽습니다.
$P_CUT_INV / $AC_CUT_INV	공구가 회전하여 회전의 스핀들 방향이 전도되어야 하는 경우에는 TRUE 값을 공급하십시오. 이를 위해서는 읽기 작업에서 참조해야 하는 블록이 다음의 4가지 조건을 충족해야 합니다. 1. 선삭 공구 또는 연삭 공구가 활성화된 경우 (공구 종류 400~599 및/또는 SD42950 $SC_TOOL_LENGTH_TYPE = 2). 2. 옵셋 데이터의 수정이 CUTMOD 또는 CUTMODK 명령으로 활성화되었습니다. 3. 오리엔테이션 가능 공구 캐리어 또는 키네마틱 체인으로 지정된 오리엔테이션 변환이 활성 상태입니다(CUTMOD 또는 CUTMODK 명령으로 선택됨). 4. 오리엔테이션 가능 공구 캐리어 또는 키네마틱 오리엔테이션 변환에 의해 공구가 회전되어 공구 절삭날의 결과 법선이 최초 위치에 대해 90° 이상(일반적으로 180°) 회전됩니다. 지정된 4개의 조건 중 하나 이상 충족되지 않으면 변수가 FALSE 값을 반환합니다. 절삭날 위치가 정의되지 않은 변수 값은 항상 FALSE입니다.

시스템 변수	의미	
$P_CUTMOD_ERR	마지막 CUTMOD 기능 호출 뒤 에러 상태 CUTMOD 기능은 공구 변경을 위해 암시적으로 호출될 수도 있습니다. 리셋 시 변수는 0으로 리셋됩니다. 공구 변경 때마다 리셋되고 필요한 경우 재작성됩니다. 변수는 비트 코딩됩니다. 각 비트는 다음을 나타냅니다.	
	비트 0:	활성 공구에 대해 유효한 절삭 방향이 지정되어 있지 않습니다.
	비트 1:	활성 공구의 절삭날 각도(공차 각도 및 홀더 각도)가 모두 0입니다.
	비트 2:	활성 공구의 공차 각도에 허용 불가 값(< 0° 또는 > 180°)이 있습니다.
	비트 3:	활성 공구의 홀더 각도에 허용 불가 값(< 0° 또는 > 90°)이 있습니다.
	비트 4:	활성 공구의 플레이트 각도에 허용 불가 값(< 0° 또는 > 90°)이 있습니다.
	비트 5:	활성 공구의 절삭날 위치 및 홀더 각도 조합이 허용되지 않는 값입니다(절삭날 위치 1부터 4까지의 경우 홀더 각도는 ≤ 90°, 5부터 8까지는 홀더 각도가 ≥ 90°여야 함).
	비트 6:	활성 공구의 위반 회전 공구가 활성 가공 평면에서 ± 90° 벗어나서 회전했습니다(약 1°의 공차). 따라서 절삭날 위치가 가공 평면에 더 이상 정의되어 있지 않습니다.
	비트 7:	절삭 플레이트가 가공 평면에 있지 않고 절삭 플레이트와 가공 평면 사이의 각도가 설정 데이터 SD42998 $SC_CUTMOD_PLANE_TOL로 지정한 상한을 초과합니다.
	비트 8:	절삭 플레이트가 가공 평면에 있지 않습니다. 각도 α가 1°보다 큽니다. 각도 α는 각도 β의 회전 축과 각도 γ의 회전축에 수직인 좌표축 주위 회전 각도입니다(G18의 X 축).

$P_...: 전처리 변수

$AC_...: 메인 실행 변수

모든 메인 실행 변수는 동기화된 동작으로 읽을 수 있습니다. 사전 처리 작업에서 읽기 액세스 작업은 전처리 정지를 생성합니다.

평면 변경

수정된 절삭날 위치, 절삭 방향 및 홀더나 공차 각도를 결정하기 위해서는 활성 평면(G17 - G19) 에서 절삭날의 평가가 중요합니다.

하지만 설정 데이터 SD42940 $SC_TOOL_LENGTH_CONST (평면 선택 시 공구 길이 콤포넌트 변경) 가 0이 아닌 유효한 값 (+/- 17, 18 또는 19) 을 가진 경우 이 데이터의 내용이 수량을 평가하는 평면을 정의합니다.

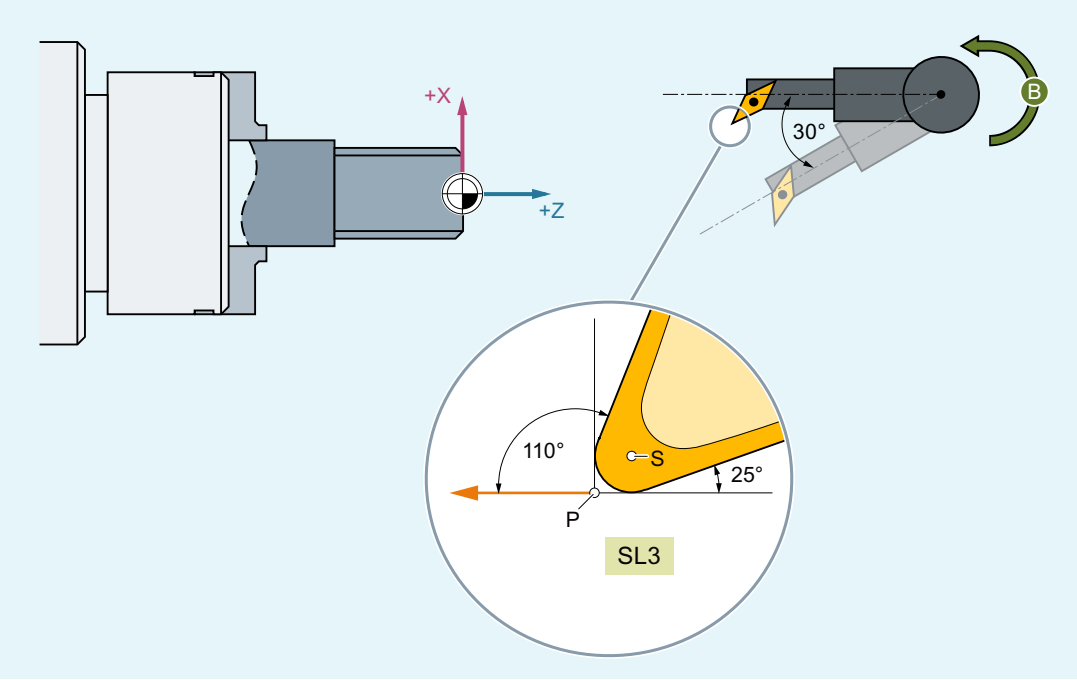
G 코드보다 설정 데이터에 우선순위를 주는 규칙은 머신 데이터 $MC_TOOL_PARAMETER_DEF_MASK의 비트 18을 설정하여 비활성화할 수 있습니다. 즉, 이 비트가 설정되면 그룹 6의 G 명령으로 정의된 평면이 여전히 유효하게 됩니다.

수정된 절삭 데이터의 효율성

수정된 절삭날 위치와 수정된 절삭날 원점은 공구가 이미 활성화된 경우에도 프로그래밍을 하는 즉시 적용됩니다. 이를 위해 공구를 다시 선택할 필요는 없습니다.

예

절삭날 위치가 3인 공구와 B축을 중심으로 공구를 회전할 수 있는 오리엔테이션 가능 공구 캐리어의 경우 공구 회전 후 CUTMOD 명령을 사용하여 절삭날 위치를 수정해야 합니다.



S: 마지막 위치

P: 절삭날 원점

SL: 절삭날 위치

프로그램 코드	설명
N10 $TC_DP1[1,1]=500	
N20 $TC_DP2[1,1]=3	; 절삭날 위치
N30 $TC_DP3[1,1]=12	
N40 $TC_DP4[1,1]=1	
N50 $TC_DP6[1,1]=6	
N60 $TC_DP10[1,1]=110	; 홀더 각도
N70 $TC_DP11[1,1]=3	; 절삭 방향
N80 $TC_DP24[1,1]=25	; 공차 각도
N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0	; B 축
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0 $TC_CARR12[2]=1	; C축
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	

주석에 있는 숫자값은 기계 좌표계에서 X → Y → Z 순서로 블록 위치의 끝을 지정합니다.

설명

N180 블록에서, 먼저 CUTMOD=0에 대해 공구가 선택되고, 오리엔테이션할 수 있는 비회전 공구 홀더가 선택됩니다. 오리엔테이션될 수 있는 공구 홀더의 모든 옵셋 벡터는 0이므로 $TC_DP3[1,1]와 $TC_DP4[1,1]에 지정된 공구 길이에 해당하는 위치가 접근됩니다.

B축을 중심 30° 회전으로 오리엔테이션할 수 있는 공구 홀더가 N200 블록에서 활성화됩니다. CUTMOD=0으로 인해 절삭날 위치가 수정되지 않기 때문에 이전과 마찬가지로 기존 절삭날 원점이 사용됩니다. N210 블록에서 기존 공구 노즈 원점을 0에서 유지하는 위치에 접근하는 이유도 이 때문입니다 (즉, 벡터 (1, 12)가 Z/X 평면에서 30° 회전).

N260 블록에서는 N200과 달리, CUTMOD=2가 적용됩니다. 오리엔테이션 가능한 공구 홀더 회전의 결과 수정된 절삭날 위치는 8이 됩니다. 그에 따라 축 위치도 달라집니다.

공구 반경 보정(TRC)는 N220 및/또는 N270 블록에서 활성화됩니다. 두 프로그램 섹션에서 절삭날 위치가 달라도 TRC가 활성화되어 있는 블록의 끝 위치에는 영향을 주지 않습니다. 따라서 이 위치들은 동일합니다. 서로 다른 절삭날 위치는 N260 및/또는 N300 블록을 선택 취소했을 때만 다시 적용됩니다.

4.13.13 공구 환경에 대한 작업

함수 개요

- 공구 환경 저장(TOOLENV) (페이지 911)
- 도구 환경 삭제(DELTOOLENV) (페이지 914)
- T, D 및 DL 번호 읽기(GETTENV) (페이지 915)
- 공구 길이 및/또는 공구 길이 컴포넌트 읽기(GETTCOR) (페이지 917)
- 공구 컴포넌트 변경(SETTCOR) (페이지 925)

시스템 변수 개요

- 저장된 공구 환경에 대한 정보 읽기($P_TOOLENVN, ($P_TOOLENV) (페이지 916)

4.13.13.1 공구 환경 저장(TOOLENV)

TOOLENV 기능은 메모리에 저장된 공구 데이터의 평가를 위해 필요한 모든 현재 상태를 저장하는 데 사용됩니다.

개별 데이터는 다음과 같습니다.

- 그룹의 활성 G 명령어:
  - 6 (G17, G18, G19)
  - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- 활성 가로축
- 머신 데이터:
  - MD18112 $MN_MM_KIND_OF_SUMCORR(TO 영역의 총합 옵셋의 속성)
  - MD20360 $MC_TOOL_PARAMETER_DEF_MASK(공구 파라미터 정의).
- 설정 데이터:
  - SD42900 $SC_MIRROR_TOOL_LENGTH(미러링 시의 신호 변경 공구 길이)
  - SD42910 $SC_MIRROR_TOOL_WEAR(미러링 시의 신호 변경 공구 마모)
  - SD42920 $SC_WEAR_SIGN_CUTPOS(절삭날 시스템을 가진 공구 마모의 신호)
  - SD42930 $SC_WEAR_SIGN(마모 신호)
  - SD42935 $SC_WEAR_TRANSFORM(공구 컴포넌트를 위한 변환)
  - SD42940 $SC_LENGTH_CONST(평면 변경을 위한 공구 길이 요소의 변경)
  - SD42942 $SC_TOOL_LENGTH_CONST_T(평면 변경 시의 선삭 공구를 위한 공구 길이 요소 변경)
  - SD42950 $SC_TOOL_LENGTH_TYPE(유형과 상관 없는 공구 길이 컴포넌트의 지정)
  - SD42954 $SC_TOOL_ORI_CONST_M(평면 변경 시의 밀링 공구를 위한 공구 오리엔테이션 요소 변경)
  - SD42956 $SC_TOOL_ORI_CONST_T(평면 변경 시의 선삭 공구를 위한 공구 오리엔테이션 요소 변경)
- 현재 전체 프레임의 오리엔테이션 컴포넌트(회전 및 미러링, 워크 옵셋이 없거나 스케일링)
- 오리엔테이션 컴포넌트 및 오리엔테이션 기능이 있는 활성 공구홀더의 결과로 도출되는 길이
- 오리엔테이션 컴포넌트 및 활성 변환의 결과로 도출된 길이

공구 환경을 설명하는 데이터에 더하여, 활성 공구의 T 번호, D 번호 및 DL 번호 역시 보관되고, 결과적으로 공구 이름을 다시 확인할 필요 없이 TOOLENV 호출과 같은 환경에서 공구에 추후 접근할 수 있습니다.

구문

```
<Status> = TOOLENV(<name>)
```

의미

TOOLENV(...) :	공구 환경을 저장하기 위해 미리 정의된 기능		
	블록에서만:		예
<Status>:	기능 반환값. 음수는 에러 상태를 의미합니다.		
	데이터 유형:		INT
	값:	0	기능 OK
		-1	공구 환경을 위해 유보된 메모리가 없음: MD18116 $MN_MM_NUM_TOOL_ENV = 0 이는 "공구 환경" 기능을 사용할 수 없다는 뜻입니다.
		-2	공구 환경을 위한 자유로운 메모리 위치를 사용할 수 없음.
		-3	공렬(Null string)은 공구 환경 이름으로 적합하지 않음.
		-4	파라미터(<name>)가 명시되지 않음.
파라미터			
1 <name>:	현재 데이터가 저장하기 위한 이름. 같은 이름의 데이터 세트가 존재한다면, 덮어쓰기가 됩니다. 이 경우 상태는 "0"입니다.		
	데이터 유형:		STRING

추가 정보

기본 치수/어댑터 치수 – 공구 길이 보정

공구 매거진 관리가 활성화된 상태일 때("공구 관리" 옵션을 사용해서만 사용 가능), 다음 머신 데이터의 값은 어댑터 길이 또는 공구 기본 치수(절삭날별 파라미터 $TC_DP21, $TC_DP22 및 $TC_DP23)가 공구 길이 계산에 통합되는지 여부를 정의합니다.

MD18104 $MN_MM_NUM_TOOL_ADAPTER(TO 영역에서의 공구 어댑터).

기계 데이터에 대한 변경은 제어 시스템에 전원이 공급된 이후에만 효과를 가지므로, 공구 환경에는 저장되지 않습니다.

오리엔테이션 기능 및 변환을 갖춘 공구홀더의 결과로 도출되는 길이:

참고

오리엔테이션 기능 및 변환을 모두 갖춘 공구홀더는 시스템 변수 또는 머신 데이터를 사용할 수 있는데, 이는 추가 공구 길이 컴포넌트로 작용하고, 실행된 회전에 대해 부분적으로 또는 완전하게 종속됩니다. 결과로 도출되는 추가 공구 길이 컴포넌트는 반드시 TOOLENV가 호출될 때 반드시 저장되어야 합니다. 왜냐하면 이들은 공구가 사용되는 환경의 일부를 대표하기 때문입니다.

어댑터 변환

어댑터 변환은 공구 어댑터의 속성이고 이에 따라 전체 공구의 속성이기도 합니다. 따라서 이는 다른 공구에 적요될 수 있는 공구 환경의 일부분이 아닙니다.

전체 공구 길이를 결정하는 데 필요한 전체 데이터를 저장함으로써, 공구가 더 이상 활성 상태가 아니거나 환경 조건(예: G 명령어 또는 설정 데이터)이 바뀌더라도 추후 시점에서의 공구의 효과적 길이를 계산할 수 있습니다. 유사하게, 상태가 저장되는 공구와 같은 조건에서 사용된다는 가정 하에 다른 공구의 효과적 길이도 계산될 수 있습니다.

공구 환경을 위한 데이터 세트의 최대 개수

머신 데이터 MD18116 $MN_MM_NUM_TOOL_ENV는 공구 환경을 설명하기 위해 저장되는 데이터 세트의 최대 개수를 정의하는 데 사용됩니다. 데이터는 TOA 영역에 있습니다. 제어 시스템을 껐을 때 동일하게 유지됩니다.

데이터는 백업 할 수 없습니다. 이는 다른 제어 시스템 간에 데이터를 이전할 수 없다는 것을 의미합니다.

4.13.13.2 도구 환경 삭제(DELTOOLENV)

DELTOOLENV 기능은 공구 환경을 설명하는 데 사용되는 데이터 세트를 삭제하는 데 사용됩니다. 삭제는 특정 이름 하에 저장된 데이터 세트에 더 이상 접근할 수 없다는 것을 의미합니다(접근 시도는 알람을 발생시킴).

참고

데이터 세트는 INITIAL.INI 다운로드 또는 콜드 스타트를 통한 DELTOOLENV 기능을 사용해야만 삭제할 수 있습니다(초기 머신 데이터를 이용한 NC 스타트업). 추가 자동 삭제 작업은 없습니다.

구문

```
<Status> = DELTOOLENV(<name>)
<Status> = DELTOOLENV()
```

의미

DELTOOLENV(...):	공구 환경을 삭제하기 위해 미리 정의된 기능		
	블록에서만:	예	
<Status>:	기능 반환값. 음수는 에러 상태를 의미합니다.		
	데이터 유형:	INT	
	값:	0	기능 OK
		-1	공구 환경을 위해 유보된 메모리가 없음: MD18116 $MN_MM_NUM_TOOL_ENV = 0 이는 "공구 환경" 기능을 사용할 수 없다는 뜻입니다.
		-2	지정된 이름의 공구 환경은 존재하지 않습니다.
파라미터			
1 <name>:	삭제할 데이터 세트 이름		
	데이터 유형:	STRING	
DELTOOLENV():	DELTOOLENV() 은 이름을 명시하지 않고 공구 환경을 설명하는 데이터 세트를 삭제합니다.		

4.13.13.3 T, D 및 DL 번호 읽기(GETTENV)

GETTENV 기능은 공구 환경에 저장된 T, D 및 DL 번호를 읽는 데 사용됩니다.

구문

```
<Status> = GETTENV(<name>, <TDDL>)
```

의미

GETTENV(...):	공구 환경을 설명하기 위한 데이터 세트에 있는 T, D 및 DL 번호를 읽기 위해 사전에 정의된 기능	
	블록에서만:	예

<Status>:	기능 반환값. 음수는 에러 상태를 의미합니다.		
	데이터 유형:		INT
	값:	0	기능 OK
		-1	공구 환경을 위해 유보된 메모리가 없음: MD18116 $MN_MM_NUM_TOOL_ENV = 0 이는 "공구 환경" 기능을 사용할 수 없다는 뜻입니다.
		-2	지정된 이름의 공구 환경은 존재하지 않습니다.
파라미터			
1 <name>:	T, D 및 DL 번호가 읽혀지는 데이터 세트 이름		
	데이터 유형:		STRING
2 <TDDL>:	이 결과 파라미터의 필드는 공구의 T, D 및 DL 번호를 포함하고 있고, 동 공구의 공구 환경은 특정 데이터 세트에 저장되어 있습니다. • <TDDL> [0]: T 번호 • <TDDL> [1]: D 번호 • <TDDL> [2]: DL 번호		
	데이터 유형:		INT[3]
GETTENV(,<TDDL>), GETTENV("",<TDDL>): 	기능 GETTENV를 호출할 때 첫 번째 파라미터를 생략하거나 공열을 첫 번째 파라미터로 전송하는 것이 허용됩니다. 이러한 두 가지 특별한 경우, <TDDL>에서는**활성** 공구의 T, D 및 DL 번호가 반환됩니다.		

4.13.13.4 저장된 공구 환경에 대한 정보 읽기($P_TOOLENVN, ($P_TOOLENV)

저장된 공구 환경에 관한 정보는 다음 시스템 변수를 사용하여 읽을 수 있습니다.

$P_TOOLENVN:	공구 환경을 설명하기 위해 TOOLENV 사용하여 정의된 (아직 삭제되지 않은) 데이터 세트의 번호를 제공합니다.			
	구문:	<n> = $P_TOOLENVN		
	의미:	<n>:	정의된 데이터 세트 개수	
			데이터 유형:	INT
			값 범위:	0 ... MD18116 $MN_MM_NUM_TOOL_ENV
	이 시스템 변수는 공구 환경이 가능하지 않더라도 접근할 수 있습니다 (MD18116 = 0). 이 경우 반환은 "0"입니다.			

$P_TOOLENV:	공구 환경을 설명하기 위한 <i>번째 데이터 세트의 이름을 제공합니다.			
	구문:	<Name> = $P_TOOLENV[<i>]		
	의미:	<name>:	번호 <i>를 가진 데이터 세트 이름	
			데이터 유형:	STRING
		<i>:	데이터 세트 번호	
			데이터 유형:	INT
			값 범위:	1 ... $P_TOOLENVN
	데이터 세트에 대한 번호 지정은 고정된 것이 아니고, 데이터 세트 삭제 또는 생성의 결과로 변경될 수 있습니다. 데이터 세트는 내부적으로 번호가 매겨집니다. 만일 <i>가 정의되지 않은 데이터 세트에 관한 것이라면 공렬은 반환됩니다. 만일 지수 <i>가 유효하지 않다면, 즉 <i>가 1보다 작거나 공구 환경(MD18116 $MN_MM_NUM_TOOLENV)을 위한 데이터 세트의 최대 번호보다 큰 경우 다음 알람이 출력됩니다. 알람 17020 "채택 불가능한 배열 지수 1"			

4.13.13.5 공구 길이 및/또는 공구 길이 컴포넌트 읽기(GETTCOR)

GETTCOR 기능은 공구 길이 또는 공구 길이 컴포넌트를 읽어내는 데 사용됩니다.

파라미터는 어떤 컴포넌트가 고려되는지 명시하고 공구가 사용되는 조건을 명시하는 데 사용됩니다.

구문

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

의미

GETTCOR(...):	공구 길이 또는 공구 길이 컴포넌트를 읽기 위해 사전에 정의된 기능	
	블록에서만:	예

<Status>:	기능 반환값. 음수는 에러 상태를 의미합니다.		
	데이터 유형:		INT
	값:	0	기능 OK
		-1	공구 환경을 위해 유보된 메모리가 없음: MD18116 $MN_MM_NUM_TOOL_ENV = 0 이는 "공구 환경" 기능을 사용할 수 없다는 뜻입니다.
		-2	<Stat> 하에서 명시된 이름을 가진 공구 환경이 존재하지 않습니다.
		-3	파라미터 <Comp>에서의 유효하지 않은 열. 오류의 원인은 유효하지 않은 문자 또는 두 번 프로그램된 문자일 수 있습니다.
		-4	유효하지 않은 T 번호
		-5	유효하지 않은 D 번호
		-6	유효하지 않은 DL 번호
		-7	존재하지 않는 메모리 모듈에 접근하려는 시도.
		-8	존재하지 않는 옵션에 접근하려는 시도(프로그램 가능한 공구 오리엔테이션, 공구 관리).
		-9	<Comp> 열은 콜론(좌표계 명시를 위한 식별자)을 포함하지만, 좌표계를 표시하는 유효한 문자가 뒤따라 나오지 않았습니다.
파라미터			

1	<Len>:	결과 벡터	
		데이터 유형:	REAL[11]
		벡터 컴포넌트는 다음 순서로 배치됩니다. • <Len> [0]: 공구 종류 • <Len> [1]: 절삭날 위치 • <Len> [2]: 가로 좌표 • <Len> [3]: 세로 좌표 • <Len> [4]: 직교 좌표 • <Len> [5]: 공구 반경 <Comp> 및 <Stat>에서 정의된 좌표계는 길이 컴포넌트를 위한 기준 좌표계로 사용됩니다. 좌표계가 <Comp>에서 정의되지 않은 경우, 공구 길이는 머신 좌표계에 표시됩니다. 기하 축에 대한 가로 좌표, 세로 좌표 및 직교 좌표의 지정은 공구 환경에서 사용된 활성 평면에 달려있습니다. 이는 G17의 경우 가로 좌표가 X에 대해 평행함, G18의 경우 가로 좌표가 Z에 평행함 등을 의미합니다. <Len>[6]에서 <Len>[10]까지의 컴포넌트는 추가 파라미터를 포함하고, 이는 공구의 기하학적 설명을 명시하는데 사용됩니다(예: 마모 또는 합계 및 설정 옵셋을 위한 기하학적 구조 및 해당 컴포넌트를 위한 $TC_DP7 to $TC_DP11). 이러한 5가지 추가 요소 및 공구 반경은 컴포넌트 E, G, S 및 W를 위해서만 정의됩니다. 평가는 <Stat>에 달려있지 않습니다. 4개의 명시된 컴포넌트 중 하나라도 공구 길이 계산에 개입되어 있다면, <Len>[6]에서 <Len>[10]까지의 해당 값은 0과 같지 않습니다. 잔여 컴포넌트는 결과에 영향을 미치지 않습니다. 치수는 제어 시스템의 기본 시스템(인치 또는 미터)을 참조합니다.	

2	<Comp>:	공구 길이 컴포넌트(옵션)			
		데이터 유형:	STRING		
		문자열은 두 개의 하위 열로 구성되고, 하위 열은 콜론에 의해 하나씩 분리됩니다. 일반 형태: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	**첫 번째 하위 열** 은 공구 길이를 계산할 때 고려되어야 할 공구 길이 컴포넌트를 지정합니다. 하위 열에서의 문자 순서, 그리고 표기법(대문자 또는 소문자)은 임의적인 것입니다. 빈칸 또는 여백을 두 문자 사이에 제한 없이 삽입할 수 있습니다. **참조:** 하위 열에서의 문자를 두 번 프로그램하는 것은 **허용되지 않습니다.**		
			문자:	-	마이너스 표시(첫 번째 문자로만 허용됨) 전체 공구 길이는 계산되고, 다음 열에 명시된 컴포넌트를 뺍니다.
				C	어댑터 또는 공구 기본 수치(두 개의 대체 컴포넌트 중 하나가 사용 중인 공구에 대해 활성화됨)
				E	셋업 옵셋
				G	기하
				K	키네마틱 변환(일반 3, 4 및 5-축 변환을 위해서만 평가됨)
				S	합계 옵셋
				T	오리엔테이션 기능이 있는 공구 홀더
				W	마모량
			첫 번째 하위 열이 비어있다면(여백은 제외), 전체 공구 길이는 모든 컴포넌트를 감안해서 계산됩니다. 이는 <Comp> 파라미터가 명시되지 않더라도 적용됩니다.		
		<Substr_2>:	옵션 **두 번째 하위 열** 은 공구 길이가 출력되게 되는 좌표계를 식별합니다. 두 번째 하위 열은 하나의 단일 관련 문자로 구성됩니다.		
			문자:	A	조정 가능한 좌표계(ACS)
				B	기본 좌표계(BCS)

				K	구조 전환의 공구 좌표계(Tool coordinate system of kinematic transformation: KCS)
				M	기계 좌표계(MCS)
				T	공구 좌표계(Tool coordinate system: TCS)
				W	공작물 좌표계 (WCS)
			어떤 좌표계도 명시되지 않았다면, 평가는 MCS(머신 좌표계)에서 실행됩니다. 어떤 회전이 감안된다면, <Stat>에서 정의된 공구 환경에 명시된 것입니다.		
3	<_Stat>:	공구 환경을 위한 데이터 세트 이름(옵션)			
		데이터 유형:	STRING		
		이 파라미터의 값은 공렬("")이거나 명시되지 않았고, 이에 따라 현재 상태가 사용됩니다. 공구가 명시되지 않으면 현재 공구가 사용됩니다.			
4	<T>:	공구의 내부 T 번호(옵션).			
		데이터 유형:	INT		
		이 파라미터가 명시되지 않았거나 값이 "0"이라면, <Stat>에 저장된 공구가 사용됩니다. I이 파라미터의 값이 "-1"이라면 그때는 활성 공구의 T 번호가 사용됩니다. 활성 공구의 번호를 명확하게 명시하는 것도 가능합니다. **참조:** <Stat>이 명시되지 않으면 실제 상태가 공구 환경으로 사용됩니다. <T> = 0은 공구 환경에 저장된 T 번호를 의미하므로, 활성 공구는 이 환경에서 사용됩니다. 즉 이러한 특수 경우에서는 파라미터 <T> = 0 및 <T> = -1이 같은 의미를 가집니다.			
5	<D>:	공구의 절삭날(옵션).			
		데이터 유형:	INT		
		이 파라미터가 명시되지 않거나 값이 "0"이면, 사용된 D 번호는 T 번호의 소스에 기초합니다. 공구 환경의 T 번호가 사용되면 공구 환경의 D 번호도 역시 읽힙니다. 그렇지 않을 경우 현재 활성 상태인 공구의 D 번호가 읽힙니다.			

6	<DL>:	위치에 따른 옵셋의 번호(옵션).	
		데이터 유형:	INT
		이 파라미터가 명시되지 않으면, 사용된 DL 번호는 T 번호의 소스에 기초합니다. 공구 환경의 T 번호가 사용되면 공구 환경의 D 번호도 역시 읽힙니다. 그렇지 않을 경우 현재 활성 상태인 공구의 D 번호가 읽힙니다.	

예

GETTCOR(_LEN)	모든 컴포넌트를 감안하면서 머신 좌표계에서 현재 활성 상태인 공구의 공구 길이를 측정합니다.
GETTCOR(_LEN,"CGW:W")	어댑터 또는 공구 기본 수치, 기하학적 구조 및 마모로 구성되는, 활성 공구를 위한 공구 길이를 계산합니다. 오리엔테이션 기능 또는 키네마틱 변환을 갖는 공구홀더와 같은 추가 컴포넌트는 고려되지 않습니다. 공작물 좌표계에서의 출력.
GETTCOR (_LEN,"-K:B")	활성 키네마틱 변환이 있을 수 있는 길이 컴포넌트를 감안하지 않고 활성 공구의 공구 길이를 계산합니다. 기본 좌표계에서의 출력
GETTCOR (_LEN,":M","Testenv1",,3)	"Testenv1"라는 이름의 공구 환경에 저장된 공구를 위한 머신 좌표계에서 전체 공구 길이를 계산합니다. 하지만, 저장된 절삭날 번호와 관계 없이 계산은 절삭날 번호 D3에 대해서만 실행됩니다.

추가 정보

어댑터 변환/오리엔테이션 능력/좌표 변환을 갖는 공구 홀더

어댑터 변환, 오리엔테이션 능력 및 좌표 변환을 갖는 공구홀더에 의해 실행되는 모든 회전 및 콤포넌트 교환은 공구 환경의 일부입니다. 따라서 해당 길이 콤포넌트가 포함되지 않더라도 이들은 항상 실행됩니다. 이것이 원치않는 상황이라고 하더라도, 공구 환경은 반드시 정의되어야 하고, 이 공구 환경에서 해당 변환은 활성화되지 않습니다. 많은 경우(즉, 오리엔테이션 능력을 갖는 변환 또는 공구 홀더가 기계에 대해 사용되지 않은 경우) 공구 환경을 위해 저장된 데이터 세트가 자동으로 이러한 조건을 충족하고 사용자가 특별히 준비하지 않아도 되는 결과를 가져옵니다.

선삭 및 연삭 공구: MD20360 $MC_TOOL_PARAMETER_DEF_MASK에 따른 공구 길이 계산

다음 머신 데이터는 직경 축이 선삭 및 연삭 도구를 위해 사용된다면 마모 및 공구 길이가 어떻게 평가될 것인지를 정의합니다.

MD20360 $MC_TOOL_PARAMETER_DEF_MASK(공구 파라미터 정의).

비트	값	
0	선삭 및 연삭 공구의 경우, 가로 축의 **마모 파라미터** 가 직경 값으로 고려됩니다.	
	= 0 (기본 설정)	아니오
	=1	예
1	선삭 및 연삭 공구의 경우, 가로 축의 **공구 길이 컴포넌트** 가 직경 값으로 고려됩니다.	
	= 0 (기본 설정)	아니오
	=1	예

개입된 바이트가 설정된다면, 관련 입력에 0.5의 팩터로 가중치가 주어집니다. 이 가중치는 GETTCOR에 의해 반환되는 공구 길이에 반영됩니다.

예제:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 $MC_DIAMETER_AX_DEF(이송 축 기능을 갖는 기하 축) = "X"

X는 직경 축입니다(표준 선삭 기계 구성)

프로그램 코드	설명
N30 $TC_DP1[1,1]=500	
N40 $TC_DP2[1,1]=2	
N50 $TC_DP3[1,1]=3.0	; 기하학적 구조 L1
N60 $TC_DP4[1,1]=4.0	
N70 $TC_DP5[1,1]=5.0	
N80 $TC_DP12[1,1]=12.0	; 마모 L1
N90 $TC_DP13[1,1]=13.0	
N100 $TC_DP14[1,1]=14.0	
N110 T1 D1 G18	
N120 R1=GETTCOR(_LEN,"GW")	
N130 R3=_LEN[2]	; 17.0 (= 4.0 + 13.0)
N140 R4=_LEN[3]	; 7.5 (= 0.5 * 3.0 + 0.5 * 12.0)
N150 R5=_LEN[4]	; 19.0 (= 5.0 + 14.0)
N160 M30	

좌표 변환의 길이 요소 및 오리엔테이션 능력을 갖는 공구 홀더

오리엔테이션 능력을 갖는 공구홀더 가 공구 길이 계산 과정에서 감안된다면, 다음 벡터가 그 계산에 포함됩니다.

종류	벡터
M	l1 및 l2
T	l1, l2 및 l3
P	공구 길이는 오리엔테이션 능력을 갖는 공구 홀더에 의한 영향을 받지 않습니다.

일반적인 **5축 변환**의 경우, 다음 머신 데이터가 변환기 유형 24 및 56을 위한 공구 길이 계산에 포함됩니다.

변환 유형	머신 데이터
24	MD24550/24650 $MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2 MD24558/24658 $MC_TRAFO5_PART_OFFSET_1/2
56	MD24550/24650 $MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2

변환 유형 56(움직이는 공구 및 움직이는 공작물)은 오리엔테이션 능력을 갖는 공구홀더를 위한 유형 M에 해당합니다.

5축 변환의 경우, 이전 소프트웨어 버전에서는 벡터 MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2(채널 내 1번째/2번째 5축 변환의 좌표 옵셋의 벡터)가 오리엔테이션 능력을 갖는 유형 M 공구 캐리어를 위한 두 벡터 l_1 및 l_3 에 해당합니다.

두 가지 경우 모두에 있어서 총합만이 변환에 대해 유효합니다. 두 개별 요소가 구성되는 방식은 무의미합니다. 하지만, 공구 길이를 계산할 때는 어떤 요소가 공구에 할당되는지, 어떤 요소가 공구 테이블에 할당되는지가 유의미합니다. 이것이 머신 데이터 MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2(테이블의 벡터 좌표 옵셋)이 도입된 이유입니다. 이는 벡터 l3에 해당합니다. 머신 데이터:MD24560/24660 $MC_TRAFO5_JOINT_OFFSET_1/2은 더 이상 l1 및 l3의 총합에 부합하지 않고, 단지 벡터 l1에 부합합니다. 만일 머신 데이터 MD24558/24658 $MC_TRAFO5_JOINT_OFFSET_PART_1/2이 0인 경우라면, 동작은 이전과 같습니다.

호환성

GETTCOR 기능은 TOOLENV 및 SETTCOR 기능과 함께, 측정 싸이클에서 이전에 외부적으로 실행된 기능의 일부분을 교체하는데 사용됩니다.

효과적인 공구 길이를 실제로 결정하는 파라미터의 일부만이 측정 싸이클에서 실행되었습니다. 위에서 언급한 기능들은 공구 길이 계산과 관련한 측정 싸이클의 동작을 재생산하기 위해 구성될 수 있습니다.

4.13.13.6 공구 컴포넌트 변경(SETTCOR)

SETTCOR 기능은 개별 컴포넌트를 평가할 때 개입될 수 있는 모든 일반 조건을 감안하면서 공구 컴포넌트를 변경하는 데 사용됩니다.

참고

용어 관련: 이후 공구 길이와 함께 공구 컴포넌트가 개입되고, 그러고 나면 벡터적인 시각에서 고려된 컴포넌트가 의미되고, 이는 전체 공구 길이를 구성합니다(예: 기하학적 구조 또는 마모). 이러한 컴포넌트는 세 개의 개별 값(L1, L2, L3)을 구성하고, 이는 이후 좌표값으로 불립니다.

따라서 공구 컴포넌트 "기하학적 구조"는 2개의 좌표값 $TC_DP3 to $TC_DP5를 구성합니다.

구문

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>,
<GeoAx>, <Stat>, <T>, <D>, <DL>])
```

의미

SETTCOR(...):	공구 컴포넌트를 변경하기 위해 사전에 정의된 기능	
	블록에서만:	예

<Status>:	기능 반환값. 음수는 에러 상태를 의미합니다.		
	데이터 유형:		INT
	값:	0	기능 OK
		-1	공구 환경을 위해 유보된 메모리가 없음: MD18116 $MN_MM_NUM_TOOL_ENV = 0 이는 "공구 환경" 기능을 사용할 수 없다는 뜻입니다.
		-2	<Stat> 하에서 명시된 이름을 가진 공구 환경이 존재하지 않습니다.
		-3	파라미터 <Comp>에서의 유효하지 않은 열. 오류의 원인은 유효하지 않은 문자 또는 두 번 프로그램된 문자일 수 있습니다.
		-4	유효하지 않은 T 번호.
		-5	유효하지 않은 D 번호.
		-6	유효하지 않은 DL 번호.
		-7	존재하지 않는 메모리 모듈에 접근하려는 시도.
		-8	존재하지 않는 옵션에 접근하려는 시도(프로그램 가능한 공구 오리엔테이션, 공구 관리).
		-9	파라미터 <CorComp> 을 위한 유효하지 않은 숫자 값.
		-10	파라미터 <CorMode>를 위한 유효하지 않은 숫자 값.
		-11	파라미터 <Comp> 및 <CorComp>의 내용은 상반됩니다.
		-12	파라미터 <Comp> 및 <CorMode>의 내용은 상반됩니다.
		-13	<GeoAx 파라미터의 내용은 기하 축을 지정하진 않습니다.
		-14	존재하지 않는 셋업 옵셋에 대한 쓰기 시도.
파라미터			

1	<CorVal>:	수정 벡터 <Stat>에 의해 정의된 공작물 좌표계(WCS)에서 다음 지정이 적용됩니다. • <CorVal> [0]: 가로 좌표 • <CorVal> [1]: 세로 좌표 • <CorVal> [2]: 직교 좌표 단 하나의 컴포넌트가 수정된다면(즉, 벡터 수정, 파라미터 <CorMode> 참조), 공구 컴포넌트가 작용하는 축과는 독립적으로 수정값은 항상 <CorVal>[0]에 있습니다. 그러면 다른 두 개의 컴포넌트의 내용은 평가되지 않습니다. 만일 <CorVal> 또는 <CorVal>의 컴포넌트가 가로 축을 기준으로 한다면, 데이터는 **반경 치수** 로 평가됩니다. 이는 예를 들어 도구가 명시된 치수에 비해 "더 길다"는 것을 의미합니다. 이는 2배나 더 큰 공작물 직경에 대한 변화를 초래합니다. 치수는 제어 시스템의 **기본 시스템** (인치 또는 미터)을 참조합니다.	
		데이터 유형:	REAL[3]

2	<Comp>:	공구 컴포넌트(들)			
		데이터 유형:	STRING		
		문자열은 두 개의 하위 열로 구성되고, 하위 열은 콜론에 의해 하나씩 분리됩니다. 일반 형태: "<SubStr_1> [: <SubStr_2]"			
		<SubStr_1>:	**첫 번째 하위열** 은 반드시 항상 사용할 수 있어야 하며, 1개 또는 2개의 문자로 구성될 수 있어야 합니다. 첫 번째 컴포넌트의 첫 번째 또는 단일 문자(Val_1)와 두 번째 컴포넌트의 두 번째 문자(Val_2)는 그 뒤의 파라미터 <CorComp>와 <CorMode>에 따라 처리됩니다.		
			문자:	C	어댑터 또는 공구 기본 수치(두 개의 대체 컴포넌트 중 하나가 사용 중인 공구에 대해 활성화됨)
				E	셋업 옵셋
				G	기하
				S	합계 옵셋
				W	마모량
		<Substr_2>:	**두 번째 문자열** 은 옵션입니다. 대체 방안으로 (개별) 문자 "W" 또는 "T"로 구성될 수 있습니다.		
			문자:	W	만일 두 번째 문자열이 비어 있거나 문자 "W"를 포함하고 있다면, 옵셋값은 **공작물**좌표계(WCS)에서 측정된 것으로 고려됩니다.
				T	만일 두 번째 문자열이 문자 "T"를 포함하고 있다면, 옵셋값은 **공구**좌표계(TCS)에서 측정된 것으로 고려됩니다.
		열 문자의 표기법(대문자 또는 소문자)은 임의적인 것입니다. 띄우기 또는 탭(여백)을 제한 없이 삽입할 수 있습니다.			

3	<CorComp> :	설명될 공구 데이터의 컴포넌트를 명시합니다(옵션).		
		데이터 유형:		INT
		값:	0	옵셋값 <CorVal>[0]은 공작물 좌표계 또는 공구 좌표계 상의 파라미터 <GeoAx>에서 이송된 기하 축을 참조합니다(파라미터 <Comp>에 대한 설명도 참조). 이는 옵셋값이 반드시 지정된 공구 컴포넌트에서 계산되어야 한다는 것을 의미하며, 이렇게 해서 공구 길이 계산에 영향을 미칠 수 있는 모든 파라미터를 고려하면서, 지정된 축 방향에서 지정된 값에 의한 총 공구 길이의 변경이 획득됩니다. 이 변경은 <Comp>에 명시된 컴포넌트 및 <CorMode>에서 명시된 상징적 알고리즘을 수정함으로써 달성되어야 합니다(다음 파라미터 참조). 따라서 결과로 도출되는 수정은 모든 3개 축 컴포넌트에 대해 영향을 미칩니다.
			1	"0"과 유사하지만 벡터적임. 벡터 <CorVal>의 내용은 공작물 좌표계 또는 공구 좌표계에서의 가로 좌표, 세로 좌표 및 직교 좌표를 참조합니다(파라미터 <Comp> 설명 참조). 다음 파라미터 <GeoAx>가 평가되지 않습니다.
			2	벡터적 옵셋, 즉 L1, L2 및 L3은 동시에 변합니다. "0 및 "1" 버전에 비해, <CorVal>에 담겨진 옵셋값은 공구의 Val_1 컴포넌트의 좌표를 참조합니다(다음 파라미터 <CorMode> 참조). 공작물 좌표계와 비교된 현재 공구의 어떤 경사도 옵셋에 영향을 미치지 않습니다.
			3 - 5	공구 길이 L1 ~ L3($TC_DP3 to $TC_DP5), 마모 관련 해당 값, 설정 또는 추가 옵셋의 수정. 옵셋값은 <CorVal>[0]에 포함되어 있습니다. 옵셋값은 공구의 Val_1 컴포넌트의 좌표에서 측정됩니다(다음 파라미터 <CorMode> 참조). 공작물 좌표계와 비교된 현재 공구의 어떤 경사도 옵셋에 영향을 미치지 않습니다.
			6	공구 반경($TC_DP6) 또는 마모 관련 해당 값, 설정 또는 추가 옵셋의 수정. 머신 데이터 MD20360 $MC_TOOL_PARAMETER_DEF_MASK 내의 비트 10 및 11(직경 평가 및/또는 반경 또는 직경으로 명시되는 직경 마모 데이터)이 고려됩니다.

			7 – 11	$TC_DP7 to $TC_DP11 또는 마모 관련 해당 값, 설정 또는 추가 옵셋의 수정. 이러한 파라미터는 공구 반경처럼 취급됩니다.
		이 파라미터가 명시되지 않으면 값은 "0"입니다.		

4	<CorMode> :	실행되어야 하는 쓰기 작업의 유형을 명시합니다(옵션).		
		데이터 유형:		INT
		값:	0	Val_{1new} = <CorVal>
			1	Val_{1new} = Val_{1old} + <CorVal>
			2	Val_{1new} = <CorVal> Val_{2new} = 0
			3	Val_{1new} = Val_{1old} + Val_{2old} + <CorVal> Val_{2new} = 0
		표기법 Val_{1old} + Val_{2old} 는 상징적인 것입니다. (<_Stat>의 상태로 인해) 두 개의 컴포넌트가 다른 방식으로 평가된다면, 즉 회전이 두 개의 컴포넌트 사이에서 효과적이라면, Val_{2old} 는 추가 이전에 변환됩니다. 그리하여 Val_{2new} 를 삭제한 후 그리고 <CorVal> 추가 이전에 결과적으로 얻어지는 공구 길이는 변하지 않습니다. <CorVal>은 항상 Val_1 을 참조합니다. <CorVal>은 파라미터 <Comp>의 두 번째 파트에 의존하는 값이고, 공작물 좌표계(WCS) 또는 공구 좌표계(TCS)에서 측정되는 값입니다. 따라서 이것은 공구 컴포넌트에 대해 이미 변환되었고, 공구 컴포넌트 내에서 측정되어야 합니다. 따라서, 이것은 저장된 값과 함께 직접 측정될 수는 없지만, Val_1 또는 Val_2 에 대한 추가 이전으로 되돌려서 변환되어야 합니다. 이는 옵셋이 <CorComp>에 의해 정의된 축과는 다른 축에 대해 작용한다는 것을 의미할 수 있고, 여러 개의 축에 대해 작동한다는 것을 의미할 수 있습니다. <CorComp> = 0의 경우, 즉 <CorVal>이 벡터를 포함하지 않지만 개별 값을 가지는 경우, 설명한 작업은 <CorVal>이 측정된 좌표에서 실행됩니다 (WCS/TCS). 특히, 이는 변형 2 및 3에서 Val_{2new} 을 영(zero)으로 설정하는 데 적용됩니다. 이 결과는 공구 좌표로 다시 변환됩니다. 0으로 설정될 좌표 값 중 어떤 것(L1, L2, L3)도 0이 되지 않거나, 이전에 0이었던 좌표 값이 0과 같지 않다는 것을 의미합니다. 하지만, 세 개의 기하 축 모두에 대해 해당 작업이 연속적으로 실행되면, 삭제될 컴포넌트의 3개 좌표값 모두가 확실히 0이 됩니다. 공구가 공작물 좌표계에 대해 회전하지 않거나 혹은 회전해서 모든 공구 컴포넌트가 좌표 축(축 교환 작업)에 대해 평행으로 남는다면, 이것 역시 단지 한 개의 공구 좌표만이 바뀌는 것을 보장합니다. 아무 시퀀스에서의 세 개 좌표 축 모두에 대해 <CorComp> = 0과 같은 작업(<CorMode>)의 연속적으로 실행하는 것은 <CorComp>=1과 같은 작업의 단일 실행과 동일합니다.		

		예를 들어, 값 "0" 및 "1", 파라미터 <Comp>는 반드시 한 개의 문자를 포함하고 있어야 하고, 파라미터 값 "2" 및 "3"의 경우에는 두 개의 문자를 포함해야 합니다. 예: <Comp>는 열 "ES", <CorMode>는 값 "2"를 포함합니다. ⇒ 셋업 옵셋$_{신규}$ = <CorVal>, 총합 옵셋$_{신규}$ = 0 파라미터 <CorMode>가 명시되지 않으면 값은 "0"입니다.	
5	<GeoAx>:	옵셋값<CorVal>[0]이 읽히는 기하 축의 지수를 명시합니다(옵션).	
		데이터 유형:	INT
		값 범위:	0 ... 2
		지수 0~2는 현재 공구 환경의 활성 평면(G17/G18/G19)에서의 가로 좌표, 세로 좌표 및 직교 좌표를 참조합니다. 이 파라미터의 내용은 파라미터 <CorComp>가 "0"의 값을 가질 때에만 평가됩니다.	
6	<Stat>:	공구 환경을 위한 데이터 세트 이름(옵션)	
		데이터 유형:	STRING
		이 파라미터의 값은 공열("")이거나 명시되지 않았고, 이에 따라 현재 상태가 사용됩니다. 공구가 명시되지 않으면 현재 공구가 사용됩니다.	
7	<T>:	공구의 내부 T 번호(옵션).	
		데이터 유형:	INT
		이 파라미터가 명시되지 않았거나 값이 "0"이라면, <Stat>에 저장된 공구가 사용됩니다. I이 파라미터의 값이 "-1"이라면 그때는 활성 공구의 T 번호가 사용됩니다. 활성 공구의 번호를 명확하게 명시하는 것도 가능합니다. **참조:** <Stat>이 명시되지 않으면 실제 상태가 공구 환경으로 사용됩니다. <T> = 0은 공구 환경에 저장된 T 번호를 의미하므로, 활성 공구는 이 환경에서 사용됩니다. 즉 이러한 특수 경우에서는 파라미터 <T> = 0 및 <T> = -1이 같은 의미를 가집니다.	
8	<D>:	공구의 절삭날(옵션).	
		데이터 유형:	INT
		이 파라미터가 명시되지 않거나 값이 "0"이면, 사용된 D 번호는 T 번호의 소스에 기초합니다. 공구 환경의 T 번호가 사용되면 공구 환경의 D 번호도 역시 읽힙니다. 그렇지 않을 경우 현재 활성 상태인 공구의 D 번호가 읽힙니다.	

9	<TL>:	위치에 따른 옵셋의 번호(옵션).	
		데이터 유형:	INT
		이 파라미터가 명시되지 않으면, 사용된 DL 번호는 T 번호의 소스에 기초합니다. 공구 환경의 T 번호가 사용되면 공구 환경의 D 번호도 역시 읽힙니다. 그렇지 않을 경우 현재 활성 상태인 공구의 D 번호가 읽힙니다. T, D 및 DL이 위치 종속 옵셋 없이 공구를 명시한다면, 어떤 총합 또는 셋업 옵셋도 파라미터 <Comp>에서 명시되지 않습니다(<Status> 내 오류 코드).	

참고

세 가지 파라미터 <Comp>, <CorComp> 및 <CorMode>의 가능한 조합 모두가 유의미한 것은 아닙니다. 예를 들어, <CorComp>에서의 알고리즘 3은 두 개의 문자가 <Comp>에서 명시되는 것을 요구합니다. 유효하지 않은 파라미터 조합이 명시되면, 해당 오류 코드는 <Status>에서 반환됩니다.

예

예 1

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; 밀링 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP12[1,1]=1.0	; 마모 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"G",0,0,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X0.000 Y0.000 **Z1.333**
N90 M30	

<CorComp>은 "0"이고, 따라서 Z 방향으로 작용하는 기하 컴포넌트의 좌표값은 반드시 옵셋 값 0.333에 의해 교체되어야 합니다.

결과로 도출되는 총 공구 길이는 다음과 같습니다. L1 = 0.333 + 1.000 = 1.333

예 2

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; 밀링 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP12[1,1]=1.0	; 마모 L1

프로그램 코드	코멘트
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"W",0,1,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X0.000 Y0.000 Z11.333
N90 M30	

<CorComp>은 "1"이고, Z 축으로 작용하는 이는 옵셋값 0.333은 마모 값 1.0에 추가됩니다.

결과로 도출되는 총 공구 길이는 다음과 같습니다. L1 = 10.0 + 1.333 = 11.333

예 3

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; 밀링 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP12[1,1]=1.0	; 마모 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"GW",0,2,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X0.000 Y0.000 Z0.333
N90 M30	

<CorComp>은 "2"이고, 이에 따라 Z축에서 효과적인 옵셋은 기하 컴포넌트에 입력되고(예전 값은 덮어쓰기 됨) 마모값은 삭제됩니다.

결과로 도출되는 총 공구 길이는 다음과 같습니다. L1 = 0.333 + 0.0 = 0.333

예 4

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; 밀링 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP12[1,1]=1.0	; 마모 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"GW",0,3,2)**	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X0.000 Y0.000 Z11.333
N90 M30	

<CorComp>은 "3"이고, 따라서 마모값 및 보정값은 기하 컴포넌트에 추가되고 마모 컴포넌트는 삭제됩니다.

결과로 도출되는 총 공구 길이는 다음과 같습니다. L1 = 11.333 + 0.0 = 11.333

예 5

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=120	; 밀링 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP12[1,1]=1.0	; 마모 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=**SETTCOR(_CORVAL,"GW",0,3,0)**	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X0.333 Y0.000 Z11.000
N90 M30	

이전 예에서처럼 <CorComp>은 "3"이지만, 보정은 이제 지수 "0"(X축)을 갖는 기하 축에 효과적인데, 지수 "0"은 밀링 공구의 경우 G17로 인한 공구 컴포넌트 L3에 지정됩니다. 결과적으로 SETTCOR를 호출할 때 공구 파라미터 $TC_DP3 and $TC_DP12에는 영향이 가지 않습니다. 대신, 보정값은 $TC_DP5에 입력됩니다.

예 6

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 선삭 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP4[1,1]=15.0	; 기하학적 구조 L2
N50 $TC_DP12[1,1]=10.0	; 마모 L1
N60 $TC_DP13[1,1]=0.0	; 마모 L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",0,3,1)**	
N110 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X24.330 Y0.000 Z17.500
N120 M30	

공구는 선삭 공구입니다. 프레임 회전은 N80에서 활성화되고, 기본 좌표계(BCS)가 공작물 좌표계(WCS)를 기준으로 회전하도록 만듭니다. WCS에서는 G18이 활성 상태이므로 보정값(N70)이 지수 1을 가진 기하 축, 즉 X축에 대해 작용합니다. <CorMode> = 3이므로, N100이 실행되면 WCS의 X축 방향으로의 공구 마모는 반드시 0이 되어야 합니다.

프로그램 끝에 있는 관련 공구 파라미터의 내용은 다음과 같습니다.

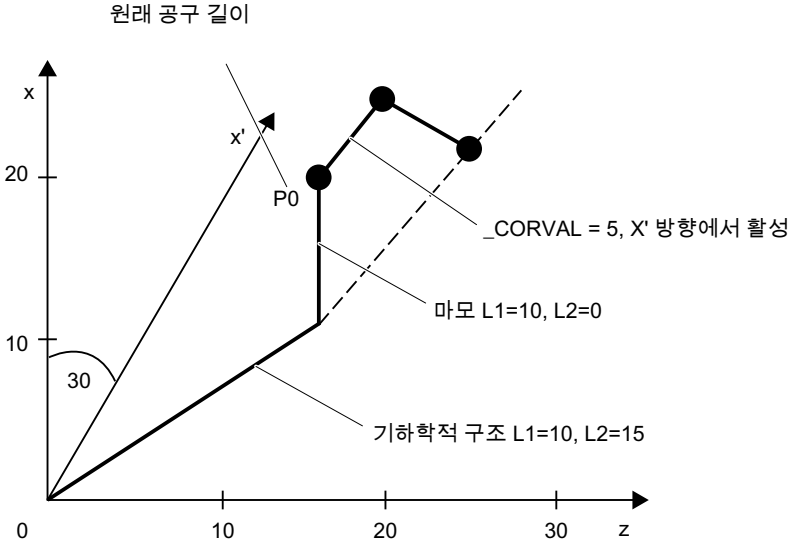
$TC_DP3[1,1]: 21.830 ; 기하학적 구조 L1

$TC_DP4[1,1] : 21.830 ; 기하학적 구조 L2

$TC_DP12[1,1] : 2.500 ; 마모 L1

$TC_DP13[1,1] : -4.330 ; 마모 L2

기하학적 관계가 아래 그림에 나와 있습니다. _CORVAL을 포함하는 총 마모는 WCS 내 X' 방향으로 지정됩니다. 이는 지점 P2를 만들어 냅니다. 이 지점의 좌표(X/Y 좌표에서 측정됨)는 공구의 기하 컴포넌트에 입력됩니다. 차이 벡터 $P_2 - P_1$ 은 마모에 남습니다. 이렇게 해서 마모는 _CORVAL 방향에서 컴포넌트를 더 이상 가지지 않습니다.



만일 프로그램 예가 다음 지시사항과 함께 N110 이후에도 지속되면, 보정이 이제 Z' 축에서 효과적인 관계로 잔여 마모는 완전히 기하학적 구조에 포함됩니다(파라미터 <GeoAx> = 0):

```
N120 _CORVAL[0]=0.0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> MCS 위치 X24.330 Y0.000 Z17.500
```

신규 보정 값이 "0"이므로, 총 공구 길이 및 이에 따른 N140에서 접근된 위치는 바뀌지 않습니다. _CORVAL이 N120에서 "0"과 같지 않다면, 신규 총 공구 길이 및 이에 따른 N140에서의 새 위치가 발생하게 됩니다. 하지만, 공구 길이의 마모 컴포넌트는 항상 0이게 되는데, 이는 총 공구 길이가 이후에 항상 공구의 기하 컴포넌트에 포함되게 된다는 것입니다.

<CorComp> = 0 파라미터로 SETTCOR를 두 번 호출함으로써 달성되는 것과 같은 결과를 <CorComp> = 1을 한번 만 호출(벡터 보정)함으로써도 도달할 수 있습니다.

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 선삭 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP4[1,1]=15.0	; 기하학적 구조 L2
N50 $TC_DP12[1,1]=10.0	; 마모 L1
N60 $TC_DP13[1,1]=0.0	; 마모 L2

프로그램 코드	코멘트
N70 _CORVAL[0]=0.0	
N71 _CORVAL[1]=5.0	
N72 _CORVAL[2]=0.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",1,3,1)**	
N110 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X24.330 Y0.000 Z17.500
N120 M30	

이 경우, N100에서의 첫 번째 SETTCOR 호출 이후 공구의 모든 마모 컴포넌트가 0으로 설정됩니다.

예 7

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 선삭 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP4[1,1]=15.0	; 기하학적 구조 L2
N50 $TC_DP12[1,1]=10.0	; 마모 L1
N60 $TC_DP13[1,1]=0.0	; 마모 L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",3,3)**	
N110 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X25.000 Y0.000 Z15.000
N120 M30	

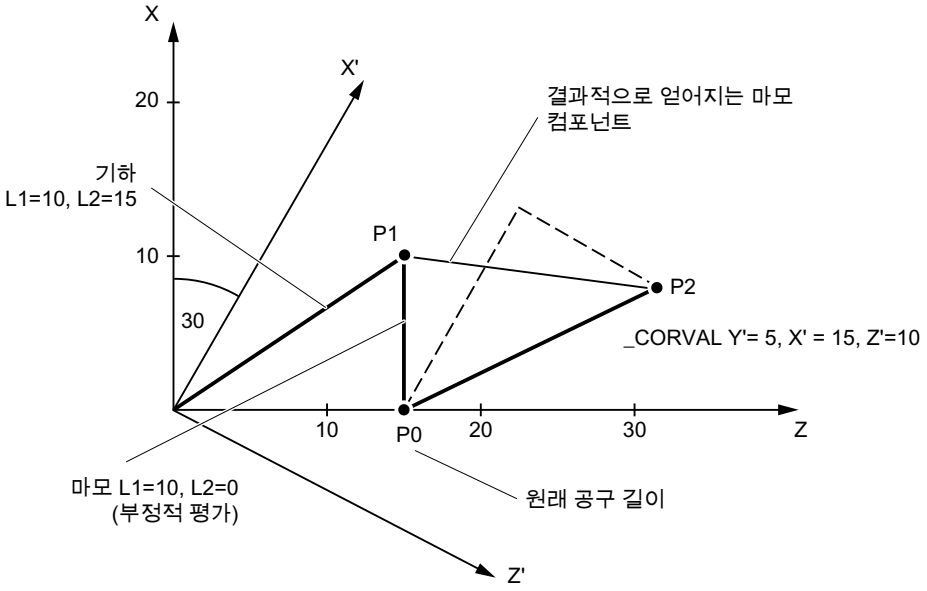
예제 6에 비교해 보면 파라미터 <CorComp> = 3 및 <GeoAx> 파라미터는 생략할 수 있습니다. _CORVAL[0]에 포함된 값은 이제 공구 길이 컴포넌트 L1에 대해 즉각 작용하고, N80에서의 회전은 결과에 대해 아무런 영향을 미치지 않고, _CORVAL[0]과 함께, 그리고 N100에서의 첫 번째 SETTCOR 호출 이후 $TC_DP13 때문에 총 공구 길이는 공구의 기하 컴포넌트에 보관된다는 결과와 함께, $TC_DP12에서의 마모 컴포넌트는 기하 컴포넌트에 포함됩니다.

예 8

프로그램 코드	코멘트
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 선삭 공구
N30 $TC_DP3[1,1]=10.0	; 기하학적 구조 L1
N40 $TC_DP4[1,1]=15.0	; 기하학적 구조 L2
N50 $TC_DP5[1,1]=20.0	; 기하학적 구조 L3
N60 $TC_DP12[1,1]=10.0	; 마모 L1
N70 $TC_DP13[1,1]=0.0	; 마모 L2
N80 $TC_DP14[1,1]=0.0	; 마모 L3

프로그램 코드	코멘트
N90 $SC_WEAR_SIGN=TRUE	
N100 _CORVAL[0]=10.0	
N110 _CORVAL[1]=15.0	
N120 _CORVAL[2]=5.0	
N130 ROT Y-30	
N140 T1 D1 G18 G0	
N150 R1=**SETTCOR(_CORVAL,"W",1,1)**	
N160 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X7.990 Y25.000 Z31.160
N170 M30	

데이터 설정:SD42930 $SC_WEAR_SIGN은 N90에서 작동 가능 상태로 됩니다. 즉, 마모가 음의 신호와 함께 반드시 평가되어야 합니다. 보정이 벡터적이고(<CorComp> = 1), 그리고 보정 벡터가 반드시 마모(<CorMode> = 1)에 추가되어야 합니다. Z/X 평면에서의 기하학적 관계가 아래 다이어그램에 나와 있습니다.



공구의 기하 컴포넌트는 <CorMode> = 1로 인해 바뀌지 않고 그대로 있습니다. WCS에서 정의된 보정 벡터(y 축 주변 회전)은 반드시 마모 컴포넌트에 포함되어야 하는데, 그래야만 그림. 3의 총 공구 길이가 지점 P_2 를 참조합니다. 따라서, 공구의 결과로 도출되는 마모 컴포넌트는 두 지점 P_1 및 P_2 의 거리에 의해 주어집니다.
하지만 셋팅 데이터 SD42930 $SC_WEAR_SIGN으로 인해 마모가 부정적으로 평가되므로, 이런 식으로 결정되는 보정은 음의 표시와 함께 보정 메모리에 입력되어야 합니다. 프로그램 끝에 있는 관련 공구 파라미터의 내용은 다음과 같습니다.

$TC_DP3[1,1]: 10.000 ; 기하학적 구조 L1(변동 없음)

$TC_DP4[1,1] : 15.000 ; 기하학적 구조 L2(변동 없음)

$TC_DP5[1,1] : 10.000 ; 기하학적 구조 L3(변동 없음)

$TC_DP12[1,1] : 2.010 ; 마모 L1 (= 10 - 15 * cos(30) + 10 * sin(30))

$TC_DP13[1,1] : -16.160 ; 마모 L2 (= -15 * sin(30) - 10 * cos(30))

$TC_DP14[1,1] : -5.000 ; 마모 L3

Y 방향으로의 L3 컴포넌트 상의 설정 데이터 SD42930 $SC_WEAR_SIGN은 프레임 회전으로 인해 발생하는 추가 문제 없이 인식될 수 있습니다.

추가 정보

선삭/연삭 공구: MD20360 $MC_TOOL_PARAMETER_DEF_MASK에 따른 공구 길이 계산

다음 머신 데이터는 직경 축이 선삭 및 연삭 공구에 사용되는 경우 마모 및 공구 길이의 평가 방식을 정의합니다.

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <Value>

<Bit>	<Value>	의미
0	0	선삭 및 연삭 공구의 경우, 가로 축의 **마모 파라미터**가 **반경 값**에 고려됩니다.
	1	선삭 및 연삭 공구의 경우, 가로 축의 **마모 파라미터**가 **직경 값**에 고려됩니다.
1	0	선삭 및 연삭 공구의 경우, 가로 축의 **공구 길이 컴포넌트**가 **반경 값**으로 고려됩니다.
	1	선삭 및 연삭 공구의 경우, 가로 축의 **공구 길이 컴포넌트**가 **직경 값**으로 고려됩니다.

개입된 바이트가 설정된다면, 관련 입력에 0.5의 팩터로 가중치가 주어집니다. SETTCOR를 사용한 수정은 시행되고, 그리하여 총 효과적 공구 길이 변경은 <CorVal>에서 이전된 값과 같습니다. 길이를 계산할 때 머신 데이터 MD20360 $MC_TOOL_PARAMETER_DEF_MASK의 결과로 길이가 0.5의 팩터를 통해 평가된다면, 이 컴포넌트의 보정은 반드시 이전된 값의 두 배를 이용하여 실현되어야 합니다.

예제

MD20360 $MC_TOOL_PARAMETER_DEF_MASK = 2(공구 길이는 반드시 0.5의 팩터를 사용하여 직경 축에서 평가되어야 함).

축 X는 직경 축입니다.

프로그램 코드	코멘트
N10 DEF REAL _LEN[11]	
N20 DEF REAL _CORVAL[3]	
N30 $TC_DP1[1,1]=500	; 공구 유형
N40 $TC_DP2[1,1]=2	; 절삭날 위치
N50 $TC_DP3[1,1]=3	; 기하 - 길이 1
N60 $TC_DP4[1,1]=4	; 기하 - 길이 2
N70 $TC_DP5[1,1]=5	; 기하 - 길이 3
N80 _CORVAL[0]=1	
N90 _CORVAL[1]=1	
N100 _CORVAL[2]=1	
N110 T1 D1 G18 G0 X0 Y0 Z0	; ==> MCS 위치 X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)	
N130 T1 D1 X0 Y0 Z0	; ==> MCS 위치 X2.5 Y6 Z5
N140 R3=$TC_DP3[1,1]	; = 5. = (3.000 + 2.*1.000)
N150 R4=$TC_DP4[1,1]	; = 5. = (4.000 + 1.000)
N160 R5=$TC_DP5[1,1]	; = 6. = (5.000 + 1.000)
N170 M30	

각 축에서 공구 길이 보정은 1mm이어야 합니다(N80~N100). 따라서 1mm는 길이 L2 및 L3에서 원래 길이에 추가됩니다. 요구되는 것처럼 총 길이를 1mm만큼 변경하기 위해 보정값(2mm)의 두 배가 L1의 원래 공구 길이에 추가됩니다. 블록 N110 및 N130에서 접근된 위치가 비교해보면, 각 축 위치가 1mm만큼 바뀌는 것을 볼 수 있습니다.

4.13.14 총 길이 L1, L2, L3의 좌표 축으로의 지정을 읽습니다(LENTOAX).

LENTOAX 기능은 **활성** 공구의 공구 길이 L1, L2 및 L3을 가로 좌표, 세로 좌표 및 직교 좌표에 지정하는 것에 대한 정보를 제공합니다. 가로 좌표, 세로 좌표 및 직교 좌표를 기하 축에 지정하는 것은 프레임과 활성 평면의 영향을 받습니다(G17 - G19).

공구($TC_DP3[<t>,<d>] to $TC_DP5[<t>,<d>])의 기하 컴포넌트만이 고려됩니다. 즉, 다른 컴포넌트를 위한 다른 축 지정(예: 마모)은 결과에 영향을 미치지 않습니다.

구문

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

기본 원칙

LENTOAX(...):	활성 공구의 공구 길이 L1, L2, L3의 좌표 축으로의 지정을 읽기 위해 사전에 정의된 기능		
	블록에서만:		예
<Status>:	기능 반환값. 음수는 에러 상태를 의미합니다.		
	데이터 유형:		INT
	값:	0	기능 OK <AxInd>에 제공된 정보는 설명하기에 충분합니다 (모든 공구 길이 컴포넌트는 기하 축과 평행함).
		1	기능은 OK이지만, <Matrix>의 내용은 정확한 설명을 위해 반드시 평가되어야 합니다(공구 길이 컴포넌트는 기하 축과 평행하지 않음).
		-1	파라미터 <Coord>에서의 유효하지 않은 열.
		-2	활성 공구 없음.
파라미터			
1 <AxInd>:	공구 길이 컴포넌트가 기하 축에 대해 평행하다면 길이 컴포넌트 L1~L3에 지정된 축 지수는 <AxInd> 배열에서 반환됩니다. • <AxInd> [0]: 가로 좌표 • <AxInd> [1]: 세로 좌표 • <AxInd> [2]: 직교 좌표		
	데이터 유형:		INT[3]
	값:	0	지정이 존재하지 않음(축이 존재하지 않음)
		1 ... 3 또는 -1 ... -3	해당 좌표 축에서 효과적인 길이의 개수. 공구 길이 컴포넌트가 음의 좌표 방향을 가리키면 기호는 음성입니다.
	모든 길이 컴포넌트가 기하 축에 대해 평행이거나 역평행인 것은 아니라면, 공구 길이 컴포넌트를 포함하는 축 지수는 <AxInd>에서 반환됩니다. 이 경우(기능이 다른 이유로 오류를 반환하지 않는다면), 반환 값은 <Status> = 1입니다. 그러면 공구 길이 컴포넌트 L1 ~ L3을 기하 축 1~3으로 매핑하는 것이 두 번째 파라미터 <Matrix>의 내용에 의해 완전하게 설명됩니다.		

2	<Matrix>:	좌표 축(가로 좌표, 세로 좌표 및 직교 좌표) 벡터에 대한 공구 길이 (L1=1, L2=1, L3=1) 벡터를 표현하는 매트릭스, 즉 공구 길이 컴포넌트는 L1, L2, L3 순서대로 열에 지정되고 축은 가로 좌표, 세로 좌표 및 직교 좌표 순서대로 라인에 할당됩니다.		
		데이터 유형:		REAL
		좌표 축에 속하는 기하 축을 사용할 수 없다고 하더라도, 즉 <AxInd>에서의 해당 입력이 0이라고 하더라도, 모든 요소는 항상 매트릭스에서 유효합니다.		
3	<Coord>:	할당에 적용되는 좌표계(옵션)		
		데이터 유형:		STRING
		문자:	MCS M	공구 길이는 머신 좌표계에 표시됩니다.
			BCS B	공구 길이는 기본 좌표계에 표시됩니다.
			WCS W	공구 길이는 공작물 좌표계에 표시됩니다(초기 설정).
			KCS K	공구 길이는 키네마틱 변환의 공구 좌표계에 표시됩니다.
			TCS T	공구 길이는 공구 좌표계에 표시됩니다.
		열 문자의 표기법(대문자 또는 소문자)은 임의적인 것입니다. 파라미터 <Coord>가 명시되지 않으면, WCS가 사용됩니다(초기 설정).		

참고

TCS에서 모든 공구 길이 컴포넌트는 항상 축에 대해 평행하거나 역평행합니다.

컴포넌트는 미러링이 활성화된 상태일 때에만 역평행일 수 있고 다음 설정 데이터가 활성화됩니다.

SD42900 $SC_MIRROR_TOOL_LENGTH(미러링 시의 신호 변경 공구 길이)

예제

표준 애플리케이션, G17을 위한 밀링 공구.

L1은 Z(직교 좌표)에서 적용되고, L2는 Y(세로 좌표)에서 적용되고, L3은 X(가로 좌표)에서 적용됩니다.

다음과 같은 형태의 기능 호출:
<Status>=LENTOAX(<AxInd>,<Matrix>,"WCS")

결과 파라미터 <AxInd>는 다음 값을 포함합니다.

<AxInd>[0] = 3

<AxInd>[1] = 2

<AxInd>[2] = 1

또는, 요약하면: (3, 2, 1)

이 경우, 관련 매트릭스(<Matrix>)는 다음과 같습니다:

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

G17에서 G18 또는 G19로의 변경은 결과를 바꾸지는 않습니다. 왜냐하면 기하 축에 대한 길이 컴포넌트의 할당이 가로 좌표, 세로 좌표 및 직교 좌표의 할당과 같은 방식으로 변경되기 때문입니다.

Z의 60도 프레임 회전은 이제 활성 G17으로 프로그램됩니다. 예
ROT Z60

직교 좌표의 방향(Z 방향)은 바뀌지 않고 그대로 입니다. L2의 메인 컴포넌트는 새로운 X축 방향에 놓입니다. L1의 새로운 컴포넌트는 이제 음의 Y축 방향으로 놓입니다. 결과적으로 반환값(<Status>)은 "1"이고, <AxInd>는 값(2, -3, 1)을 포함합니다.

이 경우, 관련 매트릭스(<Matrix>)는 다음과 같습니다:

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

4.14 경로 이송 동작

4.14.1 이송 속도 특성 (FNORM, FLIN, FCUB, FPO)

이송 속도 특성을 유연하게 정의할 수 있도록 DIN 66025에 따른 이송 속도 프로그래밍에 직선 및 입체 특성을 추가하였습니다.

큐빅 특성은 직접 프로그램 하거나 보간 스플라인으로 프로그램 할 수 있습니다. 이를 통해 가공할 공작물의 굴곡에 따라 연속적으로 완만한 속도 특성을 프로그램할 수 있습니다.

이 속도 특성을 통해 저크가 없는 가속 변경과 균일한 공작물 표면의 생산이 가능합니다.

구문

```
F… FNORM
F… FLIN
F… FCUB
F=FPO(…,…,…)
```

의미

FNORM	DIN 66025에 따른 **일반** 이송 속도(기본 설정) 이송 속도 값은 블록의 이송 경로의 기능에 의해 정의되며, 그 이후 모달 값으로 유효화됩니다.
FLIN	**선형** 경로 속도 특성 이송 속도 값은 블록 시작부터 블록 끝의 현재 값에서 이송 경로를 통해 직선으로 접근하며 그 이후에는 모달 값으로 유효화됩니다. 이송 속도 특성 FLIN은 G96/G961 및 G97/G971이 **아니라** G93, G94 및 G95로 활성화됩니다. G95와 오버라이드 또는 동기 동작으로 인한 스핀들 속도 변경으로 동기화가 약간 지연될 수 있습니다.
FCUB	**큐빅** 경로 속도 특성 블록별로 프로그램된 F 값은 블록 끝을 기준으로 스플라인에 의해 연결됩니다. 스플라인은 이전 또는 이후의 이송 속도 함수에 대한 접선에서 시작하고 끝납니다. 블록에서 F 주소가 누락된 경우, 프로그램 될 마지막 F 값이 사용됩니다. 이송 속도 특성 FCUB는 G95, G96/G961 및 G97/G971이 **아니라** G93 및 G94로 활성화됩니다.
F=FPO(...,...,...)	**폴리노미얼**을 통한 경로 속도 특성 F 주소는 현재 값에서 블록 끝까지 폴리노미얼을 통해 이송 속도 특성을 정의합니다. 종료값은 그 이후 모달 값으로 유효화됩니다.

경로 이송 특성 프로그램밍을 위한 기능은 프로그램된 이송 속도 특성과 상관 없이 적용됩니다.

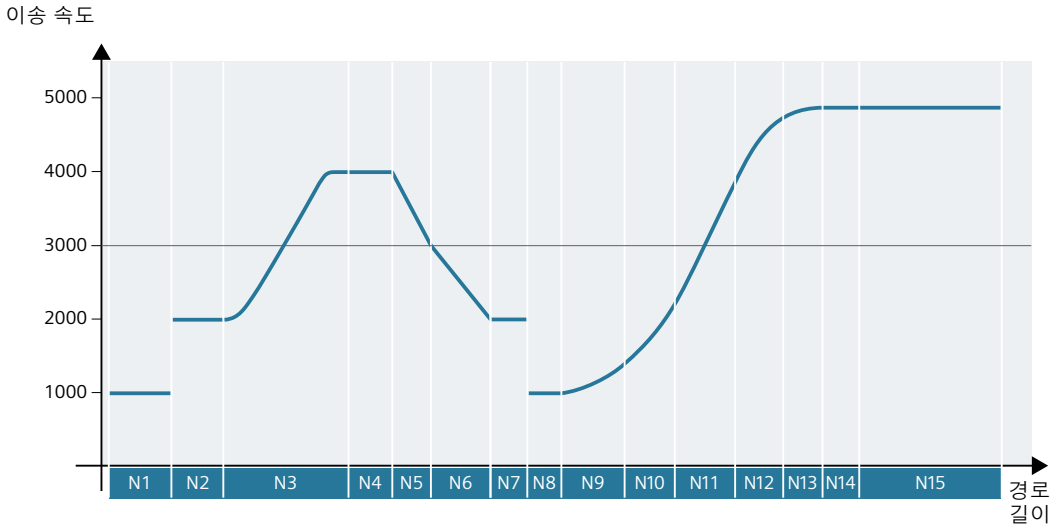
프로그램 가능 이송 속도 특성은 G90 또는 G91과 상관 없이 항상 절대적입니다.

참고

곡선화된 경로 섹션에서 이송 속도 최적화

이송 속도 폴리노미얼 F=FPO() 및 이송 속도 스플라인 FCUB는 형상 CFC에서 항상 일정한 이송 속도로 이송되어야 합니다. 이를 통해 연속 가속 목표 이송 속도 특성을 생성할 수 있습니다.

예: 다양한 피드 특성



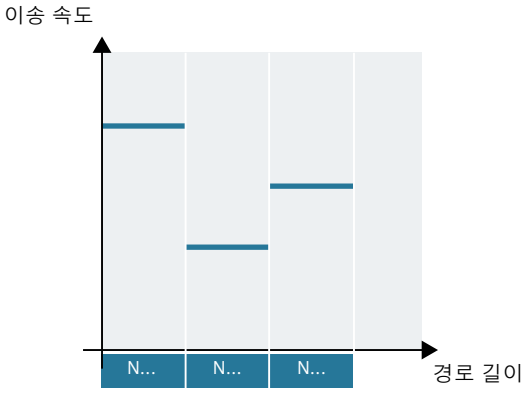
프로그램 코드	설명
N1 F1000 FNORM G1 X8 G91 G64	; 일정 이송 속도 특성, 증분 치수 데이터
N2 F2000 X7	; 지령치 속도 단계 변경
N3 F=FPO(4000, 6000, -4000)	; 블록 끝에서 이송 속도 4000으로 폴리노미얼을 통한 피드 특성
N4 X6	; 폴리노미얼 이송 속도 4000을 모달 값으로 적용
N5 F3000 FLIN X5	; 직선 피드 특성
N6 F2000 X8	; 직선 피드 특성
N7 X5	; 직선 이송 속도를 모달 값으로 적용
N8 F1000 FNORM X5	; 가속 단계 변경을 사용한 일정한 이송 속도 특성
N9 F1400 FCUB X8	; 블록에 프로그램된 다음의 모든 F 값이 스플라인과 연결
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; 스핀들 특성 끄기

프로그램 코드	설명
N14 FNORM X5	
N15 X20	

추가 정보

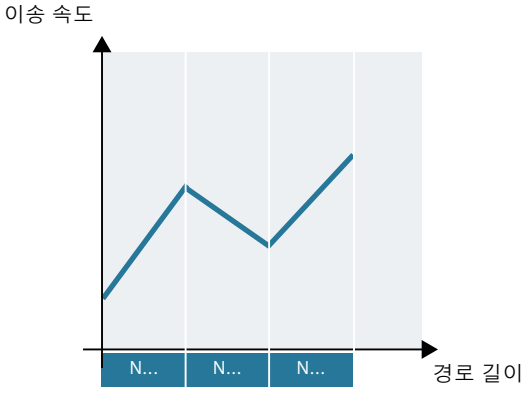
FNORM

이송 속도 주소 F는 DIN 66025에 따라 경로 이송 속도를 상수 값으로 정의합니다.



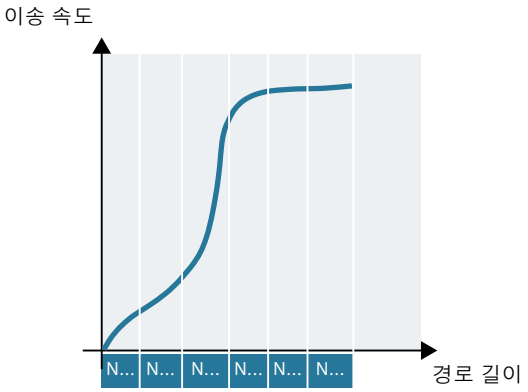
FLIN

이 이송 속도 특성은 블록 끝까지 현재 이송 속도 값에서 프로그램된 F 값으로 직선으로 접근합니다.



FCUB

이 이송 속도는 블록 끝까지 현재 피드 값에서 프로그램된 F 값으로 큐빅 특성에 따라 접근합니다. 제어 시스템이 스플라인을 사용하여 FCUB가 활성인 모든 프로그램된 이송 속도 값을 넌모달로 연결합니다. 여기에서 이송 속도 값은 스플라인 보간을 계산하기 위한 보간점 역할을 합니다.



F=FPO(... , ... , ...)

이 이송 속도 특성은 폴리노미얼을 통해 직접 프로그래밍합니다. 폴리노미얼 계수는 폴리노미얼 보간과 동일한 방법으로 지정됩니다.

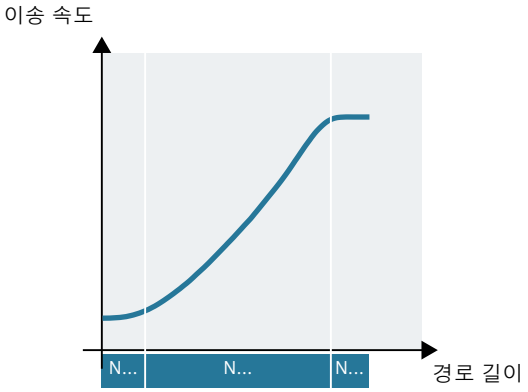
예:

```
F=FPO(endfeed, quadf, cubf)
```

endfeed, quadf 및 cubf는 사전에 정의된 변수입니다.

endfeed	블록 끝의 이송 속도
quadf	사분원 폴리노미얼 계수
cubf	큐빅 폴리노미얼 계수

FCUB가 활성화된 상태에서, 스플라인은 블록 시작과 끝에서 FPO을 통해 정의된 특성에 직선적으로 연결됩니다.



컴프레서 기능 동작 활성화

활성 컴프레서 기능(COMP...)과 여러 블록을 결합해 스플라인 세그먼트를 생성하는 경우 다음이 적용됩니다.

FNORM:	그룹 내 마지막 블록의 F 워드가 스플라인 세그먼트에 적용됩니다.
FLIN:	그룹 내 마지막 블록의 F 워드가 스플라인 세그먼트에 적용됩니다. 프로그래밍된 F 값은 세그먼트의 끝까지 적용되고 이후에는 직선으로 접근합니다.
FCUB:	생성된 이송 속도 스플라인은 머신 데이터 MD20172 $MC_COMPESS_VELO_TOL에 정의된 최대 값만큼 프로그램된 종점에서 벗어납니다.
F=FPO(...,...,...):	이 블록은 압축되지 않았습니다.

4.14.2 가속 특성

4.14.2.1 가속 모드 (BRISK, BRISKA, SOFT, SOFTA, DRIVE, DRIVEA)

다음의 가공 프로그램 명령어가 가속 모드를 프로그래밍하는데 사용됩니다:

- "BRISK, BRISKA"
  단일 축 또는 경로 축은 프로그램된 이송 속도에 도달할 때까지 (**저크 제한 없는 가속**) 최고 가속으로 이송합니다.
- "SOFT, SOFTA"
  단일 축 또는 경로 축은 프로그램된 이송속도에 도달할 때까지 (**저크 제한의 가속**) 일정한 가속으로 이송합니다.
- "DRIVE, DRIVEA"
  단일 축 또는 경로 축은 프로그램된 속도 한계 (MD 설정!) 까지 최고 가속으로 이송합니다. 가속 속도는 이때 프로그램된 이송속도에 도달할 때까지 감소됩니다 (MD 설정).

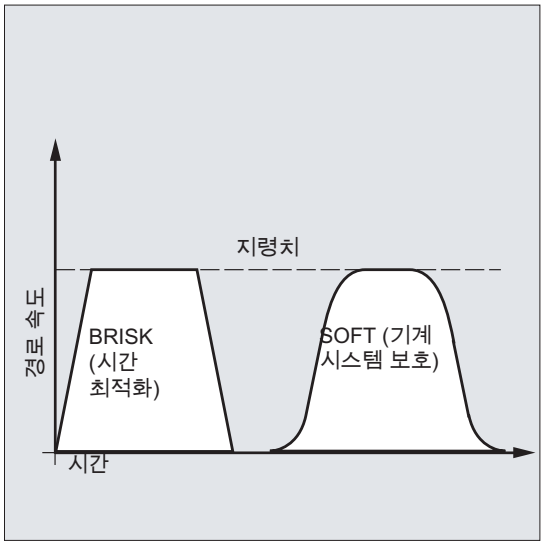


그림 4-13 BRISK와 SOFT로의 경로 속도 곡선

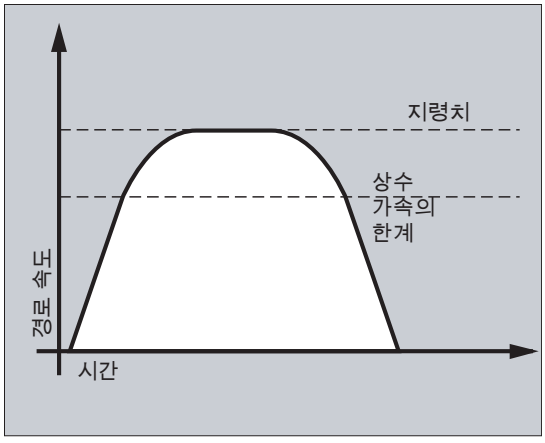


그림 4-14 DRIVE로의 경로 속도 곡선

구문

```
BRISK
BRISKA(<axis1>,<axis2>,…)
SOFT
SOFTA(<axis1>,<axis2>,…)
DRIVE
DRIVEA(<axis1>,<axis2>,…)
```

의미

BRISK:	경로 축의 "저크 제한 없는 가속" 활성화 명령어
BRISKA:	단일 축 이동의 "저크 제한 없는 가속" 활성화 명령어 (JOG, JOG/INC, 포지셔닝 축, 진동 축 등).
SOFT:	경로 축의 "저크 제한의 가속" 활성화 명령어
SOFTA:	단일 축 이동의 "저크 제한의 가속" 활성화 명령어 (JOG, JOG/INC, 포지셔닝 축, 진동 축 등).
DRIVE:	경로 축의 환경 설정된 속도 한계 (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) 가속 활성화 명령어
DRIVEA:	단일 축 이동의 환경 설정된 속도 한계 (MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT) 이상의 감소된 가속 활성화 명령어 (JOG, JOG/INC, 포지셔닝 축, 진동 축 등).
(<axis1>,<axis2>, ...):	호출된 가속 모드가 적용될 단일 축.

보충 조건

가공 중 가속 모드 변경

가공 동안 가공 프로그램에서 가속 모드가 변경되면 (BRISK ↔ SOFT) 연속 이송 모드에서도 전환 동안 블록 종점의 정위치 정지에서 블록 변경이 있게 됩니다.

예

예 1: SOFT 및 BRISKA

프로그램 코드

```
N10 G1 X… Y… F900 SOFT
N20 BRISKA(AX5,AX6)
...
```

예 2: DRIVE 및 DRIVEA

프로그램 코드

```
N05 DRIVE
N10 G1 X… Y… F1000
N20 DRIVEA (AX4, AX6)
...
```

4.14.2.2 다음 축에 대한 가속의 영향(VELOLIMA, ACCLIMA, JERKLIMA)

축 커플링 (페이지 1014)(접선 수정, 커플링 모션, 마스터 값 커플링, 전자 장치)의 경우 1개 이상의 마스터 축/스핀들에 따라 추종 축/스핀들이 이송됩니다.

추종 축/스핀들의 동적 제한범위는 가공 프로그램 또는 동기 동작에서 VELOLIMA, ACCLIMA 및 JERKLIMA 코드를 이용해 조작할 수 있습니다. 축 커플링이 이미 활성 상태인 경우도 조작이 가능합니다.

참고

JERLIMA 코드는 일부 유형의 커플링에는 사용할 수 없습니다.

- ACCLIMA 및 VELOLIMA는 스핀들 작동 중에 스핀들의 전체 동작에 작용합니다(예: 커플링 해제된 경우도 해당). 변수 $AC_SMAXACC_INFO 또는 $AC_SMAXVELO_INFO에서 식별자 22로 제한을 표시합니다.
- 스핀들 작동에서 VELOLIMA는 100 퍼센트로 제한됩니다.
- 허용 속도와 관련하여 스핀들 커플링에 다음이 적용됩니다.
  커플링 동작이 이미 최대 속도를 사용하여 프로그래밍된 기본 동작이 중지되면, 알람 22015 "추종 스핀들에 추가 동작을 위한 동력이 없음" 이 표시됩니다.

구문

```
VELOLIMA(<axis>)=<value>
ACCLIMA(<axis>)=<value>
JERKLIMA(<axis>)=<value>
```

의미

VELOLIMA:	파라미터화된 최대 **속도** 수정을 위한 명령어
ACCLIMA:	파라미터화된 최대 **가속도** 수정을 위한 명령어
JERKLIMA:	파라미터화된 최대 **저크** 수정을 위한 명령어
<axis>:	동적 제한범위를 수정해야 하는 추종 축
<Value>:	백분율 수정 값

예

예 1: 추종 축(AX4)의 동적 한계 수정

프로그램 코드	설명
...	
VELOLIMA[AX4]=75	; 머신 데이터에 저장된 최대 축 속도의 75%로 한계 수정
ACCLIMA[AX4]=50	; 머신 데이터에 저장된 최대 축 가속의 50%로 한계 수정
JERKLIMA[AX4]=50	; 머신 데이터에 저장된 최대 축 저크의 50%로 한계 수정
...	

예 2: 전자 기어

축 4는 "전자 기어" 커플링을 통해 X 축에 커플링됩니다. 추종 축의 가속 능력은 최고 가속의 70%로 제한됩니다. 최고 허용 속도는 최고 속도의 50%로 제한됩니다. 커플링이 성공적으로 활성화되면 최고 허용 속도는 다시 100%가 됩니다.

프로그램 코드	설명
...	
N120 ACCLIMA[AX4]=70	; 감소된 최대 가속.
N130 VELOLIMA[AX4]=50	; 감소된 최대 속도.
...	
N150 EGON(AX4,"FINE",X,1,2)	; EG 커플링 활성화.
...	
N200 VELOLIMA[AX4]=100	; 100% 최대 속도.
...	

예 3: 정적 동기 동작으로 마스터 값 커플링 조정

축 4는 마스터 값 커플링에 의해 X에 커플링됩니다. 가속 반응은 위치 100%에서 정적 동기 동작 2에 의해 80% 위치까지 제한됩니다.

프로그램 코드	설명
...	
N120 IDS=2 WHENEVER $AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80	; 동기 동작
N130 LEADON(AX4, X, 2)	; 마스터 값 커플링 ON
...	

4.14.2.3 기술별 동적 값의 활성화(DYNNORM, DYNPOS, DYNROUGH, DYNSEMIFIN, DYNFINISH, DYNPREC)

다양한 기술 가공 단계에 대한 동적 응답은 G그룹 59 '경로 보간을 위한 동적 응답 모드' 명령으로 활성화할 수 있습니다.

동적 값과 G 명령은 구성될 수 있으며, 머신 데이터 설정에 따라 달라질 수 있습니다.

추가 정보: 기능 매뉴얼 기본 기능

구문

동적 값 활성화:
DYNNORM/DYNPOS/DYNROUGH/DYNSEMIFIN/DYNFINISH/DYNPREC

참고

동적 값은 관련 G 명령이 프로그래밍된 블록에 이미 활성화되어 있습니다. 가공이 정지되지 않습니다.

특정 필드 요소 읽기 또는 쓰기:
R<m>=$MA...[n,X]
$MA...[n,X]=<value>

의미

DYNNORM:	**일반 동적 응답** 활성화	
DYNPOS:	**포지셔닝 모드, 태핑**을 위한 동적 응답 활성화	
DYNROUGH:	**황삭**을 위한 동적 응답 활성화	
DYNSEMIFIN:	**준정삭**을 위한 동적 응답 활성화	
DYNFINISH:	**정삭**을 위한 동적 응답 활성화	
DYNPREC:	**스무딩 정삭**을 위한 동적 응답 활성화	
R<m>:	번호 <m>이 포함된 R-파라미터	
$MA...[n,X]:	동적 응답에 영향을 주는 필드 요소가 포함된 머신 데이터	
<n>:	배열 인덱스	
	값 범위:	0 ... 5
	0	일반 동적 응답 (DYNNORM)
	1	포지셔닝 모드의 동적 응답 (DYNPOS)
	2	황삭의 동적 응답 (DYNROUGH)
	3	준정삭을 위한 동적 응답(DYNSEMIFIN)
	4	정삭을 위한 동적 응답(DYNFINISH)
	5	스무딩 정삭을 위한 동적 응답(DYNPREC)

<X>:	축 주소
<Value>:	동적 값

예

예 1: 동적 값 활성화

프로그램 코드	설명
DYNNORM G1 X10	; 초기 설정
DYNPOS G1 X10 Y20 Z30 F…	; 포지셔닝 모드, 태핑
DYNROUGH G1 X10 Y20 Z30 F10000	;황삭
DYNSEMIFIN G1 X10 Y20 Z30 F2000	; 준정삭
DYNFINISH G1 X10 Y20 Z30 F1000	;정삭
DYNPREC G1 X10 Y20 Z30 F600	; 스무딩 정삭

예 2: 특정 필드 요소 읽기 또는 쓰기

황삭의 최고 가속, 축 X

프로그램 코드	설명
R1=$MA_MAX_AX_ACCEL[2, X]	; 읽기
$MA_MAX_AX_ACCEL[2, X]=5	; 쓰기

4.14.3 피드포워드 제어 이송 (FFWON, FFWOF)

피드포워드 제어는 형상작업을 할 때 속도에 의한 과대 이송을 감소시킵니다. 피드포워드 제어를 사용한 이송에서는 경로 정확도가 향상되고 그에 따라 가공 결과도 개선됩니다.

구문

```
FFWON
FFWOF
```

의미

FFWON:	피드포워드 제어를 **활성화**하는 명령어
FFWOF:	피드포워드 제어를 **비활성화**하는 명령어

참고

피드포워드 제어와 피드포워드 제어로 이송하게 될 경로 축의 종류는 머신 데이터에서 지정됩니다.

디폴트: 속도 제어 피드포워드 제어

옵션: 가속도 제어 피드포워드 제어

예

프로그램 코드
N10 FFWON
N20 G1 X… Y… F900 SOFT

4.14.4 프로그램 가능한 형상/오리엔테이션 공차 (CTOL, OTOL, ATOL)

주소 CTOL, OTOL and ATOL은 가공 프로그램에서의 컴프레서 기능, 스무딩 및 오리엔테이션 스무딩을 위해 머신 및 셋팅 데이터를 사용하여 파라미터화된 가공 공차를 조정하는 데 사용됩니다.

프로그래밍된 공차 값은 음의 값 지정을 통해 다시 프로그래밍되거나 삭제될 때까지 유효합니다. 아울러, 이들은 프로그램 종료 시 또는 리셋될 때 삭제됩니다. 파라미터화 된 공차값은 삭제 후 다시 한번 더 효과를 갖습니다.

구문

```
CTOL=<Value>
OTOL=<Value>
ATOL[<Axis>]=<Value>
```

의미

CTOL:	**형상 공차**를 프로그램하기 위한 주소			
	용도:	• 모든 컴프레서 기능 • G641 및 G644를 제외한 모든 라운딩 유형		
	전처리 정지:	아니오		
	적용:	모달		
	<Value>:	형상 공차 값은 길이로 지정됩니다.		
		유형:	REAL	
		단위:	inch/mm (현재 치수 설정에 따라)	
		값 범위:	≥ 0:	공차 값
			<0:	프로그램 된 공차값은 삭제됩니다. ⇒ 머신 또는 셋팅 데이터 내에 파라미터화 된 공차값은 다시 한번 더 효과를 갖습니다.
OTOL:	**오리엔테이션 공차**를 프로그램하기 위한 주소			
	용도:	• 모든 컴프레서 기능 • ORISON 오리엔테이션 스무딩 • G641, G644 및 OSD를 제외한 모든 스무딩 유형		
	전처리 정지:	아니오		
	적용:	모달		
	<Value>:	오리엔테이션 공차 값은 각도로 지정됩니다.		
		유형:	REAL	
		단위:	각도	
		값 범위:	≥ 0:	공차 값
			<0:	프로그램 된 공차값은 삭제됩니다. ⇒ 머신 또는 셋팅 데이터 내에 파라미터화 된 공차값은 다시 한번 더 효과를 갖습니다.

ATOL:	**축별 공차** 프로그래밍을 위한 주소			
	용도:	• 모든 컴프레서 기능 • ORISON 오리엔테이션 스무딩 • G641, G644 및 OSD를 제외한 모든 스무딩 유형		
	전처리 정지:	아니오		
	적용:	모달		
	<Axis>:	프로그램 된 공차가 적용되는 채널 축의 이름		
	<Value>:	축 공차 값은 축 유형 (직선 축 또는 로터리 축) 에 따라 길이 또는 각도로 지정됩니다.		
		유형:	REAL	
		단위:	직선 축:	inch/mm (현재 치수 설정에 따라)
			로터리 축:	각도
		값 범위:	≥ 0:	공차 값
			<0:	프로그램 된 공차값은 삭제됩니다. ⇒ 머신 또는 셋팅 데이터 내에 파라미터화 된 공차값은 다시 한번 더 효과를 갖습니다.

참고

CTOL 및 OTOL을 이용하여 프로그램 된 채널별 공차값은 ATOL을 이용하여 프로그램 된 축별 공차값보다 높은 우선순위를 가집니다.

참고

스케일링 프레임

스케일링 프레임은 축 위치와 같은 방식으로 프로그래밍된 공차에 영향을 줍니다. 즉, 상대적 공차는 계속 동일합니다.

예

프로그램 코드	코멘트
COMPCAD G645 G1 F10000	; COMPCAD 컴프레서 기능을 활성화합니다.
X... Y... Z...	; 머신 데이터 및 셋팅 데이터가 여기에 적용됩니다.
X... Y... Z...	
X... Y... Z...	
CTOL=0.02	; 여기부터 형상 공차 0.02 mm가 적용됩니다.

프로그램 코드	코멘트
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
ASCALE X0.25 Y0.25 Z0.25	; 여기부터 형상 공차 0.005 mm가 적용됩니다.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	
CTOL=-1	; 머신 데이터 및 셋팅 데이터가 여기부터 다시 적용됩니다.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

추가 정보

시스템 변수

현재 유효한 공차는 다음 시스템 변수를 통해 판독될 수 있습니다.

- 전처리 정지를 사용해 판독(가공 프로그램 및 동기 동작)
  - $AC_CTOL
    실제 메인 주행 블록이 전처리 되었을 때 채널별 형상 공차가 적용됩니다.
    형상 공차가 적용되지 않은 경우 $AC_CTOL은 기하 축의 공차 제곱 합계의 근을 반환합니다.
  - $AC_OTOL
    실제 메인 주행 블록이 전처리 되었을 때 채널별 오리엔테이션 공차가 적용됩니다.
    오리엔테이션 공차가 적용되지 않은 경우 $AC_OTOL은 오리엔테이션 변환이 활성 상태일 때 오리엔테이션 축의 공차 제곱 합계의 근을 반환합니다. 그 밖의 경우에는 '-1' 값을 반환합니다.
  - $AA_ATOL[<axis>]
    실제 메인 주행 블록이 전처리 되었을 때 축별 형상 공차가 적용됩니다.
    활성 상태인 형상 공차가 없는 경우 $AA_ATOL[<기하 축>]은 형상 공차를 기하 축 수의 근으로 나눈 값을 반환합니다.
    오리엔테이션 공차 및 오리엔테이션 변환이 활성 상태인 경우 $AA_ATOL[<오리엔테이션 축>]은 오리엔테이션 공차를 오리엔테이션 축 수의 근으로 나눈 값을 반환합니다.

참고

공차값이 프로그램 되었다면, $A 변수는 개별 기능의 공차를 구별할 만큼 차별화되지는 않습니다.

이러한 상황은 머신 데이터 및 설정 데이터가 컴프레서 기능, 스무딩 및 오리엔테이션 스무딩에 대해 서로 다른 공차를 설정한 경우에 발생할 수 있습니다. 시스템 변수는 현재 활성 상태인 기능과 함께 발생하는 가장 큰 값을 반환합니다. 예를 들어, 컴프레서 기능이 오리엔테이션 공차 0.1°로 활성 상태이고 ORISON 오리엔테이션 스무딩이 1°로 활성 상태인 경우, $AC_OTOL 변수는 값 '1'을 반환합니다. 오리엔테이션 스무딩이 비활성화되면, $AC_OTOL은 값 "0.1"을 반환합니다.

- 전처리 정지 없이 판독(가공 프로그램만 해당)
  - $P_CTOL
    현재 활성 상태인 채널별 형상 공차.
  - $P_OTOL
    현재 활성 상태인 채널별 오리엔테이션 공차.
  - $PA_ATOL
    현재 활성 상태인 축별 형상 공차.

제한 사항

CTOL, OTOL 및 ATOL을 사용해서 프로그램 된 공차 역시 간접적으로 이러한 공차에 의존하는 기능에 영향을 미칩니다.

- 지령치 값 계산에서의 현 오류 제한
- 자유 형태 표면 모드의 기본 기능

다음 스무딩 기능은 CTOL, OTOL 및 ATOL 프로그래밍에 의해 영향을 받지 **않습니다**.

- OSD를 이용한 오리엔테이션 스무딩
  OSD는 공차를 사용하지 않고 블록 트렌지션으로부터의 거리를 사용합니다.
- G644를 이용한 스무딩
  G644는 스무딩에 사용되지 않고, 공구 변경 최적화 및 가공과 관련 없는 다른 모션에 대해 사용됩니다.
- G645를 이용한 스무딩
  G645는 사실상 G642처럼 작동하고, 이에 따라 프로그램 된 공차를 사용합니다. 머신 데이터 MD33120 $MA_PATH_TRANS_POS_TOL로부터의 공차값은 곡률에 점프가 있는, 예를 들어 접선 원호/직선 트렌지션과 같은 일정하게 접선인 블록 트렌지션에 대해서만 사용됩니다. 이러한 지점에서의 라운딩 경로도 역시 많은 애플리케이션이 덜 허용적인 프로그램 된 형상 바깥에 위치하고 있습니다. 아울러, NC 프로그래머를 개입시키지 않는 곡률에서의 변화를 보정하는 것은 작고 고정된 공차를 필요로 합니다.

4.14.5 프로그래밍 형상 정확도 켜기/끄기(CPRECON, CPRECOF)

"프로그래밍 형상 정확도" 기능은 속도를 자동 조정하여 커브 형상에서 경로 에러를 감소시킵니다.

NC 프로그램에서 G 그룹 39의 모달 유효 명령을 사용해 켜거나 끕니다(프로그래밍 형상 정확도).

구문

```
CPRECON
...
CPRECOF
```

의미

CPRECON:	"프로그래밍 형상 정확도" 기능 **켜기**
CPRECOF:	"프로그래밍 형상 정확도" 기능 **끄기**

예

프로그램 코드	설명
N10 G0 X0 Y0	
N20 CPRECON	; "프로그래밍 형상 정확도" 활성화.
N30 G1 G64 X100 F10000	; 연속 경로 모드에서 10 m/min로 가공.
N40 G3 Y20 J10	; 원호 블록에서 자동 피드 제한.
N50 G1 X0	; 제한 없이 다시 피드 (10 m/min).
...	
N100 CPRECOF	; "프로그래밍 형상 정확도" 비활성화.
N110 G0 ...	

다음 또한 참고:

프로그램 가능한 형상/오리엔테이션 공차 (CTOL, OTOL, ATOL) (페이지 955)

추가 정보

형상 정확도

형상 정확도는 기계 구성(MD20470 $MC_MC_CPREC_WITH_FFW(장비 제조업체의 정보 참조))에 따라 설정 데이터 SD42450$SC_CONTPREC 또는 프로그램된 형상 공차 CTOL을 통해 지정합니다. 값이 더 작고 기하 축의 K_V 계수가 작을수록 커브 형상에서 경로 이송 속도가 더 많이 감소됩니다.

급 이송 동작을 위한 형상 정확도

커브 형상의 공작물을 가공하고 "프로그래밍 형상 정확도" 기능이 활성화된 경우 공작물을 우회할 때 코너와 코너 라운딩 블록 등에서 급 이송 시 공구 이동 중에도 지정된 형상 정확도를 유지하기 위해 속도가 감소합니다. 급 이송 동작 중에 경로 속도의 감소를 최소화하기 위해 "프로그래밍 형상 정확도" 기능에 대해 공작물 가공에서 벗어나는 급 이송 형상 정확도를 설정할 수 있습니다.

SD42451 $SC_CONTPREC_G00_ABS (급 이송 형상 정확도)

SD42451 = 0인 경우 $SC_CONTPREC[DYNNORM]에 설정된 형상 정확도가 급 이송 동작에 적용됩니다.

최소 경로 이송 속도

사용자는 다음 설정 데이터를 사용해 "프로그래밍 형상 정확도" 기능의 최소 경로 이송 속도를 지정할 수 있습니다.

SD42460 $SC_MINFEED (CPRECON의 최소 경로 이송 속도)

더 낮은 F 값이 프로그램되어 있거나 축의 동적 제한이 더 낮은 경로 속도를 강제하지 않는 한 이송 속도를 이 값보다 낮게 제한할 수 없습니다.

포지셔닝 축에 영향을 주지 않음

"프로그래밍 형상 정확도" 기능은 경로의 기하 축만 고려합니다. 포지셔닝 축의 속도에 영향을 주지 않습니다.

가공 프로그램 시작 시 및 채널 또는 프로그램 종료 리셋 후 동작

가공 프로그램 시작 시 및 채널 또는 프로그램 종료 리셋 후 G 함수 그룹 39에 대해 정의된 초기 제어 설정이 적용됩니다(장비 제조업체의 정보 참조).

4.14.6 자동 필터 전환 활성화/비활성화(AFISON, AFISOF)

사용자는 "자동 필터 전환" 기능을 사용해 NC 프로그램 안에서 이 기능이 활성화된 모든 축이 G0 동작에 대해 2번째 필터 체인으로 자동으로 전환되는 영역을 표시할 수 있습니다. 진동이 강하게 감쇠되도록 2번째 필터 체인의 파라미터를 설정하는 경우 G0 동작에 대해 더 큰 저크를 설정할 수 있습니다. 그 결과 코너 등에서 G0 동작 중에 경로 속도가 감소하고 프로그램 런타임이 단축됩니다.

구문

기능을 켜고 끄는 음성 명령은 한 블록에 각각 단독으로 있어야 합니다.

```
AFISON
...
AFISOF
```

의미

AFISON	"자동 필터 전환" 기능 **켜기**
AFISOF	"자동 필터 전환" 기능 **끄기**

예

프로그램 코드	설명
...	
N390 G1 X1100	; 필터 체인 1 적용
	; X=1100에서 중지
N400 **AFISON**	; 필터 체인 2로 자동 전환
N410 G0 X1150	; X=1150에서 중지
N420 G1 X1200	; 필터 체인 1로 자동 전환
N430 **AFISOF**	
N440 G0 X1300	; G0 블록에도 필터 체인 1 적용
N450 G1 X1400	
N460 **AFISON**	
N470 G1 X1450	
N480 G1 X1500	
N490 **AFISOF**	
N500 G1 X1600	
N510 G0 X1700	; G0 블록에도 필터 체인 1 적용
N520 **AFISON**	
N530 G1 X1750	; X=1750에서 중지
	; 필터 체인 2로 자동 전환
N540 G0 X1800	; X=1800에서 중지
	; 필터 체인 1로 자동 전환
N550 **AFISOF**	
N560 G1 X1900	
N570 G0 X2000	; G0 블록에도 필터 체인 1 적용
	; X=2100에서 중지
N580 **AFISON**	; 필터 체인 2로 자동 전환
N590 G0 X2050	
N600 G0 X2100	; X=2100에서 중지
	; 필터 체인 1로 자동 전환
N610 **AFISOF**	
N620 G0 X2200	; G0 블록에도 필터 체인 1 적용
...	

추가 정보

필수 조건

"자동 필터 전환" 기능을 사용하려면 다음 요건을 충족해야 합니다.

- 라이선스가 필요한 "저크 조정" 옵션을 설정해야 합니다.
  MD19321 $ON_TECHNO_FUNCTION_MASK_1, bit 22 = 1
- 채널에 기능이 활성화되어야 합니다.
  MD20630 $MC_AFIS_MODE = 1
- 자동 필터 전환을 위한 각 축에 기능이 활성화되어야 합니다.
  MD32332 $MD_AFIS_ENABLE = 1
- 기능이 활성화된 모든 축에서
  - 저크 제한이 활성화되어야 합니다.
    MD32400 $MA_AX_JERK_ENABLE = 1
  - 두 가지 저크 필터 유형을 선택하고(→ MD32402 $MA_AX_JERK_MODE) 설정해야 합니다.

요건을 충족하지 못하면 알람 14782 또는 26380이 출력됩니다.

켤 때/끌 때 중지

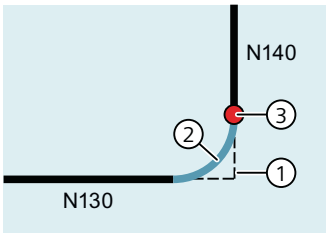
기능을 켜고 끌 때 위치 지령치 필터 체인을 전환해야 하는 경우에만 경로 동작이 중지됩니다.

스무딩

활성 스무딩(G64x) 중에 필터 체인의 중지와 자동 전환이 발생할 경우 스무딩 형상에 영향을 주지 않습니다.

예:

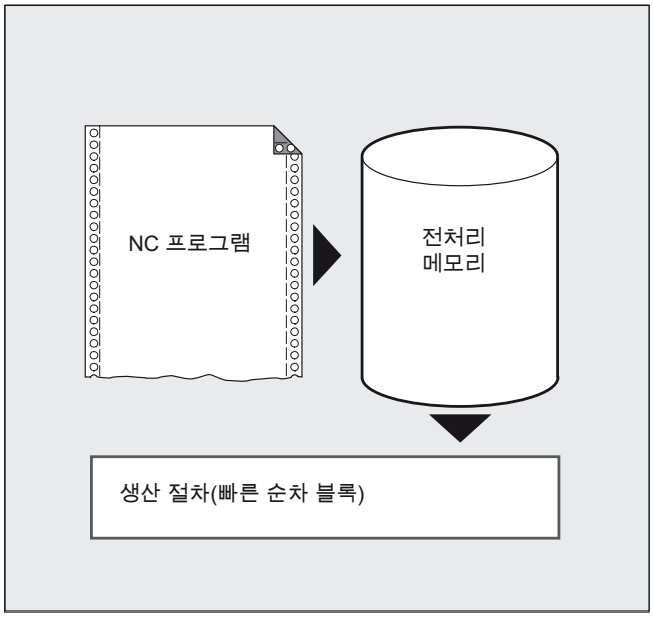
```
N100 G642 F1000 CTOL=10 CTOLG0=10
N110 G0 X0 Y0
N120 AFISON
N130 G1 X100 Y0
N140 G0 X100 Y100
```



① 프로그램 된 경로
② 스무딩 형상
③ 중지 및 2번째 필터 체인으로 전환

4.14.7 전처리 메모리를 사용한 프로그램 순서 (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

제어 시스템은 확장 정도에 따라 특정 수량의 소위 전처리 메모리를 보유하게 됩니다. 전처리 메모리에는 프로그램 실행 전에 준비된 블록이 저장되며 그 이후, 가공을 진행하는 동안 고속 순차 블록으로 출력됩니다. 이런 순서는 짧은 경로가 고속으로 이송될 수 있도록 합니다. 잔여 제어 시간이 충분하도록 제공되었기 때문에 전처리 메모리는 항상 채워져 있습니다.



가공 단계 지명

전처리 메모리에 버퍼링될 가공 단계의 시작과 끝은 가공 프로그램에서 각각 "STOPFIFO"와 "STARTFIFO"를 사용하여 확인할 수 있습니다. 전처리 후 버퍼링된 블록은 STARTFIFO 명령 이후 또는 전처리 메모리가 완전히 채워진 경우에만 처리가 시작됩니다.

자동 전처리 메모리 제어

자동 전처리 메모리 제어는 "FIFOCTRL" 명령을 사용하여 호출합니다. "FIFOCTRL"은 처음에는 "STOPFIFO"처럼 작동하비다. 프로그래밍이 어떻게 되어 있든, 전처리 메모리가 채워지기 전에는 가공이 시작되지 않습니다. 그러나 전처리 메모리를 비울 때에 대한 반응은 다릅니다. "FIFOCTRL"의 경우 전처리 메모리가 2/3까지 채워지면 메모리가 완전히 비워지고 정지 상태까지 감속되는 것을 방지하기 위해 경로 속도를 점차 줄입니다.

전처리 정지

블록에 "STOPRE" 명령을 프로그래밍하면 블록 전처리 및 버퍼링이 중단됩니다. 다음 블록은 전처리되어 저장된 모든 블록이 완전히 실행되기 전에는 실행되지 않습니다. 선행 블록은 정위치 정지로 중단됩니다 (예: G9 사용).

유의사항
프로그램 중단 공구 옵셋 또는 스플라인 보간이 활성화되면 인접한 블록 시퀀스가 중단되기 때문에 "STOPRE" 명령을 프로그래밍해서는 안됩니다.

구문

도표 4-3 가공 단계 확인:

```
STOPFIFO
...
STARTFIFO
```

도표 4-4 자동 전처리 메모리 제어:

```
...
FIFOCTRL
...
```

도표 4-5 전처리 정지:

```
...
STOPRE
...
```

참고

"STOPFIFO", "STARTFIFO", "FIFOCTRL" 및 "STOPRE" 명령은 각각 전용 블록에 프로그래밍해야 합니다.

의미

STOPFIFO:	"STOPFIFO"는 전처리 메모리에 버퍼링될 가공 단계의 시작을 식별합니다. "STOPFIFO" 처리를 중단하고 다음 조건이 충족될 때까지 전처리 메모리를 채웁니다. • "STARTFIFO" 또는 "STOPRE" 인식 또는 • 전처리 메모리가 채워짐 또는 • 프로그램 끝에 도달함
STARTFIFO:	"STARTFIFO"는 가공 단계의 고속 처리를 시작합니다. 이와 함께 전처리 메모리가 채워집니다.
FIFOCTRL:	자동 전처리 메모리 제어 활성화
STOPRE:	전처리 정지

참고

가공 단계에 버퍼링되지 않은 동작 (기준 검색, 측정 기능 등) 을 요구하는 명령이 가공 단계에 포함된 경우에는 전처리 메모리가 채워지지 않거나 채우기가 중단됩니다.

참고

상태 데이터 ($SA...) 에 대한 액세스가 발생하는 경우 제어 시스템은 내부 전처리 정지를 생성합니다.

예: 전처리 정지

프로그램 코드	코멘트
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	; 첫 번째 측정 입력 및 직선 보간에서 프로브를 통한 측정 블록.
N40 STOPRE	; 전처리 정지.
...	

4.14.8 정지 지연 섹션 정의(DELAYFSTON, DELAYFSTOF)

사전에 정의된 DELAYFSTON 및 DELAYFSTOF 절차는 가공 프로그램에서 조건적으로 중단할 수 있는 섹션을 정의하는 데 사용됩니다(정지 지연 섹션).

참고

DELAYFSTON 및 DELAYFSTOF는 동기 동작에서는 허용되지 **않습니다**.

구문

```
DELAYFSTON
...
DELAYFSTOF
```

의미

DELAYFSTON:	정지 지연 섹션 시작에 대한 정의	
	블록에서만:	예
DELAYFSTOF:	중단 지연 섹션의 끝을 지정합니다.	
	블록에서만:	예

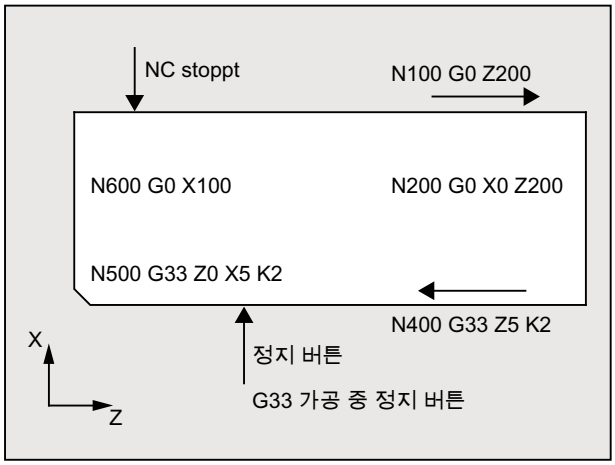
프로그래밍 예제

다음 프로그램 블록은 루프에서 반복됩니다.

프로그램 코드

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

다음 다이어그램에서처럼 사용자가 정지 지연 섹션에서 "정지"를 누르면 NC가 정지 지연 섹션 밖에서, 즉 블록 N100에서 제동을 시작합니다. 그러면 NC가 N100 시작에서 정지됩니다.



추가 정보

서브프로그램 종료

DELAYFSTOF는 DELAYFSTON이 호출되는 서브프로그램의 끝에서 암시적으로 활성화됩니다.

중첩

서브프로그램 1이 정지 지연 섹션에서 서브프로그램 2를 호출한 경우 서브프로그램 2 전체가 정지 지연 섹션이 됩니다. 특히 서브프로그램 2의 DELAYFSTOF는 영향력이 없습니다.

예제:

프로그램 코드	주석
N10010 **DELAYFSTON**	; N10xxx 프로그램 레벨 1의 블록들.
N10020 R1 = R1 + 1	
N10030 G4 F1	; 지연 정지 섹션 시작.
...	
N10040 서브프로그램2	
...	
...	; 서브프로그램 2 해석.
N20010 **DELAYFSTON**	; 적용 안됨, 시작 반복, 2번째 레벨
...	
N20020 **DELAYFSTOF**	; 적용 안됨, 다른 레벨에서 종료
N20030 RET	
N10050 **DELAYFSTOF**	; 동일 레벨에서 정지 지연 섹션 종료.
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	; 지연 정지 섹션 종료. 이제 정지는 즉시 작동함

시스템 변수

가공 프로그램 과정이 현재 정지 지연 섹션에 있는지를 결정하기 위해 다음 시스템 변수가 조회됩니다.

- $P_DELAYFST를 이용한 가공 프로그램의 경우
- $AC_DELAYFST를 이용한 동기 동작의 경우

값	의미
0	지연 정지 섹션 비활성화
1	지연 정지 섹션 활성화

4.14.9 SERUPRO을 위한 프로그램 위치 방지 (IPTRLOCK, IPTRUNLOCK)

가공을 할 때 일부 복잡한 기계적 상황에서 블록 검색 SERUPRO를 중단해야 합니다.

프로그램 가능한 중단 지점을 사용하여 "중단 지점 탐색"을 사용한 추적 불가능한 지점 이전에 개입할 수 있습니다.

NC를 아직 재입력할 수 없는 가공 프로그램 섹션에서 추적 불가능한 섹션을 정의하는 것도 가능합니다. 프로그램이 인터럽트되면 NC는 처리했던, 이후 HMI 사용자 인터페이스를 통해 검색할 수 있는 마지막 블록을 인지합니다.

구문

```
IPTRLOCK
IPTRUNLOCK
```

명령은 가공 프로그램 라인에 위치하고 있으며 프로그램 가능한 중단 지점을 허용합니다.

의미

IPTRLOCK:	추적 불가능한 프로그램 섹션의 시작
IPTRUNLOCK:	추적 불가능한 프로그램 섹션의 끝

이 두 명령은 가공 프로그램에서만 허용되지만 동기식으로는 작동하지 **않습니다**.

예

암시적 "IPTRUNLOCK"을 사용하여 두 프로그램 레벨의 추적 불가능한 프로그램 섹션을 중첩합니다. 서브 프로그램 1에 있는 암시적 "IPTRUNLOCK"은 추적 불가능한 섹션을 종료합니다.

프로그램	코드	코멘트
N10010	IPTRLOCK()	
N10020	R1 = R1 + 1	
N10030	G4 F1	; 탐색이 억제된 프로그램 섹션의 홀드 블록을 시작합니다.
...		
N10040	subprogram2	
...		; 서브프로그램 2 해석.
N20010	IPTRLOCK ()	; 적용 안됨, 시작 반복.
...		
N20020	IPTRUNLOCK ()	; 적용 안됨, 다른 레벨에서 종료
N20030	RET	
...		
N10060	R2 = R2 + 2	
N10070	RET	; 탐색이 억제된 프로그램 섹션의 끝.
N100 G4 F2		; 메인 프로그램이 계속됩니다.

중단 지점은 이후 100에서 다시 중단을 생산합니다.

추가 정보

추적 불가능한 섹션의 획득 및 찾기

검색이 불가능한 프로그램 섹션은 언어 명령 'IPTRLOCK' 및 'IPTRUNLOCK'으로 식별합니다.

'IPTRLOCK' 명령은 메인 실행(SB1)에서 실행 가능한 싱글 블록에서 인터럽트 포인터를 정지시킵니다. 이 블록은 홀드 블록 아래로 참조하게 됩니다. 프로그램이 "IPTRLOCK" 다음에 중단된 경우 HMI 사용자 인터페이스에서 이 홀드 블록을 검색할 수 있습니다.

현재 블록에서 계속

인터럽트 포인터는 "IPTRUNLOCK"이 있는 현재 블록에 다음 프로그램 섹션의 인터럽트 지점으로 저장됩니다.

일단 검색 대상이 발견되면, 홀드 블록을 사용하여 새로운 검색 대상을 반복할 수 있습니다.

사용자에 의해 편집된 중단 지점은 HMI를 통해 다시 제거되어야만 합니다.

중첩 규칙

다음 지점이 언어 명령 "IPTRLOCK" 및 "IPTRUNLOCK"과 중첩 및 서브프로그램 종료 간의 연동을 제어합니다.

1. "IPTRLOCK"는 "IPTRUNLOCK"이 호출되는 서브프로그램의 끝에서 암시적으로 활성화됩니다.
2. 추적 불가능한 섹션의 'IPTRLOCK'는 영향이 없습니다.
3. 서브프로그램 1이 추적 불가능한 섹션에 서브프로그램 2를 호출하면 서브프로그램 2 전체를 추적할 수 없게 됩니다. 특히 "IPTRUNLOCK"은 특히 서브프로그램 2에 영향을 미치지 않습니다.

추가 정보: 기능 매뉴얼 Basic Functions

시스템 변수

추적 불가능한 섹션은 가공 프로그램에서 "$P_IPTRLOCK"으로 감지할 수 있습니다.

자동 중단 지점

자동 중단 지점은 이전에 정의된 결합 유형을 자동적으로 추적 불가능한 것으로 정의합니다.
머신 데이터 사용

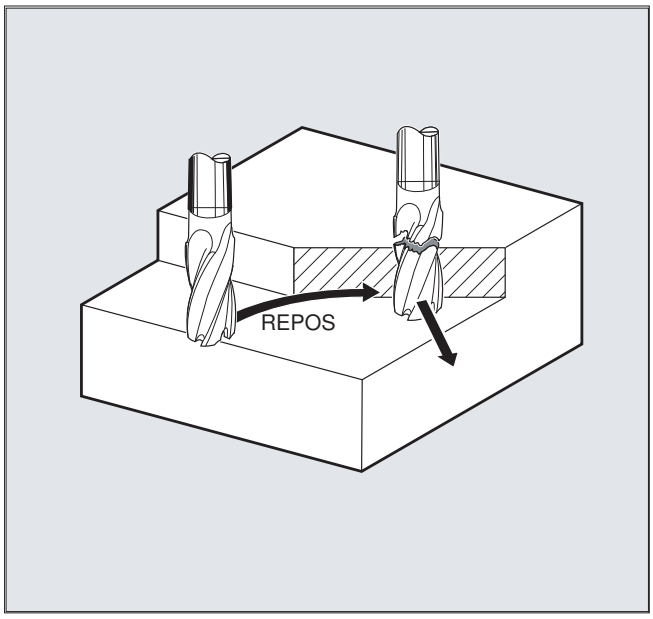
- "EGON"의 전자 기어
- "LEADON"의 축 마스터 값 커플링

자동 인터럽트 포인터가 활성화됩니다. 머신 데이터 오버랩을 통해 활성화할 수 있는 프로그래밍된 인터럽트 포인터 및 자동 인터럽트 포인터의 경우 추적이 불가능한 최대 섹션이 생성됩니다.

4.14.10 형상 리포지셔닝 (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMIBL, RMBBL, RMEBL, RMNBL)

가공 중 공구가 부러졌거나 또는 공작물을 측정하고 싶어 프로그램 실행을 중단시키고 공구를 후진시킨 경우, 프로그램에 의해 제어되고 있는 상황에서 이전 중단된 위치로 축을 복귀시킬 수 있습니다.

REPOS 명령은 ASUB에서 서브프로그램 반환으로 작용합니다 (예: M17). 이후의 블록들은 실행되지 않습니다. 프로그램 실행 중단에 대한 자세한 정보는 "인터럽트 루틴 (ASUB) (페이지 606)"을 참조하십시오.



구문

```
REPOSA RMIBL DISPR=…
REPOSA RMBBL
REPOSA RMEBL
REPOSA RMNBL
REPOSL RMIBL DISPR=…
REPOSL RMBBL
REPOSL RMEBL
REPOSL RMNBL
REPOSQ RMIBL DISPR=… DISR=…
REPOSQ RMBBL DISR=…
REPOSQ RMEBL DISR=…
REPOSQA DISR=…
REPOSH RMIBL DISPR=… DISR=…
REPOSH RMBBL DISR=…
REPOSH RMEBL DISR=…
REPOSHA DISR=…
```

의미

접근 경로 선택하기

REPOSA:	직선 상의 기하 축을 이용한 형상에 대한 리포지셔닝 모든 다른 채널 축 역시 리포지셔닝됩니다.
REPOSL:	직선 상의 기하 축을 이용한 형상에 대한 리포지셔닝 다른 축들은 명시적으로 프로그램되어야 합니다.

REPOSQ DISR=… :	DISR 사분면 상의 기하 축을 이용한 형상에 대한 리포지셔닝 다른 축들은 명시적으로 프로그램되어야 합니다.
REPOSQA DISR=… :	DISR 사분면 상의 기하 축을 이용한 형상에 대한 리포지셔닝 모든 다른 채널 축 역시 리포지셔닝됩니다.
REPOSH DISR=… :	DISR 반원 상의 기하 축을 이용한 형상에 대한 리포지셔닝 다른 축들은 명시적으로 프로그램되어야 합니다.
REPOSHA DISR=… :	DISR 반원 상의 기하 축을 이용한 형상에 대한 리포지셔닝 모든 다른 채널 축 역시 리포지셔닝됩니다.

리포지셔닝 지점 선택하기

RMIBL:	인터럽트 지점 접근
RMIBL DISPR=…:	인터럽트 지점 앞에서 mm/inch로 거리 DISPR에 있는 엔트리 포인트
RMBBL:	블록 시작점 접근
RMEBL:	블록 끝점 접근
RMEBL DISPR=… :	끝점 앞의 거리 DISPR에서 블록 종점 접근
RMNBL:	가장 가까운 경로점 접근
A0 B0 C0 :	앞으로 접근하게 될 곳에 있는 축

참고

호환성

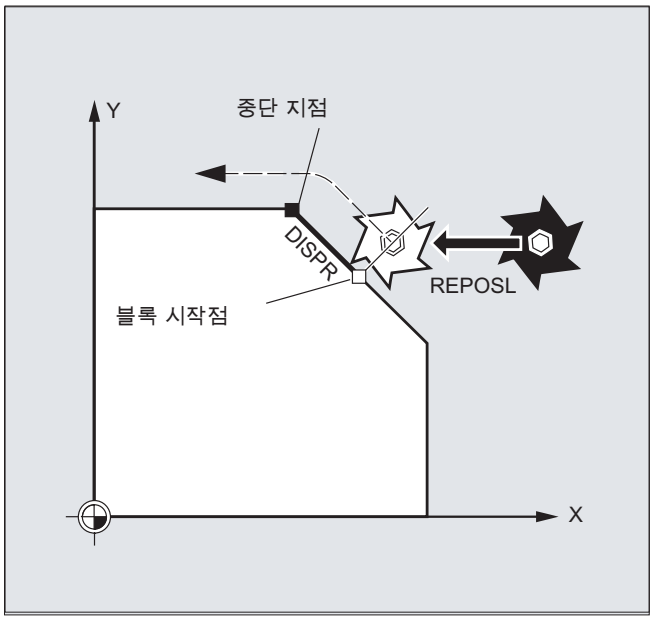
이전 소프트웨어 버전과의 호환성을 유지하기 위해, 모달 명령 RMI, RMB, RME 및 RMN을 통해서도 REPOS 접근 모드를 프로그램할 수 있습니다. ASUB 내에서 사용할 경우 PROC 구문에서 여기에 SAVE 속성을 할당해야 합니다. 그러지 않으면, 사전 설정된 RMI에서 벗어날 경우 ASUB에 사용된 모달 REPOS 접근 모드가 이후의 REPOS 프로세스에도 영향을 줍니다.

직선, REPOSA, REPOSL 상의 형상에 대한 리포지셔닝

공구는 직선을 따라 복귀 지점으로 접근합니다.

예

```
REPOSL RMIBL DISPR=6 F400
```

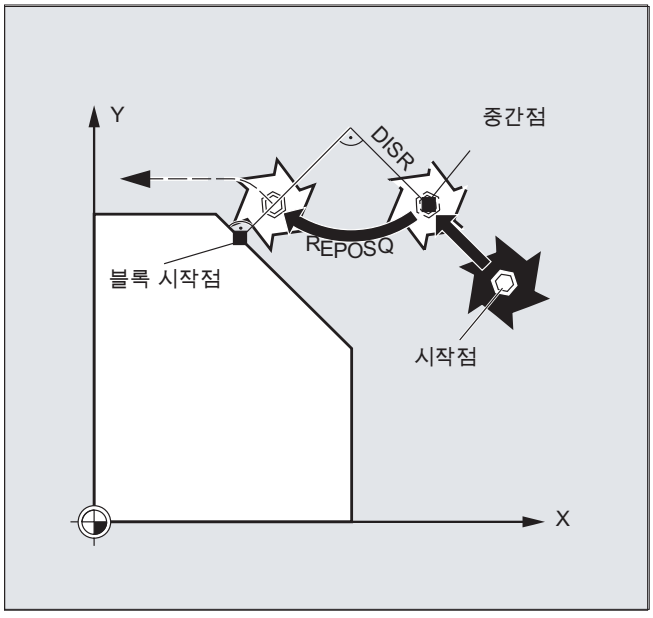


사분면, REPOSQ, REPOSQA 상의 형상에 대한 리포지셔닝

공구는 DISR=…의 반경을 사용하여 4분원을 따라 복귀 지점으로 접근합니다. 시작점과 리포지셔닝 지점 사이에 필요한 중간점은 제어 시스템이 자동으로 계산합니다.

예

```
REPOSQ RMIBL DISR=10 F400
```

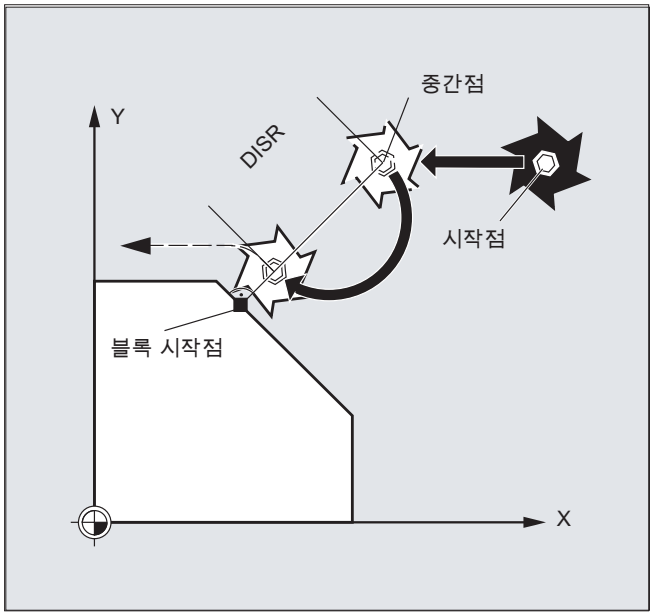


반원, REPOSH, REPOSHA 상의 형상에 대한 리포지셔닝

공구는 DISR=…의 수치를 사용하여 반원을 따라 복귀 지점으로 접근합니다. 시작점과 리포지셔닝 지점 사이에 필요한 중간점은 제어 시스템이 자동으로 계산합니다.

예

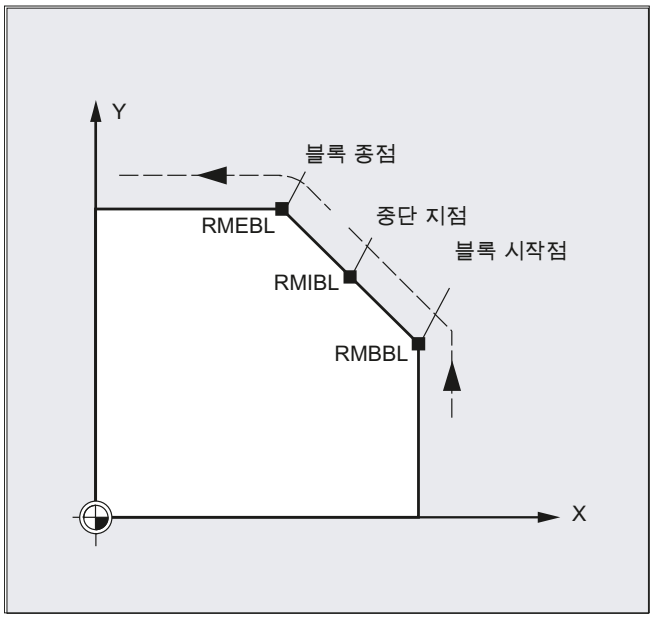
```
REPOSH RMIBL DISR=20 F400
```



리포지셔닝 지점 지정 (RMNBL을 이용한 SERUPRO 접근 제외)

프로그램 작동을 중단시킨 NC 블록을 기준으로 하여 3개의 서로 다른 복귀 지점 중 하나를 선택할 수 있습니다.

- RMIBL, 중단 지점
- RMBBL, 블록 시작점 또는 마지막 종점
- RMEBL, 블록 종점



RMIBL DISPR=... 또는 RME DISPR=...을 통해 중단점 또는 블록 종점 앞에 리포지셔닝 지점을 선택할 수 있습니다.

DISPR=...을 통해 종점 앞의 리포지셔닝 지점과 중단점 사이의 형상 거리를 mm/inch로 지정할 수 있습니다. 값이 높더라도 이 점은 블록 시작점보다 멀리 나갈 수 없습니다.

DISPR=... 명령이 프로그래밍되지 않은 경우 DISPR=0이 적용되고 이와 함께 중단 지점 (RMIBL로 지정) 또는 블록 종점 (RMEBL로 지정) 이 함께 적용됩니다.

DISPR 부호

DISPR 부호가 평가됩니다. 양수 부호인 경우, 동작은 앞과 같습니다.

음수 부호인 경우 중단점 뒤로 접근합니다. RMBBL을 사용하는 경우에는 블록 시작점 뒤로 접근합니다.

중단점과 접근점 사이의 거리는 DISPR의 값에 따라 달라집니다. 값이 높더라도 이 점은 최대한 블록 끝점에 위치할 수 있습니다.

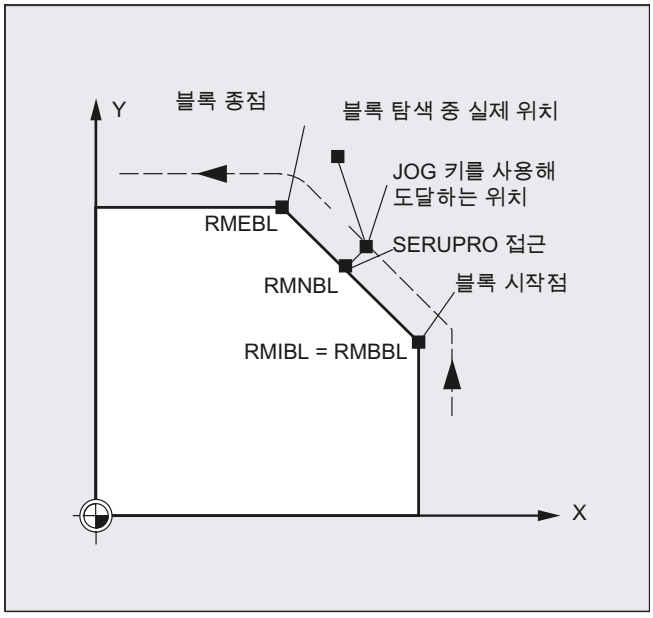
적용 예제:

센서는 클램프로 접근하는 것을 알아챕니다. 클램프를 우회하기 위해 ASUB를 시작합니다.

그런 다음 음의 DISPR을 클램프 뒤의 한 지점에 리포지셔닝하고 프로그램을 계속합니다.

RMNBL을 이용한 SERUPRO 접근

어느 위치에서건 가공이 강제로 중단된 경우 이후 이동할 거리만 처리되도록 SERUPRO 접근과 RMNBL을 사용하여 중단 지점에서 최단 경로로 접근합니다. 사용자는 중단 블록에서 SERUPRO 공정을 시작하고 JOG 키를 사용하여 대상 블록의 문제가 있는 요소 앞으로 이동합니다.



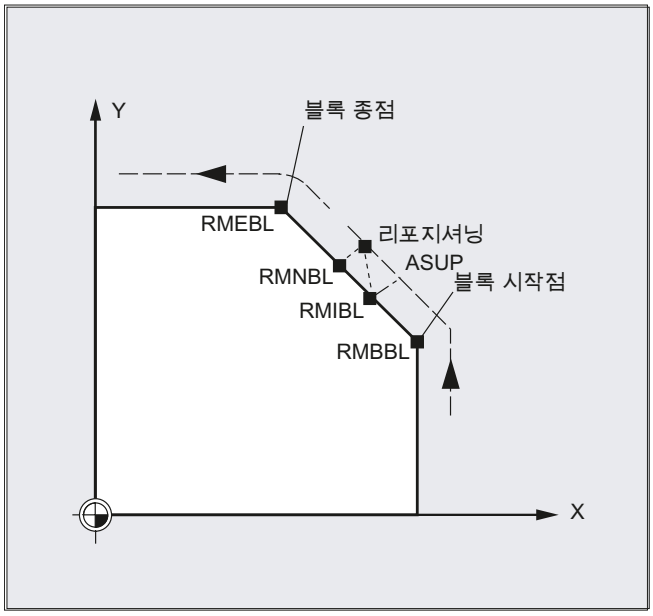
참고

SERUPRO

SERUPRO에서는 RMIBL과 RMBBL이 동일합니다. RMNBL은 SERUPRO에 제한되지 않고 일반적으로 사용 가능합니다.

RMNBL로 가장 가까운 경로 지점에서 접근

REPOSA가 해석된 경우 RMNBL로 지정한 리포지셔닝 블록은 중단 이후 전체적으로 다시 시작되지 않고 이동할 거리만 처리됩니다. 중단된 블록의 가장 가까운 경로점에 접근합니다.



유효한 REPOS 모드의 상태

중단된 블록의 유효한 REPOS 모드는 동기 동작과 $AC_ REPOS_PATH_MODE를 사용하여 읽을 수 있습니다.

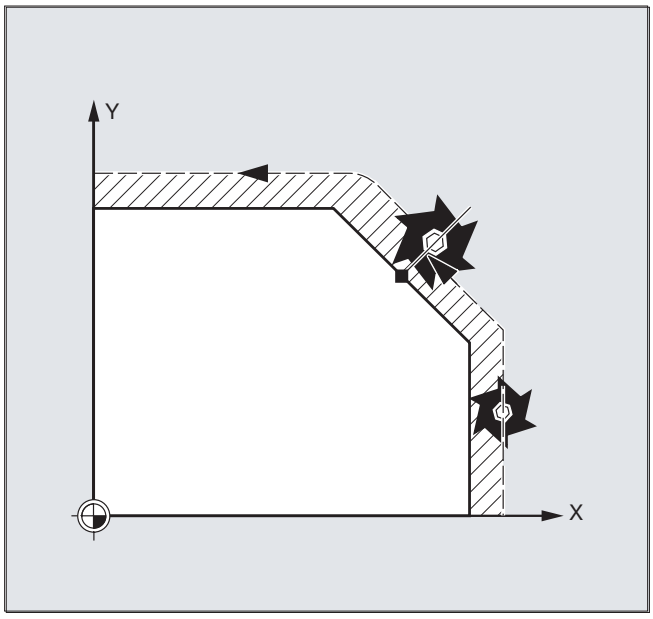
0 접근은 정의되지 않음
1 RMBBL: 시작으로 접근
2 RMIBL: 중단 지점으로 접근
3 RMEBL: 블록 끝점으로 접근
4 RMNBL: 중단된 블록의 다음 경로 지점으로 접근

새로운 공구를 사용하여 접근

공구 파손으로 인해 프로그램 작동을 중단시킨 경우, 다음이 적용됩니다.

새로운 D 번호가 프로그램된 경우, 가공 프로그램은 복귀 지점에서 수정된 공구 옵셋 값을 사용하여 계속됩니다.

공구 옵셋 값이 수정된 경우, 중단점으로 다시 접근할 수 없습니다. 이런 경우, 새로운 형상에서 중단점에 가장 가까운 지점으로 접근합니다(DISPR에 의해 수정됐을 가능성이 많음).



접근 형상

공구가 사용하는 움직임은 형상 위에서 위치가 변경되며 프로그램될 수 있습니다. 이송해야 할 축의 주소를 0으로 입력하십시오.

REPOSA, REPOSQA 및 REPOSHA 명령은 자동으로 모든 축을 리포지셔닝합니다. 개별 축에 대해 이름을 지정해야 합니다.

REPOSL, REPOSQ 및 REPOSH 명령이 프로그램된 경우 모든 기하 축이 자동으로 이송됩니다. 즉, 명령에 축을 지정할 필요가 없습니다. 다른 축은 모두 명령에서 지정해야 합니다.

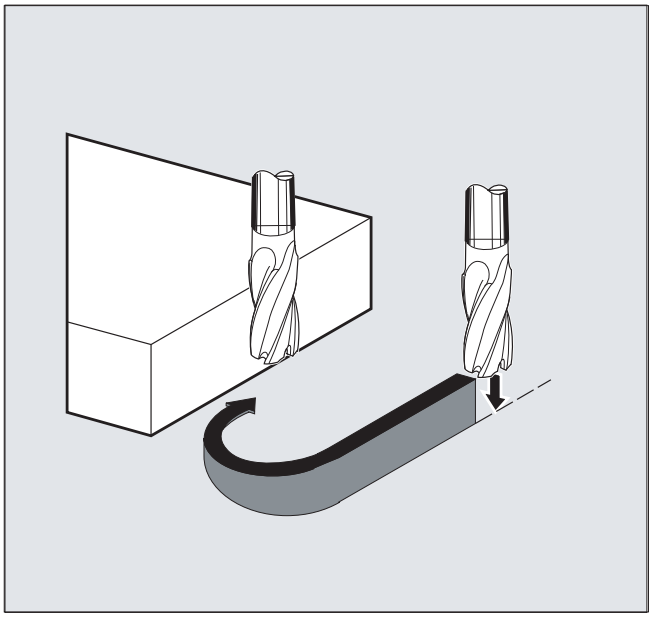
다음은 REPOSH 및 REPOSQ 원형 이동에 적용됩니다.

원호는 지정된 작업 평면 G17~G19에서 이송됩니다.

접근 블록에서 3차 형상축을 지정(절입 방향)한 경우, 복귀 지점은 공구 위치 및 절입 방향으로 프로그램된 위치가 동일하지 않을 때 나선을 따라 접근합니다.

다음 경우 직선 접근 REPOSL로 자동 전환됩니다.

- DISR에 대한 값을 지정하지 않은 경우.
- 정의된 접근 방향을 사용할 수 없는 경우 (이송 정보가 없는 블록에서 프로그램 중단).
- 현재 작업 평면에 수직이 되는 접근 방향인 경우.



4.14.11 모션 제어에 영향

4.14.11.1 최대 축 속도 또는 스핀들 속도 조정(VELOLIM)

머신 데이터를 통해 설정된 스핀들의 기어단 종속 최대 속도를 가공 프로그램에서 VELOLIM 명령을 실행하여 감소시킬 수 있습니다.

유효성

VELOLIM 작동:

- AUTOMATIC 작동 모드의 경우
- 경로 및 포지셔닝 축이 대상인 경우
- 스핀들/축 작업에서 스핀들이 대상인 경우

구문

```
VELOLIM[<Ax>]=<Value>
```

의미

VELOLIM:	속도 한도값 조정	
<Ax>:	속도 한계값을 조정해야 하는 기계 축 또는 스핀들 MD30455 $MA_MISC_FUNCTION_MASK 비트 6을 사용할 때 작동 모드(비트 6 = 0)별로 스핀들 또는 축(비트 6 = 1)을 사용할지와 관계없이 VELOLIM의 효과에 대해 설정할 수 있습니다. 조작 모드에 따라 다르게 프로그래밍하는 경우 프로그래밍을 할 때 이름을 사용하여 대상을 선택합니다. • 스핀들 작동 모드의 스핀들 식별자: S<n> • 축 모드의 축 식별자(예: 인덱스): C	
<Value>:	백분율 수정 값 MD30455 bit 6 = 0으로 설정된 축 모드의 축 또는 스핀들의 경우 보정 값은 구성된 최대 축 속도(MD32000 $MA_MAX_AX_VELO)를 나타냅니다. 설정이 MD30455 비트 6 = 1인 스핀들 또는 축 모드의 스핀들이면 수정 값이 활성 기어 단계의 최대 속도 (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])를 나타냅니다.	
	값 범위:	1 ... 100
	값 100은 속도에 영향을 미치지 않습니다.	

참고

가공 프로그램 종료 및 채널 리셋 시 응답

가공 프로그램 종료 및 채널 리셋 시 VELOLIM 응답은 머신 데이터 MD32320 $MA_DYN_LIMIT_RESET_MASK를 비트 0의 설정에 따라 달라집니다.

참고

스핀들 작동의 속도 제한

스핀들 조작 시 다음 시스템 변수를 통해 'VELOLIM'(100% 미만)이 지정한 속도 한계를 감지할 수 있습니다.

- $AC_SMAXVELO (최대 스핀들 속도)
- $AC_SMAXVELO_INFO (속도 제한 원인 식별자)

예

예 1: 기계 축 속도 제한

프로그램 코드	설명
...	
N70 VELOLIM[X]=80	; X 방향의 축 슬라이드는 축에 허용된 속도의 최대 80%까지만 이송할 수 있습니다.
...	

예 2: 속도 제한, 스핀들 1(AX5)

구성:

- MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
  (기어 유닛 1단계의 최대 속도 = 1000 rpm)
- MD30455 $MA_MISC_FUNCTION_MASK[AX5], bit 6 = 1
  (VELOLIM 프로그래밍은 프로그래밍된 식별자와 관계 없이 스핀들 조작과 축 조작에 모두 적용됨)

프로그래밍:

프로그램 코드	설명
N05 VELOLIM[S1]=90	; 스핀들 1의 최대 속도를 1000 rpm의 90%로 제한.
...	
N50 VELOLIM[C]=45	; 최대 속도를 1000 rpm의 1~45%로 제한, C는 ? S1의 축 식별자
...	

4.14.11.2 최대 축 저크 조정(JERKLIM)

JERKLIM 명령을 사용하여 경로 모션 중 축의 최대 저크(머신 데이터로 설정)를 중요 프로그램 섹션에서 감소 또는 증가시킬 수 있습니다.

요구사항

가속 모드 SOFT는 활성화되어 있어야만 합니다.

유효성

JERKLIM 작동:

- AUTOMATIC 작동 모드의 경우
- 경로 축만 해당

구문

```
JERKLIM[<Ax>]=<Value>
```

의미

JERKLIM:	저크 한계값 조정	
<Ax>:	저크 제한값을 조정하는 기계 축	
<Value>:	백분율 수정 값 수정 값은 경로 모션에 설정된 최대 축 저크 (MD32431 $MA_MAX_AX_JERK)를 참조합니다.	
	값 범위:	1 ... 200
	100 값은 저크에 영향을 주지 않습니다.	

참고

가공 프로그램 종료 및 채널 리셋 시 응답

가공 프로그램 종료 및 채널 리셋 시 JERKLIM의 동작은 머신 데이터 MD32320 $MA_DYN_LIMIT_RESET_MASK를 비트 0의 설정에 따라 달라집니다.

예

프로그램 코드	설명
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20 ACC[Y]=30	
N1200 G1 Y20 JERKLIM[Y]=200	; Y 방향에서 축 슬라이드는 축에 허용된 저크의 최대 200%까지만 가속/감속될 수 있습니다.
N1300 G1 X0 JERKLIM[X]=2	; X 방향의 축 슬라이드는 축에 허용된 저크의 최대 2%까지만 가속/감속될 수 있습니다.
N1400 G1 Y0	
M30	

4.14.11.3 최대 경로 속도 조정(FLIM)

"최대 경로 속도 조정" 기능을 사용하면 축 제한 값으로 인한 경로 동작 속도를 가공 프로그램의 중요한 프로그램 섹션에서 제한하거나 줄일 수 있습니다. 최대 경로 속도의 제한 속도 값은 주소 FLIM을 통해 프로그래밍됩니다. 프로그래밍된 값은 항상 다음 NC 리셋 또는 가공 프로그램 종료 시까지만 유효합니다.

오버라이드는 FLIM에 의해 제한되는 이송 속도에도 영향을 미칩니다. 따라서 FLIM은 오버라이드 스위치를 이용해 어느 정도 초과될 수 있습니다.

유효성

이 기능은 다음 경우에 유효합니다.

- AUTOMATIC 작동 모드의 경우
- 경로 축만 해당
- G94만 해당
- 급 이송은 해당 없음

참고

"최대 경로 속도 조절" 기능은 직선 이송 속도 G94와 함께 사용해야만 유효합니다. 이 기능은 다른 이송 유형(G93, G931, G95, G96, G97, G971, G972)과는 함께 사용할 수 없습니다.

구문

```
...
FLIM=<Value>
...
FLIM=-1
...
```

의미

FLIM	최대 경로 속도 조절을 위한 주소	
<Value>	최대 경로 속도의 제한 속도 값	
	데이터 유형:	REAL
	값 범위:	$1.0 * 10^{-6}$ ... $1.0 * 10^{38}$
	단위:	mm/min 또는 inch/min(사용하는 단위계에 따라 다름)
FLIM=-1	FLIM=<Value>에 의해 프로그래밍된 제한을 취소합니다.	

예

프로그램 코드	설명
...	
N1000 G0 X0 Y0 F10000 G64 G710	
N1100 G1 X20 RNDM=5	

프로그램 코드	설명
N1200 G1 Y20 **FLIM=5000**	; Y 방향의 축 슬라이드가 최대 5m/min의 속도로 이동합니다.
N1300 G1 X0 Y40	; X 방향과 Y 방향의 축 슬라이드 경로 이동은 최대 5m/min의 속도로 이동합니다.
N1400 G1 Y0 **FLIM=-1**	; Y 방향의 축 슬라이드가 10m/min의 속도로 이동합니다.
M30	

4.14.11.4 최대 경로 가속 조정(PACCLIM)

"최대 경로 가속 조정" 기능을 사용하면 축 제한 값으로 인한 경로 동작 가속을 가공 프로그램의 중요한 프로그램 섹션에서 제한하거나 줄일 수 있습니다. 최대 경로 가속의 감속 가속 값은 주소 PACCLIM을 통해 프로그래밍됩니다. 프로그래밍된 값은 항상 다음 NC 리셋 또는 가공 프로그램 종료 시까지만 유효합니다.

요구 사항

라이센스

이 옵션 기능을 사용하려면 라이센스가 필요하며("경로 가속 제한", 문서 번호: 6FC5800-0xP26-0YB0) 해당 라이센서는 라이센스 관리를 통해 하드웨어에 지정되어야 합니다.

유효성

이 기능은 다음 경우에 유효합니다.

- AUTOMATIC 작동 모드의 경우
- 경로 축만 해당

구문

```
...
PACCLIM=<Value>
...
PACCLIM=-1
...
```

의미

PACCLIM	최대 경로 가속 조절을 위한 주소	
<Value>	최대 경로 가속의 감속 가속 값	
	데이터 유형:	REAL
	값 범위:	$1.0 * 10^{-6}$ ... $1.0 * 10^{38}$
	단위:	m/s^2 또는 $inch/s^2$ (사용하는 단위계에 따름)
PACCLIM=-1	PACCLIM=<Value>에 의해 프로그래밍된 제한을 취소합니다.	

참고

PACCLIM의 효과는 설정 데이터 SD42500 $SC_SD_MAX_PATH_ACCEL(최대 경로 가속)의 효과와 비슷합니다. 그러나 SD42500와는 달리 PACCLIM은 블록 동기식으로 작동합니다.

참고

제한 값을 계산할 때 SD42500 $SC_SD_MAX_PATH_ACCEL의 값은 SD42502 $SC_IS_SD_MAX_PATH_ACCEL이 "TRUE"로 지정된 경우에만 고려됩니다.

PACCLIM과 SD42500 $SC_SD_MAX_PATH_ACCEL이 모두 활성 상태이면 두 제한 값 중 작은 값이 적용됩니다.

예

프로그램 코드	설명
...	
N1000 G0 X0 Y0 F10000 G64 G710	
N1100 G1 X20 RNDM=5	
N1200 G1 Y20 **PACCLIM=0.5**	; Y 방향의 축 슬라이드가 최대 0.5m/ss의 속도로 가속/감속합니다.
N1300 G1 X0 Y40	; X 방향과 Y 방향의 축 슬라이드 경로 이동은 최대 0.5m/ss의 속도로 가속/감속합니다.
N1400 G1 Y0	
M30	

4.14.12 커플링이 활성인 경우 블록 변경 동작 (CPBC)

CPBC 명령은 커플링이 활성일 때 가공 프로그램에서 블록 변경을 실행하기 위해 충족해야 하는 블록 변경 기준을 지정합니다.

구문

```
CPBC[<following axis>] = <criterion>
```

의미

CPBC:	커플링이 활성인 경우 블록 변경 기준	
<following axis>:	추종 축의 축 이름	
<criterion>:	블록 변경 기준	
	유형:	STRING
	값	**의미:** 블록 변경 실행
	"NOC"	커플링 상태 무시
	"IPOSTOP"	지령치 동기의 경우
	"COARSE"	실제값 "일반" 동기의 경우
	"FINE"	실제값 "미세" 동기의 경우

예

프로그램 코드

```
; 다음과 같이 블록 변경 실행
;  - 추종 축 X2와의 커플링 == 활성
;  - 지령치 동기 == 활성
CPBC[X2]="IPOSTOP"
```

4.15 축 기능

4.15.1 축 교체, 스핀들 교체 (RELEASE, GET, GETD)

한 개의 채널에서 하나 이상의 축 또는 스핀들 만 보간됩니다. 한 축을 두 개의 서로 다른 채널에서 번갈아 사용해야 하는 경우 (예: 팔레트 교환기) 먼저 현재 채널에서 활성화한 다음 다른 채널로 이송해야 합니다. 채널 간에 축 교체가 유효합니다.

축 교체 확장

전처리 중지 및 전처리와 기본 실행 사이 또는 전처리 중지 없이 축/스핀들을 교체할 수 있습니다. 또한 축 교체는 다음을 통해 수행할 수 있습니다.

- 함축적 GET/GETD를 사용한 축 컨테이너 회전 AXCTSWE 또는 AXCTWED
- 이 프로세스에서 해당 축이 다른 축과 연결된 경우 회전 상태의 프레임
- 동기 동작에 대해서는 이동 동기화 작업, "축 교체 RELEASE, GET"을 참조하십시오.

장비 제조업체

장비 제조업체의 설명서를 참조하십시오. 축을 교체하는 경우 구성 가능한 머신 데이터의 모든 채널에서 한 축을 고유하게 정의해야 하며 또한 가공 데이터를 사용하여 축 교체 특성을 설정할 수 있습니다.

구문

```
RELEASE (axis name, axis name, ...) or RELEASE (S1)
GET (axis name, axis name, ...) or GET (S2)
GETD (축 이름 등) 또는 GETD(S3)
```

GETD (GET Directly) 를 통해 축을 다른 채널에서 직접 가져옵니다. 다시 말해 다른 채널에서 이 GETD에 대해 RELEASE를 프로그래밍하지 않아야 합니다. 또한 다른 채널과의 통신을 연결해야 합니다 (예: 대기 마커).

의미

RELEASE (axis name, axis name, etc.):	축을 분리합니다.
GET (axis name, axis name, etc.):	축을 적용합니다.
GETD (axis name, axis name, etc.):	직접 축을 적용합니다.
Axis name:	시스템에서 축 지정: AX1, AX2, ... 또는 가공 축 이름을 지정합니다.
RELEASE(S1):	스핀들 S1, S2, ...를 분리합니다.
GET(S2):	스핀들 S1, S2, ...를 적용합니다.
GETD(S3):	직접 스핀들 S1, S2, ...를 적용합니다.

전처리 중지 없이 GET 요청

전처리 중지 **없이** GET 요청 이후 축이 RELEASE(axis) 또는 WAITP(axis) 를 통해 다시 활성화된 경우 후속 GET은 전처리 중지를 **통한** GET으로 유도됩니다.

⚠ 주의

축 지정 변경

GET를 통해 적용된 축 또는 스핀들은 키 또는 프로그램 리셋 이후에도 이 채널에 할당된 상태를 유지합니다.

프로그램을 다시 시작할 때 축이 원래 채널에서 필요한 경우 교체된 축 또는 스핀들을 프로그램에서 다시 할당해야 합니다.

전원을 켠 상태에서 머신 데이터에서 정의된 채널에 할당됩니다.

예

예 1: 두 채널 간 축 교체

채널 1에서 가공하는 경우 6개 축 중에서 1번째, 2번째, 3번째 및 4번째 축이 사용됩니다. 채널 2의 5번째 및 6번째 축은 공작물 교체 시 사용됩니다.

축 2는 두 채널 간에 교체되어야 하고 전원을 켠 후 채널 1에 할당될 수 있습니다.

채널 1의 메인 프로그램:

프로그램 코드	설명
INIT (2,"TRANSFER2")	; 채널 2에서 프로그램 TRANSFER2를 선택합니다.
N… START (2)	; 채널 2에서 프로그램을 시작합니다.
N… GET (AX2)	; 축 AX2를 적용합니다.
...	
N… RELEASE (AX2)	; 축 AX2를 분리합니다.
N… WAITM (1,1,2)	; 채널 1 및 2에서 두 채널의 동기화를 위해 WAIT 마커에 도달할 때까지 대기합니다.
...	; 축 교체 이후 프로그램의 나머지 부분을 실행합니다.
N… M30	

채널 2의 TRANSFER2 프로그램:

프로그래밍	설명
N… RELEASE (AX2)	
N160 WAITM(1,1,2)	; 채널 1 및 2에서 두 채널의 동기화를 위해 WAIT 마커에 도달할 때까지 대기합니다.
N150 GET(AX2)	; 축 AX2를 적용합니다.
...	; 축 교체 이후 프로그램의 나머지 부분을 실행합니다.
N… M30	

예 2: 동기화 없이 축 교체

축을 동기화할 필요가 없는 경우 GET에 의해 전처리 중지가 생성되지 않습니다.

프로그래밍	설명
N01 G0 X0	
N02 RELEASE(AX5)	
N03 G64 X10	
N04 X20	
N05 GET(AX5)	; 동기화가 필요 없는 경우 실행할 수 있는 블록이 아닙니다.
N06 G01 F5000	; 실행할 수 없는 블록입니다.
N07 X20	; N04의 X 위치 때문에 실행할 수 없는 블록입니다.
N08 X30	; N05 이후 실행할 수 있는 첫 번째 블록입니다.
...	

예 3: 전처리 정지 없이 축 교체 실행

요구사항: 전처리 중지 없이 축 교체는 머신 데이터를 통해 구성해야 합니다.

프로그래밍	설명
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP(B)	; B 축이 중립 축이 됩니다.
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; 축이 전처리 중지/REORG를 트리거하지 않습니다.
N041 G4 F2	
N050 M5	
N099 M30	

스핀들 또는 축 B가 180°로 이송된 후 블록 N023 바로 다음에 **PLC 축**으로 1°로 이송되면 이 축은 중립 상태로 전환되고 블록 N40에서 전처리 정지가 트리거되지 않습니다.

추가 정보

축 교체를 위한 요구사항

- 축은 가공 데이터의 축을 사용하는 모든 채널에서 정의되어야 합니다.
- **축**별 가공 데이터에서 전원을 켠 후 축이 할당되는 채널을 정의해야 합니다.

설명

축 분리: RELEASE

축을 활성화하는 경우 다음 사항에 유의하십시오.

1. 축을 변환하지 않아야 합니다.
2. 축 링크(접선 제어)에 관련된 모든 축을 활성화해야 합니다.
3. 이 경우 동시 위치 지정 축을 교체할 수 없습니다.
4. 갠트리 마스터 축의 다음 모든 축은 마스터를 통해 이송됩니다.
5. 결합된 축(결합 이동, 마스터 값 결합, 전자식 기어)에서는 그룹의 선행 축만 활성화할 수 있습니다.

축 적용: GET

실제 축 교체는 이 명령으로 수행됩니다. 명령이 프로그래밍되는 채널에서 해당 축의 모든 작업이 수행됩니다.

GET 효과:

동기화를 통한 축 교체:

다른 채널 또는 PLC에 할당되었고 "WAITP", G74를 통해 동기화되지 않은 경우 축을 항상 동기화해야 하거나 GET 이전 이동 거리를 삭제해야 합니다.

- 전처리 중지가 발생합니다(STOPRE의 경우).
- 교체가 완료될 때까지 실행이 중단됩니다.

자동 "GET"

기본적으로 축을 채널에서 사용 가능하지만 현재 "채널 축"으로 정의되지 않은 경우 GET가 자동으로 실행됩니다. 축이 이미 동기화된 경우 전처리 중지가 실행되지 않습니다.

다양한 축 교체 동작 설정

가공 데이터를 사용하여 다음과 같이 축 이송 점을 설정할 수 있습니다.

- 또한 축이 WAITP에 의해 중립 상태가 되면 (이전처럼 응답) 두 채널 간 자동 축 교체가 실행됩니다.
- 축 컨테이너 회전을 요청할 경우 실행 중인 채널에 할당할 수 있는 축 컨테이너의 모든 축이 함축적인 GET 또는 GETD를 사용하여 채널로 이송됩니다. 후속 축 교체는 축 컨테이너 회전을 완료한 경우 다시 한 번만 허용됩니다.
- 중간 블록이 기본 실행 중에 삽입되면 재구성이 필요한지 여부를 확인합니다. 재구성은 이 블록의 축 상태가 현재 축 상태와 일치하지 **않는** 경우에만 필요합니다.
- 전처리 중지 및 전처리와 기본 실행 간 동기화를 통한 GET 블록 대신 전처리 중지 없이 축을 교체할 수 있습니다. 이 경우 중간 블록은 간단히 GET 요청에 의해 생성됩니다. 기본 실행 중에 이 블록을 실행하면 시스템은 블록의 축 상태가 실제 축 상태와 일치하는지 여부를 확인합니다.

4.15.2 다른 채널로 축 이송 (AXTOCHAN)

AXTOCHAN 언어 명령은 축을 다른 채널로 이동하기 위해 축을 요청하는 데 사용할 수 있습니다. 축은 NC 가공 프로그램 및 동기 동작에서 해당 채널로 이동할 수 있습니다.

구문

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

의미

요소	설명
AXTOCHAN:	특정 채널의 축 요청
Axis name:	시스템에서 축 지정: X, Y, ... 또는 관련 가공 축 이름 입력. 실행 중인 채널이 같은 채널 또는 현재 축을 보간 처리 중인 채널일 필요가 없습니다.
Channel number:	축을 할당할 채널 이름

참고

경쟁 포지셔닝 축 및 PLC 제어 축

PLC 축은 채널을 경쟁 포지셔닝 축으로 교체할 수 없습니다. PLC에 의해서만 제어되는 축은 NC 프로그램에 할당할 수 없습니다.

추가 정보: 기능 매뉴얼 축 및 스핀들

예

NC 프로그램에서 AXTOCHAN

X 및 Y 축이 첫 번째 및 두 번째 채널에서 선언되었습니다. 현재 채널 1이 우측 보간 상태이며 다음 프로그램이 해당 채널에서 시작됩니다.

프로그램 코드	설명
N110 AXTOCHAN(Y,2)	;Y축을 두 번째 채널로 이동합니다.
N111 M0	
N120 AXTOCHAN(Y,1)	; Y축을 복귀시킵니다 (중립).
N121 M0	
N130 AXTOCHAN(Y,2,X,2)	;Y축 및 X축을 두 번째 채널로 이동합니다 (축이 중립 상태).
N131 M0	
N140 AXTOCHAN(Y,2)	; Y축을 두 번째 채널로 이동합니다 (NC 프로그램).
N141 M0	

추가 정보

NC 프로그램에서 AXTOCHAN

GET은 동일한 채널에서 NC 프로그램에 대해 축을 요청하는 경우에만 실행됩니다. 따라서 시스템은 상태가 실제로 변경될 때까지 대기합니다. 다른 채널에 대해 축을 요청하거나 축이 동일한 채널에서 중립 축이 되어야 하는 경우 그에 따라 요청이 전달됩니다.

동기 동작의 AXTOCHAN

동일한 채널에 대해 축을 요청하는 경우 동기 동작의 AXTOCHAN은 동기 동작의 GET으로 매핑됩니다. 이 경우 축은 동일한 채널에 대해 첫 번째 요청 시 중립 축이 됩니다. 두 번째 요청 시 축은 NC 프로그램에서 GET 요청과 동일한 방법으로 NC 프로그램에 할당됩니다. 동기 동작의 GET 요청에 대한 자세한 내용은 "동기화 동작"을 참조하십시오.

4.15.3 축 기능(AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

"AXNAME"은 축 이름을 인식하지 못한 경우 일반적으로 적용되는 싸이클을 생성하는 데 사용됩니다.

"AX"는 기하 축 및 동기 축을 간접 프로그래밍하는 데 사용됩니다. 축 이름은 AXIS 유형의 변수에 저장되거나 "AXNAME" 또는 "SPI"와 같은 명령에서 제공됩니다.

"SPI"는 스핀들 (예: 동기 축) 에 축 기능이 프로그래밍된 경우 사용됩니다.

"AXTOSPI"는 축 이름을 스핀들 색인으로 변환 ("SPI"로 역변환) 하는 데 사용됩니다.

"AXSTRING"은 축 이름 (데이터 유형 AXIS) 을 문자열로 변환 ("AXNAME"으로 역변환) 하는 데 사용됩니다.

"ISAXIS"는 특정 기하 축이 반드시 존재하게 하고 에러 메시지와 함께 후속 $P_AXNX 호출이 취소되지 않도록 방지하기 위해 범용 싸이클에 사용됩니다.

"MODAXVAL"은 모듈로 로터리 축의 모듈로 위치를 결정하기 위해 사용됩니다.

구문

```
AXNAME("문자열")
AX[AXNAME("문자열")]
SPI(n)

AXTOSPI(A) 또는 AXTOSPI(B) 또는 AXTOSPI(C)
AXSTRING( SPI(n) )
ISAXIS(<geometry axis number>)
<Modulo position>=MODAXVAL(<axis>,<axis position>)
```

의미

AXNAME:	입력 문자열을 축 식별자로 변환합니다. 입력 문자열에 반드시 유효한 축 이름이 포함돼야 합니다.
AX:	변수 축 식별자
SPI:	스핀들 번호를 축 식별자로 변환합니다. 전송 파라미터에 반드시 유효한 스핀들 번화가 포함돼야 합니다.
n:	스핀들 번호
AXTOSPI:	축 식별자를 정수 스핀들 색인으로 변환합니다. "AXTOSPI"은 "SPI"로 역변환하는 기능입니다.
X, Y, Z:	변수 또는 상수로 표현된 AXIS 형식의 축 식별자
AXSTRING:	관련 스핀들 번호와 함께 문자열이 출력됩니다.

ISAXIS:	지정된 기화 축의 존재 여부를 확인합니다.
MODAXVAL:	모듈로 로터리 축을 대상으로 모듈로 위치를 파악합니다. 이는 파라미터로 표시된 모듈로 범위에 명시되는 모듈로 리셋에 해당합니다(기본 설정은 0 ~ 360도입니다. 모듈로 범위 시작 및 크기는 MD30340 MODULO_RANGE_START와 MD30330 $MA_MODULO_RANGE를 사용해서 변경할 수 있습니다).

참고

SPI 확장

축 함수 SPI(n)은 프레임 요소를 판독하거나 기록하는 용도로도 사용할 수 있습니다. 예를 들어 구문 $P_PFRAME[SPI(1),TR]=2.22로 프레임을 쓸 수 있습니다.

주소 AX[SPI(1)]=<축 위치>를 사용하여 축 위치를 추가로 프로그래밍함으로써 축을 이송할 수 있습니다. 단, 스핀들이 위치 지정 모드나 축 모드여야 한다는 전제 조건이 있습니다.

예

예 1: AXNAME, AX, ISAXIS

프로그램 코드	코멘트
OVRA[AXNAME("이송 축")]=10	; 이송 축 오버라이드
AX[AXNAME("이송 축")]=50.2	; 이송 축의 종점.
OVRA[SPI(1)]=70	; 스핀들 1 오버라이드
AX[SPI(1)]=180	; 스핀들 1의 종점
IF ISAXIS(1) == FALSE GOTOF CONTINUE	; 가로 좌표 사용 가능 여부
AX[$P_AXN1]=100	; 가로 좌표 이동
CONTINUE:	

예 2: AXSTRING

AXSTRING[SPI(n)]을 프로그래밍한 경우 스핀들에 할당된 축의 축 인덱스가 더 이상 스핀들 번호로 출력되지 않습니다. 대신 문자열 "Sn"이 출력됩니다.

프로그램 코드	코멘트
AXSTRING[SPI(2)]	; 문자열 "S2"가 출력됩니다.

예 3: MODAXVAL

모듈로 로터리 축 A의 모듈로 위치를 파악해야 합니다.

축 위치 372.55가 계산에 사용될 시작값입니다.

파라미터로 표현된 모듈로 범위는 0 ~ 360도입니다.

MD30340 MODULO_RANGE_START = 0

MD30330 $MA_MODULO_RANGE = 360

프로그램 코드	코멘트
R10=MODAXVAL(A,372.55)	; 계산된 모듈로 위치 R10 = 12.55.

예 4: MODAXVAL

프로그래밍된 축 식별자가 모듈로 로터리 축을 참조하지 않는 경우, 변환돼야 할 값 (<축 위치>)이 변경되지 않고 반환됩니다.

프로그램 코드	코멘트
R11=MODAXVAL(X,372.55)	; X는 직선 축. R11 = 372.55.

4.15.4 대체 가능한 기하 축(GEOAX)

"전환 가능 기하 축" 기능을 사용하여 머신 데이터를 통해 구성된 기하 축을 다른 채널 축으로 교체할 수 있습니다.

구문

```
GEOAX(<n>,<channel axis>,<n>,<channel axis>,<n>,<channel axis>)
GEOAX()
```

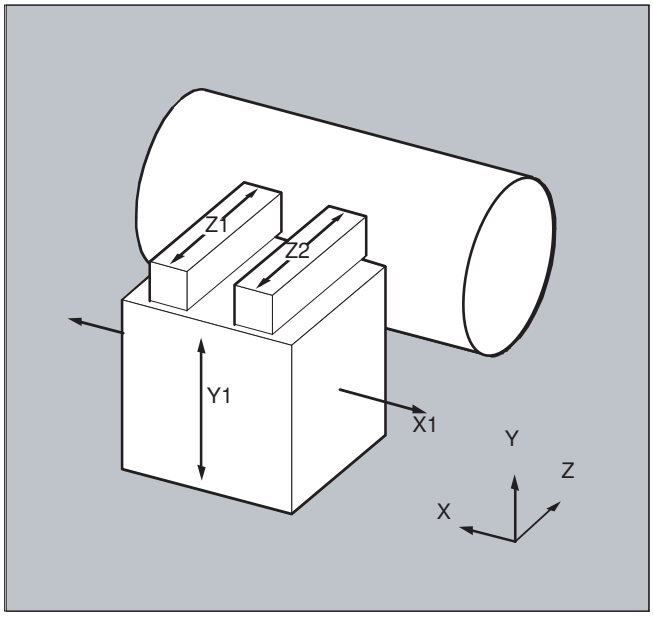
의미

GEOAX(...)	기하 축 전환 명령. **참고:** 파라미터가 지정되지 않은 GEOAX()는 머신 데이터에서 파라미터가 설정된 기하 축들의 기본 구성을 다시 활성화합니다.
<n>	지정된 채널 축으로 교체할 기하 축의 번호. 값 범위: 0, 1, 2, 3 **참고:** 0: 지정된 채널 축이 교체되지 않고 기하 축 그룹에서 제거됩니다. 1: 1번째 기하 축 ≙ WCS의 좌표 축 X (가로 좌표) 2: 2번째 기하 축 ≙ WCS의 좌표 축 Y (세로 좌표) 3: 3번째 기하 축 ≙ WCS의 좌표 축 Z (직교 좌표)
<channel axis>	기하 축 그룹에 추가될 채널 축의 이름

예

예 1: 기하 축으로 대체하도록 2개의 축 전환

채널 축 X1, Y1, Z1, Z2을 이용하여 공구 슬라이드를 이송할 수 있습니다.



전원 가동 후 먼저 Z1이 기하 축의 이름 'Z'에 준하여 3차 기하 축으로 적용되고 X1 및 Y1과 함께 기하 축 그룹을 형성하도록 기하 축들이 구성됩니다.

기하 축 Z1과 Z2이 이제 가공 프로그램에서 기하 축 Z으로 대체되어 사용됩니다.

프로그램 코드	코멘트
...	
N100 GEOAX(3,Z2)	; 채널 축 Z2가 3차 기하 축(Z) 역할을 합니다.
N110 G1 ...	
N120 GEOAX(3,Z1)	; 채널 축 Z1이 3차 기하 축(Z) 역할을 합니다.
...	

예 2: 6개 채널 축을 기하 축으로 대체

기계당 6개의 채널 축이 있습니다. 축의 이름은 XX, YY, ZZ, U, V 및 W입니다.

머신 데이터를 통한 기하 축 구성에 관한 기본 설정은 다음과 같습니다.

채널 축 XX = 1차 기하 축(X 축)

채널 축 YY = 2차 기하 축(X 축)

채널 축 ZZ = 3차 기하 축(Z 축)

프로그램 코드	코멘트
N10 GEOAX()	; 기하 축의 기본 구성이 적용됩니다.
N20 G0 X0 Y0 Z0 U0 V0 W0	; 모든 축이 위치 0으로 신속하게 이송합니다.
N30 GEOAX(1,U,2,V,3,W)	; 채널 축 U가 첫 번째 (X), V가 두 번째 (Y),
	; 그리고 W가 세 번째 기하 축 (Z) 이 됩니다.
N40 GEOAX(1,XX,3,ZZ)	; 채널 축 XX가 첫 번째 (X), ZZ가 세 번째
	; 기하 축 (Z) 이 됩니다. 채널 축 V는 두 번째
	; 기하 축 (Y) 으로 남아 있습니다.
N50 G17 G2 X20 I10 F1000	; X/Y 평면의 완전한 원. 채널 축
	; XX 및 V 이송.
N60 GEOAX(2,W)	; 채널 축 W가 두 번째 기하 축 (Y) 이 됩니다.
N80 G17 G2 X20 I10 F1000	; X/Y 평면의 완전한 원. 채널 축
	; XX 및 W 이송.
N90 GEOAX()	; 초기 상태로 리셋됩니다.
N100 GEOAX(1,U,2,V,3,W)	; 채널 축 U가 첫 번째 (X), V가 두 번째 (Y),
	; 그리고 W가 세 번째 기하 축 (Z) 이 됩니다.
N110 G1 X10 Y10 Z10 XX=25	; 채널 축 U, V, W 각각이
	; 위치 10으로 이송됩니다. 특수 축 XX는 위치 25로 이송됩니다.
N120 GEOAX(0,V)	; V가 기하 축 그룹에서 제거됩니다.
	; U와 W가 첫 번째 (X) 및 세 번째
	; 기하 축 (Z) 으로 계속 유지됩니다.
	; 두 번째 기하 축 (Y) 은 지정되지 않은 상태로 유지됩니다.
N130 GEOAX(1,U,2,V,3,W)	; 채널 축 U는 첫 번째 축 (X) 으로, V는
	; 두 번째 축 (Y) 으로, W는 세 번째 기하 축 (Z) 으로 남아 있습니다.
N140 GEOAX(3,V)	; V가 세 번째 기하 축 (Z) 이 되어 W는
	; 중첩되므로 기하 축 그룹에서
	; 제거됩니다. 두 번째 기하 축 (Y) 는
	; 여전히 지정되지 않은 상태로 유지됩니다.

머신 데이터

축 구성

채널 축에 기하 축, 특수 축 및 기계 축 할당:

- MD10000 $MN_AXCONF_MACHAX_NAME_TAB
- MD20050 $MC_AXCONF_GEOAX_ASIGN_TAB
- MD20060 $MC_AXCONF_GEOAX_NAME_TAB
- MD20070 $MC_AXCONF_MACHAX_USED

- MD20080 $MC_AXCONF_CHANAX_NAME_TAB
- MD35000 $MA_SPIND_ASSIGN_TO_MACHAX

리셋 동작

변경된 기하 축 할당의 리셋 동작:

- MD20110 $MC_RESET_MODE_MASK, bit 12
- MD20118 $MC_GEOAX_CHANGE_RESET

NC 시작 동작

- MD20112 $MC_START_MODE_MASK, bit 12

PLC 사용자 프로그램에 통보

기하 축이 변경될 때 NC/PLC 인터페이스에 출력되는 M 명령의 파라미터 설정 옵션.

- MD22532 $MC_GEOAX_CHANGE_M_CODE

보충 조건

기하 축 전환 없음

- 다음 기능들 중 하나가 활성이면, 기하 축 전환이 불가능합니다.
  - 변환
  - 스플라인 보간
  - 공구 반경 보정
  - 공구 미세 옵셋
- 기하 축과 다른 채널 축의 이름이 동일합니다.
- 기하 축 전환에 관련된 축들 중 하나가 블록 한계를 넘어서는 동작에 관여됩니다 (예: 블록 전반에 걸친 포지셔닝 축 또는 축 커플링의 추종 축).

로터리 축

로터리 축은 기하 축으로 프로그램될 수 없습니다.

대체 후 축 상태

채널 축 이름을 통한 대체 작업 후에 기하 축 그룹에서 대체 기능에 의해 대체된 축을 보충 축으로 프로그래밍할 수 있습니다.

프레임, 보호 영역, 작업 영역 제한

기하 축을 변경하고 나면 모든 프레임, 보호 영역, 그리고 작업 영역 제한이 삭제됩니다.

극 좌표계

GEOAX을 사용하여 기하 축을 대체하면 아날로그가G17-G19를 이용한 레벨 변경으로 설정되고 모달 극 좌표가 0의 값으로 설정됩니다.

DRF, WO

가능한 핸드휠 옵셋 (DRF) 또는 외부 워크 옵셋 (WO) 은 변경 이후에도 그대로 적용됩니다.

기하 축의 기본 구성

GEOAX() 명령은 기하 축 그룹의 기본 구성을 호출합니다.

전원 가동 이후 또는 '원점 복귀' 모드로 전환했을 때 시스템이 기본 구성으로 자동 복구됩니다.

공구 길이 보정

활성 상태의 공구 길이 보정 역시 대체 후에도 적용 상태를 유지합니다. 그러나 새로 추가되거나 위치가 대체된 기하 축의 경우, 아직 이동하지 않은 것으로 인식됩니다. 이 기하 축에 대한 첫 번째 이동 명령의 경우, 산출되는 이송 거리는 공구 길이 보정 총계와 프로그래밍된 이송 거리로 구성됩니다.

대체 작업 이후 축 그룹에서 기존 위치를 유지하는 기하 축은 공구 길이 보정과 관련해서도 원래 상태를 유지합니다.

변환 활성 상태와 관련한 기하 축 구성

- 변환 활성화를 위해 변환 머신 데이터를 통해 파라미터가 지정된 기하 축 구성은 "전환 가능 기하 축" 기능을 사용하여 변경할 수 없습니다.
- 변환을 위해서는 변환 머신 데이터에서 여러 기하 축 구성에 서로 다른 데이터 세트를 파라미터로 설정해야 합니다.
- GEOAX를 사용하여 변경된 기하 축 구성은 변환을 활성화함으로써 삭제할 수 있습니다.
- 기하 축과 관련해서는, 활성 변환의 변환별 기하 축 파라미터 설정이 기하 축의 전환과 관련된 파라미터 설정에 우선합니다.
  예: 특정 변환이 활성화되어 있습니다. 머신 데이터에 따르면, 이 변환은 채널 리셋 시 그대로 유지되어야 합니다. 이와 동시에, 기하 축들의 기본 구성은 채널 리셋 시 복원되어야 합니다. 변환을 위해 지정되었던 기하 축 구성은 유지됩니다.
- 변환이 해제되면, 파라미터가 지정된 기하 축 구성의 기본 설정이 다시 효력을 발휘합니다.

JOG 모드, REF 머신 기능

JOG 모드, REF 머신 기능 (기준점 접근) 으로 전환하면, 머신 데이터에서 파라미터가 지정된 기하 축 구성이 효력을 발휘합니다.

4.15.5 축 컨테이너 (AXCTSWE, AXCTSWED, AXCTSWEC)

"AXCTSWE" 또는 "AXCTSWED" 명령은 지정된 축 컨테이너의 회전을 인에이블합니다.

축 컨테이너 회전 인에이블이 이미 설정된 경우 "AXCTSWEC" 명령에 의해 취소됩니다.

구문

```
AXCTSWE(<ID>)
AXCTSWED(<ID>)
AXCTSWEC(<ID>)
```

의미

AXCTSWE:	축 컨테이너 회전 인에이블 "AXCTSWE"를 실행해도 프로그램 처리는 정지되지 않습니다. 회전은 축 컨테이너에 포함된 모든 채널이 인에이블되는 즉시 실행됩니다.
AXCTSWED:	축 컨테이너에 포함된 다른 축들을 고려하지 않고 축 컨테이너 회전 인에이블 **참고** • 가공 프로그램 또는 동기 동작 스타트업을 간소화하기 위한 명령 변용. • 축 컨테이너에 포함된 다른 축들과 관련된 동작은 다음 머신 데이터로 지정할 수 있습니다. MD12760 $MN_ AXCT_FUNCTION_MASK, bit 0
AXCTSWEC:	축 컨테이너 회전 인에이블 취소 **참고** 축 컨테이너 회전 인에이블은 회전이 시작되지 않은 경우에만 취소할 수 있습니다. $AN_AXCTSWA[<axis container>] == 0 시스템 변수에 대한 설명은 "축 컨테이너 (AXCTSWE, AXCTSWED, AXCTSWEC) (페이지 1002)"를 참조하십시오.

<ID>:	축 컨테이너 또는 컨테이너 축의 식별자:	
	CT<number>:	축 컨테이너의 디폴트 식별자: MD12750 $MN_AXCT_NAME_TAB 예: "CT1"
	<Container>:	사용자별 축 컨테이너 식별자: MD12750 $MN_AXCT_NAME_TAB 예: "CONTAINER_1"
	<axis>:	채널에 입력된 컨테이너 축 식별자

참고

증분

축 컨테이너 회전의 증분은 다음 셋팅 데이터로 설정합니다.

SD41700 $SN_AXCT_SWWIDTH

추가 정보

진단

다음 시스템 변수를 사용하여 축 컨테이너의 현재 상태를 읽을 수 있습니다.

시스템 변수	유형	설명
$AC_AXCTSWA[<name>]	BOOL	축 컨테이너의 채널별 상태
$AN_AXCTSWA[<축 컨테이너>]	BOOL	축 컨테이너의 NCU별 상태
$AN_AXCTSWE[<축 컨테이너>]	INT	축 컨테이너 회전의 슬롯별 상태 시스템 변수는 축 컨테이너 슬롯의 상태를 **비트별로** 제공합니다. 각 비트는 슬롯에 해당합니다.
$AN_AXCTAS[<축 컨테이너>]	INT	축 컨테이너가 연결된 위치 (슬롯) 의 수.

암묵적 GET/GETD를 이용한 축 컨테이너 회전

다음 머신 데이터를 사용하여 채널의 모든 컨테이너 축을 "AXCTSWE" 명령으로 암시적으로 "GET/GETD"하는 방법으로 채널로 가져오도록 설정할 수 있습니다. 축은 컨테이너 회전이 끝난 후에만 교체할 수 있습니다.

MD10722 $MN_AXCHANGE_MASK, bit 1 = 1

참고

축이 "메인 실행 축" 상태 (예: PLC 축) 경우 암시적 "GET/GETD"으로 축 컨테이너 회전을 실행할 수 **없습니다**. 축 컨테이너 회전을 위해서는 축이 메인 실행 축 상태가 아니어야 합니다.

4.15.6 유효 축 위치 대기 (WAITENC)

언어 명령 "WAITENC"를 사용하여 NC 프로그램은 MD34800 $MA_WAIT_ENC_VALID = 1으로 설정된 축에 동기 축 위치 또는 복구된 축 위치를 사용할 수 있을 때까지 대기합니다.

대기 상태에서 ASUB를 시작하거나 조작 모드를 JOG로 바꾸면 대기 상태가 중단됩니다. 프로그램이 계속되면 대기 상태가 재개됩니다.

참고

사용자 인터페이스에는 대기 상태가 보류 상태 "측정 시스템 대기"로 표시됩니다.

구문

"WAITENC"는 모든 NC 프로그램의 프로그램 섹션에 프로그래밍이 가능합니다.

단, 다음과 같이 전용 블록에서 프로그래밍해야 합니다.

```
...
WAITENC
...
```

예

"WAITENC"는 다음 적용 예와 같이 이벤트 제어 사용자 프로그램 (예: .../_N_CMA_DIR/_N_PROG_EVENT_SPF) 에서 사용됩니다.

적용 예제: 오리엔테이션 변환으로 전원 OFF 이후 공구 후퇴

공구 오리엔테이션을 적용하여 가공하는 도중 정전으로 가공이 중단되었습니다.
다시 전원이 들어오면 이벤트 제어 사용자 프로그램 .../_N_CMA_DIR/_N_PROG_EVENT_SPF가 호출됩니다.

이벤트 제어 사용자 프로그램에서 시스템은 "WAITENC"를 사용하여 동기 축 위치 또는 복구된 축 위치를 기다립니다. 이 위치를 알게 되면 프레임을 계산하고 이에 따라 공작물을 공구 방향에 정렬할 수 있습니다.

프로그램 코드	코멘트
...	
IF $P_PROG_EVENT == 4	; 런업.
IF $P_TRAFO <> 0	; 변환이 선택되었습니다.
WAITENC	; 오리엔테이션 축의 유효 축 위치를 확인할 때까지 대기합니다.
TOROTZ	; WCS의 Z축을 공구 축을 향해 회전시킵니다.
ENDIF	
M17	
ENDIF	
...	

JOG 모드에서 공구를 공구 축 방향으로 후퇴 이동시켜 공구를 후퇴시킬 수 있습니다.

4.15.7 프로그램 가능한 파라미터 세트 전환 (SCPARA)

SCPARA 명령을 사용하여 축의 특정 파라미터 세트를 전환하도록 요청할 수 있습니다.

참고

나사 절삭 중에는 파라미터 세트 전환 불가

나사 절삭 G33 및 태핑 G331/G332를 실행 중인 경우 파라미터 세트를 제어 시스템이 선택하기 때문에 변경할 수 없습니다.

파라미터 세트 전환 디스에이블

파라미터 세트 전환을 NC/PLC 인터페이스를 통해 요청할 수도 있습니다. 전환 시 충돌을 방지하기 위해 NC의 파라미터 세트 전환(SCPARA)을 NC/PLC 인터페이스에서 차단할 수 있습니다.

DB31, ... DBX9.3 (NC에서 파라미터 세트 지정 디스에이블)

참고

NC/PLC 인터페이스를 통해 파라미터 세트 전환을 잠근 상태에서 SCPARA가 파라미터 세트 전환을 요청하면 에러 메시지 없이 전환이 거부됩니다.

구문

```
SCPARA[<Axis>]=<Value>
```

의미

SCPARA:	명령: 파라미터 세트 변경	
<axis>:	축 이름 (채널 축)	
	유형:	AXIS
<Value>:	파라미터 세트 번호: 1, 2, 3, ... 최대 파라미터 블록 개수	

예

프로그램 코드	설명
...	
N110 SCPARA[X]= 3	; 선택: 축 X, 3. 파라미터 설정
...	

추가 정보

파라미터 세트 전환 인에이블

축의 파라미터 세트 전환은 다음과 같이 명시적으로 인에이블해야 합니다.

MD35590 $MA_PARAMSET_CHANGE_ENABLE[<axis>]

파라미터 세트 번호 읽기

선택한 파라미터 세트 (지정된 파라미터 세트) 의 번호는 시스템 변수 $AA_SCPAR로 읽을 수 있습니다.

4.15.8 조정 활성화/비활성화(CADAPTON, CADAPTOF)

사전 정의된 CADAPTON() 및 CADAPTOF() 과정을 사용하여 동적 응답 또는 제어 파라미터를 조정하기 위한 사전 정의 조정을 가공 프로그램에서 활성화, 업데이트 및 비활성화할 수 있습니다. 또한 CYCLE782 (페이지 1261)에서도 사용됩니다.

구문

```
...
CADAPTON(<Result>,<Axis>,<InVar>[,<InVal>])
...
CADAPTOF(<Result>,<Axis>,<InVar>)
...
```

의미

CADAPTON():	조정 관계 활성화		
CADAPTOF():	조정 관계 비활성화 **참고:** 활성 조정 관계 무시(MD16501 = 1) CADAPTOF(), 이전에 프로그래밍된 입력 값(<InVal>)이 활성 상태로 유지됩니다.		
<Result>:	결과 변수: 상태 값을 반환합니다(기준 파라미터로 호출)		
	데이터 유형:	INT	
	값:	0	에러 없음
		1	파라미터 설정된 유효한 조정 표가 없음
		2	<Axis> 파라미터가 올바르지 않음
		3	<InVar> 파라미터가 올바르지 않음
		4	예비용
		5	<InVal> 파라미터가 올바르지 않음
<Axis>:	조정 관계 입력 축의 기계 축 이름		
	데이터 유형:	AXIS	
	값 범위:	채널에 정의된 기계 축 이름	
	참고: 이 파라미터를 사용하면 조정을 처리하며, MD16504 $MN_CADAPT_INPUT_AX에서 파라미터 <Axis>에 해당하는 값을 입력합니다. 다른 <InVar> 및 <InVal> 파라미터가 이 축에 할당됩니다.		
<InVar>:	조정 관계의 입력 변수		
	데이터 유형:	INT	
	값:	1	축 관성
		2	축 위치
		3	축 속도

<InVal>:	조정 관계의 입력 값 지능형 부하 조정을 위한 선택 파라미터(<InVar>=1).		
	데이터 유형:	REAL	
	값:	> 0	실제 관성 모멘트
		=0	로드/로딩이 지정되지 않거나 알려지지 않음 이 경우에는 할당된 조정이 대체 값 1.0을 출력 변수로 제공합니다.
		< 0	<Status> = 5(파라미터 <InVal>가 올바르지 않음)를 이용해 호출이 종료됨 활성화된 마지막 입력 값이 활성 상태로 유지됩니다.
	참고: 입력 값이 프로그래밍되지 않으면 제어 시스템이 가동된 후 마지막으로 프로그래밍된 값 또는 기본값(= 0)이 활성화됩니다.		

예

프로그램 코드	**설명**
`DEF INT RESULT`	`; 결과 변수 정의`
`...`	
`CADAPTON(RESULT,MX1,2)`	`; 조정 활성화`
`IF RESULT <> 0`	`; 결과 변수 평가(각 CADAPTON/CADAPTOF 지침 후에 필요)`
`   MSG('CADAPT-RESULT=' << RESULT)`	
`   STOPRE`	
`   SETAL(61000)`	
`ENDIF`	
`...`	
`CADAPTOF(RESULT,MX1,2)`	`; 조정 비활성화`
`...`	`; 결과 변수 평가`
`...`	

추가 정보

링크 축

이 기능은 링크 축에 적용할 수 없습니다(NCU-Link 기능). 이 경우 결과 변수가 반환 값 2를 제공합니다('파라미터 <Axis>가 올바르지 않음').

블록 탐색

- **계산이 없는 블록 탐색**
  프로그램의 시작과 목표 블록 사이에 프로그래밍된 CADAPTON/CADAPTOF 설명은 무시됩니다. 가공 섹션과 관련된 조정이 선택되도록 검색 대상을 선택해야 합니다.
- **형상 또는 블록 종점에서 계산과 함께 블록 검색**
  활성화 및/또는 비활성화 명령과 가능한 프로그래밍된 입력 값이 수집되고, 검색 종료 시 실제 상태가 검색 작업 블록과 함께 출력됩니다.
- **프로그램 테스트(SERUPRO) 모드**
  SERUPRO가 비활성화되면 CADAPTON/CADAPTOF 명령을 사용하여 얻은 조정 상태가 실제 작동으로 전환됩니다.

4.15.9 FIR 저크 필터를 동적 모드로 조정(CALCFIR)

동적 모드를 변경 한 후 FIR 로우 패스 저크 필터가 활성 상태인 채널의 모든 축에 대해 동일한 댐핑 효과와 형상 정밀도를 달성하기 위해 동적 응답 의존적 FIR 필터 설정을 계산하고 NC 음성 명령 CALCFIR로 유효하게 만들 수 있습니다.

적용

CALCFIR은 CUST_832 제조업체 싸이클에서 사용됩니다.

표준 워크플로우

CYCLE832는 자유형 표면 가공을 위해 NC 프로그램 시작 시 프로그래밍됩니다. 프로그램이 실행되면 CYCLE832가 제조업체 싸이클 CUST_832를 호출하고, 장비 제조업체에 의해 CUST_832에 삽입된 NC 언어 명령 CALCFIR이 실행됩니다.

유효성

CALCFIR이 덮어쓴 필터 설정은 다음 CALCFIR 호출이 다시 덮어쓸 때까지 활성 상태로 유지됩니다. G 그룹 59(경로 보간을위한 동적 응답 모드) 명령을 사용하여 실제 동적 응답 모드로의 변경이 프로그래밍된 경우 항상 필요합니다.

제어 시스템 전원 공급/채널 리셋/프로그램 종료 리셋 시 응답

제어 시스템 전원 공급 및 채널/프로그램 종료 리셋 시 FIR 필터 설정은 G 그룹 59의 초기 설정과 일치하도록 활성화됩니다.

요구 사항

- "Top Speed" 옵션이 설정되어 있습니다.

  참고

  "Top Speed" 옵션으로 최적의 결과를 얻으려면 "Top Surface" 옵션도 사용하는 것이 좋습니다.

- 보간 그룹의 축에 대해 다음 조건을 충족해야 합니다.
  - FIR 로우 패스 저크 필터가 활성화되어 있습니다.
  - 동적 응답 종속 FIR 필터 설정이 구성되어 있습니다.
  - CALCFIR에 의한 필터 설정 덮어쓰기가 활성화되어 있습니다.
- 저크 제한 SOFT/SOFTA가 활성화되어 있습니다.

구문

CALCFIR은 개별 블록에서 프로그래밍됩니다. 제조업체 싸이클 CUST_832의 호출은 동적 응답 모드를 프로그래밍한 후에 바로 구현됩니다.

```
...
DYN...
CALCFIR
...
```

의미

DYN...:	동적 응답 모드 선택을 위한 G 그룹 59의 명령	
	DYNNORM:	**일반 동적 응답** 활성화
	DYNPOS:	**포지셔닝 모드, 태핑**을 위한 동적 응답 활성화
	DYNROUGH:	**황삭**을 위한 동적 응답 활성화
	DYNSEMIFIN:	**준정삭**을 위한 동적 응답 활성화
	DYNFINISH:	**정삭**을 위한 동적 응답 활성화
	DYNPREC:	**스무딩 정삭**을 위한 동적 응답 활성화
CALCFIR:	FIR 로우 패스 저크 필터를 동적으로 조정하는 사전 정의된 절차	

참고

CALCFIR은 FIR 로우 패스 저크 필터를 파라미터화하기 위한 축별 머신 데이터에서 뿐만 아니라 CPRECON (페이지 960)에 대해서도 FIR 필터 계산의 결과를 활성화하기 위해 암시적 NEWCONFIG를 수행합니다. 암시적 NEWCONFIG는 추가적인 NEWCONFIG 관련 머신 데이터가 활성화된다는 것을 의미합니다.

한계 조건

동기 동작에서 사용

CALCFIR은 동기 동작에서 프로그래밍할 수 **없습니다**.

4.15.10 가공 프로그램에서 드라이브 파라미터 읽기/쓰기(DRVPRD, DRVPWR)

장비 제조업체는 DRVPRD 및 DRVPWR 음성 명령을 사용해 가공 프로그램 레벨에서 축별로 드라이브 파라미터를 읽고 쓸 수 있습니다.

DRVPRD 및 DRVPWR은 NC 블록에만 있어야 하는 사전 정의된 절차이므로 언제든지 채널당 정확히 하나의 드라이브 파라미터를 **읽거나 쓸 수** 있습니다.

DRVPRD 및 DRVPWR은 장비 제조업체가 생성한 싸이클에서 사용됩니다.

필수 조건

다음과 같은 요구사항이 적용됩니다.

- NC 제어 드라이브의 스타트업이 완료되었습니다.
- CST 디렉토리(Siemens) 또는 CMA 디렉토리(제조업체)의 싸이클에는 항상 필요한 읽기 및 쓰기 권한이 있습니다.

구문

```
DRVPRD(<Result>,<Axis>,<DrvParNo>,<DrvParIdx>,<Value>)
DRVPWR(<Result>,<Axis>,<DrvParNo>,<DrvParIdx>,<Value>)
```

의미

DRVPRD():	드라이브 파라미터 **읽기**를 위한 사전 정의 절차
DRVPWR():	드라이브 파라미터 **쓰기**를 위한 사전 정의 절차

<Result>:	결과 변수: 상태 반환 값		
	데이터 유형:	INT	
	값:	=0	오류 없이 기능 실행
		> 0	오류가 발생해 기능 종료 자세한 정보는 "추가 정보" > "반환 값을 통한 진단"을 참조하십시오.
<Axis>:	기계 축 이름		
	데이터 유형:	AXIS	
	값 범위:	채널에 정의된 기계 축 이름	
<DrvParNo>:	드라이브 파라미터의 번호		
	데이터 유형:	INT	
<DrvParIdx>:	드라이브 파라미터의 인덱스		
	데이터 유형:	INT	
<Value>:	드라이브 파라미터의 값 DRVPRD: 읽기 값 DRVPWR: 쓸 값		
	데이터 유형:	REAL	

예

X축의 읽기 파라미터 p1460[0] "속도 컨트롤러 P 게인 조정 속도":

프로그램 코드	**설명**
`N100 DEF INT _STATUS`	`; 음성 명령의 반환 값을 위한 변수 정의`
`N110 DEF REAL _Result`	`; 읽기 파라미터 값을 위한 변수 정의`
`N120 DRVPRD(_Status,AX1,1460,0,_Result)`	`; 읽기 파라미터 p1460`

추가 정보

반환 값을 통한 진단

음성 명령의 반환 값이 0이 아닌 값을 반환할 경우 드라이브 파라미터에 대한 액세스가 유효하지 않습니다.

반환 값에 대해 두 가지 숫자 범위가 예약되어 있습니다.

음성 명령이 호출될 때 반환 값:

0 액세스 에러 없음
1 허용되지 않는 값
2 축 사용할 수 없음
3 드라이브 사용할 수 없음
4 사용 권한 누락
5 블록 탐색/프로그램 테스트에서 가능하지 않음
6 드라이브 액세스가 가능하지 않음

드라이브 파라미터를 쓰고 읽을 때 반환 값:

1000 허용되지 않는 파라미터 번호
1001 파라미터 값을 변경할 수 없음
1002 하한값 또는 상한값 위반
1003 하위 인덱스가 올바르지 않음
1004 배열 없음, 하위 인덱스 없음
...
1107 컨트롤러가 활성화되었을 때 쓰기 액세스가 허용되지 않음
1110 스타트업 상태에서 쓰기 액세스만 허용됨: Motor (p0010 = 3)
...
1204 쓰기 액세스 없음

음성 명령의 반환 값을 드라이브의 반환 값과 구별하기 위해 값 1000을 추가합니다.

추가 정보: SINUMERIK 진단 매뉴얼, 알람 201042 "프로젝트 다운로드 중 파라미터 에러"

프로그램 시뮬레이션 중 동작

사용 가능한 드라이브 파라미터는 사용되는 드라이브 시뮬레이션의 유형과 범위에 따라 다릅니다. 드라이브 시뮬레이션이 없을 경우 음성 명령이 반환 값 3을 반환합니다.

장비 제조업체는 변수 $P_SIM을 사용해 프로그램 시뮬레이션을 위한 특수한 처리를 제공할 수 있습니다.

다른 기능과의 상호작용

- 병렬 쓰기/읽기 액세스
  HMI 또는 다른 클라이언트로부터의 병렬 읽기/쓰기 액세스는 음성 명령의 실행 시간에 영향을 줄 수 있습니다.
- REPOS
  음성 명령의 실행이 REORG 이벤트에 의해 중단되면 명령이 반복됩니다.
- 블록 탐색/프로그램 테스트
  블록 탐색 및 프로그램 테스트(또한 SERUPRO)에서 드라이브 파라미터를 읽고 쓰기 위한 음성 명령은 지원되지 않습니다. 결과 변수는 액세스 에러 5를 반환합니다.
  가공 프로그램에서 시스템 변수 $P_SEARCH 및 $P_ISTEST를 사용해 블록 탐색 또는 프로그램 테스트가 활성화되었는지 여부를 쿼리하면 DRVPRD와 DRVPWR을 건너뛰고 교체 전략을 실행할 수 있습니다.

제한 사항

프로그래밍 중에 다음 조건을 준수해야 합니다.

- 동기 동작에서 사용
  동기 동작이 비주기적 액세스의 실행을 기다리지 않기 때문에 동기 동작에서 음성 명령을 프로그램할 수 없습니다.
  동기 동작에서 음성 명령을 사용하면 알람 12571이 출력됩니다.
- 읽기 가능/쓰기 가능 드라이브 파라미터
  SERVO DO 또는 HLA DO의 축별 드라이브 파라미터와 NC 축에 할당된 PROFIdrive 표준 드라이브의 파라미터만 읽거나 쓸 수 있습니다.
  다음 DO의 파라미터는 읽거나 쓸 수 없습니다: CU_I, CU_NX, CU3x, TM, HUB, INFEED.

4.16 축 커플링

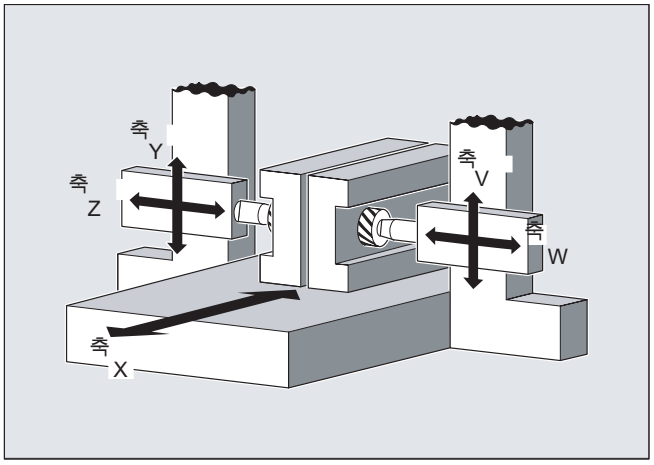
4.16.1 커플 모션 (TRAILON, TRAILOF)

정의 된 리딩 축이 움직일 때, 커플 축 (=추종 축) 은 커플링 팩터가 허용하는 리딩 축에 의해 설정된 거리를 이송하는 데 할당됩니다.

커플 축은 리딩 축과 추종 축으로 구성됩니다.

적용

- 가상의 축을 사용하여 축을 이송합니다. 리딩 축은 가상의 축으로 커플 축이 실제 축입니다. 이런 방법으로, 커플링 팩터의 기능에 의해 실제 축이 이송됩니다.
- 2개의 커플 모션 그룹을 통한 양면 가공:
  1. 리딩 축 Y, 커플 축 V
  2. 리딩 축 Z, 커플 축 W



구문

```
TRAILON(<following axis>,<leading axis>,<coupling factor>)
TRAILOF(<following axis>,<leading axis>,<leading axis 2>)
TRAILOF(<following axis>)
```

의미

TRAILON:	커플 축 그룹핑 활성화 및 정의를 위한 명령	
	적용:	모달
<추종 축>:	파라미터 1: 트레일링 축의 축 이름 **참고:** 커플 축은 다른 커플 축의 리딩 축 역할도 할 수 있습니다. 이런 방법으로 서로 다른 커플 축 그룹화의 범위를 만들 수 있습니다.	
<리딩 축>:	파라미터 2: 트레일링 축의 축 이름	

<커플링 팩터>:	파라미터 3: 커플링 팩터 커플링 팩터는 커플 축과 리딩 축의 경로 사이에 알맞은 관계를 지정합니다. <커플링 팩터> = 커플 축의 경로/리딩 축의 경로	
	유형:	REAL
	디폴트:	1
	마이너스 값을 입력하면 마스터 및 커플 축이 반대 방향으로 이송됩니다. 커플링 팩터가 프로그램되지 않은 경우, 커플링 팩터 1이 자동적으로 적용됩니다.	
TRAILOF:	커플 축 그룹핑 해제 명령	
	적용:	모달

	2개의 파라미터를 사용한 TRAILOF는 지정된 리딩 축에 대한 커플링만 해제합니다.
	`TRAILOF(<following axis>,<leading axis>)`
	커플 축 하나에 리딩 축이 2개인 경우 3개의 파라미터를 사용하여 TRAILOF를 호출하여 두 커플링을 모두 해제할 수 있습니다.
	`TRAILOF(<following axis>,<leading axis>,<leading axis 2>)`
	리딩 축을 지정하지 않고 TRAILOF를 프로그래밍해도 동일한 결과가 나옵니다.
	`TRAILOF(<following axis>)`	

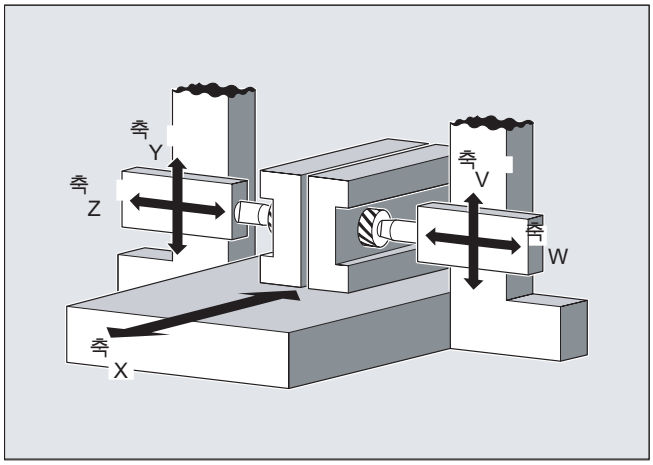
참고

커플 축 동작은 항상 기본 좌표계 (BCS) 에서 실행됩니다.

동시에 활성화될 수 있는 커플 축 그룹핑의 갯수는 기계 상에서 조합될 수 있는 최대 갯수의 조합으로 제한됩니다.

예

다이어그램에 나타나 있는 축 구성을 사용하여 공작물을 양면에서 가공합니다. 이를 위해, 2개의 커플 축 조합을 만듭니다.



프로그램 코드	코멘트
…	
N100 TRAILON(V,Y)	; 첫 번째 커플 축 그룹 활성화.
N110 TRAILON(W,Z,-1)	; 두 번째 커플 축 그룹화 활성화, 음수 커플링 팩터: 커플 축이 리딩 축과 반대 방향으로 이송함
N120 G0 Z10	; 축 반대 방향으로 Z축 및 W축 절입
N130 G0 Y20	; 동일한 축 방향으로 Y축 및 V축 절입.
…	
N200 G1 Y22 V25 F200	; 커플 축 V의 종속적 및 독립적 이동 오버레이.
…	
TRAILOF(V,Y)	; 첫 번째 커플 축 그룹화 비활성화.
TRAILOF(W,Z)	; 두 번째 커플 축 그룹화 비활성화.

추가 정보

축 종류

커플 축 그룹핑은 원하는 리니어 축 및 로터리 축의 조합으로 구성됩니다. 시뮬레이션 축 또한 리딩 축으로 지정할 수 있습니다.

커플 모션 축

동시에 최대 2개의 리딩 축을 트레일링 축으로 지정할 수 있습니다. 서로 다른 커플 축의 조합에 할당할 수 있습니다.

커플 모션 축은 사용 가능한 모든 모션 명령 (G0, G1, G2, G3 등) 을 사용하여 프로그래밍할 수 있습니다. 커플 축은 정의된 경로를 독립적으로 이송하는 것뿐 아니라 커플링 팩터를 기반으로 하여 리딩 축에서 나온 것도 이송합니다.

다이나믹 한계

다이나믹 한계는 커플 축 그룹핑의 활성화 유형에 따라 달라집니다.

- 가공 프로그램에서 활성화
  가공 프로그램에서 활성화가 이루어지고 모든 리딩 축은 활성화된 채널에서 프로그램 축의 역할을 하는 경우 커플 모션 축의 과부하를 방지하기 위해 리딩 축의 이송 과정에서 모든 커플 모션 축의 다이나믹 반응이 고려됩니다.
  가공 프로그램에서 활성화가 이루어지고 리딩 축은 활성화된 채널에서 프로그램 축의 역할을 하지 않는 경우 ($AA_TYP ≠ 1) 리딩 축의 이송 과정에서 커플 모션 축의 다이나믹 반응이 고려되지 않습니다. 이로 인해 커플링에 필요한 정도 이하의 다이나믹 반응에 따라 커플 모션 축에 과부하가 발생할 수 있습니다.
- 동기 동작의 활성화
  동기 동작에서 활성화가 이루어지면 리딩 축의 이송 과정에서 커플 모션 축의 다이나믹 반응이 고려되지 않습니다. 이로 인해 커플링에 필요한 정도 이하의 다이나믹 반응에 따라 커플 모션 축에 과부하가 발생할 수 있습니다.

⚠ 주의

축 과부하

다음과 같은 경우에서 커플 축 그룹핑이 활성화 되면:

- 동기 동작
- 커플 모션 축의 채널에서 프로그램 축 역할을 하지 않는 리딩 축을 가진 가공 프로그램의 경우

리딩 축 이송으로 인해 커플 모션 축에 과부하가 발생하지 않도록 적절한 조치를 취하는 것은 사용자/장비 제조업체의 책임입니다.

커플링 상태

축의 커플링 상태는 가공 프로그램에서 시스템 변수로 확인할 수 있습니다.

$AA_COUP_ACT[<축>]

값	의미
0	커플링 없음
8	커플링 동작

모듈로 로터리 축의 경우 커플 모션 축의 이동할 거리 표시

리딩 축과 커플 모션 축이 모듈로 로터리 축인 경우 커플링이 해제될 때까지 n * 360° (n = 1, 2, 3 ...) 에서 리딩 축의 이송 동작이 커플 모션 축의 이동할 거리 표시에 추가됩니다.

예: TRAILON, 리딩 축 B 및 추종 축 C가 지정된 프로그램 섹션

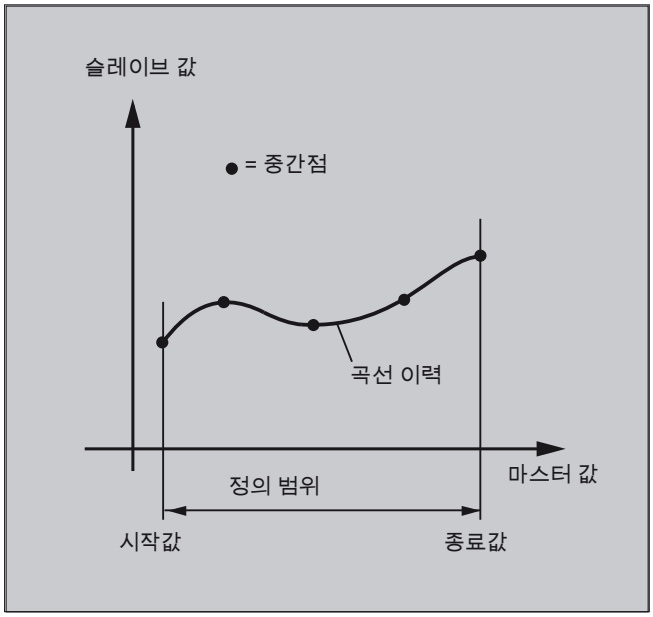
프로그램 코드	코멘트
TRAILON(C,B,1)	; 커플링 활성화
G0 B0	; 시작 위치
	; 블록 시작에서 이동할 거리:
G91 B360	; B=360, C=360
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

4.16.2 커브 테이블 (CTAB)

4.16.2.1 커브 테이블 정의 (CTABDEF, CATBEND)

커브 테이블은 가공 프로그램 또는 CTABDEF로 시작하고 CTABEND로 끝나는 가공 프로그램 섹션을 의미합니다.

이러한 가공 프로그램의 섹션에는 모션 조작을 통해 리딩 축의 개별 위치에 고유의 추종 축이 지정되며, 이 추종 축의 위치는 폴리노미얼 형태의 커브 정의를 최대 5차까지 계산할 때 중간점으로 사용됩니다.



요구사항

커브 테이블 정의에 충분한 메모리 공간을 마련하려면 이에 따라 MD를 설정해야 합니다 (장비 제조업체 사양 참조).

구문

```
CTABDEF(<following axis>,<leading
axis>,<n>,<periodicity>[,<memory location>])
...
CTABEND
```

의미

CTABDEF( ):	커브 테이블 정의 시작	
CTABEND:	커브 테이블 정의 끝	
<추종 축>:	커브 테이블을 사용하여 모션을 계산할 축	
<리딩 축>:	추종 축 모션 계산을 위한 마스터 값을 제공하는 축	
<n>:	커브 테이블 번호 (**ID**) 커브 테이블 번호는 고유한 것이며 메모리 위치와 무관합니다. 그러므로 스태틱 및 다이나믹 NC 메모리에 동일한 번호의 테이블이 존재할 수 없습니다.	
<주기성>:	테이블 주기성	
	0	테이블은 주기적이지 않습니다. (로터리 축의 경우에도 테이블은 단 한 번만 처리됩니다.)
	1	테이블은 리딩 축에 주기적으로 적용됩니다.
	2	테이블은 리딩 축 및 추종 축에 주기적으로 적용됩니다.
<메모리 위치>:	메모리 위치 지정 (옵션)	
	'SRAM'	**스태틱** NC 메모리에 커브 테이블이 생성됩니다.
	'DRAM'	**다이나믹** NC 메모리에 커브 테이블이 생성됩니다.
	참고: 이 파라미터에 값을 프로그래밍하지 않으면 MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE에 설정된 디폴트 메모리 위치가 사용됩니다.	

참고

덮어쓰기

특정 커브 테이블 번호 (<n>) 가 다른 테이블 정의에 사용되면 즉시 해당 커브 테이블을 덮어쓰게 됩니다 (예외: 커브 테이블이 축 커플링에 활성화된 상태이거나 CTABLOCK에 의해 잠긴 경우). **커브 테이블을 덮어쓴 경우에는 경고가 출력되지 않습니다.**

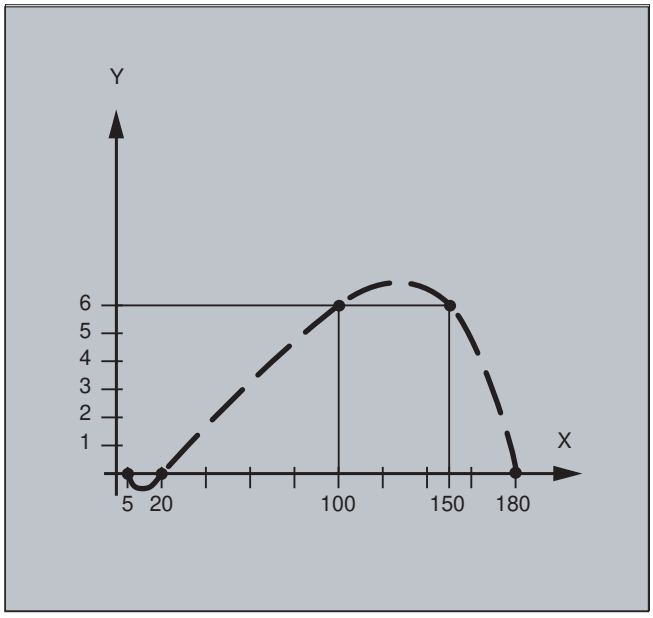
예

예 1: 커브 테이블이 되는 프로그램 섹션 정의

프로그램 섹션은 커브 테이블 정의 시 변경되지 않은 상태로 사용됩니다. 전처리 정지를 위한 STOPRE에 대한 명령은 그대로 유지되며 프로그램 섹션이 테이블 정의에 사용되지 않고 CTABDEF 및 CTABEND가 제거됨과 동시에 다시 활성화됩니다.

프로그램 코드	코멘트
…	
CTABDEF(Y,X,1,1)	; 커브 테이블 정의.
…	
($P_CTABDEF)이 아닌 경우	
STOPRE	
ENDIF	
…	
CTABEND	

예 2: 비 주기적 커브 테이블 정의



프로그램 코드	코멘트
N100 CTABDEF(Y,X,3,0)	; 번호가 3인 비주기적 커브 테이블 정의 시작.
N110 X0 Y0	; 1번째 모션 동작. 시작 값 및 첫 번째 중간점 정의. 마스터 값: 0, 추종 값: 0
N120 X20 Y0	; 2번째 보간점: 마스터 값: 0…20, 추종 값: 시작값…0
N130 X100 Y6	; 3번째 보간점: 마스터 값: 20…100, 추종 값: 0…6
N140 X150 Y6	; 4번째 보간점: 마스터 값: 100…150, 추종 값: 6…6
N150 X180 Y0	; 5번째 보간점: 마스터 값: 150…180, 추종 값: 6…0
N200 CTABEND	; 정의 종료. 커브 테이블은 내부 표현식에서 최대 5차까지의 폴리노미얼로 생성됩니다. 중간점이 지정된 커브 정의 계산은 모달로 선택된 보간 유형 (원, 직선, 스플라인 보간) 에 따라 달라집니다. 정의를 시작하기 전 가공 프로그램 상태가 복원됩니다.

예 3: 주기적 커브 테이블 정의

번호가 2인 주기적 커브 테이블의 정의, 0부터 360까지 마스터 값 범위, 0부터 45 및 다시 0으로 돌아가는 추종 축 이송:

프로그램 코드	코멘트
N10 DEF REAL DEPPOS	
N20 DEF REAL GRADIENT	
N30 CTABDEF(Y,X,2,1)	; 정의 시작.

프로그램 코드	코멘트
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	; 정의 종료.
;Y를 X에 커플링 하여 커브 테스트:	
N120 G1 F1000 X0	
N130 LEADON(Y,X,2)	
N140 X360	
N150 X0	
N160 LEADOF(Y,X)	
N170 DEPPOS=CTAB(75.0,2,GRADIENT)	; 마스터 값 75.0에 대한 테이블 기능 읽기.
N180 G0 X75 Y=DEPPOS	; 리딩 축 및 추종 축 포지셔닝.
;커플링 동작 후, 추종 축을 동기화하지 않아도 됩니다.	
N190 LEADON(Y,X,2)	
N200 G1 X110 F1000	
N210 LEADOF(Y,X)	
N220 M30	

추가 정보

커브 테이블의 시작값 및 종료값

커브 테이블의 정의의 범위 첫 부분에 사용되는 시작값은 커브 테이블 정의에서 지정된 최초의 관련 축 좌표 (최초의 이송 구문) 입니다. 커브 테이블의 정의 범위에 대한 종료값은 최종 이송 명령에 따라 결정됩니다.

사용 가능한 언어 범위

커브 테이블 정의에서, 전체 NC 언어가 사용됩니다.

참고

커브 테이블 정의에서는 다음의 입력이 허용되지 않습니다.

- 전처리 정지
- 리딩 축 동작에서의 점프 (예: 변환 변경 시)
- 추종 축 전용 이송 구문
- 리딩 축 반전 (즉 리딩 축의 위치는 항상 고유해야 합니다.)
- 여러 프로그램 레벨의 CTABDEF 및 CTABEND 구문.

모달 동작의 효과

테이블 정의가 완료되면 커브 테이블 정의에서 작성된 모든 모달 구문이 적용되지 않습니다. 따라서 테이블 정의 가공 프로그램이 만들어지기 이전 및 이후 테이블 정의 상태가 동일합니다.

R 파라미터에 대한 지정

테이블 정의에서 R 파라미터 지정은 CTABEND 다음에 리셋됩니다.

예제:

프로그램 코드	코멘트
...	
R10=5 R11=20	;R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	;R10=25
X=R10	
CTABEND	
...	;R10=5

ASPLINE, BSPLINE, CSPLINE 활성화

커브 정의 테이블 CTABDEF ... CTABEND에서 ASPLINE, BSPLINE 또는 CSPLINE이 활성화된 경우 적어도 이 스플라인 활성화 이전에 시작점을 프로그래밍해야 합니다. CTABDEF 바로 다음에 활성화해서는 안됩니다. 그렇지 않으면 스플라인이 커브 테이블 정의 이전에 현재 축 위치에 따라 달라지게 됩니다.

예:

프로그램 코드
...
CTABDEF(Y,X,1,0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND

커브 테이블 반복 사용

커브 테이블을 사용하여 계산된 리딩 축과 추종 축 사이의 기능적인 관계는 테이블이 스태틱 NC 메모리 (SRAM) 에 저장된 경우 가공 프로그램 종료 및 전원 차단 후 선택된 테이블 번호에 준하여 유지됩니다.

다이나믹 메모리 (DRAM) 에 생성된 테이블은 전원 ON 시 삭제되므로 재생성해야 할 수도 있습니다.

생성된 커브 테이블은 리딩 축 및 추종 축으로 구성된 모든 축 조합에 적용될 수 있으며 커브 테이블을 생성하는 데 사용된 축과는 무관합니다.

커브 테이블 덮어쓰기

특정 커브 테이블 번호가 다른 테이블 정의에 사용되면 즉시 그 커브 테이블을 덮어쓰게 됩니다.

예외: 커브 테이블이 축 커플링에서 활성화되거나 CTABLOCK에 의해 잠긴 경우.

참고

커브 테이블을 덮어쓴 경우에는 경고가 출력되지 않습니다.

커브 테이블 정의 활성 상태의 확인

언제든지 시스템 변수 $P_CTABDEF를 가공 프로그램에서 사용하여 커브 테이블 정의가 활성 상태인지 여부를 확인할 수 있습니다.

커브 테이블 정의 취소

커브 테이블 정의와 관련된 동작이 제외되면 가공 프로그램 섹션을 실제 가공 프로그램으로 다시 사용할 수 있습니다.

'외부 소스에서 실행'을 사용하여 커브 테이블 로딩

외부 소스로부터 커브 테이블을 실행하는 경우 전체 커브 테이블 정의를 리로드 버퍼에 동시에 저장하는 것을 지원하는 크기의 리로드 버퍼 (DRAM) 를
MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE에서 선택해야 합니다. 그렇지 않은 경우에는 알람이 발생되어 가공 프로그램 처리가 취소됩니다.

추종 축에서 점프

리딩 축에 이동이 누락된 경우 머신 데이터
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION
의 설정에 따라 추종 축에서 점프가 허용될 수도 있습니다.

4.16.2.2 커브 테이블의 존재 검사 (CTABEXISTS)

CTABEXISTS 명령은 특정 커브 테이블 번호가 NC 메모리에 있는지 여부를 검사하는 데 사용할 수 있습니다.

구문

```
CTABEXISTS(<n>)
```

의미

CTABEXISTS:	스태틱 또는 다이나믹 NC 메모리에 커브 테이블 번호 <n>이 있는지 검사합니다.	
	0	테이블이 없음
	1	테이블이 있음
<n>:	커브 테이블 번호 (**ID**)	

4.16.2.3 커브 테이블 삭제 (CTABDEL)

CTABDEL을 사용하여 커브 테이블을 삭제할 수 있습니다.

참고

축 커플링에 사용 중인 커브 테이블은 삭제할 수 없습니다.

구문

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<memory location>)
CTABDEL()
CTABDEL(,,<memory location>)
```

의미

CTABDEL:	커브 테이블 삭제 명령
<n>:	삭제할 커브 테이블 번호 (**ID**) 커브 테이블 범위 CTABDEL(<n>,<m>) 을 삭제할 때 <n>은 그 범위에서 첫 번째 커브 테이블 번호를 지정하는 데 사용됩니다.
<m>:	커브 테이블 범위 CTABDEL(<n>,<m>) 를 삭제할 때 <m>은 그 범위에서 마지막 커브 테이블 번호를 지정하는 데 사용됩니다. <m>은 <n>보다 커야 합니다.

<메모리 위치>:	메모리 위치 지정 (옵션) 메모리 위치를 **지정하지 않고** 삭제하는 경우에는 스태틱 및 다이나믹 NC 메모리에서 지정된 커브 테이블이 삭제됩니다. 메모리 위치를 **지정하고** 삭제하는 경우에는 지정된 커브 테이블 중에서 지정된 메모리 위치에 있는 테이블만 삭제됩니다. 나머지는 그대로 유지됩니다.	
	'SRAM'	**스태틱** NC 메모리에서 삭제
	'DRAM'	**다이나믹** NC 메모리에서 삭제

삭제할 커브 테이블을 지정하지 않고 CTABDEL을 프로그래밍하면 **전체** 커브 테이블 또는 지정된 메모리의 모든 커브 테이블이 삭제됩니다.

CTABDEL():	스태틱 및 다이나믹 NC 메모리의 모든 커브 테이블 삭제
CTABDEL(,,"SRAM"):	스태틱 NC 메모리의 모든 커브 테이블 삭제
CTABDEL(,,"DRAM"):	다이나믹 NC 메모리의 모든 커브 테이블 삭제

참고

CTABDEL(<n>,<m>) 또는 CTABDEL() 을 사용하여 여러 개를 삭제하는 경우 삭제할 커브 테이블 중 적어도 하나가 커플링에서 활성 상태이면 삭제 명령이 실행되지 않습니다. 즉, 지정된 커브 테이블 중 그 어느 것도 삭제되지 **않습니다**.

4.16.2.4 삭제 및 덮어쓰기 방지를 위한 커브 테이블 잠금 (CTABLOCK, CTABUNLOCK)

의도하지 않은 커브 테이블 삭제 및 덮어쓰기를 방지하기 위해 잠금을 설정할 수 있습니다. 잠금을 설정한 후에는 언제든지 취소할 수 있습니다.

구문

잠금:

```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<memory location>)
CTABLOCK()
CTABLOCK(,,<memory location>)
```

잠금 해제:

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<memory location>)
CTABUNLOCK()
CTABUNLOCK(,,<memory location>)
```

의미

CTABLOCK:	삭제/덮어쓰기 방지를 위한 잠금 **설정** 명령
CTABUNLOCK:	삭제/덮어쓰기 방지를 위한 잠금 **취소** 명령 CTABUNLOCK은 CTABLOCK을 통해 잠긴 커브 테이블을 해제합니다. 활성 커플링과 관련된 커브 테이블은 잠금 상태가 유지되며 삭제가 불가능합니다. 활성 커플링으로 인한 잠금이 커플링 비활성화로 인해 해제되면 CTABLOCK를 통한 잠금은 즉시 해제됩니다. 그러므로 이 테이블을 삭제할 수 있습니다. CTABUNLOCK를 다시 호출할 필요는 없습니다.
<n>:	잠금/해제할 커브 테이블 번호 (**ID**) 커브 테이블 범위 CTABLOCK(<n>,<m>)/ CTABUNLOCK(<n>,<m>) 를 잠금/해제할 때 <n>은 그 범위에서 첫 번째 커브 테이블 번호를 지정하는 데 사용됩니다.
<m>:	커브 테이블 범위 CTABLOCK(<n>,<m>)/ CTABUNLOCK(<n>,<m>) 를 잠금/해제할 때 <m> 은 그 범위에서 마지막 커브 테이블 번호를 지정하는 데 사용됩니다. <m>은 <n>보다 커야 합니다.
<메모리 위치>:	메모리 위치 지정 (옵션) 메모리 위치를 **지정하지 않고** 잠금/해제하는 경우에는 스태틱 및 다이나믹 NC 메모리에서 지정된 커브 테이블이 잠금/해제됩니다. 메모리 위치를 **지정하고** 잠금/해제하는 경우에는 지정된 커브 테이블 중에서 지정된 메모리 위치에 있는 테이블만 잠금/해제됩니다. 나머지는 잠금/해제되지 않습니다.
	'SRAM' — **스태틱** NC 메모리에서 잠금/해제
	'DRAM' — **다이나믹** NC 메모리에서 잠금/해제

잠금/해제할 커브 테이블을 지정하지 않고 CTABLOCK/CTABUNLOCK를 프로그래밍하면 **모든** 커브 테이블 또는 지정된 메모리의 모든 커브 테이블이 잠금/해제됩니다.

CTABLOCK():	스태틱 및 다이나믹 NC 메모리의 모든 커브 테이블 잠금
CTABLOCK(,,"SRAM"):	스태틱 NC 메모리의 모든 커브 테이블 잠금
CTABLOCK(,,"DRAM"):	다이나믹 NC 메모리의 모든 커브 테이블 잠금
CTABUNLOCK():	스태틱 및 다이나믹 NC 메모리의 모든 커브 테이블 잠금 해제

CTABUNLOCK(,,"SRAM"):	스태틱 NC 메모리의 모든 커브 테이블 잠금 해제
CTABUNLOCK(,,"DRAM"):	다이나믹 NC 메모리의 모든 커브 테이블 잠금 해제

4.16.2.5 커브 테이블: 테이블 속성 결정 (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

이 명령은 커브 테이블의 중요한 속성 (테이블 번호, 잠금 상태, 메모리 위치, 주기성) 을 조사하는 데 사용할 수 있습니다.

구문

```
CTABID(<p>)
CTABID(<p>,<memory location>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

의미

CTABID:	지정된 메모리에서 <p> 번째 커브 테이블로 입력한 **테이블 번호**를 보고합니다. 예: CTABID(1,"SRAM") 는 스태틱 NC 메모리의 첫 번째 커브 테이블 번호를 보고합니다. 이 상황에서 첫 번째 커브 테이블이란 테이블 번호가 가장 높은 커브 테이블을 의미합니다. **참고:** 예를 들어 CTABDEL로 커브 테이블을 삭제하여 CTABID의 연속적 호출 사이에서 메모리의 커브 테이블 시퀀스가 달라지면 CTABID(<p>,...)가 동일한 번호 <p>를 가진 다른 커브 테이블을 반환할 수도 있습니다.	
CTABISLOCK:	커브 테이블 번호 <n>의 **잠금 상태**를 보고합니다.	
	0	테이블이 잠기지 않음
	1	테이블이 CTABLOCK에 의해 잠겨 있음
	2	테이블이 활성 커플링에 의해 잠겨 있음
	3	테이블이 CTABLOCK 및 활성 커플링에 의해 잠겨 있음
	-1	테이블이 없음

CTABMEMTYP:	커브 테이블 번호 <n>의 **메모리 위치**를 보고합니다.	
	0	스태틱 NC 메모리의 테이블
	1	다이나믹 NC 메모리의 테이블
	-1	테이블이 없음
CTABPERIOD:	커브 테이블 번호 <n>의 **주기성**을 보고합니다.	
	0	커브 테이블이 주기적이지 않음
	1	테이블이 리딩 축에서 주기적임
	2	테이블이 리딩 축 및 추종 축에서 주기적임
	-1	테이블이 없음
<p>:	메모리에서의 입력 번호	
<n>:	커브 테이블 번호 (**ID**)	
<메모리 위치>:	메모리 위치 지정 (옵션)	
	'SRAM'	**스태틱** NC 메모리
	'DRAM'	**다이나믹** NC 메모리
	참고: 이 파라미터에 값을 프로그래밍하지 않으면 MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE에 설정된 디폴트 메모리 위치가 사용됩니다.	

4.16.2.6 커브 테이블 값 읽기 (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

다음의 커브 테이블을 가공 프로그램에서 읽을 수 있습니다.

- 커브 테이블의 시작과 끝에 있는 추종 축 및 리딩 축 값
- 커브 분할의 시작과 끝에 있는 추종 축 값
- 리딩 축 값에 대한 추종 축 값
- 추종 축 값에 대한 리딩 축 값
- 추종 축의 최소값 및 최대값
  - 커브 테이블의 전체 정의 범위 이내
    또는
  - 정의된 커브 테이블 간격 이내

구문

```
CTABTSV(<n>,<gradient>[,<following axis>])
```

```
CTABTEV(<n>,<gradient>[,<following axis>])
CTABTSP(<n>,<gradient>[,<leading axis>])
CTABTEP(<n>,<gradient>[,<leading axis>])
CTABSSV(<master value>,<n>,<gradient>[,<following axis>])
CTABSEV(<master value>,<n>,<gradient>[,<following axis>])
CTAB(<master value>,<n>,<gradient>[,<following axis>,<leading
axis>]
CTABINV(<following value>,<approximate
value>,<n>,<gradient>[,<following axis>,<leading axis>]
CTABTMIN(<n>[,<following axis>])
CTABTMAX(<n>[,<following axis>])
CTABTMIN(<n>,<a>,<b>[,<following axis>,<leading axis>])
CTABTMAX(<n>,<a>,<b>[,<following axis>,<leading axis>])
```

의미

CTABTSV:	커브 테이블 번호 <n>의 **시작**에서 추종 축 값 읽기
CTABTEV:	커브 테이블 번호 <n>의 **끝**에서 추종 축 값 읽기
CTABTSP:	커브 테이블 번호 <n> 의 **시작**에서 리딩 축 값 읽기
CTABTEP:	커브 테이블 번호 <n>의 **끝**에서 리딩 축 값 읽기
CTABSSV:	지정된 리딩 축 값 (<master value>) 에 속하는 커브 분할의 **시작**에서 추종 축 값 읽기
CTABSEV:	지정된 리딩 축 값 (<master value>) 에 속하는 커브 분할의 **끝**에서 추종 축 값 읽기
CTAB:	지정된 리딩 축 값 (<master value>) 에 대한 추종 축 값 읽기
CTABINV:	지정된 추종 축 값(<following value>) 에 대한 리딩 축 값 읽기
CTABTMIN:	추종 축 **최소값** 정의: • 커브 테이블의 전체 정의 범위 이내 또는 • 정의된 간격: <a> ... <b>
CTABTMAX:	추종 축 **최대값** 정의: • 커브 테이블의 전체 정의 범위 이내 또는 • 정의된 간격: <a> ... <b>
<n>:	커브 테이블 번호 (**ID**)
<gradient>:	<gradient> 파라미터는 계산된 위치에서 커브 테이블 기능의 **경사도**를 반환합니다.

<following axis>:	커브 테이블을 사용하여 모션을 계산할 축 (옵션)
<leasing axis>:	추종 축 모션 계산을 위한 마스터 값을 제공하는 축 (옵션)
<following value>:	CTABINV에 대한 리딩 축의 관련 값을 읽는 추종 축의 값
<leading value>:	리딩 축 값: • 연관된 추종 축 값을 CTAB으로 읽는 경우 또는 • CTABSSV/CTABSEV로 커브 분할을 선택하는 경우
<approximate value>:	CTABINV를 사용하여 추종 축 값에 지정하는 리딩 축 값이 반드시 고유한 값일 필요는 없습니다. 따라서 CTABINV은 예상되는 리딩 축 값에 대한 근사치를 파라미터로 요구합니다.
<a>:	CTABTMIN/CTABTMAX를 통한 마스터 값 간격의 **하한값**
<b>:	CTABTMIN/CTABTMAX를 통한 마스터 값 간격의 **상한값**
	참고: <a> 에서 <b> 사이의 마스터 값 간격은 반드시 커브 테이블의 정의 범위에 속해야 합니다.

예

예 1:

커브 테이블의 전체 정의 범위에 포함되는 추종 축의 최소값 및 최대값과 함께 커브 테이블의 시작과 끝에서 추종 축 및 리딩 축의 값을 정의합니다.

프로그램 코드	코멘트
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)	, 테이블 정의 시작
N110 X0 Y10	, 첫 번째 테이블 분할 시작 위치
N120 X30 Y40	; 1번째 테이블 분할 종료 위치 = 2번째 테이블 분할 시작 위치
N130 X60 Y5	, 두 번째 테이블 분할 종료 위치 = ...
N140 X70 Y30	

프로그램 코드	코멘트
N150 X80 Y20	
N160 CTABEND	, 테이블 정의 종료
...	
N200 STARTPOS=CTABTSV(1,GRADIENT)	, 커브 테이블 시작 부분의 추종축 값 = 10
N210 ENDPOS=CTABTEV(1,GRADIENT)	; 커브 테이블 종료 부분의 추종축 값 = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)	, 커브 테이블 시작 부분의 마스터축 값 = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)	, 커브 테이블 종료 부분의 마스터축 값 = 80
N240 MINVAL=CTABTMIN(1)	, Y=5인 추종축 최소값
N250 MAXVAL=CTABTMAX(1)	, Y=40인 추종축 최대값

예 2:

리딩 축 값 X=30과 관련된 커브 분할의 시작 및 종료 부분에서 추종 축 값 결정

프로그램 코드	코멘트
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	, 테이블 정의 시작.
N110 X0 Y0	, 첫 번째 테이블 분할 시작 위치
N120 X20 Y10	; 1번째 테이블 분할 종료 위치 = 2번째 테이블 분할 시작 위치
N130 X40 Y40	, 두 번째 테이블 분할 종료 위치 = ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	, 테이블 정의 종료
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	, 두 번째 분할 시작 위치 = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	, 두 번째 분할 종료 위치 = 40

추가 정보

동기 동작에서 사용

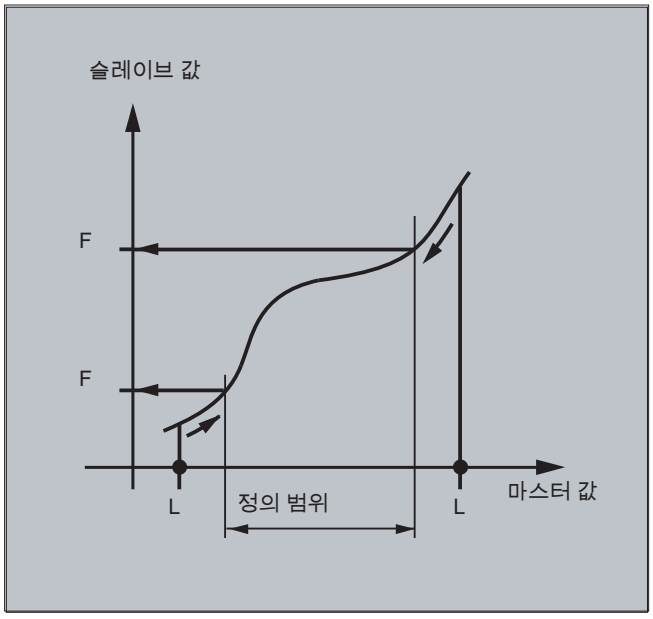
커브 테이블 값 읽기를 위한 모든 명령을 동기 동작에서도 사용할 수 있습니다 ('모션 동기 동작' 장 참조).

CTABINV, CTABTMIN 및 CTABTMAX 명령을 사용할 때는 다음 사항을 명확하게 해야 합니다.

- 실행 시 사용할 NC 전력이 충분한지
  또는
- 필요하다면 관련 테이블을 분할할 수 있도록 커브 테이블의 분할 수가 호출 전에 조회되는지

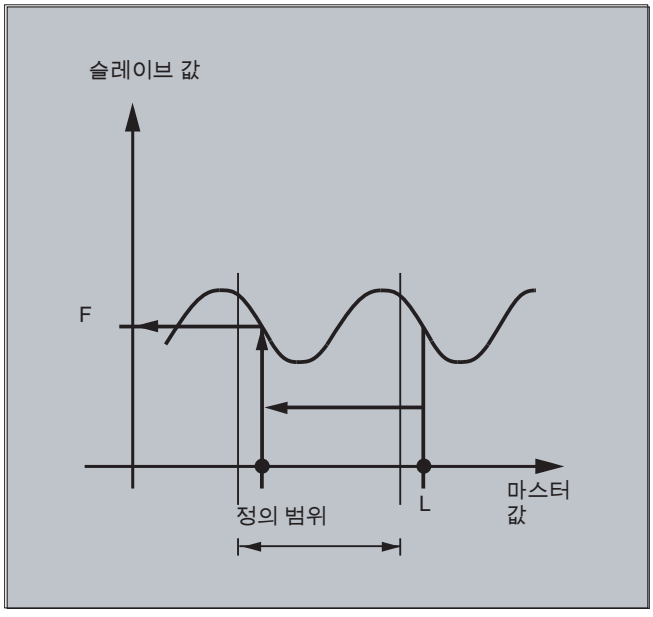
비 주기적 커브 테이블을 포함한 CTAB

지정된 <master value>가 정의 범위를 벗어나는 경우 상한값 또는 하한값은 추종 값으로 출력됩니다.



주기적 커브 테이블을 포함한 CTAB

지정된 <master value>가 정의 범위를 벗어나는 경우 정의 범위의 모듈로를 기준으로 마스터 값이 산정되고 그에 상응하는 추종 값이 출력됩니다.



CTABINV에 대한 근사값

따라서 CTABINV 명령은 예상 마스터 값의 근사값을 필요로 합니다. CTABINV는 근사치에 가장 가까운 리딩 값을 반환합니다. 예를 들어 이전의 보간 클록 사이클에서 도출한 마스터 값이 근사치로 결정될 수 있습니다.

커브 테이블 기능의 경사

경사도(<gradient>) 출력을 통해 해당 위치에서 리딩 축 또는 추종 축의 속도를 계산할 수 있습니다.

리딩 축 또는 추종 축 지정

리딩 축과 추종 축이 다른 길이 단위로 설정된 경우 리딩 축 및/또는 추종 축의 옵션 지정이 중요합니다.

CTABSSV, CTABSEV

다음의 경우 CTABSSV 및 CTABSEV 명령은 프로그래밍된 분할 조회에는 적합하지 **않습니다**.

- 원호 또는 나선형이 프로그래밍된 경우
- CHF/ RND를 이용한 모따기나 라운딩이 활성 상태인 경우
- G643을 이용한 스무딩이 활성 상태인 경우
- NC 블록 압축(COMP...)이 활성 상태인 경우

4.16.2.7 커브 테이블: 리소스 사용 확인 (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

프로그래머는 이 명령을 사용하여 커브 테이블, 테이블 분할 및 폴리노미얼을 위한 리소스 사용에 대한 최신 정보를 구할 수 있습니다.

구문

```
CTABNO
CTABNOMEM(<memory location>)
CTABFNO(<memory location>)
CTABSEGID(<n>,<memory location>)
CTABSEG(<memory location>,<segment type>)
CTABFSEG(<memory location>,<segment type>)
CTABMSEG(<memory location>,<segment type>)
CTABPOLID(<n>)
CTABPOL(<memory location>)
CTABFPOL(<memory location>)
CTABMPOL(<memory location>)
```

의미

CTABNO:	스태틱 및 다이나믹 NC 메모리에 **정의된** 커브 테이블의 총 개수 확인	
CTABNOMEM:	지정된 <memory location>에서 **정의된** 커브 테이블의 개수 확인	
CTABFNO:	지정된 <memory location>에서 **남아 있을 수 있는** 커브 테이블의 개수 확인	
CTABSEGID:	커브 테이블 번호 <n>에서 사용하는 지정된 <segment type>의 커브 분할 개수 확인	
CTABSEG:	지정된 <memory location>에서 **사용되는** 지정된 <segment type>의 커브 분할 개수 확인	
CTABFSEG:	지정된 <memory location>에 **남아 있을 수 있는** 지정된 <segment type>의 커브 분할 개수 확인	
CTABMSEG:	지정된 <memory location>에서 지정된 <segment type>의 **가능한 최대** 커브 분할 개수 확인	
CTABPOLID:	커브 테이블 번호 <n>에서 사용하는 커브 폴리노미얼의 개수 확인	
CTABPOL:	지정된 <memory location>에서 **사용된** 커브 폴리노미얼의 개수 확인	
CTABFPOL:	지정된 <memory location>에서 **남아 있을 수 있는** 커브 폴리노미얼의 개수 확인	
CTABMPOL:	지정된 <memory location>에 남아 있을 수 있는 **최대** 커브 폴리노미얼의 개수 확인	
<n>:	커브 테이블 번호 (**ID**)	
<메모리 위치>:	메모리 위치 지정 (옵션)	
	"SRAM"	**스태틱** NC 메모리
	"DRAM"	**다이나믹** NC 메모리
	참고: 이 파라미터에 값을 프로그래밍하지 않으면 MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE에 설정된 디폴트 메모리 위치가 사용됩니다.	

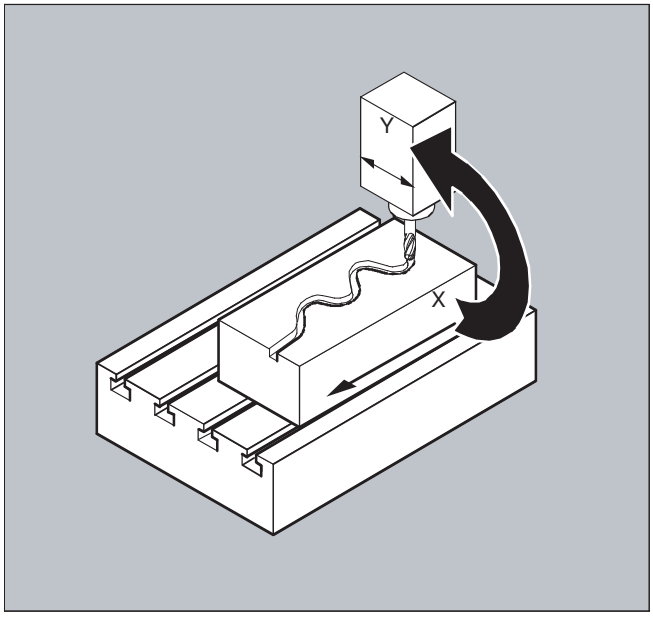
<분할 유형>:	분할 유형 지정 (옵션)	
	"L"	직선 분할
	"P"	폴리노미얼 분할
	참고: 이 파라미터에 대한 값을 프로그래밍하지 않으면 직선 분할 및 폴리노미얼 분할의 합계가 출력됩니다.	

4.16.3 축 마스터 값 커플링 (LEADON, LEADOF)

참고

이 기능은 SINUMERIK 828D에 사용할 수 없습니다!

축 관련 커플링을 사용한 경우, 커플링을 사용한 후, 리딩 축과 추종 축이 동기되어 이동합니다. 필요한 경우, 커브 테이블이나 그로 인해 얻은 폴리노미얼을 사용한 시뮬레이션으로 리딩 축의 위치에 추종 축의 위치를 할당할 수 있습니다.



리딩 축은 커브 테이블에 입력값을 제공하는 축입니다. **추종 축**은 커브 테이블에 의해 계산된 위치를 차지하는 축입니다.

실제값과 지령치 커플링

아래 항목을 마스터 값 (추종 축의 위치 계산을 위한 출력 값) 으로 사용할 수 있습니다.

- 리딩 축 위치의 실제값: 실제값 커플링
- 리딩 축 위치의 지령치: 지령치 커플링

기본 좌표계에는 항상 마스터 값 커플링이 적용됩니다.

커브 테이블 생성에 대한 자세한 정보는 "커브 테이블" 절을 참조하십시오.

구문

```
LEADON(<following axis>,<leading axis>,<n>)
LEADOF(<following axis>,<leading axis>)
```

또는 리딩 축을 지정하지 않은 채 비활성화:

```
LEADOF(<following axis>)
```

마스터 값 커플링은 가공 프로그램이나 동기 동작에서 모션 중일 때 활성화 또는 비활성화할 수 있습니다.

의미

LEADON:	마스터 값 커플링 활성화
LEADOF:	마스터 값 커플링 해제
<following axis>:	추종 축
<leading axis>:	리딩 축
<n>:	커브 테이블 번호
$SA_LEAD_TYPE:	지령치 커플링 및 실제치 커플링 간의 전환

마스터 값 커플링 해제, LEADOF

마스터 값 커플링을 해제하면 추종 축이 다시 일반 지령 축으로 전환됩니다!

축 관련 마스터 값 커플링 및 다양한 동작 상태, RESET

머신 데이터의 설정 상태에 따라 RESET을 누르면 마스터 값 커플링을 해제하기도 합니다.

동기 동작에서 마스터 값 커플링 예제

프레스 공장에서, 이송 축 및 보조 축으로 구성된 이송 시스템 축과 리딩 축 (Stanchion Shaft) 간의 일반적인 기계적 커플링을 전기적 커플링으로 대체할 수 있습니다.

기계적 이송 시스템을 전기적 이송 시스템으로 대체하는 방법은 아래와 같습니다. 커플링 및 커플링 해제 과정은 **스태틱 동기 동작**으로 구현됩니다.

리딩 축 LV(Stanchion Shaft) 에서 이송 축과 보조 축은 커브 테이블을 통해 정의된 추종 축처럼 제어됩니다.

추종 축

X 피드 또는 세로 축
YL 폐쇄 또는 가로 축
ZL 도피 축
U 롤러 이송, 보조 축
V 가이드 헤드, 보조 축
W 윤활유 공급, 보조 축

동작

발생하는 동작에는 다음과 같은 동기 동작이 포함됩니다.

- 커플링 활성화, LEADON(<following axis>,<leading axis>,<curve table number>)
- 커플링 비활성화, LEADOF(<following axis>,<leading axis>)
- 실제값 설정, PRESETON(<axis>,<value>)
- 마커 설정, $AC_MARKER[i]=<value>
- 커플링 형식: 실제/가상 마스터 값
- 축 위치 접근, POS[<axis>]=<value>

조건

부울 (Boolean) 연산자 'AND'를 사용하여 고속 디지털 입력, 실시간 변수 $AC_MARKER, 그리고 위치 비교를 조건으로 연결합니다.

참고

아래 예제에서는 단지 프로그램의 가독성 향상을 위해 라인 변경, 들여쓰기 및 **고딕체**가 사용됐습니다. 제어 시스템에서는 특정 라인 번호 이후에 나오는 모든 내용들을 하나의 라인으로 처리합니다.

코멘트

프로그램 코드	코멘트
	; 모든 스태틱 동기 동작 정의.
	; ****리셋 마커
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
	; **** E1 0=>1 전송 ON

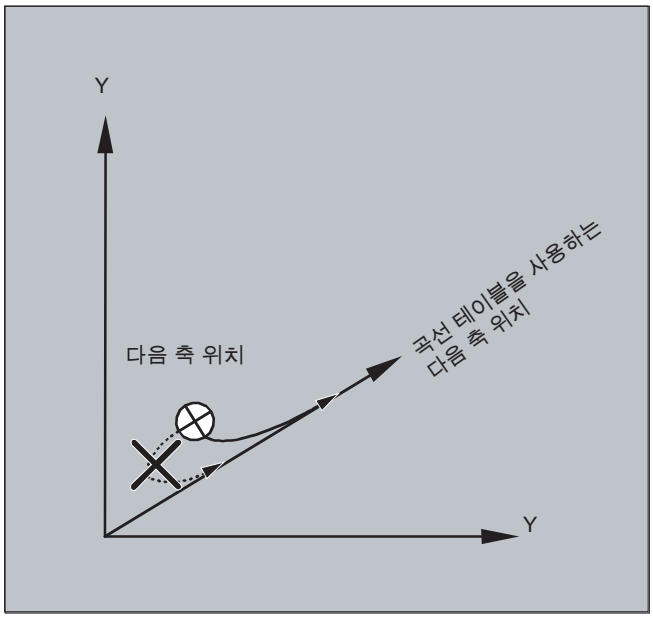
프로그램 코드	코멘트
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_MARKER[0]==0)	
DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) $AC_MARKER[0]=1	
	; **** E1 0=>1 롤러 이송 커플링 ON
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0)	
DO LEADON(U,LW,4) PRESETON(U,0) $AC_MARKER[5]=1	
	;**** E1 0->1 정렬 헤드 커플링 ON
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0)	
DO LEADON(V,LW,4) PRESETON(V,0) $AC_MARKER[6]=1	
	; **** E1 0->1 윤활 커플링 ON
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[7]==0)	
DO LEADON(W,LW,4) PRESETON(W,0) $AC_MARKER[7]=1	
	; **** E2 0=>1 커플링 OFF
N30 IDS=3 EVERY ($A_IN[2]==1)	
DO LEADOF(X,LW) LEADOF(YL,LW) LEADOF(ZL,LW) LEADOF(U,LW) LEADOF(V,LW) LEADOF(W,LW) $AC_MARKER[0]=0	
$AC_MARKER[1]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
....	
N110 G04 F01	
N120 M30	

설명

마스터 값 커플링에는 리딩 축과 추종 축의 동기화가 수반돼야 합니다. 마스터 값 커플링이 활성 상태일 때 추종 축이 커브 테이블에서 산정한 곡선 정의의 공차 범위를 벗어나지 않은 경우에 한하여 동기화가 가능합니다.

추종 축의 위치에 대한 공차 범위는 머신 데이터 MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE을 통해 정의합니다..

마스터 값 커플링이 활성화된 상태에서 추종 축이 아직 올바른 위치에 있지 않은 경우, 추종 축에 대해 산정된 위치 지령치 값이 실제 추종 축의 위치에 근접한 즉시, 동기화 실행이 자동으로 초기화됩니다. 동기화 절차가 진행되는 동안, 추종 축은 추종 축의 지령치 속도(마스터 스핀들에서 산정 및 CTAB 커브 테이블 사용) 에 의해 정의된 방향으로 이송됩니다.

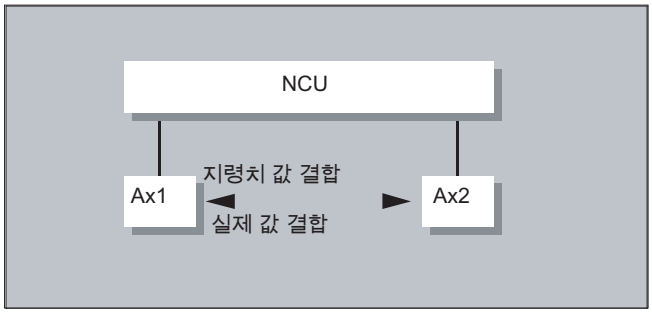


동기화 불가

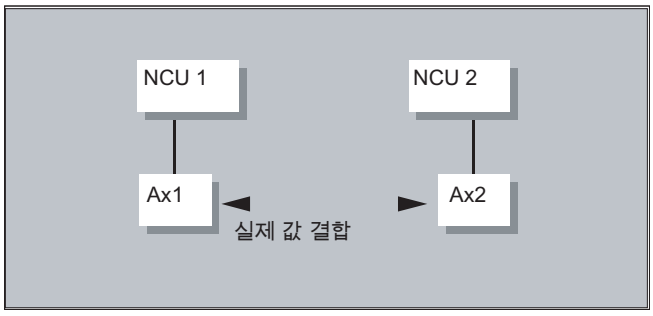
마스터 값 커플링이 활성 상태일 때 산정된 추종 축 위치가 현재 추종 축 위치를 벗어난 경우, 동기화를 실행할 수 없습니다.

실제 값과 지령치 커플링

지령치 커플링은 실제값 커플링에 비해 리딩 축과 추종 축 동기화에 더 효과적이므로 지령치 커플링이 기본값으로 설정됩니다.



리딩 축과 추종 축이 동일한 NCU에 의해 보간되는 경우에만 지령치 커플링이 가능합니다. 외부 리딩 축을 사용하는 경우, 추종 축은 실제값을 통해서만 리딩 축과 커플링될 수 있습니다.



전환은 셋팅 데이터 $SA_LEAD_TYPE을 통해 프로그래밍할 수 있습니다.

실제값과 지령치 커플링간의 전환은 반드시 추종 축이 정지했을 때 이뤄져야 합니다. 축이 정지하고 전환을 마친 후에 한하여 동기화가 가능합니다.

적용 예제

기계가 심하게 진동하는 동안에 실제값을 판독하면 오류가 발생합니다. 프레스 이송에 마스터 값 커플링을 사용하는 경우, 상황에 따라 진동이 가장 극심한 작업 단계에 실제값 커플링에서 지령치 커플링으로 전환할 필요가 있습니다.

지령치 결함을 이용한 **마스터 값 시뮬레이션**

기계 데이터를 사용해서 서보에서 리딩 축 보간자의 연결을 해제할 수 있습니다. 이런 방법으로 실제로는 리딩 축의 위치를 변경하지 않고서도 지령치 커플링용 지령치를 생성할 수 있습니다.

동기화 동작 등에 사용할 목적으로 다음 변수에서 지령치 연결로 생성된 마스터 값을 판독할 수 있습니다.

- $AA_LEAD_P 마스터 값 위치
- $AA_LEAD_V 마스터 값 속도

마스터 값 생성

다른 자체 프로그램으로 마스터 값을 생성할 수 있는 옵션이 있습니다. 이런 방법으로 생성된 마스터 값은 변수에 기록하거나 변수에서 읽어올 수 있습니다.

- $AA_LEAD_SP 마스터 값 위치
- $AA_LEAD_SV 마스터 값 속도

이 변수들을 사용하려면 먼저 셋팅 데이터 `$SA_LEAD_TYPE = 2`로 설정해야 합니다.

커플링 상태

다음과 같은 시스템 변수를 사용하여 NC 프로그램의 커플링 상태를 조회할 수 있습니다.

$AA_COUP_ACT[[축]]
0: 커플링 비활성
16: 마스터 값 커플링 활성화됨

동기화 동작의 상태 관리

전환 및 커플링 이벤트는 다음과 같은 실시간 변수로 관리됩니다.

$AC_MARKER[i] = n
관리 도구:
i 플래그 번호
n 상태 값

4.16.4 전자 기어 (EG)

'전자 기어' 기능을 사용하면 선형 이송 블록에 따라 **추종 축** 동작을 최대 5개의 **리딩 축** 기능처럼 관리할 수 있습니다. 각 리딩 축과 추종 축의 관계는 커플링 팩터로 정의됩니다.

추종 축 동작부는 모든 리딩 축 동작부에 각각 해당 커플링 팩터를 곱한 다음 가산해서 산정됩니다 전자 기어 축 그룹화가 활성화되면 지정된 위치와 관련한 추종 축을 동기화할 수 있습니다. 기어 그룹은 다음 작업이 가능합니다.

- 정의
- 활성화
- 비활성화
- 삭제

.

추종 축 동작은 다음 요소에서 선택적으로 도출할 수 있습니다.

- 리딩 축의 지령치 및
- 리딩 축의 실제값

각 리딩 축과 추종 축간의 비선형적 관계 역시 **커브 테이블**을 사용하여 확장 용도로 구현할 수도 있습니다('경로 이송 특성' 단원 참조). 전자 기어는 계단식(Cascade)으로 구성할 수 있습니다. 다시 말해서, 전자 기어의 추종 축이 전자 기어 추가용 리딩 축 역할을 할 수 있습니다.

4.16.4.1 전자 기어 정의(EGDEF)

전자 기어의 축 그룹은 추종 축 및 관련 커플링 형식이 존재하는 1개 이상 5개 이하의 리딩 축을 지정함으로써 정의됩니다.

요구사항

전자 기어 축 그룹 정의를 위한 요구사항:

추종 축에는 축 커플링을 정의할 수 없습니다 (또는 EGDEL을 사용하여 기존 축 커플링을 먼저 삭제해야 합니다).

구문

```
EGDEF(추종 축,리딩 축1,커플링 형식1,리딩 축2,커플링 형식2,...)
```

의미

EGDEF:	전자 기어 정의	
추종 축:	리딩 축의 영향을 받는 축	
리딩 축1 ,..., 리딩 축5	추종 축에 영향을 미치는 축	
커플링 형식1 ,..., 커플링 형식5	커플링 형식 모든 리딩 축에 대한 커플링 형식이 반드시 동일할 필요는 없으며 각각의 마스터 축을 기준으로 별도로 프로그래밍되어야 합니다.	
	값:	**의미:**
	0	추종 축이 해당 리딩 축의 **실제값**의 영향을 받습니다.
	1	추종 축이 해당 리딩 축의 **지령치**의 영향을 받습니다.

참고

전자 기어 축 그룹화가 정의되면 커플링 팩터가 0으로 사전 설정됩니다.

참고

EGDEF는 전처리 정지를 트리거합니다. 하나 이상의 리딩 축이 **커브 테이블**을 통해 동일한 추종 축에 영향을 미치는 경우 EGDEF의 기어박스 정의를 그대로 사용해야 합니다.

예

프로그램 코드	코멘트
EGDEF(C,B,1,Z,1,Y,1)	; EG 축 그룹 정의. 리딩 축 B, Z, Y가 지령치를 통해 추종 축 C에 영향을 미칩니다.

4.16.4.2 전자 기어박스 활성화(EGON, EGONSYN, EGONSYNE)

전자 기어 축 그룹을 활성화하는 방법으로는 세 가지가 있습니다.

구문

변용 1:

다음 구문으로 동기화하지 않은 채 전자 기어 축 그룹을 선택적으로 활성화합니다.
```
EGON(FA,"block change mode",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)
```

변용 2:

다음 구문으로 동기화한 채 전자 기어 축 그룹을 선택적으로 활성화합니다.
```
EGONSYN(FA,"block change mode",SynPosFA,[,LAi,SynPosLAi,Zi,Ni])
```

변용 3:

다음 구문으로 접근 모드를 지정하고 동기화한 채 전자 기어 축 그룹을 선택적으로 활성화합니다.
```
EGONSYNE(FA,"block change mode",SynPosFA,approach
mode[,LAi,SynPosLAi,Zi,Ni])
```

의미

변용 1:

FA	추종 축	
블록 변경 모드:	다음과 같은 모드를 사용할 수 있습니다.	
	"NOC"	즉시 블록 변경이 이뤄집니다.
	"FINE"	블록 변경이 '미세조정 동기화'로 수행됩니다.
	"COARSE"	블록 변경이 '일반 동기화'로 수행됩니다.
	"IPOSTOP"	지령치 기반 동기화를 위해 블록 변경이 수행됩니다.
LA1, ... LA5	리딩 축	

Z1, ... Z5	커플링 팩터 i의 분자
N1, ... N5	커플링 팩터 i의 분모 커플링 팩터 i = 분자 i/분모 i

EGDEF 명령으로 미리 지정한 리딩 축만 활성화 라인에 프로그래밍할 수 있습니다. 최소 하나의 리딩 축이 프로그래밍돼야 합니다.

변용 2:

FA	추종 축	
블록 변경 모드:	다음과 같은 모드를 사용할 수 있습니다.	
	"NOC"	즉시 블록 변경이 이뤄집니다.
	"FINE"	블록 변경이 '미세조정 동기화'로 수행됩니다.
	"COARSE"	블록 변경이 '일반 동기화'로 수행됩니다.
	"IPOSTOP"	지령치 기반 동기화를 위해 블록 변경이 수행됩니다.
[,LAi,SynPosLAi,Zi,Ni]	(대괄호는 입력하지 마십시오) 최소 1개, 최대 5개 연속:	
LA1, ... LA5	리딩 축	
SynPosLAi	i번째 리딩 축의 동기화 위치	
Z1, ... Z5	커플링 팩터 i의 분자	
N1, ... N5	커플링 팩터 i의 분모 커플링 팩터 i = 분자 i/분모 i	

EGDEF 명령으로 미리 지정한 리딩 축만 활성화 라인에 프로그래밍할 수 있습니다. 추종 축과 리딩 축에 프로그래밍된 "동기 위치" (SynPosFA 및 SynPosLA) 를 통해 축 그룹화가 *동기*로 해석되는 위치를 정의합니다. 그룹화가 가동됐을 때 전자 기어가 동기화된 상태가 아닌 경우, 추종 축이 지정된 동기화 위치로 이송합니다.

변용 3:

이 파라미터는 변용 2의 파라미터와 동일합니다.

접근 모드:	다음과 같은 모드를 사용할 수 있습니다.	
	"NTGT"	다음 간극의 시간 최적화 돌입
	"NTGP"	다음 간극의 경로 최적화 돌입
	"ACN"	- 방향으로 로터리 축 이동
	"ACP"	+ 방향으로 로터리 축 이동
	"DCT"	프로그래밍된 동기화 위치에 맞게 시간 최적화
	"DCP"	프로그래밍된 동기화 위치에 맞게 거리 최적화

변용 3은 모듈로 리딩 축과 커플링한 모듈로 추종 축에만 영향을 미칩니다. 시간 최적화에는 추종 축의 속도 한계가 고려됩니다.

추가 정보

활성화 버전 설명

변용 1:

그룹화가 실행된 순간, 리딩 축 및 추종 축의 위치가 '동기화된 위치'로 저장됩니다. '동기화된 위치'는 시스템 변수 $AA_EG_SYN로 판독할 수 있습니다.

변용 2:

커플링 그룹에 모듈로 축이 포함된 경우, 그 위치 값들은 절대치를 토대로 감소됩니다. 이는 다음 가능한 동기화된 위치에 접근할 수 있다록 해줍니다(소위 *상대적 동기화*: 예: 다음 톱니 간격). '추종 축 오버라이드 실행' 인터페이스 신호 DB(30 + 축 번호), DBX 26 비트 4가 발생한 경우에 한하여 동기화된 위치에 접근합니다. 대신, 위에서 언급한 신호가 설정될 때까지 EGONSYN 블록에서 프로그램이 중단되고 자체 정제 알람 16771이 출력됩니다.

변용 3:

절삭날 거리(각도)는 다음과 같이 계산됩니다: 360 * Zi/Ni. 호출 순간에 추종 축이 정지한 경우, 시간 최적화에 맞춰서 경로 최적화 반환 신호가 응답합니다.

추종 축이 이미 작동 중인 경우, 추종 축의 현재 속도와 무관하게 NTGP가 다음 절삭날 간극에서 동기화를 시작합니다. 추종 축이 이미 작동 중인 경우, 추종 축의 현재 속도에 맞춰 NTGT가 다음 절삭날 간극에서 동기화를 시작합니다. 필요한 경우, 축의 속도 역시 감소됩니다.

커브 테이블

리딩 축 중 하나에 **커브 테이블**이 사용된 경우:

Ni 선형 커플링을 위한 커플링 인자의 분모는 반드시 0으로 설정되어야 합니다. (분모 0은 선형 커플링의 경우 적합하지 않습니다.) 분모 0은 시스템에 다음 정보를 전달합니다.

Zi Zi는 사용할 커브 테이블의 번호입니다. 전원을 켰을 때 지정된 번호의 커브 테이블이 이미 정의된 상태여야 합니다.

LAi 지정된 리딩 축이 커플링 팩터를 통해 결합 (직선 커플링) 용으로 지정된 축에 응답합니다.

커브 테이블 및 캐스케이드 전자 기어 사용 및 동기화에 대한 자세한 내용은
추가 기능:
기능 매뉴얼 축 및 스핀들을 참고하십시오.

전원 ON, 리셋, 작동 모드 변경, 블록 탐색을 위한 전자 기어의 반응

- 전원을 켠 후에도 모든 커플링은 활성화되지 **않습니다**.
- 리셋이나 작동 모드 전환으로 인해 활성 커플링 상태가 달라지지 않습니다.
- 블록 탐색 도중, 전환 명령, 전자 기어 삭제 및 정의가 실행되거나 호출되지 않고 무시됩니다.

전자 기어의 시스템 변수

가공 프로그램은 전자 기어의 시스템 변수를 통해 전자 기어의 현재 축 커플링 상태를 파악하고 필요한 경우 이에 반응합니다.

전자 기어 박스의 시스템 변수는 다음과 같이 설계됩니다.

$AA_EG_ ...

또는

$VA_EG_ ...

추가 정보
기능 매뉴얼 시스템 변수

4.16.4.3 전자 기어박스 활성화(EGOFS, EGOFC)

전자 기어 축 그룹을 비활성화하는 방법으로는 세 가지가 있습니다.

프로그래밍

변용 1:

구문	의미
EGOFS(추종 축)	전자 기어가 비활성화됩니다. 추종 축이 제동으로 인해 정지합니다. 이 구문을 호출하면 사전 처리 중지 기능이 실행됩니다.

변용 2:

구문	의미
EGOFS(추종 축,리딩 축1, …,리딩 축5)	이 명령 파라미터 설정을 사용하면 각각의 리딩 축이 추종 축의 동작에 미치는 영향을 **선택적으로** 제거할 수 있습니다.

최소 하나의 리딩 축이 지정돼야 합니다. 지정된 리딩 축이 추종 축에 미치는 영향이 선택적으로 차단됩니다. 이 구문을 호출하면 사전 처리 중지 기능이 실행됩니다. 구문에 활성 상태의 리딩 축이 포함된 경우, 추종 축이 계속 리딩 축의 영향을 받습니다. 이 방법으로 모든 리딩 축의 영향이 배제된 경우, 추종 축이 제동으로 인해 정지합니다.

변용 3:

구문	의미
EGOFC(추종 스핀들1)	전자 기어가 비활성화됩니다. 비활성화 시점에 적용된 속도로 추종 스핀들이 계속 이동합니다. 이 구문을 호출하면 사전 처리 중지 기능이 실행됩니다.

참고

이 변용은 스핀들에만 사용할 수 있습니다.

4.16.4.4 전자 기어 정의 삭제(EGDEL)

전자 기어의 정의를 삭제하려면 먼저 전자 기어의 축 그룹을 비활성화해야 합니다.

프로그래밍

구문	의미
EGDEL(추종 축)	축 그룹의 커플링 정의가 삭제됩니다. 동시 활성화가 가능한 최대 축 그룹 갯수에 도달할 때까지 EGDEF를 사용해서 축 그룹을 추가로 정의할 수 있습니다. 이 구문을 호출하면 사전 처리 중지 기능이 실행됩니다.

4.16.4.5 회전 이송 속도(G95)/전자 기어(FPR)

FPR 명령을 사용하여 전자 기어의 추종 축을 회전 이송 속도를 결정하는 축으로 지정할 수 있습니다. 이 명령에 대해서는 아래 내용을 참조하십시오.

- 이송 속도는 전자 기어 추종 축의 지령치 속도에 의해 결정됩니다.
- 지령치 속도는 리딩 스핀들 및 (경로 축에 해당하지 않는) 모듈로 축의 속도와 관련 커플링 팩터로 산정됩니다.
- 선형 또는 비모듈로 리딩 축의 속도 부분과 추종 축의 오버레이 이동은 고려되지 않습니다.

4.16.5 동기 스핀들

동기식 작동에는 **동기 스핀들 쌍**으로 불리우는 추종 스핀들(FS)과 리딩 스핀들(LS)이 포함됩니다. 정의된 기능적 상관성에 따라 커플링이 활성화(동기식 작동)되면 추종 스핀들이 리딩 스핀들의 동작을 모방합니다.

채널 머신 데이터를 통해 고정된 설정에 각 기계의 동기 스핀들 쌍을 지정하거나 CNC 가공 프로그램을 통해 특정 용도에 맞게 동기 스핀들 쌍을 정의할 수 있습니다. 각 NC 채널에서 최대 2개의 동기 스핀들 쌍이 동시에 작동할 수 있습니다.

다음의 커플링 동작에 대해서는 해당 가공 프로그램을 참조하십시오.

- 정의 또는 변경
- 활성화
- 비활성화
- 삭제

.

또한 소프트웨어의 상태에 따라...

- 동기 조건을 충족할 때까지 대기할 수 있습니다.
- 블록 변경 방법을 변경할 수 있습니다.
- 지령치 커플링 또는 실제값 커플링 형식을 선택하거나 리딩 스핀들과 추종 스핀들 간의 각도 옵셋을 지정할 수 있습니다
- 커플링이 활성화되면 추종 축의 이전 프로그램이 전송됩니다.
- 측정되거나 확인된 동기 편차를 수정합니다.

4.16.5.1 동기 스핀들: 프로그래밍(COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

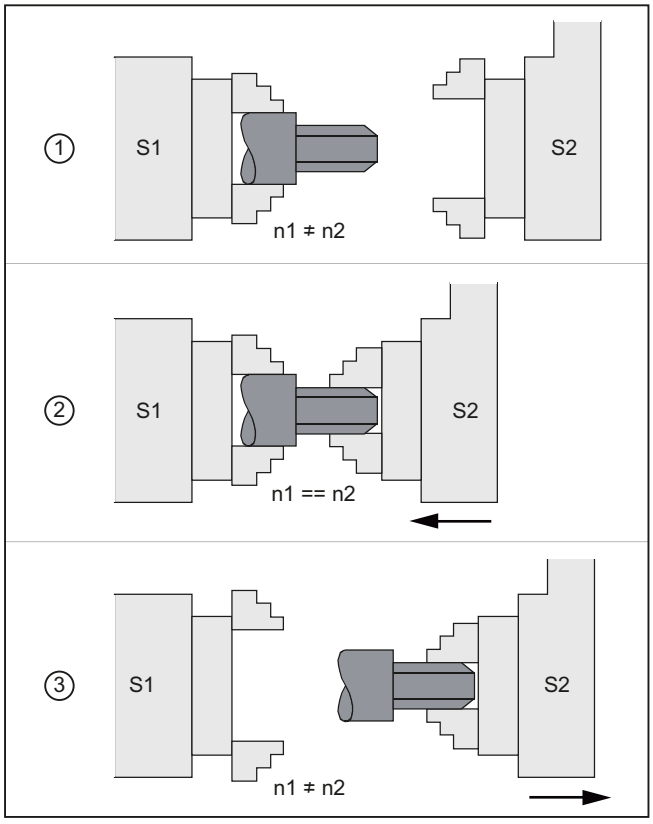
"동기 스핀들" 기능을 사용하면 프로그래밍된 변환 비율을 적용하여 추종 스핀들 (FS) 과 리딩 스핀들 (LS) 의 속도 동기 이송이 가능합니다.

이 기능은 다음의 모드를 지원합니다.

- 속도 동기 ($n_{FS} = n_{LS}$)
- 위치 동기 ($\phi_{FS} = \phi_{LS}$)
- 각도 옵셋으로 위치 동기 ($\phi_{FS} = \phi_{LS} + \Delta\phi$)

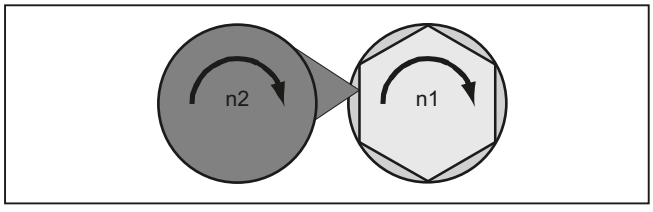
적용 예제:

- 후면 가공을 위해 플라잉 공작물 전송, 변환 비율: 1:1



① 속도 동기화
② 공작물 전송
③ 후면 가공

- 다각형 가공 (다각형 선삭), 속도 동기, 변환 비율: n_1:n_2



구문

```
COUPDEF(<FS>,<LS>,<ZFS>,<NLS>,<block change>,<coupling type>)
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>,<block change>,<LS>,<block change>)
```

참고

약어 표시

COUPOF, COUPOFS, COUPRES 및 COUPDEL 구문의 경우 리딩 스핀들을 지정하지 않고 짧게 표기할 수 있습니다.

의미

COUPDEF:	사용자 기반의 커플링 정의/변경
COUPON:	커플링 활성화. 추종 스핀들이 실제 속도를 기준으로 리딩 스핀들과 동기화합니다.
COUPONC:	이전에 프로그래밍된 M3 S... 또는 M4 S...로 동작 시 커플링. 추종 스핀들의 속도차가 즉시 전송됩니다.
COUPOF:	커플링 해제 • 즉각적인 블록 변경 시: COUPOF(<S2>,<S1>) • <POSFS> 또는 <POSLS> 비활성화 위치가 교차한 후에만 블록 변경: COUPOF(<S2>,<S1>,<POSFS>) COUPOF(<S2>,<S1>,<POSFS>,<POSLS>)
COUPOFS:	추종 스핀들 정지를 통한 커플링 해제. 즉각적인 블록 변경으로 가급적 빨리 블록 변경: COUPOFS(<S2>,<S1>) 스위치 OFF 위치를 통과한 이후에만 블록 변경: COUPOFS(<S2>,<S1>,<POSFS>)
COUPRES:	설정된 MD 및 SD로 커플링 파라미터 리셋.
COUPDEL:	사용자 정의 커플링 삭제.
WAITC:	동기 실행 조건 대기 (블록 변경 중 NOC가 IPO로 증가)
<FS>:	추종 스핀들 지정
옵션 파라미터:	

<LS>:	메인 스핀들 지정 스핀들 번호로 지정 (예: S2, S1)	
<ZFS>, <NLS>:	FS 및 LS 간의 변환 비율. <ZFS>/<NLS> = 분자/분모 초기 설정: <ZFS> / <NLS> = 1.0 ; 분모 지정은 옵션	
<block change>:	블록 변경 특성 블록 변경 특성은 다음과 같습니다.	
	"NOC"	즉시
	"FINE"	'미세 동기' 도달 시
	"COARSE"	'일반 동기' 도달 시
	"IPOSTOP"	IPOSTOP 도달 시. 즉, 지령치에 의한 동기 (표준) 이후
	블록 변경 특성은 모달로 효력이 발생합니다.	
<coupling type>:	커플링 유형: FS와 LS간의 커플링	
	"DV"	지령치 커플링 (기본 설정)
	"AV"	실제값 커플링
	"VV"	속도 커플링
	커플링 유형은 모달입니다.	
<POSFS>:	리딩 스핀들과 추종 스핀들간의 각도 옵셋	
	값 범위:	0°... 359.999°
<POSFS>,<POSLS>:	추종 스핀들과 리딩 스핀들의 스위치 OFF 위치 "POS_{FS}, POS_{LS}를 통과하고 나면 블록 변경이 실행됩니다"	
	값 범위:	0°... 359.999°

예

리딩 스핀들과 추종 스핀들을 이용한 작업

프로그램 코드	코멘트
	리딩 스핀들 = 마스터 스핀들 = 스핀들 1
	추종 스핀들 = 스핀들 2
N05 M3 S3000 M2=4 S2=500	리딩 스핀들 3000 rpm으로 회전, 추종 스핀들 500 rpm으로 회전
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	커플링 정의 (설정 가능)
...	

프로그램 코드	코멘트
N70 SPCON	리딩 스핀들을 폐-루프 위치 제어 상태로 전환 (지령치 커플링)
N75 SPCON(2)	추종 스핀들을 폐-루프 위치 제어 상태로 전환
N80 COUPON(S2,S1,45)	옵셋 위치를 45°로 즉시 커플링
...	
N200 FA[S2]=100	포지셔닝 속도 = 100도/분
N205 SPOS[2]=IC(-90)	음의 방향으로 90° 오버레이해서 이송.
N210 WAITC(S2,"Fine")	'미세조정' 동기 대기
N212 G1 X... Y... F...	가공
...	
N215 SPOS[2]=IC(180)	양의 방향으로 180° 오버레이해서 이송.
N220 G4 S50	드웰 시간 = 마스터 스핀들 50회 회전
N225 FA[S2]=0	설정된 속도 활성화 (MD)
N230 SPOS[2]=IC(-7200)	20회 회전. 음의 방향으로 설정된 속도로 회전.
...	
N350 COUPOF(S2,S1)	즉시 커플링 해제, S=S2=3000
N355 SPOSA[2]=0	0°에서 FS 정지
N360 G0 X0 Y0	
N365 WAITS(2)	스핀들 2 대기
N370 M5	FS 정지
N375 M30	

속도차 프로그래밍

프로그램 코드	코멘트
	리딩 스핀들 = 마스터 스핀들 = 스핀들 1
	추종 스핀들 = 스핀들 2
N01 M3 S500	500 rpm으로 리딩 스핀들 회전
N02 M2=3 S2=300	300 rpm으로 추종 스핀들 회전
...	
N10 G4 F1	마스터 스핀들의 드웰 시간
N15 COUPDEF (S2,S1,-1)	커플링 팩터 -1:1
N20 COUPON(S2,S1)	커플링 활성화. 스핀들의 속도는 리딩 스핀들의 속도와 커플링 팩터에 따라 달라집니다.
...	
N26 M2=3 S2=100	속도차 프로그래밍

속도차를 적용한 이동 명령 전송 예제

1. COUPON으로 추종 스핀들의 이전 프로그래밍 도중 커플링 활성화

프로그램 코드	코멘트
	리딩 스핀들 = 마스터 스핀들 = 스핀들 1
	추종 스핀들 = 스핀들 2
N05 M3 S100 M2=3 S2=200	리딩 스핀들 100 rpm으로 회전, 추종 스핀들 200 rpm으로 회전.

프로그램 코드	코멘트
N10 G4 F5	마스터 스핀들의 드웰 시간 = 5초
N15 COUPDEF(S2,S1,1)	FS에서 LS로의 변환 비율 = 1.0 (디폴트)
N20 COUPON(S2,S1)	리딩 스핀들에 즉시 커플링
N10 G4 F5	100 rpm으로 추종 스핀들 회전

2. COUPONC으로 추종 스핀들의 이전 프로그래밍 도중 커플링 활성화

프로그램 코드	코멘트
	리딩 스핀들 = 마스터 스핀들 = 스핀들 1
	추종 스핀들 = 스핀들 2
N05 M3 S100 M2=3 S2=200	리딩 스핀들 100 rpm으로 회전, 추종 스핀들 200 rpm으로 회전.
N10 G4 F5	마스터 스핀들의 드웰 시간 = 5초
N15 COUPDEF(S2,S1,1)	FS에서 LS로의 변환 비율 = 1.0 (디폴트)
N20 COUPONC(S2,S1)	리딩 스핀들에 즉시 커플링 및 S2에 이전에 사용한 속도를 전송
N10 G4 F5	300 rpm(100 rpm + 200 rpm)으로 S2 회전

3. COUPON으로 추종 스핀들 정지 커플링 활성화

프로그램 코드	코멘트
	리딩 스핀들 = 마스터 스핀들 = 스핀들 1
	추종 스핀들 = 스핀들 2
N05 SPOS=10 SPOS[2]=20	포지셔닝 모드의 추종 스핀들 S2
N15 COUPDEF(S2,S1,1)	FS에서 LS로의 변환 비율 = 1.0 (디폴트)
N20 COUPON(S2,S1)	리딩 스핀들에 즉시 커플링
N10 G4 F1	커플링 종료, S2는 20°에서 정지.

4. COUPONC으로 추종 스핀들 정지 커플링 활성화

참고

포지셔닝 모드 또는 축 모드

커플링 전에 추종 스핀들이 포지셔닝 모드 또는 축 모드에 있는 경우 추종 스핀들은 COUPON(<FS>,<LS>)과 COUPONC(<FS>,<LS>)의 경우와 동일하게 동작합니다.

참고

리딩 스핀들 및 축 동작

커플링을 정의하기 전에 리딩 스핀들이 동작에 연관된 경우, 머신 데이터의 속도 한계값 MD32000 $MA_MAX_AX_VELO (최대 축 속도)은 커플링이 활성화된 후에도 적용됩니다. 이런 동작을 방지하려면 커플링을 정의하기 전에 축을 스핀들 모드 (M3 S... 또는 M4 S...)로 전환해야 합니다.

추가 정보

커플링 구성

커플링 구성을 위한 리딩 스핀들과 추종 스핀들은 머신 데이터로 정의합니다. 설정된 스핀들은 가공 프로그램에서 변경할 수 없습니다. 쓰기 보호가 설정되어 있지 않은 경우, COUPDEF를 사용하여 가공 프로그램에서 커플링을 설정할 수 있습니다.

사용자 정의 커플링

COUPDEF를 사용하면 가공 프로그램에서 커플링을 재정의 또는 변경할 수 있습니다. 커플링이 이미 활성 상태인 경우에는 먼저 COUPDEL을 사용하여 삭제한 후 새 커플링을 정의해야 합니다.

다음에 의해 커플링이 정의됩니다:

```
COUPDEF(<FS>,<LS>,<TFS>,<TLS>, 블록 변경 동작, 커플링 유형)
```

추종 스핀들 (FS) 및 리딩 스핀들 (LS)

커플링은 FS 및 LS의 축 이름을 사용하여 고유하게 정의합니다. 모든 COUPDEF 명령에 이 축 이름을 프로그래밍해야 합니다. 다른 파라미터들은 모달이므로, 변경할 필요가 있을 경우에만 프로그램 해야 합니다.

예:

```
COUPDEF(S2,S1)
```

변환 비율

변환 비율은 FS와 LS의 속도 비율로 정의합니다.

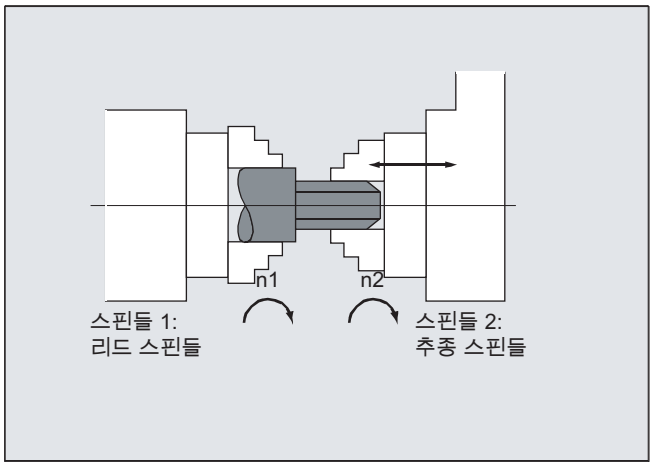
추종 스핀들/리딩 스핀들 = 분자/분모

분자를 프로그래밍해야 합니다. 분모는 프로그래밍하지 않아야 합니다. 그런 다음 분모에 대해 기본값 1.0이 설정됩니다.

예:

추종 스핀들 S2 및 리딩 스핀들 S1, 변환 비율 = 1/1

```
COUPDEF(S2, S1, 1.0)
```



참고

또한 커플링이 활성 상태이고 스핀들이 회전하는 경우 즉시 변환 비율을 변경할 수 있습니다.

블록 변경 동작 NOC, FINE, COARSE, IPOSTOP

블록 변경 동작을 프로그래밍할 때 다음 약어 표시를 사용할 수 있습니다.

- "NO": 즉시 (표준)
- "FI": '미세 동기' 도달 시
- "CO": '일반 동기' 도달 시
- "IP": IPOSTOP 도달 시, 즉 지령치 기준 동기 후

커플링 유형

참고

커플링 유형은 커플링이 비활성인 경우에만 변경할 수 있습니다.

동기 모드 활성화 COUPON, <POSFS>

- LS와 FS간의 각도 옵셋을 기준으로 커플링 활성화:
  - COUPON(S2,S1)
  - COUPON(S2)
- 각도 옵셋 <POSFS>로 커플링 활성화
  <POSFS>는 양의 회전 방향에서 리딩 스핀들의 0° 위치를 기준으로 합니다.
  <POSFS> 값 범위: 0°... 359.999°
  - COUPON(S2,S1,30)

참고

각도 옵셋은 커플링이 활성일 때도 변경할 수 있습니다.

추종 스핀들의 포지셔닝

동기 스핀들 커플링이 활성화된 경우에도 LS과 관계 없이 ±180° 범위 내에서 FS를 포지셔닝할 수 있습니다.

- SPOS로 FS의 스핀들 포지셔닝
  예: SPOS[2] = IC(-90)
  SPOS에 대한 세부 정보는
  추가 정보
  스핀들 포지셔닝 (SPOS, SPOSA, M19, M70, WAITS) (페이지 131)에 나와 있습니다.

속도 차이

속도 차이가 발생하면 속도 제어 모드가 실행되고 동기 스핀들 커플링이 활성화되어 LS 이동으로 인한 FS 속도와 스핀들 프로그래밍에 따른 FS 속도를 오버레이합니다.

- COUPONC으로 동기 스핀들 커플링
- S<FS>=<speed> [M<FS>=<direction of rotation>]

  참고

  보충 조건
  - 속도 S...를 회전 방향 M3/M4와 함께 다시 프로그래밍해야 합니다.
  - 동기 스핀들 커플링 COUPONC를 사용하여 LS 이동을 통해 스핀들 속도 (M<direction of rotation> S<FS>)를 오버레이하는 기능은 오버레이가 활성화된 경우에만 실행됩니다.
  - 리딩 스핀들의 다이나믹 반응은 추종 스핀들에 오버레이가 적용될 때 다이나믹 한계 값을 초과하지 않는 수준으로 제한해야 합니다.

  차동 속도에 대한 추가 정보는
  추가 정보
  기능 매뉴얼 축 및 스핀들을 참고하십시오.

속도, 가속도: FA, ACC, OVRA, VELOLIMA

추종 스핀들의 축 속도 및 가속도는 다음을 사용하여 프로그래밍할 수 있습니다.

- FA[SPI(S<n>)] 또는 FA[S<n>] (축 속도)
- ACC[SPI(S<n>)] 또는 ACC[S<n>] (축 가속)
- OVRA[SPI(S<n>)] 및 OVRA[S<n>] (축 오버라이드)
- VELOLIMA[SPI(S<n>)] 및 VELOLIMA[S<n>] (각각 축 속도 증가 및 감소)

<n> = 1, 2, 3 이용, ... (추종 스핀들의 스핀들 번호)

추가 정보

속도 (피드) 제어 (페이지 117)

참고

최대 축 저크의 감소 또는 증가는 스핀들에 영향을 미치지 않습니다.

축의 동적 성능에 대한 추가 정보는

추가 정보

기능 매뉴얼 축 및 스핀들에 나와 있습니다.

프로그램 가능한 블록 변경 동작 WAITC

커플링 파라미터 또는 포지셔닝 동작 변경 후와 같은 상황에서 WAITC를 사용하면 다양한 동기 조건 (일반, 미세, IPOSTOP) 을 적용하여 블록 변경 동작을 정의할 수 있습니다. 동기 조건을 지정하지 않으면 COUPDEF 정의에서 지정한 블록 변경 동작이 적용됩니다.

예

- 추종 스핀들 S2에 대한 동기 조건 FINE (미세 동기) 및 추종 스핀들 S4에 대한 동기 조건 COARSE (일반 동기) 충족 대기 WAITC(S2,"FINE",S4,"COARSE")
- COUPDEF에 따른 동기 조건 충족 대기: WAITC( )

커플링 해제 COUPOF

COUPOF는 커플링 해제 동작을 정의하는 데 사용할 수 있습니다.

- 블록 변경 후 즉시 커플링 해제:
  - COUPOF(S2,S1) (리딩 스핀들 지정)
  - COUPOF(S2) (리딩 스핀들을 지정하지 않음)
- 스위치 off 위치에 도달된 후 커플링 해제. 스위치 OFF 위치에 도달한 후 블록 변경이 이루어집니다.
  - COUPOF(S2,S1,150) (FS 스위치 OFF 위치: 150°)
  - COUPOF(S2,S1,150,30) (FS 스위치 OFF 위치: 150°, LS: 30°)

추종 스핀들 정지를 통한 커플링 해제 COUPOFS

COUPOFS는 추종 스핀들 정지를 통한 커플링 해제 동작을 정의하는 데 사용할 수 있습니다.

- 추종 스핀들 정지 및 즉각적인 블록 변경을 통한 커플링 해제:
  - COUPOFS(S2,S1) (리딩 스핀들 지정)
  - COUPOFS(S2) (리딩 스핀들을 지정하지 않음)
- 추종 스핀들 정지를 통해 스위치 off 위치를 통과한 후 커플링 해제. 스위치 OFF 위치에 도달한 후 블록 변경이 이루어집니다.
  - COUPOFS(S2,S1,150) (FS 스위치 OFF 위치: 150°)

커플링 삭제 COUPDEL

COUPDEL은 커플링을 삭제합니다.

- COUPDEL(S2,S1) (리딩 스핀들 지정)
- COUPDEL(S2) (리딩 스핀들 미지정)

커플링 파라미터 리셋 COUPRES

COUPRES는 머신 데이터 및 설정 데이터에서 설정된 커플링 값을 활성화합니다.

- COUPRES(S2,S1) (리딩 스핀들 지정)
- COUPRES(S2) (리딩 스핀들 미지정)

시스템 변수

- 추종 스핀들의 현재 커플링 상태
  추종 스핀들의 현재 커플링 상태는 다음을 통해 비트 코드로 읽을 수 있습니다.
  <Value> = $AA_COUP_ACT[<FS>]

비트	<value>	의미
-	0	커플링 없음
2	4	동기 스핀들 커플링 동작

참고

- 그 외 모든 값은 축 모드를 기준으로 합니다.
- 스핀들이 추종 스핀들이거나 커플링이 여러 개인 경우 모든 커플링의 커플링 상태 값이 전체 상태로 반환됩니다.

- 현재 각도 옵셋
  리딩 스핀들 대비 추종 스핀들의 현재 각도 옵셋은 다음을 통해 읽을 수 있습니다.
  – `$AA_COUP_OFFS[<FS>]` (지령치에 각도 옵셋)
  – `$VA_COUP_OFFS[<FS>]` (실제 값에 각도 옵셋)
  적용 예
  추종 모드 취소 후 NC 프로그램에서 각도 옵셋 차이 수정:
  각도 옵셋 차이 = 프로그래밍된 각도 옵셋 - 시스템 변수

추가 정보

시스템 변수에 대한 자세한 정보는 다음을 참조하십시오.

파라미터 매뉴얼 시스템 변수

4.16.6 일반 커플링(CP...)

"일반 커플링"은 기존 커플링 유형에 대한 모든 커플링 특성을 결합한 일반 커플링 기능을 말합니다. 예: 커플 모션, 리딩 값 커플링, 전자 변속기 및 동기 스핀들.

이 기능은 프로그래밍 유연성을 극대화합니다.

- 사용자들은 자신들이 필요로 하는 커플링 속성을 선택할 수 있습니다(빌딩 블록 원리).
- 커플링 속성을 각각 개별적으로 프로그래밍할 수 있습니다.
- 정의된 커플링에 대한 커플링 속성(예: 커플링 계수)을 변경할 수 있습니다.
- 추가 커플링 속성을 나중에 사용할 수 있습니다.
- 추종축의 참조 좌표계를(기본 좌표계 또는 기계 좌표계) 프로그래밍할 수 있습니다.
- 일부 커플링 속성은 동기 동작을 통해 프로그래밍할 수 있습니다.
  추가 정보: 기능 매뉴얼 동기 동작

참고

커플 모션 (TRAIL*), 리딩 값 커플링(LEAD*), 전자 변속기(EG*) 및 동기 스핀들(COUP*)을 필요로 하는 이전 커플링은 적응 싸이클을 통해 지원됩니다.

모든 키워드 및 커플링 특성 개요

다음 표는 일반 커플링 및 프로그래밍 커플링 특성에 대한 모든 키워드를 간략하게 소개합니다.

키워드	커플링 특성/의미	구문		
CPDEF	커플링 모듈 생성	CPDEF=(<FAx>)		
CPDEL	커플링 모듈 삭제	CPDEL=(<FAx>)		
CPLA	리딩 축의 정의	CPLA[<FAx>]=(<LAx>)		
CPLDEF	리딩 축의 정의 및 커플링 모듈의 생성 (CPDEF + CPLA와 가능)	CPLDEF[<FAx>]=(<LAx>) 또는 CPDEF=(<FAx>) CPLA[<FAx>]=(<LAx>)		
CPLDEL	커플링 모듈의 리딩 축 삭제 (CPDEF + CPLA와 가능)	CPLDEL[<FAx>]=(<LAx>) 또는 CPDEL=(<FAx>) CPLA[<FAx>]=(<LAx>)		
CPON	커플링 모듈 설정	CPON=(<FAx>)		
CPOF	커플링 모듈 해제	CPOF=(<FAx>)		
CPLON	커플링 모듈의 리딩 축 설정	CPLON[<FAx>]=<LAx>		
CPLOF	커플링 모듈의 리딩 축 해제	CPLOF[<FAx>]=<LAx>		
CPLNUM	커플링 팩터의 분자	CPLNUM[FAx,LAx]=<value>		
CPLDEN	커플링 팩터의 분모	CPLDEN[FAx,LAx]=<value>		
CPLCTID	커브 테이블 수	CPLCTID[FAx,LAx]=<value>		
CPLSETVAL	커플링 기준	CPLSETVAL[FAx,LAx]="<coupling reference>"		
		"<coupling reference>":	"CMDPOS"	지령치 커플링
			"CMDVEL"	속도 커플링
			"ACTPOS"	실제값 커플링
CPFRS	참조 좌표계	CPFRS[FAx]="<coordinate reference>"		
		"<coordinate reference>":	"BCS"	기본 좌표계
			"MCS"	기계 좌표계

키워드	커플링 특성/의미	구문		
CPBC	블록 변경 기준	CPBC[FAx]="<block change criterion>"		
		"<block change criterion>":	"NOC"	블록 변경은 커플링 상태와 따로 수행됩니다.
			"IPOSTOP"	블록 변경은 지령치 동기를 통해 수행됩니다.
			"COARSE"	블록 변경은 실제 값 동기 "일반"을 통해 수행됩니다.
			"FINE"	블록 변경은 실제 값 동기 "미세"를 통해 수행됩니다.
CPFPOS + CPON	커플링을 설정할 때 추종 축의 동기 위치	CPON=FAx CPFPOS[FAx]=<value>		
CPLPOS + CPON	커플링을 설정할 때 리딩 축의 동기 위치	CPLPOS[FAx,LAx]=<value>		

키워드	커플링 특성/의미	구문		
CPFMSON	동기화 모드	CPFMSON[FAx]="<synchronization mode>"		
		"<synchronization mode>":	"CFAST"	커플링이 시간 최적화에 근접합니다.
			"CCOARSE"	커플링 규칙에 따라 필요로 하는 추종 축 위치가 다음과 같은 현재 축 위치 범위에 있는 경우에만 커플링이 근접합니다.
			"NTGT"	다음 간극이 시간 최적화에 근접합니다.
			"NTGP"	다음 간극이 경로 최적화에 근접합니다.
			"NRGT"	다음 분할은 기어 갯수 대비 이의 갯수에 따라 시간 최적화 방식으로 접근합니다.
			"NRGP"	다음 분할은 기어 갯수 대비 이의 갯수에 따라 경로 최적화 방식으로 접근합니다.
			"ACN"	회전축만 해당됩니다. 회전축이 음(-) 축 방향의 동기 위치로 이동합니다. 동기화가 즉시 이루어집니다.
			"ACP"	회전축만 해당됩니다. 회전축이 양(+) 축 방향의 동기 위치로 이동합니다. 동기화가 즉시 이루어집니다.
			"DCT"	회전축만 해당됩니다. 회전축은 프로그래밍된 동기 위치 시간 최적화 방향으로 이동합니다.

키워드	커플링 특성/의미	구문		
				동기화가 즉시 이루어집니다.
			"DCP"	회전축만 해당됩니다. 회전축은 프로그래밍된 동기 위치 경로 최적화 방향으로 이동합니다. 동기화가 즉시 이루어집니다.
CPFMON	설정 상태에서 추종축의 동작	CPFMON[FAx]= "<switch-on behavior>"		
		"<switch-on behavior>":	"STOP"	스핀들에만 적용됩니다. 설정 상태 이전에 추종 스핀들의 활성 모션이 멈춥니다.
			"CONT"	스핀들 및 주 이송 축에만 해당됩니다. 추종축/스핀들의 현재 모션은 시작 모션인 커플링으로 전환됩니다.
			"ADD"	스핀들에만 적용됩니다. 현재 오버레이 모션(예: 현재 추종 축/스핀들 모션) 외 작동하는 커플링 모션 콤포넌트는 오버레이 모션으로 유지됩니다.

키워드	커플링 특성/의미	구문		
CPFMOF	완전 해제 상태에서 추종 축의 동작	CPFMOF[FAx]="<switch-off behavior>"		
		"<switch-off behavior>":	"STOP"	추종축/스핀들을 중지합니다. 활성 오버레이 모션도 중지됩니다. 이어서 커플링이 열립니다.
			"CONT"	스핀들 및 주 이송 축에만 해당됩니다. 비활성화 시점에 적용된 속도로 추종 스핀들이 계속 이동합니다.
CPFPOS + CPOF	설정 해제시 추종축의 해제 위치	CPOF=(FAx) CPFPOS[FAx]=<value>		

키워드	커플링 특성/의미	구문		
CPMRESET	RESET에 대한 커플링 동작	CPMRESET[FAx]="<Reset behavior>"		
		"<reset behavior>":	"NONE"	현재 커플링 상태는 그대로 유지됩니다.
			"ON"	적절한 커플링 모듈이 만들어지면 커플링을 설정합니다. 정의된 모든 리딩축 관계가 활성화됩니다. 이것은 해당 리딩축 관계의 전체 또는 일부가 활성인 경우에 수행됩니다. 예를 들어 커플링이 완전 활성화되어도 재동기화가 이루어집니다.
			"OF"	활성 오버레이 모션도 중지됩니다. 이어서 커플링이 비활성화됩니다. 명시적 정의(CPDEF)를 사용하지 않고 적절한 커플링 모듈을 만들 경우 해당 커플링 모듈은 삭제됩니다. 그렇지 않고 해당 모듈이 그대로 유지되면 아직 사용할 수 있는 것입니다.
			"OFC"	스핀들에만 적용됩니다. 비활성화 시점에 적용된 속도로 추종 스핀들이 계속 이동합니다. 커플링이 해제되었습니다. 명시적 정의(CPDEF)를 사용하지 않고 적절한 커플링 모듈을 만들 경우 해당 커플링 모듈은 삭제됩니다. 그렇지 않고 해당 모듈이 그대

키워드	커플링 특성/의미	구문		
				로 유지되면 아직 사용할 수 있는 것입니다.
			"DEL"	활성 오버레이 모션도 중지됩니다. 커플링이 비활성화된 후 즉시 삭제됩니다.
			"DELC"	스핀들에만 적용됩니다. 비활성화 시점에 적용된 속도로 추종 스핀들이 계속 이동합니다. 커플링이 비활성화된 후 즉시 삭제됩니다.

키워드	커플링 특성/의미	구문		
CPMSTART	가공 프로그램 시작 시 커플링 동작	CPMSTART[FAx]="<start behavior>"		
		"<start behavior>":	"NONE"	현재 커플링 상태는 그대로 유지됩니다.
			"ON"	커플링이 ON으로 설정됩니다. 정의된 모든 리딩축 관계가 활성화됩니다. 이것은 해당 리딩축 관계의 전체 또는 일부가 활성인 경우에 수행됩니다. 예를 들어 커플링이 완전 활성화되어도 재동기화가 이루어집니다.
			"OF"	커플링이 해제되었습니다. 명시적 정의(CPDEF)를 사용하지 않고 적절한 커플링 모듈을 만들 경우 해당 커플링 모듈은 삭제됩니다. 그렇지 않고 해당 모듈이 그대로 유지되면 아직 사용할 수 있는 것입니다.
			"DEL"	커플링이 비활성화된 후 즉시 삭제됩니다.
CPMPRT	프로그램 테스트를 통한 검색 실행 시 가공 프로그램 시작에서의 커플링 반응	CPMPRT[FAx]="<start behavior>"		
		"<start behavior>":	CPMSTART 참조	
CPLINTR	리딩 축의 입력 값에 대한 옵셋 값	CPLINTR[FAx,LAx]=<value>		
CPLINSC	리딩 축 입력 값에 대한 배율 계수	CPLINSC[FAx,LAx]=<value>		
CPLOUTTR	커플링의 출력 값에 대한 옵셋 값	CPLOUTTR[FAx,LAx]=<value>		

키워드	커플링 특성/의미	구문		
CPLOUTSC	커플링의 출력 값에 대한 배율 계수	CPLOUTSC[FAx,LAx]=<value>		
CPSYNCOP	위치 동기 "일반"의 임계값	CPSYNCOP[FAx]=<value>		
CPSYNFIP	위치 동기 "미세"의 임계값	CPSYNFIP[FAx]=<value>		
CPSYNCOP2	"일반" 위치 동기에 대한 두 번째 임계값	CPSYNCOP2[FAx]=<value>		
CPSYNFIP2	"미세" 위치 동기에 대한 두 번째 임계값	CPSYNFIP2[FAx]=<value>		
CPSYNCOV	속도 동기 "일반"의 임계값	CPSYNCOV[FAx]=<value>		
CPSYNFIV	속도 동기 "미세"의 임계값	CPSYNFIV[FAx]=<value>		
CPMBRAKE	특정 정지 신호 및 정지 명령에 대한 추종 축의 반응	CPMBRAKE[FAx]=<bit-coded value>		
CPMVDI	특정 NC/PLC 인터페이스 신호에 대한 추종 축의 반응	CPMVDI[FAx]=<bit-coded value>		
CPMALARM	특수 커플링 관련 알람 출력의 억제	CPMALARM[FAx]=<bit-coded value>		
CPSETTYPE	커플링 형식	CPSETTYPE[FAx]="<coupling type>"		
		"<coupling type>":	"CP"	자유로운 프로그래밍 가능
			"TRAIL"	커플링 형식 "커플 모션"
			"LEAD"	커플링 형식 "리딩값 커플링"
			"EG"	커플링 형식 "전자 변속기"
			"COUP"	커플링 형식 "동기 스핀들"

FAx: 추종축/스핀들

LAx: 선도 축/스핀들

참고

커플링 특성은 명시적으로 프로그래밍되지 않지만(동기 동작의 가공 프로그램에서) 기본 설정을 통해 적용됩니다.

기본 설정(CPSETTYPE="CP")이 아닌 키워드 설정 CPSETTYPE에 따라 프리셋 커플링 특성을 실행할 수 있습니다.

추가 정보

일반 커플링에 대한 자세한 사항은 다음을 참조하십시오.

- 기능 매뉴얼 축 및 스핀들

4.16.7 접선 제어

4.16.7.1 커플링 정의 (TANG)

사전 정의된 절차인 TANG(...)을 통해 회전 축 사이의 접선 커플링은 추종 축으로 정의되고, 두 개의 기하 축은 리딩 축으로 정의됩니다. 추종 축은 리딩 축의 경로 접선에 대해 지속적으로 정렬됩니다.

참고

커플링 팩터

1의 결합 팩터는 직접 프로그램하지 않아도 됩니다.
접선 축의 방향은 -1의 결합 인자를 사용하여 회전됩니다.

구문

```
TANG(<following axis>, <leading axis_1>, <leading axis_2>,
<coupling factor>, <coordinate system>, <optimization>)
```

의미

TANG(...):	접선 커플링 정의		
<추종 축>:	추종 축 (회전 축) 이름		
	데이터 유형:	AXIS	
	값 범위:	채널 축 이름	
<리딩 축_1> <리딩 축_2>:	리딩 축 (기하 축)의 이름 [1]		
	데이터 유형:	AXIS	
	값 범위:	기하 축 채널 이름	
<커플링 팩터>:	리딩 축의 경로 접선을 변경하기 위한 추종 축의 각도 변경에 대한 팩터 n: 각도 변경$_{추종\ 축}$ = 각도 변경$_{경로\ 접선}$ * n		
	데이터 유형:	REAL	
	디폴트 값:	1.0	
<좌표계>:	활성 좌표계 [2]		
	데이터 유형:	CHAR	
	값:	"B":	기본 좌표계 (디폴트 값)
		"W":	공작물 좌표계 (사용 안함)

<최적화>:	최적화 유형		
	데이터 유형:	CHAR	
	값:	"S":	표준 (디폴트 값) 회전 축의 동적 반응이 추적에 필요한 것보다 크다면 회전 축의 동적 반응은 리딩 축에 영향을 미치지 않습니다. 그리고 이 방법은 충분한 정확도를 가집니다. 회전 축의 동적 반응이 경로 접선 상의 변화를 따라갈 수 있을 정도로 충분히 크지 않다면, 회전 축의 방향은 정의되지 않은 라운딩 공차를 따라 목표 방향에서 벗어나게 됩니다.
		"P":	회전 축의 동적 반응은 리딩 축의 경로 계획에 반영됩니다. 이 목적을 위해 TANGON()을 이용한 접선 커플링의 활성화와 관련하여 두 개의 추가 파라미터가 명시되어야 합니다: • 라운딩 공차 • 각도 허용 오차 "커플링 활성화 (TANGON) (페이지 1076)" 섹션을 참조하십시오. **참고** 운동학적 변환의 경우 최적화 방식 "P"를 사용할 것을 권장합니다.
참고 디폴트 값은 명확하게 프로그래밍하지 않아도 됩니다.			

[1] **참고**

기계의 최초 위치를 기준으로 기계 좌표계 (MCS) 상에 프로그래밍된 경로를 따라 이동하는 기하 축은 접선 커플링을 위한 리딩 축으로서 반드시 사용되어야 합니다. 예를 들어, 회전 사이클 CYCLE800이 회전 헤드와 함께 밀링 기계에서 사용되는 경우, 사이클이 어떻게 구성되는가에 따라 보간이 WCS에서 (예: 기하 축 X 및 Y 에서) 수행되게 됩니다. 그러나 MCS에 프로그래밍된 경로를 따라 이동하는 리딩 축과 마찬가지로 접선 커플링은 기하 축과 함께 정의되어야 합니다. 이 목적을 위해서 기계의 **비회전** 조건 상의 기하 축은 반드시 리딩 축으로 사용되어야 합니다.

[2] **참고**

기본 좌표계 (BCS)는 MCS에 대해 회전하지 않아야 합니다. 예를 들어, BCS가 ROT 명령어 또는 회전 싸이클 CYCLE800을 통해 회전하는 경우, 접선 제어가 더 이상 정확하지 않게 됩니다.

4.16.7.2 중간 블록 생성 활성화 (TLIFT)

리딩 축의 프로그래밍된 경로 상의 어떤 위치에서의 추종 축의 접선 변화가 기계 데이터 MD37400 $MA_EPS_TLIFT_TANG_STEP에 파라미터화된 한계치를 초과하는 경우, 추가 경로 계획은 코너에 설정된 행동에 달려있습니다. 사전 정의된 절차 TLIFT(...)를 사용하지 않는 경우 TANG(...) (페이지 1072) 및 TANGON(...) (페이지 1076)과 관련하여 프로그래밍된 라운딩 행동에 맞게 경로는 이송됩니다.

중간 블록 생성 활성화

TLIFT(...)는 TANG(...) 이후에 프로그래밍되며, 제어기에 의해 자동으로 생성된 중간 블록은 코너가 전처리 중 감지되는 시점에서 삽입됩니다.

프로그램이 실행된 상태에서 중간 블록에 도달하면 리딩 축은 정지합니다. 중간 블록에서는 다음 블록의 경로 접선을 향한 최대 축 동력으로 추종 축이 회전합니다. 그러면 리딩 축은 프로그래밍된 경로에서 더 멀리 이송됩니다.

중간 블록 생성 비활성화

중간 블록 생성을 비활성화하려면 접선 커플링은 TANG(...)을 이용하여 반드시 다시 정의되어야 하지만, 이후 TLIFT(...)를 이용한 중간 블록 생성은 활성화되지 않아야 합니다.

구문

```
TLIFT(<추종 축>)
```

의미

TLIFT(...):	중간 블록 계산을 통한 코너 감지 활성화	
<추종 축>:	추종 축 (회전 축) 이름	
	데이터 유형:	AXIS
	값 범위:	채널 축 이름

추종 축의 회전 속도

경로 축

접선 커플링이 활성화되기 전에 추종 축이 경로 축으로서 이미 이송된 경우, 회전 움직임이 경로 축인 중간 블록에서 수행됩니다.

FGREF[<axis>]=0.001을 통해 기준 반경을 명시하면, 회전 움직임은 파라미터화된 최대 축 속도로 수행됩니다:

MD32000 $MA_MAX_AX_VELO[<추종 축>]

포지셔닝 축

접선 커플링이 활성화되기 전에 추종 축이 아직 경로 축으로서 이송되지 않은 경우, 회전은 위치 축인 중간 블록에서 수행됩니다.

회전 움직임은 파라미터화된 중간 축 속도로 수행됩니다:

MD32060 $MA_POS_AX_VELO[<추종 축>]

4.16.7.3 커플링 활성화 (TANGON)

사전 정의된 절차 TANGON(...)을 통해 이전에 TANG(...) (페이지 1072)을 사용하여 정의된 접선 커플링이 활성화됩니다. 그러고 나면 추종 축은 다음 이송 중 경로 접선에 대해 지속적으로 정렬됩니다.

추종 축의 각도

경로 접선에 대한 추종 축의 각도는 TANG(...)에 명시된 변환율, 기계 데이터 MD37402 $MA_TANG_OFFSET에서 파라미터화된 옵셋 각도, 그리고 추가적으로 적용되는 TANGON(...)을 위해 명시된 옵셋 각도에 의해 결정됩니다.

최적화 "P"

값 "P"가 접선 커플링 (TANG(...)) 정의에서 최적 파라미터로 명시된 경우, 커플링을 활성화할 때 파라미터 "라운딩 공차" 및 최적 파라미터 "각도 허용 오차"는 반드시 설정되어야 합니다.

각도 허용 오차로 값 0으로 명시된 경우, 파라미터 "라운딩 공차"만 활성화됩니다.

각도 허용 오차로 0보다 큰 값이 명시된 경우 활성 라운딩 공차는 파라미터화된 라운딩 공차의 최소값으로부터 나오고, 라운딩 공차는 파라미터화된 각도 허용 오차에 기초합니다.

추종 축의 동적 반응이 파라미터화된 조건들을 따를 만큼 충분하지 않은 경우, 리딩 축의 경로 속도는 이에 따라 감소합니다.

구문

```
TANGON(<추종 축>, <옵셋 각도>, <라운딩 공차>, <각도 허용 오차>)
```

의미

TANGON(...):	접선 커플링 활성화	
<추종 축>:	추종 축 (회전 축) 이름	
	데이터 유형:	AXIS
	값 범위:	채널 축 이름
<옵셋 각도>:	경로 접선에 대한 추종 축의 옵셋 각도 기준점은 회전 축의 영점입니다.	
	데이터 유형:	REAL
<라운딩 공차>:	최대 허용 라운딩 공차 라운딩 공차가 동적 조건에 따라 증가하면 리딩 축의 경로 속도는 감소합니다.	
	데이터 유형:	REAL
<각도 허용 오차>:	추종 축을 0으로 설정하는 것과 경로 접선 사이에 명시된 각도에 대한 최대 허용 오차	
	데이터 유형:	REAL

4.16.7.4 커플링 비활성화 (TANGOF)

사전 정의된 절차 TANGOF(...)를 통해 , TANG(...) (페이지 1072)으로 정의되고 TANGON(...) (페이지 1076)으로 활성화된 접선 커플링이 비활성화됩니다. 그러고 나면 추종 축은 리딩 축의 경로 접선에 대해 더 이상 정렬되지 않습니다. 그러나, 추종 축을 리딩 축에 커플링하는 것은 비활성화 이후에도 계속 유지되며, 이는 다음 기능을 방지하기 위한 것입니다. 예:

- 평면 변경
- 기하 축 변경
- 추종 축에 대한 새로운 접선 커플링 정의

리딩 축에 대한 추종 축의 커플링 연결을 최종적으로 취소하는 작업은 커플링이 TANGDEL(...) (페이지 1078)을 통해 삭제되어야 완료됩니다.

프로그래밍

```
TANGOF(<추종 축>)
```

의미

TANGOF(...):	접선 커플링 비활성화	
<추종 축>:	추종 축 (회전 축) 이름	
	데이터 유형:	AXIS
	값 범위:	채널 축 이름

4.16.7.5 커플링 삭제 (TANGDEL)

TANG(...) (페이지 1072)을 이용하여 정의된 접선 커플링은 TANGOF(...) (페이지 1077)를 이용한 접선 커플링이 비활성화된 이후에도 계속 유지됩니다. 그러면 현재의 접선 커플링은 다음 기능을 방지하기 위해 계속됩니다.

- 평면 변경
- 기하 축 변경
- 추종 축에 대한 새로운 접선 커플링 정의

사전 정의된 절차 TANGDEL(...)과 함께 현재의 접선 커플링은 TANGOF(...)를 이용하여 접선 커플링이 비활성화된 이후 삭제됩니다.

구문

```
TANGDEL(<추종 축>)
```

의미

TANGDEL(...):	TANG()을 이용하여 정의된 접선 커플링 삭제	
	적용:	넌모달
<추종 축>:	접선 커플링이 삭제된 추종 축의 이름	
	데이터 유형:	AXIS
	값 범위:	채널 축 이름

예

리딩 축 변화

새로운 접선 커플링이 추종 축을 위해 다른 리딩 축에 대해 정의되기 전에 현재의 접선 커플링은 반드시 삭제되어야 합니다.

프로그램 코드	주석
N10 **TANG**(A, X, Y,1)	; 추종 축 A를 위한 접선 커플링 정의: A에서 X 및 Y까지
N20 TANGON(A)	; 추종 축 A를 위한 접선 커플링 활성화
N30 X10 Y20	
...	
N80 TANGOF(A)	; 추종 축 A를 위한 접선 커플링 비활성화
N90 **TANGDEL**(A)	; 추종 축 A를 위한 접선 커플링 삭제
...	
N120 **TANG**(A, X, Z)	; 추종 축 A를 위한 새로운 접선 커플링 정의
N130 TANGON(A)	; 추종 축 A를 위한 새로운 접선 커플링 활성화
...	

기하 축 변경

현재의 커플링에 대해 기하 축 변경이 수행되기 전에 커플링을 반드시 먼저 삭제해야 합니다.

프로그램 코드	주석
N10 GEOAX(2,Y1)	; 두 번째 기하 축 = 기계 축 Y1
N20 TANG(A, X, Y)	; 추종 축 A를 위한 접선 커플링 정의
N30 TANGON(A, 90)	; 추종 축 A를 위한 접선 커플링 활성화
N40 G2 F8000 X0 Y0 I0 J50	; 모션 블록
N50 TANGOF(A)	; 추종 축 A를 위한 접선 커플링 비활성화
N60 **TANGDEL**(A)	; 추종 축 A를 위한 접선 커플링 삭제
N70 **GEOAX**(2, Y2)	; 두 번째 기하 축 = 기계 축 Y2
N80 TANG(A, X, Y)	; 추종 축 A를 위한 새로운 접선 커플링 정의
N90 TANGON(A, 90)	; 추종 축 A를 위한 새로운 접선 커플링 활성화
...	

4.16.8 마스터/슬레이브 커플링 (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

"마스터/슬레이브 커플링"을 통해 다음 작업을 수행할 수 있습니다.

- 축들이 정지 상태일 때 슬레이브 축을 마스터 축에 커플링.
- 속도를 제어하여 **회전** 스핀들의 커플링/커플링 해제.
- 다이나믹 구성.

참고

포지셔닝 모드

포지셔닝 모드인 축과 스핀들의 경우 정지 상태에 한하여 커플링을 체결 또는 해제할 수 있습니다.

구문

```
MASLON(<slave_1>,<slave_2>,...)
MASLOF(<slave_1>,<slave_2>,...)
MASLOFS(<slave_1>,<slave_2>,...)
```

다이나믹 구성:

```
MASLDEF(<slave_1>,<slave_2>, ... ,<master>)
MASLDEL(<slave_1>,<slave_2>,...)
```

의미

MASLON:	일시적인 마스터/슬레이브 커플링 활성화	
	<Slave_x>,...:	슬레이브 축 1 ... n
MASLOF:	활성 마스터/슬레이브 커플링 비활성화	
	<slave_1>,...:	슬레이브 축 1 ... n
MASLOFS:	마스터/슬레이브 커플링 해제 및 슬레이브 스핀들 자동 제동 ("속도 제어 모드에서 스핀들 커플링 동작" 참조)	
	<slave_1>,...:	슬레이브 축 1 ... n
MASLDEF:	가공 프로그램에서 마스터/슬레이브 그룹 생성/변경	
	<slave_1>,...:	슬레이브 축 1 ... n
	<master>:	Master axis
MASLDEL:	마스터/슬레이브 커플링 분리 및 그룹화 정의 삭제	
	<slave_1>,...:	슬레이브 축 1 ... n
	참고: 머신 데이터에 설정된 마스터/슬레이브 정의는 그대로 유지됩니다.	

참고

속도 제어 모드에서 스핀들 커플링 동작

속도 제어 모드에서 스핀들의 경우 MASLON, MASLOF, MASLOFS 및 MASLDEL의 커플링 동작을 다음 머신 데이터를 통해 명시적으로 지정합니다.

MD37263 $MA_MS_SPIND_COUPLING_MODE

기본 설정 (MD37263 = 0) 인 경우 관련 축이 정지 상태인 경우에 한하여 슬레이브 축을 커플링 및 커플링 해제합니다. MASLOFS는 MASLOF에 해당합니다.

MD37263 = 1인 경우 커플링 명령이 즉시 실행되고 따라서 모션이 시작됩니다. MASLON의 경우 커플링이 즉시 활성화되는 반면, MASLOFS나 MASLOF의 경우 커플링이 즉시 해제됩니다. MASLOF의 경우, 이 순간 회전 중이던 슬레이브 스핀들은 새 속도가 프로그래밍될 때까지 기존 속도를 유지합니다. 하지만 MASLOF 의 경우 스핀들에 자동으로 제동이 걸립니다.

참고

MASLOF/MASLOFS의 경우, 실질적인 사전 처리 중지가 필요하지 않습니다. 사전 처리 중지가 누락된 이유로 다음 프로그램이 진행될 때까지 슬레이브 축의 시스템 변수 $P에 의해 값 업데이트가 이뤄지지 않습니다.

참고

슬레이브 축의 경우 PRESETON을 사용하여 실제값을 마스터 축과 동일한 값으로 동기화할 수 있습니다. 이를 위해서는, 전원을 켰을 때 원점 복귀되지 않은 슬레이브 축의 실제값을 마스터/축의 값으로 설정할 수 있도록, 영구/슬레이브 커플링을 일시적으로 꺼야 합니다. 그런 다음 다시 영구 커플링을 구성합니다.

영구 마스터/슬레이브 커플링은 머신 데이터
MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1로 설정하여 활성화합니다.
이 설정은 임시 커플링을 위한 언어 명령에는 전혀 영향을 미치지 않습니다.

예

예 1: 마스터/슬레이브 커플링의 슬레이브 축에 실제 값 설정

영구 마스터/슬레이브 설정을 위해 PRESETON은 슬레이브 축의 실제 값을 마스터 축의 실제 값으로 설정합니다.

프로그램 코드	설명
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; 슬레이브 축의 영구 커플링 비활성화
NEWCONF	; 머신 데이터 변경 활성화
STOPRE	
MASLOF(Y1)	; 임시 커플링 비활성화
PRESETON(AX2,$VA_IM(M_AX))	; 슬레이브 축의 실제 값 = 마스터 축의 실제 값

프로그램 코드	설명
$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; 슬레이브 축의 영구 커플링 활성화
NEWCONF	; 머신 데이터 변경 활성화

예 2: 마스터/슬레이브 커플링 다이나믹 구성

축 컨테이너 회전 후 다른 스핀들과의 커플링을 활성화하려면 이전 커플링을 해제하고 구성을 삭제한 후 새 커플링을 구성해야 합니다.

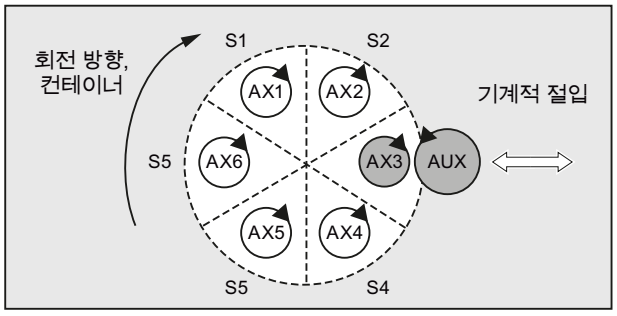


그림 4-15 축 컨테이너 회전 이전

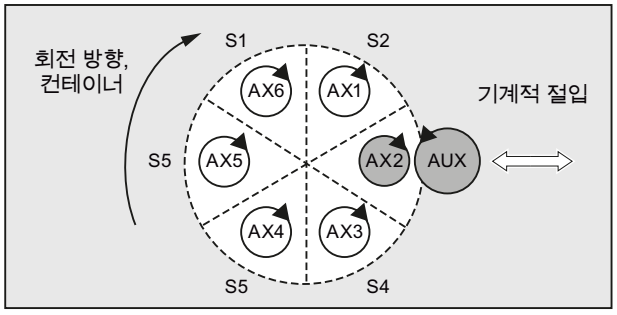


그림 4-16 한 개의 슬롯만큼 축 컨테이너 회전 이후

프로그램 코드	설명
MASLDEF(AUX,S3)	; AUX: 슬레이브, S3: 마스터 = AX3
MASLON(AUX)	; 커플링 ON
M3=3 S3=4000	; 마스터 회전
MASLDEL(AUX)	; 연결 해제 및 커플링 삭제
AXCTSWE(CT1)	; 축 컨테이너 회전 인에이블
MASLDEF(AUX,S3)	; AUX: 슬레이브, S3: 마스터 = AX2

추가 정보

- 기능 매뉴얼 기술
- 기능 매뉴얼 기본 기능

4.17 동기 동작

4.17.1 동기 동작 정의

동기 동작은 가공 프로그램 블록에 정의합니다. 동기 동작에 속하지 않는 명령들은 이 블록에 프로그래밍할 수 없습니다.

동기 동작은 다음과 같은 요소로 구성됩니다.

적용 대상, ID 번호 (옵션)	조건 부분 (옵션)			조건이 충족된 동작 부분			조건이 충족되지 않은 동작 부분 (옵션)		
	주파수	G 명령 (옵션)	조건	키워드	G 명령 (옵션)	동작	키워드	G 명령 (옵션)	동작
---[1] ID=<no.> IDS=<no.>	---[1] WHENEVER FROM WHEN EVERY	G...	논리적 표현식	DO	G...	동작 1 ... 동작 n	ELSE	G...	동작 1 ... 동작 n

[1] 프로그래밍되지 않음

구문

```
DO <action 1> ... <action n>
<frequency> [<G function>] <condition> DO <action 1> ... <action
n>
ID=<No> <frequency> [<G function>] <condition> DO <action 1> ...
<action n>
IDS=<No> <frequency> [<G function>] <condition> DO <action 1> ...
<action n>
IDS=<no.> <frequency> [<G function>] <condition> DO <action
1...n> ELSE <action 1...n>
```

추가 정보

기능 매뉴얼 동기 동작

4.18 오실레이션

4.18.1 비동기 오실레이션 (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

오실레이션 축은 오실레이션이 비활성화될 때까지 정의된 피드로 두 개의 반전 지점 1과 2 사이에서 앞/뒤로 이동합니다.

오실레이션 도중 다른 축들은 원하는 대로 보간할 수 있습니다. 포지셔닝 축 혹은 경로 이송을 통해 연속 절입이 이루어질 수 있지만, 오실레이션 이송 및 절입 이송 간에는 **아무런 관계가 없습니다**.

비동기 오실레이션 속성

- 비동기 오실레이션은 블록 제한을 초과하는 특정 축에서 작동합니다.
- 가공 프로그램을 통해 오실레이션 이동이 블록 방향으로 활성화됩니다.
- 몇 개 축을 결합하여 보간하거나 오실레이션 경로를 이중화할 수는 없습니다.

프로그래밍

다음 명령은 가공 프로그램에서 비동기 오실레이션을 활성화 및 제어하는 데 사용할 수 있습니다.

프로그램 값은 기본 실행 중에 블록 동기를 통해 해당 셋팅 데이터에 입력되며 다시 변경할 때까지 활성화 상태로 유지됩니다.

구문

```
OSP1[<axis>]=<value> OSP2[<axis>]=<value>
OST1[<axis>]=<value> OST2[<axis>]=<value>
FA[<axis>]=<value>
OSCTRL[<axis>]=(<setting option>,<reset option>)
OSNSC[<axis>]=<value>
OSE[<axis>]=<value>
OSB[<axis>]=<value>
OS[<axis>] = 1
OS[<axis>] = 0
```

의미

<축>:	오실레이션 축 이름		
OS:	오실레이션 활성화/비활성화		
	값:	1	오실레이션 **on**
		0	오실레이션 **off**

OSP1:	반전 지점 1 위치 정의		
OSP2:	반전 지점 2 위치 정의 **참고:** 증분 이동이 활성화되면 위치는 NC 프로그램에서 마지막으로 프로그래밍된 해당 반전 위치까지 증분식으로 계산됩니다.		
OST1:	[s]의 반전 위치 1에서 정지 시간 정의		
OST2:	[s]의 반전 지점 2에서 정지 시간 정의		
	<value>:	-2	정위치 정지를 기다리지 않고 보간 계속
		-1	일반 정위치 정지 대기
		0	미세 정위치 정지 대기
		>0	미세 정위치 정지를 대기하고 나서 지정된 정지 시간 동안 대기 **참고:** 정지 시간의 단위는 G4로 프로그래밍한 정지 시간의 단위와 동일합니다.
FA:	이송 속도 정의 이송 속도는 포지셔닝 축에 정의된 속도입니다. 이송 속도가 정의되지 않은 경우에는 머신 데이터에 저장된 값이 적용됩니다.		

OSCTRL:	설정 옵션 및 리셋 옵션 지정 옵션 값 0 - 3은 비활성화 상태에서 반전 지점의 동작을 암호화합니다. 0 - 3 사이의 변용 중 하나를 선택할 수 있습니다. 나머지 설정은 선택한 변용에 따라 원하는 대로 조합할 수 있습니다. + 기호를 사용하여 여러 옵션을 추가할 수 있습니다.		
	<value>:	0	오실레이션 비활성화 상태에서 다음 반전 지점에서 정지 (표준) **참고:** 값 1과 2가 리셋된 경우에만 허용됩니다.
		1	오실레이션이 비활성화된 경우 반전 지점 1에서 정지합니다.
		2	오실레이션이 비활성화된 경우 반전 지점 2에서 정지합니다.
		3	오실레이션이 비활성화된 경우 스파크아웃 스트로크가 프로그램되지 않은 경우 반전 지점에 접근하지 않습니다.
		4	스파크아웃 이후에 끝점에 접근합니다.
		8	이동할 거리를 삭제하여 오실레이션을 취소한 경우에는 스파크아웃 스트로크를 실행하고 필요하다면 끝점까지 접근해야 합니다.
		16	이동할 거리를 삭제하여 오실레이션을 취소한 경우에는 차단할 때와 같이 해당 반전 지점에 접근해야 합니다.
		32	새 피드는 다음 반전 지점 이후에만 활성화됩니다.
		64	FA가 0입니다. FA = 0: 경로 오버레이가 활성화됩니다. FA가 0이 아닙니다. FA <> 0: 속도 오버레이가 활성화됩니다.
		128	로터리 축 DC (최단 경로) 의 경우
		256	스파크아웃 스트로크는 이중 스트로크입니다 (표준). 1=단일 스트로크
		512	1차 접근 시작 위치
OSNSC:	스파크아웃 스트로크 개수 정의		

OSE:	오실레이션 비활성화 후 접근할 종료 위치 정의 (공작물 좌표계에서) **참고:** "OSE"를 프로그래밍할 때 옵션 4가 "OSCTRL"에 암시적으로 적용됩니다.
OSB:	오실레이션 활성화 전 접근할 시작 위치 정의 (공작물 좌표계에서) 시작 위치에는 반전 지점 1 이전에 접근합니다. 시작 위치가 반전 지점 1과 일치하는 경우에는 다음으로 반전 지점 2에 접근하게 됩니다. 시작 위치에 도달한 후에는 이 위치가 반전 지점 1과 일치하더라도 정지 시간이 적용되지 않습니다. 그 대신 축이 미세 정위치 정지 신호를 기다립니다. 설정된 정위치 정지 조건이 모두 충족되어야 합니다. **참고:** 시작 위치에 대한 접근을 초기화하도록 셋팅 데이터 SD43770 $SA_OSCILL_CTRL_MASK의 Bit 9를 설정해야 합니다.

예

예 1: 두 반전 지점 간의 오실레이션을 위한 오실레이션 축

오실레이션 축 Z은 위치 10에서 100 사이를 오실레이션 할 것입니다. 반전 지점 1은 미세 정위치 정지를 통해서, 반전 지점 2는 일반 정위치 정지를 통해서 접근합니다. 오실레이션 축에 대한 이송 속도는 250이 되어야 합니다. 3개의 스파크아웃 스트로크를 가공 동작 끝에서 실행하고 오실레이션은 종료 위치 200에 접근해야 합니다. 절입 축의 이송 속도는 1이어야 하며 X 방향의 절입 끝은 위치 15에 도달해야 합니다.

프로그램 코드	코멘트
WAITP(X,Y,Z)	; 초기 설정.
G0 X100 Y100 Z100	; 포지셔닝 축 작업으로 전환.
WAITP(X,Z)	
OSP1[Z]=10 OSP2[Z]=100	; 반전 지점 1, 반전 지점 2.
OSE[Z]=200	; 종료 위치.
OST1[Z]=0 OST2[Z]=-1	; U1 에서 정지 시간: 미세 정위치 정지
	; U2 에서 정지 시간: 일반 정위치 정지
FA[Z]=250 FA[X]=1	; 오실레이션 축 피드, 절입 축
OSCTRL[Z]=(4,0)	; 설정 옵션.
OSNSC[Z]=3	; 3개의 스파크아웃 스트로크
OS[Z]=1	; 오실레이션 시작.
WHEN $A_IN[3]==TRUE DO DELDTG(X)	; 이동할 거리 삭제.
POS[X]=15	; X축 시작 위치.
POS[X]=50	; X축 종료 위치.
OS[Z]=0	; 오실레이션 정지.
M30	

참고

"OSP1[Z]=..."~"OSNCS[Z]=..." 명령 시퀀스를 하나의 블록에 프로그래밍할 수도 있습니다.

예 2: 반전 지점을 온라인으로 수정하여 오실레이션

비동기 오실레이션에 필요한 셋팅 데이터를 가공 프로그램에서 설정할 수 있습니다.

셋팅 데이터가 프로그램에 직접 기술된 경우 전처리 과정에서 변경 사항이 적용됩니다. 전처리 정지 (STOPRE) 에 의해 동기화 반응이 이루어질 수 있습니다.

프로그램 코드	코멘트
$SA_OSCILL_REVERSE_POS1[Z]=-10	
$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0	; 오실레이션 축의 실제 값이 반전 지점을 초과하면 절입 축이 정지됩니다.
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; 오실레이션 활성화.
OS[Z]=0	; 오실레이션 비활성화.
M30	

추가 정보

오실레이션 축

다음은 오실레이션 축에 적용됩니다.

- 모든 축을 오실레이션 축으로 사용할 수 있습니다.
- 몇 개 오실레이션 축을 동시에 활성화할 수 있습니다 (최대: 포지셔닝 축 개수).
- 직선 보간 G1은 오실레이션 축에 대해 항상 활성 상태이며, 프로그램에서 현재 사용하는 G 명령과는 관계가 없습니다.

오실레이션 축은 다음과 같이 작동합니다.

- 다이나믹 변환을 위한 입력 축의 역할을 합니다.
- 갠트리와 커플 모션 축의 가이드 축 역할을 합니다.
- 다음 조건에서 이송될 수 있습니다.
  - 저크 제한 안함 "BRISK"
    또는
  - 저크 제한 사용 "SOFT"
    또는
  - 굴곡부를 포지셔닝 축으로 하는 가속 커브 사용

오실레이션 반점 지점

오실레이션 위치를 정의할 때 현재 옵셋을 적용해야 합니다.

- 절대 지정
  "OSP1[Z]=<값>"
  반점 지점 위치 = 옵셋 + 프로그램된 값의 합계
- 상대 지정
  "OSP1[Z]=IC(<값>)"
  반전 지점 위치 = 반전 지점 1 + 프로그램된 값

예제:

프로그램 코드

```
N10 OSP1[Z]=100 OSP2[Z]=110
...
N40 OSP1[Z]=IC(3)
```

WAITP

기하 축에 오실레이션을 수행하는 경우에는 "WAITP"로 이 축에 오실레이션을 활성화해야 합니다.

오실레이션이 완료되면 "WAITP"로 오실레이션 축을 다시 포지셔닝 축으로 입력하여 정상적으로 사용할 수 있습니다.

모션 동기 동작 및 정지 시간을 통한 오실레이션

설정된 정지 시간이 만료되고 나서 (축을 기준으로 한 새로운 이동 거리로 지정된) 오실레이션이 진행되는 동안 내부 블록이 변경됩니다. 블록이 변경될 때 비활성화 기능이 검사됩니다. 비활성화 기능은 모션 순서에 대한 제어 설정 (OSCTRL) 에 따라 정의됩니다. *이러한 다이나믹 응답은 피드 오버라이드의 영향을 받을 수 있습니다.*

스파크아웃 스트로크가 시작되거나 끝점에 접근하기 전에 오실레이션 스트로크가 실행됩니다. *비활성화 응답이 변경된 것처럼 보이더라도, 이 예제는 그 경우에 해당하지 않습니다.*

4.18.2 동기 동작에 의한 오실레이션 제어 (OSCILL)

이 오실레이션 모드를 사용할 경우, 반전 지점이나 정의된 반전 범위 내에서만 절입 동작이 실행됩니다.

요구사항에 따라 오실레이션 모션이 다음과 같이 진행될 수 있습니다.

- 계속 또는
- 절입 실행이 끝날 때까지 중단

구문

1. **오실레이션 관련 파라미터 정의**
2. **동기화 동작 정의**
3. **축 할당 및 절입 정의**

의미

OSP1[<oscillating axis>]=	반전 지점 위치 1
OSP2[<oscillating axis>]=	반전 지점 위치 2
OST1[<oscillating axis>]=	반전 지점 1에서의 정지 시간 (초)
OST2[<oscillating axis>]=	반전 지점 2에서의 정지 시간 (초)
FA[<oscillating axis>]=	오실레이션 축 피드
OSCTRL[<oscillating axis>]=	설정 또는 리셋 옵션
OSNSC[<oscillating axis>]=	스파크아웃 스트로크 개수

OSE[<oscillating axis>]=	종료 위치
WAITP(<oscillating axis>)	오실레이션 축 허용

축 할당 및 절입

```
OSCILL[<oscillating axis>]=(<infeed axis 1>,<infeed
axis 2>,<infeed axis 3>)
POSP[<infeed axis>]=(<end position>,<partial length>,<mode>)
```

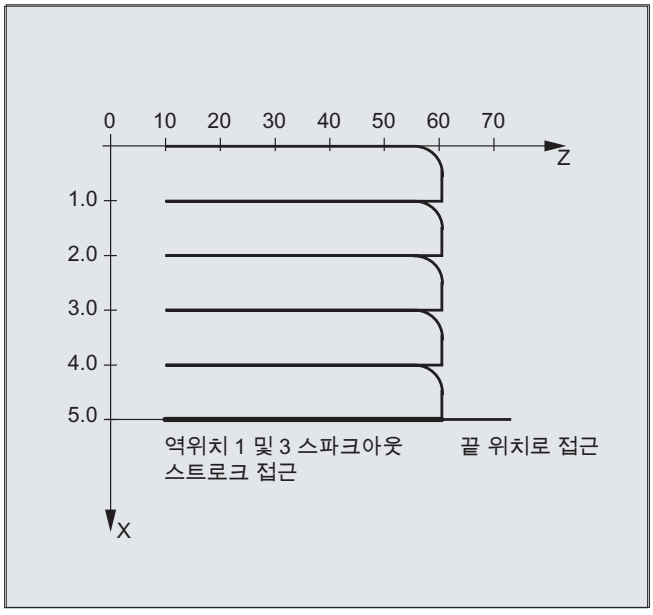
OSCILL:	오실레이션 축에 대해 절입 축 할당
POSP:	완전 절입 및 부분 절입 정의 ("파일 및 프로그램 관리" 절 참조)
End position:	모든 부분 절입 이후의 절입 축 관련 종점이 이송됩니다.
Partial length:	반전 지점/반전 범위의 부분 절입 길이
Mode:	완전 절입을 부분 절입으로 분할 = 크기가 동일한 2개의 잔여 단계 (디폴트); = 크기가 동일한 모든 부분 절입

동기화 동작

WHEN… … DO	when..., do...
WHENEVER … DO	whenever..., do...

예

반전 지점 1에서는 절입이 발생하면 안됩니다. 반전 지점 2에서는 절입이 ii2 거리에서 반전 지점 2이전에 시작되고, 오실레이션 축은 반전 지점에서 부분 절입의 종료를 기다리지 않습니다. Z 축이 오실레이션 축이고 X 축이 절입 축입니다.



1. 오실레이션 파라미터

프로그램 코드	코멘트
DEF INT ii2	; 반전 영역 2에 변수 정의
OSP1[Z]=10 OSP2[Z]=60	; 반전 지점 1 및 2 정의
OST1[Z]=0 OST2[Z]=0	; 반전 지점 1: 미세 정위치 정지 반전 지점 2: 미세 정위치 정지
FA[Z]=150 FA[X]=0.5	; 오실레이션 축 Z 이송 속도, 절입 축 X 이송 속도
OSCTRL[Z]=(2+8+16.1)	; 반전 지점 2에서 오실레이팅 모션 비활성화: DTG 삭제 후 스파크아웃 및 종료 위치 접근, DTG 삭제 후 반전 지점 접근
OSNC[Z]=3	; 스파크아웃 스트로크
OSE[Z]=70	; 종료 위치 = 70
ii2=2	; 반전 지점 범위 설정
WAITP(Z)	; Z축에 오실레이션 인에이블

2. 동기 동작

프로그램 코드	코멘트
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z] DO -> $AA_OVR[X]=0 $AC_MARKER[0]=0	; MCS에서 오실레이션 축 Z의 실제 위치가 반전 범위 2의 시작 지점보다 작은 경우, 절입 축 X의 축 오버라이드가 항상 0%로 설정되고 인덱스 0인 비트 메모리가 값 0으로 설정됩니다.

프로그램 코드	코멘트
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0	; MCS에서 오실레이션 축 Z의 실제 위치가 반전 범위 2보다 큰 경우, 오실레이션 축 Z의 축 오버라이드가 항상 0%로 설정됩니다
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[0]=1	; 부분 절입의 남은 이동 거리가 0이면 인덱스 0인 비트 메모리를 값 1로 설정합니다.
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100	; 인덱스 0인 비트 메모리가 1인 경우 절입 축 X의 축 오버라이드를 0%로 설정합니다. 이렇게 하면 정상보다 이른 절입, 즉 오실레이션 축 Z가 여전히 좌측 반전 영역 2에 있는 상태지만 절입 축 X가 새로운 절입을 준비하는 상태가 되는 것이 방지됩니다. Z축을 이동시키기 위해 오실레이션 축 Z의 축 오버라이드를 0% (두 번째 동기 동작 실행) 에서 다시 100%로 설정하십시오.

-> 반드시 싱글 블록으로 프로그래밍돼야 합니다.

3. 오실레이션 시작

프로그램 코드	코멘트
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; 축 시작 오실레이션 축 Z가 X 축에 절입 축으로 할당됩니다. X 축이 종점 5까지 1 단계로 이동해야 합니다.
M30	; 프로그램 종료

추가 정보

1. **오실레이션 파라미터 정의**
   오실레이션 관련 파라미터는 절입 축 및 오실레이션 축이 포함된 이동 블록과 절입 정의에 앞서 정의되어야 합니다('비동기식 오실레이션' 참조).
2. **모션 동기 동작 정의**
   다음과 같은 동기 조건을 정의할 수 있습니다.
   오실레이션 축이 반전 범위(ii1, ii2) 이내 또는 반전 지점(U1, U2)에 위치할 때까지 **절입 억제**.

   반전 지점에서 절입이 이뤄지는 동안 오실레이션 모션 정지.
   부분 절입 완료 시 오실레이션 동작 재시작.
   다음 부분 절입 시작 정의.
3. **오실레이션 축 및 절입 축** 외에도 **부분 절입 및 완전 절입** 정의.

오실레이션 파라미터 정의

오실레이션 축 및 절입 축 할당: OSCILL

```
OSCILL[<oscillating axis>]=(<infeed axis1>,<infeed
axis2>,<infeed axis3>)
```

축 할당 및 오실레이션 동작 시작은 "OSCILL" 명령으로 정의합니다.

오실레이션 축에 최대 3개의 절입 축을 할당할 수 있습니다.

참고

오실레이션을 시작하기에 앞서 축의 특성에 대한 동기화 조건을 정의해야 합니다.

절입 정의: POSP

```
POSP[<infeed axis>]=(<end position>,<partial length>,<mode>)
```

"POSP" 명령으로 다음과 같은 작업을 제어 시스템에 정의합니다.

- (끝점을 기준으로 한) 완전 절입
- 반전 지점이나 반전 범위의 부분 절입 길이
- (끝점을 기준으로 해서) 끝점에 도달했을 때 부분 절입 응답

Mode = 0	최종 2개의 부분 절입에 대한 대상 지점까지의 이동할 거리는 2개의 동등한 단계로 분할됩니다(기본 설정).
Mode = 1	크기가 동일한 모든 부분 절입 완전 절입을 기점으로 해서 계산됩니다.

동기 동작 정의

아래 열거된 동기 동작이 일반적인 오실레이션에 사용됩니다.

아래 주어진 개별 작업에 대한 해결 사례를 사용자 전용 오실레이션 동작용 모듈로 활용할 수 있습니다.

참고

동기화 조건은 개인적인 경우에 따라 다르게 프로그래밍할 수 있습니다.

키워드

`WHEN … DO …`	when..., do...
`WHENEVER … DO`	whenever..., do...

기능

아래 설명된 언어 리소스를 이용하여 다음과 같은 기능을 구현할 수 있습니다.

1. 반전 지점에서 절입
2. 반전 범위에서 절입

3. 두 반전 지점에서 절입
4. 반전 지점에서 오실레이션 동작 중지
5. 오실레이션 동작 재시작
6. 너무 이른 부분 절입 시작 금지

여기에 제시된 모든 동기 동작 사례는 다음 조건을 전제로 합니다.

- 반전 지점 1 < 반전 지점 2
- Z = 오실레이션 축
- X = 절입 축

참고

자세한 내용은 '동기 동작' 단락을 참조하십시오.

오실레이션 축, 절입 축, 부분 절입 및 완전 절입 정의

반전 지점 범위에서 절입

반드시 반전 지점에 도달하기 전에 반전 범위 내에서 절입 동작이 시작돼야 합니다.

이 동기 동작으로 인해 오실레이션 축이 반전 범위 내에 도달할 때까지 절입 동작이 금지됩니다.

위와 같은 가정에 따라 다음 명령을 사용합니다.

반전 범위 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X] = 0
```

MCS에서 오실레이션 축의 실제 위치가 반전 범위 1보다 큰 경우, 절입 축의 축 오버라이드가 0%로 설정됩니다.

반전 범위 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X] = 0
```

MCS에서 오실레이션 축의 실제 위치가 반전 범위 2 시작 위치보다 작은 경우, 절입 축의 축 오버라이드가 0%로 설정됩니다.

반전 지점에서 절입

오실레이션 축이 반전 지점에 도달하기 전에는 절입 축이 이동하지 않습니다.

위와 같은 가정에 따라 다음 명령을 사용합니다.

반전 범위 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS에서 오실레이션 축 Z의 실제 위치가 위치 반전 지점 1보다 크거나 작을 때마다 절입 축 X의 축 오버라이드가 0%로 설정되고 오실레이션 축 Z의 축 오버라이드가 100%로 설정됩니다.

반전 범위 2:

반점 지점 2의 경우:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS에서 오실레이션 축 Z의 실제 위치가 위치 반전 지점 2보다 크거나 작을 때마다 절입 축 X의 축 오버라이드가 0%로 설정되고 오실레이션 축 Z의 축 오버라이드가 100%로 설정됩니다.

반전 지점에서 오실레이션 동작 중지

반전 지점에서 오실레이션 축이 정지하고 그와 동시에 절입 동작이 시작됩니다. 절입 동작이 완료되면 오실레이션 모션이 계속됩니다.

그와 동시에, 아직 활성 상태인 이전 동기 동작에 의해 오실레이션 축이 정지한 경우 이 동기 동작을 이용하여 절입 동작을 시작할 수 있습니다.

위와 같은 가정에 따라 다음 명령을 사용합니다.

반전 범위 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS에서 오실레이션 축의 실제 위치가 위치 반전 지점 1과 동일할 때마다 오실레이션 축의 축 오버라이드가 0%로 설정되고 절입 축의 축 오버라이드가 100%로 설정됩니다.

반전 범위 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS에서 오실레이션 축 Z의 실제 위치가 위치 반전 지점 2와 동일할 때마다 오실레이션 축 X의 축 오버라이드가 0%로 설정되고 절입 축의 축 오버라이드가 100%로 설정됩니다.

반전 지점 온라인 검사

비교 우측에 $$로 코드화된 주 실행 변수가 있는 경우, IPO 싸이클에서 두 변수를 검사하여 서로 지속적으로 비교합니다.

참고

자세한 내용은 '동기 동작' 단락을 참조하십시오.

오실레이션 동작 재시작

이 동기 동작의 목적은 부분 절입 동작이 완료된 순간에 오실레이션 축이 계속 이동하는 것입니다.

위와 같은 가정에 따라 다음 명령을 사용합니다.

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]= 100
```

WCS에서 절입 축 X의 부분 절입과 관련한 남은 거리가 0인 경우, 오실레이션 축의 축 오버라이드가 항상 100%로 설정됩니다.

다음 부분 절입

절입이 완료됐을 때 다음 부분 절입의 너무 이른 시작이 반드시 제한돼야 합니다.

채널 관련 마커($AC_MARKER[Index])가 이 용도로 사용됩니다. 채널 관련 마커는 부분 절입(부분 절입의 이동할 거리≡ 0) 끝에 실행되고 축이 반점 범위를 벗어났을 때 삭제됩니다. 그리고 나서 동기 동작에 의해 다음 절입 동작이 제한됩니다.

주어진 전제를 토대로 반전 지점 1에 대해 다음과 같은 명령이 적용됩니다.

1. 마커 설정:

```
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[1]=1
```

WCS에서 절입 축 X의 부분 절입과 관련한 남은 거리가 0일 때마다 색인 1에 속한 비트 메모리가 항상 1의 값으로 설정됩니다.

2. 마커 삭제

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1] = 0
```

MCS에서 오실레이션 축 Z의 실제 위치가 위치 반전 지점 1의 위치보다 크거나 작을 때마다 비트 메모리 1이 0으로 설정됩니다.

3. 절입 금지

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

비트 메모리 1이 동일할 때마다 절입 축 X의 축 오버라이드가 0%로 설정됩니다.

4.19 펀칭 및 니블링

4.19.1 활성화/비활성화

4.19.1.1 펀칭 및 니블링 활성화/비활성화(SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

펀칭 및 니블링 활성화/비활성화

PON 및 SON은 펀칭 및 니블링 기능을 활성화하는 데 사용됩니다. SPOF는 모든 펀칭 및 니블링 관련 기능을 종료합니다. 모달 명령인 PON 및 SON은 상호 배타적입니다. 즉, PON이 활성화되면 SON이 비활성화됩니다.

리더를 사용한 펀칭 및 니블링

SONS 및 PONS 기능은 펀칭 또는 니블링 기능을 활성화하는 데도 사용됩니다.

보간 레벨에서 스트로크를 제어하는 SON/PON과는 반대로, 이 기능을 사용하면 스트로크 초기화에 대한 신호 관련 제어가 서보 레벨에서 이루어집니다. 이것은 스트로크 주파수가 더 커지게 하므로 펀칭 기능이 향상됩니다.

리더에서 신호가 평가되는 동안에는 니블링 또는 펀칭 축에서 위치를 변경할 수 있도록 하는 모든 기능(예: 수동 핸드휠 이송, PLC를 통한 프레임 변경, 측정 기능)이 금지됩니다.

지연 펀칭

PDELAYON은 펀칭 스트로크의 출력 지연을 유도합니다. 모달로 적용되는 명령에는 준비 기능이 있으며, 따라서 일반적으로 PON 이전에 위치합니다. 정상적인 펀칭은 PDELAYOF 이후 일반 펀칭이 재개됩니다.

참고

지연 시간은 셋팅 데이터 SD42400 $SC_PUNCH_DWELLTIME에서 설정됩니다.

경로 종속 가속

PUNCHACC는 구멍 간격에 따라 서로 다른 가속도를 정의하는 가속 특성을 지정하는 데 사용할 수 있습니다.

2차 펀칭 인터페이스

2차 펀칭 인터페이스 (보조 펀칭 유닛 또는 호환 가능한 매체) 를 사용해야 하는 장비는 제어 시스템의 2차 고속 디지털 입출력 쌍 (I/O 쌍) 으로 전환할 수도 있습니다. 전체 펀칭/니블링

기능을 양쪽 인터페이스에서 사용할 수 있습니다. SPIF1 및 SPIF2 명령은 1차 펀칭 인터페이스와 2차 펀칭 인터페이스를 전환하는 데 사용됩니다.

참고

요구사항: 머신 데이터에서 펀칭 기능에 대한 2차 I/O 페어를 정의해야 합니다 (장비 제조업체 사양 참조).

구문

```
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
```

의미

PON:	펀칭 활성화.	
SON:	니블링 활성화	
PONS:	리더를 통한 펀칭 ON.	
SONS:	리더를 통한 니블링 ON.	
SPOF:	펀칭/니블링을 해제	
PDELAYON:	지연을 포함한 펀칭 ON.	
PDELAYOF:	지연을 포함한 펀칭 비활성화.	
PUNCHACC:	이송에 따른 가속을 활성화합니다. 파라미터:	
	<Smin>	최소 홀 간격
	<Amin>	초기 가속 <Amin>이 <Amax>보다 클 수도 있습니다.
	<Smax>	최대 홀 간격
	<Amax>	최종 가속 <Amax>가 <Amin>보다 클 수도 있습니다.

SPIF1:	**1차** 펀칭 인터페이스를 활성화합니다. 스트로크는 1차 고속 I/O 페어를 사용하여 제어됩니다.
SPIF2:	**2차** 펀칭 인터페이스를 활성화합니다. 스트로크는 2차 고속 I/O 페어를 사용하여 제어됩니다.
	참고: 1차 펀칭 인터페이스는 항상 리셋 또는 제어 시스템 기동 후에 활성화됩니다. 하나의 펀칭 인터페이스만 사용될 경우에는 프로그래밍할 필요가 없습니다.

예

예 1: 니블링 활성화

프로그램 코드	코멘트
...	
N70 X50 SPOF	; 펀칭 시작이 없는 위치.
N80 X100 SON	; 니블링 활성화, 모션 (X=50) 이전 및 프로그래밍된 이동 (X=100) 완료 시 스트로크 시작.
...	

예 2: 지연 펀칭

프로그램 코드	코멘트
...	
N170 PDELAYON X100 SPOF	; 펀칭 시작이 없는 위치, 지연된 펀칭 시작 활성화.
N180 X800 PON	; 펀칭 활성화. 펀칭 스트로크는 종료 위치에 도달할 때 지연과 함께 출력됩니다.
N190 PDELAYOF X700	; 지연 펀칭 비활성화, 프로그래밍된 이동 완료 시 정상 펀칭 시작.
...	

예 3: 2개의 펀칭 인터페이스를 사용한 펀칭

프로그램 코드	코멘트
...	
N170 SPIF1 X100 PON	; 블록 종료 시 스트로크는 첫 번째 고속 출력에서 시작됩니다. '스트로크 적용' 신호가 첫 번째 입력에서 모니터링됩니다.
N180 X800 SPIF2	; 두 번째 스트로크는 두 번째 고속 출력에서 시작됩니다. '스트로크 적용' 신호가 두 번째 입력에서 모니터링됩니다.
N190 SPIF1 X700	; 이후 모든 스트로크는 첫 번째 인터페이스에서 제어합니다.
...	

추가 정보

리더가 있는 펀칭 및 니블링 (PONS/SONS)

리더가 있는 펀칭 및 니블링은 둘 이상의 채널에서 동시에 활성화될 수 없습니다. PONS 또는 SONS는 한 번에 하나의 채널에서만 활성화될 수 있습니다.

경로 종속 가속(PUNCHACC)

예:
```
PUNCHACC(2,50,10,100)
```

2mm 이하의 심공 간격:

축은 최대 가속의 50%에 해당하는 속도에서 가속화됩니다.

2 mm ~ 10 mm의 심공 간격:

가속이 간격에 비례하여 100%까지 증가됩니다.

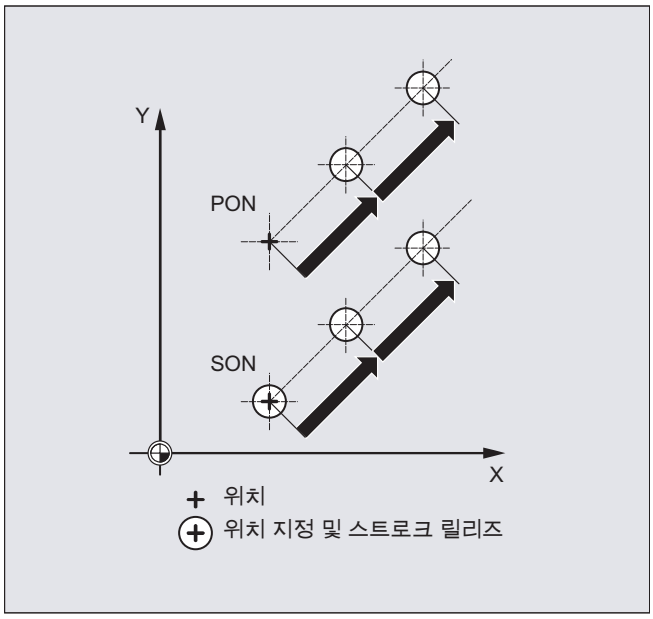
10 mm 이상의 심공 간격:

이송 가속도가 최대 가속도의 100%입니다.

최초 스트로크 시작

기능 활성화 이후에 최초 스트로크가 시작되는 순간은 니블링 또는 펀칭 선택 여부에 따라 달라집니다.

- PON/PONS:
  - 활성화 이후 첫 번째 블록의 스트로크를 포함하여 모든 스트로크를 블록 끝에서 실행합니다.
- SON/SONS:
  - 니블링 기능 활성화 이후 첫 번째 스트로크가 블록 시작 시점에 실행됩니다.
  - 이후의 모든 스트로크는 블록 끝에서 시작됩니다.



즉석 펀칭 니블링

블록에 펀칭 축이나 니블링 축 (활성 평면의 축) 에 대한 이송 정보가 포함된 경우에 한하여 스트로크가 시작됩니다.

그러나 동일한 위치에서 스트로크를 시작하려면 펀칭/니블링 축 중 하나를 이송 경로 0으로 프로그래밍하면 됩니다.

회전 가능한 공구를 이용한 가공

참고

접선에 프로그래밍된 경로까지 회전 가능한 공구의 위치를 지정하려면 접선 제어 기능을 사용하십시오.

M 명령 사용

이전 버전과 마찬가지로 매크로를 통해 언어 명령 대신 특수 M 코드를 사용할 수 있습니다 (호환성). 이전 버전에서 사용된 M 코드 및 이에 해당하는 언어 명령은 다음과 같습니다.

```
M20, M23  ≙  SPOF
M22       ≙  SON
M25       ≙  PON
M26       ≙  PDELAYON
```

매크로 파일 예제:

프로그램 코드	코멘트
DEFINE M25 AS PON	; 펀칭 ON
DEFINE M125 AS PONS	; 리더 (leader) 를 이용한 펀칭 ON
DEFINE M22 AS SON	; 니블링 ON
DEFINE M122 AS SONS	; 리더를 이용한 니블링 ON
DEFINE M26 AS PDELAYON	; 지연 펀칭 ON
DEFINE M20 AS SPOF	; 펀칭, 니블링 OFF
DEFINE M23 AS SPOF	; 펀칭, 니블링 OFF

프로그래밍 예제:

프로그램 코드	코멘트
...	
N100 X100 M20	; 펀칭 시작이 없는 위치.
N110 X120 M22	; 니블링 활성화, 모션 전후에 스트로크 시작.
N120 X150 Y150 M25	; 펀칭 활성화, 모션 종료 시 스트로크 시작
...	

4.19.2 자동 경로 분할

경로 구간으로 분할

펀칭 또는 니블링이 활성화되면 SPP 및 SPN이 경로 축에 프로그래밍된 전체 이송 범위를 동일한 길이의 여러 경로 세그먼트로 분할할 수 있습니다. 내부적으로 각 경로 구간은 하나의 블록에 해당합니다.

스트로크 횟수

펀칭의 경우 첫 번째 경로 세그먼트의 종점에서 첫 번째 스트로크가 실행됩니다. 반면 니블링의 경우 첫 번째 경로 세그먼트의 시작점에서 스트로크가 실행됩니다. 따라서 전체 이송 구간에서 다음과 같은 횟수가 산정됩니다.

펀칭: 스크로크 횟수 = 경로 구간 개수

니블링: 스크로크 횟수 = 경로 구간 개수 + 1

보조 기능

생성된 블록 중 첫 번째 블록에서 보조 기능이 실행됩니다.

구문

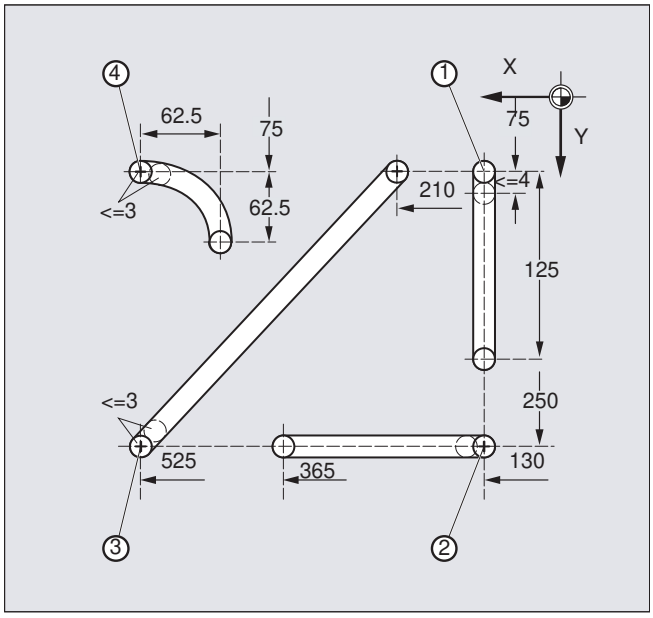
```
SPP=
SPN=
```

의미

SPP:	경로 구간의 크기(스크로크간 최대 거리), 모달
SPN:	블록당 경로 구간 개수, 모달식 활성화

예 1

프로그래밍된 니블링 구간이 자동으로 경로 구간으로 분할돼야 합니다.

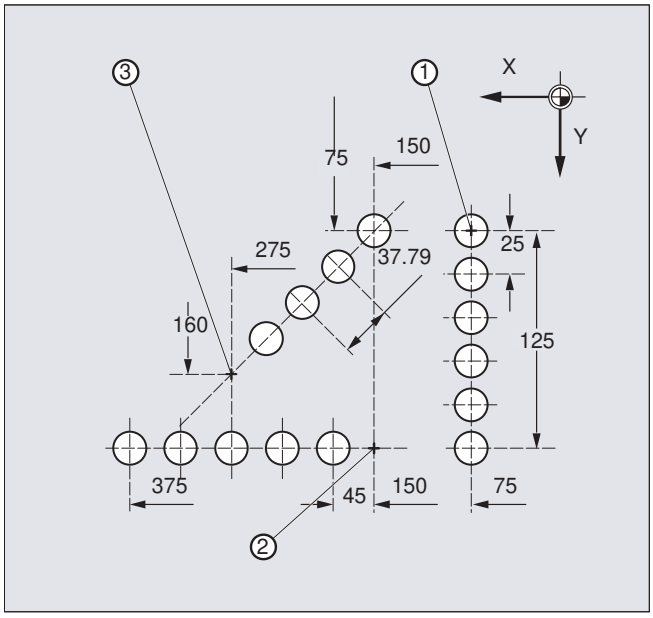


프로그램 코드	코멘트
N100 G90 X130 Y75 F60 SPOF	; 시작점 1에 포지셔닝
N110 G91 Y125 SPP=4 SON	; 니블링 ON, 자동 경로 분할 시 최대 경로 세그먼트 길이: 4 mm
N120 G90 Y250 SPOF	; 니블링 OFF, 시작점 2에 포지셔닝
N130 X365 SON	; 니블링 ON, 자동 경로 분할 시 최대 경로 세그먼트 길이: 4 mm
N140 X525 SPOF	; 니블링 OFF, 시작점 3에 포지셔닝

프로그램 코드	코멘트
N150 X210 Y75 SPP=3 SON	; 니블링 ON, 자동 경로 분할 시 최대 경로 세그먼트 길이: 3 mm
N160 X525 SPOF	; 니블링 OFF, 시작점 4에 포지셔닝
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; 니블링 ON, 자동 경로 분할 시 최대 경로 세그먼트 길이: 3 mm
N180 G00 G90 Y300 SPOF	; 니블링 OFF

예 2

개별적인 일련의 심공을 대상으로 자동 경로 분할이 이뤄져야 합니다. 분할을 위해 최대 경로 구간 길이(SPP 값)가 지정됩니다.



프로그램 코드	코멘트
N100 G90 X75 Y75 F60 PON	; 시작점 1에 포지셔닝, 싱글 홀 펀칭
N110 G91 Y125 SPP=25	; 자동 경로 분할 시 최대 경로 세그먼트 길이: 25 mm
N120 G90 X150 SPOF	; 펀칭 OFF, 시작점 2에 포지셔닝
N130 X375 SPP=45 PON	; 펀칭 ON, 자동 경로 분할 시 최대 경로 세그먼트 길이: 45 mm
N140 X275 Y160 SPOF	; 펀칭 OFF, 시작점 3에 포지셔닝
N150 X150 Y75 SPP=40 PON	; 펀칭 ON, 프로그래밍된 경로 구간 세그먼트 40 mm 대신 계산된 경로 세그먼트 길이 37.79 mm 사용
N160 G00 Y300 SPOF	; 펀칭 OFF, 포지셔닝

4.19.2.1 경로 축을 대상으로 경로 분할

SPP 경로 구간 길이

SPP 는 전체 이송 거리가 분할된 스트로크간 최대 거리와 최대 경로 구간 길이를 지정하는 데 사용됩니다. 이 명령은 SPOF 또는 SPP=0으로 활성화할 수 있습니다.

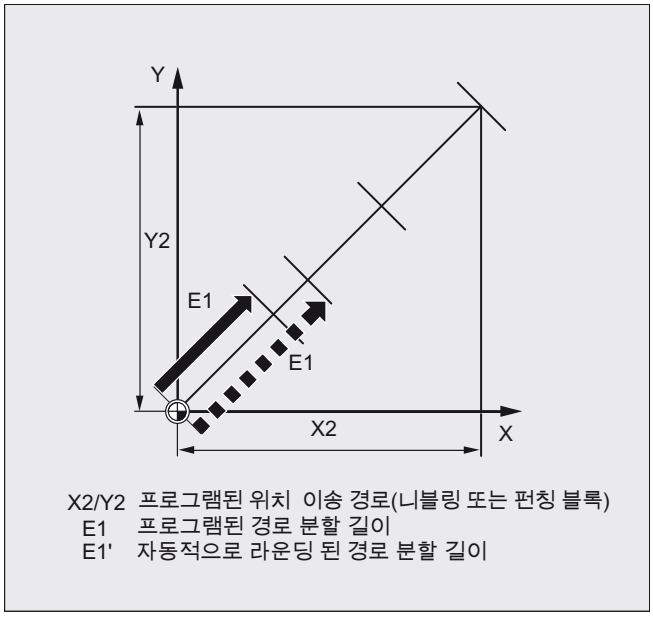
예:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

10 mm의 전체 이송 길이가 2 mm의 경로 구간 5개로 분할됩니다(SPP=2).

참고

SPP에 의해 활성화된 경로 구간은 항상 등거리(모든 구간의 길이가 동일)가 됩니다. 다시 말해서, 프로그래밍된 경로 구간 크기(SPP 설정)는 전체 이송 거리의 등거리와 SPP 값이 정수인 경우에만 유효합니다. 정수가 아닌 경우, 정수 비율을 산출하기 위해 경로 구간 크기가 내부적으로 감소됩니다.



X2/Y2 프로그램된 위치 이송 경로(니블링 또는 펀칭 블록)
E1 프로그램된 경로 분할 길이
E1' 자동적으로 라운딩 된 경로 분할 길이

예:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

전체 이송 거리가 15 mm이고 경로 구간 길이가 3.5 mm인 경우, 분할 비율은 정수값이 아닙니다(4.28). 이 경우, 산출 가능한 다음 정수 비율로 SPP 값이 감소됩니다. 이 예제에서 경로 구간 길이는 3 mm가 됩니다.

SPN 경로 구간 개수

SPN를 사용하여 전체 이송 거리에서 산출될 경로 구간 개수를 정의할 수 있습니다. 구간 길이는 자동으로 산정됩니다. SPN는 비모달식이기 때문에 펀칭이나 니블링이 PON 또는 SON으로 미리 활성화돼야 합니다.

4.19.2.2 단일 축을 대상으로 경로 분할

경로 축 왜에 단일 축이 펀칭/니블링 축으로 정의된 경우, 단일 축을 대상으로 자동 경로 분할 기능을 활성화할 수 있습니다.

SPP를 위한 단일 축의 동작

프로그래밍된 경로 구간 길이(SPP)는 기본적으로 경로 축을 참조합니다. 따라서 단일 축 이동 및 SPP 값에 더하여 경로 축이 프로그램 되지 않은 블록에서는 SPP 값이 무시됩니다.

단일 축과 경로 축이 함께 블록에 프로그래밍된 경우, 단일 축의 동작은 적합한 기계 데이터의 설정에 달려있습니다.

1. 디폴트 설정
   단일 축에 의해 이송되는 경로는 SPP 에 의해 생성된 중간 블록 사이에 균등하게 분할됩니다.

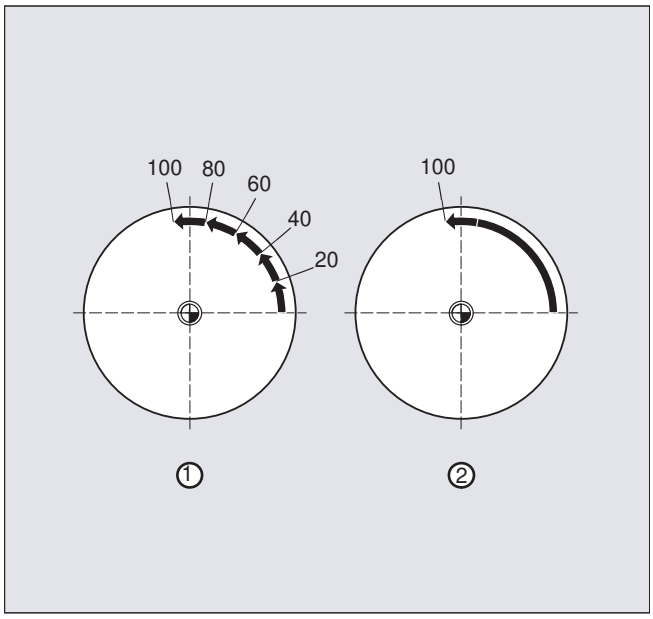
예:

```
N10 G1 SON X10 A0

N20 SPP=3 X25 A100
```

3mm의 스트로크 거리의 결과로서 15mm의 X축(경로 축)의 총 이동 거리에 대해 5개 블록이 생성됩니다.

따라서 A축은 모든 블록에서 20°로 회전합니다.



1. 경로 분할이 이뤄지지 않은 단일 축
   생성된 블록 중 첫 번째 블록의 전체 거리를 단일 축이 이송합니다.
2. 다른 경로 분할
   단일 축의 동작은 다음과 같은 경로 축의 보간에 따라 다릅니다.

- 원호 보간: 경로 분할
- 선형 보간: 경로 분할 없음

SPN을 위한 동작

경로 축이 동시에 프로그램 되지 않는다면 경로 세그먼트의 프로그램 된 숫자 역시 적용됩니다.

사전조건: 단일 축이 펀칭/니블링 축으로 정의됩니다.

4.20 연삭

4.20.1 연삭 전용 공구 모니터링 활성화/비활성화 (TMON, TMOF)

사전 정의된 절차 TMON(...) 및 TMOF(...)를 통해 연삭 전용 공구 모니터링이 활성화되거나 비활성화됩니다 (기하 및 속도 모니터링).

요구사항

공구 전용 파라미터 $TC_TPG1 to $TC_TPG9를 반드시 설정해야 합니다.

구문

```
TMON(<TNo>)
...
TMOF(<TNo>)
```

의미

TMON(...): 	연삭 전용 공구 모니터링 활성화 명령어는 반드시 연삭 전용 공구 모니터링이 활성화되는 채널에 프로그래밍되어야 합니다.
TMOF(...): 	연삭 전용 공구 모니터링 비활성화 명령어는 반드시 연삭 전용 공구 모니터링이 비활성화되는 채널에 프로그래밍되어야 합니다.
<TNo>:	T 번호 **참고:** 현재 사용되고 있는 활성 연삭 휠이 아니라 비활성 연삭 휠에 대해 모니터링이 켜지거나 꺼지는 경우에만 필요함.
TMOF(0):	모든 공구 모니터링 비활성화

4.21 확장 정지 및 후퇴 (ESR)

ESR로도 불리는 확장 정지 및 후퇴 기능을 사용하여 에러가 발생했을 때 프로세스의 일부로 에러에 유연하게 대응할 수 있습니다.

- **확장 정지**
  에러 상황이 허용하는 경우 확장 정지에 인에이블된 모든 축을 지정된 순서대로 정지합니다.
- **후퇴**
  현재 사용 중인 공구를 최대한 신속하게 공작물에서 후퇴시킵니다.
- **일반 작업 (SINAMICS 드라이브 기능 "Vdc 제어")**
  입력 전압 차단 등의 이유로 DC 링크 전압이 설정 값 이하로 떨어지면 이 용도(발전기 역할)로 사용되는 드라이브의 제동 에너지를 재생하여 후퇴에 필요한 전력을 생산합니다.

트리거 소스

일반적 소스 (NC-외부/전역 또는 모드 그룹/채널별):

- 디지털 입력(예: NCU 모듈) 또는 다시 읽어올 수 있는 제어 시스템 내부의 디지털 출력 이미지($A_IN, $A_OUT)
- 채널 상태 ($AC_STAT)
- VDI 신호 ($A_DBB)
- 여러 알람의 그룹 메시지 ($AC_ALARM_STAT)

축 소스

- 추종 축의 비상 후퇴 임계값 (전자 커플링 동기, $VA_EG_SYNCDIFF[<following axis>])
- 드라이브: DC 링크 경고 임계값 (부족전압 위험), $AA_ESR_STAT[<axis>]
- 드라이브: 발전기 최소 속도 임계값 (재생 회전 에너지 사용 불가), $AA_ESR_STAT[<axis>].

스태틱 동기 동작의 게이팅 논리: 소스/반응 링크

스태틱 동기 동작의 유연한 게이팅 기능을 활용하여 소스에 따라 특정 반응을 신속하게 트리거합니다.

스태틱 동기 동작을 사용하여 관련 소스를 모두 연결하는 작업은 사용자의 책임입니다. 소스 시스템 변수 전체를 평가하거나 비트 마스크를 사용하여 평가하고 원하는 반응을 논리 연산으로 만들 수 있습니다. 스태틱 동기 동작은 모든 조작 모드에서 적용됩니다.

활성화

기능 인에이블

발전기 작동, 셧다운 및 후퇴 기능은 제어 신호 $AA_ESR_ENABLE를 설정하여 실행할 수 있습니다. 이 제어 신호는 동기 동작으로 변경할 수 있습니다.

기능 트리거

ESR은 시스템 변수 $AC_ESR_TRIGGER를 설정하여 인에이블된 모든 축에서 함께 트리거됩니다.

DC 링크 부족전압 위험이 감지되면 드라이브에서 발전기 작동이 "자동"으로 활성화됩니다.

NC와 드라이브 사이에 통신 에러가 감지되거나 드라이브에서 DC 링크 부족전압이 감지 (구성 및 인에이블 필요) 되면 드라이브와 관련 없이 정지 및/또는 후퇴가 실행됩니다.

또한 제어 신호 $AN_ESR_TRIGGER (모든 드라이브에 명령 발송) 를 설정한 경우에도 드라이브와 관련 없이 NC에서 정지 및/또는 후퇴를 실행할 수 있습니다.

추가 정보

기능 매뉴얼 축 및 스핀들

4.21.1 NC 제어 ESR

4.21.1.1 NC 제어 후퇴 (POLF, POLFA, POLFMASK, POLFMLIN)

NC 제어 후퇴를 위해서는 몇 가지 조건을 충족해야 합니다 ("NC 제어 후퇴 (POLF, POLFA, POLFMASK, POLFMLIN) (페이지 1111) 참조"). 이 조건이 충족되는 경우에만 시스템 변수 $AC_ESR_TRIGGER (또는 단일 축인 경우 $AA_ESR_TRIGGER) 를 설정하여 채널의 후퇴 축에 구성된 고속 도피 (LIFTFAST) 를 활성화할 수 있습니다.

구문

```
POLF(<axis>)=<position>
POLFA(<axis>,<type>,<position>)
POLFMASK(<axis_1>,<axis_2>,...)
POLFMLIN(<axis_1>,<axis_2>,...)
```

```
POLFA에는 다음과 같은 단축형을 사용할 수 있습니다.
POLFA(<axis>,<type>) ; 단일 축 후퇴를 위한 단축형
POLFA(axis,0/1/2) ; 즉시 비활성화 또는 활성화
POLFA(axis,0,$AA_POLFA[axis]) ; 전처리 정지 유도
```

```
POLFA(axis,0) ; 전처리 정지 안함
```

의미

POLF:	후퇴 축의 대상 위치를 지정하기 위한 어드레스 POLF는 모달입니다.			
	<axis>:	후퇴시킬 기하 축 또는 채널/기계 축 이름		
	<position>:	후퇴 위치		
		유형:	REAL	
		기하 축에는 WCS가 사용되고 그 외 축에는 MCS가 사용됩니다. 기하 축과 채널/기계 축의 이름이 동일한 경우 후퇴는 WCS에서 진행됩니다.		
POLFA:	단일 축의 후퇴 위치 지정을 위해 사전 정의된 서브프로그램 호출			
	<axis>:	채널 축 이름		
	<type>:	위치 지정 모드		
		유형:	INT	
		값:	0:	잘못된 위치 값 표시
			1:	위치 값은 절대값입니다.
			2:	위치 값은 증분치(거리)입니다.
		참고: 축이 단일 축이 아닌 경우, 유형이 누락된 경우, 또는 유형이 0인 경우 해당 알람이 출력됩니다.		
	<위치>:	후퇴 위치 (위의 설명 참조)		
		참고: 유형 = 0으로 지정된 위치 값도 허용됩니다. 단, 이 값은 유효하지 않은 값으로 표시되고 후퇴를 실행하려면 다시 프로그래밍해야 합니다.		
POLFMASK:	고속 도피 이송 이후 **다른 축과 관계 없이** 후퇴시킬 축을 선택할 수 있도록 사전 정의된 서브프로그램 호출.			
	<axis_1>,…:	고속 도피 중에 POLF명령으로 정의된 위치까지 이송해야 할 축의 이름. 지정된 모든 축은 동일한 좌표계에 존재해야 합니다.		
	POLFMASK()에 축을 지정하지 않으면 다른 축과 관계 없이 후퇴하도록 설정된 모든 축에서 고속 도피가 비활성화됩니다.			

POLFMLIN:	고속 도피 이송 이후 **직선으로** 후퇴시킬 축을 선택할 수 있도록 사전 정의된 서브프로그램 호출.	
	<axis_1>,…:	위의 설명 참조.
	POLFMLIN()에 축을 지정하지 않으면 직선으로 후퇴하도록 설정된 모든 축에서 고속 도피가 비활성화됩니다.	

참고

POLFMASK 또는 POLFMLIN를 통해 고정 위치로 고속 후퇴를 인에이블하려면 POLF를 사용하여 선택한 축에 위치를 프로그래밍해야 합니다.

참고

POLFMASK, POLFMLIN나 POLFMLIN, POLFMASK에 의해 차례대로 축이 활성화된다면, 마지막 정의는 항상 특정 축에 적용됩니다.

참고

POLF로 프로그램된 위치와 POLFMASK나 POLFMLIN에 의한 활성화는 가공 프로그램이 시작되면 삭제됩니다. 이는 사용자가 각 가공 프로그램에서 POLF 값과 POLFMASK 나 POLFMLIN에서 선택된 축을 다시 프로그래밍해야 함을 의미합니다.

참고

단축형 POLFA를 사용할 때, 유형만 변경 가능하다면, 사용자는 후진 위치나 후진 경로에 실행가능하고 그 값의 양식이 맞는지 확인해야 합니다. 특히, 전원을 켠 후, 후진 위치와 후진 경로를 다시 설정합니다.

예

개별 축의 후진:

프로그램 코드	코멘트
MD37500 $MA_ESR_REACTION[AX1]=21	; NC 제어 후퇴.
...	
$AA_ESR_ENABLE[AX1] = 1	
POLFA(AX1,1,20.0)	; 축1에 축 후퇴 위치 20.0 (절대치) 을 지정합니다.
$AA_ESR_TRIGGER[AX1] = 1	; 여기서 후퇴를 시작합니다.

추가 정보

NC 제어 후퇴를 위한 요구 사항

- NC 제어 후퇴를 위해서는 채널에 후퇴 축을 구성해야 합니다.
  MD37500 $MA_ESR_REACTION = 21
- 이 축에 ESR이 인에이블되어 있어야 합니다.
  $AA_ESR_ENABLE = 1
- 지연 시간을 정의해야 합니다.
  MD21380 $MC_ESR_DELAY_TIME1
  MD21381 $MC_ESR_DELAY_TIME2
- POLF를 사용하여 가공 프로그램에 축 후퇴 위치가 설정되어 있어야 합니다.
- POLFMASK/POLFMLIN을 사용하여 NC 제어 후퇴를 위한 축을 선택해야 합니다.
- 후퇴 이동을 위한 활성화 신호를 설정하고 계속 설정된 상태여야 합니다.

NC 제어 반응의 활성화와 시작

시스템 변수 $AC_ESR_TRIGGER = 1로 설정되어 있고 채널에 후퇴 축이 구성 (MD37500 $MA_ESR_REACTION = 21) 되고 이 축에 $AA_ESR_ENABLE = 1이 설정된 경우, 이 채널에서 고속 도피 (LIFTFAST) 가 활성화됩니다.

POLFMASK 또는 POLFMLIN으로 선택한 축에 POLF (또는 POLFA) 로 도피 이동을 설정하면 가공 프로그램에서 이 축에 정의한 경로 모션이 이 도피 이동으로 대체됩니다.

MD21380 $MC_ESR_DELAY_TIME1과 MD21381 $MC_ESR_DELAY_TIME2를 합한 시간이 후퇴에 사용할 수 있는 최대 시간이 됩니다. 시간이 만료되면, 후진 축을 향하여 추적 급감속이 시작됩니다.

참고

확장 후퇴 ($AC_ESR_TRIGGER에 의해 트리거된 LIFTFAST/LFPOS) 는 **중단시킬 수 없습니다**. 확장 후퇴는 비상 정지를 통해서만 중간에 정지시킬 수 있습니다.

참고

이 경우 다중 후퇴를 방지하기 위해 $AC_ESR_TRIGGER에 의해 시작된 후퇴를 잠급니다.

단일 축 후퇴

단일 축 후퇴를 위해서는 POLFA로 단일 축의 후퇴 위치를 프로그래밍하고 다음 조건을 충족해야 합니다.

- $AA_ESR_ENABLE = 1
- 트리거 시점 ($A**A**A_ESR_TRIGGER = 1) 에 <Axis>이 단일 축이어야 합니다.
- <Type>이 1 또는 2여야 합니다.

고속 도피 중 후퇴 방향

빠른 상승이 활성화되는 시점에서 프레임 유효를 고려해야 합니다.

참고

회전 상태의 프레임은 POLF를 통해 상승 방향에 영향을 줍니다.

축 교체

후진 축들은 하나의 NC 채널에 정확하게 할당 되어야 하며, 채널 간에 교환 할 수 없습니다. 다른 축에서 후퇴 축을 교체하려고 시도하면 알람이 출력됩니다. $AA_ESR_ENABLE[AX] = 0으로 설정하여 이 축을 다시 비활성화해야만 새 채널에서 축을 교체할 수 있습니다. 축이 교체되면 $AA_ESR_ENABLE[AX] = 1로 설정하여 축을 다시 활성화할 수 있습니다.

중립 축

중립 축들은 NC 제어 ESR을 착수할 수 없습니다.

4.21.1.2 NC 제어 정지

시스템 변수 $AC_ESR_TRIGGER (또는 단일 축의 경우 $AA_ESR_TRIGGER) 를 설정하여 채널 축에 구성된 정지 축에 NC 제어 정지를 활성화할 수 있습니다.

요구 사항

- 채널에 NC 제어 정지를 위한 정지 축이 설정되어 있어야 합니다.
  MD37500 $MA_ESR_REACTION = 22
- 이 축에 ESR이 인에이블되어 있어야 합니다.
  $AA_ESR_ENABLE = 1
- 지연 시간을 정의해야 합니다.
  MD21380 $MC_ESR_DELAY_TIME1 (지연 시간, ESR 축)
  MD21381 $MC_ESR_DELAY_TIME2 (보간 제동을 위한 ESR 시간)

실행

이 축은 MD21380에 설정된 시간 동안 프로그래밍된 대로 보간을 계속합니다. MD21380에 지정된 시간 지연이 만료된 후, 제어 제동(램프 정지)이 시작됩니다: MD21381에 정의된 시간이 보간 제어 제동에 허용되는 최대 시간입니다. 이 시간이 경과하면 후속 추적에 따라 고속 제동이 시작됩니다.

예

단일 축 정지:

프로그램 코드	코멘트
`MD37500 $MC_ESR_REACTION[AX1] = 22`	`; NC 제어 정지.`
`MD21380 $MC_ESR_DELAY_TIME1[AX1] = 0.3`	
`MD21381 $MC_ESR_DELAY_TIME2[AX1] = 0.06`	
`...`	
`$AA_ESR_ENABLE[AX1] = 1`	
`$AA_ESR_TRIGGER[AX1] = 1`	`; 여기서 정지 시작.`

4.21.2 드라이브 통합 ESR

4.21.2.1 드라이브 통합 정지 (ESRS) 설정

드라이브 통합 ESR 기능의 "정지"를 위한 드라이브 파라미터는 `ESRS(...)` 기능을 사용하여 설정합니다.

구문

```
ESRS(<access_1>,<stopping time_1>[,...,<axis_n>,<stopping time_n>])
```

의미

ESRS(...):	ESR 기능 "정지"를 위한 드라이브 파라미터 쓰기 기능. 기능: • 블록에 단독으로 있어야 합니다. • 전처리 정지를 트리거합니다. • 동기 동작에 사용할 수 없습니다.	
<axis_1>, ..., <axis_n>:	드라이브 통합 정지를 설정해야 하는 축. 이 축의 경우에는 드라이브 파라미터 p0888 (설정) 이 드라이브에 기록됩니다. p0888 = 1	
	유형:	AXIS
	값 범위:	채널 축 이름
<stopping time_1>, ..., <stopping time_n>:	에러가 발생한 후 드라이브가 실제 속도 지령치로 계속 이동하는 시간. 지정된 축의 경우에는 드라이브 파라미터 p0892 (타이머) 가 드라이브에 기록됩니다. p0892 = <정지 시간>	
	단위:	s
	유형:	REAL
	값 범위:	0.00 - 20.00
기능 호출에 최대 5개 축을 프로그래밍할 수 있습니다 (n = 5).		

4.21.2.2 드라이브 통합 후퇴 (ESRS) 설정

드라이브 통합 ESR 기능의 "후퇴"를 위한 드라이브 파라미터는 ESRR(...) 기능을 사용하여 설정합니다.

구문

```
ESRR(<axis_1>,<retraction distance_1>,<retraction
velocity_1>[,...,<axis_n>,<retraction distance_n>,<retraction
velocity_n>])
```

의미

`ESRR(...):`	ESR 기능 "후퇴"를 위한 드라이브 파라미터 쓰기 기능. 기능: • 블록에 단독으로 있어야 합니다. • 전처리 정지를 트리거합니다. • 동기 동작에 사용할 수 없습니다.	

`<axis_1>,`
`...,`
`<axis_n>:`
	드라이브 통합 후퇴를 설정해야 하는 축.
	이 축의 경우에는 드라이브 파라미터 p0888 (설정) 이 드라이브에 기록됩니다.
	p0888 = 2	

	유형:	AXIS
	값 범위:	채널 축 이름

`<retraction distance_1>,`
`...,`
`<retraction distance_n>:`
	드라이브의 경우 후퇴 거리는 후퇴 속도로 변환됩니다. 지정된 축의 경우에는 드라이브 파라미터 p0893 (속도) 에 값이 기록됩니다.
	p0893 = (<retraction distance _n> converted into retraction speed)	

	단위:	mm/min, inch/min, degrees/min (축의 단위에 따라)
	유형:	REAL
	값 범위:	MIN - MAX

`<retraction velocity_1>,`
`...,`
`<retraction velocity_n>:`
	드라이브의 경우 후퇴 속도는 시간으로 변환됩니다. 지정된 축의 경우에는 드라이브 파라미터 p0892 (타이머) [s]에 값이 기록됩니다.
	p0892 = <retraction distance_n> / <retraction velocity _n>	

	단위:	mm/min, inch/min, degrees/min (축의 단위에 따라)
	유형:	REAL
	값 범위:	0.00 - MAX
기능 호출에 최대 5개 축을 프로그래밍할 수 있습니다 (n = 5).		

4.22 프로그램 실행 시간/ 가공 카운터

프로그램 런타임의 정보와 공구 카운터가 기계 공구 연산자를 지원하기 위해 제공됩니다.

본 정보는 NC 그리고/또는 PLC 프로그램에서 시스템 변수로 처리 될 수 있습니다. 본 정보를 작업자 인터페이스에 표시 할 수 있습니다.

4.22.1 프로그램 실행 시간

"프로그램 실행 시간" 기능은 NC 및 채널 관련 시스템 변수를 통해 가공 프로그램 및 동기 동작을 읽을 수 있는 공정을 체크하는 내부 NC 타이머를 제공합니다.

실행 시간 측정 ($AC_PROG_NET_TIME_TRIGGER) 용 트리거는 쓰기 가능하고 측정 프로그램 섹션에서 선택적으로 사용 되는 기능 시스템 변수입니다. 이는 $AC_PROG_NET_TIME_TRIGGER의 NC 프로그램 내 쓰기를 함으로써 시간 측정이 작동되고 다시 작동 불능 상태가 된다는 것을 의미합니다.

시스템 변수	의미	활동
NC 관련		
$AN_SETUP_TIME	기본 값 ('콜드 리스타트') 으로 마지막 제어기를 작동한 후 경과한 시간 (분). 기본 값으로 제어기를 가동할 때마다 자동으로 '0'으로 리셋됩니다.	• 항상 활성화
$AN_POWERON_TIME	제어기를 정상적으로 마지막 가동 ('웜 리스타트') 한 후 경과한 시간 (분). 정상적으로 제어기를 가동할 때마다 자동으로 '0'으로 리셋됩니다.	
채널별		

시스템 변수	의미	활동
$AC_OPERATING_TIME	자동 모드에서 NC 프로그램의 총 런타임 (초). 제어기를 가동할 때마다 자동으로 '0'으로 리셋됩니다.	• MD27860을 통해 활성화됨 • 자동 모드에서만
$AC_CYCLE_TIME	선택한 NC 프로그램의 런타임 (초). 새 NC 프로그램을 시작할 때마다 자동으로 '0'으로 리셋됩니다.	
$AC_CUTTING_TIME	가공 시간 (초) 경로 축 (적어도 하나가 활성 상태일 때) 의 런타임은 급 이송이 활성 상태가 아닐 때 프로그램의 NC 시작 및 종료/NC 리셋 사이에 있는 모든 NC 프로그램에서 측정됩니다. 드웰 시간이 동작 중일 때 측정이 중단됩니다. 기본 값으로 제어기를 가동할 때마다 자동으로 '0'으로 리셋됩니다.	
$AC_ACT_PROG_NET_TIME	현재 NC 프로그램의 실제 순수 런타임 (초). 새 NC 프로그램이 시작되면 자동으로 '0'으로 리셋됩니다.	• 항상 활성화 • 자동 모드에서만
$AC_OLD_PROG_NET_TIME	M30으로 정확히 종료된 프로그램의 순수 런타임 (초 단위)	
$AC_OLD_PROG_NET_TIME_COUNT	$AC_OLD_PROG_NET_TIME로 변경 POWER ON 후, $AC_OLD_PROG_NET_TIME_COUNT의 값은 "0" 입니다. 제어가 $AC_OLD_PROG_NET_TIME에 새로운 기록을 남기면 $AC_OLD_PROG_NET_TIME_COUNT 값은 항상 증가합니다.	

시스템 변수	의미		활동
$AC_PROG_NET_TIME_TRIGGER	실행 시간 측정을 위한 트리거		• 자동 모드에서만
	0	중립 상태 트리거가 비활성 상태입니다.	
	1	종료 측정 및 복사를 끝내고, $AC_ACT_PROG_NET_TIME into $AC_OLD_PROG_NET_TIME.$AC_ACT_PROG_NET_TIME이 "0"으로 설정되고 나면, 계속 작동합니다.	
	2	시작 측정을 시작하고 그렇게 하면서 $AC_ACT_PROG_NET_TIME to "0". $AC_OLD_PROG_NET_TIME이 변경되지 않도록 설정합니다.	
	3	중지 측정을 중지 합니다. 재개 될 때까지 $AC_OLD_PROG_NET_TIME 값을 변경하지 않고 $AC_ACT_PROG_NET_TIME 값을 유지합니다.	
	4	재개 측정이 재개됩니다. 즉 이전에 중단된 측정이 지속됩니다. $AC_ACT_PROG_NET_TIME이 계속됩니다. $AC_OLD_PROG_NET_TIME은 변하지 않습니다.	
POWER ON 시, 전체 시스템 변수는 0 으로 리셋됩니다.			

참고

장비 제조업체

머신 데이터 MD27860 $MC_PROCESSTIMER_MODE는 활성화 가능한 타이머를 동작시키기 위해 사용됩니다.

특정 기능 (예: GOTOS, 오버라이드 = 0%, 활성 시운전 이송, 프로그램 테스트, ASUB, PROG_EVENT 등) 의 활성 시간 측정 동작은 머신 데이터 MD27850 $MC_PROG_NET_TIMER_MODE 및 MD27860 $MC_PROCESSTIMER_MODE를 사용하여 설정합니다.

추가 정보: 기능 매뉴얼 기본 기능

참고

공작물의 잔여 시간

같은 공작물이 차례로 가공되는 경우, 다음 타이머 값을 사용하여 공작물에 대해 남아 있는 잔여 시간이 결정됩니다.

- 마지막으로 생산된 공작물의 가공 시간 ($AC_OLD_PROG_NET_TIME 참조)
- 현재 가공 시간 ($AC_ACT_PROG_NET_TIME 참조)

현재 가공 시간과 함께 잔여 시간이 사용자 인터페이스에 표시됩니다.

참고

STOPRE 사용

시스템 변수 $AC_OLD_PROG_NET_TIME 및 $AC_OLD_PROG_NET_TIME_COUNT는 암시적인 전처리 정지를 발생하지 않습니다. 이전 실행 프로그램에서 시스템 변수들의 값을 가져 온다면 가공 프로그램에서 사용 할 수 있습니다. 그러나, 실행 시간 측정 ($AC_PROG_NET_TIME_TRIGGER) 을 위한 트리거가 빈번하게 쓰여지고 결과로 $AC_OLD_PROG_NET_TIME 값이 빈번하게 변경된다면, 명시적 STOPRE는 가공 프로그램에서 사용해야 합니다.

보충 조건

- **블록 탐색**
  블록 탐색을 통해서는 프로그램 런타임이 확인되지 않습니다.
- **REPOS**
  REPOS 프로세스 시간이 현재 가공 시간 ($AC_ACT_PROG_NET_TIME) 에 추가됩니다.

예

예 1: "mySubProgrammA"의 기간 측정

프로그램 코드

```
...
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
```

프로그램이 라인 N80을 처리한 후 "mySubProgrammA"의 순수 실행 시간이 $AC_OLD_PROG_NET_TIME에 기록됩니다.

$AC_OLD_PROG_NET_TIME 값:

- M30 이후에도 유지합니다.
- 루프가 실행될 때마다 업데이트됩니다.

예 2: "mySubProgrammA"와 "mySubProgrammC"의 지속 시간 측정

프로그램 코드

```
...
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
```

4.22.2 소재 계수기

"소재 계수기" 기능은 특히 제어 시스템 내부에서 공작물 카운팅에 사용할 수 있는 다양한 계수기를 제공합니다.

계수기는 읽기 및 쓰기 액세스가 가능한 채널 시스템 변수 형식으로 존재하며 값 범위는 0~999,999,999입니다.

시스템 변수	의미
$AC_REQUIRED_PARTS	생산할 공작물 개수 (공작물 지령치 수) 이 계수기에서 실제 공작물 수 ($AC_ACTUAL_PARTS) 가 '0'으로 리셋되는 공작물 개수를 정의할 수 있습니다.
$AC_TOTAL_PARTS	가공된 공작물의 총 개수 (실제 총 공작물 수) 이 계수기는 시작 시점 이후 생산된 모든 공작물의 합계를 지정합니다. 디폴트 값으로 제어 시스템을 기동한 경우에만 이 값이 "0"으로 자동 리셋됩니다.

시스템 변수	의미
$AC_ACTUAL_PARTS	가공된 공작물 개수 (실제 총 공작물 수) 시작 시점 이후 생산된 모든 공작물의 총합이 계산됩니다. $AC_REQUIRED_PARTS > 0인 조건에서 필요한 공작물 수 ($AC_REQUIRED_PARTS) 에 도달하면 계수기는 자동으로 '0'으로 리셋됩니다.
$AC_SPECIAL_PARTS	사용자가 선택한 공작물 개수 이 계수기에서는 사용자별 공작물 개수를 설정할 수 있습니다. 공작물의 지령치 수에 도달하면 ($AC_REQUIRED_PARTS) 알람이 출력되도록 정의할 수 있습니다. 사용자가 카운터를 직접 리셋해야 합니다.

참고

디폴트 값으로 제어 시스템을 기동한 경우 모든 소재 계수기가 "0"으로 설정되어 활성화 여부에 관계 없이 읽고 쓸 수 있게 됩니다.

참고

채널별 머신 데이터를 사용하면 계수기 활성화, 계수기 리셋 시간 및 계수 알고리즘을 제어할 수 있습니다.

참고

사용자 정의 M 명령을 사용한 소재 계수

다양한 소재 계수기의 계수 펄스가 프로그램 끝이 아닌 사용자 정의된 M 명령 (M2/M30) 을 통해 실행되도록 머신 데이터를 설정할 수 있습니다.

4.23 프로그램 시뮬레이션

4.23.1 기능

현재 가공 프로그램이 프로그램 시뮬레이션에서 완전하게 계산되고 결과가 사용자 인터페이스에 그림으로 표시됩니다. 실제 축 이동 없이 프로그래밍 결과가 테스트됩니다. 프로그램된 가공 단계가 잘못된 경우 초기 단계에 이를 감지하므로 잘못된 공작물 가공 작업을 미연에 방지할 수 있습니다.

시뮬레이션 NC

시뮬레이션은 자체 NC 인스턴스를 사용합니다(시뮬레이션 NC). 따라서 시뮬레이션이 시작되기 전에 실제 NC가 시뮬레이션 NC에 정렬되어야 합니다. 이 정렬을 통해 모든 활성 머신 데이터를 NC에서 읽은 후 시뮬레이션 NC로 읽어 들입니다. NC 및 싸이클 머신 데이터는 활성 머신 데이터에 포함됩니다.

4.23.2 컴파일 싸이클과 연계한 프로그램 시뮬레이션

일부 컴파일 싸이클(CC)은 프로그램 시뮬레이션에서 지원됩니다. 제어 시스템의 전원을 켜면 지원되는 컴파일 싸이클의 머신 데이터가 정렬됩니다. "시뮬레이션 시작"과의 정렬은 발생하지 않습니다!

참고

가공 프로그램에서 지원되지 않는 CC의 CC별 언어 명령과 머신 데이터는 사용할 수 **없습니다**("가공 프로그램의 CC 명령" 항목 참조).

지원되는 CC의 특수 모션(OEM 변환)은 일부 상황에서 잘못 표시됩니다.

가공 프로그램의 CC 명령

지원되지 않는 컴파일 싸이클(OMA1 ... OMA5, OEMIPO1/2, G810 ... G829, 해당 절차 및 기능)의 가공 프로그램에서 사용되는 언어 명령은 개별적으로 처리하지 않아도 알람 메시지가 생성되고 시뮬레이션이 취소됩니다.

해결:

가공 프로그램 ($P_SIM query) 에서 누락된 CC 고유 언어 요소를 개별적으로 처리하십시오.

예:

프로그램 코드	설명
N1 G01 X200 F500	
IF (1==$P_SIM)	
N5 X300	; CC가 시뮬레이션을 위해 활성화되지 않음.
ELSE	
N5 X300 OMA1=10	
ENDIF	

4.24 추가 기능

4.24.1 가공 데이터 활성화 (NEWCONF)

NEWCONF 명령은 모든 머신 데이터를 활성화합니다. 또한 "MD 데이터 활성" 소프트 키를 눌러 HMI 사용자 인터페이스에서 기능을 활성화할 수 있습니다.

NEWCONF 기능이 실행되면 함축적인 전처리 정지가 실행됩니다. 다시 말해 경로 이동이 중단됩니다.

구문

```
NEWCONF
```

의미

NEWCONF:	"NEW_CONFIG" 효율성 레벨의 모든 머신 데이터를 활성 상태로 설정하는 명령어.

가공 프로그램에서 채널 간 NEWCONF 실행

가공 프로그램에서 축 머신 데이터를 변경한 후 NEWCONF로 활성화한 경우, NEWCONF는 가공 프로그램 채널에 영향을 주는 변경 내용이 포함된 머신 데이터만 활성화합니다.

참고

모든 변경 내용이 적용되도록 하려면 머신 데이터의 변경 내용에 의해 영향을 받는 축 또는 기능을 계산하는 모든 채널에서 NEWCONF 명령을 실행해야 합니다.

축 머신 데이터는 NEWCONF에 영향을 미치지 않습니다.

PLC에 의해 제어되는 축은 반드시 축 RESET을 수행해야 합니다.

예

밀링: 서로 다른 방법으로 드릴 가공

프로그램 코드	코멘트
N10 $MA_CONTOUR_TOL[AX]=1.0	; 머신 데이터 변경.
N20 NEWCONF	; 머신 데이터 활성화.
...	

4.24.2 NC 언어(STRINGIS)의 범위 확인

"STRINGIS(...)" 기능을 사용하여 지정된 문자열을 실제 언어 범위에서 NC 프로그래밍 언어 요소로 사용할 수 있는지 여부를 확인할 수 있습니다.

정의

```
INT STRINGIS(STRING <Name>)
```

구문

```
STRINGIS(<Name>)
```

의미

STRINGIS:	반환값이 있는 함수
<name>:	검사할 NC 프로그래밍 언어 요소의 이름
반환값:	반환값 형식은 yxx (10진수)

NC 프로그래밍 언어 요소

다음과 같은 NC 프로그래밍 언어 요소를 검사할 수 있습니다.

- 모든 현존하는 G 코드 그룹의 G 코드, 예: "G0", "INVCW", "POLY", "ROT", "KONT", "SOFT", "CUT2D", "CDON", "RMBBL", "SPATH"
- DIN 또는 NC 어드레스 (예: "ADIS", "RNDM", "SPN", "SR", "MEAS")
- 기능, 예: "TANG(...)" 또는 "GETMDACT"
- 절차, 예: "SBLOF".
- 키워드, 예: "ACN", "DEFINE" 또는 "SETMS"
- 시스템 데이터 (예: 머신 데이터 $M..., 셋팅 데이터 $S..., 옵션 데이터 $O...)
- 시스템 변수 $A... , $V... , $P...
- 산술 파라미터 R...
- 활성화된 싸이클의 싸이클 이름
- GUD 및 LUD 변수
- 매크로 이름
- 라벨명

반환값

반환값 중 처음 세 자리만 유효합니다. 반환값 형식은 yxx입니다. 여기서 y는 기본 정보, xx는 상세 정보입니다.

반환값	의미	
000	이 시스템에서 알지 못하는 '이름' 문자열입니다.[1)]	
100	'이름' 문자열이 NC 프로그래밍 언어 요소이지만 현재 프로그래밍할 수 없습니다 (옵션/기능 비활성 상태).	
2xx	'이름' 문자열이 NC 프로그래밍 언어 요소이며 프로그래밍 가능합니다 (옵션/기능 활성 상태). 상세 정보 xx는 요소 유형에 대한 추가 정보를 포함하고 있습니다.	
	xx	의미
	01	DIN 어드레스 또는 NC 어드레스[2)]
	02	G 코드(예: G04, INVCW)
	03	반환값이 있는 함수
	04	반환값이 없는 함수
	05	키워드 (예: DEFINE)
	06	머신 데이터 ($M...), 셋팅 데이터 ($S...) 또는 옵션 데이터 ($O...)
	07	시스템 파라미터 (예: 시스템 변수 ($...) 또는 산술 파라미터 (R...)
	08	싸이클(싸이클은 NC에 로드되어야 하고 싸이클 프로그램은 활성화되어야 합니다.) [3)]
	09	GUD 변수 (GUD 변수는 GUD 정의 파일에 정의해야 하고 활성화해야 합니다.)
	10	매크로 이름 (매크로는 매크로 정의 파일에 정의해야 하고 활성화시켜야 합니다.) [4)]
	11	실제 가공 프로그램의 LUD 변수
	12	ISO G 코드(ISO 언어 모드가 활성화되어야 합니다.)
400	'이름' 문자열은 xx == 01 또는 xx == 10로 식별되지 않았던 NC 어드레스이며 G 또는 R 코드가 아닙니다. [2)]	

반환값	의미
y00	구체적인 지정 불가능

1) 제어 시스템에 따라 특정 상황에서는 Siemens NC 언어 명령의 서브세트만 인식됩니다 (예: SINUMERIK 802D sl). 이러한 시스템의 경우 원칙적으로 Siemens NC 언어 명령인 문자열에는 값 0이 반환됩니다.
MD10711 $MN_NC_LANGUAGE_CONFIGURATION을 사용하면 이와 같은 동작을 변경할 수 있습니다.
MD10711 = 1인 경우 Siemens NC 언어 명령에 항상 100이라는 값이 반환됩니다.

2) NC 어드레스는 문자 A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z 입니다. 또한, 이 NC 어드레스는 어드레스 확장자와 함께 프로그래밍할 수 있습니다. 어드레스 확장자는 STRINGIS로 검사할 때 지정할 수 있습니다. 예: 201 == STRINGIS("A1").
문자: D, F, H, L, M, N, O, P, S, T는 NC 어드레스이거나 사용자가 정의한 보조 기능입니다. 이 NC 어드레스나 보조 기능에는 값 400이라는 값이 반환됩니다. 예: 400 == STRINGIS("D"). 이 NC 어드레스는 STRINGIS로 검사할 때 어드레스 확장자와 함께 지정할 수 없습니다.
예: 000 == STRINGIS("M02"), 400 == STRINGIS("M").

3) 싸이클 파라미터의 이름은 STRINGIS로 검사할 수 없습니다.
4) G, H, M, L처럼 매크로로 정의된 어드레스는 매크로로 식별됩니다.

예제

다음 예제에서는 문자열로 지정된 NC 언어 요소를 별도 언급이 없는 한 원칙적으로 제어 시스템에 프로그래밍할 수 있다고 가정합니다.

1. 문자열 "T"는 보조 기능으로 정의됩니다.
   ```
   400 == STRINGIS("T")
   000 == STRINGIS ("T3")
   ```
2. 문자열 "X"는 축으로 정의됩니다.
   ```
   201 == STRINGIS("X")
   201 == STRINGIS("X1")
   ```
3. 문자열 "A2"는 확장자가 있는 어드레스로 정의됩니다.
   ```
   201 == STRINGIS("A")
   201 == STRINGIS("A2")
   ```
4. 문자열 "INVCW"는 G 코드로 정의됩니다.
   ```
   202 == STRINGIS("INVCW")
   ```
5. 문자열 "$MC_GCODE_RESET_VALUES"는 머신 데이터로 정의됩니다.
   ```
   206 == STRINGIS("$MC_GCODE_RESET_VALUES")
   ```
6. 문자열 "GETMDACT"는 NC 언어 함수입니다.
   ```
   203 == STRINGIS("GETMDACT ")
   ```
7. 문자열 "DEFINE"은 키워드입니다.
   ```
   205 == STRINGIS("DEFINE")
   ```
8. 문자열 "$TC_DP3"은 시스템 파라미터 (공구 길이 콤포넌트) 입니다.
   ```
   207 == STRINGIS("$TC_DP3")
   ```
9. 문자열 "$TC_TP4"는 시스템 파라미터 (공구 크기) 입니다.
   ```
   207 == STRINGIS("$TC_TP4")
   ```

10. 문자열 "$TC_MPP4"는 시스템 파라미터 (메거진 포켓 상태) 입니다.
   – 공구 메거진 관리가 활성 상태입니다. `207 == STRINGIS("$TC_MPP4") ;`
   – 공구 메거진 관리가 비활성 상태입니다. `000 == STRINGIS("$TC_MPP4")`
   "공구 메거진 관리" 단원을 참조하십시오.
11. 문자열 "MACHINERY_NAME"은 GUD 변수로 정의됩니다.
   `209 == STRINGIS("MACHINERY_NAME")`
12. 문자열 "LONGMACRO"는 매크로로 정의됩니다.
   `210 == STRINGIS("LONGMACRO")`
13. 문자열 "MYVAR"은 LUD 변수로 정의됩니다.
   `211 == STRINGIS("MYVAR")`
14. 문자열 "XYZ"는 NC, GUD 변수, 매크로 또는 싸이클 이름에서 인식할 수 없는 명령입니다.
   `000 == STRINGIS("XYZ")`

공구 메거진 관리

공구 메거진 관리 기능이 비활성이면 머신 데이터와 관계 없이 공구 메거진 관리의 시스템 파라미터에 STRINGIS를 제공합니다.

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION

값은 항상 000 입니다.

ISO 모드

"ISO 모드" 기능이 활성화 상태인 경우

- MD18800 $MN_MM_EXTERN_LANGUAGE (활성화, 외부 NC 언어)
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM (조정할 제어 시스템)

STRINGIS는 먼저 지정된 문자열이 SINUMERIK G 코드인지 검사합니다. 문자열이 SINUMERIK G 코드가 아니면 이어서 ISO G 코드인지 검사합니다.

프로그래밍된 전환 (`G290` (SINUMERIK 모드), `G291` (ISO 모드)) 은 STRINGIS에 영향을 미치지 않습니다.

예

STRINGIS(...) 기능과 관련된 머신 데이터는 다음과 같은 값을 갖습니다.

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION = 2 (옵션이 설정된 NC 언어 명령만 인식되는 것으로 간주)
- MD19410 $ON_TRAFO_TYPE_MASK = 'H0' (옵션: 변환)
- MD10700 $MN_PREPROCESSING_LEVEL='H43' (싸이클 전처리가 활성 상태)

다음 프로그램 예제는 에러 메시지 없이 실행됩니다.

프로그램 코드	코멘트
N1 R1=STRINGIS("TRACYL")	; R1 == 0, TRACYL이 변환 옵션을 찾을 수 없어 '알 수 없음'으로 인식됩니다.
N2 IF STRINGIS("TRACYL") == 204	
N3 TRACYL(1,2,3)	; N3을 스킵합니다.
N4 ELSE	
N5 G00	; 대신 N5가 실행됩니다.
N6 ENDIF	
N7 M30	

4.24.3 가공 프로그램(MMC)에서 창 양방향 호출

NC 프로그램의 사용자별 대화상자는 사전 정의된 서브프로그램 MMC(...)를 통해 사용자 인터페이스에 표시될 수 있습니다.

대화창의 구성은 다음 유형의 대화를 위해 이뤄집니다.

- Run MyScreens
- Easy XML
- 사용자 XML

추가 정보:

- 프로그래밍 매뉴얼
- 프로그래밍 매뉴얼 Easy XML

구문

```
MMC("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>","<QUIT>")
```

의미

MMC(...):	서브프로그램 이름 파라미터는 위치 코드화된 상태로 명시되고, 두 개의 열, 즉 명령 열 및 승인 열 내에서 콤마로 분리됩니다.
명령 열 이내 파라미터:	

<ADDRESS>:	구성된 사용자 대화 박스가 실행되는 작동 영역	
	기능	**조작 영역**
	"Run MyScreens" 사용자 대화창	CYCLES
	"Easy XML" 사용자 대화창	CYCLES
	사용자 XML	XML
	팝업 창 "Run MyScreens"	POPUPDLG
	팝업 창 "Easy XML"	POPUPDLG
<COMMAND>:	실행되어야 할 명령	
	기능	**명령**
	"Run MyScreens" 사용자 대화창	PICTURE_ON, PICTURE_OFF
	"Easy XML" 사용자 대화창	PICTURE_ON, PICTURE_OFF
	사용자 XML	XML_ON, XML_OFF
	팝업 창 "Run MyScreens"	PICTURE_ON, PICTURE_OFF
	팝업 창 "Easy XML"	PICTURE_ON, PICTURE_OFF
<FILE>:	표시된 대화창이 프로그램되는 파일의 이름	
	기능	**파일**
	"Run MyScreens" 사용자 대화창	<name>.com
	"Easy XML" 사용자 대화창	<name>.xml
	사용자 XML	<name>.xml
	팝업 창 "Run MyScreens"	<name>.com
	팝업 창 "Easy XML"	<name>.xml
	NC 프로그램 내 직접적인 구성을 가진 팝업 창 "Easy XML"(예 2 참조)	xmldial_emb.xml
<DIALOG>:	표시될 대화창의 이름	
	기능	**대화창 이름**
	NC 프로그램 내 직접적인 구성을 가진, 팝업 창 "Easy XML"을 제외한 모든 기능	<FILE> 파일에 구성된 대화창의 이름
	NC 프로그램 내 직접적인 구성을 가진 팝업 창 "Easy XML"(예 3 참조)	main

승인 열 이내 파라미터:		
<QUIT>:	인지 유형	
	N:	승인 없음. 명령을 전송한 후 프로그램 실행을 계속합니다. 따라서 명령이 성공적으로 실행되지 않으면 피드백이 없습니다. **참고** 디스플레이 시간(드웰 시간)이 NC 프로그램에서 프로그램 된다면 승인 유형 "N"은 반드시 사용되어야 합니다(아래 예 2 참조).
	A:	동기식 확인 명령을 출력한 후 프로그램 실행을 계속합니다. 반환값은 대화창 구성의 범위 내에서 정의되는 사용자 지정 승인 변수(GUD 변수)에 저장되고, NC 프로그램에서 읽혀질 수 있습니다.

보충 조건

- 대화창의 정의 파일 *.com은 반드시 'proj' 폴더에 저장해야 합니다.
- 대화창의 Easy XML 정의 파일 *.xml은 반드시 'appl' 폴더에 저장해야 합니다.
  정의 파일이 다른 디렉토리에 저장되면, 'appl' 디렉토리부터 시작해서 경로를 간접적으로 명시해야 합니다.
- 여러 채널의 사용자 정의 대화창은 동시에 표시될 수 없습니다.
- 시뮬레이션에서는 MMC 기능이 지원되지 않습니다.

예

예 1

대화창 표시 및 NC 프로그램에서의 사용자 작동에 대한 반응

프로그램 코드	설명
; 승인 변수 QUIT는 이미 전역 사용자 변수(GUD)로서 생성되어 있습니다.	
; 대화창이 구성되었을 때의 유형 열 관련:	
; DEF NCK STRING[20] QUIT	
QUIT = "XXX"	; 승인 변수 초기화
G4 F5	

프로그램 코드	설명
MMC("CYCLES,PICTURE_ON,test.com,test1","A ")	; 대화창 표시 ; - 작업 영역: CYCLES ; - 사진 상태: PICTURE_ON (디스플레이) ; - 대화창 스크린 파일: test.com ; - 대화창 스크린: test1
입력:	; 사용자 입력 대기
STOPRE	;전처리 정지.
IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK	; 소프트 키 "RUN"
IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK	; 소프트 키 "CHK"
GOTOB INPUT	; => 대기
WORK:	; 소프트 키 "RUN" 누름
MSG("처리를 계속함 -> NC 시작")	; 출력 메시지
MMC("CYCLES,PICTURE_OFF","N")	; 대화창 닫기
M0	; NC 시작 대기
GOTOF END	; => 프로그램 종료
확인:	; 소프트 키 "CHK" 누름
MSG("접근 위치 -> NC 시작")	; 출력 메시지
MMC("CYCLES,PICTURE_OFF","N")	; 대화창 닫기
M0	; NC 시작 대기
GOTOF END	; => 프로그램 종료
END:	
...	

예 2

대화창의 표시 시간은 드웰 시간을 이용하여 NC 프로그램에서 정의됩니다. 예:

프로그램 코드	설명
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_**ON**,xmldial_emb.xml,main","N")	; 대화창 표시
X200	
Z40	
MMC("POPUPDLG,PICTURE_**OFF**","N")	; 대화창 닫기

예 3

NC 프로그램에 팝업 문자 심기 및 사용

프로그램 코드

```
PROC POPUP_TEST
; ----------------------------- 문자 -----------------------------
```

프로그램 코드

```
; <main_dialog entry="rpara_main">
;    <let name="xpos" />
;    <let name="ypos" />
;    <let name="field_name" type="string" />
;    <let name="num" />
;    <menu name="rpara_main">
;       <open_form name="rpara_form"/>
;       <softkey_back>
;          <close_form />
;       </softkey_back>
;    </menu>
;    <form name="rpara_form">
;       <init>
;          <caption>mask from NC part program</caption>
;          <let name="count" >0</let>
;          <op>
;             xpos = 120;
;             ypos = 34;
;             "nck/Channel/Parameter/R[10]" = 10;
;          </op>
;          <!-- load the number of controls -->
;          <op>
;             num = "nck/Channel/Parameter/R[10]";
;          </op>
;          <while>
;             <condition> count < num</condition>
;             <print name="field_name" text="edit%d">count</print>
;             <op>
;                ypos = ypos + 24;
;                count = count + 1;
;             </op>
;          </while>
;       </init>
;       <paint>
;          <op>
;             xpos =  8;
;             ypos = 36;
;             count = 0;
;          </op>
;          <while>
;             <condition>count < num</condition>
```

프로그램 코드

```
;             <print name="field_name" text="R-Parameter%d">count</print>
;             <text xpos = "$xpos" ypos = "$ypos" >$$$field_name</text>
;             <op>
;                ypos = ypos + 24;
;                count = count + 1;
;             </op>
;          </while>
;       </paint>
;    </form>
; </main_dialog>
; ========================= 프로그램 선택 ============================
...
G94 F100
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")
G4 F4
X200
MMC("POPUPDLG,PICTURE_OFF","N")
G4 F2
X0
...
```

4.24.4 공정 데이터 공유 - 외부 장치/파일로 출력 (EXTOPEN, WRITE, EXTCLOSE)

가공 프로그램에서 외부 장치/파일로의 데이터 작성은 세 단계로 이뤄집니다.

1. 외부 장치/파일 열기
   EXTOPEN 명령을 사용하여 기록할 채널에 해당하는 외부 장치/파일을 엽니다.
2. 데이터 기록
   NC 언어의 문자열 기능을 사용하여 출력 데이터를 처리할 수 있습니다(예: SPRINT). 기록에는 WRITE 명령이 사용됩니다.
3. 외부 장치/파일 닫기
   채널에 할당된 외부 장치/파일은 프로그램 종료에 도달하거나 (M30) 채널 리셋이 필요할 때 EXTCLOSE 명령을 사용하여 다시 해제됩니다.

구문

```
DEF INT <Result>
DEF STRING[<n>] <Output>
…
EXTOPEN(<Result>,<ExtDev>,<SyncMode>,<AccessMode>,<WriteMode>)
```

```
…
<Output>="data output"
WRITE(<Result>,<ExtDev>,<Output>)
…
EXTCLOSE(<Result>,<ExtDev>)
```

의미

EXTOPEN:	외부 장치/파일을 열기 위한 사전 정의된 절차		
<Result>:	**파라미터 1:** 결과 변수 결과 변수 값을 사용하면 작업이 성공적이었는지 여부를 프로그램에서 평가한 다음 이에 따라 처리를 계속할 수 있습니다.		
	유형:	INT	
	값:	0	에러 없음
		1	외부 장치를 열 수 없습니다.
		2	외부 장치가 구성되지 않았습니다.
		3	외부 장치의 경로가 올바르지 않습니다.
		4	외부 장치를 사용할 수 있는 권한이 없습니다
		5	사용 모드: 외부 장치가 이미 "독점" 사용되고 있습니다.
		6	사용 모드: 외부 장치가 이미 "공유"되고 있습니다.
		7	파일 길이가 LOCAL_DRIVE_MAX_FILESIZE보다 깁니다.
		8	최대 외부 장치 수를 초과했습니다.
		9	LOCAL_DRIVE 옵션이 설정되지 않았습니다.
		11	예비
		12	기록 모드: 데이터가 extdev.ini와 충돌합니다.
		16	외부 경로가 잘못되었습니다.
		22	외부 장치가 장착되지 않았습니다.

<table>
<tr><td><ExtDev>:</td><td colspan="2">파라미터 2: 열고자 하는 외부 장치/파일의 심볼 이름</td></tr>
<tr><td></td><td>유형:</td><td>STRING</td></tr>
<tr><td></td><td colspan="2">심볼 이름은 다음으로 구성됩니다.
1. 논리적 디바이스 이름
2. 타당한 경우 이름 뒤에 파일 경로가 옵니다 ("/"를 사용하여 추가).
다음 논리적 디바이스 이름이 정의되었습니다.</td></tr>
<tr><td></td><td>"LOCAL_DRIVE":</td><td>로컬 CF 카드 (사전 정의됨)</td></tr>
<tr><td></td><td>"CYC_DRIVE":</td><td>SIEMENS 싸이클에 사용하기 위한 예비용 드라이브 이름 (사전 정의)</td></tr>
<tr><td></td><td>"/dev/ext/1",...
"/dev/ext/9":</td><td>사용 가능한 네트워크 드라이브
참고:
extdev.ini 파일에 구성해야 합니다!</td></tr>
<tr><td></td><td>"/dev/cyc/1",
"/dev/cyc/2":</td><td>SIEMENS 싸이클에 사용하기 위한 예비용 드라이브 이름
참고:
extdev.ini 파일에 구성해야 합니다!</td></tr>
<tr><td></td><td colspan="2">파일 경로:
• "LOCAL_DRIVE" 및 "CYC_DRIVE"에 대한 파일 경로를 지정해야 합니다 (예:
"LOCAL_DRIVE/my_dir/my_file.txt").
• 논리적 디바이스 이름 "/dev/ext/1...9" 및 "/dev/cyc/1...2"를 다음과 같이 구성할 수 있습니다.
– 파일을 참조하는 경우 논리 디바이스 이름만 지정할 수 있습니다 (예:
"/dev/ext/4").
– 또는 파일 경로가 지정되어야 하는 경우에는 디렉토리를 참조합니다 (예:
"/dev/ext/5/my_dir/my_file.txt").</td></tr>
<tr><td></td><td colspan="2">참고:
논리적 디바이스 이름 "/dev/ext/1...9", "/dev/v24" 및 "/dev/cyc/1...2"는 대소문자를 구분하지 않습니다. 파일에 경로를 지정할 때는 대소문자가 중요합니다. "LOCAL_DRIVE" 및 "CYC_DRIVE"에는 대문자만 허용됩니다.</td></tr>
</table>

<SyncMode>:	**파라미터 3:** 이 장치/파일에 대한 WRITE 명령의 처리 모드		
	유형:	STRING	
	값:	"SYN":	동시 쓰기 쓰기 작업이 완료될 때까지 프로그램 실행이 중단됩니다. 동시 쓰기 작업이 성공적으로 완료되었는지 여부는 WRITE 명령의 에러 변수를 평가하여 확인할 수 있습니다.
		"ASYN":	비동시 쓰기 프로그램 실행이 WRITE 명령에 의해 중단되지 않습니다. **참고:** 이 모드에서는 WRITE 명령의 결과 변수가 아무 정보도 제공하지 않으며 항상 값이 0 (에러 없음) 입니다. 이 모드에서는 WRITE 명령이 성공했는지 여부를 확인할 수 없습니다.
<액세스모드>:	**파라미터 4:** 이 장치/파일의 사용 모드		
	유형:	STRING	
	값:	"SHARED":	장치/파일이 "공유" 모드에서 요청됩니다. 다른 채널도 장치를 사용할 수 있습니다. 즉, 다른 채널도 이 모드에서 개방할 수 있습니다.
		"EXCL":	장치/파일이 채널에서 배타적으로 사용되며, 다른 채널은 장치를 사용할 수 없습니다.
<WriteMode>:	**파라미터 5:** 이 장치/파일에 대한 WRITE 명령의 쓰기 모드 (옵션)		
	유형:	STRING	
	값:	"APP":	첨부 파일은 항상 그 내용을 유지합니다. 따라서 쓰기 호출은 끝 부분에 첨부됩니다.
		"OVR":	덮어쓰기 파일 내용이 삭제되고 이어지는 쓰기 호출을 사용하여 다시 생성됩니다.
	참고: 이 파라미터를 사용하면 extdev.ini 파일에 설정된 쓰기 모드를 덮어쓸 수 없습니다. 충돌이 있는 경우에는 EXTOPEN 호출이 승인되면서 에러가 발생합니다.		

WRITE:	출력 데이터를 쓰기 위한 사전 정의된 절차

EXTCLOSE:	열려 있는 외부 장치/파일을 닫기 위한 사전 정의된 절차		
<Result>:	**파라미터 1:** 결과 변수		
	유형:	INT	
	값:	0	에러 없음
		16	외부 경로가 잘못되었습니다.
		21	외부 장치를 닫을 때 에러 발생
<ExtDev>:	**파라미터 2:** 닫고자 하는 외부 장치/파일을 위한 상징적 식별자, EXTOPEN 참조. **참고:** 식별자는 EXTOPEN 호출에서 명시된 식별자와 동일해야 합니다.		

예

프로그램 코드

```
N10       DEF INT RESULT
N20       DEF BOOL EXTDEVICE
N30       DEF STRING[80] OUTPUT
N40       DEF INT PHASE
N50       EXTOPEN(RESULT,"LOCAL_DRIVE/my_file.txt","SYN","SHARED")
N60       IF RESULT > 0
N70          MSG("Error for EXTOPEN:" << RESULT)
N80       ELSE
N90          EXTDEVICE=TRUE
N100      ENDIF
…
N200      PHASE=4
N210      IF EXTDEVICE
N220         OUTPUT=SPRINT("End phase: %D",PHASE)
N230         WRITE(RESULT,"LOCAL_DRIVE/my_file.txt",OUTPUT)
N240      ENDIF
…
```

다음 또한 참고:

문자열 연산 (페이지 497)

파일 쓰기 (WRITE) (페이지 635)

4.24.5 알람 (SETAL)

알람을 NC 프로그램 내에 설치 할 수 있습니다. 알람은 사용자 인터페이스의 별도 필드에 표시됩니다. 알람은 알람 범주를 기준으로 항상 제어 시스템의 응답에 따라 작동합니다.

구문

```
SETAL(<No>[,<String>])
```

의미

SETAL(...):	사전 정의된 알람 설정 절차 SETAL은 개별 NC 블록으로 프로그래밍되어야 합니다.		
<No>:	알람 번호		
	데이터 유형:	INT	
	값 범위:	60000 ... 64999 (예비)	SIEMENS 싸이클용 알람
		65000 ... 69999	사용자 싸이클용 알람
<String>:	문자열(옵션)		
	데이터 유형:	STRING	
	사용자 싸이클 알람을 프로그래밍할 때 최대 4개의 파라미터를 포함하는 문자열을 지정할 수 있습니다. 변수 사용자 텍스트는 이런 파라미터에 정의할 수 있습니다. 다음과 같은 사전 정의된 파라미터를 사용할 수 있습니다:		
	%1	채널 번호	
	%2	블록 번호, 레이블	
	%3	싸이클 알람용 텍스트 색인	
	%4	추가 알람 파라미터	

참고

알람 텍스트는 사용자 인터페이스에서 구성해야 합니다.

참고

알람을 사용자 인터페이스에 사용되는 언어로 출력하려면 현재 HMI에 설정된 언어에 대한 정보가 필요합니다. 이 정보는 가공 프로그램 및 동기 동작에서 시스템 변수 $AN_LANGUAGE_ON_HMI (페이지 1541)를 통해 요청할 수 있습니다.

예

프로그램 코드	설명
...	
N100 SETAL (65000)	;알람 번호 65000 설정
...	

추가 정보

알람 응답 및 확인

사용자 싸이클 알람은 알람 응답 및 확인과 관련하여 다른 숫자 범위에 지정됩니다.

번호 범위	알람 응답	알람 확인
65000 - 65499	디스플레이, NC 시작 비활성화	리셋
65500 - 65999	디스플레이, NC 시작 비활성화(설정된 MD20194의 ASUB는 해당 안 됨)	리셋
66000 - 66999	디스플레이, NC 시작 비활성화, 사전 디코딩된 블록 실행 후 모션 정지	리셋
67000 - 67999	디스플레이	취소
68000 - 68999	디스플레이, NC 시작 비활성화, 즉시 보간기 정지	리셋
69000 - 69999	디스플레이, NC 시작 비활성화, 다음 블록 종료 후 정지	리셋

4.24.6 소재를 정의하십시오(공작물).

그래픽 시뮬레이션에다 소재를 표시하려면 제어자는 소재의 형태와 크기를 알고 있어야 합니다. 따라서 사용자는 사용자 인터페이스를 통하거나 NC 프로그램에서 직접 소재를 정의할 수 있는 권한을 가집니다. 소재의 정의는 (프로그램 종료/채널/BAG) 리셋에도 불구하고 그대로 남습니다. 이들은 제어 시스템에 전원이 켜지는 다음 번에 자동으로 삭제됩니다.

구문

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

의미

WORKPIECE(...):		소재 정의를 위해 사전에 정의된 절차		
		전처리 정지:	예	
		블록에서만:	예	
파라미터:				
1	"<WP>":	공작물 이름(옵션)		
		데이터 유형:	STRING	
		하나의 채널에 여러 개의 공작물이 있는 경우에만 명시가 필요합니다. 명시 없이도 "WORKP<n>"은 자동으로 수락되는데, <n>은 선언 채널의 개수입니다.		
2	"<RefP>":	클램핑(옵션, 밀링 기계만 해당)		
		데이터 유형:	STRING	
		값 범위:	"테이블"	고정 테이블 클램핑
			"A"	회전 축 A에의 클램핑
			"B"	회전 축 B에의 클램핑
			"C"	회전 축 C에의 클램핑
		사전조건: 테이블 또는 회전 축은 소재의 클램핑을 위한 해당 머신 데이터를 통해 작동 가능 상태가 됩니다(SINUMERIK Operate 스타트업 매뉴얼).		
3	"<ZeroOffset>":	소재를 위치시키기 위한 셋터블 워크 옵셋(프로그램 할 수 없음) 소재를 위치시키기 위한 셋터블 워크 옵셋의 선택은 사용자 인터페이스를 통한 소재 입력에 대해서만 제공됩니다. 가공 프로그램에서의 직접적인 소재 정의를 위해 소재는 항상 현재 유효한 워크 옵셋을 참고합니다.		

4	"<Type>":	소재 형태		
		데이터 유형:	STRING	
		값 범위:	"CYLINDER":	실린더
			"PIPE":	파이프
			"RECTANGLE":	정육면체
			"BOX":	직육면체
			"N_CORNER":	n 에지를 갖는 다각형
5 ... 12	<Par5> ... <Par12>:	소재 형태 설명을 위한 파라미터		
		데이터 유형:	REAL	
		요구되는 파라미터의 개수 및 의미는 각각의 소재 형태 및 비트 파라미터의 값에 달려있습니다. 참고: • "소재 형태 설명을 위한 파라미터" 테이블 • "비트 파라미터" 테이블		
WORPIECE():		파라미터가 없는 WORKPIECE 호출은 모든 소재 정의를 삭제합니다.		
WORPIECE(<WP>):		공작물 이름이 있는 WORKPIECE 호출은 본 소재 정의만 삭제합니다.		

도표 4-6 소재 형태 설명을 위한 파라미터

소재 형태	파라미터							
	<Par5>	**<Par6>**	**<Par7>**	**<Par8>**	**<Par9>**	**<Par10>**	**<Par11>**	**<Par12>**
실린더	비트 파라미터 비트 코드화된 정수값으로 해석되는 실제 값. 비트는 다음 파라미터의 의미를 정의합니다("비트 파라미터" 테이블 참조).	원점 Z_0	길이 Z_1	가공 치수 Z_B	외부 직경 d_0	-	회전 축을 중심으로 한 회전	-
파이프		원점 Z_0	길이 Z_1	가공 치수 Z_B	외부 직경 d_0	내벽 두께 (inc) / 내부 직경 d_1 (abs)	회전 축을 중심으로 한 회전	-
정육면체		원점 Z_0	길이 Z_1	가공 치수 Z_B	너비 W	길이 L	회전 축을 중심으로 한 회전	-
직육면체		원점 Z_0	길이 Z_1	가공 치수 Z_B	X_0	Y_0	X_1	Y_1
n 에지를 갖는 다각형		원점 Z_0	길이 Z_1	가공 치수 Z_B	모서리 수	평면 횡단 너비	회전 축을 중심으로 한 회전	-

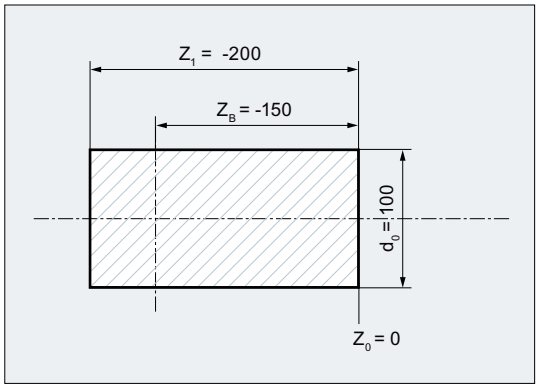
도표 4-7 비트 파라미터

비트	의미	
4 (0x0010)	직육면체: X_1	
	=0	inc
	=1	abs
5 (0x0020)	직육면체: Y_1	
	=0	inc
	=1	abs
6 (0x0040)	길이 Z_1 (최종 치수)	
	=0	inc
	=1	abs

비트	의미	
비트 7 (0x0080)	가공 치수 Z_B	
	=0	inc
	=1	abs
비트 8 (0x0100)	파이프: 내벽 두께 / 내부 직경	
	=0	inc
	=1	abs
9 (0x0200)	n 에지를 갖는 다각형	
	=0	평면 횡단 너비
	=1	에지 길이
12 (0x1000)	선삭 기계용 클램프:	
	=0	메인 스핀들
	=1	서브 스핀들
13 (0x2000)	서브 스핀들	
	=0	미러링 있음
	=1	미러링 없음

예

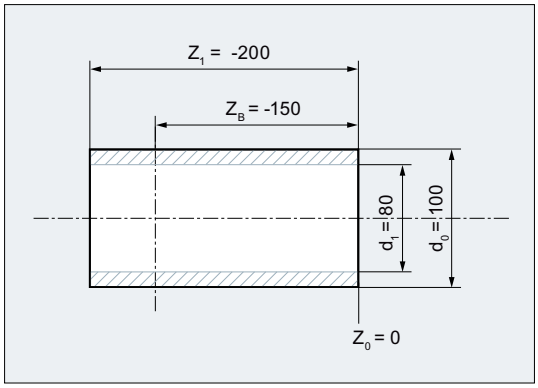
예 1: 선삭 기계 상의 원통형 소재



프로그램 코드	코멘트
. . .	

프로그램 코드	코멘트
WORKPIECE(,,,"CYLINDER",0,0,-200,-150,100)	; **소재 정의:** ; 소재 형태: 실린더 ; 비트 파라미터=0(비트 세트 없음) → 길이 값 및 가공 치수는 증분값임, 메인 스핀들에 대한 소재 ; 원점(Z0)=0 ; 길이(Z1)=-200 ; 가공 치수(ZB)=-150 ; 외부 직경(d0)=100
...	

예 2: 선삭 기계 상의 파이프 형태의 소재



프로그램 코드	코멘트
...	
WORKPIECE(,,,"PIPE",256,0,-200,-150,100,80)	; **소재 정의:** ; 소재 형태: 파이프 ; 비트 파라미터=256(Bit8=1) → 내부 직경은 절대값임; 길이 및 가공 치수는 증분값임, 메인 스핀들 상의 소재 ; 원점(Z0)=0 ; 길이(Z1)=-200 ; 가공 치수(ZB)=-150 ; 외부 직경(d0)=100 ; 내부 직경(d1)=80
...	

4.24.7 언어 전환 모드(G290, G291)

제어자는 외부 CNC 시스템으로부터 가공 프로그램을 읽어들이고 처리할 수 있는 권한을 줍니다. 선결 조건은 해당 NC 언어 모드(ISO 모드)가 스타트업 중 정의되었어야 한다는 것입니다.

ISO 모드는 각 채널에 대해 개별적으로 활성화될 수 있습니다. 예를 들어, 채널 2가 SINUMERIK 모드에서 활성화된 상태에서 채널 1은 ISO 모드에서 작동할 수 있습니다.

SINUMERIK 모드와 ISO 모드 사이의 전환은 G그룹 47의 명령을 통해 NC 프로그램에서 이뤄집니다. 활성 공구, 공구 보정 및 워크 옵셋은 전환의 영향을 받지 않습니다.

구문

```
G291
...
G290
```

의미

G290:	SINUMERIK 언어 모드 활성화	
	블록에서만:	예
	적용:	모달
G291:	ISO 언어 모드 활성화	
	블록에서만:	예
	적용:	모달

조건

SINUMERIK 모드

- G 명령의 기본 설정은 머신 데이터를 통해 각 채널별로 정의될 수 있습니다.
- ISO 모드의 어떤 언어 명령도 SINUMERIK 모드에서 프로그램 할 수 없습니다.

ISO 모드

- ISO 모드는 머신 데이터를 사용하여 제어 시스템의 기본 설정으로 설정될 수 있습니다. 그래서 ISO 모드에서 제어 시스템은 기본적으로 재부팅됩니다.
- ISO 모드의 G 명령만 프로그램 될 수 있습니다. SINUMERIK G 기능의 프로그래밍은 ISO 모드에서는 가능하지 않습니다.

- ISO 모드와 SINUMERIK 언어는 동일한 NC 블록에서 혼용할 수 없습니다.
- G 명령은 ISO 모드 M(밀링)과 ISO 모드 T(선삭) 간 전환에 사용할 수 없습니다.
- SINUMERIK 모드에서 프로그램 된 서브 프로그램을 호출할 수 있습니다.
- SINUMERIK 기능이 사용되는 경우, SINUMERIK 모드로의 전환은 반드시 먼저 이뤄져야 합니다(예 참조).

예

ISO 모드에서 선형 블록의 압축

프로그램 코드	설명
N5 G290	; SINUMERIK 언어 모드 활성화.
N10 COMPCAD	; COMPCAD는 Siemens 언어로 된 명령이고, 연속적인 선형 블록을 가능한 한 긴 경로 길이을 갖는 폴리노미얼 블록으로 교체하는 컴프레서 기능을 활성화합니다.
N15 G291	; ISO 언어 모드 활성화.
N20 G01 X100 Y100 F1000	; SINUMERIK 모드에서 COMPCAD가 활성화되었으므로 ISO 모드의 선형 블록도 이 기능으로 압축할 수 있습니다.
...	

추가 정보

기능 매뉴얼 ISO 언어 모드

4.25 사용자 선삭 프로그램

4.25.1 스톡 제거 기능 지원

사전 설정된 스톡 제거 프로그램이 제공됩니다. 또한 다음과 같은 기능을 이용해 특정한 스톡 제거 프로그램을 생성할 수 있습니다.

- 형상 테이블 생성(CONTPRON)
- 코드화 형상 테이블의 생성(CONTDCON)
- 형상 준비 종료(EXECUTE)
- 형상 요소 2개 사이의 교차점을 결정(INTERSEC)
  (CONTPRON을 이용해 생성한 테이블에서만)

- 테이블의 형상 요소를 블록별로 실행 (EXECTAB)
  (CONTPRON을 이용해 생성한 테이블에서만)
- 원 데이터 계산(CALCDAT)

참고

이런 기능은 스톡 제거 이외의 다른 목적에도 사용할 수 있습니다.

요구 사항

CONTPRON 또는 CONTDCON 기능을 호출하기 전에 완료해야 하는 작업은 다음과 같습니다.

- 충돌 방지 가공을 허용하는 시작점에 접근해야 합니다.
- 절삭 반경 보정은 G40으로 종료해야 합니다.

4.25.2 형상 테이블 생성(CONTPRON)

CONTPRON은 형상 준비 기능을 켭니다. 후속으로 호출되는 NC 블록은 실행되지 않지만 개별 이동으로 나뉘고 형상 테이블에 저장됩니다. 각 형상 요소는 형상 테이블의 2차원 배열에서 한 행에 해당합니다. 릴리프 절삭 번호가 반환됩니다.

구문

형상 준비 활성화:

```
CONTPRON(<contour table>,<machining type>,<relief cuts>,
<machining direction>)
```

형상 준비 종료 및 정상 실행 모드로 복귀:

```
EXECUTE(<ERROR>)
```

"형상 준비 종료(EXECUTE) (페이지 1165)"를 참조하십시오.

의미

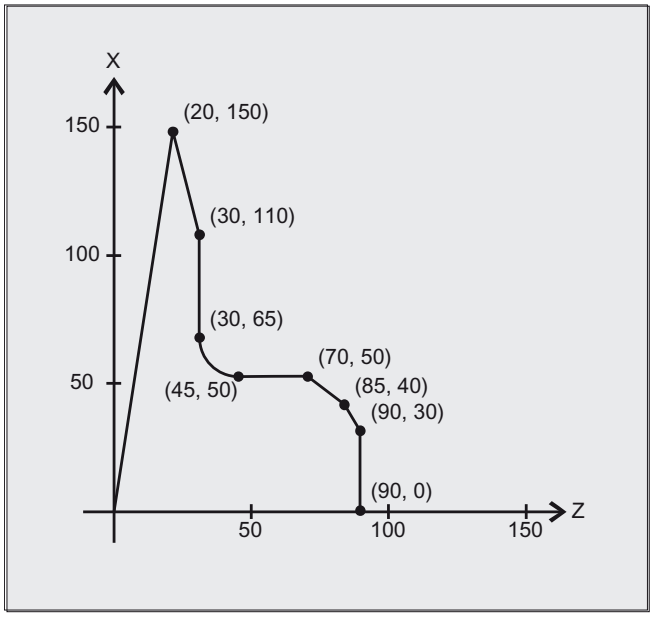
CONTPRON:	형상 테이블을 생성하기 위해 형상 준비를 활성화시키는 사전 정의된 절차
<contour table>:	형상 테이블의 이름

<machining type>:	가공 형식의 파라미터		
	유형:	CHAR	
	값:	"G":	세로 방향 선삭: 내부 기계가공
		"L":	세로 방향 선삭: 외부 가공
		"N":	단면 선삭 가공 내부 기계가공
		"P":	단면 선삭 가공 외부 가공
<relief cuts>:	생성할 릴리프 절삭 요소의 결과 변수		
	유형:	INT	
<machining direction>:	가공 방향의 파라미터		
	유형:	INT	
	값:	0	형상 준비, 정방향 (디폴트 값)
		1	양방향으로 형상 준비

예 1

형상 테이블 생성 시:

- 이름: "KTAB"
- 최대 30개의 형상 요소(원, 직선)
- 생성할 릴리프 절삭 요소의 변수 1개
- 에러 메시지를 위한 변수 1개



NC 프로그램:

프로그램 코드	코멘트
N10 DEF REAL KTAB[30,11]	; 이름이 KTAB이며 최대 30개의 형상 요소를 포함한 형상 테이블, 파라미터 값 11 (테이블 열의 수)은 고정
N20 DEF INT ANZHINT	; 이름이 ANZHINT인 릴리프 절삭 요소 번호의 변수
N30 DEF INT ERROR	; 에러 피드백 신호에 대한 변수 (0=에러 없음, 1=에러).
N40 G18	
N50 CONTPRON(KTAB,"G",ANZHINT)	; 형상 준비 활성화.
N60 G1 X150 Z20	; N60 ~ N120: 형상 설명
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE(ERROR)	; 형상 테이블 채우기를 마치고, 정상 프로그램 모드로 전환.
N140 …	; 테이블 처리 계속.

형상 테이블 KTAB:

색인 행	열									
(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

열 내용에 대한 설명:

(0) 다음 형상 요소의 포인터(해당 열에서 행 번호로)

(1) 이전 형상 요소의 포인터

(2) 이동을 위한 형상 모드 코딩

가능한 값은 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) 형상 요소의 시작점

현재 면의 (3) = 가로축, (4) = 세로축

(5), (6) 형상 요소의 시작점

현재 면의 (5) = 가로축, (6) = 세로축

(7) 최대/최소 표시기: 형상의 국소 최대와 최대 값을 표시

(8) 형상 요소와 가로축(세로 방향 가공)이나 세로축(전면 절삭) 사이의 최대값. 각도는 프로그램한 가공 형식에 따라 달라짐.

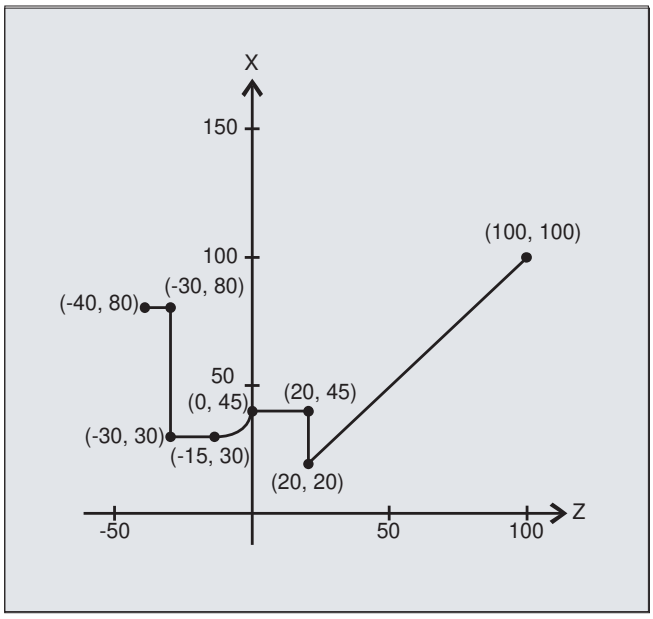
(9), (10) 원 블록인 경우에 형상 요소의 중심점 좌표

(9) = 가로축, (10) = 세로축

예 2

형상 테이블 생성 시:

- 이름: KTAB
- 최대 92개의 형상 요소(원, 직선)
- 작동 모드: 세로 선삭, 외부 가공
- 준비, 정방향 및 역방향



NC 프로그램:

프로그램 코드	코멘트
N10 DEF REAL KTAB[92,11]	; 이름이 KTAB이며 최대 92개의 형상 요소를 포함한 형상 테이블, 파라미터 값 11은 고정.
N20 DEF CHAR BT="L"	; CONTPRON 모드: 세로 선삭, 외부 가공
N30 DEF INT HE=0	; 릴리프 절삭 요소의 개수=0
N40 DEF INT MODE=1	; 준비, 정방향 및 역방향
N50 DEF INT ERR=0	; 에러 피드백 신호
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; 형상 준비 활성화.
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)	; 형상 테이블 채우기를 마치고, 정상 프로그램 모드로 전환.
...	

형상 테이블 KTAB:

형상 준비를 마친 후 형상은 양방향으로 이용할 수 있습니다.

색인	열										
행	(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

0, 1, 6, 8, 83, 85, 91 행에서 열의 내용과 비고 설명

예 1에 제시한 열의 내용 설명을 적용합니다.

테이블 행은 항상 0:

1) 선행: 라인 n은 형상 끝을 포함 (정방향)

2) 후속: 라인 n은 형상 테이블 끝을 포함 (정방향)

형상 요소 내에서 정방향으로 한 번씩:

3) 선행: 형상 시작 (정방향)

4) 후속: 형상 끝 (정방향)

항상 라인 형상 테이블 끝 (정방향) +1:

5) 선행: 릴리프 절삭 수 (정방향)

6) 후속: 릴리프 절삭 수 (역방향)

형상 요소 내에서 역방향으로 한 번씩:

7) 후속: 형상 끝 (역방향)

8) 선행: 형상 시작 (역방향)

항상 테이블의 마지막 행에서:

9) 선행: n 라인은 형상 테이블 시작 (역방향)

10) 후속: 라인 n은 형상 시작을 포함 (역방향)

추가 정보

사용할 수 있는 이송 명령(좌표계)

형상 프로그램에 사용할 수 있는 G 명령어:

- G 그룹 1: G0, G1, G2, G3

또한 다음 작업이 가능합니다.

- 라운딩과 모따기
- CIP 및 CT를 이용한 원 프로그래밍

스플라인, 폴리노미널, 나사 기능에서는 에러가 발생합니다.

프레임 활성화를 통한 좌표계 변경은 CONTPRON와 EXECUTE 사이에서 허용되지 않습니다. 또한 G70와 G71 또는 G700와 G710 사이에서도 좌표계 변경이 불가능합니다.

형상 테이블을 준비하는 동안 기하축을 GEOAX로 변경하면 알람이 발생합니다.

릴리프 절삭 요소

개별 릴리프 절삭 요소의 형상 설명은 서브프로그램이나 개별 블록에서 수행할 수 있습니다.

프로그램한 형상 방향과 독립적인 스톡 제거

CONTPRON을 이용한 형상 준비는 형상 테이블을 호출해 프로그램한 방향에 상관없이 이용할 수 있도록 확장되었습니다.

4.25.3 코드화 형상 테이블의 생성(CONTDCON)

CONTDCON으로 형상 준비를 시작하면 다음과 같은 호출한 NC 블록들은 메모리 사용을 최적화하기 위해 6열 형상 테이블에 코드화 형태로 저장됩니다. 각 형상 요소는 형상 테이블에서 한 행에 해당합니다. 아래에 설명된 코딩 규칙에 익숙해지면 테이블 행에서 싸이클용 DIN 코드 프로그램을 조합할 수 있습니다. 출력 지점의 데이터는 테이블 행에 0번으로 저장됩니다.

구문

형상 준비 활성화:
```
CONTDCON(<contour table>,<machining direction>)
```

형상 준비 종료 및 정상 실행 모드로 복귀:
```
EXECUTE(<ERROR>)
```

"형상 준비 종료(EXECUTE) (페이지 1165)"를 참조하십시오.

의미

CONTDCON:	코드화된 형상 테이블을 생성하기 위해 형상 준비를 활성화시키는 사전 정의된 절차		
<contour table>:	형상 테이블의 이름		
<machining direction>:	가공 방향의 파라미터		
	유형:	INT	
	값:	0	형상 블록의 순서에 따라 형상 준비(기본값)
		1	허용되지 않음

참고

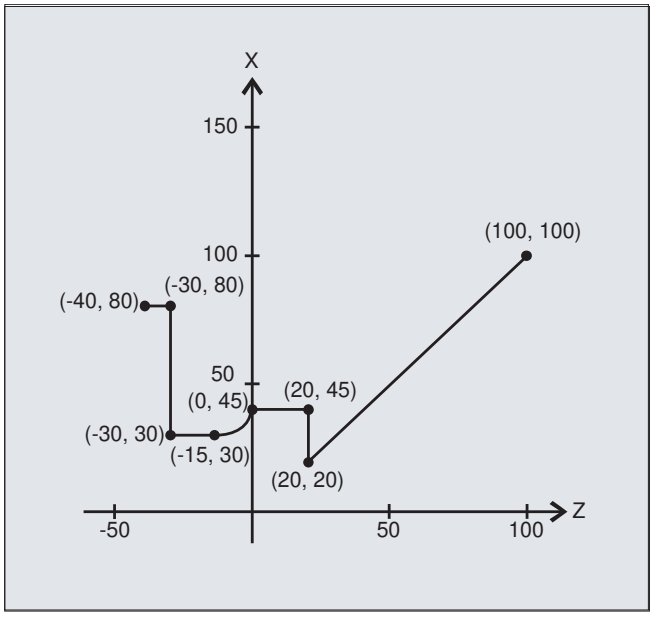
테이블에 포함될 프로그램 섹션에서 CONTDCON에 사용할 수 있는 G 코드는 CONTPRON 보다 많습니다. 또한 피드 속도 및 피드 속도 유형은 각 형상 섹션에 저장됩니다.

예

형상 테이블 생성 시:

- 이름: "KTAB"
- 형상 요소(원, 직선)

- 작동 모드: 선삭
- 가공 방향: 앞으로



NC 프로그램:

프로그램 코드	코멘트
N10 DEF REAL KTAB[9,6]	; 이름이 KTAB이고 테이블 셀이 9개인 형상 테이블. 8개의 형상 세트를 사용 가능. 파라미터 값 6(테이블의 열 개수)은 고정된 크기.
N20 DEF INT MODE = 0	; 가공 방향 변수. 표준값 0: 프로그램한 형상 방향에서만.
N30 DEF INT ERROR = 0	; 에러 피드백 신호에 대한 변수.
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; 형상 준비 호출 (MODE 생략 가능).
N110 G1 Z20 X20 F200	; 형상 설명.
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)	; 형상 테이블 채우기를 마치고, 정상 프로그램 모드로 전환.
...	

형상 테이블 KTAB:

	열 색인					
	0	1	2	3	4	5
행 색인	형상 모드	끝점 가로축	끝점 세로축	중심점 가로축	중심점 세로축	이송 속도
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

열 내용에 대한 설명:

시작점에서 0행 코딩:

0 열: 10^0 (1단위 숫자): G0 = 0
10^1 (10단위 숫자): G70 = 0, G71 = 1, G700 = 2, G710 = 3

1 열: 시작점 가로 좌표

2 열: 시작점 세로 좌표

3-4열: 0

5 열: 테이블에서 마지막 형상 조각의 라인 인덱스

1-n행: **형상 단편**의 엔트리

0 열: 10^0 (1단위 숫자): G0 = 0, G1 = 1, G2 = 2, G3 = 3
10^1 (10단위 숫자): G70 = 0, G71 = 1, G700 = 2, G710 = 3
10^2 (100단위 숫자): G90 = 0, G91 = 1
10^3 (1천 단위 숫자): G93 = 0, G94 = 1, G95 = 2, G96 = 3
10^4 (1만 단위 숫자): G60 = 0, G44 = 1, G641 = 2, G642 = 3
10^5 (10만 단위 숫자): G9 = 1

1 열: 끝점 가로축

2 열: 끝점 세로축

3 열: 원 보간의 중심점 가로축
4 열: 원 보간의 중심점 세로축
5 열: 이송 속도

추가 정보

사용할 수 있는 이송 명령(좌표계)

형상 프로그램에 사용할 수 있는 G 그룹과 G 명령어:

G 그룹 1: G0, G1, G2, G3
G 그룹 10: G60, G64, G641, G642
G 그룹 11: G9
G 그룹 13: G70, G71, G700, G710
G 그룹 14: G90, G91
G 그룹 15: G93, G94, G95, G96, G961

또한 다음 작업이 가능합니다.

- 라운딩과 모따기
- CIP 및 CT를 이용한 원 프로그래밍

스플라인, 폴리노미널, 나사 기능에서는 에러가 발생합니다.

프레임 활성화를 통한 좌표계 변경은 CONTDCON와 EXECUTE 사이에서 허용되지 않습니다. 또한 G70와 G71 또는 G700와 G710 사이에서도 좌표계 변경이 불가능합니다.

형상 테이블을 준비하는 동안 기하축을 GEOAX로 변경하면 알람이 발생합니다.

가공 방향

CONTDCON을 이용해 생성한 형상 테이블은 프로그램한 형상 방향에서 스톡 제거에 사용합니다.

4.25.4 형상 요소 2개 사이의 교차점을 결정(INTERSEC)

INTERSEC는 CONTPRON을 이용해 생성한 형상 테이블에서 두 일반화 형상 요소 간의 교차점을 결정합니다.

구문

```
<Status>=INTERSEC(<contour table_1>[<contour element_1>],
<contour table_2>[<contour element_2>],<intersection
point>,<machining type>)
```

의미

INTERSEC:	CONTPRON을 이용해 생성한 형상 테이블에서 형상 요소 2개 사이의 교차점을 결정하는 사전 정의된 기능		
<상태>:	교차점 상태의 변수		
	유형:	BOOL	
	값:	TRUE	교차점 발견
		FALSE	교차 없음
<contour table_1>:	첫 번째 형상 테이블의 이름		
<contour element_1>:	첫 번째 형상 테이블에서 형상 요소의 이름		
<contour table_2>:	두 번째 형상 테이블의 이름		
<contour element_2>:	두 번째 형상 테이블에서 형상 요소의 이름		
<교차점>:	활성 면에서 교차 좌표(G17 / G18 / G19)		
	유형:	REAL	
<가공 유형>:	가공 형식의 파라미터		
	유형:	INT	
	값:	0	파라미터 2 (표준값)로 활성 면에서 교차점 계산
		1	활성 면과 상관없이 교차점 계산

참고

변수는 명령을 사용하기 전에 정의해야 합니다.

형상을 전송할 때는 CONTPRON으로 정의한 값을 준수해야 합니다.

파라미터	의미
2	이동을 위한 형상 모드 코딩
3	형상 시작점 가로축
4	형상 시작점 세로축
5	형상 끝점 가로축

파라미터	의미
6	형상 끝점 세로축
9	가로축의 중심점 좌표(원 형상에서만)
10	세로축의 중심점 좌표(원 형상에서만)

예

TABNAME1 테이블에서 형상 요소 3과 TABNAME2 테이블에서 형상 요소 7의 교차를 계산합니다. 활성 면에서 교차 좌표는 변수 ISCOORD(첫 번째 요소는 가로축, 두 번째 요소는 세로축)으로 저장됩니다. 교차가 없으면 프로그램은 NOCUT(교차 없음)으로 이동합니다.

프로그램 코드	코멘트
DEF REAL TABNAME1[12,11]	; 형상 테이블 1
DEF REAL TABNAME2[10,11]	; 형상 테이블 2
DEF REAL ISCOORD [2]	; 교차 좌표에 대한 변수.
DEF BOOL ISPOINT	; 교차 상태에 대한 변수.
DEF INT MODE	; 가공 형식에 대한 변수.
…	
MODE=1	; 활성 면과 상관없이 계산.
N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE)	; 형상 요소의 교차점 호출.
N20 IF ISPOINT==FALSE GOTOF NOCUT	; NOCUT으로 이동.
…	

4.25.5 테이블의 형상 요소를 블록별로 실행(EXECTAB)

EXECTAB을 사용하면, 예를 들어 CONTPRON으로 생성된 테이블의 형상 요소들을 블록별로 실행할 수 있습니다.

구문

```
EXECTAB(<contour table>[<contour element>])
```

의미

EXECTAB:	형상 요소를 실행하기 위해 사전 정의된 절차
<contour table>:	형상 테이블의 이름
<contour element>:	형상 요소의 이름

예

KTAB 테이블의 0에서 2까지 형상 요소가 블록별로 실행됩니다.

프로그램 코드	코멘트
N10 EXECTAB(KTAB[0])	; KTAB 테이블의 형상 요소 0을 이송
N20 EXECTAB(KTAB[1])	; KTAB 테이블의 형상 요소 1을 이송
N30 EXECTAB(KTAB[2])	; KTAB 테이블의 형상 요소 2를 이송

4.25.6 원 데이터 계산(CALCDAT)

CALCDAT 명령을 사용하여 원호의 3개 또는 4개 지점에서 원호의 반경과 중심점 좌표를 계산할 수 있습니다. 지정하는 지점들은 서로 달라야 합니다.

4개 지점이 원에 직접 놓이지 않으면 원 중심점과 반경을 평균값으로 계산합니다.

참고

평균을 위한 계산 규칙

원호의 계산은 4회 수행됩니다.

1. 원 지점 1, 2, 3에서
2. 원 지점 1, 2, 4에서
3. 원 지점 1, 3, 4에서
4. 원 지점 2, 3, 4에서

원호 중심점 좌표의 가로 좌표와 세로 좌표 값은 4개 원호 계산에서 얻은 가로 좌표와 세로 좌표를 더한 후 4로 나누어서 계산합니다.

반경은 원호 계산으로부터 산출된 4개의 반경을 더한 값에 루트를 적용하고 그 결과에 0.5를 곱하여 계산합니다.

구문

```
<Status>=CALCDAT(<circle
points>[<number>,<type>],<number>,<result>)
```

의미

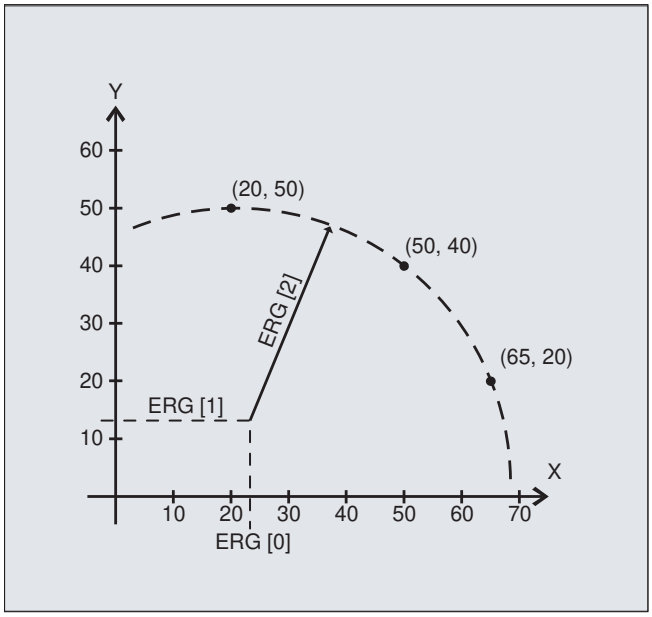
CALCDAT:	3개나 4개 지점에서 원의 반경과 중심점 좌표를 계산하는 사전 정의된 기능		
<Status>:	원 계산 상태의 변수		
	유형:	BOOL	
	값:	TRUE	지정한 지점이 원에 놓임
		FALSE	지정한 지점이 원을 **벗어남**
<circle points>[]:	파라미터를 이용해 원 지점 을 지정하는 변수		
	<number>:	원 지점의 개수(3이나 4)	
	<type>:	2점 좌표일 경우 좌표 데이터를 2로 입력	
<number>:	계산에 사용한 지점 개수의 파라미터(3이나 4)		
<result>[3]:	결과의 변수: 원의 중심점 좌표와 반경		
	0	원 중심점 좌표: 가로축 값	
	1	원 중심점 좌표: 세로축 값	
	2	반경	

참고

변수는 명령을 사용하기 전에 정의해야 합니다.

예

3개 지점을 사용할 때는 3개 지점이 원에 놓이는지 확인해야 합니다.



프로그램 코드	코멘트
N10 DEF REAL PT[3,2]=(20,50,50,40,65,20)	; 원의 지점들을 지정하는 변수.
N20 DEF REAL RES[3]	; 결과에 대한 변수.
N30 DEF BOOL STATUS	; 상태에 대한 변수.
N40 STATUS=CALCDAT(PKT,3,ERG)	; 결정된 원 데이터를 호출
N50 IF STATUS == FALSE GOTOF ERROR	; 에러로 이동.

4.25.7 형상 준비 종료(EXECUTE)

EXECUTE 명령은 형상 준비를 비활성화하고 동시에 시스템을 정상 실행 모드로 복귀시킵니다.

구문

```
EXECUTE(<ERROR>)
```

의미

EXECUTE:	형상 준비를 종료하도록 사전 정의된 절차	
<ERROR>:	에러 피드백 신호에 대한 변수	
	유형:	INT
	형상 준비 시 나타나는 변수의 값:	
	0	에러
	1	에러 없음

예

프로그램 코드

```
...
N30 CONTPRON(...)
N40 G1 X... Z...
...
N100 EXECUTE(...)
...
```

4.26 외부에서 싸이클 프로그래밍

4.26.1 공정 싸이클

4.26.1.1 소개

내용

이 절에서는 선삭, 밀링 및 연삭 기술을 위한 싸이클에 대해 설명합니다.

구조

싸이클에 대한 설명은 다음과 같이 구성되어 있습니다.

- **구문**
  전송 파라미터의 싸이클 이름 및 호출 순서
- **파라미터**
  개별 파라미터를 설명하는 테이블

파라미터 설명

파라미터에 대한 테이블에 다음 데이터가 명시되어 있습니다. 이름, 설명, 값 범위 및 다른 파라미터에 대한 의존성.

화면 형식에서 파라미터 참조 열은 외부에서 생성된 싸이클 호출이 다시 컴파일될 때 제어 시스템에서 프로그램된 값들을 더 쉽게 찾기 위해 제공됩니다.

"인터페이스 전용" 파라미터

특정 파라미터는 테이블에 "인터페이스 전용"이라고 표시됩니다. 이 파라미터는 싸이클 작동과 무관합니다. 이 파라미터는 싸이클 호출을 완벽하게 재컴파일하는 데에만 필요합니다. 파라미터를 프로그래밍하지 않아도 싸이클을 재컴파일할 수는 있습니다. 그런 경우 필드는 색상별로 식별되며 마스크에서 완성해야 합니다.

"예비용" 파라미터

"예비용"으로 표시되는 파라미터는 값 0 또는 쉼표로 프로그래밍해야 다음 호출 파라미터 지정이 내부 싸이클 파라미터와 일치하게 됩니다. 예외: 값 "" 또는 쉼표를 사용한 문자열 파라미터.

위치 패턴에서 싸이클 반복

위치 패턴에서 드릴링 및 밀링 싸이클을 반복할 수 있습니다 (모달 호출). 이러한 경우 MCALL은 싸이클 이전에 같은 라인에 쓰여져야 합니다 (예: `MCALL CYCLE83(...)`).

참고

특정 전송 파라미터 (예: `<_VARI>`, `<_GMODE>`, `<_DMODE>`, `<_AMODE>`) 가 간접적으로 파라미터로 프로그램된 경우, 재 컴파일링 시 화면 형식이 열리지만 정의된 선택 필드에 대한 명확한 할당이 없으므로 이것은 저장될 수 없습니다.

4.26.1.2 특정 기술 기반 개요

다음 개요 표는 모든 활용할 수 있는 외부 프로그래밍 가능 싸이클과 각 싸이클에 할당된 기술을 보여줍니다.

기술	공정 싸이클
드릴링	• CYCLE81 - 드릴링, 센터링 (페이지 1225) • CYCLE82 - 드릴링, 카운터보링 (페이지 1227) • CYCLE85 - 리밍 (페이지 1238) • CYCLE86 - 보링 (페이지 1240) • CYCLE83 - 심공 드릴링 1 (페이지 1230) • CYCLE830 - 심공 드릴링 2 (페이지 1277) • CYCLE84 - 동기 태핑 (페이지 1234) • CYCLE840 - 태핑 척을 사용한 태핑 (페이지 1289) • CYCLE78 - 드릴 나사 밀링 (페이지 1218) • CYCLE802 - 임의 위치 (페이지 1272) • HOLES1 - 행 위치 패턴 (페이지 1170) • CYCLE801 – 격자 또는 프레임 위치 패턴 (페이지 1270) • HOLES2 - 원호 또는 피치원 위치 패턴 (페이지 1171)
선삭	• CYCLE951 - 선삭 (페이지 1303) • CYCLE930 - 홈 (페이지 1296) • CYCLE940 – 언더컷 형상 E 및 F / 언더컷 나사 (페이지 1299) • CYCLE99 - 나사 선삭 (페이지 1253) • CYCLE98 - 나사 체인 (페이지 1246) • CYCLE92 - 절단 (페이지 1242)
형상 선삭	• CYCLE62 - 형상 호출 (페이지 1198) • CYCLE952 – 칩 제거 / 잔여 칩 제거 / 플런지 절삭 / 잔여 플런지 절삭 / 플런지 선삭 / 잔여 플런지 선삭 (페이지 1306)

기술	공정 싸이클
밀링	• CYCLE61 - 페이스 밀링 (페이지 1196) • POCKET3 - 사각 포켓 (페이지 1174) • POCKET4 - 원호 포켓 (페이지 1178) • CYCLE76 - 사각 스피곳 (페이지 1212) • CYCLE77 - 원형 스피곳 (페이지 1216) • CYCLE79 - 다각형 (페이지 1222) • SLOT1- 일반 슬롯 (페이지 1182) • SLOT2 - 원호 슬롯 (페이지 1186) • CYCLE899 - 개방형 슬롯 (페이지 1293) • LONGHOLE - 연장된 홀 (페이지 1189) • CYCLE70 - 나사 밀링 (페이지 1206) • CYCLE60 - 조각 (페이지 1192)
형상 밀링	• CYCLE62 - 형상 호출 (페이지 1198) • CYCLE72 - 경로 밀링 (페이지 1208) • CYCLE63 – 형상 포켓 밀링 / 형상 포켓 잔삭 가공 / 형상 스피곳 밀링 / 형상 스피곳 잔삭 가공 (페이지 1199) • CYCLE64 - 형상 포켓 예비 드릴링 (페이지 1203)
연삭	• CYCLE495 - 형상 트루잉 (페이지 1259) • CYCLE435 - 드레서 좌표계 설정 (페이지 1258) • CYCLE4071 - 반전 지점에서 절입을 통한 세로 방향 연삭 (페이지 1314) • CYCLE4072 - 반전 지점에서의 절입을 통한 세로 방향 연삭과 취소 신호 (페이지 1316) • CYCLE4073 - 연속 절입을 통한 세로 방향 연삭 (페이지 1320) • CYCLE4074 - 연속 절입을 통한 세로 방향 연삭 및 취소 신호 (페이지 1322) • CYCLE4075 - 반전 지점에서 절입을 통한 표면 연삭 (페이지 1326) • CYCLE4077 - 반전 지점에서의 절입을 통한 표면 연삭과 취소 신호 (페이지 1329) • CYCLE4078 - 연속 절입을 통한 표면 연삭 (페이지 1333) • CYCLE4079 - 간헐적 절입을 통한 표면 연삭 (페이지 1335)

기술	공정 싸이클
기타	• CYCLE782 - 부하에 맞게 조정 (페이지 1261) • CYCLE800 – 스위블 평면 / 스위블 공구 / 정렬 공구 (페이지 1265) • CYCLE806 - 보간 선삭 (페이지 1276) • CYCLE832 - 고속 가공 설정 (페이지 1285)
모두	• GROUP_BEGIN - 프로그램 블록 시작 (페이지 1338) • GROUP_END - 프로그램 블록 종료 (페이지 1339) • GROUP_ADDEND - 시험 절삭 추가 종료 (페이지 1339)

4.26.1.3 HOLES1 - 행 위치 패턴

구문

```
HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	X0	<SPCA>	REAL	첫 번째 축을 따라 여러 개의 홀로 이루어진 행의 원점 (abs)
2	Y0	<SPCO>	REAL	두 번째 축을 따라 여러 개의 홀로 이루어진 행의 원점 (abs)
3	α0	<STA1>	REAL	기본 회전 각도 (첫 번째 축에 대한 각도)
4	L0	<FDIS>	REAL	첫 번째 홀에서 원점까지의 거리
5	L	<DBH>	REAL	홀 사이 간격 지정
6	N	<NUM>	INT	홀 개수
7		<_VARI>	INT	예비용
8		<_UMODE>	INT	예비용
9		<_HIDE>	STRING [200]	숨겨진 위치 • 최대 198 문자 • 연속 위치 번호 지정. 예: "1,3" (위치 1과 3은 실행되지 않음)
10		<_NSP>	INT	예비용

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
11		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)

4.26.1.4 HOLES2 - 원호 또는 피치원 위치 패턴

구문

```
HOLES2(<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>,
<_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)
```

파라미터

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미	
1	X0	<CPA>	REAL	첫 번째 축을 따라 여래 개의 홀로 이루어진 원호의 중심점 (abs) 1번째 축의 원점	(XY의 경우) (XA, YB, ZC의 경우)
2	Y0	<CPO>	REAL	두 번째 축을 따라 여러 개의 홀로 이루어진 원호의 중심점 (abs) 2번째 축의 원점	(XY의 경우) (XA, YB, ZC의 경우)
3	R	<RAD>	REAL	원호 패턴의 반경	(XY의 경우)
4	α0	<STA1>	REAL	시작각 또는 1번째 로터리 축 위치	(XY의 경우) (XA, YB, ZC의 경우)

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
5	α1	<INDA>	REAL	전진 각도 (피치 원에만 해당)		(XY, XA, YB, ZC 의 경우)
					< 0 =	CW 방향
					> 0 =	CCW 방향
6	N	<NUM>	INT	위치 개수.		

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미			
7		<_VARI>	INT	가공 형식			
				UNITS:	예비용		
				TENS:	포지셔닝 유형		
					0 =	접근 위치 - 직선	
					1 =	접근 위치 - 원호 경로	
				HUNDREDS:	예비용		
				THOUSANDS:	원호 패턴		
					0 =	호환 모드, INDA = 0인 경우 완전한 원, INDA <> 0인 경우 피치 원	
					1 =	완전한 원	
					2 =	피치 원호	
				TEN THOUSANDS:	로터리 축을 사용하는 위치 패턴		
					0 =	XY (로터리 축 사용 안 함)	(XY의 경우)
					1 =	XA (X 축 및 X 중심으로 회전하는 로터리 축)	(XA만 해당)
					2 =	YB (Y 축 및 Y 중심으로 회전하는 로터리 축)	(YB만 해당)
					3 =	ZC (Z 축 및 C 중심으로 회전하는 로터리 축)	(ZC만 해당)
				ONE MILLION + HUNDRED THOUSANDS:	옵셋 (동일한 축을 중심으로 회전하는 로터리 축이 여러 개인 경우, 인덱스가 너무 크면 1번째 축만 사용.)		
					00 =	1번째 A, B 또는 C축	
					01 =	2번째 A, B, 또는 C축	
					...		
					10 =	20번째 A, B, C축	
8		<_UMODE>	INT	예비용			

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
9		<_HIDE>	STRING [200]	예비용		
10		<_NSP>	INT	예비용		
11		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)

4.26.1.5 POCKET3 - 사각 포켓

구문

```
POCKET3(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_CRAD>,
<_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>,
<_CDIR>, <_VARI>, <_MIDA>, <_AP1>, <_AP2>, <_AD>, <_RAD1>,
<_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<_RTP>	REAL	후퇴 평면 (abs)
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	포켓 깊이 (abs/inc), <_AMODE> 참조
5	L	<_LENG>	REAL	포켓 길이 (inc, 부호를 포함하여 입력)
6	W	<_WID>	REAL	포켓 너비 (inc, 부호를 포함하여 입력)
7	R	<_CRAD>	REAL	포켓의 코너 반경
8	X0	<_PA>	REAL	원점, 첫 번째 축 (abs)
9	YO	<_PO>	REAL	원점, 두 번째 축 (abs)
10	α0	<_STA>	REAL	회전 각도, 세로 축 (L) 과 첫 번째 축 사이의 각도

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
11	DZ	<_MID>	REAL	최대 깊이 절입		
12	UXY	<_FAL>	REAL	정삭 여유, 평면		
13	UZ	<_FALD>	REAL	정삭 여유, 깊이		
14	F	<_FFP1>	REAL	평면에서 이송 속도		
15	FZ	<_FFD>	REAL	깊이 절입 속도		
16		<_CDIR>	INT	밀링 방향:	0 =	하향 절삭
					1 =	상향 절삭
17		<_VARI>	INT	가공 형식		
				UNITS:		
					1 =	황삭
					2 =	정삭
					4 =	에지 정삭
					5 =	모따기
				TENS:		
					0 =	사전 드릴링, G0으로 절입
					1 =	수직, G1로 절입
					2 =	헬리컬
					3 =	포켓 세로 축의 오실레이션
				HUNDREDS:	예비용	
18	DXY	<_MIDA>	REAL	최대 평면 절입, 단위에 대해서는 <_AMODE> 참조		
19	L1	<_AP1>	REAL	사전 가공 길이 (inc)		
20	W1	<_AP2>	REAL	사전 가공 너비 (inc)		
21	AZ	<_AD>	REAL	사전 가공 깊이 (inc)		
22	ER	<_RAD1>	REAL	헬리컬 절입에서 헬리컬 경로의 반경		
	EW			오실레이션을 위한 최대 절입 각도		
23	EP	<_DP1>	REAL	헬리컬 절입에서 헬리컬 피치		
24		<_UMODE>	INT	예비용		
25	FS	<_FS>	REAL	모따기 너비 (inc)		
26	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs/inc), <_AMODE> 참조		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
27		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택/시작점 계산만	
					0 =	호환성 모드
					1 =	일반 가공
				THOUSANDS:	중심/코너를 지나는 치수 지정	
					0 =	호환성 모드
					1 =	중심을 통한 치수 지정
					2 =	코너 지점의 치수 지정, 포켓 위치 +LENG/+WID
					3 =	코너 지점의 치수 지정, 포켓 위치 -LENG/+WID
					4 =	코너 지점의 치수 지정, 포켓 위치 +LENG/-WID
					5 =	코너 지점의 치수 지정, 포켓 위치 -LENG/-WID
				TEN THOUSANDS:	전체 가공/재가공	
					0 =	호환성 모드 (전처럼 <_AP1>, <_AP2> 및 <_AD> 처리)
					1 =	전체 가공
					2 =	사후 가공

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
28		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	이송 속도 유형: G 그룹 (G94/G95), 표면 및 깊이 이송 속도의 경우	
					0 =	호환성 모드
					1 =	싸이클 호출 이전과 같은 G 코드. 표면 및 깊이 이송 속도의 경우 G94/G95는 동일
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
29		<_AMODE>	INT	대체 모드		
				UNITS:	포켓 깊이 (Z1)	
					0 =	절대 (호환 모드)
					1 =	증분치
				TENS:	평면 절입 단위 (DXY)	
					0 =	mm
					1 =	공구 직경의 %
				HUNDREDS:	모따기를 위한 삽입 깊이 (ZFS)	
					0 =	절대치
					1 =	증분치

4.26.1.6 POCKET4 - 원호 포켓

구문

```
POCKET4(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_CDIAM>, <_PA>, <_PO>,
<_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>,
<_MIDA>, <_AP1>, <_AD>, <_RAD1>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<_RTP>	REAL	후퇴 평면 (abs)
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	포켓 깊이 (abs/inc), <_AMODE> 참조
5	Ø	<_CDIAM>	REAL	포켓 직경 또는 반경, <_DMODE> 참조
6	X0	<_PA>	REAL	원점, 첫 번째 축 (abs)
7	Y0	<_PO>	REAL	원점, 두 번째 축 (abs)
8	DZ	<_MID>	REAL	최대 깊이 설정, <_VARI> = 평면 참조 최대 헬리컬 설정, <_VARI> = 헬리컬 참조

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
9	UXY	<_FAL>	REAL	정삭 여유, 평면		
10	UZ	<_FALD>	REAL	정삭 여유, 깊이		
11	F	<_FFP1>	REAL	표면 가공 이송 속도		
12	FZ	<_FFD>	REAL	깊이 절입 속도		
13		<_CDIR>	INT	밀링 방향	0 =	하향 절삭
					1 =	상향 절삭
14		<_VARI>	INT	가공 형식		
				UNITS:	가공	
					1 =	황삭
					2 =	정삭
					4 =	에지 정삭
					5 =	모따기
				TENS:	절입 유형 (황삭 및 정삭)	
					0 =	사전 드릴링, G0으로 절입 (포켓은 사전 가공)
					1 =	수직, G1로 절입
					2 =	헬리컬
				HUNDREDS:	예비용	
				THOUSANDS:		
					0 =	평면 단위
					1 =	헬리컬
15	DXY	<_MIDA>	REAL	최대 평면 절입, <_AMODE>, 0 = 0.8 x 공구 직경 참조		
16	Ø	<_AP1>	REAL	사전 가공 직경/반경 (inc)		
17	AZ	<_AD>	REAL	사전 가공 깊이 (inc)		
18	ER	<_RAD1>	REAL	헬리컬 절입에서 헬리컬 경로의 반경		
19	EP	<_DP1>	REAL	헬리컬 경로에서 절입 시 헬리컬 피치		
20		<_UMODE>	INT	예비용		
21	FS	<_FS>	REAL	모따기 너비 (inc)		
22	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs/inc), <_AMODE> 참조		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
23		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공/시작점 계산	
					0 =	호환성 모드
					1 =	일반 가공
				THOUSANDS:	예비용	
				TEN THOUSANDS:	전체 가공/재가공	
					0 =	호환성 모드 (전처럼 <_AP1> 및 <_AD> 처리)
					1 =	전체 가공
					2 =	사후 가공

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
24		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	이송 속도 유형: G 그룹 (G94/G95), 표면 및 깊이 이송 속도의 경우	
					0 =	호환성 모드
					1 =	싸이클 호출 이전과 같은 G 코드. 표면 및 깊이 이송 속도의 경우 G94/G95는 동일
				HUNDREDS:		
					0 =	호환성 모드 (반경으로 <_CDIAM>/<_AP1> 입력)
					1 =	직경으로 <_CDIAM>/<_AP1> 입력
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
25		<_AMODE>	INT	대체 모드		
				UNITS:	포켓 깊이 (Z1)	
					0 =	절대 (호환 모드)
					1 =	증분치
				TENS:	절입 너비 단위 (DXY)	
					0 =	mm
					1 =	공구 직경의 %
				HUNDREDS:	모따기를 위한 삽입 깊이 (ZFS)	
					0 =	절대치
					1 =	증분치

4.26.1.7 SLOT1- 일반 슬롯

구문

```
SLOT1 (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <WID>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>,
<CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FALD>, <_STA2>,
<_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<RTP>	REAL	후퇴 평면 (abs)
2	Z0	<RFP>	REAL	공구 축 원점 (abs)
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	슬롯 깊이 (abs)
5		<_DPR>	REAL	Z0를 기준으로 한 슬롯 깊이 (inc) (부호 없이 입력)
6		<NUM>	INT	슬롯 개수 = 1
7	L	<LENG>	REAL	슬롯 길이
8	W	<WID>	REAL	슬롯 너비
9	X0	<_CPA>	REAL	평면의 첫 번째 축 원점

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
10	Y0	<_CPO>	REAL	평면의 두 번째 축 원점		
11		<RAD>	REAL	예비용		
12	α	<STA1>	REAL	회전 각도		
13		<INDA>	REAL	예비용		
14	FZ	<FFD>	REAL	깊이 절입 속도		
15	F	<FFP1>	REAL	이송 속도		
16	DZ	<_MID>	REAL	최대 깊이 절입		
17		<CDIR>	INT	밀링 방향	0 =	하향 절삭
					1 =	상향 절삭
18	UXY	<_FAL>	REAL	평면 또는 슬롯 에지의 정삭 여유		
19		<VARI>	INT	가공 형식		
				UNITS:		
					0 =	예비용
					1 =	황삭
					2 =	정삭
					4 =	에지 정삭 (에지만 가공)
					5 =	모따기
				TENS:	접근	
					0 =	사전 드릴링, G0으로 절입 (슬롯은 사전 가공)
					1 =	수직, G1로 절입
					2 =	헬리컬
					3 =	오실레이션
				HUNDREDS:	예비용	
20	DZF	<_MIDF>	REAL	예비용		
21	FF	<FFP2>	REAL	예비용		
22	SF	<SSF>	REAL	예비용		
23	UZ	<_FALD>	REAL	정삭 여유, 깊이		
24	ER	<_STA2>	REAL	헬리컬 절입에서 헬리컬 경로의 반경		
	EW			오실레이션을 위한 최대 절입 각도		
25	EP	<_DP1>	REAL	나선에 대한 회전 당 삽입 깊이		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
26		<_UMODE>	INT	예비용		
27	FS	<_FS>	REAL	모따기를 위한 모따기 너비 (inc)		
28	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs/inc), <_AMODE> 참조		
29		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택 또는 시작점 계산만	
					1 =	일반 가공
				THOUSANDS:	원점 치수 지정, 슬롯 길이	
					0 =	중앙
					1 =	내부 좌측 +L
					2 =	내부 우측 -L
					3 =	좌측 에지 +L
					4 =	우측 에지 -L

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
30		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	소프트웨어 버전 식별	
					1 =	기능 확장 SLOT1
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단
31		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 Z1 (abs/inc)	
					0 =	호환성
					1 =	Z1 (inc)
					2 =	Z1 (abs)
				TENS:	예비용	
				HUNDREDS:	모따기를 위한 삽입 깊이 ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (inc)

참고

이전 소프트웨어 버전에는 없던 새로운 기능을 포함한 싸이클이 제공됩니다. 따라서 화면 형식에서 일부 파라미터 (<NUM>, <RAD>, <INDA>) 는 더 이상 표시되지 않습니다. "MCALL" 을 사용하고 원하는 위치 패턴 (예: HOLES2) 을 호출하여 한 위치에서 여러 슬롯을 프로그래밍할 수 있습니다.

4.26.1.8 SLOT2 - 원호 슬롯

구문

```
SLOT2(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <AFSL>, <WID>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>,
<CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FFCP>,
<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<RTP>	REAL	후퇴 평면 (abs)
2	Z0	<RFP>	REAL	공구 축 원점 (abs)
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	슬롯 깊이 (abs)
5		<_DPR>	REAL	Z0를 기준으로 한 슬롯 깊이 (inc) (부호 없이 입력)
6	N	<NUM>	INT	슬롯 개수
7	α1	<AFSL>	REAL	슬롯의 틈 각도
8	W	<WID>	REAL	슬롯 너비
9	X0	<_CPA>	REAL	원점 = 원호 중심점, 평면의 첫 번째 축
10	Y0	<_CPO>	REAL	원점 = 원호 중심점, 평면의 두 번째 축
11	R	<RAD>	REAL	원의 반경
12	α0	<STA1>	REAL	시작각
13	α2	<INDA>	REAL	증분 각도
14	FZ	<FFD>	REAL	깊이 절입 속도
15	F	<FFP1>	REAL	이송 속도
16	DZ	<_MID>	REAL	최대 깊이 절입

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
17		<CDIR>	INT	밀링 방향	0 =	하향 절삭
					1 =	상향 절삭
18	UXY	<_FAL>	REAL	평면 또는 슬롯 에지의 정삭 여유		
19		<VARI>	INT	가공 형식		
				UNITS:		
					0 =	전체 가공
					1 =	황삭
					2 =	정삭
					3 =	에지 정삭
					5 =	모따기
				TENS:		
					0 =	G0 라인으로 중간 포지셔닝
					1 =	원호 경로에 중간 포지셔닝
				HUNDREDS:	예비용	
				THOUSANDS:		
					0 =	호환 모드, <INDA> = 0이면 완전한 원, <INDA> <> 0이면 피치 원
					1 =	완전한 원
					2 =	피치 원호
20	DZF	<_MIDF>	REAL	예비용		
21		<FFP2>	REAL	예비용		
22		<SSF>	REAL	예비용		
23	FF	<_FFCP>	REAL	예비용		
24		<_UMODE>	INT	예비용		
25	FS	<_FS>	REAL	모따기 너비 (inc)		
26	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs/inc), <_AMODE> 참조		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
27		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택 또는 시작점 계산만	
					0 =	호환성 모드
					1 =	일반 가공
28		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	소프트웨어 버전 식별	
					1 =	소프트웨어 버전 2.5 기준 SLOT2 기능
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
29		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 Z1 (abs/inc)	
					0 =	호환성
					1 =	Z1 (inc)
					2 =	Z1 (abs)
				TENS:	예비용	
				HUNDREDS:	모따기를 위한 삽입 깊이 ZFS	
					0 =	ZFS (abs)
					1 =	ZFS (inc)

4.26.1.9 LONGHOLE - 연장된 홀

구문

```
LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>,
<_VARI>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<RTP>	REAL	후퇴 평면 (abs)
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	긴 홀 깊이 (abs)
5		<_DPR>	REAL	Z0를 기준으로 한 긴 홀 깊이 (inc) (부호 없이 입력)
6		<NUM>	INT	긴 홀 개수 = 1
7	L	<LENG>	REAL	긴 홀의 길이
8	X0	<_CPA>	REAL	평면의 원점 첫 번째 축:
9	Y0	<_CPO>	REAL	평면의 원점 두 번째 축:
10		<RAD>	REAL	예비용
11	α0	<STA1>	REAL	회전 각도

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
12		<INDA>	REAL	예비용		
13	FZ	<FFD>	REAL	깊이 절입 속도		
14	F	<FFP1>	REAL	이송 속도		
15	DZ	<MID>	REAL	최대 깊이 절입		
16		<_VARI>	INT	가공 형식		
				UNITS:	절입 유형	
					1 =	G1으로 수직으로
					3 =	오실레이션
				HUNDREDS:	예비용	
17		<_UMODE>	INT	예비용		
18		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택 또는 시작점 계산만	
					0 =	호환성 모드
					1 =	일반 가공
				THOUSANDS:	원점 치수 지정, 슬롯 길이	
					0 =	중앙
					1 =	내부 좌측 +L
					2 =	내부 우측 -L
					3 =	좌측 에지 +L
					4 =	우측 에지 -L

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
19		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	이송 속도 유형: G 그룹 (G94/G95), 표면 및 깊이 이송 속도의 경우	
					0 =	호환성 모드
					1 =	싸이클 호출 이전과 같은 G 코드. 표면 및 깊이 이송 속도의 경우 G94/G95는 동일
				HUNDREDS:	예비용	
				THOUSANDS:	소프트웨어 버전 식별	
					1 =	기능 확장 LONGHOLE (원점 치수 지정)
20		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 Z1 (abs/inc)	
					0 =	호환성
					1 =	Z1 (inc)
					2 =	Z1 (abs)

참고

이전 소프트웨어 버전에는 없던 새로운 기능을 포함한 싸이클이 제공됩니다. 따라서 화면 형식에서 일부 파라미터 (<NUM>, <RAD>, <INDA>) 는 더 이상 표시되지 않습니다. "MCALL" 을 사용하고 원하는 위치 패턴 (예: HOLES2) 을 호출하여 한 위치에서 여러 슬롯을 프로그래밍할 수 있습니다.

4.26.1.10 CYCLE60 - 조각

구문

```
CYCLE60(<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1		<_TEXT>	STRING [200]	조각할 텍스트 (최대 100자)
2	RP	<_RTP>	REAL	후퇴 평면 (abs)
3	Z0	<_RFP>	REAL	공구 축 원점 (abs)
4	SC	<_SDIS>	REAL	안전 거리 (기준 평면에 추가됨, 부호 없이 입력)
5	Z1	<_DP>	REAL	깊이 (abs), <_AMODE> 참조
6	Z1	<_DPR>	REAL	깊이 (inc), <_AMODE> 참조
7	X0	<_PA>	REAL	평면의 원점 첫 번째 축(abs) - 직각, <_VARI> 참조
	R			원점, 길이 (반경) - 극좌표, <_VARI> 참조
8	Y0	<_PO>	REAL	평면의 원점 두 번째 축(abs) - 직각, <_VARI> 참조
	α0			원점, 첫 번째 축 기준 각도 - 극좌표, <_VARI> 참조
9	α1	<_STA>	REAL	텍스트 방향, 첫 번째 축 기준 텍스트 라인의 각도, <_VARI> 참조
10	XM	<_CP1>	REAL	텍스트의 원호 중심, 평면의 첫 번째 축 (abs) - 직각, <_VARI> 참조
	LM			텍스트의 원호 중심, WNP 기준 길이 (반경) - 극좌표, <_VARI> 참조
11	YM	<_CP2>	REAL	텍스트의 원호 중심, 평면의 두 번째 축 (abs) - 직각, <_VARI> 참조
	αM			텍스트의 원호 중심, 첫 번째 축 기준 각도 - 극좌표, <_VARI> 참조
12	W	<_WID>	REAL	글자 높이 (부호 없이 입력)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
13	DX1 DX2	<_DF>	REAL	글자 간격 / 전체 너비, <_VARI> 참조
	α2			틈 각도, <_VARI> 참조
14	FZ	<_FFD>	REAL	깊이 절입 속도, <_DMODE> 참조
15	F	<_FFP1>	REAL	표면 가공 이송 속도

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
16		<_VARI>	INT	가공 (조각 텍스트 정렬 및 원점)		
				UNITS:	원점	
					0 =	직각
					1 =	극좌표
				TENS:	텍스트 정렬	
					0 =	한 라인으로 텍스트 정렬
					1 =	상향 커브 원호로 텍스트 정렬
					2 =	하향 커브 원호로 텍스트 정렬
				HUNDREDS:	예비용	
				THOUSANDS:	텍스트 원점, 수평	
					0 =	좌측
					1 =	중앙
					2 =	우측
				TEN THOUSANDS:	텍스트 원점, 수직	
					0 =	하단
					1 =	중앙
					2 =	상단
				HUNDRED THOUSANDS:	텍스트 길이	
					0 =	문자 크기
					1 =	전체 텍스트 너비 (직선 텍스트만)
					2 =	틈 각도 (원호 텍스트만 해당)
				ONE MILLION:	원호 중심	
					0 =	직각 (직교 좌표)
					1 =	극좌표
				TEN MILLIONS:	미러링	
					0 =	호환성
					1 =	미러링 ON
					2 =	미러링 OFF

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
17		<_CODEP>	INT	작성할 코드 페이지 번호 (현재는 1252만)		
18		<_UMODE>	INT	예비용		
19		_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택/시작점 계산만	
					0 =	호환성 모드
					1 =	일반 가공
20		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	이송 속도 유형: G 그룹 (G94/G95), 표면 및 깊이 이송 속도의 경우	
					0 =	호환성 모드
					1 =	싸이클 호출 이전과 같은 G 코드. 표면 및 깊이 이송 속도의 경우 G94/G95는 동일
21		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 (<_DP>, <_DPR>)	
					0 =	호환성
					1 =	증분치 (<_DPR>)
					2 =	절대치 (<_DP>)

4.26.1.11 CYCLE61 - 페이스 밀링

구문

```
CYCLE61(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<_RTP>	REAL	후퇴 평면 (abs)
2	Z0	<_RFP>	REAL	공구 축 원점, 소재 높이 (abs)
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	완성 부품의 높이 (abs/inc), <_AMODE> 참조
5	X0	<_PA>	REAL	첫 번째 축의 코너 지점 1 (abs)
6	Y0	<_PO>	REAL	두 번째 축의 코너 지점 1 (abs)
7	X1	<_LENG>	REAL	첫 번째 축의 코너 지점 2 (abs/inc), <_AMODE> 참조
8	Y1	<_WID>	REAL	두 번째 축의 코너 지점 2 (abs/inc), <_AMODE> 참조
9	DZ	<_MID>	REAL	최대 깊이 절입
10	DXY	<_MIDA>	REAL	최대 평면 절입, 단위에 대해서는 <_AMODE> 참조
11	UZ	<_FALD>	REAL	정삭 여유, 깊이
12	F	<_FFP1>	REAL	가공 속도

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
13		<_VARI>	INT	가공 형식		
				UNITS:	가공	
					1 =	황삭
					2 =	정삭
				TENS:	가공 방향	
					1 =	첫 번째 축과 평행, 한 방향으로
					2 =	두 번째 축과 평행, 한 방향으로
					3 =	첫 번째 축과 평행, 여러 방향으로
					4 =	두 번째 축과 평행, 여러 방향으로
14		<_LIM>	INT	한계		
				UNITS:	첫 번째 축 마이너스 방향 제한	
					0 =	아니오
					1 =	예
				TENS:	첫 번째 축 플러스 방향 제한	
					0 =	아니오
					1 =	예
				HUNDREDS:	두 번째 축 마이너스 방향 제한	
					0 =	아니오
					1 =	예
				THOUSANDS:	두 번째 축 플러스 방향 제한	
					0 =	아니오
					1 =	예

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
15		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
16		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 (<_DP>)	
					0 =	절대치
					1 =	증분치
				TENS:	평면 절입 단위 (<_MIDA>)	
					0 =	mm
					1 =	공구 직경의 %
				HUNDREDS:	예비용	
				THOUSANDS:	표면 길이	
					0 =	증분치
					1 =	절대치
				TEN THOUSANDS:	표면 너비	
					0 =	증분치
					1 =	절대치

4.26.1.12 CYCLE62 - 형상 호출

구문

```
CYCLE62(<_KNAME>,<_TYPE>, <_LAB1>, <_LAB2>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	PRG/CON	<_KNAME>	STRING [140]	_TYPE = 2에서는 형상 이름 또는 서브프로그램 이름을 프로그래밍할 필요가 없습니다.		
2		<_TYPE>	INT	형상 입력 결정		
					0 =	서브프로그램
					1 =	형상 이름
					2 =	라벨
					3 =	서브프로그램의 라벨
3	LAB1	<_LAB1>	STRING[32]	레이블 1, 형상 시작		
4	LAB2	<_LAB2>	STRING[32]	레이블 2, 형상 끝		

4.26.1.13 CYCLE63 – 형상 포켓 밀링 / 형상 포켓 잔삭 가공 / 형상 스피곳 밀링 / 형상 스피곳 잔삭 가공

구문

```
CYCLE63(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_FZ>,
<_DXY>, <_DZ>, <_UXY>, <_UZ>, <_CDIR>, <_XS>, <_YS>, <_ER>,
<_EP>, <_EW>, <_FS>, <_ZFS>, <_TR>, <_DR>, <_UMODE>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	PRG	<_PRG>	STRING [100]	제거 프로그램 이름		
2		<_VARI>	INT	가공 형식		
				UNITS:	가공 공정	
					1 =	황삭
					3 =	베이스 정삭
					4 =	에지 정삭
					5 =	모따기
				TENS:	절입 유형	
					0 =	중앙 삽입
					1 =	헬리컬 절입
					2 =	왕복 절입
				HUNDREDS:	예비용	
				THOUSANDS:	도피 모드	
					0 =	후퇴 평면으로 도피
					1 =	원점 + 안전 거리로 도피
				TEN THOUSANDS:	황삭 및 정삭 베이스의 시작점	
					0 =	자동
					1 =	수동
3	RP	<_RP>	REAL	후퇴 평면 (abs)		
4	Z0	<_Z0>	REAL	공구 축 원점 (abs)		
5	SC	<_SC>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
6	Z1	<_Z1>	REAL	최종 깊이 (<_AMODE> UNITS 참조)		
7	F	<_F>	REAL	황삭/정삭 과정에서 평면 이송 속도		
8	FZ	<_FZ>	REAL	깊이 절입 속도		
9	DXY	<_DXY>	REAL	평면 절입 - 단위 (<_AMODE> TENS 참조)		
10	DZ	<_DZ>	REAL	깊이 절입		
11	UXY	<_UXY>	REAL	정삭 여유, 평면		
12	UZ	<_UZ>	REAL	정삭 여유, 깊이		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
13		<_CDIR>	INT	밀링 방향	0 =	하향 절삭
					1 =	상향 절삭
14	XS	<_XS>	REAL	시작점 X, 절대		
15	YS	<_YS>	REAL	시작점 Y, 절대		
16	ER	<_ER>	REAL	헬리컬 절입: 반경		
17	EP	<_EP>	REAL	헬리컬 절입: 피치		
18	EW	<_EW>	REAL	왕복 절입: 최대 절입 각도		
19	FS	<_FS>	REAL	모따기를 위한 모따기 너비 (inc)		
20	ZFS	<_ZFS>	REAL	모따기할 때 공구 팁의 삽입 깊이 (<_AMODE> HUNDREDS 참조)		
21	TR	<_TR>	STRING[32]	잔삭 가공 시 기준 공구 이름		
22	DR	<_DR>	INT	잔삭 가공 시 기준 공구 D 번호		
23		<_UMODE>	INT	예비용		
24		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택/시작점 계산만	
					0 =	일반 가공 (호환 모드는 필요 없음)
					1 =	일반 가공
					2 =	예비용

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
25		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	예비용	
				HUNDREDS:	테크놀로지 모드	
					1 =	포켓
					2 =	스피곳
				THOUSANDS:	잔삭 가공	
					0 =	아니오
					1 =	예
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단
				HUNDRED THOUSANDS:	자동 프로그램 이름	
					0 =	아니오
					1 =	예

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
26		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 (Z1)	
					0 =	절대 (호환 모드)
					1 =	증분치
				TENS:	평면 절입 단위 (DXY)	
					0 =	mm
					1 =	공구 직경의 %
				HUNDREDS:	모따기를 위한 삽입 깊이 (ZFS)	
					0 =	절대치
					1 =	증분치
				THOUSANDS:	---	예비용

4.26.1.14 CYCLE64 - 형상 포켓 예비 드릴링

구문

```
CYCLE64(<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>,
<_DXY>, <_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	PRG	<_PRG>	STRING [100]	드릴링/센터링 프로그램의 이름		
2		<_VARI>	INT	가공 형식		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	도피 모드	
					0 =	후퇴 평면으로 도피
					1 =	원점 + 안전 거리로 도피

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
3	RP	<_RP>	REAL	후퇴 평면 (abs)		
4	Z0	<_Z0>	REAL	원점 (abs)		
5	SC	<_SC>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
6	Z1	<_Z1>	REAL	드릴링/센터링 깊이 (<_AMODE> UNITS 참조)		
7	F	<_F>	REAL	드릴링/센터링 이송 속도		
8	DXY	<_DXY>	REAL	평면 절입 - 단위 (<_AMODE> TENS 참조)		
9	UXY	<_UXY>	REAL	정삭 여유, 평면		
10	UZ	<_UZ>	REAL	정삭 여유, 깊이		
11		<_CDIR>	INT	밀링 방향	0 =	하향 절삭
					1 =	상향 절삭
12	TR	<_TR>	STRING[20]	기준 공구 이름		
13	DR	<_DR>	INT	기준 공구 D 번호		
14		<_UMODE>	INT	예비용		
15		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택/시작점 계산만	
					0 =	일반 가공 (호환 모드는 필요 없음)
					1 =	일반 가공
					2 =	예비용

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
25		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	테크놀로지 모드	
					1 =	사전 드릴링
					2 =	센터링
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	---	예비용
				HUNDRED THOUSANDS:	자동 프로그램 이름	
					0 =	아니오
					1 =	예
26		<_AMODE>	INT	대체 모드		
				UNITS:	드릴링/센터링 깊이 Z1	
					0 =	절대 (호환 모드)
					1 =	증분치
				TENS:	평면 절입 단위 (DXY)	
					0 =	mm
					1 =	공구 직경의 %

4.26.1.15 CYCLE70 - 나사 밀링

구문

```
CYCLE70(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>,
<_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>,
<_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	RP	<_RTP>	REAL	후퇴 평면 (abs)		
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)		
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
4	Z1	<_DP>	REAL	나사 길이 (abs/inc), <_AMODE> 참조 홀의 베이스에서 런아웃 고려 (최소 1/2 피치)		
5	Ø	<_DIATH>	REAL	나사의 호칭경		
6	H1	<_H1>	REAL	나사 깊이		
7	U	<_FAL>	REAL	정삭 여유		
8	P	<_PIT>	REAL	피치 (<_PITA> 선택: mm, inch, MODULE, threads/inch)		
9	NT	<_NT>	INT	공구 팁에서 톱니의 수 공구 길이는 항상 하단 절삭날을 기준으로 합니다.		
10	DXY	<_MID>	REAL	절삭당 최대 절입 <_MID> > <_H1>: 한 번에 전체 절삭		
11	F	<_FFR>	REAL	밀링 피드		
12		<_TYPTH>	INT	나사 유형	0 =	내경 나사
					1 =	외경 나사
13	X0	<_PA>	REAL	원호 중심 첫 번째 축 (abs)		
14	Y0	<_PO>	REAL	원호 중심 두 번째 축 (abs)		
15	αS	<_NSP>	REAL	시작 각도 (멀티 스타트 나사)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
16		<_VARI>	INT	가공 형식		
				UNITS:		
					1 =	황삭
					2 =	정삭
				TENS:		
					1 =	하향
					2 =	상향
				HUNDREDS:		
					0 =	오른 나사
					1 =	왼 나사
17		<_PITA>	INT	나사 피치 평가		
					0 =	호환성 모드
					1 =	피치 (mm)
					2 =	피치 (인치당 스레드, TPI)
					3 =	피치 (인치)
					4 =	피치 (MODULE에 따름)
18		<_PITM>	STRING[15]	피치 입력을 위한 마커로 사용되는 문자열 (인터페이스만 해당)		
19		<_PTAB>	STRING[20]	나사 테이블의 문자열 ("", "ISO", "BSW", "BSP", "UNC") (인터페이스만 해당)		
20		<_PTABA>	STRING[20]	나사 테이블에서 선택하기 위한 문자열 (예: "M 10", "M 12", ...) (인터페이스만 해당)		
21		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공/시작점 계산	
					0 =	호환성 모드
					1 =	일반 가공

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
22		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
23		<_AMODE>	INT	대체 모드		
				UNITS:	나사 길이 (<_DP>)	
					0 =	절대치
					1 =	증분치

4.26.1.16 CYCLE72 - 경로 밀링

구문

```
CYCLE72(<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>,<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1		<_KNAME>	STRING [141]	형상 서브프로그램 이름
2	RP	<_RTP>	REAL	후퇴 평면 (abs)
3	Z0	<_RFP>	REAL	공구 축 원점 (abs)
4	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
5	Z1	<_DP>	REAL	종점, 최종 깊이 (abs/inc), <_AMODE> 참조		
6	DZ	<_MID>	REAL	최대 깊이 절입 (inc, 부호 없이 입력)		
7	UXY	<_FAL>	REAL	정삭 여유, 평면 (inc), 에지 형상의 공차		
8	UZ	<_FALD>	REAL	정삭 여유 깊이 (inc), 베이스에서 공차 (부호 없이 입력)		
9	FX	<_FFP1>	REAL	형상에서의 이송 속도		
10	FZ	<_FFD>	REAL	깊이 절입 이송 속도 (또는 공간 절입)		
11		<_VARI>	INT	가공 형식		
				UNITS:	가공	
					1 =	황삭
					2 =	정삭
					5 =	모따기
				TENS:		
					0 =	G0을 사용한 중간 경로
					1 =	G1을 사용한 중간 경로
				HUNDREDS:	형상 끝에서 후퇴	
					0 =	형상의 끝에서 원점까지 후퇴
					1 =	형상의 끝에서 원점 + <_SDIS>까지 후퇴
					2 =	형상 끝에서 <_SDIS> 만큼 후퇴
					3 =	형상 끝에서 후퇴 없음, 형상 피드로 다음 시작점에 접근
				THOUSANDS:	예비용	
				TEN THOUSANDS:	형상 가공	
					0 =	형상을 전방으로 가공
					1 =	형상을 후방으로 가공 백워드 가공 제한: • 최대 170개 형상 요소 (모따기 또는 라운딩 포함) • (X/Y) 및 F 평면의 값만 평가됩니다.

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
12		<_RL>	INT	가공 방향		
					40 =	형상 중심 (G40, 접근 및 후퇴: 직선 또는 수직)
					41 =	형상 좌측 (G41, 접근 및 후퇴: 직선 또는 원호)
					42 =	형상 우측 (G42, 접근 및 후퇴: 직선 또는 원호)
13		<_AS1>	INT	형상 접근 이동		
				UNITS:		
					1 =	직선
					2 =	4분원
					3 =	반원
					4 =	수직으로 접근 및 후퇴
				TENS:		
					0 =	마지막 이동, 평면에서
					1 =	마지막 이동, 공간에서
14	L1	<_LP1>	REAL	접근 경로 또는 접근 반경 (inc, 부호 없이 입력)		
15	FZ	<_FF3>	REAL	중간 경로의 이송 속도 (형상의 경우 G94/G95)		
16		<_AS2>	INT	형상 접근 이동 (수직 접근/후퇴 아님)		
				UNITS:		
					1 =	직선
					2 =	4분원
					3 =	반원
				TENS:		
					0 =	마지막 이동, 평면에서
					1 =	마지막 이동, 공간에서
17	L2	<_LP2>	REAL	후퇴 경로 또는 후퇴 반경 (inc, 부호 없이 입력)		
18		<_UMODE>	INT	예비용		
19	FS	<_FS>	REAL	모따기 너비 (inc)		
20	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs/inc), <_AMODE> 참조		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
21		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택/시작점 계산만	
					0 =	호환성 모드
					1 =	일반 가공
22		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	이송 속도 유형: G 그룹 (G94/G95), 표면 및 깊이 이송 속도의 경우	
					0 =	호환성 모드
					1 =	싸이클 호출 이전과 같은 G 코드. 표면 및 깊이 이송 속도의 경우 G94/G95는 동일
				THOUSANDS:		
					0 =	호환성 모드: 형상 이름은 <_KNAME>에 있습니다.
					1 =	형상 이름이 CYCLE62에서 프로그래밍되어 _SC_CONT_NAME으로 전송됩니다.

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
23		<_AMODE>	INT	대체 모드		
				UNITS:	종점 Z1 (<_DP>)	
					0 =	절대 (호환 모드)
					1 =	증분치
				TENS:	평면 절입 단위	
					0 =	mm, 인치
					1 =	예비용
				HUNDREDS:	모따기를 위한 삽입 깊이 (<_ZFS>)	
					0 =	절대치
					1 =	증분치

참고

다음 전송 파라미터가 간접적으로 (파라미터로서) 프로그래밍된 경우에는 화면 형식이 리셋되지 않습니다.

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

4.26.1.17 CYCLE76 - 사각 스피곳

구문

```
CYCLE76(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_LENG>, <_WID>,
<_CRAD>, <_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>,
<_FFD>, <_CDIR>, <_VARI>, <_AP1>, <_AP2>, <_FS>, <_ZFS>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<_RTP>	REAL	후퇴 평면 (abs)
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	스피곳 깊이 (abs)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
5		<_DPR>	REAL	Z0를 기준으로 한 스피곳 깊이 (inc) (부호 없이 입력)		
6	L	<_LENG>	REAL	스피곳 길이, <_GMODE> 참조 (부호 없이 입력)		
7	W	<_WID>	REAL	스피곳 너비, <_GMODE> 참조 (부호 없이 입력)		
8	R	<_CRAD>	REAL	스피곳 코너 반경 (부호 없이 입력)		
9	X0	<_PA>	REAL	평명의 첫 번째 축에서 스피곳의 원점 (abs)		
10	Y0	<_PO>	REAL	평명의 두 번째 축에서 스피곳의 원점 (abs)		
11	α0	<_STA>	REAL	회전 각도, 세로 축 (L) 과 평면 첫 번째 축 사이의 각도		
12	DZ	<_MID>	REAL	최대 깊이 절입 (inc, 부호 없이 입력)		
13	UXY	<_FAL>	REAL	정삭 여유, 평면 (inc), 에지 형상의 공차		
14	UZ	<_FALD>	REAL	정삭 여유 깊이 (inc), 베이스에서 공차 (부호 없이 입력)		
15	FX	<_FFP1>	REAL	형상에서의 이송 속도		
16	FZ	<_FFD>	REAL	깊이 절입 속도		
17		<_CDIR>	INT	밀링 방향 (부호 없이 입력)		
				UNITS:		
					0 =	하향 절삭
					1 =	상향 절삭
18		<_VARI>	INT	가공		
				UNITS:		
					1 =	황삭
					2 =	정삭
					5 =	모따기
19	L1	<_AP1>	REAL	소재 스피곳 길이		
20	W1	<_AP2>	REAL	소재 스피곳 너비		
21	FS	<_FS>	REAL	모따기 너비 (inc)		
22	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs, inc), <_AMODE> 참조		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
23		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택 또는 시작점 계산만	
					0 =	호환성 모드
					1 =	일반 가공
				THOUSANDS:	중심 또는 코너 기준으로 스피곳 치수 지정	
					0 =	호환성 모드
					1 =	중심을 통한 치수 지정
					2 =	코너 지점의 치수 지정, 스피곳 +L +W
					3 =	코너 지점의 치수 지정, 스피곳 -L +W
					4 =	코너 지점의 치수 지정, 스피곳 +L -W
					5 =	코너 지점의 치수 지정, 스피곳 -L -W
				TEN THOUSANDS:	전체 가공 또는 재가공	
					0 =	호환성 모드
					1 =	전체 가공
					2 =	사후 가공

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
24		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	---	예비용
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단
25		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 Z1 (DP)	
					0 =	호환성
					1 =	증분치
					2 =	절대치
				TENS:	예비용	
				HUNDREDS:	모따기를 위한 삽입 깊이 (ZFS)	
					0 =	절대치
					1 =	증분치

4.26.1.18 CYCLE77 - 원형 스피곳

구문

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>,
<_VARI>, <_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	RP	<_RTP>	REAL	후퇴 평면 (abs)		
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)		
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
4	Z1	<_DP>	REAL	스피곳 깊이 (abs)		
5		<_DPR>	REAL	Z0를 기준으로 한 스피곳 깊이 (inc) (부호 없이 입력)		
6	Ø	<_CDIAM>	REAL	스피곳 직경 (부호 없이 입력)		
7	X0	<_PA>	REAL	평명의 첫 번째 축에서 스피곳의 원점 (abs)		
8	Y0	<_PO>	REAL	평명의 두 번째 축에서 스피곳의 원점 (abs)		
9	DZ	<_MID>	REAL	최대 깊이 절입 (inc, 부호 없이 입력)		
10	UXY	<_FAL>	REAL	정삭 여유, 평면 (inc), 에지 형상의 공차		
11	UZ	<_FALD>	REAL	정삭 여유 깊이 (inc), 베이스에서 공차 (부호 없이 입력)		
12	FX	<_FFP1>	REAL	형상에서의 이송 속도		
13	FZ	<_FFD>	REAL	깊이 절입 속도		
14		<_CDIR>	INT	밀링 방향 (부호 없이 입력)		
				UNITS:		
					0 =	하향 절삭
					1 =	상향 절삭
15		<_VARI>	INT	가공 형식		
				UNITS:	가공	
					1 =	최종 가공 공차까지 황삭
					2 =	정삭 (여유 X/Y/Z=0)
					5 =	모따기
16	Ø1	<_AP1>	REAL	소재 스피곳 직경		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
17	FS	<_FS>	REAL	모따기 너비 (inc)		
18	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs/inc), <_AMODE> 참조		
19		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택/시작점 계산만	
					0 =	호환성 모드
					1 =	일반 가공
				THOUSANDS:	예비용	
				TEN THOUSANDS:	전체 가공/재가공	
					0 =	호환성 모드 (전처럼 <_AP1> 처리)
					1 =	전체 가공
					2 =	사후 가공
20		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	---	예비용
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
21		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 Z1 (DP)	
					0 =	절대 (호환 모드)
					1 =	증분치
					2 =	절대치
				TENS:	예비용	
				HUNDREDS:	모따기를 위한 삽입 깊이 (ZFS)	
					0 =	절대치
					1 =	증분치

4.26.1.19 CYCLE78 - 드릴 나사 밀링

구문

```
CYCLE78(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_ADPR>, <_FDPR>,
<_LDPR>, <_DIAM>, <_PIT>, <_PITA>, <_DAM>, <_MDEP>, <_VARI>,
<_CDIR>, <_GE>, <_FFD>, <_FRDP>, <_FFR>, <_FFP2>, <_FFA>,
<_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<_RTP>	REAL	후퇴 평면 (abs)
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	최종 드릴링 깊이 (abs/inc), <_AMODE> 참조
5		<_ADPR>	REAL	드릴링 속도 (inc) 를 줄인 사전 드릴링 깊이. <_VARI> TEN THOUSANDS에서 유효함
6	D	<_FDPR>	REAL	최대 깊이 절입 (inc) D ≥ Z1 ⇒ 최종 드릴링 깊이까지 1회 절입 D < Z1 ⇒ 다중 절입 및 칩 제거를 포함한 심공 드릴링 싸이클
7	ZR	<_LDPR>	REAL	FR 피드로 관통 드릴링 (inc) 시 나머지 드릴링 깊이
8	Ø	<_DIAM>	REAL	나사의 호칭경

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
9	P	<_PIT>	REAL	숫자 값 피치		
10		<_PITA>	INT	나사 피치 P 평가		
					1 =	피치 (mm/rev)
					2 =	피치 (thread/inch)
					3 =	피치 (inch/rev)
					4 =	피치 (MODULE에 따름)
11	DF	<_DAM>	REAL	각 추가 절입 (하강) 에 대한 절대치/백분율, <_AMODE> 참조		
12	V1	<_MDEP>	REAL	최소 절입 (inc), 하강에서만 활성 상태		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
13		<_VARI>	INT	가공 형식		
				UNITS:	예비용	
				TENS:	나사 밀링 전 칩 제거	
					0 =	나사 밀링 전에 칩 제거 안 함 (최종 드릴링 깊이에서만 활성)
					1 =	나사 밀링 전에 칩 제거 (최종 드릴링 깊이에서만 활성)
				HUNDREDS:	오른손/왼손 나사	
					0 =	오른 나사
					1 =	왼 나사
				THOUSANDS:	드릴링 속도를 사용한 잔여 드릴링 깊이	
					0 =	드릴링 속도 FR에서 남은 드릴링 깊이 없음
					1 =	드릴링 속도 FR을 사용한 잔여 드릴링 깊이
				TEN THOUSANDS:	감소된 속도로 사전 드릴링	
					0 =	감소된 속도로 사전 드릴링 안 함
					1 =	이송 속도 감소를 통한 사전 드릴링 사전 드릴링 속도 = 0.3 F1, F1 < 0.15 mm/rev인 경우 사전 드릴링 속도 = 0.1 mm/rev, F1 ≥ 0.15 mm/rev인 경우
14		<_CDIR>	INT	밀링 방향	0 =	하향 절삭
					1 =	상향 절삭
					4 =	상향 절삭 + 하향 절삭 (복합 황삭 + 정삭)
15	Z2	<_GE>	REAL	나사 밀링 전의 후퇴 거리 (inc)		
16	F1	<_FFD>	REAL	드릴링 속도 (mm/min, in/min 또는 mm/rev)		
17	FR	<_FRDP>	REAL	남은 드릴링 깊이에 대한 드릴링 속도 (mm/min 또는 mm/rev)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
18	F2	<_FFR>	REAL	나사 밀링 속도 (mm/min 또는 mm/tooth)		
19	FS	<_FFP2>	REAL	<_CDIR> =4에 대한 정삭 속도 (mm/min 또는 mm/tooth)		
20		<_FFA>	INT	이송 속도의 평가		
				UNITS:	드릴링 피드 F1	
				TENS:	남은 드릴링 깊이 FR에 대한 드릴링 속도	
				HUNDREDS:	나사 밀링 F2에 대한 이송 속도	
				THOUSANDS:	정삭 속도 FS	
21		<_PITM>	STRING[15]	피치 입력을 위한 마커로 사용되는 문자열 (인터페이스만 해당)[1)]		
22		<_PTAB>	STRING[20]	나사 테이블의 문자열 ("", "ISO", "BSW", "BSP", "UNC") (인터페이스만 해당)[1)]		
23		<_PTABA>	STRING[20]	나사 테이블에서 선택하기 위한 문자열 (예: "M 10", "M 12", ...) (인터페이스만 해당)[1)]		
24		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가), 예비		
25		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
26		<_AMODE>	INT	대체 모드		
				UNITS:	드릴링 깊이 = 최종 드릴링 깊이 Z1 (abs/inc)	
					0 =	절대치
					1 =	증분치
				TENS:	각 추가 절입 (하강) 에 대한 절대치/백분율 DF	
					0 =	절대치
					1 =	백분율 (0.001 ~ 100%)

참고

[1)] 파라미터 21, 22 및 23은 화면 형식 나사 테이블에서 나사 선택에만 사용됩니다. 싸이클 런타임에서는 싸이클 정의를 통해 나사 테이블에 액세스할 수 없습니다.

4.26.1.20 CYCLE79 - 다각형

구문

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<_RTP>	REAL	후퇴 평면 (abs)
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	다각형 깊이 (abs/inc), <_AMODE> 참조
5	N	<_NUM>	INT	면 개수 (1...n)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
6	SW/L	<_SWL>	REAL	평판을 가로지르는 너비 및 에지 길이 (<_VARI>에 따라) (평판을 가로지르는 너비의 경우 "SW", 에지 길이의 경우 "L") 평판을 가로지르는 너비 (에지 수가 짝수이거나 단일 에지인 경우만 해당)		
7	X0	<_PA>	REAL	스피곳 원점, 첫 번째 축 (abs)		
8	Y0	<_PO>	REAL	스피곳 원점, 두 번째 축 (abs)		
9	α0	<_STA>	REAL	회전 각도, 첫 번째 축 기준으로 에지 중심 (X축)		
10	R1/FS1	<_RC>	REAL	<_NUM> > 2일 때의 코너 라운딩 (반경/모따기, <_AMODE> 참조) (inc, 부호 없이 입력) (반경의 경우 "R1", 모따기의 경우 "FS1")		
11	∅	<_AP1>	REAL	가공되지 않은 스피곳 직경		
12	DXY	<_MIDA>	REAL	최대 절입 너비 (단위에 대해서는 <_AMODE> 참조)		
13	DZ	<_MID>	REAL	최대 깊이 절입		
14	UXY	<_FAL>	REAL	정삭 여유, 평면		
15	UZ	<_FALD>	REAL	정삭 여유, 깊이		
16	F	<_FFP1>	REAL	가공 속도		
17		<_CDIR>	INT	밀링 방향	0 =	하향 절삭
					1 =	상향 절삭
18		<_VARI>	INT	가공 형식		
				UNITS:	가공	
					1 =	황삭
					2 =	정삭
					3 =	에지 정삭
					5 =	모따기
				TENS:	평판을 가로지르는 너비 및 에지 길이	
					0 =	평면 너비
					1 =	에지 길이
19	FS	<_FS>	REAL	모따기 너비 (inc)		
20	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs/inc), <_AMODE> 참조		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
21		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택 또는 시작점 계산만	
					1 =	일반 가공
22		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	---	예비용
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
23		<_AMODE>	INT	대체 모드		
				UNITS:	최종 깊이 (<_DP>)	
					0 =	절대치
					1 =	증분치
				TENS:	평면 절입 단위 (<_MIDA>)	
					0 =	mm
					1 =	공구 직경의 %
				HUNDREDS:	모따기를 위한 삽입 깊이 (<_ZFS>)	
					0 =	절대치
					1 =	증분치
				THOUSANDS:	코너 라운딩 (<_RC>)	
					0 =	반경
					1 =	모따기

4.26.1.21 CYCLE81 - 드릴링, 센터링

구문

```
CYCLE81(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<RTP>	REAL	후퇴 평면 (abs)
2	Z0	<RFP>	REAL	원점 (abs)
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1/Ø	<DP>	REAL	드릴링 깊이 (abs)/ 센터링 직경 (abs), <_GMODE> 참조
5	Z1	<DPR>	REAL	드릴링 깊이 (inc)
6	DT	<DTB>	REAL	최종 드릴링 깊이에서 드웰 시간, <_AMODE> 참조

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
7		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	깊이/직경을 기준으로 한 센터링	
					0 =	호환성, 깊이
					1 =	직경
8		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
9		<_AMODE>	INT	대체 모드		
				UNITS:	드릴링 깊이 Z1 (abs/inc)	
					0 =	호환성, DP/DPR 프로그래밍에서
					1 =	증분치
					2 =	절대치
				TENS:	최종 드릴링 깊이에서 드웰 시간 DT (초/회전)	
					0 =	호환성, DTB 부호에서 (> 0초 또는 < 0회전)
					1 =	초
					2 =	회전

4.26.1.22 CYCLE82 - 드릴링, 카운터보링

구문

```
CYCLE82(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	RP	<RTP>	REAL	후퇴 평면 (abs)		
2	Z0	<RFP>	REAL	원점 (abs)		
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
4	Z1	<DP>	REAL	드릴링 깊이 (abs), <_AMODE> 참조		
5	Z1	<DPR>	REAL	드릴링 깊이 (inc), <_AMODE> 참조		
6	DT	<DTB>	REAL	최종 드릴링 깊이에서 드웰 시간, <_AMODE> 참조		
7		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	팁/직경을 기준으로 한 드릴링 깊이	
					0 =	호환성, 팁
					1 =	직경

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
8		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	예비용	
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 베이직

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
9		<_AMODE>	INT	대체 모드		
				UNITS:	드릴링 깊이 Z1 (abs/inc)	
					0 =	호환성, DP/DPR 프로그래밍에서
					1 =	증분치
					2 =	절대치
				TENS:	최종 드릴링 깊이에서 드웰 시간 DT (초/회전)	
					0 =	호환성, DT 부호에서 (> 0초 또는 < 0회전)
					1 =	초
					2 =	회전
				HUNDREDS:	드릴링 깊이 ZA (abs/inc)	
					0 =	증분치
					1 =	절대치
				THOUSANDS:	사전 드릴링 이송 속도 평가	
					0 =	드릴링 이송 속도의 %로 표시
					1 =	F/min
					2 =	F/rev
				TEN THOUSANDS:	남은 드릴링 깊이 ZD (abs/inc)	
					0 =	증분치
					1 =	절대치
				HUNDRED THOUSANDS:	남은 드릴링 이송 속도 평가	
					0 =	드릴링 이송 속도의 %로 표시
					1 =	F/min
					2 =	F/rev

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
10		<_VARI>	INT	사전 드릴링/관통 드릴링 가공 유형		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	관통 드릴링	
					0 =	관통 드릴링 "아니오"
					1 =	관통 드릴링 "예"
				TEN THOUSANDS:	사전 드릴링	
					0 =	사전 드릴링 "아니오"
					1 =	사전 드릴링 "예"
11	ZA	<S_ZA>	REAL	원점 기준 증분 사전 드릴링 깊이 또는 절대 사전 드릴링 깊이 (<_AMODE> HUNDREDS 참조)		
12	FA	<S_FA>	REAL	값 또는 %로 표시되는 사전 드릴링 이송 속도 (<_AMODE> THOUSANDS와 연계)		
13	ZD	<S_ZD>	REAL	최종 드릴링 깊이 대비 남은 증분 드릴링 깊이 또는 절대 드릴링 깊이 (<_AMODE> TEN THOUSANDS 참조)		
14	FD	<S_FD>	REAL	값 또는 %로 표시되는 남은 드릴링 이송 속도 (<_AMODE> HUNDRED THOUSANDS와 연계)		

4.26.1.23 CYCLE83 - 심공 드릴링 1

구문

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>,
<_DAM>, <DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>,
<_DTD>, <_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<RTP>	REAL	후퇴 평면 (abs)
2	Z0	<RFP>	REAL	원점 (abs)
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
4	Z1	<DP>	REAL	최종 드릴링 깊이 (abs), <_AMODE> 참조		
5	Z1	<DPR>	REAL	최종 드릴링 깊이 (inc), <_AMODE> 참조		
6	D	<FDEP>	REAL	1차 드릴링 깊이 (abs), <_AMODE> 참조		
7	D	<FDPR>	REAL	1차 드릴링 깊이 (inc), <_AMODE> 참조		
8	DF	<_DAM>	REAL	각 추가 절입에 대한 하강값/백분율, <_AMODE> 참조		
9	DTB	<DTB>	REAL	드릴링 깊이에서의 드웰 시간, <_AMODE> 참조		
10	DTS	<DTS>	REAL	시작점에서 드웰 시간 (칩 제거에만 해당), <_AMODE> 참조		
11	FD1	<FRF>	REAL	첫 번째 절입의 이송 속도에 대한 백분율, <_AMODE> 참조		
12		<VARI>	INT	가공 형식		
				UNITS:	칩 분쇄/칩 제거	
					0 =	칩 분쇄
					1 =	칩 제거
13		<_AXN>	INT	공구 축		
					0 =	세 번째 기하 축
					1 =	첫 번째 기하 축
					2 =	두 번째 기하 축
					> 2	세 번째 기하 축
14	V1	<_MDEP>	REAL	최소 절입 (하강 백분률에만 해당)		
15	V2	<_VRT>	REAL	각 가공 단계 후의 후퇴 거리 (칩 분쇄에만 해당)		
					> 0	가변 후퇴 거리
					0 =	디폴트 값 1 mm
16	DT	<_DTD>	REAL	최종 드릴링 깊이에서 드웰 시간, <_AMODE> 참조		
17	V3	<_DIS1>	REAL	제한 거리 (칩 제거에만 해당), <_AMODE> 참조		
18		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	팁/직경을 기준으로 한 드릴링 깊이	
					0 =	팁
					1 =	직경

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
19		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	---	예비용
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
20		<_AMODE>	INT	대체 모드		
				UNITS:	드릴링 깊이 = 최종 드릴링 깊이 Z1 (abs/inc)	
					0 =	호환성, <DP>/<DPR> 프로그래밍에서
					1 =	증분치
					2 =	절대치
				TENS:	드릴링 깊이에서 드웰 시간 DTB (초/회전)	
					0 =	호환성, DTB 부호에서 (> 0초 또는 < 0회전)
					1 =	초
					2 =	회전
				HUNDREDS:	DTS의 시작점에서 드웰 시간 (초/회전)	
					0 =	호환성, DTS 부호에서 (> 0초 또는 < 0회전)
					1 =	초
					2 =	회전
				THOUSANDS:	최종 드릴링 깊이에서 드웰 시간 DTD (초/회전)	
					0 =	호환성, DTD 부호에서 (> 0초 또는 < 0회전)
					1 =	초
					2 =	회전
				TEN THOUSANDS:	1차 드릴링 깊이 D (abs/inc)	
					0 =	호환성, <FDEPF>/<DPR> 프로그래밍에서
					1 =	증분치
					2 =	절대치
				HUNDRED THOUSANDS:	각 추가 절입에 대한 하강값/백분율 <_DAM>	

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
					0 =	호환성, <_DAM> 부호에서 (> 0 하강값 또는 < 0 계수 0.001 ~ 1.0)
					1 =	하강값
					2 =	백분율 (0.001 ~ 100%)
				ONE MILLION:	제한 거리 V3 자동/수동	
					0 =	호환성, <_DIS1> 부호에서 (= 0 자동 또는 > 0 수동)
					1 =	자동 (싸이클에서 계산)
					2 =	수동 (프로그래밍된 값)
				TEN MILLIONS:	1차 절입 <FRF> 에 대한 이송 속도로서의 계수/백분율	
					0 =	호환성, 계수로서 (0.001 ~ 1.0, FRF = 0은 100%를 의미함)
					1 =	백분율 (0.001 ~ 999.999%)

4.26.1.24 CYCLE84 - 동기 태핑

구문

```
CYCLE84(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDAC>, <MPIT>,
<PIT>, <POSS>, <SST>, <SST1>, <_AXN>, <_PITA>, <_TECHNO>,
<_VARI>, <_DAM>, <_VRT>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<RTP>	REAL	후퇴 평면 (abs)
2	Z0	<RFP>	REAL	원점 (abs)
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<DP>	REAL	드릴링 깊이 = 최종 드릴링 깊이 (abs), <_AMODE> 참조

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미			
5	Z1	<DPR>	REAL	드릴링 깊이 = 최종 드릴링 깊이 (inc), <_AMODE> 참조			
6	DT	<DTB>	REAL	드릴링 깊이에서 머무는 초 단위의 드웰 시간			
7	SDE	<SDAC>	INT	싸이클 종료 후 회전 방향			
8		<MPIT>	REAL	"ISO 미터법" 전용의 나사 크기 (피치는 실행 도중 내부에서 계산됨)			
9	P	<PIT>	REAL	피치 값, 단위는 <_PITA> 참조			
10	αS[1)]	<POSS>	REAL	방향이 지정된 스핀들 정지를 위한 스핀들 위치			
11	S	<SST>	REAL	태핑 스핀들 속도			
12	SR	<SST1>	REAL	후퇴 스핀들 속도			
13		<_AXN>	INT	드릴링 축	0 =	세 번째 기하 축	
					1 =	첫 번째 기하 축	
					2 =	두 번째 기하 축	
					≥ 3 =	세 번째 기하 축	
14		<_PITA>	INT	피치 단위 (<PIT> 및 <MPIT> 평가)			
					0 =	피치 (mm)	- 평가<MPIT>/<PIT>
					1 =	피치 (mm)	- 평가<PIT>
					2 =	피치 (TPI)	- <PIT> 평가 (인치당 스레드 수)
					3 =	피치 (인치)	- 평가<PIT>
					4 =	MODULUS	- 평가<PIT>

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
15		<_TECHNO>	INT	테크놀로지[1)]		
				UNITS:	정위치 정지 반응	
					0 =	싸이클 호출 전과 같이 정위치 정지 반응 활성 상태
					1 =	정위치 정지 G601
					2 =	정위치 정지 G602
					3 =	정위치 정지 G603
				TENS:	피드포워드 제어	
					0 =	싸이클 호출 전과 같이 피드포워드 제어 사용/사용 안 함
					1 =	피드포워드 제어 사용 FFWON
					2 =	피드포워드 제어 사용 안함 FFWON
				HUNDREDS:	가속	
					0 =	싸이클 호출 전과 같이 SOFT/BRISK/DRIVE 활성 상태
					1 =	저크 제한 사용 SOFT
					2 =	저크 제한 사용 안함 BRISK
					3 =	가속도 감소 DRIVE
				THOUSANDS:	MCALL 스핀들 모드	
					0 =	MCALL의 스핀들 운전 재활성화
					1 =	MCALL의 경우 위치 제어로 유지됨
16		<_VARI>	INT	가공 형식		
				UNITS:	0 =	일회 가공
					1 =	칩 분쇄 (심공 태핑)
					2 =	칩 제거 (심공 태핑)
				THOUSANDS:	화면 형식의 경우에는 ISO/SIEMENS 모드가 적합하지 않음	
					0 =	ISO 호환성에 따른 호출
					1 =	SIEMENS 상황에 따른 호출
17	D	<_DAM>	REAL	최대 깊이 절입 (칩 제거/칩 분쇄만 해당)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
18	V2	<_VRT>	REAL	각 가공 단계 후의 후퇴 거리 (칩 분쇄에만 해당), <_AMODE> 참조		
19		<_PITM>	STRING[15]	피치 입력을 위한 마커로서의 문자열[2)]		
20		<_PTAB>	STRING[5]	나사 테이블의 문자열 ("", "ISO", "BSW", "BSP", "UNC")[2)]		
21		<_PTABA>	STRING[20]	나사 테이블에서 선택하기 위한 문자열 (예: "M 10", "M 12", ...)[2)]		
22		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
23		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	호환성 모드 (재컴파일 화면 형식에만 해당), MD 52216 bit0 = 1인 경우[1)]	
					0 =	테크놀로지 파라미터가 표시됨 (호환성): TECHNO 파라미터 유효
					1 =	테크놀로지 파라미터가 표시되지 않음: "싸이클 호출 이전과 같이" 테크놀로지 활성 상태
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
24		<_AMODE>	INT	대체 모드		
				UNITS:	드릴링 깊이 = 최종 드릴링 깊이 Z1 (abs/inc)	
					0 =	호환성, <DP>/<DPR> 프로그래밍에서
					1 =	증분치
					2 =	절대치
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	나사 회전 방향 우측/좌측	
					0 =	호환성, PIT/MPTI 부호에서
					1 =	우측
					2 =	좌측
				TEN THOUSANDS:	예비용	
				HUNDRED THOUSANDS:	예비용	
				ONE MILLION:	개별 가공 단계 이후 후퇴 거리 V2 수동/자동	
					0 =	호환성, <_VRT> 프로그래밍에서 (>0 변수 값 또는 ≤0 표준 값 1 mm / 0.0394인치)
					1 =	자동 (표준 값 1 mm / 0.0394인치)
					2 =	수동 (V2에서와 마찬가지로 프로그래밍)

[1] 테크놀로지 필드는 설정 날짜 SD52216 $MCS_FUNCTION_MASK_DRILL에 따라 숨겨질 수 있습니다.

[2] 파라미터 19, 20 및 21은 화면 형식 나사 테이블에서 나사 선택에만 사용됩니다. 싸이클 런타임에서는 싸이클 정의를 통해 나사 테이블에 액세스할 수 없습니다.

4.26.1.25 CYCLE85 - 리밍

구문

```
CYCLE85(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	RP	<RTP>	REAL	후퇴 평면 (abs)		
2	Z0	<RFP>	REAL	원점 (abs)		
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
4	Z1	<DP>	REAL	드릴링 깊이 (abs), <_AMODE> 참조		
5	Z1	<DPR>	REAL	드릴링 깊이 (inc), <_AMODE> 참조		
6	DT	<DTB>	REAL	최종 드릴링 깊이에서 드웰 시간, <_AMODE> 참조		
7	F	<FFR>	REAL	이송 속도		
8	FR	<RFF>	REAL	후퇴 동안 피드 속도		
9		<_GMODE>	INT	예비용		
10		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
11		<_AMODE>	INT	대체 모드 (드릴링)		
				UNITS:	드릴링 깊이 Z1 (abs/inc)	
					0 =	호환성, DP/DPR 프로그래밍에서
					1 =	증분치
					2 =	절대치
				TENS:	최종 드릴링 깊이에서 드웰 시간 DT (초/회전)	
					0 =	호환성, DT 부호에서 (> 0초 또는 < 0회전)
					1 =	초
					2 =	회전

4.26.1.26 CYCLE86 - 보링

구문

```
CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>,
<RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	RP	<RTP>	REAL	후퇴 평면 (abs)		
2	Z0	<RFP>	REAL	원점 (abs)		
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
4	Z1	<DP>	REAL	드릴링 깊이 (abs), <_AMODE> 참조		
5	Z1	<DPR>	REAL	드릴링 깊이 (inc), <_AMODE> 참조		
6	DT	<DTB>	REAL	최종 드릴링 깊이에서 드웰 시간, <_AMODE> 참조		
7	DIR	<SDIR>	INT	스핀들 회전 방향	3 =	M3
					4 =	M4
8	DX	<RPA>	REAL	X 방향의 도피 거리		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
9	DY	<RPO>	REAL	Y 방향의 도피 거리		
10	DZ	<RPAP>	REAL	Z 방향의 도피 거리		
11	SPOS	<POSS>	REAL	도피를 위한 스핀들 위치 (방향이 지정된 스핀들 정지의 경우, 단위: 도)		
12		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	도피 모드	
					0 =	도피, 호환성
					1 =	후퇴 안 함
13		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
14		<_AMODE>	INT	대체 모드		
				UNITS:	드릴링 깊이 Z1 (abs/inc)	
					0 =	호환성, <DP>/<DPR> 프로그래밍에서
					1 =	증분치
					2 =	절대치
				TENS:	최종 드릴링 깊이에서 드웰 시간 DT (초/회전)	
					0 =	호환성, DT 부호에서 (> 0초 또는 < 0회전)
					1 =	초
					2 =	회전

4.26.1.27 CYCLE92 - 절단

구문

```
CYCLE92(<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>,
<_SV1>, <_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>,
<_VARI>, <_DN>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	X0	<_SPD>	REAL	원점 (abs, 항상 직경)		
2	Y0	<_SPL>	REAL	원점 (abs)		
3	X1	<_DIAG1>	REAL	속도 감소를 위한 깊이, <_AMODE> (UNITS) 참조		
4	X2	<_DIAG2>	REAL	최종 깊이, <_AMODE> (TENS) 참조		
5	R/FS	<_RC>	REAL	라운딩 상태 또는 모따기 너비, <_AMODE> (THOUSANDS) 참조		
6	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
7	S	<_SV1>	REAL	일정한 스핀들 속도, <_AMODE> (TEN THOUSANDS) 참조		
	V			주속 일정 제어		
8	SV	<_SV2>	REAL	주속 일정 제어에서 최대 속도		
9	DIR	<_SDAC>	INT	스핀들 회전 방향	3 =	M3에 해당
					4 =	M4에 해당
10	F	<_FF1>	REAL	속도 감소를 위한 깊이만큼 절입		
11	FR	<_FF2>	REAL	최종 깊이만큼 감소된 절입		
12	SR	<_SS2>	REAL	최종 깊이만큼 감소된 속도		
13	XM	<_DIAGM>	REAL	파트-케쳐 철회 깊이 (abs, 항상 직경)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
14		<_VARI>	INT	가공 형식		
				UNITS:	후퇴	
					0 =	<_SPD> + <_SDIS>로 후퇴
					1 =	끝에서는 후퇴 없음
				TENS:	파트-케쳐	
					0 =	아니오. M 명령을 실행하지 않음
					1 =	예. CUST_TECHCYC(101)에서 호출 - open drawer, CUST_TECHCYC(102)에서 호출 - close drawer
15		<_DN>	INT	공구의 두 번째 에지의 D 번호, 프로그래밍되지 않은 경우 ⇒ D +1		
20		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
21		<_AMODE>	INT	대체 모드		
				UNITS:	속도 감소를 위한 깊이 (<_DIAG1>)	
					0 =	절대, 직경에서 가로 축의 값
					1 =	증분, 반경에서 가로 축의 값
				TENS:	최종 깊이 (<_DIAG2>)	
					0 =	절대, 직경에서 가로 축의 값
					1 =	증분, 반경에서 가로 축의 값
				HUNDREDS:	예비용	
				THOUSANDS:	반경/모따기 (<_RC>)	
					0 =	반경
					1 =	모따기
				TEN THOUSANDS:	스핀들 속도 / 절삭 속도 (<_SV1>)	
					0 =	일정한 스핀들 속도
					1 =	주속 일정 제어

4.26.1.28 CYCLE95 - 형상 절삭

구문

```
CYCLE95(<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>,
<_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	CON	<NPP>	STRING [140]	형상 이름
2	D	<MID>	REAL	황삭 중 최대 깊이 절입, <_GMODE> 참조
3	UZ	<FALZ>	REAL	Z 방향 정삭 여유
4	UX	<FALX>	REAL	X 방향 정삭 여유
5	U	<FAL>	REAL	형상과 평행한 정삭 여유 (두 축 모두에 적용)
6	F	<FF1>	REAL	황삭 이송 속도

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
7	FY	<FF2>	REAL	릴리프 절삭 삽입 이송 속도		
8	FS	<FF3>	REAL	정삭 이송 속도		
9		<_VARI>	INT	가공 형식		
				UNITS 및 TENS:		
					1 =	황삭, 세로, 외부
					2 =	황삭, 가로, 외부
					3 =	황삭, 세로, 내부
					4 =	황삭, 가로, 내부
					5 =	정삭, 세로, 외부
					6 =	정삭, 가로, 외부
					7 =	정삭, 세로, 내부
					8 =	정삭, 가로, 내부
					9 =	전체 가공, 세로, 외부
					10 =	전체 가공, 가로, 외부
					11 =	전체 가공, 세로, 내부
					12 =	전체 가공, 가로, 내부
				HUNDREDS:		
					0 =	형상에서 라운딩 실행, 잔여 코너 없음
					1 =	형상에서 라운딩 안 함
					2 =	이전 교차점에서만 라운딩. 코너가 남을 수 있음.
10	DT	<DT>	REAL	이송 인터럽트에서 드웰 시간		
11	DI	<DAM>	REAL	이송 인터럽트를 위한 거리		
12	VRT	<_VRT>	REAL	형상에서 도피 거리		
					0 =	작동 시스템 (inch 또는 미터)에 관계 없이 내부적으로 1 mm의 도피 거리가 사용됩니다.
					> 0 =	도피 거리

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
13		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	절입 깊이 평가	
					0 =	G 그룹 DIAMON/DIAMOF에 따라 절입 깊이를 계산합니다.
					1 =	절입 깊이가 반경 값이 됩니다 (DIAMON/DIAMOF와 무관).
14		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				THOUSANDS:		
					0 =	호환성 모드: 형상 이름은 NPP에 있습니다.
					1 =	형상 이름이 CYCLE62에서 프로그래밍되어 _SC_CONT_NAME으로 전송됩니다.

4.26.1.29 CYCLE98 - 나사 체인

구문

```
CYCLE98(<_PO1>, <_DM1>, <_PO2>, <_DM2>, <_PO3>, <_DM3>, <_PO4>,
<_DM4>, <APP>, <ROP>, <TDEP>, <FAL>, <_IANG>, <NSP>, <NRC>,
<NID>, <_PP1>, <_PP2>, <_PP3>, <_VARI>, <_NUMTH>, <_VRT>, <_MID>,
<_GDEP>, <_IFLANK>, <_PITA>, <_PITM1>, <_PITM2>, <_PITM3>,
<_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	Z0	<_PO1>	REAL	Z 의 원점 (abs)
2	X0	<_DM1>	REAL	X 방향 원점 (abs), 직경
3	Z1	<_PO2>	REAL	Z 방향 중간점 1 (abs/inc), <_AMODE> (UNITS) 참조
4	X1	<_DM2>	REAL	X 방향 중간점 1 (abs/inc), <_AMODE> (TENS) 참조
	X1α			나사 경사 1 (-90° ~ 90°) abs는 항상 직경, inc는 항상 반경
5	Z2	<_PO3>	REAL	Z 방향 중간점 2 (abs/inc), <_AMODE> (HUNDREDS) 참조
6	X2	<_DM3>	REAL	X 방향 중간점 2 (abs/inc), <_AMODE> (THOUSANDS) 참조
	X2α			나사 경사 2 (-90° ~ 90°) abs는 항상 직경, inc는 항상 반경
7	Z3	<_PO4>	REAL	Z 방향 종점 (abs/inc), <_AMODE> (TEN THOUSANDS) 참조
8	X3	<_DM4>	REAL	X 방향 종점 (abs/inc), <_AMODE> (HUNDRED THOUSANDS) 참조
	X3α			나사 경사 3 (-90° ~ 90°) abs는 항상 직경, inc는 항상 반경
9	LW	<APP>	REAL	나사 런인 (inc, 부호 없이 입력)
10	LR	<ROP>	REAL	나사 런아웃 (inc, 부호 없이 입력)
11	H1	<TDEP>	REAL	나사 깊이 (inc, 부호 없이 입력)
12	U	<FAL>	REAL	X 와 Z 방향 정삭 여유

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
13	DP	<_IANG>	REAL	거리 또는 각도로서의 절입 경사, <_AMODE> (MILLION) 참조		
	αP			절입 경사는 파라미터 <_VARI> (HUNDREDS)의 설정에 따라 적용됩니다.		
					<_VARI_HUNDREDS 정의 = 0 - 호환성 모드:	
					> 0 =	한 면에서 측면 절입
					0 =	나사에서 수직 절입
					< 0 =	대체 면을 포함한 측면 절입
					_VARI_HUNDREDS<>0 정의:	
					> 0 =	플러스 방향 절입
					0 =	중앙 절입
					< 0 =	마이너스 방향 절입
14	α0	<NSP>	REAL	첫 번째 나사의 시작 각도 옵셋		
15		<NRC>	INT	황삭 절삭 개수, <_VARI> (TEN THOUSANDS) 참조		
16	NN	<NID>	INT	비 절삭 개수		
17	P0	<_PP1>	REAL	나사 첫 번째 섹션의 피치, <_PITA> 참조		
18	P1	<_PP2>	REAL	나사 두 번째 섹션의 피치, <_PITA> 참조		
19	P2	<_PP3>	REAL	나사 세 번째 섹션의 피치, <_PITA> 참조		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
20		<_VARI>	INT	가공		
				UNITS:	테크놀로지	
					1 =	직선 절입을 사용한 외경 나사
					2 =	직선 절입을 사용한 내경 나사
					3 =	하강 절입을 사용한 외경 나사, 절삭 단면은 일정함
					4 =	하강 절입을 사용한 내경 나사, 절삭 단면은 일정함
				TENS:	예비용	
				HUNDREDS:	절입 유형	
					0 =	<_IANG>에 대한 호환성 모드
					1 =	한 면에서 절입
					2 =	양 면을 교대로 절입
				THOUSANDS:	예비용	
				TEN THOUSANDS:	대체 깊이 절입	
					0 =	호환성, 사전 설정된 황삭 절삭 개수 (<_NRC>)
					1 =	첫 번째 절입의 사전 설정 값 (<_MID>)
				HUNDRED THOUSANDS:	가공 형식	
					0 =	호환성 (황삭과 정삭)
					1 =	황삭
					2 =	정삭
					3 =	황삭과 정삭
				ONE MILLION:	멀티 스타트 나사의 가공 순서	
					0 =	나사의 오름차순으로
					1 =	나사의 내림차순으로
21	N	<_NUMTH>	INT	나사 턴 수		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
22		<_VRT>	REAL	후퇴 거리 (inc)		
					0 =	작동 시스템 (inch 또는 미터)에 관계 없이 내부적으로 1 mm의 도피 거리가 사용됩니다.
					> 0 =	도피 거리
23	D1	<_MID>	REAL	첫 번째 절입, <_VARI> (TEN THOUSANDS) 참조		
24	DA	<_GDEP>	REAL	나사 변경 깊이 ("다중 스타트"에서만 유효)		
					0 =	나사 변경 깊이를 준수하지 않음
					> 0 =	나사 변경 깊이 준수
25		<_IFLANK>	REAL	너비로서의 절입 경사 (인터페이스만 해당)		
26		<_PITA>	INT	나사 피치 평가		
					0 =	피치에 대한 호환성 모드, 이전처럼 작동 시스템에 따라 <_PP1> ~ <_PP3> 평가 (미터/인치)
					1 =	피치 (mm)
					2 =	피치 (인치당 스레드, TPI)
					3 =	피치 (인치)
					4 =	MODULUS
27		<_PITM1>	STRING[15]	피치 입력을 위한 마커로 사용되는 문자열 (인터페이스만 해당)		
28		<_PITM2>	STRING[15]	피치 입력을 위한 마커로 사용되는 문자열 (인터페이스만 해당)		
29		<_PITM3>	STRING[15]	피치 입력을 위한 마커로 사용되는 문자열 (인터페이스만 해당)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
30		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	---	예비용
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
31		<_AMODE>	INT	대체 모드		
				UNITS:	Z 방향 첫 번째 중간점 (Z1)	
					0 =	절대치
					1 =	증분치
				TENS:	X 방향 첫 번째 중간점 (X1)	
					0 =	절대치
					1 =	증분치
					2 =	α
				HUNDREDS:	Z 방향 두 번째 중간점 (Z2)	
					0 =	절대치
					1 =	증분치
				THOUSANDS:	X 방향 두 번째 중간점 (X2)	
					0 =	절대치
					1 =	증분치
					2 =	α
				TEN THOUSANDS:	Z 방향 종점 (Z3)	
					0 =	절대치
					1 =	증분치
				HUNDRED THOUSANDS:	X 방향 종점 (X3)	
					0 =	절대치
					1 =	증분치
					2 =	α
				ONE MILLION:	절입 경사를 각도 또는 너비로 선택	
					0 =	절입 각도 <_IANG>
					1 =	절입 경사 <_IFLANK>
				TEN MILLIONS:	단일/다중 나사	
					0 =	호환성 모드 (시작 각도 <_NSP>가 평가됨)
					1 =	단일 나사 (시작 각도 옵셋 <_NSP> 포함)
					2 =	다중

4.26.1.30 CYCLE99 - 나사 선삭

구문

```
CYCLE99(<_SPL>, <_SPD>, <_FPL>, <_FPD>, <_APP>, <_ROP>, <_TDEP>,
<_FAL>, <_IANG>, <_NSP>, <_NRC>, <_NID>, <_PIT>, <_VARI>,
<_NUMTH>, <_SDIS>, <_MID>, <_GDEP>, <_PIT1>, <_FDEP>, <_GST>,
<_GUD>, <_IFLANK>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_DMODE>,
<_AMODE>, <_S_XRS>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	Z0	<_SPL>	REAL	원점 (abs)		
2	X0	<_SPD>	REAL	원점 (abs, 항상 직경)		
3	Z1	<_FPL>	REAL	<_AMODE> (UNITS) 와 연계된 종점		
4	X1	<_FPD>	REAL	<_AMODE> (TENS) 와 연계된 종점		
5	LW/LW2	<_APP>	REAL	<_AMODE> (HUNDREDS) 와 연계된 나사 런인 또는 나사 런인 = <_AMODE> (HUNDREDS)와 연계된 나사 런아웃		
6	LR	<_ROP>	REAL	나사 런아웃		
7	H1	<_TDEP>	REAL	나사 깊이		
8	U	<_FAL>	REAL	X 와 Z 방향 정삭 여유		
9	DP	<_IANG>	REAL	거리 또는 각도 단위의 절입 경사, <_AMODE> (THOUSANDS) 와 연계		
	αP				> 0 =	플러스 방향 절입
					< 0 =	마이너스 방향 절입
					0 =	중앙 절입
10	α0	<_NSP>	REAL	시작 각도 옵셋 ("싱글 스타트"에서만 유효)		
11	ND	<_NRC>	INT	황삭 절삭 개수, <_VARI> (TEN THOUSANDS) 와 조합		
12	NN	<_NID>	INT	비 절삭 개수		
13	P	<_PIT>	REAL	피치 값, <_PITA>와 연계		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
14		<_VARI>	INT	가공 형식		
				UNITS:	테크놀로지	
					1 =	직선 절입을 사용한 외경 나사
					2 =	직선 절입을 사용한 내경 나사
					3 =	하강 절입을 사용한 외경 나사, 절삭 단면은 일정함
					4 =	하강 절입을 사용한 내경 나사, 절삭 단면은 일정함
				TENS:	예비용	
				HUNDREDS:	절입 유형	
					1 =	한 면에서 절입
					2 =	양 면을 교대로 절입
				THOUSANDS:	예비용	
				TEN THOUSANDS:	대체 깊이 절입	
					0 =	사전 설정된 황삭 절삭 개수 (<_NRC>)
					1 =	첫 번째 절입의 사전 설정 값 (<_MID>)
				HUNDRED THOUSANDS:	가공 형식	
					1 =	황삭
					2 =	정삭
					3 =	황삭과 정삭
				ONE MILLION:	멀티 스타트 나사의 가공 순서	
					0 =	나사의 오름차순으로
					1 =	나사의 내림차순으로
15	N	<_NUMTH>	INT	나사 턴 수		
16	VR	<_SDIS>	REAL	후퇴 거리, inc		
17	D1	<_MID>	REAL	첫 번째 절입 깊이, <_VARI> (TEN THOUSANDS) 와 연계		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
18	DA	<_GDEP>	REAL	나사 변경 깊이 ("다중 스타트"에서만 유효)		
					0 =	나사 변경 깊이를 준수하지 않음
					> 0 =	나사 변경 깊이 준수
19	G 코드	<_PIT1>	REAL	회전당 피치 변경		
					0 =	피치가 일정함 (G33)
					> 0 =	피치 증가 (G34)
					< 0 =	피치 감소 (G35)
20		<_FDEP>	REAL	삽입 깊이 (부호 없이 입력)		
21	N1	<_GST>	INT	시작 나사 N1 = 1...N, <_AMODE> (HUNDRED THOUSANDS) 와 연계		
22		<_GUD>	INT	예비용		
23		<_IFLANK>	REAL	너비로서의 절입 경사 (인터페이스만 해당)		
24		<_PITA>	INT	피치 단위 (PIT 및/또는 MPIT 평가)		
					0 =	피치 단위 (mm) - MPIT/PIT 평가
					1 =	피치 단위 (mm) - PIT 평가
					2 =	피치 단위 (TPI) - PIT 평가 (인치당 스레드)
					3 =	피치 단위 (인치) - PIT 평가
					4 =	MODULE - PIT 평가
25		<_PITM>	STRING[15]	피치 입력을 위한 마커로 사용되는 문자열 (인터페이스만 해당)[1)]		
26		<_PTAB>	STRING[20]	나사 테이블의 문자열 (인터페이스만 해당)[1)]		
27		<_PTABA>	STRING[20]	나사 테이블의 선택 문자열 (인터페이스만 해당)[1)]		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
28		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	나사 유형	
					0 =	수직 나사
					1 =	단면 나사
					2 =	테이퍼 나사
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 베이직

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
29		<_AMODE>	INT	대체 모드		
				UNITS:	Z 방향의 나사 길이	
					0 =	절대치
					1 =	증분치
				TENS:	X 방향의 나사 길이	
					0 =	절대, 직경에서 가로 축의 값
					1 =	증분, 반경에서 가로 축의 값
					2 =	α
				HUNDREDS:	접근/런인 경로 계산 <_APP>	
					0 =	나사 런인 <_APP>
					1 =	나사 런인 = 나사 런아웃 <_APP> = -<_ROP>
					2 =	나사 런인 경로 지정 <_APP> = -<_APP>
				THOUSANDS:	절입 경사를 각도 또는 너비로 선택	
					0 =	절입 각도 <_IANG>
					1 =	절입 경사 <_IFLANK>
				TEN THOUSANDS:	단일/다중 나사	
					0 =	단일 나사 (시작 각도 옵셋 <_NSP> 포함)
					1 =	다중
				HUNDRED THOUSANDS:	나사 시작 <_GST>	
					0 =	전체 가공
					1 =	이 나사부터 가공 시작
					2 =	이 나사만 가공
				ONE MILLION:	세로 나사에 대한 처짐 보정	
					0 =	크라운 나사의 세그먼트 높이 XS
					1 =	크라운 나사의 반경 RS
30	XS/RS	<_S_XRS>	REAL	<_AMODE>와 함께 세로 나사에 대한 처짐 보정 ONE MILLION		

참고

[1] 파라미터 <_PITM>, <_PTAB> <_PTABA>은 화면 형식 나사 테이블에서 나사 선택에만 사용됩니다.
싸이클 런타임에서는 싸이클 정의를 통해 나사 테이블에 액세스할 수 없습니다.

4.26.1.31 CYCLE435 - 드레서 좌표계 설정

구문

```
CYCLE435(<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>,
<S_PVL>, <S_PD>, <S_PL>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1		<_T>	STRING[32]	연삭 휠의 공구 이름		
2		<_DD>	INT	연삭 휠의 절삭날 번호		
3		<S_TA>	STRING[32]	드레싱 공구 원점 - 드레싱 공구 이름		
4		<S_DA>	INT	드레싱 공구의 절삭날 번호		
5		<S_AD>	REAL	드레싱 값, 직경		
6		<S_AL>	REAL	드레싱 값, 단면		
7		<S_PVD>	REAL	형상 트루잉 옵셋, 직경		
8		<S_PVL>	REAL	형상 트루잉 옵셋, 단면		
9		<S_PD>	REAL	형상 트루잉 공차, 직경		
10		<S_PL>	REAL	형상 트루잉 공차, 단면		
11		<_AMODE>	INT	대체 모드		
				UNITS:	사이클 끝에서 활성인 공구	
					0 =	드레싱 공구 활성
					1 =	휠 활성

4.26.1.32 CYCLE495 - 형상 트루잉

구문

```
CYCLE495(<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>,
<S_PA>, <S_N>, <_DMODE>, <_AMODE>, <S_FW>, <S_HW>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1		<_T>	STRING[20]	연삭 휠의 공구 이름
2		<_DD>	INT	연삭 휠의 절삭날 번호
3		<_SC>	REAL	장애물을 피하기 위한 도피 거리, 증분치
4		<_F>	REAL	형상 트루잉 속도

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
5		<_VARI>	INT	가공 형식		
				UNITS:	형상 트루잉 유형	
					1 =	축과 평행
					2 =	형상에 대해 수평
				TENS:	가공 방향	
					0 =	당김 절삭날 위치 1~4에서 가능
					1 =	밈 절삭날 위치 1~4에서 가능
					2 =	교대 절삭날 위치 1~8에서 가능
					3 =	시작 → 종료 절삭날 위치 1~8에서 가능
					4 =	종료 → 시작 절삭날 위치 1~8에서 가능
				HUNDREDS:	절입 방향	
					1 =	G18의 경우 절입 X 또는 G19의 경우 절입 Y-
					2 =	G18의 경우 절입 X 또는 G19의 경우 절입 Y+
					3 =	G18 및 G19의 경우 절입 Z-
					4 =	G18 및 G19의 경우 절입 Z+
6		<_D>	REAL	축에 평행한 형상 트루잉 유형을 위한 드레싱 값		
7		<_DX>	REAL	형상에 평행한 형상 트루잉 유형을 위해 G18의 경우 드레싱 값 X 또는 G19의 경우 드레싱 값 Y		
8		<_DZ>	REAL	형상에 평행한 형상 트루잉 유형을 위해 G18 및 G19의 경우 드레싱 값 Z		
9		<S_PA>	REAL	형상 트루잉 여유		
10		<S_N>	INT	형상 트루잉 프로그램 내 스트로크 수		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
11		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
12		<_AMODE>	INT	대체 모드		
				UNITS:	형상 트루잉 선택, 신규/계속	
					1 =	새로 만들기
					2 =	계속
				TENS:	형상 트루잉 여유 선택	
					0 =	거친 형상에서부터 형상의 최저점까지
					1 =	거친 형상에서부터 형상의 최고점까지
13		<S_FW>	REAL	드레싱 공구의 여유 각도		
14		<S_HW>	REAL	드레싱 공구의 홀더 각도		

참고

CYCLE495로 프로파일링할 때의 형상 해석은 사전 정의된 프로시저인 CONTDCON을 사용하여 수행됩니다. 공구 반경 보정(G41/G42)이 활성 상태일 때 CONTDCON이 허용되지 **않으므로** CYCLE495 호출 전에 G40으로 공구 반경 보정을 반드시 비활성화해야 합니다.

4.26.1.33 CYCLE782 - 부하에 맞게 조정

참고

CYCLE782를 사용하려면 "지능형 부하 조정" 옵션에 대한 라이선스가 필요합니다.

구문

```
CYCLE782(<S_MODE>, <S_TESTAXIS>, <S_VALUE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1		<S_MODE>	INT	처리 모드		
				UNITS:	선택/선택 해제	
					0 =	조정 비활성화
					1 =	조정 활성화
					2 =	축의 기본 관성 결정 및 활성화(스타트업 기능)
				TENS:	측정 버전(<S_MODE> HUNDREDS POSITION = 0에만 해당)	
					0 =	표준 모드
					1 =	정밀 모드(예: 마찰 계산)
				백 단위	관성 모멘트를 측정하거나 지정	
					0 =	측정 이 모드에서는 부하 파악을 위한 이송 모션이 실행됩니다.
					1 =	관성 모멘트 지정 이 모드에서는 부하 파악을 위한 이송 모션이 실행되지 않습니다. 대신 자동 서보 튜닝 기능(AST)에 의해 결정된 관성 모멘트가 전달됩니다.
				천 단위	측정 결과 표시	
					0 =	측정 결과 화면 OFF
					1 =	측정 결과 화면 ON
				TEN THOUSANDS	측정 결과 표시 기간	
					1 =	8초 후에 디스플레이가 자동으로 꺼집니다.
					3 =	NC 시작으로 승인
				HUNDRED THOUSAND	로드 선택 [1]	
					0 =	전체 로드 프로세스 결정: 축 비어 있음 + 공작물

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
					1 =	별도로 로드 결정: 공작물
2		<S_TESTAXIS>	AXIS	채널 축: • 관성 모멘트를 파악해야 하는 • 조정에 대해 활성화해야 함(<S_MODE> HUNDREDS POSITION = 0만 해당)		
3		<S_VALUE>	REAL	관성 모멘트 값(<S_MODE> HUNDREDS POSITION = 1만 해당)		

[1] 값은 머신 데이터 52212 $MCS_FUNCTION_MASK_TECH bit 18에서 자동으로 가져옵니다.

참고

스타트업 기능 "축의 기본 관성 결정 및 활성화"(<S_MODE> UNIT PLACE = 2)는 사용자 인터페이스에서 지원되지 않습니다. 이를 위해서는 싸이클 호출을 프로그래밍해야 합니다. 측정 결과는 머신 데이터 53350 $MAS_ILC_BASE_VALUE에 입력됩니다.

파라미터 "로드 선택"(<S_MODE> ONE HUNDRED THOUSANDS DIGIT)은 머신 데이터 52212 $MCS_FUNCTION_MASK_TECH Bit 18에서 사용자 인터페이스에 의해 읽혀지고 싸이클 호출에 입력됩니다.

직선 축과 로터리 축에 대해 로드를 결정하거나 지정할 수 있습니다. 직선 축의 경우 질량, 로터리 축의 경우 관성 모멘트가 항상 측정 결과 화면에 표시되거나 사용자 인터페이스에 지정됩니다.

축 유형에 상관없이 관성 모멘트는 항상 머신 데이터 53350 $MAS_ILC_BASE_VALUE에 기록됩니다.

예

예 1:

NC 프로그램 시작 시 축 MX1의 부하 레벨을 파악해야 합니다. 이 값은 적응 업데이트를 위한 기초 정보로 사용됩니다. 결과가 표시되고 NC 시작으로 디스플레이가 승인됩니다.

프로그램 코드

```
...
CYCLE782(31011,MX1)
...
```

예 2:

프로그램이 진행되는 동안 AST를 사용하여 파악한 관성 모멘트 값이 _MY_VALUE 변수, 이 값으로 호출된 부하 싸이클, MX2에 대해 활성화된 조정으로 전송됩니다. 결과가 표시되면 안 됩니다.

프로그램 코드

```
DEF REAL _MY_VALUE
_MY_VALUE=...
...
CYCLE782(101,MX2,_MY_VALUE)
...
```

예 3:

MX2 축에 대한 조정이 비활성화되어야 합니다.

프로그램 코드

```
...
CYCLE782(0,MX2)
...
```

4.26.1.34 CYCLE800 – 스위블 평면 / 스위블 공구 / 정렬 공구

구문

```
CYCLE800(<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>,
<_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1		<_FR>	INT	후퇴 모드:	0 =	후퇴 안 함
					1 =	기계 축 Z 후퇴
					2 =	기계 축 Z 및 XY 후퇴
					3 =	예비용
					4 =	공구 방향으로 최대 후퇴
					5 =	공구 방향으로 증분 후퇴

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
2		<_TC>	STRING[32]	스위블 데이터 블록 이름:	""	"" (이름 없음) 스위블 데이터 블록이 단 하나인 경우
					"0"	스위블 데이터 블록 선택 해제 (스위블 프레임 삭제)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
3		<_ST>	INT	상태 변환		
				UNITS:		
					0 =	신규, 스위블 레벨이 삭제되고 현재 파라미터를 사용하여 다시 계산됩니다.
					1 =	추가, 스위블 레벨에 추가됩니다.
				TENS:	공구 팁 추적 예/아니요 (스타트업 과정에서 SWIVEL 기능이 생성된 경우에만 활성)	
					0 =	공구 팁 추적 안함
					1 =	공구 팁 추적 (TRAORI)
				HUNDREDS:	공구 접근/정렬 (공구 스위블 화면 양식에 기능이 표시됨)	
					0 =	공구 접근 안함
					1 =	공구에 접근 (반경 밀링이 바람직함)
					2 =	**선삭** 공구 정렬 (스타트업 스위블 밀링을 위해 B축 좌표계를 설정한 경우)
					3 =	**밀링** 공구 정렬 (스타트업 스위블 밀링을 위해 B 축 좌표계를 설정한 경우)
				THOUSANDS:	내부 "JOG 스위블" 파라미터	
				TEN THOUSANDS:	방향 파라미터 <_DIR> 참조	
					0 =	스위블 "예"
					1 =	스위블 "아니오", "마이너스" 방향[3)]
					2 =	스위블 "아니오", "플러스" 방향[3)]
				HUNDRED THOUSANDS:	방향 파라미터 <_DIR> 참조	
					0 =	호환성

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
					1 =	최적화된 방향 선택 "마이너스" (사용자 인터페이스에만 해당) [4]
					2 =	최적화된 방향 선택 "플러스" (사용자 인터페이스에만 해당) [4]
4		<_MODE> [5]	INT	스위블 모드: 스위블 각도 및 스위블 순서 평가 (비트 코딩)		
				비트: 7 6	0 0:	축에 따른 스위블 각도 -> 파라미터 <_A>, <_B>, <_C> 참조
					0 1:	입체각 -> 파라미터 <_A>, <_B> 참조 [1]
					1 0:	투사각 -> 파라미터 <_A>, <_B>, <_C> 참조 [1]
					1 1:	다이렉트 로터리 축 스위블 모드 -> 파라미터 <_A>, <_B> 참조 [1]
				비트: 5 4 3 2 1 0 (입체각에는 해당되지 않음)	x x x x 0 1	X 주위로 1번째 회전 _A
					x x x x 1 0	Y 주위로 1번째 회전 _A
					x x x x 1 1	Z 주위로 1번째 회전 _A
					x x 0 1 x x	X 주위로 2번째 회전 _B
					x x 1 0 x x	Y 주위로 2번째 회전 _B
					x x 1 1 x x	Z 주위로 2번째 회전 _B
					0 1 x x x x	X 주위로 3번째 회전 _C
					1 0 x x x x	Y 주위로 3번째 회전 _C
					1 1 x x x x	Z 주위로 3번째 회전 _C
5	X0	<_X0>	REAL	회전하기 전의 원점 X		
6	Y0	<_Y0>	REAL	회전하기 전의 원점 Y		
7	Z0	<_Z0>	REAL	회전하기 전의 원점 Z		
8	X(A)	<_A>	REAL	파라미터의 설정에 따른 1번째 회전 <_MODE>		
9	Y(B)	<_B>	REAL	파라미터의 설정에 따른 2번째 회전 <_MODE>		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
10	Z(C)	<_C>	REAL	파라미터의 설정에 따른 3번째 회전 <_MODE>		
11	X1	<_X1>	REAL	회전 후의 원점 X		
12	Y1	<_Y1>	REAL	회전 후의 원점 Y		
13	Z1	<_Z1>	REAL	회전 후의 원점 Z		
14	- 또는 +	<_DIR>	INT	로터리 축의 이송 초기화 (디폴트 = -1!)		
					-1 =	로터리 축 1 또는 2 중 보다 작은 값에 포지셔닝 [2]
					+1 =	로터리 축 1 또는 2 중 보다 큰 값에 포지셔닝 [2]
					0 =	스위블하지 않음 (스위블 프레임만 계산) [1] [3]
15	FR	<_FR_I>	REAL	공구 방향 증분에서 후퇴 값 (inc)		
16		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	공구 정렬 시 베타 값의 표시	
					0 =	값
					1 =	화살표

참고

다음 전송 파라미터가 간접적으로 (파라미터로서) 프로그래밍된 경우에는 화면 형식이 리셋되지 않습니다: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

1) 스타트업 과정에서 SWIVEL 기능을 생성한 경우에만 선택할 수 있습니다.

2) 로터리 축 1 또는 2에 대한 방향 기준이 IBN SWIVEL에 설정된 경우에 선택할 수 있습니다.

방향 기준이 "아니오"이면 선택 필드가 없습니다.

3) 스위블 선택 "아니오"가 회색으로 비활성화될 수 있습니다 (SD 55221 Bit 0).

스위블 "아니오", "마이너스" 방향은 <_DIR> = 0 및 _ST TEN THOUSANDS = 1에 해당합니다.

스위블 "아니오", "플러스" 방향은 <_DIR> = 0 및 _ST TEN THOUSANDS = 2에 해당합니다.

4) 방향 기준이 있는 로터리 축이 극점 위치 (위치 값 = 0) 에 있는 경우에도 로터리 축 1 또는 2의 방향을 선택합니다.

5) 코딩 예: 축 기준 회전, 회전 순서 ZYX

이진: 00011011 십진: 27

축 식별자 XYZ는 NC 채널의 기하 축에 해당합니다. XYZ 축을 기준으로 하는 개별 회전은 허용됩니다. 예를 들어, CYCLE800의 1회 호출에서는 ZXZ 축을 기준으로 하는 회전 순서가 허용되지 않습니다.

4.26.1.35 CYCLE801 – 격자 또는 프레임 위치 패턴

구문

```
CYCLE801(<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,
<_DMODE>)
```

파라미터

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
1	X0	<_SPCA>	REAL	첫 번째 축을 따른 위치 패턴 (그리드/프레임) 의 원점 (abs)		
2	Y0	<_SPCO>	REAL	두 번째 축을 따른 위치 패턴 (그리드/프레임) 의 원점 (abs)		
3	α0	<_STA>	REAL	기본 회전 각도 (첫 번째 축에 대한 각도)	< 0 =	CW 방향 회전
					> 0 =	CCW 방향 회전
4	L1	<_DIS1>	REAL	열 간 거리 (첫 번째 축과의 위치 거리, 부호 없이 입력)		
5	L2	<_DIS2>	REAL	행 간 거리 (두 번째 축과의 거리, 부호 없이 입력)		

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
6	N1	<_NUM1>	INT	열의 수		
7	N2	<_NUM2>	INT	행 수		
8		<_VARI>	INT	가공 형식		
				UNITS:	위치 패턴	
					0 =	격자
					1 =	프레임
				TENS:	예비용	
				HUNDREDS:	예비용	
9		<_UMODE>	INT	예비용		
10	αX	<_ANG1>	REAL	첫 번째 축과의 정렬 각도 (첫 번째 축 기준 사면으로 정렬된 라인)		
					< 0 =	CW 방향 측정 (0 ~ -90도)
					> 0 =	CCW 방향 측정 (0 ~ 90도)
11	αY	<_ANG2>	REAL	두 번째 축과의 정렬 각도 (두 번째 축 기준 사면으로 정렬된 라인)		
					< 0 =	CW 방향 측정 (0 ~ -90도)
					> 0 =	CCW 방향 측정 (0 ~ 90도)
12		<_HIDE>	STRING [200]	숨겨진 위치 • 최대 198 문자 • 연속 위치 번호 지정. 예: "1,3" (위치 1과 3은 실행되지 않음)		
13		<_NSP>	INT	예비용		

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
14		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)

4.26.1.36 CYCLE802 - 임의 위치

구문

```
CYCLE802(<_XA>, <_YA>, <_X0>, <_Y0>, <_X1>, <_Y1>, <_X2>, <_Y2>,
<_X3>, <_Y3>, <_X4>, <_Y4>, <_X5>, <_Y5>, <_X6>, <_Y6>, <_X7>,
<_Y7>, <_X8>, <_Y8>, <_VARI>, <_UMODE>, <_DMODE>, <S_ABA>,
<S_AB0>, <S_AB1>, <S_AB2>, <S_AB3>, <S_AB4>, <S_AB5>, <S_AB6>,
<S_AB7>, <S_AB8>)
```

파라미터

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
1		<_XA>	INT	모든 X 위치에 대한 대체 값 (9자리 십진수) 자릿수: 876543210 (숫자 위치는 드릴링 위치 Xn에 해당)		
				위치 값:	1 =	절대 (처음 프로그래밍된 위치는 항상 절대치)
					2 =	증분치

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
2		<_YA>	INT	모든 Y 위치에 대한 대체 값 (9자리 십진수) 자릿수: 876543210 (숫자 위치는 드릴링 위치 Yn에 해당)		
				위치 값:	1 =	절대 (처음 프로그래밍된 위치는 항상 절대치)
					2 =	증분치
3	X0	<_X0>	REAL	1. 위치 X		
4	Y0	<_Y0>	REAL	1. 위치 Y		
5	X1	<_X1>	REAL	2. 위치 X		
6	Y1	<_Y1>	REAL	2. 위치 Y		
7	X2	<_X2>	REAL	3. 위치 X		
8	Y2	<_Y2>	REAL	3. 위치 Y		
9	X3	<_X3>	REAL	4. 위치 X		
10	Y3	<_Y3>	REAL	4. 위치 Y		
11	X4	<_X4>	REAL	5. 위치 X		
12	Y4	<_Y4>	REAL	5. 위치 Y		
13	X5	<_X5>	REAL	6. 위치 X		
14	Y5	<_Y5>	REAL	6. 위치 Y		
15	X6	<_X6>	REAL	7. 위치 X		
16	Y6	<_Y6>	REAL	7. 위치 Y		
17	X7	<_X7>	REAL	8. 위치 X		
18	Y7	<_Y7>	REAL	8. 위치 Y		
19	X8	<_X8>	REAL	9. 위치 X		
20	Y8	<_Y8>	REAL	9. 위치 Y		

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
21		<_VARI>	INT	가공		
				HUNDREDS:	(Jobshop에서 호출하는 경우만 해당) (현재는 0과 2만 평가)	
					0 =	스핀들 클램프 안 함
					1 =	G00 또는 G01로 수직 삽입하는 경우에만 스핀들 클램프
					2 =	가공 작업 전기간 동안 스핀들 클램프
				THOUSANDS:	예비용	
				TEN THOUSANDS:	로터리 축 - 축 조합을 사용하는/사용하지 않는 위치 패턴 (<_VARI> HUNDRED THOUSANDS 사용)	
					0 =	XY (로터리 축을 제외하고 XY만 사용, 호환성)
					1 =	X,Y 또는 Z 및 로터리 축: XA, YB, ZC (로터리 축 1개와 로터리 축이 회전하는 기준이 되는 기하 축)
					2 =	XY 및 로터리 축: XYA, XYB, XYC (로터리 축 1개와 첫 번째 및 두 번째 기하 축, TRACYL 사용 안 함)
				HUNDRED THOUSANDS:	로터리 축	
					0 =	로터리 축 없음 (XY만, 호환성)
					1 =	A 축 (X 중심으로 회전하는 로터리 축)
					2 =	B 축 (Y 중심으로 회전하는 로터리 축)
					3 =	C 축 (Z 중심으로 회전하는 로터리 축)

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미		
				TEN MILLIONS + ONE MILLION:	로터리 축 – 옵셋을 사용하는 위치 패턴 (동일한 축을 중심으로 회전하는 로터리 축이 여러 개인 경우, 인덱스가 너무 크면 1번째 축만 사용.)	
					00 =	1번째 A, B 또는 C 축 또는 호환성 목표
					01 =	2번째 A, B, 또는 C축
					...	
					19 =	20번째 A, B, C축
22		<_UMODE>	INT	클램프할 스핀들 선택: (Jobshop에서 호출한 경우만 해당) (사용자 싸이클 CUST_TECHCYC 호출)		
					3 =	메인 스핀들 클램프/해제
					23 =	카운터스핀들 클램프/해제
23		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
24		<S_ABA>	INT	모든 AB 위치에 대한 대체 값 (9자리 십진수) 자릿수: 876543210 (숫자 위치는 위치 ABn에 해당)		
				위치 값:	1 =	절대 (처음 프로그래밍된 위치는 항상 절대치)
					2 =	증분치
25	A0	<S_AB0>	REAL	로터리 축을 사용한 위치 패턴을 위한 1번째 로터리 축 위치 (<_VARI>와 연계)		
26	A1	<S_AB1>	REAL	로터리 축을 사용한 위치 패턴을 위한 2번째 로터리 축 위치		

번호	파라미터 마스크	파라미터 내부	데이터 유형	의미
27	A2	<S_AB2>	REAL	로터리 축을 사용한 위치 패턴을 위한 3번째 로터리 축 위치
28	A3	<S_AB3>	REAL	로터리 축을 사용한 위치 패턴을 위한 4번째 로터리 축 위치
29	A4	<S_AB4>	REAL	로터리 축을 사용한 위치 패턴을 위한 5번째 로터리 축 위치
30	A5	<S_AB5>	REAL	로터리 축을 사용한 위치 패턴을 위한 6번째 로터리 축 위치
31	A6	<S_AB6>	REAL	로터리 축을 사용한 위치 패턴을 위한 7번째 로터리 축 위치
32	A7	<S_AB7>	REAL	로터리 축을 사용한 위치 패턴을 위한 8번째 로터리 축 위치
33	A8	<S_AB8>	REAL	로터리 축을 사용한 위치 패턴을 위한 9번째 로터리 축 위치

참고

파라미터 X1/Y1/A1~X8/Y8/A8에 필요하지 않은 위치는 무시할 수 있습니다. 그러나 9개 위치 모두에 대해서 <_XA>, <_YA> 및 <S_ABA>의 대체 값을 제공해야 합니다.

위치 패턴 XA, YB 또는 ZC (기하 축 및 로터리 축) 의 경우 위치 패턴을 통해 이송되지 않는 가공 평면의 축 (G17의 경우 Y 및 XA) 은 싸이클 호출 전에 포지셔닝해야 합니다.

4.26.1.37 CYCLE806 - 보간 선삭

참고

CYCLE806을 이용한 보간 선삭은 라이센스가 필요한 옵션입니다.

구문

```
CYCLE806(<_MODE>, <S_XC>, <S_YC>, <_STA>, <_SV1>, <_SV2>,
<_SDIR>, <_AMODE>, <_DMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1		<_MODE>	INT	모드		
				UNITS:	선택/선택 해제	
					0 =	보간 선삭 선택 해제
					1 =	보간 선삭 선택

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
2	XC	<S_XC>	REAL	회전 중심 X		
3	YC	<S_YC>	REAL	회전 중심 Y		
4	Gamma	<_STA>	REAL	감마 각도(공구 축을 중심으로 회전); 예약됨		
5	S	<_SV1>	REAL	일정한 스핀들 속도 또는 일정한 절삭 속도(<_AMODE> UNITS 참조)		
6	PS	<_SV2>	REAL	일정한 절삭 속도로 최대 속도(예약됨)		
7		<_SDIR>	INT	예비		
					0 =	공구로부터 회전 방향
					3 =	회전 방향 M3
					4 =	회전 방향 M4
8		<_AMODE>	INT	대체 모드		
				UNITS:	스핀들 이동(<_SV1>)	
					0 =	일정한 스핀들 속도
					1 =	주속 일정 제어
9		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/18/19	
					1 =	G17 (싸이클에서만 활성 상태)
10	TC	<S_TRA>	STRING	• 변환 이름 • 비어 있음: 첫 번째 키네마틱 변환 TRAINT 사용		

4.26.1.38 CYCLE830 - 심공 드릴링 2

구문

```
CYCLE830(<RTP>, <RFP>, <SDIS>, <_DP>, <FDEP>, <_DAM>, <DTB>,
<DTS>, <FRF>, <VARI>, <_MDEP>, <_VRT>, <_DTD>, <_DIS1>, <S_FP>,
<S_SDAC2>, <S_SV2>, <S_FB>, <_SDAC>, <_SV1>, <S_SPOS>, <S_ZA>,
<S_FA>, <S_ZP>, <S_FS>, <S_ZS1>, <S_ZS2>, <S_N>, <S_ZD>, <S_FD>,
<S_FR>, <S_SDAC3>, <S_SV3>, <S_CON>, <S_COFF>, <_GMODE>,
<_DMODE>, <_AMODE>, <S_AMODE2>, <S_AMODE3>, <S_ZPV>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<RTP>	REAL	후퇴 평면 (abs)
2	Z0	<RFP>	REAL	원점 (abs)
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<_DP>	REAL	최종 드릴링 깊이 abs/inc (<_AMODE>UNITS 참조)
5	D	<FDEP>	REAL	기준점 대비 또는 파일럿 홀 깊이 대비 절대식 또는 증분식 칩 분쇄/제거를 위한 첫 번째 드릴링(사전드릴링 포함/미포함) 깊이 (<_AMODE> TEN THOUSANDS 참조)
6	DF	<_DAM>	REAL	각 추가 절입 (하강 절대치/백분율) 에 대한 절대치/백분율, <_AMODE> HUNDRED THOUSANDS 참조
7	DTB	<DTB>	REAL	각 드릴링 깊이에서 드웰 시간 (<_AMODE> TENS 참조)
8	DTS	<DTS>	REAL	칩 제거 중 시작점에서의 드웰 시간 (<_AMODE> HUNDREDS 참조)
9	FD1	<FRF>	REAL	최초 절입의 이송 속도에 대한 백분율 (<_AMODE> TEN MILLION 참조)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
10		<VARI>	INT	가공		
				UNITS:	칩 분쇄/칩 제거	
					0 =	한 번에 절삭
					1 =	칩 분쇄
					2 =	칩 제거
					3 =	칩 분쇄 및 칩 제거
				TENS:	칩 제거 중 후퇴	
					0 =	파일럿 홀 깊이까지
					1 =	안전 거리로
				HUNDREDS:	최초 절삭을 소프트 절삭	
					0 =	아니오
					1 =	예
				THOUSANDS:	관통 드릴링	
					0 =	아니오
					1 =	예
				TEN THOUSANDS:	사전 드릴링/파일럿 홀	
					0 =	사전 드릴링 없음
					1 =	사전 드릴링 있음
					2 =	파일럿 홀 있음
				HUNDRED THOUSANDS:	후퇴	
					0 =	파일럿 홀 깊이까지
					1 =	후퇴 평면
11	V1	<_MDEP>	REAL	최소 증분 절입 (하강 백분률에만 해당)		
12	V2	<_VRT>	REAL	각 증분 가공 단계 후 후퇴 거리 (칩 분쇄만 해당)		
					0 =	디폴트 값 1 mm
					> 0 =	가변 후퇴 거리
13	DT	<_DTD>	REAL	최종 드릴링 깊이에서 드웰 시간 (<_AMODE> THOUSANDS 참조)		
14	V3	<_DIS1>	REAL	칩 제거만을 위한 증분식 제한 거리(<_AMODE> ONEMILLION 참조)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
15	FP	<S_FP>	REAL	파일럿 홀 안까지 이동할 때 적용되는 이송 속도 값 또는 % (<S_AMODE2> HUNDREDS와 연계)		
16		<S_SDAC2>	INT	접근 중 스핀들 회전 방향		
					3 =	M3
					4 =	M4
					5 =	M5 (디폴트)
17	SP	<S_SV2>	REAL	접근 방법	일정한 스핀들 속도 (<S_AMODE2> TEN MILLION 참고)	
	V4				주속 일정 제어	
				드릴링 속도의 %로 표시되는 스핀들 속도		
18	F	<S_FB>	REAL	드릴링 속도 (<S_AMODE2> UNITS 참조)		
19		<_SDAC>	REAL	드릴링 중 스핀들 회전 방향		
					3 =	M3
					4 =	M4
20	S	<_SV1>	REAL	드릴링 방법	일정한 스핀들 속도 (<S_AMODE2> ONE MILLION 참고)	
	V5				주속 일정 제어	
21	SPOS	<S_SPOS>	REAL	스핀들 위치, M5로 접근하는 경우만 해당		
22	ZA	<S_ZA>	REAL	원점 기준 증분 사전 드릴링 깊이 또는 절대 사전 드릴링 깊이 (<S_AMODE3> UNITS 참조)		
23	FA	<S_FA>	REAL	사전 드릴링 이송 속도 값 또는 % (<S_AMODE2> TENS와 연계)		
24	ZP	<S_ZP>	REAL	원점 기준 증분 파일럿 홀, 절대 파일럿 홀 또는 홀 직경 팩터 (<S_AMODE3> TENS 참조)		
25	FS	<S_FS>	REAL	값 또는 %로 표시되는 첫 번째 절삭 이송 속도 (<S_AMODE2> THOUSANDS와 연계)		
26	ZS1	<S_ZS1>	REAL	일정한 이송 속도 시 각 최초 절삭의 깊이 (inc)		
27	ZS2	<S_ZS2>	REAL	이송 속도 증가 시 각 최초 절삭의 깊이 (inc)		
28	N	<S_N>	INT	각 칩 제거 이전 칩 분쇄 스트로크 횟수		
29	ZD	<S_ZD>	REAL	최종 드릴링 깊이 대비 남은 증분 드릴링 깊이 또는 절대 드릴링 깊이 (<S_AMODE3> HUNDREDS 참조)		
30	FD	<S_FD>	REAL	값 또는 %로 표시되는 남은 드릴링 이송 속도 (<S_AMODE2> TEN THOUSANDS와 연계)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
31	FR	<S_FR>	REAL	후퇴 이송 속도 (<S_AMODE2> HUNDRED THOUSANDS와 연계)		
32		<S_SDAC3>	INT	후퇴 중 스핀들 회전 방향		
					3 =	M3
					4 =	M4
					5 =	M5
33	SR	<S_SV3>	REAL	후퇴 방법	일정한 스핀들 속도 (<S_AMODE2> HUNDRED MILLION 참고)	
	V6				주속 일정 제어	
				드릴링 속도의 %로 표시되는 스핀들 속도		
34	절삭유 ON	<S_CON>	STRING[10]	절삭유 ON, M 명령 또는 서브프로그램 호출		
35	절삭유 OFF	<S_COFF>	STRING[10]	절삭유 OFF, M 명령 또는 서브프로그램 호출		
36		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	팁/직경을 기준으로 한 드릴링 깊이	
					0 =	팁
					1 =	직경

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
37		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	예비용	
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 베이직

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
38		<_AMODE>	INT	대체 모드 1		
				UNITS:	드릴링 깊이 = 최종 드릴링 깊이 Z1 (abs/inc)	
					0 =	증분치
					1 =	절대치
				TENS:	각 드릴링 깊이에서 드웰 시간 DTB (초/회전)	
					0 =	초
					1 =	회전
				HUNDREDS:	칩 제거 드웰 시간 DTS (초/회전)	
					0 =	초
					1 =	회전
				THOUSANDS:	최종 드릴링 깊이에서 드웰 시간 DT (초/회전)	
					0 =	초
					1 =	회전
				TEN THOUSANDS:	1차 드릴링 깊이 D abs/inc	
					0 =	증분치
					1 =	절대치
				HUNDRED THOUSANDS:	각 추가 절입 (하강) 에 대한 절대치/백분율 DF	
					0 =	절대치
					1 =	백분율 (0.001 ~ 100%)
				ONE MILLION:	제한 거리 V3 자동/수동	
					0 =	자동 (싸이클에서 계산)
					1 =	수동 (프로그래밍된 값)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
39		<S_AMODE2>	INT	대체 모드 2		
				UNITS:	UNITS: 드릴링 이송 속도 F	
					0 =	F/min
					1 =	F/rev
				TENS:	사전 드릴링 이송 속도 FA 평가	
					0 =	드릴링 이송 속도의 %로 표시
					1 =	F/min
					2 =	F/rev
				HUNDREDS:	파일럿 홀 안까지 이동 시 적용되는 이송 속도 FP 평가	
					0 =	드릴링 이송 속도의 %로 표시
					1 =	F/min
					2 =	F/rev
				THOUSANDS:	최초 절삭 이송 속도 FS 평가	
					0 =	드릴링 이송 속도의 %로 표시
					1 =	F/min
					2 =	F/rev
				TEN THOUSANDS:	관통 드릴링 이송 속도 FD 평가	
					0 =	드릴링 이송 속도의 %로 표시
					1 =	F/min
					2 =	F/rev
				HUNDRED THOUSANDS:	후퇴 이송 속도 FR	
					0 =	F/min
					1 =	급 이송
				ONE MILLION:	드릴링 - 스핀들 속도/절삭 속도 (S/V5)	
					0 =	일정한 스핀들 속도
					1 =	주속 일정 제어
				TEN MILLIONS:	스핀들 속도/절삭 속도 (SP/V4) 로 접근	
					0 =	일정한 스핀들 속도

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
					1 =	주속 일정 제어
					2 =	드릴링 속도의 %로 표시되는 스핀들 속도
				HUNDRED MILLIONS:	후퇴 - 스핀들 속도/절삭 속도 (SR/V6)	
					0 =	일정한 스핀들 속도
					1 =	주속 일정 제어
					2 =	드릴링 속도의 %로 표시되는 스핀들 속도
40		<S_AMODE3>	INT	대체 모드 3		
				UNITS:	드릴링 깊이 ZA (abs/inc)	
					0 =	증분치
					1 =	절대치
				TENS:	파일럿 홀 깊이 ZP	
					0 =	증분치
					1 =	절대치
					2 =	홀 직경의 팩터
				HUNDREDS:	남은 드릴링 깊이 ZD (abs/inc)	
					0 =	증분치
					1 =	절대치
41	ZPV	<S_ZPV>	REAL	파일럿 홀 깊이로부터 증분식 제한 거리		

4.26.1.39 CYCLE832 - 고속 가공 설정

구문

```
CYCLE832(<S_TOL>, <S_TOLM>, <S_OTOL>)
```

참고

CYCLE832를 사용하려면 기계를 스타트업할 때 필요한 최적화 작업을 해야 합니다. 여기에는 가공 공정에 관련된 축의 최적화 및 NCU 설정 (사전 제어, 저크 제한 등) 이 포함됩니다.

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	공차	<S_TOL>	REAL	형상 공차 형상 공차는 기하 축의 축 공차에 해당합니다.		
2		<S_TOLM>	INT	가공 형식 (테크놀로지)		
				UNITS:		
					0 =	선택 취소
					1 =	정삭
					2 =	중삭(준정삭)
					3 =	황삭
					4 =	정삭(정밀)
				TENS:		
					0 =	호환성[1] 또는 오리엔테이션 공차 없음
					1 =	파라미터 <S_OTOL> 상의 오리엔테이션 허용 오차
				백 단위 ... 십만 단위	호환성의 이유로 할당됨	
				ONE MILLION:		
					0 =	호환성 최선의 사용 가능한 주형 제작 기능은 자동적으로 사용됩니다. • 옵션 상단 표면 비활성화: ▯ 돌출 표면 • 옵션 상단 표면 활성화: ⇒ 스무딩이 있는 상단 표면
					1 =	스무딩이 없는 상단 표면
					2 =	스무딩이 있는 상단 표면

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
3	ORI 공차	<S_OTOL>	REAL	오리엔테이션 공차 또는 버전 식별자 CYCLE832 공작물 오리엔테이션을 위한 공차 파라미터 이 파라미터는 다이나믹 오리엔테이션 변환을 실행하는 기계에서 고속 가공 프로그램 (예: 5축 가공) 을 실행할 때 필요합니다. 파라미터는 <S_OTOL> **반드시** 프로그래밍되어야 합니다. 공구 오리엔테이션이 없는 프로그램을 위해 3축 기계에 적용할 때도 이 사항이 적용됩니다(<S_OTOL> = 1).

1) 싸이클 설정 데이터 SD55451 ... SD55454에서 파생된 오리엔테이션 허용 오차(다이나믹 응답 모드에 대한 오리엔테이션 허용 오차) 또는 SD55445 ... SD55449(다이나믹 응답 모드에 대한 형상 허용 오차)를 SD55441 ... SD55444 범위의 값으로 곱합니다 .
추가 정보: SINUMERIK Operate 스타트업 매뉴얼

일반 문자 입력

싸이클 호출의 가독성을 개선하기 위해서는 파라미터 <S_TOLM> (가공 유형) 역시 일반 텍스트로 입력할 수 있습니다. 일반 텍스트는 언어로부터 독립되어 있습니다. 아래 문자들을 입력할 수 있습니다.

_OFF	for	0	선택 취소
_FINISH	for	1	정삭
_SEMIFIN	for	2	중삭
_ROUGH	for	3	황삭
_PRECISION	for	4	정삭
_ORI_FINISH	for	11	= 오리엔테이션 공차를 입력하여 정삭
_ORI_SEMIFIN	for	12	오리엔테이션 공차를 입력하여 준정삭
_ORI_ROUGH	for	13	오리엔테이션 공차를 입력하여 황삭
_ORI_PRECISION	for	14	오리엔테이션 공차를 입력하여 정삭
_TOP_SURFACE_SMOOTH_OFF	for	1000000	스무딩이 없는 상단 표면
_TOP_SURFACE_SMOOTH_ON	for	2000000	스무딩이 있는 상단 표면

상단 표면의 일반 텍스트 입력을 위해서는 일반 텍스트가 다음 예에서 보여지는 것처럼 조합되어야 합니다:

```
CYCLE832(0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

참고

일반 텍스트는 G 그룹 59 (경로 보간을 위한 다이나믹 모드)의 기능 이름을 기준으로 합니다. 일반 텍스트를 입력하여 3축 기계와 다축 오리엔테이션 변환 (TRAORI) 을 수행하는 기계를 명확히 구분할 수 있습니다.

CYCLE832 선택 취소

CYCLE832의 선택을 취소하면 파라미터 <S_TOL>을 0으로 설정해야 합니다.

예: CYCLE832(0,0,1)

CYCLE832의 선택을 취소하는 경우에는 CYCLE832() 구문도 허용됩니다.

예

예 1: 오리엔테이션 변환 없이 3축 기계에서 CYCLE832 실행

a) 일반 텍스트를 입력하여 싸이클 호출

프로그램 코드	설명
G710	; 치수는 미터 단위입니다.
CYCLE832(0.004,_FINISH,1)	; CYCLE832 호출: 형상 공차 = 0.004 mm, 가공 형식: 정삭
...	; 고속 가공 프로그램 실행

b) 일반 텍스트를 입력하지 않고 싸이클 호출

프로그램 코드	설명
G710	; 상기 참조
CYCLE832(0.004,1,1)	; 상기 참조
...	; 상기 참조

예 2: 오리엔테이션 변환을 적용하여 5축 기계에서 CYCLE832 실행

a) 일반 텍스트를 입력하여 싸이클 호출 및 선택 취소

프로그램 코드	설명
G710	; 치수는 미터 단위입니다.

프로그램 코드	설명
TRAORI	; 오리엔테이션 변환 활성화.
CYCLE832(0.3,_ORI_ROUGH,0.8)	; CYCLE832 호출: 형상 공차 = 0.3 mm, 가공 형식: 오리엔테이션 공차는 0.8°로 입력하여 CYCLE832 호출
...	; 고속 가공 프로그램 실행
CYCLE832(0,_OFF,1)	; 형상 공차 = 0, 가공 형식: CYCLE832 선택 취소, 오리엔테이션 공차 = 0°

b) 일반 텍스트를 입력하지 않고 싸이클 호출 및 선택 취소

프로그램 코드	설명
G710	; 상기 참조
TRAORI	; 상기 참조
CYCLE832(0.3,13,0.8)	; 상기 참조
...	; 상기 참조
CYCLE832(0,0,1)	; 상기 참조

4.26.1.40 CYCLE840 - 태핑 척을 사용한 태핑

구문

```
CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>,
<ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>,
<_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	RP	<RTP>	REAL	후퇴 평면 (abs)
2	Z0	<RFP>	REAL	원점 (abs)
3	SC	<SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)
4	Z1	<DP>	REAL	드릴링 깊이 (abs), <_AMODE> 참조
5	Z1	<DPR>	REAL	드릴링 깊이 (inc), <_AMODE> 참조
6	DT	<DTB>	REAL	드릴링 깊이에서 드웰 시간 (초) / 후퇴 후 안전 거리, <ENC> 참조
7		<SDR>	INT	후진 시 회전 방향
8	SDE	<SDAC>	INT	싸이클 종료 후 회전 방향

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미			
9		<ENC>	INT	스핀들 장착된 엔코더를 사용한 태핑 (G33)/스핀들 장착된 엔코더를 사용하지 않은 태핑 (G63)			
					0 =	스핀들이 장착된 엔코더 사용	- <MPIT>/<PIT> 로부터의 피치 - DT 적용 안함
					20 =	스핀들이 장착된 엔코더 사용	- <MPIT>/<PIT> 로부터의 피치 - 안전 거리로 후퇴 후 DT 적용
					11 =	스핀들이 장착된 엔코더 미사용	- <MPIT>/<PIT> 로부터의 피치 - 드릴링 깊이에 DT 적용
					1 =	스핀들이 장착된 엔코더 미사용	- 프로그래밍된 이송 속도와의 피치 - 드릴링 깊이에서 DT 사용 (이송 속도 = 속도 · 피치)
10		<MPIT>	REAL	"ISO 미터법" 전용의 나사 크기 (피치는 실행 도중 내부에서 계산됨) 값 범위: 3 ~ 48 (M3 ~ M48의 경우), <PIT>를 대신함			
11		<PIT>	REAL	피치 값, 단위는 <_PITA> 참조 값 범위: > 0, MPIT를 대신함			
12		<_AXN>	INT	드릴링 축	0 =	세 번째 기하 축	
					1 =	첫 번째 기하 축	
					2 =	두 번째 기하 축	
					≥ 3 =	세 번째 기하 축	

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미			
13		<_PITA>	INT	피치 단위 (<PIT> 및 <MPIT> 평가)			
					0 =	피치 (mm)	- 평가<MPIT>/<PIT>
					1 =	피치 (mm)	- 평가<PIT>
					2 =	피치 (TPI)	- <PIT> 평가 (인치당 스레드 수)
					3 =	피치 (인치)	- 평가<PIT>
					4 =	MODULUS	- 평가<PIT>
14		<_TECHNO>	INT	테크놀로지[1]			
				UNITS:	정위치 정지 반응		
					0 =	싸이클 호출 전과 같이 정위치 정지 반응 활성 상태	
					1 =	정위치 정지 G601	
					2 =	정위치 정지 G602	
					3 =	정위치 정지 G603	
				TENS:	피드포워드 제어		
					0 =	싸이클 호출 전과 같이 피드포워드 제어 사용/사용 안 함	
					1 =	피드포워드 제어 사용 FFWON	
					2 =	피드포워드 제어 사용 안함 FFWON	
15		<_PITM>	STRING[15]	피치 입력을 위한 마커로서의 문자열[2]			
16		<_PTAB>	STRING[5]	나사 테이블의 문자열 ("", "ISO", "BSW", "BSP", "UNC")[2]			
17		<_PTABA>	STRING[20]	나사 테이블에서 선택하기 위한 문자열 (예: "M 10", "M 12", ...)[2]			
18		<_GMODE>	INT	예비용			

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
19		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:	호환성 모드 (재컴파일 화면 형식에만 해당), MD 52216 bit0 = 1인 경우[1]	
					0 =	테크놀로지 파라미터가 표시됨 (호환성): TECHNO 파라미터 유효
					1 =	테크놀로지 파라미터가 표시되지 않음: "싸이클 호출 이전과 같이" 테크놀로지 활성 상태
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단
20		<_AMODE>	INT	대체 모드		
				UNITS:	드릴링 깊이 Z1 (abs/inc)	
					0 =	호환성, <DP>/<DPR> 프로그래밍에서
					1 =	증분치
					2 =	절대치

[1] 테크놀로지 필드는 설정 날짜 SD52216 MCS_FUNCTION_MASK_DRILL에 따라 숨겨질 수 있습니다.

[2] 파라미터 15, 16 및 17은 화면 형식 나사 테이블에서 나사 선택에만 사용됩니다. 싸이클 런타임에서는 싸이클 정의를 통해 나사 테이블에 액세스할 수 없습니다.

4.26.1.41 CYCLE899 - 개방형 슬롯

구문

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>,
<_CDIR>, <_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>,
<_ZFS>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	RP	<_RTP>	REAL	후퇴 평면 (abs)		
2	Z0	<_RFP>	REAL	공구 축 원점 (abs)		
3	SC	<_SDIS>	REAL	안전 거리 (원점에 추가됨, 부호 없이 입력)		
4	Z1	<_DP>	REAL	슬롯 깊이 (abs/inc), <_AMODE> 참조		
5	L	<_LENG>	REAL	슬롯 길이 (inc)		
6	W	<_WID>	REAL	슬롯 너비 (inc)		
7	X0	<_PA>	REAL	원점/시작점 첫 번째 축 (abs)		
8	Y0	<_PO>	REAL	원점/시작점 두 번째 축 (abs)		
9	α0	<_STA>	REAL	첫 번째 축을 중심으로 한 회전 각도		
10	DZ	<_MID>	REAL	최대 절입 깊이 (inc) – 나선 밀링만 해당		
11	DXY	<_MIDA>	REAL	최대 평면 절입, <_AMODE> 참조		
12	UXY	<_FAL>	REAL	정삭 여유, 평면		
13	UZ	<_FALD>	REAL	정삭 여유, 깊이		
14	F	<_FFP1>	REAL	이송 속도		
15		<_CDIR>	INT	밀링 방향		
				UNITS:		
					0 =	하향 절삭
					1 =	상향 절삭
					4 =	교대

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
16		<_VARI>	INT	가공		
				UNITS:		
					1 =	황삭
					2 =	정삭
					3 =	베이스 정삭
					4 =	에지 정삭
					5 =	중삭
					6 =	모따기
				TENS:	예비용	
				HUNDREDS:	예비용	
				THOUSANDS:		
					1 =	나선 밀링
					2 =	플런지 절삭
17		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택/시작점 계산만	
					1 =	일반 가공
				THOUSANDS:	중심/에지를 지나는 치수 지정	
					0 =	중심을 통한 치수 지정
					1 =	에지를 사용한 "좌측" 치수 지정 (첫 번째 축 "-" 방향)
					2 =	에지를 사용한 "우측" 치수 지정 (첫 번째 축 "+" 방향)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
18		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	---	예비용
				HUNDREDS:	---	예비용
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단
19		<_AMODE>	INT	대체 모드		
				UNITS:	슬롯 깊이 Z1	
					0 =	절대치
					1 =	증분치
				TENS:	평면 절입 단위 (<_MIDA>) DXY	
					0 =	mm
					1 =	공구 직경의 %
				HUNDREDS:	모따기를 위한 삽입 깊이 ZFS	
					0 =	절대치
					1 =	증분치
20		<_UMODE>	INT	예비용		
21	FS	<_FS>	REAL	모따기 너비 (inc)		
22	ZFS	<_ZFS>	REAL	모따기에서 삽입 깊이 (공구 팁) (abs/inc), <_AMODE> 참조		

4.26.1.42 CYCLE930 - 홈

구문

```
CYCLE930(<_SPD>, <_SPL>, <_WIDG>, <_WIDG2>, <_DIAG>, <_DIAG2>,
<_STA>, <_ANG1>, <_ANG2>, <_RCO1>, <_RCI1>, <_RCI2>, <_RCO2>,
<_FAL>, <_IDEP1>, <_SDIS>, <_VARI>, <_DN>, <_NUM>, <_DBH>,
<_FF1>, <_NR>, <_FALX>, <_FALZ>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
1	X0	<_SPD>	REAL	평면 축의 원점 (항상 직경)		
2	Z0	<_SPL>	REAL	세로 축의 원점		
3	B1	<_WIDG>	REAL	홈 하단 너비		
4	B2	<_WIDG2>	REAL	홈 상단 너비 (인터페이스만 해당)		
5	T1	<_DIAG>	REAL	원점에서 홈 깊이 abs 및 세로 가공의 경우 = 직경, 기타의 경우 inc		
6	T2	<_DIAG2>	REAL	원점 반대편의 홈 깊이 (인터페이스만 해당), abs 및 세로 가공의 경우 = 직경, 기타의 경우 inc		
7	α0	<_STA>	REAL	경사 각도 (-180 ≤ <_STA> ≤ 180)		
8	α1	<_ANG1>	REAL	원점에 따라 결정되는 홈 측면의 측면 각도 1 (0 ≤ <_ANG1> < 90)		
9	α2	<_ANG2>	REAL	원점 반대편의 측면 각도 2 (0 ≤ <_ANG2> < 90)		
10	R1/FS1	<_RCO1>	REAL	라운딩 반경 또는 모따기 너비 1, 원점에서 외부		
11	R2/FS2	<_RCI1>	REAL	라운딩 반경 또는 모따기 너비 2, 원점에서 내부		
12	R3/FS3	<_RCI2>	REAL	라운딩 반경 또는 모따기 너비 3, 원점 반대편 내부		
13	R4/FS4	<_RCO2>	REAL	라운딩 반경 또는 모따기 너비 4, 원점 반대편 외부		
14	U	<_FAL>	REAL	X 및 Z 방향의 정삭 여유, <_VARI> (TEN THOUSANDS) 참조 (부호 없이 입력)		
15	D	<_IDEP1>	REAL	삽입 시 최대 깊이 절입 (부호 없이 입력)		
					0 =	전제 깊이까지 직접 1차 절삭
					> 0 =	1차 절삭 <_IDEP1>, 2차 절삭 2 <_IDEP1> 등
16	SC	<_SDIS>	REAL	안전 거리 (부호 없이 입력)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
17		<_VARI>	INT	가공 형식		
				UNITS:	예비용	
				TENS:	가공 공정	
					1 =	황삭
					2 =	정삭
					3 =	황삭과 정삭
				HUNDREDS:	세로/가로 외부/내부 +Z/+Z 및 +X/-X 포지셔닝	
					1 =	세로/외부 +Z
					2 =	가로/내부 -X
					3 =	세로/내부 +Z
					4 =	가로/내부 +X
					5 =	세로/외부 -Z
					6 =	가로/외부 -X
					7 =	세로/내부 -Z
					8 =	가로/외부 +X
				THOUSANDS:	원점 위치	
					0 =	상부 원점
					1 =	하부 원점
				TEN THOUSANDS:	정삭 여유의 효과 정의	
					0 =	형상과 평행한 정삭 여유 U
					1 =	분리된 UX 및 UZ 정삭 여유
18		<_DN>	INT	공구의 2번째 절삭날의 D 번호		
					> 0 =	홈 가공 공구의 2번째 날의 공구 옵셋을 위한 D 번호
					0 =	2번째 절삭날이 프로그램되지 않음
19	N	<_NUM>	INT	홈 개수 (0 = 1 홈)		
20	DP	<_DBH>	REAL	홈 간격 (<_NUM> > 1일 때만 필요)		
21	F	<_FF1>	REAL	이송 속도		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
22		<_NR>	INT	홈 형상 식별은 형상 선택용 수직 소프트 키에 해당합니다.		
					0 =	모따기/라운딩이 없는 90° 측면
					1 =	모따기/라운딩을 포함한 경사 측면 (α0 사용 안 함)
					2 =	1과 같으나 테이퍼 위에서 (α0 사용)
23	UX	<_FALX>	REAL	X축의 정삭 여유, <_VARI> (TEN THOUSANDS) 참조 (부호 없이 입력)		
24	UZ	<_FALZ>	REAL	Z축의 정삭 여유, <_VARI> (TEN THOUSANDS) 참조 (부호 없이 입력)		
25		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미	
26		<_AMODE>	INT	대체 모드	
				UNITS:	홈 상단 치수 지정 (인터페이스만 해당)
					0 = 원점에
					1 = 원점의 반대편에
				TENS:	깊이
					0 = 절대치
					1 = 증분치
				HUNDREDS:	너비 치수 지정 (인터페이스만 해당)
					0 = 외경에서 (상단)
					1 = 내경에서 (하단)
				THOUSANDS:	반경/모따기 1 (<_RCO1>)
					0 = 반경
					1 = 모따기
				TEN THOUSANDS:	반경/모따기 2 (<_RCI1>)
					0 = 반경
					1 = 모따기
				HUNDRED THOUSANDS:	반경/모따기 3 (<_RCI2>)
					0 = 반경
					1 = 모따기
				ONE MILLION:	반경/모따기 4 (<_RCO2>)
					0 = 반경
					1 = 모따기

4.26.1.43 CYCLE940 – 언더컷 형상 E 및 F / 언더컷 나사

CYCLE940 싸이클을 사용하여 다양한 언더컷을 프로그래밍할 수 있습니다. 일부 경우에는 파라미터 설정에 따라 크게 다릅니다.

테이블의 추가 열은 어떤 언더컷 유형에 어떤 파라미터가 필요한지를 나타냅니다. 싸이클 화면 양식에서 수직 선택 소프트 키에 해당합니다.

- E: 언더컷 형상 E
- F: 언더컷 형상F
- A-D: DIN 나사 언더컷 (A-D 형상)
- T: 나사 언더컷 (임의의 형상 정의)

구문

```
CYCLE940(<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>,
<_VARI>, <_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>,
<_FAL>, <_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>,
<_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	형상 프로그래밍				의미		
				E	F	A-D	T			
1	X0	<_SPD>	REAL	x	x	x	x	평면 축의 원점 (항상 직경)		
2	Z0	<_SPL>	REAL	x	x	x	x	세로 축의 원점 (abs)		
3	FORM	<_FORM>	CHAR	x	x	x	x	언더컷 형상 (대문자, 예: "T") 선택, 언더컷 값을 가져와야 할 테이블		
									A =	외부, 참조 DIN76, A = 정상
									B =	외부, 참조 DIN76, B = 짧음
									C =	내부, 참조 DIN76, C = 정상
									D =	내부, 참조 DIN76, D = 짧음
									E =	참조 DIN509
									F =	참조 DIN509
									T=	자유 형상

번호	파라미터 마스크	내부 파라미터	데이터 유형	형상 프로그래밍				의미		
4	POSITION	<_LAGE>	INT	x	x	x	x	언더컷의 위치 (평행 Z)	0 =	외부 +Z: ____|
									1 =	외부 -Z: |____/
									2 =	내부 +Z: /-----|
									3 =	내부 -Z: |-----\
5	SC	<_SDIS>		x	x	x	x	안전 거리 (inc)		
6	F	<_FFP>		x	x	x	x	가공 속도 (mm/rev)		
7		<_VARI>	INT	-	-	x	x	가공 형식		
								UNITS:	가공	
									1 =	황삭
									2 =	정삭
									3 =	황삭 + 정삭
								TENS:	가공 방식	
									0 =	형상에 대해 수평
									1 =	길이 방향
				언더컷 형상 E와 F는 항상 정삭과 같이 단일 통과로 가공됩니다.						
8	X1	<_EPD>		x	x	-	-	공차 X (abs/inc), <_AMODE> 참조		
				-	-	-	x	언더컷 깊이 (abs/inc), <_AMODE> 참조		
9	Z1	<_EPL>		-	x	-	-	공차 Z		
				-	-	-	x	언더컷 너비 (abs/inc), <_AMODE> 참조		
10	R1	<_R1>		-	-	-	x	경사의 라운딩 반경		
11	R2	<_R2>		-	-	-	x	코너의 라운딩 반경		
12	α	<_STA>		-	-	x	x	절입 각도		
13	VX	<_VRT>		x	x	-	-	교차 피드 X (abs/inc), <_AMODE> 참조		
				-	-	x	x	정삭 시의 교차 피드 X (abs/inc), <_AMODE> 참조		
14	D	<_MID>		-	-	x	x	깊이 절입		
15	U	<_FAL>		-	-	x	x	형상과 평행하는 정삭 여유, <_AMODE> 참조		
16	UX	<_FALX>		-	-	x	x	정삭 여유 X		
17	UZ	<_FALZ>		-	-	x	x	정삭 여유 Z		

번호	파라미터 마스크	내부 파라미터	데이터 유형	형상 프로그래밍	의미			
18	P	<_PITI>	INT	- - x -	피치 선택, 형상 A-D, M1 ... M68에 해당			
					0 = 0.20 1 = 0.25 2 = 0.30 3 = 0.35 4 = 0.40 5 = 0.45	6 = 0.50 7 = 0.60 8 = 0.70 9 = 0.75 10 = 0.80 11 = 1.00	12 = 1.25 13 = 1.50 14 = 1.75 15 = 2.00 16 = 2.50 17 = 3.00	18 = 3.50 19 = 4.00 20 = 4.50 21 = 5.00 22 = 5.50 23 = 6.00
				x x - -	반경/깊이 선택, 형상 E, F			
					0 = 0.6 x 0.3 1 = 1.0 x 0.4 2 = 1.0 x 0.2 3 = 1.6 x 0.3	4 = 2.5 x 0.4 5 = 4.0 x 0.5 6 = 0.4 x 0.2 7 = 0.6 x 0.2	8 = 0.1 x 0.1 9 = 0.2 x 0.1	
19		<_PTAB>	STRING [5]		나사 테이블의 문자열 ("", "ISO", "BSW", "BSP", "UNC") (표면만 해당)			
20		<_PTABA>	STRING [20]		나사 테이블로부터의 선택을 위한 열 (예: "M 10", "M 12", ...) (표면만 해당)			
21		<_DMODE>	INT		표시 모드			
				x x x x	UNITS:	가공 평면 G17/G18/G19		
						0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태	
						1 =	G17 (싸이클에서만 활성 상태)	
						2 =	G18 (싸이클에서만 활성 상태)	
						3 =	G19 (싸이클에서만 활성 상태)	

번호	파라미터 마스크	내부 파라미터	데이터 유형	형상 프로그래밍	의미	
22		<_AMODE>	INT		대체 모드	
				x x - x	UNITS:	파라미터 <_EPD> 여유 X 또는 언더컷 깊이
						0 = 절대 (항상 직경)
						1 = 증분치
				x x - x	TENS:	파라미터 <_EPL> 여유 Z 또는 언더컷 너비
						0 = 절대치
						1 = 증분치
				x x x x	HUNDREDS:	파라미터 <_VRT> 교차 피드 X
						0 = 절대 (항상 직경)
						1 = 증분치
				- - x x	THOUSANDS:	정삭 여유
						0 = 형상과 평행한 정삭 여유 (<_FAL>)
						1 = 별도의 가공 여유 (<_FALX>/<_FALZ>)

4.26.1.44 CYCLE951 - 선삭

구문

```
CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>,
<_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>,
<_SDIS>, <_FF1>, <_NR>, <_DMODE>, <_AMODE>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	X0	<_SPD>	REAL	원점 (abs, 항상 직경)
2	Z0	<_SPL>	REAL	원점 (abs)
3	X1	<_EPD>	REAL	종점

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
4	Z1	<_EPL>	REAL	종점		
5	XM α1 α2	<_ZPD>	REAL	중간점, <_DMODE> (TENS) 참조		
6	ZM α1 α2	<_ZPL>	REAL	중간점, <_DMODE> (TENS) 참조		
7	위치	<_LAGE>	INT	선삭 가공 코너 위치	0 =	외부/뒤
					1 =	외부/앞
					2 =	내부/뒤
					3 =	내부/앞
8	D	<_MID>	REAL	절입 시 최대 깊이 절입		
9	UX	<_FALX>	REAL	X 방향 정삭 여유		
10	UZ	<_FALZ>	REAL	Z 방향 정삭 여유		
11		<_VARI>	INT	가공 형식		
				UNITS:	좌표계 상에서 선삭 방향 (길이 또는 너비)	
					1 =	길이 방향
					2 =	단면 방향
				TENS:		
					1 =	최종 가공 공차까지 황삭
					2 =	정삭
				HUNDREDS:	예비	
				THOUSANDS:	예비	
				TEN THOUSANDS:	예비	
12	R1/FS1	<_RF1>	REAL	라운딩 반경 또는 모따기 너비 1, <_AMODE> (TEN THOUSANDS) 참조		
13	R2/FS2	<_RF2>	REAL	라운딩 반경 또는 모따기 너비 2, <_AMODE> (HUNDRED THOUSANDS) 참조		
14	R3/FS3	<_RF3>	REAL	라운딩 반경 또는 모따기 너비 3, <_AMODE> (ONE MILLION) 참조		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
15	SC	<_SDIS>	REAL	안전 거리		
16	F	<_FF1>	REAL	황삭/정삭의 이송 속도		
17		<_NR>	INT	선삭 유형 확인 (형상 선택을 위한 수직 소프트 키에 해당):		
					0 =	선삭 1, 모따기/라운딩 없는 90° 코너
					1 =	선삭 2, 모따기/라운딩 있는 90° 코너
					2 =	선삭 3, 모따기/라운딩 있는 모든 코너
18		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	입력 형태 <_ZPD>/<_ZPL>	
					0 =	Xm/Zm
					1 =	Xm/α1
					2 =	Xm/α2
					3 =	α1/Zm
					4 =	α2/Zm
					5 =	α1/α2

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
21		<_AMODE>	INT	대체 모드		
				UNITS:	X 방향 중간점	
					0 =	절대, 직경에서 가로 축의 값
					1 =	증분, 반경에서 가로 축의 값
				TENS:	Z 방향 중간점	
					0 =	절대치
					1 =	증분치
				HUNDREDS:	X 방향 종점	
					0 =	절대, 직경에서 가로 축의 값
					1 =	증분, 반경에서 가로 축의 값
				THOUSANDS:	Z 방향 종점.	
					0 =	절대치
					1 =	증분치
				TEN THOUSANDS:	반경/모따기 1	
					0 =	반경
					1 =	모따기
				HUNDRED THOUSANDS:	반경/모따기 2	
					0 =	반경
					1 =	모따기
				ONE MILLION:	반경/모따기 3	
					0 =	반경
					1 =	모따기

4.26.1.45 CYCLE952 – 칩 제거 / 잔여 칩 제거 / 플런지 절삭 / 잔여 플런지 절삭 / 플런지 선삭 / 잔여 플런지 선삭

구문

```
CYCLE952(<_PRG>, <_CON>, <_CONR>, <_VARI>, <_F>, <_FR>, <_RP>,
<_D>, <_DX>, <_DZ>, <_UX>, <_UZ>, <_U>, <_U1>, <_BL>, <_XD>,
<_ZD>, <_XA>, <_ZA>, <_XB>, <_ZB>, <_XDA>, <_XDB>, <_N>, <_DP>,
<_DI>, <_SC>, <_DN>, <_GMODE>, <_DMODE>, <_AMODE>, <_PK>, <_DCH>,
<_FS>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미
1	PRG	<_PRG>	STRING[100]	선삭 가공 프로그램 이름
2	CON	<_CON>	STRING[100]	소재의 업데이트된 형상을 읽어올 프로그램 이름 (잔삭 가공의 경우)
3	CONR	<_CONR>	STRING[100]	소재의 업데이트된 형상 (<_AMODE> TEN THOUSANDS 참조)이 기록될 프로그램 이름

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
4		<_VARI>	INT	가공 형식		
				UNITS:	선삭 유형	
					1 =	길이 방향
					2 =	단면 방향
					3 =	형상에 대해 수평
				TENS:	가공 공정 (<_GMODE> HUNDREDS 참조)	
					1 =	황삭
					2 =	정삭
					3 =	예비용
					4 =	황삭, 2채널
					5 =	정삭, 2채널
				HUNDREDS:	가공 방향	
					1 =	가공 방향 X -
					2 =	가공 방향 X +
					3 =	가공 방향 Z -
					4 =	가공 방향 Z +
				THOUSANDS:	절입 방향	
					1 =	외부 X -
					2 =	내부 X +
					3 =	전면 Z -
					4 =	후면 Z +
				TEN THOUSANDS:	정삭 여유의 효과 정의	
					0 =	분리된 UX 및 UZ 정삭 여유
					1 =	형상과 평행한 정삭 여유 U
				HUNDRED THOUSANDS:	라운딩	
					0 =	호환성, 자동 라운딩
					1 =	형상에 라운딩 적용
					2 =	라운딩 안함
					3 =	자동 라운딩
				ONE MILLION:	릴리프 절삭	

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
					0 =	홈 가공 시 위치가 평가되지 않음, - 잔여 소재 및 홈 선삭, - 나머지
					1 =	릴리프 절삭 가공
					2 =	릴리프 절삭 가공 안 함
				TEN MILLIONS:	터닝 센터의 뒤쪽/앞쪽	
					0 =	터닝 센터의 앞쪽에서 가공
					1 =	예비용
5	F	<_F>	REAL	황삭/정삭의 이송 속도		
	FZ			절입 가로 좌표 홈 선삭		
6	FR	<_FR>	REAL	릴리프 절삭부로 삽입을 위한 이송 속도, 황삭		
	FX			절입 세로 좌표 홈 선삭		
7	RP	<_RP>	REAL	내부 가공을 위한 후퇴 평면 (abs, 항상 직경)		
8	D	<_D>	REAL	황삭 절입 (<_AMODE> UNITS 참조)		
9	DX	<_DX>	REAL	X 절입 (<_AMODE> UNITS 참조)		
10	DZ	<_DZ>	REAL	Z 절입 (<_AMODE> UNITS 참조)		
11	UX	<_UX>	REAL	정삭 여유 X, (<_VARI> TEN THOUSANDS 참조)		
12	UZ	<_UZ>	REAL	정삭 여유 Z, (<_VARI> TEN THOUSANDS 참조)		
13	U	<_U>	REAL	형상과 평행한 정삭 여유, (<_VARI> TEN THOUSANDS 참조)		
14	U1	<_U1>	REAL	정삭 시 추가 정삭 여유 (<_AMODE> THOUSANDS 참조)		
15	BL	<_BL>	INT	소재 정의	1 =	여유가 포함된 실린더
					2 =	정삭된 파트의 형상 공차
					3 =	소재의 형상이 지정됩니다
16	XD	<_XD>	REAL	소재 X의 정의 (<_AMODE> HUNDRED THOUSANDS 참조)		
17	ZD	<_ZD>	REAL	소재 Z의 정의 (<_AMODE> ONE MILLION 참조)		
18	XA	<_XA>	REAL	한계 1 X (abs, 항상 직경)		
19	ZA	<_ZA>	REAL	한계 1 Z (abs)		
20	XB	<_XB>	REAL	한계 2 X (<_AMODE> TEN MILLIONS 참조)		
21	ZB	<_ZB>	REAL	한계 2 Z (<_AMODE> HUNDRED MILLIONS 참조)		
22	XDA	<_XDA>	REAL	전면의 첫 번째 홈 가공을 위한 홈 가공 한계 1 (abs, 항상 직경)		
23	XDB	<_XDB>	REAL	전면의 두 번째 홈 가공을 위한 홈 가공 한계 2 (abs, 항상 직경)		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
24	N	<_N>	INT	슬롯 개수		
25	DP	<_DP>	REAL	홈 간격	일반 홈: Z 축과 평행	
					가로 축 홈: X 축과 평행	
26	DI	<_DI>	REAL	절입 중단을 위한 거리	0 =	인터럽션 없음
					> 0 =	인터럽션 적용
27	SC	<_SC>	REAL	장애물 회피를 위한 안전 거리, 증분		
28	D2	<_DN>	INT	두 번째 에지의 D 번호가 프로그램 되지 않은 경우 ⇒ D+1		

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
29		<_GMODE>	INT	형상 모드 (프로그래밍된 형상 데이터 평가)		
				UNITS:	예비용	
				TENS:	예비용	
				HUNDREDS:	가공 선택/시작점 계산만	
					0 =	일반 가공 (호환 모드는 필요 없음)
					1 =	일반 가공
					2 =	시작점 계산 - 가공 안 함 (ShopMill/ShopTurn의 호출에만 해당)
				THOUSANDS:	한계	
					0 =	아니오
					1 =	예
				TEN THOUSANDS:	한계 1 X 입력	
					0 =	아니오
					1 =	예
				HUNDRED THOUSANDS:	한계 2 X 입력	
					0 =	아니오
					1 =	예
				ONE MILLION:	한계 1 Z 입력	
					0 =	아니오
					1 =	예
				TEN MILLIONS:	한계 2 Z 입력	
					0 =	아니오
					1 =	예

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
30		<_DMODE>	INT	표시 모드		
				UNITS:	가공 평면 G17/G18/G19	
					0 =	호환성, 싸이클 호출 이전에 유효한 평면은 계속 활성 상태
					1 =	G17 (싸이클에서만 활성 상태)
					2 =	G18 (싸이클에서만 활성 상태)
					3 =	G19 (싸이클에서만 활성 상태)
				TENS:	테크놀로지 모드	
					1 =	형상과 함께 스톡 제거
					2 =	형상 홈 가공
					3 =	홈 선삭
				HUNDREDS:	잔삭 가공	
					0 =	아니오
					1 =	예
				THOUSANDS:	---	예비용
				TEN THOUSANDS:	싸이클 화면 형식에서 테크놀로지 스케일링 (페이지 1340)	
					0 =	입력: 전체
					1 =	입력: 간단
				HUNDRED THOUSANDS:	자동 프로그램 이름	
					0 =	아니오
					1 =	예

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
31		<_AMODE>	INT	대체 모드		
				UNITS:	절입 선택	
					0 =	형상과 평행한 선삭 가공을 위한 DX 및 DZ 절입
					1 =	D 절입
				TENS:	절입 방식	
					0 =	가변 절삭 깊이 (90 ... 100%)
					1 =	일정 절삭 깊이
				HUNDREDS:	절삭 분할	
					0 =	균일
					1 =	에지에 정렬
				THOUSANDS:	형상 공차 U1 선택, 이중 정삭	
					0 =	아니오
					1 =	예
				TEN THOUSANDS:	소재 선택 업데이트	
					0 =	아니오
					1 =	예
				HUNDRED THOUSANDS:	소재 XD에 대한 공차 선택	
					0 =	절대, 직경에서 가로 축의 값
					1 =	증분, 반경에서 가로 축의 값
				ONE MILLION:	소재 ZD에 대한 공차 선택	
					0 =	절대치
					1 =	증분치
				TEN MILLIONS:	한계 2 XB 선택	
					0 =	절대, 직경에서 가로 축의 값
					1 =	증분, 반경에서 가로 축의 값
				HUNDRED MILLION:	한계 2 ZB 선택	
					0 =	절대치
					1 =	증분치
				ONE BILLION:		
					0 =	선두 채널

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
					1 =	후속 채널
32		<_PK>	INT	기계에 사용 가능한 채널이 2개 이상인 경우 파트너 채널의 번호.		
33	DCH	<_DCH>	REAL	채널 옵셋		
34	FS	<_FS>	REAL	전체 가공 시 정삭 속도		

4.26.1.46 CYCLE4071 - 반전 지점에서 절입을 통한 세로 방향 연삭

구문

```
CYCLE4071(<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

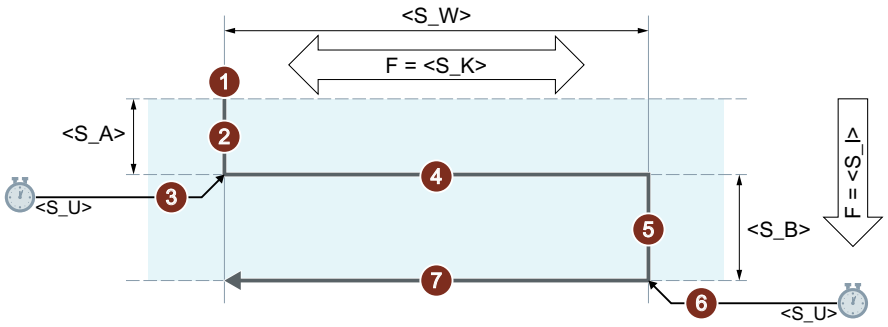
파라미터

번호	파라미터	데이터 유형	의미
1	<S_A>	REAL	시작 시 절입 깊이
2	<S_B>	REAL	종료 시 절입 깊이
3	<S_W>	REAL	연삭 너비
4	<S_U>	REAL	스파크아웃 시간
5	<S_I>	REAL	절입을 위한 이송 속도
6	<S_K>	REAL	가로 방향 절입을 위한 이송 속도
7	<S_H>	INT	반복 횟수
8	<S_A1>	AXIS	절입 축(옵션) 또는 첫 번째 기하 축
9	<S_A2>	AXIS	오실레이션 축(옵션) 또는 두 번째 기하 축

기능

이 싸이클은 절입 반복에서 사용됩니다. 시작과 종료 시 절입 깊이가 서로 다를 수 있습니다. 각 절입 간에는 접선 모션이 존재합니다.

순서



① 오실레이션 축의 현재 위치에서 싸이클 시작.

② 시작 <S_A>에서 절입 이송 속도 <S_I>로 절입 깊이까지 절입 축 이송.

③ 스파크아웃 시간을 이용한 스파크<S_U>

④ 이동 경로로 연삭 너비 <<S_W>>와 절입을 위한 이송 속도 <S_K>로 오실레이션 축 이송

⑤ 종료 <S_B>에서 절입을 위한 이송 속도 <S_I>로 절입 축을 절입 깊이까지 이송

⑥ 스파크아웃 시간을 이용한 스파크<S_U>

⑦ 이동 경로로 연삭 너비 <<S_W>>와 절입을 위한 이송 속도 <S_K>로 시작점까지 오실레이션 축 이송

반복 단계를 나타냅니다.

프로그래밍된 반복 횟수 <S_H>에 도달할 때까지 순서가 반복됩니다.

참고

이 순서는 단일 블록으로 중단할 수 없습니다.

예

다음 싸이클 파라미터를 사용한 두 왕복 운동의 실행:

- 시작 시 절입 깊이: 0.02 mm
- 종료 시 절입 깊이: 0.01 mm
- 스트로크: 100mm
- 스파크아웃 시간: 1 s
- 절입 이송 속도: 1 mm/min
- 가로 방향 이송 속도: 1000mm/min

- 반복 횟수: 2
- 오실레이션 축 및 절입 축: 표준 기하 축

프로그램 코드

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

4.26.1.47 CYCLE4072 - 반전 지점에서의 절입을 통한 세로 방향 연삭과 취소 신호

구문

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

파라미터

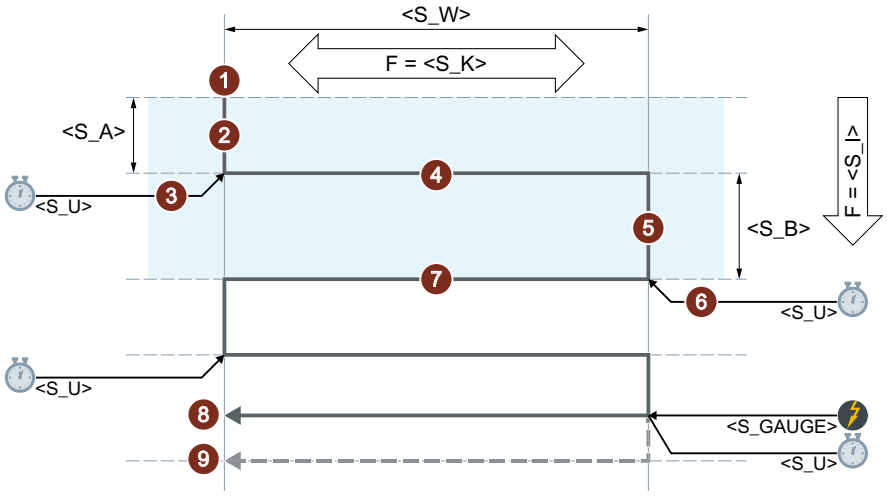
번호	파라미터	데이터 유형	의미
1	<S_GAUGE>	STRING	절입 취소 조건: 1. 고속 입력 번호 2. 논리 표현식
2	<S_A>	REAL	시작 시 절입 깊이
3	<S_B>	REAL	종료 시 절입 깊이
4	<S_W>	REAL	연삭 너비
5	<S_U>	REAL	스파크아웃 시간
6	<S_I>	REAL	절입을 위한 이송 속도
7	<S_K>	REAL	가로 방향 절입을 위한 이송 속도
8	<S_H>	INT	반복 횟수
9	<S_A1>	AXIS	절입 축(옵션) 또는 첫 번째 기하 축
10	<S_A2>	AXIS	오실레이션 축(옵션) 또는 두 번째 기하 축

기능

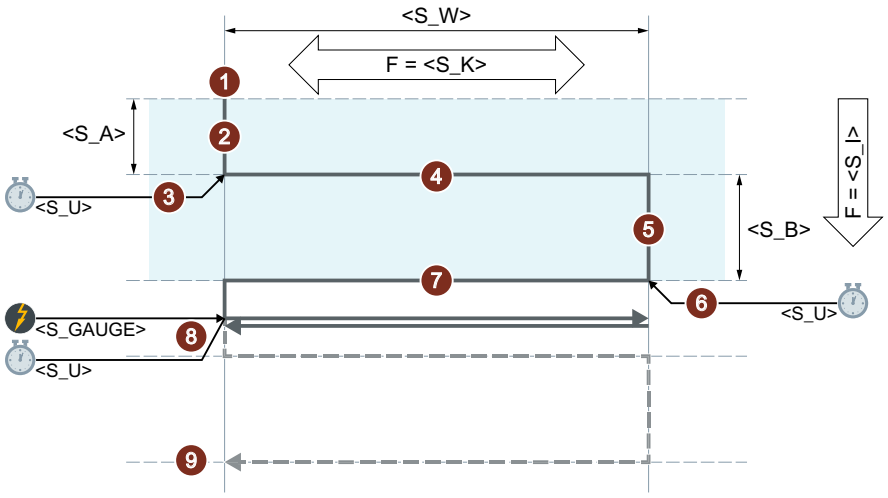
이 싸이클은 외부 취소 신호를 고려해 절입을 실행하는 데 사용됩니다. 시작과 종료 시 절입 깊이가 서로 다를 수 있습니다. 각 절입 간에는 접선 모션이 존재합니다. 취소 조건이 충족되면 깊이 절입이 취소됩니다. 깊이 절입이 취소되고 나면 항상 전체 스트로크가 수행됩니다.

순서

종료 시 절입 취소



시작 시 절입 취소



① 오실레이션 축의 현재 위치에서 싸이클 시작.

② 시작 <S_A>에서 절입 이송 속도 <S_I>로 절입 깊이까지 절입 축 이송.

③ 스파크아웃 시간을 이용한 스파크<S_U>

④ 이동 경로로 연삭 너비 <<S_W>>와 절입을 위한 이송 속도 <S_K>로 오실레이션 축 이송

⑤ 종료 <S_B>에서 절입을 위한 이송 속도 <S_I>로 절입 축을 절입 깊이까지 이송

⑥ 스파크아웃 시간을 이용한 스파크 <S_U>

⑦ 이동 경로로 연삭 너비 <S_W>와 절입을 위한 이송 속도 <S_K>로 시작점까지 오실레이션 축 이송

⑧ 신호 취소: 다음 시작점에 도달하면 가공이 정지됩니다.

⑨ 신호 취소 없음: 프로그래밍된 반복 횟수 <S_H>에 도달할 때까지 순서가 반복됩니다.

반복 단계를 나타냅니다.

참고

이 순서는 단일 블록으로 중단할 수 없습니다.

리소스

싸이클은 블록 전반에 걸친 동기 동작과 동기 동작 변수를 사용합니다. 동기 동작은 동기 동작 범위의 자유 영역으로부터 역동적으로 결정됩니다 (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). 동기 동작 변수로는 SYG_IS[1]이 사용됩니다.

예

예 1: 2 스트로크 오실레이션:

싸이클 파라미터

- 시작 시 절입 깊이: 0.02 mm
- 종료 시 절입 깊이: 0.01 mm
- 스트로크: 100mm
- 스파크아웃 시간: 1 s
- 절입 이송 속도: 1 mm/min
- 가로 방향 이송 속도: 1000mm/min
- 반복 횟수: 2
- 오실레이션 축 및 절입 축: 표준 기하 축

신호 취소: 신속한 입력 1 ($A_IN[1])

프로그램 코드

```
N10 T1 D1
N20 CYCLE4072("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

예 2: 2 스트로크 오실레이션:

싸이클 파라미터

- 시작 시 절입 깊이: 0.02 mm
- 종료 시 절입 깊이: 0.01 mm
- 스트로크: 100mm
- 스파크아웃 시간: 1 s
- 절입 이송 속도: 1 mm/min
- 가로 방향 이송 속도: 1000mm/min
- 반복 횟수: 2
- 오실레이션 축 및 절입 축: 표준 기하 축

신호 취소: 변수 $A_DBR[20] < 0.01

프로그램 코드

```
N10 T1 D1
```

프로그램 코드

```
N20 CYCLE4072("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

4.26.1.48 CYCLE4073 - 연속 절입을 통한 세로 방향 연삭

구문

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

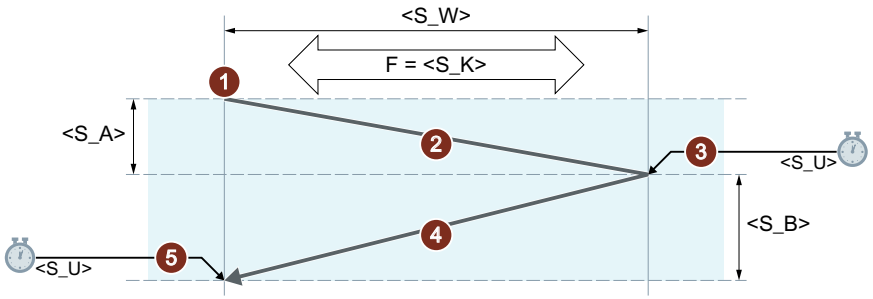
파라미터

번호	파라미터	데이터 유형	의미
1	<S_A>	REAL	시작 시 절입 깊이
2	<S_B>	REAL	종료 시 절입 깊이
3	<S_W>	REAL	연삭 너비
4	<S_U>	REAL	스파크아웃 시간
5	<S_K>	REAL	가로 방향 절입을 위한 이송 속도
6	<S_H>	INT	반복 횟수
7	<S_A1>	AXIS	절입 축(옵션) 또는 첫 번째 기하 축
8	<S_A2>	AXIS	오실레이션 축(옵션) 또는 두 번째 기하 축

기능

이 싸이클은 절입 반복에서 사용됩니다. 시작에서 종료까지의 절입과 종료에서 시작까지의 절입이 서로 다를 수 있습니다.

순서



① 절입 깊이 0으로 오실레이션 축의 현재 위치에서 싸이클 시작

② 시작 <S_A>의 절입 깊이까지 절입 깊이의 연속적인 증가와 함께 이동 경로로서 연삭 너비 <S_W>와 이송 절입을 위한 이송 속도 <S_K>로 오실레이션 축 이송

③ 스파크아웃 시간을 이용한 스파크 <S_U>

④ 종료 <S_B>의 절입 깊이까지 절입 깊이의 연속적인 증가와 함께 이동 경로로서 연삭 너비 <S_K>와 이송 절입을 위한 이송 속도 <S_W>로 시작점까지 오실레이션 축 이송

⑤ 스파크아웃 시간을 이용한 스파크 <S_U>

반복 단계를 나타냅니다.

프로그래밍된 반복 횟수 <S_H>에 도달할 때까지 순서가 반복됩니다.

참고

이 순서는 단일 블록으로 중단할 수 없습니다.

예

2 스트로크 오실레이션:

싸이클 파라미터

- 시작 시 절입 깊이: 0.02 mm
- 종료 시 절입 깊이: 0.01 mm
- 스트로크: 100mm
- 스파크아웃 시간: 1 s
- 가로 방향 이송 속도: 1000mm/min
- 반복 횟수: 2
- 오실레이션 축 및 절입 축: 표준 기하 축

프로그램 코드

```
N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30
```

4.26.1.49 CYCLE4074 - 연속 절입을 통한 세로 방향 연삭 및 취소 신호

구문

```
CYCLE4074(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)
```

파라미터

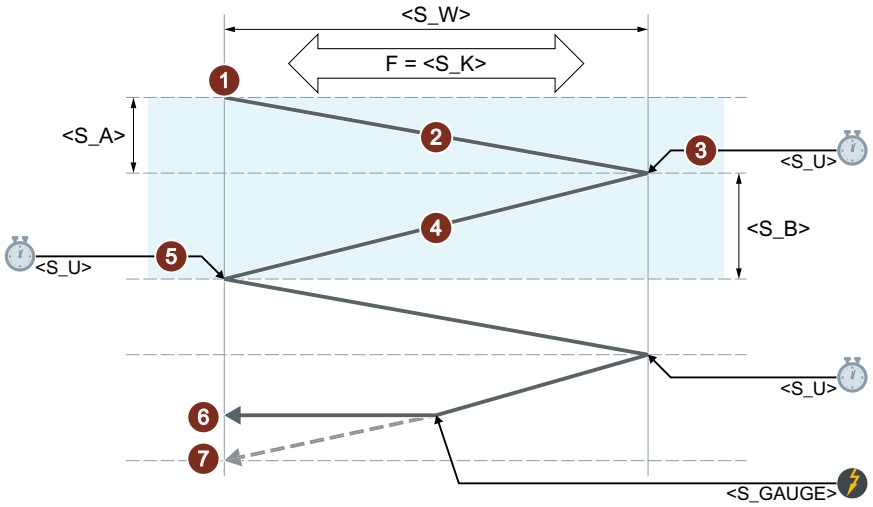
번호	파라미터	데이터 유형	의미
1	<S_GAUGE>	STRING	절입 취소 조건: 1. 고속 입력 번호 2. 논리 표현식
2	<S_A>	REAL	시작 시 절입 깊이
3	<S_B>	REAL	종료 시 절입 깊이
4	<S_W>	REAL	연삭 너비
5	<S_U>	REAL	스파크아웃 시간
6	<S_K>	REAL	가로 방향 절입을 위한 이송 속도
7	<S_H>	INT	반복 횟수
8	<S_A1>	AXIS	절입 축(옵션) 또는 첫 번째 기하 축
9	<S_A2>	AXIS	오실레이션 축(옵션) 또는 두 번째 기하 축

기능

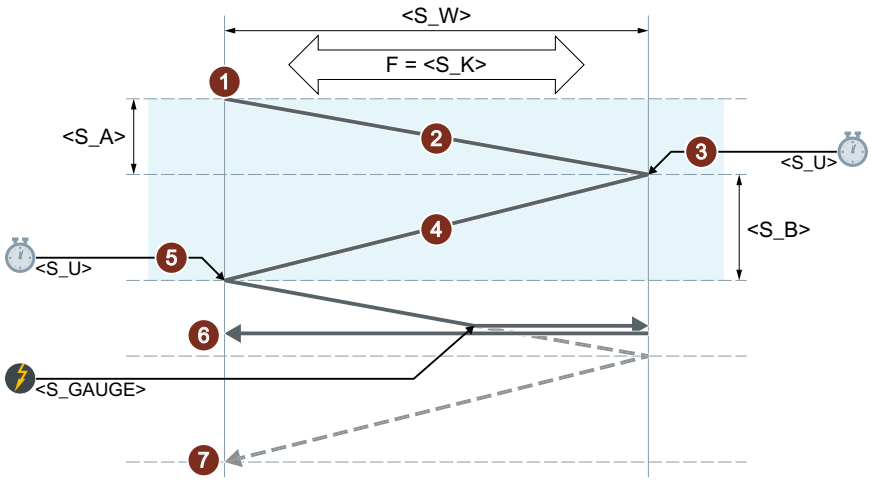
이 싸이클은 외부 취소 신호를 고려해 절입을 반복 실행하는 데 사용됩니다. 시작과 종료 시 절입 깊이가 서로 다를 수 있습니다. 취소 조건이 충족되면 깊이 절입이 취소됩니다. 깊이 절입이 취소되고 나면 항상 전체 스트로크가 수행됩니다.

순서

종료에서 시작까지의 절입에 대한 취소



시작에서 종료까지의 절입에 대한 취소



① 절입 깊이 0으로 오실레이션 축의 현재 위치에서 싸이클 시작

② 시작 <S_A>의 절입 깊이까지 절입 깊이의 연속적인 증가와 함께 이동 경로로 연삭 너비<<S_W>와 절입을 위한 이송 속도 <S_K>로 오실레이션 축 이송

③ 스파크아웃 시간을 이용한 스파크<S_U>

④ 종료 <S_B>의 절입 깊이까지 절입 깊이의 연속적인 증가와 함께 이동 경로로 연삭 너비 <S_W>와 절입을 위한 이송 속도 <S_K>로 시작점까지 오실레이션 축 이송

⑤ 스파크아웃 시간을 이용한 스파크<S_U>

⑥ 신호 취소: 깊이 절입이 취소됩니다. 다음 시작점에 도달하면 가공이 정지됩니다.

⑦ 신호 취소 없음: 프로그래밍된 반복 횟수 <S_H>에 도달할 때까지 순서가 반복됩니다.

반복 단계를 나타냅니다.

참고

이 순서는 단일 블록으로 중단할 수 없습니다.

리소스

싸이클은 블록 전반에 걸친 동기 동작과 동기 동작 변수를 사용합니다. 동기 동작은 동기 동작 범위의 자유 영역으로부터 역동적으로 결정됩니다 (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). 동기 동작 변수로는 SYG_IS[1]이 사용됩니다.

예

예 1: 2 스트로크 오실레이션:

싸이클 파라미터

- 시작 시 절입 깊이: 0.02 mm
- 종료 시 절입 깊이: 0.01 mm
- 스트로크: 100mm
- 스파크아웃 시간: 1 s
- 가로 방향 이송 속도: 1000mm/min
- 반복 횟수: 2
- 오실레이션 축 및 절입 축: 표준 기하 축

신호 취소: 신속한 입력 1 ($A_IN[1])

프로그램 코드

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

예 2: 2 스트로크 오실레이션:

싸이클 파라미터

- 시작 시 절입 깊이: 0.02 mm
- 종료 시 절입 깊이: 0.01 mm
- 스트로크: 100mm
- 스파크아웃 시간: 1 s
- 가로 방향 이송 속도: 1000mm/min
- 반복 횟수: 2
- 오실레이션 축 및 절입 축: 표준 기하 축

신호 취소: 변수 $A_DBR[20] < 0.01

프로그램 코드

```
N10 T1 D1
N20 CYCLE4074("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

4.26.1.50 CYCLE4075 - 반전 지점에서 절입을 통한 표면 연삭

구문

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>,
<S_A1>, <S_A2>)
```

파라미터

번호	파라미터	데이터 유형	의미
1	<S_I>	REAL	시작 시 절입 깊이
2	<S_J>	REAL	종료 시 절입 깊이
3	<S_K>	REAL	전체 절입 깊이
4	<S_A>	REAL	연삭 너비
5	<S_R>	REAL	절입을 위한 이송 속도
6	<S_F>	REAL	가로 방향 절입을 위한 이송 속도
7	<S_P>	REAL	스파크아웃 시간
8	<S_A1>	AXIS	절입 축 (옵션)
9	<S_A2>	AXIS	오실레이션 축 (옵션)

기능

이 싸이클은 절입 단계에서 전체 절입 깊이로 가공하는 데 사용됩니다. 시작과 종료 시의 절입 깊이가 서로 다를 수 있습니다. 각 절입 간에는 접선 모션이 존재합니다.

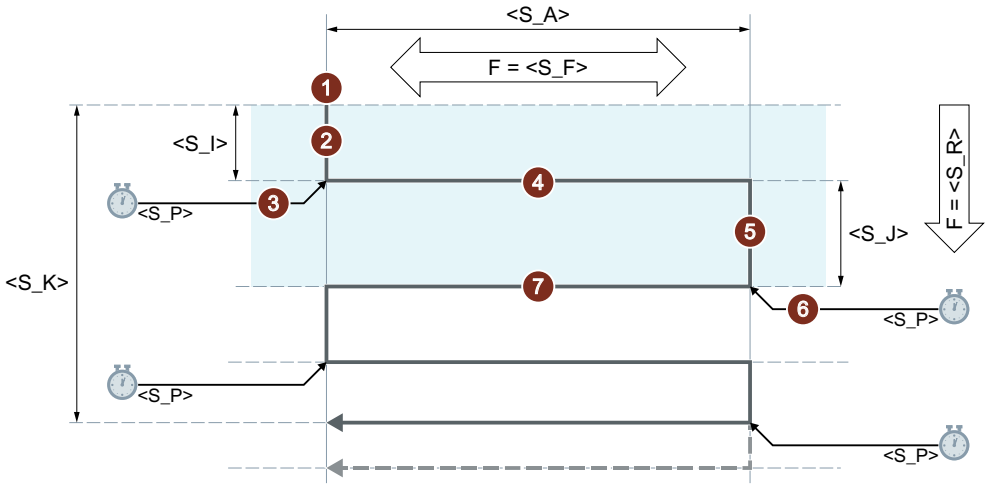
위치 데이터 P1 - P4는 음의 값일 수도 있고 양의 값일 수도 있습니다.

절입 축 및/또는 오실레이션 축의 지정은 선택사항입니다. 한 파라미터 또는 두 파라미터 모두가 지정되지 않을 경우 싸이클은 채널의 첫 번째 2개의 기하 축을 사용합니다.

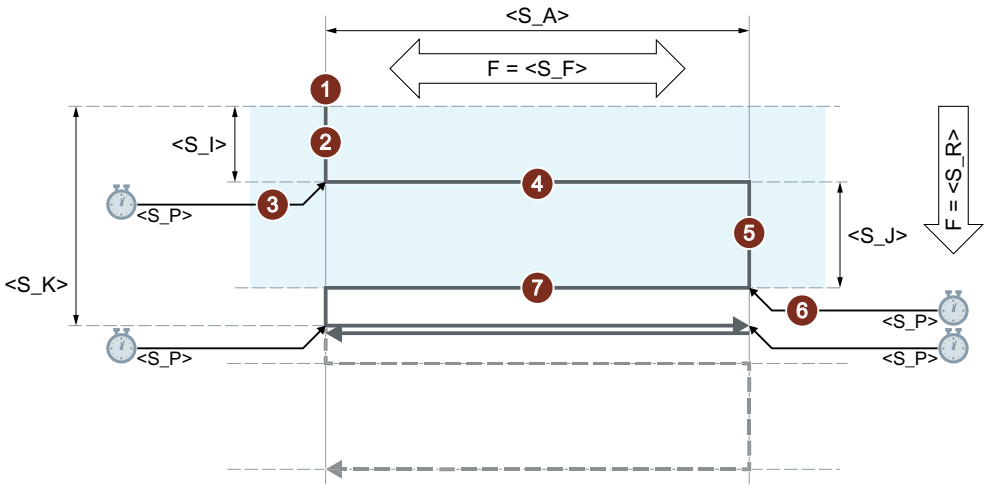
시작과 종료의 절입 깊이 합계가 0이거나 전체 절입 깊이가 0이면 스파크아웃 스트로크만 실행됩니다.

순서

두 번째 반전 지점에서 절입을 통해 도달된 총 절입 깊이



첫 번째 반전 지점에서 절입을 통해 도달된 총 절입 깊이



① 오실레이션 축의 현재 위치에서 싸이클 시작.

② 시작 <S_I>에서 절입 이송 속도 <S_R>로 절입 깊이까지 절입 축 이송.

③ 스파크아웃 시간을 이용한 스파크<S_P>

④ 이동 경로로 연삭 너비 <S_A>와 절입을 위한 이송 속도 <S_F>로 오실레이션 축 이송

⑤ 종료 <S_J>에서 절입을 위한 이송 속도 <S_R>로 절입 축을 절입 깊이까지 이송

⑥ 스파크아웃 시간을 이용한 스파크 <S_P>

⑦ 이동 경로로 연삭 너비 <S_A>와 절입을 위한 이송 속도 <S_F>로 시작점까지 오실레이션 축 이송

반복 단계를 나타냅니다.

이 순서는 전체 절입 깊이 <S_K>에 도달할 때까지 반복됩니다. 그 이후 마지막 스트로크가 고르지 않게 분배됩니다.

참고

이 순서는 단일 블록으로 중단할 수 없습니다.

예

다음과 같은 오실레이션:

- 시작 시 절입 깊이 0.02 mm
- 종료 시 절입 깊이 0.01 mm
- 전체 절입 깊이 1mm

- 100mm 스트로크
- 절입 속도 1mm/min
- 가로 이송 속도 1,000mm/min
- 스파크아웃 시간 1초
- 표준 기하 축

프로그램 코드

```
N10 T1 D1
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30
```

4.26.1.51 CYCLE4077 - 반전 지점에서의 절입을 통한 표면 연삭과 취소 신호

구문

```
CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>,
<S_P>, <S_A1>, <S_A2>)
```

파라미터

번호	파라미터	데이터 유형	의미
1	<S_GAUGE>	STRING	절입에 대한 취소 조건: • 고속 입력 번호 • 논리 표현식
2	<S_I>	REAL	시작 시 절입 깊이
3	<S_J>	REAL	종료 시 절입 깊이
4	<S_K>	REAL	전체 절입 깊이
5	<S_A>	REAL	연삭 너비
6	<S_R>	REAL	절입을 위한 이송 속도
7	<S_F>	REAL	가로 방향 절입을 위한 이송 속도
8	<S_P>	REAL	스파크아웃 시간
9	<S_A1>	AXIS	절입 축 (옵션)
10	<S_A2>	AXIS	오실레이션 축 (옵션)

기능

이 싸이클은 절입 단계에서 전체 절입 깊이로 가공하는 데 사용됩니다. 시작과 종료 시의 절입 깊이가 서로 다를 수 있습니다. 각 절입 간에는 접선 모션이 존재합니다. 고속 입력의 취소 신호가 1이거나 취소 조건을 충족하면 깊이 절입이 취소됩니다. 취소 후 항상 전체 스트로크가 실행됩니다.

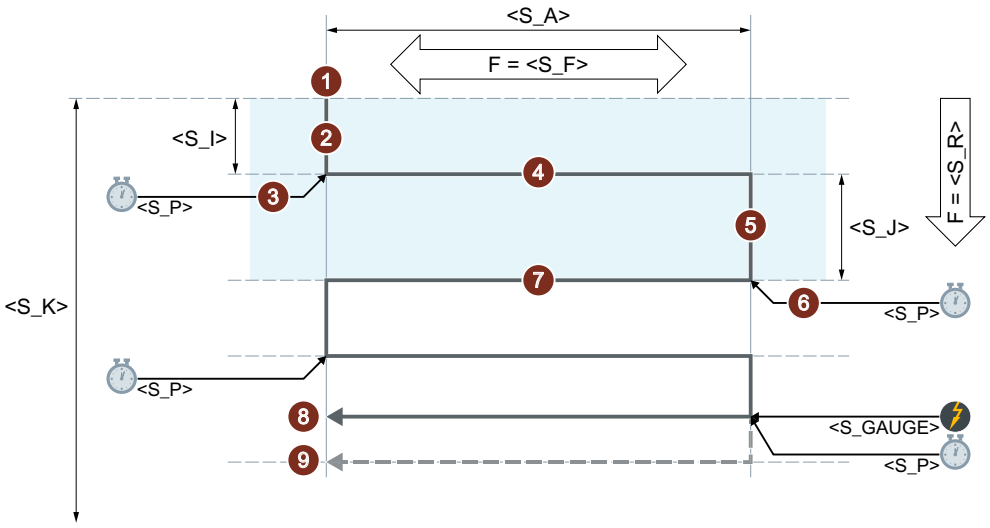
위치 데이터 P2 - P5는 음의 값일 수도 있고 양의 값일 수도 있습니다.

절입 축 및/또는 오실레이션 축의 지정은 선택사항입니다. 한 파라미터 또는 두 파라미터 모두가 지정되지 않을 경우 싸이클은 채널의 첫 번째 2개의 기하 축을 사용합니다.

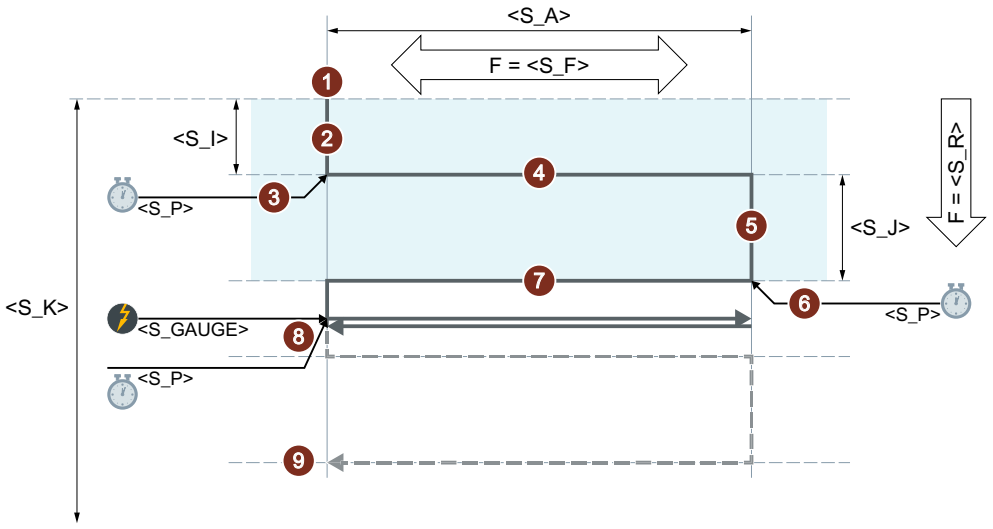
시작과 종료의 절입 깊이 합계가 0이거나 전체 절입 깊이가 0이면 스파크아웃 스트로크만 실행됩니다.

순서

종료 시 절입 취소



시작 시 절입 취소



① 오실레이션 축의 현재 위치에서 싸이클 시작.

② 시작 <S_I>에서 절입을 위한 이송 속도 <S_R>로 절입 축을 절입 깊이까지 이송

③ 스파크아웃 시간을 이용한 스파크 <S_P>

④ 이동 경로로 연삭 너비 <S_A>와 절입을 위한 이송 속도 <S_F>로 오실레이션 축 이송

⑤ 종료 <S_J>에서 절입을 위한 이송 속도 <S_R>로 절입 축을 절입 깊이까지 이송

⑥ 스파크아웃 시간을 이용한 스파크 <S_P>

⑦ 이동 경로로 연삭 너비 <S_A>와 절입을 위한 이송 속도 <S_F>로 시작점까지 오실레이션 축 이송

⑧ 신호 취소: 다음 시작점에 도달하면 가공이 정지됩니다.

⑨ 신호 취소 없음: 이 순서는 전체 절입 깊이 <S_K>에 도달할 때까지 반복됩니다. 그 이후 마지막 스트로크가 고르지 않게 분배됩니다.

반복 단계를 나타냅니다.

참고

이 순서는 단일 블록으로 중단할 수 없습니다.

리소스

싸이클은 블록 전반에 걸친 동기 동작과 동기 동작 변수를 사용합니다. 동기 동작은 동기 동작 범위의 자유 영역으로부터 역동적으로 결정됩니다 (CUS.DIR - 1 ..., CMA.DIR - 1000 ..., CST.DIR – 1199 ...). 동기 동작 변수로는 SYG_IS[1]이 사용됩니다.

예

예 1

다음과 같은 오실레이션:

- 시작 시 절입 깊이 0.02 mm
- 종료 시 절입 깊이 0.01 mm
- 전체 절입 깊이 1mm
- 100mm 스트로크
- 절입 속도 1mm/min
- 가로 이송 속도 1,000mm/min
- 스파크아웃 시간 1초
- 표준 기하 축

신호 취소: 신속한 입력 1 ($A_IN[1])

프로그램 코드

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

예 2

다음과 같은 오실레이션:

- 시작 시 절입 깊이 0.02 mm
- 종료 시 절입 깊이 0.01 mm
- 전체 절입 깊이 1mm
- 100mm 스트로크
- 절입 속도 1mm/min
- 가로 이송 속도 1,000mm/min
- 스파크아웃 시간 1초
- 표준 기하 축

신호 취소: 듀얼 포트 RAM 변수 20, 0.01 미만($A_DBR[20] < 0.01)

프로그램 코드

```
N10 T1 D1
N20 CYCLE4077("($A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
```

프로그램 코드
N30 M30

4.26.1.52 CYCLE4078 - 연속 절입을 통한 표면 연삭

구문

```
CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>,
<S_A2>)
```

파라미터

번호	파라미터	데이터 유형	의미
1	<S_I>	REAL	시작에서 종료까지 절입 깊이
2	<S_J>	REAL	종료에서 시작까지 절입 깊이
3	<S_K>	REAL	전체 절입 깊이
4	<S_A>	REAL	연삭 너비
5	<S_F>	REAL	이송 속도
6	<S_P>	REAL	스파크아웃 시간
7	<S_A1>	AXIS	절입 축 (옵션)
8	<S_A2>	AXIS	오실레이션 축 (옵션)

기능

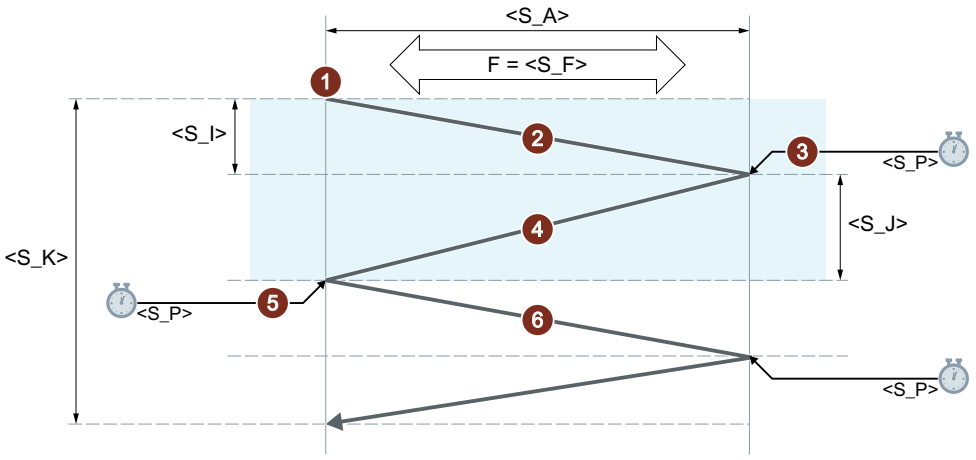
이 싸이클은 연속 절입을 이용해 전체 절입 깊이로 가공하는 데 사용됩니다. 시작에서 종료까지 그리고 종료에서 시작까지의 절입 깊이가 서로 다를 수 있습니다.

위치 데이터 P1 - P4는 음의 값일 수도 있고 양의 값일 수도 있습니다.

절입 축 및/또는 오실레이션 축의 지정은 선택사항입니다. 한 파라미터 또는 두 파라미터 모두가 지정되지 않을 경우 싸이클은 채널의 첫 번째 2개의 기하 축을 사용합니다.

절입 깊이 P1 및 P2의 합계가 0이거나 전체 절입 깊이가 0이면 스파크아웃 스트로크만 실행됩니다.

순서



① 절입 깊이 0으로 오실레이션 축의 현재 위치에서 싸이클 시작

② 시작 <S_I>의 절입 깊이까지 절입 깊이의 연속적인 증가와 함께 이동 경로로 연삭 너비 <S_A>와 이송 속도 <S_F>로 오실레이션 축 이송

③ 스파크아웃 시간을 이용한 스파크 <S_P>

④ 종료 <S_J>의 절입 깊이까지 절입 깊이의 연속적인 증가와 함께 이동 경로로서 연삭 너비 <S_A>와 이송 절입을 위한 이송 속도 <S_F>로 시작점까지 오실레이션 축 이송

⑤ 스파크아웃 시간을 이용한 스파크 <S_P>

⑥ 이동 경로로 연삭 너비 <S_A>와 이송 속도 <S_F>로 시작점까지 오실레이션 축 이송

반복 단계를 나타냅니다.

이 순서는 전체 절입 깊이 <S_K>에 도달할 때까지 반복됩니다. 그 이후 마지막 스트로크가 고르지 않게 분배됩니다.

참고

이 순서는 단일 블록으로 중단할 수 없습니다.

예

다음과 같은 오실레이션:

- 시작 시 절입 깊이 20 mm
- 종료 시 절입 깊이 10 mm
- 전체 절입 깊이 100 mm

- 100mm 스트로크
- 이송 속도 1,000 mm/min
- 스파크아웃 시간 1초
- 표준 기하 축

프로그램 코드

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

4.26.1.53 CYCLE4079 - 간헐적 절입을 통한 표면 연삭

구문

```
CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>,
<S_A1>, <S_A2>)
```

파라미터

번호	파라미터	데이터 유형	의미
1	<S_I>	REAL	시작 시 절입 깊이
2	<S_J>	REAL	종료 시 절입 깊이
3	<S_K>	REAL	전체 절입 깊이
4	<S_A>	REAL	연삭 너비
5	<S_R>	REAL	절입을 위한 이송 속도
6	<S_F>	REAL	가로 방향 절입을 위한 이송 속도
7	<S_P>	REAL	스파크아웃 시간
8	<S_A1>	AXIS	절입 축 (옵션)
9	<S_A2>	AXIS	오실레이션 축 (옵션)

기능

이 싸이클은 절입 단계에서 전체 절입 깊이로 가공하는 데 사용됩니다. 시작과 종료 시의 절입 깊이가 서로 다를 수 있습니다. 각 절입 간에는 접선 모션이 존재합니다.

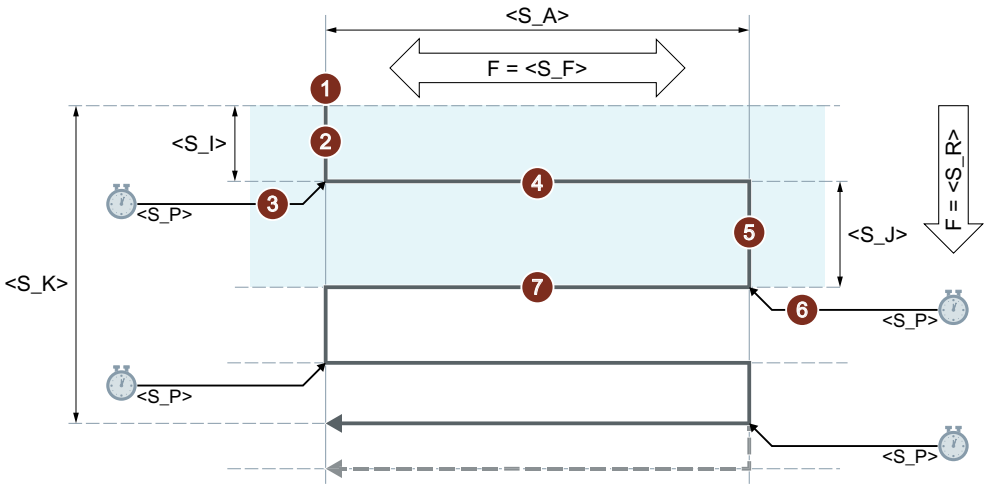
위치 데이터 P1 - P4는 음의 값일 수도 있고 양의 값일 수도 있습니다.

절입 축 및/또는 오실레이션 축의 지정은 선택사항입니다. 한 파라미터 또는 두 파라미터 모두가 지정되지 않을 경우 싸이클은 채널의 첫 번째 2개의 기하 축을 사용합니다.

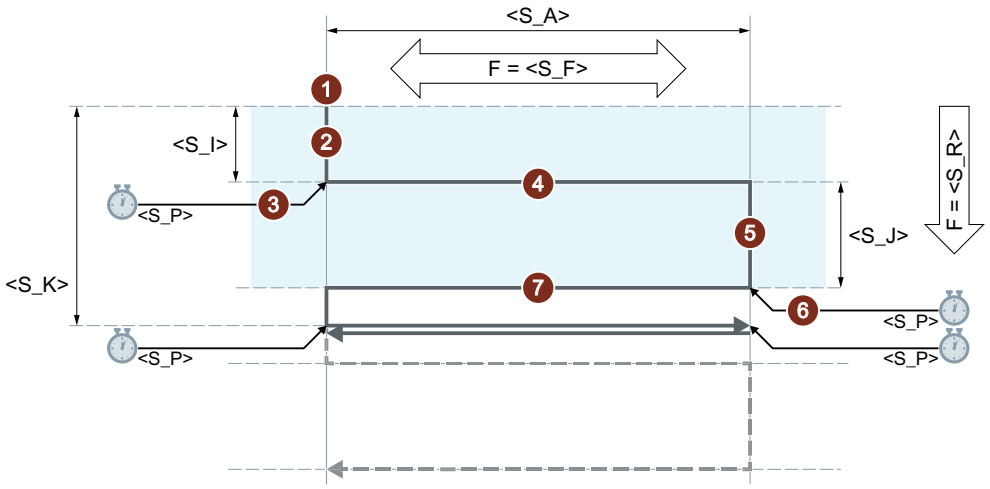
시작과 종료의 절입 깊이 합계가 0이거나 전체 절입 깊이가 0이면 스파크아웃 스트로크만 실행됩니다.

순서

두 번째 반전 지점에서 절입을 통해 도달된 총 절입 깊이



첫 번째 반전 지점에서 절입을 통해 도달된 총 절입 깊이



① 오실레이션 축의 현재 위치에서 싸이클 시작.

② 시작 <S_I>에서 절입을 위한 이송 속도 <S_R>로 절입 축을 절입 깊이까지 이송

③ 스파크아웃 시간을 이용한 스파크 <S_P>

④ 이동 경로로 연삭 너비 <S_A>와 절입을 위한 이송 속도 <S_F>로 오실레이션 축 이송

⑤ 종료 <S_J>에서 절입을 위한 이송 속도 <S_R>로 절입 축을 절입 깊이까지 이송

⑥ 스파크아웃 시간을 이용한 스파크 <S_P>

⑦ 이동 경로로 연삭 너비 <S_A>와 절입을 위한 이송 속도 <S_F>로 시작점까지 오실레이션 축 이송

반복 단계를 나타냅니다.

이 순서는 전체 절입 깊이 <S_K>에 도달할 때까지 반복됩니다. 그 이후 마지막 스트로크가 고르지 않게 분배됩니다.

참고

이 순서는 단일 블록으로 중단할 수 없습니다.

예

다음과 같은 오실레이션:

- 시작 시 절입 깊이 0.02 mm
- 종료 시 절입 깊이 0.01 mm
- 전체 절입 깊이 1mm

- 100mm 스트로크
- 절입 속도 1mm/min
- 가로 이송 속도 1,000mm/min
- 스파크아웃 시간 1초
- 표준 기하 축

프로그램 코드

```
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30
```

4.26.1.54 GROUP_BEGIN - 프로그램 블록 시작

구문

```
GROUP_BEGIN(<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미	
1		<_LEVEL>	INT	레벨	
				0 =	메인 레벨
				1 =	첫 번째 서브레벨
2		<_NAME>	STRING[128]	블록 이름	
3		<_SP>	INT	스핀들	
				0 =	스핀들 없음
				1 =	메인 스핀들
				2 =	서브 스핀들

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미		
4		<_MODE>	INT	모드		
				비트 0	=1	GROUP_ADDEND이 존재합니다.
				비트 1	=1	ShopTurn: 자동 후퇴(공구 변경점까지의 이송)
				비트 12	예비	
				비트 13	예비	
5		<S_ICON>	STRING[32]	아이콘 이름(조작자 인터페이스만 해당)		

4.26.1.55 GROUP_END - 프로그램 블록 종료

구문

```
GROUP_END(<_LEVEL>, <_SP>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미	
1		<_LEVEL>	INT	레벨	
				0 =	메인 레벨
				1 =	첫 번째 서브레벨
2		<_SP>	INT	스핀들	
				0 =	스핀들 없음
				1 =	메인 스핀들
				2 =	서브 스핀들

4.26.1.56 GROUP_ADDEND - 시험 절삭 추가 종료

구문

```
GROUP_ADDEND(<_LEVEL>, <_SP>)
```

파라미터

번호	파라미터 마스크	내부 파라미터	데이터 유형	의미	
1		<_LEVEL>	INT	레벨	
				0 =	메인 레벨
				1 =	첫 번째 서브레벨
2		<_SP>	INT	스핀들	
				0 =	스핀들 없음
				1 =	메인 스핀들
				2 =	서브 스핀들

4.26.1.57 보충 조건

싸이클 화면 형식에서 테크놀로지 스케일링

테크놀로지 스케일링이 활성화되면, 가장 중요한 싸이클 파라미터만이 표시되는 다양한 싸이클 화면 형식들에 대해 간소화된 입력을 선택할 수 있습니다.

예를 들어, 다음 싸이클 화면 형식들에 대해 간소화된 입력을 선택할 수 있습니다.

테크놀로지	싸이클 화면 형식
드릴링	심공 드릴링
	태핑
밀링	사각 포켓
	형상 밀링: 포켓
선삭	나사 선삭: 길이 방향
	형상 선삭: 선삭 가공
	형상 선삭: 홈 가공
	형상 선삭: 홈 선삭

해당 싸이클 화면 형식의 사용자 인터페이스에서, "입력: **간단**"과 "입력: **전체**" 옵션을 사용할 수 있습니다.

표시되지 않는 싸이클 파라미터

간소화된 입력에 표시되지 않는 싸이클 파라미터들은 사전에 지정되어 고정되어 있고 기술적으로 유용하며 변화하지 않는 값들입니다. 또는, 채널별 싸이클 셋팅 데이터를 통해 싸이

클 파라미터에 값들이 지정되기도 합니다. "스타트업" > "채널별 싸이클 셋팅 데이터" 아래의 단락을 참조하십시오.

"입력: 전체" > "입력: 간단"으로 전환

싸이클 화면 형식을 "입력 전체 설정으로 채운 다음 "입력 간단"으로 전환하면, 싸이클 호출을 생성할 때 더 이상 표시되지 않는 파라미터들에 디폴트 또는 셋팅 데이터 값들이 사용됩니다.

스타트업

채널 관련 환경설정 머신 데이터

싸이클 화면 형식의 테크놀로지 스케일링은 다음과 같은 머신 데이터를 통해 활성화할 수 있습니다:

MD52210 $MCS_FUNCTION_MASK_DISP, 비트 9 = 1 ("단순 입력" 디스플레이 선택)

채널 싸이클 셋팅 데이터

싸이클 화면 형식에서 간소화된 입력이 활성화된 경우, 다음 셋팅 데이터를 통해 특정 싸이클 파라미터의 값을 지정할 수 있습니다.

번호	식별자	의미
SD55300	$SCS_EASY_SAFETY_CLEARANCE	안전 거리
SD55301	$SCS_EASY_DWELL_TIME	드웰 시간
SD55305	$SCS_EASY_DRILL_DEEP_FD1	심공 드릴링: 백분율: 1차 이송 속도
SD55306	$SCS_EASY_DRILL_DEEP_DF	심공 드릴링: 백분율: 인피드
SD55307	$SCS_EASY_DRILL_DEEP_V1	심공 드릴링: 최소 절입 깊이
SD55308	$SCS_EASY_DRILL_DEEP_V2	심공 드릴링: 후퇴 거리
SD55309	$SCS_EASY_THREAD_RETURN_DIST	나사 선삭: 복귀 거리

4.26.2 측정 싸이클

측정 싸이클은 특정 측정 작업을 해결하기 위해 Siemens에서 제공하는 특수 서브프로그램입니다. 일반적인 싸이클의 경우, 파라미터 설정을 통해 개별 문제들에 측정 싸이클을 적용할 수 있습니다.

측정 싸이클은 다음과 같은 영역과 테크놀로지에 사용할 수 있습니다:

- 공구 측정, 선삭/밀링
- 공작물 측정, 선삭/밀링

추가 정보

측정 싸이클 프로그래밍 매뉴얼

4.26.2.1 측정 싸이클 파라미터 개요

CYCLE973 측정 싸이클 파라미터

```
PROC CYCLE973(INT S_MVAR,INT S_PRNUM,INT S_CALNUM,REAL S_SETV,INT S_MA,INT S_MD,REAL
S_FA,REAL S_TSA,REAL S_VMS,INT S_NMSP,INT S_MCBIT,INT _DMODE,INT _AMODE)
```

도표 4-8 CYCLE973 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식(기본값 = 0012103)	
			값:	UNITS: 표면, 에지 또는 홈에서 계측 0 = 지령치를 알고 있는 표면/에지에서 길이 (WCS) 1 = 지령치를 알고 있는 표면에서 반경 (WCS) 2 = 홈에서 길이 (WCS), S_CALNUM 참조 3 = 홈에서 반경 (WCS), S_CALNUM 참조
				TENS: 예비 0 = 0
				HUNDREDS: 예비 0 = 0
				THOUSANDS: 계측을 위한 측정 축 및 측정 방향 선택 [2)] 0 = 지정 안함 (홈 바닥에서 표면을 계측하는 경우, 측정 축 및 측정 방향 선택 안함) [4)] 1 = 측정 축 및 측정 방향 선택 지정, S_MA, S_MD 참조 (측정 축 1개에 측정 방향 1개 선택) 2 = 측정 축 선택 지정 S_MA (측정 축 1개에 측정 방향 2개 선택)
				TEN THOUSANDS: 위치 편향 (프로브 스큐) 지정 [2), 3)] 0 = 위치 편향 지정 1 = 위치 편향 지정 안함
				HUNDRED THOUSANDS: 예비 0 = 0
				ONE MILLION:공구 길이 조정 [7)] 0 = 공구 길이 조정 안 함(트리거 지점만 결정) 1 = 공구 길이 조정
2	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호(프로브 번호 아님) (기본값 = 1)	
3		S_CALNUM	홈에서 계측을 위한 계측 홈 번호(기본값 = 1) [5)]	
4		S_SETV	표면에서 계측을 위한 지령치	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
5	X0	S_MA	측정 축(축 번호) [6] (기본값 = 1)	
			값:	1 = 평면의 첫 번째 축 (G18의 경우 Z) 2 = 평면의 두 번째 축 (G18의 경우 X) 3 = 평면의 세 번째 축 (G18의 경우 Y) [6]
6	+-	S_MD	측정 방향(기본값 = 1)	
			값:	0 = 양의 측정 방향 1 = 음의 측정 방향
7	DFA	S_FA	측정 경로	
8	TSA	S_TSA	안전 영역	
9	VMS	S_VMS	계측을 위한 가변 측정 속도 [2]	
10	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (기본값 = 1)	
11		S_MCBIT	예비	
12		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
13		_AMODE	대체 모드	

[1] 모든 기본값 = 0 또는 기본값 = x로 표시

[2] 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

[3] 두 축 방향에서 계측하는 경우만 해당.

[4] 프로브의 절삭날 위치 (SL) 에서 측정 축 및 측정 방향을 자동 결정하는 경우만 해당. SL = 8 → -X , SL = 7 → -Z

[5] 계측 홈의 개수(n)는 다음 일반 설정 데이터를 기준으로 합니다(MCS의 모든 위치).
절삭날 위치 SL=7인 경우:
SD54615 $SNS_MEA_CAL_EDGE_BASE_AX1[n] 평면의 첫 번째 축(G18의 경우 Z)에 있는 홈 바닥의 위치
SD54621 $SNS_MEA_CAL_EDGE_PLUS_DIR_AX2[n] 평면의 두 번째 축(G18의 경우 X)의 양의 방향에 있는 홈 측면의 위치
SD54622 $SNS_MEA_CAL_EDGE_MINUS_DIR_AX2[n] 평면의 두 번째 축의 음의 방향에 있는 홈 측면의 위치
절삭날 위치 SL=8인 경우:
SD54619 $SNS_MEA_CAL_EDGE_BASE_AX2[n] 평면의 두 번째 축에 있는 홈 바닥의 위치
SD54620 $SNS_MEA_CAL_EDGE_UPPER_AX2[n] 평면의 두 번째 축에 있는 홈 상부 에지의 위치(프로브 사전 포지셔닝에만 사용)

SD54617 $SNS_MEA_CAL_EDGE_PLUS_DIR_AX1[n] 평면의 첫 번째 축의 양의 방향에 있는 홈 측면의 위치
SD54618 $SNS_MEA_CAL_EDGE_MINUS_DIR_AX1[n] 평면의 첫 번째 축의 음의 방향에 있는 홈 측면의 위치
참고:
+/- 방향에 있는 홈 벽의 위치 값은 근사치로 지정할 수 있습니다.
홈 벽의 위치 값 차이에서 홈 너비를 정확하게 지정할 수 있어야 합니다(정밀 다이얼 게이지).
홈에서 계측할 때 계측된 축에서 프로브의 공구 길이를 0으로 가정합니다.
홈 베이스의 위치 값은 기계에서 정확하게 지정해야 합니다(도면 치수 아님).

6) 평면의 실제 세 번째 축 (G18의 경우 Y) 에서 표면 및 선삭 공구 계측을 위한 측정 축 `S_MA=3` .

7) 홈에서 길이를 계측할 때 또는 표면의 길이에 대해 공구 길이를 지정합니다.
선반의 공작물 프로브는 2개의 길이 (X Z) 를 사용하여 정의할 수 있습니다.
선삭 프로브, 유형 580
절삭날 위치 7: 길이 계측의 경우에는 선택적으로 Z 길이가 수정됩니다.
선삭 프로브, 유형 580
절삭날 위치 8: 길이 계측의 경우에는 선택적으로 X 길이가 수정됩니다.
측정 버전, 홈의 반경 또는 표면의 반경에 대해서는 공구 길이가 지정되지 않습니다.
해당 트리거 지점만 저장됩니다.

CYCLE974 측정 싸이클 파라미터

```
PROC CYCLE974(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,INT S_MA,REAL
S_FA,REAL S_TSA,REAL S_STA1,INT S_NMSP,STRING[32] S_TNAME,INT S_DLNUM,REAL S_TZL,REAL
S_TDIF,REAL S_TUL,REAL S_TLL,REAL S_TMV,INT S_K,INT S_EVNUM,INT S_MCBIT,INT _DMODE,INT
_AMODE,INT _DP)
```

도표 4-9 CYCLE974 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식	
			값:	UNITS: 0 = 전면 단면 측정 1 = 내부 측정 2 = 외부 측정
				TENS: 예비
				HUNDREDS: 수정 대상 0 = 측정만 수행 (WO 또는 공구 옵셋 수정 안함) 1 = 측정, WO 결정 및 수정 (S_KNUM 참조) [3)] 2 = 측정 및 공구 옵셋 (S_KNUM1 참조)
				THOUSANDS: 예비
				TEN THOUSANDS: 메인 스핀들(선삭 스핀들)을 반전시켜/반전 없이 측정 0 = 반전 없이 측정 1 = 반전시켜 측정

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	워크 옵셋(WO) 또는 기본 WO 또는 기본 원점의 수정 [2)]	
			값:	UNITS:
				TENS: 0 = 수정 안함 워크 옵셋 번호 1~99 기본 옵셋 번호 1~16
				HUNDREDS: 예비
				THOUSANDS: WO, 기본 WO 또는 기본 원점 수정 0 = 조정 가능한 WO 수정 1 = 채널별 기본 WO 수정 2 = 기본 원점 수정 3 = 전역 기본 WO 수정 9 = 활성 WO 수정 또는 G500의 경우 마지막으로 활성화된 채널별 기본 WO 수정
				TEN THOUSANDS: WO, 기본 WO 또는 기본 원점 일반/미세 수정 0 = 미세 수정 [6)] 1 = 일반 수정

번호	화면 양식 파라미터	싸이클 파라미터	의미	
3	선택	S_KNUM1	공구 옵셋 수정 [2), 4)]	
			값:	UNITS:
				TENS:
				HUNDREDS: 0 = 수정 안함 공구 옵셋을 위한 D 번호 1~999, 추가 옵셋 및 셋업 옵셋의 경우 S_DLNUM 참조
				THOUSANDS: 0 또는 고유 D 번호
				TEN THOUSANDS: 0 또는 고유 D 번호 1~32000, MD의 고유 D 번호가 설정된 경우
				HUNDRED THOUSANDS: 공구 옵셋 [2)] 0 = 지정 안함 (공구 형상 옵셋) 1 = 길이 L1 옵셋 2 = 길이 L2 옵셋 3 = 길이 L3 옵셋 4 = 반경 옵셋
				ONE MILLION: 공구 옵셋 [2)] 0 = 지정 안함 (공구 길이 마모 옵셋) 1 = 공구 옵셋, 추가 옵셋 (AO) [5)] 공구 옵셋 값을 기존 AO에 추가 2 = 공구 옵셋, 셋업 옵셋 (SO) [5)] SO (신규) = SO (기존) + AO (기존) 옵셋 값, AO (신규) = 0 3 = 공구 옵셋, 셋업 옵셋 (SO) [5)] 공구 옵셋 값을 기존 SO에 추가 4 = 공구 옵셋, 기하
				TEN MILLION: 공구 옵셋 [2)] 0 = 지정 안함 (공구 형상 일반 옵셋, 반전 없음) 1 = 반전 옵셋
				HUNDRED MILLIONS: 공구 옵셋 0 = 대체 공구가 없는 공구 옵셋 1 = 대체 공구의 공구 옵셋(_DP)
4	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호(프로브 번호 아님) (기본값 = 1)	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
5	X0	S_SETV	지령치	
6	X	S_MA	측정 축(축 번호)(기본값 = 1)	
			값:	1 = 평면의 첫 번째 축 (G18의 경우 Z) 2 = 평면의 두 번째 축 (G18의 경우 X) 3 = 평면의 세 번째 축 (G18의 경우 Y)[5]
7	DFA	S_FA	측정 경로	
8	TSA	S_TSA	안전 영역	
9	α	S_STA1	반전시켜 측정하는 경우 시작 각도	
10	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (기본값 = 1)	
11	T	S_TNAME	공구 이름 [2]	
12	DL	S_DLNUM	셋업/추가 옵셋 DL 번호 [5]	
13	ST	_DP	수정할 대체 공구 번호 (듀플로 번호)	
14	TZL	S_TZL	워크 옵셋 [2], [4]	
15	DIF	S_TDIF	치수 차이 검사 [2], [4]	
16	TUL	S_TUL	공차 상한값 (지령치로 증분) [4]	
17	TLL	S_TLL	공차 하한값 (지령치로 증분) [4]	
18	TMV	S_TMV	평균 계산을 위한 옵셋 범위 [2]	
19	FW	S_K	평균 계산을 위한 가중치 [2]	
20	EVN	S_EVNUM	실험값 평균값 메모리 번호 [2], [7]	
21		S_MCBIT	예비	
22		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
23		_AMODE	대체 모드	
			값:	UNITS: 치수 공차 사용 여부 0 = 아니오 1 = 예

[1] 모든 기본값 = 0 또는 기본값 = x로 표시

[2] 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

3) 반전 없이 측정의 경우 WO 수정만 가능

4) 공구 옵셋의 경우 채널별 MD 20360 TOOL_PARAMETER_DEF_MASK 의 비트 0 및 비트 1 적용

5) "셋업/추가 옵셋" 기능이 일반 MD 18108 $MN_MM_NUM_SUMCORR 에 설정된 경우만 해당. 또한 일반 MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK 의 비트 8이 1로 설정되어 있어야 합니다.

6) MD에 WO "미세" 수정이 설정되지 않은 경우 WO를 "일반" 수정

7) 공구 옵셋에만 실험 값 평균 계산 가능
실험 평균 값 메모리의 값 범위:
실험 값 메모리 번호 1~20, 채널별 SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1] 참조
평균 값 메모리 번호 10000~200000 (n), 채널별 SD55625 $SCS_MEA_AVERAGE_VALUE[n-1] 참조

CYCLE994 측정 싸이클 파라미터

```
PROC CYCLE994(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,INT S_MA,REAL
S_SZA,REAL S_SZO,REAL S_FA,REAL S_TSA,INT S_NMSP,STRING[32] S_TNAME,INT S_DLNUM,REAL
S_TZL,REAL S_TDIF,REAL S_TUL,REAL S_TLL,REAL S_TMV,INT S_K,INT S_EVNUM,INT S_MCBIT,INT
_DMODE,INT _AMODE,INT _DP)
```

도표 4-10 CYCLE994 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식	
			값:	UNITS: 내부 또는 외부 측정(기본값 = 1) 1 = 내부 측정 2 = 외부 측정
				TENS: 예비
				HUNDREDS: 수정 대상 0 = 측정만 수행 (WO 또는 공구 옵셋 수정 안함) 1 = 측정, WO 결정 및 수정 (S_KNUM 참조) [3)] 2 = 측정 및 공구 옵셋 (S_KNUM1 참조)
				THOUSANDS: 우회 영역 0 = 우회 영역 없음 1 = 우회 축, 평면의 첫 번째 축(G18의 경우 Z). 측정 축은 S_MA 참조. 2 = 우회 축, 평면의 두 번째 축(G18의 경우 X). 측정 축은 S_MA 참조. 3 = 우회 축, 평면의 세 번째 축(G18의 경우 Y). 측정 축은 S_MA 참조. [8)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	워크 옵셋(WO) 또는 기본 WO 또는 기본 원점의 수정 [2]	
			값:	UNITS:
				TENS: 0 = 수정 안함 워크 옵셋 번호 1~99 기본 옵셋 번호 1~16
				HUNDREDS: 예비
				THOUSANDS: WO, 기본 WO 또는 기본 원점 수정 0 = 조정 가능한 WO 수정 1 = 채널별 기본 WO 수정 2 = 기본 원점 수정 3 = 전역 기본 WO 수정 9 = 활성 WO 수정 또는 G500의 경우 마지막으로 활성화된 채널별 기본 WO 수정
				TEN THOUSANDS: WO, 기본 WO 또는 기본 원점 일반/미세 수정 0 = 미세 수정 [6] 1 = 일반 수정

번호	화면 양식 파라미터	싸이클 파라미터	의미	
3	선택	S_KNUM1	공구 옵셋 수정 [2), 4)]	
			값:	UNITS:
				TENS:
				HUNDREDS: 0 = 수정 안함 공구 옵셋을 위한 D 번호 1~999, 추가 옵셋 및 셋업 옵셋의 경우 S_DLNUM 참조
				THOUSANDS: 0 또는 고유 D 번호
				TEN THOUSANDS: 0 또는 고유 D 번호 1~32000, MD의 고유 D 번호가 설정된 경우
				HUNDRED THOUSANDS: 공구 옵셋 [2)] 0 = 지정 안함 (공구 형상 옵셋) 1 = 길이 L1 옵셋 2 = 길이 L2 옵셋 3 = 길이 L3 옵셋 4 = 반경 옵셋
				ONE MILLION: 공구 옵셋 [2)] 0 = 지정 안함 (공구 길이 마모 옵셋) 1 = 공구 옵셋, 추가 옵셋 (AO) [5)] 공구 옵셋 값을 기존 AO에 추가 2 = 공구 옵셋, 셋업 옵셋 (SO) [5)] SO (신규) = SO (기존) + AO (기존) 옵셋 값, AO (신규) = 0 3 = 공구 옵셋, 셋업 옵셋 (SO) [5)] 공구 옵셋 값을 기존 SO에 추가 4 = 공구 옵셋, 기하.
				TEN MILLION: 공구 옵셋 [2)] 0 = 지정 안함 (공구 형상 일반 옵셋, 반전 없음) 1 = 반전 옵셋
				HUNDRED MILLIONS: 공구 옵셋 0 = 대체 공구가 없는 공구 옵셋 1 = 대체 공구의 공구 옵셋(_DP)
4	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호(프로브 번호 아님) (기본값 = 1)	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
5	X0	S_SETV	지령치	
6	X	S_MA	측정 축 번호(기본값=1)	
			값:	1 = 평면의 첫 번째 축 (G18의 경우 Z) 2 = 평면의 두 번째 축 (G18의 경우 X) 3 = 평면의 세 번째 축 (G18의 경우 Y) [8]
7	X1	S_SZA	측정된 축의 우회 거리	
8	Y1	S_SZO	우회 축의 우회 거리	
9	DFA	S_FA	측정 경로	
10	TSA	S_TSA	안전 영역	
11	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (기본값 = 1)	
12	T	S_TNAME	공구 이름 [2]	
13	DL	S_DLNUM	셋업/추가 옵셋 DL 번호 [5]	
14	ST	_DP	수정할 대체 공구 번호 (듀플로 번호)	
15	TZL	S_TZL	워크 옵셋 [2], [4]	
16	DIF	S_TDIF	치수 차이 검사 [2], [4]	
17	TUL	S_TUL	공차 상한값 (지령치로 증분) [4]	
18	TLL	S_TLL	공차 하한값 (지령치로 증분) [4]	
19	TMV	S_TMV	평균 계산을 위한 옵셋 범위 [2]	
20	FW	S_K	평균 계산을 위한 가중치 [2]	
21	EVN	S_EVNUM	실험값 메모리 번호 [2], [7]	
22		S_MCBIT	예비	
23		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
24		_AMODE	대체 모드	
			값:	UNITS: 치수 공차 사용 여부 0 = 아니오 1 = 예

1) 모든 기본값 = 0 또는 기본값 = x로 표시

2) 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

3) 반전 없이 측정의 경우 WO 수정만 가능

4) 공구 옵셋의 경우 채널 MD 20360 TOOL_PARAMETER_DEF_MASK 적용

5) "셋업/추가 옵셋" 기능이 일반 MD 18108 $MN_MM_NUM_SUMCORR 에 설정된 경우만 해당. 또한 일반 MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK 의 비트 8이 1로 설정되어 있어야 합니다.

6) MD에 WO "미세" 수정이 설정되지 않은 경우 WO를 "일반" 수정

7) 공구 옵셋에만 실험 값 평균 계산 가능
실험 평균 값 메모리의 값 범위:
실험 값 메모리 번호 1~20, 채널별 SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1] 참조
평균 값 메모리 번호 10000~200000 (n), 채널별 SD55625 $SCS_MEA_AVERAGE_VALUE[n-1] 참조

8) 기계에 Y 축이 있는 경우

CYCLE976 측정 싸이클 파라미터

```
PROC CYCLE976(INT S_MVAR,INT S_PRNUM,REAL S_SETV,REAL S_SETV0,INT S_MA,INT S_MD,REAL
S_FA,REAL S_TSA,REAL S_VMS,REAL S_STA1,INT S_NMSP,INT S_SETV1,INT _DMODE,INT _AMODE)
```

도표 4-11 CYCLE976 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식(기본값 = 1000)	
			값:	UNITS: 표면, 계측 구면 또는 계측 링에서 계측 [2)] 0 = 지령치를 알고 있는 표면에서 길이 1 = 직경 (지령치) 및 중심점을 있고 있는 계측 링의 반경 2 = 직경 (지령치) 은 알고 중심점은 모르는 계측 링의 반경 3 = 계측 구면의 반경 및 길이 4 = 지령치를 알고 있는 에지에서 반경. 참고: 측정 축 및 측정 방향 선택. [3)] 5 = 지령치 및 에지 안전 거리를 알고 있는 2개 에지 사이의 반경. 측정 축을 선택해야 합니다.
				TENS: 예비 0 = 0
				HUNDREDS: 예비 0 = 0
				THOUSANDS: 계측 중 측정 축 및 측정 방향 선택 0 = 지정 안함 (측정 축 및 측정 방향 선택 불필요) [8)] 1 = 측정 축 및 측정 방향 선택 지정, S_MA, S_MD 참조 (측정 축 1개에 측정 방향 1개 선택) 2 = 측정 축 선택 지정 S_MA (측정 축 1개에 측정 방향 2개 선택)
				TEN THOUSANDS: 위치 편향 (프로브 스큐) 지정 [2)] 0 = 프로브의 위치 편향 지정 [6)] 1 = 위치 편향 지정 안함
				HUNDRED THOUSANDS: 패럭시얼 계측 또는 특정 각도에서 계측 0 = 활성 WCS에서 패럭시얼 계측 1 = 특정 각도에서 계측 [7)]
				ONE MILLION: 표면 또는 구면에서 계측 중 공구 길이 지정 0 = 공구 길이가 결정되지 않음 1 = 공구 길이가 결정됨 [4)] 2 = 인피드 축이 구면에서 계측됨, 공구 길이가 결정됨, 공구 길이 측정 차이가 계측 데이터에 입력됨

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호(프로브 번호 아님) (기본값 = 1)	
3		S_SETV	지령치	
4	Z0	S_SETV0	구면 계측을 위한 세로 방향 기준의 지령치	
5	X / Y / Z	S_MA	측정 축(축 번호) [2), 6)] (기본값 = 1)	
			값:	1 = 평면의 첫 번째 축 (G17의 경우 X) 2 = 평면의 두 번째 축 (G17의 경우 Y) 3 = 평면의 세 번째 축 (G17의 경우 Z)
6	+-	S_MD	측정 방향 [2), 6)]	
			값:	0 = 양의 방향 1 = 음의 방향
7	DFA	S_FA	측정 경로	
8	TSA	S_TSA	안전 영역	
9	VMS	S_VMS	계측을 위한 가변 측정 속도 [2)]	
10	α	S_STA1	시작 각도 [2), 5)]	
11	측정	S_NMSP	동일 위치에서 측정 횟수 [2)] (기본값 = 1)	
12	X0	S_SETV1	2개 에지 사이를 계측할 때 에지 원점 [3)]	
13		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
14		_AMODE	대체 모드	

[1)] 모든 기본값 = 0 또는 기본값 = x로 표시

[2)] 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

[3)] "계측 링에서 반경" 계측의 경우 링의 직경 및 중심점을 알아야 합니다 (4개 측정 방향).
"두 에지에서 반경" 계측의 경우 측정 축 방향에서 에지까지의 거리를 알아야 합니다 (2개 측정 방향).
"한 에지에서 반경" 계측의 경우 표면의 지령치를 알아야 합니다.

[4)] 표면에서 계측 (표면에서 길이) 측정 방식만 해당. S_MD 및 S_MA에 따라 공구 길이 수정.

5) "중심점을 알고 있는 계측 링" 측정 방식만 해당 (S_MVAR=1xxx02).

6) 계측 방식이 S_MVAR=0 ,=xx1x01 ,=xx2x01 또는 =20000
의 측정 축만 해당: "표면에서 계측" → 측정 축 및 측정 방향 선택
"중심점을 알고 있는 계측 링" → 한 측정 방향 선택 및 측정 축과 측정 방향 선택
또는 "중심점을 알고 있는 계측 링" → 두 측정 방향 선택 및 측정 축 선택
또는 "프로브 길이 결정" → S_MA=3 → 평면의 세 번째 축 (G17의 경우 Z)

7) 계측 링 또는 계측 구면에서 계측 측정 방식만 해당
"계측 구면에서 계측" 및 특정 각도에서 계측의 경우 축이 구면의 균분원을 따라 회전.

8) 중심점을 모르는 "계측 링에서 반경" 계측의 경우 평면에 4개 측정 방향 지정 (G17의 경우 +X, -X, +Y, -Y).
"표면에서 길이" 계측의 경우 공구 축의 음의 방향 (G17의 경우 -Z)

CYCLE978 측정 싸이클 파라미터

```
PROC CYCLE978(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL
S_TSA,INT S_MA,INT S_MD,INT S_NMSP,STRING[32] S_TNAME,INT S_DLNUM,REAL S_TZL,REAL
S_TDIF,REAL S_TUL,REAL S_TLL,REAL S_TMV,INT S_K,INT S_EVNUM,INT S_MCBIT,INT _DMODE,INT
_AMODE,INT _DP)
```

도표 4-12 CYCLE978 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식	
			값:	UNITS: 형상 요소 0 = 표면 측정
				TENS: 예비
				HUNDREDS: 수정 대상 0 = 측정만 수행 (WO 또는 공구 옵셋 수정 안함) 1 = 측정, WO 결정 및 수정 (S_KNUM 참조) 2 = 측정 및 공구 옵셋 (S_KNUM1 참조)
				THOUSANDS: 예비
				TEN THOUSANDS: 스핀들 반전 포함/제외한 측정 또는 전환 방향으로 측정 프로브 정렬 [9)] 0 = 스핀들 반전 없이 측정, 프로브 정렬 안함 1 = 스핀들을 반전시켜 측정 2 = 전환 방향으로 프로브 정렬

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	워크 옵셋(WO) 또는 기본 WO 또는 기본 원점의 수정 [2)]	
			값:	UNITS:
				TENS: 0 = 수정 안함 워크 옵셋 번호 1~99 기본 옵셋 번호 1~16
				HUNDREDS: 예비
				THOUSANDS: WO, 기본 WO 또는 기본 원점 수정 0 = 조정 가능한 WO 수정 1 = 채널별 기본 WO 수정 2 = 기본 원점 수정 3 = 전역 기본 WO 수정 9 = 활성 WO 수정 또는 G500의 경우 마지막으로 활성화된 채널별 기본 WO 수정
				TEN THOUSANDS: WO, 기본 WO 또는 기본 원점 일반/미세 수정 0 = 미세 수정 [6)] 1 = 일반 수정

번호	화면 양식 파라미터	싸이클 파라미터	의미	
3	선택	S_KNUM1	공구 옵셋 수정 [2]	
			값:	UNITS:
				TENS:
				HUNDREDS: 0 = 수정 안함 공구 옵셋을 위한 D 번호 1~999, 추가 옵셋 및 셋업 옵셋의 경우 S_DLNUM 참조
				THOUSANDS: 0 또는 고유 D 번호
				TEN THOUSANDS: 0 또는 고유 D 번호 1~32000, MD의 고유 D 번호가 설정된 경우
				HUNDRED THOUSANDS: 공구 옵셋 [2] 0 = 지정 안함 (공구 형상 옵셋) 1 = 길이 L1 옵셋 2 = 길이 L2 옵셋 3 = 길이 L3 옵셋 4 = 반경 옵셋
				ONE MILLION: 공구 옵셋 [2] 0 = 지정 안함 (공구 반경 마모 옵셋) 1 = 공구 옵셋, 추가 옵셋 (AO) [5] 공구 옵셋 값을 기존 AO에 추가 2 = 공구 옵셋, 셋업 옵셋 (SO) [5] SO (신규) = SO (기존) + AO (기존) 옵셋 값, AO (신규) = 0 3 = 공구 옵셋, 셋업 옵셋 (SO) [5] 공구 옵셋 값을 기존 SO에 추가 4 = 공구 옵셋, 기하
				TEN MILLION: 공구 옵셋 [2] 0 = 지정 안함 (공구 형상 일반 옵셋, 반전 없음) 1 = 반전 옵셋
				HUNDRED MILLIONS: 공구 옵셋 0 = 대체 공구가 없는 공구 옵셋 1 = 대체 공구의 공구 옵셋(_DP)
4	아이콘+번호	S_PRNUM	프로브 파라미터 필드의 번호 (프로브 번호 아님) (값 범위 1~40)	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
5	X0	S_SETV	지령치	
6	DFA	S_FA	측정 경로	
7	TSA	S_TSA	안전 영역	
8	X	S_MA	측정 축 번호 [7] (값 범위 1~3)	
			값:	1 = 평면의 첫 번째 축 (G17의 경우 X) 2 = 평면의 두 번째 축 (G17의 경우 Y) 3 = 공구 방향에서의 평면 (G17 Z의 경우) 측정의 세 번째 축
9		S_MD	측정 축의 측정 방향	
			값:	1 = 양의 측정 방향 2 = 음의 측정 방향
10	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (값 범위 1~9)	
11	TR	S_TNAME	공구 이름 [3]	
12	DL	S_DLNUM	셋업/추가 옵셋 DL 번호 [5]	
13	ST	_DP	수정할 대체 공구 번호 (듀플로 번호)	
14	TZL	S_TZL	워크 옵셋 [2], [3]	
15	DIF	S_TDIF	치수 차이 검사 [2], [3]	
16	TUL	S_TUL	공차 상한값 (지령치로 증분) [3]	
17	TLL	S_TLL	공차 하한값 (지령치로 증분) [3]	
18	TMV	S_TMV	평균 계산을 위한 옵셋 범위 [2]	
19	FW	S_K	평균 계산을 위한 가중치 [2]	
20	EVN	S_EVNUM	데이터 세트, 실험값 메모리 [2], [8]	
21		S_MCBIT	예비	
22		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
23		_AMODE	대체 모드	
			값:	UNITS: 치수 공차 사용 여부 0 = 아니오 1 = 예

1) 모든 기본값 = 0 또는 값 범위 a~b로 표시

2) 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

3) 공구 옵셋만 해당. 그 외 경우 파라미터 = ""

4) 공구 옵셋 및 치수 공차 사용 여부가 "예"인 경우만 해당. 그 외 경우 파라미터 = 0

5) "셋업/추가 옵셋" 기능이 일반 MD 18108 $MN_MM_NUM_SUMCORR 에 설정된 경우만 해당. 또한 일반 MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK 의 비트 8이 1로 설정되어 있어야 합니다.

6) MD에 WO "미세" 수정이 설정되지 않은 경우 WO를 "일반" 수정

7) 공구 형상 옵셋:
평면에서 측정하는 경우 (S_MA=1 또는 S_MA=2) 공구 반경 옵셋
공구 방향에서 측정하는 경우 (S_MA=3) 공구 길이 L1 옵셋

8) 공구 옵셋 및 WO 수정의 경우 실험값 평균 계산 가능.
실험값 평균값 메모리의 값 범위:
실험값 메모리 번호 1~20, 채널별 SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1] 참조
평균값 메모리 번호 10000~200000 (n), 채널별 SD55625 $SCS_MEA_AVERAGE_VALUE[n-1] 참조

9) 스핀들을 반전시켜 측정하는 경우 프로브의 반경/직경을 정확하게 지정해야 합니다. 이는 링, 에지 또는 구면에서 CYCLE976의 계측 방식을 통해 이뤄집니다. 그렇지 않으면 측정 결과가 정확하지 않습니다.

CYCLE998 측정 싸이클 파라미터

```
PROC CYCLE998(INT S_MVAR,INT S_KNUM,INT S_RA,INT S_PRNUM,REAL S_SETV,REAL S_STA1,REAL
S_INCA,REAL S_FA,REAL S_TSA,INT S_MA,INT S_MD,REAL S_ID,REAL S_SETV0,REAL S_SETV1,REAL
S_SETV2,REAL S_SETV3,INT S_NMSP,INT S_EVNUM,INT _DMODE,INT _AMODE)
```

도표 4-13 CYCLE998 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식(기본값 = 5)	
			값:	UNITS: 형상 요소 5 = 에지 측정 (각도 1개) 6 = 평면 측정 (각도 2개)
				TENS: 예비
				HUNDREDS: 수정 대상 0 = 측정만 실행, WO 수정 안함 1 = 측정, WO 결정 및 수정 (S_KNUM 참조)
				THOUSANDS: 보호 영역 0 = 보호 영역 적용 안함 1 = 보호 영역 적용.
				TEN THOUSANDS: 스핀들을 반전시켜 측정 (차동 측정) 0 = 스핀들 반전 없이 측정 1 = 스핀들을 반전시켜 측정
				HUNDRED THOUSANDS: 특정 각도에서 측정 또는 패럭시얼 측정 0 = 특정 각도에서 측정 1 = 패럭시얼 측정

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	워크 옵셋(WO) 또는 기본 WO 또는 기본 원점의 수정 [2]	
			값:	UNITS:
				TENS: 0 = 수정 안함 워크 옵셋 번호 1~99 기본 옵셋 번호 1~16
				HUNDREDS: 예비
				THOUSANDS: WO, 기본 WO 또는 기본 원점 수정 0 = 조정 가능한 WO 1 수정 = 채널별 기본 WO 2 수정 = 기본 원점 수정 9 = 활성 WO 수정 또는 G500의 경우 마지막으로 활성화된 채널별 기본 WO 수정
				TEN THOUSANDS: WO, 기본 WO 또는 기본 원점 coarse/fine 수정 [3] 0 = Fine 수정 1 = Coarse 수정
3	A, B, C	S_RA	옵셋 대상 좌표 회전 또는 로터리 축 [14]	
			값:	0 = 파라미터 S_MA 에 따라 축을 중심으로 수정 대상 좌표 회전 [4] >0 = 수정 대상 로터리 축. 채널 축 개수, 로터리 축 (가능한 경우 로터리 테이블) 개수. 로터리 축 WO의 변환 콤포넌트에서 각도 옵셋이 이뤄집니다.
4	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호 (기본값 =1)	
5	DX / DY / DZ	S_SETV	시작 위치부터 측정 축 (S_MA) 의 측정 지점 P1까지의 거리 (증분) [5]	
6	α	S_STA1	평면의 첫 번째 축 (G17의 경우 X) 을 기준으로 "에지 정렬" 또는 "평면 정렬"하는 경우 각도 지령치 [9]	
7	β	S_INCA	평면의 두 번째 축 (G17의 경우 Y) 을 기준으로 "평면 정렬"하는 경우 각도 지령치 [9]	
8	DFA	S_FA	측정 경로	
9	TSA	S_TSA	안전 영역 각도 지령치 [°] 와의 각도 차이 모니터링 [6]	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
10	X / Y / Z	S_MA	측정 축, 옵셋 축 [7] (기본값 = 201)	
			값:	UNITS: 측정 축 번호 1 = 평면의 첫 번째 축 (G17의 경우 X) 2 = 평면의 두 번째 축 (G17의 경우 Y) 3 = 평면의 세 번째 축 (G17의 경우 Z)
				TENS: 예비
				HUNDREDS: 옵셋 축 번호 1 = 평면의 첫 번째 축 (G17의 경우 X) 2 = 평면의 두 번째 축 (G17의 경우 Y) 3 = 평면의 세 번째 축 (G17의 경우 Z)
11	+-	S_MD	측정 축의 측정 방향 [8]	
			값:	0 = 측정 축의 지령치와 실제 위치로 측정 방향 결정 (호환성) 1 = 양의 측정 방향 2 = 음의 측정 방향
12	L2	S_ID	"에지 정렬" 측정 방식의 경우: 옵셋 축의 측정 지점 P1과 P2 간의 거리 (증분) (값 > 0) "평면 정렬" 측정 방식에는 다음 파라미터가 적용됩니다.	
13	L2	S_SETV0	평면의 첫 번째 축에 있는 측정 지점 P1과 P2 간의 거리 [10]	
14		S_SETV1	평면의 두 번째 축에 있는 측정 지점 P1과 P2 간의 거리 [11], [12]	
15	L3x	S_SETV2	평면의 첫 번째 축에 있는 측정 지점 P1과 P3 간의 거리 [11]	
16	L3y	S_SETV3	평면의 두 번째 축에 있는 측정 지점 P1과 P3 간의 거리 [10]	
17	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (기본값 = 1)	
18		S_EVNUM	데이터 세트, 실험값 메모리 [2], [13]	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
19		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
20		_AMODE	예비 (대체 모드)	

1) 모든 기본값 = 0 또는 기본값 = x로 표시

2) 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

3) 수정 대상이 로터리 축이고 MD 52207 $MCS_AXIS_USAGE_ATTRIB[n] 비트 6 = 1인 경우에만 WO "fine". MD에 WO가 설정되지 않은 경우 수정이 WO "coarse"를 따릅니다.

4) 좌표 회전 옵셋의 예: S_MA=102 측정 축이 Y, 옵셋 축이 X면 Z를 중심으로 좌표 회전을 실시합니다 (G17의 경우).

5) 보호 영역 사용 여부가 "예"인 경우에만 유효한 값 (S_MVAR THOUSANDS 위치 = 1)

6) 옵셋 축에서 측정 지점 P1에서 측정 지점 P2로 포지셔닝할 때 S_STA1 및 S_TSA 파라미터에 설정된 각도를 추가합니다.

7) 측정 축의 번호는 옵셋 축의 번호와 달라야 합니다 (예: 101은 사용할 수 없습니다).

8) "에지 정렬" 및 "패럭시얼 측정" (S_MVAR=10x105) 에만 적용되는 측정 방향

9) "에지 정렬"의 경우 각도 범위 S_STA1 ±45
"평면 정렬"의 경우 각도 범위 S_STA1 0~+60 및 S_INCA ±30

10) "에지 정렬" 및 "평면 정렬" 측정 방식만 해당

11) "평면 측정" 및 "패럭시얼 측정" 측정 방식만 해당

12) 측정 싸이클 버전 SW04.04는 해당 안됨

13) WO에서 수정을 위한 경험치 생성; 실험값 평균 값의 메모리 값 범위:
실험값 메모리 1~20번 (n), 채널별 SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1] 참조

14) 파라미터 S_RA는 측정 방식 "에지 정렬"(S_MVAR xxxxx5)에만 관련됩니다.

CYCLE977 측정 싸이클 파라미터

```
PROC CYCLE977(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,REAL S_SETV0,REAL
S_SETV1,REAL S_FA,REAL S_TSA,REAL S_STA1,REAL S_ID,REAL S_SZA,REAL S_SZO,INT S_MA,INT
S_NMSP,STRING[32] S_TNAME,INT S_DLNUM,REAL S_TZL,REAL S_TDIF,REAL S_TUL,REAL S_TLL,REAL
S_TMV,INT S_K,INT S_EVNUM,INT S_MCBIT,INT _DMODE,INT _AMODE,REAL S_XM,REAL_S_YM,INT _DP)
```

도표 4-14 CYCLE977 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식	
			값:	UNITS: 형상 요소 (값 범위 1~6) 1 = 홀 측정 2 = 스피곳 (샤프트) 측정 3 = 홈 측정 4 = 립 측정 5 = 사각형 측정, 내부 6 = 사각형 측정, 외부
				TENS: 예비
				HUNDREDS: 수정 대상 0 = 측정만 수행 (WO 또는 공구 옵셋 수정 안함) 1 = 측정, WO 결정 및 수정 (S_KNUM 참조) 2 = 측정 및 공구 옵셋 (S_KNUM1 참조)
				THOUSANDS: 보호 영역 0 = 보호 영역 적용 안함 1 = 보호 영역 적용.
				TEN THOUSANDS: 스핀들 반전 (차동 측정) 포함/제외한 측정 또는 전환 방향으로 측정 프로브 정렬 0 = 스핀들 반전 없이 측정, 프로브 정렬 안함 1 = 스핀들을 반전시켜 측정 2 = 전환 방향으로 프로브 정렬

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	워크 옵셋(WO) 또는 기본 WO 또는 기본 원점의 수정 [2)]	
			값:	UNITS:
				TENS: 0 = 수정 안함 워크 옵셋 번호 1~99 기본 옵셋 번호 1~16
				HUNDREDS: 예비
				THOUSANDS: WO, 기본 WO 또는 기본 원점 수정 0 = 조정 가능한 WO 수정 1 = 채널별 기본 WO 수정 2 = 기본 원점 수정 3 = 전역 기본 WO 수정 9 = 활성 WO 수정 또는 G500의 경우 마지막으로 활성화된 채널별 기본 WO 수정
				TEN THOUSANDS: WO, 기본 WO 또는 기본 원점 일반/미세 수정 0 = 미세 수정 [6)] 1 = 일반 수정

번호	화면 양식 파라미터	싸이클 파라미터	의미	
3	선택	S_KNUM1	공구 옵셋 수정 [2]	
			값:	UNITS:
				TENS:
				HUNDREDS: 0 = 수정 안함 공구 옵셋을 위한 D 번호 1~999, 추가 옵셋 및 셋업 옵셋의 경우 S_DLNUM 참조
				THOUSANDS: 0 또는 고유 D 번호
				TEN THOUSANDS: 0 또는 고유 D 번호 1~32000, MD의 고유 D 번호가 설정된 경우
				HUNDRED THOUSANDS: 공구 옵셋 [2] 0 = 지정 안함 (공구 반경 옵셋) 1 = 길이 L1 옵셋 2 = 길이 L2 옵셋 3 = 길이 L3 옵셋 4 = 반경 옵셋
				ONE MILLION: 공구 옵셋 [2] 0 = 지정 안함 (공구 반경 마모 옵셋) 1 = 공구 옵셋, 추가 옵셋 (AO) [5] 공구 옵셋 값을 기존 AO에 추가 2 = 공구 옵셋, 셋업 옵셋 (SO) [5] SO (신규) = SO (기존) + AO (기존) 옵셋 값, AO (신규) = 0 3 = 공구 옵셋, 셋업 옵셋 (SO) [5] 공구 옵셋 값을 기존 SO에 추가 4 = 공구 옵셋, 기하
				TEN MILLION: 공구 옵셋 [2] 0 = 지정 안함 (공구 형상 일반 옵셋, 반전 없음) 1 = 반전 옵셋
				HUNDRED MILLIONS: 공구 옵셋 0 = 대체 공구가 없는 공구 옵셋 1 = 대체 공구의 공구 옵셋(_DP)
4	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호 (프로브 번호 아님) (값 범위 1~40)	

번호	화면 양식 파라미터	싸이클 파라미터	의미
5	X0	S_SETV	지령치
6	X0	S_SETV0	평면의 첫 번째 축 (G17의 경우 X) 에 있는 사각형 지령치
7	Y0	S_SETV1	평면의 두 번째 축 (G17의 경우 Y) 에 있는 사각형 지령치
8	XM	S_XM	지령치 중심점 입력 기하 축 X
9	YM	S_YM	지령치 중심점 입력 기하 축 Y
10	DFA	S_FA	측정 경로
11	TSA	S_TSA	안전 영역
12	α 0	S_STA1	시작 각도
13	DZ	S_ID	절대치/증분치 1. 평면의 세 번째 축 (G17의 경우 Z) 의 증분 절입 S_ID 부호로 결정되는 절입 방향 스피곳, 립 및 사각형 외부 측정의 경우 S_ID 를 사용하여 측정 높이 감소량을 정의합니다. 2. 보호 영역 적용 홀, 홈 및 사각형 내부 측정에 보호 영역을 적용하는 경우 S_ID 를 사용하여 초과 이동 높이를 정의합니다.
14	X1	S_SZA	보호 영역의 직경 또는 길이 (너비) [7]
15	Y1	S_SZO	"사각형 측정"의 경우: 평면의 두 번째 축의 보호 영역 너비
16	X	S_MA	측정 축의 번호 [7] (홈 또는 립 측정만 해당) 값: 1 = 평면의 첫 번째 축 (G17의 경우 X) 2 = 평면의 두 번째 축 (G17의 경우 Y)
17	ST	_DP	수정할 대체 공구 번호 (듀플로 번호)
18	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (값 범위 1~9)
19	TR	S_TNAME	공구 이름 [2]
20	DL	S_DLNUM	셋업/추가 옵셋 DL 번호 [5]
21	TZL	S_TZL	워크 옵셋 [2], [4]
22	DIF	S_TDIF	치수 차이 검사 [2], [4]
23	TUL	S_TUL	공차 상한값 (지령치로 증분) [4]
24	TLL	S_TLL	공차 하한값 (지령치로 증분) [4]
25	TMV	S_TMV	평균 계산을 위한 옵셋 범위 [2]
26	FW	S_K	평균 계산을 위한 가중치 [2]
27		S_EVNUM	데이터 세트, 실험값 평균값 메모리 [2], [8]
28		S_MCBIT	예비

번호	화면 양식 파라미터	싸이클 파라미터	의미	
29		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
30		_AMODE	대체 모드	
			값:	UNITS: 치수 공차 사용 여부 0 = 아니오 1 = 예

1) 모든 기본값 = 0 또는 값 범위 a~b로 표시

2) 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

3) 공구 옵셋만 해당. 그 외 경우 파라미터 = ""

4) 공구 옵셋 및 치수 공차 사용 여부가 "예"인 경우만 해당. 그 외 경우 파라미터 = 0

5) "셋업/추가 옵셋" 기능이 일반 MD 18108 $MN_MM_NUM_SUMCORR에 설정된 경우만 해당. 또한 일반 MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK 의 비트 8이 1로 설정되어 있어야 합니다.

6) MD에 WO "미세" 수정이 설정되지 않은 경우 WO를 "일반" 수정

7) 홀 또는 홈 내부의 보호 영역 직경 또는 너비
스피곳 또는 립 외부의 보호 영역 직경 또는 너비

8) 공구 옵셋에만 실험값 평균 계산 가능
실험값 평균값 메모리의 값 범위:
실험값 메모리 번호 1~20, 채널별 SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1] 참조
평균값 메모리 번호 10000~200000 (n), 채널별 SD55625 $SCS_MEA_AVERAGE_VALUE[n-1] 참조

CYCLE961 측정 싸이클 파라미터

```
PROC CYCLE961(INT S_MVAR,INT S_KNUM,INT S_PRNUM,REAL S_SETV0,REAL S_SETV1,REAL
S_SETV2,REAL S_SETV3,REAL S_SETV4,REAL S_SETV5,REAL S_SETV6,REAL S_SETV7,REAL S_SETV8,REAL
S_SETV9,REAL S_STA1,REAL S_INCA,REAL S_ID,REAL S_FA,REAL S_TSA,INT S_NMSP,INT S_MCBIT,INT
_DMODE,INT _AMODE)
```

도표 4-15 CYCLE961 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식(기본값 ≥ 6)	
			값:	UNITS: 형상 요소 5 = 직각 내부 코너 설정, 각도 및 거리 A1~A3의 지령치 지정 6 = 직각 외부 코너 설정, 각도 및 거리 A1~A3의 지령치 지정 7 = 내부 코너 설정, 각도 및 거리 A1~A4의 지령치 지정 8 = 외부 코너 설정, 각도 및 거리 A1~A4의 지령치 지정
				TENS: 거리 또는 4개 지점으로 지령치 지정 0 = 거리 (극좌표) 로 지령치 지정 1 = 4개 지점을 사용하여 지령치 지정 (측정 지점 P1~P4)
				HUNDREDS: 수정 대상 0 = 측정만 수행 *(WO 수정 또는 공구 옵셋 안함) 1 = 측정, WO 결정 및 수정, S_KNUM 참조
				THOUSANDS: 보호 영역 0 = 보호 영역 적용 안함 1 = 보호 영역 (장애물) 적용, S_ID 참조
				TEN THOUSANDS: WCS에서 코너 위치 0 = 코너 위치를 파라미터 S_STA1 을 통해 (호환성) 1 = 측정 시작 위치에 있는 코너 위치 1 [6)] 2 = 코너 위치 2, 평면의 첫 번째 축(G17의 경우 X)에서 거리가 음수 (S_SETV0, S_SETV1, 참조) 3 = 코너 위치 3, 평면의 첫 번째 및 두 번째 축(G17의 경우 XY)에서 거리가 음수(S_SETV0 ~ S_SETV3 참조) 4 = 코너 위치 4, 평면의 두 번째 축(G17의 경우 Y)에서 거리가 음수 (S_SETV2, S_SETV3 참조)

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	워크 옵셋(WO) 또는 기본 WO 또는 기본 원점의 수정 [2]	
			값:	UNITS:
				TENS: 0 = 수정 안함 워크 옵셋 번호 1~99 기본 옵셋 번호 1~16
				HUNDREDS: 예비
				THOUSANDS: WO, 기본 WO 또는 기본 원점 수정 0 = 조정 가능한 WO 1 수정 = 채널별 기본 WO 2 수정 = 기본 원점 수정 9 = 활성 WO 수정 또는 G500의 경우 마지막으로 활성화된 채널별 기본 WO 수정
				TEN THOUSANDS: WO, 기본 WO 또는 기본 원점 일반/미세 수정 0 = 미세 수정 [5] 1 = 일반 수정
3	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호 (프로브 번호 아님) (값 범위 1~40)	
4	L1/X1	S_SETV0	평면의 첫 번째 축 (G17의 경우 X) 에서 측정 지점 P1과 극점 간의 거리 L1 [3] (실제 거리 L1 = 0인 경우 L1 = M_SETV1 / 2를 자동 계산) 또는 평면의 첫 번째 축 (G17의 경우 X) 의 시작점 P1x [4]	
5	L2/Y1	S_SETV1	평면의 첫 번째 축에서 측정 지점 P2와 극점 간의 거리 L2 [3] 또는 평면의 두 번째 축 (G17의 경우 Y) 의 시작점 P1y [4]	
6	L3/X2	S_SETV2	평면의 두 번째 축에서 측정 지점 P3과 극점 간의 거리 L3 [3] (실제 거리 L3 = 0인 경우 직각인 아닌 코너의 경우 L3 = M_SETV3 / 2를 자동 계산) 또는 평면의 첫 번째 축의 시작점 P2x [4]	
7	L4/Y2	S_SETV3	코너가 직각이 아닌 경우 평면의 두 번째 축에서 측정 지점 P3과 극점 간의 거리 L4 [3] 또는 평면의 두 번째 축의 시작점 P2y [4]	
8	XP/X3	S_SETV4	평면의 첫 번째 축에서 극점 위치 [3] 또는 평면의 첫 번째 축의 시작점 P3x [4]	
9	XP/Y3	S_SETV5	평면의 두 번째 축에서 극점 위치 [3] 또는 평면의 두 번째 축의 시작점 P3y [4]	
10	X4	S_SETV6	평면의 첫 번째 축의 시작점 P4x [4]	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
11	Y4	S_SETV7	평면의 두 번째 축의 시작점 P4y [4]	
12	X0	S_SETV8	WO 수정을 위해 평면의 첫 번째 축에서 측정된 코너의 지령치	
13	Y0	S_SETV9	WO 수정을 위해 평면의 두 번째 축에서 측정된 코너의 지령치	
14	α0	S_STA1	평면의 첫 번째 축의 양의 방향부터 MCS의 공작물 기준 에지까지 시작 각도 (+/-270)	
15	α1	S_INCA	직각이 아닌 코너를 측정할 때 공작물 기준 에지들 간의 각도 [7]	
16	DZ	S_ID	보호 영역이 활성인 경우 각 측정 지점에서 측정 높이 기준 절입량 (S_MVAR 참조)	
17	DFA	S_FA	측정 경로	
18	TSA	S_TSA	안전 영역 각도 지령치 [ ] 와의 각도 차이 모니터링	
19	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (값 범위 1~9) [2]	
20		S_MCBIT	예비	
21		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
22		_AMODE	대체 모드	

[1] 모든 기본값 = 0 또는 값 범위 a~b로 표시

[2] 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

[3] 증분 각도 S_INCA의 지점 3 또는 4 측정 시 시작 각도 S_STA1 을 고려하여 극 좌표에 측정 지점 입력

[4] 직각 좌표계에 측정 지점 입력 (4개 지점을 사용하여 입력)

[5] MD에 WO "미세" 수정이 설정되지 않은 경우 WO를 "일반" 수정.

[6] 각도 값 범위 S_INCA: -180 ~ +180°

[7] 시작 각도 S_STA1, 값 범위: 직각 코너: +- 90도, 모든 코너: +/- 45도

CYCLE979 측정 싸이클 파라미터

```
PROC CYCLE979(INT S_MVAR,INT S_KNUM,INT S_KNUM1,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL
S_TSA,REAL S_CPA,REAL S_CPO,REAL S_STA1,REAL S_INCA,INT S_NMSP,STRING[32] S_TNAME,REAL
S_DLNUM,REAL S_TZL,REAL S_TDIF,REAL S_TUL,REAL S_TLL,REAL S_TMV,INT S_K,INT S_EVNUM,INT
S_MCBIT,INT _DMODE,INT _AMODE,REAL S_XM,REAL S_YM,INT _DP)
```

도표 4-16 CYCLE979 호출 파라미터 [0)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식	
			값:	UNITS: 형상 요소 1 = 홀 측정 2 = 스피곳 (샤프트) 측정
				TENS: 예비
				HUNDREDS: 수정 대상 0 = 측정만 수행 (WO 또는 공구 옵셋 수정 안함) 1 = 측정, WO 결정 및 수정 (S_KNUM 참조) 2 = 측정 및 공구 옵셋 (S_KNUM1 참조)
				THOUSANDS: 측정 지점 개수 0 = 측정 지점 3개 1 = 측정 지점 4개
				TEN THOUSANDS: 스핀들 반전 (차동 측정) 포함/제외한 측정 또는 전환 방향으로 측정 프로브 정렬 0 = 스핀들 반전 없이 측정, 프로브 정렬 안함 1 = 스핀들을 반전시켜 측정 2 = 전환 방향으로 프로브 정렬

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	워크 옵셋(WO) 또는 기본 WO 또는 기본 원점의 수정 [2)]	
			값:	UNITS:
				TENS: 0 = 수정 안함 워크 옵셋 번호 1~99 기본 옵셋 번호 1~16
				HUNDREDS: 예비
				THOUSANDS: WO, 기본 WO 또는 기본 원점 수정 0 = 조정 가능한 WO 수정 1 = 채널별 기본 WO 수정 2 = 기본 원점 수정 3 = 전역 기본 WO 수정 9 = 활성 WO 수정 또는 G500의 경우 마지막으로 활성화된 채널별 기본 WO 수정
				TEN THOUSANDS: WO, 기본 WO 또는 기본 원점 일반/미세 수정 0 = 미세 수정 [6)] 1 = 일반 수정

번호	화면 양식 파라미터	싸이클 파라미터	의미	
3	선택	S_KNUM1	공구 옵셋 수정 [2]	
			값:	UNITS:
				TENS:
				HUNDREDS: 0 = 수정 안함 공구 옵셋을 위한 D 번호 1~999, 추가 옵셋 및 셋업 옵셋의 경우 S_DLNUM 참조
				THOUSANDS: 0 또는 고유 D 번호
				TEN THOUSANDS: 0 또는 고유 D 번호 1~32000, MD의 고유 D 번호가 설정된 경우
				HUNDRED THOUSANDS: 공구 옵셋 [2] 0 = 지정 안함 (공구 반경 옵셋) 1 = 길이 L1 옵셋 2 = 길이 L2 옵셋 3 = 길이 L3 옵셋 4 = 반경 옵셋
				ONE MILLION: 공구 옵셋 [2] 0 = 지정 안함 (공구 반경 마모 옵셋) 1 = 공구 옵셋, 추가 옵셋 (AO) [5] 공구 옵셋 값을 기존 AO에 추가 2 = 공구 옵셋, 셋업 옵셋 (SO) [5] SO (신규) = SO (기존) + AO (기존) 옵셋 값, AO (신규) = 0 3 = 공구 옵셋, 셋업 옵셋 (SO) [5] 공구 옵셋 값을 기존 SO에 추가 4 = 공구 옵셋, 기하
				TEN MILLION: 공구 옵셋 [2] 0 = 지정 안함 (공구 형상 일반 옵셋, 반전 없음) 1 = 반전 옵셋
				HUNDRED MILLIONS: 공구 옵셋 0 = 대체 공구가 없는 공구 옵셋 1 = 대체 공구의 공구 옵셋(_DP)
4	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호 (프로브 번호 아님) (값 범위 1~40)	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
5	X0	S_SETV	지령치	
6	DFA	S_FA	측정 경로	
7	TSA	S_TSA	안전 영역	
8	X0	S_CPA	평면의 첫 번째 축 (G17의 경우 X) 의 중심점	
9	Y0	S_CPO	평면의 두 번째 축 (G17의 경우 Y) 의 중심점	
10	XM	S_XM	지령치 중심점 입력 기하 축 X	
11	YM	S_YM	지령치 중심점 입력 기하 축 Y	
12	α 0	S_STA1	시작 각도 [7]	
13	α 1	S_INCA	증분 각도 [8]	
14	측정	S_NMSP	동일 위치에서 측정 횟수 [1] (값 범위 1~9)	
15	ST	_DP	수정할 대체 공구 번호 (듀플로 번호)	
16	T	S_TNAME	공구 이름 [2]	
17	DL	S_DLNUM	셋업/추가 옵셋 DL 번호 [1], [4]	
18	TZL	S_TZL	워크 옵셋 [1], [2]	
19	DIF	S_TDIF	치수 차이 검사 [1], [2]	
20	TUL	S_TUL	공차 상한값 (지령치로 증분) [2]	
21	TLL	S_TLL	공차 하한값 (지령치로 증분) [2]	
22	TMV	S_TMV	평균 계산을 위한 옵셋 범위 [1]	
23	FW	S_K	평균 계산을 위한 가중치 [1]	
24		S_EVNUM	데이터 세트, 실험값 메모리 [1], [6]	
25		S_MCBIT	예비	
26		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
27		_AMODE	대체 모드	
			값:	UNITS: 치수 공차 사용 여부 0 = 아니오 1 = 예

[0] 모든 기본값 = 0 또는 값 범위 a~b로 표시

[1] 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

[2] 공구 옵셋만 해당. 그 외 경우 파라미터 = ""

[3] 공구 옵셋 및 치수 공차 사용 여부가 "예"인 경우만 해당. 그 외 경우 파라미터 = 0

[4] "셋업/추가 옵셋" 기능이 일반 MD 18108 $MN_MM_NUM_SUMCORR 에 설정된 경우만 해당.

[5] MD에 WO "미세" 수정이 설정되지 않은 경우 WO를 "일반" 수정

[6] 공구 옵셋에만 실험 값 평균 계산 가능
실험 평균 값 메모리의 값 범위:
실험 값 메모리 번호 1~20, 채널별 SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1] 참조
평균 값 메모리 번호 10000~200000 (n), 채널별 SD55625 $SCS_MEA_AVERAGE_VALUE[n-1] 참조

[7] 시작 각도 값 범위: -360~+360°

[8] 측정 지점이 4개인 경우 증분 각도의 값 범위 >0~≤90 , 측정 지점이 3개인 경우 증분 각도의 값 범위 >0~≤120 .

CYCLE997 측정 싸이클 파라미터

```
PROC CYCLE997 (INT S_MVAR,INT S_KNUM,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL S_TSA,REAL
S_STA1,REAL S_INCA,REAL S_SETV0,REAL S_SETV1,REAL S_SETV2,REAL S_SETV3,REAL S_SETV4,REAL
S_SETV5,REAL S_SETV6,REAL S_SETV7,REAL S_SETV8,REAL S_TNVL,INT S_NMSP,INT S_MCBIT,INT
_DMODE,INT _AMODE)
```

도표 4-17 CYCLE997 호출 파라미터 [1), 2)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식(기본값 = 9)	
			값:	UNITS: 형상 요소 9 = 구면 측정
				TENS: 반복 측정 0 = 반복 측정 안함 1 = 반복 측정
				HUNDREDS: 수정 대상 0 = 측정만 수행 (WO 수정 안함) 1 = 측정, WO 결정 및 수정 (S_KNUM 참조)
				THOUSANDS: 측정 방식 0 = 패럭시얼 측정, 시작 각도 없음, SD55740에 따른 프로브 정렬, 비트 1 1 = 회전 측정, 시작 각도 있음, SD55740에 따른 프로브 정렬, 비트 1 2 = 회전 측정, 시작 각도 있음, 전환 방향으로 프로브 정렬 3 = 패럭시얼 측정, 시작 각도 있음, SD55740에 따른 프로브 정렬, 비트 1 4 = 패럭시얼 측정, 시작 각도 있음, 전환 방향으로 프로브 정렬
				TEN THOUSANDS: 측정할 구면의 개수 0 = 구면 1개 측정 1 = 구면 3개 측정
				HUNDRED THOUSANDS: 측정 지점 개수, 특정 각도에서 측정인 경우만 해당 (측정 방식 참조: THOUSANDS 위치 > 0) 0 = 특정 각도에서 측정 지점 3개 측정(구면을 따라 이송) 1 = 특정 각도에서 측정 지점 4개 측정(구면을 따라 이송)
				ONE MILLION: 구면의 직경 지령치 결정 0 = 구면의 직경 지령치 결정 안함 1 = 구면의 직경 지령치 결정

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	워크 옵셋(WO), 기본 WO 또는 기본 원점 수정 [3]	
			값:	UNITS:
				TENS: 0 = 수정 안함 워크 옵셋 번호 1~99 기본 옵셋 번호 1~16
				HUNDREDS: 예비
				THOUSANDS: WO, 기본 WO 또는 기본 원점 수정 0 = 조정 가능한 WO 수정 1 = 채널별 기본 WO 수정 2 = 기본 원점 수정 3 = 전역 기본 WO 수정 [7] 9 = 활성 WO 수정 또는 G500의 경우 마지막으로 활성화된 채널별 기본 WO 수정
				TEN THOUSANDS: WO, 기본 WO 또는 기본 원점 일반/미세 수정 0 = 미세 수정 [6] 1 = 일반 수정
3	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호 (프로브 번호 아님) (값 범위 1~40)	
4		S_SETV	구면의 직경 [4]	
5	DFA	S_FA	측정 경로	
6	TSA	S_TSA	안전 영역	
7	α 0	S_STA1	특정 각도에서 측정하는 경우 시작 각도	
8	α 1	S_INCA	특정 각도에서 측정하는 경우 증분 각도	
9	X1	S_SETV0	구면 3개 측정의 경우 평면의 첫 번째 축(G17의 경우 X)에 있는 첫 번째 구면의 설정 위치	
10	Y1	S_SETV1	구면 3개 측정의 경우 평면의 두 번째 축(G17의 경우 Y)에 있는 첫 번째 구면의 설정 위치	
11	Z1	S_SETV2	구면 3개 측정의 경우 평면의 세 번째 축(G17의 경우 Z)에 있는 첫 번째 구면의 설정 위치	
12	X2	S_SETV3	구면 3개 측정의 경우 평면의 첫 번째 축의 두 번째 구면의 설정 위치	
13	Y2	S_SETV4	구면 3개 측정의 경우 평면의 두 번째 축의 두 번째 구면의 설정 위치	
14	Z2	S_SETV5	구면 3개 측정의 경우 평면의 세 번째 축의 두 번째 구면의 설정 위치	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
15	X3	S_SETV6	구면 3개 측정의 경우 평면의 첫 번째 축의 세 번째 구면의 설정 위치	
16	Y3	S_SETV7	구면 3개 측정의 경우 평면의 두 번째 축의 세 번째 구면의 설정 위치	
17	Z3	S_SETV8	구면 3개 측정의 경우 평면의 세 번째 축의 세 번째 구면의 설정 위치	
18	TVL	S_TNVL	구면 3개 측정의 경우 삼각형 변형(편차의 합) 한계값 [5]	
19	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (값 범위 1~9)	
20		S_MCBIT	예비	
21		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
22		_AMODE	대체 모드	

[1] 모든 기본값 = 0 또는 값 범위 a~b로 표시

[2] 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

[3] 중간 포지셔닝, 구면의 균분원을 따라 회전

[4] 구면 3개 측정: 세 구면 모두에 동일한 직경 지령치 적용(_SETV)

[5] 기본값 S_TNVL=1.2
WO 수정: 확인된 변형이 S_TNVL 한계값 미만인 경우만 WO에서 수정을 실시합니다.

[6] MD에 WO "미세" 수정이 설정되지 않은 경우 WO를 "일반" 수정

[7] "구면 3개 측정" 방식의 경우 전역 기본 프레임(S_KNUM = 3001~3016) 에 회전 콤포넌트가 없기 때문에 수정할 수 없습니다.

CYCLE995 측정 싸이클 파라미터

```
PROC CYCLE995 (INT S_MVAR,INT S_KNUM,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL S_TSA,REAL
S_STA1,REAL S_INCA,REAL S_DZ,REAL S_SETV0,REAL S_SETV1,REAL S_SETV2,REAL S_TUL,REAL
S_TZL,INT S_NMSP,INT S_MCBIT,INT _DMODE,INT _AMODE)
```

도표 4-18 CYCLE995 호출 파라미터 [1)]

번호	화면상의 파라미터 표시	싸이클 파라미터	의미	
1		S_MVAR	측정 방식 (디폴트 = 5)	
			값	UNITS: 형상 요소 5 = 스핀들 기하 (공구 축에 평행)
				TENS: 반복 측정 1 = 반복 측정 사용
				HUNDREDS: 옵셋 대상 없음 0 = 측정만 수행
				THOUSANDS: 측정 방식 2 = 특정 각도에서 측정, 전환 방향으로 측정 프로브 정렬
				TEN THOUSANDS: 측정할 구면의 개수 0 = 구면 측정
				HUNDRED THOUSANDS: 측정 지점 개수 1 = 특정 각도 (구면 원호) 에서 측정할 때 4개 측정 지점
				ONE MILLION: 구면의 직경 지령치 결정 0 = 구면의 직경 지령치 결정 안함 1 = 구면의 직경 지령치 결정
2	선택	S_KNUM	수정 대상 0 = 0	
3	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호 (프로브 번호 아님) (값 범위 1~40)	
4	DM	S_SETV	계측 볼의 직경 [4)]	
5	DFA	S_FA	측정 경로	
6	TSA	S_TSA	안전 영역 [5)]	
7	α 0	S_STA1	특정 각도에서 측정하는 경우 시작 각도 [3)]	
8		S_INCA	특정 각도에서 측정하는 경우 증분 각도 [2)]	
9	DZ	S_DZ	프로브 샤프트에서 첫 번째 측정 P1에서 두 번째 측정 P2까지의 거리	
10		S_SETV0	평면의 첫 번째 축(G17의 경우 X)에 있는 구면의 지령치 위치 [2)]	

번호	화면상의 파라미터 표시	싸이클 파라미터	의미	
11		S_SETV1	평면의 두 번째 축(G17의 경우 Y)에 있는 구면의 지령치 위치 [2]	
12		S_SETV2	평면의 세 번째 축(G17의 경우 Z)에 있는 구면의 지령치 위치 [2]	
13	TUL	S_TUL	각도 편차의 공차 상한값	
14	TZL	S_TZL	워크 옵셋 범위 [1], [4]	
15	번호	S_NMSP	동일 위치에서 측정 횟수 [2] (값 범위 1~9)	
16		S_MCBIT	예비 [2]	
17		_DMODE	표시 모드	
			값	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
18		_AMODE	대체 모드	
			값	UNITS: 치수 공차 사용 여부 0 = 아니오 1 = 예

모든 디폴트 값 = 0 또는 값 범위 a~b로 표시

[1] 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

[2] 파라미터는 사용되지 않으며 입력 화면에 표시되지 않습니다.
파라미터 증분 각도 S_INCA는 90°로 고정되어 있습니다.

[3] 시작 각도 값 범위: -360~+360°

[4] 치수 공차의 경우 예:
측정된 각도가 워크 옵셋 범위 TZL의 값보다 작은 경우 각도 (_OVR[2], _OVR[3]) 및 편차 (_OVR[7], _OVR[8])의 결과 파라미터는 0으로 설정됩니다.
TZL 표시는 일반 셋팅 데이터 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE bit25=1을 사용하여 수행됩니다.
(각도, 스핀들을 측정할 때 선택한 워크 옵셋 활성화)

[5] 파라미터 TSA는 계측 구면의 첫 번째 측정값을 나타냅니다.

CYCLE996 측정 싸이클 파라미터

```
PROC CYCLE996(INT S_MVAR,INT S_TC,INT S_PRNUM,REAL S_SETV,REAL S_STA1,REAL S_SETV0,REAL
S_SETV1,REAL S_SETV2,REAL S_SETV3,REAL S_SETV4,REAL S_SETV5,REAL S_TNVL,REAL S_FA,REAL
S_TSA,INT S_NMSP,INT S_MCBIT,INT _DMODE,INT _AMODE)
```

도표 4-19 CYCLE996 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식(기본값 = 1)	
			값:	UNITS: 측정 순서 0 = 좌표계 계산 (선택 방법: 결과 디스플레이, 프로토콜 및 스위블 데이터 세트 변경(필요한 경우 작업자 승인), _AMODE 참조. 1 = 1차 측정 2 = 2차 측정 3 = 3차 측정
				TENS: 예비 0 = 0
				HUNDREDS: 1차~3차 측정의 측정 방식 0 = 계측 구면 패럭시얼 측정 1 = 스핀들 수정 없이 계측 구면을 특정 각도에서 측정[3)] 2 = 프로브의 전환 방향에서 계측 구면 측정 및 스핀들 수정 [3)] 3 = 패럭시얼 측정, 시작 각도 있음 [8)] 4 = 패럭시얼 측정, 시작 각도 있음, 프로브 전환 방향으로 스핀들 이동 [8)]
				THOUSANDS: 좌표계의 수정 대상 계산 [4)] 0 = 측정만 수행. 스위블 데이터 세트를 계산하지만 변경하지 않음 1 = 스위블 데이터 세트를 계산. 필요한 경우 작업자가 승인하면 스위블 데이터 세트 변경 [4)]
				TEN THOUSANDS: 측정 축 (로터리 축 1 또는 2) 또는 좌표계 계산의 경우 벡터 체인 열기/닫기 0 = 벡터 체인 닫기(좌표계 계산만 해당) 1 = 로터리 축 1(1차~3차 측정만 해당) 2 = 로터리 축 2(1차~3차 측정만 해당) [5)] 3 = 벡터 체인 열기(좌표계 계산만 해당)
				HUNDRED THOUSANDS: 좌표계 계산의 경우 로터리 축 1 정규화 0 = 로터리 축 1 배율 없음 1 = 평면의 첫 번째 축(G17의 경우 X) 방향에서 배율 2 = 평면의 두 번째 축(G17의 경우 Y) 방향에서 배율 3 = 평면의 세 번째 축(G17의 경우 Z) 방향에서 배율
				ONE MILLION: 좌표계 계산의 경우 로터리 축 2 정규화 [5)] 0 = 로터리 축 2 배율 없음 1 = 평면의 첫 번째 축(G17의 경우 X) 방향에서 배율

번호	화면 양식 파라미터	싸이클 파라미터	의미	
				2 = 평면의 두 번째 축(G17의 경우 Y) 방향에서 배율 3 = 평면의 세 번째 축(G17의 경우 Z) 방향에서 배율
				TEN MILLION: 로그 파일 0 = 프로토콜 파일 없음 1 = 계산된 벡터(공구 캐리어)가 포함된 프로토콜 파일 생성 및 MD에 설정된 경우 1차 동적 5축 변환(TRAORI(1)).
2		S_TC	스위블 데이터 레코드 (공구 캐리어) 번호	
3	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호(프로브 번호 아님) (기본값 = 1)	
4		S_SETV	계측 구면의 직경	
5	α 0	S_STA1	특정 각도에서 측정하는 경우 시작 각도	
6	α 0	S_SETV0	로터리 축 1의 위치 값 (로터리 축이 수동 또는 반자동인 경우)	
7	α 1	S_SETV1	로터리 축 2의 위치 값 (로터리 축이 수동 또는 반자동인 경우) [6]	
8	XN	S_SETV2	로터리 축 1 정규화를 위한 위치 값	
9	XN	S_SETV3	로터리 축 2 정규화를 위한 위치 값 [6]	
10	델타	S_SETV4	옵셋 벡터 I1~I4의 공차 값	
11	델타	S_SETV5	로터리 축 벡터 V1 및 V2의 공차 값	
12	TVL	S_TNVL	로터리 축의 각도 세그먼트 한계값(값 범위 1~60°) (기본값 = 20) [7]	
13	DFA	S_FA	측정 경로	
14	TSA	S_TSA	안전 영역	
15	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (기본값 = 1)	
16		S_MCBIT	예비	
17		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)

번호	화면 양식 파라미터	싸이클 파라미터	의미	
18		_AMODE	대체 모드	
			값:	UNITS: 공차 검사 여부 0 = 아니오 1 = 예: 벡터 S_SETV4 및 S_SETV5의 공차 값 평가
				TENS: 계산된 벡터를 스위블 데이터 세트에 입력할 때 작업자가 승인 [4] 0 = 예: 변경을 위해서 작업자 승인 필요 1 = 아니오: 계산된 벡터를 즉시 입력 (HUNDREDS 및 THOUSANDS 위치가 0인 경우만 해당)
				HUNDREDS: 측정 결과 표시 [5] 0 = 아니오 1 = 예
				THOUSANDS: 측정 결과 편집 가능 여부 0 = 아니오 1 = 예: 편집 가능(HUNDREDS 위치 = 1인 경우만 해당)

1) 모든 기본값 = 0 또는 기본값 = x로 표시

2) 표시 방식은 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 결정

3) 예를 들어 90° 각도 위치에 이 측정 방식을 사용하는 경우 계측 구면을 고정하는 샤프트와 충돌하지 않고 계측 구면에서 좌표계를 측정할 수 있습니다. 시작 각도 S_STA1 (0~360°)를 입력할 수 있습니다. 구면을 회전시킬 때 증분 각도는 90°입니다.
원호 경로를 따라 이송할 때는 채널별 SD55634 $SCS_MEA_FEED_PLANE_VALUE 의 이송 속도를 사용합니다.

4) 입력하기 전에 M0에서 작업자 프롬프트가 표시됩니다. NC 시작을 해야 벡터가 입력됩니다.
RESET 으로 측정 프로그램을 중단하면 계산된 벡터가 입력되지 않습니다.
벡터는 계산 중에 옵셋 벡터 공차를 초과하지 않은 경우에만 입력됩니다.

5) 측정 결과는 좌표계 계산 측정 방식에만 표시됩니다.
1차~3차 측정 후에도 측정 결과를 표시하려면 채널별 SD55613 $SCS_MEA_RESULT_DISPLAY을 설정하면 됩니다.

6) 로터리 축이 2개인 좌표계에만 로터리 축 2를 사용할 수 있습니다.

7) 로터리 축의 각도 세그먼트 한계값. S_TNVL 의 값 범위는 20~60°입니다. S_TNVL 의 값이 < 20°인 경우 프로브의 마이크로미터 범위에서 측정이 부정확해지기 때문에 정확도가 떨어질 수 있습니다. 한계값을 위반하면 에러 메시지 61430이 출력됩니다. 이 메시지에 최소 한계값이 함께 표시됩니다.

8) SD54760 비트 17 = 1인 경우 프로브 전환 방향으로 스핀들이 이동합니다.

CYCLE9960 측정 싸이클 파라미터

```
PROC CYCLE9960(INT S_MVAR,STRING[40] S_TNAME,INT S_PRNUM,REAL S_SETV,REAL S_SETV1,REAL
S_START_RA1,REAL S_END_RA1,INT S_CMEA_RA1,REAL S_POS_RA2,REAL S_SETV2,REAL
S_START_RA2,REAL S_END_RA2,INT S_CMEA_RA2,REAL S_POS_RA1,REAL S_SETV4,REAL S_FA,REAL
S_TSA,INT S_NMSP,INT S_DMODE,INT S_AMODE,INT S_KNUM)
```

도표 4-20 CYCLE9960 호출 파라미터 [1)]

번호	화면상의 파라미터 표시	싸이클 파라미터	의미	
1	선택	S_MVAR	측정 방식(기본값 = 1)	
			값:	UNITS: 측정 방식 0 = 좌표계 측정 및 계산(선택: 결과 디스플레이, 프로토콜, 스위블 데이터 세트 변경(천 단위 S_MVAR 참조)) 1 = 기준 헤드 측정 2 = 헤드를 기준 헤드에 맞도록 조정 3 = 보간 지점 측정 및 계산(E996)
				TENS: 예비
				HUNDREDS: 볼 측정 방식 2 = 프로브의 전환 방향으로 계측 볼 측정 및 스핀들 추적 4 = 패럭시얼 측정, 시작 각도, 프로브의 전환 방향으로 스핀들 추적 [2)]
				THOUSANDS: 좌표계의 수정 대상 계산 [3)] 0 = 측정 및 계산. 데이터 세트가 계산되고 변경되지 않음 1 = 오퍼레이터 확인 후 계산된 데이터 세트를 변경할 수 있음 2 = 이전에 측정된 좌표계를 계산하고, 해당되는 경우 오퍼레이터 확인 후 변경
				TENTHOUSANDS: 측정 축(로터리 축 1 또는 2) 1 = 모든 기존 로터리 축을 측정 및 계산 4 = 로터리 축 1만 측정 및 계산 5 = 로터리 축 2만 측정 및 계산
2		S_TNAME	변환 이름(키네마틱 체인에 기반한 스위블 데이터 세트 또는 변환)	
3	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호(프로브 번호 아님) (기본값 = 1)	
4		S_SETV	계측 볼의 직경	

번호	화면상의 파라미터 표시	싸이클 파라미터	의미	
5	α 1	S_SETV1	첫 번째 로터리 축의 각도에서 측정하기 위한 시작 각도	
6			S_START_RA1: 첫 번째 로터리 축의 시작 각도	
7			S_END_RA1: 첫 번째 로터리 축의 최종 각도	
8			S_CMEA_RA1: 첫 번째 로터리 축의 측정 횟수, 좌표계 측정 및 계산의 경우 3. 보간 지점의 경우 최대 12회의 측정이 가능합니다.	
9			S_POS_RA2: 첫 번째 로터리 축을 측정하는 동안 두 번째 로터리 축의 위치 [4)]	
10	α 2	S_SETV2	두 번째 로터리 축의 각도에서 측정하기 위한 시작 각도 [4)]	
11			S_START_RA2: 두 번째 로터리 축의 시작 각도 [4)]	
12			S_END_RA2: 두 번째 로터리 축의 최종 각도 [4)]	
13			S_CMEA_RA2: 두 번째 로터리 축의 측정 횟수, 좌표계 측정 및 계산의 경우 3. 보간 지점의 경우 최대 12회의 측정이 가능합니다.[4)]	
14			S_POS_RA1: 두 번째 로터리 축을 측정하는 동안 첫 번째 로터리 축의 위치	
15	델타	S_SETV4	옵셋 벡터의 공차 값	
16	DFA	S_FA	측정 경로	
17	TSA	S_TSA	안전 영역	
18	숫자	S_NMSP	동일 위치에서 측정 횟수 [5)] (기본값 = 1)	
19		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전에 활성 상태이던 평면은 활성 상태 유지 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)

번호	화면상의 파라미터 표시	싸이클 파라미터	의미	
20		_AMODE	대체 모드	
			값:	ONES: 공차 확인 여부 0 = 아니오 1 = 예: 벡터의 공차 값 평가 -> S_SETV4, S_SETV5
				TENS: 자동 계측 0 = 자동 계측 없음 1 = 자동 계측
				HUNDREDS: 자동 시작 각도 0 = 설정된 시작 각도가 사용됨 1 = 각 측정 지점별로 시작 각도의 자동 계산
21		S_KNUM	기준 헤드 측정을 위해 수정할 WO의 번호	
			값	UNITS: 수정은 항상 coarse 옵셋에 적용되고, fine 옵셋은 삭제됩니다.
				TENS: 1 ... 99 조정 가능 WO의 번호(1=G54)
				THOUSANDS: 9 수정 활성화, 조정 가능 WO

1) 모든 기본값 = 0 또는 기본값 = xx로 표시

2) SD54760 bit 17 = 1인 경우 프로브의 전환 방향으로 스핀들이 추적됩니다.

3) 입력하기 전에 M0에서 작업자 프롬프트가 표시됩니다. NC가 시작될 때까지 벡터를 입력할 수 없습니다. RESET으로 측정 프로그램이 중단되면 계산된 벡터가 입력되지 않습니다. 계산에서 옵셋 벡터의 공차가 초과되지 않는 경우에만 벡터가 입력됩니다.

4) 로터리 축이 2개인 좌표계에만 로터리 축 2를 사용할 수 있습니다.

5) 디스플레이는 일반 SD54760 $SNS_MEA_FUNCTION_MASK_PIECE에 따라 달라집니다.

CYCLE982 측정 싸이클 파라미터

```
PROC CYCLE982(INT S_MVAR,INT S_KNUM,INT S_PRNUM,INT S_MA,INT S_MD,REAL S_ID,REAL S_FA,REAL
S_TSA,REAL S_VMS,REAL S_STA1,REAL S_CORA,REAL S_TZL,REAL S_TDIF,INT S_NMSP,INT S_EVNUM,INT
S_MCBIT,INT _DMODE,INT _AMODE)
```

도표 4-21 CYCLE982 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식	
			값:	UNITS: 계측/측정 0 = 공구 프로브 계측 1 = 단일 공구 측정 [3)] 2 = 여러 공구 측정, 공기 길이 및 반경 (밀링 공구의 경우) 결정
				TENS: MCS 또는 WCS에서 계측/측정 0 = 기계 관련 [4)] 1 = 공작물 관련
				HUNDREDS: 측정 시 밀링 공구 반전 여부 0 = 반전 없이 측정 1 = 반전시켜 측정
				THOUSANDS: 밀링 공구의 수정 대상 0 = 길이 또는 길이 및 반경 결정(S_MVAR 의 첫 번째 위치 참조) 1 = S_MVAR 의 첫 번째 위치가 1인 경우 반경 결정 2 = S_MVAR 의 첫 번째 위치가 1 또는 2인 경우 길이 및 반경 결정(단면) 3 = 측면 밀링 커터 상단 절삭날(후면) 및 길이와 반경 결정 [5)]
				TEN THOUSANDS: 밀링 공구 또는 드릴 위치 0 = 밀링 공구 또는 드릴의 축 위치, 평면의 두 번째 축(G18의 경우 X)에서 반경 [7)] 1 = 밀링 공구 또는 드릴의 반경 위치, 평면의 첫 번째 축(G18의 경우 Z)에서 반경 [7)]
				HUNDRED THOUSANDS: 증분 계측/측정 0 = 지정 안함 1 = 증분 계측/측정
				ONE MILLION: 시작 각도 S_STA1에 스핀들 포지셔닝 (밀링 공구 측정만 해당) 0 = 스핀들 포지셔닝 안 함 1 = 시작 각도 S_STA1에 스핀들 포지셔닝

번호	화면 양식 파라미터	싸이클 파라미터	의미	
2	선택	S_KNUM	옵셋 방식 [2]	
			값:	UNITS: 공구 옵셋 0 = 지정 안함 (기하에서 공구 옵셋) 1 = 마모에서 공구 옵셋
3	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호(프로브 번호 아님) (기본값 = 1)	
4	X0	S_MA	측정 축	
			값:	1 = 평면의 첫 번째 축 (G18의 경우 Z) 2 = 평면의 두 번째 축 (G18의 경우 X)
5	+-	S_MD	측정 방향	
			값:	0 = 선택 안함 (실제값에 따라 측정 방향 결정) 1 = 양의 방향 2 = 음의 방향
6	Z2	S_ID	옵셋	
7	DFA	S_FA	측정 경로	
8	TSA	S_TSA	안전 영역	
9	VMS	S_VMS	계측을 위한 가변 측정 속도 [2]	
10	alpha1	S_STA1	밀링 공구 측정 시 시작 각도	
11	alpha2	S_CORA	반전시켜 밀링 공구 측정 시 옵셋 각도 [8]	
12	TZL	S_TZL	밀링 공구 측정 시 워크 옵셋. S_TZL = 0 계측 시	
13	DIF	S_TDIF	치수 차이 검사	
14	측정	S_NMSP	동일 위치에서 측정 횟수 [2] (기본값 = 1)	
15	EVN	S_EVNUM	실험값 평균값 메모리 번호 [2], [9]	
16		S_MCBIT	예비	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
17		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
				TENS: 선삭 및 밀링 공구의 절삭날 위치 (입력 화면 1~9에 표시하는 경우만 해당)
				HUNDREDS: 공구 종류 0 = 선삭 공구 1 = 밀링 공구 2 = 드릴
				THOUSANDS: 접근 방식은 공구 프로브에 따라 결정 0 = PLUS [X/Z]; 공구 위치가 축인 경우 X, 공구 위치가 반경인 경우 Z 1 = MINUS [X/Z]; 공구 위치가 축인 경우 X, 공구 위치가 반경인 경우 Z
18		_AMODE	대체 모드	
			값:	UNITS: 예비
				TENS: 예비
				HUNDREDS: 예비
				THOUSANDS: 계측을 위한 측정 및 단일 측정 후 시작점에 접근 (S_MVAR - UNITS 참조) 0 = 공구 위치 확인, 프로브 에지를 기준으로 DFA에 의해 옵셋 1 = 시작 위치에 접근

1) 모든 기본값 = 0 또는 기본값 = x로 표시

2) 표시 방식은 일반 SD54762 _MEA_FUNCTION_MASK_TOOL에 따라 결정

3) 선삭 공구, 밀링 공구 또는 드릴 측정. 파라미터 S_MA
에 측정 축 설정. 선삭 공구는 절삭날 위치 1~8로 지정, 밀링 공구는 파라미터 S_MVAR에
HUNDREDS~THOUSANDS 위치로 지정.

4) 측정 및 계측은 기본 좌표계(좌표계 변환이 꺼진 경우 MCS)에서 수행합니다.

5) 증분 측정은 제외

6) 다중 측정 (S_MVAR=x2x02 또는 x3x02) 만 해당 (예: 디스크 유형 또는 홈 밀링 공구)

7) 채널 SD42950 $SC_TOOL_LENGTH_TYPE = 2인 경우 선삭 공구와 동일한 공구 길이 콤포넌트가 지정됩니다.

8) 반전시켜 측정하는 경우만 해당 (S_MVAR=xx1x1)

9) 실험값 생성
실험값 메모리의 값 범위: 실험값 메모리 1~20번 (n), 채널 SD55623 $SCS_MEA_EMPIRIC_VALUE[n-1] 참조.

CYCLE971 측정 싸이클 파라미터

```
PROC CYCLE971(INT S_MVAR,INT S_KNUM,INT S_PRNUM,INT S_MA,INT S_MD,REAL S_ID,REAL S_FA,REAL
S_TSA,REAL S_VMS,REAL S_TZL,REAL S_TDIF,INT S_NMSP,REAL S_F1,REAL S_S1,REAL S_F2,REAL
S_S2,REAL S_F3,REAL S_S3,INT S_EVNUM,INT S_MCBIT,INT _DMODE,INT _AMODE)
```

도표 4-22 CYCLE971 호출 파라미터 [1)]

번호	화면 양식 파라미터	싸이클 파라미터	의미	
1		S_MVAR	측정 방식	
			값:	UNITS: 0 = 공구 프로브 계측 1 = 고정 스핀들을 사용한 공구 측정 (길이 또는 반경) 2 = 회전 스핀들을 사용한 공구 측정 (길이 또는 반경), 파라미터 S_F1 ~S_S4 참조.
				TENS: 기계 좌표계 또는 공작물 좌표계에서 측정 0 = MCS 에서 측정 (기계 관련), 공구 측정 또는 공구 프로브 계측 1 = WCS에서 측정 (공작물 관련), 공구 측정 또는 공구 프로브 계측
				HUNDREDS: 절삭날 개별 검사 0 = 아니오 1 = 예
				THOUSANDS: 0 = 0
				TEN THOUSANDS: 증분 계측/측정 0 = 지정 안함 1 = 증분 계측/측정
				HUNDRED THOUSANDS: 공구 프로브 자동 계측 0 = 공구 프로브 자동 계측 안함 1 = 공구 프로브 자동 계측
				ONE MILLION: 스핀들을 반전시켜 평면에서 계측 0 = 스핀들 반전 없이 평면에서 계측 1 = 스핀들을 반전시켜 평면에서 계측
2	선택	S_KNUM	옵셋 방식 [2)]	
			값:	UNITS: 공구 옵셋 0 = 지정 안함 (기하에서 공구 옵셋) 1 = 마모에서 공구 옵셋
3	아이콘 + 숫자	S_PRNUM	프로브 파라미터 필드의 번호 (프로브 번호 아님)	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
4	X0	S_MA	측정 축, 옵셋 축 [4]	
			값:	UNITS: 측정 축 번호 1 = 평면의 첫 번째 축 (G17의 경우 X) 2 = 평면의 두 번째 축 (G17의 경우 Y) 3 = 평면의 세 번째 축 (G17의 경우 Z)
				TENS: 0 = 0
				HUNDREDS: 옵셋 축 번호 0 = 옵셋 축 아님 1 = 평면의 첫 번째 축(G17의 경우 X) 2 = 평면의 두 번째 축(G17의 경우 Y)
5	+-	S_MD	측정 방향	
			값:	0 = 선택 안함 (실제값에 따라 측정 방향 결정) 1 = 양의 방향 2 = 음의 방향
6	V	S_ID	옵셋	
			값:	0 = 옵셋이 없는 공구의 경우 >0 = • 계측: 계측 공구 직경이 프로브의 상부 직경보다 크면 평면의 세 번째 축 (G17의 경우 Z) 에 옵셋이 적용됩니다. 공구는 공구 반경에서 S_ID 의 값을 뺀 값만큼 프로브 중심에서 옵셋됩니다. 옵셋 축 역시 S_MA 에 지정합니다. • 측정: 절삭날이 여러 개인 경우 반경 측정에는 공구 길이 옵셋 및 절삭날 최고 지점을 지정하고, 길이 측정에는 공구 반경 옵셋 및 절삭날 최고 지점을 지정해야 합니다.
7	DFA	S_FA	측정 경로	
8	TSA	S_TSA	안전 영역	
9	VMS	S_VMS	계측을 위한 가변 측정 속도 [2]	
10	TZL	S_TZL	워크 옵셋 (공구 측정만 해당)	
11	DIF	S_TDIF	공구 측정의 경우 치수 차이 검사 (S_MVAR=xx1 또는 S_MVAR=xx2)	
12	측정	S_NMSP	동일 위치에서 측정 횟수 [2]	
13	F1	S_F1	회전 스핀들과 접촉하는 경우 1차 이송 속도 [2]	
14	S1	S_S1	회전 스핀들과 접촉하는 경우 1차 속도 [2]	

번호	화면 양식 파라미터	싸이클 파라미터	의미	
15	F2	S_F2	회전 스핀들과 접촉하는 경우 2차 이송 속도 [2]	
16	S2	S_S2	회전 스핀들과 접촉하는 경우 2차 속도 [2]	
17	F3	S_F3	회전 스핀들과 접촉하는 경우 2차 이송 속도 [3]	
18	S3	S_S3	회전 스핀들과 접촉하는 경우 2차 속도 [3]	
19	EVN	S_EVNUM	실험값 메모리 번호 [2]	
20		S_MCBIT	_CBIT 또는 _CHBIT의 화면 형식	
21		_DMODE	표시 모드	
			값:	UNITS: 가공 평면 G17/G18/G19 0 = 호환성, 싸이클 호출 전부터 활성이던 평면 1 = G17 (싸이클 중에만 활성) 2 = G18 (싸이클 중에만 활성) 3 = G19 (싸이클 중에만 활성)
22		_AMODE	대체 모드	
			값:	UNITS: 공구 옵셋 반경 측정 1 = 아니오 2 = 예
				TENS: 평면의 세 번째 축에서 반경을 측정할 때 공구 옵셋 방향 (G17의 경우 Z) 1 = 양의 방향 2 = 음의 방향
				HUNDREDS: 길이를 측정할 때 또는 세 번째 축에서 프로브를 계측할 때의 공구 옵셋 0 = 호환성, 자동 1 = 아니오 2 = 예
				THOUSANDS: 옵셋 축에서 길이를 측정할 때 공구 옵셋 방향 (S_MA HUNDREDS 참조) 1 = 양의 방향 2 = 음의 방향

[1] 모든 기본값 = 0 또는 기본값 = x로 표시

[2] 표시 방식은 일반 SD54762 MEA_FUNCTION_MASK_TOOL에 따라 결정

[3] 공구 옵셋 및 치수 공차 사용 여부가 "예"인 경우만 해당. 그 외 경우 파라미터 = 0

[4] 자동 측정 (S_MVAR=1x00xx) 의 경우 측정 축 및 옵셋 축 표시 안함 ⇒ S_MA=0.

CYCLE150 측정 싸이클 파라미터

```
PROC CYCLE150(INT S_PICT,INT S_PROT,STRING[160] S_PATH) SAVE
ACTBLOCNO DISPLOF
```

도표 4-23 CYCLE150 호출 파라미터

번호	화면 파라미터	싸이클 파라미터	의미	
1	측정 결과 화면	S_PICT	결과 표시 선택(기본값 = 0)	
			값:	UNITS: 0 = 측정 결과 화면 OFF 1 = 측정 결과 화면 ON
				Tens: 표시 모드 선택(SD55613의 값) 1 = 측정 결과 화면 표시 - 8초 후 자동으로 선택 해제 3 = 측정 결과 화면 표시 - NC Start를 사용하여 승인 4 = 측정 결과 화면 표시 - 알람에만 해당 (61303 ... 61306)
2		S_PROT	로깅 선택(기본값 = 0)	
	로그		값:	UNITS: 프로토콜 off / on / 마지막 측정 선택 0 = 로그 OFF 1 = 로그 ON 2 = 마지막 측정 로그
	로그 유형			TENS: 로그 유형 선택 0 = 표준 로그 1 = 사용자 로그 (임의로 정의 가능)
	로그 형식			HUNDREDS: 로그 형식 선택 0 = 텍스트 형식 1 = 테이블 형식 (Excel로 들여오는 경우)
	로그 데이터			THOUSANDS: 다시 쓰기 또는 선택 사항 첨부 0 = 새로 만들기 1 = 첨부

번호	화면 파라미터	싸이클 파라미터	의미	
	로그 아카이브			TEN THOUSANDS: 로그 아카이브 선택 0 = 가공 프로그램 1 = 디렉토리
3		S_PATH	로그 아카이브 선택을 위한 로그 파일 경로 (전체 경로 이름 또는 파일 이름만, 예: "//NC:/WKS.DIR/NAME.WPD 또는 "MESSPROTOKOLL.TXT"	

테이블

5

5.1 작업

참고

싸이클

작업 목록은 NC 프로그램(G 코드)에서 발생하는 모든 싸이클을 포함하고 있습니다. 즉, 모든 싸이클은 마스크를 사용하여 프로그램 편집기에서 프로그램 되거나 또는 프로그래밍 지원 없이 루프용으로 프로그램 되어야 합니다. 호환성 목적으로 제어 시스템 내에서 여전히 사용 가능하지만 더 이상 SINUMERIK Operate 프로그램 편집기("호환성 싸이클")를 이용하여 편집할 수 없는 싸이클은 고려되지 않습니다.

작업 A ... C

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
:	O	NC 메인 블록 번호, 점프 레이블 종료, 연결 연산자		+		PM-NC
*	O	곱셈 연산자		+		PM-NC
+	O	덧셈 연산자		+		PM-NC
-	O	뺄셈 연산자		+		PM-NC
<	O	비교 연산자, 보다 작은		+		PM-NC
<<	O	문자열 연결 연산자		+		PM-NC
<=	O	비교 연산자, 작거나 같음		+		PM-NC
=	O	지정 연산자		+		PM-NC
>=	O	비교 연산자, 크거나 같음		+		PM-NC
/	O	나눗셈 연산자		+		PM-NC
/0 ... ... /7		블록 건너뛰기 (첫 번째 건너뛰기 수준) ... ... 블록 건너뛰기 (여덟 번째 건너뛰기 수준)		+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
A	A	축 이름	m/s	+		PM-NC
A2	A	공구 오리엔테이션: RPY 또는 Euler 각도	s	+		PM-NC
A3	A	공구 오리엔테이션: 방향 벡터의 1번째 콤포넌트	s	+		PM-NC
A4	A	공구 오리엔테이션: 블록 시작에서 표면 법선 벡터의 1번째 콤포넌트	s	+		PM-NC
A5	A	공구 오리엔테이션: 블록 끝에서 표면 법선 벡터의 1번째 콤포넌트	s	+		PM-NC
A6	A	공구 오리엔테이션: 테이퍼 회전 축의 방향 벡터의 1번째 콤포넌트	s	+		PM-NC
A7	A	공구 오리엔테이션: 테이퍼의 원주면에서 중간 오리엔테이션의 1번째 벡터 콤포넌트	s	+		PM-NC
ABS	F	절대치 (양)		+	+	PM-NC
AC	K	좌표/위치의 절대 치수	s	+		PM-NC
ACC	K	현재 축 가속의 효과	m	+	+	PM-NC
ACCLIMA	K	현재 최대 축 가속의 효과	m	+	+	PM-NC
ACN	K	로터리 축의 절대 치수, 마이너스 방향으로 접근	s	+		PM-NC
ACOS	F	역코사인 (삼각형. 함수)		+	+	PM-NC
ACP	K	로터리 축의 절대 치수, 양의 방향으로 접근	s	+		PM-NC
ACTBLOCNO	P	"현재 블록 표시 억제" (DISPLOF) 가 활성이어도 발생하는 알람 블록의 현재 블록의 출력 수!		+		PM-NC
ADDFRAME	F	측정된 프레임 포함 및 활성화 가능		+	-	PM-NC, FM-B
ADIS	A	경로 코드 G1, G2, G3, ...의 라운딩 공차	m	+		PM-NC
ADISPOS	A	급 이송 G0의 라운딩 공차	m	+		PM-NC
ADISPOSA	P	IPOBRKA에서 공차 범위 크기	m	+	+	PM-NC
AFISOF	P	자동 필터 체인 전환 비활성화	m	+	-	PM-NC
AFISON	P	자동 필터 체인 전환 활성화	m	+	-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
ALF	A	고속 도피 각도	m	+		PM-NC
AMIRROR	G	프로그래밍 미러링	s	+		PM-NC
AND	K	Logical AND		+		PM-NC
ANG	A	형상 각도	s	+		PM-NC
AP	A	극점 각도	m/s	+		PM-NC
APR	K	읽기/표시 액세스 방지		+		PM-NC
APRB	K	읽기 액세스 권한, OPI		+		PM-NC
APRP	K	읽기 액세스 권한, 가공 프로그램		+		PM-NC
APW	K	쓰기 액세스 방지		+		PM-NC
APWB	K	쓰기 액세스 권한, OPI		+		PM-NC
APWP	K	쓰기 액세스 권한, 가공 프로그램		+		PM-NC
APX	K	지정된 언어 요소를 실행하기 위한 사용 권한 정의		+		PM-NC
AR	A	틈 각도	m/s	+		PM-NC
AROT	G	프로그래밍 회전	s	+		PM-NC
AROTS	G	입체각 프로그래밍 프레임 회전	s	+		PM-NC
AS	K	매크로 정의		+		PM-NC
ASCALE	G	프로그래밍 배율	s	+		PM-NC
ASIN	F	산술 함수, 원호 사인		+	+	PM-NC
ASPLINE	G	Akima 스플라인	m	+		PM-NC
ATAN2	F	아크 탄젠트 2		+	+	PM-NC
ATOL	A	컴프레서 기능을 위한 축 관련 공차, 오리엔테이션 스무딩 및 스무딩 종류	m	+		PM-NC
ATRANS	G	부가 프로그래밍 가능한 워크 옵셋	s	+		PM-NC
AUXFUDEL	P	채널별로 전역 목록에서 보조 기능 삭제		+	-	FM-B
AUXFUDELG	P	채널별로 전역 목록에서 보조 기능 그룹의 모든 보조 기능 삭제		+	-	FM-B
AUXFUMSEQ	P	M 보조 기능의 출력 순서 결정		+	-	FM-B
AUXFUSYNC	P	보조 기능의 전역 목록의 문자열로서 채널 관련 SERUPRO end ASUB에 대한 전체 가공 프로그램 생성		+	-	FM-B

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
AX	K	변수 축 식별자	m/s	+		PM-NC
AXCTSWE	P	축 컨테이너 회전		+	-	PM-NC
AXCTSWEC	P	축 컨테이너 회전 활성화 취소		+	+	PM-NC
AXCTSWED	P	축 컨테이너 회전 (스타트업을 위한 명령 변용)		+	-	PM-NC
AXIS	K	축 이름, 축 주소		+		PM-NC
AXNAME	F	입력 문자열을 축 이름으로 변환		+	-	PM-NC
AXSTRING	F	문자열 스핀들 번호 변환		+	-	PM-NC
AXTOCHAN	P	특정 채널의 축 요청. NC 프로그램과 동기 동작에서 가능.		+	+	PM-NC
AXTOSPI	F	축 이름을 스핀들 인덱스로 변환		+	-	PM-NC
B	A	축 이름	m/s	+		PM-NC
B2	A	공구 오리엔테이션: RPY 또는 Euler 각도	s	+		PM-NC
B3	A	공구 오리엔테이션: 방향 벡터의 콤포넌트	s	+		PM-NC
B4	A	공구 오리엔테이션: 블록 시작에서 표면 법선 벡터의 2번째 콤포넌트	s	+		PM-NC
B5	A	공구 오리엔테이션: 블록 끝에서 표면 법선 벡터의 2번째 콤포넌트	s	+		PM-NC
B6	A	공구 오리엔테이션: 테이퍼 회전 축의 방향 벡터의 2번째 콤포넌트	s	+		PM-NC
B7	A	공구 오리엔테이션: 테이퍼의 원주면에서 중간 오리엔테이션의 2번째 벡터 콤포넌트	s	+		PM-NC
B_AND	O	비트 단위 AND		+		PM-NC
B_OR	O	비트 단위 OR		+		PM-NC
B_NOT	O	비트 단위 부정		+		PM-NC
B_XOR	O	비트 단위 익스클루시브 OR		+		PM-NC
BAUTO	G	다음의 세 점을 사용한 첫 번째 스플라인 섹션 정의	m	+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
BLOCK	K	키워드 TO와 함께 간접 서브프로그램 호출 시 처리할 프로그램 부분 정의		+		PM-NC
BLSYNC	K	인터럽트 루틴의 처리는 다음 블록 변경을 시작할 때만 가능		+		PM-NC
BNAT[6]	G	첫 번째 스플라인 블록으로 자연적인 트렌지션	m	+		PM-NC
BOOL	K	데이터 유형: 부울 값은 TRUE/FALSE 또는 1/0		+		PM-NC
BOUND	F	값이 정의된 범위 내에 있는지 여부를 테스트합니다. 값이 같으면 테스트 값을 반환합니다.		+	+	PM-NC
BRISK[6]	G	빠른 가감속	m	+		PM-NC
BRISKA	P	프로그램한 축에서 brisk 경로 가속 시작		+	-	PM-NC
BSPLINE	G	B 스플라인	m	+		PM-NC
BTAN	G	첫 번째 스플라인 블록으로 접선 트렌지션	m	+		PM-NC
C	A	축 이름	m/s	+		PM-NC
C2	A	공구 오리엔테이션: RPY 또는 Euler 각도	s	+		PM-NC
C3	A	공구 오리엔테이션: 방향 벡터의 3번째 콤포넌트	s	+		PM-NC
C4	A	공구 오리엔테이션: 블록 시작에서 표면 법선 벡터의 3번째 콤포넌트	s	+		PM-NC
C5	A	공구 오리엔테이션: 블록 끝에서 표면 법선 벡터의 3번째 콤포넌트	s	+		PM-NC
C6	A	공구 오리엔테이션: 테이퍼 회전 축의 방향 벡터의 3번째 콤포넌트	s	+		PM-NC
C7	A	공구 오리엔테이션: 테이퍼의 원주면에서 중간 오리엔테이션의 3번째 벡터 콤포넌트	s	+		PM-NC
CAC	K	절대 위치 접근		+		PM-NC
CACN	K	마이너스 방향으로 테이블에 열거된 절대치로 접근		+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CACP	K	양의 방향으로 테이블에 열거된 절대치로 접근		+		PM-NC
CADAPTOF	P	부하 조정 비활성화		+	-	PM-NC
CADAPTON	P	부하 조정 활성화		+	-	PM-NC
CALCDAT	F	3지점 또는 4지점을 사용하여 원호의 반경 및 중심점 계산		+	-	PM-NC
CALCFIR	P	FIR 저크 필터를 동적 응답 모드로 조정		+	-	PM-NC
CALCPOSI	F	보호 영역 위반, 작업 영역 제한 및 소프트웨어 제한 점검		+	-	PM-NC
CALL	K	간접 서브프로그램 호출		+		PM-NC
CALLPATH	P	서브프로그램 호출에서 프로그래밍 검색 경로		+	-	PM-NC
CANCEL	P	모달 동기 동작 취소		+	-	FM-SA
CANCELSUB	P	현재 서브프로그램 레벨 취소		+	+	FM-SA
CASE	K	조건부 프로그램 분기		+		PM-NC
CDC	K	위치 직접 이동		+		PM-NC
CDOF [6]	G	충돌 감지 비활성화	m	+		PM-NC
CDOF2	G	3D 원주 밀링 중 충돌 감지 비활성화	m	+		PM-NC
CDON	G	충돌 감지 활성화	m	+		PM-NC
CFC [6]	G	형상에서의 일정한 이송 속도	m	+		PM-NC
CFIN	G	외부 반경이 아닌 내부 반경에서만 일정한 이송 속도	m	+		PM-NC
CFINE	F	미세 옵셋을 프레임 변수에 지정		+	-	PM-NC
CFTCP	G	공구 중심점의 이송 속도 일정(중심점 경로)	m	+		PM-NC
CHAN	K	데이터의 유효 범위 지정		+		PM-NC
CHANDATA	P	채널 데이터 액세스용 채널 번호 설정		+	-	PM-NC
CHAR	K	데이터 유형: ASCII 문자		+		PM-NC
CHF	A	Chamfer: 값 = 모따기 길이	s	+		PM-NC
CHKDM	F	매거진 내의 고유성 검사		+	-	FM-TM

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CHKDNO	F	독자적 D 번호를 점검		+	-	PM-NC
CHR	A	모따기: 값 = 이동 방향에서 모따기 길이		+		PM-NC
CIC	K	증분에 의한 위치 접근		+		PM-NC
CIP	G	중간점을 통한 원호 보간	m	+		PM-NC
CLEARM	P	채널 코디네이션을 위해 1개/여러 개 마커를 리셋		+	+	PM-NC
CLRINT	P	인터럽트 선택 취소		+	-	PM-NC
CMIRROR	F	좌표 축의 미러링		+	-	PM-NC
COARSEA	K	"일반 정위치 정지"에 도달하면 이동이 끝남	m	+		PM-NC
COLLPAIR	F	충돌 쌍의 일부인지 여부 확인		+		PM-NC
COMPCAD	G	컴프레서 기능 COMPCAD 활성화	m	+		PM-NC
COMPCURV	G	컴프레서 기능 COMPCURV 활성화	m	+		PM-NC
COMPLETE		데이터 읽기 및 쓰기를 위한 제어 명령		+		PM-NC
COMPOF [6]	G	NC 블록 압축 비활성화	m	+		PM-NC
COMPON	G	컴프레서 기능 COMPON 활성화	m	+		PM-NC
COMPSURF	G	컴프레서 기능 COMPSURF 활성화	m	+		PM-NC
CONTDCON	P	테이블 형상 디코딩 활성화		+	-	PM-NC
CONTPRON	P	기준 전처리 활성화		+	-	PM-NC
CORROF	P	모든 활성 오버레이 이동 선택 취소.		+	-	PM-NC
CORRTC	F	기계 측정에 따라 오리엔테이션 전환이 가능한 공구 캐리어의 옵셋 벡터 또는 방향 벡터를 수정하십시오.		+	-	PM-NC
CORRTRAFO	F	기계의 좌표계 모델에서 오리엔테이션 축의 옵셋 벡터 또는 방향 벡터 수정		+	-	PM-NC
COS	F	코사인 (삼각형. 함수)		+	+	PM-NC
COUPDEF	P	ELG 그룹/동기 스핀들 그룹 정의		+	-	PM-NC
COUPDEL	P	ELG 그룹 삭제		+	-	PM-NC
COUPOF	P	ELG 그룹/동기 스핀들 쌍 비활성화		+	-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
COUPOFS	P	추종 스핀들 정지로 ELG 그룹/동기 스핀들 페어 비활성화		+	-	PM-NC
COUPON	P	ELG 그룹/동기 스핀들 쌍 활성화		+	-	PM-NC
COUPONC	P	이전 프로그래밍으로 ELG 그룹/동기 스핀들 페어의 활성화 이송		+	-	PM-NC
COUPRES	P	ELG 그룹 리셋		+	-	PM-NC
CP [6]	G	경로 모션	m	+		PM-NC
CPBC	K	일반 커플링: 블록 변경 기준		+	+	FM-A
CPDEF	K	일반 커플링: 커플링 모듈 생성		+	+	FM-A
CPDEL	K	일반 커플링: 커플링 모듈 삭제		+	+	FM-A
CPFMOF	K	일반 커플링: 완전 해제 상태에서 추종 축의 동작		+	+	FM-A
CPFMON	K	일반 커플링: 설정 상태에서 추종축의 동작		+	+	FM-A
CPFMSON	K	일반 커플링: 동기화 모드		+	+	FM-A
CPFPOS	K	일반 커플링: 추종 축의 동기 위치		+	+	FM-A
CPFRS	K	일반 커플링: 참조 좌표계		+	+	FM-A
CPLA	K	일반 커플링: 리딩 축의 정의		+	-	FM-A
CPLCTID	K	일반 커플링: 커브 테이블 수		+	+	FM-A
CPLDEF	K	일반 커플링: 리딩 축의 정의 및 커플링 모듈의 생성		+	+	FM-A
CPLDEL	K	일반 커플링: 커플링 모듈의 리딩 축 삭제		+	+	FM-A
CPLDEN	K	일반 커플링: 커플링 팩터의 분모		+	+	FM-A
CPLINSC	K	일반 커플링: 리딩 축 입력 값에 대한 배율 계수		+	+	FM-A
CPLINTR	K	일반 커플링: 리딩 축의 입력 값에 대한 옵셋 값		+	+	FM-A
CPLNUM	K	일반 커플링: 커플링 팩터의 분자		+	+	FM-A
CPLOF	K	일반 커플링: 커플링 모듈의 리딩 축 해제		+	+	FM-A
CPLON	K	일반 커플링: 커플링 모듈의 리딩 축 설정		+	+	FM-A

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CPLOUTSC	K	일반 커플링: 커플링의 출력 값에 대한 배율 계수		+	+	FM-A
CPLOUTTR	K	일반 커플링: 커플링의 출력 값에 대한 옵셋 값		+	+	FM-A
CPLPOS	K	일반 커플링: 리딩 축의 동기 위치		+	+	FM-A
CPLSETVAL	K	일반 커플링: 커플링 기준		+	+	FM-A
CPMALARM	K	일반 커플링: 특수 커플링 관련 알람 출력의 억제		+	+	FM-A
CPMBRAKE	K	일반 커플링: 특정 정지 신호 및 정지 명령에 대한 추종 축의 반응		+	-	FM-A
CPMPRT	K	일반 커플링: 기계 잠금을 통해 블록 검색 실행 시 가공 프로그램 시작에서 커플링 반응		+	+	FM-A
CPMRESET	K	일반 커플링: RESET에 대한 커플링 동작		+	+	FM-A
CPMSTART	K	일반 커플링: 가공 프로그램 시작 시 커플링 동작		+	+	FM-A
CPMVDI	K	일반 커플링: 특정 NC/PLC 인터페이스 신호에 대한 추종 축의 반응		+	+	FM-A
CPOF	K	일반 커플링: 커플링 모듈 해제		+	+	FM-A
CPON	K	일반 커플링: 커플링 모듈 설정		+	+	FM-A
CPRECOF[6]	G	형상 정확도 취소	m	+		PM-NC
CPRECON	G	형상 정확도 활성화	m	+		PM-NC
CPRES	K	일반 커플링: 동기 스핀들 커플링의 구성 데이터 활성화		+	-	FM-A
CPROT	P	채널별 보호 영역 활성화/비활성화		+	-	PM-NC
CPROTDEF	P	채널 특정 보호 영역 정의		+	-	PM-NC
CPSETTYPE	K	일반 커플링: 커플링 형식		+	+	FM-A
CPSYNCOP	K	일반 커플링: 위치 동기 "일반"의 임계값		+	+	FM-A
CPSYNCOP2	K	일반 커플링: 위치 동기 "일반" 2의 임계값		+	+	FM-A
CPSYNCOV	K	일반 커플링: 속도 동기 "일반"의 임계값		+	+	FM-A

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CPSYNFIP	K	일반 커플링: 위치 동기 "미세"의 임계값		+	+	FM-A
CPSYNFIP2	K	일반 커플링: 위치 동기 "미세" 2의 임계값		+	+	FM-A
CPSYNFIV	K	일반 커플링: 속도 동기 "미세"의 임계값		+	+	FM-A
CR	A	원호 반경	s	+		PM-NC
CROT	F	현재 좌표계 회전		+	-	PM-NC
CROTS	F	입체각 프로그래밍 프레임 회전 (지정 축에서 회전)	s	+	-	PM-NC
CRPL	F	임의의 평면에서 프레임 회전		+	-	FM-B
CSCALE	F	여러 축의 배율 계수		+	-	PM-NC
CSPLINE	F	큐빅 스플라인	m	+		PM-NC
CT	G	접선 트렌지션에 의한 원호	m	+		PM-NC
CTAB	F	커브 테이블에서 리딩 축 위치에 따라 추종 축 위치를 정의		+	+	PM-NC
CTABDEF	P	테이블 정의 활성화		+	-	PM-NC
CTABDEL	P	커브 테이블 삭제		+	-	PM-NC
CTABEND	P	테이블 정의 비활성화		+	-	PM-NC
CTABEXISTS	F	번호 n에 해당하는 커브 테이블 점검		+	+	PM-NC
CTABFNO	F	메모리에서 여전히 설정 가능한 커브 테이블의 개수		+	+	PM-NC
CTABFPOL	F	메모리에서 여전히 설정 가능한 폴리노미얼 개수		+	+	PM-NC
CTABFSEG	F	메모리에서 여전히 설정 가능한 커브 세그먼트 개수		+	+	PM-NC
CTABID	F	n번째 커브 테이블의 테이블 번호 반환		+	+	PM-NC
CTABINV	F	커브 테이블에서 추종 축 위치에 따라 리딩 축 위치를 정의		+	+	PM-NC
CTABISLOCK	F	번호 n에 해당하는 커브 테이블의 잠금 상태 반환		+	+	PM-NC
CTABLOCK	P	삭제 및 덮어쓰기, 잠금		+	+	PM-NC
CTABMEMTYP	F	커브 테이블 번호 n 작성 시 메모리 반환.		+	+	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CTABMPOL	F	메모리에서 여전히 설정 가능한 폴리노미얼 최대 개수		+	+	PM-NC
CTABMSEG	F	메모리에서 여전히 설정 가능한 커브 세그먼트의 최대 개수		+	+	PM-NC
CTABNO	F	SRAM 또는 DRAM에서 정의한 커브 테이블의 개수		+	+	FM-A
CTABNOMEM	F	SRAM 또는 DRAM에서 정의한 커브 테이블의 개수		+	+	PM-NC
CTABPERIOD	F	커브 테이블 번호 n의 테이블 주기성 반환		+	+	PM-NC
CTABPOL	F	메모리에서 이미 사용한 폴리노미얼의 개수		+	+	PM-NC
CTABPOLID	F	번호 n에 해당하는 커브 테이블에서 사용한 커브 폴리노미얼의 개수		+	+	PM-NC
CTABSEG	F	메모리에서 이미 사용한 커브 세그먼트의 개수		+	+	PM-NC
CTABSEGID	F	번호 n에 해당하는 커브 테이블에서 사용한 커브 세그먼트의 개수		+	+	PM-NC
CTABSEV	F	커브 테이블 분할에서 추종 축의 끝 값 반환		+	+	PM-NC
CTABSSV	F	커브 테이블 분할에서 추종 축의 시작 값 반환		+	+	PM-NC
CTABTEP	F	커브 테이블 끝에서 리딩 축 값 반환		+	+	PM-NC
CTABTEV	F	커브 테이블 끝에서 추종 축 값 반환		+	+	PM-NC
CTABTMAX	F	커브 테이블에서 추종 축의 최대값 반환		+	+	PM-NC
CTABTMIN	F	커브 테이블에서 추종 축의 최소값 반환		+	+	PM-NC
CTABTSP	F	커브 테이블 시작에서 리딩 축 값 반환		+	+	PM-NC
CTABTSV	F	커브 테이블 시작에서 추종 축 값 반환		+	+	PM-NC
CTABUNLOCK	P	삭제 및 덮어쓰기 취소 잠금		+	+	PM-NC
CTOL	A	컴프레서 기능을 위한 형상 공차, 오리엔테이션 스무딩 및 스무딩 종류	m	+	-	PM-NC
CTOLG0	A	급 이송 동작을 위한 형상 공차	m	+	-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CTRANS	F	여러 축에 대한 워크 옵셋		+	-	PM-NC
CUT2D [6]	G	2D TRC	m	+		PM-NC
CUT2DD	G	차동 공구 관련 2½ D TRC	m	+		PM-NC
CUT2DF	G	현재 프레임에 대한 2D TRC(경사 평면)	m	+		PM-NC
CUT2DFD	G	현재 프레임 관련 차동 공구 관련 2½D TRC(경사면)	m	+		PM-NC
CUT3DC	G	원주 밀링을 위한 3D TRC	m	+		PM-NC
CUT3DCC	G	3D 반경 보정을 사용하는 한계 표면을 고려하는 원주 밀링용 3D TRC: 가공 표면의 형상	m	+		PM-NC
CUT3DCCD	G	공구 중심점 경로에서 차동 공구를 사용하는 한계 표면을 고려하는 원주 밀링용 3D TRC: 한계 표면으로 절입	m	+		PM-NC
CUT3DCD	G	원주 밀링의 차동 공구 관련 3D TRC	m	+		PM-NC
CUT3DF	G	오리엔테이션 변경을 사용하는 페이스 밀링을 위한 3D TRC	m	+		PM-NC
CUT3DFD	G	오리엔테이션 변경과 함께 페이스 밀링의 차동 공구 관련 3D TRC	m	+		PM-NC
CUT3DFF	G	일정한 오리엔테이션을 사용하는 페이스 밀링을 위한 3D TRC 공구 오리엔테이션은 G17 - G19에 의해 정의된 방향이고, 일부 경우 프레임에 의해 회전됩니다.	m	+		PM-NC
CUT3DFS	G	일정한 오리엔테이션을 사용하는 페이스 밀링을 위한 3D TRC 공구 오리엔테이션은 G17 - G19에 의해 정의되고, 프레임의 영향은 받지 않습니다.	m	+		PM-NC
CUTCONOF [6]	G	공구 반경 보정 비활성화	m	+		PM-NC
CUTCONON	G	공구 반경 보정 활성화	m	+		PM-NC
CUTMOD	A	회전 공구에 대한 옵셋 데이터 수정 활성화(오리엔테이션 가능 공구 캐리어 관련)	m	+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CUTMODK	A	회전 공구에 대한 옵셋 데이터 수정 활성화(운동학적 체인에 의해 정의된 오리엔테이션 변환 관련)	m	+		PM-NC
CYCLE60	C (T)	조각 싸이클		+		PM-NC
CYCLE61	C (T)	페이스 밀링		+		PM-NC
CYCLE62	C (T)	형상 호출		+		PM-NC
CYCLE63	C (T)	형상 포켓 밀링		+		PM-NC
CYCLE64	C (T)	형상 포켓 사전 드릴링		+		PM-NC
CYCLE70	C (T)	나사 밀링		+		PM-NC
CYCLE72	C (T)	경로 밀링		+		PM-NC
CYCLE76	C (T)	사각 스피곳 밀링		+		PM-NC
CYCLE77	C (T)	원주 밀링		+		PM-NC
CYCLE78	C (T)	절삭 나사 밀링		+		PM-NC
CYCLE79	C (T)	다각형		+		PM-NC
CYCLE81	C (T)	드릴링, 센터링		+		PM-NC
CYCLE82	C (T)	드릴링, 카운터보링		+		PM-NC
CYCLE83	C (T)	심공 드릴링		+		PM-NC
CYCLE84	C (T)	보정 척을 사용하지 않고 태핑		+		PM-NC
CYCLE85	C (T)	리밍		+		PM-NC
CYCLE86	C (T)	보링		+		PM-NC
CYCLE92	C (T)	소재 절단		+		PM-NC
CYCLE95	C (T)	형상과 함께 스톡 제거		+		PM-NC
CYCLE98	C (T)	나사 체인		+		PM-NC
CYCLE99	C (T)	나사 절삭		+		PM-NC
CYCLE116	C (M)	원호의 중심점 및 반경 계산		+		PM-MC
CYCLE119	C (M)	공간 위치 결정		+		PM-MC
CYCLE150	C (M)	측정 결과 디스플레이/로깅		+		PM-MC
CYCLE435	C (T)	드레싱 공구 위치 계산		+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CYCLE495	C (T)	형상 트루잉		+		PM-NC
CYCLE750	C (A)	CYCLE751 ... CYCLE759의 내부 작동 싸이클(실제 기능 호출을 위한 MMC 명령 포함)		-		FM-A
CYCLE751	C (A)	최적화 세션 열기/수행/닫기		M		FM-A
CYCLE752	C (A)	최적화 세션에 축 추가		M		FM-A
CYCLE753	C (A)	최적화 모드 선택		M		FM-A
CYCLE754	C (A)	언어 블록 추가/제거		M		FM-A
CYCLE755	C (A)	데이터 레코드 백업/복원		M		FM-A
CYCLE756	C (A)	최적화 결과 활성화		M		FM-A
CYCLE757	C (A)	최적화 데이터 저장		M		FM-A
CYCLE758	C (A)	파라미터 값 변경		M		FM-A
CYCLE759	C (A)	읽기 파라미터 값		M		FM-A
CYCLE782	C (T)	부하에 맞게 조정		+		PM-NC
CYCLE800	C (T)	스위블		+		PM-NC
CYCLE801	C (T)	격자/프레임		+		PM-NC
CYCLE802	C (T)	임의적 위치		+		PM-NC
CYCLE806	C (T)	보간 선삭		+		PM-NC
CYCLE830	C (T)	심공 드릴링 2		+		PM-NC
CYCLE832	C (T)	고속 가공 설정		+		PM-NC
CYCLE840	C (T)	태핑 척을 사용한 태핑		+		PM-NC
CYCLE899	C (T)	개방형 슬롯 밀링		+		PM-NC
CYCLE930	C (T)	홈		+		PM-NC
CYCLE940	C (T)	언더컷 가공		+		PM-NC
CYCLE951	C (T)	스톡 제거		+		PM-NC
CYCLE952	C (T)	형상 홈 가공		+		PM-NC
CYCLE961	C (M)	공작물 코너(내부 또는 외부) 위치를 확인하여 워크 옵셋으로 삽입합니다.		+		PM-MC
CYCLE971	C (M)	공구 프로브를 조정하고 공구 길이 및/또는 공구 반경을 측정합니다(밀링만 해당).		+		PM-MC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CYCLE973	C (M)	공작물 표면 또는 홈에서 워크 프로브를 조정합니다(선삭만 해당).		+		PM-MC
CYCLE974	C (M)	단일 지점 측정으로 선택한 측정 축에서 공작물 원점을 결정하고 공구 옵셋을 결정합니다(선삭만 해당).		+		PM-MC
CYCLE976	C (M)	작업 평면에 있는 계측 링 또는 계측 볼이나 특정 축 및 방향의 에지에서 워크 프로브를 조정합니다.		+		PM-MC
CYCLE977	C (M)	평면에서 중심과 너비 또는 직경을 결정합니다.		+		PM-MC
CYCLE978	C (M)	공작물 좌표계에서 날의 위치를 측정합니다.		+		PM-MC
CYCLE979	C (M)	평면에서 중심을 결정하고 원호 세그먼트의 반경을 측정합니다.		+		PM-MC
CYCLE982	C (M)	공구 프로브를 조정하고 선삭 드릴링 및 밀링 공구를 측정합니다(선삭만 해당).		+		PM-MC
CYCLE994	C (M)	2개 지점 측정으로 선택한 측정 축에서 공작물 원점을 결정합니다(선삭만 해당).		+		PM-MC
CYCLE995	C (M)	공작 기계에서 스핀들의 각도를 측정합니다.		+		PM-MC
CYCLE996	C (M)	로터리 축을 이용한 좌표계 변환을 위해 변환 관련 데이터를 결정합니다.		+		PM-MC
CYCLE997	C (M)	볼의 중심과 직경을 결정하고 분산된 3개의 볼의 중심을 측정합니다.		+		PM-MC
CYCLE998	C (M)	공작물 좌표계에서 작업 평면을 기준으로 표면(평면)의 각도 위치를 결정하고 날 각도를 결정합니다.		+		PM-MC
CYCLE4071	C (T)	반전 지점에서 절입을 통한 수직 연삭		+		PM-NC
CYCLE4072	C (T)	반전 지점 및 취소 신호에서 절입을 통한 수직 연삭		+		PM-NC
CYCLE4073	C (T)	연속 절입을 통한 수직 연삭		+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
CYCLE4074	C (T)	연속 절입 및 취소 신호를 통한 수직 연삭		+		PM-NC
CYCLE4075	C (T)	반전 지점에서 절입을 통한 표면 연삭		+		PM-NC
CYCLE4077	C (T)	반전 지점 및 취소 신호에서 절입을 통한 표면 연삭		+		PM-NC
CYCLE4078	C (T)	연속 절입을 통한 표면 연삭		+		PM-NC
CYCLE4079	C (T)	간헐적 절입을 통한 표면 연삭		+		PM-NC
CYCLE9960	C (M)	좌표계를 완전히 측정		+		PM-MC

작업 D ... F

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1][2][3][4][5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
D	A	공구 옵셋 번호		+		PM-NC
D0	A	D0로 공구 옵셋 비활성화		+		PM-NC
DAC	K	절대치 넌모달 직경 프로그램	s	+		PM-NC
DC	K	로터리 축의 절대 치수. 위치에 직접 접근	s	+		PM-NC
DCI	K	데이터 클래스 I (= 개인) 지정 (SINUMERIK 828D 전용)		+		PM-NC
DCM	K	데이터 클래스 M (= 제조업체) 지정 (SINUMERIK 828D 전용)		+		PM-NC
DCU	K	데이터 클래스 U (= 사용자) 지정 (SINUMERIK 828D 전용)		+		PM-NC
DEF	K	변수 정의		+		PM-NC
DEFAULT	K	CASE 분기에서 분기		+		PM-NC
DEFINE	K	매크로 정의 키워드		+		PM-NC
DELAYFSTOF	P	지연 섹션 정지 끝 지정	m	+	-	PM-NC
DELAYFSTON	P	지연 섹션 정지 시작 지정	m	+	-	PM-NC
DELDL	F	추가 옵셋 삭제		+	-	PM-NC
DELDTG	P	이동할 거리 삭제		-	+	FM-SA

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1][2][3][4][5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
DELETE	P	지정된 파일을 삭제 경로 및 파일 식별자로 파일 이름 지정		+	-	PM-NC
DELMLOWNER	F	공구 소유자 매거진 포켓 삭제		+	-	FM-TM
DELMLRES	F	매거진 포켓 예약 삭제		+	-	FM-TM
DELMT	P	멀티공구 삭제		+	-	FM-TM
DELOBJ	F	좌표계 체인, 보호 영역, 보호 영역 요소, 충돌 페어 및 변환 데이터에서 요소 삭제		+		PM-NC
DELT	P	공구 삭제		+	-	FM-TM
DELTC	P	공구 캐리어 데이터 레코드 삭제		+	-	FM-TM
DELTOOLENV	F	공구 환경 설명 데이터 기록 삭제		+	-	PM-NC
DIACYCOFA	K	축 관련 모달 직경 프로그래밍: 싸이클에서 OFF	m	+		FM-A
DIAM90	G	G90의 경우 직경 프로그램, G91의 경우 반경 프로그램	m	+		PM-NC
DIAM90A	K	G90 및 AC의 경우 직경 프로그램, G91 및 IC의 경우 반경 프로그램	m	+		PM-NC
DIAMCHAN	K	MD 축 기능에서 직경 프로그래밍 채널 상태로 모든 축 전송		+		PM-NC
DIAMCHANA	K	직경 프로그래밍 채널 상태 전송		+		PM-NC
DIAMCYCOF	G	채널 관련 직경 프로그래밍: 싸이클에서 OFF	m	+		FM-A
DIAMOF[6]	G	직경 프로그래밍: OFF 정상 위치. 장비 제조업체 사양 참조	m	+		PM-NC
DIAMOFA	K	축 관련 모달 직경 프로그래밍: OFF 정상 위치. 장비 제조업체 사양 참조	m	+		PM-NC
DIAMON	G	직경 프로그래밍: ON	m	+		PM-NC
DIAMONA	K	축 관련 모달 직경 프로그래밍: ON 활성화. 장비 제조업체 사양 참조	m	+		PM-NC
DIC	K	상대 넌모달 축 관련 직경 프로그래밍	s	+		PM-NC
DILF	A	후퇴 경로 (길이)	m	+		PM-NC
DISABLE	P	인터럽트 OFF		+	-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1][2][3][4][5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
DISC	A	트렌지션 원 오버슈트 공구 반경 보정	m	+		PM-NC
DISCL	A	고속 절입 모션의 종점과 가공 평면 사이의 공차		+		PM-NC
DISPLOF	PA	현재 블록 디스플레이 억제		+		PM-NC
DISPLON	PA	현재 블록 디스플레이의 억제 취소		+		PM-NC
DISPR	A	리포지셔닝 경로 차이	s	+		PM-NC
DISR	A	리포지셔닝 거리	s	+		PM-NC
DISRP	A	스무딩 접근과 후퇴 동안 후퇴 평면과 가공 평면 사이의 거리		+		PM-NC
DITE	A	나사 런아웃 경로	m	+		PM-NC
DITS	A	나사 런인 경로	m	+		PM-NC
DIV	K	정수 분할		+		PM-NC
DL	A	위치별 추가 공구 옵셋 (DL, 전체 셋업 옵셋) 선택	m	+		PM-NC
DO	K	동기 동작: 조건을 충족하면 동작 트리거		-	+	FM-SA
DRFOF	P	핸드휠 옵셋 (DRF) 해제	m	+	-	PM-NC
DRIVE	G	속도 의존 가감속	m	+		PM-NC
DRIVEA	P	프로그램된 축의 니(knee) 형태 가속 특성 활성화		+	-	PM-NC
DRVPRD	P	드라이브 파라미터 읽기		+	-	PM-NC
DRVPWR	P	드라이브 파라미터 쓰기		+	-	PM-NC
DYNFINISH	G	정삭에 대한 동적 응답	m	+		PM-NC
DYNNORM[6]	G	표준 동적 응답	m	+		PM-NC
DYNPOS	G	포지셔닝 모드에 대한 동적 응답. 태핑	m	+		PM-NC
DYNPREC	G	스무딩 정삭에 대한 동적 응답	m	+		PM-NC
DYNROUGH	G	황삭에 대한 동적 응답	m	+		PM-NC
DYNSEMIFIN	G	준정삭에 대한 동적 응답	m	+		PM-NC
DZERO	P	TO 유닛의 D 번호를 모두 유효하지 않은 것으로 표시		+	-	PM-NC
EAUTO	G	마지막 세 점을 사용한 마지막 스플라인 섹션 정의	m	+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1][2][3][4][5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
EGDEF	P	전자 기어 정의		+	-	PM-NC
EGDEL	P	추종 축에 대한 커플링 정의 삭제		+	-	PM-NC
EGOFC	P	지속적으로 전자 기어 OFF		+	-	PM-NC
EGOFS	P	선택적으로 전자 기어 OFF		+	-	PM-NC
EGON	P	전자 기어 ON		+	-	PM-NC
EGONSYN	P	전자 기어 ON		+	-	PM-NC
EGONSYNE	P	전자 기어 ON. 접근 모드 지정		+	-	PM-NC
ELSE	K	NC 프로그램: IF 조건을 충족하지 않으면 프로그램 분기		+	-	PM-NC
ELSE	K	동기 동작: 조건을 충족하지 않으면 동작 트리거		-	+	FM-SA
ENABLE	P	인터럽트 ON		+	-	PM-NC
ENAT [6]	G	다음 이송 블록으로 자연적 트렌지션	m	+		PM-NC
ENDFOR	K	FOR 카운터 루프의 끝 라인		+		PM-NC
ENDIF	K	IF 분기의 끝 라인		+		PM-NC
ENDLABEL	K	REPEAT를 사용한 가공 프로그램 반복 종료 라벨		+		PM-NC, FM-B
ENDLOOP	K	무한 프로그램 루프 (LOOP) 의 종료 라인 지정		+		PM-NC
ENDPROC	K	PROC로 시작된 프로그램의 끝 라인.		+		
ENDWHILE	K	WHILE 루프의 끝 라인		+		PM-NC
ESRR	P	드라이브에서 드라이브 자율 ESR 후퇴 파라미터화		+		PM-NC
ESRS	P	드라이브에서 드라이브 자율 ESR 차단 파라미터화		+		PM-NC
ETAN	G	스플라인 시작에서 다음 이송 블록으로 접선 트렌지션	m	+		PM-NC
EVERY	K	조건이 FALSE에서 TRUE로 변하면 동기 동작 실행		-	+	FM-SA
EX	K	지수 표시에서 값 지정을 위한 키워드		+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1][2][3][4][5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
EXECSTRING	P	가공 프로그램 라인 실행에서 문자 변수 전송		+	-	PM-NC
EXECTAB	P	모션 테이블의 요소 실행		+	-	PM-NC
EXECUTE	P	프로그램 실행 ON		+	-	PM-NC
EXP	F	지수 함수 ex		+	+	PM-NC
EXTCALL	A	외부 서브프로그램 실행		+	+	PM-NC
EXTCLOSE	P	쓰기를 위해 열었던 외부 장치 / 파일 종료		+	-	PM-NC
EXTERN	K	파라미터 전송을 통한 서브프로그램 선언		+		PM-NC
EXTOPEN	P	쓰기를 위해 채널의 외부 장치 / 파일 열기		+	-	PM-NC
F	A	이송 속도 값 (G4와 연계하여 F로 드웰 시간도 프로그래밍)		+	+	PM-NC
FA	K	축 이송 속도	m	+	+	PM-NC
FAD	A	소프트 접근 및 후퇴를 위한 절입 속도		+		PM-NC
FALSE	K	논리 상수: 잘못됨		+	+	PM-NC
FB	A	넌모달 이송 속도		+		PM-NC
FCTDEF	P	폴리노미얼 기능 정의		+	-	PM-NC
FCUB	G	큐빅 스플라인 형 이송	m	+		PM-NC
FD	A	핸드휠 오버라이드의 경로 이송 속도	s	+		PM-NC
FDA	K	핸드휠 오버라이드의 축 이송 속도	s	+		PM-NC
FENDNORM[6]	G	코너 감속 OFF	m	+		PM-NC
FFWOF[6]	G	피드포워드 제어 OFF	m	+		PM-NC
FFWON	G	피드포워드 제어 ON	m	+		PM-NC
FGREF	K	로터리 축의 기준 반경 또는 오리엔테이션 축의 경로 기준 팩터 (벡터 보간)	m	+		PM-NC
FGROUP	P	축의 피드 그룹 정의		+	-	PM-NC
FI	K	프레임 데이터에 액세스하기 위한 파라미터: 미세 옵셋		+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1][2][3][4][5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
FIFOCTRL	G	전처리 버퍼 제어	m	+		PM-NC
FILEDATE	P	파일의 최종 수정 날짜 보고		+	-	PM-NC
FILEINFO	P	FILEDATE, FILESIZE, FILESTAT 및 FILETIME을 열거한 요약 정보 보고		+	-	PM-NC
FILESIZE	P	현재 파일 크기 보고		+	-	PM-NC
FILESTAT	P	파일의 읽기, 쓰기, 실행, 디스플레이, 삭제 (rwxsd) 권한 상태 보고		+	-	PM-NC
FILETIME	P	파일의 최종 수정 시간 보고		+	-	PM-NC
FINEA	K	'미세 정위치 정지'에 도달할 때 모션 종료	m	+		PM-NC
FL	K	동기 축의 한계 속도	m	+		PM-NC
FLIM	A	최대 경로 속도 조정	m	+	-	PM-NC
FLIN	G	직선형 이송	m	+		PM-NC
FMA	K	축 다중 이송 속도	m	+		PM-NC
FNORM [6]	G	DIN 66025에 따른 일반 이송 속도	m	+		PM-NC
FOC	K	넌모달 토크/힘 제한	s	-	+	FM-SA
FOCOF	K	모달 토크/힘 제한 해제	m	-	+	FM-SA
FOCON	K	모달 토크/힘 제한 설정	m	-	+	FM-SA
FOR	K	고정 실행 횟수 카운터 루프		+		PM-NC
FP	A	고정 정지점: 접근할 고정 정지점의 수	s	+		PM-NC
FPO	K	폴리노미얼을 통해 프로그래밍된 이송 속도 특성		+		PM-NC
FPR	P	로터리 축 이름		+	-	PM-NC
FPRAOF	P	회전 당 이송 속도 OFF		+	-	PM-NC
FPRAON	P	회전 당 이송 속도 ON		+	-	PM-NC
FRAME	K	좌표계 정의를 위한 데이터 유형		+		PM-NC
FRC	A	반경 및 모따기를 위한 이송 속도	s	+		PM-NC
FRCM	A	반경 및 모따기를 위한 이송 속도, 모달	m	+		PM-NC
FROM	K	일단 조건이 충족되고 동기 동작이 활성 상태인 경우 동작이 실행됨		-	+	FM-SA
FTOC	P	미세 공구 옵셋 변경		-	+	FM-SA

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1][2][3][4][5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
FTOCOF[6]	G	온라인 미세 공구 옵셋 OFF	m	+		PM-NC
FTOCON	G	온라인 미세 공구 옵셋 ON	m	+		PM-NC
FXS	K	고정 정지점으로 이동 ON	m	+	+	PM-NC
FXST	K	고정 정지점으로 이동을 위한 토크 한계	m	+	+	PM-NC
FXSW	K	고정 정지점으로 이동을 위한 모니터링 창		+	+	PM-NC
FZ	K	절삭날 이송 속도	m	+		PM-NC

작업 G ... L

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
G0	G	급 이송 직선 보간 (급 이송 모션)	m	+		PM-NC
G1[6]	G	절삭 이송 직선 보간 (직선 보간)	m	+		PM-NC
G2	G	CW 방향 원호 보간	m	+		PM-NC
G3	G	CCW 방향 원호 보간	m	+		PM-NC
G4	G	드웰 시간. 프리셋	s	+		PM-NC
G5	G	사면 플런지 연삭	s	+		PM-NC
G7	G	사면 플런지 연삭 동안의 보상 동작	s	+		PM-NC
G9	G	정위치 정지 - 감속	s	+		PM-NC
G17[6]	G	X/Y 작업 평면 선택	m	+		PM-NC
G18	G	X/Z 작업 평면 선택	m	+		PM-NC
G19	G	Y/Z 작업 평면 선택	m	+		PM-NC
G25	G	하위 작업 영역 제한	s	+		PM-NC
G26	G	상위 작업 영역 제한	s	+		PM-NC
G33	G	리드 일정 나사 절삭	m	+		PM-NC
G34	G	리드 증가 나사 절삭	m	+		PM-NC
G35	G	리드 감소 나사 절삭	m	+		PM-NC
G40[6]	G	공구 반경 보정 OFF	m	+		PM-NC

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
G41	G	공구 반경 좌측 보정	m	+		PM-NC
G42	G	공구 반경 우측 보정	m	+		PM-NC
G53	G	현재 제로 옵셋의 억제 (비모달)	s	+		PM-NC
G54	G	1번째 설정 가능한 제로 옵셋	m	+		PM-NC
G55	G	2번째 설정 가능한 제로 옵셋	m	+		PM-NC
G56	G	3번째 설정 가능한 제로 옵셋	m	+		PM-NC
G57	G	4번째 설정 가능한 제로 옵셋	m	+		PM-NC
G58 (840D sl)	G	프로그래밍 가능한 절대 워크 옵셋(일반 옵셋)	s	+		PM-NC
G58 (828D)	G	5번째 설정 가능한 제로 옵셋	m	+		PM-NC
G59 (840D sl)	G	프로그래밍 가능한 추가 워크 옵셋(미세 옵셋)	s	+		PM-NC
G59 (828D)	G	6번째 설정 가능한 제로 옵셋	m	+		PM-NC
G60 [6]	G	정위치 정지 - 감속	m	+		PM-NC
G62	G	공구 반경 옵셋 활성 시 (G41, G42) 내부 코너에서 코너 감속	m	+		PM-NC
G63	G	태핑 척을 사용한 태핑	s	+		PM-NC
G64	G	연속 경로 모드	m	+		PM-NC
G70	G	인치계 치수 (길이)	m	+	+	PM-NC
G71 [6]	G	미터계 치수 (길이)	m	+	+	PM-NC
G74	G	원점 복귀	s	+		PM-NC
G75	G	고정 정지점 접근	s	+		PM-NC
G90 [6]	G	절대 치수	m/s	+		PM-NC
G91	G	증분 치수	m/s	+		PM-NC
G93	G	역 시간 이송 속도 rpm	m	+		PM-NC
G94 [6]	G	직선 이송 속도 F (mm/min 또는 inch/min 및 °/min)	m	+		PM-NC
G95	G	회전 당 이송 속도 F (mm/rev 또는 inch/rev)	m	+		PM-NC
G96	G	회전 당 이송 속도(G95 관련) 및 주속 일정 제어	m	+		PM-NC

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
G97	G	회전 당 이송 속도 및 일정 스핀들 속도 (주속 일정 제어 OFF)	m	+		PM-NC
G110	G	최종 프로그램 설정점 위치에 상대적인 극점 프로그래밍	s	+		PM-NC
G111	G	현재 공작물 좌표계의 원점에 상대적인 극점 프로그래밍	s	+		PM-NC
G112	G	최종 유효 극점에 상대적인 극점 프로그래밍	s	+		PM-NC
G140 [6]	G	G41/G42가 정의한 SAR 접근 방향	m	+		PM-NC
G141	G	형상 좌측으로의 SAR 접근 방향	m	+		PM-NC
G142	G	형상 우측으로의 SAR 접근 방향	m	+		PM-NC
G143	G	접선 의존 SAR 접근 방향	m	+		PM-NC
G147	G	직선으로 소프트 접근	s	+		PM-NC
G148	G	직선으로 소프트 후퇴	s	+		PM-NC
G153	G	베이직 프레임을 포함한 현재 프레임 억제	s	+		PM-NC
G247	G	4분원으로 소프트 접근	s	+		PM-NC
G248	G	4분원으로 소프트 후퇴	s	+		PM-NC
G290 [6]	G	SINUMERIK 모드로 전환 ON	m	+		FM-TM
G291	G	ISO2/3 모드로 전환 ON	m	+		FM-TM
G331	G	리지드 태핑, 양의 리드, CW 방향	m	+		PM-NC
G332	G	리지드 태핑, 음의 리드, CCW 방향	m	+		PM-NC
G335	G	CW 방향으로 볼록 나사 선삭	m	+		PM-NC
G336	G	CCW 방향으로 볼록 나사 선삭	m	+		PM-NC
G340 [6]	G	입체 접근 (X, Y, Z (헬릭스))	m	+		PM-NC
G341	G	수직 절입 (Z) 후 평면 절입 (X,Y)	m	+		PM-NC
G347	G	반원으로 소프트 접근	s	+		PM-NC
G348	G	반원으로 소프트 후퇴	s	+		PM-NC
G450 [6]	G	트렌지션 원	m	+		PM-NC
G451	G	등거리의 교차	m	+		PM-NC
G460 [6]	G	접근 및 후퇴 블록의 충돌 탐지 활성화	m	+		PM-NC

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
G461	G	원호를 TRC 블록에 삽입	m	+		PM-NC
G462	G	직선을 TRC 블록에 삽입	m	+		PM-NC
G500 [6]	G	전체 설정 가능한 프레임 해제, 베이직 프레임은 활성 상태	m	+		PM-NC
G505 ... G599	G	5 ... 99 설정 가능한 워크 옵셋	m	+		PM-NC
G601 [6]	G	미세 정위치 정지에서 블록 변경	m	+		PM-NC
G602	G	일반 정위치 정지에서 블록 변경	m	+		PM-NC
G603	G	IPO 블록 끝에서 블록 변경	m	+		PM-NC
G621	G	모든 코너에서 코너 감속	m	+		PM-NC
G641	G	거리 기준에 따라 스무딩하는 연속 경로 모드(= 프로그래밍 라운딩 공차)	m	+		PM-NC
G642	G	정해진 공차 안에서 스무딩하는 연속 경로 모드	m	+		PM-NC
G643	G	정해진 공차 안에서 스무딩하는 연속 경로 모드(블록 내경)	m	+		PM-NC
G644	G	동적 응답을 가능 최대치로 하여 스무딩하는 연속 경로 모드	m	+		PM-NC
G645	G	정해진 공차 안에서 스무딩 및 접선 블록 트렌지션을 포함하는 연속 이송 모드	m	+		PM-NC
G700	G	인치계 치수 (길이, 이송 속도)	m	+	+	PM-NC
G710 [6]	G	미터계 치수 (길이, 이송 속도)	m	+	+	PM-NC
G810 [6], ..., G819	G	OEM 사용자를 위한 G 그룹		+		PM-NC
G820 [6], ..., G829	G	OEM 사용자를 위한 G 그룹		+		PM-NC
G931	G	이동 시간을 통해 지정된 이송 속도, 일정 경로 속도 해제	m	+		
G942	G	주속 일정 제어 포함 주축 속도 및 직선 이송 동결	m	+		
G952	G	회전 당 이송 속도 포함 주속 일정 제어 동결	m	+		
G961	G	직선 이송 속도(G94 관련) 및 주속 일정 제어	m	+		PM-NC

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]**에 대한 설명은 범례 (페이지 1403)를 참조하십시오.**						
G962	G	직선 또는 회전 당 이송 속도 및 주속 일정 제어	m	+		PM-NC
G971	G	직선 이송 속도 및 일정 스핀들 속도(주속 일정 제어 OFF)	m	+		PM-NC
G972	G	직선 이송 속도 또는 회전 당 이송 속독 및 일정 스핀들 속도(주속 일정 제어 OFF)	m	+		PM-NC
G973	G	스핀들 속도 제한 없는 회전 당 이송 속도 및 일정 스핀들 속도(ISO 모드를 위한 LIMS 없는 G97)	m	+		PM-NC
GEOAX	P	새로운 채널 축을 기하 축 1 - 3 으로 할당		+	-	PM-NC
GET	P	채널 간 활성 축 교체		+	+	PM-NC
GETACTT	F	그룹에서 이름이 동일한 공구 가져오기		+	-	FM-TM
GETACTTD	F	절대 D 번호와 연결된 T 번호 가져오기		+	-	PM-NC
GETD	P	채널 간의 직접 축 교체		+	-	PM-NC
GETDNO	F	공구 (T) 절삭날 (CE) 의 D 번호 보고		+	-	PM-NC
GETEXET	P	로드된 T 번호 읽기		+	-	FM-TM
GETFREELOC	P	매거진에서 특정 공구에 사용 가능한 포켓 찾기		+	-	FM-TM
GETSELT	P	선택한 T 번호 보고		+	-	FM-TM
GETT	F	공구 이름의 T 번호 가져오기		+	-	FM-TM
GETTCOR	F	공구 길이 및/또는 공구 길이 구성요소 읽기		+	-	PM-NC
GETTENV	F	T, D 및 DL 번호 읽기		+	-	PM-NC
GETVARAP	F	시스템/사용자 변수에 대한 사용 권한 읽기		+	-	PM-NC
GETVARDFT	F	시스템/사용자 변수의 기본 값 읽기		+	-	PM-NC
GETVARLIM	F	시스템/사용자 변수의 한계값 읽기		+	-	PM-NC
GETVARPHU	F	시스템/사용자 변수의 물리 단위 읽기		+	-	PM-NC
GETVARTYP	F	시스템/사용자 변수의 데이터 유형 읽기		+	-	PM-NC

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]**에 대한 설명은 범례 (페이지 1403)를 참조하십시오.**						
GFRAME0 ... GFRAME100	G	채널에서 데이터 관리의 연삭 프레임 <n>의 활성화	m	+		PM-NC
GOTO	K	전방으로 점프 후 후방으로 점프 (처음에는 프로그램 종료 방향 그 다음 프로그램 시작 방향)		+		PM-NC
GOTOB	K	후방 점프 (프로그램 시작 방향으로)		+		PM-NC
GOTOC	K	GOTO과 유사하나, 알람 14080 "점프 대상을 찾을 수 없음"을 억제		+		PM-NC
GOTOF	K	전방 점프 (프로그램 종료 방향으로)		+		PM-NC
GOTOS	K	프로그램 시작으로 점프		+		PM-NC
GP	K	위치 특성의 간접 프로그래밍 키워드		+		PM-NC
GROUP_ ADDEND	C (T)	시험 절삭 추가 종료		+		PM-NC
GROUP_BEGIN	C (T)	프로그램 그룹 시작		+		PM-NC
GROUP_END	C (T)	프로그램 그룹 종료		+		PM-NC
GWPSOF	P	연삭 휠 원주 속도 일정 제어 (GWPS) 선택 취소	s	+	-	PM-NC
GWPSON	P	연삭 휠 원주 속도 일정 제어 (GWPS) 선택	s	+	-	PM-NC
H...	A	PLC로 보조 기능 출력		+	+	PM-NC, FM-B
HOLES1	C (T)	여러 개의 홀로 이루어진 행		+		PM-NC
HOLES2	C (T)	여러 개의 홀로 이루어진 원호		+		PM-NC
I	A	보간 파라미터	s	+		PM-NC
I1	A	중간점 좌표	s	+		PM-NC
IC	K	증분 치수	s	+		PM-NC
ICYCOF	P	ICYCOF 이후 고정 싸이클의 전체 블록은 하나의 보간 싸이클로 처리됩니다.		+	+	FM-SA
ICYCON	P	ICYCON 이후 고정 싸이클의 개별 블록은 별도의 보간 싸이클로 처리됩니다.		+	+	FM-SA
ID	K	모달 동기 동작의 식별자	m	-	+	FM-SA
IDS	K	모달 정적 동기 동작의 식별자		-	+	FM-SA

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
IF	K	가공 프로그램/공정 싸이클에서 조건 점프		+	+	PM-NC
INDEX	F	입력 문자열에서 문자 인덱스 정의		+	-	PM-NC
INICF	K	NEWCONF의 변수 초기화		+		PM-NC
INIPO	K	전원을 켤 때 변수 초기화		+		PM-NC
INIRE	K	리셋할 때 변수 초기화		+		PM-NC
INIT	P	특정 채널에서 실행할 특정 NC 프로그램 선택		+	-	PM-NC
INITIAL		모든 영역에서 INI 파일 생성		+		PM-NC
INT	K	데이터 유형: 부호가 있는 정수		+		PM-NC
INTERSEC	F	두 형상 요소 간의 교차 계산		+	-	PM-NC
INVCCW	G	인벌류트 트레이스, CCW 방향	m	+		PM-NC
INVCW	G	인벌류트 트레이스, CW 방향	m	+		PM-NC
INVFRAME	F	프레임에서 역 프레임 계산		+	-	FM-B
IP	K	변수 보간 파라미터		+		PM-NC
IPOBRKA	P	정지 동작 활성화의 모션 기준	m	+	+	
IPOENDA	K	"IPO 정지"에 도달할 때 모션 종료	m	+		PM-NC
IPTRLOCK	P	다음 기계 기능 블록에서 추적 불가능한 프로그램 절의 시작 동결.	m	+	-	PM-NC
IPTRUNLOCK	P	중단할 경우 현재 블록에서 추적 불가능한 프로그램 절의 끝 설정.	m	+	-	PM-NC
IR	A	볼록 나선 선삭 시 원호 가운데 좌표 (X 축)		+		PM-NC
ISAXIS	F	기하 축 1이 파라미터로 지정되었는지 확인		+	-	PM-NC
ISD	A	삽입 깊이	m	+		PM-NC
ISFILE	F	파일이 NC 애플리케이션 메모리에 있는지 점검		+	-	PM-NC
ISNUMBER	F	입력 문자열을 숫자로 전환 가능한지 점검		+	-	PM-NC

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
ISOCALL	K	ISO 언어로 프로그래밍된 프로그램의 간접 호출		+		PM-NC
ISVAR	F	전송 파라미터가 NC에서 선언된 변수를 포함하는지 점검		+	-	PM-NC
J	A	보간 파라미터	s	+		PM-NC
J1	A	중간점 좌표	s	+		PM-NC
JERKA	P	프로그래밍된 축의 MD를 통한 가속 반응 활성화		+	-	
JERKLIM	K	최대 축 저크 조정	m	+		PM-NC
JERKLIMA	K	최대 슬레이브 축 저크의 감소 또는 오버슈트	m	+	+	PM-NC
JR	A	볼록 나선 선삭 시 원호 가운데 좌표 (Y 축)		+		PM-NC
K	A	보간 파라미터	s	+		PM-NC
K1	A	중간점 좌표	s	+		PM-NC
KONT	G	공구 옵셋에서 형상 주위로 이동	m	+		PM-NC
KONTC	G	연속 커브 폴리노미얼로 접근/후퇴	m	+		PM-NC
KONTT	G	연속 접선 폴리노미얼로 접근/후퇴	m	+		PM-NC
KR	A	볼록 나선 선삭 시 원호 가운데 좌표 (Z 축)		+		PM-NC
L	A	서브프로그램 번호	s	+	+	PM-NC
LEAD	A	리드 각도 1차 기본 공구 방향 2차 방향 폴리노미얼	m	+		PM-NC
LEADOF	P	축 관련 마스터 값 커플링 OFF		+	+	PM-NC
LEADON	P	축 관련 마스터 값 커플링 ON		+	+	PM-NC
LENTOAX	F	활성 공구의 공구 길이 L1, L2 및 L3을 가로축, 세로축 및 직교축에 지정하는 데에 대한 정보를 제공		+	-	PM-NC
LFOF [6]	G	나사 절삭을 위한 고속 후퇴 OFF	m	+		PM-NC
LFON	G	나사 절삭을 위한 고속 후퇴 ON	m	+		PM-NC

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
LFPOS	G	POLF를 사용하여 프로그래밍된 절대 축 위치로 POLFMASK 또는 POLFMLIN을 사용하여 선언된 축의 후퇴	m	+		PM-NC
LFTXT [6]	G	고속 후퇴를 위한 후퇴 이동 평면은 경로 접선 및 현재 공구 방향에서 결정됨	m	+		PM-NC
LFWP	G	고속 후퇴를 위한 후퇴 이동 평면은 현재 작업 평면(G17/G18/G19)에 의해 결정됨	m	+		PM-NC
LIFTFAST	K	고속 후퇴		+		PM-NC
LIMS	K	G96/G961 및 G97에 대한 속도 제한	m	+		PM-NC
LLI	K	변수의 하한값		+		PM-NC
LN	F	자연 로그		+	+	PM-NC
LOCK	P	ID로 동기 동작 비활성화 (고정 싸이클 정지)		-	+	FM-SA
LONGHOLE	C (T)	연장된 홀		+		PM-NC
LOOP	K	무한 루프 개시		+		PM-NC

작업 M ... R

작업	종류[1]	의미	W [2]	TP[3]	SA[4]	설명 참조 [5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
M0		프로그램 정지		+	+	PM-NC
M1		선택적 정지		+	+	PM-NC
M2		프로그램 끝, 메인 프로그램 (M30)		+	+	PM-NC
M3		CW 방향 스핀들 회전		+	+	PM-NC
M4		CCW 방향 스핀들 회전		+	+	PM-NC
M5		스핀들 정지		+	+	PM-NC
M6		공구 교체		+	+	PM-NC
M17		서브프로그램 종료		+	+	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
M19		SD43240에 입력된 위치로 스핀들 포지셔닝		+	+	PM-NC
M30		프로그램 끝, 메인 프로그램 (M2)		+	+	PM-NC
M40		자동 기어 변경		+	+	PM-NC
M41 ... M45		기어 단수 1 ... 5		+	+	PM-NC
M70		축 모드로 전환		+	+	PM-NC
MASLDEF	P	마스터/슬레이브 축 그룹핑 정의		+	+	PM-NC
MASLDEL	P	마스터/슬레이브 축 그룹핑 해제 및 그룹핑 정의 제거		+	+	PM-NC
MASLOF	P	임시 커플링의 해제		+	+	PM-NC
MASLOFS	P	자동 커플링의 해제 축 정지로 임시 커플링의 해제		+	+	PM-NC
MASLON	P	임시 커플링의 동작		+	+	PM-NC
MATCH	F	문자열에서 문자열 검색		+	-	PM-NC
MAXVAL	F	두 변수 중 높은 값 (산술 함수)		+	+	PM-NC
MCALL	K	모달 서브프로그램 호출		+		PM-NC
MEAC	K	이동할 거리를 삭제하지 않고 연속 축 측정	s	+	+	PM-NC
MEAFRAME	F	측정 지점에서 프레임 계산		+	-	PM-NC
MEAS	A	이동할 거리를 삭제하고 측정	s	+		PM-NC
MEASA	K	이동할 거리를 삭제하고 축 측정	s	+	+	PM-NC
MEASURE	F	공작물 및 공구 측정을 위한 계산 방법		+	-	FM-TE
MEAW	A	이동할 거리를 삭제하지 않고 측정	s	+		PM-NC
MEAWA	K	이동할 거리를 삭제하지 않고 축 측정	s	+	+	PM-NC
MI	K	프레임 데이터에 액세스: 미러링		+		PM-NC
MINDEX	F	입력 문자열에서 문자 인덱스 정의		+	-	PM-NC
MINVAL	F	두 변수 중 낮은 값 (산술 함수)		+	+	PM-NC
MIRROR	G	프로그래밍 미러링	s	+		PM-NC
MMC	P	HMI 상에서 가공 프로그램으로부터 대화형 프로그램 창 호출		+	-	PM-NC
MOD	K	모듈로 분할		+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
MODAXVAL	F	모듈로 로터리 축의 모듈로 위치 결정		+	-	PM-NC
MOV	K	축 포지셔닝 시작		-	+	FM-SA
MOVT	A	공구 방향에서 이송 모션의 종점 지정				FM-B
MSG	P	프로그램 가능한 메시지	m	+	-	PM-NC
MVTOOL	P	공구 이동을 위한 언어 명령		+	-	FM-TM
N	A	NC 블록 번호		+		PM-NC
NAMETOINT	F	시스템 변수 인덱스 결정		+		PM-NC
NC	K	데이터의 유효 범위 지정		+		PM-NC
NEWCONF	P	수정한 머신 데이터 적용 ('머신 데이터 활성화'에 해당)		+	-	PM-NC
NEWMT	F	새 멀티공구 생성		+	-	FM-TM
NEWT	F	새 공구 작성		+	-	FM-TM
NORM [6]	G	공구 옵셋을 사용한 시작 지점과 종점의 표준 설정	m	+		PM-NC
NOT	K	논리적 NOT (부정)		+		PM-NC
NPROT	P	기계 특정 보호 영역 ON/OFF		+	-	PM-NC
NPROTDEF	P	기계 특정 보호 영역 정의		+	-	PM-NC
NUMBER	F	입력 문자열을 숫자로 전환		+	-	PM-NC
OEMIPO1	G	OEM 보간 1	m	+		PM-NC
OEMIPO2	G	OEM 보간 2	m	+		PM-NC
OF	K	CASE 분기의 키워드		+		PM-NC
OFFN	A	프로그램된 형상의 허용치	m	+		PM-NC
OMA1	A	OEM 주소 1	m	+		PM-NC
OMA2	A	OEM 주소 2	m	+		PM-NC
OMA3	A	OEM 주소 3	m	+		PM-NC
OMA4	A	OEM 주소 4	m	+		PM-NC
OMA5	A	OEM 주소 5	m	+		PM-NC
OR	K	로직 연산자, OR 연산		+		PM-NC
ORIAXES	G	기계 축 또는 오리엔테이션 축의 직선 보간	m	+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
ORIAXPOS	G	로터리 축 위치의 가상 오리엔테이션 축을 통한 오리엔테이션 각도	m	+		PM-NC
ORIC [6]	G	외부 코너에서의 오리엔테이션 변경은 삽입될 원호 블록과 겹쳐집니다	m	+		PM-NC
ORICONCCW	G	CCW 방향으로 원주면 보간	m	+		PM-NC, FM-TR
ORICONCW	G	CW 방향으로 원주면 보간	m	+		PM-NC, FM-TR
ORICONIO	G	중간 오리엔테이션 설정으로 원주면 보간	m	+		PM-NC, FM-TR
ORICONTO	G	접선 트렌지션에서 원주면 보간 (최종 오리엔테이션)	m	+		PM-NC, FM-TR
ORICURVE	G	공구 두 접촉 지점의 모션 지정으로 오리엔테이션 보간	m	+		PM-NC, FM-TR
ORID	G	오리엔테이션 변경은 원호 블록 이전에 수행됩니다	m	+		PM-NC
ORIEULER [6]	G	오일러 각을 통한 오리엔테이션 각	m	+		PM-NC
ORIMKS	G	기계 좌표계 공구 오리엔테이션	m	+		PM-NC
ORIPATH	G	경로 관련 공구 오리엔테이션	m	+		PM-NC
ORIPATHS	G	경로와 관련된 공구 오리엔테이션, 오리엔테이션 특징의 소음이 감소됩니다.	m	+		PM-NC
ORIPLANE	G	평면에서 보간 (ORIVECT 에 일치), 대형 반경 원호 보간	m	+		PM-NC
ORIRESET	P	최대 3개의 오리엔테이션 축이 있는 초기 공구 오리엔테이션		+	-	PM-NC
ORIROTA [6]	G	절대 회전 방향 회전 각도	m	+		PM-NC
ORIROTC	G	경로 접선과 관계된 접선 회전 벡터	m	+		PM-NC
ORIROTR	G	시작과 종료 오리엔테이션 사이의 평면에 관련한 회전 각	m	+		PM-NC
ORIROTT	G	오리엔테이션 벡터의 변경에 관련한 회전 각	m	+		PM-NC
ORIRPY	G	RPY각(XYZ) 을 통한 오리엔테이션 각	m	+		PM-NC
ORIRPY2	G	RPY각 (ZYX) 을 통한 오리엔테이션 각	m	+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]**에 대한 설명은 범례 (페이지 1403)를 참조하십시오.**						
ORIS	A	오리엔테이션 변경	m	+		PM-NC
ORISOF [6]	G	오리엔테이션 특성 스무딩 OFF	m	+		PM-NC
ORISOLH	F	오리엔테이션 계산		+		PM-NC
ORISON	G	오리엔테이션 특성 스무딩 ON	m	+		PM-NC
ORIVECT [6]	G	대형 원호 보간(ORIPLANE과 동일)	m	+		PM-NC
ORIVIRT1	G	가상 오리엔테이션 축을 통한 오리엔테이션 각(정의 1)	m	+		PM-NC
ORIVIRT2	G	가상 오리엔테이션 축을 통한 오리엔테이션 각(정의 1)	m	+		PM-NC
ORIWKS [6]	G	공작물 좌표계의 공구 오리엔테이션 각도	m	+		PM-NC
OS	K	오실레이션 ON/OFF		+		PM-NC
OSB	K	오실레이션: 시작점	m	+		PM-NC
OSC	G	연속적인 공구 오리엔테이션 스무딩	m	+		PM-NC
OSCILL	K	축: 1 -3 절입 축	m	+		PM-NC
OSCTRL	K	오실레이션 옵션	m	+		PM-NC
OSD	G	SD로 스무딩 거리를 지정 공구 오리엔테이션 스무딩	m	+		PM-NC
OSE	K	오실레이션 종점	m	+		PM-NC
OSNSC	K	오실레이션: 스파크아웃 싸이클 개수	m	+		PM-NC
OSOF [6]	G	공구 오리엔테이션 스무딩 OFF	m	+		PM-NC
OSP1	K	오실레이션: 좌측 반전 지점	m	+		PM-NC
OSP2	K	오실레이션 우측 반전 지점	m	+		PM-NC
OSS	G	블록 끝에서의 공구 오리엔테이션 스무딩	m	+		PM-NC
OSSE	G	블록 시작과 끝에서의 공구 오리엔테이션 스무딩	m	+		PM-NC
OST	G	SD(프로그래밍된 오리엔테이션 특성과의 최대 편차)를 사용한 각도 공차 지정으로 공구 오리엔테이션 스무딩	m	+		PM-NC
OST1	K	오실레이션: 좌측 반전 지점의 정지 지점	m	+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]**에 대한 설명은 범례 (페이지 1403)를 참조하십시오.**						
OST2	K	오실레이션: 우측 반전 지점의 정지 지점	m	+		PM-NC
OTOL	A	컴프레서 기능에 대한 오리엔테이션 공차. 오리엔테이션 스무딩 및 스무딩 종류	m	+	-	PM-NC
OTOLG0	A	급 이송 동작을 위한 오리엔테이션 공차	m	+	-	PM-NC
OVR	K	속도 옵셋	m	+		PM-NC
OVRA	K	축 속도 옵셋	m	+	+	PM-NC
OVRRAP	K	급 이송 오버라이드	m	+		PM-NC
P	A	서브프로그램 반복 횟수		+		PM-NC
PACCLIM	A	최대 경로 가속 조정	m	+	-	PM-NC
PAROT	G	공작물로 공작물 좌표계 정렬	m	+		PM-NC
PAROTOF [6]	G	공작물과 관련된 프레임 회전 해제	m	+		PM-NC
PCALL	K	절대 경로 및 파라미터 전송을 사용하여 서브프로그램 호출		+		PM-NC
PDELAYOF	G	지연 OFF 상태의 펀칭	m	+		PM-NC
PDELAYON [6]	G	지연 ON 상태의 펀칭	m	+		PM-NC
PHI	K	테이퍼의 방향 축에 대한 회전 오리엔테이션 각도		+		PM-NC
PHU	K	변수의 물리 단위		+		PM-NC
PL	A	1. B 스플라인: 노드 공차 2. 폴리노미얼 보간을 위한 파라미터 간격의 폴리노미얼 보간 길이	s	+		PM-NC
PM	K	분 당		+		PM-NC
PO	K	폴리노미얼 보간을 위한 폴리노미얼 계수	s	+		PM-NC
POCKET3	C (T)	사각 포켓 밀링		+		PM-NC
POCKET4	C (T)	원호 포켓 밀링		+		PM-NC
POLF	K	LIFTFAST 후퇴 위치	m	+		PM-NC
POLFA	P	$AA_ESR_TRIGGER을 사용하여 단일 축의 후퇴 시작 위치	m	+	+	PM-NC
POLFMASK	P	축 사이의 연결 없는 후퇴를 위한 축 활성화	m	+	-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
POLFMLIN	P	축 사이가 직선으로 연결된, 후퇴를 위한 축 활성화	m	+	-	PM-NC
POLY	G	폴리노미얼 보간	m	+		PM-NC
POLYPATH	P	AXIS 또는 VECT 축 그룹에 대해 폴리노미얼 보간 선택 가능	m	+	-	PM-NC
PON	G	펀칭 ON	m	+		PM-NC
PONS	G	보간 싸이클 내 펀치 ON	m	+		PM-NC
POS	K	축 포지셔닝		+	+	PM-NC
POSA	K	블록 경계를 가로 지르는 축 포지셔닝		+	+	PM-NC
POSM	P	매거진 포지셔닝		+	-	FM-TM
POSMT	P	포켓 번호의 공구 홀더에 멀티공구 포지셔닝		+	-	FM-TM
POSP	K	여러 위치에 축 포지셔닝 (오실레이션)		+		PM-NC
POSRANGE	F	현재 보간된 축의 위치 지령치가 미리 정의된 기준 위치 범위에 있는지 확인합니다.		+	+	FM-SA
POT	F	제곱 (산술 함수)		+	+	PM-NC
PR	K	회전 당		+		PM-NC
PREPRO	PA	준비를 통한 서브프로그램 확인		+		PM-NC
PRESETON	P	원점 복귀 상태 없이 실제값 설정		+	+	PM-NC
PRESETONS	P	원점 복귀 상태를 유지하면서 실제값 설정		+	+	PM-NC
PRIO	K	프로세스 인터럽트 순서 설정 키워드		+		PM-NC
PRLOC	K	로컬 변경 후에만 리셋할 때 변수 초기화		+		PM-NC
PROC	K	프로그램의 첫 번째 동작		+		PM-NC
PROTA	P	충돌 모델 재계산 요청		+		PM-NC
PROTD	F	두 보호 영역 사이 거리 계산		+		PM-NC
PROTS	P	보호 영역 상태 설정		+		PM-NC
PSI	K	테이퍼의 개방 각도		+		PM-NC
PTP	G	지점 간 모션(PTP 이송)	m	+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
PTPG0	G	G0만으로 지점 간 모션, 그렇지 않으면 경로 모션 CP	m	+		PM-NC
PTPWOC	G	방향 변경에 의한 보정 이동이 없는 지점 간 모션	m	+		PM-NC
PUNCHACC	P	니블링 시 이송별 가속		+	-	PM-NC
PUTFTOC	P	병렬 드레싱을 위한 공구 미세 옵셋		+	-	PM-NC
PUTFTOCF	P	FCTDEF로 정의된 병렬 드레싱에 따른 공구 미세 옵셋		+	-	PM-NC
PW	A	B 스플라인, 점 무게	s	+		PM-NC
QU	K	고속 추가 (보조) 기능 출력		+		PM-NC
R...	A	설정 가능한 주소 식별자 및 숫자로 구성된 산술 파라미터		+		PM-NC
RAC	K	넌모달 축 지정 절대 반경 프로그래밍	s	+		PM-NC
RDISABLE	P	읽기 비활성화됨		-	+	FM-SA
READ	P	지정된 파일에서 하나 또는 그 이상의 라인을 읽고 그 정보를 배열에 보관함		+	-	PM-NC
REAL	K	데이터 유형: 부호가 있는 실수		+		PM-NC
REDEF	K	시스템 변수, 사용자 변수 및 NC 언어 명령 다시 정의		+		PM-NC
RELEASE	P	축 교체를 위한 기계 축 릴리즈		+	+	PM-NC
REP	K	동일 값으로 배열의 모든 요소를 초기화하는 키워드		+		PM-NC
REPEAT	K	프로그램 루프 반복		+		PM-NC
REPEATB	K	프로그램 라인 반복		+		PM-NC
REPOSA	G	모든 축 직선 리포지셔닝	s	+		PM-NC
REPOSH	G	반원으로 리포지셔닝	s	+		PM-NC
REPOSHA	G	반원 내 모든 축, 기하 축의 리포지셔닝	s	+		PM-NC
REPOSL	G	직선 리포지셔닝	s	+		PM-NC
REPOSQ	G	4분원 리포지셔닝	s	+		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
REPOSQA	G	사분원 내 모든 축, 기하 축의 직선 리포지셔닝	s	+		PM-NC
RESETMON	P	지령치 활성화를 위한 언어 명령		+	-	FM-TM
RET	P	서브프로그램 종료		+	+	PM-NC
RETB	P	서브프로그램 종료		+	+	PM-NC
RIC	K	넌모달 축 지정 상대적 반경 프로그래밍	s	+		PM-NC
RINDEX	F	입력 문자열에서 문자 인덱스 정의		+	-	PM-NC
RMB	G	블록 시작으로 리포지셔닝	m	+		PM-NC
RMBBL	G	블록 시작으로 리포지셔닝	s	+		PM-NC
RME	G	블록 끝으로 리포지셔닝	m	+		PM-NC
RMEBL	G	블록 끝으로 리포지셔닝	s	+		PM-NC
RMI [6]	G	인터럽트 지점으로 리포지셔닝	m	+		PM-NC
RMIBL [6]	G	인터럽트 지점으로 리포지셔닝	s	+		PM-NC
RMN	G	가장 가까운 경로 지점으로 리포지셔닝	m	+		PM-NC
RMNBL	G	가장 가까운 경로 지점으로 리포지셔닝	s	+		PM-NC
RND	A	형상 코너 라운딩	s	+		PM-NC
RNDM	A	모달 라운딩	m	+		PM-NC
ROT	G	프로그래밍 회전	s	+		PM-NC
ROTS	G	입체각 프로그래밍 프레임 회전	s	+		PM-NC
ROUND	F	소수 자리 반올림		+	+	PM-NC
ROUNDUP	F	입력 값 반올림		+	+	PM-NC
RP	A	극점 반경	m/s	+		PM-NC
RPL	A	평면에서 회전	s	+		PM-NC
RT	K	프레임 데이터에 액세스하기 위한 파라미터: 회전		+		PM-NC
RTLIOF	G	직선 보간 없이 G0 이송 (단일 축 보간)	m	+		PM-NC
RTLION [6]	G	G0로 직선 보간	m	+		PM-NC

작업 S ... Z

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
S	A	스핀들 속도 (G4, G96/G961에서는 다른 의미 사용)	m/s	+/-	+/-	PM-NC
SAVE	PA	서브프로그램이 호출될 때 정보를 저장하기 위한 속성		+/-		PM-NC
SBLOF	P	싱글 블록 억제		+/-	-	PM-NC
SBLON	P	싱글 블록 억제 취소		+/-	-	PM-NC
SC	K	프레임 데이터에 액세스하기 위한 파라미터: 스케일링		+/-		PM-NC
SCALE	G	프로그래밍 배율	s	+/-		PM-NC
SCC	K	G96/G961/G962로의 가로 축의 선택적 지정 축 식별자는 기하, 채널 또는 기계 축 형태를 띨 수 있습니다.		+/-		PM-NC
SCPARA	K	서보 파라미터 세트 프로그래밍		+/-	+/-	PM-NC
SD	A	스플라인 각도	s	+/-		PM-NC
SET	K	수록된 값으로 배열의 모든 요소를 초기화하는 키워드		+/-		PM-NC
SETAL	P	알람 설정		+/-	+/-	PM-NC
SETDNO	F	공구 (T) 절삭날 (CE) 의 D 번호 지정		+/-	-	PM-NC
SETINT	K	NC 입력이 있을 때 활성화될 인터럽트 루틴을 정의		+/-		PM-NC
SETM	P	전용 채널에서 마커 설정		+/-	+/-	PM-NC
SETMS	P	머신 데이터에서 정의된 마스터 스핀들로 리셋		+/-	-	PM-NC
SETMS(n)	P	스핀들 n 을 마스터 스핀들로 설정		+/-		PM-NC
SETMTH	P	마스터 공구 홀더 번호를 설정함		+/-	-	FM-TM
SETPIECE	P	스핀들에 할당된 모든 공구에 가공 수량 설정		+/-	-	FM-TM
SETTA	P	마모 그룹에서 공구 활성화		+/-	-	FM-TM
SETTCOR	F	모든 보충 조건을 고려한 공구 콤포넌트 수정		+/-	-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
SETTIA	P	마모 그룹에서 공구 비활성화		+/-	-	FM-TM
SF	A	나사 절삭의 시작점 옵셋	m	+/-		PM-NC
SIN	F	사인 (삼각 함수)		+/-	+/-	PM-NC
SIRELAY	F	SIRELIN, SIRELOUT 및 SIRELTIME로 파라미터화된 안전 기능 활성화		-	+/-	FM-SI
SIRELIN	P	펑션 블록의 입력 변수 초기화		+/-	-	FM-SI
SIRELOUT	P	펑션 블록의 출력 변수 초기화		+/-	-	FM-SI
SIRELTIME	P	펑션 블록의 타이머 초기화		+/-	-	FM-SI
SLOT1	C (T)	일반 홈		+/-		PM-NC
SLOT2	C (T)	원호 슬롯		+/-		PM-NC
SOFT	G	소프트 경로 가속	m	+/-		PM-NC
SOFTA	P	프로그램된 축의 저크 제한 축 가속 활성화		+/-	-	PM-NC
SON	G	니블링 ON	m	+/-		PM-NC
SONS	G	보간 싸이클 내 니블링 ON	m	+/-		PM-NC
SPATH [6]	G	FGROUP 축의 경로 기준은 호 길이입니다.	m	+/-		PM-NC
SPCOF	P	스핀들을 마스터 스핀들로 또는 위치 제어에서 속도 제어로 전환	m	+/-	-	PM-NC
SPCON	P	스핀들을 마스터 스핀들로 또는 속도 제어에서 위치 제어로 전환	m	+/-	-	PM-NC
SPI	F	스핀들 번호를 축 이름으로 변환		+/-	-	PM-NC
SPIF1 [6]	G	펀치/니블링 바이트 1을 위한 빠른 NC 입력/출력	m	+/-		FM-TE
SPIF2	G	펀치/니블링 바이트 2를 위한 빠른 NC 입력/출력	m	+/-		FM-TE
SPLINEPATH	P	스플라인 그룹화 정의		+/-	-	PM-NC
SPN	A	블록 당 경로 섹션 수	s	+/-		PM-NC
SPOF [6]	G	스트로크 OFF, 니블링, 펀칭 OFF	m	+/-		PM-NC
SPOS	K	스핀들 위치	m	+/-	+/-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
SPOSA	K	블록 경계의 스핀들 위치	m	+/-		PM-NC
SPP	A	경로 섹션의 길이	m	+/-		PM-NC
SPRINT	F	포맷된 입력 문자열 반환		+/-		PM-NC
SQRT	F	제곱 근 (산술 함수)		+/-	+/-	PM-NC
SR	A	동기 동작을 위한 오실레이션 후퇴 경로	s	+/-		PM-NC
SRA	K	동기 동작을 위한 외부 입력 축을 포함한 오실레이션 후퇴 경로	m	+/-		PM-NC
ST	A	동기 동작을 위한 오실레이션 스파크아웃 시간	s	+/-		PM-NC
STA	K	동기 동작을 위한 오실레이션 스파크아웃 시간 축	m	+/-		PM-NC
START	P	현재 프로그램의 여러 채널에서 선택된 프로그램을 동시에 시작		+/-	-	PM-NC
STARTFIFO [6]	G	실행과 동시에 전처리 메모리 채움	m	+/-		PM-NC
STAT		조인트의 위치	s	+/-		PM-NC
STOLF	A	G0 공차 팩터	m	+/-		PM-NC
STOPFIFO	G	가공 정지; STARTFIFO가 탐지되거나, 전처리 메모리가 채워지거나 프로그램이 종료될 때까지 전처리 메모리를 채웁니다.	m	+/-		PM-NC
STOPRE	P	메인 실행에서 준비된 블록이 모두 실행될 때까지 전처리 정지		+/-	-	PM-NC
STOPREOF	P	전처리 중지 취소		-	+/-	FM-SA
STRING	K	데이터 유형: 문자열		+/-		PM-NC
STRINGIS	F	현재 NC 언어와 NC 싸이클 이름, 사용자 변수, 매크로 및 라벨 이름의 유효성, 정의 또는 활성 여부를 확인하기 위해 특히 해당 명령에 해당하는 이 항목들의 현재 범위를 확인합니다.		+/-	-	PM-NC
STRLEN	F	문자열 길이 정의		+/-	-	PM-NC
SUBSTR	F	입력 문자열에서 문자 인덱스 정의		+/-	-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
SUPA	G	프로그래밍된 옵셋, 시스템 프레임, 핸드휠 옵셋 (DRF), 외부 워크 옵셋 및 오버레이 이동을 포함한 현재의 워크 옵셋 억제	s	+/-		PM-NC
SUPD	G	활성 공구 옵셋 억제	s	+/-	-	PM-NC
SVC	K	공구 절삭 속도	m	+/-		PM-NC
SYNFCT	P	모션 동기 동작의 조건 함수로서 폴리노미얼 평가		-	+/-	FM-SA
SYNR	K	변수는 실행과 동시에 읽혀집니다.		+/-		PM-NC
SYNRW	K	변수는 실행과 동시에 작성 및 읽혀집니다.		+/-		PM-NC
SYNW	K	변수는 실행과 동시에 작성됩니다.		+/-		PM-NC
T	A	공구 호출 (머신 데이터에 지정될 때만 변경. 그렇지 않으면 M6 명령어가 필요함)		+/-		PM-NC
TAN	F	접선 (삼각 함수)		+/-	+/-	PM-NC
TANG	P	접선 제어: 커플링 정의		+/-	-	PM-NC
TANGDEL	P	접선 제어: 커플링 삭제		+/-	-	PM-NC
TANGOF	P	접선 제어: 커플링 비활성화		+/-	-	PM-NC
TANGON	P	접선 제어: 커플링 활성화		+/-	-	PM-NC
TCA (828D: _TCA)	P	공구 상태와 관계 없이 공구 선택/공구 교환		+/-	-	FM-TM
TCARR	A	공구 홀더 요청 (번호 "m")		+/-		PM-NC
TCI	P	버퍼에서 매거진으로 공구 로드		+/-	-	FM-TM
TCOABS[6]	G	현재 공구 오리엔테이션에서 공구 길이 구성요소 결정	m	+/-		PM-NC
TCOFR	G	활성화된 프레임의 오리엔테이션에서 공구 길이 요소 결정	m	+/-		PM-NC
TCOFRX	G	공구 선택 시 활성 프레임의 공구 오리엔테이션을 판정, 공구는 X 방향을 가리킴	m	+/-		PM-NC
TCOFRY	G	공구 선택 시 활성 프레임의 공구 오리엔테이션을 판정, 공구는 Y 방향을 가리킴	m	+/-		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
TCOFRZ	G	공구 선택 시 활성 프레임의 공구 오리엔테이션을 판정, 공구는 Z방향을 가리킴	m	+/-		PM-NC
THETA	A	회전 각도	s	+/-		PM-NC
TILT	A	틸트 각도	m	+/-		PM-NC
TLIFT	P	접선 제어: 중간 블록 생성 활성화		+/-	-	PM-NC
TML	P	매거진 포켓 번호를 사용하여 공구 선택		+/-	-	FM-TM
TMOF	P	공구 모니터링 선택을 해제		+/-	-	PM-NC
TMON	P	공구 모니터링 활성화		+/-	-	PM-NC
TO	K	FOR 카운터 루프에서 끝 값 지정		+/-		PM-NC
TOFF	A	인덱스에 지정된 기하 축에 대해 평행인 공구 길이 콤포넌트 방향에 있는 공구 길이 옵셋.	m	+/-		PM-NC
TOFFL	A	공구 길이 콤포넌트 L1, L2 또는 L3 방향에서 공구 길이 옵셋	m	+/-		PM-NC
TOFFLR	A	동시 공구 길이 옵셋 및 공구 반경 옵셋	m	+/-		PM-NC
TOFFOF	P	온라인 공구 옵셋 비활성화		+/-	-	PM-NC
TOFFON	P	온라인 공구 길이 옵셋 활성화		+/-	-	PM-NC
TOFFR	A	공구 반경 옵셋	m	+/-		PM-NC
TOFRAME	G	프레임을 회전시켜 WCS의 Z축을 공구 오리엔테이션과 평행하게 정렬	m	+/-		PM-NC
TOFRAMEX	G	프레임을 회전시켜 WCS의 X축을 공구 오리엔테이션과 평행하게 정렬	m	+/-		PM-NC
TOFRAMEY	G	프레임을 회전시켜 WCS의 Y축을 공구 오리엔테이션과 평행하게 정렬	m	+/-		PM-NC
TOFRAMEZ	G	TOFRAME과 동일	m	+/-		PM-NC
TOLOWER	F	문자열의 글자를 소문자로 변환		+/-	-	PM-NC
TOOLENV	F	메모리에 저장된 공구 데이터 평가에 중요한 현재 상태를 저장		+/-	-	PM-NC
TOOLGNT	F	공구 그룹의 공구 번호 확인		+/-	-	FM-TM
TOOLGT	F	공구 그룹의 공구 T 번호 확인		+/-	-	FM-TM

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
TOROT	G	프레임을 회전시켜 WCS의 Z축을 공구 오리엔테이션과 평행하게 정렬	m	+/-		PM-NC
TOROTOF [6]	G	공구 방향에서 프레임 회전 OFF	m	+/-		PM-NC
TOROTX	G	프레임을 회전시켜 WCS의 X축을 공구 오리엔테이션과 평행하게 정렬	m	+/-		PM-NC
TOROTY	G	프레임을 회전시켜 WCS의 Y축을 공구 오리엔테이션과 평행하게 정렬	m	+/-		PM-NC
TOROTZ	G	TOROT와 동일	m	+/-		PM-NC
TOUPPER	F	문자열의 글자를 대문자로 변환		+/-	-	PM-NC
TOWBCS	G	기본 좌표계(BCS) 의 마모값	m	+/-		PM-NC
TOWKCS	G	좌표계 변환을 위한 공구 헤드의 좌표계에서 마모 값 (공구 회전을 통한 기계 좌표계와 다름)	m	+/-		PM-NC
TOWMCS	G	기계 좌표계(MCS)로의 마모 값	m	+/-		PM-NC
TOWSTD [6]	G	공구 길이 옵셋의 초기 설정 값	m	+/-		PM-NC
TOWTCS	G	공구 좌표계에서 마모값 (공구 홀더에서의 공구 홀더 기준점 T)	m	+/-		PM-NC
TOWWCS	G	공작물 좌표계(WCS)로의 마모값	m	+/-		PM-NC
TR	K	프레임 변수의 옵셋 콤포넌트		+/-		PM-NC
TRAANG	P	경사 축 변환		+/-	-	PM-NC
TRACON	P	계단식 변환		+/-	-	PM-NC
TRACYL	P	실린더: 원주면 변환		+/-	-	PM-NC
TRAFOOF	P	채널에서 활성 변환 비활성화		+/-	-	PM-NC
TRAFOON	P	키네마틱 체인을 이용해 지정된 변환 활성화		+/-	-	PM-NC
TRAILOF	P	비동기 커플 모션 OFF		+/-	+/-	PM-NC
TRAILON	P	비동기 커플 모션 ON		+/-	+/-	PM-NC
TRANS	G	절대 프로그래밍 가능한 워크 옵셋	s	+/-		PM-NC
TRANSMIT	P	극 변환 (면 가공)		+/-	-	PM-NC
TRAORI	P	4-축, 5-축 변환, 일반 변환		+/-	-	PM-NC
TRUE	K	논리 상수: True		+/-		PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1] [2] [3] [4] [5]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
TRUNC	F	소수 자리 잘라내기		+/-	+/-	PM-NC
TU		축 각도	s	+/-		PM-NC
TURN	A	나선형의 회전 수	s	+/-		PM-NC
ULI	K	변수의 상한값		+/-		PM-NC
UNLOCK	P	ID로 동기 동작 활성화 (고정 싸이클 지속)		-	+/-	FM-SA
UNTIL	K	REPEAT 루프 끝의 조건		+/-		PM-NC
UPATH	G	FGROUP 축의 경로 기준은 곡선 파라미터입니다.	m	+/-		PM-NC
VAR	K	키워드: 파라미터 전송 유형		+/-		PM-NC
VELOLIM	K	최대 축 속도 또는 스핀들 속도 조정	m	+/-	-	PM-NC
VELOLIMA	K	최대 슬레이브 축 속도의 감소 또는 오버슈트	m	+/-	+/-	PM-NC
WAITC	P	축/스핀들에 대해 충족시켜야 할 커플링 블록 변경 기준 대기		+/-	-	PM-NC
WAITE	P	다른 채널에서 프로그램이 끝날 때까지 대기.		+/-	-	PM-NC
WAITENC	P	동기 축 위치 또는 복구된 축 위치 대기		+/-	-	PM-NC
WAITM	P	지정된 채널에서 마커 대기, 정위치 정지를 통해 이전 블록 종료		+/-	-	PM-NC
WAITMC	P	지정된 채널에서 마커 대기. 다른 채널이 마커에 도달하지 않았을 경우에만 정위치 정지		+/-	-	PM-NC
WAITP	P	포지셔닝 축의 이송 종료 대기		+/-	-	PM-NC
WAITS	P	스핀들 위치에 도달할 때까지 대기		+/-	-	PM-NC
WALCS0 [6]	G	공작물 좌표계 작업 영역 제한 선택이 취소됨	m	+/-	-	PM-NC
WALCS1	G	WCS 작업 구역 제한 그룹 1의 활성화	m	+/-	-	PM-NC
WALCS2	G	WCS 작업 구역 제한 그룹 2의 활성화	m	+/-	-	PM-NC
WALCS3	G	WCS 작업 구역 제한 그룹 3의 활성화	m	+/-	-	PM-NC
WALCS4	G	WCS 작업 구역 제한 그룹 4의 활성화	m	+/-	-	PM-NC

작업	종류[1]	의미	W[2]	TP[3]	SA[4]	설명 참조[5]
[1) 2) 3) 4) 5)]에 대한 설명은 범례 (페이지 1403)를 참조하십시오.						
WALCS5	G	WCS 작업 구역 제한 그룹 5의 활성화	m	+/-	-	PM-NC
WALCS6	G	WCS 작업 구역 제한 그룹 6의 활성화	m	+/-	-	PM-NC
WALCS7	G	WCS 작업 구역 제한 그룹 7의 활성화	m	+/-	-	PM-NC
WALCS8	G	WCS 작업 구역 제한 그룹 8의 활성화	m	+/-	-	PM-NC
WALCS9	G	WCS 작업 구역 제한 그룹 9의 활성화	m	+/-	-	PM-NC
WALCS10	G	WCS 작업 구역 제한 그룹 10의 활성화	m	+/-	-	PM-NC
WALIMOF	G	BCS 작업 영역 제한 OFF	m	+/-	-	PM-NC
WALIMON[6]	G	BCS 작업 영역 제한 ON	m	+/-	-	PM-NC
WHEN	K	조건이 충족될 때마다 동작이 한 번씩 실행됨		-	+/-	FM-SA
WHENEVER	K	조건이 충족될 때 각 보간기 싸이클에서 동작이 반복해서 실행됨		-	+/-	FM-SA
WHILE	K	WHILE 프로그램 루프 시작		+/-		PM-NC
WRITE	P	파일 시스템에 텍스트 기록 지정된 파일 끝에 블록을 추가합니다.		+/-	-	PM-NC
WRTPR	P	OPI 변수에 문자열 작성		+/-	-	PM-NC
X	A	축 이름	m/s	+/-	+/-	PM-NC
XOR	O	로직 배타적 OR		+/-		PM-NC
Y	A	축 이름	m/s	+/-	+/-	PM-NC
Z	A	축 이름	m/s	+/-	+/-	PM-NC

범례

1) 작업의 종류:

A 주소

값이 할당되는 식별자입니다 (예: OVR=10). 값을 할당하지 않고 기능을 설정하거나 해제하는 주소도 일부 있습니다 (예: CPLON 및 CPLOF).

C (A) AST 싸이클

AST(Automatic Servo Tuning)를 이용한 자동 사후 최적화를 위한 사전 정의 NC 프로그램. 파라미터는 특정 최적화 상황에 맞추는 데 사용됩니다. 이러한 파라미터는 호출 시 전송됩니다.

C 측정 싸이클

(M) 원통형 공작물의 내경 결정 같은 일반적으로 유효하고 구체적인 측정 작업이 프로그램되는 사전 정의 NC 프로그램. 파라미터는 특정 최적화 상황에 맞추는 데 사용됩니다. 이러한 파라미터는 호출 시 전송됩니다.

C (T) 기술 싸이클

나사 태핑이나 포켓 밀링 같은 일반적으로 유효하고 구체적인 가공 작업이 프로그램되는 사전 정의 NC 프로그램. 파라미터는 특정 최적화 상황에 맞추는 데 사용됩니다. 이러한 파라미터는 호출 시 전송됩니다.

F 사전 정의된 기능 (반환값 제공)

사전 정의된 기능 호출은 표현식에 피연산자로 포함시킬 수 있습니다.

G G 명령

G 명령은 G 그룹으로 나눠집니다. 동일한 그룹에 속한 G 명령은 한 블록에서 하나만 프로그래밍 할 수 있습니다. G 명령은 모달(동일한 그룹 내의 다른 명령에 의해 취소될 때까지 적용)이거나 프로그래밍된 블록에 대해서만 유효합니다(넌모달).

K 키워드

블록 구문을 정의하는 식별자입니다. 키워드에는 값이 할당되지 않으며 키워드를 사용하여 NC 기능을 설정/해제할 수 없습니다.

예: 제어 구조 (IF, ELSE, ENDIF, WHEN, ...), 프로그램 실행 (GOTOB, GOTO, RET ...)

O 연산자

수학, 비교 또는 논리 연산을 위한 연산자

P 사전 정의된 절차 (반환값 제공하지 않음)

PA 프로그램 속성

프로그램 속성은 서브프로그램 정의 행 끝에 있습니다.

```
PROC <프로그램 이름>(...) <프로그램 속성>
```

서브프로그램을 실행하는 동안 동작을 결정합니다.

[2] 조작의 효과:

m 모달

s 넌모달

[3] 가공 프로그램에서 프로그래밍:

+ 프로그램 가능

- 프로그램 가능하지 않음

M 장비 제조업체만 프로그램 가능

[4] 동기 동작에서 프로그래밍:

+ 프로그램 가능

- 프로그램 가능하지 않음

T 기술 싸이클에서만 프로그램 가능

[5] 세부적인 조작 설명이 포함된 참고 자료:

FM-A 기능 매뉴얼 축 및 스핀들

FM-B 기능 매뉴얼 기본 기능

FM-SA 기능 매뉴얼 동기 동작

FM-SI 기능 매뉴얼, Safety Integrated

FM-TE 기능 매뉴얼 기술

FM-TM 기능 매뉴얼, 공구 관리

FM-TR 기능 매뉴얼 변환

PM-MC 측정 싸이클 프로그래밍 매뉴얼

PM-NC 프로그래밍 매뉴얼 NC 프로그래밍

[6] 프로그램 시작 시 기본 설정(별도로 프로그램하지 않은 경우 시스템의 공장 출하 설정)

5.2 어드레스

5.2.1 어드레스 문자

문자	의미	수치 확장
A	셋터블 어드레스 식별자	x
B	셋터블 어드레스 식별자	x

문자	의미	수치 확장
C	셋터블 어드레스 식별자	x
D	공구 길이 보정 선택 및 선택 해제, 공구 절삭날	
E	셋터블 어드레스 식별자	x
F	이송 속도 초 단위 드웰 시간	x
G	G 코드	
H	H 기능	x
I	셋터블 어드레스 식별자	x
J	셋터블 어드레스 식별자	x
K	셋터블 어드레스 식별자	x
L	서브프로그램 이름, 서브프로그램 호출	
M	M 코드	x
N	서브블록 번호	
O	미지정	
P	프로그램 실행 횟수	
Q	셋터블 어드레스 식별자	x
R	변수 식별자 (R 파라미터) 셋터블 어드레스 식별자 (수치 확장 없음)	x
S	스핀들 값 스핀들 회전수 단위 드웰 시간	x x
T	공구 번호	x
U	셋터블 어드레스 식별자	x
V	셋터블 어드레스 식별자	x
W	셋터블 어드레스 식별자	x
X	셋터블 어드레스 식별자	x
Y	셋터블 어드레스 식별자	x
Z	셋터블 어드레스 식별자	x
%	파일 전송을 위한 시작 문자 및 구분자	
:	메인 블록 번호	
/	스킵 식별자	

5.2.2 고정 어드레스

축 확장자가 없는 고정 주소

주소 식별자	주소 유형	모달/넌모달	G70/G71	G700/G710	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	지정된 값의 데이터 유형
D	옵셋 번호	m								x	미지정 INT
F	이송, 드웰 시간	m, s	x							x	미지정 REAL
FLIM	최대 경로 속도	m									미지정 REAL
G	G 명령어	G 명령어 목록 참조									미지정 INT
H	보조 기능	s								x	M: 미지정 INT H: REAL
L	서브프로그램 번호	s									미지정 INT
M	보조 기능	s								x	M: 미지정 INT H: REAL
N	블록 번호	s									미지정 INT
OVR	오버라이드	m									미지정 REAL

주소 식별자	주소 유형	모달/넌모달	G70/G71	G700/G710	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	지정된 값의 데이터 유형
OVRRAP	급 이송 속도 오버라이드	m									미지정 REAL
P	서브프로그램 반복 횟수	s									미지정 INT
PACCLIM	최대 경로 가속	m									미지정 REAL
S	스핀들, 드웰 시간	m, s								x	미지정 REAL
SCC	가로 축을 G96 /G961/ G962에 지정	m									REAL
SPOS	스핀들 위치	m				x	x	x			REAL
SPOSA	블록 경계의 스핀들 위치	m				x	x	x			REAL
T	공구 번호	m								x	미지정 INT

축 확장자 있는 고정 주소

주소 식별자	주소 유형	모달/넌모달	G70/G71	G700/G710	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	지정된 값의 데이터 유형
ACC	축 가속	m									미지정 REAL
ACCLIMA	추종 축의 축 가속 제한	m									미지정 REAL
AX	변수 축 이름	[1)]	x	x	x	x	x	x			REAL
FA	축 이송 속도	m	x							x	미지정 REAL
FDA	핸드휠 오버라이드의 축 이송 속도	s	x								미지정 REAL
FGREF	기준 반경	m	x	x							미지정 REAL
FL	축 이송 속도 한계	m	x								미지정 REAL
FMA	축 동기화 이송 속도	m									미지정 REAL
FOC	토크가 제한된 넌모달 이송	s									REAL
FOCOF	토크가 제한된 모달 이송 OFF	m									REAL
FOCON	토크가 제한된 모달 이송 ON	m									REAL

주소 식별자	주소 유형	모달/넌모달	G70/G71	G700/G710	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	지정된 값의 데이터 유형
FXS	고정 정지점으로 이동 ON	m									미지정 INT
FXST	고정 정지점으로 이동 시 토크 한계	m									REAL
FXSW	고정 정지점으로 이동 모니터링 창	m									REAL
IP	변수 보간 파라미터	s	x	x	x	x	x				REAL
JERKLIM	축 저크 제한	m									미지정 REAL
JERKLIMA	추종 축의 축 저크 제한	m									미지정 REAL
MEAC	주기적 측정	s									INT 모드 및 1~4개 트리거 이벤트
MEASA	이동할 거리를 삭제하고 축 측정	s									INT 모드 및 1~4개 트리거 이벤트
MEAWA	이동할 거리를 삭제하지 않고 축 측정	s									INT 모드 및 1~4개 트리거 이벤트

주소 식별자	주소 유형	모달/넌모달	G70/G71	G700/G710	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	지정된 값의 데이터 유형
MOV	축 포지셔닝 시작	m	x	x	x	x	x	x	x		REAL
OS	오실레이션 ON/OFF	m									미지정 INT
OSB	오실레이션 시작점	m	x	x	x	x	x	x			REAL
OSCILL	오실레이션의 축 지정, 오실레이션 활성화	m									축: 1~3개 절입 축
OSCTRL	오실레이션 옵션	m									미지정 INT: 설정 옵션, 미지정 INT: 리셋 옵션
OSE	오실레이션 종료 위치	m	x	x	x	x	x	x			REAL
OSNSC	스파크아웃 싸이클 개수 (오실레이션)	m									미지정 INT
OSP1	좌측 반전 지점 (오실레이션)	m	x	x	x	x	x	x			REAL
OSP2	우측 반전 지점 (오실레이션)	m	x	x	x	x	x	x			REAL

주소 식별자	주소 유형	모달/넌모달	G70/G71	G700/G710	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	지정된 값의 데이터 유형
OST1	좌측 반전 지점에서 정지 시간 (오실레이션)	m									REAL
OST2	우측 반전 지점에서 정지 시간 (오실레이션)	m									REAL
OVRA	축 오버라이드	m	x								미지정 REAL
PO	폴리노미얼 계수	s	x	x		x	x	x			미지정 REAL
POLF	LIFTFAST 위치	m	x	x							미지정 REAL
POS	포지셔닝 축	m	x	x	x	x	x	x	x		REAL
POSA	블록 경계의 포지셔닝 축	m	x	x	x	x	x	x	x		REAL
POSP	여러 위치에 축 포지셔닝 (오실레이션)	m	x	x	x	x	x	x			REAL: 끝 위치 Real: 부분 길이 INT: 옵션
STA	스파크아웃 시간 (축)	m									미지정 REAL

주소 식별자	주소 유형	모달/넌모달	G70/G71	G700/G710	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	지정된 값의 데이터 유형
SRA	외부 입력의 후퇴 경로(축)	m									미지정 REAL
VELOLIM	축 속도 제한	m									미지정 REAL
VELOLIMA	추종 축의 축 속도 제한	m									미지정 REAL

[1] 절대 종점: 모달, 증분 종점: 넌모달 또는 구문을 결정하는 G 명령어에 따라 모달/넌모달.

5.2.3 셋터블 어드레스

어드레스 식별자 (기본 설정)	어드레스 유형	모달/넌모달	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	최대 개수	지정된 값의 데이터 유형
축 값 및 종점											
X, Y, Z, A, B, C	축	[1]	x	x	x	x				8	REAL
AP	극점 각도	m/s [1]	x	x	x					1	REAL
RP	극점 반경	m/s [1]	x	x	x					1	미지정 REAL
공구 오리엔테이션											
A2, B2, C2	Euler 각도 또는 RPY 각도	s								3	REAL

어드레스 식별자 (기본 설정)	어드레스 유형	모달/넌모달	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	최대 개수	지정된 값의 데이터 유형
A3, B3, C3	방향 벡터 컴포넌트	s								3	REAL
A4, B4, C4	블록 시작점에서의 표면 법선 벡터 컴포넌트	s								3	REAL
A5, B5, C5	블록 끝에서의 표면 법선 벡터 컴포넌트	s								3	REAL
A6, B6, C6	테이퍼의 로터리 축의 방향 벡터 컴포넌트	s								3	REAL
A7, B7, C7	테이퍼의 주변 표면에서의 중간 오리엔테이션을 위한 벡터 컴포넌트	s								3	REAL
LEAD	리드 각도	m								1	REAL
THETA	회전 각, 공구 방향 회전	m		x	x					1	REAL
TILT	틸트 각도	m								1	REAL
ORIS	오리엔테이션 변경 (경로 기준)	m								1	REAL
보간 파라미터											
I, J, K	보간 파라미터 중간 지점 좌표	s		x[2)]	x[2)]					3	REAL
I1, J1, K1		s	x	x	x					3	REAL
RPL	평면에서 회전	s								1	REAL

어드레스 식별자 (기본 설정)	어드레스 유형	모달/넌모달	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	최대 개수	지정된 값의 데이터 유형
CR	원호 반경	s								1	미지정 REAL
AR	틈 각도	s								1	미지정 REAL
TURN	나선의 회전 개수	s								1	미지정 INT
PL	파라미터 간격 길이	s								1	미지정 REAL
PW	가중치	s								1	미지정 REAL
SD	스플라인 각도	m								1	미지정 INT
TU	축 각도	s								1	미지정 INT
STAT	조인트의 위치	m								1	미지정 INT
SF	나사 절삭을 위한 시작점 옵셋	m								1	REAL
DISCL	안전 거리 SAR	s								1	미지정 REAL
DISR	리포지셔닝 공차/SAR 공차	s								1	미지정 REAL
DISPR	리포지셔닝 경로 편차	s								1	미지정 REAL
ALF	고속 도피 각도	m								1	미지정 INT
DILF	고속 도피 길이	m								1	REAL
FP	고정 정지점: 접근할 고정 정지점의 수	s								1	미지정 INT

어드레스 식별자 (기본 설정)	어드레스 유형	모달/넌모달	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	최대 개수	지정된 값의 데이터 유형
RNDM	모달 라운딩	m								1	미지정 REAL
RND	넌모달 라운딩	s								1	미지정 REAL
CHF	넌모달 모따기	s								1	미지정 REAL
CHR	원래 이동 방향에서 모따기	s								1	미지정 REAL
ANG	형상 각도	s								1	REAL
ISD	삽입 깊이	m								1	REAL
DISC	트렌지션 원 오버슈트 공구 반경 보정	m								1	미지정 REAL
OFFN	옵셋 형상 법선	m								1	REAL
DITS	나사 런인 경로	m								1	REAL
DITE	나사 런아웃 경로	m								1	REAL
코너 라운딩 기준											
ADIS	라운딩 공차	m								1	미지정 REAL
ADISPOS	급 이송의 라운딩 공차	m								1	미지정 REAL
측정											
MEAS	터치 프로브를 사용한 측정	s								1	미지정 INT

어드레스 식별자 (기본 설정)	어드레스 유형	모달/넌모달	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	최대 개수	지정된 값의 데이터 유형
MEAW	이동할 거리를 삭제하지 않고 터치 프로브로 측정	s								1	미지정 INT
축/스핀들 동작											
LIMS	스핀들 속도 제한	m								1	미지정 REAL
COARSEA	블록 변경 동작: 축별 일반 정위치 정지	m									
FINEA	블록 변경 동작: 축별 미세 정위치 정지	m									
IPOENDA	블록 변경 동작: 축별 보간자 정지	m									
DIACYCOFA	가로 축: 싸이클에서 축 직경 프로그래밍 OFF	m									
DIAM90A	가로 축: G90 축 직경 프로그래밍	m									
DIAMCHAN	가로 축: 직경 프로그래밍 채널 상태에서 모든 가로 축 이송	m									

어드레스 식별자 (기본 설정)	어드레스 유형	모달/넌모달	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	최대 개수	지정된 값의 데이터 유형
DIAMCHANA	가로 축: 직경 프로그래밍 채널 상태 전송	m									
DIAMOFA	가로 축: 축별 직경 프로그래밍 OFF	m									
DIAMONA	가로 축: 축별 직경 프로그래밍 ON	m									
GP	위치: 위치 속성 간접 프로그래밍	m									
이송 속도											
FAD	저속 피드 이송 속도	s						x		1	미지정 REAL
FD	핸드휠 오버라이드의 경로 이송 속도	s								1	미지정 REAL
FRC	반경 및 모따기를 위한 이송 속도	s								1	미지정 REAL
FRCM	반경 및 모따기를 위한 이송 속도, 모달	m								1	미지정 REAL
FB	넌모달 이송 속도	s								1	미지정 REAL
니블링 및 펀치											
SPN	블록 당 경로 섹션 수	s								1	INT

어드레스 식별자 (기본 설정)	어드레스 유형	모달/넌모달	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	최대 개수	지정된 값의 데이터 유형
SPP	경로 섹션의 길이	m								1	REAL
연삭											
ST	스파크아웃 시간	s								1	미지정 REAL
SR	후퇴 경로	s								1	미지정 REAL
공구 선택											
TCARR	공구 케리어	m								1	INT
공구 관리											
DL	총 공구 옵셋	m								1	INT
OEM 어드레스											
OMA1	OEM 어드레스 1	m		x	x	x				1	REAL
OMA2	OEM 어드레스 2	m		x	x	x				1	REAL
OMA3	OEM 어드레스 3	m		x	x	x				1	REAL
OMA4	OEM 어드레스 4	m		x	x	x				1	REAL
OMA5	OEM 어드레스 5	m		x	x	x				1	REAL
기타											

어드레스 식별자 (기본 설정)	어드레스 유형	모달/넌모달	G90/G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	최대 개수	지정된 값의 데이터 유형
CUTMOD	회전식 공구에 대한 옵셋 데이터 수정(오리엔테이션 가능 공구 캐리어와 함께)	m									INT
CUTMODK	회전식 공구에 대한 옵셋 데이터 수정(키네마틱 체인을 통해 지정된 오리엔테이션 변환과 함께)	m									STRING
TOFF	지정된 기하 축과 평행한 공구 길이 옵셋	m									REAL
TOFFL	공구 길이 콤포넌트 L1, L2 또는 L3 방향에서 공구 길이 옵셋	m									REAL
TOFFR	공구 반경 옵셋	m									REAL
TOFFLR	동시 공구 길이 옵셋 및 공구 반경 옵셋	m									REAL

[1] 절대 종점: 모달, 증분 종점: 넌모달, 그렇지 않은 경우 구문을 결정하는 G 코드에 따라 모달/넌모달.

[2] IPO 파라미터는 원호 중심점에 따라 증분식으로 적용됩니다. 이 파라미터는 AC를 사용하여 절대 모드에서 프로그래밍할 수 있습니다. 파라미터가 다른 의미 (예: 나사 리드) 를 가지면 수정된 어드레스는 무시됩니다.

5.3 G 명령

5.3.1 G 코드

G 명령은 G 그룹으로 나눠집니다. 가공 프로그램 또는 동기 동작에서, 한 블록에 오직 한 G 그룹의 한 G 명령만을 사용할 수 있습니다. G 명령은 모달이거나 넌모달일 수 있습니다.

모달: 동일한 G 그룹 내 다른 G 명령의 프로그래밍까지.

5.3.2 G 그룹 1: 동작 명령어로 유효한 모달 명령어

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G0	1	급 이송	+	m		
G1	2	직선 보간 (직선 보간)	+	m	x	
G2	3	CW 방향 원호 보간	+	m		
G3	4	CCW 방향 원호 보간	+	m		
CIP	5	중간점을 통한 원호 보간	+	m		
ASPLINE	6	Akima 스플라인	+	m		
BSPLINE	7	B 스플라인	+	m		
CSPLINE	8	큐빅 스플라인	+	m		
POLY	9	폴리노미얼 보간	+	m		
G33	10	리드 일정 나사 절삭	+	m		
G331	11	태핑	+	m		
G332	12	후진 (태핑)	+	m		
OEMIPO1	13	예비	+	m		
OEMIPO2	14	예비	+	m		
CT	15	접선 트렌지션에 의한 원호	+	m		
G34	16	리드 증가 나사 절삭	+	m		
G35	17	리드 감소 나사 절삭	+	m		
INVCW	18	나선 보간 (CW)	+	m		
INVCCW	19	나선 보간 (CCW)	+	m		

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G335	20	CW 방향으로 볼록 나사 선삭	+	m		
G336	21	CCW 방향으로 볼록 나사 선삭	+	m		

5.3.3 G 그룹 2: 넌모달로 적용되는 동작, 드웰 시간

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G4	1	드웰 시간. 프리셋	-	s		
G63	2	비 동기화 태핑	-	s		
G74	3	동기화 원점 복귀	-	s		
G75	4	고정 정지점 접근	-	s		
REPOSL	5	직선 리포지셔닝	-	s		
REPOSQ	6	4분원 리포지셔닝	-	s		
REPOSH	7	반원 리포지션	-	s		
REPOSA	8	모든 축 직선 리포지셔닝	-	s		
REPOSQA	9	사분원 내 모든 축, 기하 축의 직선 리포지셔닝	-	s		
REPOSHA	10	반원 내 모든 축, 기하 축의 리포지셔닝	-	s		
G147	11	직선을 따라 형상에 접근	-	s		
G247	12	4분원을 따라 형상에 접근	-	s		
G347	13	반원을 따라 형상에 접근	-	s		
G148	14	직선을 따라 형상에서 출발	-	s		
G248	15	4분원을 따라 형상에서 출발	-	s		
G348	16	반원을 따라 형상에서 출발	-	s		
G5	17	사면 플런지 연삭	-	s		
G7	18	사면 플런지 연삭 동안의 보상 동작	-	s		

5.3.4 G 그룹 3: 프로그래밍 프레임, 작업 구역 제한과 극점 프로그래밍

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
TRANS	1	변환: 프로그램 옵셋	-	s		
ROT	2	회전: 프로그래밍 회전	-	s		
SCALE	3	배율: 프로그래밍 배율	-	s		
MIRROR	4	미러링: 프로그래밍 미러링	-	s		
ATRANS	5	추가 변환: 추가 프로그래밍 변환	-	s		
AROT	6	추가 회전: 프로그래밍 회전	-	s		
ASCALE	7	추가 SCALE: 프로그래밍 배율	-	s		
AMIRROR	8	추가 미러링: 프로그래밍 미러링	-	s		
-	9	미지정	-	-		
G25	10	최소 작업 영역 제한/스핀들 속도 제한	-	s		
G26	11	최대 작업 영역 제한/스핀들 속도 제한	-	s		
G110	12	최종 프로그램 설정점 위치에 상대적인 극점 프로그래밍	-	s		
G111	13	현재 공작물 좌표계의 원점에 상대적인 극점 프로그래밍	-	s		
G112	14	최종 유효 극점에 상대적인 극점 프로그래밍	-	s		
G58	15	절대 프로그래밍 가능한 워크 옵셋	-	s		
G59	16	부가 프로그래밍 가능한 워크 옵셋	-	s		
ROTS	17	입체각을 이용한 회전	-	s		
AROTS	18	입체각 사용 추가 회전	-	s		

5.3.5 G 그룹 4: FIFO

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
STARTFIFO	1	FIFO 시작 실행과 동시에 전처리 메모리 채우기	+	m	x	
STOPFIFO	2	FIFO 정지 가공 정지. STARTFIFO가 탐지되거나 FIFO가 가득 차거나 혹은 프로그램이 종료될 때까지 전처리 메모리 채우기.	+	m		
FIFOCTRL	3	전처리 메모리 자동 제어 활성화	+	m		

5.3.6 G 그룹 6: 평면 선택

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G17	1	평면 선택 1. – 2. 기하 축	+	m	x	
G18	2	평면 선택 3. – 1. 기하 축	+	m		
G19	3	평면 선택 2. – 3. 기하 축	+	m		

5.3.7 G 그룹 7: 공구 반경 보정

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G40	1	공구 반경 보정 취소	+	m	x	
G41	2	형상 좌측 공구 반경 보정	-	m		
G42	3	형상 우측 공구 반경 보정	-	m		

5.3.8 G 그룹 8: 설정 가능한 워크 옵셋

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G500	1	설정 가능한 워크 옵셋 비활성화 (G54~G57, G505~G599)	+	m	x	
G54	2	1번째 설정 가능한 워크 옵셋	+	m		
G55	3	2번째 설정 가능한 워크 옵셋	+	m		
G56	4	3번째 설정 가능한 워크 옵셋	+	m		
G57	5	4번째 설정 가능한 워크 옵셋	+	m		
G505	6	5번째 설정 가능한 워크 옵셋	+	m		
...	...	...	+	m		
G599	100	99번째 설정 가능한 워크 옵셋	+	m		
G 그룹의 각 G 명령어는 조정 가능한 사용자 프레임 $P_UIFR[ ]을 활성화하는 데 사용됩니다. G54 는 프레임 $P_UIFR[1]와 일치하고, G505는 프레임 $P_UIFR[5]와 일치합니다. 조정 가능한 사용자 프레임 개수와 이 G 그룹의 G 명령어 개수는 머신 데이터 MD28080 $MC_MM_NUM_USER_FRAMES를 사용하여 설정할 수 있습니다.						

5.3.9 G 그룹 9: 프레임 및 공구 옵셋 억제

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G53	1	현재 프레임 억제: TOROT 및 TOFRAME 시스템 프레임을 포함한 프로그래밍 가능한 프레임과 활성 조정 가능 프레임 (G54~ G57, G505~ G599)	-	s		
SUPA	2	실제 값 설정, 스크래칭, 외부 워크 옵셋, 핸드휠 옵셋(DRF)을 포함한 PAROT, [외부 워크 옵셋], 오버레이 이동에 대한 시스템 프레임의 억제를 포함하는 G153 관련	-	s		
G153	3	G53의 경우 모든 채널 및/또는 NCU 글로벌 기본 프레임 억제 포함	-	s		
SUPD	4	활성 공구 옵셋 억제	-	s		

5.3.10 G 그룹 10: 정위치 정지 / 연속 경로 모드

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G60	1	정위치 정지	+	m	x	
G64	2	연속 경로 모드	+	m		
G641	3	거리 기준에 따라 스무딩하는 연속 경로 모드 (= 프로그래밍 라운딩 공차)	+	m		
G642	4	정해진 공차 안에서 스무딩하는 연속 경로 모드	+	m		
G643	5	정해진 공차 안에서 스무딩하는 연속 경로 모드 (블록 내경)	+	m		
G644	6	동적 응답을 가능 최대치로 하여 스무딩하는 연속 경로 모드	+	m		
G645	7	정해진 공차 안에서 스무딩 및 접선 블록 트렌지션을 갖는 연속 경로 모드	+	m		

5.3.11 G 그룹 11: 정위치 정지, 넌모달

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G9	1	정위치 정지	-	s		

5.3.12 G 그룹 12: 정위치 정지의 블록 변경 기준(G60/G9)

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G601	1	미세 정위치 정지에서 블록 변경	+	m	x	
G602	2	일반 정위치 정지에서 블록 변경	+	m		
G603	3	IPO 블록 끝에서 블록 변경	+	m		

5.3.13 G 그룹 13: 인치/미터 시스템

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G70	1	인치 입력 시스템 (길이)	+	m		
G71	2	미터 입력 시스템 (길이)	+	m	x	
G700	3	인치 입력 시스템, inch/min (길이 + 속도 + 시스템 변수)	+	m		
G710	4	미터 입력 시스템, mm/min (길이 + 속도 + 시스템 변수)	+	m		

5.3.14 G 그룹 14: 절대/증분 시스템

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G90	1	절대 치수	+	m	x	
G91	2	증분 치수	+	m		

5.3.15 G 그룹 15: 이송 형식

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G93	1	역 시간 이송 속도 rpm	+	m		
G94	2	분당 이송 (mm/min, inch/min)	+	m	x	
G95	3	회전당 이송 (mm/rev, inch/rev)	+	m		
G96	4	회전 당 이송 속도(G95 관련) 및 주속 일정 제어	+	m		
G97	5	회전 당 이송 속도 및 일정 스핀들 속도(주속 일정 제어 OFF)	+	m		
G931	6	이동 시간을 통해 지정된 이송 속도, 일정 경로 속도 해제	+	m		
G961	7	직선 이송 속도(G94 관련) 및 주속 일정 제어	+	m		

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G971	8	직선 이송 속도 및 일정 스핀들 속도(주속 일정 제어 OFF)	+	m		
G942	9	직선 이송 속도 및 주속 일정 제어 또는 일정 스핀들 속도	+	m		
G952	10	회전 당 이송 속도 및 주속 일정 제어 또는 일정 스핀들 속도	+	m		
G962	11	직선 또는 회전 당 이송 속도 및 주속 일정 제어	+	m		
G972	12	직선 이송 속도 또는 회전 당 이송 속독 및 일정 스핀들 속도(주속 일정 제어 OFF)	+	m		
G973	13	스핀들 속도 제한 없는 회전 당 이송 속도 및 일정 스핀들 속도(ISO 모드를 위한 LIMS 없는 G97)	+	m		

5.3.16 G 그룹 16: 커브 내부 및 외부의 피드 오버라이드

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CFC	1	내부 및 외부 반경에 형상에서 이송 속도 일정	+	m	x	
CFTCP	2	공구 중심점의 이송 속도 일정(중심점 경로)	+	m		
CFIN	3	내부 반경에서만 이송 속도 일정, 외부 반경의 경우는 가속	+	m		

5.3.17 G 그룹 17: 공구 옵셋의 접근 및 후퇴 동작

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
NORM	1	시작점과 종점에서 정상 위치	+	m	x	
KONT	2	시작점과 종점에서 형상 주위를 따라 이동	+	m		
KONTT	3	일정한 접선을 통해 접근/후퇴	+	m		
KONTC	4	일정한 곡률을 통해 접근/후퇴	+	m		

5.3.18 G 그룹 18: 공구 옵셋 코너 동작

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G450	1	원호 트렌지션 (공구는 공작물 코너에서 원호 경로를 따라 이동)	+	m	x	
G451	2	등거리 경로의 교차 (공구는 공작물 코너로부터 후퇴)	+	m		

5.3.19 G 그룹 19: 스플라인 시작점에서의 커브 트렌지션

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
BNAT	1	첫 번째 스플라인 블록으로 자연적인 트렌지션	+	m	x	
BTAN	2	첫 번째 스플라인 블록으로 접선 트렌지션	+	m		
BAUTO	3	다음의 세 점을 사용한 첫 번째 스플라인 섹션 정의	+	m		

5.3.20 G 그룹 20: 스플라인 끝에서 커브 트렌지션

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ENAT	1	다음 이송 블록으로 자연적 트렌지션	+	m	x	
ETAN	2	다음 이송 블록으로 접선 트렌지션	+	m		
EAUTO	3	마지막 세 점을 사용한 마지막 스플라인 섹션 정의	+	m		

5.3.21 G 그룹 21: 가속 특성

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
BRISK	1	빠른 가감속	+	m	x	
SOFT	2	부드러운 가감속	+	m		
DRIVE	3	속도 의존 가감속	+	m		

5.3.22 G 그룹 22: 공구 옵셋 유형

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CUT2D	1	2½D TRC	+	m	x	
CUT2DF	2	현재 프레임에 대한 2½D TRC(경사 평면)	+	m		
CUT3DC	3	원주 밀링을 위한 3D TRC	+	m		
CUT3DF	4	오리엔테이션 변경을 사용하는 페이스 밀링을 위한 3D TRC	+	m		
CUT3DFS	5	일정한 오리엔테이션을 사용하는 페이스 밀링을 위한 3D TRC 공구 오리엔테이션은 G17 - G19에 의해 정의되고, 프레임의 영향은 받지 않습니다.	+	m		
CUT3DFF	6	일정한 오리엔테이션을 사용하는 페이스 밀링을 위한 3D TRC 공구 오리엔테이션은 G17 - G19에 의해 정의된 방향이고, 일부 경우 프레임에 의해 회전됩니다.	+	m		
CUT3DCC	7	3D 반경 보정을 사용하는 한계 표면을 고려하는 원주 밀링용 3D TRC: 가공 표면의 형상	+	m		
CUT3DCCD	8	공구 중심점 경로에서 차동 공구를 사용하는 한계 표면을 고려하는 원주 밀링용 3D TRC: 한계 표면으로 절입	+	m		
CUT2DD	9	차동 공구 관련 2½ D TRC	+	m		
CUT2DFD	10	현재 프레임 관련 차동 공구 관련 2½D TRC(경사면)	+	m		

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CUT3DCD	11	원주 밀링의 차동 공구 관련 3D TRC	+	m		
CUT3DFD	12	오리엔테이션 변경과 함께 페이스 밀링의 차동 공구 관련 3D TRC	+	m		

5.3.23 G 그룹 23: 내부 형상에서 충돌 모니터링

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CDOF	1	충돌 탐지 OFF	+	m	x	
CDON	2	충돌 탐지 ON	+	m		
CDOF2	3	3D 원주 밀링의 충돌 탐지 OFF	+	m		

5.3.24 G 그룹 24: 사전 제어

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
FFWOF	1	피드포워드 제어 OFF	+	m	x	
FFWON	2	피드포워드 제어 ON	+	m		

5.3.25 G 그룹 25: 공구 오리엔테이션 기준

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIWKS	1	공작물 좌표계의 공구 오리엔테이션(WCS)	+	m	x	
ORIMKS	2	기계 좌표계의 공구 오리엔테이션 (MCS)	+	m		

5.3.26 G 그룹 26: REPOS 리포지셔닝 모드(모달)

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
RMB	1	블록 시작으로 리포지셔닝	-	m		
RMI	2	인터럽트 지점으로 리포지셔닝	-	m	x	
RME	3	블록 끝으로 리포지셔닝	-	m		
RMN	4	가장 가까운 경로 지점으로 리포지셔닝	-	m		

5.3.27 G 그룹 27: 외부 코너에서 오리엔테이션 변경을 위한 공구 옵셋

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIC	1	외부 코너에서의 오리엔테이션 변경은 삽입될 원호 블록과 겹쳐집니다	+	m	x	
ORID	2.	오리엔테이션 변경은 원호 블록 이전에 수행됩니다	+	m		

5.3.28 G 그룹 28: 작업 영역 제한

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
WALIMON	1	작업 영역 제한 ON	+	m	x	
WALIMOF	2	작업 영역 제한 OFF	+	m		

5.3.29 G 그룹 29: 반경/직경 프로그래밍

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
DIAMOF	1	채널 관련 모달 직경 프로그래밍 OFF 비활성화는 채널별 반경 프로그램을 활성화합니다.	+	m	x	
DIAMON	2	채널 관련 직경 프로그래밍 ON 그 효과는 프로그래밍된 치수 모드와 무관합니다(G90/G91).	+	m		
DIAM90	3	채널 관련 모달 직경 프로그래밍 ON 그 효과는 프로그래밍된 치수 모드의 영향을 받습니다(G90/G91).	+	m		
DIAMCYCOF	4	싸이클 처리 과정에서 채널 관련 모달 직경 프로그래밍 OFF	+	m		

5.3.30 G 그룹 30: NC 블록 압축

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
COMPOF	1	NC 블록 압축 OFF	+/-	m	x	
COMPON	2	컴프레서 기능 COMPON ON	+/-	m		
COMPCURV	3	컴프레서 기능 COMPCURV ON	+/-	m		
COMPCAD	4	컴프레서 기능 COMPCAD ON	+/-	m		
COMPSURF	5	COMPSURF ON 컴프레서 기능	+/-	m		

5.3.31 G 그룹 31: OEM G 명령

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G810	1	OEM G 명령	-	m		
G811	2	OEM G 명령	-	m		

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G812	3	OEM G 명령	-	m		
G813	4	OEM G 명령	-	m		
G814	5	OEM G 명령	-	m		
G815	6	OEM G 명령	-	m		
G816	7	OEM G 명령	-	m		
G817	8	OEM G 명령	-	m		
G818	9	OEM G 명령	-	m		
G819	10	OEM G 명령	-	m		
두 개의 G그룹은 OEM 사용자를 위해 유보됩니다. OEM은 이 코드 그룹을 이용하여 원하는 기능을 프로그래밍할 수 있습니다.						

5.3.32 G 그룹 32: OEM G 명령

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G820	1	OEM G 명령	-	m		
G821	2	OEM G 명령	-	m		
G822	3	OEM G 명령	-	m		
G823	4	OEM G 명령	-	m		
G824	5	OEM G 명령	-	m		
G825	6	OEM G 명령	-	m		
G826	7	OEM G 명령	-	m		
G827	8	OEM G 명령	-	m		
G828	9	OEM G 명령	-	m		
G829	10	OEM G 명령	-	m		
두 개의 G그룹은 OEM 사용자를 위해 유보됩니다. OEM은 이 코드 그룹을 이용하여 원하는 기능을 프로그래밍할 수 있습니다.						

5.3.33 G 그룹 33: 공구 옵셋 미세 조정 옵셋

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
FTOCOF	1	온라인 미세 공구 옵셋 OFF	+	m	x	
FTOCON	2	온라인 미세 공구 옵셋 ON	-	m		

5.3.34 G 그룹 34: 공구 오리엔테이션 스무딩

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
OSOF	1	공구 오리엔테이션 스무딩 OFF	+	m	x	
OSC	2	연속적인 공구 오리엔테이션 스무딩	+	m		
OSS	3	블록 끝에서의 공구 오리엔테이션 스무딩	+	m		
OSSE	4	블록 시작과 끝에서의 공구 오리엔테이션 스무딩	+	m		
OSD	5	경로 길이 지정으로 블록 내부 스무딩	+	m		
OST	6	각도 공차 지정으로 블록 내부 스무딩	+	m		

5.3.35 G 그룹 35: 펀칭 및 니블링

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
SPOF	1	스트로크 US, 펀칭 및 니블링 OFF	+	m	x	
SON	2	니블링 ON	+	m		
PON	3	펀칭 ON	+	m		
SONS	4	보간 싸이클 내 니블링 ON	-	m		
PONS	5	보간 싸이클 내 펀치 ON	-	m		

5.3.36 G 그룹 36: 지연 펀칭

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
PDELAYON	1	지연 ON 상태의 펀칭	+	m	x	
PDELAYOF	2	지연 OFF 상태의 펀칭	+	m		

5.3.37 G 그룹 37: 이송 특성

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
FNORM	1	DIN 66025에 따른 일반 이송 속도	+	m	x	
FLIN	2	직선형 이송	+	m		
FCUB	3	큐빅 스플라인 형 이송	+	m		

5.3.38 G 그룹 38: 펀치/니블링을 위한 고속 입력/출력 지정

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
SPIF1	1	펀칭/니블링 바이트 1을 위한 빠른 NC 입력/출력	+	m	x	
SPIF2	2	펀칭/니블링 바이트 2을 위한 빠른 NC 입력/출력	+	m		

5.3.39 G 그룹 39: 프로그램 형상 정확도

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CPRECOF	1	프로그램 형상 정확도 OFF	+	m	x	
CPRECON	2	프로그램 형상 정확도 ON	+	m		

5.3.40 G 그룹 40: 공구 반경 보상 상수

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CUTCONOF	1	일정 공구 반경 보정 OFF	+	m	x	
CUTCONON	2	일정 공구 반경 보정 ON	+	m		

5.3.41 G 그룹 41: 나사 절삭 중단

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
LFOF	1	나사 절삭 중단 OFF	+	m	x	
LFON	2	나사 절삭 중단 ON	+	m		

5.3.42 G 그룹 42: 공구 케리어

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
TCOABS	1	현재 공구 오리엔테이션에서 공구 길이 구성요소 결정	+	m	x	
TCOFR	2	활성화된 프레임의 오리엔테이션에서 공구 길이 요소 결정	+	m		
TCOFRZ	3	공구 선택 시 활성 프레임의 공구 오리엔테이션을 판정, 공구는 Z방향을 가리킴	+	m		
TCOFRY	4	공구 선택 시 활성 프레임의 공구 오리엔테이션을 판정, 공구는 Y 방향을 가리킴	+	m		
TCOFRX	5	공구 선택 시 활성 프레임의 공구 오리엔테이션을 판정, 공구는 X 방향을 가리킴		m		

5.3.43 G 그룹 43: SAR 접근 방향

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G140	1	G41/G42가 정의한 SAR 접근 방향	+	m	x	
G141	2	형상 좌측으로의 SAR 접근 방향	+	m		
G142	3	형상 우측으로의 SAR 접근 방향	+	m		
G143	4	접선 의존 SAR 접근 방향	+	m		

5.3.44 G 그룹 44: SAR 경로 분할

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G340	1	입체적 접근 블록; 즉, 한 블록에서 절입 깊이 및 면 접근	+	m	x	
G341	2	수직 축(Z)에서 절입으로 시작, 그 후 면에 접근	+	m		

5.3.45 G 그룹 45: FGROUP 축의 경로 기준

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
SPATH	1	FGROUP 축의 경로 기준은 호 길이입니다.	+	m	x	
UPATH	2	FGROUP 축의 경로 기준은 곡선 파라미터입니다.	+	m		

5.3.46 G 그룹 46: 고속 후퇴를 위한 면 선택

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
LFTXT	1	면은 경로 접선 및 현재 공구 오리엔테이션에서 결정됩니다.	+	m	x	
LFWP	2	면은 현재 작업 평면(G17/G18/G19) 에 의해 결정됩니다.	+	m		
LFPOS	3	위치로 축 후진	+	m		

5.3.47 G 그룹 47: 외부 NC 코드 모드로의 전환

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G290	1	SINUMERIK 언어 모드 활성화	+	m	x	
G291	2	ISO 언어 모드 활성화	+	m		

5.3.48 G 그룹 48: 공구 반경 보정을 통한 접근 및 후퇴 반응

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G460	1	접근 및 후퇴 블록에 대한 충돌 탐지 ON	+	m	x	
G461	2	TRC 블록에 교차점이 없는 경우 원호를 사용하여 경계 블록 확장	+	m		
G462	3	TRC 블록에 교차점이 없는 경우 직선을 사용하여 경계 블록 확장	+	m		

5.3.49 G 그룹 49: 지점 간 모션

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
CP	1	경로 모션	+	m	x	
PTP	2	지점 간 모션(동기 축 모션)	+	m		
PTPG0	3	G0만으로 지점 간 모션, 그렇지 않으면 경로 모션 CP	+	m		
PTPWOC	4	오리엔테이션 변화로 인한 보정 모션이 없는 지점간 모션	+	m		

5.3.50 G 그룹 50: 오리엔테이션 프로그래밍

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIEULER	1	오일러 각을 통한 오리엔테이션 각	+	m	x	
ORIRPY	2	RPY각(회전 시퀀스 XYZ)을 통한 오리엔테이션 각	+	m		
ORIVIRT1	3	가상 오리엔테이션 축을 통한 오리엔테이션 각(정의 1)	+	m		
ORIVIRT2	4	가상 오리엔테이션 축을 통한 오리엔테이션 각(정의 2)	+	m		
ORIAXPOS	5	로터리 축 위치의 가상 오리엔테이션 축을 통한 오리엔테이션 각도	+	m		
ORIRPY2	6	RPY각(회전 시퀀스 ZYX)을 통한 오리엔테이션 각	+	m		

5.3.51 G 그룹 51: 오리엔테이션 프로그래밍을 위한 보간 유형

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIVECT	1	대형 원호 보간(ORIPLANE과 동일)	+	m	x	
ORIAXES	2	기계 축 또는 오리엔테이션 축의 직선 보간	+	m		
ORIPATH	3	경로를 기준한 공구 오리엔테이션 궤도	+	m		
ORIPLANE	4	평면의 보간(ORIVECT와 동일)	+	m		
ORICONCW	5	테이퍼 원주면에서 CW 방향으로 보간	+	m		
ORICONCCW	6	테이퍼의 주변 표면에서 시계 반대 방향으로 보간	+	m		
ORICONIO	7	중간 오리엔테이션 설정으로 원추 원주 표면에 보간	+	m		
ORICONTO	8	접선 트렌지션으로 원추의 원주 표면 보간	+	m		
ORICURVE	9	오리엔테이션의 추가적 공간 곡선으로 보간	+	m		
ORIPATHS	10	경로 기준 공구 오리엔테이션. 오리엔테이션 특성에서 소음이 감소됩니다.	+	m		

5.3.52 G 그룹 52: 공작물에 대한 프레임 회전

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
PAROTOF	1	공작물에 대한 프레임 회전 OFF	+	m	x	
PAROT	2	공작물에 대한 프레임 회전 ON 공작물 좌표계가 공작물 위에 정렬됩니다.	+	m		

5.3.53 G 그룹 53: 공구에 대한 프레임 회전

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
TOROTOF	1	공구에 대한 프레임 회전 OFF	+	m	x	
TOROT	2	프레임을 회전시켜 WCS의 Z축을 공구 오리엔테이션과 평행하게 정렬	+	m		
TOROTZ	3	TOROT와 동일	+	m		
TOROTY	4	프레임을 회전시켜 WCS의 Y축을 공구 오리엔테이션과 평행하게 정렬	+	m		
TOROTX	5	프레임을 회전시켜 WCS의 X축을 공구 오리엔테이션과 평행하게 정렬	+	m		
TOFRAME	6	프레임을 회전시켜 WCS의 Z축을 공구 오리엔테이션과 평행하게 정렬	+	m		
TOFRAMEZ	7	TOFRAME과 동일	+	m		
TOFRAMEY	8	프레임을 회전시켜 WCS의 Y축을 공구 오리엔테이션과 평행하게 정렬	+	m		
TOFRAMEX	9	프레임을 회전시켜 WCS의 X축을 공구 오리엔테이션과 평행하게 정렬	+	m		

5.3.54 G 그룹 54: 폴리노미얼 프로그래밍을 위한 벡터 회전

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORIROTA	1	절대 벡터 회전	+	m	x	
ORIROTR	2	상대 벡터 회전	+	m		
ORIROTT	3	접선 벡터 회전	+	m		
ORIROTC	4	경로 접선과 관계된 접선 회전 벡터	+	m		

5.3.55 G 그룹 55: 직선 보간 유무의 급이송

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
RTLION	1	직선 보간을 통한 급 이송 모션 ON	+	m	x	
RTLIOF	2	직선 보간을 통한 급 이송 모션 OFF 급 이송 모션은 단일 축 보간을 통해 이루어집니다.	+	m		

5.3.56 G 그룹 56: 공구 마모 고려

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
TOWSTD	1	공구 길이 옵셋의 초기 설정 값	+	m	x	
TOWMCS	2	기계 좌표계(MCS) 에서 마모 값	+	m		
TOWWCS	3	공작물 좌표계(WCS) 에서 마모값	+	m		
TOWBCS	4	기본 좌표계(BCS) 의 마모값	+	m		
TOWTCS	5	공구 좌표계의 마모값(공구 홀더에서의 공구 홀더 원점 T)	+	m		
TOWKCS	6	좌표계 변환을 위한 공구 헤드 좌표계에서 마모값 (공구 회전을 통한 기계 좌표계와 다름)	+	m		

5.3.57 G 그룹 57: 코너 감속

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
FENDNORM	1	코너 감속 OFF	+	m	x	
G62	2	공구 반경 보정이 활성 상태일 때 내부 코너에서 코너 감속(G41/G42)	+	m		
G621	3	모든 코너에서 코너 감속	+	m		

5.3.58 G 그룹 59: 경로 보간을 위한 동적 응답 모드

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
DYNNORM	1	표준 동적 응답	+	m	x	
DYNPOS	2	포지셔닝 모드, 태핑	+	m		
DYNROUGH	3	황삭	+	m		
DYNSEMIFIN	4	중삭	+	m		
DYNFINISH	5	정삭	+	m		
DYNPREC	6	스무딩 정삭	+	m		

5.3.59 G 그룹 60: 작업 영역 제한

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
WALCS0	1	공작물 좌표계 작업 영역 제한 OFF	+	m	x	
WALCS1	2	WCS 작업 구역 제한 그룹 1의 활성화	+	m		
WALCS2	3	WCS 작업 구역 제한 그룹 2의 활성화	+	m		
WALCS3	4	WCS 작업 구역 제한 그룹 3의 활성화	+	m		
WALCS4	5	WCS 작업 구역 제한 그룹 4의 활성화	+	m		
WALCS5	6	WCS 작업 구역 제한 그룹 5의 활성화	+	m		
WALCS6	7	WCS 작업 구역 제한 그룹 6의 활성화	+	m		
WALCS7	8	WCS 작업 구역 제한 그룹 7의 활성화	+	m		
WALCS8	9	WCS 작업 구역 제한 그룹 8의 활성화	+	m		
WALCS9	10	WCS 작업 구역 제한 그룹 9의 활성화	+	m		
WALCS10	11	WCS 작업 구역 제한 그룹 10의 활성화	+	m		

5.3.60 G 그룹 61: 공구 오리엔테이션 스무딩

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
ORISOF	1	공구 오리엔테이션 스무딩 OFF	+	m	x	
ORISON	2	공구 오리엔테이션 스무딩 ON	+	m		

5.3.61 G 그룹 62: REPOS 리포지셔닝 모드(넌모달)

G 명령	번호 [1]	의미	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
RMBBL	1	블록 시작으로 리포지셔닝	-	s		
RMIBL	2	인터럽트 지점으로 리포지셔닝	-	s	x	
RMEBL	3	블록 끝으로 리포지셔닝	-	s		
RMNBL	4	가장 가까운 경로 지점으로 리포지셔닝	-	s		

5.3.62 G 그룹 64: 연삭 프레임

G 명령	번호 [1]	의미 **채널 $P_GFRAME =에서의 활성 연삭 프레임**	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
GFRAME[0]	1	데이터 관리 $P_GFR[0] (프레임 없음)의 연삭 프레임	+	m	x	
GFRAME[1]	2	데이터 관리 $P_GFR[1]의 연삭 프레임	+	m		
GFRAME[2]	3	데이터 관리 $P_GFR[2]의 연삭 프레임	+	m		
...	...		+	m		
GFRAME[100]	101	데이터 관리 $P_GFR[100]의 연삭 프레임	+	m		

범례

1) PLC 인터페이스의 내부 번호

2) 가공 프로그램의 가동, 리셋, 또는 종료 시 G 코드를 G 그룹에 대한 리셋 설정으로 구성할 수 있는지 여부 (MD20150 $MC_GCODE_RESET_VALUES 사용):

   \+ 구성 가능

   \- 구성 불가능

3) G 코드의 효과:

   m 모달

   s 넌모달

4) 리셋 설정, 다음 머신 데이터 참조

   - MD20149GCODE_RESET_S_VALUES (G 그룹의 리셋 위치 (고정))
   - MD20150 $MC_GCODE_RESET_VALUES (G 그룹의 재설정 위치)
   - MD20151GCODE_RESET_S_MODE (G 그룹의 리셋 동작 (고정))
   - MD20152 $MC_GCODE_RESET_MODE (G 그룹의 재설정 행동)
   - MD20154 $MC_EXTERN_GCODE_RESET_VALUES (ISO 모드 상의 G 그룹의 재설정 위치)
   - MD20156 $MC_EXTERN_GCODE_RESET_MODE (외부 G 그룹의 재설정 행동)

   SAG **S**iemens **AG** 표준 설정

   MM **장비 제조업체** 표준 설정(장비 제조업체 사양 참조)

5.4 사전 정의된 절차

사전 정의된 절차를 호출하면 사전 정의된 NC 기능이 실행됩니다. 사전 정의된 기능과 달리 사전 정의된 절차는 반환값을 제공하지 **않습니다**.

좌표계					
식별자	**파라미터**				**설명**
	1번째	**2번째**	**3번째 - 15번째**	**4번째 - 16번째**	

좌표계					
식별자	**파라미터**				**설명**
PRESETON	AXIS *) 기계 축의 축 이름	REAL: 사전 설정된 옵셋 G700/G710 컨텍스트	1 ...로	2 ...로	원점 복귀 상태를 삭제하고 프로그램된 축에 실제 값 설정
PRESETONS	AXIS *) 기계 축의 축 이름	REAL: 사전 설정된 옵셋 G700/G710 컨텍스트	1 ...로	2 ...로	원점 복귀 상태를 유지하면서 프로그램된 축에 실제 값 설정
DRFOF					채널에 할당된 모든 축의 DRF 옵셋을 삭제

) 기본적으로 기준이 명확하다면 기하 축 또는 특수 축 이름을 기계 축 이름 대신 사용할 수 있습니다.

축 그룹화					
식별자	**파라미터**				**설명**
GEOAX	**1번째**	**2번째**	**3번째 / 5번째**	**4번째 / 6번째**	평행 좌표계 선택
	INT: 기하 축 번호 1 ... 3	AXIS: 채널 축 이름	1로	2로	
FGROUP	**1번째 – 8번째**				변수 F 값 기준: 경로 이송 속도의 기준이 되는 축 정의 최대 축 개수: 8 F 값 기준의 기본 설정은 파라미터 없이 FGROUP ( )으로 활성화됩니다.
	AXIS: 채널 축 이름				

축 그룹화				
식별자	**파라미터**			**설명**
SPLINEPATH	**1번째**	**2번째 – 9번째**		스플라인 그룹화 정의 최대 축 개수: 8
	INT: 스플라인 그룹화 (반드시 1)	AXIS: 추가 기하 축 이름		
POLYPATH	**1번째**	**2번째**		선택한 축 그룹에 폴리노미얼 보간 활성화
	STRING	STRING		

커플 모션							
식별자	**파라미터**						**설명**
	1번째	**2번째**	**3번째**	**4번째**	**5번째**	**6번째**	
TANG	AXIS: 축 이름 추종 축	AXIS: 리딩 축 1	AXIS: 리딩 축 2	REAL: 커플링 팩터	CHAR: 옵션: "B": BCS에서 추적 "W": WCS에서 추적	CHAR 최적화: "S": 표준 "P": 라운딩 공차 및 각도 공차 자동 적용	접선 제어: 커플링 정의 추적의 접선은 지정된 두 마스터 축에 의해 결정됩니다. 커플링 팩터는 접선 각의 변경과 추종 축 사이의 관계를 지정합니다. 이는 보통 1입니다.
TANGON	AXIS: 축 이름 추종 축	REAL: 옵셋 각도	REAL: 라운딩 공차	REAL: 각도 공차			접선 제어: 커플링 활성화
TANGOF	AXIS: 축 이름 추종 축						접선 제어: 커플링 비활성화
TLIFT	AXIS: 추적 대상 축						접선 제어: 중간 블록 생성 활성화
TRAILON	AXIS: 추종 축	AXIS: 리딩 축	REAL: 커플링 팩터				비동기 커플 모션 OFF

커플 모션							
식별자	파라미터						설명
	1번째	2번째	3번째	4번째	5번째	6번째	
TRAILOF	AXIS: 추종 축	AXIS: 리딩 축					결합 축 동작 비활성화
TANGDEL	AXIS: 추종 축						접선 제어: 커플링 삭제

커브 테이블						
식별자	파라미터					설명
	1번째	2번째	3번째	4번째	5번째	
CTABDEF	AXIS: 추종 축	AXIS: 리딩 축	INT: 테이블 번호	INT: 정의된 범위 끝에서의 동작	STRING: 저장 위치 지정	테이블 정의 ON 추종 모션 블록이 커브 테이블 결정
CTABEND	AXIS: 추종 축	AXIS: 리딩 축	INT: 테이블 번호	INT: 정의된 범위 끝에서의 동작		테이블 정의 OFF
CTABDEL	INT: 테이블 번호 n	INT: 테이블 번호 m	STRING: 저장 위치 지정			커브 테이블 삭제
CTABLOCK	INT: 테이블 번호 n					커브 테이블을 번호 n으로 잠궈 이 테이블을 삭제하거나 덮어쓰지 못하게 방지.
CTABUNLOCK	INT: 테이블 번호 n					CTABLOCK으로 보호된 테이블 번호가 n인 테이블을 다시 해제
LEADON	AXIS: 추종 축	AXIS: 리딩 축	INT: 테이블 번호			마스터 값 커플링 ON
LEADOF	AXIS: 추종 축	AXIS: 리딩 축				마스터 값 커플링 OFF

축 가속 특성		
식별자	**파라미터**	**설명**
	1번째 – 8번째	
BRISKA	AXIS	프로그램된 축의 단계별 축 가속 활성화
SOFTA	AXIS	프로그램된 축의 저크 제한 축 가속 활성화
DRIVEA	AXIS	프로그램된 축의 니(knee) 형태 가속 특성 활성화
JERKA	AXIS	프로그램된 축에 머신 데이터 $MA_AX_JERK_ENABLE에 설정된 가속 특성을 활성화

회전 당 이송 속도			
식별자	**파라미터**		**설명**
FPRAON	**1번째**	**2번째**	축 회전 당 이송 속도 ON
	AXIS: 회전 당 이송 속도가 활성화된 축	AXIS: 회전 당 이송 속도가 도출되는 축/스핀들 프로그램된 축이 없다면 회전 당 이송 속도는 마스터 스핀들에서 도출됩니다.	
FPRAOF	**1번째 - n번째**		축 회전 당 이송 속도 OFF 회전 당 이송 속도는 여러 축에서 동시에 비활성화할 수 있습니다. 한 블록 내 허용된 축의 수 만큼 프로그램할 수 있습니다.
	AXIS: 회전 당 이송 속도가 비활성화된 축		

회전 당 이송 속도			
식별자	**파라미터**		**설명**
FPR	**1번째**		G95를 위해 경로의 회전 당 이송 속도를 도출하는 로터리 축 또는 스핀들 선택. FPR로 지정한 설정은 모달로 적용됩니다.
	AXIS: 회전 당 이송 속도가 도출되는 축/스핀들 프로그램된 축이 없다면 회전 당 이송 속도는 마스터 스핀들에서 도출됩니다.		

변환				
식별자	**파라미터**			**설명**
	1번째	**2번째**	**3번째**	
TRACYL	REAL: 작업 직경	INT: 변환 번호		원통: 원주면 변환 한 채널에 여러 변환을 설정할 수 있습니다. 변환 번호는 활성화될 변환을 지정합니다. 두 번째 파라미터가 생략되는 경우, MD를 통한 변환 그룹 설정이 활성화됩니다.
TRANSMIT	INT: 변환 번호			전송: 극 변환 한 채널에 여러 변환을 설정할 수 있습니다. 변환 번호는 활성화될 변환을 지정합니다. 파라미터가 생략되었다면 MD에서 정의된 변환 그룹이 활성화됩니다.
TRAANG	REAL: 각도	INT: 변환 번호		경사 축 변환 한 채널에 여러 변환을 설정할 수 있습니다. 변환 번호는 활성화될 변환을 지정합니다. 두 번째 파라미터가 생략되는 경우, MD를 통한 변환 그룹 설정이 활성화됩니다. 각도가 프로그램되지 않았다면 (TRAANG (,2) 또는 TRAANG) 마지막 각도가 모달로 적용됩니다.
TRAORI	INT: 변환 번호			4축/5축 변환 한 채널에 여러 변환을 설정할 수 있습니다. 변환 번호는 활성화될 변환을 지정합니다.

변환				
식별자	**파라미터**			**설명**
	1번째	**2번째**	**3번째**	
TRACON	INT: 변환 번호	REAL: 추가 파라미터, MD-의존		계단식 변환 파라미터의 의미는 케스케이딩 유형에 따라 달라집니다.
TRAFOOF				변환 비활성화
TRAFOON	STRING: 변환 데이터 세트 이름	REAL: 기준 또는 작업 직경 (TRACYL만 해당)	BOOL: 홈 측면 옵셋 포함/미포함 (TRACYL만 해당)	키네마틱 체인을 이용해 지정된 변환 활성화

스핀들			
식별자	**파라미터**		**설명**
	1	**2번째 - n번째**	
SPCON	INT: 스핀들 번호	INT: 스핀들 번호	위치 제어 상태의 스핀들 조작으로 전환
SPCOF	INT: 스핀들 번호	INT: 스핀들 번호	속도 제어 상태의 스핀들 조작으로 전환
SETMS	INT: 스핀들 번호		이 스핀들을 현재 채널의 마스터 스핀들로 선언 SETMS( )의 경우 별도의 파라미터 설정 없이 머신 데이터의 기본 설정이 자동으로 적용됩니다.

연삭		
식별자	**파라미터**	**설명**
	1번째	
GWPSON	INT: 스핀들 번호	연삭 휠 원주 속도 일정 제어 ON 스핀들 번호가 프로그램되지 않은 경우 활성 공구의 스핀들에 연삭 휠 원주 속도가 선택됩니다.
GWPSOF	INT: 스핀들 번호	연삭 휠 원주 속도 일정 제어 OFF 스핀들 번호가 프로그램되지 않은 경우 활성 공구의 스핀들에 연삭 휠 원주 속도가 선택 해제됩니다.

연삭		
식별자	**파라미터**	**설명**
	1번째	
TMON	INT: T 번호	연삭 공구 모니터링 ON T 번호가 프로그램되지 않은 경우 활성 공구에 대한 모니터링이 활성화됩니다.
TMOF	INT: T 번호	공구 모니터링 OFF T 번호가 프로그램되지 않은 경우 활성 공구에 대한 모니터링이 비활성화됩니다.

스톡 제거					
식별자	**파라미터**				**설명**
	1번째	**2번째**	**3번째**	**4번째**	
CONTPRON	REAL [,11]: 형상 테이블	CHAR: 가공 형식	INT: 릴리프 절삭 수	INT: 계산 상태	기준 전처리 활성화 다음 단계에서 호출되는 형상 프로그램이나 NC 블록은 개별 이동으로 나뉘고 형상 테이블에 저장됩니다. 릴리프 절삭 번호가 반환됩니다.
CONTDCON	REAL [, 6]: 형상 테이블	INT: 가공 방향			형상 디코딩 형상을 위한 블록은 블록 당 하나의 테이블 라인으로 명명된 테이블에 저장되고 메모리에 저장하기 위해 암호화됩니다.
EXECUTE	INT: 에러 상태				프로그램 실행 활성화 이는 기준점 편집 모드에서 또는 보호 영역 설정 후 정상 프로그램 실행으로 다시 전환합니다.

테이블 실행		
식별자	**파라미터**	**설명**
	1번째	
EXECTAB	REAL [11]: 모션 테이블의 요소	모션 테이블의 요소 실행

보호 영역						
식별자	**파라미터**					**설명**
	1번째	**2번째**	**3번째**	**4번째**	**5번째**	
CPROTDEF	INT: 보호 영역의 번호	BOOL: TRUE: 공구별 보호 영역	INT: 0: 4번째 및 5번째 파라미터가 평가되지 않습니다. 1: 4번째 파라미터를 평가합니다. 2: 5번째 파라미터를 평가합니다. 3: 4번째 및 5번째 파라미터를 평가합니다.	REAL: 플러스 방향으로 제한	REAL: 마이너스 방향으로 제한	채널 특정 보호 영역 정의

보호 영역						
식별자	파라미터					설명
	1번째	2번째	3번째	4번째	5번째	
NPROTDEF	INT: 보호 영역의 번호	BOOL: TRUE: 공구별 보호 영역	INT: 0: 4번째 및 5번째 파라미터가 평가되지 않습니다. 1: 4번째 파라미터를 평가합니다. 2: 5번째 파라미터를 평가합니다. 3: 4번째 및 5번째 파라미터를 평가합니다.	REAL: 플러스 방향으로 제한	REAL: 마이너스 방향으로 제한	기계 특정 보호 영역 정의

보호 영역						
식별자	**파라미터**					**설명**
	1번째	**2번째**	**3번째**	**4번째**	**5번째**	
CPROT	INT: 보호 영역의 번호	INT: 옵션 0: 보호 영역 OFF 1: 보호 영역 사전 활성화 2: 보호 영역 ON 3: 보호 영역이 활성인 경우에만 조건부 정지로 보호 영역을 예비 활성화	REAL: 첫 번째 기하 축의 보호 영역 옵셋	REAL: 두 번째 기하 축의 보호 영역 옵셋	REAL: 세 번째 기하 축의 보호 영역 옵셋	채널 특정 보호 영역 ON/OFF
NPROT	INT: 보호 영역의 번호	INT: 옵션 0: 보호 영역 OFF 1: 보호 영역 사전 활성화 2: 보호 영역 ON 3: 보호 영역이 활성인 경우에만 조건부 정지로 보호 영역을 예비 활성화	REAL: 첫 번째 기하 축의 보호 영역 옵셋	REAL: 두 번째 기하 축의 보호 영역 옵셋	REAL: 세 번째 기하 축의 보호 영역 옵셋	기계 특정 보호 영역 ON/OFF

전처리/싱글 블록		
식별자	**파라미터**	**설명**
STOPRE		메인 실행에서 준비된 블록이 모두 실행될 때까지 전처리 정지
SBLOF		싱글 블록 처리 억제
SBLON		싱글 블록 처리 억제 취소

인터럽트		
식별자	**파라미터**	**설명**
	1번째	
DISABLE	INT: 인터럽트 입력 번호	지정된 하드웨어 입력에 지정된 인터럽트 루틴 비활성화. 고속 후퇴가 실행되지 않습니다. 하드웨어 입력과 SETINT로 만들어진 인터럽트 루틴 사이의 할당은 유효하게 유지되며 ENABLE로 다시 활성화될 수 있습니다.
ENABLE	INT: 인터럽트 입력 번호	DISABLE로 비활성화된 인터럽트 루틴 지정을 재활성화
CLRINT	INT: 인터럽트 입력 번호	인터럽트 입력에 지정된 인터럽트 루틴 및 속성 삭제. 인터럽트 루틴이 비활성화되고 중단이 발생하여도 반응은 나타나지 않습니다.

동기 동작		
식별자	**파라미터**	**설명**
	1번째 - n번째	
CANCEL	INT: 동기 동작의 번호	지정된 ID를 가진 모달 동기 동작을 중단합니다. 콤마 (,) 를 사용하여 여러 ID를 지정할 수 있습니다.
CANCELSUB		현재 서브프로그램 레벨 취소

기능 정의					
식별자	**파라미터**				**설명**
	1번째	**2번째**	**3번째**	**4번째-7번째**	
FCTDEF	INT: 기능 번호	REAL: 하한값	REAL: 상한값	REAL: 계수 a0~a3	폴리노미얼 기능 정의 이는 SYFCT 또는 PUTFTOCF 에서 평가됩니다.

통신						
식별자	**파라미터**					**설명**
	1번째	**2번째**				
MMC	STRING: 명령	CHAR: 승인 모드*) "N": 승인 안 함 "S": 동기 승인 'A': 비동기 승인				NC 프로그램을 사용한 Windows 환경설정에 대한 HMI 명령어 해석을 위한 명령
	1번째	**2번째**	**3번째**	**4번째**	**5번째**	
DRVPRD	VAR INT: 상태	AXIS: 기계 축 이름	INT: 드라이브 파라미터의 번호	INT: 드라이브 파라미터의 인덱스	VAR REAL: 드라이브 파라미터의 읽기 값	가공 프로그램 레벨에서 드라이브 파라미터 읽기
DRVPWR	VAR INT: 상태	AXIS: 기계 축 이름	INT: 드라이브 파라미터의 번호	INT: 드라이브 파라미터의 인덱스	REAL: 쓸 드라이브 파라미터 값	가공 프로그램 레벨에서 드라이브 파라미터 쓰기

) 명령은 실행 콤포넌트 (채널, NC 등) 가 요청 시 승인됩니다.

프로그램 코디네이션					
식별자	**파라미터**				**설명**
INIT	**1번째**	**2번째**	**3번째**		채널에서 실행을 위한 NC 프로그램 선택
	INT: 채널 번호 **또는** MD20000에 지정된 채널 이름 *)	STRING: 경로 지정	CHAR: 승인 모드 **)		
	1번째 - n번째				
START	INT: 채널 번호 **또는** MD20000에 지정된 채널 이름 *)				현재 프로그램의 여러 채널에서 선택된 프로그램을 동시에 시작 이 명령은 해당 채널에는 영향을 주지 않습니다.
WAITE	INT: 채널 번호 **또는** MD20000에 지정된 채널 이름 *)				하나 이상의 다른 채널에서 프로그램 종료 대기
	1번째	**2번째 - n번째**			
WAITM	INT: 마커 번호	INT: 채널 번호 **또는** MD20000에 지정된 채널 이름 *)			지정된 채널에서 마커에 도달할 때까지 대기 이전 블록은 정위치 정지로 종료됩니다.
WAITMC	INT: 마커 번호	INT: 채널 번호 **또는** MD20000에 지정된 채널 이름 *)			지정된 채널에서 마커에 도달할 때까지 대기 정위치 정지는 다른 채널이 아직 마커에 도달하지 않은 경우에만 시작됩니다.
	1번째 - n번째				

프로그램 코디네이션		
식별자	**파라미터**	**설명**
SETM	INT: 마커 번호	채널 코디네이션에 하나 이상의 마커 설정 자체 채널 내의 처리는 이 명령의 영향을 받지 않습니다.
CLEARM	INT: 마커 번호	채널 코디네이션을 위해 하나 이상의 마커 삭제 자체 채널 내의 처리는 이 명령의 영향을 받지 않습니다.
	1번째 - n번째	
WAITP	AXIS: 축 이름	이전에 POSA로 프로그램한 포지셔닝 축이 프로그램된 종점에 도달할 때까지 대기
WAITS	INT: 스핀들 번호	이전에 SPOSA로 프로그램한 스핀들이 프로그램된 종점에 도달할 때까지 대기

프로그램 코디네이션					
식별자	**파라미터**				**설명**
RET	**1번째**	**2번째**	**3번째**	**4번째**	PLC에 기능을 출력하지 않고 서브프로그램 종료 첫 번째 파라미터 (점프 대상) 을 지정하면, 리턴 점프가 호출 블록 이후의 블록으로 먼저 실행됩니다. 그 다음, 프로그래밍별로(RET 또는 RETB) 다음 전략에 따라 대상을 추적합니다. • RET: 프로그램의 종료 방향으로 검색합니다. 이 검색에 성공하지 못한 경우, 프로그램의 시작을 향하여 검색이 진행됩니다. • RETB: 프로그램의 시작 방향으로 검색합니다. 검색이 실패하면 프로그램 끝 방향으로 검색이 다시 진행됩니다.
	INT (또는 STRING): 리턴할 점프 대상 (블록 번호/마커)	INT: 0: 1번째 파라미터에서 점프 대상으로 리턴 점프 > 0: 다음 블록으로 리턴	INT: 건너뛸 서브프로그램 레벨 개수	BOOL: 메인 프로그램의 첫 번째 블록으로 리턴	
RETB	INT (또는 STRING): 리턴할 점프 대상 (블록 번호/마커)	INT: 0: 1번째 파라미터에서 점프 대상으로 리턴 점프 > 0: 다음 블록으로 리턴	INT: 건너뛸 서브프로그램 레벨 개수	BOOL: 메인 프로그램의 첫 번째 블록으로 리턴	
	1번째 - n번째				
GET	AXIS: 축 이름 ***)				기계 축 지정 지정된 축들은 RELEASE를 이용해 다른 채널에서 분리해야 합니다.
GETD	AXIS: 축 이름 ***)				기계 축 직접 할당 지정된 축을 RELEASE로 분리해서는 **안됩니다**.
RELEASE	AXIS: 축 이름 ***)				기계 축 분리
	1번째	**2번째**	**3번째**	**4번째**	

프로그램 코디네이션					
식별자	**파라미터**				**설명**
PUTFTOC	REAL: 옵셋 값	INT: 파라미터 번호	INT: 채널 번호 **또는** MD20000에 지정된 채널 이름 *)	INT: 스핀들 번호	미세 공구 보정 변경
PUTFTOCF	INT: 기능 번호	VAR REAL: 기준 값	INT: 파라미터 수	INT: 채널 번호 **또는** MD20000에 지정된 채널 이름 *)	FCTDEF로 정의한 기능 (최대 3차 폴리노미얼) 에 따라 미세 공구 보정 변경 여기에 사용되는 번호는 반드시 FCTDEF에 지정해야 합니다.
AXTOCHAN	**1번째**	**2번째**	**3번째 - n번째**	**4번째 - m번째**	다른 채널로 이송된 축
	AXIS: 축 이름	INT: 채널 번호 **또는** MD20000에 지정된 채널 이름 *)	1 ...로	2 ...로	

) 채널 번호 대신 MD20000 $MC_CHAN_NAME에 정의된 채널 이름을 프로그래밍할 수도 있습니다.

**) 명령은 실행 콤포넌트 (채널, NC 등) 가 요청 시 승인됩니다.

***) SPI 함수 역시 축 대신에 스핀들을 프로그램하는데 사용될 수 있습니다. 예: GET(SPI(1))

데이터 액세스		
식별자	**파라미터**	**설명**
CHANDATA	**1번째** INT: 채널 번호	채널 데이터 액세스를 위한 채널 번호 설정(초기화 블록에서만 허용됨). 후속 액세스는 CHANDATA로 설정된 채널을 참조합니다.
NEWCONF		변경된 머신 데이터 적용

메시지			
식별자	**파라미터**		**설명**
	1번째	**2번째**	
MSG	STRING: 메시지	INT: 실행	사용자 인터페이스에 임의의 문자열을 메시지로 출력
WRTPR	STRING: 문자열	INT: 실행	OPI 변수에 문자열 작성

파일 액세스						
식별자	**파라미터**					**설명**
READ	**1번째**	**2번째**	**3번째**	**4번째**	**5번째**	파일 시스템에서 블록 읽기
	VAR INT: 에러	CHAR[160]: 파일 이름	INT: 읽어올 파일 섹션 시작 라인	INT: 읽어올 라인 개수	VAR CHAR[255]: 읽기 정보가 저장되는 변수 배열	

파일 액세스						
식별자	**파라미터**					**설명**
WRITE	**1번째**	**2번째**	**3번째**	**4번째**		파일 시스템(또는 외부 장치/파일)에 블록 쓰기
	VAR INT: 에러	CHAR[160]: 파일 이름	STRING: 외부 출력을 위한 장치/파일	CHAR[200]: 블록		
DELETE	**1번째**	**2번째**				파일 삭제
	VAR INT: 에러	CHAR[160]: 파일 이름				

알람			
식별자	**파라미터**		**설명**
	1번째	**2번째**	
SETAL	INT: 알람 번호 (싸이클 알람)	STRING: 문자열	알람 설정 4 개의 파라미터까지의 문자열은 알람 번호에 추가하여 지정될 수 있습니다. 다음 사전 정의된 파라미터를 사용할 수 있습니다: %1 = 채널 번호 %2 = 블록 번호, 라벨 %3 = 경고 싸이클의 텍스트 색인 %4 = 추가 경고 파라미터

공구 관리				
식별자	**파라미터**			**설명**
	1번째	**2번째**		
DELDL	INT: T 번호	INT: D 번호		절삭날 또는 D 가 지정되지 않은 공구의 모든 추가 옵셋 삭제

공구 관리							
식별자	**파라미터**						**설명**
DELT	STRING[32]: 공구 식별자	INT: 듀플로 번호					공구 삭제 듀플로 번호는 생략 가능
DELTC	INT: 데이터 세트 번호 n	INT: 데이터 세트 번호 m					공구 캐리어 데이터 세트 번호 n~m 삭제
DZERO							채널에 할당된 TO 유닛의 모든 공구 D 번호를 무효로 설정
	1번째	**2번째**	**3번째**	**4번째**	**5번째**	**6번째**	
GETFREELOC	VAR INT: 매거진 번호 (반환값)	VAR INT: 포켓 번호 (반환값)	INT: T 번호	INT: 기준 매거진 번호	CHAR: 4번째 파라미터에 따라 지정	INT: 예약 모드	공구를 위해 빈 포켓 찾기
	1번째	**2번째**					
GETSELT	VAR INT: T 번호 (반환값)	INT: 스핀들 번호					스핀들에 사전 선택된 공구의 T 번호 반환
GETEXET	VAR INT: T 번호 (반환값)	INT: 스핀들 번호					NC 프로그램 입장에서 활성인 공구의 T 번호 반환
GETTENV	STRING: 공구 환경의 이름	INT ARRAY[3]: 반환값					공구 환경에 저장된 T, D 및 DL 번호 읽기
	1번째	**2번째**	**3번째**	**4번째**			
POSM	INT: 포지셔닝을 위한 포켓 번호	INT: 이동할 매거진 번호	INT: 내부 매거진의 포켓 번호	INT: 내부 매거진의 매거진 번호			매거진 포지셔닝

공구 관리						
식별자	**파라미터**					**설명**
RESETMON	VAR INT: 상태 = 작업 결과 (반환 값)	INT: 내부 T 번호	INT: 공구의 D 번호	INT: 비트 코딩 파라미터 (옵션)		공구의 실제 값을 지령치로 설정
SETDNO	**1번째**	**2번째**	**3번째**			공구 (T) 절삭날의 옵셋 번호 (D) 지정
	INT: T 번호	INT: 절삭날 번호	INT: D 번호			
SETMTH	**1번째**					공구 캐리어 번호 설정
	INT: 공구 캐리어 번호					
SETPIECE	**1번째**	**2번째**				스핀들의 소재 계수기 감소 가공 과정과 관련된 공구들의 계수 모니터링 데이터 업데이트
	INT: 계수 감소에 사용된 값	INT: 스핀들 번호				
	1번째	**2번째**	**3번째**	**4번째**		
SETTA	VAR INT: 상태 = 작업 결과 (반환 값)	INT: 매거진 번호	INT: 마모 그룹 번호	INT: 공구 하위 그룹		마모 그룹에서 공구 활성화
SETTIA	VAR INT: 상태 = 작업 결과 (반환 값)	INT: 매거진 번호	INT: 마모 그룹 번호	INT: 공구 하위 그룹		마모 그룹에서 공구 비활성화

공구 관리							
식별자	**파라미터**						**설명**
TCA	**1번째**	**2번째**	**3번째**				공구 상태와 관계 없이 공구 선택/공구 교환
	STRING[32]: 공구 식별자	INT: 듀플로 번호	INT: 공구 캐리어 번호				
TCI	**1번째**	**2번째**					버퍼에서 매거진으로 공구 로드
	INT: 버퍼 번호	INT: 공구 캐리어 번호					
MVTOOL	**1번째**	**2번째**	**3번째**	**4번째**	**5번째**		공구 이동을 위한 언어 명령
	INT: 상태	INT: 매거진 번호	INT: 포켓 번호	INT: 이동 후 매거진 번호	INT: 이동 후 대상 포켓 번호		

공구 오리엔테이션				
식별자	**파라미터**			**설명**
	1번째	**2번째**	**3번째**	
ORIRESET	REAL: 첫 번째 기하축 초기 설정	REAL: 두 번째 기하축 초기 설정	REAL: 세 번째 기하축 초기 설정	공구 오리엔테이션 초기 설정

동기 스핀들							
식별자	**파라미터**						**설명**
	1번째	**2번째**	**3번째**	**4번째**	**5번째**	**6번째**	
COUPDEF	AXIS: 추종 스핀들	AXIS: 리딩 스핀들	REAL: 전송 비율의 분자	REAL: 전송 비율의 분모	STRING[8]: 블록 변경 동작	STRING[2]: 커플링 형식	동기 스핀들 그룹화 정의
COUPDEL	AXIS: 추종 스핀들	AXIS: 리딩 스핀들					동기 스핀들 그룹화 삭제
COUPRES	AXIS: 추종 스핀들	AXIS: 리딩 스핀들					설정된 MD 및 SD 값으로 커플링 파라미터 리셋
COUPON	AXIS: 추종 스핀들	AXIS: 리딩 스핀들	REAL: 추종 스핀들의 실행 위치				동기 스핀들 커플링 실행 추종 스핀들에 실행 위치가 지정(양의 회전 방향에서 LS의 0° 위치를 기준(절대치 또는 증분치)으로 하는 FS와 LS 사이의 각도 오정렬)된 경우 커플링은 지정된 위치를 통과해야만 실행됩니다.
COUPONC	AXIS: 추종 스핀들	AXIS: 리딩 스핀들					동기 스핀들 커플링 실행 COUPONC의 경우 커플링이 실행되면 추종 스핀들의 현재 활성 속도를 가져옵니다 (M3/M4 S..).

동기 스핀들							
식별자	**파라미터**						**설명**
	1번째	**2번째**	**3번째**	**4번째**	**5번째**	**6번째**	
COUPOF	AXIS: 추종 스핀들	AXIS: 리딩 스핀들	REAL: 추종 스핀들의 해제 위치 (절대치)	REAL: 리딩 스핀들의 해제 위치 (절대치)			동기 스핀들 커플링 해제 위치가 지정되면 지정된 모든 위치를 통과해야만 커플링이 취소됩니다. 추종 스핀들은 커플링 비활성화 이전에 프로그램된 최종 속도로 회전을 계속합니다.
COUPOFS	AXIS: 추종 스핀들	AXIS: 리딩 스핀들	REAL: 추종 스핀들의 해제 위치 (절대치)				추종 스핀들을 정지시켜 동기 스핀들 커플링 해제 위치가 지정되면 지정된 위치를 통과해야만 커플링이 취소됩니다.
WAITC	AXIS: 추종 스핀들	STRING[8]: 블록 변경 동작	AXIS: 추종 스핀들	STRING[8]: 블록 변경 동작			스핀들 (최대 2개)의 커플링 블록 변경 기준을 충족할 때까지 대기 블록 변경 동작이 지정되지 않은 경우 COUPDEF로 정의한 블록 변경 동작이 적용됩니다.

전자 기어							
식별자	**파라미터**						**설명**
EGDEL	**1번째**						추종 축에 대한 커플링 정의 삭제
	AXIS: 추종 축						
EGDEF	**1번째**	**2번째 / 4번째 / 6번째 / 8번째 / 10번째**	**3번째 / 5번째 / 7번째 / 9번째 / 11번째**				전자 기어 정의
	AXIS: 추종 축	AXIS: 리딩 축	INT: 커플링 형식				
EGON	**1번째**	**2번째**	**3번째 / 6번째 / 9번째 / 12번째 / 15번째**	**4번째 / 7번째 / 10번째 / 13번째 / 16번째**	**5번째 / 8번째 / 11번째 / 14번째 / 17번째**		동기화 없이 전자 기어 ON
	AXIS: 추종 축	STRING: 블록 변경 동작	AXIS: 리딩 축	REAL: 커플링 팩터의 분자	REAL: 커플링 팩터의 분모		

전자 기어									
식별자	**파라미터**								**설명**
EGONSYN	**1번째**	**2번째**	**3번째**	**4번째 / 8번째 / 12번째 / 16번째 / 20번째**	**5번째 / 9번째 / 13번째 / 17번째 / 21번째**	**6번째 / 10번째 / 14번째 / 18번째 / 22번째**	**7번째 / 11번째 / 15번째 / 19번째 / 23번째**		동기화와 함께 전자 기어 ON
	AXIS: 추종 축	STRING: 블록 변경 동작	REAL: 추종 축의 동기 위치	AXIS: 리딩 축	REAL: 리딩 축의 동기 위치	REAL: 커플링 팩터의 분자	REAL: 커플링 팩터의 분모		
EGONSYNE	**1번째**	**2번째**	**3번째**	**4번째**	**5번째 / 9번째 / 13번째 / 17번째 / 21번째**	**6번째 / 10번째 / 14번째 / 18번째 / 22번째**	**7번째 / 11번째 / 15번째 / 19번째 / 23번째**	**8번째 / 12번째 / 16번째 / 20번째 / 24번째**	동기화와 함께 전자 기어 ON 및 접근 모드 지정
	AXIS: 추종 축	STRING: 블록 변경 동작	REAL: 추종 축의 동기 위치	STRING: 접근 모드	AXIS: 리딩 축	REAL: 리딩 축의 동기 위치	REAL: 커플링 팩터의 분자	REAL: 커플링 팩터의 분모	
EGOFS	**1번째**	**2번째 - n번째**							선택적으로 전자 기어 OFF
	AXIS: 추종 축	AXIS: 리딩 축							
EGOFC	**1번째**								전자 기어 OFF (스핀들만 해당)
	AXIS: 추종 스핀들								

니블링					
식별자	**파라미터**				**설명**
	1번째	**2번째**	**3번째**	**4번째**	
PUNCHAAC	REAL: 최소 홀 간격	REAL: 초기 가속	REAL: 최대 홀 간격	REAL: 최종 가속	이송에 따라 가속 활성화

패시브 파일 시스템의 정보 기능				
식별자	**파라미터**			**설명**
	1번째	**2번째**	**3번째**	
FILEDATE	VAR INT: 에러 메시지	CHAR[160]: 파일 이름	VAR CHAR[8]: "dd.mm.yy" 형식의 날짜	파일에 마지막 쓰기 액세스한 날짜 반환
FILETIME	VAR INT: 에러 메시지	CHAR[160]: 파일 이름	VAR CHAR[8]: "hh.mm.ss" 형식의 시간	파일에 마지막 쓰기 액세스한 시간 반환
FILESIZE	VAR INT: 에러 메시지	CHAR[160]: 파일 이름	VAR INT: 파일 크기	파일의 현재 크기 반환
FILESTAT	VAR INT: 에러 메시지	CHAR[160]: 파일 이름	VAR CHAR[5]: "rwxsd" 형식의 날짜	다음 권한을 기준으로 파일의 상태 반환 • 읽기 (r: read) • 쓰기 (w: write) • 실행 (x: execute) • 표시 (s: show) • 삭제 (d: delete)
FILEINFO	VAR INT: 에러 메시지	CHAR[160]: 파일 이름	VAR CHAR[32]: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss" 형식의 날짜	FILEDATE, FILETIME, FILESIZE 및 FILESTAT으로 리드아웃할 수 있는 파일의 모든 정보 반환

축 컨테이너		
식별자	**파라미터** **1번째 - n번째**	**설명**
AXCTSWE	AXIS: 축 컨테이너	축 컨테이너 회전
AXCTSWED	AXIS: 축 컨테이너	축 컨테이너 회전 (스타트업을 위한 명령 변용)
AXCTSWEC:	AXIS: 축 컨테이너	축 컨테이너 회전 활성화 취소

마스터/슬레이브 커플링		
식별자	**파라미터** **1번째 - n번째**	**설명**
MASLON	AXIS: 축 이름	마스터/슬레이브 커플링 실행
MASLOF	AXIS: 축 이름	마스터/슬레이브 커플링 분리
MASLOFS	AXIS: 축 이름	마스터/슬레이브 커플링을 분리하고 자동으로 슬레이브 스핀들 정지
MASLDEF	AXIS: 축 이름	마스터/슬레이브 커플링 정의 마지막 축이 마스터 축이 됩니다.
MASLDEL	AXIS: 축 이름	마스터/슬레이브 커플링 분리 및 그룹화 정의 삭제

온라인 공구 길이 보정			
식별자	**파라미터**		**설명**
	1번째	**2번째**	
TOFFON	AXIS: 옵셋 방향	REAL: 옵셋 방향의 옵셋 값	지정된 옵셋 방향에서 온라인 공구 길이 보정 활성화
TOFFOF	AXIS: 옵셋 방향		지정된 옵셋 방향에서 온라인 공구 길이 보정 리셋

SERUPRO		
식별자	**파라미터**	**설명**
IPTRLOCK		추적 불가능한 프로그램 섹션 시작
IPTRUNLOCK		검색이 억제되는 프로그램 섹션의 끝

후퇴				
식별자	**파라미터**			**설명**
	1번째 - n번째			
POLFMASK	AXIS: 기하 축/기계 축 이름			축에 고속 후퇴 활성화 (축이 서로 연결되지 않은 경우)
POLFMLIN	AXIS: 기하 축/기계 축 이름			축에 직선 고속 후퇴 활성화
POLFA	**1번째**	**2번째**	**3번째**	단일 축의 후퇴 위치
	AXIS: 채널 축 이름	INT: 유형	REAL: 값	

충돌 방지			
식별자	**파라미터**		**설명**
	1번째		
PROTA	STRING: "R"		충돌 모델 재계산 요청
PROTS	**1번째**	**2번째 - n번째**	보호 영역 상태 설정
	CHAR: 상태	STRING: 보호 영역 이름	

지능형 부하 조정					
식별자	**파라미터**				**설명**
CADAPTON	**1번째**	**2번째**	**3번째**	**4번째**	부하 조정 활성화
	INT: 상태	AXIS: 기계 축 이름	INT: 입력 변수	REAL: 입력 값(선택 사항)	
CADAPTOF	**1번째**	**2번째**	**3번째**		부하 조정 비활성화
	INT: 상태	AXIS: 기계 축 이름	INT: 입력 변수		

저크 조정		
식별자	**파라미터**	**설명**
CALCFIR		FIR 저크 필터를 동적 응답 모드로 조정 CALCFIR은 CUST_832 제조업체 싸이클에서 사용됩니다.

저크 조정		
식별자	**파라미터**	**설명**
AFISON		자동 필터 체인 전환 활성화
AFISOF		자동 필터 체인 전환 비활성화

5.5 동기 동작을 위해 사전 정의된 절차

다음과 같은 사전 정의 절차는 동기 동작에만 사용할 수 있습니다.

동기 절차		
식별자	**파라미터**	**설명**
STOPREOF		전처리 정지 취소 STOPREOF 명령으로 동작을 동기화하면 그 다음 출력 블록 (메인 실행을 위한 블록) 후에 전처리 정지가 실행됩니다. 전처리 정지는 출력 블록 종료 또는 STOPREOF 조건이 충족될 때 취소됩니다. 따라서 STOPREOF 명령으로 동기화된 모든 동작은 실행된 것으로 해석됩니다.
RDISABLE		리드인 디스에이블
DELDTG	**1.** AXIS: 이동할 거리를 삭제할 축 (옵션). 이 축을 지정하지 않으면 경로 거리를 대상으로 이동할 거리 삭제가 실행됩니다.	이동할 거리 삭제 DELDTG 명령으로 동작을 동기화하면 그 다음 출력 블록 (메인 실행을 위한 블록) 후에 전처리 정지가 실행됩니다. 전처리 정지는 출력 블록 종료 또는 DELDTG 조건이 충족될 때 취소됩니다. 이동할 거리의 축 삭제의 대상 지점까지의 축 거리는 $AA_DELT[축]에 보관되고 이동할 거리는 $AC_DELT에 보관됩니다.

공정 싸이클의 프로그램 코디네이션		
식별자	**파라미터**	**설명**
	1.	

공정 싸이클의 프로그램 코디네이션		
식별자	**파라미터**	**설명**
LOCK	INT: 디스에이블할 동기 동작의 ID	이 ID를 가진 동기 동작을 잠그거나 공정 싸이클 정지 하나 이상의 ID를 프로그래밍할 수 있습니다.
UNLOCK	INT: 잠금 해제할 동기 동작의 ID	이 ID를 가진 동기 동작의 잠금을 해제하거나 공정 싸이클 계속 하나 이상의 ID를 프로그래밍할 수 있습니다.
ICYCON		ICYCON 이후 공정 싸이클의 개별 블록은 별도의 보간 싸이클로 처리됩니다.
ICYCOF		ICYCOF 이후 공정 싸이클의 전체 블록은 하나의 보간 싸이클로 처리됩니다.

폴리노미얼 기능						
식별자	**파라미터**					**설명**
SYNFCT	**1.**	**2.**	**3.**			모션 동기 동작의 조건이 충족되면 첫 번째 표현식에서 지정한 폴리노미얼에 입력 변수를 적용해 평가합니다. 값의 상한값과 하한값이 제한되고 결과 변수가 지정됩니다.
	INT: FCTDEF로 정의된 폴리노미얼 기능의 번호	VAR REAL: 결과 변수 *)	VAR REAL: 입력 변수 **)			
FTOC	**1.**	**2.**	**3.**	**4.**	**5.**	FCTDEF로 정의한 기능 (최대 3차 폴리노미얼) 에 따라 미세 공구 보정 변경. 여기에 사용되는 번호를 FCTDEF에 지정해야 합니다.
	INT: FCTDEF로 정의된 폴리노미얼 기능의 번호	VAR REAL: 입력 변수 **)	INT: 길이 1, 2, 3	INT: 채널 번호	INT: 스핀들 번호	

*) 특수 시스템 변수만이 결과 변수로 허용됩니다(기능 매뉴얼의 동기 동작 참조).

**) 특수 시스템 변수만이 입력 변수로 허용됩니다(기능 매뉴얼의 동기 동작 참조).

5.6 사전 정의된 기능

사전 정의된 기능을 호출하면 사전 정의된 NC 기능이 실행됩니다. 사전 정의된 절차와 달리 사전 정의된 NC 기능은 반환값을 제공합니다. 사전 정의된 기능 호출은 표현식에 피연산자로 포함시킬 수 있습니다.

좌표계						
식별자	**반환값**	**파라미터**				**설명**
		1.	**2.**	**3. - 15.**	**4. - 16.**	
CTRANS	FRAME	AXIS: 축 이름	REAL: 옵셋	AXIS: 축 이름	REAL: 옵셋	변환: 여러 축에 대한 COARSE 워크 옵셋
CFINE	FRAME	AXIS: 축 이름	REAL: 옵셋	AXIS: 축 이름	REAL: 옵셋	변환: 여러 축에 대한 FINE 워크 옵셋
CSCALE	FRAME	AXIS: 축 이름	REAL: 배율 계수	AXIS: 축 이름	REAL: 배율 계수	배율: 여러 축의 배율 계수
		1.	**2.**	**3. 및 5.**	**4. 및 6.**	
CROT	FRAME	AXIS: 축 이름	REAL: 회전	AXIS: 축 이름	REAL: 회전	회전: 현재 좌표계 회전 최대 파라미터 개수: 6 (기하 축 당 하나의 축 식별자와 하나의 값)
CROTS	FRAME	AXIS: 축 이름	REAL: 입체각을 이용한 회전	AXIS: 축 이름	REAL: 입체각을 이용한 회전	회전: 입체각을 이용해 현재 좌표계 회전 최대 파라미터 개수: 6 (기하 축 당 하나의 축 식별자와 하나의 값)
CMIRROR		**1.**	**2. – 8.**			미러링: 좌표 축의 미러링
	FRAME	AXIS	AXIS			

좌표계						
식별자	**반환값**	**파라미터**				**설명**
		1.	**2.**			
CRPL	FRAME	INT: 로터리 축	REAL: 회전 각도			임의의 평면에서 프레임 회전
ADDFRAME	INT: 0: OK 1: 대상(문자열)이 잘못 명시되었습니다. 2: 대상 프레임이 설정되어 있지 않습니다. 3: 프레임에서 회전이 허용되지 않습니다.	FRAME: 추가 측정 또는 계산된 프레임	STRING: 지정된 대상 프레임			문자열로 지정한 대상 프레임 계산 기존의 전체 프레임과 전송된 프레임을 연결한 새로운 전체 프레임이 생성되도록 대상 프레임을 계산합니다.
INVFRAME	FRAME	**1.** FRAME				프레임에서 반대 프레임 계산 한 프레임과 그 반대 프레임을 연결한 프레임은 항상 제로 프레임이 됩니다.
MEAFRAME	FRAME	**1.** REAL[3,3]: 측정된 공간 지점의 좌표	**2.** REAL[3,3]: 지정된 지점의 좌표	**3.** VAR REAL: FRAME 계산 품질에 대한 정보를 반환하는 변수		공간의 세 개 측정 지점에서 프레임 계산

기하 기능					
식별자	**반환값**	**파라미터**			**설명**
		1.	**2.**	**3.**	
CALCDAT	BOOL: 에러 상태	VAR REAL [n, 2]: 점 1에서 n까지의 표 (가로축, 세로축)	INT: 점 개수	VAR REAL [3]: 결과: 가로축, 세로축 및 계산된 원 중심점의 반경	3지점 또는 4지점을 사용하여 원호의 반경 및 중심점을 계산합니다. 지점들은 반드시 서로 달라야 합니다.
INTERSEC	BOOL: 에러 상태	VAR REAL [11]: 첫 번째 형상 요소	VAR REAL [11]: 두 번째 형상 요소	VAR REAL [2]: 교차 좌표에 대한 결과 변수 가로축 및 세로축	두 형상 요소의 교차점을 계산합니다. 에러 상태가 교차 발견 여부를 표시합니다.

커브 테이블 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTAB	REAL: 추종 축 위치	REAL: 리딩 축 위치	INT: 테이블 번호	VAR REAL[ ]: 피치 결과	AXIS: 스케일링 대상 추종 축	AXIS: 스케일링 대상 리딩 축		커브 테이블에서 지정된 리딩 축 위치에 대한 추종 축 위치를 결정합니다. 파라미터 4/5가 프로그램되지 않은 경우 표준 배율을 적용해 계산합니다.
CTABINV	REAL: 리딩 축 위치	REAL: 추종 축 위치	REAL: 리딩 위치	INT: 테이블 번호	VAR REAL[ ]: 피치 결과	AXIS: 스케일링 대상 추종 축	AXIS: 스케일링 대상 리딩 축	커브 테이블에서 지정된 추종 축 위치에 대한 리딩 축 위치를 결정합니다. 파라미터 5/6가 프로그램되지 않은 경우 표준 배율을 적용해 계산합니다.

커브 테이블 기능								
식별자	반환값	파라미터						설명
		1.	2.	3.	4.	5.	6.	
CTABID	INT: 커브 테이블 번호	INT: 메모리 입력 번호	STRING: 저장 위치: "SRAM", "DRAM"					메모리에 지정된 번호 하에 입력된 커브 테이블 번호를 결정합니다.
CTABISLOCK	INT: 잠금 상태	INT: 테이블 번호						커브 테이블의 잠금 상태를 결정합니다. >0: 테이블이 잠깁니다. 1: CTABLOCK 2: 활성 커플링 3: CTABLOCK 및 활성 커플링 0: 테이블이 잠기지 않음 -1: 테이블이 없음
CTABEXISTS	INT: 존재 유무	INT: 테이블 번호						커브 테이블이 스태틱 NC 메모리와 다이나믹 NC 메모리 중 어디에 있는지를 결정합니다. 0: FALSE 1: TRUE
CTABMEMTYP	INT: 저장 위치	INT: 테이블 번호						커브 테이블의 저장 위치를 결정합니다. 1: DRAM 0: SRAM -1: 테이블이 없음

커브 테이블 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABPERIOD	INT: 주기성	INT: 테이블 번호						커브 테이블의 주기성을 결정합니다. 0: 비주기적 1: 리딩 축에서 주기적 2: 리딩 축 및 추종 축에서 주기적 -1: 테이블이 없음
CTABNO	INT: 커브 테이블 개수							(스태틱 및 다이나믹 NC 메모리에 있는) 정의된 커브 테이블의 개수를 결정합니다.
CTABNOMEM	INT: 커브 테이블 개수	STRING: 저장 위치: "SRAM", "DRAM"						지정된 메모리에서 정의된 커브 테이블의 개수 확인
CTABFNO	INT: 테이블 개수	STRING: 저장 위치: "SRAM", "DRAM"						지정된 메모리에서 아직 사용 가능한 커브 테이블의 개수 확인

커브 테이블 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABSEG	INT: 커브 세그먼트 개수	STRING: 저장 위치: "SRAM", "DRAM"	STRING: 세그먼트 유형: "L": 직선 "P": 폴리노미얼					지정된 메모리에서 지정된 세그먼트 유형을 사용한 커브 세그먼트 개수를 결정합니다. >=0: 번호 -1: 유효하지 않은 메모리 유형 파라미터 2를 프로그램하지 않은 경우 직선 및 폴리노미얼 세그먼트의 합계가 출력됩니다.
CTABFSEG	INT: 커브 세그먼트 개수	STRING: 저장 위치: "SRAM", "DRAM"	STRING: 세그먼트 유형: "L": 직선 "P": 폴리노미얼					지정된 메모리에서 지정된 세그먼트 유형의 아직 사용 가능한 커브 세그먼트 개수 확인. >=0: 번호 -1: 유효하지 않은 메모리 유형
CTABSEGID	INT: 커브 세그먼트 개수	INT: 테이블 번호	STRING: 세그먼트 유형: "L": 직선 "P": 폴리노미얼					커브 테이블이 사용하는 지정된 세그먼트 유형의 커브 세그먼트 개수를 결정합니다. >=0: 번호 -1: 테이블이 없음

커브 테이블 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABMSEG	INT: 커브 세그먼트 개수	STRING: 저장 위치: "SRAM", "DRAM"	STRING: 세그먼트 유형: "L": 직선 "P": 폴리노미얼					지정된 메모리에서 지정된 세그먼트 유형의 커브 세그먼트 최대 개수 확인. >=0: 번호 -1: 테이블이 없음
CTABPOL	INT: 커브 폴리노미얼 개수	STRING: 저장 위치: "SRAM", "DRAM"						지정된 메모리에서 사용된 커브 폴리노미얼의 개수 확인 >=0: 번호 -1: 테이블이 없음
CTABPOLID	INT: 커브 폴리노미얼 개수	INT: 테이블 번호						커브 테이블에서 사용한 커브 폴리노미얼의 개수 확인 >=0: 번호 -1: 테이블이 없음
CTABFPOL	INT: 커브 폴리노미얼 개수	STRING: 저장 위치: "SRAM", "DRAM"						지정된 메모리에 저장 가능한 커브 폴리노미얼 최대 개수를 결정합니다. >=0: 번호 -1: 테이블이 없음
CTABMPOL	INT: 커브 폴리노미얼 개수	STRING: 저장 위치: "SRAM", "DRAM"						지정된 메모리에 저장 가능한 커브 폴리노미얼 최대 개수를 결정합니다. >=0: 번호 -1: 테이블이 없음
CTABSSV	REAL: 추종 축 위치	REAL: 리딩 축 위치	INT: 테이블 번호	VAR REAL[ ]: 피치 결과	AXIS: 스케일링 대상 추종 축	AXIS: 스케일링 대상 리딩 축		지정된 리딩 축 값에 해당하는 커브 세그먼트의 시작에서의 추종 축 위치를 결정합니다.

커브 테이블 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABSEV	REAL: 추종 축 위치	REAL: 리딩 축 위치	INT: 테이블 번호	VAR REAL[ ]: 피치 결과	AXIS: 스케일링 대상 추종 축	AXIS: 스케일링 대상 리딩 축		지정된 리딩 축 값에 해당하는 커브 세그먼트의 끝에서의 추종 축 위치를 결정합니다.
CTABTSV	REAL: 추종 축 위치	INT: 테이블 번호	VAR REAL[ ]: 테이블 시작에서 피치 결과	AXIS: 추종 축				커브 테이블 시작 지점에서의 추종 축 위치를 결정합니다.
CTABTEV	REAL: 추종 축 위치	INT: 테이블 번호	VAR REAL[ ]: 테이블 끝에서 피치 결과	AXIS: 추종 축				커브 테이블 끝 지점에서의 추종 축 위치를 결정합니다.
CTABTSP	REAL: 리딩 축 위치	INT: 테이블 번호	VAR REAL[ ]: 테이블 시작에서 피치 결과	AXIS: 리딩 축				커브 테이블 시작 지점에서의 리딩 축 위치를 결정합니다.
CTABTEP	REAL: 리딩 축 위치	INT: 테이블 번호	VAR REAL[ ]: 테이블 끝에서 피치 결과	AXIS: 리딩 축				커브 테이블 끝 지점에서의 리딩 축 위치를 결정합니다.
CTABTMIN	REAL: 최저값	INT: 테이블 번호	REAL: 리딩 값 간격 하한값	REAL: 리딩 값 간격 상한값	AXIS: 추종 축	AXIS: 리딩 축		정의된 커브 테이블 범위 전체 또는 지정된 간격에서 추종 축의 최소값 확인

커브 테이블 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABTMAX	REAL: 최대값	INT: 테이블 번호	REAL: 리딩 값 간격 하한값	REAL: 리딩 값 간격 상한값	AXIS: 추종 축	AXIS: 리딩 축		정의된 커브 테이블 범위 전체 또는 지정된 간격에서 추종 축의 최대값 확인
참고: 커브 테이블 기능은 동기 동작에도 프로그래밍할 수 있습니다.								

축 기능						
식별자	**반환값**	**파라미터**				
		1.	**2.**	**3.**	**4.**	**설명**
AXNAME	AXIS: 축 이름	STRING [ ]: 입력 문자열				입력 문자열을 축 이름으로 변환
AXSTRING	STRING[ ]: 축 이름	AXIS: 축 이름				축 이름을 문자열로 변환
ISAXIS	BOOL: 축 있음 (TRUE) 또는 없음 (FALSE)	INT: 기하 축의 개수 (1부터 3까지)				머신 데이터 MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB에 따라 파라미터로 지정된 기하 축 1~3 이 존재하는지 확인
SPI	AXIS: 축 이름	INT: 스핀들 번호				스핀들 번호를 축 이름으로 변환
AXTOSPI	INT: 스핀들 번호	AXIS: 축 이름				축 이름을 스핀들 번호로 변환

축 기능						
식별자	**반환값**	**파라미터**				
		1.	**2.**	**3.**	**4.**	**설명**
MODAXVAL	REAL: 모듈로 값	AXIS: 축 이름	REAL: 축 위치			입력된 축 위치에서 모듈로 레스트를 계산합니다. 지정된 축이 모듈로 축이 아닌 경우 축 위치를 변경하지 않고 반환합니다.
POSRANGE	BOOL: 위치 창 내에 위치 지령치 있음(TRUE) 또는 없음(FALSE)	AXIS: 축 이름	REAL: 좌표계의 기준 위치	REAL: 위치 창 너비	INT: 좌표계	축의 위치 지령치가 사전 지정된 기준 위치의 창에 있는지 확인

공구 관리					
식별자	**반환값**	**파라미터**			**설명**
		1.	**2.**	**3.**	
CHKDM	INT: 상태: 확인 결과:	INT: 매거진 번호	INT: D 번호		D 번호가 매거진 내에서 고유한지 검사
CHKDNO	INT: 상태: 확인 결과:	INT: 첫 번째 공구의 T 번호	INT: 두 번째 공구의 T 번호	INT: D 번호	D 번호가 고유한지 검사
GETACTT	INT: 상태	INT: T 번호	STRING [32]: 공구 이름		공구 그룹에서 이름이 동일한 활성 공구 확인
GETACTTD	INT: 상태: 확인 결과:	VAR INT: 발견된 T 번호(반환값)	INT: D 번호		절대 D 번호와 연결된 T 번호를 결정합니다.
GETDNO	INT: D 번호	INT: T 번호	INT: 절삭날 번호		공구(T) 의 절삭날 D 번호를 결정합니다.
GETT	INT: T 번호	STRING [32]: 공구 이름	INT: 듀플로 번호		공구 이름의 T 번호를 결정합니다.

공구 관리					
식별자	**반환값**	**파라미터**			**설명**
		1.	**2.**	**3.**	
NEWT	INT: T 번호	STRING [32]: 공구 이름	INT: 듀플로 번호		새로운 공구를 설정합니다 (공구 데이터를 제공합니다.). 듀플로 번호는 생략될 수 있습니다
TOOLENV	INT: 상태	STRING: 이름			지정된 이름을 가진 공구 환경을 스태틱 NC 메모리에 저장합니다.
DELTOOLENV	INT: 상태	STRING: 이름			지정된 이름을 가진 공구 환경을 스태틱 NC 메모리에서 삭제합니다. 이름을 지정하지 않으면 모든 공구 환경 삭제
GETTENV	INT: 상태	STRING: 이름	VAR INT: T 번호 [0] D 번호 [1] DL 번호 [2]		지정된 이름을 가진 공구 환경으로부터 T 번호, D 번호 및 DL 번호를 결정합니다.

산술					
식별자	**반환값**	**파라미터**			**설명**
		1.	**2.**	**3.**	
SIN	REAL	REAL			사인
ASIN	REAL	REAL			아크 사인
COS	REAL	REAL			코사인
ACOS	REAL	REAL			아크 코사인
TAN	REAL	REAL			탄젠트
ATAN2	REAL	REAL	REAL		아크 탄젠트 2
SQRT	REAL	REAL			제곱근
POT	REAL	REAL			제곱
TRUNC	REAL	REAL			정수
ROUND	REAL	REAL			반내림

산술					
식별자	**반환값**	**파라미터**			**설명**
		1.	**2.**	**3.**	
ROUNDUP	REAL	REAL			반올림
ABS	REAL	REAL			절대치
LN	REAL	REAL			자연 로그
EXP	REAL	REAL			지수 함수 e^x
MINVAL	REAL	REAL	REAL		두 파라미터 중 작은 값을 결정합니다.
MAXVAL	REAL	REAL	REAL		두 파라미터 중 큰 값을 결정합니다.
BOUND	REAL: 점검 상태	REAL: 하한값	REAL: 상한값	REAL: 참조값	참조값이 한계값 내에 있는지를 결정합니다.
참고: 산술 함수는 동기 동작에도 프로그래밍할 수 있습니다. 이 산술 함수는 메인 실행에서 계산 및 평가됩니다. 동기 동작 파라미터 $AC_PARAM[<n>]은 계산에 사용하거나 버퍼로도 사용할 수 있습니다.					

문자열 기능					
식별자	**반환값**	**파라미터**			**설명**
		1.	**2.**	**3.**	
ISNUMBER	BOOL	STRING: 입력 문자열			입력 문자열을 숫자로 변환할 수 있는지 검사
NUMBER	REAL	STRING: 입력 문자열			입력 문자열을 숫자로 변환
TOUPPER	STRING	STRING: 입력 문자열			입력 문자열을 대문자로 바꿔줍니다.
TOLOWER	STRING	STRING: 입력 문자열			입력 문자열을 소문자로 바꿔줍니다.
STRLEN	INT	STRING: 입력 문자열			입력 문자열 (/0) 끝까지 길이를 결정합니다.

문자열 기능					
식별자	**반환값**	**파라미터**			**설명**
		1.	**2.**	**3.**	
INDEX	INT	STRING: 입력 문자열	CHAR: 문자 검색		입력 문자열의 문자 위치를 왼쪽에서 오른쪽으로 결정합니다. 문자열의 왼쪽 첫 번째 문자는 인덱스 0을 가집니다.
RINDEX	INT	STRING: 입력 문자열	CHAR: 문자 검색		입력 문자열의 문자 위치를 오른쪽에서 왼쪽으로 결정합니다. 문자열의 오른쪽 첫 번째 문자는 인덱스 0을 가집니다.
MINDEX	INT	STRING: 입력 문자열	STRING: 문자 검색		입력 문자열의 두 번째 파라미터에 명시된 문자 위치를 왼쪽에서 오른쪽으로 결정합니다. 입력 문자열의 왼쪽 첫 번째 문자는 인덱스 0을 가집니다.
SUBSTR	STRING	STRING: 입력 문자열	INT	INT	시작 문자 (두 번째 파라미터)와 문자의 개수(세 번째 파라미터)에 의해 정의된 입력 문자열의 하위문자열을 결정합니다.
SPRINT	STRING	STRING: 입력 문자열			포맷된 입력 문자열을 결정합니다.

측정 싸이클을 위한 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CALCPOSI	INT: 상태	REAL[3]: WCS의 시작 위치	REAL[3]: 시작 위치를 기준으로 증분 경로 지정	REAL[5]: 모니터링 한계까지의 최소 거리	REAL[3]: 가능 증분 거리에 대한 복귀 배열	BOOL: 측정 시스템 변환 여부	INT: 한계 모니터링 유형	기하 축이 지정된 시작점에서 출발하여 축 한계를 위반하지 않고 사전 정의된 경로를 따라 이송되는지 검사. 정의된 경로로 이송할 수 없는 경우 최대 허용 가능한 값은 반환됩니다.
GETTCOR	INT: 상태	REAL[11]:	STRING: 공구 길이 콤포넌트: 좌표계	STRING: 공구 환경의 이름	INT: 공구의 내부 T 번호	INT: 공구의 절삭날 번호(D 번호)	INT: 위치 종속 옵셋의 번호(공구의 DL 번호)	공구 환경 또는 현 환경에서 공구 길이와 공구 길이 콤포넌트를 결정합니다.
LENTOAX	INT: 상태	INT[3]: 기하 축의 축 지정	REAL[3]: 공구 길이를 좌표계에 매핑하기 위한 매트릭스	STRING: 지정된 좌표계				활성 공구의 공구 길이 L1, L2, L3이 가로 좌표, 세로 좌표, 직교 좌표에 어떻게 지정되어 있는지에 대한 정보를 결정합니다. 기하 축에의 할당은 프레임과 활성 면의 영향을 받습니다(G17 - 19)

SETTCOR	INT: 상태	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	
		REAL [3]: 공간 옵셋 벡터	STR.: 콤포넌트 식별자	INT: 수정 대상 콤포넌트 0 -11	INT: 쓰기 작업 유형 0~3	INT: 기하 축의 인덱스	STRING: 공구 환경의 이름	INT: 공구의 내부 T 번호	INT: 공구의 D 번호	INT: 공구의 DL 번호	개별 콤포넌트 평가에 포함된 모든 보충 조건을 고려하여 공구 콤포넌트 변경합니다.

기타 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
STRINGIS	INT: 문자열 정보	STRING: 검사할 요소의 이름						지정된 문자열을 현재 언어 범위에 있는 NC 프로그래밍 언어의 요소로 사용할 수 있는지 검사
ISVAR	BOOL: 변수 인식 여부	STRING: 변수 이름						전송 파라미터가 NC(머신 데이터, 설정 데이터, 시스템 변수, GUD와 같은 일반 변수) 에 인식된 변수를 포함하고 있는지 검사
GETVARTYP	INT: 데이터 유형	STRING: 변수 이름						시스템/사용자 변수의 데이터 유형을 결정합니다.
GETVARPHU	INT: 물리 단위의 숫자 값	STRING: 변수 이름						시스템/사용자 변수의 물리 단위를 결정합니다.

기타 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
GETVARAP	INT: 액세스를 위한 사용 권한	STRING: 변수 이름	STRING: 액세스 유형					시스템/사용자 변수에 대한 사용 권한을 결정합니다.
GETVARLIM	INT: 상태	STRING: 변수 이름	CHAR: 리드아웃할 한계값 지정	VAR REAL: 한계값 반환				시스템/사용자 변수의 하한값/상한값을 결정합니다.
GETVARDFT	INT: 상태	STRING: 변수 이름	VAR REAL / STRING/ FRAME: 기본 값 반환	INT: 1차원 인덱스 (옵션)	INT: 2차원 인덱스 (옵션)	INT: 3차원 인덱스 (옵션)		시스템/사용자 변수의 기본 값을 결정합니다.
COLLPAIR	INT: 점검 결과	STRING: 첫 번째 보호 영역의 이름	STRING: 2번째 보호 영역의 이름	BOOL: 알람 억제(옵션)				충돌 쌍의 일부인지 여부를 확인합니다.
PROTD	REAL: 두 보호 영역 간의 간격	STRING: 첫 번째 보호 영역의 이름	STRING: 2번째 보호 영역의 이름	VAR REAL: 반환값: 3차원 간격 벡터	BOOL: 간격 및 간격 벡터 측정 시스템 (옵션)			지정된 두 보호 영역 간의 간격을 결정합니다.

기타 기능								
식별자	**반환값**	**파라미터**						**설명**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
DELOBJ	INT: 에러 번호	STRING: 삭제할 콤포넌트 유형	INT: 삭제할 콤포넌트의 시작 인덱스(옵션)	INT: 삭제할 콤포넌트의 끝 인덱스(옵션)	BOOL: 알람 억제 (옵션)			키네마틱 체인, 보호 영역, 보호 영역 요소, 충돌 페어 및 변환 데이터에서 요소를 삭제합니다.
NAMETOINT	INT: 시스템 변수 인덱스	STRING: 시스템 변수 배열의 이름	STRING: 문자열/이름	BOOL: 알람 억제 (옵션)				문자열을 기초로 관련 시스템 변수 인덱스를 확인합니다.
ORISOLH	INT: 에러 번호	INT: 기능의 동작 제어.	REAL: 첫 번째 각도	REAL: 두 번째 각도				선삭 공구가 공작물에 대해 키네마틱 독립적인 지정 위치에 올 수 있도록 사용자가 기계의 로터리 축 위치를 설정하도록 지원합니다. 요구사항: 키네마틱 체인으로 파라미터화된 6축 변환이 활성화되어 있습니다.

기타 기능								
식별자	반환값	파라미터						설명
		1.	2.	3.	4.	5.	6.	
CORRTRAFO	INT: 에러 번호	REAL: 수정 벡터	INT: 수정할 요소	INT: 수정 모드	BOOL: 알람 억제 (옵션)			기계의 키네마틱 모델에서 오리엔테이션 축의 옵셋 벡터 또는 방향 벡터를 수정합니다.
CORRTC	INT: 에러 번호	REAL: 수정 벡터	INT: 수정할 요소	INT: 수정 모드	BOOL: 알람 억제 (옵션)			기계 측정에 따라 오리엔테이션 전환이 가능한 공구 캐리어의 옵셋 벡터 또는 방향 벡터를 수정하십시오.

5.7 현재 HMI에서 설정된 언어

아래 테이블은 사용자 인터페이스에서 사용 가능한 언어를 모두 열거한 것입니다.

현재 설정된 언어는 다음 시스템 변수를 사용하여 가공 프로그램 및 동기 동작에서 쿼리할 수 있습니다.

$AN_LANGUAGE_ON_HMI = <값>

<값>	언어	언어 코드
1	독일어 (독일)	GER
2	프랑스어	FRA
3	영어 (영국)	ENG
4	스페인어	ESP
6	이탈리아어	ITA
7	네델란드어	NLD
8	중국어 간체	CHS
9	스웨덴어	SVE
18	헝가리어	HUN
19	핀란드어	FIN
28	체코어	CSY

<값>	언어	언어 코드
50	포르투갈어 (브라질)	PTB
53	폴란드어	PLK
55	덴마크어	DAN
57	러시아어	RUS
68	슬로바키아어	SKY
72	루마니아어	ROM
80	중국어 번체	CHT
85	한국어	KOR
87	일본어	JPN
89	터키어	TRK

참고

다음 경우 $AN_LANGUAGE_ON_HMI가 업데이트됩니다.

- 시스템 부팅 후
- NC 및/또는 PLC가 리셋된 후.
- M2N 범위 내의 다른 NC로 전환한 후.
- HMI에서 언어를 변경한 후

부록 A

A.1 약어 목록

A	
A	출력
AFIS	Automatic Filter Switch: 자동 필터 전환
ASCII	American Standard Code for Information Interchange: 정보교환에 사용되는 미국 코드 표준
ASIC	Application Specific Integrated Circuit: 사용자 전환 회로
ASUP	비동기 서브프로그램
AUTO	"자동" 조작 모드
AUXFU	Auxiliary Function: 보조 기능
AWL	명령문 목록

B	
BAG	모드 그룹
BCD	Binary Coded Decimals: 2진 코드로 암호화된 10진수
BICO	Binector Connector
BIN	Binary Files: 이진 파일
BKS	기본 좌표계
BM	작동 알람
BO	Binector Output
BTSS	화면 조작반 인터페이스

C	
CLC	간격 제어
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
COM	Communication

C	
CP	Communication Processor
CPU	Central Processing Unit: 중앙 처리 장치
CST	Configured Stop: 구성된 정지

D	
DB	데이터 블록(PLC)
DBB	데이터 블록 바이트(PLC)
DBD	데이터 블록 더블 워드(PLC)
DBW	데이터 블록 워드(PLC)
DBX	데이터 블록 비트(PLC)
DDS	Drive Data Set: 드라이브 데이터 세트
DIR	Directory: 디렉토리
DO	Drive Object
DRF	Differential Resolver Function: 핸드휠 옵셋 기능 (핸드휠)
DRY	Dry Run: 드라이런 이송 속도
DW	데이터 워드
DWORD	더블 워드(현재 32비트)

E	
E	입력
EES	Execution from External Storage
E/A	입력/출력
ESR	확장 정지 및 후퇴
ETC	ETC 키 ">"; 같은 메뉴의 소프트 키 확장

F	
FB	펑션 블록(PLC)
FC	Function Call: 펑션 블록(PLC)
FDD	Feed Disable: 피드 비활성
FdStop	Feed Stop: 이송 속도 정지

F	
FIFO	First In First Out: 주소 지정 없이 작동하고 데이터를 저장한 순서대로 읽는 메모리
FM	오류 메시지
FUP	제어 시스템 흐름도(PLC 프로그래밍 방법)
FW	Firmware

G	
GEO	기하(예: 기하 축)
GP	기본 프로그램(PLC)
GUD	Global User Data: 전역 사용자 데이터

H	
HEX	16진수의 약어
HiFu	보조 기능
HMI	Human Machine Interface: SINUMERIK 사용자 인터페이스
HSA	메인 스핀들 드라이브
HT	Handheld Terminal
HW	하드웨어

I	
IBN	스타트업
INC	Increment: 증분
INI	Initializing Data: 데이터 초기화
IPO	Interpolator

J	
JOG	JOG 모드: 설정 모드

K	
KOP	래더 다이어그램 (PLC 프로그래밍 방법)

L	
LED	Light Emitting Diode: 발광 다이오드
LMS	위치 측정 시스템
LR	위치 컨트롤러

M	
Main	Main program: 메인 프로그램(OB1, PLC)
MCP	Machine Control Panel: 기계 조작반
MD	머신 데이터
MDA	Manual Data Automatic: 수동 입력
MDS	Motor Data Set: 모터 데이터 세트
MELDW	메시지 워드
MKS	기계 좌표계
MM	Motor Module
MPF	Main Program File: 메인 프로그램(NC)
MPI	Multi Point Interface
MSTT	기계 조작반

N	
NC	Numerical Control: 블록 준비, 이송 범위 등을 포함한 수치 제어
NCU	Numerical Control Unit: NC 하드웨어 장치
NCK	Numerical Control Kernel
NCSD	NC Start Disable
NST	인터페이스 신호
NV	워크 옵셋
NX	Numerical Extension: 축 확장 모듈

O	
OB	PLC내 조직 블록
OEM	Original Equipment Manufacturer
OP	Operation Panel: 작동 장비

P	
PCU	PC Unit: PC 박스(컴퓨터 유닛)
PG	프로그래밍 장치
PLC	Programmable Logic Control: 컨트롤러
PN	PROFINET
PO	Power On
POS	위치/포지셔닝
PPO	파라미터 공정 데이터 오브젝트: PROFIBUS DP 전송을 위한 주기적 데이터 텔레그램 및 "가변 속도 드라이브" 특성.
PPU	Panel Processing Unit (SINUMERIK 828D 등 패널 기반 CNC의 핵심 하드웨어)
PROFIBUS	Process Field Bus: 직렬 데이터 버스
PRT	프로그램 테스트
PTP	Point to Point: 포인트-포인트
PZD	공정 데이터: PPO의 공정 데이터 부분

R	
REF	원점 복귀 기능
REPOS	리포지셔닝 기능
RESU	Retrace support
RID	Read In Disable
RP	R 파라미터, 산술 파라미터, 사전 정의 사용자 변수

S	
SA	동기 동작
SBL	Single Block: 싱글 블록

S	
SBT	Safe Brake Test
SCC	Safety Control Channel
SCL	Structured Control Language
SD	Settingdatum 또는 설정 데이터
SDI	Safe Direction
SERUPRO	Search-Run by Program Test: 프로그램 테스트를 통한 블록 탐색
SIC	Safety Info Channel
SKP	Skip: 가공 프로그램 블록을 건너뛰기 위한 기능
SLP	Safe Limited Position
SLS	Safely Limited Speed
SMI	Sensor Module Integrated
SOS	Safe Operating Stop
SPF	Sub Program File: 서브프로그램(NC)
SS1	Safe Stop 1
SS2	Safe Stop 2
STO	Safe Torque Off
STW	제어 워드
SUG	연삭 휠 원주 속도
SW	소프트웨어

T	
TCU	Thin Client Unit
TM	Terminal Module (SINAMICS)
TO	Tool Offset: 공구 옵셋
TOA	Tool Offset Active: 공구 옵셋 식별자(파일 유형)
TOFF	온라인 공구 길이 옵셋
TRANSMIT	Transform Milling Into Turning: 터닝 머신에서 밀링 작업을 위해 좌표 변환

U	
UP	서브프로그램
USB	Universal Serial Bus

V	
VDI	NC와 PLC 간의 내부 통신 인터페이스

W	
WKS	공작물 좌표계
WPD	Work Piece Directory: 공작물 디렉토리
WZ	공구
WZV	공구 관리

Z	
ZSW	상태 워드 (드라이브)

A.1 약어 목록

인덱스

-

- 시험 절삭 추가 종료 - GROUP_ADDEND
  외부 프로그래밍, 1339

"

"Siemens Industry Online Support" 앱, 27

$

$A_PROBE, 691, 699
$A_PROBE_LIMITED, 699
$AA_ACC, 143
$AA_ATOL, 959
$AA_COUP_ACT
  축 마스터 값 커플링, 1042
  커플 모션 중, 1018
$AA_ESR_ENABLE, 1111
$AA_FGREF, 125
$AA_FGROUP, 125
$AA_G0MODE, 202
$AA_LEAD_SP, 1042
$AA_LEAD_SV, 1042
$AA_MM, 692
$AA_MM1...4, 700
$AA_MW, 692
$AA_MW1...4, 700
$AC_ACT_PROG_NET_TIME, 1120
$AC_ACTUAL_PARTS, 1124
$AC_AXCTSWA, 1003
$AC_AXCTSWE, 1003
$AC_CTOL, 959
$AC_CTOL_G0_ABS, 206
$AC_CUT_INV, 907
$AC_CUTMOD, 906
$AC_CUTMOD_ANG, 906
$AC_CUTMODK, 907
$AC_CUTTING_TIME, 1120
$AC_CYCLE_TIME, 1120
$AC_DELAYFST, 970
$AC_ESR_TRIGGER, 1111
$AC_F_TYPE, 159
$AC_FGROUP_MASK, 125
$AC_FZ, 159
$AC_MEA, 692, 699
$AC_OLD_PROG_NET_TIME, 1120
$AC_OLD_PROG_NET_TIME_COUNT, 1120
$AC_OPERATING_TIME, 1120
$AC_OTOL, 959
$AC_OTOL_ G0_ABS, 206
$AC_PROG_NET_TIME_TRIGGER, 1121
$AC_REPOS_PATH_MODE, 979
$AC_REQUIRED_PARTS, 1123
$AC_S_TYPE, 109
$AC_SMAXVELO, 982
$AC_SMAXVELO_INFO, 982
$AC_SPECIAL_PARTS, 1124
$AC_STOLF, 205
$AC_SVC, 109
$AC_TOFF, 97
$AC_TOFFCR, 98
$AC_TOFFL, 97
$AC_TOFFR, 98
$AC_TOTAL_PARTS, 1123
$AC_TRAFO_CORR_ELEM_P, 827
$AC_TRAFO_CORR_ELEM_T, 827
$AC_TRAFO_ORIAX_LOC, 827
$AN_AXCTAS, 1003
$AN_AXCTSWA, 1003
$AN_ESR_TRIGGER, 1111
$AN_LANGUAGE_ON_HMI, 1541
$AN_POWERON_TIME, 1119
$AN_SETUP_TIME, 1119
$NT_CLOSE_CHAIN_T, 827
$NT_CNTRL, 827
$NT_CORR_ELEM_P, 826
$NT_CORR_ELEM_T, 826
$NT_NAME, 819
$NT_ROT_AX_NAME, 901
$NT_TRAFO_INDEX, 819
$P_ACTBFRAME, 732
$P_AD, 906
$P_AEP, 311
$P_APDV, 311
$P_APR, 311
$P_BFRAME, 732
$P_CHBFRAME, 731
$P_CHBFRMASK, 732
$P_CTOL, 959
$P_CTOL_ G0_ABS, 206
$P_CUT_INV, 907
$P_CUTMOD, 906
$P_CUTMOD_ANG, 906
$P_CUTMOD_ERR, 908
$P_CUTMODK, 907

$P_DELAYFST, 970
$P_F_TYPE, 160
$P_FGROUP_MASK, 126
$P_FZ, 160
$P_GWPS, 116
$P_IFRAME, 733
$P_IS_EES_PATH, 631
$P_NCBFRAME, 731
$P_NCBFRMASK, 732
$P_ORI_DIFF, 896
$P_ORI_POS, 895
$P_ORI_SOL, 897
$P_ORI_STAT, 900
$P_OTOL, 959
$P_OTOL_ G0_ABS, 206
$P_PATH, 630
$P_PFRAME, 733
$P_PROG, 630
$P_PROGPATH, 630
$P_S_TYPE, 109
$P_SIM, 703, 1014, 1125
$P_STACK, 630
$P_STOLF, 205
$P_SUBPAR, 556
$P_SVC, 109
$P_TOFF, 97
$P_TOFFCR, 97
$P_TOFFL, 97
$P_TOFFR, 97
$P_TOOLENVN, 916
$P_TOOLENV은, 917
$P_WORKAREA_CS_COORD_SYSTEM, 392
$P_WORKAREA_CS_LIMIT_MINUS, 393
$P_WORKAREA_CS_LIMIT_PLUS, 393
$P_WORKAREA_CS_MINUS_ENABLE, 393
$P_WORKAREA_CS_PLUS_ENABLE, 392
$PA_ATOL, 959
$PA_FGREF, 125
$PA_FGROUP, 126
$SA_LEAD_TYPE, 1042
$SC_PA_ACTIV_IMMED, 653
$SN_PA_ACTIV_IMMED, 653
$TC_CARR_CORR_ELEM, 886
$TC_CARR1...14, 871
$TC_CARR13, 879
$TC_CARR14, 879
$TC_CARR18...23, 871
$TC_CARR18[m], 876
$TC_DP1 ... 25, 828
$TC_TP_MAX_VELO, 105

*****

* (산술 함수), 490

/

/ (산술 함수), 490

+

+ (산술 함수), 490

<

< (비교 연산자), 492
<< (연결 연산자), 500
<= (관계 연산자), 492
<> (비교 연산자), 492

=

== (비교 연산자), 492

>

> (비교 연산자), 492
>= (관계 연산자), 492

0

0 문자, 498

1

10진수 상수, 423
16진법 상수, 424

A

A 스플라인, 675
ABS, 490
AC, 167
ACC, 142
ACCLIMA, 951
ACN, 174
ACOS, 490
ACP, 174

ACTBLOCNO, 568
ACTFRAME, 709
ADIS, 327
ADISPOS, 327
ADISPOSA, 704
AFISOF, 962
AFISON, 962
ALF
    나사 절삭 중 고속 후퇴, 256
    형상에서 고속 후퇴, 615
AMIRROR, 363
AND, 492
ANG, 234
ANG1, 234
ANG2, 234
AP, 194
APR, 458
APRB, 458
APRP, 458
APW, 458
APWB, 458
APWP, 458
AR
    원호 경로 프로그래밍, 214
AROT, 348
AROTS, 355
AS, 546
ASCALE, 360
ASIN, 490
ASPLINE, 667
ATAN2, 490
ATOL, 955
ATRANS, 342
AV, 1051
AX, 995
AXCTSWE, 1002
AXCTSWEC, 1002
AXCTSWED, 1002
AXIS, 439
AXNAME, 499
AXSTRING, 995
AXTOCHAN, 993
AXTOSPI, 995

B

B 스플라인, 676
B_AND, 492
B_NOT, 492
B_OR, 492
B_XOR, 492
BAUTO, 667
BCS, 43
BFRAME, 709
BLOCK, 597
BLSYNC, 608
BNAT, 667
BOOL, 439
BOUND, 471
BRISK, 948
BRISKA, 948
BSPLINE, 667
BTAN, 667
BZS, 45

C

C 스플라인, 677
CAC, 666
CACN, 666
CACP, 666
CADAPTOF, 1006
CADAPTON, 1006
CALL, 596
CALLPATH, 600
CASE, 522
CDC, 666
CDOF, 315
CDOF2, 315
CDON, 315
CFC, 148
CFIN, 148
CFINE, 719
CFTCP, 148
CHAN, 439
CHANDATA, 632
CHAR, 439
CHF, 271
CHKDNO, 867
CHR, 271
CIC, 666
CIP, 219
CLEARM, 540
CLRINT, 611
COARSE, 1051
COARSEA, 704
COMPCAD, 680
COMPCURV, 680
COMPLETE, 632
COMPOF, 680
COMPON, 680
COMPSURF, 680
CONTDCON, 1157
CONTPRON, 1150

CORROF, 372
CORRTC, 882
CORRTRAFO, 820
COS, 490
COUPDEF, 1051
COUPDEL, 1051
COUPOF, 1051
COUPOFS, 1051
COUPON, 1051
COUPONC, 1051
COUPRES, 1051
CP, 794
CPBC, 1064
CPDEF, 1063
CPDEL, 1063
CPFMOF, 1067
CPFMON, 1066
CPFMSON, 1065
CPFPOS + CPOF, 1067
CPFPOS + CPON, 1064
CPFRS, 1063
CPLA, 1063
CPLCTID, 1063
CPLDEF, 1063
CPLDEL, 1063
CPLDEN, 1063
CPLINSC, 1070
CPLINTR, 1070
CPLNUM, 1063
CPLOF, 1063
CPLON, 1063
CPLOUTSC, 1071
CPLOUTTR, 1070
CPLPOS, 1064
CPLSETVAL, 1063
CPMALARM, 1071
CPMBRAKE, 1071
CPMPRT, 1070
CPMRESET, 1068
CPMSTART, 1070
CPMVDI, 1071
CPOF, 1063
CPON, 1063
CPRECOF, 960
CPRECON, 960
CPROT, 650
CPROTDEF, 646
CPSETTYPE, 1071
CPSYNCOP, 1071
CPSYNCOP2, 1071
CPSYNCOV, 1071
CPSYNFIP, 1071
CPSYNFIP2, 1071
CPSYNFIV, 1071
CR, 212
CROTS, 355
CSPLINE, 667
CT, 222
CTAB, 1030
CTABDEF, 1019
CTABDEL, 1026
CTABEND, 1019
CTABEXISTS, 1025
CTABFNO, 1035
CTABFPOL, 1035
CTABFSEG, 1035
CTABID, 1029
CTABINV, 1030
CTABISLOCK, 1029
CTABLOCK, 1027
CTABMEMTYP, 1029
CTABMPOL, 1035
CTABMSEG, 1035
CTABNO, 1035
CTABNOMEM, 1035
CTABPERIOD, 1029
CTABPOL, 1035
CTABPOLID, 1035
CTABSEG, 1035
CTABSEGID, 1035
CTABSEV, 1030
CTABSSV, 1030
CTABTEP, 1030
CTABTEV, 1030
CTABTMAX, 1030
CTABTMIN, 1030
CTABTSP, 1030
CTABTSV, 1030
CTABUNLOCK, 1027
CTOL, 955
CTOLG0, 202
CTRANS, 719
CUT2D, 317
CUT2DD, 317
CUT2DF, 317
CUT2DFD, 317
CUT3DC, 845
CUT3DCC, 856
CUT3DCCD, 856
CUT3DCD, 845
CUT3DF, 850
CUT3DFD, 850
CUT3DFF, 850
CUT3DFS, 850

CUTCONOF, 320
CUTCONON, 320
CUTMOD, 902
CUTMODK, 902
CYCLE4071
    외부 프로그래밍, 1314
CYCLE4072
    외부 프로그래밍, 1316
CYCLE4073
    외부 프로그래밍, 1320
CYCLE4074
    외부 프로그래밍, 1322
CYCLE4075
    외부 프로그래밍, 1326
CYCLE4077
    외부 프로그래밍, 1329
CYCLE4078
    외부 프로그래밍, 1333
CYCLE4079
    외부 프로그래밍, 1335
CYCLE435 - 드레서 좌표계 설정
    외부 프로그래밍, 1258
CYCLE495 - 형상 트루잉
    외부 프로그래밍, 1259
CYCLE60 - 조각
    외부 프로그래밍, 1192
CYCLE61 - 페이스 밀링
    외부 프로그래밍, 1196
CYCLE62 - 형상 호출
    외부 프로그래밍, 1198
CYCLE63 – 형상 포켓 밀링 / 형상 포켓 잔삭 가공 / 형상 스피곳 밀링 / 형상 스피곳 잔삭 가공
    외부 프로그래밍, 1199
CYCLE64 - 형상 포켓 예비 드릴링
    외부 프로그래밍, 1203
CYCLE70 - 나사 밀링
    외부 프로그래밍, 1206
CYCLE72 - 경로 밀링
    외부 프로그래밍, 1208
CYCLE76 - 사각 스피곳
    외부 프로그래밍, 1212
CYCLE77 - 원주
    외부 프로그래밍, 1216
CYCLE78 - 드릴 나사 밀링
    외부 프로그래밍, 1218
CYCLE782 - 부하에 맞게 조정
    외부 프로그래밍, 1262
CYCLE79 - 다각형
    외부 프로그래밍, 1222
CYCLE800 – 스위블 평면 / 스위블 공구 / 정렬 공구
    외부 프로그래밍, 1265
CYCLE801 – 격자 또는 프레임 위치 패턴
    외부 프로그래밍, 1270
CYCLE802 - 임의 위치
    외부 프로그래밍, 1272
CYCLE806 - 보간 선삭
    외부 프로그래밍, 1276
CYCLE81 - 센터링
    외부 프로그래밍, 1225
CYCLE82 - 드릴링
    외부 프로그래밍, 1227
CYCLE83 - 심공 드릴링 1
    외부 프로그래밍, 1230
CYCLE830 - 심공 드릴링 2
    외부 프로그래밍, 1277
CYCLE832 - 고속 가공 설정
    외부 프로그래밍, 1285
CYCLE84 - 동기 태핑
    외부 프로그래밍, 1234
CYCLE840 - 태핑 척을 사용한 태핑
    외부 프로그래밍, 1289
CYCLE85 - 리밍
    외부 프로그래밍, 1238
CYCLE86 - 보링
    외부 프로그래밍, 1240
CYCLE899 - 개방형 슬롯
    외부 프로그래밍, 1293
CYCLE92 - 절단
    외부 프로그래밍, 1242
CYCLE930 - 홈
    외부 프로그래밍, 1296
CYCLE940 – 언더컷 형상 E 및 F / 언더컷 나사
    외부 프로그래밍, 1300
CYCLE95 - 형상 절삭
    외부 프로그래밍, 1244
CYCLE951 - 선삭
    외부 프로그래밍, 1303
CYCLE952 – 칩 제거 / 잔여 칩 제거 / 플런지 절삭 / 잔여 플런지 절삭 / 플런지 선삭 / 잔여 플런지 선삭
    외부 프로그래밍, 1306
CYCLE98 - 나사 체인
    외부 프로그래밍, 1246
CYCLE99 - 나사 선삭
    외부 프로그래밍, 1253

D

D 번호
    Check, 867
    이름 변경, 868
    자유롭게 할당, 867
D..., 88
D0, 88

DAC, 183
DC, 174
DEF, 439
DEFAULT, 522
DEFINE ... AS, 546
DELAYFSTOF, 968
DELAYFSTON, 968
DELDL, 832
DELETE, 639
DELOBJ, 808
DELTOOLENV, 914
DIACYCOFA, 183
DIAM90, 180
DIAM90A, 183
DIAMCHAN, 183
DIAMCHANA, 183
DIAMCYCOF, 180
DIAMOF, 180
DIAMOFA, 183
DIAMON, 180
DIAMONA, 183
DIC, 183
DILF, 256
DIN 서브프로그램 이름, 628
DIN 66217, 42
DISABLE, 611
DISC, 295
DISCL, 298
DISPLOF, 568
DISPLON, 568
DISPR, 972
DISR, 298
DISRP, 298
DITE, 253
DITS, 253
DIV, 490
DL, 831
DO, 1083
DRFOF, 375
DRIVE, 948
DRIVEA, 948
DRVPRD, 1011
DRVPWR, 1011
DV, 1051
DYNFINISH, 952
DYNNORM, 952
DYNPOS, 952
DYNPREC, 952
DYNROUGH, 952
DYNSEMIFIN, 952

E

Easy XML, 1131
EAUTO, 667
EES, 622
EES 표기법, 624
EG
  전자 기어, 1043
EGDEF, 1044
EGDEL, 1049
EGOFC, 1048
EGOFS, 1048
EGON, 1045
EGONSYN, 1045
EGONSYNE, 1045
ELSE, 534, 1083
ENABLE, 611
ENAT, 667
ENDFOR, 536
ENDIF, 534
ENDLABEL, 525
ENDLOOP, 535
ENDWHILE, 538
ESR, 1110
ESRR, 1117
ESRS, 1116
ETAN, 667
Euler 각도, 755
EVERY, 1083
EXECSTRING, 488
EXECTAB, 1162
EXECUTE, 1165
EXP, 490
EXTCALL
  SINUMERIK 840D sl 관련, 602
EXTCLOSE, 1136
EXTERN, 590
EXTOPEN, 1136

F

F...
  나사 절삭 G34 G35, 252
  이송 속도, 117
  직선 보간, 206
FA, 137
FAD, 298
FALSE, 439
FB, 154
FCTDEF, 841

FCUB, 943
FD, 144
FDA, 144
FENDNORM, 703
FFWOF, 954
FFWON, 954
FGREF, 117
FGROUP, 117
FIFOCTRL, 965
FILEDATE, 644
FILEINFO, 644
FILESIZE, 644
FILESTAT, 644
FILETIME, 644
FINE, 1051
FINEA, 704
FL, 117
FLIM, 984
FLIN, 943
FMA, 150
FNORM, 943
FOR, 536
FP, 396
FPO, 943
FPR, 137
FPRAOF, 137
FPRAON, 137
FRAME, 439
FRC, 271
FRCM, 271
FROM, 1083
FTOCOF, 844
FTOCON, 844

G

G 그룹
    기술, 952
G 명령, 1466
G 코드
    간접 프로그래밍, 484
G0 공차, 202
G1, 206
G110, 192
G111, 192
G112, 192
G140, 298
G141, 298
G142, 298
G143, 298
G147, 298
G148, 298
G153
    워크 옵셋, 160
    프레임 선택 해제, 371
G17, 164
G18, 164
G19, 164
G2, 209
G247, 298
G248, 298
G25
    스핀들 속도 제한, 116
    작업 영역 제한, 388
G26
    스핀들 속도 제한, 116
    작업 영역 제한, 388
G290, 1148
G291, 1148
G3, 209
G33, 245
G335, 260
G336, 260
G34, 252
G340, 298
G341, 298
G347, 298
G348, 298
G35, 252
G4, 405
G40, 277
G41, 277
G42, 277
G450, 295
G451, 295
G460, 311
G461, 311
G462, 311
G5, 790
G500
    워크 옵셋, 160
G505 ...G599, 160
G53
    워크 옵셋, 160
    프레임 선택 해제, 371
G54 ...G57, 160
G58, 346
G59, 346
G60, 324
G601, 324
G602, 324
G603, 324
G62, 703
G621, 703

G64, 327
G641, 327
G642, 327
G643, 327
G644, 327
G645, 327
G7, 790
G70, 176
G700, 176
G71, 176
G710, 176
G74, 395
G75, 396
G810 ...G819, 702
G820 ...G829, 702
G9, 324
G90, 167
G91, 169
G93, 117
G94, 117
G95, 117
G96, 110
G961, 110
G962, 110
G97, 110
G971, 110
G972, 110
G973, 110
GEOAX, 997
GET, 988
GETACTTD, 869
GETD, 988
GETDNO, 868
GETTCOR, 917
GETTENV, 915
GETVARAP, 475
GETVARDFT, 478
GETVARDIM, 478
GETVARLIM, 477
GETVARPHU, 474
GETVARTYP, 480
GFRAME0 ... GFRAME100, 377
GOTO, 519
GOTOB, 519
GOTOC, 519
GOTOF, 519
GOTOS, 517
GP, 485
GROUP_ADDEND - 시험 절삭 추가 종료
    외부 프로그래밍, 1339
GROUP_BEGIN - 프로그램 블록 시작
    외부 프로그래밍, 1338
GROUP_END - 프로그램 블록 종료
    외부 프로그래밍, 1339
GUD, 440
GWPS, 115
GWPSOF, 115
GWPSON, 115

H

HOLES1 - 행 위치 패턴
    외부 프로그래밍, 1170
HOLES2 - 원호 또는 피치원 위치 패턴
    외부 프로그래밍, 1171

I

I...
    나사 절삭 G33, 245
    나사 절삭 G34 G35, 252
    원호 보간, 209
IC, 169
ID, 1083
IDS, 1083
IF, 534
IFRAME, 709
INDEX, 503
INICF, 439
INIPO, 439
INIRE, 439
INIT, 540
INITIAL, 632
INITIAL_INI, 632
INT, 439
INTEGER 상수, 424
INTERSEC, 1160
INVCCW, 229
INVCW, 229
IPOBRKA, 704
IPOENDA, 704
IPOSTOP, 1051
IPTRLOCK, 970
IPTRUNLOCK, 970
IR, 260
ISAXIS, 995
ISFILE, 642
ISNUMBER, 499
ISOCALL, 598
ISVAR, 472

J

J...
    나사 절삭 G34 G35, 252
    원호 보간, 209
JERKLIM, 983
JERKLIMA, 951
JR, 260

K

K...
    나사 절삭 G33, 245
    나사 절삭 G34 G35, 252
    원호 보간, 209
KONT, 288
KONTC, 288
KONTT, 288
KR, 260

L

L..., 588
Label, 525
LEAD, 752
LEADOF, 1037
LEADON, 1037
LENTOAX, 940
LF, 52
LFOF, 256
LFON, 256
LFPOS, 256
LFTXT, 256
LFWP, 256
LIFTFAST, 613
LIMS, 110
LINE FEED, 52
LLI, 453
LN, 490
LONGHOLE - 연장된 홀
    외부 프로그래밍, 1189
LOOP, 535
LUD, 440

M

M 코드, 381
M..., 381
M0, 381
M1, 381
M17, 572
M19
    M 코드, 381
    스핀들 포지셔닝, 131
M2, 381
M3, 99
M30, 572
M4, 99
M40, 381
M41, 381
M42, 381
M43, 381
M44, 381
M45, 381
M5, 99
M6, 72
M70, 131
MASLDEF, 1079
MASLDEL, 1079
MASLOF, 1079
MASLOFS, 1079
MASLON, 1079
MATCH, 503
MAXVAL, 471
MCALL, 594
MCS, 41
MD10010, 540
MD10210, 882
MD10240, 178
MD10260, 176
MD10280, 540
MD10651, 261
MD10710, 255
MD15800, 436
MD18104, 913
MD18116, 914
MD18156, 436
MD20360, 923
MD24558, 924
MD24658, 924
MD9440, 90
MEAC, 692
MEAFRAME, 725
MEAS, 689
MEASA, 692
MEAW, 689
MEAWA, 692
MINDEX, 503
MINVAL, 471
MIRROR, 363
MMC, 1131

MOD, 490
MODAXVAL, 995
MPF, 620
MSG, 384
mySupport 문서, 25

N

NAMETOINT, 813
NC, 439
NC 고급 언어, 50
NC 프로그램
    생성, 57
NC 프로그램 작성
    문자 세트, 57
NCK 표기법, 624
NEWCONF, 1126
NOC, 1051
NORM, 288
NOT, 492
NPROT, 650
NPROTDEF, 646
NUMBER, 499
NUT, 764

O

OEM 어드레스, 702
OEM 함수, 702
OEMIPO1/2, 702
OFFN, 277
OMA1 ... OMA5, 702
OpenSSL, 28
ORIAXES, 761
ORIC, 861
ORICONCCW, 764
ORICONCW, 764
ORICONIO, 764
ORICONTO, 764
ORICURVE, 767
ORID, 861
ORIEULER, 761
ORIMKS, 759
ORIPATH, 775
ORIPATHS, 775
ORIPLANE, 764
ORIRESET(A, B, C), 750
ORIROTA, 771
ORIROTC
    공구 오리엔테이션 회전, 771
    공구 회전 보간 중, 777
ORIROTR, 771
ORIROTT, 771
ORIRPY, 761
ORIRPY2, 761
ORIS, 861
ORISOF, 783
ORISOLH, 889
ORISON, 783
ORIVECT, 761
ORIVIRT1, 761
ORIVIRT2, 761
ORIWKS, 759
OS, 1084
OSB, 1084
OSC, 861
OSCILL, 1090
OSCTRL, 1084
OSD, 861
OSE, 1084
OSNSC, 1084
OSOF, 861
OSP1, 1084
OSP2, 1084
OSS, 861
OSSE, 861
OST, 861
OST1, 1084
OST2, 1084
OTOL, 955
OTOLG0, 202
OVR, 141
OVRA, 141
OVRRAP, 141

P

P..., 592
P_ACTFRAME, 733
PACCLIM, 986
PAROT, 368
PAROTOF, 368
PCALL, 599
PDELAYOF, 1098
PDELAYON, 1098
PFRAME, 709
PHI
    오리엔테이션 폴리노미얼, 770
    테이퍼의 원주면을 따라 오리엔테이션, 764
PHU, 455
PL
    스플라인 보간, 667
    폴리노미얼 보간, 681

PLC
-축, 414
PM, 298
PO, 681
PO[PHI]
공구 오리엔테이션 회전, 775
오리엔테이션 폴리노미얼, 770
테이퍼의 원주면을 따라 오리엔테이션, 764
PO[PSI]
공구 오리엔테이션 회전, 775
오리엔테이션 폴리노미얼, 770
테이퍼의 원주면을 따라 오리엔테이션, 764
PO[THT]
공구 오리엔테이션 회전, 775
오리엔테이션 폴리노미얼, 770
PO[XH]
두 접점의 오리엔테이션 지정, 767
오리엔테이션 폴리노미얼, 770
PO[YH]
두 접점의 오리엔테이션 지정, 767
오리엔테이션 폴리노미얼, 770
PO[ZH]
오리엔테이션 폴리노미얼, 770
PO[ZH]:
두 접점의 오리엔테이션 지정, 767
POCKET3 - 사각 포켓
외부 프로그래밍, 1174
POCKET4 - 원호 포켓
외부 프로그래밍, 1178
POLF
NC 제어 후퇴, 1111
나사 절삭 중 고속 후퇴, 256
POLFA, 1111
POLFMASK
NC 제어 후퇴, 1111
나사 절삭 중 고속 후퇴, 256
POLFMLIN
NC 제어 후퇴, 1111
나사 절삭 중 고속 후퇴, 256
POLY, 681
POLYPATH, 681
PON, 1107
PONS, 1098
POS, 126
POSA, 126
POSFS, 1051
POSP, 126
POT, 490
PR, 298
PREPRO, 572
PRESETON, 722
PRESETONS, 723
PRIO, 608
PRLOC, 439
PROTA, 815
PROTD, 817
PROTS, 816
PSI
오리엔테이션 폴리노미얼, 770
테이퍼의 원주면을 따라 오리엔테이션, 764
PTP, 794
PTPG0, 794
PTPWOC, 794
PUD, 440
PUNCHACC, 1098
PUTFTOC, 844
PUTFTOCF, 843
PW, 667

Q

QU, 380

R

RAC, 183
READ, 640
REAL, 439
REAL 상수, 424
REDEF, 446
RELEASE, 988
REP, 464
REPEAT, 525
REPEATB, 525
REPOSA, 972
REPOSH, 972
REPOSHA, 972
REPOSL, 972
REPOSQ, 972
REPOSQA, 972
RET, 574
RET (파라미터 설정 가능), 575
RETB (파라미터 지정 가능), 582
RG, 435
RIC, 183
RINDEX, 503
RMBBL, 972
RMEBL, 972
RMIBL, 972
RMNBL, 972
RND, 271
RNDM, 271
ROT, 348

ROTS, 355
ROUND, 490
ROUNDUP, 496
RP, 194
RPL, 348
RPY, 755
Run MyScreens, 1131

S

S, 99
S 값
    해석, 101
SAR, 298
SAVE, 562
SBLOF, 563
SBLON, 563
SCALE, 360
SCC, 110
SCPARA, 1005
SD, 667
SD41610, 827
SD41611, 827
SD42010, 255
SD42440, 170
SD42442, 170
SD42450$SC_CONTPREC, 961
SD42451 $SC_CONTPREC_G00_ABS, 961
SD42460 $SC_MINFEED (CPRECON의 최소 경로 이송 속도), 962
SD42465, 334
SD42466, 334
SD42475, 782
SD42476, 782
SD42477, 782
SD42900, 835
SD42910, 835
SD42920, 836
SD42930, 837
SD42935, 839
SD42940, 840
SD42984, 906
SD43240, 133
SD43250, 133
SET, 464
SETAL, 1141
SETDNO, 868
SETINT, 608
SETM, 540
SETMS, 99
SETTCOR, 925
SF, 245
Siemens Industry Online Support
    앱, 27
SIN, 490
SINUMERIK, 21
SLOT1- 일반 슬롯
    외부 프로그래밍, 1182
SLOT2 - 원호 슬롯
    외부 프로그래밍, 1186
SOFT, 948
SOFTA, 948
SON, 1098
SONS, 1098
SPCOF, 130
SPCON, 130
SPF, 620
SPI, 995
SPIF1, 1098
SPIF2, 1098
SPLINEPATH, 679
SPN, 1104
SPOF, 1098
SPOS, 131
SPOSA, 131
SPP, 1104
SPRINT, 507
SQRT, 490
SR, 150
SRA, 150
ST, 150
STA, 150
START, 540
STARTFIFO, 965
STAT, 795
STOLF, 202
STOPFIFO, 965
STOPRE, 965
STRING, 439
STRINGIS, 1127
STRLEN, 502
SUBSTR, 504
SUPA
    워크 옵셋, 160
    프레임 선택 해제, 371
SVC, 103
SYNR, 439
SYNRW, 439
SYNW, 439
SZS, 45

T

T0, 71

TAN, 490
TANG, 1072
TANGDEL, 1078
TANGOF, 1077
TANGON, 1076
TCARR, 877
TCOABS, 877
TCOFR, 877
TCOFRX, 877
TCOFRY, 877
TCOFRZ, 877
THETA
    공구 오리엔테이션 회전, 771
    공구 회전 보간 중, 777
TILT, 752
TLIFT, 1075
TMOF, 1108
TMON, 1108
TOFF, 92
TOFFL, 92
TOFFLR, 92
TOFFOF, 886
TOFFON, 886
TOFFR, 92
TOFRAME, 368
TOFRAMEX, 368
TOFRAMEY, 368
TOFRAMEZ, 368
TOLOWER, 502
TOOLENV, 911
TOROT, 368
TOROTOF, 368
TOROTX, 368
TOROTY, 368
TOROTZ, 368
TOUPPER, 502
TOWBCS, 837
TOWKCS, 837
TOWMCS, 837
TOWSTD, 837
TOWTCS, 837
TOWWCS, 837
TRAANG
    각도 프로그램이 가능한, 788
TRACON, 792
TRACYL, 785
TRAFOOF, 808
TRAFOON, 819
TRAILOF, 1014
TRAILON, 1014
TRANS, 342
TRANSMIT, 784
TRAORI, 749
TRUE, 439
TRUNC, 490
TU, 800
TURN, 226

U

ULI, 453
UNTIL, 538

V

VELOLIM, 981
VELOLIMA, 951

W

WAITC, 1051
WAITE, 540
WAITENC, 1004
WAITM, 540
WAITMC, 540
WAITP, 126
WAITS, 131
WALCS<n>, 391
WALCS0, 391
WALIMOF, 388
WALIMON, 388
WCS, 46
    공작물에 정렬, 368
WHEN, 1083
WHEN-DO, 1094
WHENEVER, 1083
WHENEVER-DO, 1094
WHILE, 538
WRITE, 635
WRTPR, 386

X

X..., 191
XOR, 492

Y

Y..., 191

Z

Z..., 191

가

가공 시간, 1120
가공 평면, 38
가로 축, 188
가속 모드, 948

각

각도 옵셋을 적용하여 위치 동기, 1051

간

간접 프로그래밍
    G 코드, 484
    가공 프로그램 라인, 488
    어드레스, 482
    위치 속성, 485

값

값 지정, 53

개

개방형 슬롯 - CYCLE899
    외부 프로그래밍, 1293

건

건너뛰기 레벨, 55

검

검색 경로
    프로그램 가능한 검색 경로, 600

격

격자 또는 프레임 위치 패턴 – CYCLE801
    외부 프로그래밍, 1270

경

경로 가속
    최대, 986
경로 검색
    서브프로그램 호출, 629
경로 계산, 418
경로 기준 공구 오리엔테이션, 774
경로 밀링 - CYCLE72
    외부 프로그래밍, 1208
경로 속도
    최대, 984
경로 지정, 625
경로 지정 및 파라미터를 통한 서브프로그램, 599
경로 축, 412

고

고속 가공 설정 - CYCLE832
    외부 프로그래밍, 1285
고속 후퇴
    나사 절삭, 256
고정 정지점, 400
    접근, 396

공

공구
    M6을 사용해 교체, 72
    T 번호를 사용한 교환, 71
    교환 지점, 40
    그룹, 78
    길이 보정, 75
    반경 보정, 76
    오리엔테이션, 861
    옵셋 메모리, 76
    절삭날, 88
    종류, 78
    종류 번호, 78
    팁, 77
    -파라메타, 828
공구 반경 보정
    CUT2DF, 320
    외경 코너에서, 295
    코너 감속, 703
공구 속도
    최대, 105
공구 오리엔테이션
    경로 기준, 774

공구 옵셋
    마모값에 대한 좌표계, 837
    옵셋, 92
    옵셋 메모리, 828
    추가, 831
공구 체인, 825
공구 캐리어
    오리엔테이션, 877
공구 홀더
    원점, 40
    키네마틱스, 871
공작물, 1142
    계수기, 1123
    디렉토리, 621
    메인 디렉토리, 620
    영점, 39
    형상, 190
공작물 좌표계, 46
공작물 체인, 825
공정 데이터 공유, 1136

교

교육, 27

구

구조
    결정된, 876

극

극 변환, 741
극 좌표, 34, 192
극점 각도, 34
극점 반경, 34

글

글로벌 가공 프로그램 메모리 (GDIR), 622

기

기계
    영점, 39
    -축, 411
기계 좌표계, 41
기능
    사전 정의된, 1524
기본 옵셋, 45
기본 제로 시스템, 45
기본 좌표계, 43
기술 지원, 27
기준 반경, 124
기하
    -축, 409
기하 축
    전환, 997

나

나사
    -다중, 245
    -절삭 G33, 245
    절삭 G34 G35, 252
    체인, 246
    피치, 252
    -회전 방향, 246
나사 밀링 - CYCLE70
    외부 프로그래밍, 1206
나사 선삭- CYCLE99
    외부 프로그래밍, 1253
나사 체인 - CYCLE98
    외부 프로그래밍, 1246
나선 보간:, 226
나선형, 229

내

내부 전처리 정지, 407
내부 코너에서 코너 감속, 704

넌

넌모달, 51

논

논리 연산자, 492

니

니블링, 1098

다

다각형 - CYCLE79
    외부 프로그래밍, 1222

단

단면 나사, 250
단위 시스템, 176
단일 위치, 761

대

대소문자 구분, 623

데

데이터 매트릭스 코드, 28

동

동기
    미세조정, 1054
    일반, 1054
    축, 413
동기 스핀들
    쌍 정의, 1057
    커플링, 1050
동기 태핑 - CYCLE84
    외부 프로그래밍, 1234
동기식 오실레이션
    다음 부분 절입, 1097
    동기 동작, 1094
    반전 지점 범위에서 절입, 1095
    오실레이션 축 및 절입 축 할당, 1094
    절입 이동, 1095
    절입 정의, 1094
    평가, IPO 싸이클, 1097

드

드라이브 이름, 625
드라이브 파라미터 읽기/쓰기, 1011
드레서 좌표계 설정 - CYCLE435
    외부 프로그래밍, 1258
드릴, 81
드릴 나사 밀링 - CYCLE78
    외부 프로그래밍, 1218
드릴링 - CYCLE82
    외부 프로그래밍, 1227
드웰 시간, 405

디

디렉토리 경로, 626

또

또는, 492

라

라운딩, 271

런

런타임
    점검 구조의 응답, 532

로

로터리 축
    거리 벡터, 871
    방향 벡터, 871
    회전 각도, 871

리

리딩 축
    축 마스터 값 커플링, 1037
리밍 - CYCLE85
    외부 프로그래밍, 1238

링

링크
    리드 링크 축, 416
    변수, 437
    -축, 414

마

마스터 값 시뮬레이션, 1042
마스터 값 커플링
    리딩 축과 추종 축 동기화, 1040
    실제값과 지령치 커플링, 1041
마스터 스핀들, 411

매

매크로, 546

메

메모리
    작업, 632
    전처리, 965
    프로그램, 619
메시지, 384
메인 항목, 186

명

명령, 49
    축, 414

모

모달, 51
모든 코너에서 코너 감속, 704
모따기, 271

목

목표점, 190

문

문자 세트, 57
문자열
    길이, 502
    연결, 500
    -연산, 497
    포맷, 507

밀

밀링 공구, 78
밀링 공구 가공 지점, 849
밀링 공구 원점, 849
밀링 공구 팁, 849

반

반경
    적용, 124
반경 프로그래밍, 180
반올림, 496

방

방향 벡터, 756

배

배열, 464
    요소, 464
    정의, 464
배열 인덱스, 467

변

변수
    사용자 정의, 439
    유형 변환, 481, 498
변환
    3축, 4축 및 5축 변환, 749
    연결된 변환, 737
    연쇄, 792
    오리엔테이션 변환, 736
    운동학적 변환, 737
변환 관련 제한사항, 807

보

보간 선삭 - CYCLE806
    외부 프로그래밍, 1276
보링 - CYCLE86
    외부 프로그래밍, 1240
보정
    공구 길이, 75
    공구 반경, 76
    평면, 320
보조 기능 출력
    고속 가공, 380
    보조 기능의 특성, 378
    연속 경로 모드에서, 381
보호 영역, 646

볼

볼록 나사, 260

부

부하에 맞게 조정 - (CYCLE782)
    외부 프로그래밍, 1262

분

분기 라벨
    프로그램 점프용, 520
분모 폴리노미얼, 686

블

블록, 49
    End, 52
    -end LF, 58
    건너뛰기, 54
    길이, 52
    번호, 52
    성명의 순서, 52
블록 건너뛰기, 54
블록 디스플레이
    억제, 568

비

비교 연산자, 492
비동기 오실레이션, 1084

사

사각 변환(TRAANG)
    각도 프로그램이 가능한, 788
사각 스피곳 - CYCLE76
    외부 프로그래밍, 1212
사각 포켓 - POCKET3
    외부 프로그래밍, 1174
사면 플런지 연삭, 790
사용자 XML, 1131

산

산술 파라미터
    전역, 435
    채널별, 433

삽

삽입 깊이, 848

상

상수, 423

서

서브프로그램
    반복, 592
    반환, 파라미터 설정 가능(RET ...), 575
    반환, 파라미터 지정 가능(RETB...), 582
    이름, 551
    적용, 550
    파라미터 전송 없이 호출, 588
    파라미터 전송을 통한 호출, 590
    프로그램 가능한 검색 경로, 600
    호출, 간접, 596
    호출, 모달, 594

선

선삭 - CYCLE951
    외부 프로그래밍, 1303
선삭 공구, 84
선택적 정지, 384
선행 제어, 332

세

세 손가락 법칙, 42

센

센터링 - CYCLE81
    외부 프로그래밍, 1225

소

소재 정의, 1142

속

속도 동기, 1051
속도 커플링, 1054

수

수치 확장, 421

스

스무딩
    오리엔테이션 특성, 783
스위블 평면 / 스위블 공구 / 정렬 공구 – CYCLE800
    외부 프로그래밍, 1265
스케일 팩터, 360
스톡 제거
    지원 기능, 1149
스플라인
    보간, 667
    유형, 675
스플라인 그룹, 679
스핀들
    M 코드, 384
    교체, 988
    메인, 411
    속도 제한, 116
    지정, 99
    포지셔닝, 131
    회전 방향, 99

슬

슬로팅 톱, 87

시

시뮬레이션, 1124
시스템 프레임, 730
시작점, 40
시작점 - 목표점, 190
시작점 옵셋
    나사 절삭, 245

식

식별자, 48
    문자열, 58
    시스템 변수, 58
    특수 숫자 값, 58

실

실제값 커플링, 1054

심

심공 드릴링 1 - CYCLE83
    외부 프로그래밍, 1230
심공 드릴링 2 - CYCLE830
    외부 프로그래밍, 1277

싱

싱글 블록
    억제, 563

알

알람
    NC 프로그램에 설정, 1141

어

어드레스, 420
    값 지정, 53
어드레스 문자, 1450
어드레스 지정, 624

언

언더컷 형상 E 및 F / 언더컷 나사 – CYCLE940
    외부 프로그래밍, 1300
언어 모드, 1148

연

연결
    문자열, 500
연삭 공구, 82
연속 경로 모드, 327

연속 루프, 535
연장된 홀 - LONGHOLE
    외부 프로그래밍, 1189

영

영점
    선삭, 187

오

오른 나사, 247
오리엔테이션 기능이 있는 공구 홀더, 871
오리엔테이션 벡터 THETA, 771
오리엔테이션 변환 TRAORI
    기계 구조, 739
    오리엔테이션 이동, 738
    오리엔테이션 프로그래밍, 750
    오리엔테이션 프로그래밍의 변수, 751
    일반 5/6축 변환, 740
오리엔테이션 축, 761
오리엔테이션 프로그래밍, 761
오실레이션
    동기식 오실레이션, 1090
    동기화 동작을 이용한 제어, 1090
    부분 절입, 1093
    비동기, 1084
    비동기 오실레이션, 1084
오실레이션 모션
    반전 범위, 1093
    반전 지점, 1093
    반전 지점에서 절입, 1095

온

온라인 공구 길이 옵셋, 886

옵

옵셋
    공구 길이, 92, 93
    공구 반경, 92
옵셋 메모리, 828

외

외부 워크 옵셋, 720
외부 프로그래밍, 1338

왼

왼 나사, 247

운

운동학적 유형, 876

워

워크 옵셋
    설정 가능, 160
    외부, 720
    프로그램 가능, 342

원

원 데이터
    계산, 1163
원점, 39, 40
원점 복귀, 395
원주 밀링 - CYCLE77
    외부 프로그래밍, 1216
원통 표면 변환, 741
원통형 나사, 250
원통형 좌표, 195
원호 경로 프로그래밍
    극 좌표, 217
    반경 및 종점, 212
    보간 유형, 209
    중심점 및 종점, 210
    틈 각도 및 중심점, 214
원호 또는 피치원 위치 패턴 - HOLES2
    외부 프로그래밍, 1171
원호 보간
    나선형 보간, 226
원호 슬롯 - SLOT2
    외부 프로그래밍, 1186
원호 포켓 - POCKET4
    외부 프로그래밍, 1178

위

위치 동기, 1051
위치 속성
    간접 프로그래밍, 485
위치 지령치 필터 체인, 962

이

이동 명령어, 189
이동 종료 기준
    프로그램 가능, 704
이송 속도
    경로 축, 120
    규칙, 117
    단위, 123
    동기 축, 122
    -속도, 206
    오버라이드, 146
    -오버라이드, 141
    인버스-타임, 121
    포지셔닝 축, 137
    핸드휠 오버라이드 사용, 144
이진 상수, 425

인

인터럽트 루틴
    비활성화/활성화, 611
    삭제, 611
    새롭게 할당, 610
    프로그래밍 이송 방향, 615
    형상에서 고속 후퇴, 613
    후퇴 이동, 615

일

일반 데이터 보호 규정, 28
일반 슬롯 - SLOT1
    외부 프로그래밍, 1182

임

임의 위치 - CYCLE802
    외부 프로그래밍, 1272

자

자동 경로 분할, 1103
자동 중단 지점, 972

작

작업, 49
작업 메모리, 632
작업 순서
    사전 정의된, 1491
작업 영역 제한
    BCS에서, 388

잔

잔여 시간
    공작물, 1122

장

장 구조, 22

저

저크
    옵셋, 983
    제한, 948

적

적용 대상, 22

전

전자 기어, 1043
전처리
    -메모리, 965
전처리 정지
    내부, 407
전환 가능 기하 축, 997

절

절단 - CYCLE92
    외부 프로그래밍, 1242
절대 치수, 35
절삭 속도, 103
절삭날
    -관련 위치, 322
    반경, 77
    번호, 89
    -원점, 322
    위치, 77
    중심점, 77
    형상 공구 개수, 319
절삭날 번호, 867

점

점검
    구조, 532
점프
    마커 (라벨)로 점프, 519
    프로그램 시작으로, 517
점프 마커
    프로그램 섹션 반복, 525

접

접근 지점/각도, 291

정

정위치 정지, 324
정지
    NC 제어, 1115
    드라이브 자동, 1116
    사이클 끝에서, 384
    옵션, 384
    프로그램된, 384

제

제로 시스템
    기본, 45
    설정 가능, 45
제로 프레임, 161
제품 지원, 26

조

조각 - CYCLE60
    외부 프로그래밍, 1192

좌

좌표
    극점, 34
    원통형, 195
    직교 좌표계, 32
좌표 변환 (프레임), 46
좌표계
    개요, 41
    기본, 43

주

주속 일정 제어, 110

중

중간점 및 종점을 가진
    원호 보간, 219
중첩 깊이
    점검 구조, 532

증

증분 치수, 37, 169
증분치, 37

지

지령치 커플링, 1054

직

직경 프로그래밍, 180
직교 PTP 이송, 793
직교 좌표, 32
직선
    -보간, 206

차

차단점, 39

채

채널
    축, 412

초

초기 공구 오리엔테이션 설정 ORIRESET, 751
초기화
    배열, 464
초기화 프로그램, 632

추

추적 불가능한 섹션의 획득 및 찾기, 971

추종 축
축 마스터 값 커플링, 1037

축

축
PLC, 414
경로, 412
교체, 988
기계, 411
기하, 409
동기, 413
리드 링크 축, 416
링크, 414
메인, 409
명령, 414
-유형, 408
채널, 412
커플 모션, 1017
컨테이너, 415
특수, 410
포지셔닝, 412
축 관련 마스터 값 커플링, 1037

출

출력
외부 장치/파일로, 1136

충

충돌 탐지, 315

측

측정 싸이클 파라미터
CYCLE961, 1373
CYCLE971, 1398
CYCLE973, 1343
CYCLE974, 1346
CYCLE976, 1356
CYCLE977, 1368
CYCLE978, 1359
CYCLE979, 1376
CYCLE982, 1394
CYCLE994, 1351
CYCLE995, 1384
CYCLE996, 1388
CYCLE9960, 1391
CYCLE997, 1381
CYCLE998, 1364

치

치수
로터리 축 및 스핀들, 174
반경 단위, 180
옵션, 167
직경 단위, 180

칩

칩 제거 / 잔여 칩 제거 / 플런지 절삭 / 잔여 플런지 절삭 / 플런지 선삭 / 잔여 플런지 선삭 – CYCLE952
외부 프로그래밍, 1306

카

카운트 루프, 536

커

커플 모션, 1014
커플 모션 축, 1017
커플 축 조합, 1014
커플링
일반, 1062
커플링 상태
축 마스터 값 커플링, 1042
커플 모션 중, 1018
커플링 유형, 1054
커플링 팩터, 1014

코

코멘트, 54

클

클램핑 토크
-고정 정지점, 403

타

타사 웹 사이트, 23

태

태핑 척을 사용한 태핑 - CYCLE840
    외부 프로그래밍, 1289

테

테이퍼 나사, 251

특

특수 공구, 86
특수 문자, 58
특수 축, 410

파

파라미터
    기계, 828
    서브프로그램 호출 시 전송, 555
    서브프로그램 호출을 위한 전송, 590
    실제, 554
    유형, 554
파일
    -정보, 644
파일 이름, 627

펀

펀치 테이프 형식, 49
펀칭, 1098

페

페이스 밀링, 758
페이스 밀링 - CYCLE61
    외부 프로그래밍, 1196

포

포인트-투-포인트 이송, 793
포지셔닝 축, 412

폴

폴리노미얼 계수, 683
폴리노미얼 보간, 681

표

표준 범위, 23

프

프레임, 337
    -명령문, 339
    -미러링, 363
    -배율, 360
    선택 해제, 371
    시스템, 730
    전역, 729
    채널별, 730
    -체인, 735
    프레임 체인, 718
    할당, 717
    호출, 716
프레임 변수
    값 할당, 713
    사전 정의된 프레임 변수, 709
    좌표 변환 호출, 707
프레임 콤포넌트
    FI, 715
    MI, 715
    RT, 715
    SC, 715
    TR, 715
프로그램
    런타임, 1119
    메모리, 621
    반복, 592
    분기, 522
    시뮬레이션, 1124
    어드레스 지정, 624
    이름, 48
    점프, 519
    종료, 51
    초기화, 632
    헤더, 59
프로그램 루프
    IF loop, 534
    REPEAT 루프, 538
    WHILE 루프, 538
    루프의 끝, 535
    카운트 루프, 536
프로그램 메모리
    파일 유형, 620
    표준 디렉토리, 620
프로그램 블록 시작 - GROUP_BEGIN, 1338

프로그램 블록 종료 - GROUP_END
    외부 프로그래밍, 1339
프로그램 섹션
    -반복, 525
프로그램 섹션 반복
    CALL 간접 프로그래밍, 597
프로그램 정지, 384

피

피드백 보내기, 25

핸

핸드휠
    오버라이드, 144

행

행 위치 패턴 - HOLES1
    외부 프로그래밍, 1170

형

형상
    -element, 189
    계산기, 235
    리포지셔닝, 972
    접근 / 출발, 288
    -준비, 1150
    -코딩, 1157
    테이블, 1150
형상 요소
    이동, 1162
형상 절삭 - CYCLE95
    외부 프로그래밍, 1244
형상 정의 프로그래밍, 234
형상 정확도
    프로그램 가능, 960
형상 준비
    에러 피드백 신호, 1165
형상 코너
    라운딩, 271
    모따기, 271
형상 트루잉 - CYCLE495
    외부 프로그래밍, 1259
형상 포켓 밀링 / 형상 포켓 잔삭 가공 / 형상 스피곳 밀링 / 형상 스피곳 잔삭 가공 – CYCLE63
    외부 프로그래밍, 1199
형상 포켓 사전 드릴링 - CYCLE64
    외부 프로그래밍, 1203
형상 호출 - CYCLE62
    외부 프로그래밍, 1198
형상에서 고속 후퇴, 613

홀

홀드 블록, 971

홈

홈 - CYCLE930
    외부 프로그래밍, 1296

확

확장 어드레스 표기, 421

회

회전
    오리엔테이션 벡터, 771
    프로그래밍 가능, 348
회전 벡터 보간, 772
회전하는 직선 축을 통한 변환, 748

후

후퇴
    NC 제어, 1111
    나사 절삭 방향, 257
    드라이브 자동, 1117