

5. 프로그램 예

본 장은 다양한 예제 응용프로그램을 수록하여 사용자가 원하는 작업을 쉽게 구현할 수 있도록 돕는 것을 목적으로 하고 있습니다. 처음 사용자를 위한 부분과 기본적인 단순 이동 프로그램부터 채널간 협조 운전 및 SYNC문과 SPDR을 사용한 이동중 속도 변경기능과 같은 고급기능을 소개합니다.

5-1 기초 프로그램 작성(처음 사용자를 위하여)

여기에서는 로봇이나 모션 제어기 또는 본 제어기를 처음 사용하시는 분들을 위하여 가장 기본적인 모션 프로그램의 작성과 실행에 대하여 설명합니다.

- 1) 먼저 원점을 수행합니다. 제어기에게 로봇의 현재 위치를 알려주는 역할을 합니다. Incremental엔코더를 사용한 로봇에서는 센서를 사용하여 로봇의 현재 위치를 파악하기 때문에 센서가 있는 위치로 이동하는 작업을 하게 되고 이를 원점 복귀하고 합니다. 원점 복귀에는 여러가지 방법이 있으며 원점 복귀를 수행하지 않고 제어기의 전원을 켜고 때 로봇의 위치를 원점으로 설정하는 방법도 있습니다. 원점 방법과 관련한 파라미터는 7장을 참조하십시오.

원점 복귀 명령은 오퍼레이팅 로더나 접점, PC를 사용하여 줄 수 있습니다. 오퍼레이팅 로더를 이용한 방법은 3장을 참조하시고, 접점을 이용한 방법은 6장을, PC를 이용한 방법은 별도의 Hostpack 매뉴얼을 참조하십시오.

- 2) 모션 프로그램을 작성합니다. 가장 기본적인, 두 점을 왕복하는 프로그램을 예로 듭니다. 모션 프로그램은 오퍼레이팅 로더나 PC를 사용하여 작성할 수 있습니다. 3장을 참조하여 다음과 같은 프로그램을 만듭니다. SPD명령어를 사용하여 로봇의 이동속도를 설정합니다. 이 명령어를 사용하지 않으면 기본값인 파라미터의 JogSpd0값이 사용됩니다. P0와 P1은 로봇의 특정위치를 나타내는 위치값으로 포인트 파일에 저장되어 있습니다. ‘//’이후에 오는 문자들은 실행과 관련이 없는 프로그램의 설명을 위한 주석이므로 입력하지 않아도 됩니다. 또한 오퍼레이팅 로더에서는 한글입력이 지원되지 않습니다.

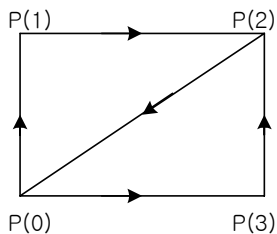
L000	SPD=1000	//10%로 속도를 설정합니다. 10000일 경우 100%속도입니다.
L001	TAG A	//레이블을 설정합니다. GOTO를 사용하여 이곳으로 이동할 수 있습니다.
L002	MPTP P0	//P0 위치로 이동합니다. P0가 없을 경우 에러 발생.
L003	MPTP P1	//P1 위치로 이동합니다. P1이 없을 경우 에러 발생.
L004	GOTO A	//‘A’ 레이블이 있는 L001로 이동합니다.
L005	MEND	//프로그램을 종료합니다.

- 3) 포인트 파일을 만들어 P0와 P1의 위치를 지정합니다. 모션 프로그램이 처음 실행될 때에 자신의 프로그램 번호와 같은 포인트 파일을 찾아서 있을 경우에 이를 사용하게 됩니다. 포인트 파일은 모션 프로그램을 만들 때 자동으로 생성되지 않으니 주의하십시오. 포인트 파일내에 해당 포인트가 존재하지 않는 경우에는 모션 프로그램을 실행할 도중에 에러가 발생하게 됩니다.
3장을 참조하시어 모션 프로그램과 번호가 같은 포인트 파일을 만듭니다(이름은 달라도 무관합니다). 포인트 파일안에는 1000개까지의 포인트가 저장될 수 있습니다. 첫번째와 두 번째에 저장된 포인트를 각각 P0, P1이라고 합니다. 포인트값을 만드는 방법은 손이나 Jog로 로봇을 움직여서 그 위치를 저장하는 방법과 직접 값을 입력하는 방법이 있습니다.
- 4) 모션 프로그램을 실행합니다. 오퍼레이팅 로더나 점점, PC를 이용하여 실행할 수 있으며, 오퍼레이팅 로더를 사용할 경우는 3장을 참조하십시오. 그 외 점점을 이용한 실행은 6장을 PC를 이용한 실행은 Hostpack 매뉴얼을 참조하십시오.
- 5) 모션 프로그램을 정지합니다. 오퍼레이팅 로더의 STOP 버튼을 누릅니다. 서보의 전원은 ON 상태가 유지됩니다.
- 6) 서보 전원을 OFF 합니다. 오퍼레이팅 로더에서 2nd + Servo 키를 누릅니다.

5-2 단순 PTP 이동 반복 / 횟수 제한 프로그램

프로그램 설명 # 1

1. 표시된 위치를 설정한 입력 점점이 입력되면 50%의 속도로 단순 반복한다.
2. 1회 반복 후 출력 점점 B(2).0 을 On시킨 후 50[ms] 후에 Off 시킨다.



프로그램 작성 Tip

1. 설정한 입력 점점이 입력되면 ; STBY B(0).0==1명령 사용
2. 단순 반복한다 ; TAG ~ GOTO 명령 사용
3. 설정한 출력 점점을 On ; B(2).0=1 명령 사용
4. 50[ms] 후에 ; WAIT=5 명령 사용

프로그램 예 #1

```

L000 SPD=5000          // 속도를 50%로 지정합니다.
L001 STBY B(0).0==1    // 입력 접점이 ON이 될 때까지 대기합니다.
L002 LOOP (B(0).0 == 1) // 입력 접점이 OFF 될 때까지 무한 수행합니다.
L003 MPTP P0           // 임의의 위치에서 P0 위치로 PTP 이동합니다.
L004 MPTP P1           // P1 위치로 PTP 이동합니다.
L005 MPTP P2           // P2 위치로 PTP 이동합니다.
L006 MPTP P0           // P0 위치로 PTP 이동합니다.
L007 MPTP P3           // P3 위치로 PTP 이동합니다.
L008 MPTP P2           // P2 위치로 PTP 이동합니다.
L009 B(2).0=1          // 출력 접점을 On시킵니다.
L010 WAIT=5            // 50[ms]동안 대기합니다.
L011 B(2).0=0          // 출력 접점을 Off시킵니다.
L012 ENDL              // 무한 루프의 끝을 지정합니다.
L013 MEND              // 주 프로그램 끝을 선언합니다.

```

프로그램 설명 #2

- 상기 프로그램 조건을 갖추되 횟수는 5회로 제한한다.
- 모두 끝난 후 설정한 입력 접점이 On될 때까지 대기한다.

프로그램 작성 Tip

- 횟수는 5회로 제한 ; FOR ~ ENDF 명령 사용
- 모두 끝난 후 설정한 입력 접점이 On될 때까지 대기 ; 5회 실행 후 프로그램 Line이 STBY B(0).0==1 위로 올라가게 프로그램 작성

프로그램 예 #2

```

L000 INT I              // 정수형 변수 I를 선언합니다.
L001 SPD=5000           // 속도를 50%로 지정합니다.
L002 TAG START          // Tag를 설정합니다.
L003 STBY B(0).0==1     // 입력 접점이 ON 될 때까지 대기합니다.
L004 FOR I=1 TO 5       // 횟수를 5회(I=1,2,3,4,5)로 지정합니다.
L005 MPTP P0            // 임의의 위치에서 P0 위치로 PTP 이동합니다.
L006 MPTP P1            // P1 위치로 PTP 이동합니다.
L007 MPTP P2            // P2 위치로 PTP 이동합니다.
L008 MPTP P0            // P0 위치로 PTP 이동합니다.
L009 MPTP P3            // P3 위치로 PTP 이동합니다.
L010 MPTP P2            // P2 위치로 PTP 이동합니다.
L011 B(0).0=1           // 출력 접점 0번을 On시킵니다.

```

```

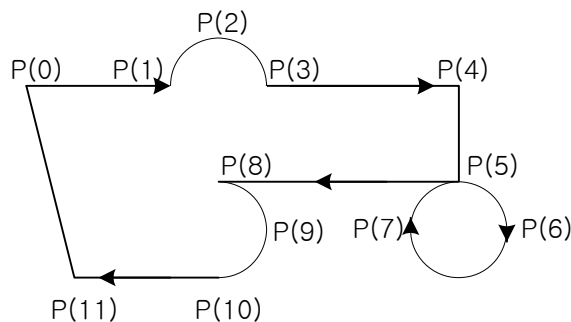
L012  WAIT=5           // 50[ms]동안 대기합니다.
L013  B(0).0=0         // 출력 접점 0번을 Off시킵니다.
L014  ENDF             // FOR문을 닫아줍니다.
L015  GOTO START       // 프로그램 진행을 START Tag가 있는 곳으로 이동합니다.
L016  MEND             // 주 프로그램 끝을 선언합니다.

```

5-3 직선, 원호, 원 보간 프로그램

프로그램 설명

- 1) 표시된 위치를 P0 ~ P5 구간까지는 70%의 속도로 이동한다.
- 2) P6 ~ P8 구간에서는 30%의 속도로 이동한다.
- 3) P8 ~ P0 구간에서는 20%의 속도로 이동한다.
- 4) 구간별 끊어짐이 없이 전 구간을 1회 반복한 후 프로그램을 빠져 나온다.



프로그램 작성 Tip

- 1) 직선 구간 ; MLIN 사용
- 2) 원호 구간 ; MARC 사용
- 3) 원 구간 ; MCIR 사용
- 4) 속도 변화 ; SPD 명령 사용
- 5) 구간별 끊어짐이 없이 ; ACCR, DECR명령어를 사용하여 가감속 시간을 1로 준다

프로그램 예

```

L000  SPD=7000         // 속도를 70%로 지정합니다.
L001  ACCR 1           // 샘플링 시간 동안 한번에 목표 속도까지 가속하여 이동합니다.
L002  DECR 1           // 샘플링 시간 동안 한번에 정지합니다.
L003  MPTP P0          // 현재 위치에서 P0 위치로 PTP 이동합니다.
L004  MLIN P1          // P1 위치로 직선 보간 이동합니다.
L005  MARC P2,P3       // P2 경유점을 지나 P3 위치로 원호 보간 이동합니다.
L006  MLIN P4          // P4 위치로 직선 보간 이동합니다.
L007  MLIN P5          // P5 위치로 직선 보간 이동합니다.
L008  SPD=3000         // 속도를 30%로 지정합니다.
L009  MCIR P6,P7       // P6 경유점을 지나 P7 위치로 원 보간 이동합니다.

```

```

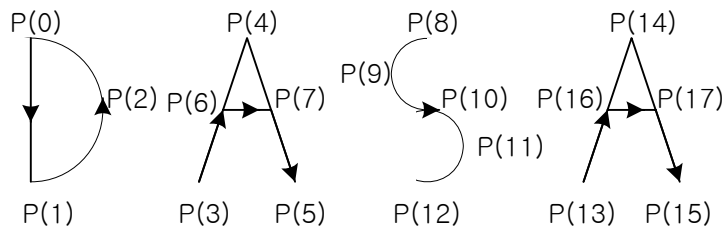
L010  SPD=2000      // 속도를 20%로 지정합니다.
L011  MLIN P8        // P8 위치로 직선 보간 이동합니다.
L012  MARC P9,P10    // P9 경유점을 지나 P10 위치로 원호 보간 이동합니다.
L013  MLIN P11       // P11 위치로 직선 보간 이동합니다.
L014  MLIN P0        // P0 위치로 직선 보간 이동합니다.
L015  MEND           // 주 프로그램 끝을 선언합니다.

```

5-4 직선, 원호, 원 보간 응용 프로그램

프로그램 설명

- 1) 표시된 글자를 Drawing하는데 대기 위치에서 이동, 각각의 속도는 50%의 속도로 한다.
- 2) 입력 접점 B(0).0 이 On되면 “D” 를 Drawing 한다.
- 3) 입력 접점 B(0).1 이 On되면 첫번째 “A” 를 Drawing 한다.
- 4) 입력 접점 B(0).2 이 On되면 “S” 를 Drawing 한다.
- 5) 입력 접점 B(0).3 이 On되면 두번째 “A” 를 Drawing 한다.



프로그램 작성 Tip

- 1) 직선 구간 ; MLIN 사용
- 2) 원호 구간 ; MARC 사용
- 3) 대기 위치에서 이동 ; 임의의 한 점을 대기 위치에서 사용 P20(임의 설정 가능)에 사용
- 4) 입력 접점 B(0).0 ~ B(0).2번이 On되면 ; IF 조건 비교문 사용
- 5) D(A,S)를 Drawing ; Sub 프로그램 또는 TAG ~ GOTO 문 사용

프로그램 예

```

L000  SPD=5000      // 속도를 50%로 지정합니다.
L001  MPTP P20       // Drawing 전 대기 위치로 이동합니다.
L002  LOOP 1         // ENDL까지 무한 루프로 지정합니다.
L003  IF B(0).0==1   // 입력 접점이 On이 되는지 판단합니다.
L004  CALL DRWD      // 조건이 맞으면 DRWD 이름을 갖는 Sub 프로그램 호출합니다.
L005  ENDI           // IF문을 닫아줍니다.
L006  IF B(1).0==1   // 입력 접점이 On이 되는지 판단합니다.
L007  CALL DRWA1     // 조건이 맞으면 DRWA1 이름을 갖는 Sub 프로그램 호출합니다.

```

L008	ENDI	// IF문을 닫아줍니다.
L009	IF B(2).0==1	// 입력 접점이 On이 되는지 판단합니다.
L010	CALL DRWS	// 조건이 맞으면 DRWS 이름을 갖는 Sub 프로그램 호출합니다.
L011	ENDI	// IF문을 닫아줍니다.
L012	IF B(3).0==1	// 입력 접점이 On이 되는지 판단합니다.
L013	CALL DRWA2	// DRWA2 이름을 갖는 Sub 프로그램 호출합니다.
L014	ENDI	// IF문을 닫아줍니다.
L015	ENDL	// LOOP문을 닫아줍니다.
L016	MEND	// 주 프로그램의 끝을 선언합니다.
L017	SPGM DRWD	// DRWD의 이름을 갖는 Sub 프로그램을 시작합니다.
L018	MPTP P0	// P0 위치로 PTP 이동합니다.
L019	MLIN P1	// P1 위치로 직선 보간 이동합니다.
L020	MARC P2,P0	// P2 경유점을 지나 P3 위치로 원호 보간 이동합니다.
L021	MPTP P20	// 대기 위치로 이동합니다.
L022	SEND	// DRWD의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.
L023	SPGM DRWA1	// DRWA1의 이름을 갖는 Sub 프로그램을 시작합니다.
L024	MPTP P3	// P3 위치로 PTP 이동합니다.
L025	MLIN P4	// P4 위치로 직선 보간 이동합니다.
L026	MLIN P5	// P5 위치로 직선 보간 이동합니다.
L027	ARCH=10	// 위치 이동 전 Z축의 위치를 원점 기준 10mm로 설정합니다.
L028	MPTP P6	// Z축 먼저 10mm 위치로 상승 후 P6 위치로 PTP 이동합니다.
L029	MLIN P7	// P7 위치로 직선 보간 이동합니다.
L030	ARCH=OFF	// ARCH 설정을 해제합니다.
L031	MPTP P20	// 대기 위치로 이동합니다.
L032	SEND	// DRWA1의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.
L033	SPGM DRWS	// DRWS의 이름을 갖는 Sub 프로그램을 시작합니다.
L034	MPTP P8	// P8 위치로 PTP 이동합니다.
L035	MARC P9,P10	// P9 경유점을 지나 P10 위치로 원호 보간 이동합니다.
L036	MARC P11,P12	// P11 경유점을 지나 P12 위치로 원호 보간 이동합니다.
L037	MPTP P20	// 대기 위치로 이동합니다.
L038	SEND	// DRWS의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.
L039	SPGM DRWA2	// DRWA2의 이름을 갖는 Sub 프로그램을 시작합니다.
L040	MPTP P13	// P13 위치로 PTP 이동합니다.
L041	MLIN P14	// P14 위치로 직선 보간 이동합니다.
L042	MLIN P15	// P15 위치로 직선 보간 이동합니다.
L043	ARCH=10	// 위치 이동 전 Z축의 위치를 원점 기준 10mm로 설정합니다.
L044	MPTP P16	// Z축 먼저 10mm 위치로 상승 후 P16 위치로 PTP 이동합니다.
L045	MLIN P17	// P17 위치로 직선 보간 이동합니다.

```

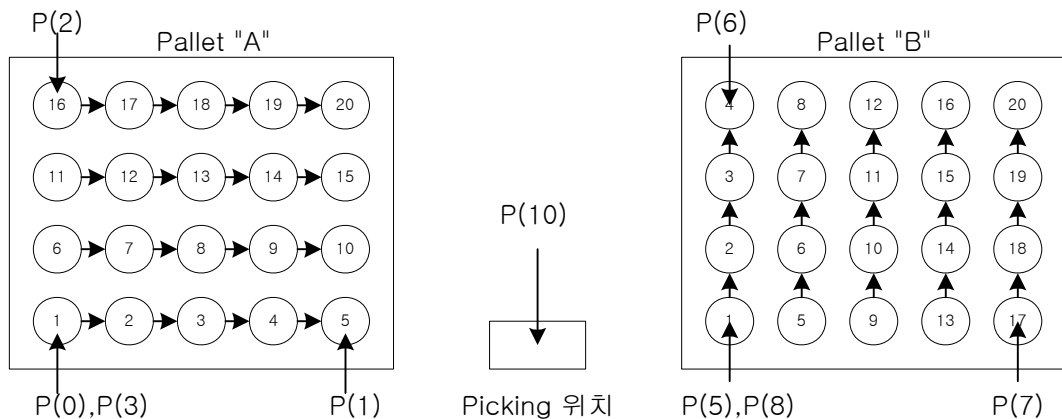
L046  ARCH=0          // ARCH 설정을 해제합니다.
L047  MPTP P20         // 대기 위치로 이동합니다.
L048  SEND            // DRWA2의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.

```

5-5 Palletizing 프로그램

프로그램 설명

- 1) Picking 위치에서 물품을 Picking하여 Pallet "A"에 먼저 Place한 후, 두 번째 물품을 Picking하여 Pallet "B"에 Place 한다.
- 2) 이 Place 작업은 Pallet "A"와 Pallet "B"를 교대로 한다.
- 3) 작업의 순서는 그림의 표기와 같이 실시한다.
- 4) Picking하는 위치와 Pallet의 높이는 맞지 않으므로 주의한다.
- 5) Picking 물품 도착 신호는 입력접점 B(0).0, Hand On 신호는 입력접점 B(0).1, Hand Off 신호는 입력접점 B(0).2 로 한다
- 6) Picking 위치 도착 신호는 출력접점 B(2).0번, Pallet "A" 위치 도착 신호는 출력접점 B(2).1, Pallet "B" 위치 도착 신호는 출력접점 B(2).2로 한다.
- 7) Hand On/Off의 외부 Solenoid 신호는 출력접점 B(2).5로 한다.



프로그램 작성 Tip

- 1) Pallet "A", Pallet "B" ; Pallet 2개가 있으므로 MPLT 명령 2개
- 2) 교대로 한다 ; Picking 위치를 Sub 프로그램으로 구성
- 3) 작업의 순서는 ; Pallet "A"와 Pallet "B"의 Teaching을 그림과 같이 구성
- 4) 높이가 맞지 않으므로 ; ARCH 명령 사용

3. 작업 팔레트 환경 설정

- 1) Pallet "A"의 설정 : PWRK0, PCNT0에 설정
PWRK(0) = (5,4,1)
PCNT(0) = (1,1,1)

2) Pallet “B”의 설정 : PWRK1,PCNT1에 설정

PWRK(1) = (4,5,1)

PCNT(1) = (1,1,1)

4. 프로그램 예

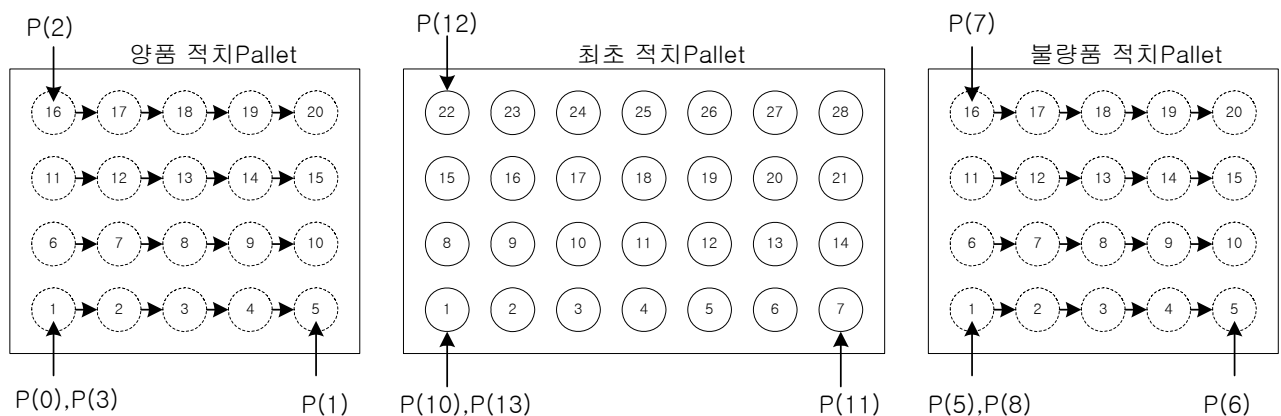
```

L000  ARCH=5           // 위치 이동 전 Z축의 위치를 원점 기준 5mm로 설정합니다.
L001  SPD=10000        // 속도를 100%로 지정합니다.
L002  PCNT(0)=(1,1,1)  // 팔레트 0번의 Number를 초기화 합니다.
L003  PCNT(1)=(1,1,1)  // 팔레트 1번의 Number를 초기화 합니다.
L004  PWRK(0)=(5,4,1)  // 팔레트 0번의 작업 형태를 설정 합니다.
L005  PWRK(1)=(4,5,1)  // 팔레트 1번의 작업 형태를 설정 합니다.
L006  TAG START       // Tag START을 지정합니다.
L007  CALL PICK       // PICK 이름을 갖는 Sub 프로그램 호출합니다.
L008  MPLT<0,P0,0>    // 팔레트 0번의 P0 위치로 이동합니다.
L009  B(2).1=1        // 로봇의 Pallet “A” 위치 도착 신호를 출력합니다.
L010  STBY B(0).2==1  // Hand Off 신호를 기다립니다.
L011  B(2).5=0        // 외부 Solenoid를 Off하여 Hand Off 시킵니다.
L012  B(2).1=0        // 로봇의 Pallet “A” 위치 도착 신호를 해제합니다.
L013  CALL PICK       // PICK 이름을 갖는 Sub 프로그램 호출합니다.
L014  MPLT<1,P5,0>    // 팔레트 1번의 P5 위치로 이동합니다.
L015  B(2).2=1        // 로봇의 Pallet “B” 위치 도착 신호를 출력합니다.
L016  B(0).2=1        // Hand Off 신호를 기다립니다.
L017  B(2).5=0        // 외부 Solenoid를 Off하여 Hand Off 시킵니다.
L018  B(2).2=0        // 로봇의 Pallet “B” 위치 도착 신호를 해제합니다.
L019  GOTO START      // 프로그램 진행을 START Tag가 있는 곳으로 이동합니다.
L020  MEND            // 주 프로그램 끝을 선언합니다.
L021  SPGM PICK       // PICK의 이름을 갖는 Sub 프로그램을 시작합니다.
L022  MPTP P10        // Picking 위치 P10 위치로 PTP 이동합니다.
L023  B(2).0=1        // Picking 위치 도착 신호를 출력합니다.
L024  STBY B(0).0==1  // 물품 도착 신호를 기다립니다.
L025  STBY B(0).1==1  // Hand On 신호를 기다립니다.
L026  B(2).5=1        // 외부 Solenoid를 On하여 Hand On 시킵니다.
L027  B(2).0=0        // Picking 위치 도착 신호를 해제합니다.
L028  SEND           // PICK의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.
    
```

5-6 Palletizing 프로그램 응용 #1

1. 프로그램 설명

- 1) 최초 적치 Pallet에서 물품을 Picking하여 양품과 불량품의 신호를 받는다.
- 2) 양품의 신호는 입력 접점 B(0).0번이 On, 불량품의 신호는 입력 접점 B(0).3번이 On으로 받는다.
- 3) 최초 적치 Pallet는 모든 물품이 선별되면 빠지고, 양(불량)품 Pallet는 Pallet에 모두 채워지면 빠진다.
- 4) 각 Pallet의 도착 신호는 최초 적치 Pallet는 입력 접점 B(0).5, 양품 적치 Pallet(빈 Pallet)는 입력 접점 B(0).6, 불량품 적치 Pallet(빈 Pallet)는 입력 접점 B(0).7번으로 한다.
- 5) Picking하는 위치와 Pallet의 높이는 맞지 않으므로 주의한다.
- 6) Pallet 도착 후 외부 PLC로부터 받는 Hand On 신호는 입력 접점 B(0).1번, Hand Off 신호는 입력 접점 B(0).2번으로 한다.
- 7) 물품의 검사는 최초 적치 Pallet에서 물품을 들어 올린 후 Vision 검사로 한다.
- 8) 최초 적치 Pallet 도착 신호는 출력 접점 B(3).0번, 양품 Pallet 위치 도착 신호는 출력 접점 B(3).1번, 불량품Pallet 위치 도착 신호는 출력 접점 B(3).2번으로 한다.
- 9) Vision 검사 도착 위치는 출력 접점 B(3).3번으로 하고, 검사 시작과 완료는 입력 접점 B(0).4번으로 한다.
- 10) Hand On/Off 요구 외부 Solenoid 신호는 출력 접점 B(3).5, 출력 접점 B(3).6번으로 한다.
- 11) 최초 적치 Pallet의 모든 선별 작업이 끝나면 출력 접점 B(4).0을 50[ms] 동안 발생한다.
- 12) 양품 적치 Pallet의 모든 적치가 끝나면 출력 접점 B(4).1을 50[ms] 동안 발생한다.
- 13) 불량품 적치 Pallet의 모든 적치가 끝나면 출력 접점 B(4).2를 50[ms] 동안 발생한다.



2. 프로그램 작성 Tip

- 1) 최소 적치 Pallet에서 ... 양품 ... 불량품 ; MPLT 명령 3개 사용
- 2) 신호 ; BIN, BOUT 명령 사용
- 3) 모두 선별되면 빠지고 모두 채워지면 빠진다 ; INT 명령 사용 개수 Count
- 4) Pallet 높이는 맞지 않으므로 ; ARCH 명령 사용 Z축 Arch 형태로 이동
- 5) 물품을 들어 올린 후 Vision으로 ; 검사 Point 설정

3. 작업 팔레트 환경 설정

프로그램 예 5-4 Palletizing 프로그램의 작업 팔레트 설정과 형식으로 설정

4. 프로그램 예

```

L000  INT I,J,K          // 정수형 변수 I, J, K를 선언합니다.
L001  I=1                 // 정수형 변수 I를 1로 초기화 합니다.
L002  J=1                 // 정수형 변수 J를 1로 초기화 합니다.
L003  K=1                 // 정수형 변수 K를 1로 초기화 합니다.
L004  ARCH=5             // 위치 이동 전 Z축의 위치를 원점 기준 5mm로 설정합니다.
L005  SPD=10000          // 속도를 100%로 지정합니다.
L006  PWRK(0)=(7,4,1)    // 팔레트 0번의 작업 형태를 설정 합니다.
L007  PWRK(1)=(5,4,1)    // 팔레트 1번의 작업 형태를 설정 합니다.
L008  PWRK(2)=(5,4,1)    // 팔레트 2번의 작업 형태를 설정 합니다.
L009  TAG DASA           // DASA Tag를 지정합니다.
L010  PCNT(0)=(1,1,1)    // Work ID 0번(최초 적치 Pallet)의 Number를 초기화 합니다.
L011  PCNT(1)=(1,1,1)    // Work ID 1번(양품 적치 Pallet)의 Number를 초기화 합니다.
L012  PCNT(2)=(1,1,1)    // Work ID 2번(불량품 적치 Pallet)의 Number를 초기화 합니다.
L013  TAG START         // START Tag를 설정합니다.
L014  CALL PICK          // PICK 이름을 갖는 Sub 프로그램 호출합니다.
L015  CALL INSP          // INSP 이름을 갖는 Sub 프로그램 호출합니다.
L016  IF I==28 THEN      // 정수형 변수 I가 28이 되는지 조건 판단합니다.
L017  GOTO BLAN          // I가 28이 되면 BLAN Tag가 있는 곳으로 이동합니다.
L018  IF J==20 THEN      // 정수형 변수 J가 20이 되는지 조건 판단합니다.
L019  GOTO GOOD          // J가 20이 되면 GOOD Tag가 있는 곳으로 이동합니다.
L020  IF K==20 THEN      // 정수형 변수 K가 20이 되는지 조건 판단합니다.
L021  GOTO BAD           // J가 20이 되면 BAD Tag가 있는 곳으로 이동합니다.
L022  ENDI               // IF K==20... 의 조건문을 닫습니다.
L023  ENDI               // IF J==20... 의 조건문을 닫습니다.
L024  ENDI               // IF I==28... 의 조건문을 닫습니다.
L025  GOTO START         // 프로그램 진행을 START Tag가 있는 곳으로 이동합니다.
L026  TAG BLAN           // BLAN Tag를 설정합니다.
L027  I=1                // 정수형 변수 I를 1로 초기화 합니다.
L028  B(3).0=1           // 최초 적치 Pallet의 선별작업이 끝났음을 알립니다.
L029  WAIT=50            // 50[ms]동안 대기합니다.
L030  B(3).0=0           // 작업 완료 신호를 Off 합니다.
L031  GOTO START         // 프로그램 진행을 START Tag가 있는 곳으로 이동합니다.
L032  TAG GOOD           // GOOD Tag를 설정합니다.
L033  J=1                // 정수형 변수 J를 1로 초기화 합니다.
L034  B(3).1=1           // 양품 적치 Pallet가 모두 찾음을 알립니다.

```

```

L035 WAIT=5          // 50[ms]동안 대기합니다.
L036 B(3).1=0        // 작업 완료 신호를 Off 합니다.
L037 GOTO START      // 프로그램 진행을 START Tag가 있는 곳으로 이동합니다.
L038 TAG BAD         // BAD Tag를 설정합니다.
L039 K=1             // 정수형 변수 K를 1로 초기화 합니다.
L040 B(3).2=1        // 불량품 적치 Pallet가 모두 찾음을 알립니다.
L041 WAIT=5          // 50[ms]동안 대기합니다.
L042 B(3).2=0        // 작업 완료 신호를 Off 합니다.
L043 GOTO START      // 프로그램 진행을 START Tag가 있는 곳으로 이동합니다.
L044 GOTO DASA       // 프로그램 진행을 DASA Tag가 있는 곳으로 이동합니다.
L045 MEND            // 주 프로그램 끝을 선언합니다.
L046 SPGM PICK       // PICK의 이름을 갖는 Sub 프로그램을 시작합니다.
L047 STBY B(0).5==1  // 최초 적치 Pallet가 있는지 확인합니다.
L048 MPLT<0,P10,0>   // Pallet 위치로 이동합니다.
L049 B(2).0=1        // Pallet 위치 도착 신호를 출력합니다.
L050 B(2).5=1        // Hand On 요청 Solenoid 신호를 출력합니다.
L051 STBY B(0).1==1  // 외부에서 Hand On 신호를 받을 때까지 대기합니다.
L052 B(2).0=0        // Pallet 위치 도착 신호를 Off합니다.
L053 B(2).5=0        // Hand On 요청 Solenoid 신호를 Off합니다.
L054 I=I+1           // 1회 작업 후 정수형 변수 I 값을 1 증가시킵니다.
L055 IF (PCNT(0).1 == 7) && (PCNT(0).2==4) // 최초 적치 팔레트 작업 끝
L056 B(4).0 = 1      // 최초 적치 팔레트 작업 끝 신호 On
L057 WAIT=5          // 50[ms]동안 대기 합니다.
L058 B(4).0 = 0      // 최초 적치 팔레트 작업 끝 신호 Off
L059 ENDI
L060 SEND            // PICK의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.
L061 SPGM INSP       // INSP의 이름을 갖는 Sub 프로그램을 시작합니다.
L062 MPTP P50        // 검사 위치로 이동합니다.
L063 B(2).3=1        // 검사 위치 이동 완료 신호를 출력합니다.
L064 STBY B(0).4==1  // 검사 시작 신호를 받을 때까지 대기합니다.
L065 IF (B(0).0==1)&&(B(0).4==0) // 양품 신호와 검사 완료 신호가 들어오는지 판단
L066 THEN            // 양품 신호와 검사 완료 신호가 모두 들어오면
L067 CALL GOOD       // GOOD 이름을 갖는 Sub 프로그램을 호출합니다.
L068 IF (B(0).3==1)&&(B(0).4==0) // 불량품 신호와 검사 완료 신호가 들어오는지 판단
L069 THEN            // 불량품 신호와 검사 완료 신호가 모두 들어오면
L070 CALL BAD        // BAD 이름을 갖는 Sub 프로그램을 호출합니다.
L071 ENDI            // IF(IN11==1)... 조건문을 닫습니다.
L072 ENDI            // IF(IN12==1)... 조건문을 닫습니다.

```

```

L073  SEND          // INSPECT의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.
L074  SPGM GOOD     // GOOD 이름을 갖는 Sub 프로그램을 시작합니다.
L075  MPLT<1,D(0),0> // 양품 적치 Pallet로 이동합니다.
L076  B(2).1=1      // 양품 Pallet 위치 도착 신호를 출력합니다.
L077  B(2).6=1      // Hand Off 요청 Solenoid 신호를 출력합니다.
L078  STBY B(0).2==1 // 외부에서 Hand Off 신호를 받을 때까지 대기합니다.
L079  B(2).1=0      // 양품 Pallet 위치 도착 신호를 Off합니다.
L080  B(2).6=0      // Hand Off 요청 Solenoid 신호를 Off합니다.
L081  J=J+1         // 1회 양품 적치 후 정수형 변수 J 값을 1 증가시킵니다.
L082  IF (PCNT(1).1 == 5) && (PCNT(1).2==4) // 양품 적치 팔레트 작업 끝
L083  B(4).1 = 1     // 양품 적치 팔레트 작업 끝 신호 On
L084  WAIT=5         // 50[ms]동안 대기 합니다.
L085  B(4).1 = 0     // 양품 적치 팔레트 작업 끝 신호 Off
L086  ENDI
L087  SEND          // GOOD의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.
L088  SPGM BAD      // BAD 이름을 갖는 Sub 프로그램을 시작합니다.
L089  MPLT<2,P5,0>  // 불량품 적치 Pallet로 이동합니다.
L090  B(2).2=1      // 불량품 Pallet 위치 도착 신호를 출력합니다.
L091  B(2).6=1      // Hand Off 요청 Solenoid 신호를 출력합니다.
L092  STBY B(0).2==1 // 외부에서 Hand Off 신호를 받을 때까지 대기합니다.
L093  B(2).2=0      // 불량품 Pallet 위치 도착 신호를 Off합니다.
L094  B(2).6=0      // Hand Off 요청 Solenoid 신호를 Off합니다.
L095  K=K+1         // 1회 불량품 적치 후 정수형 변수 K 값을 1 증가시킵니다.
L096  IF (PCNT(2).1 == 5) && (PCNT(2).2==4) // 불량품 적치 팔레트 작업 끝
L097  B(4).2 = 1     // 불량품 적치 팔레트 작업 끝 신호 On
L098  WAIT=5         // 50[ms]동안 대기 합니다.
L099  B(4).2 = 0     // 불량품 적치 팔레트 작업 끝 신호 Off
L100  ENDI
L101  SEND          // BAD의 이름을 갖는 Sub 프로그램을 되돌려 줍니다.

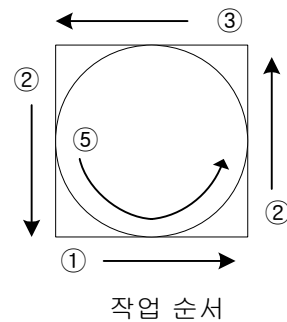
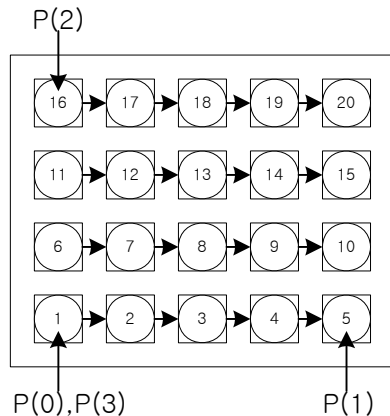
```

5-7 Palletizing 프로그램 응용 #2

1. 프로그램 설명

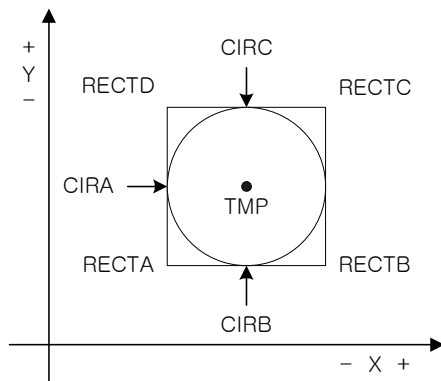
- 1) 일정하게 배열된 정사각형 및 원을 그린다.
- 2) 정사각형 및 원의 크기는 동일하고 최소한의 Point를 Teaching하여 작업한다.
- 3) 예제 프로그램은 가로 x 세로 10mm의 정사각형, 지름이 10mm인 원을 그린다.
- 4) 입력 포트 B(0)가 “1100 1000”로 입력되면 작업 시작한다.

- 5) 모든 작업을 종료한 후 출력 포트 B(3)을 “0011 0101”로 출력한다.
- 6) 작업 시작과 종료는 원점 위치에서 시작한다.



2. 프로그램 작성 Tip

- 1) 일정하게 배열된 : Palletizing 응용
- 2) 정사각형 및 원의 크기는 동일 : 위치형 변수 사용(크기를 하나의 변수로 지정함)
- 3) IN, OUT 처리 : 8접점 동시 처리
- 4) 원의 중심점에서 각각의 위치는 다음과 같이 저장
- 5) 중심점의 위치 : 위치 변수 “TMP”
- 6) 직사각형 위치 : 위치 변수 “RECTA, RECTB, RECTC, RECTD”
- 7) 원의 위치 : 위치 변수 “CIRA, CIRB, CIRC”



* 위치 설명 :

TMP : 원의 중심점

RECTA = TMP + <-5.000, -5.000, 0.000>

RECTB = TMP + <5.000, -5.000, 0.000>

RECTC = TMP + <5.000, 5.000, 0.000>

RECTD = TMP + <-5.000, 5.000, 0.000>

CIRA = TMP + <-5.000, 0.000, 0.000>

CIRB = TMP + <0.000, -5.000, 0.000>

CIRC = TMP + <0.000, 5.000, 0.000>

3. 프로그램 예

```

L000  INT I                      // 정수형 변수 I를 선언합니다.
L001  PNT TMP,RECTA,RECTB        // 위치형 변수 TMP, RECTA, RECTB를 선언합니다.
L002  PNT RECTC,RECTD            // 위치형 변수 RECTC, RECTD를 선언합니다.
L003  PNT CIRA,CIRB, CIRC        // 위치형 변수 CIRA, CIRB, CIRC를 선언합니다.
L004  SPD=10000                  // 속도를 3000[rpm]으로 지정합니다.
L005  ARCH=5                     // 위치 이동 전 Z축의 위치를 원점 기준 5mm로 설정.

```

```

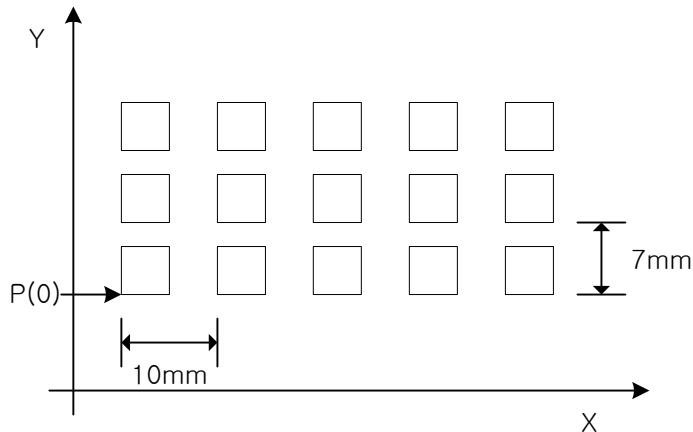
L006  TAG DASA                // DASA Tag 를 설정합니다.
L007  TAG START               // START Tag를 설정합니다.
L008  I=1                     // 정수형 변수 I를 1로 초기화 합니다.
L009  STBY B(0)==0XC8         // 입력이 “1100 1000”이 될 때까지 대기
L010  PCNT(0)=(1,1,1)         // 팔레트 0번의 Number를 초기화 합니다.
L011  PWRK(0)=(5,4,1)         // 팔레트 0번의 작업 형태를 설정 합니다.
L012  TAG WSTA                // WSTA Tag를 설정합니다.
L013  MPLT<0,P0,0>            // Work ID 0번의 초기 위치로 이동합니다.
L014  FINE 5                  // 이동 후 목표 위치의 펄스 오차가 5 Pulse가 될 때
L015  TMP=PLTP                // 현재 위치를 위치형 변수 TMP로 저장합니다.
L016  RECTA = TMP + <-5.0,-5.0,0.0,0.0>    // RECTA 위치를 계산합니다.
L017  RECTB = TMP + <5.0,-5.0,0.0,0.0>     // RECTB 위치를 계산합니다.
L018  RECTC = TMP + <5.0,5.0,0.0,0.0>      // RECTC 위치를 계산합니다.
L019  RECTD = TMP + <-5.0,5.0,0.0,0.0>     // RECTD 위치를 계산합니다.
L020  CIRA = TMP + <-5.0,0.0,0.0,0.0>      // CIRA 위치를 계산합니다.
L021  CIRB = TMP + <0.0,-5.0,0.0,0.0>      // CIRB 위치를 계산합니다.
L022  CIRC = TMP + <0.0,5.0,0.0,0.0>      // CIRC 위치를 계산합니다.
L023  PASS %10                // 중첩 이동을 목표점 대비 10%로 설정합니다.
L024  MPTP RECTA              // 계산된 RECTA 위치로 PTP 이동합니다.
L025  MLIN RECTB              // 계산된 RECTB 위치로 직선 보간 이동합니다.
L026  MLIN RECTC              // 계산된 RECTC 위치로 직선 보간 이동합니다.
L027  MLIN RECTD              // 계산된 RECTD 위치로 직선 보간 이동합니다.
L028  MPTP CIRA               // 계산된 CIRA 위치로 PTP 이동합니다.
L029  MCIR CIRB CIRC          // CIRB와 CIRC를 경유하는 원 보간 이동합니다.
L030  PASS OFF                // 중첩 이동을 해제합니다.
L031  I=I+1                   // 정수형 변수 I에 1을 증가시킵니다.
L032  IF I<=20 THEN           // 정수형 변수 I가 20보다 적거나 같을 때
L033  GOTO WSTA                // WSTA 이름을 가진 Tag 위치로 이동합니다.
L034  ELSE                    // 그렇지 않으면
L035  GOTO WSTOP               // WSTOP 이름을 가진 Tag 위치로 이동합니다.
L036  ENDI                     // IF문을 종료합니다.
L037  TAG WSTOP                // WSTOP Tag를 설정합니다.
L038  B(3)=0X35                // 출력 포트 B(3)을 “0011 0101”로 출력합니다.
L039  GOTO START               // START 이름을 가진 Tag 위치로 이동합니다.
L040  GOTO DASA                // DASA 이름을 가진 Tag 위치로 이동합니다.
L041  MEND                     // 주 프로그램을 종료합니다.

```

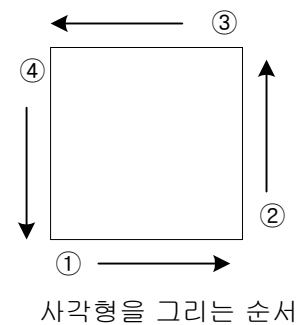
5-8 한 포인트로 여러 크기의 정사각형 그림을 그리는 프로그램

프로그램 설명

- 1) 여러 개의 정사각형은 작업에 따라 크기가 변한다.
- 2) 정사각형의 종류는 잘 변하므로 Point Teaching은 하나만 한다.
- 3) 예제 프로그램은 가로 x 세로 5mm의 정사각형을 그린다.
- 4) 위치 간격 데이터는 P10에 저장한다고 가정
(아래에서의 예 : P10=<10.000, 7.000>)



정사각형 하나의 크기는 가로,세로 모두 5mm라 가정



프로그램 작성 Tip

- 1) 작업에 따라 크기가 변한다 : 위치형 변수 사용
- 2) Point Teaching은 하나만 한다. : 편차 이동 사용

프로그램 예

```

L000  INT I           // 정수형 변수 I를 선언합니다.
L001  PNT XA, XB, XC  // 위치형 변수 XA, XB, XC를 선언합니다.
L002  PNT XD, XE      // 위치형 변수 XD, XE를 선언합니다.
L003  I=1             // 정수형 변수 I를 1로 초기화 합니다.
L004  XE=D10          // 편차값을 XE 변수에 치환합니다.
L005  SPD=7000        // 속도를 2100[rpm]으로 지정합니다.
L006  TAG RECT        // RECT Tag를 설정합니다.
L007  IF I<=15        // 15개의 정사각형을 그리므로 15가 될 때까지 I 변수를 판단
L008  GOTO DRAW       // 정수형 변수 I가 15보다 작거나 같으면 DRAW Tag로 이동
L009  ELSE            // 그렇지 않으면
L010  GOTO DREND      // DREND Tag로 이동합니다.
L011  ENDI            // IF문을 종료합니다.
L012  TAG DRAW        // DRAW Tag를 설정합니다.

```

L013 MPTP XE // 첫번째 사각형의 초기점으로 PTP 이동합니다.
L014 XA=XE+<10.0,0.0,0.0,0.0> // XA 좌표를 설정합니다.
L015 MLIN XA // XA 좌표로 직선 보간 이동합니다.(그림에서의 ①번 구간)
L016 XB=XA+<0.0,7.0,0.0,0.0> // XB 좌표를 설정합니다.
L017 MLIN XB // XB 좌표로 직선 보간 이동합니다.(그림에서의 ②번 구간)
L018 XC=XB+<-10.0,0.0,0.0,0.0> // XC 좌표를 설정합니다.
L019 MLIN XC // XC 좌표로 직선 보간 이동합니다.(그림에서의 ③번 구간)
L020 XD=XC+<0.0,-7.0,0.0,0.0> // XD 좌표를 설정합니다.
L021 MLIN XD // XD 좌표로 직선 보간 이동합니다.(그림에서의 ④번 구간)
L022 XE=XE+XE // 첫번째 좌표에 두번째 정사각형의 편차값을 더합니다.
L023 I=I+1 // 정사각형 1개를 그리고 정수형 변수 I를 1 증가시킵니다.
L024 GOTO RECT // RECT Tag로 이동합니다.
L025 TAG DREND // DREND Tag를 설정합니다.
L026 STOP PGM // 프로그램을 완전 종료합니다.
L027 MEND // 주 프로그램 끝을 선언합니다.

부가 설명

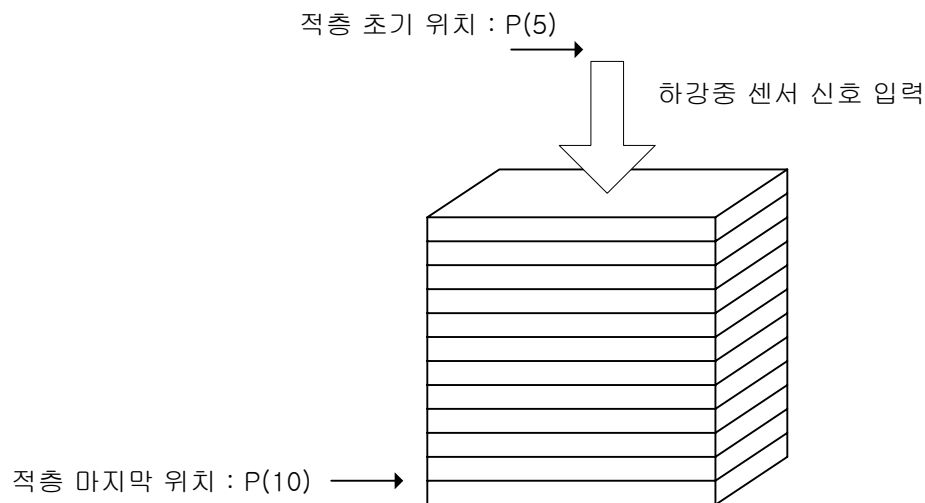
좌표 값을 더하는 것은 위치형 변수를 직접 더하여 사용할 수 있습니다. 예를 들어 각각의 편차가 각각의 포인트 번지에 P0=<10.000,0.000>, P1=<0.000,7.000>로 저장되어 있을 경우 위 프로그램은 다음과 같이 변경될 수 있습니다.

L015 XA=XE+<10.0,0.0,0.0,0.0> → XA=XE+P0
L017 XB=XA+<0.0,7.0,0.0,0.0> → XB=XA+P1
L019 XC=XB+<-10.0,0.0,0.0,0.0> → XC=XB-P0
L021 XD=XC+<0.0,-7.0,0.0,0.0> → XD=XC-P1

5-9 적층된 물품을 감지하여 옮기는 프로그램

프로그램 설명

- 1) 적층된 물품을 센서로 감지하여 다른 위치로 옮긴다.
- 2) 하나씩 옮길 때마다 적층 높이는 달라진다.
- 3) 하강 중 센서 신호가 입력되면 하강 정지한 후 물품을 흡착한다.
- 4) 센서 신호의 입력은 B(0).0번, 흡착 On 요청 신호는 출력 B(2).0번, 흡착 Off 요청 신호는 출력 B(2).1번, 흡착 On 신호는 입력 B(0).1번, 흡착 Off 신호는 입력 B(0).2번으로 한다.
- 5) 다른 위치는 P1의 위치로 한다.



프로그램 작성 Tip

- 1) 적층 높이는 달라진다. : 이동 중 조건 판단 실행(SYNC)
- 2) 신호 입출력 : B(0).0=1 명령 사용

프로그램 예

```

L000  SPD=5000          // 속도를 1500[rpm]으로 지정합니다.
L001  TAG DASA          // DASA Tag를 설정합니다.
L002  MPTP P5           // 적층 초기점으로 이동합니다.
L003  ARCH=0           // Arch 모션을 해제합니다.
L005  SYNC              // 이동 중 설정 조건에 따른 조건 판단을 시작합니다.
L006  MPTP P10          // P10의 위치로 PTP 이동합니다.
L008  ACT B(0).0==1     // P5의 위치에서 P10의 위치로 이동 중 센서 신호가 입력되면
L009  GOTO ABC          // ABC Tag로 이동합니다.
L012  ENDS              // SYNC문을 종료합니다.
L013  B(2).0=1          // 흡착 On 요청 신호를 출력합니다.
L014  STBY B(0).2==1    // 흡착 Off 될 때 까지 대기합니다.
L015  B(2).1=0          // 흡착 Off 요청 신호를 해제합니다.

```

L016	ARCH=5	// 위치 이동 전 Z축의 위치를 원점 기준 5mm로 설정합니다.
L017	GOTO DASA	// DASA Tag로 이동합니다.
L018	TAG ABC	// ABC Tag를 설정합니다.
L019	STOP MOVE	// 이동을 정지합니다.
L021	ARCH=5	// 위치 이동 전 Z축의 위치를 원점 기준 5mm로 설정합니다.
L022	B(2).0=1	// 흡착 On 요청 신호를 출력합니다.
L023	STBY B(0).1==1	// 흡착 On 될 때 까지 대기합니다.
L024	B(2).0=0	// 흡착 On 요청 신호를 해제합니다.
L025	MPTP P1	// P1 위치로 ARCH 모션 형태로 PTP 이동합니다.
L026	B(2).1=1	// 흡착 Off 요청 신호를 출력합니다.
L027	STBY B(0).2==1	// 흡착 Off 될 때 까지 대기합니다.
L028	B(2).1=0	// 흡착 Off 요청 신호를 해제합니다.
L029	GOTO DASA	// DASA Tag로 이동합니다.
L030	MEND	// 주 프로그램을 종료합니다.

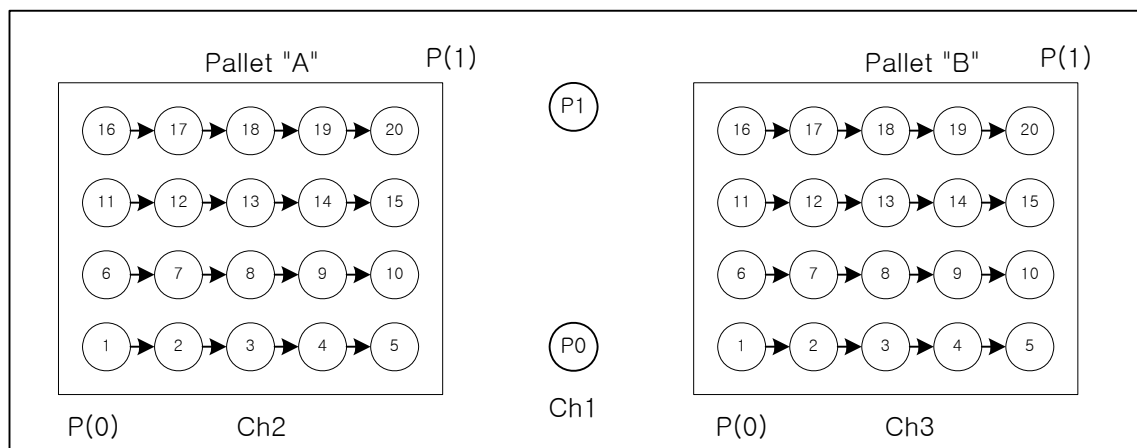
5-10 다 채널간 협조 운전이 필요한 프로그램

기구부 및 설정 파라미터 설명

- 1) X축 1개와 YZ축을 2 Set 가지는 5축 구성 기구부.
- 2) 파라미터 설정(Channel Setting)
 - Ch1 : X, Ch2 : XZ, Ch3 : XZ
 - Ch1은 X 축 만으로 구성하고 Ch2,3은 YZ 각 1 Set로 구성한다
 - Ch2,Ch3은 ARCH 동작을 위해서 XZ 로 설정한다

프로그램 설명

- 1) Pallet "A"는 YZ1 Set 기구부로 작업하고 Pallet "B"는 YZ2 Set 기구부로 작업한다.
- 2) 먼저 Ch1(X 축)이 작업 대기 위치에 도달하면 Ch2(YZ1), Ch3(YZ2)에 전달되는 도달완료 접점을 Set 한다.
- 3) Ch2(YZ1), Ch3(YZ2) 모두 1 라인의 작업이 완료 되면 Ch1(X)에 전달되는 1라인 작업완료 접점을 Set 한다
- 4) Ch1(X)은 Ch2(YZ1), Ch3(YZ2) 의 작업완료 신호를 확인 후 다시 다음 작업 위치로 이동한다. 이후 설정한 개수의 작업 완료 시 까지 2)~4)의 동작을 반복한다
- 5) 채널간 동기를 맞추기 위하여 작업완료 신호의 Set은 해당 채널에서 하고 Clear는 확인한 채널에서 한다
- 6) 각 Pallet의 시작점(P0)과 대각선 점(P1)의 두 점만 Teaching 하고 이동 거리는 내부에서 연산해서 사용한다
- 7) 작업물의 개수는 전역 정수변수를 사용해서 변경이 가능하게 한다



사용 내부 변수

- 1) B(150) : Ch2(YZ1)에 전달하는 Ch1(X) 1 Line 이동 완료 신호로 사용
Ch1(X)축 이동 완료 시 0XF0로 SET 하고 Ch2(YZ1)에서 0XF0을 확인하면 0X00로 Clear 시킴
- 2) B(151) : Ch3(YZ2)에 전달하는 Ch1(X) 1 Line 이동 완료 신호로 사용
Ch1(X)축 이동 완료 시 0XF0로 SET 하고 Ch3(YZ2)에서 0XF0을 확인하면 0X00로 Clear 시킴
- 3) B(160) : Ch1(X)에 Ch2(YZ1) 1 Line 이동 완료 신호로 사용
Ch2(YZ1) 이동 완료 시 0XF0로 SET 하고 Ch1(X)에서 0XF0을 확인하면 0X00로 Clear 시킴
- 4) B(161) : Ch1(X)에 Ch3(YZ2) 1 Line 이동 완료 신호로 사용
Ch3(YZ2) 이동 완료 시 0XF0로 SET 하고 Ch1(X)에서 0XF0을 확인하면 0X00로 Clear 시킴
- 5) B(170) : Ch1(X)에 Ch2(YZ1) 전체 작업 완료 신호로 사용
- 6) B(171) : Ch1(X)에 Ch3(YZ2) 전체 작업 완료 신호로 사용
- 7) GINT(206) : X축 이송 방향 개수
- 8) GINT(207) : Y축 이송 방향 개수

프로그램 예(Ch1)

```

L000  INT I1           // 정수형 변수 I1을 선언 합니다.
L001  PNT DISTX        // 실수형 변수 DISTX를 선언 합니다.
L002  SPD=5000         // 속도를 50%으로 지정합니다.
L003  GFLT(210)=P0.1   // 시작 포인트
L004  GFLT(211)=P1.1   // 마지막 포인트
L005  GFLT(212)=(GFLT(211)-GFLT(210))/GINT(206) // Ch1(X) 1회 이동량
L006  DISTX.1=P0.1     // Ch1(X) 지령위치(매회 GFLT(212)를 더함)
L007  B(150)=0X00      // 이동완료 내부 접점 Clear(Ch2(YZ1)에서 사용)
L008  B(151)=0X00      // 이동완료 내부 접점 Clear(Ch3(YZ2)에서 사용)
L009  TAG DASA         // DASA Tag를 설정합니다.
L010  STBY B(0).0==1   // 작업 개시 신호 대기
L011  MPTP P0          // Ch1(X)이 먼저 작업 대기점으로 이동 합니다.
L012  B(150)=0XF0      // 이동완료 내부 접점 Set(Ch2(YZ1)에서 사용)
L013  B(151)=0XF0      // 이동완료 내부 접점 Set(Ch3(YZ2)에서 사용)
L014  FOR I1=2 TO GINT(206) // X 축 이송 개수만큼 반복 이동
L015  STBY (B(160)==0XF0 && B(170)==0XF0) // YZ1,YZ2 1Line 작업완료 대기
L016  B(160)=0X00      // Ch2(YZ1) 1Line 작업완료 내부 접점 Clear
L017  B(170)=0X00      // Ch3(YZ2) 1Line 작업완료 내부 접점 Clear
L018  DISTX.1=DISTX.1+GFLT(212) // 다음 X축 작업 위치계산
    
```

```

L019  MPTP DISTX
L020  B(150)=0XF0    // 이동완료 내부 접점 Set(Ch2(YZ1)에서 사용)
L021  B(151)=0XF0    // 이동완료 내부 접점 Set(Ch3(YZ2)에서 사용)
L022  ENDF
L023  I1=0
L024  STBY (B(161)==0XF0 && B(171)==0XF0) // Ch2,3 전체 작업완료 대기
L025  B(161)=0X00    // Ch2(YZ1) 전체 작업완료 내부접점 Clear
L026  B(171)=0X00    // Ch3(YZ2) 전체 작업완료 내부접점 Clear
L027  MPTP P0        // 최초 작업 대기점 이동(1 팔레트 작업 완료)
L028  GOTO DASA

```

프로그램 예(Ch2)

```

L000  INT J1,J2, NSUMA    // 정수형 변수를 선언 합니다.
L001  PNT ZPOS, DISTA    // 실수형 변수를 선언 합니다.
L002  GFLT(220)=P0.1      // 시작 포인트
L003  GFLT(221)=P1.1      // 마지막 포인트
L004  GFLT(222)=(GFLT(221)-GFLT(220))/GINT(207) // Y축 1회 이동량
L005  DISTA.1=P0.1        // Y1 지령위치(매회 GFLT(222)를 더함)
L006  DISTA.2=P1.2        // Z1 지령위치
L007  SPD=5000            // 50% 속도 설정
L008  NSUMA=0
L009  B(160)=0X00         // Ch2 1Line 이동완료 내부 접점 Clear
L010  B(161)=0X00         // Ch2 전체 이동완료 내부 접점 Clear
L011  TAG DASA
L012  LOOP NSUMA<GINT(206)// Ch1(X 축) 이송 개수
L013  STBY B(150)==0XF0    // Ch1(X) 이송 완료 대기
L014  B(150)=0X00         // X축 이송 완료 Flag Clear
L015  FOR J1=1 TO GINT(207) // Y 축 이송 개수
L016  MPTP DISTA          // Ch2(YZ1) 작업위치로 이동
L017  ZPOS=DISTA
L018  DISTA.1=DISTA.1+GFLT(222) // 다음번 작업위치 계산
L019  ENDF
L020  ZPOS.2=30.0          // 1Line Y축 이송이 끝나면 Z축 UP
L021  MPTP ZPOS           // 1Line Y축 이송이 끝나면 Z축 UP
L022  NSUMA=NSUMA+1
L023  IF NSUMA==GINT(206) //X 축 이송 개수이면 작업 완료로 이동
L024  GOTO PF1
L025  ENDI

```

```

L026  DISTA.1=DISTA.1-GFLT(222)    // 같은자리 작업을 위해 한번 뺌
L027  B(160)=0XF0                  // Ch2(YZ1) 1Line 이송완료 Flag SET
L028  STBY B(150)==0XF0            // Ch1(X축) 이송 완료 대기
L029  B(150)=0X00                  // Ch1(X축) 이송 완료 Flag Clear
L030  FOR J2=1 TO GINT(207)        // 반대 방향 작업 시작
L031  MPTP DISTA
L032  ZPOS=DISTA
L033  DISTA.1=DISTA.1-GFLT(222)
L034  ENDF
L035  ZPOS.2=30.0                  // 1Line Y축 이송이 끝나면 Z축 UP
L036  MPTP ZPOS                    // 1Line Y축 이송이 끝나면 Z축 UP
L037  NSUMA=NSUMA+1
L038  IF NSUMA==GINT(206)          // X 축 이송 개수이면 작업 완료로 이동
L039  GOTO PF1
L040  ENDI
L041  DISTA.1=DISTA.1+GFLT(222)    //같은자리 작업을 위해 한번 더함
L042  B(160)=0XF0                  // Ch2(YZ1) 1Line 이송완료 Flag SET
L043  ENDL
L044  TAG PF1
L045  B(161)=0XF0                  // Ch2(YZ1) 전체 작업 완료 Flag SET
L046  DISTA.1=P0.1
L047  DISTA.2=P1.2
L048  MPTP DISTA                    //작업 대기점으로 이동
L049  B(160)=0X00
L050  NSUMA=0
L051  J1=1
L052  J2=1
L053  GOTO DASA

```

프로그램 예(Ch3)

```

L000  INT K1,K2, NSUMB             // 정수형 변수를 선언 합니다.
L001  PNT ZPOS, DISTB              // 실수형 변수를 선언 합니다.
L002  GFLT(230)=P0.1               // 시작 포인트
L003  GFLT(231)=P1.1               // 마지막 포인트
L004  GFLT(232)=(GFLT(231)-GFLT(230))/GINT(207) // Y축 1회 이동량
L005  DISTB.1=P0.1                 // Y2 지령위치(매회 GFLT(232)를 더함)
L006  DISTB.2=P1.2                 // Z2 지령위치
L007  SPD=5000                     // 50% 속도 설정

```

```

L008  NSUMB=0
L009  B(170)=0X00          // Ch3 1Line 이동완료 내부 점점 Clear
L010  B(171)=0X00          // Ch3 전체 이동완료 내부 점점 Clear
L011  TAG DASA
L012  LOOP NSUMB<GINT(206)// Ch1(X 축) 이송 개수
L013  STBY B(151)==0XF0    // Ch1(X) 이송 완료 대기
L014  B(151)=0X00          // X축 이송 완료 Flag Clear
L015  FOR K1=1 TO GINT(207) // Y 축 이송 개수
L016  MPTP DISTB           // Ch3(YZ2) 작업위치로 이동
L017  ZPOS=DISTB
L018  DISTB.1=DISTB.1+GFLT(232) // 다음번 작업위치 계산
L019  ENDF
L020  ZPOS.2=30.0          // 1Line Y축 이송이 끝나면 Z축 UP
L021  MPTP ZPOS            // 1Line Y축 이송이 끝나면 Z축 UP
L022  NSUMB=NSUMB+1
L023  IF NSUMB==GINT(206)  //X 축 이송 개수이면 작업 완료로 이동
L024  GOTO PF1
L025  ENDI
L026  DISTB.1=DISTB.1-GFLT(232) // 같은자리 작업을 위해 한번 뺌
L027  B(170)=0XF0          // Ch3(YZ2) 1Line 이송완료 Flag SET
L028  STBY B(151)==0XF0    // Ch1(X축) 이송 완료 대기
L029  B(151)=0X00          // Ch1(X축) 이송 완료 Flag Clear
L030  FOR K2=1 TO GINT(207) // 반대 방향 작업 시작
L031  MPTP DISTB
L032  ZPOS=DISTB
L033  DISTB.1=DISTB.1-GFLT(232)
L034  ENDF
L035  ZPOS.2=30.0          // 1Line Y축 이송이 끝나면 Z축 UP
L036  MPTP ZPOS            // 1Line Y축 이송이 끝나면 Z축 UP
L037  NSUMB=NSUMB+1
L038  IF NSUMA==GINT(206)  // X 축 이송 개수이면 작업 완료로 이동
L039  GOTO PF1
L040  ENDI
L041  DISTB.1=DISTB.1+GFLT(232) //같은자리 작업을 위해 한번 더함
L042  B(170)=0XF0          // Ch3(YZ2) 1Line 이송완료 Flag SET
L043  ENDL
L044  TAG PF1
L045  B(171)=0XF0          // Ch3(YZ2) 전체 작업 완료 Flag SET

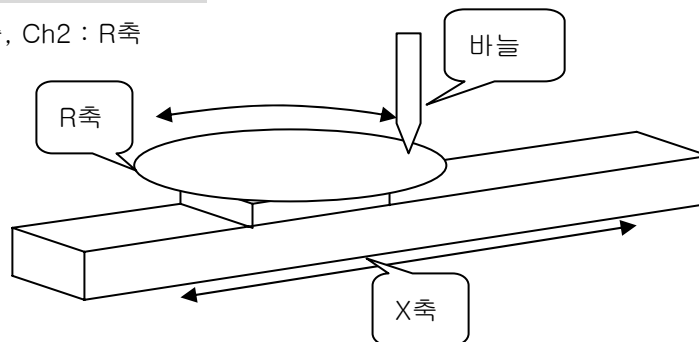
```

L046 DISTB.1=P0.1
 L047 DISTB.2=P1.2
 L048 MPTP DISTB //작업 대기점으로 이동
 L049 B(170)=0X00
 L050 NSUMB=0
 L051 K1=1
 L052 K2=1
 L053 GOTO DASA

5-11 다른 채널의 X축 위치에 따른 실시간 속도 변경 프로그램(고급사용자용)

기구부 및 설정 파라미터 설명

1) Ch1 : X축, Ch2 : R축

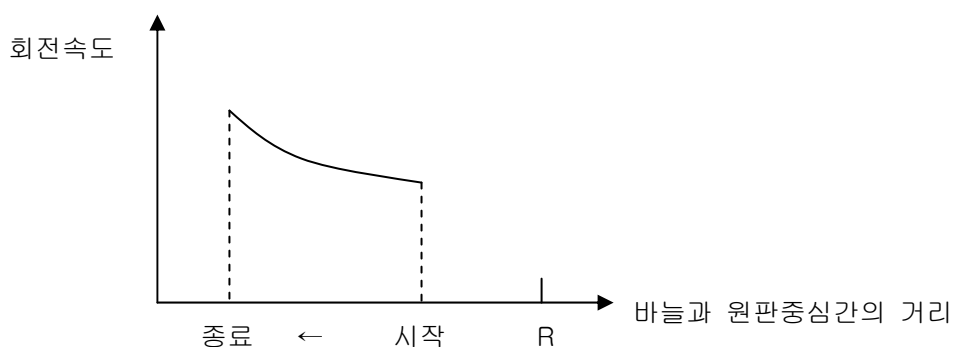


프로그램 설명

1) X축이 이동하면서 R축과 바늘끝이 만나는 곳의 선속도가 일정하도록 제어합니다. 바늘이 회전 중심에 가까워질수록 원판의 속도가 증가해야 합니다.

$$v(\text{바늘이 원판과 만나는 점에서의 선속도}) = R(\text{원판의 반지름}) \times \omega(\text{원판의 회전속도})$$

위의 수식에 따라 원판의 중심에 바늘이 가까워짐에 따라 속도는 반비례하여 증가해야 합니다. 이러한 작업을 통하여 일정한 밀도로 원판에 액체나 빛을 분사할 수 있습니다.



- 2) X축은 대기위치와 속도제어 시작위치, 속도제어 종료위치등을 설정합니다. 전역실수를 사용하여 채널간에 공유할 수 있도록 합니다.
- 3) 대기위치에서 작업시작 접점을 받으면 R축이 먼저 설정속도로 warming-up 한 후, X축이 이동합니다.

- 4) 작업이 종료되면 R축은 멈추고, X축은 다시 대기위치로 이동합니다.

프로그램 작성 Tip

- 1) SPDR 명령어를 사용하여 이동 중 속도를 변경합니다.
- 2) SPDR 명령어의 인자의 범위는 0부터 1까지이므로 처음에는 0.5등의 작은 값에서 시작하여 중심에 가까워질수록 차차 증가시킵니다. 최대 속도에 도달했을 때의 값이 1보다 작도록 조정해야 합니다.
- 3) 채널 1 프로그램에서 TMP변수를 사용하지 않고 GFLT(210)에 직접 값을 대입값을 대입한 후 이 값을 1.0과 비교하여 1보다 큰 경우 1로 제한하도록 되어 있는 경우, 1로 제한되기 전에 채널 2의 SPDR명령어가 실행되면 Runtime Error가 발생하므로 임시변수를 사용합니다. SPDR명령어의 인자값이 1보다 크면 Runtime Error가 발생합니다.

프로그램 예(Ch1)

```

L000  PNT PE,PHOME
L001  FLT R                      // 원판의 반지름
L002  FLT TMP                    // 채널간 공용변수는 미리 계산후 최종적으로 저장하기위함.
L003  SPD=GINT(200)
L004  PHOME.1=GFLT(200)
L005  PE.1=GFLT(202)+50.0
L006  R=200.0
L007  LOOP 1
L008  GFLT(210)=0.5
L009  MPTP PHOME                // Home위치로 이동하여 대기합니다.
L010  STBY B(0).0==1
L011  B(10).0=1                 // Ch2-R축이 Warmin-up하도록 합니다.
L012  B(10).1=0
L013  STBY B(10).1==1          // R축의 Warming-up이 완료되기까지 기다립니다.
L014  B(10).0=0
L014  SYNC
L015  MPTP PE
L016  ACT GPNT(252).1>GFLT(201)&&GPNT(252).1<GFLT(202)
L017  TMP=R/(R-(GPNT(252).1-GFLT(201)))*0.5
L018  IF TMP>1.0
L019  TMP=1.0
L020  ENDI
L021  GFLT(210)=TMP
L022  ENDS
L023  B(10).3=0

```

```
L024 B(10).2=1
L025 STBY B(10).3==1
L026 ENDL
L027 MEND
```

프로그램 예(Ch2)

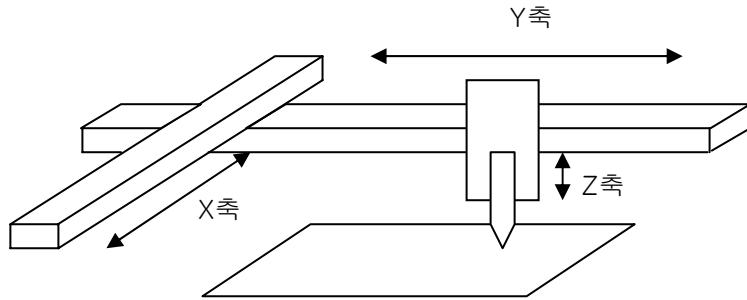
```
L000 SPD=2000 // 300RPM으로 회전
L001 LOOP 1
L002 SPDR=0.5
L003 PSET <0> // 현재 위치를 0으로 reset
L004 STBY B(10).0==1
L005 SYNC
L006 MPTP <10000.0> // 작업이 끝나기에 충분히 긴 거리로 설정
L007 ACT GPNT(253).1>200.0
L008 B(10).1=1
L009 ACT 1
L010 SPDR=GFLT(210)
L011 ACT B(10).2==1
L012 STOP MOVE
L013 B(10).3=1
L013 B(10).2=0
L013 GOTO EXIT
L014 ENDS
L015 TAG EXIT
L016 ENDL
L017 MEND
```

5 - 1 2 레이저 센서를 이용한 높이 보정 프로그램(고급사용자용)

레이저 센서는 거리에 따라 전압을 출력하여 거리를 알 수 있도록 하는 장치이다. 이 센서를 사용하여 Dispensing시 평탄도와 같은 에러를 보상하는 프로그램을 작성할 수 있는데, Dispensing 대상의 높낮이의 변화가 심한 경우 2채널(XY-X)로 구성하여 실시간으로 보상하도록 하는 방법이 있고, 그렇지 않고 대상이 일정한 평면으로 다만 약간 기울어져 있는 경우에는 XYZ로 구성할 수도 있다. 여기에서는 XYZ로 구성한 프로그램을 예로 든다.

기구부 및 설정 파라미터 설명

- 1) 기구부 종류 : XYZ, Analog Option 보드 장착



프로그램 설명

1) Calibration작업을 통해서 '레이저 센서의 출력'과 '바늘끝과 유리면과의 거리'와의 관계를 알아 낸다.

Calibration block에서 레이저 센서의 출력이 0일때의 Z축 위치를 100이라고 하고, 바늘이 버튼을 눌러 신호가 들어올때의 Z축 위치를 120이라고 한다면, $120 - 100 = 20$ 만큼의 거리를 더 이동하면 바늘끝이 유리면에 닿는 다는 것을 알 수 있다. 더 정확히는 버튼의 신호가 들어오는 위치와 레이저 센서의 빛이 반사되는 면의 위치의 차이를 X라고 하면 $(20 + X)$ 가 된다. 예를 들어 버튼이 Calibration Block의 면에서 10만큼 들어갈 때 신호가 들어온다고 하면 30이라는 결과가 나온다. 또한 calibration block에서 레이저 센서의 출력이 0일때 레이저 센서와 calibration block와의 거리와 유리면에서 레이저 센서의 출력이 0일때 레이저 센서와 유리면의 거리가 반사등에 의해서 다를 수 있으므로 이도 고려해주어야 한다. 결론적으로 레이저 센서의 출력값을 알면 그 때의 위치에서 얼마를 더 하강하면 바늘끝이 물체에 닿는지를 계산해 낼 수 있다. 높이를 계산하는 새로운 식은 다음과 같다.

$$GFLT(100+I) = (GFLT(209) - GFLT(228)) + GFLT(236) + GFLT(110+I)$$

$GFLT(209)$ = 바늘이 버튼을 눌러 신호가 들어올 때의 Z축의 높이

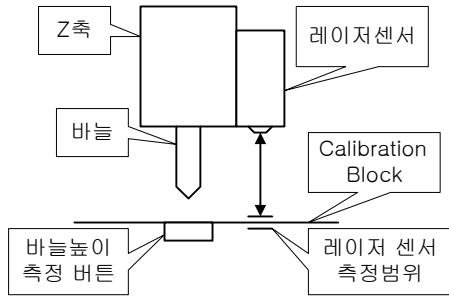
$GFLT(228)$ = 레이저 센서의 출력이 0일 때의 Z축의 높이

$GFLT(236)$ = 버튼을 눌러 신호가 들어올 때의 버튼 면의 높이와 레이저 센서의 빛이 반사되는 면 사이의 높이와 유리면의 반사특성 및 Dispensing시 유리와 바늘의 거리를 고려한 **종합적인 보상량**

$GFLT(110+I)$ = SCAN I위치에서 레이저 센서의 출력이 0일 때의 높이

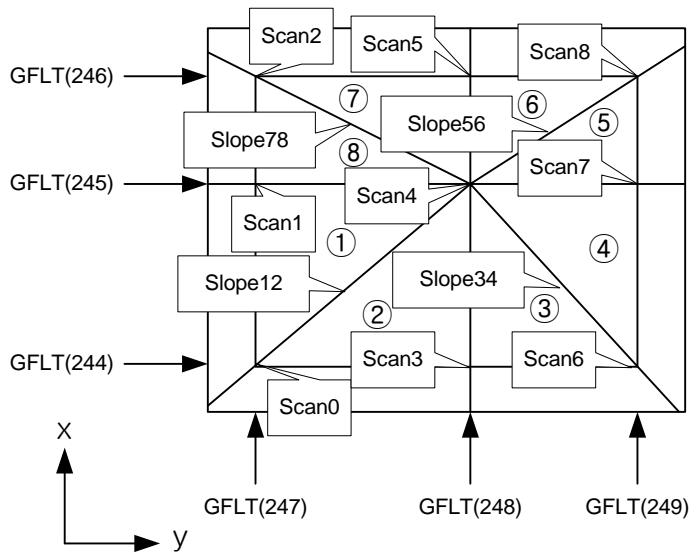
$GFLT(100+I)$ = SCAN I위치에서의 Dispensing 작업시 Z축 높이(스캔이 끝난후 모션프로그램을 정지하여 확인해볼 필요가 있다. 이 값을 확인한 후 각 Scan포인트에서 Z축을 그 위치로 실제 이동시켜 봐서 값이 제대로 나왔는지 확인하는 것이 가능하다.)

실제 작업시 높이가 잘 맞지 않는다면 $GFLT(236)$ 의 값을 조정함으로써 높이를 변화시키면 된다. 프로그램을 실행하기 전에 $GFLT(236)$ 의 값이 지나치게 큰 값이 들어있지는 않은지 확인해야 한다. 우선 처음에는 0으로 하고 나중에 미세조정하는 것이 좋은 것 같다. Calibration후 $GFLT(209)$ 의 값과 $GFLT(228)$ 의 값을 확인하고 Scan후 $GFLT(110)$ 의 값을 확인하여 $GFLT(209) - GFLT(228) + GFLT(110)$ 의 값을 구하고 Scan0위치에서 Z축을 이 위치로 이동한 후 그 위치에서 Dispensing작업을 하는데 모자라는 거리만큼을 $GFLT(236)$ 에 넣어서 사용하면 된다.



2) Dispensing 대상물의 구배(기울기)를 구한다.

기울기를 계산하기 위해서 사각형의 9점의 높이를 구한 후, 9점이 이루는 8개의 사각형에 대해서 평면의 방정식을 구하여 평면내의 임의의 점에서의 높이를 계산한다.



위의 그림과 같이 가로, 세로 3개의 선을 정한 후에 가로/세로 선의 교차점 9개의 위치에서의 Dispensing시 바늘의 높이를 레이저 센서를 사용하여 Scan한다.

9점을 스캔하면 위와 같이 점을 연결하여 8개의 삼각형을 만들 수 있고, 삼각형을 나타내는 평면의 방정식을 구할 수 있다. 9점에 의해서 8개의 영역이 만들어 지는데 그러면 임의의 점이 주어질 때 그 점이 어떤 영역에 속하는지를 결정하여 높이를 계산해낼 수 있다. 단 어떤 영역에 속하는지 계산을 편하게 하기 위해서 점들이 가로, 세로로 일직선 상에 위치해야 하는 제약조건이 있다.

예를 들어,

SCAN0,1,4로 이루어지는 삼각형을 생각하면,

SCAN0=<281.367, 19.79, 64>=P0

SCAN1=<488.617, 19.79, 63>=P1

SCAN4=<488.617, 289.791, 62>=P2

위의 세 점으로 이루어지는 평면의 방정식을 구하기 위해서, 우선 평면에 수직인 벡터를 구한다.

삼각형의 두 변의 벡터를 외적하면 되므로,

$\text{cross}(P1-P0, P2-P0) = \text{Normal vector} = \langle 270.000, 207.25, 55957.707 \rangle$

예상대로 Z축 값이 크게 나옴을 볼 수 있다.

평면의 방정식인 $ax+by+cz+d=0$ 에서 $\langle a,b,c \rangle$ 가 normal vector이므로

$a=270, b=207.25, c=5595.771$ 이고 d 를 구하기 위해서는 평면의 한 점을 대입하면 된다.

P0를 대입하면 $d=-(ax+by+cz)=-3661364.113$

삼각형의 무게중심에서의 Z축의 높이를 구해보면, 무게중심 계산 공식에 의해 $(P0+P1+P2)/3=\langle 419.534, 109.79, 63 \rangle$ 으로 Z축값이 63이고, 위에서 구한 평면의 방정식을 이용하여 구한 Z축의 높이도 63이 나옴을 알 수 있다.

프로그램 작성 Tip

- 1) Calibration시에는 SYNC 문을 사용하여 이동중에 버튼 감지와 레이저 센서값 판단을 수행합니다. 정확성을 높이기 위해서는 최대한 낮은 속도로 이동합니다.
- 2) 공통된 연산에는 서브 프로그램(SPGM)을 만들어 사용합니다.

프로그램 예

(* 본 프로그램은 field에서 실제 사용된 것으로 가감없이 게재합니다.)

```

1  INT MDLNO
2  INT INIT // 한번만 실행하기 위한 코드를 위해 사용. 0일 때 한번 계산하고 1로 만든 후 0이 아니면 계산안함
3  PNT RECTA,RECTB,RECTC,RECTD,RECTS // S:사각형에서 토출시작점.A,B,C,D:모서리
4  PNT RECTE,RECTE2,RECTS2 // E:바늘 상승,E2:토출정지,S2:하강점
5  PNT RECTE3,RECTE4 // E3:정지위치 이동후 전진,E4:전진후 후진위치
6  PNT PTMP // 계산값 임시 저장용
7  FLT ZDNMAX // 유리에 부딪치는 것을 막기 위해 Z축의 이동범위를 제한
8  INT CALOK // Calibration이 정상적일 때 1로 함
9  INT LSENOK//LASER SENSING OK IN CALIBRATION
10 INT I,J // 사각형 그릴 때 카운터로 사용
11 INT ITMP // 임시변수
12 PNT HTIN // 9점 스캔후 높이 계산시 입력값
13 FLT HTOUT // 9점 스캔후 높이 계산시 결과값
14 FLT SLOPE12,SLOPE34,SLOPE56,SLOPE78 //LINE SLOPE
15 FLT A1,B1,C1,D1 // 삼각형1 평면의 방정식 계수
16 FLT A2,B2,C2,D2 // 삼각형2 평면의 방정식 계수
17 FLT A3,B3,C3,D3 // 삼각형3 평면의 방정식 계수
18 FLT A4,B4,C4,D4 // 삼각형4 평면의 방정식 계수
19 FLT A5,B5,C5,D5 // 삼각형5 평면의 방정식 계수
20 FLT A6,B6,C6,D6 // 삼각형6 평면의 방정식 계수
21 FLT A7,B7,C7,D7 // 삼각형7 평면의 방정식 계수
22 FLT A8,B8,C8,D8 // 삼각형8 평면의 방정식 계수
23 PNT CR1,CR2,CR3 // 외적을 구하는데 사용
24 PNT CR12,CR13,N // 외적을 구하는데 사용
25 FLT D // 평면의 방정식에서 d값 구하는데 사용
26 FLT SURGAPX,SURGAPY //dummy 사각형 그릴 때 사용
27 FLT SURLENX,SURLENY //LENGTH
28 FLT SURORGX,SURORGY //ORG
29 FLT SURARCL // ARC의 반지름 길이
30 PNT SURS,SURE // dummy 사각형 그릴 때 각 선분의 시작과 끝점 위치
31 PNT SURS2,SURE2 // dummy 사각형 그릴 때 토출 시작과 정지 위치
32 INT FILTCNT // 필터 카운터
33 FLT ARCSPPDBX,ARCSPPDBY // ARC그릴때 속도 변경을 위한 변수
34 FLT ARCSPPDCX,ARCSPPDCY
35 FLT ARCSPPDDX,ARCSPPDDY
36 FLT ARCSPPDAX,ARCSPPDAY
37 PNT LEFTLOW,RIGHTHI // 작업영역의 범위 제한
38 FLT LC //SCAN시 레이저 위치 보정용
39 INT DUMN,DUMC//DUMMY LINE용
40 FLT LCZI,LCZIX,LCZIY //LASER SCAN CENTER Z INTERPOLATION VALUE
41 FLT X,Y//TEMP
42 ZDNMAX=91 // Z축 이동 제한 값
43 INIT=1 // 초기에 한번만 계산할 변수값들을 위해 사용됨. 한 번 계산후 0으로 하여 다시 계산하지 않도록 함
44 DECR 200 // 감속시간을 200ms로 함
45 ACCR 200 // 가속시간을 200ms로 함
46 GINT(210)=1//X1 CNT, 소그룹 x 카운터
47 GINT(211)=1//X2 CNT, 대그룹 x 카운터
48 GINT(212)=1//Y1 CNT, 소그룹 y 카운터
49 GINT(213)=1//Y2 CNT, 대그룹 y 카운터
50 IF GINT(200)<1
51 GINT(200)=1
52 ENDI

```

```

53 IF GINT(201)<1
54 GINT(201)=1
55 ENDI
56 IF GINT(202)<1
57 GINT(202)=1
58 ENDI
59 IF GINT(203)<1
60 GINT(203)=1
61 ENDI
62 IF GINT(231)<1 // ARCH
63 GINT(231)=1
64 ENDI
65 SPD GINT(220) // MLIN을 위한 기본 속도를 적용함
66 MDLNO=B(0)&0xF // 모델 번호를 읽어옴
67 B(6).7=0 //CLEAR USER ALARM
68 SLOPE12=(GFLT(248)-GFLT(247))/(GFLT(245)-GFLT(244))
69 SLOPE34=(GFLT(248)-GFLT(249))/(GFLT(245)-GFLT(244))
70 SLOPE56=(GFLT(249)-GFLT(248))/(GFLT(246)-GFLT(245))
71 SLOPE78=(GFLT(247)-GFLT(248))/(GFLT(246)-GFLT(245))
72 LOOP 1 //INFINITE WORK LOOP
73 IF B(4).5==1 // 테스트용
74 B(4).5=0
75 CALL LSCAN // 레이저 9점 스캔을 시작함
76 ENDI
77 IF B(3).0==1 // 테스트용
78 B(3).0=0
79 CALL CAL // calibration을 수행함
80 ENDI
81 IF B(4).0==1 // 테스트용
82 B(4).0=0
83 CALL ONERECT // 한 개의 사각형(바깥쪽과 안쪽)을 그림
84 ENDI
85 IF B(130).1==1//ALARM RESET
86 B(6).7=0//USER ALARM RESET
87 ENDI
88 IF B(3).5==1|B(0).5==1 //START CMD
89 B(3).5=0
90 IF B(0).4==1 //DISPENSER OK
91 IF GINT(210)==1&&GINT(211)==1&&GINT(212)==1&&GINT(213)==1 //1 PLATE START AND FIRST
RECTANGLE, 카운트가 모두 1이면 첫번째 사각형임
92 IF INIT==1
93 INIT=0 // 한 번만 수행해야 할 작업을 마치면 0으로 함
94 CALL CAL // calibration을 수행함. 한 번만 수행
95 CALL CALCVAR // dummy 테두리를 그리기 위한 변수값 계산. 한번만 수행
96 CALL CHKBDRY// dummy선이 작업영역크기(P33,P34)를 벗어나는지 검사
97 ENDI
98 IF B(6).7==0&&B(0).5==1 //에러가 아니고, start command가 들어오면
99 CALL LSCAN // 레이저 스캔
100 CALL PURGE // 퍼지
101 CALL CLEAN // 닦기
102 ENDI
103 IF B(0).7==1&&B(6).7==0&&B(0).5==1//DRAW SURROUNDING RECT CMD IF NO ERROR
104 CALL SURRECT // dummy선을 그림
105 ENDI
106 ENDI//FIRST RECTANGLE
107 IF B(6).7==0&&B(0).5==1 //NO ERROR, CALIBRATION IS OK
108 B(6).4=1 //1 GRID START SIGNAL ON
109 CALL ONERECT // 사각형을 한 개 그림
110 B(6).4=0 //1 GRID START SIGNAL OFF
111 IF GINT(210)==1&&GINT(211)==1&&GINT(212)==1&&GINT(213)==1 //1 PLATE COMPLETE
112 B(6).5=1 //1 PLATE COMPLETE OUTPUT SIGNAL ON
113 CALL HOME // 홈위치로 이동함
114 B(6).5=0 //1 PLATE COMPLETE OUTPUT SIGNAL OFF
115 ENDI //1 PLATE COMPLETE
116 ENDI //NO ERROR
117 ENDI //DISPENSER OK

```

```

118 ENDI //START CMD
119 IF B(130).3==1 // 홈위치 이동명령이 들어오면
120 B(130).3=0
121 CALL HOME // 홈위치로 이동
122 ENDI
123 IF B(130).4==1 || B(0).6==1 //RESTART명령이 들어오면, B(130).4는 사양에서 제외되었나?
124 B(130).4=0
125 GINT(210)=1 // 모든 카운터를 0으로 함
126 GINT(211)=1
127 GINT(212)=1
128 GINT(213)=1
129 B(3).5=1 // 사각형 그리기 명령을 줌. 한 개의 사각형을 그림
130 ENDI
131 ENDL // 전체 작업 무한 루프
132 MEND // 메인 프로그램 종료
133 SPGM HOME //GO TO THE HOME POSITION
134 ARCH 1 // Z축을 최대한 들어올려서 홈위치로 이동함
135 SPD 2000 // 홈위치 이동시 속도를 정함
136 MPTP P20 //GO TO THE HOME POSITION
137 SEND
138 SPGM CHKBDRY//CHECK BOUNDARY LIMIT
139 LEFTLOW.1=P(MDLNO).1-(GFLT(200)-GFLT(204)) // dummy사각형위치를 계산함
140 LEFTLOW.2=P(MDLNO).2-(GFLT(201)-GFLT(205))
141 RIGHTHI.1=P(MDLNO).1+ GINT(200)*GFLT(200)+( GINT(202)-1)*GFLT(202)
142 RIGHTHI.2=P(MDLNO).2+ GINT(201)*GFLT(201)+( GINT(203)-1)*GFLT(203)
143 IF LEFTLOW.1<P33.1 || LEFTLOW.2<P33.2 || RIGHTHI.1>P34.1 || RIGHTHI.2>P34.2 // P33,34를 벗어나면
예러
144 B(6).7=1//USER ALARM
145 ENDI
146 SEND
147 SPGM CLEAN // 바늘을 닦음
148 SPD GINT(223) // MPTP 기본 속도를 사용함.
149 ARCH 60
150 PTMP=P22
151 PTMP.3=60//HIGH PARALLEL, 하강시작높이
152 MPTP PTMP//하강시작위치로 이동
153 SPD 400//하강시 속도 지정
154 ARCH OFF
155 PTMP=P22
156 PTMP.3=GFLT(212)//DOWN
157 MPTP PTMP //CLEAN POSITION, 하강
158 PTMP.3=60 //UP
159 MPTP PTMP // 상승
160 SEND
161 SPGM PURGE // 퍼지
162 SPD GINT(223) // MPTP 기본 속도를 사용
163 ARCH 60
164 PTMP=P32
165 PTMP.3=60//HIGH PARALLEL
166 MPTP PTMP // 퍼지 하강시 시작 위치
167 ARCH OFF
168 PTMP=P32
169 PTMP.3=GFLT(212)//DOWN
170 MPTP PTMP //PURGE POSITION, 하강
171 PTMP.2=PTMP.2+ GFLT(213)//LOW PARALLEL, 하강후 옆으로 이동위치 계산
172 B(7).7=1//DISPENSER ON
173 SPD 200 // 옆으로 이동시 속도 지정
174 MPTP PTMP // 옆으로 이동
175 B(7).7=0//DISPENSER OFF
176 PTMP.3=60 // 상승 위치
177 SPD GINT(223) // MPTP 기본 속도 사용
178 MPTP PTMP // 상승
179 SEND
180 SPGM CAL //MECHANICAL CALIBRATION, 순서는 바늘과 버튼을 하고나서 레이저 센서를 한다
181 CALL PURGE // 먼저 바늘에 남아 있는 액을 쏜다.
182 CALL CLEAN // 바늘 끝을 닦는다.

```

```

183 ARCH 60 // MPTP 이동시 Z축 높이를 60으로 하여 이동한다.
184 SPD 2000 // 바늘 calibration 시작 위치로 이동시 속도 지정
185 PTMP=P21 // calibration x,y위치
186 PTMP.3=60 // calibration에서 하강시작z축위치를 60으로 한다.
187 MPTP PTMP // 하강시작위치로 이동한다.
188 ARCH OFF // ARCH동작없이 MPTP시 목표 위치로 Z축을 상승하지 않고 이동한다
189 PTMP=P21
190 PTMP.3=ZDNMAX // Z축의 최대이동위치를 제한한다
191 CALOK=0 // 아직 calibration이 완료되지 않았다
192 SPD 50 // 천천히 이동한다. 속도가 빠르면 calibration 오차가 커진다
193 DECR 1 // 오차를 줄이기 위해 버튼의 신호가 들어온 후 정지시 감속시간을 최소로 하여 빨리 정지한다
194 SYNC // 이동중 레이저의 출력을 검사한다
195 MPTP PTMP // 버튼을 누르기 위해 이동한다
196 ACT B(1).6==0 // 레이저 센서가 달려 있는지 검사한다
197 LSENOK=1 // 레이저 센서가 발견되었음을 기억한다
198 ACT B(1).7==1 // 버튼이 눌러졌음을 나타낸다
199 STOP MOVE // 버튼이 눌러지면 정지한다
200 WAIT 50 // 완전히 정지하기까지 약간 기다린다
201 GFLT(208)=GFLT(209) // 이전 calibration값을 보관한다. 디버깅시 유용
202 ITMP=GPNT(252).3*100+ 0.5 // 새 calibration값을 계산한다. GPNT(252)에는 현재좌표값이 저장되어 있다.
소수점 2째자리에서 반올림한다.
203 GFLT(209)=ITMP*0.01 // 새 calibration값을 저장한다
204 CALOK=1 // calibration이 정상적으로 진행되었음을 기억한다
205 GOTO CALEXIT // 바늘 calibration을 빠져 나간다
206 ENDS
207 TAG CALEXIT
208 DECR 200 //감속시간을 원래대로 한다.
209 IF CALOK==1&&LSENOK==1 // 바늘calibration시 에러가 없고 레이저 센서가 발견되었으면 레이저
calibration을 시작한다.
210 ARCH OFF
211 PTMP=P21
212 PTMP.3=GFLT(222)// z축 하강시 시작위치
213 SPD 10 // 버튼에서와 마찬가지로 천천히 이동한다
214 DECR 1 // 오차를 줄이기 위해 감속시간을 최소로 한다
215 SYNC
216 MPTP PTMP
217 ACT B(1).6==0&&GINT(1)>=0 // 레이저 출력이 OK이고 출력값이 0보다 크면
218 STOP MOVE // 정지
219 WAIT 50 // 정지할때까지 기다림
220 GINT(228)=GINT(229)
221 GINT(229)=GINT(1) // 새 레이저 출력값
222 GFLT(229)=GFLT(228)
223 GFLT(228)=GPNT(252).3+ GINT(229)*2.0/4096 // 레이저 출력이 0일 때의 위치를 저장한다
224 GOTO CALEXIT2
225 ENDS
226 TAG CALEXIT2
227 DECR 200
228 ELSE //CALIBRATION FAILURE
229 B(6).7=1 //USER ALARM
230 ENDI
231 SEND
232 SPGM LSCAN //9 POINT LASER SCAN
233 ARCH OFF
234 SPD 2000 // 스캔 시의 속도 지정
235 SPDR 1.0 // 실시간 속도 변경값을 원래로 복귀. 이값은 arc에서 속도 변경기능에 의해 1이 아닌 값이 된다.
236 ITMP=GFLT(222) // 스캔 위치로 이동시 z축 상승값
237 ARCH ITMP
238 PTMP.1=GFLT(244)
239 PTMP.2=GFLT(247)
240 PTMP.3=GFLT(222)
241 MPTP PTMP
242 ARCH OFF
243 PTMP.3=80//DOWN MAX
244 SPD 100 // 천천히 하강
245 SYNC
246 MPTP PTMP // 하강

```

```

247 ACT B(1).6==0 //OK신호가 들어오면
248 STOP MOVE // 정지후
249 GOTO LSEXIT2 // 빠져나간다
250 ENDS
251 TAG LSEXIT2 // 빠져나와서
252 PTMP.3=GPNT(252).3-5 // 현재 위치에서 5mm 상승한다
253 MPTP PTMP // 상승
254 PTMP.3=PTMP.3+ 10 // 하강위치계산. 현재 위치에서 10만큼 더 하강. 레이저 센서는 2mm짜리 이므로
10mm면 충분
255 FILTCNT=0
256 SPD 10 // 더 천천히 이동한다
257 DECR 1
258 SYNC
259 MPTP PTMP // 하강
260 ACT B(1).6==0&&GINT(1)<=0 // OK신호가 들어오고 유리면과 가까워져서 출력값이 음수가 되면
261 FILTCNT=FILTCNT+ 1 //필터 사용
262 IF FILTCNT>10 // 10번이 넘으면
263 STOP MOVE // 정지
264 WAIT 50 // 레이저 센서의출력이 안정화 될때까지 기다림
265 GFLT(110)=GPNT(252).3+ GINT(1)*2.0/4096.0 // 레이저 센서의 출력이 0일때의 z축 위치 계산
266 GFLT(100)=(GFLT(209)-GFLT(228))+ GFLT(236)+ GFLT(110)
267 GOTO LSEXIT
268 ENDI
269 ENDS
270 TAG LSEXIT
271 SPD 2000
272 DECR 200
273 LC=0
274 FOR I=1 TO 8 // SCAN1~8까지 수행함
275 PTMP.1=GFLT(244+I%3)// SCAN 위치
276 PTMP.2=GFLT(247+I/3)
277 PTMP.3=GFLT(110)+ LC //LASER SCAN HEIGHT
278 TAG SR //SCAN RESTART
279 SYNC
280 MPTP PTMP
281 ACT 0//((GINT(1)>-2000&&GINT(1)<-1500)| |(GINT(1)>1500&&GINT(1)<2000))
282 STOP MOVE
283 WAIT 50
284 LC=LC+ GINT(1)*2.0/4096.0
285 PTMP.3=PTMP.3+ LC
286 MPTP PTMP
287 GOTO SR
288 ENDS
289 WAIT 50 //WAIT LASER OUTPUT OK
290 GFLT(110+I)=PTMP.3+ GINT(1)*2.0/4096.0
291 GFLT(100+I)=(GFLT(209)-GFLT(228))+ GFLT(236)+ GFLT(110+I)
292 ENDF
293 LCZIX=((GFLT(246)-GFLT(245))*GFLT(103)+ (GFLT(245)-GFLT(244))*GFLT(105))/(GFLT(246)-
GFLT(244))
294 LCZIY=((GFLT(249)-GFLT(248))*GFLT(101)+ (GFLT(248)-GFLT(247))*GFLT(107))/(GFLT(249)-
GFLT(247))
295 LCZI=(LCZIX+ LCZIY)*0.5
296 IF GFLT(104)-LCZI>0.2//ETCHING AREA COMPENSATION
297 GFLT(104)=LCZI
298 ENDI
299 //REGION 1, 각 영역의 평면의 방정식을 구한다.
300 CR1.1=GFLT(244)
301 CR1.2=GFLT(247)
302 CR2.1=GFLT(245)
303 CR2.2=GFLT(248)
304 CR3.1=GFLT(245)
305 CR3.2=GFLT(247)
306 CR1.3=GFLT(100)
307 CR2.3=GFLT(104)
308 CR3.3=GFLT(101)
309 CALL CROSS

```

```
310  A1=N.1
311  B1=N.2
312  C1=N.3
313  D1=D
314  //REGION 2
315  CR1.1=GFLT(244)
316  CR1.2=GFLT(247)
317  CR2.1=GFLT(244)
318  CR2.2=GFLT(248)
319  CR3.1=GFLT(245)
320  CR3.2=GFLT(248)
321  CR1.3=GFLT(100)
322  CR2.3=GFLT(103)
323  CR3.3=GFLT(104)
324  CALL CROSS
325  A2=N.1
326  B2=N.2
327  C2=N.3
328  D2=D
329  //REGION 3
330  CR1.1=GFLT(244)
331  CR1.2=GFLT(248)
332  CR2.1=GFLT(245)
333  CR2.2=GFLT(249)
334  CR3.1=GFLT(245)
335  CR3.2=GFLT(248)
336  CR1.3=GFLT(103)
337  CR2.3=GFLT(106)
338  CR3.3=GFLT(104)
339  CALL CROSS
340  A3=N.1
341  B3=N.2
342  C3=N.3
343  D3=D
344  //REGION 4
345  CR1.1=GFLT(244)
346  CR1.2=GFLT(249)
347  CR2.1=GFLT(245)
348  CR2.2=GFLT(249)
349  CR3.1=GFLT(245)
350  CR3.2=GFLT(248)
351  CR1.3=GFLT(106)
352  CR2.3=GFLT(107)
353  CR3.3=GFLT(104)
354  CALL CROSS
355  A4=N.1
356  B4=N.2
357  C4=N.3
358  D4=D
359  //REGION 5
360  CR1.1=GFLT(245)
361  CR1.2=GFLT(248)
362  CR2.1=GFLT(245)
363  CR2.2=GFLT(249)
364  CR3.1=GFLT(246)
365  CR3.2=GFLT(249)
366  CR1.3=GFLT(104)
367  CR2.3=GFLT(107)
368  CR3.3=GFLT(108)
369  CALL CROSS
370  A5=N.1
371  B5=N.2
372  C5=N.3
373  D5=D
374  //REGION 6
375  CR1.1=GFLT(245)
```

```

376 CR1.2=GFLT(248)
377 CR2.1=GFLT(246)
378 CR2.2=GFLT(249)
379 CR3.1=GFLT(246)
380 CR3.2=GFLT(248)
381 CR1.3=GFLT(104)
382 CR2.3=GFLT(108)
383 CR3.3=GFLT(105)
384 CALL CROSS
385 A6=N.1
386 B6=N.2
387 C6=N.3
388 D6=D
389 //REGION 7
390 CR1.1=GFLT(245)
391 CR1.2=GFLT(248)
392 CR2.1=GFLT(246)
393 CR2.2=GFLT(248)
394 CR3.1=GFLT(246)
395 CR3.2=GFLT(247)
396 CR1.3=GFLT(104)
397 CR2.3=GFLT(105)
398 CR3.3=GFLT(102)
399 CALL CROSS
400 A7=N.1
401 B7=N.2
402 C7=N.3
403 D7=D
404 //REGION 8
405 CR1.1=GFLT(245)
406 CR1.2=GFLT(247)
407 CR2.1=GFLT(245)
408 CR2.2=GFLT(248)
409 CR3.1=GFLT(246)
410 CR3.2=GFLT(247)
411 CR1.3=GFLT(101)
412 CR2.3=GFLT(104)
413 CR3.3=GFLT(102)
414 CALL CROSS
415 A8=N.1
416 B8=N.2
417 C8=N.3
418 D8=D
419 SEND
420 SPGM CROSS //CROSS PRODUCT
421 CR12=CR2-CR1
422 CR13=CR3-CR1
423 N.1=CR13.2*CR12.3-CR13.3*CR12.2
424 N.2=CR13.3*CR12.1-CR13.1*CR12.3
425 N.3=CR13.1*CR12.2-CR13.2*CR12.1
426 D=-N.1*CR1.1-N.2*CR1.2-N.3*CR1.3
427 SEND
428 SPGM CALCVAR
429 SURARCL=GINT(204)*0.1
430 SURLNX=GFLT(204)-SURARCL*2//2=BOTH CORNER
431 SURGAPX=GFLT(200)-GFLT(204)
432 SURORGX=P(MDLNO).1
433 SURLNY=GFLT(205)-SURARCL*2
434 SURGAPY=GFLT(201)-GFLT(205)
435 SURORGY=P(MDLNO).2
436 SEND
437 SPGM SRLINE // DRAW ONE LINE SECTION IN DUMMY LINE
438 CALL HTSUR
439 ARCH GINT(231)
440 SPD GINT(223)
441 SPDR GFLT(235)

```

```
442  MPTP SURS2
443  SPDR 1.0
444  SPD GINT(221)
445  B(7).7=1 //DISPENSER ON
446  MLIN SURE2
447  B(7).7=0 //DISPENSER OFF
448  SEND
449  SPGM SRRECT
450  HTIN=SURA
451  CALL HEIGHT1
452  SURA.3=HTOUT
453  HTIN=SURB
454  CALL HEIGHT1
455  SURB.3=HTOUT
456  HTIN=SURC
457  CALL HEIGHT1
458  SURC.3=HTOUT
459  HTIN=SURD
460  CALL HEIGHT1
461  SURD.3=HTOUT
462  ARCH GINT(231)
463  SPD GINT(223)
464  SPDR GFLT(235)
465  MPTP SURA
466  SPDR 1.0
467  SPD GINT(221)
468  B(7).7=1 //DISPENSER ON
469  MLIN SURB
470  MLIN SURC
471  MLIN SURD
472  MLIN SURA
473  B(7).7=0 //DISPENSER OFF
474  SEND
475  SPGM SURRECT //DRAW SURROUNDING RECTANGLE, dummy line을 그림
476  IF GINT(226)>1
477  DUMN=2
478  ELSE
479  DUMN=1
480  ENDI
481  DUMC=0
482  LOOP DUMC<DUMN
483  DUMC=DUMC+ 1
484  IF DUMC==1
485  SURGAPY=GFLT(238)
486  SURGAPX=GFLT(240)
487  ELSE
488  SURGAPY=GFLT(238)+ GFLT(239)
489  SURGAPX=GFLT(240)+ GFLT(241)
490  ENDI
491  IF GFLT(204)!=0.0&&GFLT(205)!=0.0 //DRAW SURROUNDING RECT
492  IF DUMC==2&&GFLT(239)!=0&&GFLT(241)!=0
493  SURA.1=SURORGX-GFLT(240)
494  SURA.2=SURORGY-GFLT(238)
495  SURB.1=SURA.1
496  SURB.2=SURA.2-GFLT(239)
497  SURC.1=SURA.1-GFLT(241)
498  SURC.2=SURA.2-GFLT(239)
499  SURD.1=SURA.1-GFLT(241)
500  SURD.2=SURA.2
501  CALL SRRECT
502  ENDI
503  FOR I=1 TO GINT(202)//X2
504  FOR J=1 TO GINT(200)//X1
505  SURS.1=SURORGX+ (J-1)*GFLT(200)+ (I-1)*GFLT(202)+ SURARCL
506  SURS.2=SURORGY-SURGAPY
507  SURE=SURS
```

```

508 SURE.1=SURE.1+ SURLENX
509 SURS2=SURS
510 SURS2.1=SURS2.1-GFLT(242)
511 SURE2=SURE
512 SURE2.1=SURE2.1+ GFLT(243)
513 CALL SRLINE
514 ENDF
515 ENDF
516 IF DUMC==2&&GFLT(239)!=0&&GFLT(241)!=0
517 SURA.1=SURORGX+ (GINT(200)-1)*GFLT(200)+ (GINT(202)-1)*GFLT(202)+ GFLT(204)+ GFLT(240)
518 SURA.2=SURORGY-GFLT(238)
519 SURB.1=SURA.1+ GFLT(241)
520 SURB.2=SURA.2
521 SURC.1=SURA.1+ GFLT(241)
522 SURC.2=SURA.2-GFLT(239)
523 SURD.1=SURA.1
524 SURD.2=SURA.2-GFLT(239)
525 CALL SRRECT
526 ENDI
527 FOR I=1 TO GINT(203)//Y2
528 FOR J=1 TO GINT(201)//Y1
529 SURS.1=SURORGX+ (GINT(200)-1)*GFLT(200)+ (GINT(202)-1)*GFLT(202)+ GFLT(204)+ SURGAPX
530 SURS.2=SURORGY+ (J-1)*GFLT(201)+ (I-1)*GFLT(203)+ SURARCL
531 SURE=SURS
532 SURE.2=SURE.2+ SURLENY
533 SURS2=SURS
534 SURS2.2=SURS2.2-GFLT(242)
535 SURE2=SURE
536 SURE2.2=SURE2.2+ GFLT(243)
537 CALL SRLINE
538 ENDF
539 ENDF
540 IF DUMC==2&&GFLT(239)!=0&&GFLT(241)!=0
541 SURA.1=SURORGX+ (GINT(200)-1)*GFLT(200)+ (GINT(202)-1)*GFLT(202)+ GFLT(204)+ GFLT(240)
542 SURA.2=SURORGY+ (GINT(201)-1)*GFLT(201)+ (GINT(203)-1)*GFLT(203)+ GFLT(205)+ GFLT(238)
543 SURB.1=SURA.1
544 SURB.2=SURA.2+ GFLT(239)
545 SURC.1=SURA.1+ GFLT(241)
546 SURC.2=SURA.2+ GFLT(239)
547 SURD.1=SURA.1+ GFLT(241)
548 SURD.2=SURA.2
549 CALL SRRECT
550 ENDI
551 FOR I=1 TO GINT(202)//X2
552 FOR J=1 TO GINT(200)//X1
553 SURE.1=SURORGX+ (GINT(200)-J)*GFLT(200)+ (GINT(202)-1)*GFLT(202)+ SURARCL
554 SURE.2=SURORGY+ (GINT(201)-1)*GFLT(201)+ (GINT(203)-1)*GFLT(203)+ GFLT(205)+ SURGAPY
555 SURS=SURE
556 SURS.1=SURS.1+ SURLENX
557 SURS2=SURS
558 SURS2.1=SURS2.1+ GFLT(242)
559 SURE2=SURE
560 SURE2.1=SURE2.1-GFLT(243)
561 CALL SRLINE
562 ENDF
563 ENDF
564 IF DUMC==2&&GFLT(239)!=0&&GFLT(241)!=0
565 SURA.1=SURORGX-GFLT(240)
566 SURA.2=SURORGY+ (GINT(201)-1)*GFLT(201)+ (GINT(203)-1)*GFLT(203)+ GFLT(205)+ GFLT(238)
567 SURB.1=SURA.1-GFLT(241)
568 SURB.2=SURA.2
569 SURC.1=SURA.1-GFLT(241)
570 SURC.2=SURA.2+ GFLT(239)
571 SURD.1=SURA.1
572 SURD.2=SURA.2+ GFLT(239)
573 CALL SRRECT

```

```
574   ENDI
575   FOR I=1 TO GINT(203)//Y2
576   FOR J=1 TO GINT(201)//Y1
577     SURE.1=SURORGX-SURGAPX
578     SURE.2=SURORGY+ (GINT(201)-J)*GFLT(201)+ (GINT(203)-I)*GFLT(203)+ SURARCL
579     SURS=SURE
580     SURS.2=SURE.2+ SURLENY
581     SURS2=SURS
582     SURS2.2=SURS2.2+ GFLT(242)
583     SURE2=SURE
584     SURE2.2=SURE2.2-GFLT(243)
585     CALL SRLINE
586   ENDF
587   ENDF
588   ENDI
589   ENDL
590   SEND
591   SPGM ONERECT //DRAW INNER AND OUTER RECTANGLE
592   IF GFLT(204)!=0.0&&GFLT(205)!=0.0 //DRAW OUTER RECTANGLE
593     IF GINT(212)&0X01==1
594       IF GINT(213)&0X01==1
595         RECTA.1=P(MDLNO).1+ (GINT(210)-1)*GFLT(200)+ (GINT(211)-1)*GFLT(202)
596       ELSE
597         RECTA.1=P(MDLNO).1+ (GINT(210)-1)*GFLT(200)+ (GINT(202)-GINT(211))*GFLT(202)
598       ENDI
599     ELSE
600       IF GINT(213)&0X01==1
601         RECTA.1=P(MDLNO).1+ (GINT(200)-GINT(210))*GFLT(200)+ (GINT(211)-1)*GFLT(202)
602       ELSE
603         RECTA.1=P(MDLNO).1+ (GINT(200)-GINT(210))*GFLT(200)+ (GINT(202)-GINT(211))*GFLT(202)
604       ENDI
605     ENDI
606     RECTA.2=P(MDLNO).2+ (GINT(212)-1)*GFLT(201)+ (GINT(213)-1)*GFLT(203)
607     RECTB=RECTA
608     RECTB.1=RECTA.1+ GFLT(204)
609     RECTC=RECTA
610     RECTC.1=RECTA.1+ GFLT(204)
611     RECTC.2=RECTA.2+ GFLT(205)
612     RECTD=RECTA
613     RECTD.2=RECTA.2+ GFLT(205)
614     RECTS=RECTA
615     IF GINT(210)==1&&GINT(211)==1&&GINT(212)==1&&GINT(213)==1
616       RECTS.1=RECTA.1+ GFLT(204)*0.5-GFLT(211)
617     RECTE=RECTA
618     RECTE.1=RECTA.1+ GFLT(204)*0.5+ GFLT(214)
619     RECTE2=RECTE
620     RECTE2.1=RECTE2.1-GFLT(216)
621     RECTS2=RECTS
622     RECTS2.1=RECTS2.1-GFLT(219)
623   ELSE
624     RECTS.1=RECTA.1+ GFLT(204)*0.5-GFLT(215)
625     RECTE=RECTA
626     RECTE.1=RECTA.1+ GFLT(204)*0.5+ GFLT(221)
627     RECTE2=RECTE
628     RECTE2.1=RECTE2.1-GFLT(217)
629     RECTS2=RECTS
630     RECTS2.1=RECTS2.1-GFLT(218)
631   ENDI
632   RECTE3=RECTE
633   RECTE3.1=RECTE3.1+ GFLT(224)
634   RECTE4=RECTE
635   RECTE4.1=RECTE4.1-GFLT(225)
636   CALL HEIGHT
637   RECTE3.3=RECTE3.3-GFLT(223)
638   RECTE4.3=RECTE4.3-GFLT(223)
639   ARCSPDBX=RECTB.1-GINT(204)*0.1-GFLT(234)
```

```

640  ARCSPDBY=RECTB.2+ GINT(204)*0.1+ GFLT(234)
641  ARCSPDCX=RECTC.1-GINT(204)*0.1-GFLT(234)
642  ARCSPDCY=RECTC.2-GINT(204)*0.1-GFLT(234)
643  ARCSPDDX=RECTD.1+ GINT(204)*0.1+ GFLT(234)
644  ARCSPDDY=RECTD.2-GINT(204)*0.1-GFLT(234)
645  ARCSPDAX=RECTA.1+ GINT(204)*0.1+ GFLT(234)
646  ARCSPDAY=RECTA.2+ GINT(204)*0.1+ GFLT(234)
647  ARCH GINT(231)
648  SPD GINT(223)
649  SPDR GFLT(235)
650  MPTP RECTS2
651  PASS GINT(204)
652  SPD GINT(220)
653  DECR 1
654  //MLIN RECTS
655  //WAIT GINT(224)
656  SYNC
657  MLIN RECTB
658  ACT GPNT(252).1>RECTS.1
659  B(7).7=1 //DISPENSER ON
660  ACT GPNT(252).1>ARCSPDBX
661  SPDR GFLT(233)
662  ENDS
663  SYNC
664  MLIN RECTC
665  ACT GPNT(252).2<ARCSPDBY
666  SPDR GFLT(233)
667  ACT GPNT(252).2>=ARCSPDBY&&GPNT(252).2<=ARCSPDCY
668  SPDR GFLT(235)
669  ACT GPNT(252).2>ARCSPDCY
670  SPDR GFLT(233)
671  ENDS
672  SYNC
673  MLIN RECTD
674  ACT GPNT(252).1>ARCSPDCX
675  SPDR GFLT(233)
676  ACT GPNT(252).1<=ARCSPDCX&&GPNT(252).1>=ARCSPDDX
677  SPDR GFLT(235)
678  ACT GPNT(252).1<ARCSPDDX
679  SPDR GFLT(233)
680  ENDS
681  SYNC
682  MLIN RECTA
683  ACT GPNT(252).2>ARCSPDDY
684  SPDR GFLT(233)
685  ACT GPNT(252).2<=ARCSPDDY&&GPNT(252).2>=ARCSPDAY
686  SPDR GFLT(235)
687  ACT GPNT(252).2<ARCSPDAY
688  SPDR GFLT(233)
689  ENDS
690  //MLIN RECTE2
691  //B(7).7=0 //DISPENSER OFF
692  //MLIN RECTE
693  PASS 10
694  SYNC
695  MLIN RECTE3
696  ACT B(7).7==1&&GPNT(252).1>RECTE.1
697  B(7).7=0 //DISPENSER OFF
698  ACT GPNT(252).1>ARCSPDAX
699  SPDR GFLT(235)
700  ENDS
701  PTMP=RECTE4
702  PTMP.3=PTMP.3-GFLT(226)
703  DECR 200
704  MLIN PTMP
705  PTMP=GPNT(252)

```

```
706 PTMP.3=PTMP.3-GFLT(210)
707 SPDR 1.0
708 ARCH OFF
709 MPTP PTMP //Z UP
710 ITMP=GINT(223)*0.5
711 SPD ITMP
712 ENDI
713 X=GFLT(204)-2*GFLT(206)
714 Y=GFLT(205)-2*GFLT(207)
715 IF GFLT(206)!=0.0&&GFLT(207)!=0.0&&X>0&&Y>0 //DRAW RECT2(INNER)
716 IF GINT(212)&0X01==1
717 IF GINT(213)&0X01==1
718 RECTA.1=P(MDLNO).1+ (GINT(210)-1)*GFLT(200)+ (GINT(211)-1)*GFLT(202)+ GFLT(206)
719 ELSE
720 RECTA.1=P(MDLNO).1+ (GINT(210)-1)*GFLT(200)+ (GINT(202)-GINT(211))*GFLT(202)+ GFLT(206)
721 ENDI
722 ELSE
723 IF GINT(213)&0X01==1
724 RECTA.1=P(MDLNO).1+ (GINT(200)-GINT(210))*GFLT(200)+ (GINT(211)-1)*GFLT(202)+ GFLT(206)
725 ELSE
726 RECTA.1=P(MDLNO).1+ (GINT(200)-GINT(210))*GFLT(200)+ (GINT(202)-
GINT(211))*GFLT(202)+ GFLT(206)
727 ENDI
728 ENDI
729 RECTA.2=P(MDLNO).2+ (GINT(212)-1)*GFLT(201)+ (GINT(213)-1)*GFLT(203)+ GFLT(207)
730 RECTB=RECTA
731 RECTB.1=RECTA.1+ X
732 RECTC=RECTA
733 RECTC.1=RECTA.1+ X
734 RECTC.2=RECTA.2+ Y
735 RECTD=RECTA
736 RECTD.2=RECTA.2+ Y
737 RECTS=RECTA
738 RECTS.1=RECTA.1+ X*0.5-GFLT(215)
739 RECTE=RECTA
740 RECTE.1=RECTA.1+ X*0.5+ GFLT(221)
741 RECTE2=RECTE
742 RECTE2.1=RECTE2.1-GFLT(217)
743 RECTS2=RECTS
744 RECTS2.1=RECTS2.1-GFLT(218)
745 RECTE3=RECTE
746 RECTE3.1=RECTE3.1+ GFLT(224)
747 RECTE4=RECTE
748 RECTE4.1=RECTE4.1-GFLT(225)
749 CALL HEIGHT
750 RECTE3.3=RECTE3.3-GFLT(223)
751 RECTE4.3=RECTE4.3-GFLT(223)
752 ARCSPDBX=RECTB.1-GINT(205)*0.1-GFLT(234)
753 ARCSPDBY=RECTB.2+ GINT(205)*0.1+ GFLT(234)
754 ARCSPDCX=RECTC.1-GINT(205)*0.1-GFLT(234)
755 ARCSPDCY=RECTC.2-GINT(205)*0.1-GFLT(234)
756 ARCSPDDX=RECTD.1+ GINT(205)*0.1+ GFLT(234)
757 ARCSPDDY=RECTD.2-GINT(205)*0.1-GFLT(234)
758 ARCSPDAX=RECTA.1+ GINT(205)*0.1+ GFLT(234)
759 ARCSPDAY=RECTA.2+ GINT(205)*0.1+ GFLT(234)
760 ARCH GINT(231)
761 ITMP=GINT(223)*0.5
762 SPD ITMP
763 SPDR GFLT(235)
764 MPTP RECTS2
765 PASS GINT(205)
766 SPD GINT(220)
767 DECR 1
768 //MLIN RECTS
769 //WAIT GINT(224)
770 SYNC
```

```

771 MLIN RECTB
772 ACT GPNT(252).1>RECTS.1
773 B(7).7=1 //DISPENSER ON
774 ACT GPNT(252).1>ARCSPEBX
775 SPDR GFLT(233)
776 ENDS
777 SYNC
778 MLIN RECTC
779 ACT GPNT(252).2<ARCSPEBY
780 SPDR GFLT(233)
781 ACT GPNT(252).2>=ARCSPEBY&&GPNT(252).2<=ARCSPECY
782 SPDR GFLT(235)
783 ACT GPNT(252).2>ARCSPECY
784 SPDR GFLT(233)
785 ENDS
786 SYNC
787 MLIN RECTD
788 ACT GPNT(252).1>ARCSPECX
789 SPDR GFLT(233)
790 ACT GPNT(252).1<=ARCSPECX&&GPNT(252).1>=ARCSPEDX
791 SPDR GFLT(235)
792 ACT GPNT(252).1<ARCSPEDX
793 SPDR GFLT(233)
794 ENDS
795 SYNC
796 MLIN RECTA
797 ACT GPNT(252).2>ARCSPEDY
798 SPDR GFLT(233)
799 ACT GPNT(252).2<=ARCSPEDY&&GPNT(252).2>=ARCSPEAY
800 SPDR GFLT(235)
801 ACT GPNT(252).2<ARCSPEAY
802 SPDR GFLT(233)
803 ENDS
804 //MLIN RECTE2
805 //B(7).7=0 //DISPENSER OFF
806 //MLIN RECTE
807 PASS 10
808 SYNC
809 MLIN RECTE3
810 ACT B(7).7==1&&GPNT(252).1>RECTE.1
811 B(7).7=0 //DISPENSER OFF
812 ACT GPNT(252).1>ARCSPEAX
813 SPDR GFLT(235)
814 ENDS
815 PTMP=RECTE4
816 PTMP.3=PTMP.3-GFLT(226)
817 DECR 200
818 MLIN PTMP
819 PTMP=GPNT(252)
820 PTMP.3=PTMP.3-GFLT(210)
821 SPDR 1.0
822 ARCH OFF
823 ITMP=GINT(223)*0.5
824 SPD ITMP
825 MPTP PTMP
826 ENDI
827 IF GINT(210)<GINT(200) //X1,CALCULATE NEXT WORK CNT
828 GINT(210)=GINT(210)+ 1
829 ELSE
830 GINT(210)=1 //RESET X1 CNT
831 IF GINT(212)<GINT(201) //Y1
832 GINT(212)=GINT(212)+ 1
833 ELSE
834 GINT(212)=1 //RESET Y1 CNT
835 IF GINT(211)<GINT(202) //X2
836 GINT(211)=GINT(211)+ 1

```

```
837 ELSE
838 GINT(211)=1 //RESET X2 CNT
839 IF GINT(213)<GINT(203) //Y2
840 GINT(213)=GINT(213)+ 1
841 ELSE
842 GINT(213)=1 //RESET Y2 CNT
843 ENDI
844 ENDI
845 ENDI
846 ENDI
847 TAG OREXIT
848 SEND
849 SPGM HTSUR
850 HTIN=SURS
851 CALL HEIGHT1
852 SURS.3=HTOUT
853 HTIN=SURE
854 CALL HEIGHT1
855 SURE.3=HTOUT
856 HTIN=SURS2
857 CALL HEIGHT1
858 SURS2.3=HTOUT
859 HTIN=SURE2
860 CALL HEIGHT1
861 SURE2.3=HTOUT
862 SEND
863 SPGM HEIGHT //CALCULATE HEIGHT FROM LASER SCAN RESULT
864 HTIN=RECTS2
865 CALL HEIGHT1
866 RECTS2.3=HTOUT
867 HTIN=RECTS
868 CALL HEIGHT1
869 RECTS.3=HTOUT
870 HTIN=RECTA
871 CALL HEIGHT1
872 RECTA.3=HTOUT
873 HTIN=RECTB
874 CALL HEIGHT1
875 RECTB.3=HTOUT
876 HTIN=RECTC
877 CALL HEIGHT1
878 RECTC.3=HTOUT
879 HTIN=RECTD
880 CALL HEIGHT1
881 RECTD.3=HTOUT
882 HTIN=RECTE2
883 CALL HEIGHT1
884 RECTE2.3=HTOUT
885 HTIN=RECTE
886 CALL HEIGHT1
887 RECTE.3=HTOUT
888 HTIN=RECTE3
889 CALL HEIGHT1
890 RECTE3.3=HTOUT
891 HTIN=RECTE4
892 CALL HEIGHT1
893 RECTE4.3=HTOUT
894 SEND
895 SPGM HEIGHT1
896 IF HTIN.1<GFLT(245) //X CENTER
897 IF HTIN.2<GFLT(248) //Y CENTER
898 IF HTIN.2<SLOPE12*(HTIN.1-GFLT(244))+ GFLT(247) //REGION 1
899 HTOUT=-(D1+ A1*HTIN.1+ B1*HTIN.2)/C1
900 ELSE //REGION 2
901 HTOUT=-(D2+ A2*HTIN.1+ B2*HTIN.2)/C2
902 ENDI
```

```
903 ELSE//LEFT-UPPER
904 IF HTIN.2<SLOPE34*(HTIN.1-GFLT(244))+ GFLT(249) //REGION 3
905 HTOUT=-(D3+ A3*HTIN.1+ B3*HTIN.2)/C3
906 ELSE //REGION 4
907 HTOUT=-(D4+ A4*HTIN.1+ B4*HTIN.2)/C4
908 ENDI
909 ENDI
910 ELSE
911 IF HTIN.2<GFLT(248) //Y CENTER
912 IF HTIN.2<SLOPE56*(HTIN.1-GFLT(246))+ GFLT(249) //REGION 6
913 HTOUT=-(D6+ A6*HTIN.1+ B6*HTIN.2)/C6
914 ELSE //REGION 5
915 HTOUT=-(D5+ A5*HTIN.1+ B5*HTIN.2)/C5
916 ENDI
917 ELSE//RIGHT-UPPER
918 IF HTIN.2<SLOPE78*(HTIN.1-GFLT(246))+ GFLT(247) //REGION 8
919 HTOUT=-(D8+ A8*HTIN.1+ B8*HTIN.2)/C8
920 ELSE //REGION 7
921 HTOUT=-(D7+ A7*HTIN.1+ B7*HTIN.2)/C7
922 ENDI
923 ENDI
924 ENDI
925 SEND
```