

MANUAL

Magical PLC

Contents

- Intro
- History
- UI Screen
 - 상단 화면버튼
 - 모니터링 : XY-인터페이스 화면
 - 모니터링 : D-메모리 화면
- Programming
 - 프로그램 작성방법
 - 디바이스 사용방법
 - 평선블록 사용방법
- Example

Intro

- 유형 : 웹 기반 소프트웨어
- 목적 : PLC 개념 차용한 프로그램 코딩
- 디바이스 소개

디바이스	내용	개수
X	외부 입력비트	16
Y	외부 출력비트	16
D	워드단위	100
T	타이머비트	20
P	펄스비트	100
L	래치비트	100

UI Screen

- 상단 화면버튼

모니터링 :

- XY-인터페이스

XY-External Address

X입력	X상태	Y출력	Y상태
X0	Off	Y0	Off
X1	Off	Y1	Off
X2	Off	Y2	Off
X3	Off	Y3	Off
X4	Off	Y4	Off
X5	Off	Y5	Off
X6	Off	Y6	Off
X7	Off	Y7	Off
X8	Off	Y8	Off
X9	Off	Y9	Off
X10	Off	Y10	Off
X11	Off	Y11	Off
X12	Off	Y12	Off
X13	Off	Y13	Off
X14	Off	Y14	Off
X15	Off	Y15	Off

UI Screen

- D-메모리 화면

D-Memory Address

[illegible]

표 내의 칸을 클릭하면, 값을 입력할 수 있습니다.
숫자만 입력해야 합니다.

※ 보조: WORD - BIT 변환검산기능

* Word Calculator

[illegible]

Programming

- 프로그램 작성방법

1. Scan 폴더 안에서, Program 텍스트파일 열기



2. 아래의 { 와 } 사이에서 프로그램 내용을 작성
(작성요령은 다음 장 사용방법을 통해 학습)

```
function ScanProgram()
{
/*
    프로그램 작성영역
*/
};
```

※ 주석 사용방법

1. // : 이후의 우측 내용 주석
2. /* [내용] */ : 사이의 내용 주석

Programming

- 디바이스 사용방법

1. X, Y 디바이스

가장 기초적이고 단순한 디바이스
외부장치 인터페이스에 활용(예정)

※ 다음과 같이 식을 세워서 확인할 수 있다.

```
// X, Y 사용
```

```
Y[0] = X[0];
```

<Ladder 비교>



Programming

- 디바이스 사용방법

- 2. D 디바이스

워드(16비트)로 표현하는 디바이스
평션블록을 통해서 비트로 활용하거나
워드연산에 활용할 수 있다.

사례1) D디바이스의 비트조작

```
// D 사용

D[0] = SetBIT(D[0], 0, X[0]);
D[0] = SetBIT(D[0], 1, X[1]);

Y[0] =
AND([
    GetBIT(D[0],0),
    GetBIT(D[0],1)
]);
```

※ **D[x]**=**SetBIT**(**D[x]**, y, z)
-> Dx 디바이스의 y번 비트에 z 대입

※ **GetBIT**(D[x], y)
-> Dx 디바이스의 y번 비트값 반환

Programming

- 디바이스 사용방법

- 2. D 디바이스

사례2) D디바이스의 비교연산

```
// D 사용(비교연산)
Y[0] = GetCOMP("=", D[0], D[1]);
Y[1] = GetCOMP("<>", D[0], D[1]);
Y[2] = GetCOMP(">", D[0], D[1]);
Y[3] = GetCOMP("<", D[0], D[1]);
Y[4] = GetCOMP(">=", D[0], D[1]);
Y[5] = GetCOMP("<=", D[0], D[1]);
```

※ **GetCOMP**(부등식, 비교1, 비교2)

-> 비교1 부등식 비교2 관계에 따라
맞으면 On, 틀리면 Off 반환

D-Memory Address			
D000	+0	+1	
+00	0	0	0

두 값을 입력하면서
Y출력 확인

Programming

- 디바이스 사용방법

- 3. T 디바이스

타이머 사용을 위한 디바이스

일정 시간 후에 상태변화하는 것이 특징

```
//T 사용
T[0].SetTIMER(100, X[0]);

Y[0]=T[0].Stop;
Y[1]=T[0].Busy;
Y[2]=T[0].a;
Y[3]=T[0].b;
```

※ **SetTIMER**(설정시간, z)

-> z가 On인 동안, 타이머 작동

설정시간 경과 후, 타이머출력 활성화

※ Stop : 타이머가 비활성화일 때

※ Busy : 타이머가 활성화 중일 때

※ a : 타이머시간 경과 후 On되는 비트

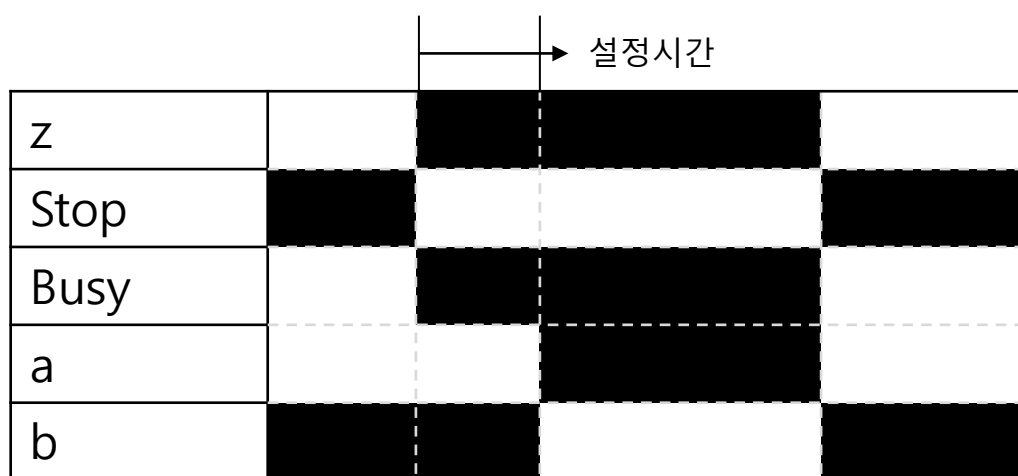
※ b : 타이머시간 경과 후 Off되는 비트

Programming

- 디바이스 사용방법

- 3. T 디바이스

- 타이밍 차트



Programming

- 디바이스 사용방법

- 4. P 디바이스

펄스입력을 위한 디바이스

상태변화를 감지하는 것이 특징

```
//P사용
P[0].o = X[0];
P[1].o = X[1];

Y[0] =
AND([
    OR([
        P[0].Ua, Y[0]
    ]),
    P[1].Ub
]);
```

※ o : 펄스동작에 활용할 조건식 대입

※ a : a접점

※ b : b접점

※ Ua : Off에서 On일 때, 1스캔 On

※ Ub : Off에서 On일 때, 1스캔 Off

※ Da : On에서 Off일 때, 1스캔 On

※ Db : On에서 Off일 때, 1스캔 Off

Programming

- 디바이스 사용방법

4. P 디바이스

타이밍 차트

		1스캔		1스캔	
o					
a					
b					
Ua					
Ub					
Da					
Db					

Programming

- 디바이스 사용방법

- 5. L 디바이스

래치동작을 위한 디바이스

기존 입력을 취소해도 값을 유지하는 것이 특징

```
//L사용
L[0].st = X[0];
L[0].rs = X[1];

Y[0] = L[0].a;
Y[1] = L[0].b;
```

- ※ st : Set을 위한 조건식
- ※ rs : Reset을 위한 조건식
- ※ a : Set 상태
- ※ b : Reset 상태

Programming

- 디바이스 사용방법

- 5. L 디바이스
타이밍 차트

st						
rs						
a						
b						

Programming

- 평션블록 사용방법

1. 논리연산

AND([조건1, 조건2, ...]);
OR([조건1, 조건2, ...]);

조건들끼리 모두 AND 또는 OR관계를 가진다.

NOT(조건);

조건의 결과를 반전시킨다.

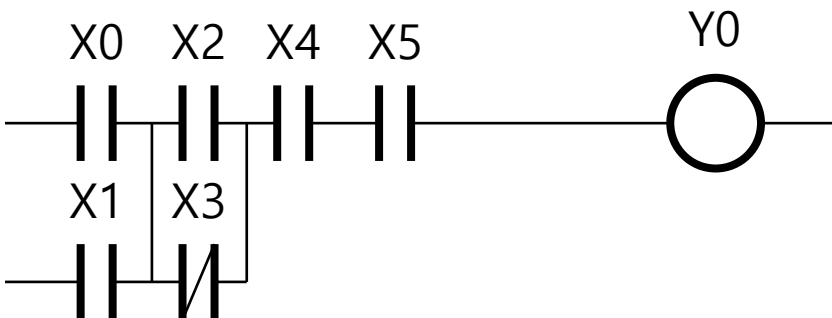
Programming

- 펄스블록 사용방법

※ 펄스 내의 또 다른 펄스를 사용
조합하여 조건식 사용가능

```
//AND, OR, NOT 조합  
Y[0] =  
AND([  
    OR([  
        X[0],  
        X[1],  
    ]),  
    OR([  
        X[2],  
        NOT(X[3])  
    ]),  
    X[4],  
    X[5]  
]);
```

<Ladder 비교>



Programming

- 평선블록 사용방법

3. SetMOV("키워드", 원본, 대상, 조건식)

SetMOV("키워드", 원본, 대상, 조건식);

원본의 데이터 값을 대상에 대입하는 기능
조건식 값이 On이면, 기능이 활성화된다.

키워드?

- "D"를 넣으면

- > 원본의 D번호 값을 대상의 D주소에 대입한다.

- "" 또는 null을 넣으면

- > 원본의 값을 대상의 D주소에 대입한다.

//MOV 사용

```
SetMOV('D', 10, 11, X[0]);
```

```
SetMOV("", 20, 12, X[1]);
```

Programming

- 평선블록 사용방법

3. SetBMOV("키워드", 원본, 대상, 조건식)

SetBMOV("키워드", 원본, 대상, 개수, 조건식);

원본의 데이터 값을 대상에 지정개수만큼
대입하는 기능, 조건식 값이 On이면,
기능이 활성화된다.

키워드?

- "D"를 넣으면

- > 원본의 D번호 값을 대상의 D주소에 대입한다.

- "" 또는 null을 넣으면

- > 원본의 값을 대상의 D주소에 대입한다.

```
//BMOV 사용
```

```
SetBMOV('D', 30, 40, 5, X[2]); //D30~D34의 데이터를 D40~D44에 각각 대입
```

```
SetBMOV(null, 33, 40, 5, X[3]); //값30을 D40~D44에 동일하게 대입
```

Example

- 종합예제1