



NC154 / NC156 / NC158 AXIS CONTROLLER CNC Programming Introduction - CNC 프로그래밍 안내서-

**Rev. 2.0 (December 12, 1998)
Model No. MA4CNC1.E**

We reserve the right to change the content of this manual without prior notice. The information contained herein is believed to be accurate as of the date of publication, however, B&R makes no warranty, expressed or implied, with regards to the products or the documentation contained within this document. B&R shall not be liable in the event if incidental or consequential damages in connection with or arising from the furnishing, performance or use of these products. The software names, hardware names and trademarks used in this document are registered by the respective companies.

I Versions

Version	Date	Comment	Edited by
1.0	Oct 16, 2015	First Translation (Korean V.1.00)	Sunggu Kang

Table 1: Versions

II Distribution

Name	Company, Department	Amount	Remarks

Table 2: Distribution

III Table of Contents

1 NC-프로그램에 대한 일반적인 정보	5
2 Main Program / Subprogram	5
2.1 Local subprogram 의 정의.....	5
2.2 Main program 의 정의	5
2.3 Local subprograms 호출	6
2.4 Global subprogram 호출	6
3 G-Functions	7
3.1 Overview	7
3.2 G00 / G01 / G02 / G3: 보간법 타입 (Interpolation type)	8
3.2.1 G00: 급속 보간법 (Rapid Interpolation).....	8
3.2.2 G01: 선형 보간법(Liner Interpolation)	8
3.2.3 G02 / G03: 시계 / 반시계 방향 원형 보간법 (Circular Interpolation Clockwise/Counter Clockwise)	9
3.2.3.1 반경 진입을 이용한 원형 프로그래밍 (starting with V4.00).....	10
3.3 G04 / \$TIME: Dwell.....	10
3.4 G17 / G18 / G19: 평면 선택	11
3.5 G20 / ... / G23: Mirroring.....	11
3.6 G25: Tangential Transition Arcs (starting with V4.00)	12
3.7 G40 / G41 / G42: Cutter Diameter Compensation.....	13
3.7.1 Transitions 예제	14
3.8 G53 / ... / G59: 절대 영점 오프셋(Absolute Zero Point Offset).....	16
3.9 G60: Accuracy Hold	17
3.10 G90 / G91: 절대 /상대 위치 좌표	17
3.11 G92: 프로그램 된 영점 오프셋	18
3.12 G108 / G109 / G110: 가속 / 감속 설정 (Starting with V5.26)	18
3.13 G125: Rounding Edges (Starting with V5.20).....	19
3.14 G138 / G139: CDC Selection and Deselection 의 타입.....	20
3.14.1 G138: Direct CDC Selection / Deselection.....	20
3.14.2 G139: CDC Selection / Deselection with Linear Transition(s)	21
3.15 G159: Expanded Zero Point Offset.....	22
3.16 G161 / G16: 원 중심의 절대 / 상대 좌표	22
3.17 G170: 디코더 동기화.....	23
3.18 G180: Contour Elements 연결 (Starting with V5.00)	23
3.18.1 G180=000: Tangential Line-Circle-Programming.....	23
3.19 G200: 측정을 위한 Trigger Event 후 Position Latch (starting with V5.26)	26
3.20 G201: 트리거 이벤트 이후 포지션 래치 그리고 중단 (starting with V5.26)	29
4 M - Function	32
4.1 동기화된 M – Function.....	32

4.2 비동기화된 M – Function	32
4.3 미리 설정된 M-Function	33
4.3.1 M00: Programmed Stop	33
4.3.2 M01: Optional Stop	33
4.3.3 M02 / M29 / M30: 프로그램 / 서브 프로그램 종료	33
5 Technology Parameters	34
5.1 Feed	34
5.2 Spindle RPM (S-Function)	34
5.3 톨 번호 (T-Function).....	35
5.4 톨 데이터 번호.....	35
6 좌표 프로그래밍	37
6.1 위치 좌표.....	37
6.2 원 중심에 대한 좌표	37
7 파라미터.....	39
7.1 Direct 파라미터 (R 파라미터).....	39
7.2 Indirect 파라미터 (P 파라미터).....	39
7.3 외부 파라미터 (starting with V5.10)	40
8 Comments	40
9 수학연산자.....	41
9.1 기본 연산자	41
9.2 카운팅 기능	41
9.3 기하학적 기능.....	41
9.4 컨버전 기능	41
9.5 현재 Cutter Radius 구하기.....	41
10 Control Blocks	42
10.1 조건	42
10.1.1 IF 문	43
10.1.2 ELSEIF 체인	43
10.2 SWITCH 문	44
10.3 FOR 문	45
10.4 WHILE 문	45
10.5 DO 문	46

1 NC-프로그램에 대한 일반적인 정보

NC 프로그램은 CNC 기계의 처리 순서를 관리합니다. 이 프로그램은 DIN 66025 또는 ISO 6983 에 맞춰 구성된 명령어(NC records) 시퀀스로 구성된 ASCII 형식의 문자입니다.

NC 프로그램이 *ncaction(ncsys, ncPROGRAMM, ncSTART)*를 포함한 PCC 에 의해 작동되기 전에, ncCNC_PROG 타입을 가지고 있는 NC 데이터 모듈은 반드시 프로그램 텍스트로 만들어 저장하며 *ncaction(ncsys, ncDAT_MOD, ncDOWNLOAD)*으로 전달되어야 합니다.

2 Main Program / Subprogram

NC 프로그램 안에 있는 NC Record 는 Blocks, Local subprograms, main program 으로 구성되어 있습니다.

2.1 Local subprogram 의 정의

Local subprogram 은 “%L<prognumber>” 으로 정의 됩니다. Subprogram 은 M02, M29, M30 나 다음 프로그램 정의(%)하는 것으로 끝납니다. 하나의 NC 프로그램에서 최대 50 개의 (V5.26: 20 까지) Local subprogram 을 정의 할 수 있습니다. “prognumber”는 0 에서 ($2^{31} - 1$)까지 모든 숫자를 입력 할 수 있습니다.

Syntax : %L<prognumber>

Example: %L120

... (Sequence of NC records)
M02 / M29 / M30

Note: 모든 Local subprogram 은 Main program 전에 정의되어야 합니다.

2.2 Main program 의 정의

Main program 또는 global subprogram 은 “%<prognumbe>”으로 정의됩니다. 만약 Local subprogram 이 존재하지 않는다면 이 프로그램의 정의가 실패 할 수 있습니다. Main program 은 M02, M29, M30 으로 끝이 납니다. “prognumber”는 0 에서 ($2^{31} - 1$)까지 모든 숫자를 입력 할 수 있습니다.

Syntax : %<prognumber>

Example: %100

... (Sequence of NC records)
M02 / M29 / M30

Note: 에러가 발생한 경우에는 program number “prognumber”가 에러 기록에 입력됩니다. 이것이 Main program 을 위한 프로그램 정의가 항상 필요한 이유입니다.

2.3 Local subprograms 호출

Local subprogram 은 main program 을 포함하고 있는 NC 프로그램 텍스트의 한 부분입니다. 그리고 main program 과 다른 local subprogram 에서 호출되어 집니다.

Syntax : *LLprognumber*

Example: LL120
... (Next NC block)

Local subprogram “*prognumber*”가 호출됩니다. Subprogram 이 종료되었을 때, NC 프로그램의 진행이 subprogram 호출을 뒤따르는 NC record 와 함께 계속됩니다.

2.4 Global subprogram 호출

원칙적으로 Global subprogram 은 다른 프로그램에서 호출 될 수 있는 main program 과 나뉘져 있습니다.

Syntax : *Lpname*

Example: Lcontour1
... (Next NC block)

데이터 모듈 이름 “*pname*”을 가진 global NC 프로그램이 호출 됩니다. Subprogram 이 종료되었을 때, NC 프로그램의 진행이 subprogram 호출을 뒤따르는 NC record 와 함께 계속됩니다.

Note: Subprogram 호출은 중첩 될 수 있습니다. 최대 중첩 깊이는 20 입니다.

3 G-Functions

3.1 Overview

G-Function	Modal Valid	Record Valid	Group	Meaning
G00	x		1	Rapid Interpolation
G01	x		1	Linear Interpolation
G02	x		1	Circular Interpolation - Clockwise
G03	x		1	Circular Interpolation - Counter Clockwise
G04		x		Dwell in sec
G17	x		2	Plane Selection XY
G18	x		2	Plane Selection ZX
G19	x		2	Plane Selection YZ
G20	x		3	Deactivate Mirroring
G21	x		3	Mirroring over the Y Axis
G22	x		3	Mirroring over the X Axis
G23	x		3	Mirroring over the X Axis and Y Axis
G25	x		11	Tangential Transition Arcs
G40	x		5	Deactivate the CDC
G41	x		5	CDC Left
G42	x		5	CDC Right
G53	x		6	Deactivate Zero Point Offsets
G54	x		6	Zero Point Offset 1
G55	x		6	Zero Point Offset 2
G56	x		6	Zero Point Offset 3
G57	x		6	Zero Point Offset 4
G58	x		6	Zero Point Offset 5
G59	x		6	Zero Point Offset 6
G60		x		Accuracy Hold
G90	x		8	Absolute Position Coordinates
G91	x		8	Relative Position Coordinates
G92	x			Programmed Zero Point Offset
G108	x		12	Set Acceleration
G109	x		13	Set Deceleration
G110	x		12, 13	Set Acceleration and Deceleration
G125	x		11	Rounding Edges
G138	x		9	CDC Activate/Deactivate Directly
G139	x		9	CDC Activate/Deactivate with Transitions
G159=0 - 199	x		6	Expanded Zero Point Offset
G161	x		10	Absolute Coordinates for Center of Circle
G162	x		10	Relative Coordinates for Center of Circle
G170		x		Decoder- Synchronization
G180=0 - 199		x		Connecting Contour Elements
G200		x		Position Latch
G201		x		Position Latch with following Halt

Modal valid G-Function 은 같은 그룹에서 다른 것에 의해 비활성화 되기 전까지 유효합니다.

Record valid G-Function 은 프로그래밍 된 record 안에서만 유효합니다.

3.2 G00 / G01 / G02 / G3: 보간법 타입 (Interpolation type)

3.2.1 G00: 급속 보간법 (Rapid Interpolation)

Example: G00 G90 X100 Y100 Z100

급속 보간법은 G00 와 타겟 위치 진입 경로 정보를 이용하여 프로그래밍합니다. CNC 축의 한계 값(limit values)을 고려하여 가능한 최대 속도 및 가속도로 타겟 위치에 직선으로 접근합니다. 궤적 속도 및 가속도에 대한 CNC 리미트 값이 초과 될 수도 있습니다.

초과하지 않는 한계:

PCC: axis_dat.limits.v
axis_dat.limits.a

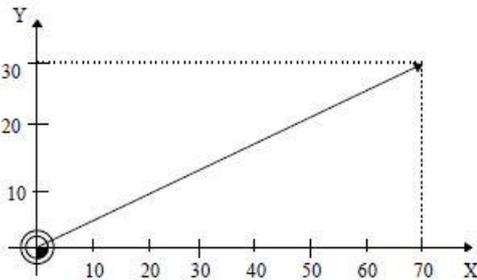
초과 할 수 있는 한계:

PCC: cncsys_dat.limit.v
cncsys_dat.limit.a_pos or cncsys_dat.limit.a_neg

이전 Feed(F-Word)는 tact 에 남아있고 다른 경로 정보를 가지고 다시 활성화 됩니다 (e.g.: G01).

3.2.2 G01: 선형 보간법(Liner Interpolation)

Example: G90 G01 X70 Y30 F200



선형 보간법은 G01, 타겟 위치 진입, 궤적 feed 의 경로 정보를 사용하여 프로그래밍 합니다.

선형 보간법은 아래의 움직임을 발생시킵니다...

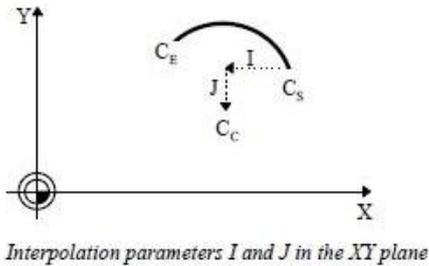
- ...하나 또는 그 이상의 축 방향 안에
- ...시작 위치부터 절대값 또는 상대값으로 정해진 타겟 위치까지
- ...프로그램 된 feed 를 가진(F-Word; 단위는 **units/min** 입니다)

만약 선형 보간법이 다른 움직임 타입에 의해 실행되지 않는다면, 타겟 위치에 접근할 때 궤적 속도는 ramp 형태로 0 까지 줄어듭니다. 또한 속도와 가속도는 정해진 CNC 한계 값이 초과되지 않는 지점까지 줄어듭니다.

PCC: cncsys_dat.limit.v_jump
cncsys_dat.limit.a_jump
cncsys_dat.limit.v_traject **units/sec!**

3.2.3 G02 / G03: 시계 / 반시계 방향 원형 보간법
(Circular Interpolation Clockwise/Counter Clockwise)

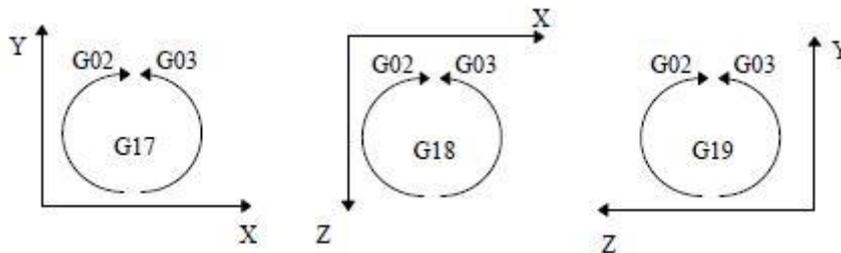
Example: G03 X200 Y200 I-50 K-50 F200



- C_s 원형의 Interpolation 시작 위치
- C_E 원형의 Interpolation 끝나는 위치
- C_C 원의 중심
- I/J X/Y 축에 대한 Interpolation 파라미터
(원형 움직임의 시작 위치에서 X/Y 방향에서 원의 중심의 상대 위치)

Axis	Interpolation Parameter
X	I
Y	J
Z	K

원형 보간법은 경로 정보 G02 또는 G03 그 다음 파라미터 (타겟 위치, 원의 중심, 그리고 궤적 feed)를 입력하여 프로그래밍 합니다. 타겟 위치에 프로그램 된 속도로 접근합니다. 다양한 평면에서 회전 방향은 다음과 같이 정의됩니다.



이 평면의 아래쪽으로는 그 면에 수직인 축을 보여줍니다 (e.g. 첫번째 그림에서 Z 축은 아래 방향이고, 두번째 그림에서는 Y 축이, 세번째 그림에서는 X 축이 아래 방향입니다). G02 는 시계방향으로 움직이고 G03 는 반시계 방향으로 움직입니다.

원의 중심은 보간법 파라미터 I, J, K 에 의해 결정됩니다. 이러한 파라미터는 상대적인 시작 위치(G162) 또는 절대 좌표(G161)을 중 하나를 사용합니다.

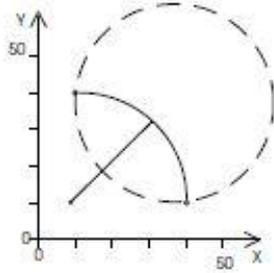
```
PCC: cncsys_dat.decoder.parameter.center = ncABSOLUTE
ncRELATIVE
```

보간법 파라미터 I, J, K 는 하나의 record 에만 유효합니다. 만약 파라미터가 입력되지 않는다면 0 이 입력된다는 의미입니다. => 절대 중앙 좌표를 사용할 때 경고가 발생합니다.

3.2.3.1 반경 진입을 이용한 원형 프로그래밍 (starting with V4.00)

반경 프로그래밍으로 반원을 위한 하나의 방법이 있습니다 ($\alpha = 180^\circ$ 를 포함한 각도). 다른 경우를 위한 두가지 방법이 가능합니다. ($\alpha < 180^\circ$ 인 하나의 각도와 $\alpha > 180^\circ$ 인 하나의 각도). 반경 진입을 위한 신호는 원호가 선택되었는지 확인합니다. 만약 반경 진입이 양이면, 작은 원호($\alpha < 180^\circ$)가 선택됩니다. 음의 반경은 큰 원호($\alpha > 180^\circ$)가 선택됩니다.

Example:



시작 위치 "X10, Y40"에서 다음의 명령은 왼쪽의 윤곽을 따릅니다.

```
G02 X40 Y10 R = 30
```

만약 R = -30 이 입력된다면, 점선으로 된 원 위에 큰 원호가 선택됩니다.

Note: 완전한 원은 이러한 프로그래밍 타입으로 만들 수 없습니다.

3.3 G04 / \$TIME: Dwell

Syntax: G04 *time* or \$TIME *time*

"time"에는 아래의 내용을 입력할 수 있습니다:

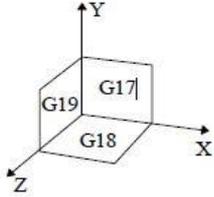
- 모든 숫자 또는 소수점
- 직접, 간접 또는 외부 파라미터
- 산술식

시간은 초 단위(seconds)로 입력됩니다. 추가되는 함수는 같은 record 에서 사용할 수 없습니다. CNC 시스템은 특정 시간동안 NC 프로그램 작동을 중지시킵니다. 그리고나서 NC 프로그램 의 다음 명령을 수행합니다.

이 함수는 예를 들어 NC 프로그램에서 스피들이 작동하면서 특정 시간동안 feed 동작을 멈추게 하는데 사용될 수 있습니다. 이것은 breaking shavings 과 free cutting 을 위해 사용됩니다.

3.4 G17 / G18 / G19: 평면 선택

이 명령에는 Cutter diameter compensation(CDC)와 원형 보간에 대한 메인 평면이 지정됩니다.



G-Function	Plane
G17	XY
G18	ZX
G19	YZ

The three main planes in the cartesian coordinate system

3.5 G20 / ... / G23: Mirroring

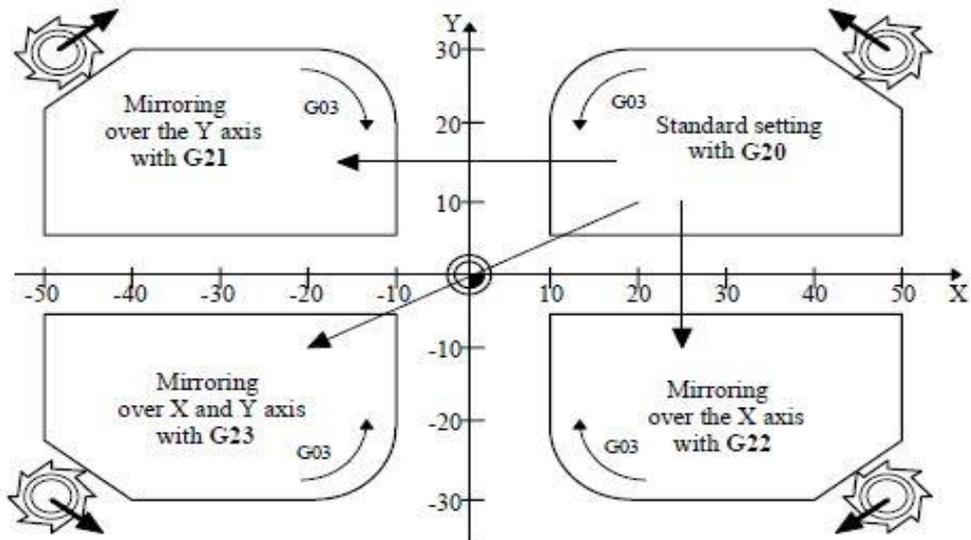
Mirroring은 X 축(G22), Y 축(G21) 그리고 X와 Y 축(G23)으로 가능합니다. Mirroring은 영점(Zero point)에서 발생합니다. Mirrored 축에 대한 신호가 변경 됩니다.

Note: 이것은 서브 프로그램 처럼 미러링되는 프로그램 섹션을 실현하는 것을 추천합니다. 이러한 서브 프로그램은 각각의 미러 기능이 활성화 된 후에 호출 될 수 있습니다.

Example:

```

LL10 (Call subprogram 10; process the contour)
G22 (Mirroring over X axis)
LL10 (Call subprogram 10; process the mirrored contour)
G20 (Deactivate mirroring)
    
```



Principle of mirroring over X and/or Y axes

축을 기준으로 미러링 할 때, CDC 방향(G41 => G42; G42 => G41), 원형 보간을 위한 회전 방향(G02 => G03; G03 => G02) 그리고 각각의 보간 파라미터를 위한 신호들이 변경 됩니다. 길이 보정 값, 축 오프셋 그리고 영점 오프셋은 미러링 되지 않습니다.

미러링을 위해 영점은 오프셋 될 수 있습니다(zero point offsetting 을 참고하세요). 그러면 미러링은 오프셋 좌표 시스템의 축을 통해 이뤄집니다.

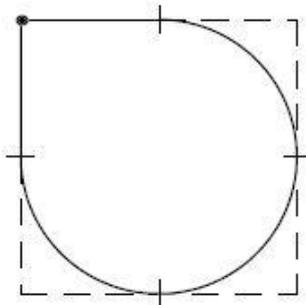
만약 두 축이 미러링 되었다면 (G23), 미러링이 두 축의 변경된 신호에 의해 발생합니다. 이 경우에는 Cutter diameter correction(CDC)와 원형 보간을 위한 회전 방향은 변경되지 않습니다.

3.6 G25: Tangential Transition Arcs (starting with V4.00)

Activate G25 Radius
 Deactivate: G25 0

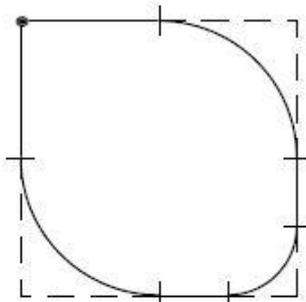
최근, Tangential transitions arcs 가 선형 구간에 포함되었습니다. 이 기능을 활성화시킨 후에, X, Y, Z 축의 경로가 선택된 메인 평면에서 움직이고, 기능이 비활성화 될 때까지 transitions arcs 는 평면에 만들어 집니다. 이 반경은 함수를 재활성화 시켜서 변경 할 수 있습니다. 메인 평면 위에 없는 축의 경로는 수정되지 않습니다. 경로가 없는 Record (e.g. G04)는 tangential arcs 의 끝에서 출력됩니다. 만약 tangential arcs 함수(G17, G18, G19)가 활성화 되어있는 동안 평면이 변경되었다면, 이 함수는 자동으로 비활성화 됩니다. 새로운 평면에서 적절한 경로를 가진 record 가 함수를 다시 활성화 시킵니다. 평면 변경은 이 record 를 가지고 시작하게 됩니다.

Examples:



The following instructions create the contour shown on the left:

```
G91
G01 X 100 G25 50
G01 Y-100
G01 X-100
G01 Y 100 G25 0
```

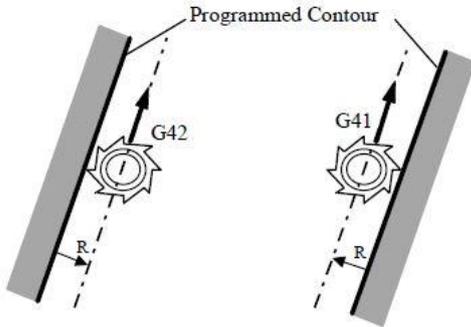


The following instructions create the contour shown on the left:

```
G91
G01 X 100 G25 50
G01 Y-100 G25 25
G01 X-100 G25 50
G01 Y 100 G25 0
```

3.7 G40 / G41 / G42: Cutter Diameter Compensation

Cutter diameter correction (CDC)는 고려 사항을 통해 획득한 cutter diameter 가진 재료의 왼쪽(G41) 또는 오른쪽(G42)에 수정된 궤도를 계산합니다.



Cutter diameter Correction with G41 and G42

G40: 비활성화 CDC

Cutter diameter correction 이 비활성화 됩니다. 하지만 작업 평면 위에 축 움직임이 적어도 한번은 프로그램 된 후에 취소됩니다.

G41: CDC Left

툴을 위한 Cutter diameter correction 은 작업 구간 왼쪽에서 발생합니다.

G42: CDC Right

툴을 위한 Cutter diameter correction 은 작업 구간 오른쪽에서 발생합니다.

Cutter diameter 수정은 선택된 평면 (G17 / G18 / G19)위에서 발생합니다. Cutter diameter 는 툴 데이터 번호(D nnn)에 의해 결정됩니다.

PCC: 툴 데이터 파라미터는 nca_tptab 데이터 타입을 포함한 데이터 테이블에서 정의됩니다.

```

Radius:          tptab.record[i].radius      = ...
Length correction: tptab.record[i].length    = ...
Axis offsets:    tptab.record[i].offset[0]   = ...
                 tptab.record[i].offset[1]   = ...
                 tptab.record[i].offset[2]   = ...
    
```

Cutter diameter correction 을 포함한 NC 프로그램이 작동되기 전, ncTOOL 타입을 가진 NC 데이터 모듈은 이 데이터 테이블을 포함해서 생성되어야 하고, ncaction(ncsys, ncDAT_MOD, ncDOWNLOAD)를 포함한 NC 모듈로 전달되어야 하며 ncaction(ncsys, ncTOOL_TAB, ncSET)을 포함하여 활성화 되어야 합니다.

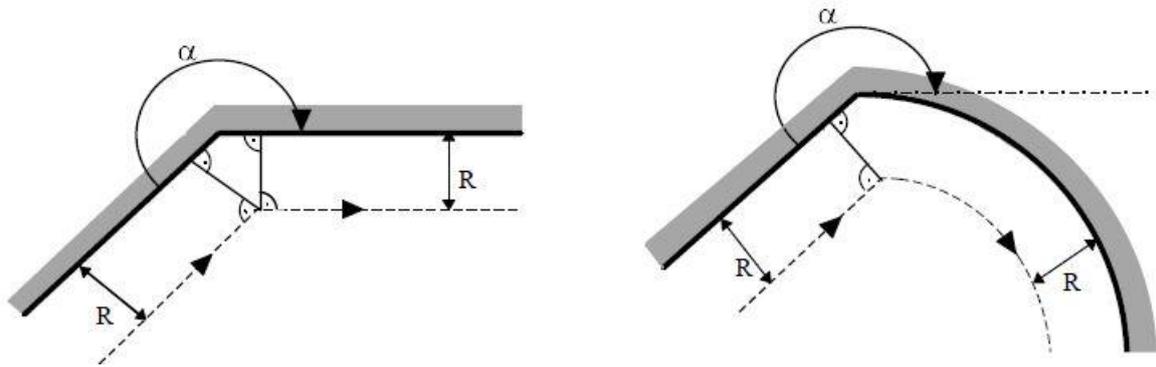
Cutter diameter correction 선택은 경로 정보 없이 record 에서 프로그램 될 수 있습니다; 하지만 적어도 하나의 축이 선택된 평면 위에서 프로그램 되어 적용됩니다. 활성화는 선형 record (G00 또는 G01)에서만 허용됩니다.

3.7.1 Transitions 예제

CDC 를 위해, 현재 record 에서 두 개의 record 가 컨트롤러에 의해 미리 읽혀집니다. 모든 Transition 을 위해 수정된 궤도의 교차점이 계산됩니다. 아래의 그림은 다양한 교차점들의 수정 사항을 보여줍니다.

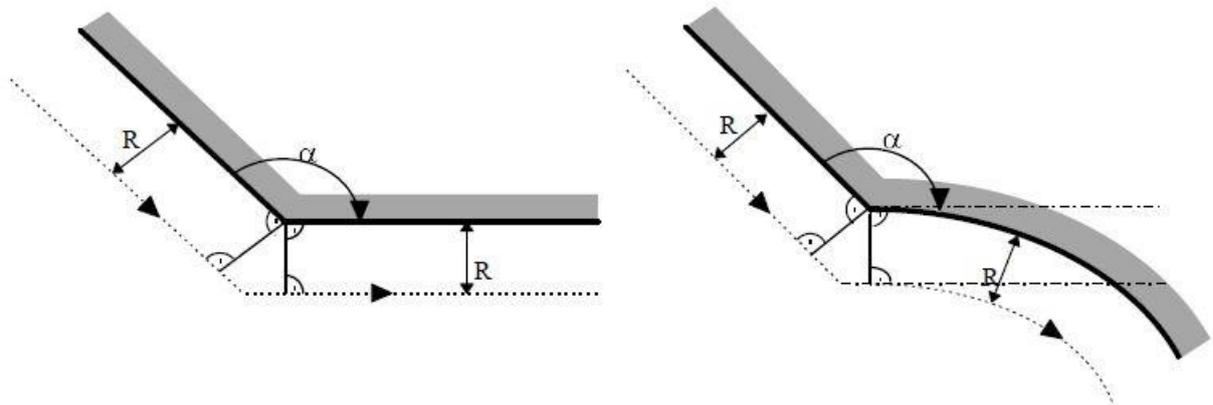
$\alpha > 180^\circ$ 인 Transition :

만약 $\alpha > 180^\circ$ 이라면, transition 이 삽입 될 수 없습니다. 수정된 두 개의 궤도의 교차 지점이 계산된다.



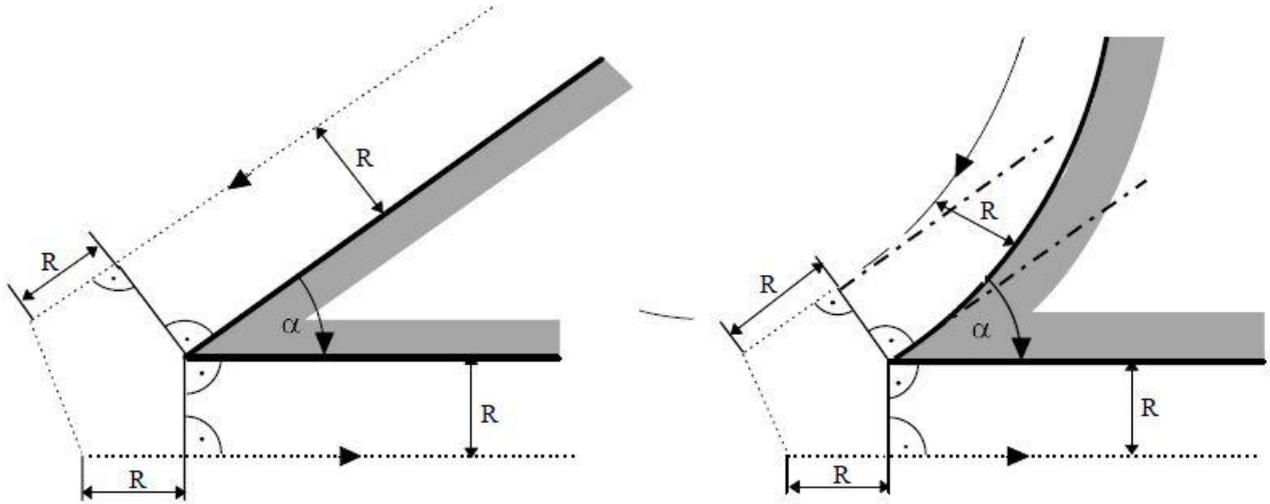
$90^\circ \leq \alpha \leq 180^\circ$ 인 Transition :

만약 각도가 90° 와 180° 사이라면, transitions 이 삽입됩니다. 이것은 수정된 궤도가 일정하게 교차지점으로 확장되었다는 것을 의미합니다.



$\alpha < 90^\circ$ 인 Transition :

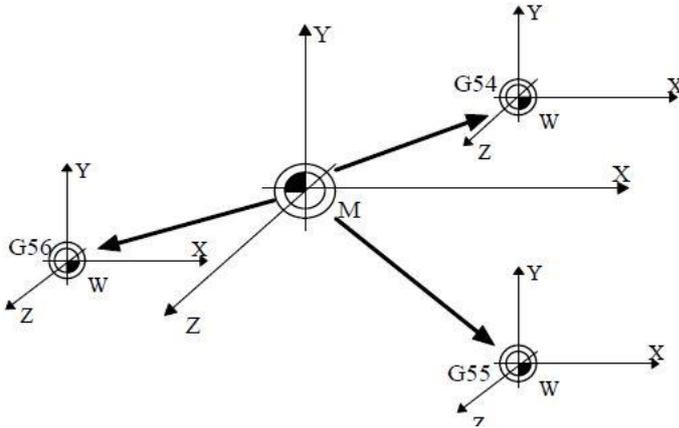
만약 α 가 90° 보다 작다면, 수정된 궤도가 Cutter 반지름 길이에 의해 일정하게 확장됩니다. 수정된 궤도는 다른 선형 구간과 서로 연결됩니다.



3.8 G53 / ... / G59: 절대 영점 오프셋(Absolute Zero Point Offset)

영점 오프셋은 기계 영점과 공작물 영점(Work piece zero point)의 차이이다. 기계 영점은 기계 좌표 시스템의 영점으로 지정되어 있다.

Note: 절대축 좌표의 진입은 항상 지정된 공작물 영점을 참고합니다.



공작물 영점, W 는 영점 오프셋을 사용하는 프로그래머에 의해 정의 될 수 있습니다. 최대 6 개의 영점 오프셋이 이 선택을 사용하여 영점 오프셋 테이블으로 정의 할 수 있습니다. 영점 오프셋은 항상 기계 영점, M 을 참조하여 절대 좌표처럼 결정됩니다.

G53: 영점 오프셋 비활성화

기계 영점이 영점으로 설정됩니다.

G54 ~ G59: Record 1- 6 을 포함한 영점 오프셋

각각의 테이블 record 의 값이 영점으로 설정됩니다.

Note: 함수 G159 = i 는 더 많은 영점 오프셋 record 를 활성화하는데 사용 할 수 있습니다.

PCC: 영점 오프셋이 ncda_zptab 데이터 타입을 포함한 데이터 테이블에서 정의 됩니다:

```

zptab.record_cnt = 6
zptab.startindex = 1

zptab.record[0].offset[0] = ...    X    Values for record 1    G54
zptab.record[0].offset[1] = ...    Y
zptab.record[0].offset[2] = ...    Z
...
zptab.record[5].offset[0] = ...    X    Values for record 6    G59
zptab.record[5].offset[1] = ...    Y
zptab.record[5].offset[2] = ...    Z
    
```

영점 오프셋을 포함한 NC 프로그램이 작동하기 전에, ncZERO_DAT 타입을 포함한 NC 데이터 모듈은 이러한 데이터 타입으로 생성되고, naction(ncsys, ncDAT,_MOD, ncDOWNLOAD)를 포함한 NC 모듈로 전송되고, naction(ncsys, ncZERO_TAB, ncSET)을 가지고 활성화 되어야합니다.

3.9 G60: Accuracy Hold

만약 accuracy hold 가 NC 동작 record(G00, G01, G02, G03)안에 프로그램 되었다면, Feed 는 record 의 끝 부분에서 Ramp 의 형식으로 0 까지 줄어듭니다. 만약 axis / axes 가 accuracy hold 허용 오차 안에 있다면 NC 프로그램은 일시적으로 머무는 시간 이후에 계속됩니다.

```
PCC: x_axis_dat.controller.inpos_tol ... Accuracy hold tolerance
      x_axis_dat.controller.t_inpos ... Transient time
```

이 경우에, non-tangential transitions 또는 방향 변경, 가속도 변경은 최소화 됩니다.

Note: accuracy hold 는 하나의 record 에서만 유효합니다.

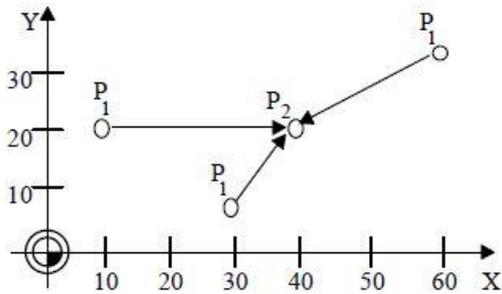
3.10 G90 / G91: 절대 /상대 위치 좌표

좌표 시스템에서 위치 구동은 절대좌표 또는 상대좌표를 이용하여 명시 할 수 있습니다.

G90: 절대 위치 좌표

좌표의 진입이 절대적 입니다. 이것은 주어진 값이 현재 영점을 참고한다는 의미 입니다.

```
Example: G90 G01 X40 Y20
```

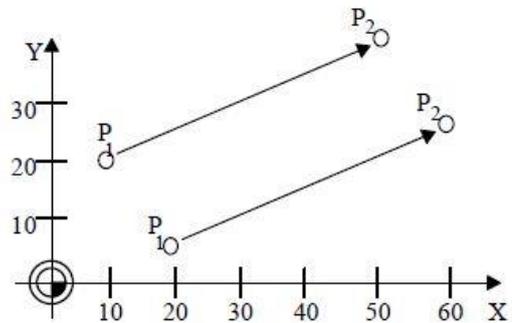


X 축은 절대 위치 40 으로 이동하고 Y 축은 절대 위치 20 으로 이동합니다.

G91: 상대 위치 좌표

좌표의 진입이 상대적입니다. 이것은 주어진 값이 현재 위치를 참고한다는 의미 입니다.

```
Example: G91 G01 X40 Y20
```



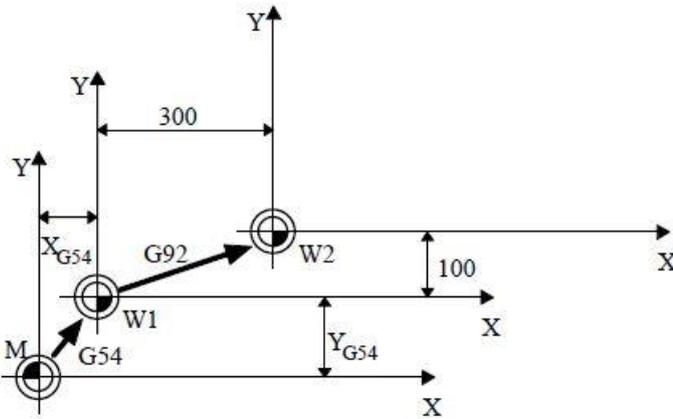
X 축은 현재 위치에서 양의 방향으로 40 units 이동합니다. 그리고 Y 축은 양의 방향으로 20 units 이동합니다.

3.11 G92: 프로그램 된 영점 오프셋

프로그램 된 영점 오프셋은 명시된 모든 축의 영점을 설정합니다. 입력된 좌표는 정의된 영점 오프셋 값에 추가됩니다.

Example: Absolute zero point offset G54 with a programmed "additive" zero point offset (X300 and Y100):

```
G54
G92 X300 Y100
...
```



3.12 G108 / G109 / G110: 가속 / 감속 설정 (Starting with V5.26)

G108 는 CNC Limit 값에 정의된 값의 1~100% 사이에서 가속도를 설정 할 수 있습니다.

G109 는 CNC Limit 값에 정의된 값의 1~100% 사이에서 감속도를 설정 할 수 있습니다.

G110 는 CNC Limit 값에 정의된 값의 1~100% 사이에서 가속도와 감속도를 설정 할 수 있습니다.

PCC: cncsys_dat.limit.a_pos ...Acceleration
 cncsys_dat.limit.a_neg ...Deceleration

Examples: G108 40.5 (Set acceleration to 40.5%)

 G109 80 (Set deceleration to 80%)

 G110 100 (Set acceleration and deceleration to 100%)

함수 G108, G109, G110 은 Modal 입니다. NC 프로그램 시작시, 이 값들은 100%로 초기화됩니다.

3.13 G125: Rounding Edges (Starting with V5.20)

이 함수는 “라운드” 만들기 위해 프로그램 된 점을 허용합니다. CNC 시스템은 사용자에게 의해 정의된 구체 안에 있는 이 점 주위에 스플라인을 계산합니다. 최근에, 스플라인은 직선 사이에 삽입됩니다. 이 함수를 선택한 후에, 프로그래밍 된 점은 함수가 비활성화 될 때 까지 원형입니다. G125는 메인 계층에 영향을 주지 않기 때문에 어떤 위치에서도 경로를 등글게 할 수 있습니다.

CNC 시스템은 정의된 구체에 일치하는 최대 크기를 가진 최적의 스플라인을 계산합니다, 하지만 작아질 수도 있습니다. 스플라인 크기의 다른 제한사항은 원형으로 되는 직선 라인의 길이이고, 스플라인은 각각의 직선 길이의 반보다 클 수 없습니다.

단일 스텝 작동을 위해, 스플라인이 2 개의 같은 크기 섹션으로 구성되어 있다는 것을 유의해야 합니다.

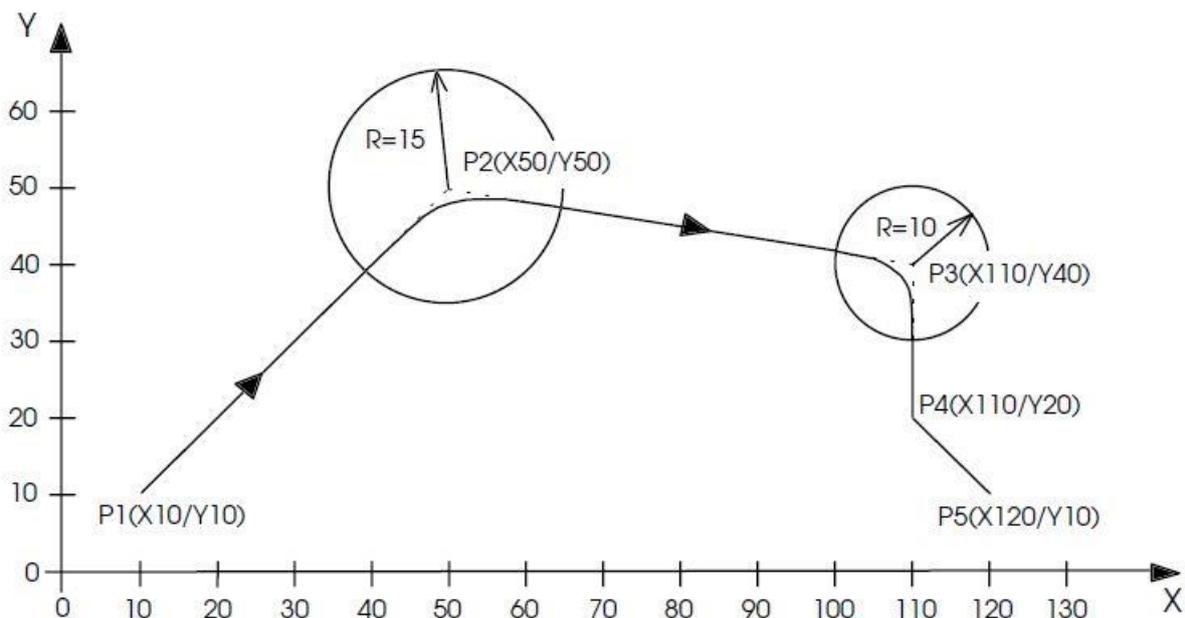
이것은 직선이 스플라인 시작까지 실행되고 나서, 스플라인의 첫번째 섹션과 두번째 섹션까지 이동 한다는 것을 의미합니다.

Selection: G125 Radius

Deselection: G125 0

Example:

```
G90 F20000
G01 X10 Y10
X50 Y50
G125 15
X110 Y40
G125 10
X110 Y20
G125 0
X120 Y10
M30
```



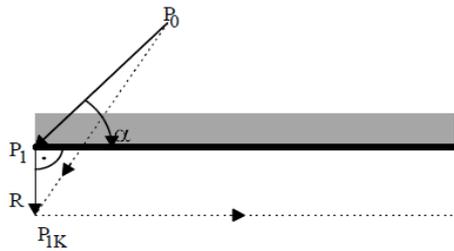
3.14 G138 / G139: CDC Selection and Deselection 의 타입

3.14.1 G138: Direct CDC Selection / Deselection

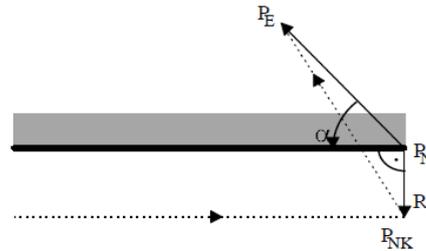
만약 CDC 가 G41, G42 / G40 과 함께 활성화 / 비활성화 되고, G138 은 수정된 윤곽이 직접적으로 입력되거나 종료될지 선택하는데 사용됩니다.

수정된 윤곽으로 향하는 entry 움직임은 entry 각도에 상관없이 첫번째 등거리에 직접적으로 CDC (G41 / G42)가 활성화 된 후 첫번째 record 안에 끝이 납니다. (왼쪽 그림 참조)

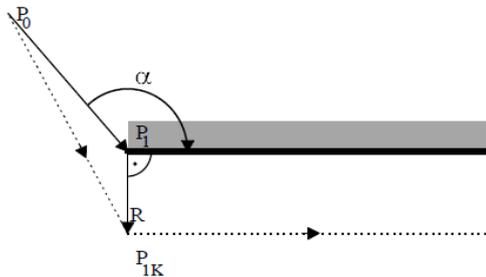
수정된 윤곽에서 나가는 움직임은 나가는 각도에 상관없이 마지막 등거리에서 직접적으로 CDC (G41 / G42)가 비활성화 된 후 첫번째 record 안에 끝이 납니다. (오른쪽 그림 참조)



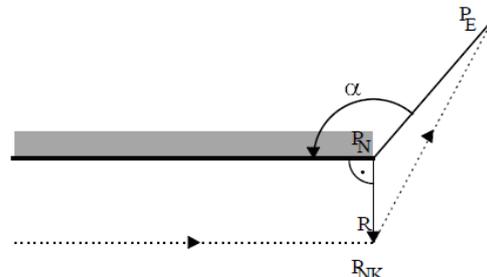
Entry angle $\alpha < 180^\circ$



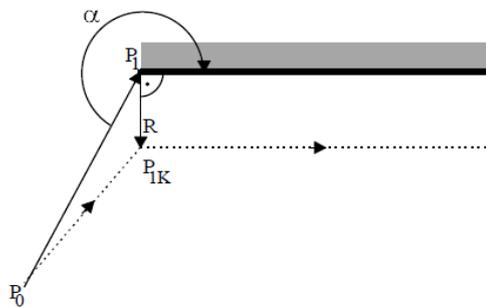
Exit angle $\alpha < 180^\circ$



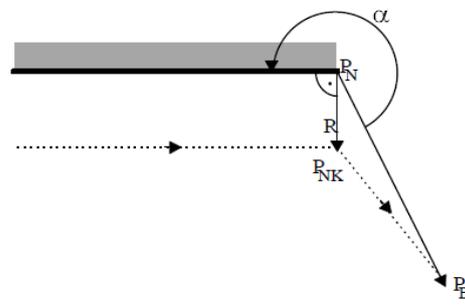
Entry angle $90^\circ \leq \alpha \leq 180^\circ$



Exit angle $90^\circ \leq \alpha \leq 180^\circ$



Entry angle $\alpha > 180^\circ$



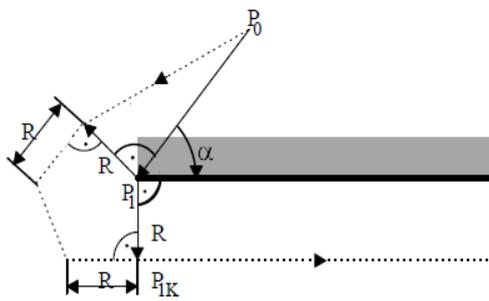
Exit angle $\alpha > 180^\circ$

3.14.2 G139: CDC Selection / Deselection with Linear Transition(s)

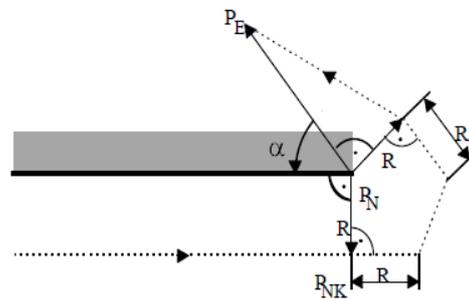
만약 CDC 가 G41, G42 / G40 과 함께 활성화 / 비활성화 되고, G139 는 수정된 윤곽이 Transition 하면서 입력되거나 종료될지 선택하는데 사용됩니다.

수정된 윤곽으로 진입하는 움직임은 진입 각도에 의해 좌우되는 첫번째 등거리 위에서 transition 을 포함하는 CDC (G41 / G42) 활성화 이후에 첫번째 record 가 끝납니다. (왼쪽 그림 참조)

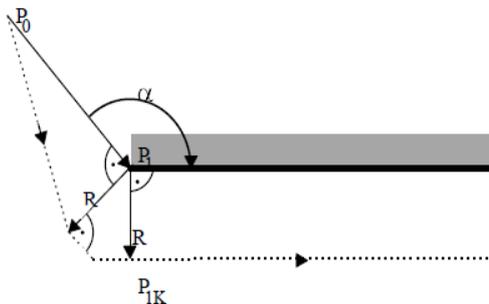
수정된 윤곽에서 나가는 움직임은 나가는 각도에 의해 좌우되는 마지막 등거리 위에서 transition 을 포함하는 CDC (G40) 비활성화 이후에 첫번째 record 에 끝납니다. (오른쪽 그림 참조)



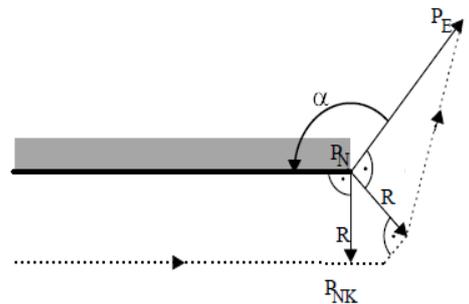
Entry angle $\alpha < 180^\circ$



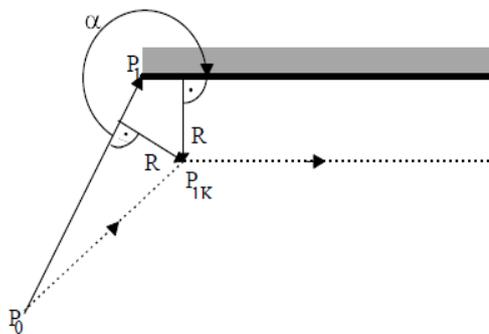
Exit angle $\alpha < 180^\circ$



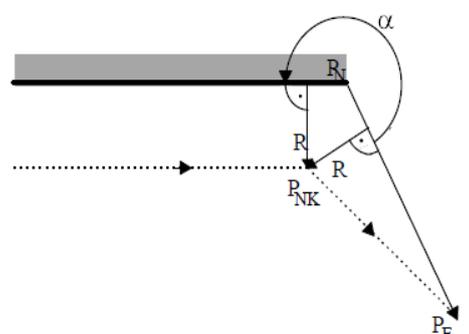
Entry angle $90^\circ \leq \alpha \leq 180^\circ$



Exit angle $90^\circ \leq \alpha \leq 180^\circ$



Entry angle $\alpha > 180^\circ$



Exit angle $\alpha > 180^\circ$

3.15 G159: Expanded Zero Point Offset

이 함수를 가지고, 영점 Record 는 확장된 테이블(인덱스 0 에서 199)에서 선택 할 수 있습니다. Record 1 에서 6 까지는 G-Functions G54 에서 G59 까지 상응합니다. 비활성화 되어 있는 확장된 영점 오프셋은 G53 과 함께 끝냅니다.

Syntax: G159 = *index*

다음은 “*index*”에 입력 될 수 있습니다.

- 모든 숫자 또는 십진수
- 직접적, 간접적 또는 외부 파라미터
- 산술식

Example:

```
G159=151      (Activate zero point offset with record 151)
...
G53          (Activate zero point offset function)
```

```
PCC: zptab.record_cnt = 2
      zptab.startindex = 150

      zptab.record[0].offset[0] = ...      ;Values for record 150
      zptab.record[0].offset[1] = ...
      zptab.record[0].offset[2] = ...

      zptab.record[1].offset[0] = ...      ;Values for record 151
      zptab.record[1].offset[1] = ...
      zptab.record[1].offset[2] = ...
```

3.16 G161 / G162: 원 중심의 절대 / 상대 좌표

원의 중심의 entry(I, J, K)는 절대적 또는 상대적으로 만들어 질 수 있습니다.

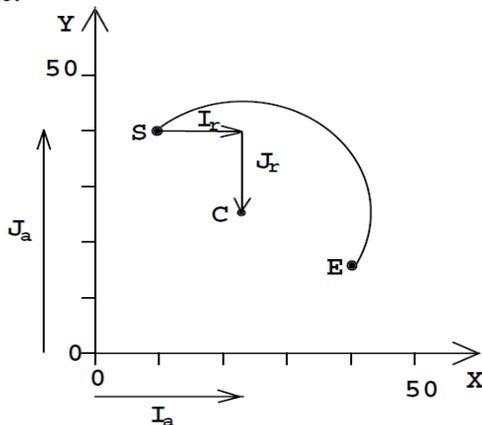
G161: 원 중심의 절대 좌표(Absolute Coordinates for Center of Circle)

보간 파라미터 I, J, K 는 절대적으로 해석됩니다. 이 파라미터 값은 **현재 영점**을 참조합니다.

G162: 원 중심의 상대 좌표(Relative Coordinates for Center of Circle)

보간 파라미터 I, J, K 는 상대적으로 해석됩니다. 이 파라미터 값은 **현재 위치**(원형 움직임의 시작 지점)을 참조합니다.

Example:



- S:** Start position of circular move
Coordinates: X10.0 Y40.0
- E:** End position of circular move
Coordinates: X40.0 Y18.0
- M:** Center position of circular move
Coordinates: X22.8 Y26.0

Absolute (with Ia and Ja):

```
G161 G90
G02 X40 Y18 I22.8 J26.0
```

Relative (with Ir and Jr):

```
G162 G91
G02 X30 Y-22 I12.8 J-14.0
```

3.17 G170: 디코더 동기화

G170 이 NC 프로그램에 포함되어 있다면, 디코더 안에서 프로그램 진행이 이 위치에서 인터럽트 됩니다. 예를 들어 이 방법으로 E 파라미터를 naction(ncsys, ncR_PARAM, ncSET)과 naction(ncsys, ncR_PARAM, ncREDA)으로 설정하거나 읽을 수 있고, R 파라미터를 PCC 에서 CNC 로까지 전달하는 것이 가능합니다. naction(ncsys, ncDECODER, ncSYNC) 포함하는 디코더 안에서 NC 프로그램 처리를 진행 할 때, 새로운 데이터는 즉시 사용됩니다. M-, S-, T- Function 사용은 G170 과 같은 라인에서만 허용됩니다.

3.18 G180: Contour Elements 연결 (Starting with V5.00)

이 명령은 두 개의 Contour elements 를 Transition element 와 연결합니다. 디코더에서 첫 번째 contour element 가 하나의 element 처럼 transition element 와 함께 처리됩니다. 요구되는 G 코드 갯수를 줄이기 위해, 이 명령은 Sub-명령(index 0 – 199, 확장된 영점 오프셋과 마찬가지로)에 확장됩니다.

3.18.1 G180=000: Tangential Line-Circle-Programming

G180=000 으로 아래의 기능을 지원합니다.:

- 2 개의 프로그램 된 원 사이에 탄젠트 자동 삽입
- 시작 위치에서부터 원까지 탄젠트 프로그램 (starting with V5.26)
- 원에서부터 끝나는 위치까지 탄젠트 프로그램 (starting with V5.26)

만약 원 중심의 상대 좌표가 G162 로 선택되었다면, 모든 원의 중심 좌표가 첫 번째 원의 시작 지점(또는 첫 번째 탄젠트의 시작지점)을 참조합니다.

만약 원 중심의 절대 좌표가 G161 로 선택되었다면, 모든 원의 중심 좌표가 현재 영점으로 참조됩니다.

만약 위치의 상대 좌표가 G91 로 선택이 되었다면, 마지막 원의 마지막 위치(또는 마지막 탄젠트의 마지막 위치)의 좌표는 첫 번째 원의 시작점(또는 첫 번째 탄젠트의 시작점) 을 참조합니다.

만약 위치의 상대 좌표가 G91 로 선택이 되었다면, 마지막 원의 마지막 위치(또는 마지막 탄젠트의 마지막 위치)의 좌표는 현재 영점을 참조합니다.

Note: 만약 아래의 상태 중 하나가 포함된다면, 두 원 사이의 탄젠트가 결정되지 않습니다.

1. 하나의 원이 완벽하게 다른 원 안에 있다(완전하게 가려졌다).
2. 원의 반경 중 하나가 0, 또는 모두 0 이다.
3. 원이 겹치고 회전 방향이 반대

만약 아래의 상태가 유효하다면, 시작 지점에서부터 원까지 탄젠트가 결정 될 수 없습니다:

1. 시작 위치가 원 안에 있다.

만약 아래의 상태에 해당한다면, 원에서부터 끝 지점까지 탄젠트가 결정 될 수 없습니다.

1. 끝 지점이 원 안에 있다.

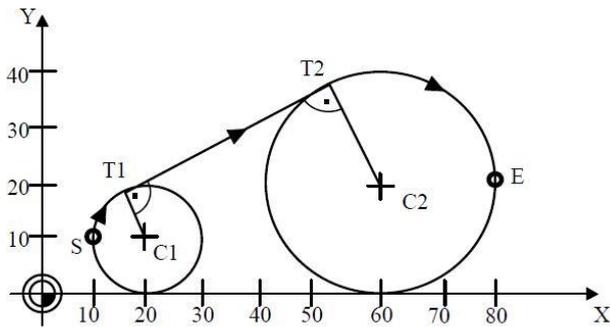
만약 이러한 상태들에 해당한다면, 에러 메시지가 발생하고 NC 프로그램이 정지됩니다.

Example:

G91 G162 (Relative position coordinates, relative coordinates for center of circle)
 G02 I10 J0 G180=000 (1st Circle: defined using center and start position)
 G02 X70 Y10 I50 J10 (2nd Circle: defined using center and end position)

or:

G90 G161 (Absolute position coordinates, absolute coordinates for center of circle)
 G02 I20 J10 G180=000 (1st Circle: defined using center and start position)
 G02 X80 Y20 I60 J20 (2nd Circle: defined using center and end position)

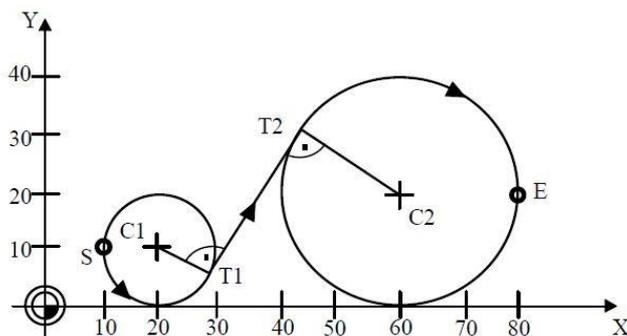


Both circles with the same direction of rotation (G02)

위의 CNC 프로그램, 움직임은 점 S 부터 점 E 까지 정의됩니다. 점 T1 과 T2 는 CNC 시스템에 의해 계산됩니다. CNC 프로그램을 생성 할 때, 점 S, E, C1, C2, 원의 반경이 알려집니다.

Example:

G91 G162 (Relative position coordinates, relative coordinates for center of circle)
 G03 I10 J0 G180=000 (1st Circle: defined using center and start position)
 G02 X70 Y10 I50 J10 (2nd Circle: defined using center and end position)



Circles with different directions of rotation (1st Circle: G03, 2nd Circle: G02)

위의 CNC 프로그램, 움직임은 점 S 부터 점 E 까지 정의됩니다. 점 T1 과 T2 는 CNC 시스템에 의해 계산됩니다. CNC 프로그램을 생성 할 때, 점 S, E, C1, C2, 원의 반경이 알려집니다.

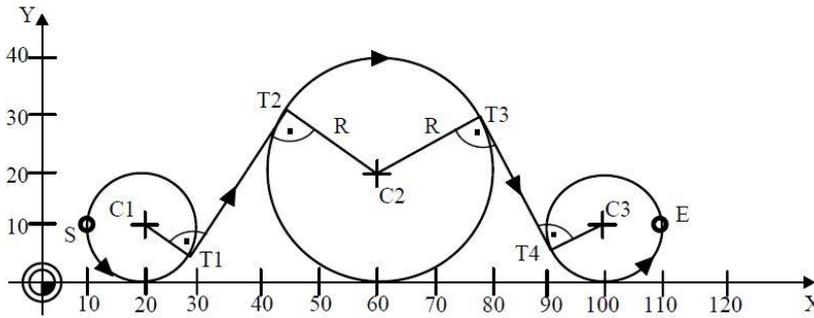
만약 2 개 이상의 원이 각각 다른 원에 연결되고 중심과 반경은 “Middle”안에 원으로 알려집니다. 각각의 원을 정의하기 위해 다음의 기술을 사용합니다:

G02 I10 J10 R=40 반경 entry 를 이용한 원 프로그래밍

G162 를 사용한 중심은 첫 번째 원의 시작 점을 참조합니다.

Example:

G91 G162	(Rel. position coordinates, rel. coordinates for center of circle)
G03 I10 J0 G180=000	(1 st Circle: defined using center and start position)
G02 I50 J10 R=20 G180=000	(2 nd Circle: defined using center position and radius)
G03 I90 J0 X100 Y0	(3 rd Circle: defined using center and end position)



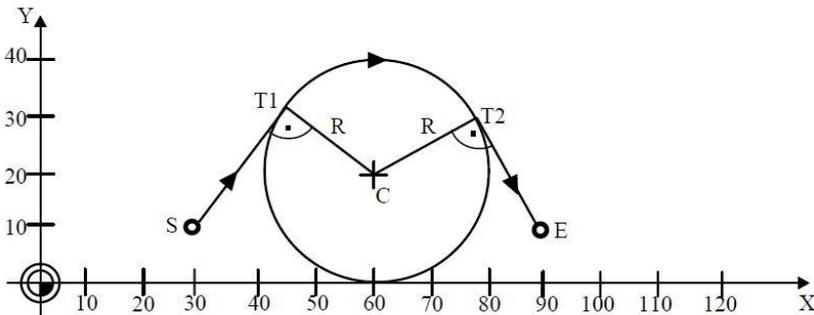
Circles with different directions of rotation (1st Circle: G03, 2nd Circle: G02, 3rd Circle: G03)

위의 CNC 프로그램, 움직임은 점 S 부터 점 E 까지 정의됩니다. 점 T1, T2, T3, T4 는 CNC 시스템에 의해 계산됩니다. CNC 프로그램을 생성 할 때, 점 S, E, C1, C2, C3, 그리고 원의 반경이 알려집니다.

G180=000 은 G01 과 결합됩니다.

Example:

G91 G162	(Rel. position coordinates, rel. coordinates for center of circle)
G01 G180=000	(Tangent to the following circle)
G02 I30 J10 R=20 G180=000	(Circle: defined using center position and radius)
G01 X60 Y0	(Tangent to the specified end position)



Circle (G02) with tangential entry and exit move

위의 CNC 프로그램, 움직임은 점 S 부터 점 E 까지 정의됩니다. 점 T1, T 는 CNC 시스템에 의해 계산됩니다. CNC 프로그램을 생성 할 때, 점 S, E, M 그리고 원의 반경이 알려집니다.

3.19 G200: 축정을 위한 Trigger Event 후 Position Latch (starting with V5.26)

이 함수는 Latch 함수를 활성화 합니다. 이것은 G00 또는 G01 과 함께 사용 될 수 있습니다. 해당 움직임 record(G00 또는 G01)는 트리거 기능을 위한 contour element 를 정의합니다. 만약 특정 엣지 타입이 이 contour element 안에서 트리거 입력이 발생한다면, 위치가 latch 됩니다.

Syntax: G200<Axis><Edge>{<HW-Index>}
 <Axis>: latch 되는 포지션의 CNC 축(i.e. 'X', 'Y', 'Z') 식별자
 <Edge>: latch 를 발생시키는 트리거 엣지의 식별자:
 'P': Positive edge
 'N': Negative edge
 <HW - Index>: HW 트리거 입력 인덱스

Examples: G200 XP 만약 positive 엣지가 X 축의 트리거 입력에서 발생한다면, 그 위치는 축 위에서 latch 됩니다.
 G200 YN 만약 negative 엣지가 Y 축의 트리거 입력에서 발생한다면, 그 위치는 축 위에서 latch 됩니다.
 G200 YN1 만약 negative 엣지가 HW 트리거 입력 1 에서 발생한다면, 그 위치는 Y 축 위에서 latch 됩니다.

Note: G200 은 반듯이 G00 또는 G01 과 함께 사용해야 합니다.

Flag <Axis><Edge>.EV 는 특정 트리거 이벤트가 발생했는지 안했는지에 따라, 테스트를 위한 CNC 프로그램을 실행 할 수 있습니다. 만약 트리거 이벤트가 활성화되면 flag 는 0 으로 설정됩니다. 특정 트리거 이벤트가 트리거 기능에 대해 정의된 contour element 내에서 발생하는 경우, 이 Flag 는 1 로 설정됩니다.

만약 특정 트리거 이벤트가 발생했다면, CNC 축의 위치는 <Axis><Edge>.<Axis>로 부터 읽기가 가능합니다. 트리거 기능이 활성화 되었던 축의 위치만 NC154 HW 모듈에서 래치됩니다. 이 참고 축의 위치를 가지고, 다른 CNC 축들의 위치는 contour element 에 대하여 결정됩니다.

Example:
 G01 X200 Y100 G200 XP (Position latch for the X axis with positive edge at the trigger input of the X axis)
 ...
 \$IF XP.EV == 1 (Positive edge at the trigger input of the X axis has occurred)
 R1 = XP.X + 20 (Latch position for the X axis)
 R2 = XP.Y + 30 (Corresponding position for the Y axis)
 \$ELSE
 ...
 \$ENDIF

Note: 만약 이 트리거 Flag 가 IF branch 안에서 테스트 되었다면, 디코더는 특정 트리거 이벤트가 발생하거나 해당 contour element 가 완료 될 때까지 멈추게 됩니다. IF 블록은 이러한 상태들 중 하나가 충족 될 때까지 작동하지 않습니다.

Example:

```
G91 G162 (Relative position coordinates, relative coordinates for center of circle)
G01 X70 Y0 G200 XP (Linear interpolation from P1 to P2 with activation of the trigger function)
G03 X12 Y12 J12 (Circular interpolation from P2 to P3)
G01 X0 Y16 (Linear interpolation from P3 to P4)
```



G200 with G01

위의 예제에서 트리거 기능을 위한 contour element 는 P1 부터 P2 까지의 직선입니다.

Example:

```
G91 G162 (Relative position coordinates, relative coordinates for center of circle)
G01 X0 Y16 (Linear interpolation from P1 to P2)
G03 I12 G180=0 (Circle: defined using center and start position)
G01 X70 Y0 G200 XP (Tangent from T1 to P3 with activation of the trigger function)
```



G200 with G01 and G180=0 for tangential exit move from a circle

위의 예제에서, 트리거 기능을 위한 contour element 는 T1 부터 P3 까지의 직선입니다. T1 은 원 선에 접선을 형성한 위치입니다.

Note:

다음의 예제에서, 디코더가 첫 번째 contour element 를 처리하기 위해, 두 번째 contour element 를 알아야 합니다. 왜냐하면 그것들 사이에 tangential transition 때문입니다.

만약 이번 예제에서 트리거 Flag 의 상태가 트리거를 포함하는 CNC record 활성화 이후에 즉시 체크된다면, contour element 는 디코더로 사용 할 수 없게 되고 CNC 프로그램이 막힙니다.

Example:

```
G91 G162 (Rel. position coordinates, rel. coordinates for center of circle)
G01 X24 Y34 G25 16 (Linear interpolation from P1 to P2)
G01 X90 Y0 G25 12 G200 XP (Linear interpolation from P2 to P3 with trigger function)
G01 X0 Y-28 G25 0 (Linear interpolation from P3 to P4)
```

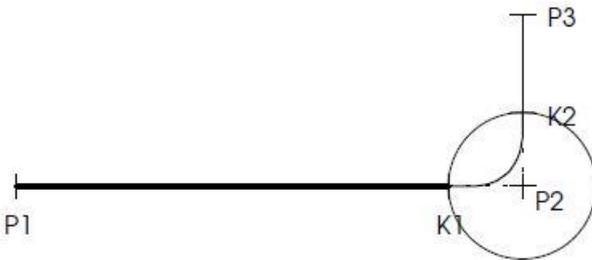


G200 with G01 and G25

이번 예제에서 트리거 기능을 위한 contour element 는 T2 부터 T3 까지 직선입니다. 예를들어, G25 가 사용 중이라면, contour element 는 transition arcs 때문에 짧아집니다. “T 포지션”과 프로그래밍 된 위치의 간격은 정의된 반경보다 짧거나 같거나 또는 더 커질 수 있습니다. 이것은 연속된 직선 사이의 각도에 따라 달라집니다.

Example:

```
G91 G162 (Relative position coordinates, relative coordinates for center of circle)
G01 X70 Y0 G200 XP (Linear interpolation from P1 to P2 with activation of the trigger function)
G125 12 (Rounding of edge with a spline)
G01 X0 Y16 (Linear interpolation from P2 to P3)
G125 0 (Deactivate rounding)
```



G200 with G01 and G125

이번 예제에서는 트리거 기능을 위한 contour element 는 P1 에서 K1 까지 직선입니다. 예를 들어, G125 가 사용 중이라면, contour element 는 구형 반지름에 의해 짧아질 수 있습니다.

Example:

```
G91 G162 (Rel. position coordinates, rel. coordinates for center of circle)
G01 G180=0 G200 XP (Tangent from P1 to the circle with activation of the trigger function)
G03 I70 J12 R=12 G180=0 (Circle: defined using center and start position)
G01 X0 Y16 (Tangent from T2 to P2)
```



G200 with G01 and G180=000 for tangential entry move to a circle

이번 예제에서 트리거 contour element 는 P1 부터 T1 까지의 직선에 해당합니다. T1 은 원의 접선을 형성하는 점입니다.

3.20 G201: 트리거 이벤트 이후 포지션 래치 그리고 중단 (starting with V5.26)

이 함수는 정지가 되어있는 래치 함수를 활성화 합니다. 이것은 G00 또는 G01 과 함께 사용 할 수 있습니다. 해당 동작 record (G00 또는 G01)는 트리거 기능에 대한 contour element 를 정의합니다. 만약 정해진 엣지 타입이 이 contour element 내부 트리거 입력에서 발생한다면, 위치는 래치됩니다. 게다가 CNC-움직임이 멈출 때까지 동작이 느려집니다. 그리고 정지 위치가 현재 contour element 그리고 다음 동작 record 의 시작 위치로 설정됩니다.

Syntax: G201<Axis><Edge>{<HW-Index>}
 <Axis>: 각각의 위치 위의 CNC 축 (i.e. 'X', 'Y', 'Z')의 식별자가 latch 됩니다.
 <Edge>: 그 지점의 latch 를 발생시키는 트리거 엣지의 식별자:
 'P': Positive edge
 'N': Negative edge
 <HW - Index>: HW 트리거 입력의 인덱스

Examples: G201 XP 만약 positive 엣지가 X 축의 트리거 입력에서 발생한다면, 그 위치는 축 위에서 latch 가 되고 움직임이 멈춥니다.
 G201 YN 만약 negative 엣지가 Y 축의 트리거 입력에서 발생한다면, 그 위치는 축 위에서 latch 되고 움직임이 멈춥니다.
 G201 YN1 만약 negative 엣지가 HW 트리거 입력 1 에서 발생한다면, 그 위치는 Y 축 위에서 latch 되고 움직임이 멈춥니다.

Note: G201 은 반듯이 G00 또는 G01 과 함께 사용되어야 합니다.

Flag <Axis><Edge>.EV 는 특정 트리거 이벤트 발생 유무에 따라, 테스트를 위한 CNC 프로그램을 실행 할 수 있습니다. 트리거 기능이 활성화 되면, Flag 는 0 으로 설정됩니다. 만약 트리거 기능에 대해 정의된 contour element 내에서 특정 트리거 이벤트가 발생한다면, Flag 는 1 로 설정됩니다.

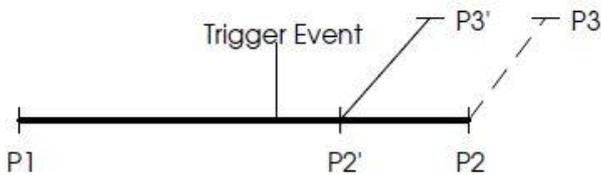
특정 트리거 이벤트가 발생했다면, CNC 축의 위치는 <Axis><Edge>.<Axis>로 부터 읽는 것이 가능합니다. 트리거 기능이 활성화 되었던 축의 위치만 NC154 HW 모듈에서 래치됩니다. 이 참고 축 위치를 가지고, 다른 CNC 축들의 위치는 contour element 에 대하여 결정됩니다.

Example:

```
G01 X200 Y100 G201 XP (Position latch for the X axis with positive edge at the trigger input of the X axis)
...
$IF XP.EV == 1 (Positive edge at the trigger input of the X axis occurred)
  R1 = XP.X + 20 (Latch position for the X axis)
  R2 = XP.Y + 30 (Corresponding position for the Y axis)
$ELSE
...
$ENDIF
```

Example:

```
G91 (Relative position coordinates)
G01 X70 Y0 G201 XP (Linear interpolation from P1 to P2 with activation of the trigger function)
G01 X0 Y16 (Linear interpolation from P2 to P3)
```

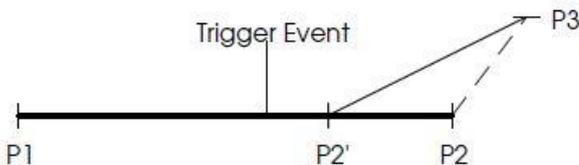


G201 with G01 (relative position coordinates)

이번 예제에서 트리거 기능을 위한 contour element 는 P1 부터 P2 까지의 직선입니다. 한번 트리거 이벤트가 발생하면, 정지 할 때까지 축은 느려집니다. 정지 위치 P2'는 다음 동작 record 에 대한 시작 위치로 설정됩니다. 상대 위치 좌표를 위해 이 직선의 끝나는 위치는 P3 부터 P3'까지 변경합니다.

Example:

```
G90 (Absolute position coordinates)
G01 X70 Y0 G201 XP (Linear interpolation from P1 to P2 with activation of the trigger function)
G01 X82 Y16 (Linear interpolation from P2 to P3)
```

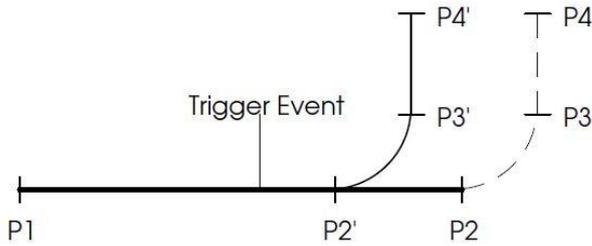


G201 with G01 (absolute position coordinates)

이번 예제에서 프로그래밍 된 contour 는 이전의 예제와 유사합니다. 상대 위치 좌표와 절대 위치 좌표만 다르고 트리거 contour 를 따르는 직선 위에 끝 점 P3 의 위치는 변경되지 않습니다.

Example:

```
G91 G162 (Relative position coordinates, relative coordinates for center of circle)
G01 X70 Y0 G201 XP (Linear interpolation from P1 to P2 with activation of the trigger function)
G03 X12 Y12 J12 (Circular interpolation from P2 to P3)
G01 X0 Y16 (Linear interpolation from P3 to P4)
```



G201 with G02 (relative position coordinates)

만약 위치 좌표와 가운데 점의 좌표 모두 상대적으로 결정되어 있다면, 원형의 interpolation 을 사용하는 record 는 G201 을 사용하여 직선을 따라갈 수 있습니다. 절대적 프로그래밍을 포함한, 원 시작 점의 변화는 NC 프로그램 안에서 contour 에러를 발생시킬 수 있습니다. 트리거 이벤트가 발생했을 때, 움직임이 정지를 하게 됩니다. 정지 위치 P2'는 다음 record 의 시작 위치로 사용됩니다. P3 와 P4 를 대신해서 P3'와 P4'가 사용됩니다.

Note: 만약 위치 좌표 또는 중간 점 좌표가 절대적으로 명시되어 있다면, 원형 보간법을 사용한 record 가 G201 을 포함하고 있는 record 는 실행 할 수 없습니다.

G201 은 다음 contour element(e.g. G25, G125, G180)으로 전환되어 결정하는 동작 record 를 위해 사용되어야 합니다. 현재 contour element 의 끝 점이 알려질 때까지 다음 contour element 로 전환이 결정되지 않기 때문에, 트리거 기능에 대한 contour element 는 처리되지 않습니다.

4 M - Function

M-Function 은 CNC 시스템과 PCC 사이 통신을 관리합니다. 만약 NC 프로그램이 처리되는 동안 M-Function 이 도달하면, CNC 시스템 (cncsys_dat.cnc_plc.data.m_funcnt[i])의 사용자 데이터 안에 각각의 BYTE 는 1 로 설정됩니다. M-Function 은 항상 프로그래밍된 NC record 의 시작단계에서 유효하며, 단지 예외는 record 끝에서 유효한 M00 과 M01 입니다.

Syntax: Mindex

다음과 같이 0 에서 127 까지 Integer 값은 “rpm”을 위해 입력될 수 있습니다:

- 모든 숫자
- 직접, 간접적 또는 외부 파라미터
- 연산식

Example: CNC: G90 G01 X100 Y100 F1000 M40

PCC: X와 Y 축이 움직이기 전에, M – Function 은 PCC 로 전달됩니다.
그 다음에 함수 nccnccom (cncsys)은 다음의 원인을 가지고 옵니다.
cncsys_dat.cnc_plc.data.m_funcnt[40] = 1

두 영역으로 나뉜 CNC 시스템에서 128 까지 M – Function 이 가능합니다.

- 동기화된 M – Function
- 비동기화된 M – Function

4.1 동기화된 M – Function

만약 CNC 시스템이 동기화된 M-Function 에 도달했다면 (e.g. M50), CNC 시스템 (cncsys_dat.cnc_plc.data.m_funcnt[50]) 의 사용자 데이터 각각의 BYTE 가 1 로 설정되고 BYTE 가 PCC 에 의해 다시 0 으로 설정될 때 까지 NC 프로그램의 처리가 중단됩니다.

4.2 비동기화된 M – Function

만약 CNC 시스템이 비동기화된 M-Function 에 도달했다면 (e.g. M40), CNC 시스템 (cncsys_dat.cnc_plc.data.m_funcnt[40]) 의 사용자 데이터 각각의 BYTE 가 1 로 설정되고, NC 프로그램의 처리가 중단됩니다.

Note: 만약 같은 비동기화된 M-Function 이 호출하고 있는 nccnccom(cncsys)에 의해 PCC 에 전달되는 것 없이 NC 프로그램에 의해 세 번 설정 된다면, 여러 메시지가 CNC 시스템의 사용자 데이터 안에 입력되고 NC 프로그램이 중단됩니다.

PCC: 동기화 및 비동기화된 M-Function 영역에 대한 파라미터는 다음과 같이 설정됩니다.

```
cncsys_dat.cnc_plc.parameter.first_M      = ... Index of first M-Function without Synchr.
cncsys_dat.cnc_plc.parameter.last_M       = ... Index of last M-Function without Synchr.
cncsys_dat.cnc_plc.parameter.first_M_S    = ... Index of first M-Function with Synchr.
cncsys_dat.cnc_plc.parameter.last_M_S     = ... Index of last M-Function with Synchr.
```

이러한 영역의 파라미터는 naction(cncsys,ncCNC_PLC,ncINIT)을 포함한 초기값을 설정합니다.

동기화 및 비 동기화 된 M-Function 을 위한 영역은 겹치는 것을 금지합니다. 이러한 영역을 위해 미리 결정한 셋팅은:

M40 ... M49 비동기화된 M-Function
M50 ... M59 동기화된 M-Function

4.3 미리 설정된 M-Function

아래에 설명된 M-Function 은 B&R CNC 소프트웨어에 미리 정의되어 있습니다.

Note:

CNC 시스템 데이터(ncnsys_dat.cnc_plc.data.m_func) 내부에 미리 정의된 M-Functions BYTE 는 M-Function 이 정의된 영역 (cncsys_dat.cnc_plc.parameter.first_M ... last_M_S)에 있을 때 설정됩니다. 이 경우에, PCC 프로그램은 이러한 M-Function 에 의존하여 추가적인 기능을 실행 할 수 있다.

4.3.1 M00: Programmed Stop

M00 명령은 NC 프로그램의 작동을 정지시킵니다. 만약 PCC 가 naction(ncnsys, ncMOVE, ncCONTINUE) 명령을 준다면, 작동이 계속됩니다. M00 은 NC record 끝에서 유효합니다.

4.3.2 M01: Optional Stop

만약 조작 모드 “Stop at M1”이 켜져있다면, M01 명령은 M00 과 비슷하게 작동합니다.

PCC: naction(ncnsys, ncM1_STOP, ncSW_ON) 조작모드“Stop at M1” 켜짐
 naction(ncnsys, ncM1_STOP, ncSW_OFF) 조작모드“Stop at M1” 꺼짐

4.3.3 M02 / M29 / M30: 프로그램 / 서브 프로그램 종료

메인 프로그램에서 영향

명령어 M02, M29, M30 은 메인 프로그램을 종료시킵니다.

서브 프로그램에서 영향

명령어 M02, M29, M30 은 메인 프로그램으로 돌려줍니다. 이것은 서브 프로그램 작동이 끝나고 서브 프로그램 호출 뒤에 실행되는 NC record 에서 작동이 계속된다는 것을 의미합니다.

5 Technology Parameters

5.1 Feed

Feed 는 궤적 위에 동작 속도를 결정하고 **units/min** 의 단위로 프로그램 됩니다. 만약 너무 빠른 Feed 가 입력이 되면, 자동적으로 최대 궤적 속도로 제한됩니다(cncsys_dat.limit.v_trajectory).

Syntax: Ffeed

“feed”에는 아래의 값 사이의 양수가 입력이 될 수 있습니다:

- 모든 숫자 또는 10 진수
- 직접, 간접적 또는 외부 파라미터
- 연산식

Example: F1300 FR1 FR1*R2

0 과 $(2^{31}-1)$ /디코더 해상도) 사이 값이 “feed”에 입력 될 수 있습니다. 또한 그 숫자는 소수점을 포함 할 수 있습니다.

Note for CNC SW until V5.26:

이 버전에서 CNC 디코더는 내부적으로 INT32 형식을 가지고 계산합니다. 이를 위해 feed 에 디코더 해상도(cncsys_dat.decoder.parameter.resolution)를 곱합니다. 그래서 0 과 $(2^{31}-1)$ /디코더 해상도) 사이 값이 “feed”에 입력됩니다.

5.2 Spindle RPM (S-Function)

NC 프로그램에서 2 개의 스핀들(S0 과 S1)을 조작하는 것이 가능합니다. 입력된 값은 PCC 로 전달됩니다.

Syntax: Srpm 스핀들 1 에 대한 RPM
 S0=rpm 스핀들 1 에 대한 RPM
 S1=rpm 스핀들 2 에 대한 RPM

“rpm”에 0 에서 $(2^{31}- 1)$ 까지의 Integer 값이 입력 가능합니다.

- 모든 숫자
- 직접적, 간접적 또는 외부 파라미터
- 연산식

Example: CNC: S300 S1 = 4000

PCC: 그 다음 함수 nccnccom (cycsys)는 다음의 원인을 가지고 옵니다.
 cncsys_dat.cnc_plc.data.s_func [0] = 300
 cncsys_dat.cnc_plc.data.s_func [1] = 4000

5.3 툴 번호 (T-Function)

툴 번호는 처리과정에서 어떤 단계에 요구되는 툴을 결정합니다. T-Function 은 자동 툴 교환을 위해 사용할 수 있습니다. 원하는 툴은 번호와 함께 정의됩니다.

Syntax: Tnumber

0 에서 255 까지 Integer 값이 “number”에 입력 될 수 있습니다.

- 모든 숫자
- 직접적, 간접적 또는 외부의 파라미터
- 연산식

Beispiel: CNN: T31

PCC: 그 다음 함수 nccnccom (cycsys)는 다음의 원인을 가지고 옵니다.
 cncsys_dat.cnc_plc.data.t_funcnt [0] = 31

5.4 툴 데이터 번호

툴을 위해 수정된 변수(반지름, 길이, 오프셋)는 툴 데이터 record 에서 정의됩니다. 툴 데이터 번호를 포함하여 어떤 툴 데이터 record 의 수정된 변수는 선택 될 수 있습니다. 축 오프셋의 툴 길이 수정과 hte 수정이 활성화되고 cutter diameter 는 cutter diameter 수정을 준비합니다.

Syntax: Dnumber 툴 데이터 record “number” 의 수정된 변수 선택
 D0 툴 데이터 비활성화

다음과 같이 0 에서 255 까지 Integer 변수가 “number”에 입력 될 수 있습니다.

- 모든 숫자
- 직접적, 간접적 또는 외부의 파라미터
- 연산식

RPS: 툴 데이터 record 는 **ncta_tptab** 데이터 타입을 포함한 데이터 테이블 안에서 정의 될 수 있다.

```

totab.record_cnt          = 2
totab.startindex         = 50

totab.record[0].radius    = ...      Values for record 50
totab.record[0].length    = ...
totab.record[0].offset [0] = ...
totab.record[0].offset [1] = ...
totab.record[0].offset [2] = ...

totab.record[1].radius    = ...      Values for record 51
totab.record[1].length    = ...
totab.record[1].offset [0] = ...
totab.record[1].offset [1] = ...
totab.record[1].offset [2] = ...
    
```

Cutter diameter 수정을 포함한 NC 프로그램이 시작되기 전에, ncTOOL 타입을 포함한 NC 데이터 모듈이 반듯이 데이터 테이블로 생성되어야 하고, ncaction

(cncsys, ncDAT_MOD, ncDOWNLOAD)을 포함하여 NC 데이터 모듈로 전송되어야 하며, naction(cncsys, ncTOOL_TAB, ncSET)을 포함하여 활성화 되어 합니다.

축 오프셋이 X, Y, Z 에 입력 됩니다. 길이 설정은 Working Plane 오른쪽 부분에 있는 Tool Axis 안에서 작동합니다.

Working Plane	G-Function	Tool Axis
XY	G17	Z
ZX	G18	Y
YZ	G19	X

Note:
만약 각각의 축이 프로그램 되어 있다면, 수정 obly 가 활성화 됩니다.

Example: G17 D10 (Activate tool data of record 10)
 G01 Z20 (Set tool length and axis offset)
 G01 X200 G41 (Activate CDC left)
 ...
 G40 D0 (Deactivate CDC, Deactivate tool data)

6 좌표 프로그래밍

NC 프로그램에서 좌표는 CNC units 형태로 입력됩니다. 각각의 축에 대한 좌표는 해당 위치 units 으로 변환됩니다 (위치 명령을 받은 축은 모든 units 에서 동작 할 수 있습니다). 이 변환을 위해 어떤 factor 가 각각의 축에 정의됩니다(cncsys_dat.axis [i].unitfactor).

Example: Positioning unit: 0.01 mm
 CNC unit: 1.00 mm
 ⇒ cncsys_dat.axis [i].unitfactor = 1.00 / 0.01 = 100.0

6.1 위치 좌표

축은 주소 값(e.g. 'X')을 통해 접속이 됩니다. 각각의 축에 대한 주소 값은 CNC 시스템의 사용자 데이터 (cncsys_dat.axis [i].name)에서 정의 됩니다. 좌표가 절대적 또는 상대적으로 디코더 파라미터 (cncsys_dat.decoder.parameter.co_ords)에서 만들어 지거나 G90 / G91 을 포함하여 만들어진 경우 설정을 결정합니다.

Syntax: addressletter coordinate

다음은 "coordinate"에 입력이 될 수 있습니다.

- 모든 숫자 또는 소수
- 직접적, 간접적 또는 외부 파라미터
- 연산식

Example: X1300 Y - 12.5 Z+ .25
 XR1 ZP150 XR1*R2

Note for CNC SW until V5.26:

이 버전에서 CNC 디코더는 내부적으로 INT32 형식으로 계산합니다. 이 때문에 모든 좌표는 디코더 해상도 (cncsys_dat.decoder.parameter.resolution)가 곱해집니다. 이것이 축의 최대 동작 범위가 ((-2³¹) / 디코더 해상도)와 ((2³¹) / 디코더 해상도) 사이인 이유입니다. 더 큰 디코더 해상도가 정의 되고 더 크거나 더 작은 것이 최대 동작 범위 입니다.

Example: Decoder resolution: 10000
 INT32 range: -2147483648 ... 2147483647
 CNC unit: 1.00 mm

 Movement range: -2147483648 mm / 10000 ... 2147483647 mm / 10000
 -214748 mm ... 214748 mm
 -214.748 m ... 214.748 m

6.2 원 중심에 대한 좌표

X, Y, Z 축에서 원 중심은 좌표 I, J, K (interpolation 파라미터)에 의해 결정됩니다. 원 중심에 대한 좌표는 절대값 (work piece 영점 기준) 또는 원의 시작점이 기준인 상대값으로 입력됩니다. 이것에 대한 기본

설정은 디코더 파라미터 (cncsys_dat.decoder.parameter.center)에서 정의됩니다. NC 프로그램에서 이것은 G161 또는 G162 로 변경 될 수 있습니다. Interpolation 파라미터는 하나의 record 에서만 유효합니다. 만약 파라미터가 설정되지 않았다면 0 이 사용됩니다.

Syntax: I / J / K coordinate X, Y, 또는 Z 축에 대한 interpolation 파라미터

위치 좌표에서 사용되는 같은 entry 는 “coordinate”에 주어질 수 있습니다.

7 파라미터

파라미터는 어떤 값이 할당될 수 있는 변수들과 비교 될 수 있습니다.

7.1 Direct 파라미터 (R 파라미터)

Syntax: $R_{index} = value$

“index”에 모든 숫자 0 부터 999 까지 입력 될 수 있습니다.

다음은 “value”에 입력 될 수 있습니다.

- 모든 숫자 또는 소수
- 직접, 간접 또는 외부 파라미터
- 연산식

Example: $R1=R2$ 파라미터 R1 은 R2 부터 값을 할당 받습니다.
 $R3=R1 / 100$ R3 은 R1 을 100 나눈 값을 할당 받습니다.

7.2 Indirect 파라미터 (P 파라미터)

Syntax: $R_{index} = value$

“index”에 모든 숫자 0 부터 999 까지 입력 될 수 있습니다.

다음은 “value”에 입력 될 수 있습니다.

- 모든 숫자 또는 소수
- 직접, 간접 또는 외부 파라미터
- 연산식

Pindex 를 가진 R-파라미터가 접속되고, index 는 Rindex 값으로 저장됩니다(간접적 주소, R-파라미터 Rindex “point”).

Example: $R150 = 100$ R150 에 R100 의 인덱스를 저장합니다.
 $P150 = 203.5$ R100 은 P150 을 통해 203.5 값이 할당됩니다.

Note: PCC 로 부터, 몇몇 R 파라미터 데이터 모듈이 생성 될 수 있고 NC 모듈에 전달 될 수 있습니다. `ncaction(ncsys, ncR_PARTAB, ncSET)`을 포함하는, 이러한 R 파라미터는 CNC 시스템 리스트로 전달 될 수 있습니다. 그것들은 다음 프로그램 시작 후에 사용됩니다. 개별적으로 R 파라미터 설정은 즉시 유효합니다.

7.3 외부 파라미터 (starting with V5.10)

외부 파라미터는 NC 모듈의 듀얼 포트 RAM(DPR)에 위치합니다. 이것이 이 파라미터들이 PCC 와 CNC 에 의해 직접적으로 접속 할 수 있는 이유입니다. 만약 이 파라미터 중 하나가 새로운 변수에 할당된다면, 이것은 즉시 PCC 와 CNC 에서 유효합니다.

	CNC		RPS
Syntax:	<i>EXBindex=value</i>	Byte parameter (datatype INT8)	<i>ex_param.EXB [index] =value</i>
	<i>EXWindex=value</i>	Word parameter (data type INT16)	<i>ex_param.EXW [index] =value</i>
	<i>EXLindex=value</i>	Long parameter (datatype INT32)	<i>ex_param.EXL [index] =value</i>
	<i>EXFindex=value</i>	Float parameter (data type FLOAT)	<i>ex_param.EXF [index] =value</i>

“index”에는 0 에서 99 까지 모든 숫자가 입력 될 수 있습니다.

“value”에는 숫자, 연산식 표현, 직접적 또는 간접적 파라미터가 입력 될 수 있습니다(파라미터의 데이터 타입에 해당)

PCC 에 의해 만약 다이나믹 변수 *ex_param* (데이터 타입 *ncXXXexpar*)가 아래의 명령을 가진 해당 데이터 범위에 연결이 되어 있다면 외부 파라미터에 접근이 가능합니다.

```
ex_param access cncsys_dat.cnc_plc.ex_param.access_adr
```

Note:

PCC 와 CNC 의 동시 접속이 포함된 데이터 일관성은 2 바이트(EXL 과 EXF)보다 더 큰 EX 파라미터를 보장 할 수 없습니다.

다음의 상태에서 CNC 가 갑자기 외부 파라미터에 접근이 되지 않습니다.

```
.move.status.cnt_ncprog = 0
.decoder.status.halt     = ncTRUE
.decoder.status.sync     = ncTRUE
```

CNC 프로그램 활성화 상태에서 PCC 로부터 외부 파라미터가 변경이 되야한다면 , CNC 프로그램을 포함한 이러한 변경을 동기화 하기 위해서 CNC 프로그램에 G170 명령을 사용하는 것을 추천합니다. 만약 이게 불가능 하다면, 적어도 *ncaction(cncsys, ncEX_PARM, cnSW_ON)*과 *ncaction(cncsys, ncEX_PARM, ncSW_OFF)*이 외부 파라미터와 연결을 위한 락킹(locking)을 on/off 를 스위칭하기 위해서 호출해야 합니다.

8 Comments

프로그램 색션은 쿠멘트(주석)으로 설명 할 수 있습니다. 코멘트는 CNC 컨트롤러에 의해 걸러집니다. “(...)” 사이의 모든 문자는 무시됩니다.

Syntax: (comment)

9 수학적연산자

수학적연산자는 (“[]”) 기호를 사용해서 중첩 될 수 있습니다 (최대 중첩의 깊이 = 10). 중첩된 표현에 대한 hierarchical 순서는 표준 연산자 표현에 부합됩니다(계산을 위한 수학적 우선순위)

Example: SIN [R1 * [R2 + R3]]

9.1 기본 연산자

Operation	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
**	Power (x to the y power)
MOD	Modulo Function

9.2 카운팅 기능

Operation	Description
ABS []	Absolute value
SQR []	Squared value
SQRT []	Square root value
EXP []	Exponential function to base e
LN []	Logarithm to base e
DEXP []	Exponential function to base 10
LOG []	Logarithm to base 10

9.3 기하학적 기능

기하학적 기능을 위한 “degree”는 unit 으로 사용됩니다.

Operation	Description
SIN []	Sine of a value
COS []	Cosine of a value
TAN []	Tangent of a value
ASIN []	Arcsine of a value
ACOS []	Arccosine of a value
ATAN []	Arctangent of a value

9.4 컨버전 기능

Operation	Description
INT []	Whole number part of a value
FRACT []	Part of a value after the decimal point
ROUND []	Rounded value (whole number)

9.5 현재 Cutter Radius 구하기

Operation	Description
Rad	Current set cutter radius {without parameter}

10 Control Blocks

컨트롤 블록 생성은 “C” 프로그래밍 언어와 유사합니다. 각 컨트롤 블록은 요구된 표현에 따라 실행 될 수 있도록 합니다. Record 숫자 그리고/또는 블록을 스킵하기 위한 문자 ‘/’는 컨트롤 블록 이전에 입력되도록 실행됩니다. 만약 다른 entry 들이 만들어 지면, 다양한 에러가 발생합니다.

Note:

컨트롤 블록에 대한 최대 중첩 깊이는 20 입니다.

10.1 조건

조건 결정은 컨트롤 블록으로 만들어 질 수 있습니다. 이것은 어떤 명령 record 가 파라미터 또는 연산식 표현을 의존하여 발생합니다.

하나의 표현 또는 적어도 두 개의 표현으로 구성된 조건이 비교 연산자를 통해 연결됩니다.

Compare Operator	Description
<	less than
<=	less than or equal
==	equal
!=	not equal
>=	greater than or equal
>	greater than

만약 두개 또는 그 이상의 표현이 비교 연산자와 연결 되었다면, 2 개의 결과만 가능합니다 (0 과 1):

- 1 ... 비교는 참 (e.g. 0<5; 20 == 20; 30>15)
- 0 ... 비교는 거짓 (e.g. 0>5; 20 != 20; 30<15)

또한 조건은 데이지 체인으로 가능합니다. 만약 모든 조건이 만족된다면, 데이지 체인 조건은 만족합니다. 만약 하나의 표현으로 구성되어 있다면, 이 표현의 값은 거절됩니다:

Value	Rating
< 0.5	0
>= 0.5	1

10.1.1 IF 문

IF – ELSE 문은 조건을 만드는데 사용됩니다. ELSE 섹션은 옵션으로 필수 요소는 아닙니다. 다음 Syntax 가 사용됩니다:

```
Syntax:    $IF Condition
           Block1
           $ELSE
           Block2
           $ENDIF
```

IF 문은 조건을 확인합니다. 만약 조건을 만족하면, Block1 이 실행됩니다. 조건을 만족하지 않고, ELSE 섹션이 존재한다면 Block 2 가 실행됩니다. IF 섹션 또는 ELSE 섹션은 반듯이 ENDIF 로 마무리 되어야합니다.

10.1.2 ELSEIF 체인

IF-/ELSEIF 시퀀스 명령이 선택의 다양성을 이용한 의사 결정을 위한 가장 보편적인 방법입니다. 다음의 시퀀스에서, 조건이 가장 첫번째 순서로 평가됩니다. 이 조건들 중 하나가 유효해 지자마자, 각각의 블록이 실행되고 모든 체인이 끝납니다. 프로그램이 ENDIF 명령 이후에 계속됩니다.

```
Syntax:    $IF Condition 1
           Block 1
           $ELSEIF Condition 2
           Block 2
           $ELSEIF Condition 3
           Block 3
           :
           :
           $ELSE
           Block n
           $ENDIF
```

만약 조건이 하나도 유효하지 않다면, ELSE 섹션이 사용됩니다. 만약 특정 동작이 각 케이스에서 발생하지 않는다면, ELSE 섹션을 그냥 나오거나 이 섹션이 에러 처럼 “impossible” 조건을 나타내도록 사용 할 수 있습니다.

```
Example:    $IF R1==2      (If the value from R1 is equal to 2,)
           G01 X10      (the X axis is moved to position 10)
           $ELSE
           G01 Y20      (otherwise the Y axis is moved to position 20)
           $ENDIF
```

10.2 SWITCH 문

SWITCH 문은 몇몇 선택을 위한 특별한 선택 타입입니다. 만약 몇몇 구조의 값(CASE 문)이 일치한다면 표현(SWITCH 문)이 확인하기 위해 점검됩니다; 만약 케이스에 해당한다면, 각각의 블록이 실행됩니다.

각각의 CASE 문은 BREAK 로 끝나야 합니다. 이것은 즉시 끝내기 위해 SWITCH 문을 사용하고 프로그램은 ENDSWITCH 구조 이후에 계속 됩니다.

```

Syntax:      $SWITCH Value
                  $CASE 0
                      Block 1
                  $BREAK

                  $CASE 1
                      Block 2
                  $BREAK

                  :
                  :
                  $DEFAULT
                      Block n

                $ENDSWITCH
  
```

만약 CASE 문이 SWITCH 값과 일치하지 않는다면, DEFAULT 가 실행됩니다. DEFAULT 구조는 옵션으로 요구사항이 아닙니다.

```

Example:    $$SWITCH R1          (If the parameter R1)
                  $CASE 1          (has value 1,)
                      X10          (the X axis is moved to position 10,)
                  $BREAK

                  $CASE 2          (for value 2)
                      X20          (to position 20,)
                  $BREAK

                  $DEFAULT          (in any other case)
                      X0           (to position 0)

                  $ENDSWITCH
  
```

10.3 FOR 문

카운터 루프가 FOR 문을 사용해서 생성 될 수 있습니다. 시작 조건과 끝 조건과 단계 크기는 이 명령에 입력되어야 합니다. FOR 문은 반듯이 ENDFOR 로 끝나야 합니다. 만약 파라미터가 TO 보다 더 큰 값(또는 카운터 방향이 negative 이 때, TO 보다 더 작은 값)을 받는다면, 카운터 루프는 정지됩니다.

Syntax: \$FOR *Parameter=FROM, TO [,STEP]*
 Block
 \$ENDFOR

또한 연산식이 FROM, TO, STEP 으로 주어질 수 있습니다. 만약 단계 크기가 주어지지 않았다면, +1 값이 자동적으로 사용됩니다. 만약 카운팅 방향이 negative 라면 negative 값은 반듯이 STEP 으로 받아야 합니다.

BREAK ...카운터 문의 독립적 루프에서 빠져나옵니다.
 프로그램은 ENDFOR 명령 이후에 계속됩니다.

CONTINUE ...다음 명령 고려하지 않고 반복되는 루프의 결과를 가져옵니다.

Example: \$FOR R1=10,100,5 (The X axis is moved in steps of 5 from 10 to 100)
 XR1 G60
 \$ENDFOR

10.4 WHILE 문

시작에서 동작 조건을 포함한 루프는 WHILE 문과 함께 만들어 질 수 있습니다. 루프가 조건을 만족 할 때까지 계속 반복됩니다. 그렇지 않으면 루프를 빠져나가고 ENDWHILE 명령 이후에 프로그램이 계속됩니다.

Syntax: \$WHILE *Condition*
 Block
 \$ENDWHILE

BREAK ...루프를 빠져나갑니다.
 ENDWHILE 명령 후에 프로그램이 계속됩니다.

CONTINUE ...다음 명령을 고려하지 않고 반복하는 루프의 결과를 가져옵니다.

Beispiel: R1=100
 \$WHILE R1 (The loop is repeated as long as R1 is larger or equal to 0.5)
 R1=R1-1.5
 YR1 G60
 \$ENDWHILE

10.5 DO 문

마지막에서 동작 조건을 포함한 루프는 DO 문과 함께 생성됩니다. 루프는 조건을 만족 할 때까지 반복됩니다. 그렇지 않으면 루프를 빠져나가고 ENDDO 명령 이후에 프로그램이 계속됩니다.

조건이 루프의 마지막에 있기 때문에, WHILE 루프와 다르게, 루프는 적어도 한 번은 작동이 되고 블럭이 실행됩니다.

Syntax: \$DO
 Block
 \$ENDDO *Condition*

BREAK ... 루프를 빠져나갑니다.
 ENDDO 명령 이후에 프로그램이 계속됩니다.

CONTINUE ...다음 명령을 고려하지 않고 반복하는 루프의 결과를 가져옵니다.

Example: R1=10
 \$DO
 R1=R1-0.6
 XR1
 \$ENDDO R1 (The loop is repeated as long as R1 is larger or equal to 0.5)