반도체형 프로그래머블 로직 콘트롤러

TinyPLC
TPC3X
TPC26

사용자 설명서

- TPC26 도 TPC3X 시리즈중 하나입니다.

COMFILE TECHNOLOGY

컴파일 테크놀로지 주식회사
www.comfile.co.kr

머리말

TPC3X 시리즈는 1998 년에 발표된 제품이므로, 새로운 프로젝트를 시작하시는 분들은 CUBLOC 을 사용하시기 바랍니다.

TPC3X 와 관련해서 가장 많이 질문하시는 부분.

- TPC3X 시리즈는 TPCWORKS 가 아닌, TinyPLC Studio 를 사용해야 합니다.
- TinyPLC Studio 는 WINDOWS XP 에서만 사용가능합니다. (7,10 지원안함)

- PLC 셋업에서 <소스업로드 가능>을 체크해놓고 다운로드한 경우에만 업로드기능을 사용하실 수 있습니다.
  <소스업로드 가능>을 체크하지 않고 다운로드했다면, TPC3x 모듈에 소스관련 정보가 포함되어 있지 않으므로, 업로드가 불가능합니다.
  만약, 소스업로드를 실행하였는데, 에러가 발생하거나 정상적으로 소스가 표시되지 않는다면, <소스업로드 가능>을 체크하지 않은 상태에서 다운로드한 것입니다.

- 저희 회사는 TPC3X 코어모듈을 제조한 회사입니다. 간혹 시스템 (예를들면 포장시스템)에서 문제가 발생했다고, 저희 회사에 전화하셔서 해결을 요구하시는 분들이 있는데, 이런 경우 그 시스템을 설계하신 분에게 연락을 하셔야 합니다.
  저희 회사는 해당시스템을 설계하지 않았으므로 그 내용에 대해서 알고있지 않습니다. 이점 양지하여주시기 바랍니다.

- TPC26 과 TPC3X 시리즈는 통신기능을 지원하지 않습니다.

컴파일 테크놀로지 (주)

주의사항

1. 본 제품을 현장에 바로 설치하지 마시고, 사무실에서 충분히 사전 검토작업을 거쳐서 동작에 대한 확신이 있을 때 현장 적용을 하시기 바랍니다.

2. 인체에 유해를 가할 수 있는 어플리케이션에는 사용을 금지하여 주시기 바랍니다. (예 : 의료용 장비, 각종 절단기)

3. 기존 타사 PLC 에 대하여 잘 알고 있는 유저분들도 본 사용설명서를 반드시 읽어보신 후 사용을 부탁드립니다.

4. PLC 의 특성상 열악한 환경에서 동작하는 경우가 많습니다. 전원입력단에 노이즈필터를 연결하는 등, 각종 노이즈 대책에 신경써 설치하여 주십시오.

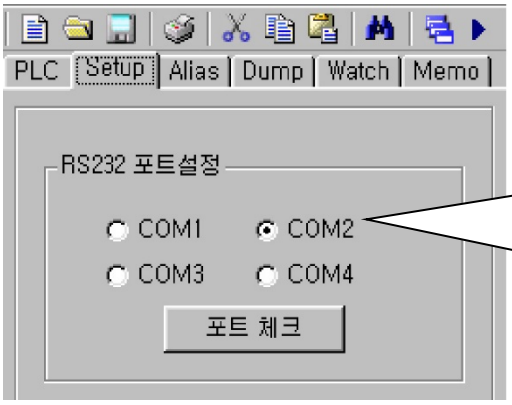
5. 문의전화가 폭주하고 있습니다. 대부분의 문의전화는 본 사용설명서에 있는 내용이였습니다. 본 사용설명서를 반드시 숙지하신 뒤에 제품을 사용해주시기 바랍니다.

특히 다운로드가 안된다는 문의전화가 많습니다. 전화하시기 전에 본 사용설명서를 충분히 참조해주시기 바랍니다.

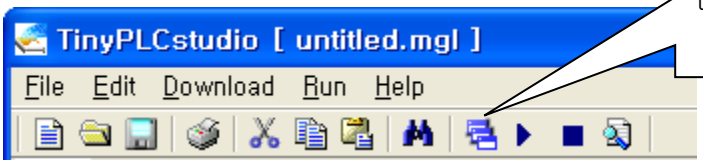
TPC3X 로 다운로드하는법

3-PIN 다운로드 케이블을 사용하여 PC 와 TPC3X 를 연결하여 주십시오.

TinyPLC Studio 에서 RS232C 포트를 셋팅해 주어야 합니다.



연결한 COM 포트를 선택하세요. 설정상태는 저장되므로 프로그램 기동 시 마다 매번 설정할 필요는 없습니다

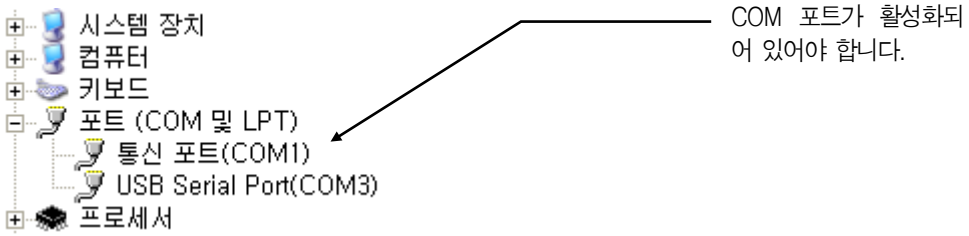


AUTO 아이콘을 누르면 다운로드가 시작됩니다.

만약 다운로드가 안된다면 다음과 같은 사항을 점검해보세요.

WINDOWS 가 COM 포트를 제대로 인식하고 있는지 살펴보세요.

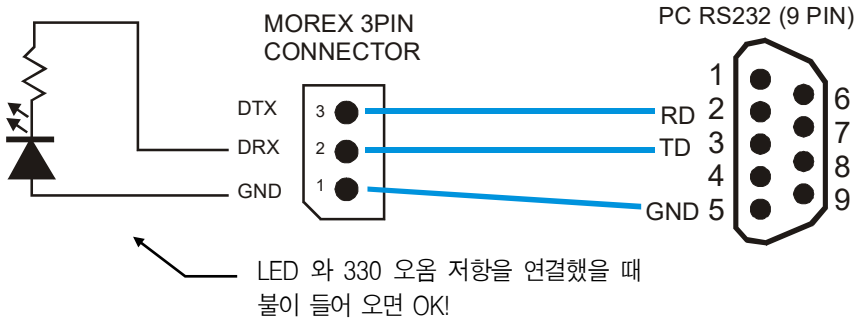
<내컴퓨터> -- <제어판> -- <시스템>까지 더블 클릭으로 들어가 보면 아래와 같은 화면이 표시됩니다. 여기에서 해당 COM 포트가 활성화되어있는지 확인해야 합니다.



Invalid RS232C port Error 또는 Device Unavailable 에러 발생시...

이 경우에는 PC 상의 COM 포트 셋업에 문제가 있는 경우입니다. BIOS 상의 셋업상태와 윈도우 시스템에서의 셋업상태를 다시 한번 점검해보시기 바랍니다. 그래도 안된다면, 전문가의 도움을 받아야 할 것입니다. 이 에러메시지는 TinyPLC Studio 프로그램이 COM 포트를 찾지 못하는 문제이므로, TPC3X 하드웨어 상태와는 무관한 메시지입니다.

다음은 컴파일에서 제공하는 **케이블의 연결도** 입니다. 직접 케이블을 제작하신 분들은 아래의 연결도와 똑같이 배선되어 있는지 확인하기 바랍니다.



그래도 문제가 해결되지 않았다면, 마지막으로 해볼 수 있는 방법은 다른 PC 에서 설치해 보는 것입니다. 이 과정에서 기존 PC 의 문제점을 찾아낼 수도 있습니다.

USB 포트 다운로드

USB 포트로 다운로드하기 위해서는 USB_RS232 변환 케이블이 있어야 PC 와 PLC 간에 통신이 가능합니다.

USB-RS232 케이블과 3 핀
다운로드 케이블을 결합해서 사용

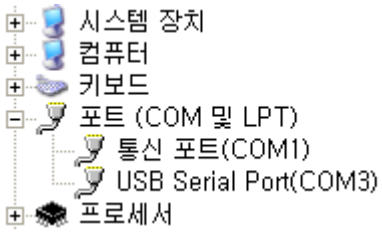
http://www.comfile.co.kr/product/03_42usbrs232cable.htm ← USB 드라이버

USB 드라이버를 다운받아서 설치한뒤에 USB-RS232 변환케이블을 PC 에 장착하세요.

주의사항

예전에 판매하였던, 아래 제품은 사용하지 마십시오.
드라이버 충돌이 일어날 가능성이 있습니다.

"제어판 => 시스템 => 장치 관리자" 에서 포트를 보면 USB Serial Port 라는 항목이 나옵니다. 아래 그림을 보면 COM3 이로 할당된 것을 보실 수 있습니다.

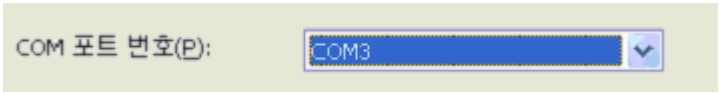


만약 COM4 를 넘는 번호가 할당되었다면, 다음과 같은 방법으로 바꿀 수 있습니다.

위 화면에서 USB Serial Port 를 우클릭한뒤 <속성>을 선택하십시오.

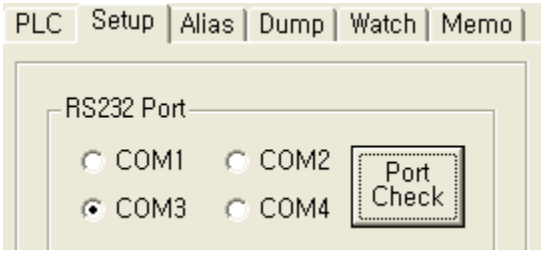
그럼 위와 같은 화면이 나옵니다. 여기서 <고급>을 클릭하세요.. 그럼 다음과 같은 화면이 나옵니다. 이곳에서 COM 포트번호를 조정할 수 있습니다.

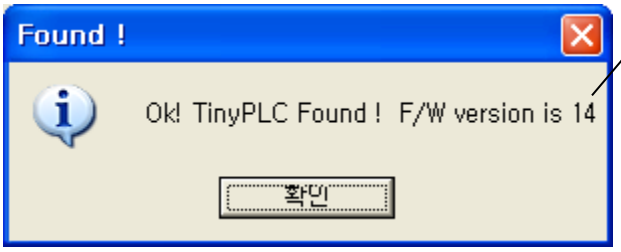
이 값을 1 ~ 4 사이의 값으로 (현재 사용하지 않는 번호를 확인한뒤) 선택하세요.. 여기서 선택한 값으로 TinyPLC Studio COM 포트 셋업을 바꾸신후 사용하세요.



포트 체크 방법

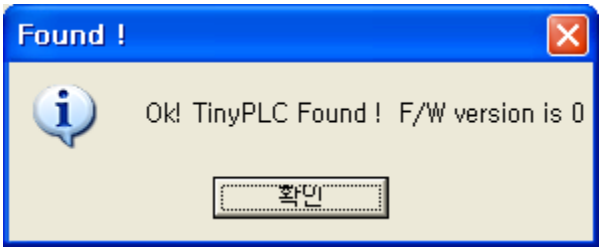
Setup 탭에 있는 <Port Check> 를 누르면 TPC3x 와 통신해서 F/W 버전 번호를 알려줍니다.



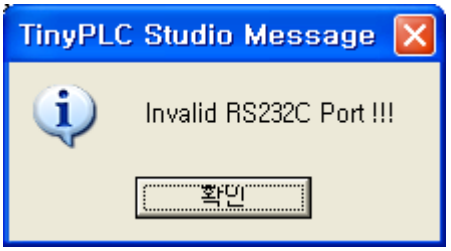


번호가 있어야 정상!

이 버전번호가 0 이 나오면, COM 포트자체는 이상이 없지만, TPC3X 가 연결 안된 것입니다.



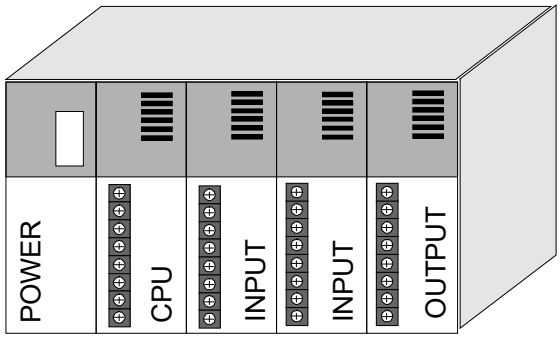
만약 "Invalid RS232C Port" 가 나온다면, 해당 COM 포트는 쓸수 없는 포트입니다.



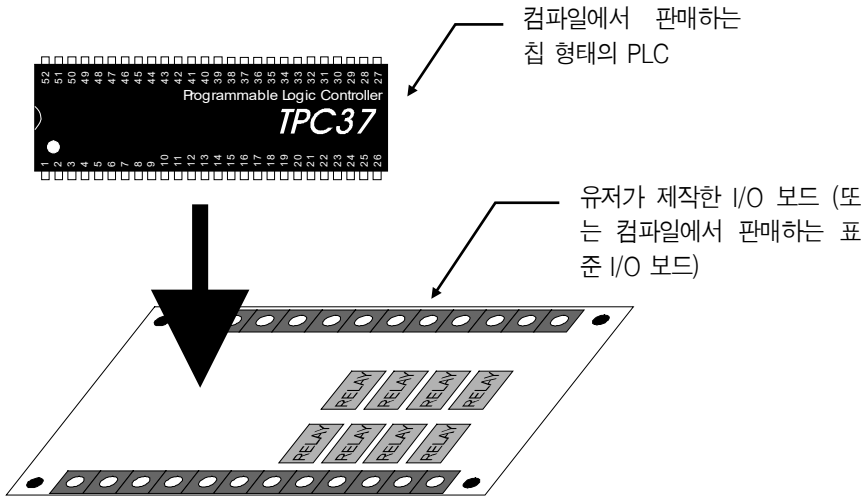
제 1 장
TPC3X 개요

TPC3X 의 개요

프로그래머블 로직 컨트롤러 (PLC)는 산업현장에서 가장 많이 쓰이고 있는 대표적인 컨트롤러입니다. 아래 그림은 지금까지 여러분이 보아왔던 보편적인 형태의 PLC 입니다. 일체형 케이스안에 전원부, CPU, I/O 등이 모두 내장되어 있는 형태로 되어 있어서, 제어반 내부에 장착하고, 필요시 UNIT 등을 추가해서 원하는 기능을 구현할 수 있도록 되어 있습니다.

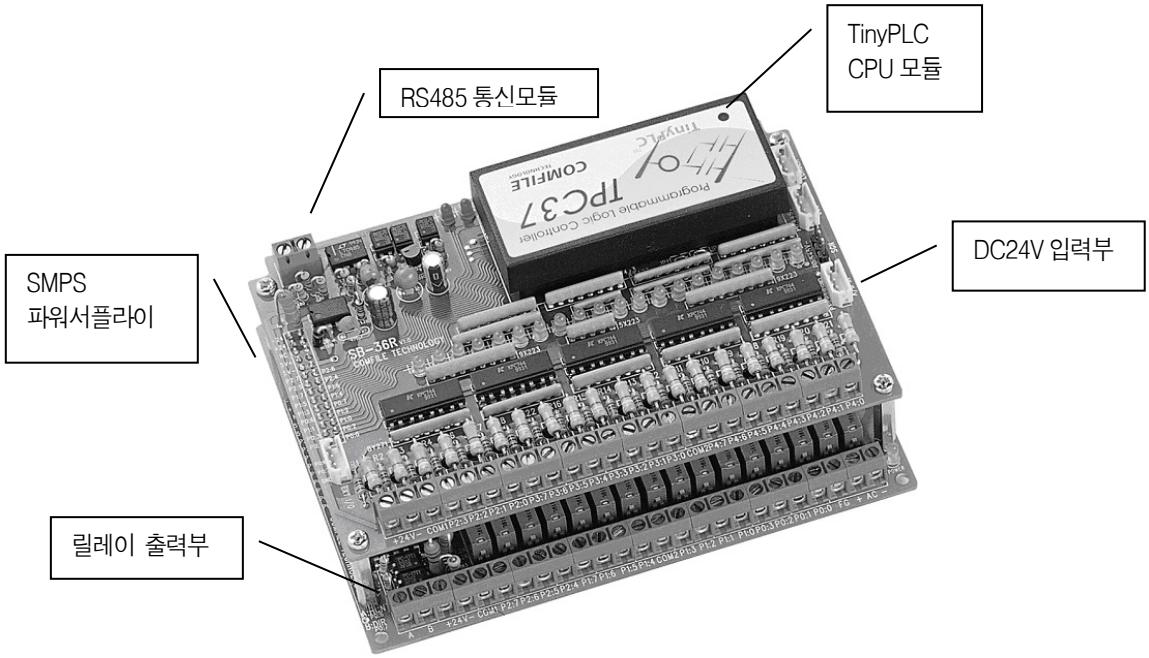


TPC3X 는 이러한 종래의 구성법과는 전혀 다른 방법을 채택하고 있습니다. PLC 는 칩 형태로 (기존 PLC 의 CPU 에 해당하는 부분) 판매하고 있으며, 여기에 필요한 I/O 회로를 추가해서 유저가 원하는 자신만의 독특한 PLC 를 구성할 수 있습니다.



일체형 PLC

일체형 PLC 는 CPU 모듈과 SMPS (스위칭 전원 모듈) 릴레이 출력, DC24V 입력부를 모두 갖춘 제품군입니다. 다음은 SB-36R 의 구성을 설명한 그림입니다.



다음은 일체형 PLC 제품군입니다.

일체형 모델명	CPU	전원입력	입출력 점수
SB-14R	TPC33	AC85V ~ 264V	DC24V 입력 : 8 점 RELAY 출력 : 6 점 RS485 통신포트 내장
SB-22R	TPC33	AC85V ~ 264V	DC24V 입력 : 12 점 RELAY 출력 : 10 점 RS485 통신포트 내장
SB-36R	TPC37	AC85V ~ 264V	DC24V 입력 : 20 점 RELAY 출력 : 16 점 RS485 통신포트 내장, 엔코더 입력포트 내장

TPC3X 모듈의 특징

TPC3X CPU 모듈의 기능 및 용량 비교표입니다.

항목	TPC 26	TPC 33	TPC 37	TPC 38
프로그램 메모리	16K byte	16K byte	128K byte	128K byte
스캔타임	2.5mS	2.5 또는 5mS	2.5 또는 5mS	5mS
기본 명령어	27 종	27 종	27 종	27 종
응용 명령어	84 종	84 종	84 종	84 종
입출력 릴레이 (P 영역)	26 점 (26 점 모두 I/O 설정 가능)	24 점 (24 점 모두 I/O 설정 가능)	40 점 (24 점 I/O 설정 가능, 8 점은 입력 전용 , 8 점은 출력 전용)	68 점 (24 점 I/O 설정 가능, 20 점은 입력 전용 , 24 점은 출력 전용)
DA 출력 (DA 영역)	없음	없음	없음	10 비트, 2 채널 (PWM)
보조릴레이 (M 영역)	256 점	256 점	1024 점	1024 점
스탭콘트롤 (S 영역)	16 조, 255 스탭	16 조, 255 스탭	32 조, 255 스탭	32 조, 255 스탭
KEEP 릴레이 (K 영역)	512 점	512 점	512 점	512 점
타이머 (T 영역)	64 점 (워드)	64 점 (워드)	256 점 (워드)	256 점 (워드)
카운터 (C 영역)	32 점 (워드)	32 점 (워드)	256 점 (워드)	256 점 (워드)
데이터 (D 영역)	220 워드	220 워드	1024 워드	1024 워드
LCD 디스플레이 버퍼 (CH 영역)	20 by 4 (80 바이트)	20 by 4 (80 바이트)	20 by 4 (80 바이트)	20 by 4 (80 바이트)
SGN 디스플레이 버퍼 (G 영역)	5 by 8 (40 바이트)	5 by 8 (40 바이트)	5 by 8 (40 바이트)	5 by 8 (40 바이트)
AD 보관용 (AD 영역)	8 워드 (10 비트, 8 채널 A/D 내장) (P2:X 를 A/D 입력으로 사용)	8 워드 (10 비트, 8 채널 A/D 내장) (P2:X 를 A/D 입력으로 사용)	8 워드 (10 비트, 8 채널 A/D 내장) (P3:X 를 A/D 입력으로 사용)	8 워드 (10 비트, 8 채널 A/D 내장) (P3:X 를 A/D 입력으로 사용)
고속 타이머 (CNT 영역)	1 워드 (16 BIT)	1 워드 (16 BIT)	1 워드 (16 BIT)	1 워드 (16 BIT)
펙케지	40 핀 DIP 형 (반도체 칩)	36 핀 DIP 형	52 핀 DIP 형	80 핀 DIP 형

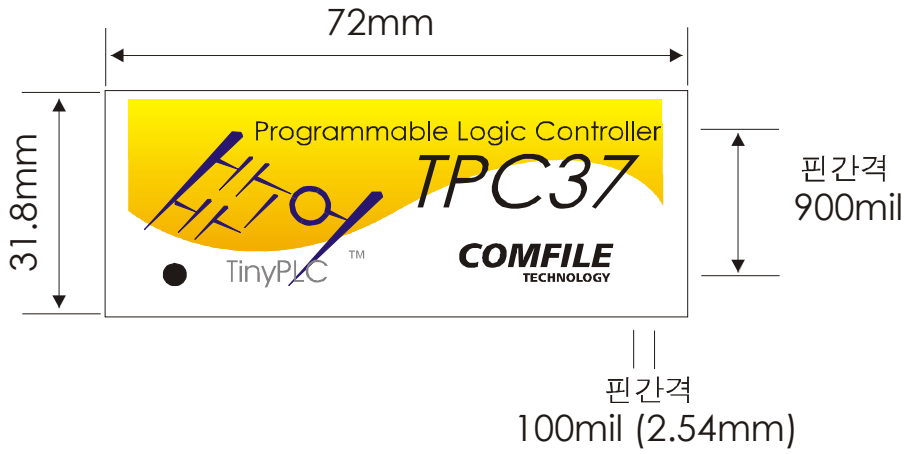
- SGN 은 컴파일 테크놀로지의 “세븐 세그먼트 디스플레이 모듈”의 제품명입니다.
- LCD 모듈은 컴파일 테크놀로지의 “시리얼 LCD 모듈”만 접속 가능 합니다. (1 선 접속방식)

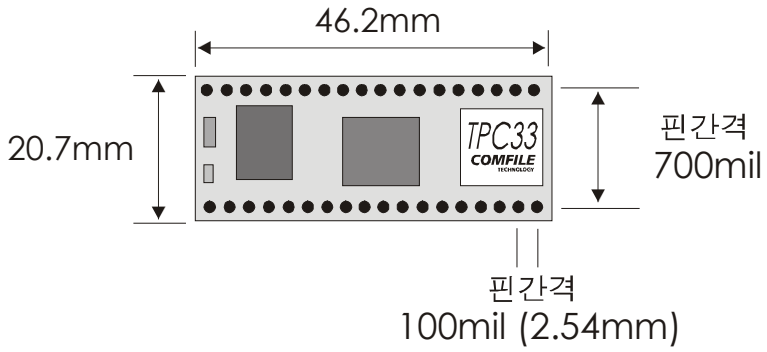
TPC3X 규격

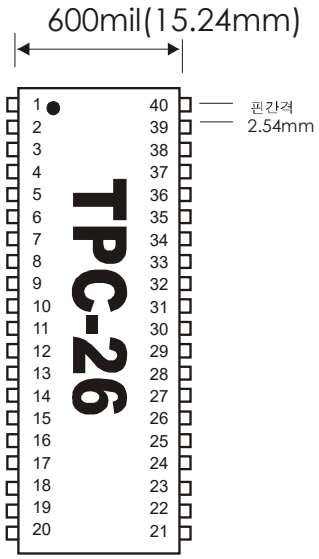
TPC3X 에 적용되는 일반적인 규격에 대한 설명입니다.

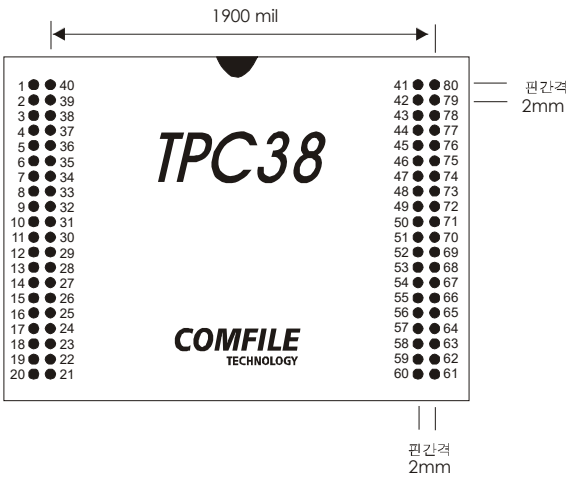
항목	규격
SMPS 전원 입력	AC 85V~254V 50~60Hz (AU-0524 적용 시)
TPC3X 전원	DC 5V 단일전원
소비 전력	10VA
정전 허용	10mS
동작 온도	0~55 도
보관 온도	-10 ~ 70 도
사용 습도	5 ~95% RH, 이슬 맺힘이 없을 것
사용 환경	먼지와 개스가 없을 것
노이즈 내력	1500V Vpp, 1uS
내압	AC 1500V / 분
절연저항	10M 옴 이상 (DC 500V 절연저항계)
내진동	16.7Hz 복진 2mm, 2 시간
충격	10G (3 shock each in 3 axes)
그라운드	100 옴

TPC3X 외형


72mm
31.8mm
Programmable Logic Controller
TPC37
TinyPLC
COMFILE
TECHNOLOGY
핀간격
900mil
핀간격
100mil (2.54mm)

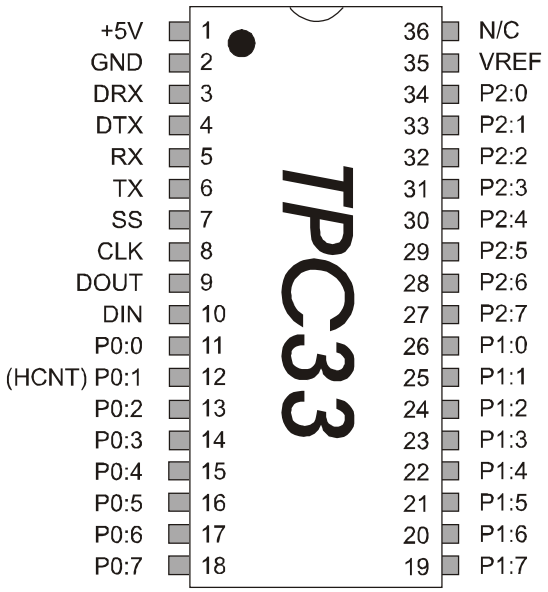

46.2mm
20.7mm
TPC33
COMFILE
핀간격
700mil
핀간격
100mil (2.54mm)


600mil(15.24mm)
TPC-26
핀간격
2.54mm


1900 mil
TPC38
COMFILE
TECHNOLOGY
핀간격
2mm
핀간격
2mm

TPC 33 핀아웃

다음은 TPC33 모델의 핀 아웃 (PIN OUT)에 대한 설명입니다.



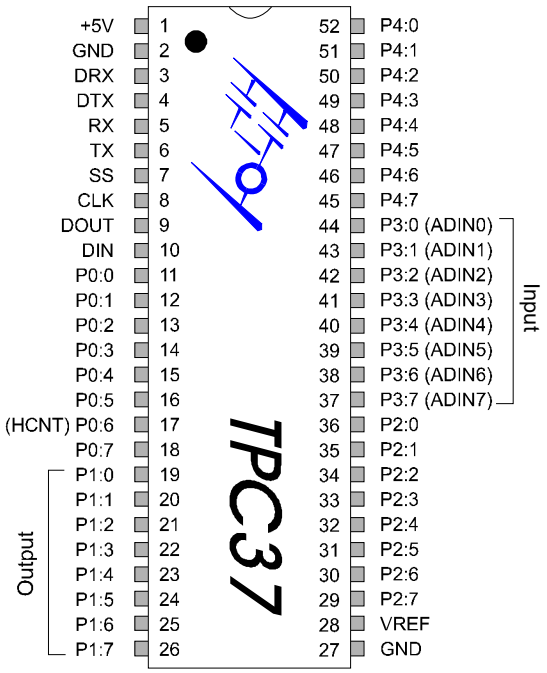
핀번호	명칭	입출력*	기능설명
1	+5V	Power	전원입력 단자 (4.5V~5.5V 사이의 전압을 공급)
2	GND	Power	그라운드 단자
3	DRX	Input	다운로드용 RX 단자 (PC 의 RS232C-TD 단과 접속)
4	DTX	Output	다운로드용 TX 단자 (PC 의 RS232C-RD 단과 접속)
5	RX	Input	범용통신 RX 단자 (Host 의 TD 와 접속)
6	TX	Output	범용통신 TX 단자 (Host 의 RD 와 접속)
7	485TE	Output	RS485 송신허가 신호
8	CLK	Output	확장용 (Clock 신호)
9	Dout	Output	확장용 (Data Output 신호)
10	Din	Input	확장용 (Data Input 신호)
11	P0:0	I/O	입출력 포트
12	P0:1	I/O	입출력 포트 (고속 카운터 입력포트)
13	P0:2	I/O	입출력 포트
14	P0:3	I/O	입출력 포트
15	P0:4	I/O	입출력 포트
16	P0:5	I/O	입출력 포트
17	P0:6	I/O	입출력 포트
18	P0:7	I/O	입출력 포트
19	P1:7	I/O	입출력 포트
20	P1:6	I/O	입출력 포트
21	P1:5	I/O	입출력 포트

22	P1:4	I/O	입출력 포트
23	P1:3	I/O	입출력 포트
24	P1:2	I/O	입출력 포트
25	P1:1	I/O	입출력 포트
26	P1:0	I/O	입출력 포트
27	P2:7	I/O	입출력 포트 (아날로그 Ch7 입력포트)
28	P2:6	I/O	입출력 포트 (아날로그 Ch6 입력포트)
29	P2:5	I/O	입출력 포트 (아날로그 Ch5 입력포트)
30	P2:4	I/O	입출력 포트 (아날로그 Ch4 입력포트)
31	P2:3	I/O	입출력 포트 (아날로그 Ch3 입력포트)
32	P2:2	I/O	입출력 포트 (아날로그 Ch2 입력포트)
33	P2:1	I/O	입출력 포트 (아날로그 Ch1 입력포트)
34	P2:0	I/O	입출력 포트 (아날로그 Ch0 입력포트)
35	Vref	I	아날로그 기준 전압 입력
36	N/C		연결 없음

* 회색부분 (핀번호 8,9,10)은 사용하지 않는 핀입니다.
* 장착시 1 번핀의 위치에 주의하시기 바랍니다.

TPC 37 핀아웃

다음은 TPC37 모델의 핀 아웃 (PIN OUT)에 대한 설명입니다.



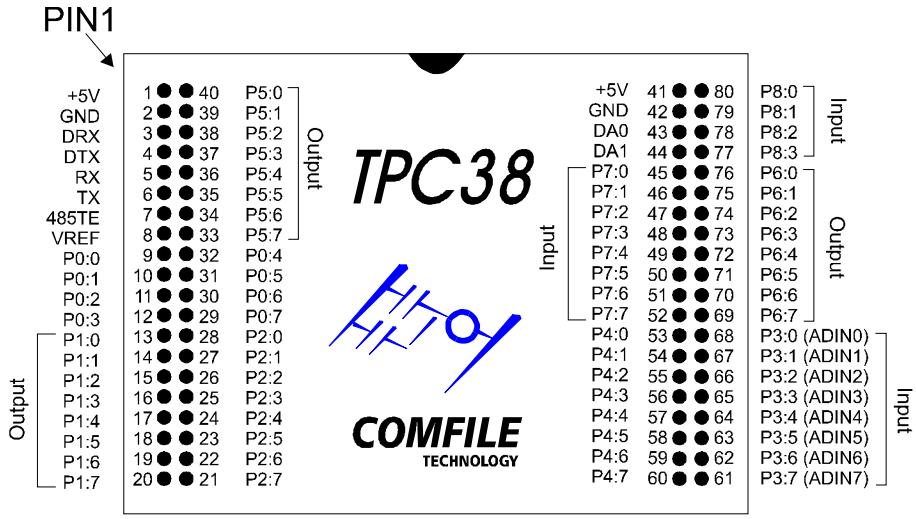
핀번호	명칭	입출력*	기능설명
1	+5V	Power	전원입력 단자 (4.5V~5.5V 사이의 전압을 공급)
2	GND	Power	그라운드 단자
3	DRX	Input	다운로드용 RX 단자 (PC 의 RS232C-TD 단과 접속)
4	DTX	Output	다운로드용 TX 단자 (PC 의 RS232C-RD 단과 접속)
5	RX	Input	범용통신 RX 단자 (Host 의 TD 와 접속)
6	TX	Output	범용통신 TX 단자 (Host 의 RD 와 접속)
7	485TE	Output	RS485 송신허가 신호
8	CLK	Output	확장용 (Clock 신호)
9	Dout	Output	확장용 (Data Output 신호)
10	Din	Input	확장용 (Data Input 신호)
11	P0:0	I/O	입출력 포트
12	P0:1	I/O	입출력 포트
13	P0:2	I/O	입출력 포트
14	P0:3	I/O	입출력 포트
15	P0:4	I/O	입출력 포트
16	P0:5	I/O	입출력 포트
17	P0:6	I/O	입출력 포트 (고속 카운터 입력포트)
18	P0:7	I/O	입출력 포트
19	P1:0	Output	출력 전용 포트

20	P1:1	Output	출력 전용 포트
21	P1:2	Output	출력 전용 포트
22	P1:3	Output	출력 전용 포트
23	P1:4	Output	출력 전용 포트
24	P1:5	Output	출력 전용 포트
25	P1:6	Output	출력 전용 포트
26	P1:7	Output	출력 전용 포트
27	GND	Power	그라운드 단자
28	VREF	Input	A/D 용 기준전압 입력 (0V~VREF 사이의 전압을 AD 변환 한다. 최대 5V 까지 입력가능)
29	P2:7	I/O	입출력 포트
30	P2:6	I/O	입출력 포트
31	P2:5	I/O	입출력 포트
32	P2:4	I/O	입출력 포트
33	P2:3	I/O	입출력 포트
34	P2:2	I/O	입출력 포트
35	P2:1	I/O	입출력 포트
36	P2:0	I/O	입출력 포트
37	P3:7	Input	입력 전용 포트 (아날로그 Ch7 입력포트)
38	P3:6	Input	입력 전용 포트 (아날로그 Ch6 입력포트)
39	P3:5	Input	입력 전용 포트 (아날로그 Ch5 입력포트)
40	P3:4	Input	입력 전용 포트 (아날로그 Ch4 입력포트)
41	P3:3	Input	입력 전용 포트 (아날로그 Ch3 입력포트)
42	P3:2	Input	입력 전용 포트 (아날로그 Ch2 입력포트)
43	P3:1	Input	입력 전용 포트 (아날로그 Ch1 입력포트)
44	P3:0	Input	입력 전용 포트 (아날로그 Ch0 입력포트)
45	P4:7	I/O	입출력 포트
46	P4:6	I/O	입출력 포트
47	P4:5	I/O	입출력 포트
48	P4:4	I/O	입출력 포트
49	P4:3	I/O	입출력 포트
50	P4:2	I/O	입출력 포트
51	P4:1	I/O	입출력 포트
52	P4:0	I/O	입출력 포트

* 회색부분 (핀번호 8,9,10)은 사용하지 않는 핀입니다.
* 장착시 1 번핀의 위치에 주의하시기 바랍니다.

TPC 38 핀아웃

다음은 TPC38 모델의 핀 아웃 (PIN OUT)에 대한 설명입니다.



핀번호	명칭	입출력*	기능설명
1	+5V	Power	전원입력 단자 (4.5V~5.5V 사이의 전압을 공급)
2	GND	Power	그라운드 단자
3	DRX	Input	다운로드용 RX 단자 (PC 의 RS232C-TD 단과 접속)
4	DTX	Output	다운로드용 TX 단자 (PC 의 RS232C-RD 단과 접속)
5	RX	Input	범용통신 RX 단자 (Host 의 TD 와 접속)
6	TX	Output	범용통신 TX 단자 (Host 의 RD 와 접속)
7	485TE	Output	RS485 송신허가 신호
8	VREF	Input	A/D 용 기준전압 입력 (0V ~ VREF 사이의 전압을 A/D 변환한다. 최대 5V 까지 입력가능)
9	P0:0	I/O	입출력 포트
10	P0:1	I/O	입출력 포트
11	P0:2	I/O	입출력 포트
12	P0:3	I/O	입출력 포트
13	P1:0	Output	출력 전용 포트
14	P1:1	Output	출력 전용 포트
15	P1:2	Output	출력 전용 포트
16	P1:3	Output	출력 전용 포트
17	P1:4	Output	출력 전용 포트
18	P1:5	Output	출력 전용 포트
19	P1:6	Output	출력 전용 포트
20	P1:7	Output	출력 전용 포트
21	P2:7	I/O	입출력 포트

22	P2:6	I/O	입출력 포트
23	P2:5	I/O	입출력 포트
24	P2:4	I/O	입출력 포트
25	P2:3	I/O	입출력 포트
26	P2:2	I/O	입출력 포트
27	P2:1	I/O	입출력 포트
28	P2:0	I/O	입출력 포트
29	P0:7	I/O	입출력 포트
30	P0:6	I/O	입출력 포트 (고속 카운터 입력포트)
31	P0:5	I/O	입출력 포트
32	P0:4	I/O	입출력 포트
33	P5:7	Output	출력 전용 포트
34	P5:6	Output	출력 전용 포트
35	P5:5	Output	출력 전용 포트
36	P5:4	Output	출력 전용 포트
37	P5:3	Output	출력 전용 포트
38	P5:2	Output	출력 전용 포트
39	P5:1	Output	출력 전용 포트
40	P5:0	Output	출력 전용 포트
41	+5V	Power	전원입력 단자 (4.5V~5.5V 사이의 전압을 공급)
42	GND	Power	그라운드 단자
43	DA0	Output	DA 출력 전용 포트 0 (PWM 출력, 항상 출력이 나옴)
44	DA1	Output	DA 출력 전용 포트 1 (PWM 출력, 항상 출력이 나옴)
45	P7:0	Input	입력 전용 포트
46	P7:1	Input	입력 전용 포트
47	P7:2	Input	입력 전용 포트
48	P7:3	Input	입력 전용 포트
49	P7:4	Input	입력 전용 포트
50	P7:5	Input	입력 전용 포트
51	P7:6	Input	입력 전용 포트
52	P7:7	Input	입력 전용 포트
53	P4:0	I/O	입출력 포트
54	P4:1	I/O	입출력 포트
55	P4:2	I/O	입출력 포트
56	P4:3	I/O	입출력 포트
57	P4:4	I/O	입출력 포트
58	P4:5	I/O	입출력 포트
59	P4:6	I/O	입출력 포트
60	P4:7	I/O	입출력 포트
61	P3:7	Input	입력 전용 포트 (아날로그 Ch7 입력포트)
62	P3:6	Input	입력 전용 포트 (아날로그 Ch6 입력포트)
63	P3:5	Input	입력 전용 포트 (아날로그 Ch5 입력포트)
64	P3:4	Input	입력 전용 포트 (아날로그 Ch4 입력포트)
65	P3:3	Input	입력 전용 포트 (아날로그 Ch3 입력포트)
66	P3:2	Input	입력 전용 포트 (아날로그 Ch2 입력포트)

67	P3:1	Input	입력 전용 포트 (아날로그 Ch1 입력포트)
68	P3:0	Input	입력 전용 포트 (아날로그 Ch0 입력포트)
69	P6:7	Output	출력 전용 포트
70	P6:6	Output	출력 전용 포트
71	P6:5	Output	출력 전용 포트
72	P6:4	Output	출력 전용 포트
73	P6:3	Output	출력 전용 포트
74	P6:2	Output	출력 전용 포트
75	P6:1	Output	출력 전용 포트
76	P6:0	Output	출력 전용 포트
77	P8:3	Input	입력 전용 포트
78	P8:2	Input	입력 전용 포트
79	P8:1	Input	입력 전용 포트
80	P8:0	Input	입력 전용 포트

* 장착시 1 번핀의 위치에 주의하시기 바랍니다.

TPC38 에는 포트를 정리하면 다음과 같습니다.

입출력 겸용 포트 : 24 개 ..... (P0, P2, P4)
입력 전용포트 : 20 개 .....(P3, P7, P8)
출력 전용포트 : 24 개.....(P1, P5, P6)
DA 출력 포트 : 2 개
총 : 70 개

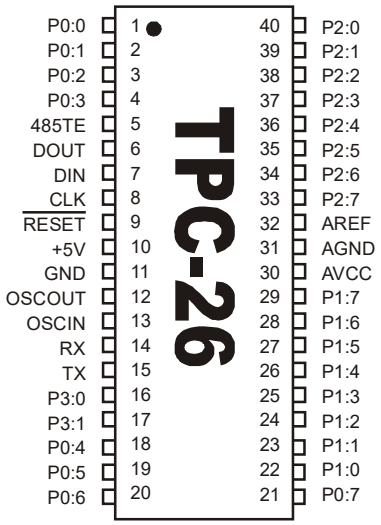
주의사항

TPC38 의 P5 ~ P8 포트에서 다음명령에서 사용할 수 없습니다.

LCDOUT, SGNOUT, THIN, KEYSCAN

TPC26 의 핀아웃

TPC26 은 다른 TPC3X 모듈과는 달리 실제 "반도체 칩"으로 되어 있는 모델입니다.



핀 번호	명칭	입출력	기능 설명
1	P0:0	I/O	입출력 포트
2	P0:1	I/O	입출력 포트 (고속 카운터 입력포트)
3	P0:2	I/O	입출력 포트
4	P0:3	I/O	입출력 포트
5	485TE	Output	RS485 송신 허가 신호
6	DOUT	Output	확장용 (DATA OUTPUT)신호
7	DIN	Input	확장용 (DATA INPUT)신호
8	CLK	Output	확장용 (CLOCK)신호
9	/RESET	Input	리셋 신호 (Low 인가시 칩이 리셋됨, 평상시는 High 인가)
10	+5V	Power	전원 입력단자 (4.5V~5.5V 사이의 전압을 공급)
11	GND	Power	그라운드 단자
12	OSCOUT	Output	발진소자 연결단자
13	OSCIN	Input	발진소자 연결단자 (7.3728MHz 크리스털을 연결)
14	RX	Input	다운로드 및 RS232/485 통신 연결포트 (19200 보레이트)
15	TX	Output	다운로드 및 RS232/485 통신 연결포트 (19200 보레이트)
16	P3:0	I/O	입출력 포트
17	P3:1	I/O	입출력 포트
18	P0:4	I/O	입출력 포트
19	P0:5	I/O	입출력 포트
20	P0:6	I/O	입출력 포트

21	P0:7	I/O	입출력 포트
22	P1:0	I/O	입출력 포트
23	P1:1	I/O	입출력 포트
24	P1:2	I/O	입출력 포트
25	P1:3	I/O	입출력 포트
26	P1:4	I/O	입출력 포트
27	P1:5	I/O	입출력 포트
28	P1:6	I/O	입출력 포트
29	P1:7	I/O	입출력 포트
30	AVCC	Power	아날로그용 전원 입력단자
31	AGND	Power	아날로그용 그라운드 입력단자
32	AREF	Input	아날로그 입력 기준전압 입력단자 3V 입력시, 0~3V 까지 변환, 5V 입력시 0~5V 까지 변환됨
33	P2:7	I/O	입출력 포트 (아날로그 CH7 입력포트)
34	P2:6	I/O	입출력 포트 (아날로그 CH6 입력포트)
35	P2:5	I/O	입출력 포트 (아날로그 CH5 입력포트)
36	P2:4	I/O	입출력 포트 (아날로그 CH4 입력포트)
37	P2:3	I/O	입출력 포트 (아날로그 CH3 입력포트)
38	P2:2	I/O	입출력 포트 (아날로그 CH2 입력포트)
39	P2:1	I/O	입출력 포트 (아날로그 CH1 입력포트)
40	P2:0	I/O	입출력 포트 (아날로그 CH0 입력포트)

- P 로 시작되는 핀들은 모두 I/O 핀입니다. 유저정의에 따라 Input 또는 Output 핀으로 사용할 수 있습니다.
- CLKIN 단자에서는 하드웨어적으로 라이징 에지를 검출하므로 20KHz 정도의 고속 펄스도 카운트 할 수 있습니다.
- OSCIN, OSCOUT 에는 **반드시 7.3728MHz 크리스털을** 연결해야 합니다.
- /RESET 단자는 별도의 추가회로 없이 5V 와 직접 연결해야 합니다.

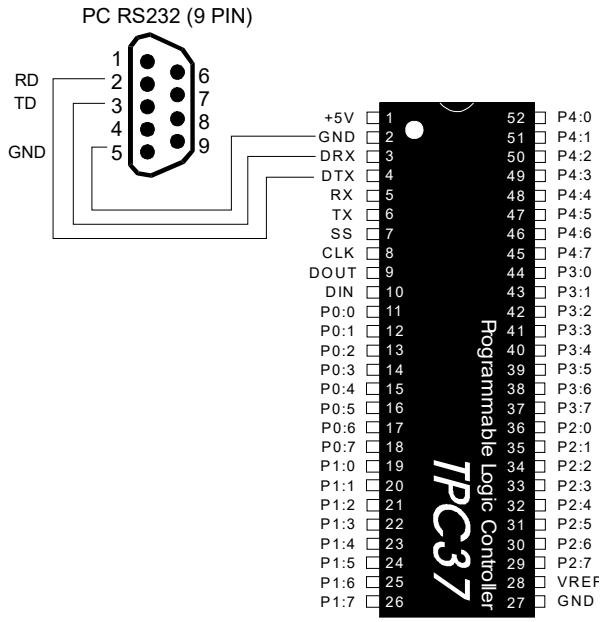
주의사항

- TPC26 칩을 일반적인 롬라이터로 읽거나 지우기를 수행하면 동작에 치명적인 영향을 주게 되며, 칩을 못쓰게 되는 되는 경우도 있습니다. 이러한 작업은 하지 않는 것이 좋습니다.
- TPC26 칩은 칩상태로 출하되므로 정전기에 매우 취약합니다. 이동시나 보관시에는 정전기 보호가 가능한 팩케지에 보관하시기 바랍니다.
- 전원 (+5V, GND) 단에 가장 가까운 쪽에 반드시 0.1uF 정도의 "바이패스 콘덴서"를 부착하기 바랍니다. 바이패스 콘덴서는 칩의 안정적인 동작을 필수적인 부품입니다. (세라믹 또는 모놀리딕 타입의 콘덴서)
- **TPC26 의 데이터 통신은 19200 보레이트 입니다.** (기존 TPC 3X 는 9600 보레이트임) 데이터 통신은 다운로드 포트와 동일합니다.

다운로드용 RS232C 케이블 규격

TPC33/37/38 모델의 핀 3,4 번의 DRX 와 DTX 는 다운로드및 모니터링시 사용되는 통신 포트입니다. PC 의 RS232C 포트의 RD, TD 단자와 별도의 변환장치 없이 직접 접속합니다.

TPC3X 의 핀	PC 측의 RS232C (9 핀 사용시)	PC 측의 RS232C (25 핀 사용시)
2 (GND)	5 (GND)	7 (GND)
3 (DRX)	3 (TD)	2 (TD)
4 (DTX)	2 (RD)	3 (RD)

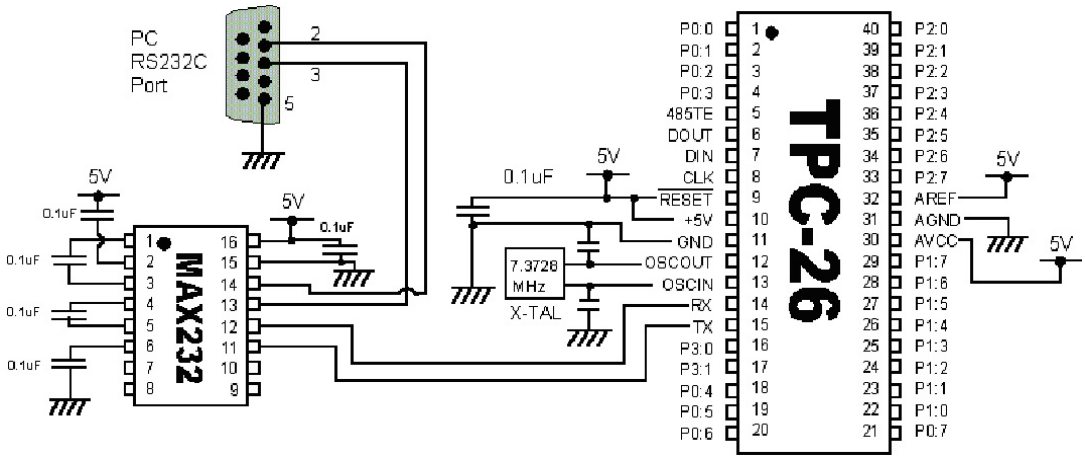


통신 프로토콜은 RS232C, 19200 보레이트, N,8,1 입니다. 다운로드용 포트 (핀 3,4 번)는 TPC 모듈안에 레벨 변환 회로가 내장되어 있으므로, RS232C 라인과 직접 접속할 수 있습니다.

*** 주의사항 : 다운로드용 케이블의 길이는 최대 2 미터까지 가능하며, 운전시 에는 반드시 분리시켜야 합니다. (불필요한 노이즈가 유입될 수 있기 때문입니다.)**

TPC26 다운로드를 위한 최소한의 회로구성

TPC26 의 경우에는 다른 모듈과 조금은 다른 구성으로 PC 와 접속해야 합니다. 다음은 TPC26 을 동작시키기 위한 최소한의 회로 구성입니다.



PC 의 RS232C 포트와는 MAX232 라는 레벨 변환칩을 통해서 연결합니다. (대부분의 일체형 PLC 와 베이스보드에는 이러한 회로가 내장되어 있으므로, 유저는 단지 케이블만 연결하는 것만으로 곧바로 사용가능 합니다.) TPC26 은 반도체 타입이므로 가장 기본적인 발진회로와 전원회로,리셋회로를 연결해주어야 동작 가능한 상태가 됩니다.

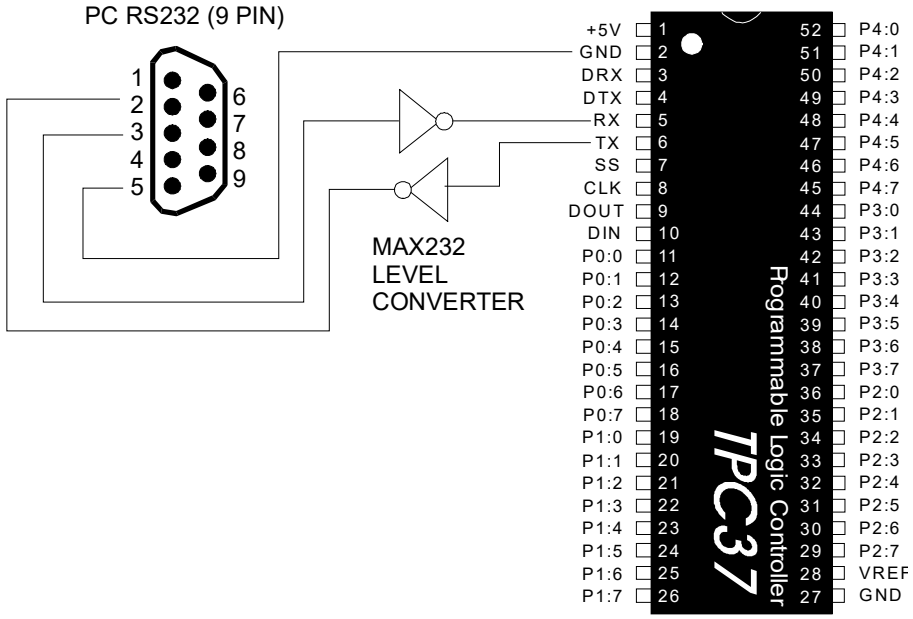
TPC26 용 기본 베이스 보드 BASE-F24R 은 위의 회로도와 동일한 구성으로 되어 있으므로, 편리하게 이용할 수 있습니다.

컴퓨터와의 링크시

TPC3X 는 컴퓨터와의 링크 기능을 가지고 있으며, RS232 (또는 RS485)를 통하여 TPC3X 상의 데이터 메모리 내용을 읽거나 쓸 수 있습니다. RS485 를 사용하면 멀티드롭 방식으로 데이터를 교환할 수 있습니다. 핀 5,6 번의 RX 와 TX 는 이러한 범용통신에 사용되는 통신 포트입니다. (프로토콜은 9600 보레이트, N, 8, 1 로 고정되어 있습니다.)

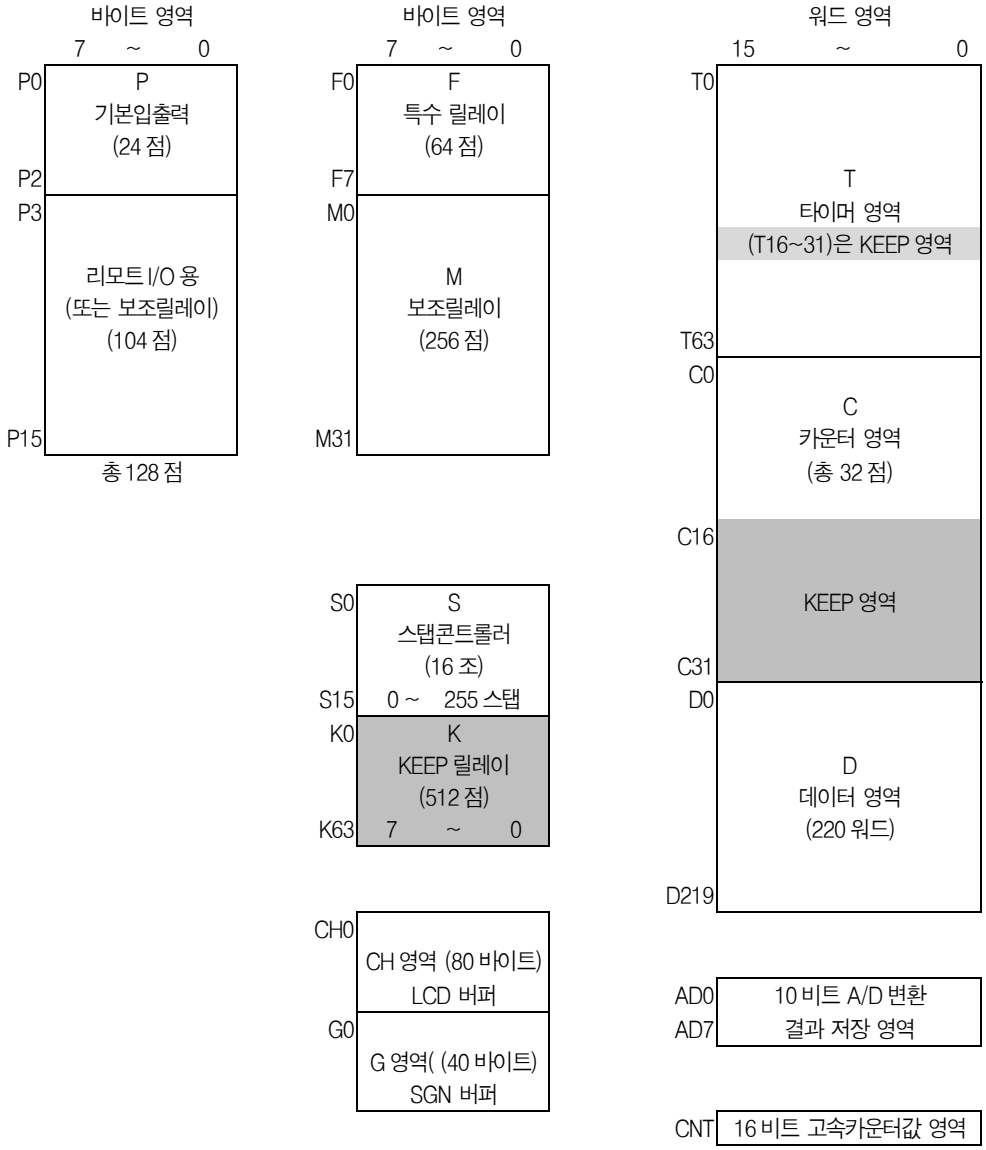
TPC3X 의 핀	PC 측의 RS232C (9 핀 사용시)	PC 측의 RS232C (25 핀 사용시)
2 (GND)	5 (GND)	7 (GND)
5 (RX)	3 (TD)	2 (TD)
6 (TX)	2 (RD)	3 (RD)

범용 통신용 포트 (핀 5, 6 번) 는 레벨 변환회로가 내장되어 있지 않으므로, 반드시 아래 그림과 같이 레벨 변환용 칩을 통해서 5V 레벨의 신호를 입력해주어야 합니다.
(베이스보드에는 아래의 RS232C 용 레벨 변환 칩이 포함되어 있습니다.)



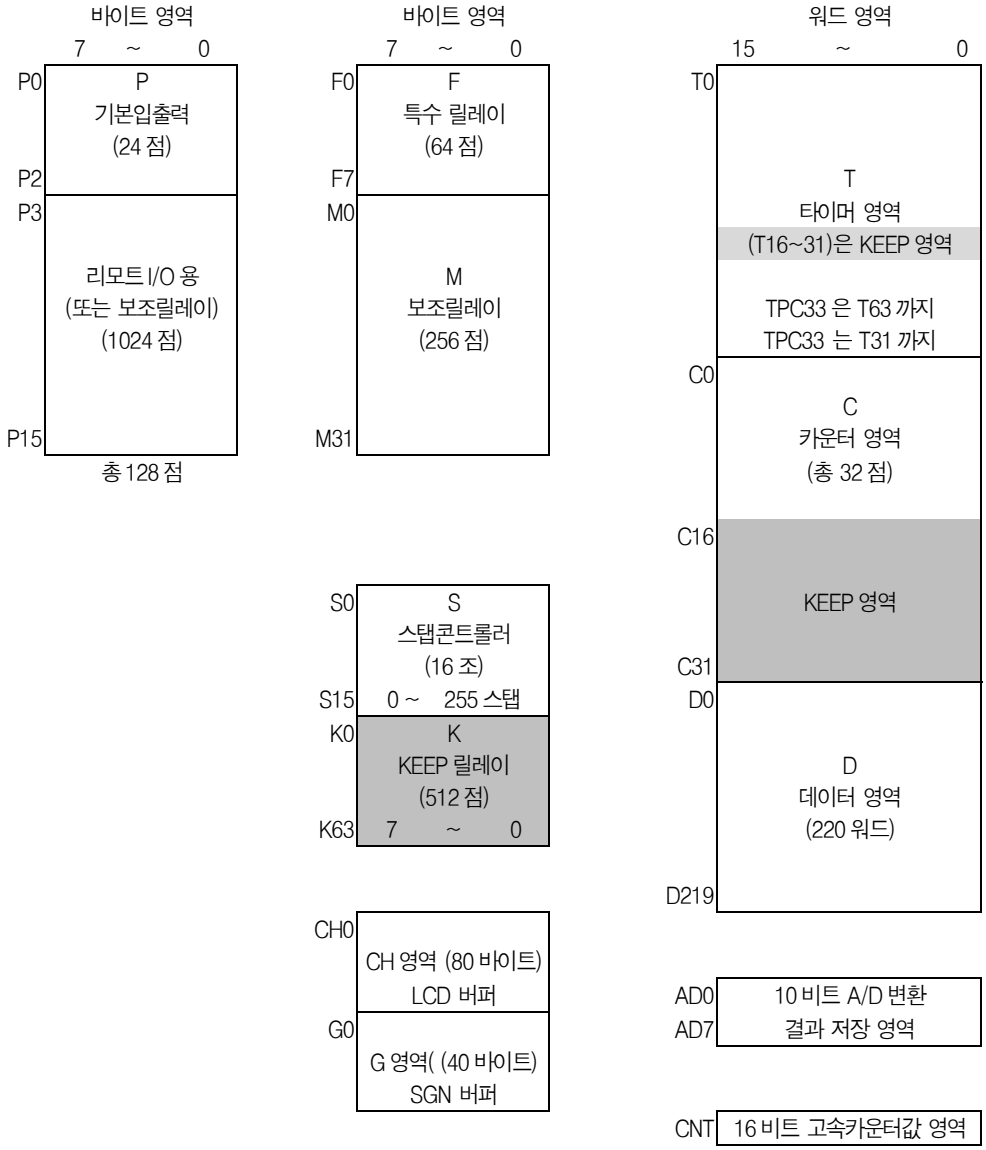
단, TPC26 의 경우에는 다운로드포트를 통하여 데이터통신도 같이 수행합니다. 데이터 통신에 대한 자세한 설명은 제 6 장 데이터 통신편에서 다룹니다.

TPC26 데이터 메모리 맵



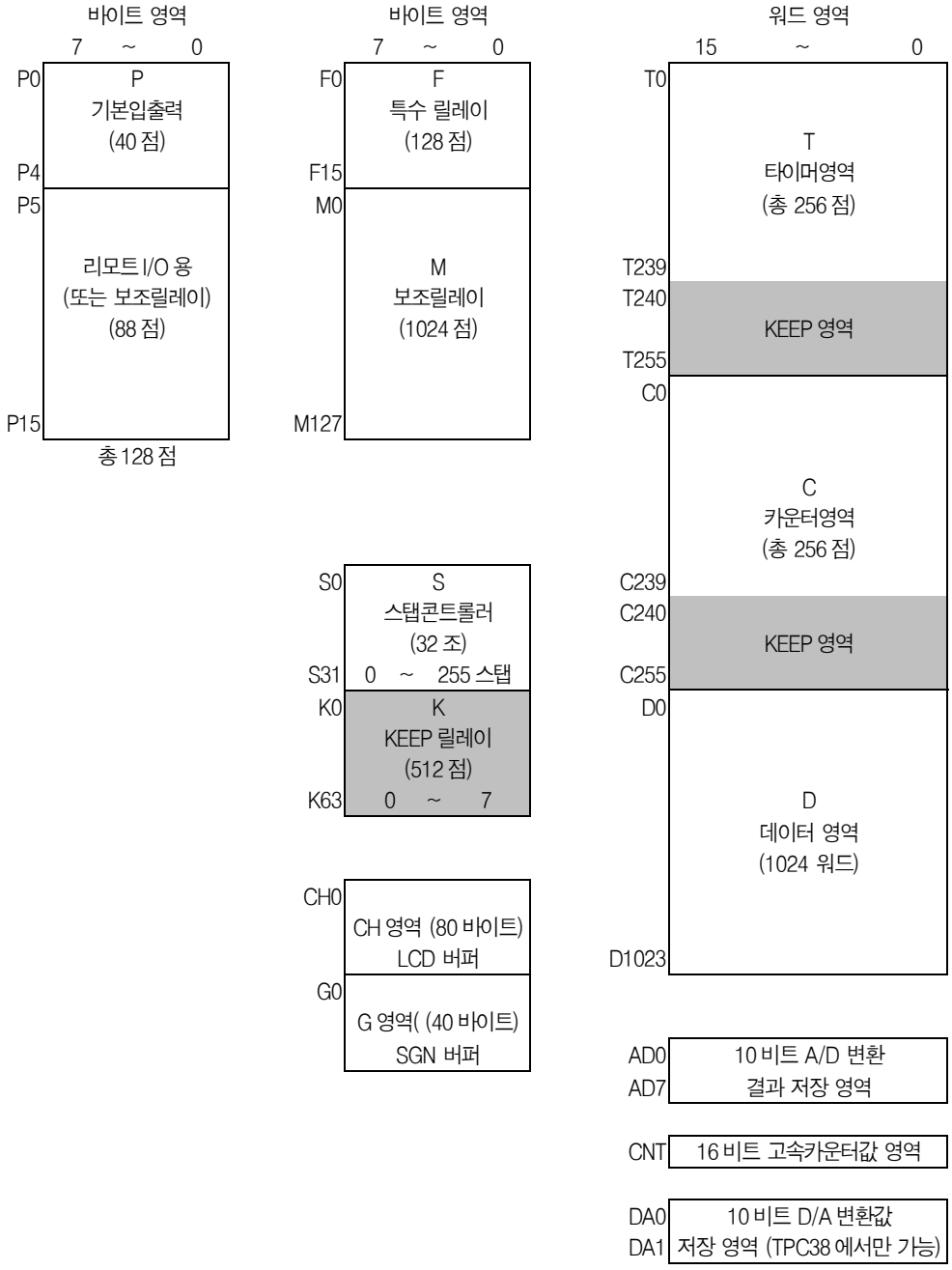
- 타이머/카운터 영역중 일부는 정전시에도 내용을 보존할 수 있는 KEEP 영역으로 되어 있습니다.
- 타이머/카운터 영역은 16 비트 (1 워드) 단위로 되어있으며, 다른 영역 (P,M,K,CH,G)은 바이트 단위로 되어 있습니다.
- 워드단위 영역은 데이터를 상위/하위 바이트 순으로 저장합니다. (낮은 번지에 상위바이트가 들어감)
- KEEP 영역을 제외한 모든 영역은 파워 ON 시 0 이 됩니다.
- KEEP 영역은 EEPROM 에 기록되므로 별도의 밧데리 백업이 필요 없습니다.

TPC33 데이터 메모리 맵



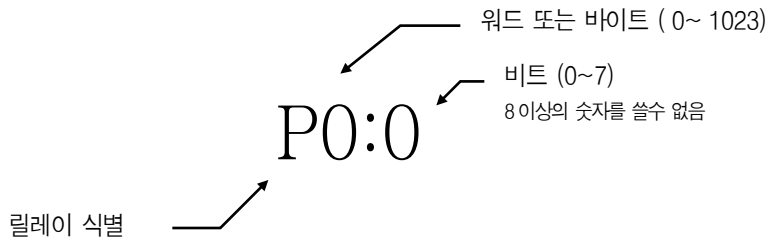
- 타이머/카운터 영역 중 일부는 정전 시에도 내용을 보존할 수 있는 KEEP 영역으로 되어 있습니다.
- 타이머/카운터 영역은 16 비트 (1 워드) 단위로 되어있으며, 다른 영역 (P,M,K,CH,G)은 바이트 단위로 되어 있습니다.
- 워드단위 영역은 데이터를 상위/하위 바이트 순으로 저장합니다. (낮은 번지에 상위바이트가 들어감)
- 모든 영역은 파워 ON 시 0 이 됩니다.
- KEEP 영역은 EEPROM 에 기록되므로 별도의 밧데리 백업이 필요 없습니다.

TPC37, 38 데이터 메모리 맵



릴레이 표현

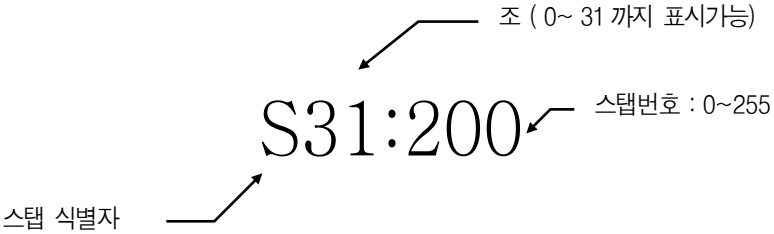
TPC3X 에서의 모든 릴레이는 다음과 같은 형식으로 표현합니다.



릴레이 종류	릴레이 식별	워드 표현가능 범위	비트 표현가능 범위	기능
입출력 릴레이	P	0~15	0~7	외부에 나와있는 I/O 포트를 ON/OFF 하거나, 핀의 상태를 읽어옵니다.
특수릴레이	F	0~15	0~7	
(내부)보조릴레이	M	0~127	0~7	
KEEP 릴레이	K	0~31	0~7	정전시에도 그전상태를 보존합니다.
타이머	T	0~255	작성안함	타이머 접점
카운터	C	0~255	작성안함	카운터 접점
데이터	D	0~1023	작성안함	데이터 영역

워드(또는 바이트)단위로 처리하고자 할 때에는 워드단위까지만 사용하면 됩니다. 전송명령등에서 바이트 단위로 전송하고자 할 때 사용하면 편리합니다. (예 : MOVE P0, P2)

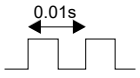
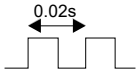
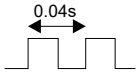
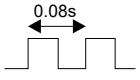
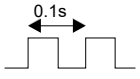
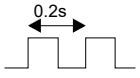
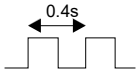
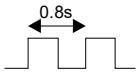
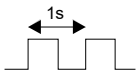
스탭 콘트롤에 사용되는 S 릴레이는 다음과 같이 표현합니다.

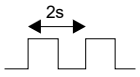
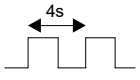
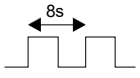
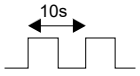
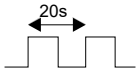
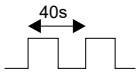
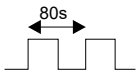


릴레이 종류	릴레이 식별	워드 표현가능 범위	비트 표현가능 범위	기능
스탭콘트롤 릴레이	S	0~31	0~255	스탭 콘트롤 (32 조 256 스탭 가능)

특수릴레이

특수릴레이는 프로그램 작성시 편리하게 이용할 수 있는 신호와 각종 모드 결정, CPU 의 상태등을 취급하는 릴레이입니다.

특수	릴레이	기능	비고
F0	F0:0	상시 OFF	
	F0:1	상시 ON	
	F0:2	최초 전원 투입시 1 스캔동안 ON	
	F0:3	최초 전원 투입시 1 스캔동안 OFF	
	F0:4		
	F0:5		
	F0:6		
	F0:7		
	F1:7		
F1		사용안함	
F2	F2:0	0.01 초 주기로 펄스 발생 (5mS on : 5mS off) 1 스캔타임 간격으로 on /off 를 반복	
	F2:1	0.02 초 주기로 펄스 발생 (10mS on : 10mS off)	
	F2:2	0.04 초 주기로 펄스 발생 (20mS on : 20mS off)	
	F2:3	0.08 초 주기로 펄스 발생 (40mS on : 40mS off)	
F3	F3:0	0.1 초 주기로 펄스 발생 (0.05 초 on : 0.05 초 off)	
	F3:1	0.2 초 주기로 펄스 발생 (0.1 초 on : 0.1 초 off)	
	F3:2	0.4 초 주기로 펄스 발생 (0.2 초 on : 0.2 초 off)	
	F3:3	0.8 초 주기로 펄스 발생 (0.4 초 on : 0.4 초 off)	
	F4:0	1 초 주기로 펄스 발생 (0.5 초 on : 0.5 초 off)	

F4	F4:1	2 초 주기로 펄스 발생 (1 초 on : 1 초 off)	
	F4:2	4 초 주기로 펄스 발생 (2 초 on : 2 초 off)	
	F4:3	8 초 주기로 펄스 발생 (4 초 on : 4 초 off)	
F5	F5:0	10 초 주기로 펄스 발생 (5 초 on : 5 초 off)	
	F5:1	20 초 주기로 펄스 발생 (10 초 on : 10 초 off)	
	F5:2	40 초 주기로 펄스 발생 (20 초 on : 20 초 off)	
	F5:3	80 초 주기로 펄스 발생 (40 초 on : 40 초 off)	
F6		한글 LCD 사용시 한글의 폰트를 결정하는 레지스터 0 : SMALL 샘물체 1 : SMALL 고딕체 2 : BIG 명조체 3 : BIG 태고딕체	WRITE 용
F7		RS485 통신용 어드레스 0~ 255 까지 지정가능	
F8		실시간으로 동작되는 초 0 ~ 59 까지만 동작함 F8:0 은 1 초 ON / 1 초 OFF 동작	TPC33 은 사용불가
F9		분 (0~ 59 까지만 동작)	TPC33 은 사용불가
F10		시간 (0~255 까지 동작, 255 를 넘으면 다시 0 이 됨)	TPC33 은 사용불가

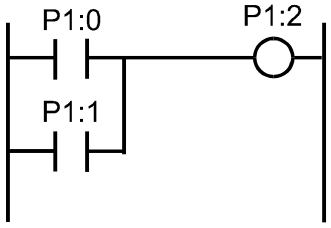
리얼타임 클록에 관하여...

F8,F9, F10 에 위치한 RTC (리얼타임클록)은 TPC37, TPC38 에서만 동작 가능합니다. RTC 는 파워온시 자동으로 0 으로 클리어되며, 파워 OFF 시 유지되는 기능은 없습니다. 동작 중 정확한 시간을 필요로 하는 경우에 사용할 수 있습니다.

제 2 장
PLC 기초

PLC 란 무엇인가?

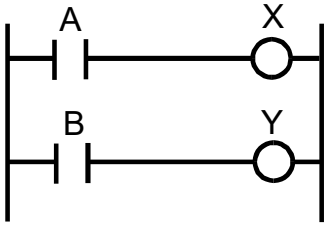
현재, PLC (Programmable Logic Controller)는 산업용 컨트롤러로 확고한 자리를 잡고 있으며 폭넓은 유저층을 확보하고 있는 대표적인 공장자동화용 컨트롤러입니다. 릴레이 레더 다이어그램 (Relay Ladder Diagram)에 의한 프로그램은, 이해하기 쉬운 그림 형태로 되어 있어서, 전자/전산을 전공하지 않은, 기계, 전기 기술자들도 쉽게 배울 수 있고, 많은 실전 경험이 없어도 단기간에 응용할 수 있는 프로그램 언어입니다. PLC 는 이러한 릴레이 레더 다이어그램으로 움직일 수 있는 컨트롤러라고 할 수 있습니다.



<릴레이 레더 다이어그램의 예>

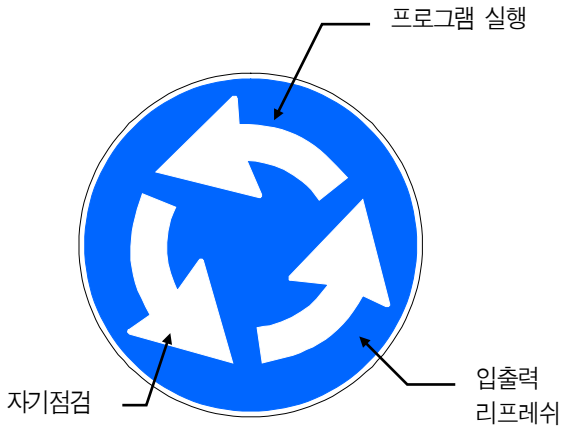
C, 어셈블리, BASIC 과 같은 일반 프로그래밍 언어와의 가장 큰 차이점은 바로 "멀티 테스킹"입니다. (멀티 테스킹; 동시에 여러가지 일을 수행하는 것을 의미함) C 나 어셈블리와 같은 언어는 시퀀셜 (Sequential)하게 프로그램을 작성해나가는 방법으로 프로그래밍하는 언어입니다. 이는 어느 한순간에 한가지 동작밖에 처리할 수 없다는 것을 의미하기도 합니다.

하지만 PLC 에서의 레더 다이어그램은 기본적으로 멀티 테스킹을 지원하고 있기 때문에, 유저가 이 문제로 고민할 필요는 없습니다. 아래 그림에서처럼 A 입력이 들어오면 X 가 ON 되고, 동시에 B 입력이 들어오면 Y 가 ON 되는 구조로 되어 있습니다.



이와같은 레더 프로그램이 아무리 복잡하게 연결되어 있어도, 레더상에 있는 입력접점과 릴레이등은 항상 동작가능한 상태로 준비하고 있으며, 입력이 있을 때 바로 상응하는 동작을 하도록 되어 있습니다. 이 때문에 PLC 에는 스캔 타임 (SCAN TIME)이라는 개념이 존재하고 있습니다. 스캔 타임이란 "릴레이 레더"프로그램의 처음부터 끝까지 실행하는 시간을 의미하며, 일정한 스캔타임 간격으로 릴레이 레더 프로그램은 반복적으로 실행됩니다.

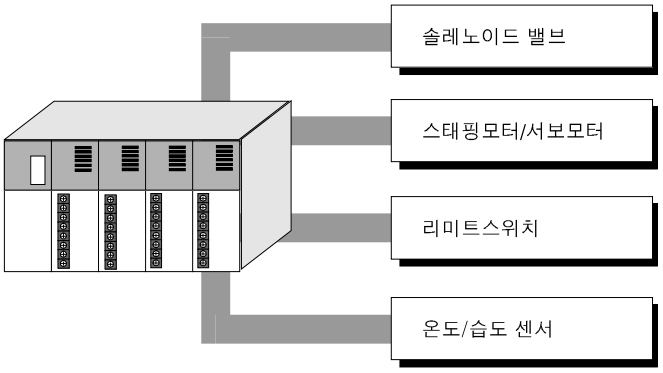
스캔 타임중에는 릴레이 레더 프로그램의 해석/실행 이외에도 입출력 리프레쉬, 자기점검등의 일을 수행하는데, TPC3X 는 5mS 의 고정 스캔타임 방식을 도입하고 있습니다. 이것은 프로그램 길이와 상관없이 항상 같은 스캔 타임을 유지하는 방식으로, 프로그램 길이에 따라 스캔타임이 가변되는 방식에 비해, 효율적으로 시간을 관리할 수 있는 이점이 있습니다. (고정 스캔타임은 기종에 따라 변경될 수 있습니다.)



릴레이 레더 프로그램을 사용할 때 스캔타임을 크게 의식할 필요는 없지만, 기본적으로 스캔 타임이라는 골격 위에 프로그램이 놓인다는 사실을 염두 해두어야 할 것입니다. 바로 이점이 C 나 어셈블러와 같은 순차적 기술언어와의 가장 큰 차이점입니다.

PLC 가 하는일은?

다른 프로세서 (원칩 마이컴, 원보드 컴퓨터, PC)와 마찬가지로 PLC 가 하는 일은 여러가지 주변기기를 중앙에서 제어 및 관리, 운용하는 역할을 수행합니다. 다른 프로세서와 구별되는 점이 있다면, 설비 및 생산자동화 분야에서 사용되는 릴레이 배전반의 대체용으로 각종 생산라인의 주변장치들을 효율적으로 제어할 수 있는 기능들을 내장하고 있다는 것입니다. 예를들면 컨베이어의 총괄제어, 각종 근접센서와 온도센서와 솔레노이드 벨브, 모터등을 접속한 뒤 일정한 시퀀스로 움직이게 하는 등의 작업을 수행합니다.



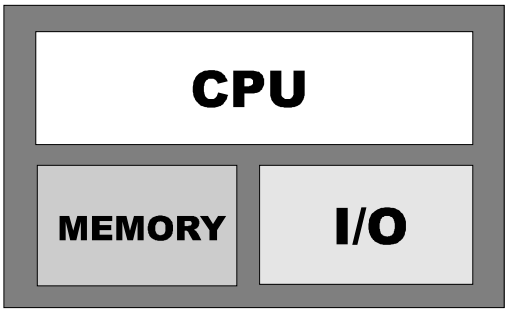
과거, 릴레이 배전반의 대체용으로 제작된 PLC 는 가장 기본적인 기능 (입출력 제어, 타이머, 카운터 기능) 이외에도 여러가지 다양한 기능을 내장 시켜 현대에는 종합적인 공장 자동화용 컨트롤러로 각광받고 있습니다. 지금은 PLC 를 빼놓고 자동화를 말할 수 없을 정도로 PLC 의 영향력은 가히 지대적이라고 할 수 있습니다. 다음은 PLC 가 적용되는 분야를 요약한 것입니다.

PLC 응용분야

자동창고제어, 반송라인 제어, 산업용 로보트 제어, 공작기계 제어, 송배수 펌프 제어, 각종온도 컨트롤러, 컴프레서 제어, 반도체 제조/검수라인의 로더, 언로더, 산업용 발전기의 메인 컨트롤러, 컨베이어등과 같은 생산라인의 자동 제어, 쓰레기 재처리 시설의 메인 컨트롤러, 공해 방지시스템의 메인 컨트롤러 등등 다수

PLC 의 일반적인 구조

PLC 는 3 가지 기본적인 구성을 가지고 있습니다.



CPU 부는 사람의 머리에 해당하는 부분으로 메모리에 있는 니모닉(명령)을 읽어와서 해석한뒤 실행하는 역할을 수행합니다. **I/O 부**는 사람의 눈, 귀, 손과 발에 해당하는 부분으로 실제로 외부에서 입력을 받아들이고 (눈, 귀,) 출력을 처리 (손과 발)하는 역할을 수행합니다.**메모리**는 기억을 담당하는 부분으로 유저가 작성한 레더 프로그램과 동작중에 발생되는 각종 정보를 이곳에 수록하게 됩니다.

메모리는 RAM, ROM, EEPROM, FLASH ROM 등의 여러가지 종류로 나누어져 있습니다.

- RAM 은 Random Access Memory 라는 뜻으로 자유롭게 읽고 쓸 수 있는 메모리입니다.
- ROM 은 Read Only Memory 라는 뜻으로, 읽기만 가능한 메모리입니다.
- EEPROM 은 전기적으로 소거하거나, 동작 중에 내용을 변경할 수 있고, 전원이 없어도 계속 유지되는 특성을 가지고 있습니다. 반면 RAM 은 전원이 없으면 그 내용이 소실됩니다.
- FLASH ROM 은 동작상으로 EEPROM 과 동일한 특성을 가지고 있지만, 제조방법등에서 차이가 있으므로 EEPROM 과 구분하고 있습니다.

TPC3X 에서 유저가 작성한 레더 다이어그램은 실행 가능한 코드로 번역되어 FLASH ROM 에 기록됩니다. 운영 중 발생되는 각종 정보는 RAM 에 기록되며, 이중에서 전원 OFF 시에도 보존이 요구되는 정보는 별도의 EEPROM 에 기록하게 됩니다. (KEEP 영역)

TPC3X 에서 I/O 부는 가장 기본적인 TTL 레벨* 형태의 I/O 포트로 구성되어 있습니다. (*TTL 레벨이란 0V 를 0 으로 5V 를 1 로 인식하는 방식을 의미합니다.) 따라서 여기에 릴레이, 포토 커플러 등을 부착해서 대용량의 부하를 ON /OFF 할 수 있습니다. 컴파일에서는 이러한 I/O 확장부분만을 "베이스 보드"라는 제품형태로 공급하고 있습니다.

릴레이의 종류

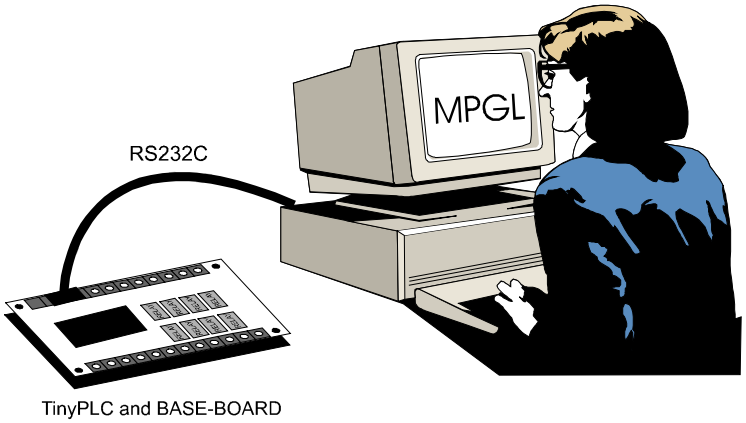
PLC 에서 말하는 릴레이는 외부에 연결하는 부품형태의 릴레이를 의미하기도 하지만, 메모리상으로만 존재하는 내부 릴레이도 릴레이라는 이름으로 통칭하고 있습니다. 다음은 TPC3X 에서 취급하는 릴레이에 대한 설명입니다. 릴레이는 식별자(P, M, K, T, C)로 구분하므로, 식별자를 잘 익혀두시기 바랍니다.

식별	이 름	기 능
P	입출력 릴레이	외부의 릴레이 또는 입력부와 바로 접속되는 릴레이입니다. 입력 접점은 외부 입력포트의 HIGH / LOW 상태를 그대로 전달해줍니다. 출력접점은 외부의 솔레노이드 벨브, 모터등과 연결되어 있으며 출력릴레이를 변경하면 이들의 상태에 그대로 반영됩니다. TPC3X 에는 예비영역을 포함해서 총 128 점의 P 영역이 있으며, 외부 I/O 와 연결된 부분을 제외한 나머지 부분은 REMOTE I/O 등의 목적으로 활용합니다. 만약 REMOTE I/O 를 사용하지 않는다면, 보조릴레이로 사용할 수 있습니다.
M	보조 릴레이	메모리상으로만 존재하는 릴레이로써, 외부에 직접 입출력은 불가능하지만, 정보를 전달해주는 등의 보조적인 역할을 수행합니다. (M 영역은 미분 입력 DF, DFN 용으로 사용할 수 있는 유일한 영역입니다.)
K	킵 릴레이	보조릴레이와 역할과 동일하지만, KEEP 릴레이는 전원이 없는 상태에서도 그 이전 상태를 계속 유지하므로, 정전시 데이터를 유지해야 할 필요가 있을때 사용합니다.
F	특수 릴레이	PLC 의 내부적인 동작상태나 연산의 결과, 각종 시간 정보 등을 표시해주는 일종의 상태 표시용 릴레이입니다. 예를 들면 1 초단위로 ON / OFF 를 반복하는 릴레이 등이 있습니다.
T	타이머	시간을 제어하는 목적으로 사용하는 릴레이입니다. 기본적으로 ON 타이머와 OFF 타이머가 있으며, 10mS 또는 100mS 단위로 시간을 계수하여, 설정 치에 다다르면 접점이 ON 됩니다. 자세한 설명은 명령어 설명을 참조하시기 바랍니다.

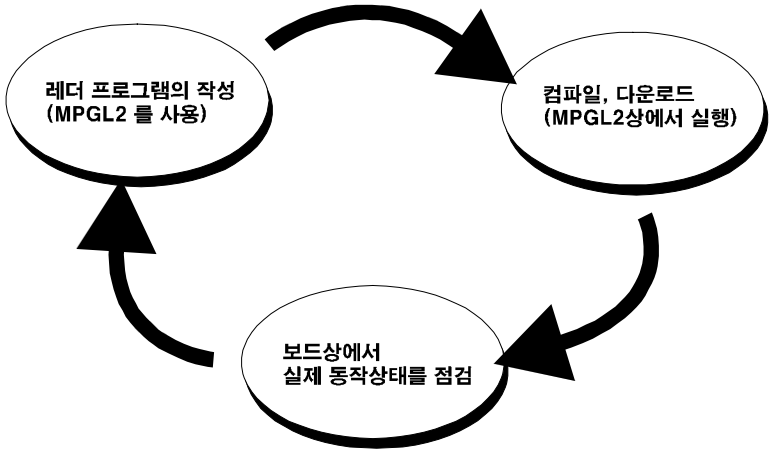
C	카운터	펄스의 갯수를 카운트하는 목적으로 사용합니다. 기본적으로 UP 타이머와 DOWN 타이머가 있으며, 입력펄스의 상승 에지 부분을 카운트하며, 설정 치에 다다르면 접점이 ON 됩니다. 자세한 설명은 명령어 설명을 참조하시기 바랍니다. (스캔 타임보다 빠른 펄스는 고속카운터로 측정해야 합니다.)
D	데이터	연산에 사용되는 데이터등을 저장하는 영역으로 기본적으로 워드(16비트)단위로 데이터를 취급합니다. TPC3X 에는 데이터 영역을 대상으로하는 여러가지 연산명령이 준비되어 있습니다.
S	스텝 컨트롤러 릴레이	순차적으로 처리하고자 하는 작업을 위해서 사용하는 특수한 릴레이입니다. PLC 는 그 특성상 스캔타임 간격으로 무조건 실행이 반복되기 때문에, 다른 언어에서는 쉽게 처리할 수 있는 순차처리가 힘든 경우가 간혹 발생합니다. 이 경우에 스탭 컨트롤러 릴레이를 사용하면 손쉽게 처리할 수 있습니다.
CH	LCD 디스플레이 영역	이 영역은 LCD 의 디스플레이용 버퍼입니다. 이 영역에 ASCII 코드 데이터를 써넣으면, LCD 상에 그대로 표시됩니다. 반대로 이 영역을 모두 블랭크 (ASCII 코드 20H)로 써넣으면 LCD 는 클리어 됩니다.
G	SGN 디스플레이 영역	이 영역은 세븐 세그먼트 디스플레이를 위한 버퍼영역 입니다. 이 영역에 ASCII 코드를 써넣으면 SGN (컴파일의 7 세그먼트 모듈 제품 이름)에 그대로 표시됩니다.
AD	AD 컨버터 결과 저장영역	A/D 변환 결과가 기록되는 영역입니다. TPC3X 에서는 A/D 변환을 위해 유저가 별도의 명령을 사용할 필요 없이, 항상 A/D 변환을 수행하고 있으며 그 결과는 이 영역에 보관됩니다. 유저는 단지 이 영역을 참조하는 것으로 A/D 결과를 얻을 수 있습니다.
CNT	고속카운터 결과 저장영역	고속 카운터의 카운터 결과가 기록되는 영역입니다. TPC3X 에서 고속카운터를 위해 유저가 별도의 명령을 지시할 필요는 없으며, 단지 이 영역을 참조하는 것으로 고속카운터 기능을 활용할 수 있습니다.

레더의 입력과 실행

PLC 를 사용하기 위해서는 레더 프로그램 입력기와 다운로드용 케이블 (RS232C 케이블) 그리고 PLC 본체를 필요로 합니다.

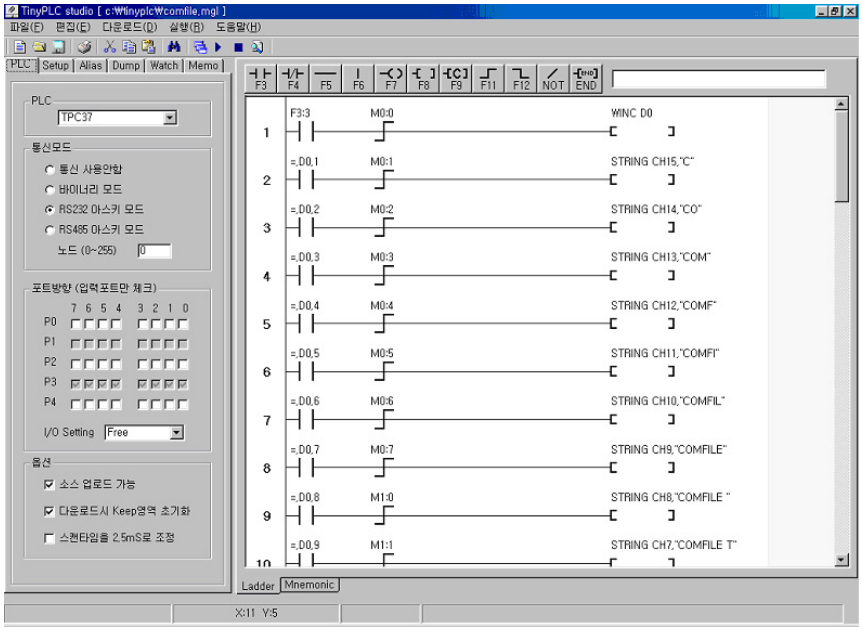


레더 입력을 위해 별도의 핸디로더(Handy Loader)등을 사용하는 기종들도 많이 있지만, 요즘에는 대부분 PC 를 이용하는 추세입니다. PC 상에서는 그래픽 등으로 보다 알기 쉽게 레더를 표현할 수 있으며, 데이터 백업 등이 용이하기 때문입니다. PC 상에서 작성된 레더 다이어그램은 RS232C 케이블을 통해서 PLC 로 다운로드 하거나 업로드 할 수 있습니다. 일단 다운로드된 프로그램은 PC 가 없어도 스탠드 얼론(Stand-Alone)상태에서 동작가능하며, 전원이 꺼진 후라도 입력된 프로그램은 항상 보존되어 있으므로, 언제든지 다시 전원을 켜면 입력해 둔 프로그램이 실행됩니다.



TPC3X 에서의 프로그램 개발은, **레더작성-->컴파일/다운로드-->동작점검**의 3 가지 상태를 반복하면서 진행하게 됩니다.

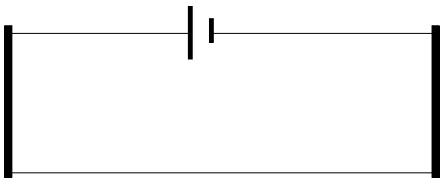
다음은 TPC3X 용 레더입력 프로그램인 TinyPLC Studio 의 실행화면 입니다. (TinyPLC Studio 는 윈도우즈 용입니다.



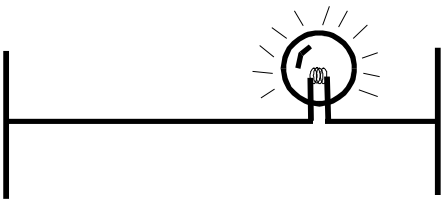
레더심볼의 기초

레더 심볼은 생각보다 매우 간단한 구조로 되어 있습니다. 레더에는 아래와 같은 두개의 모선이 기본적으로 필요합니다.

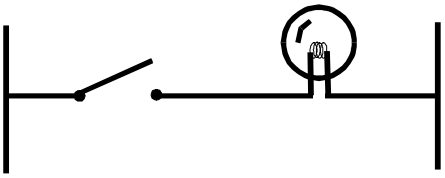
이 모선을 전원 선으로 생각하면 됩니다.



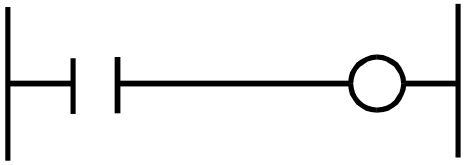
전원선 (모선)에 램프를 연결하면 당연히 불이 켜질 것입니다.



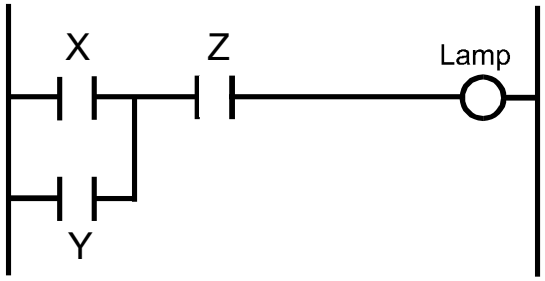
여기에 스위치를 연결하면 아래와 같은 그림이 됩니다. 스위치를 ON 하면 불이 켜지고, OFF 하면 불이 꺼집니다.



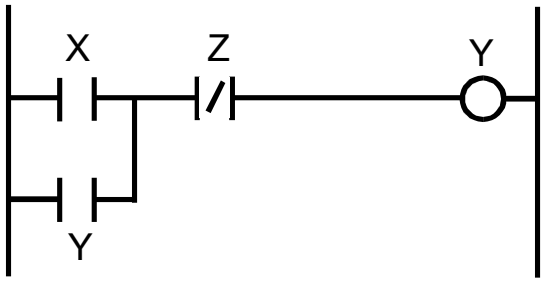
이 회로를 기호형태로 표시한 것이 바로 릴레이 레더 다이어그램입니다.



입력심볼 여러개를 아래와 같이 구성하면 AND, OR와 같은 논리 조건을 구현할 수 있습니다. X와 Z는 AND 조건, Y는 OR조건으로 한 회로에 접속되어 있습니다. X와 Z가 ON되면 램프가 ON됩니다. 또는 Y와 Z가 ON되어도 램프는 ON됩니다.



릴레이 레더 회로는 간단해 보이지만, 이를 응용하면 여러가지 다양한 효과를 낼 수도 있습니다. 다음은 "자기유지"를 하는 회로입니다. "자기유지"회로란 입력한 상태를 기억 (래치)하고 있다가 해제신호 입력시 풀어지는 회로를 말합니다.



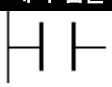
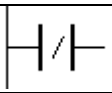
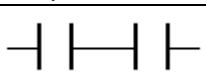
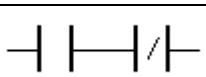
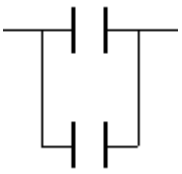
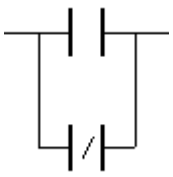
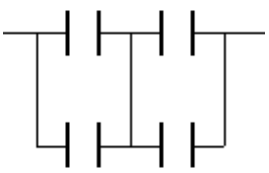
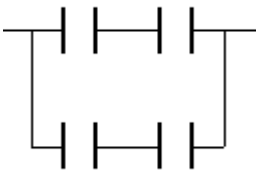
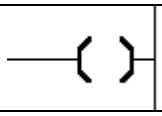
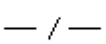
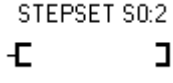
이 회로에서 X를 누르면 Y가 켜집니다. 출력 Y는 입력에 OR조건으로 연결되어 있으므로, 일단 한번 켜진 Y 출력은 계속 ON 상태를 유지하게 됩니다. Z 를 누르면 ON 상태를 해제합니다. (Z는 평상시 ON되어 있다가 입력이 들어오면 OFF 되는 B 접점 스위치입니다.)

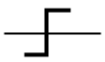
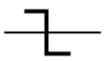
제 3 장
TPC3X
기본명령어

TPC3X 의 기본명령어에 대한 자세한 설명을 다루는 장입니다.

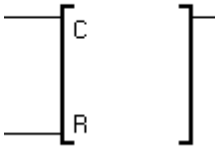
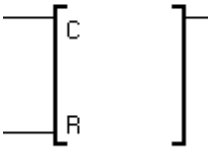
명령어 요약표

기본 명령어

이름	명령어	레더 심볼	설명
Load	LOAD		A 접점 (Normal Open) 으로 부터 연산의 시작
Load Not	LOADN		B 접점 (Normal Close) 으로 부터 연산의 시작
And	AND		A 접점 직렬 접속
And Not	ANDN		B 접점 직렬 접속
Or	OR		A 접점 병렬 접속
Or Not	ORN		B 접점 병렬 접속
And Stack	ANDS		블록간의 AND 접속
Or Stack	ORS		블록간의 OR 접속
Output	OUT		연산 결과 출력
Not	NOT		명령전까지의 연산결과를 반전
Step Sequential Set	STEPSET		스텝 콘트롤러 출력 (순차제어)

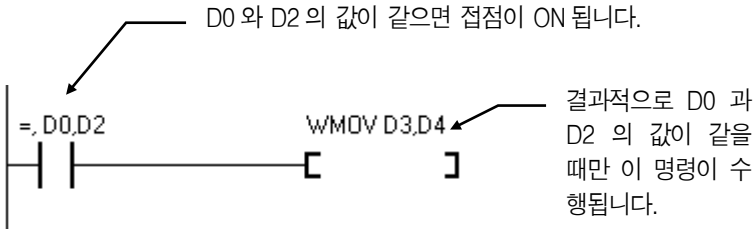
Step Output	STEPOUT	STEPOUT S0:1	스텝 콘트롤러 출력 (후입우선)
Master Control Set	MCS	MCS 0	마스터 콘트롤 릴레이 시작
Master Control Clear	MCSCLR	MCSCLR 0	마스터 콘트롤 릴레이 종료
Differential	DF		입력조건 상승시 1 스캔타임만 On 출력 (미분 입력)
Differential Not	DFN		입력조건 하강시 1 스캔타임만 Off 출력 (미분 반전 입력)
Set Output	SETOUT	SETOUT P0:0	접점출력을 On 으로 SET
Reset Output	RSTOUT	RSTOUT P0:1	접점출력을 Off 로 Reset
Save Status	SAVES	레더 없음	현재의 연산상태를 SAVE
Read Status	RDS	레더 없음	SAVE 해둔 상태를 읽어옴 SAVES 와 RDS 는 레더의 분기에 사용함
End	END	END	프로그램의 끝

타이머/카운터

이름	명령어	레더 심볼	설명
On Timer (10mS)	TON	TON T1,100	0.01 초 ON 딜레이 타이머 (최대 327.67 초) 입력이 들어오면 타이머 가동 개시되고, 안 들어오면 타이머는 리셋됩니다. 타이머 값이 설정 치에 도달하면 접점은 On 됩니다.
Off Timer (10mS)	TOFF	TOFF T1,100	0.01 초 OFF 딜레이 타이머 (최대 327.67 초) 입력이 들어오면 접점은 On 됩니다. 이후 입력이 끊기면 접점은 바로 Off 되지 않고 설정치 만큼의 시간이 경과된 뒤 Off 됩니다.
On Timer (100mS)	TAON	TAON T1,100	0.1 초 ON 딜레이 타이머 (최대 3276.7 초) 동작은 TON 명령과 동일
Off Timer (100mS)	TAOFF	TAOFF T1,100	0.1 초 OFF 딜레이 타이머 (최대 3276.7 초) 동작은 TOFF 명령과 동일
Up Counter	CTU	CTU C1,100 	업 카운터 (최대 65535 까지 카운트 가능) 입력이 들어오면 1 씩 증가되다가, 설정 치에 도달하면 접점이 On 됩니다. 리셋 입력이 들어오면 카운터는 0 이 됩니다.
Down Counter	CTD	CTD C1,100 	다운 카운터 (최대 65535 부터 카운트 가능) 입력이 들어오면 1 씩 감소되다가 0 이 되면 접점이 On 됩니다. 리셋 입력이 들어오면 카운터는 설정 치로 세트 됩니다.

비교명령어

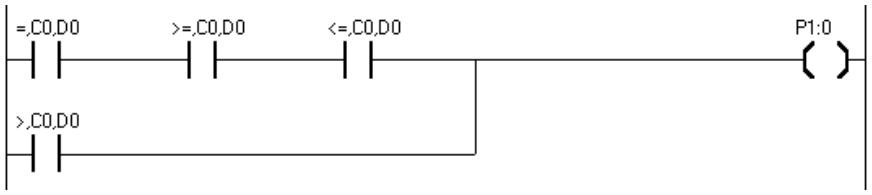
TPC3X 에서의 비교명령은 아래에 표시된 레더처럼, 입력심볼에 부등호로 시작하는 명령을 사용해서 구현할 수 있습니다.



S1, S2 는 인수 1, 인수 2 를 의미합니다.

분 류	명령어 형식	설 명
16 비트 비교명령	=, S1, S2	S1 과 S2 가 같을 때 접점이 ON 됩니다. 1 워드 (16 비트) 값을 비교합니다.
	>, S1, S2	S1 > S2 일때 접점이 ON 됩니다.
	<, S1, S2	S1 < S2 일때 접점이 ON 됩니다.
	<=, S1, S2	S1 <= S2 일때 접점이 ON 됩니다.
	>=, S1, S2	S1 >= S2 일때 접점이 ON 됩니다.
	<>, S1, S2	S1 과 S2 가 다른 값일 때 접점이 ON 됩니다.
32 비트 비교명령	D=, S1, S2	S1 과 S2 가 같을때 접점이 ON 됩니다. 더블워드 (32 비트)값을 비교합니다.
	D>, S1, S2	S1 > S2 일때 접점이 ON 됩니다.
	D<, S1, S2	S1 < S2 일때 접점이 ON 됩니다.
	D<=, S1, S2	S1 <= S2 일때 접점이 ON 됩니다.
	D>=, S1, S2	S1 >= S2 일때 접점이 ON 됩니다.
	D<>, S1, S2	S1 과 S2 가 다른 값일 때 접점이 ON 됩니다.

비교명령도 일반 접점입력과 마찬가지로 AND, OR 접속을 자유롭게 할 수 있습니다.



LOAD, LOADN A/B 접점 시작

개요

LOAD 는 A 접점의 시작, LOADN 은 B 접점의 시작 명령입니다.

프로그램 예

사용가능 영역

명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
LOAD	O	O	O	O	O	O	O				
LOADN	O	O	O	O	O	O	O				

자세한 설명

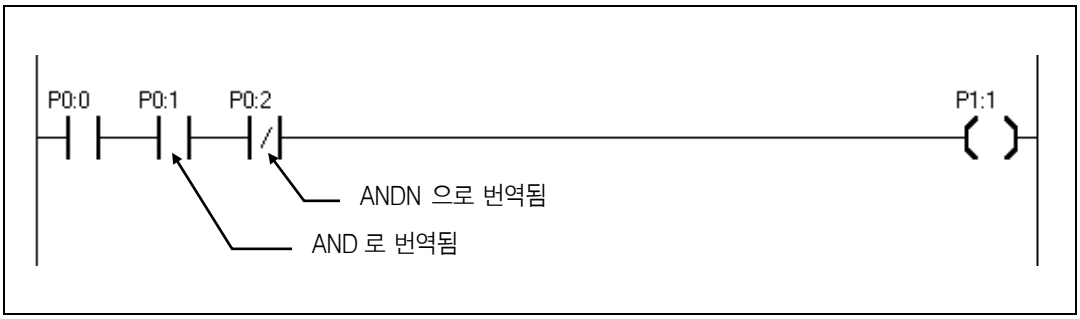
A 접점 P0:0 이 ON 되면 출력 P1:0 이 ON 됩니다.
B 접점 P0:1 이 ON 되면 출력 P1:1 이 OFF 됩니다.
(입력으로 사용된 P0:0 과 P0:1 은 TinyPLC Studio 프로그램에서 입력으로 설정해주어야 합니다. 마찬가지로 출력으로 사용된 P1:0 과 P1:1 은 출력으로 설정해 주어야 합니다. 입출력 설정이 잘못되었을 경우에는 올바르게 동작하지 않습니다.)

AND, ANDN A/B 접점의 직렬접속

개요

AND 는 A 접점의 직렬접속, ANDN 은 B 접점의 직렬접속 명령입니다.

프로그램 예



사용가능 영역

명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
AND	O	O	O	O	O	O	O				
ANDN	O	O	O	O	O	O	O				

자세한 설명

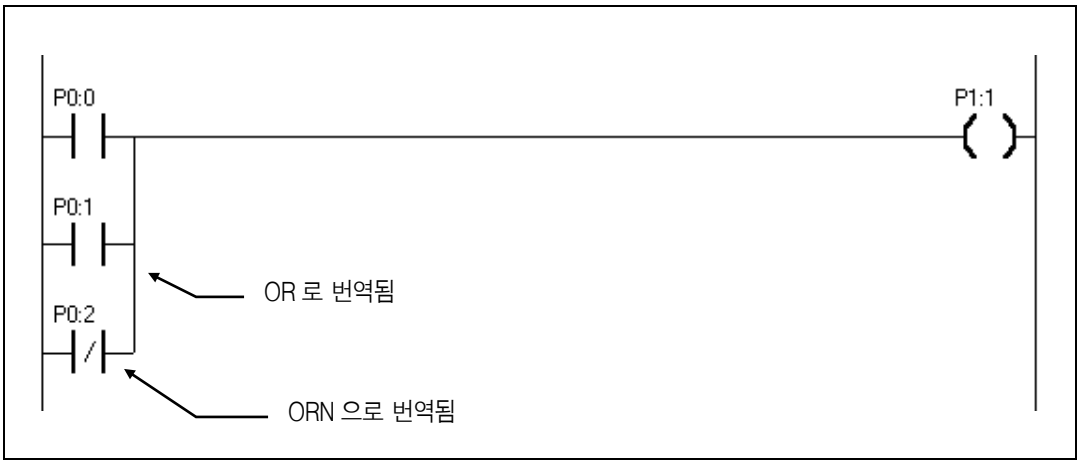
P0:0 과 P0:1 이 ON 되면 출력 P1:1 이 ON 됩니다.
B 접점 P0:2 이 ON 되면 출력 P1:1 은 OFF 됩니다.

OR,ORN A/B 접점의 병렬접속

개요

OR 는 A 접점의 병렬접속, ORN 은 B 접점의 병렬접속 명령입니다.

프로그램 예



사용가능 영역

명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
OR	O	O	O	O	O	O	O				
ORN	O	O	O	O	O	O	O				

자세한 설명

P0:0 또는 P0:1 이 ON 되면 또는 P0:2 가 OFF 되면 출력 P1:1 이 ON 됩니다.

OUT 연산결과의 출력

개요

연산결과를 접점에 출력 합니다.

프로그램 예

사용가능 영역

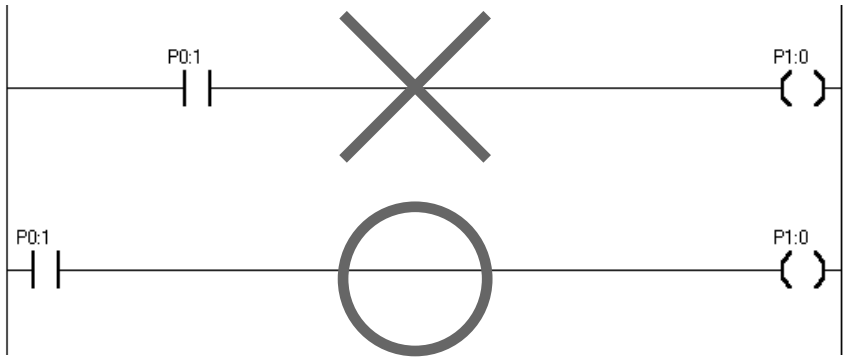
명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
OUT	O	O		O							

자세한 설명

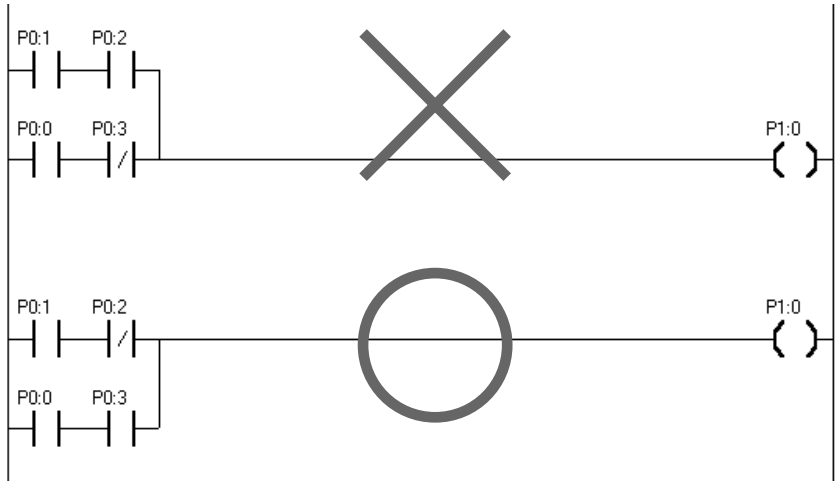
P0:0 이 ON 되면 출력 P1:1 이 ON 됩니다.

레더 입력시 주의사항

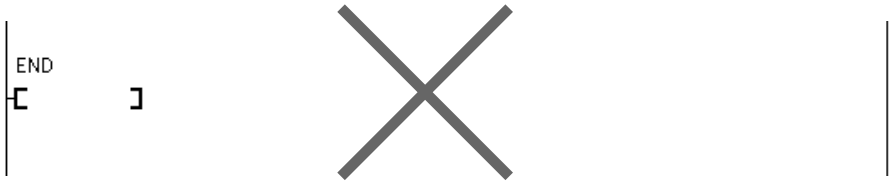
LOAD 명령은 반드시 첫번째 컬럼 (모선의 바로 옆)에서 시작해야 합니다.



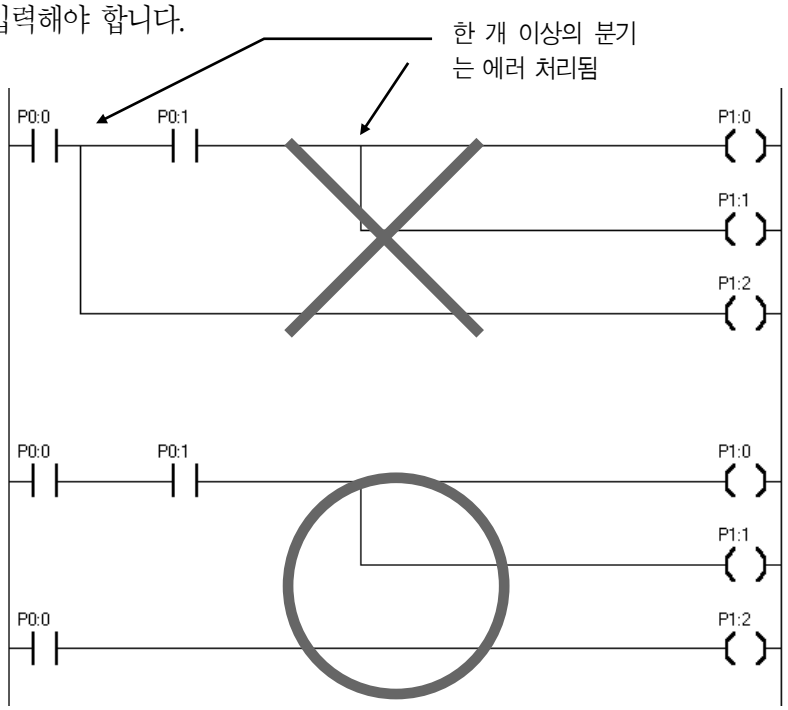
레더는 아래 방향으로 전개해야 합니다.



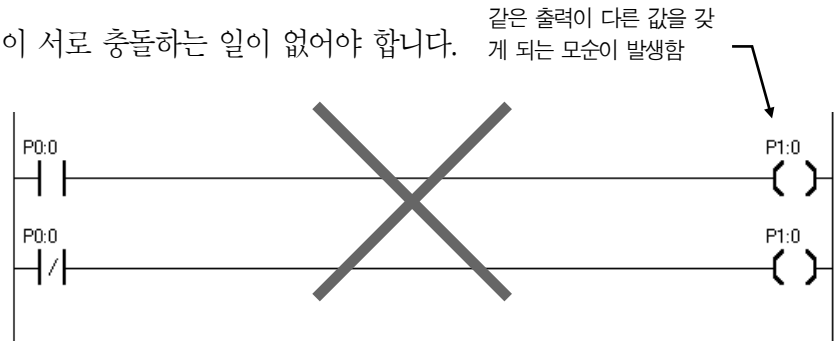
출력 심볼은 첫번째 컬럼 (왼쪽 모선의 바로 옆)에 위치할 수 없습니다.



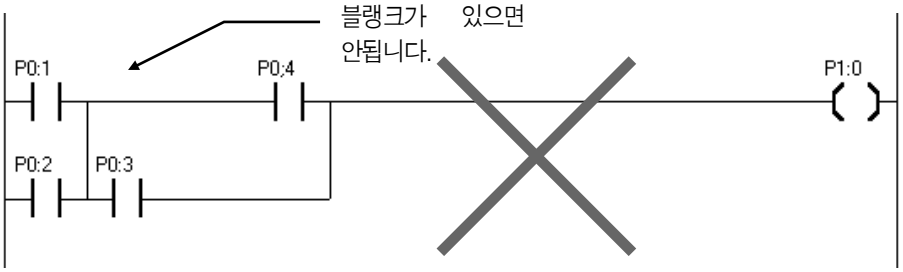
한개의 라인에서 여러개의 출력접점을 분리 연결할 수 없습니다. 이 경우에는 아래그림처럼 고쳐서 입력해야 합니다.



같은 출력이 서로 충돌하는 일이 없어야 합니다.



필요없는 블랭크(횡접속선)를 삽입하면 번역오류가 발생합니다. (블랭크 없이 붙여서 작성해야 합니다.)

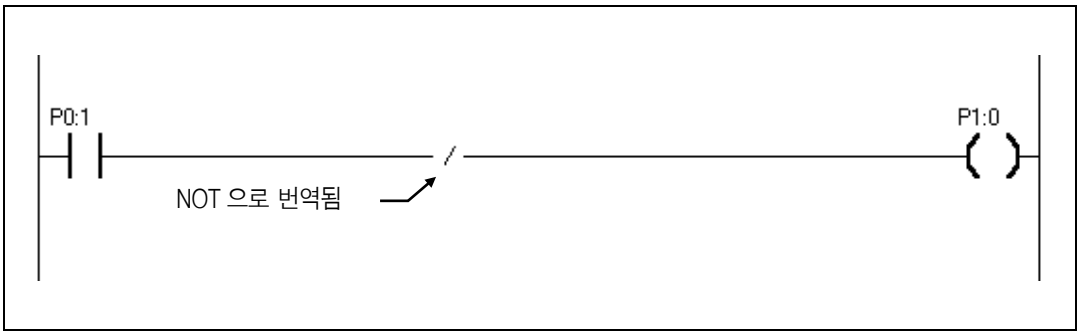


NOT 연산결과의 출력

개요

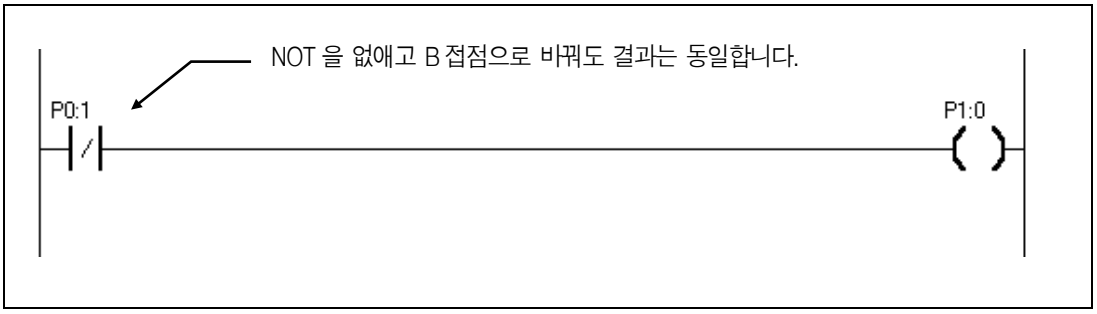
NOT 이전까지의 연산결과를 반전합니다.

프로그램 예



자세한 설명

P0:1 이 ON 되면 출력 P1:1 이 OFF 됩니다. 위 프로그램은 다음 레더와 같은 효과를 갖게 됩니다.



END 프로그램의 끝

개요

전체 프로그램의 끝을 나타냅니다. (항상 프로그램의 마지막에 위치해야 합니다.)

프로그램 예

자세한 설명

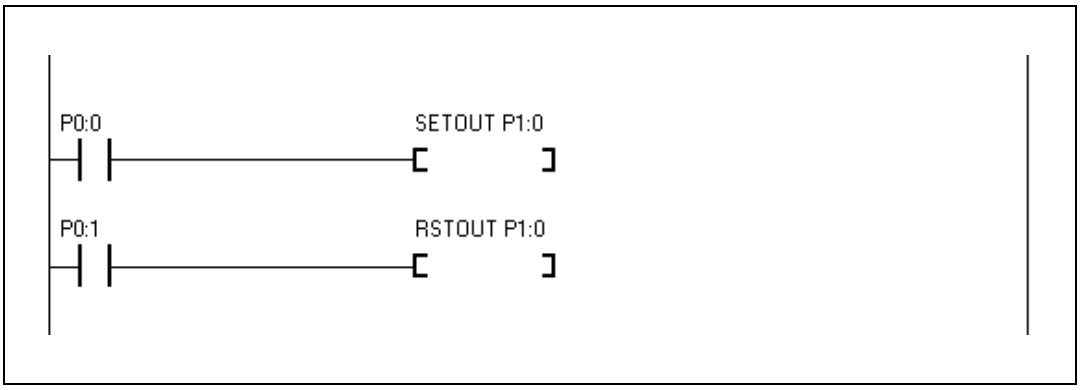
TinyPLC Studio 는 END 명령이 있는 곳까지만 번역하고 저장합니다. END 이후에 작성된 명령은 모두 무시되므로 주의해야 합니다.

SETOUT 출력을 ON 상태로 유지

개요

출력 접점을 ON 상태로 유지합니다.

프로그램 예



사용가능 영역

명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
SETOUT	O	O		O							

자세한 설명

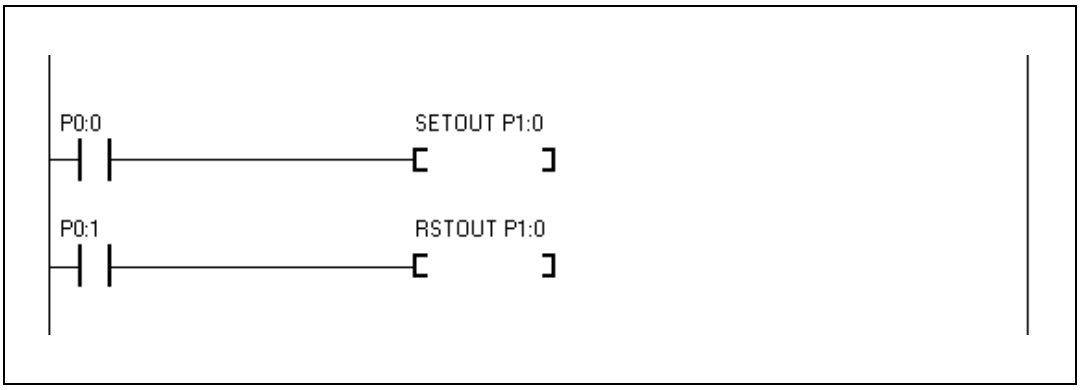
P0:0 이 ON 되면 P1:0 출력 접점이 ON 상태를 유지합니다.

RSTOUT 출력을 OFF 상태로 유지

개요

출력 접점을 OFF 상태로 유지합니다.

프로그램 예



사용가능 영역

명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
RSTOUT	O	O		O							

자세한 설명

P0:1 이 ON 되면 P1:0 출력 접점을 OFF 상태로 유지합니다.

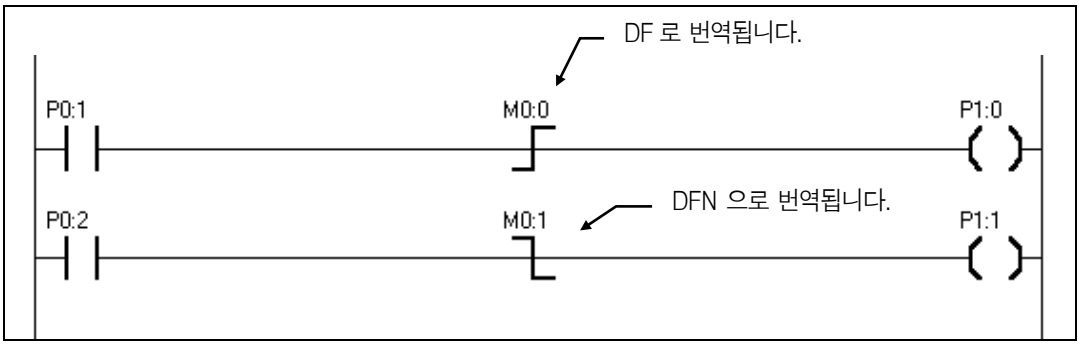
DF, DFN 미분입력

개요

DF : 입력 조건의 상승 에지 (Off-->On)가 검출되면 출력 접점을 1 스캔타임동안 ON 합니다.

DFN : 입력 조건의 하강 에지 (On-->Off)가 검출되면 출력 접점을 1 스캔타임동안 ON 합니다.

프로그램 예



사용가능 영역

명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
DF		O									
DFN		O									

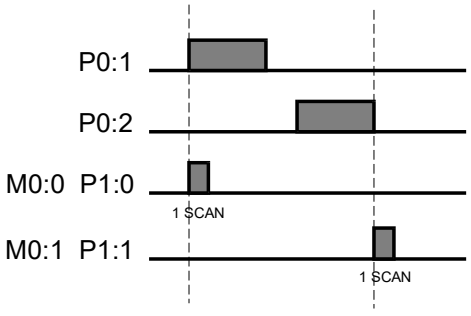
*** 주의 : DF 와 DFN 명령은 M 영역만 사용할 수 있습니다.**

자세한 설명

P0:1 이 ON 되는 순간 P1:0 이 1 스캔타임 동안 ON 됩니다.

P0:2 이 OFF 되는 순간 P1:1 이 1 스캔타임 동안 ON 됩니다.

Time Chart

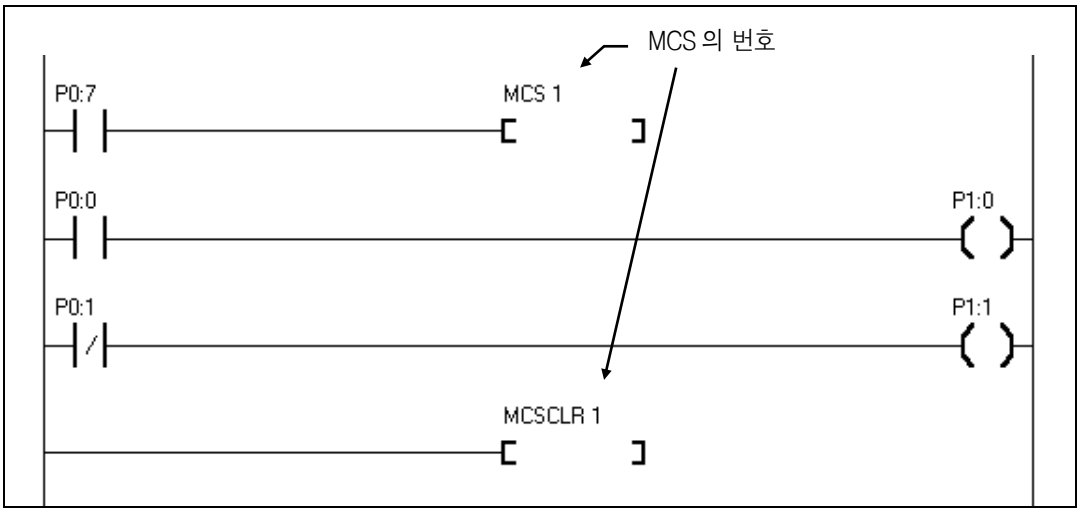


MCS, MCSCLR 마스터 콘트롤 릴레이

개요

MCS 의 입력조건이 ON 하면 같은 번호를 가진 MCSCLR 까지 실행하고, 입력 조건이 OFF 되면 실행하지 않습니다. 실행하지 않을 때, MCS ~ MCSCLR 범위 안에 있는 모든 출력은 OFF 됩니다.

프로그램 예

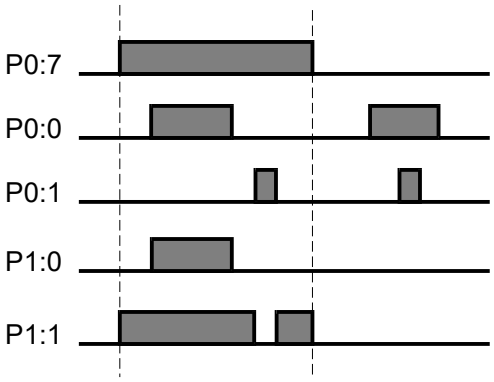


자세한 설명

P0:7 이 ON 되면 MCS 1 ~ MCSCLR 1 사이에 있는 레더가 실행됩니다. P0:7 이 OFF 되면 MCS 1 ~ MCSCLR 1 사이에 있는 레더는 실행되지 않고, 출력 P1:0 과 P1:1 은 OFF 됩니다.

MCS 번호는 0 에서 7 까지 사용가능 합니다. 우선순위는 0 이 가장 높고, 7 이 가장 낮으므로, 우선순위가 높은 MCS 를 해제하면, 그 밑에 있는 MCS 까지 같이 해제 됩니다.

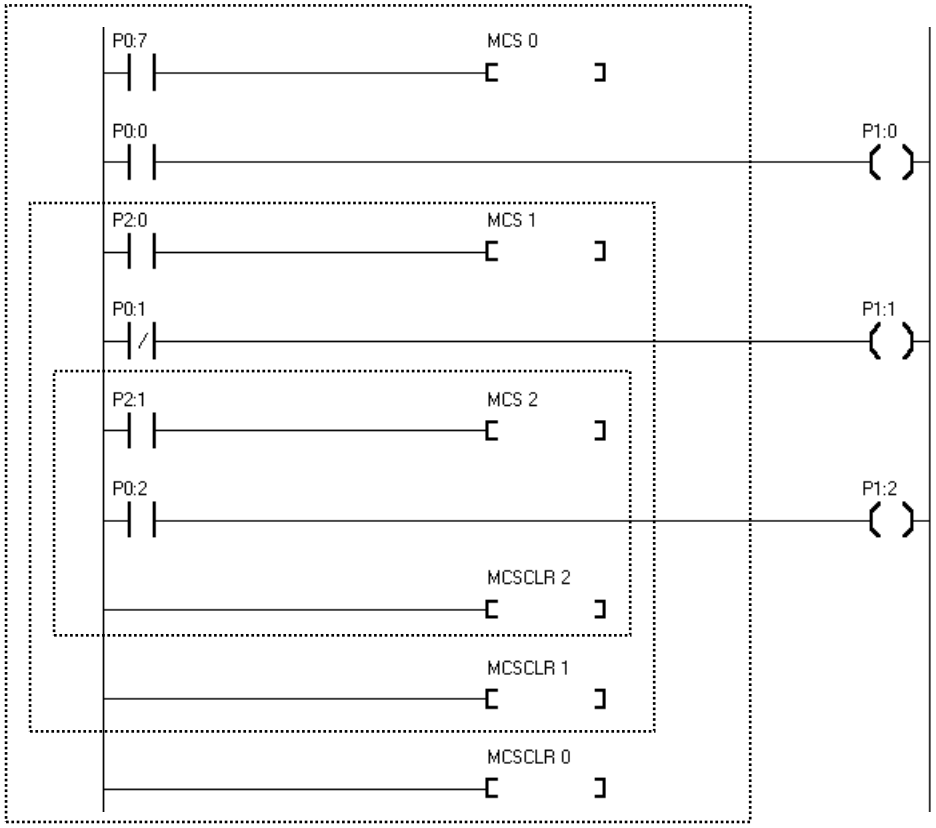
Time Chart



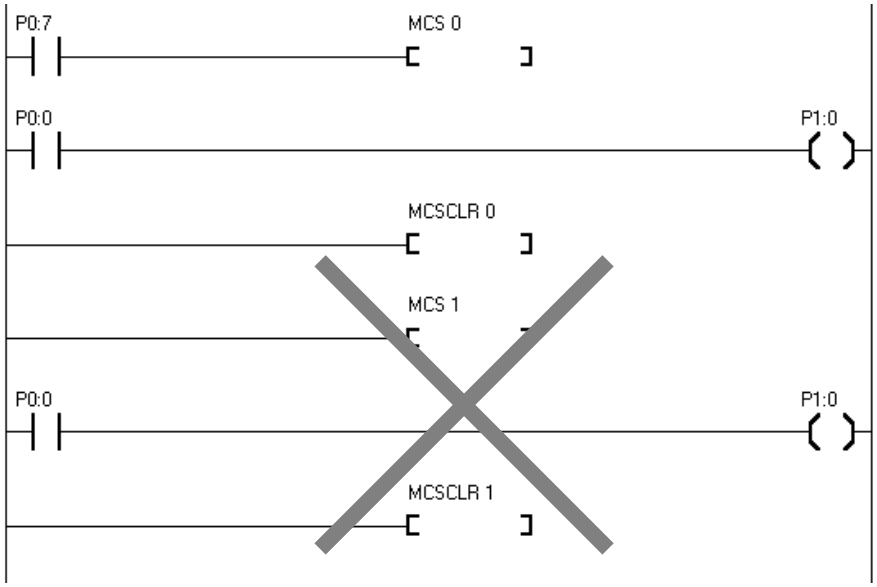
다음은 MCS 루프안에서 영향을 받는 명령어들을 정리한 표입니다.

명령어	MCS 가 ON 조건일때	MCS 가 OFF 조건일때
OUT	정상 동작	무조건 OFF
SETOUT	정상 동작	MCS 가 OFF 되기 전 상태를 계속 유지
RSTOUT	정상 동작	MCS 가 OFF 되기 전 상태를 계속 유지
타이머 명령 (TON, TOFF..)	정상 동작	초기값으로 리셋
카운터 명령 (CTU, CTD)	정상 동작	MCS 가 OFF 되기 전 상태를 계속 유지
기타 명령	정상 동작	실행 안함

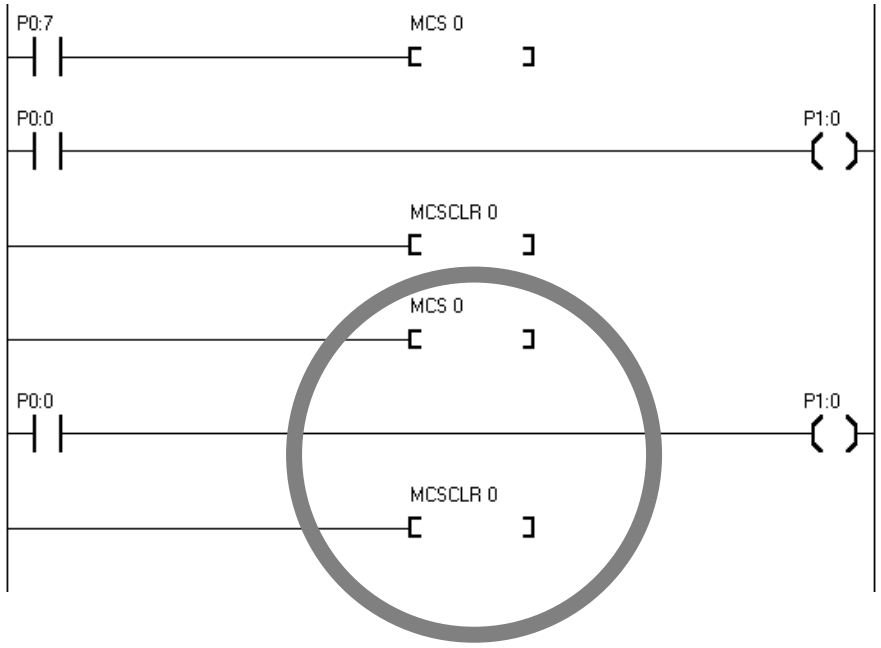
다음은 MCS, MCSCLR 명령의 중첩사용 예입니다. (최대 8 레벨까지 중첩 가능하며, **반드시 낮은번호를 먼저 사용하고, 큰 번호가 그 안에 들어가도록 배치**해야 합니다.)



주의사항 : 만약 중첩하지 않고, 여러 개의 MCS 블록이 계속된다면, MCS 0 만 사용해야 합니다. (MCS 0 과 MCLSCLR 0 을 반복적으로 사용)



위의 경우는 다음과 같이 수정해야 합니다.

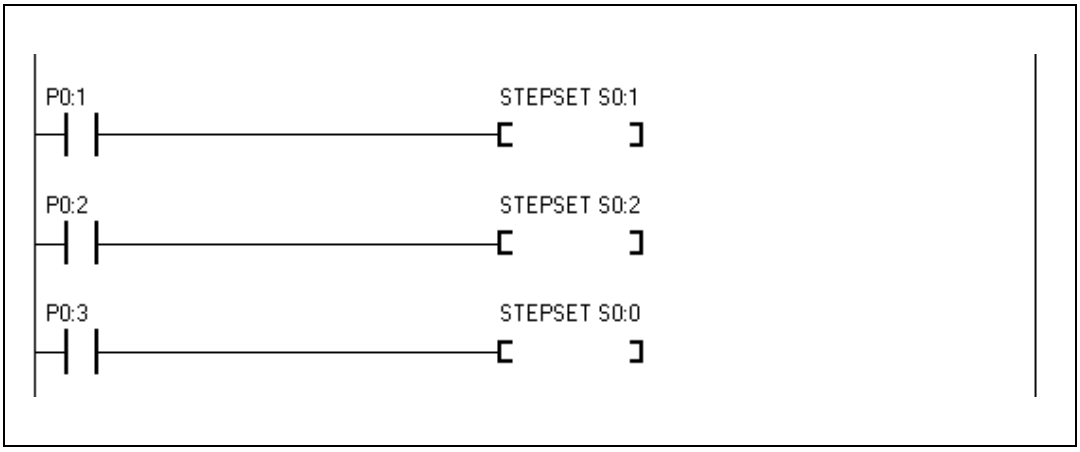


STEPSET 스탭콘트롤(순차제어)

개요

동일 조 내의 바로 전 번호가 ON 되었을 때, 현재 번호가 ON 되고, 이전번호는 OFF 됩니다. (순차적으로만 ON 되기 때문에 순차제어라고 부름) 0 에서 255 스탭까지 사용가능

프로그램 예



사용가능 영역

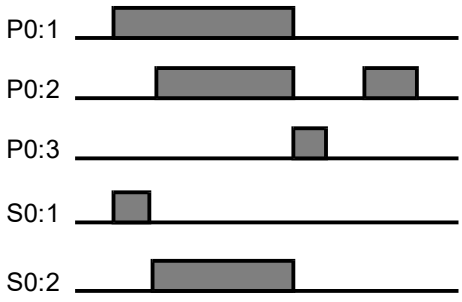
명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
STEPSET					O						

자세한 설명

P0:2 이 ON 되면 0 조의 2 번 스탭을 ON 하려고 시도합니다. 이때 같은조의 1 번 스탭이 ON 되어 있었다면 1 번 스탭은 OFF 되고 2 번스탭이 ON 됩니다.

P0:3 이 ON 되면 무조건 0 번스탭으로 되돌립니다. (0 번 스탭은 리셋 용도로 사용합니다.)

Time Chart

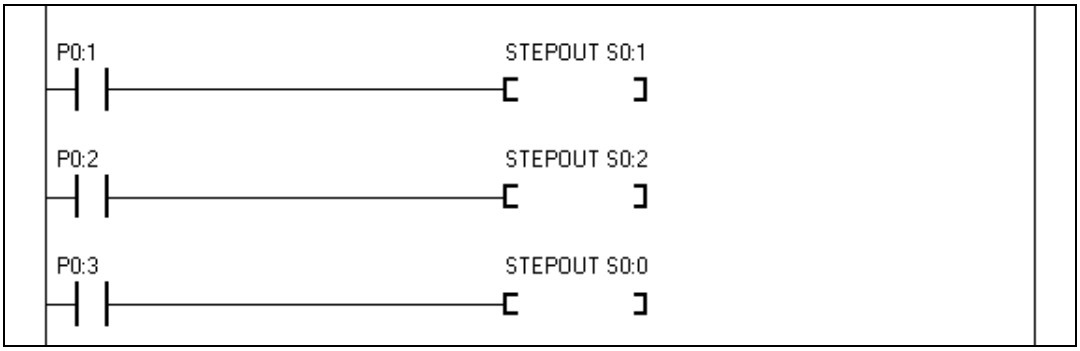


STEPOUT 스탭콘트롤(후입우선)

개요

동일 조 내에서 여러 개의 입력이 들어와도 맨 나중에 들어온 스탭만 ON 되고 나머지 스탭은 OFF 됩니다. (나중에 들어온 것이 우선하기 때문에 후입 우선이라고 부릅니다.) 0~255 스탭까지 사용 가능

프로그램 예



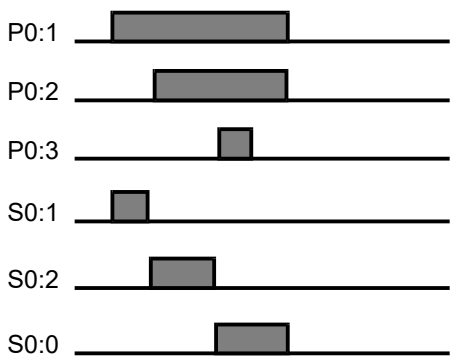
사용가능 영역

명령	릴 레 이					카운터	타이머	기타			상수
	P	M	F	K	S	C	T	AD	CH	G	
STEPOUT					O						

자세한 설명

P0:1 이 ON 되면 0 조의 1 번 스탭이 ON 됩니다. 이후에 P0:3 이 ON 되면 0 번 스탭이 ON 됩니다. 이후에 P0:2 이 ON 되면 2 번스탭이 ON 됩니다. 무조건 나중에 들어온 것만 ON 되고 나머지는 OFF 됩니다.

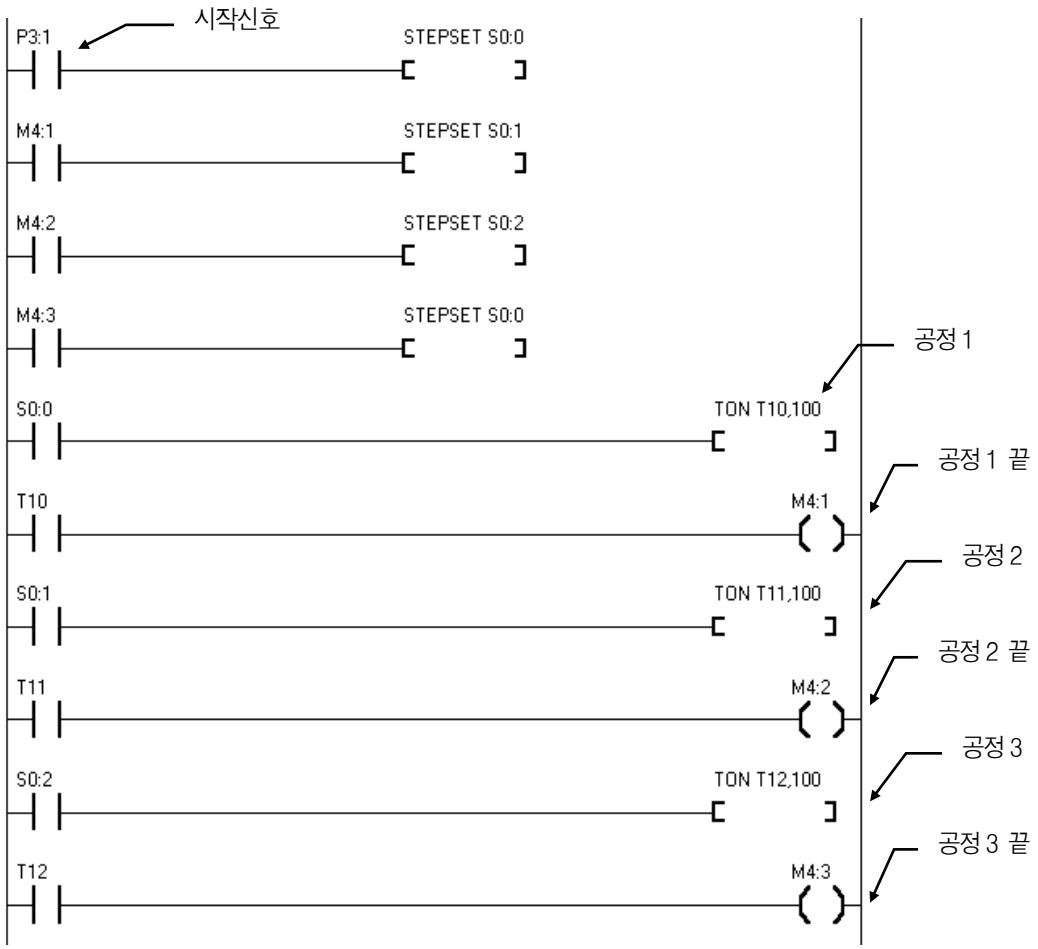
Time Chart



스탭 콘트롤러에 대한 추가설명

C 나 어셈블리를 사용하는 프로세서에서는 쉽게 처리할 수 있는 순차제어를, PLC 에서는 스캔타임이라는 특성때문에 구현할 수 없는 경우가 있습니다. 이 때문에 생겨난 개념이 "스탭 콘트롤러"라는 방식인데, 어떤 일을 순서에 입각해서 처리하고자 할때 필요한 기능입니다.

아래 그림에서 처럼 공정 1 이 끝나면 다음 스탭을 ON 하고, 공정 2 가 끝나면 그 다음 스탭을 ON 하는 방식으로 차례,차례 일을 수행할 수 있습니다. (아래 프로그램에서는 공정대신 타이머를 사용하고 있습니다.)



- 스탭 릴레이는 자기 보존 기능을 가지고 있습니다. (다른 입력이 있기 전까지는 현 상태 유지)
- 한 조에서는 반드시 하나의 출력만 ON 되어 있습니다. (인터 록 개념과 동일)
- 순차제어의 경우 ; 한 칸씩 뒤로만 이동 가능합니다. (이전 번호가 ON 되어 있을때만 ON 됨)
- 후입우선의 경우 ; 여러 개 의 입력이 들어와도 다 무시하고, 가장 나중에 들어온 것 하나만 ON 시킵니다.

TON, TAON

ON 딜레이 타이머

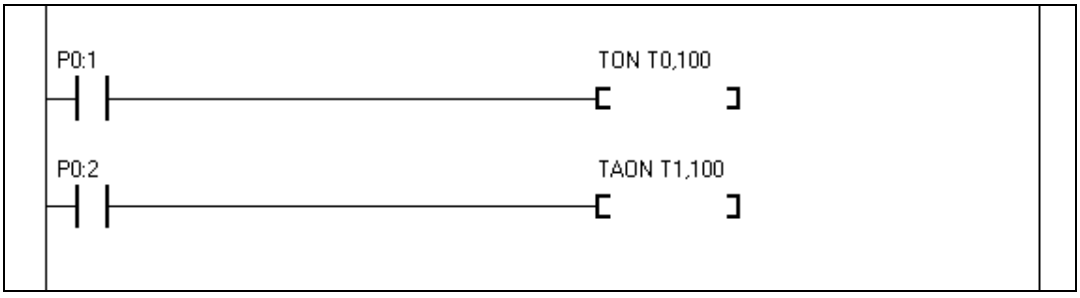
Usage : TON t, n

개요

입력조건이 ON 되면 타이머가 가동 개시되고, OFF 되면 타이머는 리셋됩니다. 타이머 값이 설정치에 도달하면 출력접점이 ON 됩니다. 시간단위가 틀린 두 종류의 타이머가 있습니다.

타이머종류	분해능	최대치
TON	0.01 초	327.68 초
TAON	0.1 초	3276.8 초

프로그램 예



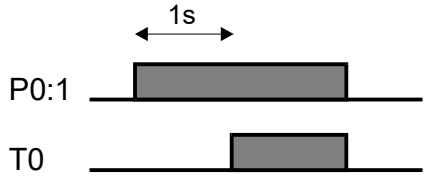
사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타			상수
	P	M	F	K	S	C	T	D	AD	CH	G	
t (타이머 접점)							O					
n (설정치)	O	O		O		O	O	O	O			O

자세한 설명

P0:1 이 ON 된 후 1 초가 경과하면 T0 접점이 ON 됩니다. P0:2 가 ON 된 후 10 초가 경과하면 T1 접점이 ON 됩니다.

Time Chart



TOFF, TAOFF

OFF 딜레이 타이머

Usage : TOFF t, n

개요

입력조건이 ON 되면 출력 접점은 바로 ON 됩니다. 이후 입력이 OFF 되면 접점은 바로 OFF 되지 않고 설정치 만큼의 시간이 경과된 뒤 OFF 됩니다. 시간단위가 틀린 두 종류의 타이머가 있습니다.

타이머종류	분해능	최대치
TOFF	0.01 초	327.68 초
TAOFF	0.1 초	3276.8 초

프로그램 예

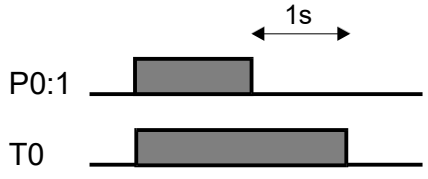
사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타			상수
	P	M	F	K	S	C	T	D	AD	CH	G	
t (타이머 접점)							O					
n (설정치)	O	O		O		O	O	O	O			O

자세한 설명

P0:1 이 ON 되면 T0 이 바로 ON 됩니다. P0:1 이 OFF 되면 1 초가 경과하면 T0 접점도 OFF 됩니다.

Time Chart



CTU

UP 카운터

Usage : CTU c, n

개요

카운터 입력이 들어올 때마다 카운터 값을 1 증가 시킵니다. 카운터 값이 설정치와 일치하면 출력접점을 ON 합니다. 접점 ON 이후에도 입력이 들어오면, 카운터는 계속 증가됩니다. (최대 65535 까지 증가됩니다. 65535 가 넘으면 카운터는 다시 0 부터 시작하며, 접점상태는 계속 유지됩니다.) 리셋 입력이 들어오면 카운터 값은 0 이 됩니다.

프로그램 예

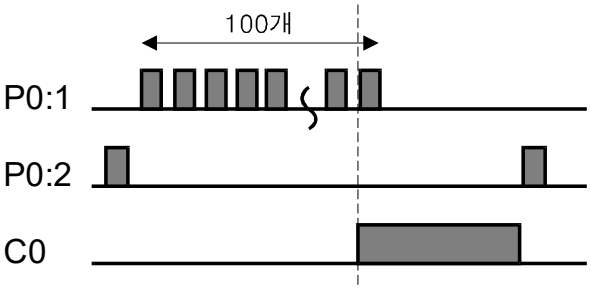
사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타			상수
	P	M	F	K	S	C	T	D	AD	CH	G	
c (카운터 접점)						O						
n (설정치)	O	O		O		O	O	O	O			O

자세한 설명

P0:0 이 100 번 ON 되면 C0 접점이 ON 됩니다. P0:1 접점이 ON 되면 카운터는 리셋되고 접점도 OFF 됩니다.

Time Chart



CTD

DOWN 카운터

Usage : CTD c, n

개요

카운터 입력이 들어올 때마다 카운터 값을 1 감소합니다. 카운터 값이 0 이 되면 출력접점을 ON 합니다. 리셋 입력이 들어오면 카운터 값은 설정치가 됩니다. (파워온시 설정치로 세트 됩니다.)

프로그램 예

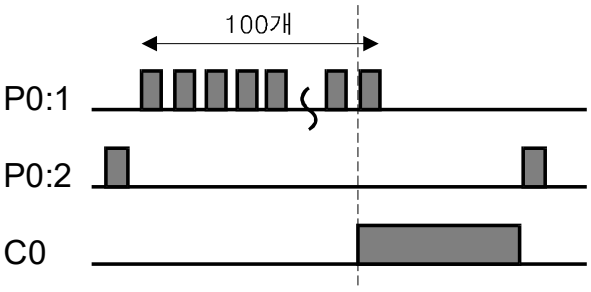
사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타			상수
	P	M	F	K	S	C	T	D	AD	CH	G	
c (카운터 접점)						O						
n (설정치)	O	O		O		O	O	O	O			O

자세한 설명

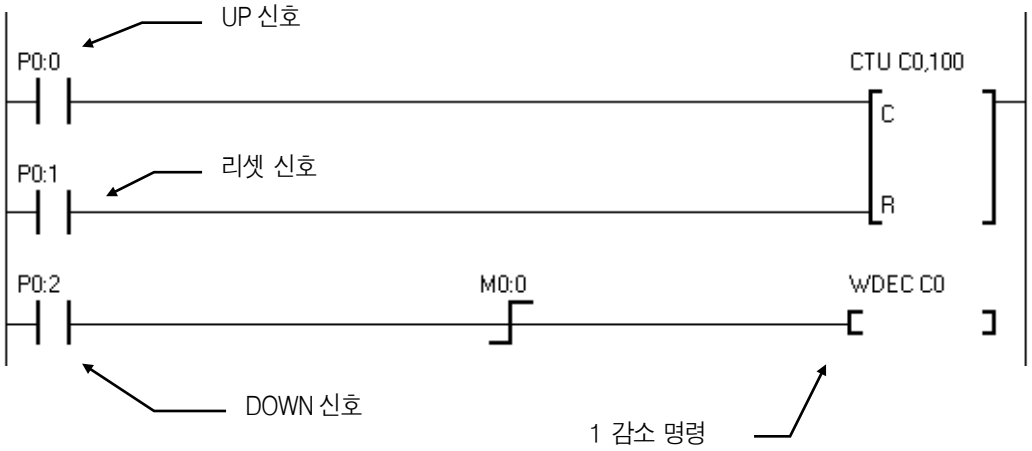
P0:0 이 100 번 ON 되면 C0 접점이 ON 됩니다. P0:1 접점이 ON 되면 카운터에는 100 이 세트고 접점도 OFF 됩니다.

Time Chart



업/다운 카운터의 구현

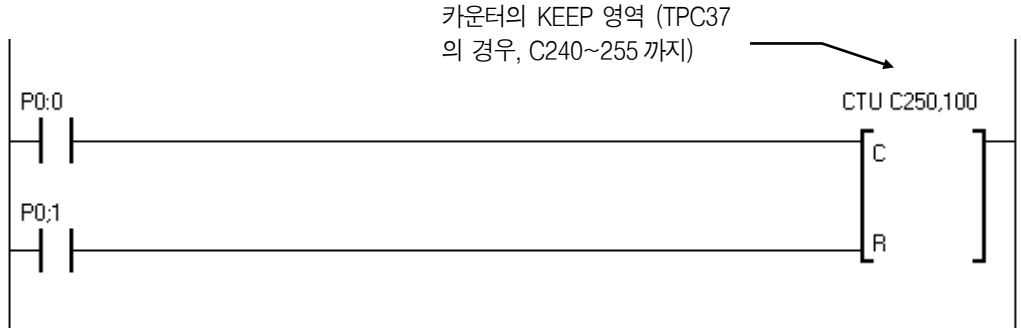
TPC3X 에는 업/다운 카운터 명령이 별도로 준비되어 있지는 않지만, 아래와 같이 입력하면 UP / DOWN 카운터를 구현할 수 있습니다.



P0:2 신호에 DF 명령을 써서 미분한 뒤, 1 감소 명령인 WDEC 를 실행하면 카운터 C0 값이 1 씩 감소 됩니다. 따라서 P0:0 은 증가입력, P0:2 는 감소입력으로 사용할 수 있습니다.

타이머와 카운터의 KEEP 영역

메모리 맵을 살펴보면, TIMER 와 COUNT 영역 중 일부분 (빗금 친 부분)은 정전 시 상태가 보존되는 KEEP 영역입니다. 이 영역을 사용하면, 진행중인 타이머 (또는 카운터)값이 정전이 되도 사라지지 않고, 계속 보존하고 있다가, 전기가 다시 들어오면, 그 이후 값부터 계속 진행됩니다.



주의: 카운터의 KEEP 영역사용시 CTD 명령은 POWER ON 시 해당 값으로 초기화되므로, CTU 명령을 사용해야만 POWER OFF 전의 값을 계속해서 카운트 할 수 있습니다.

= , <>, >, <, >=, <= 워드비교명령

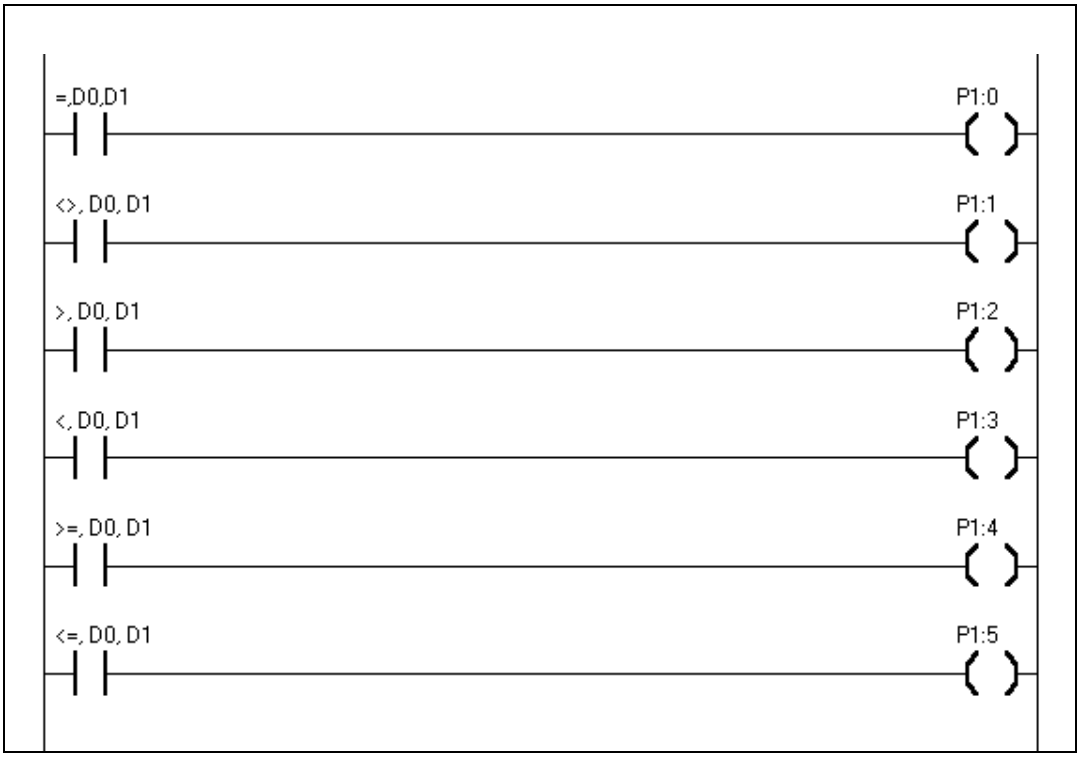
Usage : =, s1, s2

개요

두개의 워드 값을 비교해서 조건을 만족하면 접점을 ON 합니다. 총 6 개의 비교 명령이 있습니다.

비교 명령	동작
= , s1, s2	s1 과 s2 가 같을 때 접점이 on 됩니다.
<> , s1, s2	s1 과 s2 가 서로 다를 때 접점이 on 됩니다.
>, s1, s2	s1 > s2 일 때 접점이 on 됩니다.
<, s1, s2	s1 < s2 일 때 접점이 on 됩니다.
>=, s1, s2	s1 >= s2 일 때 접점이 on 됩니다.
<=, s1, s2	s1 <= s2 일 때 접점이 on 됩니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O

자세한 설명

D0 와 D1 이 같으면 P1:0 이 ON 됩니다.
D0 와 D1 이 다르면 P1:1 이 ON 됩니다.
D0 > D1 이면 P1:2 가 ON 됩니다. D0 < D1 이면 P1:3 가 ON 됩니다. D0 >= D1 이면 P1:4 가 ON 됩니다. D0 <= D1 이면 P1:5 가 ON 됩니다.

다음 그림과 같이 여러 개의 조건을 AND, OR 접속해서 사용하는 것도 가능합니다.

C2 가 30 과 같고, C0 가 100 과 같거나 크면 P1:0 이 ON 됩니다. 또는 D1 < 3 이고 C0 >= 100 일 때에도 P1:0 은 ON 됩니다.

D= , D<>, D>, D<, D>=, D<=

더블 워드비교명령

Usage : D=, s1, s2

개요

두개의 더블워드 값을 비교해서 조건을 만족하면 접점을 ON 합니다. 총 6 개의 비교 명령이 있습니다.

비교 명령	동작
D= , s1, s2	s1 과 s2 가 같을 때 접점이 on 됩니다.
D<> , s1, s2	s1 과 s2 가 서로 다를 때 접점이 on 됩니다.
D>, s1, s2	s1 > s2 일 때 접점이 on 됩니다.
D<, s1, s2	s1 < s2 일 때 접점이 on 됩니다.
D>=, s1, s2	s1 >= s2 일 때 접점이 on 됩니다.
D<=, s1, s2	s1 <= s2 일 때 접점이 on 됩니다.

사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O

자세한 설명

더블워드 (32 비트 값)을 비교한다는 것을 제외한 모든 동작은 앞에서 설명한 "워드비교명령"과 동일 합니다.

제 4 장
TPC3X
응용명령어

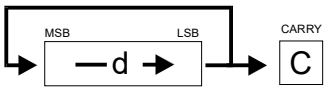
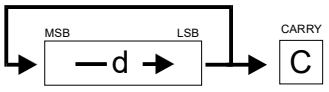
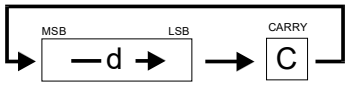
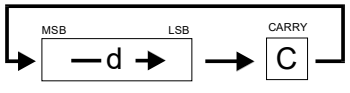
TPC3X 의 응용명령어에 대한 자세한 설명을 다루는 장입니다.

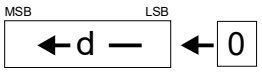
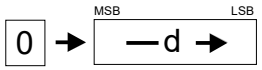
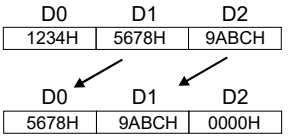
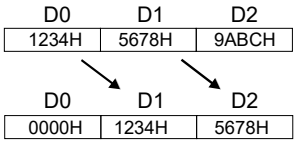
응용명령어

s 는 소스, d 는 목적지를 의미합니다.
n 은 갯수(또는 임의의 숫자)를 의미합니다.

분류	기능	명령어 형식	설 명
전송명령	Byte Move	MOVE s, d	8 비트 데이터 전송 (s) --> (d)
	Byte Complement Move	CMOVE s, d	8 비트 데이터 반전 전송 반전된(s) --> (d)
	Word Move	WMOV s, d	16 비트 데이터 전송 (s) --> (d)
	Double Word Move	DWMOV s,d	32 비트 데이터 전송 (s, s+1) ---> (d, d+1)
	Word Complement Move	WCMOV s,d	16 비트 데이터 반전 전송 반전된 (s) --> (d)
	Double Word Complement Move	DWCMOV s,d	32 비트 데이터 반전 전송 반전된 (s, s+1) --> (d, d+1)
	Word Negative Move	WNEG s,d	16 비트 2 의 보수(음수)화 전송 2 의 보수 (s) --> (d)
	Double Word Negative Move	DWNEG s,d	32 비트 2 의 보수(음수)화 전송 2 의 보수 (s, s+1) --> (d, d+1)
	Word Exchange Move	WXCHG s,d	16 비트 데이터의 교환 (s) <--> (d)
	Double Word Exchange Move	DWXCHG s,d	32 비트 데이터의 교환 (s, s+1) <--> (d, d+1)
	Fill Move	FMOV s,d,n	데이터 채움 명령 (s) ---> (d)부터 n 개
	Group Move	GMOV s,d,n	그룹 전송 명령 (s)부터 n 개 -----> (d)부터 n 개
변환명령	Word Binary to BCD code	WBCD s,d	16 비트 2 진 데이터를 4 Digit BCD 코드로 변환 (16bit Binary) ---> (4 digit BCD)
	Double Word Binary to BCD code	DWBCD s,d	32 비트 2 진 데이터를 8 Digit BCD 코드로 변환 (32bit Binary) ---> (8 digit BCD)
	Word BCD code to Binary	WBIN s,d	4 Digit BCD 코드를 16 비트 2 진 데이터로 변환 (4 digit BCD) ---> (16 bit Binary)

	Double Word BCD code to Binary	**DWBIN s,d**	8 Digit BCD 코드를 32 비트 2 진 데이터로 변환 (8 digit BCD) ----> (32 bit Binary)
증감 명령	Word Increment	**WINC d**	16 비트 1 증가 명령 (d) + 1 ----> (d)
	Double Word Increment	**DWINC d**	32 비트 1 증가 명령 (d, d+1) + 1 --> (d, d+1)
	Word Decrement	**WDEC d**	16 비트 1 감소 명령 (d) + 1 ----> (d)
	Double Word Decrement	**DWDEC d**	32 비트 1 감소 명령 (d, d+1) + 1 --> (d, d+1)
산술연산	Word Addition	**WADD s1, s2, d**	16 비트 덧셈 명령 (s1) + (s2) --> (d)
	Double Word Addition	**DWADD s1, s2, d**	32 비트 덧셈 명령 (s1, s1+1) + (s2, s2+1) --> (d, d+1)
	Word Subtraction	**WSUB s1, s2, d**	16 비트 뺄셈 명령 (s1) – (s2) --> (d)
	Double Word Subtraction	**DWSUB s1, s2, d**	32 비트 뺄셈 명령 (s1, s1+1) – (s2, s2+1) --> (d, d+1)
	Word Multiplication	**WMUL s1, s2, d**	16 비트 곱셈 명령 (결과는 32 비트) (s1) * (s2) --> (d, d+1)
	Word Division	**WDIV s1, s2, d**	16 비트 나눗셈 명령 (s1) / (s2) --> 몫은 (d), 나머지는 (d+1)
논리	Byte AND	**BAND s1, s2, d**	8 비트 AND 연산 (s1) AND (s2) --> (d)
	Word AND	**WAND s1, s2, d**	16 비트 AND 연산 (s1) AND (s2) --> (d)
	Double Word AND	**DWAND s1, s2, d**	32 비트 AND 연산 (s1, s1+1) AND (s2, s2+1) --> (d, d+1)
	Byte OR	**BOR s1, s2, d**	8 비트 OR 연산 (s1) OR (s2) --> (d)
	Word OR	**WOR s1, s2, d**	16 비트 OR 연산 (s1) OR (s2) --> (d)
	Double Word OR	**DWOR s1, s2, d**	32 비트 OR 연산 (s1, s1+1) OR (s2, s2+1) --> (d,

			d+1)
연	Byte XOR	**BXOR s1, s2, d**	8 비트 XOR 연산 (s1) XOR (s2) --> (d)
산	Word XOR	**WXOR s1, s2, d**	16 비트 XOR 연산 (s1) XOR (s2) --> (d)
	Double Word XOR	**DWXOR s1, s2, d**	32 비트 XOR 연산 (s1, s1+1) XOR (s2, s2+1) --> (d, d+1)
	Word Rotate Left	**WROL d**	
	Double Word Rotate Left	**DWROL d**	
회	Word Rotate Right	**WROR d**	
전	Double Word Rotate Right	**DWROR d**	
명	Word Rotate Left with Carry	**WRCL d**	
령	Double Word Rotate Left with Carry	**DWRCL d**	
	Word Rotate Right with Carry	**WRCR d**	
	Double Word Rotate Right with Carry	**DWRCR d**	

쉬프트 명령	Bit Shift Left	**BSHL d,n**	비트 Left 쉬프트 명령 (d)를 n 비트만큼 왼쪽으로 쉬프트 
	Bit Shift Right	**BSHR d,n**	비트 Right 쉬프트 명령 (d)를 n 비트만큼 오른쪽으로 쉬프트 
	Word Shift Left	**WSHL s1, s2**	워드 Left 쉬프트 명령 s1 에서 s2 까지를 1 워드씩 왼쪽으로 쉬프트 
	Word Shift Right	**WSHR s1, s2**	워드 Right 쉬프트 명령 s1 에서 s2 까지를 1 워드씩 오른쪽으로 쉬프트 
디스플레이 명령	Convert for 7 SEGment	**SEG s,d**	7 세그먼트를 위한 변환명령 s 에 있는 16 비트값을 7 세그먼트에 표시할 수 있는 4 바이트 데이터로 변환
	LCD area Clear	**LCDCLS**	CH 영역에 모두 SPACE (ASCII 코드 20H)를 채움 (LCD 화면이 클리어 됨)
	LCD data Output	**LCDOUT port,mode**	CH 영역에 있는 데이터를 LCD 로 송신 개시합니다.
	SGN data Output	**SGNOUT port**	G 영역에 있는 데이터를 SGN 으로 송신 개시합니다.
	String Store	**STRING d,"string"**	d 에 문자열 "string"을 저장합니다. "string" --> (d)
	Word Bin to ASCII HEX	**HEX s,d**	16 비트 2 진값을 16 진형태의 ASCII 코드로 변환합니다.

	format		(s) --> (d)로부터 4 바이트
	Double Word Bin to ASCII HEX format	**DHEX s,d**	32 비트 2 진값을 16 진형태의 ASCII 코드로 변환합니다. (s, s+1) --> (d)로부터 8 바이트
	Word Bin to ASCII DEC format	**ASC s,d,n**	16 비트 2 진값을 10 진형태의 ASCII 코드로 변환합니다. (s, s+1) --> (d)로부터 n 바이트
	Double Word Bin to ASCII DEC format	**DASC s,d,n**	32 비트 2 진값을 10 진형태의 ASCII 코드로 변환합니다. (s, s+1) --> (d)로부터 n 바이트
분 기 명 령	Goto, LABEL	**GOTO label** **LABEL label**	점프 명령 라벨 정의 명령
	Subroutine CALL, RET	**CALLS label** **SBRT label** **RET**	서브루틴 콜 명령 서브루틴 정의 리턴 명령
	Looping	**LOOP label, n**	반복 수행 명령, n 이 0 이 될때까지 label 로 점프
기 타 명 령	Key Metrix Scan	**KEYSCAN**	키 매트릭스 스캔 명령 최대 8 * 8 (64 키)까지 연결가능 맴브레인 키 판넬등을 연결할때 사용
	Output all off	**OUTOFF**	출력으로 설정된 전 I/O 포트 OFF
	Distribute	**DIST s,d,n**	16 비트 데이터를 4 비트씩 나누어서, d 에 저장
	Combine	**UNIT s,d,n**	하위 4 비트만 있는 여러개의 16 비트 데이터를 1 워드로 조합
	Encode	**ENCO s,d**	s 의 하위 4 비트를 엔코드해서 d 에 저장합니다.
	Decode	**DECO s,d**	s 의 16 비트 값을 디코드해서 d 의 하위 4 비트에 저장합니다.
	Thermometer Input	**THIN port, d**	디지털 온도센서 DS1820 으로 부터 현재 온도를 읽어와서 d 에 저장합니다.
	High Speed Count Clear	**HCNTCLR**	고속 카운터를 0 으로 만듭니다.

TPC3X 에서 취급하는 수에 대한 설명

응용명령어를 설명하기에 앞서, TPC3X 에서 취급하는 수의 체계와 각종 코드 (BCD 코드, ASCII 코드)에 대한 개념을 설명합니다. 수에는 2 진수, 10 진수, 16 진수의 3 가지 진법이 있으며, 다음과 같이 표기합니다.

10 진	2 진	16 진
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

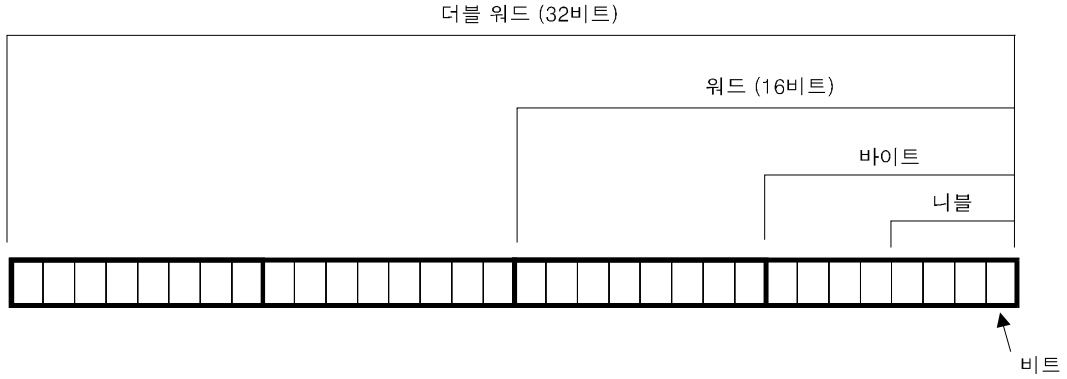
- 2 진수 4 자리를 하나의 16 진수로 표시할 수 있습니다.
- 16 진수에서 10~15 를 표기하기위해 알파벳 A~F 를 사용합니다.

레더 프로그램상에서 숫자 표기 방법

2 진수 표기 방법 : 00011010B
10 진수 표기 방법 : 123, 80
16 진수 표기 방법 : 0ABH, 12345H

영문자로 시작하는 16 진수는 반드시 맨 앞에 0 을 붙여야 합니다.
(예 : 0ABH, 0CDH, 0DEDEH)

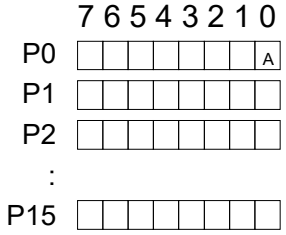
수의 크기를 구분하기 위한 가장 기본적인 단위로는 비트, 바이트, 워드, 더블워드가 있습니다.



비트는 1 과 0 을 나타낼 수 있는 가장 기본적인 단위입니다. 비트 4 개가 모이면 1 니블이 됩니다. 비트 8 개가 모이면 1 바이트, 16 개가 모이면 1 워드가 됩니다.

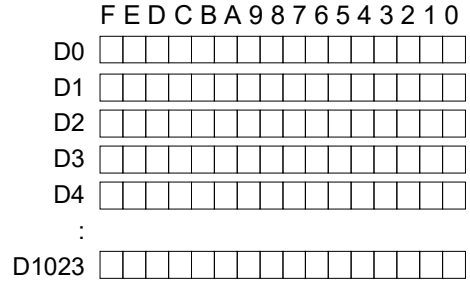
TPC3X 에는 바이트 단위로 구분하는 영역과 워드 단위로 구분되는 영역이 있습니다.

바이트 단위로 되어 있는 영역	워드 단위로 되어 있는 영역
P 영역 (입출력)	T 영역 (타이머)
K 영역 (KEEP)	C 영역 (카운터)
M 영역 (보조 릴레이)	D 영역 (데이터)
CH 영역 (LCD 디스플레이 데이터)	AD 영역 (AD 결과값)
G 영역 (SGN 데이터)	CNT 영역 (고속 카운터 값)

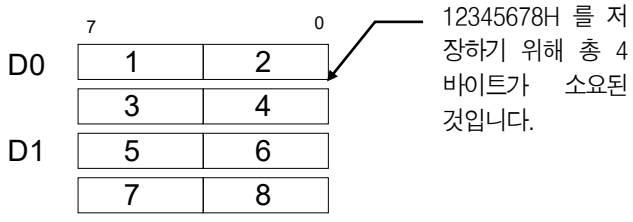


P 영역은 바이트 단위로 구분 되는 영역 입니다. P0:0 으로 표기하면 P0 바이트의 0 번 비트 (A 지점)가 됩니다. 따라서 P0:8 과 같은 표기는 잘못 된것 입니다. 8 번 비트 이후는 존재하지 않기 때문입니다.

데이터를 저장하는 D 영역은 워드 단위로 구분하고 있습니다.



TPC3X 에서의 모든 산술연산, 논리연산 명령은 워드단위가 기본입니다. 그리고 2 개의 워드를 합친 "더블워드"를 취급하는 명령도 있습니다. 더블워드 취급 시 지정한 워드 번호의 바로 다음 워드번호까지 함께 사용됩니다.
예를 들어 DWMOV 12345678H, D0 (D0 에 12345678H 를 저장하는 명령)을 실행 시키면 D0 번지와 D1 번지가 사용됩니다. **데이터는 상위바이트부터 차례대로 저장**됩니다. 다음 그림은 D0 번지 (더블워드)에 12345678H 가 저장된 상태를 그림으로 표현한 것입니다.



ASCII 코드

ASCII 코드란 디스플레이, 프린트 목적으로 사용되는 코드 체계로써, 1 바이트가 숫자 하나를 의미합니다. 다음은 ASCII 코드표 입니다. (읽는 방법은 상위 비트부터 읽고, 하위비트를 읽으면 됩니다. 숫자 8 의 ASCII 코드는 38H 입니다.)

		하 위 4 비 트															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
상	2		!	“	#	$	%	&	‘	(	)	*	+	,	−	.	/
위	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
비	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	₩	]	^	_
트	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	7	p	q	r	s	t	u	v	w	x	y	z	{	|	}		

BCD 코드

BCD (2 진화 10 진수)코드는 2 진수 4 BIT 로, 10 진수 (0 ~ 9) 를 표시하도록 한 코드 체계입니다. (4 비트단위로 끊어서 10 진수 1 자리를 표시합니다.) 1 바이트는 2 digit 의 10 진수를 표시할 수 있고, 2 바이트로는 4 digit 의 10 진수를 표시할 수 있습니다. 다음은 BCD 코드 표입니다.

10 진수	BCD 코드
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

10 진수를 2 진수로 표시한 것과 비슷해 보이지만 다음 예를 보면 2 진수와 BCD 코드간의 차이점을 확실히 알 수 있습니다.

10 진수 (본래의 숫자)	2 진수로 변환하면	BCD 코드로 변환하면
100	01100100 (64H)	000100000000 (100H)

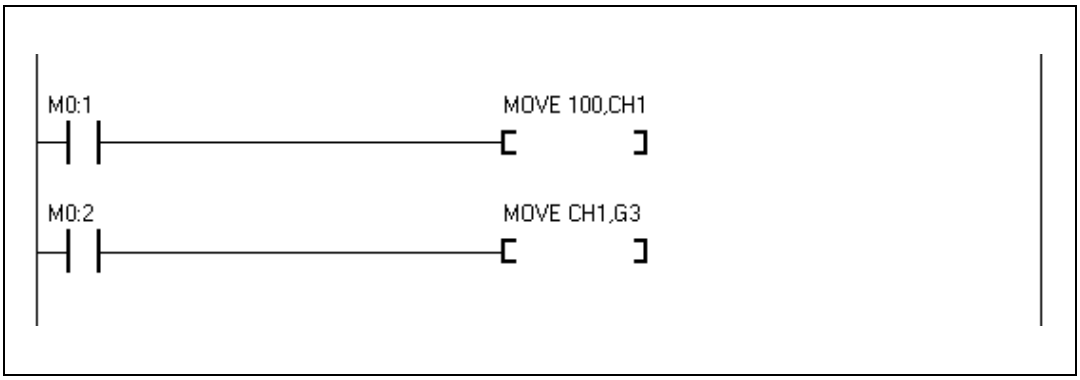
MOVE 8 비트 전송

Usage : MOVE s, d

개요

8 비트 데이터 전송 명령입니다. s 에 들어있는 값 (또는 8 비트 상수)을 d 에 전송합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 CH1 에 100 을 전송하고, M0:2 가 ON 되면 CH1 값을 G3 에 전송합니다. 상수 사용시 255 이상의 값은 사용할 수 없습니다. (바이트는 0~255 까지 표현 가능)

C, T, D 영역 사용 시 주의사항 :

MOVE 명령은 바이트 단위 전송 명령이고, C,T,D 영역은 워드단위로 되어 있기 때문에 약간의 주의를 요합니다. "MOVE 100, D0"이라고 명령하면, D0 의 상위바이트에 100 이 저장되는 것이기 때문에 D0 전체의 값은 엉뚱한 값이 됩니다. 이 경우에는 뒤에서 설명하는 WMOV 명령 (워드 단위 전송)을 사용해야 합니다.

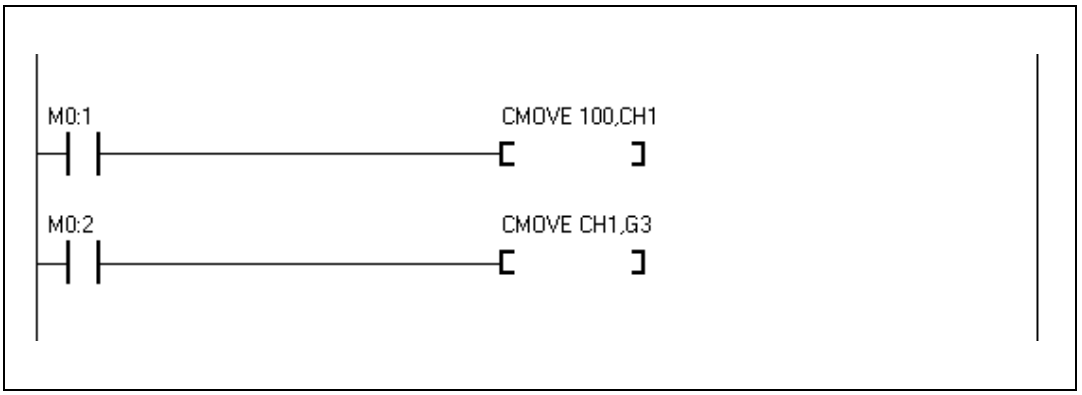
CMOVE 8 비트 반전 전송

Usage : CMOVE s, d

개요

8 비트 데이터 반전 전송 명령입니다. s 에 들어있는 값 (또는 8 비트 상수)을 반전한 뒤 d 에 전송합니다.

프로그램 예

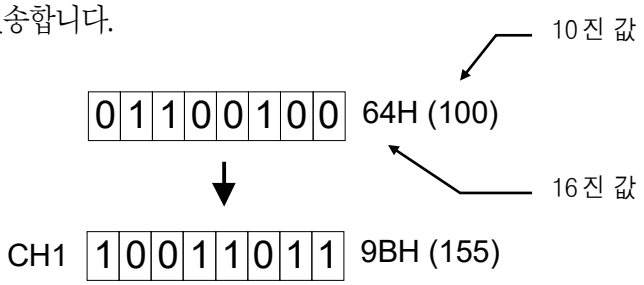


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 CH1 에 100 을 반전한 값 (155)를 전송합니다. M0:2 가 ON 되면 CH1 값을 반전해서 G3 에 전송합니다.



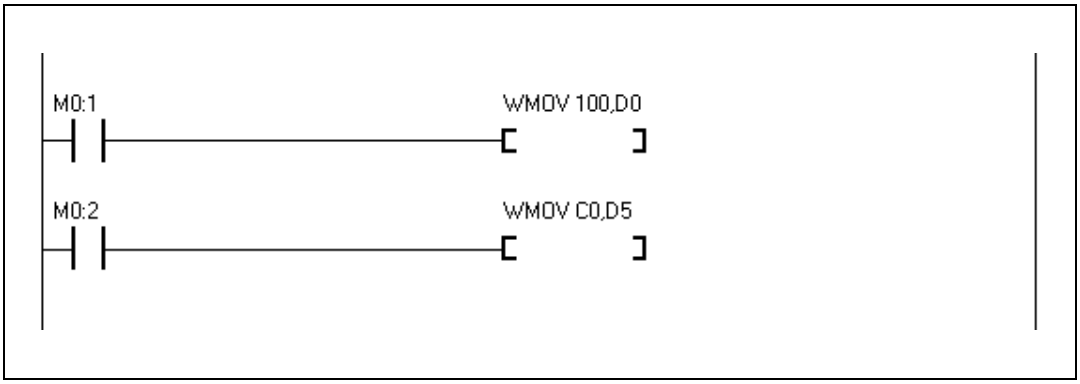
WMOV 16 비트 전송

Usage : WMOV s, d

개요

16 비트 데이터 전송 명령으로, s 에 들어있는 값 (또는 16 비트 상수)을 d 에 전송합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0 에 100 을 전송합니다. M0:2 가 ON 되면 C0 값을 D5 에 전송합니다. 상수 사용 시 65535 이상의 값은 사용할 수 없습니다. (워드는 0~65535 까지 표현 가능)

바이트 영역 사용 시 주의사항 :

WMOV 명령은 워드 단위 전송 명령이므로 바이트 단위 영역 (P, M, K 등)을 취급할 때에는 약간의 주의가 필요합니다. WMOV 명령을 실행하면 2 바이트를 대상으로 명령이 수행됩니다. 즉 "WMOV 1234H, CH0"라고 명령하면, CH0 에 12H, CH1 에 34H 가 저장됩니다.

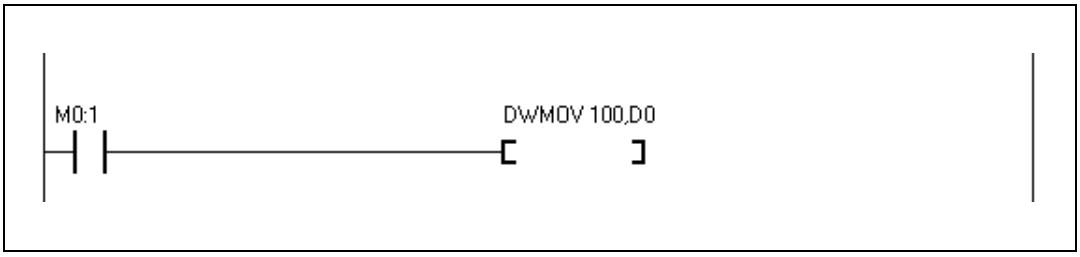
DWMOV 32 비트 전송

Usage : DWMOV s, d

개요

32 비트 데이터 전송 명령입니다. s 에 들어있는 값 (또는 32 비트 상수)을 d 에 전송합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0 에 0, D1 에 100 을 전송합니다. 결과는 옆에 그림처럼 D1 에만 100 이 들어가고 D0 에는 0 이 들어갑니다. (상위 바이트부터 저장됨)

상수 사용 시 2,147,418,112 이상 (16 진수 7FFF0000H)의 값은 사용할 수 없습니다.

	HIGH byte	LOW byte
D0	00H	00H
D1	00H	64H
D2		
D3		

바이트, 워드, 더블워드 데이터 저장에 대하여...

데이터 전송명령은 3 가지 종류가 있습니다.

분류	전송명령	반전전송 명령
바이트 단위	MOVE	CMOVE
워드 단위	WMOV	WCMOV
더블워드 단위	DWMOV	DWCMOV

16 진수 12H 를 각각의 명령으로 D0 에 전송해보면, 결과는 아래 그림처럼 됩니다.

MOVE 100,D0

	HIGH byte	LOW byte
D0	64H	
D1		
D2		
D3		

WMOV 100,D0

	HIGH byte	LOW byte
D0	00H	64H
D1		
D2		
D3		

DWMOV 100,D0

	HIGH byte	LOW byte
D0	00H	00H
D1	00H	64H
D2		
D3		

상위 값부터 저장되는 것에 유의 !

각 단위별로 사용 가능한 상수의 범위는 아래와 같습니다.

분류	범위(10 진)	범위(16 진)
바이트 단위	0~255	0~0FFH
워드 단위	0~65535	0~0FFFFH
더블워드 단위	0~2147418112	0~7FFF0000H

참고사항

워드 명령은 W 로 시작합니다. (예 : WMOV)
더블워드 명령은 DW 로 시작합니다. (예 : DWMOV)

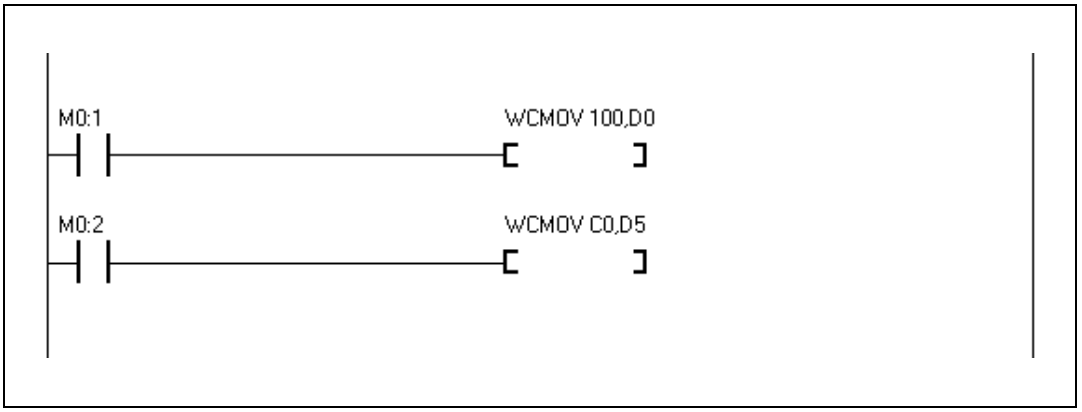
WCMOV 16 비트 반전 전송

Usage : WCMOV s, d

개요

16 비트 반전 전송 명령입니다. s 에 들어있는 값 (또는 16 비트 상수)을 반전해서 d 에 전송합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0 에 65435 (100 의 반전된 값)을 전송합니다. M0:2 가 ON 되면 C0 값을 반전해서 D5 에 전송합니다. 상수 사용 시 65535 이상의 값은 사용할 수 없습니다. (워드는 0~ 65535 까지 표현 가능)

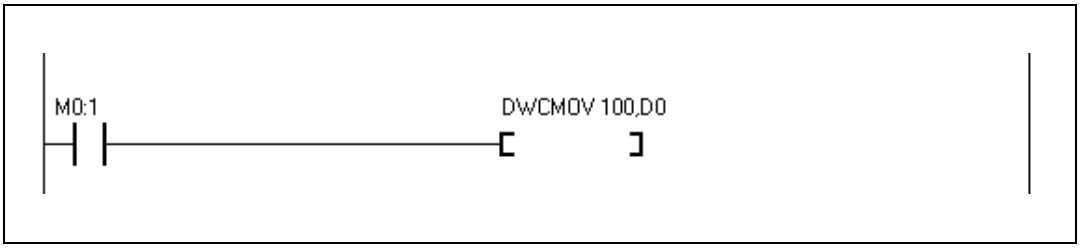
DWCMOV 32 비트 반전 전송

Usage : DWCMOV s, d

개요

32 비트 반전 전송 명령입니다. s 에 들어있는 값 (또는 32 비트 상수)을 반전해서 d 에 전송합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0 에 0, D1 에 반전된 100 을 전송합니다. 결과는 옆에 그림과 같습니다.

	HIGH byte	LOW byte
D0	0FFH	0FFH
D1	0FFH	09BH
D2		
D3		

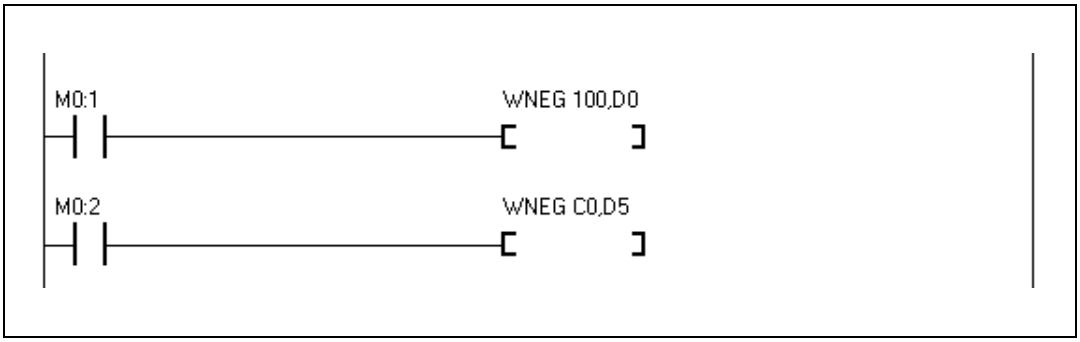
WNEG 16 비트 2 의 보수 전송

Usage : WNEG s, d

개요

16 비트 2 의 보수화 전송 명령입니다. s 에 들어있는 값 (또는 16 비트 상수)를 2 의 보수로 변환하여 d 에 전송합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0 에 -100 (2 의 보수화 된 100)을 전송합니다. M0:2 가 ON 되면 C0 값을 2 의 보수로 변환하여 D5 에 전송합니다.

2 의 보수란?

컴퓨터에서는 2 의 보수는 곧 음수를 의미합니다. 정상적인 수를 반전시켜 1 을 더한 값이 2 의 보수입니다. (1 의 2 의 보수는 0FFFFH 입니다.) 1 + (-1) = 0 이 되듯이, 1 + 0FFFFH 도 0 이 됩니다. 이런 원리때문에 2 의 보수를 이용하면 모든 수의 음수표현이 가능한 것입니다. TPC3X 에서도 특별한 음수처리명령이 없지만, 2 의 보수를 이용하면, 음수 처리 및 표현이 가능합니다.

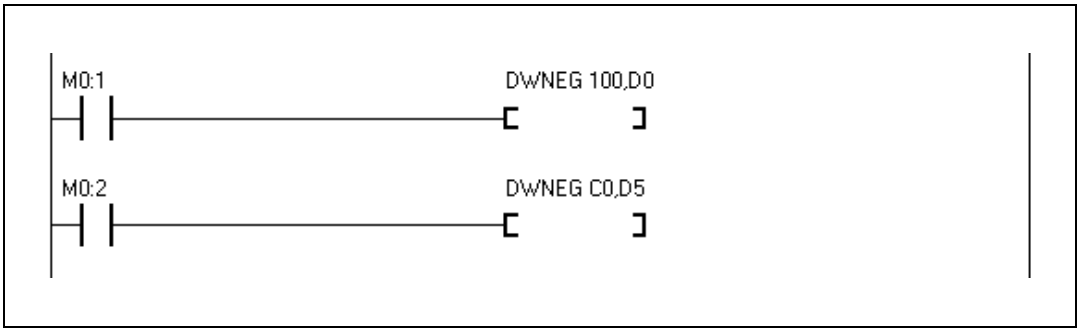
DWNEG 32 비트 2 의 보수 전송

Usage : DWNEG s, d

개요

32 비트 2 의 보수화 전송 명령입니다. s 에 들어있는 값 (또는 32 비트 상수)를 2 의 보수로 변환하여 d 에 전송합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0, D1 에 -100 (2 의 보수화 된 100)을 전송합니다. M0:2 가 ON 되면 C0,C1 값을 2 의 보수로 변환하여 D5, D6 에 전송합니다.

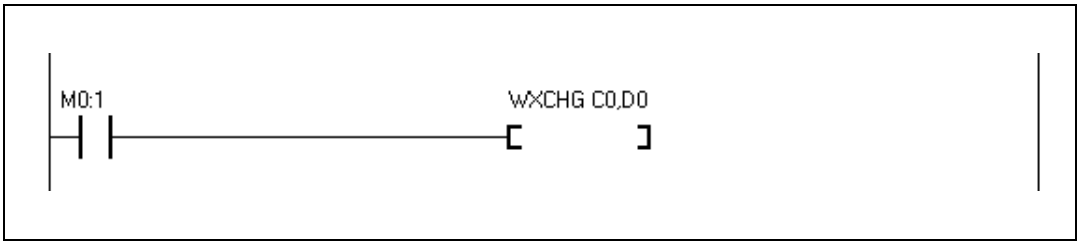
WXCHG 16 비트 데이터 교환

Usage : WXCHG s, d

개요

16 비트 데이터 교환 명령 입니다. s 에 들어있는 값과 d 에 있는 값을 서로 교환합니다.

프로그램 예

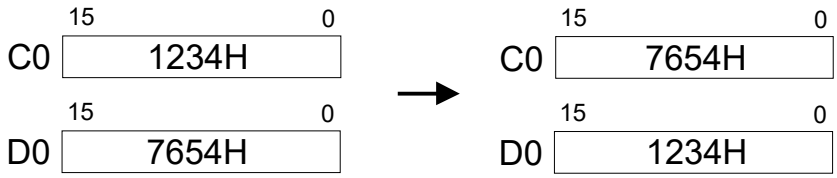


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0 에 있는 값과 C0 에 있는 값을 서로 맞 교환 합니다.



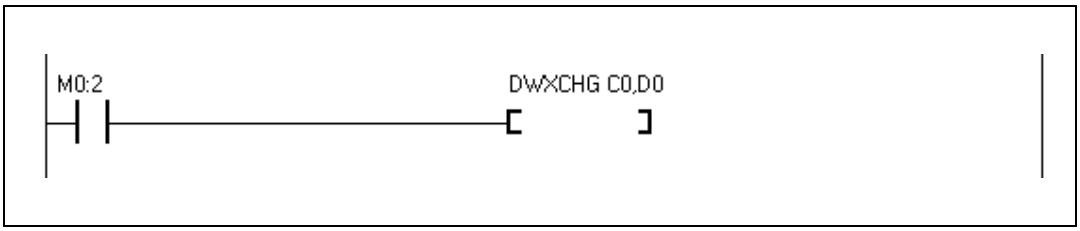
DWXCHG 32 비트 데이터 교환

Usage : DWXCHG s, d

개요

32 비트 데이터 교환 명령 입니다. s 에 들어있는 값과 d 에 있는 값을 서로 교환합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0,D1 에 있는 값과 C0,C1 에 있는 값을 서로 맞 교환 합니다.

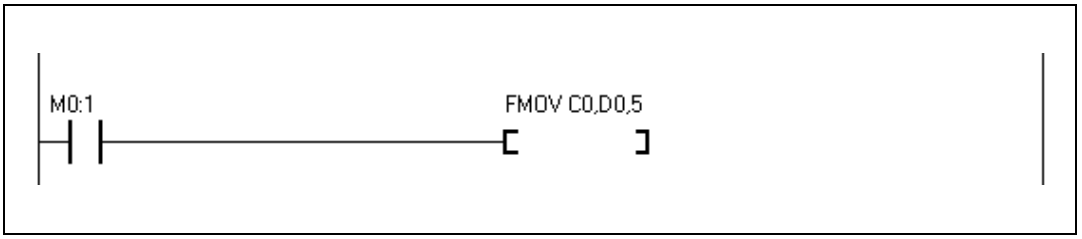
FMOV 데이터 채움

Usage : FMOV s, d, n

개요

s 의 값 (또는 16 비트 상수 값)을 d 를 선두번지로 해서 지정된 개 수 n 만큼 같은 값으로 저장합니다.

프로그램 예

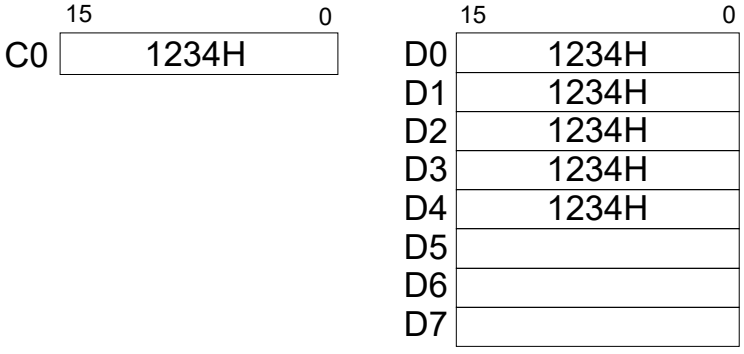


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	
n(갯수)													O

자세한 설명

M0:1 이 ON 되면 C0 에 있는 값 (16 비트 값)을 D0, D1, D2, D3, D4 에 저장합니다.



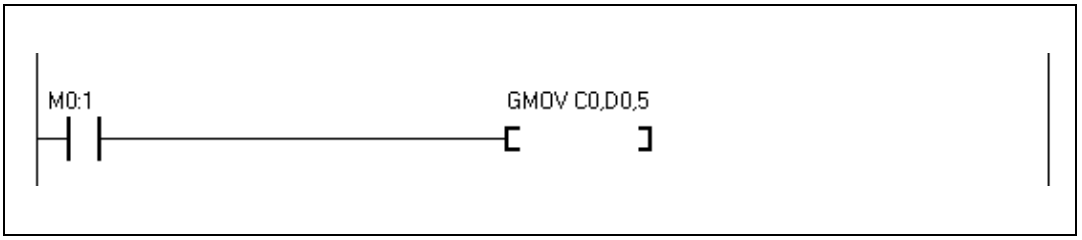
GMOV 그룹 전송

Usage : GMOV s, d, n

개요

여러 개의 데이터를 한번에 전송하는 그룹 전송 명령 입니다. s 번지로부터 값을 읽어 d 번지에 지정된 갯수 n 만큼 전송합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	
d (목적지)	O	O		O		O	O	O	O	O	O	O	
n (갯수)													O

자세한 설명

M0:1 이 ON 되면 C0-->D0, C1-->D1, C2-->D2, C3-->D3, C4-->D4 로 전송합니다.

	HIGH byte	LOW byte
C0	00H	64H
C1	01H	65H
C2	02H	66H
C3	03H	67H
C4	04H	68H
C5	05H	69H
C6		
C7		

→

	HIGH byte	LOW byte
D0	00H	64H
D1	01H	65H
D2	02H	66H
D3	03H	67H
D4	04H	68H
D5	05H	69H
D6		
D7		

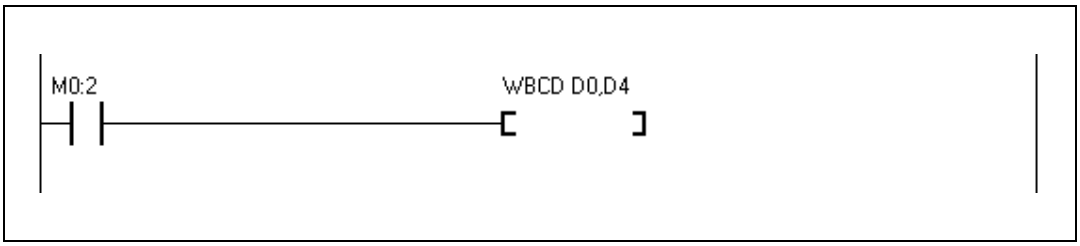
WBCD 16 비트 BIN to BCD 변환

Usage : WBCD s, d

개요

16 비트의 2 진 데이터를 4 digit BCD 데이터로 변환 합니다.

프로그램 예

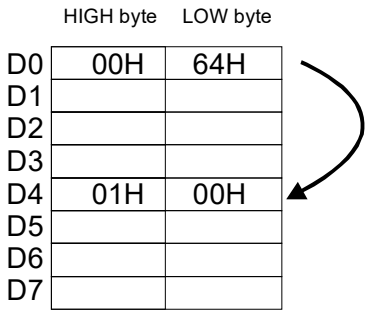


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:2 이 ON 되면 D0 에 있는 값을 4digit BCD 코드로 변환해서 D4 에 저장합니다.



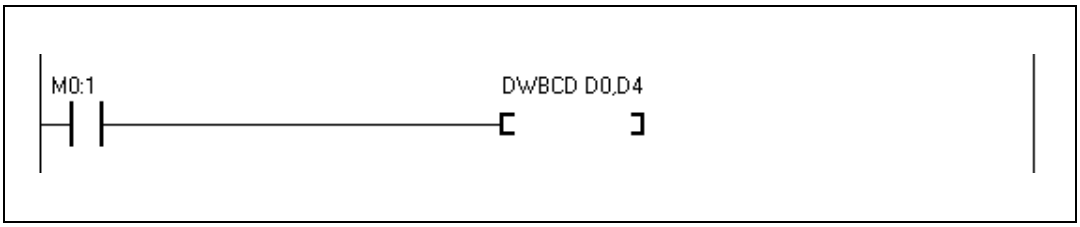
DWBCD 32 비트 BIN to BCD 변환

Usage : DWBCD s, d

개요

32 비트의 2 진 데이터를 8 digit BCD 데이터로 변환 합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:1 이 ON 되면 D0, D1 에 있는 값을 8digit BCD 코드로 변환해서 D4 부터 저장합니다.
(0BC614EH 는 10 진수로 12345678 입니다.)

	HIGH byte	LOW byte
D0	00H	BCH
D1	61H	4EH
D2		
D3		
D4	12H	34H
D5	56H	78H
D6		
D7		

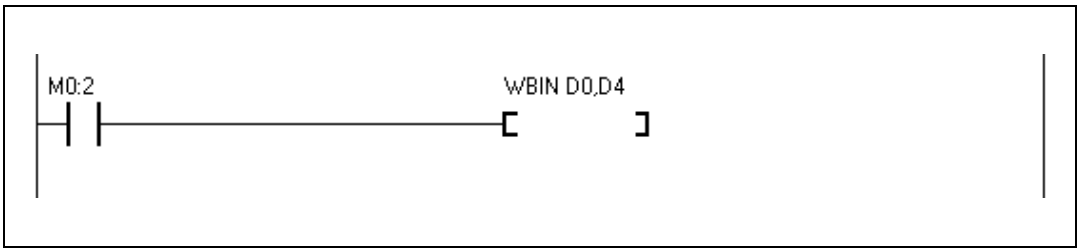
WBIN 16 비트 BCD to BIN 변환

Usage : WBIN s, d

개요

4 digit BCD 데이터를 16 비트의 2 진 데이터로 변환 합니다.

프로그램 예

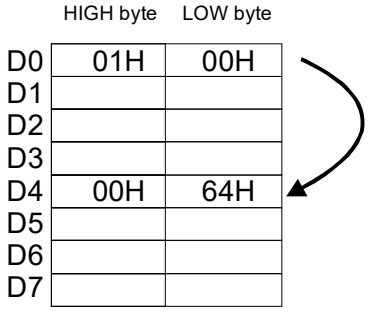


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:2 이 ON 되면 D0 에 있는 4digit BCD 코드를 2 진 값으로 변환해서 D4 에 저장합니다.



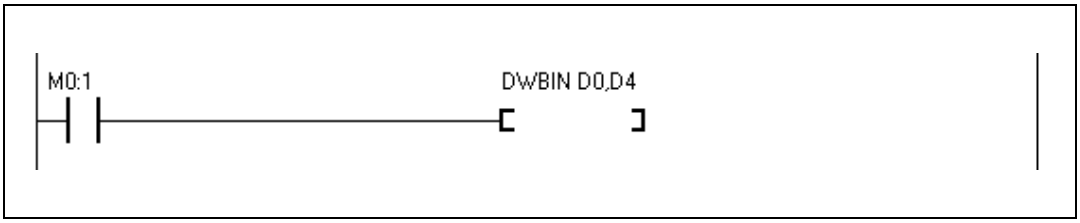
DWBIN 32 비트 BCD to BIN 변환

Usage : DWBIN s, d

개요

8 digit BCD 데이터를 32 비트의 2 진 데이터로 변환 합니다.

프로그램 예

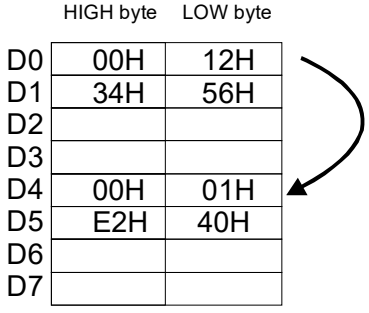


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s (소스)	O	O		O		O	O	O	O	O	O	O	O
d (목적지)	O	O		O		O	O	O	O	O	O	O	

자세한 설명

M0:2 이 ON 되면 D0 에 있는 8digit BCD 코드를 2 진 값으로 변환해서 D4 에서 부터 저장합니다.



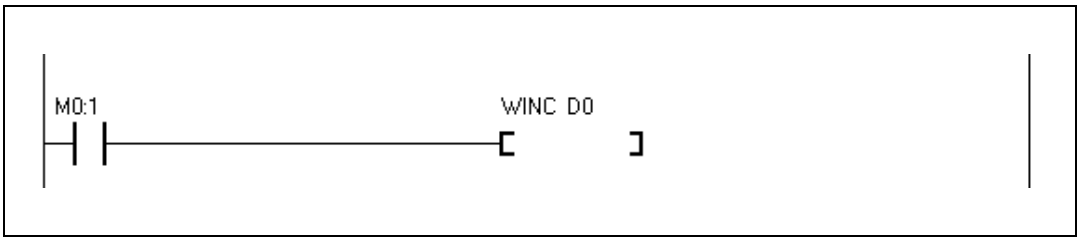
WINC 16 비트 1 증가

Usage : WINC d d = d + 1

개요

16 비트 1 증가 명령입니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d						O	O	O	O	O			

자세한 설명

M0:1 이 ON 되면 D0 가 1 증가 됩니다.

(D0 = D0 + 1)

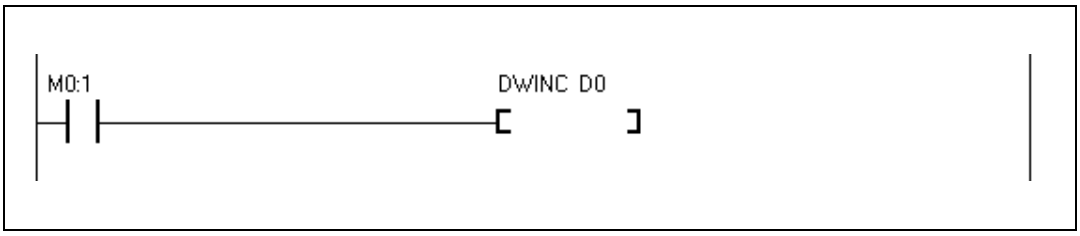
DWINC 32 비트 1 증가

Usage : DWINC d d = d + 1

개요

32 비트 1 증가 명령입니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d						O	O	O	O	O			

자세한 설명

M0:1 이 ON 되면 D0,D1 이 1 증가 됩니다.

(D0 ,D1 = D0,D1 + 1)

WDEC

16 비트 1 감소

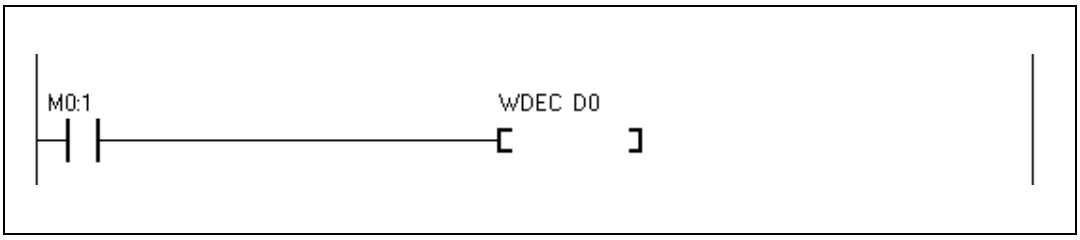
Usage : WDEC d

d = d - 1

개요

16 비트 1 감소 명령입니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d						O	O	O	O	O			

자세한 설명

M0:1 이 ON 되면 D0 가 1 감소 됩니다.

(D0 = D0 - 1)

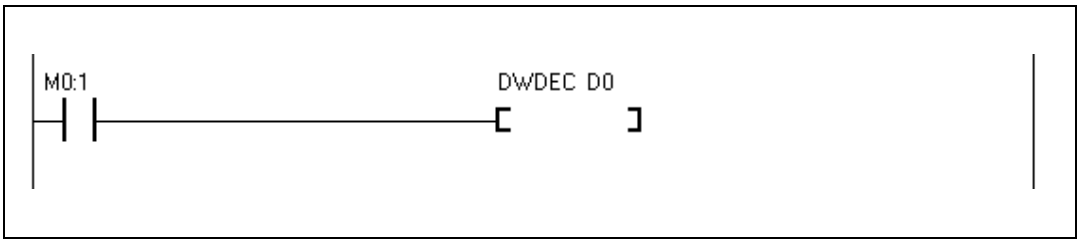
DWDEC 32 비트 1 감소

Usage : DWDEC d d = d - 1

개요

32 비트 1 감소 명령입니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d						O	O	O	O	O			

자세한 설명

M0:1 이 ON 되면 D0,D1 이 1 감소 됩니다.

(D0 ,D1 = D0, D1 - 1)

WADD

16 비트 덧셈

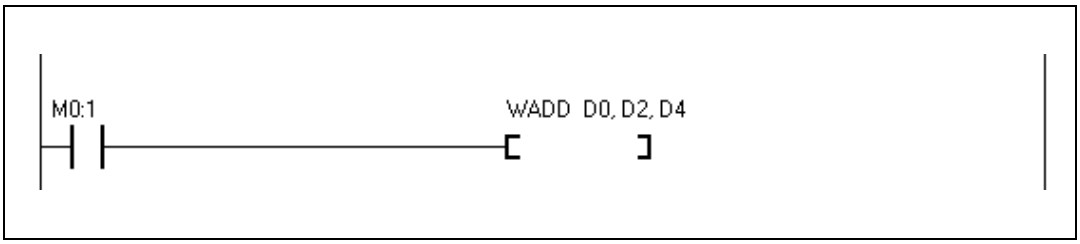
Usage : WADD s1,s2, d

d = s1 + s2

개요

16 비트 덧셈 명령입니다. s1 과 s2 를 더해서 그 결과를 d 에 저장합니다.

프로그램 예

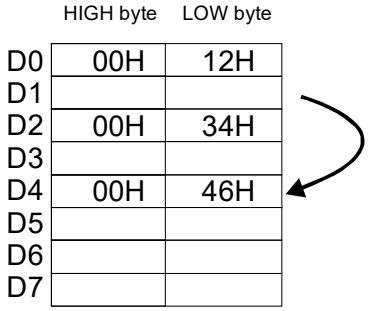


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0 과 D2 를 더해서 그 결과값을 D4 에 저장합니다.



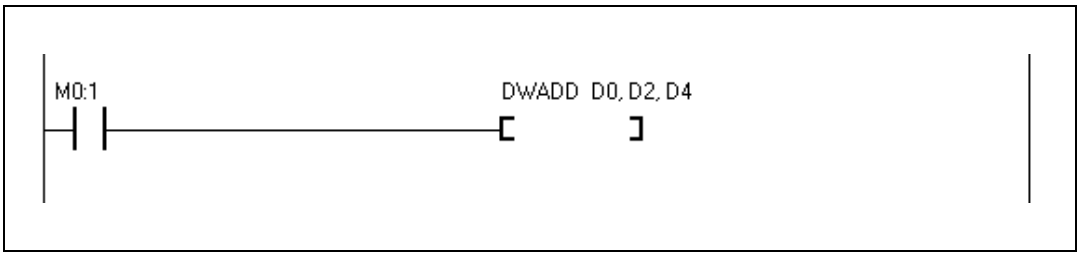
DWADD 32 비트 덧셈

Usage : DWADD s1,s2, d d = s1 + s2

개요

32 비트 덧셈 명령입니다. s1 과 s2 를 더해서 그 결과를 d 에 저장합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0,D1 과 D2,D3 를 더해서 그 결과값을 D4,D5 에 저장합니다.

	HIGH byte	LOW byte
D0	00H	01H
D1	03H	12H
D2	00H	02H
D3	04H	34H
D4	00H	03H
D5	07H	46H
D6		
D7		

WSUB

16 비트 뺄셈

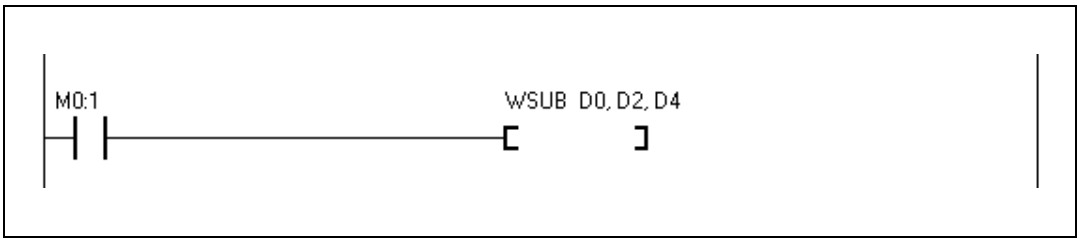
Usage : WSUB s1,s2, d

d = s1 - s2

개요

16 비트 뺄셈 명령입니다. s1 에서 s2 를 뺀뒤 그 결과를 d 에 저장합니다.

프로그램 예

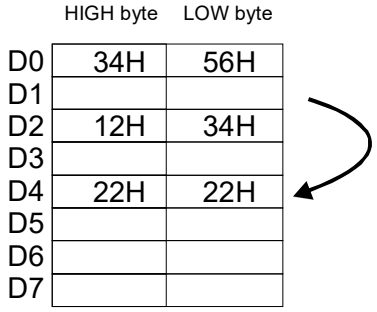


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0 에서 D2 를 뺀뒤 그 결과값을 D4 에 저장합니다.



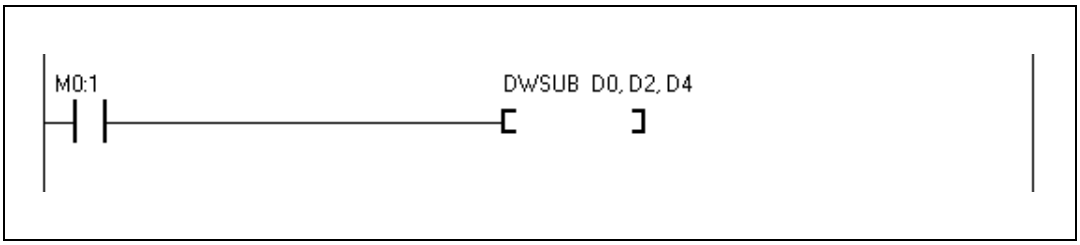
DWSUB 32 비트 뺄셈

Usage : DWSUB s1,s2, d d = s1 - s2

개요

32 비트 뺄셈 명령입니다. s1 에서 s2 를 뺀뒤 그 결과를 d 에 저장합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0,D1 에서 D2,D3 를 뺀뒤 그 결과값을 D4,D5 에 저장합니다.

	HIGH byte	LOW byte
D0	34H	56H
D1	78H	9AH
D2	12H	34H
D3	56H	78H
D4	22H	22H
D5	22H	22H
D6		
D7		

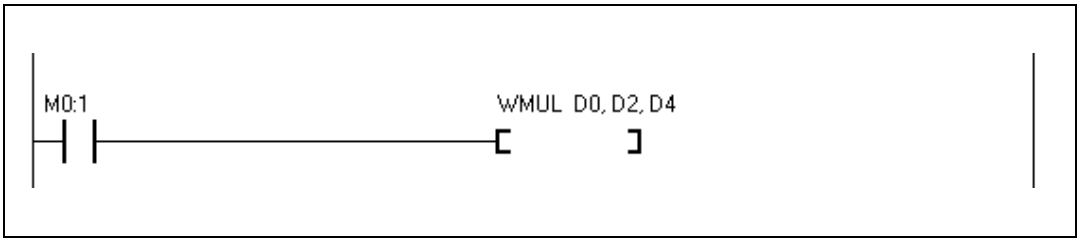
WMUL 16 비트 곱셈

Usage : WSUB s1,s2, d d = s1 * s2

개요

16 비트 곱셈 명령입니다. s1 에서 s2 를 곱한 결과를 d 에 32 비트로 저장합니다.

프로그램 예

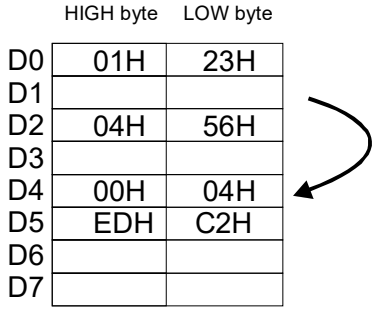


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0 에서 D2 를 곱한 값을 32 비트로 D4 에 저장합니다. (결과는 더블워드가 되므로 주의가 필요합니다.)



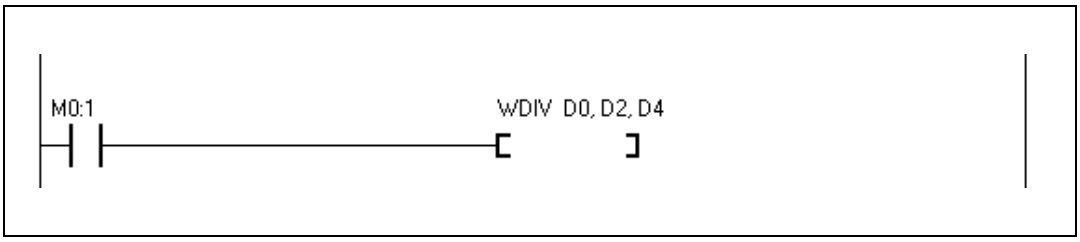
WDIV 16 비트 나눗셈

Usage : WDIV s1,s2, d d = s1 / s2

개요

16 비트 나눗셈 명령입니다. s1 에서 s2 를 나눈 몫을 d 에, 나머지를 d+ 1 에 저장합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0 에서 D2 를 나눈 결과 (몫)을 D4 에, 나머지는 D5 에 저장합니다.

	HIGH byte	LOW byte	
D0	12H	34H	
D1			
D2	00H	03H	
D3			
D4	06H	11H	몫
D5	00H	01H	나머지
D6			
D7			

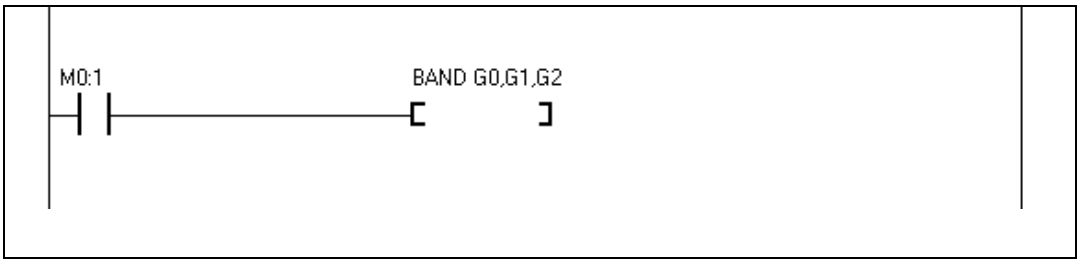
BAND 8 비트 AND 연산

Usage : BAND s1,s2, d d = s1 and s2

개요

8 비트 AND 연산 명령입니다. s1 와 s2 를 AND 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예

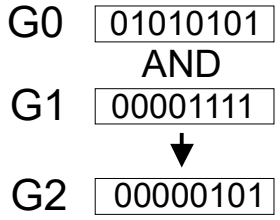


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 G0 와 G1 를 AND 연산한 뒤 그 결과값을 G2 에 저장합니다.



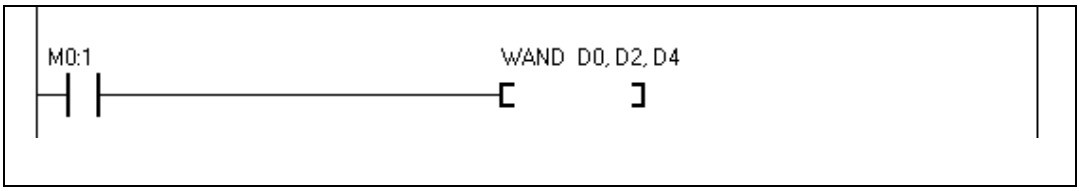
WAND 16 비트 AND 연산

Usage : WAND s1,s2, d d = s1 and s2

개요

16 비트 AND 연산 명령입니다. s1 와 s2 를 AND 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예

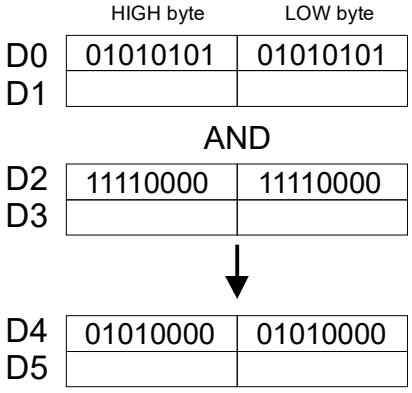


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0 와 D2 를 AND 연산한 뒤 그 결과값을 D4 에 저장합니다.



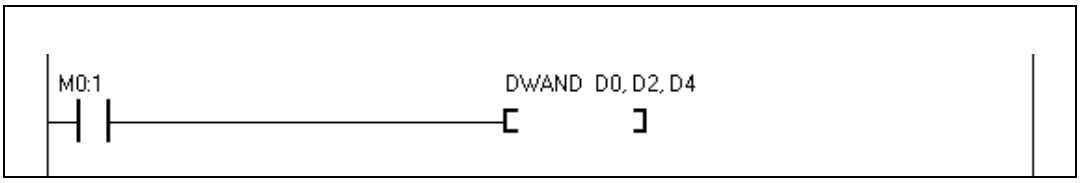
DWAND 32 비트 AND 연산

Usage : DWAND s1,s2, d d = s1 and s2

개요

32 비트 AND 연산 명령입니다. s1 와 s2 를 AND 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0,D1 와 D2,D3 를 AND 연산한 뒤 그 결과값을 D4,D5 에 저장합니다.

	HIGH byte	LOW byte
D0	01010101	01010101
D1	01010101	01010101

AND

D2	11110000	11110000
D3	00001111	00001111

↓

D4	01010000	01010000
D5	00000101	00000101

BOR

8 비트 OR 연산

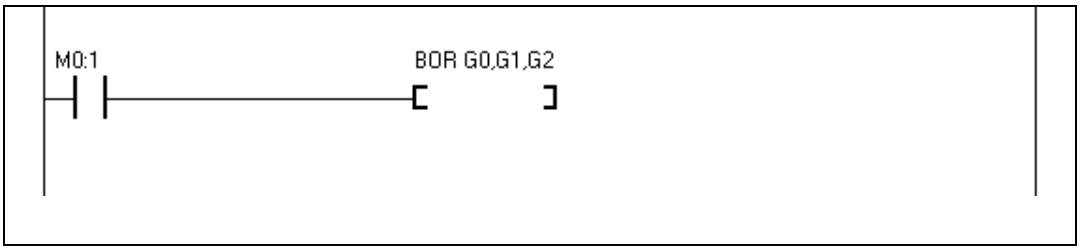
Usage : BOR s1,s2, d

d = s1 or s2

개요

8 비트 OR 연산 명령입니다. s1 와 s2 를 OR 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예

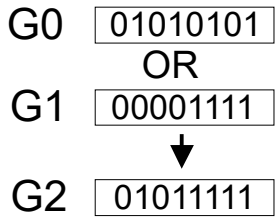


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 G0 와 G1 를 OR 연산한 뒤 그 결과값을 G2 에 저장합니다.



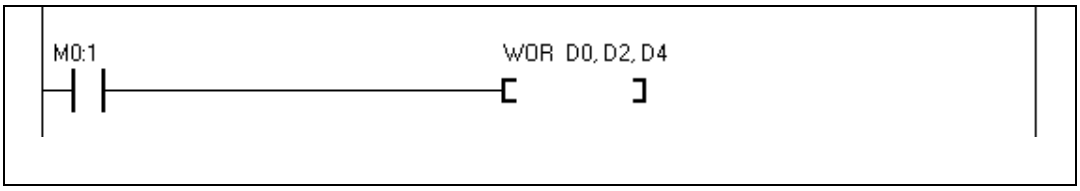
WOR 16 비트 OR 연산

Usage : WOR s1,s2, d d = s1 or s2

개요

16 비트 OR 연산 명령입니다. s1 와 s2 를 OR 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예

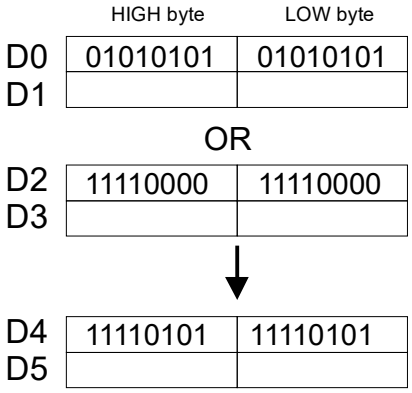


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0 와 D2 를 OR 연산한 뒤 그 결과값을 D4 에 저장합니다.



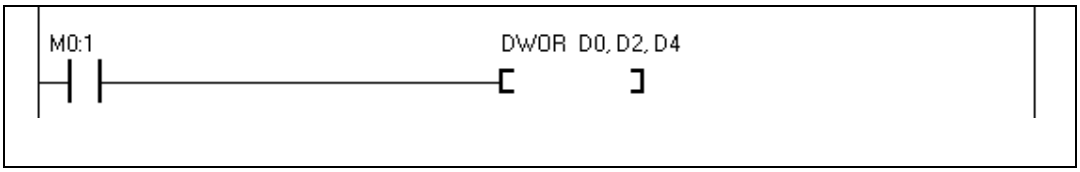
DWOR 32 비트 OR 연산

Usage : DWOR s1,s2, d d = s1 or s2

개요

32 비트 OR 연산 명령입니다. s1 와 s2 를 OR 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예

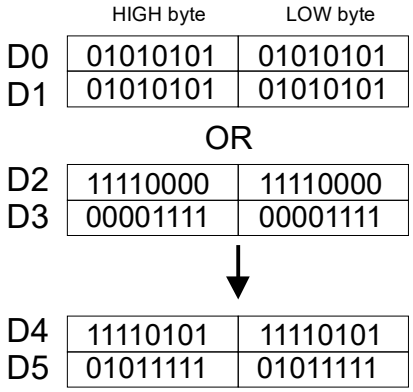


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0,D1 와 D2,D3 를 OR 연산한 뒤 그 결과값을 D4,D5 에 저장합니다.



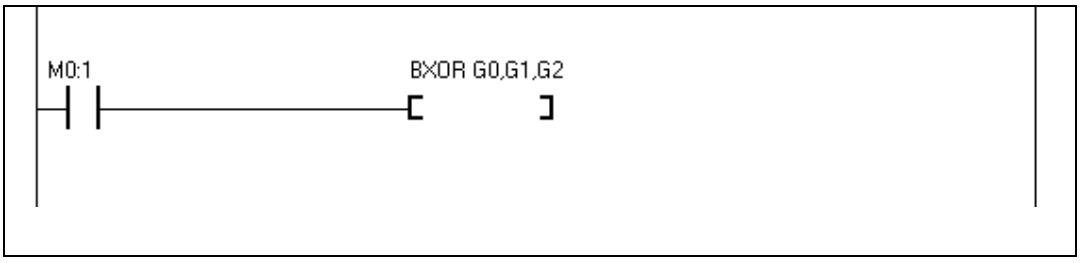
BXOR 8 비트 XOR 연산

Usage : BXOR s1,s2, d d = s1 xor s2

개요

8 비트 XOR 연산 명령입니다. s1 와 s2 를 XOR 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예

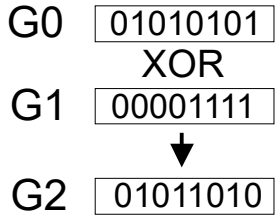


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 G0 와 G1 를 XOR 연산한 뒤 그 결과값을 G2 에 저장합니다.



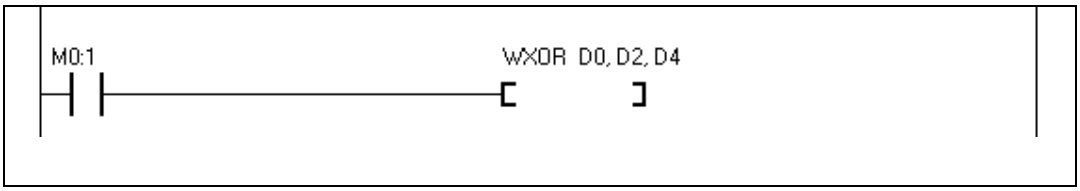
WXOR 16 비트 XOR 연산

Usage : WXOR s1,s2, d d = s1 xor s2

개요

16 비트 XOR 연산 명령입니다. s1 와 s2 를 XOR 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예

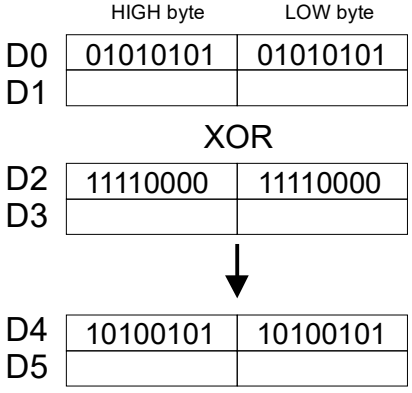


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0 와 D2 를 XOR 연산한 뒤 그 결과값을 D4 에 저장합니다.



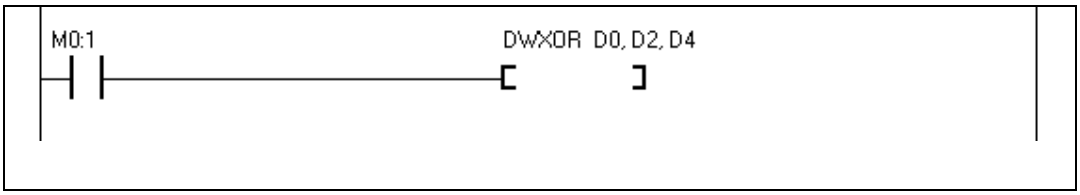
DWXOR 32 비트 XOR 연산

Usage : DWXOR s1,s2, d d = s1 xor s2

개요

32 비트 XOR 연산 명령입니다. s1 와 s2 를 XOR 한 뒤 그 결과를 d 에 저장합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1						O	O	O	O	O			O
s2						O	O	O	O	O			O
d						O	O	O					

자세한 설명

M0:1 이 ON 되면 D0,D1 와 D2,D3 를 XOR 연산한 뒤 그 결과값을 D4,D5 에 저장합니다.

	HIGH byte	LOW byte
D0	01010101	01010101
D1	01010101	01010101
	XOR	
D2	11110000	11110000
D3	00001111	00001111
	↓	
D4	10100101	10100101
D5	01011010	01011010

WROL

16 비트 좌회전

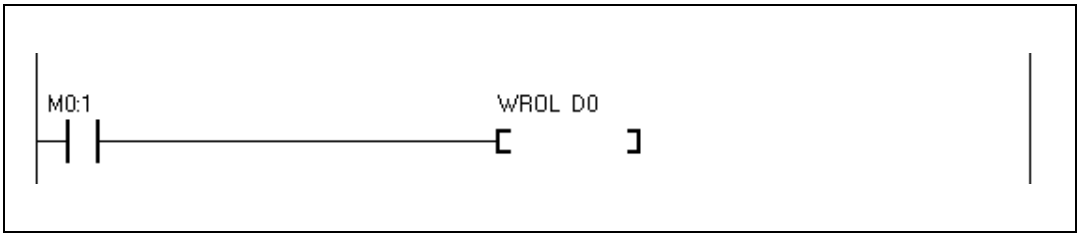
Usage : WROL d

d = d << 1

개요

16 비트 좌회전 명령입니다. d 를 한 비트씩 왼쪽으로 회전 시킵니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0 을 한 비트씩 왼쪽으로 회전시킵니다. 최상위 비트(MSB)에 있던 값은 CARRY 비트로 이동합니다.

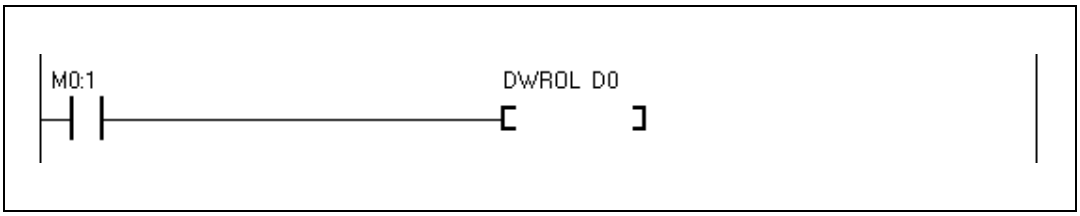
DWROL 32 비트 좌회전

Usage : DWROL d d = d << 1

개요

32 비트 좌회전 명령입니다. d 를 한 비트씩 왼쪽으로 회전 시킵니다.

프로그램 예

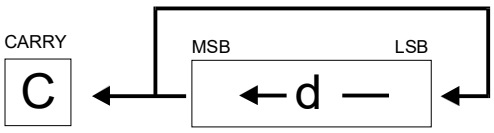


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0,D1 을 한 비트씩 왼쪽으로 회전시킵니다. 최상위 비트(MSB)에 있던 값은 CARRY 비트로 이동합니다.



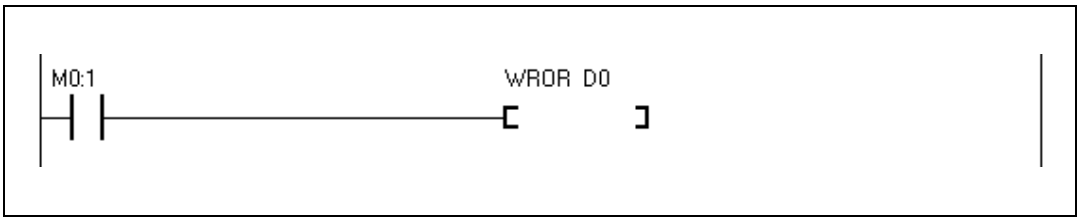
WROR 16 비트 우회전

Usage : WROR d d = d >> 1

개요

16 비트 우회전 명령입니다. d 를 한 비트씩 오른쪽으로 회전 시킵니다.

프로그램 예

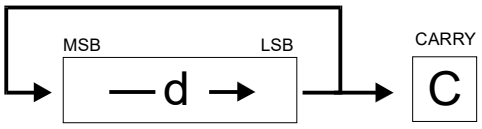


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O	O			O	

자세한 설명

M0:1 이 ON 되면 D0 을 한 비트씩 오른쪽으로 회전시킵니다. 최하위 비트(LSB)에 있던 값은 CARRY 비트로 이동합니다.



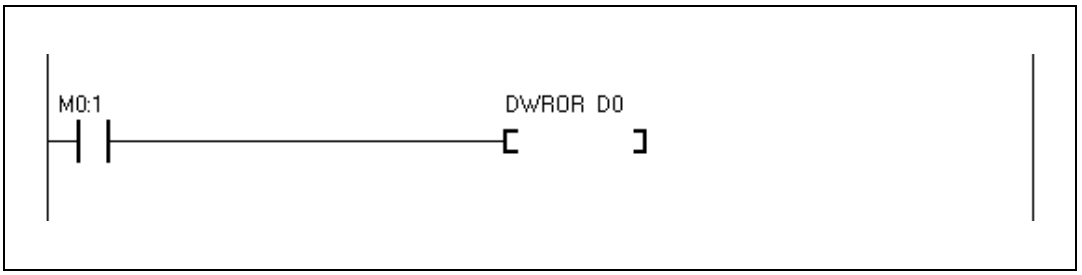
DWROR 32 비트 우회전

Usage : DWROR d d = d >> 1

개요

32 비트 우회전 명령입니다. d 를 한 비트씩 오른쪽으로 회전 시킵니다.

프로그램 예

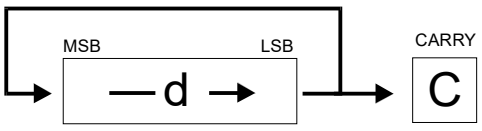


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0,D1 을 한 비트씩 오른쪽으로 회전시킵니다. 최하위 비트(LSB)에 있던 값은 CARRY 비트로 이동합니다.



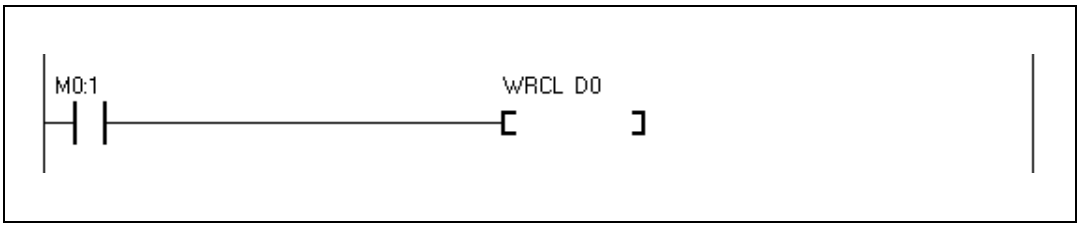
WRCL 16 비트 좌회전(Carry 포함)

Usage : WRCL d d = d << 1 with C

개요

16 비트 좌회전 명령입니다. d 를 CARRY 를 포함해서 한 비트씩 왼쪽으로 회전 시킵니다.

프로그램 예

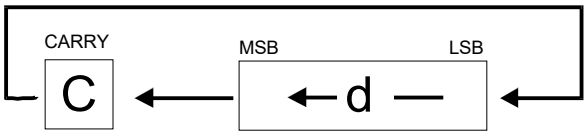


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0 을 캐리 플레그와 함께 한 비트씩 왼쪽으로 회전시킵니다. 최상위 비트(MSB)에 있던 값은 CARRY 비트로 이동합니다.



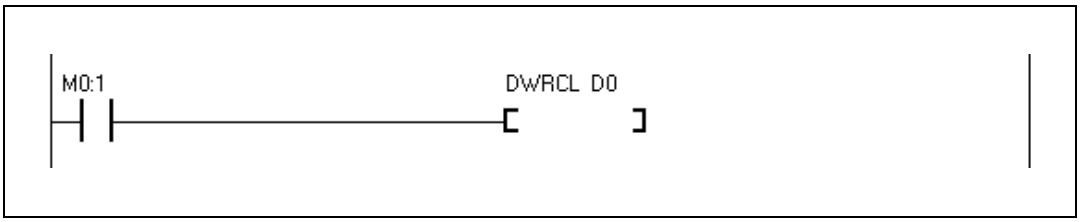
DWRCL 32 비트 좌회전(Carry 포함)

Usage : DWRCL d d = d << 1 with C

개요

32 비트 좌회전 명령입니다. d 를 CARRY 를 포함해서 한 비트씩 왼쪽으로 회전 시킵니다.

프로그램 예

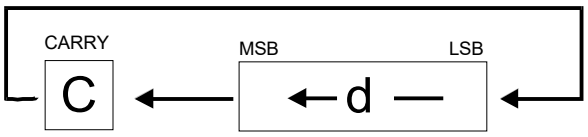


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0,D1 을 캐리 플레그와 함께 한 비트씩 왼쪽으로 회전시킵니다. 최상위 비트(MSB)에 있던 값은 CARRY 비트로 이동합니다.



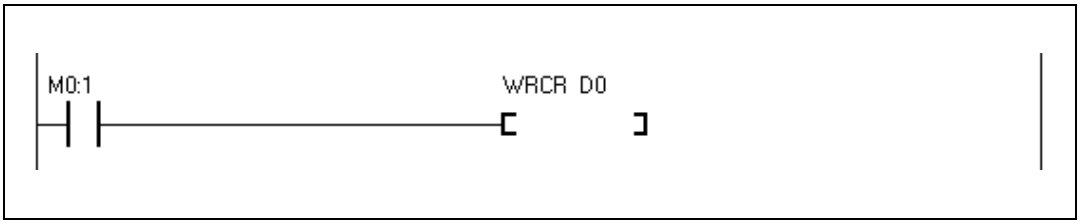
WRCR 16 비트 우회전(Carry 포함)

Usage : WRCR d d = d >> 1 with C

개요

16 비트 우회전 명령입니다. d 를 CARRY 를 포함해서 한 비트씩 오른쪽으로 회전 시킵니다.

프로그램 예

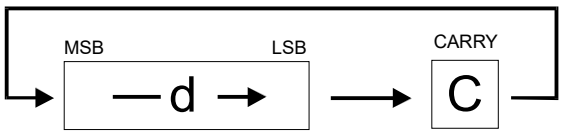


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0 을 CARRY 를 포함해서 한 비트씩 오른쪽으로 회전시킵니다. 최하위 비트(LSB)에 있던 값은 CARRY 비트로 이동합니다.



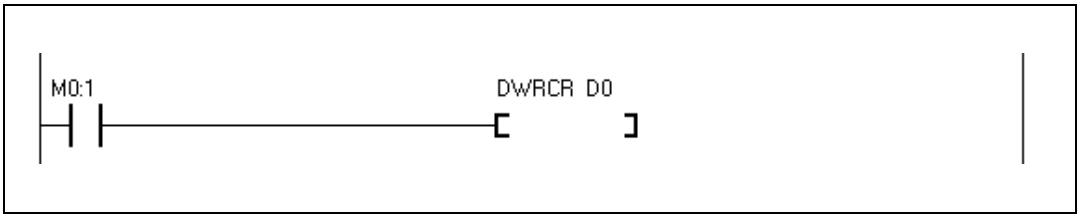
DWRCR 32 비트 우회전(Carry 포함)

Usage : DWRCR d d = d >> 1 with C

개요

32 비트 우회전 명령입니다. d 를 CARRY 와 함께 한 비트씩 오른쪽으로 회전 시킵니다.

프로그램 예

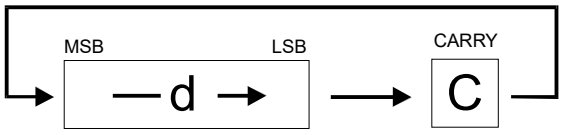


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0,D1 을 CARRY 와 함께 한 비트씩 오른쪽으로 회전시킵니다. 최하위 비트(LSB)에 있던 값은 CARRY 비트로 이동합니다.



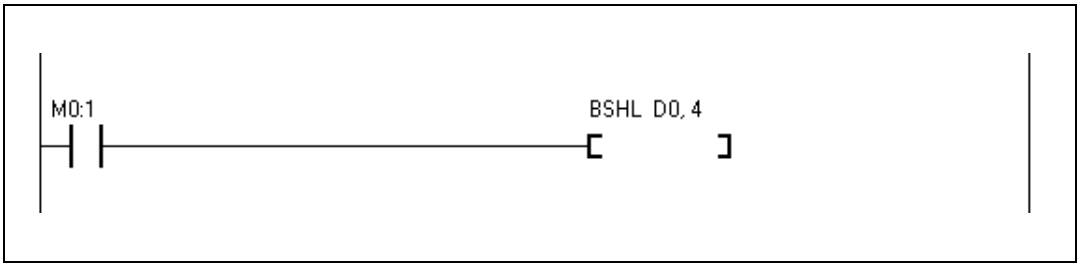
BSHL 비트단위 좌 쉬프트

Usage : BSHL d,n d = d << n

개요

비트 단위 Left 쉬프트 명령입니다. d 를 n 비트 만큼 왼쪽으로 쉬프트 합니다. 새로 삽입되는 비트는 전부 0 으로 합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	
n												O	

자세한 설명

M0:1 이 ON 되면 D0 를 4 비트 왼쪽으로 회전시킵니다. 최상위에 있던 비트들은 밀려나가고 (소실됩니다.) 최하위비트에 밀려들어오는 비트들은 전부 0 이 됩니다.

	HIGH byte	LOW byte
D0	00001111	1110 0011
D1		

↓

D2	11111110	00110000
D3		

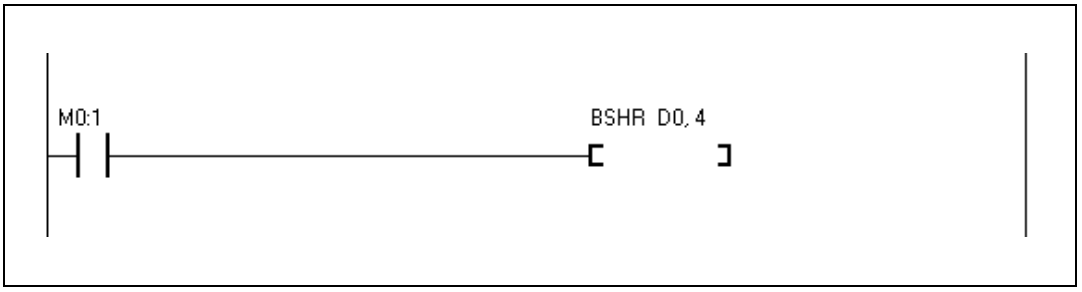
BSHR 비트단위 우 쉬프트

Usage : BSHR d,n d = d >> n

개요

비트 단위 Right 쉬프트 명령입니다. d 를 n 비트 만큼 오른쪽으로 쉬프트 합니다. 새로 삽입되는 비트는 전부 0 으로 합니다.

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0 를 4 비트 오른쪽으로 회전시킵니다. 최하위에 있던 비트들은 밀려나가고 (소실됩니다.) 최상위 비트에 밀려들어오는 비트들은 전부 0 이 됩니다.

	HIGH byte	LOW byte
D0	00001111	1110 0011
D1		

↓

D2	00000000	11111110
D3		

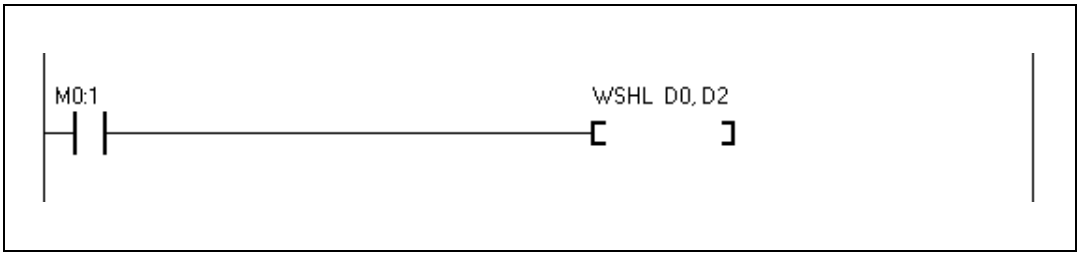
WSHL 워드단위 좌 쉬프트

Usage : BSHL s1,s2

개요

워드 단위 Left 쉬프트 명령입니다. s1 에서 s2 까지를 1 워드씩 왼쪽으로 쉬프트합니다. 새로 삽입되는 워드는 전부 0 이 됩니다.

프로그램 예

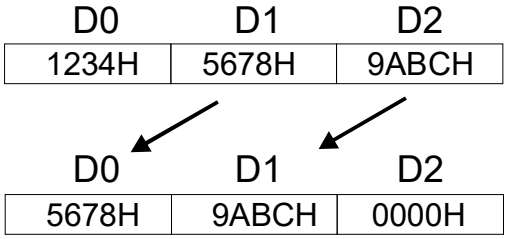


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1,s2	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0 부터 D2 까지 왼쪽으로 1 워드씩 회전시킵니다. D0 은 소실되고 D2 에는 0 이 저장됩니다.



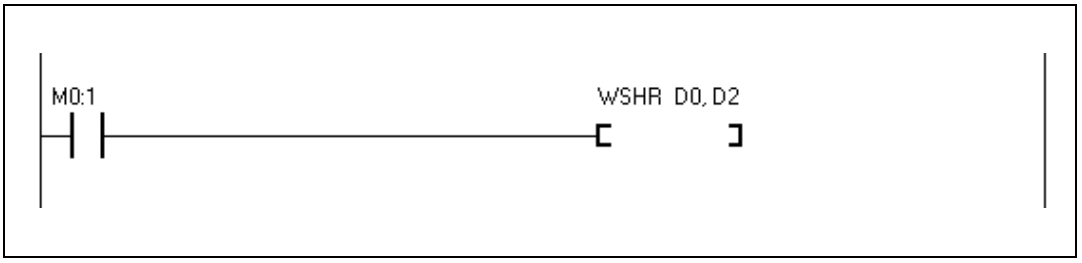
WSHR 워드단위 우 쉬프트

Usage : BSHR s1,s2

개요

워드 단위 Right 쉬프트 명령입니다. s1 에서 s2 까지를 1 워드씩 오른쪽으로 쉬프트합니다. 새로 삽입되는 워드는 전부 0 이 됩니다.

프로그램 예

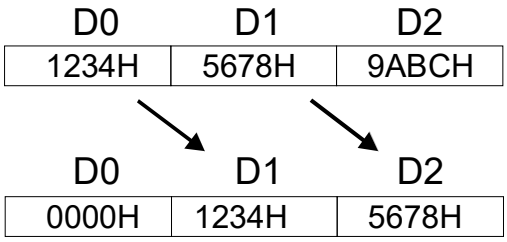


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s1,s2	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D0 부터 D2 까지 오른쪽으로 1 워드씩 회전시킵니다. D2 은 소실되고 D0 에는 0 이 저장됩니다.



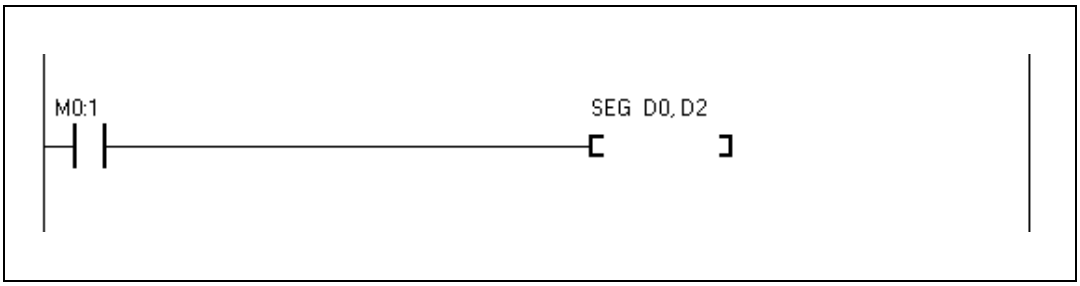
SEG 7 세그먼트용 데이터 변환

Usage : SEG s,d

개요

s 에 있는 16 비트 데이터를 4 digit 의 7 세그먼트용 데이터로 변환합니다.

프로그램 예

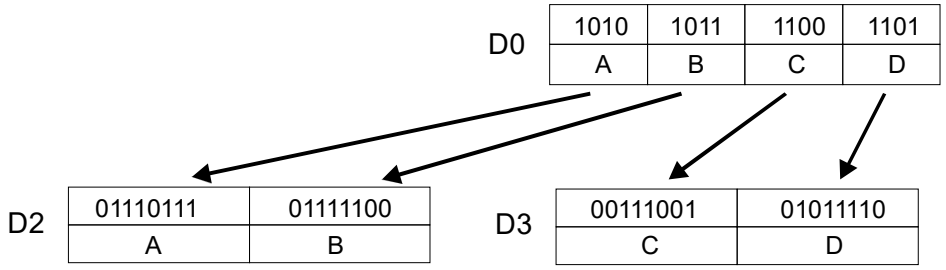


사용가능 영역

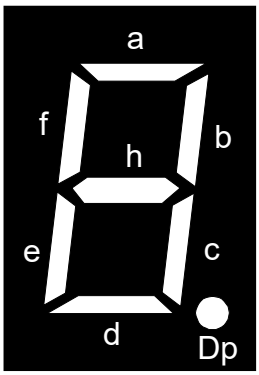
오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O		O		O	O	O	O	O	O	O	O
d	O	O		O		O	O	O			O	O	

자세한 설명

세븐 세그먼트를 위한 데이터 변환 명령입니다. M0:1 이 ON 되면 D0 에 있는 16 비트 데이터를 4 digit 의 세븐 세그먼트용 데이터로 변환하여 2 워드(4 바이트)로 저장합니다.



다음은 세븐세그먼트 데이터 변환 표입니다.



16 진	Dp	g	f	e	d	c	b	a	7 세그먼트 표시
0	0	0	1	1	1	1	1	1	0
1	0	0	0	0	0	1	1	0	1
2	0	1	0	1	1	0	1	1	2
3	0	1	0	0	1	1	1	1	3
4	0	1	1	0	0	1	1	0	4
5	0	1	1	0	1	1	0	1	5
6	0	1	1	1	1	1	0	1	6
7	0	0	1	0	0	1	1	1	7
8	0	1	1	1	1	1	1	1	8
9	0	1	1	0	1	1	1	1	9
A	0	1	1	1	0	1	1	1	A
B	0	1	1	1	1	1	0	0	B
C	0	0	1	1	1	0	0	1	C
D	0	1	0	1	1	1	1	0	D
E	0	1	1	1	1	0	0	1	E
F	0	1	1	1	0	0	0	1	F

TPC3X 에서 세븐 세그먼트를 구동하는 방법은 크게 두 가지가 있습니다. I/O 포트에 직접 세븐세그먼트 를 연결해서 SEG 명령을 써서 구동하는 방법과, 별도의 SGN (컴파일에서 시판하고 있는 시리얼 세븐 세그먼트 모듈)을 사용하는 방법입니다.
1 개 또는 2 개정도의 세븐 세그먼트만을 구동하는 경우에는 SEG 명령을 써서, 직접 연결하는 것이 경제적이지만, 포트가 많이 필요하다는 단점이 있습니다. SGN 모듈을 사용하면 단 1 개의 I/O 포트만을 사용하므로 포트가 절약되고, 한개의 선으로 최대 8 개의 SGN 모듈 (40 개의 7 세그먼트)을 연결할 수 있으므로, 많은 수의 세븐 세그먼트를 제어할 수 있습니다. (SGN 에 대한 자세한 설명은 SGNOUT 명령 설명을 참고하시기 바랍니다.)

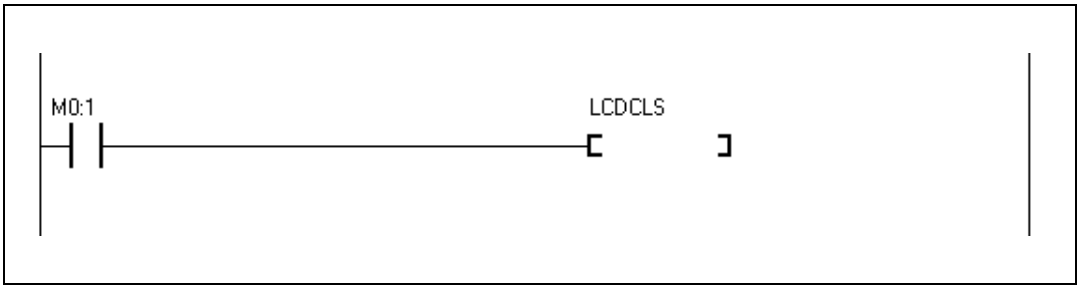
LCDCLS LCD 화면 클리어

Usage : LCDCLS

개요

LCD 디스플레이 영역 CH를 모두 20H (ASCII 코드의 SPACE)로 채웁니다.

프로그램 예



자세한 설명

LCD 디스플레이 영역 CH를 모두 20H (ASCII 코드의 SPACE)로 채웁니다.
LCDOUT 명령이 없으면 LCD에 효과가 나타나지는 않습니다. 단지 CH 영역만을 클리어 하는 명령입니다.

LCDOUT

LCD 송신 개시

Usage : LCDOUT port,mode

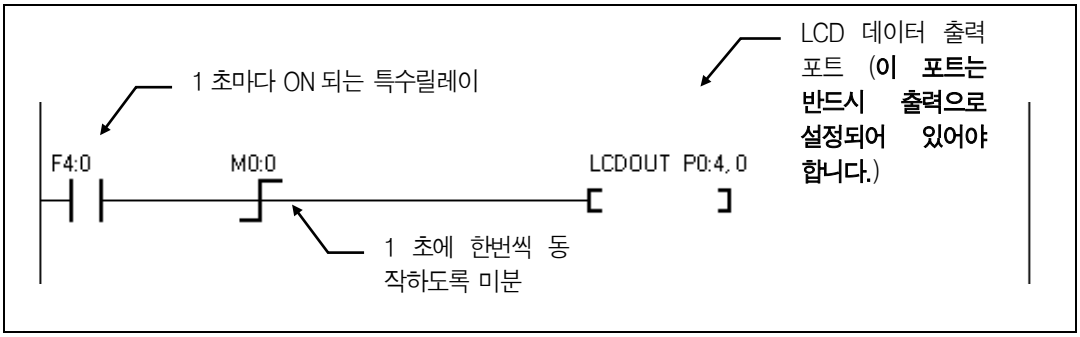
개요

CH 영역에 있는 데이터를 시리얼 LCD 모듈로 송신합니다.

- port 는 LCD 모듈이 연결되어 있는 포트번호 기술 (예: P0:4)
- mode 는 연결된 LCD 의 종류입니다.
  - 0 : 16 by 2 영문 LCD 모듈 (ELCD162)
  - 1 : 16 by 4 영문 LCD 모듈 (ELCD164)
  - 2 : 20 by 4 영문 LCD 모듈 (ELCD204)
  - 3 : 한글 LCD (HLCD114)
  - 4 : 한글 LCD (HLCD112)

참고: HLCD154, HLCD168 등 큰 사이즈의 LCD 는 TPC3X 에서 사용 불가능

프로그램 예



자세한 설명

F4:0 은 1 초 주기로 ON/OFF 를 반복하는 특수 릴레이 입니다. 이것을 미분해서 1 초마다 한번씩 LCDOUT 명령이 동작되도록 합니다. LCDOUT 명령은 CH 영역에 있는 데이터를 LCD 로 송신하는 명령입니다. 한번 LCDOUT 명령이 실행되면 CH 영역에 있는 모든 데이터가 LCD 쪽으로 송신 됩니다. (약 200mS~500mS 소요)

송신되는 중간에 또 송신하는 일이 없도록 각별히 신경써 주어야 합니다. 그래서 위의 예처럼 1 초에 한번씩 송신되는 방법을 사용하거나, 사용자의 키입력이 있었을 때 표시하도록 프로그램 해야합니다.

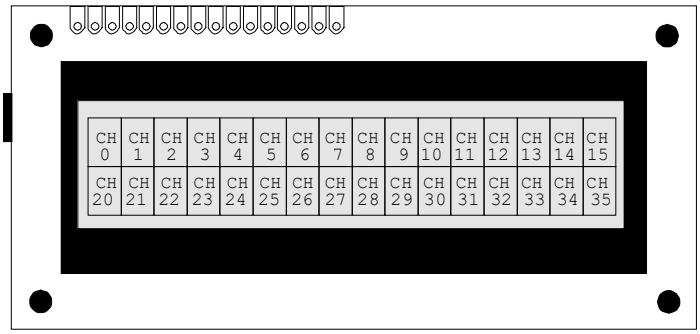
LCD 모듈은 컴파일사의 시리얼 LCD 제품만 사용 가능합니다. 시리얼 LCD 모듈은 반드시 19200 모드로 셋팅해야 합니다. (점퍼 등으로 셋팅 하도록 되어 있습니다.)

시리얼 영문 LCD 모듈에 대하여...

컴파일에서 시판하고 있는 시리얼 LCD 모듈은 기존의 LCD 모듈과는 달리 3 선으로 연결하는 방식의 LCD 모듈입니다. (기존의 방식은 14 선으로 연결하는 병렬접속 방식입니다.)

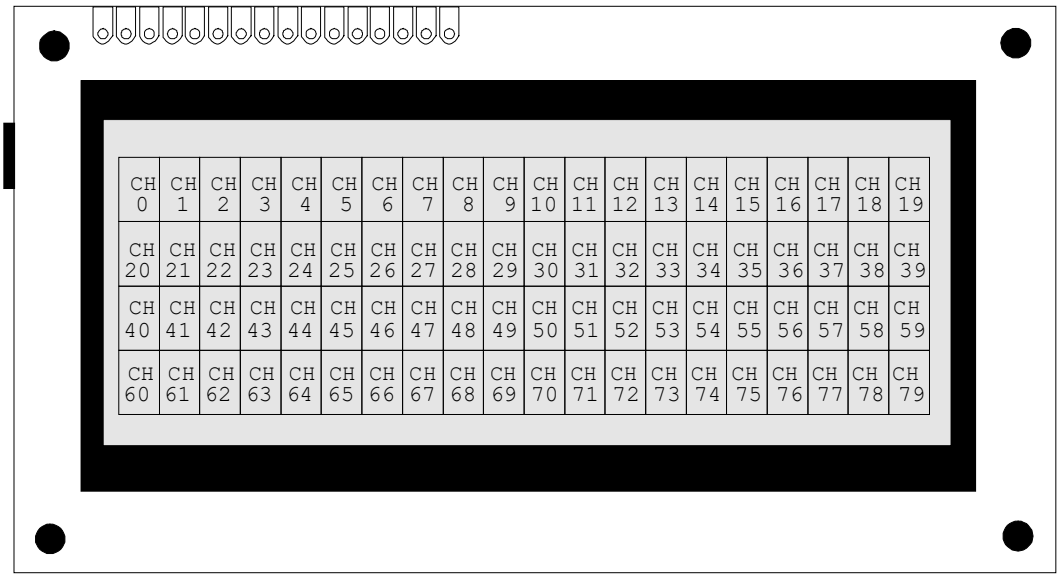
3 선중 2 선은 전원선 (+5V, GND)이고 나머지 한선으로 데이터를 수신합니다. 데이터는 5V 레벨의 RS232C, 19200 보레이트를 사용하며, ASCII 코드를 송신하면 원하는 위치에 글자를 표시할 수 있습니다. TPC3X 의 LCDOUT 명령은 CH 영역에 있는 모든 데이터 (ASCII 코드)를 RS232 으로 송신하는 명령이라고 할 수 있습니다.

LCD 상의 표시위치는 CH 번지와 밀접한 관계가 있습니다. CH0 에 기록하면 왼쪽, 맨 윗줄에 표시됩니다. 다음은 16 by 2 LCD 와 CH 영역과의 관계를 나타낸 것입니다.



첫번째 행은 CH0 부터 할당되어 있고, 두 번째 행은 CH20 부터 할당되어 있습니다. 16by2 사용시 CH40~79 와 같은 영역에 데이터를 저장하면 LCD 화면에 표시되지 않습니다.

다음은 20 by 4 라인의 경우입니다. (20 by 4 LCD 는 CH 영역 80 바이트 전부를 다 표시할 수 있습니다.)



다음은 TPC3X 에 연결 가능한 시리얼 LCD 모듈입니다.

모델명	제품명	화면 사이즈 (mm)
ELCD162	16 by 2 영문 LCD	64.5 * 13.8
ELCD162-BL	16 by 2 영문 LCD 백라이트	64.5 * 13.8
ELCD164	16 by 4 영문 LCD	61.8 * 25.2
ELCD164-BL	16 by 4 영문 LCD 백라이트	61.8 * 25.2
ELCD204	20 by 4 영문 LCD	76 * 25.2
ELCD204-BL	20 by 4 영문 LCD 백라이트	76 * 25.2
ELCD162-BIG-BL	더블 사이즈 16 by 2 영문 LCD 백라이트	99 * 36
HLCD112-BL	한글 LCD 11 by 2 라인 백라이트	60.5 * 18.5
HLCD114-BL	한글 LCD 11 by 4 라인 백라이트	70.7 * 38.8

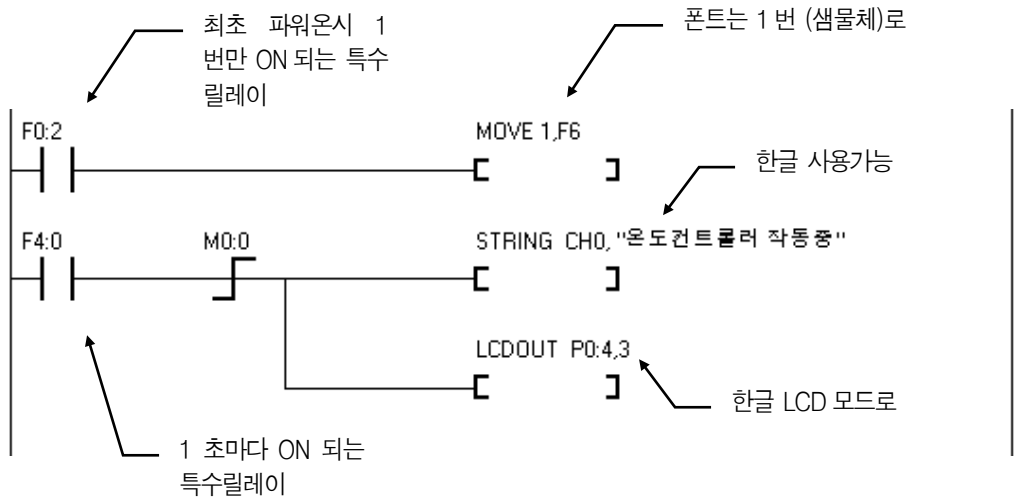
참고 사항

출력포트를 바꾸면 여러개의 LCD 모듈을 구동하는 것도 가능합니다. 여러개의 LCD 모듈을 구동하는 경우에는 출력 시간 (200~500mS)가 서로 충돌하는 일이 없도록 유의해야 합니다. (**SGNOUT 명령과도 충돌하면 안됩니다.**)

한글 LCD 모듈에 대하여...

컴파일에서 시판하고 있는 한글 LCD 모듈은 단 3 가닥의 선으로 한글을 LCD 상에 표현하는 디스플레이 장치입니다. (실제로는 그래픽 LCD 에 한글폰트를 표시하는 원리로 구동됩니다.) 이를 위해서 LCD 뒷면에 부착된 보드에는 한글 폰트롬등이 실장되어 있습니다.

한글 LCD 도 LCDOUT 명령으로 구동할 수 있습니다. 모드는 3 번을 사용하며, 다음은 한글 LCD 구동시 레더 프로그램 예입니다.



한글 폰트는 특수릴레이 F6 의 값으로 결정됩니다. 파워온시 F6 에 폰트값을 넣으면 됩니다.

- 0 : 작은 샘물체 (영문은 최대 21 자, 한글은 11 자까지 표시가능)
- 1 : 작은 고딕체 (영문은 최대 21 자, 한글은 11 자까지 표시가능)
- 2: 큰 명조체 (영문은 최대 16 자, 한글은 8 자까지 표시가능)
- 3: 큰 태고딕체 (영문은 최대 16 자, 한글은 8 자까지 표시가능)

SGNOUT

SGN 송신 개시

Usage : SGNOUT port, n

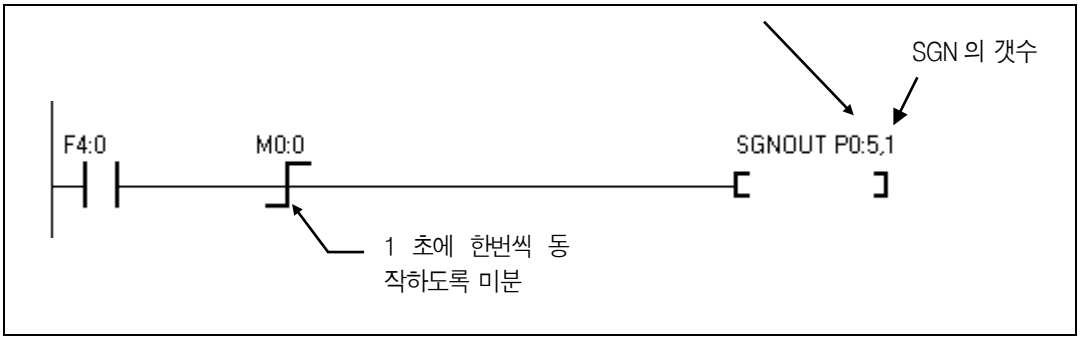
개요

G 영역에 있는 데이터를 시리얼 7 세그먼트 모듈(SGN)로 송신합니다. port 는 LCD 모듈이 연결되어 있는 포트번호 기술 (예: P0:5), n 은 연결된 SGN 모듈의 갯 수를 기술합니다.

* SGN 은 컴파일 테크놀로지의 제품명입니다. (Seven seGment Network)

프로그램 예

SGN 데이터 출력 포트 (**이 포트는 반드시 출력으로 설정되어 있어야 합니다.**)



자세한 설명

F4:0 은 1 초 주기로 ON/OFF 를 반복하는 특수 릴레이 입니다. 이것을 미분해서 1 초마다 한번씩 SGNOUT 명령이 동작하도록 합니다. SGNOUT 명령은 G 영역에 있는 데이터를 SGN 으로 송신하는 명령입니다. 한번 SGNOUT 명령이 실행되면 G 영역에 있는 데이터가 SGN 쪽으로 송신 됩니다. (연결된 SGN 의 개수에 따라 전송되는 데이터의 양도 결정됩니다. 약 50mS 소요)

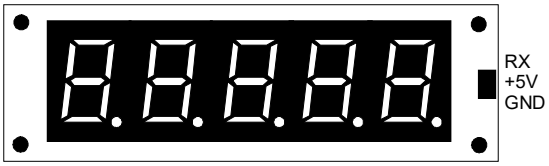
송신되는 중간에 또 송신하는 일이 없도록 각별히 신경써 주어야 합니다. 그래서 위의 예처럼 1 초에 한번씩 송신되는 방법을 사용하거나, 사용자의 키입력이 있었을때 표시하도록 프로그램 해야합니다.

SGN 모듈 하나에는 최대 5 개의 7 세그먼트가 부착되어 있지만, 최대 8 개의 SGN 모듈끼리 서로 부착할 수 있으므로, 하나의 I/O 포트로 제어할 수 있는 최대 7 세그먼트는 40 개입니다.

SGN 모듈에 대하여...

SGN 모듈이란 컴파일 테크놀로지에서 시판하고 있는 7 세그먼트 디스플레이 모듈 이름입니다. (Seven seGment Network 의 약자) 일반적으로 7 세그먼트를 구동시키기 위해 다이내믹 디스플레이라는 방법을 많이 사용하지만, 많은 선을 필요로 하고, 타이밍을 잘 맞추어 구동 해야하는 등의 불편한 점이 있었습니다. SGN 모듈은 기판 위에 7 세그먼트를 최대 5 개까지 다이내믹 디스플레이 방식으로 구동 시켜주는 전용칩과 함께 실장되어 있으며, 5V 레벨의 RS232C, 9600 보레이트로 데이터를 보내주기만 하면, 원하는 위치에 표시해 주는 편리한 제품입니다.

TPC3X 에는 이러한 SGN 모듈을 구동 시킬 수 있는 SGNOUT 이라는 명령이 준비되어 있으며, TPC3X 의 I/O 포트 중 하나를 SGN 과 연결하는 것만으로 간단하게 설치할 수 있습니다.



SGN 모듈 끼리 부착할 수 있는 별도의 포트가 준비되어 있으며, 하나의 I/O 포트로 최대 8 개의 SGN 모듈을 제어할 수 있습니다.

SGN 모듈 뒷면에 각각의 어드레스를 셋팅할 수 있는 딥 스위치가 있습니다. 딥스위치 셋팅에 따라 G 영역이 할당됩니다.

DIP 스위치	G 영역 할당
0 0 0 0	G0 ~ G4
0 0 0 1	G5 ~ G9
0 0 1 0	G10 ~ G14
0 0 1 1	G15 ~ G19
0 1 0 0	G20 ~ G24
0 1 0 1	G25 ~ G29
0 1 1 0	G30 ~ G34
0 1 1 1	G35 ~ G39

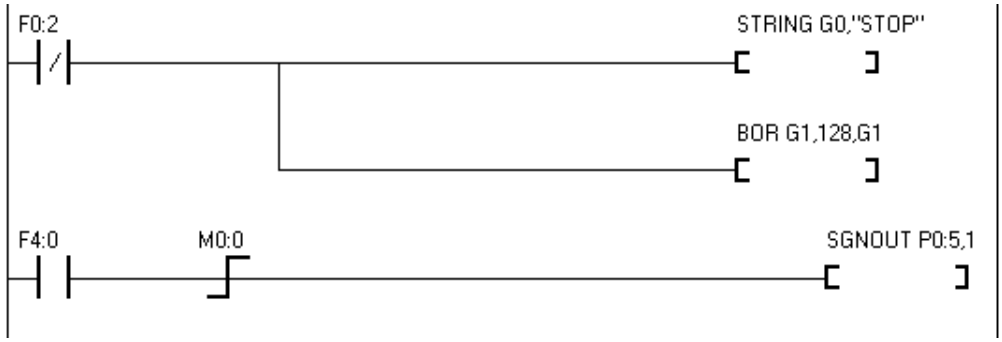
SGN 모듈 하나를 사용할 경우 G 영역 0 번지는 왼쪽 1 번째 7 세그먼트에 할당되어 있습니다.

따라서 G0 에 31H (ASCII 코드로 1) , G1 에 32H, G2 에 33H, G3 에 34H, G4 에 35H 가 있고, SGNOUT 명령이 실행되었다면 다음과 같이 출력됩니다.

SGN 에 데이터를 출력하기 위해서는 ASC, HEX 명령과 STRING 명령을 사용해서 G 영역에 원하는 데이터를 (ASCII 코드로)저장한 뒤 SGNOUT 으로 송신하면 됩니다.

SGN 의 특정위치에 DOT 표시

Dot 를 ON 하고 싶은 경우에는 ASCII 코드의 최상위 비트를 1 로 지정하면 됩니다. 이 경우에는 BOR 명령을 써서 특정 위치의 최상위 비트를 1 로 만듭니다.



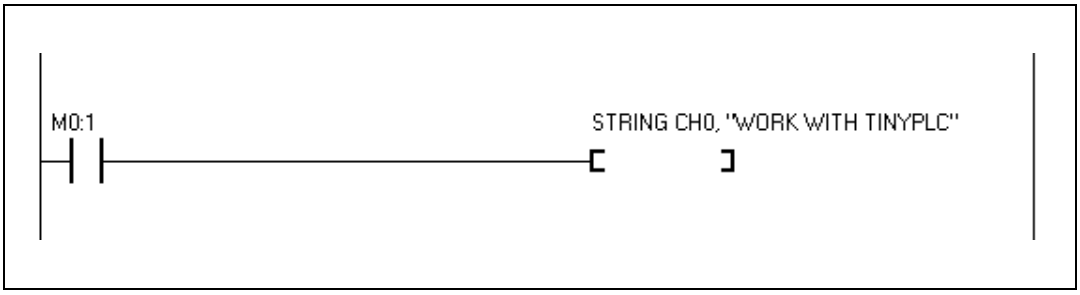
STRING 문자열 저장

Usage : STRING d,"string"

개요

문자열 "string"을 d에 문자열의 갯 수만큼 저장합니다. (한글 LCD 사용 시 "string" 부분에 한글을 직접 사용할 수 있습니다. 예 : STRING CH0, "아름다운 우리나라")

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d	O	O		O		O	O	O			O	O	

자세한 설명

LCD 또는 SGN에 문자를 디스플레이 하기 위한 문자열 저장 명령입니다. M0:1 이 ON 되면 CH0 번지에 "WORK WITH TINYPLC"라는 문자열을 ASCII 코드로 저장합니다. 문자열이 한글일 경우에는 조합형 한글 코드로 저장합니다. (한글 LCD 에서 조합형을 사용하기 때문)

CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	CH10	CH1	CH12	CH13	CH14	CH15	CH16	CH17
W	O	R	K		W	I	T	H		T	I	N	Y	P	L	C	
57	4F	52	4B	20	57	49	54	48	20	54	49	4E	59	50	4C	43	

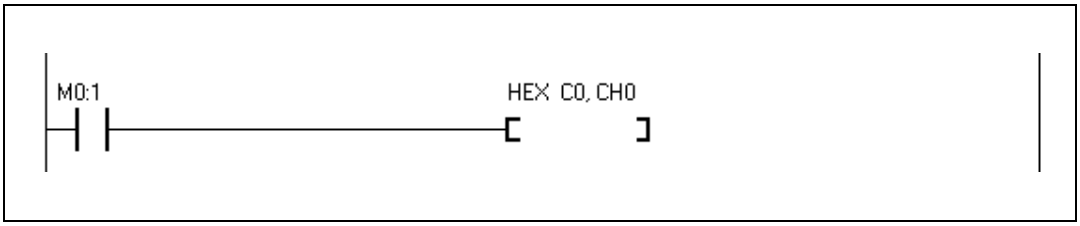
HEX 16 비트 -> 16 진 ASCII 변환

Usage : HEX s,d

개요

16 비트 2 진 값을 HEX 형 ASCII 코드로 변환합니다.

프로그램 예

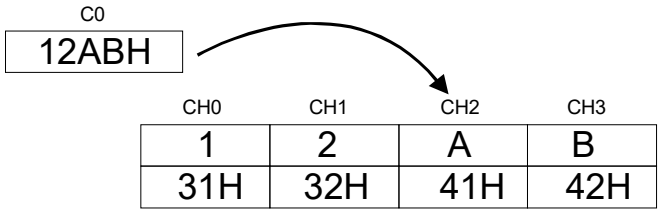


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O	O	O		O	O	O	O	O	O	O	O
d	O	O		O		O	O	O			O	O	

자세한 설명

LCD 또는 SGN 에 문자를 디스플레이 하기 위한 ASCII 코드 변환 명령입니다. M0:1 이 ON 되면 C0 값을 16 진 ASCII 코드로 변환해서 CH0 에서 부터 저장합니다.



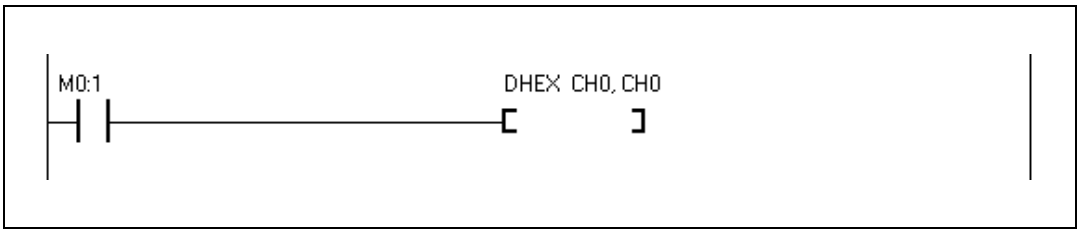
DHEX 32 비트 -> 16 진 ASCII 변환

Usage : DHEX s,d

개요

32 비트 2 진 값을 HEX 형 ASCII 코드로 변환합니다.

프로그램 예

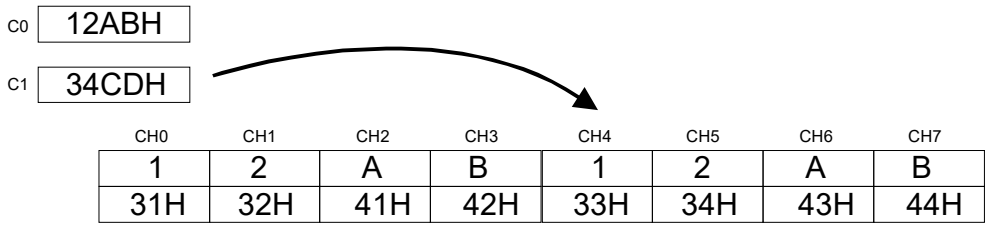


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O	O	O		O	O	O	O	O	O	O	O
d	O	O		O		O	O	O			O	O	

자세한 설명

LCD 또는 SGN 에 문자를 디스플레이 하기 위한 ASCII 코드 변환 명령입니다. M0:1 이 ON 되면 C0,C1 값 (더블워드) 을 16 진 ASCII 코드로 변환해서 CH0 에서 부터 저장합니다.



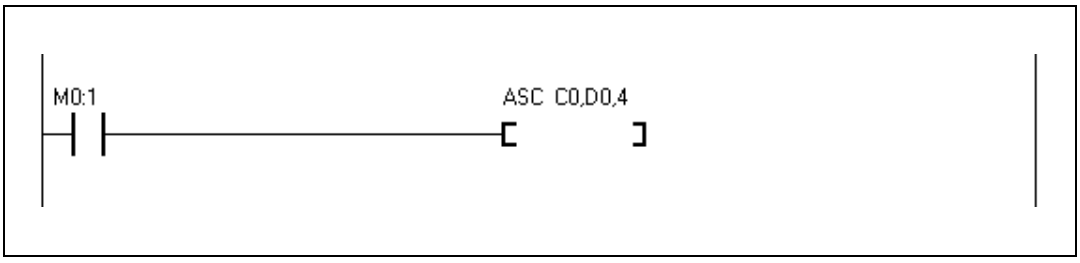
ASC 16 비트 -> 10 진 ASCII 변환

Usage : ASC s,d,n

개요

16 비트 2 진 값을 10 진수 ASCII 코드로 변환합니다.

프로그램 예

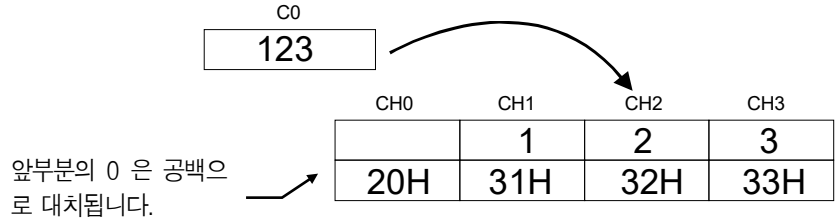


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O	O	O		O	O	O	O	O	O	O	O
d	O	O		O		O	O	O			O	O	
n (변환 갯수)													O

자세한 설명

LCD 또는 SGN 에 문자를 디스플레이 하기 위한 ASCII 코드 변환 명령입니다. M0:1 이 ON 되면 C0 값을 10 진 ASCII 코드로 변환해서 CH0 에서 부터 4 바이트로 저장합니다. (앞부분의 0 은 공백으로 처리됩니다.)



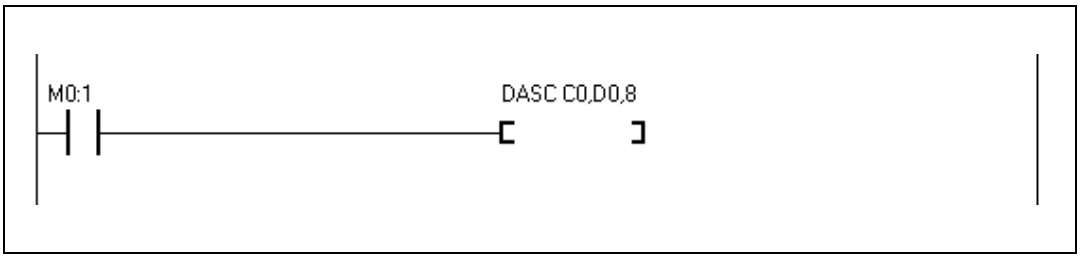
DASC 32 비트 -> 10 진 ASCII 변환

Usage : DASC s,d,n

개요

32 비트 2 진 값을 10 진 ASCII 코드로 변환합니다.

프로그램 예

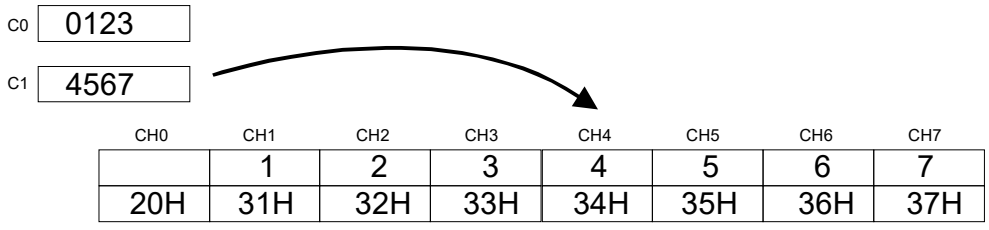


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O	O	O		O	O	O	O	O	O	O	O
d	O	O		O		O	O	O			O	O	
n (변환 개수)													O

자세한 설명

LCD 또는 SGN 에 문자를 디스플레이 하기 위한 ASCII 코드 변환 명령입니다. M0:1 이 ON 되면 C0,C1 값 (더블워드) 을 10 진 ASCII 코드로 변환해서 CH0 에서 부터 8 바이트로 저장합니다. (앞부분의 0 은 공백 처리 됩니다.)



GOTO, LABEL

점프와 라벨

Usage : GOTO label

Usage : LABEL label

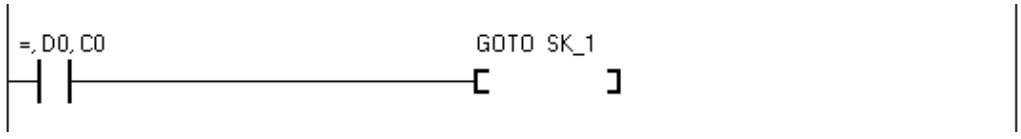
개요

GOTO 지정한 라벨로 점프하는 분기 명령입니다. LABEL 은 라벨을 선언하는 명령입니다.

프로그램 예

자세한 설명

M0:1 이 ON 되면 SK_1 으로 점프 합니다. 아래와 같이 사용하면 조건 분기명령으로 활용할 수도 있습니다. (D0 = C0 이면 SK_1 으로 점프합니다.)



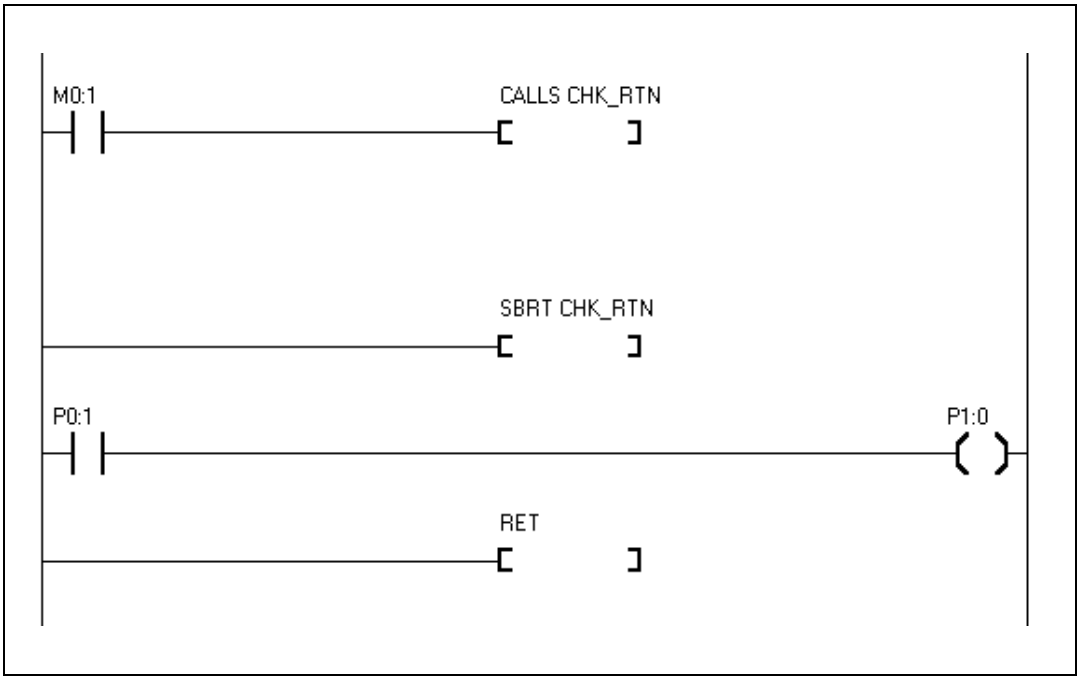
CALLS, SBRT 브루틴 콜, 서브루틴 시작
RET 서브루틴 복귀

Usage : CALLS label
Usage : SBRT label

개요

서브루틴과 관련된 명령어들입니다. CALLS 는 서브루틴을 콜하고, SBRT 는 서브루틴을 정의합니다. RET 는 서브루틴의 맨 끝에 위치해야 합니다.

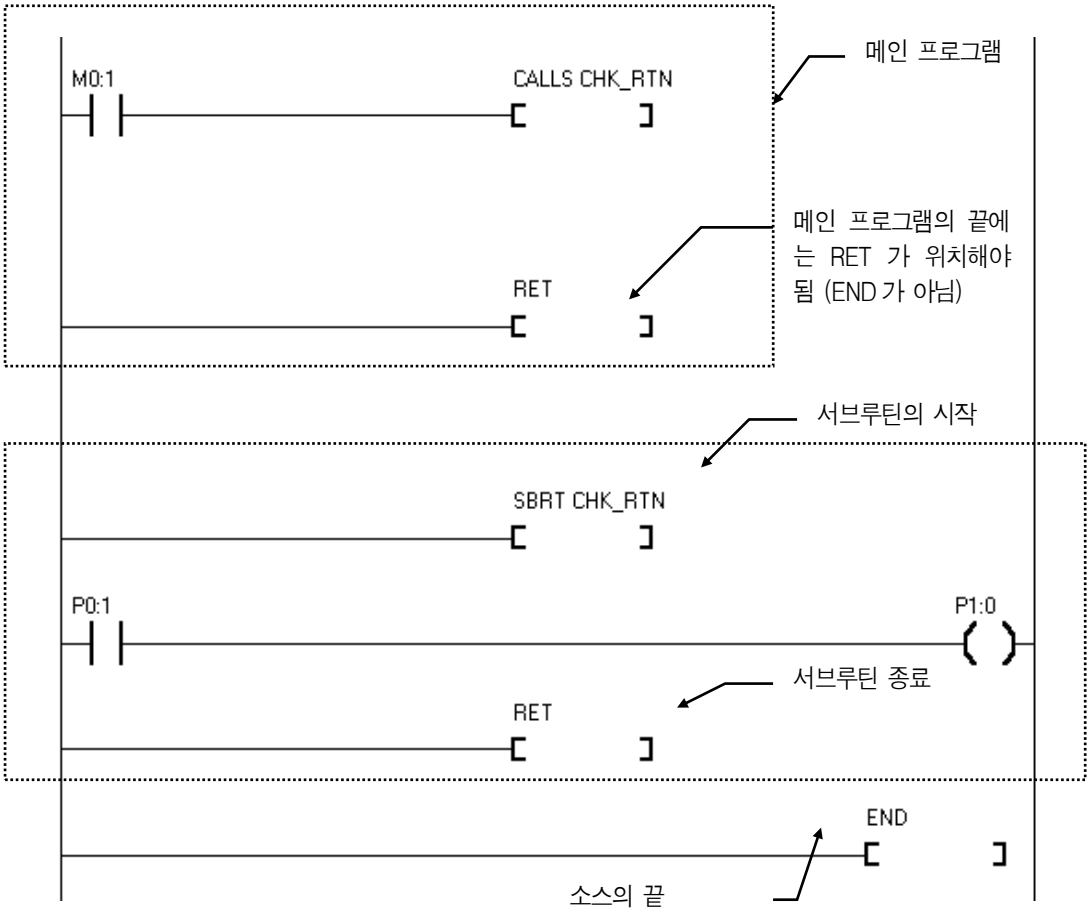
프로그램 예



자세한 설명

M0:1 이 ON 되면 CHK_RTN 을 콜 합니다. CALL 의 중첩실행은 최대 16 레벨까지 가능합니다. (중첩실행이란 서브루틴안에서 다른 서브루틴을 부르는 것을 의미합니다.)

메인루틴과 서브루틴을 작성할 때는 다음과 같은 순서대로 작성하기 바랍니다.



TPC3X 에서 END 명령은 단순히 소스프로그램의 끝을 의미한다는 것에 주의하기 바랍니다. MAIN 루틴의 끝에는 반드시 RET 명령을 써주어야 합니다.

KEYSCAN 키 스캔입력

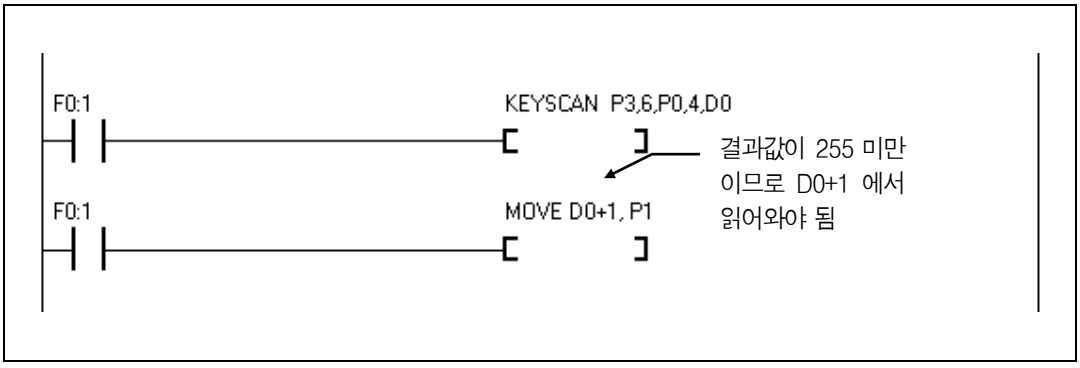
Usage : KEYSCAN a,b,c,d,e

개요

키 매트릭스를 스캐닝 방식으로 읽어오는 명령입니다.

- a : 입력측 포트의 바이트 번호 (예 : P3)
- b : 입력측 포트의 갯수
- c : 출력측 포트의 바이트 번호 (예 : P0)
- d : 출력측 포트의 갯수
- e : 키 스캔코드가 저장될 곳의 번지 (워드형식으로 저장)

프로그램 예



사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
d						O	O	O	O	O	O	O	

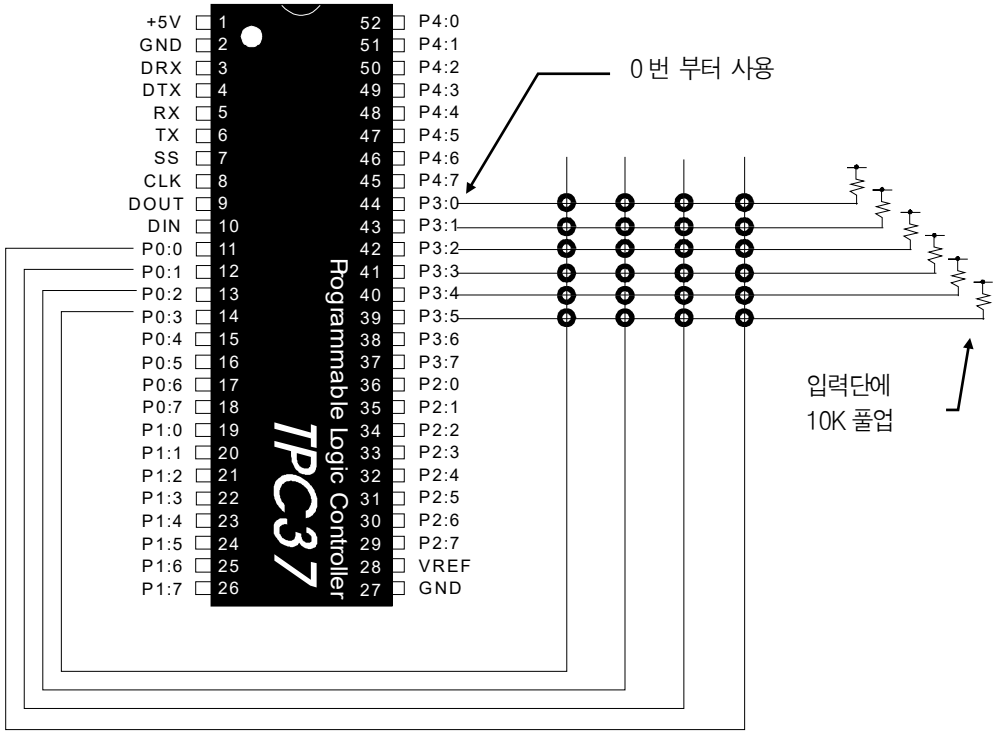
자세한 설명

F0:1 은 상시 ON 이므로 KEYSCAN 명령은 무조건 실행됩니다. P3 과 P0 포트에 연결된 6 * 4 키메트릭스에서 현재 눌려진 키를 D0 에 저장합니다. 밑에 있는 MOVE 명령은 D0 에 저장된 키스캔코드 값을 포트 1 에 그래로 표시하는 명령입니다. 포트 1 에 전부 LED 가 연결되어 있다면 눌려진 키 스캔코드를 쉽게 볼 수 있습니다.

이 명령은 매트릭스 방식의 맴브레인 키보드를 읽기 위해서 만들어진 명령입니다. (자동화 분야의 전용기에는 맴브레인 키보드를 많이 사용합니다.) 이 명령이 실행되기 위해서는 하드웨어적으로 몇 가지 지켜주어야 할 의무사항이 있습니다.

1. 포트의 0 번부터 배치해야 됨. 즉, P0 을 사용시 P0:0 부터 연결해야 합니다.
2. 입력포트에는 풀업저항 10K 오옴을 연결해야 됩니다.

KEYSCAN 명령은 최대 8* 8 키 메트릭스를 읽어올 수 있으며, 결과는 워드 크기로 저장되지만, 실제 값은 255 를 초과하지 않습니다. 키 입력이 없는 경우에는 결과가 0 이 됩니다. 만약 6 * 4 정도만 연결했다면, 해당 바이트의 남는 포트는 입력포트로만 사용 가능합니다. (출력 포트로 사용할 경우에는 엉뚱한 데이터가 출력될 것입니다.) 다음은 6*4 키메트릭스를 TPC3X 모듈에 부착한 그림입니다.



* 본 명령은 TPC33 의 경우 P1, P2 만 사용 가능합니다.
* 본 명령은 TPC37 의 경우 P0, P1, P2, P3 에서만 사용 가능합니다.
* 본 명령은 TPC38 의 경우 P0, P1, P2, P3 에서만 사용 가능합니다.

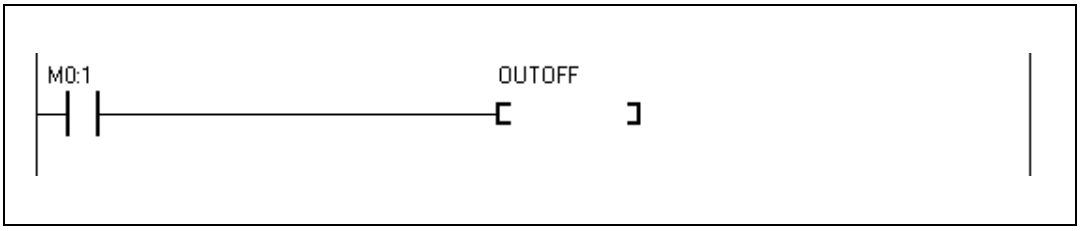
OUTOFF 전출력 OFF

Usage : OUTOFF

개요

출력으로 설정된 전 I/O 포트의 출력을 OFF 합니다.

프로그램 예



자세한 설명

M0:1 이 ON 되면 전 출력을 OFF 합니다.

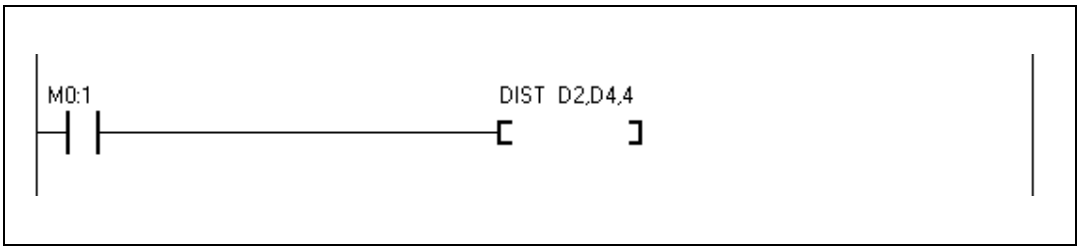
DIST 16 비트 분산

Usage : DIST s,d,n

개요

16 비트 값을 4 비트 단위로 나누어서 d 영역에 저장합니다.

프로그램 예

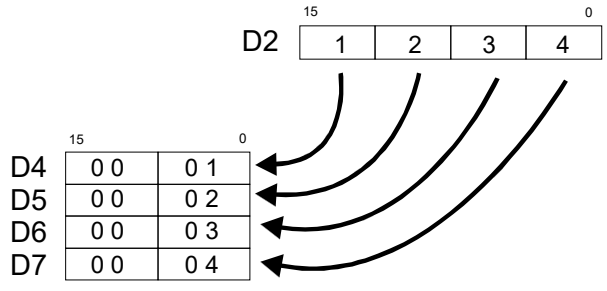


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O	O	O		O	O	O	O	O	O	O	O
d	O	O		O		O	O	O			O	O	
n						O	O	O					O

자세한 설명

M0:1 이 ON 되면 D2 에 있는 데이터를 4 비트 (니블 단위) 씩 나누어서 D4 부터 D7 까지, 나누어 저장합니다. (n 으로 나누어지는 갯 수를 지정할 수 있습니다. n 은 5 보다는 작은 값 이여야 합니다.)



UNIT

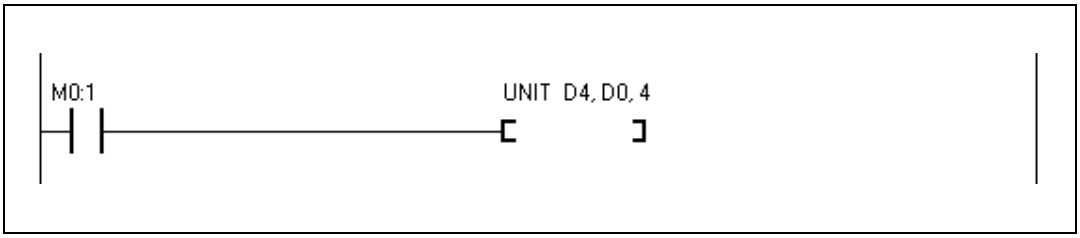
16 비트 결합

Usage : UNIT s,d,n

개요

하위 4 비트만 존재하는 데이터영역 (최대 4 워드)를 조합해서 하나의 워드로 만듭니다. (DIST 명령으로 분산시켰던 값을 조합합니다.)

프로그램 예

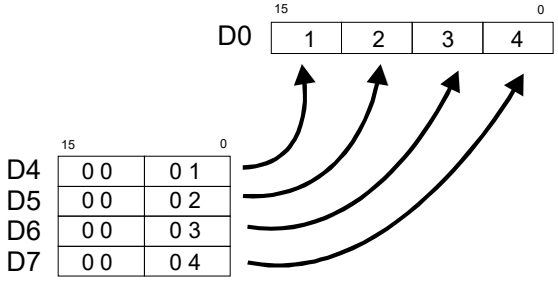


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O		O		O	O	O			O	O	
d	O	O		O		O	O	O			O	O	
n						O	O	O					O

자세한 설명

M0:1 이 ON 되면 D4 부터 D7 까지의 하위 니블 만을 조합해서 하나의 워드로 만들어 D0 에 저장합니다. (n 으로 조합하는 개 수를 지정할 수 있습니다. n 은 5 보다는 작은 값이여야 합니다.)



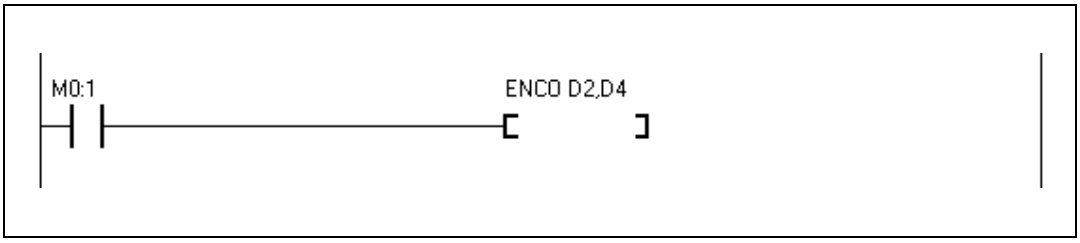
ENCO 16 비트 엔코더

Usage : ENCO s,d

개요

s 를 엔코드해서 d 에 저장합니다.

프로그램 예

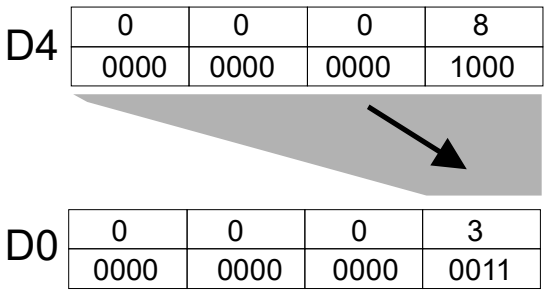


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O		O		O	O	O	O	O	O	O	O
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D4 의 값을 엔코드해서 D0 에 저장합니다. (엔코드된 데이터를 다른 비트열로 이동하고자 할 때에는 비트 쉬프트 명령 BSHL, BSHR 과 함께 사용 하십시요)



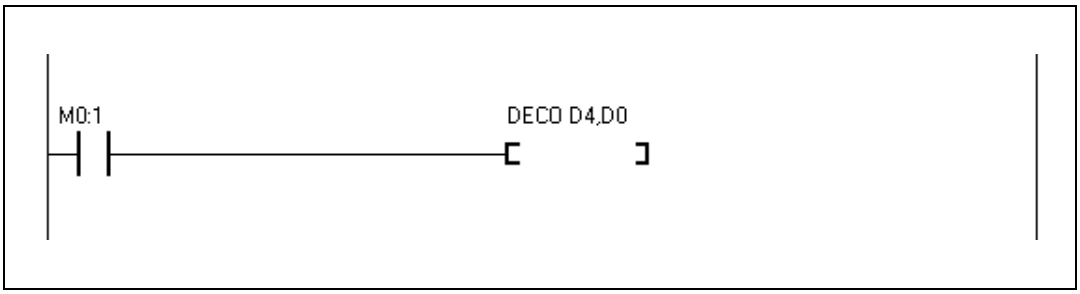
DECO 16 비트 디코더

Usage : DECO s,d

개요

s 의 하위 4 비트를 디코드해서 d 에 저장합니다.

프로그램 예

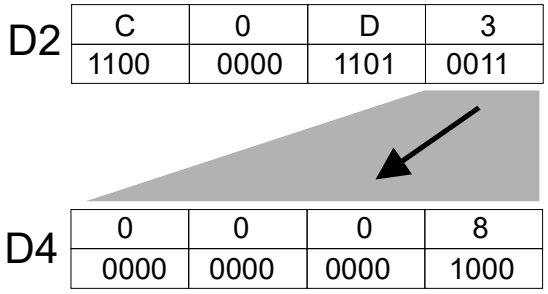


사용가능 영역

오퍼랜드	릴 레 이					카운터	타이머	데이터	기타				상수
	P	M	F	K	S	C	T	D	AD	CNT	CH	G	
s	O	O		O		O	O	O	O	O	O	O	O
d	O	O		O		O	O	O			O	O	

자세한 설명

M0:1 이 ON 되면 D2 의 하위 4 비트를 디코드해서 D4 에 저장합니다. (하위 4 비트가 아닌 다른 비트를 디코드 할 때에는 비트 쉬프트 명령 BSHR 과 함께 사용 하십시요)



HCNTCLR 고속카운터 클리어

개요

고속 카운터의 값을 0 으로 만듭니다.

프로그램 예

자세한 설명

고속 카운터 영역 (HCNT)를 0 으로 만드는 명령입니다.

제 5 장
TinyPLC Studio
사용설명

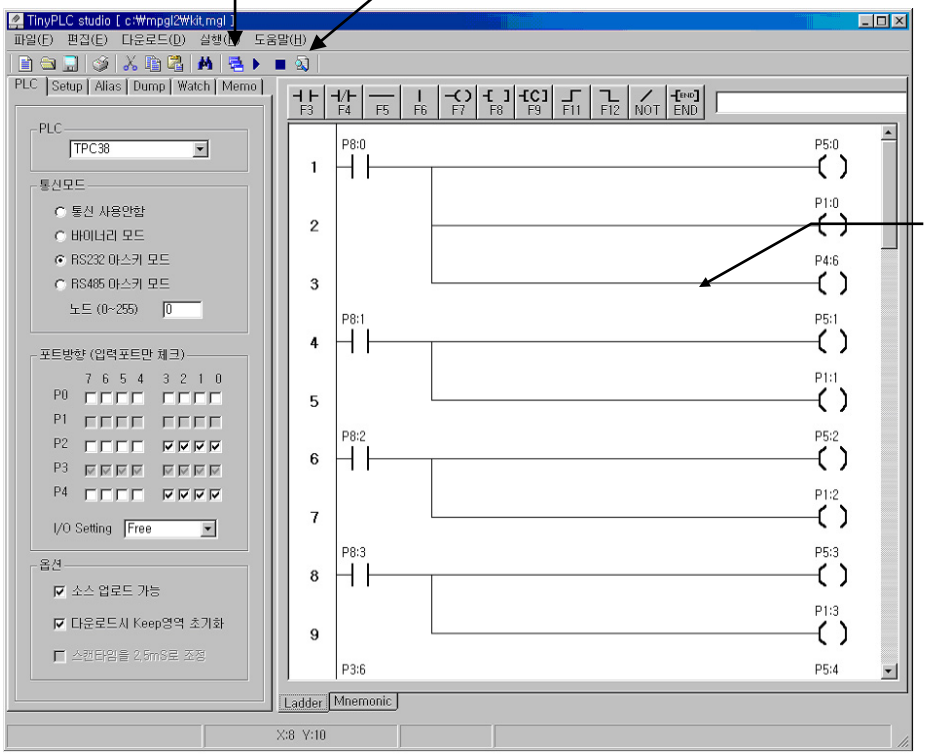
TinyPLC Studio 는 TPC3X 만을 위한 개발환경 소프트웨어입니다.
TPC9X 와 TPC-FXX 시리즈는 TPCWORKS 소프트웨어를 사용하십시오.

TinyPLC Studio 의 개요

TinyPLC Studio 는 PC 상에서 레더를 입력하고, 컴파일, 다운로드 및 모니터링을 해주는 TinyPLC 운용 소프트웨어입니다. PC 의 윈도우 XP 환경에서 실행되며 7 이후 버전은 지원되지 않습니다. 다음은 TinyPLC Studio 실행화면 입니다.

원터치로 컴파일에서 다운로드까지 실행합니다.

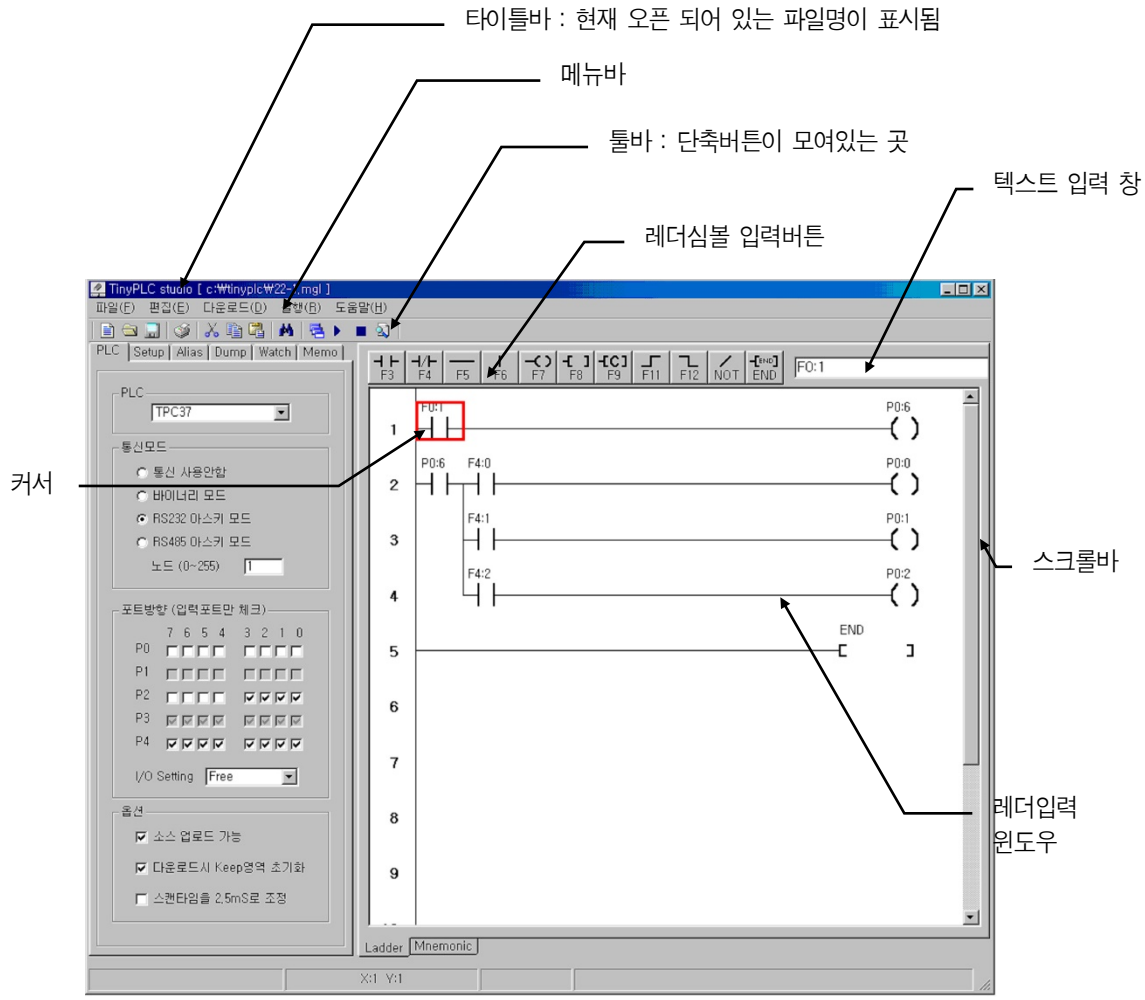
실시간 모니터링도 가능합니다.

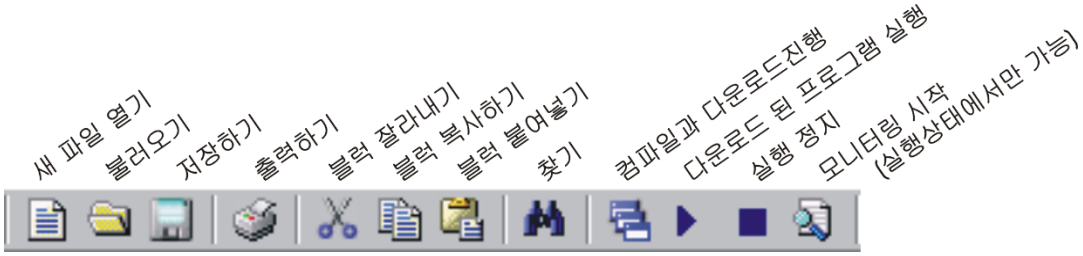


레더를 작성하고 수정하는 등의 작업을 할 수 있습니다.

TinyPLC Studio 는 TPC3X 만을 위한 통합개발 환경입니다.
TinyPLC Studio 는 무료로 공급하지만, 상업적인 목적으로 사용할 수는 없습니다.

TinyPLC Studio 의 화면구성





레더 심볼의 입력

입력을 원하는 지점에 커서를 위치시키고, 펑션키를 누르면 펑션키에 할당된 심볼이 표시됩니다. (아래 표 참조) 이 상태에서 텍스트 (영,숫자)를 입력하고 ENTER 키를 치면 심볼 윗부분에 입력한 텍스트가 표시됩니다.

펑션키	심볼	설명
F1		컴파일과 다운로드 진행.
F2		특정 문자열을 찾음 (릴레이 명 또는 명령어 등)
F3		A 접점 연산개시 (LOAD) 심볼 입력
F4		B 접점 연산개시 (LOADN) 심볼 입력
F5		AND 연결 및 다음 행 연결 (토글 방식으로 누를 때마다 표시 또는 삭제됨)
F6		OR 연결 및 삭제 (토글 방식으로 누를 때마다 표시 또는 삭제됨)
F7		결과 출력 (OUT)심볼 입력
F8		응용명령 심볼 입력 (TON, TOFF, WMOV, WMUL 등의 명령에 사용)
F9		카운터 심볼 입력 (CTU, CTD)
F11		상승에지 디퍼런셜 심볼 입력 (DF)
F12		하강에지 디퍼런셜 심볼 입력 (DFN)
/ (슬래쉬)		반전명령 (NOT) 심볼 입력

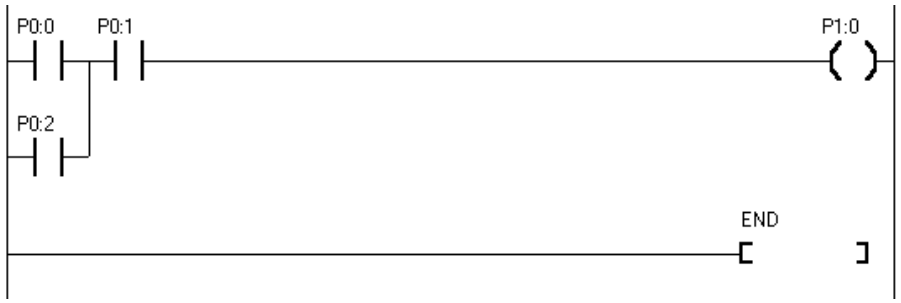
다른 키에 대한 사용 설명입니다.

키입력	동작
SPACE 키	현재 커서위치에 있는 레더심볼과 텍스트를 모두 삭제합니다.
Ctrl + I	빈 줄을 삽입합니다. (툴바에 있는 INS 단축버튼을 누른것과 동일합니다.)
Ctrl + D	현재 커서가 위치한 줄을 삭제합니다. (툴 바에 있는 DEL 단축버튼을 누른 것과 동일합니다.)
Ctrl + U	지운 줄을 복귀합니다. (툴 바에 있는 RES 단축버튼을 누른 것과 동일합니다.)
ENTER	현재 커서가 위치한 지점의 텍스트를 수정합니다. ENTER 키를 치면 커서위치의 텍스트가 "텍스트 입력창"에 곧바로 표시됩니다. 이때 End 키를 한번 더 누르면, 커서가 맨뒤로 이동하고, HOME 키를 누르면 맨 앞으로 이동합니다. 좌우 이동키를 누르면서 원하는 지점부터 수정할 수 있습니다. **수정이 끝난 후 반드시 ENTER 키를 눌러야 합니다.**

레더 작성 예

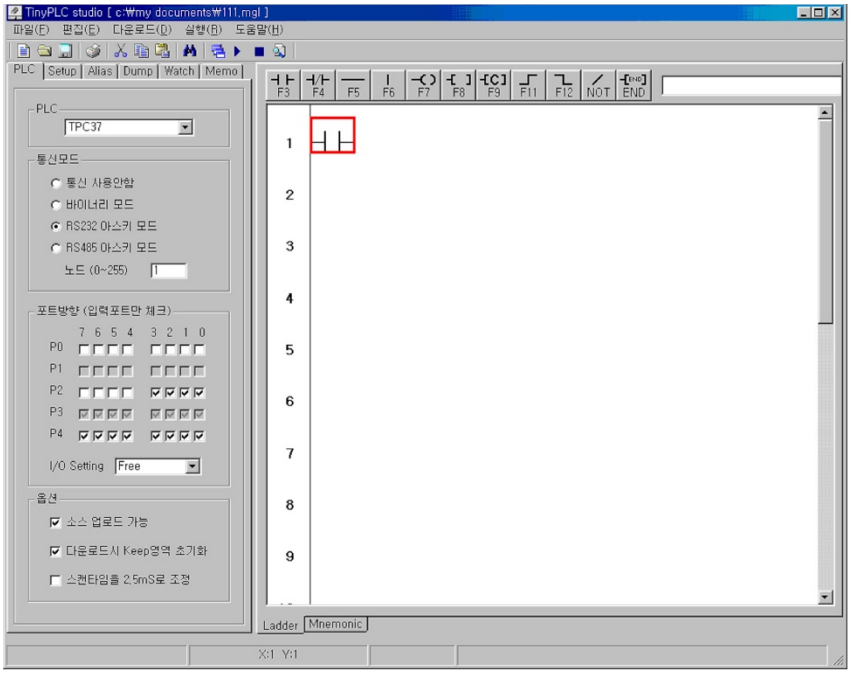
레더 프로그램 작성방법을 한 스텝씩 설명한 것입니다. 이곳에서 지시하는 대로 따라 해 보시기 바랍니다.

*참고사항
릴레이 번호 입력시 P00 이라고 입력하면 자동으로 P0:0 이 됩니다. P153 이라고 입력하면 P15:3 이 됩니다. 즉, 가장 뒷자리를 비트번호로 자동인식합니다.
이 기능은 심볼(F3, F4, F7, F11, F12)일 경우에는 해당되나, (F8)일 경우에는 자동변환되지 않습니다.



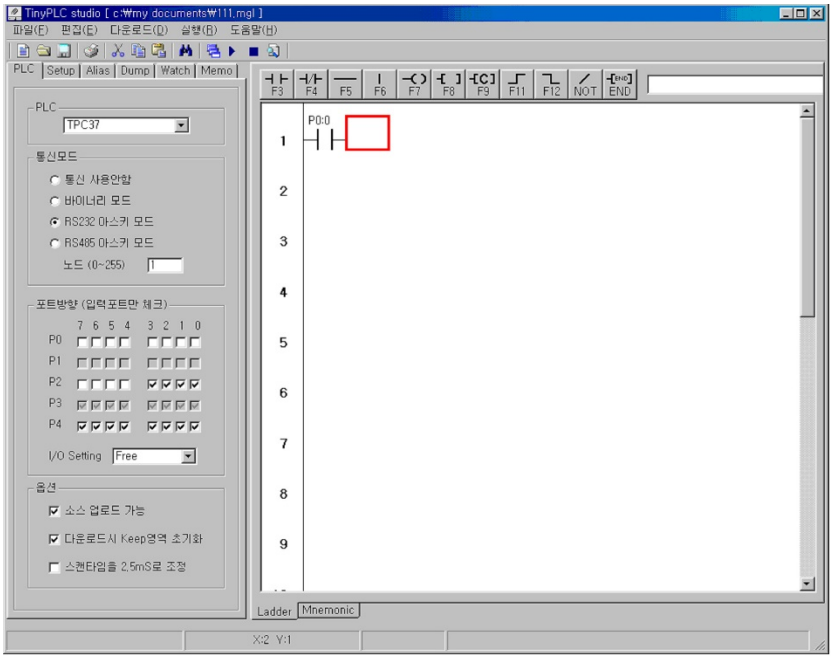
F3 을 누릅니다. (커서위치에 A 접점입력 심볼이 표시됩니다.)

화면상에 있는 레더를 모두 지우기 위해서는 파일 메뉴에 있는 새로만들기를 선택하면 됩니다. (잠깐, 저장을 해야 될 파일이면 먼저 저장을 하십시오)

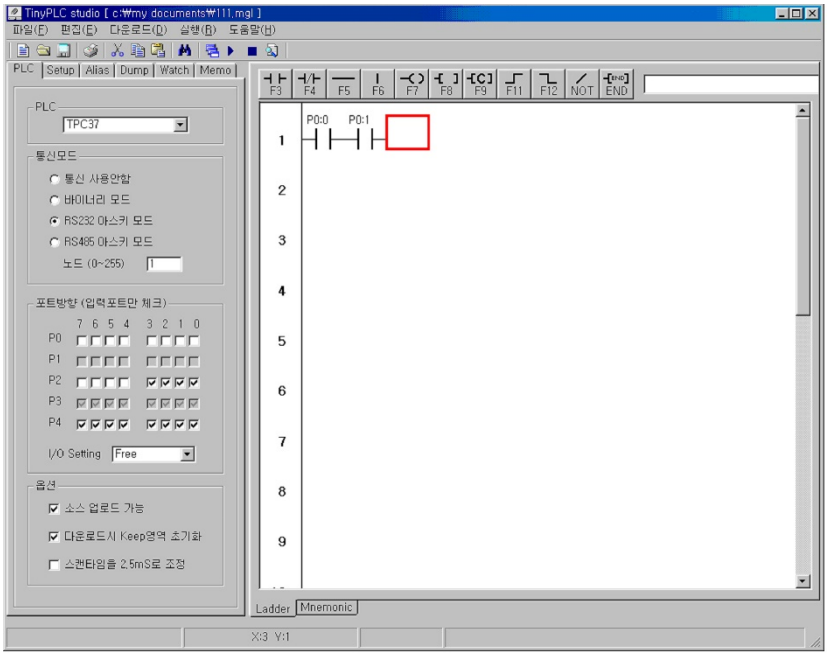


릴레이 번호 P00 을 입력하고 ENTER 키를 누릅니다.

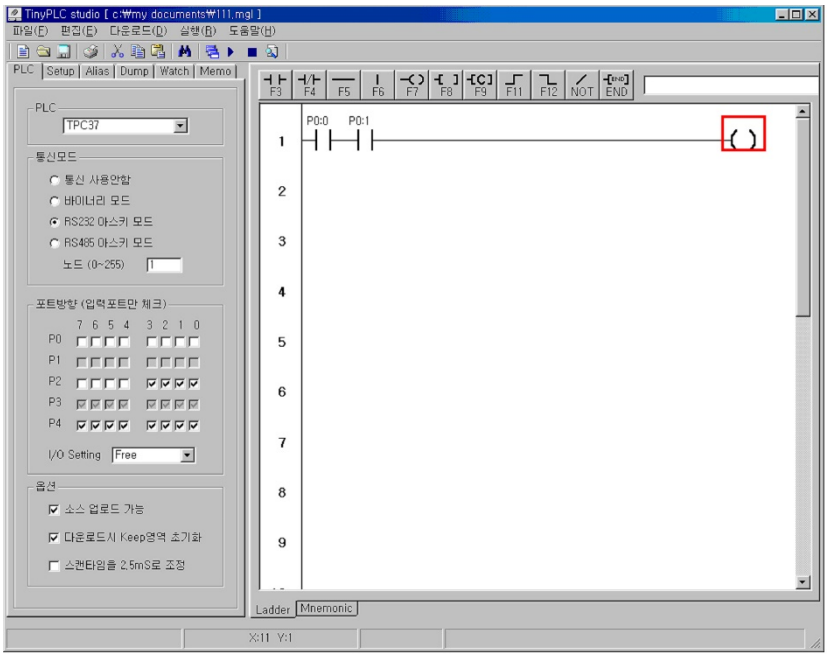
(커서는 자동적으로 다음 칸으로 이동합니다.)



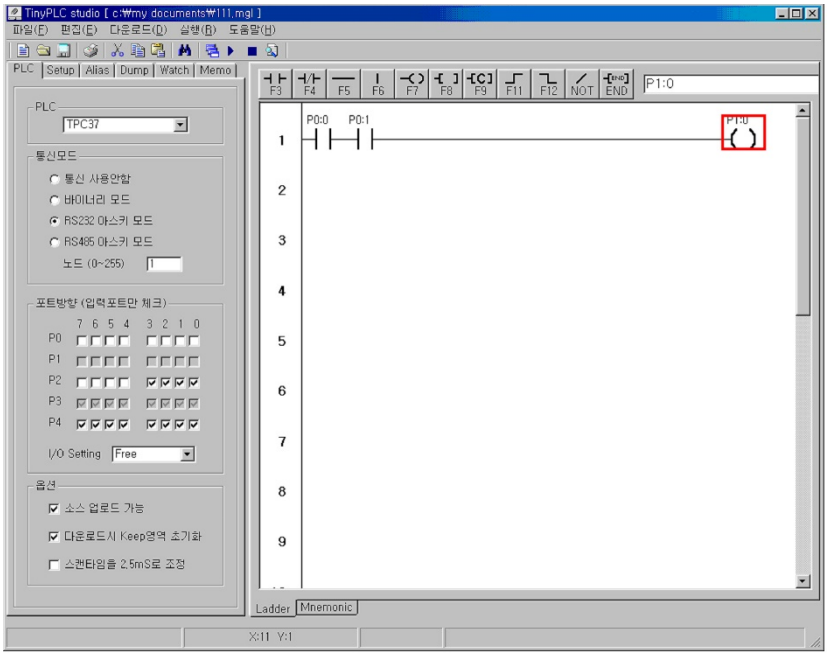
P01 을 입력하고 ENTER 키를 누릅니다. (텍스트만 입력할 경우에는 바로 전에 입력한 심볼이 자동적으로 같이 표시됩니다.)



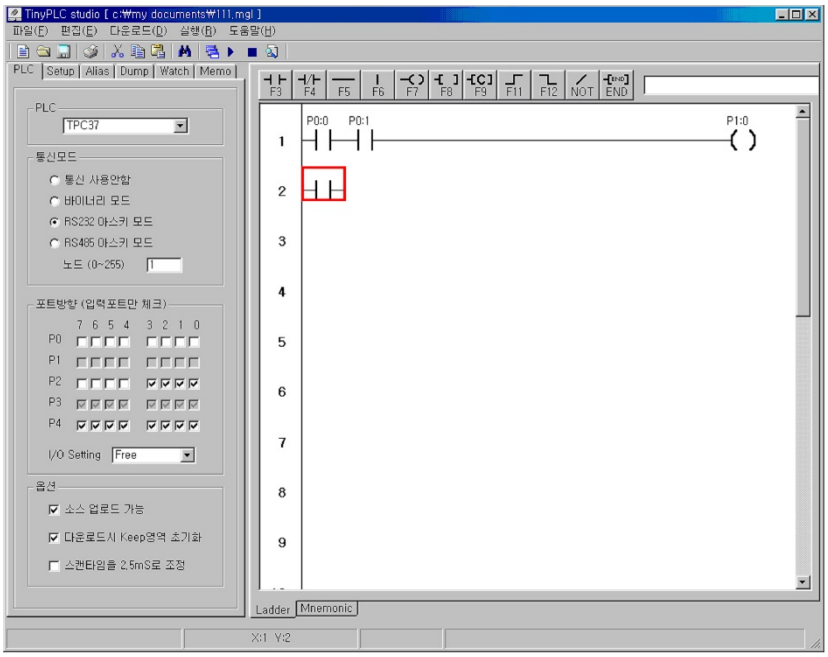
F7 키를 누릅니다. (자동으로 왼쪽 끝까지 이동해서 OUT 심볼이 표시됩니다.)



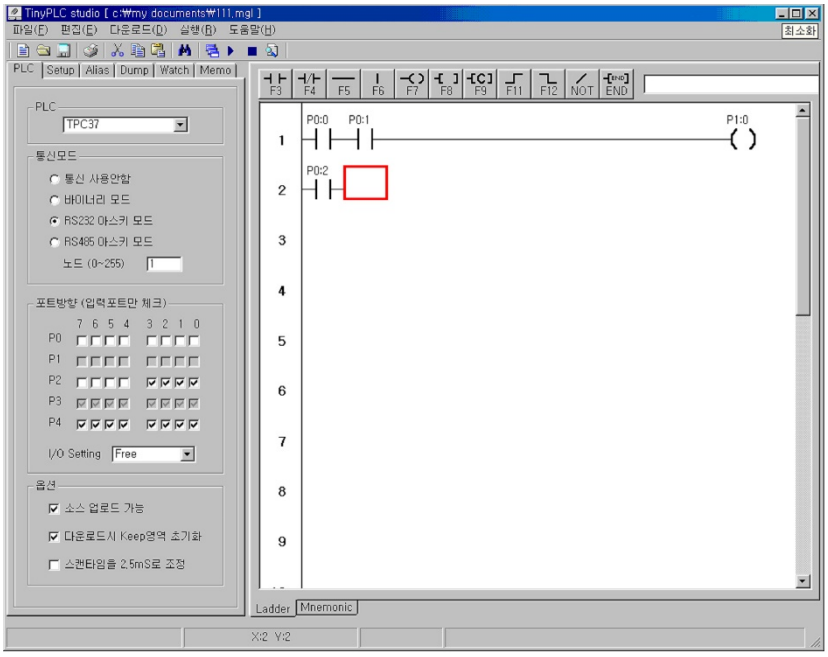
출력할 릴레이 번호 P10 을 입력하고 ENTER 키를 누릅니다.



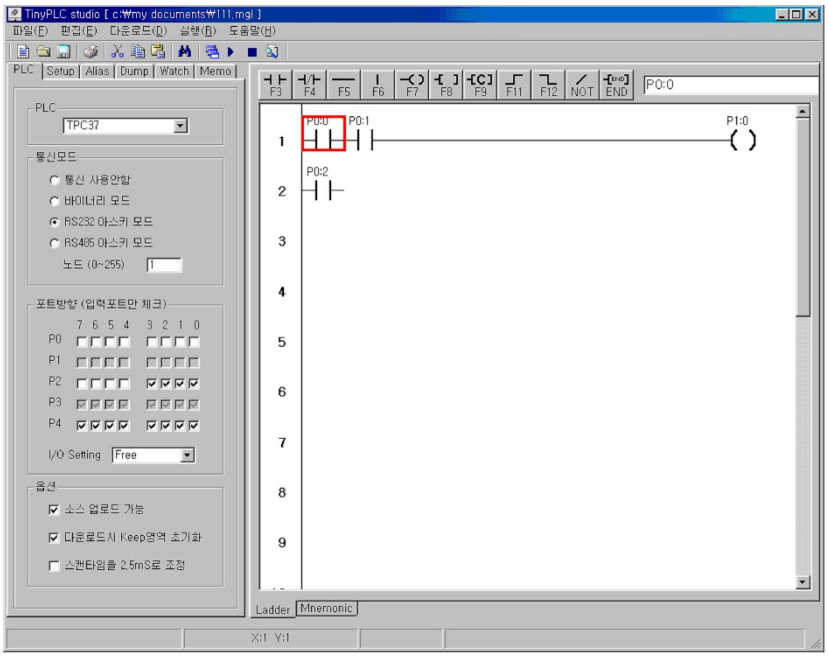
커서를 아래로 이동하고 다시 한번 F3 키를 누릅니다.



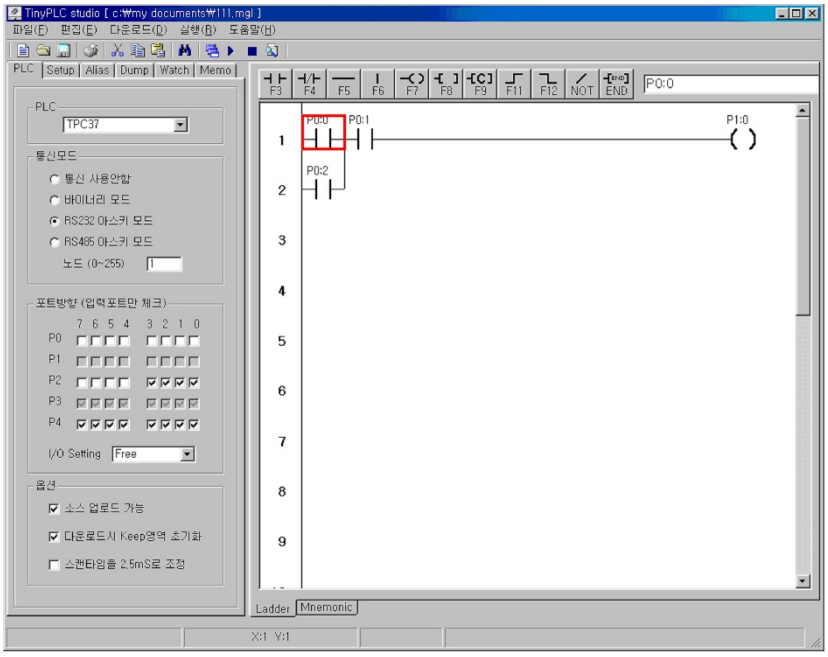
P02 를 입력하고 ENTER 키를 누릅니다.



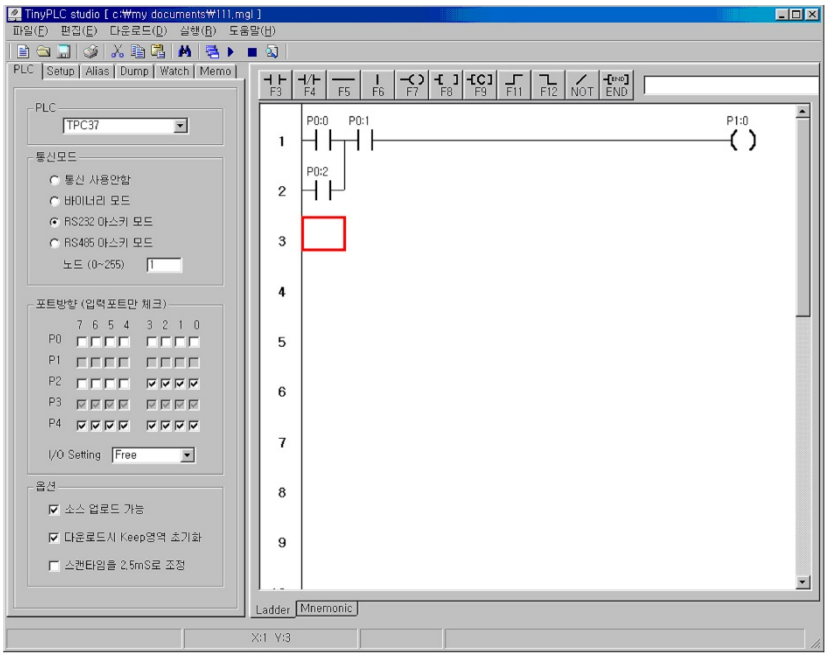
커서를 다시 가장 처음의 위치로 옮깁니다.



OR 접속선을 표시하는 F6 키를 누릅니다. (OR 접속선은 커서의 오른쪽 아래에서 생성됩니다. 같은 자리에서 한 번 더 누르면 OR 접속선이 지워집니다. : 토글사용)

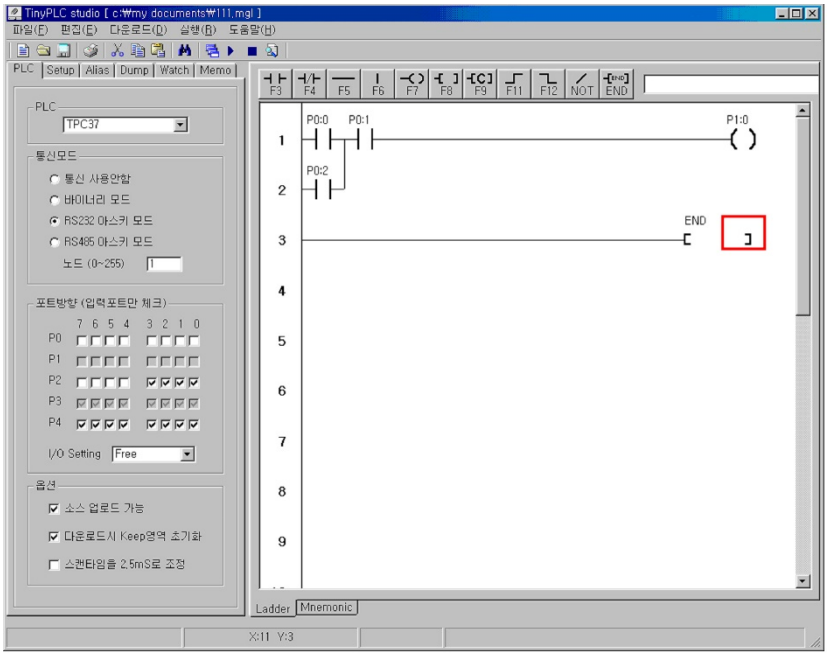


다시 커서를 3 번행으로 이동합니다.
(END 명령을 입력하기 위해서 입니다.)

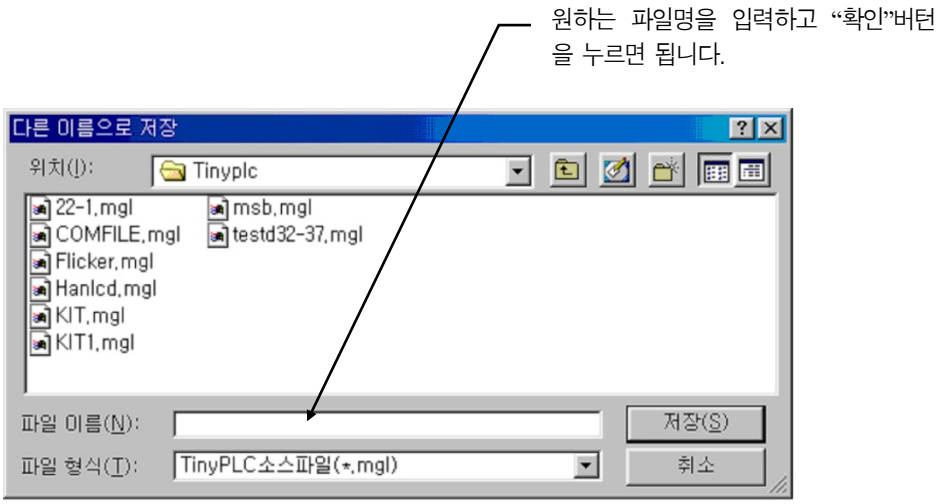


SHIFT + F8 을 누르면 오른쪽 끝까지 자동으로 연결 선을 그려줍니다. END 를 입력한 후 ENTER 키를 누릅니다.
(심볼 입력버튼 중

버튼을 누르면 END 가 자동으로 생성됩니다.)

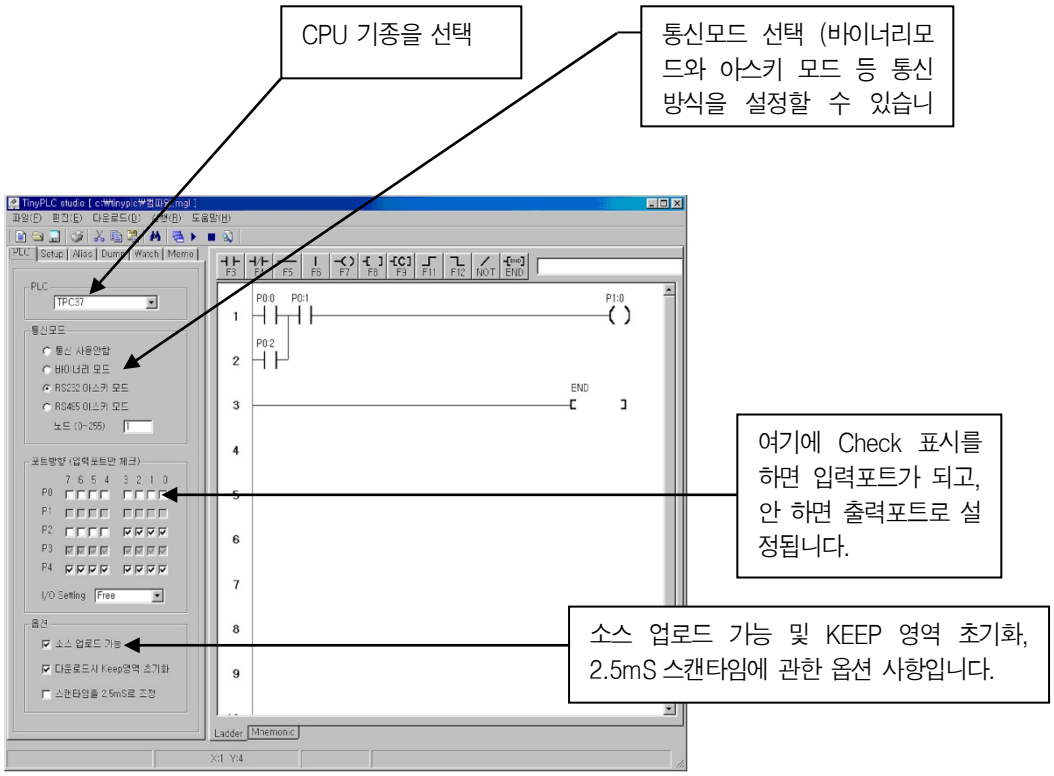


끝으로 “파일” 메뉴에 있는 “저장”을 선택해서 저장을 하면 됩니다.

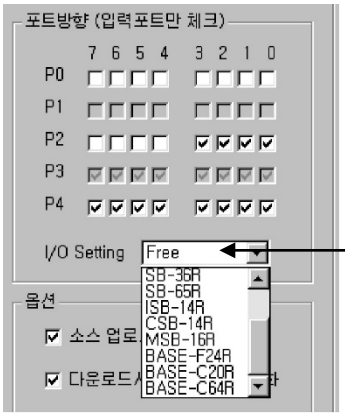


입출력 방향 및 PLC 셋업 관련

TPC3X 의 입출력 포트방향 및 기종 선택 등을 관리하는 셋업 메뉴입니다. 프로그램을 다운로드 하기전 설정을 하셔야 합니다.



I/O Port Direction 에 있는 체크 박스에 check 표시를 하면 입력 포트가 됩니다. (TPC38 의 경우 P5 이후 포트는 입출력방향이 결정되어 있기 때문에 본 메뉴에서 조정할 필요가 없습니다.)



컴파일사에서 판매하는 베이스보드나 일체형 PLC 를 사용 시에는...
I/O Setting 을 이용해 보드를 선택하면 입,출력이 **자동으로 설정** 됩니다.
수동 설정 시에는 **Free** 로 설정하고 입력 포트일 경우 해당되는 릴레이에 **check** 표시를 하면 됩니다.

통신모드

통신모드는 사용안함을 선택하세요. TPC26, TPC3X 는 통신기능을 지원하지 않습니다.

소스 업로드 가능 (No check is protect mode)

Option 에서는 TPC3X 의 기본구성과 관련된 중요한 몇 가지 사항을 결정합니다.
우선 Source Upload 를 체크하면, 나중에 소스프로그램을 업 로드 할 수 있습니다. 프로그램 작성자가 아닌 다른 사람들도 PC 또는 노트북만 있으면, 언제든지 프로그램을 업 로드해서 볼 수 있으므로, 유지보수에 많은 도움이 됩니다. 하지만, 보안을 요하는 프로그램이나, 판매용 제품에는 불필요한 기능입니다. 오히려 제품의 소스가 유출되어 쉽게 복제품을 만들 수 있도록 도와주는 셈이 됩니다 .이런 경우에는 Source Upload 를 체크하지 않으면 됩니다. 이때, 프로그램을 작성한 사람조차도 소스 업 로드는 불가능하므로, 반드시 소스 프로그램은 안전한 곳에 백업을 해두어야만 합니다. TPC3X 에는 컴파일 된 후의 오브젝트 파일만 들어가게 되므로, 이것을 다시 레더형태로 바꿀 수 있는 방법은 없습니다. 각별한 주의가 요망됩니다.

다운로드시 Keep 영역 초기화

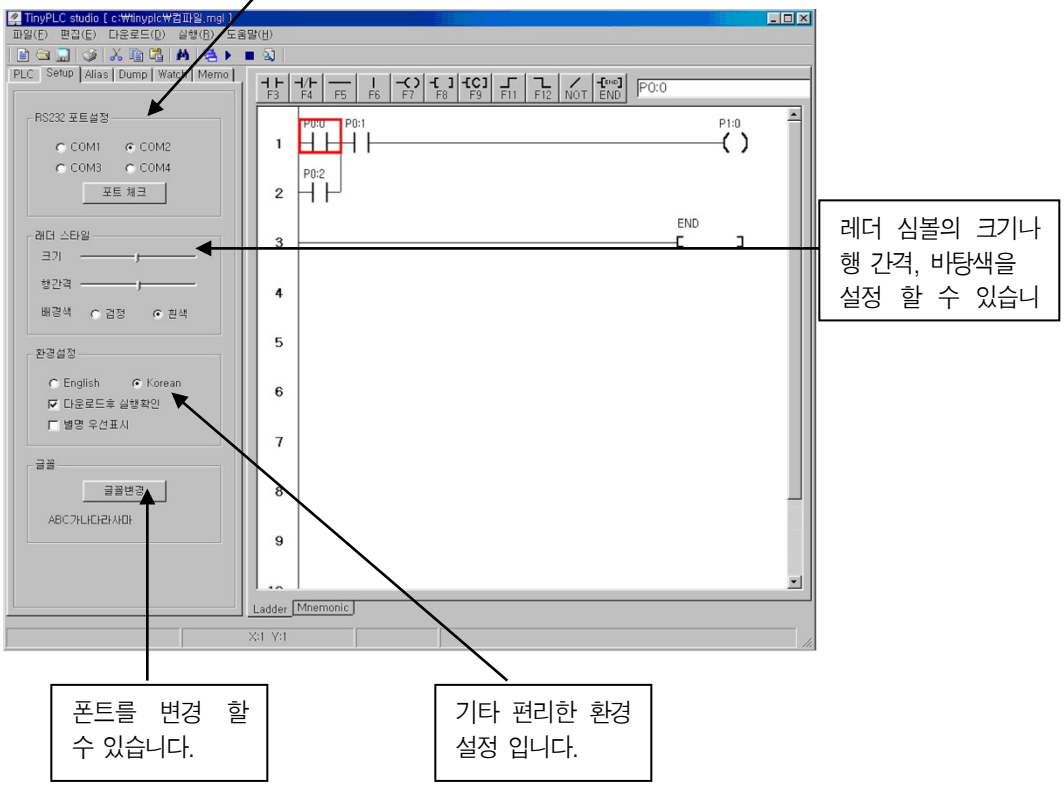
Auto clear KEEP area when download 는 다운로드시 KEEP 영역을 클리어 해 주는 기능입니다. KEEP 영역은 전원이 없어도 항상 그 값을 기억하는 영역입니다. 때로는 이 영역을 다운로드시 클리어 시켜주어야만 프로그램이 정상 동작하는 경우가 있습니다. 이때 이 메뉴를 체크 해 놓은 후 다운로드를 하시면 KEEP 영역 (C 나 T 영역 포함)은 자동적으로 모두 0 이 됩니다.

스캔타임을 2.5ms 로 조정

2.5ms Scan Time 은 스캔 타임 주기를 2.5mS 로 조정할 때 사용합니다. TPC26 의 경우에는 2.5mS 로 고정되어 있지만, TPC33 이나 TPC37 의 경우에는 디폴트로 5mS 로 되어 있습니다. 좀더 빠른 주기로 스캔을 원할 경우에는 이 메뉴를 클릭해 두면 됩니다. 단, TPC38 의 경우에는 5mS 로 고정되어 있으며, 2.5mS 로 조정할 수 없도록 되어 있습니다.

Setup 메뉴

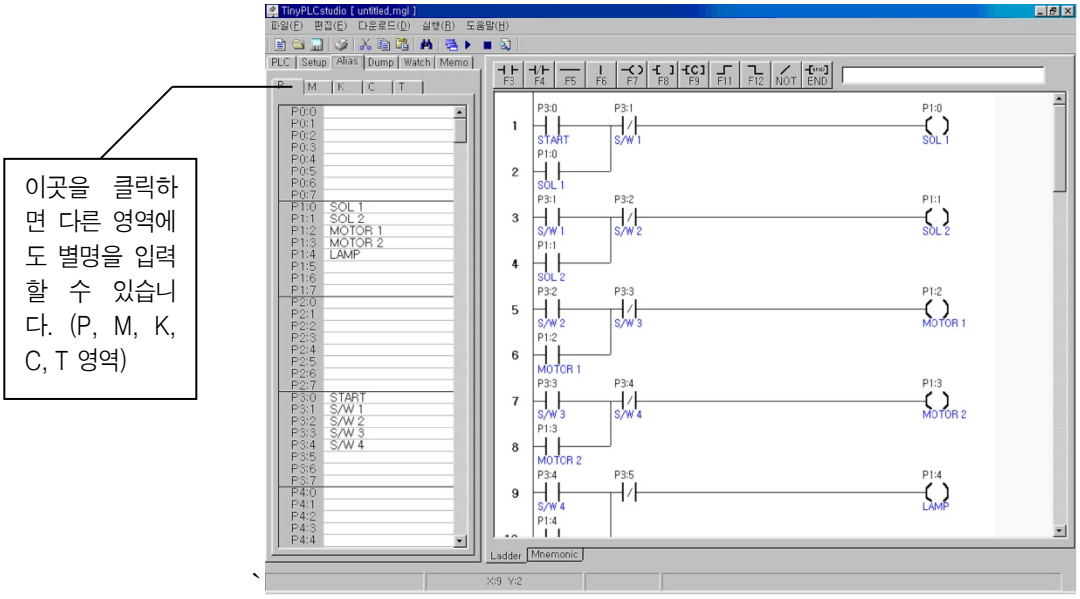
Setup 메뉴는 통신포트설정, 레더스타일, 폰트 등 유저가 원하는 환경을 직접 셋팅해서 사용할 수 있도록 제공되는 기능입니다.

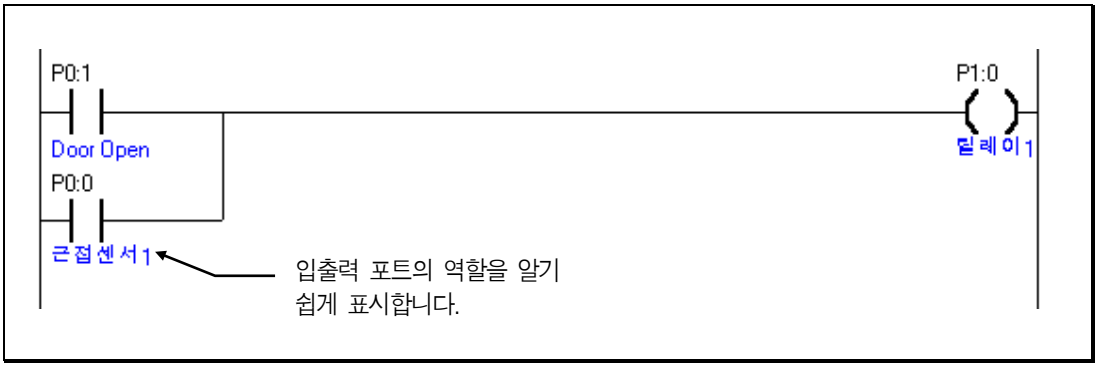


* " **환경설정** " 메뉴에 있는 " **별명 우선표시** " 기능을 사용하면 프로그램 상에 있는 릴레이 번호가 Alias 에 기록한 별명으로 변환되어서 프로그램상에 표시됩니다.

Alias(별명) 기능

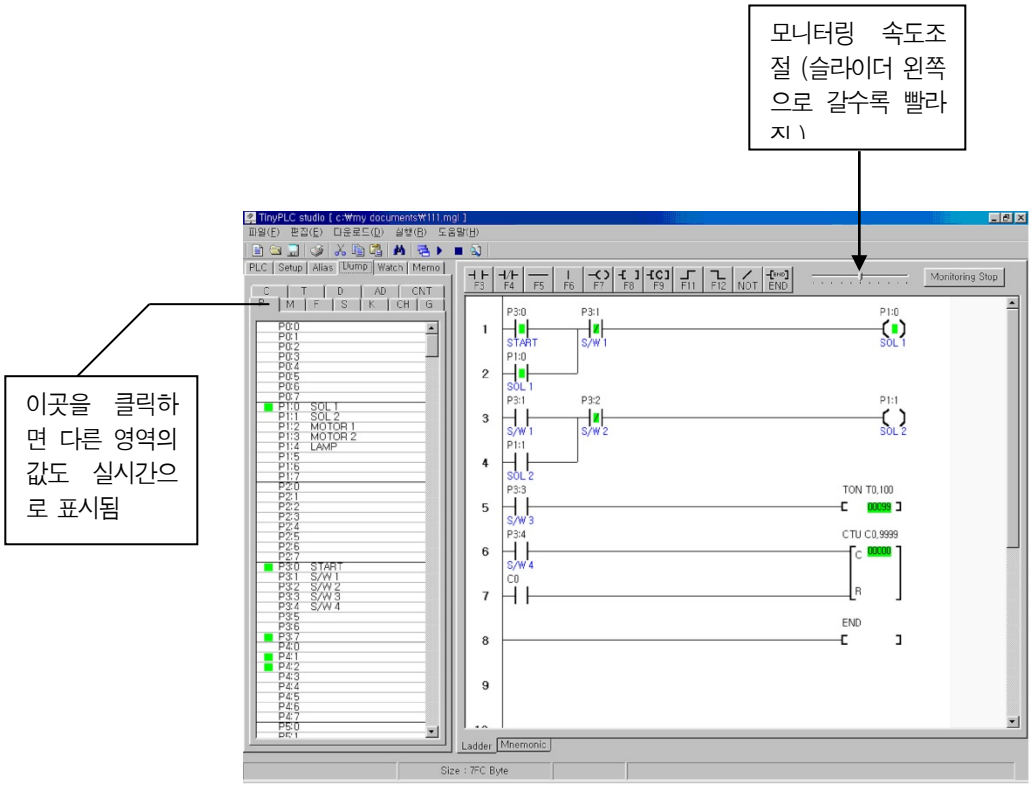
TinyPLC Studio 에는 각각의 릴레이 영역에 간단한 별명을 붙일 수 있는 기능이 준비되어 있습니다. Setup 메뉴에 있는 Alias 를 선택하면 아래와 같은 화면이 표시됩니다. 원하는 지점을 클릭한 뒤 적절한 별명을 입력한 후 (한글 사용 가능) Enter 버튼을 누르면 해당 레더 심볼 하단에 입력한 별명이 표시됩니다. 이 별명 테이블은 소스파일과 같이 저장됩니다.





Dump 기능

실행상태에 있을 때, 화면상단에 있는  (Monitor ON)버턴을 누르면 모니터링 상태가 됩니다. 모니터링 상태에는 반드시 TPC3X 모듈과 PC 가 케이블로 연결되어 있어야 합니다. Dump 기능은 모니터링 모드에서만 사용이 가능합니다. Dump 기능은 각종 접점의 ON/ OFF 상태, 카운터값, 타이머값 등이 현재의 레더 위에 실시간으로 표시됩니다.



Dump 기능 사용시에는 프로그램 편집 및 다운로드가 불가능 합니다.

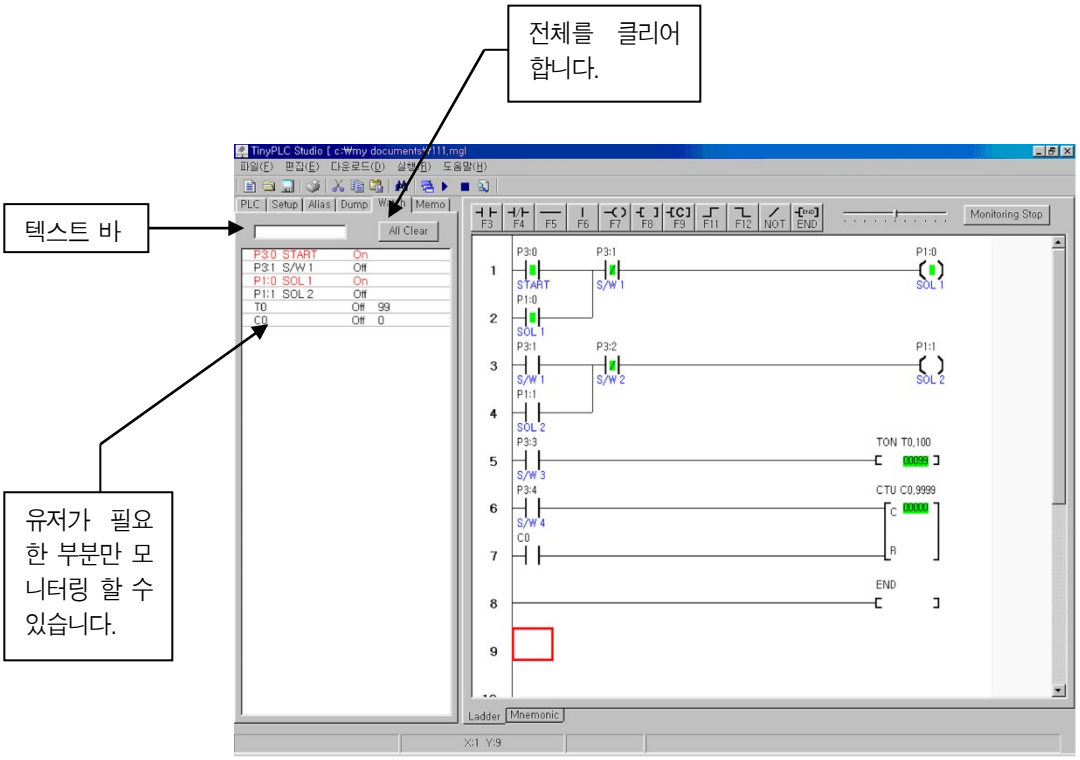
Dump 에서는 P, M, K 영역 등 바이트 단위 영역은 접점의 ON, OFF 상태를 나타내고, 데이터영역(D)과 타이머(T)와 카운터(C) 그리고 아날로그(AD)영역 등 워드영역은 현재 각각의 어드레스에 들어있는 값을 표시해 줍니다.

- 모니터링시 Dump 화면은 TPC3X 에 많은 부하를 주게 됩니다.
  Dump 보다는 Watch 윈도우를 이용하여 필요한 부분만 확인하는 것이 좋습니다.

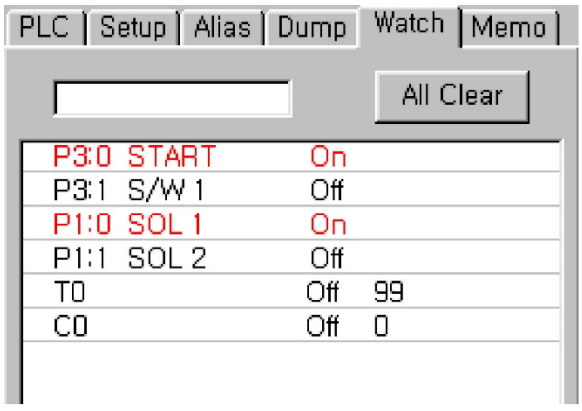
모니터링 간격이 짧아지면, 화면 갱신시간이 빨라져서 PLC 의 상태가 빠르게 반영되지만, PLC 에는 많은 무리가 따르게 됩니다. 반대로 모니터링 간격을 넓게하면, PLC 에는 무리가 줄어들지만, 화면 갱신 시간은 느려집니다.

Watch 기능

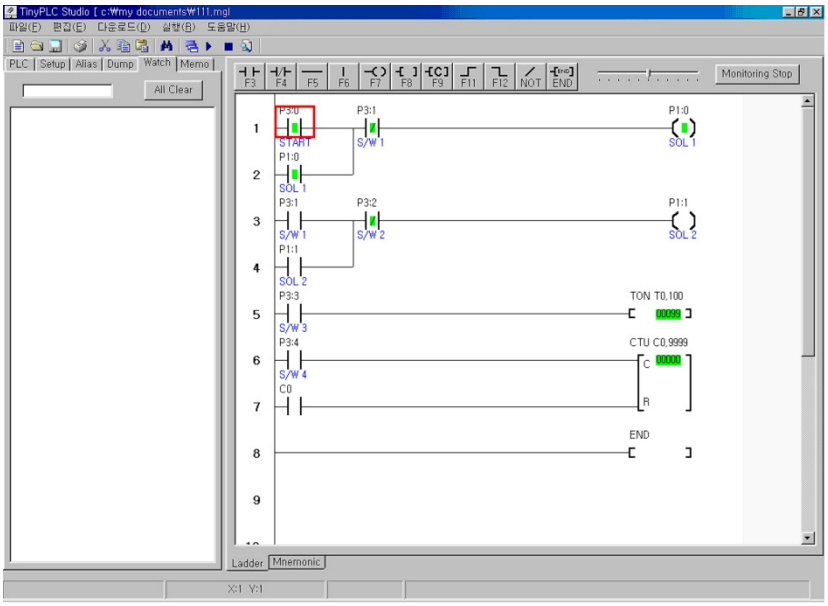
Watch 기능은 모니터링 모드에서 특정 릴레이만 따로 모아 볼 수 있는 기능입니다. 텍스트 바에 해당되는 특정 릴레이를 입력 하거나 아니면 레더프로그램의 특정 릴레이에 마우스 오른쪽 버튼을 클릭하면, Watch 창에 클릭한 릴레이가 모니터링 됩니다. Watch 기능은 모니터링 상태에서만 사용이 가능합니다.



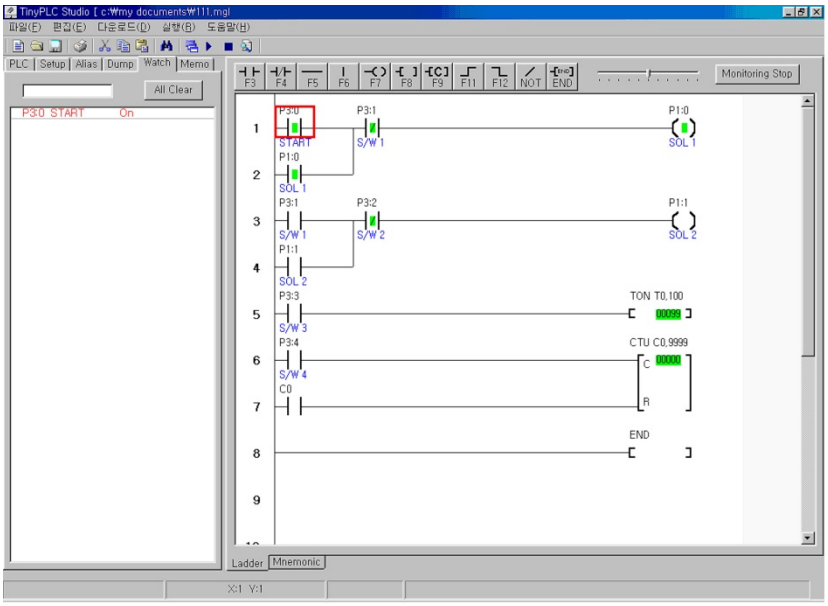
다음 그림과 같은 Watch 기능을 사용하는 방법을 한 스텝씩 설명합니다.



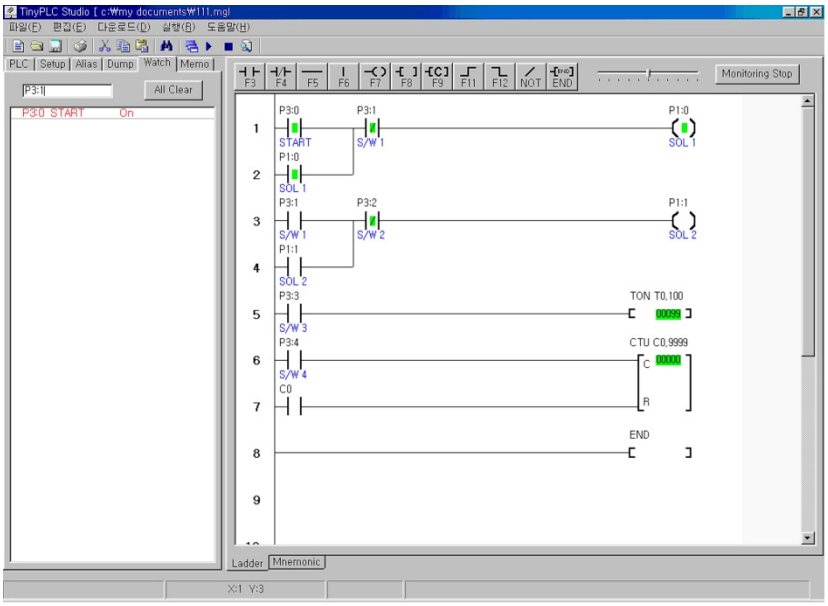
모니터링 모드로 설정한 다음 해당되는 릴레이에 마우스 커서를 옮겨 놓습니다.



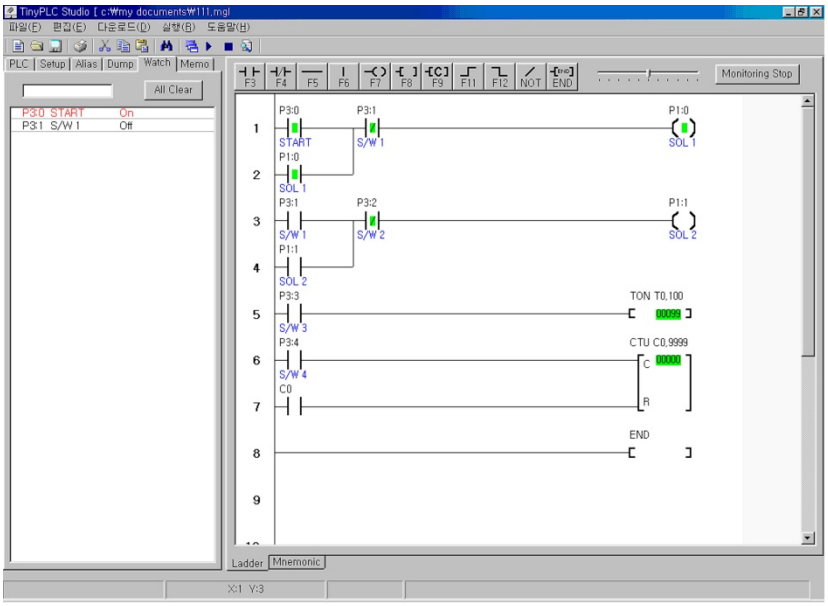
그 상태에서 마우스 오른쪽 버튼을 클릭하면 다음과 같이 모니터링 됩니다.



다른 방법은 텍스트바에 P31 이라고 입력하고 ENTER 키를 누릅니다.



다음과 같이 모니터링 됩니다.

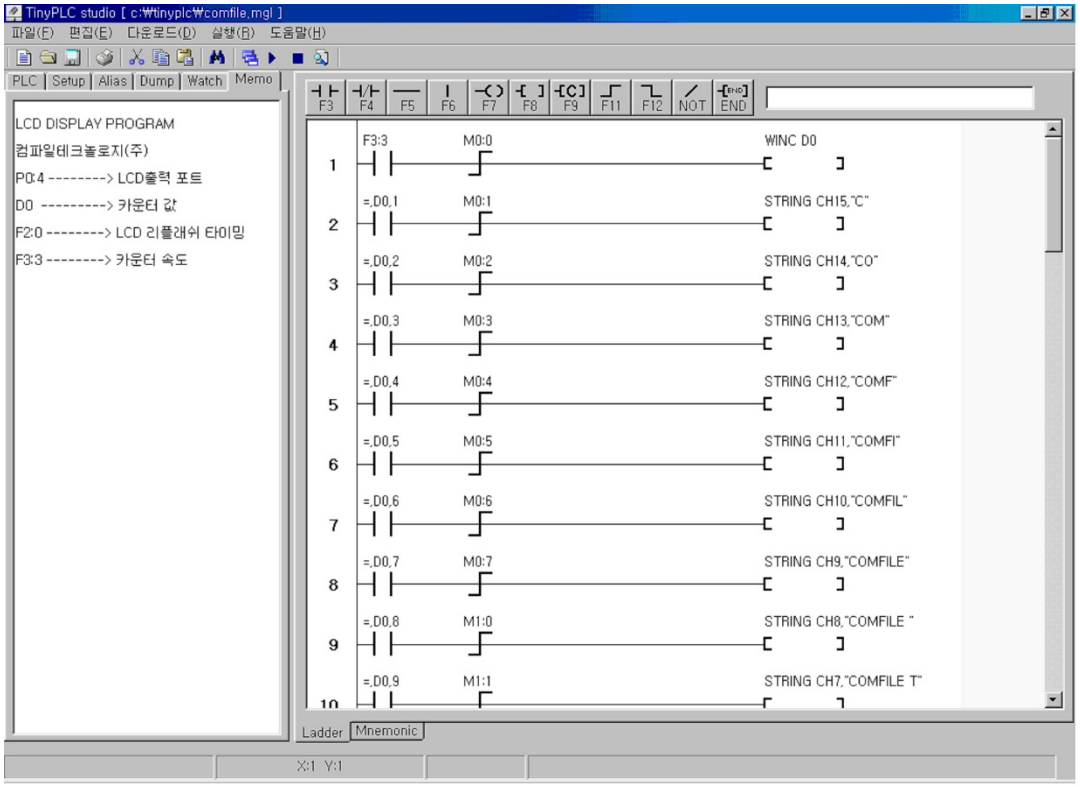


* 만약 Watch 기능에서 모니터링 되고 있는 영역을 없애고자 할 때는 해당되는 영역에 마우스 오른쪽 버튼을 누르면 그 영역만 없어집니다. 전체를 없애고자 할때는 All Clear 버튼을 누르시면 됩니다.

Memo 기능

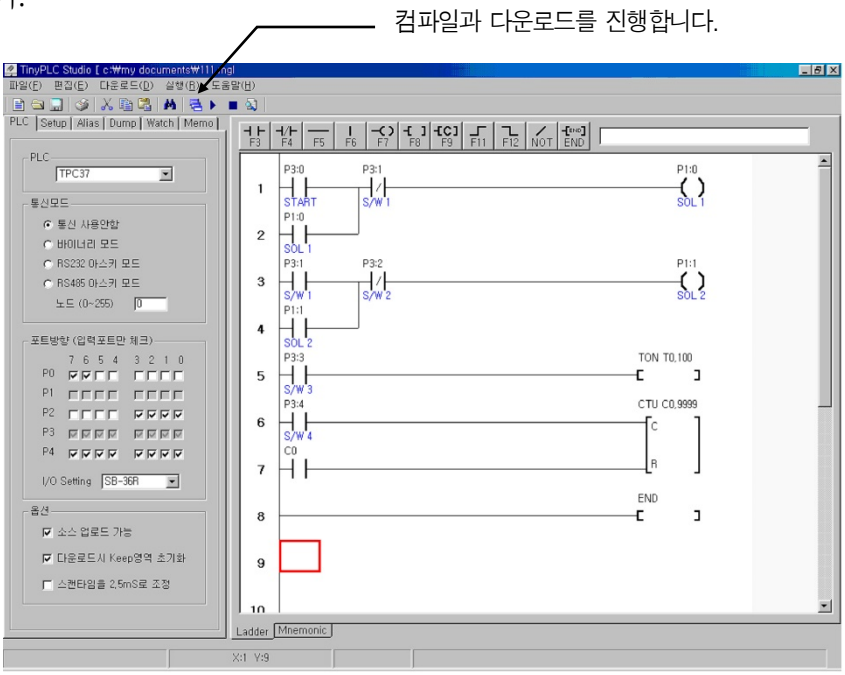
프로그램에 대한 간단한 설명이나 주의 사항 등을 입력할 수 있습니다. I/O 맵, 스팩 등 작업과 관계가 있는 내용들을 입력하면 프로그램을 작성할 때마다 I/O 맵을 찾아보는 수고를 덜게 해줄 것입니다. 주로 프로젝트명, 시작한 날짜, 작성자, I/O 맵, 납품 처 등을 입력하게 됩니다.

*소스파일과 같이 저장되지 않고 **소스파일명.txt 로 다른 파일이 생성됩니다.**



컴파일, 다운로드

소스의 입력, 포트의 방향설정, TPC3X 기종선택이 모두 끝났으면, 이제 컴파일과 다운로드를 해야 합니다. 화면 상단에 있는 AUTO 버튼을 누르면 컴파일과 다운로드를 한번에 진행합니다.

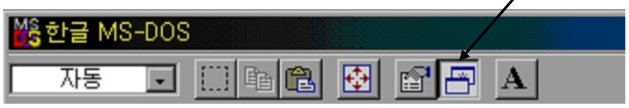


- 다운로드시 “TinyPLC NOT Found”에러가 발생하면 앞 부분에서 설명한 “문제해결 방법”을 참고하시기 바랍니다.
- 다운로드시 “Writing Logic Fail”에러가 발생하면, TinyPLC Studio 를 종료한 후 다시 기동하고, TPC3X 의 전원도 OFF 한 후 다시 ON 한 뒤 시도하십시오. (다운로드작업은 전기적으로 안정된 상황이 필요하므로, 심한 노이즈가 발생하는 환경을 피해주시기 바랍니다.)
- 컴파일 시 에러는 두 가지 종류가 있습니다. 레더의 입력 실수 등으로 생기는 “문법에러”와 컴파일 과정에서 발생하는 “컴파일 에러”입니다. 두 가지 경우 모두 화면상에 빨간색으로 ERROR 라고 표시되므로 비교적 쉽게 원인을 찾을 수 있습니다.
- 현재 TPC3X CPU 에 들어있는 레더 프로그램을 지우고자 할 때는 **Shift + Del** 키를 누르면 CPU 안에 들어있는 레더 프로그램이 전부 지워집니다.

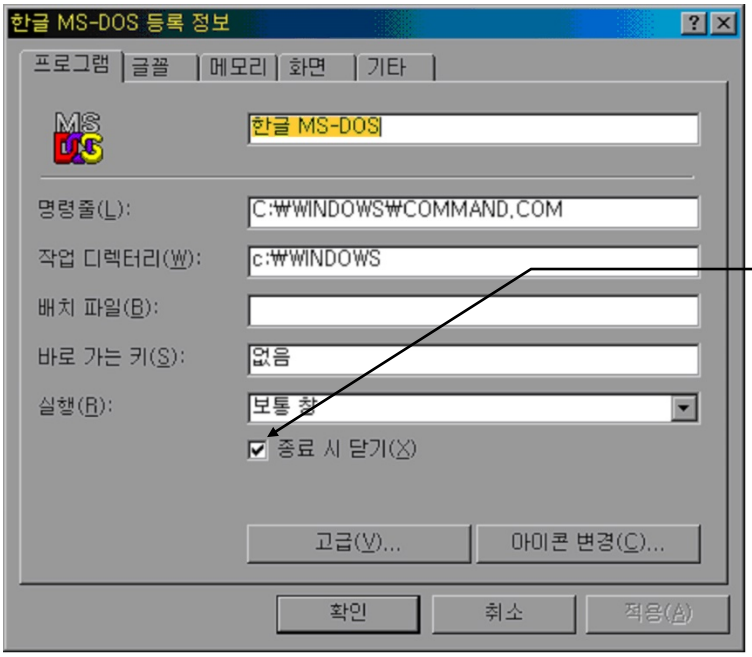
TPC26 사용시

TPC26 의 경우 다운로드중 전원이 OFF 되거나, 외부의 영향으로 인해, 칩 내부에 있는 플레쉬의 내용이 엉뚱하게 들어갈 수 있습니다. 이런 경우에는 다운로드 및 모니터링을 수행할 수 없는 “불능”상태가 되고 맙니다. 이런 경우 해결방법은 다음과 같습니다. TPC26 의 전원을 껐다 켜는 것과 동시에 < 다운로드 > 메뉴에 있는 < 오브젝트 다운로드(O) >를 누르면 다운로드 됩니다. 이후로는 정상적으로 사용할 수 있습니다.

컴파일은 실행하는 과정에서 잠깐 도스 창이 표시됩니다. **만약 도스 창이 자동으로 닫히지 않는다면, 도스창의 <등록정보>에 있는 <종료 시 닫기> 옵션을 선택**해야 합니다.



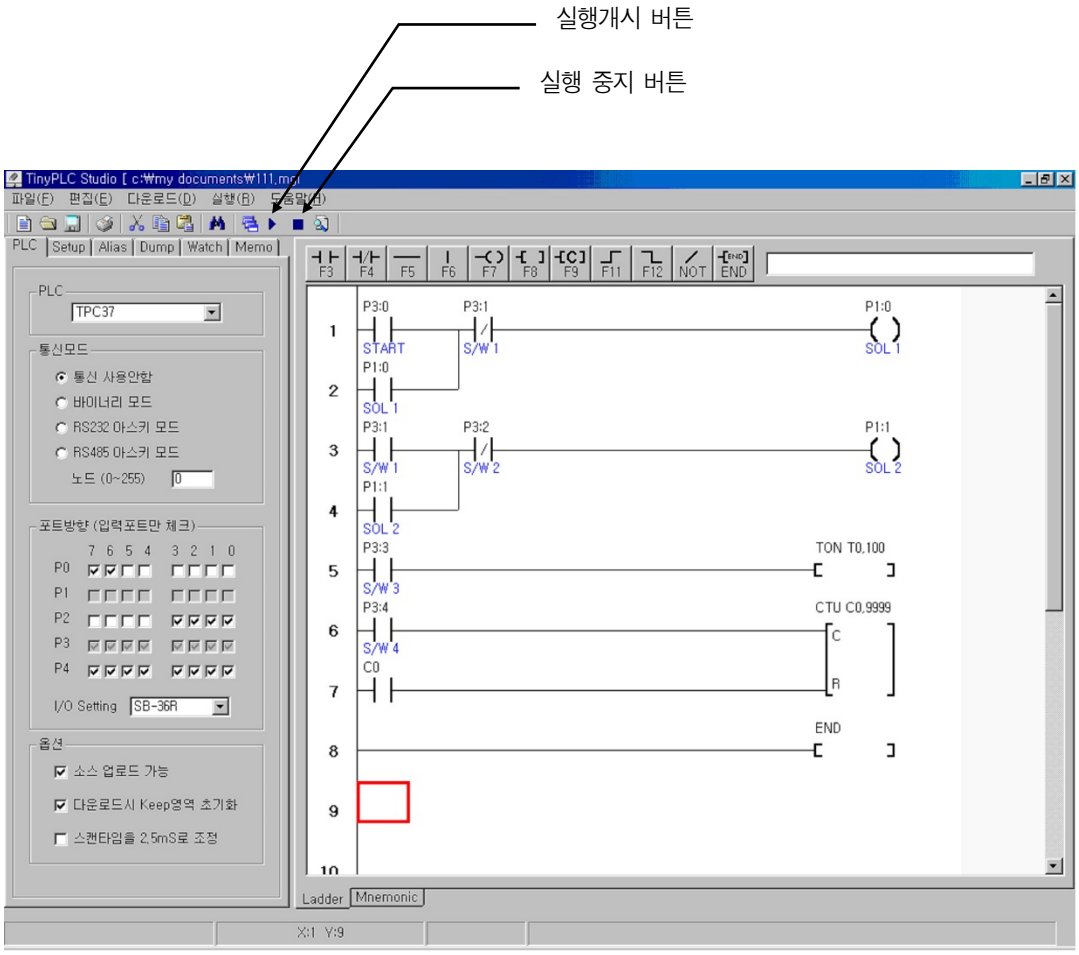
이 아이콘을 누르면 등록정보 윈도우를 표시됩니다.



이 옵션을 선택하십시오.

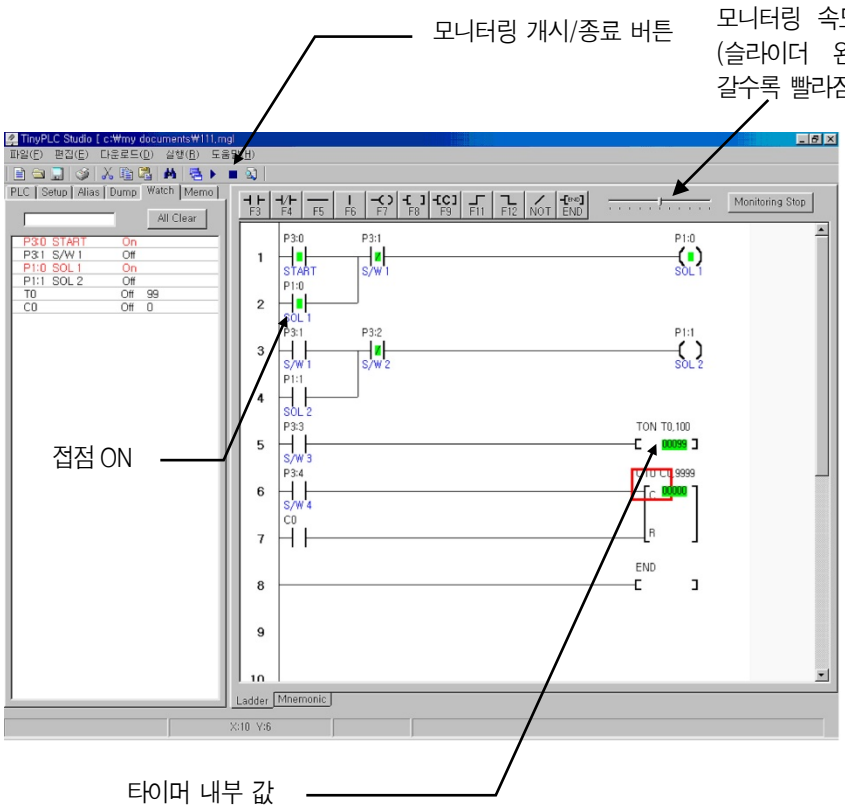
실행, 정지

다운로드가 끝나면, TinyPLC 모듈은 정지상태로 다음명령을 대기합니다.(옵션 조정가능) 이때 화면 상단에 있는 RUN 버튼을 누르면 실행 개시합니다. 실행을 정지시키고 싶을 때에는 바로 옆에 있는 실행 중지버튼을 누르면 바로 정지상태가 됩니다. (여기에서 말하는 정지는 리셋을 의미합니다.)



모니터링

실행상태에 있을 때, 화면상단에 있는 Monitor On 버튼을 누르면 모니터링 상태가 됩니다. 모니터링 상태에서는 반드시 TPC3X 모듈과 PC 가 케이블로 연결되어 있어야 합니다. (다운로드용 RS232C 포트) 모니터링 상태에서는 각종 접점의 ON/ OFF 상태, 카운터값, 타이머 값 등이 현재의 레더 위에 실시간으로 표시됩니다. 다음은 모니터링 중인 화면입니다.



주의사항

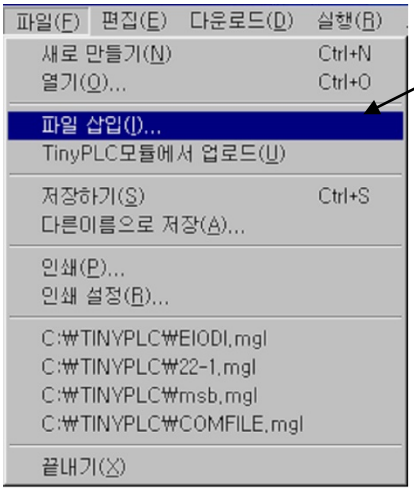
- 모니터링은 시스템에 많은 LOAD 가 걸리므로, 다른 프로그램에 영향을 줄 수 있습니다. TinyPLC Studio 를 사용할 때에는 가급적 다른 프로그램을 모두 종료한 후에 사용하기 바랍니다. 모니터링 중 PC 와의 접속이 끊어지면 모니터링이 중단됩니다.

- 모니터링 모드에서 "상위툴바"에 있는 슬라이더를 조정하면 모니터링 간격을 조정할 수 있습니다.

  모니터링 간격이 짧아지면, 화면 갱신시간이 빨라져서 PLC 의 상태가 빠르게 반영되지만, PLC 에는 많은 무리가 따르게 됩니다. 반대로 모니터링 간격이 넓게하면, PLC 에는 무리가 줄어들지만, 화면 갱신 시간은 느려집니다.

파일 삽입

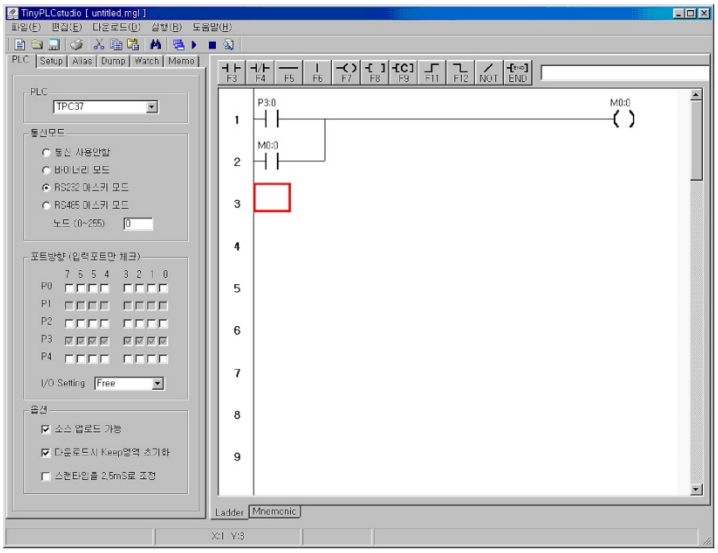
레더 프로그램 속에 다른 레더 프로그램을 삽입 할 수 있는 기능 입니다.



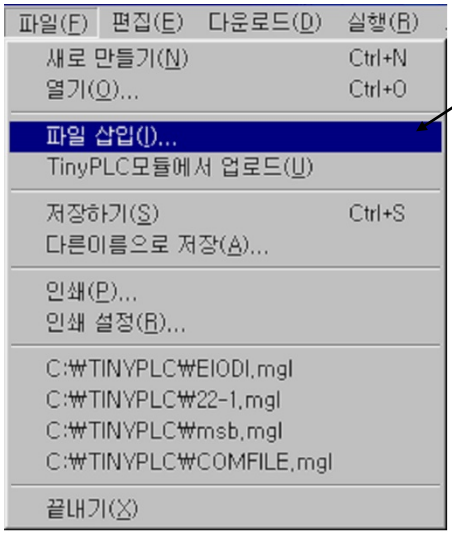
파일메뉴에 있는 파일삽입(I) 메뉴를 선택합니

* 파일메뉴의 **파일삽입**은 지금 편집하고 있는 레더의 중간에 다른 파일을 삽입할 수 있는 기능입니다.
* 다른 파일에 사용하고 있는 레더 프로그램을 복사해 올 때 사용하는 기능입니다.
* 레더만 복사해 올뿐, 셋업상태는 전혀 변동이 없다는 것을 기억해 두십시오.

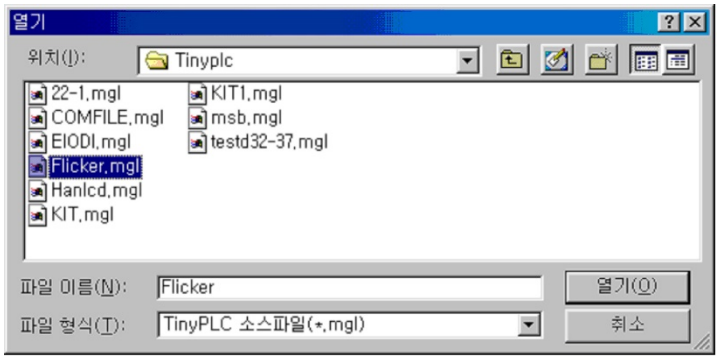
* 다음은 파일삽입 기능을 입력하는 방법을 설명한 것입니다.



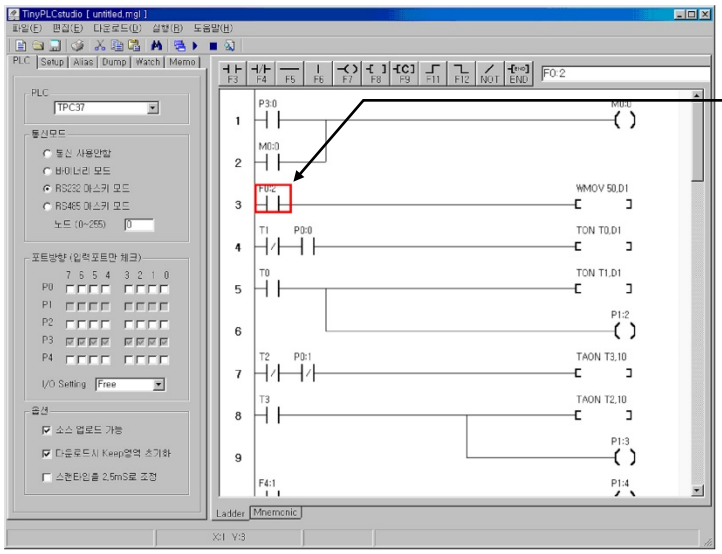
레더 프로그램을 작성하고 커서를 다른 레더 프로그램의 삽입 하고자 하는 위치에 놓아 둡니다.



파일메뉴에 파일삽입을 선택합니다.



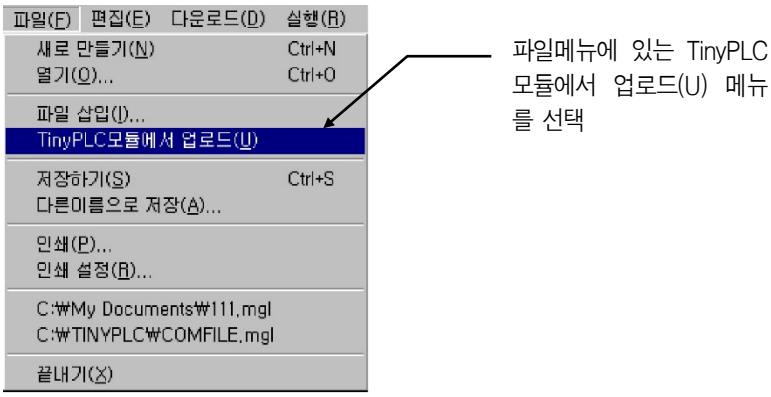
삽입하고자 하는 파일을 선택한 다음 "열기"버턴을 누릅니다.



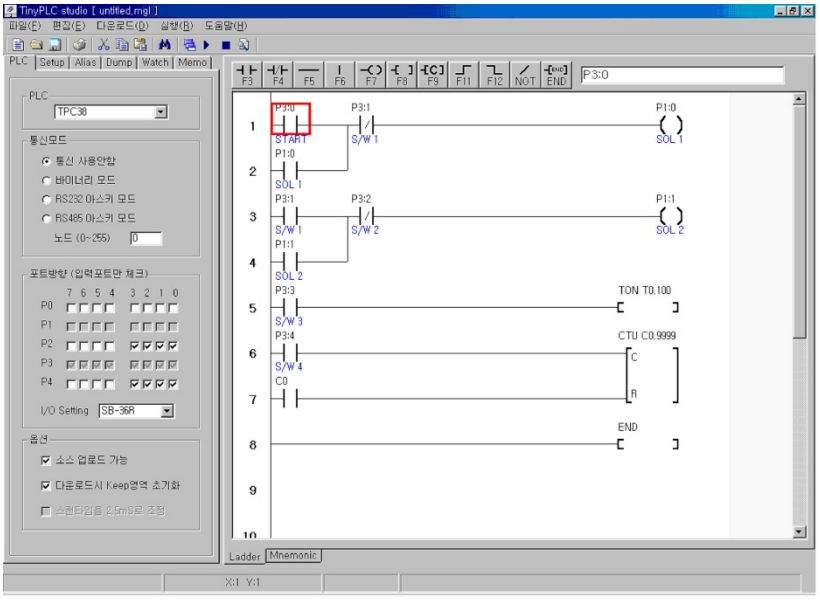
기존에 작성한 프로그램에 커서가 위치한 부분부터 "Flicker" 라는 파일이 삽입 되었습니다.

소스 업로드

유저가 작성한 레더 프로그램 (소스파일) 은 TPC3X 모듈에 함께 다운로드 됩니다. 추후에 현장에서 소스를 업로드해서 수정하거나, 디스켓에 보관할 수 있습니다.



TPC3X 모듈에서 업로드(U)메뉴를 선택하면 TPC3X 모듈에 저장되어 있는 소스를 업로드해서 화면에 표시합니다.



- 옵션을 변경하면 업 로드 기능을 금지 시킬 수 있습니다. (환경설정메뉴에 대한 설명 참조)
- **업 로드 기능을 금지하면, 소스를 보호할 수 있습니다. 다른 사람이 소스를 읽어볼 수 없으므로 제품의 무단 복제 등을 막을 수 있습니다. 실행에는 전혀 영향을 주지 않습니다.**
- 소스는 압축된 상태로 저장하고, 업 로드 시 자동으로 압축을 해제합니다.

파일의 오픈과 저장 및 새파일

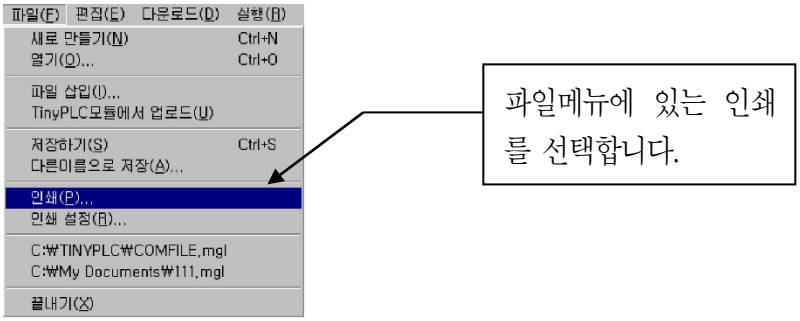
TinyPLC Studio 상단에 있는 OPEN 을 () 누르면 아래와 파일을 OPEN 할 수 있는 다이얼로그 박스가 표시됩니다. OPEN 을 원하는 파일을 선택한 뒤 확인버튼을 누르면 파일이 오픈 됩니다. (TinyPLC Studio 용 파일의 확장자는 .MGL 입니다.)

편집중인 파일을 저장할 때에는 파일 메뉴에 있는 저장하기 ()메뉴를 선택하면 됩니다. OPEN 했을 때의 이름으로 저장됩니다. 다른 이름으로 저장하고 싶을 때는 다른이름으로 저장 메뉴를 선택하면 됩니다.
TinyPLC Studio 는 OO.MGL 이라는 확장자를 가진 하나의 파일에 관련된 모든 정보를 보관하고 있습니다. 단 메모에 입력한 내용만은 OO.txt 라는 파일에 보관됩니다. 따라서 파일을 이동, 보관 할 때에는 해당 MGL 파일과 txt 파일을 같이 취급하면 됩니다.

새 프로그램을 작성 하고자 할 때는 파일메뉴에서 새로만들기()를 선택하면 됩니다.

레더의 프린터

파일메뉴에 있는 인쇄()를 선택하시면, 현재 OPEN 되어 있는 레더 프로그램이 윈도우 기본 프린터로 출력 됩니다.

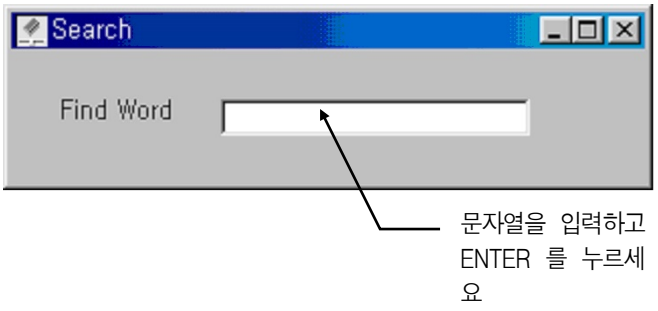


윈도우 기본 프린터 셋팅 상태를 바꾸려면, "내 컴퓨터" 안에 있는 "프린터"에서 원하는 프린터를 선택한 후 "기본 프린터로 설정"을 선택하시면 됩니다.

문자 찾기

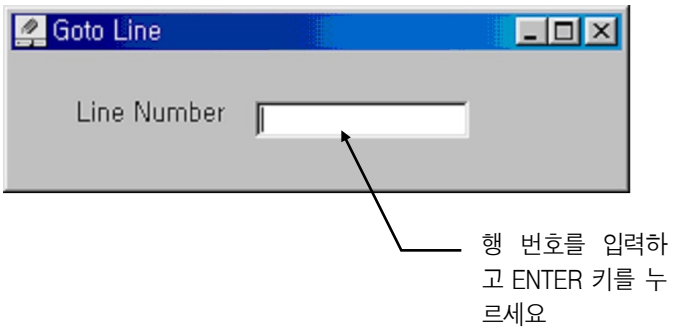
F2 키를 누르면 다음과 같은 팝업 박스가 표시됩니다. 아래의 빈칸에 찾고자 하는 문자열(릴레이 번호 등)을 입력한 뒤 ENTER 키를 입력하면, 문자열이 있는 행으로 이동합니다.

단축 아이콘은 () 입니다.



라인 이동

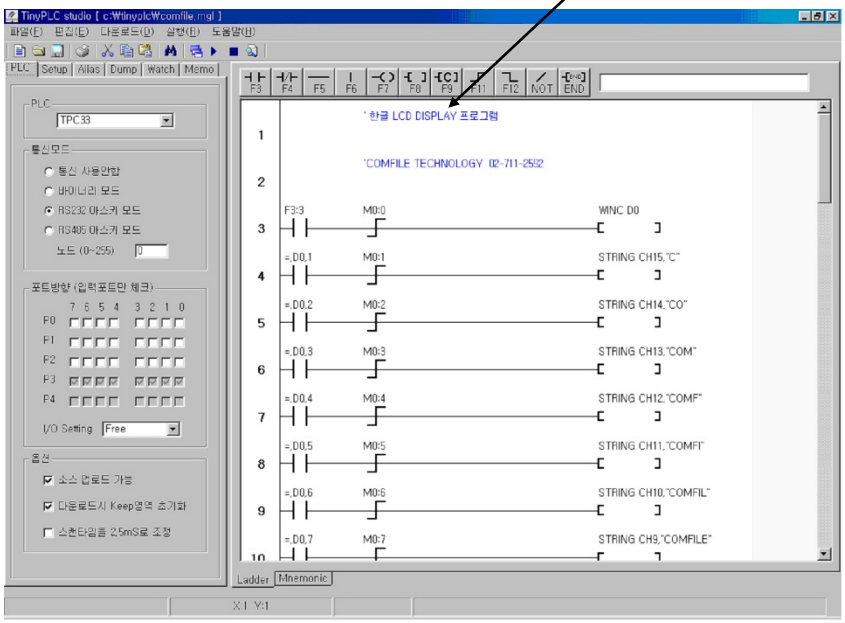
Ctrl + G 키를 누르면 이동을 원하는 행 번호를 입력할 수 있는 팝업 박스가 표시됩니다.



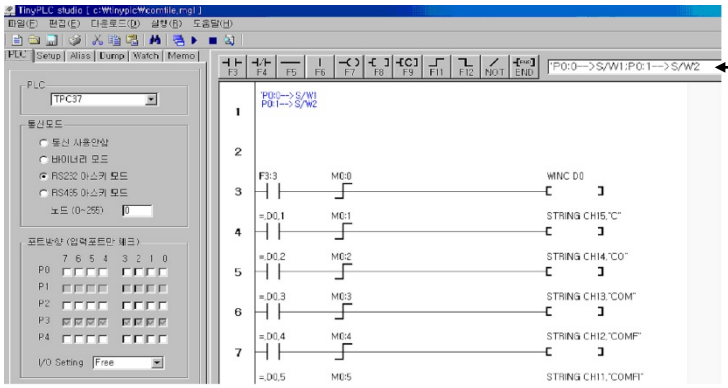
코멘트 입력

레더 프로그램 중간에 설명문을 자유롭게 입력할 수 있습니다. 코멘트는 제일 앞에는 반드시 ' (어포스트로피)기호를 붙여주어야 합니다. 코멘트 작성시 세미콜론(;)으로 줄 내려쓰기가 가능해졌습니다. 코멘트는 반드시 어포스트로피(')로 시작되어야 하며 레더 실행에 영향을 미치지 않는 아무 곳에나 작성할 수 있습니다.

코멘트는 파란색으로 표시됩니다.



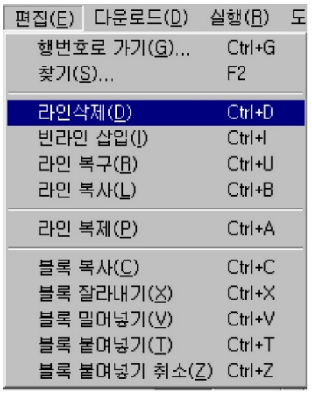
● 줄내려 쓰기 방법을 설명한 것입니다.



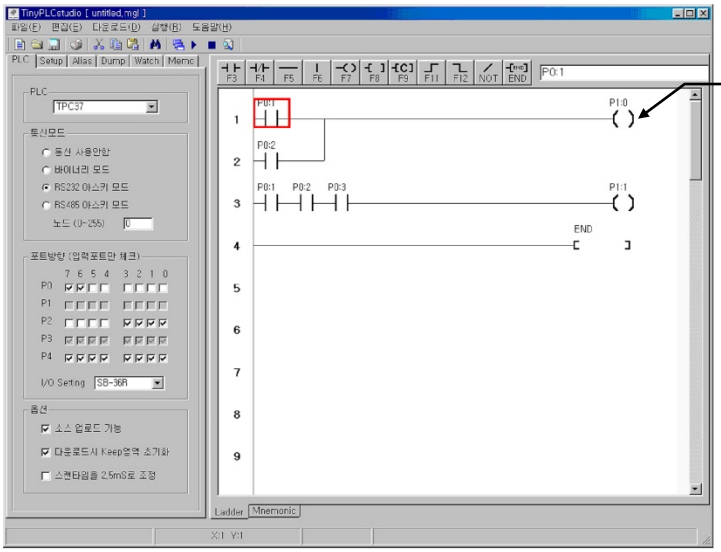
(')기호를 붙이고 코멘트를 작성합니다. 작성 중 세미콜론(;)을 입력하면 세미콜론을 입력한 뒤쪽에 있는 코멘트는 다음 라인에 쓰여집니다.

라인단위 수정, 복사

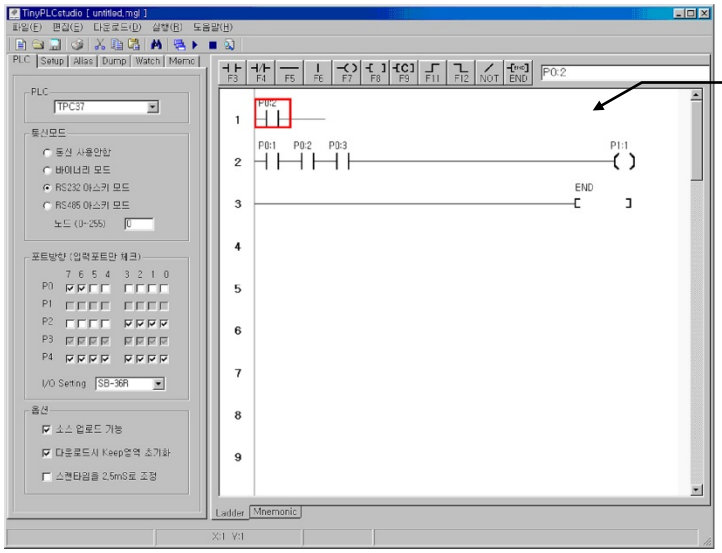
라인단위의 수정, 복사기능을 사용하면 좀더 편리하게 레더작성을 할 수 있습니다.



4 개의 라인단위 수정기능이 있습니다.

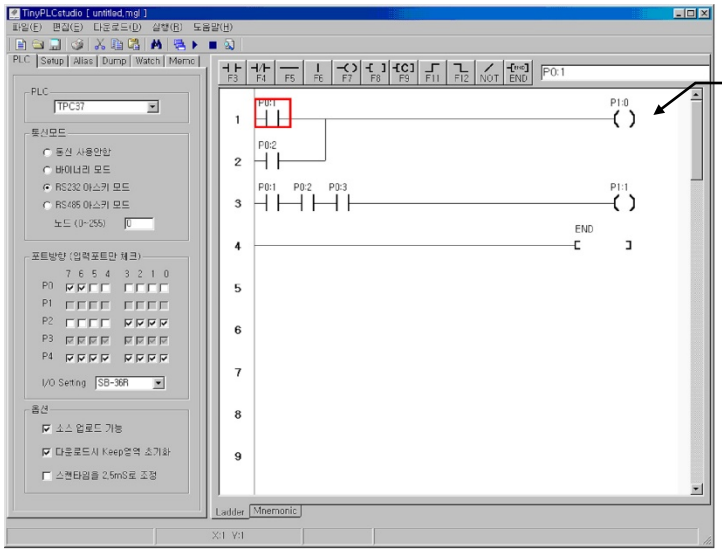


이 라인을 삭제하고 싶다면 커서를 이 라인에 위치시킨 뒤 Ctrl-D 키를 누릅니다.



좀 전에 있던 라인이 삭제되고 한 칸씩 위로 올라왔습니다.

(실제로는 삭제되지 않고 버퍼로 이동합니다.)

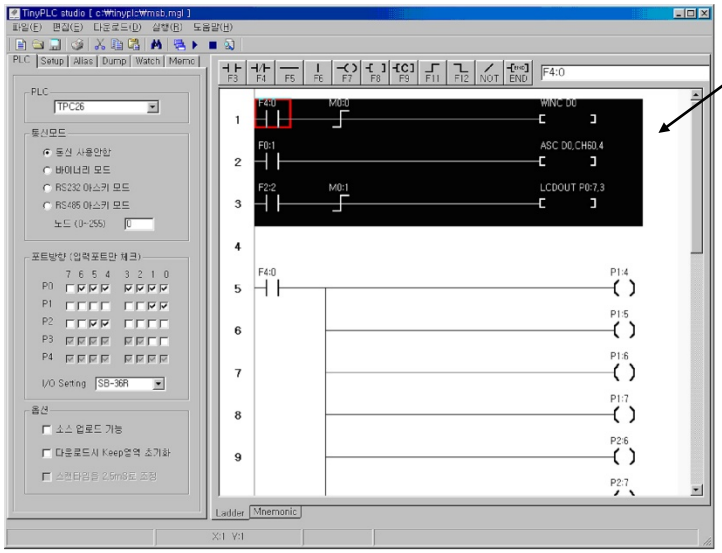


Ctrl-U 키를 누르면 앞에서 삭제되었던 라인이 다시 복구됩니다.

(버퍼에 있던 내용이 복구되는 것입니다.)

레더의 블록이동, 복사

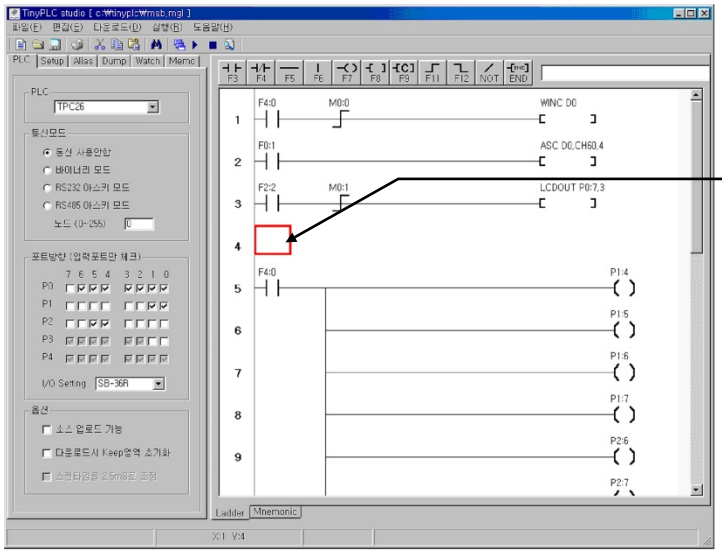
레더의 일부분을 블록으로 지정하여, 따로 보관하거나, 이동, 복사 등을 할 수 있습니다.



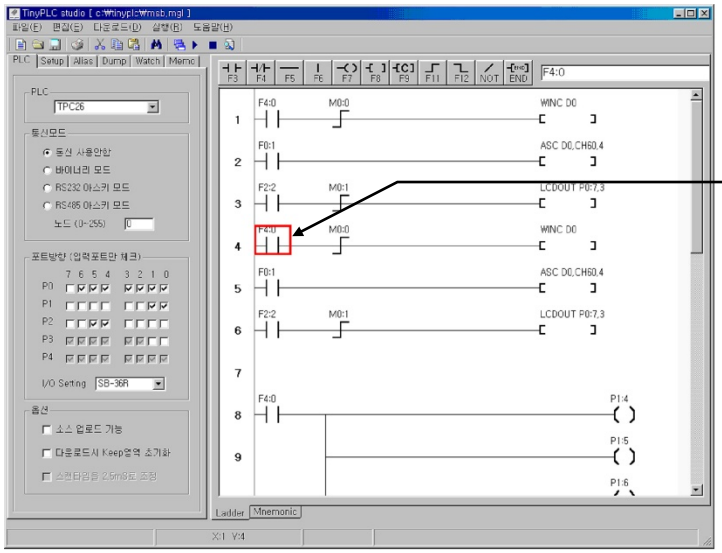
마우스로 블록을 지정합니다. (시작 지점에서 클릭한 뒤, 버튼을 누른 채로 끝 지점까지 이동)

Ctrl-C 를 눌러서 버퍼로 복사합니다.

(Ctrl-X 를 누르면 지워지면서 복사됩니다. CUT 기능)

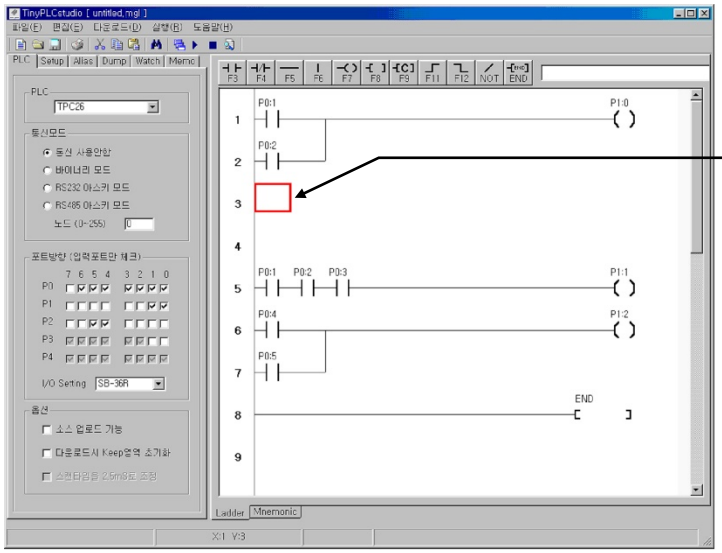


복사하고자 하는 지점으로 커서 이동

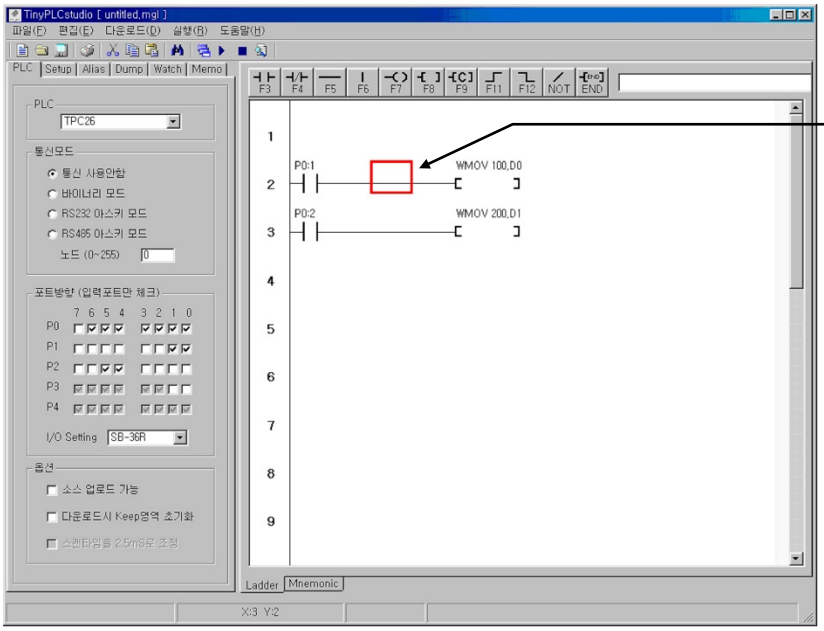


빈 칸 삽입

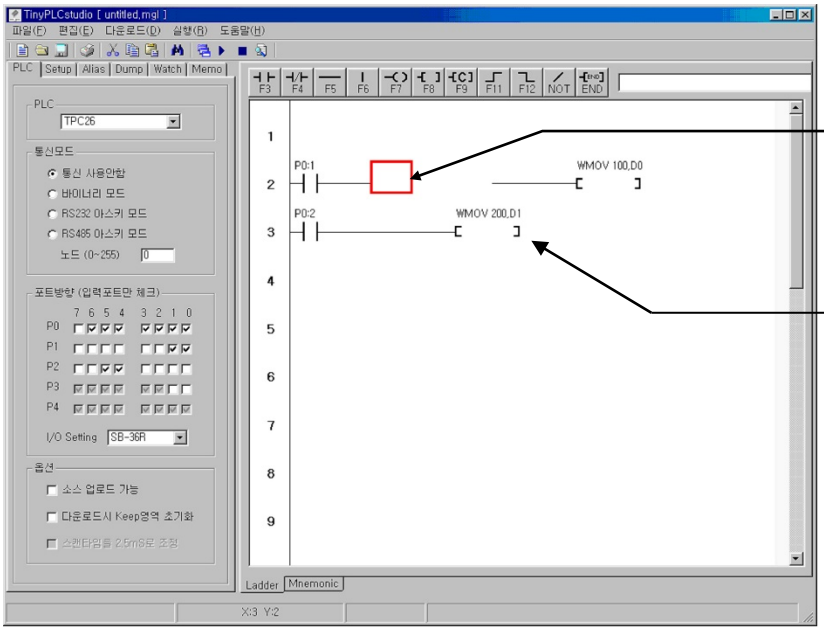
새로운 내용을 입력하기 위한 빈칸 삽입 기능입니다. 빈칸삽입을 원하는 곳에 커서를 위치시킨 후 Ctrl-I 를 누르면 됩니다.



셀 단위 삽입과 삭제



삽입 또는 삭제 하고자 하는 지점으로 이동.

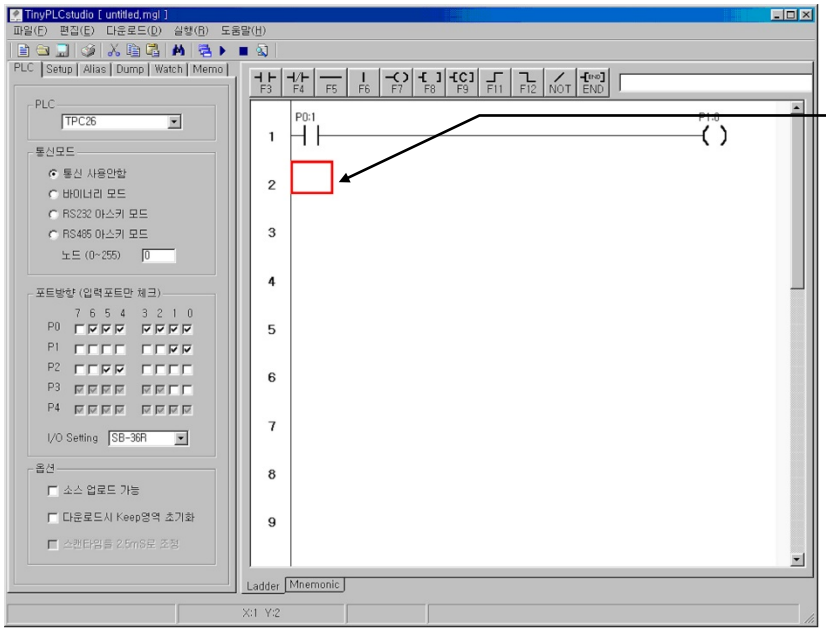


Insert 키를 3 번 누르면 빈 공백이 생기고 나머지 칸은 옆으로 밀려납니다.

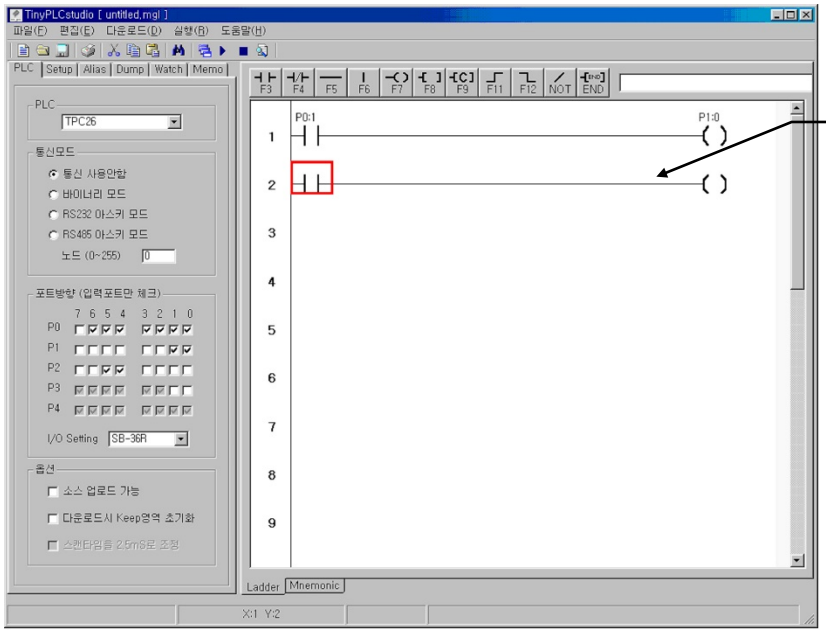
다시 Del 키를 3 번 누르면 원 상태가 됩니다.

행의 복사

똑같은 모양의 레더가 여러개 반복될 경우, 매우 편리한 기능입니다.



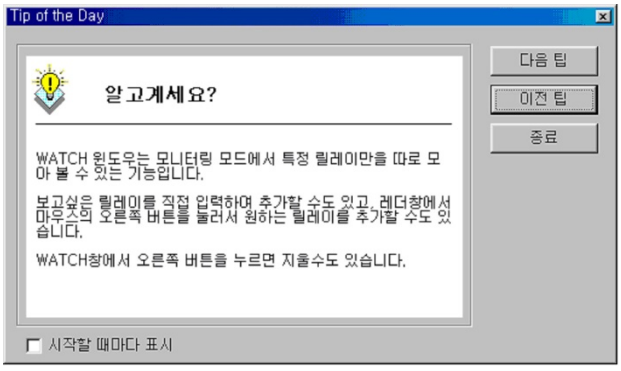
이 상태에서 Ctrl-A 를 누릅니다.



레더만 윗 줄과 같은 모양으로 삽입됩니다. (유저는 텍스트만 다시 입력하면 됩니다.)

도움말(H)

도움말을 선택하면 " **오늘의 팁(T)** " 메뉴가 있습니다. 이 메뉴는 레더 프로그램 사용시 주의사항이나 기타 편리한 기능에 대한 자세한 설명이 들어있습니다.



● **TinyPLC Studio 에 대하여(A)....** 를 선택하면 현재 TinyPLC Studio 의 버전을 확인할 수 있습니다.

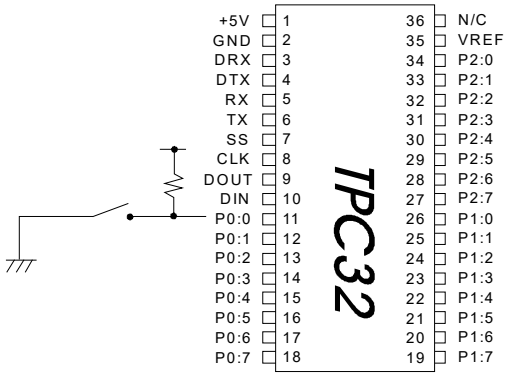
제 6 장
설치 및 구성

이 장에서는 TPC3X 의 I/O 연결 방법과 베이스 보드 설치방법에 대하여 설명합니다.

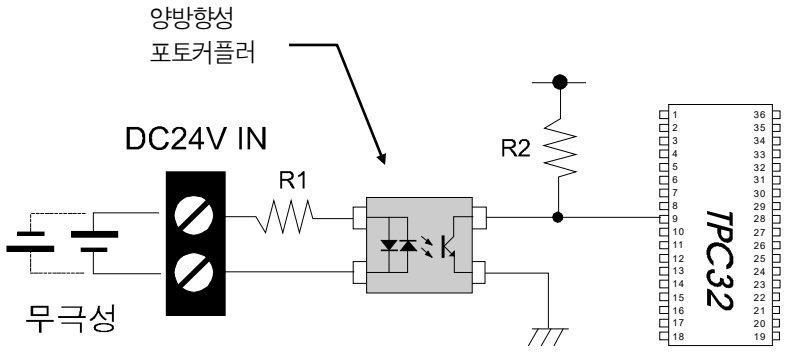
입력포트의 구성법

TPC3X 모듈의 I/O 는 일반 CMOS 타입의 TTL 칩과 동일한 특성을 가지고 있습니다. 입력으로 사용시에는 전류가 흐르지 않는 "하이 임피던스" 모드가 되며, 5V 를 1 로, 0V 를 0 으로 판단합니다.

그림과 같이 스위치 등을 직접 포트에 연결할 수 있으며, 이 때에는 반드시 그림과 같은 풀업저항 (4.7KΩ또는 10KΩ저항)을 붙여주어야 합니다. 이 회로는 스위치를 누르면 LOW (0)이 되고 띄면, HIGH (1)이 되는 B 접점 회로입니다.



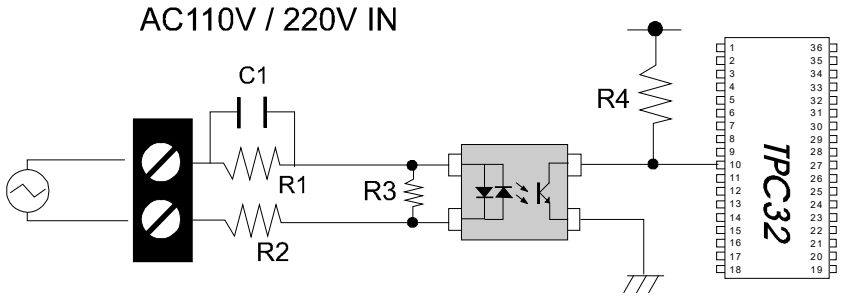
A 접점으로 하기 위해서는 풀다운을 하고, 스위치의 끝을 5V 와 연결해주어야 합니다.
5V 이외의 신호, 예를들면, 12V, 24V 와 같은 신호를 입력하기 위해서는 "포토 커플러"를 통한 신호 전달이 필요합니다. "포토 커플러"는 빛으로 신호를 전달해주는 소자입니다.



이 회로에서 R1 값을 조정하면 입력 전압을 조정할 수 있습니다. R2 는 10KΩ이 적당합니다.

입력전압	R1 값
24V	22KΩ (1/2W)
12V	10KΩ (1/2W or 1/4W)
5V	330Ω (1/4W)

다음은 AC110V 또는 AC220V 를 연결하기 위한 회로도입니다.

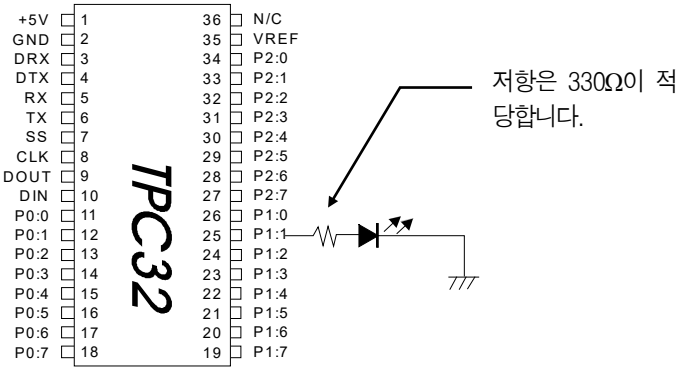


입력전압	C1 값	R1 값	R2 값	R3 값
220V	0.27uF 내압 AC250V	470KΩ (1/2W)	220Ω (1/4W)	4.3KΩ (1W)
110V	0.27Uf 내압 AC132V	470KΩ (1/2W)	220Ω (1/4W)	8.6KΩ (2W)

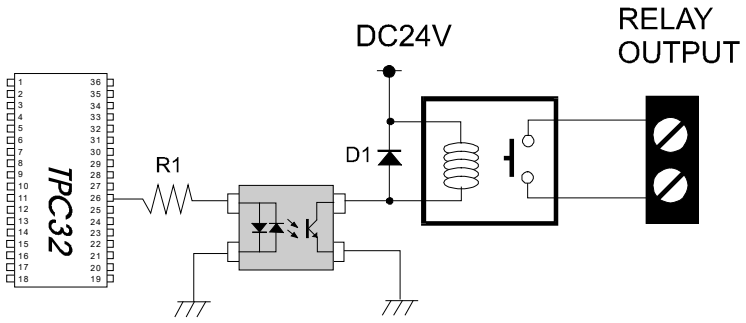
R4 는 10KΩ이 적당합니다.

출력포트의 구성법

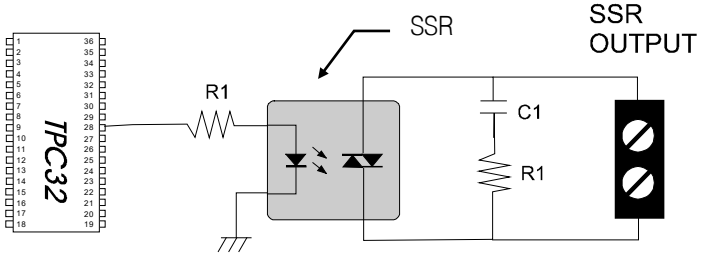
출력으로 지정된 포트는 항상 HIGH 또는 LOW 를 출력중에 있으며, 25mA ~ 30mA 의 전류를 구동시킬 수 있습니다. 아래 그림처럼 LED 하나정도는 거뜬히 구동시킬 수 있는 전류량입니다.



출력포트에 릴레이를 연결할 때에는 아래 그림과 같은 “포토커플러”를 이용해서 연결합니다. 이렇게 하면, 외부 노이즈나 ON/OFF 시 발생되는 노이즈로부터 보호할 수 있습니다.



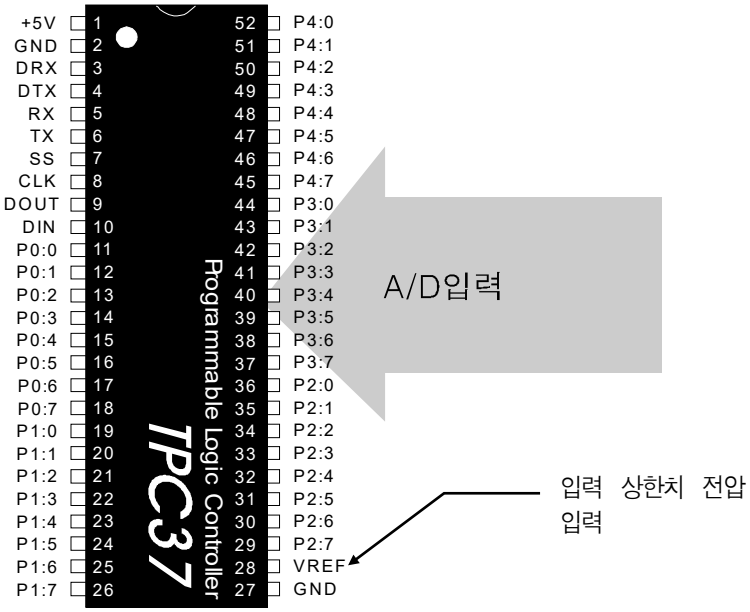
R1 은 330Ω이 적당하고, D1 은 1N4148 과 같은 일반 스위칭 다이오드를 사용합니다. (다이오드는 ON/OFF 시 발생하는 역기전류를 제거해주는 중요한 역할을 하고 있습니다.) 다음은 무접점 릴레이인 SSR 을 연결했을 때의 회로도입니다.



A/D 컨버터

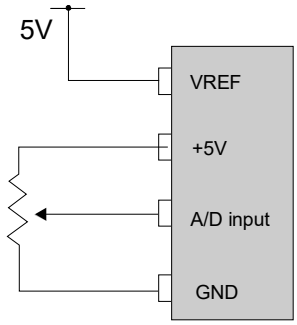
TPC3X 에는 10 비트 해상도의 8 채널 A/D 컨버터가 내장되어 있습니다. A/D 변환을 위해서 유저가 별도의 명령이나 설정을 변경할 필요는 없으며, TPC3X 동작 중에는 항상 A/D 변환을 실행하고 있습니다. 변환 결과는 AD0 ~ AD7 영역에 저장됩니다.

결과값이 저장되는 곳	TPC32 의 변환포트	TPC37 의 변환포트
AD0	P2:0	P3:0
AD1	P2:1	P3:1
AD2	P2:2	P3:2
AD3	P2:3	P3:3
AD4	P2:4	P3:4
AD5	P2:5	P3:5
AD6	P2:6	P3:6
AD7	P2:7	P3:7



VREF 단자는 A/D 신호의 입력 상한 전압을 결정해주는 단자입니다. VREF 에 5V 를 입력하면, 0V~5V 사이의 전압을 1024 등분하는 것이고, 3V 를 입력하면 0V~3V 사이의 전압을 1024 등분하는 것입니다. (**TPC33 의 경우** A/D 입력포트가 입출력 겸용이므로, **A/D 입력을 받는 포트는 반드시 입력으로 설정해야 합니다.**)

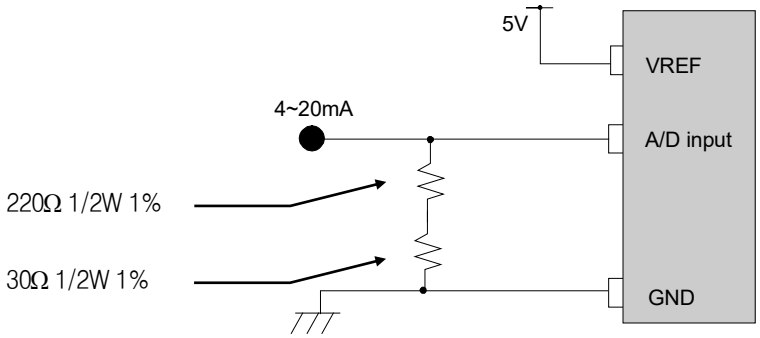
아날로그 입력 회로 예



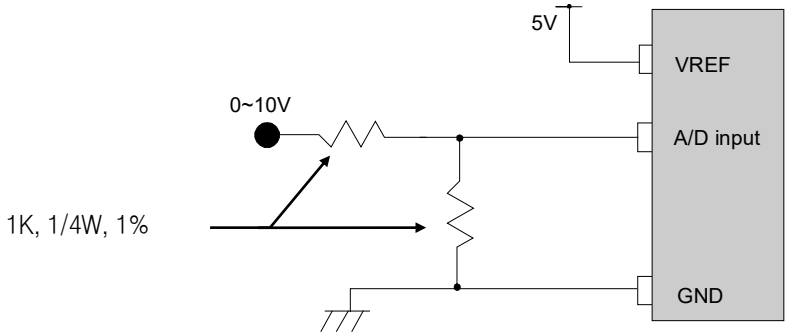
포텐션메터 (Potentionmeter)를 사용할 경우의 회로도입니다. 타이머나 카운터의 셋팅 값을 유저가 선택하도록 하는 경우에 사용하거나, 온도 값 등을 셋팅하는 용도로 사용할 수 있습니다. 포텐션메터는 1KΩ ~ 5KΩ이 적당합니다.

주의사항 : A/D 포트에 VREF 를 넘는 전압이 인가되면 안됩니다.

다음은 4~20mA 를 출력하는 센서종류와 연결했을 때의 회로도입니다.

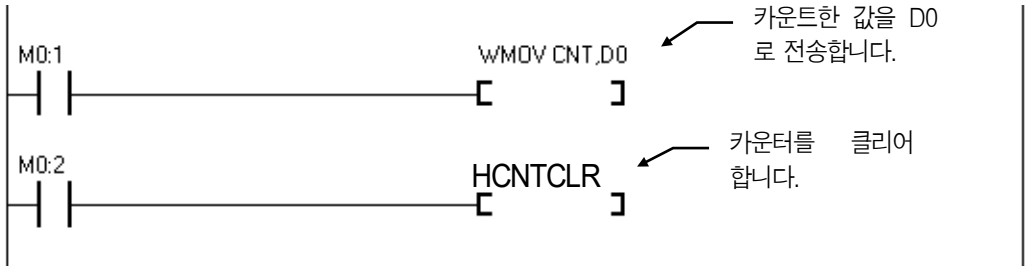


다음은 0~10V 입력시의 회로도 입니다. (저항 2 개를 써서 분압처리 합니다.)



고속카운터

TPC3X 에는 16 비트 고속카운터가 내장되어 있습니다. TPC37 의 경우 포트 P0:6 과 겸용하도록 되어 있으며, 고속 카운터로 사용시 반드시 입력으로 설정해야 합니다. TPC3X 에서는 고속카운터를 사용하기 위해, 특별한 조작이 필요 없으며, 입력 포트로만 설정되어 있으면, 항상 P0:6 포트로 입력되는 펄스를 카운트해서 CNT 영역에 저장합니다. 이를 읽기 위해서는 WMOV 등의 전송명령어를 사용합니다.



위의 레더를 보면, M0:1 이 ON 되면 CNT 의 값을 D0 로 전송합니다. DO 에는 M0:1 이 ON 되었을때까지 카운트한 결과가 저장되는 것입니다. M0:2 가 ON 되면 카운트영역은 클리어 됩니다.
(HCNTCLR 는 고속카운터의 값을 0 으로 만드는 명령입니다.) TPC32 의 경우 포트 P0:1 을 고속카운터 입력 포트로 사용합니다.

고속카운터가 필요한 이유

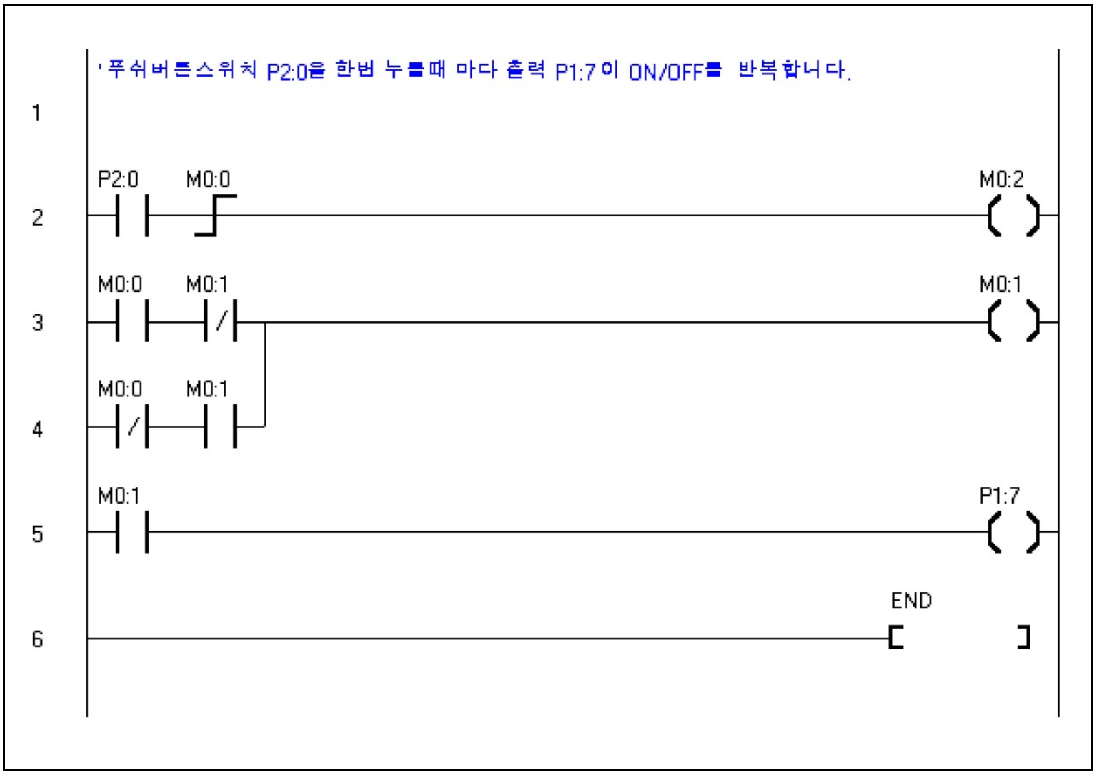
PLC 에 있는 카운트 명령 (CTU, CTD)는 스캔 타임마다 한번씩 펄스를 체크하므로, 스캔 타임보다 빠른 신호는 놓치는 현상이 발생합니다. 그래서 고속카운터가 필요한 것입니다. 고속 카운터는 스캔타임과 상관없이, 항상 입력되는 펄스를 카운트합니다. (PLC 내부에 별도의 하드웨어로 구성되어 있습니다.) PLC 에서는 단지 고속카운터가 카운트한 결과만을 참조하는 것입니다.

제 7 장
응용사례

TOGGLE : 스위치 입력때마다 출력을 반전

P2:0 에 연결되어 있는 스위치를 입력할 때마다 출력 P1:7 을 반전시키는 프로그램입니다.
다음은 사용된 포트의 입출력 설정표입니다.

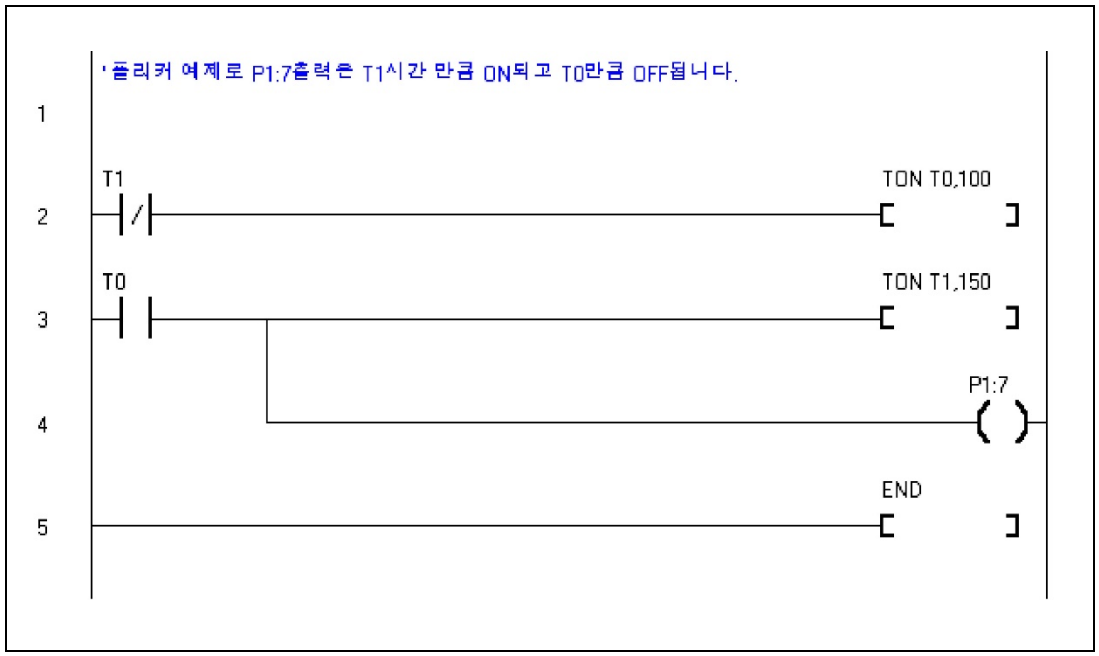
사용한 포트	입출력 설정
P2:0	INPUT
P1:7	OUTPUT



FLICKER : 타이머로 ON/OFF 반복 실행

타이머를 사용해서 일정한 시간 간격으로 ON/OFF 를 반복하는 프로그램 예제입니다.

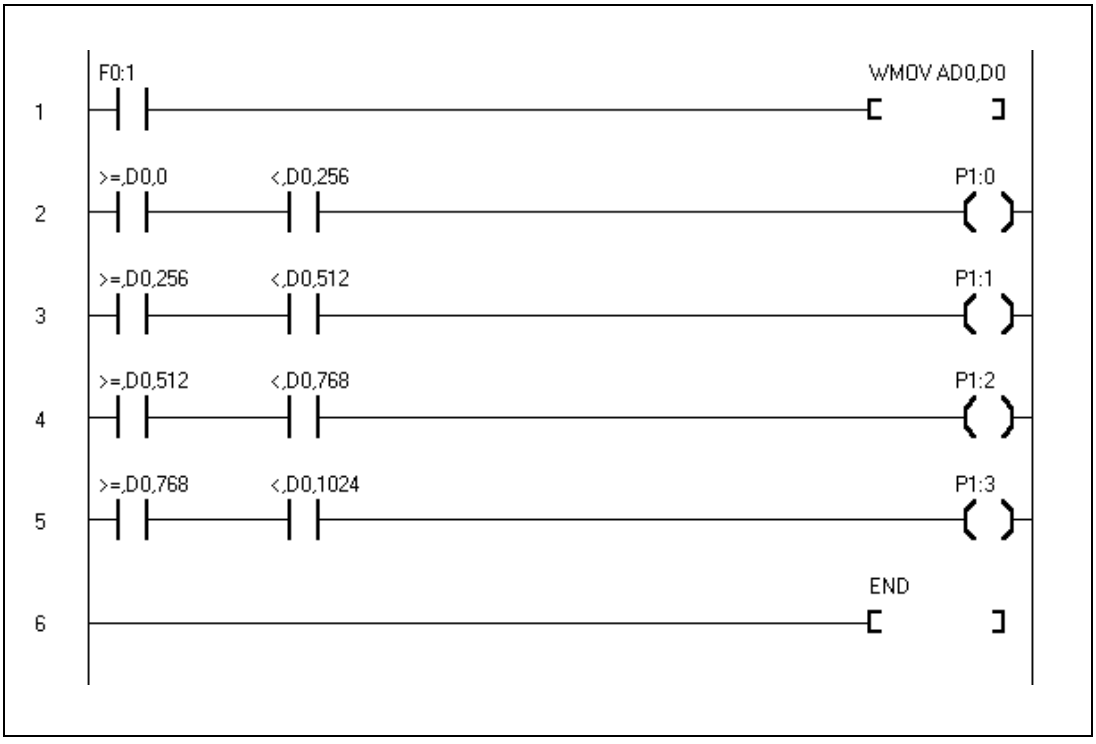
사용한 포트	입출력 설정
P1:7	OUTPUT



AD-STEPS : A/D 결과값에 따라 출력을 결정

AD0 포트에서 A/D 변환한 값의 크기에 따라 어떤 출력을 ON 할것인지를 결정하는 프로그램입니다. AD0 의 값이 256 보다 작으면 P1:0 이 출력되고, 512 보다 작으면 P1:1 이 출력됩니다.

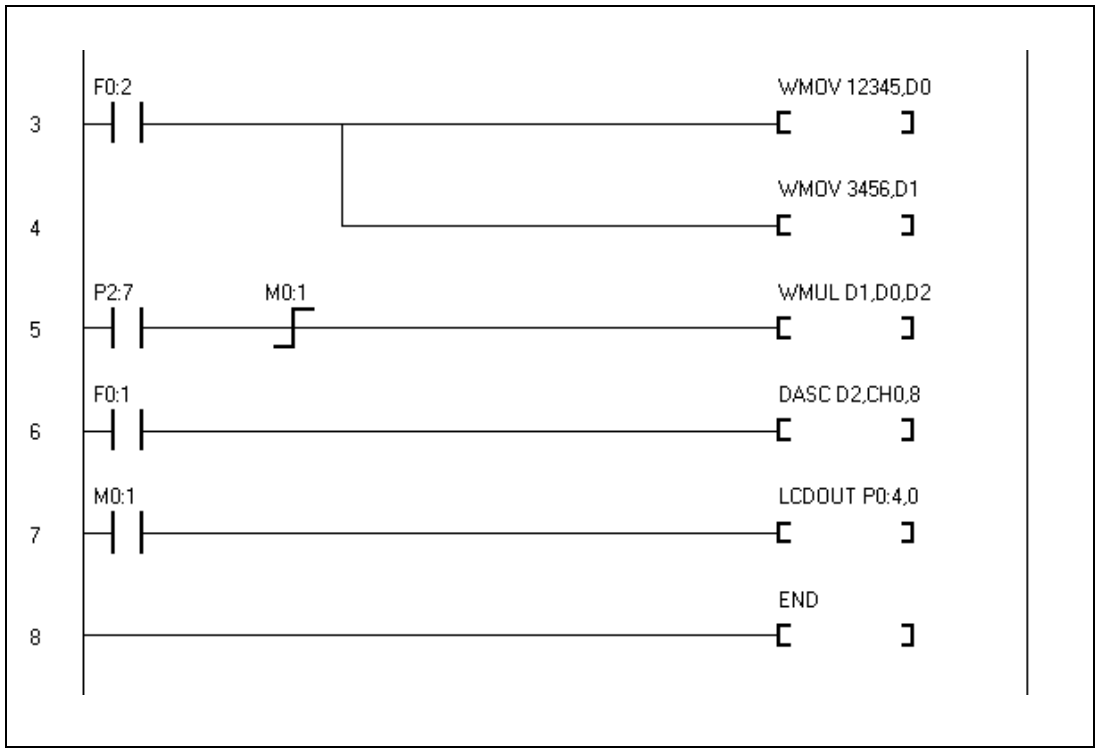
사용한 포트	입출력 설정
P1:0	OUTPUT
P1:1	OUTPUT
P1:2	OUTPUT
P1:3	OUTPUT
P3:0 (TPC37) P2:0 (TPC32)	INPUT



WMUL2 : 곱셈결과를 LCD 에 표시

두수를 곱하는 프로그램입니다. 12345 x 3456 = 42664320 이 되며, 결과는 LCD 에 표시됩니다. 초기 전원투입시에 D0 에 12345 가 저장되며, D1 에는 3456 이 저장됩니다. 연산결과는 D2 에 저장되며 LCD 에 표시하기 위해 DASC 명령을 사용했습니다. LCDOUT 입력단의 M0:1 은 입력스위치 P2:7 이 ON 되는 1 스캔동안만 ON 됩니다.

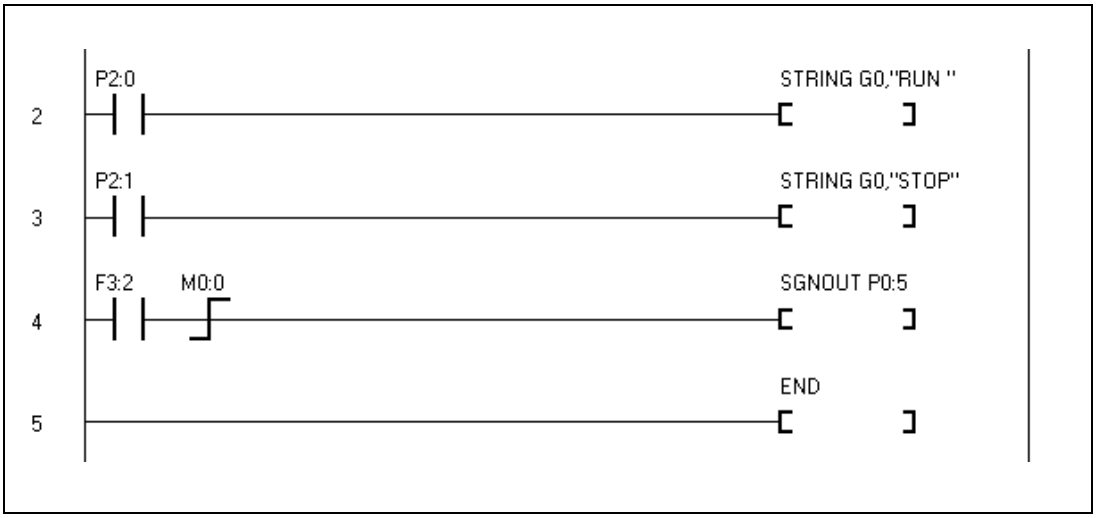
사용한 포트	입출력 설정
P2:7	INPUT
P0:4	OUTPUT



SGN-SW : 문자열을 SGN 에 표시

입력 P2:0 이 ON 되면 SGN 모듈에 "RUN"이 표시되고, P2:1 이 ON 되면 "STOP"이 표시됩니다. 표시할 문자열은 STRING 명령으로 미리 G 영역에 저장합니다.

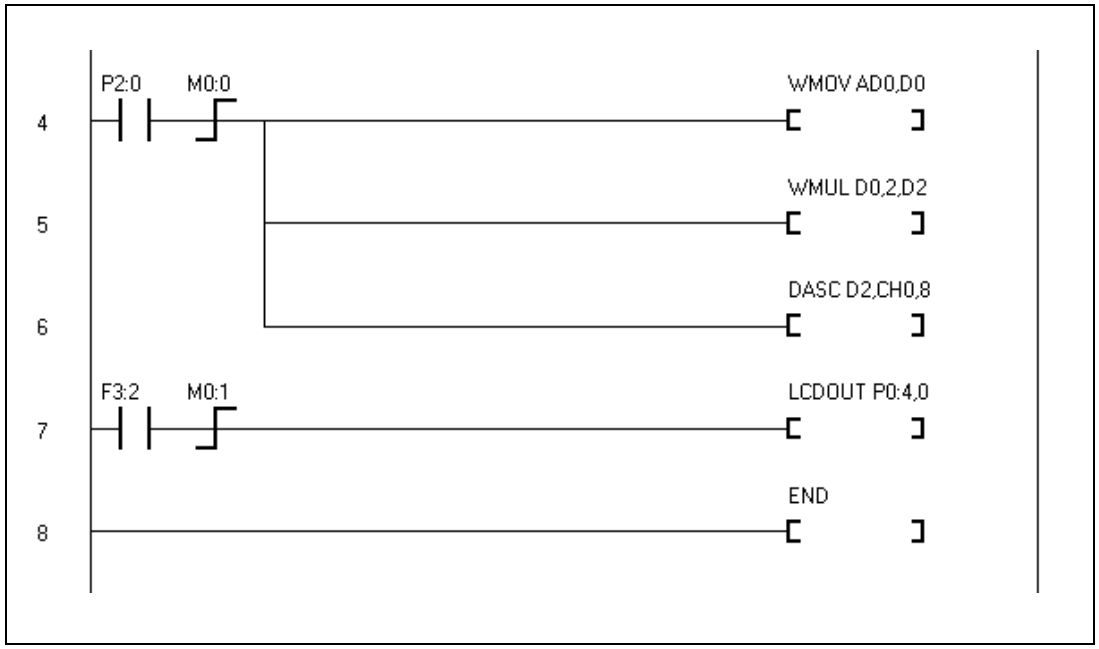
사용한 포트	입출력 설정
P2:0	INPUT
P0:5	OUTPUT



LCD-AD : AD 변환결과를 LCD 에 표시

입력 P2:0 이 ON 될때마다 AD0 의 값을 읽어서 2 를 곱하는 연산을 한 후 LCD 에 표시하는 프로그램입니다. WMUL 은 곱셈명령으로 결과는 D2 에 32BIT 형태로 저장됩니다. 32BIT 값을 10 진수로 바꾸기 위해서 DASC 명령을 사용했습니다. (LCD 모듈은 ELCD162 를 사용했습니다.)

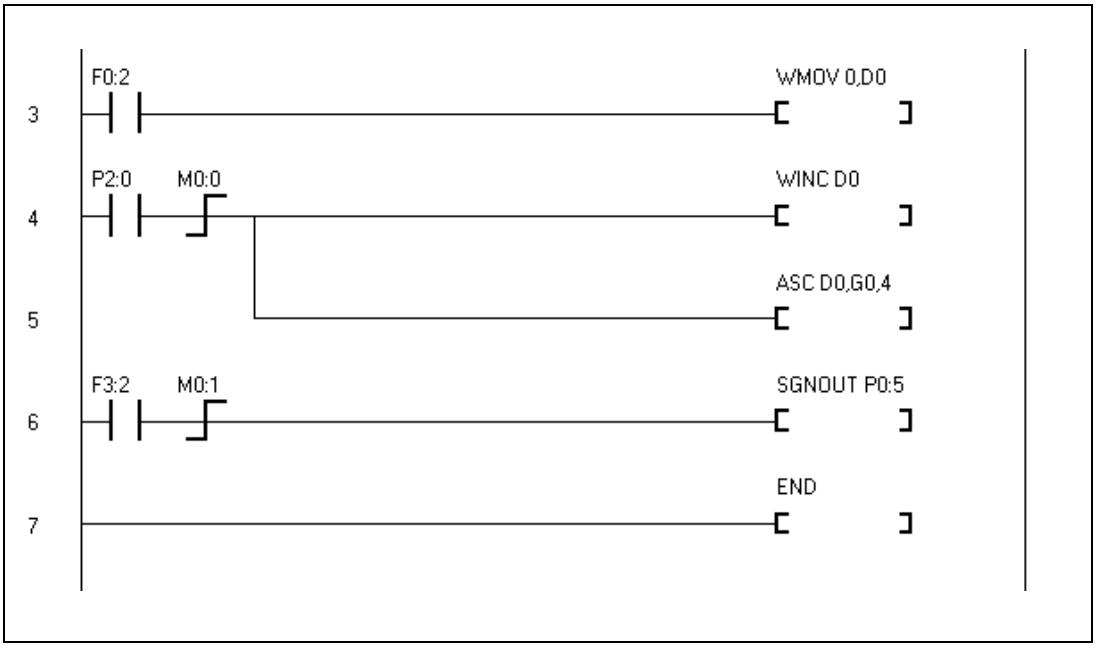
사용한 포트	입출력 설정
P2:0	INPUT
P0:4	OUTPUT
P3:0 (TPC37) P2:0 (TPC32)	INPUT



SGN-CNT : 카운트결과를 SGN 에 표시

입력 P2:0 이 ON 될때마다 16 비트 변수 D0 의 값이 1 씩 증가됩니다. (카운트의 효과) 이 D0 의 값을 10 진형태의 ASCII 코드로 변환해서 SGN 모듈에 표시하는 프로그램입니다.

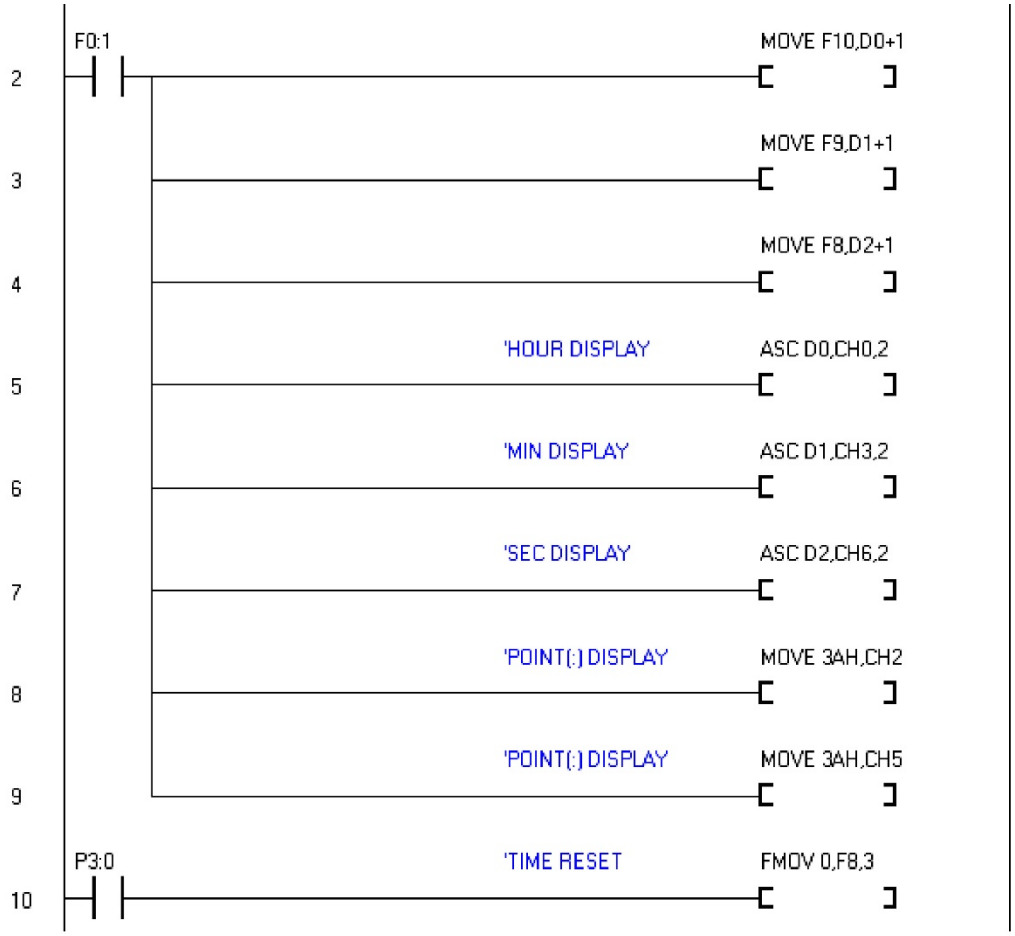
사용한 포트	입출력 설정
P2:0	INPUT
P0:5	OUTPUT



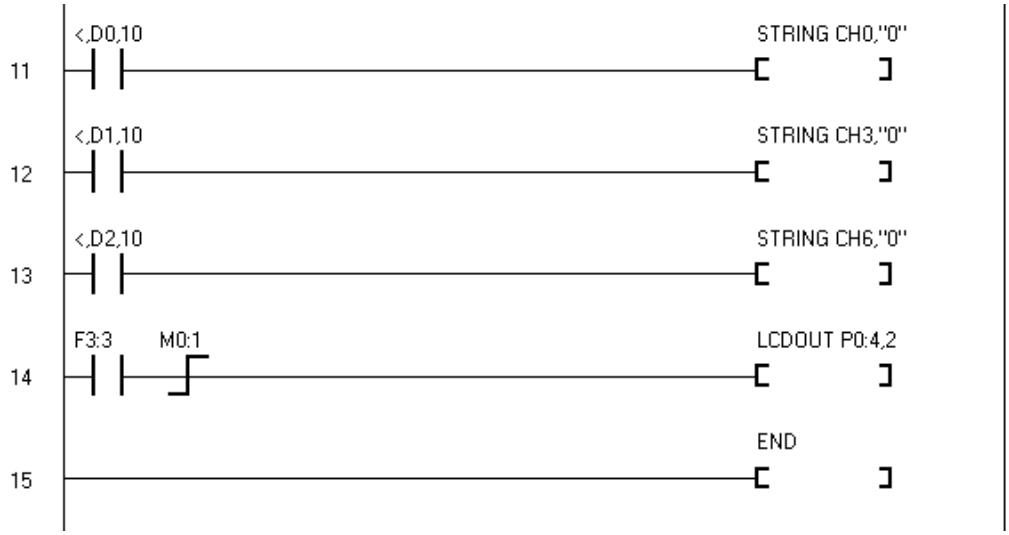
CLOCK: 디지털 시계

LCD 상에 실제 시간을 표시해 주는 레더 프로그램입니다. P1:0 에 입력신호가 들어오면 RESET 되어 12:00:00 이 되며, 타임은 TPC37 (또는 TPC38)에 내장된 리얼타임 클록 F8, F9, F10 을 이용하여 동작합니다.

사용한 포트	입출력 설정
P1:0	INPUT
P0:4	OUTPUT



계속…


<,D0,10
STRING CH0,"0"
11
<,D1,10
STRING CH3,"0"
12
<,D2,10
STRING CH6,"0"
13
F3:3
M0:1
LCDOUT P0:4,2
14
END
15

<끝>