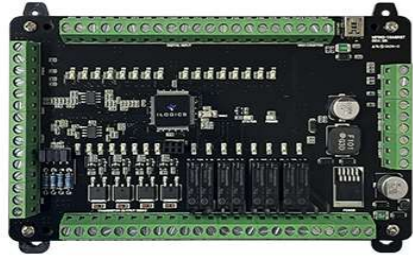


MPINO SERIES MPINO-16A8R8T

사용 설명서

저희 (주)아이로직스 제품을 구입해 주셔서 감사합니다.



사용 전에 안전을 위한 주의사항을 반드시 읽고 사용하십시오.

안전을 위한 주의사항

- ※ '안전을 위한 주의사항'은 제품을 안전하고 올바르게 사용하여 사고나 위험을 미리 막기 위한 것이므로 반드시 지켜야 합니다.
 - ※ 주의사항은 '경고'와 '주의' 두 가지로 구분되어 있으며, '경고'와 '주의'의 의미는 다음과 같습니다.
- 지시사항을 위반하였을 때.
- 경고** 심각한 상해나 사망이 발생할 가능성이 있는 경우
 - 주의** 경미한 상해나 제품 손상이 발생할 가능성이 있는 경우
- ※ 제품과 취급설명서에 표시된 그림기호의 의미는 다음과 같습니다.
- ⚠는 특정조건 하에서 위험이 발생할 우려가 있으므로 주의하라는 기호입니다.

경고

- 인명이나 재산상에 영향이 큰 기기(예: 원자력 제어장치, 의료기기, 선박, 차량, 철도, 항공기, 연소장치, 안전장치, 방범/방재장치 등)에 사용할 경우에는 반드시 2중으로 안전장치를 부착한 후 사용해야 합니다. 화재, 인사사고, 재산상의 막대한 손실이 발생할 수 있습니다.
- 자사 수리 기술자 이외에는 제품을 개조하지 마십시오. 감전이나 화재의 우려가 있습니다.

주의

- 실외에서 사용하지 마십시오. 제품의 수명이 짧아지는 원인이 되며 감전의 우려가 있습니다. 본 제품은 실내 환경에 적합하도록 제작되었습니다. 실내가 아닌 외부환경 으로부터 영향을 받을 수 있는 장소에서 사용할 수 없습니다. (예 : 비, 황사, 먼지, 서리, 햇빛, 결로 등)
- 인화성, 폭발성 가스 환경에서 사용하지 마십시오. 화재 및 폭발의 우려가 있습니다.
- 사용 전압 범위를 초과하여 사용하지 마십시오. 제품이 파손될 수 있습니다.
- 전원의 극성 등 오배선을 하지 마십시오. 제품이 파손될 수 있습니다.
- 진동이나 충격이 많은 곳에서 사용하지 마십시오. 제품이 파손될 수 있습니다.
- 청소 시 물, 유기 용제를 사용하지 마십시오. 감전 및 화재의 우려가 있습니다.

손해배상책임

(주)아이로직스는 제품을 사용하다 발생하는 인적, 물적자원에 대해 책임을 지지 않습니다. 충분한 테스트와 안전장치를 사용하여 주시기 바랍니다.

사양서

구분	개수	접점명	설명
전원	-	전원전압	• DC 12 ~ 24V (24V 0.5A 이상)
디지털 입력	8 포인트 (절연) (양방향)	I(22) ~ I(29) / COM0	• 오퍼레이팅 입력 전압 : DC 0 ~ 40V • HIGH 인식 전압 : DC 5V 이상 • 입력저항 : 2.2kΩ
		I(30) ~ I(37) / COM1	
트랜지스터 출력 (Sink)	8포인트 (비절연)	O(39) ~ O(46)	• 오퍼레이팅 출력 전압 - 0 ~ 55V D.C • 최대 출력 전류 - 3A / 1POINT, 15A / 1COM • ON : GND와 연결 • OFF : Floating
릴레이 출력	8 포인트 (절연)	R(62) ~ R(69)	• 오퍼레이팅 연결 전압 - 0 ~ 30V D.C, 0 ~ 250V A.C • 최대 출력 허용전류 : 5A / 1POINT • R(62), R(63) / COM3 • R(64), R(65) / COM4 • R(66), R(67) / COM5 • R(68), R(69) / COM6 • ON : COM과 연결 • OFF : Floating
아날로그 입력	4 포인트 (비절연)	A(0) ~ A(3)	• 0 ~ 20mA (Default), 4 ~ 20mA • DC 0 ~ 5V (점퍼핀 해제) • DC 0 ~ 10V (옵션, 점퍼핀 해제) • 분해능 : 10Bit (0~1023) • 입력저항 : 2kΩ (0~5V 전압입력)
온도센서 입력	2 포인트 (비절연)	T(4), T(5)	• 입력저항 : 10kΩ (Pull-up) • 분해능 : 10Bit (0~1023)
아날로그 출력	2 포인트 (비절연)	AO(6) ~ AO(7) / TIMER4	• 출력 전압 : 0 ~ 5V D.C • 분해능: 16Bit (0~65535)
고속펄스 입력	2 포인트 (절연)	TCNT4, TCNT5	• 오퍼레이팅 입력 전압 : DC 0 ~ 80V • HIGH 인식 전압 : DC 5V 이상 • 최대입력 주파수 : 5kHz • TCNT4 : TIMER4 자원사용 • TCNT5 : TIMER5 자원사용
펄스 출력	4 포인트 (비절연)	P(11),P(12) / TIMER1	• 오퍼레이팅 출력 전압 - LOW (0V D.C), HIGH (5V D.C) • 오퍼레이팅 최대 출력 전류 : 30mA • 출력 저항: 150Ω (쇼트 보호저항) • 분해능: 16Bit (0~65535)
		P(5),P(2) / TIMER3	• P(11),P(12) : TIMER1 자원사용 • P(5), P(2) : TIMER3 자원사용
인터럽트	1 포인트 (비절연)	P(2)	• 오퍼레이팅 입력 전압 : DC 0~5V • HIGH 인식 전압 : DC 3V 이상
통신 채널	4 채널 (비절연)	I2C, RS232, RS485, UART	• I2C 1채널 지원 • RS232 1채널 지원 (Serial1) • RS485 1채널 지원 (Serial2) • UART 1채널 지원 (Serial3)

메모리 사양서

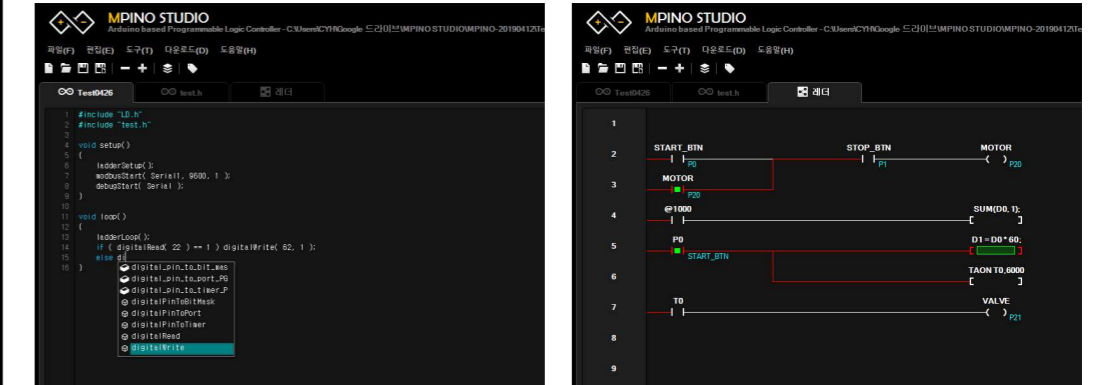
- 200Kbyte Flash Memory
- 7Kbyte Data Memory
- 4Kbyte EEPROM Memory (비휘발성 메모리)

MPINO STUDIO 사용방법 [요약]

① MPINO STUDIO 설치

• 아이로직스 쇼핑몰 자료실 에서 "MPINO STUDIO"를 다운로드 받고, 설치합니다. (<https://www.ilogics.co.kr/article/자료실/7/24/> =)

• 아래는 "MPINO STUDIO"의 화면입니다. 자세한 사용법은 MPINO STUDIO 사용설명서를 참조 바랍니다. (<https://www.ilogics.co.kr/article/자료실/7/22/>)



[아두이노 C언어]

[LADDER LOGIC]

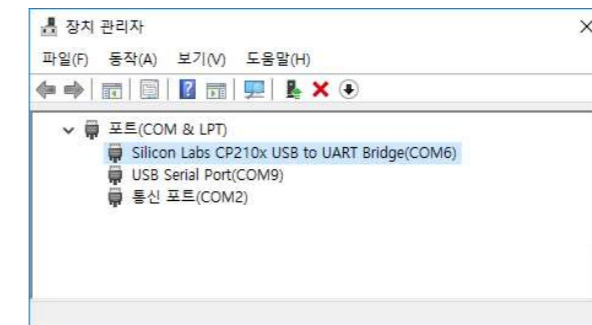
② 컴퓨터의 USB포트와 제품(MPINO-16A8R8T)에 "MP 다운로드 케이블"을 연결합니다.

③ 만약, 드라이버가 나타나지 않는다면 (주)아이로직스 홈페이지의 자료실에서 "다운로드 케이블 드라이버" 게시물에서 FTDI 드라이버를 다운로드 받아 설치합니다. (<https://www.ilogics.co.kr/article/자료실/7/18/>)

④ "MPINO STUDIO"를 실행하고 도구창에서 "도구 -> 디바이스 -> 아이로직스-> MPINO-16A8R8T" 를 선택합니다.

⑤ 다운로드 포트를 확인합니다 설정합니다.

• 윈도우의 장치관리자에서 아래 그림처럼 선택된 COM포트를 확인합니다.



⑥ MPINO STUDIO의 도구창에서 "도구 -> 포트"의 COM포트 목록중에 위에서 확인한 다운로드 COM포트를 선택합니다.

⑦ 프로그램을 작성하고 다운로드 합니다.

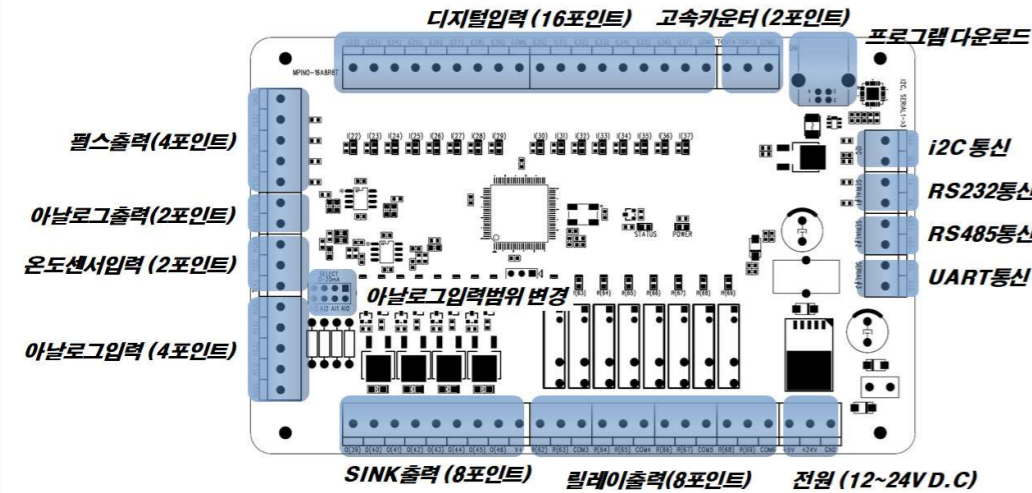
명령어 설명서

Arduino IDE에서 도움말 -> 참조를 실행하거나 다음 링크에서 확인할 수 있습니다.
(<https://www.arduino.cc/reference/en/>)

EEPROM과 I2C(Wire) 등을 보다 쉽게 사용할 수 있는 라이브러리는 다음 링크에서 확인할 수 있습니다.

(<https://www.arduino.cc/en/Reference/Libraries>)

기능별 위치

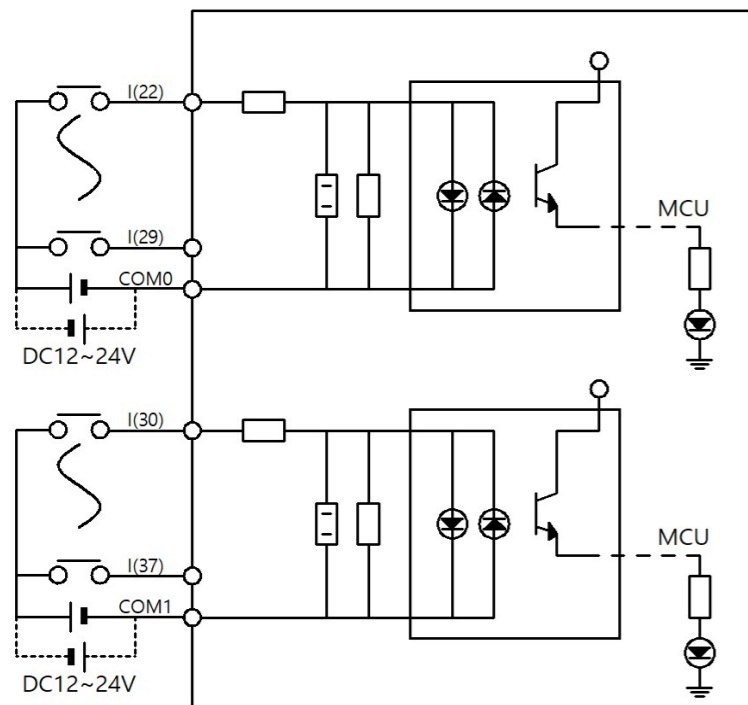


전원

- "+24V" 단자와 "GND" 단자에 24V D.C를 연결하여 사용할 수 있습니다.
- 24V D.C를 전원전압으로 사용할 경우, "+5V" 단자에는 5V D.C, 1A 출력으로 사용할 수 있습니다.
- "+5V" 단자와 "GND" 단자에 5V D.C를 연결하여 사용할 수 있습니다. 단, +5V 단자는 +24V에서처럼 필터 및 레귤레이터의 여과 없이 MCU 및 I.C 등의 전원으로 사용되므로 안정적인 전원공급이 요구됩니다. (배터리 연결 등)

디지털 입력

I(22) ~ I(37) 포트에 +5V ~ 24V D.C의 전압이 입력되었을 때, 포트의 메모리 상태가 High가 됩니다.



관련 명령어 (디지털 입력/출력)

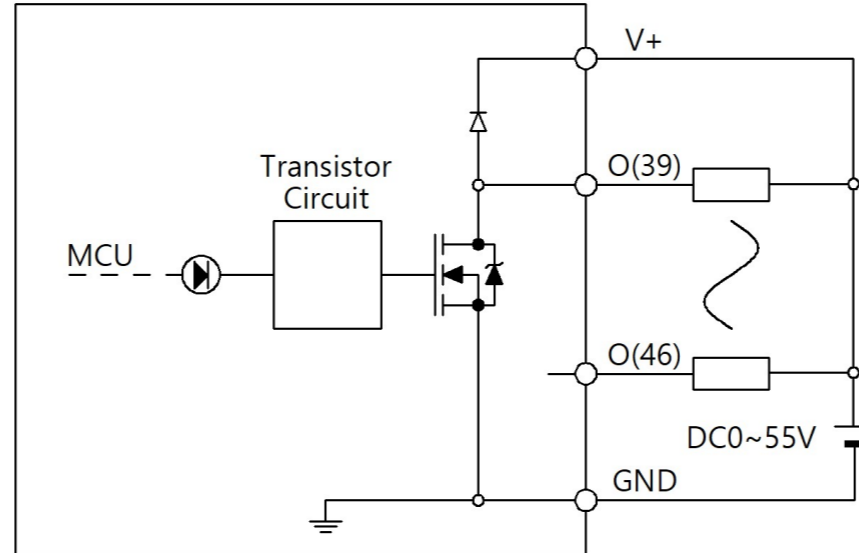
`pinMode(Pin, INPUT/OUTPUT)` Pin포트를 INPUT 또는 OUTPUT으로 설정.

`digitalRead(Pin)` Pin포트의 입력상태를 "0" 또는 "1"로 반환.

`digitalWrite(pin, 0/1)` pin포트의 출력상태를 LOW 또는 HIGH로 변환.

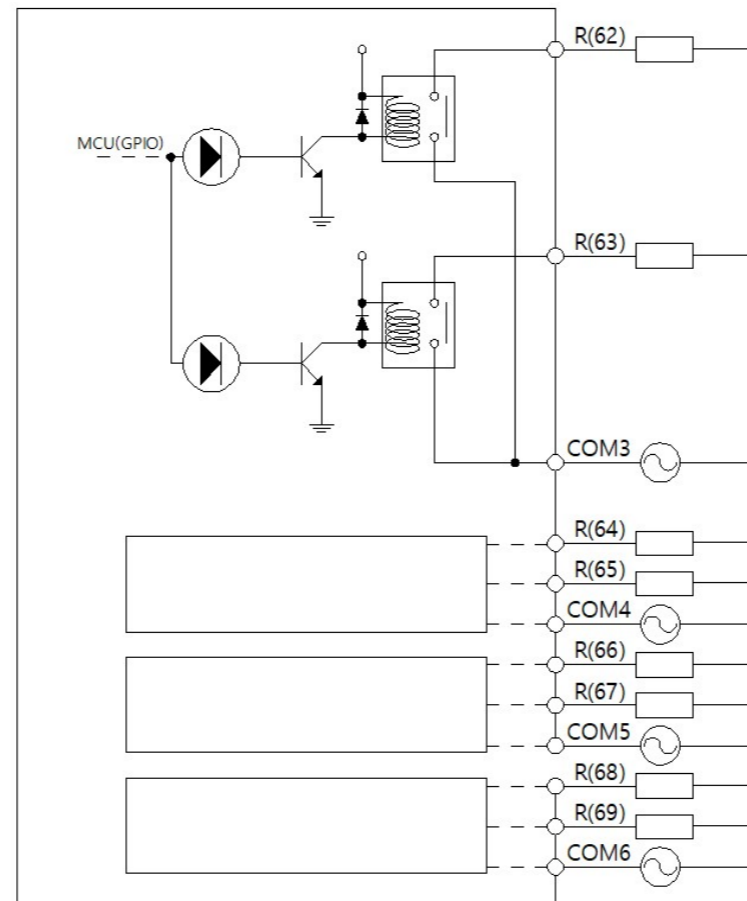
트랜지스터 출력(Sink)

O(39) ~ O(46)의 메모리 상태가 ON될 때, 각각의 트랜지스터 출력포트에 GND를 출력합니다.



릴레이 출력

R(62) ~ R(69)의 메모리 상태가 On될 때, 각각의 릴레이 접점이 COM과 물리적으로 연결되는 상태가 됩니다.



디지털 입력 및 릴레이 출력 프로그램 간단 예제

MPINO STUDIO : C코드를 사용할 경우

```
#include "LD.h" // Ladder 명령어를 사용하기 위한 함수참조

void setup() {
    ladderSetup(); // Ladder를 실행하기 위한 설정함수
}

void loop() {
    // 22번 디지털 입력이 ON되면, 릴레이 출력 62번을 ON
    if (digitalRead(22) == 1) digitalWrite(62, HIGH);
    // 그렇지 않다면, 릴레이 출력 62번을 OFF
    else digitalWrite(62, LOW);
    ladderLoop();
}
```

MPINO STUDIO : LADDER LOGIC을 사용할 경우

하드웨어 설정에서 I(22)를 P0로 바인딩하고, O(39)를 P20으로 바인딩합니다. 디지털입/출력은 Px 형식으로 사용자가 바인딩할 수 있습니다.

MP 하드웨어 설정

명칭	핀 번호	핀 유형	바인딩
I(22)	22	INPUT	P0
I(23)	23	INPUT	P1
I(24)	24	INPUT	P2
I(25)	25	INPUT	P3
I(26)	26	INPUT	P4
I(27)	27	INPUT	P5
I(28)	28	INPUT	P6
I(29)	29	INPUT	P7
I(30)	30	INPUT	P8
I(31)	31	INPUT	P9
I(32)	32	INPUT	P10
I(33)	33	INPUT	P11
I(34)	34	INPUT	P12
I(35)	35	INPUT	P13
I(36)	36	INPUT	P14
I(37)	37	INPUT	P15
O(39)	39	OUTPUT	P20

확인 취소

디지털 입력 I(22):P0 포트에 전압이 입력되면 I(22):P0 접점이 On되고, O(39):P20 출력접점이 On되어 O(39):P20 트랜지스터 출력이 On됩니다.



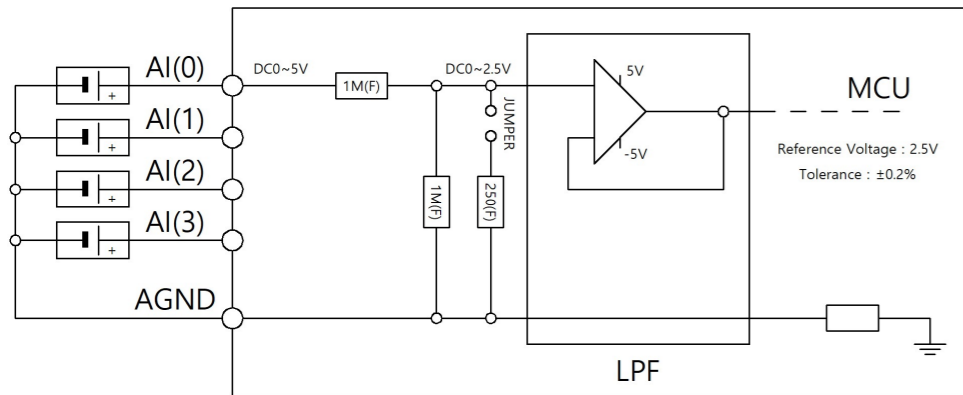
□ 상태 LED

LED_BUILTIN 변수명 또는 D13핀으로 제품에 삽입되어 있는 STATUS LED를 ON/OFF 시킬 수 있습니다.

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT); //LED_BUILTIN을 출력모드로 설정
}
void loop() {
  // I(22)가 HIGH 이면, LED_BUILTIN를 ON 시킵니다.
  if (digitalRead(22) == 1) { digitalWrite(LED_BUILTIN, HIGH); }
  // I(22)가 LOW 이면, LED_BUILTIN를 OFF 시킵니다.
  else { digitalWrite(LED_BUILTIN, LOW); }
}
```

□ 아날로그 입력

아날로그 입력포트 A(0) ~ A(3)에 입력된 아날로그 신호를 analogRead(pin) 명령어를 사용하여 디지털 값으로 변환하여 사용합니다.
아날로그 및 온도센서 입력의 정확도를 위해서 analog Reference 핀에 ±0.2% 오차율 Reference Regulator I.C를 사용하여 레퍼런스 전압을 공급하고 있습니다.

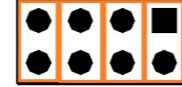


관련 명령어

- analogReference(EXTERNAL)** : 아날로그 입력의 기준전압을 MCU의 Vref 핀에 연결된 전압으로 설정합니다.
- analogRead(Pin)** : Pin 포트의 아날로그 신호를 디지털값으로 변환합니다.

□ 아날로그 입력 범위

SELECT
0-20mA



AI3 AI2 AI1 AI0 < 제품 왼쪽 하단에서 전압/전류를 선택할 수 있습니다 >

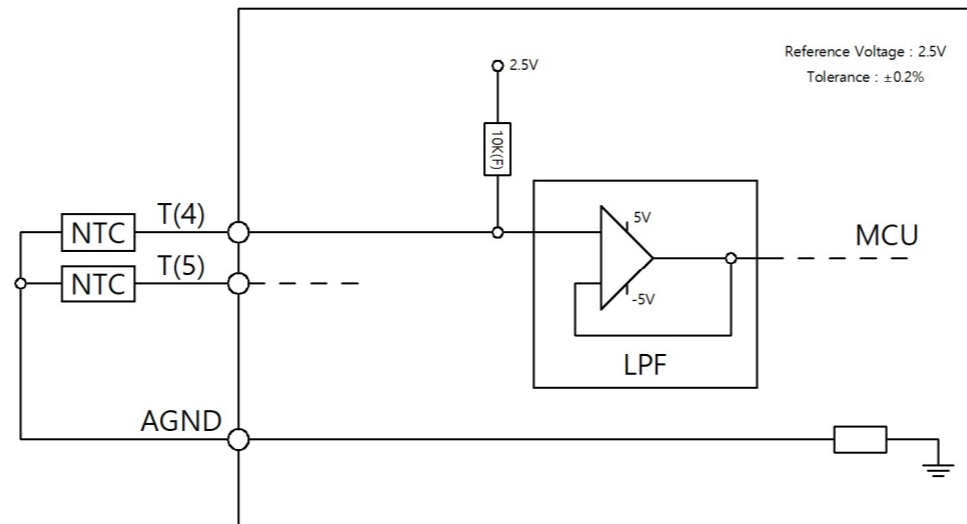
아날로그 입력 범위는 제품출하시 “SELET 0-20mA” 핀헤더에 헤더캡을 씌워지므로 0~20mA 전류입력범위로 설정되어 있습니다. 0~20mA 전류입력모드 이지만 DC 0~5V 입력도 가능합니다. 단, 부하저항이 250Ω으로 매우 작습니다. 때문에, 상대방 기기의 아날로그 출력 사양에 따라서 동작이 어려울 수 있습니다.

아날로그 입력 범위를 DC 0~5V로 사용하실려면 “SELECT 0-20mA” 핀헤더에서 사용하고자 하는 아날로그 입력포트 번호의 헤더캡을 제거하고 사용해주시기 바랍니다.

아날로그 입력 범위를 DC 0~10V로 사용하실려면 제품 주문시 옵션을 선택하여 주문해 주시면 DC 0~10V로 입력 범위를 변경하여 출고해 드립니다.

□ 온도센서 입력

10kΩ(@25℃) NTC 써미스터를 연결하여 사용할 수 있습니다. 10kΩ 풀업 저항이 내장되어 있으므로 T(4) 또는 T(5)와 AGND에 NTC의 두 선을 연결하여 사용합니다.

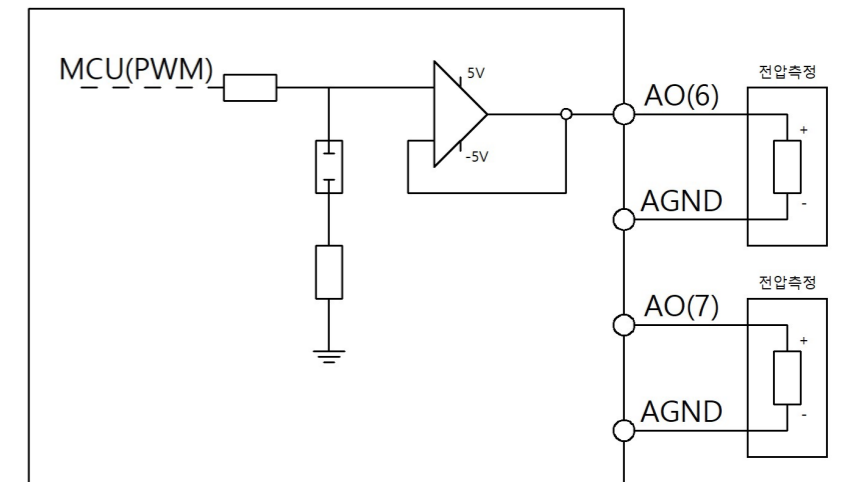


관련 명령어

- NTEMP(Pin)** : 온도센서의 온도값을 반환합니다. (Pin : 4 또는 5)
- analogReference(EXTERNAL)** : 아날로그 입력의 기준전압을 MCU의 Vref핀에 연결된 전압으로 설정합니다.
- analogRead(Pin)** : Pin 포트의 아날로그 신호를 디지털값으로 변환합니다.

□ 아날로그 출력

아날로그 출력포트 AO(6), AO(7)에 0 ~ 5V D.C를 출력합니다.



관련 명령어

- analogWrite(Pin, Value)**: Pin포트에 Value값만큼 아날로그 출력을 합니다.
 - Pin : 6, 7
 - Value : 0 ~ 255(Default) 또는 0 ~ 65535 (aoRange로 최대값 변경)
- aoRange(Value)**: analogWrite 함수의 Value의 Max값을 Value값으로 변경합니다. (Value : 0 ~ 65535)

□ 아날로그 입력 및 출력, 온도센서 입력 프로그램 간단 예제

MPINO STUDIO : C코드를 사용할 경우

```
#include "LD.h" // LADDER 명령어를 사용하기 위한 함수참조
```

```
int ADC0 = 0; // ADC0 메모리를 int로 정의
int Temp = 0;
```

```
void setup() {
  ladderSetup(); // Ladder를 실행하기 위한 설정함수
}
```

```
void loop() {
  ADC0 = analogRead(0); //AI(0)의 0~5V를 디지털값으로 변환하여 저장

  Temp = NTEMP(4); T(4)에 연결된 NTC 온도센서의 값을 Temp 변수에 저장.
```

```
  analogWrite(6, 327670); // AO(6)에 2.5V를 출력합니다.
  출력전압 계산식은 5V * (Value(32767)/65535)입니다.
```

```
}
```

□ 아날로그 출력 리플감소

아날로그 출력의 리플이 클 경우, aoRange(Value) 명령어의 Value값을 감소시켜서 리플을 감소시킬 수 있습니다.

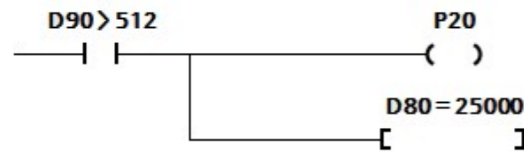
aoRange(10000); 명령어를 실행할 경우 analogWrite(pin, value) 명령어의 value는 0 ~ 10000까지 사용할 수 있습니다.

MPINO STUDIO : LADDER LOGIC을 사용할 경우
아날로그 입/출력은 Dx 형식으로 사용자가 바인딩할 수 있습니다.

명칭	핀 번호	핀 유형	바인딩
R(67)	67	OUTPUT	P33
R(68)	68	OUTPUT	P34
R(69)	69	OUTPUT	P35
AI(0)	0	ADIN	D90
AI(1)	1	ADIN	D91
AI(2)	2	ADIN	D92
AI(3)	3	ADIN	D93
T(4)	4	ADIN	D94
T(5)	5	ADIN	D95
AO(6)	6	PWM	D80
AO(7)	7	PWM	D81
P(11)	11	PWM	D70
P(12)	12	PWM	D71
P(5)	5	PWM	D72
P(2)	2	PWM	D73
TCNT4	4	HSC	
TCNT5	5	HSC	D76

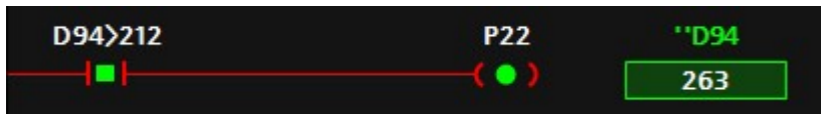
⚠ 아날로그 출력을 사용할 경우, TCNT4의 바인딩을 하지 말아야 합니다.

하드웨어 설정만으로 D 메모리에 아날로그 값이 자동으로 저장되고, D 메모리 값을 변경함으로써 아날로그 출력을 사용할 수 있습니다.



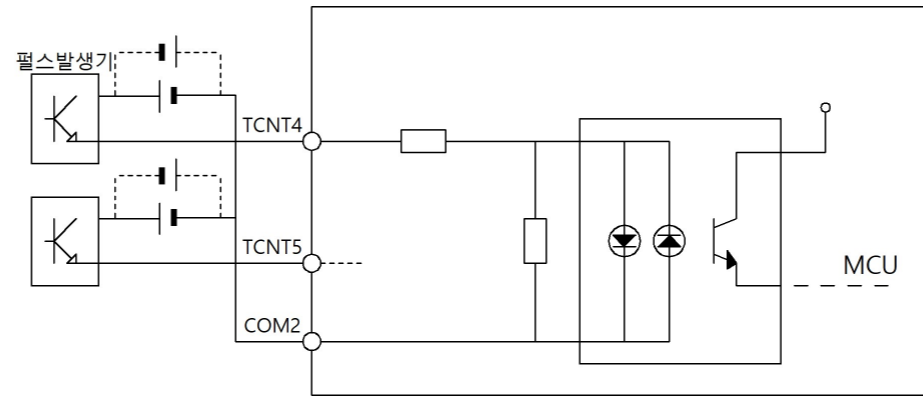
아날로그 입력 AI(0)에 입력된 아날로그 전압이 2.5V(512) 초과되면 트랜지스터 출력 O(39):P20을 On시키고, D80 메모리에 25000을 기록시킵니다.

아날로그 출력 AO(6)포트는 0~65535에서 25000의 값이 0~5V 범위로 출력되므로 약 1.9V(25000/65535 * 5)가 출력됩니다.



T(4)에 바인딩한 D94 워드 메모리는 자동으로 T(4)에 연결한 NTC 온도센서의 저항 값을 온도 값으로 변환하여 저장합니다.
T(4)에 연결한 온도센서의 온도 값이 21.2도를 넘으면 P22:O(41) 트랜지스터 출력을 On시킵니다.

고속카운터 (펄스입력 카운터)



관련 명령어

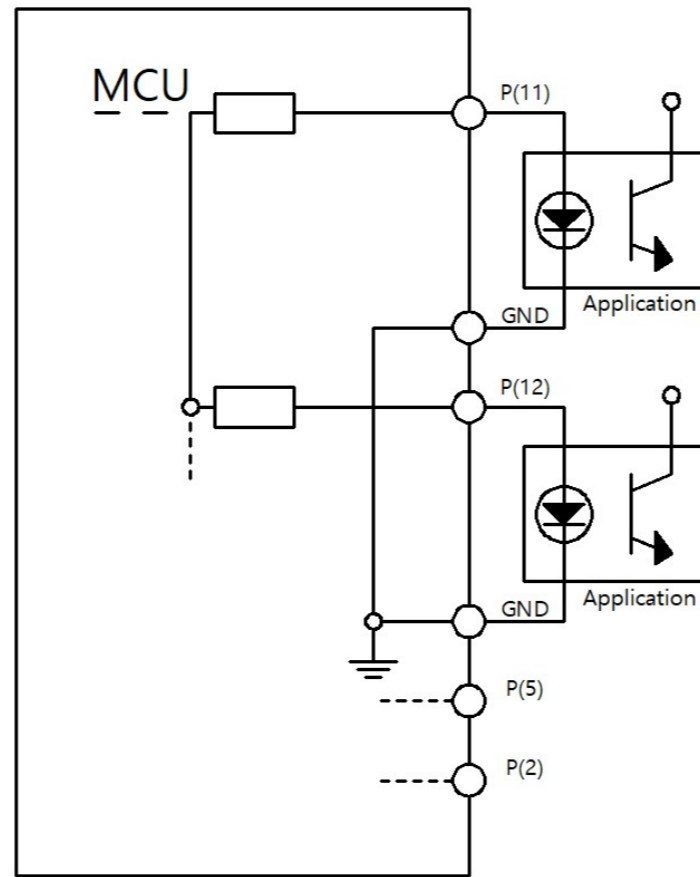
hcntRead(Channel) : 고속펄스 카운터 값을 읽어옵니다.

Return : unsigned long

Channel : TCNT4는 4, TCNT5는 5를 사용합니다.

hcntReset(Channel) : 고속펄스 카운터 값을 0으로 초기화합니다.

PWM 펄스출력



관련 명령어

analogWrite(Pin, Duty): PWM(Pin)포트에 Duty길이의 펄스를 출력합니다.

Pin : P(5)는 5, P(11)는 11을 사용합니다.

Duty : 0 ~ 255(Default) 또는 0 ~ 65535(Width 변경할 경우)를 사용합니다.

pwm1Width(Width): P(11), P(12)의 Width를 변경합니다. (Width : 0 ~ 65535)

pwm3Width(Width): P(5),P(2)의 Width를 변경합니다. (Width : 0 ~ 65535)

고속카운터 및 PWM 펄스출력 간단 예제

MPINO STUDIO : C코드를 사용할 경우

```
#include "LD.h"

unsigned long HCNT0, HCNT1;

void setup(void) {
    ladderSetup();
}

void loop(void) {
    HCNT0 = hcntRead(4);
    HCNT1 = hcntRead(5);
    if (HCNT0 > 70000) hcntReset(4);
    if (HCNT1 >= 90000) {
        analogWrite(11, 32767); //P(11)포트에 32767길이(Duty)의 펄스출력
        analogWrite(6, 10000); //P(5)포트에 10000길이(Duty)의 펄스출력
    }
}
```

TCNT4와 TCNT5의 고속펄스 카운터 값을 각각 HCNT0과 HCNT1에 저장합니다. 이후, HCNT0이 70000을 초과하면 TCNT4의 카운터 값을 0으로 초기화합니다. 또한, HCNT1이 90000 이상이 되면 P(11)에 65535길이 중 32767의 길이의 Duty를 가진 PWM 펄스가 출력되고 P(5)에 10000길이의 Duty를 가진 PWM 펄스가 출력됩니다.

MPINO STUDIO : LADDER LOGIC을 사용할 경우

아래와 같이 하드웨어설정에서 바인딩을 해주어야 합니다.

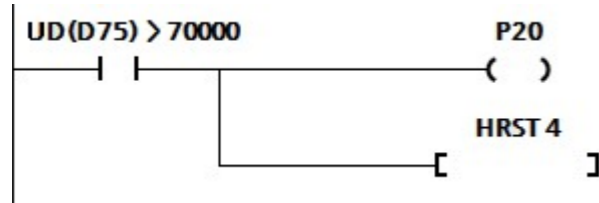
D75,D76에 TCNT4 고속카운터 값이 저장되고, D77,D78에 TCNT5 고속카운터 값이 저장됩니다.

명칭	핀 번호	핀 유형	바인딩
R(67)	67	OUTPUT	P33
R(68)	68	OUTPUT	P34
R(69)	69	OUTPUT	P35
AI(0)	0	ADIN	D90
AI(1)	1	ADIN	D91
AI(2)	2	ADIN	D92
AI(3)	3	ADIN	D93
T(4)	4	ADIN	D94
T(5)	5	ADIN	D95
AO(6)	6	PWM	
AO(7)	7	PWM	
P(11)	11	PWM	D70
P(12)	12	PWM	D71
P(5)	5	PWM	D72
P(2)	2	PWM	D73
TCNT4	4	HSC	D75
TCNT5	5	HSC	D77

⚠ TCNT4를 사용할 경우, 위 하드웨어 설정과 같이 AO(6)와 AO(7)를 바인딩하지 말아야 합니다.

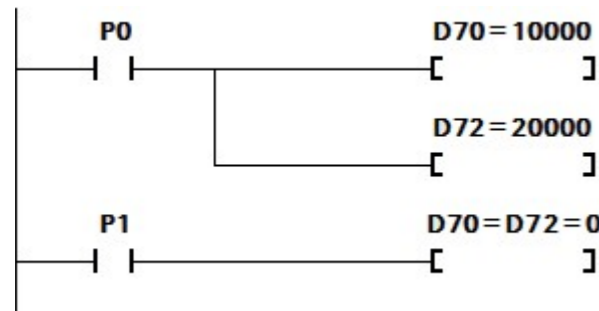
☞ 하드웨어 설정에서 TCNTx는 Double Word영역으로 바인딩됩니다. 때문에, D영역이 겹치는지 주의하셔야 하며, LADDER LOGIC에서 UD()를 사용하여 아래처럼 사용해야 합니다.

☞ 아래는 TCNT4:UD(75)가 70000을 초과하면 O(62):P20 출력이 On되고, TCNT4의 고속카운터 값을 0으로 리셋 시키는 LOGIC 입니다.



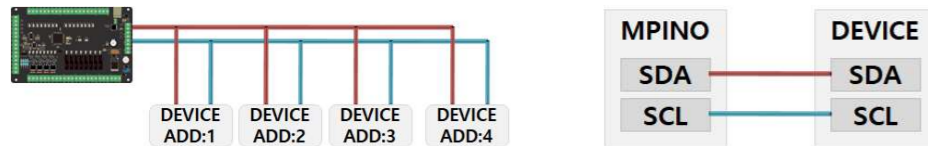
☞ 아래는 I(22):P0포트가 On되면, P(11):D70에 10000이 저장되면서 P(11)포트에 10000 길이의 PWM 펄스가 출력됩니다. 또한, P(5):D72에 20000이 저장되면서 P(5) 포트에 20,000 길이의 PWM 펄스가 출력됩니다.

☞ I(23):P1포트가 On되면, P(11):D70과 P(5):D72에 0이 입력되면서 P(11)포트와 P(5) 포트의 펄스출력은 정지됩니다.



□ I2C 통신

I2C 통신은 2개의 선으로 구성되며, 아래와 같이 여러 장비와의 통신이 가능합니다.



- ☞ 전송 속도 : 100k bps(Default), 400k bps
- ☞ 전송 길이 : 최대 10m
- ☞ 결선 방법 : 마스터와 모든 슬레이브 디바이스를 1:1 연결합니다.

☞ 참조
#include "Wire.h" 를 C코드 창 상단에 기입하여, Wire 함수들을 사용할 수 있습니다.

☞ 관련 명령어

- 마스터 모드 (송신)
 - **Wire.begin()** : I2C를 마스터 모드로 시작합니다.
 - **Wire.beginTransmission(Address)** : 슬레이브 Address를 저장합니다.
 - **Wire.write(Data)** : Data를 저장합니다.
 - **Wire.endTransmission(Stop)** : 저장한 Address와 Data를 전송합니다. Stop이 True이면, 전송이 끝나고 I2C라인을 정지시키고 false이면 정지시키지 않습니다. 정지되지 않는다는 것은 ReStart 된다는 것입니다.
 - **Wire.endTransmission()** : 위 함수에서 Stop이 True인 것과 동일한 함수입니다.

- 마스터 모드 (수신)

- **Wire.requestFrom(Address, Quantity, Stop)** : Address 주소의 디바이스에서 Quantity 개수의 Byte를 요청합니다. STOP이 True이면, 마지막에 I2C라인을 정지시키고 False이면 정지시키지 않습니다.
- **Wire.requestFrom(Address, Quantity)** : 위 함수에서 Stop이 True인 것과 동일한 함수입니다.
- **Wire.available()** : 수신된 데이터의 바이트 수를 리턴 합니다.
- **Wire.read()** : 수신된 데이터의 1개의 바이트를 읽어 옵니다.

☞ 그 외의 함수들은 [Arduino Wire Reference](#)에서 확인할 수 있습니다.

☞ MPINO STUDIO : C코드에서만 지원합니다.

```
#include "LD.h"
#include "Wire.h"

void setup(void) {
  ladderSetup();
  Wire.begin(); // I2C를 마스터 모드로 시작합니다.
  Serial.begin( 9600 ); // 0번 채널(다운로드포트)을 Open 합니다.
}

void loop(void) {
  ladderLoop();
  Wire.beginTransmission( 1 ); // 슬레이브 Address를 1로 저장합니다.
  Wire.write( 0x30 ); // 전송할 Data를 0x30으로 저장합니다.
  Wire.endTransmission( false ); //I2C시작,Address 전송, 0x30과 0x31전송, I2C재시작을 순서대로 실행합니다.

  Wire.requestFrom( 1, 10, true ); // 슬레이브 Address가 1인 디바이스에서 10 바이트를 읽어오고 I2C를 정지시킵니다.
  while ( Wire.available() ) // 읽은 바이트가 0이 될 때까지 루프를 형성
  {
    byte iRxData = Wire.read(); // 한 바이트를 읽어서iRxData에 저장
    Serial.println( iRxData ); // iRxData를 시리얼 모니터로 전송
  }
}
```

□ 시리얼 통신

시리얼 통신은 RS232, RS485, UART를 사용할 수 있습니다.

☞ RS232 : Serial1
1 : 1 통신이며, 최대 10m 거리의 통신을 할 수 있습니다.

☞ RS485 : Serial2
1 : N 통신이며, 최대 1km 거리의 통신을 할 수 있습니다.

☞ UART : Serial3
1 : 1 통신이며, 최대 1m 이하의 통신을 할 수 있습니다.

☞ 관련 명령어

- ☞ 함수의 Serial에는 각 채널명 Serial1,Serial2...로 치환하여 사용
- **Serial.begin()** : 시리얼 포트를 Open합니다.
- **Serial.write(byte)** : 1개의 Byte를 전송합니다.
- **Serial.write(array,length)** : Array에서 Length만큼 전송합니다.
- **Serial.available()** : 수신된 Data(Byte)의 개수를 리턴 합니다.
- **Serial.Read()** : 수신된 1개의 Byte를 읽어 옵니다.

☞ 더 많은 함수들을 [Arduino Serial Reference](#)에서 확인할 수 있습니다.

□ Modbus RTU Slave

- ☞ 산업용에서 범용적으로 쓰이는 프로토콜입니다.
- ☞ 통신 영역은 LADDER LOGIC의 메모리를 사용합니다.
- ☞ [MPINO STUDIO 사용설명서](#)에서 자세한 사용방법을 확인하실 수 있습니다.

☞ 관련 명령어

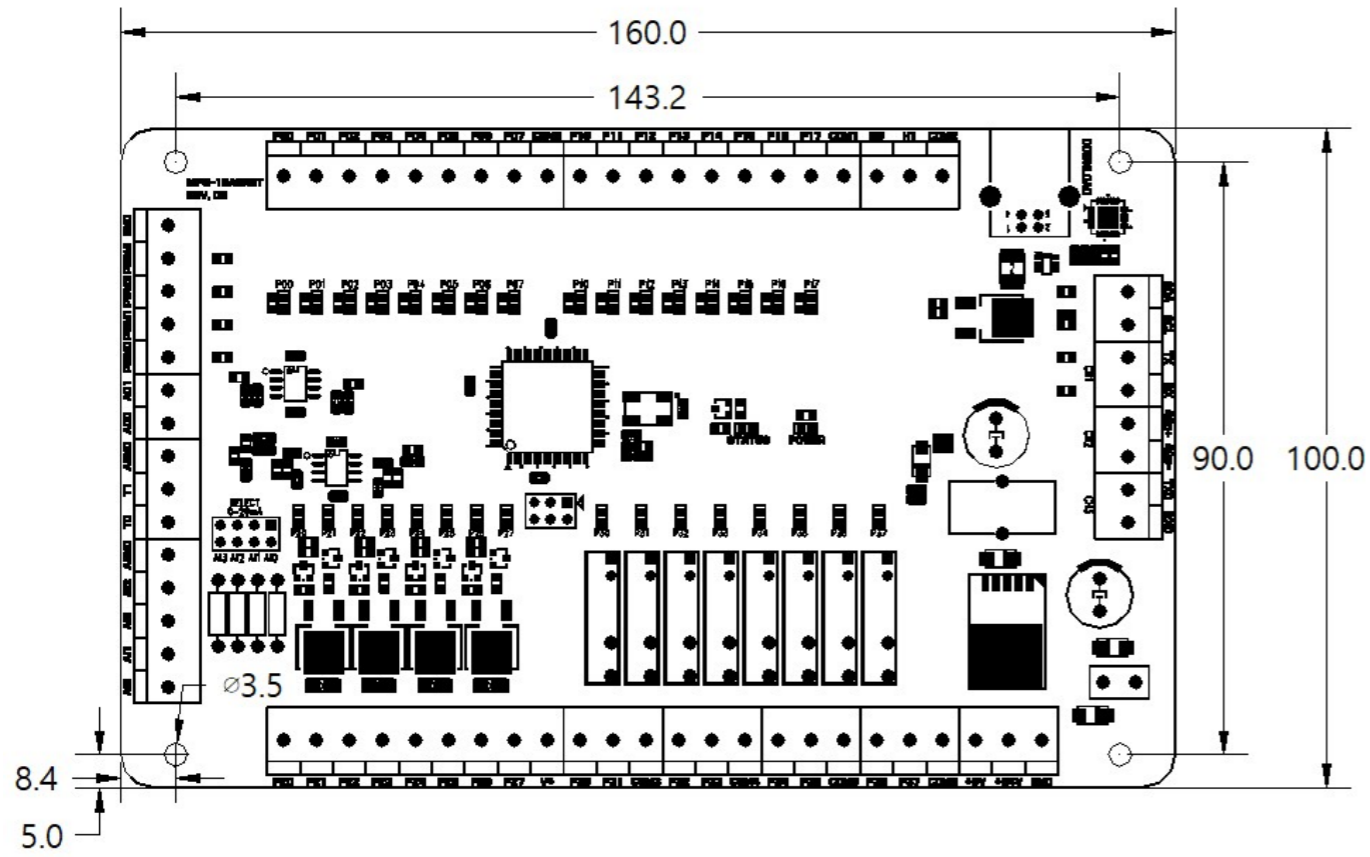
- **modbusStart(Serial, BaudRate, SlaveAddress)** : Serial 포트를 BaudRate와 SlaveAddress로 modbus RTU slave로 지정.
- **modbusStop()** : Modus RTU Slave를 모두 해제 합니다.

```
#include "LD.h"

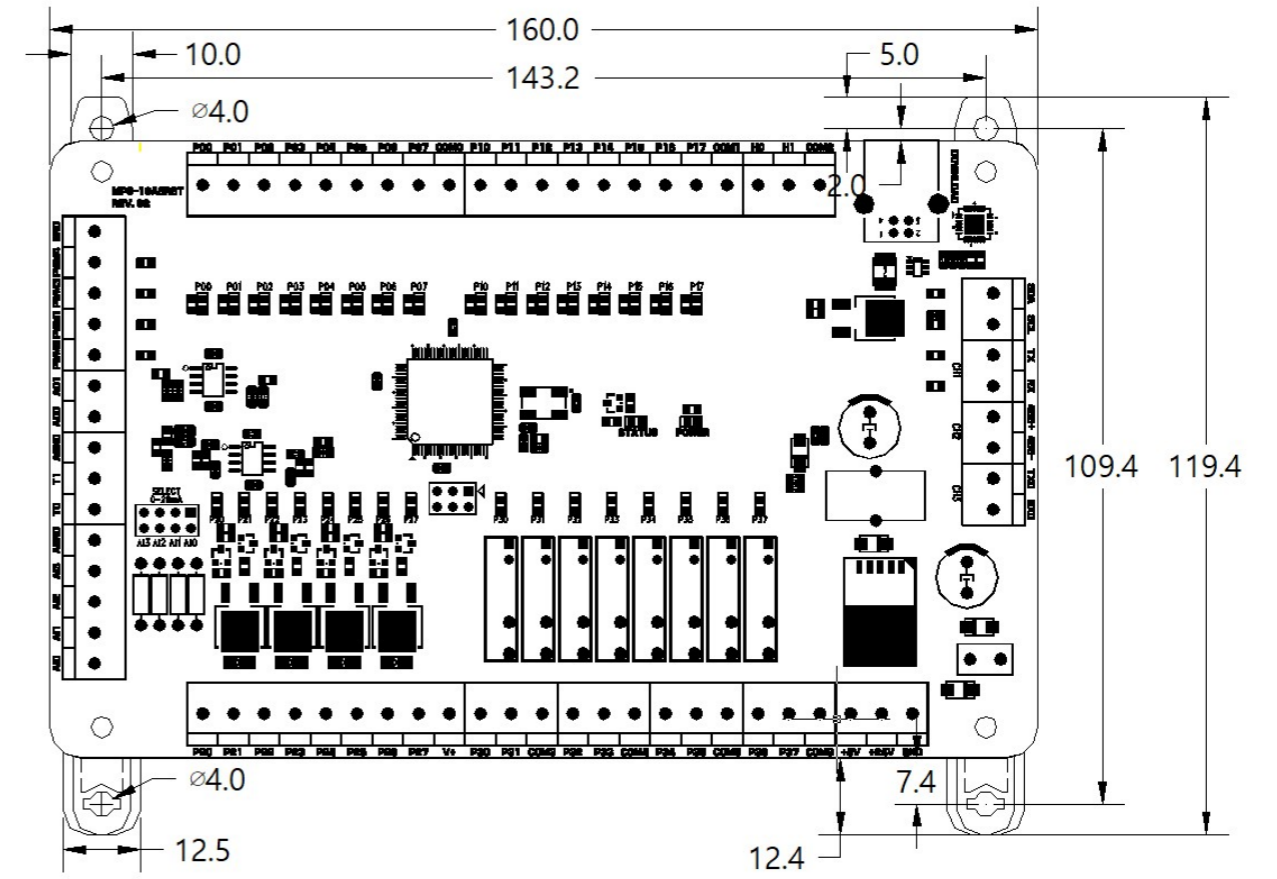
void setup(void) {
  ladderSetup();
  modbusStart( Serial1, 9600, 1 ); // Serial1 채널을 9600 보레이트와 1 슬레이브 어드레스로 modbus RTU slave를 시작합니다.
}

void loop(void) {
  ladderLoop(); // 모드버스 처리 루틴이 포함되어 있으므로 반드시 기입해야 합니다.
  D0 = 1234; //D0레지스터리에 1234값을 저장 // D0는 0x0000 시작어드레스
}
```

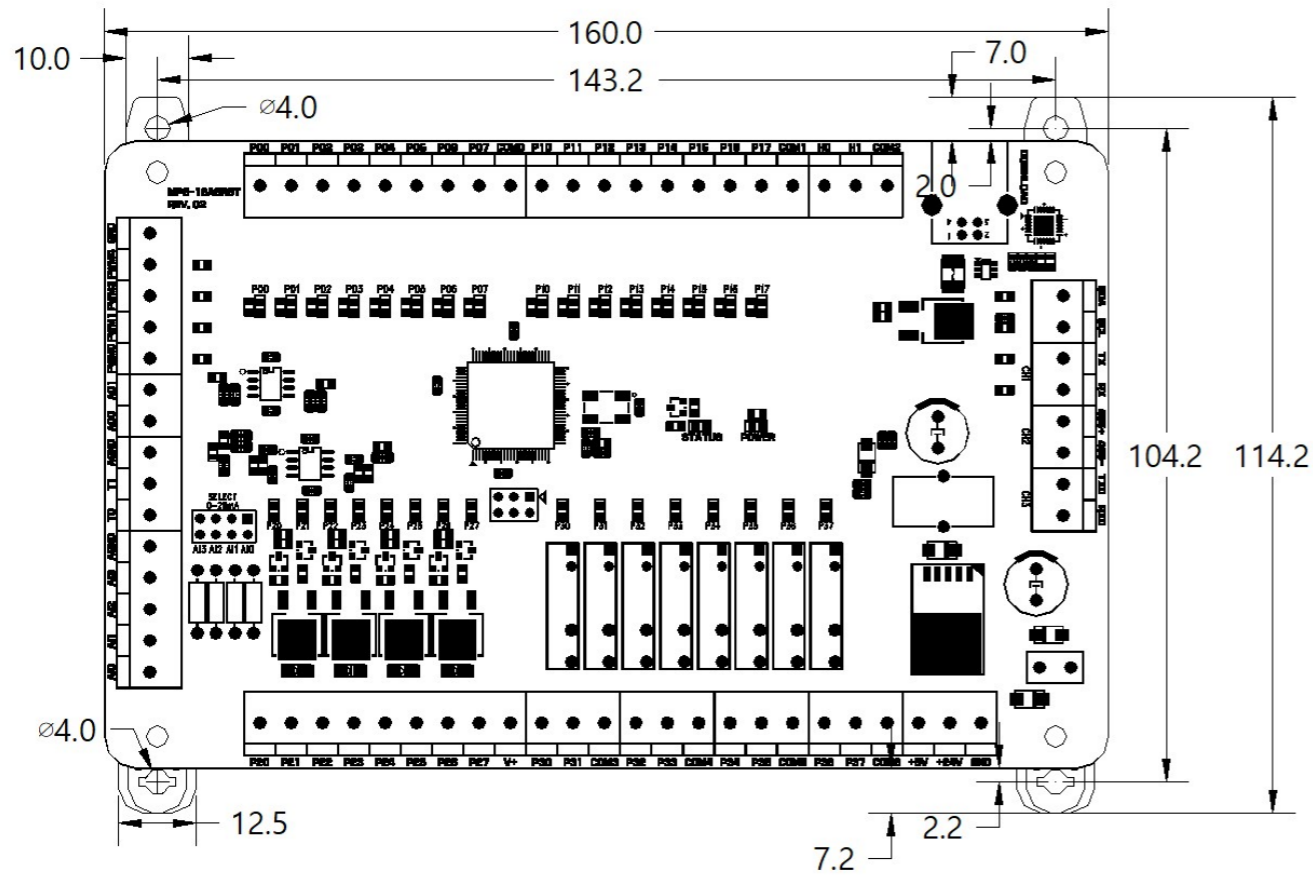
□ DIMENSION (PCB / 단레일 사용하지 않을 경우)



□ DIMENSION (클립 열었을 때 / 단레일 체결 전)



□ DIMENSION (클립 닫았을 때 / 단 레일 체결 후)



□ DIMENSION (단 레일 : 35mm)

