

## ASIA

**Delta Electronics, Inc.**  
Taoyuan1  
31-1, Xingbang Road, Guishan Industrial Zone,  
Taoyuan County 33370, Taiwan, R.O.C.  
TEL: 886-3-362-6301 / FAX: 886-3-362-7267

**Delta Electronics (Jiang Su) Ltd.**  
Wujiang Plant3  
1688 Jiangxing East Road,  
Wujiang Economy Development Zone,  
Wujiang City, Jiang Su Province,  
People's Republic of China (Post code: 215200)  
TEL: 86-512-6340-3008 / FAX: 86-769-6340-7290

**Delta Electronics (Japan), Inc.**  
Tokyo Office  
Delta Shibadaimon Building, 2-1-14 Shibadaimon,  
Minato-Ku, Tokyo, 105-0012, Japan  
TEL: 81-3-5733-1111 / FAX: 81-3-5733-1211

**Delta Electronics (Korea), Inc.**  
Donghwa B/D 3F, 235-6, Nonhyun-dong,  
Kangnam-gu, Seoul 135-010, Korea  
TEL: 82-2-515-5303/5 / FAX: 82-2-515-5302

**Delta Electronics (Singapore) Pte. Ltd.**  
8 Kaki Bukit Road 2, #04-18 Ruby Warehouse Complex,  
Singapore 417841  
TEL: 65-747-5155 / FAX: 65-744-9228

**Delta Energy Systems (India) Pvt. Ltd.**  
Plot No. 27 & 31, Sector-34, EHTP,  
Gurgaon-122001 Haryana, India  
TEL: 91-124-4169040 / FAX: 91-124-4036045

## AMERICA

**Delta Products Corporation (USA)**  
Raleigh Office  
P.O. Box 12173, 5101 Davis Drive,  
Research Triangle Park, NC 27709, U.S.A.  
TEL: 1-919-767-3813 / FAX: 1-919-767-3939

## EUROPE

**Deltronics (The Netherlands) B.V.**  
Eindhoven Office  
De Witbogt 15, 5652 AG Eindhoven, The Netherlands  
TEL: 31-40-2592850 / FAX: 31-40-2592851

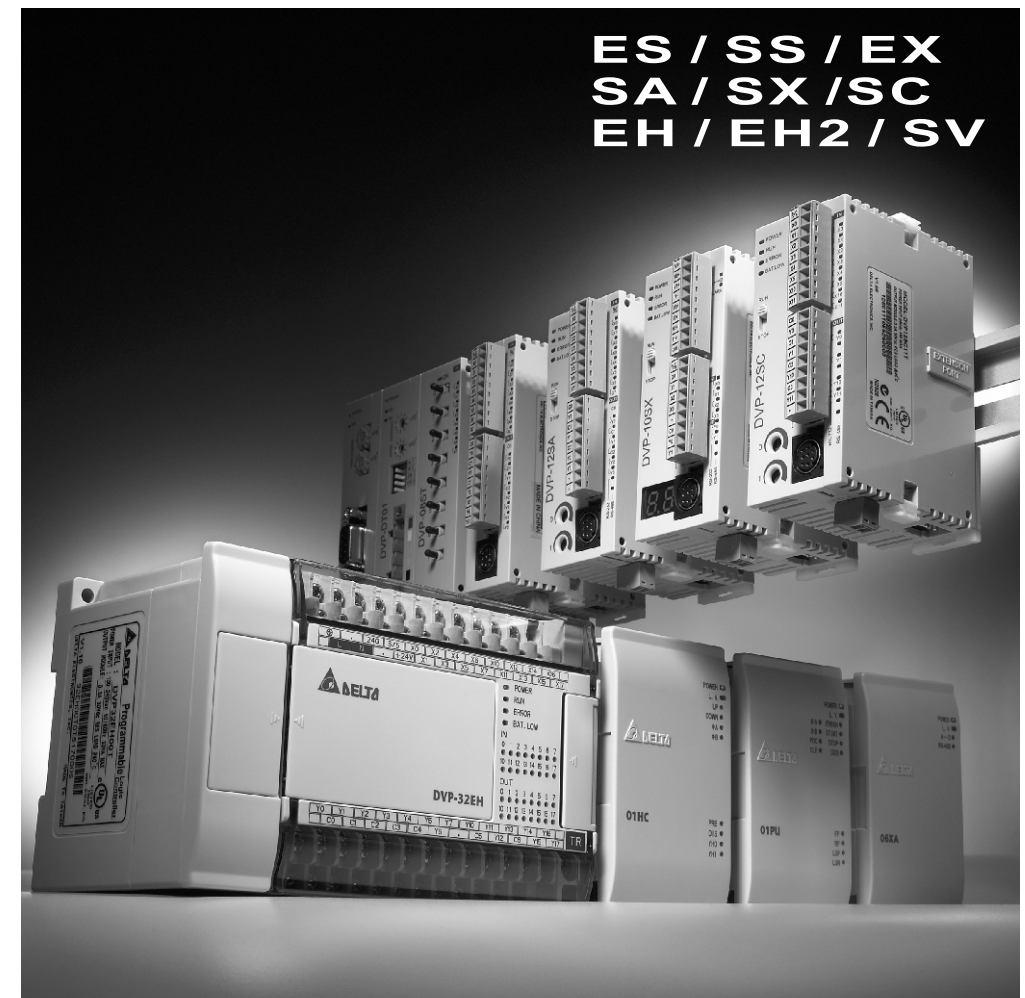
\*We reserve the right to change the information in this manual without prior notice

# DVP-PLC

## 어플리케이션 메뉴얼

### 프로그래밍

### Revision II



# DVP-PLC 응용 매뉴얼

## 【프로그래밍】

### 목 차

#### Chapter 1 PLC Ladder 다이어그램의 작동원리

머리말- PLC 의 배경과 기능 .....	1-1
1.1 Ladder 다이어그램의 작동 원리 .....	1-1
1.2 기존 Ladder 다이어그램과 PLC Ladder 다이어그램과의 차이 .....	1-2
1.3 Ladder 다이어그램의 편집 설명 .....	1-4
1.4 PLC Ladder 다이어그램의 편집 .....	1-8
1.5 PLC 명령의 변환과 각 다이어그램의 구조 .....	1-12
1.6 Ladder 다이어그램의 단순화 .....	1-14
1.7 기본 프로그램 설계 예제 .....	1-17

#### Chapter 2 DVP-PLC 기능

2.1 DVP-PLC 장치 번호 요약 .....	2-1
2.2 값, 상수 [K] / [H] .....	2-7
2.3 외부 입/출력 접점의 기능과 넘버링 [X] / [Y] .....	2-9
2.4 보조 릴레이의 기능과 넘버링 [M] .....	2-11
2.5 스텝릴레이의 기능과 넘버링 [S] .....	2-12
2.6 타이머의 기능과 넘버링 [T] .....	2-13
2.7 카운터의 기능과 넘버링 [C] .....	2-16
2.8 레지스터 번호와 기능 [D], [E], [F] .....	2-30
2.8.1 데이터 레지스터 [D] .....	2-30
2.8.2 인덱스 레지스터 [E], [F] .....	2-31
2.8.3 파일 레지스터 기능과 특성 .....	2-32
2.9 함유 레벨 포인터[N], 포인터[P], 인터럽트 포인터 [I] .....	2-32
2.10 특수 보조 릴레이와 특수 레지스터 .....	2-36
2.11 특수 보조 릴레이와 특수 레지스터 기능 .....	2-61
2.12 오류 코드 정보 .....	2-106

#### Chapter 3 기본 명령

3.1 기본 명령어와 단계별 명령어 요약 .....	3-1
------------------------------	-----

3.2 기본 명령 설명 .....	3-3
--------------------	-----

## Chapter 4 스텝 Ladder 명령

4.1 스텝 Ladder 명령 [STL], [RET] .....	4-1
4.2 연속 기능 도표 (SFC) .....	4-2
4.3 스텝 Ladder 명령 설명 .....	4-3
4.4 스텝 Ladder 프로그램 설계의 나머지 .....	4-8
4.5 절차의 분류 .....	4-9
4.6 IST 명령 .....	4-18

## Chapter 5 응용 명령

5.1 파라미터 정리 .....	5-1
5.2 응용 명령 구조 .....	5-7
5.3 숫자 값의 처리 .....	5-12
5.4 순서 기록 E, F .....	5-15
5.5 명령의 순서 .....	5-17

## Chapter 6 응용 명령 API 00-49

• (API00~09) 루프 제어 .....	6-1
• (API10~19) 전송 비교 .....	6-18
• (API20~29) 4 가지 기본 산술 연산 .....	6-33
• (API30~39) 회전과 변위 .....	6-49
• (API40~49) 데이터 조작 .....	6-60

## Chapter 7 응용 명령 API 50-99

• (API50~59) 고속 프로세싱 .....	7-1
• (API60~69) 간이 명령 .....	7-37
• (API70~79) 외부 I/O 디스플레이 .....	7-56
• (API80~88) 직렬 I/O .....	7-75

## Chapter 8 응용 명령 API 100-149

• (API100~109) Delta AC 모터 드라이브의 통신 명령 .....	8-1
• (API110~119) 부동 연산 .....	8-20
• (API120~129) 부동 연산 .....	8-28
• (API130~139) 삼각 연산 .....	8-38

- (API140~149) 새로운 부가 명령 ..... 8-50

## **Chapter 9 응용 명령 API 150-199**

- (API150~154) 새로운 부가 명령 ..... 9-1
- (API155~159) 위치 제어 ..... 9-14
- (API160~169) 실시간 클록 ..... 9-41
- (API170~171) 그레이 코드 변환 ..... 9-50
- (API180~190) 행렬 처리 ..... 9-52
- (API196~199) 고속 카운터 명령 ..... 9-69

## **Chapter 10 응용 명령 API 215-246**

- (API215~223) 접점 타입 논리 연산 명령 ..... 10-1
- (API224~246) 접점 타입 비교 명령 ..... 10-4

## 머리말- PLC 의 배경과 기능

PLC(프로그램으로 짤 수 있는 논리 제어기)는 하나의 전기 장비입니다. 이 제품은 전에 “시퀀스 제어기로” 불렸습니다. 1978 년 NEMA(국가 전기제품 생산 조합)에 의해 PLC(프로그램으로 짤 수 있는 논리 제어기)로 명명되었으며 전기 장비로 정의 되었다. PLC 의 동작은 아래와 같습니다.

스텝 1. 키패드, 스위치 그리고 펄스와 같은 외부 입력 신호를 읽어드립니다.

스텝 2. 마이크로프로세서를 사용 논리, 시퀀스, 타이머 카운터 그리고 스텝 1 에서 읽혀진 입력 신호 값과 동작상태에 따른 공식의 계산을 실행하고 relay 의 open 또는 close, 제어기기의 동작 혹은 자동기기를 제어하기 위한 동작 또는 제조 작동과 같은 유사한 외부신호를 얻기 위해 프로그램들을 내부에 pre-write 해 저장합니다.

또한 PLC 는 주변장치(PC/ 핸드헬드 프로그래밍 패널)들을 특화 혹은 편집하여 생산 프로그램을 보존하거나 조절하는데 사용될 수 있습니다. PLC 의 대표적인 프로그램 언어는 ladder 다이어그램 입니다.

PLC 내부엔 개발 및 전기 기술의 어플리케이션 요구사항에 관한 position 제어, 네트워크 그리고 기타와 같은 보다 강력한 기능들이 있습니다. 출력/입력 신호들은 DI(디지털 입력), AI(아날로그 입력), PI(펄스 입력), DO(디지털 출력), AO(아날로그 출력) 그리고 PO(펄스 출력)입니다. 그러므로 PLC feature 산업에서 매우 중요한 역할을 담당합니다.

### 1.1 Ladder 다이어그램의 작동 원리

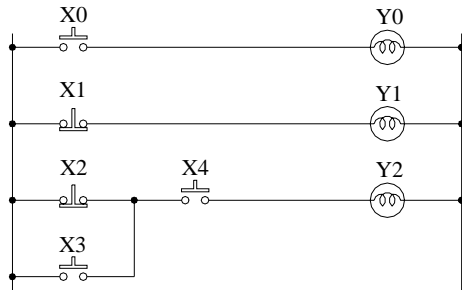
Ladder 다이어그램은 세계 2 차 대전 중 개발된 자동 제어 다이어그램 언어의 한 종류입니다. 처음에 이 언어는 단순히 A 점점(일반적으로 open), B 점점(일반적으로 close), 출력 코일, 타이머, 카운터 그리고 기타(파워 패널은 이 기초 부품들로 구성된다)와 같은 기초 부품들만 가지고 있었습니다. 다른 점점, 래치 코일(latched coil) 그리고 어플리케이션 명령, 더하기, 빼기, 곱하기 그리고 나누기 계산과 같은 전통적인 파워 패널이 PLC 가 개발 되기 전까지 구현하지 못한 좀더 많은 기능들을 가지고 있습니다.

전통적 Ladder 다이어그램과 PLC Ladder 다이어그램의 작동원리는 서로 매우 비슷합니다. 딱 하나 차이점은 전통적 Ladder 다이어그램을 위한 심벌은 그것의 오리지널 본체에 근접한 형식을 표현했고, PLC Ladder 다이어그램은 데이터 시트와 컴퓨터에서 사용될 때 보다 명확한 심벌들을 사용 하였다. Ladder 다이어그램 논리는 조합 논리와 연속 논리들로 나누어 질 수 있고 아래와 같이 묘사 됩니다.

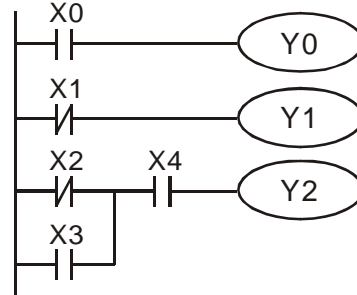
#### 1. 조합 논리(Combination logic)

아래와 같은 예는 전통적 Ladder 다이어그램 과 PLC Ladder 다이어그램의 조합 논리(Combination logic) 분리하여 보여 줍니다.

전통적 Ladder 다이어그램



PLC Ladder 다이어그램



설명 1: 회로 1 번은 일반적으로 “A”스위치 혹은 접점이라 알려지고 평상시엔 OFF 상태(눌러지지 않았을 때)여서 Y0가 OFF 상태인 한 X0 (NO: 일반적으로 오픈(일반적으로 열림)) 스위치를 사용합니다. 그러나, 스위치 동작이 행해졌을 때(버튼이 눌러졌을 때) 그 접점은 ON 이 되고 출력 포인트인 Y0 도 ON 상태가 됩니다.

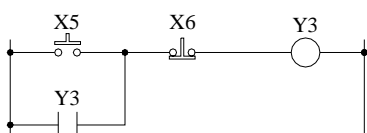
설명 2: 비슷하게, 회로 2 번은 일반적으로 “B”스위치 혹은 접점이라 알려지고 평상시엔 ON 상태여서 Y1 이 ON 상태인 한 X1 (NC: 일반적으로 닫힘(일반적으로 닫힘)) 스위치를 사용합니다. 그러나, 스위치 동작이 행해졌을 때(OFF 상태) 출력 포인트인 Y1 은 OFF 상태가 됩니다.

설명 3: 이 조합논리 출력의 예는 하나 이상의 입력 장비를 가지고 있습니다. X2가 OFF 상태 일 때 혹은 X3 그리고 X4가 ON 상태 일 때 출력 포인트 Y2는 ON 상태 일 것입니다.

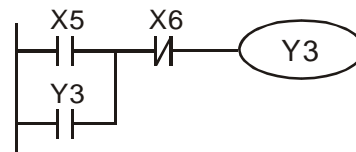
## 2. 연속논리 (Sequential Logics)

연속논리(Sequential Logics)들은 회로의 출력결과를 재송신(Draw back) 하고 그것을 내부 상태처럼 제공하는 “재송신(Draw-Back)” 구조를 가지고 있습니다. 그러므로 동일한 입력 상태에서, 다른 출력 결과물은 이전 상태와 다른 명령의 작동에 따라 생성됩니다. 아래와 같은 연속 논리들(Sequential Logics)들은 전통적 다이어그램과 PLC Ladder 다이어그램의 예를 분리해서 보여줍니다.

전통적 Ladder 다이어그램



PLC Ladder 다이어그램



위 회로들에 전원이 공급 되었을 때, X6 스위치가 ON 이어도 X5 스위치는 여전히 OFF 입니다. 그러므로 출력 릴레이 Y3는 ON 상태입니다. Y3로부터 ON 상태를 포함하는 피드백 신호가 X5의 A 접점에 병렬로 전달된다; 고로 이 회로 또한 자체 래치(self-latched) 회로로 알려져 있습니다. 회로 작동 값은 아래 표와

같이 나타난다.

스텝 \ 장비 상태	X5	X6	Y3
1	N	N	OFF
2	Y	N	ON
3	N	N	ON
4	N	Y	OFF
5	N	N	OFF

N: OFF 상태 Y: ON 상태

위 표로부터, 우리는 같은 입력이라도 다른 결과를 얻을 수 있다는 것을 알 수 있습니다. 예를 들어, 스텝 1 과 3에 있어, X5와 X6 상태는 OFF 상태이지만 스텝 1에서 Y3 값은 OFF 상태이고 스텝 3에선 ON 상태입니다. 이것은 자가 래치 회로 피드백 입력에 기인합니다. 이 예는 A 점점, B 점점과 출력 코일로 설명됩니다. 다른 장비들의 사용법도 이와 같습니다. 자세한 내용은 챕터 3을 참조하십시오..

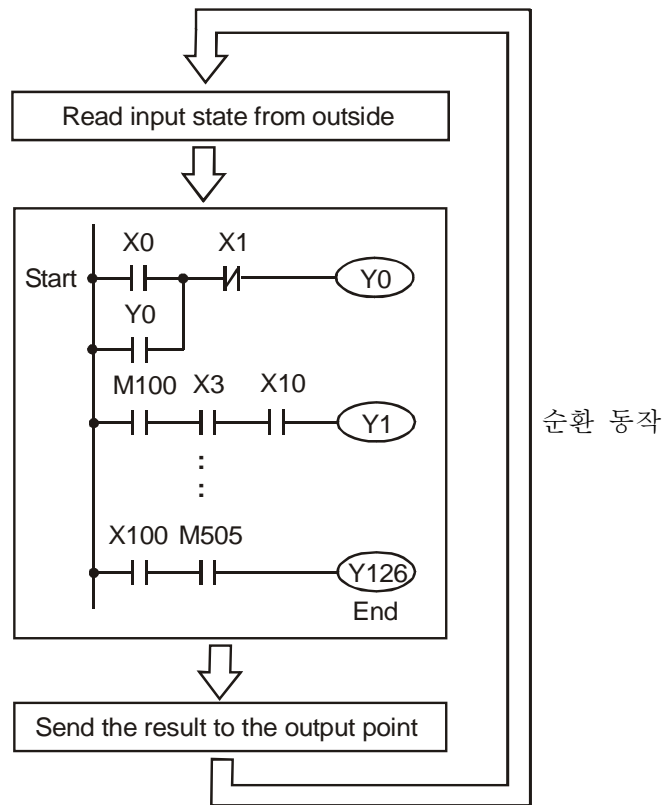
## 1.2 기존 Ladder 다이어그램과 PLC Ladder 다이어그램의 차이

비록 작동 원리들이 전통적 Ladder 다이어그램과 PLC Ladder 다이어그램각자에 따르지만, PLC는 물론 마이크로컴퓨터 칩(MCU)을 사용해 입력 장비들과 출력 코일들의 상태를 하나 하나 검토하는 정밀 조사 방법인 전통적 Ladder 다이어그램 방법을 시뮬레이트 합니다. 그 후에 상태를 고려하여 Ladder 다이어그램의 조합상태의 논리들에 기인한 전통적 Ladder 다이어그램의 결과와 같은 출력을 계산하고 만들어낸다.

그러나, 하나의 MCU만을 사용하게 된 이래로, Ladder 다이어그램 프로그램을 사용해 한 회로가 후에 다른 회로를 검토하는 방법만이 유일한 방법입니다. 프로그램의 수락에 의한 출력 결과물, 입력/출력 상태 그리고 마지막으로 출력 외부 인터페이스 출력을 계산합니다.: 그 후, 입력 상태 정보 판독, 계산, 출력을 다시 시작하고 다시 위에 언급한 동작들을 반복합니다.; 모든 세트의 순환 동작을 완료 하는데 필요한 시간을 한번의 판독 시간(Scan Time)이라 합니다.

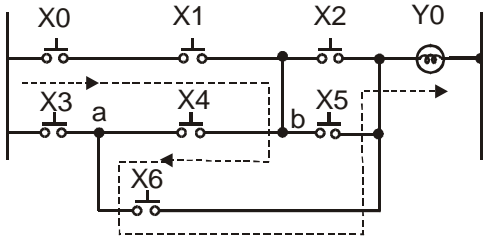
판독 시간(Scan Time)은 프로그램의 증가에 따라 길어지게 됩니다. 이 판독 시간(Scan time)에 있어, 반복적인 입력 탐지에 빠질 수 있습니다. 그러므로 출력 결과물의 반응이 지연되고 지연 시간(Delay time)이 길어지고 가장 치명적인 예러는 제어에 있습니다. 이 상황은 제어 요구를 위해 부적격 합니다.고 할 수 있습니다. 이에 의해, PLC(빠른 판독 시간(Scan Time)을 가진)가 이 역할을 수행하는데 선택 되었다; 고로 판독 속도(Scan speed)는 PLC에 있어 꼭 명기될 필요가 있는 것 입니다. ASIC (IC의 특별 기능들), 마이크로컴퓨터의 진보된 기술에 많은 감사를 하며 현재의 PLC는 판독 속도에 있어 엄청난 발전을 이루었고 PLC Ladder Daigram Program의 판독 차트는 다음과 같습니다.

Ladder 다이어그램  
알고리즘(외부 출력 포인트로  
바로 출력하지 않지만 내부  
장비가 즉각적으로  
출력합니다.)에 의한 계산

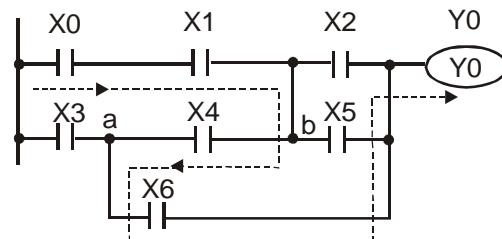


추가적으로 판독 시간(Scan time)의 차이에 대해 말하자면, PLC Ladder 다이어그램 과 전통적 다이어그램은 “역 전류(reverse current)”에 있어 차이점을 가지고 있습니다. 다음과 같은 전통적 Ladder 다이어그램의 차트에서, 만약 X0, X1, X4 와 X6 들은 ON 상태이고 다른 것들은 OFF 상태이면, 출력 포인트인 Y0 는 아래 다이어그램의 점선에서 보여지는 바와 같이 ON 상태 일 것입니다. 그러나 PLC Ladder 다이어그램에 있어선 주변 장치에서 에러가 나타날 것입니다. - MCU 의 판독 방법인 WPL Soft 는 위에서 아래로 그리고 왼쪽에서 오른쪽으로 진행 합니다.

전통적 Ladder 다이어그램의 역 전류(Reverse current)



PLC Ladder 다이어그램의 역전류(Reverse current)



Ladder 다이어그램의 세번째 줄에 오류가 있습니다.

## 1.3 Ladder 다이어그램의 편집 설명

Ladder 다이어그램은 자동 제어와 전기제어 회로의 심볼들로 구성된 다이어그램 언어 중 하나입니다. PLC



절차는 Ladder 다이어그램 에디터가 Ladder 다이어그램을 수정한 후에 끝난다. 다이어그램을 나타내는 제어 흐름을 이해하는 것은 쉬우며 전기 제어 회로 스태프에게 적합합니다.

많은 기초 심볼들과 Ladder 다이어그램 동작들은 버튼, 스위치, 릴레이, 타이머 그리고 기타등등은 전통적 자동 파워 패널의 전기, 기계적 장비들과 같습니다.

PLC 내부 장비의 양과 종류들은 브랜드들에 따라 다르다. 비록 내부 장비는 릴레이, 코일 그리고 접점과 같은 전통적 전기 제어 회로의 이름을 가지고 있지만 PLC 안에 실제로 부품들이 존재 하는 것은 아니다. 내부 메모리의 기본 유닛을 가지고 있을 뿐입니다.

만약 bit 가 1 이면 그것은 코일이 ON 되었음을 의미 하고 bit 가 0 이면 코일이 OFF 되었음을 의미합니다. 당신은 반드시 접점을 사용할 때 bit 의 통신 값을 읽어야 합니다.(보통 OPEN(일반적으로 열림),NO 혹은 접점 A). 다른 경우라면, 당신은 접점을 사용할 때 bit 의 통신 값에 대한 반대 값을 (보통 Close(일반적으로 닫힘), NC 혹은 접점 B) 읽어야 합니다.

많은 릴레이들은 많은 bit 들을 필요로 할 것입니다. 8-bit 는 한 Byte 를 구성합니다. 2 Byte 는 한 단어를 구성하고, 두 단어는 이중 단어를 구성합니다. 더하기/빼기 혹은 이동과 같은 계산을 위해 많은 수의 릴레이들을 사용할 때, 당신은 byte, 단어 혹은 이중단어를 사용할 수 있습니다. 더욱이, PLC 에 있어 타이머와 카운터 이 두 장비들은 코일을 가질 뿐만 아니라 시간 계산 값과 시간을 가집니다.

마지막으로, 각 내부 저장 유닛은 고정 저장 유닛을 점유합니다. 이 장비들을 사용할 때 통신 콘텐츠는 bit,byte 혹은 단어로 읽혀진다. PLC 의 내부 장비 기초 설명에 대한 자세한 내용은 챕터 2 를 참조 하십시오..

입력 릴레이	<p>입력 릴레이는 외부 입력 스위치와 연결하고 외부 입력 신호를 받아들이는 터미널 역할을 하는 외부 입력 포인트와 통신하는 내부 메모리의 기본 저장 유닛입니다. 외부 입력 신호는 0 과 1 을 표시 하는 것을 결정합니다. 당신은 HPP 를 통한 강제 ON/OFF 또는 프로그램 디자인에 의해 입력 릴레이의 상태를 바꿀 수 없다. 접점들(접점 A, B)는 무제한 적으로 사용가능 합니다. 만약 입력 신호가 없을 시, 통신 입력 릴레이는 비어 있을 수 있고 다른 기능등과 함께 사용될 수 있습니다.</p> <p>☞ 장비 표시 방법: X0, X1,...X7, X10, X11,...장비 심벌은 X 와 8 진수를 사용합니다. MPU 에는 입력 포인트 수식 표시기와 확장 유닛이 있습니다.</p>
출력 릴레이	<p>외부 릴레이는 외부 로드(load)에 접속하기 위해 사용되는 외부 출력 포인트와 통신하는 내부 메모리의 기초 저장 유닛입니다. 입력 릴레이, 다른 입력 장비 그리고 자기자신과의 접촉에 의해 작동 될 수 있습니다. 그것은 외부 로드(load)와 입력 접점들처럼 무제한으로 사용될 수 있는 다른 접점들과 연결 되기 위한 일반적으로 열림(NO) 접점으로 이용됩니다. 그것은 통신 출력 릴레이를 가지고 있지 않다. 그것은 내부 릴레이로 사용될 수 있습니다.</p> <p>☞ 장비 표시 방법: Y0, Y1,...Y7, Y10, Y11,... 장비 심벌은 Y 와 8 진수를 사용합니다.</p>

	MPU에는 입력 포인트 수식 표시기와 확장 유닛이 있습니다.
내부 릴레이	<p>내부 릴레이는 외부에 직접적으로 연결되지 않습니다. PLC의 보조 릴레이입니다. 전기 제어 회로의 보조 릴레이와 같은 기능을 합니다. 각 보조 릴레이는 통신 기초 유닛을 가지고 있습니다. 입력 릴레이, 출력 릴레이 혹은 다른 내부 기기와 접촉해 작동합니다. 내부 릴레이의 접점들은 제한 없이 사용될 수 있습니다. 내부 보조 릴레이는 직접 출력 불가하며 출력 포인트를 통해 출력해야만 합니다.</p> <p>☞ 장비 표시: M0, M1, ..., M4, M5. 장비 심벌은 M과 십진수를 사용합니다.</p>
STEP 스텝	<p>DVP PLC는 스텝 동작의 제어 프로그램을 위한 입력 방법을 제공합니다. STL 명령의 제어 스텝 S의 변환을 사용해 제어 프로그램을 작성하는 것은 매우 쉽다. 만약 프로그램 안에 스텝 프로그램이 없다면 스텝 포인트 S는 내부 릴레이 M 또는 알람 포인트로 사용 가능합니다.</p> <p>☞ 장비 표시: S0, S1, ..., S1023/ 장비 심벌은 S와 십진수를 사용합니다.</p>
Timer 타이머	<p>타이머는 시간을 제어하기 위해 사용됩니다. 코일, 접점 그리고 타이머 저장공간이 있습니다. 코일이 ON 상태 일 때, 그것의 접점은 목적 시간이 되었을 때 동작(접점 A는 CLOSE, 접점 B는 OPEN)합니다. 타이머의 시간 값은 셋팅에 의해 정해지고 각 타이머는 고유의 정기적 주기를 가집니다. 사용자는 타이머 값을 결정하고 각 타이머는 개별 시간 주기를 가집니다. 코일이 한번 OFF 되면, 접점은 동작하지 않을 것이며(접점 A는 OPEN, 접점 B는 CLOSE) 타이머는 0으로 초기화 될 것입니다.</p> <p>☞ 장비 표시: T0, T1, ..., T255. 장비 심벌은 T와 십진수를 사용합니다. 다른 숫자 범위는 다른 시간 주기와 상응 합니다.</p>
Counter 카운터	<p>카운터는 세기 위해 사용됩니다. 카운터는 사용 전 설정해야만 합니다.(예를 들어 카운터의 펄스) 코일, 접점들과 카운터 저장 유닛이 카운터 안에 위치하며, 코일이 OFF에서 ON으로 변화할 때, 한과동의 펄스를 카운터에 입력하고 카운터는 1을 더합니다. 16-bit, 32bit와 high-speed 카운터가 있습니다.</p> <p>☞ 장비 표시: C0, C1, ..., C255. 장비 심벌은 C와 십진수를 사용합니다.</p>
Data register 데이터 레지스터	<p>PLC는 타이머 값과 카운터 값과 같은 각 명령들을 제어할 때 데이터와 동작을 처리해야 합니다. 데이터 레지스터는 데이터 혹은 파라미터들을 저장하기 위해 사용됩니다. 16-bit 바이너리 숫자, 예를 들어 워드(word),를 각 레지스터에 저장합니다. 이중 워드들을 저장하기 위해 데이터 레지스터의 두 개의 연속된 숫자를 사용합니다.</p> <p>☞ 장비 표시: D0, D1, ..., D9,999. 장비 심벌은 D와 십진수를 사용합니다.</p>
File register 파일 레지스터	<p>파일레지스터는 PLC 레지스터가 데이터와 파라미터를 처리하는데 충분 하지 않을 때 데이터와 파라미터를 저장하기 위해 사용됩니다. 워드(word)와 같은 16-bit 바이너리 숫자를 각 파일 레지스터에 저장할 수 있으며, 이중 워드(double word)를 처리 하기 위해</p>

	<p>파일 레지스터의 두 개의 연속된 숫자를 사용합니다. SA/SX/SC 시리즈를 위한 1600 개의 파일 레지스터와 EH 시리즈들을 위한 10000 개의 파일 레지스터가 있습니다. 파일 레지스터가 실제 장비로 존재하는 것은 아니다, 그러므로 API148 MEMR, API149 MEMW 명령들 혹은 HPP 와 WPLSoft 같은 주변 장치들을 통해 파일 레지스터의 읽고/쓰기를 실행해야 합니다.</p> <p>☞ 장비 표시: K0~K9,999. 장비 심볼은 없으며 십진수를 사용합니다.</p>
<p>Index register</p> <p>인덱스 레지스터</p>	<p>인덱스 레지스터 E 와 F 는 16-bit 데이터 레지스터로 일반 데이터 레지스터와 같습니다. 자유롭게 쓰기 읽기가 가능하며 bit 장치와 상수와 같은 문자 장치를(Character device) 사용 하기 위한 인덱스 표시 기능을 가지고 있습니다.</p> <p>☞ 장비 표시: E0~E7, F0~F7. 장비 심벌을 E, F 그리고 십진수를 사용합니다.</p>

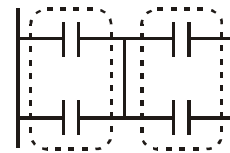
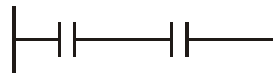
Ladder Diagram 구조와 설명:

Ladder Diagram 구조	설명	명령	장비
	일반적으로 열림, 점점 A	LD	X, Y, M, S, T, C
	일반적으로 닫힘, 점점 B	LDI	X, Y, M, S, T, C
	직렬일반적으로 열림	AND	X, Y, M, S, T, C
	병렬일반적으로 열림	OR	X, Y, M, S, T, C
	병렬 일반적으로 닫힘	ORI	X, Y, M, S, T, C
	상승 엣지 트리거 스위치	LDP	X, Y, M, S, T, C
	하강 엣지 트리거 스위치	LDF	X, Y, M, S, T, C
	직렬 상승 엣지 트리거	ANDP	X, Y, M, S, T, C
	직렬 하강 엣지 트리거	ANDF	X, Y, M, S, T, C
	병렬 상승 엣지 트리거	ORP	X, Y, M, S, T, C
	병렬 하강 엣지 트리거	ORF	X, Y, M, S, T, C
	직렬 블록	ANB	없음
	병렬 블록	ORB	없음

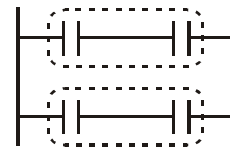
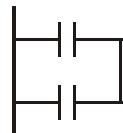
Ladder Diagram 구조	설명	명령	장비
	다중 출력	MPS MRD MPP	없음
	코일 드라이브의 출력 명령	OUT	Y, M, S
	스텝 Ladder	STL	S
	기초 명령, 어플리케이션 명령	Application command	기초 명령은 챕터 3, 어플리케이션 명령은 챕터 5를 참조
	반대 논리(inverse logic )	INV	none

블록: 블록은 직렬 또는 병렬 혹은 위 장비들로 구성된 ladder 다이어그램입니다. 동작 특성에 따라 병렬 블록 또는 직렬 블록을 사용합니다.

직렬 블록

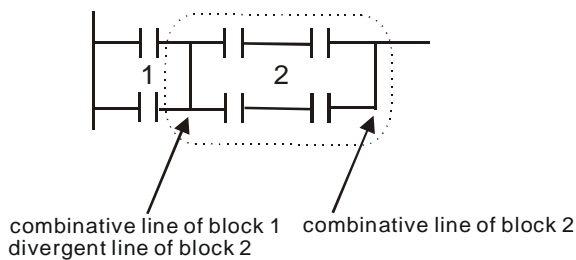


병렬 블록



나누는 선과 조합선: 세로 선은 보통 장치들을 나누는 역할을 합니다. 이 선은 왼쪽장치를 위한 조합 선입니다.

이는 최소 두 개의 기둥 혹은 위 회로 왼쪽은 세로 선에 연결 된다는 뜻입니다. 이 선은 오른쪽 장비에겐 나누는 선입니다. (이는 최소 두 선 혹은 위쪽 회로가 이선과 연결 된다는 뜻입니다.)



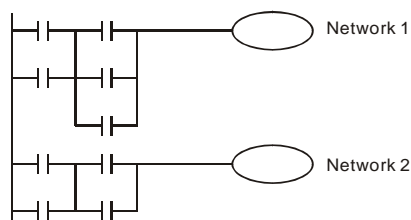
블록 1의 조합선 이자

블록 2의 조합선

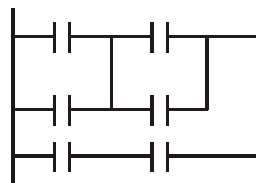
블록 2의 나누는 선

네트워크: 이것은 장치들과 블록들로 구성된 완전한 네트워크 입니다. 선이 연결할 수 있는 세로선 혹은 연속된 선과 블록 또는 장비는 같은 네트워크 입니다.

독립 네트워크:

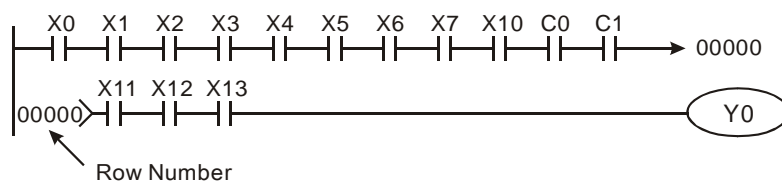


불완전 네트워크:



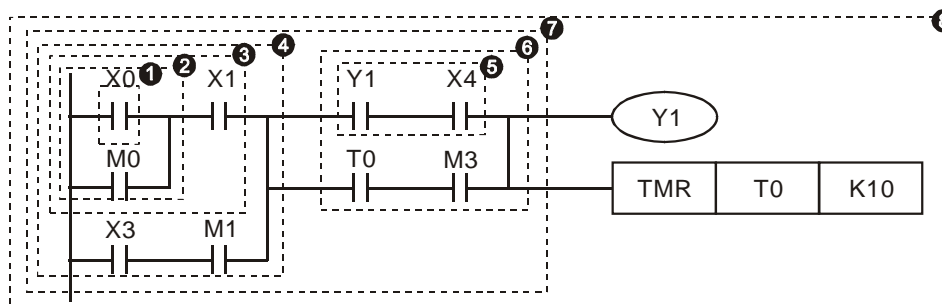
## 1.4 The Edition of PLC Ladder 다이어그램의 편집

프로그램 에디트 방법은 왼쪽 파워 선에서 오른쪽 파워 선으로 합니다. (WPLSoft의 에디트중 오른쪽 파워 선은 생략된다). 줄 에디트 후에는 다음 줄을 에디트 합니다. 한 줄에 가능한 최대 접점은 11 개다. 만약 당신께서 11 개 이상이 필요할 때에는 더 많은 입력 장치들을 계속하기 위해 연속된 선을 시작 하실 수 있습니다. 연속된 숫자들은 자동적으로 그리고 같은 입력 포인트를 만들어 낼 것이며 반복적으로 사용될 것입니다. 아래와 같이 그려집니다.



Ladder 다이어그램의 동작은 왼쪽 윗부분 구석에서부터 오른쪽 아래 구석으로 검색합니다. 출력은 코일 동작 프레임과 어플리케이션 명령을 포함하고, 처리합니다. 보통 Ladder 다이어그램의 오른쪽에 위치 합니다.

아래 예를 참조해; 우리는 진행과정을 순차적으로 분석합니다. 오른쪽 구석에 있는 숫자는 명령을 설명합니다.



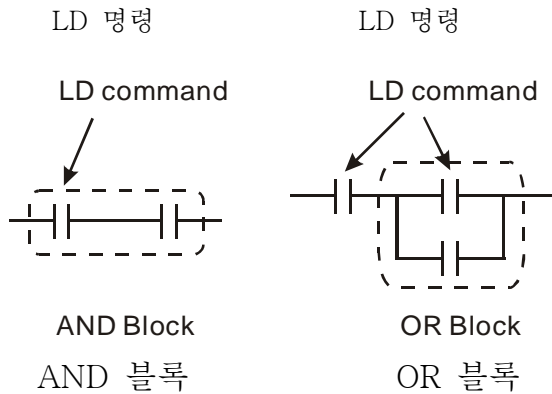
커맨드 명령(Command order) 소개:

1	LD	X0
2	OR	M0
3	AND	X1
4	LD	X3
	AND	M1
	ORB	
5	LD	Y1
	AND	X4
6	LD	T0
	AND	M3

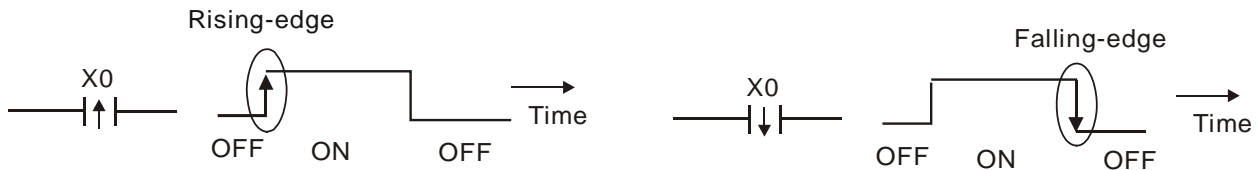
ORB  
7 ANB  
8 OUT Y1  
TMR T0 K10

Ladder 다이어그램의 기초 구조에 대한 자세한 설명

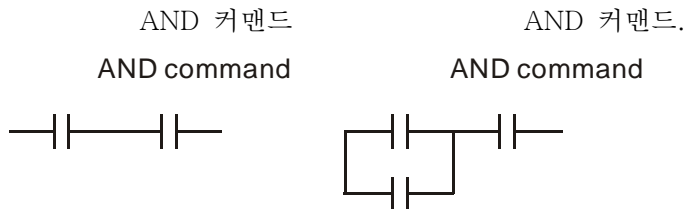
1. LD (LDI) 커맨드: LD 또는 LDI 커맨드는 블록 시작 시 작성합니다.



LDP와 LDF 커맨드는 LD 커맨드와 비슷합니다. 다른 점은 LDP와 LDF 커맨드는 아래와 같이 접점이 ON일 때 상승 엣지 혹은 하강 엣지에서 작동한다는 것입니다.

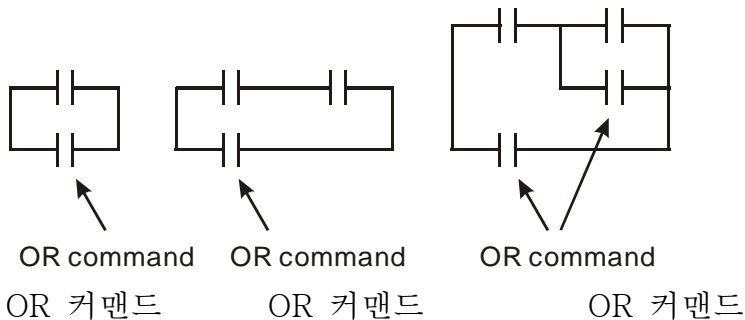


2. AND (ANI) 커맨드: 단일 장치가 연속하여 장치 또는 블록에 연결



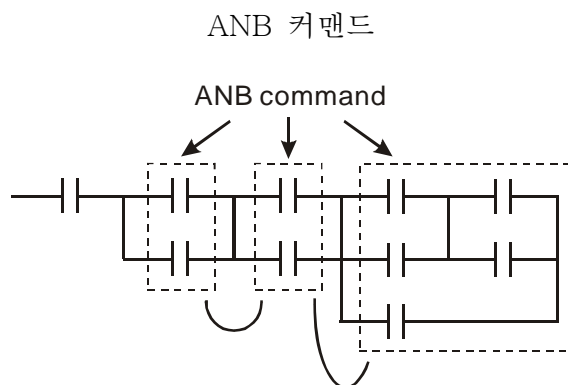
ANDP와 ANDF 구조들은 같다 그러나 상승 엣지 혹은 하강 엣지에서 동작합니다.

3. OR (ORI) 커맨드: 단일 장치가 장치 혹은 단일 블록에 연결.

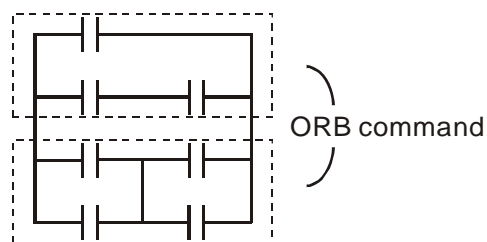


ORP와 ORF 구조들은 같다 그러나 상승 엣지 혹은 하강 엣지에서 동작합니다.

4. ANB 커맨드: 단일 블록이 연속해서 장치 혹은 블록에 연결



5. ORB 커맨드: 단일 블록이 병렬로 장치 또는 블록에 연결



만약 ANB 혹은 ORB 가 작동할 때 몇몇 블록들이 있다면, 그 커맨드들은 위에서 아래로 또는 왼쪽에서 오른쪽으로 블록 또는 네트워크에 결합합니다.

6. MPS, MRD, MPP 커맨드: 다중-출력의 분기된 메모리는 많은 수의 다양한 출력들을 생산할 수 있습니다.

MPS 커맨드는 분기점의 시작입니다. 분기점은 가로선과 세로선이 만나는 곳을 의미합니다.

우리는 같은 세로선 에서 접점 메모리 커맨드를 가지거나 접점 상태에 따르지 않을 것인지를 결정해야만 합니다.

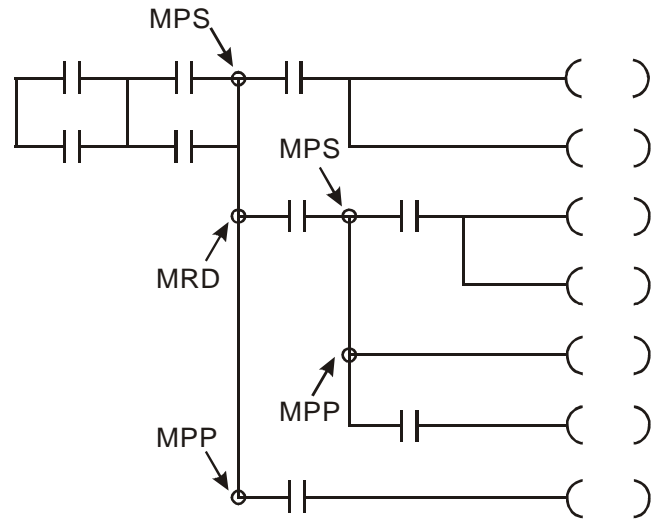
기초적으로, 각 접점은 메모리 커맨드를 가질 수 있다 하지만 Ladder 다이어그램 변환의 몇 곳에서는 PLC 동작을 생략하는 것이 편리하고 수용력에 제한이 있습니다. MPS 커맨드는 8 번 연속해서 사용할 수 있고 당신은 이 커맨드를 심볼 “┐”를 통해 상기 할 수 있습니다.

MRD 커맨드는 분기점의 메모리를 읽어드리는데 사용합니다. 왜냐하면 논리 상태는 가로선과 같기 때문이고 다른 Ladder 다이어그램을 계속 분석하기 위해 원본 접점 상태를 읽어 들여야 하기 때문입니다. 당신은 MRD 커맨드를 심볼 “└”를 통해 상기 할 수 있습니다.

MPP 커맨드는 가장 높은 레벨 상태에서 읽기 시작하고 그것을 스택(일시기억용 장치)으로부터 잠시 빼낸다. 왜냐하면 가로선의 마지막 아이템이기 때문입니다. 이는 가로선이 끝난 상태임을 의미 합니다.

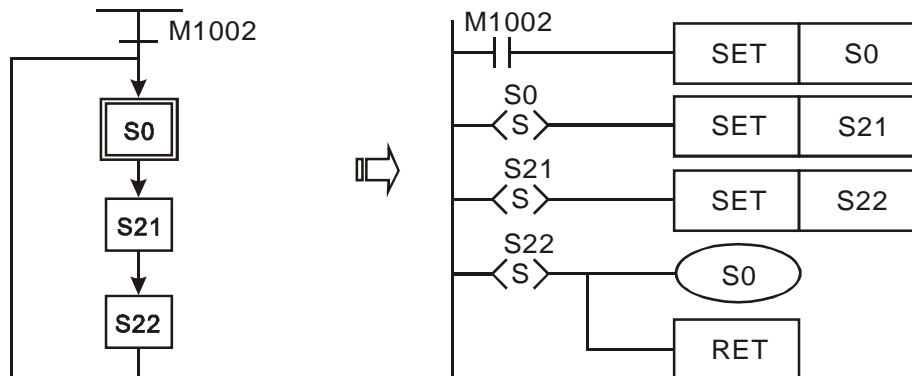
당신은 이 커맨드를 심벌 “L”을 통해 상기할 수 있습니다. 기본적으로, 분석을 위해 위 방법을 사용하는 것은 괜찮다 하지만 가끔 컴파일러가 오른쪽과 같이

같은 출력물을 생략 할 수 있습니다.

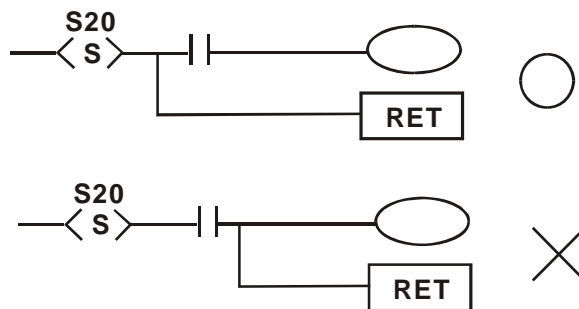


7. STL 커맨드: 이 커맨드는 연속 기능 차트(Sequential Function Chart (SFC))의 구문 설계에 사용됩니다. 이 커맨드는 프로그래머가 프로그램 처리 절차에 있어 현명한 생각을 가질 수 있도록 돕고, 그러므로 처리 절차가 보다 읽기 쉬워진다.

다음과 같은 다이어그램들에서 보이듯이, 우리는 명확한 처리 절차를 얻을 수 있고 원본 스텝 포인트는 각 스텝 포인트 S가 다음 스텝 포인트로 이동된 후에 “파워 손실” 행동을 가질 수 있습니다. 이런 방법으로, 우리는 우리의 처리 절차 다이어그램을 왼쪽 다이어그램에서 아래의 PLC 구조 다이어그램으로 옮길 수 있습니다.



8. RET 커맨드: 당신은 RET 커맨드를 스텝 Ladder 프로그램을 마친 후에 추가 하여야 하고 RET 커맨드는 반드시 다음과 같이 보여지는 STL 커맨드 후에 추가 되어야 합니다.

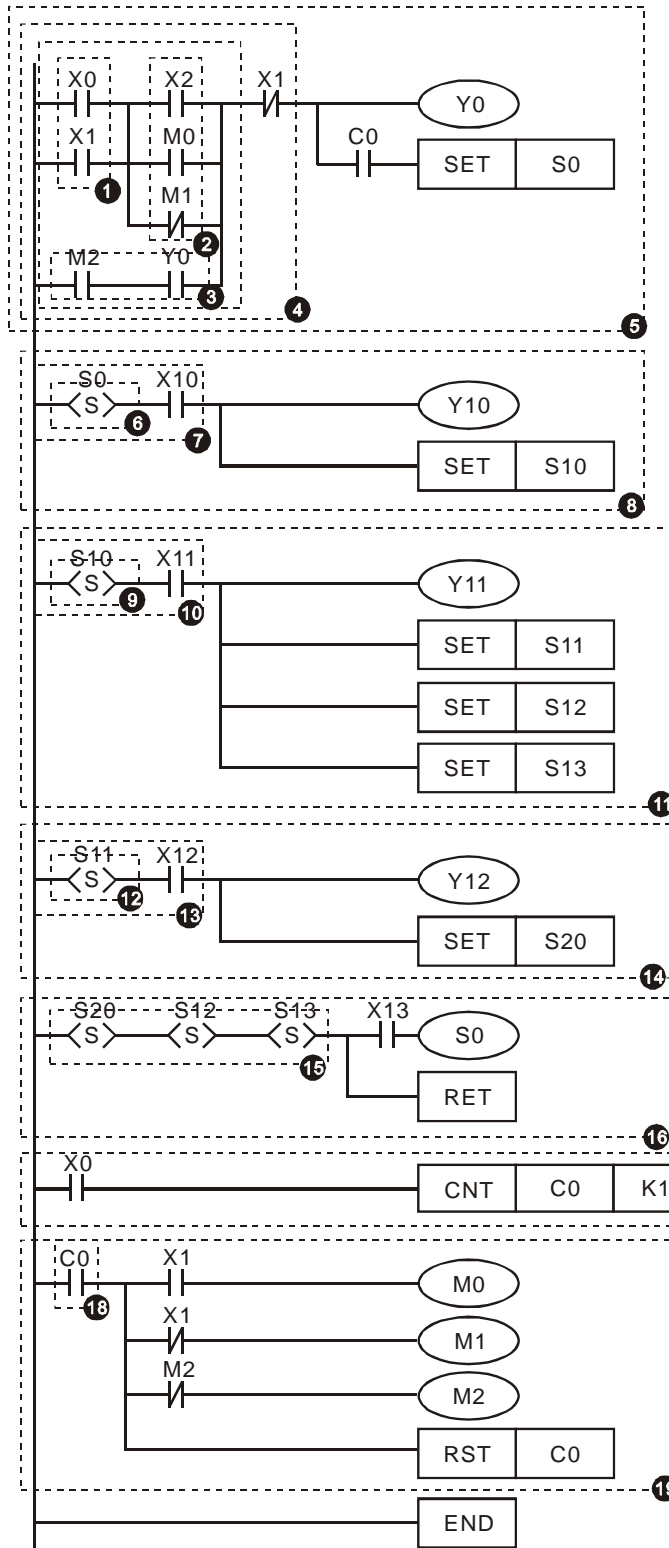


스텝 래더 [SLT], [RET]의 구조에 대해선 챕터 4를 참조 하십시오..



## 1.5 PLC 명령의 변환과 각 다이어그램 구조

Ladder 다이어그램

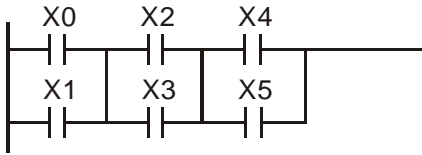


LD	X0	} ① OR block
OR	X1	
LD	X2	} ② OR block
OR	M0	
ORI	M1	} Serial block
ANB		
LD	M2	} ③ AND block
AND	Y0	
ORB		} Parallel block
ANI	X1	
OUT	Y0	} Multiple outputs
AND	C0	
SET	S0	} Step ladder Start ⑥
STL	S0	
LD	X10	} Coperand between S0 and X10
OUT	Y10	
SET	S10	} State working item and step point transfer
STL	S10	
LD	X11	} ⑨ S10 state take out
OUT	Y11	
SET	S11	} Take out X11 state
SET	S12	
SET	S13	} State working item and step point transfer
STL	S11	
LD	X12	} ⑫ S11 state take out
OUT	Y12	
SET	S20	} Take out X12 state
STL	S20	
STL	S12	} State working item and step point transfer
STL	S13	
LD	X13	} Simultaneous divergence
OUT	S0	
RET		} State working item and step point transfer
LD	X0	
CNT	C0 K10	} ⑮ End of step ladder
LD	C0	
MPS		} Read C0
AND	X1	
OUT	M0	} Multiple outputs
MRD		
ANI	X1	} Multiple outputs
OUT	M1	
MPP		} Multiple outputs
ANI	M2	
OUT	M2	} Multiple outputs
RST	C0	
END		Program End

선택스(Syntax) 가변 구조:

Ladder 다이어그램 수정의 분석적 진행은 왼쪽에서 오른쪽으로 또는 위에서 아래쪽으로 해야 합니다. 그러나 다음과 같은 몇 가지 예외가 있습니다.

예 1: 다음과 같이 보여지는 Ladder 다이어그램을 표현하기 위해 커맨드를 사용하기 위한 방법은 두 가지가 있습니다. 그러나 결과는 같습니다.

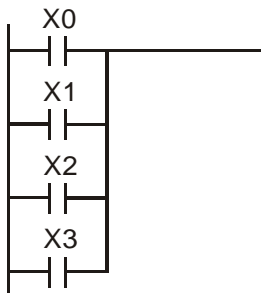


좋은 방법		나쁜 방법	
LD	X0	LD	X0
OR	X1	OR	X1
LD	X2	LD	X2
OR	X3	OR	X3
ANB		LD	X4
LD	X4	OR	X5
OR	X5	ANB	
ANB		ANB	

위의 두 프로그램들을 Ladder 다이어그램으로 변환하면 그 결과는 같습니다. 근데 왜 하나가 다른 것보다 나은 것인가? 그 이유는 MPU 동작에 있습니다. 왼쪽 프로그램의 동작은 다른 하나에게 하나의 블록으로 병합됩니다.

비록 프로그램 길이가 왼쪽과 오른쪽이 같지만 오른쪽 부분 프로그램 동작은 마지막에 병합 됩니다.(커맨드 ANB는 병합을 위해 사용되며 8 번 이상 연속 사용하지 못합니다.). 이 프로그램에선 연속 두 번으로 ANB 커맨드를 사용하면 되고 MPU는 이를 허락합니다. 그러나 프로그램이 8 번 이상 연속해서 ANB 커맨드를 사용할 필요가 있다면 MPU는 이를 허락 하지 않습니다. 그래서 최상의 방법은 한번 블록으로 병합하고 그것을 수립되게 하고 이런 방법으로 프로그래머의 논리는 정리 됩니다.

예 2: 다음과 같이 보여지는 Ladder 다이어그램을 표현하기 위해 커맨드를 사용하기 위한 방법은 두 가지가 있습니다. 그러나 결과는 같습니다.

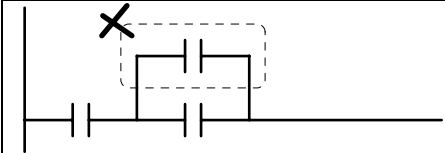
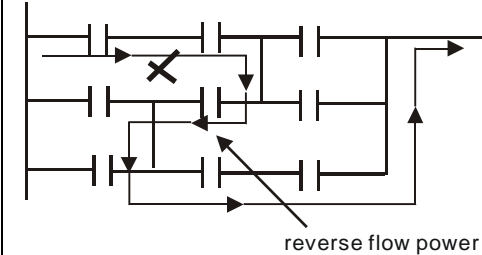
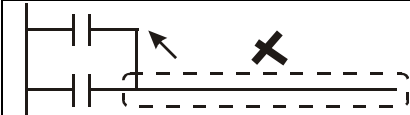
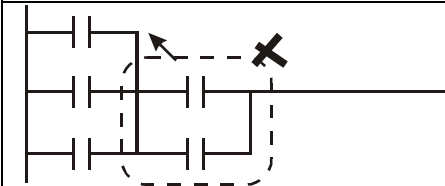
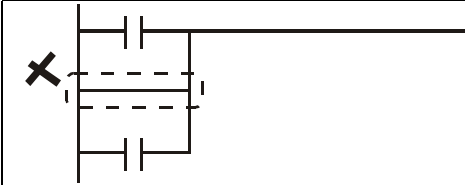
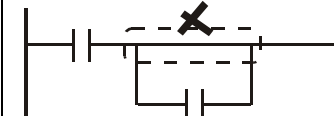
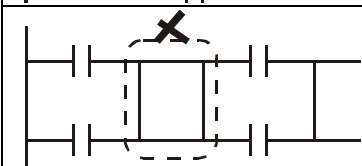
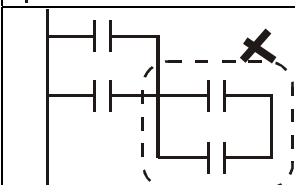
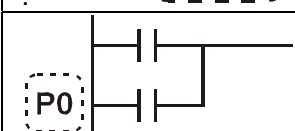
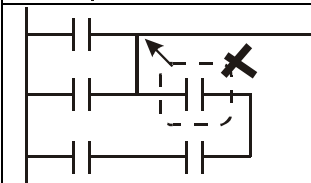


좋은 방법		나쁜 방법	
LD	X0	LD	X0
OR	X1	LD	X1
OR	X2	LD	X2
OR	X3	LD	X3
		ORB	
		ORB	
		ORB	

이 두 프로그램들의 차이점은 매우 분명합니다. 나쁜 방법에선 더 많은 프로그램 코드가 필요하고 또한 MPU 동작 메모리도 늘려야 합니다. 그래서 정의 명령을 해독하는 것이 바람직합니다.

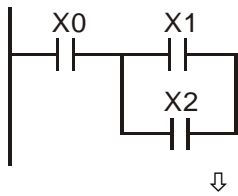
Ladder 다이어그램의 도식 예러

Ladder 다이어그램을 편집할 때, 당신은 모든 종류의 도식을 만드는데 있어 모든 ladder 심벌을 사용할 수 있습니다. Ladder 다이어그램을 그릴 때, 당신은 왼쪽 파워 라인으로부터 시작하여 오른쪽 파워라인으로 끝을 내야 합니다. (오른쪽 파워라인은 WPLSoft Ladder Diagram 을 사용할 때 생략 된다). PLC 원리에 의거 도식 프로그램을 처리하는 것은 위에서 아래로 왼쪽에서 오른쪽으로 해야 합니다.(그것은 왼쪽에서 오른쪽으로 그려지고 한 줄을 끝내고 새로운 다음 줄을 그어야 합니다.) 아래는 일반적인 도식 예러입니다.

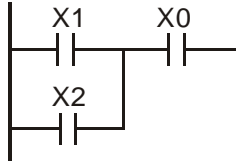
	위쪽 OR 작동 불가
	입력에서 출력 신호가 회로에 있을 때, 반대 파워 흐름이 있음
	오른쪽 위 구석으로부터 출력의 변경
	만약 병합 또는 변경을 원한다면, 명령은 반드시 왼쪽 위 구석으로부터 오른쪽 아래 구석으로 향해야 합니다. 점선 블록이 위로 옮겨져야 함
	빈 장치로 병렬 동작 불가
	빈 장치는 다른 장치들과 동작 불가
	가운데 블록에 장치가 없음
	연속된 장치는 연속으로 연결된 블록들과 병렬로 연결되어야 함
	완벽한 네트워크의 첫번째 줄에 Label P 가 위치해야 함
	연속으로 연결된 블록은 위쪽 가로줄과 병렬로 연결되어야 함

## 1.6 Ladder 다이어그램의 단순화

- Ladder 다이어그램의 앞에 블록을 위치하는 것은 연속된 블록 그리고 병렬 블록들이 연속으로 연결되었을 때 ANB 커맨드를 생략 할 수 있습니다.

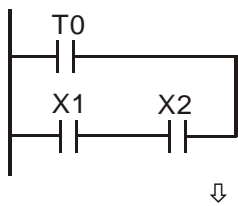


명령  
LD X0  
LD X1  
OR X2  
ANB

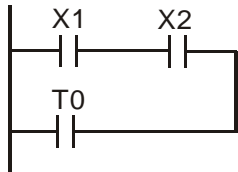


명령  
LD X1  
OR X2  
AND X0

- Ladder 다이어그램의 앞에 블록을 위치하는 것은 단일 장비와 블록을 병렬로 연결할 때 ORB 커맨드를 생략할 수 있습니다.



명령  
LD T0  
LD X1  
AND X2  
ORB



명령  
LD X1  
AND X2  
OR T0

- Ladder 다이어그램의 도식(a) 안에서 파워의 반대 흐름은 불법이 아니다. 도식(a)에서, 위쪽 블록은 낮은 쪽의 블록보다 작다, 당신은 그것들의 바꿔 합법적으로 만들 수 있습니다.

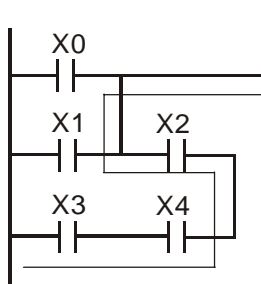


Fig. (a)

명령  
LD X0  
OR X1  
AND X2  
LD X3  
AND X4  
ORB

↓

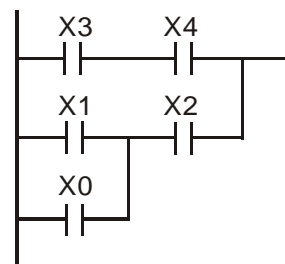
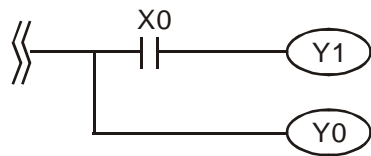


Fig. (b)

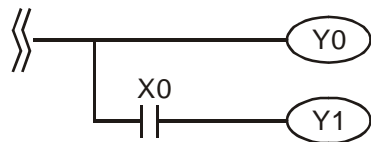
명령  
LD X3  
AND X4  
LD X1  
OR X0  
AND X2  
ORB

- 사용자는 다중 출력들이 다른 입력 장치들과 함께 동작할 필요가 없는 같은 가로 선 상에 있을 때 MPS, MPP 커맨드를 생략할 수 있습니다.



명령  
MPS  
AND X0  
OUT Y1  
MPP  
OUT Y0

↓

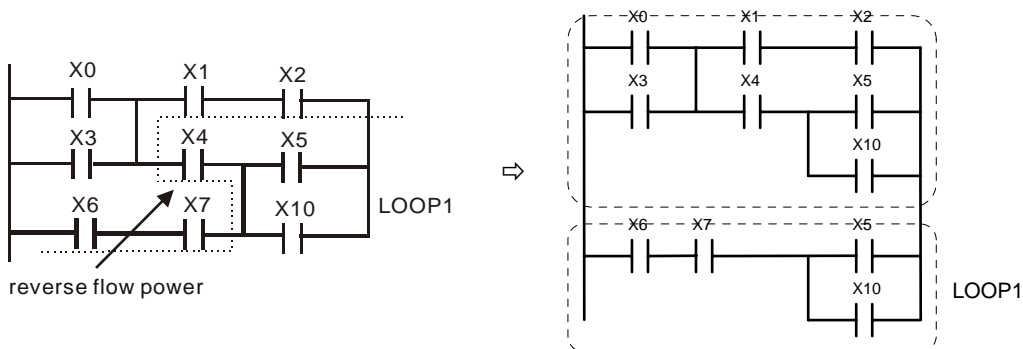


명령  
OUT Y0  
AND X0  
OUT Y1

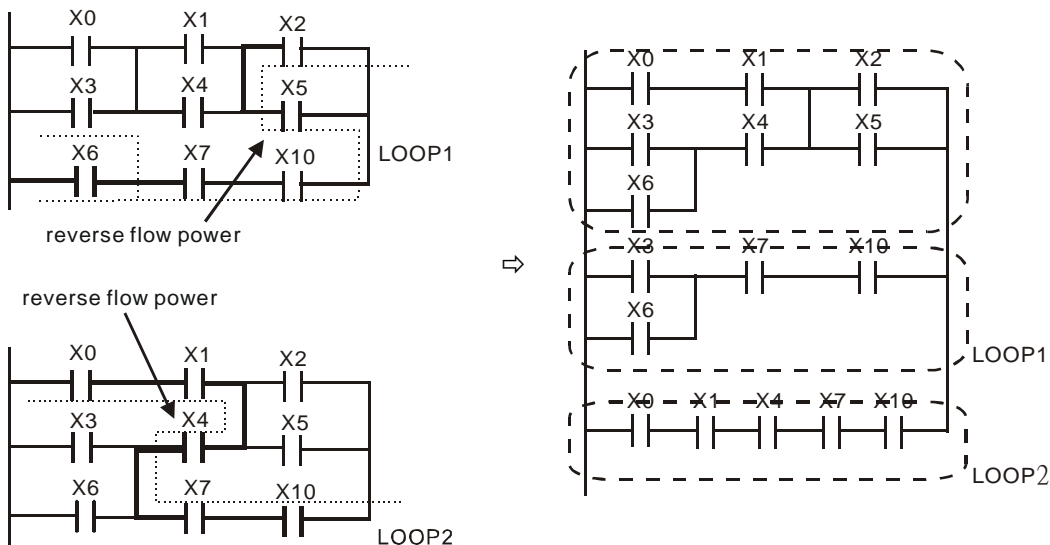
- 파워 반대 흐름의 회로 변경

다음과 같은 예들에서, 왼쪽 도식은 우리의 정의대로 그려진 Ladder 다이어그램입니다. 그러나 파워 반대 흐름이 보입니다. 그러므로 우리는 그것을 변경하고 오른쪽에서 보여줍니다.

예 1:



예 2:



## 1.7 기본 프로그램 설계 예제

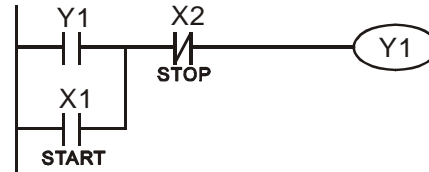
### ■ 시작, 멈춤 그리고 래칭

같은 상황에서, 일시적인 CLOSE 버튼과 OPEN 버튼을 스위치를 시작하고 멈추기 위해 필요로 합니다.

그러므로, 만약 당시 동작이 계속되기를 원합니다.면, 당신은 반드시 래칭 회로를 설계 해야 합니다. 다음과 같은 몇 개의 래칭 회로들이 있습니다.

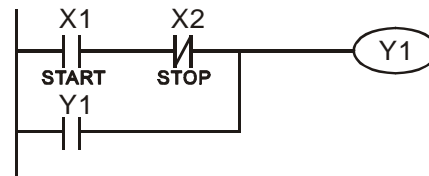
#### 예제 1: 멈춤이 우선하는 래칭 회로

보통 시작 시 일반적으로 열림 contact X1=On 로, 멈춤 시 보통 contact X2=Off 와 Y1=On 로 한번에 맞춰졌을 때, 만약 X2=0 이면 코일 Y1 는 멈춤 동작을 할 것입니다. 그러므로, 멈춤이 우선합니다.고 불립니다.



#### 예제 2: 시작이 우선하는 래칭 회로

보통 시작 시 일반적으로 열림 contact X1=On 로, 멈춤 시 보통 contact X2=Off 와 Y1=On(코일 Y1 은 활성화 될 것이고 래칭 될 것입니다.)로 한번에 맞춰졌을 때, 만약 X2=0 이면 코일 Y1 는 래치된 접점에 기인해 활성화 됩니다. 그러므로, 시작이 우선합니다.고 불립니다.



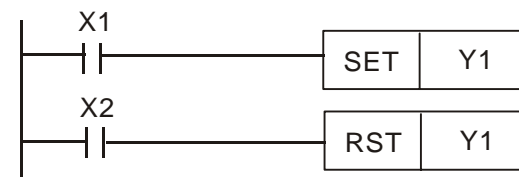
#### 예제 3: SETR 와 RST 커맨드의 래칭 회로

오른쪽의 도식은 RST 와 SET 커맨드로 구성된 래칭 회로입니다.

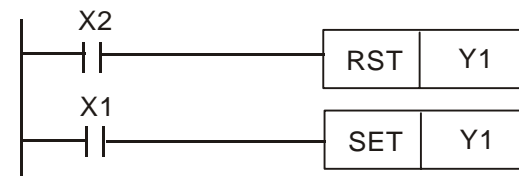
RST 커맨드가 SET 커맨드 뒤에 놓였을 때 가장 우선시되는 건 멈춤입니다. PLC 가 위에서부터 아래로 실행되고 있을 때, 코일 Y1 은 ON 그리고 코일 Y1 은 X1 과 X2 가 동시에 동작할 때 OFF 일 것입니다. 그러므로 그것은 멈춤 우선 이라 불립니다.

SET 커맨드가 RST 커맨드 후에 놓였을 때 가장 우선시되는 건 멈춤입니다. 코일 Y1 은 X1 과 X2 가 동시에 동작할 때 ON 일 것입니다. 그러므로 그것은 시작 우선 이라 불립니다.

#### Top priority of stop

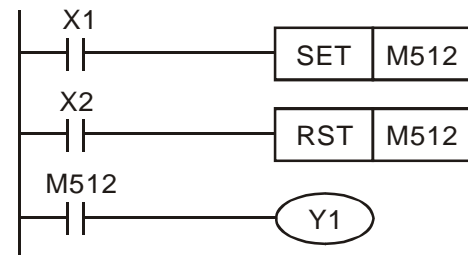


#### Top priority of start



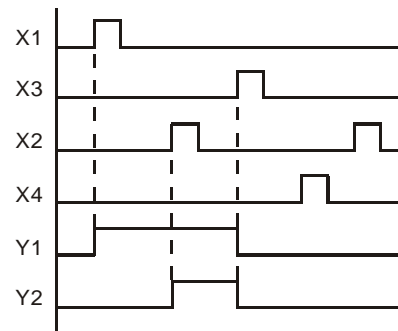
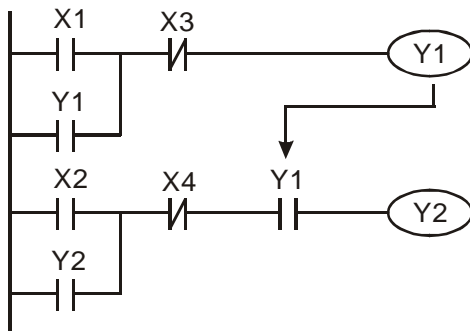
## 예제 4: 래치된

보조 릴레이 M512는 오른쪽에서 래치 되었다(PLC 사용자 매뉴얼 참조) 오른쪽 회로는 파워가 ON 일 때 래치될 것이고 파워 손실과 파워가 다시 ON 되었을 때 래치 될 것입니다. 그러므로 래치되는 것은 연속적입니다.



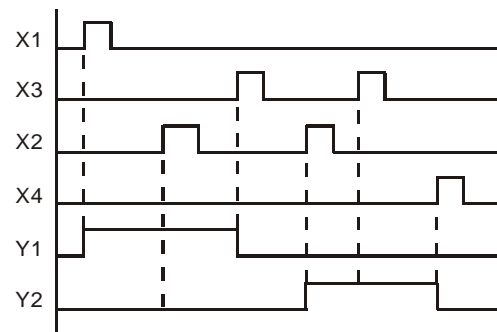
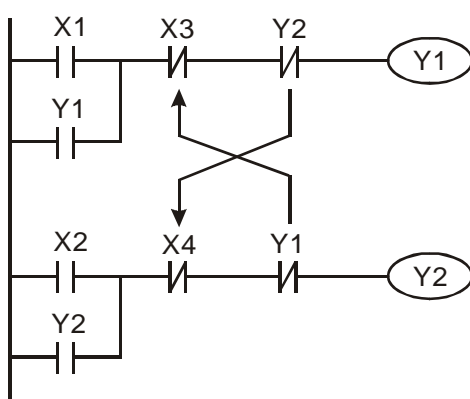
## ■ 일반 제어 회로

### 예제 5: 상태 제어



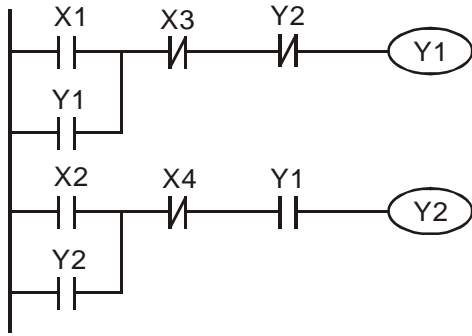
X1과 X3는 Y1 시작/멈춤을 분리하여 할 수 있고, X2와 X4는 Y2 시작/멈춤을 분리하여 할 수 있습니다. 그리고 그것들은 모두 자가 래치 회로입니다. Y1은 Y2가 AND 기능을 일반적으로 열림 접점이 Y2에 연속하여 연결되도록 하는데 있어 구성 요소입니다. 그러므로 Y1은 Y2의 입력이자 Y2는 Y1의 입력입니다.

### Example 6: 연동 제어



위쪽 도식은 연동 제어 회로입니다. Y1과 Y2는 접점 X1과 X2가 시작되는 것에 따라 동작 합니다. Y1과 Y2는 동시에 작동하지 않습니다. 하나가 동작하면 다른 하나는 동작 하지 않습니다.(이것을 연동이라 부른다). X1과 X2가 동시에 동작 하더라도 Y1과 Y2는 Ladder 다이어그램의 위에서 아래로 하는 탐지에 따라 동시에 행동 하지 않습니다. 이 Ladder 다이어그램에서 Y1이 Y2보다 우선합니다.

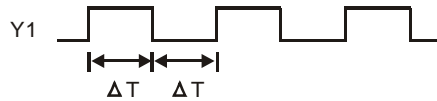
예제 7: 연속 제어



만약 일반적으로 닫힘 contact Y2 를 Y1 회로에 AND 기능을 사용하게끔 Y1 에 대한 하나의 입력으로 추가합니다.면(왼쪽에서 볼 수 있듯이) Y1 은 Y2 의 입력이고 Y2 는 Y1 을 동작 후에 멈출 수 있습니다. Y1 과 Y2 는 연속해서 실행 할 수 있습니다.

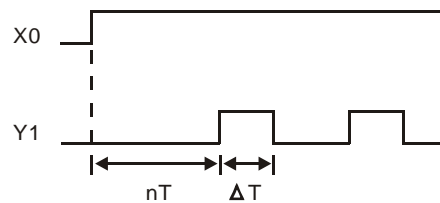
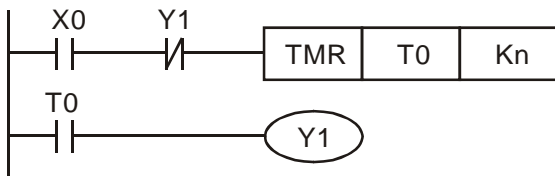
예제 8: 진동 회로

진동회로의 구간은  $\Delta T + \Delta T$  입니다.



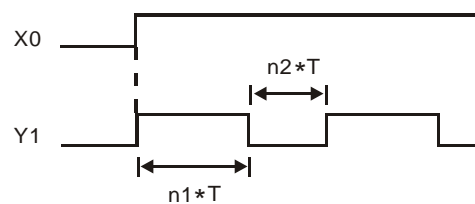
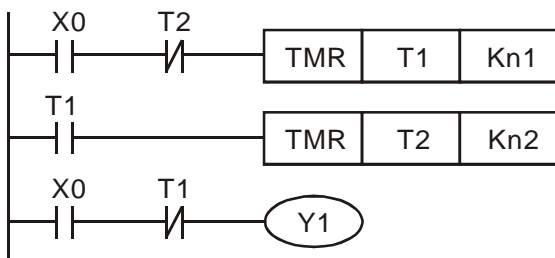
위 도식은 매우 단순한 Ladder 스텝 다이어그램입니다. Y1 일반적으로 닫힘 접점을 점검하기 시작할 때, Y1 일반적으로 닫힘 접점은 코일 Y1 이 OFF 상태여서 CLOSE 됩니다. 그때, Y1 을 점검하고 코일 Y1 은 ON 상태가 될 것이며 1 을 출력합니다. 일반적으로 닫힘 접점 Y1 의 다음 점검 구간에선, Y1 일반적으로 닫힘 접점은 Y1 이 ON 상태 일 때 OPEN 될 것입니다. 마지막으로, 코일 Y1 이 OFF 가 될 것입니다. 반복된 점검의 결과, 코일 Y 는 순환 시간과 진동 펄스를 출력 할 것입니다.  $\Delta T(On) + \Delta T(Off)$ .

순환 시간의 진동 회로  $\Delta T(On) + \Delta T(Off)$ :



위 도식은 타이머 T0 를 코일 Y1 이 ON 상태가 되도록 제어하기 위해 사용합니다. Y1 이 ON 이 된 후에, 타이머 T0 는 다음 점검 기간에 CLOSE 될 것 이며 Y1 을 출력합니다. 진동 회로는 위와 같이 보여진다.(n 은 타이머 셋팅 그리고 십진수 입니다. T 는 타이머의 기초입니다.(시계 구간))

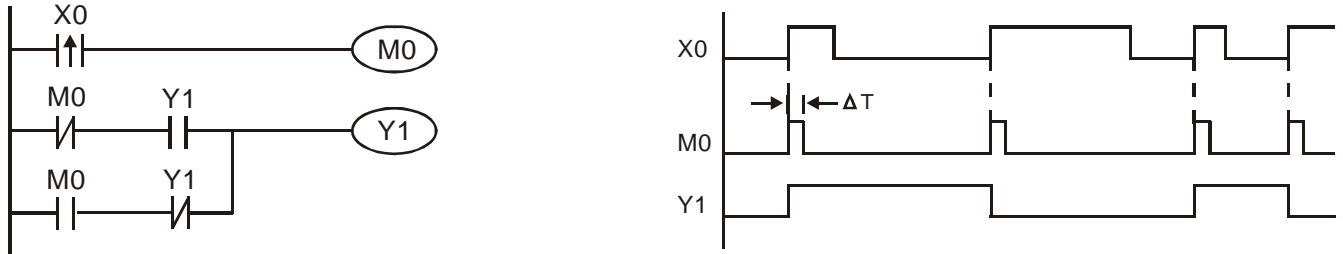
예제 9: 깜박이는 회로 (BLIKING CIRCUIT)





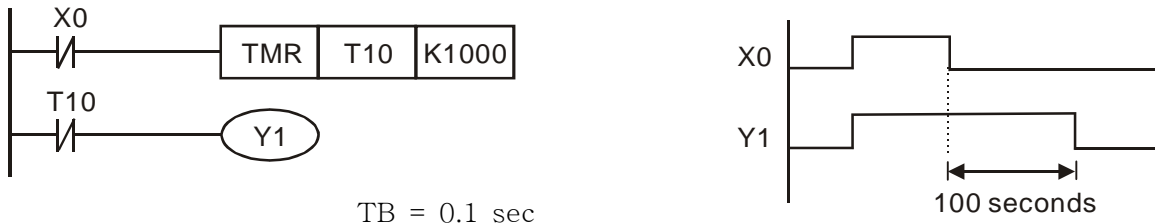
위 도식은 왕복거리는 알람 시계 혹은 빛이 깜박이는 표시기를 위해 일반적으로 사용되는 진동 회로입니다. 두 개의 타이머들은 Y1 코일의 ON/OFF 시간을 조절 하기 위해 사용됩니다. 만약 도식, n1 과 n2 가 T1 과 T2 의 타이머 셋팅 입니다. T는 타이머의 기초입니다.(시계 구간).

예제 10: 트리거가 있는 회로 (Triggered circuit)



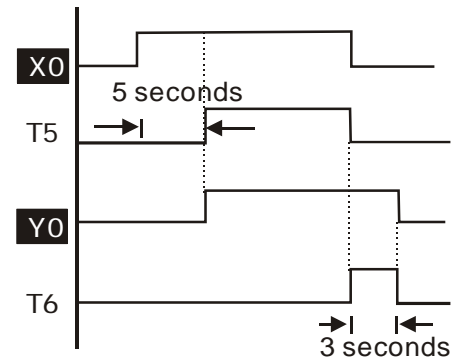
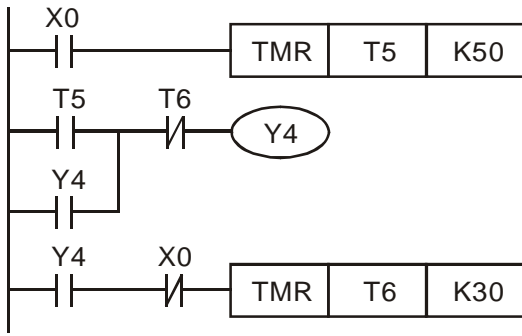
위 도식에서, X0 의 상승 엣지 차이가 있는 커맨드는 코일 M0 가  $\Delta T$ (한 단위의 점점 시간)의 단일 펄스를 가지게 합니다. Y1 은 점점 시간 동안 ON 상태 일 것입니다., 코일 M0 는 OFF 상태 일 것이며 일반적으로 닫힘 M0 와 일반적으로 닫힘 Y1 은 모두 CLOSE 상태 일 것입니다. 그러나 코일 Y1 은 계속 ON 상태 일 것이며 한 점점 시간 동안 입력 X0 과 코일 M0 가 ON 인 후에 상승 엣지가 올 때 코일 Y1 1 회 OFF 됩니다. 타이밍 차트는 위에서 보는 것과 같습니다. 그 회로는 보통 하나의 입력에 두 가지 동작들을 교차해 실행합니다. 위 타이밍에서: 입력 X0 가 T 구간 동안 사각 파동일 때, 출력 코일 Y1 은 2T 구간의 사각 파동입니다.

예제 11: 딜레이 회로(Delay Circuit)

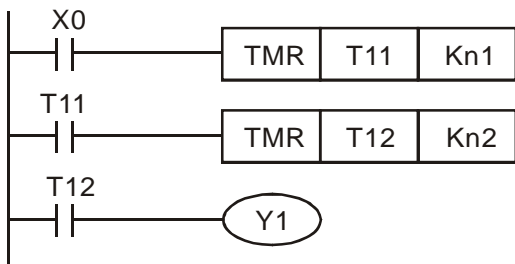


입력 X0 가 ON 일 때, 일반적으로 닫힘 점점 OFF 타이머 T10 이 OFF 되게 하는 것과 통신하는 것에 기안해 출력 코일 Y1 은 같은 시간에 ON 상태 일 것입니다. 출력 코일 Y1 은 100 초 후에 OFF 됩니다. (K1000\*0.1 초 =100 초) 1 회 입력 X0 가 OFF 이고 T10 은 ON 입니다.위 타이밍 차트를 참조 하십시오..

예제 12: 다음 예에서 출력 딜레이 회로는 두 개의 타이머로 구성됩니다. 입력 X0 가 ON 또는 OFF 인 것에 상관없이 Y4 는 늦어질(delay) 것 입니다.

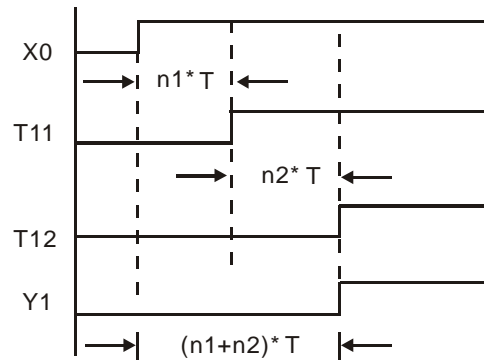


예제 13: 확장 타이머 회로(Extend Timer Circuit)

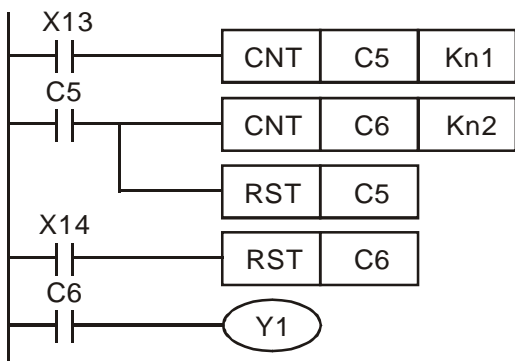


T가 시간 구간 일 때

이 회로에서, 입력 X0는 CLOSE 이고 출력 Y1는 ON 일 때 종합 딜레이 시간은  $(n1+n2)*T$  입니다.

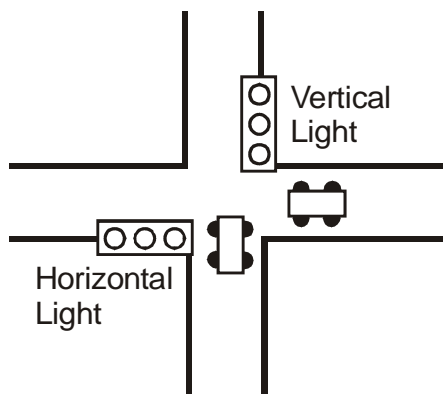


예제 14: 카운터 범위를 확대하는 방법



16-bit 카운터의 범위는 0~32,767 입니다. 만약 두 개의 카운터를 왼쪽 도식과 같이 사용합니다.면 카운터 범위는  $n1*n2$ 로 확대 됩니다. 카운터 C5가  $n1$ 에 도달 했을 때 카운터 C6는 한 타임을 세고 자신을 리셋 할 것입니다. 그리고 카운터 C6는 X13의 펄스를 셀 것입니다. 카운터 C6가  $n2$ 에 도달 했을 때, X13의 펄스는  $n1*n2$ 입니다.

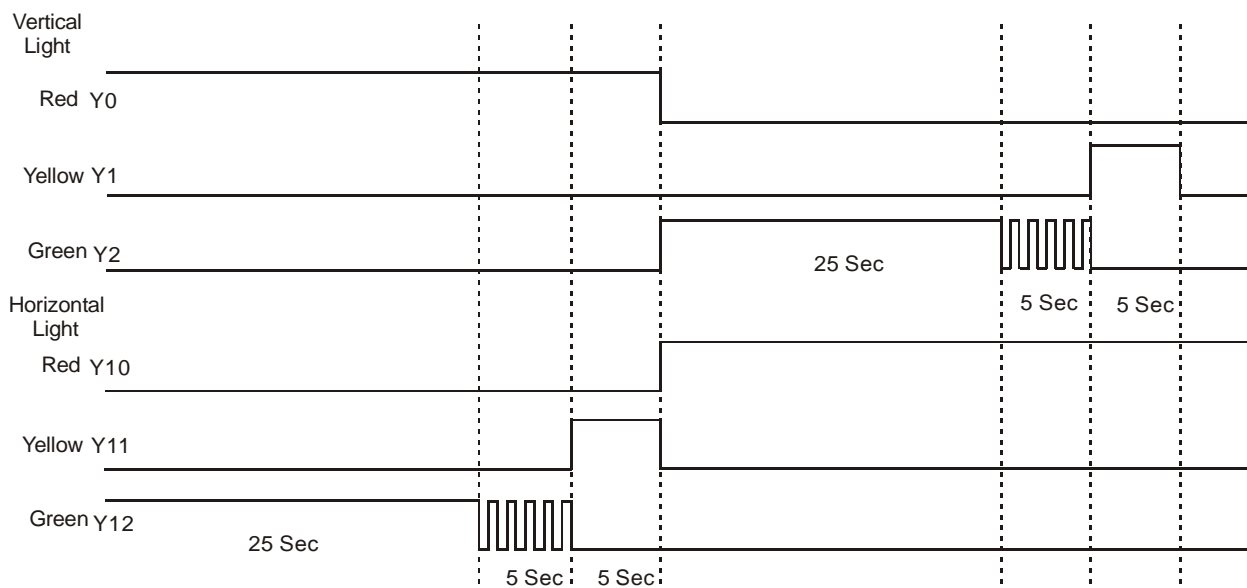
예제 15: 교통 신호 제어(스텝 Ladder 커맨드사용)



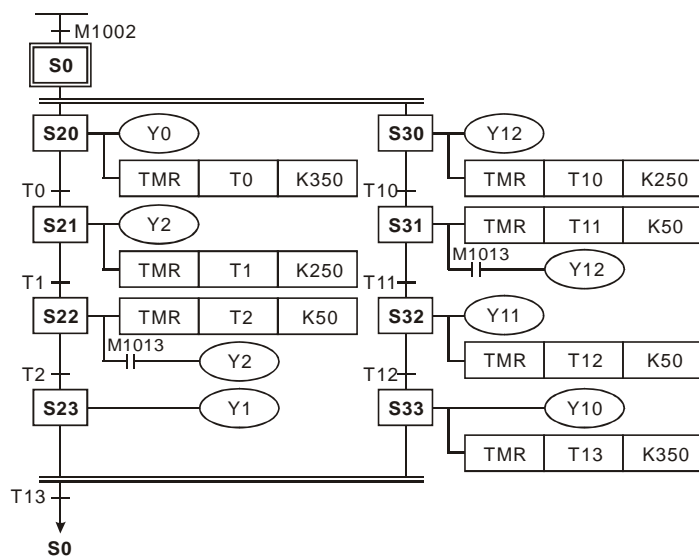
교통신호 제어

	붉을 불	노란 불	녹색 불	깜빡이는 녹색 불
세로 불	Y0	Y1	Y2	Y2
가로 불	Y10	Y11	Y12	Y12
Light Time	35 Sec	5 Sec	25 Sec	5 Sec

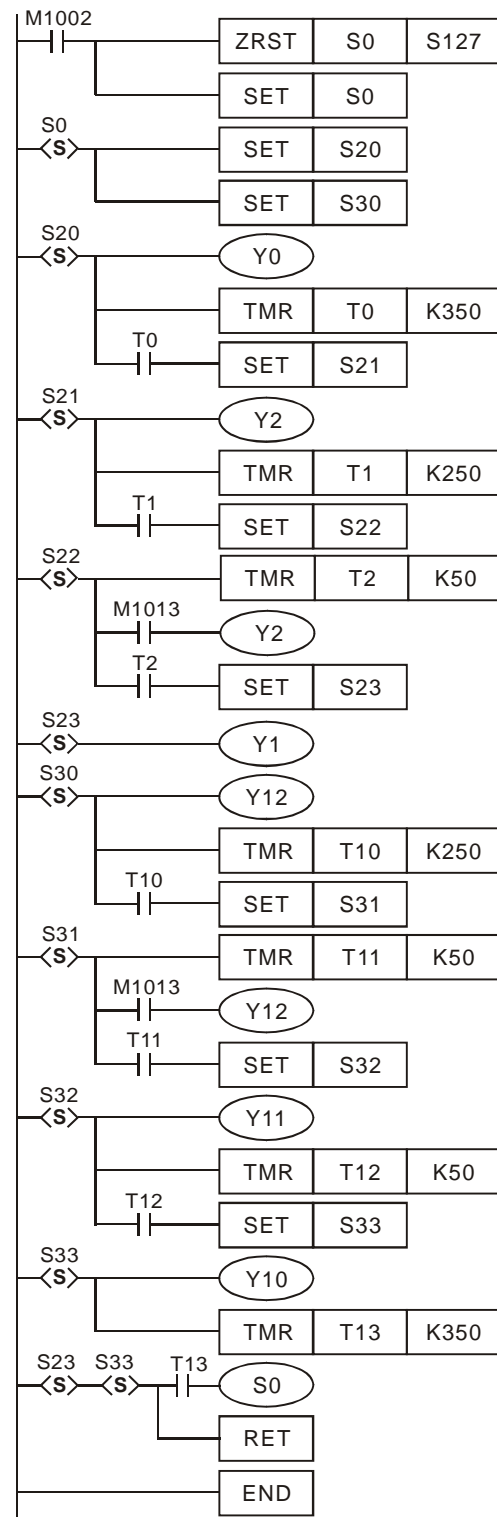
타이밍 차트:



SFC 도식:



Ladder 다이어그램:



## ■ SFC 에디터(WPLsoft)로 그리기

SFC 로 그리기	내부 Ladder 다이어그램 보기
	<p>■ LAD-0</p>
	<p>■ Transferred condition 1</p>
	<p>■ S22</p>
	<p>■ Transferred condition 4</p>
	<p>■ Transferred condition 7</p>

## 2.1 DVP-PLC 장치 번호 요약

ES, EX, SS 모델:

유형	장치	항목		사용 범위		기능	
릴레이 bit mode	X	외부 입력 릴레이		X0~X177, 128 points, 8 진수 시스템	총 256 포인트	외부 입력 포인트와 일치	
	Y	외부 출력 릴레이		Y0~Y177, 128 points, 8 진수 시스템		외부 출력 포인트와 일치함.	
	M	보조 릴레이	일반적으로 사용 시	M0~M511, M768~M999, 744 points	총 1280 포인트	프로그램을 통해 On/Off 스위치에 대한 접촉이 가능함.	
			래치 사용 시 *	M512~M767, 256 points			
			특별한 경우	M1000~M1279, 280 points (몇몇은 래치가 적용됨)			
	T	타이머	100ms 타이머	T0~T63, 64 points	총 128 포인트	타이머가 TMR 명령이 설정됐음을 가리킬 때 T 연결과 T 연결과 같은 번호는 On 이 될 것임.	
			10ms 타이머	T64~T126, 63 points (M1028 이 On 일 때, 10ms 이고, M1028 이 Off 일 때, 100ms 입니다..)			
			1ms 타이머	T127, 1 points			
	C	카운터	일반적으로 16-bit 합산	C0~C111, 112 points	총 128 포인트	타이머가 CNT(DCNT)명령이 설정됐음을 가리킬 때 C 연결과 C 연결과 같은 번호는 On 이 될 것임.	
			래치 사용 시 16-bit 합산 *	C112~C127, 16 points			
			고속카운터로 래치 사용 시 32-bit 합산하거나 가감함. *	1-phase 입력	C235~C238, C241, C242, C244, 7 points		총 13 포인트
				2 개의 1-phase 입력	C246, C247, C249, 3 points		
				2 개의 2-phase 입력	C251, C252, C254, 3 points		
S	스텝 point	래치 초기 스텝 point *	S0~S9, 10 points	총 128 포인트	스텝 ladder diagram(SFC) 장치 사용		
		0 point 를 반환하는 래치 *	S10~S19, 10 points (IST 명령과 함께 사용)				
		래치 사용 *	S20~S127, 108 points				
레지스터 WORD 데이터	T	타이머 종류		T0~T127, 128 points		타이머가 포인트를 가리키면 해당 포인트는 On 이 됨	
	C	카운터 종류		C0~C127, 16-bit 카운터, 128 C235~C254, 32-bit 카운터, 13 points		타이머가 포인트를 가리키면 해당 포인트는 On 이 됨	
	D	데이터 레지스터	일반적으로 사용 시	D0~D407, 408 points	총 600 포인트	저장 데이터가 메모리 영역에 있을 때 사용할 수 있습니다.. E 와 F 는 특정 지시 인덱스를 사용할 때만 가능합니다..	
			고리에 걸 경우 *	D408~D599, 192 points			
			특별한 경우	D1000~D1311, 312 points	총 312 포인트		
지시 인덱스	E(=D1028), F(=D1029), 2 points						
포인트	N	중첩 루프의 주 제어		N0~N7, 8 points		중첩 루프의 주 제어 포인트 제어	
	P	For CJ, CALL 명령		P0~P63, 64 points		CJ, CALL 포인트의 위치	
	I	인터럽트	시간 인터럽트	I6□□, 1 point (□□=10~99ms) (5.7 버전에서 사용)		서브루틴 인터럽트의 포인트 위치	
			외부 인터럽트	I001, I101, I201, I301, 4 points			
			입력 시간 인터럽트	I6□□, 1 point (□□=10~99, basic time=1ms ) (5.7 버전에서 사용)			

유형	장치	항목	사용 범위	기능
		커뮤니케이션	I150, 1 point	
사 랑	K	십진법	K-32,768 ~ K32,767 (16-bit 명령) K-2,147,483,648 ~ K2,147,483,647 (32-bit 명령)	
	H	16 진법	H0000 ~ HFFFF (16-bit 명령) H00000000 ~ HFFFFFFFF (32-bit 명령)	

\* 래칭이 고정되면 변경할 수 없습니다.

SA, SX, SC 모델:

유형	장치	항목	범위	기능
bit mode 릴레이	X	외부 입력 릴레이	X0~X177, 128 points, 8 진수 시스템	총 256 포인트 외부 입력 포인트와 일치
	Y	외부 출력 릴레이	Y0~Y177, 128 points, 8 진수 시스템	총 256 포인트 외부 출력 포인트와 일치
	M	보조 릴레이	일반적으로 사용 시	총 4096 포인트 프로그램에서 On/Off 스위치가 작동되는동안 접속할 수 있음. (몇몇은 래치가 적용됨)
			래치 사용 시 *	
			특별한 경우	
	T	타이머	100ms	총 256 포인트 타이머가 TMR 명령이 설정됐음을 암시할 때 T 연결과 T 연결과 같은 번호는 On 이 될 것임.
			10ms	
			1ms	
	C	카운터	16-bit 합산	총 250 포인트 타이머가 CNT(DCNT)명령이 설정됐음을 암시할 때 C 연결과 C 연결과 같은 번호는 On 이 될 것임.
			C96~C199, 104 points (*3)	
			32-bit 합산/감산	
			C200~C215, 16 points (*1) C216~C234, 19 points (*3)	
			32-bit 고속 카운터	
			C235~C244, 1-phase 1 input, 9 points (*3) C246, C247, C249, 1-phase 2 inputs, 3 points (*3) C251, C252, C254, 2-phase 2 inputs, 3 points (*3)	
	S	스텝 포인트	32-bit 고속 카운터 (SC 시리즈만 해당됨.)	총 3 포인트
			C243, C245, 1-phase 1 input, 2 point (*3), C250, 1-phase 2 input, 1 point (*3)	총 1024 포인트 스텝 ladder diagram 장치 사용
			초기 스텝 포인트	
			S0~S9, 10 points (*1)	
			0 포인트 반환	
WORD	T	타이머 종류	T0~T255, 256 points	타이머가 도달되면 연결은 On 이 됨.
	C	카운터 종류	C0~C199, 16-bit 카운터, 200 points C200~C254, 32-bit 카운터, 50 points (SC 모델에서 사용, 53 points)	타이머가 도달되면 연결은 On 이 됨.

유형	장치	항목		범위		기능
	D	데이터 레지스터	일반적인 사용	D0~D199, 200 points (*1)	Total is 5000 points	이것은 저장 데이터가 메모리 영역에 있을 때 사용할 수 있습니다.. E 와 F 는 특정 지시 인덱스를 사용할 때만 가능합니다..
			레치에서 사용*	D200~D999, 800 points (*3) D2000~D4999, 3000 points (*3)		
특별한 경우 사용	D1000~D1999, 1000 points					
지시 인덱스	E0~E3, F0~F3, 8 points (*1)					
	종류 없음	파일 레지스터 *		K0~K1599 (1600 points) (*4)		저장 데이터의 확장 레지스터임.
포인터	N	주 중첩 제어		N0~N7, 8 points		주 중첩 제어의 제어 포인트
	P	CJ, CALL 명령		P0~P255, 256 points		CJ, CALL 포인터의 위치
	I	인터럽트	외부 인터럽트	I001, I101, I201, I301, I401, I501, total is 6 points		서브루틴 인터럽트 포인터의 위치
			시간 인터럽트	I6□□, I7□□, 2 points (□□=1~99ms, time base=1ms)		
			고속 카운터가 인터럽트에 도착	I010, I020, I030, I040, I050, I060, 6 points		
			커뮤니케이션 인터럽트	I150, 1 point		
상수	K	10 진법 시스템		K-32,768 ~ K32,767 (16-bit 명령) K-2,147,483,648 ~ K2,147,483,647 (32-bit 명령)		
	H	16 진법 시스템		H0000 ~ HFFFF (16-bit 명령) H00000000 ~ HFFFFFFFF (32-bit 명령)		

\*1: 고정된 비래칭 영역은 변경될 수 없음.

\*2: 비래칭 영역은 매개 변수 설정에 의한 래칭 영역으로 전환 가능함.

\*3: 래칭 영역은 매개 변수 설정에 의한 비래칭 영역으로 전환 가능함.

\*4: 고정된 래치 영역은 수정될 수 없음.(【】마크가 표시된 영역은 변경될 수 없음.)



각 SA, SX, SC 모델 래치 설정:

M 보조릴레이	일반적인 사용 시	래치 사용 시	특정 보조 릴레이		래치
	M0~M511	M512~M999	M1000~M1999		M2000~M4095
	비래칭으로 고정됨.	공장에서부터 래치로 설정되어 생산됨. 시작: D1200(K512) 종료: D1201(K999)	몇몇은 래치이며 변경할 수 없음.		공장에서부터 래치로 생산됨. 시작: D1202(K2000) 종료: D1203(K4095)
T 타이머	100 ms	10 ms	10ms	1 ms	100 ms
	T0 ~T199	T200~T239	T240~T245	T246~T249	T250~T255
	비래칭으로 고정됨.	비래칭으로 고정됨.	누적기 타입 래칭으로 고정됨.		
C 카운터	16 bits 합산		32 bits 합산/감산		32 bits 고속 카운터로 합산/감산
	C0~C95	C96~C199	C200~C215	C216~C234	C235~C255
	비래칭으로 고정됨.	래칭으로 고정됨. 시작: D1208 (K96) 종료: D1209 (K199)	비래칭으로 고정됨.	래칭으로 고정됨. 시작: D1210 (K216) 종료: D1211 (K234)	공장에서부터 래칭으로 설정됨. 시작: D1212 (K235) 종료: D1213 (K255)
S 스텝 릴레이	일반적	래칭	특정 레지스터	래칭	일반적
	S0~S9	S10~S19	S20~S511	S512~S895	S896~S1023
	비래칭으로 고정됨.			공장에서부터 래칭으로 설정됨. 시작: D1214 (K512) 종료: D1215 (K895)	래칭으로 고정됨.
D 레지스터	일반적	래칭	특정 레지스터	래칭	
	D0~D199	D200~D999	D1000~D1999	D2000~D4999	
	비래칭으로 고정됨	공장에서부터 래칭으로 설정됨. Start: D1216 (K200) End: D1217 (K999)	몇몇은 래칭이며 변경할 수 없음.	공장에서부터 래칭으로 설정됨. 시작: D1218 (K2000) 종료: D1219 (K4999)	
데이터 레지스터	K0~K1599				
	래칭으로 고정됨.				

EH 모델:

유형	장치	항목		범위		기능	
릴레이 bit mode	X	외부 입력 릴레이		X0~X377, 256 points, 8 진수 시스템		총 512 포인트	외부 입력 포인트와 일치함.
	Y	외부 출력 릴레이		Y0~Y377, 256 points, 8 진수 시스템			외부 출력 포인트와 일치함.
	M	누전기	일반적인 사용	M0~M499, 500 points (*2)		총 4096 포인트	프로그램에서 On/Off 스위치가 작동되는 사이 접속할 수 있음.(몇몇은 래치임)
			래칭	M500~M999, 500 points (*3) M2000~M4095, 2096 points (*3)			
			특별한 경우	M1000~M1999, 1000 points (몇몇은 래치임)			
	T	타이머	100ms	T0~T199, 200 points (*2) T192~T199 는 서브루틴에 해당됨. 【T250~T255】 , 6-point 누전기 타입 (*4)		총 256 포인트	타이머가 TMR 명령으로 설정됐을 때, T 연결과 T 연결과 같은 번호는 On 이 됨.
			10ms	T200~T239, 40 points (*2) 【T240~T245】 , 6-point 누전기 타입 (*4)			
			1ms	【T246~T249】 , 4-point 누전기 타입 (*4)			

유형	장치	항목	범위	기능
비틀레이 bit mode	C	카운터	16-bit 합산	총 253 포인트 타이머가 CNT(DCNT) 명령으로 설정됐을 때, C 는 On 으로 연결됨.
			32-bit 합산/감산	
			32-bit 고속 카운터	
	S	스텝 포인트	초기 스텝 포인트	총 1024 포인트 스텝 ladder diagram (SFC) 장치의 사용
			0 포인트 반환	
			일반적	
			래칭	
			알람	
레지스터 WORD 데이터	T	타이머의 종류	T0~T255, 256 points	타이머가 도달했을 때 On 으로 접속되어 있을 것입니다..
	C	카운터의 종류	C0~C199, 16-bit counter, 200 points C200~C254, 132-bit counter, 53 points	타이머가 도달했을 때 On 으로 접속되어 있을 것입니다..
	D	데이터 레지스터	일반적	총 10000 포인트 저장 데이터의 메모리 영역입니다.. E 와 F 는 특정 지시 인덱스를 사용할 때만 가능합니다..
			래칭	
			특별한 경우	
			인덱스 사용 시	
포인트	종류 없음	파일 레지스터	K0~K9999(10000 points) (*4)	저장 데이터의 확장 레지스터 임.
	N	주 중첩 제어	N0~N7, 8 points	주 중첩 제어의 제어 포인트
	P	CJ, CALL 명령	P0~P255, 256 points	CJ, CALL 포인트의 위치
	I	비밀점	외부 인터럽트	서브루틴 인터럽트 포인트의 위치
			시간 인터럽트	
			인터럽트에 도달되는 고속 카운터	
			펄스 인터럽트	
			커뮤니케이션 인터럽트	
			입력 도수 측량 카드 인터럽트	
상수	K	10 진수 시스템	K-32,768 ~ K32,767 (16-bit 명령) K-2,147,483,648 ~ K2,147,483,647 (32-bit 명령)	
	H	16 진수 시스템	H0000 ~ HFFFF (16-bit 명령) H00000000 ~ HFFFFFFFF (32-bit 명령)	

\*1: 고정된 비래칭 영역은 변경될 수 없음.

\*2: 비래칭 영역은 매개 변수 설정에 의한 래칭 영역으로 전환 가능함.

\*3: 래칭 영역은 매개 변수 설정에 의한 비래칭 영역으로 전환 가능함.

\*4: 고정된 래치 영역은 수정될 수 없음. (【】마크가 표시된 영역은 변경될 수 없음.)

각 EH 모델에서의 래칭 설정:

M 보조 릴레이	일반적인 사용 시		래칭 사용 시		특정 보조 릴레이		래칭 시					
	M0~M499		M500~M999		M1000~M1999		M2000~M4095					
	시작: D1200(K500) 종료: D1201(K999)				몇몇은 래칭이며 변경할 수 없습니다.		시작: D1202(K2000) 종료: D1203(K4095)					
T 타이머	100 ms		10 ms		10ms		1 ms		100 ms			
	T0 ~T199		T200~T239		T240~T245		T246~T249		T250~T255			
	공장에서부터 비래칭으로 설정됨		공장에서부터 비래칭으로 설정됨		고정된 래칭의 누전기 타입							
	시작: D1204 (K-1)*1 종료: D1205 (K-1)*1		시작: D1206 (K-1)*1 종료: D1207 (K-1)*1									
C 카운터	16-bit 합산			32-bit 합산/감산			32-bit 고속 합산/감산					
	C0~C99		C100~C199		C200~C219		C220~C234		C235~C245		C246~C255	
	비래칭 (기본설정)		래칭 (기본설정)		비래칭 (기본설정)		래칭 (기본설정)		래칭 (기본설정)			
	시작: D1208 (K100) 종료: D1209 (K199)			시작: D1210 (K220) 종료: D1211 (K234)			시작: D1212 (K235) 종료: D1213 (K255)					
S 스텝 릴레이	초기값		0 포인트 반환		일반적		래칭		알람의 스텝 포인트			
	S0~S9		S10~S19		S20~S499		S500~S899		S900~S1023			
	비래칭 (기본설정)				래칭 (기본설정)				언제나 래칭임			
	시작: D1214 (K500) 종료: D1215 (K899)											
D 레지스터	일반적			래칭			특정 레지스터			래칭		
	D0~D199			D200~D999			D1000~D1999			D2000~D9999		
	비래칭 (기본설정)			래칭 (기본설정)			몇몇은 래칭이며 변경할 수 없습니다.			래칭 (기본설정)		
	시작: D1216 (K200) 종료: D1217 (K999)			시작: D1218 (K2000) 종료: D1219 (K9999)								

\* 1: K-1 은 공장에서부터 비래칭으로 설정되었음을 뜻합니다..

On/Off 전원 또는 MPU RUN/STOP 모드 사이에서 스위칭이 발생할 때 버전 5.5 와 SA, SX, SC 보다 높은 시리즈의 메모리 타입은 아래와 같이 합니다..:

메모리 타입	전원 Off=>On	STOP=>RUN	RUN=>STOP	모든 비래칭 영역 지우기 (M1031)	모든 래칭 영역 지우기 (M1032)	공장 설정
Non-latched	지움	M1033=Off 일 때, 지움		지움	변경불가	0
		M1033=On 일 때, 변경불가				
Latched	변경불가			변경불가	지움	변경불가
특정 M, 특정 D, 인덱스 레지스터	초기값	변경불가		변경불가		초기값으로 설정

SA, SX, SC, EH 모델의 메모리 타입은 아래를 따릅니다.

메모리 타입	전원 Off=>On	STOP=>RUN	RUN=>STOP	모든 비래칭 영역 지우기 (M1031)	모든 래칭 영역 지우기 (M1032)	공장 설정
비래칭	지움	변경불가	M1033=Off 일 때, 지움	지움	변경불가	0
			M1033=On 일 때, No 변경			

래칭	변경불가		변경불가	지움	0
특정 M, 특정 D, 인덱스 레지스터	초기값	변경불가	변경불가		초기값으로 설정
파일 레지스터	변경불가				0

## 2.2 값, 상수 [K] / [H]

상수	K	10 진수	K-32,768 ~ K32,767 (16-bit 명령) K-2,147,483,648 ~ K2,147,483,647 (32-bit 명령)
	H	16 진수	H0 ~ HFFFF (16-bit 명령) H0 ~ HFFFFFFFF (32-bit 명령)

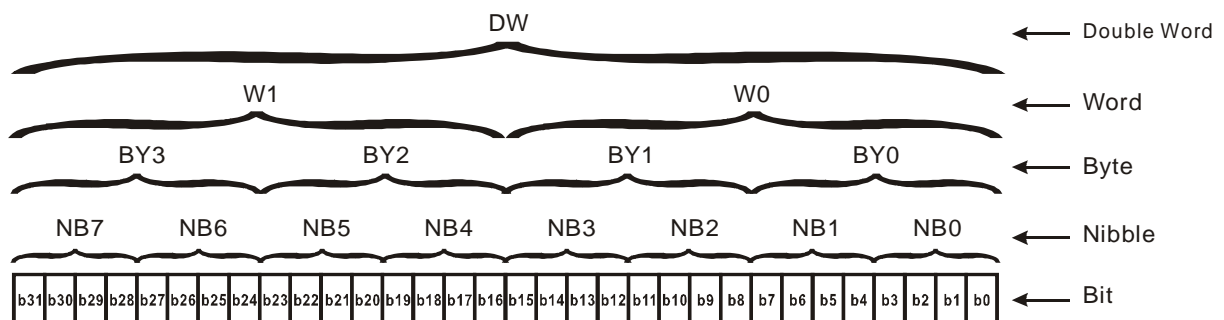
아래 다섯 가지의 DVP-PLC 타입은 각기 다른 도착점을 가집니다. 아래는 각 타입별 설명입니다..

### 1. 2 진수 (BIN)

2 진수는 PLC 내부 명령이나 저장소에서 사용됩니다. 2 진수에 관한 정보는 아래를 따릅니다.

Bit	:	Bit 는 2 진 시스템의 기본 단위이며 상태는 1 또는 0 을 가집니다.
Nibble	:	Nibble 은 b3~b0 과 같은 4bit 이상의 값으로 이루어집니다. 이것은 10 진수 0~9 의 값이나 16 진수 0~F 의 값을 나타내는데 사용되기도 합니다..
Byte	:	Byte 는 2 nibbles 이상의 값으로 이루어집니다. 바꿔말하면, b7~b0 의 값을 가진 8bit 로도 나타낼 수 있습니다.. 이것은 16 진수의 00~FF 의 값을 나타내는데 사용되기도 합니다..
Word	:	Word 는 2Byte 이상의 값으로 이루어집니다. 바꿔말하면, b15~b0 의 값을 가진 16bit 로도 나타낼 수 있습니다.. 이것은 16 진수의 0000~FFFF 의 값을 나타내는데 사용되기도 합니다..
Double Word	:	Double Word 는 2Word 이상의 값으로 이루어집니다. 바꿔말하면, b31~b0 의 값을 가진 32bit 로도 나타낼 수 있습니다.. 이것은 16 진수의 00000000~FFFFFFFF 의 값을 나타내는데 사용되기도 합니다..

bit, nibble, byte, word, 그리고 double word 사이의 2 진수간의 관계는 아래와 같습니다.



### 2. 8 진수 (OCT)

DVP-PLC 의 외부 입력 및 출력 단자는 8 진수를 사용합니다..

예:

외부 입력: X0~X7, X10~X17...(장치 번호)

외부 출력: Y0~Y7, Y10~Y17...(장치 번호)

### 3. 10 진수 (DEC)

DVP-PLC 시스템에서는 아래의 상황에서 10 진수를 사용하는 것이 적당합니다..

- TMR C0 K50 같은 T 타이머나 C 카운터의 설정 값으로 사용됩니다. (K 상수)
- 예를 들어 M10, T30 같은 S, M, T, C, D, E, F, P, I 의 장치번호로 사용됩니다. (장치 번호)
- MOV K123 D0 같은 응용 프로그램 명령의 연산자로 사용됩니다. (K 상수)

#### 4. BCD (2 진화 10 진 코드, BCD)

이것은 장치 번호에서 사용되는 10 진수나 4 비트에서 16 비트까지 확장되는 10 진수를 4 개의 수로 나타내는데 사용됩니다. BCD 코드는 보통 DIP 스위치 입력 값을 나타내거나 7-Segment 의 출력 값을 표현하는데 사용됩니다.

#### 5. 16 진수(HEX)

DVP-PLC 시스템에서는 아래의 상황에서 16 진수를 사용하는 것이 적당합니다..

- 예를 들어 MOV H1A2B D0 같은 응용 프로그램 명령의 연산자로 사용됩니다. (상수 H)

상수 K:

PLC 에서, K 는 10 진수의 앞에 위치하며 보통 10 진수를 나타내는 상수로 사용됩니다. 예를 들면, K100 은 10 진수 100 을 나타냅니다.

예외:

K 로 만들어 지는 값과 X, Y, M, S 장비에서 bit, byte, word, double word 로 사용되는 비트는 예외입니다.. 예를 들어, K2Y10, K4M100. K1 는 4-비트 데이터를 나타내며 K2~K4 는 각 8, 12, 16 비트 데이터를 나타냅니다.

상수 H:

PLC 에서, H 는 16 진수의 앞에 위치하며 보통 16 진수를 나타내는 상수로 사용됩니다. 예를 들면, H100 은 16 진수 100 을 나타냅니다.

참고 표:

2 진수 (BIN)		8 진수 (OCT)	10 진수 (DEC)	BCD (2 진화 10 진수)		16 진수 (HEX)
PLC 내부 명령으로 사용		X, Y 장치 번호	상수 K, 장치 M, S, T, C, D, E, F, P, I 의 번호	DIP Switch 와 7-Segment 의 표현		상수 H
0 0 0 0	0 0 0 0	0	0	0 0 0 0	0 0 0 0	0
0 0 0 0	0 0 0 1	1	1	0 0 0 0	0 0 0 1	1
0 0 0 0	0 0 1 0	2	2	0 0 0 0	0 0 1 0	2
0 0 0 0	0 0 1 1	3	3	0 0 0 0	0 0 1 1	3
0 0 0 0	0 1 0 0	4	4	0 0 0 0	0 1 0 0	4
0 0 0 0	0 1 0 1	5	5	0 0 0 0	0 1 0 1	5
0 0 0 0	0 1 1 0	6	6	0 0 0 0	0 1 1 0	6
0 0 0 0	0 1 1 1	7	7	0 0 0 0	0 1 1 1	7
0 0 0 0	1 0 0 0	10	8	0 0 0 0	1 0 0 0	8
0 0 0 0	1 0 0 1	11	9	0 0 0 0	1 0 0 1	9
0 0 0 0	1 0 1 0	12	10	0 0 0 1	0 0 0 0	A
0 0 0 0	1 0 1 1	13	11	0 0 0 1	0 0 0 1	B
0 0 0 0	1 1 0 0	14	12	0 0 0 1	0 0 1 0	C
0 0 0 0	1 1 0 1	15	13	0 0 0 1	0 0 1 1	D

2 진수 (BIN)		8 진수 (OCT)	10 진수 (DEC)	BCD (2 진화 10 진수)		16 진수 (HEX)
PLC 내부 명령으로 사용		X, Y 장치 번호	상수 K, 장치 M, S, T, C, D, E, F, P, I 의 번호	DIP Switch 와 7-Segment 의 표현		상수 H
0 0 0 0	1 1 1 0	16	14	0 0 0 1	0 1 0 0	E
0 0 0 0	1 1 1 1	17	15	0 0 0 1	0 1 0 1	F
0 0 0 1	0 0 0 0	20	16	0 0 0 1	0 1 1 0	10
0 0 0 1	0 0 0 1	21	17	0 0 0 1	0 1 1 1	11
:	:	:	:	:	:	:
:	:	:	:	:	:	:
:	:	:	:	:	:	:
0 1 1 0	0 0 1 1	143	99	1 0 0 1	1 0 0 1	63

### 2.3 외부 입/출력 점검의 기능과 넘버링 [X] / [Y]

입/출력 상수 번호:(8 진수)

MPU 에서, 입력과 출력 상수는 X0 와 Y0 로 합산됩니다. 이 상수는 MPU 포인트와 함께 변경될 수 있습니다.. 확장된 I/O 장치에서, 입/출력 단자 수는 MPU 의 연속적인 연결과 함께 합산됩니다.

ES, EX, SS 모델:

모델 번호	DVP-14ES	DVP-14SS	DVP-20EX	DVP-24ES	DVP-32ES	DVP-60ES	확장 I/O
X 입력	X0~X7 (8 Points)	X0~X7 (8 Points)	X0~X7 (8 Points)	X0~X17 (16 Points)	X0~X17 (16 Points)	X0~X43 (36 Points)	X20(X50)~X177 (Note)
Y 출력	Y0~Y5 (6 Points)	Y0~Y5 (6 Points)	Y0~Y5 (6 Points)	Y0~Y7 (8 Points)	Y0~Y17 (16 Points)	Y0~Y27 (24 Points)	Y20(Y30)~Y177 (Note)

Note: 게다가 DVP-60ES 는, X20 에서 확장된 장치의 입력번호로부터 시작되며 Y20 에서 확장된 장치의 출력번호로부터 시작됩니다. DVP-60ES 의 입력 번호로 시작되는 X50 과 the started output number of DVP-60ES 의 출력 번호로 시작되는 Y30 이 보통입니다.. 확장 I/O 의 번호는 8 번 증가되며 만일 8point 보다 적으면, 8point 와 함께 계산합니다..

SA, SX, SC 모델:

모델 번호	DVP-10SX(Note1)	DVP-12SA(Note1)	DVP-12SC(Note1)	확장 I/O
X 입력	X0~X3 (4 points)	X0~X7 (8 points)	X0~X5 、 X10~X11 (8 points)	X20~X177 (note 2)
Y 출력	Y0~Y1 (2 points)	Y0~Y3 (4 points)	Y0~Y1 、 Y10~Y11 (4 points)	Y20~Y177 (note 2)

Note 1: 모든 SA, SX, 그리고 SC 시리즈의 외부 장치는 SS 시리즈와 같이 공유됩니다. 4DI 와 2DO 외에 또 SX 시리즈 또한 2AI(12 비트)와 2AO(12 비트)를 가지고 있습니다..

Note 2: 확장 I/O 장치의 입력 번호는 X20 으로부터 시작되며 출력 번호는 Y20 으로부터 시작됩니다. 확장 번호의 연산자는 SS 시리즈와 거의 같습니다.

EH 모델:

모델 번호	DVP-16EH	DVP-20EH (Note1)	DVP-32EH (Note1)	DVP-48EH	DVP-64EH	DVP-80EH	확장 I/O (note3)
-------	----------	---------------------	---------------------	----------	----------	----------	-------------------

X 입력	X0~X7 (8 points)	X0~X13(12 points)	X0~X17 (16 points)	X0~X27 (24 points)	X0~X37 (32 points)	X0~X47 (40 points)	X※~X377
Y 출력	Y0~Y7 (8 points)	Y0~Y7(8 points)	Y0~Y17 (16 points)	Y0~Y27 (24 points)	Y0~Y37 (32 points)	Y0~Y47 (40 points)	Y※~Y377

**Note1:** 20EH 와 32EH 를 제외하고, 모든 MPU 모델은 출력 트랜지스터 타입을 가지며 Y0 와 Y2 는 모든 저속 출력을 가집니다. 20EH 와 32 EH 모델에서만, Y0 와 Y2 는 고속 출력을 가지며(200KHz) 그 이외는 저속입니다.(10KHz).

**Note2:** 릴레이 출력의 터미널 레이아웃과 32EH 출력 레지스터는 나머지 모델들과 비교해서 다른 특성을 가집니다. EH MPU 명령어 시트로 보내는 것을 권합니다..

**Note3:** I/O 확장 장치에서, 입/출력 번호는 연속된 MPU 의 마지막 번호를 더함으로써 시작됩니다. DVP-16EH 와 DVP-20EH 의 I/O 확장 장치는 X20 으로부터 입력번호가 시작되며 Y20 로부터 출력번호가 시작됩니다. I/O 확장 장치의 입/출력 번호는 연속된 시점부터 증가됩니다. 최고 입력 번호가 X377 보다 넘어서는 동안 출력번호는 Y377 을 넘어서나.

### ■ 입력 릴레이: X0~X377

입력 릴레이(또는 입력 터미널이라 불린다.)의 수는 8 진수를 사용합니다.. EH model 은 256 포인트보다 더 사용할 수 있으며 범위는 다음과 같습니다. : X0~X7, X10~X17, ....., X370~X377.

### ■ 출력 릴레이: Y0~Y377

출력 릴레이(또는 출력 터미널이라 불린다.)의 수는 8 진수를 사용합니다.. EH model 은 256 포인트보다 더 사용할 수 있으며 범위는 다음과 같습니다. : Y0~Y7, Y10~Y17, ....., Y370~Y377.

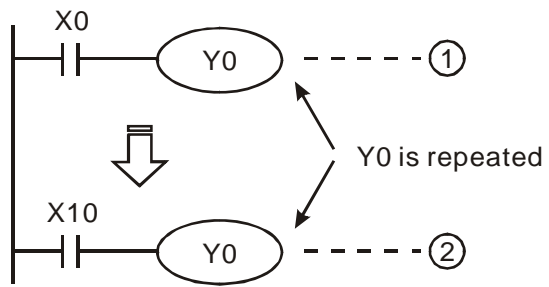
### ■ 입/출력 연결 기능

**X 접속 입력의 기능:** X 접속 입력은 입력 신호를 읽고 입력 장비와 PLC 와의 연결로 PLC 로 들어간다. 프로그램 내 각 X 접속 입력에서 A 접속 또는 B 접속 시간을 무한대로 사용할 수 있습니다.. X 접속 입력에서 On/Off 는 입력 장비의 On/Off 와 함께 변경될 수 있으나 주변장치기기(HPP 또는 WPLSoft) 사용으로 인한 변경은 할 수 없습니다.

(※ 특수 릴레이인 M1304 의 SA, SX, SC, EH 모델은 HPP 나 WPLSoft 같은 주변기기의 X On/Off 입력으로 인한 On/Off 작동이 가능하나 PLC 는 외부 입력 신호가 발생하는 시점에서 응답을 받을 수 없습니다.)

### ■ Y 접속 출력 기능:

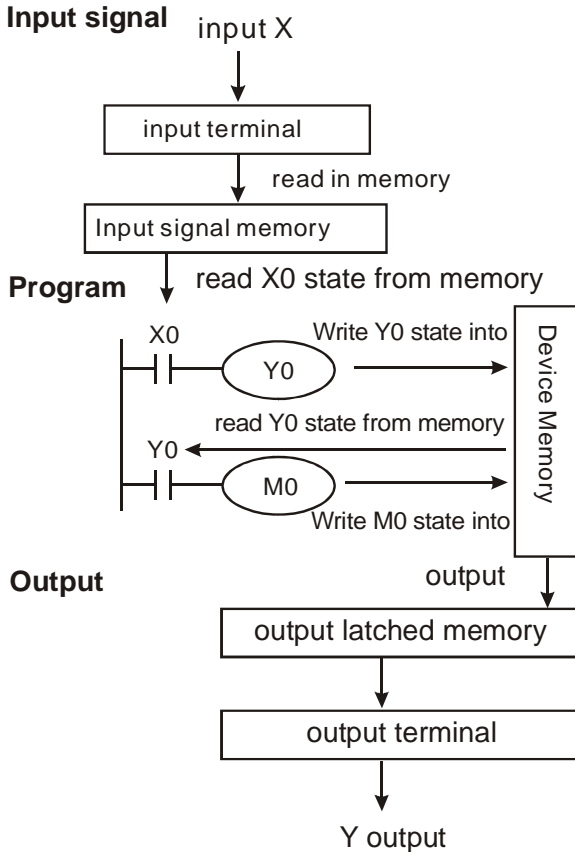
Y 접속 출력은 Y 접속 출력에 의해 보내지는 On/Off 신호를 연결시켜주는 길을 운전하는 일을 합니다.. 이 기능은 두 가지의 접속 출력을 가집니다.: 하나는 릴레이와 다른 트랜지스터와의 접속입니다.. 프로그램 내에서 각 A 나 B 접속의 Y 접속 출력은 무한대로 사용할 수 있습니다.. 그러나 프로그램 내에서 Y 코일 출력과 이 첫번째 접속 출력은 한 번만 사용하기를 권장합니다.. 다른 하나는, 마지막 Y 출력의 회로와 함께 PLC 프로그램의 입력 방법에 대한 출력을 결정하게 될 것입니다..



Y0의 출력은 ②번의 회로를 결정하게 됩니다.  
바꿔말하면, X10의 On/Off를 결정짓는다.



## PLC 프로그램의 흐름과 운용법 (I/O 배치)



### ● 입력 신호:

1. PLC 는 프로그램이 실행되기 전 입력 신호의 메모리에 입력된 On/Off 신호를 읽는다.
2. 만일 프로그램이 실행되는동안 입력 신호 On/Off 가 변하면 이 입력신호의 메모리 상태는 변할 수 없습니다. 새로운 On/Off 상태는 메모리의 다음 스캔 시 입력될 수 있습니다..
3. 외부 신호 On→Off 이나 Off→On 로의 연결의 변경은 10ms 의 지연 시간을 야기합니다..

### ● 프로그램:

입력 신호 메모리 내 On/Off 입력 신호 상태를 읽은 뒤 프로그램에서 0 번지 주소로부터 PLC 를 실행시키는 각 명령을 발생시키고 각 장치 메모리 내 각 On/Off 의 출력 코일을 저장합니다..

### ● 출력:

1. END 명령이 실행되면, Y 메모리의 On/Off 상태를 출력 래칭 메모리로 내보낸다. 사실상, 이 메모리는 출력 릴레이의 코일입니다..
2. 릴레이 코일의 접속에서 On→Off 나 Off→On 로의 변경에 대한 지연시간은 10ms 이 될 것입니다..
3. 트랜지스터 모듈 사용 시 On→Off 이나 Off→On 로의 변경에 대한 지연시간은 10~20us 가 될 것입니다..

## 2.4 보조 릴레이의 기능과 넘버링 [M]

보조 릴레이 번호:(10 진수)

ES, EX, SS 모델:

보조 릴레이 M	일반적일 때	M0~M511, M768~M999, 744 points. 비래칭 영역으로 고정됨.	총 1280 포인트
	래칭 시	M512~M767, 256 points. 래칭 영역으로 고정됨.	
	특정 시	M1000~M1279, 280 points. 몇몇은 래칭임.	

SA, SX, SC 모델:

보조 릴레이 M	일반적일 때	M0~M511, 512 points. 비래칭 영역으로 고정됨.	총 4096 포인트
	래칭 치	M512~M999, M2000~M4095, 2584 points. 전달값에 의해 비래칭 영역으로 전환 가능함.	
	특정 시	M1000~M1999, 1000 points. 몇몇은 래칭임.	

EH 모델:

보조 릴레이 M	일반적일 때	M0~M499, 500 points. 전달된 설정값에 의해 래칭으로 전환 가능함.	총 4096 포인트
	래칭 시	M500~M999, M2000~M4095, 2596 points. 전달된 설정 값에 의해 비래칭으로 전환 가능함.	
	특정 시	M1000~M1999, 1000 points. 몇몇은 래칭임.	

보조 릴레이 기능:

이것은 프로그램 내에서 무한대로 사용할 수 있는 보조 릴레이 M과 출력 릴레이 Y의 A, B 접속과 출력 코일입니다.. 사용자는 보조 릴레이의 반복을 제어할 수 있으나 외부에서 직접적으로 연결된 장치는 제어할 수 없습니다. 아래는 보조 릴레이를 세 가지 종류로 나누어 표현했다.

1. 일반적으로 사용되는 보조 릴레이 : 장치가 실행중일 때 전원을 손실하면 Off로 재설정합니다.. 전원을 손실한 후 전원이 들어오면 상태는 Off를 나타낼 것입니다..
2. 래칭에서 사용되는 보조 릴레이 : 실행중일 때 전원을 손실하거나 전원 손실 후 전원이 들어오면 전원이 손실되기 전 상태를 저장할 것입니다..
3. 특정 보조 릴레이 : 각 특정 보조 릴레이는 각각 특정 기능을 가지고 있습니다.. 불명확한 보조 릴레이의 사용은 권장하지 않는다. 2.10 번과 특정 릴레이와 특정 레지스터에서 사용되는 각 특정 보조 릴레이와 2.11 에서 소개되는 특정 보조 릴레이와 특정 레지스터의 기능을 참조하기 바랍니다.

## 2.5 스텝 릴레이의 기능과 넘버링 [S]

보조 릴레이의 번호 체계 (10 진수로 이루어짐):

ES, EX, SS 모델:

스텝 릴레이 S	래칭 초기값	S0~S9, 10 points. 래칭 영역으로 고정될 것입니다..	총 128 포인트
	래칭으로 반환되는 0 포인트	S10~S19, 10 points. (IST 명령과 함께 사용) 래칭 영역으로 고정될 것입니다..	
	래칭	S20~S127, 108 points. 래칭 영역으로 고정될 것입니다..	

SA, SX, SC 모델:

스텝 릴레이 S	초기값으로 사용	S0~S9, 10 points. 비래칭 영역으로 고정될 것임.	총 1024 포인트
	0 포인트 반환 시 사용	S10~S19, 10 points. (use with IST 명령과 함께 사용) 비래칭 영역으로 고정될 것임.	
	일반적으로 사용	S20~S511, 492 points. 비래칭 영역으로 고정될 것임.	
	래칭 시 사용	S512~S895, 384 points. 전달값에 의해 비래칭 영역으로 전환할 수 있습니다..	
	알람 시 사용	S896~S1023, 128 points. 래칭 영역으로 고정될 것임.	

EH 모델:

스텝 릴레이 S	초기값으로 사용	S0~S9, 10 points. 전달된 설정 값에 의해 래칭 영역으로 될 수 있습니다..	총 1024 포인트
	0 포인트 반환 시 사용	S10~S19, 10 points. (use with IST 명령과 함께 사용). 전달된 설정 값에 의해 래칭 영역으로 될 수 있습니다..	
	일반적으로 사용	S20~S499, 480 points. 전달된 설정 값에 의해 래칭 영역으로 될 수 있습니다..	
	래칭 시 사용	S500~S899, 400 points. 전달된 설정 값에 의해 비래칭 영역으로 될 수 있습니다..	
	알람 시 사용	S900~S1023, 124 points. 전달된 설정 값에 의해 래칭 영역으로 될 수 있습니다..	

스텝 릴레이의 기능:

스텝 릴레이 S는 스텝 Ladder diagram의 기본 장비이며 PLC운용을 손쉽게 할 수 있도록 합니다.. 스텝 Ladder diagram (또는 Sequential Function Chart (SFC)이라 불리기도 합니다..), STL, REL 등의 명령에서 사용되기도 합니다..

S0~S1023의 1024포인트는, 스텝 릴레이 S내에 존재합니다.. 출력 릴레이 Y와 같이, 코일을 출력하며 A, B접속은 각 스텝 릴레이 S와 무한대로 사용된다. 그러나 외부와 직접 연결된 장치에는 사용할 수 없습니다. 스텝 릴레이(S)는 스텝 명령을 사용할 수 없는 일반적인 보조 릴레이에서 사용됩니다. 아래는 스텝 릴레이를 4가지 종류로 나누어 표현했다.

1. 스텝 릴레이 초기값 : S0~S9, 10 points.  
Sequential Function Chart (SFC)에서는 초기값을 설정하기 위한 스텝 포인트로 사용됩니다.
2. 스텝 릴레이로 0 포인트 반환 : S10~S19, 10 points.  
S10 – S19는 프로그램 내에서 API60IST를 사용할 때 0포인트를 반환합니다.. 만일 IST 명령을 사용하지 못합니다.면, 일반적인 스텝 릴레이를 사용하게 될 것입니다..
3. 일반적인 스텝 릴레이 : SA, SX, SC 모델: S20~S511, 492 points. EH 모델: S20~S499, 480 points.  
이 스텝 포인트들은 보통 Sequential Function Chart(SFC)에서 사용됩니다. 프로그램 실행 중 전원을 손실했을 때 이 포인트는 지워진다.
4. 스텝 릴레이 래칭 : ES, EX, SS 모델: S20~S127, 108 points. SA, SX, SC 모델: S512~S895, 384points. EH 모델: S500~S899, 400 points.  
Sequential Function Chart (SFC)에서, 스텝 릴레이 래칭은 프로그램 실행 중 전원을 손실했을 때 각 포인트를 저장합니다.. 전원 손실 후 전원이 돌아왔을 때 상태는 전원을 손실하기 전 상태로 돌아와 있을 것입니다..
5. 알람을 통한 스텝 릴레이 : SA, SX, SC 모델: S896~S1023, 128 points. EH 모델: S900~S1023, 124 points.

알람을 통한 스텝 릴레이는 알람과 접속할 수 있는 API 46 ANS 알람 장치 명령과 함께 사용됩니다.. 경고 기록과 외부에서 발생한 오작동을 제거하는데 사용됩니다.

## 2.6 타이머의 기능과 넘버링 [T]

타이머의 번호 체계 (10 진수 사용):

ES, EX, SS 모델:

타이머 T	일반적인 100ms	T0~T63, 64 points	총 128 포인트
	일반적인 10ms	T64~T126, 63 points (M1028=On 일 때, 10ms 이며. M1028=Off 일 때, 100ms 입니다..)	
	일반적인 1ms	T127, 1 point	

SA, SX, SC 모델:

타이머 T	일반적인 100ms	T0~T199, 200 points. (T192~T199 는 서브루틴에서 사용되는 타이머임.)	총 256 포인트
	축적하기 위한 100ms	T250~T255, 6 points. 래칭 영역으로 고정됨.	
	일반적인 10ms	T200~T239, 40 points.	
	축적하기 위한 10ms	T240~T245, 6 points. 래칭 영역으로 고정됨.	
	축적하기 위한 1ms	T246~T249, 4 points. 래칭 영역으로 고정됨.	

EH 모델:

타이머 T	일반적인 100ms	T0~T199, 200 points. 전달된 설정 값에 의해 래칭 영역으로 될 수 있습니다.. (T192~T199 는 서브루틴에서 사용되는 타이머 임.)	총 256 포인트
	축적하기 위한 100ms	T250~T255, 6 points. 래칭 영역으로 고정됨.	
	일반적인 10ms	T200~T239, 40 points. 전달된 설정 값에 의해 래칭 영역으로 될 수 있습니다..	
	축적하기 위한 10ms	T240~T245, 6 points. 래칭 영역으로 고정됨.	
	축적하기 위한 1ms	T246~T249, 4 points. 래칭 영역으로 고정됨.	

타이머 기능:

1ms, 10ms 와 100ms 는 타이머의 단위이며 계산 방법은 각 단위를 세어 내는 것입니다.. 출력 코일은 현재 각 타이머가 동일하게 설정되면 On 으로 됩니다. K 설정은 10 진수를 뜻합니다.. D 데이터 레지스터는 아래와 같은 설정으로 사용될 수 있습니다..

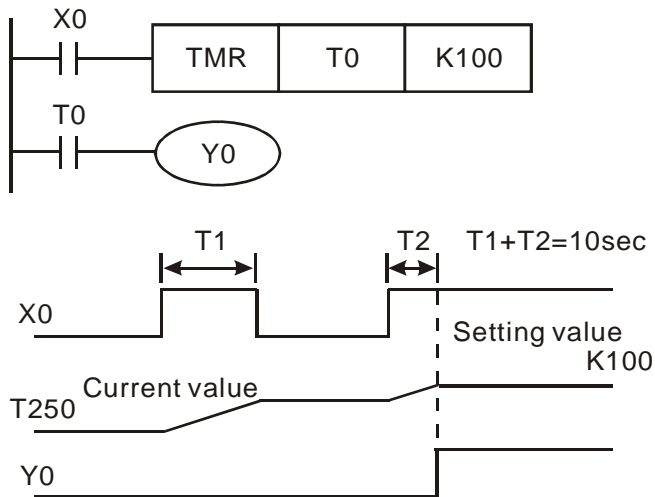
타이머의 정확한 시간 설정 = 타이머 단위 \* 설정값

아래는 타이머를 세가지 유형으로 나누어 표현했다.

### 1. 일반적인 타이머:

**ES, SA 모델 시리즈** : 해당 모델의 일반적인 타이머는 **END** 명령이 실행될 때 한 번 셴합니다.. 출력 코일에서는 **TMR** 명령이 실행되면 타이머는 **On** 으로 설정됩니다.

**EH 모델 시리즈** : 해당 모델의 일반적인 타이머는 **TMR** 명령이 실행될 때 한 번 셴합니다.. 출력 코일에서는 **TMR** 명령이 실행되면 타이머가 **On** 으로 설정됩니다.

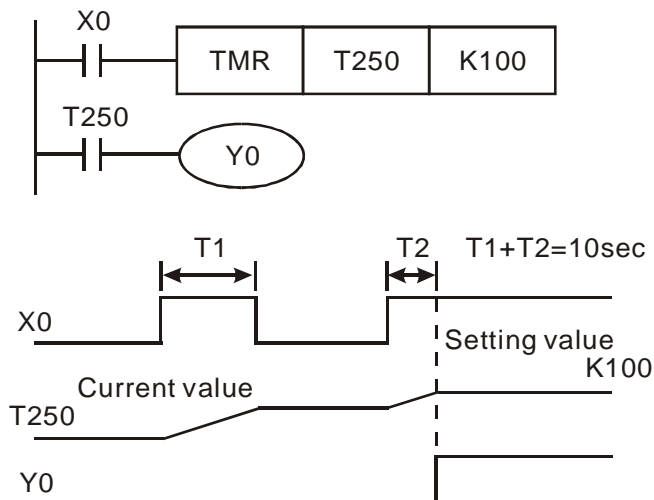


- X0=On 일 때, 타이머는 100ms 와 함께 T0 를 셴니다. 현재 타이머의 값이 설정과 동일하면(K100), 출력 코일은 T0=On 이 됩니다.
- X0=Off 이거나 전원이 내려가면, T0 타이머는 0 으로 초기화되고 T0 출력 코일은 Off 로 설정될 것입니다..

### 2. 누적 타이머:

**ES, SA 모델 시리즈** : 해당 모델의 일반적인 타이머는 **END** 명령이 실행될 때 한 번 셴합니다.. 출력 코일에서는 **TMR** 명령이 실행되면 타이머는 **On** 으로 설정됩니다.

**EH 모델 시리즈** : 해당 모델의 일반적인 타이머는 **TMR** 명령이 실행될 때 한 번 더 셴합니다.. 출력 코일에서는 **TMR** 명령이 실행되면 타이머는 **On** 으로 설정됩니다.



- X0=On 일 때, 타이머 T250 는 100ms 와 함께 셴니다. 현재 타이머의 값이 설정값과 동일할 때 (K100) 출력 코일은 T0=On 을 나타냅니다.
- 만일 X0=Off 이거나 세는 동안 전원이 내려가면, T250 타이머는 중지되며 X0=On 이후에 세고 있던 것을 가지고 있는다.. 현재 값은 설정된 값(K100)과 동일해질때까지 세어지며 출력 코일은 T0=On 입니다..

### 3. 서브루틴의 타이머

타이머가 서브루틴에서 사용되거나 인터럽트 서브루틴에 있으면 T192~T199 타이머를 사용합니다..

**SA 모델 시리즈** : 해당 모델의 일반적인 타이머는 **END** 명령이 실행될 때 한 번 셴합니다.. 출력 코일은 **TMR** 명령이 실행되면 **On** 으로 설정될 것입니다..

EH 모델 시리즈 : 해당 모델의 일반적인 타이머는 TMR 명령이 실행될 때 한 번 썸합니다.. 출력 코일은 TMR 명령이 실행되면 On 으로 설정될 것입니다..

일반적인 타이머가 서브루틴 내에서 사용되거나 인터럽트에 삽입되는 서브 루틴일 경우 그리고 서브루틴이 실행되지 않았을 때, 타이머는 정상적으로 동작할 수 없습니다.

지시 방법의 설정: 타이머의 활동 시간 설정 = 타이머의 단위 \* 설정 값

1. 지시 상수 K: K 상수는 직접적으로 지시를 설정합니다..
2. 간접 지시 D: 간접적으로 지시를 설정할때는 D 데이터 레지스터를 사용합니다..

## 2.7 카운터의 기능과 넘버링 [C]

카운터의 번호 체계 (10 진수로 이루어짐.):

ES, EX, SS 모델:

C 상수	일반적으로 16 bits 합산	C0~C111, 112 points	총 141 포인트
	래칭으로 16 bits 합산	C112~C127, 16 points. 언제나 래칭 영역임.	
C 고속 카운터로 32bit 합/감산	1-phase 입력	C235~C238, C241, C242, C244, 7 points. 언제나 래칭 영역임.	
	2 개의 1-phase 입력	C246, C247, C249, 3 points. 언제나 래칭 영역임.	
	2-phase 입력	C251, C252, C254, 3 points. 언제나 래칭 영역임.	

SA, SX, SC 모델:

C 카운터	일반적으로 16 bits 합산	C0~C95, 96 points. 비래칭 영역으로 고정됨.	
	래칭으로 16 bits 합산	C96~C199, 104 points. 설정된 전달값에 의해 비래칭 영역으로 설정될 수 있음.	
	일반적으로 32 bits 합/감산	C200~C215, 15 points. 비래칭 영역으로 고정됨.	
	래칭으로 32 bits 합/감산	C216~C234, 19 points. 설정된 전달값에 의해 비래칭 영역으로 전환할 수 있음.	
C 고속 카운터로 32 bit 합/감산	래칭으로 1-phase 입력	C235~C242, C244, 9 points. 설정된 전달값에 의해 비래칭 영역으로 전환할 수 있음.	총 250 포인트
	래칭으로 2 개의 1-phase 입력	C246, C247, C249, 3 points. 설정된 전달값에 의해 비래칭 영역으로 전환할 수 있음.	
	래칭으로 2 개의 2-phase 입력	C251, C252, C254, 3 points. 설정된 전달값에 의해 비래칭 영역으로 전환할 수 있음.	
고속 카운터로 32 bits 합/감산 (SC 시리즈만)	래칭으로 1 개의 1-phase 입력	C243, C245, 2 points. 설정된 전달값에 의해 비래칭 영역으로 전환할 수 있음.	총 3 포인트

해당)	래칭으로 2 개의 1-phase 입력	C250, 1 point. 설정된 전달값에 의해 비래칭 영역으로 전환할 수 있음.	
-----	-------------------------	--	--

EH 모델:

C 카운터	일반적으로 16-bit 합산	C0~C99, 100 points. 전달값에 의한 래칭 영역으로의 전환이 가능함.	Total is 253 points
	래칭으로 16-bit 합산	C100~C199, 100 points. 전달값에 의한 비래칭 영역으로의 전환이 가능함.	
	일반적으로 32-bit 합/감산	C200~C219, 20 points. 전달값에 의한 래칭 영역으로의 전환이 가능함.	
	래칭으로 32-bit 합/감산	C220~C234, 15 points. 전달값에 의한 비래칭 영역으로의 전환이 가능함.	
C 고속 카운터로 32bit 합/감산	소프트웨어로 1 개 1-phase 입력	C235~C240, 6 points. 전달값에 의한 비래칭 영역으로의 전환이 가능함.	
	하드웨어로 1 개 1-phase 입력	C241~C244, 4 points. 전달값에 의한 비래칭 영역으로의 전환이 가능함.	
	하드웨어로 2 개 1-phase 입력	C246~C249, 4 points. 전달값에 의한 비래칭 영역으로의 전환이 가능함.	
	하드웨어로 2 개 1-phase 입력	C251~C254, 4 points. 전달값에 의한 비래칭 영역으로의 전환이 가능함.	

특징:

항목	16 bits 카운터	32 bits 카운터	
유형	일반	일반	고속
계산 방향	합산	합/감산	
설정 범위	0~32,767	-2,147,483,648~+2,147,483,647	
지시 상수	K 상수 또는 D 데이터 레지스터	K 상수 또는 D 데이터 레지스터 (2 개를 지시함)	
현재 변경 값	카운터는 설정이 완료되면 멈춘다.	카운터는 설정이 완료되면 계산한 값을 저장합니다..	
출력 상수	계산 설정이 완료되면 On 으로 접속하며 래칭이 됩니다.	합산 설정이 완료되면 On 으로 설정되며 래칭이 됩니다. 감산 설정이 완료되면, Off 로 재설정되어 접속합니다..	
재설정 동작	RST 명령이 실행되고 Off 로	재설정되어 접속하면 현재 값은 0 으로 재설정됩니다.	
현재 레지스터	16 bits	32 bits	
상수 동작	검사 후에 함께 동작합니다..	검사 후에 함께 동작합니다..	계산이 완료되면 즉시 동작합니다.. 검사하는 동안에는 아무런 관련이 없습니다.

기능:

카운터의 펄스 입력 신호가 Off 에서 On 으로 설정되면, 카운터의 현재 값은 동일하게 설정되며 출력 코일은

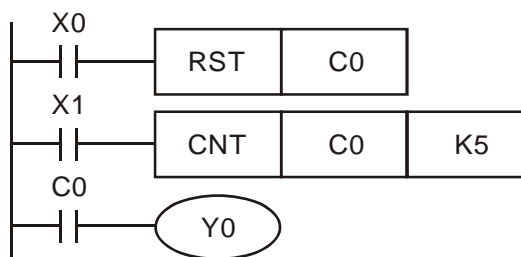
On 이 됩니다. 설정값은 10 진수로 이루어지며 D 데이터 레지스터 또한 같은 설정을 사용하게 됩니다.

### 16-bit 카운터 C0~C199:

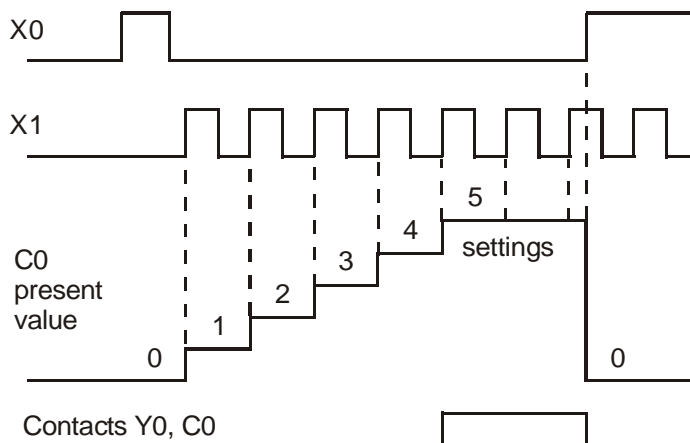
1. 16-bit 카운터의 설정 범위는 K0~K32,767 입니다.. (K0는 K1 과 같습니다.) 출력 접속으로 처음 계산 시 즉시 On 으로 설정될 것입니다..
2. 일반적인 카운터는 PLC 가 전원을 손실하면 지워진다. 만일 카운터가 래칭이면, 전원을 손실하기 전 값을 기억하고 있을것이며 전원을 상실하기 전 전원이 들어온 상태의 계산을 가지고 있을것입니다..
3. 만일 MOV 명령을 사용하면, WPLSoft 나 HPP는 C0 보다 더 크게 설정된 값을 보낼것이고 다음번에 X1 레지스터가 Off 에서 On 으로 설정되면 C0 카운터는 On 으로 접속할 것이며 현재 값은 설정된 값과 같은 값으로 설정될 것입니다..
4. 카운터를 간접적으로 설정할 때 K 상수나 D 레지스터를 사용합니다.. (다음과 같은 특정 데이터 레지스터 D1000~D1999 는 포함되지 않음.)
5. 만일 K 상수로 설정하게되면, 명백하게 숫자로만 가능하나 D 데이터 레지스터 설정의 경우 완전한 숫자로만 가능합니다.. 다음의 수는 카운터로 32,767 에서 -32,768 까지 합산합니다..

예:

```
LD    X0
RST   C0
LD    X1
CNT   C0 K5
LD    C0
OUT   Y0
```



1. X0=On 일 때, RST 명령이 실행되면, C0 는 0 으로 재설정되고 출력 상수는 Off 로 재설정 됩니다.
2. X1 이 Off 에서 On 으로 설정되면, 카운터는 한단계 합산합니다.. (1 추가).
3. C0 카운터가 On 으로 설정된 K5, C0 상수로 설정되면, C0 상수는 On 이고 On C0 = 설정 =K5 가 됩니다. C0 는 X1 트리거 신호를 받을 수 없으며 C0 는 K5 를 기억합니다..



### 일반적인 증가/서브트랙션 32-bit 카운터 C200~C234:

1. 일반적인 32-bit 카운터의 설정 범위는 K-2,147,483,648~K2,147,483,647 입니다.. (DVP ES, EX 와 SS MPU 는 해당되지 않는다.)
2. 특정 보조 릴레이는 증가/서브트랙션 32bit 카운터를 M1200~M1234 로 나눠주는 합/감산 스위치에

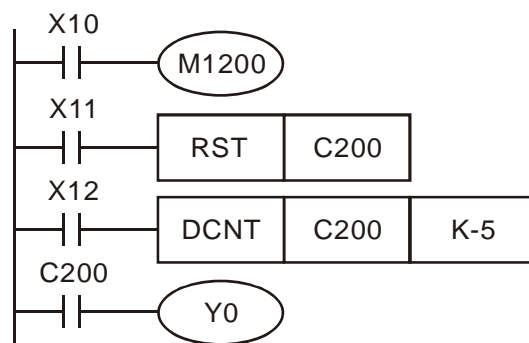


사용됩니다. 예를 들면: M1200=Off 일 때, C200 이 증가 카운터이고 M1200=On 일 때, C200 이 서브트랜잭션 카운터입니다..

3. K 상수나 D 데이터 레지스터로 설정할 수 있으며 양/음수 또한 설정할 수 있습니다..(특정 데이터 레지스터인 D1000~D1999 는 포함되지 않는다.) 만일 D 데이터 레지스터를 사용하게 되면 두 개의 연속된 데이터 레지스터를 차지하게 될 것입니다..
4. PLC 의 전원이 나가면 일반적인 카운터는 지워진다. 만일 래칭 카운터이면 카운터는 현재 값과 상수 상태를 저장하고 전원이 나가고 다시 들어왔을 때까지의 계산한 값을 저장합니다..
5. 카운터는 2,147,483,647 에서 -2,147,483,648 까지의 수를 셉니다. 같은 방법으로, -2,147,483,648 에서부터 2,147,483,648 가 될 때까지 셉니다.

예:

```
LD    X10
OUT   M1200
LD    X11
RST   C200
LD    X12
CNT   C200 K-5
LD    C200
OUT   Y0
```



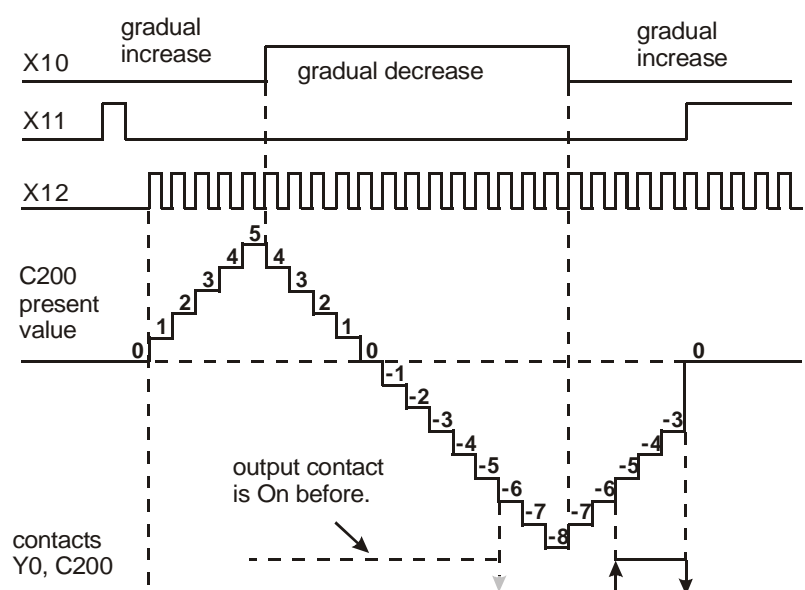
1. X10 는 M1200 으로 가 C200 의가 증가 또는 서브트랜잭션으로 갈 지를 결정합니다..

2. When X11 가 Off 에서부터 On 이 되었을 때와 RST 명령이 실행되었을 때, C200 은 0 으로 초기화 되었거나 Off 에 접속되어 있을 것입니다..

3. When X12 가 Off 에서 On 이 되면 카운터는 하나가 추가되거나(합산) 하나가 감소됩니다. (감산)

4. C200 카운터가 K-6 에서 K-5 가 되면, C200 은 Off 에서 On 으로 접속됩니다. C200 카운터가 K-5 에서 K-6 으로 되면, C200 은 On 에서 Off 로 접속될 것입니다..

5. 만일 MOV 명령을 사용하게 되면, WPLSoft 나 HPP 는 C0 에서 설정된 것보다 더 큰 값을 보낸다. register, 다음번 X1 이 Off 에서 On 으로 설정되고 C0 카운터는 On 에 접속되리 것이며 현재 값은 설정된 값과 같이 설정될 것입니다..



## 32-bit 증가/서브트랙션 고속 타운터 C235~C254:

- 32-bit 증가/서브트랙션 고속 카운터의 설정 범위는 다음과 같습니다.:  
K-2,147,483,648~K2,147,483,647.
- 32-bit 증가/서브트랙션 고속 카운터 C235~C244 의 명령은 특정 보조 릴레이인 M1235~M1244 의 On/Off 를 결정짓는데 사용됩니다. 예를 들자면  
: 만일 M1235=Off 이면, C235 는 증가이고 M1235=On 이면 C235 는 서브트랙션입니다..
- 32-bit 증가/서브트랙션 고속 타운터 C246~C254 의 명령은 특정 보조 릴레이 M1246~M1254 의 On/Off 를 결정짓는다. 예를 들자면, 만일 M1246=Off 이면, C246 은 증가이고 만일 M1246=On 이면, C246 은 서브트랙션입니다...
- K 상수나 D 데이터 레지스터는 양/음수 번호로 설정할 수 있습니다..(특정 데이터 레지스터인 D1000~D1999 는 포함되지 않는다.) 만일 D 데이터 레지스터를 사용하면, 2 개의 연결된 데이터 레지스터를 소요해야 설정할 수 있을 것입니다..
- DMOV 명령을 사용할 때, WPLSoft 나 HPP 는 어떤 고속 카운터의 설정보다 더 큰 값으로 보내지며, 같은 시간 입력 포인트 X 카운터가 Off 에서 On 이 되면, 이 접속은 어떤 변화도 가질 수 없으며 서브트랙션과 현재 값은 이를 따르게 될 것입니다..
- 카운터는 2,147,483,647 다음으로 -2,147,483,647 까지 셀 것입니다.. 같은 방법으로, 카운터는 -2,147,483,647 으로 감산한 다음 2,147,483,647 까지 셉니다.

ES / EX / SS 시리즈의 고속 카운터, 총 빈도 수: 20KHz.

입력	유형	1-phase 입력						2 개의 1-phase 입력			2-phase 입력			
		C235	C236	C237	C238	C241	C242	C244	C246	C247	C249	C251	C252	C254
X0		U/D				U/D		U/D	U	U	U	A	A	A
X1			U/D			R		R	D	D	D	B	B	B
X2				U/D			U/D			R	R		R	R
X3					U/D		R	S			S			S

U: 증가

A: A phase 입력

S: 입력 시작

D: 감소

B: B phase 입력

R: 제거 입력

입력 포인트 X0 과 X1 은 고속 타운터와 같은곳에서 사용할 수 있으며 1-phase 는 20KHz 까지 올릴 수 있습니다.. 그러나 고속 카운터의 총 주파수 대역폭은 각 X0 와 X1 고속 카운터의 20KHz 또는 더 적게 제한됩니다. 그래서, 빈번하게 발생하는 고속 카운터 AB 퓨즈의 입력은 5KHz 인 것입니다.. X2 와 X3 포인트가 입력되는 동안, 고속 카운터와 1 퓨즈는 10KHz 로 올릴 수 있습니다..

ES 시리즈 모델에서 총 사용은 네번 보다는 더 적고 DHSCS 와 DHSCR 명령 모두 일 것입니다..

SA, SX, SC 시리즈의 고속 카운터의, 총 빈도 수: 40KHz.

SC 시리즈 MPU 에서만 사용되는 3 개의 새로운 고속 카운터 C243,C245, 및 C250 가 있습니다.. 입력 포인트는 고속 카운터으로서 X10(C243)과 X11(C245)사용될 수 있고 1 단계 1 은 100KHz 개인적으로 또는 1 단계 2 입력 (X10,X11)까지 가장 높은 입력 주파수에 입력했다. 다른 입력 점 C250 은 100KHz 까지 가장 높은 입력 주파수에 있습니다.. X10~X11 을 위해 높은 속도 카운터의 총 주파수 대역폭은 130KHz 입니다..

타입	1-phase 입력	1-phase 2 입력	2-phase 입력
----	------------	--------------	------------

입력	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C249	C250	C251	C252	C254
X0	U/D						U/D			U/D		U	U	U		A	A	A
X1		U/D					R			R		D	D	D		B	B	B
X2			U/D					U/D					R	R			R	R
X3				U/D				R		S				S				S
X4					U/D													
X5						U/D												
X10									U/D						U			
X11											U/D				D			

U: 증가

A: A phase 입력

S: 시작 입력

D: 감소

B: B phase 입력

R: 제거 입력

1. 입력 점 X0 과 X1 은 20KHz 까지 고속 카운터와 1-phase 로 사용될 수 있습니다.. 그러나 X0 과 X1 모두를 위해 hig 고속 카운터의 대역폭은 20KHz 보다 적게 제한됩니다. 만약 AB 푸즈가 입력 신호라면, 고속 카운터에서 빈번하게 발생하는 4 개의 주파수 대역폭이 소요됩니다. 그래서, 5KHz 에 대한 고속 카운터 AB 푸즈는 빈번하게 발생합니다.. 입력 포인트 X2, X3, X4, 그리고 X5 는 고속 카운터이며 1 푸즈는 10KHz 보다 우월합니다..

2. 입력 포인트 X5 는 두가지 기능을 가지고 있습니다..

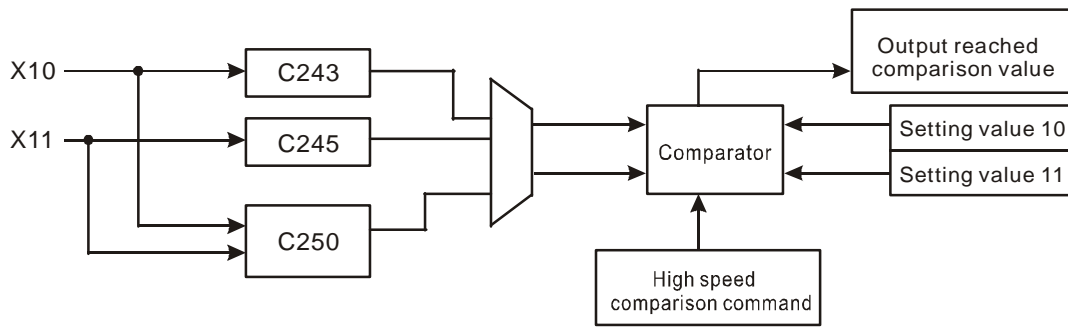
M1260=Off 일 때, C240 은 일반적인 U/D 고속 카운터를 가집니다.

When M1260=On 와 DCNT 에서의 C240 활동 시 X5 는 C235~C239 의 재설정 신호로 사용된다. 고속 카운터 C240 은 여전히 X5 의 입력 신호를 가집니다.

SA/SX/S 모델 시리즈에선 6 번보다 적거나 같게 총 사용되고 DHSCS and DHSCR 명령에 총 사용됩니다. Once DHSCS 명령이 교착되면, 계기에 표시된 고속 카운터는 DHSCS, DHSCR, 와 DHSZ 의 명령을 수행하지 않을것입니다..

SC 시리즈 MPU 에서만 사용되는 X10~X11 고속 카운터 기능,

1. When X10 과 X11 에 1 푸즈 1 입력에 사용되거나 1 푸즈 2 입력에 사용될 때, 최고 대역폭인 100KHz 를 감돈다.
2. X10 과 X11 는 상승 에지나 하강 에지 모드 같은 설정을 하기위한 선택입니다.. X10 모드는 D1167 에 의한 X11 이 실행되는동안 D1166 을 결정합니다.. K0: 상승 에지 모드; K1: 하강 에지 모드; K2: 상승에지와 하강에지 모두 (X10 에서만 지원됨.).
3. C245 가 M1245 OFF/ON 에 의해 결정되는 동안 C243 는 M1243 OFF/ON 에 의 해 결정지어지는 함/감산입니다.. 함산과 감산은 동시에 일어날 수 없습니다.
4. 각 C243 또는 C245, C250 의 사용 시 사용될 수 없습니다. C250 와 C243 그리고 C245 를 사용할 때는 사용될 수 없습니다.
5. 고속 카운터의 구조(C243, C245, C250)와 비교 명령



6. 고속 카운터 C243, C245, C250 와 각 비교 기능에 대한 설명

- a) DHSCS 와 DHSCR 명령이 발생하면 고속 카운터 C243, C245, 그리고 C250 에 사용됩니다. 명령 사용 시간은 합계와 같이 2 까지 설계된다. 예를 들어, 만일 비교 명령 DHSCS D0 C243 Y10 이 사용되면, 나머지 비교 명령들은 각 DHSCR D2 C243 Y10 나 DHSCS D4 C245 Y0.에 사용된다.
- b) DHSZ 명령이 고속 카운터 C243, C245, 그리고 C250 가 사용될 때, 한 번만 사용할 수 있도록 설계되어져 있음.
- c) 이 명령은 고속 카운터를 사용한 SA/SX 시리즈는 6 과 같은 사용시간을 남긴다. 그러나 SC 시리즈는, C243, C245, 그리고 C250 인 세 새로운 고속 카운터의 사용 시간을 증가시킬 것입니다..
- d) 만일 고속 응답 출력기에서 DHSCS 명령 고속 출력 장치가 필요하면 출력 장치와 함께 Y10 또는 Y11 과 함께 사용하기를 추천합니다.. 만일 Y10 과 Y11 을 사용하지 않으면, 출력하는데 걸리는 검색시간이 지연되거나 데이터가 0 으로 지워진다. 예를 들어, 인터럽트의 I0x0 으로 설정되어 있을 때, 인터럽트 번호는 C243, C245, 그리고 C250 로 일치합니다.: C243 는 I020 으로, C245 는 I040 으로, 그리고 C250 는 I060 으로 응답합니다...
- e) DHSCR 는 고속 카운터의 출력 장치 명령입니다.. 이 명령은 카운터의 값을 0 으로 재설정할 수 있으나 카운터가 지워지는것과 같은 결과를 초래합니다.. 고속 카운터 사용 시, DHSCR 는 C243, C245, 그리고 C250 만 사용합니다..

C235~C240 는 EH 시리즈의 고속 카운터입니다.. 이것은 프로그램에서 1-phase 로 중단된 고속 카운터(10KHz)이며 총 빈도수는 20KHz 입니다.. C241~ C254 는 HHSC 라고 불리는 하드웨어 고속 카운터로 HHSC0 와 HHSC 1 입력에 100KHz 를 올리며 HHSC2 와 HHSC3 는 30KHz 올린다.(한 개의 퓨즈와 AB 퓨즈 모두 해당).

1. HHSC0 에 C241, C246, C251 할당
2. HHSC1 에 C242, C247, C252 할당
3. HHSC2 에 C243, C248, C253 할당
4. HHSC3 에 C244, C249, C254 할당

- a) 각 HHSC 는 번호를 한 번만 사용할 수 있으며 이 번호는 DCNT 명령을 지시하는데 사용합니다..
- b) 각 HHSC 의 세 가지 모드:

- Pulse/Direction 모드라 불리며 phase 를 입력합니다..
- CW/CCW 모드라 불리며 phase 두 개를 입력합니다..
- AB phase 모드라 불리며 phase 를 입력합니다..

유형 입력	1-phase 로 중단된 프로그램 고속 카운터						하드웨어 고속 카운터											
	1-phase 입력						1-phase 입력				2 개의 1-phase 입력				2-phase 입력			
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C246	C247	C248	C249	C251	C252	C253	C254
X0	U/D						U/D				U				A			
X1		U/D									D				B			
X2			U/D				R				R				R			
X3				U/D			S				S				S			
X4					U/D			U/D				U				A		
X5						U/D						D				B		
X6								R				R				R		
X7								S				S				S		
X10									U/D				U				A	
X11													D				B	
X12									R				R				R	
X13									S				S				S	
X14										U/D				U				A
X15														D				B
X16										R				R				R
X17										S				S				S

U: 증가

A: A phase 입력

S: 시작 입력

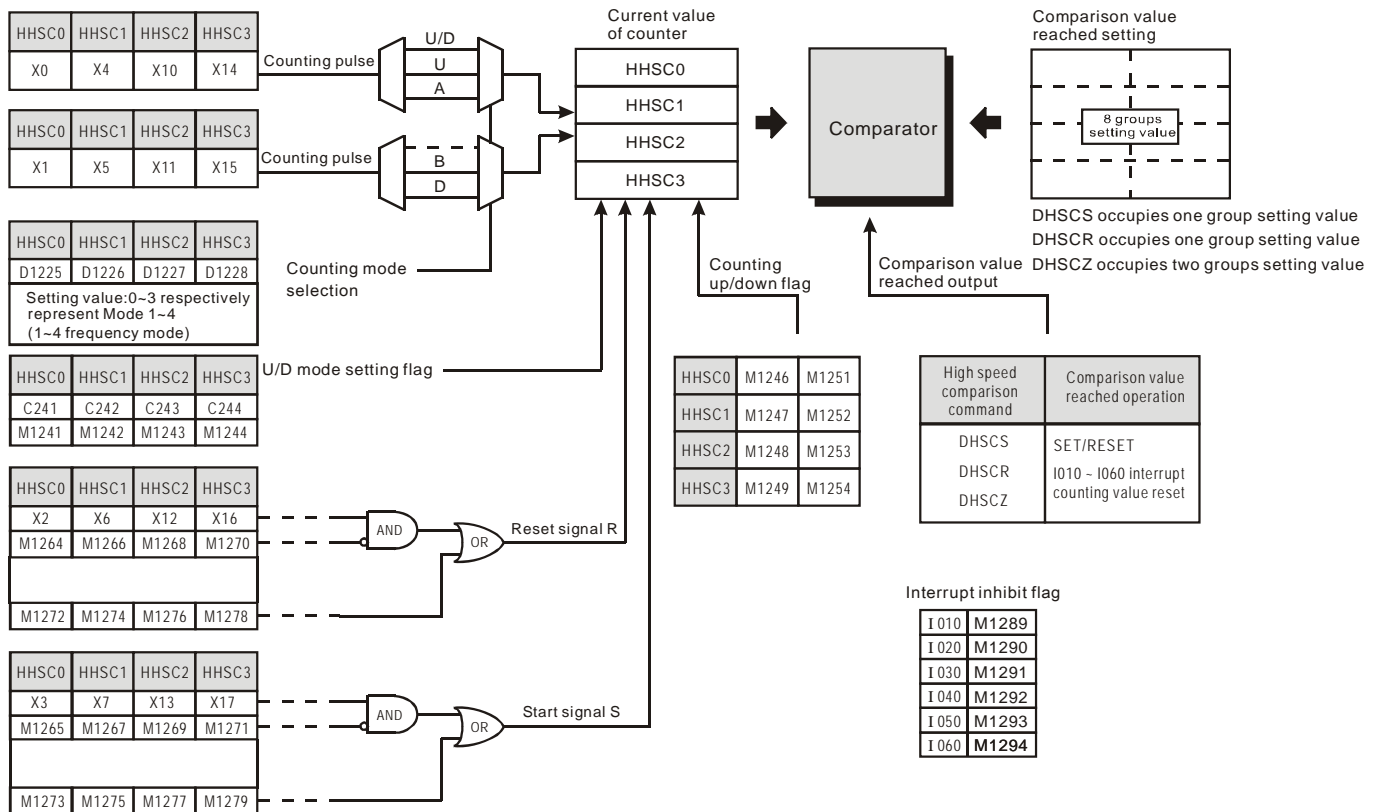
D: 감소

B: B phase 입력

R: 제거 입력

### 1. 하드웨어 고속 카운터 시스템의 구조:

- HHSC0~3 은 재시작 신호와 시작 신호가 있는 외부 입력입니다.. 또한 재시작 신호는 M, M1272 (HHSC0), M1274 (HHSC1), M1276 (HHSC2) 그리고 M1278 (HHSC3)통해 별도로 설정할 수 있습니다... 그리고 시작 신호는 M, M1273 (HHSC0), M1275 (HHSC1), M1277 (HHSC2) 그리고 M1279 (HHSC3)을 통해 별도로 설정할 수 있습니다..
- 만일 고속 카운터 사용 시 외부 입력 제어 신호인 R 과 S 가 사용되지 않으면입력 신호 기능은 M1264/ M1266/ M1268/ M1270 그리고 M1265 / M1267/ M1269/ M1271 가 True 설정되면서 단히게 됩니다. 이 외부 입력은 일반적인 입력과 유사하게 사용할 수 있습니다..
- 고속 카운터의 특정 M 사용 시 START 와 TRSET 의 입력과 활동을 제어하며 검색시간과 함께 영향을 받을것입니다..

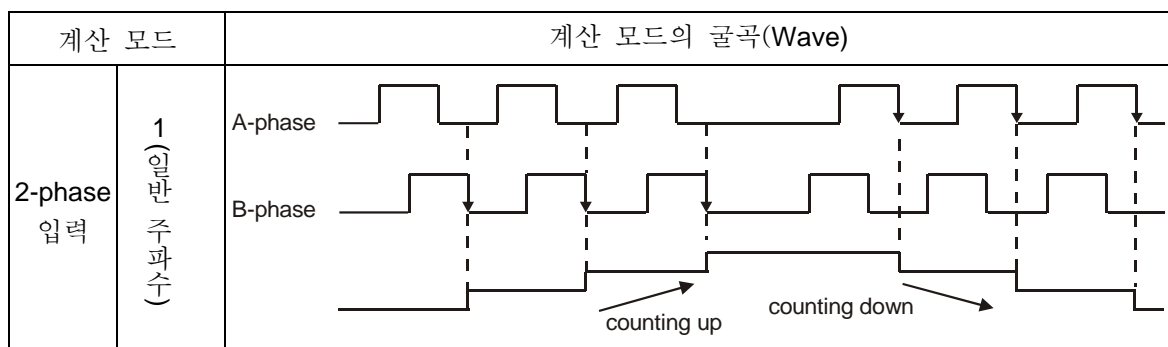


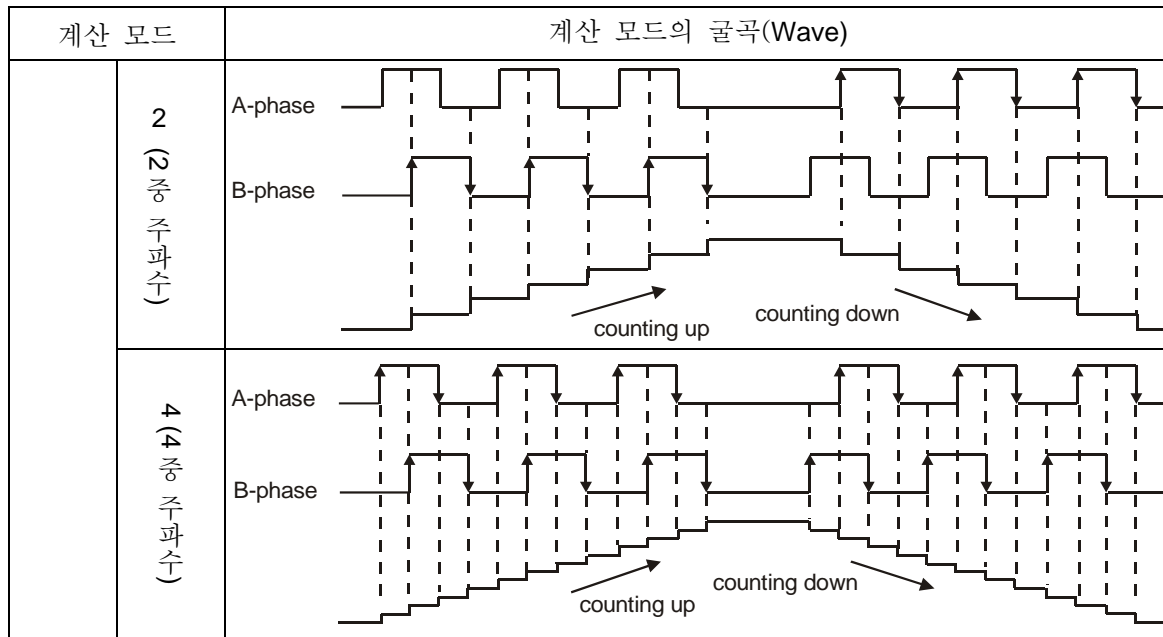
## 2. 선택 모드 계산

고속 카운터 ES, EX, SS, SA, SX, SC 는 2-phase 빈번하게 2 중으로 선택하는 입력 계산 모드 내 특정 D1022 를 사용합니다.. D1022 의 내용은 PLC 스위치가 STOP 에서 RUN 으로 되었을 때의 첫번째 검사 시간을 가집니다. (ES, EX, SS 시리즈 MPU (V5.5 와 그 이상 버전에서 사용 가능) 는 아래 기능을 지원합니다..

장치 번호	기능
D1022	카운터 계산 방식을 2 중 주파수로 설정
D1022=K1	일반적인 주파수 모드
D1022=K2	2 중 주파수 모드 (공장에서 셋팅됨.)
D1022=K4	4 중 주파수 모드

2 중 주파수 모드 (↑, ↓ 는 계산 방향을 뜻함.)





EH 하드웨어 고속 카운터의 주파수(HHSC0~3)는 1에서 4가지이며 특정 D1225~D1228로 설정됩니다. 공장에선 2 중 주파수를 설정했다.

유형	특정 D(설정)	합산(+1)	감산(-1)
1-phase 입력	1(일반적인 주파수)	U/D U/D FLAG	
	2 (2 중 주파수)	U/D U/D FLAG	
1-phase 2 입력	1(일반적인 주파수)	U D	
	2 (2 중 주파수)	U D	
2-phase 2 개 입력	1(일반적인 주파수)	A B	
	2 (2 중 주파수)	A B	
	3 (3 중 주파수)	A B	
	4 (4 중 주파수)	A B	

### 3. 고속 카운터의 장치 번호와 특정 레지스터

장치번호	기능
M1150	DHSZ 명령은 비교 모드에서 멀티 그룹 설정 시 사용함을 나타냅니다.
M1151	멀티 그룹 실행 사이클이 끝나고 설정합니다..
M1152	DHSZ 명령은 주파수 제어 모드에서 사용됩니다.
M1153	실행 시 주파수가 종료됐을 때 제어

장치번호	기능
M1235 ~ M1244	C235 ~ C244 는 고속 카운터의 방향을 계산합니다.. M12□□=Off 일 때, C2□□는 합산됩니다. M12□□=On,이면 C2□□ 감산됩니다.
M1246 ~ M1249 M1251 ~ M1254	C246 ~ C249, C251 ~ C254 는 고속 장치 내 모니터 방향을 계산합니다.. C2□□ 일 때 합산하고, M12□□=Off. C2□□ 일 때 감산하고, M12□□=On.
M1264	HHSC0 에서 재설정 신호의 외부 입력 제어 접속 불가
M1265	HHSC0 에서 시작 신호의 외부 입력 제어 접속 불가
M1266	HHSC1 에서 재설정 신호의 외부 입력 제어 접속 불가
M1267	HHSC1 에서 시작 신호의 외부 입력 제어 접속 불가
M1268	HHSC2 에서 재설정 신호의 외부 입력 제어 접속 불가
M1269	HHSC2 에서 시작 신호의 외부 입력 제어 접속 불가
M1270	HHSC3 에서 재설정 신호의 외부 입력 제어 접속 불가
M1271	HHSC3 에서 시작 신호의 외부 입력 제어 접속 불가
M1272	HHSC0 에서 재설정 신호의 외부 입력 제어 접속
M1273	HHSC0 에서 시작 신호의 외부 입력 제어 접속
M1274	HHSC1 에서 재설정 신호의 외부 입력 제어 접속
M1275	HHSC1 에서 시작 신호의 외부 입력 제어 접속
M1276	HHSC2 에서 재설정 신호의 외부 입력 제어 접속
M1277	HHSC2 에서 시작 신호의 외부 입력 제어 접속
M1278	HHSC3 에서 재설정 신호의 외부 입력 제어 접속
M1279	HHSC3 에서 시작 신호의 외부 입력 제어 접속
M1289	고속 카운터 인터럽트 I010~I060 입력 불가
M1290	EH 시리즈 고속 카운터 인터럽트 I010 입력 불가
M1291	EH 시리즈 고속 카운터 인터럽트 I020 입력 불가
M1292	EH 시리즈 고속 카운터 인터럽트 I030 입력 불가
M1293	EH 시리즈 고속 카운터 인터럽트 I040 입력 불가
M1294	EH 시리즈 고속 카운터 인터럽트 I050 입력 불가
M1312	C235 입력 시작 포인트 제어
M1313	C236 입력 시작 포인트 제어
M1314	C237 입력 시작 포인트 제어
M1315	C238 입력 시작 포인트 제어
M1316	C239 입력 시작 포인트 제어
M1317	C240 입력 시작 포인트 제어
M1320	C235 재설정 입력 포인트 제어
M1321	C236 재설정 입력 포인트 제어
M1322	C237 재설정 입력 포인트 제어
M1323	C238 재설정 입력 포인트 제어
M1324	C239 재설정 입력 포인트 제어
M1325	C240 재설정 입력 포인트 제어
M1326	C235 시작/재설정 사용 가능 제어
M1327	C236 시작/재설정 사용 가능 제어
M1328	C237 시작/재설정 사용 가능 제어
M1329	C235 입력 시작 포인트 제어
M1330	C236 입력 시작 포인트 제어
M1331	C238 시작/재설정 사용 가능 제어
M1332	C239 시작/재설정 사용 가능 제어
M1333	C240 시작/재설정 사용 가능 제어



장치번호	기능
D1022	ES, EX, SS, SA, SX, SC 모델은 AB phase 카운터로 이중 주파수를 선택합니다..
D1150	멀티 그룹의 비교 항목 설정을 저장하는 레지스터
D1151	주파수 제어 모드는 비교 항목을 저장하는 레지스터
D1152	주파수 제어 모드 내 DHSZ 실행 명령은 출력 주파수 pulse 의 낮은 word 입니다..
D1153	주파수 제어 모드 내 DHSZ 실행 명령은 출력 주파수 pulse 의 낮은 word 입니다..
D1225	C241, C246 그리고 C251 계산 모드는 첫번째 카운터 설정임.
D1226	C242, C247 그리고 C252 계산 모드는 2 번째 카운터 설정임.
D1227	C243, C248 그리고 C253 계산 모드는 3 번째 그룹 카운터 설정임.
D1228	C244, C249 그리고 C254 계산 모드는 4 번째 그룹 카운터 설정임.
D1225 ~ D1228	EH 하드웨어 고속 카운터의 계산 모드 HHSC0~ HHSC3 1 로 설정되면, 일반적인 주파수입니다.. 2 로 설정되면 2 중 주파수입니다.. (공장에서 설정됨) 3 으로 설정되면 3 중 주파수이며 4 로 설정되면 4 중 주파수입니다..

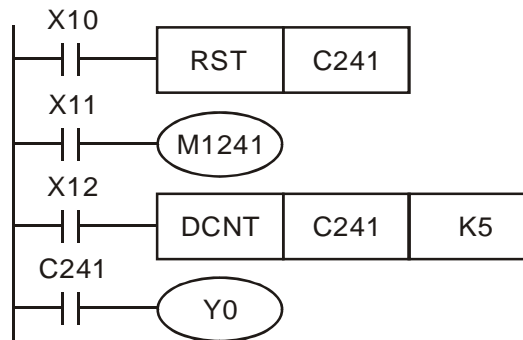
### 1-phase 입력 카운터

예:

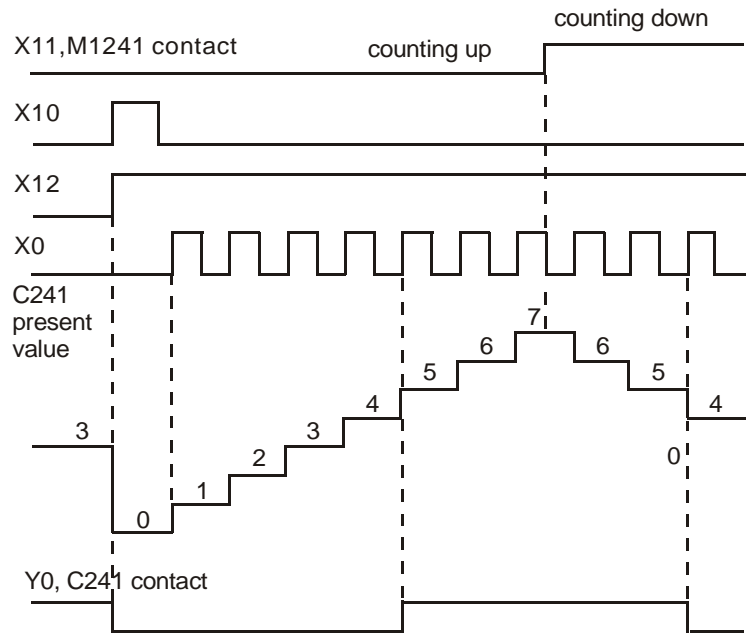
```

LD    X10
RST   C241
LD    X11
OUT   M1241
LD    X12
DCNT  C241 K5
LD    C241
OUT   Y0

```



1. X11 이 M1241 로 가서 C241 이 증가가 될 지 서브 트랜잭션이 될 지를 결정합니다..
2. X10=On 일 때와 RST 명령이 실행됐을 때, C241 은 0 으로 설정되고 출력 접속은 Off 로 재설정 됩니다.
3. X12=On 일 때, C241 은 X0 으로부터 신호를 받고 카운터는 1 합산 또는 1 감산이 이루어집니다.
4. C241 카운터가 K5 로 설정되면, C241 은 On 이 됩니다. 만일 입력 신호가 여전히 X0 이면, 계산 작업이 계속 이루어질 것입니다..



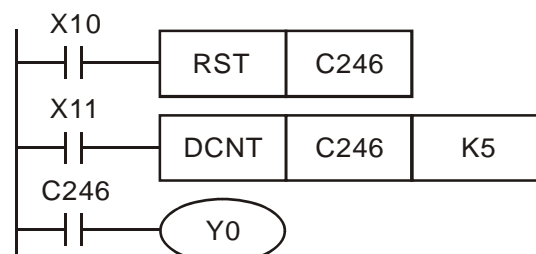
5. C241 for ES, EX, SS, SA, SX, SC 시리즈에서 사용되는 C241 은 외부 입력 재설정신호 X1 이 됩니다.
6. C241 for EH 시리즈에서 사용되는 C241 은 외부 입력 재설정 신호(X2), 시작 신호(X3)을 가집니다.
7. EH 시리즈와 외부 입력 접속 제거 신호인 C241(HHSC0)는 M1264 에 의해 사용할 수 없습니다. 외부 입력 접속 시작 신호는 M1265 에 의해 사용할 수 없습니다.
8. EH 시리즈인 내부 입력 접속 제거 신호인 C241 (HHSC0)는 M1272 에 의해 사용할 수 없습니다. 내부 입력 접속 시작 신호는 M1273 에 의해 사용할 수 없습니다.
9. EH 시리즈의 계산 모드 (일반적인 주파수나 이중 주파수) 의 C246 (HHSC0)는 D1225 에 의해 설정됩니다. 공장으로부터 이중 주파수로 설정되어 나옵니다.

2 개 입력 1-phase 고속 카운터:

예:

```

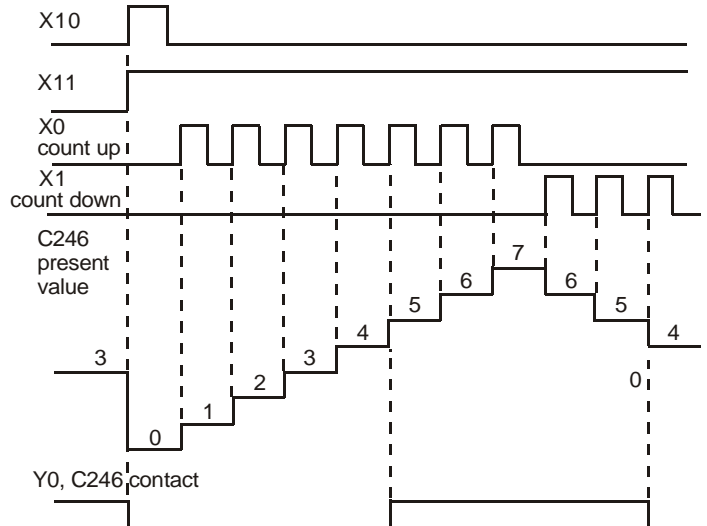
LD    X10
RST   C246
LD    X11
DCNT  C246 K5
LD    C246
OUT   Y0
    
```



1. X10=On 일 때와 RST 명령이 실행되면, clear C246 이 0 으로 지워지고 출력 접속은 Off 로 재설정됩니다.

2. X11=On 일 때와, C246 이 X0 입력 터미널로부터 계산된 신호를 받으면 카운터는 1 합산 될 것이며 X1 입력 터미널에서 받은 계산 신호를 받으면 카운터는 1 감산 될 것입니다..

3. C246 가 K5 로 설정되면, C246 은 On 으로 설정됩니다. C246 이 On 이 된 후, 카운터에 pulse 입력이 있으면 C246 은 계산을 계속 저장할 것입니다..



4. EH 시리즈의 C246 은 외부 입력 재설정 신호 X2 이거나 시작 신호 X3 입니다..

5. EH 시리즈의 C246 (HHSC0)은 D1225 에 의해 일반 주파수나 이중 주파수로 설정됩니다. 공장에서도부터 이중 주파수로 설정되어 나옵니다.

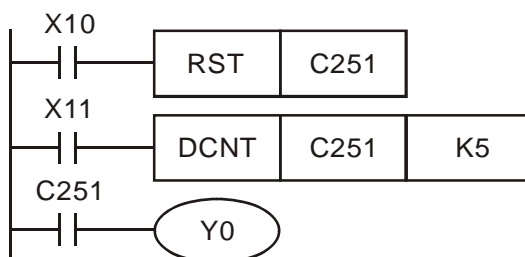
6. EH 시리즈의 외부 입력 접속 제거 신호(R)인 C246 (HHSC0)은 M1264 에 의해 사용될 수 없습니다. 외부 입력 접속 시작 신호(S)는 M1265 에 의해 사용될 수 없습니다.

7. EH 시리즈의 내부 입력 접속 제거 신호(R)인 C246 (HHSC0)은 M1272 에 의해 사용될 수 없습니다. 내부 입력 접속 시작신호(S)는 M1273 에 의해 사용될 수 없습니다.

2-phase AB 입력 고속 카운터:

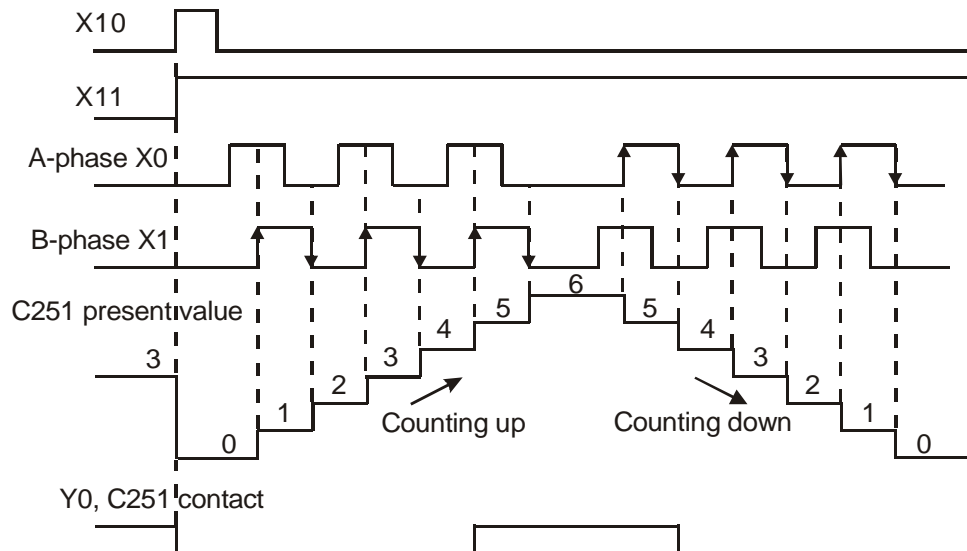
예:

```
LD    X10
RST   C251
LD    X11
DCNT  C251 K5
LD    C251
OUT   Y0
```

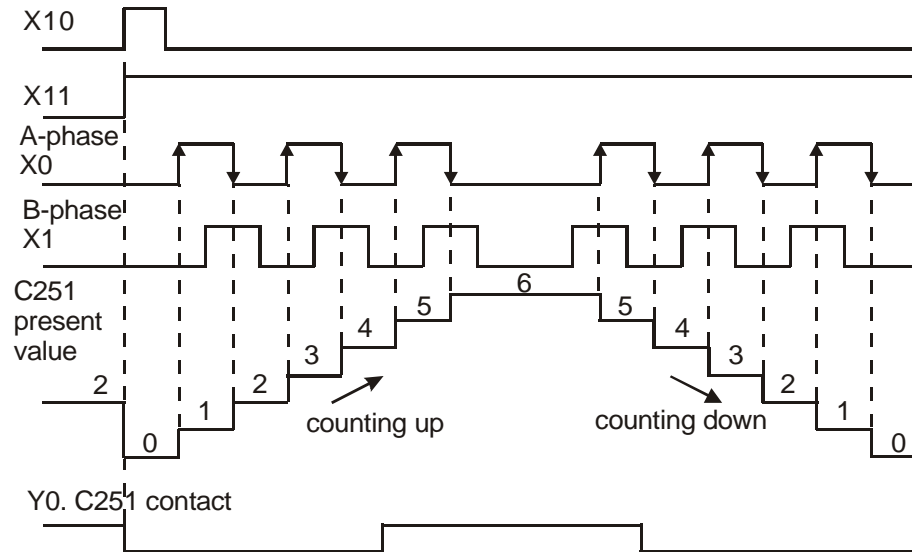


1. X11=On 일 때, RST 명령이 실행되고 C251 이 0 으로 재설정되며 출력 접속이 Off 로 재설정됩니다.
2. X12=On 일 때, C251 에서 X0 입력 터미널의 A phase 계산 신호를 받고 X1 입력 터미널의 B phase 를 받아 1 추가(합산)을 실행하거나 1 공제(감산)이 실행됩니다. EH 시리즈는 계산 모드의 다른 주파수를 설정할 수 있습니다..
3. C251 카운터가 K5 로 설정되면, C251 은 On 으로 접속됩니다. C251 이 On 이 되고난 후, 카운터 pulse 가 입력되면, C251 은 계속 계산될 것입니다..
4. ES, EX, SS, SA, SX, SC 시리즈에서, D1022(계산 설정 모드)로 일반 주파수, 이중 주파수나 4 중 주파수로 설정할 수 있습니다.. 공장에서부터는 이중 주파수로 설정되어 나옵니다.
5. EH 시리즈의 C251 은 외부 입력 재설정 신호 X2 와 시작 신호 X3 을 갖는다.
6. EH 시리즈의 C251(HHSC0)의 계산 모드(일반 주파수, 이중 주파수, 3 중 주파수, 4 중 주파수)는 D1225 에 의해 설정될 수 있습니다.. 공장에서부터는 이중 주파수로 설정되어 나옵니다.
7. EH 시리즈의 외부 입력 접속 제거 신호인 C246 (HHSC0)은 M1264 에 의해 사용될 수 없습니다. 외부 입력 접속 시작 신호는 M1265 에 의해 사용될 수 없습니다.
8. EH 시리즈의 내부 입력 접속 제거 신호인 C246 (HHSC0)은 M1272 에 의해 사용될 수 없습니다. 내부 입력 접속 시작 신호는 M1273 에 의해 사용될 수 없습니다.

ES, EX, SS, SA, SX, SC 시리즈:



EH 시리즈:(이중 주파수)



## 2.8 레지스터 번호와 기능 [D], [E], [F]

### 2.8.1 데이터 레지스터 [D]

이 레지스터는 숫자 데이터를 저장하는데 사용되며 저장할 수 있는 데이터 길이는 16-bit (-32,768~+32,767)입니다.. 왼쪽 비트는 신호 비트입니다.. 두 16-bit 레지스터를 합하여 32-bit 레지스터로 만들 수 있습니다.. (The number for each 32-bit 레지스터의 각 번호는 (D0, D1), (D2, D3)이 될 것이며 낮은 비트보다 한단계 더 올라간 수가 될 것입니다..) 왼쪽 비트는 신호 비트이며 저장 공간은 -2,147,483,648~+2,147,483,647 입니다..

ES, EX, SS 모델:

D 데이터 레지스터	일반적인 사용	D0~D407, 408 points	총 744 포인트
	래칭 시 사용 *	D408~D599, 192 points. (래칭 영역으로 고정됨.)	
	특정 상황 시 사용	D1000~D1143, 144 points. (몇몇은 래칭 영역임)	
	E, F 인덱스 레지스터	E(=D1028), F(=D1029), 2 points	

SA, SX, SC 모델:

D 데이터 레지스터	일반적인 사용	D0~D199, 200 points. (언래칭 영역으로 고정됨.)	총 5000 포인트
	래칭 시 사용	D200~D999, D2000~D4999, 3800 points. (설정된 전달값에 의해 언래칭 영역으로 사용될 수 있음.)	
	특정 상황 시 사용	D1000~D1999, 1000 points. (몇몇은 래칭 영역임)	
	E, F 인덱스 레지스터	E0~E3, F0~F3, 8 points	
파일 레지스터		K0~K1599, MPU 1600 points. (래칭 영역으로 고정됨.)	1600 포인트

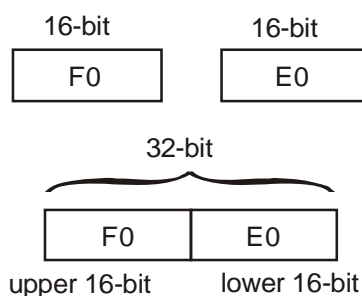
EH 모델:

D 데이터 레지스터	일반적인 사용	D0~D199, 200 points. 설정된 전달값에 의해 래칭 영역이 될 수 있습니다..	총 10000 포인트
	래칭 시 사용	D200~D999, D2000~D9999, 8800points. 설정된 저장값에 의해 비래칭 영역이 될 수 있습니다..	
	특정 상황 시 사용	D1000~D1999, 1000 points. 몇몇은 래칭입니다..	
	E,F 인덱스 레지스터	E0~E7, F0~F7, 16 points.	
파일 레지스터		K0~K9999, MPU 는 10000 포인트임. (래칭 영역으로 고정됨.)	10000 포인트

아래는 성격별로 정렬된 5 가지 레지스터 타입을 나타냅니다.:

1. 일반적인 레지스터 : PLC 스위치가 RUN 에서 STOP 으로 변경되거나 전원이 Off 가 되면 레지스터 내 데이터는 0 으로 지워질 것입니다.. PLC 스위치가 Run 에서 STOP 으로 변경되었을 때 M1033=On 이면, 데이터는 지워지지 않을 것이나 전원이 Off 가 되었을 땐 0 으로 초기화 되었을 것입니다..
2. 래칭 레지스터 : PLC 전원이 Off 가 되면 래칭 레지스터의 데이터는 지워지지 않을것입니다.. 만일 레지스터 내 데이터가 지워지길 바라면 RST 나 ZRST 명령을 사용해야 합니다..
3. 특정 레지스터 : 각 특정 레지스터는 특정한 정의나 목적을 가지고 있습니다.. 시스템 상태 저장, 에러 메시지, 모니터 상태 등을 저장하는데 사용됩니다. 2.11 에 나온 자세한 설명을 참고하기 바랍니다.
4. [E], [F] 인덱스 레지스터 : 인덱스 레지스터는 16-bit 레지스터입니다.. ES, EX, SS 모델에서는 E 와 F, 두 포인트를 가지고 있습니다.. SA, SX, SC 모델에서는 E0~E3 와 F0~F3, 8 개의 포인트를 가지고 있습니다.. EH 모델에서는 E0~E7 와 F0~F7, 16 포인트를 가지고 있습니다.. 만일 32 비트 인덱스 레지스터를 사용하고 싶다면, E 를 나타내야 할 것이며 그와 동시에 F 는 사용할 수 없습니다.
5. 파일 레지스터 : SA, SX, SC MPU 의 1600 파일 레지스터(K0~K1599) 와 EH MPU 의 10000 파일 레지스터 (K0~K9,999)가 있습니다.. 파일 레지스터는 정확한 장치 번호가 없으며 파일 레지스터의 주변장치 HPP 나 WPLSoft 의 API 148 MEMR, API 149 MEMW 읽기/쓰기 명령을 실행해야 합니다..

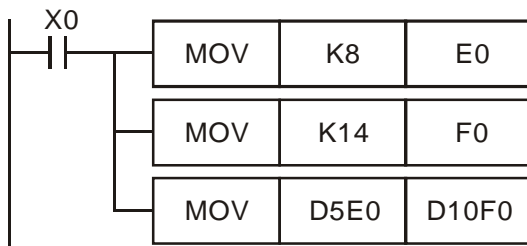
### 2.8.2 인덱스 레지스터 [E], [F]



E, F 인덱스 레지스터는 일반적인 데이터 레지스터와 같이 16-bit 의 데이터 레지스터이며 읽기/쓰기가 가능합니다.. 이 레지스터는 32-bit 레지스터처럼 사용됩니다. 그러나 이 레지스터는 E 를 나타내야 하며 그와 동시에 F 는 사용할 수 없습니다. 만약 그렇지 않으면, 데이터는 에러가 될 것입니다.. (DMOV P K0 E 명령을 사용하기를 권하며 E 와 F 는 전원이 들어왔을 때 0 으로 초기화 하는 것이 좋습니다..)

32-bit 레지스터 사용 시 E 와 F 를 합친 것은 아래와 같습니다.:

(F0, E0) , (F1, E1) (F2, E2) ....(F7, E7)



X0=On 그리고 E0=8, F0=14, D5E0=D(5+8)=D13, D10F0=D(10+14)=D24 일 때, D13 에 있는 내용은 D24 로 이동해 갈 것입니다..

인덱스 레지스터의 기능은 일반 연산자와 같습니다. 이동 또는 비교에 사용되며 byte 장치 (KnX, KnY, KnM, KnS, T, C, D)의 인덱스와 bit 장치(X, Y, M, S)의 인덱스에 사용됩니다. ES, EX, SS, SA, SX, SC 시리즈에서, 상수(K, H)는 사용할 수 없습니다. 그러나 EH 시리즈에서는 상수 (K,H)를 사용할 수 있습니다..

ES, EX, SS 모델: 2 개의 E0, F0 points

SA, SX, SC 모델: E0~E3, F0~F3, 총 8 개 포인트

EH 모델: E0~E7, F0~F7, 총 16 포인트

※ 몇몇의 명령은 인덱스 기능을 가지고 있지 않으므로 5.4 에 나와있는 E 와 F 인덱스 레지스터 사용법을 참고하기 바랍니다.

※ WPLSoft 의 명령 모드 사용 시 인덱스 레지스터에서 (K,H)상수 사용은 “@”기호가 반드시 사용되어야 합니다..

예: "MOV K10@E0 D0F0"

※ 연산자를 인덱스 하기 위한 E, F 인덱스 레지스터의 사용 시 특정 레지스터(D1000~D1999)와 특정 보조 릴레이(M1000~M1999)의 인덱스 범위를 초과해선 안됩니다.

### 2.8.3 파일 레지스터의 기능과 특성

SA, SX, SC, EH 시리즈는 PLC 의 전원이 들어올 때 아래를 체크할 것입니다..

1. M1101 (만일 파일 레지스터 시작 기능이라면)
2. D1101 (SA, SX, SC 시리즈(K0~K1599)의 파일 레지스터의 시작 번호라면, EH 시리즈는 K0~K9999 입니다..)
3. D1102 (SA, SX, SC 시리즈의 읽기 항목 번호는 K0~K1600 이고 EH 시리즈는 K0~K10000 입니다..)
4. D1103 (읽기 데이터의 저장 주소는 D 파일 레지스터에 설계된 시작 주소입니다.(K2000~K9999). 자동으로 레지스터에 설계된 파일이 이동하게 되는 것을 결정하는데 사용됩니다.)

노트:

- SA, SX, SC 모델의 D1101 이 1600 보다 크지 않거나, EH 모델의 D1101 이 8000 보다 크지 않거나 D1103 의 값이 2000 보다 작거나 9999 보다 크면, 파일 레지스터에서 읽은 D 데이터 레지스터는 실행되지 않을것입니다..
- 파일 레지스터에서 데이터 레지스터로 읽기 전용 파일이 시작 시 실행되지 않았다면, PLC 는 파일 레지스터의 주소 내 값 읽기를 중지하거나 D 데이터 레지스터의 사용 범위가 초과될 것입니다..
- SA, SX, SC 모델에는 1600 개의 파일 레지스터가 있고 EH 모델에는 10000 개의 파일 레지스터가 있습니다.. 파일 레지스터에는 활동 번호가 없으며 API 147 MEMR, API 148 MEMW 명령을 사용하거나

파일 레지스터에서 HPP02 그리고 WPLSoft 장치의 읽기/쓰기를 실행해야 합니다..

- 만일 파일 레지스터의 읽기 주소 사용 범위를 초과하게 되면, 데이터는 0 이 될 것입니다..

## 2.9 함유 레벨 포인터[N], 포인터[P], 인터럽트 포인터 [I]

포인터	N	중첩된 주 제어 사용 시	N0~N7, 8 points	중첩된 주 제어 포인트를 제어함.
	P	CJ, CALL 명령에서 사용 시	P0~P63, 64 points	CJ, CALL 포인터의 위치
	I	타이머 입력 인터럽트	I6□□, 1 point (□□=10~99ms, time base=1ms) (5.7 버전에 해당)	인터럽트 서브루틴의 포인터 위치
		외부 입력 인터럽트	I001, I101, I201, I301, 4 points	
		통신 입력 인터럽트	I150, 1 point	

SA, SX, SC 모델:

포인터	N	중첩된 주 제어 사용 시	N0~N7, 8 points	중첩된 주 제어 포인트를 제어함.
	P	CJ, CALL 명령에서 사용 시	P0~P255, 256 points	CJ, CALL 포인터의 위치
	I	외부 입력 인터럽트	I001, I101, I201, I301, I401, I501, total is 6 points	서브루틴 인터럽트 포인터 위치
		시간 입력 인터럽트	I6□□, I7□□, 2 points (□□=10~99ms, time base=1ms)	
		도달된 고속 입력 카운터 인터럽트	I010, I020, I030, I040, I050, I060, 6 points	
		통신 입력 인터럽트	I150, 1 point	

노트: 6 쌍의 인터럽트 번호 (I001, I010), (I101, I1020), (I1201, I030), (I301, I040), (I401, I050), 그리고 (I501, I060)번호가 있습니다.. PLC 내 프로그램에 쓸 때, 각 인터럽트 번호 중 한 쌍을 사용해야 합니다.. 같은 번호 두 개를 같은 시간에 사용해서 안되며 사용할 경우 구문 오류가 발생합니다..

EH 모델:

포인터	N	중첩된 주 제어 사용 시	N0~N7, 8 points	중첩된 주 제어 포인트를 제어함.
	P	CJ, CALL 명령에서 사용 시	P0~P255, 256 points	CJ, CALL 포인터의 위치
	I	외부 입력 인터럽트	I00□(X0), I10□(X1), I20□(X2), I30□(X3), I40□(X4), I50□(X5), 6 points (□=1, 상승 에지 트리거 $\uparrow$ , □=0, 하강 에지 트리거 $\downarrow$ )	서브루틴 인터럽트 포인터 위치
		시간 입력 인터럽트	I6□□, I7□□, I8□□, 2 points (□□=1~99ms, time base=1ms) I8□□, 1 point (□□=0.1~9.9ms, time base=0.1ms)	
		도달된 고속 카운터 입력 인터럽트	I010, I020, I030, I040, I050, I060, 6 points	
		Pulse 입력 인터럽트	I110, I120, I130, I140, 4 points	
		통신 입력 인터럽트	I150, I160, I170, 3 points	



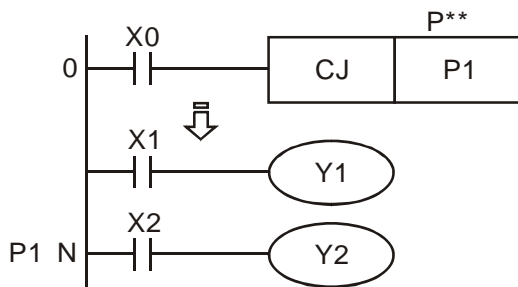
			인터럽트는 주파수 측량 카드에 트리거가 걸린다.	I180, 1 point
--	--	--	----------------------------------	---------------

노트: 고속 카운터에서 X는 입력 포인트로 사용되고 외부 인터럽트의 입력 신호로는 사용할 수 없습니다. 예를 들면, 만일 X0, X1, X2, 그리고 X3가 C251 고속 카운터의 입력 포인트로 사용됩니다.면 외부 인터럽트 번호 중 I00□(X0), I10□(X1), I20□(X2), 그리고 I30□(X3)는 사용될 수 없습니다.

중첩 레벨 포인터 N: MC와 MCR 명령에서 사용됩니다. MC는 주 시작 명령입니다.. MC 명령이 실행되면, MC와 MCR 사이의 명령은 일반적으로 실행될 것입니다.. N0에서 N7의 번호는 MC-MCR의 주 명령은 중첩된 프로그램 구조를 가질 것이며 최대 8레벨 이상일 것입니다.. 3.7의 상세 정보를 참고하기 바랍니다.

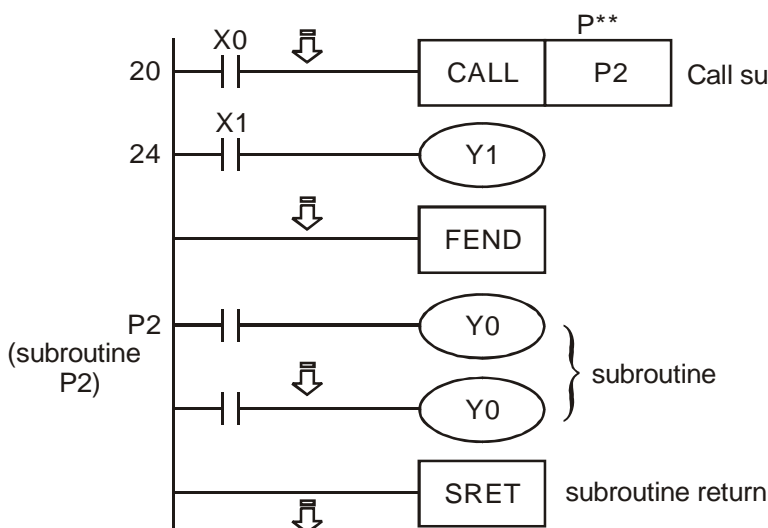
포인터 P: API 00 CJ, API 01 CALL, API 02 SRET 응용 프로그램 명령과 함께 사용됩니다.. 5.5를 참고해 CJ, CALL, SRET 명령의 사용 방법과 정보를 참고하도록 합니다..

CJ 점프 상태:



- X0=On 일 때, 프로그램은 0에서 N으로 점프할 것이며 주소가 0과 N 사이에서 실행되지 않아도 실행을 계속할 것입니다..(P1에 설계되어있음.)
- X0=Off 일 때, 프로그램은 0이 실행되며 실행이 계속 유지됩니다. CJ 명령은 이 시점에서는 실행되지 않는다.

CALL 서브루틴, SRET 서브루틴 END:



- X0이 On일 때, CALL 명령이 실행된 것과 같이 설계된 서브루틴을 따라 P2는 점프할 것입니다.. SRET 명령이 실행될 때, 24번지를 돌려주고 계속 실행합니다..

인터럽트 포인터 I:

이것은 API 04 EI, API 05 DI, API 03 IRET 응용 프로그램 명령과 함께 사용됩니다. 5.5를 통해 더 많은 정보를 참고하기 바랍니다. 아래에는 6가지 기능을 나타냈습니다. 입력 인터럽트는 EI와 함께 입력되어 사용될 것이고, 입력 인터럽트 가능, 불가능과 IRET 입력 인터럽트 등을 회신할 것입니다..

1. 외부 입력 인터럽트 : X0~X5 입력 터미널의 입력 신호가 상승 에지나 하강 에지로 트리거 되었을 때, 프로그램에 인터럽트가 준비될 것이며 설계된 서브루틴 입력 인터럽트 포인터 I00□(X0), I10□(X1), I20□(X2), I30□(X3), I40□(X4), I50□(X5) 가 실행되도록 하며 IRET 명령이 실행되었던 이전 주소 값을 회신합니다.. PLC MPU 특정 하드웨어 회전 설계로 만들어진 PLC MPU와 검사 기간동안은 영향을 미치지 않을 것입니다..
2. 타이머 입력 인터럽트 : 이것은 PLC MPU 내 특정 하드웨어 원형으로 설계된 것입니다.. 프로그램을 중지하도록 준비하며 설계된 입력 서브루틴 인터럽트에 자동으로 실행될 시간이 입력되도록 합니다.. (10ms~99ms 까지 설정할 수 있습니다.).
3. 카운터에 도달한 입력 인터럽트 : 비교가 완료되면 API 53 DHSCS 비교 명령은 고속 카운터에 인터럽트 이전 프로그램이 설계될 수 있도록 하며 설계된 서브루틴 입력 인터럽트에 점프해 I010, I020, I030, I040, I050, I060 인터럽트 포인터로 실행할 수 있도록 합니다..
4. Pulse 입력 인터럽트 : 처음 pulse 가 출력된 pulse 일 때 API 57 PLSY pulse 출력 명령 사용은 백터 인터럽트 I130(M1342 와 일치) 와 I140(M1343 와 일치)를 같은 시간에 보내는 것입니다.. 그러나 M1342 와 M1343 이 처음 시작 신호를 보내는 것과 같습니다. 그리고 I110 (M1340 와 일치) and I120(M1341 와 일치) 백터 인터럽트를 출력된 마지막 pulse 로 보내는 것과 같습니다.
5. 통신 입력 인터럽트 : RS 대화 명령을 사용할 때 특정 값을 받으면 요청 인터럽트를 설정할 수 있습니다.. I150 인터럽트 번호와 특정 값은 D1168 의 byte 보다 더 낮게 설정할 수 있습니다..  
  
 PLC 가 통신 장치와 연결하고 받는 데이터의 길이가 같지 않다면, 마지막 값은 D1168 로 설정하고 1150 으로 인터럽트 서브루틴을 구성합니다..  
 PLC 가 마지막 값을 받으면, I150 서브루틴 인터럽트를 실행할 것입니다..  
  
 RS 통신 명령 사용 시 특정 길이를 받으면 요청 인터럽트를 설정할 수 있습니다.. 인터럽트 번호는 I160 이고 특정 길이는 D1169 보다 낮은 byte 로 구성합니다.. D1169=0 이면, 요청 인터럽트는 없습니다.  
  
 일반적으로 PLC 통신 포트가 SLAVE 모드로 있을 때, PLC 는 END 명령이 들어올때까지 통신 데이터를 바로 보내주지 못합니다.. 만일 명령 조사 시간이 너무 걸리면 PLC 는 요청 데이터에 대해 즉시 반응하는 통신 데이터를 주고받는 작업을 끊어버린다. PLC 와 연결하기 위해서는 END 명령을 실행해야하는 제한 없이 즉각적인 반응을 요구하는 통신 데이터를 보내줘야합니다.. I170 명령은 PLC 통신 데이터의 요청에 즉시 대답할 수 있는 작업에서 사용됩니다.
6. 인터럽트는 주파수 측량 카드에 의해 트리거됩니다. : PLC 가 M1019 로 설정됩니다. (주파수 측정 카드가 작업 모드로 설정됩니다.) 그리고 D1034 (주파수 측정 카드가 작업 모드로 설정됩니다.), 모드 1 로 설정 (pulse 주기 측정) 그리고 모드 3 (pulse 번호 계산) I180 인터럽트가 가진 주파수 측정 카드입니다..

## 2.10 특수 보조 릴레이와 특수 레지스터

특수 보조 중계(특수 M)과 레지스터(특수 D)의 종류 및 기능은 다음과 같습니다. 동 번호의 장비도 모델명이 다른 경우에 서로 상이하다는 것에 주목해야 합니다. 다음 차트를 보면 세로 “속성”의 의미는 다음과 같습니다. “R”은 “판독전용”을, “R/W”는 “판독/작성가능”을 “-”은 “불가능”을 의미합니다. “#”은 “PLC 상태에 따른 시스템 설정”을 의미하며, 사용자는 매뉴얼에서 설정에 대한 상세한 설명을 읽을 수 있습니다. “\*”는 “설명을 위해 다음사항을 참조할 수 있음”을 의미합니다.

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 켜짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1000*	보통 접촉을 개방합니다.(a 접촉). 이접촉은 실행할 때 “On”이고 상태가 “RUN”으로 설정될 때 “On”이 됩니다..	○	○	○	#	On	Off	R	NO	Off
M1001*	보통 접촉을 단절합니다.(b 접촉). 이접촉은 실행할 때 “Off”이고 상태가 “RUN”으로 설정될 때 “Off”가 됩니다..	○	○	○	#	Off	On	R	NO	On
M1002*	RUN 후에는 1 스캔 전용 “On”. 초기 펄스는 접촉 a 임. 이는 “RUN”상태시에 양의 펄스를 얻을 것임. 펄스 폭=스캔 주기	○	○	○	#	On	Off	R	NO	Off
M1003*	RUN 후에는 1 스캔 전용 “Off”. 초기 펄스는 접촉 a 임. 이는 “RUN”상태시에 음의 펄스를 얻을 것임. 펄스 폭=스캔	○	○	○	#	Off	On	R	NO	On
M1004*	오류 발생시 “On”	○	○	○	Off	Off	-	R	NO	Off
M1005	데이터 백업 메모리 카드의 암호. MPU 암호는 일치하지 않음.	×	×	○	Off	Off	-	R	NO	Off
M1006	데이터 백업메모리 카드가 초기의 것이 아님.	×	×	○	Off	Off	-	R	NO	Off
M1007	데이터가 데이터 백업 메모리 카드의 프로그램 영역에 존재하지 않음.	×	×	○	Off	-	-	R	NO	Off
M1008*	모니터 타이머 플래그. (On: PLC WDT 종료)	○	○	○	Off	Off	-	R	NO	Off
M1009	24VD 가 충분하지 않아서 LV 신호를 일으킬 때, M1009 는 “ON” 상태가 됨.	○	○	○	Off	-	-	R	NO	Off
M1010	ES, EX, SS 및 SA, SX, SC: PLSY Y0 모드 선택. 이는 “On”상태일 때 연속적인 출력임. EH : PULS 는 END 상태에서 출력이 됨.	○	○	○	Off	-	-	R/W	NO	Off
M1011*	10ms 클록 펄스, 5ms On/5ms Off	○	○	○	Off	-	-	R	NO	Off
M1012*	100ms 클록 펄스, 50ms On / 50ms Off	○	○	○	Off	-	-	R	NO	Off
M1013*	1s 클록 펄스, 0.5s On / 0.5s Off	○	○	○	Off	-	-	R	NO	Off
M1014*	1min 클록 펄스, 30s On / 30s Off	○	○	○	Off	-	-	R	NO	Off
M1015*	고속 타이머 활성화	×	○	○	Off	-	-	R/W	NO	Off
M1016*	“Off”상태일때두개의 최우측 비트를 표시함. “On”상태일때(두개의 최우측 비트+2000)을 표시함	×	○	○	Off	-	-	R/W	NO	Off
M1017*	±30 초 조정	×	○	○	Off	-	-	R/W	NO	Off
M1018	레디안/도에 대한 플래그. 도에 대해선 “On”	×	○	○	Off	-	-	R/W	NO	Off
M1019	주파수측정 카드의 시동 플래그	×	×	○	Off	Off	-	R	否	Off
M1020	기점 플래그	○	○	○	Off	-	-	R	NO	Off
M1021	차용 플래그	○	○	○	Off	-	-	R	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 컴	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1022	운반 플래그	○	○	○	Off	-	-	R	NO	Off
M1023	PLSY Y1 모드 선택. “On” 상태일 때의 연속적인 출력임..	○	○	×	Off	-	-	R/W	NO	Off
M1024	COM1 모니터 요청	○	○	○	Off	-	-	R	NO	Off
M1025	HPP, PC 또는 HMI 가 PLC 에 접속할 때, PLC 가 불법적인 통신 요청을 수신합니다.면, M1025 가 설정되어 D1025 에서 오류 코드를 저장할 것임..	○	○	○	Off	-	-	R	NO	Off
M1026	SA, SX, SC, EH: RAMP 모듈의 시동 플래그	×	○	○	Off	-	-	R/W	NO	Off
M1027	PR 출력 플래그	×	○	○	Off	-	-	R/W	NO	Off
M1028	10ms/100ms 타임 스위치 플래그. T64~T126 의 베이스 설정 플래그는 타이머가 “Off” 상태일때는 100ms 이고, “On” 상태일 때는 10ms 입니다.	○	×	×	Off	-	-	R/W	NO	Off
M1029*	ES, EX, SS 및 SA, SX, SC: PLSY 및 PLSR 의 펄스 출력 Y0 명령어 완료 또는 기타 관련 명령어 실행 완료. EH: 최초 그룹 펄스 CH0 (Y0, Y1) 출력 실행 완료 또는 기타 관련 명령어 실행 완료.	○	○	○	Off	-	-	R	NO	Off
M1030*	ES, EX, SS 및 SA, SX, SC: PLSY 및 PLSR 의 펄스 출력 Y1 명령어 실행 완료. EH: 두번째 그룹 펄스 CH1 (Y2, Y3) 출력 실행 완료	○	○	○	Off	-	-	R	NO	Off
M1031*	모든 비래치 메모리 해제	○	○	○	Off	-	-	R/W	NO	Off
M1032*	모든 래치 메모리 해제	○	○	○	Off	-	-	R/W	NO	Off
M1033*	“STOP” 상태에서 래치된 메모리	○	○	○	Off	-	-	R/W	NO	Off
M1034*	모든 IY 출력 불능	○	○	○	Off	-	-	R/W	NO	Off
M1035*	입력점을 개시하여 RUN/STOP 스위치로 되게 하고 D1035 에 상응하도록 함. ( SA 모델에 대해선 X7 가 전용. SX 모델에 대해선 X3 가 전용. SC 모델에 대해선 X5 가 전용)	×	○	○	-	-	-	R/W	YES	Off
M1039*	지속적인 스캔 모드	○	○	○	Off	-	-	R/W	NO	Off
M1040	단계 전이금지	○	○	○	Off	-	-	R/W	NO	Off
M1041	단계 전이 개시	○	○	○	Off	-	Off	R/W	NO	Off
M1042	개시 펄스	○	○	○	Off	-	-	R/W	NO	Off
M1043	기점 복귀 완료	○	○	○	Off	-	Off	R/W	NO	Off
M1044	기점 상태	○	○	○	Off	-	Off	R/W	NO	Off
M1045	모든 출력의 금지 해제	○	○	○	Off	-	-	R/W	NO	Off
M1046	STL 상태 설정(On)	○	○	○	Off	-	-	R	NO	Off
M1047	STL 모니터 가능	×	○	○	Off	-	-	R/W	NO	Off
M1048	경보 지점 상태용 플래그	×	○	○	Off	-	-	R	NO	Off
M1049	경보 지점의 모니터 플래그	×	○	○	Off	-	-	R/W	NO	Off
M1050	I001 차폐됨	○	○	×	Off	-	-	R/W	NO	Off
M1051	I101 차폐됨	○	○	×	Off	-	-	R/W	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 켜짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1052	I201 차폐됨	○	○	×	Off	-	-	R/W	NO	Off
M1053	I301 차폐됨	○	○	×	Off	-	-	R/W	NO	Off
M1054	I401 차폐됨	×	○	×	Off	-	-	R/W	NO	Off
M1055	I501 차폐됨	×	○	×	Off	-	-	R/W	NO	Off
M1056	I6□□ 차폐됨	○	○	×	Off	-	-	R/W	NO	Off
M1057	I7□□ 차폐됨	×	○	×	Off	-	-	R/W	NO	Off
M1059	I010~I060 차폐됨	×	○	×	Off	-	-	R/W	NO	Off
M1060	시스템 오류 메시지 1	○	○	○	Off	-	-	R	NO	Off
M1061	시스템 오류 메시지 2	○	○	○	Off	-	-	R	NO	Off
M1062	시스템 오류 메시지 3	○	○	○	Off	-	-	R	NO	Off
M1063	시스템 오류 메시지 4	○	○	○	Off	-	-	R	NO	Off
M1064	운용자 오류	○	○	○	Off	Off	-	R	NO	Off
M1065	구문 오류	○	○	○	Off	Off	-	R	NO	Off
M1066	루프 오류	○	○	○	Off	Off	-	R	NO	Off
M1067*	프로그램 실행 오류	○	○	○	Off	Off	-	R	NO	Off
M1068*	실행 오류 잠금(D1068)	○	○	○	Off	-	-	R	NO	Off
M1070	ES, EX, SS 및 SA, SX, SC: PWM 명령어 Y1의 타임 펄스 단위 스위치. “On” 상태일 때 타임 펄스 단위는 100us 이고, “Off” 상태일 때는 1ms 임. EH: 최초 펄스 CH0 (Y0, Y1)의 PWM 명령어용 단위 설정. “On” 상태는 100us 이고 “Off” 상태는 1ms 입니다. .	○	○	○	Off	-	-	R/W	NO	Off
M1071	두번째 펄스 CH0 (Y2, Y3)의 PWM 명령어용 단위 설정. “On” 상태는 100us 이고 “Off” 상태는 1ms 입니다.	×	×	○	Off	-	-	R/W	NO	Off
M1072	PLC RUN 명령어 실행	○	○	○	Off	On	Off	R/W	NO	Off
M1075*	FLASH 작성 오류	×	×	○	Off	-	-	R	NO	Off
M1076*	실제 타임 클록 오류	×	○	○	Off	-	-	R	NO	Off
M1077	배터리 전압이 너무 낮거나 오기능	×	○	○	Off	-	-	R	NO	Off
M1078	PLSY 명령어 Y0 펄스 출력 정지 즉시 플래그	○	○	×	Off	-	-	R/W	NO	Off
M1079	PLSY 명령어 Y1 펄스 출력 정지 즉시 플래그	○	○	×	Off	-	-	R/W	NO	Off
M1080	COM2 모니터 요청	○	○	○	Off	-	-	R	NO	Off
M1081	FLT 명령어 변경 방향 플래그	×	○	○	Off	-	-	R/W	NO	Off
M1082	RTC를 위해 변경된 플래그	×	○	○	Off	-	-	R	NO	Off
M1083	FROM/TO 모드에서 가능/불능 실행 차단 프로그램	×	○	○	Off	-	-	R/W	NO	Off
M1086	DVP-PCC01 용 암호 기능의 ON/OFF 스위치 설정	○	○	○	Off	-	-	R/W	NO	Off
M1087	LV 신호 운용 플래그	×	×	○	Off	-	-	R/W	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 컴	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1088	매트릭스 비교 플래그. 결과가 동일하면, M1088 = 1. 결과가 다르면 M1088 = 0.	×	○	○	Off	Off	-	R/W	NO	Off
M1089	매트릭스 탐색 시동 플래그. 첫번째 비트를 비교하면 M1090=1.	×	○	○	Off	Off	-	R	NO	Off
M1090	매트릭스 탐색 개시 플래그. 첫번 제비트를 비교하면 M1090=1.	×	○	○	Off	Off	-	R	NO	Off
M1091	매트릭스 발견 비트 플래그. 발견하면 비교를 멈출 것이고 M1091=1.	×	○	○	Off	Off	-	R	NO	Off
M1092	매트릭스 지시자 오류 플래그. 지시자 Pr 이 이 범위를 초과하면 M1092=1.	×	○	○	Off	Off	-	R	NO	Off
M1093	매트릭스 지시자 증가 플래그. 이는 1 을 추가하여 지시자를 나타냄.	×	○	○	Off	Off	-	R/W	NO	Off
M1094	매트릭스 해제 플래그. 이는 현재 지시자를 0 으로 해제함.	×	○	○	Off	Off	-	R/W	NO	Off
M1095	매트릭스 회전/이동 출력용 운반 플래그	×	○	○	Off	Off	-	R	NO	Off
M1096	매트릭스 이동 출력용 보충 플래그	×	○	○	Off	Off	-	R/W	NO	Off
M1097	매트릭스 회전/이동용 방향 플래그	×	○	○	Off	Off	-	R/W	NO	Off
M1098	매트릭스 계수 비트 0 또는 1 플래그	×	○	○	Off	Off	-	R/W	NO	Off
M1099	매트릭스 계수 결과가 0 일 때 “On”	×	○	○	Off	Off	-	R/W	NO	Off
M1100	SPD 명령어의 샘플링 플래그	×	×	○	Off	-	-	R/W	NO	Off
M1101*	파일 레지스터의 개시 여부 결정	×	○	○	-	-	-	R/W	Yes	Off
M1102	펄스 출력의 Y10 종료 플래그(SC 시리즈 모델 전용)	×	○	×	Off	-	-	R/W	NO	Off
M1103	펄스 출력의 Y11 종료 플래그 (SC 시리즈 모델 전용)	×	○	×	Off	-	-	R/W	NO	Off
M1104*	DIP 스위치 기능 카드 SW1 상태/ 4DI 카드: AX0 출력지점(광접합 소자 격리)	×	×	○	Off	Off	-	R	NO	Off
M1105*	DIP 스위치 기능 카드 SW2 상태/ 4DI 카드: AX1 입력 지점(광접합 소자 격리)	×	×	○	Off	Off	-	R	NO	Off
M1106*	DIP 스위치 기능 카드 SW3 상태/ 4DI 카드: AX2 입력 지점(광접합 소자 격리)	×	×	○	Off	Off	-	R	NO	Off
M1107*	DIP 스위치 기능 카드 SW4 상태 / 4DI 카드: AX3 입력 지점(광접합 소자 격리)	×	×	○	Off	Off	-	R	NO	Off
M1108*	DIP 스위치 기능 카드 SW5 상태	×	×	○	Off	Off	-	R	NO	Off
M1109*	DIP 스위치 기능 카드 SW6 상태	×	×	○	Off	Off	-	R	NO	Off
M1110*	DIP 스위치 기능 카드 SW7 상태	×	×	○	Off	Off	-	R	NO	Off
M1111*	DIP 스위치 기능 카드 SW8 상태	×	×	○	Off	Off	-	R	NO	Off
M1112*	2DO 카드: AY0 출력 지점(트랜지스터)	×	×	○	Off	-	Off	R/W	NO	Off
M1113*	2DO 카드: AY1 출력 지점 (트랜지스터)	×	×	○	Off	-	Off	R/W	NO	Off
M1115*	가속/감속 펄스 출력용 개시 스위치	○	○	×	Off	Off	Off	R/W	NO	Off
M1116*	가속/감속 펄스 출력용 가속 플래그	○	○	×	Off	Off	Off	R/W	NO	Off
M1117*	목표 달성 주파수 플래그	○	○	×	Off	Off	Off	R/W	NO	Off
M1118*	가속/감속 펄스 출력용 감속	○	○	×	Off	Off	Off	R/W	NO	Off
M1119*	완료 기능 플래그	○	○	×	Off	Off	Off	R/W	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 켜짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1120	통신 프로토콜 보류, 설정후에는 D1120 수정 무효	○	○	○	Off	Off	-	R/W	NO	Off
M1121	RS-485 통신 데이터의 송신 대기	○	○	○	Off	On	-	R	NO	Off
M1122	송신 요청	○	○	○	Off	Off	-	R/W	NO	Off
M1123	수신 완료	○	○	○	Off	Off	-	R/W	NO	Off
M1124	수신 대기	○	○	○	Off	Off	-	R	NO	Off
M1125	통신 재설정	○	○	○	Off	Off	-	R/W	NO	Off
M1126	STX/ETX 사용자/시스템 선택	○	○	○	Off	Off	-	R/W	NO	Off
M1127	통신 명령어에 대한 데이터 송신 또는 수신 종료. RS 명령어 배제.	○	○	○	Off	Off	-	R/W	NO	Off
M1128	전송/수신 표시	○	○	○	Off	Off	-	R	NO	Off
M1129	타임 아웃 수신	○	○	○	Off	Off	-	R/W	NO	Off
M1130	STX/ETX 선택	○	○	○	Off	Off	-	R/W	NO	Off
M1131	MODRD/RDST/MODRW, 데이터가 HEX 로 변환할 때 M1131=On	○	○	○	Off	Off	Off	R	NO	Off
M1132	“On” 상태는 PLC 에 관련 통신 명령어가 없음을 의미함.	○	○	○	Off	-	-	R	NO	Off
M1133*	특수 고속 펄스(50KHz) 출력 Y0 스위치(“On”상태는 개시임.)	×	○	×	Off	Off	Off	R/W	NO	Off
M1134*	특수 고속 펄스(50KHz) 출력 Y0 스위치. “On” 상태는 연속적인 출력 스위치임.	×	○	×	Off	Off	-	R/W	NO	Off
M1135*	특수 고속 펄스(50KHz) 출력 Y0 스위치. 출력 펄스 번호 달성 플래그	×	○	×	Off	Off	Off	R/W	NO	Off
M1136*	COM3 통신 프로토콜 보류	×	×	○	Off	-	-	R/W	NO	Off
M1138*	COM1 (RS-232) 통신 프로토콜 보류. 설정 후에는 D1036 수정 무효	○	○	○	Off	-	-	R/W	NO	Off
M1139*	SLAVE 모드일 때, COM1 (RS-232)을 위해 ASCII/RTU 선택. ASCII 모드에 대해선 OFF, RTU 모드에 대해선 ON.	○	○	○	Off	-	-	R/W	NO	Off
M1140	MODRD/MODWR/MODRW 데이터 수신 오류	○	○	○	Off	Off	-	R	NO	Off
M1141	MODRD/MODWR/MODRW 명령어 오류	○	○	○	Off	Off	-	R	NO	Off
M1142	VFD-A 명령어 데이터 수신 오류	○	○	○	Off	Off	-	R	NO	Off
M1143	SLAVE 모드일 때, COM2(RS-485)에 대해선 ASCII/RTU 모드 선택. ASCII 모드에 대해선 OFF, RTU 모드에 대해선 ON. MASTER 모드일 때, COM2 (RS-485) 에 대해선 ASCII/RTU 모드 선택(MODRD / MODWR / MODRW 과 함께 사용) (ASCII 모드에서는 “Off” 상태이고 RTU 에서는 “On” 상태임.)	○	○	○	Off	-	-	R/W	NO	Off
M1144*	조정가능 기율기의 가속/감속 펄스 출력 Y0 기능의 출력 개시 스위치	×	○	×	Off	Off	Off	R/W	NO	Off
M1145*	조정가능 기율기의 가속/감속 펄스 출력 Y0 기능의 가속 플래그.	×	○	×	Off	Off	-	R	NO	Off
M1146*	조정가능 기율기의 가속/감속 펄스 출력 Y0 기능의 목표 달성 주파수 플래그.	×	○	×	Off	Off	-	R	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 컴	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1147*	조정가능 기울기의 가속/감속 펄스 출력 Y0 기능의 감속 플래그.	×	○	×	Off	Off	-	R	NO	Off
M1148*	조정가능 기울기의 가속/감속 펄스 출력 Y0 기능의 완료 기능 플래그.	×	○	×	Off	Off	Off	R/W	NO	Off
M1149*	조정가능 기울기의 가속/감속 펄스 출력 Y0 기능의 계수 일시성 플래그의 정지	×	○	×	Off	Off	-	R/W	NO	Off
M1150	다그룹 설정 비교 모드에 사용된 DHSZ 명령어 선언	×	×	○	Off	-	-	R/W	NO	Off
M1151	다그룹 설정비교 모드 실행 종료	×	×	○	Off	-	-	R	NO	Off
M1152	주파수 제어 모드에 사용된 DHSZ 명령어 선언	×	×	○	Off	-	-	R/W	NO	Off
M1153	주파수 제어 모드 실행 종료	×	×	○	Off	-	-	R	NO	Off
M1154*	조정가능 기울기의 가속/감속 펄스 출력 Y0 기능의 지정된 감속 기능 플래그의 개시	×	○	×	Off	-	-	R/W	NO	Off
M1161	8/16 비트 모드(8 비트 모드에서는 “On” 상태임)	○	○	○	Off	-	-	R/W	NO	Off
M1167	HKY 입력은 16 비트 모드	×	○	○	Off	-	-	R/W	NO	Off
M1168	SMOV 동작 모드 표시	×	○	○	Off	-	-	R/W	NO	Off
M1169	PWD 모드 선택	×	×	○	Off	-	-	R/W	NO	Off
M1170*	신호 단계 실행 개시	×	×	○	Off	-	-	R/W	NO	Off
M1171*	신호 단계 실행	×	×	○	Off	-	-	R/W	NO	Off
M1172*	2-페이지 펄스 출력 스위치(“on”상태는 개시임)	×	○	×	Off	Off	Off	R/W	NO	Off
M1173*	“On” 상태는 연속적인 출력 스위치임.	×	○	×	Off	-	-	R/W	NO	Off
M1174*	출력 펄스 번호 달성 플래그	×	○	×	Off	Off	Off	R/W	NO	Off
M1178*	VR0 전위차계 개시	×	○	○	Off	-	-	R/W	NO	Off
M1179*	VR1 전위차계 개시	×	○	○	Off	-	-	R/W	NO	Off
M1184*	MODEM 기능 시동	×	×	○	Off	-	-	R/W	NO	Off
M1185*	MODEM 초기화 기능	×	×	○	Off	-	-	R/W	NO	Off
M1186*	MODEM 초기화 실패	×	×	○	Off	-	-	R/W	NO	Off
M1187*	MODEM 초기화 완료	×	×	○	Off	-	-	R/W	NO	Off
M1188*	MODEM 접속 상태 표시	×	×	○	Off	-	-	R/W	NO	Off
M1196	디스플레이의 내용 설정(Off: Dec, On: Hex), SX 모델전용	×	○	×	Off	-	-	R/W	NO	Off
M1197	디스플레이용 둘째자리 십진법 설정, SX 모델전용	×	○	×	Off	-	-	R/W	NO	Off
M1198	디스플레이용 첫째자리 십진법 설정, SX	×	○	×	Off	-	-	R/W	NO	Off
M1200	C200 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1201	C201 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1202	C202 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1203	C203 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1204	C204 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1205	C205 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off



특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 켜짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1206	C206 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1207	C207 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1208	C208 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1209	C209 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1210	C210 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1211	C211 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1212	C212 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1213	C213 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1214	C214 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1215	C215 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1216	C216 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1217	C217 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1218	C218 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1219	C219 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1220	C220 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1221	C221 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1222	C222 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1223	C223 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1224	C224 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1225	C225 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1226	C226 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1227	C227 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1228	C228 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1229	C229 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1230	C230 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1231	C231 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1232	C232 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1233	C233 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1234	C234 계수 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1235	C235 계수 모드 설정(on: 초 읽기)	○	○	○	Off	-	-	R/W	NO	Off
M1236	C236 계수 모드 설정(on: 초 읽기)	○	○	○	Off	-	-	R/W	NO	Off
M1237	C237 계수 모드 설정(on: 초 읽기)	○	○	○	Off	-	-	R/W	NO	Off
M1238	C238 계수 모드 설정(on: 초 읽기)	○	○	○	Off	-	-	R/W	NO	Off
M1239	C239 계수기 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1240	C240 계수기 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1241	C241 계수기 모드 설정(on: 초 읽기)	○	○	○	Off	-	-	R/W	NO	Off
M1242	C242 계수기 모드 설정(on: 초 읽기)	○	○	○	Off	-	-	R/W	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 컴	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1243	C243 계수기 모드 설정(on: 초 읽기)	×	○	○	Off	-	-	R/W	NO	Off
M1244	C244 계수기 모드 설정(on: 초 읽기)	○	○	○	Off	-	-	R/W	NO	Off
M1245	C245 계수기 모드 설정(on: 초 읽기)	×	○	×	Off	-	-	R/W	NO	Off
M1246	C246 계수기 모니터(on: 초 읽기)	○	○	○	Off	-	-	R	NO	Off
M1247	C247 계수기 모니터(on: 초 읽기)	○	○	○	Off	-	-	R	NO	Off
M1248	C247 계수기 모니터(on: 초 읽기)	×	×	○	Off	-	-	R	NO	Off
M1249	C249 계수기 모니터(on: 초 읽기)	○	○	○	Off	-	-	R	NO	Off
M1250	C250 계수기 모니터(on: 초 읽기)	×	○	×	Off	-	-	R	NO	Off
M1251	C251 계수기 모니터(on: 초 읽기)	○	○	○	Off	-	-	R	NO	Off
M1252	C252 계수기 모니터(on: 초 읽기)	○	○	○	Off	-	-	R	NO	Off
M1253	C254 계수기 모니터(on: 초 읽기)	×	×	○	Off	-	-	R	NO	Off
M1254	C254 계수기 모니터(on: 초 읽기)	○	○	○	Off	-	-	R	NO	Off
M1256	EF 오류 플래그	×	×	○	Off	Off	-	R	NO	Off
M1258	PWM 명령어용 역방향 Y0 펄스 출력 신호	×	×	○	Off	-	-	R/W	NO	Off
M1259	PWM 명령어용 역방향 Y2 펄스 출력 신호	×	×	○	Off	-	-	R/W	NO	Off
M1260	X5 를 모든 고속 계수기의 재설정 입력 신호가 되게 함.	×	○	×	Off	-	-	R/W	NO	Off
M1261	고속 계수기용 DHSCR 명령어의 비교 플래그	×	×	○	Off	Off	Off	R/W	NO	Off
M1264	HHSC0 개시 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1265	HHSC0 재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1266	HHSC1 개시 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1267	HHSC1 재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1268	HHSC2 개시 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1269	HHSC2 재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1270	HHSC3 개시 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1271	HHSC3 재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1272	HHSC0 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1273	HHSC0 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1274	HHSC1 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1275	HHSC1 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1276	HHSC2 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1277	HHSC2 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1278	HHSC3 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1279	HHSC3 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1280	I00□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1281	I10□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1282	I20□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1283	I30□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 켜짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1284	I40□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1285	I50□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1286	I6□□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1287	I7□□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1288	I8□□ 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1289	I010 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1290	I020 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1291	I030 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1292	I040 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1293	I050 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1294	I060 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1295	I110 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1296	I120 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1297	I130 차폐됨	×	×	○	Off	-	-	R/W	否	Off
M1298	I140 차폐됨	×	×	○	Off	-	-	R/W	否	Off
M1299	I150 차폐됨	×	○	○	Off	-	-	R/W	否	Off
M1300	I160 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1301	I170 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1302	I180 차폐됨	×	×	○	Off	-	-	R/W	NO	Off
M1303	XCH 명령어 Swap 고/저 바이트	×	○	○	Off	-	-	R/W	NO	Off
M1304*	X 입력 지점이 On-Off 상태로 결정될 수 있음	×	○	○	Off	-	-	R/W	NO	Off
M1305	PLSV, DPLSV, DRVI, DDRVI, DRVA, DDRVA CH0(Y0・Y1) 역방향 운용 플래그	×	×	○	Off	-	-	R	NO	Off
M1306	PLSV, DPLSV, DRVI, DDRVI, DRVA, DDRVA CH1(Y2・Y3) 역방향 운용 플래그	×	×	○	Off	-	-	R	NO	Off
M1312	C235 입력지점 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1313	C236 입력지점 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1314	C237 입력지점 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1315	C238 입력지점 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1316	C239 입력지점 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1317	C240 입력지점 제어 개시	×	×	○	Off	-	-	R/W	NO	Off
M1320	C235 입력지점 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1321	C236 입력지점 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1322	C237 입력지점 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1323	C238 입력지점 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1324	C239 입력지점 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1325	C240 입력지점 제어 재설정	×	×	○	Off	-	-	R/W	NO	Off
M1328	C235 개시/재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 컴	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1329	C236 개시/재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1330	C237 개시/재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1331	C238 개시/재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1332	C239 개시/재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1333	C240 개시/재설정 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1334	CH0 (Y0, Y1) 펄스 출력 일시 정지	×	×	○	Off	-	-	R/W	NO	Off
M1335	CH1 (Y2, Y3) 펄스 출력 일시 정지	×	×	○	Off	-	-	R/W	NO	Off
M1336	CH0 (Y0, Y1) 펄스 송신 플래그	×	×	○	Off	Off	Off	R	NO	Off
M1337	CH1 (Y2, Y3) 펄스 송신 플래그	×	×	○	Off	Off	Off	R	NO	Off
M1338	CH0 (Y0, Y1) 오프셋 펄스 플래그 개시	×	×	○	Off	-	-	R/W	NO	Off
M1339	CH1 (Y2, Y3) 오프셋 펄스 플래그 개시	×	×	○	Off	-	-	R/W	NO	Off
M1340	CH0 (Y0, Y1) 펄스 송신 종료 후 차단 (I110)	×	×	○	Off	-	-	R/W	NO	Off
M1341	(Y2, Y3) 펄스 송신 종료 후 차단 (I120)	×	×	○	Off	-	-	R/W	NO	Off
M1342	CH0 (Y0, Y1) 펄스 송신과 동시에 차단 (I130)	×	×	○	Off	-	-	R/W	NO	Off
M1343	CH1 (Y2, Y3) 펄스 송신과 동시에 차단 (I140)	×	×	○	Off	-	-	R/W	NO	Off
M1344	CH0 (Y0, Y1) 보상 펄스 플래그 개시	×	×	○	Off	-	-	R/W	NO	Off
M1345	CH1 (Y2, Y3) 보상 펄스 플래그 개시	×	×	○	Off	-	-	R/W	NO	Off
M1346	명령어용 “CLEAR” 출력 신호 기능 가능	×	×	○	Off	-	-	R/W	NO	Off
M1347	첫번째 펄스 그룹의 자동 재설정 플래그	×	×	○	Off	-	-	R/W	NO	Off
M1348	두번째 펄스 그룹의 자동 재설정 플래그	×	×	○	Off	-	-	R/W	NO	Off
M1350*	PLC LINK 개시 플래그	×	○	○	Off	-	-	R/W	NO	Off
M1351*	PLC LINK 를 자동모드로 시동	×	○	○	Off	-	-	R/W	NO	Off
M1352*	PLC LINK 를 수동모드로 시동	×	○	○	Off	-	-	R/W	NO	Off
M1353	PLC 를 시동하여 32 개 이상의 외장 장치 및 16 개 이상의 판독/작성 기능(최대 100 개 기록)에 연결	×	×	○	Off	-	-	R/W	NO	Off
M1354	PLC LINK 동시 판독/작성 기능 시동	×	○	○	Off	-	-	R/W	NO	Off
M1360*	PLC LINK SLAVE ID 1 존재	×	○	○	Off	-	-	R	NO	Off
M1361*	PLC LINK SLAVE ID 2 존재	×	○	○	Off	-	-	R	NO	Off
M1362*	PLC LINK SLAVE ID 3 존재	×	○	○	Off	-	-	R	NO	Off
M1363*	PLC LINK SLAVE ID 4 존재	×	○	○	Off	-	-	R	NO	Off
M1364*	PLC LINK SLAVE ID 5 존재	×	○	○	Off	-	-	R	NO	Off
M1365*	PLC LINK SLAVE ID 6 존재	×	○	○	Off	-	-	R	NO	Off
M1366*	PLC LINK SLAVE ID 7 존재	×	○	○	Off	-	-	R	NO	Off
M1367*	PLC LINK SLAVE ID 8 존재	×	○	○	Off	-	-	R	NO	Off
M1368*	PLC LINK SLAVE ID 9 존재	×	○	○	Off	-	-	R	NO	Off
M1369*	PLC LINK SLAVE ID 10 존재	×	○	○	Off	-	-	R	NO	Off
M1370*	PLC LINK SLAVE ID 11 존재	×	○	○	Off	-	-	R	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 켜짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1371*	PLC LINK SLAVE ID 12 존재	×	○	○	Off	-	-	R	NO	Off
M1372*	PLC LINK SLAVE ID 13 존재	×	○	○	Off	-	-	R	NO	Off
M1373*	PLC LINK SLAVE ID 14 존재	×	○	○	Off	-	-	R	NO	Off
M1374*	PLC LINK SLAVE ID 15 존재	×	○	○	Off	-	-	R	NO	Off
M1375*	PLC LINK SLAVE ID 16 존재	×	○	○	Off	-	-	R	NO	Off
M1376*	PLC LINK SLAVE ID 1 활성화	×	○	○	Off	-	-	R	NO	Off
M1377*	PLC LINK SLAVE ID 2 활성화	×	○	○	Off	-	-	R	NO	Off
M1378*	PLC LINK SLAVE ID 3 활성화	×	○	○	Off	-	-	R	NO	Off
M1379*	PLC LINK SLAVE ID 4 활성화	×	○	○	Off	-	-	R	NO	Off
M1380*	PLC LINK SLAVE ID 5 활성화	×	○	○	Off	-	-	R	NO	Off
M1381*	PLC LINK SLAVE ID 6 활성화	×	○	○	Off	-	-	R	NO	Off
M1382*	PLC LINK SLAVE ID 7 활성화	×	○	○	Off	-	-	R	NO	Off
M1383*	PLC LINK SLAVE ID 8 활성화	×	○	○	Off	-	-	R	NO	Off
M1384*	PLC LINK SLAVE ID 9 활성화	×	○	○	Off	-	-	R	NO	Off
M1385*	PLC LINK SLAVE ID 10 활성화	×	○	○	Off	-	-	R	NO	Off
M1386*	PLC LINK SLAVE ID 11 활성화	×	○	○	Off	-	-	R	NO	Off
M1387*	PLC LINK SLAVE ID 12 활성화	×	○	○	Off	-	-	R	NO	Off
M1388*	PLC LINK SLAVE ID 13 활성화	×	○	○	Off	-	-	R	NO	Off
M1389*	PLC LINK SLAVE ID 14 활성화	×	○	○	Off	-	-	R	NO	Off
M1390*	PLC LINK SLAVE ID 15 활성화	×	○	○	Off	-	-	R	NO	Off
M1391*	PLC LINK SLAVE ID 16 활성화	×	○	○	Off	-	-	R	NO	Off
M1392*	PLC LINK SLAVE ID 1 ERROR	×	○	○	Off	-	-	R	NO	Off
M1393*	PLC LINK SLAVE ID 2 ERROR	×	○	○	Off	-	-	R	NO	Off
M1394*	PLC LINK SLAVE ID 3 ERROR	×	○	○	Off	-	-	R	NO	Off
M1395*	PLC LINK SLAVE ID 4 ERROR	×	○	○	Off	-	-	R	NO	Off
M1396*	PLC LINK SLAVE ID 5 ERROR	×	○	○	Off	-	-	R	NO	Off
M1397*	PLC LINK SLAVE ID 6 ERROR	×	○	○	Off	-	-	R	NO	Off
M1398*	PLC LINK SLAVE ID 7 ERROR	×	○	○	Off	-	-	R	NO	Off
M1399*	PLC LINK SLAVE ID 8 ERROR	×	○	○	Off	-	-	R	NO	Off
M1400*	PLC LINK SLAVE ID 9 ERROR	×	○	○	Off	-	-	R	NO	Off
M1401*	PLC LINK SLAVE ID 10 ERROR	×	○	○	Off	-	-	R	NO	Off
M1402*	PLC LINK SLAVE ID 11 ERROR	×	○	○	Off	-	-	R	NO	Off
M1403*	PLC LINK SLAVE ID 12 ERROR	×	○	○	Off	-	-	R	NO	Off
M1404*	PLC LINK SLAVE ID 13 ERROR	×	○	○	Off	-	-	R	NO	Off
M1405*	PLC LINK SLAVE ID 14 ERROR	×	○	○	Off	-	-	R	NO	Off
M1406*	PLC LINK SLAVE ID 15 ERROR	×	○	○	Off	-	-	R	NO	Off
M1407*	PLC LINK SLAVE ID 16 ERROR	×	○	○	Off	-	-	R	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 컴	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1408*	PLC LINK SLAVE ID 1 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1409*	PLC LINK SLAVE ID 2 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1410*	PLC LINK SLAVE ID 3 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1411*	PLC LINK SLAVE ID 4 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1412*	PLC LINK SLAVE ID 5 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1413*	PLC LINK SLAVE ID 6 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1414*	PLC LINK SLAVE ID 7 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1415*	PLC LINK SLAVE ID 8 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1416*	PLC LINK SLAVE ID 9 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1417*	PLC LINK SLAVE ID 10 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1418*	PLC LINK SLAVE ID 11 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1419*	PLC LINK SLAVE ID 12 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1420*	PLC LINK SLAVE ID 13 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1421*	PLC LINK SLAVE ID 14 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1422*	PLC LINK SLAVE ID 15 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1423*	PLC LINK SLAVE ID 16 판독 완료	×	○	○	Off	-	-	R	NO	Off
M1424*	PLC LINK SLAVE ID 1 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1425*	PLC LINK SLAVE ID 2 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1426*	PLC LINK SLAVE ID 3 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1427*	PLC LINK SLAVE ID 4 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1428*	PLC LINK SLAVE ID 5 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1429*	PLC LINK SLAVE ID 6 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1430*	PLC LINK SLAVE ID 7 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1431*	PLC LINK SLAVE ID 8 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1432*	PLC LINK SLAVE ID 9 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1433*	PLC LINK SLAVE ID 10 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1434*	PLC LINK SLAVE ID 11 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1435*	PLC LINK SLAVE ID 12 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1436*	PLC LINK SLAVE ID 13 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1437*	PLC LINK SLAVE ID 14 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1438*	PLC LINK SLAVE ID 15 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1439*	PLC LINK SLAVE ID 16 작성 완료	×	○	○	Off	-	-	R	NO	Off
M1440*	PLC LINK SLAVE ID 17 존재	×	×	○	Off	-	-	R	NO	Off
M1441*	PLC LINK SLAVE ID 18 존재	×	×	○	Off	-	-	R	NO	Off
M1442*	PLC LINK SLAVE ID 19 존재	×	×	○	Off	-	-	R	NO	Off
M1443*	PLC LINK SLAVE ID 20 존재	×	×	○	Off	-	-	R	NO	Off
M1444*	PLC LINK SLAVE ID 21 존재	×	×	○	Off	-	-	R	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 켜짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1445*	PLC LINK SLAVE ID 22 존재	×	×	○	Off	-	-	R	NO	Off
M1446*	PLC LINK SLAVE ID 23 존재	×	×	○	Off	-	-	R	NO	Off
M1447*	PLC LINK SLAVE ID 24 존재	×	×	○	Off	-	-	R	NO	Off
M1448*	PLC LINK SLAVE ID 25 존재	×	×	○	Off	-	-	R	NO	Off
M1449*	PLC LINK SLAVE ID 26 존재	×	×	○	Off	-	-	R	NO	Off
M1450*	PLC LINK SLAVE ID 27 존재	×	×	○	Off	-	-	R	NO	Off
M1451*	PLC LINK SLAVE ID 28 존재	×	×	○	Off	-	-	R	NO	Off
M1452*	PLC LINK SLAVE ID 29 존재	×	×	○	Off	-	-	R	NO	Off
M1453*	PLC LINK SLAVE ID 30 존재	×	×	○	Off	-	-	R	NO	Off
M1454*	PLC LINK SLAVE ID 31 존재	×	×	○	Off	-	-	R	NO	Off
M1455*	PLC LINK SLAVE ID 32 존재	×	×	○	Off	-	-	R	NO	Off
M1456*	PLC LINK SLAVE ID 17 활성화	×	×	○	Off	-	-	R	NO	Off
M1457*	PLC LINK SLAVE ID 18 활성화	×	×	○	Off	-	-	R	NO	Off
M1458*	PLC LINK SLAVE ID 19 활성화	×	×	○	Off	-	-	R	NO	Off
M1459*	PLC LINK SLAVE ID 20 활성화	×	×	○	Off	-	-	R	NO	Off
M1460*	PLC LINK SLAVE ID 21 활성화	×	×	○	Off	-	-	R	NO	Off
M1461*	PLC LINK SLAVE ID 22 활성화	×	×	○	Off	-	-	R	NO	Off
M1462*	PLC LINK SLAVE ID 23 활성화	×	×	○	Off	-	-	R	NO	Off
M1463*	PLC LINK SLAVE ID 24 활성화	×	×	○	Off	-	-	R	NO	Off
M1464*	PLC LINK SLAVE ID 25 활성화	×	×	○	Off	-	-	R	NO	Off
M1465*	PLC LINK SLAVE ID 26 활성화	×	×	○	Off	-	-	R	NO	Off
M1466*	PLC LINK SLAVE ID 27 활성화	×	×	○	Off	-	-	R	NO	Off
M1467*	PLC LINK SLAVE ID 28 활성화	×	×	○	Off	-	-	R	NO	Off
M1468*	PLC LINK SLAVE ID 29 활성화	×	×	○	Off	-	-	R	NO	Off
M1469*	PLC LINK SLAVE ID 30 활성화	×	×	○	Off	-	-	R	NO	Off
M1470*	PLC LINK SLAVE ID 31 활성화	×	×	○	Off	-	-	R	NO	Off
M1471*	PLC LINK SLAVE ID 32 활성화	×	×	○	Off	-	-	R	NO	Off
M1472*	PLC LINK SLAVE ID 17 ERROR	×	×	○	Off	-	-	R	NO	Off
M1473*	PLC LINK SLAVE ID 18 ERROR	×	×	○	Off	-	-	R	NO	Off
M1474*	PLC LINK SLAVE ID 19 ERROR	×	×	○	Off	-	-	R	NO	Off
M1475*	PLC LINK SLAVE ID 20 ERROR	×	×	○	Off	-	-	R	NO	Off
M1476*	PLC LINK SLAVE ID 21 ERROR	×	×	○	Off	-	-	R	NO	Off
M1477*	PLC LINK SLAVE ID 22 ERROR	×	×	○	Off	-	-	R	NO	Off
M1478*	PLC LINK SLAVE ID 23 ERROR	×	×	○	Off	-	-	R	NO	Off
M1479*	PLC LINK SLAVE ID 24 ERROR	×	×	○	Off	-	-	R	NO	Off
M1480*	PLC LINK SLAVE ID 25 ERROR	×	×	○	Off	-	-	R	NO	Off
M1481*	PLC LINK SLAVE ID 26 ERROR	×	×	○	Off	-	-	R	NO	Off

특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 컴	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1482*	PLC LINK SLAVE ID 27 ERROR	×	×	○	Off	-	-	R	NO	Off
M1483*	PLC LINK SLAVE ID 28 ERROR	×	×	○	Off	-	-	R	NO	Off
M1484*	PLC LINK SLAVE ID 29 ERROR	×	×	○	Off	-	-	R	NO	Off
M1485*	PLC LINK SLAVE ID 30 ERROR	×	×	○	Off	-	-	R	NO	Off
M1486*	PLC LINK SLAVE ID 31 ERROR	×	×	○	Off	-	-	R	NO	Off
M1487*	PLC LINK SLAVE ID 32 ERROR	×	×	○	Off	-	-	R	NO	Off
M1488*	PLC LINK SLAVE ID 17 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1489*	PLC LINK SLAVE ID 18 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1490*	PLC LINK SLAVE ID 19 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1491*	PLC LINK SLAVE ID 20 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1492*	PLC LINK SLAVE ID 21 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1493*	PLC LINK SLAVE ID 22 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1494*	PLC LINK SLAVE ID 23 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1495*	PLC LINK SLAVE ID 24 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1496*	PLC LINK SLAVE ID 25 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1497*	PLC LINK SLAVE ID 26 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1498*	PLC LINK SLAVE ID 27 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1499*	PLC LINK SLAVE ID 28 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1500*	PLC LINK SLAVE ID 29 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1501*	PLC LINK SLAVE ID 30 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1502*	PLC LINK SLAVE ID 31 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1503*	PLC LINK SLAVE ID 32 판독 완료	×	×	○	Off	-	-	R	NO	Off
M1504*	PLC LINK SLAVE ID 17 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1505*	PLC LINK SLAVE ID 18 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1506*	PLC LINK SLAVE ID 19 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1507*	PLC LINK SLAVE ID 20 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1508*	PLC LINK SLAVE ID 21 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1509*	PLC LINK SLAVE ID 22 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1510*	PLC LINK SLAVE ID 23 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1511*	PLC LINK SLAVE ID 24 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1512*	PLC LINK SLAVE ID 25 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1513*	PLC LINK SLAVE ID 26 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1514*	PLC LINK SLAVE ID 27 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1515*	PLC LINK SLAVE ID 28 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1516*	PLC LINK SLAVE ID 29 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1517*	PLC LINK SLAVE ID 30 작성 완료	×	×	○	Off	-	-	R	NO	Off
M1518*	PLC LINK SLAVE ID 31 작성 완료	×	×	○	Off	-	-	R	NO	Off



특수 M	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 켜짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
M1519*	PLC LINK SLAVE ID 32 작성 완료	×	×	○	Off	-	-	R	NO	Off

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1000*	감독 타이머 (WDT) 값 (단위: 1ms)	○	○	○	200	-	-	R/W	NO	200
D1001	DVP 모델시스템 프로그램 버전(사용자는 이 레지스터로부터 PLC 프로그램 버전을 읽을 수 있음. 예를 들면, D1001 = H XX10 는 버전 1.0 을 의미함.	○	○	○	#	-	-	R	NO	#
D1002*	프로그램 용량	○	○	○	#	-	-	R	NO	#
D1003	프로그램 메모리 합계	○	○	○	#	-	-	R	NO	#
D1004*	문법 점검 코드	○	○	○	0	0	-	R	NO	0
D1008*	WDT 타이머가 “On”상태일 때 STEP 주소	○	○	○	0	-	-	R	NO	0
D1009	LV 신호의 발생 내역이 D1009 에 저장됨	○	○	×	-	-	-	R	YES	0
D1010*	현재 스캔 시간 (단위: 0.1ms)	○	○	○	0	-	-	R	NO	0
D1011*	최소 스캔 시간 (단위: 0.1ms)	○	○	○	0	-	-	R	NO	0
D1012*	최대 스캔 시간 (단위: 0.1ms)	○	○	○	0	-	-	R	NO	0
D1015*	0~32,767(단위 t: 0.1ms) 고속 접속 타이머의 추가형	×	○	○	0	-	-	R/W	NO	0
D1018*	$\pi$ PI (저 바이트)	○	○	○	H0FDB	H0FDB	H0FDB	R/W	NO	H0FDB
D1019*	$\pi$ PI(고 바이트)	○	○	○	H4049	H4049	H4049	R/W	NO	H4049
D1020*	ES, EX, SS, SA, SX, SC, EH : X0~X7 입력 필터 (단위: ms)	○	○	○	10	-	-	R/W	NO	10
D1021*	ES, EX, SS, SA, SX, SC, EH : X10~ 입력 필터 (단위: ms)	○	○	○	10	-	-	R/W	NO	10
D1022	ES, EX, SS 및 SA, SX, SC 모델의 AB 페이지 계수기의 이중 주파수 선택	○	○	×	0	-	-	R/W	NO	0
D1025*	통신 오류 코드	○	○	○	0	-	-	R	NO	0
D1028	인덱스 레지스터 E0	○	○	○	0	-	-	R/W	NO	0
D1029	인덱스 레지스터 F0	○	○	○	0	-	-	R/W	NO	0
D1030	Y0 펄스의 출력 번호 (Low word)	○	○	×	0	-	-	R	NO	0
D1031	Y0 펄스의 출력 번호 (High word)	○	○	×	0	-	-	R	NO	0
D1032	Y1 펄스의 출력 번호 (Low word)	○	○	×	0	-	-	R	NO	0
D1033	Y1 펄스의 출력 번호 (High word)	○	○	×	0	-	-	R	NO	0
D1034	주파수 측정 카드 동작 모드	×	×	○	-	-	-	R	YES	1
D1035*	RUN/STOP 의 X 입력 지점의 번호 설정	×	×	○	-	-	-	R/W	YES	0
D1036*	COM1(RS-232) 통신 프로토콜	○	○	○	H86	-	-	R/W	NO	H86
D1037	HKY 키 반복 시간(ms)	×	×	○	-	-	-	R/W	YES	0
D1038*	PLC MPU 가 종속장치일 때 데이터 반응 지연 시간의 설정. 시간 단위는 0.1ms 임.	○	○	○	-	-	-	R/W	YES	0
D1039*	지속적인 스캔 시간(ms)	○	○	○	0	-	-	R/W	NO	0
D1040	STEP 지점 S 의 “On” 상태 번호 1	○	○	○	0	-	-	R	NO	0
D1041	STEP 지점 S 의 “On” 상태 번호 2	○	○	○	0	-	-	R	NO	0
D1042	STEP 지점 S 의 “On” 상태 번호 3	○	○	○	0	-	-	R	NO	0
D1043	STEP 지점 S 의 “On” 상태 번호 4	○	○	○	0	-	-	R	NO	0

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1044	STEP point S 의 “On” 상태 번호 5	○	○	○	0	-	-	R	NO	0
D1045	STEP point S 의 “On” 상태 번호 6	○	○	○	0	-	-	R	NO	0
D1046	STEP point S 의 “On” 상태 번호 7	○	○	○	0	-	-	R	NO	0
D1047	STEP point S 의 “On” 상태 번호 8	○	○	○	0	-	-	R	NO	0
D1049	경보 지점의 “On” 상태 번호	×	○	○	0	-	-	R	NO	0
D1050 ↓ D1055	D1050~D1055 는 Modbus 통신 명령어용 레지스터임. PLC 는 D1070~D1085 에 저장된 ASCII 데이터를 HEX 로 자동적으로 변환함.	○	○	○	0	-	-	R	NO	0
D1056*	EX/SX MPU 아날로그 입력 채널 0 (CH0) 및 EH MPU AD 카드 채널 0 (CH0)의 현재 값.	○	×	○	0	-	-	R	NO	0
D1057*	EX/SX MPU 아날로그 입력 채널 1 (CH1) 및 EH MPU AD 카드 채널 1(CH1)의 현재 값.	○	×	○	0	-	-	R	NO	0
D1058*	EX MPU 아날로그 입력 채널 2 (CH2)의 현재 값.	○	×	×	0	-	-	R	NO	0
D1059*	EX MPU 아날로그 입력 채널 3 (CH3)의 현재 값	○	×	×	0	-	-	R	NO	0
D1061	시스템 오류 메시지	○	○	×	-	-	-	R	YES	0
D1067*	알고리즘 오류 코드	○	○	○	0	0	-	R	NO	0
D1068*	알고리즘 오류 주소 잠금	○	○	○	0	-	-	R	NO	0
D1069	플래그 M1065~M1067 와 연계된 오류의 단계 번호	○	○	○	0	-	-	R	NO	0
D1070 ↓ D1085	D1070~D1085 는 Modbus 통신 명령어용 레지스터임. PLC 내장 RS-485 통신 명령어가 수신기로부터 피드백 신호를 수신할 때 그 신호는 레지스터 D1070~D1085 에 저장될 것임. 사용자는 그 레지스터에 지정된 내용을 사용하여 피드백 데이터를 점검할 수 있음	○	○	○	0	-	-	R	NO	0
D1086	DVP-PCC0 용 암호설정 Low word(ASCII 문자에 상응하는 HEX 에 의해 결정됨)	○	○	○	0	-	-	R/W	NO	0
D1087	DVP-PCC01 용 암호설정 Low word (ASCII 문자에 상응하는 HEX 에 의해 결정됨)	○	○	○	0	-	-	R/W	NO	0
D1088	DVP-PCC01 의 복사시간 설정	○	○	○	0	-	-	R/W	NO	0
D1089 ↓ D1099	D1089~D1099 는 Modbus 통신 명령어의 레지스터임. PLC 내장 RS-485 통신 명령어가 실행될 때, 전송신호는 레지스터 D1089~D1099 에 저장될 것임. 사용자는 그 레지스터에 저장된 내용을 사용하여 피드백 데이터를 점검할 수 있음	○	○	○	0	-	-	R	NO	0
D1100	LV 신호가 발생하면, Y0~Y17 는 D1100 의 해당 값에 근거한 출력을 수행할 것임.	×	×	○	0	-	-	R/W	NO	0
D1101*	파일 레지스터의 개시 주소	×	○	○	-	-	-	R/W	YES	0
D1102*	파일 레지스터의 복사 번호	×	○	○	-	-	-	R/W	YES	1600
D1103*	파일 레지스터가 저장하기 위해 개시 D 번호를 설정함(번호는 2000 이상이어야 함.)	×	○	○	-	-	-	R/W	YES	2000
D1104*	가속/감속 펄스 출력 Y0 의 파라미터 인덱스(장치 D 에 해당)	○	○	×	0	-	-	R/W	NO	0
D1109*	COM3 통신 프로토콜 설정	×	×	○	0	-	-	R/W	NO	0
D1110*	EX/SX 시리즈 아날로그 입력 채널 0 (CH 0)	○	○	○	0	-	-	R	NO	0

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
	및 EH 시리즈 AD 카드 채널 0 (CH0)의 평균									
D1111*	EX/SX 시리즈 아날로그 입력 채널 1 (CH 1) 및 EH 시리즈 AD 카드 채널 1 (CH1)의 평균	○	○	○	0	-	-	R	NO	0
D1112*	EX 시리즈 아날로그 입력 채널 2 (CH 2)의 평균	○	×	×	0	-	-	R	NO	0
D1113*	EX 시리즈 아날로그 입력 채널 3 (CH 3)의 평균	○	×	×	0	-	-	R	NO	0
D1116*	EX/SX 시리즈 아날로그 출력 채널 0 (CH 0) 및 EH 시리즈 AD 카드 채널 0 (CH0)	○	○	○	0	0	0	R/W	NO	0
D1117*	EX/SX 시리즈 아날로그 출력 채널 0 (CH0) 및 EH 시리즈 AD 카드 채널 0 (CH0)	○	○	○	0	0	0	R/W	NO	0
D1118*	아날로그 입력의 SX/EX/EH 샘플링 시간(ms)	○	×	×	5	-	-	R/W	NO	5
D1120	COM2 (RS-485) 통신 프로토콜	○	○	○	H86	-	-	R/W	NO	H86
D1121	PLC 통신 주소(PLC 통신 주소를 저장하는 주소로서 래치됨)	○	○	○	-	-	-	R/W	YES	1
D1122	전송 데이터의 잔여 단어	○	○	○	0	0	0	R	NO	0
D1123	수신 데이터의 잔여 단어	○	○	○	0	0	0	R	NO	0
D1124	개시 문자 정의(STX)	○	○	○	H3A	-	-	R/W	NO	H3A
D1125	최초 종료 문자 정의(ETX1)	○	○	○	H0D	-	-	R/W	NO	H0D
D1126	두번째 종료 문자 정의(ETX2)	○	○	○	H0A	-	-	R/W	NO	H0A
D1129	RS-485 타임 아웃 설정(ms)	○	○	○	0	-	-	R/W	NO	0
D1130	MODBUS 복귀 오류 코드 기록	○	○	○	0	-	-	R	NO	0
D1133*	특수 고속 펄스 출력 레지스터(D) 인덱스	×	○	×	0	-	-	R/W	NO	0
D1137*	운용자 오류 주소 발생	○	○	○	0	0	-	R	NO	0
D1140*	특수 확장 모듈 번호, 최대 8 비트	○	○	○	0	-	-	R	NO	0
D1142	확장 유니트의 입력 지점(X)	○	○	○	0	-	-	R	NO	0
D1143	확장 유니트의 출력 지점 (Y)	○	○	○	0	-	-	R	NO	0
D1144*	조정가능 기율기의 가속/감속 펄스 출력의 파라미터 인덱스용 레지스터 D 의 개시 번호	×	○	×	0	-	-	R/W	NO	0
D1147	MEMORY CARD TYPE b0=0: 플래스 카드 사용 안함(H0000) b0=1: 플래스 카드 사용 b8=0: 플래스 카드용 스위치는 Off (H0001) b8=1: 플래스 카드용 스위치는 On (H0101)	×	×	○	#	-	-	R	NO	0
D1149	메모리 카드 유형: 0: 카드 없음, 1: RS-232, DU-01, 2: RS-422, 3: COM3, 4: 전위차계 스위치, 5: DIP 스위치, 6: 트랜지스터 출력 카드, 8: 2AD 카드, 9: 2DA 카드, 10:주파수 측정 카드	×	×	○	#	-	-	R	NO	0
D1150	DHSZ 명령어의 다그룹 설정 비교 모드의 표 계수 레지스터	×	×	○	0	0	0	R	NO	0
D1151	DHSZ 명령어의 주파수 제어 모드에서의 표 계수 레지스터	×	×	○	0	0	0	R	NO	0
D1152	DHSZ D 의 High word 변경 값	×	×	○	0	0	0	R	NO	0
D1153	DHSZ D 의 Low word 변경 값	×	×	○	0	0	0	R	NO	0

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1154*	조정가능 기율기의 가속/감속 펄스 출력 Y0의 가속 시간의 권고된 간격(10~32767 ms)	×	○	×	200	-	-	R/W	NO	200
D1155*	조정가능 기율기의 가속/감속 펄스 출력 Y0의 감속 시간의 권고된 간격(-1~ -32700 ms)	×	○	×	-1000	-	-	R/W	NO	-1000
D1156 ↓ D1165	RTMU 명령어 (K0~K9)가 표시하는 특수 D	×	×	○	0	-	-	R/W	NO	0
D1166	X10 상승 에지 또는 하강 에지용 모드 선택(SC 모델 전용)	×	○	×	0	-	-	R/W	NO	0
D1167	X11 상승 에지 또는 하강 에지용 모드 선택(SC 모델 전용)	×	○	×	0	-	-	R/W	NO	0
D1168	(I150) RS 통신 명령어를 사용하기 위한 특수 문자를 수신할 때 차단요청	○	○	○	0	-	-	R/W	NO	0
D1169	(I160) RS 통신 명령어를 사용하기 위한 특수 길이를 수신할 때 차단요청	×	×	○	0	-	-	R/W	NO	0
D1170*	신호 단계를 실행할 때의 PC 값	×	×	○	0	0	0	R	NO	0
D1172*	2-페이즈 펄스 출력 주파수(12Hz~20KHz)	×	○	×	0	-	-	R/W	NO	0
D1173*	2-페이즈 펄스 출력 모드 선택(K1 및 K2)	×	○	×	0	-	-	R/W	NO	0
D1174*	2-페이즈 펄스 출력용 목표 번호(저 16-bit)	×	○	×	0	-	-	R/W	NO	0
D1175*	2-페이즈 펄스 출력용 목표 번호(고 16-bit)	×	○	×	0	-	-	R/W	NO	0
D1176*	2-페이즈 펄스의 현재 출력 번호(저 16-bit)	×	○	×	0	-	-	R/W	NO	0
D1177*	2-페이즈 펄스의 현재 출력 번호(고 16-bit)	×	○	×	0	-	-	R/W	NO	0
D1178*	VR0 값	×	○	○	0	-	-	R	NO	0
D1179*	VR1 값	×	○	○	0	-	-	R	NO	0
D1182	지시자 레지스터 E1	×	○	○	0	-	-	R/W	NO	0
D1183	지시자 레지스터 F1	×	○	○	0	-	-	R/W	NO	0
D1184	지시자 레지스터 E2	×	○	○	0	-	-	R/W	NO	0
D1185	지시자 레지스터 F2	×	○	○	0	-	-	R/W	NO	0
D1186	지시자 레지스터 E3	×	○	○	0	-	-	R/W	NO	0
D1187	지시자 레지스터 F3	×	○	○	0	-	-	R/W	NO	0
D1188	지시자 레지스터 E4	×	×	○	0	-	-	R/W	NO	0
D1189	지시자 레지스터 F4	×	×	○	0	-	-	R/W	NO	0
D1190	지시자 레지스터 E5	×	×	○	0	-	-	R/W	NO	0
D1191	지시자 레지스터 F5	×	×	○	0	-	-	R/W	NO	0
D1192	지시자 레지스터 E6	×	×	○	0	-	-	R/W	NO	0
D1193	지시자 레지스터 F6	×	×	○	0	-	-	R/W	NO	0
D1194	지시자 레지스터 E7	×	×	○	0	-	-	R/W	NO	0
D1195	지시자 레지스터 F7	×	×	○	0	-	-	R/W	NO	0
D1196	디스플레이의 내용 설정(SX 모델 전용)	×	○	×	0	-	-	R/W	NO	0
D1200*	M0~M999 보조 중계의 주소 개시	×	○	○	-	-	-	R/W	YES	#

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1201*	M0~M999 보조 중계의 주소 종료	×	○	○	-	-	-	R/W	YES	999
D1202*	M2000~M4095 보조 중계의 주소 래치 개시	×	○	○	-	-	-	R/W	YES	2000
D1203*	M2000~M4095 보조 중계의 주소 래치 종료	×	○	○	-	-	-	R/W	YES	4095
D1204*	100ms 타이머 T0~T199 의 래치 주소 개시	×	×	○	-	-	-	R/W	YES	-1
D1205*	100ms 타이머 T0~T199 의 래치 주소 종료	×	×	○	-	-	-	R/W	YES	-1
D1206*	10ms 타이머 T200~T239 의 래치 주소 개시	×	×	○	-	-	-	R/W	YES	-1
D1207*	10ms 타이머 T200~T239 의 래치 주소 종료	×	×	○	-	-	-	R/W	YES	-1
D1208*	16-bit 계수기 C0~C199 의 래치 주소 개시	×	○	○	-	-	-	R/W	YES	#
D1209*	16-bit 계수기 C0~C199 의 래치 주소 종료	×	○	○	-	-	-	R/W	YES	199
D1210*	32-bit 계수기 C200~C234 의 래치 주소 개시	×	○	○	-	-	-	R/W	YES	#
D1211*	32-bit 계수기 C200~C234 의 래치 주소 종료	×	○	○	-	-	-	R/W	YES	234
D1212*	32-bit 고속계수기 C235~C255 의 래치 주소 개시	×	○	○	-	-	-	R/W	YES	235
D1213*	32-bit 고속계수기 C235~C255 의 래치 주소 개시	×	○	○	-	-	-	R/W	YES	255
D1214*	단계 지점(S0~S1023)의 래치 주소 개시	×	○	○	-	-	-	R/W	YES	#
D1215*	단계 지점(S0~S1023)의 래치 주소 종료	×	○	○	-	-	-	R/W	YES	#
D1216*	레지스터 D0~D999 의 래치 주소 개시	×	○	○	-	-	-	R/W	YES	200
D1217*	레지스터 D0~D999) 의 래치 주소 종료	×	○	○	-	-	-	R/W	YES	999
D1218*	레지스터 D2000~D9999 D9 의 래치 주소 개시	×	○	○	-	-	-	R/W	YES	2000
D1219*	레지스터 D2000~D9999 의 래치 주소 종료	×	○	○	-	-	-	R/W	YES	#
D1220	최초 그룹의 펄스 출력 페이즈	×	×	○	0	-	-	R/W	NO	0
D1221	두번째 그룹의 펄스 출력 페이즈	×	×	○	0	-	-	R/W	NO	0
D1222	DRVI, DDRVI, DRVA, DDRVA, PLSV, 및 DPLSV 용 방향 신호와 펄스 출력간 CH0 의 시차 설정.	×	×	○	0	-	-	R/W	NO	0
D1223	DRVI, DDRVI, DRVA, DDRVA, PLSV, 및 DPLSV 용 방향 신호와 펄스 출력간 CH1 의 시차 설정	×	×	○	0	-	-	R/W	NO	0
D1225	최초 그룹의 계수기 계수 설정(HHSC0).	×	×	○	2	-	-	R/W	NO	2
D1226	두번째 그룹의 계수기 계수 설정(HHSC1).	×	×	○	2	-	-	R/W	NO	2
D1227	세번째 그룹의 계수기 계수 설정(HHSC2).	×	×	○	2	-	-	R/W	NO	2
D1228	네번째 그룹의 계수기 계수 설정(HHSC3)..	×	×	○	2	-	-	R/W	NO	2
D1256 ↓ D1295	RS-485 의 MODRW 명령어는 내장됨. 실행동안 송신된 문자는 D1256~D1295 에 저장됨. 사용자는 이들 레지스터의 내용에 따라 점검할 수 있음.	○	○	○	0	-	-	R	NO	0
D1296 ↓ D1311	RS-485 MODRW 명령어는 내장됨. PLC 시스템은 사용자가 표시하는 레지스터의 내용에 있는 ASCII 를 HEX 로 변환하여 D1296 - D1311 에 저장할 것임.	○	○	○	0	-	-	R	NO	0
D1313*	실제 타임 클록(RTC) 초 00~59	×	○	○	0	-	-	R/W	NO	0

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1314*	실제 타임 클록(RTC) 분 00~59	×	○	○	0	-	-	R/W	NO	0
D1315*	실제 타임 클록(RTC) 시 00~23	×	○	○	0	-	-	R/W	NO	0
D1316*	실제 타임 클록(RTC) 일 01~31	×	○	○	0	-	-	R/W	NO	1
D1317*	실제 타임 클록(RTC) 월 01~12	×	○	○	0	-	-	R/W	NO	1
D1318*	실제 타임 클록(RTC) 주 1~7	×	○	○	0	-	-	R/W	NO	6
D1319*	실제 타임 클록(RTC) 년 00 - 99	×	○	○	0	-	-	R/W	NO	0
D1320*	최초의 특수 확장 모듈 ID	×	×	○	0	-	-	R	NO	0
D1321*	두번째의 특수 확장 모듈 ID	×	×	○	0	-	-	R	NO	0
D1322*	세번째의 특수 확장 모듈 ID	×	×	○	0	-	-	R	NO	0
D1323*	네번째의 특수 확장 모듈 ID	×	×	○	0	-	-	R	NO	0
D1324*	다섯번째의 특수 확장 모듈 ID	×	×	○	0	-	-	R	NO	0
D1325*	여섯번째의 특수 확장 모듈 ID	×	×	○	0	-	-	R	NO	0
D1326*	일곱번째의 특수 확장 모듈 ID	×	×	○	0	-	-	R	NO	0
D1327*	여덟번째의 특수 확장 모듈 ID	×	×	○	0	-	-	R	NO	0
D1328	CH0 (Y0,Y1) 오프셋 펄스 번호(Low word)	×	×	○	0	-	-	R/W	NO	0
D1329	CH0 (Y0,Y1) 오프셋 펄스 번호(High word)	×	×	○	0	-	-	R/W	NO	0
D1330	CH1 (Y2,Y3) 오프셋 펄스 번호(Low word)	×	×	○	0	-	-	R/W	NO	0
D1331	CH1 (Y2,Y3) 오프셋 펄스 번호(High word)	×	×	○	0	-	-	R/W	NO	0
D1332	CH0 (Y0,Y1) 잔여 펄스 번호(Low word)	×	×	○	0	-	-	R	NO	0
D1333	CH0 (Y0,Y1) 잔여 펄스 번호(High word)	×	×	○	0	-	-	R	NO	0
D1334	CH1 (Y2,Y3) 잔여 펄스 번호(Low word)	×	×	○	0	-	-	R	NO	0
D1335	CH1 (Y2,Y3) 잔여 펄스 번호(High word)	×	×	○	0	-	-	R	NO	0
D1336	CH0 (Y0,Y1) 펄스의 현재 값 (Low word)	×	×	○	0	0	0	R	NO	0
D1337	CH0 (Y0,Y1) 펄스의 현재 값 (High word)	×	×	○	0	0	0	R	NO	0
D1338	CH1(Y2,Y3) 펄스의 현재 값 (Low word)	×	×	○	0	0	0	R	NO	0
D1339	CH1 (Y2,Y3) 펄스의 현재 값 (High word)	×	×	○	0	0	0	R	NO	0
D1340	CH0 (Y0,Y1) 최초 단계 개시 주파수 및 최종 단계 종료 주파수	×	○	○	-	-	-	R/W	YES	200
D1341	최대 출력 주파수(Low word) (200KHz 로 고정)	×	×	○	H04D0	-	-	R	YES	H04D0
D1342	최대 출력 주파수(High word) (200KHz 로 고정)	×	×	○	3	-	-	R	YES	3
D1343	CH (Y0, Y1) 가속/감속 시간	×	○	○	-	-	-	R/W	YES	100
D1344	CH0 (Y0,Y1) 보충 펄스 번호(Low word)	×	×	○	-	-	-	R/W	YES	0
D1345	CH0 (Y0,Y1) 보충 펄스 번호(High word)	×	×	○	-	-	-	R/W	YES	0
D1346	CH1 (Y2,Y3) 보충 펄스 번호(Low word)	×	×	○	-	-	-	R/W	YES	0

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1347	CH1 (Y2,Y3) 보충 펄스 번호(High word)	×	×	○	-	-	-	R/W	YES	0
D1348	CH0 (Y10) 현재 펄스 값(Low word)	×	○	×	0	-	-	R	NO	0
D1349	CH0 (Y10) 현재 펄스 값(High word)	×	○	×	0	-	-	R	NO	0
D1350	CH1 (Y11) 현재 펄스 값(Low word)	×	○	×	0	-	-	R	NO	0
D1351	CH1 (Y11) 현재 펄스 값(High word)	×	○	×	0	-	-	R	NO	0
D1352	CH1 (Y2,Y3) 최초 단계 개시 주파수 및 최종 단계 종료 주파수(가속 주파수)	×	○	○	-	-	-	R/W	YES	200
D1353	CH1 (Y2, Y3) 가속/감속 시간	×	○	○	-	-	-	R/W	YES	100
D1355*	PLC LINK SLAVE ID 1 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1356*	PLC LINK SLAVE ID 2 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1357*	PLC LINK SLAVE ID 3 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1358*	PLC LINK SLAVE ID 4 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1359*	PLC LINK SLAVE ID 5 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1360*	PLC LINK SLAVE ID 6 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1361*	PLC LINK SLAVE ID 7 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1362*	PLC LINK SLAVE ID 8 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1363*	PLC LINK SLAVE ID 9 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1364*	PLC LINK SLAVE ID 10 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1365*	PLC LINK SLAVE ID 11 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1366*	PLC LINK SLAVE ID 12 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1367*	PLC LINK SLAVE ID 13 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1368*	PLC LINK SLAVE ID 14 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1369*	PLC LINK SLAVE ID 15 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1370*	PLC LINK SLAVE ID 16 가 판독하는 통신 주소	×	○	○	H1064	-	-	R/W	NO	H1064
D1399	PLC LINK 는 개시된 SLAVE 국의 ID 번호 표시함.	×	○	○	1	-	-	R/W	NO	1
D1415*	PLC LINK SLAVE ID 1 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1416*	PLC LINK SLAVE ID 2 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1417*	PLC LINK SLAVE ID 3 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1418*	PLC LINK SLAVE ID 4 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1419*	PLC LINK SLAVE ID 5 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1420*	PLC LINK SLAVE ID 6 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1421*	PLC LINK SLAVE ID 7 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1422*	PLC LINK SLAVE ID 8 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8



특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1423*	PLC LINK SLAVE ID 9 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1424*	PLC LINK SLAVE ID 10 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1425*	PLC LINK SLAVE ID 11 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1426*	PLC LINK SLAVE ID 12 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1427*	PLC LINK SLAVE ID 13 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1428*	PLC LINK SLAVE ID 14 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1429*	PLC LINK SLAVE ID 15 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1430*	PLC LINK SLAVE ID 16 가 작성하는 통신 주소	×	○	○	H10C8	-	-	R/W	NO	H10C8
D1431*	PLC LINK 횟수	×	○	○	0	-	-	R/W	NO	0
D1432*	PLC LINK 계수	×	○	○	0	-	-	R/W	NO	0
D1433*	PLC LINK 유니트	×	○	○	0	-	-	R/W	NO	0
D1434*	PLC LINK SLAVE ID 1 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1435*	PLC LINK SLAVE ID 2 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1436*	PLC LINK SLAVE ID 3 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1437*	PLC LINK SLAVE ID 4 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1438*	PLC LINK SLAVE ID 5 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1439*	PLC LINK SLAVE ID 6 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1440*	PLC LINK SLAVE ID 7 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1441*	PLC LINK SLAVE ID 8 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1442*	PLC LINK SLAVE ID 9 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1443*	PLC LINK SLAVE ID 10 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1444*	PLC LINK SLAVE ID 11 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1445*	PLC LINK SLAVE ID 12 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1446*	PLC LINK SLAVE ID 13 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1447*	PLC LINK SLAVE ID 14 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1448*	PLC LINK SLAVE ID 15 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1449*	PLC LINK SLAVE ID 16 의 판독 항목	×	○	○	16	-	-	R/W	NO	16
D1450*	PLC LINK SLAVE ID 1 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1451*	PLC LINK SLAVE ID 2 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1452*	PLC LINK SLAVE ID 3 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1453*	PLC LINK SLAVE ID 4 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1454*	PLC LINK SLAVE ID 5 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1455*	PLC LINK SLAVE ID 6 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1456*	PLC LINK SLAVE ID 7 의 작성 항목	×	○	○	16	-	-	R/W	NO	16

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1457*	PLC LINK SLAVE ID 8 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1458*	PLC LINK SLAVE ID 9 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1459*	PLC LINK SLAVE ID 10 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1460*	PLC LINK SLAVE ID 11 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1461*	PLC LINK SLAVE ID 12 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1462*	PLC LINK SLAVE ID 13 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1463*	PLC LINK SLAVE ID 14 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1464*	PLC LINK SLAVE ID 15 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1465*	PLC LINK SLAVE ID 16 의 작성 항목	×	○	○	16	-	-	R/W	NO	16
D1466	CH0 모터 1 회전을 위한 필요 펄스 번호(Low word)	×	×	○	-	-	-	R	YES	2000
D1467	CH0 모터 1 회전을 위한 필요 펄스 번호 (High word)	×	×	○	-	-	-	R	YES	0
D1468	CH1 모터 1 회전을 위한 필요 펄스 번호 (Low word)	×	×	○	-	-	-	R	YES	2000
D1469	CH1 모터 1 회전을 위한 필요 펄스 번호 (High word)	×	×	○	-	-	-	R	YES	0
D1470	CH0 모터 1 회전을 위한 이동 거리(Low word)	×	×	○	-	-	-	R	YES	1000
D1471	CH0 모터 1 회전을 위한 이동 거리(High word)	×	×	○	-	-	-	R	YES	0
D1472	CH1 모터 1 회전을 위한 이동 거리 (Low word)	×	×	○	-	-	-	R	YES	1000
D1473	CH1 모터 1 회전을 위한 이동 거리(High word)	×	×	○	-	-	-	R	YES	0
D1474	CH0 기계적 이동 단위(Low word)	×	×	○	-	-	-	R	YES	0
D1475	CH0 기계적 이동 단위 (High word)	×	×	○	-	-	-	R	YES	0
D1476	CH1 기계적 이동 단위 (Low word)	×	×	○	-	-	-	R	YES	0
D1477	CH1 기계적 이동 단위 (High word)	×	×	○	-	-	-	R	YES	0
D1480* ↓ D1495*	SLAVE ID 1 LINK PLC 의 관독. SLAVE ID 1 관독용 통신 주소는 1355 에 있고 그 범위는 SLAVE ID 1 PLC 의 D100-D115 임	×	○	○	0	-	-	R	NO	0
D1496* ↓ D1511*	SLAVE ID 1 LINK PLC 의 작성. SLAVE ID 1 작성용 통신 주소는 D1415 에 있고 그 범위는 SLAVE ID 1 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1512* ↓ D1527*	1353=Off 일 때, SLAVE ID 2 LINK PLC 의 관독. SLAVE ID 2 관독용 통신 주소는 D1356 에 있고 그 범위는 SLAVE ID 2 PLC 의 D100-D115 임. EH 시리즈 MPU 전용. M1353=On 일 때, SLAVE ID17~32 가 관독한 통신 주소는 D1512~1527 에 저장될 것임.	×	○	○	0	-	-	R	NO	0
D1528* ↓ D1543*	M1353=Off 일 때, SLAVE ID 2 LINK PLC 의 작성. SLAVE ID 2 작성용 통신 주소는 D1416 에 있고 그 범위는 SLAVE ID 2 PLC 의 D200-D215. EH 시리즈 MPU 전용. M1353=On 일 때,	×	○	○	0	-	-	R/W	NO	0

특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
	D1528~1543 에 저장된 통신 주소는 SLAVE ID17~32 에 작성됨.									
D1544* ↓ D1559*	M1353=Off 일 때, SLAVE ID 3 LINK PLC 의 판독. SLAVE ID 3 판독용 통신 주소는 D1357 에 있고 그 범위는 SLAVE ID 3 PLC 의 D100-D115 임. EH 시리즈 MPU 전용. M1353=On 일 때, PLC 가 SLAVE ID17~32 로부터 판독하기 위한 데이터 길이는 D1544~1559 에 저장됨..	×	○	○	0	-	-	R	NO	0
D1560* ↓ D1575*	M1353=Off 일 때, SLAVE ID 3 LINK PLC 의 작성. SLAVE ID 3 작성용 통신 주소는 D1417 에 있고 그 범위는 SLAVE ID 3 PLC 의 D200-D215 임. EH 시리즈 MPU 전용. M1353=On 일 때, PLC 가 SLAVE ID 17~32 에 작성하기 위한 데이터 길이는 D1560~1575.에 저장됨.	×	○	○	0	-	-	R/W	NO	0
D1576* ↓ D1591*	1353=Off 일 때, SLAVE ID 4 LINK PLC 의 판독. SLAVE ID 4 판독용 통신 주소는 D1358.에 있고 그 범위는 SLAVE ID 4 PLC 의 D100-D115 임. EH 시리즈 MPU 전용. M1353=On 일 때, PLC 가 SLAVE ID17~32 로부터 데이터 저장을 위한 개시 레지스터는 D1576~1591 에 저장됨.	×	○	○	0	-	-	R	NO	0
D1592* ↓ D1607*	M1353=Off 일 때, SLAVE ID 4 LINK PLC 의 작성. SLAVE ID 4 작성용 통신 주소는 D1418 에 있고 그 범위는 SLAVE ID 4 PLC 의 D200-D215 임. EH 시리즈 MPU 전용. M1353=On 일 때, PLC 가 SLAVE ID17~32 에 작성한 데이터를 저장하기 위한 개시 레지스터는 D1592~1607 에 저장됨.	×	○	○	0	-	-	R/W	NO	0
D1608* ↓ D1623*	SLAVE ID 5 LINK PLC 판독. SLAVE ID 5 판독용 통신 주소는 D1359 에 있고 그 범위는 SLAVE ID 5 PLC 의 D100-D115 임 .	×	○	○	0	-	-	R	NO	0
D1624* ↓ D1639*	SLAVE ID 5 LINK PLC 의 작성 SLAVE ID 5 작성용 통신 주소는 D1419 에 있고 그 범위는 SLAVE ID 5 PLC 의 D200-D215 임	×	○	○	0	-	-	R/W	NO	0
D1640* ↓ D1655*	SLAVE ID 6 LINK PLC 의 판독. SLAVE ID 6 판독용 통신 주소는 D1360 에 있고 그 범위는 SLAVE ID 6 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1656* ↓ D1671*	SLAVE ID 6 LINK PLC 의 작성. SLAVE ID 6 작성용 통신 주소는 D1420 에 있고 그 범위는 SLAVE ID 6 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1672* ↓ D1687*	SLAVE ID 7 LINK PLC 의 판독. SLAVE ID 7 판독용 통신 주소는 D1361 에 있고 그 범위는 SLAVE ID 7 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1688* ↓ D1703*	SLAVE ID 7 LINK PLC 의 작성. SLAVE ID 7 작성용 통신 주소는 D1421 에 있고 그 범위는 SLAVE ID 7 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1704* ↓ D1719*	SLAVE ID 8 LINK PLC 의 판독. SLAVE ID 8 판독용 통신 주소는 D1362 에 있고 그 범위는 SLAVE ID 8 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0

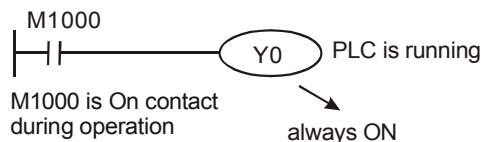
특수 D	기능	ES EX SS	SA SX SC	EH	꺼짐 ↓ 꺼짐	정지 ↓ 실행	실행 ↓ 정지	속성	래치	공장 설정
D1720* ↓ D1735*	SLAVE ID 8 LINK PLC 의 작성. SLAVE ID 8 작성용 통신 주소는 D1422 에 있고 그 범위는 SLAVE ID 8 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1736* ↓ D1751*	SLAVE ID 9 LINK PLC 의 판독. SLAVE ID 9 판독용 통신 주소는 D1363 에 있고 그 범위는 SLAVE ID 9 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1752* ↓ D1767*	SLAVE ID 9 LINK PLC 의 작성. ID 9 작성용 통신 주소는 D1423 에 있고 그 범위는 ID 9 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1768* ↓ D1783*	SLAVE ID 10 LINK PLC 의 판독. SLAVE ID 10 판독용 통신 주소는 D1364 에 있고 그 범위는 SLAVE ID 10 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1784* ↓ D1799*	SLAVE ID 10 LINK PLC 의 작성. SLAVE ID 10 작성용 통신 주소는 D1424 에 있고 그 범위는 SLAVE ID 10 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1800* ↓ D1815*	SLAVE ID 11 LINK PLC 의 판독. ID 11 판독용 통신 주소는 D1365 에 있고 그 범위는 ID 11 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1816* ↓ D1831*	SLAVE ID 11 LINK PLC 의 작성. SLAVE ID 11 작성용 통신 주소는 D1425 에 있고 그 범위는 SLAVE ID 11 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1832* ↓ D1847*	SLAVE ID 12 LINK PLC 의 판독. ID 12 판독용 통신 주소는 D1366 에 있고 그 범위는 ID 12 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1848* ↓ D1863*	SLAVE ID 12 LINK PLC 의 작성. SLAVE ID 12 작성용 통신 주소는 D1426 에 있고 그 범위는 SLAVE ID 12 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1864* ↓ D1879*	SLAVE ID 13 LINK PLC 의 판독. ID 13 판독용 통신 주소는 D1367 에 있고 그 범위는 ID 13 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1880* ↓ D1895*	SLAVE ID 13 LINK PLC 의 작성. SLAVE ID 13 작성용 통신 주소는 D1427 에 있고 그 범위는 SLAVE ID 13 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1896* ↓ D1911*	SLAVE ID 14 LINK PLC 의 판독. ID 14 판독용 통신 주소는 D1368 에 있고 그 범위는 ID 14 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1912* ↓ D1927*	SLAVE ID 14 LINK PLC 의 작성. SLAVE ID 14 작성용 통신 주소는 D1428 에 있고 그 범위는 SLAVE ID 14 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1928* ↓ D1943*	SLAVE ID 15 LINK PLC 의 판독. SLAVE ID 15 판독용 통신 주소는 D1369 에 있고 그 범위는 SLAVE ID 15 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1944* ↓ D1959*	SLAVE ID 15 LINK PLC 의 작성. SLAVE ID 15 작성용 통신 주소는 D1429 에 있고 그 범위는 SLAVE ID 15 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0
D1960* ↓ D1975*	SLAVE ID 16 LINK PLC 의 판독. SLAVE ID 16 판독용 통신 주소는 D1370 에 있고 그 범위는 SLAVE ID 16 PLC 의 D100-D115 임.	×	○	○	0	-	-	R	NO	0
D1976* ↓ D1991*	SLAVE ID 16 LINK PLC 의 작성. SLAVE ID 16 작성용 통신 주소는 D1430 에 있고 그 범위는 SLAVE ID 16 PLC 의 D200-D215 임.	×	○	○	0	-	-	R/W	NO	0

## 2.11 특수 보조 릴레이와 특수 레지스터 기능

기능 그룹	PLC 개시 플래그 (Operation Flag)
번호	M1000~M1003

### M1000:

M1000 은 작동 중에 개접점 (On contact), 즉 상개 접점 a (normally open contact a) 입니다. M1000 를 이용하여 작동 중에 램프를 켜려면 사용자는 PLC 가 RUN 상태라는 것을 알 수 있을 것입니다. M1000 은 PLC 가 RUN 일 때 항상 열려있습니다.



### M1001:

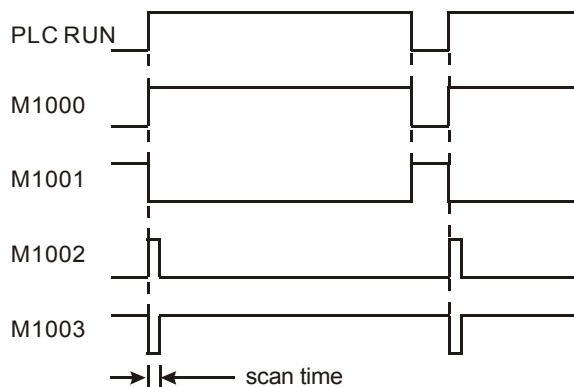
M1001 은 작동중 폐접점 (Off contact), 즉 상폐 접점 b(normally close contact b) 입니다. M1001 은 PLC 가 RUN 일 때 항상 닫혀 있습니다.

### M1002:

M1002 는 PLC 가 RUN 을 시작할 때 첫 스캔까지는 열려 있다가 그 뒤에는 닫혀 있습니다. M1002 는 스캔 초기 펄스 (scan initial pulse)로 생각할 수 있으며 펄스 폭은 스캔 시간입니다. 양성 펄스 (positive pulse)를 시작할 때 쓸 수 있습니다

### M1003:

PLC 가 RUN 일 때 첫 스캔에서는 닫혔다가 나중에는 열리게 됩니다. 즉, 음성 펄스를 시작합니다. (RUN 이라면 일단 닫힘)

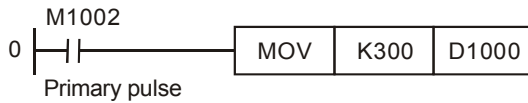


기능 그룹	모니터 타이머
번호	D1000

### 내용:

1. 모니터 타이머는 PLC 스캔 시간을 모니터 할 때 쓰입니다. 스캔 타임이 모니터 타이머의 설정 시간을 넘을 때에는 빨간 오류 LED (RED ERROR LED) 만 켜지고 다른 출력은 모두 꺼지게 됩니다.

2. 모니터 타이머의 초기 값은 **200ms** 입니다. 사용자는 MOV 명령을 사용하여 프로그램이 길거나 계산이 복잡한 경우 모니터 타이머의 설정을 변경할 수 있습니다. 아래의 예는 모니터 타이머를 300ms 로 설정하는 것을 보여 줍니다.



3. 모니터 타이머의 최대 설정은 32,767 ms 입니다. 그러나 모니터 타이머의 수치가 너무 크게 설정되면 계산 시간이 비정상적으로 늦어질 수 있습니다. 따라서 200ms 가 넘는 스캔 타임이 필요할 만큼 복잡한 계산이 아니라면 모니터 타이머를 200ms 내외로 설정하는 것이 좋습니다.
4. D1010 ~ D1012 를 모니터하여 계산이 복잡하거나 PLC MPU 가 너무 많은 모듈들을 연결하고 있어서 스캔 타임이 D1000 설정값을 넘는 일은 없는지 확인해 주십시오. 만약 이런 일이 있을 경우, D1000 설정을 조정하는 것은 물론, PLC 프로그램에서 WDT 명령 (API 07)을 이용할 수도 있습니다. CPU 가 WDT 명령을 실행하고 있을 때 내부 모니터 타이머는 0 으로 초기화되어 스캔 타임이 모니터 타이머 설정을 넘지 않을 수 있도록 합니다.

기능 그룹            제품의 능력  
번호                    D1002

**내용:**

시리즈마다 프로그램의 능력은 다릅니다.

1. ES, EX, SS 시리즈: 3792 Steps
2. SA, SX, SC 시리즈: 7920 Steps
3. EH 시리즈: 15872 Steps

기능 그룹            문법 체크  
번호                    M1004, D1004, D1137

**내용:**

1. 문법 오류가 있으면 PLC ERROR LED 가 깜빡이고 특수 계전기 M1004 는 On 가 됩니다.
2. PLC 문법을 체크하는 때: power 가 Off 에서 On 으로 바뀔 때 그외의 경우:
  - a) WPLSoft 나 HPP 로 PLC 에 프로그램을 입력할 때
  - b) SA/SX/SC/EH 시리즈와 WPLSoft 를 이용해 온라인 프로그래밍 기능을 사용할 때
3. 틀린 문자나 기기 또는 문법 오류가 있을 시 기능이 나타납니다. 특수 레지스터 D1004 에 대하여 오류 코드 정보를 체크한다면 오류를 찾을 수 있습니다. 오류 주소는 데이터 레지스터 D1137 에 저장됩니다. (일단 회로 오류일 경우에는 D1137 은 해당하지 않습니다.)
4. 2.12 장의 고장 해결과 오류 정보 (Troubleshooting and Fault Information)에서 문법 체크를 참조하십시오.

기능 그룹            자료 백업 메모리  
번호                    M1005~M1007

### 내용

자료 백업 메모리 카드가 EH MPU 에 설치되어 있다면 MPU 는 자료 백업 메모리 카드 스위치 접점의 ON/OFF 에 따라 작동할 것입니다. MPU 와 메모리 카드를 비교할 때 스위치가 ON 이라면, M1005~M1007 는 비교 문제가 생길 당시 트리거되거나 (자세한 사항은 아래를 참조하세요), 메모리 카드의 데이터가 비교 문제 없이 MPU 에 복사됩니다. 스위치가 OFF 라면 MPU 는 메모리 카드에 대해 아무런 작용도 하지 않습니다.

**M1005 :** M1005 가 On 일 때, MPU 와 데이터 백업 메모리 카드 간의 비교 코드가 에러상태이며 MPU 는 아무 작동도 하지 않습니다.

**M1006 :** M1006 가 On 일 때, 데이터 백업 메모리 카드가 초기화 되지않았음을 뜻합니다.

**M1007:** M1007 가 On 일 때 , 데이터 백업 메모리 카드의 프로그램 영역에 데이터가 존재하지 않음을 뜻합니다.

기능 그룹            스캔 시간 종료(timeout) 타이머

번호                    M1008, D1008

### 내용:

1. 작동 중 타이머 시간이 종료되면 PLC ERROR LED 에 불이 들어오고 M1008 는 On 이 됩니다.
2. WPLSoft 나 HPP 를 이용하여 WDT 타이머가 작동하고 있을 때에는 시간 종료 STEP 주소를 저장하는 D1008 를 모니터링합니다.

기능 그룹            스캔 시간 모니터

번호                    D1010~D1012

### 내용:

현재, 최소, 최대 스캔 시간이 D1010~D1012 에 저장됩니다.

D1011: 최소 스캔 시간

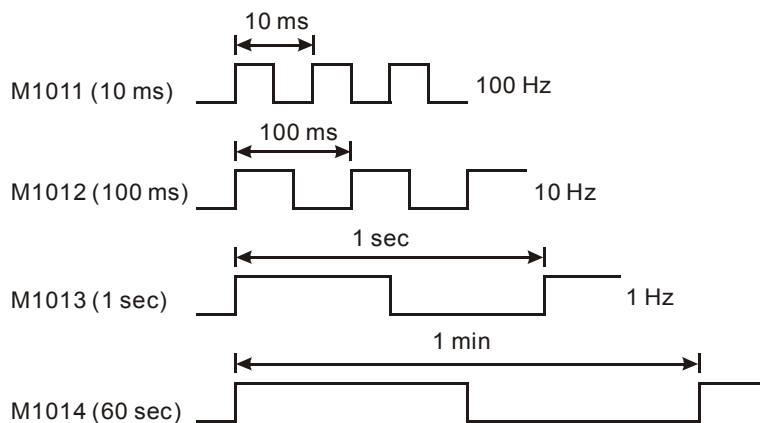
D1012: 최대 스캔 시간

기능 그룹            내부 클록 펄스 (clock pulse)

번호                    M1011~M1014

### 내용:

1. PLC 에는 다음과 같은 4 개의 클록 펄스가 있습니다. PLC 의 전원이 켜지면 이 네개의 클록 펄스는 자동으로 작동하게 됩니다.

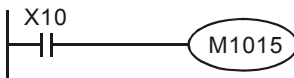


2. PLC 가 STOP 상태이면 클록 펄스도 이에 따라 작동합니다. 클록 펄스의 시작 시점과 RUN 시점은 동시에 이루어 지지 않습니다.

기능 그룹      고속 타이머  
번호      M1015, D1015

내용:

1. 특수 M 과 특수 D 를 직접적으로 사용하는 단계:
  - a) PLC 가 RUN 일 때에만 가능합니다. EH 모델에서는 PLC 가 RUN 일 때에만 가능하나, SA/SX/SC 모델은 RUN 또는 STOP 일 때 모두 가능합니다.
  - b) M1015=On 일 때, PLC 가 스캔에서 END 명령을 실행 완료하는 즉시 고속 타이머 D1015 를 시작합니다. D1015 의 최소 단위는 100us 입니다.
  - c) D1015 의 범위는 0~32,767 입니다. 32,767 까지 세고 나면 0 부터 다시 시작합니다.
  - d) M1015=Off 일 경우, D1015 는 카운팅을 즉시 중단합니다.
2. EH 시리즈에는 고속 타이머 명령 HST 가 있으므로 자세한 사항은 API 196 HST 를 참고하십시오.
3. 예:
  - a) X10 가 On 이면, M1015 를 On 으로 설정하여 고속 타이머를 작동시키고 D1015 에 기록을 시작하도록 합니다.
  - b) X10 가 Off 이면, M1015 를 Off 로 설정하여 고속 타이머를 끕니다.



기능 그룹      실시간 시계  
번호      M1016, M1017, M1076, D1313~D1319

내용:

1. 상대적 명령인 특수 M 과 특수 D.

장치	이름	기능
M1016	실시간 시계의 연도 표시	Off: 제일 우측의 2 개 bit 를 보여줍니다 On: 제일 우측이 2 개 bit + 2000 을 나타냅니다.
M1017	±30 초 조정	Off 에서 On 으로 바뀔 경우 조정됩니다. 0~29 초 이었을 경우 '분'은 바뀌지 않고 초만 0 으로 초기화 됩니다. 30~59 초 이었을 경우 분에 1 을 더하고 초를 0 으로 초기화 합니다.
M1076	실시간 시계 오작동	설정이 일정 범위를 벗어났거나 배터리가 다 소모되었을 경우 on 이 됩니다.
M1082	RTC 플러그 변경	RTC 가 바뀌었을 때 플러그는 ON 이 됩니다.
D1313	초	0~59
D1314	분	0~59
D1315	시	0~23



D1316	일	1~31
D1317	달	1~12
D1318	주	1~7
D1319	년	0~99(제일 오른쪽 2 개 bit)

- 실시간 시계 설정이 잘못되어 있으면 PLC 전원이 다시 켜졌을 때 시계는 2000 년 1 월 1 일 00:00 토요일로 초기화 됩니다.
- 영구 시계의 방법(method) 변경:
  - SA, SX, SC, EH 모드 of 내부 실시간 시계를 조정하려면 TWR 명령을 사용할 수 있습니다. TWR 에 대한 자세한 사항은 API 167 을 참조하세요.
  - DU-01 을 설정할 때에는 주변 WPLSoft 와 디지털 세팅 디스플레이를 사용하십시오.

기능 그룹             $\pi$  (PI)  
 번호                D1018, D1019

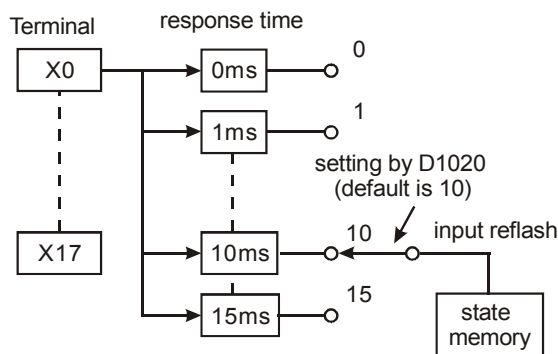
내용:

- 플로팅 포인트 값인  $\pi$  (PI)를 저장하기 위해 D1019 와 D1018 을 통합한 32 비트 데이터 레지스터를 사용합니다.
- 플로팅 포인트 값은 H 40490FDB 입니다.

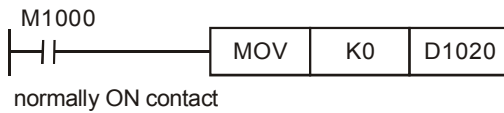
기능 그룹            입력 단말의 반응 시간 조정  
 번호                D1020, D1021

내용:

- SA/SX/SC 시리즈 모델에서는 입력 단말 X0~X7 로부터 펄스를 받는 반응시간이 D1020 의 내용에 의해 설정될 수 있습니다. 설정 범위: 0~20, 단위: ms.
- SC 시리즈 모델에서는 입력 단말 X10~X11 로부터 펄스를 받는 반응시간이 D1021 의 내용에 의해 설정될 수 있습니다. 설정 범위: 0~1000, 단위: 시간 (스캔 사이클)
- EH 시리즈 모델에서는 입력 단말 X0~X7 로부터 펄스를 받는 반응시간이 D1020 의 내용에 의해 설정될 수 있습니다. 설정 범위: 0~60, 단위: ms.
- EH 시리즈 모델에서는 입력 단말 X10~X17 로부터 펄스를 받는 반응시간이 D1021 의 내용에 의해 설정될 수 있습니다. 설정 범위: 0~60, 단위: ms.
- PLC 가 Off→On 로 바뀌면, D1020 와 D1021 의 내용은 자동으로 10 이 됩니다.



6. X0~X7 반응 시간을 0ms 로 설정하여 다음의 프로그램을 실행하려고 할 때, 입력 단말의 빠른 반응 시간은 50  $\mu$ s 이 되는데 이는 RC 필터 회로에 대한 입력 단말 연결이 연속적으로 되어 있기 때문입니다.



7. 고속 카운터를 쓸 때에는 반응 시간을 조정할 필요가 없습니다. 이는 프로그램에서의 입력을 방해하기만 합니다.
8. REFF (API 51) 명령을 사용하는 것이나 D1020 과 D1021 의 내용을 바꾸는 것은 동일한 작업입니다.

기능 그룹            작업 완료 플래그

번호                M1029, M1030

내용:

작업 완료 플래그:

1. API 52 MTR, API 71 HKY, API 72 DSW, API 74 SEGL, API 77 PR:  
명령에 대한 작업이 일단 완료되면 스캔을 위해 M1029=On 이 됩니다.
2. API 57 PLSY, API 59 PLSR:
  - a) SA, SX, SC, ES, EX, SS MPU 에서 Y0 펄스가 출력을 끝낸 뒤 M1029 는 On 이 되고, Y0 가 출력을 완료한 뒤 M1030 은 On 이 됩니다. PLSY 과 PLSR 명령이 Off 이면, M1029 과 M1030 는 Off 입니다.
  - b) EH MPU 에서 M1029 은 Y0 와 Y1 펄스가 출력을 완료한 뒤 On 이 되고, M1030 는 Y2 와 Y3 펄스가 출력을 완료한 뒤 0 이 됩니다. PLSY, PLSR 명령이 Off 이면, M1029 나 M1030 는 Off 입니다.
  - c) M1029 와 M1030 을 실행한 후에는 사용자가 초기화해야 합니다.
3. API 63 INCD: M1029 가 지정된 그룹이 비교를 끝내는 스캔 시간동안 On 입니다.
4. API 67 RAMP, API 69 SORT:
  - a) 작업 후 M1029 는 On 이고, M1029 는 사용자가 초기화해야 합니다.
  - b) 명령이 Off 라면 M1029 는 Off 가 됩니다.
5. EH MPU API 155 DABSR, API 156 ZRN, API 158 DRVI, API 158 DRVA:
  - a) 첫번째 출력 그룹인 Y0, Y1 펄스가 전송을 끝냈을 때 M1029 는 On 입니다. 그리고 두번째 출력 그룹 Y2, Y3 펄스가 전송을 끝내면 M1030 이 On 입니다.
  - b) M1029 나 M1030 는 다음번에 이 명령을 실행할 때 Off 가 되고, 작업을 완료하고 나면 On 이 됩니다.

기능 그룹            통신 에러 코드

번호                D1025

내용:

통신 에러가 생기면 M1025 는 On 이 되고 통신 에러코드가 D1025 에 기록됩니다. 에러 코드는 다음과 같습니다.:

01: 잘못된 명령

02: 잘못된 기기 주소

03: 데이터가 범위를 넘음

07: 검사합계(checksum) 오류

기능 그룹 소거(clear) 명령

번호 M1031, M1032

내용:

M1031 (비래치(unlatched) 부분 소거), M1032 (래치 부분 소거)

장치	소거될 컴포넌트
M1031 비래치 부분 소거	<ul style="list-style-type: none"> <li>■ Y, 일반 M, 일반 S 의 접합</li> <li>■ 일반 코일 및 타임 코일의 T 접점</li> <li>■ 일반 코일, 시간 코일, 리셋 코일의 C 접점</li> <li>■ D 현재 일반 레지스터</li> <li>■ T 현재 일반 레지스터</li> <li>■ C 현재 일반 레지스터</li> </ul>
M1032 래치 부분 소거	<ul style="list-style-type: none"> <li>■ 래치의 M, S 접합</li> <li>■ 누적 타이머 T 접점과 시간 코일</li> <li>■ 래치 C 와 고속 카운터 C 접점, 시간 코일</li> <li>■ 현재 래치 레지스터 D</li> <li>■ 누적 타이머 T 의 현재 레지스터</li> <li>■ 래치 C 와 고속 카운터 C 의 현재 레지스터</li> </ul>

기능 그룹 STOP 모드에 래치된 출력값

번호 M1033

내용:

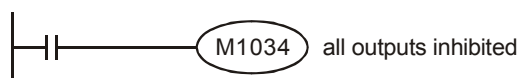
M1003 이 On 일 때, 출력의 On/Off 상태는 PLC 가 RUN 에서 STOP 으로 일단 바뀌면 계속 유지됩니다. PLC 의 출력 접점 로드가 heater 라면, heater 의 상태는 PLC 가 프로그램 변경에 의해 RUN 에서 STOP 그리고 RUN 으로 바뀔 때 고정될 것입니다.

기능 그룹 모든 출력 Y 억제

번호 M1034

내용:

M1034 가 On 으로 설정되면 모든 출력 Y 는 억제됩니다.



기능 그룹 RUN/STOP 스위치

번호 M1035, D1035

### 내용:

1. EH 시리즈에서는, M1035가 On으로 바뀌면 D1035 (0~17)의 내용에 따라 X0~X17 지점에 RUN/STOP을 입력하기 시작하여 스위치를 RUN/STOP 상태로 바꿉니다.
2. SA 시리즈에서는 M0135가 On이 되면 X7에 입력을 시작하여 스위치를 RUN/STOP 상태로 바꿉니다.

### 기능 그룹

통신 포트 기능

### 번호

M1120, M1136, M1138, M1139, M1143, D1036, D1109, D1120

### 내용:

향상된 통신 포트 기능: (SA\_V1.2, SX\_V1.2, SC, EH\_V1.1 이후의 버전에 해당됩니다.)

SC/SA/SX 시리즈 MPU에서 향상된 통신 포트는 다음과 같습니다: RS-232의 COM1, RS-485의 COM2

EH 시리즈 MPU에서 향상된 통신 포트는 다음과 같습니다: RS-232의 COM1와 RS-232/RS-485/RS-422의 COM2.

COM1와 COM2은 모두 최대 속도 115,200bps로 MODBUS와 ASCII/RTU 통신 프로토콜을 지원하며 두 포트는 동시에 사용될 수 있습니다. 또다른 통신 포트인 RS-232/RS485의 COM3은 EH 시리즈 MPU에 적용되어 최대 속도 38,400 bps로 MODBUS와 ASCII를 지원합니다.

- COM1 Slave만을 위한 것이며, 조정가능한 변조 속도(baud rate)를 지니며 ASCII/RTU 통신 프로토콜을 지원합니다. 속도는 최대 115,200 bps이며 데이터 길이까지 (Data bits, Parity bits, Stop bits.)입니다.
- COM2 Master나 Slave로 사용될 수 있으며, 조정가능한 변조 속도를 지니며 ASCII/RTU 통신 프로토콜을 지원합니다. 속도는 최대 115,200 bps 그리고 데이터 길이까지입니다. (Data bits, Parity bits, Stop bits.)
- COM3 Slave만을 위한 것이며, ASCII 통신 프로토콜을 지원합니다. (Data bits, Parity bits, Stop bits) 7, E, 1 조정가능한 변조속도를 지닙니다. 속도는 38,400 bps까지입니다. COM2와 COM3은 동시에 Slave로 사용될 수 없습니다.

### ◎ 통신 프로토콜 설정

- COM1
  1. D1036로 설정하는 통신 프로토콜
  2. M1138 통신 중지
  3. M1139를 이용하여 ASCII/RTU 모드 설정하기
- COM2
  1. D1120로 설정하는 통신 프로토콜
  2. M1120 통신 중지
  3. M1143를 이용하여 ASCII/RTU 모드 설정하기
- COM3
  1. D1109로 설정하는 통신 프로토콜
  2. M1136 통신 중지

D1036 : b8~b15는 COM1의 통신 프로토콜 RS-232 (Slave)을 지원하지 않습니다.

D1109 : b0~b3와 b8~b15는 COM3 통신 프로토콜 RS-232/RS-485 (Slave)을 지원하지 않습니다.

D1120 : COM2 통신 프로토콜 RS-232/RS-485/RS-422 을 (Master and Slave)용도로 지원합니다.

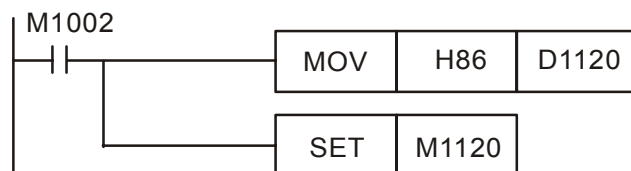
설정을 위해 아래의 표를 참조하세요

	내용	0	1
b0	데이터 길이	b0=0 : 7	b0=1 : 8
b1 b2	패리티 비트	b2, b1=00 : (없음) b2, b1=01 : (홀수) b2, b1=11 : (짝수)	
b3	정지 비트	b3=0 : 1 bit	b3=1 : 2 bit
b4 b5 b6 b7	b7~b4=0001 (H1) : 110 bps b7~b4=0010 (H2) : 150 bps b7~b4=0011 (H3) : 300 bps b7~b4=0100 (H4) : 600 bps b7~b4=0101 (H5) : 1200 bps b7~b4=0110 (H6) : 2400 bps b7~b4=0111 (H7) : 4800 bps b7~b4=1000 (H8) : 9600 bps b7~b4=1001 (H9) : 19200 bps b7~b4=1010 (HA) : 38400 bps b7~b4=1011 (HB) : 57600 bps b7~b4=1100 (HC) : 115200 bps		
b8	시작 문자	b8=0: 없음	b8=1 : D1124
b9	제일 끝 문자	b9=0: 없음	b9=1 : D1125
b10	끝에서 두번째 문자	b10=0: 없음	b10=1 : D1126
b15~b11	기능 없음		

### 예 1: COM2 통신 형태의 변형

사용자가 COM2의 통신형태를 변형하고자 한다면, 프로그램의 앞에 아래의 프로그램을 추가해야 합니다.  
 PLC가 STOP에서 RUN으로 바뀔 때, M1120이 처음에 ON이라면 이를 감지할 것입니다. M1120가 ON이라면 COM2 설정은 D1120에 의해 조정됩니다.

COM2 통신 형태를 ASCII, 9600bps, 7 Data bits, 짝수 패리티, 1 정지 비트 (9600, 7, E1)로 변경하세요



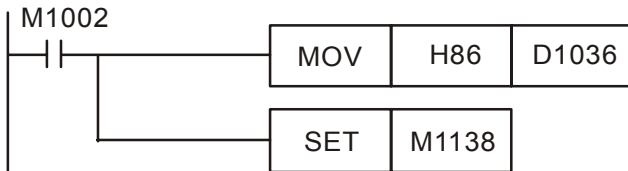
주의:

1. COM2이 Slave로 쓰일 때 아무런 통신 명령도 없도록 하십시오.
2. 통신 형태 조정을 완료한 뒤에는 PLC가 RUN에서 STOP으로 바뀌어도 통신은 변화하지 않습니다.
3. 통신 변형을 완료한 뒤에는 PLC의 통신 형태가 전원이 꺼졌다 켜지면 초기화 됩니다.

### 예 2: COM1 통신 형태의 변형

사용자가 COM1의 통신 형태를 변경하고자 한다면 프로그램의 앞부분에 다음의 프로그램을 추가해 주십시오. PLC가 STOP에서 RUN으로 바뀔 때 M1138이 ON이라면 이를 감지할 것입니다. M1138이 ON이라면, COM1 설정은 D1136에 의해 변경됩니다.

COM1 통신 형태를 ASCII, 9600bps, 7 Data bits, 짝수 패리티, 1 정지 비트 (9600, 7, E1)로 변경하세요.



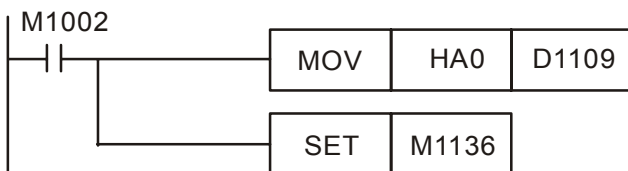
주의:

1. 통신 형태 변경을 완료하고 나면 PLC가 RUN에서 STOP으로 변해도 통신 형태는 변하지 않습니다.
2. 통신 형태 변경을 완료하고 나면 PLC의 통신 형태는 전원이 OFF에서 ON으로 바뀌면 초기화됩니다.

### 예 3: COM13 통신 형태의 변형

COM3 통신 형태는 7 Data bits, 짝수 패리티, 1 정지 비트로 고정되어 있습니다. 사용자들이 COM3의 통신을 속도 38400bps로 변경하고자 한다면 다음의 프로그램을 프로그램의 앞부분에 추가하세요. PLC가 STOP에서 RUN으로 바뀔 때 M1136이 ON이라면 이를 감지할 것입니다. M1138이 ON인 경우 COM3 설정은 D1109에 의해 변경됩니다.

COM3 통신 속도를 38400 bps로 변경하기



주의:

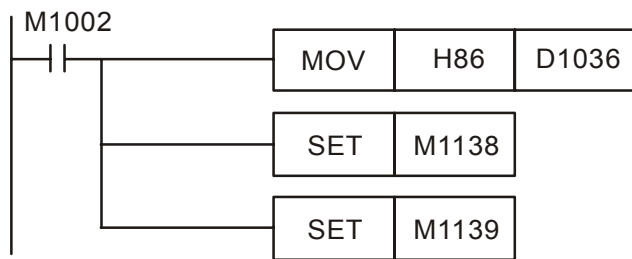
1. 통신 형태 변경을 완료하고 나면 PLC가 RUN에서 STOP이 바뀌었을 때 통신 형태가 변하지 않습니다.
2. 통신 형태 변경을 완료하고 나면 PLC의 통신 형태는 전원이 OFF되고 다시 ON될 때 초기화됩니다.

### 예 4: COM1과 COM2의 RTU 모드 설정

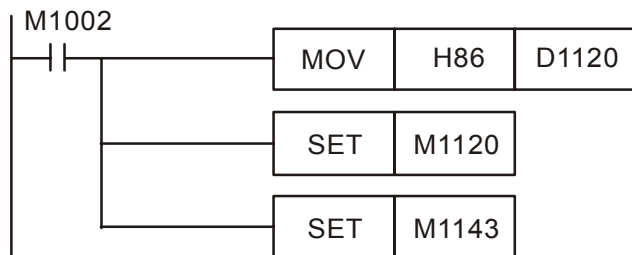
COM1과 COM2는 모두 ASCII와 RTU 모드를 지원합니다. COM1의 플래그는 M1139에 의해 설정되고, COM2는 M1143에 의해 설정됩니다. 플래그가 ON일 때, RTU 모드를 뜻합니다. 플래그가 OFF이면 ASCII 모드입니다.

COM1과 COM2의 RTU 모드 설정은 다음과 같습니다:

COM1 :

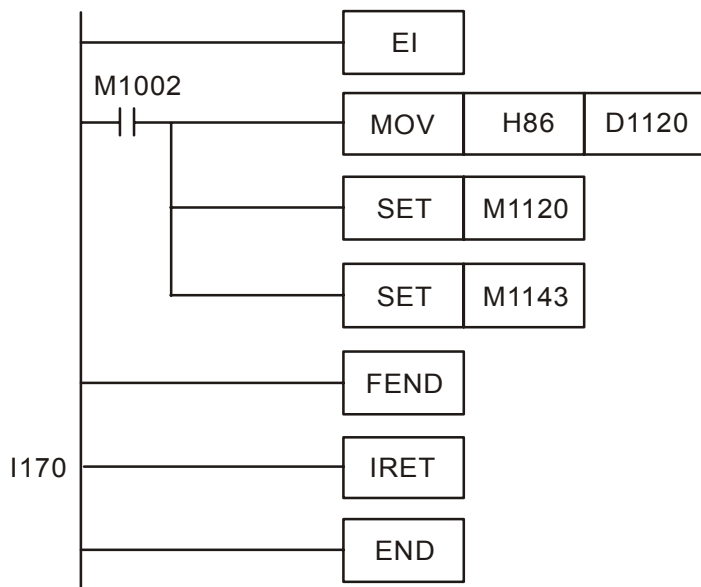


COM2 :



◎ EH 시리즈 MPU 의 COM2 이 Slave 로 쓰일 때 데이터 수신이 끝나면 인터럽트 I170 이 나타납니다. PLC 이 Slave 로 쓰일 때 END 명령 수행을 끝낼 때까지는 통신 데이터를 즉시 처리하지 않습니다. 이는 데이터를 읽는데 많은 시간이 걸리게 되면 통신 데이터가 다소 지연될 수도 있음을 뜻합니다. 이 때, 통신 인터럽트 I170 의 기능이 지연 문제를 해결할 수 있습니다

예: 통신 인터럽트 I170 (Slave 모드에서 데이터 수신 종료 후 인터럽트 수행)



I170 가 프로그램에 추가되면 PLC 는 COM2 가 Slave 모드로 이용될 때 즉시 통신 데이터를 처리하게 됩니다.

주의:

1. I170 을 이용하여 온라인으로 프로그램을 업그레이드하지 마십시오.
2. PLC 의 스캔 시간은 늘어날 수도 있습니다.

기능 그룹      통신 반응 지연  
번호              D1038

내용:

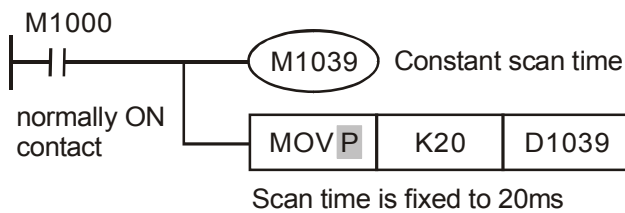
PLC 가 RS-485 통신을 이용하여 Slave 모드로 사용될 때, 0~10,000 (0~1 초)의 범위에서 통신 데이터 반응 지연을 설정할 수 있습니다. 반응 지연이 위 범위를 넘어서면 시간 단위 0.1ms 로 D1038 을 0 으로 설정하십시오. 이 설정은 D1000(WDT)의 설정보다 작은 값이어야 합니다.

기능 그룹 상수(constant) 스캔 시간

번호 M1039, D1039

내용:

1. M1039 가 On 일 때, 프로그램 스캔 시간은 D1039 에 의해 결정됩니다. 프로그램이 작업을 완료하면 상수 스캔 시간이 달성됨에 따라 다음 스캔작업을 시작하게 됩니다. D1039 가 프로그램 스캔 시간보다 작다면 프로그램 스캔 타임이 적용될 수도 있습니다.



2. 스캔 시간의 상대적 명령은 RAMP(API 67), HKY(API 71), SEGL(API 74), ARWS(API 75), PR(API 77)입니다. 이는 “상수 스캔 시간”이나 “상수 시간 삽입 인터럽트”와 함께 사용되어야 합니다.
3. HKY(API 71) 명령에 있어서, 4x4 행렬에 의한 16 개 키를 활용할 때 스캔 시간은 20ms 이상으로 설정되어야 합니다.
4. 스캔 시간 D1010~D1012 디스플레이는 상수 스캔 시간도 포함합니다.

기능 그룹 아날로그 기능

번호 D1056~D1059, D1110~D1113, D1116~D1118

내용:

1. 아날로그 입력 채널 해상도(정확도): EX MPU (0~±10V or 0~±20mA)의 경우 10 비트; SX MPU (0~±10V or 0~±20mA)의 경우 12 비트
2. 아날로그 출력 채널 해상도: EX MPU (0~10V or 0~20mA)의 경우 8 비트; SX MPU (0~±10V or 0~±20mA)의 경우 12 비트
3. EX 시리즈의 아날로그 디지털 컨버터 필터 시간 설정입니다. 기본 설정은 0 이고 단위는 1ms 입니다. D1118 ≤5 이면, 5ms 로 처리됩니다.
4. EH 아날로그 입력 AD 카드의 해상도: 12 비트(0~10V 나 0~20mA)
5. EH 아날로그 입력 DA 카드의 해상도: 12 비트(0~10V 또는 0~20mA)

장치	기능
D1056	EX/SX MPU 아날로그 입력 채널 0 (CH0)의 현재 값과 EH MPU AD 카드 채널 0 (CH0)의 현재 값
D1057	EX/SX MPU 아날로그 입력 채널 1(CH 1)과 EH MPU AD 카드 채널 1 (CH1)의 현재 값



장치	기능
D1058	EX MPU 아날로그 입력 채널 2 의 현재 값 (CH 2)
D1059	EX MPU 아날로그 입력 채널 3 (CH 3)의 현재 값
D1110	EX/SX MPU 아날로그 입력 채널 0 (CH 0)과 EH MPU AD 카드 채널 0 (CH0)의 평균값
D1111	EX/SX MPU 아날로그 입력 채널 1 (CH 1)과 EH MPU AD 카드 채널 1 (CH1)의 평균값
D1112	EX MPU 아날로그 입력 채널 2 (CH 2)의 평균값
D1113	EX MPU 아날로그 입력 채널 3 (CH 3)의 평균값
D1116	EX MPU 아날로그 출력 채널 0(CH 0), EH MPU DA 카드 채널 0(CH0)
D1117	EX MPU 아날로그 출력 채널 1(CH 1), EH MPU DA 카드 채널 1(CH1)
D1118	EX/SX/EH 시리즈 아날로그 입력 필터 설정(ms)

기능 그룹      알고리즘 오류 플래그  
 번호            M1067~M1068, D1067~D1068

내용:

1. 알고리즘 오류 플래그:

구성	설명	래치	STOP→RUN	RUN→STOP
M1067	알고리즘 오류 플래그	없음	소거	래치
M1068	알고리즘 오류 잠금 플래그	없음	불변	래치
D1067	알고리즘 오류 코드	없음	소거	래치
D1068	알고리즘 오류의 STEP 값	없음	불변	래치

2. 오류 코드 설명:

D1067 에러 코드	기능
H 0E18	BCD 변환 오류
H 0E19	분모가 0
H 0E1A	사용 한도 초과(E 와 F 포함)
H 0E1B	무리식이 음수
H 0E1C	FROM/TO 통신 오류

기능 그룹      저전압  
 번호            M1087, D1100

내용:

1. PLC 가 LV(저전압) 신호를 감지하면 M1087 이 ON 인지 아닌지 체크합니다. M1087 이 ON 이면 D1100 의 내용은 Y0~Y17 에 저장됩니다.
2. D1100 의 비트 0 (LSB)은 Y0 에 해당하고, 비트 1 은 Y1 에, 비트 8 은 Y8 와 같이 해당됩니다.

기능 그룹      파일 레지스터

번호 M1101, D1101~D1103

내용:

1. SA, SX, SC, EH 시리즈에서는 PLC 전원이 켜져 있다면 M1101로부터 시작 파일 레지스터와 D1101(SA, SX, SC 시리즈의 파일 레지스터: K0~K1, 599; EH 시리즈는 K0~K9, 999)에 있는 파일 레지스터의 시작 번호를 체크하고, D1102(SA, SX, SC 시리즈의 파일 레지스터에서 아이템을 읽어옴: K0~K1, 600; EH 시리즈에서는 K0~K8, 000)와 D1103(저장과 읽기를 위한 파일 레지스터, 지정된 데이터 레지스터 D (SA, SX, SC 시리즈: K2000~K4999, EH 시리즈: K2000~K9999)의 시작 번호에서 파일 레지스터를 지정된 데이터 레지스터로 자동으로 보낼지를 결정)에서 파일 레지스터의 아이템 번호를 읽습니다.
2. API 148 MEMR 명령과 API 149 MEMW 설명을 참조하십시오.

기능 그룹 DIP 스위치 기능 카드

번호 M1104~M1111

내용:

1. PLC 가 DIP 스위치 카드와 함께 RUN 이라면 8 개의 DIP 스위치는 M1104~M1111 에 각각 해당됩니다.
2. 자세한 내용은 API 109 SWRD 명령을 참조하십시오.
3. PLC 가 입력 AX0 에 4DI 카드에 꽂힌 채 RUN 상태라면 (포토커플러 절연), 입력 AX0~AX3 의 상태는 각각 M1104~M1107 에 해당됩니다.

기능 그룹 트랜지스터 출력 카드

번호 M1112, M1113

내용:

PLC 가 트랜지스터 출력 카드와 함께 RUN 상태라면 M1112 와 M1113 은 2 개의 트랜지스터 출력 AY0, AY1과 같습니다.

기능 그룹 가속/감속을 동반한 펄스 출력

번호 M1115~M1119, D1104

내용:

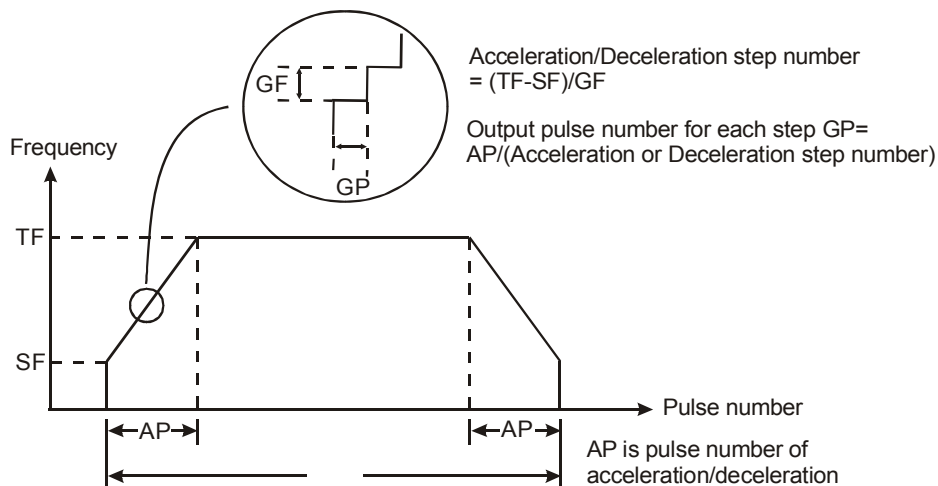
1. 가속/감속을 동반한 펄스 출력에 이용되는 특수 D 와 특수 M 의 정의::

장치	기능
M1115	가속/감속 펄스 출력 스위치 시작
M1116	가속에 쓰이는 플래그
M1117	목표 주파수 달성 플래그
M1118	감속에 쓰이는 플래그
M1119	기능 플래그 완료
D1104	변수 인덱스 활용 (D 와 동일)

2. 변수 표(주파수 범위는 25Hz~10KHz)

D 시작 번호 + 인덱스	기능	
+0	시작 주파수 (SF)	
+1	주파수 격차 (GF)	
+2	목표 주파수 (TF)	
+3	펄스 출력 숫자의 총 개수 (32 비트의 하위 16 비트)	(TP)
+4	펄스 출력 숫자의 총 개수(32 비트의 상위 16 비트)	
+5	가속 구간에서의 출력 펄스 숫자 (32 비트의 하위 16 비트)	(AP)
+6	감속 구간에서의 출력 펄스 숫자 (32 비트의 상위 16 비트)	

3. 명령은 쓸 필요가 없고, 변수 차트를 채워넣고 M1115 를 시작 설정하면 됩니다. 이 기능은 Y0 출력만을 사용할 수 있으며 시간은 다음과 같습니다.



#### 4. 주의 :

이 기능은 다음의 조건이 모두 만족될 시에만 실행되어야 합니다. 조건이 하나라도 빠질 때에는 기능이 수행될 수 없습니다.

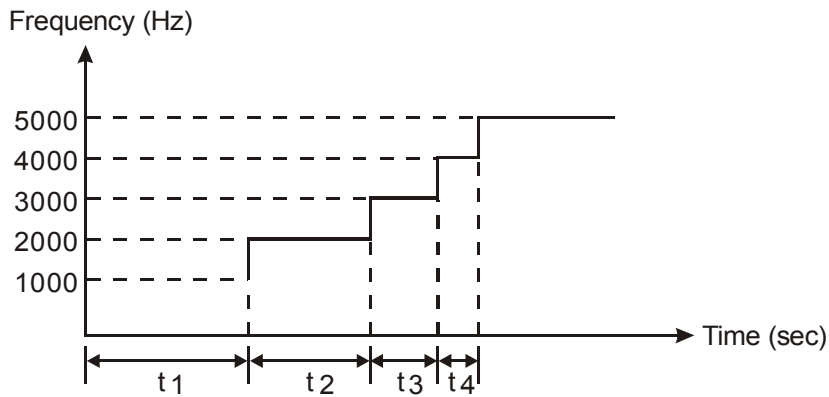
- 시작 주파수는 목표 주파수보다 작아야 합니다.
- 주파수 차이는 (목표 주파수 - 시작 주파수) 보다 작아야 합니다.
- 펄스 숫자의 총 개수는 (가속/감속 펄스 숫자 \* 2)보다 커야 합니다.
- 시작 주파수와 목표 주파수: 최소 25Hz 이고 최대 10KHz 입니다.
- 가속/감속 펄스 숫자는 가속/감속 스텝 숫자보다 커야 합니다.

M1115 이 On 에서 Off 로 바뀌면 M1119 는 소거되고 M1116, M1117, M1118 는 변하지 않습니다. PLC 가 STOP 에서 RUN 으로 바뀌거나 RUN 에서 STOP 으로 바뀌면 M1115~M1119 는 Off 로 소거됩니다. D1104 는 Off 에서 On 으로 바뀔 때에만 0 으로 소거됩니다.

“가속/감속 펄스 출력” 기능과 PLSY Y0 출력 명령이 동시에 발생하면 Y0 출력부터 하나씩 수행하게 됩니다.

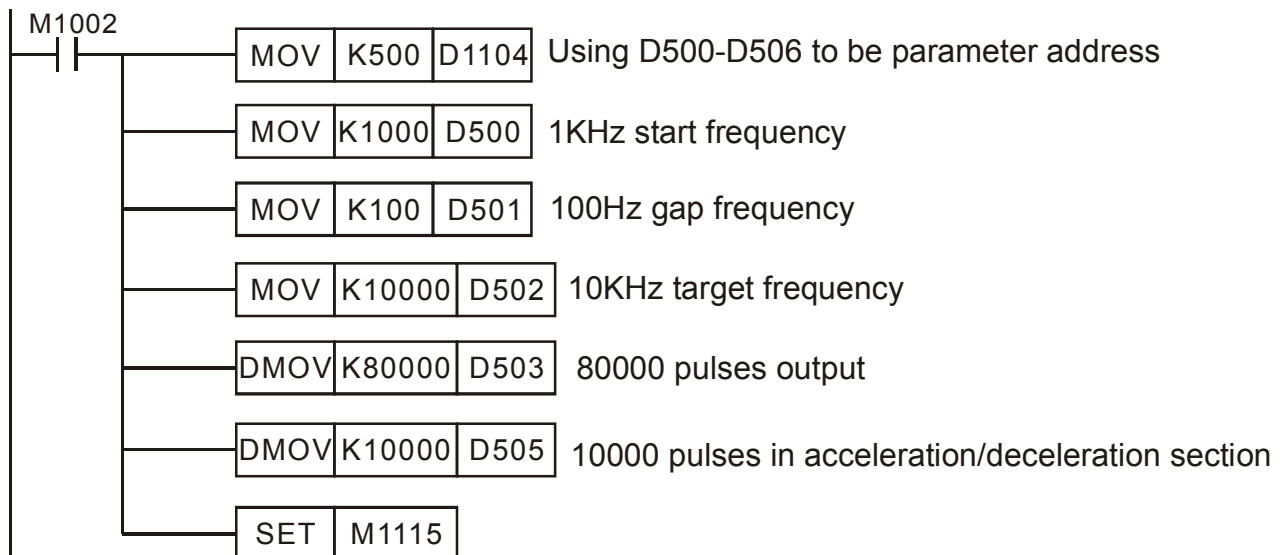
#### 5. 각 부분의 실행 시간 계산하기

- 시작 주파수가 1KHz 로, 주파수 차이가 1KHz, 목표 주파수가 5KHz, 총 펄스 숫자가 100, 가속/감속 펄스 숫자가 40 이라면 가속/감속 구간의 시간 차트는 다음과 같습니다.

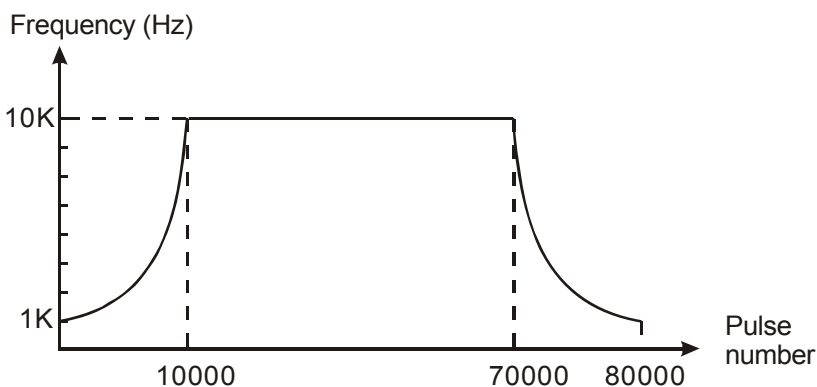


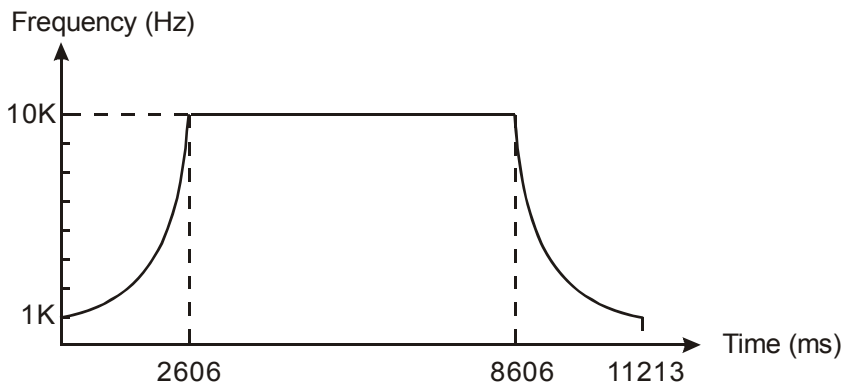
- b) 가속/감속 스텝은  $(5K - 1K) / 1K = 4$  이고 각 펄스의 출력 숫자는  $40 / 4 = 10$  이 되도록 할 수 있습니다. 따라서 이 때 다음의 그림으로부터  $t1 = (1 / 1K) * 10 = 10ms$ ,  $t2 = (1 / 2K) * 10 = 5ms$ ,  $t3 = (1 / 3K) * 10 = 3.33ms$  and  $t4 = (1 / 4K) * 10 = 2.5ms$  을 얻어낼 수 있습니다.

예: **Forward/Reverse** 가속/감속 스텝 모터 컨트롤



- PLC 가 RUN 이라면, D1104 에 의해 지정된 레지스터에 각 변수를 저장합니다.
- M1115 이 On 이라면, 가속/감속 펄스는 출력을 시작합니다.
- M1116 이 가속 중 On 이라면, 속력에 도달했을 때 M1117 은 On, 감속 시 M1118 은 On, 작업 완료 후 M1119 은 On 입니다.
- M1115 은 자동으로 초기화되지 않으며 사용자가 해야 합니다.
- 실제 펄스 출력 커브는 다음과 같습니다.





기능 그룹 특수 고속 펄스 출력  
번호 M1133~M1135, D1133

내용:

1. SA, SX, SC 시리즈에서, 특수 고속 펄스 Y0(50KHz) 출력 함수의 특수 D 와 특수 M 의 정의:

장치	기능
M1133	특수 고속 펄스 Y0 (50KHz) 출력 스위치 (On 은 작업 시작을 뜻합니다.)
M1134	특수 고속 펄스 Y0 (50KHz) 출력. (On 은 출력 계속 스위치입니다.)
M1135	특수 고속 펄스 Y0 (50KHz) 출력. 출력 펄스 수 달성 플래그
D1133	특수 고속 펄스 Y0 (50KHz) 출력. 특수 고속 펄스 출력 레지스터(D)의 인덱스

2. D1133 변수의 변수표

D 시작 번호 + 인덱스	기능
+0	Y0 의 특수 고속 출력 주파수(32 비트 중 하위 16 비트)
+1	Y0 의 특수 고속 출력 주파수 (32 비트 중 상위 16 비트)
+2	Y0 의 특수 고속 출력 펄스 수 (32 비트 중 하위 16 비트)
+3	Y0 의 특수 고속 출력 펄스 수 (32 비트 중 상위 16 비트)
+4	현재 특수 고속 Y0 출력 펄스 수(32 비트 중 하위 16 비트 )
+5	현재 특수 고속 Y0 출력 펄스 수(32 비트 중 상위 16 비트 )

3. 기능 설명: 위의 출력 주파수와 출력 숫자는 M1133 이 On 이고 M1135 가 Off 일 때 조정할 수 있습니다. 출력 주파수나 출력 목표수가 바뀌는 것은 현재 출력 펄스에 영향을 미치지 않습니다. 현재 출력 펄스 수는 스캔 시간이 업데이트 되면 나타나게 됩니다. M1133 이 Off 에서 On 으로 바뀌면 0 으로 소거되고 M1133 이 On 에서 Off 로 바뀌면 마지막 출력 숫자를 저장하게 됩니다.

주의:

1. 위의 특수 고속 펄스 출력 기능은 RUN 상태의 특수 Y0 출력 지점을 사용할 수 있습니다. 또한 PLSY Y0 과도 동시에 존재할 수 있는데 이는 PLSY (Y0)에 영향을 미치지 않습니다. PLSY(Y0) 명령이 이 기능 이전에 실행되었다면 이 함수는 사용될 수 없고 그 반대도 마찬가지 입니다. 이 함수를 실행할 때 일반 Y0 출력은 무효가 되며 Y1~Y7 의 출력 지점이 이용됩니다.
2. 이 기능과 PLSY 명령의 차이는 출력 주파수보다 큼니다. 최대 출력은 50KHz 까지 입니다.

기능 그룹 확장 연결 감지  
 번호 D1140, D1142, D1143

내용:

D1140: 특수 확장 모듈(AD, DA, XA, PT, TC, RT, HC, PU) 숫자, 최대는 8.

D1142: 디지털 확장 입력 X 지점 번호.

D1143: 디지털 확장 입력 Y 지점 번호.

기능 그룹 조정가능한 가속/감속 펄스 출력 기능 설명  
 번호 M1144~M1149, M1154, D1032, D1033, D1144, D1154, D1155

내용:

1. SA, SX, SC 시리즈에서 조정가능한 가속/감속 펄스 출력 기능에서의 특수 D 와 특수 M 의 정의:

Device	Function
M1144	가속/감속 펄스 출력 Y0 시작 스위치의 조정 가능 슬로프에 대한 펄스 출력 함수
M1145	Y0 가속 플래그의 조정 가능 슬로프에 대한 펄스 출력 함수
M1146	Y0 목표 주파수 달성 플래그의 조정 가능 슬로프에 대한 펄스 출력 함수
M1147	Y0 감속 플래그의 조정 가능 슬로프에 대한 펄스 출력 함수
M1148	Y0 완료 기능 플래그의 조정 가능 슬로프에 대한 펄스 출력 함수
M1149	Y0 일시적으로 카운팅 멈춤 플래그의 조정가능 슬로프에 대한 펄스 출력 함수
M1154	지정된 감속 시간 차이 플래그와 주파수 플래그의 조정가능 슬로프에 대한 펄스 출력 함수
D1032	Y0 펄스 누적 출력 값의 32 비트 중 하위 16 비트
D1033	Y0 펄스 누적 출력 값의 32 비트 중 상위 16 비트
D1144	Y0 변수 인덱스(D 에 해당) 사용의 조정가능 슬로프에 대한 펄스 출력 함수
D1154	지정 감속 시간차에 대한 권고값 (10~32767 ms)
D1155	지정 가속 주파수차에 대한 권고값 (-1~ - 32700 Hz)

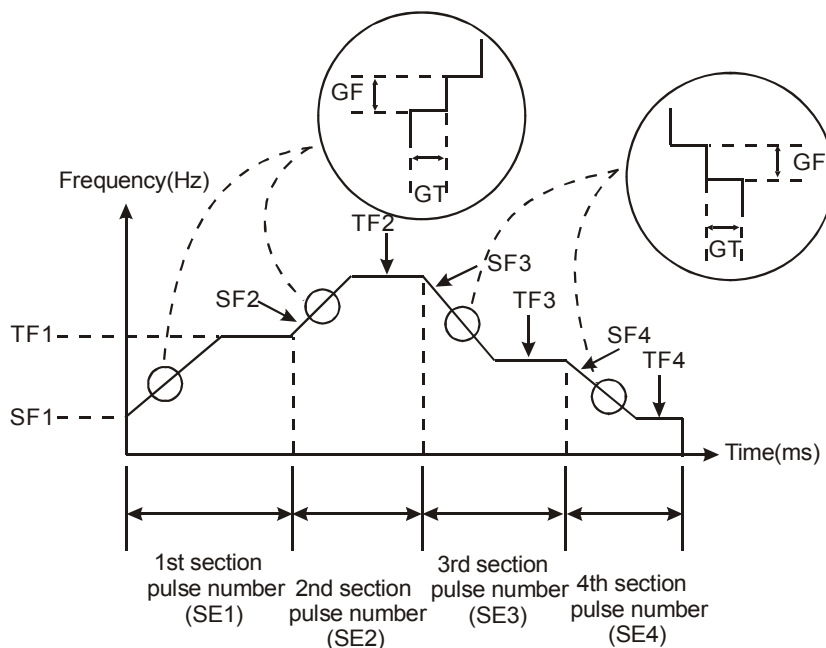
2. 변수 D1144 표

D 시작 번호 + 인덱스	기능
+0	총 세그먼트 수 (n) (최대 숫자는 10)
+1	실행 세그먼트 표시 (읽기 전용)
+2	첫번째 세그먼트의 시작 주파수 (SF1)
+3	첫번째 세그먼트의 시간 간격 (GT1)
+4	첫번째 세그먼트의 주파수 간격 (GF1)
+5	첫번째 세그먼트의 목표 주파수 (TF1)
+6	첫번째 세그먼트의 출력 펄스에 대한 목표값에서 32 비트 중 하위 16 비트
+7	첫번째 세그먼트의 출력 펄스에 대한 목표값에서 32 비트 중 상위 16 비트

D 시작 번호 + 인덱스	기능
+8	두번째 세그먼트의 시작 주파수(SF2)로서, 첫번째 세그먼트의 주파수(TF1)와는 다른 목표 주파수
+9	두번째 세그먼트의 시간 간격(GT2)
+10	두번째 세그먼트의 주파수 간격(GF2)
+11	두번째 세그먼트의 목표 주파수 (TF2)
+12	두번째 세그먼트의 출력 펄스에 대한 목표값에서 32 비트 중 하위 16 비트
+13	두번째 세그먼트의 출력 펄스에 대한 목표값에서 32 비트 중 상위 16 비트
:	:
+n*6+2	n 번째 세그먼트의 시작 주파수 (SFn)로서 n-1 번째 세그먼트의 목표 주파수(TFn-1)와는 다른 주파수
+n*6+3	n 번째 세그먼트의 시간 간격(GTn)
+n*6+4	n 번째 세그먼트의 주파수 간격 (GFn)
+n*6+5	n 번째 세그먼트의 목표 주파수(TFn)
+n*6+6	n 번째 세그먼트의 출력 펄스에 대한 목표값에서 32 비트 중 하위 16 비트
+n*6+7	n 번째 세그먼트의 출력 펄스에 대한 목표값에서 32 비트 중 상위 16 비트

### 3. 기능 설명:

이 기능은 Y0 출력 지점에서만 사용될 수 있고 시간배분은 다음과 같습니다. 변수 표를 다 채우고 난 후에는 M1144 를 설정하여 시작하십시오. (RUN 모드에서 작동되어야 합니다.)



### 4. 사용 규칙과 제한:

- 시작 주파수와 목표 주파수의 최소값은 200Hz 보다 같거나 큰 값이어야 합니다. 이보다 작을 경우에는 작업을 마치거나 아예 작업을 하지 않을 수도 있습니다.

- b) 시작 주파수와 목표 주파수의 최대값은 32700Hz 입니다. 이보다 더 큰 값일 경우에는 32700Hz 의 경우에서와 똑같이 작업하게 됩니다.
- c) 시간 간격에 대한 범위는 1~32767ms 이고 단위는 ms 입니다.
- d) 가속 세그먼트에서 주파수 간격 범위는 1Hz~32700Hz 이고 감속 세그먼트에서는 -1~32700Hz 입니다. 만약 이 숫자가 0 으로 맞춰지면 작업을 시작한 세그먼트는 목표 주파수에 도달할 수 없는데, 목표 수치에 접근하고 난 뒤 다음 세그먼트를 실행합니다.
- e) 세그먼트 펄스 출력값의 목표치는  $((GF*GT/1000)*(TF-SF)/GF)$ 보다 커야 합니다. 자세한 사항은 예 1 을 참조하십시오. 세그먼트 펄스 출력값의 목표치가  $((GF*GT/1000)*(TF-SF)/GF)$ 보다 크지 않으면 이 기능은 사용될 수 없습니다. 항상 메소드가 첨가되면 시간 간격이나 펄스 출력 목표치를 추가하게 됩니다.
- f) RUN 모드의 고속 명령에 의해 Y0 지정 출력이 있다면 Y0 출력 명령은 가장 높은 우선순위를 가지고 시작됩니다.
- g) M1144 실행 시작 후 M1148 가 기능완료 플래그 없이 출력을 하고 M1144 는 끝나버렸다면 이 기능은 감속을 시작할 것입니다. 지정된 가속 기능 플래그 M1154 가 Off 라면 이 기능은 출력 주파수가 200Hz 보다 작아질 때까지 200ms 당 200Hz 로 줄이고 M1147 을 감속 플래그로 맞출것입니다. 그러나 만약 지정 감속 플래그 M1154 가 On 이라면 이 기능은 사용자가 정의한 시간 간격과 주파수에 따라 작업합니다. 그리고 시간 간격은 0 보다 작거나 같을 수 없습니다. (만약 0 보다 작거나 같으면 기계 설정은 200ms 로 바뀝니다.) 주파수 간격은 0 보다 크거나 같을 수 없습니다. (0 과 같을 경우 기계 설정은 -1Hz 로 설정되고 0 보다 클 경우에는 음의 숫자로 변환됩니다.)
- h) M1148 이 기능 완료 플래그를 달성하고 M1144 가 닫혔다면 이 기능은 감속 기능을 시작하지 않고 M1148 플래그를 소거합니다. M1144 가 닫히면 M1149 플래그를 소거합니다.
- i) 이 기능의 작업 세그먼트는 총 세그먼트 숫자에 의해 작동됩니다. 최대 10 세그먼트까지 가능합니다.
- j) 이 기능의 가속/감속은 다음 세그먼트의 시작 주파수에 의해 작동됩니다. 다시 말하면, 작업 세그먼트의 목표 주파수가 다음 세그먼트의 시작 주파수보다 작을 때, 다음 세그먼트는 가속되고 다음 세그먼트의 목표 주파수는 시작 주파수보다 커야 합니다. 작업 세그먼트의 목표 주파수가 다음 세그먼트의 주파수보다 클 경우 다음 세그먼트는 감속되고 따라서 다음 세그먼트의 목표 주파수는 시작 주파수보다 작아야 하는 것입니다. 사용자가 이러한 방식으로 설정해두지 않으면 올바른 출력 펄스를 장담할 수 없습니다.
- k) STOP→RUN 일 때, M1144~M1149 는 Off 로 소거되어야 합니다. RUN→STOP 일 때, M1144 는 소거되지만 M1145~M1149 는 소거되지 않을 것입니다. D1144 는 Off 에서 On 일 때 소거되고 위의 경우에는서는 변하지 않습니다.
- l) SA, SX, SC 시리즈의 사용 변수 범위는 D0~D999 와 D2000~D4999 입니다. 변수가 범위를 벗어난다면 이 명령을 수행하지 않고 M1144 를 종료할 것입니다. (모든 사용 세그먼트 변수 포함)

### 예 1: 각 세그먼트의 가속/감속 출력과 목표 주파수를 계산하십시오.

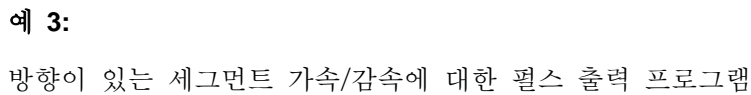
세그먼트의 시작 주파수를 200Hz, 세그먼트 시간 간격은 100ms, 세그먼트 주파수 차이는 100Hz, 세그먼트 목표 주파수는 500Hz, 세그먼트 펄스의 목표치를 1000 으로 설정합니다. 계산은 다음과 같습니다:

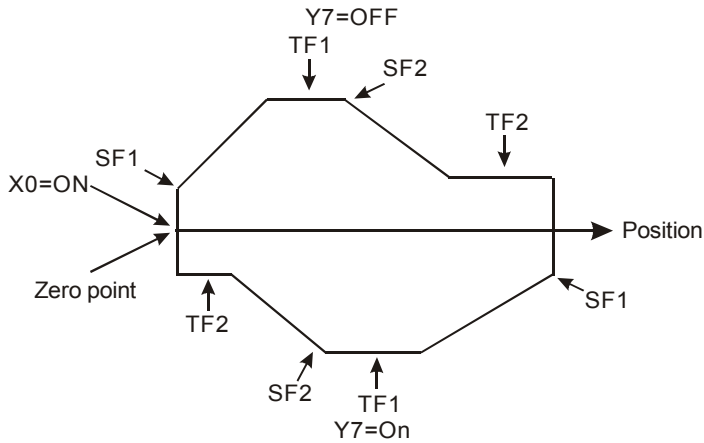
1. 가속/감속 시작 시 출력 펄스 수는  $200*100/1000 = 20 \text{ pulses}$
2. 첫번째 가속 기간의 출력 펄스 수  $300*100/1000 = 30 \text{ pulses}$
3. 두번째 가속 기간의 출력 펄스 수  $400*100/1000 = 40 \text{ pulses}$
4. 목표 주파수의 출력 펄스 수  $1000 - (40+30+20) = 910 \text{ pulses}$



5. 목표 주파수의 출력 시간  $1 / 500 * 910 = 1820 \text{ ms}$
6. 이 세그먼트의 총 시간  $1820 + 3 * 100 = 2120 \text{ ms}$

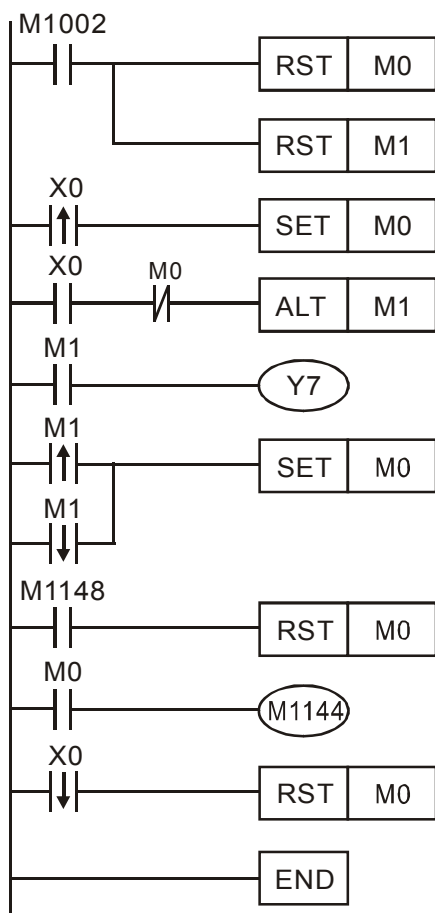
한 세그먼트의 가속과 세그먼트 감속에 대한 간단한 가속/감속 펄스 출력 프로그램





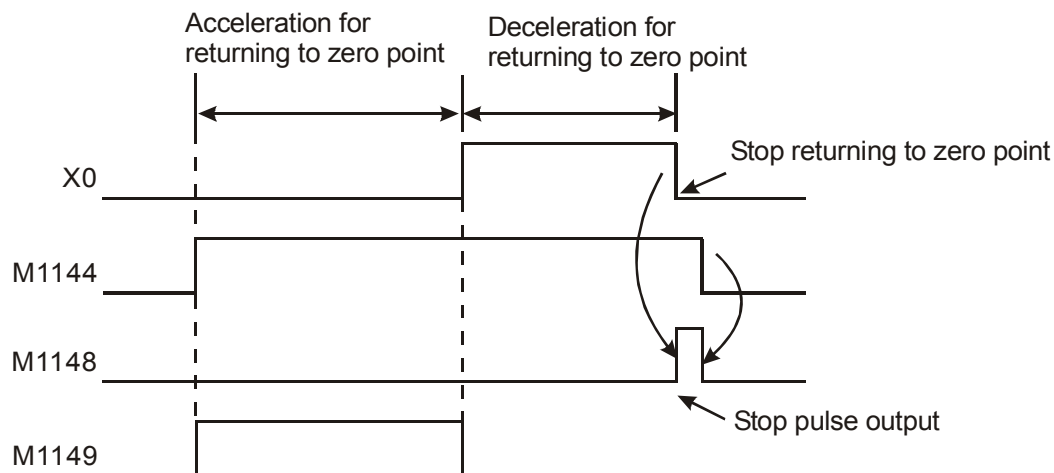
설명:

1. 가속/감속 설정은 예 2 와 같습니다.
2. 위의 그림은 위치별 움직임의 예입니다. X0 접점이 On 일 때 움직이기 시작하고 X0 이 Off 일 때 움직임을 멈출 것입니다. (Y7 은 방향 설정을 합니다.)
3. 프로그램은 아래와 같습니다.

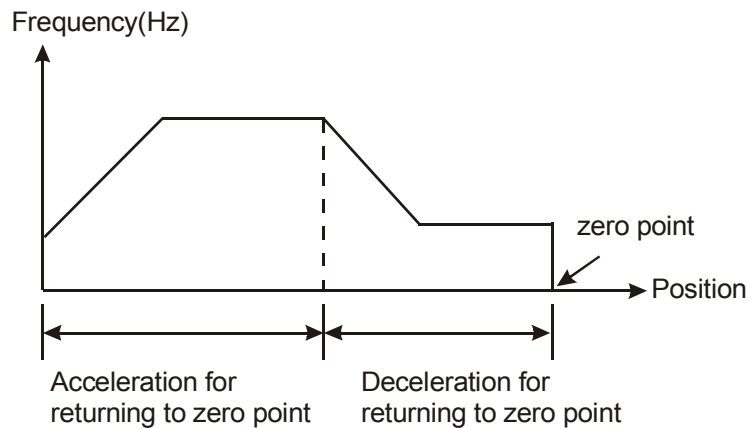


예 4:

한 세그먼트의 가속과 감속을 제로점 복귀 프로그램에 적용하기  
상대 플래그 시간배분 차트는 아래와 같습니다.



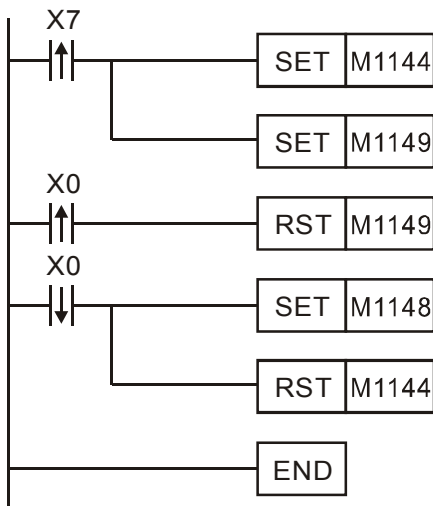
주파수와 위치의 관계는 아래와 같습니다.



가속/감속, 주파수, 펄스에 대한 숫자 설정은 다음과 같습니다. (D 와 동일)

D 의 시작점+ 인덱스	설정
+0	2
+2	250(Hz)
+3	100(ms)
+4	500(Hz)
+5	10000(Hz)
+6, +7	10(pulse)
+8	9750(Hz)
+9	50(ms)
+10	-500(Hz)
+11	250(Hz)
+12, +13	30000(pulse)

프로그램은 아래와 같습니다: (점점 X7 이 리셋 트리거 스위치를 시작한다고 가정합니다.)



설명:

1. 점점 X7 가 트리거된 뒤, M1144 는 가속을 시작하도록 설정되고 M1149 는 펄스 숫자를 세지 않도록 설정됩니다. 그리고 감속 스위치 X0 이 트리거되면 10 펄스를 보내고, 그런 뒤 감속 세그먼트로 들어갑니다.
2. M1148 을 설정하여 펄스 출력을 수동으로 끝내고 X0 가 닫힌 뒤 이 기능을 닫기위한 것입니다.

주의: 이 예는 단지 활용 방법으로서 사용자가 가속/감속 세그먼트에서 이용되는 변수 설정들을 실제 기계 특성과 한계에 맞추어 조정해야 합니다.

기능 그룹      단일 단계(single step) 실행 기능

번호              M1170, M1171, D1170

내용:

EH 시리즈 단일 단계 실행 기능의 특수 D 와 특수 M 의 정의

장치	기능 설명
M1170	단일 단계 기능의 시작 플래그
M1171	단일 단계 실행 플래그
D1170	현 PLC 실행 명령의 STEP 수

기능 설명:

1. 실행 시간: 이 플래그는 PLC 가 RUN 모드일 때 유효합니다.
2. 실행 단계:
  - a) M1170 을 시작하여 단일 단계 실행 모드로 들어가게 합니다. PLC 는 D1170 에 저장된 STEP 의 특정 명령에 머물러 있다가 이 명령을 한번만 실행하게 됩니다.
  - b) M1171 를 강제로 On 으로 만들 때, PLC 는 다음 명령을 실행하고 거기에서 멈춥니다. 동시에 PLC 가 M1171 을 Off 되도록 강제합니다. D1170 은 현재 STEP 값을 보여줍니다.
  - c) Y 출력이 단일 STEP 실행에 있다면 Y 출력 명령은 END 명령을 수행하지 않고 즉시 출력을 하게 됩니다.

주의:

- 스캔 시간에 의해 영향을 받는 명령들은 단일 STEP 실행 때문에 틀린 것이 될 수도 있습니다. 예: HKY 명령이 실행될 때에는 한 버튼의 유효 입력을 받기 위해서 8 스캔 시간이 필요합니다. 따라서 단일 단계 실행은 오류를 낳게 됩니다.
- 펄스 입력/출력, 고속 비교 명령과 같은 명령들은 하드웨어 시작에 관계하기 때문에 단일 단계에 영향을 받지 않습니다.

기능 그룹      2 단계 출력 기능

번호              M1172~M1174, D1172~D1177

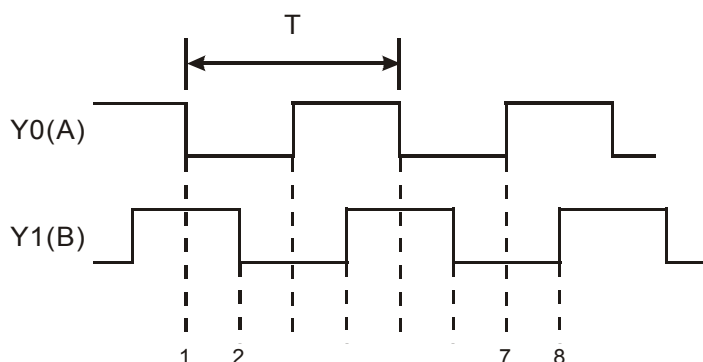
내용:

SA, SX, SC 시리즈에서 2 단계 출력 기능의 특수 D 와 특수 M 의 정의:

장치	기능 설명
M1172	2 단계 펄스 출력 스위치
M1173	On 은 계속 출력 스위치입니다.
M1174	출력 펄스 수 달성 플래그
D1172	2 단계 출력 주파수 (12Hz~20KHz)
D1173	2 단계 출력 모드 선택 (k1 와 k2)
D1174	2 단계 출력 펄스 목표 수의 32 비트 중 하위 12 비트
D1175	2 단계 출력 펄스 목표 수의 32 비트 중 상위 12 비트
D1176	2 단계 현재 출력 펄스 수의 32 비트 중 하위 12 비트
D1177	2 단계 현재 출력 펄스 수의 32 비트 중 상위 12 비트

기능 설명:

- 아래의 그림에서 볼 수 있듯이 출력 주파수는  $1/T$  입니다. 출력 모드는 k1, k2 두가지이며, k1 은 A 단계가 B 단계를 우선함을 의미하고 k2는 B 단계가 A 단계를 우선함을 의미합니다. 출력 숫자 계산은 아래와 같이 단계에 차이가 있을 때 1 을 더합니다. 아래에는 8 개의 출력 펄스가 있습니다. 출력 숫자를 달성했을 때 M1174 는 On 이 되는데, 만약 M1174 를 소거하고 싶다면 M1172 를 닫아야 합니다.



- 출력 주파수, 출력 목표 수, 모드 선택은 M1172 가 On 이고 M1174 가 Off 일 때 변경이 가능합니다. 출력 주파수와 출력 목표 수의 조정은 현재 출력 펄스수에 영향을 주지는 않지만 모드 선택 설정은 현재 출력 펄스 수를 0 으로 소거시킵니다. 현재 출력 펄스 수는 스캔 시간이 업데이트되는 대로 업데이트되고, M1172 가 Stop 에서 Run 으로 바뀔 때 0 으로 소거되며, M1172 가 Run 에서 Stop 으로 바뀔 때에는 마지막 출력 숫자를 저장합니다.

### 주의:

이 기능은 RUN 모드에서만 사용가능하고 PLSY 명령과 함께 존재할 수 있습니다. 그러나 PLSY 명령이 먼저 실행될 경우 이 기능은 사용되지 못하며, 그 역도 마찬가지 입니다.

기능 그룹      VR 전위차계  
번호              M1178~M1179, D1178~D1179

### 내용:

EH, SA, SC 시리즈에서 내장된 2 지점 VR 전위차계 기능의 특수 D 와 특수 M 의 정의:

장치	기능
M1178	VR0 전위차계 시작
M1179	VR1 전위차계 시작
D1178	VR0 값
D1179	VR1 값

### 기능 설명:

이 기능은 RUN 모드에서 사용가능합니다. M1178 이 On 일 때 VR0 의 변동값은 아라비아 숫자 0~255 로 변환되어 D1178 에 저장됩니다. M1179 가 On 일 때 VR1 의 변동값 또한 변환되어 D1179 에 저장됩니다.

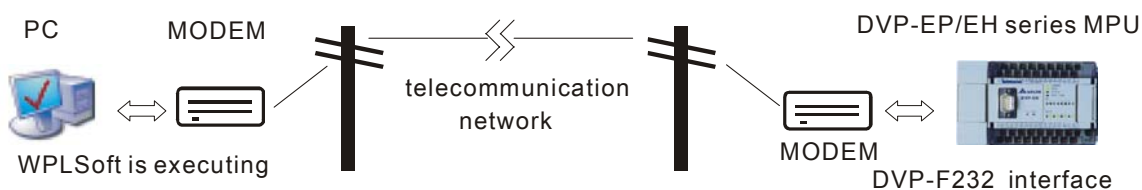
### 주의:

자세한 사항은 API 85 VRRD 명령을 참조하십시오.

기능 그룹      MODEM 연결 기능  
번호              M1184~M1188

### 내용:

#### 1. 시스템 연결



#### 2. EH 시리즈의 MODEM 연결을 위한 특수 M 의 정의:

장치	기능 설명	비고
M1184	시작 MODEM	M1184=On 일 때 아래의 기능들이 유효합니다.
M1185	시작 MODEM 초기화	초기화가 끝난 뒤 이 플래그는 Off 상태가 됩니다.
M1186	MODEM 초기화 실패	M1185=On, M1186=Off 일 경우
M1187	MODEM 초기화 성공	M1185=On, M1187=Off 일 경우
M1188	MODEM 의 연결여부를 표시	On 은 연결을 의미합니다.

주의: 특수 M 은 PLC 의 RUN/STOP 상태에 상관없이 항상 유효합니다.

### 3. 작업: (아래의 순서대로 작업해 주십시오)

- a) PLC 쪽에서 M1184=On 로 설정하십시오. (시작 MODEM)
- b) 2 단계: M1185=On 로 설정하십시오. (PLC 의 MODEM 초기화 시작)
- c) 3 단계: MODEM 초기화의 결과를 체크하십시오. M1186=On 은 초기화 성공을 의미하고 M1187=On 은 초기화 실패를 의미합니다.
- d) 4 단계: 초기화 성공 후, 멀리 떨어진 PC 쪽에서 WPL 소프트웨어에 대해 연결을 준비합니다. WPL 연결 방법: 설정 → 모뎀 연결 (모뎀 드라이버를 먼저 설치해야 합니다) → 다이얼 연결 대화상자가 뜨면 다음과 같이 다이얼 정보를 입력합니다.

### 4. 주의:

- a) PLC 쪽에서 MODEM 을 연결할 때에는 반드시 RS-232 카드와 함께 사용되어야 합니다. 그렇지 않으면 위의 특수 M 은 유효하지 않습니다.
- b) MODEM 시작(M1184=On) 후 M1185=On 로 설정하여 MODEM 을 초기화 합니다. 그렇지 않으면 PLC 쪽에서 MODEM 자동 다이얼 기능을 이용할 수 없습니다.
- c) MODEM 은 초기화 이후 자동 다이얼 모드가 됩니다.
- d) MODEM 의 경우 멀리 떨어진 PC 가 연결을 중지한 뒤 PLC 쪽에서 다이얼 모드로 돌입합니다. 사용자가 이 때 MODEM 의 전원을 끈다면 다음번에 켜를 때 꼭 초기화 해야합니다.
- e) PLC 쪽의 MODEM 연결 변조 속도는 9600bps 로 고정되고 있고 이를 변경할 수 없습니다. 또한 MODEM 속도는 9600bps 이거나 더 빨라야 합니다.
- f) PLC 쪽의 MODEM 에서 사용한 초기 포맷은 ATZ 와 ATS0 =1 입니다.
- g) PLC 가 MODEM 에서 초기화를 할 수 없다면 ATZ 와 ATS0=1 과 같이 PC 가 지원하는 초기화인 Hyper Terminal 을 사용하십시오.

기능 그룹      전력 손실 래치 범위 설정

번호              D1200~D1219

#### 내용:

1. EH, SA, SX, SC 시리즈에서 래치 범위를 정하기 위한 것입니다. 래치 범위는 시작 주소와 끝 주소가 됩니다.
2. 자세한 내용은 2.1 장을 참고하세요..

기능 그룹      강제로 ON/OFF 될 수 있는 입력 지점 X

번호              M1304

#### 내용:

1. SA, SX, SC 시리즈에서 M1304=On 일 때, MPU 의 입력 지점 X(X0-X17)은 주변의 WPLSoft 와 HPP 를 이용하여 강제로 On/Off 가 가능합니다. 그러나 하드웨어 램프는 어떤 작업도 하지 않습니다.

2. EH 시리즈에서 M1304=On 일 때, MPU의 입력 지점 X는 주변 WPLSoft와 HPP를 이용하여 강제로 On/Off 할 수 있고 하드웨어 램프도 이에 따라 작동합니다.

기능 그룹      특수 확장 모듈 ID  
번호              D1320~D1327

내용:

- EH 시리즈에서는 특수 확장 모듈에 연결되었을 때 D1320~D1327에 확장 모듈 ID를 표시합니다.
- EH 시리즈의 특수 확장 모듈 ID:

확장 모듈 이름	확장 모듈 ID	확장 모듈 이름	확장 모듈 ID
DVPEH04AD	H'0400	DVPEH01PU	H'0110
DVP04DA-H	H'0401	DVPEH01HC	H'0120
DVPEH04PT	H'0402	DVPEH02HC	H'0220
DVPEH04TC	H'0403	DVPEH01DT	H'0130
DVPEH06XA	H'0604	DVPEH02DT	H'0230
DVPEH06RT	H'0405		

기능 그룹      이지 PLC 링크  
번호              M1350-M1354, M1360-M1519, D1355-D1370, D1399, D1415-D1465, D1480-D1991

내용:

- PLC가 16개의 외부 장비와 연결할 때 (M1535 = OFF) EH/SA/SX/SC 시리즈 이지 PLC 링크 ID1-ID8의 특수 D, 특수 M에 대한 설명:

MASTER PLC															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353=Off, PLC가 32개의 주변기기를 더 연결하지 못하게 하고 16개 이상의 기능을 읽거나 쓰지 못하게 합니다. (RST M1353) 오직 16개 아이템만이 특수 D 레지스터를 읽고 쓸 수 있습니다.															
D1480   D1495	D1496   D1511	D1512   D1527	D1528   D1543	D1544   D1559	D1560   D1575	D1576   D1591	D1592   D1607	D1608   D1623	D1624   D1639	D1640   D1655	D1656   D1671	D1672   D1687	D1688   D1703	D1704   D1719	D1720   D1735
Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
장비 통신 어드레스															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
SLAVE PLC에 LINK가 있는 경우															
M1360	M1361	M1362	M1363	M1364	M1365	M1366	M1367								
마스터 PLC가 slave PLC에 할 작업에 대한 작업 지시 플래그															
M1376	M1377	M1378	M1379	M1380	M1381	M1382	M1383								
읽기/쓰기 오류 플래그															
M1392	M1393	M1394	M1395	M1396	M1397	M1398	M1399								
완료된 플래그 읽기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1408	M1409	M1410	M1411	M1412	M1413	M1414	M1415								



완료된 플래그 쓰기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1424		M1425		M1426		M1427		M1428		M1429		M1430		M1431	
↓		↓		↓		↓		↓		↓		↓		↓	
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215

읽기 기능의 통신 어드레스 기계 설정은 H1064 (D100)입니다.

쓰기 기능의 통신 어드레스 기계 설정은 H10C8 (D200)입니다.

2. PLC 가 16 개 외부 장비를 연결할 때 EH/SA/SX/SC 시리즈 이지 PLC 링크 ID9-ID16 의 특수 D 와 특수 M 에 대한 설명 (M1353=OFF):

마스터 PLC															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353=Off, PLC 가 32 개의 주변기기를 더 연결하지 못하게 하고 16 개 이상의 기능을 읽거나 쓰기 못하게 합니다. (RST M1353) 오직 16 개 아이템만이 특수 D 레지스터를 읽고 쓸 수 있습니다															
D1736   D1751	D1752   D1767	D1768   D1783	D1784   D1799	D1800   D1815	D1816   D1831	D1832   D1847	D1848   D1863	D1864   D1879	D1880   D1895	D1896   D1911	D1912   D1927	D1928   D1943	D1944   D1959	D1960   D1975	D1976   D1991
Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
장비 통신 어드레스															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
SLAVE PLC 에 LINK 가 있는 경우															
M1368	M1369	M1370	M1371	M1372	M1373	M1374	M1375	M1376	M1377	M1378	M1379	M1380	M1381	M1382	M1383
마스터 PLC 가 slave PLC 에 할 작업에 대한 작업 지시 플래그															
M1384	M1385	M1386	M1387	M1388	M1389	M1390	M1391	M1392	M1393	M1394	M1395	M1396	M1397	M1398	M1399
읽기/쓰기 오류 플래그															
M1400	M1401	M1402	M1403	M1404	M1405	M1406	M1407	M1408	M1409	M1410	M1411	M1412	M1413	M1414	M1415
완료된 플래그 읽기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1416	M1417	M1418	M1419	M1420	M1421	M1422	M1423	M1424	M1425	M1426	M1427	M1428	M1429	M1430	M1431
완료된 플래그 쓰기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1432	M1433	M1434	M1435	M1436	M1437	M1438	M1439	M1440	M1441	M1442	M1443	M1444	M1445	M1446	M1447
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215

읽기 기능의 통신 어드레스 기계 설정은 H1064 (D100)입니다.

쓰기 기능의 통신 어드레스 기계 설정은 H10C8 (D200)입니다.

3. PLC 가 32 개 외부 장비를 연결할 때 EH 시리즈 EASY PLC LINK ID1~ID8 의 특수 D 와 특수 M 에 대한 설명:

마스터 PLC															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
EH MPU 에서 M1353=On 일 때: PLC 가 32 개 이상의 주변기기를 연결하고 16 개 이상의 기기를 읽고 쓸수 있게 합니다. (SET M1353). 읽기/쓰기를 위해 일반 레지스터 D 의 시작 번호를 설정하십시오.															
D1480	D1496	D1481	D1497	D1482	D1498	D1483	D1499	D1484	D1500	D1485	D1501	D1486	D1502	D1487	D1503
Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
장치 통신 어드레스															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
SLAVE PLC 에 LINK 가 있는 경우															
M1360	M1361	M1362	M1363	M1364	M1365	M1366	M1367								
마스터 PLC 가 slave PLC 에 할 작업에 대한 작업 지시 플래그															
M1376	M1377	M1378	M1379	M1380	M1381	M1382	M1383								
읽기/쓰기 오류 플래그															
M1392	M1393	M1394	M1395	M1396	M1397	M1398	M1399								
완료된 플래그 읽기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1408	M1409	M1410	M1411	M1412	M1413	M1414	M1415								
완료된 플래그 쓰기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1424	M1425	M1426	M1427	M1428	M1429	M1430	M1431								

↓

↓

↓

↓

↓

↓

↓

↓

SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215

읽기 기능의 통신 어드레스 기계 설정은 H1064 (D100)입니다.

쓰기 기능의 통신 어드레스 기계 설정은 H10C8 (D200)입니다.

4. PLC 가 32 개 외부 장비를 연결할 때 EH 시리즈 이지 PLC 링크 ID9~ID16 의 특수 D 와 특수 M 에 대한 설명 (M1353=On):

마스터 PLC															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
EH MPU 에서 M1353=On 일 때: PLC 가 32 개 이상의 주변기기를 연결하고 16 개 이상의 기기를 읽고 쓸수 있게 합니다. (SET M1353). 읽기/쓰기를 위해 일반 레지스터 D 의 시작 번호를 설정하십시오.															
D1488	D1504	D1489	D1505	D1490	D1506	D1491	D1507	D1492	D1508	D1493	D1509	D1494	D1510	D1495	D1511
Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
장치 통신 어드레스															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
SLAVE PLC 에 LINK 가 있을 경우															
M1368	M1369	M1370	M1371	M1372	M1373	M1374	M1375								
마스터 PLC 가 slave PLC 에 할 작업에 대한 작업 지시 플래그															
M1384	M1385	M1386	M1387	M1388	M1389	M1390	M1391								
읽기/쓰기 오류 플래그															
M1400	M1401	M1402	M1403	M1404	M1405	M1406	M1407								
완료된 플래그 읽기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1416	M1417	M1418	M1419	M1420	M1421	M1422	M1423								
완료된 플래그 쓰기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1432	M1433	M1434	M1435	M1436	M1437	M1438	M1439								

↓

↓

↓

↓

↓

↓

↓

↓

SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215

읽기 기능의 통신 어드레스 기계 설정은 H1064 (D100)입니다.

쓰기 기능의 통신 어드레스 기계 설정은 H10C8 (D200)입니다.

5. PLC 가 32 개 외부 장치를 연결할 때 EH 시리즈 이지 PLC 링크 ID17~ID24 의 특수 D 와 특수 M 에 대한 설명 (M1353=On):

마스터 PLC															
SLAVE ID 17		SLAVE ID 18		SLAVE ID 29		SLAVE ID 20		SLAVE ID 21		SLAVE ID 22		SLAVE ID 23		SLAVE ID 24	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
EH MPU 에서 M1353=On 일 때: PLC 가 32 개 이상의 주변기기를 연결하고 16 개 이상의 기기를 읽고 쓸수 있게 합니다. (SET M1353). 읽기/쓰기를 위해 일반 레지스터 D 의 시작 번호를 설정하십시오.															
D1576	D1592	D1577	D1593	D1578	D1594	D1579	D1595	D1580	D1596	D1581	D1597	D1582	D1598	D1583	D1599
Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number
D1544	D1560	D1545	D1561	D1546	D1562	D1547	D1563	D1548	D1564	D1549	D1565	D1550	D1566	D1551	D1567
장치 통신 어드레스															
D1512	D1528	D1513	D1529	D1514	D1530	D1515	D1531	D1516	D1532	D1517	D1533	D1518	D1534	D1519	D1535
SLAVE PLC 에 LINK 가 있는 경우															
M1440	M1441	M1442	M1443	M1444	M1445	M1446	M1447								
마스터 PLC 가 slave PLC 에 할 작업에 대한 작업 지시 플래그															
M1456	M1457	M1458	M1459	M1460	M1461	M1462	M1463								
읽기/쓰기 오류 플래그															
M1472	M1473	M1474	M1475	M1476	M1477	M1478	M1479								
완료된 플래그 읽기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1488	M1489	M1490	M1491	M1492	M1493	M1494	M1495								
완료된 플래그 쓰기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1504	M1505	M1506	M1507	M1508	M1509	M1510	M1511								

↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

SLAVE ID 17		SLAVE ID 18		SLAVE ID 29		SLAVE ID 20		SLAVE ID 21		SLAVE ID 22		SLAVE ID 23		SLAVE ID 24	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200	D100	D200
D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215	D115	D215

읽기 기능의 통신 어드레스 기계 설정은 H1064 (D100)입니다.

쓰기 기능의 통신 어드레스 기계 설정은 H10C8 (D200)입니다.

6. PLC 가 32 개 외부 장치를 연결할 때 EH 시리즈 이지 PLC 링크 ID25~ID32 의 특수 D 와 특수 M 에 대한 설명 (M1353=On):

마스터 PLC															
SLAVE ID 25		SLAVE ID 26		SLAVE ID 27		SLAVE ID 28		SLAVE ID 29		SLAVE ID 30		SLAVE ID 31		SLAVE ID 32	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
EH MPU 에서 M1353=On 일 때: PLC 가 32 개 이상의 주변기기를 연결하고 16 개 이상의 기기를 읽고 쓸수 있게 합니다. (SET M1353). 읽기/쓰기를 위해 일반 레지스터 D 의 시작 번호를 설정하십시오.															
D1584	D1600	D1585	D1601	D1586	D1602	D1587	D1603	D1588	D1604	D1589	D1605	D1590	D1606	D1591	D1607
Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number	Item number
D1552	D1568	D1553	D1569	D1554	D1570	D1555	D1571	D1556	D1572	D1557	D1573	D1558	D1574	D1559	D1575
장치 통신 어드레스															
D1520	D1536	D1521	D1537	D1522	D1538	D1523	D1539	D1524	D1540	D1525	D1541	D1526	D1542	D1527	D1543
SLAVE PLC 에 LINK 가 있는 경우															
M1448	M1449	M1450	M1451	M1452	M1453	M1454	M1455								
마스터 PLC 가 slave PLC 에 할 작업에 대한 작업 지시 플래그															
M1464	M1465	M1466	M1467	M1468	M1469	M1470	M1471								
읽기/쓰기 오류 플래그															
M1480	M1481	M1482	M1483	M1484	M1485	M1486	M1487								
완료된 플래그 읽기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1496	M1497	M1498	M1499	M1500	M1501	M1502	M1503								
완료된 플래그 쓰기(PLC 읽기/쓰기를 완료할 때마다 이 플래그는 자동으로 Off 됩니다.)															
M1512	M1513	M1514	M1515	M1516	M1517	M1518	M1519								

↓

↓

↓

↓

↓

↓

↓

↓

SLAVE ID 25		SLAVE ID 26		SLAVE ID 27		SLAVE ID 28		SLAVE ID 29		SLAVE ID 30		SLAVE ID 31		SLAVE ID 32	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215	D100   D115	D200   D215

읽기 기능의 통신 어드레스 기계 설정은 H1064 (D100)입니다.

쓰기 기능의 통신 어드레스 기계 설정은 H10C8 (D200)입니다.

### 설명:

1. EASY PLC LINK 의 기본통신 프로토콜은 MODBUS 입니다.
2. EH 시리즈 MPU 에서 사용자는 M1353=On 로 설정하여 PLC 가 32 개 이상의 주변기기와 연결하고 16 개 이상을 읽고 쓸수 있도록 할 수 있습니다. 또한 일반 레지스터 D 의 시작 번호를 설정하여 읽고 쓰도록 할 수 있습니다. SA/SX/SC 시리즈 MPU 에서 PLC 는 오직 16 개 외부 장치를 연결할 수 있고 16 개 아이টে에 읽고 쓸수 있습니다.
3. EH 시리즈 모델에서 마스터 PLC 가 Slave 주변기기에 읽고 쓸 수있는 최대 데이터는 100 단어 입니다. SA/SX/SC 시리즈 모델에서는 16 단어 입니다.
4. 마스터 PLC 가 다른 Slave 주변기기에 COM2(RS-485)를 이용해 연결했다면 slave 주변기기와 통신 포맷의 변조 속도는 마스터 PLC 와 동일합니다. SA, SX, SC, EH MPU 가 slave 로 쓰였다면 ASCII 모드와 RTU 모드가 사용될 수 있습니다.
5. Slave PLC 가 다른 주변기기에 COM2(RS-232/RS-485/RS-422)를 이용하여 연결했다면 모든 slave 주변기기와 통신 포맷의 변조 속도는 마스터 PLC 와 같습니다. SA/SX/SC/EH MPU 가 slave 로 쓰였다면 ASCII 모드와 RTU 모드가 사용될 수 있습니다. (통신 포트의 기능 설명은 P2-P6 을 참조하십시오.)
6. Slave PLC 가 다른 주변기기에 COM1(RS-232)을 이용하여 연결했다면 모든 slave 주변기기와 통신 포맷의 변조 속도는 마스터 PLC 와 동일합니다. (PLC 의 D1036) SA/SX/SC/EH MPU 가 slave 로 쓰였다면 ASCII 모드와 RTU 모드가 사용될 수 있습니다. (통신 포트의 기능 설명은 P2-P6 을 참조하십시오.)
  - Slave PLC 가 다른 주변기기에 COM3 (RS-232/RS-485)을 이용하여 연결했다면, 다른 모든 slave 주변기기와 통신 포맷의 변조 속도는 마스터 PLC(D1109 와 같은)의 것과 같습니다. ASCII 모드는 최대 변조속도가 38,400 bps 까지일 때만 가능합니다.
7. Slave PLC 의 시작 ID 는 마스터 PLC 의 D1399 에 의해 설정될 수 있습니다. (ID 는 K1~K214 입니다). 마스터 PLC 와 slave PLC 의 ID 는 반복될 수 없습니다.
8. RS-232, RS-485, RS-422 는 일대 일 통신에 사용될 수 있습니다. PLC MPU 의 COM1, COM2, COM3 는 다중 통신 포맷을 지원합니다. (통신 포트의 설명은 P2-P6 를 참조하십시오.)
9. 일대 다 통신(One to multiple connection)은 RS-485 에 연결될 수 있습니다. PLC MPU 의 COM1, COM2, COM3 는 다중 통신 포맷을 지원합니다. (통신 포트의 설명은 P2-P6 를 참조하십시오.)

### 작업:

1. COM1\_RS-232:D1036·COM2\_RS-232/RS-485/RS-422:D1120·COM3\_RS-232/RS-485:D1109. 모든 slave 주변 기기의 변조 속도와 통신 포맷은 마스터 PLC 와 동일해야 합니다. (COM1\_RS-232: D1036, COM2\_RS-232/RS-485/RS-422: D1120, COM3\_RS-232/RS-485: D1109)
2. 먼저 마스터 PLC ID 는 D1121 를 통해서 설정하고 slave ID 도 설정하십시오. ID 는 반복될 수 없습니다. 마스터 PLC 의 ID 는 먼저 설정되어야 하고 그 뒤 Slave PLC 의 시작 ID 를 마스터 PLC 의 D1399 를 통해 설정하십시오. 마스터 PLC 와 Slave PLC 의 ID 는 반복될 수 없습니다.
3. slave 의 읽기/쓰기 아이টে를 설정합니다. (최대 개수는 16 개 입니다.) M1353=On 일 때 EH 시리즈에서는 32 개 주변기기(Slave)를 연결할 수 있고 16 개 이상의 아이টে를 읽고 쓸 수 있으며, 읽고 쓸수 있는 최대 데이터는 100 단어 입니다. 그리고 사용자는 읽기/쓰기를 위해 일반 레지스터 D 의 시작 번호를 설정할 수 있습니다. D1480~D1495/D1576~D1591 은 읽은 뒤의 저장을 위한 것이고 D1496~D1511/D1592~1607 는 쓴

뒤의 저장을 위한 것입니다. (특수 D 설정에 관해서는 위의 특수 D 설명을 참조하십시오) SA/SX/SC 시리즈에서는 오직 16 개 아이템만 읽고 쓸 수 있습니다.

4. 장치 통신 어드레스를 설정하여 **slave** 에 읽고 쓸 수 있도록 합니다. (특수 D 설정은 위의 특수 D 설명을 참조하십시오) **Slave PLC** 에 통신 어드레스와 읽고 쓸 아이템들을 설정합니다. (가능한 설정이 없다면 내부 설정이나 최소 설정을 이용하십시오)(특수 D 설정은 위의 특수 D 설명을 참조하십시오)
5. 읽기를 위한 통신 어드레스의 기계 설정은 **H1064 (D100)**이고, 쓰기는 **H10C8 (D200)**입니다. **Slave PLC** 에 읽기/쓰기를 위한 통신 어드레스를 설정하십시오. (특수 D 설정은 위의 특수 D 설명을 참조하십시오).
6. **M1354 = On** 으로 설정하여 읽고 쓰기를 동시에 할 수 있도록 합니다.
7. **M1352=ON** 일 때, PLC 는 수동 기능을 활성화 시키고 통신 시간을 **D1431** 을 통해 설정합니다.  
**M1351=ON** 일 때, PLC 는 자동 기능을 활성화 합니다. 수동 및 자동 기능은 동시에 사용될 수 없습니다.
8. **M1350 = On** 으로 설정하여 마스터 PLC 링크를 시작하십시오.
9. 작업:
  - a) PLC LINK 를 설정하여 32 개 주변기기의 기능을 사용할 수 있도록 하고 16 개 이상의 아이템을 읽고 쓸 수 있도록 합니다. (M1353) EH 시리즈에서는 PLC 링크가 32 개 주변기기 연결을 가능하게 하여 16 개 이상의 아이템을 읽고 쓸 수 있게 합니다. SA/SX/SC 시리즈에서는 PLC LINK 가 16 개 주변기기의 연결과 16 개 이상의 읽고 쓰기 기능을 가능하게 합니다.
  - b) PLC LINK 을 설정하여 읽고 쓰기 기능을 동시에 할 수 있도록 합니다. (M1354).
  - c) PLC LINK 을 설정하여 자동 모드(M1351)나 수동 모드(M1352)를 정하고 통신 시간도 정합니다. (D1431)
  - d) 마스터 PLC LINK 를 활성화 합니다. (M1350)

### 마스터 PLC 작업 설명:

1. **Slave ID** 감지: **M1350=On** 일 때, 마스터 PLC 는 시작되고 slave 번호를 감지하여 **D1433** 에 그 번호를 저장합니다. **Slave ID** 감지: 먼저, PLC LINK 를 자동 모드로 설정(**M1351=on**)하거나 수동 모드로 설정합니다(**M1352=on**). 이 두가지 모드는 동시에 ON 이 될 수 없다는 것에 주의하시기 바랍니다.  
**M1350=on** 일 때, 마스터 PLC 는 시작되고 slave 번호를 감지하여 **D1433** 에 이를 저장합니다. 감지 시간은 slave 번호에 따라, 그리고 통신 시간제한 설정에 따라 다릅니다. (D1129).
2. **slave ID 1-32** 를 각각 저장하고 있는 **M1360-M1375, M1440~M1455** 를 통해 **slave PLC** 가 존재하는 지를 알 수 있습니다.
3. 감지된 **slave PLC** 번호가 0 이라면, **M1350** 는 Off 되고 동시에 연결을 중지합니다.
4. **M1353** 와 **M1354** 는 PLC EASY LINK 가 활성화되기 이전에 설정되어야 합니다. 이 두 특수 M 은 PLC EASY LINK 의 실행에 영향을 미치지 않습니다.
5. **M1353=on** 일 때 ID1 Slave PLC 의 **D1480~D1511dml** 읽기/쓰기 버퍼는 **Slave PLC** 의 ID1~ID16 를 위해 읽기/쓰기의 레지스터 D 를 저장하는 데 쓰입니다. 레지스터 D 의 설정 범위는 0 부터 9900 입니다. 설정이 이 범위를 넘어서면 자동으로 9900 으로 설정됩니다. 또한 읽기/쓰기의 최대 수는 100 이며, 0 보다 작거나 100 보다 클 때에는 100 으로 변환됩니다.
6. **M1354=On** 이라면, PLC 이지 링크의 통신 기능이 **Modbus Function H17** (동시 read/write)로 설정되어 있다는 것을 뜻합니다. 쓰기 아이템의 설정이 0 이라면 PLC 이지 링크 기능은 **Modbus Function H03** (여러 단어 읽기)로 바뀔 것입니다. 이와 같이 읽기 아이템의 설정이 0 이라면 PLC 이지 링크 기능은 **Modbus Function H06** (한 단어 쓰기) 또는 **Modbus Function H10**(여러 단어 쓰기)로 바뀝니다.



7. 읽기/쓰기의 아이템 숫자가 16 보다 많다면 D1129 를 500ms 이상 설정하여 시간 만료가 생기지 않도록 하십시오
8. PLC 는 M1350 = On 의 시작시에만 Slave PLC 의 번호를 감지합니다.
9. 마스터 및 slave PLC 의 읽기/쓰기: slave 를 감지한 후, 마스터 PLC 는 각 slave 에 읽고 씁니다. 마스터가 읽고 쓸 수 있는 slave 는 slave ID 를 감지한 후의 slave ID 입니다. Slave PLC 가 감지작업 후 추가되면 마스터는 다음 감지작업때까지 읽고 쓸 수 없습니다.
10. 마스터 PLC 는 읽기를 먼저 하며 최대 범위는 D100 에서 시작하는 16 개 slave PLC 입니다. 읽기를 한 뒤 PLC 는 쓰며, 최대 범위는 D200 에서 시작하는 16 개 slave PLC 입니다. 마스터 PLC 는 우선적으로 설정 범위에 따라 읽기를 수행하며, 그 후에 설정에 따라 쓰기를 합니다.
11. 마스터 PLC 는 순서대로 slave PLC 에 읽고 씁니다. 다시 말하면 마스터는 한 slave 를 마친 뒤에 그 다음 slave 로 넘어갑니다.

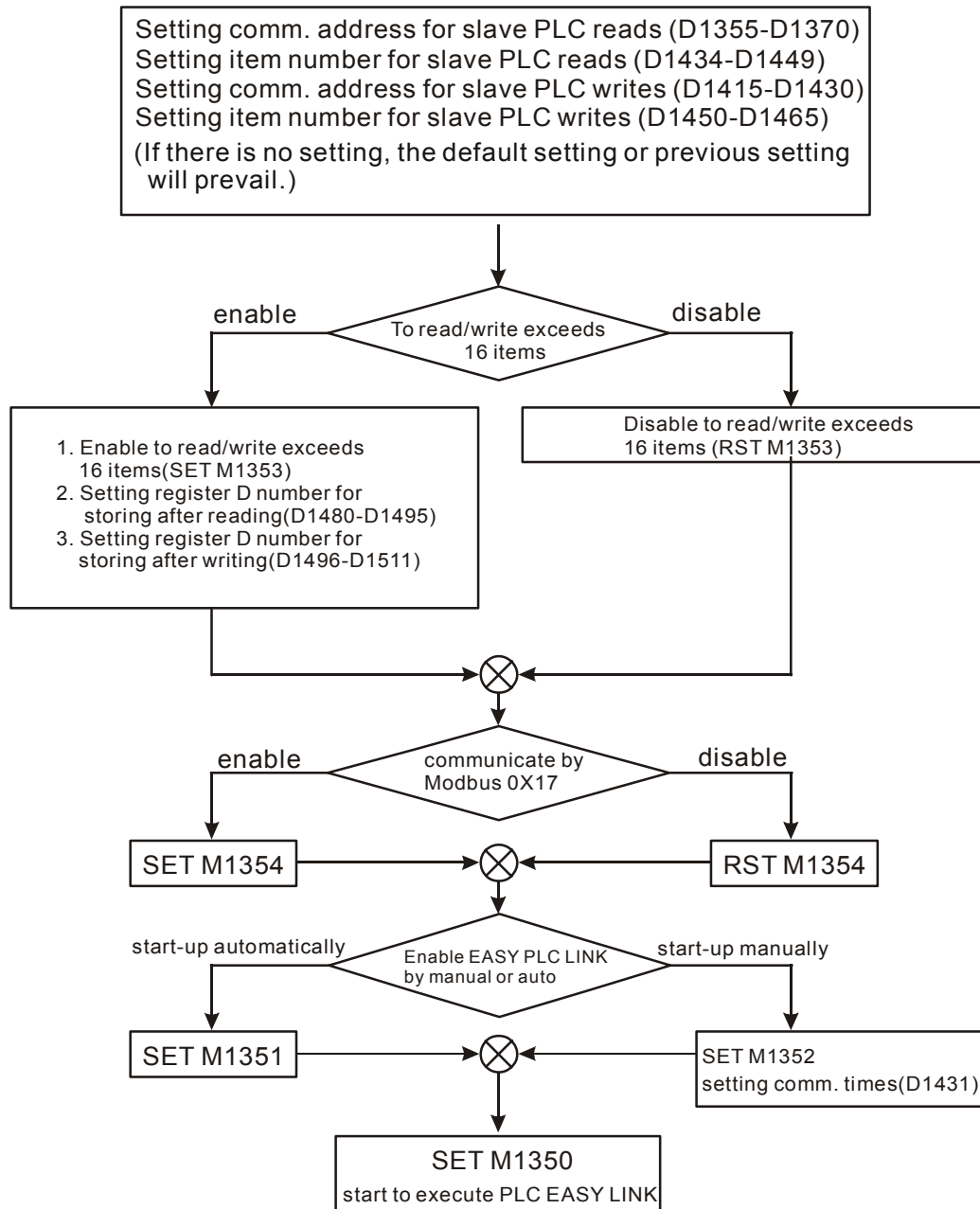
### 자동/수동 모드 설명:

1. 자동 모드: M1351 = Off 로 설정되어야 합니다. 마스터 PLC 는 M1350 이 Off 될 때까지 slave 에 읽고 씁니다.
2. 수동 모드: 읽는 시간을 D1431 에 설정해야 합니다. 한 번은 모든 slave 를 읽고 쓰는 것 완료한다는 말입니다. PLC 가 링크를 시작할 때 D1432 는 링크의 숫자를 세기 시작합니다. D1431 = D1432 일 때, PLC 는 링크를 멈추고 동시에 M1351 이 Off 되도록 합니다. M1351 이 강제로 On 이 되었다면 PLC 는 D1431 의 값에 따라 자동으로 링크를 시작합니다.

### 주의:

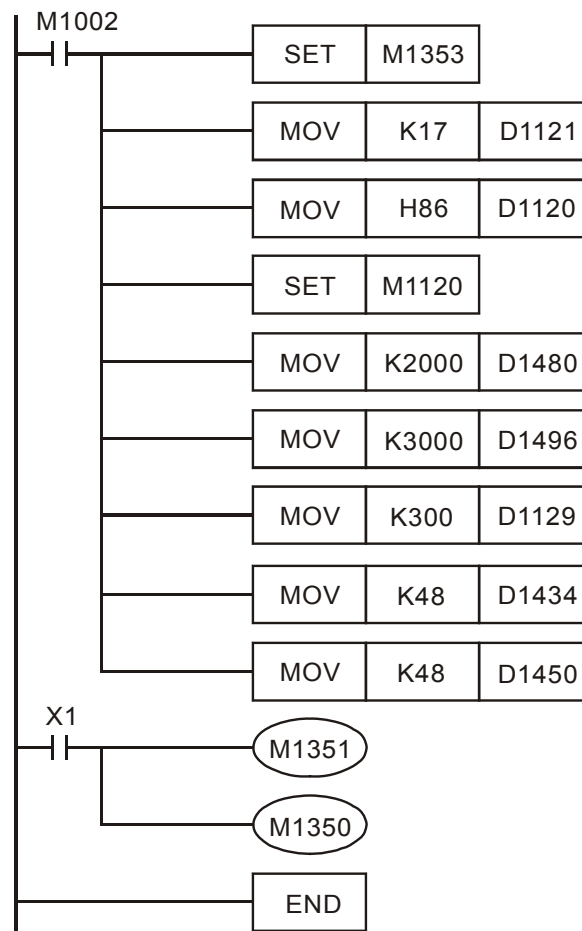
1. 자동 모드 M1351 와 수동 모드 M1352 는 동시에 On 이 될 수 없습니다. 사용자가 M1352 를 On 으로 설정한 뒤 M1351 을 On 으로 설정한다면 PLC 는 링크를 멈추고 동시에 M1350 을 Off 할 것입니다.
2. EH 모델에서는 자동/수동 모드를 변경하기 전에 M1350 을 Off 해야 합니다. SA/SX/SC 모델에서는 불필요합니다.
3. 통신 시간 제한은 D1129 에 의해 설정됩니다. 설정 범위는 200 부터 3000 까지 입니다. 이 범위를 벗어나서 200 보다 작으면 200 으로 간주되고 3000 보다 크면 3000 으로 간주됩니다. 또한 이 설정은 링크 이전에만 유효합니다. 읽기/쓰기 아이템이 16 개 보다 많을 때에는 통신 시간 제한 설정이 500ms 이상으로 설정되어야 합니다.
4. PLC LINK 기능은 변조 속도가 1200bps 를 넘을 때에만 유효합니다. 변조 속도가 9600bps 보다 작을 때에는 통신 시간 제한을 1 초 이상으로 설정하십시오.
5. 통신 불가능 읽기/쓰기 아이템이 0 일 때 입니다.
6. 32 비트 카운터 읽기/쓰기는 지원하지 않습니다.

### PLC 이지 링크의 작업 흐름



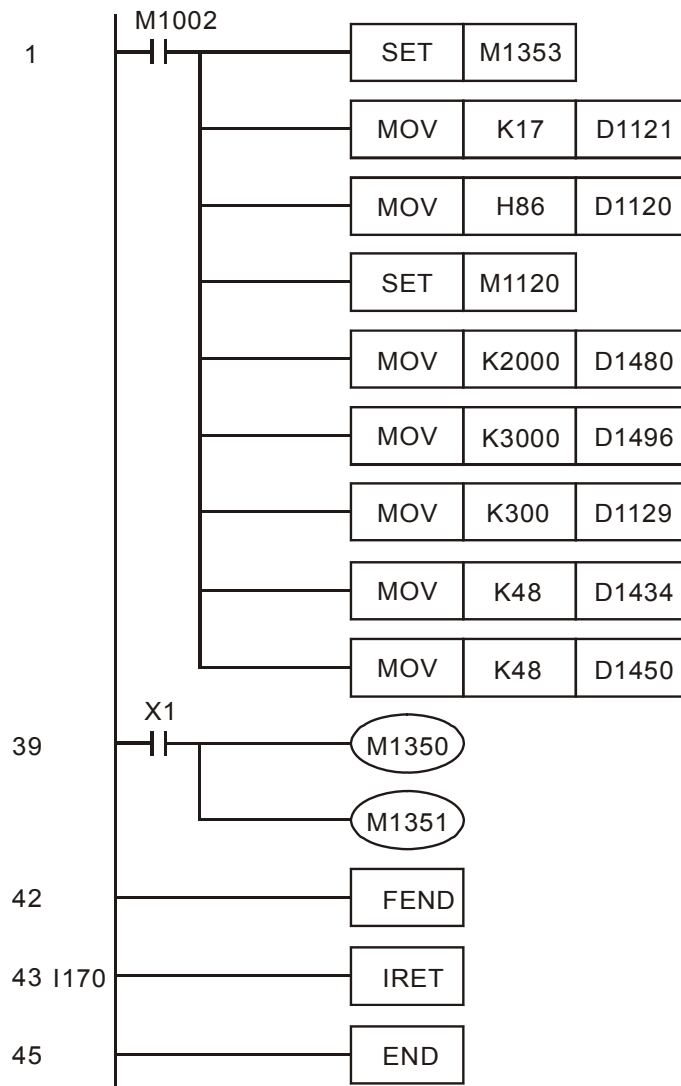
예

예 1: M1353 을 이용하여 PLC EASY LINK 사용하기



1. M1353 은 PLC 이지 링크를 활성화하기 전에 설정되어야 합니다. 활성화 되고 난 이후에는 M1353 을 감지하지 못합니다.
2. D1480~D1495 는 마스터 PLC 읽기를 위한 시작 레지스터 D 번호이고, D1496~D1511 는 마스터 PLC 쓰기를 위한 시작 레지스터 D 번호입니다. 레지스터 D 만이 지정될 수 있고 각 특수 D 는 하나의 slave PLC 를 뜻합니다. D1480 은 ID1 을, D1481 은 ID2 를 뜻하는 것입니다.
3. D1480~D1495 와 D1496~D1511 를 PLC 이지 링크 활성화 이전에 설정할 수 있습니다. 마스터 PLC 는 slave 감지 후 각 slave 에 읽고 쓸 것입니다. 마스터 PLC 가 읽고 쓸 수 있는 slave PLC 는 slave ID 감지 후 부여된 slave ID 입니다. 감지 이후 PLC 가 추가되면 마스터는 다음 감지 작업때 까지 이를 읽고 쓸 수 없습니다.
4. D1480~D1495 와 D1496~D1511 의 설정 범위는 0 ~ 9900 입니다. 설정이 이 범위를 벗어나면 9900 으로 설정됩니다.
5. D1434~D1449 는 읽기의 길이(D1434 는 ID1 읽기의 길이)이고, D1450~D1465 (D1450 는 ID1 의 쓰기 길이)는 쓰기의 길이입니다. PLC 이지 링크를 M1353 없이 실행할 때에는 이 길이의 범위가 0 ~ 16 입니다. 이 범위를 벗어나면 16 으로 설정됩니다. M1353 을 실행했을 때에는 길이 범위가 0 ~ 48 입니다. 이 범위를 벗어나면 48 로 설정됩니다.
6. PLC 이지 링크를 활성화 한 후 D1434~D1449 와 D1450~D1465 를 설정할 수는 있습니다. 그러나 이 설정은 다음 감지 작업때 까지만 유효합니다.

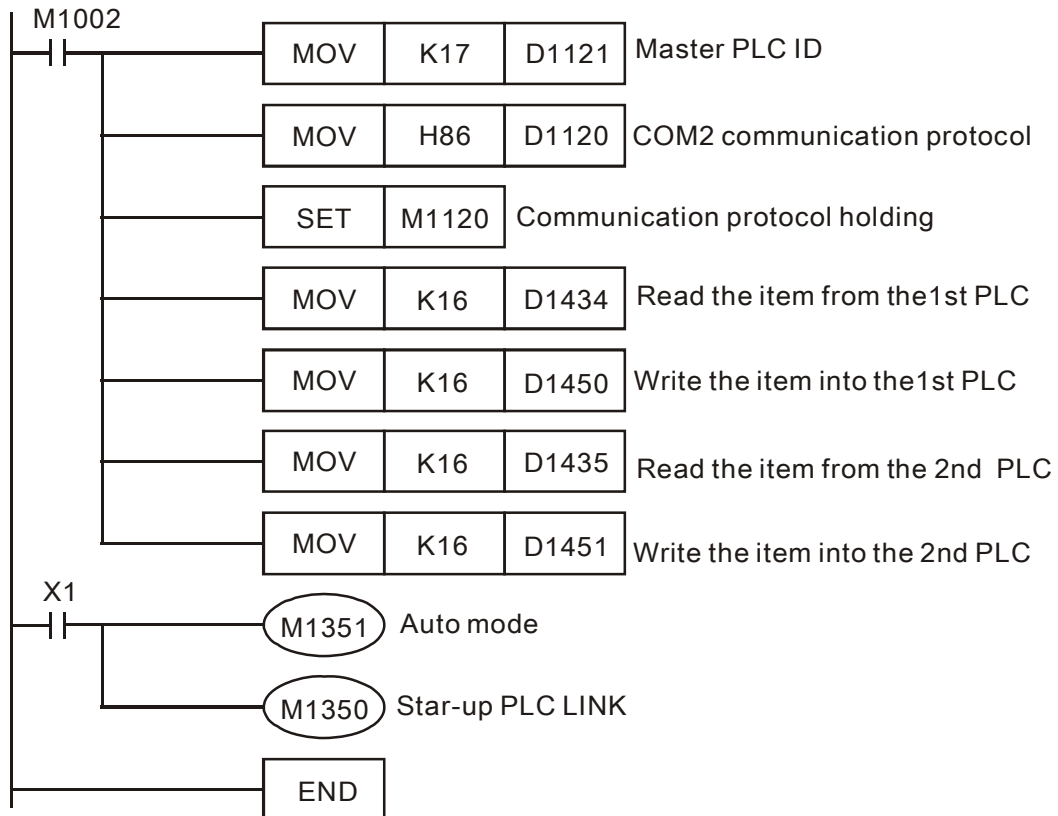
예 2: M1353 과 I170 으로 PLC 이지 링크 이용하기



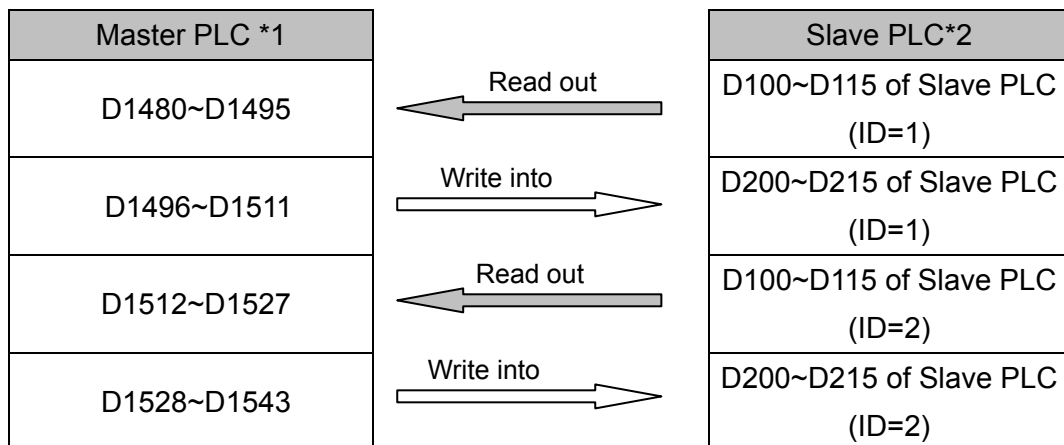
1. 데이터를 모두 전송받은 뒤에는 즉시 I170 을 활성화하여 데이터를 처리합니다.
2. slave PLC 의 반응 속도가 느리다면 I170 을 사용하는 것을 권하지 않습니다.
3. PLC 이지 링크를 위한 추가적인 특수 D(D1399)은 slave PLC 의 시작 ID 를 정해놓기 위한 것입니다.
4. D1399 는 slave PLC 의 시작 ID 를 지정하는데 쓰입니다. 구 버전(1.0 이하)에서 마스터 PLC 가 연결할 수 있는 slave ID 는 1~16 이었습니다. EH MPU 의 1.1 이후 버전에서는 사용자가 시작 ID 를 지정하고 마스터는 그 이하의 연속되는 15 개 slave ID 를 감지합니다. 예를 들면, D1399 가 k20 으로 설정되어 있을 때 마스터 PLC 는 20 부터 35 까지의 ID 를 감지합니다.

예 3: 3 개의 PLC 가 RS-485 를 통해 연결되어 있을 때 PLC 이지 링크를 이용하여 하나의 마스터 PLC 와 2 개의 slave PLC 간 16 개 아이템을 교환합니다. (M1353=OFF, 16 PLC LINK, 16 개 아이템 읽기/쓰기)

◆ 마스터 PLC (ID=17)에서 다음의 래더 다이어그램(ladder diagram) 프로그램을 씁니다:



X1=ON 일 때 PLC 링크를 사용하여 마스터 PLC 와 2 개의 slave PLC 간 데이터 교환을 자동으로 완성할 수 있습니다. 다시 말하면 두 slave PLC 의 D100~D115 에 있는 데이터가 마스터 PLC 의 D1480~D1495 와 D1512~D1527 로 자동으로 읽힌다는 것입니다. 또한 마스터 PLC 의 D1496~D1511 와 D1528~D1543 에 있는 데이터는 각각 slave PLC 의 D200~D215 에 쓰입니다.



- ◆ PLC LINK 를 활성화 하기 전에(M1350=OFF), D 의 데이터는 마스터 PLC 와 slave PLC 의 데이터 교환을 위한 것이라 가정합니다. (다음의 표와 같이):

Master PLC	설정값	Slave PLC	설정값
D1480~D1495	모두 K0	D100~D115 of Slave PLC (ID=1)	모두 K5000
D1496~D1511	모두 K1000	D200~D215 of Slave	모두 K0

Master PLC	설정값	Slave PLC	설정값
		PLC (ID=1)	
D1512~D1527	모두 K0	D100~D115 of Slave PLC (ID=2)	모두 K6000
D1528~D1543	모두 K2000	D200~D215 of Slave PLC (ID=2)	모두 K0

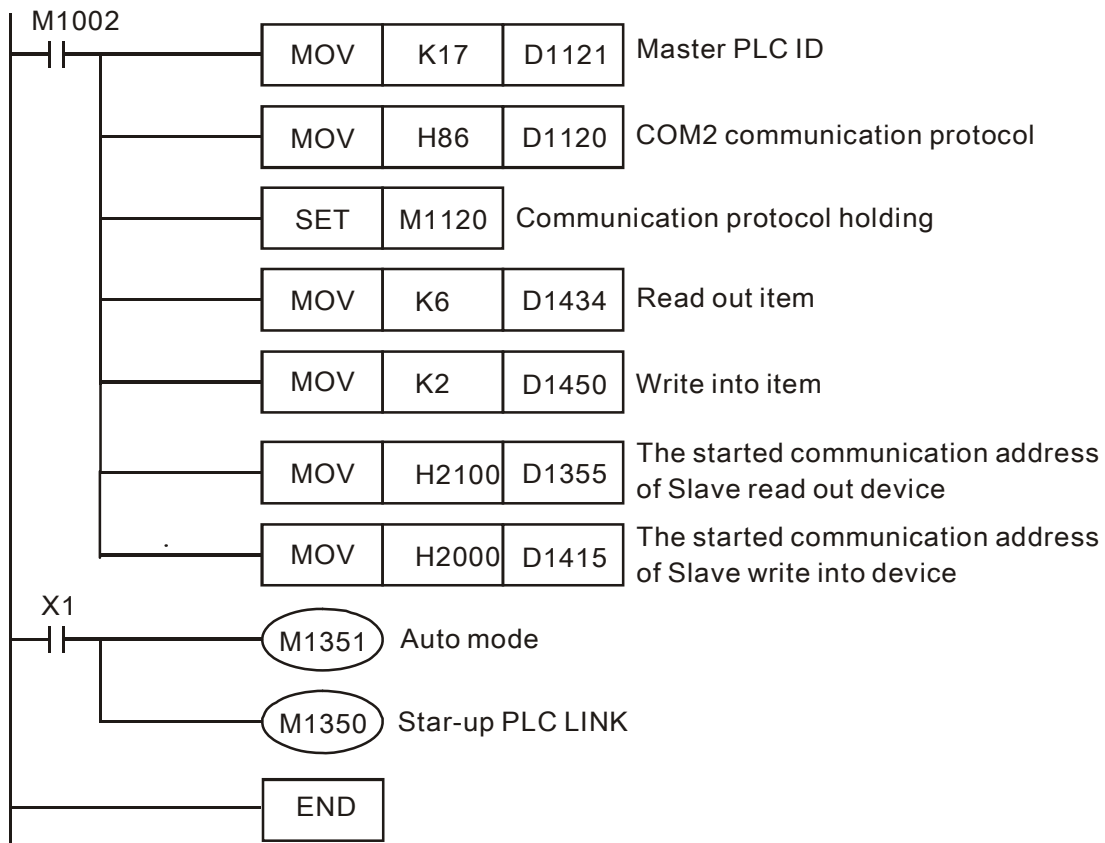
PLC LINK 을 활성화한 후 (M1350=ON), D 의 데이터는 마스터 PLC 와 Slave PLC 의 교환을 위한 것이라 가정합니다. (다음의 표와 같이):

Master PLC	설정값	Slave PLC	설정값
D1480~D1495	모두 K5000	D100~D115 of Slave PLC (ID=1)	모두 K5000
D1496~D1511	모두 K1000	D200~D215 of Slave PLC (ID=1)	모두 K1000
D1512~D1527	모두 K6000	D100~D115 of Slave PLC (ID=2)	모두 K6000
D1528~D1543	모두 K2000	D200~D215 of Slave PLC (ID=2)	모두 K2000

- ◆ 마스터 PLC 는 SA/SX/SC/EH 모델로 가정합니다. Slave PLC 는, 모든 DVP-PLC 시리즈가 가능합니다.
- ◆ Slave PLC 의 개수는 16 개까지 가능합니다. 각 slave PLC 의 D100~D115 와 D200~D215 에 있는 정보는 마스터 PLC 의 특수 D 와 일치합니다. 각 장에 있는 특수 D 와 특수 M 에 관한 설명을 참조하십시오.

예 4: Delta PLC 가 Delta VFD-M 시리즈 AC 모터 드라이브에 연결되어 있을 때 PLC 링크를 이용하여 START 를 정지시킬 수 있고 앞/뒤 회전 컨트롤 주파수를 읽고 쓸수도 있습니다.

- ◆ 마스터 PLC 에서 (ID=17), 아래의 래더 다이어그램 프로그램을 쓰십시오:



- ◆ PLC의 D1480~D1485는 AC 모터 드라이브의 변수 H2100~H2105에 해당됩니다. X1=ON가 되어 PLC 링크를 시작할 때 변수 H2100~H2105의 데이터는 D1480~D1485에 표시됩니다.
- ◆ PLC의 D1496~D1497는 AC 모터 드라이브의 변수 H2000~H2001에 해당합니다. X1=ON가 되어 PLC 링크를 시작할 때 변수 H2000~H2001의 값들은 D1496~D1497에 의해 결정됩니다.
- ◆ PLC의 D1496는 VFD에 명령을 내릴 수 있습니다. (예: D1496=H12로 설정하여 AC 모터 드라이브의 앞 회전을 시작하거나 D1496=H1로 설정하여 AC 모터 드라이브를 정지시킬 수 있습니다.)
- ◆ PLC의 D1497을 조정하여 AC 모터 드라이브의 주파수를 저장할 수 있습니다. (예: D1497=K5000일 때 AC 모터 드라이브의 주파수는 50kHz입니다.)
- ◆ 마스터 PLC는 SA/SX/SC/EH 모델로 가정합니다. AC 모터 드라이브가 slave로 쓰일 경우, VFD-A 시리즈를 제외한 모든 모델이 가능합니다.
- ◆ MODBUS 통신 프로토콜에 의해 허가를 받으면 Delat 온도 조절기, DTA, DTB 시리즈, Delata AC servo drive와 같은 장치들은 Slave로 사용될 수 있습니다. 연결 장치는 최대 16개입니다.
- ◆ 특수 레지스터에 저장되는 어드레스 데이터(읽기/쓰기를 위한)와 아이템 데이터(읽기/쓰기를 위한)에 관해서는 각 장의 특수 D와 특수 M에 관한 설명 표를 참조하십시오.

주의:

1. D1399의 설정 범위는 1~230입니다. 설정이 230을 넘으면 230으로 자동으로 설정되고 1보다 작으면 1로 설정됩니다.
2. D1399는 PLC 이치 링크를 실행하기 이전에 유효합니다.
3. 추가 기능의 이점: 다중 네트워크 구조에서 더 많은 slave PLC 연결이 가능해 집니다. 예로서 PLC 이치 링크를 구 버전(1.0 이하)에서 3중 네트워크에 이용하는 경우를 들 수 있습니다. slave ID 1-16이 두번째

레이어에 사용되면 사용자는 세번째 레이어에서 ID15 밖에 사용할 수 없습니다. 그러나 1.1 이후 버전에서는 사용자가 세번째 레이어에서도 16 개 slave 를 이용할 수 있습니다.

### DVP 시리즈 PLC 내부 장치 통신 어드레스

장 치	범 위		타입	어드레스	효 과		
					ES/EX/SS	SA/SX/SC	EH
S	000~255		bit	0000~00FF	0~127	0~1024	0~1024
S	246~511		bit	0100~01FF	-		
S	512~767		bit	0200~02FF			
S	768~1023		bit	0300~03FF			
X	000~377 (Octal)		bit	0400~04FF	0~177	0~177	0~377
Y	000~377 (Octal)		bit	0500~05FF	0~177	0~177	0~377
T	000~255		bit/word	0600~06FF	0~127	0~255	0~255
M	000~255		bit	0800~08FF	0~1279	0~4095	0~4095
M	256~511		bit	0900~09FF			
M	512~767		bit	0A00~0AFF			
M	768~1023		bit	0B00~0BFF			
M	1024~1279		bit	0C00~0CFF			
M	1280~1535		bit	0D00~0DFF	-		
M	1536~1791		bit	B000~B0FF			
M	1792~2047		bit	B100~B1FF			
M	2048~2303		bit	B200~B2FF			
M	2304~2559		bit	B300~B3FF			
M	2560~2815		bit	B400~B4FF			
M	2816~3071		bit	B500~B5FF			
M	3072~3327		bit	B600~B6FF			
M	3328~3583		bit	B700~B7FF			
M	3584~3839		bit	B800~B8FF			
M	3840~4095		bit	B900~B9FF			
C	0~199	16-bit	bit/word	0E00~0EC7	0~127	0~199	0~199
	200~255	32-bit	bit/Dword	0EC8~0EFF	232~255	200~255	200~255
D	000~256		word	1000~10FF	0~1311	0~4999	0~9999
D	256~511		word	1100~11FF			
D	512~767		word	1200~12FF			
D	768~1023		word	1300~13FF			
D	1024~1279		word	1400~14FF			
D	1280~1535		word	1500~15FF			



장 치	범 위	타 입	어 드 레 스	효 과		
				ES/EX/SS	SA/SX/SC	EH
D	1536~1791	word	1600~16FF	-	0~4999	0~9999
D	1792~2047	word	1700~17FF			
D	2048~2303	word	1800~18FF			
D	2304~2559	word	1900~19FF			
D	2560~2815	word	1A00~1AFF			
D	2816~3071	word	1B00~1BFF			
D	3072~3327	word	1C00~1CFF			
D	3328~3583	word	1D00~1DFF			
D	3584~3839	word	1E00~1EFF			
D	3840~4095	word	1F00~1FFF			
D	4096~4351	word	9000~90FF			
D	4352~4607	word	9100~91FF			
D	4608~4863	word	9200~92FF			
D	4864~5119	word	9300~93FF			
D	5120~5375	word	9400~94FF			
D	5376~5631	word	9500~95FF			
D	5632~5887	word	9600~96FF			
D	5888~6143	word	9700~97FF			
D	6144~6399	word	9800~98FF			
D	6400~6655	word	9900~99FF			
D	6656~6911	word	9A00~9AFF			
D	6912~7167	word	9B00~9BFF			
D	7168~7423	word	9C00~9CFF			
D	7424~7679	word	9D00~9DFF			
D	7680~7935	word	9E00~9EFF			
D	7936~8191	word	9F00~9FFF			
D	8192~8447	word	A000~A0FF			
D	8448~8703	word	A100~A1FF			
D	8704~8959	word	A200~A2FF			
D	8960~9215	word	A300~A3FF			
D	9216~9471	word	A400~A4FF			
D	9472~9727	word	A500~A5FF			
D	9728~9983	word	A600~A6FF			
D	9984~9999	word	A700~A70F			

## 1.12 오류 코드 정보

PLC 에 프로그램을 쓰고 난 후, PLC 오류 LED 가 깜빡이거나 특수 계전기 M1004=On 인 경우에는 프로그램이 유효하지 않은 것이었거나 문법상의 오류가 있었을 수 있습니다. 사용자는 저장된 오류 코드를 특수 레지스터 D1004 에서 볼 수 있고 다음의 표를 참고하여 오류 메시지와 D1137 에 저장된 오류 어드레스를 찾아볼 수 있습니다. (D1137 은 일반적인 루프 에러의 경우 유효하지 않습니다.)

오류 코드	설명	오류 코드	설명
0001	피연산자 비트 장치 S 가 사용 범위를 넘어섰습니다.	0E1A	구성요소가 사용 범위를 넘어섰습니다.(E, F 오류 포함)
0002	라벨 P 가 사용범위를 넘어섰거나 중복되었습니다.	0E1B	무리식이 음의 값입니다.
0003	피연산자 KnSm 가 사용범위를 넘었습니다.	0E1C	FROM/TO 통신 오류
0102	인터럽트 포인터 I 이 사용범위를 넘었거나 중복되었습니다.	0F04	피연산자 장치 D 레지스터 사용이 한계를 넘었습니다.
0202	MC 명령이 사용범위를 넘었습니다.	0F05	DCNT 오용(misuse) 피연산자 DXXX
0302	MCR 명령이 사용 범위를 넘었습니다.	0F06	SFTR 오용 피연산자
0401	피연산자 bit 장치 X 가 사용 범위를 넘었습니다.	0F07	SFTL 오용 피연산자
0403	피연산자 KnXm 가 사용 범위를 넘었습니다.	0F08	REF 오용 피연산자
0501	피연산자 bit 장치 Y 가 사용 범위를 넘었습니다.	0F09	WSFR 와 WSFL 오용 피연산자
0503	피연산자 KnYm 가 사용 범위를 넘었습니다.	0F0A	TTMR 와 STMR 가 사용범위를 넘었습니다.
0601	피연산자 bit 장치 T 가 사용 범위를 넘었습니다.	0F0B	명령 SORT 가 사용범위를 넘었습니다.
0604	피연산자 단어 장치 T 레지스터 사용이 한계를 넘었습니다.	0F0C	명령 TKY 사용 범위를 넘었습니다.
0801	피연산자 bit 장치 M 이 사용 범위를 넘었습니다.	0F0D	명령 HKY 사용 범위를 넘었습니다.
0803	피연산자 KnMm 이 사용 범위를 넘었습니다.	1000	ZRST 오용 피연산자
0D01	DECO 오용 피연산자	C400	확인되지 않은 명령 코드가 사용되었습니다.
0D02	ENCO 오용 피연산자	C401	루프 에러
0D03	DHSCS 오용 피연산자	C402	LD / LDI 가 9 번이상 연속으로 사용되었습니다.
0D04	DHSCR 오용 피연산자	C403	MPS 가 9 번이상 연속으로 사용되었습니다.
0D05	PLSY 오용 피연산자	C404	FOR-NEXT 가 6 수준을 넘었습니다.

0D06	PWM 오용 피연산자
0D07	FROM/TO 오용 피연산자
0D08	PID 오용 피연산자
0D09	SPD 오용 피연산자
0E01	피연산자 bit 장치 C 가 사용 범위를 넘었습니다.
0E04	피연산자 언어 장치 C 레지스터 사용이 한계를 넘었습니다.
0E05	DCNT 오용 피연산자 CXXX
0E18	BCD 변환 오류
0E19	나누기 오류(분모=0)

C405	STL/RET 가 FOR-NEXT 사이에 사용되었습니다. SRET/IRET 가 FOR-NEXT 사이에 사용되었습니다. MC/MCR 가 FOR-NEXT 사이에 사용되었습니다. END / FEND 가 FOR-NEXT 사이에 사용되었습니다.
C407	STL 가 9 번이상 연속으로 사용되었습니다.
C408	STL 에서 MC/MCR 명령을 사용했습니다. STL 에서 I/P 를 사용했습니다.
C409	서브루틴에서 STL/RET 을 사용했습니다. 인터럽트 프로그램에서 STL/RET 을 사용했습니다.

오류 코드	설명
C40A	서브루틴에서 MC/MCR 을 사용했습니다. 인터럽트 프로그램에서 MC/MCR 을 사용했습니다.
C40C	MC/MCR 에 해당되는 값 N 이 다릅니다.
C40D	IP 가 제대로 사용되지 않았습니다.
C40E	IRET 이 마지막 FEND 명령을 따르지 않았습니다. SRET 이 마지막 FEND 명령을 따르지 않았습니다.
C41C	확장 유닛 지점이 사용 범위를 넘었습니다.
C41D	특수 확장 모듈의 사용 범위를 넘었습니다.
C41E	특수 확장 모듈의 하드웨어 설정 오류
C41F	메모리에 쓰인 데이터 오류
C4FF	유효하지 않은 명령
C4EE	프로그램에 END 명령이 없습니다.

## 3.1 기본 명령어와 단계별 명령어 요약

## 📖 기본 명령어

명령 코드	기능	피연산자	실행속도(us)			단계	페이지
			ES	SA	EH		
LD	A 점점 논리 연산 시작	X, Y, M, S, T, C	5.6	4.6	0.24(0.56)	1~3	3-3
LDI	B 점점 논리 연산 시작	X, Y, M, S, T, C	5.68	4.68	0.24(0.56)	1~3	3-3
AND	A 점점 직렬 연결	X, Y, M, S, T, C	4.8	3.8	0.24(0.56)	1~3	3-3
ANI	B 점점 직렬 연결	X, Y, M, S, T, C	4.88	3.88	0.24(0.56)	1~3	3-4
OR	A 점점 병렬 연결	X, Y, M, S, T, C	4.8	3.8	0.24(0.56)	1~3	3-4
ORI	B 점점 병렬 연결	X, Y, M, S, T, C	4.88	3.88	0.24(0.56)	1~3	3-5
ANB	직렬 연결(병렬회로)	없음	4.4	3.4	0.24	1~3	3-5
ORB	병렬 연결(병렬회로)	없음	4.4	3.4	0.24	1~3	3-6
MPS	최근 내부 PLC 작동 저장	없음	4.64	3.64	0.24	1~3	3-6
MRD	최근 내부 PLC 작동 읽음	없음	4	3	0.24	1	3-6
MPP	최근 기억된 결과 불러오기와 제거하기	없음	4.4	3.4	0.24	1	3-6

## 📖 출력 명령어

명령코드	기능	피연산자	실행속도 (us)			단계	페이지
			ES	SA	EH		
OUT	출력 코일	Y, S, M	6.4	5.4	0.24(0.56)	1~3	3-7
SET	켜짐	Y, S, M	5.04	4.04	0.24(0.56)	1~3	3-7
RST	점점이나 레지스터 지움	Y, M, S, T, C, D, E, F	7.6	6.6	0.24(0.56)	3	3-8

## 📖 타이머, 계수기

API	명령 코드	기능	피연산자	실행 속도(us)			단계	페이지
				ES	SA	EH		
96	TMR	16 비트 타이머	T-K or T-D	9.6	8.6	25	4	3-8
97	CNT	16 비트 계수기	C-K or C-D (16 bits)	12.8	11.8	30	4	3-9
97	DCNT	32 비트 계수기	C-K or C-D (32 bits)	14.32	13.3	50	6	3-9

## 📖 주요 제어 명령어

명령 코드	기능	피연산자	실행 속도 (us)			단계	페이지
			ES	SA	EH		

MC	종합 제어 시작	N0~N7	5.6	4.6	5.6	3	3-10
MCR	종합 제어 리셋	N0~N7	5.7	4.7	5.7	3	3-1

📖 점점의 상승 에지(모서리)(모서리)/하강 에지(모서리)(모서리) 간과 명령어

API	명령 코드	기능	피연산자	실행 속도 (us)			단계	페이지
				ES	SA	EH		
90	LDP	상승 에지(모서리)(모서리) 간과 작동	S, X, Y, M, T, C	8.16	716	0.56(0.88)	3	3-11
91	LDF	하강 에지(모서리) 간과작동	S, X, Y, M, T, C	8.32	7.32	0.56(0.88)	3	3-12
92	ANDP	상승 에지(모서리) 직렬 연결	S, X, Y, M, T, C	7.68	6.68	0.56(0.88)	3	3-12
93	ANDF	하강 에지(모서리) 직렬 연결	S, X, Y, M, T, C	7.76	6.76	0.56(0.88)	3	3-12
94	ORP	상승 에지(모서리) 병렬 연결	S, X, Y, M, T, C	7.68	6.68	0.56(0.88)	3	3-13
95	ORF	하강 에지(모서리) 병렬 연결	S, X, Y, M, T, C	7.76	6.76	0.56(0.88)	3	3-13

📖 상승 에지(모서리)/하강 에지(모서리) 출력 명령

API	명령 코드	기능	피연산자	실행 속도 (us)			단계	페이지
				ES	SA	EH		
89	PLS	상승 에지(모서리) 출력	Y, M	9.92	8.92	9.92	3	3-13
99	PLF	하강 에지(모서리) 출력	Y, M	10.16	9.16	10.16	3	3-14

📖 종료 명령

명령 코드	기능	피연산자	실행 속도 (us)			단계	페이지
			ES	SA	EH		
END	프로그램 종료	없음	7.44	6.44	0.24	1	3-14

📖 Other 명령

API	명령 코드	기능	피연산자	실행 속도 (us)			단계	페이지
				ES	SA	EH		
	NOP	작동 안 함	없음	3.52	2.52	0.16	1	3-15
98	INV	작동 뒤바꿈	없음	3.92	2.92	0.24	1	3-15
	P	지시기	P0~P255	—	—	—	1	3-15

	I	프로그램 점검 가로채기	I□□□	-	-	-	1	3-16
--	---	--------------	------	---	---	---	---	------

### 계제식 명령

명령 코드	기능	피연산자	실행 속도 (us)			단계	페이지
			ES	SA	EH		
STL	계제식 변환 명령 시작	S	11.6	10.6	0.56	1	4-1
RET	계제식 변환 명령 복귀	없음	7.04	6.04	0.24	1	4-1

주의: 위의 모든 명령 테이블, ES 모델은 EX 모델과 SS 모델을 포함하고 SA 모델은 SX 모델과 SC 모델을 포함합니다.

주의: EH 직렬의 실행 속도 세로 칸에서 ()안에 적힌 값은 구체적으로 M1536~M4095 피연산자의 실행속도입니다.

## 3.2 기본 명령 설명


연산기호	기능	프로그램 단계	제어기						
LD	A 접점 실행	1	ES	EX	SS	SA	SX	SC	EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

LD 명령어는 왼쪽 버스에서 시작점을 가진 A 접점, 또는 접선회로가 A 접점으로부터 시작할 때 사용됩니다. 이 명령어의 기능은 현재 내용을 저장함과 동시에 누적레지스터 안에 취득 접점 상태를 저장하는 것입니다.

프로그램 예:

LD:	명령 코드:	작동:
	LD X0 ; X0의 A 접점 실행	
	AND X1 ; X1의 A 접점을 직렬로 연결	
	OUT Y1 ; Y1 코일 돌리기	


연산기호	기능	프로그램 단계	제어기						
LDI	B 접점 실행	1	ES	EX	SS	SA	SX	SC	EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

LDI 명령어는 왼쪽 버스에서 시작점을 가진 B 접점, 또는 접선회로가 B 접점으로부터 시작할 때 사용됩니다. 명령어의 기능은 현재 내용을 저장함과 동시에 누적레지스터 안에 취득 접점 상태를 저장하는 것입니다.

프로그램 예:

LD:	명령 코드:	작동:
	LDI X0 ; X0의 B 접점 실행	
	AND X1 ; X1의 A 접점을 직렬로 연결	
	OUT Y1 ; Y1 코일 돌리기	

연산기호	기능	프로그램 단계	제어기						
AND	A 접점 직렬 연결	1	ES	EX	SS	SA	SX	SC	EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

AND 명령어는 A 접점의 직렬 연결에서 사용됩니다. 이 명령어의 기능은 먼저 직렬 연결의 현재 특정 접점의 위치를 읽고, 접선 이전에 논리적 연산으로 AND 연산을 수행하며, 다음은 누적레지스터 안에 결과를 저장하는

것입니다.

프로그램 예:

LD:



명령 코드:

작동:

LDI X1 ;X1의 B 접점 실행

AND X0 ;X0의 A 접점을 직렬로 연결

OUT Y1 ;Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기
ANI	B 접점 직렬 연결	1	ES EX SS SA SX SC EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

ANI 명령어는 B 접점의 직렬 연결에서 사용됩니다. 이 명령어의 기능은 먼저 특정한 현재 직렬 연결 접선의 위치를 읽어내고, 논리적 연산 결과로 AND 연산을 수행하며, 다음은 누적레지스터 안에 결과를 저장하는 것입니다.

프로그램 예:

LD:



명령 코드:

작동:

LD X1 ;X1의 A 접점 실행

ANI X0 ;X0의 B 접점 직렬로 연결

OUT Y1 ;Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기
OR	A 접점 병렬 연결	1	ES EX SS SA SX SC EH

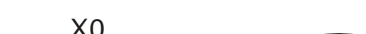
피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

OR 명령어는 A 접점의 병렬 연결에서 사용됩니다. 이 명령어의 기능은 현재 특정 직렬 연결 접선의 위치를 읽어내고, 접선 이전에 논리적 연산 결과로 OR 연산을 수행한 후, 누적레지스터 안에 결과를 저장하는 것입니다.

프로그램 예:

LD:



명령 코드:

작동:

LD X0 ;X0의 A 접점 실행



OR      X1      ; X1 의 A 접점 병렬로 연결  
OUT    Y1      ; Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기						
ORI	B 접점 병렬 연결	1	ES	EX	SS	SA	SX	SC	EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

ORI 명령어는 B 접점의 병렬연결에 사용됩니다. 이 명령어의 기능은 현재 특정 직렬연결 접점의 위치를 읽어내고, 논리적 연산결과로 OR 연산을 수행한 후, 누적레지스터 안에 결과를 저장하는 것입니다.

프로그램 예:

LD:



프로그램 코드:

작동:

LD      X1      ; A0 의 A 접점 실행  
ORI    X1      ; X1 의 B 접점 병렬로 연결  
OUT    Y1      ; Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기						
ANB	직렬 연결 (병렬회로)	1	ES	EX	SS	SA	SX	SC	EH

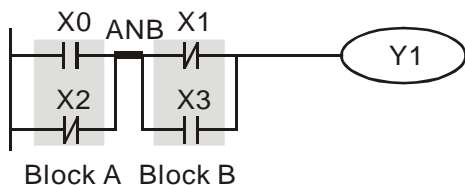
피연산자	없음
------	----

설명:

이전에 저장된 논리적 결과와 누적레지스터의 내용 사이에서 AND 연산을 실행하기 위해서입니다.

프로그램 예:

LD:



명령 코드:

작동:

LD      X0      ; X0 의 A 접점 실행  
ORI    X2      ; X2 의 B 접점 병렬로 연결  
LDI    X1      ; X1 의 B 접점 실행  
OR     X3      ; X3 의 A 접점 병렬로 연결  
ANB            ; 회로 블록 직렬로 연결  
OUT    Y1      ; Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기						
ORB	병렬 연결 (병렬회로)	1	ES	EX	SS	SA	SX	SC	EH

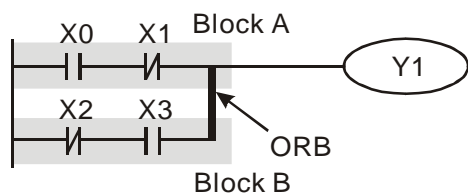
피연산자	없음
------	----

설명:

이전에 저장된 논리적 결과와 누적레지스터의 내용 사이에서 OR 연산을 실행하기 위해서입니다.

프로그램 예:

LD:



명령 코드:

LD X0 ; X0의 A 접점 실행  
 ANI X1 ; X1의 B 접점 직렬로 연결  
 LDI X2 ; X2의 B 접점 실행  
 AND X3 ; X3의 A 접점 직렬로 연결  
 ORB ; 회로 블록 병렬로 연결  
 OUT Y1 ; Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기
MPS	내부 PLC 작동의 최근 결과 기억	1	ES EX SS SA SX SC EH

피연산자	없음
------	----

설명:

누적 레지스터의 내용을 작동 결과에 저장하기 위해서입니다.(작동지시기의 결과는 1 펄스)

연산기호	기능	프로그램 단계	제어기
MRD	내부 PLC 작동의 최근 결과 읽음	1	ES EX SS SA SX SC EH

피연산자	없음
------	----

설명:

작동 결과의 내용을 누적 레지스터에 읽는 것입니다.(작동 결과의 지시기는 움직이지 변화없음)

연산기호	기능	프로그램 단계	제어기
MPP	최근 기억된 결과를 불러오기와 제거하기	1	ES EX SS SA SX SC EH

피연산자	없음
------	----

설명:

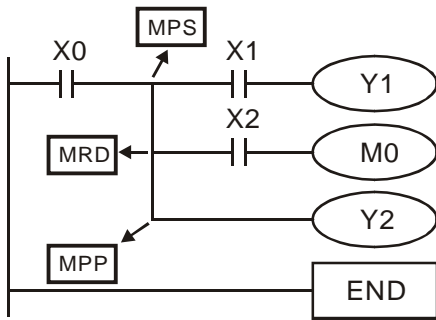
작동 결과로부터 이전에 저장된 논리적 연산 결과를 검색하고 누적 레지스터에 저장하기 위해서입니다.(결과 작동의 지시기는 -1 임)

프로그램 예:

LD:

명령 코드:

작동:



LD X0 ; X0의 A 접점 실행  
MPS ; 스택 저장  
AND X1 ; X1의 A 접점 직렬로 연결  
OUT Y1 ; Y1 코일 돌리기  
MRD ; 스택으로부터 읽기  
AND X2 ; X2의 A 접점 직렬로 연결  
OUT M0 ; M0 코일 돌리기  
MPP ; 스택으로부터 읽고 지시기 움직이기  
OUT Y2 ; Y2 코일 돌리기  
종료 ; 프로그램 종료

연산기호	기능	프로그램 단계	제어기						
OUT	코일 출력	1	ES	EX	SS	SA	SX	SC	EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	-	✓	✓	✓	-	-	-

설명:

구체적 장치에 대한 OUT 명령이 있기 전에 논리적 연산 결과를 출력합니다.

코일 접촉의 장치

작동 결과	출력 명령		
	코일	접촉	
		A 접촉(보통 열림)	B 접촉(보통 닫힘)
FALSE	꺼짐	비연속적	연속적
TRUE	켜짐	연속적	비연속적

프로그램 예:

LD: 명령 코드: 작동:

LDI X0 ; X0의 B 접점 실행  
AND X1 ; X1의 A 접점 직렬로 연결  
OUT Y1 ; Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기						
SET	걸침 (켜짐)	1	ES	EX	SS	SA	SX	SC	EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	-	✓	✓	✓	-	-	-

설명:

SET 명령이 돌아가면 구체적인 장치가 “켜짐” 상태가 되며, 이는 SET 명령이 돌아가는 아닌 “켜짐” 상태를 유지하게 됩니다. 장치를 “꺼짐”으로 조정하기 위해서는 RST 명령을 사용할 수 있습니다.

프로그램 예:

LD:



명령 코드:

작동:

LD X0 ; X0 의 A 접점 실행

ANI Y0 ; Y0 의 B 접점 직렬로 연결

SET Y1 ; Y1 걸침 (켜짐)

연산기호	기능	프로그램 단계	제어기						
RST	접촉 제거하기 또는 레지스터 제거하기	1	ES	EX	SS	SA	SX	SC	EH

피연산자	X0~X377	Y0~Y377	M0~M4,09 5	S0~S1,023	T0~T255	C0~C255	D0~D9,99 9	E, F
	—	✓	✓	✓	✓	✓	✓	✓

설명:

RST 명령이 돌아갈 때, 구체적인 장치의 상태는 다음과 같습니다.

장치	상태
S, Y, M	코일과 접촉은 “꺼짐”상태가 됩니다.
T, C	타이머나 계수기의 현 가치는 0 이 되고, 코일과 접촉은 “꺼짐” 상태가 됩니다.
D, E, F	내용가치는 0 이 됩니다.

프로그램 예:

LD:



명령 코드:

작동:

LD X0 X0 의 A 접점 실행

RST Y5 Y5 접촉 지우기

연산기호	기능	프로그램 단계	제어기						
TMR	16 비트 타이머	1	ES	EX	SS	SA	SX	SC	EH


피연산자	T-K	T0~T255, K0~K32,767
	T-D	T0~T255, D0~D9,999

설명:

TMR 명령이 실행될 때, 타이머의 구체적인 코일은 켜짐 상태가 되고 타이머는 숫자를 세기 시작할 것입니다. 타이머의 세팅 가치가(집계 가치>=세팅 가치) 달성이 되면 접촉은 다음과 같을 것입니다.

NO(보통 열림) 접촉	콜렉터 열기
NC(보통 닫힘) 접촉	콜렉터 닫기

프로그램 예:

LD:	명령 코드:	작동:
	LD X0 TMR T5 K1000 K1000 로 고정	X0 T6 타이머의 A 접점 실행

주의:  
피연산자 T 사용법을 위해 모든 모형의 설명서를 참조하십시오.

연산기호	기능	프로그램 단계	제어기							
			ES	EX	SS	SA	SX	SC	EH	
CNT	16 비트 타이머	1								

피연산자	C-K	C0~C199, K0~K32,767
	C-D	C0~C199, D0~D9,999

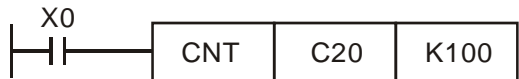
설명:

- CNT 명령이 꺼짐->켜짐이 실행될 때 카운터 코일이 돌아감을 의미하므로 1은 카운터의 가치에 더해지게 됩니다.; 카운터가 특정한 가치(카운터의 가치=세팅 가치)를 성취하였을 때, 접촉의 모양은 다음과 같습니다.

NO(보통 열림)접촉	연속적
NC(보통 닫힘)접촉	비연속적

- 집계가 달성된 후 집계펄스 입력이 있습니다.면, 접촉과 집계가치는 바뀌지 않을 것입니다. 다시 세거나 CLEAR 장치를 실행하기 위해서는 RST 명령을 이용하십시오.

프로그램 예:

LD:	명령 코드:	작동:
	LD X0 CNT C20 K100 K100 고정	X0 C20 카운터의 A 접점 실행

연산기호	기능	프로그램 단계	제어기							
			ES	EX	SS	SA	SX	SC	EH	
DCNT	32 비트 카운터	1								

피연산자	C-K	C200~C254, K-2,147,483,648~K2,147,483,647
	C-D	C200~C254, D0~D9,999


설명:

- DCNT는 카운터 C200과 C254에서 활용되는 32비트 하이 스피드 카운터를 위한 시동입니다.
- 카운터 C200~C234의 일반적 덧셈과 뺄셈을 위해서 현재 가치는 DCNT 명령이 꺼짐->켜짐으로 될 때, 올라가거나 (1을 더해) 내려갈 것입니다.(1을 빼).
- 고속도 덧셈/뺄셈 카운터 C235~C254가 꺼짐->켜짐이 될 때의 특정한 고속도 카운터 펄스 입력은 숫자 세기를 실행할 것입니다. 카운터 트리거 입력이 “켜짐” 또는 “꺼짐” 상태를 유지합니다.면, 카운터 가치는 변하지 않을 것입니다. 2.7 장을 보면 타이머 숫자와 고속도 펄스 입력 터미널(X0~X17)의 기능은 (1을

더해)올라가고 (1 을 빼)내려갈 것입니다.

4. DCNT 명령이 “꺼짐” 상태일 때, 카운터는 숫자세기를 멈추지만 카운트 가치는 지워지지 않을 것입니다. 사용자들은 카운트 가치와 접촉을 제거하기 위해 RST C2XX 명령을 사용할 수 있다. 고속 덧셈/뺄셈 카운터 C235~C254 는 카운트 가치와 접촉을 제거하기 위해 외부 특정 입력을 사용할 수 있습니다.

프로그램 예:

LD:	명령 코드:	작동:
	LD M0 DCNT C254 K1000	M0 의 A 접점과 C254 카운터 실행 K1000 고정

연산기호	기능	프로그램 단계	제어기
MC / MCR	종합 제어 시작/재시작	1	ES EX SS SA SX SC EH

피연산자	N0~N7
------	-------

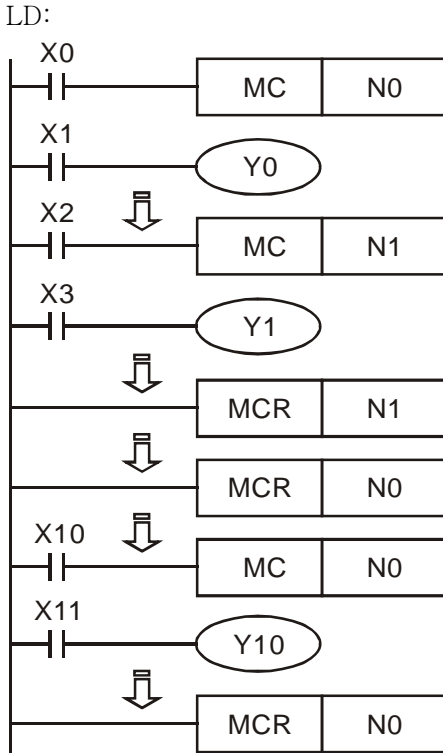
설명:

1. MC 는 주요-제어 시작 명령입니다. MC 명령이 실행될 때, MC 와 MCR 사이 명령의 실행은 방해 받지 않을 것입니다. MC 명령이 꺼졌을 때, MC 와 MCR 사이 명령의 상태는 다음과 같이 설명됩니다.:

타이머	카운트 가치는 0 으로 돌아가며, 코일과 접촉은 모두 꺼진다.
누적 타이머	코일은 꺼지며 타이머 가치와 접촉은 현재 상태에 머무른다.
서브루틴 타이머	카운트 가치는 0 으로 돌아간다. 코일과 접촉은 모두 꺼진다.
계수기	코일은 꺼지며, 카운트 가치와 접촉은 현재 상태에 머무른다.
OUT 명령에 의해 돌아가는 코일	모두 꺼짐
SET 과 RST 명령에 의해 돌아가는 장치	현재 상태에 머무른다.
응용명령	모두 작동되지 않지만 사용자에게 의해 정의된 회로 FOR-NEXT 명령은 MC-MCR 명령이 “꺼짐”상태에도 불구하고 여러 번 실행될 것입니다.

2. MCR 은 주요-제어 종료 명령이며 주요-제어 프로그램의 끝에 위치하고 MCR 명령 이전에 어떠한 접촉 명령도 있어서는 안됩니다.
3. MC-MCR 주요-제어 프로그램의 명령은 최대 8 개의 층으로 동 프로그램 구조를 지원합니다. N0~N7 순으로 명령을 사용하고 다음을 참조하라:

프로그램 예:



명령 코드:	작동:
LD X0	X0의 A 접점 실행
MC N0	N0 일반 직렬 연결 접촉 가능
LD X1	X1의 A 접점 실행
OUT Y0	Y0 코일 돌리기
:	
LD X2	X2의 A 접점 실행
MC N1	N1 일반 직렬 연결 접촉 가능
LD X3	X3의 접점 실행
OUT Y1	Y1 코일 돌리기
:	
MCR N1	N1 일반 직렬 연결 접촉 불가능
:	
MCR N0	N0 일반 직렬 연결 접촉 불가능
:	
LD X10	X10의 A 접점 실행
MC N0	N0 일반 직렬 연결 접촉 가능
LD X11	X11의 A 접점 실행
OUT Y10	Y10 코일 돌리기
:	
MCR N0	N 보통 직렬 연결 접촉 불가능

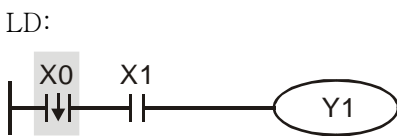
연산기호	기능	프로그램 단계	제어기
LDP	상승 에지(모서리) 간파작동	1	ES EX SS SA SX SC EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

LPD 명령의 사용은 LD 명령과 같지만, 모양은 다르다. 그것은 현재 내용을 유지함과 동시에 접촉 상승 에지(모서리)의 탐지 위치를 누적 레지스터에 저장하기도 합니다.

프로그램 예:



명령 코드:	작동:
LDP X0	Start X0 상승 에지(모서리) detection
AND X1	직렬 연결 A contact of X1
OUT Y1	Y1 코일 돌리기

Remarks:

1. 피연산자 사용을 위해 각각 모형의 설명서를 참조하십시오.
2. PLC가 켜지기 이전에 특정 상승 에지(모서리) 접촉 상태가 “켜짐” 상태일 경우 상승 에지(모서리) 접촉은 PLC가 켜진 후 True 상태가 될 것입니다.

연산기호	기능	프로그램 단계	제어기
------	----	---------	-----

LDF	하강 에지(모서리) 간파작동	1	ES	EX	SS	SA	SX	SC	EH
-----	-----------------	---	----	----	----	----	----	----	----

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

LDP 명령의 사용은 LD 명령과 같지만, 모양은 다르다. 그것은 현재 내용을 유지함과 동시에 접촉 하강 에지(모서리)의 탐지 위치를 누적 레지스터에 저장하기도 합니다.

프로그램 예:

LD:



명령 코드:

작동:

LDF	X0	X0 하강 에지(모서리) 탐지 시작
AND	X1	직렬 연결 A contact of X1
OUT	Y1	Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기
ANDP	상승 에지(모서리) 직렬 연결	1	ES EX SS SA SX SC EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

ANDP 명령은 접촉의 상승 에지(모서리) 탐지의 직렬연결에서 사용됩니다.

프로그램 예:

LD:



명령 코드:

작동:

LD	X0	X0의 A 접점 실행
ANDP	X1	X1 상승 에지(모서리) 간파 직렬 연결
OUT	Y1	Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기
ANDF	하강 에지(모서리) 직렬 연결	1	ES EX SS SA SX SC EH

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

ANDF 명령은 접촉의 하강 에지(모서리) 탐지의 직렬연결에서 사용됩니다.

프로그램 예:

LD:



명령 코드:

작동:

LD	X0	X0의 A 접점 실행
ANDF	X1	X1 하강 에지(모서리) 간파 직렬 연결



OUT      Y1      Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기						
			ES	EX	SS	SA	SX	SC	EH
ORP	상승 에지(모서리) 병렬 연결	1							

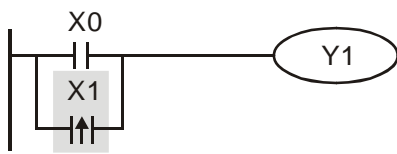
피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

ORP 명령은 접촉의 상승 에지(모서리) 탐지의 병렬연결에서 사용됩니다.

프로그램 예:

LD:



명령 코드:

LD      X0

ORP    X1

OUT    Y1

작동:

X0의 A 접점 실행

X1 상승 에지(모서리) 간파 병렬 연결

Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기						
			ES	EX	SS	SA	SX	SC	EH
ORF	하강 에지(모서리) 병렬 연결	1							

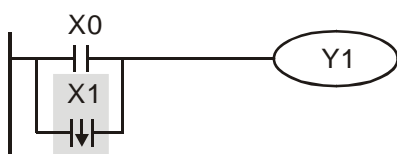
피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	—

설명:

ORP 명령은 접촉의 하강 에지(모서리) 탐지의 병렬 연결에서 사용됩니다.

프로그램 예:

LD:



명령 코드:

LD      X0

ORF    X1

OUT    Y1

작동:

X0의 A 접점 실행

X1 하강 에지(모서리) 간파 병렬 연결

Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기						
			ES	EX	SS	SA	SX	SC	EH
PLS	상승 에지(모서리) 출력	1							

피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	—	✓	✓	—	—	—	—

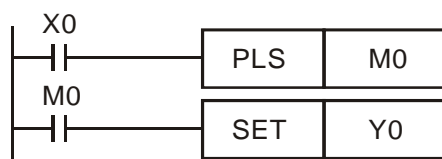
설명:

X=0 이 꺼짐->켜짐(상승 에지(모서리) 트리거)일 때, PLS 명령이 실행될 것이며 M0은 길이가 주사할 시간의

펄스를 종료시킬 것입니다.

프로그램 예:

LD:



명령 코드:

LD X0

작동:

X0의 A 접촉실행

PLS M0

M0 상승 에지(모서리) 출력

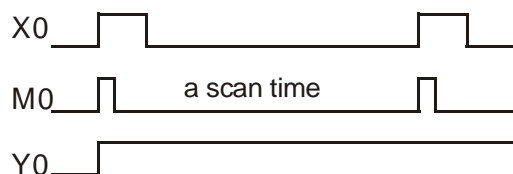
LD M0

M0의 A 접점 실행

SET Y0

Y0 걸침 (ON)

Timing Diagram:



연산기호	기능	프로그램 단계	제어기
PLF	하강 에지(모서리) 출력	1	ES EX SS SA SX SC EH

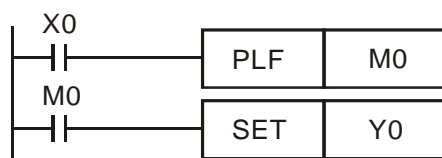
피연산자	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	—	✓	✓	—	—	—	—

설명:

X0=켜짐->꺼짐(하강 에지(모서리) 트리거)일 때, PLF 명령이 실행될 것이며 M0은 길이가 주사시킬 길이인 시간에 펄스를 전송시킬 것입니다.

프로그램 예:

LD:



명령 코드:

LD X0

작동:

X0의 A 접점 실행

PLF M0

M0 하강 에지(모서리) 출력

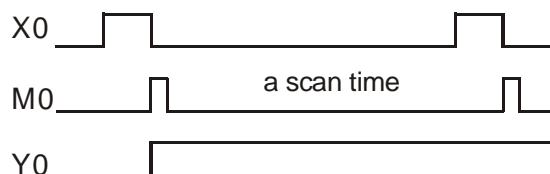
LD M0

M0의 A 접점 실행

SET Y0

Y0 걸침(켜짐)

Timing Diagram:



연산기호	기능	프로그램 단계	제어기
종료	프로그램 종료	1	ES EX SS SA SX SC EH

피연산자	없음
------	----

설명:

계제식 프로그램이나 명령 프로그램의 끝에 종료 명령을 더할 필요가 있습니다. PLC 는 번지로부터 종료 명령에까지 주사할 것이며, 이와 같은 실행 후에 다시 주사하기 위해 0 번지로 되돌아 올 것입니다.

연산기호	기능	프로그램 단계	제어기
NOP	No 작동	1	ES EX SS SA SX SC EH

피연산자	없음
------	----

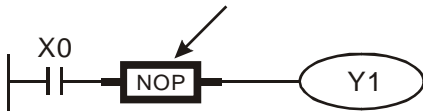
설명:

이것은 “작동 안 함” 명령이며 이전의 작동에는 영향을 미치지 않는다. NOP 은 다음과 같은 경우에 쓰인다:단계의 번호 변경 없이 명령을 지울 때(NOP 으로 다시 쓰기)

프로그램 예:

LD:	명령 코드:	작동:
	LD X0	X0 의 B 접점 실행
	NOP	작동 안 함
	OUT Y1	Y1 코일 돌리기

Command NOP will be omitted when ladder diagram displays.



연산기호	기능	프로그램 단계	제어기
INV	전화 작동	1	ES EX SS SA SX SC EH

피연산자	없음
------	----

설명:

작동 결과를 전화하고 작동 결과로서의 새로운 데이터를 사용합니다.

프로그램 예:

LD:	명령 코드:	작동:
	LD X0	X0 의 A 접점 실행
	INV	작동 결과 전화
	OUT Y1	Y1 코일 돌리기



연산기호	기능	프로그램 단계	제어기
P	지시기	1	ES EX SS SA SX SC EH

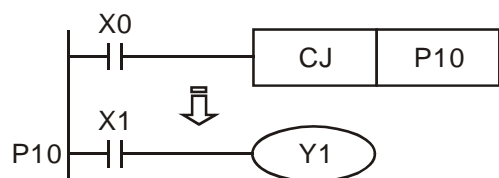
피연산자	P0~P255
------	---------

설명:

지시기 P는 다음과 같이 두 가지 방법에서 건너뛰기 명령(API 00 CJ, API 01 CALL)과 함께 사용됩니다. 굳이 숫자 0부터 시작하지 않아도 되지만, 숫자는 반복되지 못하고 예상치 못한 오류가 발생할 수 있습니다.

프로그램 예:

LD:



명령 코드:

LD X0  
CJ P10  
:

작동:

X0의 A 접점 실행  
명령 CJ부터 P10로 뛰어넘기

P10 지시기 P10

LD X1 X1의 A 접점 실행

OUT Y1 Y1 코일 돌리기

연산기호	기능	프로그램 단계	제어기						
I	프로그램 마커 가로채기	1	ES	EX	SS	SA	SX	SC	EH

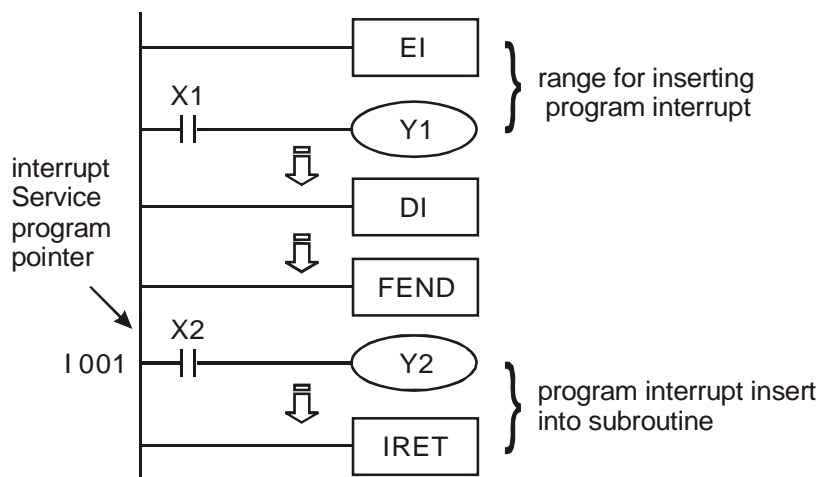
피연산자	I00□, I10□, I20□, I30□, I40□, I50□, I6□□, I7□□, I8□□ I010, I020, I030, I040, I050, I060, I110, I120, I130, I140, I150, I160, I170, I180
------	--

설명:

프로그램 가로채기는 포인터 가로채기부터 시작해 가로채기 끝과 되돌리기의 응용명령으로 끝이 납니다. 이는 반드시 응용명령 API 03 IRET, API 04 EI, API 05 DI를 사용해야 합니다. 각각의 직렬을 위한 가로채기 지시기를 고려해 2.9 장을 참조합니다.

프로그램 예:

LD:



명령 코드:

EI 가로채기 가능  
LD X1 X1의 A 접점 실행  
OUT Y1 Y1 코일 돌리기  
:  
DI 가로채기 불가능  
:  
FEND 프로그램 종료  
I001 접점 삽입  
LD X2 X2의 A 접점 실행  
OUT Y2 Y2 코일 돌리기  
:  
IRET 가로채기와 되돌리기

## 4.1 Step Ladder 명령 [STL], [RET]

연상기호	기능	프로그램 스텝	제어기						
STL	Step Transition Ladder 시작 명령	1	ES	EX	SS	SA	SX	SC	EH

피연산자	S0~S1023
------	----------

설명:

Step Ladder 명령인 STL Sn 은 스텝별 포인트로 구성되어 있고, STL 명령이 프로그램에 보여졌을 때는 프로그램이 현재 스텝별 과정에 의해 제어되는 step ladder 다이어그램 상태에 있다는 것을 의미합니다. Step Ladder 명령인 RET 은 BUS 명령으로 되돌아왔다는 step ladder 다이어그램(from S0~S9)의 끝부분을 의미합니다. SFC 다이어그램은 STL/RET 로 구성되어 있는 step ladder 다이어그램을 통해 나타내어집니다. 스텝 포인트 S 는 반복될 수 없습니다.

연상기호	기능	프로그램 스텝	제어기						
RET	Step Transition Ladder 복귀 명령	1	ES	EX	SS	SA	SX	SC	EH

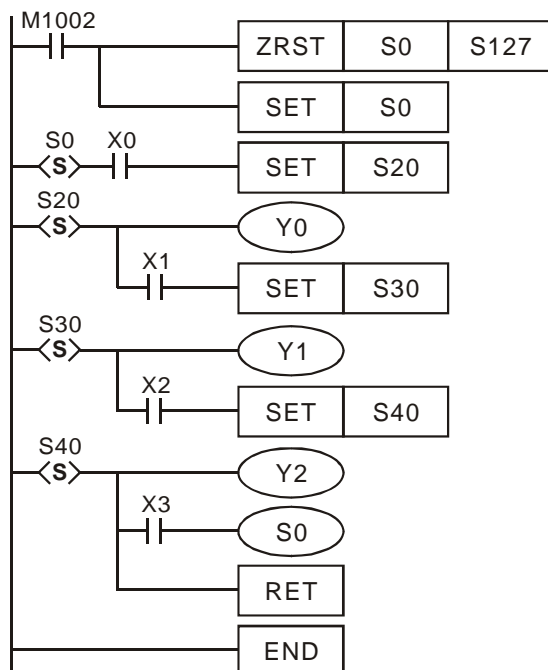
피연산자	없음
------	----

설명:

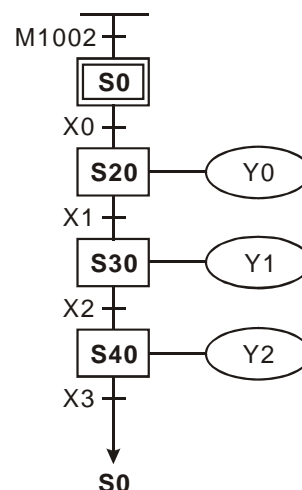
스텝별 과정의 끝부분에 반드시 RET 명령을 써야 합니다. ; RET 명령은 스텝별 과정의 끝부분을 가리킵니다. PLC 프로그램은 최대 10스텝(S0~S9)의 과정이며, 각 스텝별 과정의 끝부분에는 RET 명령을 사용할 것입니다.

프로그램 예제:

Ladder 다이어그램:



SFC:



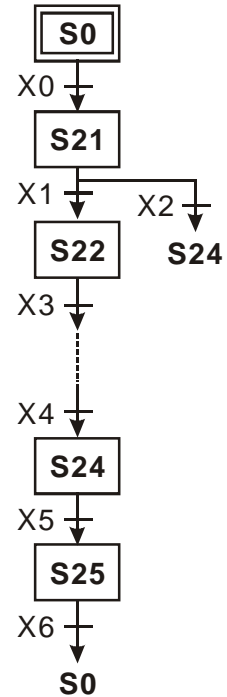
## 4.2 연속 기능 도표 (SFC)

자동제어 분야에서는 종종 목표에 도달하기 위해 전기 및 기계적 제어가 동시에 작용할 필요가 있습니다. SFC는 몇 개의 연속된 스텝(i.e. 몇 가지 양상)로 나누어질 수 있습니다. 각각의 스텝은 그 스텝의 작용으로 끝날 것이고, 보통 각 스텝에서 다음 스텝까지 이동하는 변화가 이루어진다. 그것은 전 스텝의 작용의 끝부분과 다음 스텝의 시작부분으로 연결되는 연속기능 도표의 디자인 구성개념입니다(이전 스텝은 이때 사라질 것입니다).

특징:

1. 불변상태 스텝에서는 연속기능 도표를 구상할 필요는 없습니다. PLC는 각 상태 사이에서 연동작용과 두배의 출력을 실행할 것입니다. 각각의 상태를 간단한 SFC 설계를 할 필요가 있고, 기계작업을 만들 필요가 있습니다.
2. 그 작용은 초기 PLC 작동이나 탐지와 유지보수를 조정하고 이해하기가 쉽습니다.
3. SFC 편집이론은 IEC1131-3에 의해 이루어집니다. 편집모드와 구조는 흐름도와 비슷합니다. 각 PLC 내부의 스텝별 계전기 S는 스텝별 포인트로 사용되기도 하고, 또한 흐름도의 각 스텝과 동일하게도 사용되기도 합니다. 현 스텝이 끝난 후에는 지정된 조건에 의해 다음 스텝 포인트인 S로 이동될 것입니다. 이 방법이 반복됨에 따라 사용자가 원하는 결과에 도달할 수 있을 것입니다.
4. 오른쪽 SFC 그림 설명: 각각의 스텝은 한 스텝에서 다음 스텝으로 움직이기 위한 자신만의 변환조건을 가지고 있습니다. 이 그림에서는 주 스텝포인트 S0는 이 변환조건 X0가 입증되는 즉시 스텝포인트 S21로 이동할 것이고, S21은 변환조건 X1이나 X2에 의해 S22나 S24로 이동할 수 있습니다. 그리고 S25는 변환조건 X6이 입증되면 전체과정이 끝나는 S0로 이동할 것입니다. 이 방법으로 반복되는 제어를 계산할 수 있습니다.
5. 다음은 WPLSoft SFC 다이어그램 편집용 아이콘 소개입니다.

SFC:

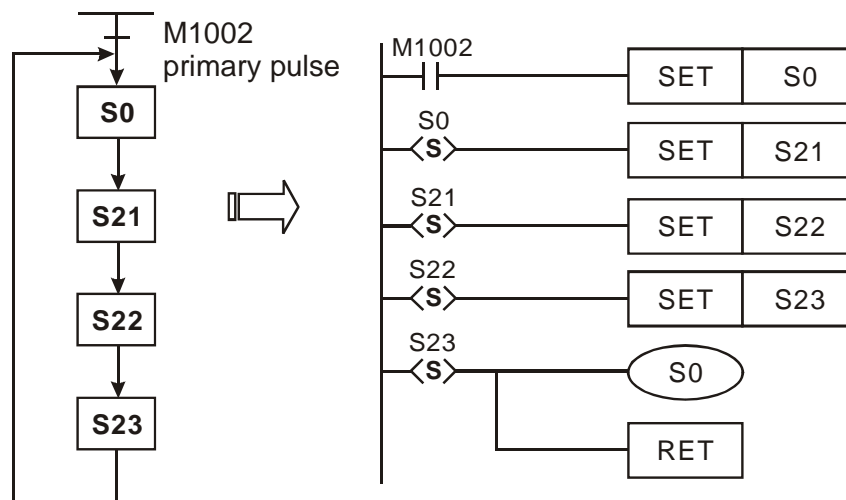


	Ladder step 모드로 사용됩니다. 이 계산은 주요 내부 프로그램이 스텝 ladder 프로그램이 아니라 일반적인 스텝 ladder 프로그램을 의미합니다.
	주 스텝 포인트를 뜻합니다. 이 두 겹의 구조는 SFC 주 스텝 포인트에 사용되고, 사용장치는 S0~S9입니다.
	일반적 스텝 포인트에 사용되고, 사용장치는 S10~S1023입니다.
	스텝포인트에서 다음이 없는 다른 스텝포인트로 이동하는 데 사용되는 JUMP 스텝포인트입니다. (같은 프로그램 과정에서 주 스텝포인트로 전환되거나 다른 프로그램 과정 사이에 점프하여 위로 올라가거나 아래로 내려가서 분리되는 데 사용될 수 있습니다.)
	각 스텝 포인트 사이에 움직이면서 사용되는 스텝 포인트의 변화상태입니다.
	선택적 분산은 다른 변환 조건에 의해 서로 다른 통신 스텝포인트로 움직이는 스텝포인트로 사용됩니다.
	동시적 집합은 변환조건에 따른 같은 스텝포인트로 움직이고, 두 스텝포인트로 사용됩니다.
	동시적 분산은 같은 변환조건에 의해 하나의 스텝포인트를 두 개의 스텝포인트로 움직이게 하는데 사용됩니다.
	어떤 조건이 동시에 수립될 때 같은 변환 조건에서 두 스텝포인트와 같은 스텝포인트로 이동하는 데 사용되는 동시적 집합입니다.

### 4.3 Step Ladder 명령 설명

**STL 명령:** 이 명령은 연속기능 도표에 체계적인 구성으로 사용됩니다. 이 명령은 연속기능 도표를 체계적으로 구성하는데 사용됩니다. 이 명령은 프로그램 과정상에 더 명백한 아이디어를 갖도록 프로그램 설계자를 도와주고, 그래서 그 과정은 좀 더 쉬워질 것입니다. 다음 다이어그램에서 보듯이 우리는 왼쪽 다이어그램을 오른쪽 PLC 구조 다이어그램으로 우리의 과정 다이어그램을 변경할 수 있습니다. 스텝과정 끝부분에서는 반드시 RET 명령을 써야 합니다. RET 명령은 스텝과정의 끝부분을 의미합니다. 스텝과정의 끝부분에 반드시 RET 명령을 쓰게 하여 몇몇 스텝과정은 같은 프로그램에 쓰여질 수 있습니다. RET 명령의 사용에 제한은 없고, 이 명령은 스텝 포인트(S0~S9)의 사용과 비교될 수 있습니다.

만약 RET 명령은 스텝 과정의 끝부분에 쓰여질 수 없습니다.면 이 오류는 편집장치에 의해 감지될 것입니다.

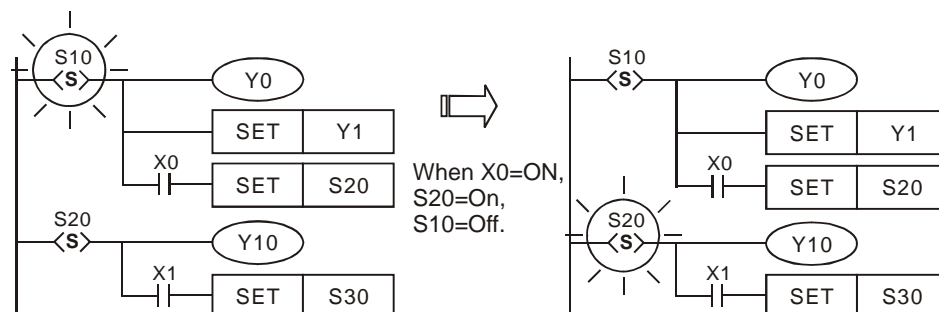


#### 1. Step Ladder 작용 :

Step Ladder 는 수많은 스텝 포인트로 구성되어 있습니다. ; 각 스텝 포인트는 하나의 제어과정 작용을 나타내고, 다음의 세가지 임무를 실행할 필요가 있습니다.:

- 구동 출력 코일
- 특수변환조건
- 스텝 포인트가 현재 스텝 포인트의 컨트롤 파워를 인계하도록 지정되어 있는지를 명시한다.

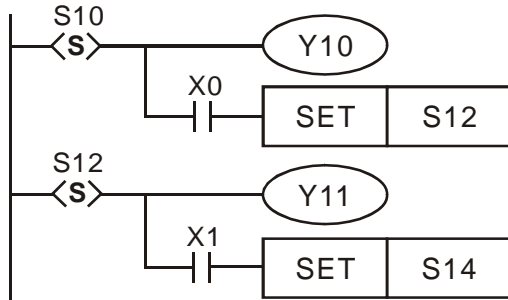
예제 :



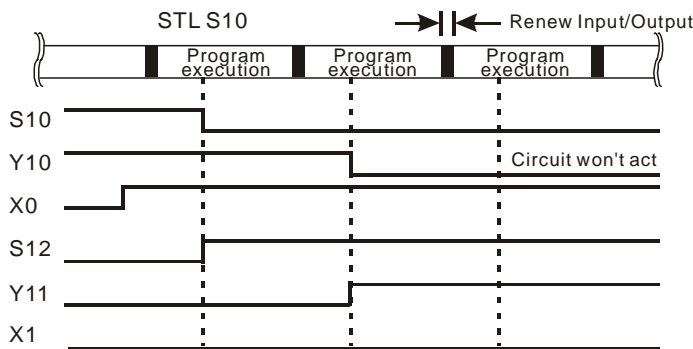
설명 :

S10=ON 일 때, Y0 와 Y1 는 켜진다. X0=ON 일 때, S20=ON 와 Y10 도 또한 켜진다. 그리고 S10 이 꺼지면 Y0 는 꺼지지만 Y1 는 켜진다. (Y1 을 SET 명령으로 사용한 후에는, ON 상태로 유지할 것입니다.)

2. Step ladder 타이밍:상태 접촉 Sn 이 켜질 때, 회로는 작동될 것이고, 회로는 상태접촉 Sn 이 꺼질 때는 작동되지 않을 것입니다. (작동에 앞서 스캔시간을 늦춘 수 실행될 것입니다.)

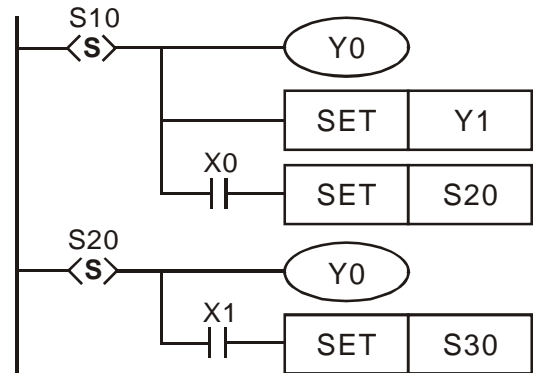


다음과 같은 유사한 순차적 기능표의 예로 왼쪽 모형을 보면, 상태접촉 S10 이 S12 로 이동될 때 같은 시간에 S12 가 켜지는 반면 S10 은 꺼질 것이지만 1 스캔 주기의 지연은 변환과정에서 발생될 것입니다. (중복 없이)



3. 출력코일의 반복 사용 :

- 같은 수의 출력코일은 서로 다른 스텝포인트에 사용될 수 있습니다.
- 오른쪽 표와 같이 서로 다른 상태에서 같은 출력장치 Y0 가 있습니다. S10 이나 S20 이 켜질지라도 Y0 는 켜질 것입니다.
- Y0 는 S10 에서 S20 으로 변환되는 동안 닫힐 것이고, S20 이 켜진 후에 Y0 가 출력됩니다. 그래서 이런 경우에는 Y0 는 S10 이나 S20 이 켜질지라도 Y0 는 켜질 것입니다.
- 일반적 ladder 표에서는 출력코일의 반복사용은 피해야 될 것입니다. 출력코일 수는 일반적 ladder 다이어그램으로 전환후 사용되는 피해야 할 스텝 포인트로 사용됩니다.

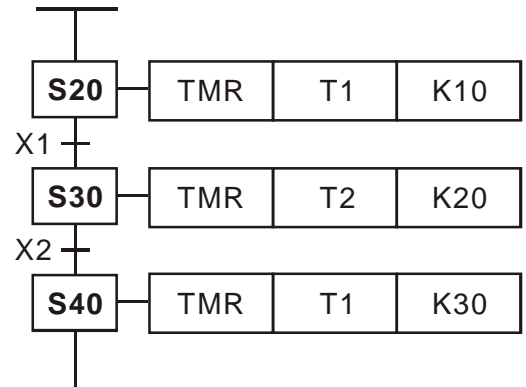


4. 타이머의 반복 사용 :



일반적 출력 코일과 같은 타이머는 서로 다른 스텝 포인트에 반복적으로 사용될 수 있습니다. (이는 스텝 ladder 다이어그램의 한 형상이지만, 출력코일의 반복사용은 피해야 할 것입니다. 출력코일 수는 일반적 ladder 다이어그램으로 전환 후 피해야 할 스텝 포인트로 사용될 것입니다.)

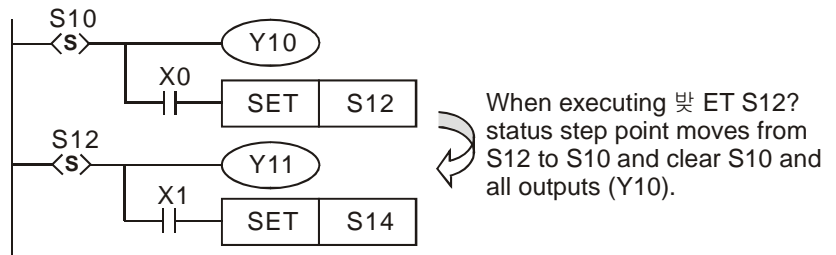
노트:오른쪽 표와 같이, ES/EX/EP/SS/SA/SX/SC 시리즈 타이머는 분리된 스텝 포인트에만 반복적으로 사용됩니다.



## 5. 스텝 포인트의 이동:

SET Sn 과 OUT Sn 명령은 또 다른 스텝포인트를 시작하는데 둘 다 사용되고, 이러한 명령어가 사용되는 경우는 서로 다를 수 있습니다.:통제전력이 또 다른 스텝포인트로 이동되었을 때 원래 스텝포인트 S와 출력 작용은 모두 지워질 것입니다. 다수의 스텝 제어 과정이 같은 프로그램에 동시에 존재할 수 있게 하기 위해 서는(스텝 ladder 다이어그램을 유도하기 위해 시작점과 종점인 S0~S9를 취함) 스텝의 이동이 같은 스텝과정 이 되거나 서로 다른 스텝과정으로 이동될 수 있습니다. 그래서, 이동명령인 스텝포인트의 SET Sn 과 OUT Sn 은 다양하게 사용되어질 것입니다 : 아래의 설명을 참조하세요 :

**SET Sn** 같은 과정에서는 다음 상태 스텝 포인트로 움직이는데 사용되고, 상태가 이동된 후에는 이전 작용의 출력상태 포인트는 사라질 것입니다.

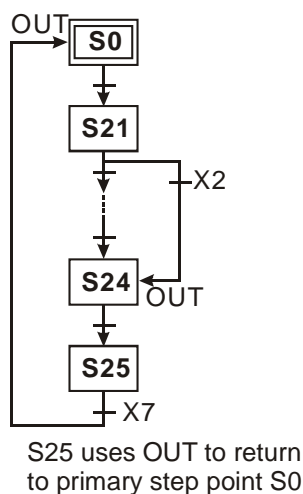


**OUT Sn** 같은 과정에서는 동시 집합점과 다른 과정의 이동은 구분되는 스텝 포인트를 구동하는 데 사용되고, 상태가 이동된 후에는 이전 작용 상태 포인트의 출력량은 사라질 것입니다.

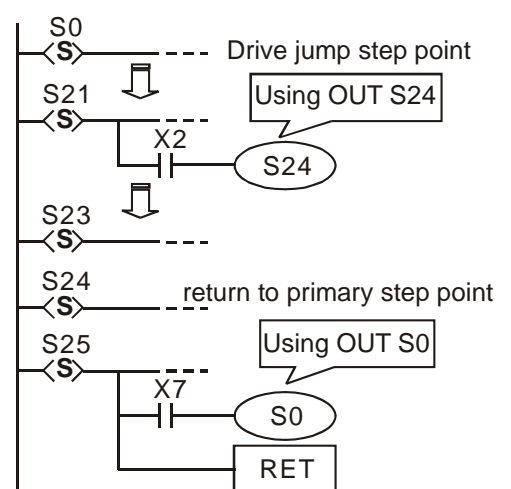
① 같은 과정에서는 주 스텝포인트로 전환하는데 사용됩니다.

② 같은 과정에서는 분리된 스텝포인트 사이에 올라가거나 내려가는 스텝포인트로 사용됩니다.

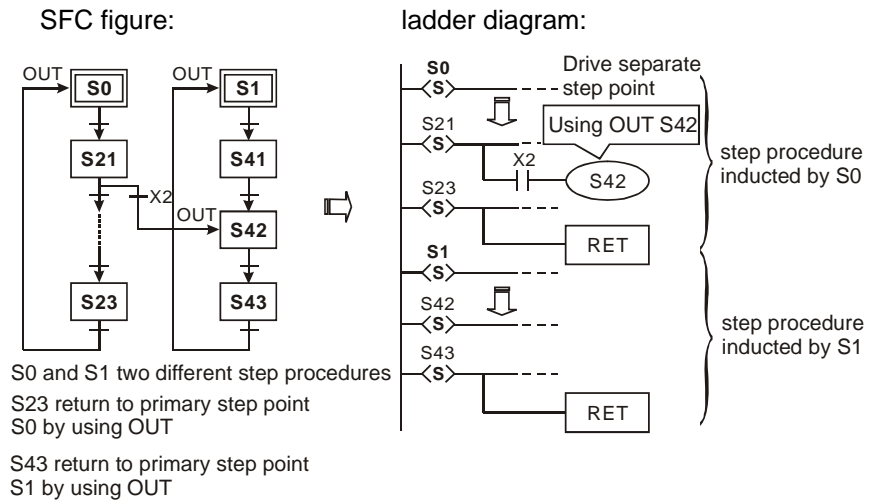
SFC diagram:



Ladder diagram:

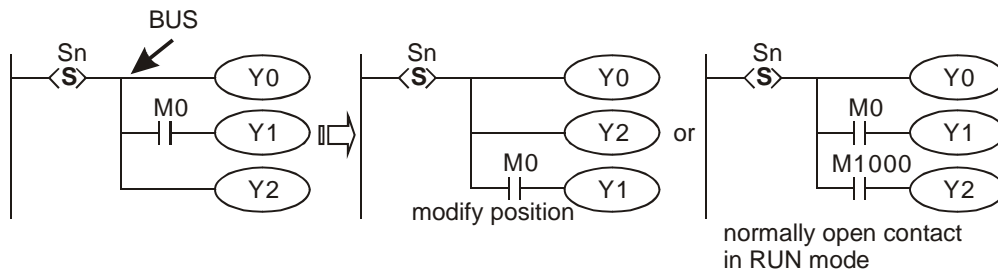


③ 다른 과정에서 분리된 스텝포인트를 구동하는데 사용됩니다.



## 6. 출력 포인트 구동의 주의점:

다음의 왼쪽 다이어그램과 같이 LD or LDI 명령이 스텝포인트 너머에 BUS의 두 번째 라인에 쓰여진 후에는 출력코일은 BUS로부터 직접적으로 연결할 수 없습니다. 쓰여지자마자 에러가 발생할 것입니다. 올바른 다이어그램으로 다음의 중간과 왼쪽 다이어그램으로 수정할 필요가 있습니다.



## 7. 부분적 명령 제한 사용 :

모든 스텝포인트의 프로그램은 일반적인 ladder 프로그램과 동일하고, 모든 시리즈 종류와 병렬연결 회로나 특정명령은 이용될 수 있지만, 명령의 일부분은 어떤 제한 하에 있습니다. 다음 설명을 참조하세요

스텝포인트에 사용된 기초 명령

기초 명령		LD/LDI/LDP/LDF AND/ANI/ANDP/ANDF OR/ORI/ORP/ORF INV/OUT/SET/RST	ANB/ORB MPS/MRD/MPP	MC/MCR
스텝포인트				
주 스텝 포인트/ 일반 스텝 포인트		Yes	Yes	No
분기 스텝 포인트/ 수렴 스텝 포인트	일반 출력	Yes	Yes	No
	스텝 포인트 이송	Yes	No	No

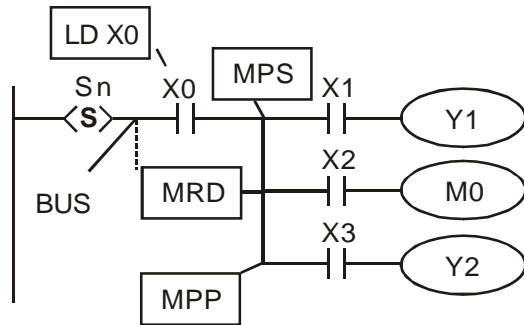
※ MC/MCR 명령은 스텝포인트에서 사용되지 않습니다.

※ The STL 명령은 일반 sub-프로그램과 서비스 중단 sub-프로그램에 사용될 수 없습니다.

※ CJ 명령의 사용은 STL 명령에서 금지되어 있지는 않지만, 작용을 복잡하게 할 것이고, 그래서 회피되어야 할 것입니다.

※ MPS/MRD/MPP 명령 위치 :

스텝 Ladder 다이어그램 :



명령 기호:

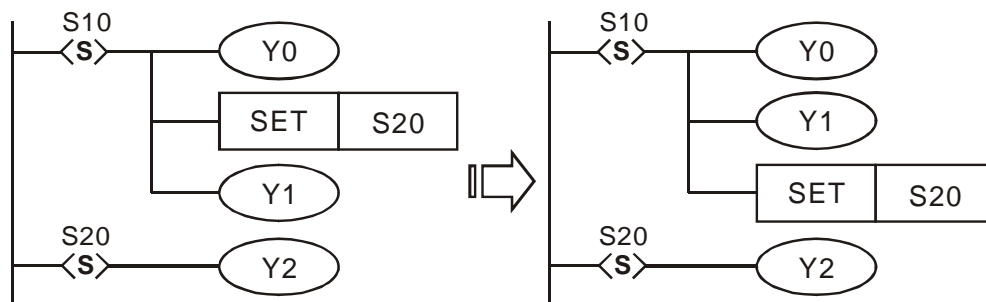
STL	Sn
LD	X0
MPS	
AND	X1
OUT	Y1
MRD	
AND	X2
OUT	M0
MPP	
AND	X3
OUT	Y2

설명:

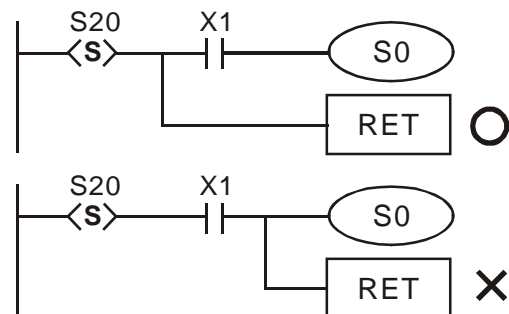
스텝포인트의 BUS 는 MPS / MRD / MPP 명령어들을 직접적으로 사용할 수 없습니다. MPS / MRD / MPP 명령들을 사용하기 전에 LD 나 LDI 명령을 사용하기 위해 필요합니다.

## 8. 그밖에 :

일반적으로, 다음 스텝으로 이동하기 위해 사용되는 명령(SET S□ or OUT S□)은 모든 관련된 출력량과 작용을 마친 후 사용하는 게 낫다. 다음 그림에서는 PLC 에 의해 실행한 후 그들이 같아지는 것입니다. 만약에 S10 에 많은 조건과 작용이 있습니다.면, 왼쪽 그림에서 오른쪽 그림으로 수정하고 모든 관련된 출력량이나 작용들을 마친 후 SET S20 을 실행하도록 권장합니다. 이렇게, 과정은 유지하는 데 명백하고 쉬워진다.



스텝 ladder 프로그램이 끝난 후 추가 RET 명령을 필요로 했고 RET 명령은 또한 오른쪽 그림과 같이 STL 이후에 추가가 필요해집니다.



## 4.4 스텝 Ladder 프로그램 설계의 나머지

1. SFC 의 정면의 스텝포인트는 주요 스텝포인트 S0~S9 라 불려진다. 과정의 시작이 되는 주요 스텝포인트를 이용하고, 완성된 과정의 마무리로 RET 명령을 사용하십시오.
2. 만약 STL 명령을 사용하지 않는다면 S 는 일반적 보조교체로 제공될 수 있습니다.
3. 스텝포인트의 수 S 는 반복적으로 사용될 수 없습니다.
4. 과정 카테고리:

단일 과정: 오직 프로그램상에만 과정이 있습니다. (선택적 분산과 집중, 동시적 분산과 집중은 포함되지 않습니다.)

복잡한 단일 과정: 오직 프로그램상에만 과정이 있고, 선택적 분산과 선택적 집중, 동시적 분산과 동시적 집중 과정을 포함합니다.

조합 과정: 프로그램상에 다수의 단일과정이 있고, 최대 10 개 과정(S0~S9)가 있습니다.

5. 과정 분리: 한 스텝 ladder 다이어그램 안에 다수의 과정을 쓸 수 있습니다.

오른쪽 그림에 S0 와 S1, 두 개의 단일과정이 있습니다. ;

프로그램의 과정은 처음 S0 ~S30 안에 쓰여지고 나서

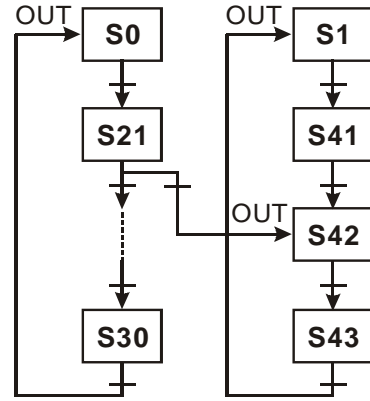
S1~S43 에 쓰여진다.

과정상 어느 한 스텝포인트는 다른 과정상에 있는 특정 스텝포인트로 올라갈 수 있습니다.

오른쪽 그림에서 S21 아래 조건이 고정되는 즉시 S1

과정상에 특정 S42 스텝포인트로 올라갈 수 있습니다. ;

이러한 움직임을 분리 스텝 포인트라 일컫는다.



6. 분산과정상 제약 규정:(다음의 예를 보세요.)

- a) 8 개의 분산 스텝포인트 이상은 분산과정으로 사용될 수 있습니다.
- b) 16 개 이상의 고리는 복수분산이나 동시적 집합과정의 조화로 사용될 수 있습니다.
- c) 과정상 어느 한 스텝포인트는 다른 과정상에 있는 특정 스텝포인트로 올라갈 수 있습니다.

7. 스텝포인트 재설정 및 출력 금지 :

- a) f 스텝포인트 섹션을 재설정하기 위해 꺼지게 하는 ZRST 명령을 사용하십시오.
- b) PLC 의 Y출력 사용 금지 (M1034=ON).

8. 스텝포인트의 유지 :

PLC 가 전력공급 실패에 직면했을 때, 유지되는 스텝포인트는 ON/OFF 상태로 기억될 것이고, 전력이 되살아난 후에는 전력공급 실패 전 실행으로 진행됩니다. S0~S127 는 일반적으로 유지 스텝포인트입니다.

9. 특정 보조용 교체 및 특정 레지스터: 4.6 장 IST 명령 세부사항을 참조하세요.

Device	설명
M1040	스텝변환 금지. M1040 이 On 일 때, 모든 스텝포인트 이동은 금지됩니다.
M1041	스텝변환 시작. IST 명령을 위한 깃발
M1042	진동 시작. IST 명령을 위한 깃발
M1043	초기 재설정 완료됨. IST 명령을 위한 깃발
M1044	원래 상태. IST 명령을 위한 깃발
M1045	모든 출력 제거 금지. IST 명령을 위한 깃발
M1046	STL 상태 설정. 스텝포인트가 켜지자마자 M1046 도 켜진다.

M1047	STL 상태체크 가능
D1040	ON state number 1 of step point S
D1041	ON state number 2 of step point S
D1042	ON state number 3 of step point S
D1043	ON state number 4 of step point S
D1044	ON state number 5 of step point S
D1045	ON state number 6 of step point S
D1046	ON state number 7 of step point S
D1047	ON state number 8 of step point S

#### 4.5 절차의 분류

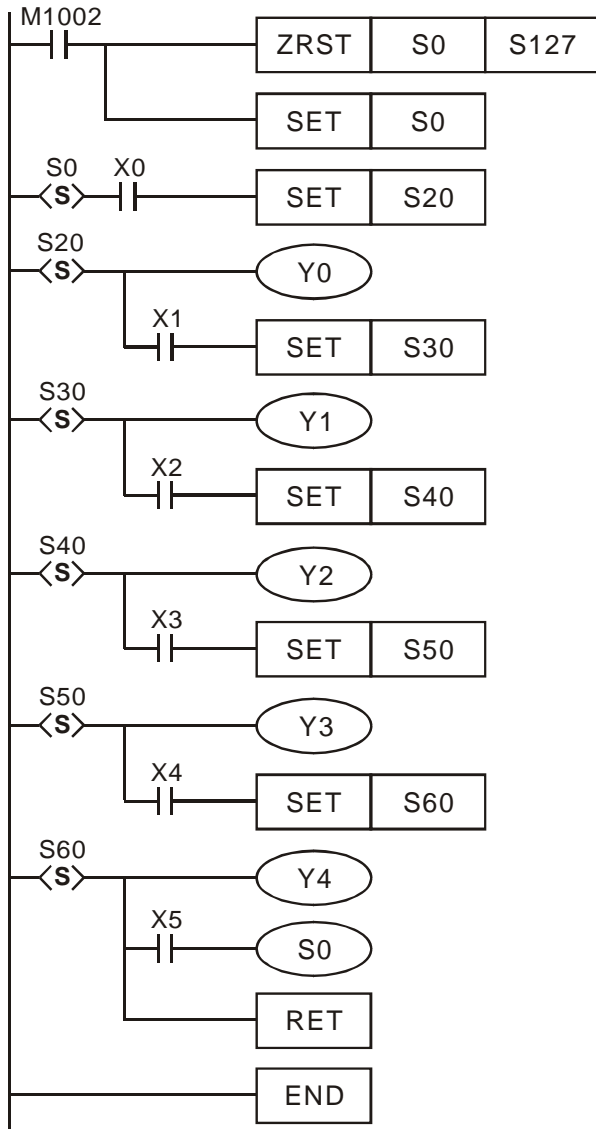
**A. 단일 과정** : 기초 스텝작용은 단일 과정 제어작용입니다.

스텝 ladder 다이어그램의 첫 스텝포인트는 주 스텝 포인트라 불리우고, 기호는 S0~S9 입니다. 주 스텝포인트 이후의 그러한 스텝포인트는 일반적 스텝 포인트라 불리고, 기호는 S10~S1023 입니다. S10~S19 는 IST 명령을 사용하면서 초기 설정 스텝포인트로서 사용될 것입니다.

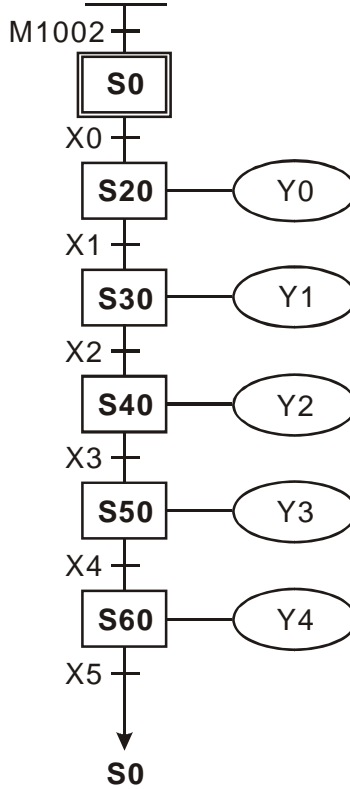
##### **A-1 분산과 집합 없는 단일 과정**

과정이 끝난 후 스텝포인트의 제어전력이 주 스텝포인트로 이동하고 있습니다.

Step Ladder Diagram

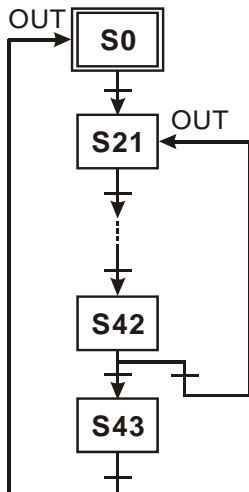


SFC diagram

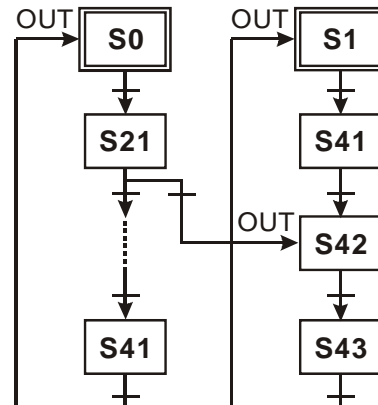


## A-2 상승과정

1. 스텝포인트의 제어전력에서 어느 스텝포인트로의 상승 이동

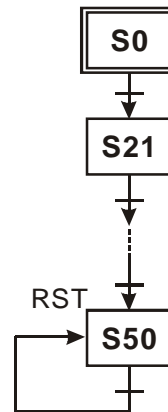


2. 스텝포인트의 제어전력에서 다른 과정 스텝포인트로의 이동



**A-3 재설정 과정**

오른쪽 그림에서 S50 은 자신으로 재설정할 것이고,  
조건이 고정됐을 때 과정은 끝이 난다.

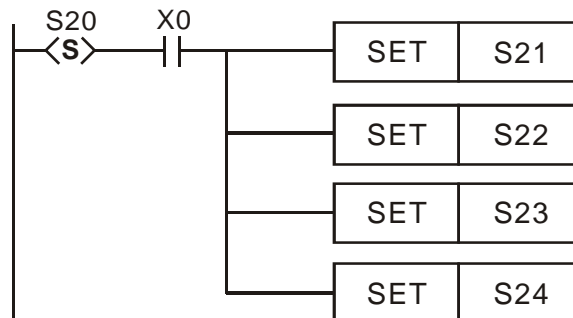
**B. 복잡한 단일 과정 :**

선택적 분산과 선택적 집합과정, 동시적 분산과 동시적 집합과정을 포함합니다.

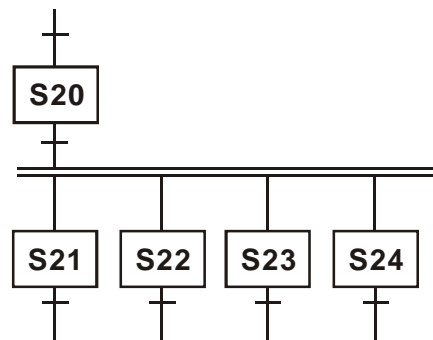
**B-1 동시적 분산 구조**

제시된 조건이 고정되었을 때 많은 상태로 이동하는 상황은 아래에서 보여지는 바와 같이 동시적 분산구조라 일컫는다. X0=On 일 때, S20 는 동시에 S21, S22, S23 그리고 S24 로 이동할 것입니다.

동시적 분산 Ladder 다이어그램 :

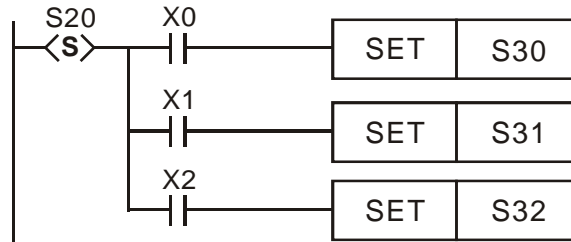


동시적 분산 SFC 다이어그램 :

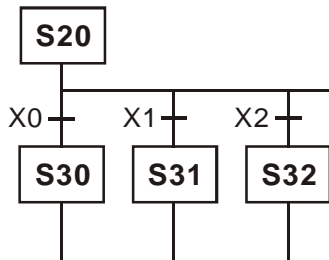
**B-2 선택적 분산 구조**

현재 상태의 각각의 조건이 고정되었을 때 각각의 상태로 이동하는 상황은 아래에서 보여지는 바와 같이 선택적 분산구조로 일컫는다. S20 은 X0=On 일 때 S30 으로 이동할 것이고, S20 은 X1=On 일 때 S31 로 이동할 것이며, S20 은 X2=On 일 때 S32 로 이동할 것입니다.

선택적 분산 Ladder 다이어그램:



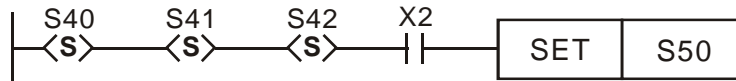
선택적 분산 SFC 다이어그램 :



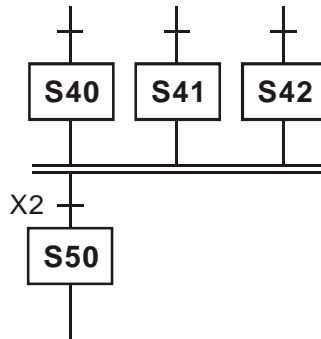
## B-3 동시적 집합 구조

지속상태가 고정되었을 때 동시에 다음 상태로 이동하는 상황은 동시적 집합이라 불립니다.

동시적 집합 Ladder 다이어그램 :



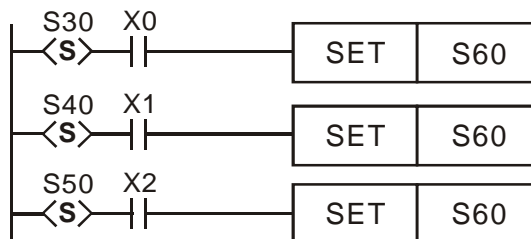
동시적 집합 SFC 다이어그램 :



## B-4 선택적 집합 구조

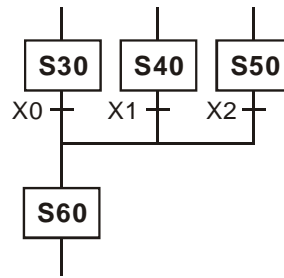
아래의 ladder 다이어그램은 선택적 집합입니다. 그것은 S30, S40, S50 중에 하나가 S60 으로 이동하는 것을 의미합니다.

선택적 집합 Ladder 다이어그램 :

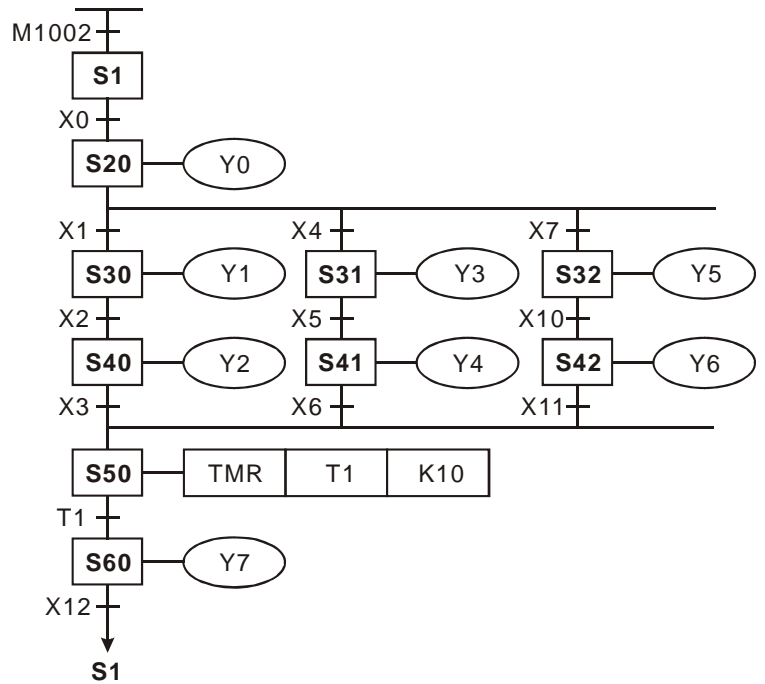
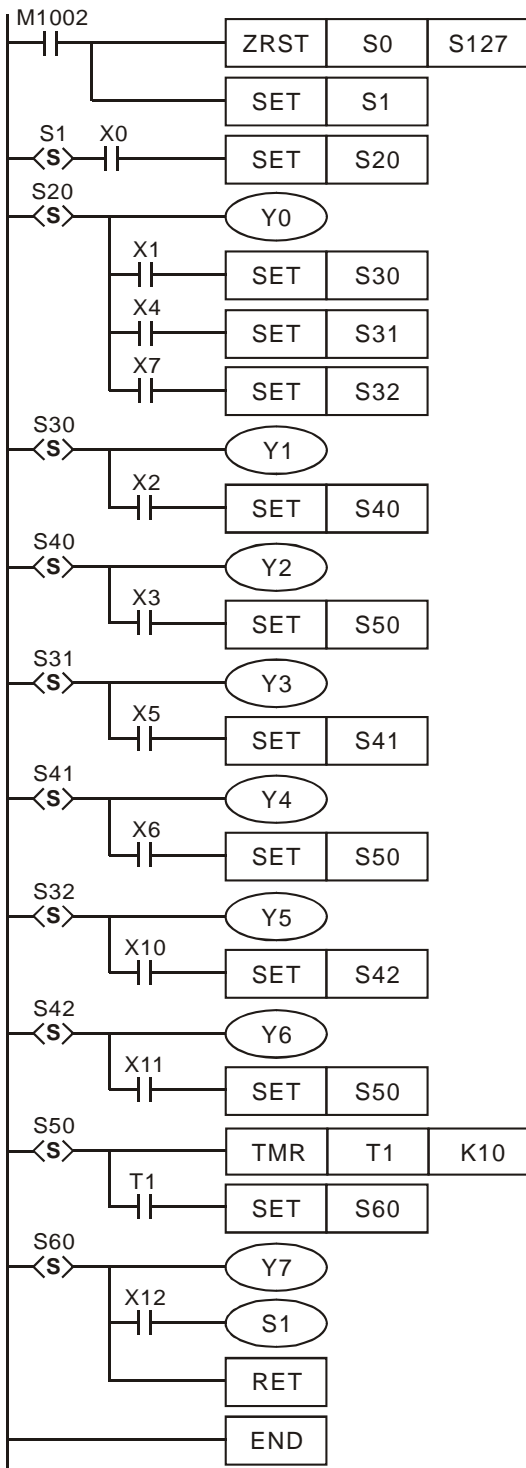


선택적 집합 SFC 다이어그램 :

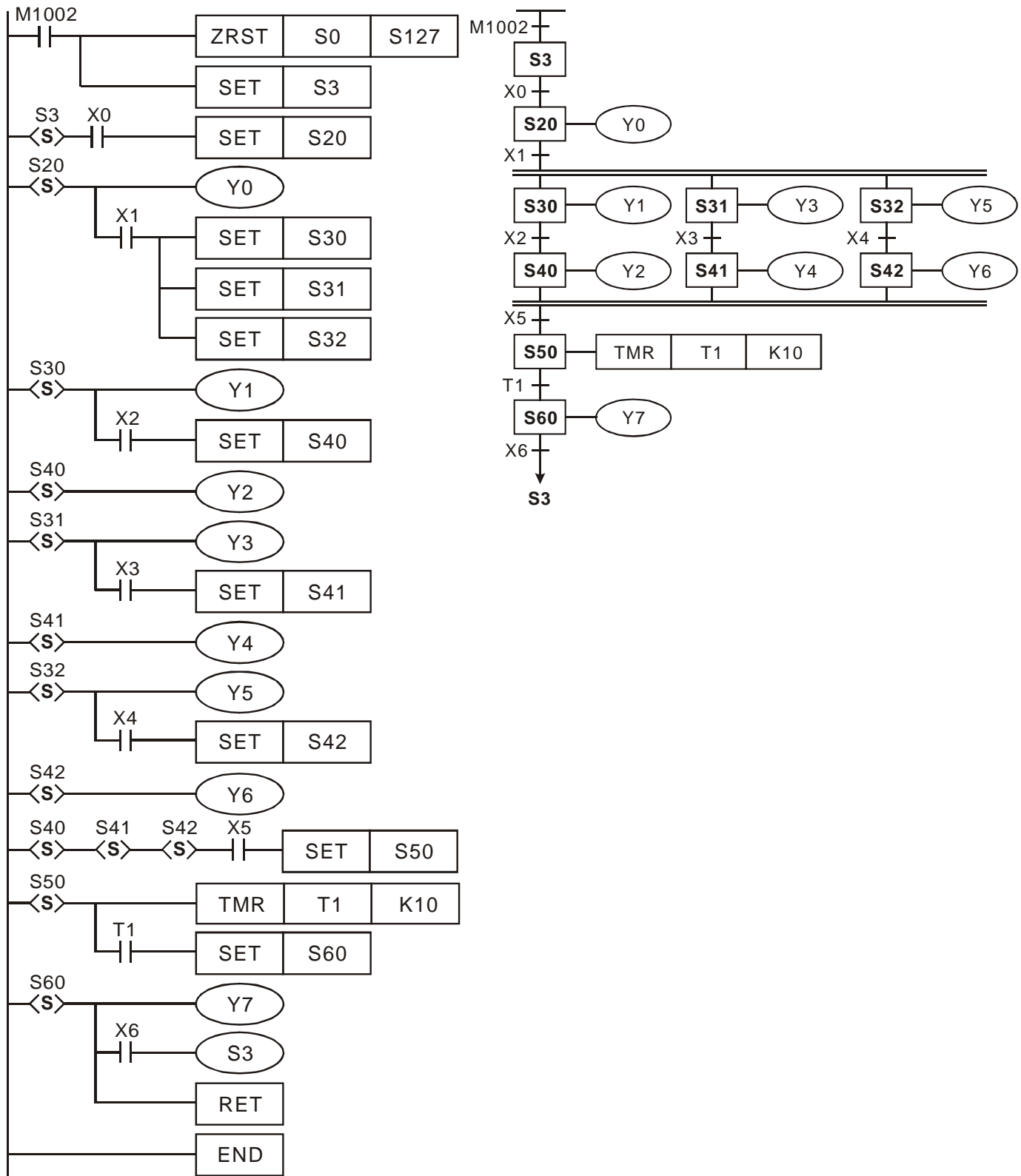




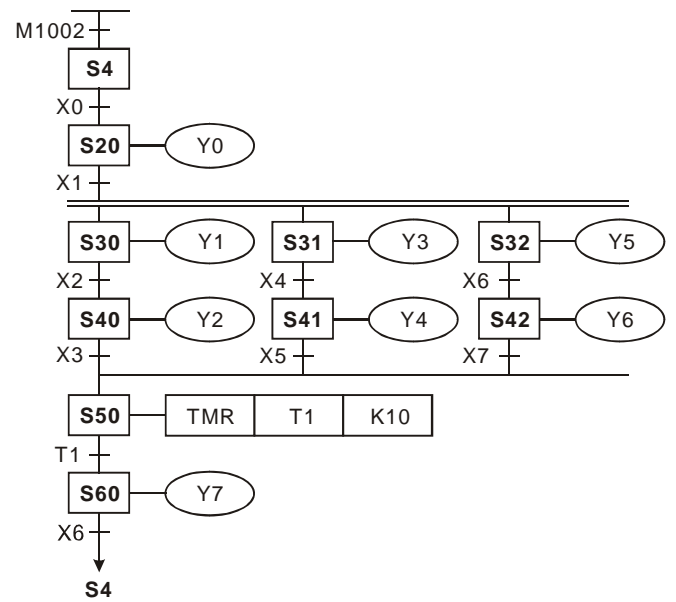
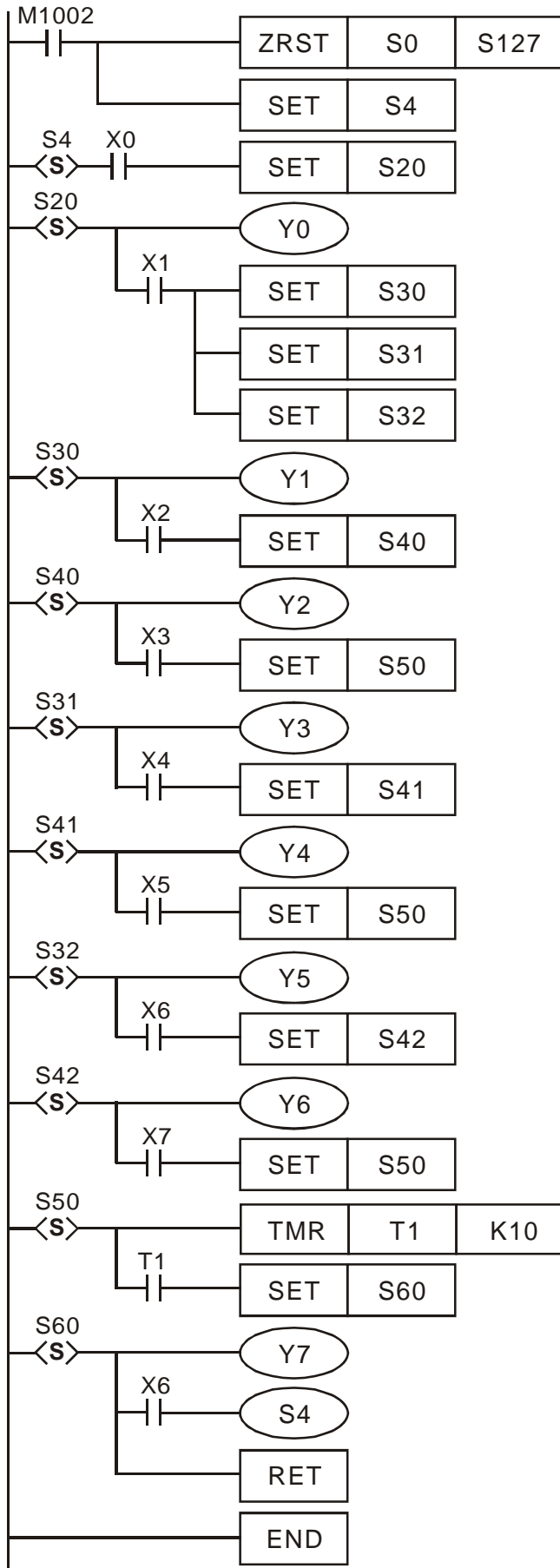
선택적 분산과 선택적 집합과정의 예



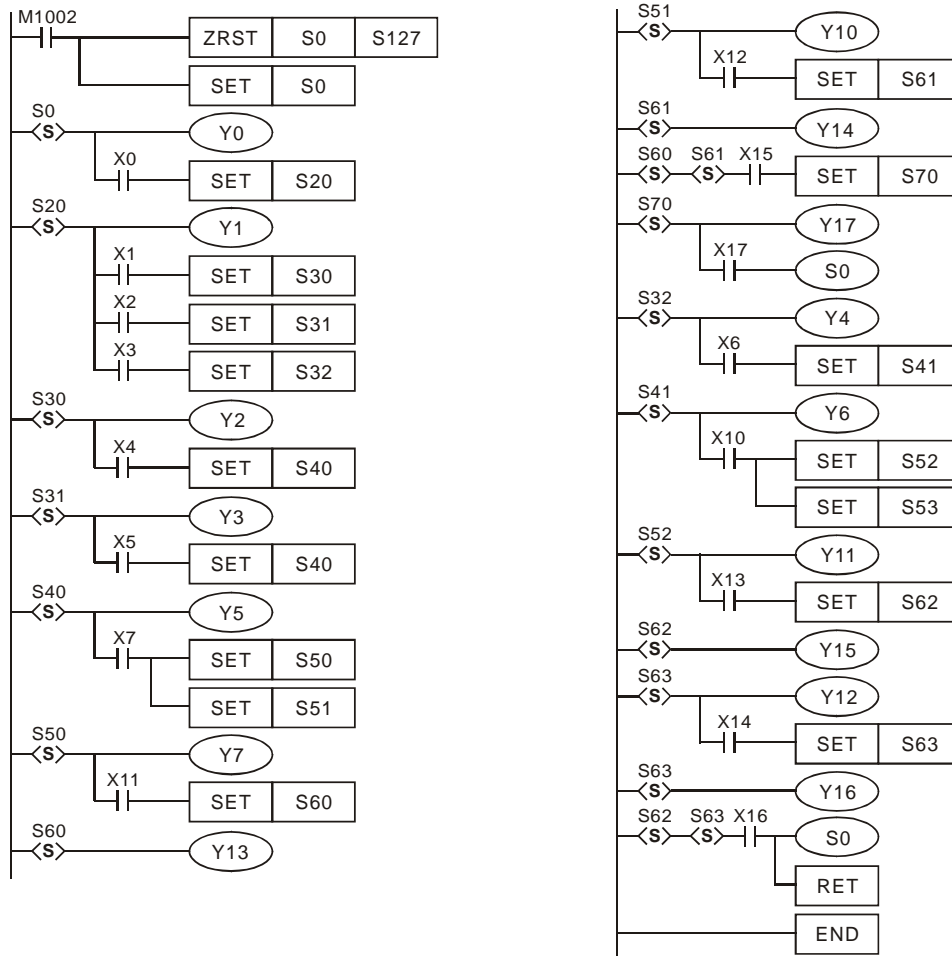
## 동시적 분산과 동시적 집합과정의 예



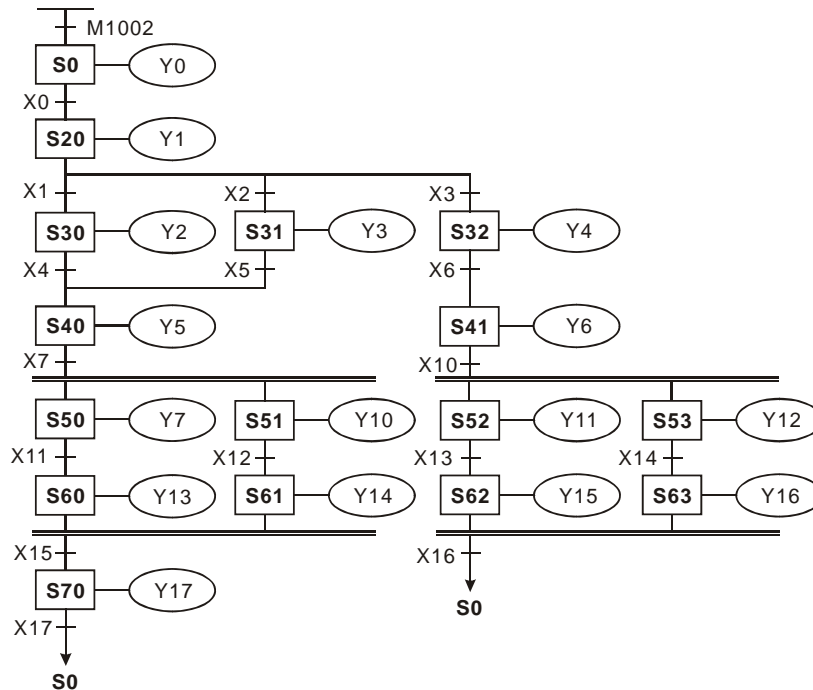
## 동시적 분산과 선택적 집합과정의 예



조합 예 1: (선택적 분산과 집합, 동시적 분산과 집합 포함)

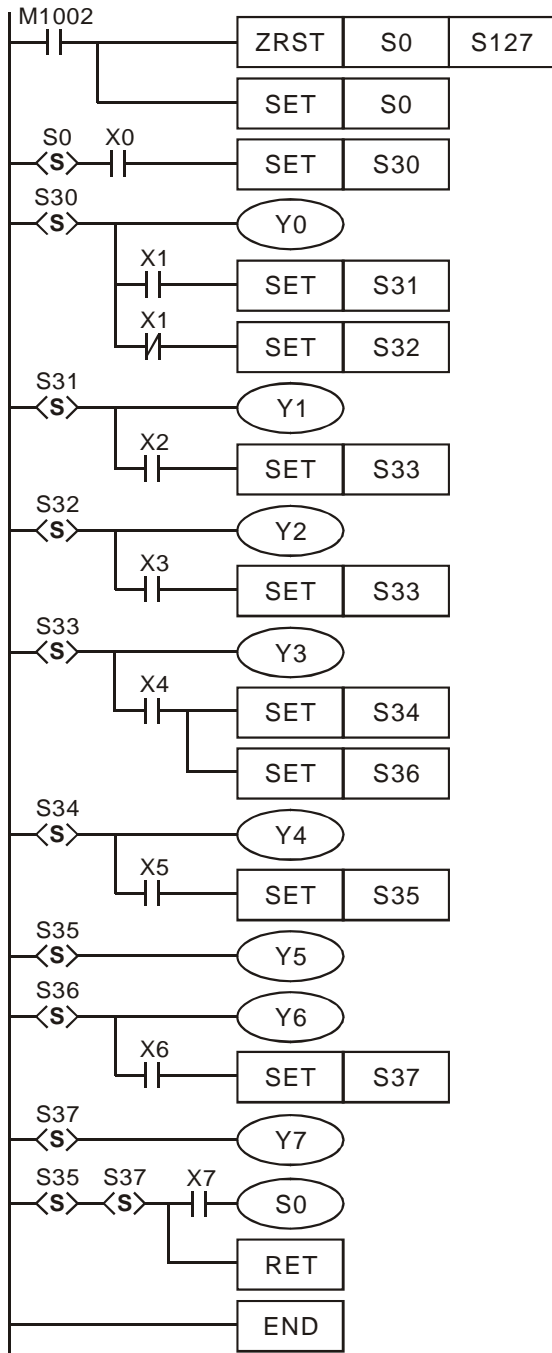


SFC Figure:

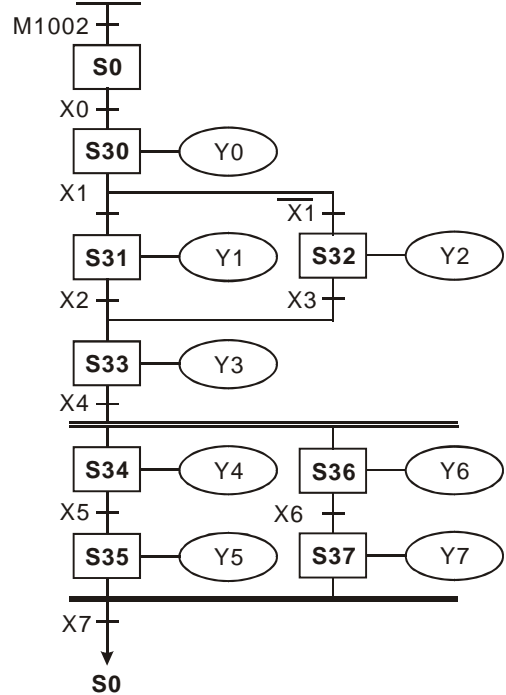


조합 예 2: (선택적 분산과 집합, 동시적 분산과 집합 포함)

Ladder Diagram:

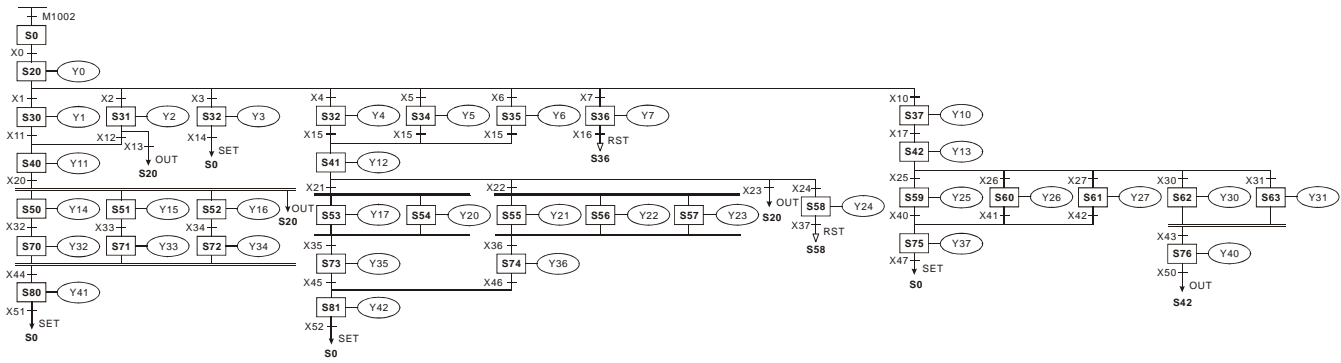


SFC Figure:



분산과정상 제약 규정 :

1. 8 개 이상의 분산 스텝포인트는 분산과정에서 사용될 수 있습니다. 다음 그림에서 스텝포인트 S20 이후의 최대 분산 스텝포인트는 8(S30 - S37)개 입니다.
2. 16 이상의 고리는 복수분산이나 동시적 집합과정의 조화로 사용되어질 수 있습니다. 아래 그림에서는 스텝포인트 S40 이후에 4 개의 스텝포인트, S41 이후에 7 개의 스텝포인트, 그리고 S42 이후에 5 개의 스텝포인트가 있습니다.는 것을 알 수 있습니다. 이 과정에서 최대는 16 개 고리가 있습니다.
3. 과정상 어느 한 스텝포인트는 다른 과정상에 있는 특정 스텝포인트로 올라갈 수 있습니다.



## 4.6 IST 명령

[illegible]

피연산자 :

**S:** 초기 입력 기호 (피연산자 **S**는 8개의 연속장치를 차지할 것입니다.)

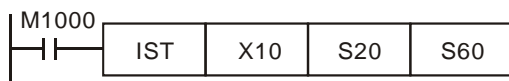
**D<sub>1</sub>:** 자동모드에서 나타내는 스텝포인트의 최소값

**D<sub>2</sub>:** 자동모드에서 나타내는 스텝포인트의 최대값

설명:

1. 피연산자 **D<sub>1</sub>** and **D<sub>2</sub>** 의 이용범위:for ES series, S20~S127; for SA/EH series, S20~S899.
2. IST는 편리한 자동 제어명령어로 특정 보조교체로 적응시키는 스텝 ladder 제어과정의 초기상태에서 특별하게 만들어 주는 편리한 명령어다.
3. IST 명령은 프로그램에서 오직 한 번만 사용될 수 있습니다.
4. ES/SA 시리즈는 index register E, F를 지원하지 않습니다.

### 프로그램 예 1 :



**S** X10: 개별 작동(수동 작동)

### X11: Zero 포인트 복귀

X12: 스텝 작동

### X13: 1 사이클 작동

X14: 지속적인 작동

### X15: Zero 포인트 복귀 시작 스위치

## X16: 시작 스위치

X17: 정지 스위치

1. IST 명령이 실행되었을 때, 다음과 같은 특정 보조교체는 자동적으로 변경될 것입니다.

M1040: 이동금지

M1041: 이동시작

M1042: 진동상태

M1047: STL 상태 체크 가능

S0: 수동 작동/최초 상태 단계점

S1: Zero 포인트 복귀/최초 상태 단계점

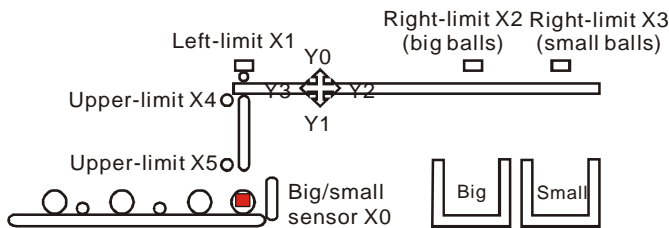
S2: 자동 작동/최초 상태 단계점

1. IST 명령이 사용되었을 때, S10~S19는 0 포인트이고, 이 상태에서는 일반적 스텝포인트로 사용될 수 없습니다. 그러나 S0~S9 스텝 포인트가 사용되었을 때, S0은 “수동 작동”, S1은 “0 포인트 전환 작동”, 그리고 S2는 “자동 작동”을 가리킵니다. 그래서 프로그램상에 쓰여 있는 이러한 세가지 초기 상태 스텝 포인트의 세 회로가 될 수 있습니다.
2. S1으로 전환할 때 (zero 포인트 복귀 모드), 0 포인트는 S10~S19가 켜져 있는 것 중 하나가 켜져 있을 때 어떤 작용이 이루어지지 않을 것입니다.
3. S2로 전환할 때 (자동 작동 모드), 자동 작동에는 D<sub>1</sub> to D<sub>2</sub>이 켜졌거나 M1043이 켜졌거나의 사이에 S의 하나에 어떤 작용이 없을 것입니다.

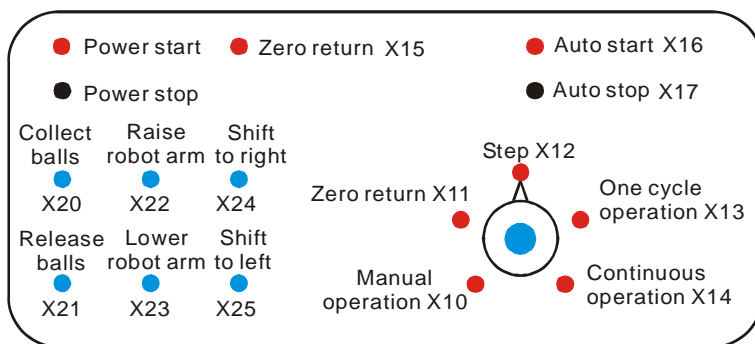
## 프로그램 예 2 :

예제: 로봇 팔 제어(IST 명령 사용):

- a) 움직임 요구: 예를 들어 두 가지 종류의 공이 서로 다른 상자에 구분되어 저장되어 있습니다. 제어를 위해 제어판을 분배합니다.
- b) 로봇 팔 움직임: 낮아진 로봇 팔, 수집된 공, 올려진 로봇 팔, 오른쪽으로 들어올림, 낮아진 로봇 팔, 놓여진 공, 들어진 로봇 팔, 마지막 움직임으로 왼쪽으로 올려짐.
- c) I/O Device:



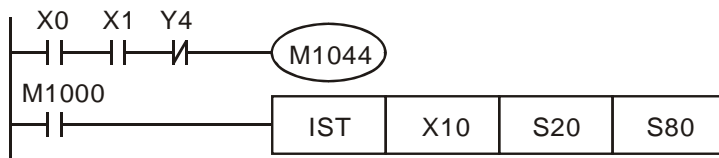
- d) Control panel



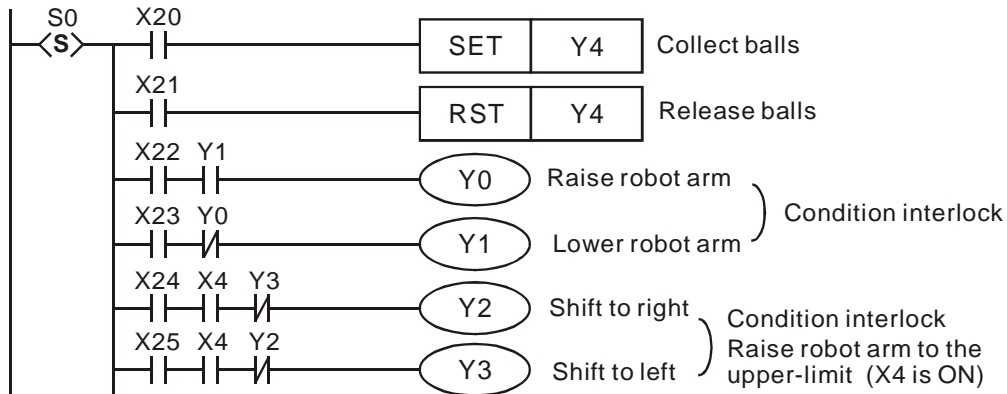
Big/small sensor X0.

로봇팔의 왼쪽 제한 X1, 오른쪽 제한 X2 (큰 공), 오른쪽 제한 X3(작은 공), 윗쪽 제한 X4, 그리고 아래쪽 제한 X5. 들어올려진 로봇 팔 Y0, 낮춰진 로봇 팔 Y1, 오른쪽으로 들어짐 Y2, 왼쪽 위로 올라감 Y3, 그리고 수집된 공 Y4.

START circuit:

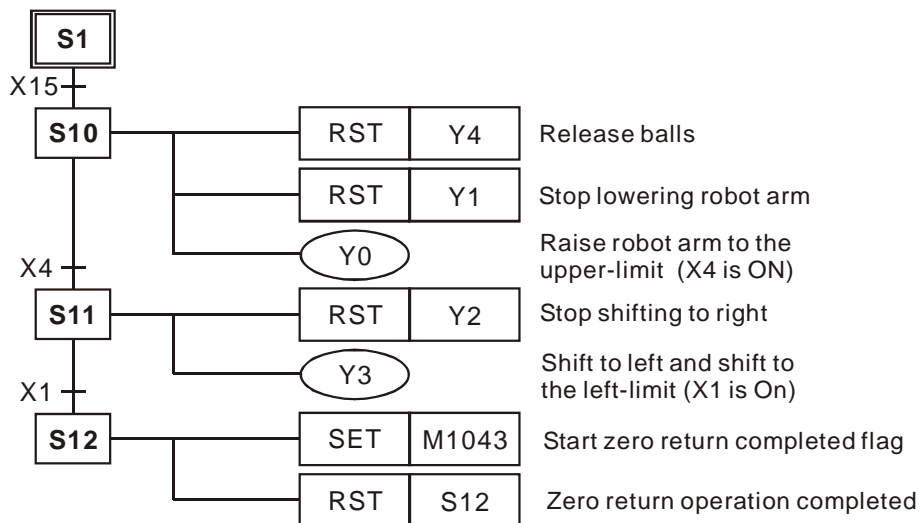


## Manual operation mode:



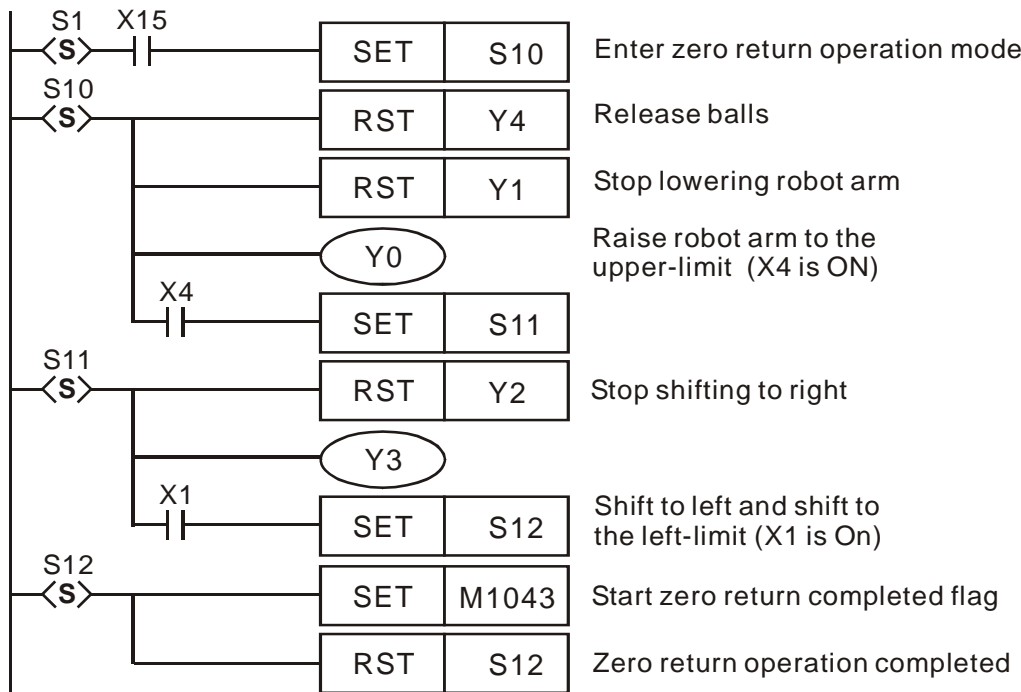
## Zero point return mode:

SFC figure:



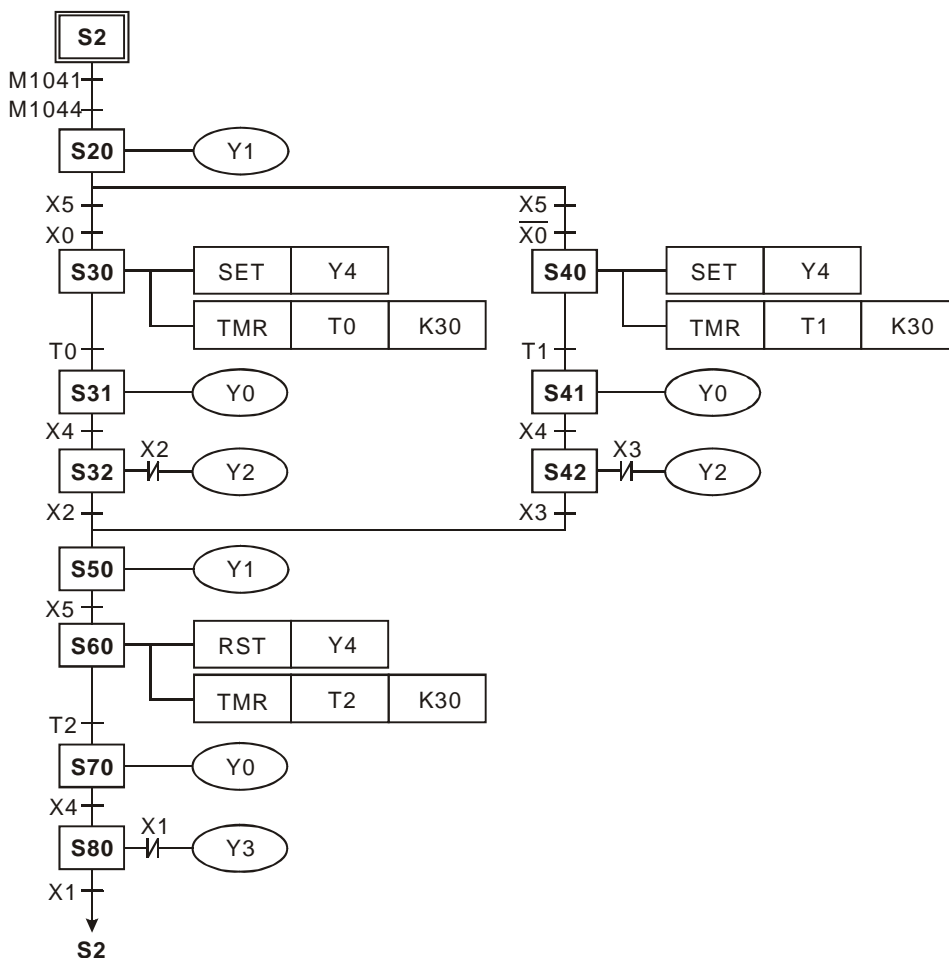
Ladder Diagram:



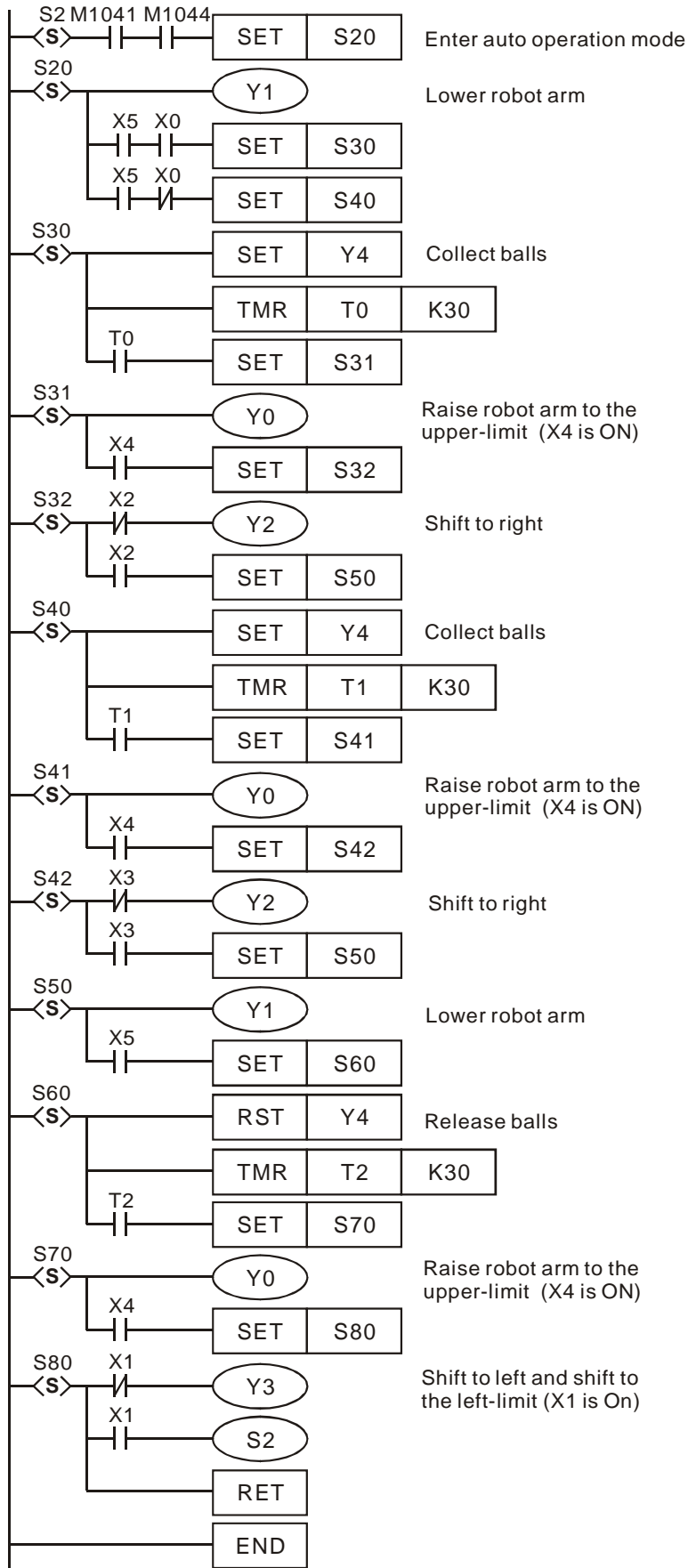


자동 작동 (스텝/1 사이클/지속 작동 모드):

SFC 도해:



Ladder Diagram:



## 5.1 파라미터의 정리

요약	API	기억용 코드		P 명령	기능	적용 모델			STEPS		페이지
		16 비트	32 비트			ES	SA	EH	16 비트	32 비트	
연결로 콘트롤	00	CJ	-	✓	전제적 이동	✓	✓	✓	3	-	6-1
	01	CALL	-	✓	서브루틴 불러오기	✓	✓	✓	3	-	6-5
	02	SRET	-	-	서브 루틴 복귀	✓	✓	✓	1	-	6-6
	03	IRET	-	-	복귀 장애	✓	✓	✓	1	-	6-8
	04	EI	-	-	실행 장애	✓	✓	✓	1	-	6-8
	05	DI	-	-	정지 장애	✓	✓	✓	1	-	6-8
	06	FEND	-	-	첫 마침	✓	✓	✓	1	-	6-12
	07	WDT	-	✓	감시 타이머 재생	✓	✓	✓	1	-	6-14
	08	FOR	-	-	FOR-NEXT 연결로의 시작	✓	✓	✓	3	-	6-15
	09	NEXT	-	-	FOR-NEXT 연결로의 종료	✓	✓	✓	1	-	6-15
변속 캠페인	10	CMP	DCMP	✓	비교	✓	✓	✓	7	13	6-18
	11	ZCP	DZCP	✓	영역 비교	✓	✓	✓	9	17	6-19
	12	MOV	DMOV	✓	데이터 이동	✓	✓	✓	5	9	6-20
	13	SMOV	-	✓	자리 이송 움직임	-	✓	✓	11	-	6-21
	14	CML	DCML	✓	증정	✓	✓	✓	5	9	6-23
	15	BMOV	-	✓	블럭 움직임	✓	✓	✓	7	-	6-25
	16	FMOV	DFMOV	✓	채우기	✓	✓	✓	7	13	6-27
	17	XCH	DXCH	✓	데이터 교환	✓	✓	✓	5	9	6-28
	18	BCD	DBCD	✓	BCD 로 BIN 를 데이터 변환	✓	✓	✓	5	9	6-30
	19	BIN	DBIN	✓	BIN 로 BCD 데이터를 변환	✓	✓	✓	5	9	6-31
네가지 기초 계산 작업	20	ADD	DADD	✓	BIN 데이터의 추가 적용	✓	✓	✓	7	13	6-33
	21	SUB	DSUB	✓	BIN 데이터의 절감 적용	✓	✓	✓	7	13	6-35
	22	MUL	DMUL	✓	BIN 데이터의 곱셈 적용	✓	✓	✓	7	13	6-37
	23	DIV	DDIV	✓	BIN 데이터의 나누기 적용	✓	✓	✓	7	13	6-39
	24	INC	DINC	✓	1 추가 적용	✓	✓	✓	3	5	6-41
	25	DEC	DDEC	✓	1 절감 적용	✓	✓	✓	3	5	6-42
	26	WAND	DAND	✓	회로 제품 (AND) 작업 적용	✓	✓	✓	7	13	6-43
	27	WOR	DOR	✓	회로 합계 (OR) 작업 적용	✓	✓	✓	7	13	6-44
	28	WXOR	DXOR	✓	배타적 회로 추가 (XOR) 작업	✓	✓	✓	7	13	6-45
	29	NEG	DNEG	✓	취소	✓	✓	✓	3	5	6-47
회전 과 이동	30	ROR	DROR	✓	우로 회전	✓	✓	✓	5	9	6-49
	31	ROL	DROL	✓	좌로 회전	✓	✓	✓	5	9	6-50
	32	RCR	DRCR	✓	운반용 깃발이 첨부된 상태에서 우로 회전.	✓	✓	✓	5	9	6-51
	33	RCL	DRCL	✓	운반용 깃발이 첨부된 상태에서 좌로 회전.	✓	✓	✓	5	9	6-52
	34	SFTR	-	✓	우측으로 특정 장치의 데이터 이송	✓	✓	✓	9	-	6-53
	35	SFTL	-	✓	좌측으로 특정 장치의 데이터 이송	✓	✓	✓	9	-	6-54
	36	WSFR	-	✓	우측으로 기록을 이송	-	✓	✓	9	-	6-55
	37	WSFL	-	✓	좌측으로 기록을 이송	-	✓	✓	9	-	6-57
	38	SFWR	-	✓	기록 쓰기 이송	-	✓	✓	7	-	6-58
	39	SFRD	-	✓	기록 읽기 이송	-	✓	✓	7	-	6-59
	40	ZRST	-	✓	특정 장치의 영역 재설정	✓	✓	✓	5	-	6-60
	41	DECO	-	✓	8 → 256 비트 디코더	✓	✓	✓	7	-	6-62
비터 작	42	ENCO	-	✓	256 → 8 비트 인코더	✓	✓	✓	7	-	6-64
	43	SUM	DSUM	✓	온 비트의 합계	✓	✓	✓	5	9	6-66

요약	API	기억용 코드		P 명령	기능	적용 모델			STEPS		페이지
		16 비트	32 비트			ES	SA	EH	16 비트	32 비트	
	44	BON	DBON	✓	특정 비트 상태 확인	✓	✓	✓	7	13	6-67
	45	MEAN	DMEAN	✓	평균 값	✓	✓	✓	7	13	6-68
	46	ANS	-	-	알람 장치 출력	-	✓	✓	7	-	6-69
	47	ANR	-	✓	알람 장치 재설정	-	✓	✓	1	-	6-70
	48	SQR	DSQR	✓	BIN 의 사각 루트	✓	✓	✓	5	9	6-72
	49	FLT	DFLT	✓	BIN 을 이진 부동 포인트로 변환	✓	✓	✓	5	9	6-73
고속 처리	50	REF	-	✓	I/O 재생	✓	✓	✓	5	-	7-1
	51	REFF	-	✓	재생 후 입력 필터의 응답시간을 조정	-	✓	✓	3	-	7-2
	52	MTR	-	-	행렬 입력	-	✓	✓	9	-	7-3
	53	-	DHSCS	-	고속 계수 비교 세트	✓	✓	✓	-	13	7-5
	54	-	DHSCR	-	고속 계수 비교 리셋	✓	✓	✓	-	13	7-14
	55	-	DHSZ	-	영역 비교 (고속 계수기)	-	✓	✓	-	17	7-16
	56	SPD	-	-	속도 감지	✓	✓	✓	7	-	7-22
	57	PLSY	DPLSY	-	파동 출력	✓	✓	✓	7	13	7-24
	58	PWM	-	-	파동 넓이 조정 출력	✓	✓	✓	7	-	7-30
	59	PLSR	DPLSR	-	가속/감속과의 파동 흐름 출력	✓	✓	✓	9	17	7-33
편리한 명령	60	IST	-	-	수동/자동 컨트롤	✓	✓	✓	7	-	7-37
	61	SER	DSER	✓	데이터 후입 선출열 검색	-	✓	✓	9	17	7-43
	62	ABSD	DABSD	-	절대 드럼 제어장치	-	✓	✓	9	17	7-44
	63	INCD	-	-	증가 드럼 제어장치	-	✓	✓	9	-	7-46
	64	TTMR	-	-	타이머 알림	-	✓	✓	5	-	7-48
	65	STMR	-	-	특수 타이머	-	✓	✓	7	-	7-50
	66	ALT	-	-	켜짐/꺼짐 교체 명령	✓	✓	✓	3	-	7-51
	67	RAMP	-	-	급작 시그널	-	✓	✓	9	-	7-52
	69	SORT	-	-	데이터 종류	-	✓	✓	11	-	7-54
외부 I/O 디스플레이	70	TKY	DTKY	-	10 단추 키패드 입력	-	✓	✓	7	13	7-56
	71	HKY	DHKY	-	16 단추 키패드 입력	-	✓	✓	9	17	7-58
	72	DSW	-	-	디지털 스위치 입력	-	✓	✓	9	-	7-60
	73	SEGD	-	✓	7 단계 디스플레이 판 해석	✓	✓	✓	5	-	7-62
	74	SEGL	-	-	7 단계 디스플레이 스캔 출력	✓	✓	✓	7	-	7-63
	75	ARWS	-	-	화살표 키패드 입력	-	✓	✓	9	-	7-66
	76	ASC	-	-	ASCII 코드 변환	-	✓	✓	11	-	7-67
	77	PR	-	-	인쇄	-	✓	✓	5	-	7-68
	78	FROM	DFROM	✓	특수 모듈 CR 데이터 읽음	✓	✓	✓	9	17	7-70
	79	TO	DTO	✓	특수 모듈 CR 데이터 기명표시	✓	✓	✓	9	17	7-71
시리얼 I/O	80	RS	-	-	시리얼 데이터 통신	✓	✓	✓	9	-	7-75
	81	PRUN	DPRUN	✓	8 진법 번호 시스템 변속	-	✓	✓	5	9	7-87
	82	ASCI	-	✓	HEX 에서 ASCII 로 변환	✓	✓	✓	7	-	7-88
	83	HEX	-	✓	ASCII 에서 HEX 로 변환	✓	✓	✓	7	-	7-92
	84	CCD	-	✓	체크 코드	-	✓	✓	7	-	7-95
	85	VRRD	-	✓	분압기 읽기	-	✓	✓	5	-	7-97
	86	VRSC	-	✓	분압기 스케일	-	✓	✓	5	-	7-99
	87	ABS	DABS	✓	절대 값	✓	✓	✓	3	5	7-100
	88	PID	DPID	-	PID 계산	✓	✓	✓	9	-	7-101
기초 명령	89	PLS	-	-	에지 상승 출력	✓	✓	✓	3	-	3-13
	90	LDP	-	-	에지 상승 감지 작업	✓	✓	✓	3	-	3-11
	91	LDF	-	-	에지 하강 감지 작업	✓	✓	✓	3	-	3-12
	92	ANDP	-	-	에지 상승 감지 작업을 위한 시리즈 연결 명령	✓	✓	✓	3	-	3-12

요약	API	기억용 코드		P 명령	기능	적용 모델			STEPS		페이지
		16 비트	32 비트			ES	SA	EH	16 비트	32 비트	
	93	ANDF	-	-	에지 하강 감지 작업을 위한 시리즈 연결 명령	✓	✓	✓	3	-	3-12
	94	ORP	-	-	에지 상승 감지 작업을 위한 평행 연결 명령	✓	✓	✓	3	-	3-13
	95	ORF	-	-	에지 하강 감지 작업을 위한 평행 연결 명령	✓	✓	✓	3	-	3-13
	96	TMR	-	-	타이머	✓	✓	✓	4	-	3-8
	97	CNT	DCNT	-	계수기	✓	✓	✓	4	6	3-9
	98	INV	-	-	작업 전환	✓	✓	✓	1	-	3-14
	99	PLF	-	-	에지 하강 출력	✓	✓	✓	3	-	3-14
텔타 AC 모터 드라이브의 통신 명령	100	MODRD	-	-	MODBUS 데이터 읽기	✓	✓	✓	7	-	8-1
	101	MODWR	-	-	MODBUS 데이터 기록	✓	✓	✓	7	-	8-5
	102	FWD	-	-	VFD-A 시리즈 드라이브 전진 명령	✓	✓	✓	7	-	8-10
	103	REV	-	-	VFD-A 시리즈 드라이브 역진행 명령	✓	✓	✓	7	-	8-10
	104	STOP	-	-	VFD-A 시리즈 드라이브 정지 명령	✓	✓	✓	7	-	8-10
	105	RDST	-	-	VFD-A 시리즈 드라이브 상태 읽기	✓	✓	✓	5	-	8-12
	106	RSTEF	-	-	VFD-A 시리즈 드라이브 이상 재설정	✓	✓	✓	5	-	8-14
	107	LRC	-	✓	LRC 에러 확인	-	✓	✓	7	-	8-15
	108	CRC	-	✓	CRC 에러 확인	-	✓	✓	7	-	8-17
	109	SWRD	-	✓	디지털 스위치 읽기	-	✓	✓	3	-	8-20
부동 작업	110	-	DECOMP	✓	이진 부동 포인트 비교	✓	✓	✓	-	13	8-21
	111	-	DEZCP	✓	이진 부동 포인트 영역 비교	✓	✓	✓	-	17	8-22
	116	-	DRAD	✓	각 → 라디안	-	✓	✓	-	9	8-23
	117	-	DDEG	✓	라디안 → 각	-	✓	✓	-	9	8-24
	118	-	DEBCD	✓	부동 포인트에서 소수점 포인트로 변환	✓	✓	✓	-	9	8-25
	119	-	DEBIN	✓	소수점 포인트에서 이진 부동 포인트로 전환	✓	✓	✓	-	9	8-26
	120	-	DEADD	✓	이진 부동 포인트 첨가	✓	✓	✓	-	13	8-28
	121	-	DESUB	✓	이진 부동 포인트 절감	✓	✓	✓	-	13	8-29
	122	-	DEMUL	✓	이진 부동 포인트 곱하기	✓	✓	✓	-	13	8-30
	123	-	DEDIV	✓	이진 부동 포인트 나누기	✓	✓	✓	-	13	8-31
	124	-	DEXP	✓	이진 부동 포인트의 지수 작업 적용	✓	✓	✓	-	9	8-32
	125	-	DLN	✓	이진 부동 포인트의 내추럴 대수 작업 적용	✓	✓	✓	-	9	8-33
	126	-	DLOG	✓	이진 부동 포인트의 대수 작업 적용	✓	✓	✓	-	13	8-34
	127	-	DESQR	✓	이진 부동 포인트의 사각 루트	✓	✓	✓	-	9	8-35
부동 작업	128	-	DPOW	✓	이진 부동 포인트의 전원 작업 적용	✓	✓	✓	-	13	8-36
	129	INT	DINT	✓	이진 부동 포인트를 BIN 정수로 전환	✓	✓	✓	5	9	8-37
	130	-	DSIN	✓	이진 부동 포인트의 정현 작업	✓	✓	✓	-	9	8-38
	131	-	DCOS	✓	이진 부동 작업의 코사인	✓	✓	✓	-	9	8-40

요약	API	기억용 코드		P 명령	기능	적용 모델			STEPS		페이지
		16 비트	32 비트			ES	SA	EH	16 비트	32 비트	
	132	-	DTAN	✓	이진 부동 포인트의 접선 작업	✓	✓	✓	-	9	8-42
	133	-	DASIN	✓	이진 부동 포인트의 호사인 작업	-	✓	✓	-	9	8-44
	134	-	DACOS	✓	이진 부동 포인트의 호 코사인 작업	-	✓	✓	-	9	8-45
	135	-	DATAN	✓	이진 부동 포인트의 호접선 작업	-	✓	✓	-	9	8-46
	136	-	DSINH	✓	이진 부동 포인트의 과대 사인 작업	-	-	✓	-	9	8-47
	137	-	DCOSH	✓	이진 부동 포인트의 과대 코사인 작업	-	-	✓	-	9	8-48
	138	-	DTANH	✓	이진 부동 포인트의 과대 접선 작업	-	-	✓	-	9	8-49
추가 명령	143	DELAY	-	✓	명령 지연	-	✓	✓	3	-	8-50
	144	GPWM	-	-	종합 파동 넓이 모듈 출력	-	✓	✓	7	-	8-51
	145	FTC	-	-	퍼지 온도 컨트롤	-	✓	✓	9	-	8-52
	147	SWAP	DSWAP	✓	고/저 바이트를 교환	✓	✓	✓	3	5	8-58
	148	MEMR	DMEMR	✓	데이터 백업 메모리 읽기	-	✓	✓	7	13	8-59
	149	MEMW	DMEMW	✓	데이터 백업 메모리 기록	-	✓	✓	7	13	8-60
	150	MODRW	-	-	MODBUS 데이터 읽기/기록	✓	✓	✓	11	-	9-1
	151	PWD	-	-	입력 파동 넓이 감지	-	-	✓	5	-	9-10
	152	RTMU	-	-	I 장애의 실행 시간 측정을 시작	-	-	✓	5	-	9-11
	153	RTMD	-	-	I 장애의 실행 시간 측정을 종료	-	-	✓	3	-	9-11
위치 컨트롤	154	RAND	-	-	랜덤 값	-	✓	✓	7	-	9-13
	155	-	DABSR	-	ABS 전류 값 읽기	-	✓	✓	7	13	9-14
	156	ZRN	DZRN	-	영점 복귀	-	-	✓	9	17	9-19
	157	PLSV	DPLSV	-	변수 속도 파동 출력	-	-	✓	7	13	9-23
	158	DRVI	DDRVI	-	증가 운행	-	-	✓	9	17	9-24
	159	DRVA	DDRVA	-	절대 운행	-	-	✓	9	17	9-29
무기한 달력	160	TCMP	-	✓	시간 비교	-	✓	✓	11	-	9-41
	161	TZCP	-	✓	시간 영역 비교	-	✓	✓	9	-	9-42
	162	TADD	-	✓	시간 추가	-	✓	✓	7	-	9-43
	163	TSUB	-	✓	시간 절감	-	✓	✓	7	-	9-44
	166	TRD	-	✓	시간 데이터 읽기	-	✓	✓	3	-	9-45
	167	TWR	-	✓	시간 데이터 기록	-	✓	✓	3	-	9-47
	169	HOURL	DHOURL	-	시간 미터	-	✓	✓	7	13	9-49
그레이 코드	170	GRY	DGRY	✓	BIN 을 그레이 코드로 변환	-	✓	✓	5	9	9-50
	171	GBIN	DGBIN	✓	그레이 코드를 BIN 으로 변환	-	✓	✓	5	9	9-51
	180	MAND	-	✓	행렬 AND	-	-	✓	9	-	9-52
	181	MOR	-	✓	행렬 OR	-	-	✓	9	-	9-54
	182	MXOR	-	✓	행렬 XOR	-	-	✓	9	-	9-55
	183	MXNR	-	✓	행렬 NOR	-	-	✓	9	-	9-56
	184	MINV	-	✓	행렬 전환	-	-	✓	7	-	9-57
행렬 관리	185	MCMP	-	✓	행렬 비교	-	-	✓	9	-	9-58
	186	MBRD	-	✓	행렬 비트 읽기	-	-	✓	7	-	9-60
	187	MBWR	-	✓	행렬 비트 쓰기	-	-	✓	7	-	9-62
	188	MBS	-	✓	행렬 비트 이송	-	-	✓	7	-	9-64
	189	MBR	-	✓	행렬 비트 회전	-	-	✓	7	-	9-66
	190	MBC	-	✓	행렬 비트 상태 카운트	-	-	✓	7	-	9-68

요약	API	기억용 코드		P 명령	기능	적용 모델			STEPS		페이지
		16 비트	32 비트			ES	SA	EH	16 비트	32 비트	
	<b>196</b>	HST	DHST	✓	고속 계수기	-	-	✓	3	-	9-69
접촉 종류 로직 작업	<b>215</b>	LD&	DLD&	-	S1 & S2 가 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-1
	<b>216</b>	LD	DLD	-	S1   S2 가 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-1
	<b>217</b>	LD^	DLD^	-	S1 ^ S2 가 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-1
	<b>218</b>	AND&	DAND&	-	S1 & S2 가 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-2
	<b>219</b>	AND	DAND	-	S1   S2 가 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-2
	<b>220</b>	AND^	DAND^	-	S1 ^ S2 가 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-2
	<b>221</b>	OR&	DOR&	-	S1 & S2 가 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-3
	<b>222</b>	OR	DOR	-	S1   S2 가 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-3
	<b>223</b>	OR^	DOR^	-	S1 ^ S2 정확 할 때 비교 접촉이 켜짐	-	✓	✓	5	9	10-3
접촉 종류 비교 명령	<b>224</b>	LD=	DLD=	-	S1 = S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	<b>225</b>	LD>	DLD>	-	S1 > S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	<b>226</b>	LD<	DLD<	-	S1 < S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	<b>228</b>	LD<>	DLD<>	-	S1 ≠ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	<b>229</b>	LD<=	DLD<=	-	S1 ≤ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	<b>230</b>	LD>=	DLD>=	-	S1 ≥ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	<b>232</b>	AND=	DAND=	-	S1 = S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	<b>233</b>	AND>	DAND>	-	S1 > S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	<b>234</b>	AND<	DAND<	-	S1 < S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	<b>236</b>	AND<>	DAND<>	-	S1 ≠ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	<b>237</b>	AND<=	DAND<=	-	S1 ≤ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	<b>238</b>	AND>=	DAND>=	-	S1 ≥ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	<b>240</b>	OR=	DOR=	-	S1 = S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	<b>241</b>	OR>	DOR>	-	S1 > S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	<b>242</b>	OR<	DOR<	-	S1 < S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6

요약	API	기억용 코드		P 명령	기능	적용 모델			STEPS		페이지
		16 비트	32 비트			ES	SA	EH	16 비트	32 비트	
	<b>244</b>	OR<>	DOR<>	-	S1 ≠ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	<b>245</b>	OR<=	DOR<=	-	S1 ≤ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	<b>246</b>	OR>=	DOR>=	-	S1 ≥ S2 가 정확 할 때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6

노트 1: 적용 모델 ES 시리즈는 이상은 EX 와 SS 시리즈를 포함 합니다; SA 는 SA 시리즈를 포괄 합니다.

노트 2: ES/EX/SS 모델을 위한 상위 명령들은 파동 실행 명령 (P command)를 갖추고 있지 않습니다.



## 5.2 응용 명령 구조

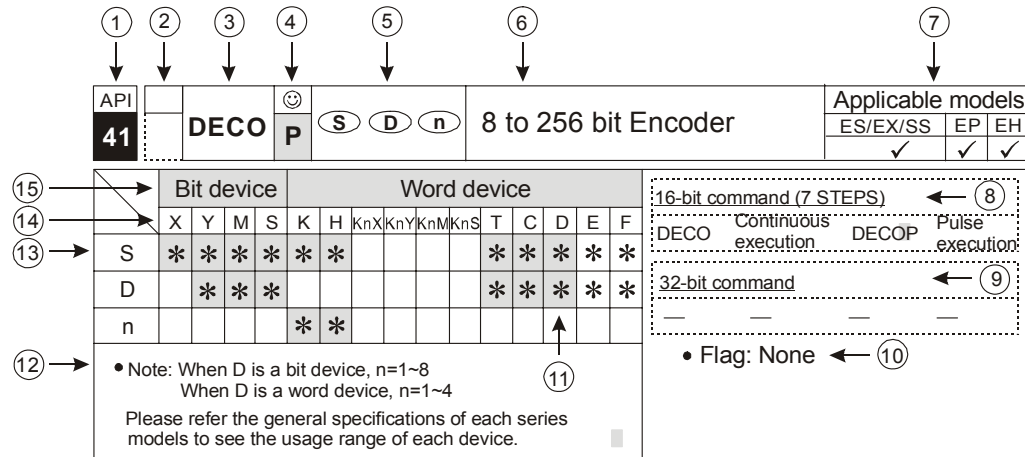
- ◆ 여러 가지 명령들은 아래와 같이 명령자 와 피연산자로 나뉘 수 있습니다:

명령자 : 명령의 독점 기능을 표시.

피연산자 : 피연산자를 계산하는 장치를 표시

명령자는 보통 한단계가 사용되게 하고 피연산자는 보통 명령이 16 비트나 32 비트 명령이냐에 따라 둘 혹은 네 단계까지 사용되게 합니다

- ◆ 적용 명령의 포맷의 설명:



① 적용 명령을 위한 API 번호

② 위줄 표시 16 비트 명령. 만약 줄의 가장자리가 점선 이라면, 16 비트 명령에서는 기능이 가능하지 않다는 뜻입니다.

아래 줄 표시 32 비트 명령. 만약 줄의 가장자리가 점선이라면, 32 비트에서는 기능이 가능하지 않다는 뜻 입니다“ 는 32 비트 명령을 표시하기 위해 연상기호 코드 위에 첨가 됩니다. (예: API 12 DMOV)

③ 적용 명령의 연상기호 코드

④ 위 줄의 기호 “☺” 는 명령이 기본적으로 파동 실행 명령을 사용함으로써 적용 됨을 표시합니다  
아래 줄에 있는 “P”는 파동 실행 명령과 함께 사용 되는 명령을 표시 합니다 API 12 MOV(P)

⑤ 적용 명령의 피연산자 포맷

⑥ 적용 명령 기능의 요약

⑦ PLC DVP 시리즈 PLC DVP 의 적용 모델

⑧ 단계 번호는 16 비트 작업의 명령, 지속적인 실행 명령의 이름, 그리고 파동 실행 명령으로 채워졌습니다.

⑨ 단계 번호는 32 비트 작업의 명령, 지속적인 실행 명령의 이름, 그리고 파동 실행 명령으로 채워졌습니다.

⑩ 적용 명령의 관련 깃발

⑪ 기호 “\*” 는 순서 기록을 사용 할 수 있는 장치이다.

⑫ 노트

⑬ 기호 “\*” 피연산자를 위해 사용 될 수 있는 장치에 주어진다.

⑭ 장치 이름

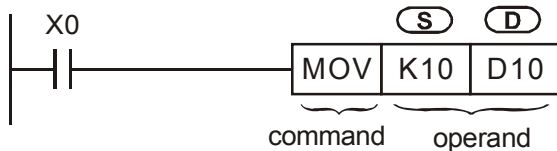
⑮ 장치 종류

## ◆ 적용 명령 입력

어떤 적용 명령들은 명령 코드 API 로만 이루어져 있습니다. 하지만 거의 대부분은 명령 코드 API 와 몇 개의 피연산자로 이루어져 있습니다.

DVP 시리즈 PLC 의 적용 명령은 명령 코드 API 00 에서 API 246 로 조정 됩니다. 각 명령 코드는 자체적인 의미를 가지고 있습니다. 예를 들면, API 12 는 MOV(이동 데이터)를 위해 존재합니다.

프로그램을 입력 시키기 위해 사다리 도표를 사용할 때, 우리는 API 명령 코드를 입력 시켜야 할 것입니다. 각 적용 명령은 그에 따른 특정한 피연산자를 가지고 있습니다.



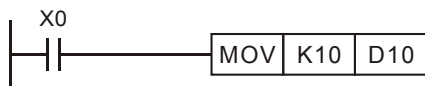
. 명령은 (S) 의 값 피연산자를 지정된 (D) 피연산자에게 이동 시킵니다.

(S)	요소적 피연산자: 만약 1 이상의 요소적 피연산자가 있다면, 우리는 디스플레이 용으로 (S1), (S2) 를 사용합니다.
(D)	목적지 피연산자: 만약 하나 이상의 피연산자가 존재 한다면, 우리는 디스플레이 용으로 (D1), (D2) 를 사용 합니다..
만약 피연산자가 일정 K,H 또는 기록만을 표출 할 수 있다면, 우리는 (m), (m1), (m2), (n), (n1), (n2) 를 디스플레이 용으로 사용합니다.	

## ◆ 피연산자의 길이 (16-비트 혹은 32-비트 명령)

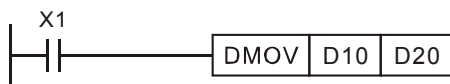
피연산자의 길이가 두 그룹으로 나눌 수 있습니다. 16 비트와 32 비트는 다른 길이의 데이터를 진행 하기 위함 입니다.명령이 32 비트에서 16 비트로 나뉘어 지기 전의 명령 “D”

16 비트 MOV 명령



☞ X0=On 일 때, K10 은 D10 으로 보내  
졌습니다..

32 비트 DMOV 명령



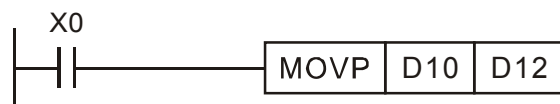
☞ X1=On 일 때, (D11, D10)의 데이터가 (D21,  
D20) 로 보내 졌습니다.

## ◆ 지속적인 실행 명령과 파동 실행 명령

명령의 실행 종류는 두 종류로 나눌 수 있습니다: 지속적인 실행 명령과 파동 실행 명령이 그들 입니다. 실행 시간이 짧아서 명령이 실행되지 않았을 때, 프로그래밍의 스캔 사이클을 줄이기 위해 가능한 한

과동 실행 명령을 사용 하십시오 “P”는 과동 실행 명령 후 직접적으로 더해지기 위함 입니다. 주로 사용되는 명령은 보통 적용을 위해 과동 실행 명령을 사용 합니다. 예, INC, DEC 그리고 MOV 등등. 관련 명령. 그러므로, 과동 실행 명령이 기호 ㉔로 명령 오른쪽 상단에 인식 됩니다.

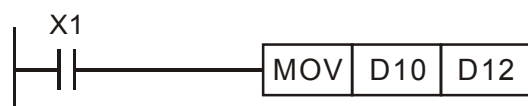
#### 과동 실행 명령



☞ X0 가 OFF→ON 으로 전환 될 때, the MOV

명령은 한번 실행되고, 명령은 프로그램의 스캔에서 재실행 되지 않습니다. 이를 과동 실행 명령이라 합니다.

#### 지속적인 실행 명령



☞ X1=ON 일 때, the MOV 명령은 프로그램의 모든

스캔에서 재 실행 될 수 있습니다. 이를 지속적인 실행 명령이라고 합니다.

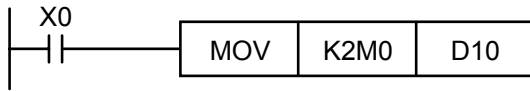
☞ 위 그림은 X0,X1=꺼짐 일 때, 명령이 실행 되지 않는 것과, 대상 피연산자 “D”의 순서가 바뀌없이 제 자리에 있음을 보여 줍니다.

#### ◆ 피연산자의 할당 장치

1. X,Y,M,S 와 같은 비트 장치는 함께 이루어져 있고, WORD 장치로 정의 됩니다. 적용 명령에서는, 비트 장치가 구동을 위해 수 값을 저장 하기 위한 워드 장치 (KnX, KnY, KnM, KnS)로써 구동 합니다.
2. 데이터 기록 D, 타이머 T, 카운터 C 그리고 순서 기록 E,F 는 모두 피연산자의 장치에 할당 되어 있습니다.
3. 데이터 기록은 보통 16 비트 기록 이자, 기록 D 이기도 합니다. 그런데 32 비트를 할당하는 것 역시 두개의 기록 D 들을 지속적인 수와 함께 할당 하는것과 같은 것을 뜻합니다.
4. 만약 32 비트 명령의 피연산자가 D0 를 할당한다면, D1 과 D0 으로 이루어진 32 비트 데이터 기록은 채워 질 것입니다. D1 은 16 비트 보다 높고, D0 는 16 비트 보다 낮습니다. 타이머 T 의 룰을 사용하는것과 16 비트 카운터 (C0-C199)는 같습니다.
5. 32 비트 카운트 (C200-C256)가 데이터 기록으로 쓰여 질 때, 한 포인트가 32 비트 길이를 표시 합니다. 32 비트 명령의 피연산자만이 지정 되고, 16 비트 명령의 피연산자는 지정 될수 없습니다.

#### ◆ 피연산자 데이터 포맷

1. X,Y,M,S 은 싱글 포인트 켜짐/꺼짐이 됩니다. 이들이 비트 장치로 정의 됩니다.
2. 그런데, 16 비트 (또는 32 비트) 장치 T,C,D,E,F 는 데이터 기록과 워드 장치로 정의 됩니다.
3. 우리는 X,Y,M,S 앞에 워드 장치로 정의 되는 Kn 첨가 할 수 있습니다. 한데, n=1 은 4 비트를 의미 합니다. 그리고 16 비트는 K1 에서 K4 로 표현 될 수 있습니다. 32 비트는 K1 에서 K8 으로 표현 될 수 있습니다. 예를 들어서, K2M0 은 M0 부터 M7 까지 8 비트가 있음을 뜻합니다.



☞ X0=On 일 때,. M0 에서 M7 의 순서를 D10 선분 0 에서 7 로 이동.후 선분 8 에서 15 는 0 으로 설정 됩니다.

## ◆ 자리수의 특정 번호

16-비트 명령	
자리수의 특정 번호 (16- 비트 명령): K-32,768~K+32,767	
16- 비트 명령: (K1~K4)	
K1 (4 포인트)	0~15
K2 (8 포인트)	0~255
K3 (12 포인트)	0~4,095
K4 (16 points)	-32,768~+32,767

32-비트 명령	
자리수의 특정 번호 (32- 비트 명령): K-2,147,483,648~K+2,147,483,647	
32- 비트 명령 (K1~K8)	
K1 (4 포인트)	0~15
K2 (8 포인트)	0~255
K3 (12 포인트)	0~4,095
K4 (16 포인트)	0~65,535
K5 (20 포인트)	0~1,048,575
K6 (24 포인트)	0~167,772,165
K7 (28 포인트)	0~268,435,455
K8 (32 포인트)	-2,147,483,648~+2,147,483,647

## ◆ 깃발

### 1. 일반 깃발

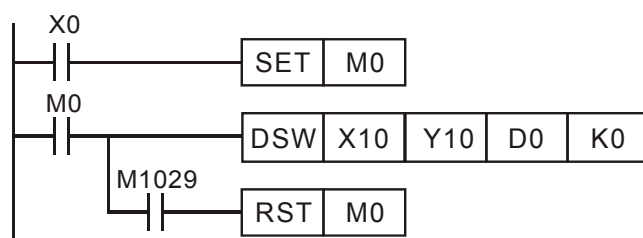
- PLC 적용 명령의 작업 결과를 위해, DVP 시리즈의 아래 깃발들이 있습니다.

예 : M1020 : 제로 깃발      102 : 운송 깃발      M1021 : 대여 깃발  
M1029 : 명령 실행 완료 깃발

명령을 실행할 때, 모든 깃발 들은 적용 명령의 작업 결과로 인해 켜지거나 꺼져 있어야 합니다. 그런데, 위에 깃발들은 특정 명령 뿐만이 아닌 여러 명령 들과 연결되어 있음을 알아 두셔야 합니다.

### 2. 명령 실행 완료 깃발의 프로그램 예시 , 예.M1029

조건 접촉이 켜져 있을 때, 디지털 스위치 입력 명령 (DSW)은 4 출력 포인트를 0.1초 주파수와 DSW 의 값을 읽기위해 자동적으로 순서대로 운행 할 것 입니다. 작업의 중반주에, 조격 접촉이 꺼진다면, DSW 명령은 제어 되고, 위에서 언급한 명령이 프로그램 순환의 시작에서 재 실행 될 것이된다. 다시 조건 접촉이 켜질 때, 장애를 막기 위해 아래 순회를 참조 하십시오.



☞ X0=ON 일 때, DSW 명령이 활성화 됩니다.

☞ X0=OFF 일 때, DSW 명령의 프로그램 순회가

종료 될 때까지 대기 하십시오. M1029=ON  
이후 M0 는 꺼질 것입니다.

### 3. 에러 작업 것발

적용 명령의 조합이 에러 혹은 피연산자의 지정 장치가 영역 밖에 있다면, 에러가 발생 하고, 에러 것발과 아래 테이블의 번호들이 적용 명령의 실행 중 나타나게 될 것 입니다.

M1067 D1067 D1069	오류 작업 발생시, M1067=On, D1067 가 오류번호를 보이고, D1069 는 오류 주소를 나타 낼 것 입니다. 만약 다른 오류가 발생 한다면, D1067 과 D1069 이 순서가 재생 될 것 입니다. (오류가 재설져○ 됐을 때, M1067=Off 입니다)
M1068 D1068	오류 작업 발생시, M1068=On, D1068 은 오류 주소를 나타 낼 것입니다. 다른 오류 발생시, D1068 의 순서는 재생 되지 않을 것 입니다. M1068 은 RST 명령을 OFF 로 재설정 하기 위해 사용 되어야 합니다. 계속 진행 시, 오류는 지속적으로 남아 있을 것 입니다.

### 4. 연장 기능을 위한 것발

어떤 적용 명령은 특수 것발을 사용 함으로써 기능을 연장 시킬 수 있습니다.

예: RS 명령은 M1161 을 사용해 RS 스위치 변속 모드 8 비트와 16 비트로 전환 할 수 있습니다.

#### ◆ 실행 명령을 위한 한정된 사용 시간

어떤 명령들은 프로그램에서 자주 사용 될 수 있습니다. 하지만 몇 몇은 프로그램에서 겨우 두번 내지 한번 밖에 사용 되지 못합니다. 그런데, 이 명령들은 피연산자에서 더 많은 명령의 기능을 연장 하는 순서 기록으로 조정 될 수 있습니다.

1. 프로그램에서 한번만 사용 될 수 있습니다:

API 58 (PWM) (ES/EX/SS/SA/SX/SC models)	API 60 (IST) (ES/EX/SS/SA/SX/SC/EH models)
API 74 (SEGL) (ES/EX/SS models)	

2. 프로그램에서 두번만 사용 될 수 있습니다.

API 57 (PLSY) (ES/EX/SS models)	API 59 (PLSR) (ES/EX/SS models)
API 74 (SEGL) (EH models)	API 77 (PR) (SA/EH models)

3.:프로그램에서 네번만 사용 될 수 있습니다.

API 169 (HOUR) (SA models)
----------------------------

4. 프로그램에서 8 번만 사용 될 수 있습니다.

API 64 (TTMR) (SA models)
---------------------------

5. API 53 (DHSCS) 와 API 54 (DHSCR), 이 명령들은 4 번 이하로 동시에 프로그램에서 실행 될수 있습니다.

6. API 53 (DHSCS), API 54 (DHSCR), API 55(DHSZ). 이 명령들은 6 번 이하로 동시에 DVP-SA 모델의 프로그램에서 실행 될 수 있습니다.

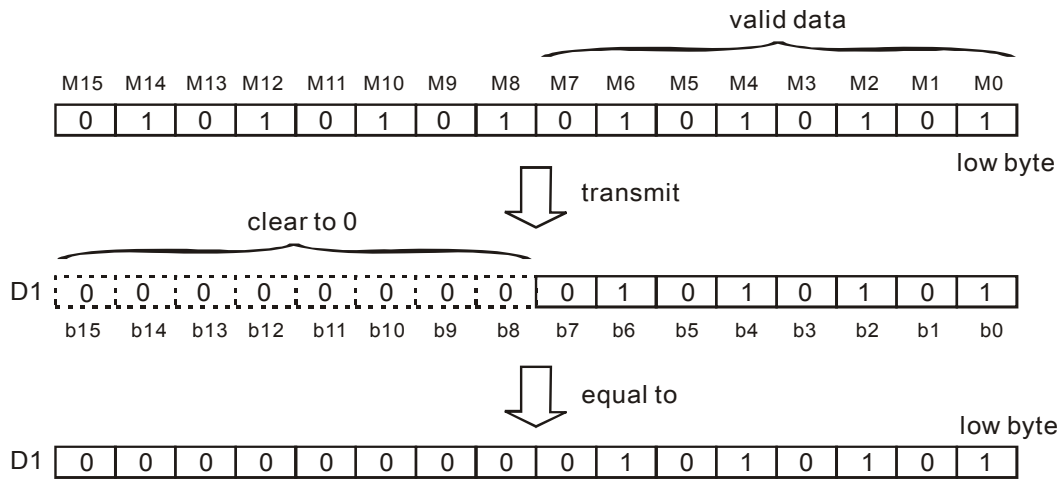
### ◆ 동시 실행 명령을 위한 제한 사용 시간

프로그램에서 같은 명령을 실행하기 위한 시간 제한은 존재 하지 않습니다. 그런데 이들이 동시에 사용 될 경우에는, 시간 제한이 존재 합니다.

1. API 52 (MTR), API 56 (SPD), API 62 (ABSD), API 63 (INCD), API 69 (SORT), API 70 (TKY), API 71 (HKY), API 72 (DSW) (SA 모델), API 74 (SEGL)(SA 모델), API 75 (ARWS), API 80 (RS), API 100 (MODRD), API 101 (MODWR), API 102 (FWD), API 103 (REV), API 104 (STOP), API 105 (RDST), API 106 (RSTEF), API 150 (MODRW), API 151 (PWD), 이 명령들은 프로그램에서 한번만 동시에 사용 될 수 있습니다.
2. API 57 (PLSY), API 58 (PWM), API 59 (PLSR), API 72 (DSW) (EH models), 이 명령들은 프로그램에서 두번만 동시에 사용 될 수 있습니다.
3. API 169 (HOUR) (EH 모델) 이 명령들은 프로그램에서 네번만 동시에 사용 될 수 있습니다.
4. API 64 (TTMR) (EH 모델) 이 명령들은 프로그램에서 여덟번만 동시에 사용 될 수 있습니다.
5. DVP EH 모델의 프로그램에서, DHSCS, DHSCR 과 DHSZ 와 같은 명령과 연관된 하드웨어 고속 카운터의 시간 제한은 없습니다. 그런데 이들을 동시에 사용 할 시 시간 제한은 존재 합니다. DHSCS, DHSCR 명령은 하나의 메모리 단위를 사용 하고 DHSZ 는 두개의 메모리 단위를 사용 할 것 입니다. 이들이 동시에 실행 될 때, 총 사용 되는 메모리 단위는 여덟 메모리 단위를 초과 할 수 없습니다. 초과시에는, 시스템이 가장 먼저 스캔되거나 실행되며 기타 무시되는 명령들의 사용되는 메모리 단위를 합산 하지 못할 것 입니다.

## 5.3 숫자 값의 처리

- ◆ X,Y,M,S 와 같은 장치는 비트 장치 이고 두 가지의 상태 켜짐과 꺼짐을 갖추고 있습니다. 그런데, T,C,D,E,F 는 데이터 기록이고 워드 장치로 정의 됩니다. 비록 비트 장치가 싱글 포인트 켜짐/꺼짐 이지만, 앞에 특정 장치를 장착 할 시, 적용 명령의 피연산자에서 수 값을 사용 할 수 있습니다. “특정 비트 장치”는 “특정 자리의 수” 이고 “n” 0 에서 8 의 영역으로 지정되는 Kn 과 비슷하게 보일 것 입니다.
- ◆ 16 비트는 K1 에서 K4 로 표현 될 수 있으며 32 비트는 K1 에서 K8 으로 표현 될 수 있습니다. 예를 들면, K2M0 는 M0 에서 M7 까지 에서 8 개의 비트가 있음을 뜻합니다.



- ◆ K1M0, K2M0, K3M0 를 16 비트 기록으로 발송하고 부족한 상위 비트 데이터는 발송 되지 않았습니다. 32 비트 기록에 K1M0, K2M0, K3M0, K4M0, K5M0, K6M0, K7M0 발송 하는것과 같고 부족한 상위 비트는 역시 발송 되지 않았습니다.
- ◆ 만약 피연산자의 순서가 16 비트 작업에서 K1 에서 K2 로 할당 되거나 K4 에서 K7 으로 32 비트 작업에서 할당 된다면, 부족한 상위 비트는 0 으로 요약 될 것 입니다. 그러므로 작업 결과는 정상임을 뜻합니다.



- ◆ 비트 장치의 수는 자유롭게 특정화 할 수 있습니다. 그런데 X와 Y 장치의 가장 낮은 자리수의 소수점 자리에서 0 을 사용 하는 것을 권장 하바다. (X0,X10,X20...Y0,Y10,Y20) M 과 S 시리즈에서는, 멀티플 8 을 쓰는 것을 권장하지만, 보통의 경우 0 을 가장 많이 쓸 것을 권장합니다.
- ◆ 지속적인 수 할당 하기

예를 들어, D 날짜 기록 처럼, D 의 지속적 번호는 D0,D1,D2,D3,D4, 등등 입니다.

For the bit device, the continuous numbers are shown as follows:

K1X0	K1X4	K1X10	K1X14.....
K2Y0	K2Y10	K2Y20	Y2X30.....
K3M0	K3M12	K3M24	K3M36.....
K4S0	K4S16	K4S32	K4S48.....

비트 장치 수는 모두 위에 표시 된거와 같습니다. 오류를 피하기 위해선, 지속 번호를 넘는 번호들을 무시 하십시오. 나아가서, K4Y0 32 비트 작업으로 사용 될 때, 상위 16 비트는 0 으로 정리 됩니다. 그러므로 32 비트 작업에서 K8Y0 을 사용 할 것을 권장 합니다.

- ◆ 부동 포인트 작업

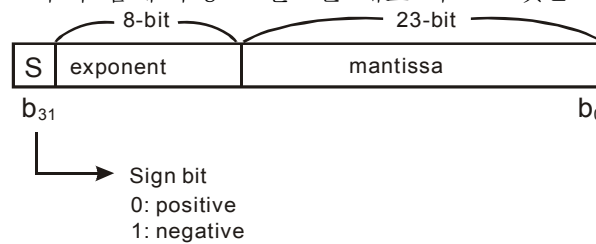
DVP 시리즈 PLC 의 내부 작업은 보통 'BIN 정수' 포맷으로 구동 됩니다. 정수 나누기 작어를 할때, 소수점은 버려 집니다. 예:  $40 \div 3 = 13$  일때 남는 수는 1 이고 소수점은 버려 집니다. 하지만 만약 부동

포인트 작업을 사용 한다면, 소수점은 주어질 수 있습니다.

API 49 (FLT),	API 110 (D ECMP),	API 111 (D EZCP),	API 116 (D RAD),
API 117 (D DEG),	API 118 (D EBCD),	API 119 (D EBIN),	API 120 (D EADD),
API 121 (D ESUB),	API 122 (D EMUL),	API 123 (D EDIV),	API 124 (D EXP),
API 125 (D LN),	API 126 (D LOG),	API 127 (D ESQR),	API 128 (D POW)
API 129 (INT)	API 130 (D SIN)	API 131 (D COS)	API 132 (D TAN)
API 133 (D ASIN)	API 134 (D ACOS)	API 135 (D ATAN)	API 136 (D SINH)
API 137 (D COSH)	API 138 (D TANH)		

## 이진 부동 포인트

DVP PLC 은 IEEE754 로 32 비트 수와 함께 부동 포인트를 대표 하고 포맷은 다음과 같습니다:



$$\text{균등화} (-1)^S \times 2^{E-B} \times 1.M; B=127$$

23 그러므로 32 비트 부동의 영역은  $\pm 2^{-126}$  to  $\pm 2^{+128}$  입니다. 예:  $\pm 1.1755 \times 10^{-38}$  에서  $\pm 3.4028 \times 10^{+38}$  까지.

예 1: 소수점 23 을 대표하기 위한 32 비트 부동 포인트 사용 합니다.

1 단계: 23 을 이진 수로 전환: 23.0:10111

2 단계: 이진을 평균화:  $10111 = 1.0111 \times 2^4$ , 0111 은 가수 이고 4 는 지수이다.

3 단계: 지수 얻어내기:  $\because E-B=4 \rightarrow E-127=4 \therefore E=131=10000011_2$

4 단계: 우리는 이제 사인, 가수 그리고 평균화된 지수를 합쳐 이진 IEEE 짧은 실제 대표수로 전환 할 수 있습니다.

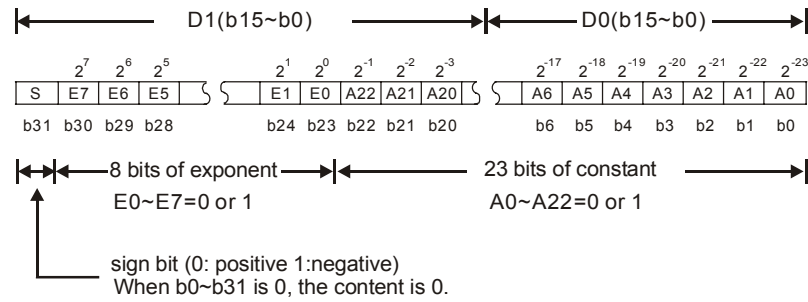
$$0 \ 10000011 \ 011100000000000000000000_2 = 41B80000_{16}$$

예 2: 소수점 -23 을 대표하기 위한 32 비트 부동 포인트 사용 합니다.

전환 과정은 소수점 23 하고 같습니다. 필요한건  $1 \ 10000011 \ 011100000000000000000000_2 = C1B80000_{16}$  값을

얻기위해 사인 비트를 0 에서 1 로 조정하는 것입니다. DVP PLC 도 지속적인 수와 함께 이진 부동 포인트를 저장 하기 위해 두 기록을 사용 합니다. 이진 부동 포인트를 저장하기 위한 기록(D1,D0) 사용의 예는 다음과 같습니다.





### 소수 부동 포인트

- ◆ 이진 부동 포인트는 보통 사람들은 이해하지 못합니다. 그러므로, 이진 부동 포인트 포맷을 사람들이 소수 작업을 하기위한 소수 부동 포인트를 전환이 가능합니다. 그런데 DVP 시리즈 PLC 은 소수 작업을 하기 위해 이진 부동 포인트를 사용 합니다.
- ◆ 소수 부동 포인트는 2 개의 지속적인 수와 함께 기록에 저장 되어 있습니다. 일정을 저장한 작은수가 딸린 기록 와 지수를 저장한 큰 수가 딸린 기록.

예를 들어, 소수 부동 포인트를 저장 하기위해 기록(D1,D0)을 사용 합니다.

소수 부동 포인트 = [일정 D0] X 10<sup>[지수 D1]</sup>

일정 D0 = ±1,000~±9,999

지수 D1 = - 41~+35

(D1, D0)의 가장 특수 비트는 기호 비트 입니다.

대신에, 일정 100 은  $1,000 \times 10^{-1}$ .과함께 보여짐에 따라 D0 에서 존재 하지 않습니다. 소수의 영역은  $\pm 1175 \times 10^{-41}$  에서  $\pm 3402 \times 10^{+35}$ . 까지 입니다.

- ◆ 소수 부동 포인트는 다음 명령에 쓰일 수 있습니다.

이진 부동 포인트를 위한 변환 명령 (D EBCD)

소수 부동 포인트를 위한 변환 명령 → 이진 부동 포인트(D EBIN)

- ◆ 제로 깃발 (M1020), 대여 깃발 (M1021) 과 운송 깃발 (M1022)

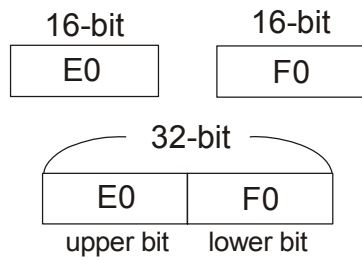
부동 명령에 사용하는 깃발들은:

- 제로 깃발:결과가 0 일 때, M1020=On.
- 대여 깃발:결과가 최소 단위보다 적을 때, M1021=On
- 운송 깃발:결과 사용 영역 초과와 절대 값 일 때, M1022=On

### 5.4 순서 기록 E, F

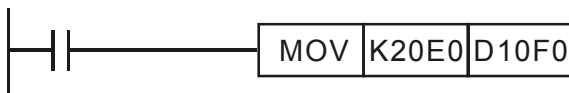
순서 기록은 16-비트 기록 입니다. ES/EX/SS 모델 (E and F)를 위한 2 개의 장치, SA 모델 (E0~E3, F0~F3)을 위한 8 개의 장치 그리고 EH 모델 (E0~E7, F0~F7)을 위한 16 개의 장치가 있습니다.

E 와 F 역시 일반 기록과 똑같은 16 비트 기록 입니다.  
쓰기/읽기가 자유롭습니다.



32-비트 기록을 사용 할때, E 를 특정화 하셔야 합니다.  
이런 조건에서는, F 는 E 로 카바 되고 더이상 사용 할 수 없게 됩니다; 그러지 않으면 E 의 순서는 틀리게 될 것입니다. (PLC 가 시작할때, F 의 순서를 처리하고 0 으로 재설정 하기 위해 **MOVP** 명령을 사용 하실 것을 권장 합니다.)

32 비트 순서 기록을 사용 할때, E, F 의 합은 다음과 같습니다. (E0, F0), (E1, F1), (E2, F2)...(E7, F7).



왼쪽 그림이 보여 주듯이, 피연산자의 순서는 E,F 의 순서에 따라 바뀌고 우리는 이러한 조정을 “순서” 라고 이름 지었 습니다.

E0=8 F0=14  
20+8=28 10+14=24  
K28 → D24 transmit

For example 예를 들어, E0=8 와 K20E0 는 일정 K28(20+8)를 대표합니다. 접속 되어 있을때, 일정 K28 는 D24 기록으로 발송 될 것입니다.

ES/EX/SS 시리즈에서 조정을 위해 순서 기록을 사용 할 수 있는 장치 들은: P, X, Y, M, S, KnX, KnY, KnM, KnS, T, C, D.

SA 시리즈에서 조정을 위해 순서 기록을 사용 할 수 있는 장치 들은: P, X, Y, M, S, KnX, KnY, KnM, KnS, T, C, D

EH 시리즈에서 조정을 위해 순서 기록을 사용 할 수 있는 장치들은: P, I, X, Y, M, S, K, H, KnX, KnY, KnM, KnS, T, C, D

위 장치들은 조정을 위한 순서 기록 E,F 를 사용 할 수 있습니다. 그런데 순서 기록 E,F 는 스스로 혹은 Kn. (K4M0E0 사용가능, K0E0M0 사용불가).를 조정 할수 없습니다. 각 적용 명령 마다, 기호“\*”가 피연산자의 테이블에 첨가 됐다면, 장치가 조정을 위한 순서 기록 E,F 를 사용 할 수 있음을 의미 합니다.

순서 기록 E,F 는 P, I, X, Y, M, S, KnX, KnY, KnM, KnS, T, C, D 장치들을 특정 조건으로 조정하기 위하여 사용 될 수 있습니다. 두 장치, E 혹은 F 는 16 비트 기록을 사용 할때 특정화 될수 있습니다. 만약 E,F 순서 기록을 일정 K,H 를 32 비트 명령에서 조정하는데 사용 한다면, 오직 한 장치, E 만이 특정화가 가능 합니다.

일정 (K,H)가 WPLSoft 명령 모드에서 순서 기능을 위해 사용 된다면, “@”기호를 사용 해야 합니다.

예: "MOV K10@E0 D0F0"

## 5.5 명령의 순서

## • 기호 별로 정리

구분	API	기억용 코드		P 명령	기능	적용 모델			단계		페이지
		16 비트	32 비트			ES	SA	EH	16-비트	32-비트	
A	87	ABS	DABS	✓	절대 값	✓	✓	✓	3	5	7-100
	62	ABSD	DABSD	-	절대 드럼 제어 장치	-	✓	✓	9	17	7-44
	155	ABSR	DABSR	-	ABS 전류 값 읽기	-	-	✓	7	13	9-14
	20	ADD	DADD	✓	BIN 데이터의 추가 적용	✓	✓	✓	7	13	6-33
	66	ALT	-	-	On/Off 교체 명령	✓	✓	✓	3	-	7-51
	218	AND&	DAND&	-	S1 & S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-2
	220	AND^	DAND^	-	S1 ^ S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-2
	219	AND	DAND	-	S1   S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-2
	234	AND<	DAND<	-	S1 < S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	237	AND<=	DAND<=	-	S1 ≤ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	236	AND<>	DAND<>	-	S1 ≠ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	232	AND=	DAND=	-	S1 = S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	233	AND>	DAND>	-	S1 > S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	238	AND>=	DAND>=	-	S1 ≥ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-5
	93	ANDF	-	-	에지 하강 감지 작업을 위한 시리즈 연결 명령	✓	✓	✓	3	-	3-12
	92	ANDP	-	-	에지 승강 방지 작업을 위한 시리즈 연결 명령	✓	✓	✓	3	-	3-12
	47	ANR	-	✓	알람 장치 재설정	-	✓	✓	1	-	6-70
	46	ANS	-	-	알람 장치 출력	-	✓	✓	7	-	6-69
	75	ARWS	-	-	화살표 키보드 입력	-	✓	✓	9	-	7-66
	76	ASC	-	-	ASCII 코드 변환	-	✓	✓	11	-	7-67
	82	ASCII	-	✓	HEX 에서 ASCII 로 전환	✓	✓	✓	7	-	7-88
B	133	-	DASIN	✓	이진 부동 포인트의 호 사인 작업	-	✓	✓	-	9	8-44
	134	-	DACOS	✓	이진 부동 포인트의 호 사인 작업	-	✓	✓	-	9	8-45
	135	-	DATAN	✓	이진 부동 포인트의 호 접선 작업	-	✓	✓	-	9	8-46
	18	BCD	DBCD	✓	데이터를 BCD 로 전환	✓	✓	✓	5	9	6-30
C	19	BIN	DBIN	✓	BCD 데이터를 BIN 으로 전환	✓	✓	✓	5	9	6-31
	15	BMOV	-	✓	블럭 움직임	✓	✓	✓	7	-	6-25
	44	BON	DBON	✓	ON 비트를 결정	✓	✓	✓	7	13	6-67
	01	CALL	-	✓	서브 루틴 불러오기	✓	✓	✓	3	-	6-5
	84	CCD	-	✓	체크 코드	-	✓	✓	7	-	7-95

구분	API	기억용 코드		P 명령	기능	적용 모델			단계		페이지
		16 비트	32 비트			ES	SA	EH	16-비트	32-비트	
	00	CJ	-	✓	조건적 이동	✓	✓	✓	3	-	6-1
	14	CML	DCML	✓	증정	✓	✓	✓	5	9	6-23
C	10	CMP	DCMP	✓	비교	✓	✓	✓	7	13	6-18
	97	CNT	DCNT	-	계수기	✓	✓	✓	4	6	3-9
	131	-	DCOS	✓	이진 부동 포인트의 코사인 작업	✓	✓	✓	-	9	8-40
	137	-	DCOSH	✓	이진 부동 포인트의 과대 코사인 작업	-	✓	✓	-	9	8-48
	108	CRC	-	✓	CRC 에러 체크	-	✓	✓	7	-	8-17
D	25	DEC	DDEC	✓	1 의 절감 적용	✓	✓	✓	3	5	6-42
	41	DECO	-	✓	8 → 256 비트 디코더	✓	✓	✓	7	-	6-62
	143	DELAY	-	✓	명령 지연	-	✓	✓	3	-	8-50
	117	-	DDEG	✓	라디안 → 각도	-	✓	✓	-	9	8-24
	05	DI	-	-	장애를 무력화함	✓	✓	✓	1	-	6-8
	23	DIV	DDIV	✓	BIN 데이터의 나누기 적용	✓	✓	✓	7	13	6-39
	159	DRVA	DDRVA	-	데이터 백업 메모리 기록	-	-	✓	9	17	9-29
	158	DRVI	DDRVI	-	증가 실행	-	-	✓	9	17	9-24
E	72	DSW	-	-	디지털 스위치 입력	-	✓	✓	9	-	7-60
	120	-	DEADD	✓	이중 부동 포인트 추가	✓	✓	✓	-	13	8-28
	118	-	DEBCD	✓	이중 부동 포인트를 소수 부동 포인트로 전환	✓	✓	✓	-	9	8-25
	119	-	DEBIN	✓	소수 부동 포인트에서 이중 부동 포인트로 전환	✓	✓	✓	-	9	8-26
	110	-	DECMP	✓	이중 부동 포인트 비교	✓	✓	✓	-	13	8-21
	123	-	DEDIV	✓	이중 부동 포인트 나누기	✓	✓	✓	-	13	8-31
	04	EI	-	-	장애를 기능화 함	✓	✓	✓	1	-	6-8
	122	-	DEMUL	✓	이중 부동 포인트 곱하기	✓	✓	✓	-	13	8-30
	42	ENCO	-	✓	256 → 8 비트 인코드	✓	✓	✓	7	-	6-64
	127	-	DESQR	✓	이중 부동 포인트의 사각 루트	✓	✓	✓	-	9	8-35
	121	-	DESUB	✓	이중 부동 포인트 절감	✓	✓	✓	-	13	8-29
	124	-	DEXP	✓	지수 작업을 실행키 위한 이중 부동 포인트 전환	✓	✓	✓	-	9	8-32
	111	-	DEZCP	✓	이중 부동 포인트 영역 비교	✓	✓	✓	-	17	8-22
F	06	FEND	-	-	첫 마침	✓	✓	✓	1	-	6-12
	49	FLT	DFLT	✓	이중 부동 포인트로 BIN 정수를 전환	✓	✓	✓	5	9	6-73
	16	FMOV	DFMOV	✓	배수 장치 움직임	✓	✓	✓	7	13	6-27
	08	FOR	-	-	FOR-NEXT 폐회로의 시작	✓	✓	✓	3	-	6-15
	78	FROM	DFROM	✓	특수 모듈 CR 데이터 읽기	✓	✓	✓	9	17	7-70
	102	FWD	-	-	VFD-A 시리즈 드라이브 전진 명령	✓	✓	✓	7	-	8-10
	145	FTC	-	-	퍼지 온도 컨트롤	-	✓	✓	-	9	8-52

구분	API	기억용 코드		P 명령	기능	적용 모델			단계		페이지
		16 비트	32 비트			ES	SA	EH	16-비트	32-비트	
G	144	GPWM	-	-	일반 파동 넓이 조정 출력	-	✓	✓	-	7	8-51
	171	GBIN	DGBIN	✓	그레이 코드를 BIN 으로 전환	-	P	P	5	9	9-51
	170	GRY	DGRY	✓	BIN 을 그레이 코드로 전환	-	P	P	5	9	9-50
H	83	HEX	-	✓	ASCII 에서 HEX 로 전환	✓	P	P	7	-	7-92
	71	HKY	DHKY	-	16-키 키보드 입력	-	P	P	9	17	7-58
H	169	HOURL	DHOURL	-	시간 미터	-	P	P	7	13	9-49
	54	-	DHSCR	-	고속 계수기 비교 재설정	✓	P	P	-	13	7-14
	53	-	DHSCS	-	고속 계수기 비교 설정	✓	✓	✓	-	13	7-5
	196	HST	DHST	✓	고속 계수기	-	-	✓	3	-	9-68
	55	-	DHSZ	-	영역 비교 (고속 계수기)	-	✓	✓	-	17	7-16
I	24	INC	DINC	✓	1 의 추가 실행	✓	✓	✓	3	5	6-41
	63	INCD	-	-	증가 드럼 제어장치	-	✓	✓	9	-	7-46
	129	INT	DINT	✓	이진 부동 포인트를 BIN 정수로 변환	✓	✓	✓	5	9	8-37
	98	INV	-	-	작업 전환	✓	✓	✓	1	-	3-14
	03	IRET	-	-	장애 복귀	✓	✓	✓	1	-	6-8
	60	IST	-	-	수동/자동 콘트롤	✓	✓	✓	7	-	7-37
L	215	LD&	DLD&	-	S1 & S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-1
	217	LD^	DLD^	-	S1 ^ S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-1
	216	LD	DLD	-	S1   S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-1
	226	LD<	DLD<	-	S1 < S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	229	LD<=	DLD<=	-	S1 ≤ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	228	LD<>	DLD<>	-	S1 ≠ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	224	LD=	DLD=	-	S1 = S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	225	LD>	DLD>	-	S1 > S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	230	LD>=	DLD>=	-	S1 ≥ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-4
	91	LDF	-	-	에지 하강 감지 작업	✓	✓	✓	3	-	3-12
	90	LDP	-	-	에지 하강 감지 작업	✓	✓	✓	3	-	3-11
	125	-	DLN	✓	내추럴 대수 작업을 위해 이진 부동 포인트를 변환	✓	✓	✓	-	9	8-33
	126	-	DLOG	✓	대수 작업을 위해 이진 부동 포인트를 변환	✓	✓	✓	-	13	8-34
	107	LRC	-	✓	LRC 에러 확인	-	✓	✓	7	-	8-15
M	180	MAND	-	✓	행렬 AND	-	-	✓	9	-	9-52
	190	MBC	-	✓	행렬 비트 상태 카운트	-	-	✓	7	-	9-68
	189	MBR	-	✓	매트릭스 비트 회전	-	-	✓	7	-	9-66
	186	MBRD	-	✓	매트릭스 비트 읽기	-	-	✓	7	-	9-60

구분	API	기억용 코드		P 명령	기능	적용 모델			단계		페이지
		16 비트	32 비트			ES	SA	EH	16-비트	32-비트	
	188	MBS	-	✓	매트릭스 비트 이송	-	-	✓	7	-	9-64
	187	MBWR	-	✓	매트릭스 비트 쓰기	-	-	✓	7	-	9-62
	185	MCMP	-	✓	매트릭스 비교	-	-	✓	9	-	9-58
	45	MEAN	DMEAN	✓	평균 값	✓	✓	✓	7	13	6-68
	148	MEMR	DMEMR	✓	데이터 백업 메모리 읽기	-	✓	✓	7	13	8-59
M	149	MEMW	DMEMW	✓	메모리 기록	-	✓	✓	7	13	8-60
	184	MINV	-	✓	행렬 교체	-	-	✓	7	-	9-57
	183	MNOR	-	✓	행렬 NOR	-	-	✓	9	-	9-56
	100	MODRD	-	-	MODBUS 데이터 읽기	✓	✓	✓	7	-	8-1
	150	MODRW	-	-	MODBUS 데이터 읽기/기록	✓	✓	✓	11	-	9-1
	101	MODWR	-	-	MODBUS 데이터 기록	✓	✓	✓	7	-	8-5
	181	MOR	-	✓	행렬 OR	-	-	✓	9	-	9-54
	12	MOV	DMOV	✓	데이터 움직임	✓	✓	✓	5	9	6-20
	52	MTR	-	-	입력 행렬	-	✓	✓	9	-	7-3
	22	MUL	DMUL	✓	BIN 데이터의 곱하기 실행	✓	✓	✓	7	13	6-37
	182	MXOR	-	✓	행렬 XOR	-	-	✓	9	-	9-55
N	29	NEG	DNEG	✓	취소	✓	✓	✓	3	5	6-47
	09	NEXT	-	-	FOR-NEXT 폐회로의 종료	✓	✓	✓	1	-	6-15
O	221	OR&	DOR&	-	S1 & S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-3
	223	OR^	DOR^	-	S1 ^ S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-3
	222	OR	DOR	-	S1   S2 가 정확할때 비교 접촉이 켜짐	-	✓	✓	5	9	10-3
	242	OR<	DOR<	-	S1 < S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	245	OR<=	DOR<=	-	S1 ≤ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	244	OR<>	DOR<>	-	S1 ≠ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	240	OR=	DOR=	-	S1 = S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	241	OR>	DOR>	-	S1 > S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	246	OR>=	DOR>=	-	S1 ≥ S2 가 정확할때 비교 접촉이 켜짐	✓	✓	✓	5	9	10-6
	95	ORF	-	-	에지 하강 감지 작업을 위한 평행 연결 명령	✓	✓	✓	3	-	3-13
	94	ORP	-	-	상승 에지 감지 작업을 위한 평행 연결 명령	✓	✓	✓	3	-	3-13
P	88	PID	-	-	PID 계산	✓	✓	✓	9	-	7-101
	99	PLF	-	-	에지 하강 출력	✓	✓	✓	3	-	3-14
	89	PLS	-	-	에지 상승 출력	✓	✓	✓	3	-	3-13
	59	PLSR	DPLSR	-	가속/감속 속도와 함께한 파동류 출력	✓	✓	✓	9	17	7-33
	157	PLSV	DPLSV	-	변수 속도 파동 출력	-	-	✓	7	13	9-23
	57	PLSY	DPLSY	-	파동 출력	✓	✓	✓	7	13	7-24

구분	API	기억용 코드		P 명령	기능	적용 모델			단계		페이지
		16 비트	32 비트			ES	SA	EH	16-비트	32-비트	
	128	-	DPOW	✓	전원 작동을 위한 이진 부동 포인트 변환	✓	✓	✓	-	13	8-36
	77	PR	-	-	인쇄	-	✓	✓	5	-	7-68
	81	PRUN	DPRUN	✓	8 진법 시스템 변속	-	✓	✓	5	9	7-87
P	151	PWD	-	-	입력 파동 넓이 감지	-	-	✓	5	-	9-10
	58	PWM	-	-	파동 넓이 조정 출력	✓	✓	✓	7	-	7-30
R	116	-	DRAD	✓	각도 → 라디안	-	✓	✓	-	9	8-23
	67	RAMP	-	-	급 시그널	-	✓	✓	9	-	7-52
	154	RAND	-	-	랜덤 값	-	✓	✓	9	-	9-13
	33	RCL	DRCL	✓	운송 깃발 첨부된 상태에서 좌로 회전	✓	✓	✓	5	9	6-52
	32	RCR	DRCR	✓	운송 깃발 첨부된 상태에서 우로 회전	✓	✓	✓	5	9	6-51
	105	RDST	-	-	VFD-A 시리즈 드라이브 상태 읽기	✓	✓	✓	5	-	8-12
	50	REF	-	✓	I/O 재생	✓	✓	✓	5	-	7-1
	51	REFF	-	✓	재생 후 출력 필터의 응답 시간 조정	-	✓	✓	3	-	7-2
	103	REV	-	-	VFD-A 시리즈 드라이브 역 명령	✓	✓	✓	7	-	8-10
	31	ROL	DROL	✓	좌로 회전	✓	✓	✓	5	9	6-50
	30	ROR	DROR	✓	우로 회전	✓	✓	✓	5	9	6-49
	80	RS	-	-	시리얼 데이터 통신	✓	✓	✓	9	-	7-75
	106	RSTEF	-	-	VFD-A 시리즈 드라이브 이상 재설정	✓	✓	✓	5	-	8-14
	153	RTMD	-	-	I 장애의 실행 시간 조정을 위한 종료	-	-	✓	3	-	9-11
	152	RTMU	-	-	I 장애의 실행 시간 조정을 위한 시작	-	-	✓	5	-	9-11
S	73	SEGD	-	✓	7 단계 디스플레이 판 해석	✓	✓	✓	5	-	7-62
	74	SEGL	-	-	7 단계 디스플레이 스캔 출력	✓	✓	✓	7	-	7-63
	61	SER	DSER	✓	데이터 후입선출열 검색	-	✓	✓	9	17	7-43
	39	SFRD	-	✓	이송 기록 읽기	-	✓	✓	7	-	6-59
	35	SFTL	-	✓	특정 장치의 데이터를 좌측으로 이송	-	✓	✓	9	-	6-54
	34	SFTR	-	✓	특정 장치의 데이터를 우측으로 이동	✓	✓	✓	9	-	6-53
	38	SFWR	-	✓	기록 쓰기 이송	-	✓	✓	7	-	6-58
	130	-	DSIN	✓	이진 부동 포인트의 사인 작업	✓	✓	✓	-	9	8-38
	136	-	DSINH	✓	이진 부동 포인트의 광대 사인 작업	-	✓	✓	-	9	8-47
	13	SMOV	-	✓	이송 움직임	-	✓	✓	11	-	6-21
	69	SORT	-	-	데이터 정리	-	✓	✓	11	-	7-54
	56	SPD	-	-	속도 감지	-	✓	✓	7	-	7-22
	48	SQR	DSQR	✓	BIN의 사각 루트	✓	✓	✓	5	9	6-72

구분	API	기억용 코드		P 명령	기능	적용 모델			단계		페이지
		16 비트	32 비트			ES	SA	EH	16-비트	32-비트	
	02	SRET	-	-	서브 루틴 복귀	✓	✓	✓	1	-	6-6
	65	STMR	-	-	특수 타이머	-	✓	✓	7	-	7-50
	104	STOP	-	-	VFD-A 시리즈 드라이브 정지 명령	✓	✓	✓	7	-	8-10
	21	SUB	DSUB	✓	BIN 데이터의 절감 실행	✓	✓	✓	7	13	6-35
	43	SUM	DSUM	✓	ON 비트의 합계	✓	✓	✓	5	9	6-66
	147	SWAP	DSWAP	✓	고/저 바이트 교환	✓	✓	✓	3	5	8-58
S	109	SWRD	-	✓	디지털 스위치 읽기	-	✓	✓	3	-	8-20
T	162	TADD	-	✓	실제 시간 시계 데이터 추가	-	✓	✓	7	-	9-43
	132	-	DTAN	✓	이진 부동 포인트의 접선 작업	✓	✓	✓	-	9	8-42
	138	-	DTANH	✓	이진 부동 포인트의 광대 접선 작업	-	✓	✓	-	9	8-49
	160	TCMP	-	✓	시간 비교	-	✓	✓	11	-	9-41
	70	TKY	DTKY	-	10-키 키보드 입력	-	✓	✓	7	13	7-56
	96	TMR	-	-	타이머	✓	✓	✓	4	-	3-8
	79	TO	DTO	✓	특수 조정 CR 데이터 기록	✓	✓	✓	9	17	7-71
	166	TRD	-	✓	시간 데이터 읽기	-	✓	✓	3	-	9-45
	163	TSUB	-	✓	시간 절감	-	✓	✓	7	-	9-44
	64	TTMR	-	-	타이머 알람	-	✓	✓	5	-	7-48
	167	TWR	-	✓	시간 데이터 기록	-	✓	✓	3	-	9-47
	161	TZCP	-	✓	시간 영역 비교	-	✓	✓	9	-	9-42
V	85	VRRD	-	✓	분압기 읽기	-	✓	✓	5	-	7-97
	86	VRSC	-	✓	분압기 스케일	-	✓	✓	5	-	7-99
W	26	WAND	DAND	✓	회로 제품 (AND) 구동 실행	✓	✓	✓	7	13	6-43
	07	WDT	-	✓	감시 타이머 재생	✓	✓	✓	1	-	6-14
	27	WOR	DOR	✓	회로 합계 (OR) 작업 실행	✓	✓	✓	7	13	6-44
	37	WSFL	-	✓	기록을 좌측으로 이송	-	✓	✓	9	-	6-57
	36	WSFR	-	✓	기록을 우측으로 이송	-	✓	✓	9	-	6-55
	28	WXOR	DXOR	✓	독점적 회로 추가 (XOR) 작업 실행	✓	✓	✓	7	13	6-45
X	17	XCH	DXCH	✓	데이터 변환	✓	✓	✓	5	9	6-28
Z	11	ZCP	DZCP	✓	영역 비교	✓	✓	✓	9	17	6-19
	156	ZRN	DZRN	-	영점 복귀	-	-	✓	9	17	9-19
	40	ZRST	-	✓	특정 장치의 영역 재설정	✓	✓	✓	5	-	6-60

노트 1. 위의 ES 시리즈 적용 모델은 EX 와 SS 시리즈를 포함 합니다. SA 는 SA 시리즈를 포괄 합니다.

노트 2. ES/EX/SS 모델을 위한 위 명령들은 파동 실행 명령 (P 명령)을 갖추고 있지 않습니다.



API	연상기호		피연산자	기능	제어기						
					ES	EX	SS	SA	SX	SC	EH
00		CJ	P	<b>(S)</b>	조건부 점프						
연산부호		범위			프로그램 스텝						
<b>(S)</b>		P0~P255			CJ, CJP: 3 스텝						

중지							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S:** 조건부 점프의 목적지 포인터

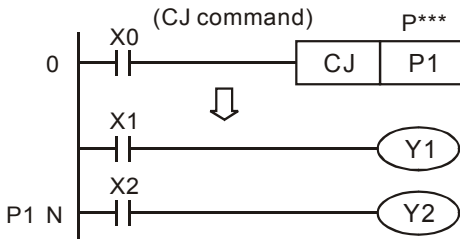
설명:

1. 피연산자 **S** 는 **P** 를 할당할 수 있습니다.
2. **P** 는 인덱스 레지스터 **E, F** 에 의해 수정될 수 있습니다.
3. **ES/EX/SS** 시리즈 모델: 피연산자 **S** 는 **P0~P63** 를 할당할 수 있습니다.
4. **SA/SX/SC/EH** 시리즈 모델: 피연산자 **S** 는 **P0~P255** 를 할당할 수 있습니다.
5. **ES/EX/SS** 시리즈 모델은 중지 실행 명령(**CJP**)을 지원하지 않습니다.
6. **CJ** 명령은 다음 조건에서 사용될 수 있습니다.:
  - a) 사용자가 **PLC** 프로그램의 일부 필요하지 않는 부분을 실행하길 원치 않을 때 프로그램 스캔 시간을 짧게 하기 위해서.
  - b) 두 배 혹은 이중의 코일 목적에서.
7. 포인터 **P** 에 의해 지시된 프로그램이 **CJ** 명령 전 일 때, **WDT**의 에러가 시간을 초과하는지 주의하십시오. 만일 **PLC** 정지가 실행된다면, 주의해서 사용하십시오.
8. **CJ** 명령은 반복적으로 동일한 포인터 **P** 를 할당할 수 있습니다. 하지만, **CJ** 명령과 **CALL** 명령은 동일한 포인터 **P** 를 할당할 수 없고, 만약 그렇지 않으면 에러가 발생할 것입니다.
9. **CJ** 명령이 실행될 때 각 장치의 설명.
  - a) 조건부 점프가 일어나기 전, **Y, M, S** 는 이전 상태로 남아있습니다.
  - b) 카운트를 실행하는 타이머 **10ms, 100ms** 는 정지할 것입니다.
  - c) 서브루틴 프로그램을 실행하는 타이머 **T192~T199** 는 계속 할 것이고, 출력 접점은 정상적으로 실행할 것입니다.
  - d) 카운트를 실행하는 고속 카운터는 계속 유지할 것이고 출력 접점은 정상적으로 실행할 것입니다.
  - e) 일반 카운터는 정지할 것입니다.
  - f) 만일 가산 타입 타이머의 리셋 명령이 조건부 점프가 있기 전에 동작한다면, 조건부 점프가 실행될 때 그 장치는 아직 리셋 상태로 남게 될 것입니다.
  - g) 일반적인 응용 명령들은 실행되지 않을 것입니다.
  - h) 실행하는 응용 명령들, 예를 들어, **API 53 DHSCS, API 54 DHSCR, API 55 DHSZ, API 56 SPD, API 57 PLSY, API 58 PWM, API 59 PLSR, API 157 PLSV, API 158 DRVI, API 159 DRVA** 는 계속 실행할 것입니다.

프로그램 예제 1:

1. **X0=On** 일 때, 프로그램은 어드레스 **0** 부터 **N (label P1)**까지 자동적으로 스킵하고 실행을 유지할 것입니다. 하지만 어드레스 **0** 과 **N** 사이 구역은 스킵 할 것이고 실행되지는 않을 것입니다.

2. X0=Off 일 때, 보통 프로그램은 어드레스 0 부터 실행을 유지할 것입니다. CJ 명령은 실행하지 않을 것입니다.



#### 프로그램 예제 2:

1. CJ 명령이 명령 MC 와 MCR 사이에서 실행될 수 있는 5 가지 상황이 있습니다.

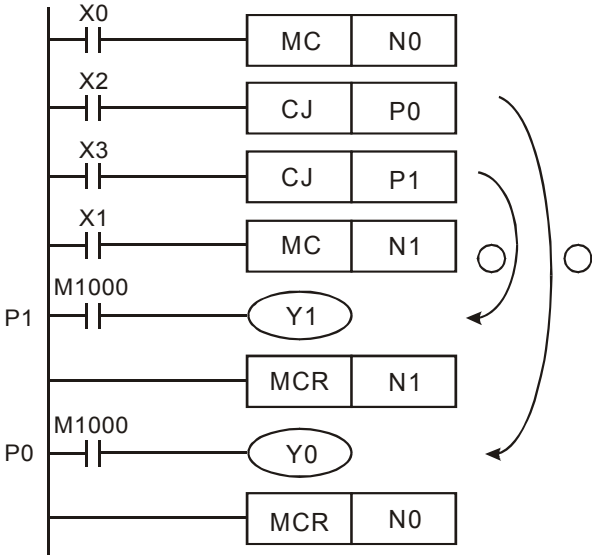
- MC~MCR 외.
- 다음 차트에서 루프 P1 에서 유효.
- 동일 레벨 N 에서, MC~MC 의 내부에.
- MC 의 내부에, MCR 의 외부에.
- 이 MC~MCR 에서부터 다른 MC~MCR 까지 점프. (1)

이 기능은 단지 V4.9 (포함) 혹은 ES/EX/SS 시리즈 모델과 SA/SX/SC/EH 시리즈 모델 보다 더 높은 버전에서 제공될 뿐입니다.

2. 그 실행은 V4.7(포함) 혹은 ES/EX/SS 시리즈 모델보다 낮은 버전을 설명합니다.

3. CJ 명령은 MC 와 MCR 명령사이에 사용됩니다. 하지만 그것은 MC~MCR 의 범위 밖이나 MC~MCR 내의 동일 레벨 N 에서 단지 사용될 뿐입니다. CJ 명령은 MC~MCR 의 범위부터 또 다른 MC~MCR 의 범위까지 점프하기 위해 사용될수 없습니다. 만일 그렇지 않다면 에러가 발생할 것 입니다. CJ 명령은 위의 언급된 조건 1 과 3 에서 정확하게 실행될 수 있습니다. 하지만 만일 다른 조건에서 사용된다면, 에러가 발생할 것입니다.

4. 사용자에게 의해 제어되는 대신 PLC 자체에 의해 제어되는 PLC 스택 내에 넣기 전에 MC 명령을 실행할 때, 스위치 (X0 다음 그림과 같음)의 상태; 그리고 명령이 실행되면, 스위치의 상태는 순서대로 스택의 가장 꼭대기부터 팝할 것입니다. 규칙적으로, 하나의 MC 명령은 하나의 MCR 명령에 일치할 것입니다.; 하지만 일단 상태가 위에서 처럼 b, d, e 로 나오면, 푸시와 팝은 규칙적인 순서로 서로 일치하지 않을 수 있습니다. 해당장애가 발생하면, 스택으로부터 팝의 값이 0 으로 끝나고 더 이상 팝이 나오지 않을 동안, 스택에서 푸시의 값은 8 레벨까지 가야 합니다. 그래서 CALL 혹은 CJ 명령을 사용할 때 PLC 스택에서 푸시와 팝과 관련한 명령에 주의하십시오.



### 프로그램 예제 3:

1. 각 장치의 상태는 다음과 같습니다.:

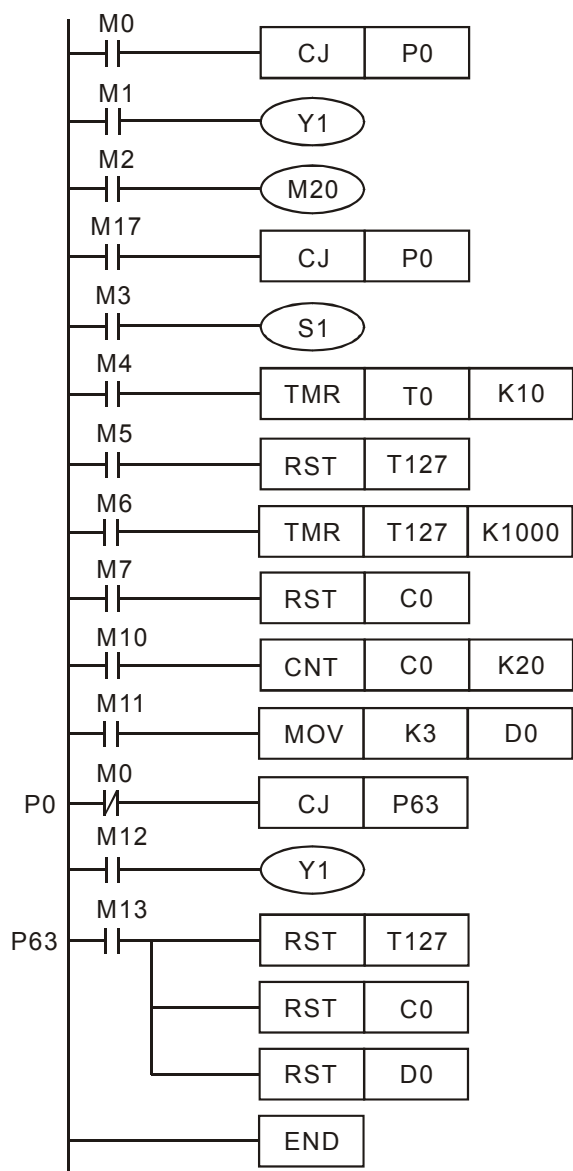
장치	CJ 실행전 접점상태	CJ 실행동안 접점 상태	실행동안 출력 코일 상태
Y, M, S	M1, M2, M3 Off	M1, M2, M3 Off→On	Y1 (주의 1), M20, S1 Off
	M1, M2, M3 On	M1, M2, M3 On→Off	Y1 (주의 1), M20, S1 On
10ms, 100ms 타이머 (ES/EX/SS/SA/S X/SC/EH)	M4 Off	M4 Off→On	타이머 T0 가 구동하지 않음
	M4 On	M4 On→Off	타이머 T0 는 즉시 멈추지만 래치됨, M0 는 On→Off, T0 는 0.
1ms, 10ms, 100ms 타이머 (가산용) SA /SX/SC/EH	M6 Off	M6 Off→On	타이머(T240)는 구동하지 않음
	M6 On	M6 On→Off	모든 가산 타이머가 정지하지만 일단 실행하면 래치됨. M0 가 On→Off 이면, T240 은 바뀌지 않음.
C0~C234	M7, M10 Off	M10 On/Off 트리거	카운터는 카운트하지 않음
	M7 Off, M10 On/Off 트리거	M10 On/Off 트리거	카운터의 인터럽트는 래치됨. M0 가 off 된 후에 카운트를 계속함.
응용 명령	M11 Off	M11 Off→On	응용 명령은 실행되지 않음.
	M11 On	M11 On→Off	스킵된 응용 명령을 실행하지 마십시오. 하지만 API 53~59, API 157~159 은 실행유지.

주의 1: Y1 은 이중 출력입니다. M0 가 Off 이면, 그것은 M1 에 의해 제어됩니다. M0 가 On 이면, 그것은 M12 에 의해 제어됩니다.

주의 2: 서브루틴이(SA/SX/SC/EH 의 T192~T199) 사용한 타이머가 CJ 명령을 실행하면, 카운트를 계속 할 것입니다. 타이머가 달성되면, 타이머의 출력 접점은 On 이 될 것입니다.

주의 3: 고속 카운터들이 (C235~C255) CJ 명령을 실행하면, 그것은 카운트를 계속 할 것이고 출력 포인트는 또한 구동을 계속 할 것입니다.

2. Y1 은 두배 혹은 이중의 코일 목적지입니다. M0=Off 이면, 그것은 M1 에 의해 제어됩니다. M0=On 이면, 그것은 에 의해 제어됩니다.



API	연상기호			피연산자	기능	제어기													
01		CALL	P	<b>(S)</b>	호출 서브루틴	ES	EX	SS	SA	SX	SC	EH							
연산부호		범위				프로그램 스텝													
<b>(S)</b>		P0~P255				CALL, CALLP: 3 스텝													
						펄스				16-비트				32-비트					
						ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S:** 호출 서브루틴의 목적지 포인터.

설명:

1. 피연산자 **S** 는 **P** 를 할당 할 수 있습니다.
2. **P** 는 인덱스 레지스터 **E, F** 에 의해 수정될 수 있습니다.
3. **ES/EX/SS** 시리즈 모델: 피연산자 **S** 는 **P0~P63** 를 할당 할 수 있습니다..
4. **SA/SX/SC/EH** 시리즈 모델: 피연산자 **S** 는 **P0~P255** 를 할당 할 수 있습니다..
5. **ES/EX/SS** 시리즈 모델은 펄스 실행 명령(**CALLP**)을 지원하지 않습니다.
6. 프로그램은 **FEND** 명령이후에 서브루틴에서 유지하지 않습니다.
7. **CALL** 명령의 서브루틴 포인터와 **CJ** 명령의 포인터는 일치하는 것을 허용하지 않습니다.
8. 만일 단지 **CALL** 명령만을 사용한다면, 시간에 제한되지않고 동일한 포인터 넘버의 서브루틴을 호출할 수 있습니다.
9. 서브루틴은 초기 **CALL** 명령을 포함해 5 레벨동안 함유될 수 있습니다. (만일 6 레벨에 들어되면, 서브루틴은 실행되지 않을 것입니다.)

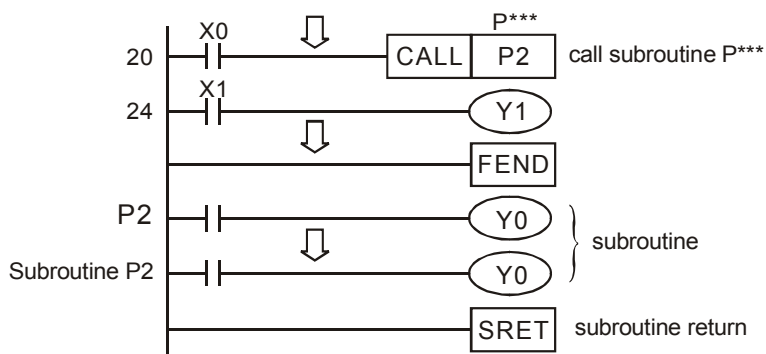
API	연상기호	기능	제어기						
			ES	EX	SS	SA	SX	SC	EH
02	SRET	서브루틴 리턴							
연상부호	설명		프로그램 스텝						
N/A	서브루틴을 구동하는 CALL 명령에 따라 즉시 그 스텝으로 자동으로 되돌아 갑니다.		SRET: 1 스텝						

**설명:**

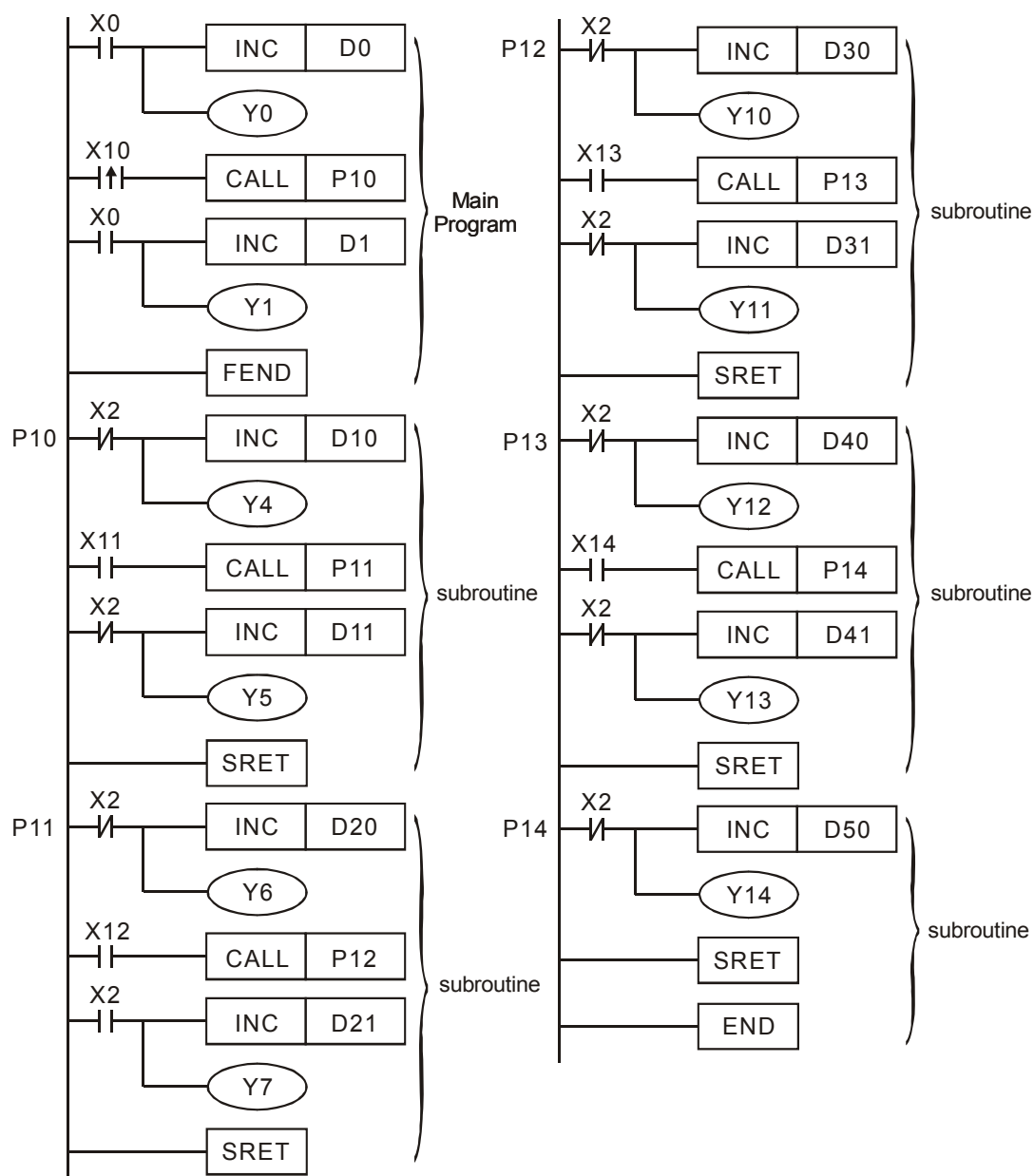
1. 피연산자 없음. 점점에 의한 구동명령은 필요하지 않습니다.
2. 서브루틴 프로그램의 끝을 지시.
3. 서브루틴은 서브루틴의 종료 후 SRET에 의해 메인 프로그램으로 되돌아가서 CALL 명령으로 다음스텝에 위치한 연속 프로그램을 실행 할 것입니다.

**프로그램 예제 1:**

X0 = ON 이면, CALL 명령을 시작하고, P2 로 점프하여 서브루틴을 실행합니다. SRET 명령을 실행하면, 어드레스 24 로 되돌아가 점프하고 계속 실행할 것입니다.

**프로그램 예제 2:**

1. X10 이 Off 에서 On 으로 가는 rising-edge 트리거된 CALL P10 명령이면, P10 으로 점프해서 서브루틴을 구동하십시오.
2. X11 이 On 이면, CALL P11 을 실행하고, P11 로 점프해서 서브루틴을 구동하십시오..
3. X12 이 On 이면, CALL P12 을 실행하고, P12 로 점프해서 서브루틴을 구동하십시오..
4. X13 이 On 이면, CALL P13 을 실행하고, P13 으로 점프해서 서브루틴을 구동하십시오.
5. X14 이 On 이면, CALL P14 을 실행하고,, P14 로 점프해서 서브루틴을 구동하십시오. SRET 명령을 실행하면, 마지막 P\*\*\* 서브루틴을 뒤로 점프하고 계속 실행하십시오.
6. P10 서브루틴에서 SRET 명령을 실행하고 메인 프로그램으로 되돌아가십시오.





API	연상기호	기능	제어기						
			ES	EX	SS	SA	SX	SC	EH
03	IRET	인터럽트 리턴							
연상부호	설명		프로그램 스텝						
N/A	IRET 는 인터럽트 서브루틴의 프로세싱을 끝내고 메인 프로그램으로 되돌아가서 실행하십시오.		IRET: 1 스텝						

## 설명:

1. 피연산자 없음. 점점에 의한 구동 명령은 필요하지 않습니다.
2. IRET 는 서브루틴 프로그램의 인터럽트를 의미합니다.
3. 인터럽트 프로그램의 프로세싱을 끝내고 IRET 명령에 의한 메인 프로그램으로 되돌아 가십시오. 다음 인터럽트 명령을 생산하기 위해 원시 프로그램을 실행하십시오.

API	연상기호	기능	제어기						
			ES	EX	SS	SA	SX	SC	EH
04	EI	인터럽트 가능							
연상부호	설명		프로그램 스텝						
N/A	가능한 인터럽트들, 이 명령의 설명은 또한 DI (불가능한 인터럽트 명령) 설명과 일치합니다. 더 많은 정보를 위해서 DI 명령을 참조하십시오. M1050~M1059, M1280~M1299		EI: 1 스텝						

## 설명:

1. 피연산자 없음. 점점에 의한 구동 명령은 필요하지 않습니다.
2. 인터럽트 신호의 펄스폭은 200us 보다 더 커야합니다.
3. 각 모델넘버의 범위를 참조하기 위해서 DI 명령의 끝부분의 주의를 참조하십시오.
4. M1050~M1059, M1280~M1299 (DI 명령의 끝부분의 주의를 참조하십시오).

API	연상기호	기능	제어기						
			ES	EX	SS	SA	SX	SC	EH
05	DI	인터럽트 불가능							
연상부호	설명		프로그램 스텝						
N/A	EI 명령은 ELC 가 인터럽트를 받아들이는 것인 가능합니다.; 타임 인터럽트 혹은 고속 카운터 인터럽트와 같음. 특수 보조 릴레이 M1050 부터 M1059 · M1280~M1299 를 인터럽트할 때 허용범위를 인터럽트 할 때 조작, 해당 인터럽트 요청은 구동되지 않을 것입니다.		DI: 1 스텝						

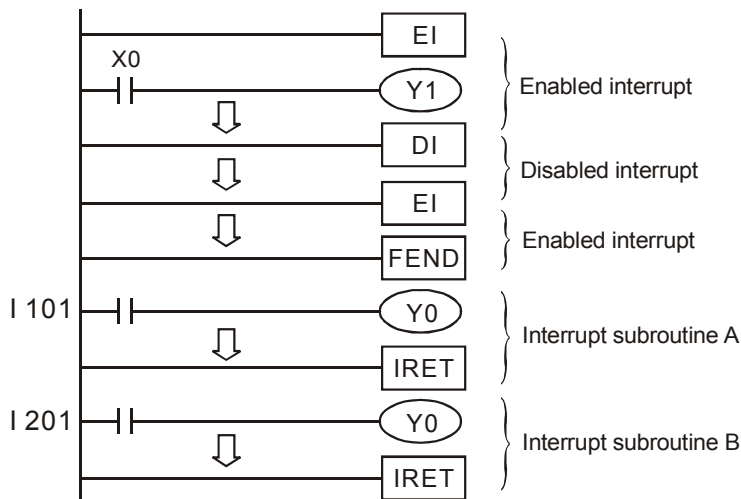
## 설명:

1. 피연산자 없음. 점점에 의한 구동명령은 필요하지 않습니다..

2. EI 명령은 인터럽트 서브루틴이 프로그램에서 처리되도록 합니다. 예를 들면, 외부 인터럽트, 타임 인터럽트, 고속 카운터 인터럽트.
3. 프로그램에서, EI 와 DI 명령사이에 인터럽트 서브루틴을 사용하는 것이 허용됩니다. 하지만, 만일 프로그램동안 인터럽트 불가능 기간이 없다면, DI 명령을 사용하는 것은 허용되지 않습니다.
4. ES/SA/SX/SC 시리즈 모델에서 M1050 부터 M1059 과 EH 시리즈 모델에서 M1280 부터 M1299 의 특수 보조 릴레이를 인터럽트할 때 인터럽트 허용 범위에서 조차, 해당 인터럽트 요청은 구동되지 않을 것입니다.
5. 인터럽트 커서(I)는 FEND 명령 후 사용되어야 합니다..
6. 다른 인터럽트들은 인터럽트 루틴 프로그램을 실행하는동안 발생하는 것을 허용하지 않습니다.
7. 대부분 인터럽트가 발생하면, 우선순위는 먼저 발생한 인터럽트에 주어집니다. 만일 인터럽트가 동시에 일어나면, 인터럽트 더 하위 포인터 넘버를 갖는 인터럽트가 더 높은 우선순위로 주어지게 됩니다.
8. DI 와 EI 명령 사이에 일어나는 어떠한 인터럽트 요청이 발생하는 것은 즉시 실행될 수 없습니다. 그 요청은 기억될 것이고 인터럽트의 가능한 범위에서 서브루틴을 실행할 것입니다.
9. ES/SA 시리즈 모델의 타임 인터럽트는 기억될 것입니다.
10. 인터럽트 포인터를 사용할 때, 동일한 X 입력 점점에 의해 구동된 고속 카운터를 반복적으로 사용하지 마십시오.
11. 인터럽트 루틴 프로그램이 실행되고 I/O 가 즉시 구동되면, I/O 의 상태는 프로그램에서 REF 명령을 써서 재생될 수 있습니다.

#### 프로그램 예제:

PLC 작동시, 프로그램은 EI 와 DI 사이의 명령들을 스캔합니다. 만일, X1 이나 X2 가 ON 이라면, 서브루틴 A 나 B 는 인터럽트 될 것입니다. IRET 가 도달되면, 메인 프로그램은 다시 시작할 것입니다.



#### 주의:

1. ES/EX/SS 시리즈 모델의 인터럽트 포인터 I 넘버들:
  - a) 외부 인터럽트: (I001, X0), (I101, X1), (I201, X2), (I301, X3) 4 포인트.
  - b) 타임 인터럽트: I6□□, 1 포인트 (□□=10~99, 기본 시간=1ms) (5.7 V 지원)
  - c) 구체적 특성을 받아들이기 위한 통신 인터럽트 (I150) (5.7 V 지원)
2. SA/SX/SC 시리즈 모델의 인터럽트 포인터 I 넘버들:

- a) 외부 인터럽트: (I001, X0), (I101, X1), (I201, X2), (I301, X3), (I401, X4), (I501, X5) 6 포인트.
- b) 타임 인터럽트: I6□□, I7□□ 2 포인트. (□□=1~99ms, 기본시간=1ms)
- c) 고속 카운터 인터럽트: I010, I020, I030, I040 4 포인트. (API 53 DHSCS 명령이 사용되어 인터럽트 신호 발생)
- d) 구체적 특성을 받아들이기 위한 통신 인터럽트 (I150)
- e) 인터럽트 포인트 I의 명령: 고속 카운터 인터럽트, 외부 인터럽트, 타임 인터럽트와 구체적 특성을 받아들이기 위한 통신 인터럽트

3. EH 시리즈 모델의 인터럽트 포인트 I 번호 :

- a) 외부 인터럽트: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4), (I50□, X5) 6 포인트. (□=0 은 falling-edge 의 인터럽트를 표시, □=1 은 rising-edge 의 인터럽트를 표시)
- b) 타임 인터럽트: I6□□, I7□□, 2 포인트. (□□=1~99ms, 기본시간=1ms) I8□□ 1 포인트. (□□=1~99ms, 기본시간=0.1ms)
- c) 고속 카운터 인터럽트: I010, I020, I030, I040 4 포인트. (API 53 DHSCS 명령을 사용하여 인터럽트 신호 발생)
- d) 인터럽트, 펄스 출력 인터럽트의 시작과 끝은 API 57 PLSY 명령으로 사용되어야 합니다. I130, I140 은 펄스 출력 명령 M1342, M1343 의 시작이 정해진 플래그에 의해 펄스 출력의 시작에서 트리거됩니다. 그러면, M1340, M1341 은 실행프로그램을 인터럽트하기 위해 펄스 출력 명령의 끝에서 I110, I120 을 트리거하고, 실행하기 위해 목적된 인터럽트 서브루틴으로 점프할 것입니다.
- e) 구체적 특성을 받아들이기 위한 통신 인터럽트(I150)
- f) 인터럽트 포인트 I의 명령: 외부 인터럽트, 타임 인터럽트, 고속 카운터 인터럽트, 펄스 출력 인터럽트.

4. ES/EX/SS 시리즈 모델의 인터럽트 금지 플래그:

플래그	기능
M1050	외부 인터럽트, I 001 마스크 됨
M1051	외부 인터럽트, I 101 마스크 됨
M1052	외부 인터럽트, I 201 마스크 됨
M1053	외부 인터럽트, I 301 마스크 됨

5. SA/SX/SC 시리즈 모델의 인터럽트 금지 플래그:

플래그	기능
M1050	외부 인터럽트, I 001 마스크 됨
M1051	외부 인터럽트, I 101 마스크 됨
M1052	외부 인터럽트, I 201 마스크 됨
M1053	외부 인터럽트, I 301 마스크 됨
M1054	외부 인터럽트, I 401 마스크 됨
M1055	외부 인터럽트, I 501 마스크 됨
M1056	타임 인터럽트, I6□□ 마스크 됨
M1057	타임 인터럽트, I7□□ 마스크 됨
M1059	고속 카운터 인터럽트, I010~I040 마스크 됨

6. EH 시리즈 모델의 인터럽트 금지 플래그:

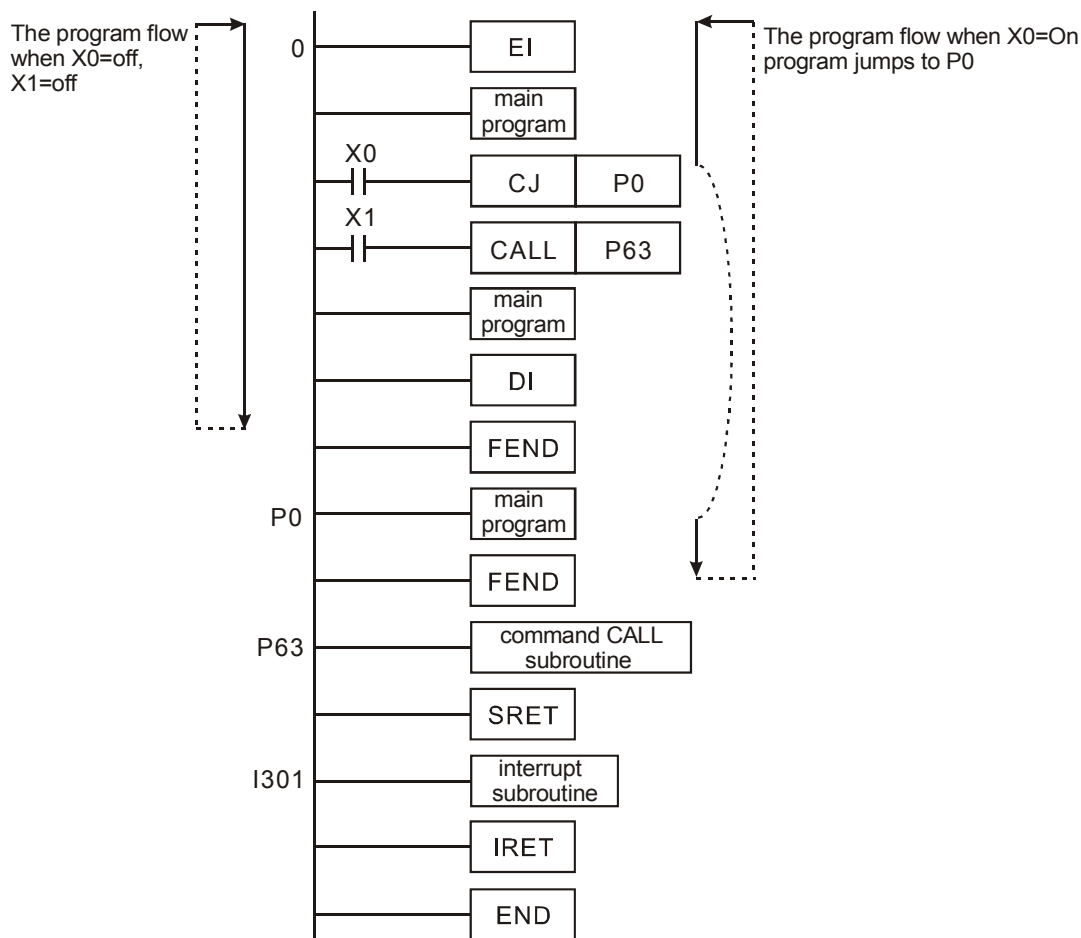
플래그	기능
M1280	외부 인터럽트, I00□마스크 됨
M1281	외부 인터럽트, I10□마스크 됨

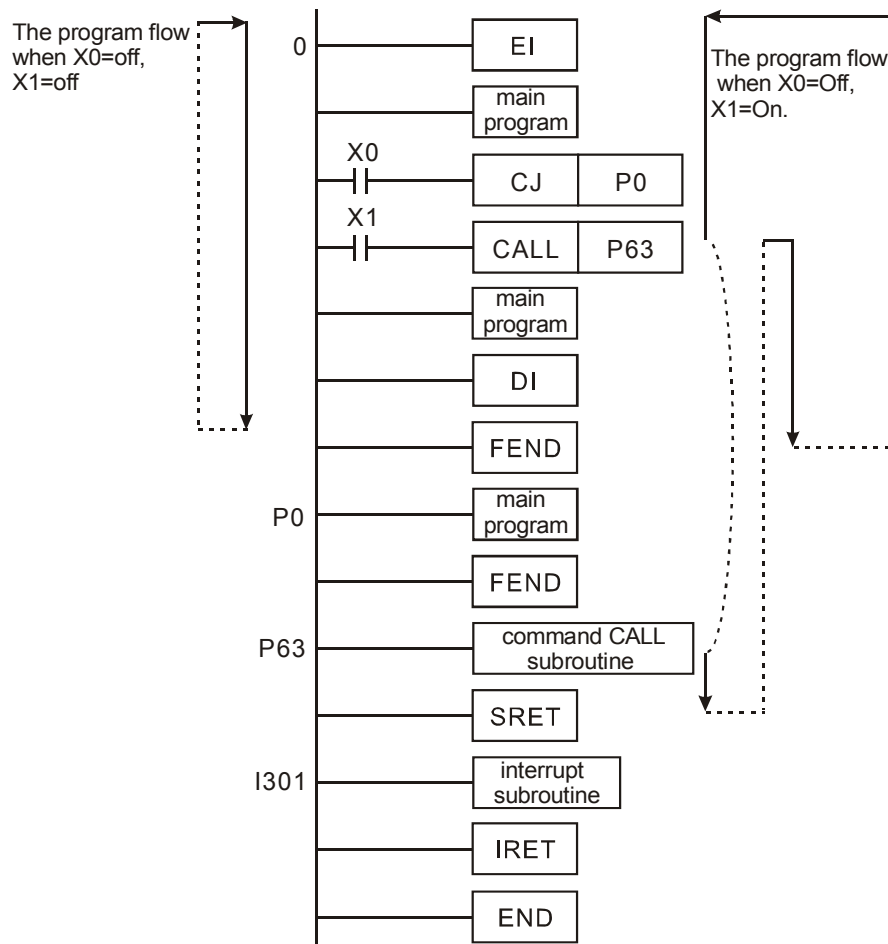
플래그	기능
M1282	외부 인터럽트, I20□ 마스크 됨
M1283	외부 인터럽트, I30□ 마스크 됨
M1284	외부 인터럽트, I40□ 마스크 됨
M1285	외부 인터럽트, I50□ 마스크 됨
M1286	타임 인터럽트, I60□ 마스크 됨
M1287	타임 인터럽트, I70□ 마스크 됨
M1288	타임 인터럽트, I80□ 마스크 됨
M1289	고속 카운터 인터럽트, I010 마스크 됨
M1290	고속 카운터 인터럽트, I020 마스크 됨
M1291	고속 카운터 인터럽트, I030 마스크 됨
M1292	고속 카운터 인터럽트, I040 마스크 됨
M1293	고속 카운터 인터럽트, I050 마스크 됨
M1294	고속 카운터 인터럽트, I060 마스크 됨
M1295	펄스 출력 인터럽트 마스크 된 I110 을 삽입.
M1296	펄스 출력 인터럽트 마스크 된 I120 을 삽입.
M1297	펄스 출력 인터럽트 마스크 된 I130 을 삽입.
M1298	펄스 출력 인터럽트 마스크 된 I140 을 삽입.
M1299	펄스 출력 인터럽트 마스크 된 I150 을 삽입.
M1340	CH0 펄스가 전송된 후, I110 인터럽트 발생.
M1341	CH1 펄스가 전송된 후, I120 인터럽트 발생.
M1342	CH0 펄스가 전송된 후; 그동안, I130 인터럽트 동시에 발생.
M1343	CH1 펄스 가 전송된 후; 그동안, I140 인터럽트 동시에 발생.

API	연상기호	기능	제어기						
			ES	EX	SS	SA	SX	SC	EH
06	FEND	메인 루틴 프로그램을 종결							
연상부호	설명		프로그램 스텝						
N/A	점점에 의한 구동명령이 필요하지 않습니다.		FEND: 1 스텝						

**설명:**

1. 피연산자 없음. 점점에 의한 구동 명령이 필요하지 않습니다.
2. 이 명령은 메인 루틴 프로그램의 끝을 의미합니다. PLC 작동시 END 명령과 동일한 기능을 갖습니다.
3. CALL 은 서브루틴의 끝에 SRET 명령을 붙이고 FEND 명령 바로뒤에 따라와야 합니다. 인터럽트 명령은 또한 서비스 프로그램 끝에 IRET 명령을 붙이고 FEND 명령 뒤에 따라와야만 합니다.
4. 만일 여러 개의 FEND 명령을 사용한다면, 마지막 FEND 와 END 명령 사이의 서브루틴과 인터럽트 서비스 프로그램을 두십시오.
5. CALL 명령이 실행되면, SRET 명령이 실행되기 전에 FEND 명령을 실행할 때 프로그램 에러가 발생할 것입니다.
6. FOR 명령이 실행되면, NEXT 명령이 실행되기 전에 FEND 명령을 실행할 때 프로그램 에러가 발생할 것입니다.

**CJ 명령프로그램 흐름:****CALL 명령프로그램 흐름:**



API	연상기호			기능	제어기							
		WDT	P		ES	EX	SS	SA	SX	SC	EH	
07				감시 타이머 리셋								
연상부호		설명				프로그램 스텝						
N/A						WDT, WDTP: 1 스텝						
					펄스				16-비트			
					ES	EX	SS	SA	SX	SC	EH	
					ES	EX	SS	SA	SX	SC	EH	
					ES	EX	SS	SA	SX	SC	EH	

**설명:**

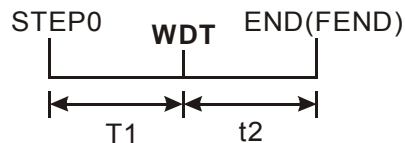
1. 피연산자 없음. ES/EX/SS 시리즈 모델은 펄스 실행 명령(WDTP)을 지원하지 않습니다.
2. WDT (감시타이머)는 DVP 시리즈 PLC 시스템에서 PLC 작동을 모니터 하는데 사용됩니다.
3. WDT 명령은 감시타이머를 리셋하기 위해 사용될 수 있습니다. 만일 PLC 스캔 타임 (스텝 0 부터 END 까지 혹은 FEND 명령)은 200ms 이상이라면, 에러 LED 가 켜질 것입니다. 사용자는 오류를 삭제하기 위해 PLC 를 끄고 다시 켜야 할 것입니다. PLC 는 RUN/ST 연상부호 스위치에 따라 실행/정지의 상태를 결정할 것입니다. 만일 RUN/ST 연상부호 스위치가 없다면, PLC 는 자동적으로 정지로 되돌아 갈 것입니다.

**4. WDT 를 사용할 때:**

PLC 시스템에서 에러가 발생하면.

- a) 프로그램의 실행시간이 너무길어서 D1000 의 내용값을 초과하는 스캔시간을 가질 수 없다면, 다음 두가지 방법을 사용하여 수정될 수 있습니다.

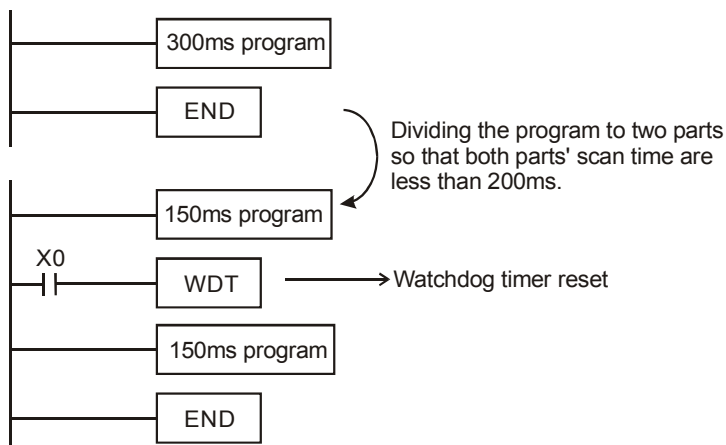
■ WDT 명령 사용



- 감시타임을 변경하기 위해 D1000 (디폴트는 200ms) 의 설정값을 사용.

**프로그램 예제:**

만일 프로그램 스캔 타임이 300ms 를 초과한다면, 사용자는 프로그램을 2 부분으로 분리할 수 있습니다. 그 사이에 감시타이머를 삽입하십시오. 그러면 두개의 프로그램의 스캔 타임은 200ms 보다 적게 될 것입니다.



API	연상기호		피연산자 s		기능				제어기									
08	FOR		S		루프 시작				ES	EX	SS	SA	SX	SC	EH			
타입 연상부호	비트 장치				워드 장치										프로그램 스텝			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FOR: 3 스텝		
S					*	*	*	*	*	*	*	*	*	*	*			

피연산자:

**S:** 포대기형 루프의 반복 수

설명:

1. 점점실행명령은 필요하지 않습니다.
2. 사용 범위를 위해 각 모델 규격을 참조하십시오.
3. **S:** 포대기형 루프의 반복 수.

API	연상기호	기능	제어기						
09	NEXT	루프 끝	ES	EX	SS	SA	SX	SC	EH
연상부호	설명		프로그램 스텝						
N/A			NEXT: 1 스텝						

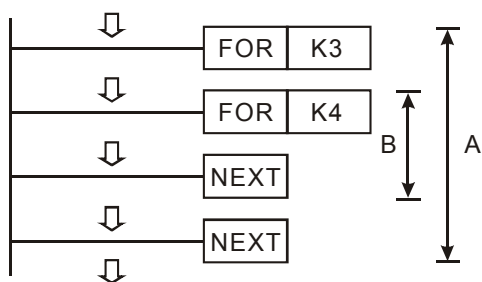
설명:

1. 피연산자 없음. 점점에 의한 구동명령은 필요하지 않습니다.
2. FOR 와 NEXT 명령은 “n” 포대기형 루프가 필요하지 않을 때 사용될 수 있습니다.
3. “N”은 K1 에서 K32767 까지 범위내에 있을 수 있습니다. 만일 범위  $N \leq K1$  라면, N 은 항상 K1 이 될 것입니다.
4. FOR 에서 NEXT 명령까지 실행하길 원하지 않는다면, CJ 명령을 사용하십시오.
5. 다음 상태에서 에러가 발생할 것입니다.:
  - a) NEXT 명령이 FOR 명령 전 일 때.
  - b) FOR 명령이 있고, NEXT 명령이 없을 때.
  - c) FEND 나 END 명령후 NEXT 명령이 있을 때.
  - d) NEXT 명령을 위한 FOR 의 수가 다를 때.
6. FOR 에서 NEXT 까지 루프는 5 레벨동안 함유될 수 있지만, 만일 너무 많은 루프가 있다면, PLC 스캔 타임은 증가할 것이고 감시타이머가 구동되게 야기할 수 있고 에러의 결과를 초래할 수 있음을 주의하십시오. 사용자는 수정하기 위해 WDT 명령을 사용할 수 있습니다.

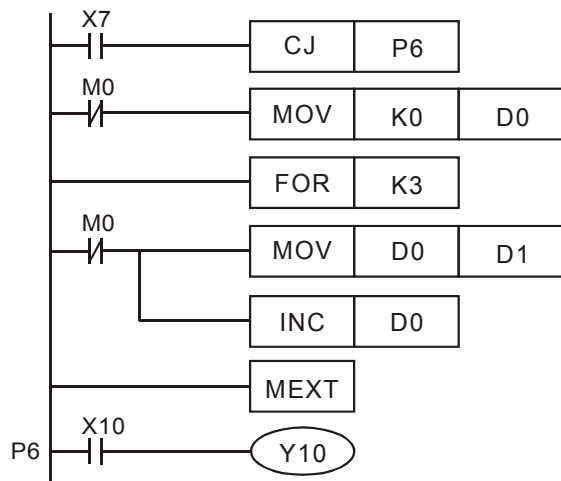
프로그램 예제 1:

루프 A 가 3 번 동작하면, NEXT 명령 후의 프로그램은 다시 시작 할 것입니다. 루프 A 의 모든 완성된 회전 을 위해서, 루프 B 는 완벽하게 4 번을 실행 할 것입니다. 그러므로 루프 B 가 동작하는 전체 횟수는  $3 \times 4 = 12$  번이 될 것입니다.

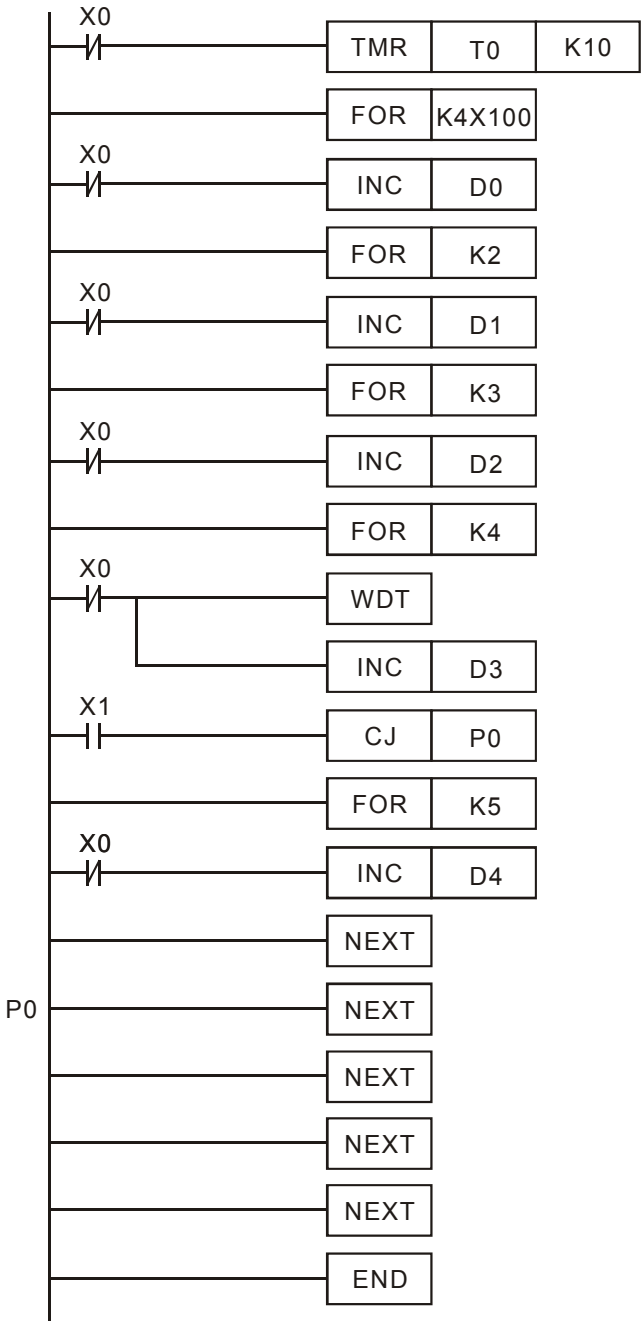


**프로그램 예제 2:**

X7 이 off 일 때 FOR 에서 NEXT 명령까지 실행하는 프로그래밍. X7 가 on 이고 CJ 명령이 P6 으로 점프할 때, FOR 에서 NEXT 명령까지를 실행하지 않습니다.

**프로그램 예제 3:**

FOR 에서 NEXT 명령까지 실행되지 않을 때, CJ 명령은 점프하기 위해 사용될 수 있습니다. FOR 에서 NEXT 까지 가장 내부 루프일 때, X1 는 ON 이 될 것이고 CJ 명령은 P0 로 점프하고 실행되지 않을 것입니다.



API	연상기호			피연산자			기능			제어기						
	D	CMP	P	S <sub>1</sub>	S <sub>2</sub>	D	비교			ES	EX	SS	SA	SX	SC	EH
10																

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F						
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*	CMP, CMPP: 7 스텝 DCMP, DCMPP: 13 스텝					
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*						
D		*	*	*																	

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

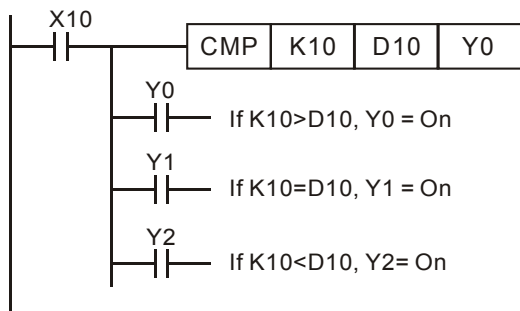
**S<sub>1</sub>**: 첫번째 비교 값    **S<sub>2</sub>**: 두번째 비교 값    **D**: 비교 결과

설명:

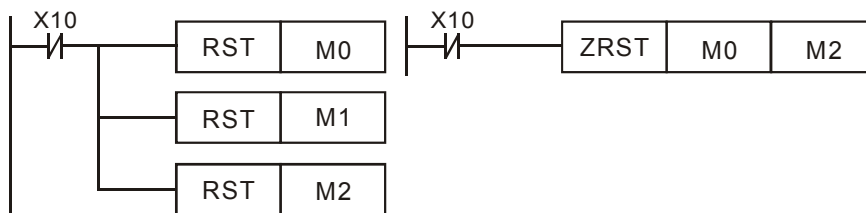
- 만일 피연산자 S<sub>1</sub>, S<sub>2</sub> 가 장치 F 를 사용한다면, 그것은 단지 16-비트 명령에서만 사용 가능할 것입니다.
- 피연산자 D 는 3 개의 유지 장치를 갖습니다.
- 사용 범위를 위해서 각 모델 규격을 참조 하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(CMPP, DCMPP)을 지원하지 않습니다.
- 비교 소스 S<sub>1</sub> 과 S<sub>2</sub> 의 내용은 비교되고 D 는 비교 결과를 표시합니다.
- 두개의 비교값은 수학상 비교되어 2 진수값으로 두개 값을 비교합니다. 만일 16-비트 명령에서 b15=1 이거나 32-비트 명령에서 b31=1 이라면, 그 비교는 그 값을 음의 2 진수값로 간주할 것입니다.

프로그램 예제:

- 만일 D 가 Y0 로 설정되면, Y0, Y1, Y2 는 아래 프로그램 예제와 같이 작동할 것입니다.
- X10=On 이면, CMP 명령은 구동되고 Y0, Y1, Y2 중 하나는 On 이 됩니다. X10=Off 이면, CMP 명령은 구동되지 않고 Y0, Y1, Y2 는 이전 상태로 남게 됩니다.
- ≥, ≤, ≠ 명령의 비교 결과는 Y0~Y2 의 병렬접속에 의해 구해질 수 있습니다.



- 비교 결과를 리셋하기 위해 RST 혹은 ZRST 명령을 사용하십시오.



API	연상기호			피연산자				기능		제어기						
11	D	ZCP	P	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>S</b>	<b>D</b>	구역 비교		ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ZCP, ZCPP: 9 스텝 DZCP, DZCPP: 17 스텝					
	S <sub>1</sub>				*	*	*	*	*	*	*	*	*	*	*						
	S <sub>2</sub>				*	*	*	*	*	*	*	*	*	*	*						
	S				*	*	*	*	*	*	*	*	*	*	*						
D		*	*	*																	

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**피연산자:**

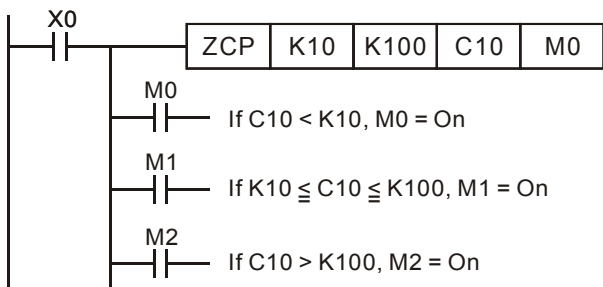
**S<sub>1</sub>**: 첫번째 비교값(최소)    **S<sub>2</sub>**: 두번째 비교값(최대)    **S**: 비교값    **D**: 비교결과

**설명:**

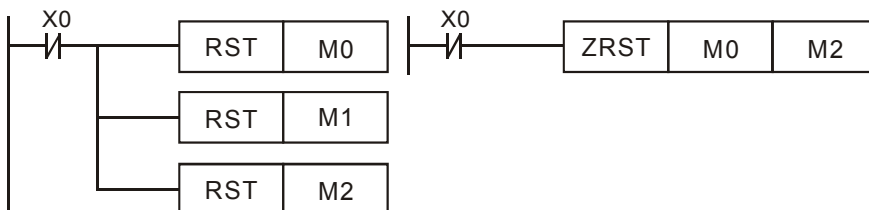
- 만일 피연산자 S<sub>1</sub>, S<sub>2</sub>, S 가 장치 F 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 피연산자 S<sub>1</sub> 은 피연산자 S<sub>2</sub> 보다 작아야 합니다.
- 피연산자 D 는 3 개의 유지 장치를 갖습니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(ZCPP, DZCPP)을 지원하지 않습니다.
- S 는 S<sub>1</sub> 과 S<sub>2</sub> 의 한계값과 비교되고 D 는 비교 결과를 표시합니다.
- S<sub>1</sub> > S<sub>2</sub> 일때, 비교하기 위한 한계값으로서 S<sub>1</sub> 을 설정하십시오.
- 두개의 비교 값은 수학상 비교되고 이 기능은 2 진수 값을 고려한 두개의 값들을 비교합니다. 만일 16-비트 명령에서 b15=1 이거나 32-비트 명령에서 b31=1 이라면, 그 비교는 음의 2 진수값으로서 그값을 간주할 것입니다.

**프로그램 예제:**

- 만일 D 가 M0 로 설정되면, M0, M1, M2 는 아래 프로그램 예제와 같이 작동할 것입니다.
- X0=On 이면, ZCP 명령은 구동되고 M0, M1, M2 중 하나는 On 이 됩니다. X0=Off 이면, ZCP 명령은 구동되지 않고 M0, M1, M2 는 이전 상태로 남아있게 됩니다.



- 비교 결과를 리셋하기 위해서는 RST 나 ZRST 명령을 사용하십시오.



API	연상기호			피연산자		기능					제어기							
12	D	MOV	P	<div>S</div>	<div>D</div>	이동					<div>ES</div> <div>EX</div> <div>SS</div> <div>SA</div> <div>SX</div> <div>SC</div> <div>EH</div>							

타입 연상부호	비트 장치				워드 장치											프로그램 스텝								
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MOV, MOVP: 5 스텝								
S					*	*	*	*	*	*	*	*	*	*	*		DMOV, DMOVP: 9 스텝							
D								*	*	*	*	*	*	*	*									

펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

**S**: 데이터 소스    **D**: 데이터 목적지

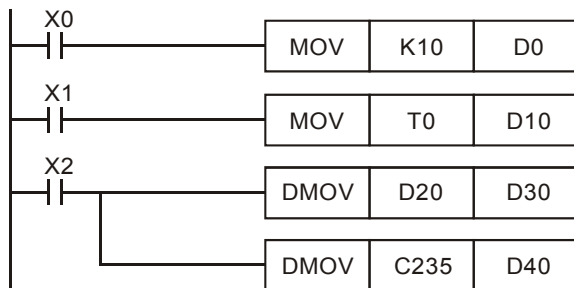
설명:

- 만일 피연산자 **S**, **D** 가 장치 **F** 를 사용하지 않는다면, 16-비트 명령에서만 사용 가능합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(MOVP, DMOVP)을 지원하지 않습니다.
- MOV 명령이 구동되면, **S** 의 데이터는 어떠한 변화없이 **D** 로 이동될 것입니다. 만일 MOV 명령이 구동되지 않으면, **D** 의 내용은 어떠한 변화도 없을 것입니다.
- 만일 계산 결과가 32-비트 출력 (즉, 응용 MUL) 이고, 32-비트 고속 카운터의 데이터라면, 사용자는 DMOV 명령을 사용해야 할 것입니다.

프로그램 예제:

- MOV 명령은 데이터를 이동하기 위해 16-비트 명령에서 사용될 수 있습니다.
  - X0=Off 라면, D10 의 내용은 어떠한 변화도 없을 것입니다. 만일 X0=On 이라면, K10 의 데이터는 D10 데이터 레지스터로 이동할 것입니다.
  - X1=Off 라면, D10 의 내용은 어떠한 변화도 없을 것입니다. 만일 X1=On 라면, the of T0 데이터는 D10 데이터 레지스터로 이동할 것입니다.
- DMOV 명령은 데이터 이동을 위해 32-비트 명령에서 사용됩니다.
 

X2=Off 라면, (D31, D30) 과 (D41, D40)의 내용은 어떠한 변화도 없을 것입니다. 만일 X2=On 이라면, (D21, D20)의 데이터는 (D31, D30) 데이터 레지스터로 이동할 것입니다. 동시에, C235 데이터는 (D41, D40) 데이터 레지스터로 이동할 것입니다.



API	연상기호			피연산자						기능		제어기							
13		SMOV	P	<b>S</b>	<b>m<sub>1</sub></b>	<b>m<sub>2</sub></b>	<b>D</b>	<b>n</b>	쉬프트 이동			ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SMOV, SMOVP: 11 스텝										
S							*	*	*	*	*	*	*	*	*											
m <sub>1</sub>					*	*																				
m <sub>2</sub>					*	*																				
D								*	*	*	*	*	*	*	*											
n					*	*																				

펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

**피연산자:**

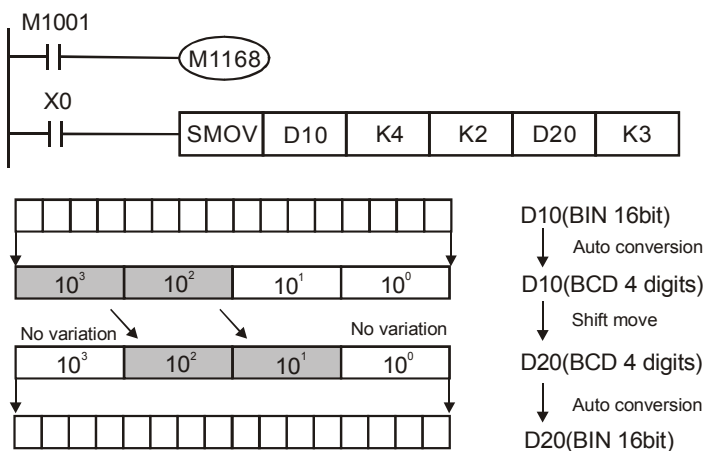
**S:** 데이터소스    **m<sub>1</sub>:** 이동하기 위한 첫번째 digit 의 소스위치(니블)    **m<sub>2</sub>:** 이동하기 위한 소스 digit(니블)의 수    **D:** 목적지    **n:** 첫번째 digit 를 위한 목적지 위치(니블)

**설명:**

1. 피연산자 m<sub>1</sub> 의 사용 범위: m<sub>1</sub>=1~ 4.
2. 피연산자 m<sub>2</sub> 의 사용 범위: m<sub>2</sub>=1~ m<sub>1</sub>.
3. 피연산자 n 의 사용 범위: n=m<sub>2</sub> ~ 4.
4. 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
5. 이 명령은 데이터를 지정하거나 조합할 수 있습니다.
6. M1168 (SMOV 의 모드 설정 작동). M1168=On 이면, BIN 모드. M1168=Off 이면, BCD 모드.

**프로그램 예제 1:**

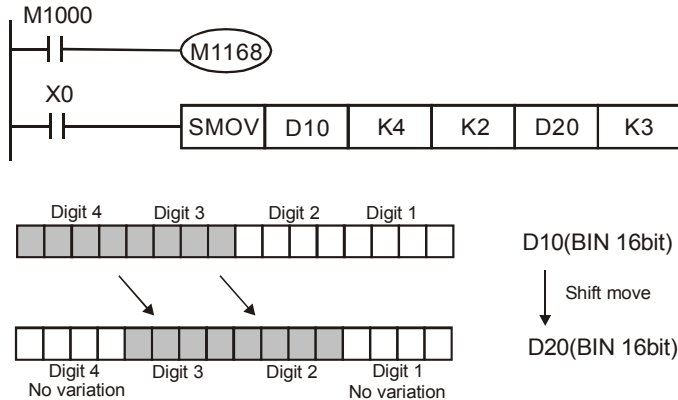
1. M1168=Off, X0=On 이면, D10 (십진수)의 4<sup>th</sup> digit (천의 자리 digit)으로부터 2<sup>nd</sup> digit 의 내용을 지정하고 데이터 D20 (십진수)의 2<sup>nd</sup> digit (백의 자리 digit)으로부터 2 digit 에 지정된 데이터를 이동하십시오. 그리고, SMOV 명령이 실행되면 D20 의 10<sup>3</sup> 과 10<sup>0</sup> 의 내용은 어떠한 변화도 없을 것입니다.
2. BCD 넘버가 9,999 보다 크거나 음수(0 부터 9,999 외의 범위)가 된다면, PLC 에서 작동에러가 발생할 것입니다. 그리고 그 명령이 실행되지 않고 M1067, M1068 는 On 이 되어, D1067 는 에러 코드 “0E18” (16 진수)를 기록할 것입니다.



만일 실행전에 D10=K1234 , D20=K5678 라면, D10 은 실행후 변화하지 않고 D20=K5128 가 될 것입니다.

**프로그램 예제 2:**

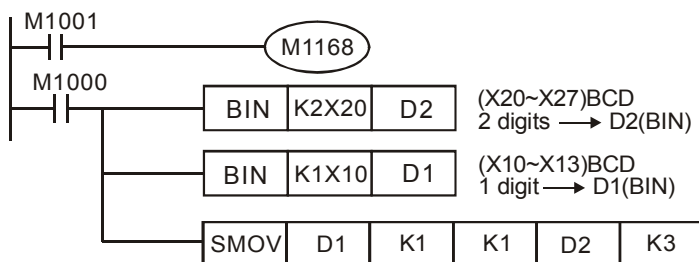
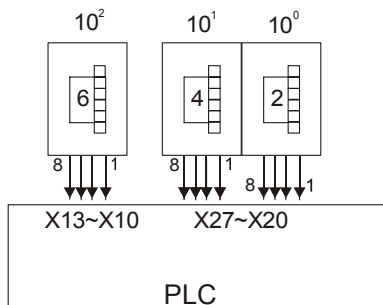
M1168=On 일 때, 만일 SMOV 명령을 사용한다면, D10 과 D20 은 BCD 포맷으로 데이터를 이동하지 않을 것입니다. 하지만 그 데이터는 BIN 넘버 4 digit 으로 이동될 것입니다.



만일 실행 전 D10=H1234 · D20=H5678 라면, 실행이 끝난후 D10 은 변화하지 않고 D20=H5128 일 것입니다.

**프로그램 예제 3:**

1. 인터럽트된 넘버 입력과 연결된 Digit 스위치는 SMOV 명령을 조합하기 위해 사용할 수 있습니다.
2. 오른쪽 두번째 digit 스위치를 D2 의 오른쪽 두번째 digit 로 이동하고 왼쪽 첫번째 digit 스위치를 D1 의 오른쪽 첫번째 digit 로 이동하십시오.
3. 첫번째 digit 을 D2 의 세번째 digit 으로 이동하기 위해 SMOV 명령을 사용하고 그 두개의 digit 스위치를 한 그룹으로 조합하십시오.



API	연상기호			피연산자	기능	제어기						
14	D	CML	P	<b>S</b> <b>D</b>	보수와 이동	ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치											프로그램 스텝
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CML, CMLP: 5 스텝
S					*	*	*	*	*	*	*	*	*	*	*	DCML, DCMLP: 9 스텝
D								*	*	*	*	*	*	*	*	

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

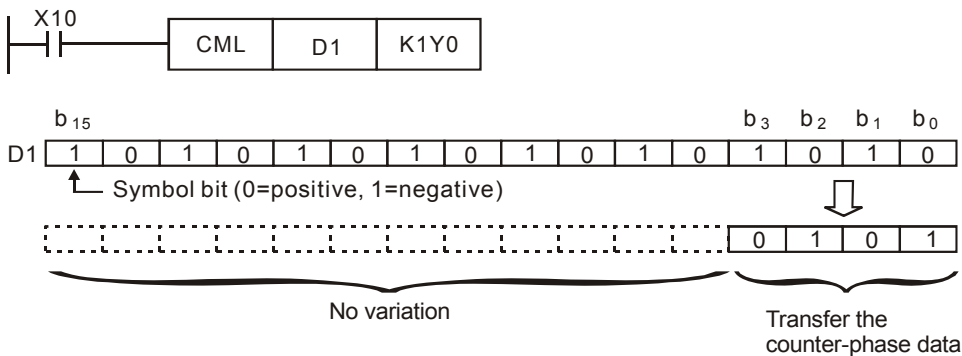
**S**: 데이터 소스    **D**: 목적지

설명:

- 만일 피연산자 **S**, **D** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(CMLP, DCMLP)을 지원하지 않습니다.
- S** (0→1, 1→0)의 내용을 거꾸로의 상태로 두고 **D** 로 전송된 내용을 가져오십시오. 만일 그 내용이 상수 **K** 라면, 이 상수 **K** 는 자동적으로 BIN 값으로 변환할 것입니다.

프로그램 예제 1:

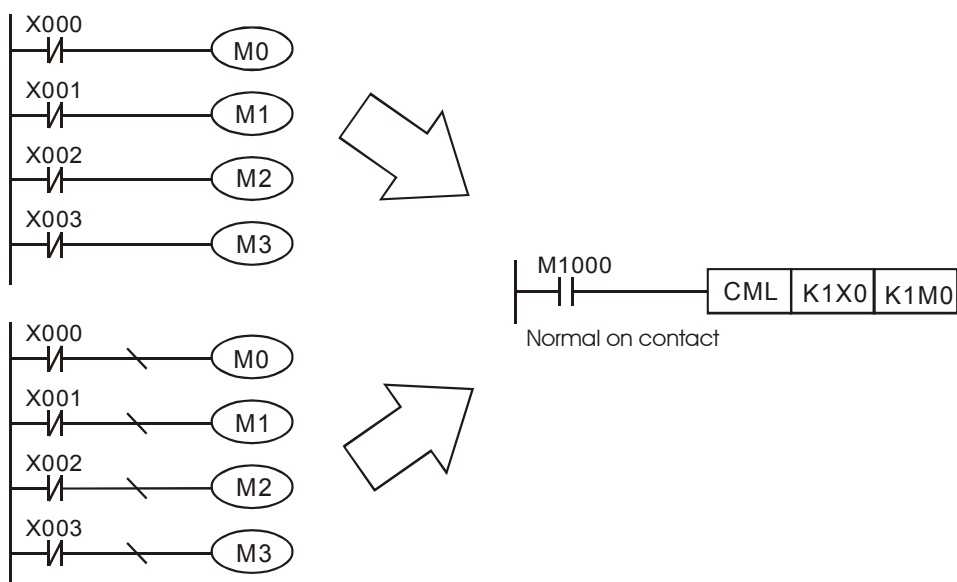
- 이 명령은 거꾸로의 상태를 출력시 사용할 수 있습니다.
- X10=ON 이면, D1, b0~b3의 내용은, K1Y0 에 거꾸로 전송될 것입니다.



프로그램 예제 2:

아래 보인 왼쪽 루프는 CML 명령을 사용한 오른쪽 프로그램 예제와 같이 표시 할 수 있습니다.





API	연상기호			피연산자			기능			제어기						
15		BMOV	P	<b>S</b>	<b>D</b>	<b>n</b>	블록 이동			ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BMOV, BMOVP: 7 스텝					
S							*	*	*	*	*	*	*								
D								*	*	*	*	*	*								
n					*	*					*	*	*								

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

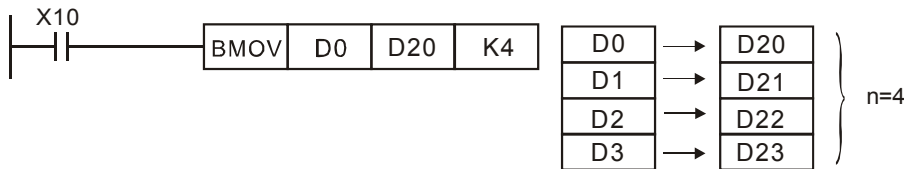
**S**: 소스    **D**: 목적지    **n**: 이동하기 위한 데이터의 수

설명:

1. 피연산자의 사용 범위는 **n=1~512**
2. 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
3. ES/EX/SS 시리즈 모델은 펄스 실행 명령(**BMOVP**) 을 지원하지 않습니다.
4. 이 명령은 새로운 목적지에 지정된 복수의 데이터를 이동하기 위해 사용할 수 있습니다. **S**-할당된 넘버들 내에 레지스터를 카운트하여 얻은 이 **n** 레지스터를 가지고, **D**-할당된 넘버 안의 **n** 레지스터로, **n** 레지스터의 내용을 이동하십시오. 만일 **n**-할당된 포인트가 이장치의 사용범위를 초과한다면, 단지 가능한 범위안으로만 이동될 것입니다.

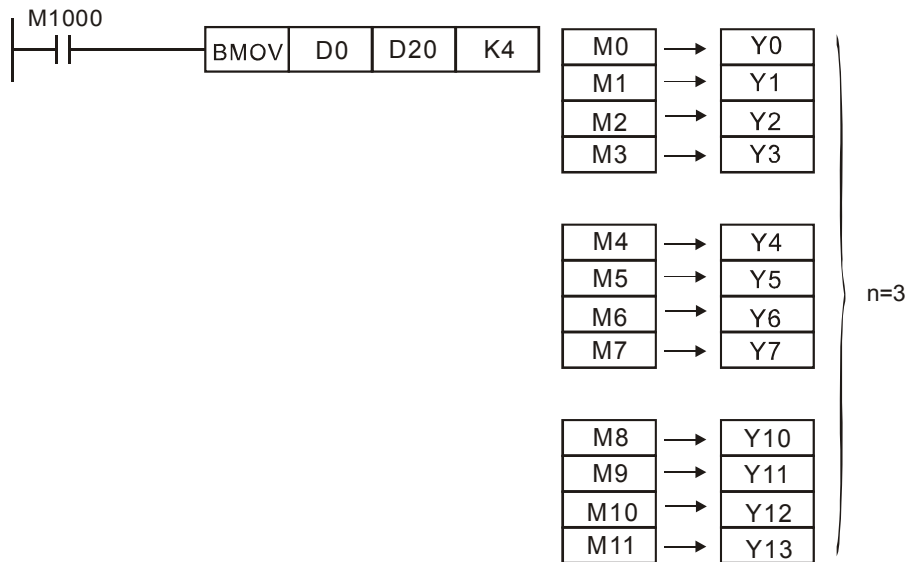
#### 5. 프로그램 예제 1:

X10=On 이면, 내용 4 개의 레지스터 D0~D3 을 그 해당 레지스터 D20~D23 에 그 내용을 이동하십시오.



#### 프로그램 예제 2:

1. 만일 지정된 비트 장치를 이동한다면, KnX, KnY, KnM, KnS, **S** 와 **D** 의 digit 넘버는 동일해야 하고 또한 **n** 의 넘버는 동일해야 함을 의미합니다.
2. ES/EX/SS 시리즈 모델은 KnX, KnY, KnM, KnS 장치를 지원하지 않습니다.



### 프로그램 예제 3:

BMOV 명령은 지정된 **S** 와 **D** 의 넘버를 포함할 때 겹쳐쓰기 에러 발생을 방지하기 위해 아래 프로그램 예제와 같이 자동 이동을 생성합니다.

1. **S > D** 이면, BMOV 명령은 ①→②→③과 같은 순서로 처리합니다.
2. **S < D** 이면, BMOV 명령은 ③→②→① 과 같은 순서로 처리합니다. 하지만, ES/EX/SS 시리즈 모델에서 **S < D** 을 때, 유지하는 지정된 넘버를 확실히 피하십시오. 그렇지 않으면, 그 실행 결과는 동일 값이 될 것입니다. 예를 들면, BMOV 명령이 ③→②→① 과 같은 순서로 처리하면, D11 부터 D13 까지 내용 값은 모두 D10 의 내용 값이 될 것입니다.



API	연상기호			피연산자			기능			제어기							
16	D	FMOV	P	<b>S</b>	<b>D</b>	<b>n</b>	채우기 이동			ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FMOV, FMOVP: 7 스텝  DFMOV, DFMOVP: 13 스텝					
					*	*	*	*	*	*	*	*	*	*	*						
								*	*	*	*	*	*								
					*	*															

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자 s:

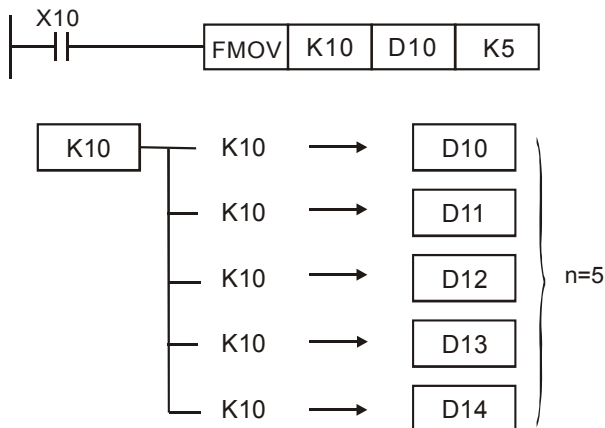
**S**: 소스    **D**: 목적지    **n**: 이동하기 위한 데이터의 넘버

설명:

- 만일 피연산자 **S, D** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 피연산자 **n** 의 사용 범위; **n=1~512**(16-비트 명령), **n=1~256** (32-비트 명령).
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(FMOVP, DFMOVP)을 지원하지 않습니다.
- 소스 장치 **S** 에 저장된 데이터는 목적지 장치의 범위내에 모든 장치로 이동합니다. **D** -할당된 넘버내에 레지스터를 카운트 함으로써 얻은 이 **n** 레지스터를 가지고 **S** 내용을 **n** 레지스터로 이동하십시오. 만일 **n**-지정한 장치가 사용범위를 초과한다면, 단지 가능한 범위 내에서만 이동될 것입니다.
- ES/EX/SS 시리즈 모델은 KnX, KnY, KnM, KnS 장치를 지원하지 않습니다.

프로그램 예제:

X0=ON 이면, D10 부터 시작하여 계속적인 5 개의 레지스터들 (D10~D14)로 상수 K10 를 이동하십시오.



API	연상기호			피연산자	기능	제어기							
17	D	XCH	P	<b>(D<sub>1</sub>)</b> <b>(D<sub>2</sub>)</b>	교환	ES	EX	SS	SA	SX	SC	EH	

타입	비트 장치				워드 장치											프로그램 스텝
연상부호	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D <sub>1</sub>								*	*	*	*	*	*	*	*	XCH, XCHP: 5 스텝
D <sub>2</sub>								*	*	*	*	*	*	*	*	DXCH, DXCHP: 9 스텝

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

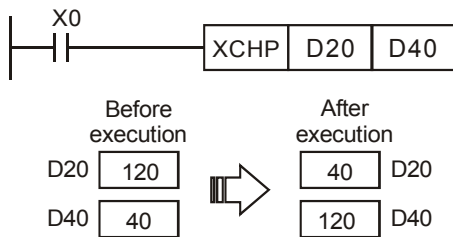
**D<sub>1</sub>**: 첫번째 교환 장치    **D<sub>2</sub>**: 두번째 교환 장치

설명:

- 만일 피연산자 **D<sub>1</sub>**, **D<sub>2</sub>** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(XCHP, DXCHP)을 지원하지 않습니다.
- D<sub>1</sub>**: 첫번째 교환 데이터    **D<sub>2</sub>**: 두번째 교환 데이터
- D<sub>1</sub>** 과 **D<sub>2</sub>** 의 내용을 서로 교환하십시오.
- 이 명령은 보통 펄스 실행 (XCHP)을 합니다.

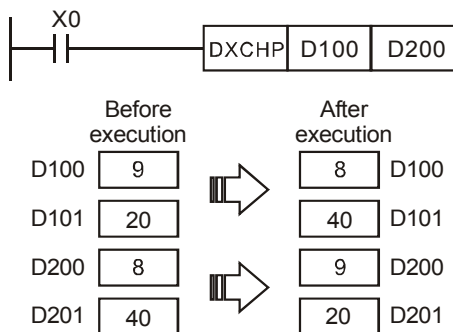
프로그램 예제 1:

X0=Off→On 이면, D20 과 D40 의 내용을 서로 교환합니다.



프로그램 예제 2:

X0=Off→On 이면, D20 과 D40 의 내용을 서로 교환합니다.

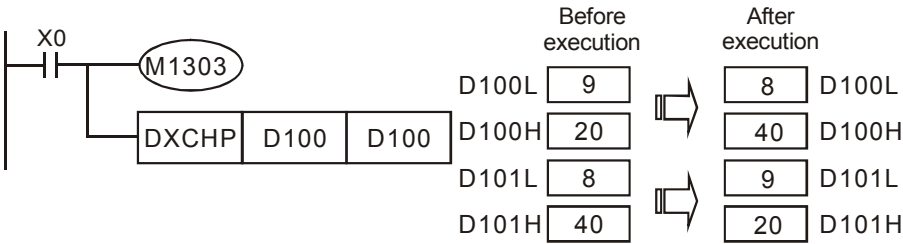


주의:

- ES 시리즈 모델은 플래그 M1303 을 지원하지 않습니다.
- 16-비트 명령에서, **D<sub>1</sub>** 과 **D<sub>2</sub>** 로 지정된 장치가 동일하고 M1303=On 이면, 지정된 장치의 상위과 하위 8-비트 내용은 교환될 것입니다.

3. 32-비트 명령에서, **D<sub>1</sub>** 과 **D<sub>2</sub>** 로 지정된 장치가 동일하고 M1303=On 이면, 32-비트 장치의 상위와 하위 16-비트 내용은 교환될 것입니다.

X0=On 이고 M1303=On 이면, D100 과 D101 의 내용은 교환될 것입니다.



API	연상기호			피연산자		기능										제어기							
18	D	BCD	P	<b>(S)</b>	<b>(D)</b>	BIN 을 BCD 로 변환										ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치											프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BCD, BCDP: 5 스텝  DBCD, DBCDP: 9 스텝							
	S						*	*	*	*	*	*	*	*	*								
	D							*	*	*	*	*	*	*	*								

펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

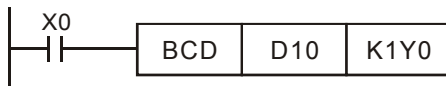
**S:** 소스    **D:** 변환 결과

설명:

- 만일 피연산자 **S, D** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS** 시리즈 모델은 펄스 실행 명령(**BCDP**, **DBCDP**)을 지원하지 않습니다.
- 소스 장치 **S** 의 **BIN** 데이터 (0 부터 9999)를 **BCD** 로 변환하고 장치 **D** 에 그 결과를 전송하십시오.
- 만일 **BCD** 변환 결과가 0 부터 9999 의 범위 밖이라면, 작동 에러가 발생하고, 에러 플래그 **M1067**, **M1068** 이 On 이 되고 **D1067** 는 에러 코드 "0E18" (16 진수)를 기록 할 것입니다.
- 만일 **DBCD** 변환 결과가 is outside the 범위 of 0 부터 99,999,999 의 범위 밖이라면, 작동에러가 발생하고, 에러 플래그 **M1067**, **M1068** 은 On 이 되고 **D1067** 은 에러 코드 "0E18" (16 진수)을 기록 할 것입니다.
- PLC 에서 4 개의 기본연산(+, -, ×, ÷)의 연산 값, **INC**, **DEC** 명령은 **BIN** 포맷에서 실행됩니다. 이 명령은 **BIN** 포맷 데이터를 **BCD** 포맷 값으로 직접 7 개 세그먼트 표시로 출력하기 위해 사용될 수 있습니다.
- M1067** (작동 에러). **M1068** (작동 에러). **D1067** (에러 코드).

프로그램 예제:

- X0=ON** 이면, **D10** 의 2 진수 데이터는 **BCD** 넘버로 변환하고, **K1Y0** (**Y0~Y3**)에 저장됩니다.



- D10=001E (Hex)=0030** (십진수) 이면, 실행 결과는 **Y0~Y3=0000(BIN)**이 될 것입니다.

API	연상기호			피연산자		기능			제어기							
19	D	BIN	P	<div>S</div>	<div>D</div>	BCD 를 BIN 으로 변환			ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BIN, BINP: 5 스텝  DBIN, DBINP: 9 스텝					
	S						*	*	*	*	*	*	*	*	*						
	D							*	*	*	*	*	*	*	*						

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S**: 소스    **D**: 변환 결과

설명:

- 만일 피연산자 **S**, **D** 가 장치 **F** 를 사용하면, 16-비트 명령에서만 사용 가능합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(BINP, DBINP)을 지원하지 않습니다.
- 소스 장치 **S** 의 BCD 데이터 (0 부터 9,999)를 BIN 으로 변환하고 그 결과를 장치 **D** 로 전송하십시오.
- 소스 장치 **S** 의 가능 범위 : BCD (0 부터 9,999), DBCD (0 부터 99,999,999)
- 만일 소스 장치 **S** 의 내용이 BCD 값 (0 부터 9 까지 범위 외에 16 진수로 표시된 각 **S** 의 digit)이 아니라면, 작동 에러가 발행할 것이고, 에러 플래그 M1067, M1068 는 On 이 될것이고 D1067 는 에러 코드 "0E18"을 기록할 것입니다.
- 상수 K 와 H 는 자동적으로 BIN 데이터로 변환됩니다. 이 명령을 사용하기 위해 상수는 필요하지 않습니다.
- M1067 (작동 에러). M1068 (작동 에러). D1067 (에러 코드)

프로그램 예제:

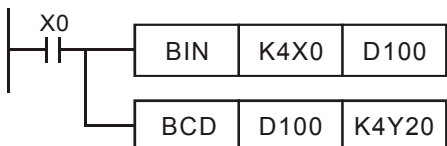
X0=ON 이면, BCD 데이터 K1M0 은 BIN 데이터로 변환되고, D10 에 그 결과를 저장합니다.



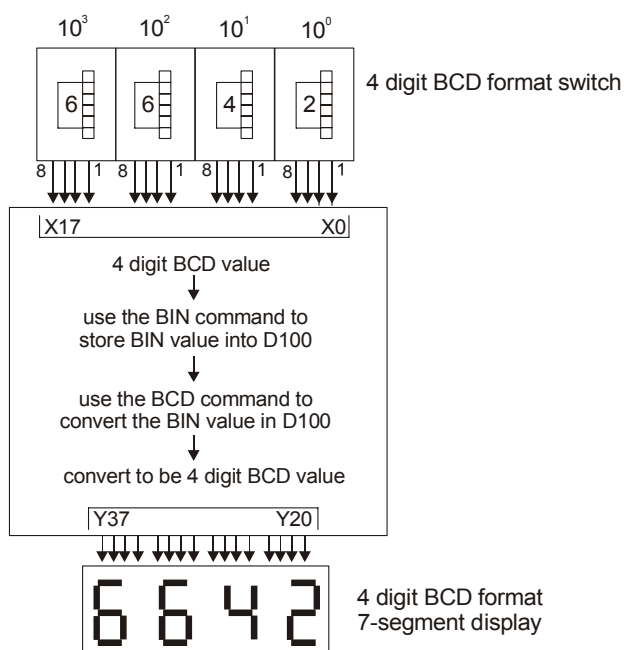
주의:

BCD 와 BIN 명령의 응용 설명:

- BIN 명령은 소스 데이터를 BIN 데이터로 변환하고 PLC 가 BCD 포맷 digit 스위치를 외부 장치로부터 읽을 때 PLC 에 저장하기 위해 사용합니다.
- BCD 명령은 is used to convert the 저장된 데이터를 into BCD 데이터로 변환하고 PLC 가 저장된 데이터를 외부 장치로부터 BCD 포맷 7-세그먼트 표시로 디스플레이 할 때, 7-세그먼트 표시를 위해 그것을 전송하려고 사용합니다.
- X0=On 이면, K4X0(BCD 데이터)를 BIN 데이터로 변환하고 D100 으로 그것을 전송합니다. 그리고, D100 의 BIN 데이터를 BCD 데이터로 변환하고 K4Y20 으로 그것을 전송합니다.







API	연상기호			피연산자			기능			제어기							
20	D	ADD	P	<div>S<sub>1</sub></div>	<div>S<sub>2</sub></div>	<div>D</div>	덧셈			ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ADD, ADDP: 7 스텝  DADD, DADDP: 13 스텝					
	S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*						
	S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*						
	D							*	*	*	*	*	*	*	*						

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S<sub>1</sub>**: 피가산수    **S<sub>2</sub>**: 가수    **D**: 덧셈 결과

설명:

- 만일 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, **D** 가 장치 **F**를 사용한다면, 16-비트 명령에서만 사용 가능합니다..
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS** 시리즈 모델은 펄스 실행 명령(**ADDP**, **DADDP**)을 지원하지 않습니다.
- S<sub>1</sub> + S<sub>2</sub> = D**. BIN 데이터 **S<sub>1</sub>** 과 BIN 데이터 **S<sub>2</sub>** 의 덧셈을 실행하고 장치 **D** 에 덧셈 결과를 저장하십시오.
- 가장 중요한 비트는 기호 비트 0 과 1 입니다. 0 은 양수를 의미하고 1 은 음수를 의미합니다. 모든 계산은 수학적으로 처리됩니다. 예를들면, 3 + (-9) = -6.
- 플래그는 2 진수 덧셈으로 바뀝니다.

16-비트 명령:

- 만일 연산결과가 "0"이라면, Zero 플래그, M1020 는 ON 으로 설정됩니다.
- 만일 연산결과가 -32,768 를 초과한다면, 차용 플래그, M1021 은 ON 으로 설정됩니다.
- 만일 연산결과가 32,767 를 초과한다면, 자리올림 플래그, M1022 은 ON 으로 설정됩니다.

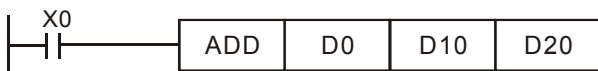
32-비트 명령:

- 만일 연산결과가 "0"이라면, Zero 플래그, M1020 은 ON 으로 설정됩니다.
  - 만일 연산결과가 -2,147,483,648 를 초과한다면, 차용 플래그, M1021 은 ON 으로 설정됩니다.
  - 만일 연산결과가 2,147,483,647 를 초과한다면, 자리올림 플래그, M1022 은 ON 으로 설정됩니다.
- M1020 (Zero 플래그). M1021 (차용 플래그). M1022 (자리올림 플래그). 자세한 것은 다음을 참조하십시오.

프로그램 예제 1:

16-비트 명령:

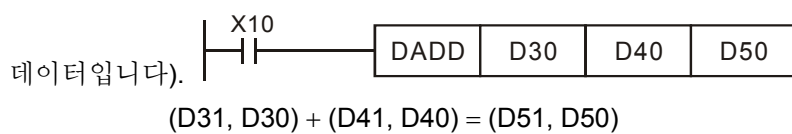
X0 가 ON 이면, 피가산수 D0 와 가수 D10 내에 보유한 데이터는 결합되고 그 합계는 결과 장치 D20 에 저장됩니다.



프로그램 예제 2:

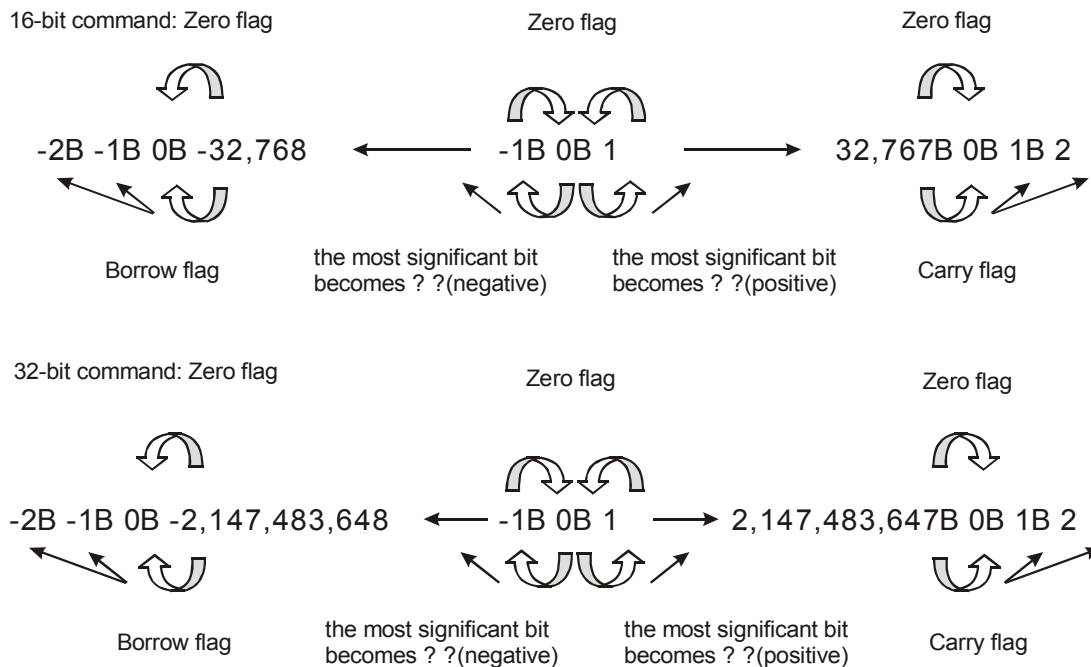
32-비트 명령:

X0 가 ON 이면, 피가산수 (D31, D30)와 가수 (D41, D40) 내에 보유한 데이터는 결합되고 그 합계는 결과 장치 (D51, D50)에 저장됩니다. (D30, D40, D50 은 하위 16-비트 데이터이고 D31, D41, D51 은 상위 16-비트



주의:

플래그 연산:



API	연상기호			피연산자			기능			제어기						
21	D	SUB	P	<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	<b>(D)</b>	뺄셈			ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SUB, SUBP: 7 스텝  DSUB, DSUBP: 13 스텝					
	S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*		*				
	S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*		*				
	D							*	*	*	*	*	*	*	*		*				

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

S<sub>1</sub>: 피감수    S<sub>2</sub>: 감수    D: 뺄셈 결과

설명:

- 만일 피연산자 S<sub>1</sub>, S<sub>2</sub>, D 가 장치 F 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(SUBP, DSUBP)을 지원하지 않습니다.
- S<sub>1</sub> - S<sub>2</sub> = D. BIN 데이터 S<sub>1</sub> 과 BIN 데이터 S<sub>2</sub> 의 뺄셈을 실행하고 그 뺄셈 결과를 장치 D 에 저장하십시오.
- 가장 중요한 비트는 부호 비트 0 and 1 입니다. 0 은 양수를 의미하고 1 은 음수를 의미합니다. 모든 계산은 수학적으로 처리됩니다.
- 플래그는 2 진수 뺄셈으로 바뀝니다.

16-비트 명령:

- 만일 연산결과가 "0"이라면, Zero 플래그, M1020 은 ON 으로 설정됩니다.
- 만일 연산결과가 -32,768 을 초과한다면, 차용 플래그, M1021 은 ON 으로 설정됩니다.
- 만일 연산결과가 32,767 을 초과한다면, 자리올림 플래그, M1022 은 ON 으로 설정됩니다.

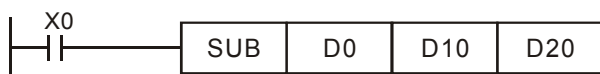
32-비트 명령:

- 만일 연산결과가 "0"이라면, Zero 플래그, M1020 은 ON 으로 설정됩니다.
  - 만일 연산결과가 -2,147,483,648 을 초과한다면, 차용 플래그, M1021 은 ON 으로 설정됩니다.
  - 만일 연산결과가 2,147,483,647 을 초과한다면, 자리올림 플래그, M1022 은 ON 으로 설정됩니다.
- SUB 명령의 플래그 연산은 이전 페이지에 ADD 명령의 플래그 연산을 참조하십시오.
  - M1020 (Zero 플래그). M1021 (차용 플래그). M1022 (자리올림 플래그). ADD 명령의 명령어 설명을 참조하십시오.

프로그램 예제 1:

16-비트 명령:

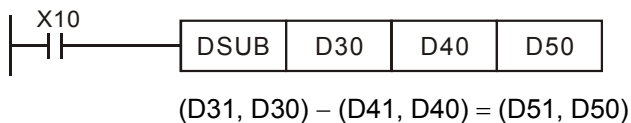
X0 가 ON 이면, 감수 D10 내에 있는 데이터는 피감수 D0 내에 있는 데이터로부터 뺄셈을 하고 이 계산의 결과는 결과 장치 D20 에 저장됩니다.



프로그램 예제 2:

32-비트 명령:

X0 가 ON 이면, 감수 (D41, D40) 내에 있는 데이터는 피감수 (D31, D30) 내에 있는 데이터로부터 뺄셈을 하고 이 계산 결과는 결과 장치 (D51, D50)에 저장됩니다. (D30, D40, D50 는 하위 16-비트 데이터이고 D31, D41, D51 는 상위 16-비트 데이터).

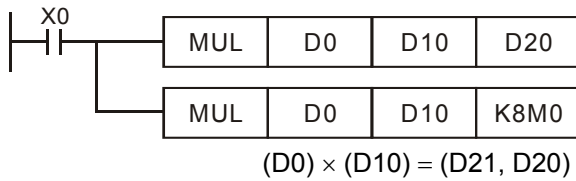


타입 연상부호	비트 장치				워드 장치											프로그램 스텝
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MUL, DMULP: 7 스텝  DMUL, DMULP: 13 스텝
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*	
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*		

## ***DVP-PLC Application Manual***

16-비트 명령:

16-비트 데이터 소스, D10 는 또 다른 16-비트 데이터 소스와, D0 를 곱하고 그 합계는 32-비트 결과, D20 이 됩니다. 상위 16-비트 데이터는 D21 에 저장되고 하위는 D20 에 저장됩니다. 그 결과의 극성은 가장 중요한 비트의 OFF/ON 으로 표시됩니다. OFF 는 양수 0 의 값을 의미하고 ON 은 음수 1 의 값을 의미합니다.



API	연상기호			피연산자			기능			제어기							
23	D	DIV	P	S <sub>1</sub>	S <sub>2</sub>	D	나눗셈			ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DIV, DIVP: 7 스텝 DDIV, DDIVP: 13 스텝					
	S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*						
	S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*						
	D							*	*	*	*	*	*	*							

펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

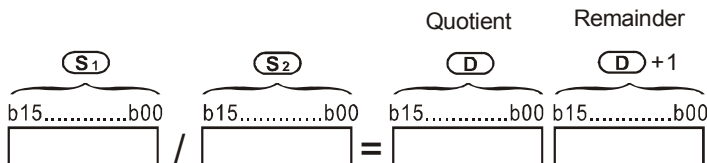
피연산자 s:

S<sub>1</sub>: 피제수    S<sub>2</sub>: 제수    D: 몫과 나머지

설명:

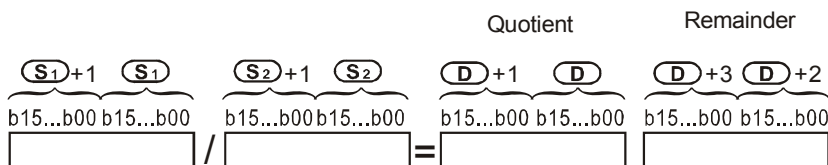
- 만일 피연산자 S<sub>1</sub>, S<sub>2</sub> 가 장치 F 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 만일 피연산자 D 가 장치 E 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 16-비트 명령에서, 피연산자 D 는 2 개의 연속장치를 점유합니다.
- 32-비트 명령에서, 피연산자 D 는 4 개의 연속장치를 점유합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(DIVP, DDIVP)을 지원하지 않습니다.
- S<sub>1</sub> ÷ S<sub>2</sub> = D. BIN 데이터 S<sub>1</sub> 와 BIN 데이터 S<sub>2</sub> 의 나눗셈을 실행하고 그 결과를 장치 D 에 저장하십시오.  
16-비트와 32-비트 명령에서 S<sub>1</sub>, S<sub>2</sub>, D 의 연산결과의 극성표시에 주의하십시오.
- 이 명령은 제수가 "0"일 때 실행되지 않습니다. 그리고 그 플래그 M1067, M1068 이 On 이 되고 D1067 은 에러 코드 "0E19"을 기록 할 것입니다.

16-비트 명령:



D 가 비트 장치라면, 그것은 16-비트 결과를 산출하기 위해 K1~K4 를 지정할 수 있고, 2 개의 연속그룹을 점유할 수 있습니다. 연산 결과에 관하여, 그 몫과 나머지는 저장됩니다. ES 모델에서, 그 연산결과는 나머지없이 단지 몫만 나타날 것입니다.

32-비트 명령:



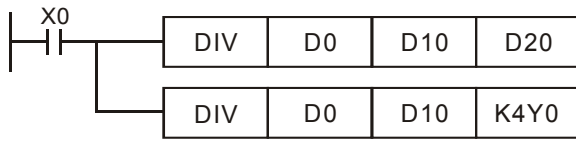
D 가 비트 장치일때, K1~K8 은 32 비트 생성을 위해 지정될 수 있고, 몫과 나머지를 산출하기 위해 32 비트 데이터의 2 그룹을 계속적으로 점유할 수 있습니다.

프로그램 예제:

X0 가 ON 일 때, 첫번째 소스 D0 (제수)는 두번째 소스 D10 (피제수)에 의해 나누어집니다. 그 몫은 D20 에



저장되기 위해 지정되고 그 나머지는 D21에 저장되기 위해 지정됩니다. 그 결과의 극성은 최상위비트의 OFF/ON에 의해 표시됩니다. OFF는 양수 값을 의미하고 ON은 음수값을 의미합니다.



API	연상기호			피연산자	기능	제어기						
24	D	INC	P	<b>(D)</b>	증분	ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치												프로그램 스텝			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		INC, INCP: 3 스텝			
D								*	*	*	*	*	*	*	*		DINC, DINCP: 5 스텝			

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**D:** 목적지

설명:

- 만일 피연산자 **D**가 장치 **F**를 사용한다면, 그것은 16-비트 명령에서만 사용 가능합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (**INCP**, **DINCP**)을 지원하지 않습니다.
- 만일 명령이 펄스 실행 명령이 아니라면, **INC** 명령이 각 사이클을 스캔할 때 마다 그 명령의 모든 실행에서 목적지 장치 **D**의 값에 “1”이 더해집니다. 이 명령은 보통 펄스 실행 (**INCP**, **DINCP**)을 합니다.
- 16-비트 명령에서, +32,767에 도달하면, “1”이 더해지고 목적지 장치에 -32,768 값을 적을 것입니다.  
32-비트 명령에서, +2,147,483,647에 도달하면, “1”이 더해지고 목적지 장치에 -2,147,483,648 값을 적을 것입니다.
- 플래그 M1020~M1022는 이 명령의 연산결과에 영향을 받지 않을 것입니다.

프로그램 예제:

X0가 ON일 때, D0내용은 1더하기를 실행 할 것입니다.



API	연상기호			피연산자	기능	제어기						
25	D	DEC	P	<b>(D)</b>	감량	ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치										프로그램 스텝	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	DEC, DECP: 3 스텝 DDEC, DDECP: 5 스텝

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**D:** 목적지

설명:

- 만일 피연산자 **D**가 장치 **F**를 사용한다면, 16-비트 명령에서만 사용 가능합니다..
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- 만일 명령이 펄스 실행 명령을 하지 않는다면, **INC** 명령이 각 사이클을 스캔할 때 마다 그 명령의 모든 실행에서 목적지 장치 **D**의 값에 “1”을 빼게 됩니다. 이 명령은 보통 펄스 실행 (**INCP**, **DINCP**)을 합니다.
- 16-비트 명령에서, -32,768에 도달될 때, “1”을 빼고 목적지 장치에 +32,767 값을 적을 것입니다. 32-비트 명령에서, -2,147,483,648에 도달될 때, “1”을 빼고 목적지 장치에 +2,147,483,647 값을 적을 것입니다.
- 플래그 M1020~M1022는 이 명령의 연산결과에 의해 영향을 받지 않을 것입니다.

프로그램 예제:

X0가 ON일 때, D0의 내용은 1 빼기를 실행할 것입니다.



API	연상기호			피연산자 s			기능			제어기							
26	W	AND		P	S <sub>1</sub>	S <sub>2</sub>	D	논리곱			ES	EX	SS	SA	SX	SC	EH
D																	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WAND, WANDP: 7 스텝  DAND, DANDP: 13 스텝					
					*	*	*	*	*	*	*	*	*	*	*						
					*	*	*	*	*	*	*	*	*	*	*						
								*	*	*	*	*	*	*	*						

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자 s:

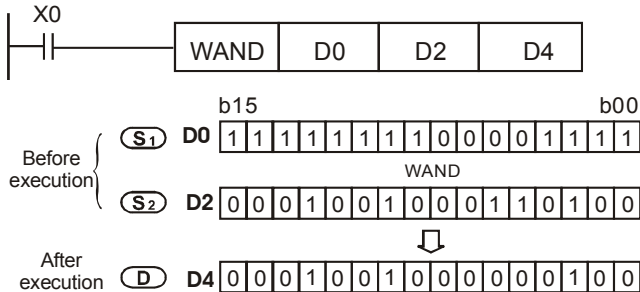
S<sub>1</sub>: 첫번째 데이터 소스    S<sub>2</sub>: 두번째 데이터 소스    D: 연산 결과

설명:

- 만일 피연산자 S<sub>1</sub>, S<sub>2</sub>, D 가 장치 F 를 사용한다면, 16-비트 명령에서만 사용 가능합니다..
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행(WANDP, DANDP)을 지원하지 않습니다.
- 데이터 소스 장치 S<sub>1</sub> 과 S<sub>2</sub>의 논리곱을 실행하고 연산결과를 장치 D 에 저장하십시오.
- 일반 연산 규칙: 만일 데이터 소스 장치내에 있는 비트중 하나가 “0”이라면, 연산결과는 또한 “0”입니다.

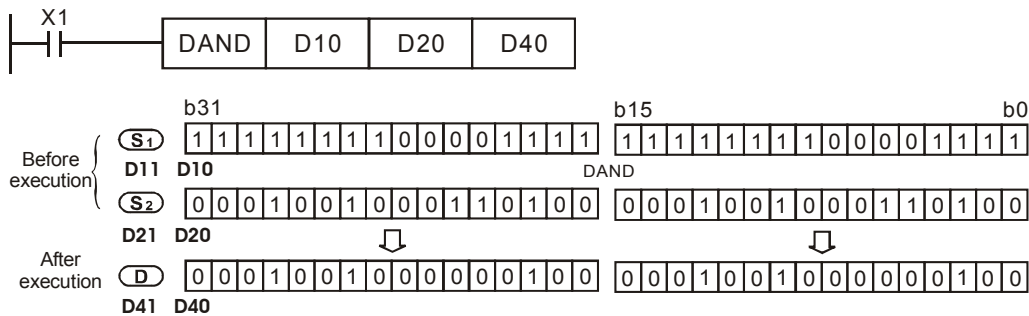
프로그램 예제 1:

X0 가 ON 일 때, 16-비트 데이터 소스 장치 D0 와 D2 는 분석되고 논리 WAND 명령의 연산결과는 장치 D4 에 저장됩니다.



프로그램 예제 2:

X1 이 ON 일 때, 32-비트 데이터 소스 장치 (D11, D10) 과 (D21, D20)은 분석되고 논리 DAND 명령의 연산결과는 장치 (D41, D40)에 저장됩니다.



API	연상기호			피연산자			기능			제어기								
27	W	OR		P	S <sub>1</sub>	S <sub>2</sub>	D	논리합			ES EX SS SA SX SC EH							
D																		

타입 연상부호	비트 장치				워드 장치										프로그램 스텝													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WOR, WOP: 7 스텝												
	S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*	DOR, DORP: 13 스텝											
	S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*												
	D							*	*	*	*	*	*	*	*	*												

펄스							16-비트							32-비트													
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

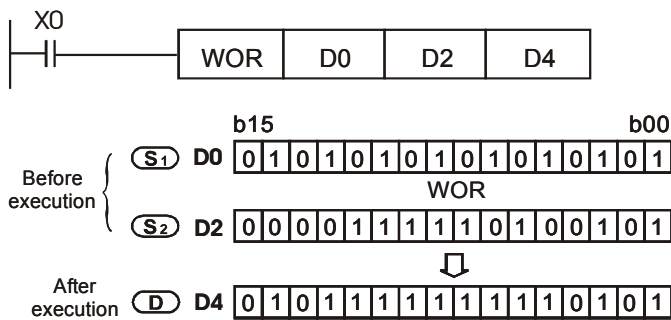
**S<sub>1</sub>**: 첫번째 데이터 소스    **S<sub>2</sub>**: 두번째 데이터 소스    **D**: 연산 결과

설명:

- 만일 피연산자 S<sub>1</sub>, S<sub>2</sub>, D 가 장치 F 를 사용한다면, 16-비트 명령에서만 사용 가능합니다..
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (WORP, DORP)을 지원하지 않습니다.
- 데이터 소스 장치 **S<sub>1</sub>** 과 **S<sub>2</sub>**의 논리합을 실행하고 장치 **D** 에 연산결과를 저장하십시오.
- 일반 연산 규칙: 만일 데이터 소스 장치내에 있는 비트중 하나가 “1”이라면, 연산결과는 또한 “1”입니다.

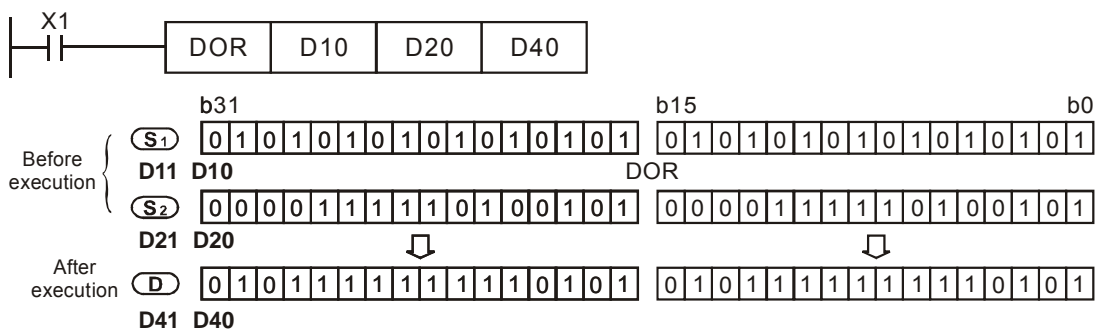
프로그램 예제 1:

X0 가 ON 일 때, 16-비트 데이터 소스 장치 D0 와 D2 는 분석되고 논리 WOR 명령의 연산결과는 장치 D4 에 저장됩니다.



프로그램 예제 2:

X1 이 ON 일 때, 32-비트 데이터 소스 장치 (D11, D10) 와 (D21, D20) 가 분석되고 논리 DOR 명령의 연산결과가 장치(D41, D40)에 저장됩니다.



API	연상기호			피연산자			기능			제어기								
28	W	XOR		P	S <sub>1</sub>	S <sub>2</sub>	D		배타적 논리합			ES	EX	SS	SA	SX	SC	EH
D																		

타입 연상부호	비트 장치				워드 장치										프로그램 스텝								
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WXOR, WXORP: 7 스텝 DXOR, DXORP: 13 스텝							
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*								
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*								
D								*	*	*	*	*	*	*	*								

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자 s:

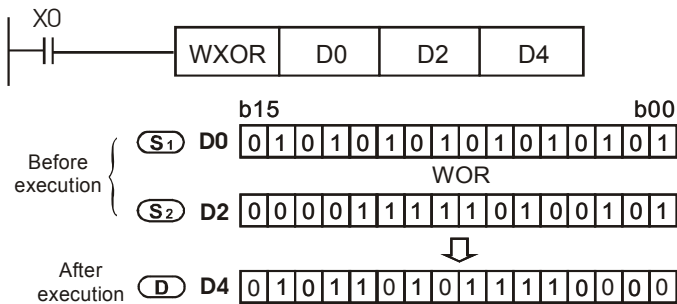
**S<sub>1</sub>**: 첫번째 데이터 소스    **S<sub>2</sub>**: 두번째 데이터 소스    **D**: 연산 결과

설명:

- 만일 피연산자 S<sub>1</sub>, S<sub>2</sub>, D 장치 F 를 사용, 16-비트 명령에서만 사용 가능합니다..
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행(WXORP, DXORP)을 지원하지 않습니다.
- 데이터 소스 장치 S<sub>1</sub> 과 D 의 배타적 논리 합을 실행하고, 장치 D 에 연산결과를 저장하십시오.
- 일반 연산 규칙: 만일 두개의 데이터 소스 장치내에 있는 두개의 비트가 동일하다면, 그 연산결과는 “0”입니다. 하지만 만일 두개의 데이터 소스 장치내에 있는 두개의 비트가 다르다면, 그 연산결과는 “1”이 됩니다.

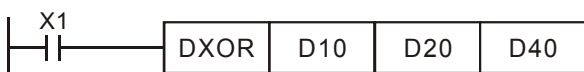
프로그램 예제 1:

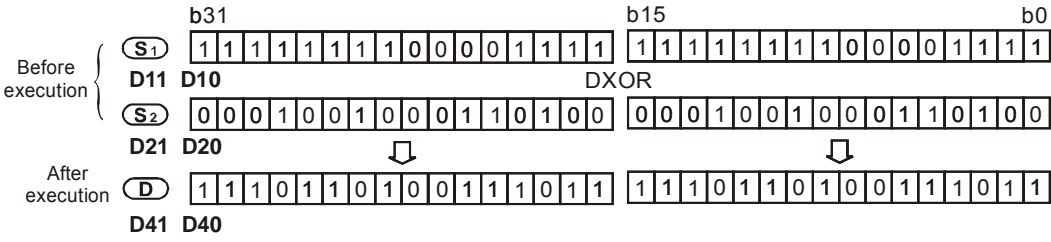
X0 가 ON 일 때, 16-비트 데이터 소스 장치 D0 와 D2 는 분석되어 논리 WXOR 명령의 연산결과는 장치 D4 에 저장됩니다.



프로그램 예제 2:

X1 이 ON 일 때, 32-비트 데이터 소스 장치 (D11, D10) 와 (D21, D20) 는 분석되어 논리 DXOR 명령의 연산결과는 장치 (D41, D40) 에 저장됩니다.





API	연상기호			피연산자	기능	제어기						
29	D	NEG	P	<b>D</b>	음의 (2 의 보수)	ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치												프로그램 스텝			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		NEG, NEGP: 3 스텝			
D								*	*	*	*	*	*	*	*		DNEG, DNEGP: 5 스텝			

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**D:** 2 의 보수의 연산을 저장

설명:

- 만일 피연산자 **D** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (**NEGP**, **DNEGP**)을 지원하지 않습니다.
- D:** 일단 명령이 실행되면, 지정된 장치, **D** 는 2 의 보수로서 제공될 것입니다.
- 이 명령은 음의 **BIN** 값을 양수로 변환 할 수 있습니다. 즉, 그것의 절대값을 구하기 위한 것입니다.
- 이 명령은 보통 펄스 실행 (**NEGP**, **DNEGP**)입니다.

프로그램 예제 1:

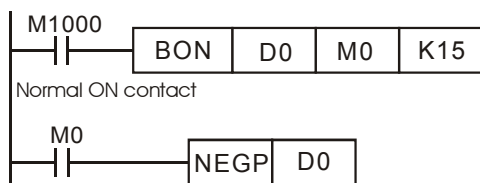
X0 이 OFF → ON 으로 되면, D10 내용의 모든 비트는 카운터(0→1, 1→0)되어 1 을 더하고 나서 원시 레지스터 D10 에 저장될 것입니다.



프로그램 예제 2:

음의 값의 절대값 구하기:

- D0 의 15<sup>th</sup> 비트 가 “1”일 때, M0 는 ON 이 됩니다. (D0 은 음의 값).
- M0 가 ON 일 때, 그 D0 의 절대값은 NEG 명령을 사용해서 구할 수 있습니다.

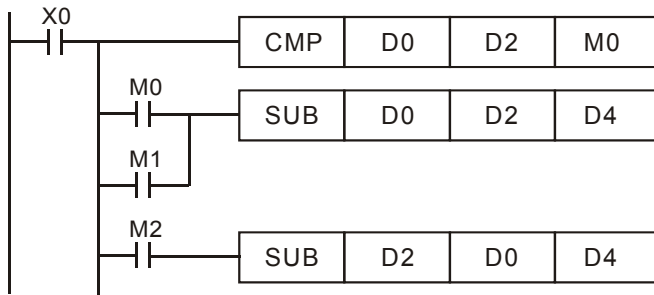


프로그램 예제 3:

뺄셈의 결과로 절대 값 구하기:

- D0>D2 일 때, M0=ON.
- D0=D2 일 때, M1=ON.
- D0<D2 일 때, M2=ON.
- 그리고나면 D4 는 구해질 것이고 양의 값으로 될 것입니다.





**주의:**

음의 값과 절대값의 표시

- 레지스터의 최상위비트 내용은 양과 음의 값을 나타냅니다. 그 내용이 “0”일때 그것은 양의 값이고 그 내용이 “1”일 때 그것은 음의 값입니다.
- 만일 그것이 is a 음의 값이라면, 절대 값은 NEG 명령(API 29)을 사용해서 구할 수 있습니다.

(D0=2)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=1)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=0)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=-1)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=1$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=-2)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=2$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=-3)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=3$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=-4)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=4$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=-5)

1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=5$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

⋮

⋮

(D0=-32,765)

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=32,765$

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=-32,766)

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=32,766$

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=-32,767)

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=32,767$

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0=-32,768)

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{(D0)}+1=-32,768$

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Max. absolute value is 32,767

API	연상기호			피연산자		기능		제어기							
30	D	ROR	P	D n		오른쪽으로 회전		ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치												프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ROR, RORP: 5 스텝 DROR, DRORP: 9 스텝								
	D							*	*	*	*	*	*	*	*									
n					*	*																		

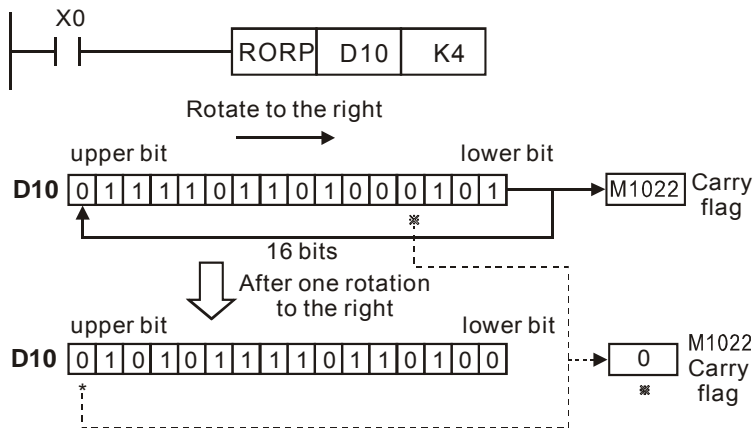
펄스								16-비트								32-비트								
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		

설명:

- 만일 피연산자 D가 장치 F을 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 만일 피연산자 D가 KnY, KnM, KnS와 같이 지정되면, 단지 K4 (16-비트)와 K8 (32-비트)가 유효합니다.
- 기본 조건:  $1 \leq n \leq 16$  (16-비트),  $1 \leq n \leq 32$  (32-비트).
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (RORP, DRORP)을 지원하지 않습니다.
- D: 회전 장치 (목적지 장치) n: 한번 회전한 비트위치.
- 장치 D의 비트 패턴은 n 비트 위치를 그 명령의 모든 연산에서 오른쪽으로 회전됩니다.
- 이 명령은 보통 펄스 실행 (RORP, DRORP)을 합니다.
- M1022 (자리올림 플래그).

프로그램 예제:

X0가 OFF에서 ON으로 되면, D10의 16 비트 데이터는 4 비트를 다이어그램에서 보인 것처럼 오른쪽으로 회전할 것입니다. 처음 D10에 위치한 b3는 자리올림 플래그 (CY) M1022로 이동될 것입니다.



API	연상기호			피연산자		기능				제어기							
31	D	ROL	P	<div><div>D</div><div>n</div></div>		왼쪽으로 회전				<div><div>ES</div><div>EX</div><div>SS</div><div>SA</div><div>SX</div><div>SC</div><div>EH</div></div>							

타입 연상부호	비트 장치				워드 장치										프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ROL, ROLP: 5 스텝  DROL, DROLP: 9 스텝						
	D							*	*	*	*	*	*	*	*							
n					*	*																

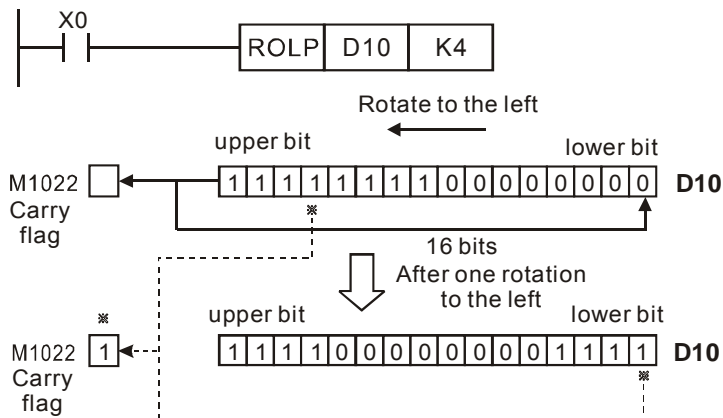
펄스								16-비트								32-비트								
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		

설명:

- 만일 피연산자 D가 장치 F를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 만일 피연산자 D가 KnY, KnM, KnS 과 같이 지정된다면, 단지 K4 (16-비트) 와 K8 (32-비트)이 유효합니다.
- 기본 조건:  $1 \leq n \leq 16$  (16-비트),  $1 \leq n \leq 32$  (32-비트).
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (ROLP, DROLP)을 지원하지 않습니다.
- D: 회전 장치 (목적지 장치) n: 한번 회전한 비트 위치
- 장치 D의 비트패턴은 n 비트 위치를 명령의 모든 연산의 왼쪽으로 회전합니다..
- 이 명령은 보통 펄스 실행(ROLP, DROLP)을 합니다.
- M1022 (자리올림 플래그).

프로그램 예제:

X0가 OFF → ON으로 되면, D10의 16 비트 데이터는 다이어그램에서 보인 것처럼, 4 비트를 왼쪽으로 회전할 것이고, 처음에 D10에 위치한 b12는 자리올림 플래그 (CY) M1022로 이동될 것입니다.



API	연상기호			피연산자	기능	제어기						
32	D	RCR	P	<div><div>D</div><div>n</div></div>	자리올림 플래그를 가지고오른쪽으로 회전	ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	RCR, RCRP: 5 스텝  DRCR, DRCRP: 9 스텝					
	D							*	*	*	*	*	*	*	*						
n					*	*															

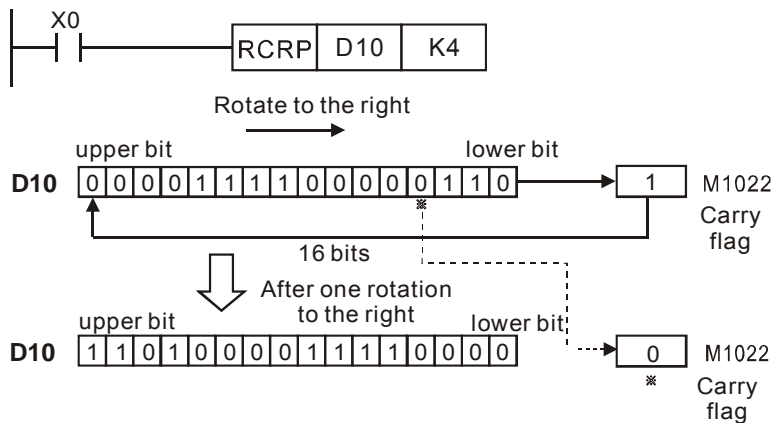
펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

설명:

- 만일 피연산자 **D** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 만일 피연산자 **D** 가 KnY, KnM, KnS 와 같이 지정한다면, 단지 K4 (16-비트)와 K8 (32-비트)은 유효합니다.
- 기본 조건:  $1 \leq n \leq 16$  (16-비트),  $1 \leq n \leq 32$  (32-비트).
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (RCRP, DRCRP)을 지원하지 않습니다.
- D**: 회전 장치 (목적지 장치) **n**: 한번 회전후 비트 위치
- 부가 자리올림 플래그 (M1022)를 가진 장치 **D** 의 비트 패턴은 그 명령의 모든 연산에서 오른쪽으로 **n** 비트 위치를 회전합니다.
- 이 명령은 보통 펄스 실행(RCRP, DRCRP)을 합니다.
- M1022 (자리올림 플래그).

프로그램 예제:

X0 가 OFF 에서 ON 으로 되면, 부가 자리올림 플래그 (M1022)를 가진 D10 의 16 비트 데이터는, 다이어그램에서 보인 것처럼 오른쪽으로 4 비트를 회전할 것입니다. 그리고 처음에 D10 에 위치한 b3 는 자리올림 플래그 M1022 로 이동될 것입니다. 그리고, 처음의 자리올림 플래그 M1022 의 내용은 b12 의 비트로 이동될 것입니다.



API	연상기호			피연산자	기능	제어기							
33	D	RCL	P	<div><div>D</div><div>n</div></div>	자리올림 플래그를 가지고오른쪽으로 회전	ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	RCL, RCLP: 5 스텝						
D								*	*	*	*	*	*	*	*	DRCL, DRCLP: 9 스텝						
n					*	*																

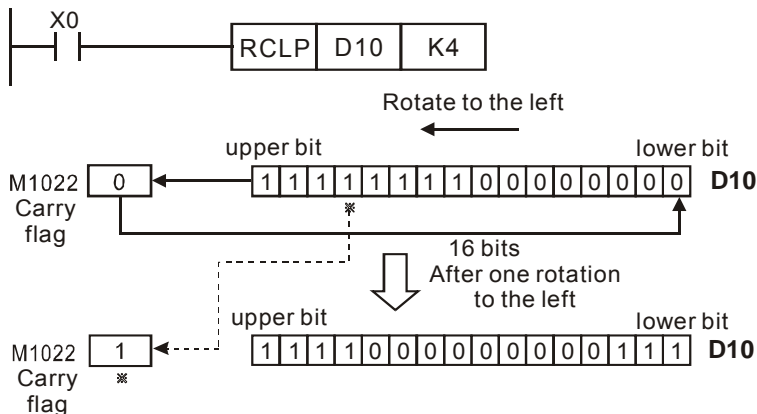
펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

## 설명:

- 만일 피연산자 D가 장치 F를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 만일 피연산자 D가 KnY, KnM, KnS와 같이 지정된다면, 단지 K4 (16-비트) 과 K8 (32-비트)이 유효합니다.
- 기본 조건:  $1 \leq n \leq 16$  (16-비트),  $1 \leq n \leq 32$  (32-비트).
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (RCLP, DRCLP)을 지원하지 않습니다.
- D**: 회전 장치 (목적지 장치) **n**: 한번 회전 후 비트 위치.
- 이 명령은 보통 펄스 실행 (RCLP, DRCLP)을 합니다.
- M1022 (자리올림 플래그).

## 프로그램 예제:

X0가 OFF에서 ON으로 되면, 부가 자리올림 플래그 (M1022)가 있는 D10의 16 비트 데이터는, 다이어그램에서 보인 것처럼 왼쪽으로 4 비트를 회전할 것입니다. 그리고 처음에 D10에 위치한 b12는 자리올림 플래그 M1022로 이동될 것이고, 자리올림 플래그 M1022의 처음 내용은 b3의 비트로 이동 될 것입니다.



API	연상기호			피연산자	기능	제어기							
34		SFTR	P	<div>S</div> <div>D</div> <div>n<sub>1</sub></div> <div>n<sub>2</sub></div>	지정된 장치의 데이터를 오른쪽으로 쉬프트	ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치											프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SFTR, SFTRP: 9 스텝							
S	*	*	*	*																			
D		*	*	*																			
n <sub>1</sub>					*	*																	
n <sub>2</sub>					*	*																	

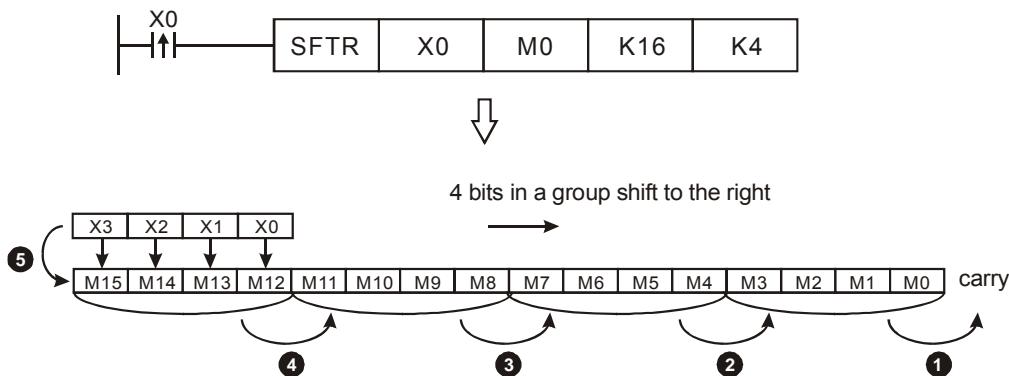
펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

설명:

- 기본 조건:  $1 \leq n_1 \leq 1024$ ,  $1 \leq n_2 \leq n_1$ .
- ES/EX/SS 시리즈 모델:  $1 \leq n_2 \leq n_1 \leq 512$ .
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (SFTRP)을 지원하지 않습니다.
- S**: 쉬프트 장치 (소스 장치)의 시작 넘버 **D**: 지정된 쉬프트 장치 (목적지 장치)의 시작 넘버  
**n<sub>1</sub>**: 데이터 길이의 지정된 비트 스텝 **n<sub>2</sub>**: 한번 쉬프트 후 비트 위치
- 장치 **S**의 **n<sub>1</sub>** 데이터 비트를 **n<sub>1</sub>** 비트에 의해 오른쪽으로 쉬프트 하십시오. **D**로 시작한 **n<sub>2</sub>** 비트는 오른쪽으로 쉬프트 됩니다.
- 이 명령은 보통 펄스 실행(SFTRP)을 합니다.
- ES 시리즈 모델은 인덱스 레지스터 E와 F를 지원하지 않습니다.

프로그램 예제:

- X0가 rising-edge 일 때, the of M0~M15의 16 비트 데이터는 오른쪽으로 4 비트 쉬프트 할 것입니다.
- SFTR 명령을 한번 스캔하기 위해 다음 ❶~❺ 단계를 참고하십시오.
  - M3~M0 → 이동
  - M7~M4 → M3~M0
  - M11~M8 → M7~M4
  - M15~M12 → M11~M8
  - X3~X0 → M15~M12 완료



API	연상기호			피연산자				기능		제어기													
35		SFTL	P	<b>S</b>	<b>D</b>	<b>n<sub>1</sub></b>	<b>n<sub>2</sub></b>	지정된 장치의 데이터를 왼쪽으로 쉬프트		<table><tr><td>ES</td><td>EX</td><td>SS</td><td>SA</td><td>SX</td><td>SC</td><td>EH</td></tr></table>							ES	EX	SS	SA	SX	SC	EH
ES	EX	SS	SA	SX	SC	EH																	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SFTL, SFTLP: 9 스텝
S	*	*	*	*												
D		*	*	*												
n <sub>1</sub>					*	*										
n <sub>2</sub>					*	*										

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

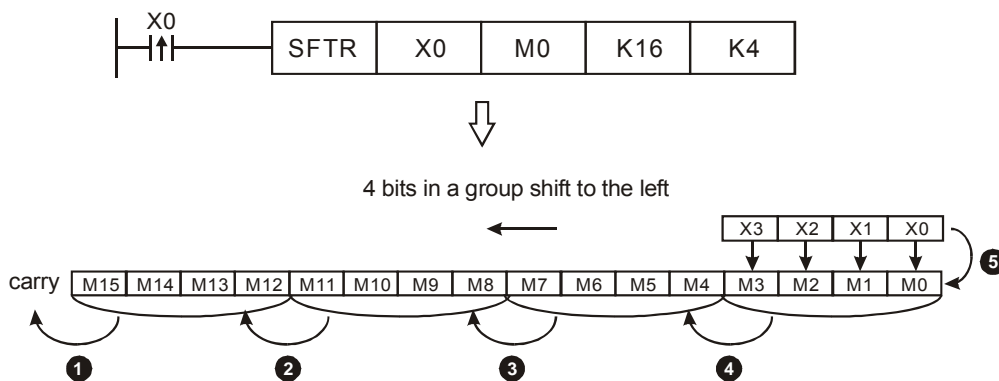
## 설명:

- 기본 조건:  $1 \leq n_1 \leq 1024$ ,  $1 \leq n_2 \leq n_1$ .
- ES/EX/SS 시리즈 모델:  $1 \leq n_2 \leq n_1 \leq 512$ .
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 (SFTLP)을 지원하지 않습니다.
- S**: 쉬프트 장치 (소스 장치)의 시작 넘버 **D**: 지정된 쉬프트 장치 (목적지 장치)의 시작 넘버  
**n<sub>1</sub>**: 데이터 길이의 지정된 비트 스텝 **n<sub>2</sub>**: 한번 쉬프트 후 비트
- 쉬프트 **s of** 장치 **S**의 **n<sub>2</sub>** 데이터 비트를 **n<sub>1</sub>** 비트에 의해 왼쪽으로 쉬프트 하십시오. **D**로 시작하는 **n<sub>2</sub>** 비트는 왼쪽으로 쉬프트 됩니다.
- 이 명령은 보통 펄스 실행 (SFTLP)을 합니다.
- ES 시리즈 모델은 인덱스 레지스터 E와 F를 지원하지 않습니다.

## 프로그램 예제:

- X0가 rising-edge에 있을 때, M0~M15의 16비트 데이터는 왼쪽으로 4비트 회전할 것입니다.
- SFTL 명령을 한번 스캔하기 위해 다음 ①~⑤ 단계를 참고하십시오.

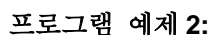
- ① M15~M12 → 이동
- ② M11~M8 → M15~M12
- ③ M7~M4 → M11~M8
- ④ M3~M0 → M7~M4
- ⑤ X3~X0 → M3~M0 완료



펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

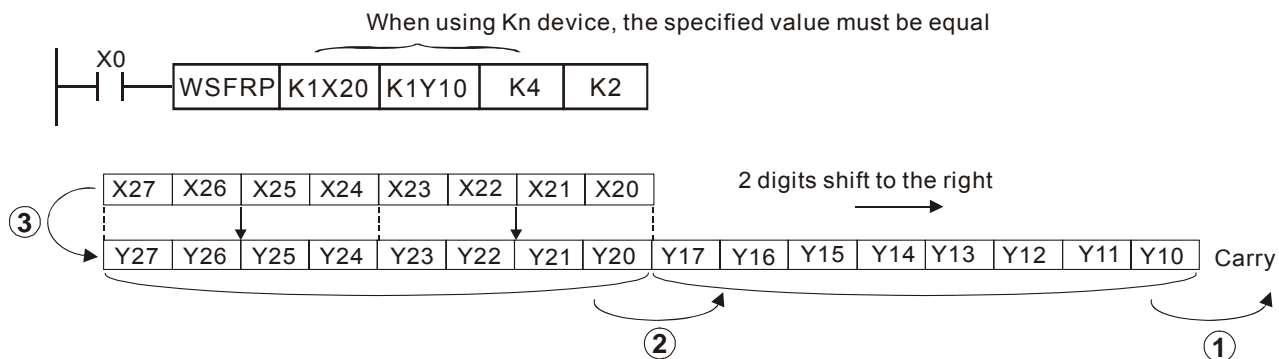
1. 피연산자 **S** (소스)와 **D** (목적지)와 같은 비트 장치를 사용하면, 지정된 장치는 동일해야 합니다. 예를 들면, 한 종류는  $K_nX$ ,  $K_nY$ ,  $K_nM$ ,  $K_nS$  이고 그 나머지 종류는 **T**, **C**, **D** 입니다.
2. 피연산자 **S** (소스)와 **D** (목적지)와 같은 비트 장치를 사용하면,  $K_n$  값은 동일해야 합니다.
3. 기본 조건:  $1 \leq n_1 \leq 512$ ,  $1 \leq n_2 \leq n_1$
4. 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
5. ES/EX/SS 시리즈 모델은 실행 (**WSFR**, **WSFRP**)을 지원하지 않습니다.
6. **S**: 쉬프트 장치 (소스 장치)의 시작 넘버    **D**: 지정된 쉬프트 장치 (목적지 장치)의 시작 넘버  
 **$n_1$** : 데이터 길이의 지정된 비트 스택     **$n_2$** : 한번 쉬프트 후 워드
7. 장치 **S** 의  **$n_1$**  데이터 워드를  **$n_1$**  워드에 의해 오른쪽으로 쉬프트 하십시오. **D** 로 시작하는  **$n_2$**  워드는 오른쪽으로 쉬프트 합니다.
8. 이 명령은 보통 펄스 실행(**WSFRP**)을 합니다.

1. X0 가 OFF 에서 ON 으로 되면, D20~D35 의 16 레지스터 데이터는 쉬프트 구역을 병렬로 하고 4 개의 레지스터를 오른쪽으로 쉬프트 합니다.
2. WSFR 명령을 한번 실행하기 위해 다음 ❶~❺ 단계를 참고하십시오.
  - ❶ D23~D20 → 이동
  - ❷ D27~D24 → D23~D20
  - ❸ D31~D28 → D27~D24
  - ❹ D35~D32 → D31~D28
  - ❺ D13~D10 → D35~D32 완료





1. X0 가 OFF 에서 ON 으로 되면, Y10~Y27 의 워드 레지스터 데이터는 쉬프트 구역을 병렬로 하고 오른쪽으로 2 digit 을 쉬프트 합니다.
2. WSFR 명령을 한번 쉬프트 하기 위해 다음 ❶~❸ 단계를 참고하십시오.
  - ❶ Y17~Y10 → 이동
  - ❷ Y27~Y20 → Y17~Y10
  - ❸ X27~X20 → Y27~Y20 완료



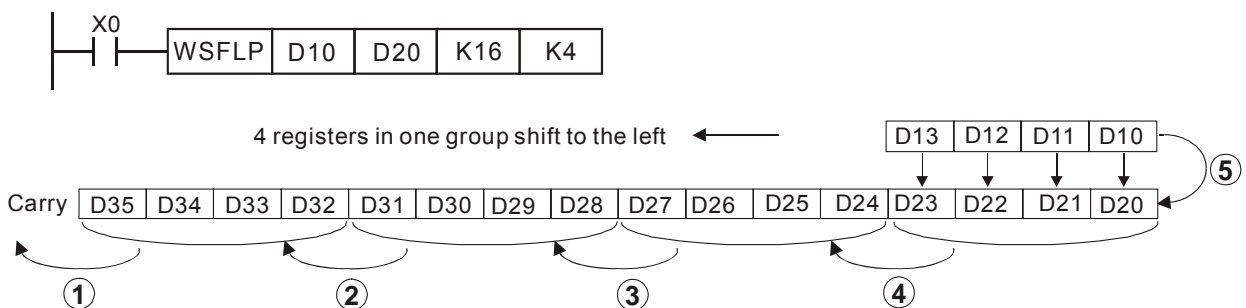
API	연상기호			피연산자				기능		제어기															
37		WSFL	P	<div>S</div>	<div>D</div>	<div>n<sub>1</sub></div>	<div>n<sub>2</sub></div>	왼쪽으로 레지스터를 쉬프트		<div>ES</div>	<div>EX</div>	<div>SS</div>	<div>SA</div>	<div>SX</div>	<div>SC</div>	<div>EH</div>									
타입 연상부호	비트 장치				워드 장치										프로그램 스텝										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WSFR, WSFRP: 9 스텝									
	S						*	*	*	*	*	*	*												
	D							*	*	*	*	*	*												
	n <sub>1</sub>				*	*																			
	n <sub>2</sub>				*	*																			
					펄스				16-비트				32-비트												
					<div>ES</div>	<div>EX</div>	<div>SS</div>	<div>SA</div>	<div>SX</div>	<div>SC</div>	<div>EH</div>	<div>ES</div>	<div>EX</div>	<div>SS</div>	<div>SA</div>	<div>SX</div>	<div>SC</div>	<div>EH</div>	<div>ES</div>	<div>EX</div>	<div>SS</div>	<div>SA</div>	<div>SX</div>	<div>SC</div>	<div>EH</div>

설명:

- 피연산자 **S** (소스)와 **D** (목적지) 와 같은 비트 장치를 사용할 때, 지정된 장치는 동일해야 합니다.  
예를들면, 한 종류는  $K_nX$ ,  $K_nY$ ,  $K_nM$ ,  $K_nS$  이고 그 나머지는  $T$ ,  $C$ ,  $D$  입니다.
- 피연산자 **S** (소스)와 **D** (목적지)와 같은 비트 장치를 사용할 때,  $Kn$  값은 동일 해야 합니다.
- 기본 조건:  $1 \leq n_1 \leq 512$ ,  $1 \leq n_2 \leq n_1$
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 실행(WSFL, WSFLP)을 지원하지 않습니다.
- S**: 쉬프트 장치 (소스 장치)의 시작 넘버 **D**: 지정된 쉬프트 장치 (목적지 장치)의 시작 넘버  
 $n_1$ : 데이터 길이의 지정된비트 스텝  $n_2$ : 한번 쉬프트 후 워드
- 쉬프트  $s$  of 장치 **S** 의  $n_1$  데이터 워드를  $n_2$  워드에 의해 왼쪽으로 쉬프트 하십시오. **D** 로 시작하는  $n_2$  워드는 왼쪽으로 쉬프트 됩니다.
- 이 명령은 보통 펄스 실행 (WSFLP)을 합니다.

프로그램 예제:

- X0 가 OFF 에서 ON 으로 되면, D20~D35 의 16 레지스터 데이터는 쉬프트 구역을 병렬로 하고 4 개의 레지스터를 오른쪽으로 쉬프트 합니다.
- WSFL 명령을 한번 쉬프트 하기 위해 다음 ①~⑤ 단계를 참고하십시오.
  - D35~D32 → 이동
  - D31~D28 → D35~D32
  - D27~D24 → D31~D28
  - D23~D20 → D27~D24
  - D13~D10 → D23~D20 완료



API	연상기호			피연산자			기능			제어기							
38		SFWR	P	<div>S</div> <div>D</div> <div>n</div>			쉬프트 레지스터 쓰기			ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SFWR, SFWRP: 7 스텝					
S					*	*	*	*	*	*	*	*	*	*	*						
D								*	*	*	*	*	*								
n					*	*															

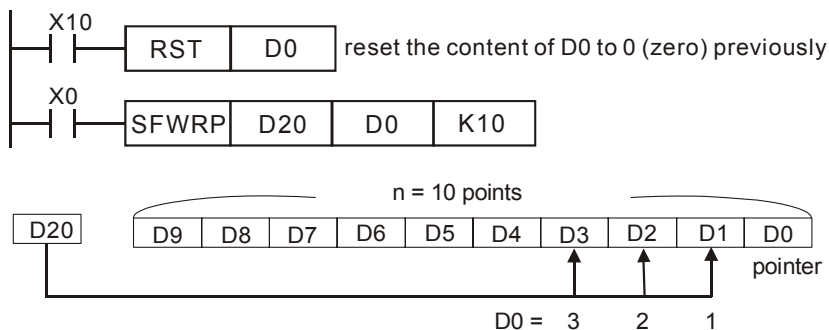
펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

**설명:**

- 기본 조건:  $2 \leq n \leq 512$ .
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 실행 (SFWR, SFWRP)을 지원하지 않습니다.
- S**: 데이터가 적힌 소스 장치 **D**: Head 어드레스 장치 **n**: 데이터 길이
- n**은 첫번째-in/첫번째-OUT 스택의 길이 이고 목적지 장치 **D**는 첫번째-in/첫번째-OUT 스택의 head 어드레스 장치입니다. 포인터 처럼 첫번째 넘버 장치 **D**를 사용하고 이 명령을 실행할 때 포인터의 내용 값에 1을 더하십시오. **S**로 지정된 장치의 내용은 첫번째-in/첫번째-OUT 스택의 포인터 **D**로 지정된 위치를 적게됩니다. 만일 포인터 **D**의 내용이 "n-1" 값을 초과하면, 첫번째-in/첫번째-OUT 스택으로 삽입이 멈출 것이고 자리올림 플래그 M1022는 ON으로 바뀔 것입니다.
- 이 명령은 보통 펄스 실행(SFWRP)을 합니다.
- M1022 (자리올림 플래그).

**프로그램 예제:**

- 첫번째, D0의 내용을 0으로 리셋하십시오. X0가 OFF에서 ON으로 되면, D20 내용이 생성되어 D1에 만들어 질 때 D0의 내용은 1이 됩니다. D20의 내용을 바꾸면, X0는 다시 OFF에서 ON으로 되기 위해 실행되고, 그리고 D20의 내용이 생성되어 D2에 만들어 질 때 D0의 내용은 2가 됩니다.
- SFWR 명령을 실행하기 위해 다음 ①~② 단계를 참고하십시오.
  - D20의 내용이 생성되어 D1에 만들어 집니다.
  - D0의 내용은 1이 됩니다.

**주의:**

이 API 38 SFWR 명령은 첫번째-in/첫번째-OUT 스택의 읽기/쓰기 제어를 실행하기 위해 API 39 SFRD 명령을 사용할 수 있습니다.

API	연상기호			피연산자			기능			제어기						
39		SFRD	P	<b>S</b>	<b>D</b>	<b>n</b>	쉬프트 레지스터 읽기			ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SFRD, SFRDP: 7 스텝					
S								*	*	*	*	*	*								
D								*	*	*	*	*	*	*	*						
n					*	*															

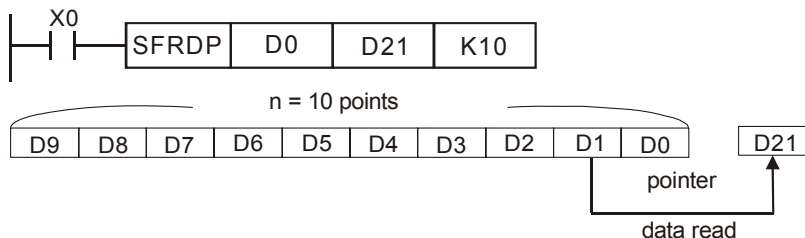
펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

설명:

- 기본 조건:  $2 \leq n \leq 512$ .
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 실행 (SFRD, SFRDP)을 지원하지 않습니다.
- S**: Head 어드레스 장치 **D**: 목적지 장치 **n**: 데이터 길이
- n**은 첫번째-in/첫번째-OUT 스택의 길이이고 소스 장치 **S**는 첫번째-in/첫번째-OUT 스택의 head 어드레스 장치입니다. 포인터처럼 첫번째 넘버 장치 **S**를 사용하고 이 명령을 실행할 때 포인터의 내용값에 1을 빼십시오. **S**로 지정된 장치의 내용은 첫번째-in/첫번째-OUT 스택의 포인터로 지정된 위치에 적어집니다. 만일 포인터 **S**의 내용이 0 (zero)과 같다면, 첫번째-in/첫번째-OUT 스택은 텅빈 것이고, 자리올림 플래그 M1022은 ON으로 바뀔 것입니다.
- 이 명령은 보통 펄스 실행(SFRDP)을 합니다.
- M1020 (Zero 플래그).

프로그램 예제:

- X0가 OFF에서 ON으로 되면, D9~D2는 모두 하나의 레지스터를 오른쪽으로 모두 쉬프트하고 D1의 내용을 읽고 D21로 쉬프트 할 때 D0의 내용은 1을 뺍니다.
- SFRD 명령을 실행하기 위해 다음 ❶~❸ 단계를 참고하십시오.
  - D1의 내용을 읽고 D21로 이동합니다.
  - D9~D2는 하나의 레지스터를 오른쪽으로 모두 쉬프트 합니다.
  - D0의 내용을 1로 빼기를 합니다.



주의:

이 API 38 SFWR 명령은 첫번째-in/첫번째-OUT 스택의 읽기/쓰기를 실행하기 위해 API 39 SFRD 명령이 사용될 수 있습니다.

API	연상기호			피연산자		기능										제어기							
	40		ZRST	P	<b>D<sub>1</sub></b>	<b>D<sub>2</sub></b>	지정된 장치의 범위 리셋										ES	EX	SS	SA	SX	SC	EH

타입 연상부호	비트 장치				워드 장치												프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ZRST, ZRSTP: 5 스텝								
	D <sub>1</sub>		*	*	*							*	*	*										
	D <sub>2</sub>		*	*	*							*	*	*										

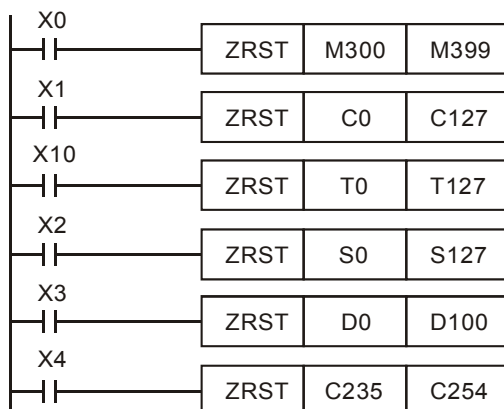
펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

**설명:**

- 기본 조건: D<sub>1</sub> 은 D<sub>2</sub>보다 작거나 같아야 합니다. (≤)
- 피연산자 D<sub>1</sub> 과 D<sub>2</sub> 는 동일한 분류로 되어야 합니다.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(ZRSTP)을 지원하지 않습니다.
- D<sub>1</sub>: 첫번째 목적지 장치 D<sub>2</sub>: 두번째 목적지 장치
- ES/EX/SS 시리즈 모델에서, 표준 속도와 고속 카운터는 혼합될 수 없습니다.
- EH/SA/SX/SC 시리즈 모델에서, 표준 속도와 고속 카운터는 혼합해서 사용할 수 있습니다.
- D<sub>1</sub> > D<sub>2</sub>이면 단지 장치 D<sub>2</sub> 만이 리셋됩니다.
- 이 명령은 보통 펄스 실행(ZRSTP)을 합니다.
- ES 시리즈 모델은 인덱스 레지스터 E 와 F 를 지원하지 않습니다

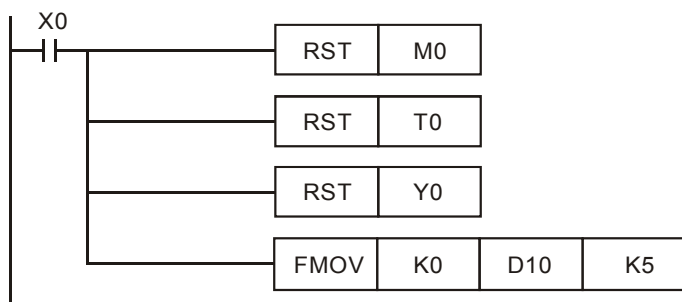
**프로그램 예제:**

- X0 가 ON 일 때, M300 부터 M399 (보조 릴레이)는 OFF 로 리셋 될 것입니다.
- X1 가 ON 일 때, C0 부터 C127 (16-비트 카운터)까지 모두 리셋 될 것입니다. (0 이 켜지고 점점과 코일은 OFF 로 리셋 될 것입니다.)
- X10 가 ON 일 때, T0 부터 T127 (타이머)까지 모두 리셋 될 것입니다. (0 은 켜지고 점점과 코일은 OFF 로 리셋 될 것입니다.)
- X2 가 ON 일 때, S0 부터 S127 까지 상태는 OFF 로 리셋 될 것입니다.
- X3 가 ON 일 때, D0 의 데이터에서 D100 (데이터 레지스터)까지 0 으로 리셋 될 것입니다.
- X4 가 ON 일 때, C235부터 C254 (32-비트 카운터)까지 모두 리셋 될 것입니다. (0이 켜지고 점점과 코일은 OFF 로 리셋 될 것입니다.)



**주의:**

1. RST 명령은 비트 장치에서 독립적으로 사용할 수 있습니다. 예를들면 Y, M, S 와 워드 장치에서 T, C, D.
2. API 16 FMOV 명령은 또한 K0 의 데이터를 워드 장치로 전송하기위해 사용할 수 있습니다. 예를들면 T, C, D 혹은 꼭 RST 명령과 같이 비트 레지스터에서, KnY, KnM, KnS.



API	연상기호			피연산자			기능			제어기							
41		DECO	P	<div>S</div> <div>D</div> <div>n</div>	8 → 256 비트 디코더				<div>ES</div> <div>EX</div> <div>SS</div> <div>SA</div> <div>SX</div> <div>SC</div> <div>EH</div>								

타입 연상부호	비트 장치				워드 장치										프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DECO, DEC 연상부호: 7 스텝						
S	*	*	*	*	*	*					*	*	*	*	*							
D		*	*	*							*	*	*	*	*							
n					*	*																

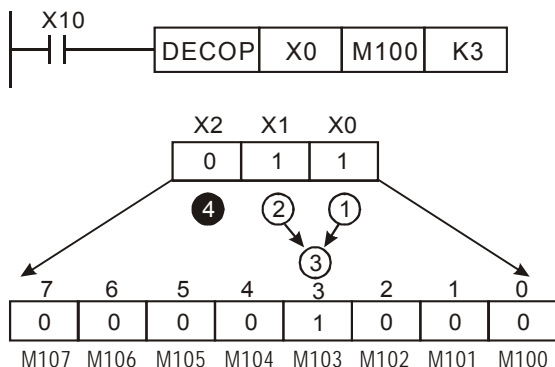
펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

## 설명:

1. 피연산자 D 가 비트 장치일 때,  $n=1\sim 8$ .
2. 피연산자 D 가 워드 장치일 때,  $n=1\sim 4$ .
3. 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
4. ES 시리즈 모델은 인텍스 레지스터 E 와 F 를 지원하지 않습니다.
5. ES/EX/SS 시리즈 모델은 펄스 실행 명령(DEC 연상부호)을 지원하지 않습니다.
6. S: 디코드 소스 장치 D: 엔코드 결과를 저장하기 위한 목적지 장치 n: 디코드 데이터 길이.
7. 소스 장치 S 의 하위 “n” 비트 데이터를 디코드하고 장치 D 에 “2<sup>n</sup>” 비트 결과를 저장하십시오.
8. 이 명령은 보통 펄스 실행(DEC 연상부호)을 합니다.

## 프로그램 예제 1:

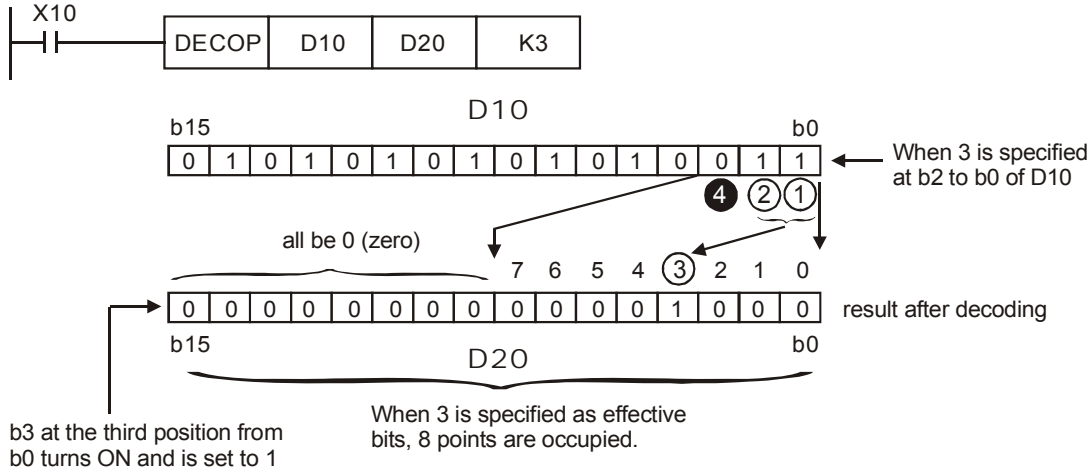
1. n 은 비트 장치  $0 < n \leq 8$  의 경우에 사용됩니다. 하지만 만일  $n=0$  거나  $n>8$  라면, 계산 에러가 발생할 것입니다.
2.  $n=8$  이면, 최대 디코드된 데이터는 2<sup>8</sup>이고 256 포인트와 같습니다. (디코딩 후 저장된 장치의 범위에 주의하십시오. 반복적으로 사용하지 마십시오.)
3. X10 이 Off → On 되면, X0~X2 의 데이터는 M100~M107 에 디코드 될 것입니다.
4. 만일 데이터 소스가 1+2=3 이라면, M100 부터 세번째 위치에 있는 M103 은 ON 으로 바뀌고 1 로 설정됩니다.
5. 실행이 완료되면, X10 는 OFF 로 바뀌게 됩니다. 디코드된 장치는 아직 구동됩니다.



## 프로그램 예제 2:

1. D 는 비트 장치,  $0 < n \leq 4$  의 경우에 사용됩니다. 하지만, 만일  $n=0$  거나  $n>4$  라면, 계산 에러가 발생할 것입니다.
2.  $n=4$  일 때, 최대 디코드된 데이터는 2<sup>4</sup>이고, 16 포인트와 같습니다.

3. X10 가 Off → On 가 되면, D10 (b2 부터 b0)의 데이터는 디코드되고 D20 (b7 부터 b0)에 저장될 것입니다. D20 (b15 부터 b8)에서 사용하지 않은 비트는 모두 0 으로 설정될 것입니다.
4. D10 에 3 개의 하위 비트를 디코드하고 D20 (한 비트는 1 이 될 것입니다) 에 8 번째 하위에 저장하십시오. 그리고, 8 개의 상위 비트의 내용은 모두 0 입니다.
5. 실행이 완료되면, X10 은 OFF 로 바뀝니다. 디코드한 장치는 아직 구동합니다.





API	연상기호			피연산자			기능			제어기						
42		ENCO	P	<b>S</b>	<b>D</b>	<b>n</b>	256 → 8 비트 엔코더			ES EX SS SA SX SC EH						

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ENCO, ENC 연상부호: 7 스텝					
S	*	*	*	*							*	*	*	*	*						
D											*	*	*	*	*						
n					*	*															

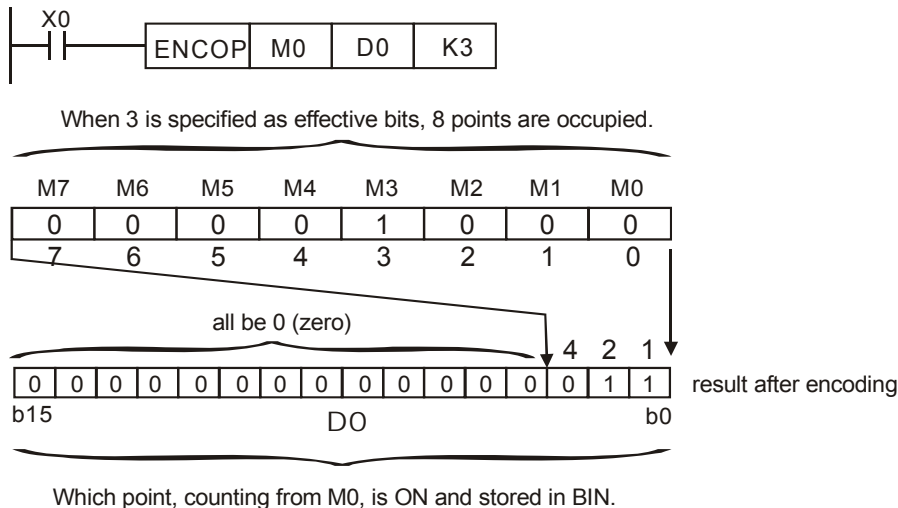
펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

## 설명:

1. 피연산자 S 가 비트 장치일 때,  $n=1\sim8$ .
2. 피연산자 S 가 워드 장치일 때,  $n=1\sim4$ .
3. 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
4. ES/EX/SS 시리즈 모델은 펄스 실행 명령(ENC 연상부호)을 지원하지 않습니다.
5. ES 시리즈 모델은 인덱스 레지스터 E 와 F 를 지원하지 않습니다.
6. S: 엔코드 소스 장치 D: 엔코드 데이터를 저장하기 위한 목적지 장치 n: 엔코드 데이터 길이
7. 소스 장치 S 에서 하위 “2<sup>n</sup>” 비트의 데이터를 엔코드하고 장치 D 에 그 결과를 저장하십시오.
8. 만일 소스 장치 S 다중 비트이고 그 값이 1 이라면, 그 처리는 마지막 비트 위치를 실행합니다.
9. 이 명령은 보통 펄스 실행(ENC 연상부호)을 합니다.
10. 만일 어떠한 비트도 소스 장치 S 에서 1 이 아니라면, M1067/M1068 =ON 이고 D1067 는 에러 코드 0E1A (16 진수)를 기록할 것입니다.

## 프로그램 예제 1:

1. S 는 비트 장치,  $0 < n \leq 8$  의 경우에 사용됩니다. 하지만 만일  $n=0$  이거나  $n>8$  라면, 계산 에러가 발생할 것입니다.
2.  $n=8$  이면, 최대 디코드된 데이터는  $2^8$ 이고 equal 256 포인트와 같습니다.
3. X0 가 Off → On 이 되면,  $2^3$  (M0 부터 M7)의 데이터는 디코드되고 D0 (b2 부터 b0)의 3 개의 하위비트에 저장될 것입니다. D0 (b15 부터 b3)에 사용되지 않는 비트는 0 으로 모두 설정될 것입니다.
4. 실행이 완료되면, X0 는 OFF 로 바뀌고 D 에 데이터는 어떠한 변화도 없을 것입니다.

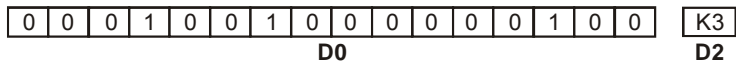




펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

1. 만일 피연산자 **S, D** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다..
2. 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
3. ES/EX/SS 시리즈 모델은 펄스 실행 명령(**SUMP**, **DSUMP**)을 지원하지 않습니다.
4. **S**: 소스 장치 **D**: 카운트된 넘버를 저장하기 위한 목적지 장치.
5. 만일 그 16 비트의 내용이 모두 “0”이라면, “Zero” 플래그, **M1020=ON**.
6. **D** 는 32-비트 명령에서 사용할 때 두개의 레지스터를 점유할 것입니다.
7. **M1020 (Zero 플래그)**.

X10 이 ON 이면, D0 안에 내용으로 “1” 을 가진 모든 비트는 카운트되고 카운트된 넘버는 D2 에 저장될 것입니다.



API	연상기호			피연산자			기능			제어기						
44	D	BON	P	<div>S</div> <div>D</div> <div>n</div>	지정된 비트 상태 체크			<div>ES</div> <div>EX</div> <div>SS</div> <div>SA</div> <div>SX</div> <div>SC</div> <div>EH</div>								

타입 연상부호	비트 장치				워드 장치										프로그램 스텝						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BON, BONP: 7 스텝  DBON, DBONP: 13 스텝					
	S				*	*	*	*	*	*	*	*	*	*	*						
	D		*	*	*																
	n					*	*					*	*	*	*		*				

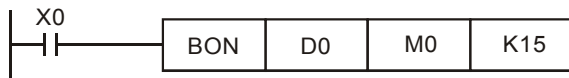
펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

## 설명:

- 만일 피연산자 **S** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
- 기본 조건: n=0~15 (16-비트), n=0~31 (32-비트).
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(BONP, DBONP)을 지원하지 않습니다.
- S**: 소스 장치 **D**: 결정된 비트를 저장하기 위한 결과 장치 **n**: 지정 결정된 비트.

## 프로그램 예제:

- X0 가 ON 일때, 만일 D0 의 15 번째 비트가 “1”이라면, M0 는 ON 이 됩니다. 하지만 만일 D0 의 15 번째 비트가 “0”이라면, M0 은 OFF 가 됩니다.
- 일단 X0 가 OFF 로 스위치되면, M0 는 이전 ON/OFF 상태로 머물 것입니다.



b15	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	b0
D0																M0=Off	

b15	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	b0
D0																M0=On	

API	연상기호			피연산자			기능			제어기							
45	D	MEAN	P	<div>S</div>	<div>D</div>	<div>n</div>	평균값			ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치												프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MEAN, MEANP: 7 스텝 DMEAN, DMEANP: 13 스텝								
	S						*	*	*	*	*	*	*											
	D							*	*	*	*	*	*	*	*									
	n					*	*	*	*	*	*	*	*	*	*									

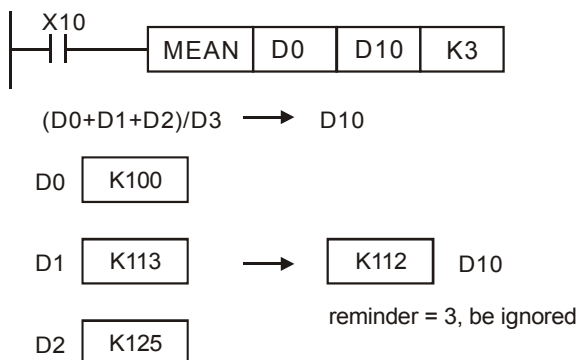
펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

**설명:**

- 만일 피연산자 D가 장치 F를 사용한다면, 16-비트 명령에서만 사용 가능합니다..
- 기본 조건: n=1~64.
- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(MEANP, DMEANP)을 지원하지 않습니다.
- ES 시리즈 모델은 인덱스 레지스터 E와 F를 지원하지 않습니다
- S**: 평균값을 구하기 위한 시작 장치 **D**: 평균값을 저장하기 위한 목적지 장치  
**n**: 평균값을 구하기 위한 장치 넘버
- S**에 의해 지정된 **n** 레지스터의 내용을 더하고, 평균값을 구하기 위해 그 합을 **n**으로 나누십시오. 이 평균값을 저장하기 위해서는 목적지 **D**에 하십시오.
- 만일 이 계산에 나머지가 있다면, 그 나머지를 무시하십시오.
- 만일 지정된 장치 넘버가 정상 사용 범위를 초과한다면, 단지 그 범위내에서만 처리될 수 있습니다.
- 만일 **n** 값이 지정된 범위 (1~64)의 밖이라면, “작동 에러”가 발생할 것입니다.

**프로그램 예제:**

X10가 ON이면, D0 (이 명령에 의해 지정된)으로부터 시작하는 3개의 레지스터의 내용을 더하고, 평균값을 구하기 위해 3으로 그 합을 나누십시오. 그리고 나서 이 평균값을 지정된 장치 D10에 저장하고 그 나머지는 무시하십시오.



API	연상기호	피연산자	기능	제어기							
46	ANS	<b>S</b> <b>m</b> <b>D</b>	경보 장치 출력	ES	EX	SS	SA	SX	SC	EH	

타입 연상부호	비트 장치				워드 장치											프로그램 스텝					
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ANS: 7 스텝					
S											*										
m					*																
D				*																	

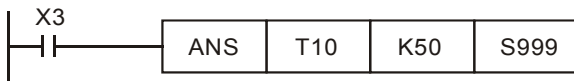
  

펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

## 설명:

1. 피연산자 **S**의 사용가능 범위: SA/SX/SC 시리즈에서: T0~T191 EH 시리즈에서: T0~T199.
2. 피연산자 **m**의 사용가능 범위: 100 ms의 단위로 K0~K32,767.
3. 피연산자 **D**의 사용가능 범위: S896~S1023 는 SA 시리즈 모델에서 사용 가능하고 S900~S1023 은 EH 시리즈 모델에서 사용 가능합니다.
4. 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
5. ES/EX/SS 시리즈 모델은 실행 명령(ANS)을 지원하지 않습니다.
6. **S**: 경보를 검출하는 타이머 **m**: 시간 셋팅 **D**: 경보 장치
7. ANS 명령은 출력 경보 장치를 구동하기 위해 사용됩니다.
8. 만일 5초 이상동안 X3 가 On일 때 경보 장치 S999=On 이면, 이 후에 X3=Off라 할지라도 S999는 On을 유지 할 것입니다. (하지만 T10 이 Off로 리셋되면, 표시 값=0).
9. M1048 (경보 포인트는 구동됨). M1049 (모티터는 사용가능함). 자세한 것은 다음을 참조하십시오.

## 프로그램 예제:



API	연상기호		피연산자	기능	제어기						
					ES	EX	SS	SA	SX	SC	EH
47		ANR	P		정보 장치 리셋						
연상부호		설명				프로그램 스텝					
N/A		접점에 의한 구동명령이 필요.				ANR, ANRP: 1 스텝					

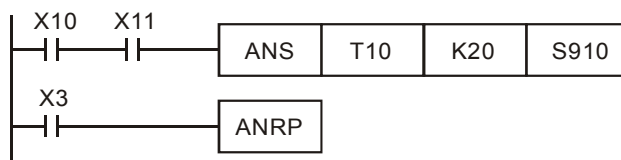
펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**설명:**

1. 피연산자 없음. ES/EX/SS 시리즈 모델은 실행 명령(ANR, ANRP)을 지원하지 않습니다.
2. ANR 명령은 정보 장치를 리셋하기 위해 사용합니다.
3. 다수의 정보 장치들이 ON 이면, 정보 장치의 하위넘버는 리셋될 것입니다.
4. 이 명령은 보통 펄스 실행(ANRP)을 합니다.

**프로그램 예제:**

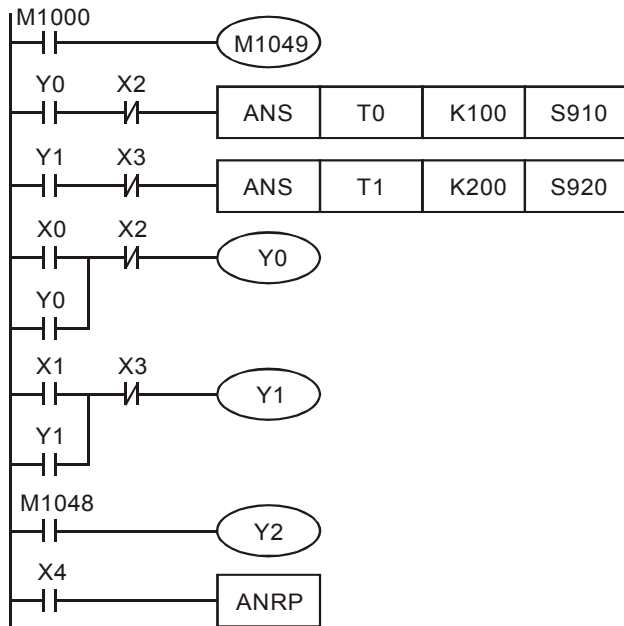
1. X10 과 X11 이 2초이상 동시에 ON 이 되면, 정보 장치 S910 은 ON 이 됩니다. 그리고 만일 X10 과 X11 이 OFF 로 바뀔지라도, 정보 장치 S910 은 아직 ON 으로 남을 것입니다. (하지만 T10 은 OFF 로 리셋될 것이고, 표시 값은 0.)
2. X10 과 X11 이 2 초 이하로 동시에 ON 이 되면, T10 의 표시값은 0 으로 리셋됩니다.
3. X3 Off → On 으로 되면,  
SA/SX/SC 시리즈에서, 구동되는 정보 장치 S896~S1023 는 리셋될 것입니다.  
EH 시리즈에서, 구동되는 정보 장치 S900~S1023 는 리셋될 것입니다.
4. X3 가 다시 Off → On 으로 되면, 그 두번째 하위 정보 장치는 리셋될 것입니다.

**주의:**

1. 플래그:
  - a) M1048 (정보 장치 구동): M1049 가 ON 이 되도록 구동할 때, 만일 S896~S1023 (SA/SX/SC 시리즈에서)/ S900~S1023 (EH 시리즈에서)에서 어느 하나의 정보장치가 출력된다면, M1048 은 ON 이 됩니다.
  - b) M1049(모니터 사용가능): M1049 를 ON 이 되게 구동할 때, D1049 는 이 명령을 실행하는 동안 가장 낮은 넘버를 자동적으로 표시할 것입니다..
2. 정보 장치의 응용:

I/O 장치 정리:

X0: 정방향 스위치, X1: 역방향 스위치, X2: 앞 위치 스위치, X3: 뒤 위치 스위치, X4: 정보 장치 리셋 버튼,  
Y0: 정방향, Y1: 정방향, Y2: 정보 표시기, S910: 정방향 정보 장치, S920: 역방향 정보.



3. M1049=On 일 때, M1048, D1049 은 사용가능합니다.
  - a) 만일 10 초이상 Y0=ON 이고 앞 위치 X2 에 도달하지 않는다면, S910=ON 입니다.
  - b) 만일 20 초이상 Y1=ON 이고 앞 위치 X3 에 도달하지 않는다면, S920=ON 입니다.
4. 역방향 스위치 X1=ON, 역방향 장치 Y1=ON 이고 그 신호는 위 위치 스위치 X3 에 도달할 때, Y1 은 OFF 로 스위치 됩니다.
  - a) 만일 구동 경고 장치가 있다면, 경고 표시기는 Y2=ON 이 됩니다.
5. 구동하는 경고 장치가 하나씩 리셋이 될 것이고, 경고 장치의 리셋 버튼 X 의 각 타임은 이 명령을 실행하는 동안 ON 이 될 것입니다.
6. 하위 구동 경고 장치는 이 명령을 모두 실행하면 리셋됩니다.



API	연상기호			피연산자		기능					제어기							
48	D	SQR	P	<b>S</b>	<b>D</b>	BIN 의 제곱근					ES EX SS SA SX SC EH							

타입 연상부호	비트 장치				워드 장치										프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SQR, SQRP: 5 스텝  DSQR, DSQRP: 9 스텝						
S					*	*							*									
D													*									

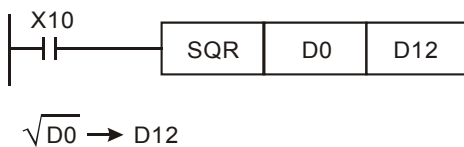
펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

**설명:**

1. 만일 피연산자 **S**, **D** 가 장치 **F** 를 사용한다면, 16-비트 명령에서만 사용 가능합니다.
2. 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
3. ES/EX/SS 시리즈 모델은 펄스 실행 명령(**SQRP**, **DSQRP**)을 지원하지 않습니다.
4. **S**: 소스 장치 **D**: 결과를 저장하는 목적지 장치
5. 이 명령은 소스 장치 **S** 에 제곱근 연산을 실행하고 그 결과를 목적지 장치 **D** 에 저장합니다.
6. **S** 는 단지 양의 값만이 될 수 있습니다. 음의 값을 제곱근 연산을 실행하는 것은 이 명령은 실행되지 않는다는 “작동 에러” 의 결과를 가져올 것입니다. 에러 플래그 M1067 과 M1068 은 On 이 될 것이고 D1067 은 에러 코드 “0E1B” (16 진수)를 기록할 것입니다.
7. **D** 의 연산결과는 단지 정수와 같이 계산되고 소수는 무시됩니다. 만일 무시한 소수가 있다면, 차용 플래그 M1021=ON.
8. **D** 연산결과가 0 일 때, Zero 플래그 M1020=On.
9. M1020 (Zero 플래그). M1021 (차용 플래그). M1067 (작동 에러).

**프로그램 예제:**

X10=On 이면, the of D0 의 내용은 제곱근 연산 후 D12 에 저장될 것입니다.



API	연상기호			피연산자		기능			제어기						
49	D	FLT	P	<b>S</b>	<b>D</b>	BIN 정수를 2 진수 부동 소수점으로 변환			ES EX SS SA SX SC EH						

타입 연상부호	비트 장치				워드 장치										프로그램 스텝							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FLT, FLTP: 5 스텝						
	S												*			DFLT, DFLTP: 9 스텝						
	D												*									

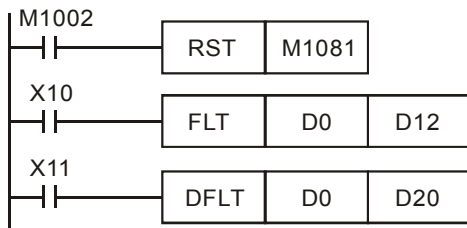
펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**설명:**

- 사용 범위를 위해서는 각 모델 규격을 참조하십시오.
- ES/EX/SS 시리즈 모델은 펄스 실행 명령(FLTP, DFLTP)을 지원하지 않습니다.
- S**: 소스 장치 **D**: 변환 결과를 저장하는 목적지 장치
- M1081 가 OFF 일 때, 소스 데이터는 BIN 정수에서 2 진수 부동 소수점으로 변환됩니다. 이 때, 16-비트 명령 FLT 의 **S** 소스 장치는 1 개의 레지스터를 점유하고 **D** 목적지 장치는 2 개의 레지스터를 점유합니다.
  - 만일 변환 결과의 절대값이 최대 부동값보다 더 크다면, 자리올림 플래그 M1022=On.
  - 만일 변환 결과의 절대값이 최소 부동값보다 더 작다면, 자리올림 플래그 M1021=On.
  - 만일 변환 결과 is 0, zero 플래그 M1020=On.
- M1081 이 ON 일 때, 소스 데이터는 2 진수 부동 소수점에서 BIN 정수로 변환됩니다. (소수 무시) 이때, 16-비트 명령 FLT 의 **S** 소스 장치는 2 개의 레지스터를 점유하고 **D** 목적지 장치는 1 개의 레지스터를 점유합니다. 이 실행은 명령 INT 와 동일합니다.
  - 만일 변환 결과가 **D** (16-비트는 -32,768~32,767 이고 32-비트는 -2,147,483,648~2,147,483,647) 의 BIN 정수 범위를 초과하면, 그것은 최대값이나 최소값을 나타낼 것입니다. 그리고 자리올림 플래그는 M1022=On 으로 설정할 것입니다.
  - 만일 변환 결과의 소수가 무시된다면, 차용 플래그 M1021=On.
  - 만일 변환 결과가 0 이라면, zero 플래그 M1020=On.
  - 변환후, **D** 는 16 비트로 저장됩니다.
- M1081 (FLT 명령기능 교환).

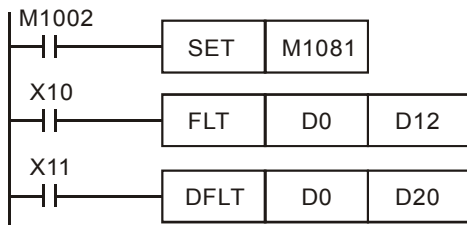
**프로그램 예제 1:**

- M1081 이 OFF 이면, 소스 데이터는 BIN 정수로부터 2 진수 부동 소수점으로 변환됩니다.
- X10 이 ON 이면, D0 (BIN 정수)는 D13, D12 (2 진수 부동 소수점)으로 변환됩니다.
- X11 이 ON 이면, D1, D0 (BIN 정수)는 D21, D20 (2 진수 부동 소수점)으로 변환됩니다.
- 만일 D0=K10, X10 가 On 이 되면, 변환 후 부동 소수점 32-비트는 H41200000 이 될것이고 32-비트 레지스터 D12(D13)에 저장될 것입니다.
- 만일 32-비트 레지스터 D0(D1)=K100,000, X11 는 On 이 되면, 변환 후 부동 소수점 32-비트는 H4735000 이 될것이고 32-비트 레지스터 D20(D21)에 저장될 것입니다.



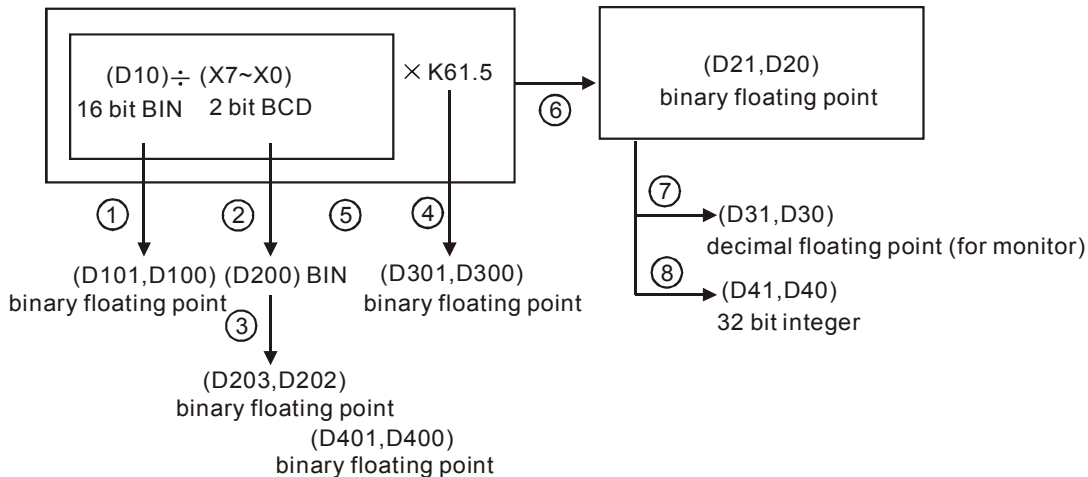
## 프로그램 예제 2:

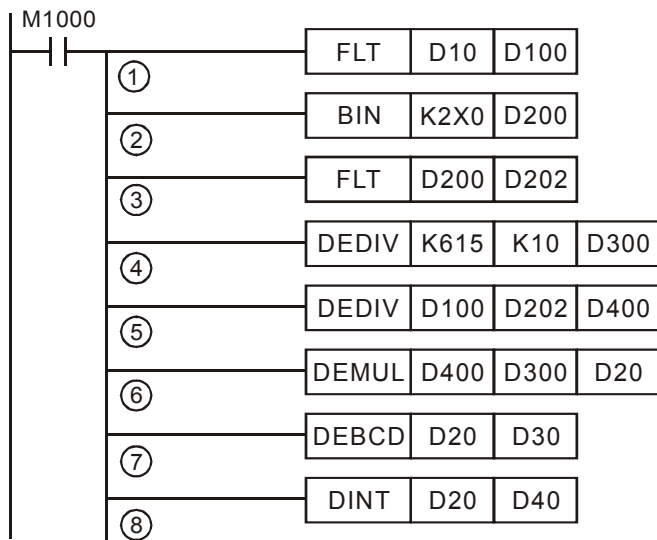
1. M1081 이 On 이면, 소스 데이터는 2 진수 부동 소수점에서 BIN 정수로 변환됩니다. (소수 무시)
2. X10 이 ON 이면, D0 과 D1(2 진수 부동 소수점)은 D12 (BIN 정수)로 변환됩니다. 만일 D0(D1)=H47C35000, 부동 소수점은 100,000 이라면, 실행 결과는 D12=K32,767 가 될 것이고, 16-비트 레지스터 D12 의 최대값을 초과하는 값 때문에, M1022=On 이 될 것입니다.
3. X11 이 ON 이면, D1, D0 (2 진수 부동 소수점)는 D21, D20 (BIN 정수)로 변환됩니다. 만일 D0(D1)=H47C35000, 부동 소수점이 100,000 라면, 그 결과는 32-비트 레지스터 D20(D21)에 저장될 것입니다.



## 프로그램 예제 3:

다음 연산을 완성하기 위해 이 응용명령을 사용하십시오.





- D10 (BIN 정수)를 D101, D100 (2 진수 부동 소수점)으로 변환.
- X7~X0 (BCD 값)의 값을 D200 (BIN 값) 으로 변환.
- D200 (BIN 정수)을 D203, D202 (2 진수 부동 소수점) 으로 변환.
- $K615 \div K10$ 의 결과를 D301, D300 (2 진수 부동 소수점)에 저장.

2 진수 부동 소수점의 나눗셈:

- $(D101, D100) \div (D203, D202)$ 의 결과를 D401, D400 (2 진수 부동 소수점)에 저장.

2 진수 부동 소수점의 곱셈:

- $(D401, D400) \times (D301, D300)$ 의 결과를 D21, D20 (2 진수 부동 소수점)에 저장.
- 2 진수 부동 소수점 (D21, D20)을 소수부동 소수점(D31, D30)으로 변환.
- 2 진수 부동 소수점 (D21, D20)을 BIN 정수(D41, D40)으로 변환.

API	연상 기호			피연산자		기능										제어기																						
50		REF	P	<b>D</b>	<b>n</b>	입출력 즉시 재생										ES	EX	SS	SA	SX	SC	EH																
유형 OP	비트 장치				워드 장치											프로그램 단계																						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	REF, REFP: 5 단계																						
	D	*	*																																			
	n					*	*																															
																		펄스				16-비트				32-비트												
																		ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**D**: 입출력 재생 소스 시작    **n**: 재생을 위한 입출력 번호

설명:

- 모든 PLC 입력과 출력의 상태는 END 까지 주사한 후 재생될 것입니다. 입력의 상태는 입력 메모리에 저장되기 위해 외부입력으로부터 읽혀집니다. 출력 터미널은 END 명령 이후 출력 메모리를 출력장치로 보냅니다. 그러므로, 이 명령은 최신 데이터를 입/출력 할 때 알고리즘 과정으로 쓰여질 수 있습니다.
- 모든 입력과 출력의 상태는 주사 된 직후 바뀔수도 있습니다. 사용자가 다음 주사때까지 기다리고 싶지 않은 경우, REF 명령이 사용될 수 있습니다.
- D** 는 i.e. 00, 10, 20, 30... 등과 같이 항상 10 의 배수여야 하며 X0, X10, Y0, Y10 등과 같이 되어야 합니다. **n** 은 i.e. 8, 16, 24, 32 등과 같이 항상 8 의 배수여야 하며 이의 가능한 범위는 8~256 입니다. **n** 의 값이 주어진 범위(8~256)를 벗어나거나 8 의 배수가 아닌 경우, “작동 오류”가 발생할 것입니다. 사용 범위는 다양한 모형에 따라 달라질 수 있으니 자세한 내용은 각주를 참조하시기 바랍니다.

프로그램 예 1:

X0=ON 일 경우, PLC 는 X0~X17 입력점의 상태를 즉시 읽고 재생시킬 것입니다. 여기서 입력 지연은 발생하지 않습니다.



프로그램 예 2:

X0 = ON 일 경우, 출력신호 Y0~Y7(8 점)은 즉시 출력 터미널로 보내지고 재생될 것입니다. END 명령이 있을 때까지 출력하지 않아도 됩니다.



주의:

ES/EX/SS 와 SA/SX/SC 의 직렬 모형에서, 이 명령에 의해 처리되는 입력 점과 출력 점은 MPU: X0~X17, Y0~Y17 and n=K8 or K16 의 입/출력입니다.

API	연상 기호			피연산자	기능										제어기						
51		REFF	P	<div>n</div>	재생과 필터 조정										ES	EX	SS	SA	SX	SC	EH

OP	유형	비트 장치				워드 장치										프로그램 단계										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	REFF, REFFP: 3 단계										
	n					*	*																			

펄스								16-비트								32-비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

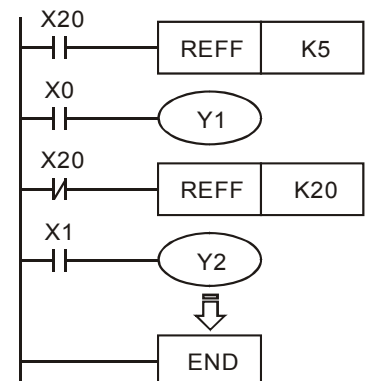
n: ms 단위로 응답시간 설정

설명:

- SA 직렬 모델에서 피연산자의 사용가능한 범위 n: 0~20ms 이며 EH 직렬 모델에서는 0~60ms 입니다.
- PLC 는 소음이나 충격을 방지하기 위해 입력 필터와 함께 제공이 됩니다. EH MPU 의 입력 X0~X17 와 SA/SX/SC MPU 의 입력 X0~X7 은 디지털 필터 기능과 함께 합니다. 명령 n 은 D1020 와 D1021 에 직접 지정될 것이며 X0~X7 과 X10~X17 의 반응 시간에 각각 조정될 것입니다.
- X0~X17 의 입력 필터가 DVP 직렬 PLC 입력일 때 응답시간의 작동 규칙은:
  - PLC 의 전원이 Off 에서 On 으로 바뀌거나 END 명령을 실행할 경우, 응답 시간은 D1020 과 D1021 의 만족 값에 따라 결정이 됩니다.
  - 프로그램 중간에 설정 값은 MOV 명령의 사용으로 D1020 과 D1021 로 옮겨질 수 있습니다.
  - 응답시간은 프로그램 실행에 있어 REFF 명령의 사용에 의해 변할 수 있습니다. 이 때, REFF 명령에 의해 지정된 응답시간은 D1020 과 D1021 로 옮겨질 것이며 다음 주사 시에 다시 조정될 것입니다.

프로그램 예:

- PLC 의 전원이 Off 에서 On 으로 바졌을 때, X0~X17 입력의 응답시간은 D1020 과 D1021 의 만족 값에 따라 결정이 됩니다.
- X20=On 일 때, REFF K5 명령이 실행되며, 응답시간은 5ms 로 바뀌고 다음 주사 시에 다시 조정될 것입니다.
- X20=Off 일 때, REFF K20 명령이 실행되며, 응답시간은 20mc 로 바뀌고 다음 주사 시에 다시 조정될 것입니다.



주의:

가로채기 매개변수, 초고속 카운터, 또는 SPD 명령(API56)을 사용할 경우, 부합하는 입력 터미널의 반응 시간은 지연되지 않을 것이며 그로 인한 작용은 이 명령과는 무관합니다.

API	연상 기호	피연산자				기능				제어기												
52	MTR	<b>(S)</b>	<b>(D<sub>1</sub>)</b>	<b>(D<sub>2</sub>)</b>	<b>(n)</b>	입력 행렬				ES	EX	SS	SA	SX	SC	EH						
유형 OP	비트 장치				워드 장치											프로그램 단계						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MTR: 9 단계						
	S	*																				
	D <sub>1</sub>		*																			
	D <sub>2</sub>		*	*	*																	
n					*	*																

**피연산자:**

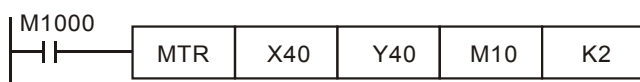
**S:** 입력 행렬의 헤드 번지    **D<sub>1</sub>:** 출력 행렬의 헤드 번지    **D<sub>2</sub>:** 행렬 주사의 반응 헤드 번지    **n:** 행렬의 열 수

**설명:**

- S** 는 모든 입력 행렬을 구체화해주는 헤드 번지이다. 입력이 구체화 되면 8 연속 입력 장치의 선택은 “입력 행렬”이라 불립니다.
- D<sub>1</sub>** 는 행렬의 트랜지스터 출력 헤드 번지입니다.
- 이 명령은 8 의 연속된 입력 장치(헤드 번지 **S**)가 여러 번(**n**)사용되도록 허락해줍니다. 각각의 입력은 한 개 이상을 가지며, 다른 **D<sub>1</sub>** 신호가 처리됩니다. 입력 신호 8 의 각각은 “열” 안에서 그룹화되며 **n** 개의 열이 있습니다. 각 열은 **D<sub>1</sub>**에서 출력의 양에 의해 선택되며, 행렬을 이루기 위해 사용된 것은 **n** 열의 수와 일치합니다. 결과는 반응 헤드 번지 **D<sub>2</sub>**에서 시작하는 행렬 테이블에 저장됩니다.
- 최대 입력은 64 입력을(8 입력의 8 줄) 달성할 수 있습니다.
- 이 명령이 가로채기 포맷에 사용이 될 때 매 25msec 마다 각 열이 처리됩니다. 이것은 8 줄 행렬에 결과가 나올 것이며 i.e.64 입력(8 입력의 8 줄)은 매 200msec 마다 읽혀집니다. 그러므로 이 명령은 켜기/끄기 속도가 200ms 가 넘는 입력 신호를 위해서는 사용되지 않습니다.
- 보조 릴레이 M1000 를 사용하는 것이 추천되며 보통 열린 접촉을 합니다.
- MTR** 명령 수행이 끝나면 완료된 표시 문자 **M1029** 의 명령실행이 **ON** 으로 켜지면서 이 명령실행은 자동적으로 **MTR** 명령이 **OFF** 가 될 때 재시동됩니다.
- 이 명령은 한번만 사용 가능합니다.

**프로그램 예:**

- X0=On** 일 때, **MTR** 명령은 실행하기 시작합니다. 외부 2 줄, 총 16 장치들이 순서대로 읽히며 결과는 내부 릴레이 M10~M17, M20~M27 에 저장됩니다.



- 아래의 그림은 **MTR** 명령의 작동의 와이어링 도형의 예입니다. 외부 2 줄은 **X40~47** 와 **Y40~Y41** 로 구성되어 있으며 총 16 장치들이 **MTR** 명령과 사용된 내부 릴레이 **M10~M17** 와 **M20~M27** 에 상응합니다. 성공적인 작동을 위한 일반적 주의사항을 위해 다이오드는 각각의 입력 장치들 이후에 배치되어야 합니다. 이 다이오드들은 0.1A,50V 의 정격을 가져야합니다.





API	연상 기호			피연산자			기능			제어기													
53	D	HSCS		<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	<b>(D)</b>	고속 카운터 설정			<table><tr><td>ES</td><td>EX</td><td>SS</td><td>SA</td><td>SX</td><td>SC</td><td>EH</td></tr></table>							ES	EX	SS	SA	SX	SC	EH
ES	EX	SS	SA	SX	SC	EH																	

유형 OP	비트 장치				워드 장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DHSCS: 13 단계
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*		
S <sub>2</sub>												*				
D		*	*	*												

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S<sub>1</sub>**: 값 비교

**S<sub>2</sub>**: 고속 카운터의 수

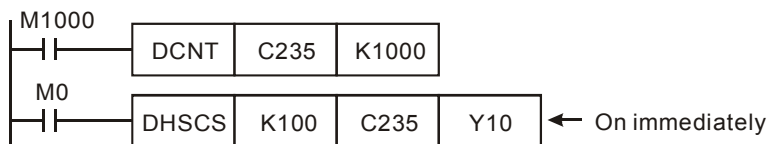
**D**: 결과 비교

설명:

1. 피연산자 **S<sub>2</sub>**의 이용가능한 범위: C235~C254.
2. 피연산자 **D**는 I0□0,□=1~6 범위로 지정가능합니다.
3. 32-비트 **DHSCS** 명령만이 작동가능합니다.
4. ES 와 SA 직렬 모형에서, 피연산자 **D**는 레지스터 E,F를 지원하지 않습니다.
5. 모든 고속 카운터는 가로채기 처리를 사용하므로 모든 비교 결과 장치 **D**는 즉시 업데이트 됩니다.
6. HSCS 명령은 선택된 비교 값 **S<sub>1</sub>**에 대해 선택된 고속 카운터 **S<sub>2</sub>**의 현재 값을 비교합니다. 카운터의 현재 값이 **S<sub>1</sub>**와 같은 값으로 변할 경우 **D**로 지정된 장치는 ON으로 설정됩니다. 비교 결과가 같지 않다 하더라도 장치 **D**의 상태는 여전히 ON일 것입니다.
7. 장치 **D**로 지정된 장치가 Y0~Y17 이라면, 고속 카운터의 비교값과 현재 값이 같을 때 비교 결과는 즉시 외부 입력 Y0~Y17에 출력될 것이며 다른 Y장치는 스캔 주기에 영향을 받을 것입니다.

프로그램 예 1:

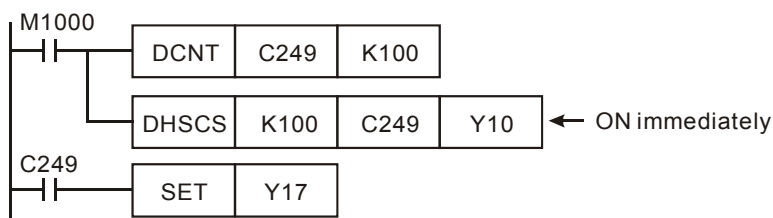
PLC가 RUN 명령을 수행하고 난 후 M0=On 이면, DHSCS 명령은 작동하기 시작합니다. C235의 현재 값이 99→100이 되거나 101→100이 되고 항상 ON일 경우, 즉시 Y10=ON이 될 것입니다.



프로그램 예 2:

DHSCS 명령의 Y 출력과 일반 Y 출력의 다른 점:

- a) C249의 값이 99→100과 101→100이 될 경우, DHSCS의 Y10 출력은 가로채기 처리를 사용함으로써 외부 출력에 출력이 되며 이는 프로그램 주사 시간과 관련이 있습니다. 그러나, 모듈 릴레이(10ms)나 트랜지스터(10us)의 출력에 의한 지연은 여전히 있을 것입니다.
- b) 고속 타이머 C249의 현재 값이 99에서 100으로 바뀔 때, C249는 활성화 될 것이며 프로그램 주사 시간에 따른 END 명령 이후 Y17은 ON이 될 것입니다.

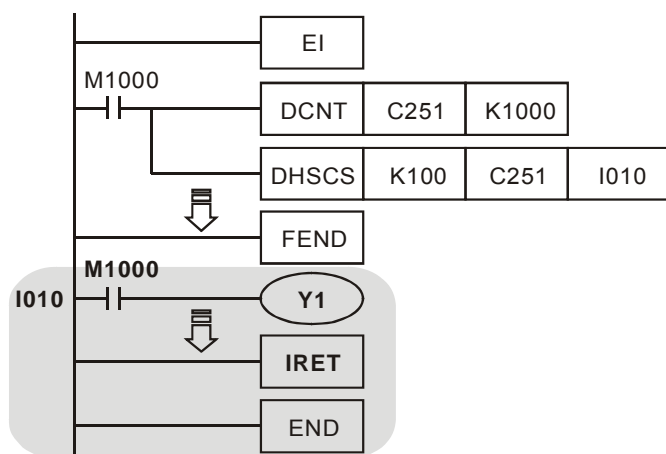


### 프로그램 예 3:

고속 카운터 가로채기:

- ES/EX/SS 직렬 모형은 고속 카운터 가로채기 기능을 지원하지 않습니다.
- SA/SX/SC 직렬 모형이 고속 카운터 가로채기를 사용할 때 한계점  
가로채기를 지정하기 위한 **DHSCS** 명령을 사용할 때, 지정된 고속 카운터는 다른 DHSCS, DHSCR, DHSZ 명령에서는 사용될수 없습니다. 만약 사용한다면, 오류가 발생할 것입니다.
- 가로채기 점 I010 에서 I060 은 **DHSCS** 명령의 피연산자 **D** 로서 사용가능하며, 이는 지정된 고속 카운터가 **DHSCS** 명령에서 값에 다다를 때 실행되는 루틴을 가로챌 수 있게 해줍니다.
- SA** 직렬 모형에서, 가로채기 기능을 제어하기 위한 고속 카운터를 위한 지정된 가로채기 숫자와 함께 6 개의 입력이 있습니다. 고속 카운터 X0 의 입력은 가로채기 숫자 I010(1 단계 2 입력 그리고 AB 단계 카운터 C246~C254 만이 I010 를 지시가능)를 지정하며, X1 은 I020 을, X2 는 I030 을, X3 은 I040 을, X4 는 I050 을, 그리고 X5 는 I060 을 지정합니다.

**C251** 의 현재값이 **99→100** 과 **101→100** 로 변할 때, 프로그램은 가로채기 루틴을 실행하기 위해 가로채기 지시기 I010 로 건너뛰게 됩니다.



- SA/SX/SC** 직렬 모형에서 **M1059** 는 고속 카운터 가로채기는 표시문자(I010~I060)을 저지합니다.
- EH** 직렬 모형에서 **M1289~M1294** 는 숨겨진 표시문자 **I010** 에서 **I060** 를 저지하는 고속 카운터 가로채기입니다. 예를 들어, **M1294** 이 ON 이 되었을 때, 가로채기 지시기 I060 이 숨겨집니다.

가로채기 지시기 I 번호	가로채기 저지 표시 문자
I010	M1289
I020	M1290
I030	M1291

가로채기 지시기 I 번호	가로채기 저지 표시 문자
I040	M1292
I050	M1293
I060	M1294

주의:

- 고속 카운터의 출력 접촉과 **DHSCS** (API 53)명령, **DHSCR** (API 54)명령, 그리고 **DHSZ**(API 55)명령의 비교 출력이 모두 입력에 카운터가 되면, 이들 모두 활성화됩니다. 고속 카운터의 현재 값을 설정 값과 같도록 바꾸기 위해 **DADD**, **DMOV** 명령 등과 같은 데이터 작동 명령을 사용하면 카운터 입력이 없기 때문에 비교가 설정되거나 출력이 설정될 것입니다.
- 고속 카운터 provided in ES/EX/SS 직렬 모형에 제공된 고속 카운터: 총 카운터 주파수는 20 KHz 입니다.

유형	1 단계 1 입력							1-단계 2 입력			2 단계 입력		
Input	C235	C236	C237	C238	C241	C242	C244	C246	C247	C249	C251	C252	C254
X0	U/D				U/D		U/D	U	U	U	A	A	A
X1		U/D			R		R	D	D	D	B	B	B
X2			U/D			U/D			R	R		R	R
X3				U/D		R	S			S			S

U: 입력 증가

A: A 단계 입력

S: 입력 시작

D: 입력 감소

B: B 단계 입력

R: 입력 재시동

- 입력 점 X0 와 X1 은 고속 카운터가 될 수 있으며 1 단계가 20KHz 가능합니다. 하지만 이 입력 점의 총 카운터 빈도는 총 빈도 20KHz 보다 적거나 같아야 합니다. 카운터 입력이 A/B 단계 신호일 때, 빈도수는 카운터 빈도의 4 배가 될 것입니다. 그러므로 A/B 단계의 카운터 빈도는 거의 5KHz 가 됩니다.
- ES/EX/SS 직렬 유형 Is 에서, **DHSCS** 와 **DHSCR** 명령은 4 번이상 사용할 수 없습니다.
- SA/SX/SC 직렬 유형 Is 에서 제공되는 고속카운터: 1-단계 고속 카운터: 총 카운터 빈도는 20 KHz 입니다.
- SC 직렬 MPU 에서는 3 개의 새로운 고속 카운터 **C243**, **C245**, **C250** 가 있습니다. 입력점 X10(C243)과 X11(C245)은 개별적으로 100KHz 이상의 최고속 입력 빈도고속 카운터와 함께 1 단계 1 입력으로 사용가능 합니다. 다른 출력점 C250 는 최고 입력 빈도 100KHz 를 가집니다.
- SC 직렬 유형 Is 의 고속카운터에서, 고속 카운터의 입력 X10~X11 의 총 대역너비는 130KHz 이상이 됩니다.

유형  Input	1 단계 1 입력											1-단계 2 입력				2 단계 입력		
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C249	C250	C251	C252	C254
X0	U/D						U/D			U/D		U	U	U		A	A	A
X1		U/D					R			R		D	D	D		B	B	B
X2			U/D					U/D					R	R			R	R
X3				U/D				R		S				S				S
X4					U/D													
X5						U/D												
X10									U/D						U			
X11											U/D				D			

U: 입력 증가

A: A 단계 입력

S: 입력 시작

D: 입력 감소

B: B 단계 입력

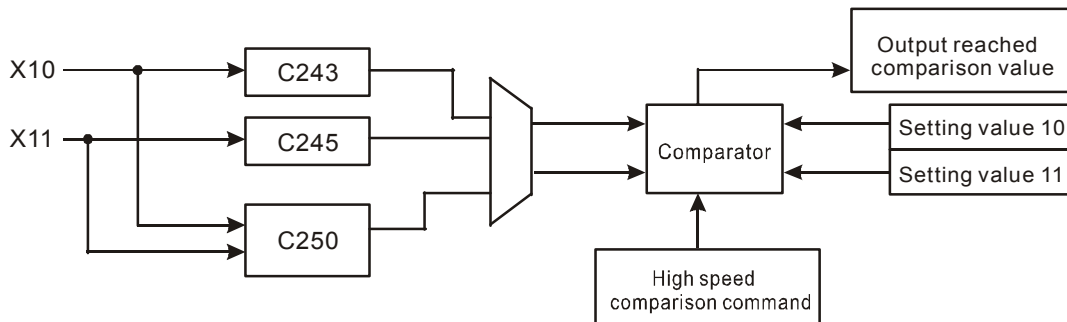
R: 입력 재시동

- 입력점 X0 과 X1 은 고속 카운터에 따라 계획될 수 있으며 1-단계는 20KHz 이상 될수 있습니다. 하지만 이러한 입력 점의 총 카운터 빈도는 총 빈도 20KHz 과 같거나 적어야 합니다. 카운터 입력이 A/B 단계 신호라면, 빈도는 카운터 빈도의 네 배가 될 것입니다. 그러므로, A/B 단계의 카운터 빈도는 거의 5KHz 가 됩니다.
- 입력 X5 는 두가지 기능을 가집니다.
  - M1260=Off 일 때, C240 는 일반 U/D 고속 카운터입니다.
  - M1260=On 일 때, X5 는 C235~C239 의 전역 재시동입니다.

3. SA/SX/SC 직렬방식 Is 에서, **DHSCS**, **DHSCR** 그리고 **DHSZ** 명령은 6 번 이상 사용할 수 없습니다.
4. **DHSCS** 명령이 **IXXX** 를 **C250** 와 같은 지정된 고속카운터에서 가로채기 수로 사용을 한다면 **DHSCS**, **DHSCR**, 그리고 **DHSZ** 명령은 **C250** 에서 작동하지 않을 것입니다.

SC 직렬 MPU 만을 위한 X10~X11 고속 카운터 기능:

1. X10 과 X11 가 1-단계 1 입력이나 1-단계 2 입력으로 사용될 경우 대역너비는 100KHz 이상입니다.
2. X10 과 X11 을 상승 에지 또는 하락 에지 유형으로 설정하는 것은 선택식입니다. X11 의 유형은 D1167 에 의해 결정되는 반면 X10 의 유형은 D1166 에 의해 결정됩니다. K0:상승 에지 유형; K1:하락 에지 유형, K2: 상승 에지와 하락 에지(X10 만 지원).
3. C245 를 총계하는 것은 M1245 의 OFF 와 ON 에 의해 결정되는 반면 C243 는 M1243 의 OFF 와 ON 에 의해 결정됩니다.
4. C243 나 C245 를 사용할 때, C250 는 사용될 수 없습니다. C250 를 사용할 때, C243 와 C245 모두 사용될 수 없습니다.
5. 고속 카운터 (C243, C245, C250)의 구조와 비교 명령.



6. 고속 카운터 C243, C245, C250 와 비교측정기 설명:

- a) **DHSCS** 와 **DHSCR** 명령이 고속카운터 **C243**, **C245**, 그리고 **C250** 에 사용될 때, 명령 사용 횟수는 총 2 회 이상이 되도록 계획합니다. 예를 들어, **DHSCS D0 C243 Y10** 와 같은 유사 명령이 사용된다면, 다른 유사 명령 사용은 **DHSCR D2 C243 Y10** 또는 **DHSCS D4 C245 Y0** 이 될 것입니다.
- b) **DHSZ** 명령이 고속 카운터 **C243**, **C245**, 그리고 **C250** 에 사용될 때, 한번만 사용되도록 계획합니다. 고속 카운터 **SA/SX** 직렬의 본래 고속카운터의 명령 사용 횟수는 여전히 6 번이며 증가된 고속 카운터 **C243**, **C245**, 그리고 **C250** 에 의해 줄어들지는 않습니다.
- c) 고속 반응 출력이 고속 유사 명령 **DHSCS** 의 장치를 위해 필요하다면, **Y10** 또는 **Y11** 을 출력 장치와 함께 쓰는 것을 추천합니다. **Y10** 와 **Y11** 이 사용되지 않으면, 출력은 출력을 ON 또는 OFF 로 설정하기 위해 한 번의 스캔 시간을 지연할 것입니다. 예를 들어, **I0x0** 를 가로채기 수로 사용하면 **C243**, **C245**, 그리고 **C250** 에 대응하는 가로채기 수는: **C243** 는 **I020**, **C245** 는 **I040**, 그리고 **C250** 는 **I060** 에 대응합니다.
- d) **DHSCR** 는 고속카운터를 위한 확실한 출력장치 명령입니다. 이것은 카운터의 값을 0 으로 재시동시킬 수 있지만 비교 카운터는 eg. **DHSCR K10 C243 C243** 와 같아야 합니다. 고속 카운터를 사용할 때, **DHSCR** 는 **C243**, **C245**, 그리고 **C250** 만을 위한 것입니다.

EH 직렬 방식 Is 에 제공되는 고속카운터

1. 프로그램 가로채기 방식 1-단계 고속 카운터, **C235~C240**: 일반 카운터 빈도는 10KHz 이상이며, 최대 총

카운터 빈도는 20 KHz 입니다.

2. C241~C254 는 하드웨어 고속카운터 (HHSC)입니다. DVP-EH 를 위한 4 개의 HHSC 가 있습니다. HHSC0/HHSC1 의 펄스 입력 빈도수는 200KHz 이상이며 HHSC2/HHSC3 는 30KHz 이상입니다.

HHSC0 의 이용가능한 장치 수: C241, C246, C251

HHSC1 의 이용가능한 장치 수: C242, C247, C252

HHSC2 의 이용가능한 장치 수: C243, C248, C253

HHSC3 의 이용가능한 장치 수: C244, C249, C254

- a) 각각의 HHSC 는 한 개의 장치 숫자당 한 번만 세부화 가능합니다. HHSC 를 세부화하기 위해서 DCNT 명령을 사용하십시오.
- b) 각 HHSC 의 이용 가능한 카운터 유형:
- 1 단계 1 입력, 펄스/방향 유형이라고도 불림
  - 1-단계 2 입력, CW/CCW 유형이라고도 불림
  - 2-단계 2 입력, AB 단계 유형이라고도 불림
3. 이용 가능한 고속카운터를 위한 다음 테이블을 참조하십시오.

카운터 유형	Program interrupt 유형 1- phase 고속 카운터						Hardware 고속 카운터											
유형	1 단계 1 입력						1 단계 1 입력		1-단계 2 입력				2-단계 입력					
Input	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C246	C247	C248	C249	C251	C252	C253	C254
X0	U/D						U/D				U				A			
X1		U/D									D				B			
X2			U/D				R				R				R			
X3				U/D			S				S				S			
X4					U/D		U/D				U				A			
X5						U/D					D				B			
X6							R				R				R			
X7							S				S				S			
X10									U/D				U			A		
X11													D			B		
X12									R				R			R		
X13									S				S			S		
X14										U/D				U				A
X15														D				B
X16										R				R				R
X17										S				S				S

U: 입력 증가

A: A 단계 입력

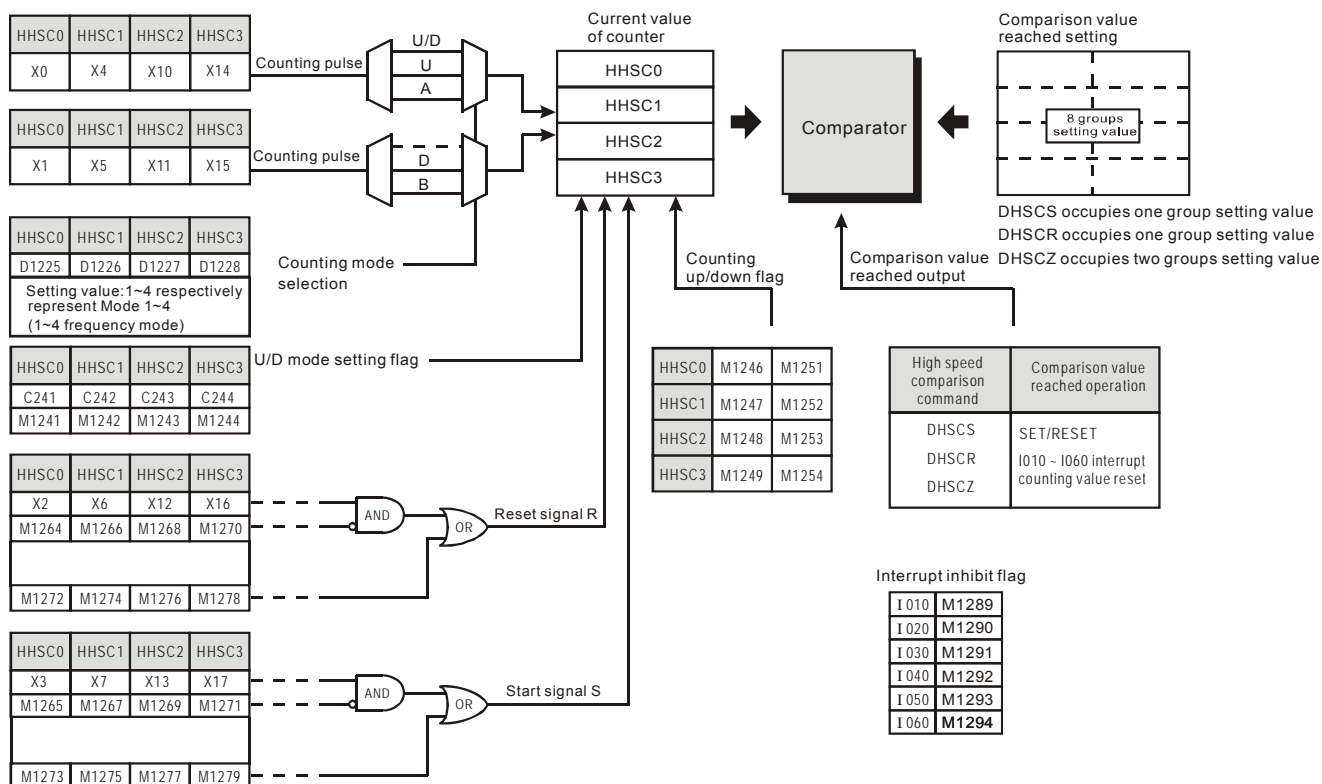
S: 입력 시작

D: 입력 감소

B: B 단계 입력

R: 입력 재시동

1. DVP EH 직렬 방식 Is 의 프로그램에서 DHSCS, DHSCR, DHSZ 와 같은 하드웨어 고속카운터 관련 명령을 사용하는 데는 한계가 없습니다. 하지만, 명령을 동시에 실행하는 데는 사용시간에 한계가 있습니다. DHSCS, DHSCR 명령은 한 그룹 설정을 사용할 것이며 DHSZ 명령은 두개의 그룹 설정을 사용할 것입니다. 이 명령들이 동시에 실행되면 총 사용된 그룹 설정은 8 개 그룹으로 초과될 것입니다. 8 그룹 설정이 초과되면 시스템은 처음 주사되고 실행된 명령의 사용된 기억 단위를 총계하고 나머지는 무시될 것입니다.
2. 하드웨어 고속카운터의 시스템 구조:



3. HHSC0~3 모두는 외부 입력의 신호를 재시동 하고 시작합니다. 재시동 신호(R)는 M1272/M1274/M1276/M1278 (HHSC0 ~3 에 따라)에 의해 설정가능하며 시작 신호는 M1273/M1275/M1277/M1279 (HHSC0 ~3 에 따라)에 의해 설정가능합니다. 고속카운터를 사용할 때 R 과 S 의 외부 신호 입력을 사용하지 않는다면 당신은 M1264/M1266/M1268/M1270 와 M1265/M1267/M1269 /M1271 를 TRUE 로 설정할 수 있습니다. 입력 신호의 작동을 닫으면 유사 외부 입력이 일반 입력으로 사용가능합니다. 사용을 위한 위의 예를 참조하십시오.

#### 4. 카운터 유형 선택

ES/EX/SS/SA/SX/SC 직렬의 고속카운터는 2-단계 2 입력 카운터유형이며 특별 장치 D1022 이 네배 주파수 형식과 함께 설정됩니다. 레지스터 D1022 의 내용값이 PLC 제어기가 Stop 에서 Run 상태로 바뀔 때 처음 주사 시간에 올려집니다. (V5.5 와 위의 DVP- ES/EX/SS 직렬 MPU 만이 이 기능을 지원)

장치 번호	기능 설명
D1022	두 배 주파수 설정을 위한 카운터의 카운터 방법 사용
D1022=K1	선택(일반 주파수) 유형
D1022=K2	선택(두 배 주파수) 유형
D1022=K4	선택(네 배 주파수) 유형







두 배 주파수유형:

카운터 방식		신호 도형
2 단계 2 입력	1 (일반 빈도수)	
	2 (2배 빈도수)	
	4 (4배 빈도수)	

카운터 유형의 다른 유형에 따라 HHSC 0~3 of EH 직렬 유형 Is 는 네 가지 빈도 유형을 장치 D1225 에서 D1228 까지 사용해 보통, 두배, 세배, 그리고 네배까지 설정이 가능합니다:

카운터 방식		신호 도형	
유형	특별 D 의 설정 값	+1 세기	-1 세기
1-단계 1 입력	0 (일반 빈도수)		
	1 (두배 빈도수)		
1-단계 2 입력	0 (세배 빈도수)		
	1 (네배 빈도수)		

카운터 방식		신호 도형	
유형	특별 D 의 설정 값	+1 세기	-1 세기
2-단계 2 입력	0 (일반 빈도수)		

	1 (2 배 빈도수)	A  B 
	2 (3 배 빈도수)	A  B 
	3 (4 배 빈도수)	A  B 

U/D FLAG 는 특별 M 장치이고, M1241~M1245 와 각각은 C241~C245 카운터 위 아래의 표시문자 설정을 지정합니다.

관련 표시문자와 고속 카운터의 특별 레지스터

표시문자	기능 설명
M1150	DHSZ 명령이 비교 방식 값 설정하는 복수 그룹으로 사용됨을 알립니다.
M1151	값 비교방식 실행이 완성되면 DHSZ 명령 복수 그룹 설정합니다.
M1152	DHSZ 명령이 빈도수 제어 방식으로 사용됨을 알립니다.
M1153	빈도수 제어 방식 실행이 완료됩니다.
M1235 ~ M1245	고속카운터 C235 ~ C245 의 카운터 방향 지정 M12□□=Off 일 때, C2□□ 올라가고 M12□□=On 일 때, C2□□ 내려감
M1246 ~ M1249 M1251 ~ M1254	고속카운터 C246~C249, C251~C254 의 카운터 방향을 모니터 합니다. C2□□ 올라가면, M12□□=Off C2□□ 내려가면, M12□□=On
M1260	X5 는 모든 고속카운터의 재시동 입력입니다.
M1261	DHSCR 명령의 고속 비교 표시문자입니다.
M1264	HHSC0 신호 재시동 (R) 외부 제어 신호 종료 입력 접촉 불능

표시문자	기능 설명
M1265	HHSC0 신호 시작 (S) 외부 제어 신호 종료 입력 접촉 불능
M1266	HHSC1 신호 재시동 (R) 외부 제어 신호 종료 입력 접촉 불능
M1267	HHSC1 신호 시작 (S) 외부 제어 신호 종료 입력 접촉 불능
M1268	HHSC2 신호 재시동 (R) 외부 제어 신호 종료 입력 접촉 불능
M1269	HHSC2 신호 시작 (S) 외부 제어 신호 종료 입력 접촉 불능
M1270	HHSC3 신호 재시동 (R) 외부 제어 신호 종료 입력 접촉 불능
M1271	HHSC3 신호 시작 (S) 외부 제어 신호 종료 입력 접촉 불능
M1272	HHSC0 신호 재시동 (R) 내부 제어 신호 입력 접촉 종료
M1273	HHSC0 신호 시작 (R) 내부 제어 신호 입력 접촉 종료
M1274	HHSC1 신호 재시동 (R) 내부 제어 신호 입력 접촉 종료
M1275	HHSC1 신호 시작 (R) 내부 제어 신호 입력 접촉 종료
M1276	HHSC2 신호 재시동 (R) 내부 제어 신호 입력 접촉 종료
M1277	HHSC2 신호 시작 (R) 내부 제어 신호 입력 접촉 종료
M1278	HHSC3 신호 재시동 (R) 내부 제어 신호 입력 접촉 종료
M1279	HHSC3 신호 시작 (R) 내부 제어 신호 입력 접촉 종료
M1289	고속 카운터 가로채기, I010 숨김
M1290	고속 카운터 가로채기, I020 숨김
M1291	고속 카운터 가로채기, I030 숨김
M1292	고속 카운터 가로채기, I040 숨김



표시문자	기능 설명
M1293	고속 카운터 가로채기, I050 숨김
M1294	고속 카운터 가로채기, I060 숨김
M1312	C235 입력 시작 제어
M1313	C236 입력 시작 제어
M1314	C237 입력 시작 제어
M1315	C238 입력 시작 제어
M1316	C239 입력 시작 제어
M1317	C240 입력 시작 제어
M1320	C235 입력 재시동 제어
M1321	C236 입력 재시동 제어
M1322	C237 입력 재시동 제어
M1323	C238 입력 재시동 제어
M1324	C239 입력 재시동 제어
M1325	C240 입력 재시동 제어
M1328	C235 가능한 제어 시작/재시동
M1329	C236 가능한 제어 시작/재시동
M1330	C237 가능한 제어 시작/재시동
M1331	C238 가능한 제어 시작/재시동
M1332	C239 가능한 제어 시작/재시동
M1333	C240 가능한 제어 시작/재시동

Special register	기능 설명
D1022	카운터 in ES/EX/SS/SA/SX/SC 직렬 방식 Is 에서 AB 단계 카운터의 두 배 빈도수 선택
D1150	다수 그룹 설정 비교 방식의 카운터 레지스터 테이블의 DHSZ 명령
D1151	빈도 제어 방식의 카운터 레지스터 테이블을 위한 DHSZ 명령
D1152 (lower-bit)	DHSZ 명령은 D1153 과 D1152 각각의 그룹의 펄스 출력 빈도수로부터 순차적으로 읽혀진 테이블 카운터 레지스터 값을 저장
D1153 (upper-bit)	
D1166	X10 의(SC 방식 Is 만)상승 에지 또는 하락 에지의 선택 방식
D1167	X11 의(SC 방식 Is 만)상승 에지 또는 하락 에지의 선택 방식
D1225	방법 설정 세는 첫 번째 카운터 C241, C246, C251 카운터 유형
D1226	방법 설정 세는 두 번째 카운터 C242, C247, C252 카운터 유형
D1227	방법 설정 세는 세 번째 카운터 C243, C248, C253 카운터 유형
D1228	방법 설정 세는 네 번째 카운터 C244, C249, C254 카운터 유형
D1225 ~ D1228	하드웨어 고속 카운터의 카운터 유형 EH 직렬 유형 I 의 HHSC0~HHSC3 설정 값이 0 일 때, 일반 빈도 카운터 유형임. 설정 값이 1 일 때, 두 배 빈도 카운터 유형임. 설정 값이 2 일 때, 세 배 빈도 카운터 유형임. 설정 값이 3 일 때, 네 배 빈도 카운터 유형임.

API	연상 기호		피연산자			기능				제어기							
54	D	HSCR	<b>(S<sub>1</sub>) (S<sub>2</sub>) (D)</b>			고속 카운터 재시동				<div>ES</div> <div>EX</div> <div>SS</div> <div>SA</div> <div>SX</div> <div>SC</div> <div>EH</div>							

유형 OP	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DHSCR: 13 단계	
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*			
S <sub>2</sub>												*					
D		*	*	*													

펄스								16-비트								32-비트								
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		

피연산자:

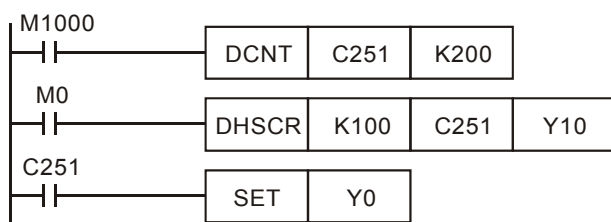
**S<sub>1</sub>**: 값 비교      **S<sub>2</sub>**: 카운터 수      **D**: 결과 비교

설명:

1. 피연산자 **S<sub>2</sub>** 는 고속카운터 C235~254 의 수를 나타내야 합니다. 더 많은 정보를 원하시면 각주 API 53 DHSCS 을 참조하십시오.
2. 피연산자 **D** 는 고속카운터(C241~C254 만)의 수 또한 지정할 수 있습니다.
3. **ES** 와 **SA/SX** 직렬 방식 **Is** 에서, 피연산자 **D** 는 **C** 장치를 지원하지 않습니다. **SC** 방식 **Is** 에서, 피연산자 **D** 는 **C243**, **C245**, 그리고 **C250** 를 지원합니다.
4. 각각의 고속카운터는 가로채기 명령에 의한 펄스 입력과 함께 각각 외부 입력 X0~X17 에 대응하는 값을 가집니다. 총계를 세기 위한 DHSCR 명령 과정에 의해 지정된 카운터가 DHSCR 명령이 지정된 카운터와 설정된 값의 현재 값 사이에 비교작업을 즉시 시작할 것입니다. 비교작업 중 지정된 고속카운터의 현재 카운터 값이 **S<sub>1</sub>** 에 의해 지정된 설정 값과 같아지면 **D** 에 의해 지정된 피연산자(또는 장치)가 OFF 가 될것입니다. 고속카운터가 **D** 가 OFF 가 된 후에도 계속 센다면, **D** 는 여전히 OFF 로 남아있을 것입니다.
5. 장치 D 로 지정된 장치가 Y0~Y17 이라면, 비교값과 고속카운터의 현재값이 같을 때 비교결과는 외부 입력 Y0~Y17(Y 로 지정된 출력이 재시동될것임)에 즉시 출력될 것이며, 다른 Y 장치들은 주사 사이클에 의해 영향을 받을 것입니다. 하지만 **M**, **S** 자치들은 즉시의 출력이며 주사 사이클에 의해 영향을 받지 않습니다.

프로그램 예 1:

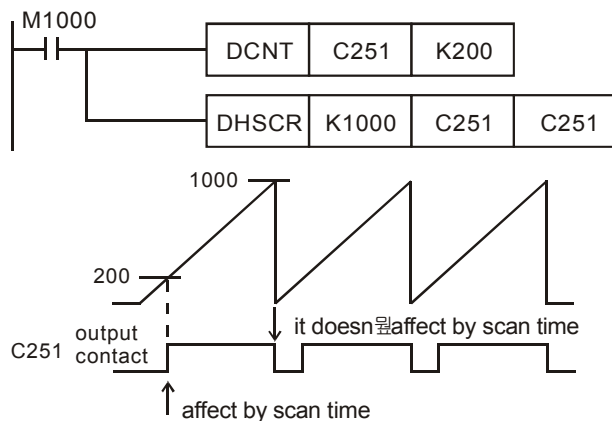
1. M0=On 이고 C251 의 현재 값이 99→100 또는 101→100 로 건너뛰면, Y10 는 Off 가 될것입니다.
2. C251 의 현재 값이 199 에서 200 건너뛰면, 접촉 C251 는 On 이 되고 Y0=On 가 되도록 만들것입니다. 하지만 프로그램 주사 시간 지연 출력은 여전히 있을 것입니다.
3. Y10 는 카운터를 지정할 때 즉시 재시동하는 장치 상태입니다.
4. 고속카운터의 같은 수를 지정하는데 사용가능합니다. 프로그램 예 2 번을 참조하십시오.



프로그램 예 2:

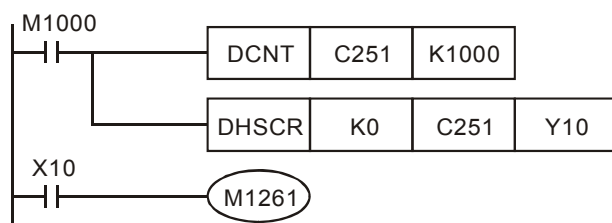
고속카운터의 수를 지정할 때 고속카운터 C251 의 현재값은 999→1000 또는 1001→1000 로 바뀔것이며

C251 접촉은 Off 로 재시동될 것입니다.



#### 주의:

1. 고속카운터와 직렬 방식 Is 에서 제공되는 그 사용 범위를 위해서는 API 53 DHSCS 명령의 각주를 참조하십시오.
2. EH 직렬에서는, M1261 는 고속카운터의 외부 재시동 방식을 세부화하기 위해 사용됩니다. 몇몇 고속카운터들은 외부 재시동을 위해 입력점을 제공합니다. 입력 점이 On 이 되면 고속카운터의 최근 대응값은 0 으로 재시동 되며 출력점은 Off 가 됩니다. 외부 출력이 실행되기 위해 고속카운터의 외부 재시동 방식이 세부화될 필요가 있을 때에는 M1261=On 으로 설정하십시오.
3. M1261 기능의 한계: 하드웨어 고속카운터 C241~C254 에서만 사용가능합니다.
4. 다음은 사용예입니다:
  - a) C251 의 외부 재시동의 입력점은 X2 입니다.
  - b) Y10=On 이라면.
  - c) M1261=Off 일 때, X2=On 이고, C251 의 최근 값은 0 으로 재시동되며 접촉은 Off 가 됩니다. DHSCR 명령이 실행되면 카운터 입력은 없어지고 비교 결과는 출력되지 않습니다. 그러므로 Y10=On 는 바뀌지 않고 있을 것입니다.
  - d) M1261=On 일 때, X2=On 이고, C251 의 최근 값은 0 으로 재시동되며 접촉은 Off 가 됩니다. DHSCR 명령이 실행되면 카운터 입력이 없다 할지라도 비교 결과는 출력될 것입니다. 그러므로 Y10 의 내용은 재시동될 것입니다.



API	연상 기호		피연산자				기능				제어기						
55	D	HSZ	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>S</b>	<b>D</b>	HSC 구역 비교				ES	EX	SS	SA	SX	SC	EH

유형 OP	비트 장치				워드 장치											프로그램 단계		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DHSCS: 17 단계		
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*			
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*			
S												*						
D		*	*	*														

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**피연산자:**

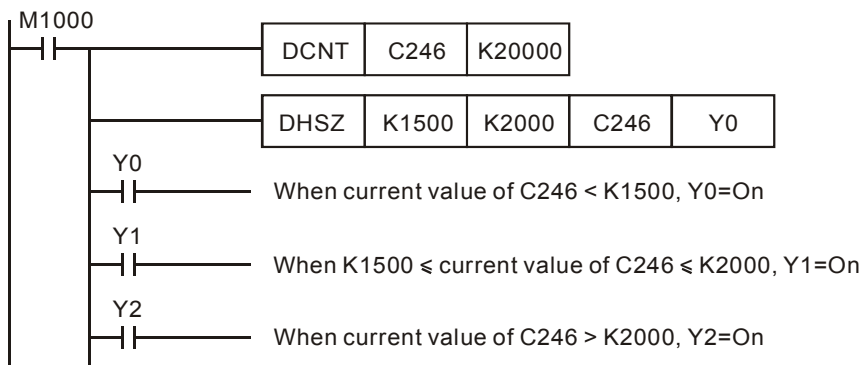
**S<sub>1</sub>**: 구역 비교의 최하 한계      **S<sub>2</sub>**: 구역 비교의 최상 한계      **S**: 카운터의 수      **D**: 비교 결과(3 개의 연속 비트 장치 차지)

**설명:**

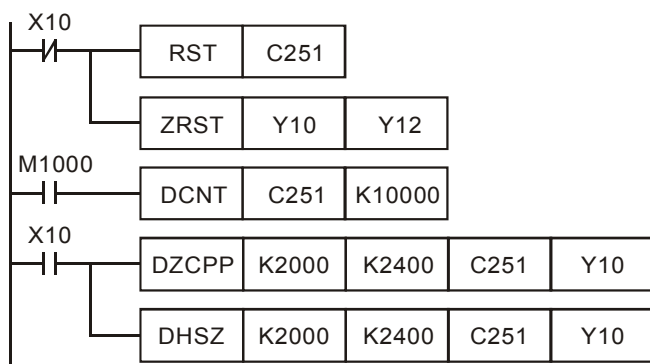
1. **S<sub>1</sub>**는 **S<sub>2</sub>** 보다 작거나 같아야 합니다.( $S_1 \leq S_2$ ).
2. 피연산자 **S** 는 고속카운터 **C235~C254** 를 가리켜야합니다.
3. 출력 작동은 주사 시간에 의해 영향을 받지 않을 것입니다.
4. 모든 출력과 구역 비교는 가로채기 작동을 사용할 것입니다.

**프로그램 예 1:**

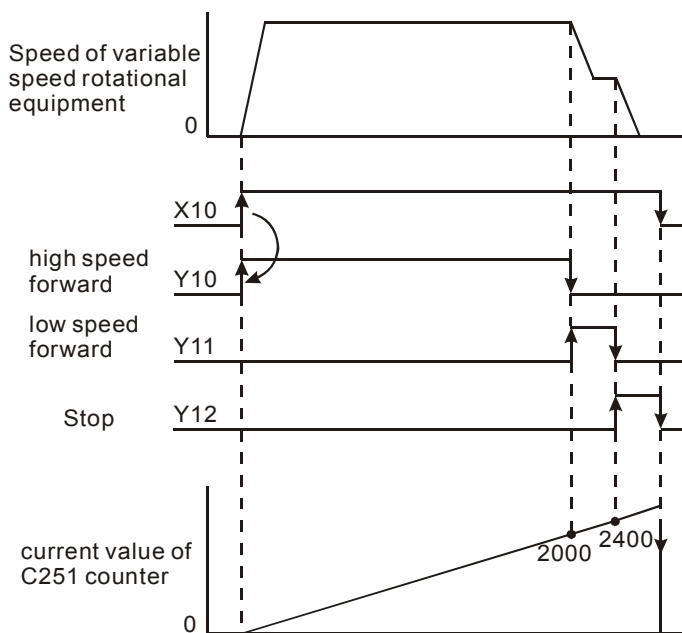
1. 구체화된 장치는 Y0 이므로 Y0~Y2 는 자동적으로 차지될 것입니다.
2. **DHSZ** 명령이 실행되고 고속 카운터 **C246** 이 세기 시작할 때 최상 최하 한계 값에 도달하면 Y0~Y2 중 한 개가 On 이 될것입니다.

**프로그램 예 2:**

1. 고/저속을 제어하고 중지하기 위한 **DHSZ** 을 사용할 때 **C251** 는 **AB** 단계 고속카운터입니다. **C251** 에 카운터 펄스가 저장될 때만 **DHSZ** 명령의 비교값 출력이 있을 것입니다. 그러므로 카운터의 현재값이 0 이라 하더라도 Y10 은 On 이 되지 않을 것입니다.
2. **X10=On** 일 때, **DHSZ** 명령은 카운터 최근값이  $\leq K2,000$  일 때 **Y10=On** 가 되게 합니다. 이 문제를 개선하기 위해 프로그램이 처음에 RUN 할 때 **C251** 와 **K2,000** 에 **DZCPP** 명령을 이용하십시오. 카운터 최근 값  $\leq K2,000$  일 때, **Y10=On** 이며 **DZCPP** 명령은 펄스 실행 명령입니다. **DZCPP** 는 프로그램에서 한번만 사용가능하며 **Y10** 는 여전히 On 일 것입니다.
3. 드라이브 접촉이 **X10=Off** 일 때, **Y10~Y12** 는 Off 로 재시동 될것입니다.



타이밍 모형

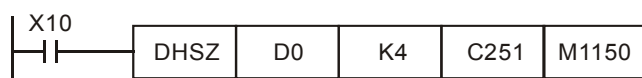


## 프로그램 예 3:

1. 유형 of DHSZ 명령의 값 비교 방식을 설정하는 다수의 그룹을 사용할 때 DHSZ 명령의 D 가 특별 보조릴레이 M1150 라면, 이는 고속카운터의 최근값을 실행할 것이며 다수 그룹 설정 값을 비교하고 출력하는 기능을 가집니다.
2. 이런 방식에서 S<sub>1</sub>은 비교 테이블의 시작 장치로 정의가 됩니다. 이는 데이터 레지스터 D 만 가능하며 색인 레지스터 E,F 에 의해 변경가능합니다. 하지만 색인 레지스터 E,F 에 의해 변경된 수는 명령이 실행된 후 변경 불가능합니다. S<sub>2</sub>는 비교 데이터의 데이터그룹으로 정의됩니다. K1~K128 나 H1~H80 만 가능하며 이 역시 색인 레지스터 E,F 에 의해 변경가능합니다. 명령이 실행된 후 이 값을 변경시키는 것은 불가능합니다. S 는 고속카운터의 수로 정의되며 C235~C254 여야 합니다. D 는 방식 설정으로 정의됩니다. 이는 M1150 만 가능하며 색인 레지스터 E,F 에 의해 변경가능합니다. 하지만 8 M1150 가 아니라면 D 는 불가능합니다.
3. S<sub>1</sub>에 의해 구체화된 레지스터의 주 숫자와 S<sub>2</sub>에 의해 구체화된 열 수(그룹 수)로 구성된 고속카운터의 비교테이블입니다.
4. 명령이 실행되기 전에 각 레지스터의 설정값을 입력해주시오.
5. S 에 의해 구체화된 고속카운터 C251 의 최근 값이 (D1,D0)의 설정 값과 같을 때 D2 에 의해 구체화된 출력은 Off (D3=K0) 또는 On (D3=K1)로 재시동되어 걸릴 것입니다. 모든 출력 Y 는 가로채기 작동을

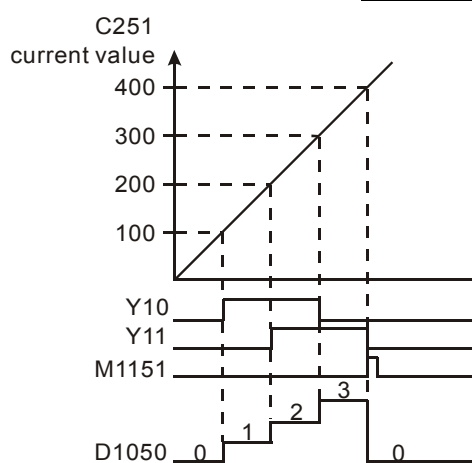
사용할 것입니다.

6. C251의 최근 값이 비교테이블에서 첫번째 그룹의 설정값과 같으면 D1150=K1가 됩니다. C251의 최근 값이 비교테이블에서 두번째 그룹의 설정값과 같으면 D1150=K2가 됩니다. 그러면 비교는 위에 설명된 순서대로 계속 실행될 것입니다. 모든 그룹의 비교가 끝나면 한번의 주사 사이클을 위해 M1151=On이 되고 D1150는 0으로 재시동될 것이며 실행을 위해 첫번째 그룹으로 건너뛸 것입니다.
7. 드라이브 접촉 X10이 Off가 되면 명령의 작동은 가로채기될 것이며 테이블 카운터 레지스터 D1150의 내용은 0으로 재시동될 것입니다. 하지만 ON/OFF 상태는 그 당시에는 변하지 않을 것입니다.
8. 이 명령이 실행되고 END 명령에 처음 주사 하면 도형안의 모든 설정값이 유효합니다.
9. 이 명령의 이 기능은 프로그램에서 한번만 사용가능합니다. SA/SX/SC 직렬 방식 Is에서, 이 기능은 제공되지 않습니다.
10. 이 명령의 이 기능은 하드웨어 고속카운터 C241~C254에서만 사용됩니다.



Comparison table

32-비트 comparison data		Number of output Y	On/Off indication		Table counting register D1150
높은 수	낮은 수				
D1 (K0)	D0 (K100)	D2 (K10)	D3 (K1)		0
D5 (K0)	D4 (K200)	D6 (K11)	D7 (K1)		1
D9 (K0)	D8 (K300)	D10 (K10)	D11 (K0)		2
D13 (K0)	D12 (K400)	D14 (K11)	D15 (K0)		3
		K10:Y10 K11:Y11	K0:Off K1:On	0→1→2→3→0 Cyclic scan	



고속카운터의 관련 표시문자와 특별 레지스터

표시문자	기능 설명
M1150	DHSZ 명령이 다수그룹 설정 값 비교 방식으로 사용된다는 것을 알림
M1151	DHSZ 명령에서 값 비교 설정하는 다수 그룹 방식 실행 완료

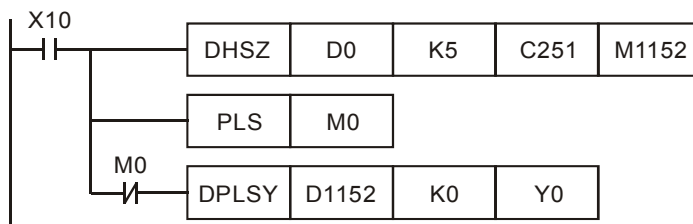
Special register	기능 설명
D1150	DHSZ 명령에서 DHSZ Y의 테이블 색인 출력

#### 프로그램 예 4:

1. 주파수 제어 작동(DHSZ와 DPLSY 명령이 섞임): DHSZ 명령의 D는 보조 중계 M1152이며 고속카운터의 현재 값을 실행할수 있고 DPLSY 명령의 펄스 출력 주파수를 제어할수 있는 기능을 가지고

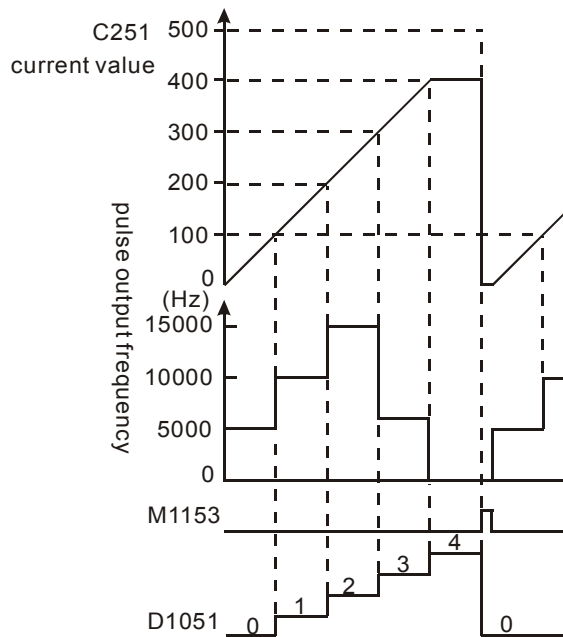
있습니다.

- 이런 방식에서, **S<sub>1</sub>** 는 비교 테이블의 시작 장치로 정의됩니다. 이는 데이터 레지스터 D 만이 될수 있고 색인 레지스터 E,F 에 의해 변경가능합니다. 하지만 색인 레지스터 E,F 에 의해 변경된 수는 명령이 실행된 후 변하지 않습니다. **S<sub>2</sub>** 는 비교 데이터의 데이터 그룹으로 정의됩니다. 이는 **K1~K128** 또는 **H1~H80** 만 되며 색인 레지스터 E,F 에 의해 변경가능합니다. 명령이 실행된 후, 이 값을 변경시키는 것은 불가능합니다. **S** 는 고속카운터의 수로 정의되며 **C235~C254** 여야 합니다. **D** 는 방식 설정으로서 정의되며 **M1152** 만 가능하며 색인 레지스터 E,F 에 의해 변경가능합니다. 하지만 **M1152** 가 아닐 경우 **D** 는 불가능합니다.
- 이 명령의 기능은 프로그램에서 한번만 사용가능합니다. **SA/SX/SC** 직렬 방식 **Is** 에서 이 기능은 제공하지 않습니다. **EH** 직렬방식 **Is** 에서는 하드웨어 고속카운터 **C241~C254** 만 사용가능합니다. 명령이 실행되기 전에 각 레지스터의 설정 값을 입력하십시오.
- S** 에 의해 세부화된 **C251** 의 현재 값이 (D1,D0)의 최상과 최하한계값 사이 범위내에 있다면 (D3,D2)의 설정값은 **DPLSY** 명령의 펄스 출력 주파수로 변환될 것입니다. 그러므로 비교 테이블에서 두번째 그룹은 계속 실행할 것입니다. 모든 그룹의 비교 작동이 완료된후 한번의 주사 주기를 위해 **M1153=On** 가 되고 **D1151** 은 0 으로 재시동되며 실행을 위해 첫번째 그룹으로 건너뛸 것입니다.
- 마지막 그룹에서 실행을 멈추고 싶다면 **K0** 으로 마지막 그룹의 내용을 설정하십시오.
- 드라이브 접촉 **X10** 이 **Off** 가 되면 명령의 작동은 가로채기될것이며 테이블 카운터 레지스터 **D1151** 의 내용은 0 으로 재시동될것입니다.



비교 테이블

32-비트 비교 데이터		펄스 출력 주파수 0~250KHz	테이블 카운터 레지스터 D1151
높은 수	낮은 수		
D1 (K0)	D0 (K0)	D3, D2 (K5,000)	0
D5 (K0)	D4 (K100)	D7, D6 (K10,000)	1
D9 (K0)	D8 (K200)	D11, D10 (K15,000)	2
D13 (K0)	D12 (K300)	D15, D14 (K6,000)	3
D17 (K0)	D16 (K400)	D19, D18 (K0)	4
			0→1→2→3→4 사이클 주사



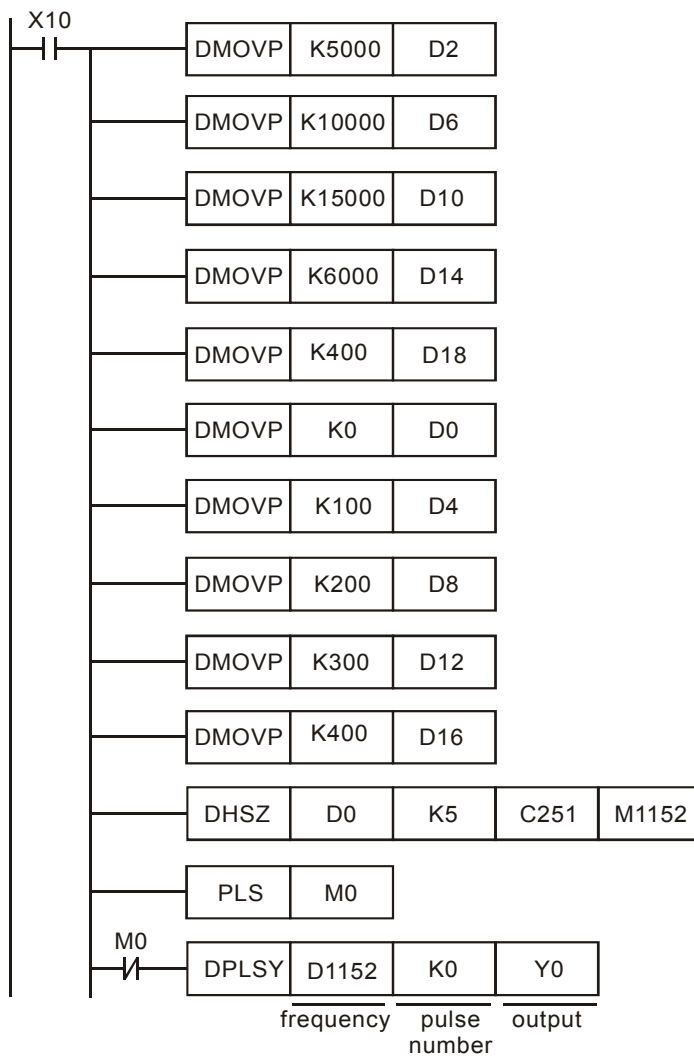
관련 표시문자와 고속카운터의 특별 레지스터:

표시문자	기능 설명
M1152	DHSZ 명령이 빈도수 제어 방법으로 사용되었다고 알림
M1153	DHSZ 명령에서 빈도수 제어 방법이 실행완료

특별 레지스터	기능 설명
D1151	DHSZ D 값에 의해 변하는 테이블 색인
D1152 (낮은 수)	DHSZ 명령이 D1153 과 D1152 각 그룹의 펄스 출력 빈도수로부터 차례로 읽어 들인 테이블 카운터 레지스터 값을 저장
D1153 (높은 수)	
D1336 (낮은 수) D1337(높은 수)	DPLSY 명령 출력의 펄스 수

아래가 완료된 프로그램입니다:





1. DHSZ 명령 중에는 비교 테이블에서 설정 값을 변경하지 마십시오.
2. 프로그램이 END 명령으로 실행되면 지정된 데이터가 위의 예 프로그램으로 보여질 것입니다. 그러므로 DPLSY 명령은 DHSZ 명령 실행 후 실행될 것입니다.

API	연상 기호	피연산자	기능	제어기						
56	SPD	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b>	Speed Detection	ES	EX	SS	SA	SX	SC	EH

OP	유형	비트 장치				워드 장치										프로그램 단계
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	
	S <sub>1</sub>	*														
	S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*
	D											*	*	*		

피연산자:

**S<sub>1</sub>**: 외부 펄스 입력      **S<sub>2</sub>**: 펄스 시간(ms)      **D**: 결과 (5 연속 장치 차지)

설명:

1. **S<sub>1</sub>**: 외부 펄스의 입력 지정

각 직렬 방식 Is 의 펄스 입력		
유형 Is	ES/EX/SS 직렬 방식 Is (V5.7 와 그 위) 과 SA/SX/SC 직렬 방식 Is	EH 직렬 방식 Is
이용 가능한 입력	X1, X2	X0~X3

2. **S<sub>2</sub>** (단위는 ms)에 의해 지정된 시간동안 **S<sub>1</sub>** 에 의해 지정된 입력에서 수신된 펄스의 수를 세고 **D** 로 지정된 레지스터에 결과를 저장합니다.
3. **D** 는 5 레지스터, **D+1**,를 차지하고, **D** 는 이전 펄스 **D+3** 의 검사값을 가리키며, **D+2** 는 펄스의 현재 축적값을 가리키며 **D+4** 는 최대 32767ms 까지 가능한 잔여 카운터 시간을 가리킵니다.
4. 측정된 펄스 빈도수:

각 직렬 방식 Is 의 펄스 속도		
유형 Is	ES/EX/SS 직렬 방식 Is (V5.7 과 그 위)와 SA/SX/SC 직렬 방식 Is	EH 직렬 방식 Is
최고측정 빈도수	X1(30KHz), X2(10KHz) 총 빈도수는 30KHz 이하	X0/X1 (200KHz) X2/X3 (10KHz)

5. EH 직렬 방식 Is 에서 이 명령을 사용할 때, 외부 입력 X0~X3 의 펄스 빈도수와 하드웨어 고속카운터의 빈도수는 같으며 둘다 200KHz 에 도달할 수 있습니다.
6. 이 명령은 주로 회전 속도의 비례값을 갖기 위해 사용됩니다. 결과 **D** 와 회전속도는 비례하게 됩니다. 다음 식은 모터의 회전 속도를 얻기 위해 사용될수 있습니다.

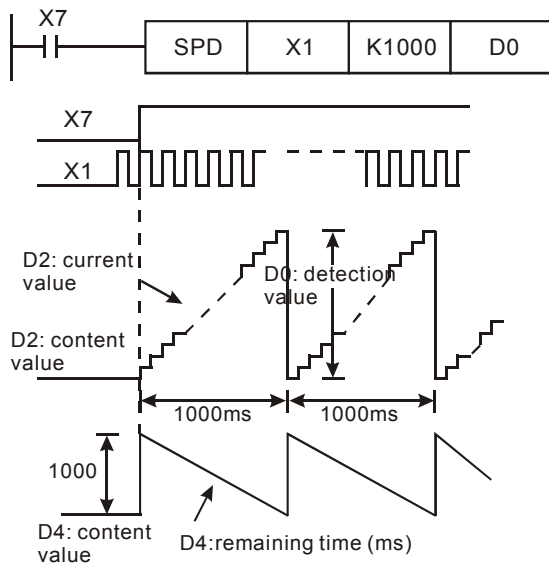
$$N = \frac{60(D0)}{nt} \times 10^3 (rpm)$$

N: 회전속도  
 n: 회전 기구의 회전 당 펄스의 수  
 t: **S<sub>2</sub>** (ms)에 의해 지정되는 검사 시간

7. X0~X3 중 하나가 지정된다면, 지정된 장치는 고속카운터의 펄스 입력 또는 외부 가로채기 신호로서 사용될수 없을것입니다.
8. SPD 명령이 실행되고 M1100 (SPD 명령 샘플링 1 회 표시문자)=On 이 되면, SPD 명령은 한번의 샘플링을 실행할 것입니다. M1100 이 Off 에서 On 이 될때 SPD 명령은 한번에 데이터를 모을 것이며 그 후에 중지할것입니다. 모음을 계속하고 싶다면 M1100 을 Off 로 하고 SPD 명령을 다시 실행하십시오.

## 프로그램 예 :

1. X7=On 일 때, D2 은 X1 으로부터 고속 펄스 입력을 셀 것입니다. 1,000ms 가 지나면, 자동적으로 세는 것을 멈추며 결과는 D0 에 저장될 것입니다.
2. 1000ms 세기가 완료된 후, D2 의 내용이 0 으로 재시동될 것입니다. X7 이 다시 On 이 되면 D2 가 다시 셀것입니다.



## 주의:

ES/EX/SS 직렬 유형 Is (V5.7 and above)에서 X1 나 X2 가 SPD 명령에 쓰인다면, 연관된 고속 카운터나 외부 가로채기 I101, I201 는 사용될 수 없습니다.

API	연상 기호		피연산자			기능	제어기						
57	D	PLSY	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>D</b>	펄스 Output	ES	EX	SS	SA	SX	SC	EH

유형 OP	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*		
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*		
D		*															
DPLSY: 7 단계																	
DPLSY: 13 단계																	

펄스							16-비트							32-비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S<sub>1</sub>**: 펄스 출력 빈도      **S<sub>2</sub>**: 펄스 출력 수      **D**: 외부 출력(Y0 와 Y1 만 지정가능)

설명:

- 장치 M1133~M1135 와 D1133 과 함께 SA 직렬 MPU 에서 펄스 출력 빈도로 지정된 **S<sub>1</sub>** 와 함께 Y0 펄스 출력 빈도는 50KHz 이상이 가능합니다. 고속 펄스의 특별 D 와 M 의 설명을 위해 2 장을 참조하십시오.

각 직렬 유형 Is 의 출력 빈도 범위			
유형 Is	ES/EX/SS 직렬 유형 Is	SA/SX/SC 직렬 유형 Is	EH 직렬
출력 빈도 범위	1~10,000Hz	Y0:1~32,000Hz Y1:1~10,000Hz	1~200,000Hz

- 펄스 출력 수로 지정된 **S<sub>2</sub>**. 16-비트 명령: 1~32,767. 32-비트 명령: 2,147,483,647.

각 직렬 유형 Is 의 연속 펄스 수		
유형 Is	ES/EX/SS 직렬 유형 Is 와 SA/SX/SC 직렬 유형 Is	EH 직렬 유형 Is (TR 유형 Is)
연속 펄스의 지정된 방법	M1010 (Y0) ON M1023 (Y1) ON	지정된 펄스 출력 수는 K0 에 설정

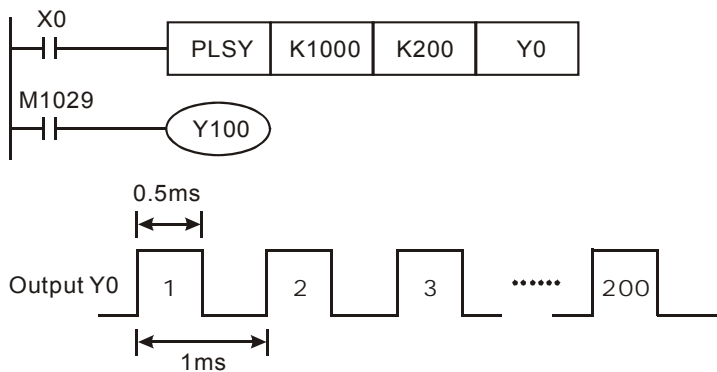
- 지정된 펄스 출력 숫자가 "0" 으로 EH 직렬 방식 Is 에 설정되면 펄스의 제한 없는 숫자가 끊임없이 출력될것이란 뜻입니다. 펄스의 끊임없는 출력의 숫자가 무제한일 때 M1010(Y0) 또는 M1023(Y1)는 On 이어야 합니다.
- 펄스 출력 장치로 지정된 D. EH 직렬 방식 Is 에서, Y0 와 Y2 만이 세부화될수 있습니다. SA/SX/SC/ ES/EX/SS 직렬 방식 Is 에서는, Y0 와 Y1 만이 세부화될수 있습니다.
- PLSY 명령이 실행되면, 펄스 **S<sub>2</sub>** 의 세부지정된 양이 펄스 결과 장치 **D** 를 통해 지정된 펄스 출력 빈도수 **S<sub>1</sub>** 에 출력될 것입니다.
- 프로그램에서 PLSY 명령을 사용할 때 PLSY 명령의 출력은 API 58 PWM 명령과 API 59 PLSR 명령에 같을 수 있습니다.
- SA/SX/SC/ES/EX/SS 직렬 방식 Is 에서 Y0 펄스 출력이 완료된 후, M1029 는 On 이 될것이며 Y1 펄스 출력이 완료되면 M1030 이 On 이 될것입니다. PLSY 명령이 Off 가 되면 M1029 또는 M1030 는 Off 가 될것입니다.
- EH 직렬 방식 Is 에서 Y0 와 Y1 펄스가 완료되면 M1029 는 On 이 되고, Y2, Y3 펄스 출력이 완료되면 M1030 는 On 이 될것입니다. PLSY 명령은 Off 가 되고 M1029 또는 M1030 는 Off 가 됩니다.
- 완료된 표시문자 M1029 와 M1030 의 실행은 명령의 실행이 완료된 후 지워져야합니다.
- PLSY 명령이 실행되면, Y 는 펄스 출력을 시작합니다. 이 때 **S<sub>2</sub>** 가 변한다면 출력은 영향을 받아선

안됩니다. 펄스 출력 수를 바꾸려면 명령 PLSY를 중지시키면 펄스 수가 변할 것입니다.

11. **S<sub>1</sub>**는 PLSY 명령이 실행될 때 바뀔 수 있으며 유효 시간을 변경시킬 수도 있습니다. **S<sub>1</sub>**는 프로그램이 명령 PLSY에 실행될 때 바뀔 수 있습니다.
12. 펄스 출력의 Off TIME 과 On TIME 의 비율은 1:1 입니다.
13. 실제 출력 펄스는 프로그램이 명령 PLSY에 실행될 때 특별 레지스터 D1336~D1339에 저장됩니다. 자세한 내용은 각주를 참조하십시오.
14. 명령 실행 중간에 바뀔수 있는 특별 레지스터 D (D1220, D1221)와 M (M1258, M1259)에 관해 자세한 정보는 “주의”를 참조하십시오.

#### 프로그램 예 :

1. X0=On 일 때, 1KHz 의 200 회 펄스는 출력 Y0 으로부터 가져오며 펄스가 완료되고 나면 M1029=On 이 Y10=On 를 트리거 합니다.
2. X0=Off 일 때, 펄스 출력 Y0 은 즉시 중지됩니다. X0 이 다시 On 이 되면 처음 펄스가 출력되기 시작합니다.



#### 주의:

표시문자 설명:

- M1010:** EH 직렬 MPU 에서 M1010= On 일 때, Y0, Y1, Y2, Y3 은 END 명령이 실행될 동안 펄스를 출력할 것입니다. 출력이 시작되면 M1010 은 자동적으로 Off 될 것입니다.
- SA/SX/SC/ES/EX/SS 직렬 MPU 에서 M1010=On 일 때, Y0 은 제한 없이 연속 펄스를 출력할 수 있습니다. M1010=Off 일 때, Y0 의 펄스 출력 수는 **S<sub>2</sub>** 에 의해 결정이 됩니다.
- M1023:** SA/SX/SC/ES/EX/SS 직렬 MPU 에서 M1023=On 일 때 Y1 은 제한 없이 연속 펄스를 출력할 수 있습니다. M1023=Off 일 때, Y1 의 펄스 출력은 **S<sub>2</sub>**에 의해 결정이 됩니다.
- M1029:** EH 직렬 MPU 에서는 Y0, Y1 펄스 출력이 완료된 후 M1029= On 가 됩니다.
- SA/SX/SC/ES/EX/SS 직렬 MPU 에서는 Y0 펄스 출력이 완료된 후 M1029= On 이 됩니다.
- M1030:** EH 직렬 MPU 에서는 Y2, Y3 펄스 출력이 완료된 후 M1030= On 이 됩니다.

SA/SX/SC/ES/EX/SS 직렬 MPU에서는 Y1 펄스 출력이 완료된 후  
M1030= On 이 됩니다.

M1078: SA/SX/SC/ES/EX/SS 직렬에서 Y0 펄스 출력이 중지됩니다.

M1079: SA/SX/SC/ES/EX/SS 직렬에서 Y1 펄스 출력이 중지됩니다.

M1258: EH 직렬 MPU에서 (PWM 명령) Y0, Y1 펄스 출력 신호가 바뀝니다.

M1259: EH 직렬 MPU에서 (PWM 명령) Y2, Y3 펄스 출력 신호가 바뀝니다.

M1334: EH 직렬 MPU에서 CH0 펄스 출력이 중지됩니다.

M1335: EH 직렬 MPU에서 CH1 펄스 출력이 중지됩니다.

M1336: EH 직렬 MPU에서 CH0 펄스가 지시 표시문자를 출력합니다.

M1337: EH 직렬 MPU에서 CH1 펄스가 지시 표시문자를 출력합니다.

M1338: EH 직렬 MPU에서 CH0 오프셋 펄스가 표시문자를 시작합니다.

M1339: EH 직렬 MPU에서 CH1 오프셋 펄스가 표시문자를 시작합니다.

M1340: EH 직렬 MPU에서 가로채기 (I110)는 CH0 펄스 출력이 완료된 후 발생  
합니다.

M1341: EH 직렬 MPU에서 가로채기 (I120)는 CH1 펄스 출력이 완료된 후 발생  
합니다.

M1342: EH 직렬 MPU에서 가로채기 (I130)는 CH0 펄스 전송과 동시에 발생  
합니다.

M1343: EH 직렬 MPU에서 가로채기 (I140)는 CH1 펄스 전송과 동시에 발생  
합니다.

M1344: EH 직렬 MPU에서 CH0 보정 펄스가 표시문자를 시작합니다.

M1345: EH 직렬 MPU에서 CH1 보정 펄스가 표시문자를 시작합니다.

SA/SX/SC 직렬 MPU의 특별 레지스터 설명:

D1030: 첫 번째 출력 그룹 Y0의 현재 총 펄스 수 (낮은 수).

D1031: 첫 번째 출력 그룹 Y0의 현재 총 펄스 수 (높은 수).

D1032: 두 번째 출력 그룹 Y1의 현재 총 펄스 수 (낮은 수).

D1033: 두 번째 출력 그룹 Y1의 현재 총 펄스 수 (높은 수).

EH 직렬 MPU의 특별 레지스터 설명:

D1220: 첫 번째 출력 그룹 Y0, Y1의 단계 설정: D1220의 마지막 두 비트에 의해  
결정, 다른 비트들은 무효함.

1. K0: Y0 출력
2. K1: Y0, Y1 AB 단계 출력, A가 B를 리드
3. K2: Y0, Y1 AB 단계 출력, B가 A를 리드
4. K3: Y1 출력

D1221: 두 번째 출력 그룹 Y2, Y3의 단계 설정: D1221의 마지막 두 비트에 의해  
결정, 다른 비트들은 무효함.

1. K0: Y2 출력
2. K1: Y2, Y3 AB 단계 출력, A가 B를 이끌다
3. K2: Y2, Y3 AB 단계 출력, B가 A를 이끌다

## 4. K3: Y3 출력

- D1328: CH0 오프셋 펄스 수 (낮은 수)  
 D1329: CH0 오프셋 펄스 수 (높은 수)  
 D1330: CH1 오프셋 펄스 수 (낮은 수)  
 D1331: CH1 오프셋 펄스 수 (높은 수)  
 D1332: CH0 잔여 펄스 수 (낮은 수)  
 D1333: CH0 잔여 펄스 수 (높은 수)  
 D1334: CH1 잔여 펄스 수 (낮은 수)  
 D1335: CH1 잔여 펄스 수 (높은 수)  
 D1336: 첫 번째 그룹의 현재 총 출력 펄스 수 (Y0, Y1) (낮은 수).  
 D1337: 첫 번째 그룹의 현재 총 출력 펄스 수 (Y0, Y1) (높은 수).  
 D1338: 두 번째 그룹의 현재 총 출력 펄스 수 (Y2, Y3) (낮은 수).  
 D1339: 두 번째 그룹의 현재 총 출력 펄스 수 (Y2, Y3) (높은 수).  
 D1344: CH0 보정 펄스 수 (낮은 수)  
 D1345: CH0 보정 펄스 수 (높은 수)  
 D1346: CH1 보정 펄스 수 (낮은 수)  
 D1347: CH1 보정 펄스 수 (높은 수)

1. 몇몇의 고속 펄스 출력 명령(PLSY, PWM, PLSR)이 Y0 을 한 프로그램에서 펄스를 출력하기 위해 사용하거나 같은 주사 주기에서 동시에 실행이 될 때, PLC 는 최소 단계의 수를 가진 명령을 수행할 것입니다.
2. PLSY 펄스 출력 명령의 설명과 EH 직렬 MPU 의 관련 장치:

PLSY 명령 설명				
피연산자		S1	S2	D
설명		빈도 설정	펄스 수	출력 장치
범위	16-비트	1~32,767Hz	0~32,767	Y0~Y3
	32-비트	1~200KHz	0~2,147,483,647	
정의		K0: 출력 없음 Kn: 지정된 빈도수 출력	K0: 연속 펄스 출력 Kn: 지정된 펄스 출력	D1220, D1221 의 설정 참조

PLSY 명령의 관련 장치 정보(Special D)					
장치 번호		데이터 포맷	속성	최초 값	내용
D1220		16-비트	R/W	K0	첫 번째 출력 펄스 그룹의 차례 설정
D1221		16-비트	R/W	K0	두 번째 출력 펄스 그룹의 차례 설정
D1328	낮은 수	32-비트	R/W	K0	첫 번째 펄스 그룹의 오프셋 펄스 수
D1329	높은 수				
D1330	낮은 수	32-비트	R/W	K0	첫 번째 펄스 그룹의 오프셋 펄스 수
D1331	높은 수				
D1332	낮은 수	32-비트	R/W	K0	첫 번째 펄스 그룹의 잔여 펄스 수
D1333	높은 수				
D1334	낮은 수	32-비트	R/W	K0	두 번째 펄스 그룹의 잔여 펄스 수
D1335	높은 수				
D1336	낮은 수	32-비트	R/W	K0	첫 번째 펄스 그룹의 최근 값(펄스 출력 수의 축적 값)
D1337	높은 수				
D1338	낮은 수	32-비트	R/W	K0	두 번째 펄스 그룹의 최근 값(펄스 출력 수의 축적 값)
D1339	높은 수				
D1341	낮은 수	32-비트	R/W	K200000	최대 출력 빈도수
D1342	높은 수				
D1344	낮은 수	32-비트	R/W	K0	첫 번째 펄스 그룹의 보정 펄스 수
D1345	높은 수				
D1346	낮은 수	32-비트	R/W	K0	두 번째 펄스 그룹의 보정 펄스 수
D1347	높은 수				

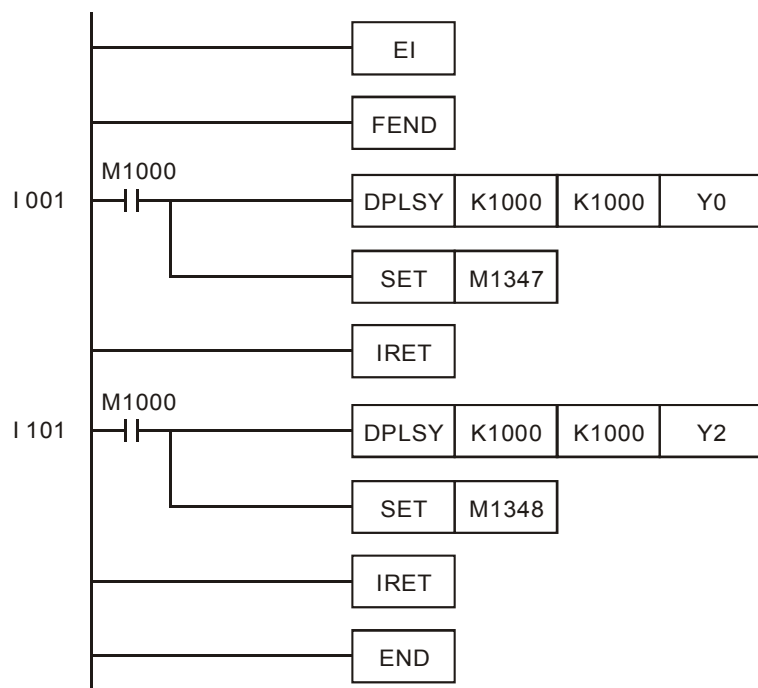
PLSY 명령의 관련 장치 설명(Special M)			
장치 번호	속성	내용	관련 설정 장치
M1010	R/W	두 개의 펄스 출력 그룹 동시에	
M1029	R	첫 번째 펄스 그룹의 표시 문자 지시 종료	
M1030	R	두 번째 펄스 그룹의 표시 문자 지시 종료	
M1334	R/W	첫 번째 펄스 그룹의 펄스 출력 중지	
M1335	R/W	두 번째 펄스 그룹의 펄스 출력 중지	
M1336	R	첫 번째 펄스 그룹의 표시 문자 지시 출력	
M1337	R	두 번째 펄스 그룹의 표시 문자 지시 출력	
M1338	R/W	첫 번째 펄스 그룹의 시작 표시문자 오프셋	D1328, D1329
M1339	R/W	두 번째 펄스 그룹의 시작 표시문자 오프셋	D1330, D1331
M1340	R/W	첫 번째 펄스 그룹 출력이 완료된 후 발생 가로채기	I110
M1341	R/W	첫 번째 펄스 그룹 출력이 완료된 후 발생 가로채기	I120
M1342	R/W	첫 번째 펄스 그룹 출력이 완료된 후 발생 가로채기	I130
M1343	R/W	첫 번째 펄스 그룹 출력이 완료된 후 발생 가로채기	I140
M1344	R/W	첫 번째 펄스 그룹의 표시문자 시작 보정	D1344, D1345
M1345	R/W	두 번째 펄스 그룹의 표시문자 시작 보정	D1346, D1347
M1347	R/W	첫 번째 펄스 그룹의 표시문자 자동 가로채기	
M1348	R/W	두 번째 펄스 그룹의 표시문자 자동 가로채기	



M1347 와 M1348 의 추가 정보:

PLC 가 PLSY 명령을 실행하기 전에 사용자는 END 명령이 주사할 때마다 M 표시문자가 켜졌는지 확인하기 이전에 M1347/M1348=ON 으로 설정 가능합니다. M1347/M1348=ON 의 상태로, PLSY 명령은 펄스 출력을 실행할 것입니다. 따라서, PLSY 의 시작 접촉은 OFF 를 ON 으로 바꾸지 않아도 여전히 ON 이 되어있을 것이며 PLSY 명령은 PLSY 가 펄스 출력을 마친 후에도 출력을 한 상태일 것입니다. PLSY 가 연속 실행한다면 다음 출력 행위까지 주사 시간 지연이 있을 것입니다.

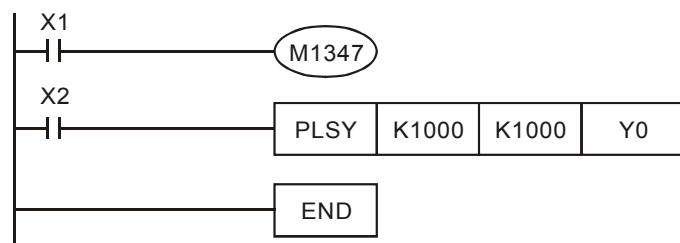
프로그램 예 1:



설명:

1. X0 가 트리거 될 때마다, Y0 이 1,000 펄스를 출력할 것이며; X1 이 트리거 될 때마다 Y2 가 1,000 펄스를 출력할 것입니다.
2. X 가 Y를 출력 펄스로 트리거 할 때는 Y가 펄스 출력을 끝내고 난 후 다음 X 트리거 때까지 PLC 의 한 개 이상의 주사 주기가 있을 것입니다.

프로그램 예 2:



설명:

X1 와 X2 가 모두 켜졌을 때, Y0 은 펄스 출력을 계속 할 것입니다. Y0 출력이 1,000 펄스가 될 때마다 다음 1,000 펄스 출력이 있을 때까지 짧은 정지상태(1 주사 주기 정도)가 있을 것입니다.

API	연상기호	연산자	기능	제어장치						
58	PWM	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b>	Pulse 폭 변조	ES	EX	SS	SA	SX	SC	EH

유형	비트 장치				워드 장치											프로그램 단계	
연산자	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PWM: 7 단계	
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*		
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*		
D		*															

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

연산자:

**S<sub>1</sub>**: Pulse 출력 폭    **S<sub>2</sub>**: Pulse 출력 주기    **D**: Pulse 출력 장치 (Y1 같은 특별한 경우에만 사용)

설명:

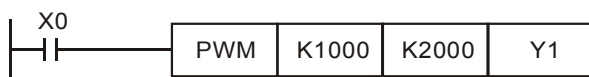
- S<sub>1</sub>**은 pulse 출력 폭이 t:0~32,767ms 와 같음을 명기합니다..
- S<sub>2</sub>**는 pulse 출력 주기가 T:1~32,767ms 이며, **S<sub>1</sub> ≤ S<sub>2</sub>** 와 같음을 명기합니다..
- D**는 pulse 출력 장치를 명기합니다. MPU 의 EH 시리즈에서, **D**는 Y0, Y2 와 같다고 할 수 있습니다. SA/SX/SC/ES/EX/SS 모델 시리즈에서, **D**는 Y1 과 같다고 할 수 있습니다.

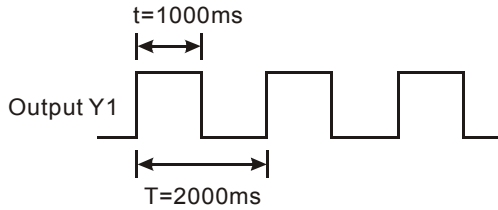
각 모델의 시리즈 별 pulse 출력 변조		
모델	ES/EX/SS/EX 모델 시리즈와 SA/SX/SC 모델 시리즈	EH 모델 시리즈
PWM 출력	Y1	Y0, Y2

- EH 모델 시리즈 프로그램 내에서 PWM 명령은 두 번 사용할 수 있습니다. SA/SX/SC/ES/EX/SS 모델 시리즈 프로그램 내에서 PWM 명령은 한 번 사용할 수 있습니다.
- PWM 명령이 프로그램 내에서 사용되는 동안 위의 출력은 API 57 PLSY, API 59 PLSR 명령의 출력과 같을 수 없습니다.
- PWM 명령이 실행되는 동안, **S<sub>1</sub>** pulse 출력 폭과 **S<sub>2</sub>** pulse 출력 주기는 **D** pulse 출력 장치를 지나게 됩니다.
- MPU 의 SA 시리즈에서, **S<sub>1</sub> ≤ 0**, **S<sub>2</sub> ≤ 0**, 또는 **S<sub>1</sub> > S<sub>2</sub>** 의 에러가 발생되면, (M1067 과 M1068 은 ON 이 아닙니다.) pulse 출력 장치를 지나가는 출력은 없습니다. **S<sub>1</sub> = S<sub>2</sub>** 일 때, pulse 출력 장치는 ON 으로 됩니다.
- MPU 의 EH 시리즈에서, **S<sub>1</sub> < 0**, **S<sub>2</sub> ≤ 0** 또는 **S<sub>1</sub> > S<sub>2</sub>** 의 에러가 발생되면, 연산자가 발생하며 (M1067 과 M1068 은 ON 이 됩니다.) pulse 출력 장치는 출력을 발생하지 않습니다. **S<sub>1</sub> = 0** 일 때, M1067 과 M1068 는 ON 이 되지 않으며 pulse 출력 장치는 출력을 발생하지 않습니다. **S<sub>1</sub> = S<sub>2</sub>** 일 때, pulse 출력 장치는 ON 을 나타냅니다.
- PWM 명령이 실행되는 동안 **S<sub>1</sub>**, **S<sub>2</sub>**는 변경될 수 있습니다.

프로그램 사용 예 :

X0=On 일 때, Y1 출력은 pulse 를 따라갑니다. X0=Off 일 때, Y1 출력 또한 Off 입니다.





**참고:**

기호 설명:

- M1010:** MPU의 EH 시리즈에서, M1010= On 일 때, END 명령이 실행되는 동안 CH0와 CH1는 pulse를 출력합니다. 출력이 시작되면, M1010는 자동으로 Off가 됩니다.
- M1067:** MPU의 EH 시리즈에서, 연산자 오류가 발생하면, M1067=On이 됩니다.
- M1070:** MPU의 SA/SX/SC/ES/EX/SS 시리즈에서, PWM 명령에서 Y1을 출력하면, pulse 장치는 변경됩니다. M1070=On일 때, pulse 장치는  $100\mu\text{s}$ 이 되며, M1070=Off일 때, pulse 장치는  $1\text{ms}$ 이 됩니다.
- MPU의 EH 시리즈에서, PWM 명령에서 첫 번째 pulse 출력 그룹이 Y0가 되면, pulse 장치는 변경됩니다. M1070=On일 때, pulse 장치는  $100\mu\text{s}$ 이 되며, M1070=Off일 때, pulse 장치는  $1\text{ms}$ 이 됩니다.
- M1071:** MPU의 EH 시리즈에서, PWM 명령에서 첫 번째 pulse 출력 그룹이 Y2가 되면, pulse 장치는 변경됩니다. M1071=On일 때, pulse 장치는  $100\mu\text{s}$ 이 되며, M1071=Off일 때, pulse 장치는  $1\text{ms}$ 이 됩니다.
- M1258:** MPU의 EH 시리즈에서, (PWM 명령 시) pulse 출력 신호가 Y0, Y1으로 변경됩니다.
- M1259:** MPU의 EH 시리즈에서, (PWM 명령 시) pulse 출력 신호가 Y2, Y3으로 변경됩니다.
- M1334:** MPU의 EH 시리즈에서, CH0 pulse 출력이 정지됩니다.
- M1335:** MPU의 EH 시리즈에서, CH1 pulse 출력이 정지됩니다.
- M1336:** MPU의 EH 시리즈에서, CH0 pulse 출력 지시 기호가 발생합니다.
- M1337:** MPU의 EH 시리즈에서, CH1 pulse 출력 지시 기호가 발생합니다.

몇몇의 고속 pulse 출력 명령(PLSY, PWM, PLSR)을 사용해 하나의 프로그램에서 Y0 pulse 출력이 발생하고 동시에 같은 검사 주기가 실행되면, PLC는 가장 적은 번호를 가진 명령을 수행합니다.

**M1070 :** 프로그램이 실행되는 동안 변경은 이루어질 수 없습니다.

**MPU의 EH 시리즈 기능:**

PWM 명령과 EH 모델 시리즈와 관련된 장치 설명

장치 번호	데이터 형식	속성	관련 장치 설정
M1010	R/W	동시에 2 개의 pulse 그룹 출력	
M1070	R/W	PWM pulse 시간 장치를 Y0와 Y1으로 변경	
M1071	R/W	PWM pulse 시간 장치를 Y2와 Y3으로 변경	

장치 번호	데이터 형식	속성	관련 장치 설정
M1258	R/W	PWM pulse 출력 신호를 Y0 와 Y1 으로 변경	
M1259	R/W	PWM pulse 출력 신호를 Y2 와 Y3 으로 변경	
M1334	R/W	첫 번째 pulse 그룹의 Pulse 출력 정지	
M1335	R/W	두 번째 pulse 그룹의 Pulse 출력 정지	
M1336	R	첫 번째 pulse 그룹의 출력 지시 기호	
M1337	R	두 번째 pulse 그룹의 출력 지시 기호	

API	연상기호			연산자				기능				제어장치							
59	D	PLSR		<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>S<sub>3</sub></b>	<b>D</b>	Pulse Ramp				ES	EX	SS	SA	SX	SC	EH	

연산자 \ 유형	비트 장치				워드 장치										프로그램 단계							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PLSR: 9 단계  DPLSR: 17 단계						
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*							
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*							
S <sub>3</sub>					*	*	*	*	*	*	*	*	*	*	*							
D		*																				

PULSE								16-bit								32-bit								
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		

연산자:

**S<sub>1</sub>**: 최대 속도(HZ)    **S<sub>2</sub>**: pulse 수    **S<sub>3</sub>**: 가/감속 시간 (ms)

**D**: Pulse 출력 장치. MPU의 EH 시리즈는 Y0와 Y2만 지시합니다. MPU의 ES/SA 시리즈는 Y0과 Y1만 지시할 수 있습니다. (트랜지스터 출력 모듈을 사용하는 것이 좋습니다.)

설명 :

- S<sub>1</sub>**: 최대 pulse 출력 빈도 수(Hz)를 뜻합니다. 16-bit 명령에서의 설정은 10에서 32,767 Hz이고 32-bit 명령에서의 설정은 10에서 200,000 Hz입니다. 최대 속도는 10의 배수라 간주되지만, 만약 아니라면, 첫 번째 장치는 자동으로 중지될 것입니다. 최대 속도의 1/10은 시간당 가/감속의 변화를 나타냅니다. 스텝 모터의 요구 가속과 스텝 모터가 충돌할 경우 결과 값이 나올 수 없음을 기억하십시오.
- S<sub>2</sub>**: Pulse 출력량(PLS)을 뜻합니다. 16-bit 명령에서의 설정값은 110~32,767 (PLS)이고 32-bit 명령에서의 설정값은 110~2,147,483,647(PLS)입니다. 만일 설정 값이 110보다 아래이면, pulse는 일반적으로 출력될 수 없습니다.
- S<sub>3</sub>**: 가/감속 시간(ms)을 뜻합니다. 설정값은 5,000ms보다 아래 값을 가집니다. 가속 시간과 감속 시간은 서로 같아지며 서로 다른 시간으로는 설정할 수 없습니다.
- 가/감속 시간은 최대 검색 시간(D1012)의 10시간을 넘게 됩니다. 만일 설정 값이 10시간보다 아래가 되면, 가/감속도의 경사는 부정확하게 됩니다.
- 가/감속 시간의 최소 설정값은 아래의 방정식에 대입된 값과 같게 할 수 있습니다.

$$\text{S}_3 \geq \frac{90000}{\text{S}_1}$$

- 만일 설정값이 위의 방정식보다 적은 값이 된다면, 가/감속 시간은 최고가 될 것이며 만일 설정값이 90000/S<sub>1</sub>보다 적으면, 결과값은 90000/S<sub>1</sub>보다 더 큰 값이 될 것이다.
- 가/감속 시간의 최대 설정값은 아래의 방정식에 대입된 값과 같게 할 수 있습니다.

$$\text{S}_3 \leq \frac{\text{S}_2}{\text{S}_1} \times 818$$

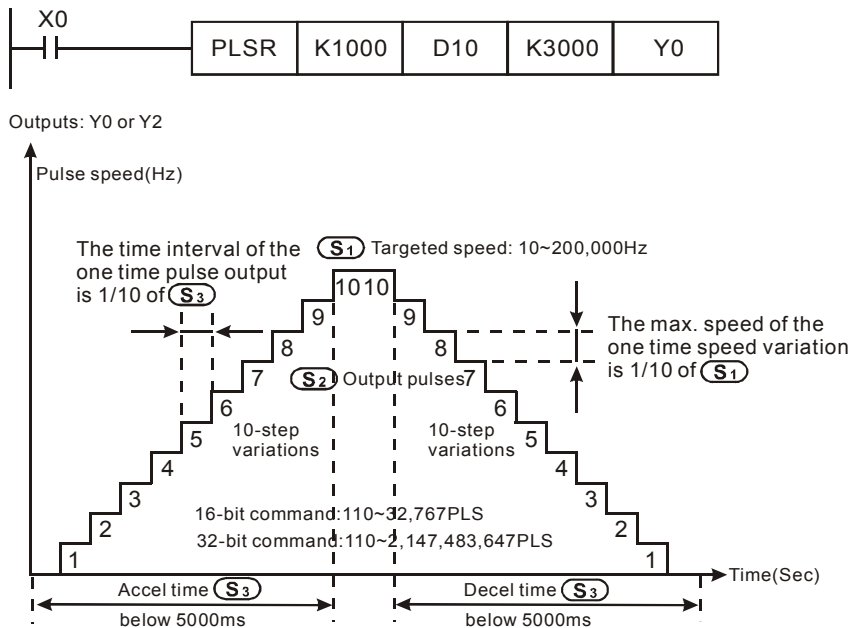
- 가/감속도의 변화값은 10으로 고정됩니다. 만일 가/감속 시간이 최대(최소) 설정값보다 더 크게(적게) 입력되면, 최대(최소) 설정 값은 더 큰 값(적은 값)을 가지게 됩니다.
- PLSR 명령은 가/감속도 기능을 가진 pulse 출력 명령입니다. pulse 곡선이 목표 속도에 도달할 때까지 정적 상태를 나타낸다면 가속이 실행되는 것이고 목표 속도에 거의 도달하면서부터 속도가 느려집니다. 목표까지 남은 거리가 한 번의 출력으로 도달되는 거리이면 pulse 곡선은 멈춰집니다.
- 총 pulse 번호의 양과 가/감속 시간 S<sub>2</sub>, 최대 빈도수 S<sub>1</sub>이 설정된 후, PLSR 명령이 실행되면, pulse 출력

장치 **D**의 출력과 같은 출력이 발생합니다. 처음으로 출력 빈도수가 최고 빈도수  $S_1/10$ 의 1/10 보다 더 높아지고 각 출력 빈도수의 시간은  $S_3$ 의 1/9와 같은 값으로 고정됩니다.

11. PLSR 명령이 실행되어 각  $S_1$ ,  $S_2$  또는  $S_3$ 의 값이 변화되면, 출력에 아무런 영향을 미칠 수 없습니다.
12. 첫 번째 pulse 그룹(Y0, Y1)의 pulse 번호 다음에  $S_2$ 에 의해 설정된 M1029=On으로 출력이 완료됩니다. PLSR 명령이 다시 활성화되면, M1029나 M1030은 0이 되고, PLSR 명령이 완료된 다음에는 1로 변경됩니다.
13. 첫 번째 출력 그룹(Y0, Y1)의 pulse 출력과 두 번째 출력 그룹(Y2, Y3)의 현재 값은 D1336~D1339까지의 특정 레지스터에 저장됩니다.
14. 각 단계별 가속이 이루어지는 동안, pulse 번호(각 시간별 빈도 수)는 정수가 아닐 것입니다. 그러나 PLC의 출력 연산자는 모든 정수로 수행됩니다. 따라서, 각 시간의 간격은 같지 않아야 할 것이며 몇몇은 이탈이 발생하기도 합니다. 손실은 빈도수에 의해 확정되며 10진법수를 포기합니다. Pulse 출력 값이 정확하게 나오도록 하기 위하여, PLC는 마지막으로 산출된 부정확한 pulse 값의 간격을 채워 넣습니다.

#### 프로그램의 예 :

1. X0=On일 때, PLSR 명령의 최고 빈도수는 1,000Hz입니다. D10의 총 pulse 값의 양은, 가/감속 시간은 3,000ms이며 pulse들의 출력은 Y0 출력입니다. 모든 pulse가 출력되었을 때 모든 시간의 출력 빈도는 1,000/10 Hz입니다. 각 pulse 출력 빈도의 시간은 3,000/9으로 고정됩니다.
2. X10이 OFF일 때, 출력은 중단될 것이며, 다시 ON이 되면 pulse는 0에서부터 계산을 계속해 나갈 것입니다.



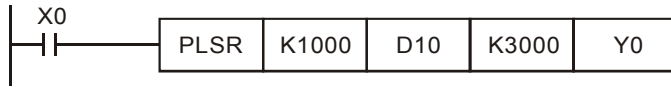
#### 참고:

1. PLSR 명령이 프로그램 내에서 사용되는 동안, API 58 PWM 명령의 출력은 API 57 PLSY의 출력과 같을 수 없습니다.
2. 몇몇의 고속 pulse 출력 명령(PLSY, PWM, PLSR)은 한 개의 프로그램에서 Y0 pulse 출력 시 사용되며 동시에 같은 검색 주기가 실행되고 PLC는 가장 적은 단계 번호의 명령을 수행할 것입니다.
3. 최고 pluse 출력 빈도는 PLSY 명령에서 수행된 것과 같습니다. MPU의 SA 시리즈와 M1133~M1135 그리고 D1133 장비는, Y0 pulse 출력 빈도를 50KHz 보다 높게 할 수 있습니다. 2장에서

설명한 특정 고속 pulse 인 특정 D 와 M 의 기능을 참고하시기 바랍니다.

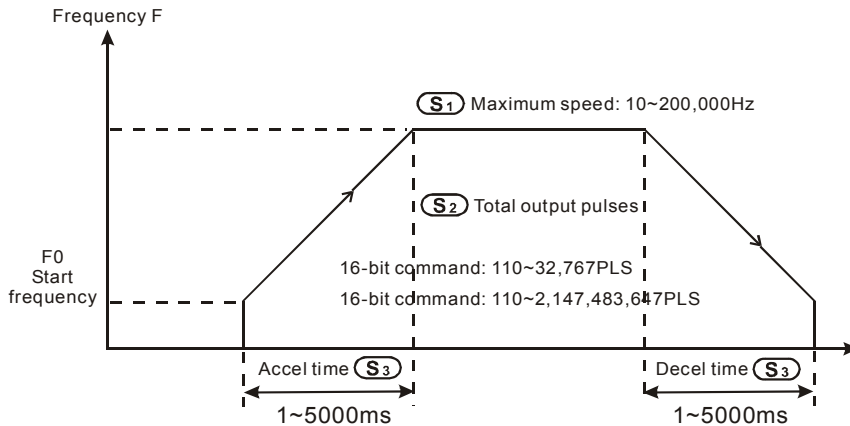
## MPU 의 EH 시리즈의 기능:

1. MPU 의 EH 시리즈 내 명령과 관련 장치에 대한 설명입니다.



2. 이 명령의 pulse 범위 속도는 10~200,000Hz 입니다. 그리고 만일 고속으로 설정되고 이 범위의 가/감속 시간이 초과되면, 명령에 의해 설정된 범위는 사용할 수 있습니다.

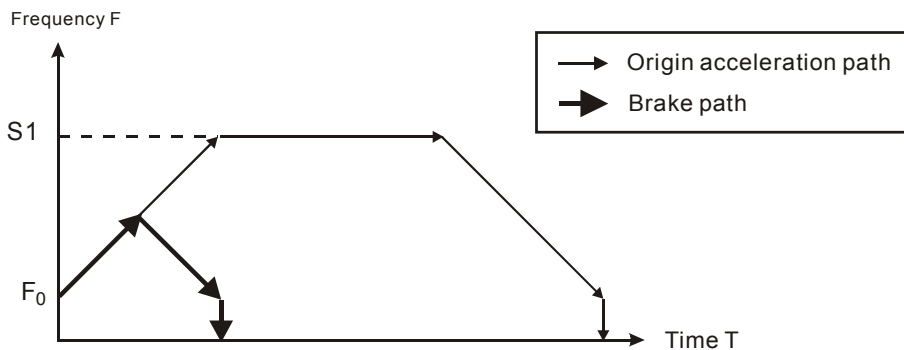
명령 설명				
연산자	S1	S2	S3	D
설명	최대 빈도수	총 pulse 양	가/감속 시간	출력 장치
범위	16-bit	10~32,767Hz	110~32,767	Y0~Y3
	32-bit	10~200KHz	110~2,147,483,647	
정의	K0: 출력이 없음 Kn: 특정 출력 빈도	Kn: 특정 pulse 출력	기호: M1067 M1068	D1220, D1221 의 설정 참조



PLSR 명령과 관련된 장치 설명(특정 장치 D)				
장치 번호	데이터 형식	속성	초기값	내용
D1220	16-bit	R/W	K0	첫 번째 출력 pulse 그룹의 설정 상태
D1221	16-bit	R/W	K0	두 번째 출력 pulse 그룹의 설정 상태
D1336 낮은 word	32-bit	R/W	K0	첫 번째 pulse 그룹의 정확한 값 (축적된 pulse 출력 값)
D1337 높은 word				
D1338 낮은 word	32-bit	R/W	K0	두 번째 pulse 그룹의 정확한 값 (축적된 pulse 출력 값)
D1339 높은 word				
D1340	16-bit	R/W	K200	시작 빈도
D1341 낮은 word	32-bit	R	K200000	최대 출력 빈도
D1342 높은 word				

PLSR 명령과 관련된 장치 설명(특정 장치 M)			
장치 번호	속성	내용	관련된 장치 설정
M1010	R/W	동시에 출력되는 두 개의 pulse 그룹	
M1029	R/W	첫 번째 pulse 그룹의 마지막 지시 기호	
M1030	R/W	두 번째 pulse 그룹의 마지막 지시 기호	
M1334	R/W	첫 번째 pulse 그룹의 pulse 출력 정지	
M1335	R/W	두 번째 pulse 그룹의 pulse 출력 정지	
M1336	R	첫 번째 pulse 그룹의 출력 지시 기호	
M1337	R	두 번째 pulse 그룹의 출력 지시 기호	
M1067	R/W	실행 프로그램 에러 기호	
M1068	R	실행 래칭 에러	D1068

3. In EH 모델 시리즈에서, 만일 가/감속 시간이 최대 가속 빈도수에 도달하지 못한다면, 가/감속 시간과 최대 빈도수는 자동으로 조정될 것입니다.
4. PLSR 명령의 매개 변수는 반드시 명령이 실행되기 전에 입력되어야 합니다.
5. 모든 가속과 감속 명령은 제동 기능과 함께 합니다. PLC의 가속 기능이 실행되었지만 접속 스위치가 갑자기 Off가 되면, 제동 기능이 활성화되며 PLC는 가속도와 같은 경사로 감속할 것입니다.





API	연상기호	연산자	기능	제어장치						
60	IST	<b>(S)</b> <b>(D<sub>1</sub>)</b> <b>(D<sub>2</sub>)</b>	수동/자동 제어	ES	EX	SS	SA	SX	SC	EH

유형	비트 장치				워드 장치											프로그램 단계	
연산자	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	IST: 7 단계	
S	*	*	*														
D <sub>1</sub>				*													
D <sub>2</sub>				*													

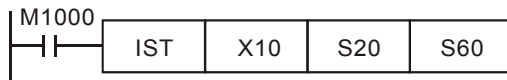
#### 연산자:

**S:** 입력 번호의 시작을 가리킵니다.(연산자 **S** 는 연속된 8 개의 장치를 점유합니다.) **D<sub>1</sub>:** 자동 모드에서 포인트의 단계 상태를 표현하는 가장 작은 번호입니다. **D<sub>2</sub>:** 자동 모드에서 포인트의 단계 상태를 표현하는 가장 큰 번호입니다.

#### 예:

- 연산자 **D<sub>1</sub>** 와 **D<sub>2</sub>** 의 유효 범위: ES 시리즈의, S20~S127; SA/EH 시리즈의, S20~S899.
- IST 명령은 프로그램 내에서 한 번만 사용할 수 있습니다.
- ES/SA 시리즈는 인덱스 레지스터 E, F 를 지원하지 않습니다.
- IST 는 특정 보조 릴레이에 알맞은 자동 제어 명령을 주는 단계별 ladder 제어 절차의 특정 초기 상태 공급하기에 알맞은 명령이다.

#### 프로그램 예 1:



**S** X10: 개별의 명령(수동 명령)  
X11: 0 포인트 반환  
X12: 단계별 명령  
X13: 명령의 한 주기

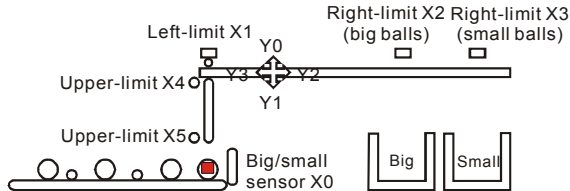
X14: 연속된 명령  
X15: 시작 스위치로 0 포인트 반환  
X16: 시작 스위치  
X17: 정지 스위치

- IST 명령이 실행되었을 때, 특정 보조 릴레이는 아래와 같이 자동으로 스위치를 갖습니다.  
M1040: 이동 억제  
M1041: 이동 시작  
M1042: pulse 상태  
M1047: STL 모니터 가능  
S0: 수동 명령/초기화 상태 단계 포인트  
S1: 0 포인트 반환/초기화 상태 단계 포인트  
S2: 자동 명령/초기화 상태 단계 포인트
- IST 명령이 사용될 때, S10~S19 는 0 포인트를 반환하는 명령을 가지며 이 상태의 단계 포인트는 일반적인 단계 포인트를 사용할 수 없습니다. 하지만, S0~S9 단계 포인트를 사용하면, S0 는 “수동 명령”으로 초기화, S1 은 “자동으로 0 포인트 반환 명령”으로 초기화, 그리고 S2 는 “자동 명령”으로 초기화 합니다. 이와 같이, 프로그램을 단계 포인트로 처음 사용할 때 세 가지의 초기화 상태의 세 가지 순환이 있습니다.
- 스위치가 S1 (0 포인트 반환 모드)일 때, 0 포인트 반환은 S10~S19 가 On 으로 된 상태가 하나라도 있다면 어떤 활동도 하지 않습니다.
- 스위치가 S2 (자동 명령 모드)일 때, 자동 명령은 S 사이의 D<sub>1</sub> 에서 D<sub>2</sub> 가 On 으로 된 상태가 하나라도 있거나 M1043=On 이면 어떤 활동도 하지 않습니다.

#### 프로그램 예 2:

예: 로봇 팔 제어 (IST 명령 사용):

- a) 요청 작동: 예를 들어, 두 종류의 공(크고 작은)을 분리하고 다른 상자에 옮깁니다. 통제를 위한 제어 판넬을 분리합니다.
- b) 로봇 팔 제어: 로봇의 팔을 내리고, 공을 선택한 후, 로봇 팔을 들고, 오른쪽으로 옮긴 후, 로봇의 팔을 내리고, 공을 놓고, 다시 로봇의 팔을 들어, 왼쪽으로 이동하면 작동에 대한 명령은 끝이 납니다.
- c) I/O 장치:



- d) 명령 모드:

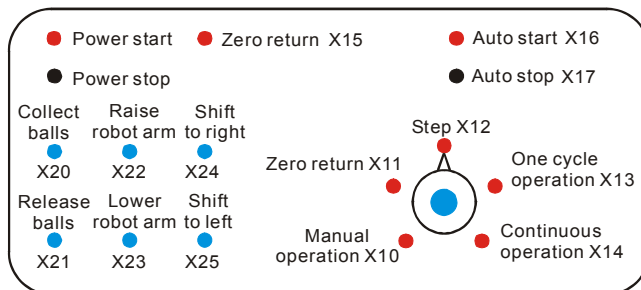
수동 명령 모드: 한 개의 버튼으로 ON 또는 OFF 명령을 가져옵니다..

0 포인트 반환 모드: 0 포인트 버튼을 누르면 기계는 자동으로 0 포인트 모드로 변환합니다..

자동 스텝 모드(단계별 명령 모드/명령 한 주기 모드/연속된 명령 모드):

- 단계별 명령 모드: 언제든지 **START** 버튼이 눌러지면 한 단계 앞으로 갑니다.
- 명령 한 주기 모드: **START** 버튼을 누르면 0 포인트에서 자동으로 명령 한 주기로 수행되며 0 포인트는 정지됩니다. **STOP** 버튼을 누르면 실행 중 정지 명령을 내립니다. **START** 버튼을 한 번 더 누르면, 명령은 계속 이어지며 0 포인트는 정지됩니다.
- 연속 명령 모드: **START** 버튼을 누르면 0 포인트에서 명령이 시작됩니다. **STOP** 버튼을 누르면, 0 포인트에서 명령이 정지됩니다.

- e) 제어 판넬

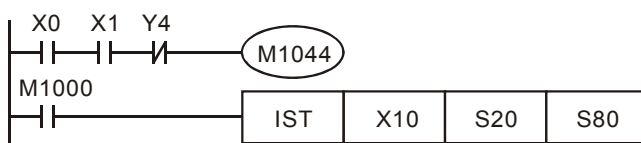


큰/작은 센서 X0.

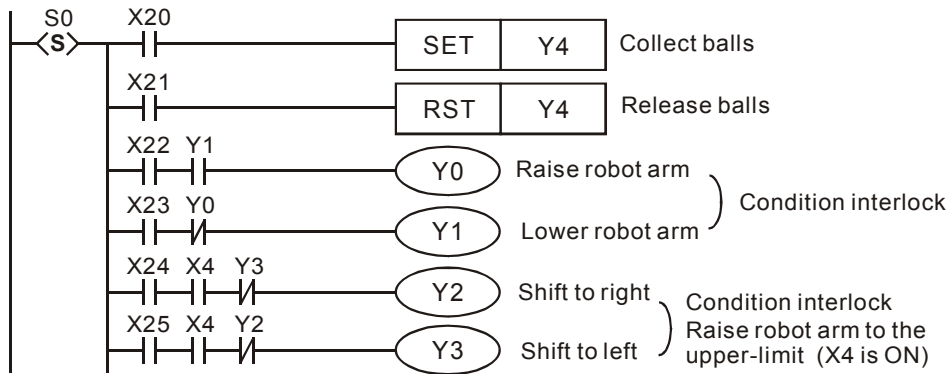
X1 은 로봇 팔의 왼쪽 영역에 있으며, X2 (큰 공)는 오른쪽 영역에, 오른쪽 경계 X3 (작은 공) 역시 오른쪽 영역에, X4 는 위쪽 영역에 있으며, 아래쪽 영역에는 X5 가 있습니다.

Y0 인 로봇 팔을 들어, Y1 인 로봇 팔을 낮추고, Y2 인 오른쪽으로 이동하며, Y3 인 왼쪽으로 다시 이동합니다. Y4 그리고 공을 집습니다.

**START 순환:**

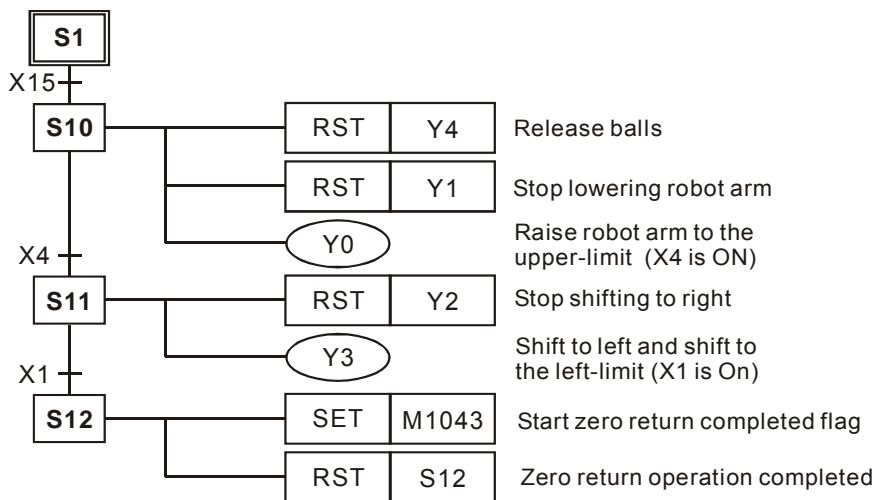


## 수동 명령 모드:

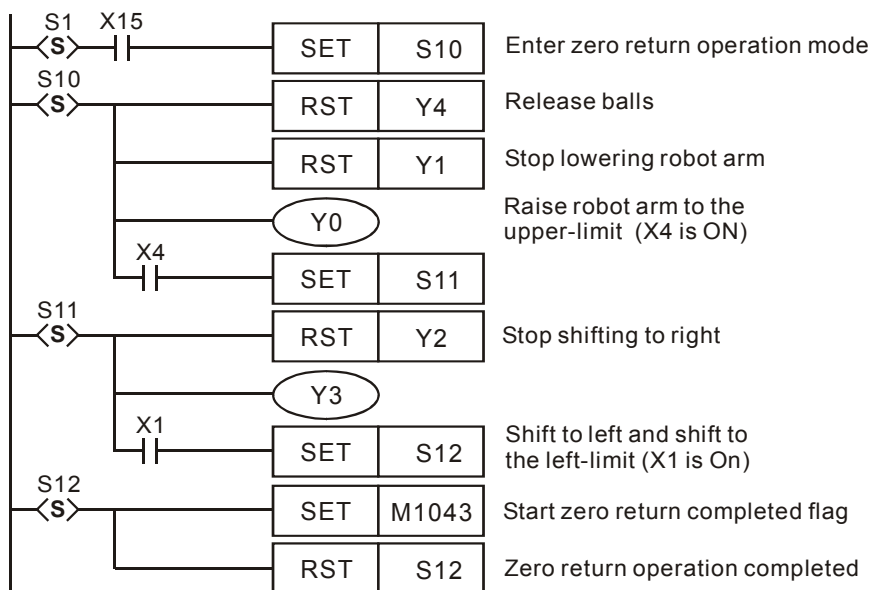


## 0 포인트 반환 모드:

SFC 형태:

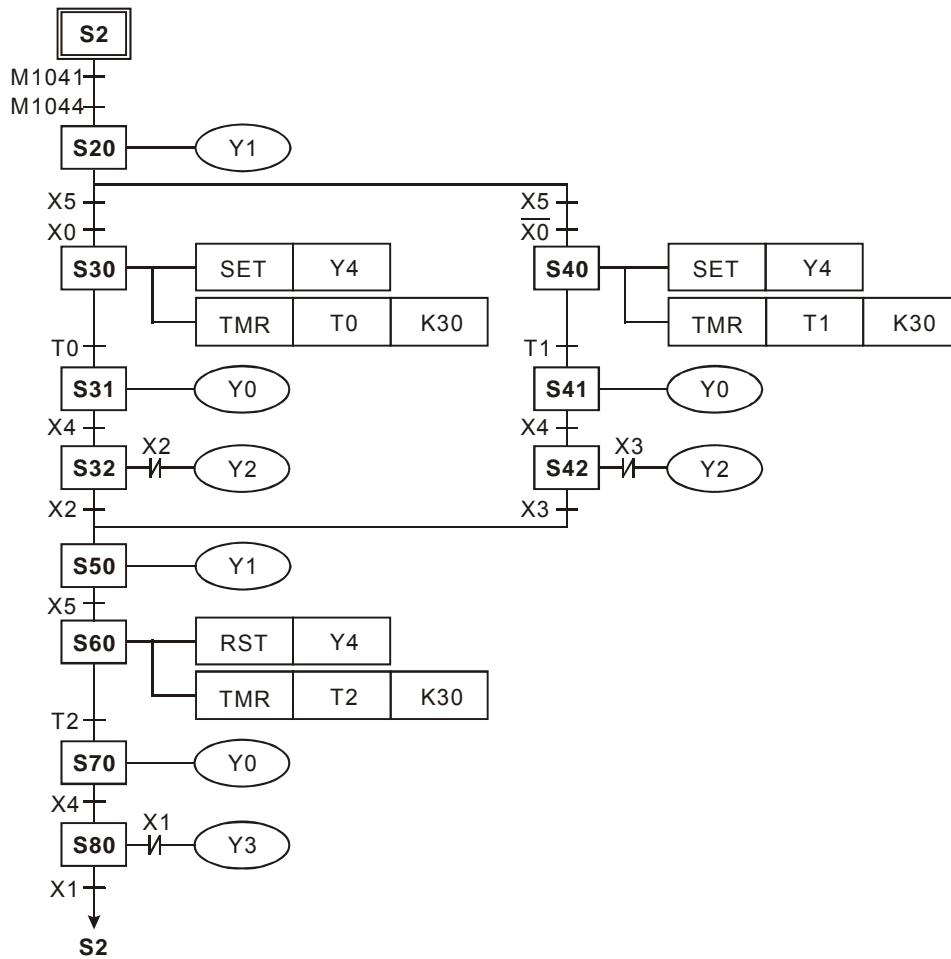


Ladder 도표:

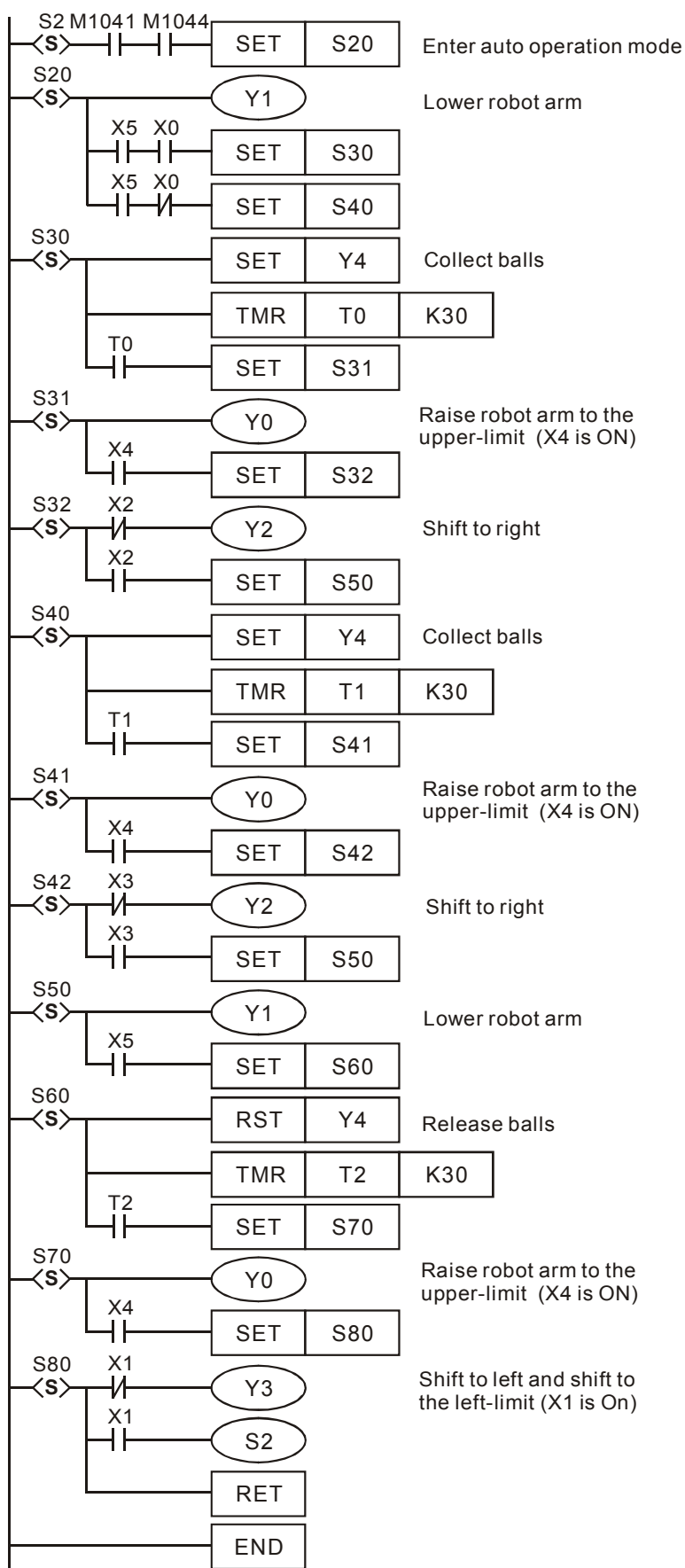


자동 명령(단계 / 한 주기 / 연속 명령 모드):

SFC 형태:



Ladder 도표:



참고:

기호 설명:

M1040: 단계 포인트의 이동이 억제됩니다. M1040=ON 일 때, 모든 단계 포인트의 이동은 억제됩니다.

1. 수동 명령 모드: M1040 은 ON 으로 지속됩니다.
2. 0 포인트 반환 모드/명령 한 주기 모드: STOP 버튼을 누르고 START 버튼을 누르는 동안, M1040 은 ON 으로 지속됩니다.
3. 단계별 명령 모드: M1040 은 ON 으로 지속되고, and START 버튼이 눌러질때만 OFF 가 됩니다.
4. 연속 명령 모드: PLC 가 STOP→RUN 이 될 때, M1040 은 ON 으로 지속되고, START 버튼이 눌러지면 OFF 가 됩니다.

M1041: 단계 포인트의 이동 시작: 특정 보조 릴레이는 다음 단계 포인트로 가기 위한 최초 단계 포인트(S2)의 이동을 반영합니다.

1. 수동 명령 모드/0 포인트 반환 모드: M1041 은 OFF 로 지속됩니다.
2. 단계별 명령 모드/한 주기 명령 모드: M1041 는 START 버튼이 눌러질때까지 OFF 로 지속될 것입니다.
3. 연속 명령 모드: START 버튼을 누를때까지 ON 이 지속되며 STOP 버튼을 누르면 OFF 가 지속됩니다.

M1042: START pulse: 버튼을 눌렀을 때 오직 한 개의 pulse 만 보내집니다.

M1043: 0 포인트 반환 완료: M1043 =ON 가 지속되면, RESET 명령이 실행되었음을 의미합니다.

M1044: 원래의 상태: 연속 명령 모드 아래, 원래의 상태 M1044 가 있으며, 다음 스텝 포인트로 이동하기 위한 초기 단계 포인트(S2)의 ON 명령이 계속 실행되어 있는 것입니다.

M1045: 모든 출력이 이루어지지 않도록 재설정 합니다.

만일 다음의 상태가 실행되면: (장비는 0 포인트가 아닙니다.):

- A. 수동 제어 S0 에서 0 포인트 반환 S1 으로 합니다.
  - B. 자동 명령 S2 에서 수동 명령 S0 으로 합니다.
  - C. 자동 명령 S2 에서 0 포인트 반환 S1 으로 합니다.
1. M1045=Off 이고 S 의 D1~D2 가 ON 일 때, 단계 포인트의 SET Y 출력과 활동은 Off 로 지워집니다.
  2. M1045 =On 일 때, SET Y 출력은 따로 보존되며 단계 포인트는 활동하는동안 Off 로 지워질 것입니다.
  3. 포인트 반환 S1 에서 수동 명령 S0 (기계는 0 포인트에 있습니다.) 이 실행된다면, 비록 M1045=On 이거나 M1045=Off 이어도, SET Y 출력은 따로 보존되며 단계 포인트 활동은 Off 로 지워질 것입니다.

M1046: STL 상태가 On 으로 설정: 만일 하나의 단계 포인트 S 가 On 이면, M1046=On 입니다. M1047 이 On 일 때, M1046 은 On 이 될 것이며 S 또한 On 이 될 것입니다. 게다가, 8 개의 포인트는 S 가 On 이 되기전에 D1040~D1047 에 저장됩니다.

M1047: STL 모니터 가능. IST 시작 명령이 실행되면, M1047 은 On 으로 될 것이며 한 번 IST 명령이 On 으로 된 각 검색 시간에 대해 On 으로 될 것입니다. 이 기호는 모든 S 를 모니터할 때 사용됩니다.

D1040~D1047: S 단계 포인트 1-8 의 상태 번호가 ON 입니다.

API	연상기호			연산자				기능				제어장치						
61	D	SER	P	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>D</b>	<b>n</b>	스택 데이터 검색				ES	EX	SS	SA	SX	SC	EH

연산자 \ 유형	비트 장치				워드 장치										프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SER, SERP: 9 단계 DSER, DSERP: 17 단계		
S <sub>1</sub>							*	*	*	*	*	*	*					
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*			
D								*	*	*	*	*	*					
N					*	*							*					

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

연산자:

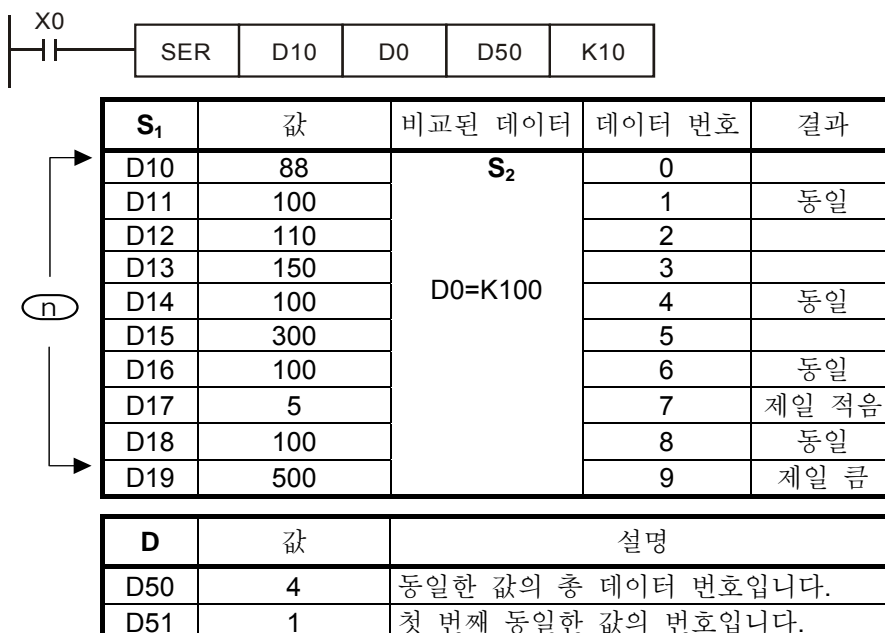
**S<sub>1</sub>**: 소스 시작    **S<sub>2</sub>**: 값 비교    **D**: 각 값을 비교해 결과 값이 저장된 목적지의 시작(연속된 5 개 장치 점유)  
**n**: 비교하기 위한 장치 번호 n=1~256(16-bit 명령) n=1~128(32-bit 명령)

설명:

- S<sub>1</sub>** 에는 레지스터를 비교하여 얻은 값이 들어 있으며 **n** 에는 비교된 수가 들어 있습니다. 특성화된 데이터는 **S<sub>2</sub>** 에 명시된 데이터를 제외한 데이터를 비교한 것입니다. 그리고 비교된 결과값은 특성화된 몇몇의 레지스터 **D** 에 저장됩니다.
- 레지스터에서 32-bit 명령을 사용하여 지시할 때, **S<sub>1</sub>**, **S<sub>2</sub>**, **D** 그리고 **n** 특성화된 32-bit 레지스터를 사용합니다..
- SA 시리즈에서, 16-bit 와 32-bit 카운터는 **D** 연산자와 섞어 사용할 수 없습니다.

프로그램 예 :

- X0=On 일 때, D10~D19 으로 구성된 스택 데이터는 D0 를 제외하고 비교되며 D50~D54 에 그 결과값을 저장합니다. 만일 같은 값이 나타나지 않는다면, D50~D52 의 내용은 모두 0 이 되어 있을 것입니다.
- 데이터는 대수학 법칙에 의거해 비교됩니다. (-10<2)
- 모든 비교된 데이터 중 가장 큰 값은 D53 에 저장되며 가장 적은 값은 D54 에 저장될 것입니다. 제일 큰 값의 번호와 적은 값의 번호가 하나 이상의 값을 가지게 되면, 제일 큰 값이 저장될 것입니다.



D52	8	마지막 동일한 값의 번호입니다.
D53	7	제일 적은 값의 번호입니다.
D54	9	제일 큰 값의 번호입니다.

API	연상기호	연산자	기능	제어장치
62	D ABSD	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b> <b>(n)</b>	절대 Drum 순서기	ES EX SS SA SX SC EH

유형	비트 장치				워드 장치											프로그램 단계
연산자	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ABSD: 9 단계 DABSD: 17 단계
S <sub>1</sub>							*	*	*	*	*	*	*			
S <sub>2</sub>												*				
D		*	*	*												
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

연산자:

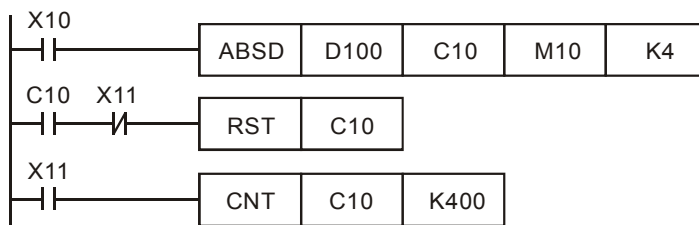
**S<sub>1</sub>**: 비교된 데이터 테이블의 장치 시작 **S<sub>2</sub>**: 카운터 번호 **D**: 비교된 결과값의 목적지의 시작 **n**: 여러 단계 그룹의 비교 (n=1~64)

설명:

- S<sub>1</sub>** 연산자가 KnX, KnY, KnM, 그리고 KnS 을 지시할 때, 16-bit 명령은 반드시 K4 를 지시하며 32-bit 명령은 반드시 K8 을 지시합니다.
- ABSD 은 일반적인 여러 pulse 출력에 대한 명령입니다. 이 명령에 의한 출력은 카운터의 현재 값과 일치하며 보통 캠 절대 제어에 사용됩니다.
- S<sub>2</sub>** 의 **DABSD** 명령은 특정 고속 카운터에서 사용할 수 있습니다. 하지만, 고속 카운터에 산출된 현재 값이 설정된 값과 비교되었을 때, 그 결과는 검색 시간에 영향을 받아 즉시 출력될 수 없습니다. 만일 즉시 출력되기 바란다면, 고속 카운터의 특수 비교 명령인 **DHSZ** 명령을 사용하길 권합니다.

프로그램 예 :

- ABSD 명령이 실행되기 전, MOV 명령을 사용하게 되면 각 D100~D107 이전에 설정된 값에 덮어 씁니다. 각 번호의 D 내용은 낮은 경계에 속한 값이며 D 에 추가된 번호는 높은 경계에 속한 값입니다.
- X10=On 일 때, C10 카운터의 현재 값은 D100~D107 위와 아래 범위를 제외한 4 개의 그룹과 비교됩니다. 비교된 결과 값은 M10~M13 에 나타납니다.
- X10=Off 일 때, M10~M13 의 원래 On/Off 상태는 변경될 수 없습니다.



- C10 의 현재 값이 낮은 경계의 값보다 같거나 동일하고 높은 경계의 값보다 낮거나 동일할 때 M10~M13 은 On 으로 될 것입니다.

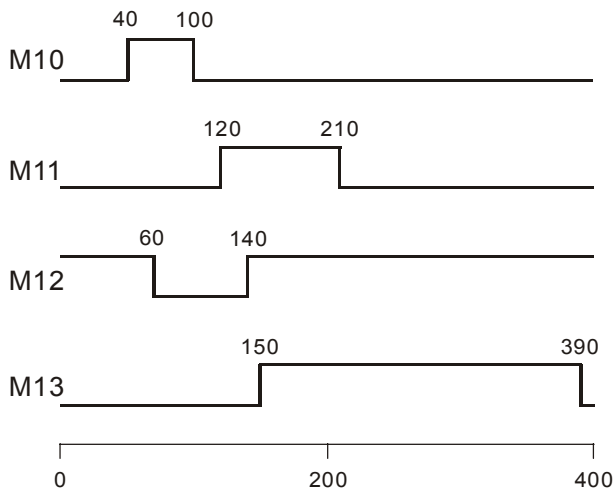
낮은 경계의 값	높은 경계의 값	C10 의 현재 값	출력
D100= 40	D101=100	$40 \leq C10 \leq 100$	M10=On
D102=120	D103=210	$120 \leq C10 \leq 210$	M11=On



D104=140	D105= 170	$140 \leq C10 \leq 170$	M12=On
D106=150	D107=390	$150 \leq C10 \leq 390$	M13=On

5. 낮은 경계의 값이 높은 경계의 값보다 높아지면, 높은 경계의 값이 될 것입니다. 만일 C10의 정확한 값이 낮은 경계의 값( $C10 < 140$ )보다 높거나 낮으면 높은 경계의 값은 ( $C10 > 140$ ), M12=On 이 됩니다.

낮은 경계 값	높은 경계 값	C10의 정확한 값	출력
D100= 40	D101=100	$40 \leq C10 \leq 100$	M10=On
D102=120	D103=210	$120 \leq C10 \leq 210$	M11=On
D104=140	D105= 60	$60 \leq C10 \leq 140$	M12=Off
D106=150	D107=390	$150 \leq C10 \leq 390$	M13=On



API	기호	연산자	기능	제어장치
63	INCD	<b>S<sub>1</sub></b> <b>S<sub>2</sub></b> <b>D</b> <b>n</b>	증분 드럼 순서기	ES EX SS SA SX SC EH

Type	비트장치				워드장치												프로그램 단계
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		INCD: 9 단계
S <sub>1</sub>							*	*	*	*	*	*	*				
S <sub>2</sub>												*					
D		*	*	*													
n					*	*											

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

연산자:

**S<sub>1</sub>**: 비교 데이터 테이블의 개시 장비    **S<sub>2</sub>**: 카운터 숫자    **D**: 비교된 결과들의 시작 번호

**n**: 다단계 비교 그룹 (**n**=1~64)

설명:

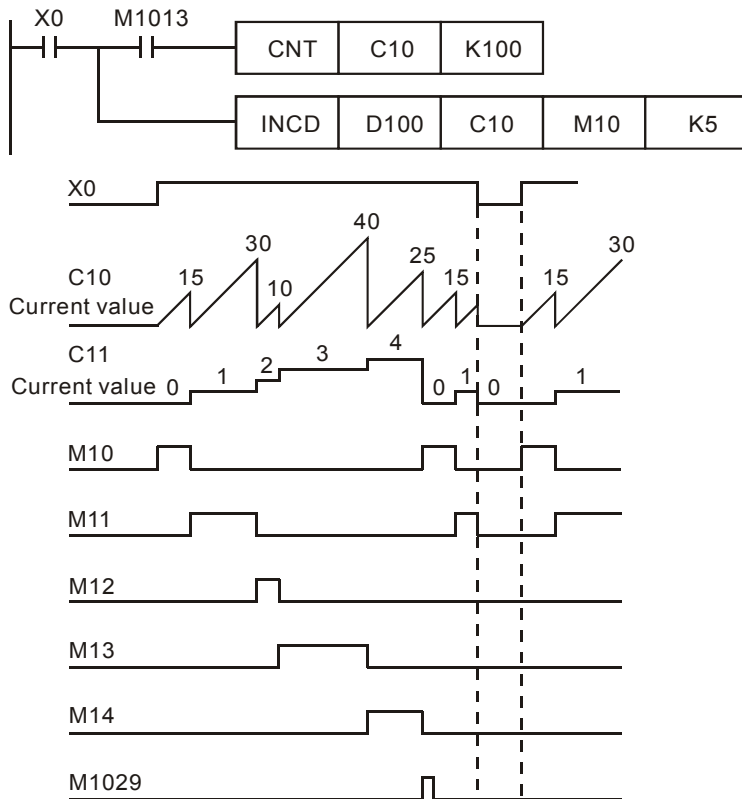
- 연산자 **S<sub>1</sub>**이 KnX, KnY, KnM, KnS 으로 특정될 때 K4 를 지칭해야 합니다. 16 비트 명령의 연산자 **S<sub>2</sub>** 는 C0~C198 을 지칭해야 합니다. (2 개 연속 카운터 점유)
- INCD 명령은 다수의 출력 펄스를 생성하는 카운터로 이루어지며 일반적으로 상대적 캠 제어에 사용됩니다.
- 전류값 **S<sub>2</sub>** 는 세팅값 **S<sub>1</sub>** 과 비교됩니다. 전류값이 세팅값과 같을 경우 전류값 **S<sub>2</sub>** 를 0 으로 리셋한후 다시

비교해야 합니다. 복귀시간은  $S_2+1$  에 저장됩니다.

4. n 집단 데이터 비교가 완료되면 1 사이클의 스캔에서 플래그 M1029의 실행이 완료될 것입니다.

프로그램 예시:

1. INCD 명령을 실행하기 전에 MOV 명령을 사용하여 각 세팅값을 D100~D104에 기록하십시오.  
D100=15, D101=30, D102=10, D103=40, D104=25.
2. 카운터 C10의 전류값이 D100~D104의 세팅값과 비교됩니다. 전류값과 세팅값이 동일할 경우 전류값 C10은 0으로 세팅된 후 다시 비교합니다.
3. 복귀시간은 C11에 저장됩니다.
4. C11의 내용이 증가하면 M10~M14역시 반응하여 변합니다. 다음의 타이밍 도식을 참조하십시오.
5. 5 집단 데이터 비교가 완료되면 1 사이클의 스캔에서 플래그 M1029의 실행이 완료될 것입니다.
6. X0이 '켜짐'에서 '꺼짐'으로 전환될 경우 C10과 C11은 0으로 리셋된 후 M10~M14는 꺼집니다. X0가 다시 '켜짐' 상태가 되면 이 명령은 처음부터 다시 실행됩니다.



API	기호	연산자	기능	제어장치						
64	TTMR	<b>D</b> <b>n</b>	대체 타이머	ES	EX	SS	SA	SX	SC	EH

Type OP	비트장치				워드장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	TTMR: 5 단계
D													*			
n					*	*										

연산자:

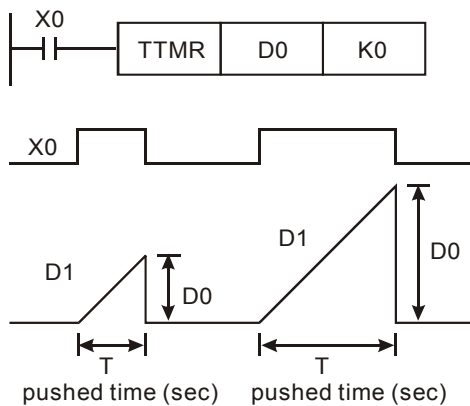
**D:** 입력에서 ‘켜짐’ 시간을 저장하는 장치 숫자(2 개의 연속장비 점유) **n:** 다수 세팅포인트 (n=0~2)

설명:

1. 프로그램 내에서 TTMR 명령은 8 회 까지 사용될 수 있습니다.
2. 외부 버튼스위치의 ‘켜짐’ 기간은 **D+1** 에 측정 및 저장되며 측정 유닛은 100ms 주기입니다. 수초간의 **D+1** 내용값은 **n** 배수 되어 **D** 에 저장됩니다
3. 배수 세팅이 n=0 일때 **D** 측정 유닛은 수초간 지속됩니다. n=1 일때 **D** 측정 유닛은 100ms(10 배수됨) 주기입니다. n=2 일때 측정유닛 **D** 는 10ms 주기입니다(100 배수됨).

프로그램 예시 1:

1. X10 버튼 스위치를 누른 시간(X0 인 기간에서)은 D1 에 저장되며, n 은 곱해진 시간을 특정하는데 쓰이며 전체 비트 시간은 D0 에 저장됩니다. 그런다음 버튼 스위치는 카운터의 세팅값을 수정하는데 사용할 수 있습니다.
2. X0 이 ‘거짐’ 일때 D1 은 0 으로 리셋되지만 D0 은 불변입니다.



3. X0 기간이 T 초간 지속될 때 D0 과 D1 의 관계와 n 은 아래의 표와 같이 나타납니다.

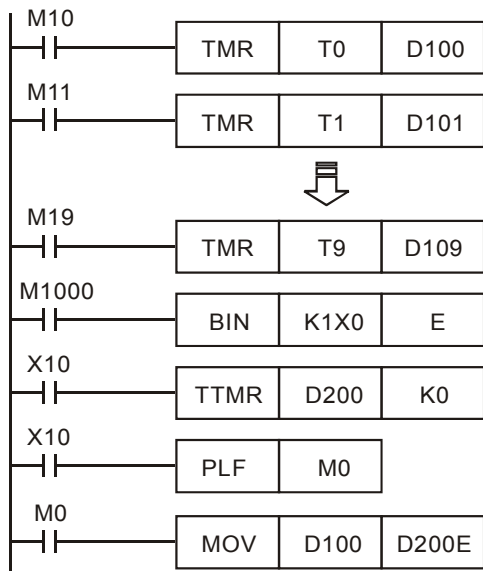
n	D0	D1(unit: 100 ms)
K0 (unit: s)	1×T	D1=D0×10
K1 (unit: 100 ms)	10×T	D1=D0
K2 (unit: 10 ms)	100×T	D1=D0/10

프로그램 예시 2:

1. TMR 명령을 사용하여 10 그룹 세팅 시간을 기록합니다.
2. D100~D109 에 세팅값을 미리 써넣습니다.
3. 다음의 카운터 T0~T9 의 측정된 유닛은 0.1 초이며 교체의 측정된 유닛은 1 초 입니다.
4. 1 비트 디지털 스위치를 X0~X3 에 연결하고 BIN 명령을 사용하여 디지털 스위치의 세팅 값을 BIN 값으로

전환하고 E 에 저장하십시오.

5. X10 의 ‘켜짐’ 기간(초당)은 D200 에 저장됩니다.
6. M0 은 대체카운터 버튼이 개방되었을 때 한 회기 스캔싸이클이 생성하는 펄스입니다.
7. 디지털 스위치의 세팅숫자를 인덱스 레지스터의 포인터로 이용하고, 그다음 D200 의 내용을 D100E 로 전송합니다(D100~D109).



주의:

1. SA/SX/SC 시리즈 모델은 TTMR 명령을 8 회만 사용할 수 있습니다. 서브루틴이나 간접 서브루틴에서 사용할 경우 오직 ‘1 회’ 사용만 가능합니다.
2. EH 시리즈 모델의 경우 최대 TTMR 명령 집단이 오직 동시에 사용할 수 있는 것은 8 개 집단입니다.

API	기호	연산자	기능	제어장치						
65	STMR	<b>S</b> <b>m</b> <b>D</b>	특수 카운터	ES	EX	SS	SA	SX	SC	EH

Type OP	비트장치				워드장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					STMR: 7 단계
m					*	*										
D		*	*	*												

**연산자:**

**S:** 카운터 숫자( SA 시리즈 모델은 T0~T191 을 사용할 수 있음. EH 시리즈 모델은 T0~T199 을 사용할 수 있음)

**m:** 카운터의 세트포인트 값(**m**=1~32,767), 100ms 의 유닛

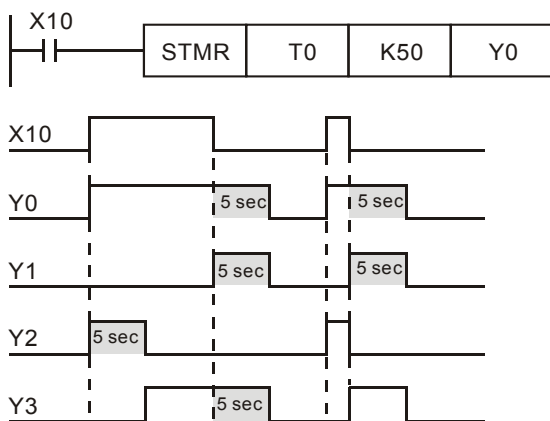
**D:** 출력의 시작장치(4 개의 연속장치 점유)

**설명:**

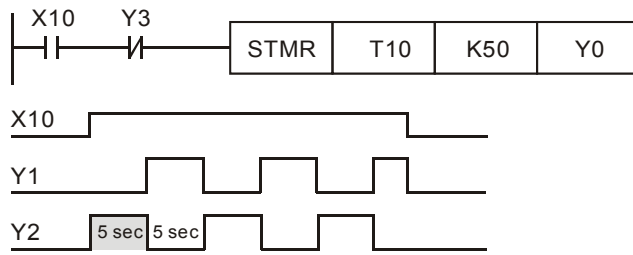
1. STMR 명령은 '꺼짐 지연' 의 명령으로 원샷, 플래쉬 루프(loop)입니다.
2. STMR 명령으로 특정된 타이머 숫자는 반복될 수 없습니다.


**프로그램 예시:**

1. X10=켜짐 일때 STMR 명령으로 특정되는 타이머 T0 의 세팅값은 5 초 입니다.
2. Y0 이 '꺼짐'지연의 연결일 때: X10 이 '꺼짐'에서 '켜짐'으로 바뀔경우, Y0=켜짐. X10 이 '켜짐'에서 '꺼짐'으로 바뀔 때와 지연이 5 초일 때 Y0=꺼짐
3. X10 이 '켜짐'에서 '꺼짐'으로 바뀔경우, Y1=켜짐 출력은 5 초간 한번 이루어집니다.
4. X10 이 '꺼짐'에서 '켜짐'으로 바뀔경우, Y2=켜짐 출력은 5 초간 한전 이루어집니다.
5. X10 이 거짐에서 '켜짐'으로 바뀔경우, 5 초의 지연 뒤 Y3=켜짐 으로 됩니다. X10 '켜짐'에서 '꺼짐'으로 바뀔경우 5 초 지연 뒤 Y3=꺼짐으로 됩니다.



6. 드라이브가 X10 을 접촉한 후 Y3 에 접촉을 더한 다음 Y1, Y2 는 출력 플래쉬 루프로 이용할 수 있습니다. X10 이 '꺼짐'으로 될 경우 Y0, Y1, Y3 는 '꺼짐'이 되고 T10 의 내용은 0 으로 리셋됩니다.



API	기호			연산자	기능						제어장치						
	66		ALT	P		Alternate ON/OFF						ES	EX	SS	SA	SX	SC

Type OP	비트장치				워드장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ALT, ALTP: 3 단계
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

연산자:

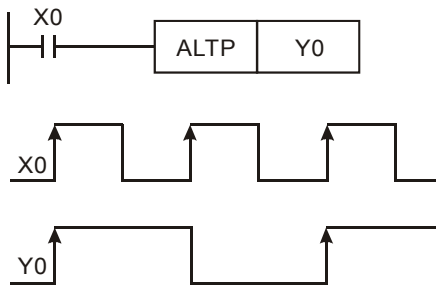
**D**: 위치 장치

설명:

이 장치는 일반적으로 펄스 수행 명령임. (ALTP).

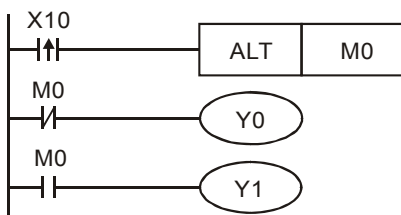
프로그램 예시 1:

X0 이 첫 번째로 '꺼짐'에서 '켜짐'으로 변할 때 Y0=켜짐 입니다. X0 이 '꺼짐'에서 '켜짐'으로 두 번째 변할 때 Y0=꺼짐 입니다.



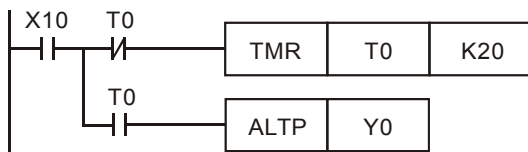
프로그램 예시 2:

ALT 는 시작/정지 모드를 제어할 때 1 개 스위치를 사용하는 명령입니다. 시작점에서 M0=꺼짐 이므로 Y0=켜짐, Y1=꺼짐 입니다. X10 이 첫 번째로 작동할 경우 M0=켜짐, Y1=켜짐, Y0=꺼짐 입니다. X10 이 두 번째로 작동할 때 M0=꺼짐, Y0=켜짐, Y1=꺼짐.



프로그램 예시 3:

출력 Y0 은 플레쉬. X10=켜짐 일 때, T0 는 매 2 초마다 펄스를 생성하며 출력 Y0 은 켜짐/꺼짐 사이에서 T0 의 펄스 변화에 따라 변환하게 됩니다.



API	기호	연산자	기능	제어장치						
67	RAMP	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b> <b>(n)</b>	상이한 Ramp 값	ES	EX	SS	SA	SX	SC	EH

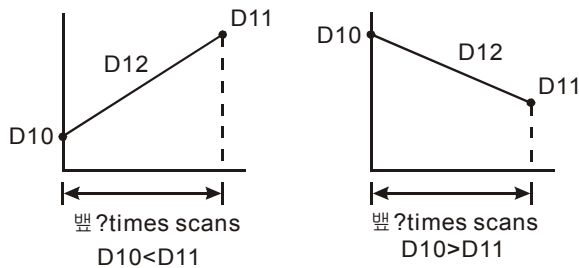
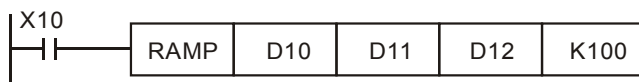
Type OP	비트장치				워드장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	RAMP: 9 단계
S <sub>1</sub>													*			
S <sub>2</sub>													*			
D													*			
n					*	*										

**연산자:**

**S<sub>1</sub>:** 램프신호 세팅포인트 시작 **S<sub>2</sub>:** 램프신호 세팅포인트 종료 **D:** 램프신호 현재시간 값(2 개 연속장치 점유)  
**n:** 스캔 시간(n=1~32,767)

**설명:**

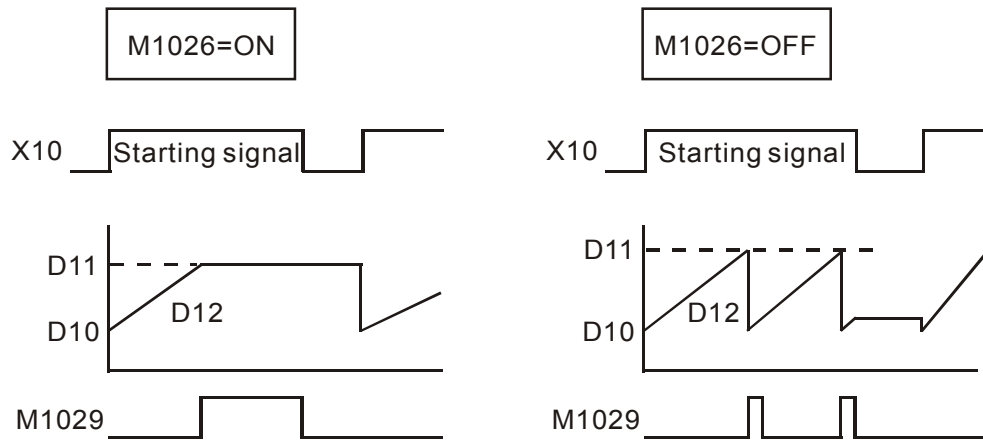
- 이 명령은 램프신호를 얻기위해 사용됩니다. 램프신호는 리니어와 스캔 시간과 연관성이 큼니다. 그러므로 RAMP 명령을 실행하기 전 스캔시간을 반드시 고정해야 합니다.
- D10 과 램프신호 종료 세팅값을 D11 로 바꾸기 위해 시작 세팅값을 미리 기록해야 합니다. X0 이 '켜짐'일 때 세팅값은 D10 에서 D11 로 전환되고(D10 의 세팅값 증가) 진행 시간은(n=100 회 스캔) D12 에 저장됩니다.
- 프로그램상에서 세팅 M1039=켜짐 일 경우 스캔시간은 고정될 수 있습니다. 그런 다음 MOV 스캔을 사용하여 고정된 스캔시간의 세팅값을 특수 레지스터 D1039 에 기록합니다. 위의 프로그램을 예로 들면, 세팅값이 30ms 이고 n=K100 이면 D10 과 D11 사이는 3 초 입니다(D3:30msx100)
- 이 명령의 실행 중 X10 시작신호가 '켜짐' 으로 될 경우, 이 명령은 작동을 멈출것입니다. X10 이 다시 '켜짐' 이 되면 D12 의 내용값은 0 으로 리셋되고 다시 계산됩니다.
- 이 명령이 완료된 후 M1029=켜짐 과 D12 의 내용값은 세팅값 D10 으로 리셋됩니다.
- 이 명령과 아날로그 신호 출력을 함께 사용하면 시작/정지 정렬 작업을 실행할 수 있습니다.
- X10=켜짐 일 때 PLC 를 '정지' 에서 '운전' 으로 시작할 경우, 프로그램 시작점에서 D12 의 내용값을 0 으로 리셋하십시오. (D12 가 래치-latch 영역일 경우)



**주의:**

시작 모드 flag M1026 의 켜짐/꺼짐 상태와 D12 의 세팅값 변화는 아래와 같음:





API	기호	연산자	기능	제어장치						
69	SORT	<b>(S)</b> <b>(m<sub>1</sub>)</b> <b>(m<sub>2</sub>)</b> <b>(D)</b> <b>(n)</b>	Data sort	ES	EX	SS	SA	SX	SC	EH

Type	비트장치				워드장치											프로그램 단계				
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SORT: 11 단계				
S													*							
m <sub>1</sub>					*	*														
m <sub>2</sub>					*	*														
D													*							
n					*	*							*							

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

연산자:

**S**: 소스 데이터 표의 시작장치 **m<sub>1</sub>**: 데이터 분류 그룹 (**m<sub>1</sub>**=1~32) **m<sub>2</sub>**: 각 데이터의 세로 칸 번호 (**m<sub>2</sub>**=1~6)

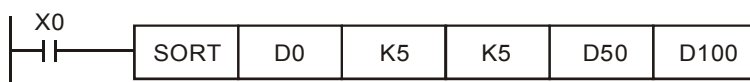
**D**: 분류 데이터의 시작장치 **n**: 분류 데이터의 참조 값 (**n**=1~**m<sub>2</sub>**)

설명:

- 결과 분류 데이터는 **D** 에 의해 특정된 시작장치에서 카운트된 **m<sub>1</sub> x m<sub>2</sub>** 레지스터에 저장됩니다. 그러므로 **S** 와 **D** 장치가 레지스터에 의해 특정되면 결과 분류 데이터는 **S** 장치의 소스 내용과 같을 것입니다.
- S** 에 의해 특정된 이상적인 선행 숫자는 0 입니다.
- SORT** 명령이 **m<sub>1</sub>** 시간동안 진행될 경우 데이터 분류가 완료됩니다. **SORT** 명령이 완료되면 Flag M1029=켜짐 입니다.

프로그램 예시:

X0 이 '켜짐' 이면, 특정 데이터를 분류하기 시작합니다. 데이터 분류가 완료되면 M1029=켜짐 입니다. **SORT** 명령 실행시 분류 데이터를 변경하지 마십시오. 데이터 분류를 다시 하고자 한다면 X0 이 '꺼짐' 에서 '켜짐'으로 바뀌어야 함을 반드시 명심해 주십시오.



EXAMPLE TABLE OF DATA SORT

		Data numbers: m <sub>2</sub>				
		Data Column				
Column		1	2	3	4	5
Row		Students No.	Chinese	English	Mathematics	Physis and Chemistry
Data numbers: m <sub>1</sub>	1	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
	2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
	3	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90
	4	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
	5	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69

D100=K3 일 때 데이터 분류표.

		<div>← Data numbers: m<sub>2</sub> →</div>				
		Data Column				
<div>Column Row</div>	1	2	3	4	5	
	Students No.	Chinese	English	Mathematics	Physis and Chemistry	
1	( D50 ) 4	( D55 ) 70	( D60 ) 60	( D65 ) 99	( D70 ) 50	
2	( D51 ) 2	( D56 ) 55	( D61 ) 65	( D66 ) 54	( D71 ) 63	
3	( D52 ) 1	( D57 ) 90	( D62 ) 75	( D67 ) 66	( D72 ) 79	
4	( D53 ) 5	( D58 ) 95	( D63 ) 79	( D68 ) 75	( D73 ) 69	
5	( D54 ) 3	( D59 ) 80	( D64 ) 98	( D69 ) 89	( D74 ) 90	

Sort data table when D100=K5.

		Data numbers: $m_2$				
		Data Column				
Column Row	1	2	3	4	5	
	Students No.	Chinese	English	Mathematics	Physis and Chemistry	
1	( D50 ) 4	( D55 ) 70	( D60 ) 60	( D65 ) 99	( D70 ) 50	
2	( D51 ) 2	( D56 ) 55	( D61 ) 65	( D66 ) 54	( D71 ) 63	
3	( D52 ) 5	( D57 ) 95	( D62 ) 79	( D67 ) 75	( D72 ) 69	
4	( D53 ) 1	( D58 ) 90	( D63 ) 75	( D68 ) 66	( D73 ) 79	
5	( D54 ) 3	( D59 ) 80	( D64 ) 98	( D69 ) 89	( D74 ) 90	

API	기호		연산자			기능			제어장치						
70	D	TKY	<b>S</b>	<b>D<sub>1</sub></b>	<b>D<sub>2</sub></b>	10 개 키 입력			ES	EX	SS	SA	SX	SC	EH

Type	비트장치				워드장치										프로그램 단계	
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												TKY: 7 단계
D <sub>1</sub>								*	*	*	*	*	*	*	*	DTKY: 13 단계
D <sub>2</sub>		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

연산자:

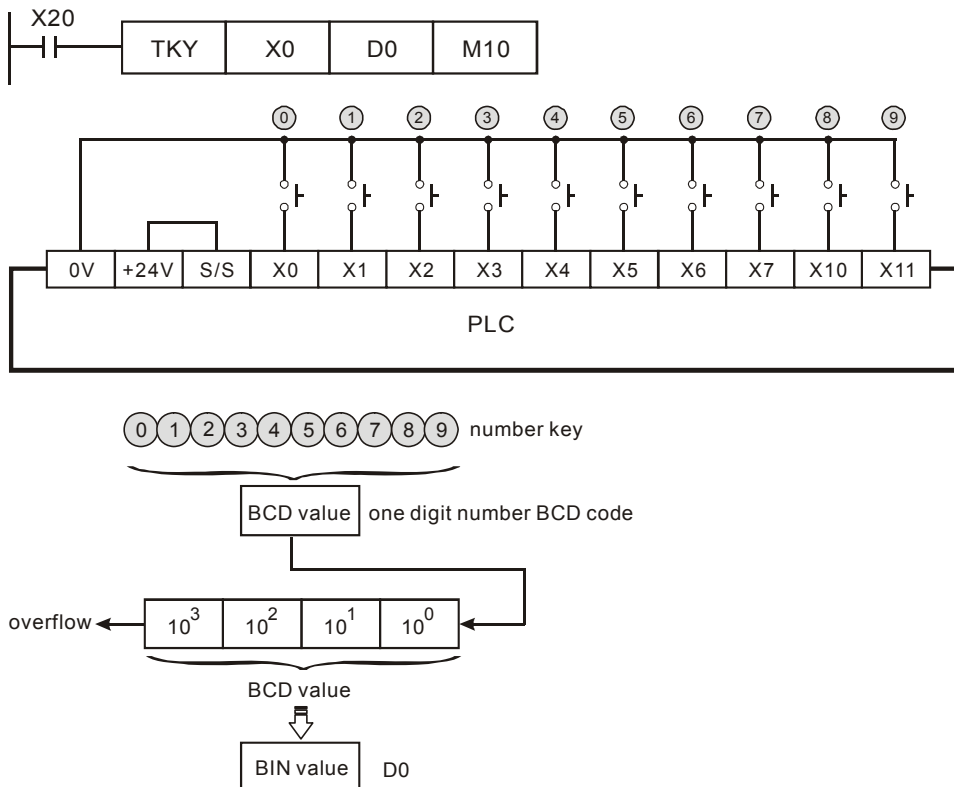
**S**: 입력장치 시작 (10 개의 연속장치 점유) **D<sub>1</sub>**: 저장키 입력 값 위치 **D<sub>2</sub>**: 입력 키 신호(10 개의 연속장치 점유)

설명:

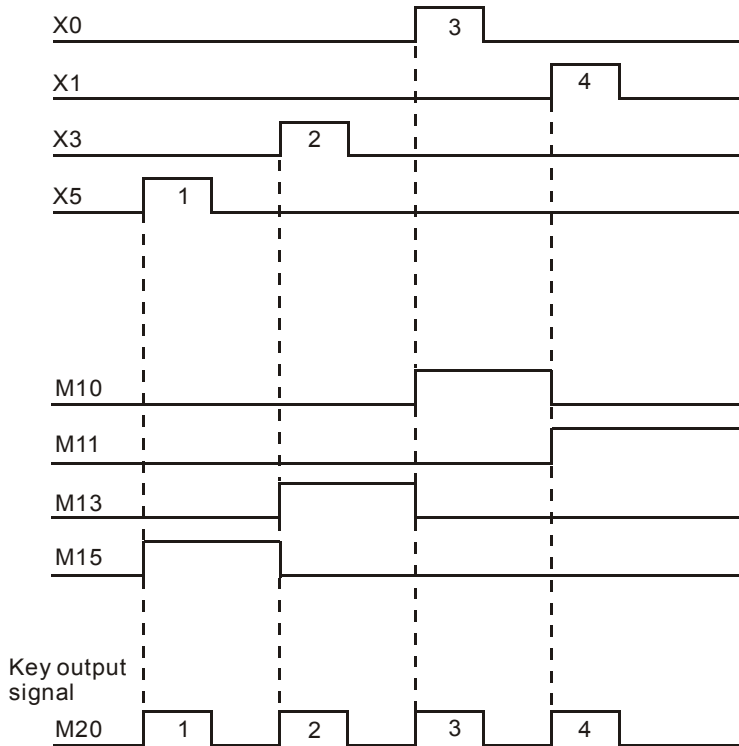
- SA 시리즈 모델은 연산자 **S** 와 **D<sub>2</sub>** 가 인덱스 레지스터 E, F 를 지원하지 않습니다.
- 이 명령은 **S** 로부터 외부입력 장치 10 개를 특정할 수 있으며, 이 10 개의 외부입력 장치는 십진수 값 0 에서 9 까지로 식별됩니다. 이들 10 개 외부입력 장치는 각각 10 개의 키와 연결되어 있습니다. 10 개의 외부입력 장치중 1 개의 키를 누를 경우 십진수 숫자 값 0 에서 9,999(최대 4 자리 16-bit 명령) 또는 0 에서 99,999,999(최대 8 자리 32-bit 명령)이 장치 **D<sub>1</sub>** 에 입력되고 저장될 수 있습니다. 장치 **D<sub>2</sub>** 는 누른 키의 상태를 저장하는데 사용됩니다.

프로그램 예시 :

- 이 명령은 '**X0**' 에서 '10 개 키(0~9 까지 숫자인)'까지 10 개의 입력 터미널을 특정할 수 있습니다. **X20**=켜짐 일 때 위 명령이 실행되며, **D0** 에 키로 입력된 BIN 값을 저장함과 동시에 **M10~M19** 를 누른 키의 상태 저장에 사용합니다.



2. 시간표가 아래와 같이 나타나있듯이 4 개 키가 숫자 키보드 X5,X3,X0,X1 와 연결되어 있습니다. 4 개의 키를 ①②③④의 순서로 누른 후 숫자 5,301 이 D0 에 입력됩니다. D0 에 들어갈 수 있는 최대숫자는 9,999 즉 4 자리 숫자 입니다. 만약 입력된 숫자가 위의 허용범위를 초과할 경우 최고높은 숫자는 범람하게 될 것입니다.
3. X2 를 누르면 다른 키를 누를 때 까지 M12=켜짐이 됩니다. 누른 키의 상태는 여타와 같습니다.
4. X0~X11 중 아무키나 누를 경우 M10~M19 의 한 장치가 '켜짐' 상태가 됩니다.
5. 아무키나 누를 경우 M20=켜짐.
6. 드라이브 접속 X20 이 '꺼짐' 일 경우, 기존의 값은 변하지 않으나 M10~M20 은 '꺼짐' 으로 바뀜.



API	기호		연산자				기능				제어장치							
71	D	HKY	<b>S</b>	<b>D<sub>1</sub></b>	<b>D<sub>2</sub></b>	<b>D<sub>3</sub></b>	16 진수 키 입력				ES	EX	SS	SA	SX	SC	EH	

Type  OP	비트장치				워드장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	HKY: 9 단계  DHKY: 17 단계	
S	*																
D <sub>1</sub>		*															
D <sub>2</sub>											*	*	*	*	*		
D <sub>3</sub>		*	*	*													

PULSE								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

**연산자:**

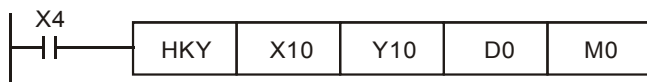
**S:** 입력장치 스캔 시작 (4 개 연속장치 점유)    **D<sub>1</sub>:** 출력장치 스캔 시작(4 개 연속장치 점유)    **D<sub>2</sub>:** 저장 키 입력 값 위치    **D<sub>3</sub>:** 키 입력 신호(8 개 연속장치 점유)

**설명:**

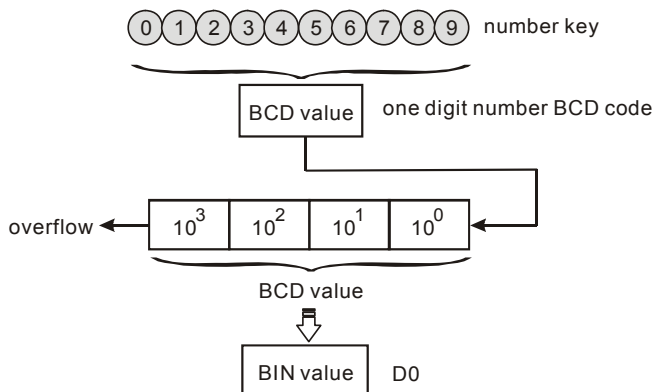
- 이 명령은 S로부터 4 개의 연속외부입력장치, D<sub>1</sub>로부터 4 개의 연속외부출력장치를 매트릭스 스캔한 복합체인 16 키 키보드 명령을 생성할 수 있습니다. 키 입력값은 D<sub>2</sub>에 저장되고 D<sub>3</sub>는 입력한 키의 상태를 저장하는데 사용됩니다.
- 이 명령이 매회 실행될 때 마다 키를 누른 시간동안 실행이 완료된 flag M1029 는 ‘켜짐’ 상태가 됩니다(1 스캔 회기).
- 두개 이상의 키가 동시에 눌러질 경우 첫번째 눌린 키만 작동하게 됩니다.
- HKY 명령이 16-bit 명령으로 쓰인다면 D<sub>2</sub> 는 0~9,999(최대 4 자리)까지 기록할 수 있습니다. DHKY 명령이 32-bit 명령으로 쓰일경우, D<sub>2</sub> 는 0~99,999,999(최대 8 자리)까지 기록할 수 있습니다. 만약 입력된 숫자가 위의 허용범위를 초과할 경우 최고높은 숫자는 범람하게 될 것입니다.

**프로그램 예시 :**

- 이 명령은 X10~X13로부터 4 개의 연속외부입력장치, Y10~Y13로부터 4 개의 연속외부출력장치 복합체인 16 키 키보드 명령을 생성할 수 있습니다. X4=켜짐 일 때 위 명령이 실행되며, D0에 키로 입력된 BIN 값을 저장함과 동시에 M0~M7 를 누른 키의 상태 저장에 사용합니다.

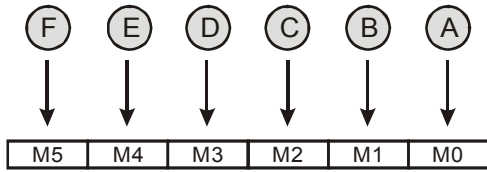


- 숫자 입력 :



- 기능 키 입력:

- a) A 키를 누르면 M0=켜짐 이고 래치 상태가 됩니다. 그 다음 D 키를 누르고, 그 후 M0=꺼짐, M3=켜짐 이 됩니다.
- b) 두개 이상의 키가 동시에 눌러질 경우 첫번째 눌린 키만 작동하게 됩니다.

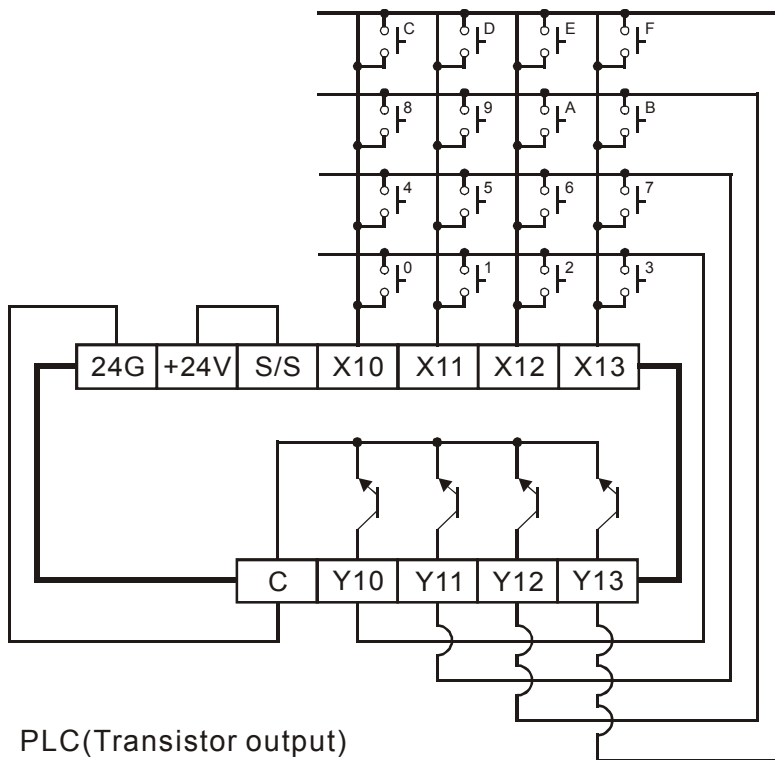


4. 키 출력 신호:

- a) A 에서 F 키까지 누르면 한번 M6=켜짐 이 됩니다.
- b) 0 에서 9 키까지 누르면 한번 M7=켜짐 이 됩니다.

5. 드라이브 접촉 X4 가 꺼짐이 될 때, 이전 값은 변하지 않으나 M0~M7 은 모두 '꺼짐' 이 됩니다.

6. 외부 배선:



주의:

1. 이 명령이 실행될 경우 키 입력값을 정확히 읽기 위해 8 번의 스캔주기가 요청됩니다. 스캔 주기가 너무 길거나 짧으면 키 입력값을 부정확하게 읽는 원인이 될 수 있습니다. 그러므로 사용자는 다음에 유의하여 문제를 피하시기 바랍니다.
  - a) 스캔 주기가 너무 짧다면 I/O 반응이 제때에 이루어지지 않고 키 입력값을 정확히 읽지 못하게 됩니다. 이 경우 사용자는 스캔시간을 수정하여 문제를 해결할 수 있습니다.
  - b) 스캔 주기가 너무 길다면 키 입력에 대한 반응이 느려질 수 있습니다. 사용자는 이 명령을 서브루트가 간섭되는 때에 입력하거나, 고정된 시간에 이 명령을 수행함으로써 문제를 해결할 수 있습니다.
2. flag M1167 의 기능:
  - a) M1167=켜짐 일 때, HKY 명령은 0~F 까지의 16 진수 값을 입력할 수 있습니다.
  - b) M1167=꺼짐 일 때, HKY 명령의 A~F 는 기능키로 사용할 수 있습니다.

API	기호	연산자				기능		제어장치						
72	DSW	<div>S</div>	<div>D<sub>1</sub></div>	<div>D<sub>2</sub></div>	<div>n</div>	디지털 스위치		ES	EX	SS	SA	SX	SC	EH

Type OP	비트장치				워드장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DSW: 9 단계
S	*															
D <sub>1</sub>		*														
D <sub>2</sub>											*	*	*			
n					*	*										

연산자:

**S:** 스위치 입력의 시작장치 **D<sub>1</sub>:** 스위치 출력의 시작장치 **D<sub>2</sub>:** 세트 포인트 값 저장 위치 장치

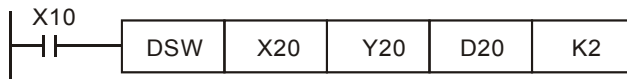
**n:** 자릿수 (n=1~2)

설명:

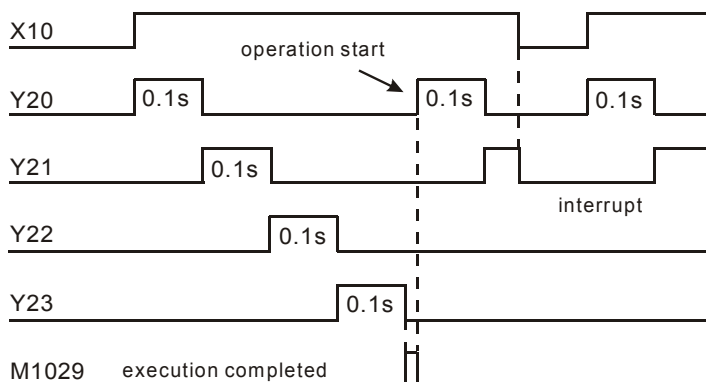
1. SA 시리즈 모델은 연산자 S와 D<sub>1</sub> 이 인덱스 레지스터 E, F 를 지원하지 않습니다.
2. 이 명령은 D<sub>1</sub>의 4 연속 외부장치와 S 의 4 또는 8 연속 외부입력장치를 1 또는 2 개 집단으로 이루어진 4 자리 스위치를 통해 읽기 위한 것이며, 세팅값을 위치 장치인 D<sub>2</sub>에 저장하기 위한 것입니다. n 이 1 일 경우 오직 한 집단의 디지털 스위치만 읽을 수 있습니다. n 이 2 일 경우 2 개 집단의 디지털 스위치를 읽을 수 있습니다.

프로그램 예시 :

1. 첫 번째 집단의 스위치는 X20~X23 와 Y20~Y23 으로 이루어져 있습니다. 두 번째 집단의 스위치는 X24~X27 와 Y20~Y23 으로 이루어져 있습니다. X10=켜짐일 경우, 이 명령은 실행 시작합니다. 첫 번째 집단 스위치의 세팅 값이 읽힌다음 BIN 값으로 변환된 후 D20 에 저장됩니다. 두 번째 집단의 세팅값도 읽힌 후 BIN 값으로 변환된 후 D21 에 저장됩니다.

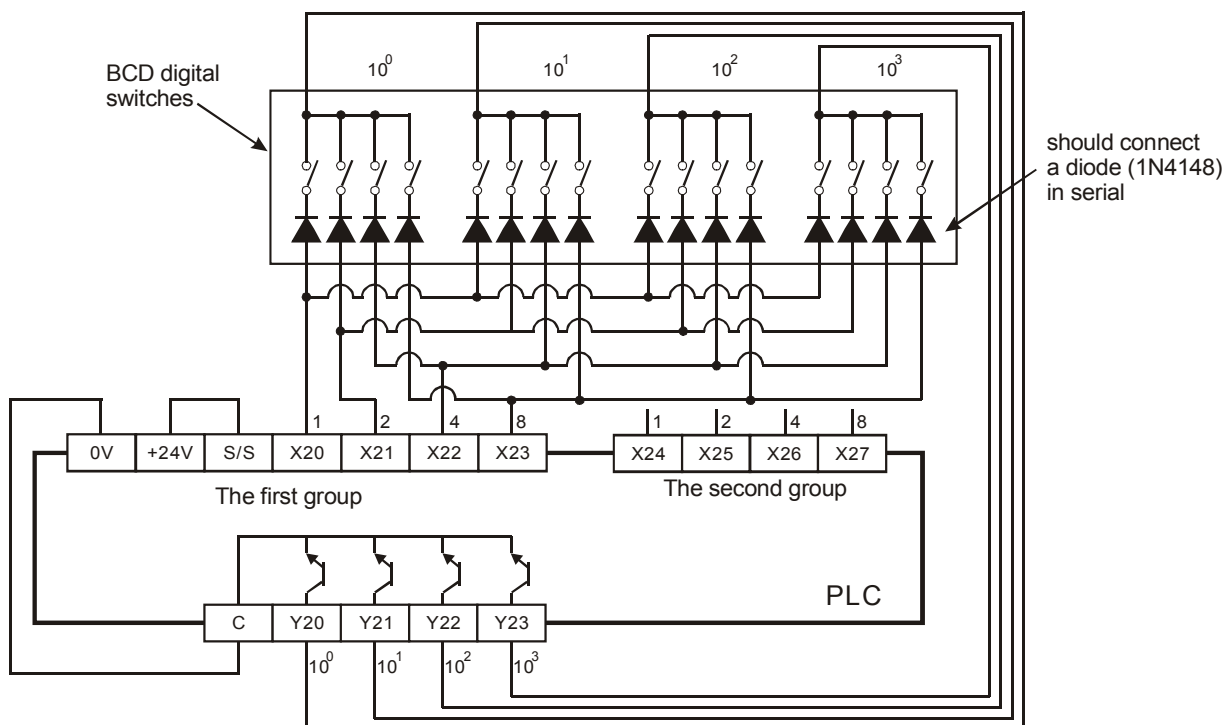


2. X10=켜짐 일 때 Y20~Y23 은 ‘켜짐’ 이고 순환 스캔이 자동 시작됩니다. 각 순환스캔이 완료된 후 실행완료된 flag M1029=켜짐 은 1 회 순환스캔 이후의 스캔 주기를 나타냅니다.
3. 출력 Y20~Y23 트랜지스터 출력을 사용하시기 바랍니다. 그 외에는 매 1,2,4,8, 터미널은 아래의 시리얼 예와 같이 PLC 의 입력에 다이오드를 연결해야 합니다(0.1A/50V).



디지털 스위치 배선도

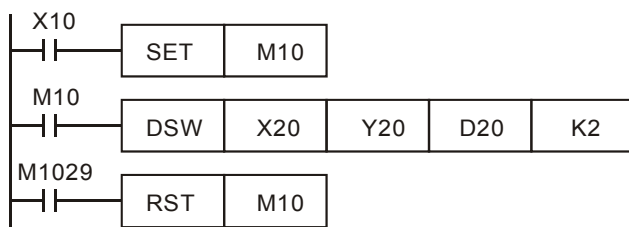




### 주의:

스캔 터미널이 출력을 중계한다면, 다음과 같은 프로그램 기능을 본 명령과 함께 사용하여 작동을 성공적으로 이끌 수 있습니다:

- X10=켜짐 이면, DSW 명령이 실행됩니다. X10 이 '꺼짐'으로 바뀌면, M10 은 DSW 명령이 1 주기 출력 스캔을 마칠 때까지 '켜짐' 이 됩니다. 그 다음 M10 은 '꺼짐' 이 됩니다.
- X10 드라이브 접촉이 버튼 스위치를 사용한다면, X10 을 누를 때 마다 DSW 명령으로 특정된 '스캔터미널 M10' 이 출력스캔 1 주기가 완료된 후 '꺼짐'으로 리셋될 것입니다. 다음, 그 명령은 실행을 정지한 후 디지털 스위치 데이터가 완전히 읽히고 버튼을 누르고 있는 동안 스캔터미널은 작동할 것입니다. 그러므로 이 상황에서 중계 출력이 사용되더라도 그 사용이 많지 않으므로 오랜 기간 중계기 사용이 가능합니다.



API	기호		연산자	기능	제어장치						
73	SEGD	P	<b>S</b> <b>D</b>	7-분할 디코더	ES	EX	SS	SA	SX	SC	EH

Type OP	비트장치				워드장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SEGD, SEGDP: 5 단계
S					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

설명:

**S**: 디코딩을 위한 원천장치 **D**: 디코딩 후 출력장치

프로그램 예시 :

X10=켜짐 일 때, D10 의 하급 4 bits (b0~b3)의 내용(0~F: 16 bits)은 7 분할로 디코딩 되어 디스플레이 창에 출력될 수 있습니다. 디코딩 결과는 10~Y17 에 저장됩니다.



Decoding Chart of the 7-segment Display Panel:

16 bits	Bit Combination	Composition of the 7-Step Display Panel	Status of Every Step							Data Displayed
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)	
0	0000		ON	ON	ON	ON	ON	ON	OFF	0
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	1
2	0010		ON	ON	OFF	ON	ON	OFF	ON	2
3	0011		ON	ON	ON	ON	OFF	OFF	ON	3
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	4
5	0101		ON	OFF	ON	ON	OFF	ON	ON	5
6	0110		ON	OFF	ON	ON	ON	ON	ON	6
7	0111		ON	ON	ON	OFF	OFF	ON	OFF	7
8	1000		ON	ON	ON	ON	ON	ON	ON	8
9	1001		ON	ON	ON	ON	OFF	ON	ON	9
A	1010		ON	ON	ON	OFF	ON	ON	ON	A
B	1011		OFF	OFF	ON	ON	ON	ON	ON	b
C	1100		ON	OFF	OFF	ON	ON	ON	OFF	c
D	1101		OFF	ON	ON	ON	ON	OFF	ON	d
E	1110		ON	OFF	OFF	ON	ON	ON	ON	e
F	1111		ON	OFF	OFF	OFF	ON	ON	ON	f

API	기호	연산자	기능	제어장치						
74	SEGL	<b>S</b> <b>D</b> <b>n</b>	래치 와 7-분할	ES	EX	SS	SA	SX	SC	EH

Type OP	비트장치				워드장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SEGL: 7 단계
D		*														
n					*	*										

연산자:

**S**: 7-분할 디스플레이 소스장치 **D**: 7-분할 디스플레이 스캔 출력 시작 장치

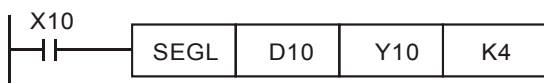
**n**: 스캔 신호와 출력신호의 극성 세트포인트 (**n**=0~7)

설명:

1. EH 시리즈 모델은 프로그램상에서 SEGL 명령이 2 회 까지 쓰일 수 있습니다
2. ES/SA 시리즈 모델에서 D 연산자에 가장 적합한 숫자는 0(zero) 이며 인덱스 레지스터 E,F 는 지원되지 않습니다.
3. 이 명령 D 에서 시작된 8 또는 12 의 연속 외부출력 포인트는 4 자리 숫자인 7 분할 디스플레이의 1 또는 2 개 그룹의 디스플레이 및 스캔 시그널로 취급될 수 있습니다. 7 분할 디스플레이 모듈은 BCD 코드를 7 분할로 변환하는 기능과 제어 신호의 래치 여부결정 기능을 가지고 있습니다.
4. n 은 4 자리 숫자인 7 분할 디스플레이의 숫자를 결정할 수 있고 PLC 출력 터미널의 극성과 입력 터미널의 7 분할 디스플레이를 나타냅니다.
5. 4 자리 숫자집단의 7 분할 디스플레이 출력 명령의 숫자 포인트는 8 이며 4 자리 숫자 2 개 집단의 포인트는 12 입니다.
6. 스캔출력 터미널은 명령 실행시 순서에 따라 순환할것입니다. 드라이브 접촉은 '꺼짐' 에서 '켜짐' 으로 바뀌고 스캔 출력은 다시한번 실행됩니다.

프로그램 예시 :

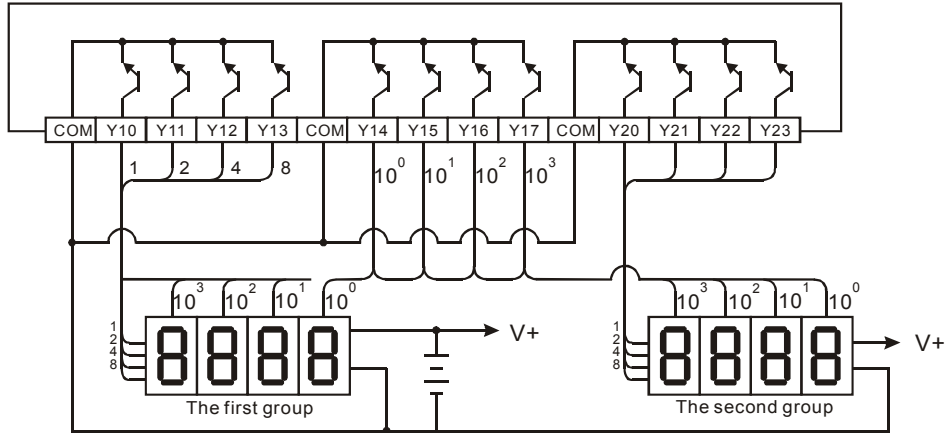
1. X10=켜짐 일 때, 명령이 실행됩니다. 7-디스플레이 스캔루프는 Y10~Y17 로 이루어집니다. D10 의 값은 BCD 코드로 전환된 후 7 분할 디스플레이의 첫번째 그룹으로 보내져 디스플레이 패널에 표시됩니다. D11 의 값은 BCD 코드로 전환된 후 7 분할 디스플레이의 두 번째 그룹으로 보내져서 디스플레이 패널에 표시됩니다. D10 과 D11 의 값이 9999 보다 크면 작업 에러가 생기게 됩니다.



2. X10=켜짐 이면, Y14~Y17 는 자동으로 순환 스캔을 실시합니다. 각 순환스캔마다 12 스캔타임이 소요됩니다. M1029=켜짐 은 1 회 순환스캔 후 스캔 시간입니다.
3. 1 개 집단의 4 자리 숫자, n=0~3.
  - a) 디코딩 된 7 분할 디스플레이의 1,2,4,8 이 자체적으로 병렬 연결될 때, PLC 의 Y10~Y13 에 연결되어야 합니다. 각 번호의 래치 터미널은 PLC 의 Y14~Y17 로 각기 연결됩니다.
  - b) X10=켜짐 이면 D10 의 내용은 7-분할 디스플레이로 전송되어 순차적으로 순환하는 Y14~Y17 에 대응하여 표시되게 됩니다.
4. 2 개 집단의 4 자리 숫자, n=4~7.
  - a) 디코딩 된 7 분할 디스플레이의 1,2,4,8 이 자체적으로 병렬 연결될 때, PLC 의 Y20~Y23 에 연결되어야 합니다. 각 번호의 래치 터미널은 PLC 의 Y14~Y17 로 각기 연결됩니다.

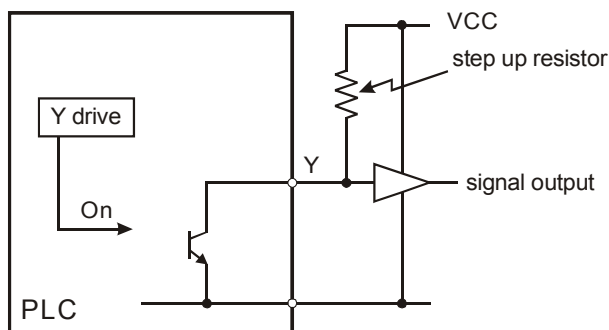
- b) D10 의 내용은 7-분할 디스플레이 첫 번째 집단으로 전송되며 D11 은 두 번째 집단으로 보내져 디스플레이 창에 표시됩니다. D10=1234 이고 D11=K4321 이라면 첫 번째 집단은 1 2 3 4 라고 표시되고 두 번째 집단은 4 3 2 1 이라고 표시됩니다.

##### 5. 7-분할 디스플레이 스캔 출력 배선



##### 주의:

1. V4.9(버전 4.9) 및 그 이상의 ES/EX/SS 시리즈는 이 명령을 제공합니다. (SEGL).
2. ES/EX/SS 시리즈의 V4.9 는 7-분할 디스플레이 4 자리 숫자집단을 제공하며 출력을 위해 8 포인트를 사용합니다. SEGL 명령은 프로그램상에서 단 한번 쓸 수 있으며 사용가능 연산자 n 은 0 에서 3 까지입니다.
3. 이 명령이 실행중일 때 스캔시간은 10ms 보다 길어야 합니다. 스캔시간이 10ms 보다 짧을경우 스캔시간 고정기능을 사용하여 10ms 로 스캔시간을 고정하십시오.
4. PLC 출력용 트랜지스터에 적절한 7-분할 디스플레이를 사용하십시오.
5. n 세팅: 이것은 트랜지스터 출력루프의 극성을 세팅하는데 사용합니다. 양극과 음극을 세팅할 수 있습니다. 이를 통해 연결할 수 있는 7-분할 디스플레이는 4 자리 숫자 1 개집단과 2 개집단 입니다.
6. PLC 트랜지스터 출력은 NPN 타입이며 출력을 모으는데 개방되어 있습니다. 배선시 출력은 전압 증가 레지스터를 VCC(30VDC 보다 낮음)에 연결해야 합니다. 그러므로 출력포인트 Y 는 '켜짐' 이고 출력은 낮은 전위가 됩니다.



## 7. BCD 코드의 정논리(음극성) 출력

BCD 값				Y 출력 (BCDcode)				출력신호			
b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	8	4	2	1	A	B	C	D
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1	0

## 8. BCD 코드의 음논리(양극성) 출력

BCD 값				Y 출력 (BCDcode)				출력신호			
b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	8	4	2	1	A	B	C	D
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	0	0
0	1	0	1	1	0	1	0	0	1	0	1
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	1	0	1	1	0	1	0	0	1

## 9. 디스플레이 스캔(래치) 신호

정논리(음극성) 출력		부정논리(양극성) 출력	
Y output (Latch)	Output control signal	Y output (Latch)	Output control signal
1	0	0	1

## 10. n 매개변수 세팅:

7-분할 디스플레이 집단 숫자	1 개 집단				2 개 집단			
Y of BCD code outputs	+		-		+		-	
Display scan latch signal	+	-	+	-	+	-	+	-
n	0	1	2	3	4	5	6	7

'+' : 정논리(음극성) 출력    '-' : 부정논리(양극성) 출력

## 11. PLC 트랜지스터의 출력 극성과 입력 극성 7-분할 디스플레이 스캔은 n 세팅에 의해 규정될 수 있습니다.

API	기호	연산자				기능				제어장치						
	ARWS	<b>S</b>	<b>D<sub>1</sub></b>	<b>D<sub>2</sub></b>	<b>n</b>	화살표 키 입력				ES	EX	SS	SA	SX	SC	EH

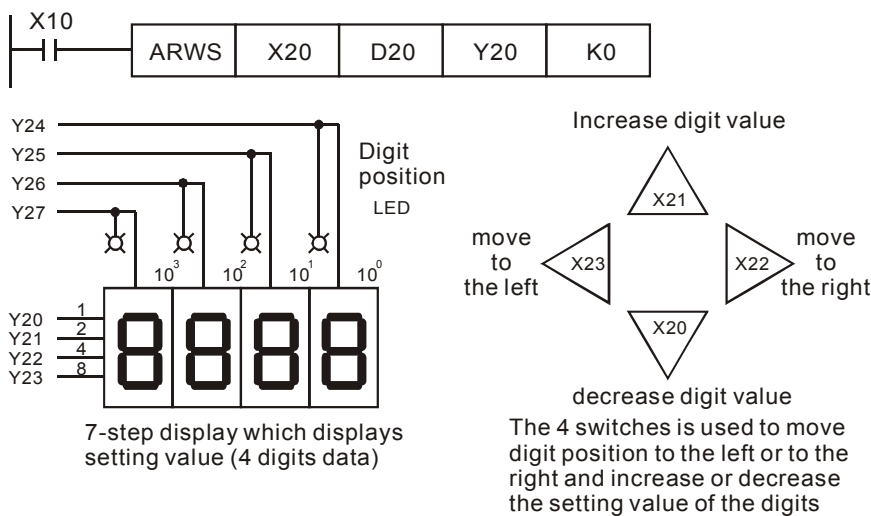
Type OP	비트장치				워드장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ARWS: 9 단계
S	*	*	*	*												
D <sub>1</sub>											*	*	*	*	*	
D <sub>2</sub>		*														
n					*	*										

**설명:**

**S:** 입력 키의 시작장치(4 연속 포인트 점유) **D<sub>1</sub>:** 7-분할 디스플레이의 디스플레이 장치 **D<sub>2</sub>:** 7-분할 디스플레이의 스캔 출력 시작장치 **n:** 출력신호와 스캔신호의 극성 세팅(n=0~3)

**프로그램 예시:**

1. ARWS 명령은 프로그램 상에서 오직 한번만 사용 가능합니다.
2. SA 시리즈 모델은 연산자 S 와 D<sub>2</sub> 의 인덱스 레지스터 E, F 를 지원하지 않으며 오직 가장 적합한 숫자인 "0" 을 가진 장치만을 지시합니다. 예를들어 Y0,Y10,Y20...
3. 명령이 실행되면 X20 은 아래방향, X21 은 위방향, X22 는 오른쪽, X23 은 왼쪽을 지칭하게 됩니다. 이 키들은 외부세팅 값을 수정하거나 표시하는데 사용됩니다. 세팅값은 D20 에 저장되고 세팅 범위는 0 에서 9,999 까지입니다.
4. X10=켜짐 일경우 10<sup>3</sup> 가 효율적인 세팅자릿수 입니다. 왼쪽방향키를 누르면 효율적 세팅자릿수가 디스플레이 되며 103→100→101→102→103→100 의 방향으로 점프하게 됩니다.
5. 오른쪽방향키를 누르면 효율적 세팅자릿수가 디스플레이 되며 103→102→101→100→103→102 의 방향으로 점프하게 됩니다. 한편, Y24 에서 Y27 까지 연결된 LED 자릿수 위치 또한 효율적인 세팅자릿수를 지시하는데 도달할 것입니다.
6. 위방향키를 누르면 효율적 숫자는 0→1→2→...8→9→0→1 바뀌게 됩니다. 아래방향키는 0→9→8→...1→0→9 로 바뀌게 되며, 바뀐 값은 7-분할 디스플레이에 표시됩니다.



API	기호	연산자	기능	제어장치						
				ES	EX	SS	SA	SX	SC	EH
76	ASC	<b>S</b> <b>D</b>	ASCII 코드 변환							

Type OP	비트장치				워드장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S																ASC: 11 단계
D											*	*	*			

연산자:

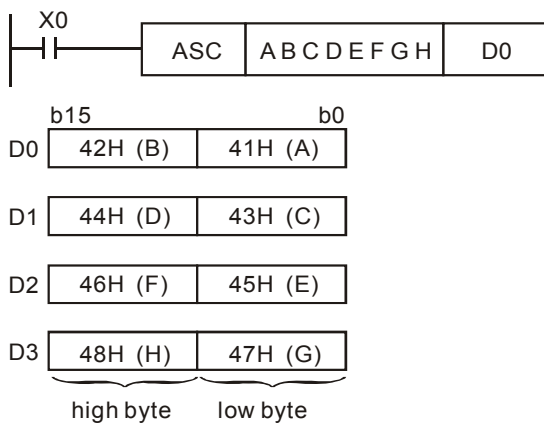
**S:** ASCII 코드로 전환될 영숫자 문자    **D:** ASCII 코드 저장 위치

설명:

1. 연산자 S 는 WPLSoft 에서 입력된 8 영어 문자이거나 HPP02 에서 입력된 ASCII 코드입니다.
2. 이 명령을 사용할 때 영숫자 문자는 7-분할 디스플레이와 연결하여 에러메시지를 직접 표시할 수 있게 됩니다.

프로그램 예시:

1. X0=켜짐 일 때, A~H 는 ASCII 코드로 변환되고 D0~D3 에 저장됩니다.



2. M1161=켜짐 일 때, 모든 문자로부터 변환된 ASCII 코드는 레지스터 상에서 낮은 8-bit(b7~b0)를 점유할 것입니다. 높은 바이트는 무효화 되고 내용은 0 으로 채워집니다. 이것은 한 레지스터에 오직 한 문자만 저장 가능하게 된다는 것을 의미합니다.

	b15	b0
D0	00 H	41H (A)
D1	00 H	42H (B)
D2	00 H	43H (C)
D3	00 H	44H (D)
D4	00 H	45H (E)
D5	00 H	46H (F)
D6	00 H	47H (G)
D7	00 H	48H (H)
	high byte	low byte

API	기호	연산자	기능	제어장치						
77	PR	<b>S</b> <b>D</b>	ASCII 코드 출력	ES	EX	SS	SA	SX	SC	EH

Type OP	비트장치				워드장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PR: 5 단계
S											*	*	*			
D		*														

연산자:

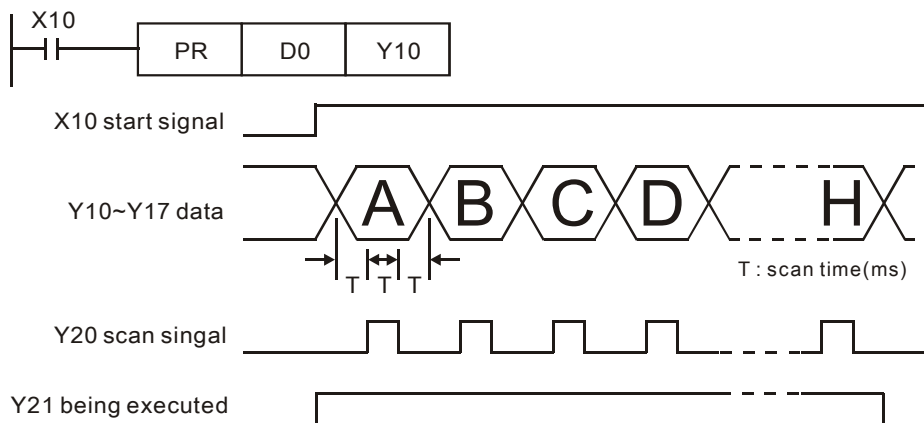
**S:** ASCII 코드 저장장치(4 개 연속장치 점유) **D:** ASCII 코드 출력 외부출력장치(10 개 연속장치 점유)

설명:

1. PR 명령은 프로그램상에서 2 회까지 사용 가능합니다.
2. SA 시리즈 모델은 연산자 D 의 인덱스 레지스터 E, F 를 지원하지 않습니다.
3. 이 명령은 D 에 의해 특정된 출력장치 순서에 따라 S 장치의 4 레지스터에 저장된 ASCII 코드를 출력합니다.

프로그램 예시 1:

1. 먼저 API 76 ASC 명령은 A~H 를 ASCII 코드로 전환하며 D0~D3 에 이를 저장합니다. 그런다음 이 명령을 사용하여 A~H 의 순서로 출력을 실행할 수 있습니다.
2. 명령 실행 중 M1027=꺼짐 일 때, X10 은 '꺼짐' 에서 '켜짐' 으로 바뀌고 Y10(낮은 바이트) 에서 Y17(높은 바이트)까지 데이터 출력 장치에 의해 특정되며 Y20 는 스캔신호로, Y21 은 모니터 신호로 특정됩니다.
3. X10 이 명령이 수행되는 동안 '꺼짐' 에서 '켜짐' 으로 바뀔경우 데이터 출력에 간섭이 발생합니다. X10 이 한 번 이상 '켜짐' 이면 데이터는 재전송 됩니다.

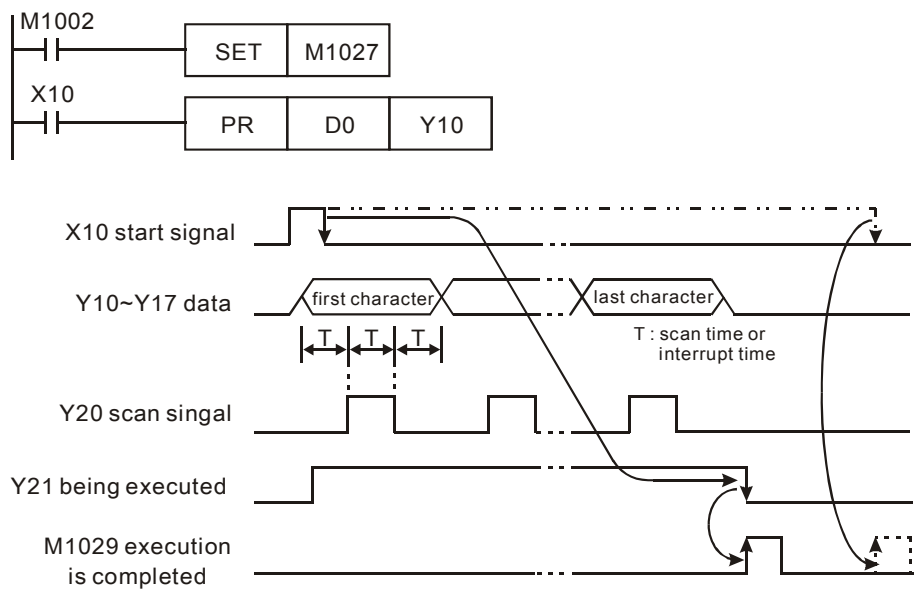


프로그램 예시 2:

1. PR 명령은 8 문자열 출력을 실행합니다. M1027=꺼짐 이면 최대 8 문자열이 연속적으로 출력될 수 있습니다. M1027=켜짐 이면 1 에서 16 문자열 출력 작업이 실행될 수 있습니다.
2. 명령 실행 중 M1027=켜짐 이면 X10 은 '꺼짐' 에서 '켜짐' 으로 바뀌고, Y10(낮은 바이트) 에서 Y17(높은 바이트)까지 데이터 출력 장치로 특정되며, Y20 은 스캔 신호로 Y21 은 모니터 신호로 특정됩니다. 이 모드는 16 개 문자열 출력 작업을 실행할 수 있습니다.
3. 문자열 00H (NUL)이 송신되었다면 문자열의 마지막 문자가 송신되었다는 뜻이며, PR 명령의 작업이 더 이상 진행되지 않는다는 의미입니다.



4. 드라이브 접촉 X10 이 '켜짐' 이지만 데이터 출력 작업을 1 회 마친후 자동적으로 멈춥니다. 그러나 X10 이 항상 '켜짐' 이라면 M1029 는 작동하지 않습니다.



주의:

1. 이 명령은 트랜지스터 출력만을 사용해야 합니다.
2. 이 명령을 수행할 때에는 스캔시간을 고정하거나 시간간섭 서브루틴 내에서 수행해야 합니다.

API	기호			연산자	기능	제어장치							
78	D	FROM	P	<b>(m<sub>1</sub>)</b> <b>(m<sub>2</sub>)</b> <b>(D)</b> <b>(n)</b>	모듈에서 CR 읽기	ES	EX	SS	SA	SX	SC	EH	

Type OP	비트장치				워드장치												프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F					
m <sub>1</sub>					*	*											FROM, FROMP: 9 단계 DFROM, DFROMP: 17 단계			
m <sub>2</sub>					*	*														
D								*	*	*	*	*	*	*	*	*				
n					*	*														

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

#### 연산자:

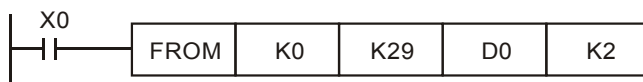
**m<sub>1</sub>**: 특별 모듈 번호(ES/SA 는 m<sub>1</sub>=0~7; EH 는 m<sub>1</sub>=0~255)    **m<sub>2</sub>**: 읽기 위한 특수모듈 CR 번호(제어 레지스터) (16-bit 명령: ES/SA 는 m<sub>2</sub>=0~48, EH 는 m<sub>2</sub>=0~254; 32-bit 명령: EH 는 m<sub>2</sub>=0~127)    **D**: 저장데이터 읽을 위치  
**n**: 한 번에 읽을 수 있는 데이터 단어 (16-bit 명령: ES/SA 는 n=1~(49- m<sub>2</sub>), EH 는 n=1~(255- m<sub>2</sub>); 32-bit 명령: ES/SA 는 n=1~(25- m<sub>2</sub>); EH 는 n=1~(127- m<sub>2</sub>))

#### 설명:

- DVP PLC 는 이 명령을 사용하여 특수 모듈의 CR 데이터를 읽습니다.
- D 가 비트 연산자를 지칭한다면, 16 비트 명령을 위해 K1~K 까지 사용가능하며 K5~K8 까지 32 비트 명령을 위해 사용할 수 있습니다.
- 특수모듈의 번호체계에 대해 더 자세히 알고싶다면 다음 보충설명을 참조하시기 바랍니다.
- ES 시리즈는 레지스터 E,F 를 지원하지 않습니다.

#### 프로그램 예시:

- PLC 의 특수모듈#0 에서 D0 까지의 내용 CR#29 와 PLC 의 특수모듈#0 에서 D1 까지의 내용 CR#30 을 읽기위한 프로그램 입니다. 2 개 데이터를 한 번에 읽을 수 있습니다(n=2)..
- X0=켜짐 일 때 명령이 수행됩니다. X0=꺼짐일 때는 명령이 수행되지 않으며 이전에 읽은 데이터는 변하지 않습니다.



API	기호			연산자	기능	제어장치						
79	D	TO	P	(m1) (m2) (S) (n)	CR 을 모듈에 기록	ES	EX	SS	SA	SX	SC	EH

Type	비트장치				워드장치											프로그램 단계	
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	TO, TOP: 9 단계 DTO, DTOP: 17 단계	
m <sub>1</sub>					*	*											
m <sub>2</sub>					*	*											
S					*	*	*	*	*	*	*	*	*	*	*		
n					*	*											

PULSE							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

#### 연산자:

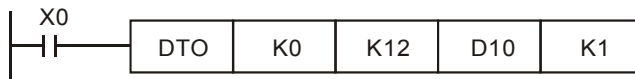
**m<sub>1</sub>**: 특수 모듈 번호(ES/SA는 m<sub>1</sub>=0~7, EH는 m<sub>1</sub>=0~255)    **m<sub>2</sub>**: 기록하기 위한 특수모듈 CR 번호(제어 레지스터)  
(16-bit 명령: ES/SA는 m<sub>2</sub>=0~48, EH는 m<sub>2</sub>=0~254; 32-bit 명령: EH는 m<sub>2</sub>=0~127)    **S**: CR에 기록하기 위한 데이터  
**n**: 한번에 기록할 수 있는 단어 수(16-bit 명령: ES/SA는 n=1~(49- m<sub>2</sub>), EH는 n=1~(255- m<sub>2</sub>); 32-bit 명령: ES/SA는 n=1~(25- m<sub>2</sub>), EH는 n=1~(127- m<sub>2</sub>))

#### 설명:

- S가 비트 연산자를 할당할 때 K1~K4는 16비트 명령에 사용될 수 있고, K1~K8은 32비트 명령에 사용될 수 있습니다.
- DVP 시리즈 PLC는 이 명령을 특수 모듈의 CR에 데이터를 기록하는데 사용할 수 있습니다.
- ES 시리즈는 인덱스 레지스터 E, F를 지원하지 않습니다.

#### 프로그램 예시 :

- 32-비트 명령 DTO를 사용하면, 프로그램은 특수모듈#0의 CR#12에 D11과 D10을 써넣을 것입니다.
- X0=켜짐 일 때 명령이 수행되며 X0=꺼짐 이면 수행되지 않습니다. 기존에 쓰인 데이터는 변하지 않습니다.



#### 3. 명령 연산자 규칙:

- m1**: 특수모듈의 번호 배열. PLC MPU에 연결된 특수모듈 숫자입니다. 근접한 곳에서 원거리까지의 특수모듈 번호배열 방식은 0에서 7까지입니다. 최대치는 8개 특수모듈이며 I/O 포인트를 점유하지 않습니다.
- m2**: CR 번호. 특수모듈의 36 그룹 메모리 내장형 16비트는 CR(Control Register)라 지칭됩니다. CR은 십진수 숫자를 사용합니다(#0~#35). 특수모듈의 모든 운영상태와 세팅값 또한 포함됩니다

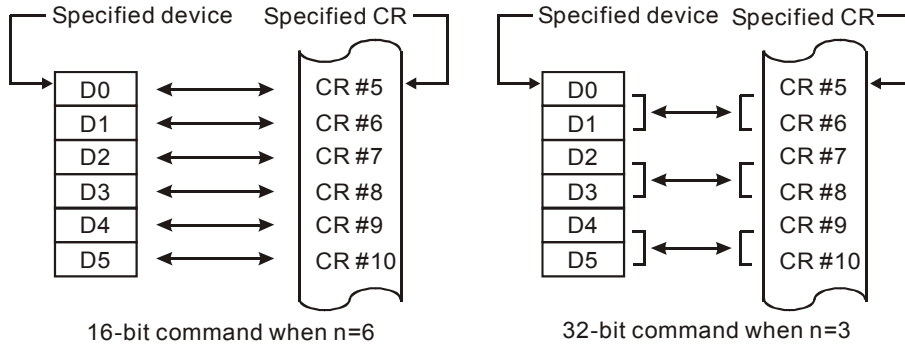
#### 주의:

- FROM/TO 명령을 사용한다면, CR의 읽기/쓰기 유닛은 한 번에 1 숫자입니다. DFROM/DTO 명령을 사용한다면, CR의 읽기/쓰기 유닛은 한 번에 2 숫자입니다.

Upper 16-bit    Lower 16-bit



- 전환집단 n의 숫자. 16비트 명령의 n=2와 32비트 명령의 n=1은 뜻이 같습니다.



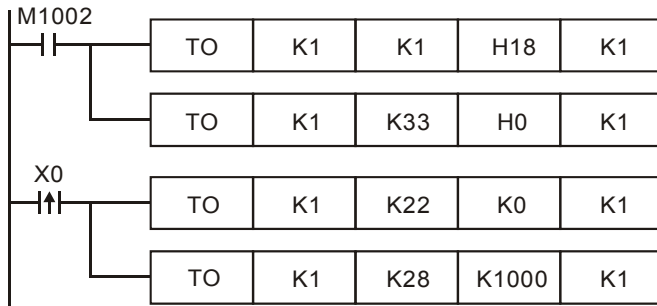
3. ES/EX/SS 시리즈 모델은, flag M1083 이 제공되지 않습니다. FROM/TO 명령이 수행되면, 모든 간섭 (내부/외부 서브루틴 간섭 포함)이 불가능해집니다. 모든 간섭은 FROM/TO 명령이 완료된 후 수행될 수 있습니다. 반면 FROM/TO 명령은 서브루틴 간섭중 실행할 수 있습니다.
4. flag M1083 의 기능(FROM/TO 모드 변환)은 SA/SX/SC/EH 시리즈 모델에서 제공됩니다
  - a) M1083=꺼짐 이고, FROM/TO 명령이 실행된다면, 모든 (내부/외부 서브루틴 간섭 포함)이 불가능해집니다. 모든 간섭은 FROM/TO 명령이 완료된 후 수행될 수 있습니다. 반면 FROM/TO 명령은 서브루틴 간섭중 실행할 수 있습니다.
  - b) M1083=켜짐 이고, 간섭이 FROM/TO 명령이 프로그램 되었을 때 발생한다면 FROM/TO 은 간섭을 실행하기 위해 중단됩니다. 그러나 FROM/TO 명령은 서브루틴 간섭 중에 실행될 수 없습니다.
5. FROM/TO 명령 적용 프로그램 예시
6. 예시 1: DVP-04AD 의 A/D 변환 성격의 커브를 OFFSET 값 CH1 에서 0v(=K0<sub>LSB</sub>)값으로 세팅하여 수정하고 CH1 에서 2.5V (=K2000<sub>LSB</sub>) 의 GAIN 값을 수정합니다.



- a) 아날로그 입모드 No.0 의 H0 에서 CR#1 까지 기록합니다. 그리고 CH1 을 모드 0 으로 세팅합니다(전압입력 : -10V to +10V).
- b) H0 에서 CR#33 까지 기록하고 CH1 에서 CH4 특성 수정을 허가합니다.
- c) X0 이 '꺼짐' 에서 '켜짐' 이 될 때, OFFSET 값의 K0<sub>LSB</sub> 는 CR#18 에 기록하고 GAIN 값의 K2000<sub>LSB</sub> 는 CR#24 에 기록됩니다.
7. 예시 2: DVP-04AD 의 A/D 변환 성격의 커브를 OFFSET 값 CH2 에서 2mA(=K400 LSB) 값으로 세팅하여 수정하고 CH2 에서 18 mA(=K3600LSB) 의 GAIN 값을 수정합니다.



- a) 아날로그 입력모드 No.0 의 H18 에서 CR#1 까지 기록합니다. 그리고 CH2 를 모드 3 으로 세팅 합니다(전류입력 : -20mA to +20mA).
  - b) H0 에서 CR#33 까지 기록하고 CH1 에서 CH4 특성 수정을 허가합니다.
  - c) X0 이 '꺼짐' 에서 '켜짐' 이 될 때, OFFSET 값의 K400<sub>LSB</sub> 는 CR#19 에 기록하고 GAIN 값의 K3600<sub>LSB</sub> 는 CR#25 에 기록됩니다.
8. 예시 3: DVP-02DA 의 D/A 변환 성격의 커브를 OFFSET 값 CH2 에서 0mA(=K0LSB) 값으로 세팅하여 수정하고 CH2 에서 10mA(=K1000LSB) 의 GAIN 값을 수정합니다.



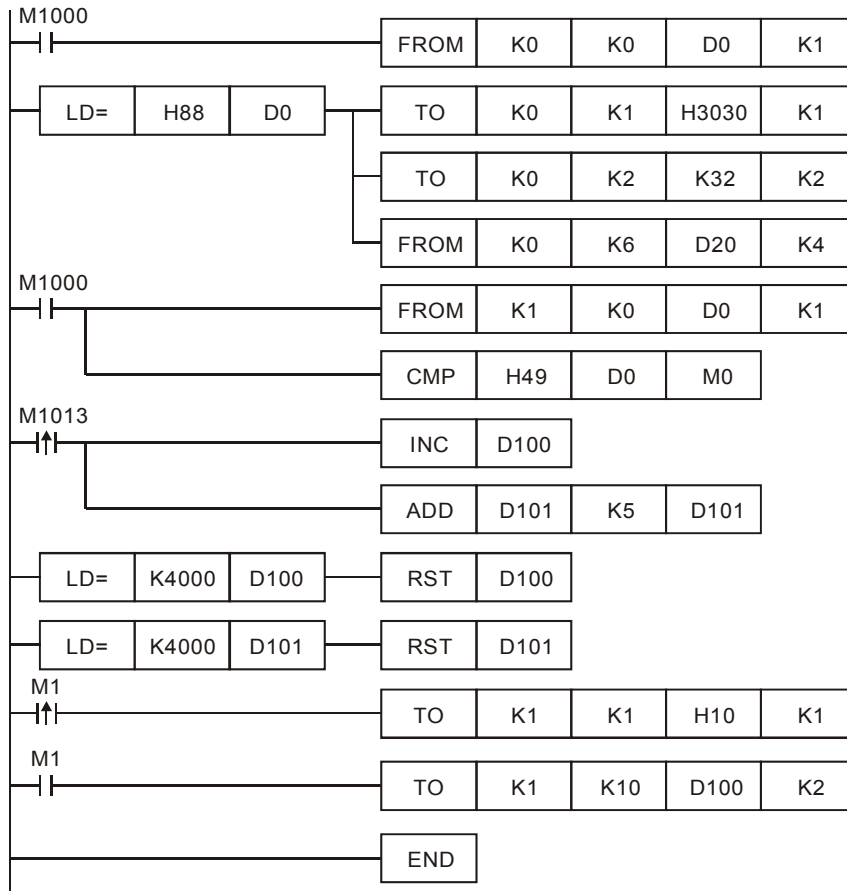
- a) 아날로그 입력모드 No.1 의 H18 에서 CR#1 까지 기록합니다. 그리고 CH2 를 모드 3 으로 세팅 합니다(전류입력: 0mA to +20mA).
  - b) H0 에서 CR#33 까지 기록하고 CH1 에서 CH2 특성 수정을 허가합니다.
  - c) X0 이 '꺼짐' 에서 '켜짐' 이 될 때, OFFSET 값의 K0<sub>LSB</sub> 는 CR#22 에 기록하고 GAIN 값의 K1000<sub>LSB</sub> 는 CR#28 에 기록됩니다.
9. 예시 4: DVP-02DA 의 D/A 변환 성격의 커브를 OFFSET 값 CH2 에서 2mA(=K400<sub>LSB</sub>) 값으로 세팅하여 수정하고 CH2 에서 18mA(=K3600<sub>LSB</sub>)의 GAIN 값을 수정합니다.



- a) 아날로그 입력모드 No.1 의 H10 에서 CR#1 까지 기록합니다. 그리고 CH2 를 모드 2 로 세팅 합니다(전류입력: +4mA to +20mA).
- b) H0 에서 CR#33 까지 기록하고 CH1 에서 CH2 특성 수정을 허가합니다.

- c) X0 이 ‘꺼짐’ 에서 ‘켜짐’ 이 될 때, OFFSET 값의 K400<sub>LSB</sub> 는 CR#23 에 기록하고 GAIN 값의 K3600<sub>LSB</sub> 는 CR#29 에 기록됩니다.

10. 예시 5: DVP-04AD 와 DVP-02DA 모듈이 함께 쓰일때의 프로그램 예시



- a) 확장 모듈 K0 에서 데이터 모델타입을 읽고 데이터가 H88 일 경우 분류하십시오(DVP-04AD 모델타입).
- b) DVP-04AD 모델타입은 드라이브 접촉 M1 이 있고 입력모드는 CR#1: (CH1, CH3)= mode 0, (CH2, CH4)= mode 3 으로 세팅합니다.
- c) CR#2 와 CR#3 의 모드를 세팅 하십시오. CH1 과 CH2 의 평균시간은 K32 입니다.
- d) CH1~CH4 의 입력 신호값을 CR#6~CR#9 로부터 읽고 D20 에서 D23 에 저장하십시오.
- e) 확장 모듈 K1 에서 데이터 모델타입을 읽고 데이터가 H49 일 경우 분류하십시오(DVP-02DA 모델타입).
- f) D100 은 K1 을 증가시키고 D101 은 K5 를 매초마다 증가시킵니다.
- g) D100 과 D101 의 값이 K4000 에 다다르면 0 으로 리셋됩니다.
- h) DVP-04AD 모델타입은 드라이브 접촉 M1 이 있고 입력모드는 CR#1: CH1 모드에서 0, CH2 모드에서 2 로 세팅합니다.
- i) 출력 세팅 CR#10 과 CR#11 을 D100 과 D101 로 세팅 하십시오. 아날로그 출력은 D100 과 D101 값으로 변환됩니다.

API	기호	연산자	기능	제어장치						
				ES	EX	SS	SA	SX	SC	EH
80	RS	<b>S</b> <b>m</b> <b>D</b> <b>n</b>	연속데이터 통신							

Type OP	비트장치				워드장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			RS: 9 단계
m					*	*							*			
D													*			
n					*	*							*			

**연산자:**

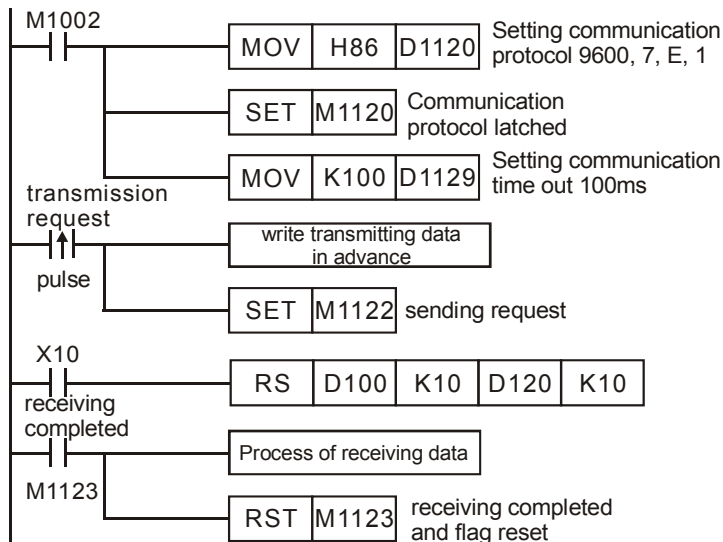
**S:** 데이터 전송 시작장치 **m:** 전송데이터 그룹 번호 (**m**=0~256) **D:** 수신 데이터 시작장치  
**n:** 수신데이터 그룹 번호 (**n**=0~256)

**설명:**

- 이 명령은 MPU가 통신 인터페이스를 직렬로 연결하기 위해 RS-485를 사용하기 편리하게 해줍니다.  
 이것은 문자데이터를 **S** 소스데이터 레지스터로 저장하고 길이를 **m**으로 세팅합니다. 또한 레지스터 **D**를 **n** 길이로 받도록 세팅할 수도 있습니다. **S**와 **D**를 위한 인덱스 레지스터 **E**, **F**를 사용할 때는 명령실행중 레지스터 **E**, **F** 값을 변경하지 마십시오; 변경할 경우 읽기/쓰기 지시에 예러가 발생할 수 있습니다
- 데이터 송신이 필요치 않다면 **m**은 K0를 지정하게 되며, 데이터 수신이 필요치 않다면 **n**이 K0를 지정하게 됩니다.
- RS 명령은 프로그램상에서 무한정 사용가능하지만 한번에 한 개 이상의 RS 명령을 수행할 수는 없습니다
- RS 명령 실행중 송신 데이터를 바꿀수 없습니다.
- 외부/주변장비가 RS-485 직렬 통신을 보유하고 있거나 장비의 통신 포맷이 공용일 경우 본 RS 명령을 사용하여 PLC와 외부/주변장비의 데이터 송신과 수신에 사용하십시오
- 만약 외부/주변장비가 MODBUS, DVP 포맷일 경우 PLC를 사용해 API 100 MODRD, API 101 MODWR, API 150 MODRW 등의 통신편의를 얻을 수 있습니다. 보다 자세한 사항을 위하여 개별명령 설명을 참조하십시오.
- M1120~M1161 특별보조중계의 정보와, RS-485 통신명령과 관련된 특별 데이터레지스터 D1120~D1131의 정보를 위하여 다음의 설명을 참조하십시오.

**프로그램 예시 1:**

- D100부터 시작된 데이터를 레지스터에 기록하고 M1122(flag 요청 보냄)를 '켜짐'으로 세팅합니다.
- RS 명령이 실행되는 중 X10=켜짐 일 때, PLC는 데이터 송신/수신 대기상태일 것입니다. PLC는 10개의 연속 데이터를 D100으로부터 보내기 시작합니다. M1122는 데이터 전송 끝무렵에 '꺼짐'으로 세팅됩니다(프로그램에서 RST M1122 명령을 실행하지 마십시오). 1ms 이후 10개의 외부 데이터를 받기시작할 것이며 D120으로 시작되는 연속레지스터에 이를 저장할 것입니다.
- 데이터 수신이 완료되면 M1123은 '켜짐'으로 세팅됩니다(프로그램은 데이터 수신이 완료되면 M1123을 '꺼짐'으로 세팅하며 데이터 송수신 대기상태로 전환됩니다). PLC 프로그램을 RST M1123의 연속 실행에 사용하지 마십시오.

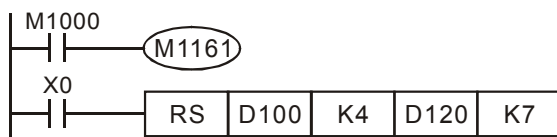


## 프로그램 예시 2:

8-비트 모드(M1161=ON) / 16-비트 모드(M1161=OFF) 스위치:

《8-비트 모드》:

1. PLC 전송 데이터의 헤드코드와 테일코드는 D1124~D1126 에 따른 M1126 과 M1130 에 의해 세팅됩니다. 세팅 후, PLC 는 RS 명령 실행중 자동적으로 세팅된 헤드코드, 테일코드를 전송하게 됩니다.
2. M1161=켜짐 일 때, 전환모드는 8-비트 입니다. 16-비트 데이터는 높은 바이트, 낮은 바이트로 분류됩니다. 높은 바이트는 무시되고 낮은 바이트는 받아들여 전송합니다.



Transmit data: (PLC → external equipment)

STX	D100L	D101L	D102L	D103L	ETX1	ETX2
Head code	<p>(S) source data register will start from low byte of D100</p> <p>(m) length = 4</p>				Tail code 1	Tail code 2

Receive data: (external equipment → PLC)

D120L	D121L	D122L	D123L	D124L	D125L	D126L
Head code	<p>(S) receive data register will start from low byte of D120</p> <p>(n) length = 7</p>				Tail code 1	Tail code 2

3. PLC 는 헤드,테일코드를 포함한 모든 외부장비로부터의 전송데이터를 모두 받습니다. n 세팅길이에 주의하시기 바랍니다.

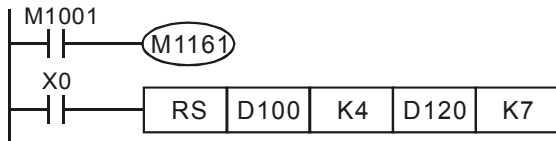
《16-비트 모드》:

1. PLC 전송 데이터의 헤드코드와 테일코드는 D1124~D1126 에 따른 M1126 과 M1130 에 의해 세팅됩니다.



세팅 후, PLC 는 RS 명령 실행중 자동적으로 세팅된 헤드코드, 테일코드를 전송하게 됩니다.

2. M1161=꺼짐 일 때, 전환모드는 16-비트 입니다. 16-비트 데이터는 높은 바이트, 낮은 바이트로 분류되고 낮은 바이트는 데이터 송수신에 사용됩니다.



송신 데이터: (PLC → 외부장비)

STX	D100L	D100L	D101L	D101L	ETX1	ETX2
Head code	<div> <div>S</div> source data register will start from low byte of D100 </div> <div> <div>m</div> length = 4 </div>				Tail code 1	Tail code 2

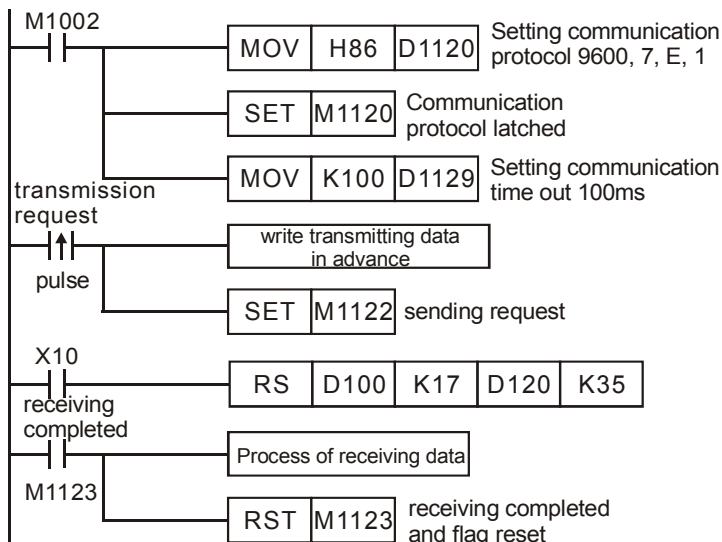
수신 데이터: (외부장비 → PLC)

D120L	D120H	D121L	D121H	D122L	D122H	D123L
Head code	<div> <div>D</div> receive data register will start from low byte of D120 </div> <div> <div>n</div> length = 7 </div>				Tail code 1	Tail code 2

3. PLC 는 헤드,테일코드를 포함한 모든 외부장비로부터의 전송데이터를 모두 받습니다. n 세팅길이에 주의하시기 바랍니다

### 프로그램 예시 3:

PLC 가 VFD-B 시리즈 AC 드라이버에 연결될 때(ASCII 모드, M1143=꺼짐), (16-비트 모드, M1161=꺼짐), VFD-B 매개변수 주소 H2101 에서 시작되는 6 개의 연속데이터를 송신할 것입니다.



PLC ⇨ VFD-B, PLC transmitting: “: 01 03 2101 0006 D4 CR LF “

VFD-B ⇨ PLC, PLC 수신: “: 01 03 0C 0100 1766 0000 0000 0136 0000 3B CR LF “

PLC 데이터 레지스터 송신 (PLC 송신 메시지)

레지스터	데이터	
D100 low byte	‘.’	3A H
D100 high byte	‘0’	30 H
D101 low byte	‘1’	31 H
D101 high byte	‘0’	30 H
D102 low byte	‘3’	33 H
D102 high byte	‘2’	32 H
D103 low byte	‘1’	31 H
D103 high byte	‘0’	30 H
D104 low byte	‘1’	31 H
D104 high byte	‘0’	30 H
D105 low byte	‘0’	30 H
D105 high byte	‘0’	30 H
D106 low byte	‘6’	36 H
D106 high byte	‘D’	44 H
D107 low byte	‘4’	34 H
D107 high byte	CR	D H
D108 low byte	LF	A H

PLC 데이터 레지스터 수신 (VFD-B 응답 메시지)

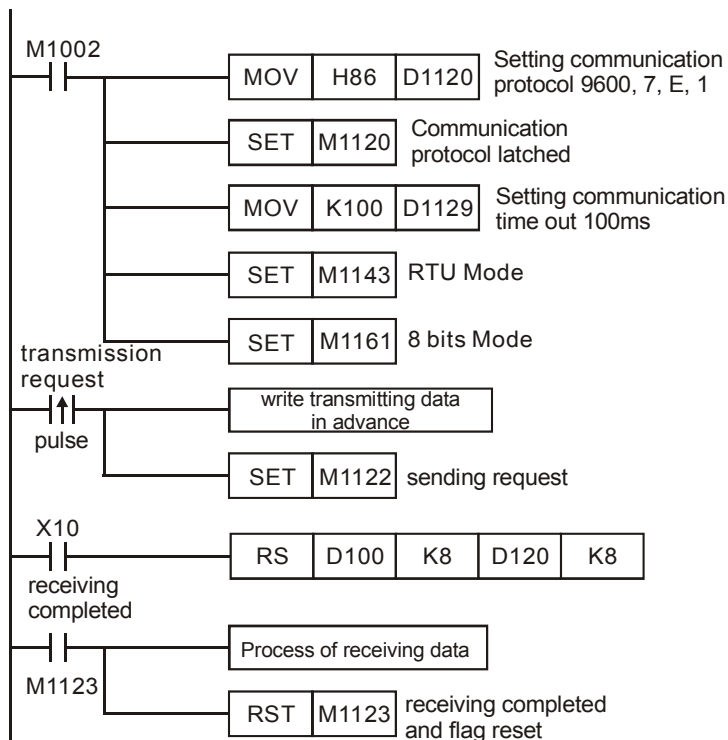
레지스터	데이터	
D120 low byte	‘.’	3A H
D120 high byte	‘0’	30 H
D121 low byte	‘1’	31 H
D121 high byte	‘0’	30 H
D122 low byte	‘3’	33 H
D122 high byte	‘0’	30 H
D123 low byte	‘C’	43 H
D123 high byte	‘0’	30 H
D124 low byte	‘1’	31 H
D124 high byte	‘0’	30 H

레지스터	데이터	
D125 low byte	‘0’	30 H
D125 high byte	‘1’	31 H
D126 low byte	‘7’	37 H
D126 high byte	‘6’	36 H
D127 low byte	‘6’	36 H
D127 high byte	‘0’	30 H
D128 low byte	‘0’	30 H
D128 high byte	‘0’	30 H
D129 low byte	‘0’	30 H
D129 high byte	‘0’	30 H
D130 low byte	‘0’	30 H
D130 high byte	‘0’	30 H
D131 low byte	‘0’	30 H
D131 high byte	‘0’	30 H
D132 low byte	‘1’	31 H
D132 high byte	‘3’	33 H
D133 low byte	‘6’	36 H
D133 high byte	‘0’	30 H
D134 low byte	‘0’	30 H
D134 high byte	‘0’	30 H
D135 low byte	‘0’	30 H
D135 high byte	‘3’	33 H

레지스터	데이터		LRC CHK 0  종료
D136 low byte	'B'	42 H	
D136 high byte	CR	D H	
D137 low byte	LF	A H	

### 프로그램 예시 4:

PLC 가 VFD-BAC 드라이브에 연결되는 경우 (RTU 모드, M1143=ON), (16 비트 모드, M1161=ON), VFD-B 파라미터 주소 H2000 에 사전에 송신 데이터 H12 를 기록



PLC ⇒ VFD-B, PLC 송신: 01 06 2000 0012 02 07

VFD-B ⇒ PLC, PLC 수신: 01 06 2000 0012 02 07

### PLC 데이터 레지스터 송신 (PLC 송신 메시지)

레지스터	데이터	
D100 low byte	01 H	주소
D101 low byte	06 H	기능
D102 low byte	20 H	데이터 주소
D103 low byte	00 H	
D104 low byte	00 H	데이터 내용
D105 low byte	12 H	
D106 low byte	02 H	CRC CHK Low
D107 low byte	07 H	CRC CHK High

### PLC 데이터 레지스터 수신 (VFD-B 응답 메시지)

레지스터	데이터	
D120 low byte	01 H	주소
D121 low byte	06 H	기능
D122 low byte	20 H	데이터 주소
D123 low byte	00 H	
D124 low byte	00 H	데이터 내용
D125 low byte	12 H	
D126 low byte	02 H	CRC CHK Low
D127 low byte	07 H	CRC CHK High

## 참고:

## 1. RS-485 통신 RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW 커맨드 관련 플래그:

플래그	기능 설명	동작
M1120	통신 프로토콜 유지. 통신 설정을 유지하기 위해 사용됩니다. PLC 는 첫 번째 프로그램 스캔 이후에 특수 데이터 레지스터 D1120 에 따라 통신 설정을 초기화하며, 두 번째 프로그램 스캔이 시작되고 RS 커맨드가 실행될 때 특수 데이터 레지스터 D1120 에 따라 통신 설정을 초기화합니다. 통신 프로토콜이 고정적인 경우, M1120 플래그를 “ON” 상태로 설정할 수 있습니다. 이 경우에는 D1120 설정이 변경되는 경우라도 RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW 커맨드 수행 후 통신 프로토콜 설정이 초기화되지 않습니다.	사용자 설정 및 확인
M1121	이 플래그가 “OFF” 상태인 경우, PLC 의 RS-485 는 통신 정보를 전송합니다.	시스템 동작
M1122	요청 전송. 사용자는 RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW 커맨드를 데이터 송수신에 사용할 경우 명령 펄스를 통해 M1122 플래그를 “ON” 상태로 설정해야 합니다. 위의 커맨드가 실행될 때, PLC 는 데이터를 송수신합니다. 위의 커맨드가 송신을 완료하면 M1122 는 초기화됩니다.	사용자 설정 후 시스템 자동 확인
M1123	수신 완료. RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW 커맨드의 실행이 종료되면 M1123 플래그가 “ON” 상태로 설정됩니다. 사용자는 M1123 이 “ON”인 상태에서 수신 데이터를 처리할 수 있으며, 수신 데이터 처리가 완료된 후에 M1123 을 “OFF” 상태로 초기화할 수 있습니다.	시스템 자동 설정 후 사용자 확인
M1124	수신 대기. M1124 가 “ON”으로 설정된 경우, PLC 가 데이터 수신 대기 상태임을 의미합니다.	시스템 동작
M1125	통신 초기화. M1125 가 “ON”으로 설정되면 PLC 통신이 초기화됩니다. 초기화 후에는 반드시 M1125 를 “OFF”로 설정해야 합니다.	사용자 설정 및 확인
M1126	STX/ETX 선택. 사용자/시스템 정의 및 STX/ETX 선택과 관련한 사항은 다음의 따로 정리된 표를 참조하십시오. (RS 커맨드)	
M1130	STX/ETX 선택. 사용자/시스템 정의 및 STX/ETX 선택과 관련한 사항은 다음의 따로 정리된 표를 참조하십시오. (RS 커맨드)	
M1127	통신 커맨드가 송수신을 완료함. RS 커맨드는 포함되지 않습니다.	시스템 자동 설정 후 사용자 확인
M1129	수신 타임 아웃. D1129 가 설정된 상태에서 데이터 수신에 지정된 시간 내에 완료되지 않을 경우 M1129 플래그가 기동됩니다.	
M1128	송수신 지시	
M1131	MODRD / RDST / MODRW 가 HEX 로 변환되는 동안 M1131 플래그가 “ON”으로 설정됩니다. 그 외의 경우 M1131 은 “OFF”로 유지됩니다.	시스템 동작
M1140	MODRD / MODWR / MODRW 데이터 에러 수신	
M1141	MODRD / MODWR / MODRW 커맨드 에러	

M1142	VFD-A 커맨드 데이터 에러 수신	
M1143	ASCII / RTU 모드 선택 “ON”은 RTU 모드이며 “OFF”는 ASCII 모드입니다. (MODRD / MODWR / MODRW 커맨드와 함께 사용)	사용자 설정 및 확인
M1161	8/16 비트 모드 설정. “ON”은 8 비트 모드이며 “OFF”는 16 비트 모드입니다.	

2. RS-485 통신 RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW 커맨드 관련 특수 레지스터:

특수 레지스터	기능 설명
D1038	ES/EX/SS/SA/SX/SC 모델 한정, PLC 의 MPU 가 슬레이브 상태일 때 데이터 수신 지연 시간 설정. 단위는 0.1ms
D1050~D1055	MODRD/RDST 커맨드 실행 후, PLC 가 D1070~D1085 의 ASCII 데이터를 HEX 로 변환하여 16 진수 데이터를 D1050~D1055 에 저장합니다.
D1070~D1085	PLC 내장 RS-485 통신 편의 커맨드. 이 커맨드를 실행하면 리시버로부터 피드백(리턴) 메시지를 받으며 받은 메시지는 D1070~D1085 에 저장됩니다. 사용자는 레지스터의 내용을 보고 리턴 메시지를 확인할 수 있습니다. (RS 커맨드 제외)
D1089~D1099	PLC 내장 RS-485 통신 편의 커맨드. 이 커맨드가 실행되면 송신 메시지가 D1089~D1099 에 저장됩니다. 사용자는 레지스터 내용을 통해 명령어가 올바른지의 여부를 파악할 수 있습니다. (RS 커맨드 제외)
D1120	RS-485 통신 프로토콜에 대해서는 다음의 따로 정리된 표를 참조하십시오.
D1121	PLC MPU 가 슬레이브 상태일 때 PLC MPU 의 통신 주소
D1122	송신 데이터의 잔류 단어
D1123	수신 데이터의 잔류 단어
D1124	시작 단어 정의 (STX). 위의 표를 참조하십시오.
D1125	첫 번째 종료 단어 정의 (ETX1). 위의 표를 참조하십시오.
D1126	두 번째 종료 단어 정의 (ETX2). 위의 표를 참조하십시오.
D1129	비정상 통신 타임 아웃. 단위는 ms. 타임 아웃 시간 설정을 위해 사용합니다. 시간 설정 값이 0 인 경우, 타임 아웃이 없음을 의미합니다. 지정 시간 값이 0 보다 큰 경우, PLC 는 수신 모드로 들어가기 위해 RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW 커맨드를 실행한 후 첫 번째 단어 혹은 어느 두 단어 사이의 시간 간격이 지정 시간 값보다 클 때 M1129 를 “ON”으로 설정합니다. 사용자는 M1129 를 통해 통신 타임 아웃을 조정할 수 있습니다. 타임 아웃 조정이 끝난 후에는 반드시 M1129 를 초기화하시기 바랍니다.
D1130	MODBUS 리턴 에러 코드 기록.
D1256~D1295	PLC 내장 RS-485 통신 편의 커맨드 MODRW. 커맨드 사용 시, 이 커맨드를 통해 송신된 문자들은 D1256~D1295 에 저장됩니다. 사용자는 레지스터의 내용을 통해 커맨드가 올바른지의 여부를 확인할 수 있습니다.
D1296~D1311	PLC 는 사용자에게 의해 지정된 레지스터에 있는 ASCII 데이터를 자동으로 HEX, 16 진수 값으로 변환합니다. (MODRW 명령)

3. D1120: RS-485 통신 프로토콜. 아래 테이블을 참조하여 설정하십시오.

	Content	0	1
<b>b0</b>	데이터 길이	7	8
<b>b1 b2</b>	패리티 비트	00 : 없음 (None) 01 : 홀수 (Odd) 11 : 짝수 (Even)	
<b>b3</b>	정지 비트	1 비트	2 비트
<b>b4 b5 b6 b7</b>	0001 (H1) : 0010 (H2) : 0011 (H3) : 0100 (H4) : 0101 (H5) : 0110 (H6) : 0111 (H7) : 1000 (H8) : 1001 (H9) : 1010 (HA) : 1011 (HB) : 1100 (HC) :	110 150 300 600 1200 2400 4800 9600 19200 38400 57600 (EH/SA/SX/SC 시리즈 모델 한정) 115200 (EH/SA/SX/SC 시리즈 모델 한정)	
<b>b8</b>	시작 단어 선택	없음	D1124
<b>b9</b>	첫 번째 종료 단어 선택	없음	D1125
<b>b10</b>	두 번째 종료 단어 선택	없음	D1126
<b>b15~b11</b>	정의 없음		

4. RS 커맨드를 사용할 때 시작 단어와 종료 단어의 제어 문자는 주변 기기의 통신 포맷에 의해 정의됩니다.

시작 단어와 종료 단어는 사용자가 D1124~D1126 에서 설정되거나 기기/장비에서 정의될 수 있습니다.

M1126, M1130, D1124~D1126 을 사용하여 시작 단어와 종료 단어를 설정하는 경우에는 RS485 통신 프로토콜의 D1120 b8~b10 은 1 로 설정되어야 합니다. 설정에 대한 사항은 아래 표를 참조하십시오.

		M1130	
		0	1
<b>M1126</b>	<b>0</b>	D1124: 사용자 정의 D1125: 사용자 정의 D1126: 사용자 정의	D1124: H 0002 D1125: H 0003 D1126: H 0000 ( 설정 없음 )
	<b>1</b>	D1124: 사용자 정의 D1125: 사용자 정의 D1126: 사용자 정의	D1124: H 003A (':') D1125: H 000D (CR) D1126: H 000A (LF)

통신 포맷 설정의 예시:

통신 포맷: 보드 레이트 (Baud rate) 9600 7, N, 2

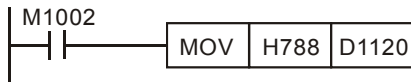
STX : “:”

ETX1 : “CR”

EXT2 : “LF”

테이블에서 D1120 을 확인함으로써 통신 포맷이 H788 임을 파악할 수 있습니다.

	b15															b0
D1120	0	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0
	0 Don't care					7	8				8					



5. STX, ETX1, ETX2 사용 시에는 특수 보조 릴레이 (special auxiliary relay) M1126 과 M1130 의 On/Off 관계에 유의하시기 바랍니다.
6. M1143: ASCII / RTU 모드 선택. “ON”은 RTU 모드, “OFF”는 ASCII 모드입니다.  
표준 MODBUS 포맷을 예를 들어 아래 표에서 설명합니다:

**ASCII 모드 (M1143=Off):**

STX	시작 단어 = ':' (3AH)
Address Hi	통신 주소: 2 개의 ASCII 코드들로 구성된 8 비트 주소
Address Lo	
Function Hi	기능 코드: 2 개의 ASCII 코드들로 구성된 8 비트 주소
Function Lo	
Data (n-1)	데이터 내용: 2n 의 ASCII 코드들로 구성된 n × 8 비트 데이터 내용
.....	
Data 0	
LRC CHK Hi	LRC 검사 합: 2 개의 ASCII 코드들로 구성된 8 비트 주소
LRC CHK Lo	
END Hi	종료 단어: END Hi = CR (0DH), END Lo = LF(0AH)
END Lo	

7. 통신 프로토콜은 MODBUS ASCII (American Standard Code for Information Interchange)코드로 구성되어 있습니다. 각각의 바이트는 2 개의 ASCII 문자들로 구성됩니다. 예를 들어, 1 바이트 데이터 64Hex 는 ASCII 로 '64'로 표시되며 '6' (36Hex)와 '4'(34Hex)로 구성됩니다. 아래의 표는 사용 가능한 16 진수 문자들과 각각에 대응하는 ASCII 코드를 표시합니다:

문자	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'
ASCII 코드	30H	31H	32H	33H	34H	35H	36H	37H

문자	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'
ASCII 코드	38H	39H	41H	42H	43H	44H	45H	46H

시작 단어 (STX): ':' (3AH)

통신 주소 (Address):

'0' '0': 모든 driver 에 동보 (Broadcast)

'0' '1': 01 주소의 drive 로

'0' 'F': 15 주소의 drive 로

'1' '0': 16 주소의 drive 로 . . . . . 같은 방식으로, 주소는 최대 255 ('F' 'F')까지 설정할 수 있음

기능 코드 (Function):

'0' '3': 레지스터들의 값을 읽음

'0' '6': 해당 레지스터에 한 단어를 기록

'1' '0': 레지스터들에 내용을 기록

데이터 내용:

사용자에 의해 전송되는 데이터의 내용

LRC check:

LRC check 는 “주소”부터 “데이터 내용”에 이르는 값을 모두 더한 값입니다. 예를 들면, the 01H + 03H + 21H +

02H + 00H + 02H = 29H, 2 의 보수를 만들면 D7H 가 됩니다.

종료 단어: END Hi = CR (0DH), END Lo = LF(0AH)

예시: drive 의 주소가 01H 로 설정되었을 때, 아래에 표시된 바와 같이 레지스터 내에 연속적으로 존재하는 2 개의 데이터 내용을 읽으십시오: 시작 레지스터 주소는 2102H 입니다.

조회 메시지 (Inquiry message):

STX	‘:’
주소	‘0’
	‘1’
기능	‘0’
	‘3’
시작 주소	‘2’
	‘1’
	‘0’
	‘2’
데이터 수 (단어 수 기준)	‘0’
	‘0’
	‘0’
	‘2’
LRC 체크	‘D’
	‘7’
종료	CR
	LF

응답 메시지 (response message):

STX	‘:’
주소	‘0’
	‘1’
기능	‘0’
	‘3’
데이터 수 (바이트 수 기준)	‘0’
	‘4’
시작 주소 2102H 의 내용	‘1’
	‘7’
	‘7’
	‘0’
주소 2103H 의 내용	‘0’
	‘0’
	‘0’
	‘0’
LRC 체크	‘7’
	‘1’
종료	CR
	LF

RTU 모드 (M1143=On):

START	아래의 설명을 참조하십시오.
Address	통신 주소: 8 비트 이진수
Function	기능 코드: 8 비트 이진수
Data (n-1)	데이터 내용: n x 8 비트 데이터
.....	
Data 0	
CRC CHK Low	CRC 체크: 2 개의 8 비트 이진수로 구성된 16 비트 CRC
CRC CHK High	
END	아래의 설명을 참조하십시오.

시작 (START):

ES/EX/SS/SA/SX/SC 시리즈: 입력 신호를 10ms 이하로 유지하십시오.

EH 시리즈:

보드 레이트(bps)	RTU 타임아웃 타이머 (ms)	보드 레이트 (bps)	RTU 타임아웃 타이머(ms)
300	40	9600	2
600	21	19200	1
1200	10	38400	1
2400	5	57600	1
4800	3	115200	1

통신 주소 (Address):

00 H: 모든 diver 에 동보 (Broadcast)

01 H: 01 주소의 drive 로



0F H: 15 주소의 drive 로

10 H: 16 주소의 drive 로 . . . . . 같은 방식으로, 주소는 최대 254 ('F' 'E')까지 설정할 수 있음

기능 코드 (Function):

'0' '3': 레지스터들의 값을 읽음

'0' '6': 해당 레지스터에 한 단어를 기록

'1' '0': 레지스터들에 내용을 기록

데이터 내용:

사용자에 의해 전송되는 데이터의 내용

CRC 체크:

CRC 체크는 “주소”에서 시작하여 “데이터 내용”에서 종료됩니다. 계산법은 아래와 같습니다.

Step 1: FFFFH 로 16 비트 레지스터를 불러옵니다. (CRC 레지스터)

Step 2: 첫 번째 8 비트 바이트 메시지 커맨드와 low 바이트의 16 비트 CRC 레지스터의 배타적 논리합(XOR) 연산을 수행하고 그 결과 값을 CRC 레지스터에 저장합니다.

Step 3: CRC 레지스터를 한 비트 오른쪽으로 이동하고 빈 자리에는 0 을 넣습니다.

Step 4: 오른쪽으로 이동된 값을 확인합니다. 만약 값이 0 이면 Step 3 의 새로운 값을 CRC 레지스터에 저장하고, 그렇지 않은 경우 A001H 와 CRC 레지스터의 배타적 논리합 연산을 수행한 후 그 결과값을 CRC 레지스터에 저장합니다.

Step 5: Step 3 과 Step 4 를 반복하여 8 비트 계산을 수행합니다.

Step 6: 모든 메시지 커맨드가 처리될 때까지 다음 8 비트 커맨드 메시지를 대상으로 Step 2~5 를 반복하여 수행합니다. 마지막으로 얻어진 CRC 레지스터 값이 CRC 체크 값이 됩니다. 메시지 커맨드의 검사 합과 CRC 체크 값은 상호 교환이 가능하도록 위치시켜야 함을 유의하시기 바랍니다.

종료 (END):

ES/EX/SS/SA/SX/SC 시리즈: 입력 신호를 10ms 이하로 유지하십시오.

EH series:

보드 레이트(bps)	RTU 타임아웃 타이머 (ms)	보드 레이트 (bps)	RTU 타임아웃 타이머 (ms)
300	40	9600	2
600	21	19200	1
1200	10	38400	1
2400	5	57600	1
4800	3	115200	1

예시: drive 의 주소가 01H 로 설정되었을 때, 아래에 표시된 바와 같이 레지스터 내에 연속적으로 존재하는 2 개의 데이터 내용을 읽으십시오: 시작 레지스터 주소는 2102H 입니다.

조회 메시지 (Inquiry message):

주소	01 H
기능	03 H
시작 데이터 주소	21 H
	02 H
데이터 수	00 H

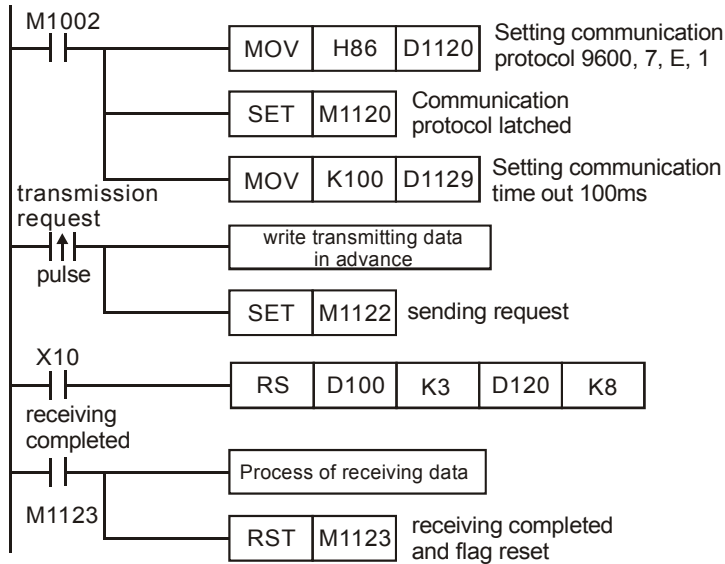
응답 메시지 (Response message):

주소	01 H
기능	03 H
데이터 수 (바이트 수 기준)	04 H
데이터 주소 8102H 의	17 H

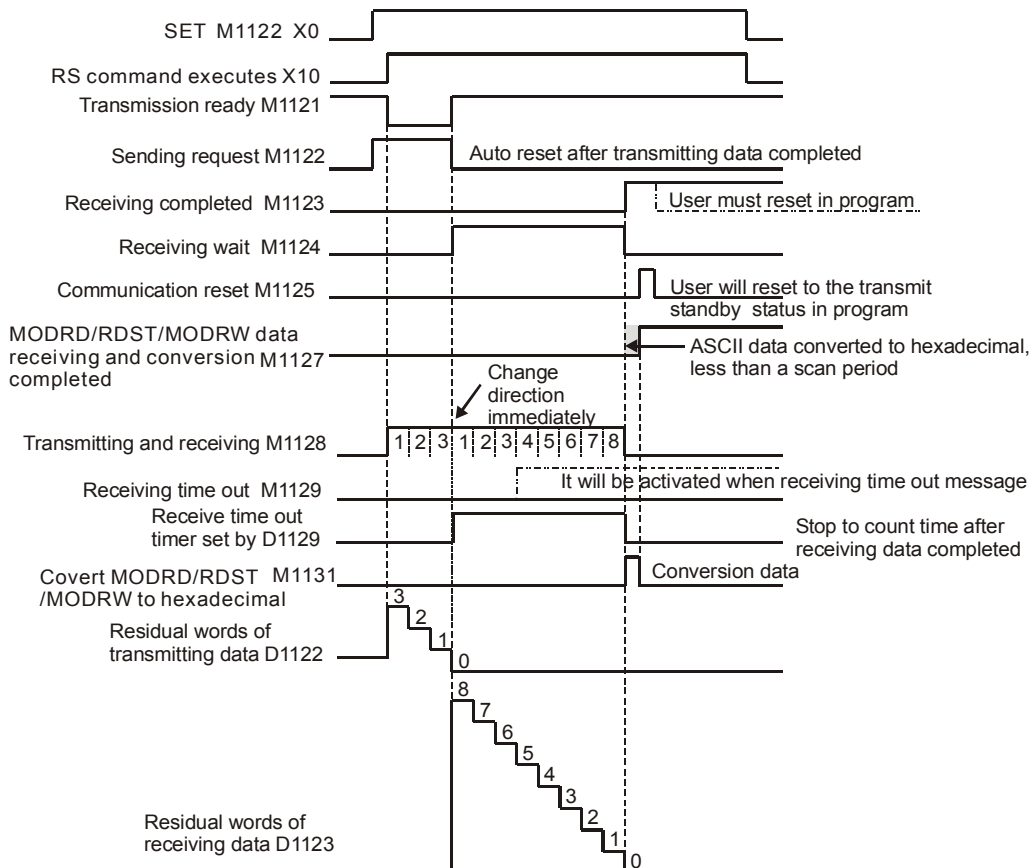
	02 H
CRC CHK Low	6F H
CRC CHK High	F7 H

내용	70 H
데이터 어드레스	00 H
8103H 의 내용	00 H
CRC CHK Low	FE H
CRC CHK High	5C H

RS-485 통신 프로그램 플래그의 타이밍 차트:



타이밍 차트:



API	연상기호 (Mnemonic)			피연산자 (Operands)		기능	컨트롤러						
							ES	EX	SS	SA	SX	SC	EH
81	D	PRUN	P	<b>(S)</b>	<b>(D)</b>	Parallel Run							

타입 피연산자	비트 장치				워드 장치											프로그램 스텝
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*		*							PRUN, PRUNP: 5 스텝 DPRUN, DPRUNP: 9 스텝
D								*	*							

펄스							16 비트							32 비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

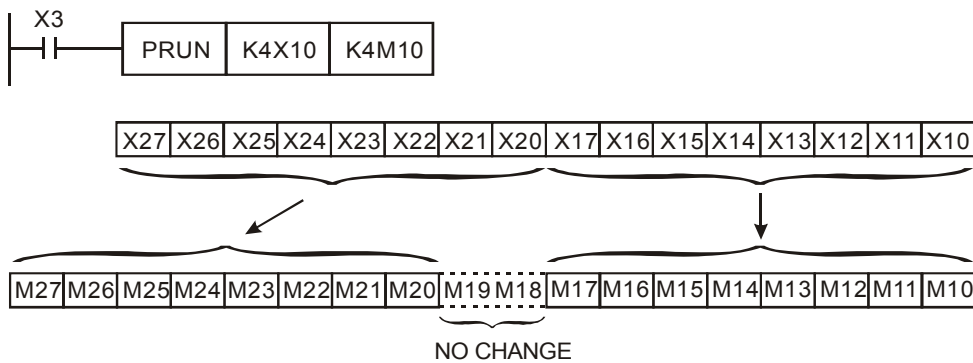
**S:** 송신 소스 디바이스 **D:** 수신 디바이스

설명:

1. 피연산자 **S** 가 KnX 를 나타내는 경우, 피연산자 **D** 는 반드시 KnM 을 나타내야 합니다.
2. 피연산자 **S** 가 KnM 을 나타내는 경우, 피연산자 **D** 는 반드시 KnY 를 나타내야 합니다.
3. **S** 의 내용을 팔진수 시스템 포맷으로 **D** 로 전송합니다.

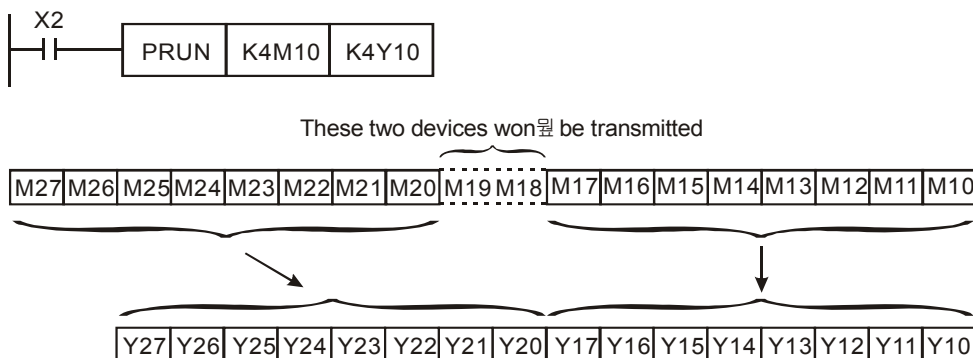
프로그램 예시 1:

X3 가 On 일 때, K4X10 의 내용을 8 진수 시스템 포맷으로 K4M10 으로 전송합니다.



프로그램 예시 2:

X2 가 On 일 때, K4M10 의 내용을 8 진수 시스템 포맷으로 K4Y10 으로 전송합니다.



API	연상기호 (Mnemonic)		피연산자 (Operands)	기능	컨트롤러						
					ES	EX	SS	SA	SX	SC	EH
82		ASCI	P	<b>(S)</b> <b>(D)</b> <b>(n)</b>	HEX 를 ASCII 로 변환						

타입 피연산자	비트 장치				워드 장치											프로그램 스텝	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ASCII, ASCIP: 7 스텝	
S					*	*	*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*											

필스							16 비트							32 비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

연산자:

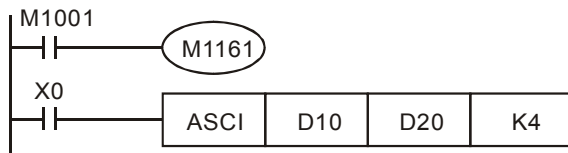
**S:** 소스 데이터 **D:** 결과 수신 디바이스 **n:** 변환할 숫자 개수 (**n=1~256**)

설명:

- 16 비트 변환 모드: M1161 이 Off 일 때, 소스 디바이스 **S** 에서 16 진수 데이터 문자 **n** 을 읽은 후 그 결과를 디바이스 **D** 의 high byte 와 low byte 에 저장합니다.
- 8 비트 변환 모드: M1161 이 On 일 때, 소스 디바이스 **S** 에서 16 진수 데이터 문자 **n** 을 읽고 ASCII 코드로 변환한 후, 그 결과를 디바이스 **D** 의 low byte 에 저장합니다. (디바이스 **D** 의 high byte 는 모두 0 으로 설정합니다.)

프로그램 예시 1:

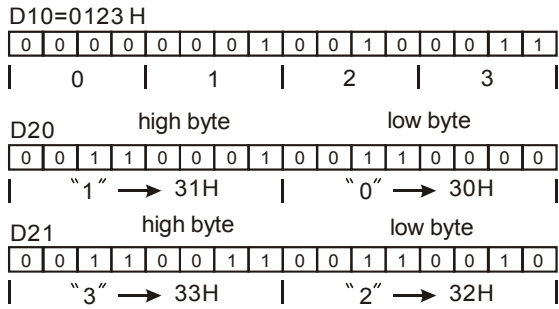
- M1161 이 Off 이면 16 비트 변환 모드입니다.
- X0 가 On 일 때, D10 으로부터 4 개의 16 진수 데이터 문자를 읽고 ASCII 코드로 변환한 후, 변환된 데이터를 D20 부터 시작하는 레지스터에 저장합니다.



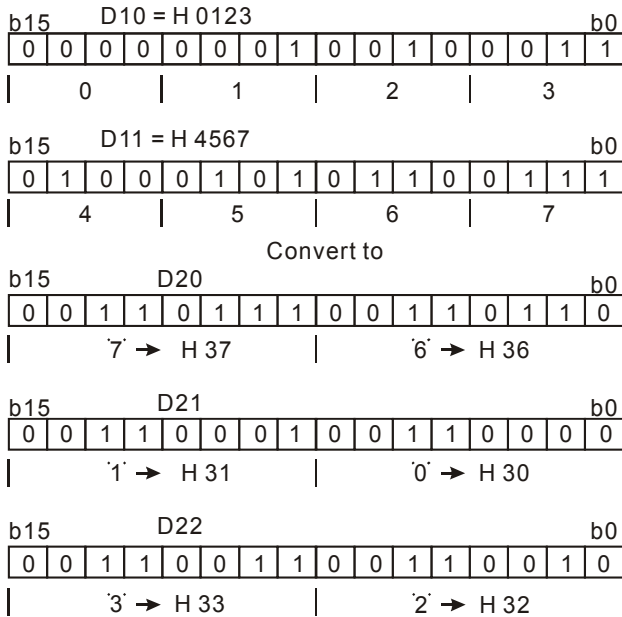
3. 가정 조건:

(D10) = 0123 H	'0' = 30H	'4' = 34H	'8' = 38H
(D11) = 4567 H	'1' = 31H	'5' = 35H	'9' = 39H
(D12) = 89AB H	'2' = 32H	'6' = 36H	'A' = 41H
(D13) = CDEFH	'3' = 33H	'7' = 37H	'B' = 42H

4. n 이 4 인 경우 비트 구조는 다음과 같습니다.



5. n 이 6 인 경우, 비트 구조는 다음과 같습니다.



6. n 이 1 에서 16 인 경우

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20 low byte	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D20 high byte		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D21 low byte			"3"	"2"	"1"	"0"	"7"	"6"
D21 high byte				"3"	"2"	"1"	"0"	"7"
D22 low byte					"3"	"2"	"1"	"0"
D22 high byte						"3"	"2"	"1"
D23 low byte							"3"	"2"
D23 high byte								"3"
D24 low byte								
D24 high byte								
D25 low byte								
D25 high byte								
D26 low byte								
D26 high byte								
D27 low byte								
D27 high byte								

변화 없음

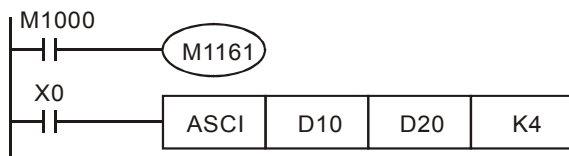
D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20 low byte	"B"	"A"	"9"	"8"	"F"	"E"	"D"	"C"
D20 high byte	"4"	"B"	"A"	"9"	"8"	"F"	"E"	"D"
D21 low byte	"5"	"4"	"B"	"A"	"9"	"8"	"F"	"E"
D21 high byte	"6"	"5"	"4"	"B"	"A"	"9"	"8"	"F"
D22 low byte	"7"	"6"	"5"	"4"	"B"	"A"	"9"	"8"

D22 high byte	"0"	"7"	"6"	"5"	"4"	"B"	"A"	"9"
D23 low byte	"1"	"0"	"7"	"6"	"5"	"4"	"B"	"A"
D23 high byte	"2"	"1"	"0"	"7"	"6"	"5"	"4"	"B"
D24 low byte	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D24 high byte		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D25 low byte			"3"	"2"	"1"	"0"	"7"	"6"
D25 high byte				"3"	"2"	"1"	"0"	"7"
D26 low byte					"3"	"2"	"1"	"0"
D26 high byte						"3"	"2"	"1"
D27 low byte							"3"	"2"
D27 high byte								"3"

변화  
없음

프로그램 예시 2:

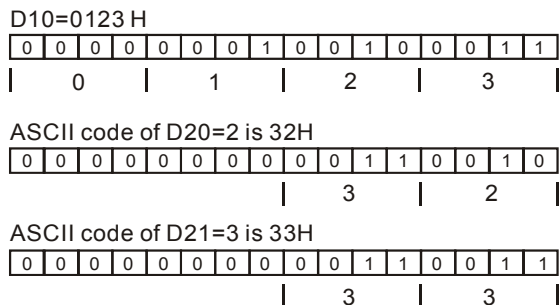
1. M1161 이 On 이면 8 비트 변환 모드입니다.
2. X0 가 On 일 때, D10 으로부터 4 개의 16 진수 데이터 문자를 읽고 ASCII 코드로 변환한 후, 변환된 데이터를 D20 부터 시작하는 레지스터에 저장합니다.



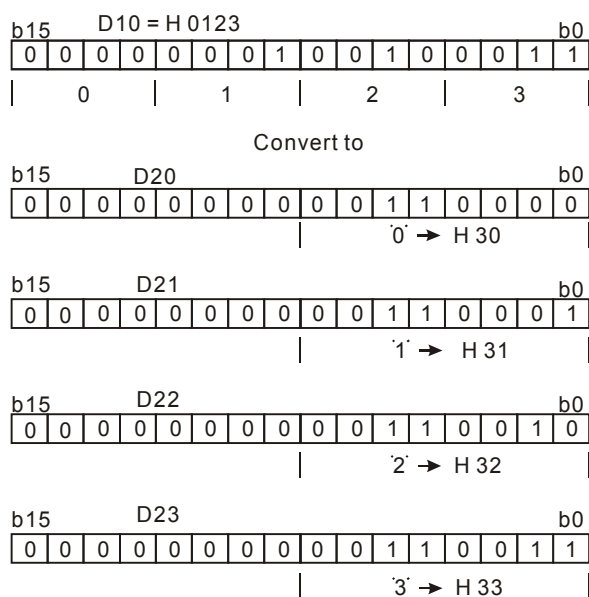
3. 가정 조건:

(D10) = 0123 H	'0' = 30H	'4' = 34H	'8' = 38H
(D11) = 4567 H	'1' = 31H	'5' = 35H	'9' = 39H
(D12) = 89AB H	'2' = 32H	'6' = 36H	'A' = 41H
(D13) = CDEFH	'3' = 33H	'7' = 37H	'B' = 42H

4. n 이 2 일 경우, 비트 구조는 아래와 같습니다.



5. n 이 4 일 경우, 비트 구조는 아래와 같습니다.



6. n=1 에서 16 인 경우:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D21		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D22			"3"	"2"	"1"	"0"	"7"	"6"
D23				"3"	"2"	"1"	"0"	"7"
D24					"3"	"2"	"1"	"0"
D25						"3"	"2"	"1"
D26							"3"	"2"
D27								"3"
D28								
D29								
D30								
D31								
D32								
D33								
D34								
D35								

변화 없음

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20	"B"	"A"	"9"	"8"	"F"	"E"	"D"	"C"
D21	"4"	"B"	"A"	"9"	"8"	"F"	"E"	"D"
D22	"5"	"4"	"B"	"A"	"9"	"8"	"F"	"E"
D23	"6"	"5"	"4"	"B"	"A"	"9"	"8"	"F"
D24	"7"	"6"	"5"	"4"	"B"	"A"	"9"	"8"
D25	"0"	"7"	"6"	"5"	"4"	"B"	"A"	"9"
D26	"1"	"0"	"7"	"6"	"5"	"4"	"B"	"A"
D27	"2"	"1"	"0"	"7"	"6"	"5"	"4"	"B"
D28	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D29		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D30			"3"	"2"	"1"	"0"	"7"	"6"
D31				"3"	"2"	"1"	"0"	"7"
D32					"3"	"2"	"1"	"0"
D33						"3"	"2"	"1"
D34							"3"	"2"
D35								"3"

변화 없음





API	연상기호 (Mnemonic)		피연산자 (Operands)	기능	컨트롤러						
					ES	EX	SS	SA	SX	SC	EH
83		HEX	P	<b>S</b> <b>D</b> <b>n</b>	ASCII 를 HEX 로 변환						

타입 피연산자	비트 장치				워드 장치											프로그램 스텝
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			HEX, HEXP: 7 스텝
D								*	*	*	*	*	*			
n					*	*										

펄스							16 비트							32 비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

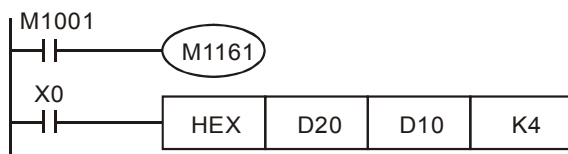
**S:** 소스 데이터 **D:** 저장된 결과 수신 디바이스 **n:** 변환할 숫자 개수 (n=1~256)

설명:

- 16 비트 변환 모드: M1161 이 Off 이면 16 비트 변환 모드입니다. S (high 와 low byte)의 16 비트 ASCII 코드를 16 진수 데이터 문자로 변환하고 한번에 4 비트씩 D 로 송신합니다. 변환된 ASCII 코드의 숫자는 n 에 의해 설정됩니다.
- 8 비트 변환 모드: M1161 이 On 이면 8 비트 변환 모드입니다. S (high 와 low byte)의 16 비트 ASCII 코드를 16 진수 데이터 문자로 변환하고 D 의 low byte 로 전송합니다. 변환된 ASCII 코드의 숫자는 n 에 의해 설정됩니다. (D 의 high byte 는 모두 0 으로 설정합니다.)

프로그램 예시 1:

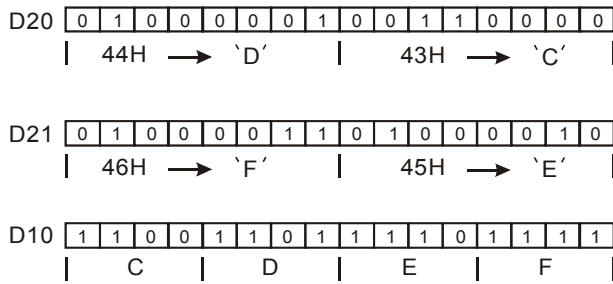
- M1161 이 Off 이면, 16 비트 변환 모드입니다.
- X0 가 On 일 때, D20 에서 시작하는 레지스터의 ASCII 바이트를 읽고 16 진수 문자로 변환한 후, D10 부터 시작하는 네 개의 레지스터에 변환된 데이터를 저장합니다. (변환된 데이터는 하나의 데이터 집합으로 변환된 4 개의 문자입니다.)



3. 가정 조건:

S	ASCII 코드	HEX 변환	S	ASCII 코드	HEX 변환
D20 low byte	H 43	"C"	D24 low byte	H 34	"4"
D20 high byte	H 44	"D"	D24 high byte	H 35	"5"
D21 low byte	H 45	"E"	D25 low byte	H 36	"6"
D21 high byte	H 46	"F"	D25 high byte	H 37	"7"
D22 low byte	H 38	"8"	D26 low byte	H 30	"0"
D22 high byte	H 39	"9"	D26 high byte	H 31	"1"
D23 low byte	H 41	"A"	D27 low byte	H 32	"2"
D23 high byte	H 42	"B"	D27 high byte	H 33	"3"

- n 이 4 인 경우, 비트 구조는 다음과 같습니다.

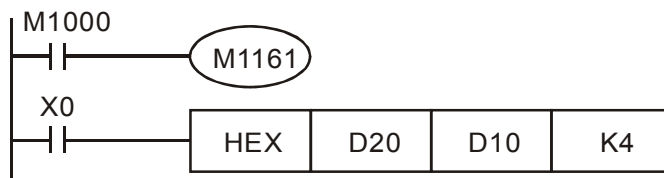


5. n 이 1 에서 16 인 경우:

D \ n	D13	D12	D11	D10
1	사용된 레지스터 중 지정되지 않은 레지스터의 값은 모두 0			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

프로그램 예시 2:

1. M1161 이 On 이면, 8 비트 변환 모드입니다.



2. 가정 조건:

S	ASCII code	HEX conversion	S	ASCII code	HEX conversion
D20	H 43	"C"	D28	H 34	"4"
D21	H 44	"D"	D29	H 35	"5"
D22	H 45	"E"	D30	H 36	"6"
D23	H 46	"F"	D31	H 37	"7"
D24	H 38	"8"	D32	H 30	"0"
D25	H 39	"9"	D33	H 31	"1"
D26	H 41	"A"	D34	H 32	"2"
D27	H 42	"B"	D35	H 33	"3"

3. n 이 2 일 때, 비트 구조는 다음과 같습니다.

D20 

								0	0	1	1	0	0	0	0
--	--	--	--	--	--	--	--	---	---	---	---	---	---	---	---

| 3
| 0

D21 

								0	1	0	0	0	0	0	1
--	--	--	--	--	--	--	--	---	---	---	---	---	---	---	---

| 4
| 1

D10 

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

| 0
| A

4. n 이 1 에서 16 일 때:

D \ n	D13	D12	D11	D10
1	사용된 레지스터 중 지정되지 않은 레지스터의 값은 모두 0			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

API	연상기호 (Mnemonic)		피연산자 (Operands)	기능	컨트롤러						
					ES	EX	SS	SA	SX	SC	EH
84		CCD	P	<b>(S)</b> <b>(D)</b> <b>(n)</b>	코드 체크						

타입 피연산자	비트 장치				워드 장치											프로그램 스텝	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CCD, CCDP: 7 스텝	
S							*	*	*	*	*	*	*				
D									*	*	*	*	*				
n					*	*							*				

펄스							16 비트							32 비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

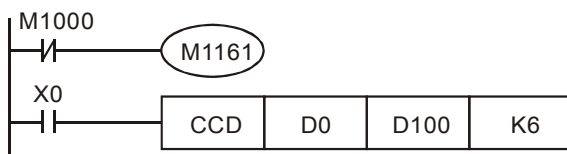
**S:** 소스 데이터 **D:** 검사 합 저장 수신 디바이스 **n:** 본 명령에서 사용된 값의 개수 (**n=1~256**)

설명:

- 본 커맨드는 통신 중 송신되는 데이터의 정합성을 보증하기 위한 검사 합을 수행하기 위해 사용됩니다.
- 16 비트 변환 모드: M1161 이 Off 이면, 16 비트 변환 모드입니다. 소스 디바이스 S 에서 지정된 레지스터로부터 n 단어 (1 바이트 안의 8 비트)의 합을 검사하여 디바이스 D 에서 지정된 레지스터에 검사합을 저장하고 패리티 비트는 D+1 에 저장합니다.
- 8 비트 변환 모드: M1161 이 On 이면, 8 비트 변환 모드입니다. 소스 디바이스 S 에서 지정된 레지스터로부터 n 단어 (1 바이트 안의 8 비트, low byte 만 유효)의 합을 검사하여 디바이스 D 에서 지정된 레지스터에 검사합을 저장하고 패리티 비트는 D+1 에 저장합니다.

프로그램 예시 1:

- M1161 이 Off 이면, 16 비트 변환 모드입니다.
- X0 가 On 일 때, D0 에서 지정된 레지스터(1 바이트 내의 8 비트, n 이 6 이면, D0~D2 지정)로부터 6 단어의 합을 검사하여 D100 에서 지정된 레지스터에 검사합을 저장하고 D101 에 패리티 비트를 저장합니다.



(S)	Content of data(words)	
D0 low byte	K100 = 0 1 1 0 0 1 0 0	
D0 high byte	K111 = 0 1 1 0 1 1 1 ① ←	
D1 low byte	K120 = 0 1 1 1 1 0 0 0	
D1 high byte	K202 = 1 1 0 0 1 0 1 0	
D2 low byte	K123 = 0 1 1 1 1 0 1 ① ←	
D2 high byte	K211 = 1 1 0 1 0 0 1 ① ←	
D100	K867	Total
D101	0 0 0 1 0 0 0 ①	

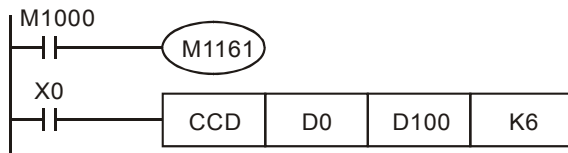
- An even result is indicated by the use of 0(zero)
- An odd result is indicated by the use of 1(one)

D100 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1

D101 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 ← Parity

프로그램 예시 2:

1. M1161 이 On 이면, 8 비트 변환 모드입니다.
2. X0 가 On 일 때, D0 에서 지정된 레지스터(1 바이트 내의 8 비트, n 이 6 이면, D0~D5 지정)로부터 6 단어의 합을 검사하여 D100 에서 지정된 레지스터에 검사합을 저장하고 D101 에 패리티 비트를 저장합니다.



(S)	Content of data(words)	
D0 low byte	K100 = 0 1 1 0 0 1 0 0	
D1 low byte	K111 = 0 1 1 0 1 1 1 ① ←	
D2 low byte	K120 = 0 1 1 1 1 0 0 0	
D3 low byte	K202 = 1 1 0 0 1 0 1 0	
D4 low byte	K123 = 0 1 1 1 1 0 1 ① ←	
D5 low byte	K211 = 1 1 0 1 0 0 1 ① ←	
D100	K867	Total
D101	0 0 0 1 0 0 0 ①	

- An even result is indicated by the use of 0(zero)
- An odd result is indicated by the use of 1(one)

D100 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1

D101 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 ← Parity

API	연상기호 (Mnemonic)		피연산자 (Operands)	기능	컨트롤러						
					ES	EX	SS	SA	SX	SC	EH
85		VRRD	P	<b>S</b> <b>D</b>	볼륨 읽기						

타입 피연산자	비트 장치				워드 장치											프로그램 스텝
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										VRRD, VRRDP: 5 스텝
D								*	*	*	*	*	*	*	*	

펄스							16 비트							32 비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

#### 피연산자

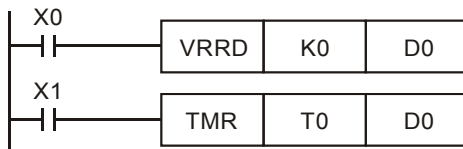
**S:** 가변 저항 번호 (Variable resistor number) (0~1)    **D:** 읽기 값 저장 디바이스

#### 설명:

1. 피연산자 S의 지정 범위: n=0~7. n=0~1 (기능 카드 없을 경우)
2. VRRD 커맨드는 No.0 과 No.1 번호가 붙는 PLC 주 연산 유닛의 두 potentiometer 를 읽거나, No.2 에서 No.7 의 번호가 붙는 기능 카드의 여섯 개의 potentiometer 를 읽기 위해 사용됩니다. 읽은 데이터 값은 0 에서 255 사이의 값으로 변환되어 저장 디바이스인 D 에 저장됩니다.
3. potentiometer 를 타이머 값 설정할 때 사용하는 경우, VR 을 조절함으로서 타이머 시간 설정을 변경할 수 있습니다. 원하는 값이 255 이상일 경우 D 에 일정 상수를 곱하십시오.

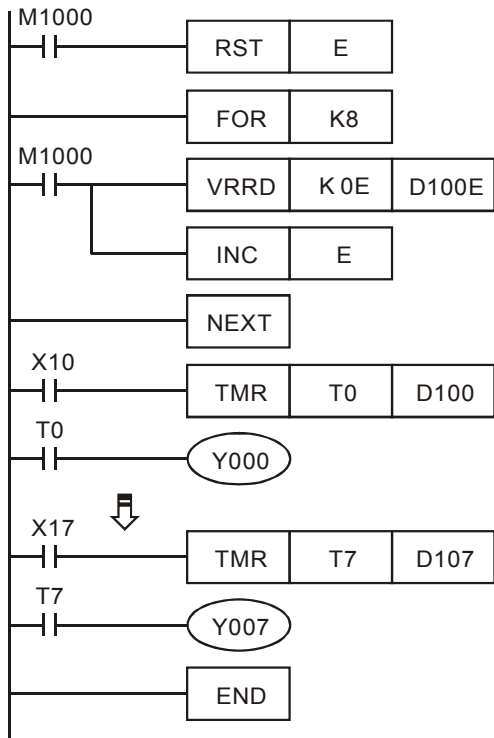
#### 프로그램 예시 1:

1. X0 가 On 이면, VRRD 커맨드에서 지정된 VR 의 NO.0 potentiometer 가 8 비트 포맷의 BIN 값 (0~255)로 변환되어 D0 에 임시로 저장됩니다.
2. X1 이 Off 이면, 타이머 T0 이 D0 의 내용을 타이머 설정 값으로 인식하고 시간 측정이 시작됩니다.



#### 프로그램 예시 2:

1. Potentiometer 읽기 순서: S의 K0 에서 K7 은 No.0 에서 No.7 의 8 개 potentiometer 에 대응합니다. 아래의 프로그램 예시는 E(E=0~7)를 이용하여 KOE (K0 에서 K7)를 변경합니다.
2. 타이머 루프는 potentiometer 의 0~10 크기를 0~255 로 변환합니다. T0 부터 T7 의 시간 단위는 0.1 초이며, 따라서 설정 값은 0 에서 25.5 초가 됩니다.



### 3. FOR~NEXT 커맨드 연산:

- FOR~NEXT 커맨드에서, K8 를 지정하는 FOR 명령어는 FOR~NEXT 명령어 사이의 루프를 8 번 반복 실행함을 의미합니다. 8 번 실행 후에는 다음 실행으로 이어집니다.
- FOR~NEXT 커맨드 사이의 (INC E)에서 E 의 내용은 0,1,2...7 이며 1 씩 증가합니다. 따라서 8 개의 potentiometer 수치는 VR0→D100, VR1→D101, VR2→D102...VR7→D107 순서로 표시되며 지정 레지스터에 기록됩니다.

#### 참고:

- VR 은 VARIABLE RESISTOR SCALE (배리스터 수치)를 의미합니다.
- For SA/SX/SC/EH 모델에 대해서는 내장된 2 포인트 VR potentiometer 가 특수 D 와 특수 M 과 함께 사용될 수 있습니다.

디바이스	기능
M1178	potentiometer VR0 시작
M1179	potentiometer VR1 시작
D1178	VR0 값
D1179	VR1 값

API	연상기호 (Mnemonic)		피연산자 (Operands)	기능	컨트롤러						
					ES	EX	SS	SA	SX	SC	EH
86		VRSC	P	<b>(S)</b> <b>(D)</b>	볼륨 scale 읽기						

타입 피연산자	비트 장치				워드 장치											프로그램 스텝
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										VRSC, VRSCP: 5 스텝
D								*	*	*	*	*	*	*	*	

펄스							16 비트							32 비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

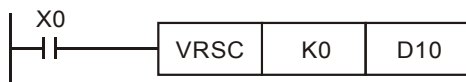
**S:** 배리스터 (Variable resistor) 넘버 (0~1) **D:** scale 값 수신 디바이스

설명:

1. 피연산자 S의 지정값: 기능 카드 사용 시 n=0~7, 기능 카드 없을 시 n=0~1
2. VRRD 커맨드는 No.0 과 No.1 번호가 붙는 PLC 주 연산 유닛 내 두 potentiometer의 scale 값을 읽거나, No.2에서 No.7의 번호가 붙는 기능 카드의 여섯 개 potentiometer의 scale 값을 읽기 위해 사용됩니다. (potentiometer scale 값은 0에서 10입니다.) 읽은 데이터 값은 0에서 255 사이의 정수값으로 저장 디바이스인 D에 저장됩니다.

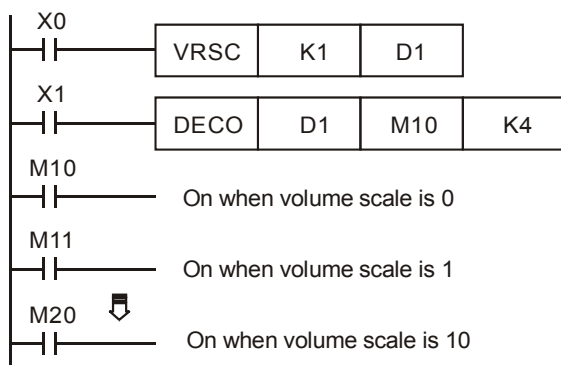
프로그램 예시 1:

X0가 On일 때, VRSC 커맨드에 의해 지정된 No.0 potentiometer scale 값(0에서 10)이 디바이스 D10에 저장됩니다.



프로그램 예시 2:

1. 디지털 스위치로 사용: potentiometer scale 0에서 10을 대응시키십시오. M10부터 M20 중 하나의 접점만 On입니다. DECO 커맨드 (API 41)을 사용하면 potentiometer scale을 M0~M15에 디코드할 수 있습니다.
2. X0가 On일 때, D1에 No.1 potentiometer scale 값 (0에서 10)을 저장하십시오.
3. X1이 On일 때, DECO 커맨드 (API 41)을 사용하여 potentiometer scale을 M10~M25로 디코드하십시오.
4. MPU에 VR 확장 카드를 삽입하지 않은 상태에서 VRRD 혹은 VRSC 커맨드를 이용한 프로그램 상에 potentiometer 넘버를 K2~K7로 설정한 경우, 구문 실행 시에 에러가 발생합니다.





API	연상기호 (Mnemonic)			피연산자 (Operands)	기능	컨트롤러						
						ES	EX	SS	SA	SX	SC	EH
87	D	ABS	P	<b>(D)</b>	절대값 (Absolute Value)							

타입 피연산자	비트 장치				워드 장치										프로그램 스텝	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	DABS, DABSP: 5 스텝

펄스							16 비트							32 비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

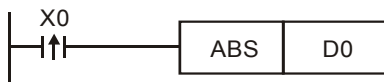
**D:** 절대값의 소스 및 수신 디바이스

설명:

1. 본 명령어가 실행되면, 지정 디바이스 D 의 절대값을 취합니다.
2. 본 명령어는 일반적으로 pulse execution (ABSP)의 형태를 갖습니다.

프로그램 예시:

X0 가 OFF 에서 ON 이 될 때, D0 내용의 절대값을 취합니다.



API	연상기호 (Mnemonic)		피연산자 (Operands)				기능	컨트롤러						
								ES	EX	SS	SA	SX	SC	EH
88	D	PID			<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	<b>(S<sub>3</sub>)</b>	<b>(D)</b>	PID 연산					

타입 피연산자	비트 장치				워드 장치											프로그램 스텝	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PID : 9 스텝	
S <sub>1</sub>													*			DPID: 17 스텝	
S <sub>2</sub>													*				
S <sub>3</sub>													*				
D													*				

펄스							16 비트							32 비트						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

#### 피연산자:

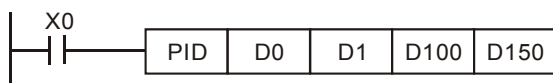
**S<sub>1</sub>**: 목표 값 (SV)    **S<sub>2</sub>**: 현재 값 (PV)    **S<sub>3</sub>**: 파라미터 (16 비트 경우에는 20 개의 연속된 디바이스를, 32 비트 경우에는 21 개의 연속된 디바이스를 사용하십시오.) **D**: 출력 값 (MV)

#### 설명:

- PID 연산 제어에 지정된 커맨드입니다. 이 명령어는 샘플링 타임에 도달하면 PID 연산을 실행합니다. PID 는 비례, 적분 및 미분 (Proportion, Integration and Differential)을 의미합니다. PID 제어는 기계 장치, 공기압 장치, 전자 장치들의 여러가지 용도에 폭넓게 활용되는 제어 방식입니다.
- S<sub>1</sub>**: 목표 값 (SV), **S<sub>2</sub>**: 현재 측정 값 (PV), **S<sub>3</sub>~S<sub>3</sub>+19** (16 비트 커맨드), **S<sub>3</sub>~S<sub>3</sub>+20** (32 비트 커맨드): PID 커맨드는 파라미터 세팅이 모두 끝난 후에 실행되며 결과는 D 에 저장됩니다. D 내용에 래치 레지스터 (latch register) 영역을 설정하지 마십시오. (D 내용에 래치 레지스터를 설정하길 원하신다면, 프로그램이 실행될 때에는 래치를 0 으로 리셋하십시오.)

#### 프로그램 예시:

- PID 커맨드를 수행하기 전 모든 파라미터 세팅을 완료하십시오.
- X0 가 ON 일 때 본 커맨드가 실행되며 그 결과는 D150 에 저장됩니다. X0 가 OFF 인 경우 본 명령어가 실행되지 않으며 이전 데이터는 변경되지 않습니다.



#### 참고:

- PID 커맨드는 V.5.7 이상, ES/EX/SS 이상의 모델에만 제공되며 다른 버전에는 제공되지 않습니다.
- PID 커맨드에 시간 제한은 없으나, **S<sub>3</sub>** 에 의해 지정된 레지스터 넘버는 반복될 수 없습니다.
- 16 비트 커맨드에서는 **S<sub>3</sub>** 가 20 개의 레지스터를 사용합니다. 위의 프로그램에서, **S<sub>3</sub>**가 지정한 PID 커맨드의 파라미터 설정 영역은 D100~D114 입니다. PID 커맨드를 실행하기 전, 사용자는 MOV 커맨드를 이용하여 설정을 지정된 레지스터에 전송하여야 합니다. 파라미터가 지정한 레지스터가 래치(latch) 영역이면 MOVP 명령어를 이용하여 전송을 수행하십시오.
- 16 비트 **S<sub>3</sub>**의 파라미터 표:

디바이스 넘버	기능	설정 영역	설명
------------	----	-------	----

디바이스 넘버	기능	설정 영역	설명
<b>S<sub>3</sub>:</b>	샘플링 시간(T <sub>S</sub> ) (단위: 10ms)	1~2,000 (단위: 10ms)	T <sub>S</sub> 가 한번의 프로그램 스캔 시간보다 작을 경우, PID 커맨드는 한번의 프로그램 스캔 시간을 실행합니다. T <sub>S</sub> =0 인 경우, PID 커맨드는 활성화되지 않습니다.
<b>S<sub>3</sub>+1:</b>	비례 이득 (K <sub>P</sub> )	0~30,000(%)	설정 값이 30,000 을 초과할 경우, 30,000 으로 설정됩니다.
<b>S<sub>3</sub>+2:</b>	적분 이득 (K <sub>I</sub> )	0~30,000(%)	
<b>S<sub>3</sub>+3:</b>	미분 이득 (K <sub>D</sub> )	-30,000~30,000(%)	
<b>S<sub>3</sub>+4:</b>	제어 방법 (Dir)	0: 정상 제어 1: 정방향 제어 (E=SV-PV) 2: 역방향 제어 (E=PV-SV) 3 : 온도에 대한 파라미터 오토튜닝 기능. 오토튜닝이 종료되면 <b>S<sub>3</sub>+4</b> 는 자동적으로 K <sub>P</sub> , K <sub>I</sub> , K <sub>D</sub> 의 최적 파라미터와 함께 K <sub>4</sub> 로 지정됩니다. (32 비트 커맨드는 본 기능을 지원하지 않습니다.) 4 : 온도 조절 제어를 위한 특수 기 (32 비트 커맨드는 본 기능을 지원하지 않습니다.)	
<b>S<sub>3</sub>+5:</b>	에러 값 (E)이 유효하지 않은 범위	0~32,767	예를 들어, 에러 값(E) 범위가 5 라면, E 의 출력값 MV 는 -5 에서 5 범위 내에서 0 입니다.
<b>S<sub>3</sub>+6:</b>	포화출력 (MV) 상계	-32,768~32,767	예를 들어, 상계(upper bound)가 1000 으로 설정되었을 때, 출력값(MV)가 1000 보다 크면 출력값이 1000 으로 제한됩니다. (상계는 하계보다 커야합니다. <b>S<sub>3</sub>+6 &gt; S<sub>3</sub>+7.</b> )
<b>S<sub>3</sub>+7:</b>	포화출력 (MV) 하계	-32,768~32,767	마찬가지로, 하계(lower bound)가 - 1000 으로 설정되었을 때 출력값(MV)가 -1000 보다 작으면 출력값이 -1000 으로 제한됩니다.
<b>S<sub>3</sub>+8:</b>	포화적분 상계	-32,768~32,767	예를 들어, 상계(upper bound)가 1000 으로 설정되었을 때, 출력값(MV)가 1000 보다 크면 출력값이 1000 으로 제한되며 적분이 이루어지지 않습니다. (상계는 하계보다 커야합니다. <b>S<sub>3</sub>+8 &gt; S<sub>3</sub>+9.</b> )
<b>S<sub>3</sub>+9:</b>	포화적분 하계	-32,768~32,767	마찬가지로, 하계(lower bound)가 - 1000 으로 설정되었을 때 출력값(MV)가 -1000 보다 작으면 출력값이 -1000 으로 제한되며 적분이 이루어지지 않습니다.
<b>S<sub>3</sub>+10,11:</b>	누적 적분값 임시 저장	32 비트 부동 소수점 범위 (32-bit floating point range)	누적 적분값입니다. 참조를 위해 사용되지만 사용자의 필요에 의해 삭제하거나 변경할 수 있습니다. (변경은 32 비트 부동 소수점으로 수행되어야 합니다.)
<b>S<sub>3</sub>+12:</b>	이전 PV 값 임시 저장	—	참조를 위한 현재 측정값입니다. 사용자의 필요에 의해 변경될 수 있습니다.

디바이스 넘버	기능	설정 영역	설명
<b>S<sub>3</sub> +13:</b> ? <b>S<sub>3</sub> +19:</b>	시스템 용도용. 사용자 사용 불가.		

- 파라미터 설정 값이 설정 범위를 벗어나는 경우, 상계나 하계값으로 설정됩니다. 다만 제어 방법이 범위 밖의 값을 가지는 경우, 0으로 설정됩니다.
- PID 커맨드는 서브루틴 인터럽트, 스텝 포인트, CJ 커맨드에 사용될 수 있습니다.
- 에러 타임  $T_s$ 의 최대 범위는 - (스캔 타임+1ms) ~+ (스캔 타임) 입니다. 만약 에러 값이 출력에 영향을 줄 경우, 스캔 타임을 고정시키거나 PID 커맨드를 타이머의 인터럽트 서브루틴에서 실행하십시오.
- 샘플링 타임  $T_s$ 의 설정값이 스캔 타임보다 작거나 같은 경우, CPU가 에러 코드 (K6740: PID operation error)를 표시합니다. 이 때, CPU는 PID 피연산자의 실행을 위해  $T_s$ 를 스캔 타임으로 재설정합니다. 이러한 상황이 발생할 경우, PID 커맨드를 인터럽트 서브루틴(I6□□~I8□□)에서 실행하시기 바랍니다.
- PID 커맨드 실행 이전에는 현재 측정 값(PV)이 안정적인 값을 가져야 합니다. PID 연산을 수행하기 위해 DVP-04AD / DVP-04XA / DVP-04PT / DVP-04TC 모듈의 입력 값을 사용할 경우, 상기 모듈들의 A/D 변환 시간에 주의를 기울이시기 바랍니다.
- 32 비트 커맨드 **S<sub>3</sub>**는 21개의 레지스터를 점유합니다. **S<sub>3</sub>**에 의해 지정된 PID의 파라미터 설정 영역이 D100~D120 일 경우, PID 커맨드 실행 이전에 MOV 커맨드를 이용하여 설정값을 지정된 레지스터로 전송해야 합니다.
- 32 비트 **S<sub>3</sub>** 파라미터 표:

디바이스 넘버	기능	설정 범위	설명
<b>S<sub>3</sub> :</b>	샘플링 타임(T <sub>S</sub> ) (단위: 10ms)	1~2,000 (단위: 10ms)	T <sub>S</sub> 가 한번의 프로그램 스캔 시간보다 작을 경우, PID 커맨드는 한번의 프로그램 스캔 시간을 실행합니다. T <sub>S</sub> =0 인 경우, PID 커맨드는 활성화되지 않습니다.
<b>S<sub>3</sub> +1:</b>	비례 이득 (K <sub>P</sub> )	0~30,000(%)	설정 값이 30,000 을 초과할 경우, 30,000 으로 설정됩니다.
<b>S<sub>3</sub> +2:</b>	적분 이득 (K <sub>I</sub> )	0~30,000(%)	
<b>S<sub>3</sub> +3:</b>	미분 이득 (K <sub>D</sub> )	-30,000~30,000(%)	
<b>S<sub>3</sub> +4:</b>	제어 방법 (Dir)	0: 정상 제어 1: 정방향 제어 (E=SV-PV) 2: 역방향 제어 (E=PV-SV) 3 : 온도에 대한 파라미터 오토튜닝 기능. 오토튜닝이 종료되면 <b>S<sub>3</sub> +4</b> 는 자동적으로 KP, KI, KD 의 최적 파라미터와 함께 K4 로 지정됩니다. (32 비트 커맨드는 본 기능을 지원하지 않습니다.) 4 : 온도 조절 제어를 위한 특수 기 (32 비트 커맨드는 본 기능을 지원하지 않습니다.)	
<b>S<sub>3</sub> +5, 6:</b>	32 비트 에러 값(E)이 유효하지 않은 범위	0~2,147,483,647	예를 들어, 에러 값(E) 범위가 5 라면, E 의 출력값 MV 는 - 5 에서 5 범위 내에서 0 입니다.
<b>S<sub>3</sub> +7, 8:</b>	32 비트 포화 출력	-2,147,483,648~	예를 들어, 상계(upper

	(MV) 상계	2,147,483,647	bound)가 1000 으로 설정되었을 때, 출력값(MV)가 1000 보다 크면 출력값이 1000 으로 제한됩니다. (상계는 하계보다 커야합니다. $S_3+7, 8 > S_3+9, 10$ .)
$S_3 +9, 10$ :	32 비트 포화 출력(MV) 하계	-2,147,483,648~ 2,147,483,647	마찬가지로, 하계(lower bound)가 -1000 으로 설정되었을 때 출력값(MV)가 -1000 보다 작으면 출력값이 -1000 으로 제한됩니다.
$S_3 +11, 12$ :	32 비트 포화 적분 상계	-2,147,483,648~ 2,147,483,647	예를 들어, 상계(upper bound)가 1000 으로 설정되었을 때, 출력값(MV)가 1000 보다 크면 출력값이 1000 으로 제한되며 적분이 이루어지지 않습니다. (상계는 하계보다 커야합니다. $S_3+11, 12 > S_3+13, 14$ .)
$S_3 +13, 14$ :	32 비트 포화 적분 하계	-2,147,483,648~ 2,147,483,647	마찬가지로, 하계(lower bound)가 -1000 으로 설정되었을 때 출력값(MV)가 -1000 보다 작으면 출력값이 -1000 으로 제한되며 적분이 이루어지지 않습니다.
$S_3 +15, 16$ :	32 비트 누적 적분값 임시 저장	32 비트 부동 소수점 범위 (32-bit floating point range)	누적 적분값입니다. 참조를 위해 사용되지만 사용자의 필요에 의해 삭제하거나 변경할 수 있습니다. (변경은 32 비트 부동 소수점으로 수행되어야 합니다.)
$S_3 +17, 18$ :	32 비트 이전 PV 값 임시 저장	—	참조를 위한 현재 측정값입니다. 사용자의 필요에 의해 변경될 수 있습니다.
$S_3 +19$ : ? $S_3 +20$ :	시스템 용도용. 사용자 사용 불가.		

12. 32 비트  $S_3$  와 16 비트  $S_3$  의 설명은 대체적으로 유사하며  $S_3+5 \sim S_3+20$  의 용량(capacity)에서 차이가 있습니다.

#### PID 수식:

- 이 커맨드는 측정 값의 속도와 미분 종류에 따라 PID 연산을 수행합니다.
- PID 연산은 3 개의 제어 방법을 갖습니다: 정상(normal), 정방향(forward), 역방향(inverse) 제어. 제어 방법은  $S_3 +4$  에 의해 설정됩니다. 그 밖의 PID 연산과 관련 있는 설정은  $S_3 \sim S_3 +5$  에 의해 설정됩니다.
- PID 수식:

$$MV = K_p * E(t) + K_I * E(t) \frac{1}{S} + K_D * PV(t)S$$

제어 방법	PID 수식
정방향 제어, 정상 제어	$E(t) = SV - PV$

역방향 제어	$E(t) = PV - SV$
--------	------------------

참고:  $PV(t)S$  는  $PV(t)$  의 미분값을 의미하며  $E(t)\frac{1}{S}$  는  $E(t)$ 의 적분값을 의미합니다.

위의 수식을 통해 본 커맨드가 일반적인 PID 커맨드와 다르다는 것을 알 수 있습니다. 차이점은 미분 방식의 변화에 있습니다. 맨처음 일반적 PID 커맨드를 실행할 때 과도미분값 (transient different value)이 너무 크게 되는 경우를 방지하기 위해, 본 커맨드는 현재 측정 값(PV)의 미분값에 대한 모니터링을 통해 현재 측정 값(PV)의 변화가 너무 클 경우 출력 값(MV)을 감소시킵니다.

#### 4. 기호 설명:

$MV$	: 출력 값
$K_p$	: 비례 이득
$E(t)$	: 에러 값. 정방향 제어 $E(t) = SV - PV$ , 역방향 제어 $E(t) = PV - SV$
$PV$	: 현재 측정 값
$SV$	: 목표 값
$K_D$	: 미분 이득
$PV(t)S$	: $PV(t)$ 의 미분 값
$K_I$	: 적분 이득
$E(t)\frac{1}{S}$	: $E(t)$ 의 적분 값

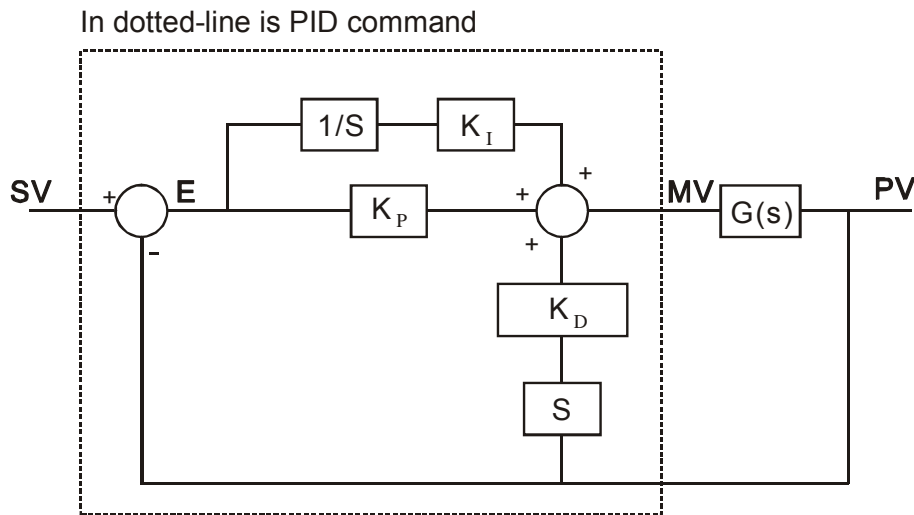
#### 5. 온도 제어 수식:

S3 +4 가 K3 과 K4 일 때, 다이어그램 II 의 내부 PID 수식은 아래와 같습니다.

$$MV = \frac{1}{K_p} \left[ E(t) + \frac{1}{K_I} \left( E(t) \frac{1}{S} \right) + K_D * PV(t)S \right], \text{ where } E(t) = SV - PV$$

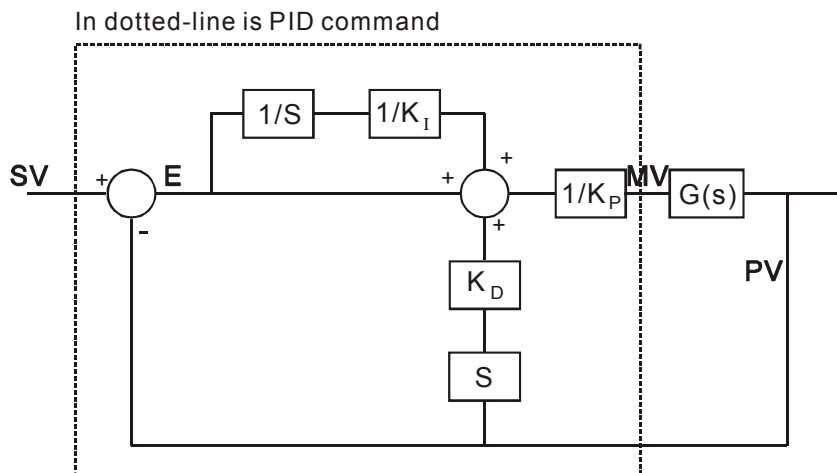
이 함수는 온도 제어에 특화된 구조를 가지고 있습니다. 샘플링타임(TS)이 4 초 (K400)으로 설정된 경우, MV 의 출력 범위가 K0~K4000 이며 대응하는 GPWM 커맨드가 4 초 (K4000)으로 설정되어야 함을 의미합니다. 만약 사용자가 온도 제어를 위해 **S<sub>3</sub>** 의 각각 파라미터를 어떻게 설정해야 할지 모른다면, 오토-튜닝을 위해 **S<sub>3</sub> +4** 를 K3 로 설정하십시오. 오토-튜닝이 종료되면 (제어 방법은 K4 로 맞추어질 것입니다.) 사용자는 제어 결과를 통해 **S<sub>3</sub>** 의 파라미터를 향상시킬 수 있습니다.

#### 6. 제어 다이어그램:



다이어그램 I:  $S_3+4$  가 제어 방법 K0~K2 으로 설정되어 있습니다.

다이어그램 I 에서, S 는 미분이며, 그 정의는 현재 PV 값에서 이전 PV 값을 뺀 결과를 샘플링 타임으로 나눈 것이 됩니다.  $1/S$  는 적분이며, 그 정의는 이전 적분 값에 현재 차이 값을 더하고 그 결과를 샘플링 타임으로 나눈 것이 됩니다.  $G(s)$  는 제어되는 디바이스를 의미합니다.



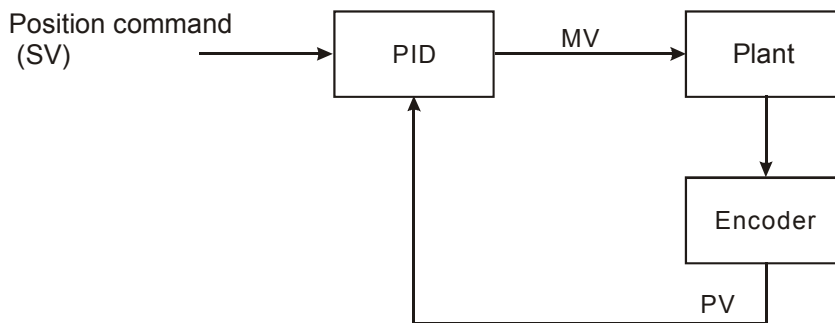
다이어그램 II:  $S_3+4$  가 제어 방법 K3~K4 로 설정되어 있습니다.

다이어그램 II 에서, 기호  $1/K_I$  과  $1/K_P$  는 각각  $K_I$  과  $K_P$  로 나뉘어지는 함수를 의미합니다. 이 다이어그램이 온도 제어를 위해 특화된 PID 커맨드를 보여주고 있기 때문에, 사용자는 GPWM 커맨드를 이 제어 방법과 함께 사용해야 합니다. 예시는 응용 방법 3 을 참고하십시오.

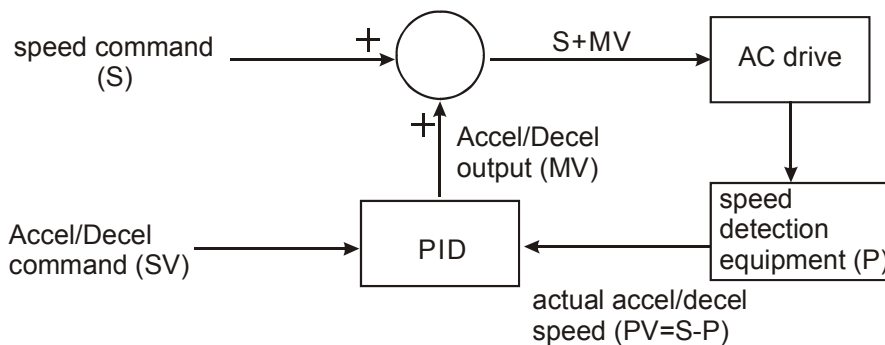
#### 7. 참고 및 제언:

- 디바이스  $S_3+6 \sim S_3+13$  은 SA/SX/SC/EH 시리즈와 ES 시리즈 (v5.7 이상) 모델에만 적용됩니다.
- ES 시리즈 (v5.6 이하)에서 PID 커맨드는 한 번만 사용할 수 있습니다. EX(v5.7 이상)/SA/SX/SC/EH 시리즈에서는 PID 커맨드 이용에 제한이 없습니다.
- EX, SA/SX/SC (v1.1 이하), 그리고 EH (v1.0 이하) 시리즈 모델에서  $S_3+3$  의 파라미터 가능 범위는 0~300,000 입니다.
- PID 커맨드는 다양한 환경에 유연하게 적용할 수 있습니다. 따라서 사용 환경에 따라 제어 함수를 주의 깊게 정의하시기 바랍니다. 예를 들어, 온도 제어를 위해 파라미터  $S_3+4=K3$  을 설정한 경우, 이 함수를 모터 제어에 사용하게 되면 장치의 손상이 올 수 있습니다.

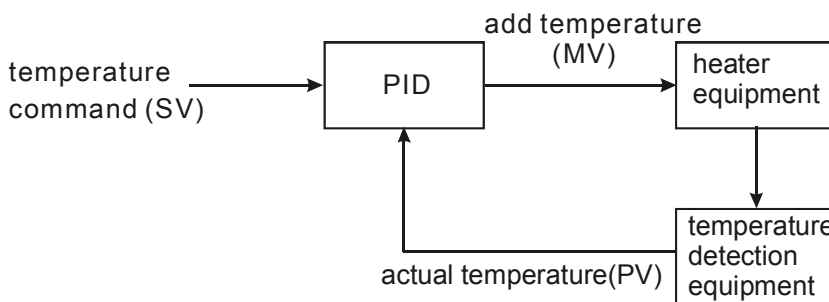
- e) 3 대 주요 파라미터인  $K_P$ ,  $K_I$ ,  $K_D$ ( $S_3+4=K_0\sim K_2$ )를 조절할 때에는,  $K_P$ (경험에 의한 설정)를 먼저 조절하고  $K_I$ 와  $K_D$ 를 0으로 설정하십시오. 제어를 위한 조정 시에는  $K_I$ (작은 쪽부터 큰 쪽으로)를 조정하고  $K_D$ (작은 쪽부터 큰 쪽으로)를 조정하십시오. 조정에 대한 자세한 사항은 예시 4를 참조하십시오.  $K_P$ 가 100이면 100%를 의미합니다.  $K_P$ 가 100%보다 작은 경우, 에러 값이 감소하며  $K_P$ 가 100%보다 큰 경우 에러 값은 증폭됩니다.
- f) 사용자가 온도 제어를 위해 특화된 함수를 사용할 경우 ( $S_3+4=K_3$  or  $K_4$ ), 전력 절단에 의한 파라미터 손실을 예방하기 위해 래치 타입의 레지스터 D에 조정된 파라미터 값을 저장하시는 것이 바람직합니다. 오토-튜닝 이후의 파라미터 값이 모든 환경 조건에 유효하지는 않습니다. 따라서, 사용자는 필요에 따라 오토-튜닝 이후에 파라미터 값을 변경할 수 있으나, 사용자가  $K_1$ 과  $K_D$  외의 다른 값을 수정하는 것은 바람직하지 않습니다.
- g) 본 커맨드는 다양한 파라미터로 제어되어야 합니다. 오류 방지를 위해 아래의 설정법을 따르십시오.  
 예시 1: 위치 제어를 위한 PID 커맨드 회로 다이어그램 (제어 방법  $S_3+4$ 는 0으로 설정되어야 합니다.)



예시 2: 속도 제어를 위한 PID 커맨드 회로 다이어그램 (제어 방법  $S_3+4$ 는 0으로 설정되어야 합니다.)



예시 3: 온도 제어를 위한 PID 커맨드 회로 다이어그램 (작용 지시  $S_3+4$ 는 1로 설정되어야 합니다.)



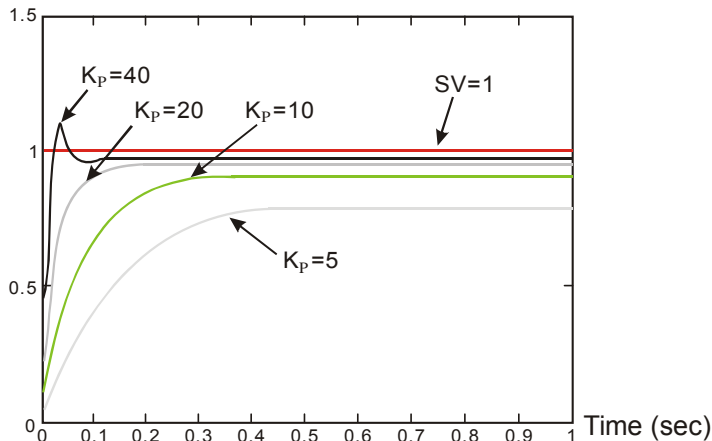
예시 4: PID 조정에 대한 단계 예시



제어 시스템 내 산업 기계의 이동 함수  $G(s) = \frac{b}{s+a}$  를 가정합니다. (대부분의 일반적인 AC 드라이브

모델은 이 함수를 따릅니다.) 커맨드 값 SV 는 1 이고, 샘플링 타임 Ts 는 10ms 입니다. 단계 예시는 아래와 같습니다:

**Step1:** 먼저  $K_I$  과  $K_D$  을 0 으로 설정하고,  $K_P$  를 5, 10, 20, 40 차례대로 설정하여 (SV)와 (PV) 값을 기록합니다. 그 결과는 아래 그림과 같이 나타납니다.



**Step 2:** 위의 그림에서  $K_P$  가 10 이 되는 지점을 선택할 수 있습니다. 그 이유는 아래와 같습니다:

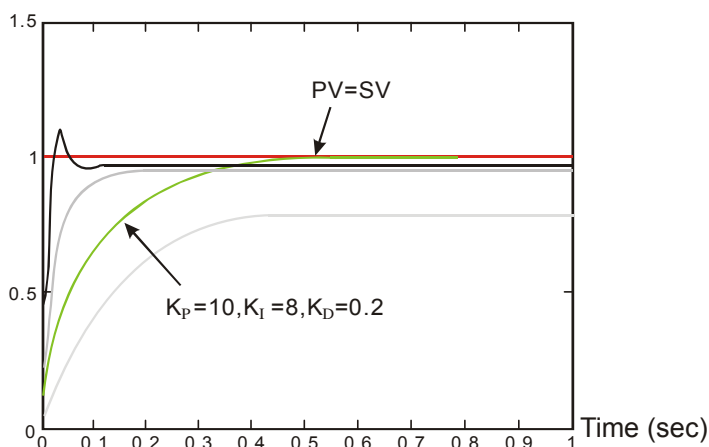
$K_P$  가 40 일 때는 응답치가 목표를 넘어서서 오버슈트가 발생하기 때문에 사용할 수 없습니다.

$K_P$  가 20 일 때에는 PV 응답치가 SV 에 근접하고, 오버슈트가 발생하지 않지만, 너무 빠른 스타트업 때문에 과도 MV 가 커지게 되므로 사용할 수 없습니다.

$K_P$  가 10 일 때, PV 응답치가 SV 에 가깝고 부드러운 곡선을 보여줍니다. 따라서 이 값을 사용합니다.

$K_P$  가 5 일 때, 응답이 너무 늦어 사용할 수 없습니다.

**Step 3:**  $K_P=10$  곡선을 사용하기로 결정한 후,  $K_I$  을 작은 쪽에서 큰 쪽으로 순서대로 (예를 들어 1,2,4,8 순으로) 배열하되  $K_P$  값보다 작게 유지하십시오. 그리고  $K_D$  를 작은 쪽에서 큰 쪽으로 순서대로 (예를 들어 0.01, 0.05, 0.1, 0.2 순으로) 배열하되  $K_P$  의 10%보다 작게 유지하십시오. 최종적으로 사용자는 아래 그림과 같은 PV 와 SV 관계에 대한 그래프를 얻을 수 있습니다.



**Note:** 이 예시는 참고로만 사용하십시오. 사용자는 실제 제어 시스템에 따른 적절한 제어 파라미터를 설정해야 합니다.

## 응용 예시:

응용 예시 1: PID 커맨드를 이용한 압력 제어 시스템 (예시 1의 회로도 사용)

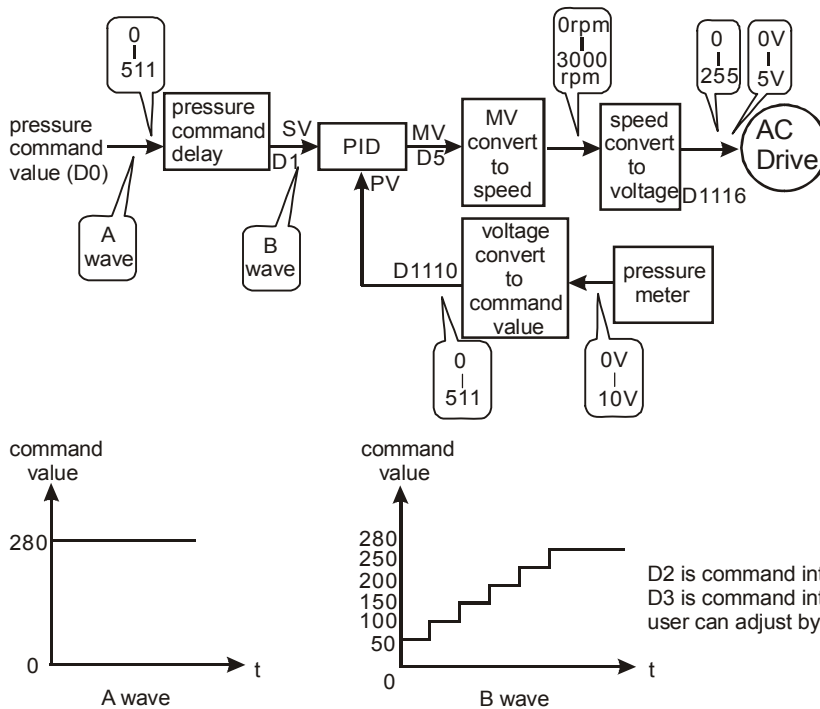
제어 목적: 제어 시스템이 목표 압력 값에 도달하도록 합니다.

제어 특성: 이 시스템은 제어 목표에 점진적으로 도달해야 합니다. 따라서 제어 목적에 너무 빠르게 도달하고자 한다면 시스템 고장 혹은 과부하가 발생할 수 있습니다.

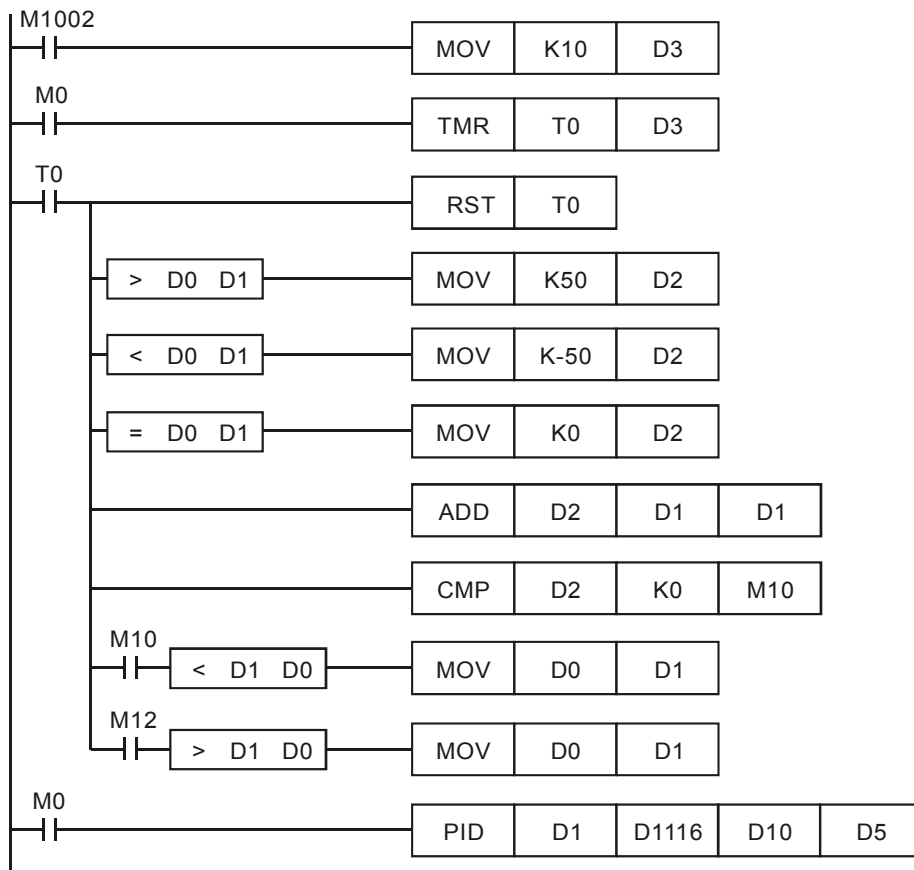
바람직한 해결 방법

방법 1: 샘플링 타임을 길게 잡으십시오.

방법 2: 딜레이 커맨드를 사용하십시오. 딜레이 커맨드의 제어 회로도는 아래와 같습니다.



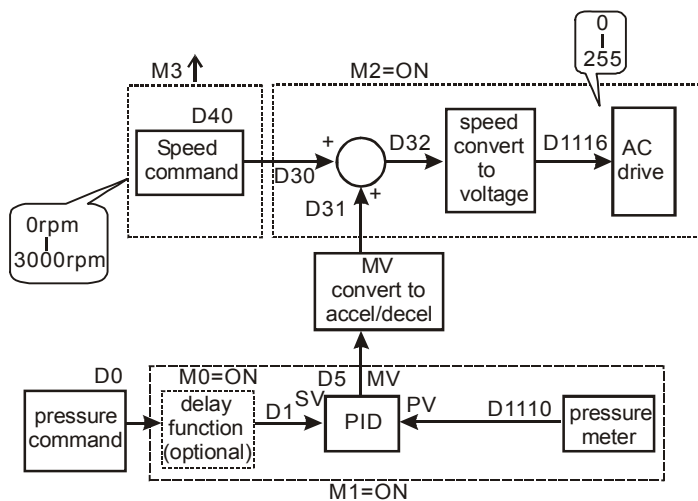
커맨드 딜레이에 대한 프로그램 어플리케이션은 아래와 같습니다:



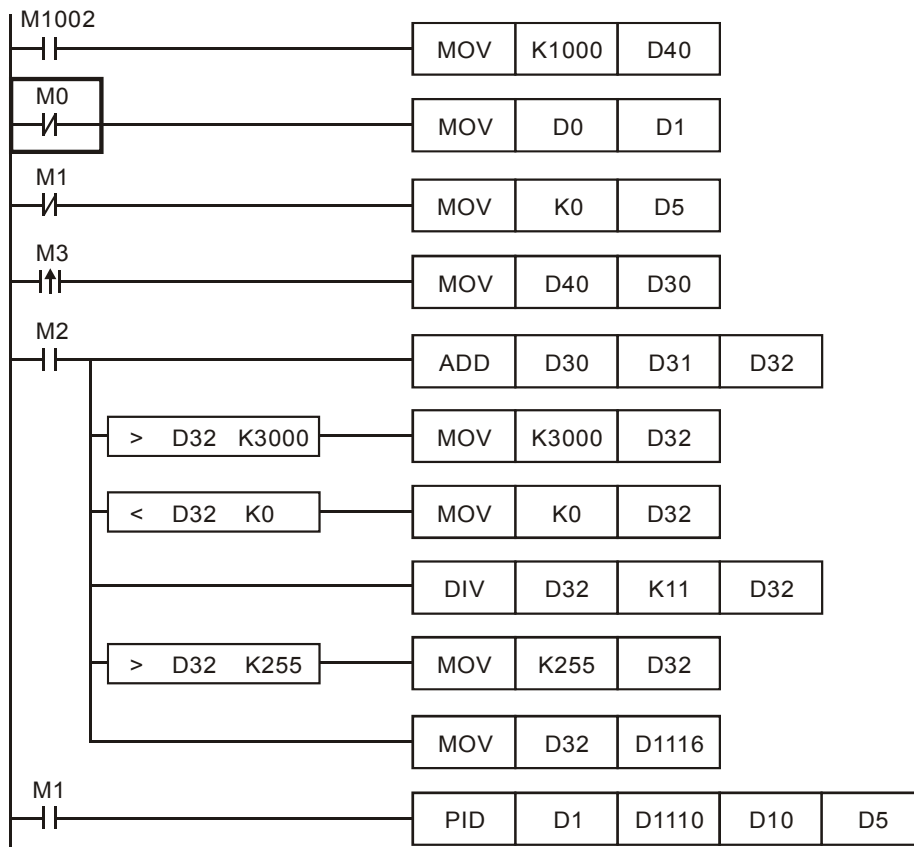
응용 예시 2: 속도 제어와 압력 제어의 개별 제어 (예시 2의 회로도를 사용하십시오.)

제어 목적: 일정 시간 동안 개회로(open loop)를 이용하여 속도를 조절한 후에 압력 조절을 위해 압력 조절 시스템 (PID 커맨드)를 추가

제어 특성: 이 아키텍처는 속도와 압력 두 시스템 간 아무런 연관성이 없기 때문에 개회로(open loop)를 사용하여 속도 제어 목적을 달성한 후, 폐회로 압력 제어를 사용하여 압력 제어 목적을 달성합니다. 응용 예시 1의 커맨드 딜레이 함수를 사용하여 압력 제어 시스템이 너무 빠르게 변화하는 것을 예방할 수 있습니다. 제어 회로도는 아래와 같습니다:



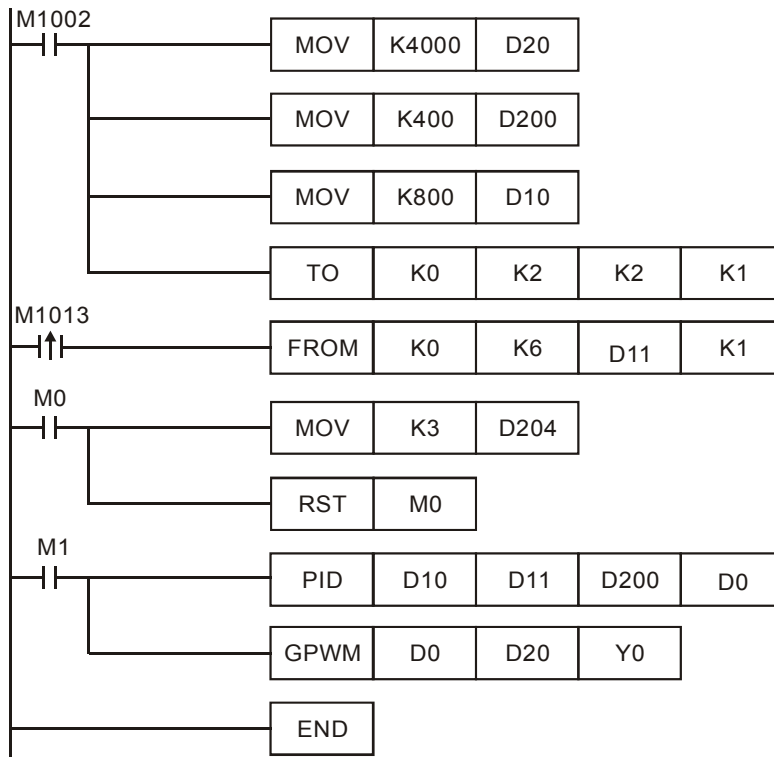
부분 프로그램 어플리케이션은 아래와 같습니다:



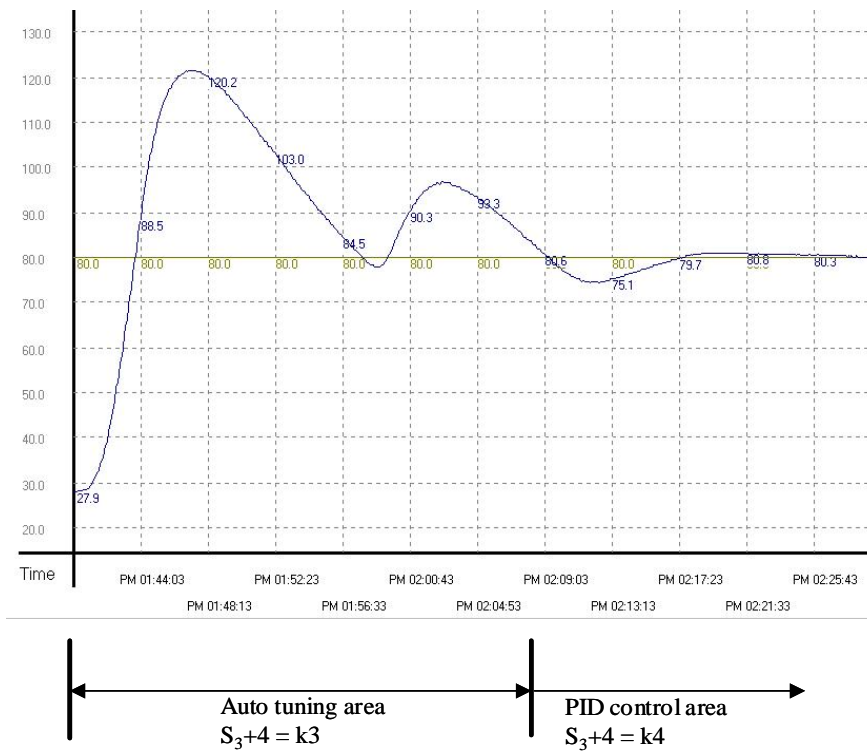
응용 예시 3: 온도 제어를 위한 오토-튜닝 사용

제어 목적: 온도 제어를 위한 PID 최적 파라미터 계산

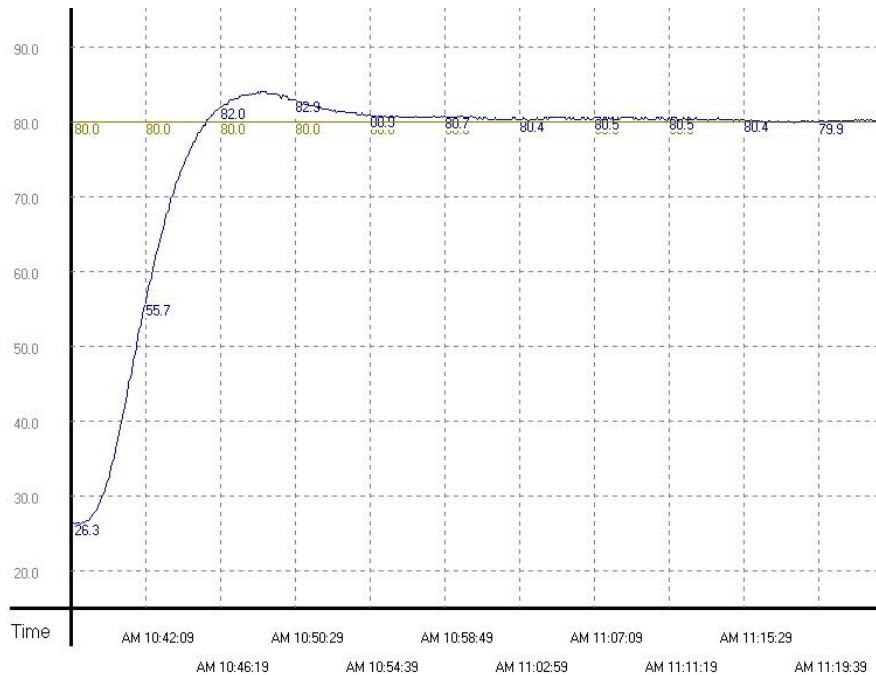
제어 특성: 일반 사용자의 경우 처음에는 온도 제어의 환경 특성에 익숙하지 않을 수 있으므로, 초기 설정을 위해서 오토-튜닝 기능( $S_3 + 4 = K3$ )을 사용할 수 있습니다. 설정이 끝난 후에는 특정 온도 제어( $S_3 + 4 = K4$ )를 위한 제어 기능을 자동으로 변경합니다. 실제 응용에 있어서의 프로그램 예시는 아래와 같습니다:



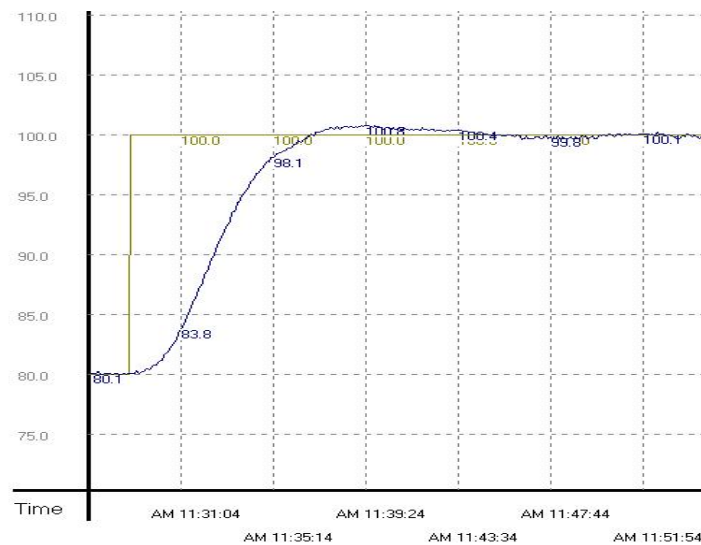
오토-튜닝의 실험 결과는 아래와 같습니다:



오토-튜닝 이후 특정 온도 제어의 파라미터를 이용한 실험 결과는 아래와 같습니다:



위의 그래프를 보면, 오토-튜닝 이후의 온도 제어 상태가 더 향상되고 안정되었음과 함께, 오토-튜닝에 단지 20 분 정도의 시간만이 소요된 것을 명확히 알 수 있습니다. 또 다른 케이스로 목표 온도를 80° 에서 100° 로 변환하는 상황을 들 수 있으며, 그 결과는 아래와 같습니다.



위의 그래프를 보면, 80° 에서 설정된 파라미터가 온도가 100°에 도달하더라도 여전히 유효함과 함께, 제어에 이르기까지 시간도 크게 소요되지 않음을 알 수 있습니다.



API	간략기호	피연산자	기능	제어기						
100	MODRD	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(n)</b>	Modbus 데이터 읽기	E S	E X	S S	S A	S X	S C	E H

연산자	유형	비트 장치				워드 장치										프로그램 단계	
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MODRD: 7 단계
S <sub>1</sub>						*	*							*			
S <sub>2</sub>						*	*							*			
n						*	*							*			

피연산자:

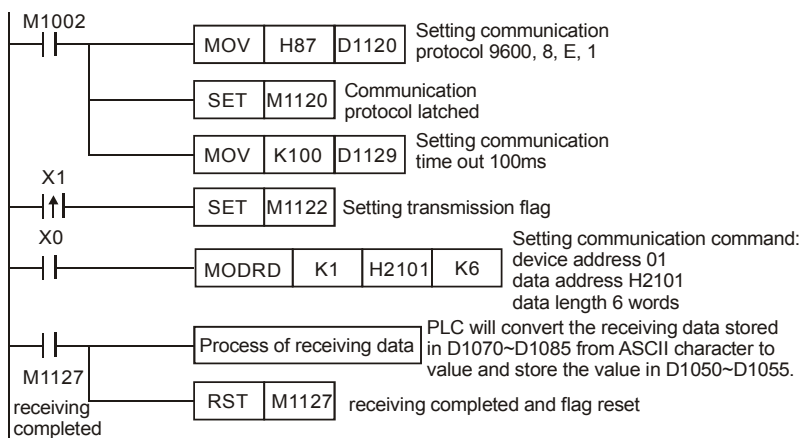
**S<sub>1</sub>**: 통신 주소. (K0~K254) **S<sub>2</sub>**: 데이터 **n** 을 읽기 위한 주소: 데이터 읽기 길이. ( $K1 \leq n \leq K6$ )

설명:

- ES 시리즈 모델은 인덱스 레지스터 E 와 F 를 지원하지 않습니다.
- MODRD 는 MODBUS ASCII 모드와 RTU 모드 통신을 위한 특정 명령어 입니다. 델타 VFD 시리즈 드라이브 전기종(VFD-A 시리즈는 제외)에 탑재되어 있는 RS-485 통신은 MODBUS 통신 기능을 가집니다. 그러므로 MODRD 명령어는 델타 VFD 시리즈 AC 드라이브로부터 통신데이터를 읽기 위해서 사용되어질 수 있습니다. 좀 더 상세한 내용을 원하시면 델타 VFD 시리즈 매뉴얼을 참조하시기 바랍니다.
- S<sub>2</sub>** 는 데이터를 읽기 위한 주소 입니다. 만약 그 주소가 잘못 지정되어 있으면, 사용자는 오류 메시지를 받습니다. M1141 가 ON 으로 변경되는 동안 그 오류 코드는 D1130 에 저장 됩니다.
- 주변 장치로부터의 피드백데이터(반환값)는 D1070 부터 D1085 에 저장되어 집니다.  
피드백데이터(반환값)의 수신이 완료된 후 PLC 는 모든 피드백데이터(반환값)들이 정확한지를 체크 합니다. 오류가 있을 경우 M1140 가 ON 이 됩니다.
- ASCII 모드를 사용하는 경우, 피드백데이터(반환값)들이 모두 ASCII 문자들이므로 PLC 는 그 데이터들을 변환하여 D1050 부터 D1055 에 저장합니다. 만일 RTU 모드를 사용할 경우 D1050 부터 D1055 는 사용되어 지지 않습니다.
- M1140 혹은 M1141 이 On 된 후, 한개의 정정데이터가 주변장치로 전송 되어 집니다. 만일 피드백데이터(반환값)들이 모두 맞다면, M1140 및 M1141 의 플래그는 지워집니다.

프로그램 예제 1:

PLC 와 VFD-S 시리즈 AC 드라이브간의 통신 (ASCII 모드, M1143=Off)





PLC ⇨ VFD-S, PLC 전송: “01 03 2101 0006 D4”

VFD-S ⇨ PLC, PLC 수신: “01 03 0C 0100 1766 0000 0000 0136 0000 3B”

PLC 전송 데이터 레지스터(전송 메시지)

레지스터	데이터			
D1089 하위	'0'	30 H	ADR 1	ADR (1,0) 는 AC 드라이브 주소
D1089 상위	'1'	31 H	ADR 0	
D1090 하위	'0'	30 H	CMD 1	CMD (1,0) 는 명령어 코드
D1090 상위	'3'	33 H	CMD 0	
D1091 하위	'2'	32 H	데이터 시작 주소	
D1091 상위	'1'	31 H		
D1092 하위	'0'	30 H		
D1092 상위	'1'	31 H		
D1093 하위	'0'	30 H	데이터수 (word 단위)	
D1093 상위	'0'	30 H		
D1094 하위	'0'	30 H		
D1094 상위	'6'	36 H		
D1095 하위	'D'	44 H	LRC CHK 1	LRC CHK (0,1) 는 오류 체크 코드
D1095 상위	'4'	34 H	LRC CHK 0	

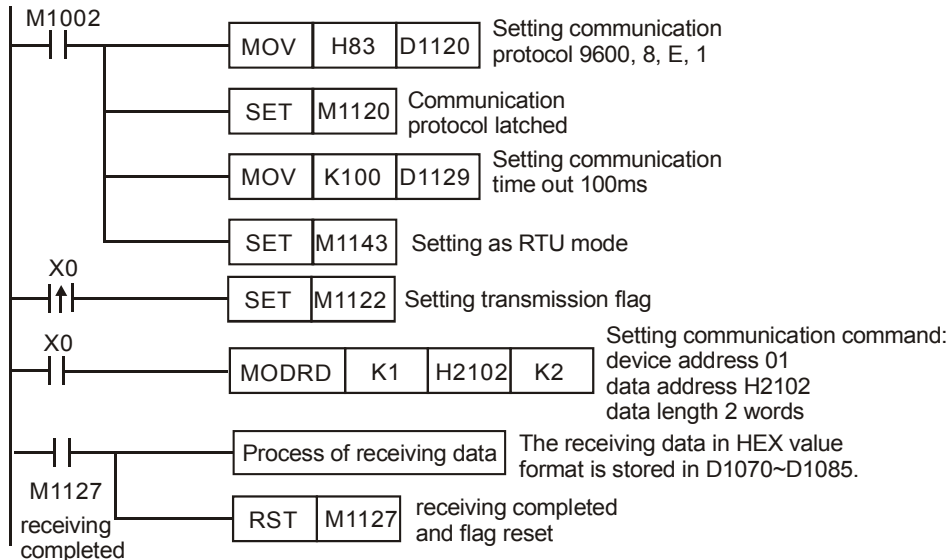
PLC 수신 데이터 레지스터 (응답 메시지)

레지스터	데이터			
D1070 하위	‘0’	30 H	ADR 1	
D1070 상위	‘1’	31 H	ADR 0	
D1071 하위	‘0’	30 H	CMD 1	
D1071 상위	‘3’	33 H	CMD 0	
D1072 하위	‘0’	30 H	데이터수 (byte 단위)	
D1072 상위	‘C’	43 H		
D1073 하위	‘0’	30 H	주소 2101 H 의 내용	PLC 는 자동으로 ASCII 코드를 변환하여 그 변환된 값을 D1050 = 0100 H 에 저장 합니다.
D1073 상위	‘1’	31 H		
D1074 하위	‘0’	30 H		
D1074 상위	‘0’	30 H		
D1075 하위	‘1’	31 H	주소 2102 H 의 내용	PLC 는 자동으로 ASCII 코드를 변환하여 그 변환된 값을 D1051 = 1766 H 에저장 합니다.
D1075 상위	‘7’	37 H		
D1076 하위	‘6’	36 H		
D1076 상위	‘6’	36 H		
D1077 하위	‘0’	30 H	주소 2103 H 의 내용	PLC 는 자동으로 ASCII 코드를 변환하여 그 변환된 값을 D1052 = 0000 H 에저장 합니다.
D1077 상위	‘0’	30 H		
D1078 하위	‘0’	30 H		
D1078 상위	‘0’	30 H		
D1079 하위	‘0’	30 H	주소 2104 H 의 내용	PLC 는 자동으로 ASCII 코드를 변환하여 그 변환된 값을 D1053 = 0000 H 에저장 합니다.
D1079 상위	‘0’	30 H		
D1080 하위	‘0’	30 H		
D1080 상위	‘0’	30 H		
D1081 하위	‘0’	30 H	주소 2105 H 의 내용	PLC 는 자동으로 ASCII 코드를 변환하여 그 변환된 값을 D1054 = 0136 H 에저장 합니다.
D1081 상위	‘1’	31 H		
D1082 하위	‘3’	33 H		
D1082 상위	‘6’	36 H		
D1083 하위	‘0’	30 H	주소 2106 H 의 내용	PLC 는 자동으로 ASCII 코드를 변환하여 그 변환된 값을 D1055 = 0000 H 에저장 합니다.
D1083 상위	‘0’	30 H		
D1084 하위	‘0’	30 H		
D1084 상위	‘0’	30 H		

D1085 하위	'3'	33 H	LRC CHK 1
D1085 상위	'B'	42 H	LRC CHK 0

## 프로그램 예제 2:

PLC 와 VFD-S 시리즈 AC 드라이브간의 통신 (RTU 모드, M1143= On)



PLC ⇨ VFD-S, PLC 전송: **01 03 2102 0002 6F F7**

VFD-S ⇨ PLC, PLC 수신: **01 03 04 1700 0000 FE 5C**

PLC 전송 데이터 레지스터(전송 메시지)

레지스터	데이터	
D1089 하위	01 H	주소
D1090 하위	03 H	기능
D1091 하위	21 H	데이터 시작 주소
D1092 하위	02 H	
D1093 하위	00 H	데이터수 (word 단위)
D1094 하위	02 H	
D1095 하위	6F H	CRC CHK 하위
D1096 하위	F7 H	CRC CHK 상위

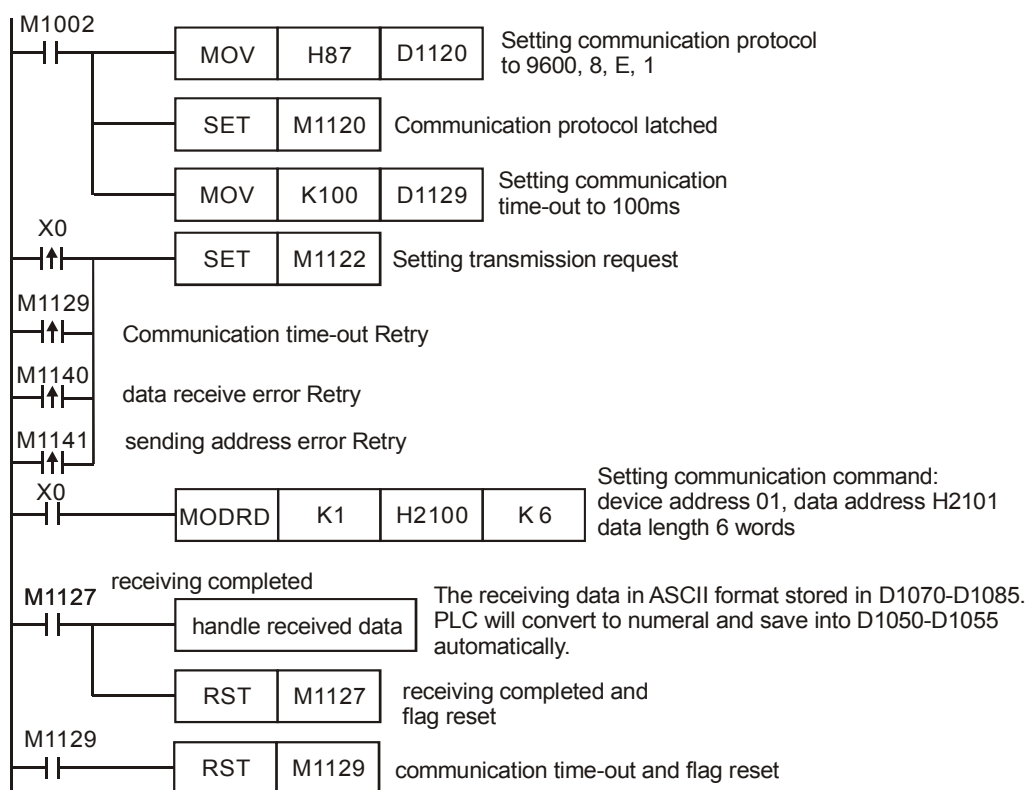
PLC 수신 데이터 레지스터 (응답 메시지)

레지스터	데이터	
D1070 하위	01 H	주소
D1071 하위	03 H	기능
D1072 하위	04 H	데이터수 (byte 단위)
D1073 하위	17 H	주소 2102 H 의 내용
D1074 하위	70 H	
D1075 하위	00 H	주소 2103 H 의 내용
D1076 하위	00 H	
D1077 하위	FE H	CRC CHK 하위
D1078 하위	5C H	CRC CHK 상위

## 프로그램 예제 3:

1. PLC 를 VFD-S 시리즈 AC 드라이브에 연결합니다. (ASCII 모드, M1143=Off) 통신시간경과된 후, 데이터 수신 혹은 주소 전송 과정에서 오류가 발생시 다시 시작합니다.

2. X0=On 일때, 장치 01 (VFD-S)의 주소 H2100 으로부터 데이터를 읽고 ASCII 포맷으로 D1070~D1085 에 저장시키기 바랍니다. PLC 는 자동으로 그것의 내용들을 숫자로 변환하여 D1050~D1055 에 저장 할 것입니다.
3. 통신이 시간경과될때 플래그 M1129 가 On 이 되고 프로그램은 M1129 로부터의 요청을 전달하여 M1122 가 다시 데이터를 읽도록 요청합니다.
4. 오류신호를 받았을 경우 플래그 M1140 가 On 이 되고 프로그램은 M1140 로부터의 요청을 전달하여 M1122 가 다시 데이터를 읽도록 요청합니다.
5. 주소오류를 받았을 경우 플래그 M1141 가 On 이 되고 프로그램은 M1141 로부터의 요청을 전달하여 M1122 가 다시 데이터를 읽도록 요청합니다.
- 6.



#### 주의:

LDP, ANDP, ORP 등의 상승접점(Rising-edge contact)과 LDF, ANDF, ORF 등의 하향접점(falling-edge contact)들은 API 100 MODRD, API 105 RDST, API 150 MODRW (기능코드 H03)등의 세가지 명령어를 사용하기전에 사용되어 질 수 없습니다. 그렇지 않으면, 전달받은 레지스터에 저장되어 있는 데이터들은 부정확하게 됩니다.

API	간략기호	피연산자	기능	제어기						
101	MODWR	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(n)</b>	Modbus 데이터 쓰기	E S	E X	S S	S A	S X	S C	E H

연산자 \ 유형	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MODWR: 7 단계	
S <sub>1</sub>					*	*							*				
S <sub>2</sub>					*	*							*				
n					*	*							*				

피연산자:

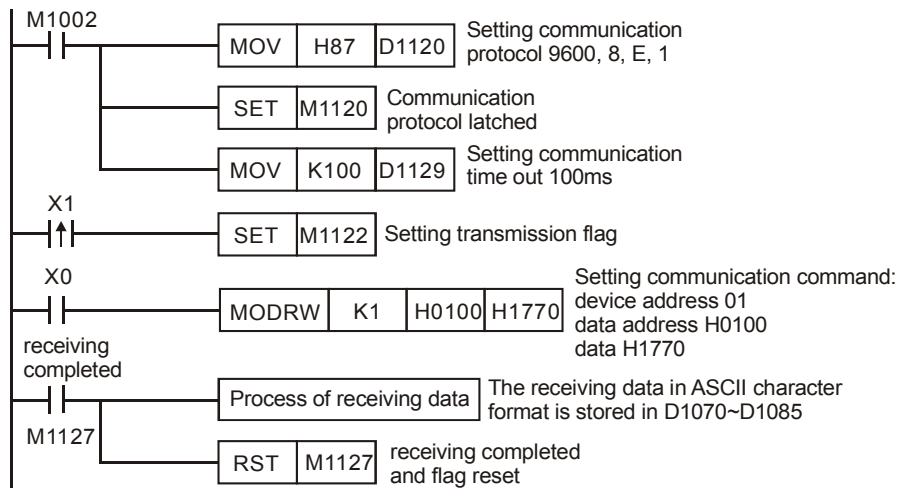
**S<sub>1</sub>**: 통신 주소, K0~K254 **S<sub>2</sub>**: 데이터 **n** 을 쓰기 위한 주소: 데이터쓰기

설명:

- ES 시리즈 모델은 인덱스 레지스터 E 와 F 를 지원하지 않습니다.
- MODWR 는 MODBUS ASCII 모드와 RTU 모드 통신을 위한 특정 명령어 입니다. 델타 VFD 시리즈 드라이브 전기종(VFD-A 시리즈는 제외)에 탑재되어 있는 RS-485 통신은 MODBUS 통신 기능을 가집니다. 그러므로 MODWR 명령어는 델타 VFD 시리즈 AC 드라이브로부터 통신데이터를 읽기 위해서 사용되어질 수 있습니다. 좀 더 상세한 내용을 원하시면 델타 VFD 시리즈 매뉴얼을 참조하시기 바랍니다.
- S<sub>2</sub>** 는 데이터를 읽기 위한 주소 입니다. 만약 그 주소가 잘못 지정되어 있으면, 사용자는 오류 메시지를 받습니다. M1141 가 ON 으로 변경되는 동안 그 오류 코드는 D1130 에 저장 됩니다. 예를 들면, VFD-S 에 대해 4000H 가 잘못된 주소라면, M1141 는 ON 이 되고 D1130=2 가 됩니다. 오류 코드에 대한 정보는 VFD-S 시리즈 매뉴얼을 참고하시기 바랍니다.
- 주변 장치로부터의 피드백데이터(반환값)는 D1070 부터 D1085 에 저장되어 집니다. 피드백데이터(반환값)의 수신이 완료된 후 PLC 는 모든 피드백데이터(반환값)들이 정확한지를 체크 합니다. 오류가 있을 경우 M1140 가 ON 이 됩니다.
- M1140 혹은 M1141 이 On 된 후, 한개의 정정데이터가 주변장치로 전송 되어 집니다. 만일 피드백데이터(반환값)들이 모두 맞다면, M1140 및 M1141 의 플래그는 지워집니다.

프로그램 예제 1:

PLC 와 VFD-S 시리즈 AC 드라이브간의 통신(ASCII 모드, M1143= Off)



PLC ⇨ VFD-B, PLC 전송: “ 01 06 0100 1770 71 ”

VFD-B ⇨ PLC, PLC 수신: “ 01 06 0100 1770 71 ”

PLC 전송 데이터 레지스터(전송 메시지)

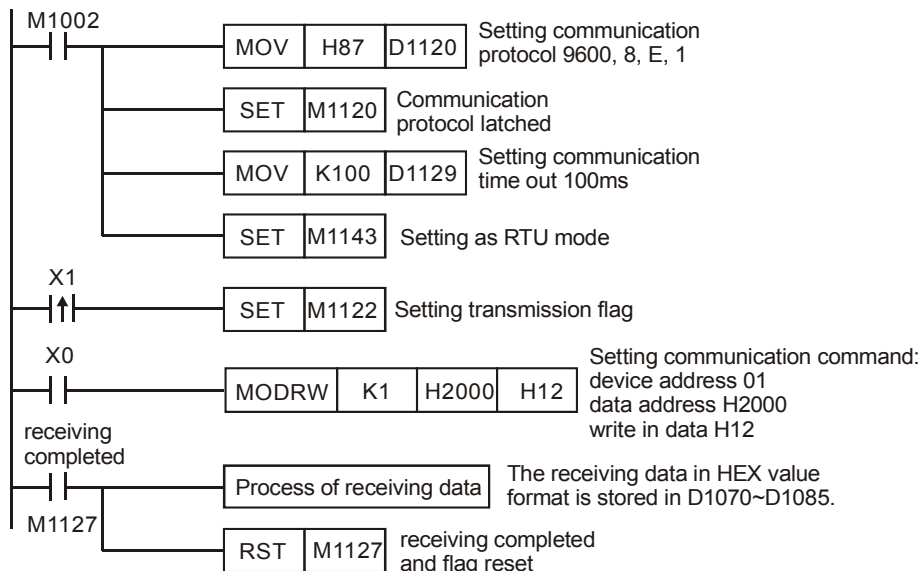
레지스터	데이터			
D1089 하위	‘0’	30 H	ADR 1	ADR (1,0) 는 AC 드라이브 주소
D1089 상위	‘1’	31 H	ADR 0	
D1090 하위	‘0’	30 H	CMD 1	CMD (1,0) 는 명령어 코드
D1090 상위	‘6’	36 H	CMD 0	
D1091 하위	‘0’	30 H	데이터주소	
D1091 상위	‘1’	31 H		
D1092 하위	‘0’	30 H		
D1092 상위	‘0’	30 H		
D1093 하위	‘1’	31 H	데이터내용	
D1093 상위	‘7’	37 H		
D1094 하위	‘7’	37 H		
D1094 상위	‘0’	30 H		
D1095 하위	‘7’	37 H	LRC CHK 1	LRC CHK (0,1) 는 오류 체크 코드
D1095 상위	‘1’	31 H	LRC CHK 0	

PLC 수신 데이터 레지스터 (응답 메시지)

레지스터	데이터		
D1070 하위	‘0’	30 H	ADR 1
D1070 상위	‘1’	31 H	ADR 0
D1071 하위	‘0’	30 H	CMD 1
D1071 상위	‘6’	36 H	CMD 0
D1072 하위	‘0’	30 H	데이터주소
D1072 상위	‘1’	31 H	
D1073 하위	‘0’	30 H	
D1073 상위	‘0’	30 H	
D1074 하위	‘1’	31 H	데이터내용
D1074 상위	‘7’	37 H	
D1075 하위	‘7’	37 H	
D1075 상위	‘0’	30 H	
D1076 하위	‘7’	37 H	LRC CHK 1
D1076 상위	‘1’	31 H	LRC CHK 0

프로그램 예제 2:

PLC 와 VFD-S 시리즈 AC 드라이브간의 통신(RTU 모드, M1143=On)



PLC ⇨ VFD-S, PLC 전송: **01 06 2000 0012 02 07**

VFD-S ⇨ PLC, PLC 수신: **01 06 2000 0012 02 07**

PLC 전송 데이터 레지스터(전송 메시지)

레지스터	데이터	
D1089 하위	01 H	주소
D1090 하위	06 H	기능
D1091 하위	20 H	데이터주소
D1092 하위	00 H	
D1093 하위	00 H	데이터내용
D1094 하위	12 H	
D1095 하위	02 H	CRC CHK 하위
D1096 하위	07 H	CRC CHK 상위

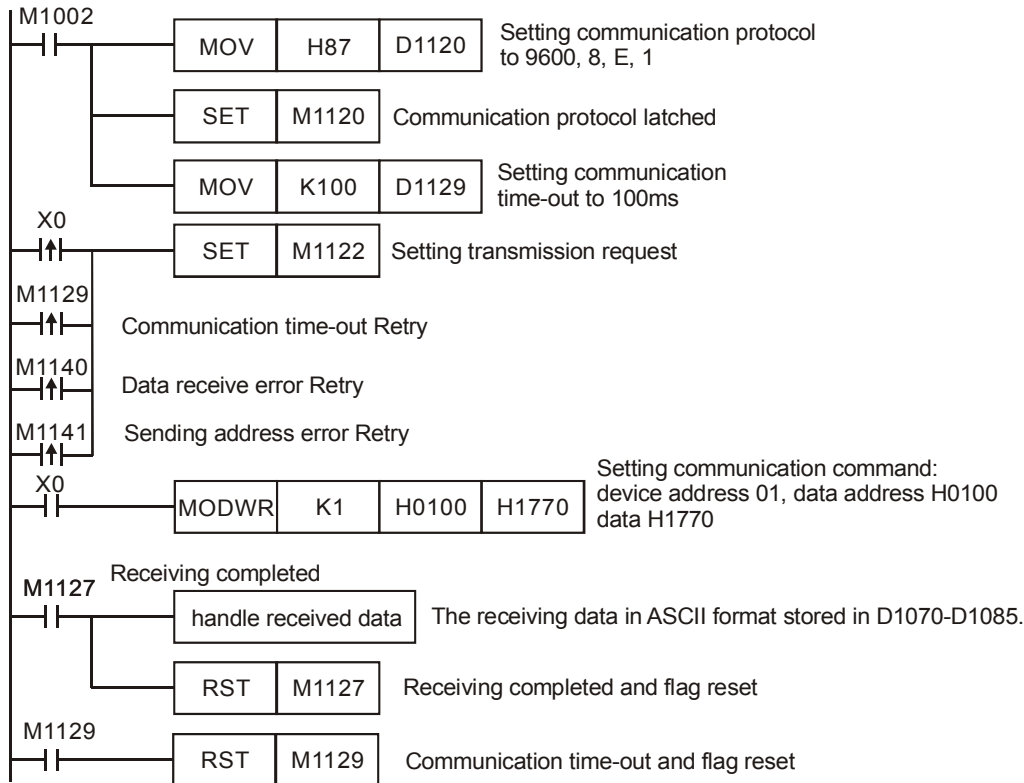
PLC 수신 데이터 레지스터 (응답 메시지)

레지스터	데이터	
D1070 하위	01 H	주소
D1071 하위	06 H	기능
D1072 하위	20 H	데이터주소
D1073 하위	00 H	
D1074 하위	00 H	데이터내용
D1075 하위	12 H	
D1076 하위	02 H	CRC CHK 하위
D1077 하위	07 H	CRC CHK 상위

### 프로그램 예제 3:

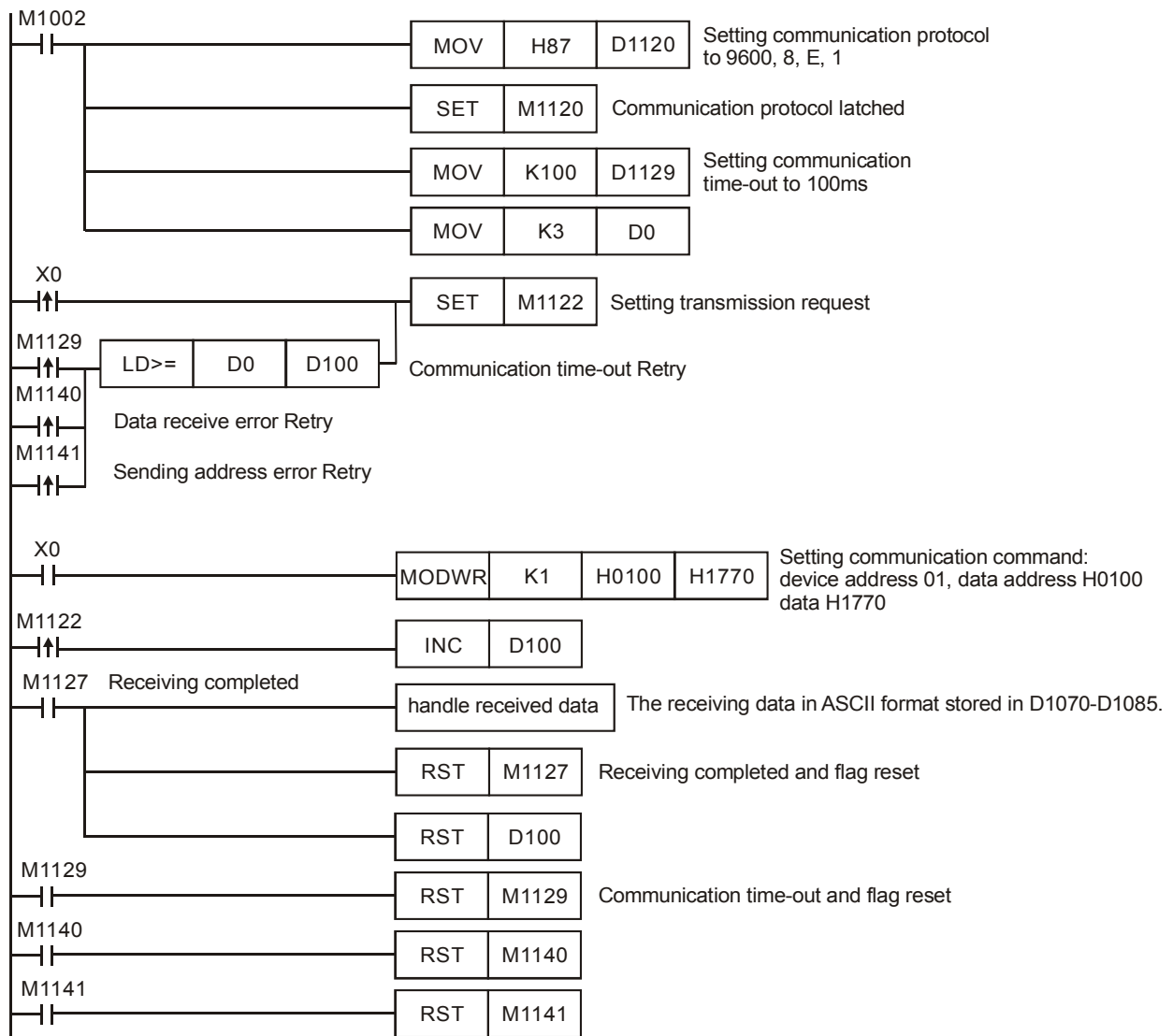
1. PLC 를 VFD-S 시리즈 AC 드라이브에 연결합니다. (ASCII 모드, M1143=Off) 통신시간경과된 후, 데이터 수신 혹은 주소 전송 과정에서 오류가 발생시 다시 시작합니다.
2. X0=On 일 때, PLC 는 데이터 H1770(K6000)를 장치 01 (VFD-S)의 주소 H0100 에 쓸 것 입니다.
3. 통신이 시간경과될때 플래그 M1129 가 On 이 되고 프로그램은 M1129 로부터의 요청을 전달하여 M1122 가 다시 데이터를 읽도록 요청합니다.
4. 오류신호를 받았을 경우 플래그 M1140 가 On 이 되고 프로그램은 M1140 로부터의 요청을 전달하여 M1122 가 다시 데이터를 읽도록 요청합니다.
5. 주소오류를 받았을 경우 플래그 M1141 가 On 이 되고 프로그램은 M1141 로부터의 요청을

전달하여 M1122 가 다시 데이터를 읽도록 요청합니다.



#### 프로그램 예제 4:

1. 일단 PLC 와 VFD-S 가 연결되면(ASCII 모드, M1143=OFF), 통신시간경과 및 데이터 받기 오류 또는 주소 전송 오류일때 Retry time 을 D0=3 로 설정하시기 바랍니다.
2. X0=On 일 때, PLC 는 데이터 H1770(K6000)를 장치 01 (VFD-S)의 주소 H0100 에 쓸 것 입니다
3. 통신이 시간경과될때 플래그 M1129 가 On 이 되고 프로그램은 M1129 로부터의 요청을 전달하여 M1122 가 다시 데이터를 읽도록 요청합니다. 시도횟수 D0=3 입니다.
4. 데이터수신오류가 발생하였을 때 플래그 M1140 가 On 이 되고 프로그램은 M1140 로부터의 요청을 전달하여 M1122 가 다시 데이터를 읽도록 요청합니다. 시도횟수 D0=3 입니다.
5. 주소전달오류가 발생하였을 때 플래그는 On 이 되고 프로그램은 M1141 로부터의 요청을 전달하여 M1122 가 다시 데이터를 쓰도록 요청합니다. 시도횟수 D0=3 입니다.

**주의:**

1. 관련 플래그 및 특별한 레지스터의 상세 정보에 관해서는 API 80 RS 명령어의 각주를 참고하시기 바랍니다.
2. 만일 101 MODWR 및 API 150 MODRW (기능 코드 H06 및 H10) 전에 LDP, ANDP, ORP 와 같은 상승점(rising-edge) / LDF, ANDF, ORF 와 같은 하향점(falling-edge)를 사용하면, 정확한 작동을 위해 전송요청 M1122 을 시작해야 합니다.



API	간략기호	피연산자	기능	제어기						
102	FWD	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(n)</b>	VFD-A 시리즈 드라이브 정주행 명령어	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GPWM: 7 단계	
S <sub>1</sub>					*	*							*				
S <sub>2</sub>					*	*							*				
n					*	*							*				

API	간략기호	피연산자	기능	제어기						
103	REV	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(n)</b>	VFD-A 시리즈 드라이브 역주행 명령어	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GPWM: 7 단계	
S <sub>1</sub>					*	*							*				
S <sub>2</sub>					*	*							*				
n					*	*							*				

API	간략기호	피연산자	기능	제어기						
104	ST 연산자	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(n)</b>	VFD-A 시리즈 드라이브 정지 명령어	E S	E X	S S	S A	S X	S C	E H

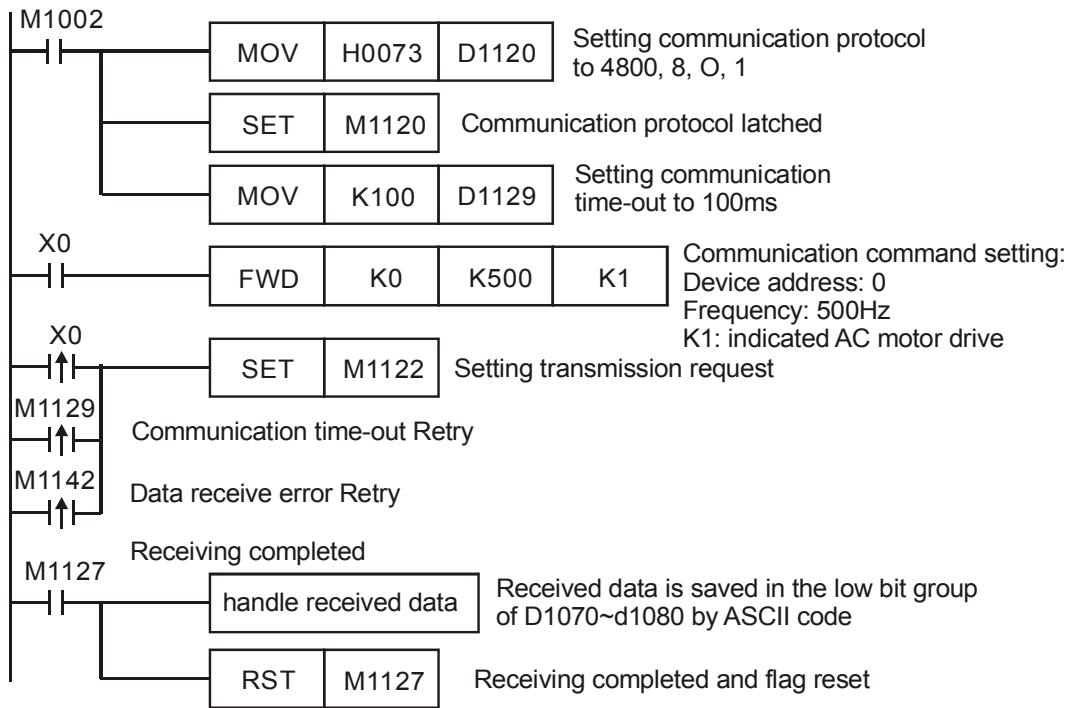
유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GPWM: 7 단계	
S <sub>1</sub>					*	*							*				
S <sub>2</sub>					*	*							*				
n					*	*							*				

## 설명:

- ES 시리즈 모델은 인덱스 레지스터 E 와 F 를 지원하지 않습니다.
- S<sub>1</sub>**: 통신 주소. (K0~K31) **S<sub>2</sub>**: AC 드라이브 마스터주파수. (n=K1 또는 K2) **n**: 명령객체.
- FWD/REV/ST 연산자는 델타 A/H 시리즈 드라이브의 통신명령어 입니다. 이 명령어들을 적용할 때 통신초과시간설정(communication overtime setting):(D1129)를 사용할지 확인 하십시오.
- S<sub>2</sub>**는 AC 드라이브 마스터주파수를 표시합니다. VFD-A 시리즈의 AC 드라이브 마스터주파수는 K0000 에서 K4000 이고 이것은 0.0Hz 에서 400.0Hz 를 나타냅니다. H 시리즈의 AC 드라이브 마스터 주파수는 K0000 에서 K1500 이고 이것은 0Hz 에서 1500Hz 를 나타냅니다.
- 명령객체 **n** 이 n=1 이면 드라이브 한개와 통신합니다. n=2 이면 연결된 모든 드라이브와 통신합니다.
- 주변 장치로부터의 피드백데이터(반환값)는 D1070 부터 D1080 에 저장되어 집니다.  
피드백데이터(반환값)의 수신이 완료된 후 PLC 는 모든 피드백데이터(반환값)들이 정확한지를 체크 합니다. 오류가 있을 경우 M1142 가 ON 이 됩니다. 만일 n = 2 이면 PLC 는 그 데이터들을 수신하지 않습니다.

## 프로그램 예제:

PLC 와 VFD-A 시리즈 AC 드라이브간의 통신, 통신시간경과 및 수신데이터오류를 위한 재시도.



PLC ⇨ VFD-A, PLC 전송: “C ♥ ☺ 0001 0500 ”

VFD-A ⇨ PLC, PLC 수신: “C ♥ ♠ 0001 0500 ”

PLC 전송 데이터 레지스터(전송 메시지)

레지스터	데이터		
D1089 하위	‘C’	43 H	명령시작워드
D1090 하위	‘♥’	03 H	체크섬
D1091 하위	‘☺’	01 H	명령객체
D1092 하위	‘0’	30 H	통신주소
D1093 하위	‘0’	30 H	
D1094 하위	‘0’	30 H	
D1095 하위	‘1’	31 H	
D1096 하위	‘0’	30 H	동작명령
D1097 하위	‘5’	35 H	
D1098 하위	‘0’	30 H	
D1099 하위	‘0’	30 H	

PLC 수신 데이터 레지스터 (응답 메시지)

레지스터	데이터		
D1070 하위	‘C’	43 H	명령시작워드
D1071 하위	‘♥’	03 H	체크섬
D1072 하위	‘♠’	06 H	응답인증 (맞음: 06H, 오류: 07 H)
D1073 하위	‘0’	30 H	통신주소
D1074 하위	‘0’	30 H	
D1075 하위	‘0’	30 H	
D1076 하위	‘1’	31 H	
D1077 하위	‘0’	30 H	동작명령
D1078 하위	‘5’	35 H	
D1079 하위	‘0’	30 H	
D1080 하위	‘0’	30 H	

API	간략기호	피연산자	기능	제어기						
105	RDST	<b>(S)</b> <b>(n)</b>	VFD-A 시리즈 드라이브상태 읽기	E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GPWM: 7 단계	
S					*	*							*				
n					*	*							*				

## 설명:

1. **S**: 통신 주소, K0~K31    **n**: 상태객체 K0~K3
2. ES 시리즈 모델은 인덱스 레지스터 E 와 F 를 지원하지 않습니다.
3. RDST 는 델타 VFD-A 시리즈 AC 드라이브를 위한 특정 통신용 간이명령어이며 AC 드라이브의 실행상태를 읽기위해 사용 합니다.
4. **n**: 상태객체  
     **n = 0** 주파수명령                      **n = 2** 출력전류  
     **n = 1** 출력주파수                    **n = 3** 동작명령
5. 주소 D1070로부터 D1080 의 하위바이트에 저장된 피드백데이터(반환값)들은 모두 11 워드입니다.(VFD-A 시리즈 매뉴얼을 참조하시기 바랍니다.)

"Q, S, B, Uu, Nn, ABCD"

피드백 (반환값)	설명	데이터 저장소
Q	시작워드: 'Q' (51H).	D1070 하위
S	체크섬 코드: 03H.	D0171 하위
B	명령인증. 정확: 06H, 오류: 07H.	D1072 하위
U	통신주소 (주소: 00~31). "Uu" = ("00"~"31") ASCII 로 표시.	D1073 하위
u		D1074 하위
N	상태객체(00~03)."Nn" = ("00~03") ASCII 로 표시.	D1075 하위
n		D1076 하위
A	상태 데이터. "ABCD"의 내용은 상태객체(00~03)에 따라서 다를 수 있습니다. 00~03 은 각각 주파수, 전류, 동작모드를 표시합니다. 상세한 내용은 아래의 설명을 참조하시기 바랍니다.	D1077 하위
B		D1078 하위
C		D1079 하위
D		D1080 하위
	Nn = "00"      주파수 명령 = ABC.D (Hz) Nn = "01"      출력 명령 = ABC.D (Hz) Nn = "02"      출력 전류 = ABC.D (A) PLC 는 자동으로 "ABCD"의 ASCII 단어를 숫자값으로 변환시켜 D1050 에 저장합니다. 예를 들면, "ABCD"가 "0600"이라면, PLC 는 자동으로 ASCII 단어를 K0600(0258 H)이라는 값으로 변환시켜 D1050 에 있는 특별 레지스터에 저장 합니다. Nn = "03"      동작명령	

	'A' =	'0'	정지,	'5'	조그(정)
		'1'	정방향 동작,	'6'	조그(역)
		'2'	정지,	'7'	조그(역)
		'3'	역방향 동작	'8'	비정상
		'4'	조그(정),		

		ES 시리즈 PLC 는 자동으로 "A"의 ASCII 단어를 숫자값으로 변환시켜 D1051 에 저장합니다. 예를 들면, "A"가 "3"이라면, PLC 는 자동으로 ASCII 단어를 K0003(03H)이라는 값으로 변환시켜 D1051 에 있는 특별 레지스터에 저장 합니다.					
	'B' =	b7	b6	b5	b4	동작명령소스	
0		0	0	0	디지털 키패드		
0		0	0	1	1 차 속도		
0		0	1	0	2 차 속도		
0		0	1	1	3 차 속도		
0		1	0	0	4 차 속도		
0		1	0	1	5 차 속도		
0		1	1	0	6 차 속도		
0		1	1	1	7 차 속도		
1		0	0	0	조그 주파수		
1		0	0	1	아날로그 신호 주파수 명령		
1		0	1	0	RS-485 통신 인터페이스		
1		0	1	1	업/다운 제어		
b3		=	0	DC 제동정지 없음	1 DC 제동정지		
b2		=	0	제동기동 없음	1 DC 제동기동		
b1		=	0	정방향	1 역방향		
b0		=	0	정지	1 동작		
ES 시리즈를 위해 PLC 는 특별한 외부 릴레이 M1168 (b0)에서 M1175(b7)에 "B"의 값을 저장합니다. SA/EH 시리즈를 위해 PLC 는 특별한 레지스터 D1051 의 상위 바이트에 "B"의 16 진수값을 저장합니다.							
		"CD" =	"00"	비정상기록 없음		"10"	OcA
"01"			oc		"11"	Ocd	
"02"			ov		"12"	Ocn	
"03"			oH		"13"	GFF	
"04"			oL		"14"	Lv	
"05"			oL1		"15"	Lv1	
"06"			EF		"16"	cF2	
"07"	cF1		"17"	bb			
"08"	cF3		"18"	oL2			
"09"	HPF		"19"				
ES/SA/EH 시리즈를 위해, PLC 는 자동으로 "CD"의 ASCII 단어를 숫자값으로 변환시켜 D1052 에 저장합니다. 예를 들면, "CD"가 "06"이라면, PLC 는 자동으로 ASCII 단어를 0006H 이라는 값으로 변환시켜 D1052 에 있는 특별 레지스터에 저장 합니다.							

주의:

LDP, ANDP, ORP 등의 상승접점(Rising-edge contact)과 LDF, ANDF, ORF 등의 하향접점(falling-edge contact)들은 API 100 MODRD, API 105 RDST, API 150 MODRW (기능코드 H03)등의 세가지 명령어를 사용하기전에 사용되어 질 수 없습니다. 그렇지 않으면, 전달받은 레지스터에 저장되어 있는 데이터들은 부정확하게 됩니다.

API	간략기호	피연산자	기능	제어기						
106	RSTEF	<b>S</b> <b>n</b>	VFD-A 시리즈 드라이브 비정상상태 복귀	E S	E X	S S	S A	S X	S C	E H

연산자 유형	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GPWM: 7 단계	
S					*	*							*				
n					*	*							*				

**Explanations:**

1. **S**: 통신 주소. (K0~K31)    **n**: 명령객체. (n=K1 또는 K2)
2. ES 시리즈 모델은 인덱스 레지스터 E 와 F 를 지원하지 않습니다.
3. RSTEF 는 델타 VFD-A 시리즈 AC 드라이브를 위한 특정 통신용 간이명령어이며 비정상실행 후 AC 드라이브를 복귀하기 위해 사용 합니다.
4. **n**: 명령객체, n=1 이면 드라이브 한개와 통신합니다. n=2 이면 연결된 모든 드라이브와 통신합니다.
5. 주변 장치로부터의 피드백데이터(반환값)는 D1070 부터 D1080 에 저장되어 집니다. 만일 n = 2 이면 피드백데이터(반환값)은 없습니다.
6. LDP, ANDP, ORP 등의 상승접점(Rising-edge contact)과 LDF, ANDF, ORF 등의 하향접점(falling-edge contact)들은 API 100 MODRD, API 105 RDST, API 150 MODRW (기능코드 H03)등의 세가지 명령어를 사용하기전에 사용되어 질 수 없습니다. 그렇지 않으면, 전달받은 레지스터에 저장되어 있는 데이터들은 부정확하게 됩니다.

**주의:**

관련 플래그 및 특별 레지스터들의 상세정보는 API 80 RS 명령어 각주를 참고하시기 바랍니다.

API	간략기호			피연산자			기능			제어기							
107		LRC	P	<b>S</b>	<b>n</b>	<b>D</b>	LRC 발생기				E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S													*			LRC, LRCP: 7 단계	
n					*	*							*				
D													*				

펄스								16-비트								32-비트							
E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H	

피연산자:

**S:** 체크섬동작을 위한 시작 장치 (ASCII 모드)    **n:** 피연산자수    **D:** 동작결과를 저장하기 위한 시작 장치.

LRC 체크: 각주를 참고하시기 바랍니다.

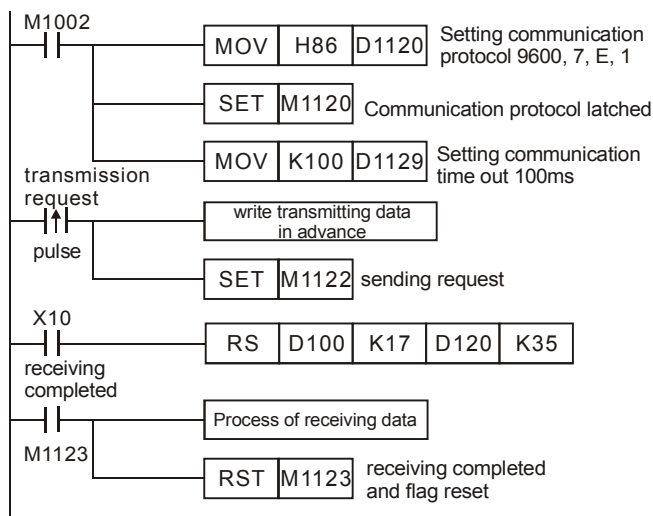
**설명:**

1. **n**: 피연산자수는 짝수이고 K1~K256 의 범위에 있어야 합니다. 만일 범위를 벗어나면, 오류가 발생되고 명령은 실행되지 않습니다. 이때, M1067 과 M1068 은 On 이 되며 오류코드 0E1A 는 D1067 에 기록됩니다.
2. **16-비트 변환모드**: M1161=Off 일 때, 원장치(Source Device) **S**로부터 시작된 16 진수 데이터는 상위 8 비트와 하위 8 비트로 나누어지며 **n** 개의 LRC 명령동작을 수행합니다. 그 다음에, 장치 **D**의 상위 및 하위 8 비트에 결과를 저장합니다.
3. **8-비트 변환모드**: M1161= On 일 때, 원장치(Source Device) **S**로부터 시작된 16 진수 데이터는 상위 8 비트와 하위 8 비트로 나누어지며 **n** 개의 LRC 명령동작을 수행합니다. 그 다음에, 장치 **D**의 하위 8 비트에 결과를 저장하고 2 개의 레지스터를 사용 합니다. (**D**의 상위 8 비트는 모두 0 입니다. )

### 프로그램 예제:

PLC 와 VFD-B 시리즈 AC 드라이브간의 통신(ASCII 모드, M1143= Off), (8-비트 모드, M1161=On),

VFD-B 파라미터 주소 H2101 로 부터 여섯개의 데이터를 읽기 전에 전송중인 데이터 쓰기.



PLC ⇒ VFD-B, PLC 전송: “: 01 03 2101 0006 D4 CR LF ”

## PLC 전송 데이터 레지스터 (전송 메시지)

레지스터	데이터			
D100 하위	‘.’	3A H	STX	
D101 하위	‘0’	30 H	ADR 1	ADR (1,0) 는 AC 드라이브 주소
D102 하위	‘1’	31 H	ADR 0	
D103 하위	‘0’	30 H	CMD 1	CMD (1,0) 는 명령어 코드
D104 하위	‘3’	33 H	CMD 0	
D105 하위	‘2’	32 H	데이터시작주소	
D106 하위	‘1’	31 H		
D107 상위	‘0’	30 H		
D108 하위	‘1’	31 H		
D109 하위	‘0’	30 H	데이터수(word 기준)	
D110 하위	‘0’	30 H		
D111 하위	‘0’	30 H		
D112 하위	‘6’	36 H		

레지스터	데이터			
D113 하위	'D'	44 H	LRC CHK 1	LRC CHK (0,1) 는 오류 체크 코드
D114 하위	'4'	34 H	LRC CHK 0	
D115 하위	CR	A H	END	
D116 하위	LF	D H		

위에 기술된 LRC CHK (0,1)는 오류체크코드이며, LRC 명령에 의해 연산되어 질 수 있습니다. (8-비트 모드, M1161=On)



LRC 체크:  $01\text{ H} + 03\text{ H} + 21\text{ H} + 01\text{ H} + 00\text{ H} + 06\text{ H} = 2\text{C H}$ , 다음 2의 보수를 취해서 D4H 임. 이때, ‘D’(44 H)는 D113의 하위 8 비트에 저장되고, ‘4’ (34 H)는 D114의 하위 8 비트에 저장 됩니다.

#### 주의:

- 통신데이터의 ASCII 모드, 포맷은 아래에 있습니다:

STX	‘.’	시작 단어= ‘.’ (3AH)
주소 Hi	‘0’	통신: 2 개의 ASCII 코드로 이루어진 8-비트 주소
주소 Lo	‘1’	
기능 Hi	‘0’	기능 코드: 2 개의 ASCII 코드로 이루어진 8-비트 함수
기능 Lo	‘3’	
데이터 (n-1)	‘2’	데이터 내용: n × 2 개의 ASCII 코드로 이루어진 8-비트 데이터 내용
.....	‘1’	
데이터 0	‘0’	
	‘2’	
	‘0’	
	‘0’	
	‘0’	
	‘2’	
LRC CHK Hi	‘D’	LRC 체크: 2 개의 ASCII 코드로 이루어진 8-비트 체크섬
LRC CHK Lo	‘7’	
END Hi	CR	종료어: END Hi = CR (0DH), END Lo = LF (0AH)
END Lo	LF	

- 통신프로토콜은 MODBUS ASCII (American Standard Code for Information Interchange)로 만들어져 있습니다. 각가의 바이트는 2 개의 ASCII 문자로 구성 됩니다.
- LRC 체크는 “주소”로 부터 “데이터 내용”까지를 더한 값입니다. 예를 들면,  $01\text{ H} + 03\text{ H} + 21\text{ H} + 02\text{ H} + 00\text{ H}$

+ 02H = 29H 한 후, 2 의 보수인 D7H 를 취합니다.

API	간략기호			피연산자	기능	제어기						
108		CRC	P	<b>S</b> <b>n</b> <b>D</b>	CRC 발생기	E S	E X	S S	S A	S X	S C	E H

연산자 \ 유형	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S													*			CRC, CRCP: 7 단계	
n					*	*							*				
D													*				

펄스							16-비트							32-비트						
E	E	S	S	S	S	E	E	E	S	S	S	S	E	E	E	S	S	S	S	E
S	X	S	A	X	C	H	S	X	S	A	X	C	H	S	X	S	A	X	C	H

피연산자:

**S:** 체크섬동작을 위한 시작 장치 (RTU 모드)    **n:** (n=K1~K256)에서 동작을 수행하기 위한 값의 수    **D:** 동작결과를 저장하기 위한 목적지

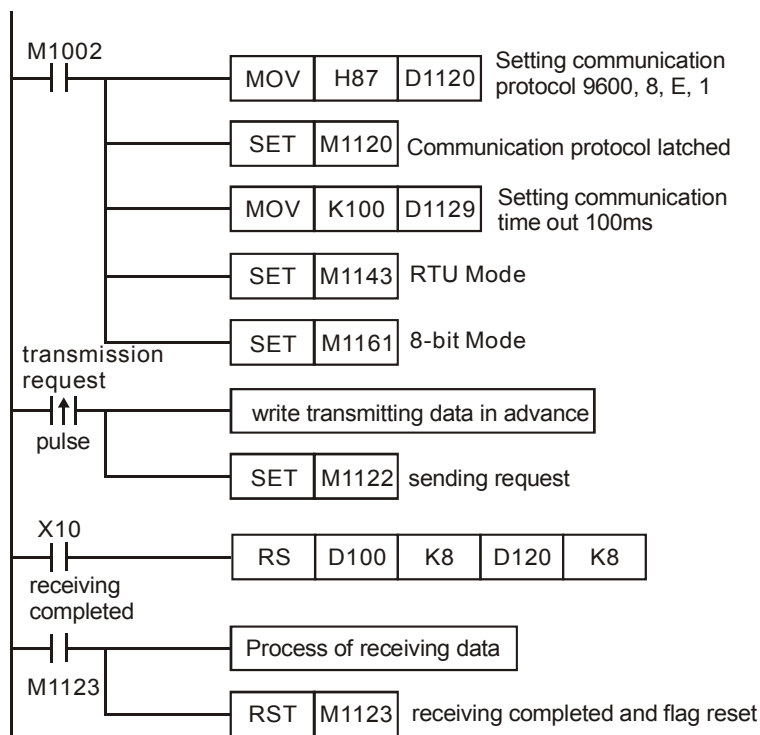
설명:

- S:** 체크섬동작을 위한 시작 장치 (RTU 모드)    **n:** 피연산자수    **D:** 동작결과를 저장하기 위한 시작 장치.  
CRC 체크: 각주를 참고하시기 바랍니다.
- n:** K1~K256 의 범위에 있어야 합니다. 만일 범위를 벗어나면, 오류가 발생되고 명령은 실행되지 않습니다.  
이때, M1067 과 M1068 은 On 이 되며 오류코드 0E1A 는 D1067 에 기록 됩니다.
- 16-비트 변환모드: M1161=Off 일 때, 원장치(Source Device) **S**로부터 시작된 16 진수 데이터는 상위비트와 하위비트로 나누어 집니다. **n** 개의 CRC 명령을 수행하고 그 결과를 장치 **D** 의 상위 및 하위 8 비트에 저장합니다.
- 8-비트 변환모드: M1161= On 일 때, 원장치(Source Device) **S**로부터 시작된 16 진수 데이터는 상위바이트(무효데이터)와 하위바이트로 나누어집니다. **n** 개의 CRC 명령동작을 수행하고, 장치 **D** 의 하위바이트에 결과를 저장하고 2 개의 레지스터를 사용 합니다. (**D** 의 상위 8 비트는 모두 0 입니다.)

프로그램 예제:

PLC 가 VFD-S AC 드라이브에 연결할때 (RTU 모드, M1143=ON), (16-비트 모드, M1161=ON),  
VFD-S 파라미터주소 H2000 에 앞당겨 H12 인 전송데이터를 쓰기



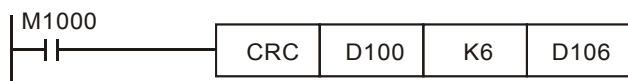


PLC ⇨ VFD-S, PLC 전송: **01 06 2000 0012 02 07**

PLC 전송 데이터 레지스터 (PLC 전송 메시지)

레지스터	데이터	
D100 하위	01 H	주소
D101 하위	06 H	기능
D102 하위	20 H	데이터 주소
D103 하위	00 H	
D104 하위	00 H	데이터 내용
D105 하위	12 H	
D106 하위	02 H	CRC CHK 0
D107 하위	07 H	CRC CHK 1

위에 기술된 CRC CHK (0,1)는 오류체크코드이며, CRC 명령에 의해 연산되어 질 수 있습니다. (8-비트 모드, M1161=On).



CRC 체크: 이때, 02 H 는 D106 의 하위 8 비트에 저장되고, 07 H 는 D107 의 하위 8 비트에 저장 됩니다.

주의:

1. 통신데이터의 RTU 모드, 포맷은 아래에 있습니다:

START	아래의 설명을 참조하시기 바랍니다.
주소	통신 주소: 8-비트 이진값
기능	기능 코드: 8-비트 이진값
DATA (n-1)	데이터 내용: n × 8-비트 데이터
.....	
DATA 0	
CRC CHK 하위	CRC 체크: 2 개의 8 비트 이진값으로 된 16-비트 CRC 체크섬
CRC CHK 상위	

END	아래의 설명을 참조하시기 바랍니다.
-----	---------------------

## 2. CRC 체크:

CRC 체크는 “주소”에서 시작하여 “데이터 내용”에서 끝납니다. 계산법은 아래와 같습니다:

단계 1: 16 비트 레지스터(CRC 레지스터)에 FFFFH 값을 올립니다.

단계 2: 첫째 8 비트 바이트에 하위바이트의 16 비트 CRC 레지스터를 배타적 논리합을 시킵니다. 그 다음, 결과값을 CRC 레지스터에 저장합니다.

단계 3: 그 CRC 레지스터를 오른쪽으로 1 비트 시프트 시키고 상위 비트들을 0 으로 채웁니다.

단계 4: 오른쪽으로 시프트한 값을 체크 합니다. 만일 그 값이 0 이라면, 단계 3 에서 새로운 값을 받아 CRC 레지스터에 저장합니다. 다른 방법으로는 A001H 와 CRC 레지스터값을 배타적 논리합을 시킨 후 그 결과를 CRC 레지스터에 저장합니다.

단계 5: 단계 3 과 4 를 반복하고 그 8 비트를 계산합니다.

단계 6: 모든 명령어가 처리 될때까지 다음 8 비트 메시지 명령을 위하여 단계 2 에서 5 를 반복합니다. 그리고 최종적으로 획득한 CRC 레지스터의 값이 CRC 체크값 입니다. 주목해야 할 것은 CRC 체크는 메시지명령의 체크성에 호환성 있게 놓여져야 합니다.

API	간략기호			피연산자	기능										제어기						
109		SWRD	P		디지털 스위치 읽기										E S	E X	S S	S A	S X	S C	E H

연산자	유형	비트 장치				워드 장치										프로그램 단계				
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SWRD, SWRDP: 3 단계			
D								*	*	*	*	*	*							

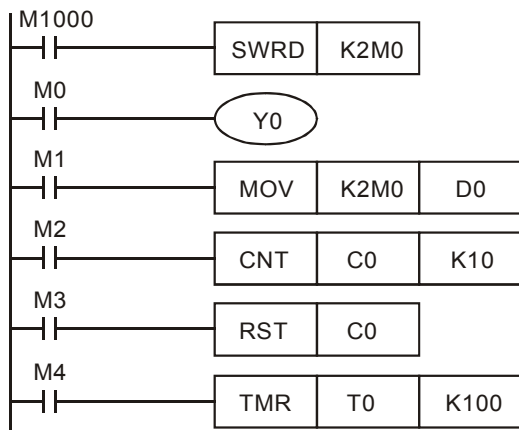
펄스								16-비트					32-비트							
E	E	S	S	S	S	E	E	E	S	S	S	S	E	E	E	S	S	S	S	E
S	X	S	A	X	C	H	S	X	S	A	X	C	H	S	X	S	A	X	C	H

**설명:**

1. **D**: 읽은값을 저장하기 위한 장치
2. **D**의 하위바이트에 디지털스위치 기능카드로 부터 읽은 값을 저장합니다.
3. 모든 디지털스위치는 결합 **BIT** 를 가지고 있습니다.
4. 만일 디지털스위치기능카드 없이 이 명령이 실행 될 경우, 어떠한 결과 및 오류메세지 **C400(Hex)**도 발생하지 않습니다.

**프로그램 예제:**

1. 디지털 스위치 기능카드에는 8 개의 **DIP** 스위치가 있습니다. 값을 읽기 위해서 **SWRD** 명령을 사용한 후, 이 8 개의 **DIP** 스위치는 접점 **M0** 에서 **M7** 과 함께 연동 됩니다.



2. **M0** 에서 **M7** 까지는 각각의 접점명령의 사용에 의해서 실행됩니다.
3. **END** 명령이 실행되었을때, 입력처리가 종료 됩니다. **REF(I/O 리프레시)**명령은 무효가 됩니다.
4. **SWRD** 명령이 디지털 스위치 기능카드의 입력 데이터를 사용할 때, 1 회읽기비트(read one time bits)는 4 비트 입니다.

**주의:**

디지털 스위치 기능카드가 삽입될때, 8 개의 **DIP** 스위치들은 각각 **M1104~M1111** 에 대응 됩니다.



API	간략기호			피연산자				기능				제어기						
111	D	EZCP	P	<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	<b>(S)</b>	<b>(D)</b>	부동 영역 비교				<b>E</b> <b>S</b>	<b>E</b> <b>X</b>	<b>S</b> <b>S</b>	<b>S</b> <b>A</b>	<b>S</b> <b>X</b>	<b>S</b> <b>C</b>	<b>E</b> <b>H</b>

유형 연산자	비트 장치				워드 장치												프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
S <sub>1</sub>					*	*								*			DEZCP, DEZCPP: 17 단계	
S <sub>2</sub>					*	*								*				
S					*	*								*				
D		*	*	*														

펄스								16-비트								32-비트							
E	E	S	S	S	S	E		E	E	S	S	S	S	E		E	E	S	S	S	S	E	
S	X	S	A	X	C	H		S	X	S	A	X	C	H		S	X	S	A	X	C	H	

피연산자:

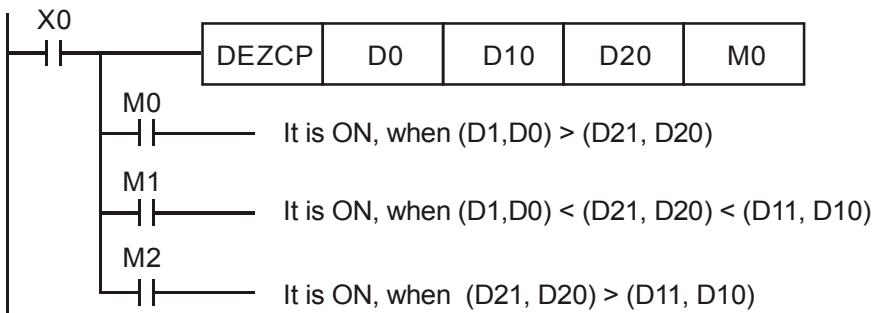
**S<sub>1</sub>**: 영역비교의 최저한계   **S<sub>2</sub>**: 영역비교의 최고한계   **S**: 비교값   **D**: 비교결과(점유된 세개의 연속된 장치)

설명:

- S** 데이터는 **S<sub>1</sub>~S<sub>2</sub>** 범위의 데이터와 비교되어 그 결과(>, =, <)가 **D**에 있는 세개의 비트장치에 의해 표시됩니다.
- 만일 원시 피연산자인 **S<sub>1</sub>** 혹은 **S<sub>2</sub>**가 상수인 **K** 또는 **H**로 표시 되어진다면, 그 정수값은 자동적으로 비교하기 위해서 이진 부동 소수점으로 변환 됩니다.
- S<sub>1</sub> > S<sub>2</sub>** 일때 **S<sub>1</sub>**는 비교를 위한 최고 또는 최저한계로서 사용 되어 집니다.

프로그램 예제:

- 특정 장치를 **M10** 이라 하면, **M10~M12**가 자동적으로 사용되어 집니다.
- X0=On** 이라고 하고 **DEZCP** 명령을 실행하면 **M10~M12** 중 하나는 **On** 이 됩니다. **X0=Off** 이고 **DEZCP** 명령을 실행하지 않으면 **M10~M12**는 **X0=Off** 이전의 상태를 유지 합니다
- 결과를 재설정 하기위해 **RST** 혹은 **ZRST** 명령을 사용하십시오.



주의:

부동 소수점의 연산함수에 관해서는 **CH 5.3** 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
116	D	RAD	P	<b>(S)</b> <b>(D)</b>	각도 → 라디안	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치												프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DRAD, DRADP: 9 단계	
S					*	*							*				
D													*				

펄스							16-비트							32-비트						
E	E	S	S	S	S	E	E	E	S	S	S	S	E	E	E	S	S	S	S	E
S	X	S	A	X	C	H	S	X	S	A	X	C	H	S	X	S	A	X	C	H

피연산자:

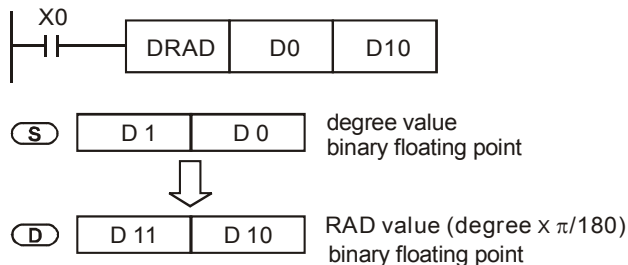
**S**: 근원지(각도)    **D**: 목적지 결과(라디안)

설명:

1. 각도를 라디안값으로 변환하기 위해서 아래의 함수를 사용하세요:  
라디안 = 각도 × ( $\pi/180$ )
2. 만일 변화결과의 절대값이 최고 부동소수점보다 클 경우 플래그를 자리올림하고 M1022=On 입니다.
3. 만일 변화결과의 절대값이 최고 부동소수점보다 적을 경우 플래그 자리내림하고 M1022=On 입니다.
4. 만일 변환결과가 0 이면, 제로 플래그이고 M1020=On 입니다.

프로그램 예제:

X0=On 일때, 특정 이진 부동 소수점(D1, D0)의 각도값을 (D11, D10)에 저장하기 위해 라디안으로 변환하고 그 결과는 이진 부동 소수점 입니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
117	D	DEG	P	<b>(S)</b> <b>(D)</b>	라디안 → 각도	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DDEG, DDEGP: 9 단계	
S					*	*							*				
D													*				

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

피연산자:

**S:** 데이터근원 (라디안)    **D:** 변환결과 (각도)

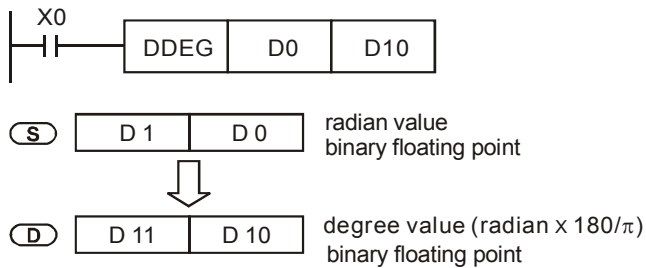
설명:

1. 라디안값을 각도로 변환하기 위해서 아래의 함수를 사용하세요:  

$$\text{각도} = \text{라디안} \times (180/\pi)$$
2. 만일 변환결과와 절대값이 최고 부동소수점보다 클 경우 플래그를 자리올림하고 M1022=On 입니다.
3. 만일 변환결과와 절대값이 최고 부동소수점보다 적을 경우 플래그 자리내림하고 M1022=On 입니다.
4. 만일 변환결과가 0 이면, 제로 플래그이고 M1020=On 입니다.

프로그램 예제:

X0=On 일때, 특정 이진 부동 소수점(D1, D0)의 각도값을 (D11, D10)에 저장하기 위해 라디안으로 변환하고 그 결과는 이진 부동 소수점 입니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능										제어기						
118	D	EBCD	P	<b>S</b> <b>D</b>	부동소수를 과학단위계로 변환										E S	E X	S S	S A	S X	S C	E H

연산자 /  유형	비트 장치				워드 장치										프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEBCD, DEBCDP: 9 단계		
													*					
													*					

펄스								16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H		E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

피연산자:

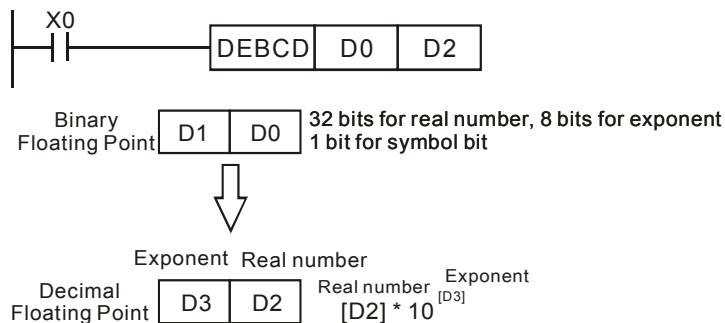
**S**: 데이터근원    **D**: 수신지 결과

설명:

1. **S**에 의해서 정의된 레지스터에 있는 이진 부동소수점값을 **D**에 의해 정의된 레지스터에 저장된 십진 부동소수점으로 변환하시기 바랍니다.
2. PLC 부동소수점은 이진부동소수점에 의해 수행 됩니다. **DEBCD** 명령은 이진부동소수점을 십진부동소수점으로 변환시키는 특정 명령어 입니다.
3. 만일 변화결과의 절대값이 최고 부동소수점보다 클 경우 플래그를 자리올림하고 **M1022=On** 입니다.
4. 만일 변화결과의 절대값이 최고 부동소수점보다 적을 경우 플래그 자리내림하고 **M1022=On** 입니다.
5. 만일 변환결과가 0 이면, 제로 플래그이고 **M1020=On** 입니다.

프로그램 예제:

X0=On 일때, D1, D0 에 있는 이진부동소수점값은 D3, D2 에 저장되어 있는 십진부동소수점으로 변환되어 집니다.



주의:

부동 소수점의 연산함수에 관해서는 **CH 5.3** 상세 수치 처리를 참조하시기 바랍니다.



API	간략기호			피연산자	기능	제어기						
119	D	EBIN	P	<b>(S)</b> <b>(D)</b>	과학단위를 부동소수로 변환	E S	E X	S S	S A	S X	S C	E H

연산자 \ 유형	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S													*			DEBIN, DEBINP: 9 단계	
D													*				

펄스								16-비트								32-비트							
E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H	

피연산자:

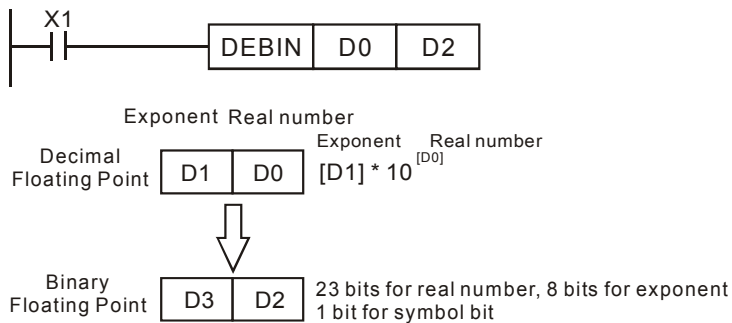
**S:** 데이터근원 **D:** 수신지 결과

설명:

- S**에 의해서 정의된 레지스터에 있는 십진 부동소수점값을 **D**에 의해 정의된 레지스터에 저장된 이진 부동소수점으로 변환하시기 바랍니다.
- 예를 들면, **S=1234**, **S+1=8** 는 **D=1.2345 x 10<sup>5</sup>** 가 됩니다.
- D**는 반드시 이진 부동 소수점 형태가 되어야 합니다. **S** 및 **S+1** 는 각각 실수와 부동소수의 지수를 나타냅니다.
- DEBIN 명령은 십진부동소수점을 이진부동소수점으로 변환시키는 특정 명령어 입니다.
- 십진부동소수점의 실수의 범위는 **-9.999~+9,999** 이고, 지수의 범위는 **-41~+35** 입니다. 십진부동소수점의 실제 범위는 **±1175×10<sup>-41</sup> to ±3402×10<sup>+35</sup>** 입니다. 만일 계산 결과가 0 이면, **M1020=On** 플래그 없음 하시기 바랍니다. 왜냐하면, 입력값이 기 언급되었던 범위의 한계를 넘어서면, 결과값을 산정시 오류가 발생 될 수 있습니다.

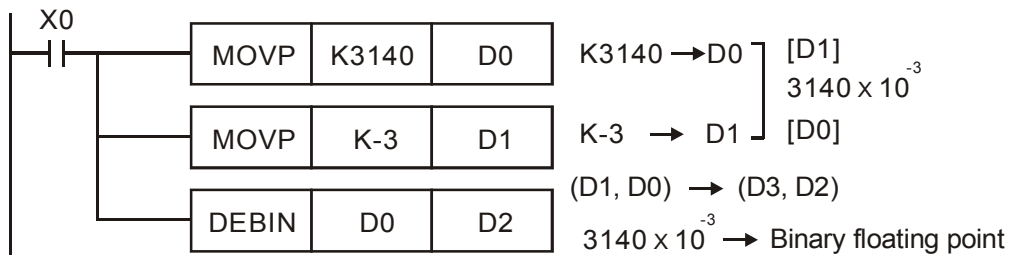
프로그램 예제 1:

X1=On 일때, D1, D0 에 있는 십진부동소수점값은 D3, D2 에 저장되어 있는 이진부동소수점으로 변환되어 집니다.



프로그램 예제 2:

- 부동소수연산을 수행하기 전에, **BIN** 정수를 이진부동소수로 변환하기 위해 **FLT (API 49)** 명령을 반드시 사용하시기 바랍니다. 원데이터(변화되어질 값)은 반드시 **BIN** 정수 이어야 합니다. 그렇지 않으면, **DEBIN** 명령을 사용하여 부동소수값을 이진부동소수값으로 바꿀 수 있습니다.  
X0=On 일때, 십진 부동 소수 형태를 발생시키기 위해 **K3140** 를 **D0** 로 이동 시키고 **K-3** 을 **D1** 으로 이동시키십시오. (**3.14 = 3140 x 10<sup>-3</sup>**)



**주의:**

부동 소수점의 연산함수에 관해서는 **CH 5.3** 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
120	D	EADD	P	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b>	부동 소수 덧셈	E S	E X	S S	S A	S X	S C	E H

연산자 \ 유형	비트 장치				워드 장치												프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEADD, DEADDP: 13 단계		
S <sub>1</sub>					*	*							*					
S <sub>2</sub>					*	*							*					
D													*					

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

피연산자:

**S<sub>1</sub>**: 피가산수 **S<sub>2</sub>**: 가수 **D**: 덧셈 결과

설명:

- S<sub>1</sub> + S<sub>2</sub> = D.** **S<sub>1</sub>** 및 **S<sub>2</sub>**에 의해 정의된 레지스터내의 부동 소수값은 더해져서 그 결과는 **D**에 의해 정의된 레지스터에 저장되어 집니다. 모든 원천 데이터들은 부동 소수 형태로 연산되어 지고 그 결과는 또한 부동 소수 형태로 저장 되어 집니다.
- 만일 피연산자 **S<sub>1</sub>** 또는 **S<sub>2</sub>**가 상수 **K** 또는 **H**로 표시 되어지면, 그 정수는 자동적으로 덧셈 연산을 수행하기 위해 이진 부동 소수로 변환 되어 집니다.
- S<sub>1</sub>** 및 **S<sub>2</sub>**는 동일한 레지스터 번호를 정의할 수 있습니다.(동일한 장치가 **S<sub>1</sub>** 및 **S<sub>2</sub>**으로 사용되어 질 수 있습니다. ) 만일 이러한 경우에, **DEADD** 명령의 지속적인 실행 도중, 레지스터의 데이터는 조건점점(condition contact)이 On 일때, 그 실행주기(Cycle) 동안 모든 스캔 프로그램안에서 한번만 더해 질 수 있습니다. 그 결과, 펄스실행명령(**DEADDP**)이 일반적으로 사용되어 집니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 클 경우 플래그를 자리올림하고 **M1022=On** 입니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 적을 경우 플래그 자리내림하고 **M1022=On** 입니다.
- 만일 변환결과가 0 이면, 제로 플래그이고 **M1020=On** 입니다.

프로그램 예제 1:

**X0=On** 일때, (**D1, D0**)의 이진부동소수값과 (**D3, D2**)의 이진부동소수값을 더하고 그 결과를 (**D11, D10**)에 저장합니다.



프로그램 예제 2:

**X2=On** 일때, (**D11, D10**)의 이진부동소수값과 **K1234** (자동적으로 이진부동소수로 변환된)를 더하고 그 결과를 (**D21, D20**)에 저장합니다.



주의:

부동 소수점의 연산함수에 관해서는 **CH 5.3** 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자		기능			제어기							
121	D	ESUB	P	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>D</b>	부동 소수 뺄셈			E S	E X	S S	S A	S X	S C	E H

연산자 /  유형	비트 장치				워드 장치												프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DESUB, DESUBP: 13 단계		
					*	*							*					
					*	*							*					
													*					

펄스								16-비트								32-비트							
E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H	

피연산자:

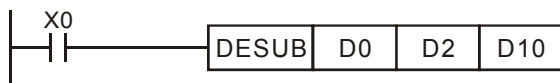
**S<sub>1</sub>**: 피감산수 **S<sub>2</sub>**: 감수 **D**: 뺄셈 결과

설명:

- S<sub>1</sub> - S<sub>2</sub> = D.** **S<sub>2</sub>**에 의해 정의된 레지스터의 부동소수값은 **S<sub>1</sub>**에 의해 정의된 부동소수값에 의해 감해지고 그 결과는 **D**에 의해 정의된 레지스터에 저장 됩니다. 모든 원천 데이터들은 부동 소수 형태로 연산되어 지고 그 결과는 또한 부동 소수 형태로 저장 되어 집니다.
- 만일 피연산자 **S<sub>1</sub>** 또는 **S<sub>2</sub>**가 상수 **K** 또는 **H**로 표시 되어지면, 그 정수는 자동적으로 뺄셈 연산을 수행하기 위해 이진 부동 소수로 변환 되어 집니다.
- S<sub>1</sub>** 및 **S<sub>2</sub>**는 동일한 레지스터 번호를 정의할 수 있습니다.(동일한 장치가 **S<sub>1</sub>** 및 **S<sub>2</sub>**으로 사용되어 질 수 있습니다.) 만일 이러한 경우에, **DESUB** 명령의 지속적인 실행 도중, 레지스터의 데이터는 조건점점(condition contact)이 On 일때, 그 실행주기(Cycle) 동안 모든 스캔 프로그램안에서 한번만 더해 질 수 있습니다. 그 결과, 펄스실행명령(**DESUBP**)이 일반적으로 사용되어 집니다.
- 만일 변화결과의 절대값이 최고 부동소수점보다 클 경우 플래그를 자리올림하고 **M1022=On** 입니다.
- 만일 변화결과의 절대값이 최고 부동소수점보다 적을 경우 플래그 자리내림하고 **M1022=On** 입니다.
- 만일 변환결과가 0 이면, 제로 플래그이고 **M1020=On** 입니다.

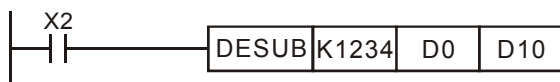
프로그램 예제 1:

**X0=On** 일때, (**D3, D2**)의 이진부동소수값이 (**D1, D0**)의 이진 부동 소수값으로 부터 빼지고 그 결과를 (**D11, D10**)에 저장합니다.



프로그램 예제 2:

**X2=On** 일때, (**D1, D0**)의 이진 부동 소수값이 **K1234** (자동적으로 이진부동소수로 변환된)으로 부터 빼지고 그 결과를 (**D11, D10**)에 저장합니다.



주의:

부동 소수점의 연산함수에 관해서는 **CH 5.3** 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
122	D	EMUL	P	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b>	부동 소수 곱셈	E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEMUL, DEMULP: 13 단계	
S <sub>1</sub>					*	*							*				
S <sub>2</sub>					*	*							*				
D													*				

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

피연산자:

**S<sub>1</sub>**: 피승수    **S<sub>2</sub>**: 승수    **D**: 곱셈 결과

설명:

- S<sub>1</sub> × S<sub>2</sub> = D.** **S<sub>1</sub>**에 의해 정의된 레지스터의 부동소수값은 **S<sub>2</sub>**에 의해 정의된 부동소수값에 의해 곱해지고 그 결과는 **D**에 의해 정의된 레지스터에 저장 됩니다. 모든 원천 데이터들은 부동 소수 형태로 연산되어지고 그 결과는 또한 부동 소수 형태로 저장 되어 집니다.
- 만일 피연산자 **S<sub>1</sub>** 또는 **S<sub>2</sub>**가 상수 **K** 또는 **H**로 표시 되어지면, 그 정수는 자동적으로 뱃셈 연산을 수행하기 위해 이진 부동 소수로 변환 되어 집니다.
- S<sub>1</sub>** 및 **S<sub>2</sub>**는 동일한 레지스터 번호를 정의할 수 있습니다.(동일한 장치가 **S<sub>1</sub>** 및 **S<sub>2</sub>**으로 사용되어 질 수 있습니다. ) 만일 이러한 경우에, **DEMUL** 명령의 지속적인 실행 도중, 레지스터의 데이터는 조건점점(condition contact)이 On 일때, 그 실행주기(Cycle) 동안 모든 스캔 프로그램안에서 한번만 더해 질 수 있습니다. 그 결과, 펄스실행명령(**DEMULP**)이 일반적으로 사용되어 집니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 클 경우 플래그를 자리올림하고 **M1022=On** 입니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 적을 경우 플래그 자리내림하고 **M1022=On** 입니다.
- 만일 변환결과가 0 이면, 제로 플래그이고 **M1020=On** 입니다.

프로그램 예제 1:

X1=On 일때, (D1, D0)의 이진부동소수값이 (D11, D10)의 이진 부동 소수값에 의해 곱해지고 그 결과를 (D21, D20)에 저장합니다.



프로그램 예제 2:

X2=On 일때, (D1, D0)의 이진 부동 소수값이 K1234 (자동적으로 이진부동소수로 변환된)에 의해 곱해지고 그 결과를 (D11, D10)에 저장합니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자		기능			제어기							
123	D	EDIV	P	S <sub>1</sub>	S <sub>2</sub>	D	부동 소수 나눗셈			E S E X S S S A S X S C E H						

유형 연산자	비트 장치				워드 장치												프로그램 단계 DEADD, DEADDP: 13 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
					*	*							*					
					*	*							*					
													*					

펄스								16-비트								32-비트							
E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H	

피연산자:

**S<sub>1</sub>**: 피제수 **S<sub>2</sub>**: 제수 **D**: 몫과 나머지

설명:

- S<sub>1</sub> ÷ S<sub>2</sub> = D**. **S<sub>1</sub>**에 의해 정의된 레지스터의 부동소수값은 **S<sub>2</sub>**에 의해 정의된 부동소수값에 의해 나뉘지고 그 결과는 **D**에 의해 정의된 레지스터에 저장 됩니다. 모든 원천 데이터들은 부동 소수 형태로 연산되어지고 그 결과는 또한 부동 소수 형태로 저장 되어 집니다.
- 만일 피연산자 **S<sub>1</sub>** 또는 **S<sub>2</sub>**가 상수 **K** 또는 **H**로 표시 되어지면, 그 정수는 자동적으로 뱃셈 연산을 수행하기 위해 이진 부동 소수로 변환 되어 집니다.
- 만일 **S<sub>2</sub>**가 0 이면 연산은 실패하고 “연산지 오류”를 발생하고 그 명령은 수행되지 않습니다.  
M1068=ON 일때, D1067 는 오류코드 H'0E19 를 기록 합니다.
- 만일 변화결과와 절대값이 최고 부동소수정보다 클 경우 플래그를 자리올림하고 M1022=On 입니다.
- 만일 변화결과와 절대값이 최고 부동소수정보다 적을 경우 플래그 자리내림하고 M1022=On 입니다.
- 만일 변환결과가 0 이면, 제로 플래그이고 M1020=On 입니다.

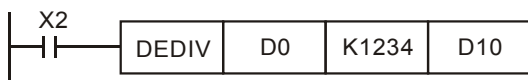
프로그램 예제 1:

X1=On 일때, (D1, D0)의 이진부동소수값을 (D11, D10)의 이진 부동 소수값으로 나누고 그 결과를 (D21, D20)에 저장합니다.



프로그램 예제 2:

X2=On 일때, (D1, D0)의 이진 부동 소수값을 K1234 (자동적으로 이진부동소수로 변환된)으로 나누고 그 결과를 (D11, D10)에 저장합니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
124	D	EXP	P	<b>(S)</b> <b>(D)</b>	부동 소수 지수 연산	E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEXP, DEXPP: 9 단계	
S					*	*							*				
D													*				

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	S H	E S	E X	S S	S A	S X	S C	S H	E S	E X	S S	S A	S X	S C	S H

피연산자:

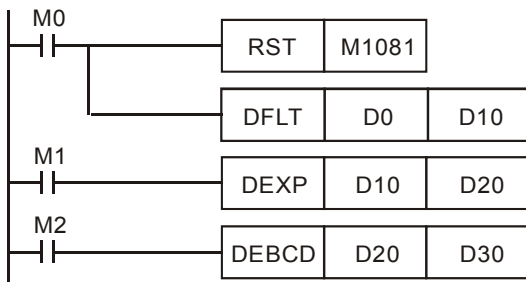
**S:** 피연산자 원시 장치    **D:** 피연산자 결과 장치

설명:

- 예를 들면, 밑  $e=2.71828$  이고 지수는 **S**:
- $EXP[S+1, S]=[D+1, D]$
- S**가 음수이든 양수이든 상관 없습니다. 특정 레지스터 **D**는 32bit 형태와 연산을 위한 부동소수를 사용하기 위해 필요합니다.
- 피연산자  $D=e^S$  일때,  $e=2.71828$  이고 **S**는 특정 원시데이터 입니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 클 경우 플래그를 자리올림하고 **M1022=On** 입니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 적을 경우 플래그 자리내림하고 **M1022=On** 입니다.
- 만일 변환결과가 0 이면, 제로 플래그이고 **M1020=On** 입니다.

프로그램 예제:

- M0=On** 일때, (**D0, D1**)을 이진 부동 소수로 변환하고 레지스터(**D10, D11**)에 저장하시기 바랍니다.
- M1=On** 일때, (**D10, D11**)을 지수 연산을 수행하기 위한 지수로 사용하기 바랍니다. 그 값은 이진 부동 소수이며, 레지스터(**D20, D21**)에 저장하시기 바랍니다.
- M2=On** 일때, (**D20, D21**)의 이진 부동 소수를 십진 부동 소수로 변환하고 레지스터(**D30, D31**)에 저장하시기 바랍니다. (이때, **D31**는 10의 거듭제곱에 대한 **D30**을 의미합니다.)



주의:

부동 소수점의 연산함수에 관해서는 **CH 5.3** 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능										제어기						
125	D	LN	P	<b>S</b> <b>D</b>	부동 자연 대수 연산										E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치												프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DLN, DLNP: 9 단계				
					*	*							*							
													*							

펄스								16-비트								32-비트							
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H			

피연산자:

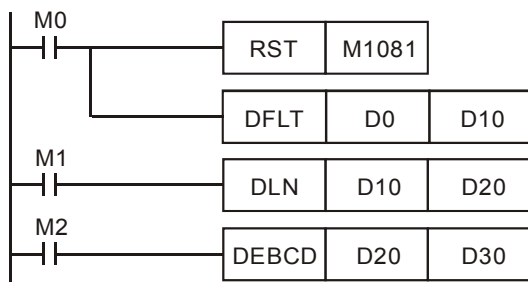
**S**: 피연산자 원시 장치 **D**: 피연산자 결과 장치

설명:

- 예를 들면, 피연산자 **S** 에 대한 자연 대수 **ln** 을 수행:  
 $LN[S+1, S]=[D+1, D]$
- S** 에 대하여 오직 양수만이 가능합니다. 특정 레지스터 **D** 는 32bit 형태와 연산을 위한 부동소수를 사용하기 위해 필요합니다. 그러므로, **S** 는 부동소수로 변환되어야 합니다.
- 피연산자  $e^D=S$  일때, 피연산자 **D**=ln**S** 및 **S** 는 특정 원시데이터 입니다.
- 만일 변화결과의 절대값이 최고 부동소수점보다 클 경우 플래그를 자리올림하고 **M1022=On** 입니다.
- 만일 변화결과의 절대값이 최고 부동소수점보다 적을 경우 플래그 자리내림하고 **M1022=On** 입니다.
- 만일 변환결과가 0 이면, 제로 플래그이고 **M1020=On** 입니다.

프로그램 예제:

- M0=On** 일때, (**D0**, **D1**)을 이진 부동 소수로 변환하고 레지스터(**D10**, **D11**)에 저장하시기 바랍니다.
- M1=On** 일때, (**D10**, **D11**)을 자연 대수 연산을 수행하기 위한 실수로 사용하기 바랍니다. 그 값은 이진 부동 소수이며, 레지스터(**D20**, **D21**)에 저장하시기 바랍니다.
- M2=On** 일때, (**D20**, **D21**)의 이진 부동 소수를 십진 부동 소수로 변환하고 레지스터(**D30**, **D31**)에 저장하시기 바랍니다. (이때, **D31** 는 10 의 거듭제곱에 대한 **D30** 을 의미합니다.)



주의:

부동 소수점의 연산함수에 관해서는 **CH 5.3** 상세 수치 처리를 참조하시기 바랍니다.



API	간략기호			피연산자	기능										제어기						
126	D	LOG	P	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b>	부동 대수 연산										E S	E X	S S	S A	S X	S C	E H

연산자 유형	비트 장치				워드 장치											프로그램 단계 DLOG, DLOGP: 13 단계		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
	S <sub>1</sub>					*	*							*				
	S <sub>2</sub>					*	*							*				
	D													*				

펄스								16-비트								32-비트							
E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H	E S	E X	S S	S A	S X	S C	S H		

피연산자:

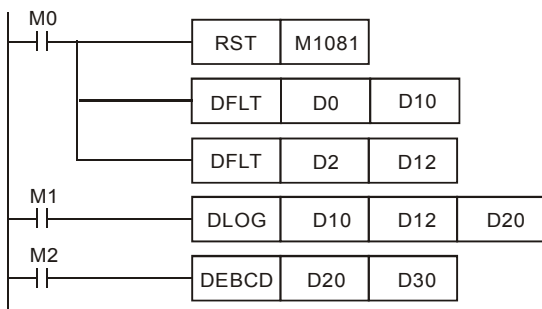
**S<sub>1</sub>**: 피연산자 기준 장치    **S<sub>2</sub>**: 피연산자 원시 장치    **D**: 피연산자 결과 장치

설명:

- S<sub>1</sub>** 및 **S<sub>2</sub>**에 대한 대수 연산을 수행 합니다. 그리고 그 결과를 **D**에 저장합니다.
- S<sub>1</sub>** 및 **S<sub>2</sub>**에 대하여 오직 양수만이 가능합니다. 특정 레지스터 **D**는 32bit 형태와 연산을 위한 부동소수를 사용하기 위해 필요합니다. 그러므로, **S<sub>1</sub>** 및 **S<sub>2</sub>**는 부동소수로 변환되어야 합니다.  
 $S1^D=S2$ 를 보면,  $D=? \rightarrow \log_{S1} S2=D$   
 $S1=5, S2=125$ 를 보면,  $D=\log_5 125=?$   
 $S1^D=S2 \rightarrow 5^D=125 \rightarrow D=\log_5 125=3$
- 만일 변화결과의 절대값이 최고 부동소수정보다 클 경우 플래그를 자리올림하고 M1022=On 입니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 적을 경우 플래그 자리내림하고 M1022=On 입니다.
- 만일 변환결과가 0 이면, 제로 플래그이고 M1020=On 입니다.

프로그램 예제:

- M0=On 일때, (D0, D1) 및 (D2, D3)를 이진 부동 소수로 변환하고 32 비트 레지스터(D10, D11)와(D12, D13)에 각각 저장하시기 바랍니다.
- M1=On 일때, (D10, D11) 및 (D12, D13)의 32 비트 레지스터의 이진 부동 소수를 대수 연산을 수행하기 위하여 사용하고, 32 비트 레지스터(D20, D21)에 저장하시기 바랍니다.
- M2=On 일때, (D20, D21)의 32 비트 레지스터의 이진 부동 소수를 십진 부동 소수로 변환하고 레지스터(D30, D31)에 저장하시기 바랍니다. (이때, D31은 10의 거듭제곱에 대한 D30을 의미합니다.)



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능										제어기						
127	D	ESQR	P	<b>S</b> <b>D</b>	이진 부동 소수의 제곱근										E S	E X	S S	S A	S X	S C	E H

연산자 /  유형	비트 장치				워드 장치										프로그램 단계						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DESQR, DESQRP: 9 단계					
					*	*							*								
													*								

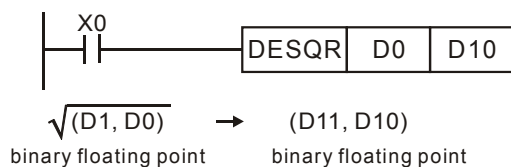
펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

**설명:**

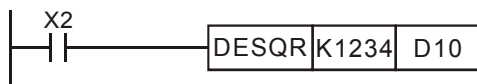
- S**: 제곱근에 의해 수행되어지는 원시 장치. **D**: 제곱근 연산의 결과
- 이 명령어는 원시장치 **S**의 부동 소수값에 대한 제곱근 연산을 수행 하고 목적장치 **D**에 그 결과를 저장합니다. 모든 원시 데이터들은 부동 소수 형태로 연산되어 지고 그 결과는 또한 부동 소수 형태로 저장 되어 집니다.
- 만일 피연산자 **S<sub>1</sub>** 또는 **S<sub>2</sub>**가 상수 **K** 또는 **H**로 표시 되어지면, 그 정수는 자동적으로 덧셈 연산을 수행하기 위해 이진 부동 소수로 변환 되어 집니다.
- 만일 **D**의 연산 결과가 0 이면, M1020=On 플래그 없음 하시기 바랍니다.
- S**는 오직 양수일 수 있습니다. 음수에 대한 제곱근 연산을 수행하는 것은 “연산 오류”의 결과를 가져오고 이 명령은 실행되지 않습니다. M1067 및 M1068은 On 이 되며, 오류 코드 “0E1B”가 D1067에 기록되어 집니다.

**프로그램 예제 1:**

X0=On 일때, 이진 부동 소수 (D1, D0)의 제곱근은 제곱근 연산 후 (D11, D10)에 의해 정의된 레지스터에 저장 됩니다.

**프로그램 예제 2:**

X2=On 일때, K1234 (자동적으로 이진부동소수로 변환된)의 제곱근은(D11, D10)에 저장 됩니다.

**주의:**

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
128	D	POW	P	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b>	부동 소수 거듭제곱 연산	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DPOW, DPOWP: 13 단계	
S <sub>1</sub>					*	*							*				
S <sub>2</sub>					*	*							*				
D													*				

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

피연산자:

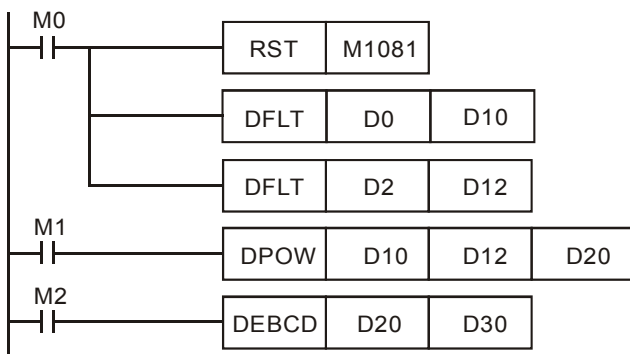
**S<sub>1</sub>**: 기본 장치.    **S<sub>2</sub>**: 지수.    **D**: 결과

설명:

- 이진 부동 소수 **S<sub>1</sub>** 및 **S<sub>2</sub>**에 대한 거듭제곱 연산을 수행 합니다. 그리고 그 결과를 **D**에 저장합니다.  
POW [**S<sub>1</sub>+1, S<sub>1</sub>**][**S<sub>2</sub>+1, S<sub>2</sub>**]= **D**
- S<sub>1</sub>**은 오직 양수만 가능하며, **S<sub>2</sub>**의 내용은 양수, 음수 둘 다 가능 합니다. 특정 레지스터 **D**는 32bit 형태와 연산을 위한 부동소수를 사용하기 위해 필요합니다. 그러므로, **S<sub>1</sub>** 및 **S<sub>2</sub>**는 부동소수로 변환되어져야 합니다.  
S1<sup>S2</sup>=D 일때, D=?  
S1=5,S2=3 이면, D=53=?  
D=53=125
- 오류 플래그 M1067 및 M1068 는 D1067 및 D1068 를 읽습니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 클 경우 플래그를 자리올림하고 M1022=On 입니다.
- 만일 변화결과의 절대값이 최고 부동소수정보다 적을 경우 플래그 자리내림하고 M1022=On 입니다.
- 만일 변환결과가 0 이면, 제로 플래그이고 M1020=On 입니다.

프로그램 예제:

- M0=On 일때, (D0, D1) 및 (D2, D3)를 이진 부동 소수로 변환하고 레지스터(D10, D11)와(D12, D13)에 각각 저장하시기 바랍니다.
- M1=On 일때, (D10, D11) 및 (D12, D13)의 32 비트 레지스터의 이진 부동 소수를 거듭제곱 연산을 수행하기 위하여 사용하고, 32 비트 레지스터(D20, D21)에 저장하시기 바랍니다.
- M2=On 일때, (D20, D21)의 32 비트 레지스터의 이진 부동 소수를 십진 부동 소수로 변환하고 레지스터(D30, D31)에 저장하시기 바랍니다. (이때, D31 는 10 의 거듭제곱에 대한 D30 을 의미합니다.)



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
129	D	INT	P	<b>S</b> <b>D</b>	부동 소수 정수화	E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S													*			INT, INTP: 5 단계	
D													*			DINT, DINTP: 9 단계	

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

피연산자:

**S**: 원시 장치(2 개의 연속 장치를 점유)    **D**: 결과를 저장하는 목적 장치

설명:

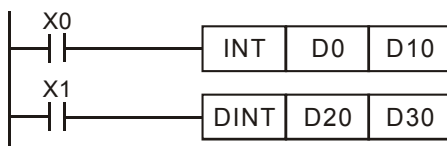
- S**에 의해 정의된 레지스터의 이진 부동 소수값은 BIN 정수로 변환되어 **D**로 정의된 레지스터에 저장되어 집니다. BIN 정수의 소수는 버려 집니다.
- 이 명령은 API 49 (FLT) 명령의 역입니다.
- 만일 **D**의 연산 결과가 0 이면, M1020=On 플래그 없음 하시기 바랍니다.  
만일 소수가 버려지게 되면, 자리내림플래그 M1021 은 On 이 됩니다.  
만일 결과값이 아래의 설정범위를 넘어서면(오버플로우가 발생하면), 자리올림플래그 M1022 은 On 이 됩니다.

16 비트 명령 : -32,768~32,767

32 비트 명령 : -2,147,483,648~2,147,483,647

프로그램 예제:

- X0=On 일때, (D1, D0)의 이진 부동 소수값은 BIN 정수로 변환되고 그 결과는 (D10)에 저장 됩니다.  
BIN 정수의 소수는 버려집니다.
- X1=On 일때, (D21, D20)의 이진 부동 소수값은 BIN 정수로 변환되고 그 결과는 (D31, D30)에 저장 됩니다.  
BIN 정수의 소수는 버려집니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능										제어기						
130	D	SIN	P	<b>(S)</b> <b>(D)</b>	부동 소수의 사인 연산										E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치												프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
	S				*	*							*					
	D												*					

펄스												16-비트					32-비트				
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	

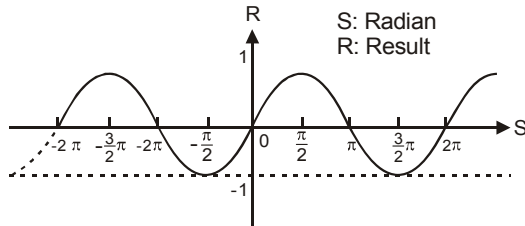
피연산자:

**S**: 정의된 RAD 값 ( $0^\circ \leq S < 360^\circ$ ) **D**: 연산된 결과가 저장되는 장소

설명:

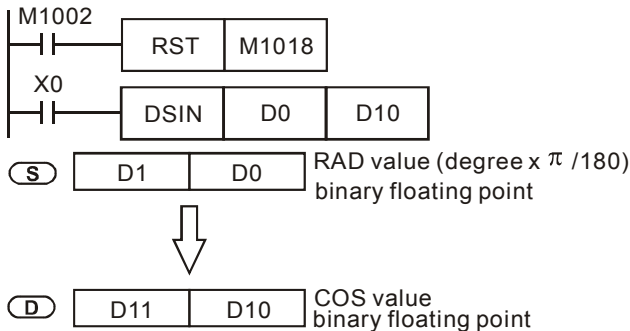
- S**에 의해서 지정된 원시값은 플래그 **M1018**에 의해 라디안이나 각도가 될 수 있습니다.
- M1018=Off** 일때, 라디안 모드로 설정됩니다.  $RAD = \text{각도} \times \pi / 180$ .
- M1018=On** 일때, 각도 모드로 설정됩니다. 각도 범위:  $0^\circ \leq \text{각도} < 360^\circ$ .
- 계산 결과가 0 이면, **M1020=ON** 이 됩니다.
- S**에 의해서 정의된 각도 데이터의 **SIN** 값은 계산 되어져서 그 연산 결과는 **D**에 의해 정의된 레지스터에 저장 됩니다.

Following shows the relation between radian and result:



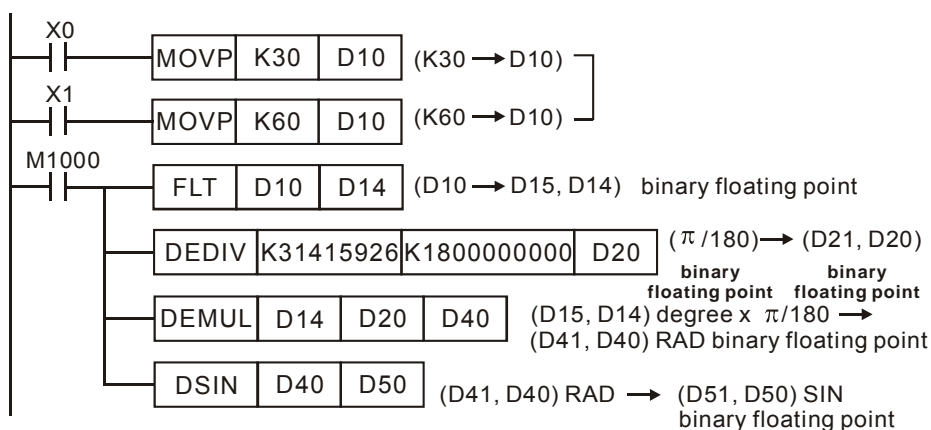
프로그램 예제 1:

**M1018=Off** 일 때, 라디안 모드 입니다. **X0=On** 일때, RAD 값을 (**D1, D0**)으로 정의 하십시오. 각도의 **SIN** 값을 계산하고 그 결과를(**D11, D10**)에 저장 하십시오. (**D11, D10**)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



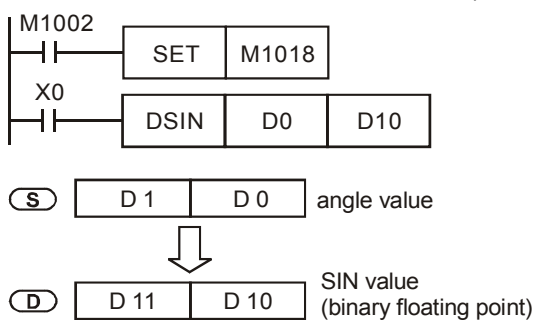
프로그램 예제 2:

**M1018=Off** 일 때, 라디안 모드 입니다. **Select angle** 입력 **X0** 및 **X1** 로 부터 각도를 선택하고 **SIN** 값을 계산하기 위해 그것을 **RAD** 값으로 변환 하십시오.



### 프로그램 예제 3:

M1018=On 일 때, 각도 모드입니다. X0=On 일 때, (D1, D0)의 각도값을 지정합니다. 각도의 범위는  $0^{\circ} \leq$  각도값  $< 360^{\circ}$  입니다. SIN 값으로 변환 후(D11, D10)에 이진 부동 소수로 저장 하시기 바랍니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
131	D	COS	P	<b>(S)</b> <b>(D)</b>	부동 소수의 코사인 연산	E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치												프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		DCOS, DCOSP: 9 단계	
S					*	*							*					
D													*					

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

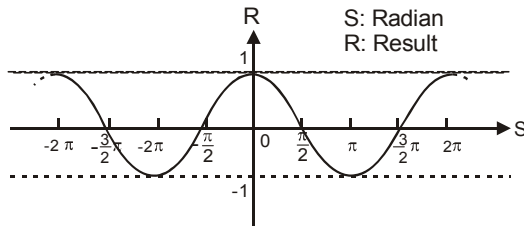
피연산자:

**S**: 정의된 RAD 값( $0^\circ \leq S < 360^\circ$ ) **D**: 연산된 결과가 저장되는 장소

설명:

- S**에 의해서 지정된 원시값은 플래그 M1018에 의해 라디안이나 각도가 될 수 있습니다.
- M1018=Off 일때, 라디안 모드로 설정됩니다.  $RAD = \text{각도} \times \pi / 180$ .
- M1018=On 일때, 각도 모드로 설정됩니다. 각도 범위:  $0^\circ \leq \text{각도} < 360^\circ$ .
- 계산 결과가 0이면, M1020=ON 이 됩니다.
- S**에 의해서 정의된 각도 데이터의 COS 값은 계산 되어져서 그 연산 결과는 **D**에 의해 정의된 레지스터에 저장 됩니다.

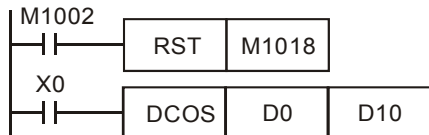
Following shows the relation between radian and result:



- 플래그 M1018 라디안/각도 스위치: M1018=Off 일 때, **S**은 RAD 값 입니다. M1018=On 일 때, **S**는 각도값 입니다. (0-360).

프로그램 예제 1:

M1018=Off 일 때, 라디안 모드 입니다. X0=On 일때, RAD 값을 (D1, D0)으로 정의 하십시오. 각도의 COS 값을 계산하고 그 결과를(D11, D10)에 저장 하십시오. (D1, D0)에 있는 값과 (D11, D10)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



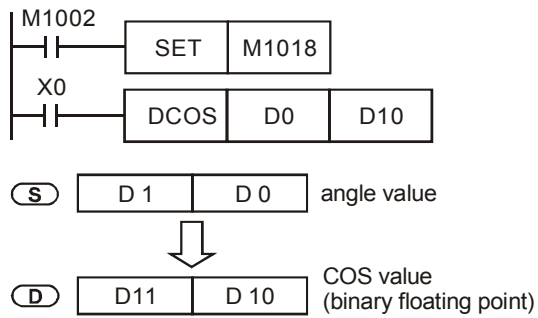
**(S)** D1 D0 RAD value (degree x  $\pi / 180$ )  
binary floating point



**(D)** D11 D10 COS value  
binary floating point

프로그램 예제 2:

M1018=On 일 때, 각도 모드 입니다. X0=On 일 때, 그것은 특정(D1, D0)의 각도값 입니다. 각도의 범위는  $0^{\circ} \leq \text{각도} < 360^{\circ}$  입니다. COS 값으로 변환 후(D11, D10) 에 이진 부동 소수로 저장 하시기 바랍니다.



**주의:**

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.



API	간략기호			피연산자	기능	제어기						
132	D	TAN	P	<b>(S)</b> <b>(D)</b>	부동 소수의 탄젠트 연산	E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DTAN, DTANP: 9 단계	
S					*	*							*				
D													*				

펄스							16-비트					32-비트				
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S

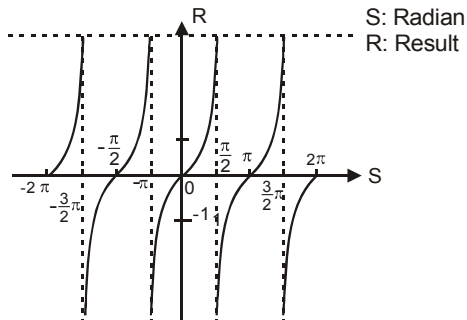
피연산자:

**S**: 정의된 RAD 값( $0^\circ \leq S < 360^\circ$ ) **D**: 연산된 결과가 저장되는 장소

설명:

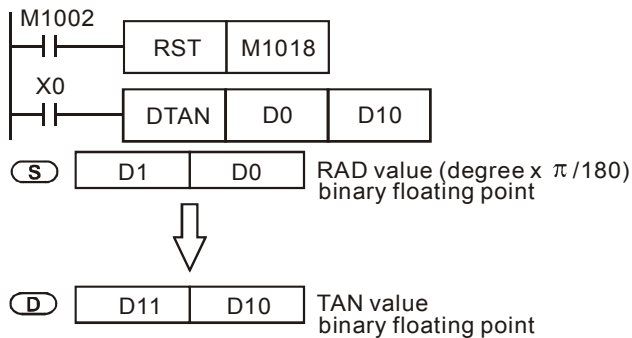
- S**에 의해서 지정된 원시값은 플래그 **M1018**에 의해 라디안이나 각도가 될 수 있습니다.
- M1018=Off** 일때, 라디안 모드로 설정됩니다.  $RAD = \text{각도} \times \pi / 180$ .
- M1018=On** 일때, 각도 모드로 설정됩니다. 각도 범위:  $0^\circ \leq \text{각도} < 360$ .
- 계산 결과가 0 이면, **M1020=ON** 이 됩니다.
- S**에 의해서 정의된 각도 데이터의 **TAN** 값은 계산 되어져서 그 연산 결과는 **D**에 의해 정의된 레지스터에 저장 됩니다.

Following shows the relation between radian and result:



#### 프로그램 예제 1:

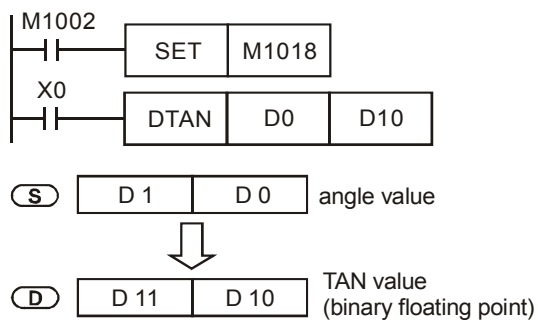
**M1018=Off** 일 때, 라디안 모드 입니다. **X0=On** 일때, RAD 값을 (D1, D0)으로 정의 하십시오. 각도의 TAN 값을 계산하고 그 결과를(D11, D10)에 저장 하십시오. (D1, D0)에 있는 값과 (D11, D10)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



#### 프로그램 예제 2:

**M1018=On** 일 때, 각도 모드 입니다. **X0=On** 일 때, 그것은 특정(D1, D0)의 각도값 입니다. 각도의 범위는  $0^\circ$

$\leq$ 각도  $< 360^\circ$  입니다. TAN 값으로 변환 후(D11, D10) 에 이진 부동 소수로 저장 하시기 바랍니다.



#### 주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
133	D	ASIN	P	<b>(S)</b> <b>(D)</b>	부동 소수의 아크사인 연산	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DASIN, DASINP: 9 단계	
S					*	*							*				
D													*				

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	S H	E S	E X	S S	S A	S X	S C	S H	E S	E X	S S	S A	S X	S C	S H

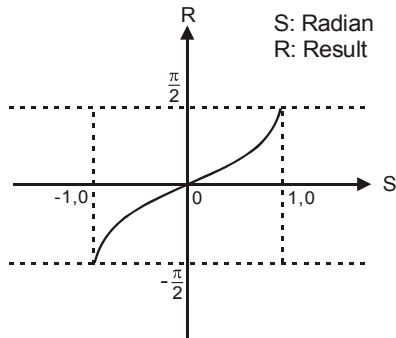
피연산자:

**S**: 특정소스(이진 부동 소수)    **D**: 연산된 결과가 저장되는 장소

설명:

1. ASIN 값 =  $\text{SIN}^{-1}$

Following shows the relation between radian and result:

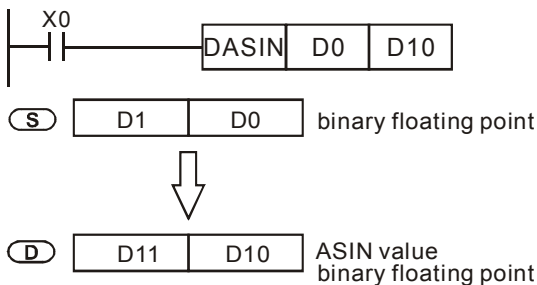


피연산자 **S**에 의해 표시되어 지고, 사인값의 십진 부동 소수값은  $-1.0 \sim +1.0$ 의 범위에 있어야 합니다. 그 값이 제한된 범위를 넘으면, M1067 및 M1068은 어떠한 동작없이 ON이 됩니다.

만일 변환값이 0이면, 제로플래그 M1020은 ON이 됩니다.

프로그램 예제:

X0=On일 때, 이진 부동 소수(D1, D0)을 정의 하십시오. ASIN 값을 계산하고 그 결과를(D11, D10)에 저장 하십시오. (D11, D10)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
134	D	ACOS	P	<b>(S)</b> <b>(D)</b>	부동 소수의 아크코사인 연산	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치												프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		DACOS, DACOSP: 9 단계	
S					*	*							*					
D													*					

펄스								16-비트								32-비트							
E S	E X	S S	S A	S X	S C	E H		E S	E X	S S	S A	S X	S C	E H		E S	E X	S S	S A	S X	S C	E H	

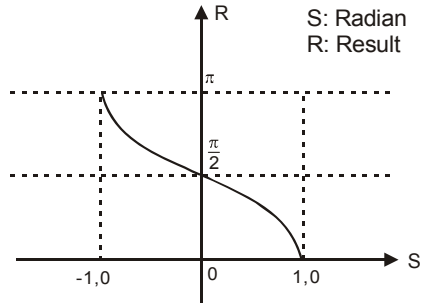
피연산자:

**S:** 특정소스(이진 부동 소수)    **D:** 연산된 결과가 저장되는 장소

설명:

1.  $ACOS \text{ 값} = \cos^{-1}$

Following shows the relation between radian and result:

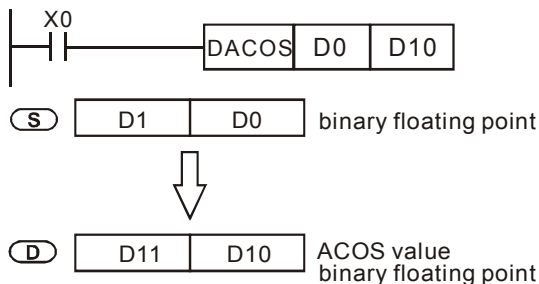


피연산자 **S**에 의해 표시되어 지고, 사인값의 십진 부동 소수값은  $-1.0 \sim +1.0$ 의 범위에 있어야 합니다. 그 값이 제한된 범위를 넘으면, M1067 및 M1068은 어떠한 동작없이 ON이 됩니다.

만일 변환값이 0이면, 제로플래그 M1020은 ON이 됩니다.

프로그램 예제:

X0=On일 때, 이진 부동 소수(D1, D0)을 정의 하십시오. ACOS 값을 계산하고 그 결과를(D11, D10)에 저장 하십시오. (D11, D10)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능	제어기						
135	D	ATAN	P	<b>(S)</b> <b>(D)</b>	부동 소수의 아크탄젠트 연산	E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DATAN, DATANP: 9 단계	
S					*	*							*				
D													*				

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

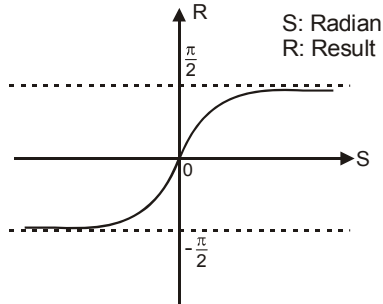
피연산자:

**S:** 특정소스(이진 부동 소수)    **D:** 연산된 결과가 저장되는 장소

설명:

1.  $ATAN \text{ 값} = \tan^{-1}$

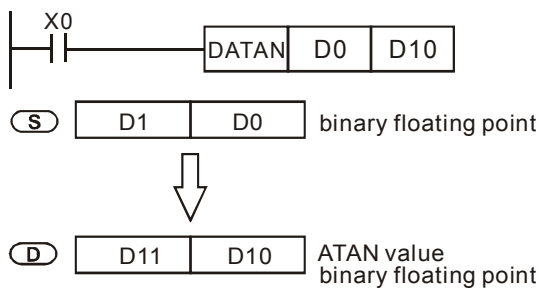
Following shows the relation between radian and result:



만일 변환값이 0 이면, 제로플래그 M1020 는 ON 이 됩니다.

프로그램 예제:

X0=On 일 때, 이진 부동 소수(D1, D0)을 정의 하십시오. ATAN 값을 계산하고 그 결과를(D11, D10)에 저장 하십시오. (D11, D10)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



주의:

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자	기능										제어기						
136	D	SINH	P	<b>S</b> <b>D</b>	이진 부동 소수의 쌍곡선 사인 연산										E S	E X	S S	S A	S X	S C	E H
연산자	비트 장치				워드 장치										프로그램 단계						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DSINH, DSINHP: 9 단계					
					*	*							*								
S																					
D																					

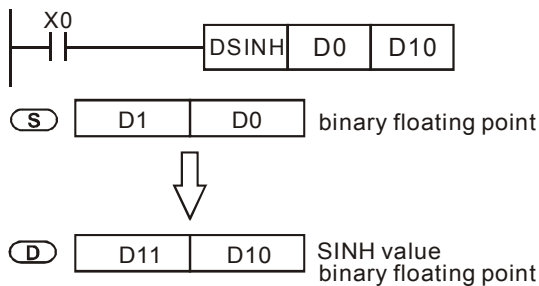
펄스								16-비트								32-비트							
E	E	S	S	S	S	E		E	E	S	S	S	S	E		E	E	S	S	S	S	E	
S	X	S	A	X	C	H		S	X	S	A	X	C	H		S	X	S	A	X	C	H	

**설명:**

- S**: 특정소스(이진 부동 소수) **D**: 연산된 결과가 저장되는 장소
- $SINH 값 = (e^S - e^{-S})/2$

**프로그램 예제:**

X0=On 일 때, 이진 부동 소수(D1, D0)을 정의 하십시오. **SINH** 값을 계산하고 그 결과를(D11, D10)에 저장 하십시오. (D11, D10)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



만일 변환 결과의 절대값이 표시되어 질 수 있는 가장 큰 부동 소수 보다 클 경우 자리올림 플래그 **M1022** 는 **ON** 이 됩니다.

만일 변환 결과의 절대값이 표시되어 질 수 있는 가장 적은 부동 소수 보다 적을 경우 자리내림 플래그 **M1021** 는 **ON** 이 됩니다

만일 변환값이 0 이면, 제로플래그 **M1020** 는 **ON** 이 됩니다.

**주의:**

부동 소수점의 연산함수에 관해서는 **CH 5.3** 상세 수치 처리를 참조하시기 바랍니다.

API	간략기호			피연산자		기능										제어기						
137	D	COSH	P	<b>S</b>	<b>D</b>	이진 부동 소수의 쌍곡선 코사인 연산										E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계				
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCOSH, DCOSHP: 9 단계				
S					*	*							*							
D													*							

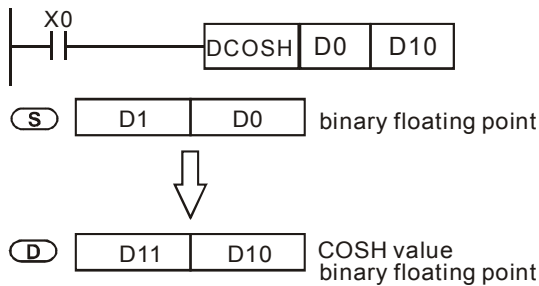
펄스								16-비트								32-비트							
E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H		E S	E X	S S	S A	S X	S C	S H	

**설명:**

1. **S**: 특정소스(이진 부동 소수) **D**: 연산된 결과가 저장되는 장소
2.  $\text{COSH 값} = (e^s + e^{-s})/2$

**프로그램 예제:**

X0=On 일 때, 이진 부동 소수(D1, D0)을 정의 하십시오. COSH 값을 계산하고 그 결과를(D11, D10)에 저장 하십시오. (D11, D10)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



만일 변환 결과의 절대값이 표시되어 질 수 있는 가장 큰 부동 소수 보다 클 경우 자리올림 플래그 M1022 는 ON 이 됩니다.

만일 변환 결과의 절대값이 표시되어 질 수 있는 가장 적은 부동 소수 보다 적을 경우 자리내림 플래그 M1021 는 ON 이 됩니다

만일 변환값이 0 이면, 제로플래그 M1020 는 ON 이 됩니다.

**주의:**

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.

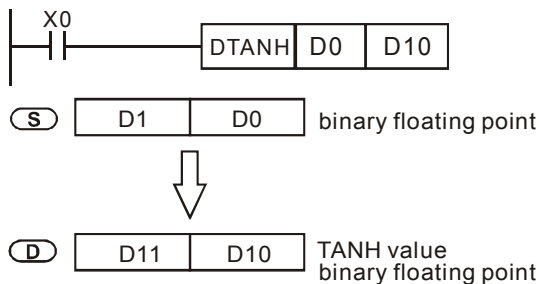
API	간략기호			피연산자	기능										제어기																						
138	D	TANH	P	<b>(S)</b> <b>(D)</b>	이진 부동 소수의 쌍곡선 탄젠트 연산										E S	E X	S S	S A	S X	S C	E H																
유형 연산자	비트 장치				워드 장치										프로그램 단계																						
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DTANH, DTANHP: 9 단계																					
					*	*							*																								
													*																								
																	펄스					16-비트					32-비트										
																	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H

**설명:**

- S:** 특정소스(이진 부동 소수) **D:** 연산된 결과가 저장되는 장소
- $TANH 값 = (e^s - e^{-s}) / (e^s + e^{-s})$

**프로그램 예제:**

X0=On 일 때, 이진 부동 소수(D1, D0)을 정의 하십시오. COSH 값을 계산하고 그 결과를(D11, D10)에 저장 하십시오. (D11, D10)에 저장된 결과는 모두 이진 부동 소수 형태 입니다.



만일 변환 결과의 절대값이 표시되어 질 수 있는 가장 큰 부동 소수 보다 클 경우 자리올림 플래그 M1022 는 ON 이 됩니다.

만일 변환 결과의 절대값이 표시되어 질 수 있는 가장 적은 부동 소수 보다 적을 경우 자리내림 플래그 M1021 는 ON 이 됩니다

만일 변환값이 0 이면, 제로플래그 M1020 는 ON 이 됩니다.

**주의:**

부동 소수점의 연산함수에 관해서는 CH 5.3 상세 수치 처리를 참조하시기 바랍니다.





API	간략기호	피연산자	기능	제어기						
144	GPWM	<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b> <b>(D)</b>	일반 PWM 출력	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GPWM: 7 단계	
S <sub>1</sub>													*				
S <sub>2</sub>													*				
D		*	*	*													

피연산자:

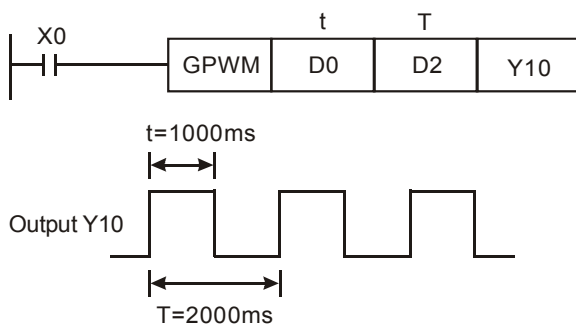
**S<sub>1</sub>**: 펄스 출력폭    **S<sub>2</sub>**: 펄스 출력주기(3 개의 장치를 점유)    **D**: 펄스 출력장치

설명:

1. **S<sub>1</sub>** 은 펄스 출력폭으로 정의 합니다. t:0~32,767ms.
2. **S<sub>2</sub>** 는 펄스 출력주기로 정의 합니다. T:1~32,767ms, **S<sub>1</sub>** ≤ **S<sub>2</sub>**.
3. 시스템에 대하여 **S<sub>2</sub> + 1** 및 **S<sub>2</sub> + 2** 는 사용하지 마시기 바랍니다.
4. **D** 는 펄스 출력 장치 입니다: Y, M 및 S.
5. GPWM 명령이 실행 됐을 때, 펄스 출력폭 **S<sub>1</sub>** 및 펄스 출력주기 **S<sub>2</sub>** 는 펄스 출력장치 **D** 를 통해서 나오는 출력 입니다.
6. **S<sub>1</sub>** ≤ 0 일 때, 펄스 출력장치로 부터의 펄스 출력은 없습니다. **S<sub>1</sub>** ≥ **S<sub>2</sub>** 일 때, 펄스 출력장치는 항상 On 입니다.
7. **S<sub>1</sub>** 및 **S<sub>2</sub>** 는 PWM 명령을 실행시 변경 되어 질 수 있습니다.

프로그램 예제:

X0=On 일 때, D0=K1000 와 D2=K2000 및 Y10 는 아래와 같은 펄스를 출력합니다. X0=Off 일 때, Y10 출력은 항상 Off 입니다.



설명:

1. 이 명령은 스캔주기에 의해 계수 되서 최대 오프셋은 PLC 스캔주기가 됩니다. **S<sub>1</sub>** 과 **S<sub>2</sub>** 및 (**S<sub>2</sub> - S<sub>1</sub>**)의 값은 PLC 스캔주기보다 반드시 커야 합니다. 그렇지 않으면, GPWM 출력에 오류가 발생 합니다.
2. 이 명령을 서브루틴이나 인터럽트에 사용한다면, GPWM 출력이 정확하지 않을 수도 있음을 주지하시기 바랍니다.

API	간략기호	피연산자	기능	제어기						
145	FTC	<b>S<sub>1</sub></b> <b>S<sub>2</sub></b> <b>S<sub>3</sub></b> <b>D</b>	퍼지 온도 제어	E S	E X	S S	S A	S X	S C	E H

연산자 유형	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FTC: 9 단계	
S <sub>1</sub>					*	*							*				
S <sub>2</sub>					*	*							*				
S <sub>3</sub>													*				
D													*				

피연산자:

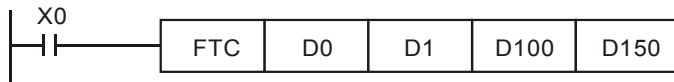
**S<sub>1</sub>**: 목표값(SV)    **S<sub>2</sub>**: 현재 측정값 (PV)    **S<sub>3</sub>**: 파라미터 (연속된 7 개의 장치를 점유)    **D**: 출력값 (MV)

설명:

- 0.1°C ~500°C 를 보기 위한 피연산자 S<sub>1</sub> 가용범위는 1~5000 입니다. 기본 단위는 0.1°C 입니다. (S<sub>3</sub> +1) (각주 참조)는 0.1°C~500°C 를 보여주기 위해 K0 로 설정 합니다.
- 0.1°C ~500°C 를 보기 위한 피연산자 S<sub>2</sub> 가용범위는 1~5000 입니다. 기본 단위는 0.1°C 입니다. (S<sub>3</sub> +1) (각주 참조)는 0.1°C~500°C 를 보여주기 위해 K0 로 설정 합니다.
- 그러므로, 사용자가 온도센서로부터 얻은 아날로그값을 디지털로 변환시킨 결과를 얻을 때, 그 값을 4 개의 기본적인 연산을 사용하여 1~5000 의 범위의 값으로 변환시킬 필요가 있습니다.
- S<sub>3</sub> 는 샘플링시간 설정 입니다. 만일 설정값이 K1 보다 적을 경우, 명령은 작동하지 않습니다. 만일 설정값이 K200 보다 클 경우, 그값은 K200 으로 간주 됩니다.
- 파라미터 설정시, “bit0=0”는 °C 를 의미하고, “bit0=1”는 °F 를 의미하며, “bit1=0”는 필터함수가 없음을 의미하고, “bit2~bit5”는 4 가지 유형의 외부환경가열설정(Environment Heating Setting)을 의미합니다. “bit6~bit15”는 예약되어 있습니다. 주석을 참조하시기 바랍니다.
- 피연산자의 가용범위는 0 ~ (샘플링시간\*100) 값 입니다. FTC 명령어를 사용할때, 사용자는 가열기의 유형에 의거하여 기타 필수적인 명령어들을 FTC 와 함께 사용 하여야 합니다. 예를 들면, 펄스출력을 제어하기 위해 FTC 와 함께 GPWM 명령어를 사용한다면, (샘플링시간\*100)은 GPWM 의 출력주기가 될 것이며, 출력값 MV 는 GPWM 의 펄스폭이 될 것 입니다. (예제 1 참조)
- 사용자는 이 명령을 사용시 가열기의 유형에 따라 기타 명령들을 함께 사용하여야 합니다. 예를 들면, 그것은 각주에 있는 바와 같이 펄스출력을 제어하기 위해 GPWM 명령과 함께 사용 할 수 있습니다. (예제 1)
- FTC 명령에 사용상 제약은 없습니다. 그러나, 표시된 피연산자는 반복적으로 오류가 발생하는 것을 막을 수는 없습니다.

프로그램 예제:

- FTC 명령을 실행시키기전에 파라미터 설정을 마치시기 바랍니다.
- X0=On 일 때, 명령은 실행되어지고 결과는 D150 에 저장 됩니다. X0=Off 일 때, 명령은 실행되지 않고 이전 데이터는 바뀌지 않습니다.

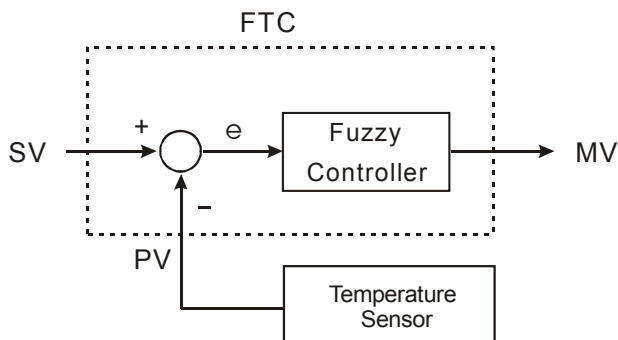


주의:

1. **S<sub>3</sub>**의 설정은 아래와 같습니다:

장치번호	기능	설정범위	설명
<b>(S<sub>3</sub>)</b> :	샘플링율 (T <sub>s</sub> ) (단위: 100ms)	1~200 (단위: 100ms)	만일 T <sub>s</sub> 가 스캔시간보다 적을 경우, PID 명령은 스캔시간으로서 T <sub>s</sub> 를 극복할 것입니다. 만일 T <sub>s</sub> =0 이면, PID 명령은 작동하지 않을 것입니다. 즉, 최소한의 T <sub>s</sub> 값은 스캔시간보다는 커야 합니다.
<b>(S<sub>3</sub>)</b> +1 :	b0: 온도 단위 b1: 여과 함수 b2~b5 : 가열환경선택 b6~b15 : 예약됨	b0 =0 는 °C 를 의미 b0 =1 는 °F 를 의미	설정값이 상한값보다 클 경우, 범위의 최대치를 사용하세요.
		b1=0 는 여과함수 없음 b1=1 는 여과함수 있음	여과함수가 없다면, PV=현재 테스트된값 입니다. 여과함수가 있다면, PV=(현재 테스트된값+이전 PV) /2 입니다.
		b2=1	저속가열선택
		b3=1	일반가열선택
		b4=1	급속가열선택
		b5=1	고속가열선택
<b>(S<sub>3</sub>)</b> +2 : <b>(S<sub>3</sub>)</b> +6 :	시스템 파라미터 사용 전용		

2. 제어 다이어그램:



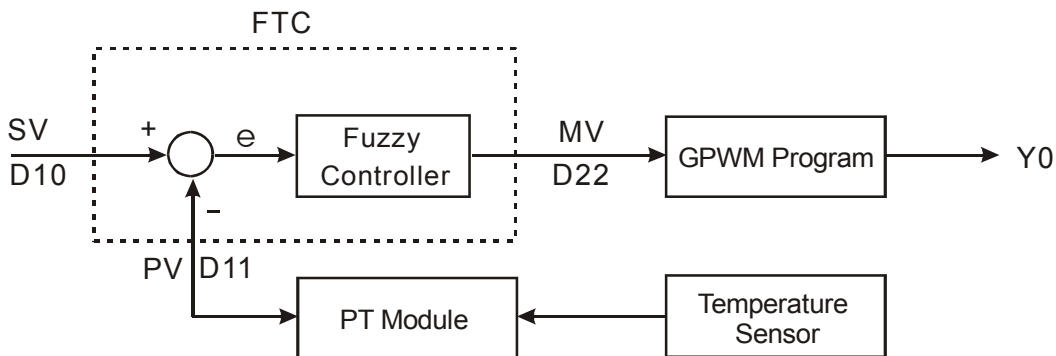
주의 및 제안:

좀 더 나은 온도제어를 하기 위해 샘플링시간을 온도센서의 샘플링시간보다 두배 또는 그 이상으로 설정하시기를 권고 합니다.

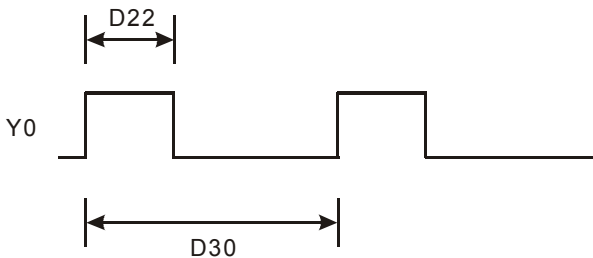
장치  $S_3+1$  의 Bit2~Bit5 는 FTC 명령의 응답속도를 제어하기 위한 선택들을 제공 합니다. 만일 사용자가 파라미터를 설정하지 않거나, 선택 할 줄 모른다면, FTC 명령은 자동적으로 일반가열선택을 작동시킵니다. 만일 주변온도의 변화가 느리다면, 사용자는 목표온도에 도달하는 시간을 개선하기 위해 저속가열선택을 사용 할 수 있습니다. 반대로, 주변온도가 급격히 변화한다면, 사용자는 온도제어의 응답속도를 완화시키기 위해 급속가열선택을 사용 할 수 있습니다.

장치  $S_3+1$  의 Bit2~Bit5 가 모두 1 로 설정되거나 1 개이상 선택이 될 경우, FTC 명령은 Bit2 부터 Bit 5 를 차례대로 체크 합니다. 명령이 1 로 설정된 선택설정을 체크 할때, 그 선택은 즉시 작동 됩니다. 또한 선택은 작동중에 변경될 수 있습니다.

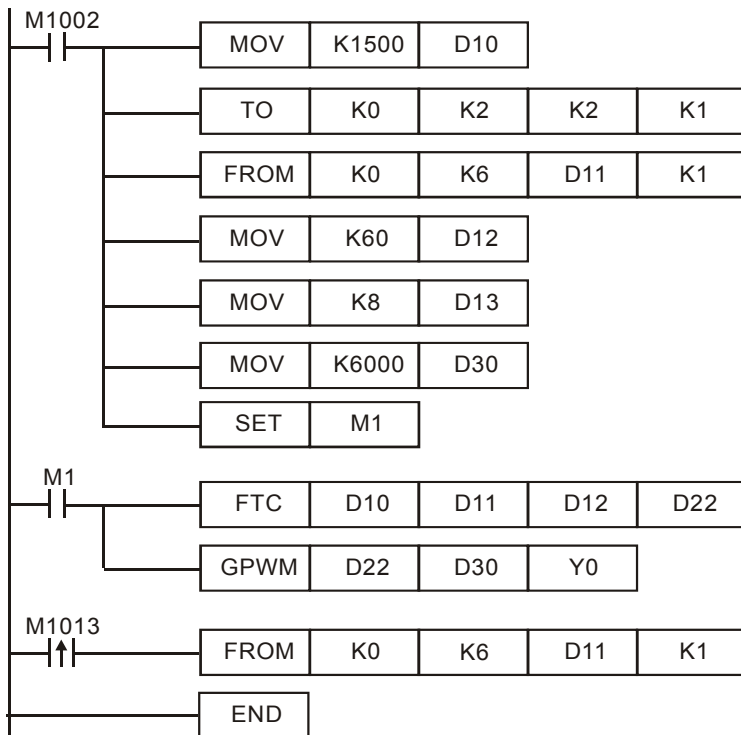
### 3. 예제 1: 제어 다이어그램



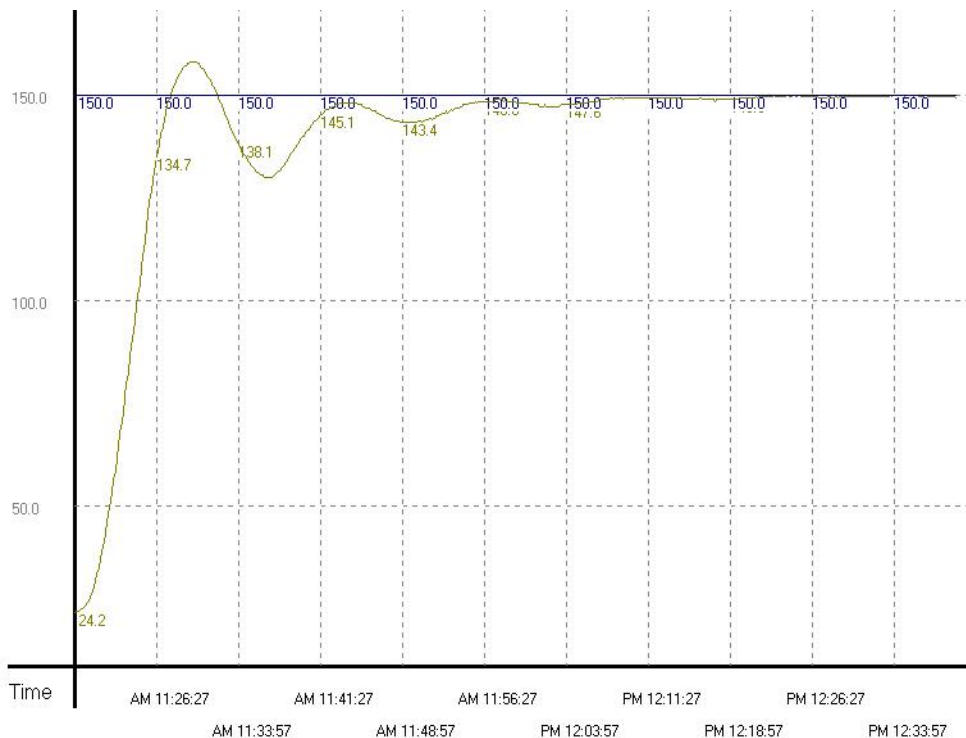
FTC 명령 출력 D22 (MV)는 GPWM 명령의 입력 D22 입니다. 그것은 펄스의 총격계수(duty cycle)를 조정하는데 사용됩니다. D30 는 펄스의 고정총격계수이고, Y0 의 출력순차도는 아래와 같습니다:



예제 1: FTC 명령을 위해 , D10=k1500 (목표온도), D12=k60 (샘플링율: 6 sec.), D13=k8 (Bit3=1) and D30=k6,000 (=D12\*100)으로 설정하시기 바랍니다. 프로그램의 내용은 아래와 같습니다:

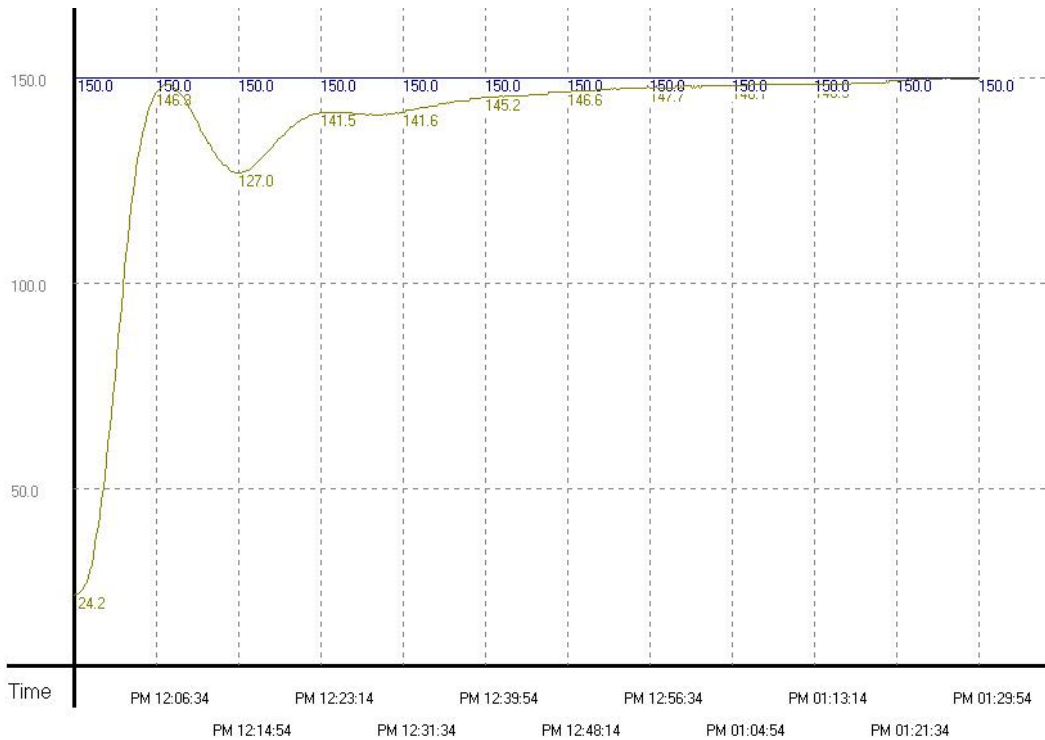


실험환경: 오븐 (온도는 250℃ 까지 입니다). 아래의 그림은 목표온도와 실제온도를 표시하고 있습니다. 목표온도를  $\pm 1^{\circ}\text{C}$  편차로 도달할때까지 약 48 분이 걸렸으며, 상한보다 약 10℃ 정도 오버 하였습니다.



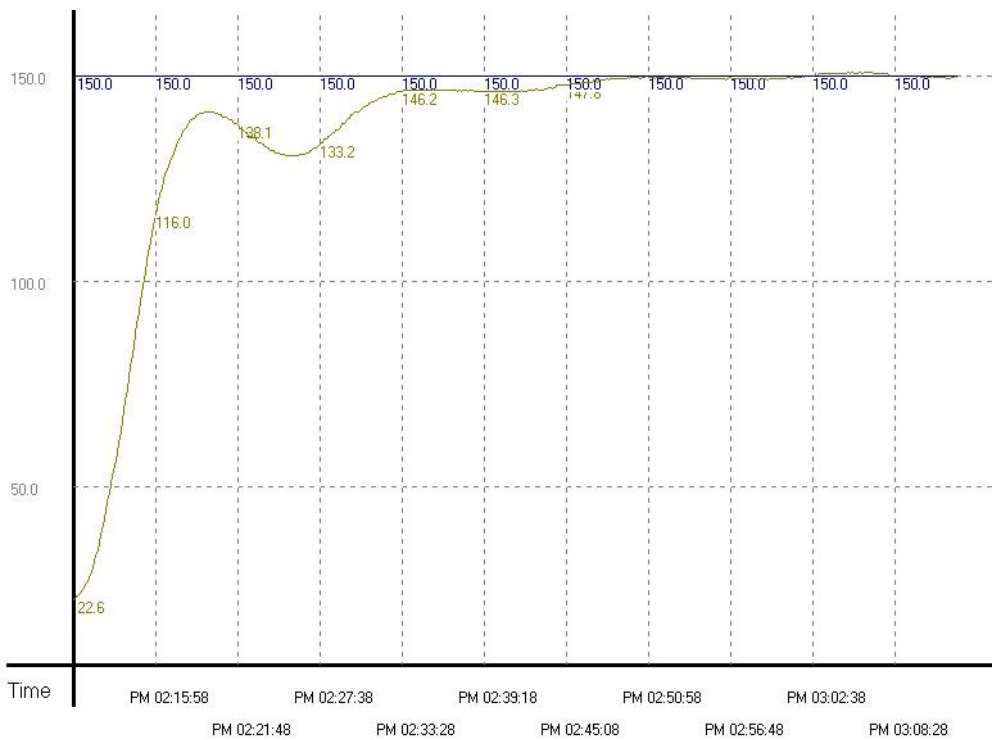
예제 2: 온도가 급격히 치솟는 경향때문에 급속가열선택(D13=k16)으로 제어방법을 변경. 실험결과 아래와 같습니다:

아래 그림에 의하면, 목표온도를  $\pm 1^{\circ}\text{C}$  편차로 도달할때까지 약 1 시간 15 분이 걸렸습니다. 이것은 선택이 옳았다는것을 보여주나, 목표에 도달하기 위한 온도제어에 대한 전반적인 지연에서 샘플링율이 결과보다 좀 더 길은 듯 합니다.



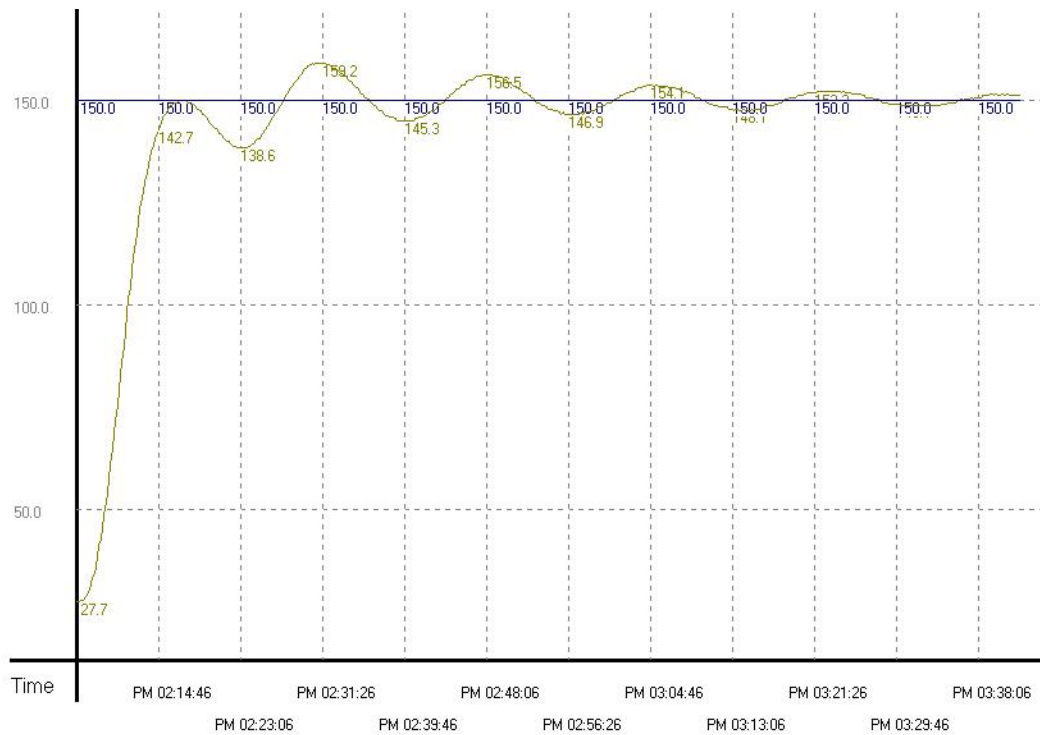
예제 3: 예제 2의 목표온도에 도달하기 위한 시간을 개선하기 위해 샘플링율을 4 초로 수정하였습니다(D12=k40, D30=k4000). 실험결과는 다음과 같습니다:

아래 그림에 의하면, 제어시간은 3 분으로 전반적으로 개선 되었습니다. 그것은 목표온도에 대한 제어시간은 샘플링율을 수정함으로써 나아질 수 있다는것을 증명 합니다.



예제 4: 예제 3의 목표온도에 도달하기 위한 시간을 개선하기 위해 샘플링율을 2 초로 수정하였습니다. (D12=k20, D30=k2000). 실험결과는 다음과 같습니다:

아래의 그림을 참조하면, 너무 짧은 샘플링율은 민감한 제어 시스템에 의해서 극렬한 온도의 변동을 일으킨다는 것을 분명히 보여 줍니다.





API	간략기호			피연산자	기능	제어기						
147	D	SWAP	P	<b>S</b>	상위/하위 바이트 바꿔치기	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	

펄스					16-비트					32-비트				
E S	E X	S S	S A	S X	S C	S H	E S	E X	S S	S A	S X	S C	S H	E S

피연산자:

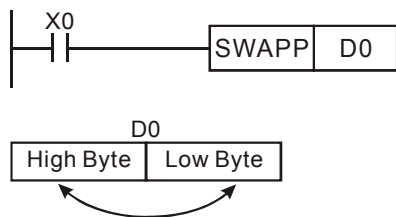
**S**:상위/하위 바이트를 바꿔치기할 장치.

설명:

1. 16 비트 명령이 될 때, 상위/하위 바이트의 내용을 바꿉니다..
2. 32 비트 명령이 될 때, 2 개의 레지스터의 바이트를 개별적으로 바꿉니다.
3. 이 명령은 일반적으로 펄스에 의해서 실행되는 것 입니다. (SWAPP, DSWAPP).

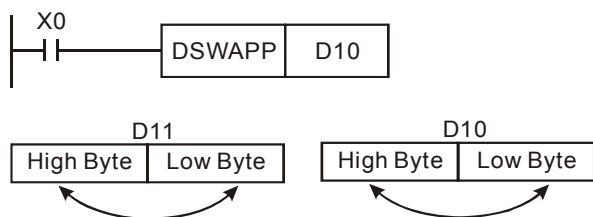
프로그램 예제 1:

X0=ON 일 때, D0 의 상위/하위 바이트의 내용을 서로 바꿉니다..



프로그램 예제 2:

X0=ON 일 때, D11 의 상위 8 비트와 하위 8 비트를 교환하고 D10 의 상위 8 비트와 하위 8 비트를 교환합니다.



API	간략기호			피연산자	기능	제어기						
148	D	MEMR	P	<b>m</b> <b>D</b> <b>n</b>	파일 메모리 읽기	E S	E X	S S	S A	S X	S C	E H

유형 연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MEMR, MEMRP: 7 단계 DMEMR, DMEMRP: 13 단계	
m					*	*							*				
D													*				
n					*	*							*				

펄스							16-비트					32-비트				
E S	E X	S S	S A	S X	S C	E H	E S	E X	S S	S A	S X	S C	E H	E S	E X	S S

**피연산자:**

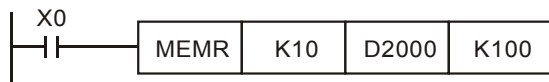
**m**:파일 레지스터의 데이터 읽기를 위한 주소(상수) (m=K0~K1,599 SA 시리즈 MPU 용 ; m=K0~K9,999 EH 시리즈 MPU 용) **D**: 읽은 데이터를 저장하기 위한 주소(상수) (D2000~D4999 SA 시리즈 MPU 용; D2000~D9999 EH 시리즈 MPU 용) **n**: 1회 판독 데이터의 수량(16비트명령: n=K1~ K1,600 SA 시리즈 MPU 용; n=K1 ~ K8,000 EH 시리즈 MPU 용, 32 비트명령: n=K1~ K800 SA 시리즈 MPU 용; n=K1 ~ K4,000 EH 시리즈 MPU 용)

**설명:**

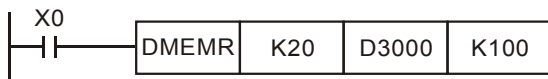
- SA/SX/SC/EH 시리즈 모델들은 파일 레지스터의 데이터를 읽기 위해 이 명령을 사용하고 그 데이터 레지스터에 읽은 데이터를 저장합니다.
- SA/SX/SC 시리즈모델들은 1,600 개의 16 비트 파일 레지스터를 제공하고 EH 시리즈모델들은 10,000 개의 16 비트 파일 레지스터를 제공 합니다.
- SA/SX/SC 시리즈 모델용 피연산자 **m** 및 **n** 은 레지스터 **E** 및 **F** 를 제공하지 않습니다.
- EH 시리즈모델들은 10,000 개의 16 비트 파일 레지스터를 제공 합니다.
- 만일 피연산자 **m** 과 **D** 및 **n** 이 범위를 벗어 나면, 피연산자 오류가 발생합니다. M1067, M1068=On 이 되고 오류코드 0E1A 는 D1067 에 저장 됩니다.

**프로그램 예제 1:**

- 16 비트명령인 MEMR 는 파일 레지스터의 10 번째 주소로 부터 100 개 항목의 데이터를 읽고 읽은 데이터를 D2000 번부터 시작하는 데이터 레지스터에 저장합니다.
- X0=On 일 때, 명령이 실행 됩니다. X0 가 Off 되면, 명령은 실행되지않고 이전에 읽었던 데이터의 내용도 변경되지 않습니다.

**프로그램 예제 2:**

- 32 비트명령인 DMEMR 는 파일 레지스터의 20 번째 주소로 부터 100 개 항목의 데이터를 읽고 읽은 데이터를 D3000 번부터 시작하는 데이터 레지스터에 저장합니다.
- X0=On 일 때, 명령이 실행 됩니다. X0 가 Off 되면, 명령은 실행되지않고 이전에 읽었던 데이터의 내용도 변경되지 않습니다.



API	간략기호			피연산자	기능	제어기						
149	D	MEMW	P	<b>(S)</b> <b>(m)</b> <b>(n)</b>	파일 메모리 쓰기	E S	E X	S S	S A	S X	S C	E H

연산자	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S													*			MEMW, MEMWP: 7 단계 DMEMW, DMEMWP: 13 단계	
m					*	*							*				
n					*	*							*				

펄스							16-비트							32-비트						
E S	E X	S S	S A	S X	S C	S H	E S	E X	S S	S A	S X	S C	S H	E S	E X	S S	S A	S X	S C	S H

**피연산자:**

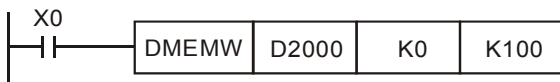
**S:** 데이터 쓰기 위한 주소(상수), ( D2000~D4999 SA 시리즈 MPU 용; D2000~D9999 EH 시리즈 MPU 용) **m:** 파일 레지스터 쓰기용 주소(상수)(m=K0~K1,599 SA 시리즈 MPU 용; m=K0~K9,999 EHMPU 용) **n:** 1 회 판독 데이터의 수량, (16 비트명령: n=K1~ K1,600 SA 시리즈 MPU 용; n=K1 ~ K8,000 EH 시리즈 MPU 용, 32 비트명령: n=K1~ K800 SA 시리즈 MPU 용; n=K1 ~ K4,000 EH 시리즈 MPU 용)

**설명:**

- SA/SX/SC/EH 시리즈 모델들은 파일 레지스터의 데이터를 읽기 위해 이 명령을 사용하고 그 데이터 레지스터에 읽은 데이터를 저장합니다.
- SA/SX/SC 시리즈모델들은 1,600 개의 16 비트 파일 레지스터를 제공하고 EH 시리즈모델들은 10,000 개의 16 비트 파일 레지스터를 제공 합니다.
- SA/SX/SC 시리즈 모델용 피연산자 **m** 및 **n** 은 레지스터 **E** 및 **F** 를 제공하지 않습니다.
- EH 시리즈모델들은 10,000 개의 16 비트 파일 레지스터를 제공 합니다.
- 만일 피연산자 **m** 과 **D** 및 **n** 이 범위를 벗어 나면, 피연산자 오류가 발생합니다. M1067, M1068=On 이 되고 오류코드 0E1A 는 D1067 에 저장 됩니다.

**프로그램 예제:**

- X0=On 일 때, 배정도워드명령 **DMEMW** 가 실행 됩니다. D2001, D2000 으로 시작되는 100 개의 32 비트데이터항목들을 주소 0 번지에서 199 번지까지의 파일 레지스터에 쓰십시오.
- X0=On 일 때, 명령이 실행 됩니다. X0 가 Off 되면, 명령은 실행되지않고 이전에 읽었던 데이터의 내용도 변경되지 않습니다.

**파일 레지스터:**

- EH 시리즈모델:** EH 시리즈 PLC 가 전원이 켜졌을때, EH 시리즈 PLC 는 M1101 (파일 레지스터의 기능을 시작할 것인지), D1101 (번호를 부여하기 위해 파일 레지스터가 시작됨, K0~K9,999), D1102 (읽을 수 있는 레지스터 갯수, K1~K8,000), D1103 (파일 레지스터의 읽은 데이터를 저장하는 목적장치, 번호부여를 시작한 정의된 데이터 레지스터 D, K2,000~K9,999)을 결정하고 자동적으로 파일 레지스터의 내용물을 정의된 데이터 레지스터로 전달시킬 수 있는지를 결정합니다.
- EH 시리즈모델:** D1101 의 값이 0 보다 적거나 혹은 D1103 의 값이 2,000 보다 적거나 9,999 보다 클때, 파일 레지스터로부터 데이터 레지스터로의 데이터 읽기는 중단 됩니다.

3. **SA/SX/SC 시리즈모델:** PLC 가 전원이 켜졌을때, SA/SX/S 시리즈 PLC 는 M1101 (파일 레지스터의 기능을 시작할 것인지), D1101 (번호를 부여하기 위해 파일 레지스터가 시작됨, K0~K1,599), D1102 (읽을 수 있는 레지스터 갯수, K1~ K6,000), D1103 (파일 레지스터의 읽은 데이터를 저장하는 목적장치, 번호부여를 시작한 정의된 데이터 레지스터 D, K2,000~K9,999)을 결정하고 자동적으로 파일 레지스터의 내용물을 정의된 데이터 레지스터로 전달시킬 수 있는지를 결정합니다.
4. **SA/SX/SC 시리즈모델:** D1101 의 값이 0 보다 적거나 또는 1,599 보다 크거나 혹은 D1103 의 값이 2,000 보다 적거나 4,999 보다 클때, 파일 레지스터로부터 데이터 레지스터로의 데이터 읽기는 중단 됩니다.
5. 파일 레지스터가 데이터 레지스터 D 에 데이터를 읽을 때, 파일 레지스터 또는 데이터 레지스터의 주소가 한계를 넘으면, PLC 는 읽기를 중단 합니다.
6. 파일 레지스터의 데이터 읽기 및 쓰기에 관해서는, PLC 프로그램은 읽기에는 API 명령 147 번인 MEMR 을, 쓰기에는 API 명령 148 번인 MEMW 만을 사용 합니다. 파일 레지스터의 상세한 정보에 대해서는 CH2 의 2.8.3 절을 참고 하시기 바랍니다.
7. 32,768 개의 파일 레지스터가 있습니다. 파일 레지스터는 실제 숫자를 가지고 있지 않습니다. 그러므로, 파일 레지스터에서 읽기/쓰기는 API 명령 147 번인 MEMR 과 API 명령 148 번인 MEMW 에 의해 수행되거나 또는 보조장비인 HPP 와 WPLSoft 소프트웨어를 사용하여 수행됩니다.
8. 목적장치(Destination Device)는 항상 연속적이지 않습니다. 한 부분은 내부 SRAM 상에 있고 다른 부분은 SRAM CARD 상에 있습니다. 만일 사용자가 SRAM CARD 를 삽입하지 않고 2,000 번지를 초과하는 주소를 읽었다면, 읽은 값들은 모두 0 이 됩니다.
9. 관련 특수 릴레이 및 파일 레지스터의 레지스터들:

플래그	기능 설명
M1101	파일 레지스터 기능의 시작 유무, 래치됨, 기본값은 Off

특수 D	기능 설명
D1101	D1101 는 파일 레지스터의 시작된 번호임: K0~K1,500 SA 시리즈용, K0~K9,999 EH 시리즈용; 래치됨; 공장출하시 설정은 0.
D1102	D1102 는 읽기를 위한 항목번호를 저장하기위해 사용: K1~K1,600 SA 시리즈용, K1~K8,000 EH 시리즈용; 래치됨; 공장출하시 설정은 0.
D1103	D1103 는 읽기를 위한 주소를 저장하기위해 사용. 시작된 특정 데이터 레지스터의 갯수 D, K2,000~K4,999 SA 시리즈용, K2,000~K9,999 EH 시리즈용; 래치됨; 공장출하시 설정은 0.

API	의사기호	피연산자	기능	제어기							
150	MODRW	<b>(S<sub>1</sub>) (S<sub>2</sub>) (S<sub>3</sub>) (S) (n)</b>	MODBUS 판독/ 기록	ES	EX	SS	SA	SX	SC	EH	

타입 OP	Bit 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MODRW: 11 단계	
S <sub>1</sub>					*	*							*				
S <sub>2</sub>					*	*							*				
S <sub>3</sub>					*	*							*				
S													*				
n					*	*							*				

펄스								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

**피연산자:**

**S<sub>1</sub>:** 외부 장치 주소 (K0~K254) **S<sub>2</sub>:** 기능 부호 ( K3(H3), K6(H6), K16(H10) ) **S<sub>3</sub>:** 외부장치 안에 기록되고 있거나 외부장치로부터 판독된 자료 주소 **S:** 판독 또는 기록된 레지스터  
**n:** 판독/기록 데이터의 길이.

**설명:**

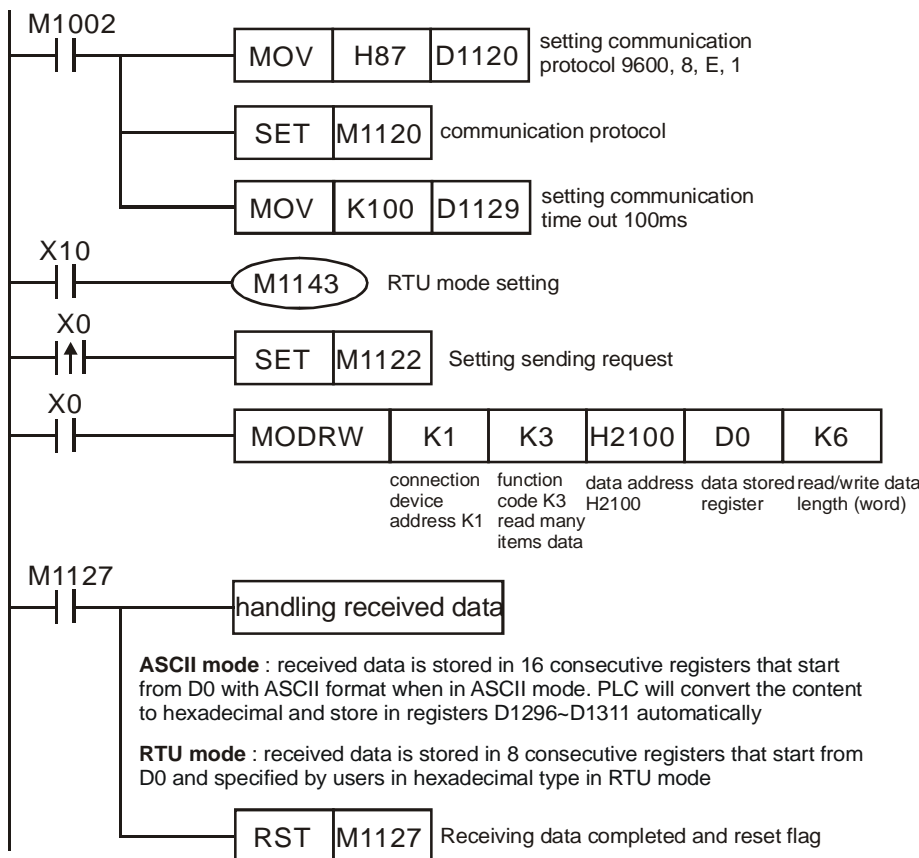
- ES 시리즈 모델은 색인 레지스터 E 와 F 를 지원하지 않습니다.
- S<sub>1</sub>:** 연결 장치 주소 **S<sub>2</sub>:** 기능코드 **S<sub>3</sub>:** 판독 또는 기록된 주소 **S:** 판독 또는 기록된 레지스터  
**n:** 판독/기록 데이터의 길이.
- S<sub>1</sub>:** 연결 장치 주소 (장치 주소). K0 에서 K255 까지의 설정 범위.
- S<sub>2</sub>:** 기능 부호. 예: 많은 항목을 판독하기위한 AC 드라이브 또는 DVP-PLC 의 명령은 H03 입니다. AC 드라이브 또는 DVP-PLC 의 기록 명령은 H06, 많은 항목 기록의 명령은 H10 입니다. 오직 위의 세 기능 부호만 제공되며, 다른 기능 부호는 금지됩니다. 다음에 나오는 프로그램 예들을 참조하십시오.
- S<sub>3</sub>:** 판독/ 기록 데이터의 장치 주소 (장치 주소). 이것은 연결 장치의 내부 장치 주소입니다. 만약 주소가 지정된 장치에 적합하면 D1130에 저장된 장애코드가 나타나는 동시에 M1141이 켜질 것입니다. 예를 들면, 4000H는 VFD-S 에 오류이고, M1141 는 ON 그리고 D1130 = 2 일 것입니다. 자세한 장애코드는 VFD-S 사용설명서를 참고 하십시오.
- S:** 판독/기록되는 발생원 또는 수신지(발생원 또는 수신지). 사용자는 판독후 저장된 자료 또는 판독전에 자료 길이 기록을 위한 레지스터를 설정할 수 있습니다.
- n:** 판독/기록 데이터 길이(자료 길이). ES/SA 시리즈에서 M1143=OFF (ASCII 모드)일때, 지정된 범위는 K1~K8 (Word)입니다. M1143=ON (RTU 모드)일때, 지정된 범위는 K1~K16 (Word)입니다. EH 시리즈에서, 지정된 범위는 K1~K16 (Word).

**프로그램 예 1:**

- 기능 부호 K3(H3) : 많은 항목 데이터 판독  
 PLC 가 VFD-S AC 드라이브에 연결됨. ( M1143=OFF 일때 ASCII 모드)  
 PLC 가 VFD-S AC 드라이브에 연결됨. ( M1143=ON 일때 RTU 모드)
- 수신된 데이터는 ASCII 모드에 있을 때 ASCII 형식으로 D0 로부터 시작된 16 연속 레지스터에 저장된 데이터 입니다. PLC 은 그 내용을 16 진으로 변환하고 D1296~D1311 에 자동으로 저장할 것입니다. 그것이 16 진으로 변환하기 시작하면 M1131=ON 이고 M1131 은 변환이 끝난 후에 OFF 일 것입니다.
- 사용자는 사용하기위한 범용 레지스터에 16 진 데이터로 저장한 D1296~D1311 를 옮기기 위한 MOV,

DMOV 또는 BMOV 명령어를 사용할 수 있습니다. ES 시리즈에 적합한 다른 명령은 이 영역에 적절하지 않다.

4. 수신된 데이터는 RTU 모드안에 16 진 포맷에서 사용자의 의해 지정되고 D0로부터 시작된 8 진 연속레지스터에 저장됩니다. 동시에 D1296~D1311 는 적합하지않다.
5. ASCII 모드 또는 RTU 모드에서, PLC 는 D1256~D1295 안에 전송 데이터로 저장될 것입니다. 사용자들은 이들 레지스터를 MOV, DMOV 또는 BMOV 명령을 사용하여 범용 레지스터로 옮길 수 있습니다. 다른 명령은 이 영역에 적절하지 않다.
6. AC 드라이브로부터 수신된 데이터는 사용자의 의해 지정된 레지스터에 저장됩니다. 데이터 수신 완료 후 PLC 는 자동으로 수신된 데이터가 올바른지 확인할 것입니다. 만약 어떤 장애가 있습니다.면, M1140 는 On 에 설정 될 것입니다.
7. 만약 AC 드라이브의 내부 데이터 주소가 지정된 드라이브에 오류라면, 장애 부호를 갖게 될 것입니다. 장애 부호는 D1130 에 저장되고 M114 는 켜질 것입니다. 예를들면, 8000H 는 VFD-S 의 오류이고 1141=ON 와 D1130=2 이 됩니다. 장애코드는 VFD-S 사용 설명서를 참고 하십시오.
8. M1140=ON 또는 M1141=ON 이면, AC 드라이브에 올바른 데이터로 전송될 것입니다. 만약 수신된 데이터가 올바르다면, M1140 와 M1141 는 리셋 될 것입니다.



9. ASCII 모드: PLC 는 VFD-S AC 드라이브에 연결.

PLC ⇨ VFD-S, PLC 전송: **"01 03 2100 0006 D5"**

VFD-S ⇨ PLC, PLC 수신: **"01 03 0C 0100 1766 0000 0000 0136 0000 3B"**

PLC 은 데이터 레지스터를 전송합니다. (전송 메시지)

레지스터	데이터		설명	
D1256 Low	‘0’	30 H	ADR 1	ADR (1,0) 는 AC 드라이브 주소
D1256 High	‘1’	31 H	ADR 0	
D1257 Low	‘0’	30 H	CMD 1	CMD (1,0) 는 명령 부호
D1257 High	‘3’	33 H	CMD 0	
D1258 Low	‘2’	32 H	데이터 주소 시작	
D1258 High	‘1’	31 H		
D1259 Low	‘0’	30 H		
D1259 High	‘0’	30 H		
D1260 Low	‘0’	30 H	데이터의 번호 (word 에 의해 계산)	
D1260 High	‘0’	30 H		
D1261 Low	‘0’	30 H		
D1261 High	‘6’	36 H		
D1262 Low	‘D’	44 H	LRC CHK 1	LRC CHK (0,1)는 오류 검사
D1262 High	‘5’	35 H	LRC CHK 0	

PLC 는 데이터 레지스터 D0 를 수신합니다. (응답 메시지)

레지스터	데이터		설명	
D0 Low	‘0’	30 H	ADR 1	
D0 High	‘1’	31 H	ADR 0	
D1 Low	‘0’	30 H	CMD 1	
D1 High	‘3’	33 H	CMD 0	
D2 Low	‘0’	30 H	데이터의 번호 (Byte 에 의해 계산)	
D2 High	‘C’	43 H		
D3 Low	‘0’	30 H	2100H 의 데이터 내용	PLC 은 자동적으로 ASCII 코드를 숫자로 전송하고 D1296=H0100 에 저장합니다.
D3 High	‘1’	31 H		
D4 Low	‘0’	30 H		
D4 High	‘0’	30 H		
D5 Low	‘1’	31 H	2101H 의 데이터 내용	PLC 은 자동적으로 ASCII 코드를 숫자로 전송하고 D1297=H1766 에 저장합니다.
D5 High	‘7’	37 H		
D6 Low	‘6’	36 H		
D6 High	‘6’	36 H		
D7 Low	‘0’	30 H	2102H 의 데이터 내용	PLC 은 자동적으로 ASCII 코드를 숫자로 전송하고 D1298=H0000 에 저장합니다.
D7 High	‘0’	30 H		
D8 Low	‘0’	30 H		
D8 High	‘0’	30 H		
D9 Low	‘0’	30 H	2103H 의 데이터 내용	PLC 은 자동적으로 ASCII 코드를 숫자로 전송하고 D1299=H0000 에 저장합니다.
D9 High	‘0’	30 H		
D10 Low	‘0’	30 H		
D10 High	‘0’	30 H		
D11 Low	‘0’	30 H	2104H 의 데이터 내용	PLC 은 자동적으로 ASCII 코드를 숫자로 전송하고 D1300=H0136 에 저장합니다.
D11 High	‘1’	31 H		
D12 Low	‘3’	33 H		
D12 High	‘6’	36 H		
D13 Low	‘0’	30 H	2105H 의 데이터 내용	PLC 은 자동적으로 ASCII 코드를 숫자로 전송하고 D1301=H0000 에 저장합니다.
D13 High	‘0’	30 H		
D14 Low	‘0’	30 H		
D14 High	‘0’	30 H		
D15 Low	‘3’	33 H	LRC CHK 1	
D15 High	‘B’	42 H	LRC CHK 0	

10. RTU 모드: PLC 는 VFD-S AC 드라이브에 연결합니다.

PLC ⇨ VFD-S, PLC 전송: **01 03 2100 0006 CF F4**

VFD-S ⇨ PLC, PLC 수신: **01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5**

PLC 는 데이터 레지스터를 전송합니다. (전송 메시지)

레지스터	데이터	설명
D1256 Low	01 H	주소
D1257 Low	03 H	기능
D1258 Low	21 H	데이터 주소 시작
D1259 Low	00 H	
D1260 Low	00 H	데이터의 번호 (word 에 의한 계산)
D1261 Low	06 H	
D1262 Low	CF H	CRC CHK Low
D1263 Low	F4 H	CRC CHK High

PLC 는 데이터 레지스터를 수신합니다. (응답 메시지)

레지스터	데이터	설명	
D0 Low	01 H	주소	
D1 Low	03 H	기능	
D2 Low	0C H	데이터의 번호 (Byte)	
D3 Low	00 H	2100 H 의 데이터 내용	PLC 는 자동적으로 데이터 번호를 D1296=H0000 에 저장합니다.
D4 Low	00 H		
D5 Low	05 H	2101 H 의 데이터 내용	PLC 는 자동적으로 데이터 번호를 D1297=H0503 에 저장합니다.
D6 Low	03 H		
D7 Low	0B H	2102 H 의 데이터 내용	PLC 는 자동적으로 데이터 번호를 D1298=H0BB8 에 저장합니다.
D8 Low	B8 H		
D9 Low	0B H	2103 H 의 데이터 내용	PLC 는 자동적으로 데이터 번호를 D1299=H0BB8 에 저장합니다.
D10 Low	B8 H		
D11 Low	00 H	2104 H 의 데이터 내용	PLC 는 자동적으로 데이터 번호를 D1300=H0000 에 저장합니다.
D12 Low	00 H		
D13 Low	01 H	2105 H 의 데이터 내용	PLC 는 자동적으로 데이터 번호를 D1301=H012D 에 저장합니다.
D14 Low	2D H		
D15 Low	8E H	CRC CHK Low	
D16 Low	C5 H	CRC CHK High	

## 프로그램 예 2:

1. 기능 부호 K6(H6) : 한 단어 데이터를 레지스터 안에 기록

PLC 는 VFD-S AC 드라이브에 연결합니다. (M1143=OF 일때 ASCII Mode )

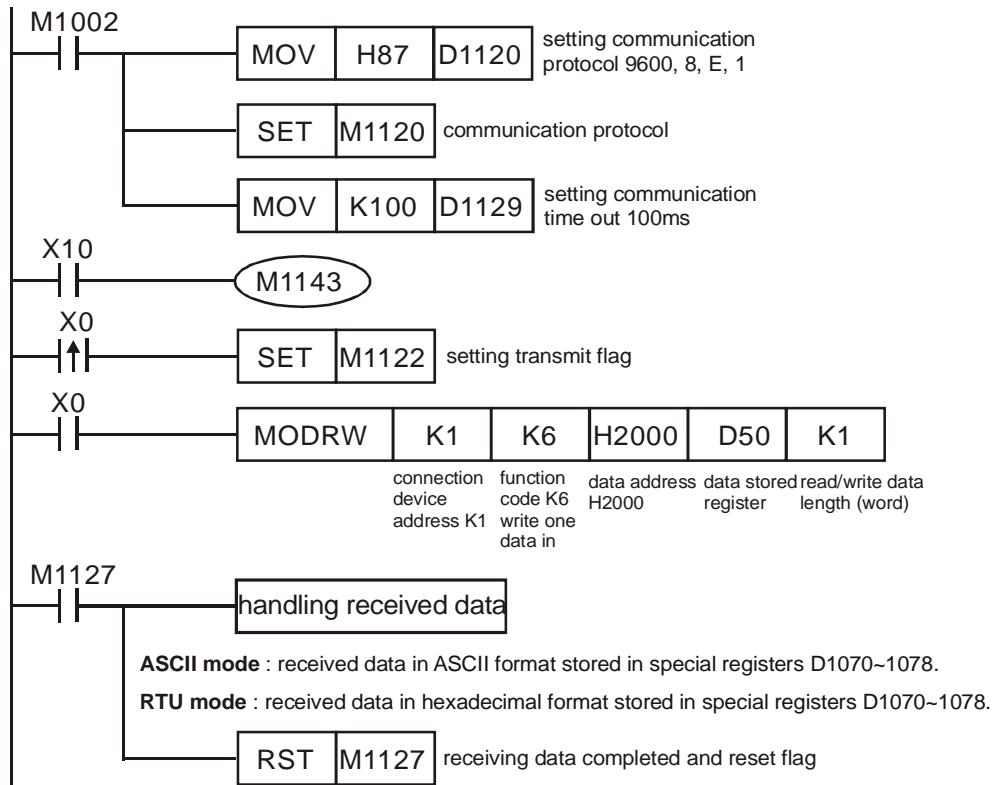
PLC 는 VFD-S AC 드라이브에 연결합니다. (M1143=ON 일때 RTU Mode)

2. ASCII 에 있을 때, 사용자는 지정된 레지스터 D50 에서 ASCII 형식의 AC 드라이브에 기록될 데이터를 저장합니다. AC 드라이브로부터 수신된 데이터는 D1070~D1076 레지스터 안에 저장될 것입니다.
3. RTU 모드에 있을 때, 사용자는 지정된 레지스터 D50 에 16 진 형식의 AC 드라이브에 기록될 데이터를 저장합니다. AC 로부터 수신된 데이터는 D1070~D1077 레지스터에 저장될 것입니다.
4. ASCII 모드 또는 RTU 모드안에 있을 때, PLC 은 전송 데이터를 레지스터 D1256~D1295 에 저장할



것입니다. 사용자는 이들 데이터를 MOV, DMOV 또는 BMOV 명령을 이용하여 범용 레지스터에 저장할 수 있습니다. ES 시리즈에서 다른 명령은 이 영역에 적절하지 않다.

5. 데이터 수신 완료 후, PLC 는 자동으로 수신된 데이터가 올바른지를 확인할 것입니다. 만약 어떤 장애가 있습니다.면 M1140 는 ON 에 설정될 것입니다.
6. AC 드라이브의 내부 데이터 주소가 지정된 장치에 오류일 때, 장애 부호를 갖는다. 장애 부호는 D1130 에 저장되고 M1141 는 켜질 것입니다. 예를들면, 8000H 는 VFD-S 에 오류이고, M1141=ON 이 되고 D1130=2 이 됩니다. 장애 부호는 VFD-S 사용 설명서를 참고 하십시오.
7. M1140=ON 또는 M1141=ON 이 된 후, AC 드라이브에 올바른 데이터를 전송할 것입니다. 수신된 데이터가 올바를 때, M1140 와 M1141 는 리셋 될 것입니다.



8. ASCII 모드: PLC 는 VFD-S AC 드라이브에 연결합니다.

PLC ⇒ VFD-S, PLC 전송: “ 01 06 0100 1770 71 ”

VFD-S ⇒ PLC, PLC 수신: “ 01 06 0100 1770 71 ”

PLC 는 데이터 레지스터를 전송합니다. (전송 메시지)

레지스터	데이터		설명	
D1256 Low	‘0’	30 H	ADR 1	ADR (1,0) 는 AC 드라이브 주소입니다.
D1256 High	‘1’	31 H	ADR 0	
D1257 Low	‘0’	30 H	CMD 1	CMD (1,0) 는 명령 부호입니다.
D1257 High	‘6’	36 H	CMD 0	
D1258 Low	‘0’	30 H	데이터 주소	
D1258 High	‘1’	31 H		
D1259 Low	‘0’	30 H		
D1259 High	‘0’	30 H		
D1260 Low	‘1’	31 H	데이터 내용	D50 의 내용 (H1770=K6000)
D1260 High	‘7’	37 H		
D1261 Low	‘7’	37 H		

D1261 High	'0'	30 H		
D1262 Low	'7'	37 H	LRC CHK 1	LRC CHK (0,1)는 오류 점사입니다.
D1262 High	'1'	31 H	LRC CHK 0	

PLC 는 데이터 레지스터를 수신합니다.(응답 메시지)

레지스터	데이터		설명
D1070 Low	'0'	30 H	ADR 1
D1070 High	'1'	31 H	ADR 0
D1071 Low	'0'	30 H	CMD 1
D1071 High	'6'	36 H	CMD 0
D1072 Low	'0'	30 H	데이터 주소
D1072 High	'1'	31 H	
D1073 Low	'0'	30 H	
D1073 High	'0'	30 H	
D1074 Low	'1'	31 H	데이터 내용
D1074 High	'7'	37 H	
D1075 Low	'7'	37 H	
D1075 High	'0'	30 H	
D1076 Low	'7'	37 H	LRC CHK 1
D1076 High	'1'	31 H	LRC CHK 0

9. RTU Mode: PLC 는 VFD-S AC 드라이브에 연결합니다.

PLC ⇨ VFD-S, PLC 전송 **01 06 2000 0012 02 07**

VFD-S ⇨ PLC, PLC 수신: **01 06 2000 0012 02 07**

PLC 은 데이터 레지스터를 전송합니다.(전송 메시지)

레지스터	데이터	설명
D1256 Low	01 H	주소
D1257 Low	06 H	기능
D1258 Low	20 H	데이터 주소
D1259 Low	00 H	
D1260 Low	00 H	데이터 내용
D1261 Low	12 H	
D1262 Low	02 H	CRC CHK Low
D1263 Low	07 H	CRC CHK High

PLC 은 데이터 레지스터를 수신합니다.(응답 메시지)

레지스터	데이터	설명
D1070 Low	01 H	주소
D1071 Low	06 H	기능
D1072 Low	20 H	데이터 주소
D1073 Low	00 H	
D1074 Low	00 H	데이터 내용
D1075 Low	12 H	
D1076 Low	02 H	CRC CHK Low
D1077 Low	07 H	CRC CHK High

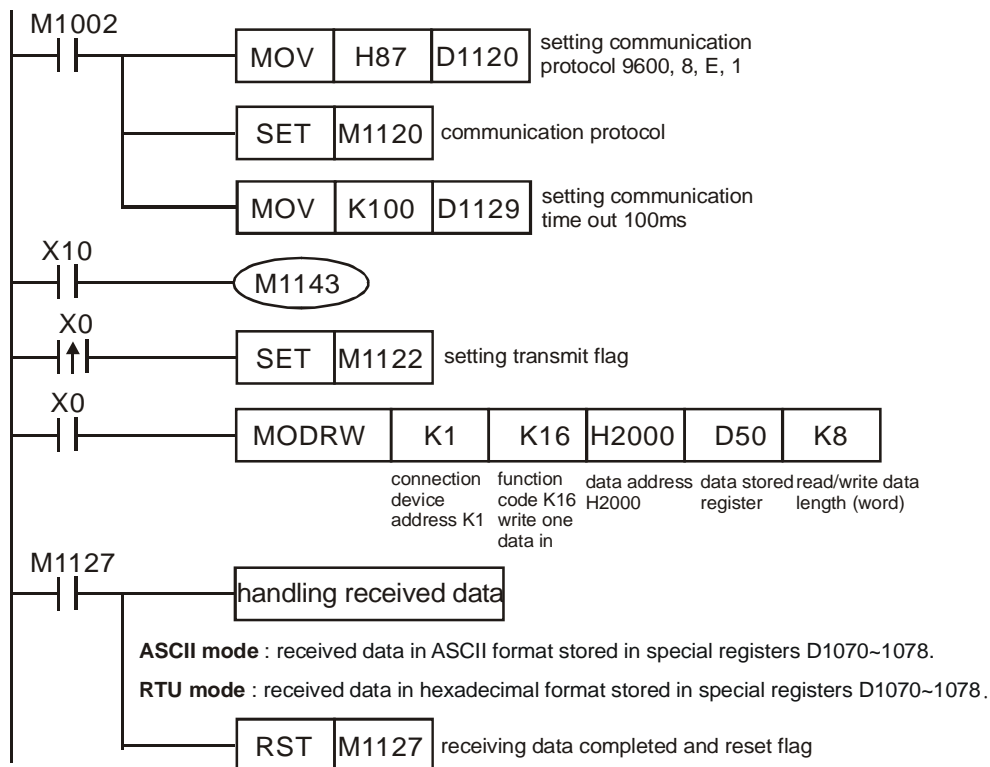
프로그램 예 3:

1. 기능 부호 K16(H10) : 많은 항목 워드 데이터를 레지스터안에 기록

PLC 는 AC 드라이브에 연결합니다. ( M1143=OF 일 때 ASCII Mode )

PLC 는 VFD-S AC 드라이브에 연결합니다. ( M1143=ON 일 때 RTU Mode )

2. ASCII 모드안에 있을 때, 사용자는 D50 으로부터 시작된 8 진 연속 지정된 레지스터에 ASCII 형식의 AC 드라이브에 기록될 데이터를 저장합니다. AC 드라이브로부터 수신된 데이터는 레지스터 D1070~D1078 에 저장될 것입니다.
3. RTU 모드안에 있을 때, 사용자는 사용자는 D50 으로부터 시작된 8 진 연속 지정된 레지스터에 ASCII 형식의 AC 드라이브에 기록될 데이터를 저장합니다. AC 드라이브로부터 수신된 데이터는 레지스터 D1070~D1078 에 저장될 것입니다.
4. ASCII 모드 또는 RTU 모드안에 있을 때, PLC 는 D1256~D1295 에 전송 데이터를 저장할 것입니다. 사용자는 MOV, DMOV or BMOV 명령을 사용하여 범용레지스터에 이들 데이터를 옮길 수 있습니다. ES 시리즈에서 다른 명령은 이 영역에 적합하지 않다.
5. 데이터 수신 완료 후, PLC 은 자동적으로 수신된 데이터가 올바른지 확인할 것입니다. 어떤 장애가 있을 때, M1140 는 ON 으로 설정될 것입니다.
6. 만약 AC 드라이브의 내부 데이터 주소가 지정된 드라이브의 오류일 때, 장애코드를 갖게 될 것입니다. 장애코드는 D1130 에 저장되고 M1141 는 켜질 것입니다. 예를들면, 8000H 는 VFD-S 에 오류가 되고 M1141=ON 되고, D1130=2 가 됩니다. VFD-S 사용설명서의 장애코드를 참고하십시오.
7. M1140=ON 또는 M1141=ON 이 된 후에, AC 드라이브의 올바른 데이터를 전송할 것입니다. 만약 전송된 데이터가 올바르면, M1140 와 M1141 는 리셋될 것입니다.



8. ASCII Mode: PLC 는 VFD-S AC 드라이브에 연결합니다.

PLC ⇨ VFD-S, PLC 전송: “ 01 10 2000 0002 04 0012 1770 30 ”

VFD-S ⇨ PLC, PLC 수신: “ 01 10 2000 0002 CD ”

PLC 은 데이터 레지스터를 전송합니다.(전송 메시지)

레지스터	데이터		설명	
D1256 Low	'0'	30 H	ADR 1	ADR (1,0) 는 AC 드라이브 주소입니다.
D1256 High	'1'	31 H	ADR 0	
D1257 Low	'1'	31 H	CMD 1	CMD (1,0) 는 명령 부호입니다.
D1257 High	'0'	30 H	CMD 0	
D1258 Low	'2'	32 H	데이터 주소	
D1258 High	'0'	30 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'0'	30 H	레지스터 번호	
D1260 High	'0'	30 H		
D1261 Low	'0'	30 H		
D1261 High	'2'	32 H		
D1262 Low	'0'	30 H	Byte Count	
D1262 High	'4'	34 H		
D1263 Low	'0'	30 H	데이터 내용 s 1	레지스터 D50 의 내용 (H12)
D1263 High	'0'	30 H		
D1264 Low	'1'	31 H		
D1264 High	'2'	32 H		
D1265 Low	'1'	31 H	데이터 내용 s 2	레지스터 D51 의 내용(H1770=K6000)
D1265 High	'7'	37 H		
D1266 Low	'7'	37 H		
D1266 High	'0'	30 H		
D1267 Low	'3'	33 H	LRC CHK 1	LRC CHK (0,1) 는 오류 검사 입니다.
D1267 High	'0'	30 H	LRC CHK 0	

PLC 은 데이터 레지스터를 수신합니다.(응답 메시지)

레지스터	데이터		설명
D1070 Low	‘0’	30 H	ADR 1
D1070 High	‘1’	31 H	ADR 0
D1071 Low	‘1’	31 H	CMD 1
D1071 High	‘0’	30 H	CMD 0
D1072 Low	‘2’	32 H	데이터 주소
D1072 High	‘0’	30 H	
D1073 Low	‘0’	30 H	
D1073 High	‘0’	30 H	
D1074 Low	‘0’	30 H	레지스터 번호
D1074 High	‘0’	30 H	
D1075 Low	‘0’	30 H	
D1075 High	‘2’	32 H	
D1076 Low	‘C’	43 H	LRC CHK 1
D1076 High	‘D’	44 H	LRC CHK 0

9. RTU Mode: PLC 는 VFD-S AC 드라이브에 연결합니다.

PLC ⇨ VFD-S, PLC 전송: **01 10 2000 0002 04 0012 1770 C4 7F**

VFD-S ⇨ PLC, PLC 수신: **01 10 2000 0002 4A 08**

PLC 은 데이터 레지스터를 전송합니다.(전송 메시지)

레지스터	데이터	설명
D1256 Low	01 H	주소
D1257 Low	10 H	기능
D1258 Low	20 H	데이터 주소
D1259 Low	00 H	
D1260 Low	00 H	레지스터 번호
D1261 Low	02 H	

D1262 Low	04 H	Byte 계산	
D1263 Low	00 H	데이터 내용 1	레지스터 D50 의 내용 (H12)
D1264 Low	12 H		
D1265 Low	17 H	데이터 내용 2	레지스터 D51 의 내용 (H1770=K6000)
D1266 Low	70 H		
D1262 Low	C4 H	CRC CHK Low	
D1263 Low	7F H	CRC CHK High	

PLC 은 데이터 레지스터를 수신합니다.(응답 메시지)

레지스터	데이터	설명
D1070 Low	01 H	주소
D1071 Low	10 H	기능
D1072 Low	20 H	데이터 주소
D1073 Low	00 H	
D1074 Low	00 H	레지스터 번호
D1075 Low	02 H	
D1076 Low	4A H	CRC CHK Low
D1077 Low	08 H	CRC CHK High

주의:

- (연결) MODRD, RDST, MODRW 이들 세가지 명령전의 실행 조건은 상승 에지 연결과 하강 에지를 이용할 수 없습니다. 반면, 수신된 레지스터에 저장된 데이터는 부정확하게 될 것입니다.
- RS-485 통신의 연관된 플래그와 특수 레지스터 MODRW 명령: API 80 RS 명령의 더 자세한 정보는 각주를 참고하십시오.

플래그/특수 레지스터	기능 기술
M1120	래치 설정된 통신. 설정후의 D1120 의 변화는 적절하지 않을 것 입니다.
M1121	Off 상태가 되면 PLC 의 RS-485 는 통신 데이터를 보낼 것입니다.
M1122	배달 요청
M1123	수신 완료
M1124	대기 메시지 수신
M1125	수신 상태 기능 억제
M1126	STX/ETX 시스템 정의 선택
M1127	MODRD / RDST / MODRW 명령 데이터 수신 완료
M1128	전송/수신표시
M1129	수신 시간 종료
M1130	사용자/시스템 정의 STX/ETX
M1131	MODRD / MODWR / MODRW 데이터 HEX, M1131=ON 로 변환
M1140	MODRD / MODWR / MODRW 데이터의 에러 수신
M1141	MODRD / MODWR / MODRW 명령 인자 에러
M1142	VFD-A 편의 명령 데이터의 에러 수신
M1143	ASCII/RTU 모드 선택 (명령어 MODRD/MODWR/MODRW 으로 사용) (Off 는 ASCII 모드, ON 은 RTU 모드입니다.)
D1070~D1085	이것은 RS-485 통신 편의 명령어에 내장된 PLC 입니다. 이 명령어는 수행 중에 메시지를 보낼 것입니다. 그리고 만약 수신자가 수신하면, 메시지를 돌려보내고, D1070~D1085 에 저장할 것입니다. 사용자는 이 레지스터 내용에 의해 돌려보내진 것을 볼 수 있습니다.
D1120	RS-485 통신 프로토콜
D1121	PLC 통신 주소.(기능이 래치된 PLC 통신 주소 저장)
D1122	배달 데이터의 잔류 문자
D1123	수신 데이터의 잔류 문자
D1124	텍스트 정의 시작 (STX)

D1125	첫번째 끝 문자의 정의 (ETX1)
D1126	두 번째 끝 문자의 정의 (ETX2)
D1129	통신 시간 종료 이상. 시간 단위: (ms)
D1130	MODBUS 의 장애코드 복귀
D1256~D1295	이것은 RS-485 통신 편의 명령어에 내장된 PLC 입니다. 수행 중에 이 명령을 보낸 메시지는 D1256~D1295 에 저장될 것입니다. 사용자는 이 레지스터 내용에 따라 확인할 수 있습니다.
D1296~D1311	PLC 은 자동적으로 16 진 형식으로 사용자에게 의해 지정된 레지스터안에 저장된 ASCII 데이터로 저장할 것입니다.

API	의사기호	피연산자	기능	제어기
151	PWD	<b>S</b> <b>D</b>	입력 펄스 폭 검출	ES EX SS SA SX SC EH

타입 OP	Bit 장치				워드 장치										프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MTR: 9 단계		
S	*																	
D													*					

펄스								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH		

피연산자:

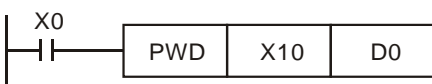
**S:** 원시 장치    **D:** 탐지 결과를 저장하는 수신지 장치

설명:

- 이 명령은 X10~X17 입력의 On 펄스 폭을 검출하는것에 이용되고, 시간단위는 10us 입니다. 만약 On 펄스 폭이 10us 보다 작다면, 지정된 D 의 값은 0 과 같다. PWD 의 입력값은 X10~X17 DCNT 과 ZRN 에 의해 표시된 입력값 X10~X17 과 같을 수 없습니다.
- D 는 두개의 연속 장치를 사용합니다. 최장 탐지 시간은 21,474.83647 초 이고, 약 3,579.139 분, 약 5.9652 시간입니다.

프로그램 예 :

X0=On 일 때, 입력 X10 의 On 펄스 폭을 기록, D1, D0 에 저장합니다.



API	의사기호	피연산자	기능	제어기						
152	RTMU	<b>D</b> <b>n</b>	중단의 실행 시간 측정 시작	ES	EX	SS	SA	SX	SC	EH

Type	비트 장치				워드 장치										프로그램 단계					
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MTR: 9 단계				
D					*	*							*							
n					*	*							*							

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**피연산자:**

**D:** 시간 측정을 저장하는 수신지 장치 (시간 단위는 1us)    **n:** 시간 기준 측정, 변수 범위: K10~K500  
(시간단위: 1us)

**설명:**

- D의 제한 범위는 지정된 특별 D 레지스터인 K0~K9 이고, 지정된 특별 D 레지스터의 수는 D1156~1165 이고 이중 최대 10 개의 인터럽트 서브루틴을 측정할 수 있습니다. 예를들면 D의 값이 K5 일때, 그것은 지정된 특별 D 레지스터의 수가 D1161 임을 의미합니다.
- RTMU 명령 실행 후, 만약 D의 범위, 사용자에게 의해 입력된 n이 적절합니다.면, 이 명령은 I 인터럽트의 실행 시간을 측정과 동시에 D에서 0 까지 지정된 특별 D 레지스터의 내용의 재설정을 시작한 타이머를 얻게 될 것입니다. RTMD 명령에 이르면, 타이머는 중지되고 I 인터럽트의 측정 실행 시간이 끝나게 될 것입니다. 동시에 RTMD 명령에 지정된 특별 D 레지스터에 측정 실행 시간을 지정합니다.
- 다음에 소개된 RTMD 명령으로 사용된 이 RTMU 명령과 이들 두 명령은 모두 사용자가 프로그램 전개의 시작에서 ISR (Interrupt Service Routine) 의 실행 시간공급을 제한하고 빠른 속도 반응을 다루는 I 인터럽트 서비스 프로그램의 수행시간을 측정하는데 사용됩니다.

API	의사기호	피연산자	기능	제어기						
153	RTMD	<b>D</b>	인터럽트의 실행 시간 측정의 종료	ES	EX	SS	SA	SX	SC	EH

Type	비트 장치				워드 장치										프로그램 단계					
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MTR: 9 steps				
D					*	*							*							

펄스							16-bit							32-bit							
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	

**피연산자:**

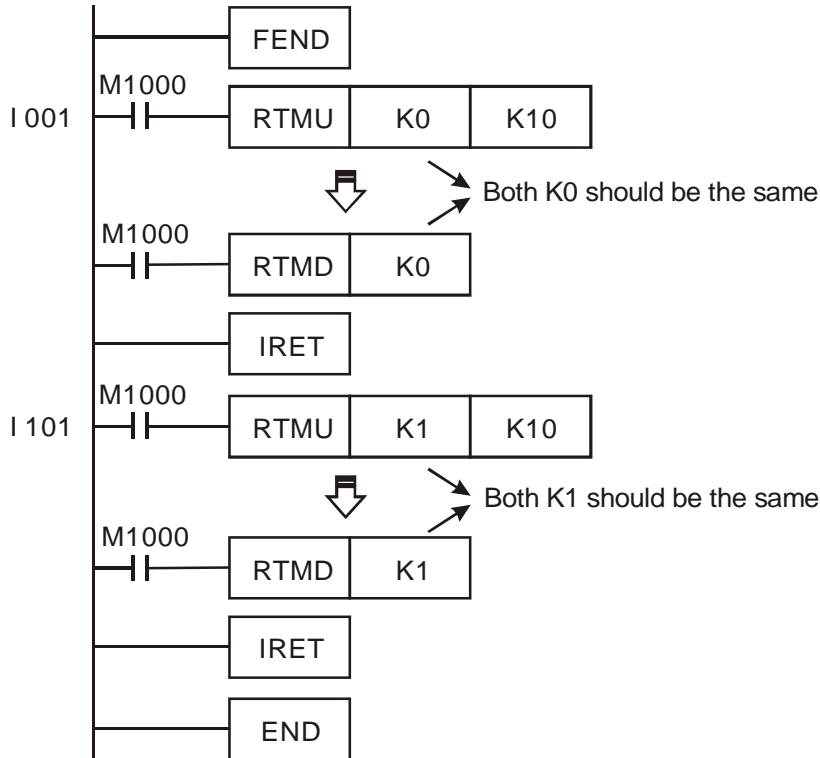
**D:** 시간 측정을 저장하는 수신지 장치 (시간 단위는 1us)

**설명:**

D의 제한 범위는 지정된 특별 D 레지스터인 K0~K9 이고, 지정된 특별 D 레지스터의 수는 D1156~1165 이고 이중 최대 10 개의 인터럽트 서브루틴을 측정할 수 있습니다. 예를들면 D의 값이 K5 일때, 그것은 지정된 특별 D 레지스터의 수가 D1161 임을 의미합니다. 피연산자 D의 수는 API 152 RTMU와 협력하는 피연산자 D와 같음에 틀림없거나 또는, 예상되지않은 결과가 측정에 나타날 것입니다.

프로그램 예 :

1001 인터럽트 서브루틴안에 들어가는 X0 가 Off→On 일 때, RTMU 명령은 8-bit 타이머를 시작할 것입니다. (단위시간은 10us 입니다.). RTMD 명령이 K0 에 이를 때, 타이머는 중지하고 특별 D 레지스터에 측정시간을 저장합니다. (총 10 개의 레지스터 D1156~D1165 가 있고 K0~K9 로 지정되어있습니다.)



주의:

1. 개발 PLC 프로그램 완료 후, 사용자가 반드시 이 명령을 제거 할 것을 권합니다.
2. 추가 설명:
  - a) 덜 중요한 RTMU 명령에 의해 실행된 인터럽트 시간 때문에 (다른 인터럽트보다 덜 중요합니다.), 타이머는 작동 되지 않을 것이고, RTMU 명령의 수행 기간동안 높은 속도의 펄스 입력을 지정하거나 또는 계산하는 빠른 속도 펄스 입력을 수행할 때 계산하지 않습니다.
  - b) 만약 사용자가 RTMU 명령을 실행하면, 그러나 RTMD 명령을 프로그램 인터럽트의 종료 전에 실행하지 않으면, 인터럽트는 종료되지 않을 것입니다.
  - c) 특히 RTMU 명령이 PLC 하나의 내부 타이머 인터럽트를 실행하면서 수행하는 것을 인지하십시오. 만약 그 결과로 다양한 RTMU 또는 RTMD 명령어가 일제히 실행되면 타이머는 난잡해 질 것입니다. .
3. D1156~D1165: RTMU, RTMD 명령에 의해 지정된 특별 1D 레지스터 (번호는 K0 부터 K9 까지).



API	의사기호			피연산자	기능	제어기						
154		RAND	P	<b>S<sub>1</sub></b> <b>S<sub>2</sub></b> <b>D</b>	Random Number	E S	E X	S	S A	S X	S C	E H

Type OP	비트 장치				워드 장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	RAND, RANDP: 7 steps
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*	
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**피연산자:**

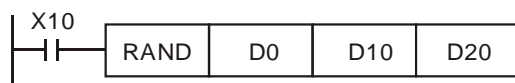
S<sub>1</sub>: 임의의 숫자를 생성하기 위한 낮은 제한 S<sub>2</sub>: 임의의 숫자를 생성하기 위한 높은 제한 D: 임의의 숫자 결과

**설명:**

1. 사용자가 S<sub>1</sub> > S<sub>2</sub> 을 입력하면, PLC 은 피연산자 에러가 발생할 것이고, 그것을 수행하지 않을것입니다.  
그리고 나서 M1067, M1068=On 이 되고, 오류부호 0E1A 을 D1067 에기록합니다.(HEX) .
2. 피연산자 S<sub>1</sub> 와 S<sub>2</sub> 가능한 범위: K0 ≤ S<sub>1</sub> , S<sub>2</sub> ≤ K32,767.

**프로그램 예 :**

X10=On 일때, 낮은 경계 D0 와 높은 경계 D10 안의 임의의 숫자는 D20 에 저장될 것입니다.



API	의사기호		피연산자	기능	제어기						
155	D	ABSR	<b>(S)</b> <b>(D<sub>1</sub>)</b> <b>(D<sub>2</sub>)</b>	절대 위치 판독	ES	EX	SS	SA	SX	SC	EH

Type	비트 장치				워드 장치											프로그램 단계	
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DABSR: 13 steps	
S	*	*	*	*													
D <sub>1</sub>		*	*	*													
D <sub>2</sub>								*	*	*	*	*	*	*			

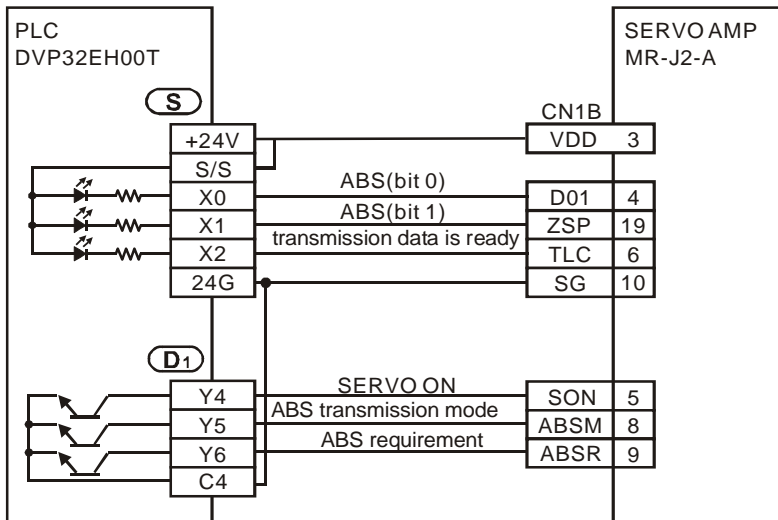
펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

#### 피연산자:

**S:** Servo 로 부터 기호 입력 (3 개의 연속장치 사용)    **D<sub>1</sub>:** **D<sub>1</sub>:** Servo 제어를 위한 제어 신호 (3 개의 연속장치 사용)    **D<sub>2</sub>:** Servo 로부터 절대 위치 데이터(32bit) 판독 (2 개의 연속장치 사용)

#### 설명:

- 이 명령은 mitsubishi 서보 드라이브 MR-J2 의 연속 절대 위치 데이터 판독 기능을 제공합니다.  
(절대위치 점검 기능).
- S** 는 Servo 로 부터의 입력 신호이고 3 개의 연속 장치 **S**, **S + 1**, **S + 2** 를 사용할 것입니다. 장치 **S** 와 **S + 1** 은 데이터 전송을 위해 서보의 ABS (bit0, bit1) 에 연결됩니다. 장치 **S + 2** 는 전송데이터 준비 플래그를 위한 Servo 에 연결됩니다.
- D<sub>1</sub>** 은 서보(Servo)를 제어하기위한 제어 신호이고,3 개의 연속장치 **D<sub>1</sub>**, **D<sub>1</sub>+ 1**, **D<sub>1</sub>+ 2** 를 사용할 것입니다. 장치 **D<sub>1</sub>** 은 서보(Servo)의 Servo On 에 연결되고,장치 **D<sub>1</sub>+1** 는 서보(Servo)의 ABC 데이터 전송 모드에 연결되고 , **D<sub>1</sub>+2** 는 ABS 데이터 요청 신호에 연결됩니다. 자세한 사항은 아래 배선도를 참고 하십시오.



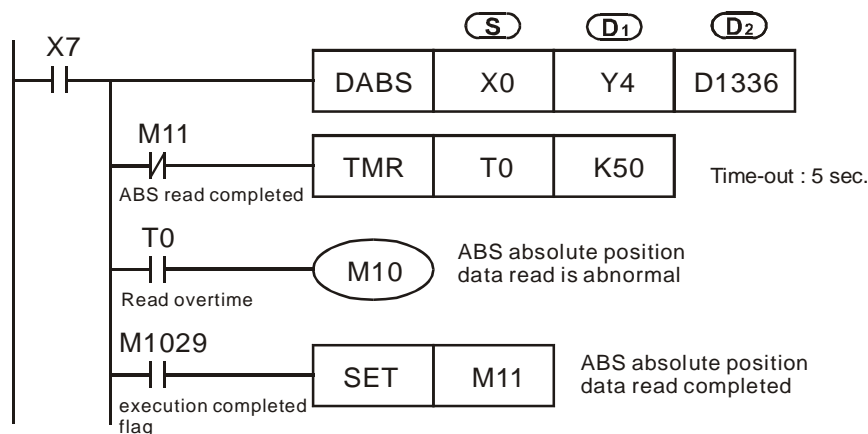
**D<sub>2</sub>** 는 Servo 로부터의 절대 위치 데이터 (32 bit)판독이고, 2 개의 연속장치 **D<sub>2</sub>**, **D<sub>2</sub>+ 1** 를 사용합니다. **D<sub>2</sub>** 는 하위 워드 이고 **D<sub>2</sub>+1** 는 상위 워드입니다. 절대 위치 데이터는 EH MPUCH0 펄스에 부합하는 현재 값 레지스터 (D1337, D1336) 또는 CH1 펄스에 부합하는 현재 값 레지스터 (D1339, D1338)에 저장되어야 합니다. 그래서 이들 두개 레지스터를 조건으로 지정하도록 요합니다. 만약 다른 장치를 열거 합니다.면 결국 사용자들은 여전히 데이터를 CH0 펄스에 부합하는 현재 값 레지스터 (D1337, D1336) 또는 CH1 펄스에 부합하는 현재 값 레지스터에 (D1339, D1338) 전송해야만 합니다. **SC MP** 에, 절대 위치 데이터는 **CH0** (Y10)펄스에 부합하는 전류 값 레지스터 (D1348, D1349)또는 **CH1**(Y11)에 부합하는 전류 값 레지스터 (D1350, D1351) 에 저장되어야 합니다. 그래서 이들 두개의 레지스터를 조건으로

지정하도록 요합니다. 만약 다른 장치들을 조건으로 지정합니다.면, 사용자들은 여전히 CH0 (Y10)펄스에 부합하는 전류 값 레지스터 (D1348, D1349)또는 CH1(Y11)에 부합하는 전류 값 레지스터 (D1348, D1349)에 데이터를 전송해야만 합니다.

4. DABSR 명령드라이브접촉이 ON로 되면, 판독이 시작하면, 명령 수행이 플래그 M1029, M1030 M1029, 완료하면, M1030은 전압이 가해질 것입니다. 플래그는 반드시 사용자에게 의해 리셋 되어야 합니다.
5. DABSR 명령이 수행중일 때, 정상적으로 개방 접촉을 조건으로 지정하십시오,만약 DABSR 명령의 드라이브 접촉이 Off로 되면 DABSR 명령 판독이 시작될 때, 절대 전류 값 판독의 수행이 중지 될 것이고, 결과는 부정확 할 것입니다. 이점에 유의, 주의 하십시오.

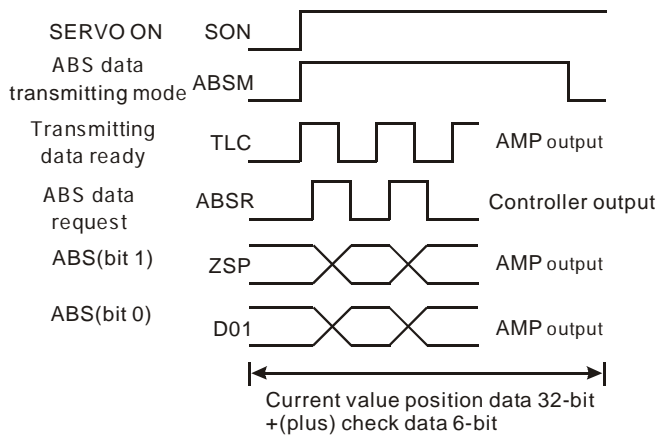
#### 프로그램 예 :

1. X7= On 일때, Servo 로 부터의 절대 위치 데이터(32 bit) 판독은 EH MPU CH0 펄스에 부합하는 전류 값 레지스터(D1337, D1336)에 저장되어야 합니다.동시에 타이머가 5 초를 세기위해 T10를 작동 시킵니다. 만약 5초가 경과하고, 절대 위치 데이터(32 bit)판독이 완료 되지 않으면, 그것이 M10=On으로 작동시킵니다. 이것은 절대위치 데이터 (32 bit) 판독이 비정상임을 의미합니다.
2. 시스템에 연결할 때, DVP-PLC와 SERVO AMP 이 동시에 켜지도록 또는 DVP-PLC의 전원보다 SERVO AMP의 전원이 먼저 켜지도록 전원을 설정하십시오.

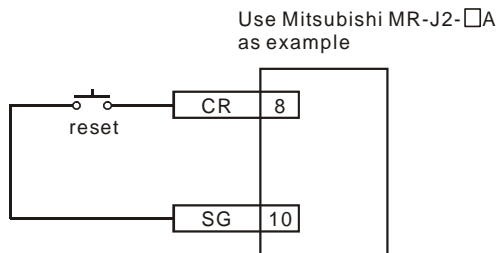


#### 주의:

1. PLC가 서보 주소 판독 명령을 수행할 때, 만약 PLC가 명령 완료 전에 판독 명령을 중지 합니다.면, 정보 메시지 (ALE5)가 Servo로 나타날 것입니다.
2. 절대위치 데이터 판독 DABSR 명령의 시간 차트 설명
  - a) DABSR 명령이 수행을 시작할 때, 그것은 Servo On(SON)의 신호와 출력하기 위한 ABS 데이터 전송 모드를 실행 시킬 것입니다.
  - b) 데이터를 전송하면서 준비 플래그와 ABS 요청 신호는 전송과 양쪽의 수취를 확인할 수 있고, 전류 값 위치 데이터와(32bit) 점검 데이터(6bit)의 데이터 전송을 처리할 수 있습니다.
  - c) 데이터는 ABS (bit0, bit1) 두개의 비트에 의해 전송됩니다.



3. 이 명령은 Mitsubishi MR-J2-A Servo 드라이브 처럼 연결된 절대 위치 기능을 갖춘 서보 모터에 적용됩니다.
4. 절대 위치 기능의 Servo 모터는 1 회전 이상 회전되어야 하고, 장비 제조 전에 리셋 신호를 받아야 합니다. 처음의 영점 복귀를 진행하기 위해서 다음의 방법중 한가지를 사용하십시오.
  - a) API 156 ZRN 명령 수행을 위해 리셋 신호 기능을 사용함으로써 영점 복귀를 완료.
  - b) 장치의 영점 위치를 조절하기 위해서 JOG 또는 매뉴얼 연산 이용 후에 리셋 신호 SERVO AMP 를 입력합니다. 리셋 신호 입력에 관한, 출력을 위한 DVP-PLC 제어기를 사용할지를 보기 위해 밑의 외부 스위치 도식을 참고 하십시오. DVP-PLC 와 Mitsubishi MR-J2-□A 간의 배선에 관한 자세한 사항은 API 159 DRVA.을 참고하십시오.



5. 표시문자 설명:
  - M1010 : EH 시리즈 MPU 에서, M1010=ON 일 때, CH0 (Y0, Y1) 과 CH1 (Y2, Y3) 는 END 명령이 수행 되는 동안 펄스를 출력 할 것입니다. 출력이 시작되면, M1010 은 자동적으로 꺼진다
  - M1029 : EH 시리즈 MPU 에서, 첫번째 그룹 (Y0, Y1) 펄스 출력을 완료하거나, 또 다른 연관 있는 명령어의 수행을 완료 한 후. M1029=On
  - M1030 : EH 시리즈 MPU 에서, 두 번째 그룹 CH1 (Y2, Y3) 펄스 출력이 완료된 후 M1030= ON
  - M1102 : SC 시리즈 MPU 에서, 첫 번째 그룹 CH0 (Y10) 펄스 출력이 완료된 후 M1102= ON
  - M1103 : SC 시리즈 MPU 에서, 두 번째 그룹 CH1 (Y11) 펄스 출력이 완료된 후 M1103= ON
  - M1336: EH 시리즈 MPU 에서, CH0 (Y0, Y1) 펄스는 표시 플래그를 출력합니다.
  - M1337: EH 시리즈 MPU 에서, CH1 (Y2, Y3) 펄스는 표시 플래그를 출력합니다.
  - M1346: EH 시리즈 MPU 에서, ZRN 명령 CLEAR 은 플래그를 작동시키는 신호를 출력합니다.
  - M1258 : EH 시리즈 MPU 에서, M1258=ON 일 때 역 방향에서 CH0 (Y0, Y1) 펄스 출력을 설정합니다.
  - M1259 : EH 시리즈 MPU 에서, M1258=ON 일 때 역 방향에서 CH1 (Y2, Y3) 펄스 출력을 설정합니다.
  - M1305 : EH MPU 에서, 그것은 명령 PLSV, DPLSV, DRVI, DDRVI, DRVA, 와 DDRV 의 CH0 (Y0, Y1)를

위한 역방향 플래그에서의 실행입니다.

M1306 : EH MPU 에서, 그것은 명령 PLSV, DPLSV, DRVI, DDRVI, DRVA, 와 DDRV 의 CH1 (Y2, Y3) 을 위한 역 방향 플래그에서의 실행입니다..

M1334 : EH 시리즈 MPU 에서, CH0 (Y0, Y1) 펄스는 M1334= On 일 때 출력을 멈춥니다.

M1334=ON 의 SC 시리즈 MPU 에서, DDRVI 와 DDRVA 명령이 실행을 멈출 때 CH0 (Y10) 펄스 출력은 감속 없이 즉시 멈출 것입니다.

M1335 : EH 시리즈 MPU 에서, CH1 (Y2, Y3) 펄스는 M1335= On 일 때 출력을 멈춥니다.

M1335=ON 의 SC 시리즈 MPU 에서, DDRVI 와 DDRVA 이 실행을 멈출 때, CH1 (Y11)펄스 출력은 감속 없이 즉시 멈출 것입니다.

M1336 : EH 시리즈 MPU 에서, CH0 (Y0, Y1)펄스는 표시 플래그를 출력합니다.

M1337 : EH 시리즈 MPU 에서, CH1 (Y2, Y3) 펄스는 표시 플래그를 출력합니다.

M1346 : EH 시리즈 MPU 에서, ZRN 명령 CLEAR 는 플래그 가능한 신호를 출력합니다.

#### 6. 특별한 레지스터 설명:

D1337, D1336: 1. EH MPU 에서, 제어 명령(API 156 ZRN, API 157 PLSV, API 158 DRVI, API 159 DRVA) 의 위치를 지정하는 현재 값 레지스터를 나타내는 D1337 (HIGH WORD), D1336(LOW WORD) 은 현재 값이 회전의 방향으로 감소하거나 증가하는 첫 번째 출력 그룹 Y0, Y1 에 출력합니다.

2. EH MPU 에서, 펄스 출력 명령(API 57 PLSY, API 59 PLSR) 의 출력 펄스의 총 번호를 나타내는 D1337(HIGH WORD), D1336(LOW WORD) 는 첫 번째 출력 그룹 Y0, Y1 에 출력합니다.

D1338, D1339: 1. EH MPU 에서, 제어 명령 (API 156 ZRN, API 157 PLSV, API 158 DRVI, API 159 DRVA)의 위치를 지정하는 현재 값 레지스터를 나타내는 D1339(HIGH WORD), D1338(LOW WORD) 는 현재 값이 회전의 방향으로 감소하거나 증가하는 첫 번째 출력 그룹 Y2, Y3 에 출력합니다.

2. EH MPU 에서, 펄스 출력 명령(API 57 PLSY, API 59 PLSR) 의 출력 펄스의 총 번호를 나타내는 D1339(HIGH WORD), D1338(LOW WORD) 은 첫 번째 출력 그룹 Y2, Y3 에 출력합니다.

D1340(D1352) 제어 명령(API 156 ZRN, API 158 DRVI, API 159 DRVA) 의 위치를 찾는 출력 CH0 (CH1)이 수행될 때, 첫 단계 가속과 마지막 단계 감속의 주파수 설정으로 사용됩니다.

설정 범위: EH MPU 에서, 출력 속도는 10Hz 보다 작을 수 없습니다. 만약 속도가 10Hz 보다 작거나 또는 출력 주파수의 최고 값 보다 크면, 10Hz 이 우세할 것입니다.

공장 설정: 200Hz. SC 시리즈에서, 출력 속도는 100~100kHz 의 범위에 이른다. 만약 속도가 100Hz 보다 작다면, 100Hz 는 출력에 우세할 것입니다. 만약 100kHz 보다 크다면, 100kHz 는 우세할 것입니다. 공장 설정: 100Hz.

참고: 스테핑 모터를 제어할 때, 속도를 설정하는 동안 스테핑 모터의 반향과 초기 주파수의 한계를 고려하십시오..

D1341, D1342: EH MPU 에서, 제어 명령((API 156 ZRN, API 158 DRVI, 156 ZRN, API 158 DRVI, API 159

DRVA)의 위치를 찾는 것이 수행 될 때, D1342(HIGH WORD), D1341(LOW WORD)은 최대 속도 설정으로 나타냅니다.

D1343(D1353) 제어 명령(API 156 ZRN, API 158 DRVI, API 159 DRVA) 의 위치를 찾는 출력 CH0 (CH1) 이 수행될 때, 첫 단계 가속과 마지막 단계 감속의 가속. 감속 시간 설정으로 사용됩니다. 설정 범위: EH 시리즈 MPU 에, 가속/감속 시간은 10ms 보다 짧을 수 없습니다. 만약 설정 시간이 10ms 보다 짧고 10,000ms 보다 길다면, 10ms 이 출력에 우세할 것입니다. 공장 설정: 100ms. SC MPU 에, 시간 설정은 50~20,000ms 의 범위에 이른다. 만약 시간 설정이 범위밖에 있습니다.면, 20,000ms 이 상위 한계에 우세할 것이고 반면 50ms 은 하위 한계에 우세할 것입니다.

참고: 스테핑 모터를 제어할 때, 속도를 설정하는 동안 스테핑 모터의 반향과 초기 주파수의 한계를 고려하십시오.

D1348.D1349 SC 시리즈에서, D1348 (High word) 와 D1349 (Low word)는 제어 명령(API 156 ZRN · API 158 DRVI · API 159 DRVA)의 위치를 찾는 첫 번째 그룹 Y10 이 수행 될 때 현재 펄스 출력 값을 저장하는데 사용됩니다. 기록된 현재 펄스 출력 값 순방향 회전과 역방향 회전 때문에 증가할 것입니다.

D1350 .D1351 SC 시리즈에서, D1350 (Low word) 와 D1351 (High word)는 제어명령 (API 156 ZRN · API 158 DRVI · API 159 DRVA) 의 위치를 찾는 첫 번째 그룹 Y10 이 수행 될 때 현재 펄스 출력 값을 저장하는데 사용됩니다. 기록된 현재 펄스 출력 값 순방향 회전과 역방향 회전 때문에 증가할 것입니다.

API	의사기호		피연산자				기능		제어기						
156	D	ZRN		<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	<b>(S<sub>3</sub>)</b>	<b>(D)</b>	영점 복귀	ES	EX	SS	SA	SX	SC	EH

Type	비트 장치				워드 장치										프로그램 단계	
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*	
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*	
S <sub>3</sub>	*	*	*	*												
D		*														

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

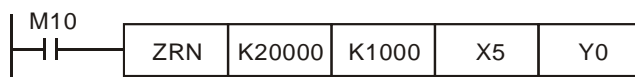
S<sub>1</sub>: 영점 복귀 속도    S<sub>2</sub>: 크리프 속도    S<sub>3</sub>: 접근 점 신호(DOG)    D: 펄스 출력 장치

설명:

- S<sub>1</sub> 은 영점 복귀 속도로 지정됩니다. EH MPU 에, 16-bit command 명령의 설정 범위는 10~32,767Hz 이고, 32-bit 명령은 10~200,000Hz 입니다. 나타난 속도가 10Hz 보다 작으면, 10Hz 이 영점 복귀 속도로서 우위를 차지할 것입니다. 비슷하게, 속도가 200kHz 보다 높으면, 200kHz 이 우위를 차지할 것입니다. SC MPU 에, 32-bit 명령의 설정 범위는 100~100,000Hz 입니다. 나타난 영점 복귀 속도가 100Hz 보다 작으면, 100Hz 이 우위를 차지할 것입니다. 만약 100kHz 보다 높으면, 100kHz 이 우위를 차지할 것입니다.
- S<sub>2</sub> 는 접근 점 신호(DOG) 가 켜진후에 더 낮은 속도, 사용가능 범위가 10 to 32,767Hz 인 크리프 속도로 지정됩니다. SC MPU 에 사용가능한 범위는 100~100,000Hz 입니다.
- S<sub>3</sub> 는 접근 점 신호 (DOG)출력(A contact input)으로 지정됩니다. EH MPU 에, 만약 X10~X17 이외의 접근 점 신호 출력을 X, Y, M 과 S 로 지정합니다.면 X, Y, M 와 S 가 PLC 의 주사 주기에 의해 영향을 받기 때문에 영점의 분산이 발생할 것입니다. 또한, ZRN 의 출력은 DCNT 와 PWD 에 의해 나타난 출력 X10~X17 과 같을수 없습니다. SC MPU 에, 오직 X10 와 X11 이 사용될수 있습니다. ZRN 의 출력은 DCNT 에 의해 나타난 출력 X10, X11 과 같을 수 없습니다.
- 펄스 출력장치 **D** 에, EH MPU 은 Y0 와 Y2 에 나타낼 수 있습니다. while SC MPU can indiate to Y10 and Y11.
- 상대 위치를 위한 API 158 DRVI 과 절대 위치를 위한 API 159 DRVA 명령을 실행할 때, PLC 는 현재 값 레지스터(for EH MPU, Y0: D1337,D1336, Y2: D1339, D1338; for SC MPU, Y10: D1348, D1349, Y11: D1350, D1351) 안에서 작동하는 동안 증가 또는 감소하는 FWD/REV 에 저장합니다. 그래서 그것은 기계 위치로 항상 알려질 수 있습니다.그러나 그 데이터는 PLC 의 전력이 꺼지면 손실 될 수 있습니다. 그러므로 기계의 영점 복귀 데이터 안에서 처음 수행하는 동안 영점 복귀의 데이터 기록이 필요합니다.

프로그램 예 :

M10= On 일 때, a frequency of 20KHz 의 주파수는 모터가 영점 복귀 동작을 수행하도록 하기 위해 Y10 으로부터 출력합니다. 그것이 접근 점 신호(DOG)에 도달하면, X5= On 과 그것은 크리프 속도를 바꿀 것입니다. 그러면 1KHz 의 주파수는 Y0 로부터 출력하고 명령은 전압을 받게 될 것입니다. 펄스 출력은 X5=Off 일 때 까지 멈출 것입니다.



주의:

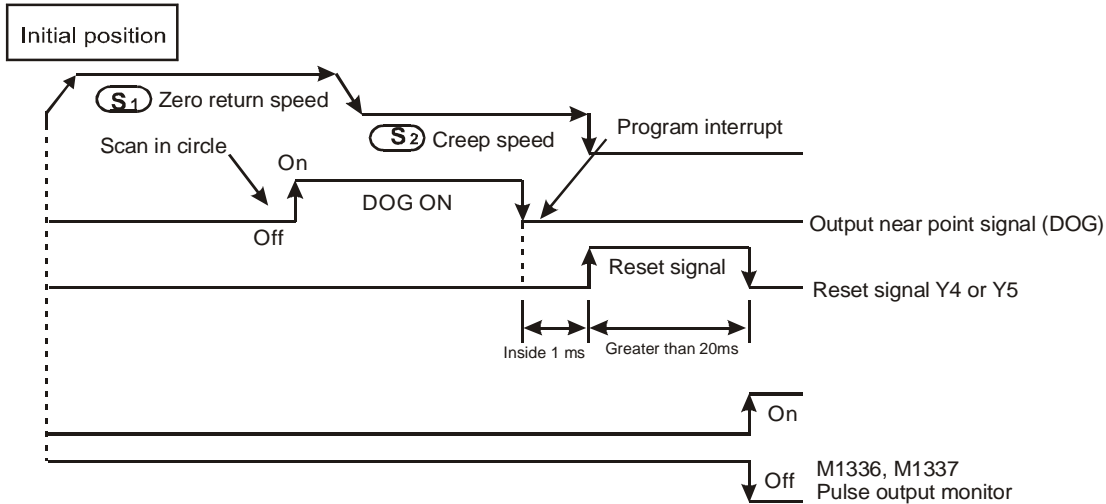
1. 리셋 신호 출력의 타임 차트 설명: EH MPU 를 위한 명확한 신호 출력의 순차적인 도표 설명은 SC MPU 에는 적합하지 않다.

a) 리셋 플래그 M1346= On 일 때, 리셋 신호는 영점 복귀가 완료됐을 때 서보로 보내진다.

b) 리셋 신호의 출력 장치 I:

CH0(Y0, Y1) 리셋 출력 장치 (Y4)

CH1(Y2, Y3) 리셋 출력 장치 (Y5)



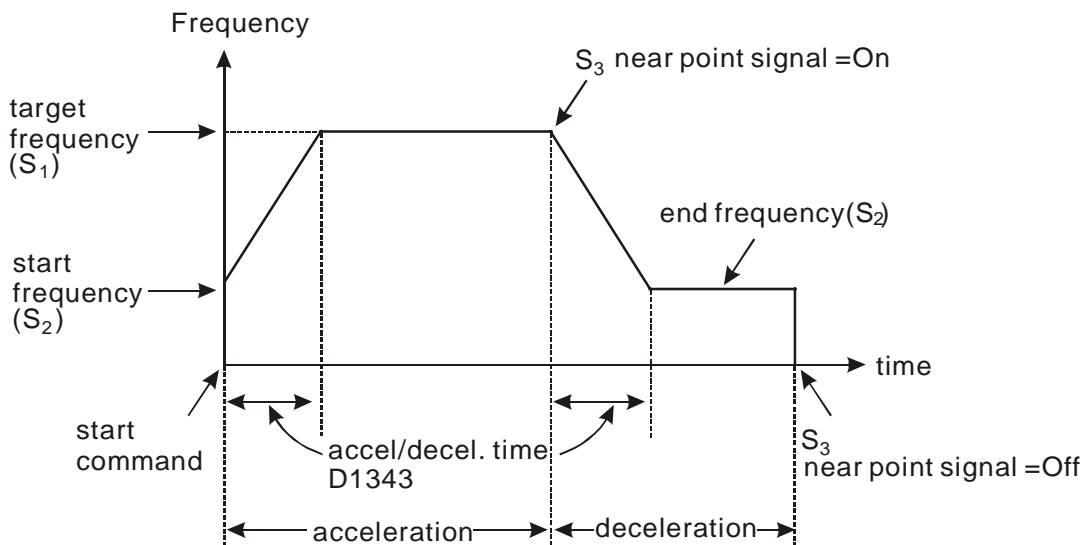
참고: 표시된 외부 출력 X10~X17 을 포함하면, X, Y, M, 와 S 같은 다른 장치들은 가장 나쁜 영향으로써 주기의 두배인 스캔 주기에의해 영향을 받을 것입니다.

2. 영점 복귀 작동의 설명:

- a) ZRN 이 수행될 때, EH 시리즈 모델에, CH0 (CH1)출력의 첫 번째 단계 가속의 주파수는 D1340 (D1352)에 의해 설정됩니다. SC 시리즈 모델에, CH0 (CH1)출력은 초기 주파수로써 크리프 속도를 사용할 것입니다.; EH/ES 시리즈 모델에, 속도는 가속/감속 시간 D1343 (D1353) 설정에 의해 영점 복귀 속도 S1 에 가속될 것입니다. 접근 점 신호(DOG)가 OFF 로부터 ON 으로 가면, 속도는 가속/감속 시간 설정 D1343 (D1353)에 의해 크리프 속도 S2 까지 감속될 것입니다.
- b) 접근 점 신호(DOG)가 Off 로부터 On 으로 가면, 속도는 가속/ 감속시간 설정 D1343/D1353.에 의해 크리프 속도 S<sub>2</sub> 까지 감속될 것입니다.
- c) 접근 점 신호(DOG) On 으로부터 Off 로 가면 펄스 출력이 멈추는 동시에, CH0 펄스의 현재 값 레지스터(D1337, D1336) 의 내용 값 또는 CH1 펄스의 현재 값 레지스터 (D1339, D1338)이 0 이 될 것입니다. 또한, 만약 리셋 신호 플래그 M1346= On 이면, 리셋 신호 Y4 (CH0)또는 Y5(CH1) 일제히 출력할 것입니다. SC MPU 에, "0"은 CH1 의 현재 값 레지스터는 CH0 펄스의 현재 값 레지스터(D1348, D1349) 와 CH1 의 현재값 레지스터(D1350, D1351) 에 쓰여질 것입니다.
- d) EH 시리즈 MPU 에, 펄스 출력의 작동이 완료되고, 플래그 M1029, M1030 이 사용될 때, CH0 펄스에 의해 보내진 표시 플래그 M1336 또는 CH1 펄스 에 의해 보내진 표시 플래그 M1337 는 Off 가 될 것입니다. SC MPU 에, 펄스 출력의 작동이 완료되었을 때, , 플래그 M1102 와 M1103 는 사용됩니다.
- e) ZRN(DZRN)명령이 접근 점 신호 (DOG)의 위치를 찾을 수 없기 때문에, 영점 복귀는 오직 한 방향 안에서만 진행될 수 있습니다. EH 시리즈 MPU 에, CH0 펄스의 현재 값 레지스터(D1337, D1336) 의 값 또는 영점 복귀를 위한 현재 값 레지스터(D1339, D1338) 의 값은 감소할 것입니다. SC MPU 에, CH0



펄스의 현재값 레지스터(D1348, D1349)의 값 또는 현재 값 레지스터(D1350, D1351)의 값은 감소할 것입니다.



- f) ZRN 명령은 전원이 꺼져도 현재 위치를 기록할 수 있는 Mitsubishi MR-J2-A 와 같은 절대 위치 추적의 기능으로 서보 모터에 적용 가능합니다. 게다가 서보의 현재 위치가 DVP-EH/SC 시리즈 PLC 의 API 155 DABSR 명령에 의해 읽혀 질 수 있기 때문에, ZRN 명령은 단 한번만 실행 되어져야 합니다. 전원이 꺼진후 ZRN 명령을 수행하기위해 ZRN 명령을 재 실행할 필요가 없습니다.
- g) EH 시리즈 MPU 에, ZRN 명령이 수행되면, CH0 와 CH1 의 펄스 출력의 현재값은 (D1337, D1336) 와 (D1339, D1338)에 표시될 것입니다. ZRN 이 완료되면, 0 이 (D1337, D1336) 와 (D1339, D1338)에 읽혀질 것입니다. SC 시리즈 MPU 에, ZRN 명령이 실행되면, CH0 와 CH1 의 펄스 출력의 현재 값이 (D1348, D1349) 와 (D1350, D1351)에 표시될 것입니다. ZNR 이 완료 되면, "0" 은 (D1348, D1349) 와 (D1350, D1351)에 읽혀질 것입니다.
- h) ZRN 명령의 드라이브 접촉이 On 일 때, CH0 와 CH1 펄스는 가속 시간으로 D1343/D1353 에 의해 설정된 내용 값을 읽을 것입니다. 영점 복귀 스피드까지 가속한 후에, 접근 점 신호(DOG)의 입력을 기다리고, 감속에 의한 낮은 속도의 크리프 속도를 출력합니다. 접근 점 신호(DOG)가 꺼지면 즉기 출력 펄스를 멈춥니다.
- i) SC MPU 에, 많은 ZRN 명령은 프로그램에서 동일하게 구성됩니다. 그러나 언제 PLC 가 프로그램을 실행 할때마다 간에 오직 하나의 명령만이 실행 될 수 있습니다. 예를 들면, 만약 Y10 출력이 명령에 의해 이미 실행 되었다면, Y10 출력을 실행시키는데 사용된 다른 명령어는 실행되지 않을 것입니다. 그래서 명령수행 우선원이 명령 실행의 순서에 기반합니다.
- j) SC MPU 에, 출력 장치로 Y10 을 사용할 때, 사용자는 가속에서부터 감속에 이르는 변환 지역의 근접 점 신호 출력으로 X10 또는 X11 을 선택할 수 있습니다. 유사하게, 출력 장치로 Y11 을 사용할 때, 근접 점 신호 출력으로 X10 또는 X11 을 선택할 수 있습니다.
- k) SC MPU 에, ZRN 명령을 위한 출력 번호는 없습니다., Y10 을 사용할 때, 변환 상태가 근접 점 신호(from OFF to ON)에 의해 입력될 것입니다. 반면 그 명령은 가속으로부터 감속에 이르지 않을 것입니다. 또한 ON 을 위한 트리거 시간은 10us 이상이 더 좋고, 또는 어떤 반응이 없는 소음으로써 간주될 것입니다.
- l) SC MPU 에, 명령이 감속지대로 들어가고 출력 주파수가 크리프 속도에 도달하면, 출력은 근접 점 신호가 ON 으로부터 OFF 일 때 멈출 것입니다.

- m) SC MPU 에, Y10 의 현재 축적된 출력 번호는 D1348 와 D1349 에, Y11 의 번호는 D1350 와 D1351 에 저장됩니다. 프로그램이 STOP 에서 RUN 까지 또는 RUN 에서 STOP 까지 일 때 그것은 “0”까지 소거되지 않을 것입니다.
- n) SC MPU 에, M1102=ON 일 때, Y10 펄스 출력 이 끝남을 의미합니다. M1103=ON 일 때, Y11 펄스 출력이 끝남을 의미합니다.
- o) SC MPU 에, ZRN 명령이 ON 일 때, 모든 매개 변수는 명령이 꺼질 때 까지 변경될 수 없습니다.
- p) SC MPU 에, ZRN 명령이 OFF 일 때, 모든 출력은 출력이 무엇이건 상관없이 멈출 것입니다.

API	의사기호			피연산자	기능	제어기						
157	D	PLSV		<b>S</b> <b>D<sub>1</sub></b> <b>D<sub>2</sub></b>	가변속도 펄스 출력	E S	E X	S S	S A	S X	S C	E H

Type OP	비트 장치				워드 장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PLSV: 7 steps DPLSV: 13 steps
S					*	*	*	*	*	*	*	*	*	*	*	
D <sub>1</sub>		*														
D <sub>2</sub>		*	*	*												

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**피연산자:**

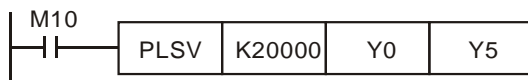
S: 펄스 출력 주파수 D<sub>1</sub>: 펄스 출력 장치(출력 모드로 트랜지스터를 사용하십시오..) D<sub>2</sub>: 회전 방향 신호

**설명:**

- S 는 펄스 출력 주파수 16-bit 1 에서 32,767Hz 과-1 에서-32,768 Hz 또는 32-bit 1 에서 200,000Hz, -1 에서 -200,000 Hz. 로서 지정됩니다. (+) 와 (-) 부호는 양방향, 음 방향을 나타냅니다.펄스가 출력중 일때에도 바뀔 수 있습니다.
- 펄스 출력 장치 D<sub>1</sub> 는 오직 Y0, Y2 로 지정될 수 있습니다. .
- D<sub>2</sub> 는 회전 방향 신호로 지정되고, 그것은 S 의 양극을 따라 작용합니다.S가 양 (+)일 때 D<sub>2</sub> 는 켜지고 S 가 음(-)일 때, D<sub>2</sub> 는 꺼진다.
- PLSV 명령은 가속/ 감속 설정 기능을 갖고 있지 않다. 그러므로 가속/ 감속은 시작과 멈춤에서 수행오디지 않습니다. 만약, 쿠션 시작과 멈춤이 요구되면, API 67 RAMP 명령을 사용하여 출력 펄스 주파수를 줄이거나 늘리시오
- PLSV 명령이 출력 펄스를 수행하는 동안에 만약 드라이브 접촉이 꺼지면, 기계는 감속 없이 멈출 것입니다.
- PLSV 명령의 드라이브 접촉이 꺼지면, 그 펄스가 CH0 펄스의 표시 플래스 M1336 을 보냈거나 설정된 CH1 펄스 의 표시 플래스 M1337 을 보냈을 지라도 PLSV 를 실행하는 것은 불가능합니다.
- DPLSV 명령에, 만일, 입력 주파수의 절대 값이 200kHz 이상이면, 200kHz 이 우세일 것입니다.
- D1222 (D1223)는 설정 방향의 신호와 CH0 (CH1)의 펄스 출력점 간의 시간차입니다.

**프로그램 예 :**

M10= On 일 때, 20KHz 주파수는 Y0 으로부터 출력합니다. Y5= On 은 양방향(음,양)을 나타냅니다.



API	의사기호		피연산자	기능	제어기						
158	D	DRVI	<b>S<sub>1</sub></b> <b>S<sub>2</sub></b> <b>D<sub>1</sub></b> <b>D<sub>2</sub></b>	상대 위치 제어	E S	E X	S S	S A	S X	S C	E H

Type OP	비트 장치				워드 장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DRVI: 9 steps DDRVI: 17 steps
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*	
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*	
D <sub>1</sub>		*														
D <sub>2</sub>		*	*	*												

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S<sub>1</sub>**: 펄스의 번호(목표 장치)    **S<sub>2</sub>**: 펄스 출력 주파수 **D<sub>1</sub>**: 펄스 출력 지시 장치 **D<sub>2</sub>**: 회전 방향 신호

설명:

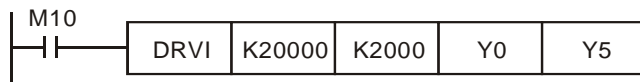
- SC 시리즈 MPU 에, 오직 32-bit DDRVI 명령이 작업가능하고, 피연산자 S<sub>1</sub>/S<sub>2</sub> 는 오직 K, H, 와 D 장치를 돕는다. SA/SX MPU 에, 이명령은 도움되지 않습니다.
- S<sub>1</sub> 는 펄스 (상대 위치)의 번호로 지정됩니다. S<sub>1</sub> 의 가능한 번호는: EH 시리즈 MPU 의, 16-bit 명령: -32,768 에서+32,767 과 32-bit 명령: -999,999 에서 +999,999 입니다. 양 (+)과 음(-)의 기호는 순방향과 역방향을 나타냅니다. SC MPU 에, 32-bit 명령: -2,147,483,648 ~ +2,147,483,647 입니다. 양(+) 과 음(-)기호는 순방향과 역방향을 나타냅니다.
- S<sub>2</sub> 는 펄스 출력 주파수로 지정됩니다. S<sub>2</sub> 의 사용가능한 번호는: EH 시리즈 MPU 에, 16-bit 명령: 10 에서 32,767Hz 와 32-bit 명령: 10 에서 200,000Hz 입니다. SC MPU 에서, 32-bit 명령: 100 ~ 100,000Hz 입니다.
- D<sub>1</sub> 는 펄스 출력 지시 장치로 지정됩니다. EH 시리즈 모델 에서, 그것은 오직 Y0,Y2 로 지정될 수 있습니다. SC MPU 에서, 그것은 Y10, Y11 로 지정될 수 있습니다.
- D<sub>2</sub> 는 회전 방향 신호로 지정되며, S 의 우선순에 따라 작동합니다. S 이 양(+)일 때 ,D<sub>2</sub> 는 On 입니다. S 가 음(-)일 때, D<sub>2</sub> 는 Off 입니다.
- EH 시리즈 MPU 에, 펄스의 번호는 CH0(Y0, Y1) 펄스의 현재값 레지스터(D1337 high byte, D1336 low byte)또는 CH1(Y2, Y3) 펄스의 현재 값 레지스터(D1339 high byte, D1338 low byte) 에 저장될 것입니다. 회전 방향이 음일 때, 현재 값 레지스터의 내용 값은 감소할 것입니다.
- SC 시리즈 MPU 에, 지시된 펄스 출력 번호 S<sub>1</sub> 은 CH0 (Y10) 펄스의 현재 값 레지스터 D1348 (low word) 와 D1349 (high word)또는 현재값 레지스터 D1350 (low word) and D1351 (high word)의 32-bit 명령 데이터 내용의 상대 위치에 저장될 것입니다.회전 방향이 음 일 때, 현재 값 레지스터의 값은 감소할 것입니다. 프로그램이 STOP 에서 RUN 으로 또는 RUN 에서 STOP 으로 일 때, 현재 값 레지스터의 값은 똑같이 남아있을 것입니다.
- DRVI 명령이 수행되는 동안 각 피연산자의 내용은 바뀔 수 없습니다. 내용은 다음 수행이 실행될 때 바뀔 것입니다.
- DRVI 명령이 수행될 때 만약 드라이브 접촉이 꺼지면 EH 시리즈 MPU 를 위해 기계가 감속하고, 멈출 것이고, 수행이 완료된 플래그 M1029, M1030 does 는 켜지지 않습니다. SC MPU 에, 명령 수행 완료 후, 플래그 M1102 와 M1103 는 ON 일 것입니다..
- EH MPU 에, DRVI 의 드라이브 접촉이 꺼지면, CH0 펄스의 표시 플래그 M1366 에 보내진 펄스 또는

CH!펄스의 표시 플래스 M1377 이 보내진 펄스라 하더라도 DRV! 명령을 다시 수행하는건 불가능합니다.

11. EH 시리즈 MPU 에, DDRVI 를 위한 출력 주파수의 절대 값이 200kHz 이상일 때, 200kHz 은 우세합니다.또는 그것이 10Hz 보다 작을 때, 10Hz 이 우세합니다.
12. D1343 (D1353)는 CH0 (CH1) 펄스를 위한 첫 단계 가속의 시간을 정하는데 사용됩니다. 10,000ms, 10ms 은 출력에 우세 할 것입니다. 공장 설정: 100ms. SC MPU 에, 가속/감속의 사용가능한 범위는 50~20,000ms 입니다. 만약 사용가능한 범위 밖에 있습니다.면, 20,000ms 이상출력 에서 20,000ms 이 우세하고, 50ms 보다 작은 출력은 50ms 이 우세합니다.
13. D1340 (D1352) 은 Y10 (Y11)에 ON/OFF 의 주파수를 설정하기 위해 사용됩니다.만약 S<sub>2</sub>에 의해 나타난 출력 주파수가 Y10 (Y11) 의 ON/OFF 주파수의 주파수와 같거나 더 작다면, ON/OFF 주파수는 펄스 출력 주파수로 수행될 것입니다.
14. D1222 (D1223)설정 방향 신호와 CH0(CH1)의 펄스 출력 점간의 시간 차입니다.

프로그램 예 :

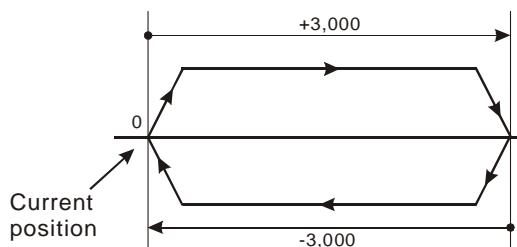
M10= On 일 때, Y0(상대 위치 찾기)로부터 2KHz 주파수 펄스의 twenty thousands (20000) 을 출력합니다. Y5= On 은 양 방향을 나타냅니다.



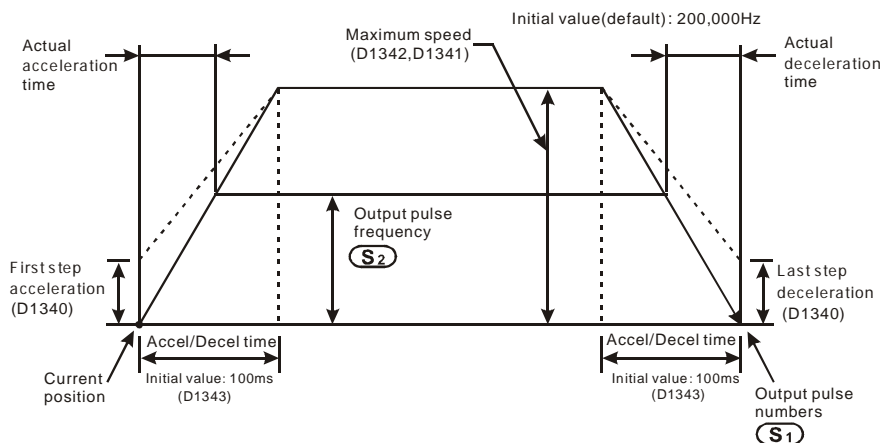
주의:

1. EH 시리즈 MPU 의 설명:

- a) 상대 위치 찾기 제어의 작동 설명: 현재 위치로부터 이동 거리를 지정하기 위한 양 또는 음의 부호 사용은 상대 위치 제어의 실행 방법 중 하나입니다.

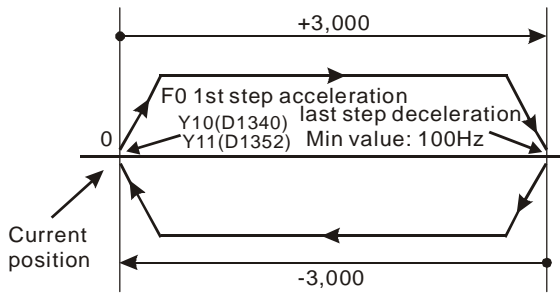


- b) 상대 위치찬기의 설정과 작동 속도:

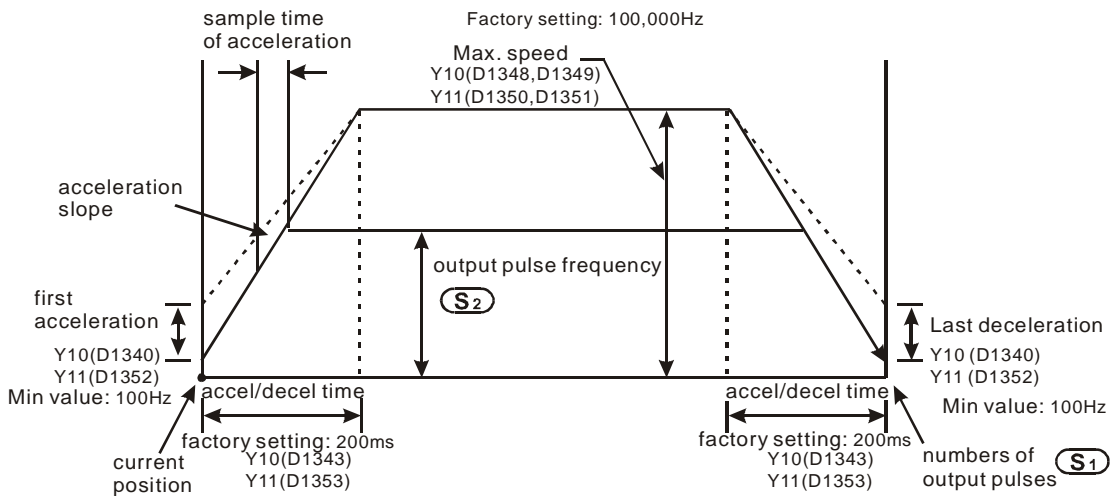


2. SC 시리즈 MPU 의 설명:

- a) 상대 위치찾기 제어의 작동 설명: 현재 위치로부터 이동 거리를 지정하기 위한 양 또는 음의 부호 사용은 상대 위치 제어의 실행 방법 중 하나입니다.



- b) 상대 위치의 설정 아이템과 가속/ 감속 D1343 (D1353) 의 설정은 첫 단계 가속의 시간을 설정하는데 사용됩니다. D1340 (D1352) 은 Y10 (Y11)을 위해 마지막 단계 감속의 시간을 정하는데 사용됩니다. D1340 (D1352) 은 Y10 (Y11). 를 위한 처음과 끝 주파수를 설정하는데 사용됩니다.



- c) SC MPU 에, 많은 DRVI 명령이 프로그램안에서 동일하게 구성될 수 있습니다.그러나 PLC 가 프로그램을 실행 할때마다 오직 하나의 명령이 실행 될 수 있습니다. . 예를 들면, 만약 Y10 출력이 명령에 의해 이미 실행 되었다면, Y10 출력을 실행시키는데 사용된 다른 명령어는 실행되지 않을 것입니다.그래서 명령수행 우선원이 명령 실행의 순서에 기반합니다.
- d) Y10 가 DDRVI 명령에 의해 수행될 때, Y10 의 원래의 기능은 DDRVI 이 OFF 될때까지 가능하지 않을 것입니다. Y11 역시 그럴것입니다.
- e) DDRVI 가 수행되면, 모든 다른 매개변수들이 DDRVI 이 OFF 가 될때까지 변경될 수 없습니다.
- f) 이 명령이 꺼지고, 출력이 완료되지 않았을 때:  
만약 M1334=ON 이면, Y10 은 출력을 즉시 멈출 것입니다.  
만약 M1334=OFF 이면, Y10 은 감속 시간으로부터 끝의 주파수에 까지 감속될 것이고 펄스 출력을 멈출 것입니다.  
Y11 출력에 일치하는 M1335 와 같은 방법입니다..

플래그 설명:

- M1010: EH 시리즈 MPU 에서, M1010= On 일 때, Y0, Y1 와 Y2, Y3 은 END 명령이 수행되는 동안 펄스를 출력할 것입니다.
- M1029: EH 시리즈 MPU 에서, Y0, Y1 펄스 출력 완료 후 M1029= On
- M1030: EH 시리즈 MPU 에서, Y2, Y3 펄스 출력 완료 후 M1030= On.

- M1102 : SC 시리즈 MPU 에서, 첫 번째 그룹 CH0 (Y10) 펄스 출력 완료 후 M1102= ON.
- M1103 : SC 시리즈 MPU 에서, 두 번째 그룹 CH1 (Y11) 펄스 출력 완료 후 M1103= ON
- M1258 : EH 시리즈 MPU 에서, M1258=ON 일 때 역 방향에서 CH0 (Y0, Y1) 펄스출력을 설정하십시오.
- M1259 : EH 시리즈 MPU 에서, M1259=ON 일 때 역방향에서 CH1 (Y2, Y3) 펄스출력을 설정하십시오.
- M1305 : EH MPU 에서, 그것은 플래그 for CH0 (Y0, Y1) of 명령어 PLSV, DPLSV, DRVI, DDRVI, DRVA, 와 DDRV 의 CH0 (Y0, Y1)를 위한 역방향 플래그에서 작동합니다.
- M1306 : EH MPU 에서, 그것은 플래그 for CH0 (Y2, Y3) of 명령어 PLSV, DPLSV, DRVI, DDRVI, DRVA, 와 DDRV 의 CH0 (Y0, Y1)를 위한 역방향 플래그에서 작동합니다.
- M1334: EH 시리즈 MPU 에서, M1334= On 일 때, CH0 (Y0, Y1) 펄스는 출력을 멈춥니다.  
SC 시리즈 MPU 에서 M1334=ON 으로, DDRVI 와 DDRVA 명령이 수행을 멈출 때 CH0 (Y10) 펄스 출력은 감속 없이 즉시 멈출 것입니다.
- M1335: EH 시리즈 MPU 에서, M1335= On 일 때 CH1 (Y2, Y3) 펄스 출력을 멈춥니다.  
SC 시리즈 MPU 에서 M1335=ON 으로, DDRVI 와 DDRVA 명령이 수행을 멈췄을 때, CH1 (Y11) 펄스 출력은 감속 없이 즉시 멈출 것입니다.
- M1336: EH 시리즈 MPU 에서, CH0 (Y0, Y1) 펄스 출력 표시 플래그
- M1337: EH 시리즈 MPU 에서, CH1 (Y2, Y3) 펄스 출력 표시 플래그

### 3. 특별한 레지스터 설명:

- D1222(D1223): EH 시리즈 MPU 를 위한 CH1 (CH0) 의 펄스 출력과 방향 신호간의 시간차 설정 .
- D1337, D1336: 1. EH MPU, D1337(HIGH WORD), D1336(LOW WORD)는 회전의 방향에 따라 현재 값이 증가 또는 감소하는 첫 번째 출력 그룹 Y0, Y1 에 출력하는 제어 명령(API 157 PLSV, API 158 DRVI, API 159 DRVA) 의 위치를 지정하는 현재 값 레지스터를 나타냅니다.
2. EH MPU 에서, D1337(HIGH WORD), D1336(LOW WORD)은 첫 번째 출력 그룹 Y1, Y0,에 출력하는 펄스 출력 명령(API 57 PLSY, API 59 PLSR) 의 총 번호를 나타냅니다.
- D1338, D1339: 1. EH MPU 에서, D1339(HIGH WORD), D1338(LOW WORD)은 EH MPU, D1337(HIGH WORD), D1336(LOW WORD)는 회전의 방향에 따라 현재 값이 증가 또는 감소하는 첫 번째 출력 그룹 Y2, Y3 에 출력하는 제어 명령(API 157 PLSV, API 158 DRVI, API 159 DRVA) 의 위치를 지정하는 현재 값 레지스터를 나타냅니다.
2. EH MPU 에서, D1339(HIGH WORD), D1338(LOW WORD)는 첫 번째 출력 그룹 Y3, Y2,에 출력하는 펄스 출력 명령(API 57 PLSY, API 59 PLSR) 의 총 번호를 나타냅니다.
- D1340 (D1352): 제어 명령 API 156 ZRN, API 158 DRVI, API 159 DRVA) 위치지정의 출력 CH0 (CH1)가 수행 될 때, 첫 단계 가속과 마지막 단계 감속의 주파수 설정으로 사용됩니다.
- 설정 범위: EH MPU 에서, 출력 속도는 10Hz 보다 작을 수 없습니다. 만약 속도가

10Hz 보다 작거나 출력 주파수의 최고 값 보다 크면, 10Hz 이 우세할 것입니다. 공장 설정: 200Hz. SC 시리즈에, 출력 속도는 100~100kHz 범위 내에 있습니다. 만약 속도가 100Hz 보다 작으면, 100Hz 는 출력에 우세할 것입니다. 만약 100kHz 보다 작으면, 100kHz 가 우세할 것입니다. 공장 설정: 100Hz.

참고: 스텝핑 모터를 제어할 때, 속도가 설정되는 동안 스텝핑 모터의 반향과 초기 주파수의 한계를 고려하십시오.

**D1341, D1342:** EH MPU 에서 제어 명령(API 156 ZRN, API 158 DRVI, API 159 DRVA) 의 위치지정이 수행될 때 D1342(HIGH WORD), D1341(LOW WORD) 는 최대 속도 설정으로 나타냅니다.

설정 범위: 10 에서 200,000Hz 까지, 공장 설정(default)은 200,000Hz 입니다.

참고: 출력 펄스 주파수는 반드시 최대 속도 아래 있어야 하는 API 158 DRVI 명령의 피연산자에 의해 지정됩니다.

**D1343 (D1353):** 제어 명령(API 156 ZRN, API 158 DRVI, API 159 DRVA) 위치지정의 출력 CH0 (CH1) 이 수행 될 때, 첫 단계 가속과 마지막 단계 감속의 가속/감속 시간 설정으로 사용됩니다.

설정 범위: EH 시리즈 MPU에, 가속/감속 시간은 10ms 보다 짧을 수 없습니다. 만약 설정 시간이 10ms 보다 짧거나 또는 10,000ms 보다 길면, 10ms 이 출력에 우세 할 것입니다. 공장 설정: 100ms. SC MPU 에, 시간 지정은 50~20,000ms 범위 내에 있습니다. 만약 시간 설정이 범위 밖에 있습니다.면, 20,000ms 이 상위한계에 우세할 것이고, 반면 50ms 이 하위 한계에 우세할 것입니다.

참고: 스텝핑 모터를 제어할 때, 속도가 설정되는 동안 스텝핑 모터의 반향과 초기 주파수의 한계를 고려하십시오..

**D1348 · D1349 :** SC 시리즈에서, D1348 (High word) 와 D1349 (Low word)는 제어 명령(API 156 ZRN · API 158 DRVI · API 159 DRVA)의 위치를 찾는 첫 번째 그룹 Y10 이 수행 될 때 현재 펄스 출력 값을 저장하는데 사용됩니다. 기록된 현재 펄스 출력 값 순방향 회전과 역방향 회전 때문에 증가할 것입니다.

**D1350 · D1351 :** SC 시리즈에서, D1350 (Low word) 와 D1351 (High word)는 제어명령 (API 156 ZRN · API 158 DRVI · API 159 DRVA) 의 위치를 찾는 첫 번째 그룹 Y10 이 수행 될 때 현재 펄스 출력 값을 저장하는데 사용됩니다. 기록된 현재 펄스 출력 값 순방향 회전과 역방향 회전 때문에 증가할 것입니다.



API	의사기호		피연산자				기능		제어기						
159	D	DRVA	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>D<sub>1</sub></b>	<b>D<sub>2</sub></b>	절대 위치 제어		E S	E X	S	S A	S X	S C	E H

Type OP	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DRVA: 9 steps DDRVA: 17 steps	
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*		
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*		
D <sub>1</sub>		*															
D <sub>2</sub>		*	*	*													

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**피연산자:**

S<sub>1</sub>: 펄스의 번호(목표 장치) S<sub>2</sub>: 펄스 출력 주파수 D<sub>1</sub>: 펄스 출력 장치

D<sub>2</sub>: 회전 방향 신호

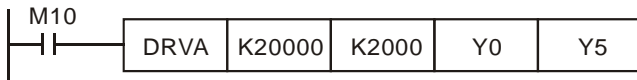
**설명:**

- SC 시리즈 MPU 에, 오직 32-bit DDRVI 명령만이 작업 가능하고, 피연산자 S<sub>1</sub>/S<sub>2</sub> 오직 K, H, 와 D 장치를 돕는다.SA/SX MPU 에, 이 명령은 도움이 되지 않습니다.
- S<sub>1</sub> 는 펄스 (상대 위치)의 번호로 지정됩니다. S<sub>1</sub> 의 가능한 번호는: EH 시리즈 MPU 의, 16-bit 명령: -32,768 에서 +32,767 과 32-bit 명령: -999,999 에서 +999,999 입니다. 양 (+)과 음(-)의 기호는 순방향과 역방향을 나타냅니다. SC MPU 에, 32-bit 명령: -2,147,483,648 ~ +2,147,483,647 입니다. 양(+) 과 음(-)기호는 순방향과 역방향을 나타냅니다.
- S<sub>2</sub> 는 펄스 출력 주파수로 지정됩니다. S<sub>2</sub> 의 사용가능한 번호는: EH 시리즈 MPU 에, 16-bit 명령: 10 에서 32,767Hz 와 32-bit 명령: 10 에서 200,000Hz 입니다. SC MPU 에서, 32-bit 명령: 100 ~ 100,000Hz 입니다.
- D<sub>1</sub> 은 펄스 출력 지시 장치로 지정됩니다. EH 시리즈 모델 에서, 그것은 오직 Y0,Y2 로 지정될 수 있습니다. SC MPU 에서, 그것은 Y10, Y11 로 지정될 수 있습니다.
- D<sub>2</sub> 는 회전 방향 신호로 지정되며, S 의 우선순에 따라 작동합니다. S 이 양(+)일 때 ,D<sub>2</sub> 는 On 입니다. S 가 음(-)일 때, D<sub>2</sub> 는 Off 입니다.
- EH 시리즈 MPU 에, 펄스의 번호는 CH0(Y0, Y1) 펄스의 현재값 레지스터(D1337 high byte, D1336 low byte)또는 CH1(Y2, Y3) 펄스의 현재 값 레지스터(D1339 high byte, D1338 low byte) 에 저장될 것입니다. 회전 방향이 음일 때, 현재 값 레지스터의 내용 값은 감소할 것입니다.
- SC 시리즈 MPU 에, 지시된 펄스 출력 번호 S<sub>1</sub> 은 CH0 (Y10) 펄스의 현재 값 레지스터 D1348 (low word) 와 D1349 (high word)또는 현재값 레지스터 D1350 (low word) and D1351 (high word)의 32-bit 명령 데이터 내용의 상대 위치에 저장될 것입니다.회전 방향이 음 일 때, 현재 값 레지스터의 값은 감소할 것입니다. 프로그램이 STOP 에서 RUN 으로 또는 RUN 에서 STOP 으로 일 때, 현재 값 레지스터의 값은 똑같이 남아있을 것입니다.
- EH 시리즈 MPU 에,DRVA 명령이 수행되는 동안 각 피연산자의 내용은 바뀔 수 없습니다. 내용은 다음 수행이 실행될 때 바뀔 것입니다.

9. EH 시리즈 MPU 에 DRVI 명령이 수행될 때 만약 드라이브 접촉이 꺼지면 EH 시리즈 MPU 를 위해 기계가 감속하고, 멈출 것이고, 수행이 완료된 플래그 M1029, M1030 does 는 켜지지 않습니다. SC MPU 에, 명령 수행 완료 후, 플래그 M1102 와 M1103 는 ON 일 것입니다..
10. EH 시리즈 MPU 에, EH MPU 에, DRVA 의 드라이브 접촉이 꺼지면, CH0 펄스의 표시 플래그 M1366 에 보내진 펄스 또는 CH1 펄스의 표시 플래그 M1377 이 보내진 펄스라 하더라도 DRVA 명령을 다시 수행하는건 불가능합니다.
11. EH 시리즈 MPU 에 EH 시리즈 MPU 에, DDRVI 를 위한 출력 주파수의 절대 값이 200kHz 이상일 때, 200kHz 은 우세합니다.또는 그것이 10Hz 보다 작을 때, 10Hz 이 우세합니다.
12. D1343 (D1353)는 CH0 (CH1) 펄스를 위한 첫 단계 가속의 시간을 정하는데 사용됩니다. 10,000ms, 10ms 은 출력에 우세 할 것입니다. 공장 설정: 100ms. SC MPU 에, 가속/감속의 사용가능한 범위는 50~20,000ms 입니다. 만약 사용가능한 범위 밖에 있습니다.면, 20,000ms 이상출력 에서 20,000ms 이 우세하고, 50ms 보다 작은 출력은 50ms 이 우세합니다.
13. D1340 (D1352) 은 Y10 (Y11)에 ON/OFF 의 주파수를 설정하기 위해 사용됩니다.만약  $S_2$ 에 의해 나타난 출력 주파수가 Y10 (Y11) 의 ON/OFF 주파수의 주파수와 같거나 더 작다면, ON/OFF 주파수는 펄스 출력 주파수로 수행될 것입니다.

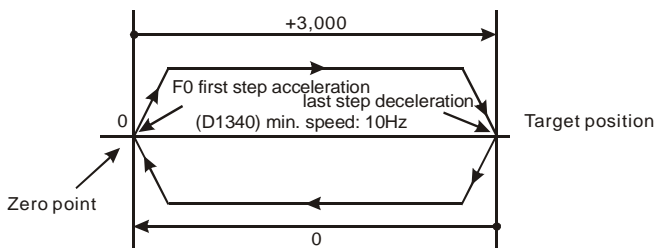
**프로그램 예 :**

M10= On 일 때, M10= On 일 때, Y0(상대 위치 찾기)로부터 2KHz 주파수 펄스의 twenty thousands (20000) 을 출력합니다. Y5= On 은 양 방향을 나타냅니다.

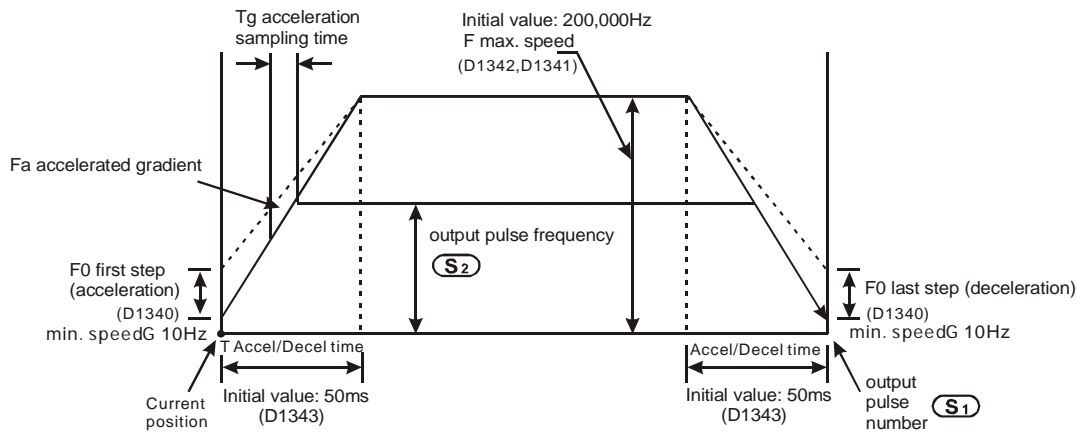


**주의:**

1. EH 시리즈 MPU 의 설명:
2. Ope 상대 위치 찾기 제어의 작동 설명: 현재 위치로부터 이동 거리를 지정하기 위한 양 또는 음의 부호 사용은 상대 위치 제어의 실행 방법 중 하나입니다.

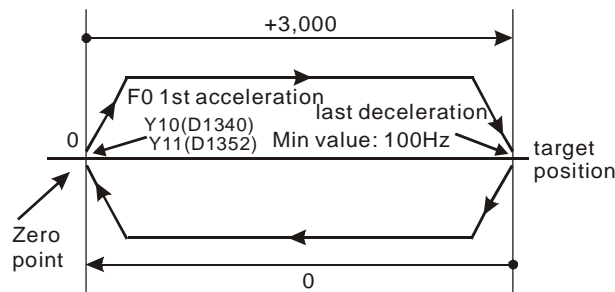


- a) 상대 위치찬기의 설정과 작동 속도:

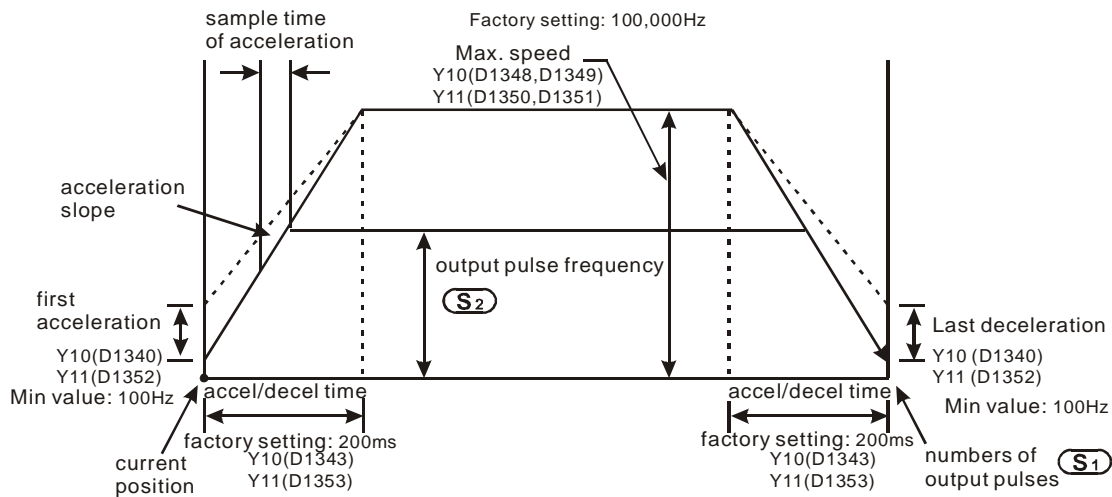


### 3. SC 시리즈 MPU 의 설명:

- a) 절대 위치 찾기 제어의 작동 설명: 절대 위치찾기 제어의 드라이브 방법이라고 도 불리는 영점으로 부터의 여행 거리로 지정하기 위한 방법.



- b) Th 상대 위치의 설정 아이템과 가속/ 감속 D1343 (D1353) 의 설정은 첫 단계 가속의 시간을 설정하는데 사용됩니다. D1340 (D1352) 은 Y10 (Y11)을 위해 마지막 단계 감속의 시간을 정하는데 사용됩니다. D1340 (D1352) 은 Y10 (Y11). 를 위한 처음과 끝 주파수를 설정하는데 사용됩니다.



- c) SC MPU 에, 많은 DRVI 명령이 프로그램안에서 동일하게 구성될 수 있습니다. 그러나 PLC 가 프로그램을 실행 할때마다 오직 하나의 명령이 실행 될 수 있습니다. . 예를 들면, 만약 Y10 출력이 명령에 의해 이미 실행 되었다면, Y10 출력을 실행시키는데 사용된 다른 명령어는 실행되지 않을 것입니다. 그래서 명령수행 우선권이 명령 실행의 순서에 기반합니다.

- d) DDRVI 명령에 의해 수행될 때, Y10의 원래의 기능은 DDRVI 이 OFF 될때까지 가능하지 않을 것입니다. Y11 역시 그럴것입니다. ,
- e) DDRVI 가 수행되면, 모든 다른 매개변수들이 DDRVI 이 OFF 가 될때까지 변경될 수 없습니다.
- f) 이 명령이 꺼지고, 출력이 완료되지 않았을 때:

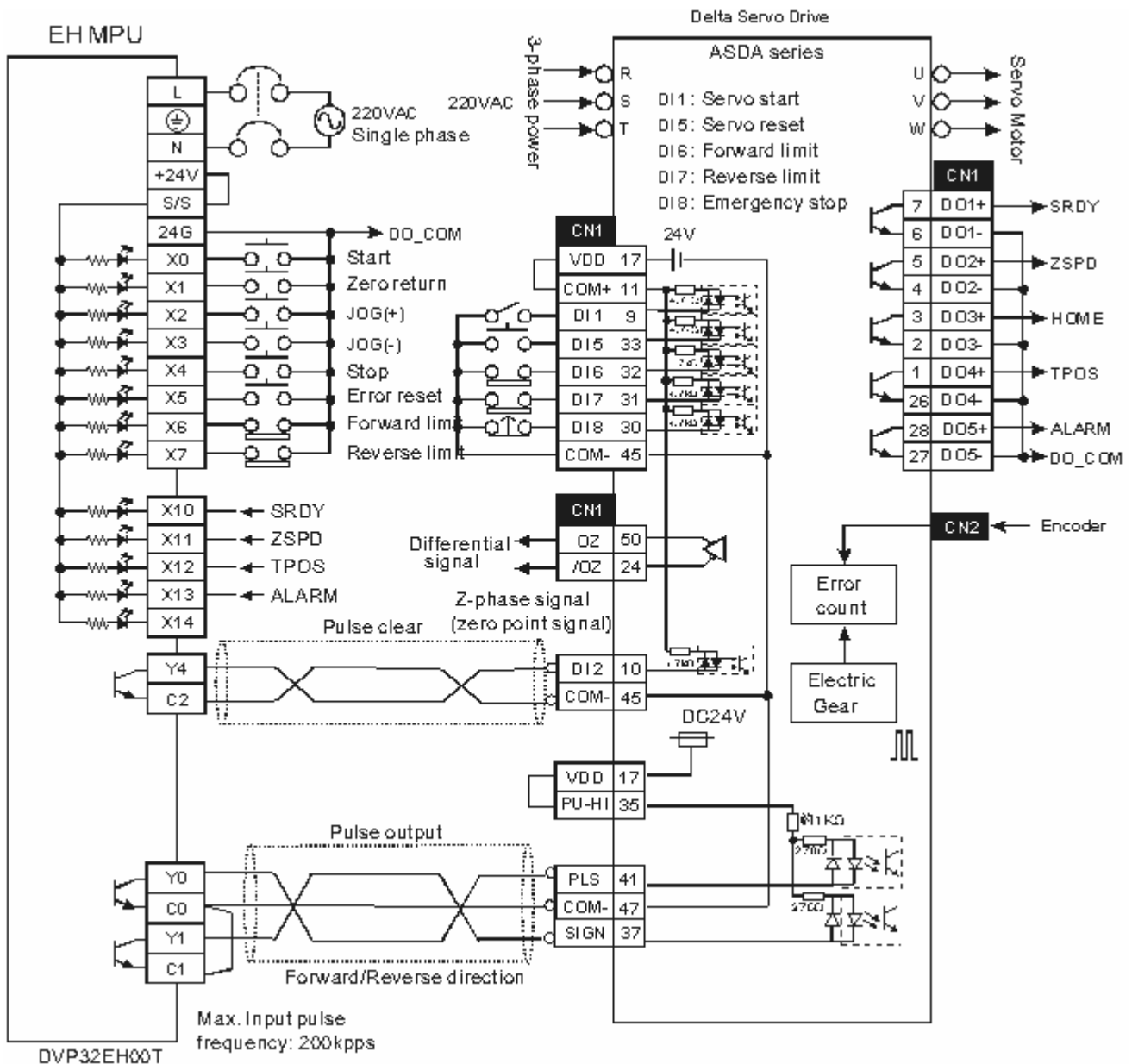
만약 M1334=ON 이면, Y10은 출력을 즉시 멈출 것입니다.

만약 M1334=OFF 이면, Y10은 감속 시간으로부터 끝의 주파수에 까지 감속될 것이고 펄스 출력을 멈출 것입니다.

Y11 출력에 일치하는 M1335와 같은 방법입니다.

플래그 설명을 위하여 DDRVI 메모를 참고하십시오..

#### 4. DVP-EH 시리즈의 배선과 Delta ASDA 서보 드라이브



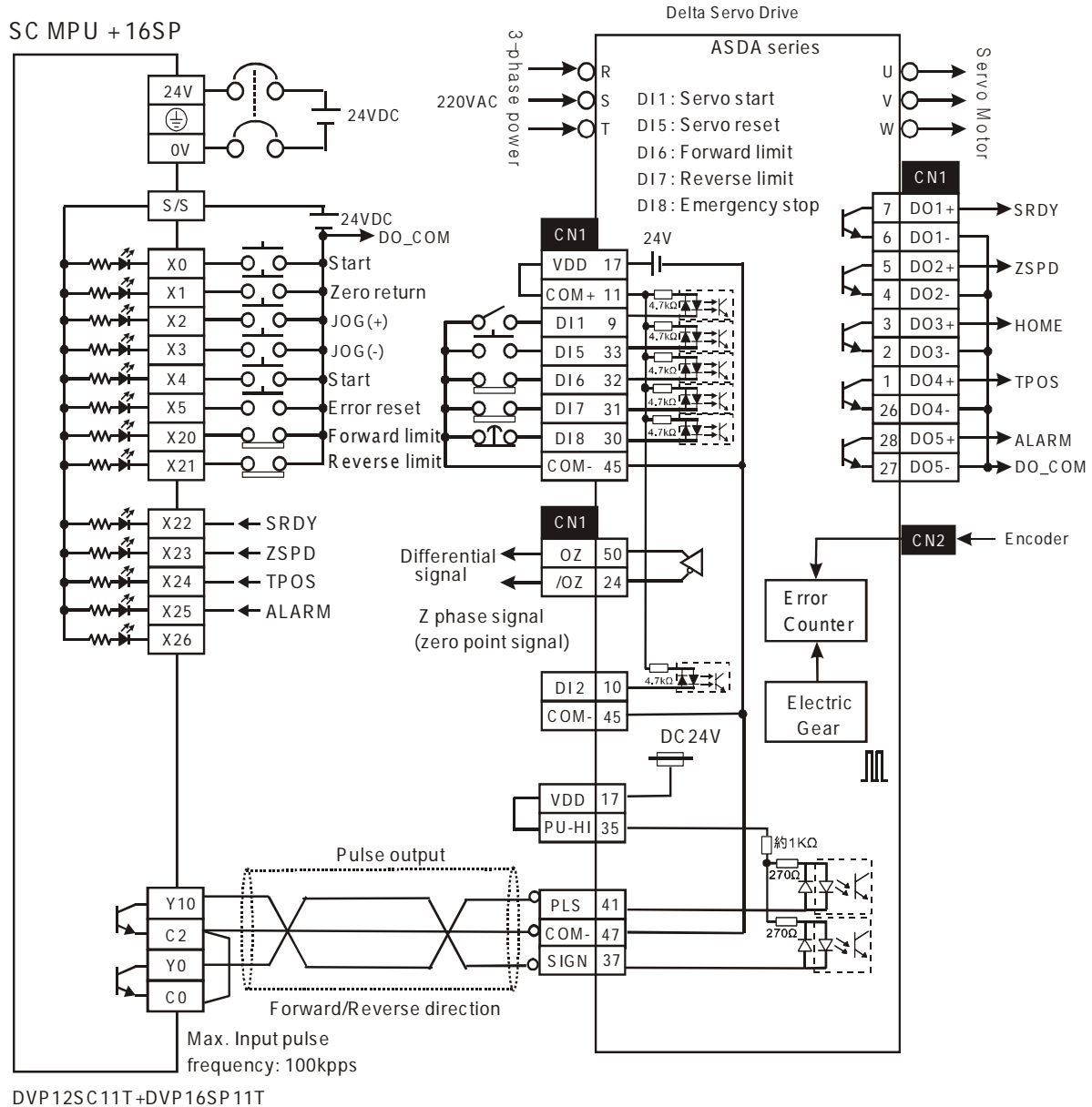
참고:

Delta ASDA 서보 드라이브의 매개변수 설정: P1-01 는 위치 모드로 설정되었고, P1-00 펄스 입력 형태는 펄스+DIR 로 설정 되었다.

2. 순방향/역방향의 제한 스위치는 **SERVO AMP** 에 연결되도록 되어있습니다.

3. 펄스 지움 신호는 현재 보내는 과정이 완료 되지 않은 서보의 현재 펄스를 지울 것입니다.

## 5. DVP-SC 시리즈의 배선 과 Delta ASDA 서보 드라이브



참고:

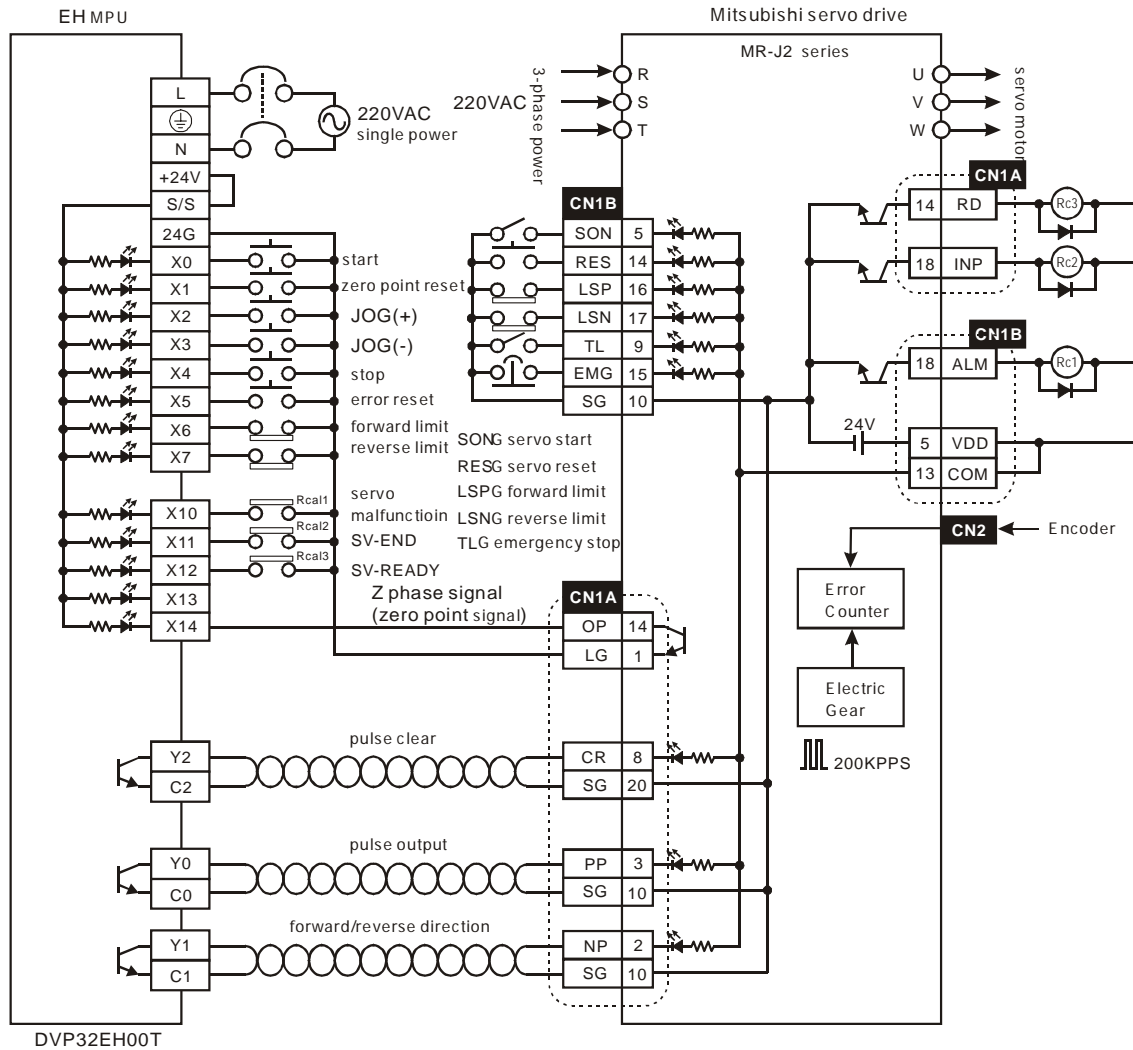
1. Delta ASDA 서보 드라이브의 매개변수 설정:

P1-01: 위치 모드

P1-00: 펄스+DIR 로의 펄스 입력형태

2. SERVO AMP 에 순방향/ 역방향 제한 스위치 연결.

## 6. DVP-EH 시리즈 PLC 와 Mitsubishi MR-J2-□A Servo drive 간의 연결의 배선 예제:



Note:

- DABS 명령을 사용하여 절대 위치를 찾을 때, the to Delta EH 시리즈 PLC에 연결 되는 Mitsubishi MR-J2-□A servo drive 의 매개변수 설정:  
P0: 위치 모드.  
P1: 절대 값 사용의 시스템.  
P21: 펄스+DIR 로서의 펄스 출력 형태
- SERVO AMP 에 연결된 순방향/역방향 한계 스위치.
- 서보의 OP (Z phase signal)을 사용할 때, 만약 Z 위상신호는 고속 모터 작동 동안 높은 주파수의 신호입니다. 높은 주파수 신호는 PLC 에 의해 조사될 수 있는 사용 가능한 범위 안에 있을 수 있습니다.

7. 위치 제어 프로그램을 지시할 때 주의 사항:

- 제어 명령 API 156 ZRN, API 157 PLSV, API 158 DRVI, API 159 DRVA 을 위한 제한 시간을 사용하지 않습니다. 사용자는 이들 명령어를 프로그램에서 여러 번 사용 가능하지만 다음 아래 주의를 따라야 합니다.
  - 동시에 같은 출력 CH0(Y0, Y1) 또는 CH1(Y2, Y3) 을 사용하는 위치 제어 명령을 실행하지 말라. 그렇지 않으면 그것들은 이중코일로 취급되고 올바른 기능을 할 수 없을 것입니다.

ii. 제어 프로그램을 설계하기 위해 사다리식 명령 단계를 사용하는 것이 좋겠습니다. (아래 보이는 프로그래밍 예제를 보시오)

b) 펄스 출력명령 API 57 PLSY, API 58 PWM, API 59 PLSR 으로 위치 제어 명령 API 156 ZRN, API 157 PLSV, API 158 DRVI, API 159 DRVA 을 사용할 때 참고하십시오.

CH0 펄스의 현재 값 레지스터(D1337 high byte, D1336 low byte) 또는 CH1의 현재 값 레지스터(D1339 high byte, D1338 low byte) 모두 위치 제어 명령과 펄스 출력 명령에 사용될 것이고, 복잡한 연산안에 결과가 나타날 것입니다.

c) CH0 펄스 펄스 출력 단말기 Y0, Y1 와 CH1 펄스의 Y2, Y3 설명:

전압 범위: DC5V to DC24V

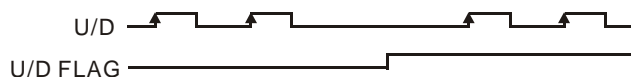
전력 범위 10 mA to 100 mA

출력 펄스 주파수: Y0, Y2 는 200KHz 이고, Y1, Y3 는 10KHz 입니다.

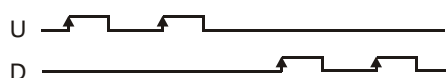
8. 위치 지정 연산의 펄스 출력 신호 설정 output signal settings of positioning operation:

a) DVP-EH 시리즈 PLC 에 3 가지의 위치 지정 연산의 펄스 출력 신호가 있습니다.:

위상 1 출력+ 방향 (이방법 사용을 추천합니다.)



1-위상 2 출력 (주파수 한계는 10KHz)



2-위상 2 출력 (주파수 한계는 10KHz)



9. SERVO AMP 또는 스테핑 모터에 매개변수의 펄스 입력 형태를 지정하기에 앞서 PLC 출력 지정을 따르시오.

10. EH 시리즈 MPU 플래그 설명:

M1010: EH 시리즈 MPU 에서, M1010= On 일 때, CH0 (Y0, Y1) 와 CH1 (Y2, Y3) 는 END 명령이 수행될 수행되는 동안 출력 할 것입니다. 출력을 시작하면 M1010 은 자동으로 꺼질 것입니다.

M1029: EH 시리즈 MPU 에서, 첫 번째 그룹 펄스 CH0 (Y0, Y1)가 펄스출력 완료 또는 다른 관계 있는 명령이 수행을 마친 후, M1029= On 입니다.

M1030: EH 시리즈 MPU 에서, 두 번째 그룹 펄스 CH1 (Y2, Y3)이 펄스 출력 완료한 후 M1030= On

M1258 : EH 시리즈 MPU 에서, M1258=ON 일 때 역 방향 에서 CH0 (Y0, Y1) 펄스 출력 설정

M1259 : EH 시리즈 MPU 에서, M1259=ON 일 때 역 방향에서 CH1 (Y2, Y3) 펄스 출력 설정

M1305 : EH MPU 에서, 플래그 for of commands PLSV, DPLSV, DRVI, DDRVI, DRVA, 와 DDRV 명령의 CH0 (Y0, Y1)를 위해 역 플래그를 수행

M1306 : EH MPU 에서, 플래그 for CH1 (Y2, Y3) of commands PLSV, DPLSV, DRVI, DDRVI, DRVA, 와 DDRV 명령의 CH1 (Y2, Y3)을 위해 역 플래그를 수행.

M1334 : EH 시리즈 MPU 에서, M1334= On 일 때, CH0 (Y0, Y1) 펄스 출력을 중단합니다.

SC 시리즈 MPU 에서 M1334=ON 으로, DDRVI 와 DDRVA 명령이 수행을 중지 할 때. CH0 (Y10) 펄스 출력은 즉시 감속 없이 중지 할 것입니다.

M1335 : In EH 시리즈 MPU, CH1 (Y2, Y3) 펄스 stop output when M1335= On.

SC 시리즈 MPU 에서 M1335=ON 으로 , DDRVI 와 DDRVA 명령은 수행을 중지 합니다. CH1 (Y11) 펄스 출력은 곧 감속 없이 중단할 것입니다.

M1336: EH 시리즈 MPU 에서, CH0 (Y0, Y1) 펄스 출력 표시플래그

M1337: EH 시리즈 MPU 에서, CH1 (Y2, Y3) 펄스 출력 표시 플래그

M1346: EH 시리즈 MPU 에서, ZRN 명령 출력 신호는 플래그를 작동시킵니다.

#### 11. EH 시리즈 MPU 의 특수 레지스터 설명:

D1220: 첫 번째 그룹 Y0, Y1 의 위상 설정: D1220 의 마지막 두 비트 에 의해 결정, 다른 비트들은 효력이 없습니다.

1. K0: Y0 출력
2. K1: Y0, Y1 AB 위상출력, A 가 B 를 이끈다
3. K2: Y0, Y1 AB 위상출력, B 가 A 를 이끈다
4. K3: Y1 출력

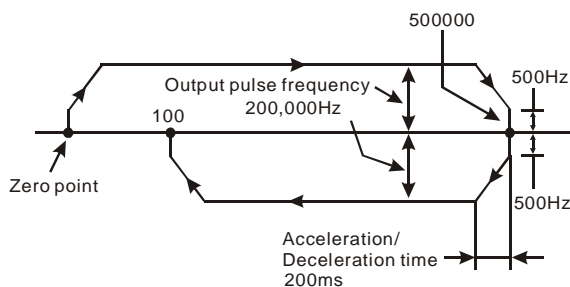
D1221: 첫 번째 그룹 Y2, Y3 의 위상 설정: D1221 의 마지막 두 비트 에 의해 결정, 다른 비트들은 효력이 없습니다.

1. K0: Y2 출력
2. K1: Y2, Y3 AB 위상출력, A 가 B 를 이끈다.
3. K2: Y2, Y3 AB 위상출력, B 가 A 를 이끈다.
4. K3: Y3 출력

12. EH MPU, 몇몇의 고속 펄스 출력 명령(PLSY, PWM, PLSR)과 위치 제어 명령 (ZRN, PLSV, DRVI, DRVA) 모두 한 프로그램안에 그리고 동시에 같은 스캔 주기에 실행 되어진 펄스를 출력하는데 Y0 를 사용할 때, PLC 는 가장 적은 단계 번호를 가진 명령을 수행할 것입니다.

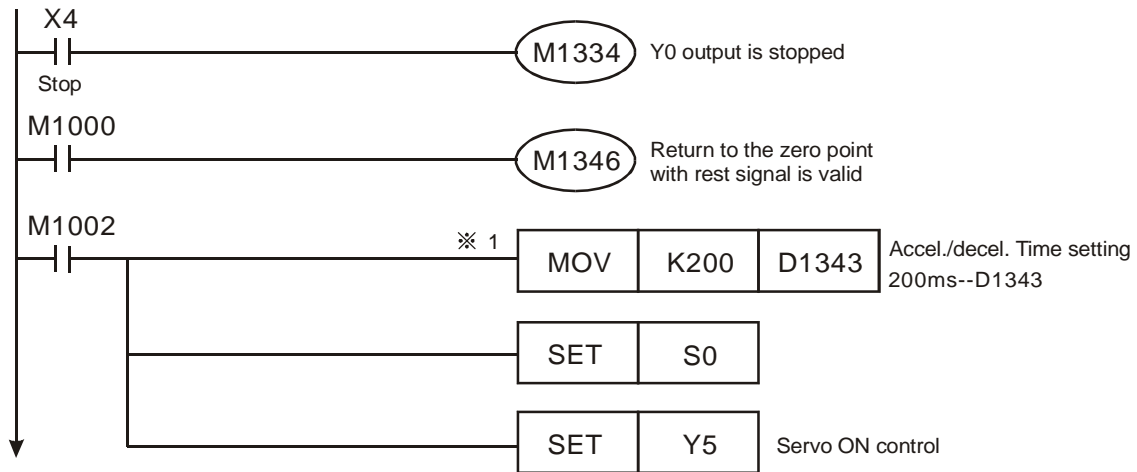
#### 정방향/역방향 연산을 위한 프로그래밍 예:

1. 배선을 위해, DVP-EH 시리즈 PLC 와 Mitsubishi MR-J2-□A Servo 드라이브 간 연결의 배선 예를 참고 하십시오..
2. 하나의 연산 위치 지정은 아래 보이는 절대 위치 를 이용함으로써 수행됩니다.

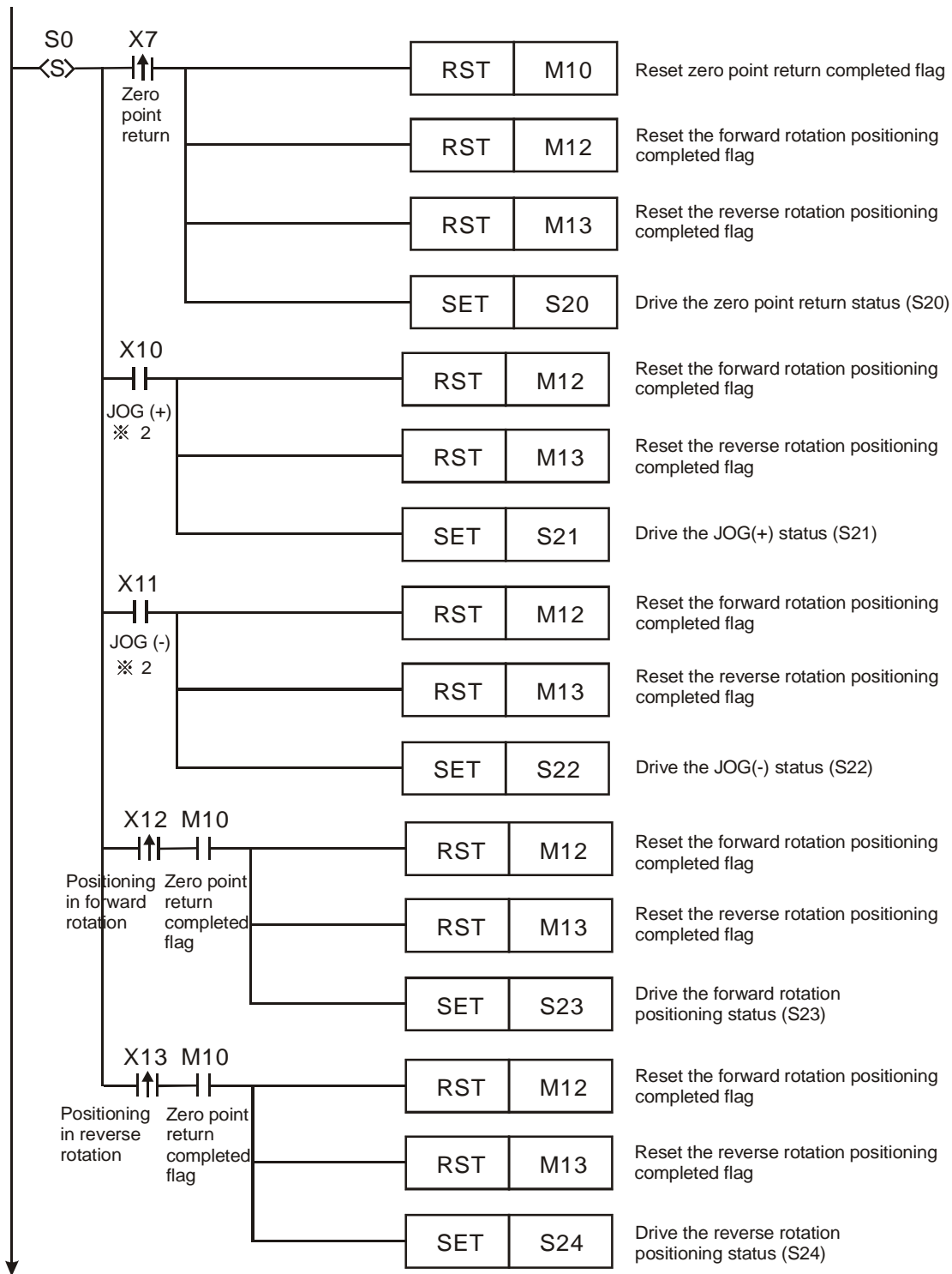


3. 다음은 사다리형 단계 도표의 프로그램 예입니다.(STL):

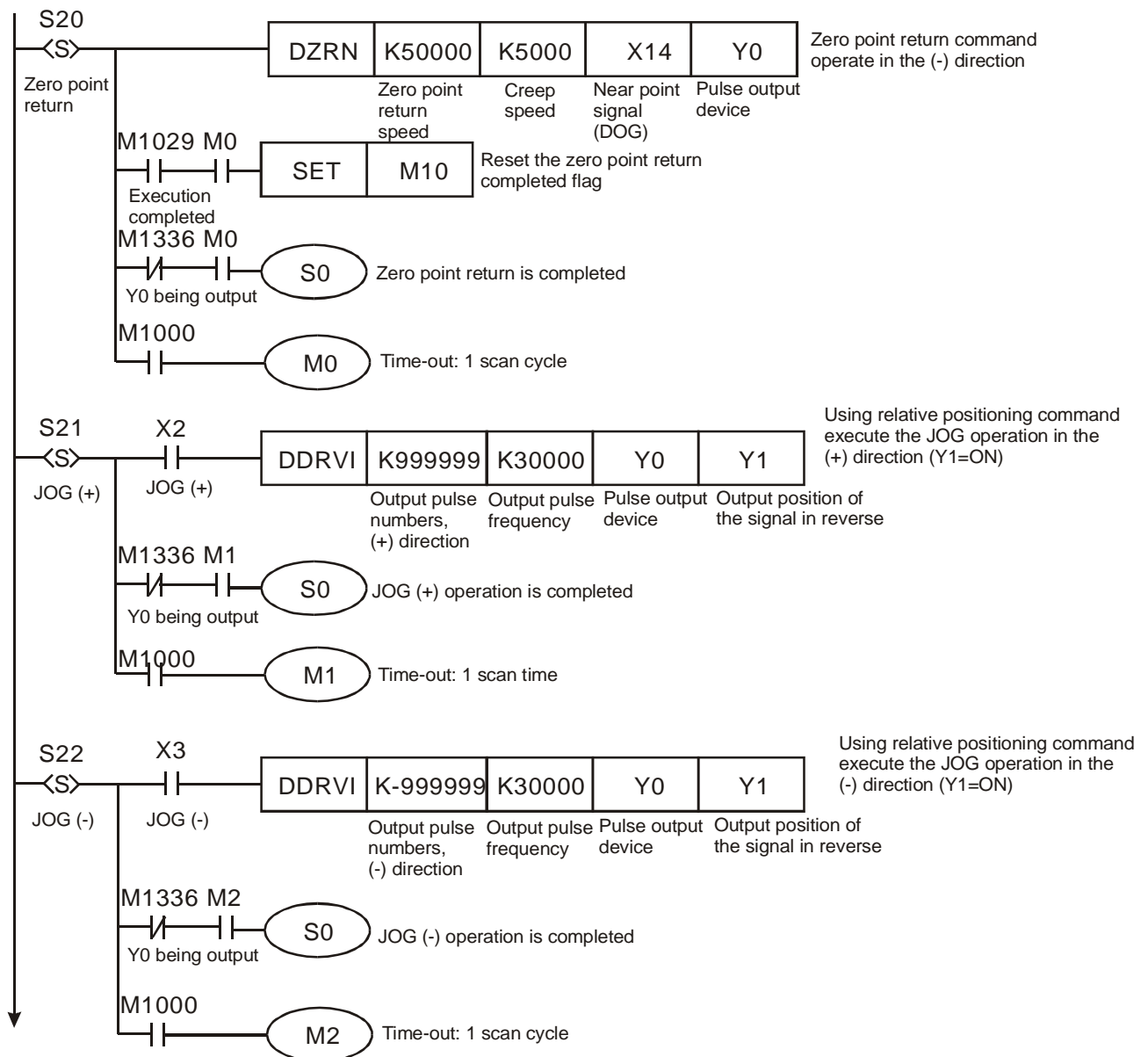


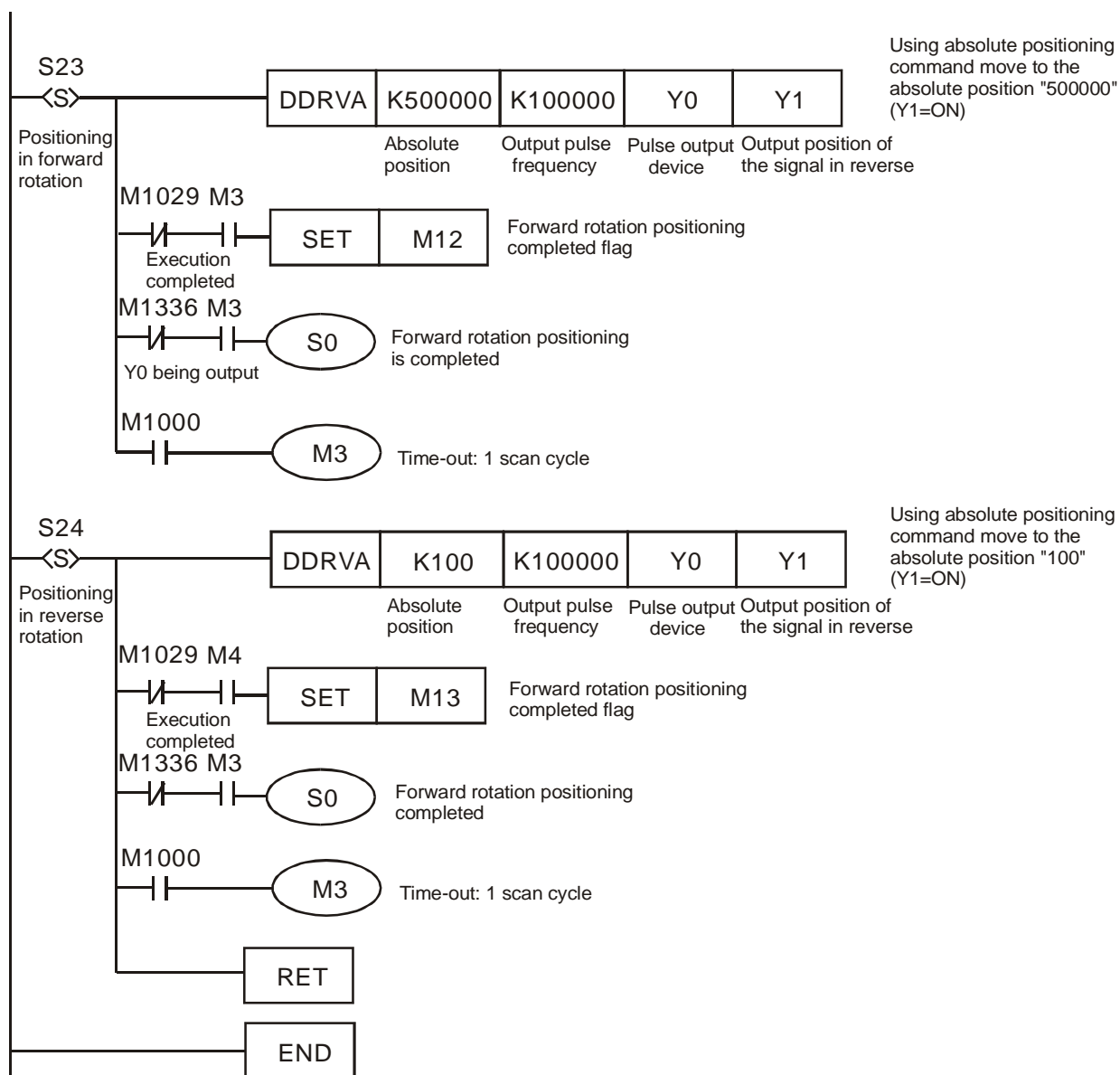


※1. 만약 CH1의 가속/감속 시간(D1343)이 공장 설정에 의해 설정될 수 있습니다.면, 위의 프로그램은 무시될수 있을 것입니다. 공장 설정: 100ms.



※2. 2.최고 JOG 수행의 거리는 최고 상태와 동일합니다. API 158 DDRVI 명령으로부터 오는 외부자극의 수. 만약 더큰 거리가 요구됩니다.면 JOG 명령을 재가동 시키시오.





API	의사기호	피연산자	기능	제어기
160	TCMP P	<b>(S<sub>1</sub>) (S<sub>2</sub>) (S<sub>3</sub>) (S) (D)</b>	시간 비교	ES EX SS SA SX SC EH

Type	비트 장치				워드 장치											프로그램 단계
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	TCMP, TCMPP: 11 steps
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*	
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*	
S <sub>3</sub>					*	*	*	*	*	*	*	*	*	*	*	
S											*	*	*			
D		*	*	*												

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**피연산자:**

**S<sub>1</sub>:** 비교시간의 Hour, 적용범위는 K0~K23 **S<sub>2</sub>:** 비교시간의 Minute, 적용범위는 K0~K59

**S<sub>3</sub>:** 비교시간의 Second, 적용범위는 K0~K59 **S:** 달력의 현재시간(연속적인 3 개의 장치를 차지합니다.)

**D:** 비교 결과 (연속적인 3 개의 장치를 차지합니다.)

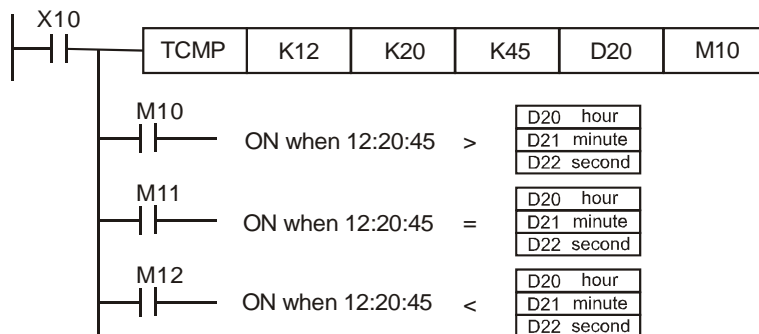
피연산자의 범위 S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>: S<sub>1</sub>=0~23, S<sub>2</sub>=S<sub>3</sub>=K0~59

**설명:**

- S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>** 는 헤드주소 S 의 현재 값과 비교됩니다. 그리고 비교된 결과는 D 에 저장됩니다.
- S<sub>1</sub>** 은 현재시간의 시이며 내용은 K0~K23 입니다. **S<sub>2</sub>** 는 현재시간의 분이며 내용은 K0~K59 입니다. **S<sub>3</sub>** 는 현재시간의 초이며 내용은 K0~K59 입니다.
- S 에 의해 구체화된 실제시계의 현재 시간은 이전의 TRD 명령을 이용함으로써 인식되고 TZCP 명령을 이용함으로써 비교됩니다. 만약 S 의 내용 범위가 초과되면 작동 오류의 결과를 초래합니다. 동시에 프로그램은 수행되지 못하고 M1067 과 M1068 이 켜지며 D1067 에 오류 코드 0E1A (HEX)가 D1067 에 기록 됩니다.

**프로그램 예 :**

- X10 이 켜지면 명령은 수행되고 (D20~D22)에 있는 실제 시계의 현재 시간은 12:20:45 의 값과 비교되어지며, 그 결과는 M10~M12 에 나타난다. X10 이 켜진 상태에서 꺼지면 명령은 수행되지 않지만 꺼지기 전의 M10~M12 의 상태는 지속됩니다.
- M10~M12 연결이나 병렬상태는 ≥, ≤, ≠ 의 결과가 주어진다.



API	의사기호	피연산자	기능	제어기
161	TZCP	P (S <sub>1</sub> ) (S <sub>2</sub> ) (S) (D)	달력 존 비교	ES EX SS SA SX SC EH

Type	비트 장치				워드 장치										프로그램 단계	
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	TZCP, TZCPP: 9 steps
S <sub>1</sub>											*	*	*			
S <sub>2</sub>											*	*	*			
S											*	*	*			
D		*	*	*												

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

**피연산자:**

**S<sub>1</sub>** : 최저한계 시간 데이터    **S<sub>2</sub>**: 최고한계 시간 데이터    **S**: 달력의 현재 시간

**D**: 비교 결과 (3 개의 연속적인 장치를 차지합니다.)

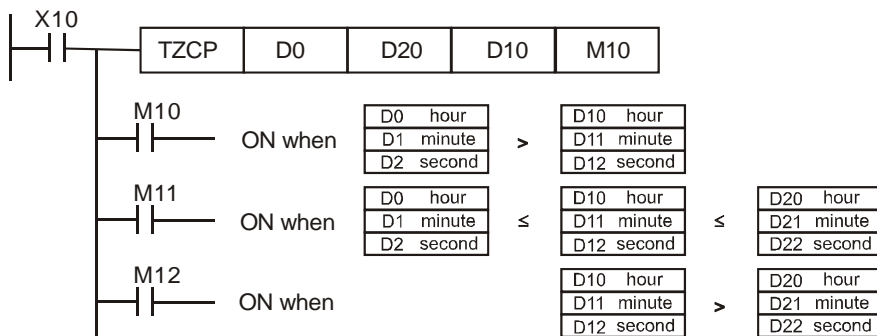
피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, **S** 3 개의 연속적인 장치를 차지합니다.    **S<sub>1</sub>** 은 **S<sub>2</sub>** 보다 작아야 합니다. . **S<sub>1</sub> ≤ S<sub>2</sub>**

**설명:**

- S** 는 S<sub>1</sub>~S<sub>2</sub> 의 시간 주기 에 비교되고 비교 결과는 D 안에 저장됩니다.
- S<sub>1</sub>** , **S<sub>1</sub> +1** , **S<sub>1</sub> +2**: 각각은 시간, 분 , 초의 최저한계 시간데이터를 나타냅니다.
- S<sub>2</sub>**, **S<sub>2</sub> +1** , **S<sub>2</sub> +2**: 각 각은 시간, 분, 초의 최고한계 시간 데이터를 나타냅니다. .
- S** , **S +1** , **S +2**: 각 각은 시간, 분, 초의 만세력의 현재 시간을 나타냅니다.
- S** 에 의해 구체화된 실제시계의 현재 시간은 이전의 TRD 명령을 이용함으로써 인식되고 TZCP 명령을 이용함으로써 비교됩니다. 만약 S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub> 의 범위가 초과되면 작동오류의 결과를 초래하고 동시에 프로그램은 수행되지 못하고 M1067 과 M1068 이 켜지며 D1067 에 오류 코드 0E1A (HEX)가 D1067 에 기록 됩니다.
- 만약 **S < S<sub>1</sub>** 이면, **D** 가 켜진다. 만약 **S > S<sub>2</sub>** 이면, **D +2** 가 켜진다. 이 둘의 경우엔 **D +1** 가 켜진다. (하한선 **S<sub>1</sub>** 은 상한선 **S<sub>2</sub>**.보다 작아야 합니다.)

**프로그램 예 :**

X10 이 켜지면 명령이 수행되고 M10~M12 중의 하나가 켜진다. X10 이 꺼지면, 명령은 수행되지 않지만 X10 이 꺼지기 전 M10~M12 의 상태는 지속됩니다.



API	의사기호			피연산자			기능					제어기							
162		TADD	P	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>D</b>	달력 데이터 추가					ES	EX	SS	SA	SX	SC	EH	

Type OP	비트 장치				워드 장치											프로그램 단계 TADD, TADDP: 7 steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
S <sub>1</sub>											*	*	*					
S <sub>2</sub>											*	*	*					
D											*	*	*					

필스								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

**S<sub>1</sub>**: 피가수된 시간    **S<sub>2</sub>**: 가수된 시간    **D**: 가수된 시간 결과

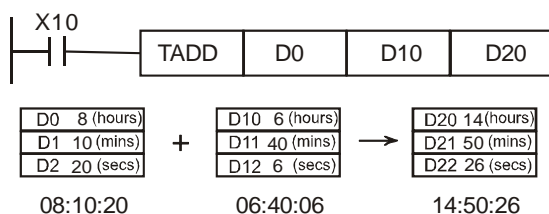
**S<sub>1</sub>**, **S<sub>2</sub>**, **D** 연속적인 3 개의 장치를 차지합니다.

설명:

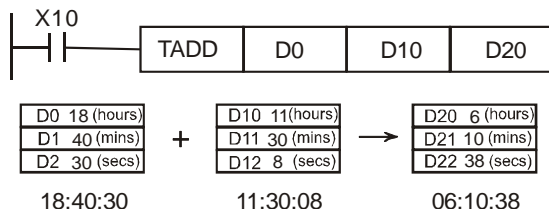
- S<sub>1</sub> + S<sub>2</sub> = D. S<sub>1</sub> + S<sub>2</sub> = D.** S<sub>1</sub>에 의해 구체화된 레지스터안의 시간 데이터는 S<sub>2</sub>에 의해 구체화된 레지스터안의 시간데이터에 더해지고 그결과는 D에 의해 구체화된 레지스터안에 저장됩니다.
- S<sub>1</sub>, S<sub>2</sub> 안에 있는 시간 데이터가 범위를 초과하면 작동오류를 초래합니다. 그러면 프로그램은 수행되지 못하고 M1067 과 M1068 이 켜지며 D1067 에 오류 코드 0E1A (HEX)가 기록 됩니다.
- 만약 덧셈 결과가 24 시간 값보다 커지면 올림 표시 문자 M1022 가 켜지고 , D 에 나타나는 결과의 값은 24 시를 넘는 시간으로 남아있습니다.
- 만약 덧셈 결과가 0 과 같다면 0 표시문자 M1020 이 켜진다.

프로그램 예 :

- X10 이 켜지면 명령이 수행되고 .D0~D2 에 의해 구체화된 더해진 시간 데이터와 D10~D12 는 D20~D22 에 의해 구체화된 레지스터의 결과에 저장됩니다.



- 만약 덧셈 결과가 24 시보다 크다면 올림 표시문자 M1022 가 켜진다.



API	의사기호			피연산자			기능			제어기							
163		TSUB	P	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>D</b>	탈력 기능 감수			ES	EX	SS	SA	SX	SC	EH	

Type OP	비트 장치				워드 장치											프로그램 단계 TSUB, TSUBP: 7 steps	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S <sub>1</sub>											*	*	*				
S <sub>2</sub>											*	*	*				
D											*	*	*				

펄스								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

**S<sub>1</sub>**: 피 감수된 시간    **S<sub>2</sub>**: 감수된 시간    **D**:감수된 시간 결과

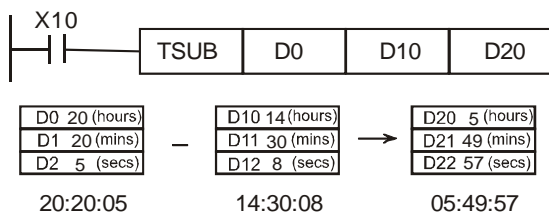
**S<sub>1</sub>, S<sub>2</sub>, D** 연속적인 3 개의 장치를 차지합니다.

설명:

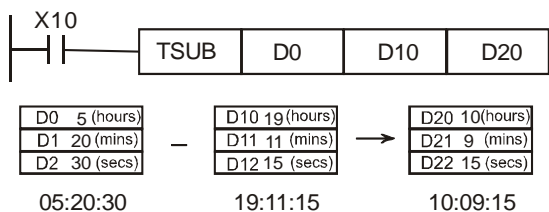
- S<sub>1</sub> - S<sub>2</sub> = D.** S<sub>2</sub>에 의해 구체화된 레지스터안의 시간 데이터는 S<sub>1</sub>에 의해 구체화된 레지스터 안의 시간 데이터를 빼고 그 결과는 D에 의해 구체화된 레지스터안에 저장됩니다.
- S<sub>1</sub>, S<sub>2</sub> 안에 있는 시간 데이터가 범위를 초과하면 작동오류를 초래합니다. 그러면 프로그램은 수행되지 못하고 M1067 과 M1068 이 켜지며 D1067 에 오류 코드 0E1A (HEX)가 기록 됩니다.
- 만약 감수 결과가 0 보다 작으면, 0 표시 문자 M1020 이 켜진다. D 에 나타나는 결과의 값은 0 시간 이하로 남아있습니다.
- 만약 감수 결과가 0 이면, 0 표시문자 M1020 이 켜진다.
- API 166 TRD 명령을 제외하고, MOV 명령은 특정한 레지스터 D1315(시), D1314(분), D1313(초)를 실제 시계에 현재 시간을 읽을 수 있도록 이동시켜 레지스터를 구체화하는데 쓰일 수 있습니다.

프로그램 예 :

- X10 이 켜지면 명령이 수행됩니다. D10~D12 에 의해 구체화된 시간 데이터는 D0~D2 에 의해 구체화된 시간 데이터에의 해 감수 되고 그 결과는 D20~D22 에 의해 구체화된 레지스터에 저장됩니다.



- 만약 감수 결과가 0 보다 작으면 발림표시문자 M1021 이 켜진다.





API	의사기호			피연산자	기능	제어기						
166		TRD	P	<b>D</b>	달력 데이터 판독	ES	EX	SS	SA	SX	SC	EH

Type OP	비트 장치				워드 장치												프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
D												*	*	*			TRD, TRDP: 3 steps

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**D:** 이 장치는 달력의 현재시간을 저장합니다. (연속적인 7 개의 장치를 차지합니다.)

설명:

- 만세력 시계는 EH/SA 시리즈 PLC 에 내장되어 있고, 이 시계는 연도, 주, 월, 일, 시, 분, 초 7 개의 데이터 장치를 D1319~D1313 에 저장되어 있습니다. TRD 명령의 기능은 프로그램 디자이너가 만세력의 시간을 바로 읽을 수 있게 하고 D 에 의해 구체화되어있는 7 개 데이터 레지스터안에 리딩 데이터를 저장합니다.
- D1319 는 두개의 디지털 숫자로 읽어지고 이 적용은 4 개의 디지털 숫자로 바뀔 수 있습니다. 자세한건 API 167 TWR 명령에 관한 각주를 참고 할 수 있습니다.

프로그램 예 :

- X0 가 켜지면, 만세력이 현재시각을 읽으므로써 D0~D6 레지스터를 구체화 시킵니다.
- D1318 의 구성은 1 은 월요일, 2 는 화요일, ..., 7 은 일요일을 의미합니다.



Special D device	Meaning	Content		General D device	Meaning
D1319	Year (A.D.)	00~99	→	D0	Year (A.D.)
D1318	Day (Mon.~Sun.)	1~7	→	D1	Day (Mon.~Sun.)
D1317	Month	1~12	→	D2	Month
D1316	Date	1~31	→	D3	Date
D1315	Hours	0~23	→	D4	Hours
D1314	Minutes	0~59	→	D5	Minutes
D1313	Seconds	0~59	→	D6	Seconds

주의:

- Error 플래그 of the real time clock built in EH/SA 시리즈 PLC:

Device	Name	기능
M1016	만세력의 연도 표시	D1319 가 꺼졌을 때 2 개의 디지털 숫자로 년도를 표시합니다.
M1017	±30 초 오차 조절	전원이 꺼진 후 다시 켜졌을 때 (만약 초단위가 0~29 사이면, 0 초에서 다시 시작합니다. 만약 초단위가 30~59 사이면 1 분을 더하고 0 에서 시작합니다.)
M1076	만세력의 고장	이것은 배터리가 나가거나 적용범위를 벗어난 세팅을 한 경우에 켜진다.(전원이 켜진 경우에만 작동합니다.)

Device	Name	Range
D1313	초	0-59
D1314	분	0-59
D1315	시	0-23
D1316	일	1-31
D1317	월	1-12
D1318	주	1-7
D1319	연도	0-99 (two right-most digit number of year)

2. 만세력 고치는 방법:

3. 내장된 API 만세력을 고치는 방법은 두가지가 있습니다.:

a) 구체적인 명령

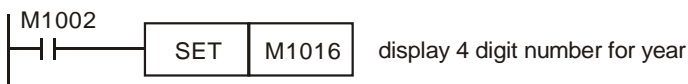
명령어 TWR (API 167)를 참고하십시오.

b) 주변장치의 조정

조정시 WPLSoft 이용 (사다리식 도표를 편집하는 소프트웨어

4. 연도와 관련한 디지털 네자리수의 표시:

a) 그것은 보통 년도와 관련된 2 자리수를 표시합니다. 만약 네자리 숫자를 표현하고 싶다면 다음에 나오는 프로그램을 시스템 초기에 적용시켜야 합니다.



b) 2 자리 년도 표시 숫자는 네자리 숫자로 교체됩니다.

만약 다음번에 네자리표시를 원합니다.면, 2000-2099 사이에서 끝에 두 자리 00-99 사이의 숫자만 교체할 수 있습니다., 예를 들어 00=2000 년 50= 2050, 99=2099 로 표시됩니다.

API	의사기호			피연산자	기능							제어기													
167		TWR	P	<b>D</b>	달력 데이터 기록							ES	EX	SS	SA	SX	SC	EH							
Type OP	비트 장치				워드 장치											프로그램 단계									
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	TWR, TWRP: 3 steps									
											*	*	*												
					펄스							16-bit				32-bit									
					ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

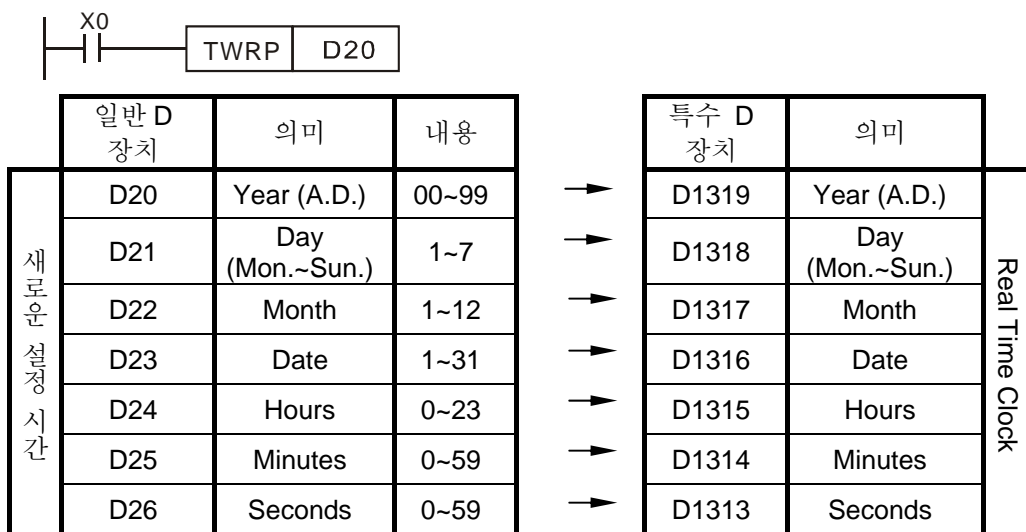
**D:** 새로운 달력계의 자원 (연속적인 7 개의 장치를 차지합니다).

설명:

- S:** 새로운 만세력 시간을 저장하는 장치
- 만세력 시계는 EH/SA 시리즈 PLC 에 장착됩니다. 이 명령은 만세력을 조정할 때 현재의 정확한 시간을 만세력에 기재할 때 쓰 일 수 있습니다.
- 이 명령을 실행 시에는 , 새로 세팅 된 시간은 만세력 시계 외부에 바로 표시됩니다. 그러므로 이 명령을 실행할 경우에 현재 시간이 맞다면 새로 세팅 된 시간이라는 것을 명심해라 합니다.
- 만약에 시간 데이터 S 가 범위를 초과합니다.면, “실행 에러”를 초래합니다. 이 때에는 명령이 수행되지 않으며 M1067=On, M1068=On, 에러코드 0E1A (HEX) in D1067 가 나타난다.

프로그램 예 1:

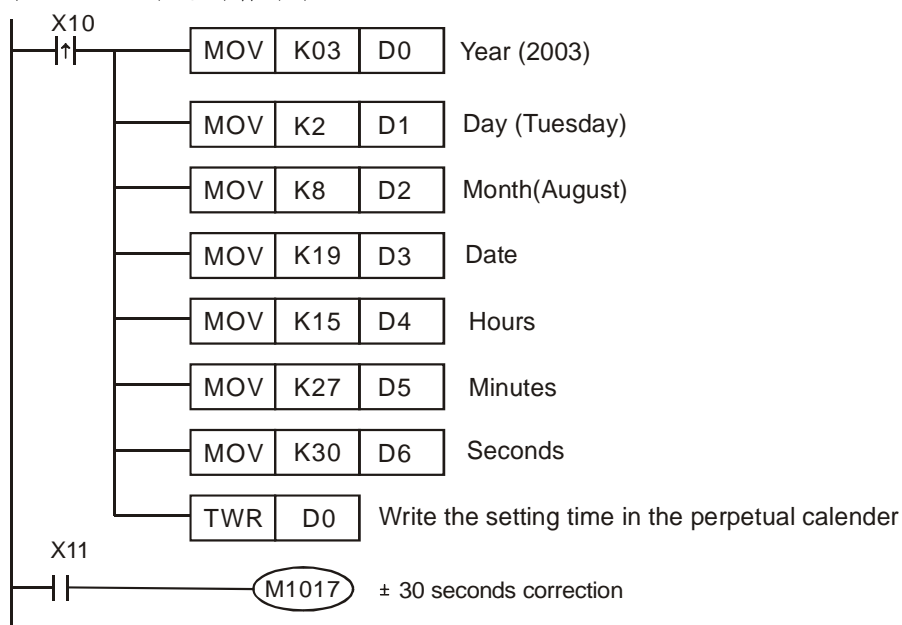
X0 가 켜지면, 현재 정확한 시간이 만세력에 표시됩니다.



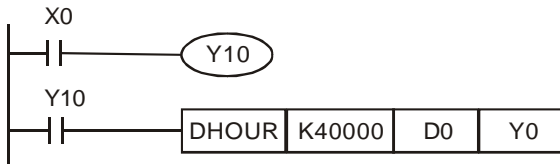
프로그램 예 2:

- 2002/03/23, Tuesday, 15:27:30 으로 현재 시간을 맞추고 조정합니다. (다음의 프로그램 예를 참고로 하십시오.).
- D0~D6 의 내용은 만세력의 새로운 시간 세팅에 관여합니다.
- X10 이 켜지면, 만세력의 현재 시간 조정을 바꿀 수 있습니다.
- X11 이 켜지는 모든 시간에는, 만세력 시계  $\pm 30$  초 단위로 시간 조정을 수행합니다.즉 만약 만세력 시계의 초가 1~29 사이에 위치합니다.면 시간은 자동적으로 0 초로 계산되고 시간은 변하지 않습니다.

그러나 만약 만세력 시계의 초가 30~59 사이에 위치 합니다.면 , 시간은 자동적으로 0 초로 계산되고 시간은 1 분이 증가합니다.







API	의사기호			피연산자		기능		제어기						
170	D	GRY	P	<b>(S)</b>	<b>(D)</b>	BIN → GRAY CODE		ES	EX	SS	SA	SX	SC	EH

Type OP	비트 장치				워드 장치											프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	GRY, GRYP: 5 steps
D								*	*	*	*	*	*	*	*	DGRY, DGRYP: 9 steps

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S**: 원시 장치 **D**: 결과를 저장하기 위한 종착 점

설명:

- S**에 의해 지정된 장치에서 **BIN** 값은 **GRAY CODE**의 등가로 변환되고, 변환된 결과는 **D**에 의해 지정된 장소에 저장됩니다.
- GRAY CODE**로 변환 될 수 있는 **S**의 범위는 보여지는 다음과 같다.

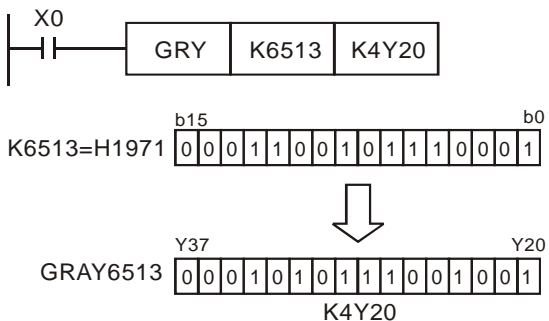
16-bit 명령 : 0~32,767

32-bit 명령 : 0~2,147,483,647

만일 **BIN** 값이 위에 보여지는 범위 밖에 있습니다.면, “연산 오류”로 결정 될 것입니다. 동시에 명령은 수행되지 않을 것이고 **D1067**에 오류 코드 **0E1A (HEX)**를 기록하는 **M1067=On**, **M1068=On**이 될 것입니다.

프로그램 예 :

**X0=On** 일 때, 상수 **K 6513**는 **GRAY CODE**로 변환되고 **K4Y20**에 저장됩니다.



API	의사기호			피연산자	기능	제어기						
171	D	GBIN	P	<b>S</b> <b>D</b>	GRAY CODE → BIN	ES	EX	SS	SA	SX	SC	EH

Type OP	비트 장치				워드 장치											프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S					*	*	*	*	*	*	*	*	*	*	*	GBIN, GBINP: 5 steps	
D							*	*	*	*	*	*	*	*	*	DGBIN, DGBINP: 9 steps	

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S**: 원시 GRAY CODE    **D**: 변환된 BIN 결과를 저장하는 종착 점

설명:

- S**에 의해 지정된 장치에서 GRAY CODE 값은 BIN의 증가로 변환되고, 변환된 결과는 **D**에 의해 지정된 장소에 저장됩니다.
- 이 명령은 PLC 입력에 연결되는 절대 위치 타입 엔코더(일반적으로 그레이 코드 엔코더입니다.)로부터 읽는데 사용될 수 있습니다.
- 프로그램 스캔 시간과 입력 반응 시간의 합은 **S**에 의해 지정된 출력 지연 시간과 동일합니다.
- 만약 원시 값이 X0~X17을 입력하는것으로 설정되어있습니다.면, REFF 명령(API151) 또는 D1020 (입력 반응 시간을 조절)을 사용함으로써 입력 반응 시간의 속도를 높일 수 있습니다.
- GRAY CODE로 변환 될 수 있는 **S**의 범위는 보여지는 다음과 같다:

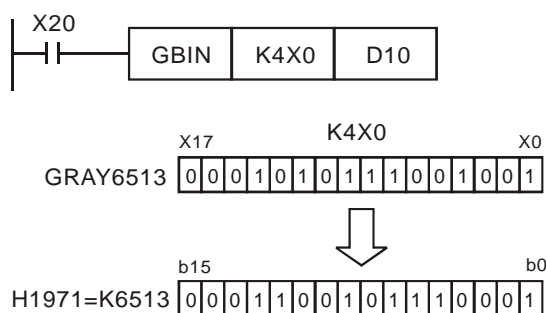
16-bit 명령 : 0~32,767

32-bit 명령 : 0~2,147,483,647

만일 GRAY CODE 값이 위에 보여지는 범위 밖의 값이라면, “연산 오류”로 결정 됩니다.

프로그램 예 :

X20=On 일 때, X0~X17 입력에 연결된 절대 위치 형태 엔코더 안의 GRAY CODE 값은 BIN 값으로 변환되고, D10에 저장됩니다.



API	의사기호		피연산자				기능	제어기						
180	MAND	P	<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	<b>(D)</b>	<b>(n)</b>	행렬 AND	ES	EX	SS	SA	SX	SC	EH

Type	비트 장치				워드 장치											프로그램 단계	
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MAND, MANDP: 9 단계	
S <sub>1</sub>							*	*	*	*	*	*	*				
S <sub>2</sub>							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*							*				

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S<sub>1</sub>**: 행렬 출처 장치 1    **S<sub>2</sub>**: 행렬 출처 장치 2    **D**: 결과를 계산하는 영역이 저장됩니다.

**n**: 행렬 길이 (n=K1~K256)

설명:

1. **n** 의 길이에 의한 행렬 출처 장치 1 과 2 에 매트릭스 AND 연산을 하고, 결과를 **n** 에 저장합니다.
2. 행렬 AND 의 연산 규칙 : 2 bits 가 모두 1 일때 bit 은 1 이고, 다른경우는 0.
3. 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, 와 **D** 는 iKnX, KnY, KnM, 와 KnS 에 표시합니다. SA 시리즈에서 오직 n=4 만 표시 가능합니다.  
EH 시리즈에서 오직 n≤4 만 표시 할 수 있습니다.

프로그램 예 :

X0=On 일 때, 16-bit 레지스터의 3 열(D0-D2)과 (D10-D12)에 MAND 과 행렬 AND 연산을 하여라. 그리고 나서 16-bit 레지스터의 3 열 (D20-D22)에 결과를 저장하여라.

Before execution

S<sub>1</sub>

D0

1

1

1

1

1

1

1

1

1

1

0

0

0

1

1

D1

1

1

1

1

1

1

1

1

1

1

0

0

0

1

1

D2

1

1

1

1

1

1

1

1

1

1

0

0

0

1

1

MAND

D10

0

0

0

1

0

0

1

0

0

0

1

1

0

1

0

0

D11

0

0

0

1

0

0

1

0

0

0

1

1

0

1

0

0

D12

0

0

0

1

0

0

1

0

0

0

1

1

0

1

0

0

After execution

D

D20

0

0

0

1

0

0

1

0

0

0

0

0

0

0

0

0

D21

0

0

0

1

0

0

1

0

0

0

0

0

0

0

0

0

D22

0

0

0

1

0

0

1

0

0

0

0

0

0

0

0

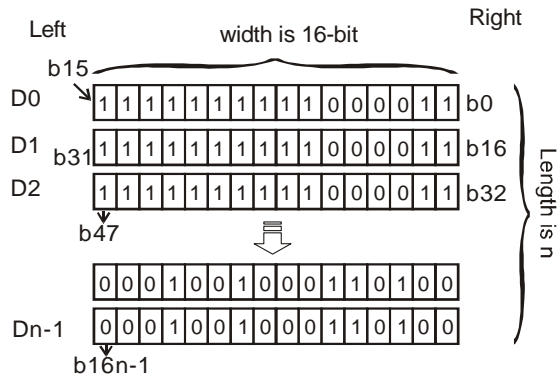
0

주의:

1. 행렬 명령에 관한 설명:
  - a) 행렬은 1 과 연속적인 16 비트 레지스터들 이상이 만들어 냅니다. 행렬이 만들어 낸 레지스터 수는 행렬 길이 n 이라 합니다. 행렬을 위한 16 X n 비트(dots)와 수행단위를 위한 이전 한 비트(dot) 이 있습니다.
  - b) 16 X n 비트들 (연속적인 수 b<sub>0</sub> - b<sub>16n-1</sub>) 은 행렬 명령을 위한 일련의 단일 포인트의 세트 처럼 간주될 것인다. 그러므로 설정에서 구체적인 점으로 작용하는 것은 의미가 없습니다.



- c) 행렬 명령은 이동, 복사, 비교, 검색 등에서 처럼 단일-포인트와 다중-포인트 또는 다중-포인트와 다중-포인트와의 관계를 위한 편리하고 중요한 응용 명령입니다.
- d) 일반적으로 행렬연산 중에  $16 \times n$  포인트의 한 포인트를 가리키기 위해 16-비트 레지스터가 필요합니다. Setting 범위는  $0 - 16n-1$  이고, 행렬에서 개별적으로 to  $b_0 - b_{16n-1}$  에 상응합니다.
- e) 실행들이 있습니다. : 연산 동안에 왼쪽 이동, 오른쪽 이동 또는 회전이 있습니다. 다음에 보여지는 것 처럼 큰 수는 왼쪽에 정의 되고 , 작은 수는 오른쪽에 정의됩니다.



- f) 행렬의 고정된 너비는 16-비트 입니다.
- g) Pr : 행렬 포인터. Pr 이 15 라면, 지정된 포인트는 b15 임을 의미합니다.
- h) 행렬의 길이는 n 이고, n 은 1-256 입니다.

예: D0 와 n=3, D0=HAAAA, D1=H5555, D2=HAAFF 으로 구성된 행렬

	C <sub>15</sub>	C <sub>14</sub>	C <sub>13</sub>	C <sub>12</sub>	C <sub>11</sub>	C <sub>10</sub>	C <sub>9</sub>	C <sub>8</sub>	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	
R <sub>0</sub>	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	D0
R <sub>1</sub>	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D1
R <sub>2</sub>	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	D2

예: K2X0 와 n=3, K2X0=H37, K2X10=H68, K2X20=H45 로 구성된 행렬

	C <sub>15</sub>	C <sub>14</sub>	C <sub>13</sub>	C <sub>12</sub>	C <sub>11</sub>	C <sub>10</sub>	C <sub>9</sub>	C <sub>8</sub>	C <sub>7</sub>	C <sub>6</sub>	C <sub>5</sub>	C <sub>4</sub>	C <sub>3</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>0</sub>	
R <sub>0</sub>	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X <sub>0</sub> ~X <sub>7</sub>
R <sub>1</sub>	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X <sub>10</sub> ~X <sub>17</sub>
R <sub>2</sub>	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X <sub>20</sub> ~X <sub>27</sub>

값이 비면 0 에서 R0(C<sub>15</sub>-C<sub>8</sub>), R1(C<sub>15</sub>-C<sub>8</sub>), R2(C<sub>15</sub>-C<sub>8</sub>)까지 채워져야 합니다.

API	의사기호			피연산자				기능		제어기						
181		MOR	P	<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	<b>(D)</b>	<b>(n)</b>	Matrix OR		ES	EX	SS	SA	SX	SC	EH

Type OP	비트 장치				워드 장치												프로그램 단계
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MOR, MORP: 9 steps	
S <sub>1</sub>							*	*	*	*	*	*	*				
S <sub>2</sub>							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*							*				

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S<sub>1</sub>**: 행렬 원시 장치 1    **S<sub>2</sub>**: 행렬 원시 장치 2.    **D**: Area where calculated result is stored

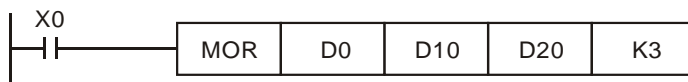
**n**: 행렬 길이 (**n**=K1~K256)

설명:

- n** 의 길이로 행렬 원시 장치 1 과 2 를 위한 행렬 OR 연산을 합니다. 그리고 **n** 에 결과를 저장합니다.
- 행렬 OR 의 연산 규칙은 : 두 비트가 서로 다를 때, 비트는 1 이고, 그렇지 않다면 비트는 0 입니다.
- 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, **D** 는 KnX, KnY, KnM, KnS 을 나타냅니다. SA 시리즈는 **n**=4 만 가능하고, EH 시리즈에서는, **n**≤4 만 가능합니다.

프로그램 예 :

X0=On 일 때, MOR 하고, 16-비트 레지스터의 3 열(D0-D2) 와 16-비트 레지스터의 3 열(D10-D12)을 위해 행렬 OR 연산을 합니다. 그리고 나서, 16-비트 레지스터의 3 열 (D20-D22)에 결과를 저장합니다.



Before Execution																	
		b15								b0							
		<b>(S<sub>1</sub>)</b> D0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		D1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		D2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		MOR															
After Execution	<b>(D)</b>	D20	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1
		D21	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1
		D22	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1

API	의사기호			피연산자				기능				제어기							
182		MXOR	P	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>D</b>	<b>n</b>	Matrix XOR				ES	EX	SS	SA	SX	SC	EH	

Type OP	비트 장치				워드 장치											프로그램 단계				
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MXOR, MXORP: 9 steps				
S <sub>1</sub>							*	*	*	*	*	*	*							
S <sub>2</sub>							*	*	*	*	*	*	*							
D								*	*	*	*	*	*							
n						*	*						*							

펄스								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

**S<sub>1</sub>:** 행렬 원시 장치 1    **S<sub>2</sub>:** 행렬 원시 장치 2    **D:** 연산된 결과가 저장되는 장소

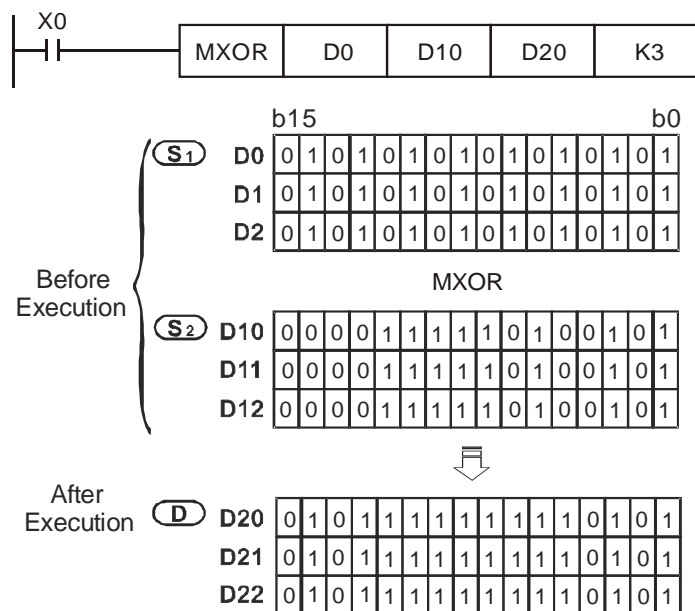
**n:** 행렬 길이 ( $n=K1 \sim K256$ )

설명:

1.  $n$  의 길이로 행렬 원시 장치 1 과 2 를 위한 행렬 XOR 연산을 합니다. 그리고  $n$  에 결과를 저장합니다.
2. 행렬 XOR 의 연산 규칙은 : 두 비트가 서로 다를 때, 비트는 1 이고, 그렇지 않다면 비트는 0 입니다.
3. 피연산자  $S_1, S_2, D$  는 KnX, KnY, KnM, KnS 을 나타냅니다. SA 시리즈는  $n=4$  만 가능하고, EH 시리즈에서는,  $n \leq 4$  만 가능합니다.

프로그램 예 :

X0=On 일 때, MXOR 하고, 16-비트 레지스터의 3 열(D0-D2) 와 16-비트 레지스터의 3 열(D10-D12)을 위해 행렬 XOR 연산을 합니다. 그리고 나서, 16-비트 레지스터의 3 열 (D20-D22)에 결과를 저장합니다.



API	의사기호			피연산자				기능				제어기							
183		MXNR	P	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>D</b>	<b>n</b>	Matrix XNR				ES EX SS SA SX SC EH							

Type OP	비트 장치				워드 장치												프로그램 단계 MXNR, MXNRP: 9 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F			
S <sub>1</sub>							*	*	*	*	*	*	*					
S <sub>2</sub>							*	*	*	*	*	*	*					
D								*	*	*	*	*	*					
n					*	*							*					

펄스								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

**S<sub>1</sub>**: 행렬 원시 장치 1    **S<sub>2</sub>**: 행렬 원시 장치 2    **D**: 연산 결과가 저장되는 장소

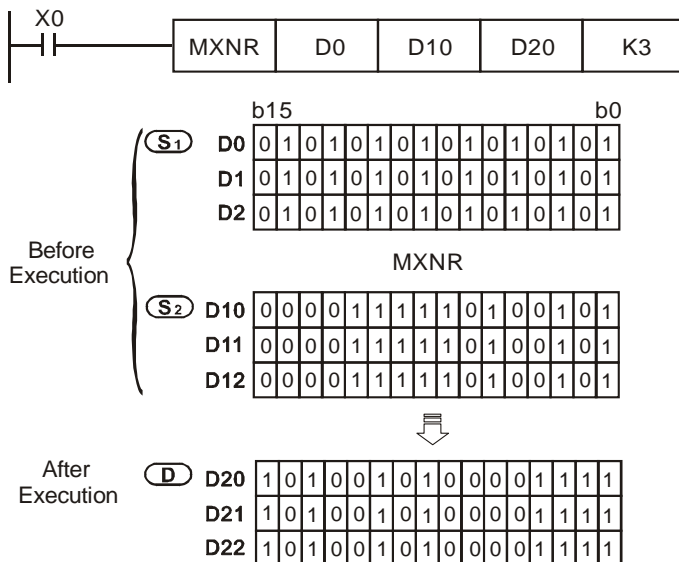
**n**: 행렬 길이 (K1~K256)

설명:

- n** 의 길이로 행렬 원시 장치 1 과 2 를 위한 행렬 XNR 연산을 합니다. 그리고 **n** 에 결과를 저장합니다.
- 행렬 XNR 의 연산 규칙은 : 두 비트가 서로 다를 때, 비트는 1 이고, 그렇지 않다면 비트는 0 입니다.
- 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, **D** 는 KnX, KnY, KnM, KnS 을 나타냅니다. SA 시리즈는 n=4 만 가능하고, EH 시리즈에서는, n≤4 만 가능합니다.

프로그램 예 :

X0=On 일 때, MXNR 하고, 16-비트 레지스터의 3 열(D0-D2) 와 16-비트 레지스터의 3 열(D10-D12)을 위해 행렬 XNR 연산을 합니다. 그리고 나서, 16-비트 레지스터의 3 열 (D20-D22)에 결과를 저장합니다.



API	의사기호			피연산자			기능			제어기							
184		MINV	P	<div>S</div>	<div>D</div>	<div>n</div>	역 행렬			ES	EX	SS	SA	SX	SC	EH	

Type OP	비트 장치				워드 장치										프로그램 단계	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MINV, MINVP: 7 steps
							*	*	*	*	*	*	*			
								*	*	*	*	*	*			
n					*	*							*			

필스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

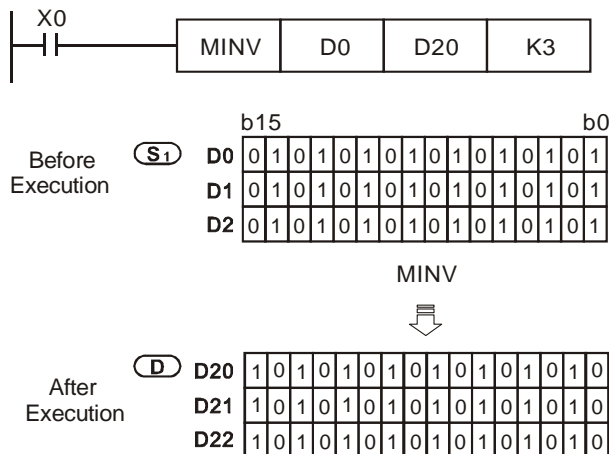
**S:** 행렬 원시 장치    **D:** 결과    **n:** 행렬 길이 (K1~K256)

설명:

1. **n** 의 길이로 행렬 원시 장치 1 을 위한 행렬 역 연산을 하고, **n** 에 결과를 저장합니다.
2. 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, **D** 는 KnX, KnY, KnM, KnS 를 나타냅니다. SA 시리즈를 위해서는, n=4 만 가능합니다. EH 시리즈는 n≤4 만 나타낼 수 있습니다.

프로그램 예 :

X0=On 일 때, 16-비트 레지스터의 3 열(D0-D2) 와 16-비트 레지스터의 3 열(D10-D12)을 위해 행렬 MINV 연산을 합니다. 그리고 나서, 16-비트 레지스터의 3 열 (D20-D22)에 결과를 저장합니다.



API	의사기호			피연산자				기능	제어기						
185		MCMP	P	<b>S<sub>1</sub></b>	<b>S<sub>2</sub></b>	<b>n</b>	<b>D</b>	행렬 비교	ES	EX	SS	SA	SX	SC	EH

Type	비트 장치				워드 장치												프로그램 단계
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MCMP, MCMPP: 9 steps	
S <sub>1</sub>							*	*	*	*	*	*	*				
S <sub>2</sub>							*	*	*	*	*	*	*				
n					*	*							*				
D							*	*	*	*	*	*	*	*	*		

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S<sub>1</sub>**: 행렬 원시 장치 1    **S<sub>2</sub>**: 행렬 원시 장치 2

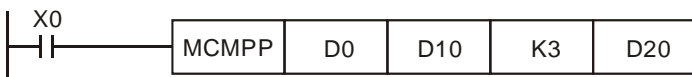
**n**: 목표 주소(K1~K256) 를 저장하는 포인터 **Pr**    **D**: 연산 결과가 저장되는 곳

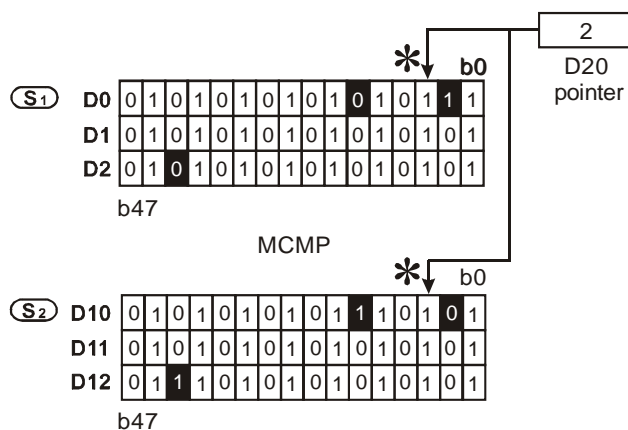
설명:

- 각각의 비교를 위해, Pr 주소로부터 **S<sub>1</sub>**, **S<sub>2</sub>** 의 각 비트를 비교할 것입니다. 다른 값의 주소를 찾아내고, 이 비교를 완료하기 위해 **n** 에 주소를 저장합니다.
- 비교 플래그 M1088로부터 비교 결과를 찾을 수 있습니다. 같다면, 차이를 위해, M1088=1, M1088=0 입니다. 비교가 완료 되면, 즉시 배교를 중단하고 flag M1091=1 을 찾는 bit 을 설정할 것입니다. 비교가 마지막 비트에 도달하며는 행렬 검색 마침 플래그 M1089는 On 될 것이고, 비교 도달 수는 **D** 에 저장됩니다. 다음 조사 기간을 위해 처음 비트로부터 비교가 시작 되고, 행렬 검색 시작 플래그 M1090=1 이 될 것입니다. **D** 가 사용 범위를 초과하면, , 포인트 에러 플래그 M1092 =1 입니다.
- 일반적으로 연산을 위한 행렬에서 16n 포인트의 하나를 가리키기 위해 16-비트 레지스터가 필요합니다. 이 레지스터를 포인터, Pr 이라 합니다. 이 Pr 은 사용자에게 의해 지정되고, 범위는 bit b<sub>0</sub> – b<sub>16n-1</sub> 에 각각 일치하는 0-16n-1 입니다. 올바른 비교 검색을 위해 연산에서 Pr 을 바꾸는 것을 피해야합니다. Pr 값이 이 범위를 초과하면, 행렬 포인터 에러 플래그 M1092 은 1 이 될 것 이고, 이 명령은 수행되지 않을 것입니다.
- 행렬검색 마침 플래그 M1089 과 set bit search flag M1091 은 같은 시점에 1 이 될 것입니다.
- 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, **D** 는 KnX, KnY, KnM, KnS 를 가리킨다. SA 시리즈에서는, n=4 만 가리킬 수 있습니다. EH 시리즈에서는 n≤4 만 가리킬 수 있습니다.

프로그램 예 :

- X0 가 Off→On 로 될 때, 행렬 검색 시작 플래그 M1090=0 이고, 현재값+ 1 인 비트로부터 다른비트를 찾기위해 비교를 시작할 것입니다. (M1088=0 은 다름을 의미합니다.)
- 포인터 D20=2 의 현재값일 때 ,4 개의 결과에 따를 수 있습니다., 4 번 동안 X0 가 Off→On 까지 수행될 때, 다음 네개의 결과 (❶, ❷, ❸, ❹)를 얻을 수 있습니다.
  - ❶ D20=5, 행렬 비트 검색 플래그 M1091=1, 행렬 검색 마침 플래그 M1089=0.
  - ❷ D20=45, 행렬 비트 검색 플래그 M1091=1, 행렬 검색 마침 플래그 M1089=0.
  - ❸ D20=47, 행렬 비트 검색 플래그 M1091=0, 행렬 검색 마침 플래그 M1089=1.
  - ❹ D20=1, 행렬 비트 검색 플래그 M1091=1, 행렬 검색 마침 플래그 M1089=0





### 주의:

플래그 신호를 위한 설명

M1088: 행렬 비교 플래그, 비교 결과가 같다면, M1088=1, 다르다면 M1088=0.

M1089: 행렬 검색 마침 플래그, 마지막 비트를 위한 비교일 때, M1089=1.

M1090: 행렬 검색 시작 플래그, 첫 번째 비트로부터 비교를 시작, M1090=1.

M1091: 행렬 비트 검색 플래그, 비교가 M1091=1 에 이르면 비교를 멈출 것입니다. .

M1092: 행렬 포인터 에러 플래그, 포인터 Pr 이 범위를 초과, M1092=1.

API	의사기호		피연산자	기능	제어기						
186	MBRD	P	<b>S</b> <b>n</b> <b>D</b>	Matrix Bit 판독	ES	EX	SS	SA	SX	SC	EH

Type	비트 장치				워드 장치											프로그램 단계	
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MBRD, MBRDP: 7 steps	
S							*	*	*	*	*	*	*				
n					*	*							*				
D							*	*	*	*	*	*	*	*	*		

펄스							16-bit							32-bit						
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S**: 행렬 원시 장치

**n**: 행렬 길이 (K1~K256)).

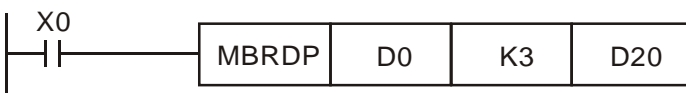
**D**: 포인터 Pr, 타깃 어드레스에 저장

설명:

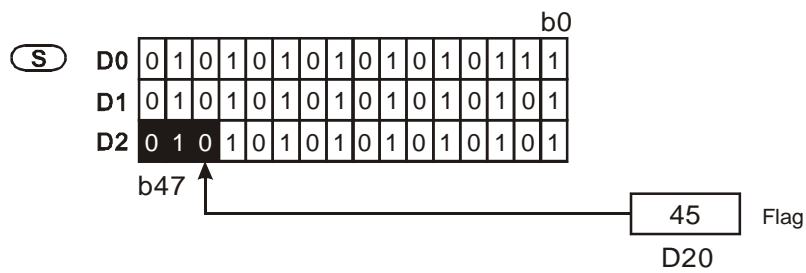
- 명령을 수행할 때 만약 **M1094** (매트릭스 포인터 제거 플래그)가 On 이면, 그것은 보는 것을 시작할 것입니다. 만약 그것이 On 상태이면, 포인터 **D** 을 제거하고 0 비트로부터 **S** 를 읽을 것이고, **M1095** (매트릭스 rotate/shift/output/carry)에 각 비트의 ON/OFF 상태를 읽을 것입니다.만약 비트를 읽은 후에 **M1093** (매트릭스 포인터 증가 플래그)가 On 이면 그것을 볼 것입니다.그리고 만약 그것이 켜져 있습니다.면 1 이 **D** 까지 증가할 것입니다..마지막 미트를 읽을때 **M1089** (매트릭스가 마지막 플래그 검색) =On 이면, 포인터 **D** 는 판독된 비트의 번호를 기록하고, 이 명령을 수행합니다.
- Pr (포인터)이 사용자에게 의해 지정되고, 범위가 to bit b0 – b16n-1 각각에 상응하는 0-16n-1 입니다. Pr 값이 이 범위를 초과하면, 행렬 포인터 에러 플래그 **M1092** 는 1 일 될것이고, 이 명령은 수행되지 않을 것입니다.
- 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, **D** 는 KnX, KnY, KnM, KnS 를 가리킨다. SA 시리즈는 n=4 만 가리킨다. EH 시리즈는 n≤4 만을 가리킨다.

프로그램 예 :

- X0 이 Off→On 되면, 포인터 클리어 플래그 **M1094=On**, 행렬포인터 증가 플래그 **M1093=1**, 그리고 비트를 기다린 후에 포인터 Pr 은 1 증가합니다.
  - 포인터 **D20=45** 의 값이 존재할 때, X0 가 3 번 동안 Off→On 이 수행되면, 다음의 세 결과를 얻을 수 있습니다. (❶, ❷, ❸).
- ❶ D20=46, matrix rotate/shift/output carry 플래그 **M1095=0**, 매트릭스는 끝난 플래그 **M1089=0** 를 검색합니다.
  - ❷ D20=47, matrix rotate/shift/output carry 플래그 **M1095=1**, 매트릭스는 끝난 플래그 **M1089=0** 를 검색합니다.
  - ❸ D20=47, matrix rotate/shift/output carry 플래그 **M1095=1**, 매트릭스는 끝난 플래그 **M1089=1** 를 검색합니다.





**주의:**

플래그 신호를 위한 설명:

M1089: 마지막 비트에 비교했을 때 M1089=1 이면, 매트릭스는 끝난 플래그를 검색합니다.

M1092: 매트릭스 포인터 오류 플래그, 포인터 Pr 가 그 범위를 넘을 때 M1092=1.

M1093: 매트릭스 포인터 증가 플래그, 현재 포인터에 1 을 추가

M1094: 매트릭스 포인터가 플래그 제거, 현재 포인터를 0 으로제거

M1095: matrix rotate/shift/output carry 플래그.

API	의사기호			피연산자			기능			제어기							
187		MBWR	P	<b>S</b>	<b>n</b>	<b>D</b>	Matrix Bit 기록			ES	EX	SS	SA	SX	SC	EH	

Type OP	비트 장치				워드 장치											프로그램 단계 MBWR, MBWRP: 7 steps			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F				
S								*	*	*	*	*	*						
n					*	*							*						
D								*	*	*	*	*	*	*	*				

펄스								16-bit								32-bit								
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		

피연산자:

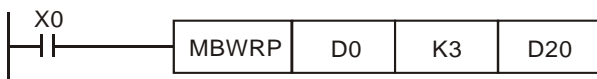
**S**: 매트릭스 원시 장치    **n**: 매트릭스 길이(K1~K256)    **D**: 목표 주소를 저장하는 포인터 Pr,

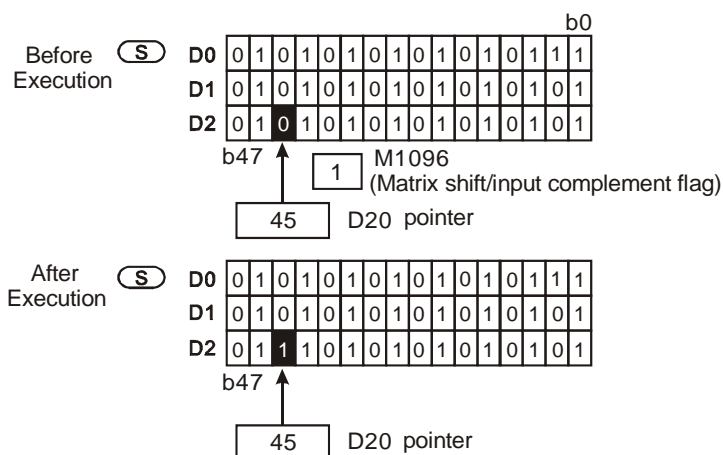
설명:

- 명령을 수행할 때 만약 **M1094** (매트릭스 포인터 제거 플래그) 가 On 상태 이면, 그것은 보기 시작할 것입니다. 만약 그것이 On 상태이면, 포인터 **D** 는 0으로 제거하고, **S** 의 0 비트에 **M1096** (matrix shift/input complement 플래그)를 기록 할 것입니다. 비트를 기록한 후 **M1093** (매트릭스 포인터 증가 플래그)이 On 상태 이면 그것이 불 것입니다. 만약 그것이 On 상태라면, 1 이 **D** 까지 증가할 것입니다. 마지막 비트에 기록할 때, **M1089** (매트릭스 검색 종료 플래그) =On 이면, 포인터 **D** 는 판독된 비트의 번호를 기록하고 이 명령 수행을 종료합니다. 만약 **D** 가 범위를 초과하면 **M1092=1** 이 됩니다.
- Pr (포인터)는 사용자에게 의해 설계되고 그 범위는 각각 bit b0 – b16n-1 에 대응하는 0-16n-1 입니다. 만약 Pr 값이 이 범위를 초과하면, 매트릭스 포인터 오류 플래그 **M1092** 이 1 이 될 것이고, 이 명령은 수행되지 않을 것입니다.
- 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, 와 **D** 는 KnX, KnY, KnM, 와 KnS 을 나타냅니다. SA 에서, 그것은 n=4 만 나타낼 수 있습니다. EH 시리즈에서, n≤4 만 나타낼 수 있습니다.

프로그램 예 :

- X0 가 Off→On 일 때, 포인터 제거 플래그 **M1094=On**, 매트릭스 포인터 증가 플래그 **M1093=1**, 그리고 비트를 기록한 후에 포인터 Pr 까지 1 이 증가합니다.
- 현재 포인터가 **D20=45** 일 때, **M1094** (matrix shift/input complement 플래그) =1 입니다. X0 가 Off→On 으로 수행 되면, 다음의 결과를 얻을 수 있습니다.:





### 주의:

플래그 신호의 설명

M1089: 매트릭스 검색 종료 플래그, 마지막 비트에 비교할 때, M1089=1.

M1092: 매트릭스 포인터 오류플래그, 포인터 Pr 이 그 범위를 초과할 때, M1092=1.

M1093: 매트릭스 포인터 증가 플래그, 현재 포인터에 1 추가

M1094: 매트릭스 포인터 제거 플래그, 현재 포인터를 0 으로 제거

M1096: matrix shift/input complement 플래그

API	의사기호			피연산자			기능			제어기						
188		MBS	P	<b>S</b>	<b>D</b>	<b>n</b>	Matrix Bit 이동			ES	EX	SS	SA	SX	SC	EH

Type OP	비트 장치				워드 장치										프로그램 단계		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E			F
S							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*							*				

펄스							16-bit							32-bit							
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	

피연산자:

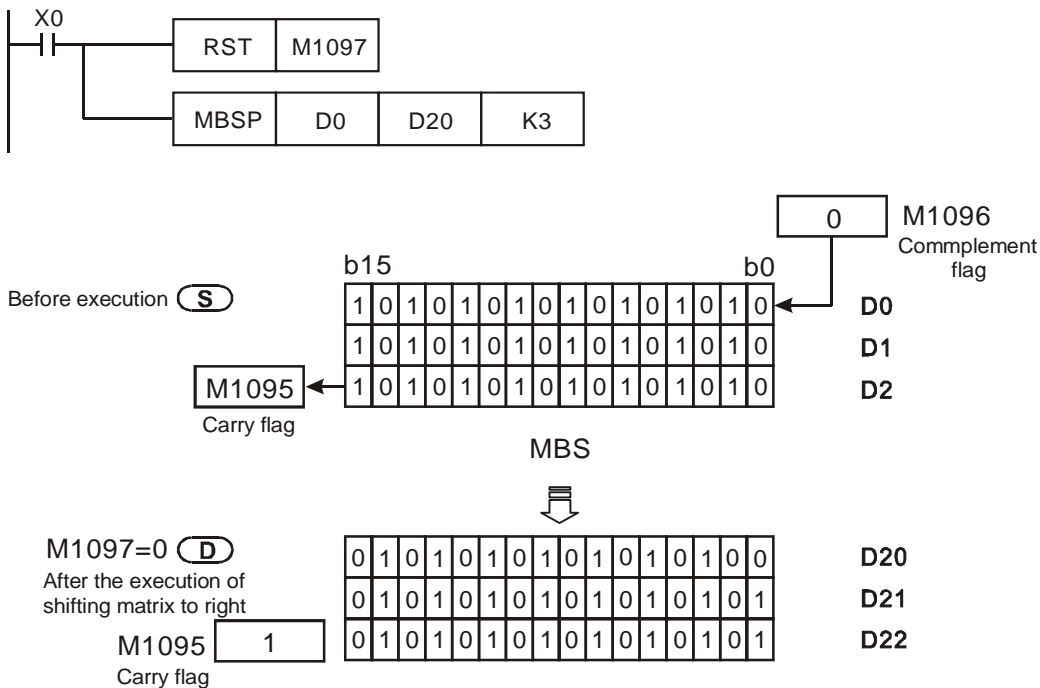
**S:** 매트릭스 원시 장치 **D:** 결과 **n:** 매트릭스 길이(K1~K256)

설명:

- 이 명령은 매트릭스 길이에 의해 오른쪽, 왼쪽으로 **S** 이동하는데 사용됩니다. M1097=0 은 왼쪽으로 움직이고, M1097=1 은 오른쪽으로 움직인다. 각 비트를 위한 발송 때문에 공 비트(왼쪽 이동은 b0 이고 오른쪽 이동은 b16n-1 입니다.)를 채우는데 M1096 (보완 플래그)상태 사용을 필요로 합니다.16n-1). 만약 발송(shift to left is b16n-1 and shift to right is b0), 때문에 하나의 비트가 더 있습니다.면, **D** 결과를 저장하고 M1095 (운반 플래그)에 상태를 보내는 것을 필요로 합니다..
- 이 명령의 최 상의 사용은 펄스 수행 명령입니다.(MBSP).
- 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, 와 **D** 는 KnX, KnY, KnM, 와 KnS 를 나타냅니다. SA 시리즈에서, 그것은 n=4 만 나타내고, EH 시리즈에서, 그것은 n≤4 만 나타냅니다.

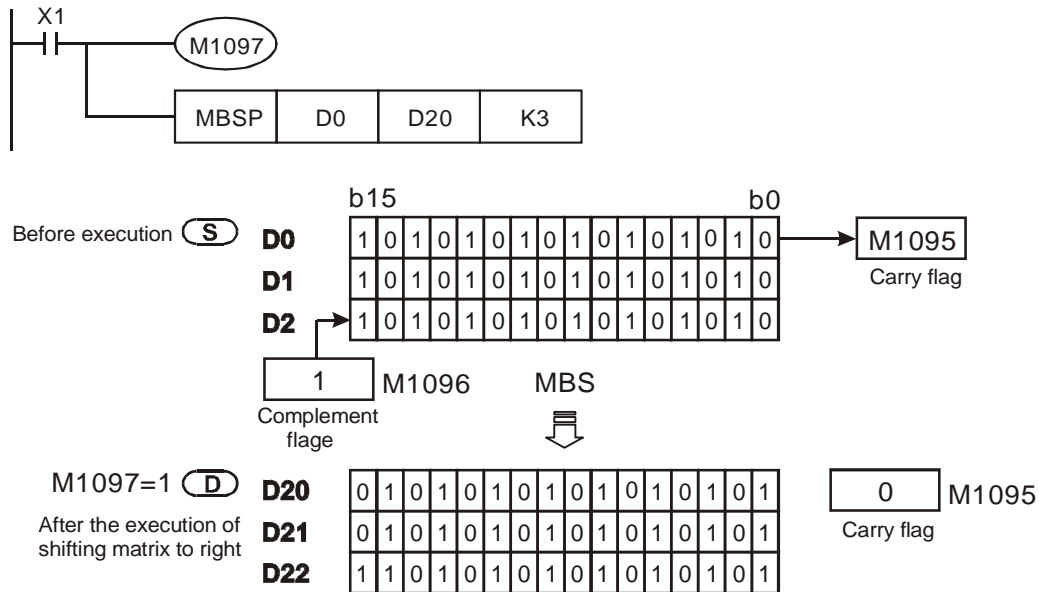
프로그램 예 1:

X0=On 일 때, M1097=Off 는 매트릭스를 왼쪽으로 이동하십시오.는 의미입니다. 보완 플래그 M1096=0 를 설정할 때, 16-bit 레지스터를 D0-D2 를 왼쪽으로 이동시키고, 16-bit 레지스터 D20-D22 결과를 16-bit 레지스터 D20-D22 에 저장하고. 운반 플래그 M1095 는 1 이 될 것입니다.



## 프로그램 예 2:

X1=On 일 때, M1097=On 은 매트릭스를 오른쪽으로 이동시킵니다. 보완 플래그 M1096=1 를 설정하면, 16-bit 레지스터 D0-D2 는 오른쪽으로 이동하고, 16-bit 레지스터 D20-D22 에 그 결과를 저장합니다. 그리고 운반 플래그 1095 는 0 이 될 것입니다.



## 설명:

플래그 신호 설명

M1095: 매트릭스 회전/이동/출력 운반 플래그

M1096: 매트릭스 이동/출력 보완 플래그

M1097: 매트릭스 회전/이동 방향 플래그

API	의사기호				피연산자				기능				제어기							
189		MBR	P	<div>S</div> <div>D</div> <div>n</div>	Matrix Bit Rotator				<div>ES</div> <div>EX</div> <div>SS</div> <div>SA</div> <div>SX</div> <div>SC</div> <div>EH</div>											

<div>Type</div> <div>OP</div>	비트 장치				워드 장치											프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MBR, MBRP: 7 steps			
	S						*	*	*	*	*	*	*						
	D							*	*	*	*	*	*						
	n					*	*						*						

펄스								16-bit								32-bit								
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		

피연산자:

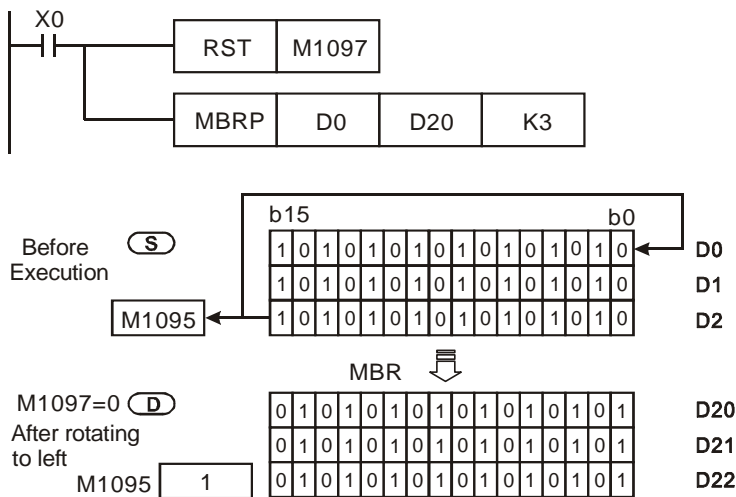
**S:** 매트릭스 원시 장치 **D:** 결과 **n:** 매트릭스 길이(K1~K256)

설명:

- 이 명령은 **S** 를 매트릭스 길이에 의해 오른쪽 또는 왼쪽으로 회전시키기 위해 사용됩니다. **M1097=0** 는 왼쪽으로 움직이고, **M1097=1** 는 오른쪽으로 움직인다. 공비트(왼쪽으로의 회전은 **b0** 이고, 오른쪽으로의 이동은 **16n-1** 입니다.) 는 회전 때문에 회전된 그 비트(왼쪽으로 회전은 **b16n-1** 이고 오른쪽으로 이동은 **b0** 입니다.)에 의해 회전될 것이고, **D** 에 저장될 것입니다. 회전된 그 비트는 공 비트를 채우는데 사용될 뿐만 아니라 운반 플래그 **M1095** 에 그 상태를 보내는데도 사용됩니다.
- 이 명령어의 최상의 사용은 펄스 수행 명령입니다.(**MBRP**).
- 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, 와 **D** 는 **KnX**, **KnY**, **KnM**, 와 **KnS** 를 나타냅니다. **SA** 시리즈에서, 그것은 **n=4** 만 나타내고, **EH** 시리즈에서, 그것은 **n≤4** 만 나타냅니다.

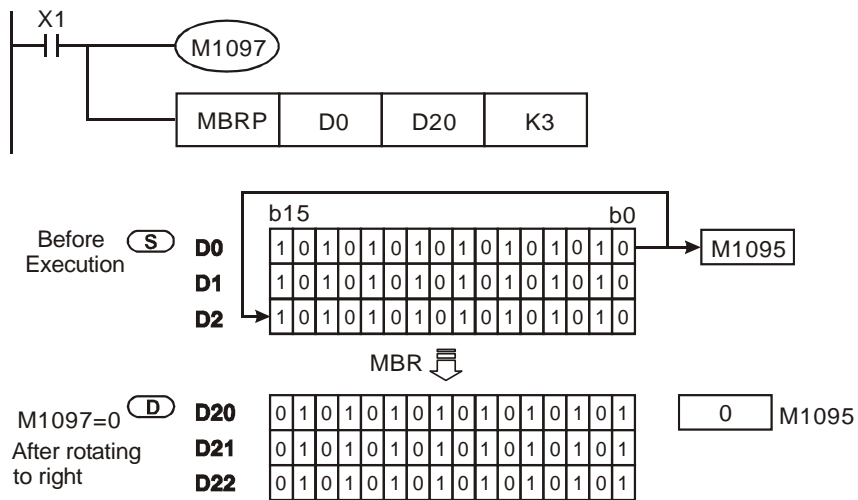
프로그램 예 1:

**X0=On** 일 때 **M1097=Off** 는 매트릭스를 왼쪽으로 회전시키는 것을 의미합니다. 16-bit 레지스터 **D0-D2** 를 왼쪽으로 회전시키고 16-bit 레지스터 **D20-D22** 에 저장합니다. 운반 플래그 **M1095** 는 1 이 될 것입니다.



프로그램 예 2:

**X1=On** 일 때, **M1097=On** 은 매트릭스를 오른쪽으로 회전시킵니다. 16-bit 레지스터 **D0-D2** 를 오른쪽으로 회전시키고, 16-bit 레지스터 **D20-D22** 에 결과를 저장합니다. 운반 플래그 **M1095** 는 0 이 될 것입니다.

**주의:**

플래그 신호 설명

M1095: 매트릭스 회전/이동/출력 운반 플래그

M1097: 매트릭스 회전/이동 방향 플래그

API	의사기호			피연산자			기능								제어기							
190		MBC	P	S	n	D	Matrix Bit State Count								ES	EX	SS	SA	SX	SC	EH	

Type OP	비트 장치				워드 장치												프로그램 단계							
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MBC, MBCP: 7 steps								
S							*	*	*	*	*	*	*											
n					*	*							*											
D								*	*	*	*	*	*	*	*									

펄스								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

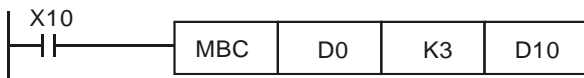
**S**: 행렬 지원 장치    **n**: 행렬 길이(K1~K256)    **D**: 결과

설명:

1. 행렬 길이 **n** 과 **D** 에 있는 숫자에 의해 1 비트나 0 비트 숫자가 더해진다.
2. M1098=1 일 때, 1 비트의 숫자가 계수됩니다. 그리고 M1098 이 0 일 경우엔 0 비트의 숫자가 계수 됩니다.
3. 피연산자 **S<sub>1</sub>**, **S<sub>2</sub>**, 그리고 **D** 는 KnX, KnY, KnM, 그리고 KnS 를 가리킨다. SA 시리즈에선, n=4 만을 가리킨다. EH 시리즈에선, n≤4 만을 가리킨다.

프로그램 예 :

X10 가 켜지면 , D0-D2 의 숫자에 비트 1 이 계수되고 D10 에 모든 숫자가 저장됩니다. M1098=0 이면, D0-D2 의 숫자에 비트 0 이 계수되고 D10 에 모든 숫자가 저장됩니다.



D0	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D2	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1

D10    12    M1098=0

D10    36    M1098=1

주의:

행렬 명령에 관한 설명:

M1098: 행렬 계수 비트 1 이나 비트 0 의 표시 신호

M1099: 계산 결과가 0 이면 실행됩니다.



API	의사기호			피연산자	기능										제어기						
196		HST	P	<b>(S)</b>	High Speed Counter										ES	EX	SS	SA	SX	SC	EH

Type OP	비트 장치				워드 장치										프로그램 단계								
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	HST, HSTP: 3 steps							
S					*	*																	

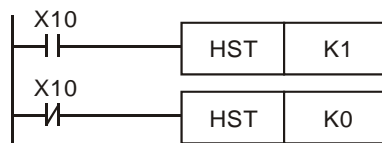
펄스								16-bit								32-bit							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

**설명:**

- S**: 초고속 타이머의 시작을 멈추는 상태. 피연산자의 가능범위 **S**: S=K0(H0), K1(H1).
- S=1** 일 때, 초고속 타이머를 시작하고 M1015 를 켜고. 초고속 타이머를 시작하고 D1015 에 현재값이 기록됩니다. D1015 의 단위는 100us 입니다.
- D1015 의 계산 범위는 K0-K32767 입니다. 계산이 K32767 까지 되면, 다음 계산은 0 에서 시작 됩니다.
- S=0** 일 때 ,초고속 타이머와 M1015 는 꺼지고, D1015 는 계산을 즉시 멈춥니다.
- S** 가 1 이나 0 이 아닐 경우, **HSTMR** 명령은 발동되지 않습니다.

**프로그램 예 :**

- X10 이가 켜지면, **M1015** 가 켜진다. 그것은 초고속 타이머를 시작시키고 D1015 에 현재값이 저장됩니다.
- X10 이 꺼지면, **M1015** 가 꺼지고. 이것은 초고속 타이머를 멈추게 합니다.

**주의:**

- 표시문자에 관한 설명  
**M1015**: 초고속 타이머 시작 신호  
**D1015**: 초고속 타이머
- 이 명령은 EH 모델을 지원하지 않습니다. 다음은 특별한 M 신호와 D 신호를 직접적으로 이용하는 방법에 관한 설명들입니다.
  - 이것은 PLC 가 실행되는 경우에만 사용됩니다.
  - M1015** 가 켜지면, 스캔 기간 동안 PLC 의 종료 명령이 수행 되면 D1015 초고속 타이머만 실행 됩니다.
  - D1015 의 범위는 K0-K32767 입니다. K32767 까지 계산되면, 다음 계산은 0 에서 시작됩니다.
  - M1015** 가 꺼지면, 종료 명령이나 HST 명령에 D1015 가 계산을 멈추게 됩니다.
- 이 명령은 SA/SX/SC 모델을 지원하지 않는다. 다음은 특별한 M 신호와 D 신호를 직접적으로 이용하는 방법에 관한 설명들입니다.
  - PLC 의 수행여부에 관계없이 언제나 유효합니다.
  - D1015 가 켜지면, 초고속 타이머 D1015 가 즉시 실행됩니다. D1015 의 단위는 100us 입니다.
  - D1015 의 범위는 K0-K32767 입니다. K32767 까지 계산이 된 경우엔 다음 계산은 0 에서 시작합니다.
  - M1015** 가 꺼지면, D1015 계산은 즉시 멈춥니다.

API	연상기호			피연산자		기능								제어기																						
215~217	D	LD#		<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	논리 접촉 동작								ES	EX	SS	SA	SX	SC	EH																
형태 OP	비트 장치				워드 장치												프로그램 단계																			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LD#: 5 단계																				
	S <sub>1</sub>				*	*	*	*	*	*	*	*	*	*	*	*	DLD#: 9 단계																			
	S <sub>2</sub>				*	*	*	*	*	*	*	*	*	*	*	*																				
																펄스				16 비트				32 비트												
																ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

S<sub>1</sub>: 데이터 출처 장치 1    S<sub>2</sub>: 데이터 출처 장치 2

설명:

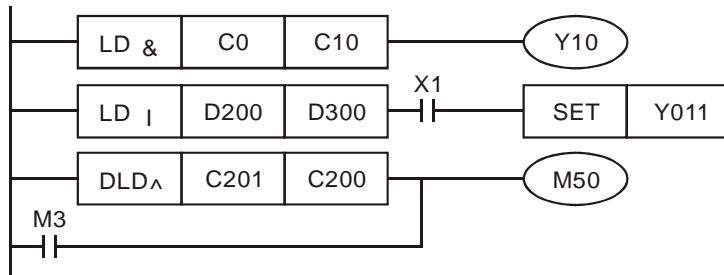
- S<sub>1</sub> 와 S<sub>2</sub> 의 내용을 비교하십시오. “LD&” 를 예로 들어, 비교 결과가 0 이 아니면, 접촉은 연속적이며, 0 일 경우 접촉은 비연속적입니다.
- LD# 명령은 BUS 에 직접 연결 가능합니다..

API No.	16 비트 명령	32 비트 명령	연속적 상태	비연속적 상태
215	LD&	DLD&	S <sub>1</sub> & S <sub>2</sub> ≠ 0	S <sub>1</sub> & S <sub>2</sub> = 0
216	LD	DLD	S <sub>1</sub>   S <sub>2</sub> ≠ 0	S <sub>1</sub>   S <sub>2</sub> = 0
217	LD^	DLD^	S <sub>1</sub> ^ S <sub>2</sub> ≠ 0	S <sub>1</sub> ^ S <sub>2</sub> = 0

- & : 논리적 “AND” 동작
- | : 논리적 “OR” 동작
- ^ : 논리적 “XOR” 동작
- 32 비트 길이 카운터(C200~)가 비교를 위해 이 명령에 놓인다면, 32 비트 명령(DLD#)를 사용하십시오. 16 비트 명령(LD#)이 활성화될 경우 CPU 는 “프로그램 오류”로 결정하여 MPU 창에 빨간 “오류” 지시가 반짝거릴 것입니다.

프로그램 예 :

- C0 과 C10 내용을 비교하기 위해 사용된 LD&(논리적 “AND” 동작)명령의 결과가 0 과 같지 않으면 Y10=ON 입니다.
- D200 과 D300 의 내용을 비교하기 위해 사용된 LD| (논리적 “OR” 동작)명령의 결과가 0 과 같지 않고 X1=ON 이면, Y10=ON 으로 유지됩니다.
- C201 과 C200 의 내용을 비교하기 위해 사용된 LD^(논리적 “XOR”동작)명령의 결과가 0 이 아니거나 M3=ON 일 경우, M50=ON 이 됩니다.



API	연상기호			피연산자		기능			제어기							
218~220	D	AND#		<b>(S<sub>1</sub>) (S<sub>2</sub>)</b>		연속적 논리 접촉 동작			<div>ES</div> <div>EX</div> <div>SS</div> <div>SA</div> <div>SX</div> <div>SC</div> <div>EH</div>							

형태 OP	비트 장치				워드 장치										프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	AND#: 5 단계  DAND#: 9 단계		
S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*	*			
S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*			

펄스								16 비트								32 비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

S<sub>1</sub>: 데이터 출처 장치 1    S<sub>2</sub>: 데이터 출처 장치 2

설명:

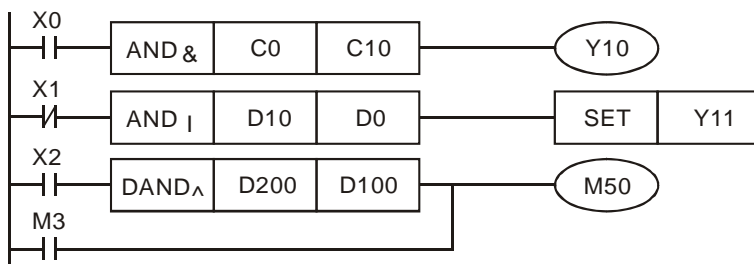
- S<sub>1</sub> 와 S<sub>2</sub> 의 내용을 비교하십시오. “AND&” 를 예로 들어, 비교 결과가 0 이 아니면, 접촉은 연속적이며, 0 일 경우 접촉은 비연속적입니다.
- AND# 명령은 직렬 접촉에 연결하는데 사용됩니다.

API No.	16 비트 명령	32 비트 명령	연속적 상태	비연속적 상태
218	AND&	DAND&	S <sub>1</sub> & S <sub>2</sub> ≠ 0	S <sub>1</sub> & S <sub>2</sub> = 0
219	AND	DAND	S <sub>1</sub>   S <sub>2</sub> ≠ 0	S <sub>1</sub>   S <sub>2</sub> = 0
220	AND^	DAND^	S <sub>1</sub> ^ S <sub>2</sub> ≠ 0	S <sub>1</sub> ^ S <sub>2</sub> = 0

- & : 논리적 “AND” 동작
- | : 논리적 “OR” 동작
- ^ : 논리적 “XOR” 동작
- 32 비트 길이 카운터(C200~)가 비교를 위해 이 명령에 놓인다면, 32 비트 명령(DAND#)를 사용하십시오. 16 비트 명령(AND#)이 활성화될 경우 CPU 는 “프로그램 오류”로 결정하여 MPU 창에 빨간 “오류” 지시가 반짝거릴 것입니다.

프로그램 예 :

- X0=ON 이고, C0 과 C10 내용을 비교하기 위해 AND&(논리적 “AND” 동작)명령의 결과가 0 과 같지 않으면 Y10=ON 입니다.
- X1=OFF 이고, D10 과 D0 의 내용을 비교하기 위해 사용된 ANDI (논리적 “OR” 동작)명령의 결과가 0 과 같지 않으면, Y11=ON 으로 유지됩니다.
- X2=ON 이고, 32 비트 레지스터 D200(D201)과 D100(D101)의 내용을 비교하기 위해 사용된 AND^(논리적 “XOR”동작)명령의 결과가 0 이 아니거나 M3=ON 일 경우, M50=ON 이 됩니다.



API	연상기호			피연산자		기능								제어기							
221~223	D	OR#		<b>(S<sub>1</sub>)</b> <b>(S<sub>2</sub>)</b>		병렬 논리적 접촉 동작								ES	EX	SS	SA	SX	SC	EH	

형태 OP	비트 장치				워드 장치										프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	OR#: 5 단계  DOR#: 9 단계		
	S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*		*	
S <sub>2</sub>						*	*	*	*	*	*	*	*	*	*	*		

펄스								16 비트								32 비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

**S<sub>1</sub>**: 데이터 출처 장치 1      **S<sub>2</sub>**: 데이터 출처 장치 2

설명:

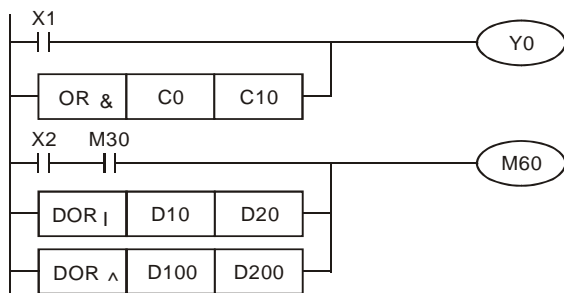
- S<sub>1</sub>** 와 **S<sub>2</sub>** 의 내용을 비교하십시오. “OR&” 를 예로 들어, 비교 결과가 0 이 아니면, 접촉은 연속적이며, 0 일 경우 접촉은 비연속적입니다.
- OR#명령은 병렬 접촉을 연결하기 위해 사용됩니다.

API No.	16 비트 명령	32 비트 명령	연속적 상태	비연속적 상태
221	OR&	DOR&	<b>S<sub>1</sub> &amp; S<sub>2</sub> ≠ 0</b>	<b>S<sub>1</sub> &amp; S<sub>2</sub> = 0</b>
222	OR	DOR	<b>S<sub>1</sub>   S<sub>2</sub> ≠ 0</b>	<b>S<sub>1</sub>   S<sub>2</sub> = 0</b>
223	OR^	DOR^	<b>S<sub>1</sub> ^ S<sub>2</sub> ≠ 0</b>	<b>S<sub>1</sub> ^ S<sub>2</sub> = 0</b>

- & : 논리적 “AND” 동작
- | : 논리적 “OR” 동작
- ^ : 논리적 “XOR” 동작
- 32 비트 길이 카운터(C200~)가 비교를 위해 이 명령에 놓인다면, 32 비트 명령(DOR#)를 사용하십시오. 16 비트 명령(OR#)이 활성화될 경우 CPU 는 “프로그램 오류”로 결정하여 MPU 창에 빨간 “오류” 지시가 반짝거릴 것입니다.

프로그램 예 :

- X1=ON 이고, C0 과 C10 내용을 비교하기 위해 사용한 OR&(논리적 “AND” 동작)명령의 결과가 0 과 같지 않으면, Y0=ON 입니다.
- X2 와 M30 이 모두 “ON”이거나 D10 과 D20 내용을 비교하기 위해 사용한 ORI(논리적 “OR” 동작)명령의 결과가 0 과 같지 않거나, 또는 D100 과 D200 의 내용을 비교하기 위해 사용한 OR^(논리적 “XOR” 동작)의 결과가 0 과 같지 않으면, M60=ON 입니다.



API	연상기호			피연산자		기능								제어기							
224~230	D	LD*		<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	Contact Comparison								ES	EX	SS	SA	SX	SC	EH	

형태 OP	비트 장치				워드 장치										프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LD*: 5 단계  DLD*: 9 단계		
	S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*		*	
S <sub>2</sub>						*	*	*	*	*	*	*	*	*	*	*		

펄스								16 비트								32 비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

**S<sub>1</sub>**: 데이터 출처 장치 1    **S<sub>2</sub>**: 데이터 출처 장치 2

설명:

1. **S<sub>1</sub>** 와 **S<sub>2</sub>** 의 내용을 비교하십시오. API 224 “LD=” 를 예로 들어, 비교 결과가 “=”이면, 접촉은 연속적이며, “≠”일 경우 접촉은 비연속적입니다.

2. LD\* 명령은 BUS 에 직접적으로 연결 가능합니다..

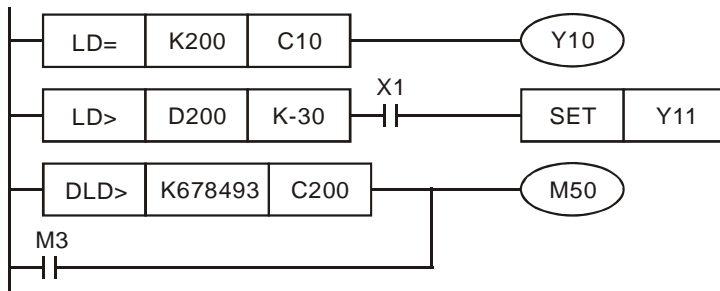
API No.	16 비트 명령	32 비트 명령	연속적 상태	비연속적 상태
224	LD=	DLD=	<b>S<sub>1</sub>=S<sub>2</sub></b>	<b>S<sub>1</sub>≠S<sub>2</sub></b>
225	LD>	DLD>	<b>S<sub>1</sub>&gt;S<sub>2</sub></b>	<b>S<sub>1</sub>≤S<sub>2</sub></b>
226	LD<	DLD<	<b>S<sub>1</sub>&lt;S<sub>2</sub></b>	<b>S<sub>1</sub>≥S<sub>2</sub></b>
228	LD<>	DLD<>	<b>S<sub>1</sub>≠S<sub>2</sub></b>	<b>S<sub>1</sub>=S<sub>2</sub></b>
229	LD≤	DLD≤	<b>S<sub>1</sub>≤S<sub>2</sub></b>	<b>S<sub>1</sub>&gt;S<sub>2</sub></b>
230	LD≥	DLD≥	<b>S<sub>1</sub>≥S<sub>2</sub></b>	<b>S<sub>1</sub>&lt;S<sub>2</sub></b>

3. 왼쪽 대부분 비트가 MSB (16 비트 명령: b15, 32 비트 명령: b31), **S<sub>1</sub>** 부터 **S<sub>2</sub>** 가 1 일때 , 가치에 대한 비교는 비교의 부정적 가치로서 보여집니다.

4. 32 비트 길이 카운터(C200~)가 비교를 위해 이 명령에 놓인다면, 32 비트 명령(DLD\*)를 사용하십시오. 16 비트 명령(LD\*)이 활성화될 경우 CPU 는 “프로그램 오류”로 결정하여 MPU 창에 빨간 “오류” 지시가 반짝거릴 것입니다.

프로그램 예 :

- C10 카운터의 내용이 K200 과 같으면, Y10=ON 입니다.
- D200 의 내용이 K-20 이하이고 X1=ON 일 때, Y11=ON 으로 유지됩니다.
- C200 의 내용이 K678,493 보다 적거나 M3=ON 이면, M50=ON 입니다.



API	연상기호			피연산자		기능			제어기						
232~238	D	AND*		<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	Series Contact Comparison			ES EX SS SA SX SC EH						

OP	형태	비트 장치				워드 장치										프로그램 단계			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	AND*: 5 단계  DAND*: 9 단계			
	S <sub>1</sub>					*	*	*	*	*	*	*	*	*	*		*		
	S <sub>2</sub>					*	*	*	*	*	*	*	*	*	*	*			

펄스								16 비트								32 비트							
ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH		ES	EX	SS	SA	SX	SC	EH	

피연산자:

S<sub>1</sub>: 데이터 출처 장치 1    S<sub>2</sub>: 데이터 출처 장치 2

설명:

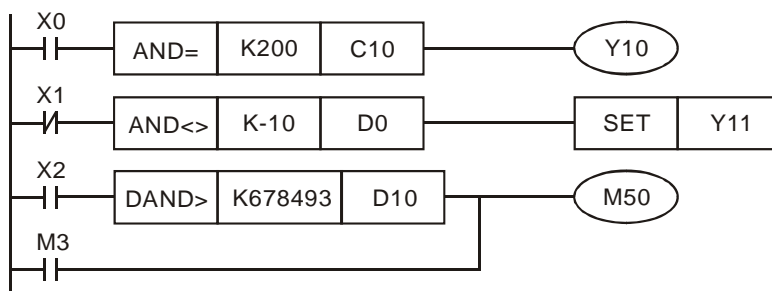
- S<sub>1</sub> 와 S<sub>2</sub> 의 내용을 비교하십시오. API 232 “AND=” 를 예로 들어, 비교 결과가 “=”이면, 접촉은 연속적이며, “≠” 일 경우 접촉은 비연속적입니다.
- AND\* 명령은 직렬 접촉을 연결하는 비교 명령입니다.

API No.	16 비트 명령	32 비트 명령	연속적 상태	비연속적 상태
232	AND=	DAND=	S <sub>1</sub> =S <sub>2</sub>	S <sub>1</sub> ≠S <sub>2</sub>
233	AND>	DAND>	S <sub>1</sub> >S <sub>2</sub>	S <sub>1</sub> ≤S <sub>2</sub>
234	AND<	DAND<	S <sub>1</sub> <S <sub>2</sub>	S <sub>1</sub> ≥S <sub>2</sub>
236	AND<>	DAND<>	S <sub>1</sub> ≠S <sub>2</sub>	S <sub>1</sub> =S <sub>2</sub>
237	AND≤	DAND≤	S <sub>1</sub> ≤S <sub>2</sub>	S <sub>1</sub> >S <sub>2</sub>
238	AND≥	DAND≥	S <sub>1</sub> ≥S <sub>2</sub>	S <sub>1</sub> <S <sub>2</sub>

- 왼쪽 대부분 비트가 MSB (16 비트 명령: b15, 32 비트 명령: b31), S<sub>1</sub> 부터 S<sub>2</sub> 가 1 일때 , 가치에 대한 비교는 비교의 부정적 가치로서 보여집니다.
- 32 비트 길이 카운터(C200~C254)가 비교를 위해 이 명령에 놓인다면, 32 비트 명령(DAND\*)를 사용하십시오. 16 비트 명령(AND\*)이 활성화될 경우 CPU 는 “프로그램 오류”로 결정하여 MPU 창에 빨간 “오류” 지시가 반짝거릴 것입니다.

프로그램 예 :

- X=ON 이고 C10 카운터의 최근 값이 K200 과 같으면, Y10=ON 입니다.
- X1=OFF 이고 레지스터 D0 의 내용이 K-10 과 같지 않다면, Y11=ON 으로 유지됩니다.
- X2=ON 이고 32 비트 레지스터 D11 과 D0 의 내용이 K678,493 보다 적으면, M50=ON 입니다.



API	연상기호			피연산자		기능				제어기										
240~246	D	OR*		<b>(S<sub>1</sub>)</b>	<b>(S<sub>2</sub>)</b>	Parallel Contact Comparison				ES	EX	SS	SA	SX	SC	EH				
형태 OP	비트 장치				워드 장치										프로그램 단계					
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	OR*: 5 단계  DOR*: 9 단계				
	S <sub>1</sub>				*	*	*	*	*	*	*	*	*	*	*					
	S <sub>2</sub>				*	*	*	*	*	*	*	*	*	*	*					
펄스						16 비트						32 비트								
ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH	ES	EX	SS	SA	SX	SC	EH

피연산자:

**S<sub>1</sub>**: 데이터 출처 장치 1    **S<sub>2</sub>**: 데이터 출처 장치 2

설명:

- S<sub>1</sub>** 와 **S<sub>2</sub>** 의 내용을 비교하십시오. API 240 (OR=) 를 예로 들어, 비교 결과가 “=”이면, 접촉은 연속적이며, “≠”일 경우 접촉은 비연속적입니다.
- OR\* 명령은 병렬 접촉을 연결하는 비교 명령입니다.

API No.	16 비트 명령	32 비트 명령	연속적 상태	비연속적 상태
240	OR=	DOR=	<b>S<sub>1</sub>=S<sub>2</sub></b>	<b>S<sub>1</sub>≠S<sub>2</sub></b>
241	OR>	DOR>	<b>S<sub>1</sub>&gt;S<sub>2</sub></b>	<b>S<sub>1</sub>≤S<sub>2</sub></b>
242	OR<	DOR<	<b>S<sub>1</sub>&lt;S<sub>2</sub></b>	<b>S<sub>1</sub>≥S<sub>2</sub></b>
244	OR<>	DOR<>	<b>S<sub>1</sub>≠S<sub>2</sub></b>	<b>S<sub>1</sub>=S<sub>2</sub></b>
245	OR≤	DOR≤	<b>S<sub>1</sub>≤S<sub>2</sub></b>	<b>S<sub>1</sub>&gt;S<sub>2</sub></b>
246	OR≥	DOR≥	<b>S<sub>1</sub>≥S<sub>2</sub></b>	<b>S<sub>1</sub>&lt;S<sub>2</sub></b>

- 왼쪽 대부분 비트가 MSB (16비트 명령: b15, 32비트 명령: b31), **S<sub>1</sub>** 부터 **S<sub>2</sub>** 가 1 일때 , 가치에 대한 비교는 비교의 부정적 가치로서 보여집니다.
- 32 비트 길이 카운터(C200~C254)가 비교를 위해 이 명령에 놓인다면, 32 비트 명령(DOR\*)를 사용하십시오. 16 비트 명령(OR\*)이 활성화될 경우 CPU 는 “프로그램 오류”로 결정하여 MPU 창에 빨간 “오류” 지시가 반짝거릴 것입니다.

프로그램 예 :

- X1=ON 이고 C10 카운터의 최근 값이 K200 과 같으면, Y0=ON 입니다.
- X2 와 M30 이 모두 “ON”이거나 32 비트 레지스터 D101 와 D100 의 내용이 K100,000 이상이면, M60=ON 입니다.

