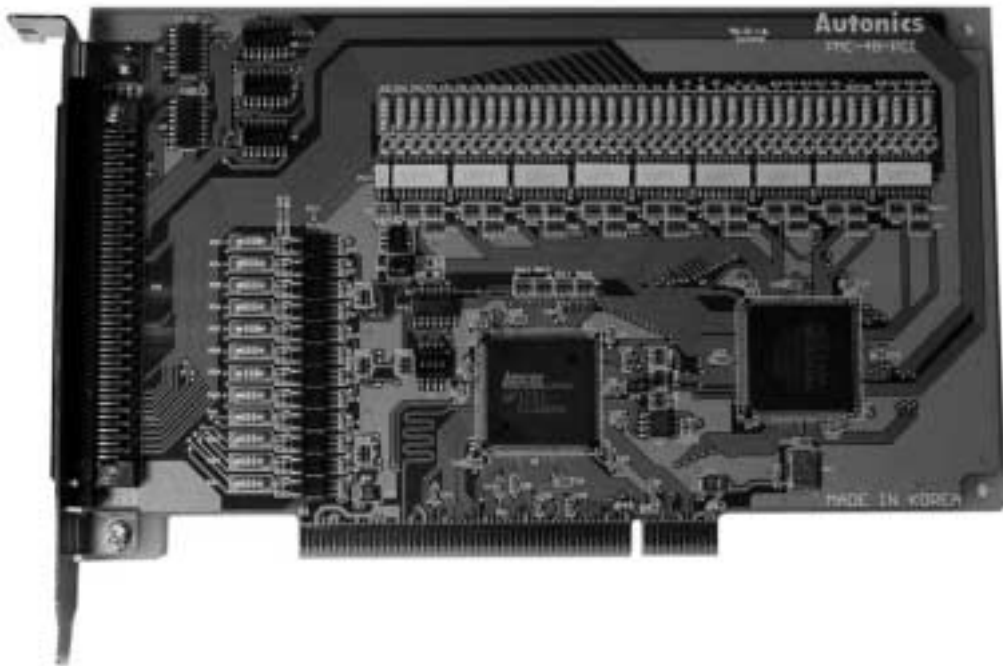


USER MANUAL

PROGRAMMABLE MOTION CONTROLLER
4-AXIS PC-BASED MOTION CONTROLLER

PMC-4B-PCI



〈PMC-4B-PCI 주요 특징〉

- 4축 독립 AC Servo Motor 및 Step Motor 제어 가능
- PC-PCI Type Card
- 자동 원점 복귀 실행 및 동기동작 기동
- 원호/직선/Bit Pattern/연속/가감속 Drive에서 보간 가능
- 2축/3축 선속 일정
- Windows 98, NT, 2000, XP 호환 가능
- C++ 환경에서 동작 가능한 라이브러리 제공

Autonics

SENSOR & CONTROLLER

저희 Autonics의 프로그램 모션 컨트롤러(PMC) 제품을 선택하여 주셔서 정말 감사합니다.
 제품을 사용하기 전에 본 취급설명서를 충분히 읽으신 후 신호전압, 신호 타이밍, 동작 Parameter 값 등,
 기술된 사양 범위 안에서 올바르게 사용하시기를 부탁드립니다.

본 자료의 기재된 내용은 기술 진보 및 기능향상 등에 의해 예고 없이 변경하는 경우가 있습니다.
 최신 자료는 당사의 홈 페이지(<http://www.autonics.com>)로 부터 제공 받으실 수 있습니다.

■ 안전을 위한 주의사항

※'안전을 위한 주의사항'은 제품을 안전하고 올바르게 사용하여 사고나 위험을 미리 막기 위한 것이므로 반드시 지켜 주십시오.

※주의사항은 '경고'와 '주의'의 두가지로 구분되어 있으며 '경고'와 '주의'의 의미는 다음과 같습니다.

⚠ 경고 지시사항을 위반하였을 때, 심각한 상해나 사망이 발생할 가능성이 있는 경우

⚠ 주의 지시사항을 위반하였을 때, 경미한 상해나 제품 손상이 발생할 가능성이 있는 경우

※제품과 사용설명서에 표시된 그림기호의 의미는 다음과 같습니다.

⚠는 특정조건 하에서 위험이 발생할 우려가 있으므로 주의하라는 기호입니다.

※모두 중요한 내용이 기재되어 있기 때문에, 반드시 지켜 주십시오.

취급 설명서를 읽고 난 뒤에는 제품을 사용하는 사람이 항상 볼 수 있는 곳에 보관하여 주십시오.

◆ 본 제품의 취급

⚠ 경고

- 인명이나 재산상의 영향이 큰 기기(예:원자력 제어, 의료기기, 차량, 철도, 항공, 연소 장치, 오락기기등 또는 안전 장치)에 사용할 경우 반드시 2중으로 안전장치 부착 후 사용하여 주십시오.
화재, 인명 사고, 재산상의 손실이 발생할 수 있습니다
- 취급 설명서에 기재된 일반 사양의 환경에서 사용하십시오. 부식성 가스, 인화성 가스가 있는 장소, 고온, 다습, 진동, 화재, 오동작, 제품의 손상 또는 열화의 원인이 됩니다.
화재, 인명사고, 재산상의 손실이 발생할 수 있습니다.
- 제품을 개조하지 마십시오.
화재, 인명사고, 재산상의 손실이 발생할 수 있습니다
- 운전 중에 전원을 차단하지 말아 주십시오.
인명사고, 재산상의 손실, 오작동의 원인이 됩니다
- 운전시에는 항상 비상 정지가 가능하도록 하십시오.
장치 파손 및 인명 사고의 우려가 있습니다.
- 운전 중에 커넥터 및 집프핀을 분리하지 마십시오.
인명사고, 재산상의 손실, 오작동의 원인이 됩니다.
- 제품의 폐기 시에는 산업 폐기물로서 처리하여 주십시오.

⚠ 주의

- 전원이 인가된 상태에서 결선 및 점검, 보수를 하지 마십시오.
감전, 오동작의 원인이 됩니다.
- 자사 기술자 이외에는 제품을 수리하지 마십시오.
감전이나 화재의 우려가 있습니다. 수리가 필요할 시에는 당사로 문의하여 주십시오.
- 만드시 정격/성능 범위에서 사용하여 주십시오.
제품의 수명이 짧아지는 원인이 되며 화재의 우려가 있습니다.
- 청소 시 물, 유기 용제를 사용하지 마십시오.
감전, 화재, 제품의 손상의 원인이 됩니다.
- 본 제품의 표면에 먼지나 배선 찌꺼기가 유입되지 않도록 하여 주십시오.
감전, 화재, 제품의 손상의 원인이 됩니다.

◆제품의 보관

⚠ 주의

본 제품을 사용 후, 보관시에는 PC와 I/O케이블을 분리하여 정전기 방지용 포장지로 포장하여 정격 및 사양에서 표기된 온도 및 습도 이내에서 보관 하십시오.

◆PC로의 설치

기판의 Edge connector를 PC의 PCI bus connector에 정확하게 삽입하여, 취부 금속부를 나사로 조여 주십시오.
설치작업은 반드시 PC의 전원을 차단한 후에 실행하여 주십시오.

⚠ 경고

1. 반드시 PCI bus connector에 장착하여 사용 하십시오.
장치파손 및 화재나 감전, 부상의 우려가 있습니다.

⚠ 주의

1. 전원입력은 반드시 절연 트랜스를 장착하여 사용 하십시오.
화재나 감전, 부상의 우려가 있습니다.

◆입/출력신호의 접속

외부전원이나 입력/출력신호와의 접속에 있어서 극성을 반대로 하거나 정격범위를 넘는 전압/전류를 가하면 회로 소자를 파괴하거나 동작의 신뢰성을 저하시키는 원인이 됩니다. 충분히 배선을 확인한 후에 접속하여 주십시오.

⚠ 경고

1. 접속은 반드시 접속도를 기초로 해서 배선하여 주십시오.
화재나 감전 및 제품 파손의 우려가 있습니다.
2. 외부 전원의 이상, 컨트롤러의 고장이라도, 반드시 시스템 전체가 안전하도록 컨트롤러의 외부에 안전 보호 장치를 설치하십시오.
화재나 감전 및 제품 파손의 우려가 있습니다.
3. 리미트(한계) 스위치를 반드시 설치하십시오.
인명사고, 재산상의 손실이 발생할 수 있습니다.
4. 긴급 정지 스위치를 반드시 설치하십시오.
인명사고, 재산상의 손실이 발생할 수 있습니다.

⚠ 주의

1. 설치 및 배선 작업 등을 할 때에는 반드시 전원을 차단하고 설치하십시오.
감전, 제품 손상의 우려가 있습니다.
2. 설치 및 배선 작업 등을 할 때에는 배선끼리 서로 단락되지 않도록 주의하십시오.
감전, 제품 손상의 우려가 있습니다.
3. 사용하지 않는 단자는 아무것도 결선하지 말고 다른 단자와 단락되지 않도록 주의하십시오.
감전, 제품 손상의 우려가 있습니다.

■특수 용어 설명

- active(Active)** : 어떤 신호에 있어, 그 신호가 갖는 기능이 유효한 상태에 있는 것.
- drive(Drive)** : Pulse열 입력의 서보 모터 또는 스테핑 모터의 driver 구동 디바이스에 대하여, 모터를 회전시키기 위한 pulse를 출력하는 동작.
- 정량 pulse drive** : 지정된 pulse 양만큼 pulse를 출력하는 drive.
- 연속 pulse drive** : 정지 신호가 active 될 때까지 drive pulse출력을 계속 출력하는 drive.
- CW** : 시계 방향(clockwise의 약어)
- CCW** : 반 시계 방향(counter clockwise의 약어).
- 보간 노드** : 연속 보간을 구성하는 하나 하나의 보간 drive.
- 가속도** : 단위 시간당의 가속도/감속도의 증가 / 감소율
- 2의 보수** : 2 진수에 있어서 부의 값의 표현 방법.
(예) 16 bit 길이의 data로는, -1은 FFFFh, -2는 FFFEh, -3은 FFFDh,
-32768은8000h로 표현합니다.

■특수 문자 기호 설명

- n○○○○** : X, Y, Z, U의 각축의 신호명을 n○○○○ 라고 기술하고 있습니다.
이 때, "n"는 X, Y, Z 및 U를 나타냅니다.
- ↑** : 신호가 Low level에서 Hi level로 변화할 때의 상승 에지.
- ↓** : 신호가 Hi level에서 Low level로 변화할 때의 하강 에지.
- nPP, nPM** : nPP는 nP+P, nP+N을 nPM은 nP-P, nP-N을 나타냅니다.
- nECA, nECB, nECZ** : EC는 엔코더를 나타내며 A, B, C는 출력상을 나타냅니다.
레지스터 설명시 nIN2는 nECZ를 나타냅니다.

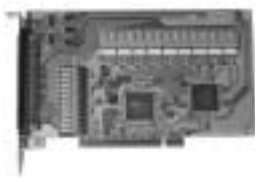
■기능의 제한

본 보드에서는 편차 카운터 클리어 기능, BUSYN, SCLK은 지원하지 않습니다.

■제품의 구성

본 모션 콘트롤러 PMC-4B-PCI는 아래와 같이 구성되어 있습니다.

PMC-4B-PCI Board



I/O 케이블



사용자 CD



취급설명서



■윈 드라이브 인스톨

제공된 CD의 PMC-4B-PCI/winDriver Install/Autonics.exe 파일을 실행합니다. 윈 드라이브 인스톨 방법은 winDriver Install 폴더안에 DOC파일(winDriver Install 방법)로 작성되어 있으며 Autonics.exe 파일로 윈 드라이브가 인스톨 되지 않으면 수동으로 mc8041p.inf의 위치를 설정합니다. mc8041p.inf파일은 PMC-4B-PCI/winDriver Install 폴더와 PMC-4B-PCI/Test program 폴더 안에 있습니다.

■I/O test

PMC-4B-PCI보드의 입/출력 테스트를 위한 프로그램입니다. 본 프로그램의 소스는 PMC-4B-PCI/Test program/iotest source 폴더 안에 있습니다. PC에 카드를 장착하고, 윈 드라이브를 인스톨 한 후, PMC-4B-PCI/Test program 폴더 안의 ioTest.exe 를 실행하면 다음과 같은 화면이 활성화 합니다.

목차

1. 개 요	1
2. 기능 설명	6
2.1 정량 Pulse drive와 연속 Pulse drive	6
2.1.1 정량 Pulse drive	6
2.1.2 연속 PPulse drive	7
2.2 속도 Curve	8
2.2.1 정속 Drive	8
2.2.2 직선 가감속 Drive	9
2.2.3 비대칭 직선 가감속 Drive	10
2.2.4 S자형 가감속 Drive	11
2.2.5 비대칭 S자형 가감속 Drive	14
2.2.6 Drive pulse 폭과 속도 정밀도	15
2.3 포지션(Position) 관리	16
2.3.1 논리 위치 Counter와 실제위치 Counter	17
2.3.2 비교 Register와 소프트웨어 Limit	17
2.3.3 위치 Counter의 가변 링(Ring)	17
2.3.4 외부 신호에 의한 실제 위치 Counter의 Ring	18
2.4 보 간	19
2.4.1 2축/3축 직선보간	19
2.4.2 원호보간	21
2.4.3 2축/3축 Bit pattern 보간	23
2.4.4 선속일정	27
2.4.5 연속보간	28
2.4.6 가감속 Drive에서의 보간	31
2.4.7 보간 Step 전송(Command)	33
2.5 자동 원점 복귀 출력	33
2.5.1 각 Step의 동작	34
2.5.2 서치 속도와 Mode의 설정	35
2.5.3 자동 원점 출력의 실행과 상황(Status)	36
2.5.4 자동 원점 출력시의 에러	37
2.5.5 자동 원점 출력시의 주의점	38
2.5.6 자동 원점 복귀의 예	39
2.6 동기 동작	43
2.6.1 동기 동작의 예	44
2.6.2 동기 동작의 지연 시간	49
2.6.3 동기 동작의 주의점	50
2.7 Interrupt	50
2.8 입력 신호 필터	51
2.9 그 밖의 기능	53
2.9.1 외부 신호에 의한 Drive 조작	53
2.9.2 Pulse 출력 방식의 선택	55
2.9.3 Pulse 입력 방식의 선택	55
2.9.4 하드웨어 Limit 신호	56
2.9.5 서보(Servo)모터 Driver 대응의 신호	56
2.9.6 긴급 정지	56
2.9.7 Drive 상태의 출력	56
2.9.8 범용 출력 신호	57

3. 신호의 설명	57
4. Read / Write register	59
4.1 16Bit data bus의 Register address	59
4.2 8Bit data bus의 Register address	61
4.3 WR0 Command register	61
4.4 WR1 Mode register 1	62
4.5 WR2 Mode register 2	63
4.6 WR3 Mode register 3	64
4.7 WR5 보간 Mode register	65
4.8 WR6, 7 Write data register 1, 2	66
4.9 RR0 주 Status register	66
4.10 RR1 Status register 1	67
4.11 RR2 Status register 2	68
4.12 RR3 Status register 3	69
4.13 RR4, 5 Input register 1, 2	69
4.14 RR6, 7 Read register 1, 2	70
5. 명령 일람	70
6. Data 기록(Write) 명령	71
6.1 Range 설정 R	71
6.2 가속도 증가율 설정 K	72
6.3 가속도 설정 A	72
6.4 감속도 설정 D	72
6.5 기동속도 설정 SV	72
6.6 Drive 속도 설정 V	73
6.7 출력 Pulse 수 / 보간 종점 설정 P	73
6.8 매뉴얼(Manual) 감속점 설정 DP	73
6.9 원호 중심점 설정 C	73
6.10 논리 위치 Counter 설정 LP	74
6.11 실제 위치 Counter 설정 EP	74
6.12 Comp+ register 설정 CP	74
6.13 Comp- register 설정 CM	74
6.14 가속 Counter offset 설정 AO	74
6.15 감속도 증가율 설정 L	74
6.16 확장 Mode 설정 EM	75
6.17 원점 검출 속도 설정 HV	76
6.18 동기 동작 Mode 설정 SM	76
7. Data 읽기(Read) 명령	77
7.1 논리위치 Counter 읽기 LP	77
7.2 실제위치 Counter 읽기 EP	77
7.3 현재 Drive 속도 읽기 CV	77
7.4 현재 가/감속도 읽기 CA	77
7.5 동기 버퍼 Register 읽기 BR	77
8. Drive 명령	77
8.1 +방향 정량 Pulse drive	78
8.2 -방향 정량 Pulse drive	78
8.3 +방향 연속 Pulse drive	78
8.4 -방향 연속 Pulse drive	78
8.5 Drive 시작 홀드(Hold)	78
8.6 Drive 시작(Free) / 종료 Status clear	79
8.7 Drive 감속정지	79
8.8 Drive 즉시정지	79

9. 보간 명령	79
9.1 2축 직선보간 Drive	79
9.2 3축 직선보간 Drive	79
9.3 CW원호보간 Drive	79
9.4 CCW원호보간 Drive	80
9.5 2축 Bit patten 보간 Drive	80
9.6 3축 Bit patten 보간 Drive	80
9.7 BP Register 기입 가능	80
9.8 BP Register 기입 불가	80
9.9 BP Data stack	80
9.10 BP Data clear	80
9.11 보간 Single step	80
9.12 감속 유효	81
9.13 감속 무효	81
9.14 보간 Interrupt clear	81
10. 그 밖의 명령	81
10.1 자동 원점 복귀 실행	81
10.2 동시 동작 기동	81
10.3 NOP	81
11. 입/출력 신호 접속 예	82
11.1 Drive pulse 출력 신호의 접속 (nP+P/N, nP-P/N)	82
11.2 범용 출력 신호의 접속 (nOUT7~4)	82
11.3 Encoder 입력 신호의 접속	82
11.4 입력 신호의 접속 (nIN1~3, nALRAM, nEXP+/-, EMG)	83
11.5 Limit 입력 신호의 접속 (nLMIT+/-)	83
12. 제어 프로그램의 예	83
13. 정격 사양	93
14. 입/출력 신호 타이밍	94
14.1 파워-온 타이밍	94
14.2 Drive 시작/종료 시	94
14.3 보간 Drive 시	94
14.4 Drive 시작 Free	95
14.5 Drive 즉시 정지	95
14.6 Drive 감속 정지	95
15. 외형 치수	96
16. 사양 정리	96
[부록 A] 가감속 Drive의 속도 프로파일	99
◎40kpps 대칭 S자 가감속	99
◎8000pps 대칭 S자 가감속	99
◎400kpps 대칭 S자 가감속	99
◎40kpps 비대칭 S자 가감속	99
◎40kpps 비대칭 대자 가감속	100

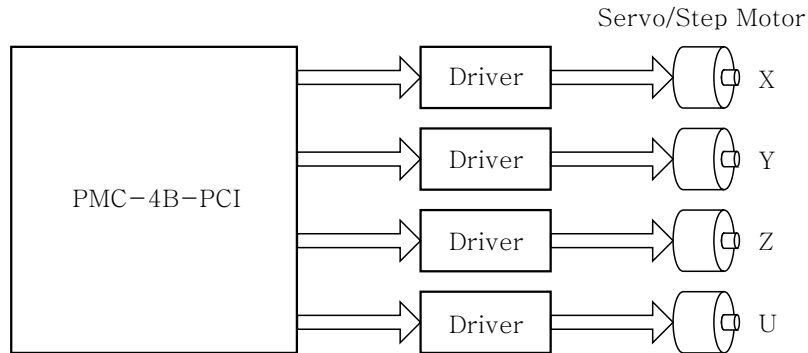
1. 개요

PMC-4B-PCI는, 1Board로 4축의 pulse 입력의 서보 모터, Step 모터의 위치 결정 제어(Positioning control), 보간 Drive(Interpolation drive), 또는 속도 제어(Speed control)가 가능한 PC/AT 호환기 PCI bus 대응의 회로 기판입니다. 다음의 기능을 갖추고 있습니다.

◎독립 4축 Drive

4축을 독립적으로 제어할 수 있습니다. 4축의 기능은 모두 동일합니다.

정속 Drive, 직선 가감속 Drive, S자 가감속 Drive 등 4축 모두 동일하게 제어할 수 있습니다.



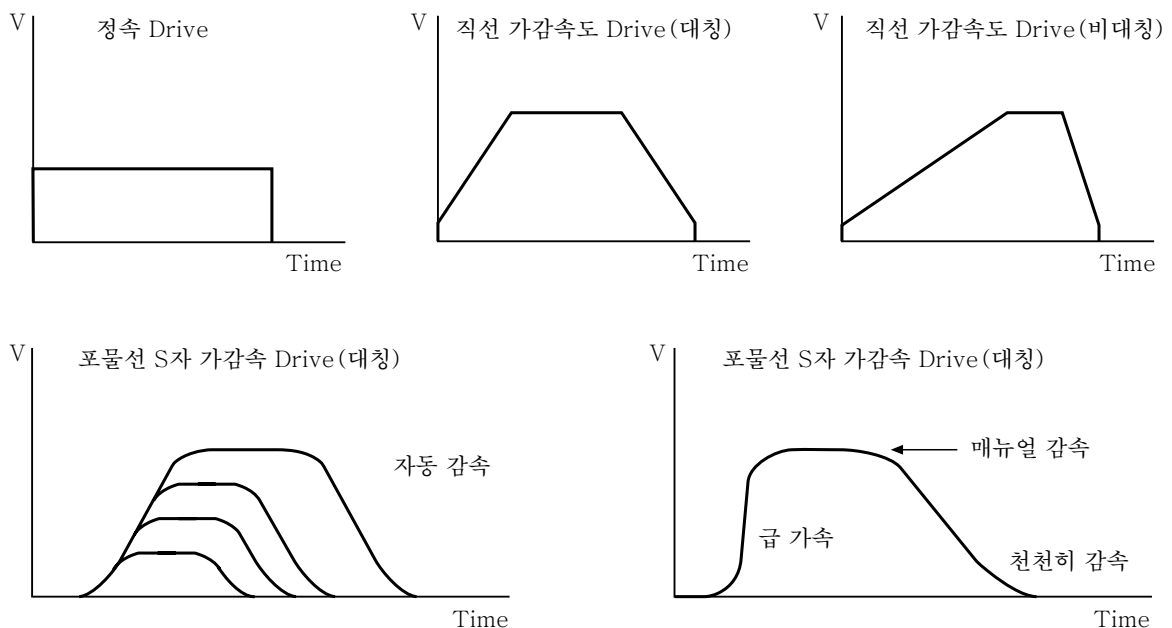
◎속도 제어

Drive 속도는 1 pps에서 최고 4 Mpps까지 출력할 수 있으며 정속 Drive, 직선 가감속 Drive, S자 가감속 Drive가 가능합니다. 출력되는 Drive pulse의 속도 정밀도는 설정치에 대하여 $\pm 0.1\%$ 이하입니다(CLK=16MHz 표준 시). 또한, Drive 중에 Drive 속도를 자유롭게 변경 가능합니다.

◎가감속 Drive

각 축 가감속 Drive에는 정속 Drive, 직선 가감속 Drive(대칭/비대칭), S자 가감속 Drive(대칭 / 비대칭)를 실행할 수 있습니다. 직선 가감속 정량 Pulse drive에서는 대칭형 직선 가감속 정량 Pulse drive와 비대칭형 직선 가감속 정량 Pulse drive 모두, 자동 감속이 가능합니다.

S자 가감속은 1차 직선으로 증가 / 감소하는 방식을 사용하므로 속도 커브는 2차 포물선 가속 / 감속으로 나타납니다. 또한, S자 가감속 정량 Pulse drive에 있어서는 대칭 S자에 한하여 자동 감속이 가능하고 S자 가감속 driver 중 삼각파형 방지 기능도 있습니다.

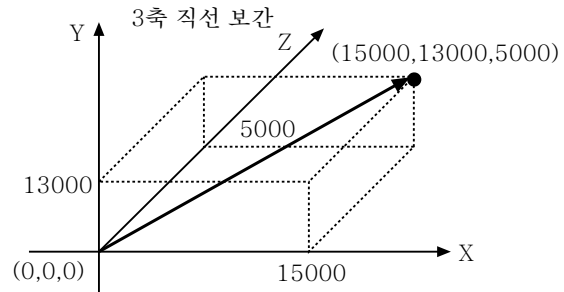
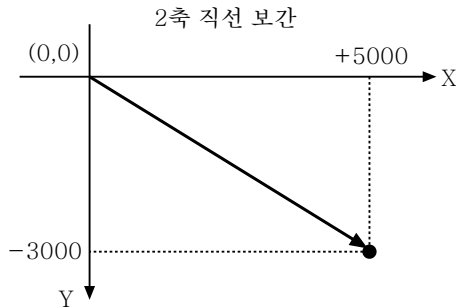


◎ 2축/3축 직선보간

4축의 Driver 중에서 임의의 2축, 또는 3축을 선택하고 2축/3축 직선 보간 Drive를 실행할 수 있습니다.

보간 좌표 범위는 현재 위치에서 $-2,147,483,646 \sim +2,147,483,646$ 입니다.

지정된 직선에 대한 위치 오차는 전 보간 범위 내에서 ± 0.5 LSB 입니다. 보간 속도는 1PPS \sim 4 MPPS입니다.

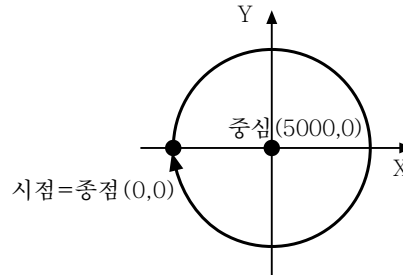
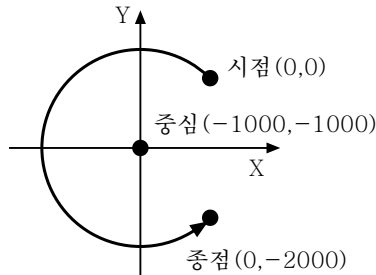


◎ 원호보간

4축의 Driver 중에서 임의의 2축을 선택하고 원호보간 Drive를 실행할 수 있습니다.

보간 좌표 범위는 현재 위치에서 $-2,147,483,646 \sim +2,147,483,646$ 입니다.

지정된 원호 곡선에 대한 위치 오차는 전 보간 범위 내에서 ± 1 LSB 입니다. 보간 속도는 1 PPS \sim 4 MPPS 입니다.

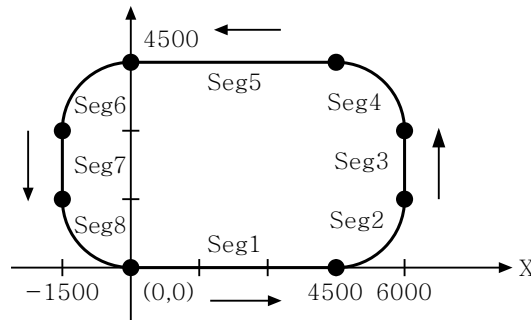


◎ 2축/3축 Bit pattern 보간

상위 CPU 로 연산된 Bit pattern화된 보간 Data를 각 축 16 bit 단위로 받아들여, 지정된 Drive 속도로 보간 Pulse를 연속적으로 출력하는 보간 Drive입니다. 이 기능을 이용하여 상위 CPU에서 만들어진 다양한 궤적을 그릴 수 있습니다.

◎ 연속 보간 동작

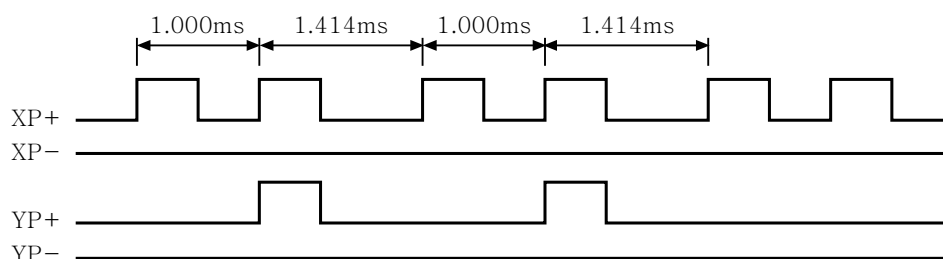
직선 보간 \rightarrow 원호보간 \rightarrow 직선 보간 $\rightarrow \dots$ 처럼, 보간 drive를 각각의 보간 명령마다 정지하지 않고, 연속적으로 실행할 수 있습니다. 연속 보간 시의 Drive 속도는 최고 2 MHz 입니다.



◎ 선속 일정 제어

선속 일정 제어는 보간을 실행하고 있는 축의 합성속도를 항상 일정하게 하는 기능입니다.

2축 동시 Drive pulse를 발생할 때 2축의 Pulse 주기를 1.414 배, 3축 동시에 Drive pulse를 발생할 때는 3축의 Pulse 주기를 1.732 배 합니다.



◎포지션 관리 기능

4축 모두, 모션 컨트롤 IC 내부에서 Drive pulse 출력을 제어하는 논리위치 Counter와 외부 Encoder로부터 Pulse를 제어하는 실제위치 Counter 2개(32bit)를 내장하고 있습니다.

◎비교 Register와 소프트 Limit 기능

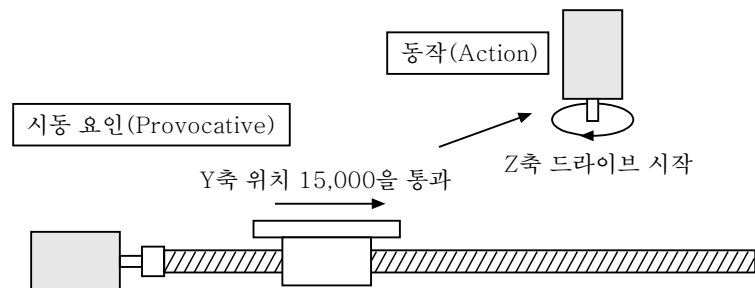
논리 위치 Counter와 실제위치 Counter와의 위치를 비교하기 위한 32bit 비교 Register를 각 축 2개씩 가지고 있습니다. Drive 중에 이러한 비교 Register와 논리/실제 위치 Counter와의 대소 관계를 실시간적으로 읽을 수 있으며, 대소 관계의 변화를 이용하여 Interrupt 발생도 가능합니다. 또한, 2개의 비교 Register를 소프트 Limit로 동작시키는 것도 가능합니다.

◎자동 원점 출력

PMC-4B-PCI는 특별한 명령 없이 고속 원점 근접서치 → 저속 원점서치 → Encoder Z상서치 → Offset 이동 등의 일련의 원점출력 시퀀스를 자동으로 실행하는 기능을 가지고 있습니다. 다축 제어에 있어서 CPU의 부담을 줄입니다.

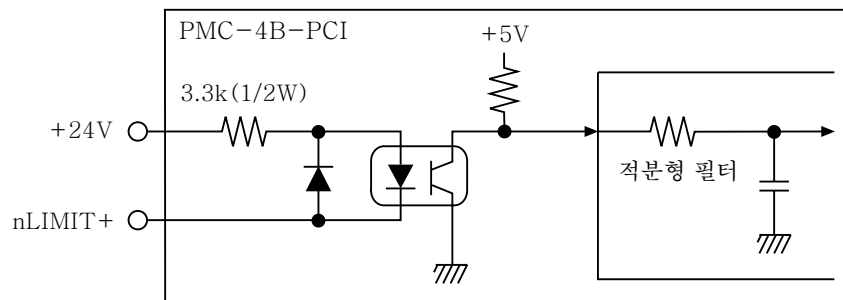
◎동기 동작

동기 동작은 각 축 사이 및 모션 컨트롤러 이외의 디바이스 사이에 있어 설정된 동작 신호가 발생하면 Drive 시작이나 정지 등의 지정된 동기 동작을 실행하는 기능입니다. 동기 동작의 시동 요인으로서 지정 위치 통과, Drive 시작/종료, 외부 입력 신호의 상승/하강 등 10 종류가 준비되어 있으며 동기 동작 발생 후, 동작으로서 Drive 시작/정지, 위치 Counter 값 저장, Drive 속도 기록 등 14종류가 준비되어 있습니다.



◎입력신호 필터

각 입력신호의 입력단에는 적분필터를 갖추고 있습니다. 입력신호 마다 필터 기능을 유효하게 할지, 무효하게 할지를 설정할 수 있습니다. 또한, 필터의 통과 시간은 8종류 중에서 하나를 선택 할 수 있습니다.



◎외부 조작 신호

각 축은 외부 신호에 의하여 +/- 방향의 정량 Pulse drive, 연속 Pulse drive를 실시할 수 있습니다. 이 기능이 수동 Pulse 출력(MANUAL JOG DRIVE)으로 가능하여 상위 CPU의 부담을 줄일 수 있습니다.

◎서보 모터용 각종 신호

2상 Encoder 신호, 인포지션(Inposition), Alarm 등의 서보 모터 Driver 출력 신호를 입력할 수 있습니다.

◎Interrupt 발생 기능

각 축 모두, 가감속 Drive 중의 정속 시작 시, 정속 종료 시, Drive 종료 시, 위치 Counter와 비교 Register의 대소 관계가 변경 되었을 경우 등 다양한 요인으로 Interrupt를 발생시킬 수 있습니다. 또한, 연속 보간, Bit pattern 보간에서는 다음 Data 요구에 대한 Interrupt 발생도 가능 합니다.

◎실시간 처리 모니터 기능

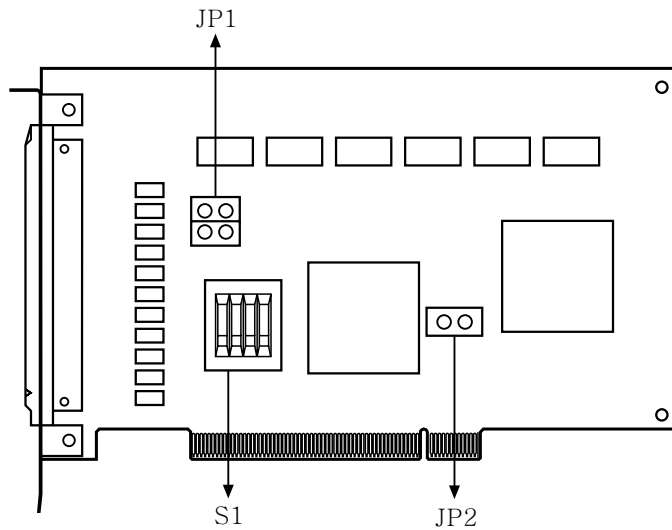
Drive 중에 현재의 논리 위치, 실제 위치, Drive 속도, 가속도, 가감속 상태(가속 중, 정속 중, 감속 중)등을 실시간으로 읽을 수 있습니다.

◎16 bit bus 대응

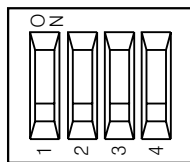
상위 CPU와의 data bus는 16 bit로 접속이 가능합니다.

X, Y, Z, U 4축의 기능은 완전히 동일하며 보간 Drive 시에는 주축(AX1)으로 지정된 축의 기본 Pulse 발진의 타이밍에 보간 연산을 합니다. 정속 Drive에서도 가감속 Drive에서도 실시할 수가 있습니다.

◎스위치 및 점퍼 핀의 용도



●S1 스위치의 용도 및 설정

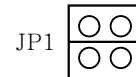


S1 스위치는 PMc-4B-PCI 보드를 여러장 사용할 때, 보드의 ID를 구분하기 위한 스위치로 특별한 기준없이 다른 보드와 스위치 설정을 다르게 하여 사용하시면 됩니다.

총 16개의 보드를 동시 사용 가능합니다.

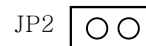
●JP1 점퍼 핀의 용도 및 설정

긴급 정지(EMG)의 Active 레벨을 Hi로 할지 Low로 할지를 선택합니다.



긴급정지의 액티브 레벨 설정은 2.9.6절을 참조하십시오.

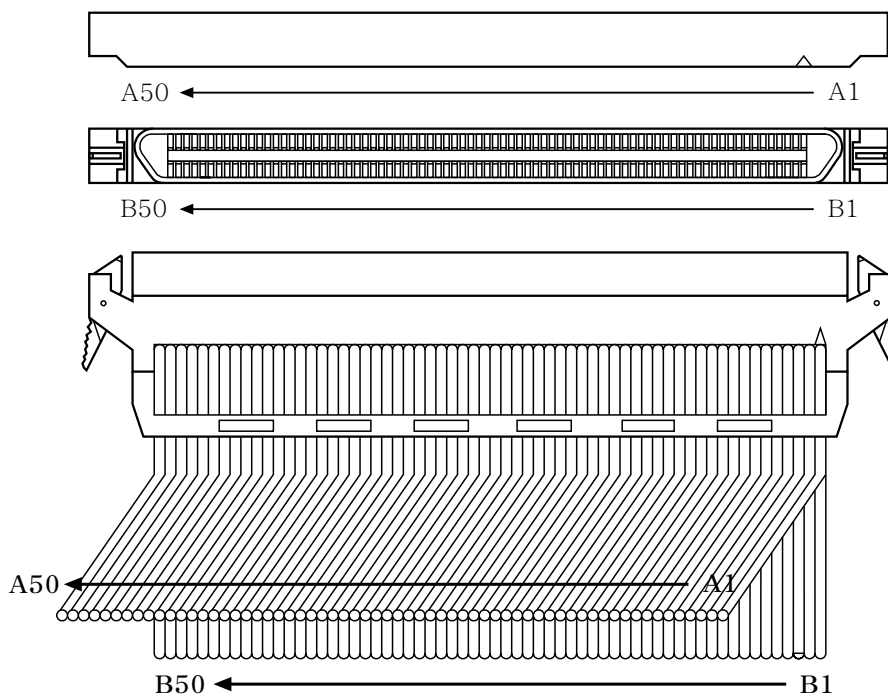
●JP2 점퍼 핀의 용도 및 설정



EEPROM의 프로그램 쓰기 마스크로써 사용자는 JP2 점퍼핀을 임의로 제거할 수 없습니다.

◎입/출력 사양

I/O Connector의 핀 배치



핀 번호	핀 이름	핀 설명	핀 번호	핀 이름	핀 설명
A1	VEX	12~24VDC	B1	VEX	12~24VDC
A2	EMG	비상정지(4축 정지)	B2	-	-
A3	XLIMIT+	X축 +방향 리미트	B3	ZLIMIT+	Z축 +방향 리미트
A4	XLIMIT-	X축 -방향 리미트	B4	ZLIMIT-	Z축 -방향 리미트
A5	XIN1	입력신호(원점신호)	B5	ZIN1	입력신호(원점신호)
A6	XIN0	입력신호(원점근접신호)	B6	ZIN0	입력신호(원점근접신호)
A7	XIN3	입력신호	B7	ZIN3	입력신호
A8	YLIMIT+	Y축 +방향 리미트	B8	ULIMIT+	U축 +방향 리미트
A9	YLIMIT-	Y축 -방향 리미트	B9	ULIMIT-	U축 -방향 리미트
A10	YIN1	입력신호(원점신호)	B10	UIN1	입력신호(원점신호)
A11	YIN0	입력신호(엔코더 Z상신호)	B11	UIN0	입력신호(원점근접신호)
A12	YIN3	입력신호	B12	UIN3	입력신호
A13	XINPOS	위치결정완료 입력	B13	ZINPOS	위치결정완료 입력
A14	XALARM	알람 입력	B14	ZALARM	알람 입력
A15	XECAP	X축 엔코더 A상+	B15	ZECAP	Z축 엔코더 A상+
A16	XECAN	X축 엔코더 A상-	B16	ZECAN	Z축 엔코더 A상-
A17	XECBP	X축 엔코더 B상+	B17	ZECBP	Z축 엔코더 B상+
A18	XECBN	X축 엔코더 B상-	B18	ZECBN	Z축 엔코더 B상-
A19	XECZP	X축 엔코더 Z상+	B19	ZECZP	Z축 엔코더 Z상+
A20	XECZN	X축 엔코더 Z상-	B20	ZECZN	Z축 엔코더 Z상-
A21	YINPOS	위치결정완료 입력	B21	UINPOS	위치결정완료 입력
A22	YALARM	알람 입력	B22	UALARM	알람 입력
A23	YECAP	Y축 엔코더 A상+	B23	UECAP	U축 엔코더 A상+
A24	YECAN	Y축 엔코더 A상-	B24	UECAN	U축 엔코더 A상-
A25	YECBP	Y축 엔코더 B상+	B25	UECBP	U축 엔코더 B상+
A26	YECBN	Y축 엔코더 B상-	B26	UECBN	U축 엔코더 B상-
A27	YECZP	Y축 엔코더 Z상+	B27	UECZP	U축 엔코더 Z상+
A28	YECZN	Y축 엔코더 Z상-	B28	UECZN	U축 엔코더 Z상-
A29	XEXP+	X축 매뉴얼 +드라이브	B29	ZEXP+	Z축 매뉴얼 +드라이브
A30	XEXP-	X축 매뉴얼 -드라이브	B30	ZEXP-	Z축 매뉴얼 -드라이브
A31	YEXP+	Y축 매뉴얼 +드라이브	B31	UEXP+	U축 매뉴얼 +드라이브
A32	YEXP-	Y축 매뉴얼 -드라이브	B32	UEXP-	U축 매뉴얼 -드라이브
A33	GND	GND	B33	GND	GND
A34	XOUT4/CMPP	범용출력	B34	ZOUT4/CMPP	범용출력
A35	XOUT5/CMPP	범용출력	B35	ZOUT5/CMPP	범용출력
A36	XOUT6/ASND	범용출력	B36	ZOUT6/ASND	범용출력
A37	XOUT7/DSND	범용출력	B37	ZOUT7/ SND	범용출력
A38	XP+P	X축 +방향 드라이브	B38	ZP+P	Z축 +방향 드라이브
A39	XP+N	X축 +방향 드라이브	B39	ZP+N	Z축 +방향 드라이브
A40	XP-P	X축 -방향 드라이브	B40	ZP-P	Z축 -방향 드라이브
A41	XP-N	X축 -방향 드라이브	B41	ZP-N	Z축 -방향 드라이브
A42	GND	GND	B42	GND	GND
A43	YOUT4/CMPP	범용출력	B43	UOUT4/CMPP	범용출력
A44	YOUT5/CMPP	범용출력	B44	UOUT5/CMPP	범용출력
A45	YOUT6/ASND	범용출력	B45	UOUT6/ASND	범용출력
A46	YOUT7/DSND	범용출력	B46	UOUT7/DSND	범용출력
A47	YP+P	Y축 +방향 +드라이브	B47	UP+P	U축 +방향 +드라이브
A48	YP+N	Y축 +방향 -드라이브	B48	UP+N	U축 +방향 -드라이브
A49	YP-P	Y축 -방향 +드라이브	B49	UP-P	U축 -방향 +드라이브
A50	YP-N	Y축 -방향 -드라이브	B50	UP-N	U축 -방향 -드라이브

2. 기능 설명

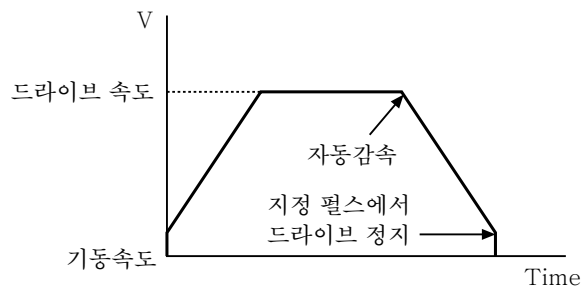
2.1 정량 Pulse drive와 연속 Pulse drive

각 축의 Drive pulse 출력은 기본적으로 + 방향/- 방향의 정량 Pulse drive 명령, 또는 연속 Pulse drive 명령이 있습니다.

2.1.1 정량 Pulse drive

정량 Pulse drive는 지정된 출력 Pulse수 만큼 정속 Drive 또는, 가감속 Drive를 합니다. 이동 대상물을 일정한 위치로 이동시킬 때, 정해진 펄스 양(정해진 이동 거리)만큼 동작을 실시하고 싶을 때에 사용합니다.

가속도와 감속도가 동일한 가감속에서의 정량 Pulse drive의 동작은, <그림 2.1>과 같이 출력 Pulse의 나머지가, 가속시에 소비된 Pulse 수 보다 작으면 자동 감속을 시작하고, 지정된 출력 Pulse를 출력하면 Drive를 종료합니다.



<그림 2.1 정량 Pulse Drive>

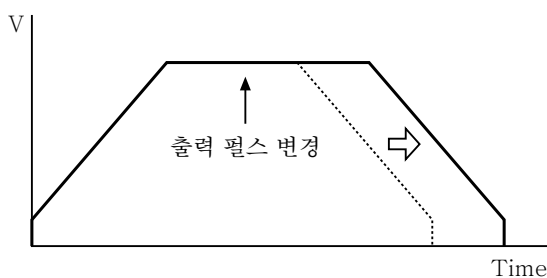
정량 Drive를 가감속 Drive로 실시하기 위해서는 다음과 같은 Parameter 설정이 필요합니다.

●Range	R
●가/감속도	A/D
●기동속도	SV
●Drive 속도	V
●출력 pulse수	P

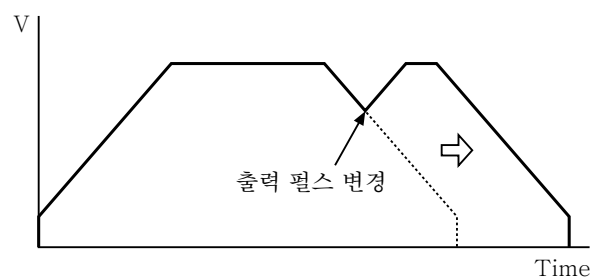
◎Drive 중, 출력 Pulse수 변경

정량 Pulse drive중에 출력 Pulse 수를 변경할 수 있습니다.

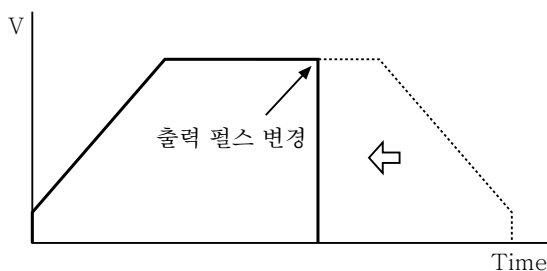
가감속 Drive 중, 출력 Pulse의 나머지가 가속시 Pulse수 보다 적게 되어, 감속으로 들어갈 때 출력 Pulse 수가 변경되는 경우는 다시 가속을 시작합니다. <그림 2.2>. 또한, 변경된 출력 Pulse 수가, 이미 종료된 Pulse 수보다 작은 경우는, 즉시정지<그림 2.4> 합니다. S자 가감속에서는, <그림 2.3>과 같이 감속시에 pulse수가 변경되면 올바른 S자 커브를 그릴 수가 없기 때문에 주의하십시오.



<그림 2.2 Drive 중 출력 Pulse 수 변경>



<그림 2.3 감속시 출력 Pulse 수 변경도>



<그림 2.4 Drive 중 출력 Pulse 수 변경>

◎가감속 정량 Pulse drive에 있어서의 메뉴얼 감속

가감속 정량 Pulse drive에서는, 일반적으로 <그림 2.1>과 같이 계산된 감속점에서 자동 감속하지만 감속점을 메뉴얼로 지정할 수도 있습니다. 아래와 같은 경우, 자동 감속점이 지정되지 않거나 올바른 감속점을 산출할 수 없기 때문에 메뉴얼로 감속점을 지정해야 합니다.

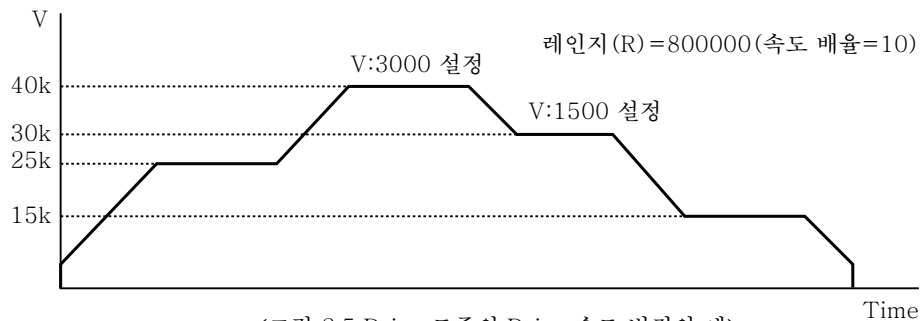
- 직선 가감속 정량 Drive에 있어서 Drive 도중에 속도 변경을 할 경우
- S자 가감속 정량 Drive에 있어서 가속도와 감속도, 가속도 증가율과 감속도 증가율을 개별로 설정 할 경우
- 원호보간, bit pattern 보간, 연속보간을 가감속으로 실시 할 경우

메뉴얼 감속 Mode로 하려면, WR3 register의 D0 bit를 1로 하여 메뉴얼 감속점 설정 명령(07h)에 의해 감속점을 설정합니다. 그 외의 조작은 일반적인 정량 Drive와 같습니다.

◎Drive 중 Drive 속도 변경

직선 가감속, 및 정속의 정량 Drive에 대해서는 Drive 도중에 Drive속도(V)를 변경할 수가 있습니다.

다만, 직선 가감속 정량 Pulse drive에서 Drive 속도를 변경하면 Drive가 정지되는 경우가 발생하기 때문에 기동속도가 낮은 경우에는 주의해 주십시오. 또한, S자 가감속 정량 Drive에서는 Drive 도중에 Drive 속도(V)를 변경할 수 없습니다.

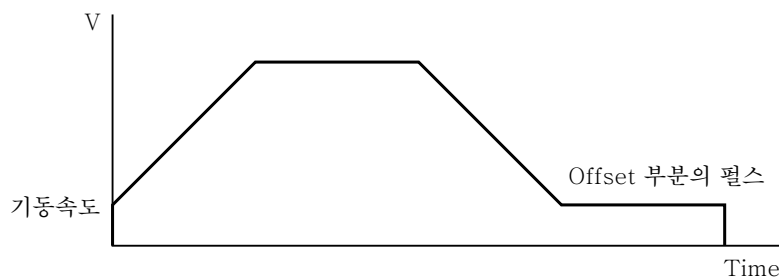


<그림 2.5 Drive 도중의 Drive 속도 변경의 예>

◎가감속 정량 Pulse drive의 가속 Counter offset

가감속 정량 drive의 동작에서는 가속시에 가속으로 소비되는 Pulse를 가속 Counter로 Count합니다.

설정되어 있는 출력 Pulse 수의 나머지가 가속 Counter의 값보다 적게 되면 감속을 시작하고, 감속에는 가속과 같은 Pulse수를 출력합니다. 가속 Counter offset은 가속 Counter로 지정된 Offset 값을 <그림 2.6>과 같이 추가합니다. 가속 Counter offset은 Reset시 8로 설정됩니다. 일반적으로는 직선 가감속 Drive를 실시하는 경우에는 본 Parameter를 재 설정할 필요는 거의 없습니다. 비대칭 사다리꼴 가감속이나 S자 가감속 정량 Drive로, Drive 종료 시에 속도가 기동속도까지 떨어지지 않을 때 등에, 가속 Counter offset을 적당한 값에 설정하여 보정합니다.



<그림 2.6 가속 Counter offset>

2.1.2 연속 Pulse drive

연속 Drive는 정지 명령, 또는 외부로부터 정지 신호가 Active될 때까지 연속하여 Drive pulse를 출력합니다.

원점 서치, 스캐닝 조그 이송, 또는 속도 제어로 모터를 회전시킬 때 사용합니다.

연속 Pulse drive에서는, Drive 도중에 Drive 속도를 자유롭게 변경할 수가 있습니다.

정지 명령에는 감속정지 명령과 즉시정지 명령이 있습니다.

외부로부터 감속/즉시정지 신호는 각 축 IN3~IN0(+/-)의 4점이 준비되어 있으며 각 신호는 유효/무효, Active level을 설정할 수가 있습니다.

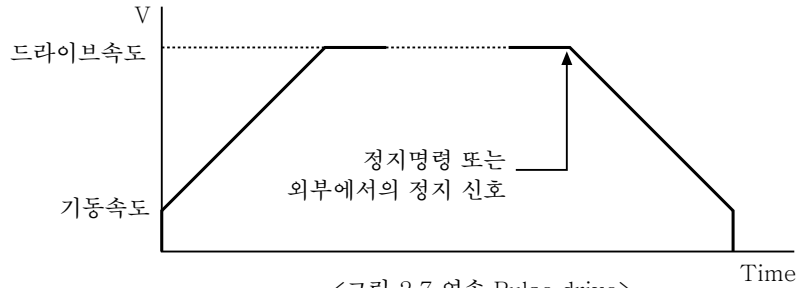
◎연속 Pulse drive에 의한 원점 검출 동작

Encoder Z상신호, 원점신호, 원점 근접신호를 nECZ, nIN1, nIN0에 설정합니다. 각 축의 WR1 register로 각 신호의 유효/무효, 논리 level을 설정합니다. 고속서치의 경우, 가감속으로 Drive하여 연속 Pulse drive를 실시합니다.

이때, 설정된 신호가 Active level이 되면 감속정지 합니다.

저속서치의 경우는 정속으로 연속 Pulse drive를 실시합니다. 설정된 신호가 Active level이 되면 즉시 정지합니다.

자동원점복귀 기능을 사용하는 경우에는 Z상신호는 nECZP/N, 원점 신호는 nIN1, 원점 근접 신호는 nIN0으로 설정됩니다. 연속 Drive를 가감속으로 실시하려면 출력 Pulse 수 이외는 정량 Drive와 같은 Parameter로 설정합니다.



<그림 2.7 연속 Pulse drive>

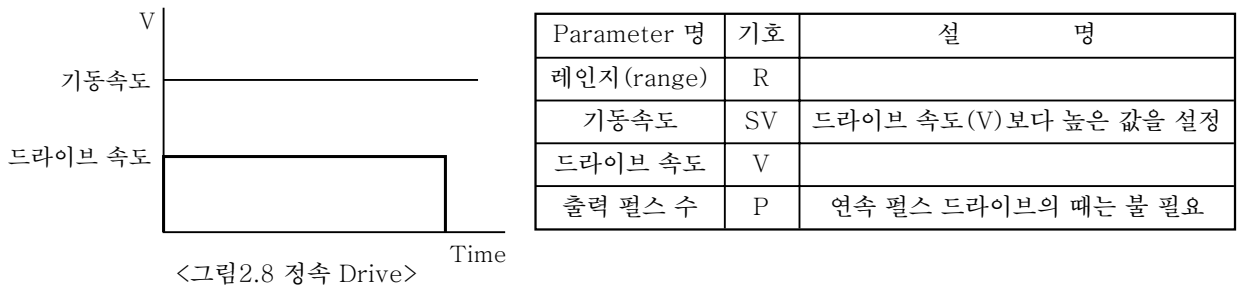
2.2 속도 Curve

각 축의 Drive pulse 출력은 기본적으로 +방향/-방향의 정량 Drive 명령, 또는 연속 Drive 명령으로 실시하지만, 이러한 Drive를 mode 설정, 동작 Parameter의 값에 의해 정속, 직선 가감속, 비대칭직선 가감속, S자 가감속, 비대칭 S자 가감속의 속도 커브로도 설정할 수가 있습니다.

2.2.1 정속 Drive

정속 Drive는 항상 일정한 속도로 Drive pulse를 출력합니다. 본 제품은 Drive속도가 기동속도보다 낮은 경우에는 가감속 Drive는 실행되지 않고, 시작부터 일정속도 drive가 됩니다.

원점 서치나, Encoder의 Z상 서치 등의 신호가 검출될 때 즉시정지시키고 싶을 때는 가감속 Drive를 실시하지 않고 처음부터 낮은 속도로 정속 Drive를 실시합니다. 정속 Drive를 실시하려면, 다음 Parameter를 설정합니다.

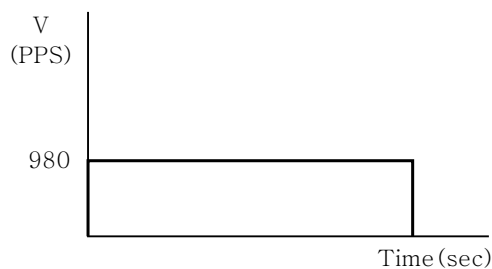


<그림2.8 정속 Drive>

◎Parameter 설정 예

아래와 같이 980PPS로, 정속 Drive 합니다.

Range R = 8,000,000 : 배율=1
 기동속도 SV = 980 : 기동속도 ≥ Drive 속도의 값을 설정
 Drive 속도 V = 980
 출력 Pulse 수 P = 2,450



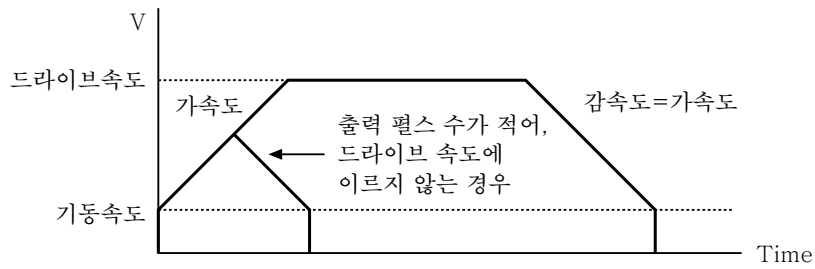
각 Parameter에 대해서는 6장을 참조해 주십시오.

2.2.2 직선 가감속 Drive

직선 가감속 Drive는, Drive 시작 시의 기동속도에서 지정된 가속도의 기울기를 가지는 일차 직선으로 Drive 속도까지 가속합니다. 정량 Drive의 가속도와 감속도가 같은 값(대칭 사다리꼴)인 경우는, 가속시에 소비하는 Pulse가 출력 Pulse의 나머지 보다 작으면 감속을 시작하여 가속도와 같은 기울기를 가지는 일차 직선으로 기동 속도까지 감속하여 모든 출력 Pulse를 출력하면 정지합니다(자동감속).

가속 중에 감속정지가 발생했을 경우, 또는 정량 Drive에서 출력 Pulse수가, Drive 속도까지의 가속이 필요로 하는 Pulse수에 못 미친 경우는, <그림 2.9>와 같이 가속중에 감속합니다. 삼각방지 Mode로 하면 출력 Pulse 수가 적어도 이러한 삼각파형을 사다리꼴 파형으로 발생 가능합니다. 가속도와 감속도는 일반적으로, 같은 값으로 가속도의 값을 사용하지만, 감속도를 개별로 설정할 수도 있습니다.

감속도를 개별로 설정 할 경우는 정량 Drive의 자동 감속이 불가능해지므로 Manual로 감속시켜 주십시오.



<그림 2.9 직선 가감속 Drive(대칭 사다리꼴)>

감속도를 개별로 설정하기 위해서는 WR3 register의 D2 ~ D0 bit가 다음과 같이 설정되어야 합니다.

모드 설정 비트	기호	설정치
WR3/D0	MANLD	0
WR3/D1	DSNDE	0
WR3/D2	SACC	0

WR3 register의 자세한 것은 4.6절을 참조하십시오.

또한, 다음의 Parameter를 설정할 필요가 있습니다.

Parameter 명	기호	설 명
레인지	R	
가속도	A	감속시도 이 값으로 감속합니다.
기동속도	SV	
드라이브 속도	V	
출력 펄스 수	P	연속 펄스 드라이브 시에는 불 필요

◎Parameter 설정 예

아래 그림과 같이, 기동속도:500 PPS, drive 속도:15, 000 PPS까지를 0.3초에 직선 가속/감속합니다.

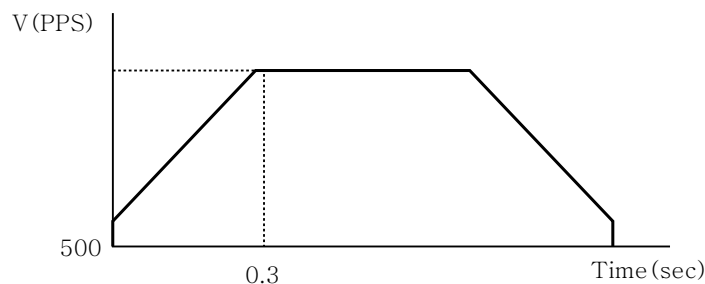
Range R = 4000000 : 배율=2

가속도 A = 193 : $(15000-500)/0.3=48333\text{PPS/SEC}$
: $(48333/125)/2=193$

기동속도 SV = 250 : $500/2=250$

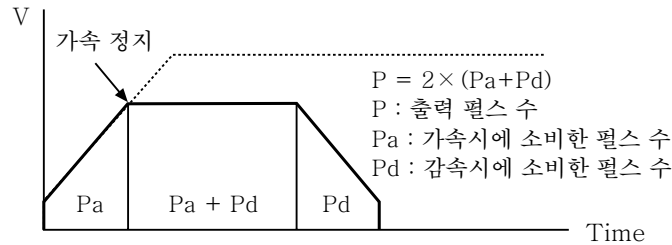
Drive 속도 V = 7500 : $15000/2=7500$

각 Parameter에 대해서는 6장을 참조해 주십시오.



◎정량 Pulse drive의 삼각방지

삼각방지 기능은, 직선 가감속의 정량 Pulse drive에 대해서 출력 Pulse 수가 적을 때 발생하는 삼각파형을 방지하는 기능입니다. 가속중에 소비하는 펄스가 가속시와 감속시에 소비하는 Pulse의 출력 Pulse수의 1/2보다 크면 가속을 정지하고 정속으로 들어갑니다. 따라서 출력 Pulse 수가 아무리 적어도 출력 Pulse 수의 1/2이 정량 Drive됩니다. 삼각 파형 방지 기능은, Reset후에는 적용되지 않습니다. 확장 Mode 설정 명령(60h)의 WR6/D3(AVTRI) bit를 1로 설정하면 유효가 됩니다. 확장 Mode 설정 명령의 자세한 내용은 6.16절을 참조하십시오

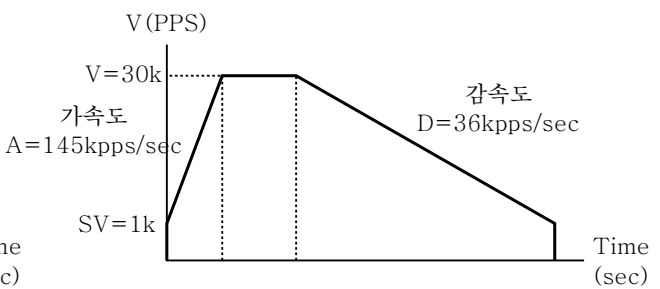
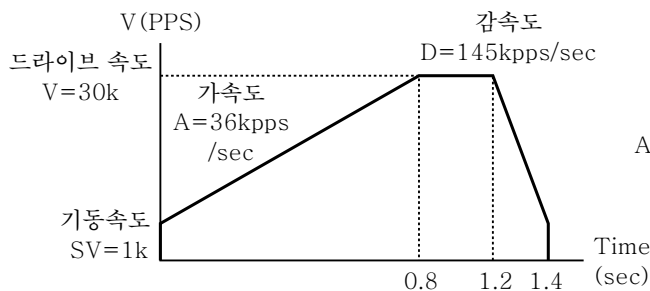


<그림 2.10 직선 가감속 Drive의 삼각 파형 방지>

2.2.3 비대칭 직선 가감속 Drive

다양한 작업을 수행하는 디바이스 중에서, 수직 방향으로 대상물을 움직이는 경우, 대상물에 대해서 중력 가속도의 영향으로 상하 이동의 가속도와 감속도를 바꾸고 싶은 경우가 있습니다. 가속도와 감속도가 다른 비대칭 직선가감속의 정량 Drive에 대해서도 자동 감속이 가능합니다. 미리 계산된 메뉴얼 감속점을 설정하여 <그림 2.11>과 같이 가속도보다 감속도가 큰 예, <그림 2.12>는 감속도보다 가속도가 큰 예를 보입니다.

이러한 비대칭의 직선 가감속에 대해서 출력 Pulse수 P와 각 속도 Parameter로부터 감속 시작점을 설정합니다.



<그림2.11 비대칭 직선 가감속 Drive(가속도<감속도)>

<그림 2.12 비대칭 직선 가감속 Drive(가속도>감속도)>

비대칭 직선 가감속의 정량 Pulse drive를 자동 감속시키려면, WR3 register의 D2~D0 bit를 다음과 같이 설정합니다.

모드 설정 비트	기호	설정치	설 명
WR3/D0	MANLD	0	자동 감속
WR3/D1	DSNDE	1	감속시에 감속도 설정치를 사용한다.
WR3/D2	SACC	0	직선 가감속

또, 다음의 Parameter를 설정할 필요가 있습니다.

Parameter 명	기호	설 명
레인지	R	
가속도	A	
가속도	D	
기동속도	SV	
드라이브 속도	V	
출력 펄스 수	P	연속 펄스 드라이브 시에는 불 필요

[주의]

- 가속도 > 감속도 <그림 2.12>의 경우, 가속도와 감속도의 비율에 대하여 다음과 같은 조건이 있습니다.

$$D > A \times \frac{V}{4 \times 10^6}$$

D : 감속도(pps/sec)
A : 가속도(pps/sec)
V : 드라이브 속도(pps)
CLK = 16MHz

예를 들면, drive속도 V=100 kpps로 하면, 감속도 D는 가속도 A의 값의 1/40보다 큰 값으로 하지 않으면 안됩니다.

- 가속도 > 감속도 <그림 2.12>의 경우, 가속도 A와 감속도 D의 비율이 커지면 감속시 Pulse가 많아져 Drive 속도에 도달하기 전에 감속 Pulse가 발생할 수 있습니다.
이러한 경우에는 ① 기동속도를 올리거나 ② 가속 Counter offset에 마이너스 값을 설정 합니다.

◎Parameter 설정 예

<그림 2.11>의 비대칭 직선 가감속 (가속도<감속도) 정량 Pulse drive의 Parameter 설정은 다음과 같습니다.

WR3 ← 0002h		: WR3 register의 Mode 설정
Range	R = 800000	: 배율 = 10
가속도	A = 29	: (30000-1000)/0.8 = 36250pps/sec (36250/125)/10 = 29
감속도	D = 116	: (30000-1000)/0.2 = 145000pps/sec (145000/125)/10 = 116
기동속도	SV = 100	: 1000/10 = 100
Drive 속도	V = 3000	: 30000/10 = 3000
출력 Pulse수	P = 27500	

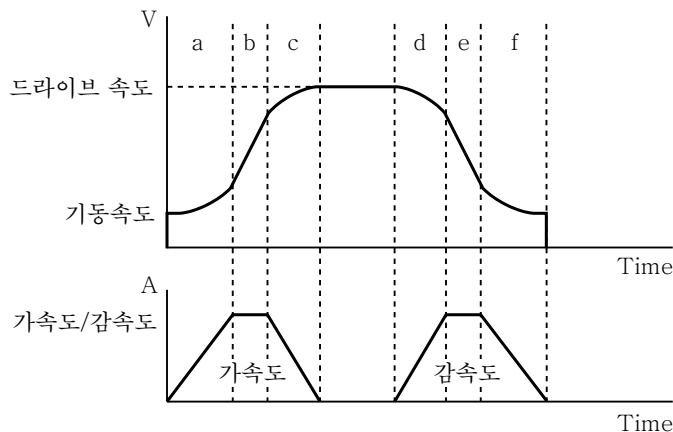
2.2.4 S자형 가감속 Drive

Drive 속도의 가속 및 감속시 가속도/감속도를 일차 직선으로 증가/감소시키는 것으로 속도의 S자 커브를 만들 수 있습니다. 가속과 감속이 대칭형 S자 가감속 Drive는 <그림 2.13>과 같은 동작을 실시합니다.

가속시, 가속도가 0에서 지정된 가가속도(K)로 직선 증가합니다. 따라서, 이 때의 속도 커브는 2차의 포물선 곡선이 됩니다(a구간). 가속도가 (A)로 되면 가속도는 그대로 유지합니다. 이 때의 속도 커브는 직선으로 가속됩니다.

(b구간) Drive의 속도(V)와 현재 속도와의 차가, 가속도를 증가했을 때에 소비하던 Pulse수보다 적어지면, 가속도는 0으로 감소합니다. 감소의 비율은 증가할 때와 같으며 지정된 가가속도(K)에서 직선으로 감소합니다.

이 때의 속도 커브는 2차 포물선이 됩니다(c구간). 이와 같이 가속에 있어서 가속도가 일정하게 되는 부분을 가지는 경우 부분 S자 가속이라고 합니다. 한편, a구간에서 가속도가 지정치(D)에 도달하기 전에 Drive 속도(V)와 현재 속도와의 차가 가속을 증가하였을 때, 소비한 Pulse 수보다 적어지면 b구간은 없어지고 a에서 c구간으로 연결됩니다. 이와 같이, 가속도에 있어서 가속도가 일정하게 되는 부분(b구간)이 없는 가속을 완전 S자 가속이라고 합니다.



<그림 2.13 S자 가감 속도 Drive>

S자 가감속 Drive를 실시 하려면, nWR3 register의 D2, D1, D0 bit를 아래 표와 같이 설정합니다.

모드 설정 비트	기호	설정치	설 명
WR3/D0	MANLD	0	자동 감속
WR3/D1	DSNDE	0	감속시에 가속도, 가속도 증가율 설정치를 사용한다.
WR3/D2	SACC	1	S자 가감속

또한, 다음과 같이 Parameter를 설정할 필요가 있습니다.

Parameter 명	기호	설 명
레인지	R	
가속도 증가율	K	
가속도	A	반드시 최대치 8000으로 설정한다. (*1)
기동속도	SV	
드라이브 속도	V	
출력 펄스 수	P	연속 펄스 드라이브 시에는 불 필요

(*1) :

가속도를 낮게 설정하면, S자 가속시의 가속도 증가 및 감속도 증가에 있어, 가/감속도가 설정된 값(A)까지 상승하지 못합니다.

◎정량 Pulse drive에서의 삼각파형 방지기능

가속과 감속이 대칭형 S자 가감속의 정량 Drive에서는, 출력 Pulse가 drive 속도까지의 가속에 사용된 Pulse 수와 Drive 속도에서 기동속도까지 감속하는데 사용된 Pulse의 수가 같을 경우의 속도 커브는 삼각파형이 됩니다.

기동속도를 0으로 했을 때, 가속도를 가가속도로 시간 t까지 증가시킵니다.

이 때, 시간 t에 있어서의 속도는, $V(t) = at^2$ 로 나타낼 수 있습니다. 따라서, 0에서부터 시간 t까지 소비하는 Pulse 수는, 0에서부터 시간 t까지 속도 $v(t)$ 를 적분한 값이기 때문에, $P(t) = 1/3 \times at^2$ 됩니다.

이 값은 가가속도의 값에 관계없이 $at^2 \times t$ 의 1/3인 것을 나타내고 있습니다.

정량 Drive에 있어서 0에서 시간 t까지 가속도를 어느 가가속도로 증가시켜, 시간 t에서와 같은 가가속도로 감속시킵니다. 가속도가 0이면 감속시에서도 같은 가가속도로 감속을 하여 전체 소비되는 Pulse수는 다음과 같이

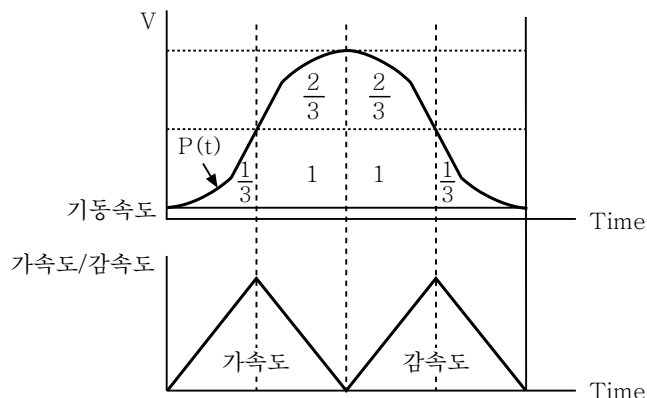
$1/3 + 2/3 + 1 + 2/3 + 1 + 1/3 = 4$ 의 Pulse수가 됩니다.

(*1) : 가속도를 낮게 설정하면, S자 가속시의 가속도 증가 및 감속도 증가에 대해서, 가/감속도가 설정된 값(A) 이상으로 상승하지 않고, 속도 커브가 직선 부분으로 나타납니다. 이 값은, 가속도 증가율의 값에 관계없이, $at^2 \times t$ (그림 중의 1의 Pulse수)의 1/3인 것을 나타내고 있습니다.

정량 Pulse drive에서 0에서 시간 t까지 가속도에 해당하는 가속도 증가율로 증가시켜, 시간 t로부터 같은 가속도 증가 비율로 가속도를 감속시킵니다. 가속도가 0되면, 감속시에도 같은 가속도 증가율로 감소를 실시하고, 전체 소비되는 Pulse수는 <그림 2.14>와 같은 Pulse수가 됩니다.

따라서, 기동시간 0에서부터 시간 t까지의 Pulse수(1/3)는 전체 Pulse수의 1/12 이상으로 S자 가감속 정량 Pulse drive는 가속도 증가시 Pulse가 총 출력 Pulse의 1/12보다 커지면 가속도를 감소하여 <그림 2.14>와 같은 속도 커브를 그립니다. [1/12 법칙]

그러나, 이 방식은 엄밀하게 기동속도 = 0 일 때 이상적인 커브가 됩니다. 기동속도는 실제로 0에는 시작할 수 없기 때문에, 그림 중의 속도 0에서부터 기동속도까지의 Pulse수가 남게 되어, 이 부분은 피크 속도시에 출력되게 됩니다.

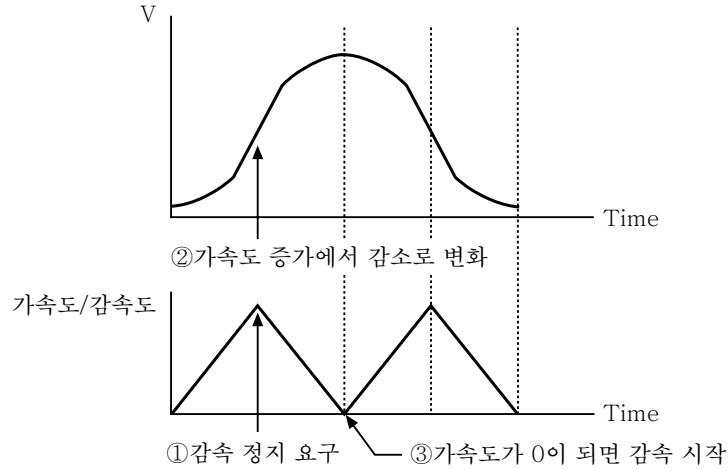


<그림 2.14 포물선 가감 속도의1/12 법칙>

◎감속정지에서의 삼각파형 방지기능

직선 가감속 Drive에서는 가속시에 감속정지를 시켰을 때는 속도 커브가 삼각파형이 됩니다.

S자 가감속 Drive에서는 속도 커브의 부드러움이 중요하기 때문에 <그림 2.15>와 같이 가속시에 감속정지가 걸린 경우, 바로 감속하지 않고 가속도를 일단 0까지 감속시킨 다음 감속을 실행합니다.



<그림 2.15 포물선 가감 속도의1/12 법칙>

◎S자 가감속 Drive시 주의 사항

- S자 가감속의 정량 Drive에 있어서, Drive 속도는 Drive 도중에 변경할 수 없습니다.
- S자 가감속의 정량 Drive에 있어서, 감속 중에 출력 Pulse 수를 변경하면 올바른 S자 커브를 그릴수 없습니다.
- 원호보간, Bit pattern 보간, 연속보간은 S자 가감속에서는 Drive 할 수 없습니다.
- S자 가감속의 정량 Drive에서는 기동속도를 너무 낮게 설정하면 감속시에 기동속도까지 떨어지기 전에 Drive pulse 를 종료하거나 기동속도까지 도달해도 기동속도로 나머지의 Drive pulse 출력하는 현상이 발생하는 경우가 있습니다.

◎Parameter 설정의 예 (대칭 S자 가감속)

그림과 같이, 기동속도 100pps로 Drive 속도 40kpps까지 0.4초 동안 S자 가속을 하는 예입니다. 가속시에는 일정한 가가속도(k)에 따라 가속도를 직선 증가시키면, 이 때의 적분치(사선의 면적)가 상승하는 속도 형태가 됩니다.

가속 시간($t=0.4\text{sec}$)의 반($t/2$)으로, 정확히 속도가 기동속도(SV)로부터 드라이브 속도(V)의 반($(V-SV)/2$)이 되는 가가속도(k)를 구하는 식은, 좌변의 k를 사용한 사선부의 면적이 우변과 동일한 가를 알아보면 됩니다. k를 요구하는 식은 다음과 같이 됩니다.

$$\frac{k}{2} \left(\frac{t}{2}\right)^2 = \frac{V-SV}{2}$$

$$k = \frac{4(V-SV)}{t^2}$$

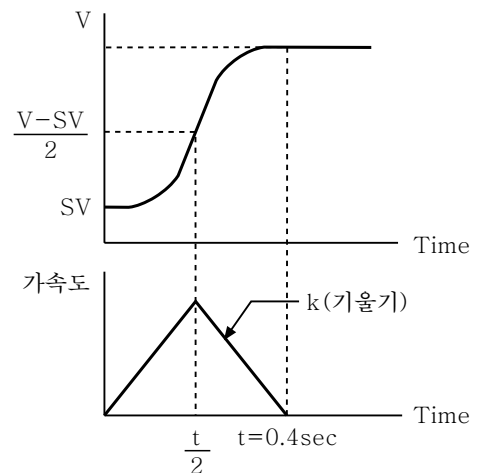
$$k = \frac{4(40000-100)}{0.4^2} = 997,500 \text{ pps/sec}^2$$

가속도 증가율 k : pps/sec²

드라이브 속도 V : pps

기동속도 SV : pps

가속시간 t : sec

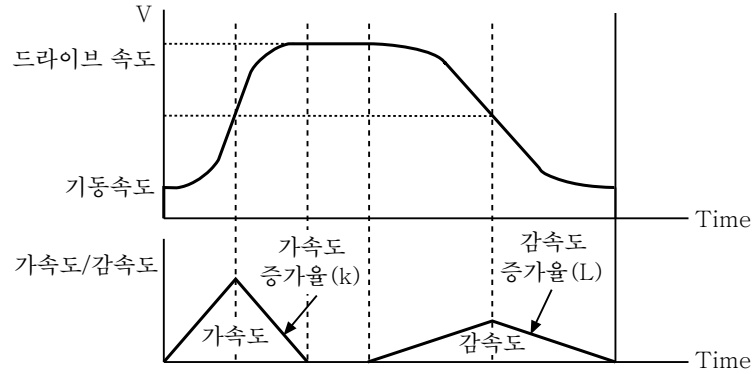


Parameter 설정은, 다음과 같습니다.

WR3 ← 0004h		: WR3 register의 Mode 설정
range	R=800000	: 배율=10
가가속도	K=627	: $(62.5 \times 106/k) \times \text{배율} = (62.5 \times 106/997500) \times 10$
가속도	A=8000	: 최대치에 고정
기동속도	SV=10	: $100/10=10$
drive 속도	V=4000	: $40000/10=4000$
출력 pulse수	P=25000	: 정량 Pulse drive의 경우, 설정합니다.
가속 counter offset	A0=0	

2.2.5 비대칭 S자 가감속 Drive

<그림 2.16>과 같이 S자 가감속 Drive에 대해서 가속시 가가속도와 감속시의 가가속도를 다르게 설정하여 비대칭의 S자 커브를 만들어 낼 수가 있습니다. 정량 Drive의 경우, 대칭 S자 가감속 Drive는 자동으로 감속할 수 없기 때문에 매뉴얼 조작(수동)으로 감속점을 지정할 필요가 있으며 가/감속시의 가가속도, 출력 Pulse 수의 값에 대한 드라이브 속도를 설정할 필요가 있습니다.



<그림 2.16 비대칭형 S자 가감속 Drive>

비대칭형 S자 가감속 Drive를 실행 하려면, nWR3 register의 D2, 1, 0 bit를 아래와 같이 설정합니다.

모드 설정 비트	기호	설정치	설 명
WR3/D0	MANLD	1	매뉴얼 감속
WR3/D1	DSNDE	1	감속시에 가속도, 가속도 증가율 설정치를 사용한다.
WR3/D2	SACC	1	S자 가감 속도

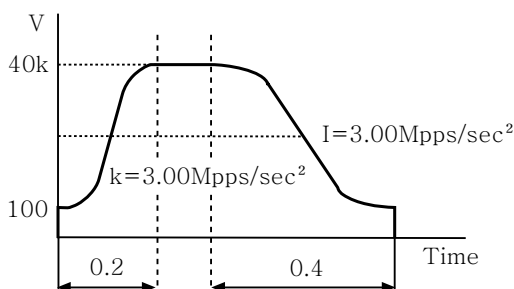
Parameter 설정은 다음과 같습니다.

Parameter 명	기호	설 명
레인지	R	
가속도 증가율	K	
감속도 증가율	L	
가속도	A	반드시 최대치 8000으로 설정한다.
감속도	D	반드시 최대치 8000으로 설정한다.
기동속도	SV	
드라이브 속도	V	
출력 펄스 수	P	연속 펄스 드라이브 시에는 불 필요.
매뉴얼 감속 점	DP	출력 펄스(P)중 감속시에 소비되는 펄스 수를 계산하여 설정한다. 연속 펄스 드라이브시에는 불 필요 .

◎Parameter 설정의 예(비대칭 S자 가감속)

그림과 같이, 가속시에는 기동속도(SV) 100pps로, Drive 속도(V) 40kpps 까지를 0.2초에 가속시키고, 감속시에는 Drive 속도(V) 40kpps에서 기동속도(SV) 100pps 까지 0.4초에 감속시키는 비대칭 S자 가감속의 예입니다.

대칭 S자 가감속 파라미터 설정 예의 식을 사용하여, 가속시의 가가속도, 감속시의 가가속도를 구하면 다음과 같습니다. (배율 : 10)



$$\text{가속도의 증가율 } k = \frac{4(40000-100)}{0.2^2}$$

$$= 3.99\text{Mpps/sec}^2$$

$$\text{감속도의 증가율 } I = \frac{4(40000-100)}{0.4^2}$$

$$= 0.9975\text{Mpps/sec}^2$$

Parameter 설정은 다음과 같습니다.

$$\text{가속도의 증가율 } k = \frac{62.5 \times 10^6}{0.2^2} \times \text{배율} = \frac{62.5 \times 10^6}{3.99 \times 10^6} \times 10 = 157$$

$$\text{감속도의 증가율 } L = \frac{62.5 \times 10^6}{I} \times \text{배율} = \frac{62.5 \times 10^6}{0.9975 \times 10^6} \times 10 = 627$$

비대칭 S자 가감속에서는 자동 감속을 할 수 없기 때문에, 메뉴얼 감속점(DP)을 설정합니다.

메뉴얼 감속점은, 출력 Pulse수(P)로부터 감속시 소비되는 Pulse(Pd)를 설정해야 하므로, 감속시 소비 Pulse(Pd)를 산출합니다.

$$\text{감속시 소비되는 펄스 } Pd = (V+SV) \sqrt{\frac{V-SV}{I}} = (40000+100) \sqrt{\frac{40000-100}{0.9975 \times 10^6}} = 8020$$

출력 Pulse 수를 20000으로 하면, 메뉴얼 감속점(DP)은 다음과 같습니다.

$$\text{메뉴얼 감속점 } DP = P - Pd = 20000 - 8020 = 11980$$

따라서, Parameter 설정은 다음과 같이 됩니다.

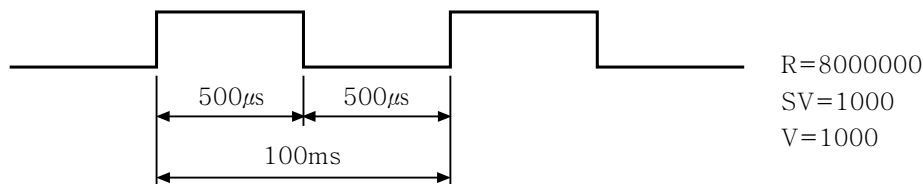
WR3 ← 0007h		: WR3 register의 Mode 설정
Range	R=8000000	: 배율=10
가속시 가가속도	K=157	: $(62.5 \times 106/k) \times \text{배율} = (62.5 \times 106/3.99 \times 106) \times 10$
감속시 가가속도	L=627	: $(62.5 \times 106/L) \times \text{배율} = (62.5 \times 106/0.9975 \times 106) \times 10$
가속도	A=8000	: 최대치에 고정
감속도	D=8000	: 최대치에 고정
기동속도	SV=10	: $100/10=10$
drive 속도	V=4000	: $40000/10=4000$
출력 pulse수	P=20000	
메뉴얼 감속점	DP=11980	
가속 Counter offset	A0=0	

2.2.6 Drive pulse 폭과 속도 정밀도

◎Drive pulse의 Pulse 비율

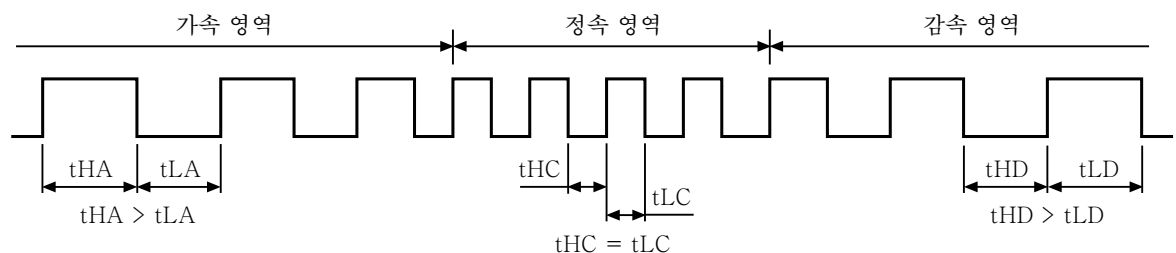
각 축의 +방향/-방향의 Drive pulse에 있어서 Drive 속도에 의해서 결정되는 Pulse 주기의 시간은 연산상의 오차 $\pm 1\text{SCLK}$ (CLK=16MHz 때 $\pm 125\text{ nsec}$)이지만 기본적으로 Hi level과 Low level에 50%씩 나누어집니다.

예를 들면, 아래 그림과 같이 R=8000000, V=1000(배율=1, Drive 속도=1000pps)으로 설정하면, Drive pulse는 Hi level 폭=500 μ s, Low level 폭=500 μ s, 주기=1.00ms의 Pulse를 출력합니다.



<그림 2.17 Drive pulse 출력의 Hi/Low level 폭(V=1000pps)>

그러나, 가감속 Drive의 가속시에 1개의 Drive pulse를 출력하고 있는 동안에도 Drive 속도는 상승해가기 때문에 Low level의 pulse 폭이 Hi level보다 짧아집니다. 반대로, 감속시에는 Low level의 Pulse 폭이 Hi level보다 길어집니다.



<그림 2.18 가감속 Drive시 Drive pulse 폭 비교>

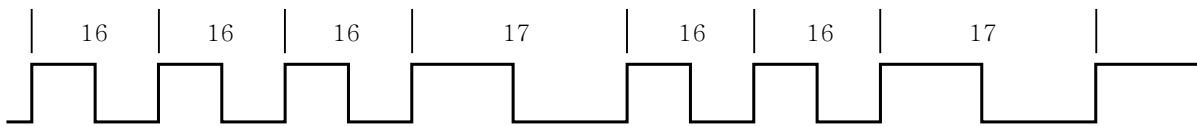
◎Drive 속도의 정도

Drive pulse를 생성하는 회로는, 모두 입력 클럭 신호(CLK)를 내부에서 2분주한 SCLK으로 동작하고 있습니다. CLK 입력이 표준인 16MHz이면, SCLK은 8MHz가 됩니다.

어떤 주파수의 Drive pulse를 Jitter가 없는 균일한 주파수의 Drive pulse로 만들려고 하면, SCLK 주기의 정수배의 주기를 가진 주파수 밖에 만들수 없습니다.

예를 들면, 2배:4.000 MHz, 3배:2.667 MHz, 4배:2.000 MHz, 5배:1.600 MHz, 6배:1.333 MHz, 7배:1.143 MHz, 8배:1.000 MHz, 9배:889 KHz, 10배:800 KHz,의 주파수 밖에 출력하지 못하고, 임의의 Drive 속도를 설정할 수가 없게 됩니다. 그래서 다음과 같은 방식에 의해서 임의의 Drive 속도를 출력하고 있습니다.

예를 들면, Range 설정치:R=80,000(배율=100), Drive 속도 설정치:V=4900 으로 하면, $4900 \times 100 = 490\text{kpps}$ 의 Drive 펄스 출력하지만, 이 주기는 SCLK의 주기의 정수배가 아니기 때문에, 균일한 주파수로 490kpps를 출력할 수 없습니다. <그림 2.19>와 같이 SCLK의 16정수배의 500kpps의 주파수와 17정수배의 471kpps의 주파수를 합성해 출력하고 습니다. 490kpps의 주기는, SCLK(8MHz)의 주기의 16.326배이므로, SCLK의 16배 주기의 Pulse와 17배 주기의 Pulse를 674:326의 비율로 출력하여, 단위 시간 당의 평균 주기가 16.326이 되도록 하고 있습니다.



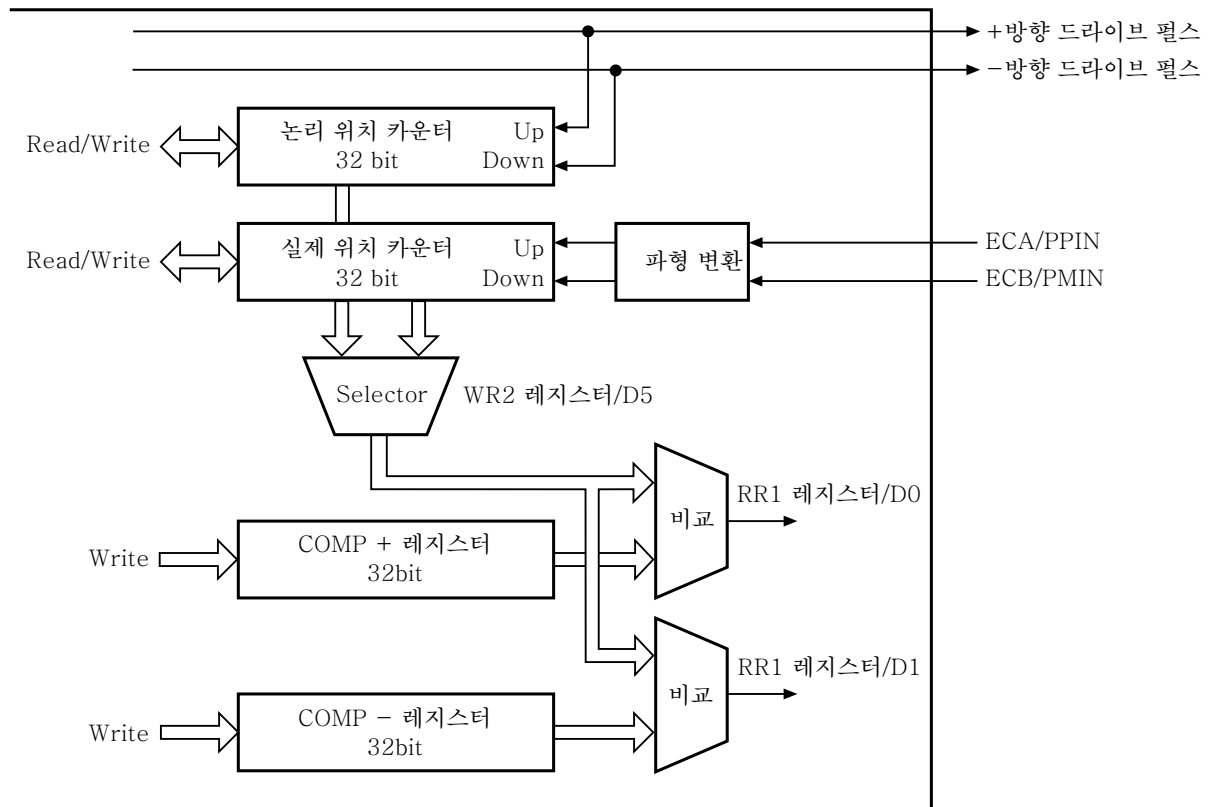
<그림 2.19 SCLK 주기에 대한 490kpps Drive pulse의 주기>

이 방식에 의해, 지정된 속도의 Drive pulse의 정도를 좋게 출력할 수가 있습니다. 속도 배율을 올려도, 지정된 속도에 대한 실제로 출력되는 Drive pulse의 속도 정도는 $\pm 0.1\%$ 이하로 유지됩니다.

Drive pulse를 오실로스코프(Oscilloscope)로 관측하면, Drive pulse의 주기가 SCLK 주기의 정수배가 아닐 때에는 <그림 2.19>와 같이 Pulse 주기에 1SCLK(125 nsec)의 시간차이가 생기기 때문에 이것이 Jitter와 같이 보이지만 1SCLK의 시간차이는 모터를 돌리는 경우 부하의 관성에 흡수되어 대부분 문제가 되지 않습니다.

2.3 포지션(Position) 관리

<그림 2.20>은 1축에 해당하는 포지션 관리부의 블록도입니다. 각 축 모두 현재 위치 관리를 위한 32 bit Up/Down counter 2개와 현재 위치의 대소 비교를 위한 비교 Register 2개를 가지고 있습니다.



<그림 2.20 포지션 관리부 블록도>

2.3.1 논리위치 Counter와 실제위치 Counter

논리위치 Counter는 <그림 2.20>과 같이 +방향/-방향의 Drive출력 Pulse를 Count 합니다.

+방향 1Pulse로 1Count up, -방향 1Pulse로 1Count down 합니다.

실제위치 Counter는 Encoder등의 외부입력 Pulse를 Count 합니다. 입력 Pulse를 2상 펄스 입력 신호로 할지, 독립 2Pulse(Up/Down) 펄스 입력 신호로 할지를 Command로 선택할 수가 있습니다. 2.9.3절을 참조 하십시오.

Counter는 Data의 읽기/쓰기가 가능하며 Count 범위는 -2,147,483,648 ~ +2,147,483,647 입니다.

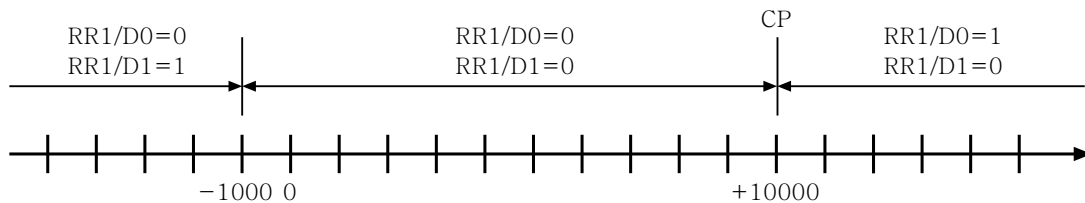
2.3.2 비교 Register와 소프트웨어 리미트

각 축은 <그림 2.20>과 같이 논리위치 Counter 또는 실제위치 Counter와 대소 비교를 할 수 있는 2개의 32bit 레지스터(COMP+, COMP-)를 가지고 있습니다. 2개의 비교 Register의 비교 대상을 논리위치 Counter로 할지, 실제위치 Counter로 할지는 WR2 register의 D5(CMPSEL) bit로 선택합니다.

COMP+ register는 논리/실제위치 Counter에 대해서, 어느 범위의 상한을 검출하기 위한 Register로서, 논리/실제위치 Counter의 값이 COMP+ register의 값보다 커지면 RR1 register의 D0(CMP+) bit가 1이 됩니다.

COMP- register는 논리/실제위치 Counter에 대해서 어느 범위의 하한을 검출하기 위한 Register로서, 논리/실제위치 Counter가 COMP- register의 값보다 작아지면 RR1 register의 D1(CMP-) bit가 1이 됩니다.

<그림 2.21>은 COMP + register=10000, COMP - register=-1000을 설정한 예입니다.



<그림 2.21 COMP +/- register 설정 예>

COMP+ register와 COMP- register를 각각 +방향/-방향의 소프트웨어 Limit로서 사용할 수 있으며 WR2 register의 D0, D1(SLMT+, SLMT-) bit를 1로 소프트웨어 Limit를 유효하게 하면, Drive 중에 논리/실제위치 Counter가 COMP+ register보다 커지면 감속정지하여, RR2 register의 D0(SLMT+) bit가 1이 됩니다. 이 에러는 -방향의 Drive 명령을 실행하여, 논리/실제위치 Counter가 COMP+ register 보다 작아지면 해제됩니다. COMP- register의 -방향에 대해서도 같습니다.

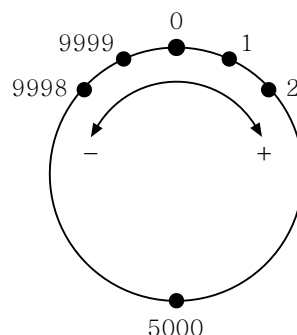
COMP+ register와 COMP- register는 언제든지 설정이 가능합니다. Reset후 내용은 적용되지 않습니다.

2.3.3 위치 Counter의 가변 링

논리위치 Counter 및 실제위치 Counter는 32 bit 의 Up/Down counter입니다.

따라서 일반적으로, 32 bit의 최대치로 FFFFFFFFh에서 +방향으로 count up하면 0으로 돌아옵니다.

또, 0의 값에서 -방향으로 Count down 하면 FFFFFFFFh로 돌아옵니다. 가변 링 기능은 이 링 Counter의 최대치를 임의의 값으로 설정하는 기능입니다. 위치결정축이 직선운동은 아니고, 회전운동을 하는 축의 위치 관리를 하는 디바이스에 편리한 기능입니다. 가변 링 기능을 유효하게 하려면, 확장 Mode 설정 명령(60h)의 WR6 register / D4(VRING) bit를 1로 설정하여 논리위치 Counter의 최대치를 COMP+ register에 실제위치 Counter의 최대치를 COMP- register에 설정합니다.



<그림 2.22 위치 Counter 링 최대치 9999의 동작>

10,000 pulse로 1회 회전하는 회전축의 경우, 다음과 같이 설정합니다.

- ①가변 링 기능을 유효하게 하기 위해서 확장mode 설정 명령(60h)의 WR6/D4bit에 1을 설정.
 - ②논리위치 Counter의 최대치로서 COMP+ register에 9,999(270Fh)를 설정.
 - ③실제위치 Counter도 사용하는 경우는 COMP- register에 9,999(270Fh)를 설정.
- 이 때의 Count 동작은, +방향에 Count up 시에는 ... → 9998 → 9999 → 0 → 1 → ... 이 됩니다.
-방향에 Count down 시에는 ... → 1 → 0 → 9999 → 9998 → ... 이 됩니다.

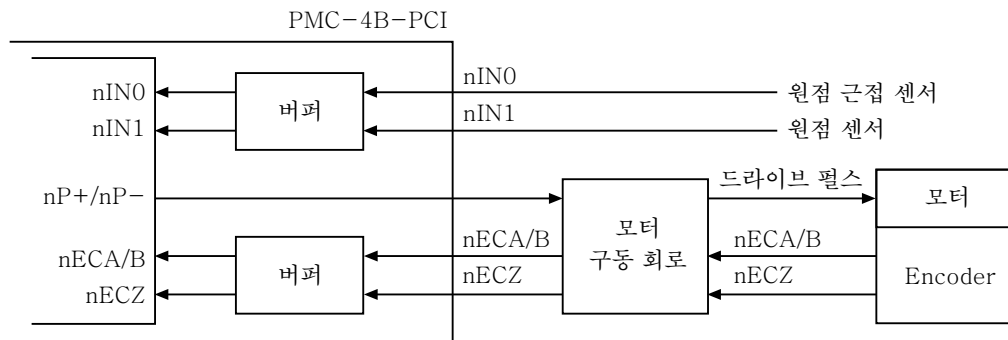
【주의】

- 가변 링 기능의 유효/무효는 각 축 마다 설정가능 합니다. 논리위치 Counter와 실제위치 Counter는 각각 개별적으로 유효/무효로 설정할 수 없습니다.
- 가변 링 기능을 유효하게 하면 소프트 리미트 기능은 사용할 수 없습니다.

2.3.4 외부 신호에 의한 실제위치 Counter의 Clear

원점출력에 있어서 Z상 서치를 실행하면 Z상 신호가 Active되는 Level의 시작에서 실제위치 카운터를 Clear시키는 기능입니다. 일반적으로, 원점복귀는 원점근접신호, 원점신호, Encoder Z상 신호 등을 nIN0, nIN1, nECZ 단자로 접속하여 연속 Drive를 실행하는 것으로써 실시합니다. 지정된 신호가 Active되면 Drive가 정지하기 때문에 논리위치/실제위치 Counter를 Clear 합니다. Z상 서치는 저속의 Drive 속도로 서보계 혹은 기계계의 지연으로부터 발생하는 Z상 검출의 위치 차이가 문제가 되는 경우에 본 기능을 사용하면 편리합니다.

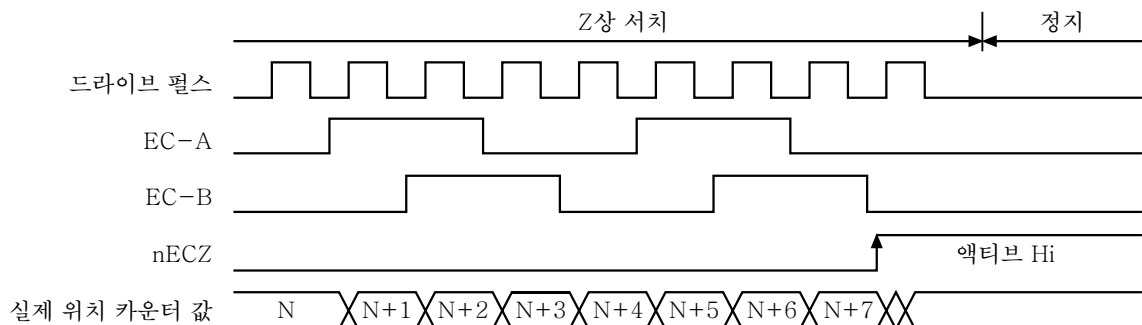
Encoder Z상 서치 시에 Z상 신호에서 실제위치 Counter를 Clear 하려면, <그림 2.23>과 같이 Z상 신호를 nECZ 신호에 연결하고 실제위치 Counter clear를 동반하는 Z상 서치의 Mode 및 Command를 다음과 같이 설정합니다.



<그림 2.23 IECZ 신호에 의한 실제 위치 Counter clear 신호의 접속 예>

- ①Range, 기동속도를 설정.
- ②Z상서치의 Drive 속도를 설정. Drive 속도를 기동속도보다 낮은 값으로 설정하면 가감속 Drive는 실행되지 않으므로 Z상을 검출하면 Drive pulse는 즉시 정지합니다.
- ③nECZ 신호의 유효와 Active level을 설정.
WR1/D5(IN2-E) : 1
D4(IN2-L) : 0(Low active), 1(Hi active)
- ④nECZ 신호에 의한 실제위치 Counter clear를 유효하게 설정.
WR6/D0(EPCLR)를 1로 설정하여 확장 Mode 설정명령(60h)을 발행.
【주의】 확장 Mode 설정명령의 다른 bit도 동시 설정됩니다.
- ⑤ +방향 또는 -방향 연속 Pulse drive 명령을 발행.

이상의 조작을 실시하면 <그림 2.24>와 같이 지정된 방향으로 Drive를 시작하여, Z상 신호가 Active level이 되면 Drive pulse가 정지됨과 동시에 실제위치 Counter는 Z상신호 Active level의 시작에서 Clear 됩니다.



<그림 2.24 IN2 신호에 의한 실제위치 Counter clear의 동작 예>

【주의】

- 실제위치 Counter를 Clear 할 수 있는 신호는 nECZ 신호입니다. nIN3, nIN1, nIN0 신호에서는 Clear 할 수 없습니다.
- nECZ 신호의 Active level 폭은 입력신호 필터가 무효인 경우, 4CLK 사이클 이상 필요합니다.
입력신호 필터가 유효인 경우는 입력신호 지연의 배 이상의 시간이 필요합니다.
- Z상 서치는 위치검출 정도를 올리기 위해서 반드시 한 방향으로 검출하는 것을 추천 합니다.
- 실제위치 Counter clear 기능을 유효하게 하기 위해서, WR6/D0(EPCLR)를 1로 합니다.
그리고 이미 nECZ신호가 Active level이 되어 있는 경우는 확장 Mode 설정 명령을 발행한 시점에서 실제위치 Counter가 Clear 됩니다.

2.4 보간

4축 중 임의의 2축, 또는 3축을 선택하여 직선보간, 원호보간, Bit pattern 보간 Drive가 가능합니다.

보간을 실시하는 축 지정은 WR5 register의 D0, D1(ax1), D2, D3(ax2), D4, D5(ax3)에 축 코드를 설정합니다. 보간 Drive에서는 주축(ax1)으로 지정된 축의 기본 Pulse 타이밍에 보간 연산을 합니다. 따라서, 보간 명령을 발행하기 전에 주축(ax1)의 기동속도, Drive 속도 등의 Parameter가 설정되어 있어야 합니다.

주축이란 ax1로 지정된 축을 말하며 각각의 축으로 보간 명령에 필요한 Parameter를 설정하여 보간 Drive 명령을 WR0 Command register에 기입하면 보간 드라이브는 시작됩니다. 보간 Drive 중에 RR0(주 Status register)의 D8(I-DRV) bit가 1이 되어 Drive가 끝나면 0으로 돌아옵니다.

또한, 보간 Drive 중, 보간을 실시하고 있는 축의 n-DRV bit에도 1이 설정됩니다. 보간 연산은 직선보간, 원호보간, Bit pattern 보간 모두 최고 4Mpps까지 실시할 수 있습니다. 다만, 연속보간 일 경우에는 최고 2Mpps 까지 입니다.

◎보간 시의 Over run limit 등의 에러

보간 Drive에 있어서 Drive하는 각 축의 하드웨어 Limit, 소프트웨어 Limit는 항상 작동합니다.

즉, 보간 Drive 중 해당 축의 Limit가 Active되면 보간 Drive는 정지합니다. 에러로 정지했을 경우는, RR0(주 Status register)의 보간이 지정되고 있는 축의 에러 Bit를 확인하여 1로 설정되면 그 축의 RR2(에러 Register)를 읽을 수 있습니다.

【주의】 원호보간, 및 Bit pattern보간에서는 +방향/-방향 중, 어느 방향의 하드웨어 Limit 및 소프트웨어 Limit가 Active 되어도 보간이 정지하는 경우가 있습니다. 따라서 원호보간 및 Bit pattern 보간에서는 Limit 영역으로 부터 탈출할 수 없기 때문에 주의해 주십시오.

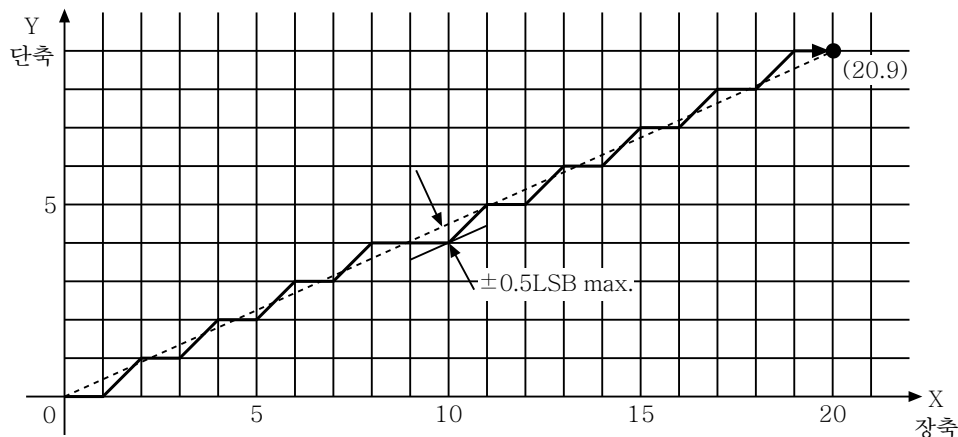
◎서보 모터용 위치결정 신호의 대응

보간 Drive에 있어서 Drive하는 각 축의 위치 결정신호(nINPOS)를 유효하게 하면 보간 Drive 종료 후, 모든 축의 nINPOS신호가 Active level이 되는 것을 확인하고 나서 RR0 register의 D8(I-DRV) bit가 0으로 돌아옵니다.

2.4.1 2축/3축 직선보간

4축 중, 임의의 2축, 또는 3축을 선택하여 직선보간 Drive를 실시합니다. 직선보간은 현재 좌표에 대한 종점 좌표를 설정하여 2축, 또는 3축 직선보간 명령을 발생하여 실행됩니다. <그림 2.25>는 2축 보간의 예입니다.

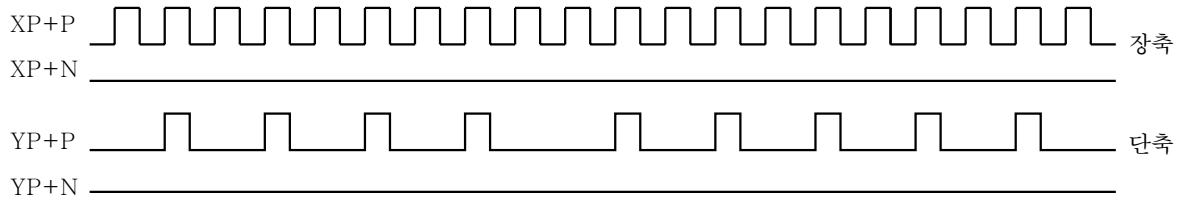
종점좌표는 현재 위치에 대한 상대치로 각축의 출력 Pulse 수로 설정합니다. 출력 Pulse 수는 각 축 독립으로 움직일 때는, 부호 없는 값으로 설정하지만 보간 Drive 시에는 현재 위치에 대한 종점좌표를 상대치로 설정하기 때문에 주의해 주십시오. 지정 직선에 대한 위치정도는 <그림 2.25>와 같이 전 보간 범위 내에서 ± 0.5 LSB입니다.



<그림 2.25 직선 보간의 위치 정밀도>

<그림 2.26>은, 직선보간의 Drive pulse출력 예 입니다. 설정된 종점의 값중 절대치가 큰 축이 장축이 되고 보간 Drive 중에는 항상 Pulse를 출력합니다. 다른 축은 단축 되어 직선보간 연산 결과에 의해 Pulse를 발생할 때, 발생하지 않을 때가 있습니다. 직선보간의 좌표 범위는 부호를 첨부하여 32 bit입니다.

각 축 모두 현재 위치로부터 -2,147,483,646 ~ +2,147,483,646의 범위에서 보간을 할 수 있습니다.

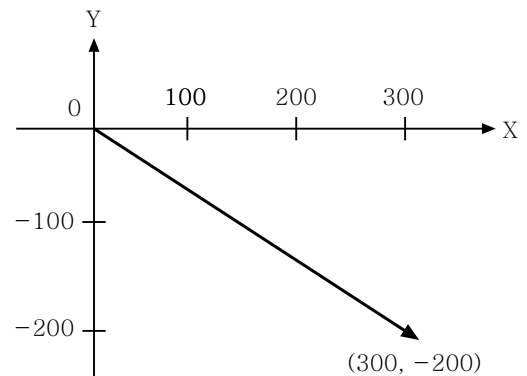


<그림 2.26 종점(X:20, Y:9)의 Drive pulse 출력의 예>

◎2축 직선보간 Drive의 예

X, Y축에 대하여 현재 위치에서 종점좌표(X:+300, Y:-200)까지 직선보간 합니다. 보간 Drive 속도는 1000pps의 정속 Drive입니다.

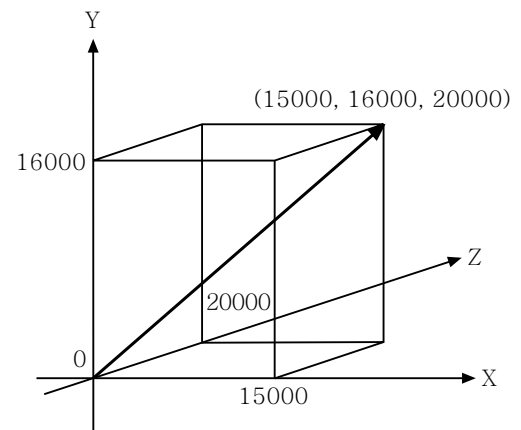
```
WR5 ← 0004h write ; ax1:X축, ax2:Y축 지정
WR6 ← 1200h write ; range : 8,000,000(배율:1)
WR7 ← 007Ah write
WR0 ← 0100h write
WR6 ← 03E8h write ; 기동속도 : 1000pps
WR0 ← 0104h write
WR6 ← 03E8h write ; drive 속도 : 1000pps
WR0 ← 0105h write
WR6 ← 012Ch write ; 종점 X축 : 300
WR7 ← 0000h write
WR0 ← 0106h write
WR6 ← FF38h write ; 종점 Y축 : -200
WR7 ← FFFFh write
WR0 ← 0206h write
WR0 ← 0030h write ; 2축 직선보간 Drive
```



◎3축 직선보간 Drive의 예

X, Y, Z축에 대하여 현재 위치에서 종점 좌표(X:15000, Y:16000, Z:20000)까지 3축 직선보간 합니다. 기동속도:500pps, 가감 속도:40,000pps/sec, Drive속도:5,000pps의 직선 가감속 Drive입니다.

```
WR5 ← 0024h write ; ax1:X축, ax2:Y축, ax3:Z축 지정
WR6 ← 1200h write ; range : 8,000,000(배율:1)
WR7 ← 007Ah write
WR0 ← 0100h write
WR6 ← 0140h write ; 가감속도 : 40,000pps/sec
WR0 ← 0102h write ; 40000/125/1 = 320
WR6 ← 01F4h write ; 기동속도 : 500pps
WR0 ← 0104h write
WR6 ← 1388h write ; drive 속도 : 5000pps
WR0 ← 0105h write
WR6 ← 3A98h write ; 종점 X : 15,000
WR7 ← 0000h write
WR0 ← 0106h write
WR6 ← 3E80h write ; 종점 Y : 16,000
WR7 ← 0000h write
WR0 ← 0206h write
WR6 ← 4E20h write ; 종점 Z : 20,000
WR7 ← 0000h write
WR0 ← 0406h write
WR0 ← 003Bh write ; 감속 유효
WR0 ← 0031h write ; 3축 직선보간 Drive
```

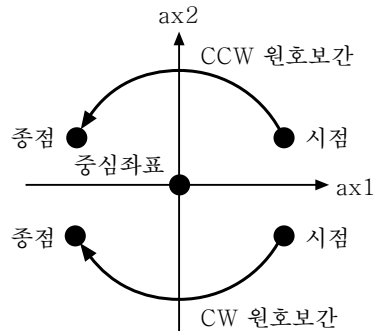


2.4.2 원호보간

4축 중, 임의의 2축을 선택하여 원호보간 Drive를 실시합니다. 원호보간은 현재 좌표에 대한 원호의 중심 좌표 및 종점 좌표를 설정하여 CW 원호보간 명령 또는, CCW 원호보간 명령을 기입하는 것으로 실행됩니다.

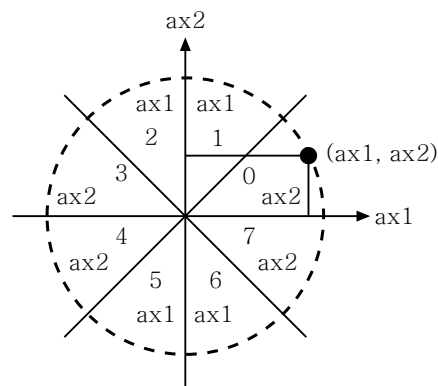
중심좌표 및 종점좌표의 지정은 현재좌표에 대한 상대치로 설정해야 합니다.

CW 원호보간은 현재 좌표에서 종점좌표까지 중심좌표를 중심으로 시계방향으로 실시하고, CCW 원호보간은 반시계 방향으로 원호를 그립니다. 종점을 (0, 0)으로 하면 진원을 그릴 수가 있습니다.

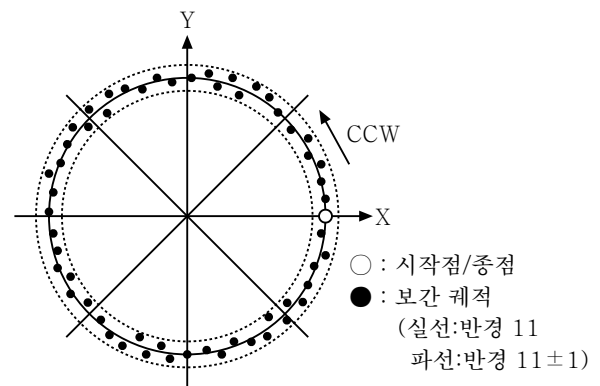


<그림 2.27 CW/CCW 원호보간>

PMC-4B-PCI의 원호보간의 연산에서는 <그림 2.28>과 같이 제 1축(ax1)과 제 2축(ax2)에 의한 평면의 중심 좌표를 중심으로 0~7, 총 8개의 상한으로 나누고 있습니다. 그림에서 0상한에서는 원호상을 이동하는 보간 좌표(ax1, ax2)는 항상 ax2의 절대치가 ax1의 절대치보다 작습니다. 절대치 값이 작은 축을 단축으로 1, 2, 5, 6 상한에서는 제 1축(ax1)이 단축이 되고, 0, 3, 4, 7 상한은 제 2축(ax2)이 단축이 됩니다.



<그림 2.28 원호보간 연산의 0~7 상한과 단축>



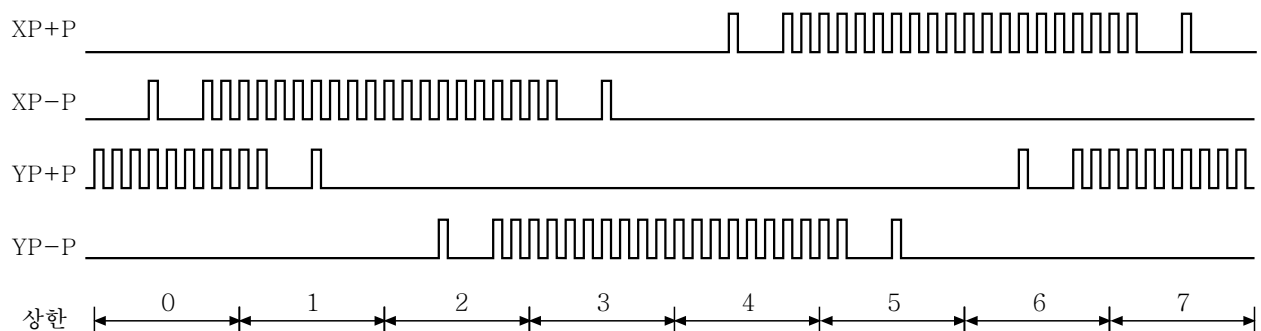
<2.29 원호 보간>

<그림 2.29>는 현재 좌표 중심 (-11, 0), 종점(0,0)을 지정으로, 반경 11의 진원을 그린 예입니다.

또, <그림 2.30>은 이 때의 Drive pulse 출력을 나타냅니다.

중심좌표 및 종점좌표의 지정범위는 현재 위치로부터 -2,147,483,646 ~ +2,147,483,646입니다.

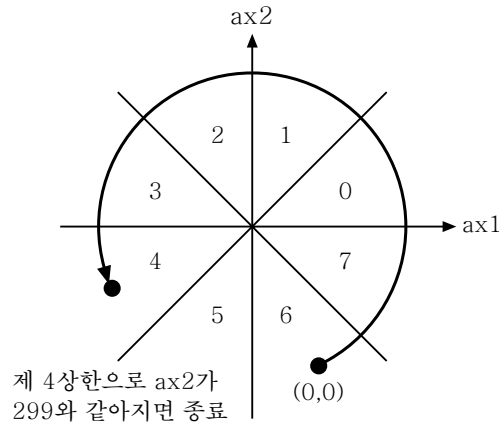
원호에 대한 위치오차는 전 보간 범위 내에서 $\pm 1\text{LSB}$ 입니다. 보간속도는 1pps ~ 4Mpps 입니다.



<그림 2.30 원호보간 Drive pulse 출력의 예>

◎중점 판정

원호보간은 보간 Drive를 시작하기 전, 현재좌표와 중심좌표로 설정되어 반경이 결정되면 원호의 궤적을 그립니다. 원호 연산의 오차는 보간 좌표 범위 안에서 ± 1 LSB이기 때문에 지정된 중점이 반드시 원호의 궤적상에 있다고 할 수 없습니다. 그래서 중점이 있는 상한에 대해서 중점의 단축의 값과 비등하게 되었을 때 원호보간 종료라고 판단하고 있습니다. <그림 2.31>은 현재위치(0,0), 중심(-200, 500)으로 중점(-702, 299)이 결정되어 CCW원호보간의 예입니다. 현재위치(0,0)과 중심(-200, 500)으로부터 정해지는 반경에 의해 CCW방향으로 보간을 실시하여 지정된 중점(-702, 299)은 4상한으로 제 2축(ax2)이 단축이 되고 중점값(-702, 299)에서 제 2축의 299에 이르렀을 때에 보간 종료라고 판단합니다.



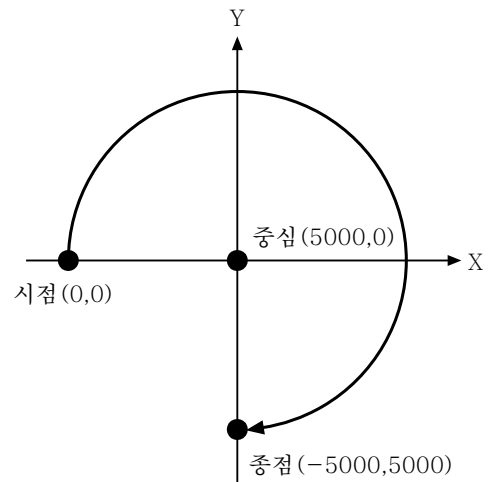
<그림 2.31 원호보간 종료 판정의 예>

◎CW원호보간 Drive의 예

X, Y축에 대하여 현재위치(시점:0,0)로부터 중심 (X:5000, Y:0), 중점(X:5000, Y:-5000)에서 CW 원호보간을 합니다. 보간 드라이브 속도는 1000pps의 정속 Drive를 실시하며 선속일정 Mode로 보간 합니다.

```

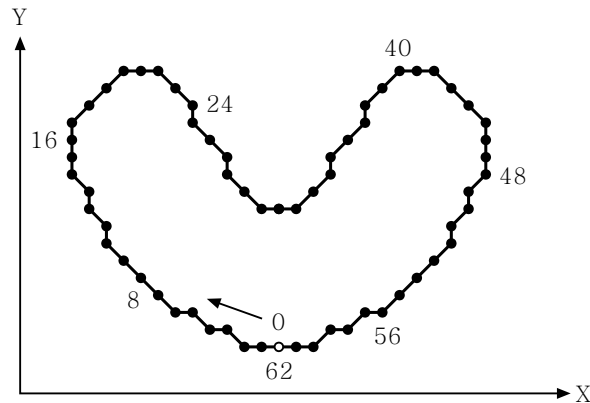
WR5 ← 0104h write ; ax1:X축, ax2:Y축 지정, 선속일정
WR6 ← 0900h write ; range : 4,000,000(배율:2)
WR7 ← 003Dh write
WR0 ← 0100h write
WR6 ← 4DC0h write ; 2축 선속 일정을 위한 Range: 4,000,000 × 1.414 = 5,656,000
WR7 ← 0056h write ;
WR0 ← 0200h write
WR6 ← 01F4h write ; 기동속도 : 500 × 2 = 1000pps
WR0 ← 0104h write
WR6 ← 01F4h write ; drive 속도 : 500 × 2 = 1000pps
WR0 ← 0105h write
WR6 ← 1388h write ; 중심 X : 5000
WR7 ← 0000h write
WR0 ← 0108h write
WR6 ← 0000h write ; 중심 Y : 0
WR7 ← 0000h write
WR0 ← 0208h write
WR6 ← 1388h write ; 중점 X : 5000
WR7 ← 0000h write
WR0 ← 0106h write
WR6 ← EC78h write ; 중점 Y : -5000
WR7 ← FFFFh write
WR0 ← 0206h write
WR0 ← 0032h write ; CW원호보간 Drive
    
```



2.4.3 2축/3축 Bit pattern보간

상위 CPU에서 작성한 Bit pattern화 된 보간 Data를 Packet(결정된 양의 Data 결정체)으로 받아, 지정된 드라이브 속도로 보간 Pulse를 연속적으로 출력하는 보간 Drive입니다. Bit pattern보간에서는 2축 또는, 3축의 +방향, -방향의 드라이브 Pulse를 1bit, 1pulse로 각 각의 Register에 설정합니다. Drive pulse를 출력할 때는 "1", 출력하지 않을 때는 "0"으로 설정합니다.

예를 들면, <그림 2.32>와 같은 궤적을 그리는 경우, X+방향, X-방향, Y+방향, Y-방향 각각의 Drive pulse를 발생할 때는 "1", 발생하지 않을 때는 "0"으로 하면 Bit pattern data는 아래와 같이 됩니다.



<그림 2.32 Bit pattern 도형의 예>

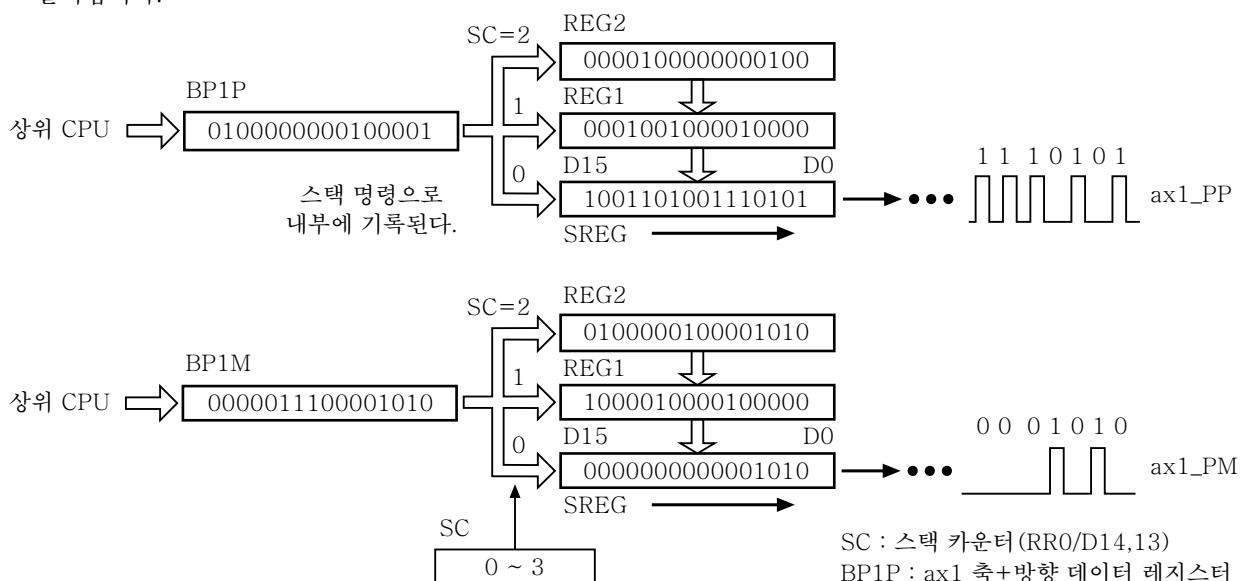
← 56 ← 48 ← 40 ← 32 ← 24 ← 16 ← 8 ← 0

01000000 00000000 00011111 11011011 11110110 11111110 00000000 00000000 : XPP(X+방향)
 01111111 11110101 00000000 00000000 00000000 00000000 00101011 11111111 : XPM(X-방향)
 00000000 00000000 00000000 11111111 00000000 00001111 11111111 11010100 : YPP(Y+방향)
 00001010 11111111 11111100 00000000 00111111 11000000 00000000 00000000 : YPM(Y-방향)

<그림 2.33>은 Bit pattern보간의 제 1축의 Register 구성과 bit data의 움직임을 나타내고 있습니다.

BP1P register, BP1M register는 상위 CPU로부터 Bit pattern data를 기입하는 16 Bit register입니다.

(8bit bus의 경우는 L byte, H byte로 나누어 기입합니다.) +방향의 16 bit의 bit data는 BP1P register에, -방향의 data는 BP1M register에 기입합니다. Bit pattern 보간이 시작되면 D0부터 차례대로 Drive pulse를 출력합니다.



<그림 2.33 Bit pattern보간의 Register구성과 bit data의 움직임(ax1축)>

SC : 스택 카운터(RR0/D14,13)
 BP1P : ax1 축+방향 데이터 레지스터
 BP1M : ax1 축-방향 데이터 레지스터
 SREG : 16 비트 시프트 레지스터
 REG1 : 16 비트 시프트 레지스터
 REG2 : 16 비트 시프트 레지스터

Stack Counter(SC)는 Bit pattern data를 Count하는 Counter로 0에서 3까지 변화합니다.

RR0 register의 D14, D13 bit가 stack counter의 값을 나타내고 있습니다. BP1P, BP1M register에 기입된 Data는 BPdata stack 명령에 의해 내부 16bit shift register(SREG), 또는 2개의 16 bit register(REG1, REG2)의 어느 쪽인가에 기입해집니다. 이 때, Stack counter SC=0 일 때는 SREG에, SC=1 때는 REG1에, SC=2 때는 REG2에 기입해집니다. Data가 모두 기입되면 stack counter(SC)는 1개 증가합니다.

2축 또는 3축 bit pattern 보간 명령에 의해, Bit pattern 보간이 시작되면 모든 축은 주축의 기본 펄스에 동기되어 16 bit shift register(SREG)의 D0 bit의 값에 의해 Drive pulse를 출력합니다.

D0의 값이 "1"일 때는 Drive pulse가 출력되고 "0"일 때는 출력되지 않습니다. Shift register의 16 bit가 모두 출력을 마치면 REG1 register의 Data가 shift register에, REG2 register의 Data가 REG1 register로 옮겨져 Stack Counter(SC)가 하나 감소합니다.

상위 CPU는, Stack counter(SC)가 3이 되면 더 이상 Bit pattern data를 내부에 Stack 할 수 없습니다.

하지만, 보간 Drive가 시작되면 Drive pulse의 출력에 따라, Stack counter(SC)의 값이 3→2→1으로 감소하기 때문에 다시 Data를 기입할 수가 있습니다. Stack counter(SC)=0는 보간 Drive 종료를 의미하기 때문에 연속하여 Bit pattern을 보간 하는 경우는 SC=2또는 1사이에 다음의 Data를 설정하지 않으면 안되기 때문에 SC의 값이 2에서 1로 바뀌었을 때 상위 CPU에 대해서 Interrupt를 발생하여 Data 기입을 요구합니다.

◎보간 Drive 속도의 제한

Bit pattern보간의 Drive 속도는 최고 4MHz까지 가능합니다. 그러나 bit 수가 48 bit를 넘는 경우에는 보간 Drive 중에 Data를 보충해 나가지 않으면 안되기 때문에, 보간 Drive 속도는 CPU의 Pattern data의 Setup에 필요로 하는 시간에 의존하게 됩니다.

예를 들면, 2축 Bit pattern보간으로, CPU측이 4×16 bit의 연산과 Data set, 및 BP data stack 명령의 발행으로 100 μsec 걸린다고 하면, 보간 Drive 속도는 1/(100 μsec/16)=160kpps 이하의 속도입니다.

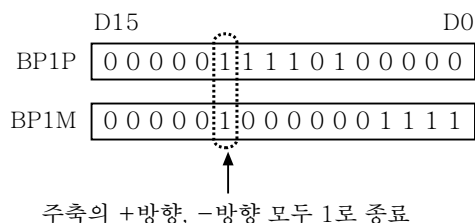
◎Bit Pattern 보간의 종료

Bit Pattern 보간에는 다음과 같이 2가지 방법으로 종료합니다.

①제 1축 Data에 종료 코드를 기입한다.

주축(ax1)의 +방향, -방향의 Bit data를 동시에 "1"로 하면, Bit pattern 보간 종료라고 판단합니다.

종료 코드를 검출하면 Stack Counter(SC)는 강제적으로 0이 되어, 이후에 Stack 된 Bit pattern data는 모두 무효가 됩니다.



②Data 기입을 중지한다.

BPdata stack 명령에 의한 내부 Register의 Bit pattern data의 기입을 중지하면 모든 Bit pattern data를 Drive pulse로서 출력하여 SC=0이 되어 보간 Drive를 종료합니다.

◎정지 명령에 의한 보간 Drive 중단

Bit pattern보간 drive를 실시하고 있는 주축(ax1)에 즉시정지 명령, 혹은 감속정지 명령을 설정하면 보간 중에 Drive는 정지합니다. 여기서 다시 Bit pattern보간 명령을 설정하면 Bit pattern보간을 계속할 수 있습니다.

정지 명령에 의해 Drive를 정지하여 보간을 종료하는 경우는 반드시 BP data clear 명령에 의해 설정되어 있는 Data를 모두 clear 해야 합니다.

◎하드웨어 Limit, 소프트웨어 Limit에 의한 정지

보간 Drive 중에 어느 축의 하드웨어 Limit 또는, 소프트웨어 Limit가 Active 되어도 보간 Drive는 정지합니다.

그대로 보간을 종료하는 경우에는 반드시, BP data clear 명령에 의해 기입되어 있는 Data를 모두 Clear 해야 합니다.

Bit pattern 보간에서는 +방향/-방향 중, 어느 방향의 하드웨어 Limit 및 소프트웨어 Limit가 Active되어 보간이 정지하는 경우가 있습니다. Bit pattern 보간에서는 Limit over 영역으로부터 탈출 할 수 없으므로 주의해 주십시오.

◎Bit pattern data기입 register

16 bit bus 및 8 bit bus에 의한 ax1축 부터 ax3축의 Bit pattern data 기입 Register의 Address를 각각 아래 표에 나타냅니다.

16bit data bus의 Bit pattern data 기입 Register address

Address			Register	내 용	응답 Register
A2	A1	A0			
0	0	0			WR0
0	0	1			nWR1
0	0	0	BP1P	ax1 축 +방향 Data register	nWR2
0	0	1	BP1M	ax1 축 -방향 Data register	nWR3
1	1	0	BP2P	ax2 축 +방향 Data register	WR4
1	1	1	BP2M	ax2 축 -방향 Data register	WR5
1	1	0	BP3P (주1)	ax3 축 +방향 Data register	WR6
1	1	1	BP3M (주1)	ax3 축 -방향 Data register	WR7

※ (주1) : BP3P, BP3M은 각각 WR6, 7 레지스터와 공용입니다.

8bit data bus의 Bit pattern data 기입 Register address

Address				Register	Address				Register
A3	A2	A1	A0		A3	A2	A1	A0	
0	0	0	0		1	0	0	0	BP2PL
0	0	0	1		1	0	0	1	BP2PH
0	0	1	0		1	0	1	0	BP2ML
0	0	1	1		1	0	1	1	BP2MH
0	1	0	0	BP1PL	1	1	0	0	BP3PL
0	1	0	1	BP1PH	1	1	0	1	BP3PH
0	1	1	0	BP1ML	1	1	1	0	BP3ML
0	1	1	1	BP1PH	1	1	1	1	BP3MH

BPmPL, BPmPH, BPmML, BPmMH는 각각 아래와 같은 byte를 나타냅니다. (m는 1~3)

BPmPL : BPmP의 하위 byte(D7~D0)

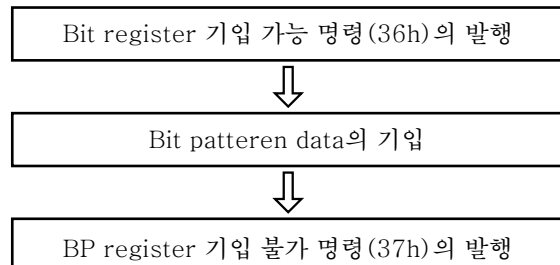
BPmPH : BPmP의 상위 byte(D15~D8)

BPmML : BPmM의 하위 byte(D7~D0)

BPmMH : BPmM의 상위 byte(D15~D8)

Bit pattern data기입 register는, nWR2~WR7 register와 같은 Address입니다.

Reset 후에는 Bit pattern data register에 data를 기입할 수 없습니다. Data 기입은 다음의 순서로 실시합니다.



【주의】 Bit pattern data의 기입 종료 후, BP register 기입 불가 명령(37h)을 발행하지 않으면 해당 bit의 블랭크가 교체된 상태가 되기 때문에, nWR2~WR5 register의 기입을 할 수 없습니다.

◎Bit pattern보간 Drive의 예

주축(ax1)=X축, 제 2축(ax2)=Y축으로 설정하고 <그림 2.32>의 Bit pattern 도형을 1000pps의 정속 Drive, 선속 일정 Mode로 보간 합니다.

WR5 ← 0104h write ; ax1:X축, ax2:Y축 지정, 선속일정
WR6 ← 0900h write ; 주축 속도 Parameter 설정
WR7 ← 003Dh write ; Range:4,000,000(배율:2)
WR0 ← 0100h write

WR6 ← 4DC0h write ; 2축 선속 일정을 위한 $\text{Range:}4,000,000 \times 1.414 = 5,656,000$
WR7 ← 0056h write ;
WR0 ← 0200h write
WR6 ← 01F4h write ; 기동속도: $500 \times 2 = 1000\text{pps}$
WR0 ← 0104h write

WR6 ← 01F4h write ; 드라이버 속도: $500 \times 2 = 1000\text{pps}$
WR0 ← 0105h write
WR0 ← 0039h write ; BPdata clear
WR0 ← 0036h write ; BP register 기입 가능

BP1P ← 0000h write ; 포인트 0~15 X축 +방향
BP1M ← 2BFFh write ; X축-방향
BP2P ← FFD4h write ; Y축+방향
BP2M ← 0000h write ; Y축-방향
WR0 ← 0038h write ; BP data stack
BP1P ← F6FEh write ; 포인트 16~31 X축+방향
BP1M ← 0000h write ; X축-방향

BP2P ← 000Fh write ; Y축+방향
BP2M ← 3FC0h write ; Y축-방향
WR0 ← 0038h write ; BP data stack

BP1P ← 1FDBh write ; 포인트 32~47 X축+방향
BP1M ← 0000h write ; X축-방향
BP2P ← 00FFh write ; Y축+방향
BP2M ← FC00h write ; Y축-방향
WR0 ← 0038h write ; BP data stack

WR0 ← 0034h write ; 2축 Bit pattern 보간:Drive 시작
J1 RR0 /D14, 13 read ; Stack counter가 2 이하로 될 때까지
만약, D14=D13=1 이라면 J1으로 점프

BP1P ← 4000h write ; 포인트 48~61 X축 +방향
BP1M ← 7FF5h write ; X축 -방향
BP2P ← 0000h write ; Y축 +방향
BP2M ← 0AFFh write ; Y축 -방향
WR0 ← 0038h write ; BP data stack

(Note 1)

WR0 ← 0037h write ; BP register 기입 불가
J2 RR0 /D8 read ; 보간 Drive 종료까지 기다린다.
만약, D8=1 이면 J2로 점프

(Note 1)BP data가 모두 없어질 때 까지 이 구간을 반복합니다.

◎Interrupt를 이용한 Bit pattern 보간 Drive

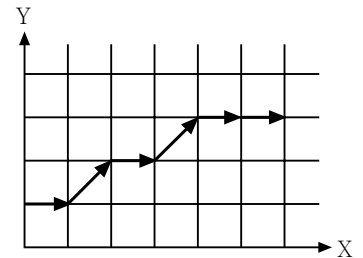
Bit pattern보간 Drive에서는 Drive 중에 Stack Counter(SC)의 값이 2에서 1로 바뀌었을 때 Interrupt가 발생하여 Data 기입을 요구할 수가 있습니다. Interrupt를 발생시키려면 WR5 register의 D15 bit를 1으로 합니다.

이후, Bit pattern 보간 Drive를 시작하면 Stack Counter(SC)의 값이 2에서 1로 바뀌었을 때, INTN 출력 신호가 Low level로 떨어집니다. Bit pattern 보간 data는 16 bit 또는 32 bit의 Pattern data이며 BP data stack 명령을 기입하면 Interrupt는 해제됩니다. 보간 Drive에서 발생시킨 Interrupt는 보간 Interrupt clear 명령(3Dh)으로 해제할 수 있습니다. 또한, INTN 출력 신호를 Low 상태로 두더라도, 보간 Drive가 종료하면 해제되어 hi-Z로 돌아옵니다. 연속으로 BP data가 계속되는 경우는 반복합니다.

2.4.4 선속일정

선속일정 제어는 보간을 실시하고 있는 축의 합성 속도를 항상 일정하게 하는 기능입니다. <그림 2.34>는 2축 보간의 궤적을 나타내고 있습니다.

주축의 기본 Pulse에 따라서 각 축이 Drive pulse를 출력하는 것으로 그림과 같이 X, Y축 양쪽 모두 Drive pulse가 출력될 때에는 1축만의 드라이브 Pulse 출력에 비해서 1.414배 긴 거리를 이동하기 때문에 양축 모두 Drive pulse가 출력될 때의 속도를 1축만의 Drive pulse 속도 보다 1.414배로 해야 합니다.



<그림 2.34 2축 보간의 예>

◎2축 선속일정

2축 선속일정을 설정하려면 WR5 register의 D9, D8bit를 각각 0, 1로 합니다. 그리고, 보간 제 2축의 Range parameter를 주축의 Range parameter의 1.414배의 값으로 설정해 두면 1축만의 Drive pulse 출력 시에는 주축의 Range parameter가 사용되고 양축이 Drive pulse를 출력하면 자동적으로 제 2축의 Range parameter가 사용되어 Pulse 주기가 1.414배로 변경됩니다.

◎3축 선속일정

3축 선속일정도 마찬가지로 WR5 register의 D9, D8bit를 1, 1로 합니다. 그리고, 제 2축의 레인지 Parameter는 주축의 Range의 1.414배로, 제 3축의 Range parameter는 주축의 Range의 1.732배로 설정합니다.

보간 Drive가 시작하면 3축 중, 어느 축이든 1축만의 Drive pulse 출력 시에는 주축의 Range parameter가 사용되고 2축의 Drive pulse 출력시에는, 제 2축의 Range parameter가 사용됩니다. 그리고 제 3축의 Drive pulse 출력시에는, 제 3축의 Range parameter가 사용됩니다. (<그림 2.36 참조>)

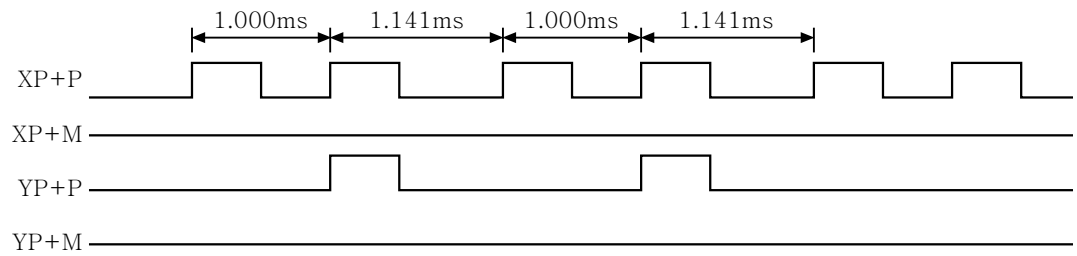
3축 보간에도 주축과 제2축만의 2축 선속일정이 가능합니다. 이 경우는 WR5 register의 D9, D8 bit를 0, 1로 합니다.

◎선속일정 보간 Drive의 예

아래와 같이 주축(ax1)=X축, 제 2축(ax2)=Y축으로 설정하고, 1000pps의 정속 Drive, 선속일정 Mode로 직선보간을 실시하면 <그림 2.35>와 같이 Drive pulse가 출력됩니다.

```

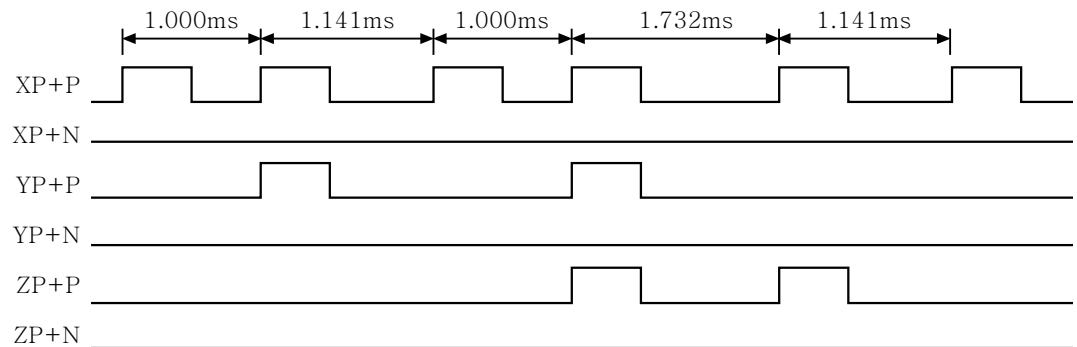
WR5 ← 0104h write ; ax1:X축, ax2:Y축 지정, 선속일정
WR6 ← 0900h write ; 주축 속도 Parameter 설정
WR7 ← 003Dh write ; range:4,000,000(배율 : 2)
WR0 ← 0100h write
WR6 ← 4DC0h write ; 2축선속 일정을 위한 range:4,000,000×1.414=5,656,000
WR7 ← 0056h write
WR0 ← 0200h write
WR6 ← 01F4h write ; 기동속도:500×2=1000pps
WR0 ← 0104h write
WR6 ← 01F4h write ; drive 속도:500×2=1000pps
WR0 ← 0105h write
WR6 ← 03E8h write ; 종점 X값
WR7 ← 0000h write ;
WR0 ← 0106h write
WR6 ← 0190h write ; 종점 Y값
WR7 ← 0000h write ;
WR0 ← 0206h write
WR0 ← 0030h write ; 2축 직선보간 시작
    
```



<그림 2.35 2축 보간 선속 일정한 Pulse 출력의 예(선속도:1000 pps)>

【주의】 양축 Drive pulse 출력 시, Pulse 주기가 1.414배 일 때, Drive pulse의 Hi level의 상승 부분까지 Low level을 Pulse 1주기가 1.414배가 되도록 연장됩니다.

3축 선속일정 시, 한 주기가 1.732배 증가시킬 때에도 마찬가지로 Low level만 연장됩니다.

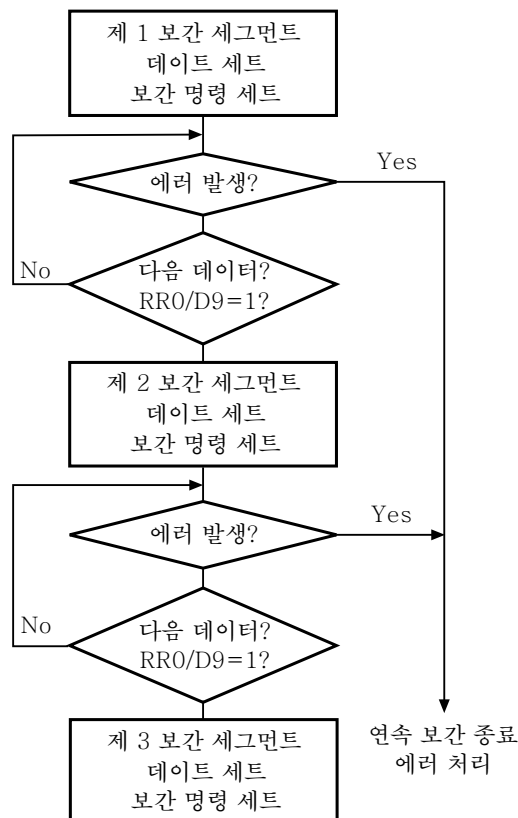


<그림 2.36 3축보간 선속 일정한 Pulse 출력의 예(선속도:1000pps)>

2.4.5 연속보간

연속보간은 직선보간 → 원호보간 → 직선보간 → ...과 같이, 각 각의 보간노드(node)를 Drive 도중에 정지하지 않고, 연속해서 보간을 실시하는 동작입니다. 연속보간 Drive는 현재 실행하고 있는 보간 Drive와 다음 보간 Drive 사이에 Parameter data 및 보간 명령을 설정하여 연속적인 보간 Drive를 실시합니다.

따라서, 모든 보간노드(node)는 그 드라이브 시작부터 종료까지의 시간으로 다음의 보간노드(node)의 Data, 명령을 설정할 시간 이상이 필요합니다. 아래의 그림은 연속보간의 조작 순서를 나타내고 있습니다.



연속보간에서는 RR0 register의 D9(CNEXT) Bit를 사용합니다. 이 Bit는 보간 Drive중, 다음의 보간노드(node)의 Data 및 보간 Drive 명령의 설정 여부를 나타냅니다. 1은 설정가능, 0은 설정 불가능을 나타냅니다.

Drive 정지시에는 0이 되고 보간 Drive가 시작되면 즉시 1이 되어 다음의 보간노드(node)의 Data 및 보간 Drive 명령이 설정됩니다. 다음의 보간노드(node)의 보간드라이브 명령이 설정되면 0(설정 불가)으로 변경되고 다음의 보간 세그먼트가 Drive를 시작하면 다시 1이 되어 다음의 보간 세그먼트의 Data 및 보간 Drive 명령이 설정 가능해 집니다.

◎Interrupt를 이용한 연속보간

WR5 register의 D14 bit는 연속보간 시, Interrupt 허가/금지를 설정하는 Bit입니다.

이 Bit를 1로 하면, RR0 register의 D9(CNEXT) bit가 1(interrupt 허가)이 되어 INTN 출력신호가 Low level로 떨어집니다. Interrupt 처리 루틴에서는 RR0 register의 D9(CNEXT) bit를 확인합니다. 1(기입가능)이면 다음 보간 노드(node)의 Data 및 보간 Drive 명령을 기입합니다. 연속보간 Interrupt의 경우는, 다음 보간 Drive 명령을 기입 하면 INTN 신호는 hi-Z로 돌아옵니다. 다음 보간노드(node)의 Data기입 전에 Clear 명령(3Dh)을 이용하여 Interrupt를 해제하는 것도 가능합니다. 보간 Interrupt는 보간 Drive가 종료하면 강제로 해제되어 INTN 신호는 hi-Z로 돌아옵니다.

◎연속보간 중의 에러 발생

연속보간 Drive 도중에 Limit over run 등의 에러가 발생하면 현재 Drive 노드에서 정지합니다. 정지한 보간노드에서는 Drive 중에 다음의 노드(node)의 data 및 보간명령을 설정하지만 이 보간명령은 무효가 됩니다.

각 보간 노드(node)의 Data 및 보간명령 설정 전에 에러 체크를 하지 않으면 에러로 정지한 후, 즉시 2개의 보간노드 (node)가 실행되기 때문에, 각 보간노드(node)의 data 및 보간명령 설정 전에는, 반드시 에러 체크를 실시하여 에러 이면, 연속보간 루프를 빠져 나가도록 해야 합니다.

◎연속보간의 주의 사항

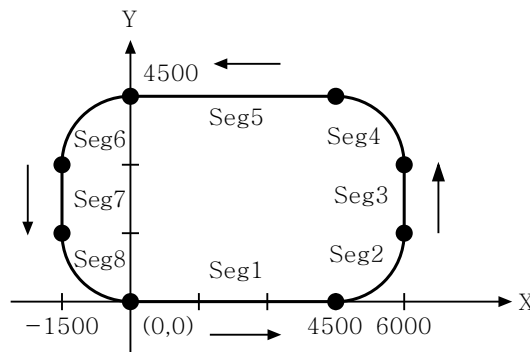
- 각 보간노드(node)는 필요한 data를 설정 한 후에, 보간명령을 설정 합니다. 반대로 하지 말아 주십시오.
- 연속보간의 Drive 속도는 최고 2 MHz까지 입니다.
- 모든 보간노드(node)를 Drive할 시간은 보간 축의 에러 체크 다음, 보간노드(node)의 Data 및 명령을 설정할 시간 이상이 필요합니다. 만약, 다음 보간노드(node)의 Data를 설정하고 있는 동안에, 현재의 보간노드(node)의 Drive가 종료했을 경우에는 RR0 register의 D9(CNEXT) bit는 0이 되지만 다음 보간노드의 Drive 명령이 기입해지면 일단 정지 후, 계속해서 연속보간을 하게 됩니다.
- 연속보간 속에 원호보간이 있는 경우, 원호보간은 종점의 단축이 종점값보다 ± 1 LSB 어긋나는 경우가 있기 때문에 각 노드(node)의 오차가 누적하지 않게 미리 각각의 원호 보간 종점을 확인하고 나서, 연속보간을 실행합니다.
- 2축 보간을 시작으로 3축 보간을 실행 또는 3축 보간을 시작으로 2축 보간을 실행하는 연속보간은 할 수 없습니다.
- 연속보간 도중에 보간 축지정 변경은 할 수 없습니다.

◎연속보간의 예

<그림 2.37>은 (0,0)를 시점으로 노드(node) 1부터 2, 3 ... 노드(node) 8까지를 연속보간 하는 예입니다.

노드(node) 1, 3, 5, 7은 직선보간이며, 노드(node) 2, 4, 6, 8은 반경 1500의 1/4원 입니다.

보간 속도는 1000pps의 정속 드라이브로 선속은 일정합니다.



<그림 2.37 연속보간 궤적의 예>

WR5 ← 0104h write ; ax1:X축, ax2:Y축 지정, 선속일정
 WR6 ← 0900h write ; 주축 속도 Parameter 설정
 WR7 ← 003Dh write ; range:4,000,000(배율:2)
 WR0 ← 0100h write
 WR6 ← 4DC0h write ; 2축선속 일정한 위한 range:4,000,000×1.414=5,656,000
 WR7 ← 0056h write
 WR0 ← 0200h write
 WR6 ← 01F4h write ; 기동속도:500×2=1000pps
 WR0 ← 0104h write
 WR6 ← 01F4h write ; drive 속도:500×2=1000pps
 WR0 ← 0105h write
 WR6 ← 1194h write ; 종점 X치:4500
 WR7 ← 0000h write ;
 WR0 ← 0106h write
 WR6 ← 0000h write ; 종점 Y치:0
 WR7 ← 0000h write ;
 WR0 ← 0206h write
 WR0 ← 0030h write ; 2축 직선 보간

Seg1

J1	RR0 /D5, 4	read ; X, Y축으로 에러가 있으면 D5orD4 =1라면 Error로 점프하여 에러 처리
	RR0 /D9	read ; 다음의 노드(node) Data D9=0 이라면 J1에 점프 ; 기입가능 대기

A처리

WR6 ← 0000h write ; 중심 X치:0
 WR7 ← 0000h write ;
 WR0 ← 0108h write
 WR6 ← 05DCh write ; 중심 Y치:1500
 WR7 ← 0000h write ;
 WR0 ← 0208h write
 WR6 ← 05DCh write ; 종점 X치:1500
 WR7 ← 0000h write ;
 WR0 ← 0106h write
 WR6 ← 05DCh write ; 종점 Y치:1500
 WR7 ← 0000h write ;
 WR0 ← 0206h write
 WR0 ← 0033h write ; CCW 원호보간

Seg2

A처리

WR6 ← 0000h write ; 종점 X치:0
 WR7 ← 0000h write ;
 WR0 ← 0106h write
 WR6 ← 05DCh write ; 종점 Y치:1500
 WR7 ← 0000h write ;
 WR0 ← 0206h write
 WR0 ← 0030h write ; 2축직선 보간

Seg3

A처리

이하 Seg4~8에 대해서도와 같이 실시한다.

2.4.6 가감속 Drive에서의 보간

보간은 일반적으로 정속 Drive로 실시합니다만, 직선 가감속 Drive, 또는 S자 가감속 Drive(직선보간만)에서도 가능합니다. 보간 Drive에서는 연속보간에 대해서 가감속 Drive를 가능하게 하기 위해서 감속 유효명령(3Bh), 감속 무효명령(3Ch)을 사용합니다. 감속 유효명령은 보간 Drive에 대해서 자동 감속, 또는 메뉴얼 감속을 유효하게 하는 명령으로, 감속 무효명령은 감속 유효명령을 무효로 하는 명령입니다. 가감속으로 단독 보간 Drive를 한다면, Drive 시작 전에 반드시 감속 유효상태로 해야 합니다. Drive의 도중에 감속 유효명령의 설정은 적용되지 않습니다.

◎2축/3축 직선보간의 가감속 Drive

2축/3축 직선보간에서는 직선 가감속 Drive 및 S자 가감속 Drive가 가능합니다. 또, 감속에 대해서는 자동 감속과 메뉴얼 감속 모두가 가능합니다. 메뉴얼 감속의 경우는 종점좌표의 각 축 중에서 절대치가 가장 큰 값을 주축의 메뉴얼 감속점으로 설정합니다. 예를 들면, 주축:X, 제2축:Y, 제3축:Z축으로 두어, 종점(X:-20000, Y:30000, Z:-50000)까지의 3축 직선보간을 실시하는 경우, 감속에 필요로 하는 Pulse수를 만일 5000으로 하면, Z축의 종점의 절대치가 가장 크기 때문에, 50000-5000=45000을 주축 X축의 메뉴얼 감속점으로 설정합니다.

직선보간의 가감속 Drive의 예는 2.4.1절의 3축 직선보간 Drive의 예를 참조하십시오.

◎원호보간, Bit pattern보간의 가감속 Drive

원호보간, Bit pattern보간에서는 메뉴얼 감속으로 직선 가감속 Drive만이 가능합니다. S자 가감속 Drive와 자동 감속은 사용할 수 없습니다. 그림과 같이 반경 10000의 원원의 궤적을 직선 가감속 Drive로 그린다면, 원호보간은 자동 감속할 수 없기 때문에 메뉴얼 감속점을 미리 설정해야 합니다.

반경 10000의 원은 0에서 7상한 모두를 통과합니다. 각 상한에 대해서 단축은 항상 Pulse를 출력하기 때문에 단축은 1상한 당 $10000/2=7071$ Pulse 출력하게 됩니다. 따라서, 주축으로부터 출력되는 기본 Pulse의 Pulse수는 원 전체로 $7071 \times 8=56568$ 이 됩니다. 또, 기동속도를 500pps로 하고, Drive 속도를 20000pps 까지를 0.3초에 직선 가속시킨다면 가속도는 $(20000 - 500) / 0.3 = 65000\text{pps/sec}$ 가 되어, 가속 시에 소비되는 Pulse수는 그림에서 사선부의 면적이 되기 때문에 $(500 + 20000) \times 0.3/2 = 3075$ 가 됩니다.

따라서, 메뉴얼 감속점은 $56568 - 3075 = 53493$ 으로 설정합니다.

【주의】 선속일정 Mode에서 이 계산식은 성립되지 않습니다.

WR3 ← 0001h write ; 감속 시작점:메뉴얼
WR5 ← 0004h write ; 보간 ax1:X축, ax2:Y축 지정
WR6 ← 8480h write ; Range:2,000,000(배율:4)
WR7 ← 001Eh write
WR0 ← 0100h write

WR6 ← 0082h write ; 가속도: $130 \times 125 \times 4=65000\text{pps/sec}$
WR0 ← 0102h write
WR6 ← 007Dh write ; 기동속도: $125 \times 4=500\text{pps}$

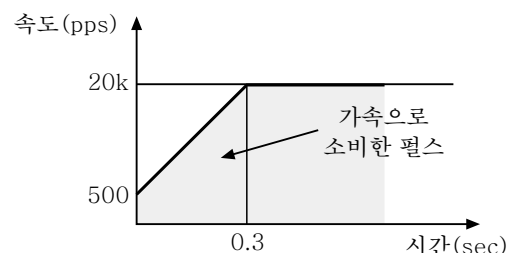
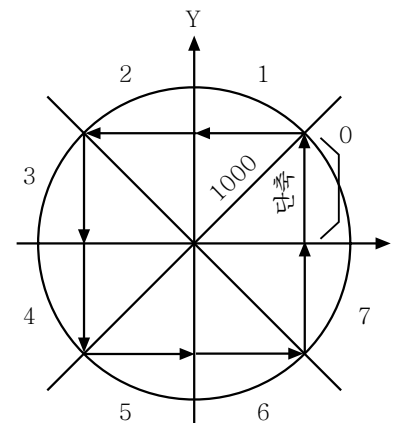
WR0 ← 0104h write
WR6 ← 1388h write ; drive 속도: $5000 \times 4=20000\text{pps}$
WR0 ← 0105h write ;

WR6 ← D8F0h write ; 중심 X:-10000
WR7 ← FFFFh write
WR0 ← 0108h write

WR6 ← 0000h write ; 중심 Y:0
WR7 ← 0000h write
WR0 ← 0208h write

WR6 ← 0000h write ; 종점 X:0
WR7 ← 0000h write
WR0 ← 0106h write

WR6 ← 0000h write ; 종점 Y:0
WR7 ← 0000h write
WR0 ← 0206h write



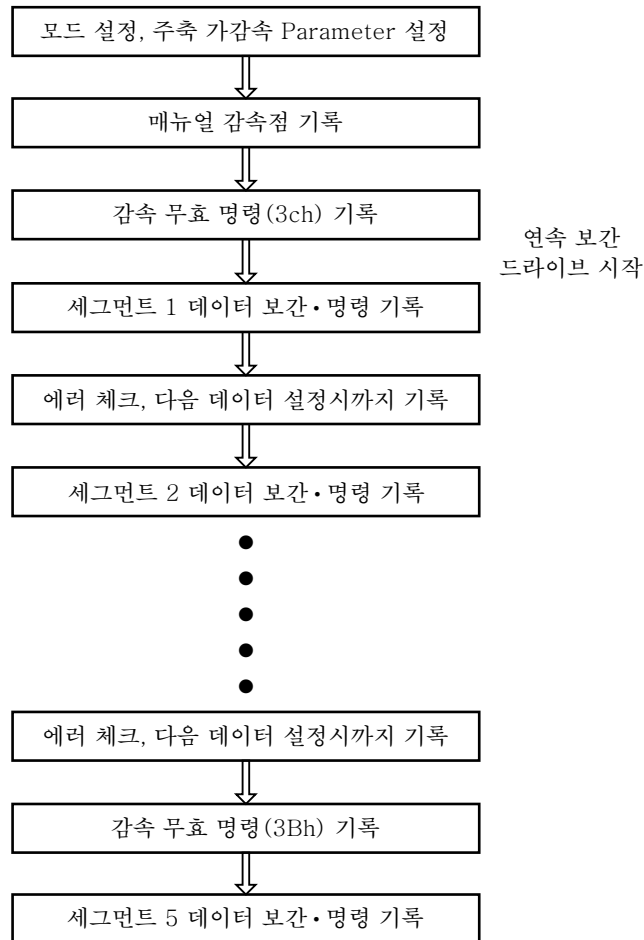
WR6 ← D0F5h write ; 메뉴얼 감속점 : 53493
 WR7 ← 0000h write
 WR0 ← 0107h write
 WR0 ← 003Bh write ; 감속 유효
 WR0 ← 0033h write ; CCW 원호보간 Drive

◎연속보간의 가감속 Drive

연속보간에서도 메뉴얼 감속의 직선 가감속 Drive만이 가능합니다. S자 가감속 Drive와 자동 감속은 사용할 수 없습니다. 연속보간에서는, 미리 메뉴얼 감속점을 설정해 두지 않으면 안됩니다. 메뉴얼 감속점은 감속을 실시하는 최종 노드(node)로 출력되는 주축으로부터 기본 Pulse에 대한 값을 설정합니다.

연속 보간에서는 처음부터 감속을 무효로하여 보간 Drive를 시작합니다. 그리고, 감속시키는 최종노드(node)의 보간 명령 설정 전에 감속유효명령을 설정합니다. 최종노드(node)의 Drive에 들어가면 감속유효상태가 되고 최종노드의 시작시점에서 주축의 기본 Pulse수가 Count되어 메뉴얼 감속점의 값을 넘었을 때에 감속이 시작됩니다.

예를 들면, 노드(node) 1에서 5까지의 연속보간에 대해서, 최종노드(node) 5로 메뉴얼 감속시키는 경우에는, 다음과 같이 설정됩니다.



메뉴얼 감속점은, 노드(node) 5부터 주축의 기본 pulse수에 대한 값이므로, 주의해 주십시오.

예를 들면, 감속 Pulse가 2000 소비된다면 노드(node) 5로 출력되는 기본 Pulse의 총 Pulse수가 5000, 5000-2000=3000을 메뉴얼 감속점으로 설정합니다.

감속을 시작하고 나서 정지할 때까지는 반드시 하나의 노드(node) 안에서 설정해야 합니다.

즉, 감속 정지의 최종노드(node)는 주축으로부터 출력되는 기본 Pulse의 총수가, 감속에 소비하는 Pulse 수 이상 필요합니다.

2.4.7 보간 Step전송(Command)

보간 Drive를 1Pulse마다의 Step 전송하는 동작입니다. Command로 실시하는 방법으로 Step 전송 때에는, 보간 주축은 정속 Drive로 설정합니다. 각 축에서 출력되는 Drive pulse의 Hi level 폭은 보간의 주축으로 설정되어 있는 Drive속도에 의해 정해진 Pulse 주기의 1/2값이 됩니다.
Low level 폭은 다음의 Command가 발생할 때까지 연장됩니다.

◎Command에 의한 보간 Step 전송

보간 Drive를 Step 전송하는 Command로는 보간 Single step(3Ah)명령이 있습니다. WR5 register의 D12 bit를 1로 하면 Command에 의한 보간 Step전송이 가능하게 됩니다. Command에 의한 보간 Step전송 방법은 아래와 같습니다.

①WR5 register의 D12 bit를 1로 한다.

Command에 의한 보간 step mode가 됩니다.

②보간 주축의 기동속도와 Drive속도를 같은 값으로 설정한다.

기동속도와 Drive속도를 같은 값으로 하면 정속 Drive가 됩니다. 이 때의 속도치는 Single step 명령을 설정하는 사이클보다 빠르게 설정해야 합니다. 예를 들면, Single step명령을 최고 1msec의 사이클로 설정된다면 기동속도와 Drive속도는 1000pps보다 빠른 값으로 설정합니다.

③보간 Data를 설정 한다. (중점, 중심점 등)

④보간 명령을 기입한다.

보간 명령을 기입해도 Command에 의한 보간 Step mode로 되어 있기 때문에, 각 축의 Drive pulse는 출력하지 않습니다.

⑤보간 Single step(3Ah) 명령을 기입한다.

보간 연산 결과로 Drive pulse가 각 축에서 출력됩니다. 보간 Drive가 종료할 때까지 Single step(3Ah)명령을 기입합니다. 보간 Step 전송을 도중에 중지하는 경우는, 주축에 대해서 즉시정지 명령(27h)을 설정하여 Drive속도에서의 1펄스 주기 이상의 시간지연을 주고 다시 보간 Single step 명령을 설정하면 Drive가 정지합니다.

보간 Drive종료 후에 기입해진 보간 Single step명령은 적용되지 않습니다.

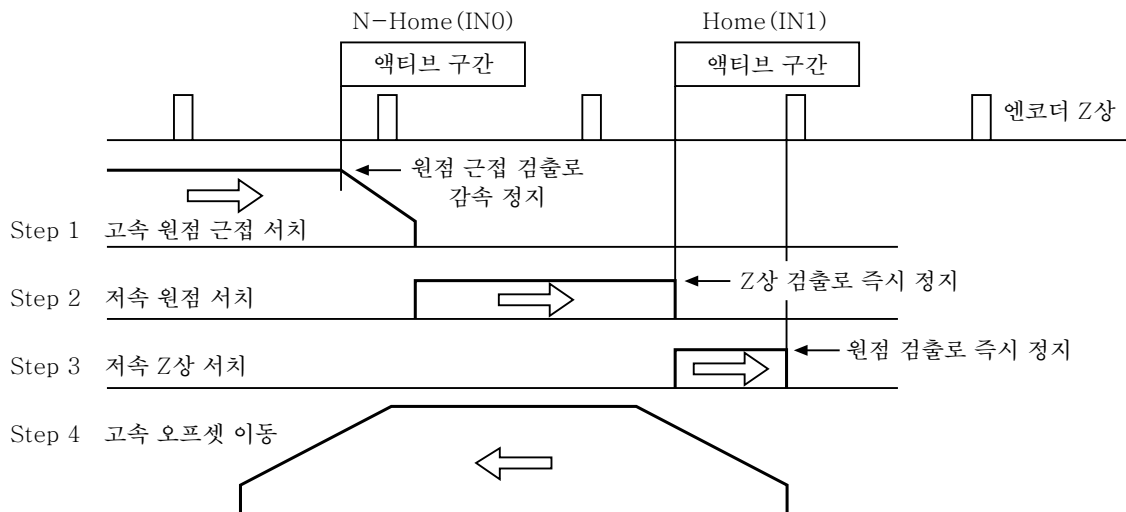
2.5 자동 원점 복귀 출력

프로그램 설정없이 고속 원점근접서치→ 저속 원점서치→ Encoder Z상서치→ Offset 이동 등의 원점복귀 순서를 자동으로 실행하는 기능을 가지고 있습니다. 자동 원점복귀는 아래 표와 같이 Step1부터 Step4를 설정한 순서에 따라 실행합니다. 각 Step에 대해서 실행/비실행을 선택, 서치 방향을 설정합니다.

Step 1, 4는 Drive 속도로 고속서치동작을 합니다. 또, Step 2, 3은 원점 검출 속도로 저속 서치동작을 합니다.

Step 번호	동 작	서치 속도	검출 신호
Step 1	고속 원점근접서치	드라이브 속도(V)	nIN0(*1)
Step 2	저속 원점서치	원점 검출 속도(HV)	nIN1
Step 3	저속 Z상서치	원점 검출 속도(HV)	nECZ
Step 4	고속 오프셋 이동	드라이브 속도(V)	-

(*1) : 원점신호를 nIN0, nIN1 양쪽 모두에 입력하는 것으로써, 원점 신호 1점만으로도 고속 원점 서치가 가능합니다. (2.5.6절 ◎원점 신호만의 원점 복귀 예를 참조)



<그림 2.39 자동 원점 복귀의 개략도>

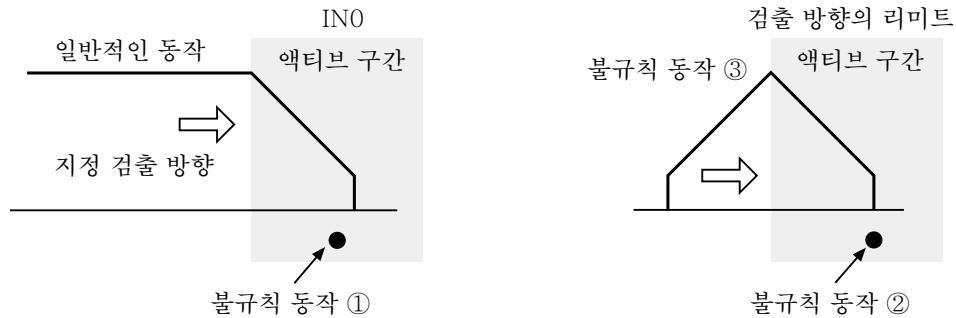
2.5.1 각 Step의 동작

각 Step은 실행/비실행, 검출방향(+방향 / -방향)을 설정할 수 있습니다.

비실행으로 설정하면 해당 Step은 실행되지 않고 다음 Step으로 진행됩니다.

◎Step 1 고속 원점근접서치

고속 원점근접서치는 설정된 Drive 속도(V) 및 방향으로 Drive pulse출력 중에 원점근접신호(nIN0)가 Active되면 Step 2로 진행합니다. 고속 서치동작을 실행하기 위해서는 Drive 속도(V)를 기동속도(SV)보다 빠르게 설정합니다. 가감속 Drive 중에 원점근접신호(nIN0)가 액티브 되면 감속 정지합니다.

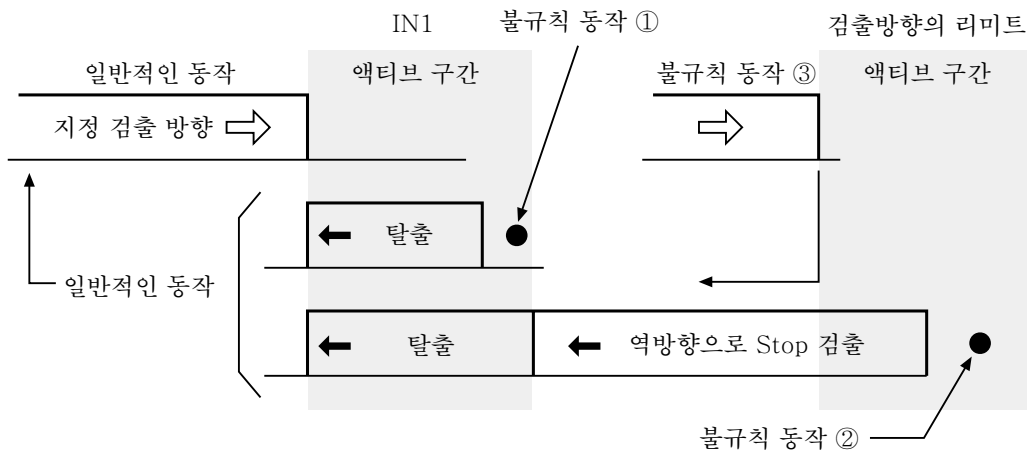


불규칙 동작

- ①Step 1 시작 전에 이미 원점근접신호(nIN0)가 Active 되어 있다. → Step 2로 진행됩니다.
- ②Step 1 시작 전에 검출방향의 Limit 신호가 Active 되어 있다. → Step 2로 진행됩니다.
- ③실행 중에 검출방향의 Limit 신호가 액티브 되었다. → Drive를 정지하고 Step 2로 진행됩니다.

◎Step 2 저속 원점서치

저속 원점서치는 Drive pulse출력 중에 원점신호(nIN1)가 액티브되면 원점검출속도(HV), 지정된 방향으로 저속서치 동작을 실행합니다. 저속 서치동작을 실행하기 위해서는 원점검출속도(HV)를 기동속도(SV)보다 낮은 값으로 설정합니다. 정속 드라이브시 원점신호(nIN1)가 액티브 되면 즉시 정지합니다.

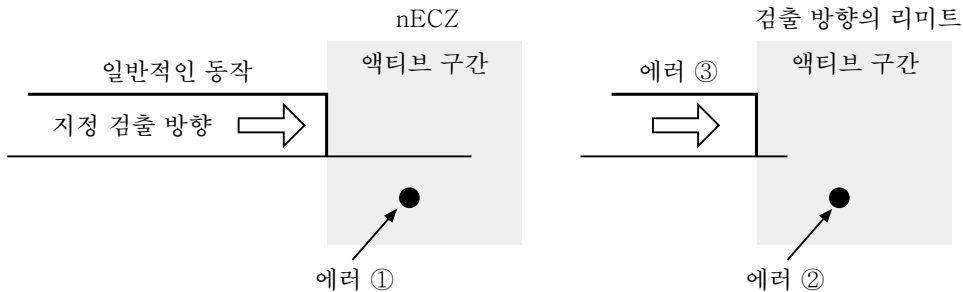


불규칙 동작

- ①Step 2 시작 전에 이미 원점신호(nIN1)가 Active 되어 있다.
 - 원점신호(nIN1)가 비액티브 될 때까지, 지정의 검출 방향과 반대의 방향으로 원점검출속도(HV)로 이동하여 원점신호(nIN1)가 비액티브 되면 Step 2를 시작 합니다.
- ②Step 2 시작 전에 검출방향의 Limit 신호가 Active 되어 있다.
 - 원점신호(nIN1)가 액티브 될 때까지 지정된 검출방향과 반대의 방향으로 원점검출속도(HV)로 이동합니다. 원점신호(nIN1)가 액티브되면 원점신호(nIN1)가 비액티브 될 때까지 지정된 검출방향과 반대 방향으로 원점검출속도(HV)로 이동합니다. 원점신호(nIN1)가 비액티브 되면, Step 2를 실행합니다.
- ③실행 중에 검출방향의 Limit신호가 Active 되었다 .
 - Drive를 정지하고 → ②와 동일한 동작을 합니다 .

◎Step 3 저속 Z상서치

Encoder Z상신호(nECZ)가 active 되면 원점검출속도(HV), 지정된 방향으로 Drive pulse를 출력합니다. 저속 서치동작을 실시하기 때문에 원점검출속도(HV)를 기동속도(SV) 보다 낮은 값으로 설정합니다. 정속 Drive실행 중, Encoder Z상 신호(nECZ)가 액티브되면 즉시 정지합니다. 검출 조건으로서 Encoder Z상 신호(nECZ)와 원점 신호(nIN1)의 AND 조건으로 정지시킬 수도 있습니다.



Encoder Z상신호(nECZ)가 Active될 때, 실제위치 Counter(EP)를 Clear 시킬 수도 있습니다. 2.3.4절을 참조하십시오.

【주의】

- ①Step 3 시작 전에 이미 Encoder Z상신호(nECZ)가 Active 되어 있으면 에러가 되어, nRR2 register 의 D7bit가 1이 되어 자동원점복귀는 종료합니다. Step 3은 반드시 Encoder Z상신호(nECZ)가 비액티브 상태(안정상태)에서 시작하도록 기계 시스템을 조정해야 합니다.
- ②Dtep 3 시작 전에 검출 방향의 Limit신호가 Active되어 있으면 에러가 되어, nRR2 register의 검출방향 Limit 에러 bit(D2 또는 D3)가 1이 되어 자동원점복귀는 종료합니다.
- ③실행 중에 검출 방향 Limit신호가 Active되면 검출 동작은 중단되어 nRR2 register의 검출방향의 Limit 에러 bit(D2 또는 D3)가 1이 되어 자동원점복귀는 종료합니다.

◎Step 4 고속 Offset 이동

Step 4는 Drive 속도(V), 지정된 방향으로 출력 Pulse수(P)로 설정되어 있는 Pulse수 만큼 Pulse를 출력합니다. 기계적 원점에서 작업 원점으로 이동시키고 싶은 경우에 사용합니다. Mode 설정에 의해서 이동 종료 후, 논리위치 Counter 및 실제위치 Counter를 Clear 시킬 수도 있습니다. Step 4 시작 전, 또는 실행 중에 이동 방향의 Limit신호가 Active되면 에러가 되어 nRR2 register의 검출 방향 Limit 에러 bit(D2 또는 D3)가 1이 되어 자동원점복귀는 종료합니다.

2.5.2 서치 속도와 Mode의 설정

자동 원점 복귀를 실행하기 위해서는 속도 Parameter와 Mode 설정이 필요합니다.

◎속도 Parameter의 설정

속도 Parameter	명령 코드	내 용
드라이브 속도(V)	05	스텝1, 4의 고속 서치속도가 됩니다. 가감속 드라이브를 시키기 위해서 레인지(R), 가속도(A), 기동속도(SV)를 적절한 값으로 설정할 필요가 있습니다. 2.2.2절 참조.
원점검출 속도(HV)	61	스텝 2, 3의 저속 서치 속도가 됩니다. 검출 신호가 액티브 될 때, 즉시 정지시키기 위해서 기동속도(SV)보다 낮은 값으로 설정합니다. 2.2.1절 참조.

◎자동원점복귀의 Mode 설정

자동원점복귀의 Mode 설정은 확장 Mode 설정 명령(60h)에 의해 설정합니다. 아래와 같이 WR7 register의 각 bit를 설정합니다. 또, 자동원점복귀 종료 시에 Interrupt를 발생시키는 경우는, WR6 register D5(HMINT)를 1로 설정합니다. 확장 Mode 설정 명령(60h)은 WR6 및 WR7의 각 Bit data가 동시에 내부 Register에 설정되기 때문에 WR6 register의 다른 bit에 대해서도 적절히 설정할 필요가 있습니다.

WR7	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	DCCW2	DCCW1	DCCW0	DCC-L	DCC-E	LIMIT	STAND	PCLR	ST4-D	ST4-E	ST3-D	ST3-E	ST2-D	ST2-E	ST1-D	ST1-E
	편차 카운터 클리어 출력								Step 4		Step 3		Step 2		Step 1	

WR6	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	FL2	FL1	FL0	FE4	FE3	FE2	FE1	FE0	SMODE	0	HMINT	VRING	AVTRI	POINV	EPINV	EPCLR

WR7/D6, 4, 2, 0	STm-E	각 Step의 동작의 실행/비실행을 설정합니다. 0:비실행 1:실행 각 Step에서 검출하는 입력신호의 논리설정은 WR1 register로 설정합니다. 4.4절 참조.
WR7/D7, 5, 3, 1	STm-D	각 step의 검출/이동 방향을 설정합니다. 0:+ 방향, 1:- 방향
WR7/D8	PCLR	1로 설정하면 Step 4 종료 후, 논리위치 Counter 및 실제위치 Counter가 Clear 됩니다.
WR7/D9	SAND	1로 설정하면 Step 3 동작은 원점신호(nIN1)가 Active 될 때, Encoder Z상 신호(nIN2)가 Active 될 때 정지합니다.
WR7/D10	LIMIT	Over run limit신호(nLMT+ 또는 nLMT-)를 사용하여 자동원점복귀를 실시할 때 1로 설정합니다.
WR7/D11	DCC-E	0으로 설정합니다. (본 보드에서는 편차카운터 클리어 기능은 지원하지 않습니다.)
WR7/D12	DCC-L	0으로 설정합니다.
WR7/D15~13	DCCW2 ~ 0	0으로 설정합니다.
WR5/D5	HMINT	자동원점복귀 종료 후, Interrupt 신호(INTN)를 발생시킵니다. 본 Bit를 1하면, 자동원점복귀 종료 후, Interrupt 신호(INTN)가 Low active 되어, Interrupt를 발생시킨 축의 RR3/D8(HMEND) bit를 1로 설정합니다. Interrupt를 발생시킨 축의 RR3 register를 읽으면, RR3 register의 Bit는 0으로 Clear 되어 Interrupt출력 신호는 Hi-Z으로 돌아옵니다.

리셋 시에는, 각 축의 Mode 설정 Bit는 모두 0으로 설정됩니다.

2.5.3 자동원점복귀의 실행과 상황(Status)

◎자동원점복귀의 실행

자동원점복귀는, 자동원점복귀 실행명령(62h)에 의해 설정됩니다. 각 축의 자동원점복귀 Mode와 속도 Parameter를 설정한 후, WR0 register에 축 지정과 함께 명령코드 62h를 기입하는 것으로 시작합니다.

각 축, 개별적으로 실행 가능하며 전축 동시 실행도 가능합니다.

◎자동원점복귀의 중단

자동원점복귀를 도중에 중단시키고 싶을 때는, 실행하고 있는 축에 대해 Drive 감속정지명령(26h), 또는 드라이브 즉시정지명령(27h)을 설정합니다. 현재 실행하고 있는 Step은 중단되어 자동원점복귀를 종료합니다.

◎주Status register

주Status register RR0의 D3~D0은 각 축의 Drive 실행 상태를 나타내는 bit입니다. 자동원점복귀 실행 시도 본 bit로 상태 확인이 가능합니다. 각 축의 자동원점복귀가 시작되면 이 Bit가 1이 되어 Step 1 동작 시작부터 Step 4 동작 종료까지 1을 나타내고 있습니다. Step 4를 종료하면 0으로 돌아옵니다.

RR0	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	-	BPSC1	BPSC0	ZONE2	ZONE1	ZONE0	CNEXT	I-DRV	U-ERR	Z-ERR	Y-ERR	X-ERR	U-DRV	Z-DRV	Y-DRV	X-DRV
	편차 카운터 클리어 출력								각 축의 드라이브							

각 축의 에러를 나타내는 D7~D4(n-ERR) Bit는 Step 1, 2의 불규칙 동작으로 검출 방향의 Limit신호 등에 있어서 정상 동작에도 불구하고 1을 나타낼 때가 있기 때문에 주의하십시오. 이러한 에러 비트는 자동원점복귀 실행 시에는 확인하지 않고 자동원점복귀가 종료한 후에 확인하도록 하십시오.

◎Status register 2

Status register 2(RR2)는 D7~D0에는 에러 정보가 표시되고 D12~D8에는 원점복귀 실행 Status가 표시됩니다.

RR2	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	-	0	0	HMST4	HMST3	HMST2	HMST1	HMST0	HOME	0	ENG	ALARM	HLMT-	HLMT+	SLMT-	SLMT+
	자동 원점 출력 실행 스테이트								자동 원점출력시의 신호 에러							

에러 정보 Bit 중, D7(HOME) bit는 자동원점복귀 실행 중, Step 3 시작시에 이미 Encoder Z상신호(nIN2)가 Active 되어 있으면 1이 됩니다. 이 Bit는, 다음의 Drive 명령 또는, 자동원점복귀 명령을 설정하면 Clear 됩니다.

그리고 종료 Status clear 명령(25h)으로도 Clear 할 수가 있습니다. 자동원점복귀 실행 Status는 자동원점복귀 실행 중에 현재 실행하고 있는 동작 상태를 나타냅니다.

실행 스테이트	실행 Step	동작 내용
0		자동원점출력 실행 명령 기다림.
3	Step 1	지정검출 방향으로서, IN0 신호의 액티브 기다림.
8		지정검출 반대 방향으로서, IN1 신호의 액티브 기다림. (불규칙적임 동작)
12	Step 2	지정검출 반대 방향으로서, IN1 신호의 비 액티브 기다림. (불규칙적임 동작)
15		지정검출 방향으로서, IN1 신호의 액티브 기다림.
20	Step 3	지정검출 방향으로서, ECZ 신호의 액티브 기다림.
25	Step 4	지정검출 방향으로 오프셋 이동 중.

2.5.4 자동 원점 출력시의 에러

자동원점 복귀 실행 중, 아래 표와 같은 에러 발생이 일어날 가능성이 있습니다.

에러 발생 요인	에러 발생 후의 동작	종료 표시
Step 1 ~ 4에서 Alarm 신호 액티브	검출 드라이브 즉시 정지, 이후 스텝은 실행하지 않고 종료	RR0-D7~4:1, nRR2-D4:1, nRR1-D14:1
Step 1 ~ 4에서 EMG 신호 액티브	검출 드라이브 즉시 정지, 이후 스텝은 실행하지 않고 종료	RR0-D7~4:1, nRR2-D5:1, nRR1-D15:1
Step 3에서 진행 방향의 리미트 신호(LMT+/-) 액티브	검출 드라이브는 즉시/감속 정지, 이후 스텝은 실행하지 않고 종료	RR0-D7~4:1, nRR2-D3/2:1, nRR1-D13/12:1
Step 4에서 진행 방향의 리미트 신호(LMT+/-) 액티브	오프셋 이동 즉시/감속 정지.	RR0-D7~4:1, nRR2-D3/2:1, nRR1-D13/12:1
Step 3에서 시작 전에 ECZ 신호가 액티브	이후 스텝은 실행하지 않고 종료.	RR0-D7~4:1, nRR2-D7:1

자동원점복귀 종료 후, 반드시 각 축의 에러 Bit(RR0-D7~D4)를 확인해야 합니다. 1로 설정된 에러 bit는 올바른 자동원점복귀를 실행하지 않습니다. 한편, 자동원점복귀 실행 도중에 각 축의 에러 Bit를 확인하는 것은 올바르지 않습니다. Step 1, 2의 불규칙 동작으로 인하여 에러 Bit가 1로 설정될 수 있기 때문입니다.

◎센서 고장시의 증상

원점신호나 Limit신호 등의 센서회로의 고장시에 일어날 수 있는 증상에 대해서 기술합니다. 배선으로 인한 노이즈나 소자의 불안정 동작 등의 원인에 의한 고장에 대해서는 해석이 어렵고 여기에서는 시스템 개발 시, 신호 Level의 논리 설정 및 신호연결 시 일어날 수 있는 경우만 기술합니다.

고장 요인		증상
리미트 센서 및 배선 경로의 고장	항상 ON	해당 방향으로 움직이지 않고, 종료시에 리미트 에러 비트(nRR2-D3/2)가 1이 되어 있다.
	항상 OFF	해당 방향의 기계적 종점에 부딪치고, 원점출력동작이 종료되지 않는다.
원점근접(nIN0) 센서 및 배선 경로의 고장	항상 ON	Step 1을 유효로 설정하고 신호가 OFF의 위치에서 자동원점출력을 시작하여도 Step 1(고속 원점근접서치)을 실행하지 않고, Step 2로 넘어간다.
	항상 OFF	Step 1(고속 원점 근접 서치)로 리미트에서 정지하고 나서 Step 2의 불규칙 동작으로 진행된다. 원점출력의 결과는 올바르지만 일반적인 동작이 아니다.
원점근접(nIN1) 센서 및 배선 경로의 고장	항상 ON	Step 2(저속 원점서치)로 역방향으로 움직이다 역방향의 리미트에서 정지한다. 종료 시에 역방향 리미트의 에러 비트(nRR2-D3/2)가 1이 된다.
	항상 OFF	Step 2(저속 원점서치)로 지정된 방향의 리미트에서 역방향으로 이동하고, 역방향의 리미트에서 종료한다. 종료시에 역방향 리미트의 에러 비트(nRR2-D3/2)가 1이 된다.
Z상(nECZ) 센서 및 배선 경로의 고장	항상 ON	Step 3(저속 Z상서치)에서 종료한다. nRR2-D7이 1이 된다.
	항상 OFF	Step 3(저속 Z상서치)로, 지정된 방향의 리미트에서 정지한다. 종료 시에 지정 방향 리미트의 에러 비트(nRR2-D3/2)가 1이 된다.

2.5.5 자동 원점 출력시의 주의점

◎서치 속도

원점검출속도(HV)는 원점복귀 시, 위치 정밀도를 올리기 위해서 저속으로 설정할 필요가 있습니다. 입력 신호가 Active되면 즉시정지 하도록 기동속도 이하의 값으로 설정합니다. Step 3의 Encoder Z상서치를 실시하는 경우에는 Z상 신호의 지연과 원점검출속도(HV)의 관계가 중요하므로 다음과 같이, Z상 신호의 Photo-coupler의 지연시간과 적분 필터의 지연시간이 최대 500 μ sec 걸린다면, Encoder의 Z상 출력이 1msec 이상 ON하도록, 원점검출속도를 설정할 필요가 있습니다.

◎Step 3(Z상서치) 시작 위치

Step 3의 Z상서치는 Z상(nECZ) 신호가 비Active 상태에서 Active로 변화할 때에 검출 Drive를 정지시킵니다. 따라서, Step 3의 시작위치(Step 2의 정지 위치)가 이 변화지점에서 벗어나야 합니다. 일반적으로, Step 3의 시작위치가 Encoder Z상 위치의 180° 반대 측이 되도록 기계적으로 조정합니다.

◎소프트웨어 Limit

자동원점복귀 실행 중, 소프트웨어 Limit는 적용되지 않습니다. 소프트웨어 Limit를 적용하면 자동원점복귀는 올바르게 실행하지 않습니다. 자동원점복귀를 정상 완료 후, 논리위치 Counter, 실제위치 Counter를 올바르게 설정한 후에 소프트웨어 Limit를 설정해야 합니다.

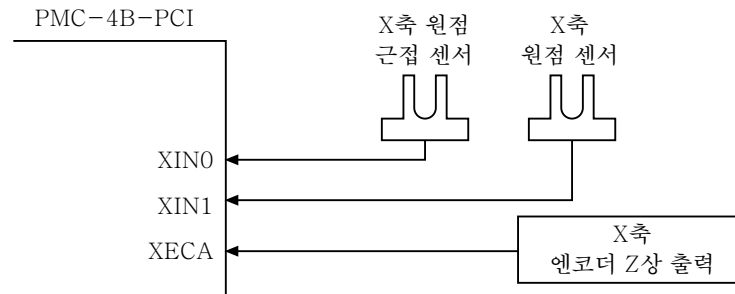
◎각 입력 신호의 논리설정

자동원점복귀로 사용하는 입력신호의 설정은 WR1 register의 Bit(WR1-D2, D4, D7)로 설정합니다.

2.5.6 자동 원점 복귀의 예

◎원점근접, 원점, Z상신호에 의한 원점복귀의 예

Step	입력신호와 논리레벨	검출방향	검출속도
Step 1	원점근접 (IN0) 신호, Low 액티브	—	20,000pps
Step 2	원점 (IN1) 신호, Low 액티브	—	500pps
Step 3	Z 상(XECZ) 신호, Hi 액티브	+	500pps
Step 4	(+ 방향으로 3500 펄스 오프셋 이동)	+	20,000pps



- Step 1의 고속 서치, 및 Step 4의 Offset이동은 가감속 Drive로 실시합니다.
기동속도:1,000pps로 20,000pps 까지를 0.2초에(가감속도도=19,000/0.2=95,000pps/sec) 직선 가감속 시킵니다.
- Step4 완료 후, 논리위치 Counter, 실제위치 Counter의 값을 Clear 합니다.

[Parameter 및 Mode 설정]

WR0 ← 010Fh write : X축선택
 WR1 ← 0010h write : 입력 신호 논리 설정:XIN0, XIN1:Low active, XIN2:Hi active (4.4절 참조)
 WR6 ← 5D00h write : 확장 Mode 설정
 WR7 ← 015Fh write : WR6에 입력 신호 필터의 Mode 기록 (2.8절 참조)
 : D15~D13 010 필터 지연:512μsec
 : D9 0 XIN2 신호:필터 무효
 : D8 1 XIN1, 0 신호:필터 유효
 : WR7에 자동 원점 복귀의 Mode 기록
 : D15~D13 000
 : D12 0
 : D11 0
 : D10 0 Limit 신호를 원점 신호로서 사용:무효
 : D9 0 Z상신호 AND 원점 신호:무효
 : D8 1 논리/실제위치 Counter clear:유효
 : D7 0 Step 4 이동 방향:+방향
 : D6 1 Step 4:유효
 : D5 0 Step 3 검출 방향:+방향
 : D4 1 Step 3:유효
 : D3 1 Step 2 검출 방향:-방향
 : D2 1 Step 2:유효
 : D1 1 Step 1 검출 방향:-방향
 : D0 1 Step 1:유효
 WR0 ← 0160h write : X축으로 확장 Mode 설정 명령 기록
 WR6 ← 3500h write : Range:8,000,000(배율:10)
 WR7 ← 000Ch write
 WR0 ← 0100h write

WR6 ← 004Ch write : 가감속도:95,000pps/sec
 WR0 ← 0102h write : 95000/125/10=76
 WR6 ← 0064h write : 기동속도:1000pps
 WR0 ← 0104h write
 WR6 ← 07D0h write : Step 1, 4의 속도 : 20000pps
 WR0 ← 0105h write
 WR6 ← 0032h write : Step 2, 3의 속도 : 500pps
 WR0 ← 0161h write
 WR6 ← 0DACH write : Offset 이동 Pulse량:3500
 WR7 ← 0000h write
 WR0 ← 0106h write
 WR0 ← 0162h write : 자동원점복귀 실행시작

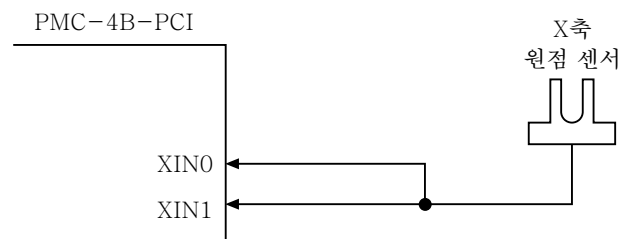
실행 시작 후, RR0-D0(X-DRV) bit가 1에서 0으로 돌아오면 자동원점복귀는 종료합니다.

종료 후, RR0-D4(X-ERR)에 Limit가 1이 되어 있으면 자동원점 중에 어떠한 에러가 발생하여 자동원점 복귀가 정상적으로 종료되지 않는 것입니다. XRR2-D7, D5~D0bit, XRR1-D15~D12 bit를 이용하여 해석을 실시합니다.

◎원점신호 만의 원점복귀 예

원점신호를 IN0와 IN1 단자에 공동 입력하여 하나의 원점신호로 고속원점복귀를 실시하는 예입니다.

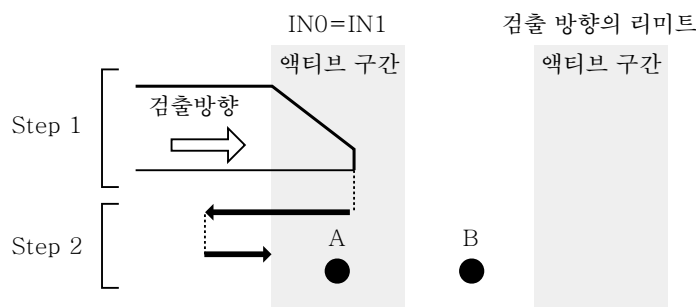
Step	입력 신호와 논리 레벨	검출방향	검출속도
Step 1	원점 근접(IN0) 신호, Low 액티브	-	20,000pps
Step 2	원점(IN1) 신호, Low 액티브	-	500pps
Step 3	-		
Step 4	(+ 방향으로 3500 펄스 오프셋 이동)	+	20,000pps



위의 표와 같이, Step 1과 Step 2의 신호논리 Level과 검출방향은 같게 합니다. (논리 Level을 반대로 설정하는 방법도 있습니다.) Step 1에서 고속으로 원점을 서치하여 원점신호가 Active 되면 감속 정지합니다.

정지위치가 원점신호의 액티브 구간이면, Step 2의 불규칙 동작 ①에 의해, 역방향으로 탈출하고 나서, Step 2의 동작에 들어가 원점을 검출합니다. 만약, Step 1의 정지 위치가 원점신호의 Active 구간을 넘겨 버렸을 경우에는, Step 2로 검출 방향의 Limit에 걸리므로 불규칙 동작 ③의 동작이 됩니다.

자동원점복귀 시작위치가 그림 A점에 있는 경우에는, Step 1은 실행되지 않고 Step 2의 불규칙 동작 ①을 실행합니다. 그림에서 B점에 있는 경우에는 Step 1에서 검출방향의 리미트에 걸린 후, Step 2의 불규칙 동작 ②를 실행합니다.



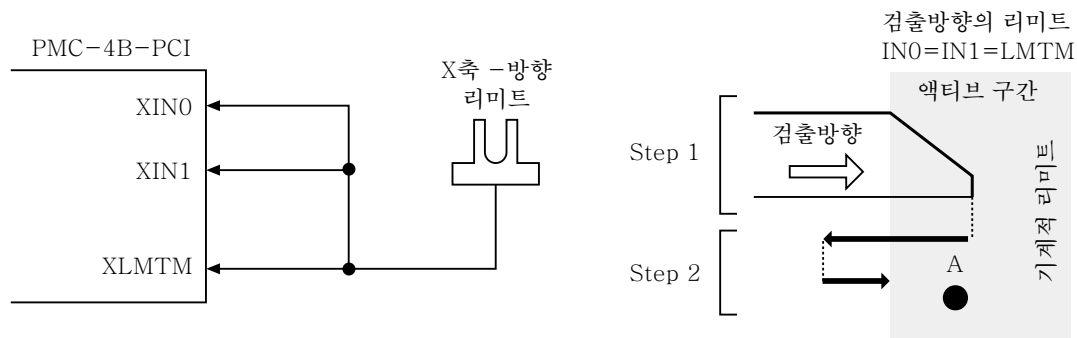
[Parameter 및 Mode 설정]

WR0 ← 010Fh write : X축선택
 WR1 ← 0000h write : 입력 신호 논리 설정 : XIN0:Low active, XIN1:Low active (4.4절 참조)
 WR6 ← 5F00h write : 확장 Mode 설정
 WR7 ← 014Fh write : WR6에 입력 신호 필터의 Mode 기록 (2.8절 참조)
 : D15~D13 010 필터 지연:512 μ sec
 : D8 1 XIN1, 0 신호:필터 유효
 : WR7에 자동 원점 복귀의 Mode 기록
 : D15~D13 000
 : D12 0
 : D11 0
 : D10 0 Limit 신호를 원점 신호로서 사용:무효
 : D9 0 Z상신호 AND 원점 신호:무효
 : D8 1 논리/실제위치 Counter clear:유효
 : D7 0 Step 4 이동 방향:+방향
 : D6 1 Step 4:유효
 : D5 0 Step 3 검출 방향:+방향
 : D4 0 Step 3:유효
 : D3 1 Step 2 검출 방향:-방향
 : D2 1 Step 2:유효
 : D1 1 Step 1 검출 방향:-방향
 : D0 1 Step 1:유효
 WR0 ← 0160h write : X축으로 확장 Mode 설정 명령 기록
 WR6 ← 3500h write : Range : 8,000,000(배율:10)
 WR7 ← 000Ch write
 WR0 ← 0100h write
 WR6 ← 004Ch write : 가감속도:95,000pps/sec
 WR0 ← 0102h write : 95000/125/10=76
 WR6 ← 0064h write : 기동속도 : 1000pps
 WR0 ← 0104h write
 WR6 ← 07D0h write : Step 1, 4의 속도 : 20000pps
 WR0 ← 0105h write
 WR6 ← 0032h write : Step 2, 3의 속도 : 500pps
 WR0 ← 0161h write
 WR6 ← 0DACH write : Offset 이동 Pulse량 : 3500
 WR7 ← 0000h write
 WR0 ← 0106h write
 WR0 ← 0162h write : 자동원점복귀 실행시작

◎Limit신호를 이용한 원점복귀의 예

한쪽의 limit신호를 원점신호로 대응하는 방식입니다. 다만, 다음의 2항의 조건이 있습니다.

- 고속 검출동작을 실시하는 경우는, Limit 신호가 Active 되는 위치에서 기계적 Limit 까지의 거리 내에서 충분히 감속 정지할 수 있는 것.
- 자동원점복귀를 시작하는 위치가 Limit 검출 영역 밖에 있지 않을 것.
여기에서는 -방향 Limit신호를 원점신호로 대응하는 예를 나타냅니다.
 - XLMT-입력을 그림과 같이 XIN0(XIN0+, XIN0-)와 XIN1 입력 단자에 접속합니다.
 - Step1의 고속서치를 실시하기 때문에 Limit 정지 Mode를 감속정지로 설정합니다. (4.5절 WR2/D2bit)
 - XLMT-, XIN0, XIN1 신호의 논리 Level을 모두 같게 설정합니다. (4.5절 WR2/D4, 4.4절 WR1/D0, 2 bit)
 - 확장 Mode 설정의 WR7/D10(Limit 신호 사용) Bit를 1로 합니다.



[동작]

그림과 같이, Step 1은 -방향으로 고속으로 Limit 까지 이동합니다. -Limit 신호가 Active되면 감속정지하고 Step 2로 진행됩니다. Step 2의 불규칙 동작 ②에 의해서 역방향으로 Limit를 탈출하고 나서 저속으로 검출방향으로 이동 후, Limit 신호 Active를 검출하여 정지합니다. 자동원점 복귀시작 위치가 Limit 내에 있을 때는(그림의 A점) Step 1의 동작은 실행되지 않고, Step 2부터 시작됩니다.

[Parameter 및 Mode 설정]

WR0 ← 010Fh write : X축선택

WR1 ← 0000h write : 입력 신호 논리 설정 : XIN0:Low active, XIN1:Low active (4.4절 참조)

WR2 ← 0004h write : D4 0 -limit 신호 논리 : Low active (4.5절 참조)
: D2 1 limit 정지 mode : 감속 정지
: 확장 Mode 설정

WR6 ← 5F00h write : WR6에 입력 신호 필터의 Mode를 기록 (2.8절 참조)
: D15~D13 010 필터 지연:512μsec
: D8 1 XLMTM, XIN1, 0 신호 : 필터 유효

WR7 ← 054Fh write : WR7에 자동 원점 복귀의 Mode를 기록
: D15~D13 000
: D12 0
: D11 0
: D10 1 Limit 신호를 원점 신호로서 사용 : 유효
: D9 0 Z상신호 AND 원점 신호 : 무효
: D8 1 논리/실위치 Counter clear : 유효
: D7 0 Step 4 이동 방향 : +방향

WR0 ← 0160h write : D6 1 Step 4 : 유효

WR6 ← 3500h write : D5 0 Step 3 검출 방향 :

WR7 ← 000Ch write : D4 0 Step 3 : 무효

WR0 ← 0100h write : D3 1 Step 2 검출 방향 : -방향

WR6 ← 004Ch write : D2 1 Step 2 : 유효

WR0 ← 0102h write : D1 1 Step 1 검출 방향 : -방향

WR6 ← 0064h write : D0 1 Step 1 : 유효

WR0 ← 0104h write : X축으로 확장 mode 설정 명령 기록

WR6 ← 07D0h write : Range : 8,000,000(배율:10)

WR0 ← 0105h write : 가감 속도도 : 95,000pps/sec

WR6 ← 0032h write : 95000/125/10=76

WR0 ← 0161h write : 기동속도 : 1000pps

WR6 ← 0DACH write : Step 1, 4의 속도 : 20000pps

WR7 ← 0000h write : Step 2의 속도 : 500pps

WR0 ← 0106h write : Offset 이동 Pulse량 : 3500

WR0 ← 0162h write : 자동원점복귀 실행시작

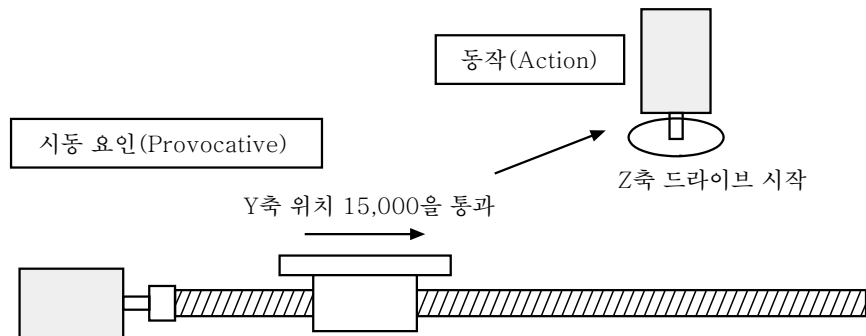
[Limit 신호 사용시의 주의]

- Step1, 2의 검출방향은 반드시 같은 방향으로 합니다. 또, Step 3(Z상서치) 동작을 실시하는 경우는 Step1, 2의 방향과는 반대 방향으로 합니다. Step4(Offset 이동)는 Step1, 2의 방향과 반대로 설정합니다.
- 반드시 Limit active 구간 밖에서 자동원점복귀를 종료시키도록 하십시오.
- Step 3 동작의 경우, Z상신호와 원점신호(IN1)의 AND는 적용되지 않습니다. 확장 Mode 설정의 WR7/D9(SAND) bit는 반드시 0으로 하여야 합니다.

2.6 동기 동작

동기 동작은 각 축 사이와, 다른 디바이스 사이에 있어 Drive 시작・정지 등의 동작(Action)을 연계시키는 기능입니다. 예를 들면 다음과 같은 동작을 실행할 수가 있습니다.

- 예 1 : Y축이 15,000을 통과하면 Z축의 Drive를 시작한다.



- 예 2 : X 축이 -320000을 통과하면 Y, Z 축의 Drive를 정지시킨다.
- 예 3 : 입력신호가 들어오면 X, Y, Z 축의 위치 data를 저장한다.

일반적으로, 이러한 동기 동작은 프로그램을 작성하여 실행할 수도 있지만, 소프트웨어 실행의 지연시간이 허락되지 않을 때, 이 기능을 사용하면 편리합니다. 동기 동작은 지정된 기동 요인이 발생하면 즉시 지정된 동작을 실행합니다. 이 동작은 모션 컨트롤 IC의 간섭 없이 실행되기 때문에 정밀한 작업의 동기가 필요한 경우 유익합니다.

동기 동작은 동기 동작 Mode register로 기동 요인과 동작을 다음과 같이 설정합니다.

WR6 register에 기동 요인(Provocative)과 다른 축 기동을 지정하여, WR7 register에 동작(Action)을 지정하고 나서 WR0 register에 축 지정과 함께 동기 동작 Mode 설정 명령 64h를 설정합니다.

기동 요인은 아래와 같은 WR6 register로 지정하는 10종류, 동작은 WR7로 지정하는 14종류가 있습니다.

WR6	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	AXIS3	AXIS2	AXIS1	0	0	0	CMD	LPRD	IN3↓	IN3↑	D-END	D-STA	P≥C-	P<C-	P<C+	P≥C+
	축 지정								기동 요인(Provocative)							

기동 요인, 다른 축 기동의 Bit를 1로 설정하면 유효, 0을 설정하면 무효가 됩니다.

- | | | |
|--------|--------|--|
| D0 | P ≥ C+ | 논리/실제위치 Counter가 COMP+ register 보다 크다.
(논리/실제위치 counter의 선택은 WR2/D5(CMPSEL) Bit로 설정합니다.) |
| D1 | P < C+ | 논리/실제위치 Counter가 COMP+ register보다 작다. |
| D2 | P < C- | 논리/실제위치 Counter가 COMP- register보다 작다. |
| D3 | P ≥ C- | 논리/실제위치 Counter가 COMP- register보다 크다. |
| D4 | D-STA | Drive 시작. |
| D5 | D-END | Drive 종료. |
| D6 | IN3 ↑ | nIN3 신호가 Low에서 Hi level로 변경. |
| D7 | IN3 ↓ | nIN3 신호가 Hi에서 Low level로 변경. |
| D8 | LPRD | 논리위치 Counter 명령(10h) 설정.
(자/외 축의 동작(Action)에 의한 논리 위치 Counter(LP) 저장, 실제 위치 Counter(EP) 저장 등을 설정하여 동시 읽기 가능.) |
| D9 | CMD | 동기 동작 기동 명령(65h) 설정. |
| D15~13 | | AXIS3~1 설정 축의 기동 요인에 의해 동작시키는 것 외 축을 지정한다. 1:유효 |

현재 지정된 축(자축)	D15 (AXIS3)	D14 (AXIS2)	D15 (AXIS3)
X	U축 기동	Z축 기동	Y축 기동
Y	X축 기동	U축 기동	Z축 기동
Z	Y축 기동	X축 기동	U축 기동
U	Z축 기동	Y축 기동	X축 기동

WR7	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	INT	OUT	0	0	VLSET	OPSET	EPSET	LPSET	EPSAV	LPSAV	ISTOP	SSTOP	CDRV-	CDRV+	FDRV-	FDRV+
동작(Action)																

동작(Action) 지정의 각 bit를 1로 설정하면 유효, 0으로 설정하면 무효가 됩니다.

- D0 FDRV+ +방향 정량 drive.
- D1 FDRV- -방향 정량 drive.
- D2 CDRV+ +방향 연속 drive.
- D3 CDRV- -방향 연속 drive.
- D4 SSTOP drive 감속정지.
- D5 ISTOP drive 즉시정지.
- D6 LPSAV 현재의 논리위치 Counter(LP)를 동기 버퍼 Register(BR)에 저장. LP → BR
- D7 EPSAV 현재의 실제위치 Counter(EP)를 동기 버퍼 Register(BR)에 저장. EP → BR
- D8 LPSET WR6, WR7 register의 값을 논리위치 Counter(LP)에 저장. LP ← WR6, 7
◎2.6.3 동기 동작 주의-(3) 참조.
- D9 EPSET WR6, WR7 register의 값을 실제 위치 Counter(EP)에 저장. EP ← WR6, 7
◎2.6.3 동기 동작 주의-(3) 참조.
- D10 OPSET WR6, WR7 register의 값을 출력 Pulse 수(P)에 설정. P ← WR6, 7
◎2.6.3 동기 동작 주의-(3) 참조.
- D11 VLSET WR6 register의 값을 Drive 속도(V)에 설정. V ← WR6
◎2.6.3 동기 동작 주의-(3) 참조.
- D14 OUT 사용안함. (0으로 설정)
- D15 INT Interrupt 신호(INTN)발생.
Interrupt 신호 (INTN) 가 Low active 되면 Interrupt를 발생시킨 축의 RR3/D9 (SYNC)비트가 1을 나타냅니다. Interrupt를 발생시킨 축의 RR3 register가 모션 컨트롤 IC에 의해 판독되면 RR3 register의 비트는 0으로 Clear되어 Interrupt 출력 신호는 Hi-Z로 돌아옵니다.
Reset시에는 모든 기동 요인, 동작이 무효가 됩니다.

2.6. 1 동기 동작의 예

◎예 1. Y축이 15,000 통과 시 → Z축 +방향 정량 drive 시작

Parameter, Commend를 아래와 같이 설정합니다.

Y축 Drive 시작 후, Y축이 15,000 pulse를 통과하면, Z축의 +방향 정량 Drive가 시작합니다.

WR6 ← 3500h

WR7 ← 000Ch

Y, Z축 Range : 800,000(배율:10)

WR0 ← 0600h

WR6 ← 0190h

WR7 ← 0000h

Y, Z축 가속도 : $400 \times 125 \times 10 = 500\text{kpps/sec}$

WR0 ← 0602h

WR6 ← 0032h

WR7 ← 0000h

Y, Z축 기동속도 : $50 \times 10 = 500\text{pps}$

WR0 ← 0604h

WR6 ← 0BB8h
 WR7 ← 0000h
 WR0 ← 0605h

Y, Z축 Drive 속도 : $3000 \times 10 = 30\text{kpps}$

WR6 ← C350h
 WR7 ← 0000h
 WR0 ← 0206h

Y축 출력 Pulse 수 : 50,000

WR6 ← 2710h
 WR7 ← 0000h
 WR0 ← 0406h

Z축 출력 Pulse 수 : 10,000

WR6 ← 3A98h
 WR7 ← 0
 WR0 ← 020Bh

Y축 COMP+에 15,000을 Set

WR6 ← 0
 WR7 ← 0
 WR0 ← 0609h

Y, Z축 논리 Counter(LP) clear

WR6 ← 2001h
 WR7 ← 0000h
 WR0 ← 0264h

기동 요인 : $P \geq C+$, 다른 축 기동 : Z Y축 동기 동작 Mode 설정
 자축동작 : 없음

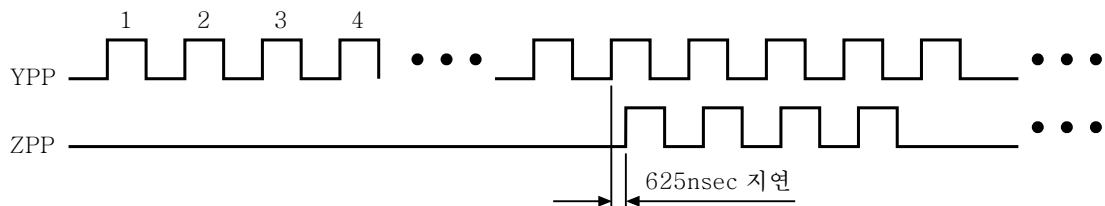
WR6 ← 0000h
 WR7 ← 0001h
 WR0 ← 0464h

자축동작 : +방향 정량 Drive Z축 동기 동작 Mode 설정

WR0 ← 0220h

Y축 +방향 정량 Drive 시작

Y축 15,000번째 Pulse 시작에서 Z축의 제 1pulse의 시작까지의 지연 시간은 5SCLK (625nsec CLK=16MHz) 입니다.



◎예 2. X 축이 -320,000 통과 → Y, Z 축 Drive 정지

WR6 ← 3500h
 WR7 ← 000Ch X, Y, Z축 Range : 800,000(배율:10)
 WR0 ← 0700h

WR6 ← 0190h
 WR7 ← 0000h X, Y, Z축 가속도 : $400 \times 125 \times 10 = 500\text{kpps/sec}$
 WR0 ← 0702h

WR6 ← 0032h
 WR7 ← 0000h X, Y, Z축 기동속도 : $50 \times 10 = 500\text{pps}$
 WR0 ← 0704h

WR6 ← 0BB8h
 WR7 ← 0000h X, Y, Z축 Drive 속도 : $3000 \times 10 = 30\text{kpps}$
 WR0 ← 0705h

WR6 ← A120h
 WR7 ← 0007h X축 출력 Pulse 수 : 500,000
 WR0 ← 0106h

WR6 ← 1E00h
 WR7 ← FFFBh X축 COMP-에 -320,000
 WR0 ← 010Ch

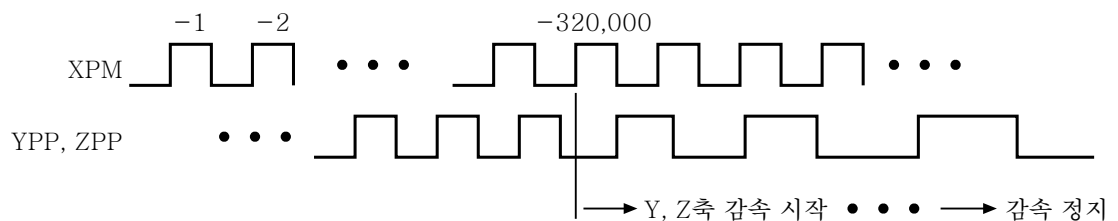
WR6 ← 0
 WR7 ← 0 X축 논리 counter(LP) clear
 WR0 ← 0109h
 WR6 ← 6004h 기동 요인 : $P < C-$, 다른 축 기동 : Y, Z X축 동기 동작 Mode 설정
 WR7 ← 0000h 자축동작 : 없음
 WR0 ← 0164h

WR6 ← 0000h
 WR7 ← 0010h 자축동작 : 감속 정지 Y, Z축 동기 동작 Mode 설정
 WR0 ← 0664h

WR0 ← 0622h Y, Z축 +방향 연속 Drive 시작

WR0 ← 0121h X축 -방향 정량 Drive 시작

예 2에서는 Y, Z축을 연속 Drive 시작 후, X축을 -방향의 정량 Drive로 시작합니다. X축이 -320,000 Pulse를 통과하면 Y, Z축은 감속 정지합니다.



Y, Z 축의 동기 동작 지정을 즉시정지로 지정하면 X 축이 -320,000 pulse를 통과하면 Y, Z 축은 즉시 정지합니다.

◎예 3. 입력 신호(XIN3) 입력 → X, Y, Z 축의 위치 Data를 저장

X, Y, Z 의 3축 Drive 시작 후, XIN3 신호가 입력되면 3축의 논리 위치 Counter 값을 각 축의 버퍼 Register (BR)에 저장합니다. 동작 절차는 X축에 Interrupt 출력 신호(INTN)를 Low active시켜 각 축의 버퍼를 읽어냅니다.

WR6 ← 3500h
 WR7 ← 000Ch X, Y, Z축 Range : 800,000(배율:10)
 WR0 ← 0700h

WR6 ← 0190h
 WR7 ← 0000h X, Y, Z축 가속도 : $400 \times 125 \times 10 = 500\text{kpps/sec}$
 WR0 ← 0702h

WR6 ← 0032h
 WR7 ← 0000h X, Y, Z축 기동속도 : $50 \times 10 = 500\text{pps}$ WR0 ← 0704h

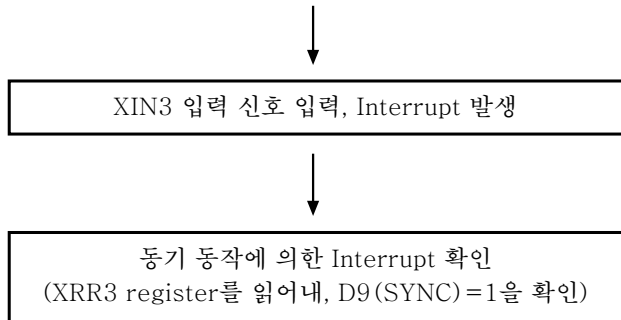
WR6 ← 0BB8h
WR7 ← 0000h X, Y, Z축 Drive 속도 : $3000 \times 10 = 30\text{kpps}$
WR0 ← 0705h

WR6 ← 0
WR7 ← 0 X, Y, Z축 논리 Counter(LP) clear
WR0 ← 0709h

WR6 ← 6080h 기동 요인 : XIN3↓, 다른 축 기동 : Y, Z X축 동기 동작 Mode 설정
WR7 ← 8040h 자축동작 : LP 저장, Interrupt 발생
WR0 ← 0164h

WR6 ← 0000h
WR7 ← 0040h 자축동작 : LP 저장 Y, Z축 동기 동작 Node 설정
WR0 ← 0664h

WR0 ← 0722h X, Y, Z축 +방향 연속 Drive 시작



WR0 ← 0114h
RR6 → read X축 버퍼 Read
RR7 → read
WR0 ← 0214h
RR6 → read Y축 버퍼 Read
RR7 → read

WR0 ← 0414h
RR6 → read Z축 버퍼 Read
RR7 → read

◎예4. 정량 Pulse drive의 연속 동작

동기 동작 기능을 사용하면 Drive 종료 후, 다음 Drive를 시작시켜 연속적인 정량 Drive를 실행하는 일이 생깁니다.
아래 예는, +15,000 이동 종료 후, 즉시 -5,000으로 이동합니다.

WR6 ← 3500h
WR7 ← 000Ch X축 Range : 800,000(배율:10)
WR0 ← 0100h

WR6 ← 0190h
WR7 ← 0000h X축 가속도 : $400 \times 125 \times 10 = 500\text{kpps/sec}$
WR0 ← 0102h

WR6 ← 0032h
WR7 ← 0000h X축 기동속도 : $50 \times 10 = 500\text{pps}$
WR0 ← 0104h

WR6 ← 0BB8h
WR7 ← 0000h
WR0 ← 0105h

X축 Drive 속도 : $3000 \times 10 = 30\text{kpps}$

WR6 ← 0
WR7 ← 0
WR0 ← 0109h

X축 논리 Counter(LP) clear

WR6 ← 3A98h
WR7 ← 0000h
WR0 ← 0106h

X축 출력 Pulse 수 : 15,000

WR6 ← 0020h
WR7 ← 8402h
WR0 ← 0164h

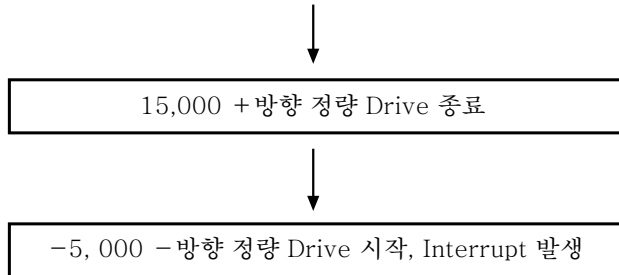
기동 요인 : Drive 종료
자축동작 : P ← WR6, 7 X축 동기 동작 Mode 설정
-방향 정량 Drive interrupt 발생

WR6 ← 1388h
WR7 ← 0000h

다음 Drive의 출력 Pulse 수 : 5,000

WR0 ← 0120h

X축 +방향 정량 Drive 시작



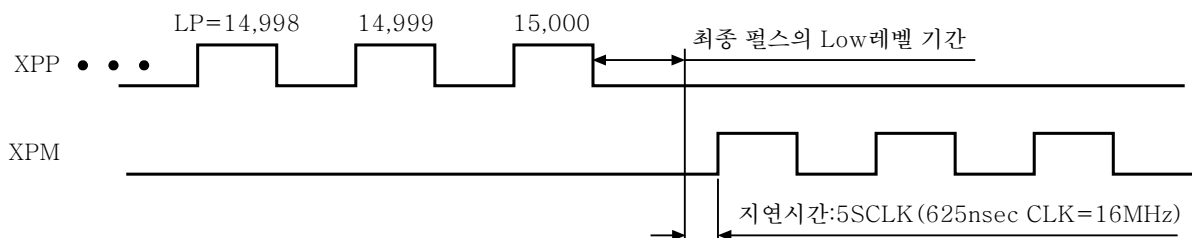
XRR3 → read

동기 동작에 의한 인트럽트 발생 확인
D9(SYNC)=1을 확인.

WR6 ← 0000h
WR7 ← 0000h
WR0 ← 0164h

동기 동작 Mode의 해제 인트럽트 루틴내의 처리

+15,000 이동 종료부터 -5,000의 이동 시작까지의 지연시간은 5SCLK (625nsec CLK=16MHz 시) 입니다.



예 4에서 -5,000의 Drive를 시작하는 것과 동시에 Interrupt를 발생시켜, Interrupt 루틴 내에서 동기 동작 Mode를 해제하고 있습니다. 만약, 해제하지 않으면 -방향 정량 Drive만 실행합니다.

또한, +15,000으로 이동 중에 +방향 Limit(LMT+)나 긴급정지(EMG)등으로 Drive가 중단했을 경우에도, 다음의 -5,000의 Drive는 실행되어 버립니다. 이러한 현상이 시스템상 문제가 되는 경우에는 동기 동작의 사용은 주의하십시오.

2.6.2 동기 동작의 지연시간

동기 동작의 지연시간은, 아래 표와 같이 기동 요인 발생으로부터의 지연과 동작(Action)까지 지연의 합입니다.

◎기동 요인 발생으로부터의 지연

1SCLK=125nsec (CLK=16MHz 때)

시동 요인	지연 시작의 정의		지연 시간(SCLK)		
			최소	표준	최대
P≥C+ P<C+ P<C- P≥C-	P=LP	LP값이 CMP+/- register와 비교 조건에 일치할 때 드라이브 펄스의 ↑까지		1	
	P=EP(A/B 상 입력)	EP 값이 CMP+/- register와 비교 조건에 일치할 때의 nECA/B 입력 신호의 ↑↓까지	3		4
D-STA	드라이브 명령 기록 시 WRN 신호의 ↓까지		1		2
D-END	최종 드라이브 펄스의 Low 레벨 종료까지			1	
IN3 ↑	nIN3 신호의 ↑까지 (내장 필터 무효 시)		0		1
IN3 ↓	nIN3 신호의 ↓까지 (내장 필터 무효 시)		0		1
LPRD	LP 판독 명령(10h)을 기록하고 WRN 신호의 ↓까지		0		1
CMD	동기 동작 시동 명령(65h)을 기록하고 WRN 신호의 ↓까지		0		1

◎동작(Action)까지의 지연

1SCLK=125nsec (CLK=16MHz 때)

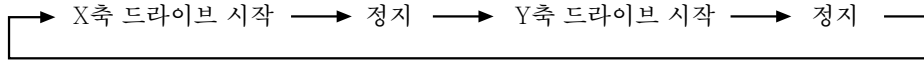
동작	지연 시작의 정의	지연 시간(SCLK)
FDRV+ FDRV- CDRV+ CDRV-	제 1 드라이브 펄스의 ↑까지	4
SSTOP	감속을 시작할 때까지	* 1
ISTOP	드라이브를 정지할 때까지	* 1
LPSAV EPSAV	LP, EP의 값이 BR(버퍼)에 세이브될 때까지	1
LPSET EPSET OPSET VLSET	WR6, 7의 값이 LP, EP, P, V에(로) 세트될 때까지	1
INT		1

*1 : 현재 출력중의 1Drive pulse가 종료할 때까지의 시간

예를 들면, IN3 입력신호의 ↑부터 논리위치 Counter(LP)를 동기 버퍼 Register(BR)에 저장할 때까지의 지연시간은, IN3 ↑지연 시간(0~1 SCLK)과 LP 저장지연시간(1SCLK)을 합산한, 최소 1SCLK부터 최대 2SCLK가 됩니다. CLK=16MHz일 때는, 최소 125nsec부터 최대 250nsec가 됩니다.

2.6.3 동기 동작의 주의점

- (1) 동기 동작은, 동작(Action)에 Interrupt를 동기 지정하는 등, 희망하는 동기 동작이 시작된 후에는 동기 동작 Mode 설정 명령 64h를 다시 설정하여 동기 동작 지정을 해제해야 합니다. 해제하지 않으면 생각지도 않은 곳에서 동작 하는 경우가 있습니다.
- (2) 동기 동작의 기능을 이용하면 다음과 같이 끝이 없는 Drive가 가능하게 됩니다.



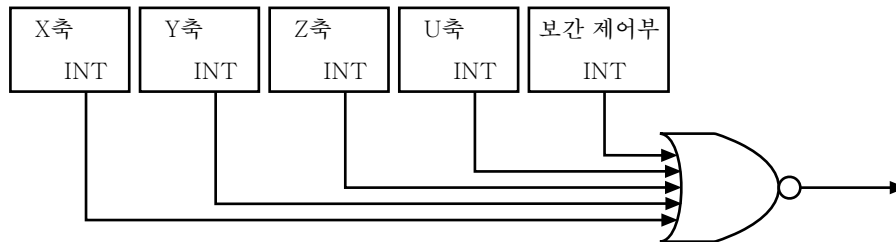
이러한 루프를 정지시키려면, 동기 동작 Mode 설정 명령 64h를 다시 설정하여 기동 요인 및 동작의 각 Bit를 무효로 해야 합니다. Drive 하고 있는 축에 대해 즉시정지 명령이나 감속정지 명령은 루프의 해제에는 적용되지 않습니다.

- (3) 동작(Action) 지정 D8(LPSET), D9(EPSET), D10(OPSET), D11(VLSET)은, 동기 동작이 기동되기 전에 WR6, WR7에 data를 기입해야 합니다. 그런데 동기 동작을 연속적으로 실행할 경우에는 WR6, 7의 기록 타이밍이 늦어 동기 동작의 기동과 겹치면 부정 data가 받아들여지는 일이 있습니다. WR6, 7의 기록은 동기 동작이 기동하지 않는 기간에 설정합니다.
- (4) 현재 Drive 중, Drive 기동의 동작이 발생했을 경우에는 그 동작은 무시됩니다. 또한, 현재 정지 중일 때에도 감속정지, 즉시정지의 동작은 무시됩니다.

2.7 Interrupt

Interrupt의 발생은 X, Y, Z, U 각 축으로 발생시키는 Interrupt와 보간 Drive 중, Bit pattern보간 및 연속보간 시에 발생하는 Interrupt가 있습니다. Interrupt 신호는 INTN신호 1개입니다.

따라서, 아래 그림과 같이 각 축의 Interrupt 신호, 및 Bit pattern 보간, 연속보간의 Interrupt 신호는 모두 OR 게이트를 거쳐 Interrupt를 발생 합니다.



< 그림 2.42 모션 콘트롤 IC내의 Interrupt 신호 경로 >

각 축의 Interrupt 요인 및 보간 Drive 시 Interrupt 요인은, 모두 Interrupt 허가/금지를 설정합니다. Reset 시에는 모두 금지 상태가 됩니다.

◎X, Y, Z, U축의 Interrupt

아래 표는 X, Y, Z, U축의 Interrupt 발생 요인입니다.

	발생의 확인 nRR3 register	Interrupt 발생 요인
D8 (PULSE)	D0 (PULSE)	1 개의 드라이브 펄스 출력(정론리 펄스의 경우 펄스의 ↑에서 발생)
D9 ($P \geq C-$)	D1 ($P \geq C-$)	논리 / 실제위치 카운터가 COMP- register(CM) 보다 크다.
D10 ($P < C-$)	D2 ($P < C-$)	논리 / 실제위치 카운터가 COMP- register(CM) 보다 작다.
D11 ($P < C+$)	D3 ($P < C+$)	논리 / 실제위치 카운터가 COMP+ register(CP) 보다 작다.
D12 ($P \geq C+$)	D4 ($P \geq C+$)	논리 / 실제위치 카운터가 COMP+ register(CP) 보다 크다.
D13 (C-END)	D5 (C-END)	가감속 드라이브에서 정속영역에서 펄스 출력을 종료 .
D14 (C-STA)	D6 (C-STA)	가감속 드라이브에서 정속영역에서 펄스 출력을 시작 .
D15 (D-END)	D7 (D-END)	드라이브 종료 .

각각의 Interrupt 발생 요인은 nWR1 register로 Interrupt 발생의 허가(1)/금지(0)를 설정합니다.

Drive를 시작하여 Interrupt가 발생되면 nRR3 register의 bit가 1이 되어, Interrupt 출력 신호(INTN)가 Low level로 됩니다. Interrupt를 발생시킨 축의 RR3 register를 Read하면 RR3 register의 1로 되어 있던 bit는 0으로 Clear되어 Interrupt출력 신호(INTN)는 Hi-Z으로 돌아옵니다.

【주의】 8 bit data bus의 경우는, RR3L register를 읽으면 모두 Clear되기 때문에, 자동 원점복귀종료 D8(HMEND), 동기 동작 기동 D9(SYNC)를 사용하는 경우에는, 반드시 RR3H를 먼저 읽고 난 후, RR3L register를 설정합니다. 다음의 자동원점복귀종료, 동기 동작 기동의 Interrupt를 나타냅니다.

허가/금지의 설정	발생의 확인 nRR3 register	Interrupt 발생 요인
확장 모드 설정 명령 (60h) WR6/D5 (HMINT)	D8 (H-END)	자동원점출력 종료
동기 동작 지정 명령 (64h) WR7/D15 (INT)	D9 (SYNC)	지정된 기동 요인에 의하여 동기 동작 기동

◎보간 Drive의 Interrupt

()안은 Interrupt clear 방법

허가/금지의 설정 WR5 레지스터	발생의 확인 RR0 register	Interrupt 발생 요인
D14 (CHINT)	D9 (CENEXT)	연속 보간 드라이브로 다음 보간 세그먼트의 데이터와 보간 드라이브 명령이 기록 가능하게 되었다. (다음 보간 드라이브 명령을 기록하면 인터럽트는 클리어 된다.)
D15 (BPINT)	D14, 13 (BPS1, 0)	비트 패턴 보간에서 스택 카운터(SC)가 2에서 1로 변하고, 다음 BP 데이터의 스택이 가능해졌다. (BP 데이터를 스택 한다면 인터럽트는 클리어 된다.)

보간 Drive 중, 발생되는 Interrupt는 보간 Interrupt clear명령 (3Dh)으로도 해제할 수 있습니다.

또한, INTN 출력신호가 Low이면 보간 Drive 종료로 해제되어 hi-Z로 돌아옵니다.

보간 Drive의 Interrupt사용 방법에 대해서는 Bit pattern 보간, 연속보간을 참조하십시오.

2.8 입력 신호 필터

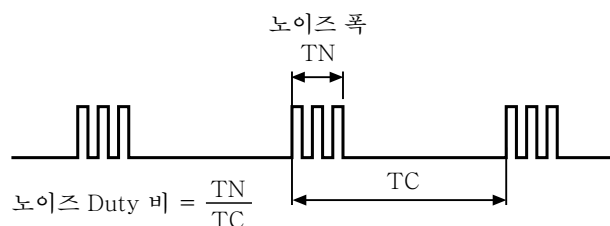
각 입력신호의 입력단은 적분형의 필터를 내장하고 있습니다. 그림 2.43은 X축의 각 입력신호의 필터 구성을 나타내고 있습니다. (Y, Z, U축 동일) 필터의 통과시간은, 그림의 T발진 회로에 의해 정해집니다. 확장 Mode 설정명령 (60h)의 WR6 register D15~13(FL2~0) bit로 8 종류의 필터 통과시간 중에서 하나를 선택합니다.

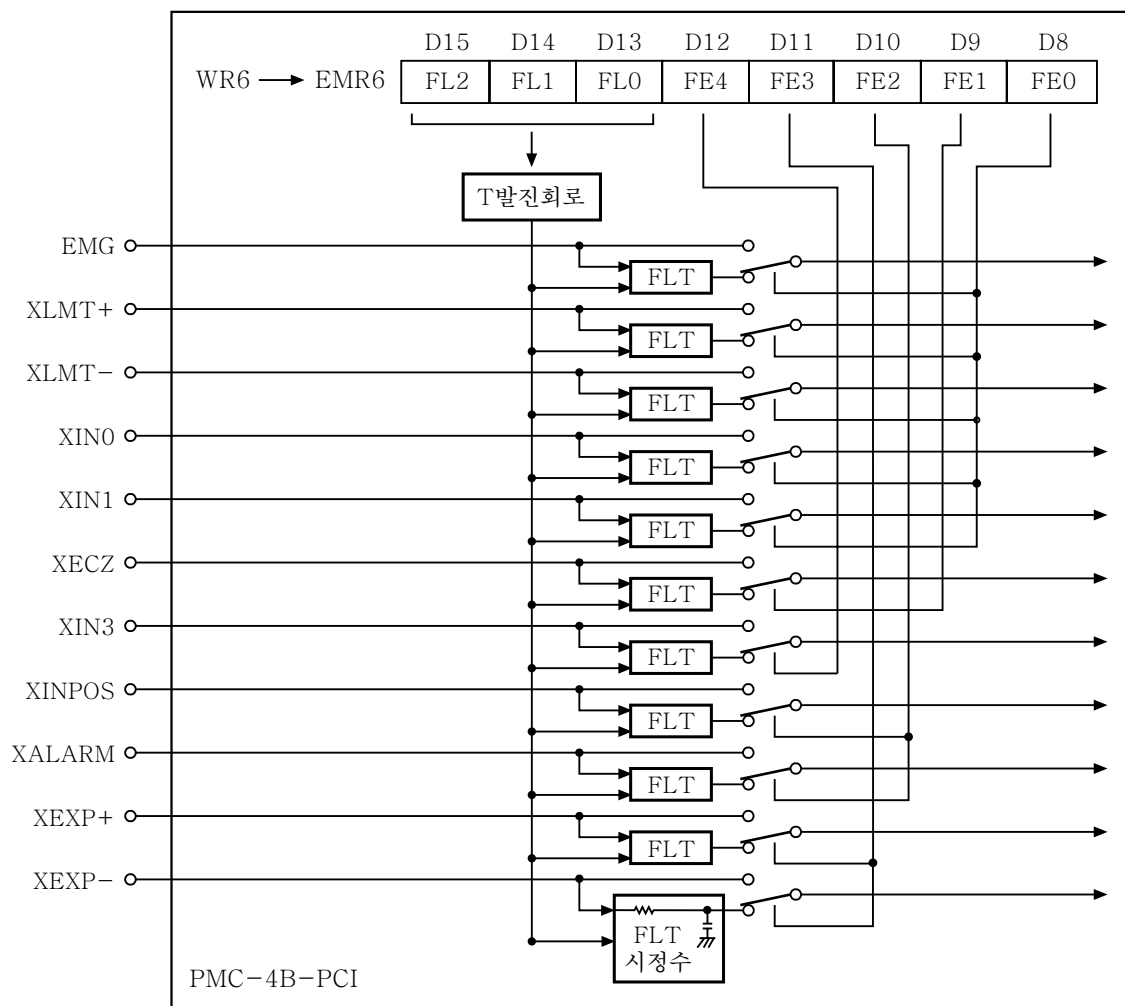
그리고, 같은 WR6 register의 D12~8(FE4~0) bit에 의해 어떤 입력에 필터 기능을 설정할 것인지를 설정합니다. Reset 시에는 확장 Mode의 모든 bit는 Clear 되기 때문에, 모든 입력신호는 필터 기능이 적용되지 않습니다.

확장 Mode 설정명령에 대해서는 6.16절을 참조해 주십시오.

필터의 통과시간은 아래 표와 같이 8 단계가 있습니다. 통과시간을 올리면 제거 가능한 최대 노이즈폭은 증가하지만, 신호의 지연시간이 길 수 있으므로 적절한 값을 설정합니다. 일반적으로는, FL2~0을 2 또는 3으로 설정하는 것을 추천합니다.

FL2 ~ 0	제거 가능한 최대 노이즈 폭	입력신호 지연시간
0	1.75 μ sec	2 μ sec
1	224 μ sec	256 μ sec
2	448 μ sec	512 μ sec
3	896 μ sec	1.024 msec
4	1.792 msec	2.048 msec
5	3.584 msec	4.096 msec
6	7.168 msec	8.192 msec
7	14.336 msec	16.384 msec





<그림 2.43 입력신호 필터 회로 개념도>

노이즈 Duty비(신호대 노이즈 발생 시간의 비율)가, 어떠한 경우에도 1/4이하인 것이 조건입니다.

각 입력 신호의 필터기능을 유효하게 하든지, 입력신호를 필터를 거치지 않고 통과 시킬 것인지는 아래 표와 같이 확장 Mode 설정명령(60h)의 WR6 register D12~8(FE4~0) bit로 설정합니다. 각 Bit에 1을 설정하면 그 신호의 필터기능이 유효가 됩니다.

지정 Bit	유효의 신호
WR6/D8 (FE0)	EMG(*2), nLMT+, nLMT-, nIN0, nIN1
WR6/D9 (FE1)	nECZ
WR6/D10 (FE2)	INPOS, nALARM
WR6/D11 (FE3)	nEXPM
WR6/D12 (FE4)	nIN3

(*2) : EMG 신호는 X축의 WR6 register D8 bit로 설정합니다.

◎입력 신호 필터의 설정 예

EMG와 X, Y 축의 LMT+, LMT-, IN0, IN1, EXP+, EXP- 입력 신호에 512 μ sec 시간지연 필터를 설정하고 X, Y축의 다른 입력신호는 필터기능을 설정하지 않습니다.

Z, U 축의 LMT+, LMT-, IN0, IN1, EXP+, EXP- 입력 신호에 2msec 시간지연 필터를 설정하고 Z, U 축의 다른 입력신호는 필터기능을 설정하지 않습니다.

WR6 ← 4900h write : X, Y축 확장 Mode 설정
 : WR6에 입력신호 필터의 Mode를 기록
 : D15~D13 010 필터 지연:512 μ sec
 : D12 0 IN3 신호:필터 무효
 : D11 1 EXP+, EXP-:필터 유효
 : D10 0 INPOS, ALARM 신호:필터 무효
 : D9 0 ECZ 신호 : 필터 무효
 : D8 1 EMG, LMT+, LMT-, IN1, IN0 신호:필터 유효
 : D7~D0 내장 필터 기능 이외의 Mode(적정치를 설정해 주십시오. 6.16절 참조)

WR7 ← 0000h write : 자동 원점 복귀를 실시하는 경우는 적정치를 설정해 주십시오. (2.5절 참조)

WR0 ← 0360h write : X, Y축으로 확장 Mode 설정 명령 기록
 : Z, U축 확장 Mode 설정

WR6 ← 8900h write : WR6에 입력 신호 필터의 Mode를 기록
 : D15~D13 100 필터 지연:2msec
 : D12 0 IN3 신호:필터 무효
 : D11 1 EXP+, EXP- 신호:필터 유효
 : D10 0 INPOS, ALARM 신호:필터 무효
 : D9 0 ECZ 신호:필터 무효
 : D8 1 LMT+, LMT-, IN1, IN0 신호:필터 유효
 : D7~D0 내장 필터 기능 이외의 Mode
 (적정치를 설정해 주십시오 6.16절 참조)

WR7 ← 0000h write : 자동원점복귀를 실시하는 경우는 적정치를 설정해 주십시오. (2.5절 참조)

WR0 ← 0C60h write : Z, U축으로 확장 Mode 설정 명령 기록

2.9 그 밖의 기능

2.9.1 외부 신호에 의한 Drive 조작

정량 Drive나 연속 Drive를 Commend로 설정하지 않고 신호 입력을 이용하여 시작시키는 기능입니다.

시스템 그리고 제어하는 모터의 축이 많아지면 각 축의 단순 조그 이동과 매뉴얼 조작을 하나의 모션 콘트롤 IC가 모두 실시하려고 한다면 CPU의 부담이 커져, 충분한 응답을 할 수 없게 될 가능성이 있습니다. 외부 신호에 의한 Drive조작 기능으로 이러한 CPU의 부담을 경감할 수가 있습니다. 또, 2상 Encoder 신호를 입력해, 각 축의 조그 이동을 실시할 수가 있습니다. 각 축은 nEXP+와 nEXP- 의 조작 신호 입력을 가지고 있습니다.

정량 Drive mode, 연속 Drive mode에서는 nEXP+ 신호는 +방향, nEXP- 신호는 -방향의 Drive조작을 합니다. WR3 register의 D4, D3 bit로 정량 Drive 또는, 연속 Drive를 설정합니다. 또, 정량 Drive 혹은 연속 Drive에 필요한 Parameter는 Commend에 의한 기동과 같이, 미리 설정합니다. nEXP+와 nEXP- 신호는 일반적으로, Hi level로 설정 합니다. 2상 Encoder 신호 Mode(2상 펄스 입력 신호)에서는 nEXP+ 입력에 A상 신호를 nEXP- 입력에 B상 신호를 접속합니다.

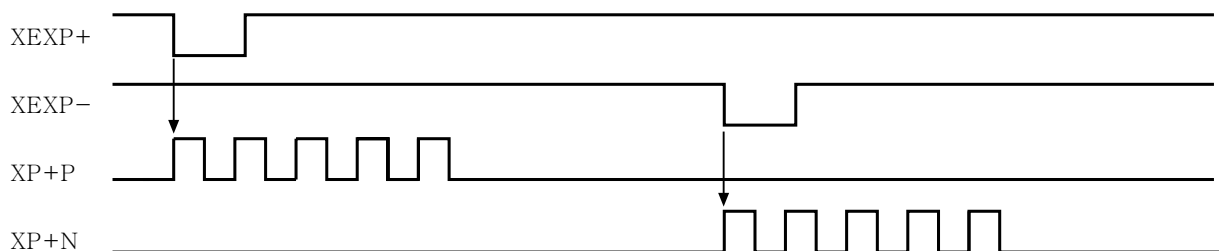
◎정량 Drive mode

WR3 register의 D4, 3 bit를 1,0으로 설정하여 Drive에 필요한 속도 Parameter, 출력 Pulse수를 설정합니다.

nEXP+ 신호를 Hi level에서 Low level로 변경하면 +방향의 정량 Drive가 시작합니다.

nEXP- 신호의 경우 Hi level에서 Low level로 변경하면 -방향의 정량 Drive가 시작합니다.

각 입력 조작 신호의 Low level 폭은, 최소 4 CLK 사이클 이상 필요합니다.

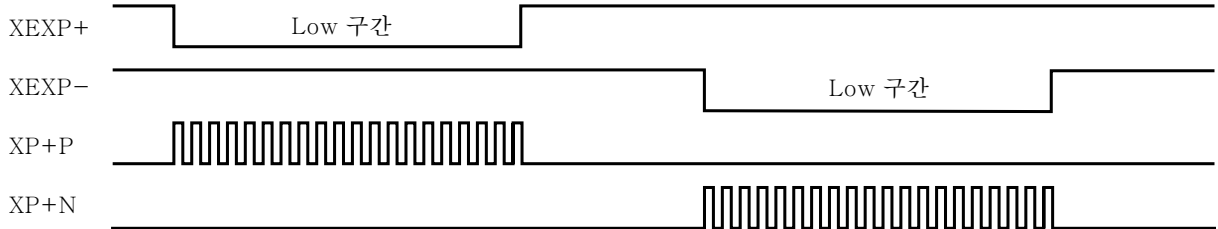


<그림 2.44 외부 조작 신호에 의한 출력 Pulse 5의 정량 Drive의 예>

◎연속 Pulse drive mode

WR3 register의 D4, D3 bit를 0,1으로 설정하여 Drive에 필요한 속도 Parameter를 설정합니다.

nEXP+ 신호를 Hi level에서 Low level로 변경하면 설정된 Low level의 폭으로 연속적인 +방향의 Drive pulse를 출력합니다. nEXP+ 신호를 Low에서 Hi level로 변경하면 가감속 Drive는 감속정지, 정속 Drive는 즉시정지합니다. nEXP- 신호의 경우도 마찬가지로 -방향의 Drive pulse를 연속해서 출력합니다.

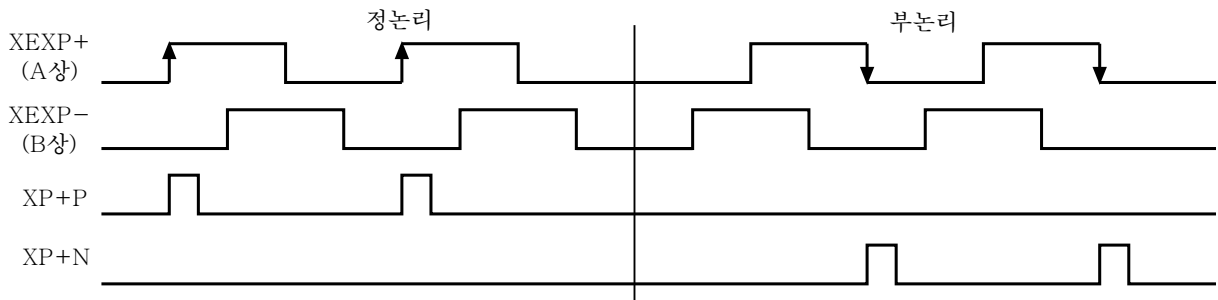


<그림 2.45 외부 조작 신호에 의한 연속 Pulse drive의 예>

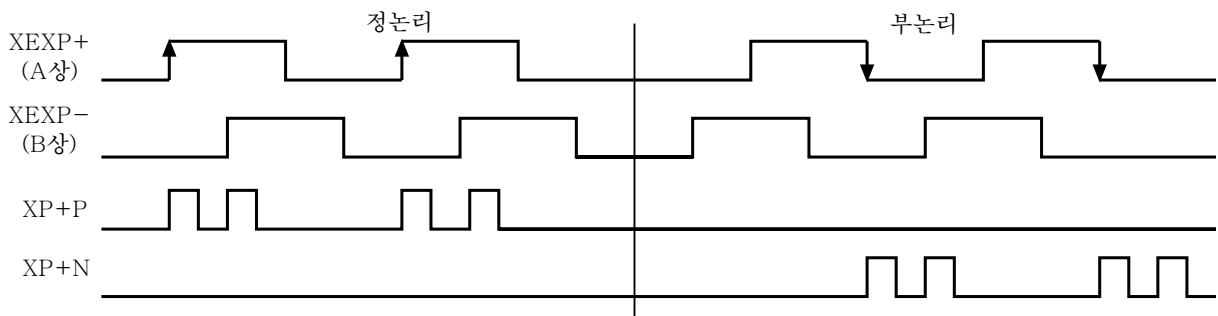
◎2상 Encoder 신호 Mode

WR3 register의 D4, D3 bit를 1,1로 설정하여 Drive에 필요한 속도 Parameter, 출력 Pulse수를 설정합니다.

A상 신호를 nEXP+ 입력에, B상 신호를 nEXP- 입력에 접속합니다. nEXP- 신호가 Low level일 때, nEXP+ 신호가 Hi level로 변경될 경우(상승 에지)는 +정량 Drive, nEXP- 신호가 Low level일 때 nEXP+ 신호가 Low level로 변경되는 경우(하강 에지)는 -정량 Drive가 기동합니다. 출력 Pulse수의 설정이 1이면 nEXP+ 신호의 각 상승, 하강 에지에서 1개의 Drive pulse를 출력합니다. 출력 Pulse수의 설정이 P이면 P개의 Drive pulse를 출력합니다.



<그림 2.46 2상 Encoder 신호 출력 Pulse 1의 Drive의 예>



<그림 2.47 2상 Encoder 신호 출력 Pulse 2의 Drive의 예>

nEXP+ 신호의 Hi, Low 에지 에서 다음 신호의 Hi, Low 에지 사이에 P개의 Drive pulse를 출력 하기 위해서 속도 Parameter는 다음과 같은 조건으로 설정합니다.

$$V \geq F \times P \times 2$$

V : Drive 속도(pps)

P : 출력 Pulse

F : 2상 Encoder 신호의 최고속시의 주파수(Hz)

예를 들면, 2상 Encoder 신호의 최고주파수를 F=500 Hz, 출력 Pulse는 P=1로 하면, Drive 속도 V=1000pps 이상의 값으로 설정합니다. 또, 가감속 Drive는 실시하지 않으므로 기동속도 SV는 Drive 속도 V와 같은 값으로 설정합니다. 다만, 구동 모터가 Step 모터의 경우는 모터의 자기동 주파수를 넘지 않는 범위 내에서 Drive 속도를 설정합니다.

2.9.2 Pulse 출력 방식의 선택

Drive 출력 방식은 아래 표와 같이 2개의 Pulse 출력 방식을 선택할 수가 있습니다. 독립 2 Pulse 방식은 +방향 Drive시에는 nP+P/nP+N에, -방향 Drive시에는 nP-P/nP-N에 Drive pulse를 출력합니다.

또, 1 Pulse 방식은 nP+P/nP+N로 Drive pulse를 출력, nP-P/nP-N은 방향을 결정합니다.

(Pulse/방향은 정논리 설정 시)

펄스 출력 방식	드라이브 방향	출력 신호 파형	
		nP+P/nP+N 신호	nP-P/nP-N 신호
독립 2 Pulse 방식	+방향 드라이브 출력시		Low 레벨
	-방향 드라이브 출력시	Low 레벨	
1 Pulse 방식	+방향 드라이브 출력시		Low 레벨
	-방향 드라이브 출력시		Hi 레벨

Pulse 출력 방식의 선택은 WR2 register의 D6 (PLSMD) bit로 설정 합니다. 또, Pulse 출력, 방향 설정 및 논리 Level도 선택할 수가 있습니다.

【주의】 1 pulse 방식의 경우는 Pulse 신호(nPLS)와 방향 신호(nDIR)가 출력되는 타이밍을 14.2, 14.3절로 확인해 주십시오.

2.9.3 Pulse 입력 방식의 선택

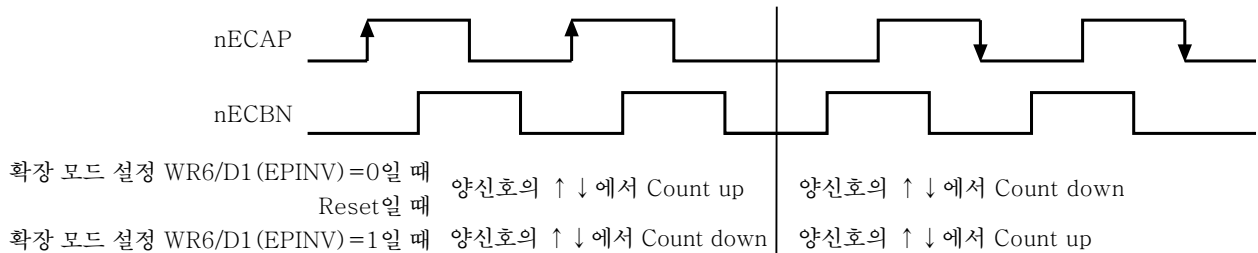
실제위치 Counter의 Up/Down count 입력이 되는 Encoder pulse 입력은 2상 Pulse 입력과 Up/Down pulse 입력 중에서 선택할 수가 있습니다.

◎2상 Pulse 입력 Mode

WR2 register의 D9 (PINMD) bit를 0으로 설정하면 2상 Pulse 입력 Mode가 됩니다. 이 Mode에서는 정논리 Pulse로 A상이 진행되고 있을 때는 Count up하고 B상이 진행되고 있을 때는 Count down 합니다.

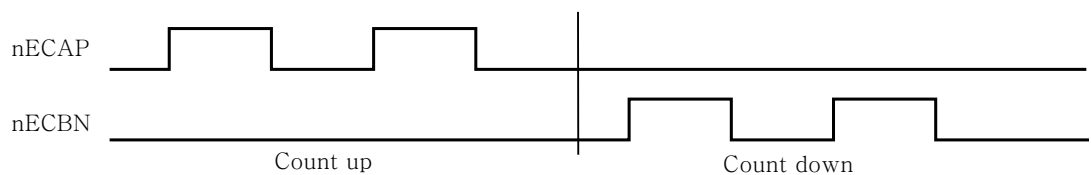
양신호의 상승에지, 하강에지에서 Count up, Down합니다. 확장 Mode 설정으로 실제위치 Counter 증감 반전 bit (WR6/D1)를 1로 하면 실제위치 Counter의 Up down 동작이 거꾸로 됩니다. (6.16절 참조)

또, 2상 Pulse 입력 Mode에서는 입력 Pulse를 1/2, 1/4로 분주시킬 수도 있습니다.



◎Up/down pulse 입력 Mode

WR2 register의 D9 (PINMD) bit를 1로 설정하면 Up/Down pulse 입력 mode가 됩니다. nECAP/nECAN가 Count up 입력, nECBP/nECBN가 Count down 입력이 됩니다. 각각, Pulse의 ↑에서 Count 합니다.



Pulse 입력 방식의 선택은 WR2 register의 D9 (PINMD) bit로, Encoder 2상 Pulse 입력의 분주비는 D11, D10 (PIND1, 0) bit로 설정 합니다.

【주의】 입력 Pulse의 Pulse폭, Pulse 주기 등에는 시간 규정이 있습니다. 13장 13.2.5 입력 Pulse를 참조하십시오.

2.9.4 하드웨어 Limit 신호

하드웨어 Limit 신호(nLMT+, nLMT-)는 +방향/-방향의 drive pulse를 정지시키는 신호 입력입니다. Limit신호의 논리 Level과 Limit신호가 Active되었을 때 감속정지 또는, 즉시정지는 Commend로 선택할 수가 있습니다. WR2 register의 D3, D4(HLMT+, HLMT-), D2(LMTMD) bit로 설정합니다.

2.9.5 서보(Servo)모터 Driver 대응의 신호

서보모터 Driver와의 접속을 위한 입력 신호로서 위치 결정완료신호를 입력하는 nINPOS와 Alarm 신호를 입력하는 nALARM이 있습니다. 각각의 신호의 유효/무효 및 논리 Level을 설정할 수가 있습니다.

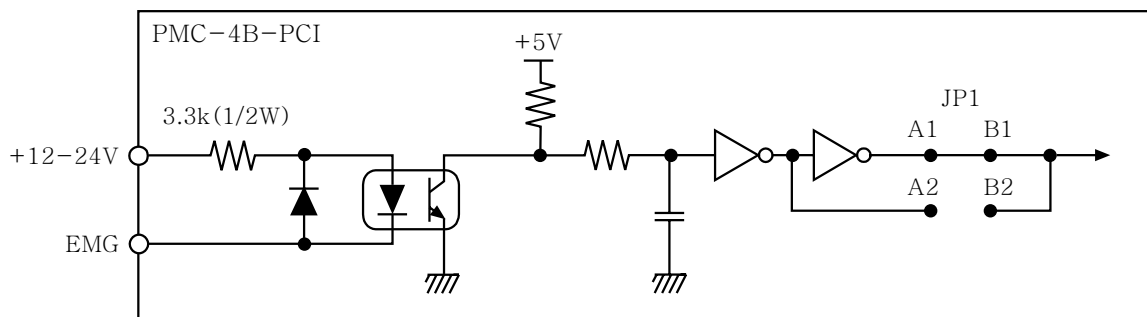
설정은 WR2 register의 D15~D12 bit로 실시합니다. nINPOS 입력신호는 서보모터 Driver의 위치결정완료신호에 대응합니다. 유효로 설정하면 Drive 종료 후, nINPOS 입력 신호가 Active 되면, RR0주소status register의 n-DRV bit가 0으로 돌아옵니다. nALARM 입력신호는 서보모터 Driver로부터의 Alarm신호를 수신합니다.

유효로 설정하면 nALARM 입력신호를 항상 감시하며, Active 상태의 경우는 RR2 register의 D4(ALARM) bit가 1이 됩니다. Drive중이면, Drive를 즉시정지합니다.

이러한 서보모터 Driver용 입력 신호는 RR5, 6 register로 그 상태를 상시 읽어낼 수가 있습니다.

2.9.6 긴급 정지

4축 모든 Drive를 긴급 정지시키기 위한 입력 신호로서 EMG 신호가 있습니다. EMG 신호는 일반적으로 Hi level입니다. Low level이 되면 Drive중의 전축이 즉시정지하고 전축의 RR2 register의 D5(EMG) Bit가 1이 됩니다. EMG 신호는 논리 Level을 선택할 수가 없기 때문에 PMC-4B-PCI Board의 JP1 점프핀을 이용하여 논리레벨을 설정합니다.

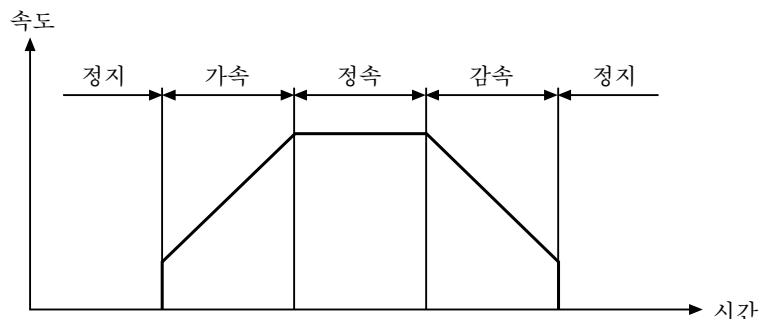


4축에 대해서 긴급정지를 설정하기 위해서는 다음과 같은 방법이 있습니다.

- 4 축에 대해 동시에 즉시정지 명령을 발행한다.
WR0 register에 4축 모두를 지정하여 즉시정지명령(27h)을 기입합니다.
- 소프트웨어 Reset을 설정한다.
WR0 register에 8000 h를 기입하면 소프트웨어 Reset가 걸립니다.

2.9.7 Drive 상태의 출력

각 축의 Drive중/정지 상태는 RR0 register D3~0(n-DRV) bit에 출력됩니다.



각 축 Drive중, Drive 속도의 가속/정속/감속 상태는 각 축의 RR1 register의 D2(ASND), D3(CNST), D4(DSND) bit와 nOUT6/ASND, nOUT7/DSND 신호에 출력됩니다. 다만, 신호출력은, 범용 출력신호와 단자를 겸용하고 있기 때문에 Drive 상태를 출력하려면 WR3 register의 D7(OUTSL) bit를 1으로 합니다.

PMC-4B-PCI에서는 범용출력신호 단자로는 사용할 수 없습니다.

각 축 Drive중, Drive 속도의 가속/정속/감속 상태는 각 축의 RR1 register의 D2(ASND), D3(CNST), D4(DSND) bit와 nOUT6/ASND, nOUT7/DSND 신호에 출력됩니다. 다만, 신호출력은, 범용 출력신호와 단자를 겸용하고 있기 때문에 Drive 상태를 출력하려면 WR3 register의 D7(OUTSL) bit를 1으로 합니다.

PMC-4B-PCI에서는 범용출력신호 단자로는 사용할 수 없습니다.

드라이브 상태	레지스터				출력신호	
	RR0/n-DRV	nRR1/ASND	nRR1/CNST	nRR1/DSND	nOUT6/ASND	nOUT7/DSND
정지	0	0	0	0	Low	Low
가속	1	1	0	0	Hi	Low
정속	1	0	1	0	Low	Low
감속	1	0	0	1	Low	Hi

또, S자 가감속 Drive에 있어서 가속도, 감속도의 증가/ 일정/ 감소 상태도, RR1 register의 D5(AASND), D6(ACNST), D7(ADSND) bit로 출력됩니다.

2.9.8 범용 출력 신호

각 축 모두 nOUT7~4의 4개의 범용 출력 신호를 가지고 있지만, 위치 비교출력, Drive상태 출력과 단자를 공용하고 있기 때문에, 출력으로 사용하는 경우는 사용할 수 없습니다. nOUT7~4 신호는 WR3 register의 D7(OUTSL)로 범용 출력을 사용하는 Mode로 설정합니다. 그리고 WR3 register의 D11~8(OUT7~4)의 각 Bit에 출력 Level의 값을 설정 하면 출력됩니다. 범용 출력 신호는 모터 Driver의 여자 OFF, Alarm reset 등에 사용할 수가 있습니다. Reset시에는 nWR3 register의 각 bit는 clear 되어 모든 출력은 Low level로 됩니다.

3. 각 신호의 설명

◎신호의 설명

신호명의 X○○○, Y○○○, Z○○○, U○○○은 각각 X축, Y축, Z축, U축의 입/출력 신호입니다.

또, n○○○의 "n"은, X, Y, Z, U를 표현하고 있습니다. "-F-" 기호가 붙은 입력신호는 모션 콘트롤 IC 내부의 입력단에 적분 필터 회로를 가지고 있습니다. 필터 기능에 대해서는 2.8절을 참조하십시오.

신 호	단자번호	입/출력	신호의 설명
XP+P/ XP+N YP+P/ YP+N ZP+P/ ZP+N UP+P/ UP+N	A38/A39 A78/A48 B38/B39 B47/B48	출력	Pulse + / Pulse : +방향의 Drive pulse를 출력합니다. Reset시의 상태는 Low레벨이며 Drive 동작으로 들어가면 듀티비 50%(정속시)의 정 Pulse가 출력됩니다. 정 Pulse/부 Pulse는 Mode로 선택할 수 있습니다. 또, Mode 선택으로, 1 Pulse 방식이 선택되었을 경우에는, 본 단자에서 Drive pulse가 출력됩니다.
XP-P/ XP-N YP-P/ YP-N ZP-P/ ZP-N UP-P/ UP-N	A40/A41 A49/A50 B40/B41 B49/B50	출력	Pulse - / Direction : -방향의 Drive pulse를 출력합니다. Reset시의 상태는 Low level이며 Drive 동작으로 들어가면 듀티 50%(정속시)의 정 Pulse가 출력됩니다. 정 Pulse/부 Pulse는 Mode로 선택할 수 있습니다. 또, Mode 선택으로, 1 Pulse 방식이 선택되었을 경우에는, 본 단자는 방향 신호가 됩니다.
XECAP/XECAN YECAP/YECAN ZECAP/ZECAN UECAP/UECAN	A15/A16 A23/A24 B15/B16 B23/B24	입력	Encoder-A / Pulse+in : Encoder A상 신호의 입력으로, 모션 콘트롤 IC 내부에서 Up/Down pulse로 변환되어 실제위치 Counter의 Count 입력이 됩니다. 또, Mode 선택을 Up/Down pulse 입력으로 선택하면 Up pulse 입력으로 설정되며, 입력 Pulse의 ↑에서 실제위치 Counter가 Count up 됩니다.
XECBP/XECBN YECBP/YECBN ZECBP/ZECBN UECBP/UECBN	A17/A18 A25/A26 B17/B18 B25/B26	입력	Encoder-B / Pulse-in : Encoder B상 신호의 입력으로, 모션 콘트롤 IC 내부에서 Up/Down pulse로 변환되어 실제위치 Counter의 Count 입력이 됩니다. 또, Mode 선택을 Up/Down pulse 입력으로 선택하면 Down pulse 입력으로 설정되며, 입력 Pulse의 ↑에서 실제위치 Counter가 Count down 됩니다.

신 호	단자번호	입/출력	신호의 설명
XOUT7/DSND YOUT7/DSND ZOUT7/DSND UOUT7/DSND	A37 A46 B37 B46	출력 A	Universal Output7 / Descend : 범용 출력 신호입니다. nOUT7~4 출력은, WR0 register로 축지정 후, WR3 register의 D11~8로 출력 level을 결정합니다. Reset시는 Low가 됩니다. Mode 선택으로, Drive 상태 출력 Mode로 하면 감속 Drive 출력 신호가 됩니다. Drive 명령 실행 중, 감속 상태가 되면 Hi level이 됩니다.
XOUT6/ASND YOUT6/ASND ZOUT6/ASND UOUT6/ASND	A36 A45 B36 B45	출력 A	Universal Output6 / Ascend : 범용 출력 신호입니다. 설정은 nOUT7과 같습니다. Mode 선택으로, Drive 상태 출력 Mode로 하면 가속 Drive 출력 신호가 됩니다. Drive 명령 실행중, 가속 상태가 되면 Hi level이 됩니다.
XOUT5/CMPP YOUT5/CMPP ZOUT5/CMPP UOUT5/CMPP	A35 A44 B35 B44	출력 A	Universal Output5 / Compare- : 범용 출력 신호입니다. 설정은 nOUT7과 같습니다. Mode 선택으로, Drive 상태 출력 Mode로 하면 논리/실제 위치 Counter가 COMP- register보다 작을 때 Hi level로, 클 때 Low level이 됩니다.
XOUT4/CMPP YOUT4/CMPP ZOUT4/CMPP UOUT4/CMPP	A34 A43 B34 B43	출력	Universal Output4 / Compare+ : 범용 출력 신호입니다. 설정은 nOUT7과 같습니다. Mode 선택으로, Drive 상태 출력 Mode로 하면 논리/실제 위치 Counter가 COMP+ register보다 클 때 Hi level로, 작을 때 Low level이 됩니다.
XINPOS YINPOS ZINPOS UINPOS	A13 A21 B13 B21	입력 -F-	Inposition : 서보모터 Driver의 Inposition(위치결정완료) 출력에 대응하는 입력 신호입니다. 유효/무효, 논리 Level은 Commend로 설정할 수 있습니다. 유효로 설정하면 Drive 종료 후, 이 신호가 Active되는 것을 기다리고 나서, 주status register의 n-DRV bit가 0으로 돌아옵니다.
XALARM YALARM ZALARM UALARM	A14 A22 B14 B22	입력 -F-	Servo Alarm : 서보모터 Driver의 Alarm 출력에 대응하는 입력 신호입니다. 유효/무효, 논리 Level은 Mode로 선택할 수가 있습니다. 유효로 설정하고 신호가 Active level이 되면 RR2 register의 ALARM bit가 1이 됩니다.
XLMT+ YLMT+ ZLMT+ ULMT+	A3 A8 B3 B8	입력 -F-	Over Run Limit + : +방향의 Over run limit 신호입니다. +방향의 Drive pulse 출력 중, Active하면 Drive는 감속정지 또는 즉시정지합니다. 필터 기능이 무효인 경우, 2CLK 이상의 Active pulse 폭이 필요합니다. 감속정지/즉시정지, 논리 Level을 Mode로 선택할 수가 있습니다. 또, 이 신호가 Active level이 되면, RR2 register의 HLMT+ bit가 1이 됩니다.
XLMT- YLMT- ZLMT- ULMT-	A4 A9 B4 B9	입력 -F-	Over Run Limit - : -방향의 Over run limit 신호입니다. -방향의 Drive pulse 출력 중, Active하면 Drive는 감속 정지 또는 즉시 정지합니다. 필터 기능이 무효인 경우, 2CLK 이상의 Active pulse폭이 필요합니다. 감속 정지/즉시 정지, 논리 Level을 Mode로 선택할 수가 있습니다. 또, 이 신호가 Active level이 되면, RR2 register의 HLMT- bit가 1이 됩니다.
XINO YINO ZINO UINO	A6 A11 B6 B11	입력 -F-	원점근접서치 동작의 입력 신호로 액티브 레벨을 선택하여 설정합니다. 필터 기능이 무효인 경우, 2CLK 이상의 Active pulse폭이 필요합니다. RR4/RR5 register로 신호의 상태를 읽어낼 수가 있습니다. 오픈 컬렉터 입력입니다. 결선방법은 11.3절을 참조하십시오.

신 호	단자번호	입/출력	신호의 설명
XECAP/N YECAP/N ZECAP/N UECAP/N	A19~A20 A27~A28 B19~B20 B27~B28	입력 -F-	Encoder-C ECZ는 encoder Z상 신호에 할당됩니다.
XIN3, XIN1~0 YIN3, YIN1~0 ZIN3, ZIN1~0 UIN3, UIN1~0	A7, A5, A6 A12, A10, A11 B7, B5, B6 B12, B10, B11	입력 -F-	Input3~0 : Drive 도중에 감속 정지 또는 즉시 정지시키기 위한 신호입니다. 서치 동작의 입력 신호로 사용합니다. 필터 기능이 무효인 경우, 2CLK 이상의 Active pulse 폭이 필요합니다. IN3~IN1 각각에 대하여 유효/무효, 논리 Level을 설정할 수가 있습니다. 자동 원점 복귀에서는 IN1은 원점 신호에, IN0은 원점 근접신호에 할당됩니다. 또, RR4/RR5 register로 신호의 상태를 읽어낼 수가 있습니다.
XEXP+ YEXP+ ZEXP+ UEXP+	A29 A31 B29 B31	입력 -F-	External Operation + : 외부에서 +방향의 Drive를 기동하는 신호입니다. 외부 정량 Pulse drive mode로 하면, 신호의 ↓에서 +정량 Pulse drive가 기동합니다. 외부 연속 Pulse drive mode로 하면, Low level에서 +연속 Pulse drive를 기동합니다. 2상 Encoder 신호 Mode의 경우는, Encoder A상 신호를 입력합니다.
XEXP- YEXP- ZEXP- UEXP-	A30 A32 B30 B32	입력 -F-	External Operation - : 외부로부터 -방향의 Drive를 기동하는 신호입니다. 외부 정량 Pulse drive mode로 하면, 신호의 ↓에서 -정량 Pulse drive가 기동합니다. 외부 연속 Pulse drive mode로 하면, Low level에서 -연속 Pulse drive를 기동합니다. 2상 Encoder 신호 Mode의 경우는, Encoder B상신호를 입력합니다.
EMG	A2	입력 -F-	Emergency Stop : 전 축의 Drive를 긴급정지시키는 입력 신호입니다. 이 신호를 Low level로 하면, 보간 Drive를 포함하여 전 축의 Drive가 즉시 정지하면, 각 축의 RR2 register의 EMG bit가 1이 됩니다. 필터 기능이 무효인 경우, 2CLK 이상의 Low level pulse 폭이 필요합니다. 【주의】 이 신호는, JP1으로 논리Level을 선택할 수 있습니다.

4. Read/Write register

이 장에서는, 각 축을 제어하기 위한 Read/Write register에 대해서 상세하게 기술합니다.

Bit pattern 보간용 Register(BP1P/M, BP2P/M, BP3P/M)에 대해서는 2.4.3절의 Bit pattern 보간을 참조하십시오.

4.1 16 Bit data bus의 Register address

아래 표와 같이, 16 Bit data bus를 사용하는 경우에는 16 bit의 Read/write register를 액세스 하는 Address가 있습니다.

◎16 Bit data bus에 있어서의 Write register

모든 Register는 16 bit 입니다.

ADDRESS A2 A1 A0	REGISTER 기 호	REGISTER	내 용
0 0 0	WR0	Command register	축지정, 명령 코드 설정
0 0 1	XWR1 YWR1 ZWR1 UWR1	X축 Mode register 1 Y축 Mode register 1 Z축 Mode register 1 U축 Mode register 1	각 축의 외부 감속 정지 신호의 논리 Level의 유효/무효 설정. 각 축의 Interrupt 허가/금지 설정.
0 1 0	XWR2 YWR2 ZWR2 UWR2	X축 Mode register 2 Y축 Mode register 2 Z축 Mode register 2 U축 Mode register 2	각 축의 Limit 신호의 Mode 설정. Drive pulse의 Mode 설정. Encoder 입력 신호의 Mode 설정. 서보모터용 신호의 논리 Level의 유효/무효 설정.
	BP1P	BP1P register	Bit pattern 보간 제1축 +방향 bit data 설정.

0 1 1	XWR3 YWR3 ZWR3 UWR3	X축 Mode register 3 Y축 Mode register 3 Z축 Mode register 3 U축 Mode register 3	각 축의 매뉴얼 감속, 감속도, S자 가감속 Mode 설정. 외부 조작 Mode 설정. 범용 출력 OUT7 ~ 4 설정.
	BP1M	BP1M register	Bit pattern 보간 제 1축 -방향 bit data 설정.
1 0 0	WR4	Output register	범용 출력 OUT3 ~ 0 설정(사용 안함)
	BP2P	BP2P register	Bit pattern 보간 제 2축 +방향 bit data 설정.
1 0 1	WR5	보간 Mode register	축지정. 선속일정 Mode, Step 출력 Mode, Interrupt 설정.
	BP2M	BP2M register	Bit pattern 보간 제 2축 +방향 bit data 설정.
1 1 0	WR6	Write data register 1	범용 출력 OUT3 ~ 0 설정(사용 안함)
	BP3P	BP3P register	Bit pattern 보간 제 2축 +방향 bit data 설정.
1 1 1	WR7	Write data register 2	Write data 상위 16 bit(D31~D16) 설정.
	BP3M	BP3M register	Bit pattern 보간 제 3축 -방향 bit data 설정.

- 위의 표와 같이 각 축 모두 WR1, WR2, WR3(Mode register 1, 2, 3)를 가지고 있습니다.
이러한 Register는 동일 Address에 기입하여 실시하게 됩니다. 어느 축의 Mode register에 기입할지는 직전에 설정한 축지정에 의해 정해집니다.
- Bit pattern보간용의 Bit data register BP1P~3P, BP1M~3M은 Reset 직후에는 기입할 수가 없습니다.
이러한 Register의 기입은 BP register 기입 가능명령(36 h)으로 가능하게 됩니다.
BP register 기입 가능명령(36h)을 발행 후, nWR2~3의 기입을 할 수 없게 되기 때문에 Bit pattern 보간으로 Bit data의 기입이 끝나면 BP register 기입 불가명령(37 h)을 발행해야 합니다.
- WR6 register와 BP3P register, WR7 register와 BP3M register에 대해서는 하드웨어적으로 동일 Register를 공용하고 있으므로 주의해 주십시오.
- Reset는 nWR1, nWR2, nWR3, WR4, WR5 register의 모든 bit를 0으로 clear시킵니다.
그 외의Register는 적용되지 않습니다.

◎16 Bit data bus에 있어서의 Read register

모든 Register는 16 bit 입니다.

ADDRESS A2 A1 A0	REGISTER 기 호	REGISTER	내 용
0 0 0	RR0	주 Status register	각 축의 Drive, 에러 상태를 표시. 보간의 Drive, 연속보간시 다음 Data 설정. 원호보간의 상한 설정. BP보간시 Stack counter표시.
0 0 1	XRR1 YRR1 ZRR1 URR1	X축 Status register 1 Y축 Status register 1 Z축 Status register 1 U축 Status register 1	위치 : COMP register 비교 가/감속 상태, 가/감속 증가/감소 상태 표시. 종료 Status의 표시.
0 1 0	XRR2 YRR2 ZRR2 URR2	X축 Status register 2 Y축 Status register 2 Z축 Status register 2 U축 Status register 2	에러 발생 요인 표시. 자동원점복귀 실행 Status 표시.
0 1 1	XRR3 YRR3 ZRR3 URR3	X축 Status register 3 Y축 Status register 3 Z축 Status register 3 U축 Status register 3	Interrupt 발생 요인 표시.
1 0 0	RR4	Input register 1	X축, Y축 입력 신호 상태 표시
1 0 1	RR5	Input register 2	Z축, U축 입력 신호 상태 표시
1 1 0	RR6	Read data register 1	Read data 하위 16 bit(D15~D0) 표시.
1 1 1	RR7	Read data register 2	Read data 상위 16 bit(D31~D16) 표시.

Write register와 같이, 각 축 모두 RR1, RR2, RR3(각 축status register 1,2,3)를 가지고 있습니다. 이러한 Register는 동일 Address로 읽기를 실시하게 됩니다. 어느 축의 Status register를 읽을 지는 직전에 설정한 축지정에 의해 정해집니다.

4.2 8 Bit data bus의 Register address

8 Bit data bus로 액세스 하는 경우는 16 Bit register를 상위 바이트, 하위 바이트로 나누어 액세스 합니다. 아래 표에서, ****L는 16 bit register****의 하위 바이트(D7~D0), ****H는 16 bit register****의 상위 바이트(D15~D8)를 나타내고 있습니다. Commend register(WR0L, WR0H)만은, 반드시 상위 바이트(WR0H)를 먼저, 하위 바이트(WR0L) (을)를 나중에 기입합니다.

◎8 Bit data bus에 있어서의 Write register

ADDRESS A3 A2 A1 A0	Write 하는 Register
0 0 0 0	WR0L
0 0 0 1	WR0H
0 0 1 0	XWR1L, YWR1L, ZWR1L, UWR1L
0 0 1 1	XWR1H, YWR1H, ZWR1H, UWR1H
0 1 0 0	XWR2L, YWR2L, ZWR2L, UWR2L, BP1PL
0 1 0 1	XWR2H, YWR2H, ZWR2H, UWR2H, BP1PH
0 1 1 0	XWR3L, YWR3L, ZWR3L, UWR3L, BP1ML
0 1 1 1	XWR3H, YWR3H, ZWR3H, UWR3H, BP1MH
1 0 0 0	WR4L, BP2PL
1 0 0 1	WR4H, BP2PH
1 0 1 0	WR5L, BP2ML
1 0 1 1	WR5H, BP2MH
1 1 0 0	WR6L, BP3PL
1 1 0 1	WR6H, BP3PH
1 1 1 0	WR7L, BP3ML
1 1 1 1	WR7H, BP3MH

◎8 Bit data bus에 있어서의 Read register

ADDRESS A3 A2 A1 A0	Read 하는 Register
0 0 0 0	RR0L
0 0 0 1	RR0H
0 0 1 0	XRR1L, YRR1L, ZRR1L, URR1L
0 0 1 1	XRR1H, YRR1H, ZRR1H, URR1H
0 1 0 0	XRR2L, YRR2L, ZRR2L, URR2L
0 1 0 1	XRR2H, YRR2H, ZRR2H, URR2H
0 1 1 0	XRR3L, YRR3L, ZRR3L, URR3L
0 1 1 1	XRR3H, YRR3H, ZRR3H, URR3H,
1 0 0 0	RR4L
1 0 0 1	RR4H
1 0 1 0	RR5L
1 0 1 1	RR5H
1 1 0 0	RR6L
1 1 0 1	RR6H
1 1 1 0	RR7L
1 1 1 1	RR7H

4.3 WR0 commend register

각 축에 대하여 축지정을 하여, 명령을 기입하는 Register입니다. Register는 축을 지정하는 bit, 명령 코드를 설정하는 bit 및 Commend reset bit로 구성되어 있습니다.

본 Register에 축지정 및 명령 코드를 기입하면, 그 명령은 즉시 실행됩니다. Drive 속도설정 등의 Data 기입 명령은 WR6, 7 register에 Data를 먼저 기입해야 합니다. 또, Data read명령은 이 Commend register에 명령을 기입하면 내부 회로로부터 RR6, 7 register에 Data가 설정 됩니다. 8bit data bus는 반드시 상위 바이트(H)를 먼저, 하위 바이트(L)를 나중에 기입합니다. 모든 명령 코드의 명령 처리에 필요로 하는 시간은 최대 250nsec(CLK=16 MHz의 경우)입니다.

WRO	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	RESET	0	0	0	U	Z	Y	X	0							

축 지정
명령코드

D6~0 명령 코드를 설정합니다. 명령 코드는 5장 이후의 각 명령의 설명을 참조하십시오.

D11~8 실행 할 축을 지정합니다. 각 축의 해당 bit를 1로 설정하면 그 축이 지정됩니다.

축의 지정은 동시에 여러 축을 지정할 수 있으며 같은 명령을 실행하거나 같은 Parameter값을 입력할 수 있습니다. 다만, Data 읽기 명령의 경우는 1축만 지정하십시오.

보간 관계의 명령에서는, 축지정 bit는 모두 0으로 하십시오.

D15 RESET 모션 컨트롤 IC를 Reset 합니다. 이 bit를 1로하면 모션 컨트롤 IC는 reset 됩니다.

Commend 기입 후, 최대 875nsec(CLK=16 MHz의 경우)동안 모션 컨트롤 IC의 Register를 액세스 할 수 없습니다. 8 Bit data bus의 경우는, WROH(80h)의 기입으로 Reset 됩니다.

RESET bit는, 일반적으로 명령 기입에서는 반드시 0으로 둡니다.

1로 하면 모션 컨트롤 IC 내부 회로의 테스트 명령이 기동하여 생각하지 않는 동작이 발생합니다.

4.4 WR1 mode register1

Mode register1은 4축 모두, 각각 개별적으로 가지고 있습니다. 어느 축의 Mode register를 사용 할 지는 축지정에 의해서 정해집니다. Mode register1은 Drive 도중에 감속정지/즉시정지 입력 신호 IN3~IN0의 유효/무효와 유효일 때의 논리 Level을 설정하는 Bit와 Interrupt의 허가/금지를 설정하는 Bit로 구성됩니다.

IN3~IN0를 유효로 하여 정량 Pulse drive, 또는 연속 Pulse drive로 Drive를 시작하여 지정된 IN신호가 설정해 둔 논리 Level이 되면 Drive는 감속정지 또는 즉시정지합니다. 가감속 Drive이면 감속정지, 정속 Drive이면 즉시정지합니다.

WR1	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	D-END	C-STA	C-END	P _≥ C+	P<C-	P<C-	P _≥ C-	PULSE	IN3-E	IN3-L	IN2-E	IN2-L	IN1-E	IN1-L	IN0-E	IN0-L

Interrupt의 허가/금지
드라이브 정지 입력 신호의 유효/무효, 논리

D7, 5, 3, 1 INm-E Drive 정지 입력 신호 INm(m:0~3)의 유효/무효를 설정하는 bit입니다.

0:무효, 1:유효

D6, 4, 2, 0 INm-L 입력 신호 INm의 유효 시, 논리 Level을 설정하는 bit입니다.

0:Low 에서 정지, 1:Hi에서 정지

자동원점복귀에서 사용하는 INm신호의 논리 Level을, 설정하는 bit입니다

유효/무효 bit(D5, D3, D1)는 무효로 합니다.

이하의 bit는 interrupt 허가/금지 Bit입니다. 1로 하면 interrupt 허가, 0으로 하면 interrupt 금지가 됩니다.

D8 PULSE Drive pulse마다 Pulse의 ↑로 Interrupt가 발생합니다.

(Drive pulse 정논리 설정 시)

D9 P_≥C- 논리/실제위치 Counter의 값이 COMP- register의 값보다 클 때, Interrupt가 발생합니다.

D10 P<C- 논리/실제위치 Counter의 값이 COMP- register의 값보다 작을 때, Interrupt가 발생합니다.

D11 P<C+ 논리/실제위치 Counter의 값이 COMP+ register의 값보다 작을 때, Interrupt가 발생합니다.

D12 P_≥C+ 논리/실제위치 Counter의 값이 COMP+ register의 값보다 클 때, Interrupt가 발생합니다.

D13 C-END 가감속 Drive시에 정속 영역에서 Pulse 출력을 종료했을 때, Interrupt가 발생합니다.

D14 C-STA 가감속 Drive시에, 정속 영역에서 Pulse 출력을 시작했을 때, Interrupt가 발생합니다.

D15 D-END Drive가 종료했을 때, Interrupt가 발생합니다.

Reset시에는 D15 ~ D0는 모두 0으로 설정 됩니다.

4.5 WR2 mode register 2

Mode register 2는 4축 모두, 각각 개별적으로 가지고 있습니다. 어느 축의 Mode register를 사용 할 지는 축 지정에 의해 정해집니다. Mode register 2는 Limit입력 신호의 Mode설정, Drive pulse의 Mode 설정, Encoder입력 신호의 Mode설정, 및 서보모터용 신호의 논리 Level, 유효/무효의 설정을 실시합니다.

WR2	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	INP-E	INP-L	ALM-E	ALM-L	PIND1	PIND0	PINMD	DIR-L	PLS-L	PLSMD	CMPSL	HLMT-	HLMT+	LMTMD	CLMT-	SLMT+

D0 SLMT+ COMP+ register를 +방향의 소프트웨어 limit로 설정합니다. 1로 하면 유효, 0으로 하면 무효가 됩니다. 유효하게 하면 +방향의 drive중, 논리/실제위치 Counter가 COMP+ register의 값 보다 크게되어 감속정지합니다. 또, RR2 register의 D0(SLMT+) Bit가 1이 됩니다.
이 상태에서 +방향의 Drive 명령은 실행되지 않습니다.

[주의] 확장 Mode의 위치 Counter 가변 링을 동작시키는 경우에는, 소프트웨어 Limit를 사용 할 수가 없습니다.

D1 SLMT- COMP- register를 -방향의 소프트웨어 Limit로 설정합니다. 1로 하면 유효, 0으로 하면 무효가 됩니다. 유효로 하면, -방향의 Drive 중에 논리/실제위치 Counter가 COMP- register의 값보다 작으면 감속정지합니다. 또, RR2 register의 D1(SLMT-) Bit가 1이 됩니다.
이 상태에서 -방향의 Drive 명령은 실행되지 않습니다.

D2 LMTMD 하드웨어 Limit(nLMTP, nLMTM 입력 신호)가 Active 되었을 때의 Drive 정지 방식을 설정합니다. 0으로 하면 즉시정지, 1로 하면 감속정지합니다.

D3 HLMT+ +방향 Limit 입력신호(nLMTP)의 논리 Level을 설정합니다.

0:Low active, 1:Hi active

D4 HLMT- -방향 Limit 입력신호(nLMTM)의 논리 Level을 설정합니다.

0:Low active, 1:Hi active

D5 CMPSL COMP+/- register의 비교 대상을 논리위치/실제위치 Counter로 설정합니다.

0:논리위치 Counter, 1:실제위치 Counter

D6 PLSMD Drive pulse의 출력 방식을 설정합니다.

0:독립 2pulse 방식, 1:1 pulse 방식

독립 2pulse 방식으로 하면 출력 신호 nP+P/nP+N에 +방향 Pulse가, 출력 신호 nP-P/nP-N에 -방향 pulse가 출력됩니다. 1pulse 방식으로 하면 출력 신호 nP+P/nP+N에 + / - 양방향의 Drive pulse가, 출력 신호 nP-P/nP-N에 Pulse의 방향 신호가 출력됩니다.

【주의】 1pulse 방식의 경우는 Pulse 신호와 방향 신호가 출력되는 타이밍 14.2, 14.3절을 참조하십시오.

D7 PLS-L Drive pulse의 논리 Level을 설정합니다. 0:정논리 Pulse, 1:부논리 Pulse

정논리 펄스 :  부논리 펄스 : 

D8 DIR-L Drive pulse의 방향 출력 신호의 논리 Level을 설정합니다.

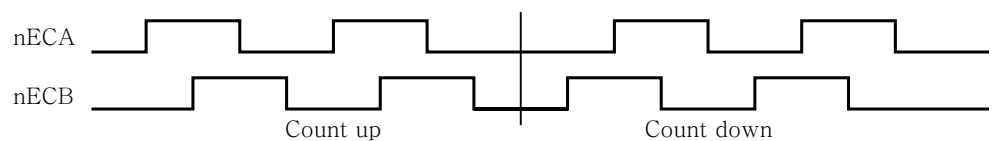
이 bit의 값에 의해, nP-P/nP-N 출력 신호의 전압 Level은 아래 표와 같이 출력됩니다.

D8 (DIR-L)	+ 방향 Pulse 출력 시	- 방향 Pulse 출력 시
0	Low	Hi
1	Hi	Low

D9 PINMD Encoder 입력신호(nECAP/nECAN, nECBP/nECBN)를 2상 Pulse 입력 또는 Up/Down pulse 입력으로 선택합니다. Encoder 입력 신호는 실제위치 Counter를 Count up/down 합니다.

0:2상 Pulse 입력, 1:Up/down pulse 입력

이 Bit를 2상 Pulse 입력 Mode로 설정하면, 정논리 Pulse로 A상이 진행되고 있을 때는 Count up, B상이 진행되고 있을 때는 Count down 합니다. 양신호의 ↑, ↓에서 Count up, Count down 합니다.



이 Bit를 Up/down pulse 입력의 Mode로 설정하면, nECAP/nECAN 가 Count up 입력에, nECBP/nECBN 가 Count down 입력이 됩니다. 각각, 정 Pulse의 ↑에서 Count 합니다.

D11, 10 PIND1, 0 Encoder 2상 Pulse 입력의 분주비를 설정합니다.

D11	D10	2상 Pulse 입력의 분주비
0	0	1/1
0	1	1/2
1	0	1/4
1	1	무효

D12 ALM-L nALARM 입력 신호의 논리 Level을 설정합니다.

0:Low active, 1:Hi Active

D13 ALM-E 서보모터 Alarm용 입력 신호 nALARM의 유효/무효를 설정합니다.

0:무효, 1:유효

유효로 설정하면 nALARM 입력 신호가 Active 상태로 되면 RR2 register의 D14 (ALARM) bit가 1이 됩니다. Drive중에 Active level이 되면, Drive는 즉시정지합니다.

D14 INP-L nINPOS 입력 신호의 논리 Level을 설정합니다.

0:Low active, 1:Hi active

D15 INP-E 서보모터 위치 결정 완료용 입력 신호 nINPOS의 유효/무효를 설정합니다.

0:무효, 1:유효

유효로 설정하면 Drive 종료 후, nINPOS 신호가 Active되면 RRO(주status) register의 n-DRV bit가 0으로 돌아옵니다.

Reset시에는, D15~D0는 모두 0으로 설정 됩니다.

4.6 WR3 mode register3

Mode register3은 4축, 각각 개별적으로 가지고 있습니다. 어느 축의 Mode register를 사용할 지는 축 지정에 의해 정해집니다. Mode register3은 메뉴얼 감속, 감속도 개별, S자 가감속 Mode, 외부 조작 Mode의 설정과 범용 출력 OUT7~4를 설정 합니다.

WR3	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	0	OUT7	OUT6	OUT5	OUT4	OUTSL	0	0	EXOP1	EXOP0	SACC	DSNDE	MANLD

D0 MANLD 가감속 정량 Pulse drive에 있어서 감속을 자동 감속/메뉴얼 감속으로 설정합니다.

0:자동 감속, 1:메뉴얼 감속

메뉴얼 감속 Mode로 설정할 경우는 메뉴얼 감속점이 설정되어 있어야 합니다.

D1 DSNDE 직선 가감속 Drive 감속시의 감속도를 가속도의 값을 사용할 지, 개별적인 감속도의 값으로 사용할 지를 설정합니다. 또, S자 가감속 Drive의 감속시의 감속도 증가율을 가속도 증가율의 값을 사용할 지, 개별의 감속도 증가율 값으로 사용할 지를 설정합니다.

D1 (DSNDE) 직선(사다리꼴)가감 속시의 감속도 S자 가감속시의감속도 증가율 가감속 커브의 형상

D1 (DSNDE)	직선(사다리꼴)가감 속시의 감속도	S자 가감속시의 감속도 증가율	가감속 커브의 형상
0	가속도(A)의 값을 사용	가속도 증가율(K)의 값을 사용	대칭
1	감속도(D)의 값을 사용	감속도 증가율(L)의 값을 사용	비대칭

가속과 감속이 대칭형 가감속 Drive로 실시할 때에는 이 Bit를 0으로, 비대칭형 가감속 Drive를 실시할 때에는 1로 합니다.

비대칭의 S자 가감속 • 정량 Pulse drive에서는 자동 감속할 수 없기 때문에, D0(MANLD) bit를 1로 하여 메뉴얼 감속점(DP)을 설정해야 합니다.

D2 SACC 직선(사다리꼴) 가감속/S자 가감속을 설정을 합니다.

0:직선(사다리꼴) 가감속, 1:S자 가감속

S자 가감속의 경우는 가속도 증가율(K), (감속도 증가율(L))이 설정되어 있어야 합니다.

D4, 3 EXOP1, 0 외부입력신호(nEXP+, nEXP-)에 의한 Drive 조작을 설정합니다.

D4(EXOP1)	D3(EXOP0)	내 용
0	0	외부 입력 신호에 의한 Drive 조작 무효
0	1	연속 Pulse drive mode
1	0	정량 Pulse drive mode
1	1	2상 Encoder 신호 mode

연속 Pulse drive mode에서는 nEXP+ 신호의 Low level 기간 동안 +방향의 Drive pulse를 연속적으로 출력합니다. nEXP- 신호의 경우도 이와 같이 -방향의 Drive pulse를 연속해서 출력합니다. 정량 Pulse drive mode에서는 nEXP+ 신호를 Hi level에서 Low level로 변경될 때, ↓에서 +방향의 정량 pulse drive가 기동합니다.

nEXP- 신호의 경우도 -방향의 정량 Pulse drive를 기동합니다.

2상 Encoder 신호 Mode에서는 nEXP- 신호가 Low level로 nEXP+ 신호의 ↑에서 +방향의 정량 Pulse drive가 기동합니다. 또, nEXP- 신호가 Low level로 nEXP+ 신호의 ↓에서 -방향의 정량 Pulse drive가 기동합니다.

D7 OUTSL 출력 신호 nOUT7 ~ 4를 범용 출력 / Drive 상태를 출력할 지를 선택 합니다.
0:범용 출력으로 사용합니다. D11~D8의 내용이 nOUT7~4 단자로 출력됩니다.
1:nOUT7~4는 아래 표와 같이 Drive 상태를 출력합니다.

신호명	출력 내용
nOUT4/CMPP	논리/실제위치 counter가 COMP+ register보다 클 때 Hi level로, 작을 때 Low level이 됩니다.
nOUT5/CMPM	논리/실제위치 counter가 COMP- register보다 작을 때 Hi level로, 클 때 Low level이 됩니다.
nOUT6/ASND	Drive 명령 실행 중, 가속 상태가 되면 Hi level이 됩니다.
nOUT7/DSND	Drive 명령 실행 중, 감속 상태가 되면 Hi level이 됩니다.

D11~8 OUTm 출력 신호 nOUT7~4를 범용 출력으로 설정합니다.
0:Low level 출력. 1:Hi level 출력.
Reset시, D15~D0는 모두 0으로 설정됩니다. D15~12, 6, 5 bit는 항상 0으로 설정합니다.

4.7 WR5 보간 Mode register

보간 Drive를 실행하기 위한 축 지정, 선속일정 Mode, 보간 Step 출력 Mode, 보간 시 Interrupt 설정을 실시합니다.

WR5	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	BPINT	CIINT	0	CMPLS	EXPLS	0	SPD1	SPD0	0	0	AX31	AX30	AX21	AX20	AX11	AX10
	Interrupt 허가			펄스 출력			선속 일정				제 3축		제 2축		제 1축(주축)	

D1, 0 AX11, 10 보간 Drive를 실시하는 제1축(주축)을 지정합니다. 축코드를 아래 표에 나타냅니다.

축	코드(2진)
X	0 0
Y	0 1
Z	1 0
U	1 1

제 1축(주축)으로 지정된 축은, 보간연산을 기동하는 기본 Pulse를 발생하기 때문에, 정속/가감속 Drive에 필요한 속도 Parameter를 설정해야 합니다.

D3, 2 AX21, 20 보간 Drive를 실시하는 제 2축을 지정합니다.
D5, 4 AX31, 30 3축 보간 Drive를 실시하는 제 3축을 지정합니다.
2축 보간 Drive시에는 임의의 값을 지정합니다.

D9, 8 LSPD1, 0 보간 Drive의 선속일정 Mode를 설정합니다.

D9	D8	동작 Mode
0	0	선속일정 무효
0	1	2축 선속일정
1	0	(설정 불가)
1	1	3축 선속일정

2축 선속일정 Mode의 경우는, 제 2축의 Range(R)를 주축 R의 1.414배의 값으로 설정합니다. 3축 선속일정 Mode의 경우는, 제 2축의 Range(R)를 주축 R의 1.414 배의 값으로, 제 3축의 Range(R)를 주축 R의 1.732 배의 값으로 설정합니다.

D11 EXPLS 0으로 설정합니다. (사용 안함)

D12 CMPLS 1로 하면 보간 Drive를 Commend로 step출력 하는 Mode가 됩니다.

D14 CHINT 연속보간 시, Interrupt 발생의 허가/금지를 설정합니다. 0:금지, 1:허가

D15 BPINT Bit pattern 보간 시, Interrupt 발생의 허가/금지를 설정합니다. 0:금지, 1:허가
Reset시, D15~D0는, 모두 0으로 설정합니다.

4.8 WR6, 7 Write data register1, 2

Data 기록명령을 설정하는 Register입니다. WR6 register에는 Write data 하위 16 bit(WD15~WD0), WR7 register에는 Write data 상위 16 bit(WD31~WD16)를 설정합니다.

WR6	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	WD15	WD14	WD13	WD12	WD11	WD10	WD9	WD8	WD7	WD6	WD5	WD4	WD3	WD2	WD1	WD0
WR7	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	WD31	WD30	WD29	WD28	WD27	WD26	WD25	WD24	WD23	WD22	WD21	WD20	WD19	WD18	WD17	WD16

Data 기록 명령은 우선, 각각의 명령으로 지정되어 있는 Data 길이를 이러한 Write data register에 기입합니다. Write data register WR6, 7(8 bit data bus의 경우는 WR6L, WR6H, WR7L, WR7H)을 Write한 후, Commend register에 명령 코드를 기입하면 Write data register의 내용이 각각의 내부 Register에 들어갑니다. 기입해지는 수치 Data는 모두 바이너리(Binary, 2진수)입니다. 또, 부의 값은 2의 보수로 취급합니다. 각 명령의 Data는, 반드시 지정되는 Data 길이로 설정해야 합니다. Reset시에는 WR6, WR7 register의 내용은 부정입니다.

4.9 RR0 주status register

각 축의 Drive, 에러 상태를 표시합니다. 또한, 보간 Drive, 연속 보간시 다음 Data 가능, 원호보간의 상한, BP보간의 Stack counter를 표시합니다.

WR0	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	-	BPSC1	BPSC0	ZONE2	ZONE1	ZONE0	CNEXT	I-DRV	U-ERR	Z-ERR	Y-ERR	X-ERR	U-DRV	Z-DRV	Y-DRV	X-DRV
									각 축의 에러 표시				각 축의 드라이브 표시			

D3~0 n-DRV 각 축의 Drive 상태를 나타냅니다. 이 Bit가 1이 되면 그 축이 Drive pulse를 출력 중인 것을 나타내고 있습니다. 0이면 그 축이 Drive를 종료하고 있는 것을 나타냅니다. 자동 원점 복귀 실행시에는 실행하고 있는 동안 이 Bit가 1이 됩니다.

서보모터 위치 결정 완료용 입력 신호의 nINPOS를 유효로 설정하면 Drive pulse를 출력 후, nINPOS 신호가 Active되고 나서 0으로 돌아옵니다.

D7~4 n-ERR 각 축의 에러 발생 상태를 표시합니다. 각 축의 RR2 register의 에러 Bit(D7~D0), 및 RR1 register의 에러 종료 bit(D15~D12) 중, 하나라도 1이되면, 이 bit가 1이 됩니다.

D8 I-DRV 보간 Drive 상태를 나타냅니다. 이 bit가 1이 되면, 보간 Drive pulse를 출력중인 것을 나타내고 있습니다.

D9 CNEXT 연속보간시 다음 Data 기입 가능성을 나타냅니다. 연속보간 Drive로, 이 Bit가 1이 되면, 다음 세그먼트(Segment)를 위한 Parameter data, 및 보간 명령을 기입하는 것이 가능하게 됩니다.

D3~0	n-DRV	각 축의 Drive 상태를 나타냅니다. 이 Bit가 1이 되면 그 축이 Drive pulse를 출력 중인 것을 나타내고 있습니다. 0이면 그 축이 Drive를 종료하고 있는 것을 나타냅니다. 자동 원점 복귀 실행시에는 실행하고 있는 동안 이 Bit가 1이 됩니다. 서보모터 위치 결정 완료용 입력 신호의 nINPOS를 유효로 설정하면 Drive pulse를 출력 후, nINPOS 신호가 Active되고 나서 0으로 돌아옵니다.
D7~4	n-ERR	각 축의 에러 발생 상태를 표시합니다. 각 축의 RR2 register의 에러 Bit(D7~D0), 및 RR1 register의 에러 종료 bit(D15~D12) 중, 하나라도 1이되면, 이 bit가 1이 됩니다.
D8	I-DRV	보간 Drive 상태를 나타냅니다. 이 bit가 1이 되면, 보간 Drive pulse를 출력중인 것을 나타내고 있습니다.
D9	CNEXT	연속보간시 다음 Data 기입 가능을 나타냅니다. 연속보간 Drive로, 이 Bit가 1이 되면, 다음 세그먼트(Segment)를 위한 Parameter data, 및 보간 명령을 기입하는 것이 가능하게 됩니다.
D1~10	ZONEm	원호보간 Drive에 대해서 현재 Drive 중의 상한을 나타냅니다.

D12	D11	D10	현재 Drive 중의 상한
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

D14, 13 BPSC1, 0 Bit pattern 보간 Drive에 대해서, Stack Counter(SC)의 값을 나타냅니다.

D14	D13	Stack Counter(SC)의 값
0	0	0
0	1	1
1	0	2
1	1	3

Bit pattern보간의 Drive중, SC=3일 때는, Bit data stack이 가득 차 있다는 것을 나타내고 있습니다. SC=2일 때는 각 축당 16 bit가 비어있으며, SC=1 일 때는, 각 축당 16 bit×2가 비어 있습니다. SC=0 는 Bit data를 모두 출력하여 Drive가 종료한 것을 나타냅니다.

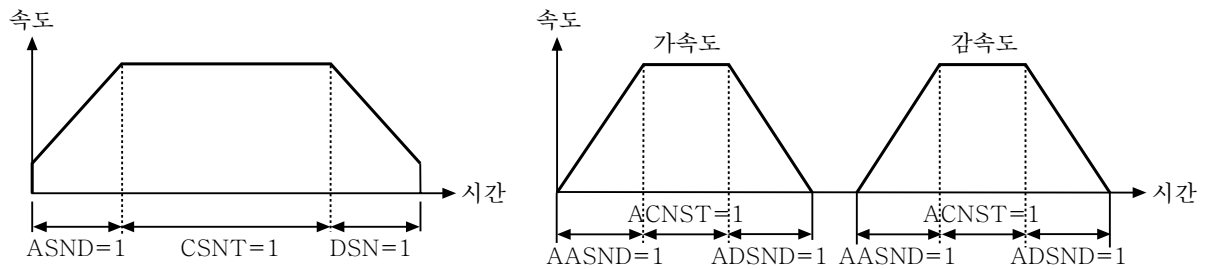
4.10 RR1 status register 1

Status register1은 4축, 각각 개별적으로 가지고 있습니다. 어느 축의 Status register를 읽을 것인가는 축 지정에 의해 정해집니다. Status register1은, 논리/실제위치 Counter와 COMP± register의 대소 비교, 가감속 Drive의 가속 상태, S자 가감의 가가속 상태를 표시합니다. 또, Drive 종료 Status를 표시합니다.

WR1	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	EMG	ALARM	LMT-	LMT+	IN3	IN2	IN1	IN0	ADSND	ACNST	AASND	DSND	CMST	ASND	CMP-	CMP+

————— 드라이브 종료 상황 —————

D0	CMP+	논리/실제위치 Counter와 COMP+ register의 대소 관계를 나타냅니다. 1:논리/실제위치 Counter ≥ COMP+ register 0:논리/실제위치 Counter < COMP+ register
D1	CMP-	논리/실제위치 Counter와 COMP- register의 대소 관계를 나타냅니다. 1:논리/실제위치 Counter < COMP- register 0:논리/실제 위치 Counter ≥ COMP- register



D2	ASND	가감속 Drive 시, 가속 때, 1이 됩니다.
D3	CNST	가감속 Drive 시, 정속 때, 1이 됩니다.
D4	DSND	가감속 Drive 시, 감속 때, 1이 됩니다.
D5	AASND	S자 가감속 Drive로, 가속도/감속도가 증가할 때, 1이 됩니다.
D6	ACNST	S자 가감속 Drive로, 가속도/감속도가 일정할 때, 1이 됩니다.
D7	ADSND	S자 가감속 Drive로, 가속도/감속도가 감소할 때, 1이 됩니다.
D11~8	IN3~0	Drive가, 외부 감속 정지 신호(nIN3~0)에 의해 정지했을 때, 1이 됩니다.
D12	LMT+	Drive가, +방향 Limit 신호(nLMT+)에 의해 정지했을 때, 1이 됩니다.
D13	LMT-	Drive가, -방향 Limit 신호(nLMT-)에 의해 정지했을 때, 1이 됩니다.
D14	ALARM	Drive가, 서보모터용 Alarm 신호(nALARM)에 의해 정지했을 때, 1이 됩니다.
D15	EMG	Drive가, 긴급정지 신호(EMG)에 의해 정지했을 때, 1이 됩니다.

◎Drive 종료 Status bit에 대해

Drive 종료 Status bit는, Drive를 종료시킨 요인의 정보를 보관, 유지하는 bit입니다. 정량 Pulse drive, 연속 Pulse drive는 다음과 같은 요인에 의해 종료합니다.

- ①정량 Pulse drive로, 출력 Pulse를 모두 출력 했을 때.
- ②감속정지, 또는 즉시정지 명령이 기입되었을 때.
- ③소프트웨어 Limit가 유효로 설정되어 Active 되었을 때.
- ④정량/연속 Pulse drive로, 감속 정지시키는 외부 신호(nIN3, 2,1,0)가 Active 되었을 때.
- ⑤Limit입력 신호(nLMT+P, nLMT-)가 Active 되었을 때.
- ⑥nALARM신호가 유효로 설정되어 Active 되었을 때.
- ⑦EMG신호가 Low level이 되었을 때.

여기서 ①, ②의 요인에 대해서는 상위 CPU에 의해 제어 가능한 요소이며, ③의 요인은 Drive종료 후에도 RR2 register로 확인할 수가 있습니다. 그러나, ④~⑦ 요인은 Drive를 종료시킨 원인이 되었어도 Drive가 정지할 때까지 반드시 Active상태가 되어 있다고는 할 수 없습니다. Drive종료 Status bit는 ④~⑦ 요인에 대해서, Drive를 종료시킨 요인의 Bit가 1이 된 후, 신호가 Non active 되어도 Bit 정보를 유지합니다.

Drive종료status bit 가운데, 에러 요인이 되는 D15~D12의 Bit가 1이 되면, RR0 주status register의 n-ERR bit가 1이 됩니다. Drive종료status bit는, 다음 Drive명령의 기입으로 자동적으로 Clear되지만, 종료statusclear 명령(25h)에 의해 Clear할 수가 있습니다.

4.11 RR2 status register2

Status register2는 4축, 각각 개별적으로 가지고 있습니다. 어느 축의 Status register를 읽을 것인가는 축 지정에 의해 정해집니다. Status register2는 에러 정보, 및 자동원점복귀 실행시의 실행 status를 표시하는 Register입니다. 에러정보(D7~D0)의 각 bit가 1이 되면, 그 Bit에 해당하는 에러가 발생한 것을 나타냅니다. 이 RR2 register의 D7~D0의 몇개의 bit가 1이 되면, RR0주status register의 n-ERR bit가 1이 됩니다.

WR2	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	-	0	0	HMST4	HMST3	HMST2	HMST1	HMST0	HOME	0	EMG	ALARM	HLMT-	HLMT+	SLMT-	SLMT+
자동 원점 출력 실행 스테이트									에러 정보							

D0	SLMT+	COMP+ register를 소프트웨어 Limit로서 유효하게 하고, +방향 Drive시에, 논리/실제위치 count가 COMP+ register의 값보다 클 때.
D1	SLMT-	COMP- register를 소프트웨어 Limit로서 유효하게 하고, -방향 Drive시에, 논리/실제위치 count가 COMP- register의 값보다 작을 때.
D2	HLMT+	+방향 Limit신호(nLMT+)가 Active level일 때.
D3	HLMT-	-방향 Limit신호(nLMT-)가 Active level일 때.

- D4 ALARM 서보모터용 Alarm신호(nALARM)가 유효로 설정되고, Active level일 때.
- D5 EMG 긴급정지 신호(EMG)가 Low level일 때.
- D7 HOME 자동원점복귀 실행시의 에러입니다. Step 3시작 시, Encoder Z상신호(nIN2)가 Active되어 있으면 1이 됩니다.
- D12~8 HMST4~0 자동원점복귀 실행 Status는, 자동원점복귀 실행중에 현재 실행하고 있는 동작 내용을 나타냅니다. 2.5.4절 참조
- Drive중에 진행 방향의 Hard/Soft limit가 작동하면, Drive는 감속정지 또는 즉시정지합니다.
- 정지 후, 정지 전과 동일한 방향으로의 Drive 명령은 실행되지 않습니다.
- SLMT+/- bit는 역방향 Drive시, 조건이 되어도 1이 되지 않습니다.

4.12 RR3 status register3

Status register3은 4축, 각각 개별적으로 가지고 있습니다. 어느 축의 Status register를 읽을 것인지는 축 지정에 의해 정해집니다. Status register3은 Interrupt 발생 요인을 나타내는 Register입니다. Interrupt가 발생하면 그 Interrupt 발생 요인의 Bit가 1이 됩니다.

D0에서 D7의 Interrupt를 발생시키려면 WR1 register의 각 발생 요인을 Interrupt 허가로 설정해야 합니다.

WR3	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
							SYNC	HMEND	D-END	C-STA	C-END	P _≥ C+	P<C+	P _≥ C-	P<C-	PLUSE
Interrupt 발생 요인																

- D0 PULSE Drive pulse가 발생. (Drive pulse 정논리 설정 시)
- D1 P_≥C- 논리/실제위치 Counter가 COMP- register보다 크다.
- D2 P<C- 논리/실제위치 Counter가 COMP- register보다 작다.
- D3 P<C+ 논리/실제위치 Counter가 COMP+ register보다 작다.
- D4 P_≥C+ 논리/실제위치 Counter가 COMP+ register보다 크다.
- D5 C-END 가감속 Drive시, 정속 영역 Pulse 출력을 종료.
- D6 C-STA 가감속 Drive시, 정속 영역 Pulse 출력을 시작.
- D7 D-END Drive가 종료.
- D8 HMEND 자동원점복귀가 종료. (2.5절 참조)
- D9 SYNC 동기 동작이 기동. (2.6절 참조)

어떤 Interrupt 요인의 Interrupt가 발생하면, 이 Register의 bit가 1이 되어, Interrupt 출력 신호가 Low level이 됩니다. CPU가 Interrupt를 발생시킨 축의 RR3 register를 읽으면, RR3 register의 bit는 0으로 복귀되어 Interrupt 출력신호는 Non-active Level로 돌아옵니다.

【주의】 8 bit data bus의 경우는, RR3L register의 읽기로 모두 Clear되기 때문에, D8(HMEND), D9(SYNC)bit를 사용하는 경우에는, 반드시 RR3H를 먼저 읽어내고 나서, RR3L register를 읽어냅니다.

4.13 RR4, 5 Input register1, 2

Input register1, 2는 각 축의 입력신호 상태를 직접 표시합니다. 입력신호가 Low level일 때는 0, Hi level때는 1을 나타냅니다. 이러한 입력신호의 상태를 나타내지 않을 때에는, 범용 입력신호로 사용할 수 있습니다.

WR4	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	Y-ALM	Y-INP	Y-EX-	Y-EX+	Y-IN3	Y-IN2	Y-IN1	Y-IN0	X-ALM	X-INP	X-EX-	X-EX+	X-IN3	X-IN2	X-IN1	X-IN0

WR5	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	U-ALM	U-INP	U-EX-	U-EX+	U-IN3	U-IN2	U-IN1	U-IN0	Z-ALM	Z-INP	Z-EX-	Z-EX+	Z-IN3	Z-IN2	Z-IN1	Z-IN0

Bit 명	입력 신호	Bit 명	입력 신호
n-IN0	n-IN0+/n-IN0-	n-EX+	nEXP
n-IN1	n-IN1	n-EX-	nEXP-
n-IN2	n-IN2	n-INP	nINPOS
n-IN3	n-IN3	n-ALM	nALARM

4.14 RR6, 7 read data register1, 2

Data 읽기 명령에 의해, 내부 Register의 Data가 read data register에 설정 됩니다. RR6 register에는 Read data 하위 16 bit(RD15~RD0)가, RR7 register에는 Read data 상위 16 bit(RD31~RD16)가 설정 됩니다.

WR4	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	Y-ALM	Y-INP	Y-EX-	Y-EX+	Y-IN3	Y-IN2	Y-IN1	Y-IN0	X-ALM	X-INP	X-EX-	X-EX+	X-IN3	X-IN2	X-IN1	X-IN0

WR5	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	U-ALM	U-INP	U-EX-	U-EX+	U-IN3	U-IN2	U-IN1	U-IN0	Z-ALM	Z-INP	Z-EX-	Z-EX+	Z-IN3	Z-IN2	Z-IN1	Z-IN0

Data는 모두 2진수(Binary)입니다. 또, 부의 값은 2의 보수로 취급합니다.

5. 명령 일람

◎Data 기입 명령

코 드	명 령	Parameter 기호	Data 범위	Data 길이
00h	Range 설정	R	8,000,000(배율:1)~16,000(배율:500)	4 Byte
01	가속도 증가율 설정	K	1 ~ 65,535	2
02	가속도 설정	A	1 ~ 8,000	2
03	감속도 설정	D	1 ~ 8,000	2
04	기동속도 설정	SV	1 ~ 8,000	2
05	Drive 속도 설정	V	1 ~ 8,000	2
06	출력 Pulse 수/보간 종점 설정	P	출력 Pulse 수 : 0 ~ 4,294,967,295 보간 종점 : -2,147,483,646 ~ +2, 147,483,646	4
07	메뉴얼 감속점 설정	DP	0 ~ 4,294,967,295	4
08	원호 중심점 설정	C	-2,147,483,646 ~ +2,147,483,646	4
09	논리위치 Counter 설정	LP	-2,147,483,648 ~ +2,147,483,647	4
0A	실제위치 Counter 설정	EP	-2,147,483,648 ~ +2,147,483,647	4
0B	COMP+ register 설정	CP	-2,147,483,648 ~ +2,147,483,647	4
0C	COMP- register 설정	CM	-2,147,483,648 ~ +2,147,483,647	4
0D	가속 Counter offset 설정	AO	-32,768 ~ +32,767	2
0E	감속도 증가율 설정	L	1 ~ 65,535	2
60	확장 Mode 설정	EM	(Bit data)	4
61	원점 검출 속도 설정	HV	1 ~ 8,000	2
64	동기 동작 Mode 설정	SM	(Bit data)	4

【주의】 Data를 설정할 때는 반드시 지정된 Data 길이로 설정 합니다.

【Parameter 계산식】

$$\text{배율} = \frac{8,000,000}{R}$$

$$\text{가감속 증가율(pps/sec}^2\text{)} = \frac{6.25 \times 10^6}{K} \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

$$\text{감속도 증가율(pps/sec}^2\text{)} = \frac{6.25 \times 10^6}{L} \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

$$\text{가감속(pps/sec)} = A \times 125 \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

$$\text{감속도(pps/sec)} = D \times 125 \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

$$\text{드라이브 속도(pps)} = V \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

$$\text{기동속도(pps)} = SV \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

◎Data 읽기 명령

코 드	명 령	Parameter 기호	Data 범위	Data 길이
10h	논리위치 Counter 읽기	LP	-2,147,483,648 ~ +2,147,483,647	4 Byte
11	실제위치 Counter 읽기	EP	-2,147,483,648 ~ +2,147,483,647	4
12	현재 Drive 속도 읽기	CV	1 ~ 8,000	2
13	현재 가/감속도 읽기	CA	1 ~ 8,000	2
14	동기 버퍼 Register 읽기	SB	-2,147,483,648 ~ +2,147,483,647	4

◎Drive 명령

코 드	명 령
20h	+방향 정량 Pulse drive
21	-방향 정량 Pulse drive
22	+방향 연속 Pulse drive
23	-방향 연속 Pulse drive
24	Drive 시작 Hold
25	Drive 시작 Free/종료 Status clear
26	Drive 감속 정지
27	Drive 즉시 정지

◎보간 명령

코 드	명 령
30h	2축 직선보간 Drive
31	3축 직선보간 Drive
32	CW 원호보간 Drive
33	CCW 원호보간 Drive
34	2축 Bit pattern보간 Drive
35	3축 Bit pattern보간 Drive
36	BP register 기입 가능 (*1)
37	BP register 기입 불가
38	BP data stack
39	BP data clear
3A	보간 Single step
3B	감속 유효
3C	감속 무효
3D	보간 Interrupt clear

※(*1) BP : "Bit Pattern"의 약어

◎그 외의 명령

코 드	명 령
62	자동원점복귀 실행
65	동기 동작 기동
OF	NOP(축 전환용)

【주의】 이외의 명령 코드를 Commend register에 기입하면 모션 컨트롤 IC 내부 회로의 테스트 명령이 기동하여, 생각하지 않는 동작을 하는 경우가 있습니다.

6. Data 기록(Write) 명령

Data기록 명령은, 기록 Data를 수반하는 명령입니다. Drive를 위한 가속도, Drive 속도, 출력 Pulse수 등의 동작 Parameter를 설정합니다. 복수의 축을 지정 하면, 해당 Data를 지정된 모든 축에 설정할 수 있습니다.

Data 기록 명령은 지정된 Data 길이가 2바이트 이면 WR6 register에, Data 길이가 4바이트 이면 WR6, 7 register에 설정 됩니다. 그리고, WR0 register에 축을 지정하고 명령 코드를 기입하면 실행됩니다. WR6, 7 write data register에 설정하는 수치 Data는 모두 2진수(Binary)입니다. 또, 부의 값은 2의 보수로 취급 합니다. 각각의 Data는 반드시, Data 범위 내의 값을 설정해야 합니다. 범위의 값을 설정하면 올바른 Drive 동작이 실행되지 않습니다.

【주의사항】

- ①Data 기입 명령의 명령 처리에 필요로 하는 시간은, 최대로 250 nsec(CLK=16 MHz의 경우)입니다.
명령을 기입 후 명령을 처리하는 시간에는 다음 data 명령을 기입하지 마십시오.
- ②가속 Counter offset(AO)를 제외한, 모든 동작 Parameter는 Reset시 후에는 적용되지 않습니다.
Drive에 필요한 Parameter에 대해서는 Drive 전에 반드시 적절한 값을 설정 합니다.

6.1 Range 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
00h	Range 설정	R	8,000,000(배율:1) ~ 16,000(배율:500)	4 Byte

Range는 기동속도, Drive 속도, 가속도, 감속도, 가속도 증가율, 감속도 증가율의 배율을 결정하는 Parameter 입니다. Range 설정치를 R로 하면, 배율은 다음식과 같습니다.

$$\text{배율} = \frac{8,000,000}{R}$$

Drive속도, 기동속도, 가감속도 등의 Parameter는, 설정 범위가 1~ 8000 이므로, 이 보다 크게 설정 하는 경우에는, 배율을 올리지 않으면 안됩니다. 배율을 크게 하면, 고속으로 Drive할 수가 있습니다만, 속도 분해능은 엉성해집니다. 사용하시는 속도 범위를 커버할 수 있는 최소의 값으로 설정합니다. 예를 들면, 40kpps 속도까지 사용한다면, 속도 설정 범위가 1~8000, 배율은 5배 이므로 R을 1,600,000으로 설정합니다.

Range(R)를 Drive 중에 변경하면 속도가 불연속적으로 변화합니다.

6.2 가속도 증가율 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
01h	가속도 증가율 설정	K	1 ~ 65,535	2 Byte

가속도 증가율 설정치는, S자 가감속에 있어서 가속도의 단위시간 당 증가/감소율을 결정하는 Parameter입니다. 가속과 감속이 대칭인 대칭형 S자 가감속 Drive(WR3/D1=0)에서는, 감속시에도 이 가속도 증가율의 값이 사용됩니다. 가속도 증가율의 설정치를 K로 하면, 가속도 증가율은 다음식과 같이 됩니다.

$$\text{가감속 증가율(pps/sec}^2\text{)} = \frac{6.25 \times 10^6}{K} \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

가속도 증가율 설정치(K)의 설정 범위가 1~ 65,535 이기 때문에 가속도 증가율 범위는 다음과 같이 됩니다.

배율=1 일 때, 954pps/sec² ~ 62.5 × 10⁶ pps/sec²

배율= 500 일 때, 477×10³ pps/sec² ~ 31.25×10⁹ pps/sec²

6.3 가속도 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
02h	가속도 설정	A	1 ~ 8,000	2 Byte

직선 가감속 Drive(사다리꼴)의 가속시, 가속도를 결정하는 Parameter입니다. 가속과 감속이 대칭형 직선 가감속 Drive(WR3/D1=0)에서는, 감속시에도 이 가속도의 값이 사용됩니다. S자 가감속 Drive에서는, 이 Parameter는 항상 최대치 8,000을 설정 합니다. 가속도 설정치를 A로 하면, 가속도는 다음 식과 같이 됩니다.

$$\text{가속도(pps/sec)} = A \times 125 \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

가속도 설정치(A)의 설정 범위가 1~ 8,000 이기 때문에 실제의 가속도 범위는 다음과 같이 됩니다.

배율=1 일 때, 125pps/sec² ~ 1 × 10⁶ pps/sec²

배율= 500 일 때, 62.5×10³ pps/sec² ~ 500×10⁶ pps/sec²

6.5 감속도 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
03h	감속도 설정	D	1 ~ 8,000	2 Byte

비대칭형 직선 가감속 Drive(WR3/D1=1)에서, 감속시의 감속도가 되는 Parameter 입니다.

비대칭형 S자 가감속 Drive에서, 이 Parameter는 항상 최대치 8,000으로 설정 합니다.

감속도 설정치를 D로 하면, 감속도는 다음 식과 같이 됩니다.

$$\text{감속도(pps/sec)} = D \times 125 \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

6.4 기동속도 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
04h	기동속도 설정	SV	1 ~ 8,000	2 Byte

가감속 Drive의 가속 시작시의 속도와 감속 종료시의 속도입니다.

기동속도 설정치를 SV로 하면, 기동속도는 다음 식과 같습니다.

$$\text{기동속도(pps)} = SV \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

Step 모터의 경우는, 자기동 주파수내의 값으로 설정합니다. 서보모터의 경우에도, 너무 낮게 값을 설정하면, 정량 Pulse drive의 감속 종료시에, 기동속도로 Drive가 출력이 유지될 수 있습니다. 이러한 경우는, 다음과 같이 실시합니다.

- 가속/감속 대칭형 직선 가감속 drive의 경우
 - 가속 Counter offset(AO)을 0으로 설정
 - 삼각 방지 기능 유효(확장 명령 60h WR6/D3(AVTRI)=1)
- 가속/감속 비대칭형 직선 가감속 drive의 경우
 - 가속 Counter offset(AO)을 0으로 설정
 - 삼각 방지 기능 유효(확장 명령 60 h WR6/D3(AVTRI)=1)

그러나, 가속도 > 감속도의 경우, 가속도 A와 감속도 D의 비율이 커지면 정속 영역이 없어집니다. 이러한 경우에는 기동속도를 올립니다.

6.6 Drive 속도 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
05h	Drive 속도 설정	V	1 ~ 8,000	2 Byte

가감속 Drive에서 정속 영역에 이르렀을 때의 속도입니다. 정속 Drive에서는 처음부터 이 속도가 됩니다.

Drive 속도 설정치를 V로 하면 Drive속도는 다음 식과 같이 됩니다.

이 Drive 속도를 기동속도 이하로 설정하면 가감속 Drive는 실행하지 않고 처음부터 정속 Drive가 됩니다.

Encoder의 Z상서치를 저속으로 Drive하여 즉시정지시키고 싶을 때는 Drive 속도를 기동속도 이하로 설정합니다.

Drive속도는 Drive 도중에도 자유롭게 변경할 수가 있습니다. 가감속 Drive의 정속 영역의 Drive 속도를 재설정하면, 가속 또는 감속을 시작하여 재설정된 속도에 이르면 정속 Drive합니다. 자동원점복귀에서의 Drive 속도는 Step 1, Step 4는 고속 검출속도로 고속이동 합니다.

【주의 사항】

- S자 가감속 정량 Pulse drive는, Drive 도중에 Drive 속도 변경을 할 수 없습니다. 또, S자 가감속 연속 Pulse drive에서도 가속중이나 감속중에 속도를 변경하면, 올바른 S자 커브를 그릴 수가 없습니다. 정속 영역에서 변경하십시오.
- 직선 가감속 정량 Pulse drive에서 Drive 도중에 빈번하게 Drive 속도를 변경하면, 출력 Pulse 종료시의 감속 영역에서 기동속도로 Drive시킬 수 있습니다.

6.7 출력 Pulse수/보간 중점 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
06h	출력 Pulse수/ 보간중점설정	P	출력 Pulse수 : 0 ~ 4,294,967,295 보간 중점 : -2,147,483,646~+2,147,483,646	4 Byte

출력 Pulse수는 정량 Pulse drive의 총 출력 Pulse수입니다. 부호 없이 32bit로 설정 합니다.

직선보간, 원호보간 Drive는 각 축의 중점을 설정합니다. 중점좌표는 32bit로 현재위치에 대한 상대치를 부호 첨부하여 설정 합니다. 출력 Pulse수는, Drive 도중에 변경할 수가 있습니다.

6.8 매뉴얼(Manual) 감속점 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
07h	매뉴얼 감속점 설정	DP	0 ~ 4,294,967,295	4 Byte

매뉴얼 감속 Mode의 가감속 정량 Pulse drive의 감속점을 설정합니다.

매뉴얼 감속 Mode는, WR3 register의 D0 bit를 1로 하고, 감속점은 다음과 같이 설정합니다.

매뉴얼 감속점 = 출력 Pulse수 - 감속시에 소비하는 Pulse수

6.9 원호 중심점 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
08h	원호 중심점 설정	C	-2,147,483,646 ~ +2,147,483,646	4 Byte

원호보간 Drive의 중심점을 설정합니다. 중심좌표는 현재위치에 대한 상대치를 부호 첨부하여 설정 합니다.

6.10 논리위치 Counter 설정

명령코드	명령	Parameter 기호	Data 범위	Data 길이
09h	논리위치 Counter 설정	LP	-2,147,483,648 ~ +2,147,483,647	4 Byte

논리위치 Counter의 값을 설정합니다. 논리위치 Counter는, +방향/-방향의 Drive 출력 Pulse를 Up/Down count 합니다. 논리위치 Counter의 값은 항상 기록 가능하며, Data 읽기 명령으로 항상 읽어낼 수도 있습니다.

6.11 실제위치 Counter 설정

명령코드	명령	Parameter 기호	Data 범위	Data 길이
0Ah	실제위치 Counter 설정	EP	-2,147,483,648 ~ +2,147,483,647	4 Byte

실제위치 Counter의 값을 설정합니다. 실제위치 Counter는, Encoder 입력 Pulse를 Up/Down count 합니다. 실제위치 Counter의 값은 항상 기록 가능하며, Data 읽기 명령으로 항상 읽어낼 수도 있습니다.

6.12 COMP + register 설정

명령코드	명령	Parameter 기호	Data 범위	Data 길이
0Bh	COMP + register 설정	CP	-2,147,483,648 ~ +2,147,483,647	4 Byte

COMP + register의 값을 설정합니다. COMP + register는, 논리/실제위치 Counter와 대소 비교하는 Register로 비교 결과는 RR1 register의 D0와 nOUT4/CMPP 신호로 출력됩니다. +방향의 소프트웨어 Limit로도 사용합니다. COMP + register의 값은 항상 기록 가능합니다.

6.13 COMP - register 설정

명령코드	명령	Parameter 기호	Data 범위	Data 길이
0Ch	COMP - register 설정	CM	-2,147,483,648 ~ +2,147,483,647	4 Byte

COMP - register의 값을 설정합니다. COMP - register는, 논리/실제위치 Counter와 대소 비교하는 Register로, 비교 결과는 RR1 register의 D1과 nOUT5/CMPP 신호로 출력됩니다. -방향의 소프트웨어 Limit로도 사용합니다. COMP - register의 값은 항상 기록 가능합니다.

6.14 가속 Counter offset 설정

명령코드	명령	Parameter 기호	Data 범위	Data 길이
0Dh	가속 Counter offset 설정	AO	0 ~ 65,535	2 Byte

가속 Counter의 Offset값을 설정합니다. 가속 Counter의 Offset값은, Reset시에 8이 Set됩니다. 기동속도를 낮게 설정해 가감속의 정량 Pulse drive를 실시하는 경우, 이 Parameter값은 0으로 설정해 주십시오.

6.15 감속도 증가율 설정

명령코드	명령	Parameter 기호	Data 범위	Data 길이
0Eh	감속도 증가율 설정	L	0 ~ 65,535	2 Byte

감속도 증가율 설정치는, 가속과 감속이 비대칭형 S자 가감속 Drive(WR3/D1 = 1)에 있어서 감속도의 단위 시간당의 증가/감소율을 결정하는 Parameter입니다. 가속과 감속이 대칭인 대칭형 S자 가감속 Drive에서는 사용되지 않습니다. 감속도 증가율의 설정치를 L로 하면, 감속도 증가율은 아래식과 같습니다.

$$\text{감속도증가율 (pps/sec}^2\text{)} = \frac{6.25 \times 10^6}{L} \times \underbrace{\frac{8,000,000}{R}}_{\text{배율}}$$

감속도 증가율 설정치(L)의 설정 범위가 1 ~ 65,535 이기 때문에, 감속도 증가율 범위는 다음과 같이 됩니다.

배율=1 일 때, 954pps/sec² ~ 62.5 × 10⁶ pps/sec²

배율= 500 일 때, 477 × 10³ pps/sec² ~ 31.25 × 10⁹ pps/sec²

6.16 확장 Mode 설정

명령코드	명령	Parameter 기호	Data 범위	Data 길이
60h	확장 Mode 설정	EM		4 Byte

확장 Mode의 설정은, 아래와 같이 WR6과 WR7 register의 각 Bit에 적정치를 설정한 후에, WR0 register에 축지정과 함께 명령 코드(60h)를 기입하면 WR6, 7 register의 내용이 확장 Mode register(EM6, 7)에 설정되고, Reset시에는 확장 Mode register(EM6, 7)의 모든 Bit는 Clear됩니다.

WR6	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	FL2	FL1	FL0	FE4	FE3	FE2	FE1	FE0	SMODE	0	HMINT	VIRING	AVTRI	POINV	EPINV	EPCLR

WR7	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	DCCW2	DCCW1	DCCW0	DCC-L	DCC-E	LIMIT	SAND	PCLR	ST4-D	ST4-E	ST3-D	ST3-E	ST2-D	ST2-E	ST2-D	ST2-E

편차 카운터 클리어 출력
Step 4
Step 3
Step 2
Step 1

- WR6/D0 EPCLR nIN2 신호에 의해 Drive가 정지했을 때 실제위치 Counter를 Clear 합니다.
 이 bit를 1로 하고 Drive 중에, nIN2 신호가 Active level로 변화했을 때, Drive가 정지됨과 함께 실제위치 Counter(EP)가 Clear 됩니다. WR1/D5(IN2-E) bit는 1로, WR1/D4(IN2-L) Bit는 유효 Level로 설정해야 합니다. (4.4 절 참조)
- WR6/D1 EPIINV 실제위치 Counter의 증감을 반전시킵니다.

WR6/D1 (EPIINV)	입력펄스 Mode	실제위치의 Counter(EP) 증감
0	A/B상 Mode	A상 진행 시, Count up. B상 진행 시, Count down.
	Up/down 펄스 mode	nECA+/- Pulse 입력 시, Count up. nECB+/- Pulse 입력 시, Count down.
1	A/B상 Mode	B상 진행 시, Count up. A상 진행 시, Count down.
	Up/down 펄스 mode	nECB+/- Pulse 입력 시, Count up. nECA+/- Pulse 입력 시, Count down.

- WR6/D2 POINV Drive pulse 출력, nP+P/nP+N(+방향의 Drive pulse)와 nP-P/nP-N(-방향의 Drive pulse)의 출력신호를 바꿉니다. 이 Bit를 1로 하면, +방향의 Drive에서는 nP-P/nP-N 신호의 Drive pulse가 출력되고, -방향의 Drive에서는 nP+P/nP+N 신호의 Drive pulse가 출력됩니다.
- WR6/D3 AVTRI 정량 Pulse drive의 직선 가감속(사다리꼴)에 있어서 삼각파형을 방지합니다.
 0 : 무효, 1 : 유효.
- WR6/D4 VRING 논리위치 Counter 및 실제위치 Counter의 가변 링 기능을 유효하게 합니다.
 0 : 무효, 1 : 유효
- WR6/D5 HMINT 자동원점복귀 종료 후, interrupt 신호를 발생시킵니다. 본 bit를 1하면, 자동원점출력 종료 후, Interrupt 신호가 Low active되어 Interrupt를 발생시킨 축의 RR3/D8 (HMEND) bit가 1을 나타냅니다. Interrupt를 발생시킨 축의 RR3 register를 읽고나면, RR3 register의 bit는 0으로 clear 되어 Interrupt 출력 신호는 Hi-Z으로 돌아옵니다.
- WR6/D7 SMODE S자 가감속 Drive 시, 지정된 Drive 속도에 도달하는 것을 우선적으로 설정시키고 싶을 때에 1으로 합니다.
- WR6/D12~8 FE4~0 입력 신호 시, 모션 콘트롤 IC 내부의 필터 기능을 유효/무효로 할지를 설정합니다.
 0 : 무효, 1 : 유효.

지정 Bit	필터 유효의 신호
WR6/D8 (FE0)	EMG(*1), nLMT+, nLMT-, nIN0, nIN1
WR6/D9 (FE1)	nIN2
WR6/D10(FE2)	n INPOS, nALARM
WR6/D11(FE3)	nEXP+, nEXP-
WR6/D12(FE4)	nIN3

(*1) : EMG신호는 X축의 WR6 register D8 bit로 설정합니다.

WR6/D15~13 FL2~0 필터 통과시간을 설정합니다. 입력 신호 필터 기능의 자세한 내용은 2.8절을 참조하십시오.

WR6/D15~13(FL2~0)	제거 가능한 최대 노이즈 폭	입력 신호 지연 시간
0	1.75 μ sec	2 μ sec
1	224 μ sec	256 μ sec
2	448 μ sec	512 μ sec
3	896 μ sec	1.024msec
4	1.792msec	2.048msec
5	3.584msec	4.096msec
6	7.168msec	8.192msec
7	14.336msec	16.384msec

WR7 register의 각 Bit는 자동원점복귀 Mode를 설정을 합니다.

각 Bit의 상세한 내용에 대해서는 2.5.2 절의 "◎ 자동원점 복귀 mode의 설정"을 참조하십시오.

【주의】 확장 Mode의 설정 명령은 WR6과 WR7 register의 내용이 모두 모션 컨트롤 IC 내부의 확장 Mode register(EM6, 7)에 설정 되기 때문에, 반드시 WR6 와 WR7 register의 양쪽 모두에 적정치를 설정해야 합니다.

6.17 원점 검출속도 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
61h	원점검출속도 설정	HV	1 ~ 8,000	2 Byte

자동원점 복귀의 Step 2, 3은 저속저속도를 설정합니다.

원점검출속도 설정치를 HV로 하면, 원점검출속도는 다음 식과 같이 됩니다.

$$\text{드라이브 속도(pps)} = \text{HV} \times \frac{8,000,000}{R}$$

배율

검출 신호가 Active 되었을 때 즉시정지시키기 위해서, 기동속도(SV)보다 낮은 값으로 설정합니다.

자동원점 복귀에 대해서는 2.5 절에 자세하게 기술되어 있습니다.

6.18 동기 동작 Mode 설정

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
64h	동기 동작 Mode 설정	SM		4 Byte

동기 동작 Mode의 설정은, 아래와 같이 WR6 과 WR7 register의 각 Bit에 적정치를 설정한 후에, WR0 register에 축 지정과 함께 명령 코드(64h)를 기입하면 WR6, 7 register의 내용이 모션 컨트롤 IC 내부의 동기 동작 Mode register(SM6, 7)에 설정 됩니다. Reset시에는 모션 컨트롤 IC 내부의 동기 동작 Mode register(SM6, 7)의 모든 bit는 0으로 Clear 됩니다.

WR6	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	AXIS3	AXIS2	AXIS1	0	0	0	CMD	LPRD	IIN3	IIN3	D-END	D-STA	P _≥ C-	P<C-	P<C+	P _≥ C+
	타축 시동								기동 요인(Provocative)							
WR7	H								L							
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
	INT	OUT	0	0	VLSET	OPSET	EPSET	LPSET	EPSAV	LPSAV	ISTOP	SSTOP	CDRV-	CDRV+	FDRV-	FDRV+
	동작(Action)															

각 bit의 상세한 설명은 2.6절을 참조하십시오.

7. Data 읽기(Read) 명령

Data 읽기 명령은 각 축의 Register 내용을 Read data register로 읽어내는 명령입니다.

WR0 register에 축 지정과 Data 읽기 명령 코드를 기입하면 지정된 Data가 RR6, 7 Register에 설정 됩니다.

CPU는 RR6, 7 register를 읽어내는 것에 의해 지정된 Data를 얻을 수 있습니다.

읽기 Data는 모두 2진수(Binary)입니다. 또, 부의 값은 2의 보수로 취급합니다.

【주의사항】

①Data 읽기 명령의 명령 처리에 필요로 하는 시간은 최대 250 nsec(CLK=16 MHz의 경우)입니다. 명령을 기입하고 나서, 이 시간 이후에 RR6, 7 register를 읽어야 합니다.

②축 지정은 반드시 1축만 지정합니다. 2축이상 지정했을 경우는 $X > Y > Z > U$ 의 우선 순위로 순위가 높은 축의 Data가 읽어집니다.

7.1 논리위치 Counter 읽기

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
10h	논리위치 Counter 읽기	LP	-2,147,483,648 ~ +2,147,483,647	4 Byte

논리위치 Counter의 현재값이 RR6, 7 Read data register에 설정 됩니다.

7.2 실제위치 Counter 읽기

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
11h	실제위치 Counter 읽기	EP	-2,147,483,648 ~ +2,147,483,647	4 Byte

실제위치 Counter의 현재값이 RR6, 7 Read data register에 설정 됩니다.

7.3 현재 Drive 속도 읽기

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
12h	현재 Drive 속도 읽기	CV	1 ~ 8,000	2 Byte

Drive중, 현재 Drive 속도의 값이 RR6, 7 Read data register에 설정 됩니다. Drive 정지시에는 0으로 설정되며, Data의 단위는 Drive속도 설정치(V)와 같습니다.

7.4 현재 가/감속도 읽기

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
13h	현재가/감속도 읽기	CA	1 ~ 8,000	2 Byte

Drive중, 현재 가속도 또는 감속도의 값이 RR6, 7 Read data register에 설정 됩니다.

Data의 단위는 가속도 설정치(A)와 같습니다.

7.5 동기 동작 버퍼 Register 읽기

명령코드	명 령	Parameter 기호	Data 범위	Data 길이
14h	동기 동작 버퍼 Register 읽기	BR	-2,147,483,648 ~ +2,147,483,647	4 Byte

동기 동작 버퍼 Register의 값이 RR6, 7 Read data register에 설정 됩니다.

8. Drive 명령

Drive 명령은, 각 축의 Drive pulse를 출력하는 명령과 출력에 필요한 부수적인 명령입니다. 기록 Data는 필요하지 않고, WR0 Command register에 축 지정과 명령 코드를 기입하면, 즉시 실행됩니다. 복수의 축을 지정하여, 같은 명령을 동시에 실행할 수도 있습니다.

Drive 중에는 RR0주 status register의 각 축의 n-DRV bit가 1이 되고 Drive가 종료하면 n-DRV Bit는 0으로 돌아옵니다.

서보모터 Driver용의 nINPOS 신호를 유효로 설정하면, nINPOS 입력 신호가 Active level이 되고 RR0 주 Status register의 n-DRV bit는 0으로 돌아옵니다.

【주의사항】 Drive 명령의 명령처리에 필요로 하는 시간은 최대 250 nsec(CLK=16MHz의 경우)입니다.

다음 명령을 기입할 때에는, 이 시간 이후에 설정합니다.

8.1 + 방향 정량 Pulse drive

명령코드	명 령
20h	+ 방향 정량 Pulse drive

설정되어 있는 출력 Pulse수를 $nP+P/nP+N$ 출력 신호에 Pulse를 출력합니다.

Drive중은, Drive pulse를 1 pulse 출력할 때 마다 논리위치 Counter가 1개 Count up합니다.

Drive 명령 기입 전에 출력시키고 싶은 속도 커브에 필요한 Parameter와 출력 Pulse수가 올바르게 설정되어 있어야 합니다.

○ : 설정이 필요

Parameter	출력 시키고 싶은 속도 커브				
	정속	대칭 직선 가감속	비대칭 직선 가감속	대칭 S자 가감속	비대칭 S자 가감속
Range(R)	○	○	○	○	○
가속도 증가율(K)				○	○
감속도 증가율(L)					○
가속도(A)		○	○	○(8000)	○(8000)
감속도(D)			○		○(8000)
기동속도(SV)	○	○	○	○	○
Drive속도(V)	○	○	○	○	○
출력 pulse수(P)	○	○	○	○	○
메뉴얼 감속점(DP)					○

8.2 - 방향 정량 Pulse drive

명령코드	명 령
21h	- 방향 정량 Pulse drive

설정되어 있는 출력 Pulse수를 $nP-P/nP-N$ 출력 신호에 Pulse를 출력합니다.

Drive중에는 Drive pulse를 1 pulse 출력할 때 마다 논리위치 Counter가 1개 Count down 합니다.

Drive 명령 기입 전에 출력시키고 싶은 속도 커브에 필요한 Parameter와 출력 Pulse수가 올바르게 설정되어 있어야 합니다.

8.3 + 방향 연속 Pulse drive

명령코드	명 령
22h	+ 방향 연속 Pulse drive

정지 Command 또는 지정된 외부 신호가 Active될 때까지, 연속해서 $nP+P/nP+N$ 출력 신호에 Pulse를 출력합니다.

Drive중에는 Drive pulse를 1 pulse 출력할 때 마다 논리위치 Counter가 1개 Count up합니다.

Drive 명령을 기입하기 전에 출력시키고 싶은 속도 커브에 필요한 Parameter가 올바르게 설정되어 있어야 합니다.

8.4 - 방향 연속 Pulse drive

명령코드	명 령
23h	- 방향 연속 Pulse drive

정지 Command 또는 지정된 외부 신호가 Active될 때까지, 연속해서 $nP-P/nP-N$ 출력 신호에 Pulse를 출력합니다.

Drive중에는 Drive pulse를 1 pulse 출력할 때 마다 논리위치 Counter가 1개 Count down합니다.

Drive 명령을 기입하기 전에 출력시키고 싶은 속도 커브에 필요한 Parameter가 올바르게 설정되어 있어야 합니다.

8.5 Drive 시작 홀드(Hold)

명령코드	명 령
24h	Drive 시작 Hold

Drive의 시작을 일시정지시켜 놓습니다. 복수의 축 Drive를 동시 스타트 시킬 때 사용합니다.

동시 스타트 시키고 싶은 축에 본 명령을 기입한 후, Drive 시작 명령(25h)으로 전 축 동시에 Drive를 시작합니다.

Drive중에 본 명령을 기입해도, Drive는 정지하지 않습니다.

8.6 Drive 시작(Free)/종료 Status clear

명령코드	명령
25h	Drive 시작/종료 Status clear

Drive 시작 Hold 명령(24h)에 의해 Drive시작이 Hold 되고 있는 상태를 해제합니다.

RR1 register의 drive 종료 Status bit D15~8을 clear 합니다.

RR2 register의 자동원점복귀 IN2 신호 에러 bit D7(HOME)를 clear 합니다.

8.7 Drive 감속정지

명령코드	명령
26h	Drive 감속정지

Drive pulse 출력 중에 감속정지시킵니다. Drive 속도가 기동속도보다 낮은 경우에는 즉시정지합니다.

보간 Drive중에 주축에 대해서 본 명령 또는 Drive 즉시정지 명령을 기입하면, 보간 Drive는 정지합니다.

Drive가 정지하고 있을 때의 기입은 아무런 동작을 하지 않습니다.

8.8 Drive 즉시정지

명령코드	명령
27h	Drive 즉시정지

Drive pulse 출력 중에 즉시정지시킵니다. 가감속 Drive에 대해서도 즉시정지합니다.

Drive가 정지하고 있을 때의 기입은 아무런 동작을 하지 않습니다.

9. 보간 명령

보간 명령은 2축/3축 직선보간, CW/CCW원호 보간, 2축/3축 Bit pattern보간, 및 보간 Drive에 필요한 부수적인 명령으로 구성됩니다. 보간 명령은 WR0 commend register의 D11~8 Bit의 축 지정이 필요 없습니다.

0으로 설정 합니다.

어느 보간을 실시하는 경우도, 보간 Drive를 시작하기 전에 공통으로 필요한 것은, 다음의 2가지 입니다.

①보간을 실시하는 축을 지정한다. (WR5 register의 D5~D0 set.)

②주축으로 지정한 축의 속도 Parameter를 설정 한다.

보간 Drive중에는 RR0주status register의 D8(I-DRV) bit가 1이 되고, Drive가 종료하면 0으로 돌아옵니다.

보간 Drive중에는 보간을 실시하고 있는 축의 n-DRV bit도 1이 됩니다.

【주의 사항】 보간 명령의 명령 처리에 필요로 하는 시간은 최대 250 nsec(CLK=16MHz의 경우)입니다.

다음 명령의 기입은 이 시간의 이후에 설정하십시오.

9.1 2축 직선보간 Drive

명령코드	명령
30h	2축 직선보간 Drive

현재좌표에서 종점좌표까지 2축 직선보간 합니다.

Drive전에, 보간을 실시하는 2축의 각 종점을 상대치로 출력 Pulse(P)에 설정합니다.

9.2 3축 직선보간 Drive

명령코드	명령
31h	3축 직선보간 DDrive

현재좌표로부터 종점좌표까지 3축 직선보간 합니다.

Drive전에, 보간을 실시하는 3축의 각 종점을 상대치로 출력 Pulse(P)에 설정합니다.

9.3 CW원호보간 Drive

명령코드	명령
32h	CW 원호보간 Drive

지정된 중심좌표를 중심으로, 현재좌표에서 종점좌표까지 시계방향으로 원호보간 합니다.

Drive전에, 보간을 실시하는 2축의 현재위치를 원호 중심점(C)에 설정하고, 현재위치에 대한 종점을 상대치로 출력 pulse(P)에 설정합니다. 종점좌표를 (0, 0)으로 설정하면 진원을 그립니다.

9.4 CCW원호보간 Drive

명령코드	명 령
33h	CCW 원호보간 Drive

지정된 중심 좌표를 중심으로, 현재좌표에서 종점좌표까지 반시계방향으로 원호보간 합니다.

Drive전에, 보간을 실시하는 2축의 현재위치를 원호 중심점(C)에 설정하고, 현재위치에 대한 종점을 상대치로 출력 Pulse(P)에 설정합니다. 종점좌표를 (0, 0)으로 설정하면 진원을 그립니다.

9.5 2축 Bit pattern보간 Drive

명령코드	명 령
34h	2축 Bit pattern보간 Drive

2축 Bit pattern보간을 실행합니다. Drive전에, 보간을 실시하는 2축의 +방향/-방향의 Bit data를 설정합니다.

Drive전에 설정할 수 있는 Bit data의 크기는 각 축 $16 \times 3 = 48$ bit입니다. 이것을 넘는 경우에는 Drive중에 보충해 갑니다.

9.6 3축 Bit pattern보간 Drive

명령코드	명 령
35h	3축 Bit pattern보간 Drive

3축 Bit pattern보간을 실행합니다. Drive전에, 보간을 실시하는 3축의 +방향/-방향의 Bit data를 설정합니다.

Drive전에 설정할 수 있는 Bit data의 크기는 각 축 $16 \times 3 = 48$ bit입니다. 이것을 넘는 경우에는 Drive중에 보충해 갑니다.

9.7 BP register 기입 가능

명령코드	명 령
36h	BP register 기입 가능

Bit pattern보간의 Bit pattern data를 기입하는 Register(BP1P/M, BP2P/M, BP3P/M)의 기입을 가능하게 하는 명령으로, 이 명령 발행에 의해서 nWR2~nWR5 register의 기입은 할 수 없습니다.

Reset시에는 Bit pattern data 기입은 할 수 없습니다.

9.8 BP register 기입 불가

명령코드	명 령
37h	BP register 기입 불가

Bit pattern보간의 Bit pattern data를 기입하는 Register(BP1P/M, BP2P/M, BP3P/M)에 기입을 불가능하게 설정합니다. 명령에 의해서 nWR2~nWR5 register에의 기입은 가능하게 됩니다.

9.9 BP data stack

명령코드	명 령
38h	BP data stack

Bit pattern data 기입 Register(BP1P/M, BP2P/M, BP3P/M)에 기입해진 Bit pattern data를 register로 이동시켜 저장 합니다. BP data stack 명령을 발행하면, Stack Counter(SC)가 1개 증가합니다.

Stack Counter(SC)가 3이면 본 명령을 기입할 수 없습니다.

9.10 BP data clear

명령코드	명 령
39h	BP data clear

내부에 축적된 Bit pattern data를 모두 Clear하여, Stack Counter(SC)를 0으로 합니다.

9.11 보간 Single step

명령코드	명 령
3Ah	보간 Single step

보간 Drive를 1 pulse마다 Step 출력 합니다. WR5 register의 D12 Bit를 1로 하여, Command에 의한 보간 Step mode로 설정하고, 보간 Drive 명령을 발행하고 나서, Single step을 실시합니다.

9.12 감속 유효

명령코드	명 령
3Bh	감속 유효

가감속으로 보간 Drive를 실시할 때, 자동 감속 또는 메뉴얼 감속을 유효로 합니다.

단독의 보간 Drive를 가감속으로 실시할 때는, Drive전에 반드시 본 명령을 실행할 필요가 있습니다. 연속 보간으로 감속을 무효로 하여, 보간 Drive를 시작합니다. 감속시키는 최종 보간 세그먼트(Segment)의 보간 명령 기입전에 감속 유효 명령을 기입합니다. Reset시에는, 감속 무효 상태가 됩니다. 본 명령에 의해 감속을 유효 상태로 하면, 감속 무효 명령(3C)이 쓰기 전까지, Reset 할 때까지 유효 상태가 됩니다.

감속 유효/무효는, 보간 Drive시에만 가능하며, 각 축을 독립적으로 Drive할 경우에는 자동 감속 또는 메뉴얼 감속은 항상 유효 상태입니다.

9.13 감속 유효

명령코드	명 령
3Ch	감속 유효

가감속으로 보간 Drive를 실시할 때, 자동 감속 또는 메뉴얼 감속을 무효 상태로 합니다.

9.14 보간 Interrupt clear

명령코드	명 령
3Dh	보간 Interrupt clear

Bit pattern보간, 또는 연속보간으로 발생한 Interrupt를 Clear합니다.

Bit pattern보간에서는 WR5 register의 D15 bit를 1로 하면, Stack counter(SC)가 2에서 1로 바뀌어 Interrupt가 발생합니다. 연속보간에서는 WR5 register의 D14 bit를 1로 하면 다음 보간 세그먼트(Segment)의 Data 및 보간 Drive 명령이 기입되면 Interrupt가 발생합니다.

10. 그 밖의 명령

【주의사항】 명령의 명령 처리에 필요로 하는 시간은 최대 250 nsec(CLK=16MHz의 경우)입니다.

다음 명령을 기입할 때에는 이 시간 이후에 기입하십시오.

10.1 자동원점복귀 실행

명령코드	명 령
62h	자동원점복귀 실행

자동원점복귀를 실행합니다. 실행전에, 자동원점복귀 Mode나 각 Parameter를 올바르게 설정해 둘 필요가 있습니다. 자동원점복귀의 자세한 것은 2.5절을 참조하십시오.

10.2 동기 동작 기동

명령코드	명 령
65h	동기 동작 기동

동기 동작을 본 명령에 의해서 기동시킵니다. 동기 동작 Mode 설정 명령에 의해, 기동 요인의 WR6/D9(CMD) bit를 1로 설정해야 합니다. 동기 동작의 자세한 것은 2.6절을 참조하십시오.

10.3 NOP

명령코드	명 령
0Fh	NOP

명령은 아무것도 실행되지 않습니다.

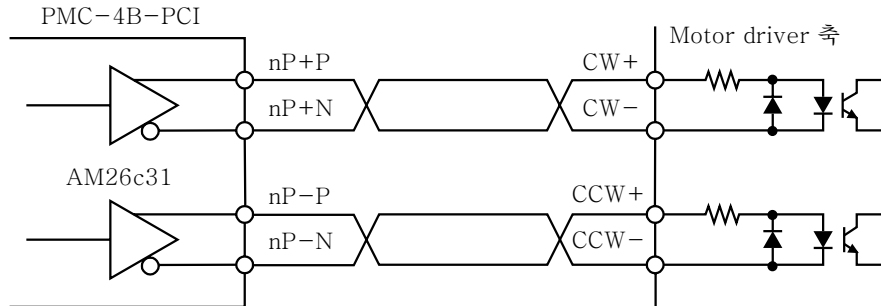
각 축의 WR1~3 register, RR1~3 register를 선택하여 축 변경에 사용합니다.

11. 입/출력 신호 접속예

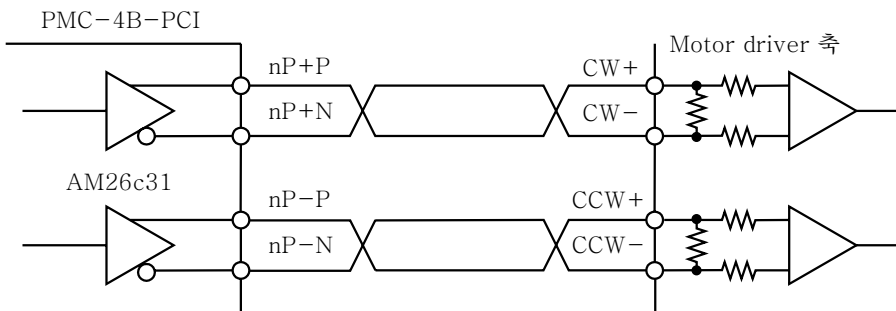
11.1 Drive pulse 출력 신호의 접속(nP+P/N, nP-P/N)

Driver pulse의 출력은 +방향/-방향의 Drive pulse신호를 차동출력의 Line driver(AM26c31)를 이용하여 출력합니다. Poto coupler 및 Line driver입력을 갖는 Motor driver와의 접속예를 나타냅니다.

◎Poto coupler 입력을 갖는 Motor driver와의 접속 예



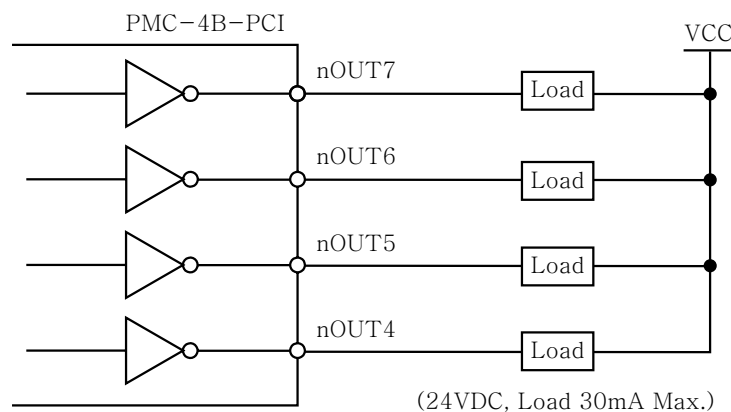
◎Line driver 입력을 갖는 Motor driver와의 접속 예



Drive pulse 출력 신호는, EMC를 고려해, Twisted pair 쉴드(Shield)선을 사용하는 것을 추천합니다.

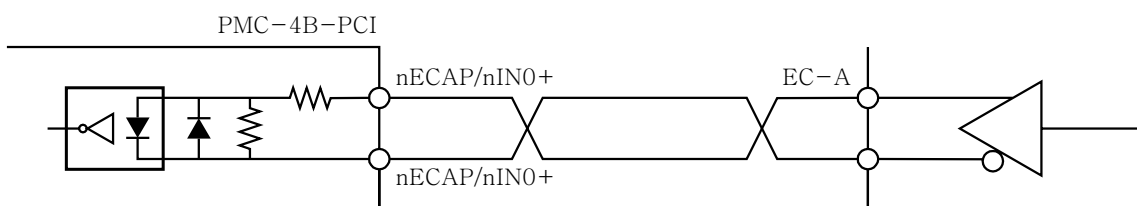
11.2 범용 출력 신호의 접속 (nOUT7~4)

출력 신호는 버퍼(74LS06)를 출력되며 Reset 후에는 모든 출력은 OFF됩니다.



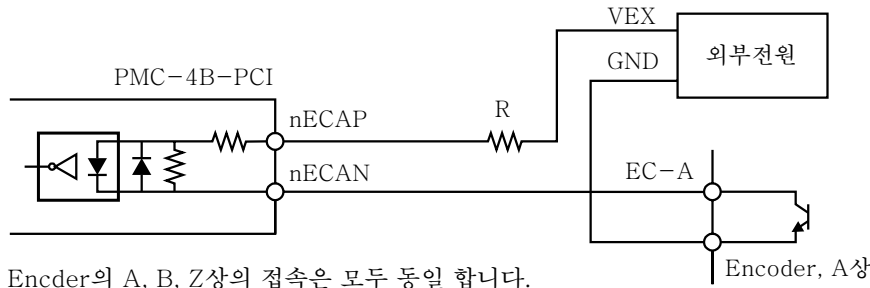
11.3 Encoder 입력 신호의 접속

◎Encoder 입력신호와 차동출력의 Line driver와의 접속 예



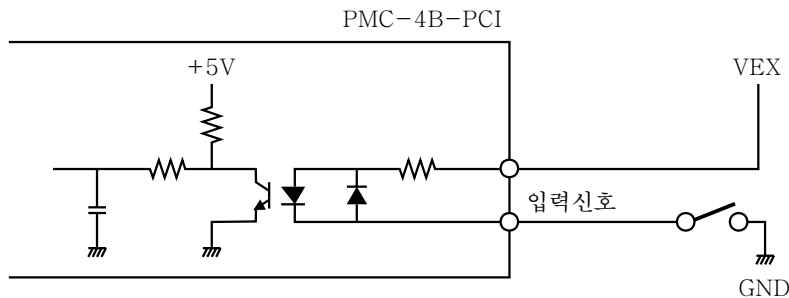
Encoder의 A, B, C상 모두 접속은 동일 합니다.

◎Encoder 입력신호와 Open collector출력 Encoder와의 접속 예



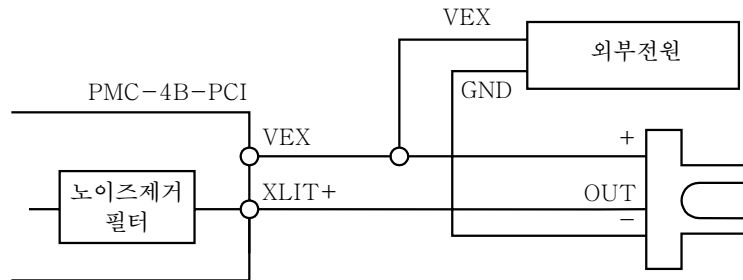
외부전원전압	저항(R)
5V	0
12V	820 1/4W
24V	2k 12W

11.4 입력 신호의 접속(nIN0, 1, 3, nINPOS, nALRAM, nEXP+/-, EMG)



11.5 Limit 입력 신호의 접속(nLMIT+/-)

Limit신호는 일반적으로, 외부로 배선의 노출이 불가피하므로 노이즈에 취약합니다. Photo-coupler 만으로는 노이즈제거가 불가능하므로 PMC-4B-PCI내의 필터 기능을 유효하게 하여, 적당한 통과 시간(FL=2, 3)을 설정합니다.



12. 제어 프로그램의 예

이 장에서는, C언어에 의한 모션 컨트롤러의 제어 프로그램의 예를 나타냅니다. 16 Bit bus 구성의 프로그램입니다.

```
#include <stdio.h>
#include <conio.h>
// ----- 모션컨트롤 IC register address 정의 -----
#define adr 0x2a0 // 베이스 address
#define wr0 0x0 //command register
#define wr1 0x2 //mode register 1
#define wr2 0x4 //mode register 2
#define wr3 0x6 //mode register 3
#define wr4 0x8 //output register
#define wr5 0xa //보간 mode register
#define wr6 0xc //하위 write data register
#define wr7 0xe //상위 write data register
#define rr0 0x0 //주소 status register
#define rr1 0x2 //status register 1
#define rr2 0x4 //status register 2
#define rr3 0x6 //status register 3
#define rr4 0x8 //input register 1
#define rr5 0xa //input register 2
#define rr6 0xc //하위 read data register
#define rr7 0xe //상위 read data register
```

```

#define          bp1p          0x4          //B P 제 1축+방향 data register
#define          bp1m          0x6          //B P 제 1축-방향 data register
#define          bp2p          0x8          //B P 제 2축+방향 data register
#define          bp2m          0xa          //B P 제 2축-방향 data register
#define          bp3p          0xc          //B P 제 3축+방향 data register
#define          bp3m          0xe          //B P 제 3축-방향 data register
// wreg1(축지정, data) ----- write register 1 설정
void wreg1(int axis, int wdata)
{
    outpw(adr+wr0, (axis << 8) + 0xf); //축지정
    outpw(adr+wr1, wdata);
}
// wreg2(축지정, data) ----- write register 2 설정
void wreg2(int axis, int wdata)
{
    outpw(adr+wr0, (axis << 8) + 0xf); //축지정
    outpw(adr+wr2, wdata);
}
// wreg3(축지정, data) ----- write register 3 설정
void wreg3(int axis, int wdata)
{
    outpw(adr+wr0, (axis << 8) + 0xf); //축지정
    outpw(adr+wr3, wdata);
}
// command(축지정, 명령 코드)----- 명령 기입
void command(int axis, int cmd)
{
    outpw(adr+wr0, (axis << 8) + cmd);
}
// range(축지정, data) ----- range(R) 설정
void range(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x00);
}
// acac(축지정, data) ----- 가속도 증가율(K) 설정
void acac(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x01);
}
// dcac(축지정, data) ----- 감속도 증가율(L) 설정
void dcac(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x0e);
}
// acc(축지정, data) ----- 가속도(A) 설정
void acc(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x02);
}

```

```

// dec(축지정, data) ----- 감속도(D) 설정
void dec(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x03);
}
// startv(축지정, data) ----- 기동속도(SV) 설정
void startv(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x04);
}
// speed(축지정, data) ----- drive 속도(V) 설정
void speed(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x05);
}
// pulse(축지정, data) ----- 출력 pulse수/중점(P) 설정
void pulse(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x06);
}
// decp(축지정, data) ----- 메뉴얼 감속점(DP) 설정
void decp(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x07);
}
// center(축지정, data) ----- 원호 중심점(C) 설정
void center(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x08);
}
// lp(축지정, data) ----- 논리 위치 counter(LP) 설정
void lp(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x09);
}
// ep(축지정, data) ----- 실제 위치 counter(EP) 설정
void ep(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x0a);
}

```

```

// compp(축지정, data) ----- COMP + (CP) 설정
void compp(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x0b);
}
// compm(축지정, data) ----- COMP - (CM) 설정
void compm(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x0c);
}
// accofst(축지정, data) ----- 가속 counter 오프set(offset) (AO) 설정
void accofst(int axis, long wdata)
{
    outpw(adr+wr7, (wdata >> 16) & 0xffff);
    outpw(adr+wr6, wdata & 0xffff);
    outpw(adr+wr0, (axis << 8) + 0x0d);
}
// hsspeed(축지정, data) ----- 원점 검출 속도(HV) 설정
void hsspeed(int axis, int wdata)
{
    outpw(adr+wr6, wdata);
    outpw(adr+wr0, (axis << 8) + 0x61);
}
// expmode(축지정, data) ----- 확장 mode(EM) 설정
void expmode(int axis, int em6data, int em7data)
{
    outpw(adr+wr6, em6data);
    outpw(adr+wr7, em7data);
    outpw(adr+wr0, (axis << 8) + 0x60);
}
// syncmode(축지정, data) ----- 동기 동작 mode(SM) 설정
void syncmode(int axis, int sm6data, int sm7data)
{
    outpw(adr+wr6, sm6data);
    outpw(adr+wr7, sm7data);
    outpw(adr+wr0, (axis << 8) + 0x64);
}
// readlp(축지정) ----- 논리 위치 counter치(LP) 읽기
long readlp(int axis)
{
    long a; long d6; long d7;
    outpw(adr+wr0, (axis << 8) + 0x10);
    d6 = inpw(adr+rr6); d7 = inpw(adr+rr7);
    a = d6 + (d7 << 16);
    return(a);
}
// readep(축지정) ----- 실제 위치 counter치(EP) 읽기
long readep(int axis)
{
    long a; long d6; long d7;
    outpw(adr+wr0, (axis << 8) + 0x11);
}

```

```

d6 = inpw(adr+rr6) ;d7 = inpw(adr+rr7);
a = d6 + (d7 << 16);
return(a);
}
// wait(축지정) ----- drive 종료 대기
void wait(int axis)
{
while(inpw(adr+rr0) & axis);
}
// next_wait() ----- 연속 보간다음 data set 대기
void next_wait(void)
{
while((inpw(adr+rr0) & 0x0200) == 0x0);
}
// bp_wait() ----- BP보간다음 data set 대기
void bp_wait(void)
{
while((inpw(adr+rr0) & 0x6000) == 0x6000);
}
// homesrch() ----- 전축 · 원점 서치
//
// ----- X축원점 서치-----
// Step1 -방향에 20,000pps 로 원점 근접(IN0) 신호 고속 서치
// Step2 -방향에 500pps 로 원점(IN1) 신호 저속 서치
// Step3 -방향에 500pps 로 Z상(IN2) 신호 저속 서치
// Z상검출시, 편차 counter clear 출력
// Step4 +방향에 20,000pps 로 3500 pulse 오프set(offset) 고속 이동
// ----- Y축원점 서치-----
// Step1 -방향에 20,000pps 로 원점 근접(IN0) 신호 고속 서치
// Step2 -방향에 500pps 로 원점(IN1) 신호 저속 서치
// Step3 -방향에 500pps 로 Z상(IN2) 신호 저속 서치
// Z상검출시, 편차 counter clear 출력
// Step4 +방향에 20,000pps 로 700 pulse 오프set(offset) 고속 이동
// ----- Z축원점 서치-----
// Step1 고속 서치 : 없음
// Step2 +방향에 400pps 로 원점(IN1) 신호 저속 서치
// Step3 Z상서치 : 없음
// Step4 -방향에 400pps 로 20 pulse 오프set(offset) 이동
// ----- U축원점 서치-----
// Step1 고속 서치 : 없음
// Step2 -방향에 300pps 로 원점(IN1) 신호 저속 서치
// Step3 Z상서치 : 없음
// Step4 오프set(offset) 이동 : 없음
void homesrch(void)
{
// X, Y축원점 서치의 parameter 설정
// (mode 설정은 main 의 초기설정을 참조)
speed(0x3, 2000); // Step1, 4 고속 속도: 20000pps
hsspeed(0x3, 50); // Step2, 3 저속 속도: 500pps
pulse(0x1, 350 0); // X축오프set(offset):3500 pulse
pulse(0x2, 700 ); // Y축오프set(offset):700 pulse
// Z축원점 서치의 parameter 설정
speed(0x4, 40); // Step4 이동 속도: 400pps
hsspeed(0x4, 40); // Step2 서치 속도: 400pps
pulse(0x4, 20) ; // 오프set(offset):20 pulse

```

```

// U축원점 서치의 parameter 설정
hsspeed(0x8, 30); // Step2 서치 속도: 300pps
command(0xf, 0 x62); // 전축자동 원점 복귀 실행
wait(0xf); // 전축종료 대기
if(inpw(adr+rr0) & 0x0010) // 에러 표시
{
printf("X-axis Home Search Error \n");
}
if(inpw(adr+rr0) & 0x0020)
{
printf("Y-axis Home Search Error \n");
}
if(inpw(adr+rr0) & 0x0040)
{
printf("Z-axis Home Search Error \n");
}
if(inpw(adr+rr0) & 0x0080)
{
printf("U-axis Home Search Error \n");
}
}
void main(void)
{
int count;
outpw(adr+wr0 , 0x8000); //소프트 reset
for(count = 0; count < 2; ++count);
command(0x3, 0 xf); //----- X, Y축mode 설정-----
outpw(adr+wr1 , 0x0000); //mode register 1
//D15~8: 0 interrupt 모두 금지
//D7: 0 IN3 신호:무효
//D6: 0 IN3 신호 논리 :Low active
//D5: 0 IN2 신호:무효
//D4: 0 IN2 신호 논리 :Low active
//D3: 0 IN1 신호:무효
//D2: 0 IN1 신호 논리 :Low active
//D1: 0 IN0 신호:무효
//D0: 0 IN0 신호 논리 :Low active
outpw(adr+wr2 , 0xe000); //mode register 2
//D15:1 INPOS 입력:유효
//D14:1 INPOS 입력 논리 :Hi active
//D13:1 ALARM 입력:유효
//D12:0 ALARM 입력 논리 :Low active
//D11:0
//D10:0 encoder 입력 분주:1/1
//D9: 0 encoder 입력 방식:2상pulse
//D8: 0 drive pulse 방향 논리:
//D7: 0 drive pulse 논리:정논리
//D6: 0 drive pulse 방식:2 pulse
//D5: 0 COMP 대상:논리 위치 counter
//D4: 0 -limit 논리 :Low active
//D3: 0 +limit 논리 :Low active
//D2: 0 limit 정지 mode:감속 정지
//D1: 0 소프트 limit -:무효
//D0: 0 소프트 limit +:무효

```

```

outpw(adr+wr3, 0x0000);          //mode register 3
//D15~12:0000
//D11:0 범용 출력 OUT7:Low
//D10:0 범용 출력 OUT6:Low
//D9: 0 범용 출력 OUT5:Low
//D8: 0 범용 출력 OUT4:Low
//D7: 0 drive 상태 출력:무효
//D6: 0
//D5: 0
//D4: 0 외부 조작 신호 동작:무효
//D3: 0
//D2: 0 가감 속도 커브 : 직선 가감 속도(사다리꼴)
//D1: 0 가감 속도의 대칭/비대칭:대칭
//D0: 0 정량 pulse drive의 감속:자동 감속
expmode(0x3, 0x5d08, 0x497f);    //확장 mode
//[입력 신호 필터 그 외]
//W6/D15~13:010 입력 신호 필터 지연:512μ
//W6/D12:1 IN3 신호 필터:유효
//W6/D11:1 EXPP, EXPM, EXPLS 필터:유효
//W6/D10:1 INPOS, ALARM 신호 필터:유효
//W6/D9: 0 IN2 신호 필터:무효
//W6/D8: 1 EMGN, LMTP/M, IN1, 0 필터:유효
//W6/D7: 0
//W6/D6: 0
//W6/D5: 0 자동 원점 복귀 종료 새치기 : 금지
//W6/D4: 0 LP/EP 가변 링 기능:무효
//W6/D3: 1 직선 가감 속도시의 삼각 방지:유효
//W6/D2: 0 pulse 출력의 교체:무효
//W6/D1: 0 EP 증감 반전:무효
//W6/D0: 0 IN2 신호에 의한 EP clear:무효
//[자동 원점 복귀 mode]
//W7/D15~D13 010 편차 counter clear pulse폭 : 100μsec
//W7/D12 0 편차 counter clear 출력의 논리 level : Hi
//W7/D11 1 편차 counter clear 출력 : 유효
//W7/D10 0 limit 신호를 원점 신호로서 사용 : 무효
//W7/D9 0 Z상신호 AND 원점 신호 : 무효
//W7/D8 1 논리/실제 위치 counter clear : 유효
//W7/D7 0 step 4 이동 방향 : +방향
//W7/D6 1 step 4 : 유효
//W7/D5 1 step 3 검출 방향 : -방향
//W7/D4 1 step 3 : 유효
//W7/D3 1 step 2 검출 방향 : -방향
//W7/D2 1 step 2 : 유효
//W7/D1 1 step 1 검출 방향 : -방향
//W7/D0 1 step 1 : 유효
//----- X, Y축동작 parameter 초기설정-----
accfst(0x3, 0);                  // AO = 0
range(0x3, 800000);              // R = 800000(배율= 10)
acac(0x3, 1010);                 // K = 1010 (가/감속 증가율= 619kpps/sec2 )
dcac(0x3, 1010);                 // L = 1010 (감속도 증가율= 619kpps/sec2 )
acc(0x3, 100);                   // A = 100 (가/감속도= 125kpps/sec)
dec(0x3, 100);                   // D = 100 (감속도= 125kpps/sec)
startv(0x3, 100);                // SV= 100 (기동속도= 1000pps)
speed(0x3, 4000);                // V = 4000 (drive 속도= 40000pps)
pulse(0x3, 100000);              // P = 100000 (출력 pulse수= 100000)

```

```

lp(0x3, 0); // LP= 0 (논리 위치 counter= 0)
ep(0x3, 0); // EP= 0 (실리 위치 counter= 0)
command(0xc, 0 xf); //----- Z, U축mode 설정-----
outpw(adr+wr1, 0x0000); //mode register 1
//D15~8: 0 interrupt 모두 금지
//D7: 0 IN3 신호:무효
//D6: 0 IN3 신호 논리 :Low active
//D5: 0 IN2 신호:무효
//D4: 0 IN2 신호 논리 :Low active
//D3: 0 IN1 신호:무효
//D2: 0 IN1 신호 논리 :Low active
//D1: 0 IN0 신호:무효
//D0: 0 IN0 신호 논리 :Low active
outpw(adr+wr2, 0x0000); //mode register 2
//D15:0 INPOS 입력:무효
//D14:0 INPOS 입력 논리 :Low active
//D13:0 ALARM 입력:무효
//D12:0 ALARM 입력 논리 :Low active
//D11:0
//D10:0 encoder 입력 분주:1/1
//D9: 0 encoder 입력 방식:2상pulse
//D8: 0 drive pulse 방향 논리:
//D7: 0 drive pulse 논리:정논리
//D6: 0 drive pulse 방식:2 pulse
//D5: 0 COMP 대상:논리 위치 counter
//D4: 0 -limit 논리 :Low active
//D3: 0 +limit 논리 :Low active
//D2: 0 limit 정지 mode:감속 정지
//D1: 0 소프트 limit-:무효
//D0: 0 소프트 limit+:무효
outpw(adr+wr3, 0x0000); //mode register 3
//D15~12:0000
//D11:0 범용 출력 OUT7:Low
//D10:0 범용 출력 OUT6:Low
//D9: 0 범용 출력 OUT5:Low
//D8: 0 범용 출력 OUT4:Low
//D7: 0 drive 상태 출력:무효
//D6: 0
//D5: 0
//D4: 0 외부 조작 신호 동작:무효
//D3: 0
//D2: 0 가감 속도 커브 : 직선 가감 속도(사다리꼴)
//D1: 0 가감 속도의 대칭/비대칭:대칭
//D0: 0 정량 pulse drive의 감속:자동 감속
//Z축과 U축의 자동 원점 복귀가 다르므로,
//이하의 확장 mode는 개별적으로 설정한다.
expmode(0x4, 0 x5d08, 0x01c4); //Z축확장 mode
//[입력 신호 필터 그 외]
//W6/D15~13:010 입력 신호 필터 지연:512μ
//W6/D12:1 IN3 신호 필터:유효
//W6/D11:1 EXPP, EXPM, EXPLS 필터:유효
//W6/D10:1 INPOS, ALARM 신호 필터:유효
//W6/D9: 0 IN2 신호 필터:무효
//W6/D8: 1 EMGN, LMTP/M, IN1, 0 필터:유효
//W6/D7: 0

```

```

//W6/D5: 0 자동 원점 복귀 종료 새치기 : 금지
//W6/D4: 0 LP/EP 가변 링 기능:무효
//W6/D3: 1 직선 가감 속도시의 삼각 방지:유효
//W6/D2: 0 pulse 출력의 교체:무효
//W6/D1: 0 EP 증감 반전:무효
//W6/D0: 0 IN2 신호에 의한 EP clear:무효
//[자동 원점 복귀 mode]
//W7/D15~D13 000 편차 counter clear pulse폭 :
//W7/D12 0 편차 counter clear 출력의 논리 level :
//W7/D11 0 편차 counter clear 출력 : 무효
//W7/D10 0 limit 신호를 원점 신호로서 사용 : 무효
//W7/D9 0 Z상신호 AND 원점 신호 : 무효
//W7/D8 1 논리/실제 위치 counter clear : 유효
//W7/D7 1 step 4 이동 방향 : -방향
//W7/D6 1 step 4 : 유효
//W7/D5 0 step 3 검출 방향 :
//W7/D4 0 step 3 : 무효
//W7/D3 0 step 2 검출 방향 : +방향
//W7/D2 1 step 2 : 유효
//W7/D1 0 step 1 검출 방향 :
//W7/D0 0 step 1 : 무효
expmode(0x8, 0 x5d08, 0x010c); //U축확장 mode
//[입력 신호 필터 그 외]
//W6/D15~13:010 입력 신호 필터 지연:512μ
//W6/D12:1 IN3 신호 필터:유효
//W6/D11:1 EXPP, EXPM, EXPLS 필터:유효
//W6/D10:1 INPOS, ALARM 신호 필터:유효
//W6/D9: 0 IN2 신호 필터:무효
//W6/D8: 1 EMGN, LMTP/M, IN1, 0 필터:유효
//W6/D7: 0
//W6/D6: 0
//W6/D5: 0 자동 원점 복귀 종료 새치기 : 금지
//W6/D4: 0 LP/EP 가변 링 기능:무효
//W6/D3: 1 직선 가감 속도시의 삼각 방지:유효
//W6/D2: 0 pulse 출력의 교체:무효
//W6/D1: 0 EP 증감 반전:무효
//W6/D0: 0 IN2 신호에 의한 EP clear:무효
//[자동 원점 복귀 mode]
//W7/D15~D13 000 편차 counter clear pulse폭 :
//W7/D12 0 편차 counter clear 출력의 논리 level :
//W7/D11 0 편차 counter clear 출력 : 무효
//W7/D10 0 limit 신호를 원점 신호로서 사용 : 무효
//W7/D9 0 Z상신호 AND 원점 신호 : 무효
//W7/D8 1 논리/실제 위치 counter clear : 유효
//W7/D7 0 step 4 이동 방향 :
//W7/D6 0 step 4 : 무효
//W7/D5 0 step 3 검출 방향 :
//W7/D4 0 step 3 : 무효
//W7/D3 1 step 2 검출 방향 : -방향
//W7/D2 1 step 2 : 유효
//W7/D1 0 step 1 검출 방향 :
//W7/D0 0 step 1 : 무효
//----- Z, U축동작 parameter 초기설정-----
accfst(0xc, 0); // AO = 0
range(0xc, 800000); // R = 800000(배율= 10)

```

```

acac(0xc, 1010); // K = 1010 (가/감속도 증가율= 619kpps/sec2 )
dcac(0xc, 1010); // L = 1010 (감속도 증가율= 619kpps/sec2 )
acc(0xc, 100); // A = 100 (가/감속도= 125kpps/sec)
dec(0xc, 100); // D = 100 (감속도= 125kpps/sec)
startv(0xc, 50); // SV= 50 (기동속도= 500pps)
speed(0xc, 40); // V = 40 (drive 속도= 400pps)
pulse(0xc, 10); // P = 10 (출력 pulse수= 10)
lp(0xc, 0); // LP= 0 (논리 위치 counter= 0)
//----- 범용 출력 register 초기설정--
outpw(adr+wr4, 0x0000); // 00000000 00000000
//----- 보관 mode register 초기설정--
outpw(adr+wr5, 0x0124); // 00000001 00100100
// ax1=x, ax2=y, ax3=z, 선속일정
//----- drive 시작-----
// homesrch(); //----- 전속원점 서치-----
//----- X, Y축직선 가감 속도 drive-----
acc(0x3, 200); // A = 200 (가/감속도= 250kpps/sec)
speed(0x3, 4000); // V = 4000 (drive 속도= 40000pps)
pulse(0x1, 80000); // xP = 80000
pulse(0x2, 40000); // yP = 40000
command(0x3, 0 x20); // +정량 pulse drive
wait(0x3); // drive 종료 대기
//----- X축비대칭 직선 가감 속도 drive-----
wreg3(0x1, 0x0002); //가속 · 감속 개별(비대칭) mode
acc(0x1, 200); // xA = 200 (가/감속도= 250kpps/sec)
dec(0x1, 50); // xD = 50 (감속도= 62.5kpps/sec)
speed(0x1, 4000); // xV = 4000 (drive 속도= 40000pps)
pulse(0x1, 80000); // xP = 80000
command(0x1, 0 x20); // +정량 pulse drive
wait(0x1); // drive 종료 대기
wreg3(0x1, 0x0000); // 가속 · 감속 개별 mode 해제
//----- X, Y축S자 가감 속도 drive-----
wreg3(0x3, 0x0004); //S자 mode
acac(0x3, 1010); // K = 1010 (가속도 증가율= 619kpps/sec2 )
acc(0x3, 200); // A = 200 (가/감속도= 250kpps/sec)
speed(0x3, 4000); // V = 4000 (drive 속도= 40000pps)
pulse(0x1, 50000); // xP = 50000
pulse(0x2, 25000); // yP = 25000
command(0x3, 0 x21); // -정량 pulse drive
wait(0x3);
wreg3(0x3, 0x0000); // S자 가감 속도 mode 해제
//----- Z축정속 drive-----
startv(0x4, 40); // SV= 40 (기동속도= 400pps)
speed(0x4, 40); // V = 40 (drive 속도= 400pps)
pulse(0x4, 700); // P = 700
command(0x4, 0 x20); // +정량 pulse drive
wait(0x4); // (400pps 로 700 pulse+방향에 이동)
pulse(0x4, 350); // P = 350
command(0x4, 0 x21); // -정량 pulse drive
wait(0x4); // (400pps 로 350 pulse-방향에 이동)
//----- X, Y축직선 보관 drive-----
outpw(adr+wr5, 0x0124); // ax1=x, ax2=y, ax3=z, 선속일정
range(0x1, 800000); // ax1/R = 800000(배율= 10)
range(0x2, 1131371); // ax2/R = 800000×1.414
speed(0x1, 100); // ax1/V = 100 (drive 속도= 1000pps 정속)

```

```

pulse(0x1, 5000);                // xP = +5000 (중점 X= +5000)
pulse(0x2, -2000);               // yP = -2000 (중점 Y= -2000)
command(0x0, 0x30);              // 2축직선 보간
wait(0x3);
//----- X, Y축원호보간 drive-----
outpw(adr+wr5, 0x0124);           // ax1=x, ax2=y, ax3=z, 선속일정
range(0x1, 800000);              // ax1/R = 800000(배율= 10)
next_wait();
pulse(0x1, -45 00);               // Seg 5
pulse(0x2, 0);
command(0, 0x3 0);
next_wait();
center(0x1, 0);                  // Seg 6
center(0x2, -1 500);
pulse(0x1, -15 00);
pulse(0x2, -15 00);
command(0, 0x3 3);
next_wait();
pulse(0x1, 0);                   // Seg 7
pulse(0x2, -15 00);
command(0, 0x3 0);
next_wait();
center(0x1, 15 00);              // Seg 8
center(0x2, 0);
pulse(0x1, 150 0);
pulse(0x2, -15 00);
command(0, 0x3 3);
wait(0x3);
//----- 동기 동작(2.61 절-예 1) -----
// Y 축이 위치 15000 을 통과하면(자),
// Z 축의 +방향 정량 pulse drive 시작.
range(0x6, 800000);              // R = 800000(배율= 10)
acc(0x6, 400);                   // A = 400 (가/감속도= 500kpps/sec)
startv(0x6, 50);                 // SV= 50 (기동속도= 500pps)
speed(0x6, 3000);                // V = 3000 (drive 속도= 30kpps)
pulse(0x2, 50000);               // yP = 50000 (Y 축출력 pulse수)
pulse(0x4, 10000);              // zP = 10000 (Z 축출력 pulse수)
compp(0x2, 15000);               // yCP+ = 15000 (Y 축CMP+)
lp(0x6, 0);                      // LP= 0 (논리 위치 counter= 0)
syncmode(0x2, 0x2001, 0x0000);   // Y 축동기 동작 mode
// 기동 요인 : P≥C+, 타축기동 : Z
// 자축동작 : 없음
syncmode(0x4, 0x0000, 0x0001);   // Z 축동기 동작 mode
// 자축동작 : +방향 정량 pulse drive
command(0x2, 0 x20);             // Y 축+정량 pulse drive 시작
wait(0x6); // Y, Z 축종료 대기
}

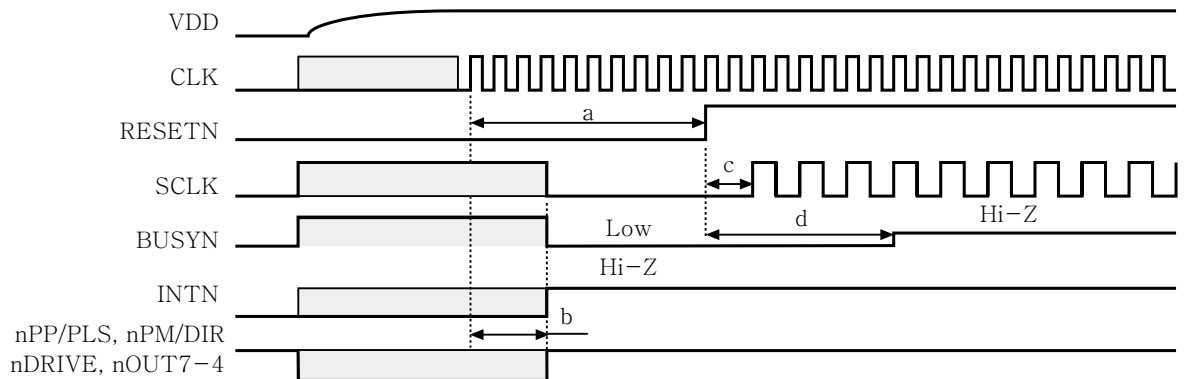
```

13. 정격 사양

전원 전압	PC 내부전원 사용 (5VDC ±10%)
외부 전원 전압	12~24VDC
허용 전압 변동 범위	전원 전압의 90~110%
동작 온도 범위	0℃ ~ +45℃ (단, 결빙 또는 결로되지 않은 상태)
저장온도	-10℃ ~ +55℃ (단, 결빙 또는 결로되지 않은 상태)
사용 주위 습도	35 ~ 85%RH

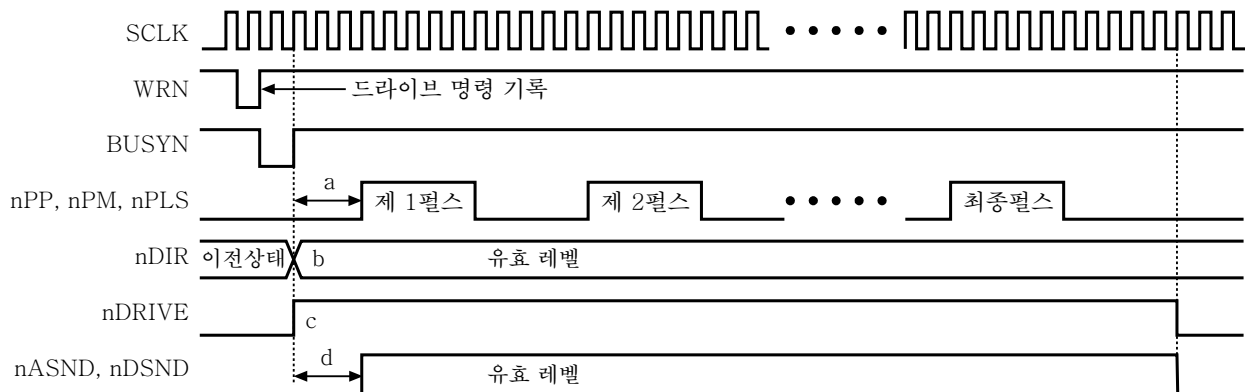
14. 입/출력 신호 타이밍

14.1 파워-온 타이밍



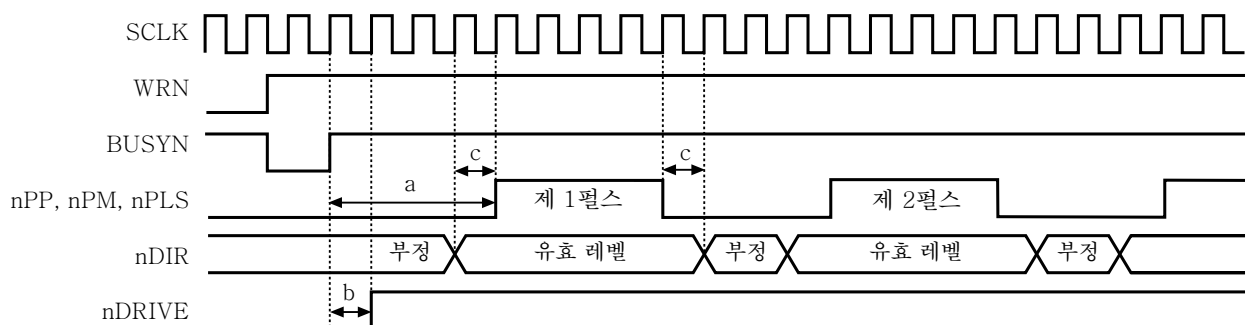
- Reset 입력 신호 RESETN은, CLK 입력 후, $CLK \times 4$ 사이클 이상 Low level이 필요합니다.
- 전원 투입시, 출력 신호는 RESETN이 Low level이며, CLK가 입력이 최대 $CLK \times 4$ 사이클 후에, Level이 확정됩니다.
- SCLK는, RESETN이 Hi level 후, 최대 $CLK \times 2$ 사이클 후에 출력됩니다.
- BUSYN은, RESETN이 Hi level 후, 최대 $CLK \times 8$ 사이클의 후에 확정됩니다. 이사에는, 모션 컨트롤 IC를 Read/Write는 할 수 없습니다.

14.2 Drive 시작/종료 시



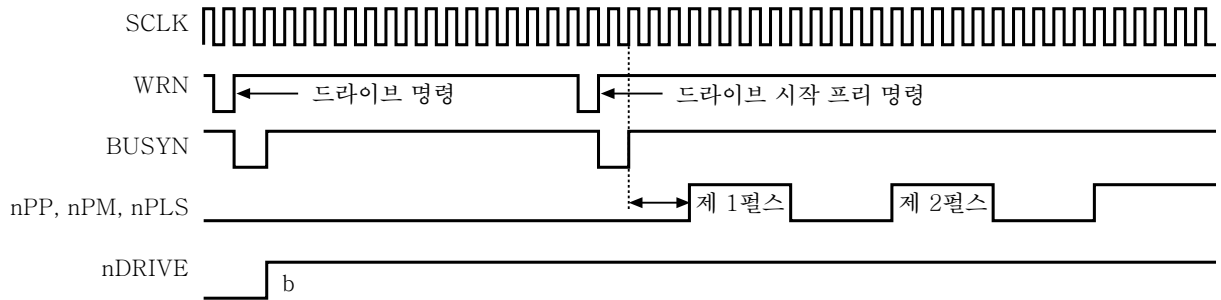
- 그림의 drive pulse (nPP, nPM, nPLS)는 정 pulse의 경우를 나타내고 있습니다. BUSYN의 \uparrow 에서부터 SCLK 3 사이클 후, 제 1 pulse가 출력됩니다.
- Drive 출력 Pulse 방식을 1 pulse 방식으로 설정했을 때의 nDIR(방향) 신호는, BUSYN의 \uparrow 에서 유효 Level로 변화하여, Drive 종료후에도 다음의 Drive 명령이 기입해질 때까지 그 Level을 유지합니다. 다만, 보간 Drive 시는 적용되지 않습니다.
- nDRIVE는, BUSYN의 \uparrow 에서 Hi level이 되어, 최종 Pulse의 Low 기간 후에, Low level로 돌아옵니다.
- nASND, nDSND는 BUSYN의 \uparrow 에서 SCLK 3 사이클 후에 유효 level이 되어, nDRIVE의 최종 Pulse의 Low 시작과 같이 Low level로 됩니다.

14.3 보간 Drive 시



- a. 보간 Drive시의 Drive pulse(nPP, nPM, nPLS)는 BUSYN의 ↑에서 SCLK 4 사이클 후에 제1 pulse가 출력됩니다.
- b. nDRIVE는 BUSYN의 ↑에서 SCLK 1 사이클 후에 Hi level이 됩니다.
- c. Drive 출력 Pulse 방식을 1 pulse 방식으로 설정했을 때, nDIR(방향) 신호는 보간 Drive시, Drive pulse의 Hi level폭과 그 전후 SCLK 1 사이클의 사이, 유효 Level이 됩니다. (Drive pulse : 정논리 pulse일 때)

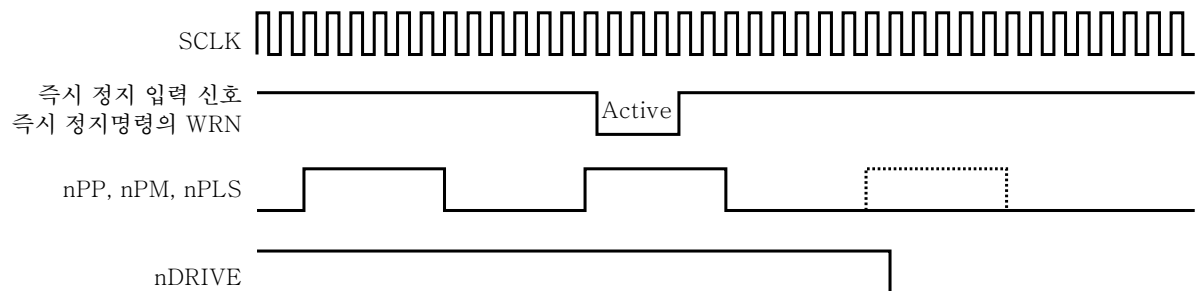
14.4 Drive 시작 프리



- a. 각 축의 Drive pulse(nPP, nPM, nPLS)는 Drive 시작 프리 명령 기입의 BUSYN의 ↑에서 부터 SCLK 3 사이클 후, 동시에 제 1 Pulse가 출력됩니다.
- b. nDRIVE 는 각 축의 Drive 명령 기입의 BUSYN의 ↑에서 각각 Hi level이 됩니다.

14.5 Drive 즉시 정지

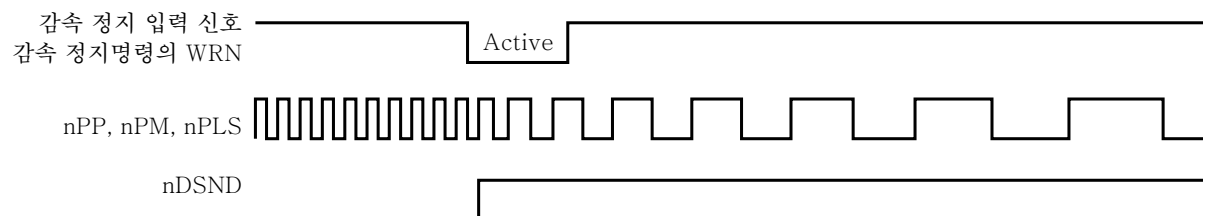
즉시 정지 입력 신호와 즉시 정지 명령의 동작 타이밍입니다. 즉시 정지 입력 신호는 EMG, nLMTP/M(즉시 정지 Mode로 설정 시), nALARM입니다. 즉시 정지 입력 신호가 Active level 또는, 즉시 정지 명령이 기입되면, 현재 출력중의 Drive pulse를 출력한 후에, Pulse 출력을 정지합니다.



즉시정지 입력신호는, 입력신호 필터가 무효인 경우에도 CLK 2 사이클 이상의 Pulse 폭이 필요합니다. 입력신호 필터를 유효로 하면, 필터의 통과시간에 의해 입력신호는 지연 됩니다.

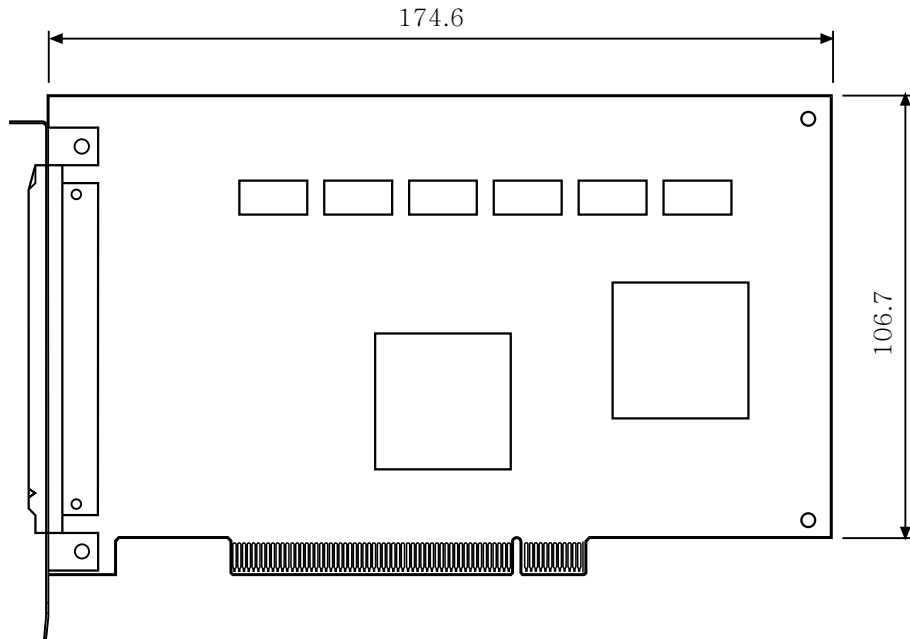
14.6 Drive 감속 정지

감속정지 입력신호와 감속정지 명령의 동작 타이밍입니다. 감속정지 입력신호는 nIN3~0, nLMT+/- (감속 정지 Mode로 설정 시) 입니다. 감속 정지 입력 신호가 Active level 또는, 감속 정지 명령이 기입되면, 현재 출력중의 Drive pulse를 출력한 후에, 감속정지합니다.



입력신호 필터를 유효로 하면, 필터의 통과시간에 의해 입력신호는 지연 됩니다.

15. 외형 치수



(단위:mm)

16. 사양 정리

- ◎제어축 4축
- ◎CPU Data bus 길이 16/8 Bit 선택 가능
- 〈보간기능〉
 - ◎2/3축 직선보간
 - 보간범위 각 축 $-2,147,483,646 \sim +2,147,483,646$
 - 보간속도 1pps~ 4Mpps
 - 보간위치 정밀도 ± 0.5 LSB 이하 (전 보간 범위내에서)
 - ◎원호보간
 - 보간범위 각 축 $-2,147,483,646 \sim +2,147,483,646$
 - 보간속도 1pps~ 4Mpps
 - 보간위치 정밀도 ± 1 LSB 이하 (전 보간 범위내에서)
 - ◎2/3축 Bit pattern 보간
 - 보간속도 1pps~ 4Mpps(단지 CPU data setup 시간에 의존)
 - ◎그 밖의 보간 기능
 - 임의의 축 선택 가능
 - 선속일정
 - 연속보간
 - 보간 Step 전송(Command, 외부 신호)
- 〈각 축 공통 사양〉
 - ◎Driver pulse 출력(CLK=16MHz시)
 - 출력회로 범위 1pps~ 4Mpps
 - 출력속도 정밀도 $\pm 0.1\%$ 이하(설정치에 대하여)
 - 속도 배율 1 ~ 500

- S자용 가가속도 $954 \sim 62.5 \times 10^6 \text{ pps/sec}^2$ (배율=1일 때)
(가속도/감속도의 증가율) $477 \times 10^3 \sim 31.25 \times 10^6 \text{ pps/sec}^2$ (배율=500일 때)
- 가/감속도 $125 \sim 1 \times 10^6 \text{ pps/sec}$ (배율=1일 때)
 $62.5 \times 10^3 \sim 500 \times 10^6 \text{ pps/sec}$ (배율=500일 때)
- 초속도 $1 \sim 8,000 \text{ pps}$ (배율=1일 때)
 $500 \sim 4 \times 10^6 \text{ pps}$ (배율=500일 때)
- Drive 속도 $1 \sim 8,000 \text{ pps}$ (배율=1일 때)
 $500 \text{ pps} \sim 4 \times 10^6 \text{ pps}$ (배율=500일 때)
- 출력 Pulse수 $0 \sim 4,294,967,295$ (정량 pulse drive)
- 속도 Curve 정속 / 대칭 • 비대칭 직선 가감속 / 대칭 • 비대칭 포물선 S자가감속 Drive
- 정량 Pulse drive의 감속 Mode 자동감속(비대칭 직선 가감속 가능) / Manual 감속
- Drive 중의 출력 Pulse, Drive 속도의 변경 가능
- 직선 가감속 정량 Pulse drive의 삼각 방지, S자 가감 속도 정량 Pulse drive의 삼각방지 기능
- 독립 2 pulse / 1 pulse • 방향 방식 선택 가능
- Drive pulse의 논리 Level 선택 가능, 출력 단자 변경 가능

◎Encoder 입력 Pulse

- 2상 Pulse / up down pulse 입력 선택 가능
- 2상 Pulse 1, 2, 4체배 선택 가능

◎위치 Counter

- 논리위치 Counter(출력 pulse용) count범위 $-2,147,483,648 \sim +2,147,483,647$
- 실제위치 Counter(입력 pulse용) count범위 $-2,147,483,648 \sim +2,147,483,647$
상시 write, read 가능
가변 링 Counter 기능, 실제 위치 Counter의 증감 반전 기능, IN2 신호에 의한 실제 위치 Counter clear 기능,
항상 쓰기 및 읽기 가능

◎비교 register

- COMP+ register 위치 비교범위 $-2,147,483,648 \sim +2,147,483,647$
- COMP- register 위치 비교범위 $-2,147,483,648 \sim +2,147,483,647$
- 위치 Counter와의 대소를 Status 출력 및 신호 출력
- Software limit로서 동작 가능

◎자동 원점 복귀

- Step 1(고속 원점 근접 서치) → Step 2(저속 원점 서치) → Step 3(저속 Encoder Z상서치) → Step 4
(고속 Offset 이동)를 차례대로 자동 실행
각 Step의 유효/무효, 검출 방향 선택 가능

◎동기 동작

- 기동 요인
위치 Counter \geq COMP+ 변화, 위치 Counter $<$ COMP+ 변화,
위치 Counter $<$ COMP- 변화, 위치 Counter \geq COMP- 변화, Drive 시작, Drive 종료, IN3 신호 \uparrow , IN3
신호 \downarrow , LP 읽기 명령, 기동 명령. 각 Step의 유효/무효, 검출 방향 선택 가능.
- 동작
+/-정량 Pulse drive 시작, +/-연속 Pulse drive 시작, Drive 감속 정지, Drive 즉시 정지,
위치 Counter값 저장, 위치 Counter set, 출력 Pulse수 Set, Drive 속도 set, 인터럽트 발생
자속의 요인으로 임의의 타속동작 기동 가능

◎Interrupt 기능(보간 제외)

- Interrupt 발생요인
 - 1 drive pulse 출력
 - 위치 counter \geq COMP- 변할 때
 - 위치 counter \geq COMP+ 변할 때
 - 위치 counter $<$ COMP- 변할 때
 - 위치 counter $<$ COMP+ 변할 때
 - 가감속 Drive중의 정속 시작 시
 - 가감속 Drive중의 정속 완료 시
 - Drive 종료 때
 - 자동 원점 복귀 종료
 - 동기 동작.
- 어느 요인에 대하여도 유효 / 무효 선택 가능

◎외부 신호에 의한 Drive 조작

- EXP+, EXP-신호에 의한, +/-방향의 정량/연속 Pulse drive가 가능
- 2상 Encoder 신호 Mode(Encoder 입력) drive 가능.

◎외부 감속정지 / 즉시 정지신호

- IN0 ~ 3 각 축 4점
- 신호의 유효/무효 및 논리 level 선택 가능. 범용 입력으로 사용 가능

◎서보 모터용 입력신호

- ALARM (alarm), INPOS(위치 결정 완료) 신호의 유효/무효 및 논리 Level 선택 가능

◎범용 출력신호

- OUT4~7 각 축 4점(Drive 상태 출력 신호와 단자 공용)

◎Drive 상태 신호 출력

- ASND(가속중), DSND(감속중), CMPP(위치 \geq COMP+), CMPM(위치 $<$ COMP-).
Drive 상태는, Status register에서도 읽기 가능

◎ Overrun limit 신호 입력

- +방향, -방향 각 1점. 논리 Level 선택 가능. Active 시, 즉시 정지/감속 정지 선택 가능

◎긴급 정지 신호 입력

- EMG 1점. Low level로 전축의 Drive pulse를 즉시 정지

◎적분형 필터 내장

- 각 입력 신호의 입력단에 적분 필터를 장비. 통과시간(8 종류) 선택 가능

◎전기적 특성

- 전원전압: PC내부전원 사용(5VDC)
- 외부 전원전압: 12~24VDC
- 동작 온도 범위 : 0 ~ +45℃
- 입력 클럭 : 16.000 MHz(표준)

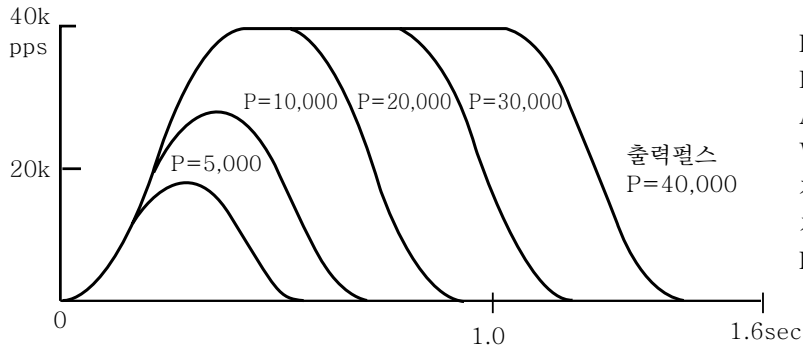
◎그 밖의 보간 기능

- 임의의 축 선택 가능
- 선속일정
- 연속보간
- 보간 Step 전송(Command, 외부 신호)

<부록 A 가감 속도 Drive의 속도 프로파일>

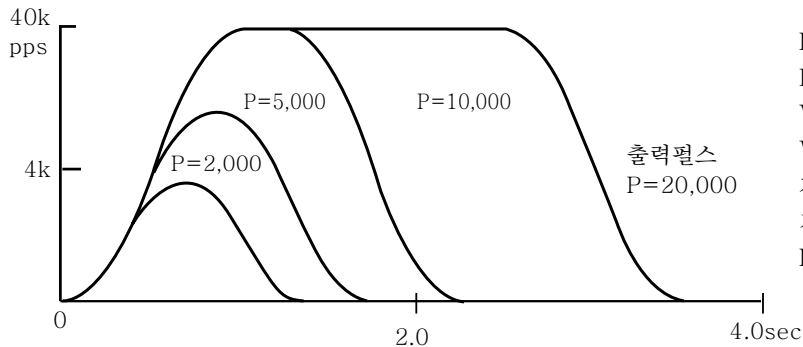
아래와 같은 Parameter를 설정했을 때, 출력되는 Drive pulse의 속도 커브를 표시합니다.

◎40kpps 대칭 S자 가감속



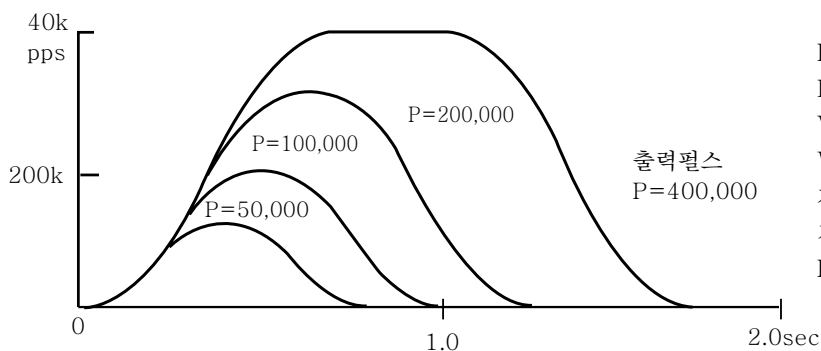
R=800000(배율:10),
K=700(A=8000), SV=10, V=4000,
A0=0
WR3/D2, 1, 0:1, 0, 0 자동 감속 Mode
가속도 증가율= 893kpps/sec²
기동속도= 100pps
Drive 속도= 40kpps

◎8,000pps 대칭 S자 가감속



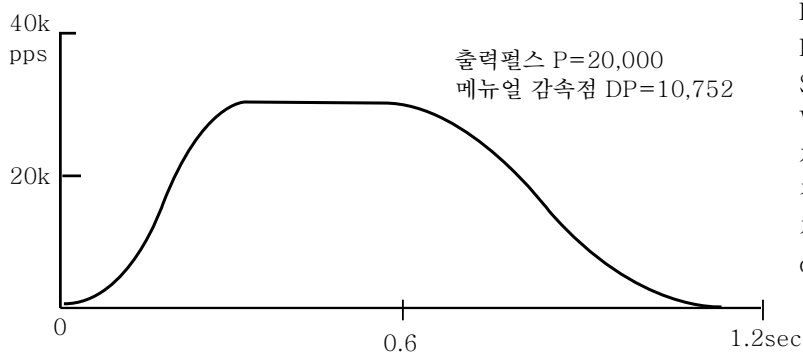
R=8000000(배율:1),
K=2000(A=8000), SV=10,
V=8000, A0=0
WR3/D2, 1, 0:1, 0, 0 자동 감속 Mode
가속도 증가율= 31kpps/sec²
기동속도= 10pps
Drive 속도= 8000kpps

◎400pps 대칭 S자 가감속



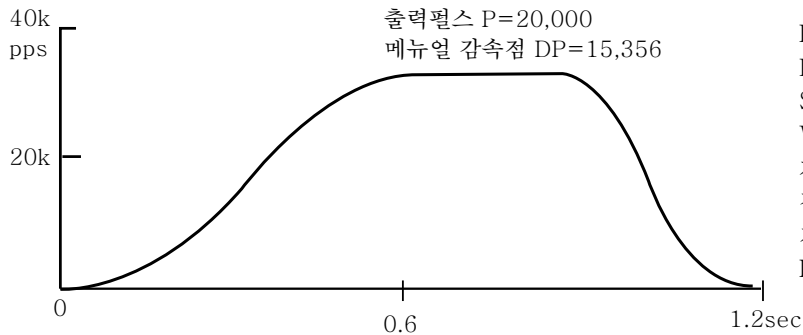
R=80000(배율:100),
K=2000(A=8000), SV=10,
V=4000, A0=0
WR3/D2, 1, 0:1, 0, 0 자동 감속 Mode
가속도 증가율= 3.13Mpps/sec²
기동속도= 1000pps
Drive 속도= 400kpps

◎40kpps 비대칭 S자 가감속(1)



R=800000(배율:10),
K=500, L=2000(A=D=8000),
SV=10, V=3000, A0=0
WR3/D2, 1, 0:1, 1, 1 메뉴얼 감속 Mode
가속도 증가율= 1.25Mpps/sec²
감속도 증가율= 0.31Mpps/sec²
기동속도= 100pps
drive 속도= 30kpps

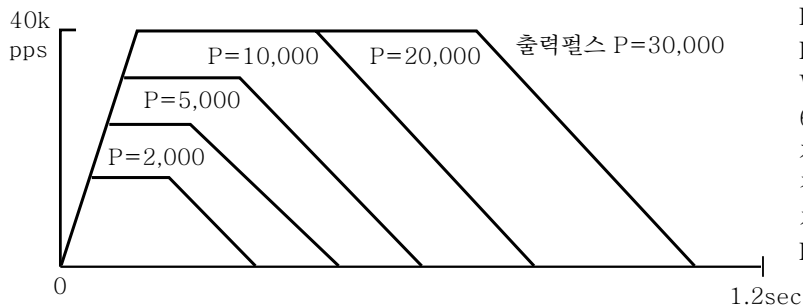
◎40kpps 비대칭 S자 가감속(2)



R=800000(배율:10),
K=2000, L=500(A=D=8000),
SV=10, V=3000, A0=0
WR3/D2, 1, 0:1, 1,1 메뉴얼 감속 mode
가속도 증가율= 0.31Mpps/sec^2
감속도 증가율= 1.25Mpps/sec^2
기동속도= 100pps
Drive 속도= 30kpps

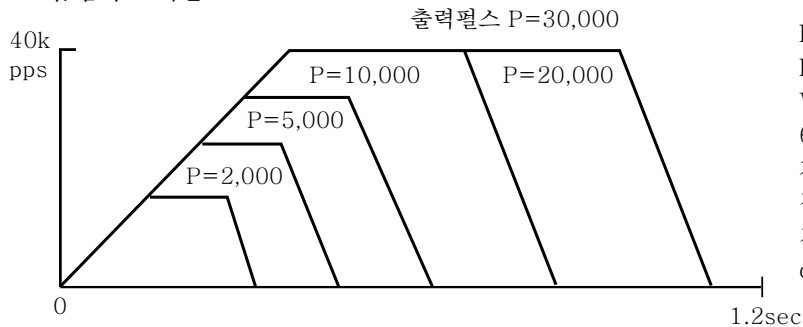
◎40kpps 비대칭 사다리꼴 가감속

a. 가/감속도 비율 4 : 1



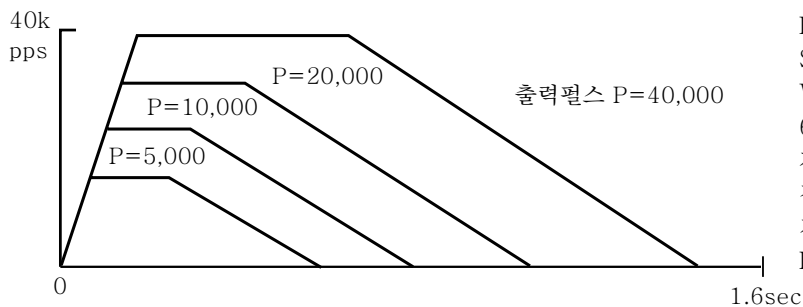
R=800000(배율:10), A=400,
D=100, SV=40, V=4000, A0=0
WR3/D2, 1, 0:0, 1, 0 자동 감속 Mode
60H/WR6/D3:1 삼각 방지 ON
가속도=500kpps/sec
감속도=125kpps/sec
기동속도=400pps
Drive 속도=40kpps

b. 가/감속도 비율 1 : 4



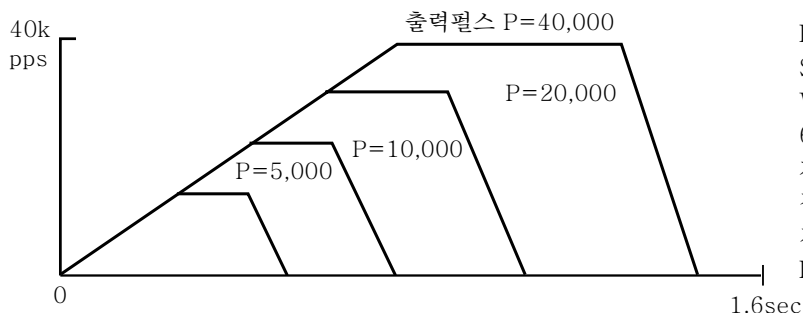
R=800000(배율:10), A=100,
D=400, SV=40, V=4000, A0=0
WR3/D2, 1, 0:0, 1, 0 자동 감속 Mode
60H/WR6/D3:1 삼각 방지 ON
가속도= 125kpps/sec
감속도= 500kpps/sec
기동속도= 400pps
drive 속도= 40kpps

c. 가/감속도 비율 10 : 1



R=800000(배율:10), A=400, D=40,
SV=50, V=4000, A0=0
WR3/D2, 1, 0:0, 1, 0 자동 감속 Mode
60H/WR6/D3:1 삼각 방지 ON
가속도= 500kpps/sec
감속도= 50kpps/sec
기동속도= 500pps
Drive 속도= 40kpps

d. 가/감속도 비율 1 : 10



R=800000(배율:10), A=40, D=400,
SV=50, V=4000, A0=0
WR3/D2, 1, 0:0, 1, 0 자동 감속 Mode
60H/WR6/D3:1 삼각 방지 ON
가속도= 50kpps/sec
감속도= 500kpps/sec
기동속도= 500pps
Drive 속도= 40kpps

Autonics

www.autonics.com

■ 공장자동화의 만족스런 파트너-오토닉스가 함께합니다.

■ 본사(공장)

경남 양산시 웅상읍 용당리 41-5번지

TEL : (055)371-5051 FAX : (055)372-4432

■ 서울사무소 : 경기도 부천시 원미구 약대동 193번지 부천테크노파크 402동 3층

TEL : (032)328-2669 FAX : (032)323-3008

■ 광주사무소: 광주광역시 북구 운암동 1640번지

TEL : (062)521-6716/7 FAX : (062)521-6717

■ 대구사무소 : 대구광역시 동구 신천3동 283-9

TEL : (053)741-7673/4 FAX : (053)741-7674

■ A/S 080 수신자 부담 서비스 안내

080-519-3333 (서비스지역 : 부산, 울산, 경남, 대구, 경북, 광주, 전남, 전북, 제주)

080-529-3333 (서비스지역 : 서울, 인천, 경기, 대전, 충남, 충북, 강원도)

■ MAIN PRODUCTS

- | | | | | | |
|---------|------------------------|---------|--|---------------|------------|
| ■ 카운터 | ■ 타이머 | ■ 온도조절기 | ■ 판넬메타 | ■ 타코/스피드/펄스메타 | ■ 디스플레이 유닛 |
| ■ 근접센서 | ■ 포토센서 | ■ 압력센서 | ■ 광화이버 센서 | ■ 로타리 엔코더 | ■ 센서 컨트롤러 |
| ■ 전력조정기 | ■ 스테핑 모터 & 드라이버 & 컨트롤러 | | ■ 레이저 마킹 시스템(CO ₂ , Nd:YAG) | | |