

MPAINO SERIES MPAINO-8A8RX

사용설명서 [ARDUINO IDE]

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.
- 경고**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, 방법/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 외부환경으로부터 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

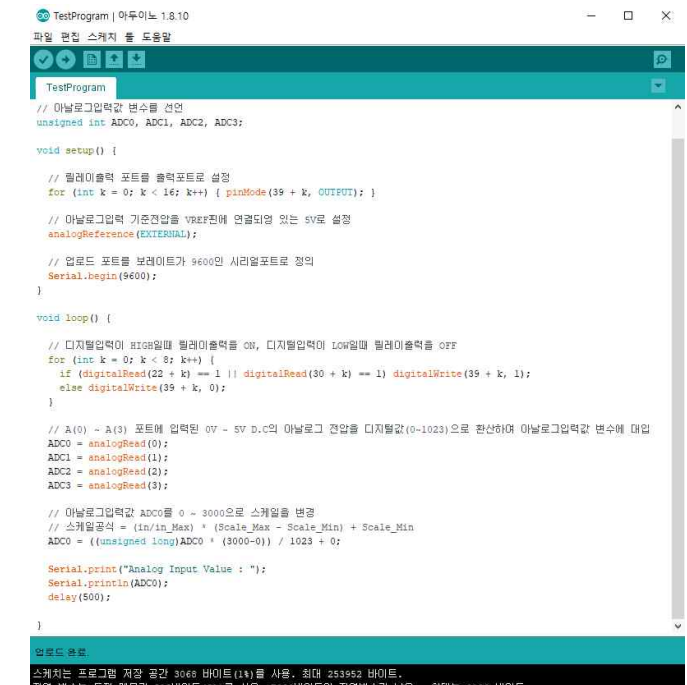
구분	개수	접점명	설명
보드	-	-	• MEGA2560
전원	-	전원전압	• DC 12V ~ 24V • SMPS 요구사항 : DC 24V 0.5A 이상
디지털 입력	8 포인트 < 절연 >	D2 ~ D5 /COM0 D6 ~ D9 /COM1	• 오퍼레이팅 입력 전압 : DC 0 ~ 40V • HIGH 인식 전압 : DC 5V 이상 • 4P / 1COM • 1COM당 NPN 및 PNP 선택 입력가능
		0	• 오퍼레이팅 연결 전압 - DC 0 ~ 30V , AC 0 ~ 250V
릴레이 출력	8 포인트 < 절연 >	D22 ~ D25 /COM2 D26 ~ D29 /COM3	• 4P / 1COM • 최대 출력 허용전류 - 5A / 1POINT - 8A / 1COM
		1	• 각 채널당 아래의 기능을 선택적 사용 - DC 0 ~ 5V / DC 1 ~ 5V - DC 0 ~ 10V - 0 ~ 20mA / 4 ~ 20mA - NTC 서미스터 온도센서 • 15비트 해상도 (0 ~ 32,767) • ± 0.2% 이내 정밀도 (half 입력 기준) • 128 SPS 샘플링 (Default) ~ 860 SPS
아날로그 입력	4 채널 < 비절연 >	SPI 통신사용 SS : D53 ADS1118 라이브러리 참조하여 사용	
		1채널 < 비절연 >	• I2C (WIRE 라이브러리 사용)
통신 채널	3 채널 < 비절연 >	RS232 RS485, UART	• Serial1 : RS232 • Serial2 : RS485 • Serial3 : UART

□ 메모리 사양서

- 256Kbyte Flash Memory
- 8Kbyte Data Memory

□ 프로그램 코딩 및 다운로드 소프트웨어

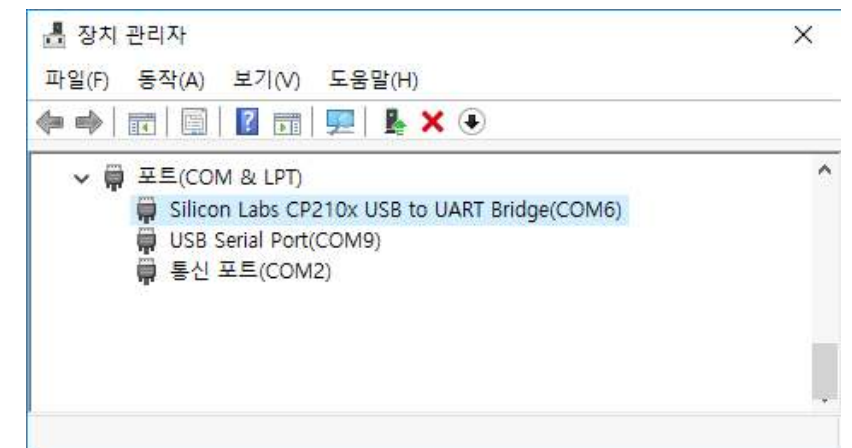
Arduino IDE는(<https://www.arduino.cc/en/Main/Software>)에서 다운로드 받을 수 있습니다.



[Arduino IDE]

□ 사용방법 [요약]

- 컴퓨터의 USB포트와 제품(MPAINO-8A8RX)에 다운로드 포트를 연결합니다.
- 아이로직스 자료실에서 Silicon Labs CP210x Usb Driver를 다운로드를 받아 설치합니다. (http://ilogics.synology.me/Drivers/CP210x_Windows_Drivers.zip)
- 윈도우의 장치관리자에서 COM포트를 확인합니다. (PC와 MPAINO-8A8RX 제품이 USB로 연결되어 있어야 합니다)



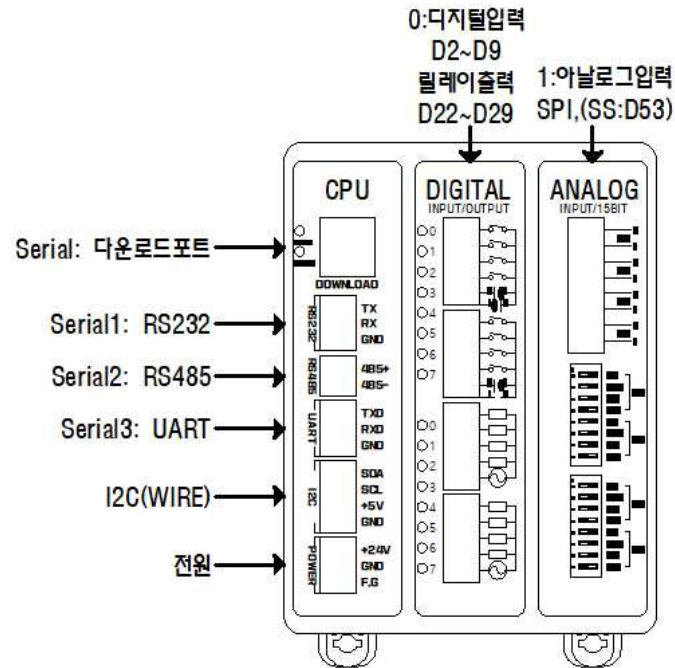
- Arduino IDE에서 위에서 확인한 COM포트를 선택합니다. (툴 -> 포트)
- Arduino IDE에서 Arduino Mega or Mega 2560을 선택합니다. (툴 -> 보드)
- 프로그래밍을 하고, 업로드를 합니다.

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

- 전원입력은 DC 12V~24V를 사용할 수 있습니다. DC 12V~24V는 LM2576 DC-DC Regulator를 통하여 DC 5V로 전환되어 내부회로에 전원을 공급합니다.
- 다운로드 포트에 USB 연결선으로 컴퓨터와 연결하면, 컴퓨터의 5V 전원을 사용하여 제품이 동작됩니다. (단, 아날로그 입력모듈 제외)
- I2C 터미널블럭의 +5V 단자는 최대 1A의 DC 5V 전원을 출력할 수 있습니다.

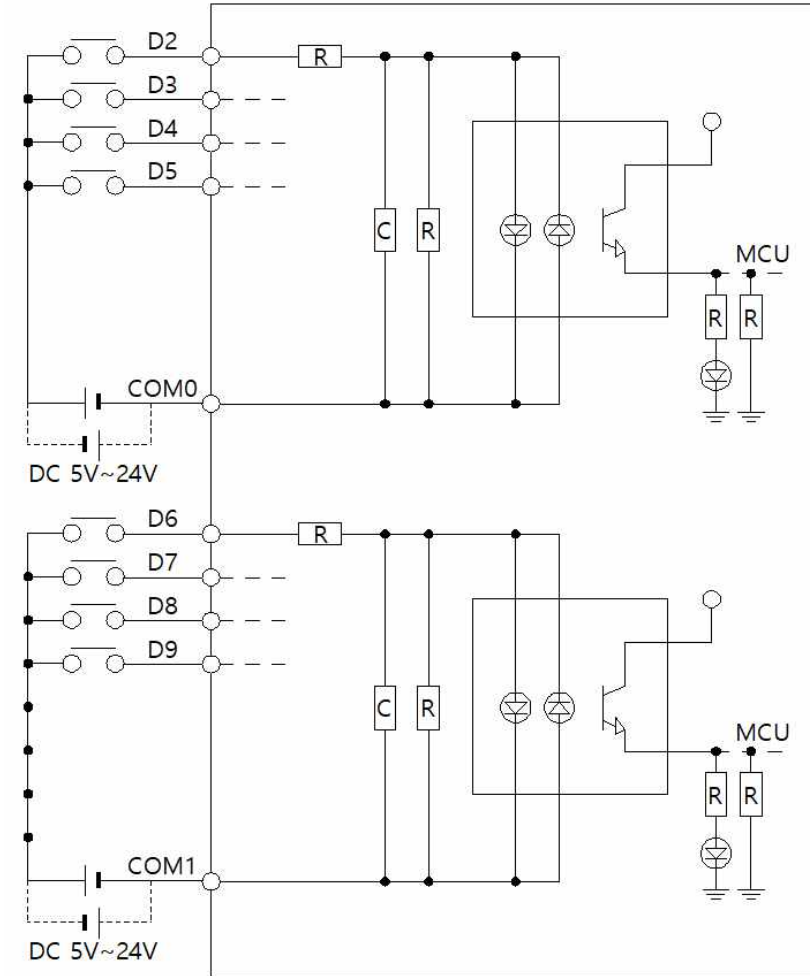
정전유지

- 제품은 DC 5V전원으로 모든 동작(아날로그 입력 제외)이 가능하도록 설계되어 있습니다. 따라서, DOWNLOAD (USB-B TYPE) 컨넥터에 배터리 등을 연결하여, 정전시에도 제품동작 및 메모리를 유지하게 할 수 있습니다.
- 배터리 연결이 불가할 경우와 장기간 정전시에도 데이터의 보존을 원할 경우에는 MCU에 내장된 EEPROM을 사용해주시기 바랍니다. 비휘성 메모리인 EEPROM을 이용하여 메모리를 보존할 수 있습니다. 단, EEPROM은 100,000번 이상 기록(Write)를 할 경우, 해당 섹션의 불량률이 발생할 수 있으므로 수시로 변경되는 데이터를 기록하는 것은 올바르지 않습니다.

디지털 입력

디지털입력 포트는 D2~D9 핀번호를 사용하며, 포토커플러를 사용하여 다양한 전압을 입력 받을 수 있고 양방향(NPN, PNP) 입력을 지원합니다. 또한, 외부와 내부의 회로를 분리하여 노이즈에 강하며, 채터링 방지가 됩니다.

D2 ~ D9에 DC 5V~24V가 스위치, 센서 등에 의해 입력된다면, COM0는 GND를 연결해야 합니다. 반대로, D2~D9에 GND가 스위치, 센서 등에 의해 입력된다면, COM0는 DC 5V~24V를 연결해야 합니다.



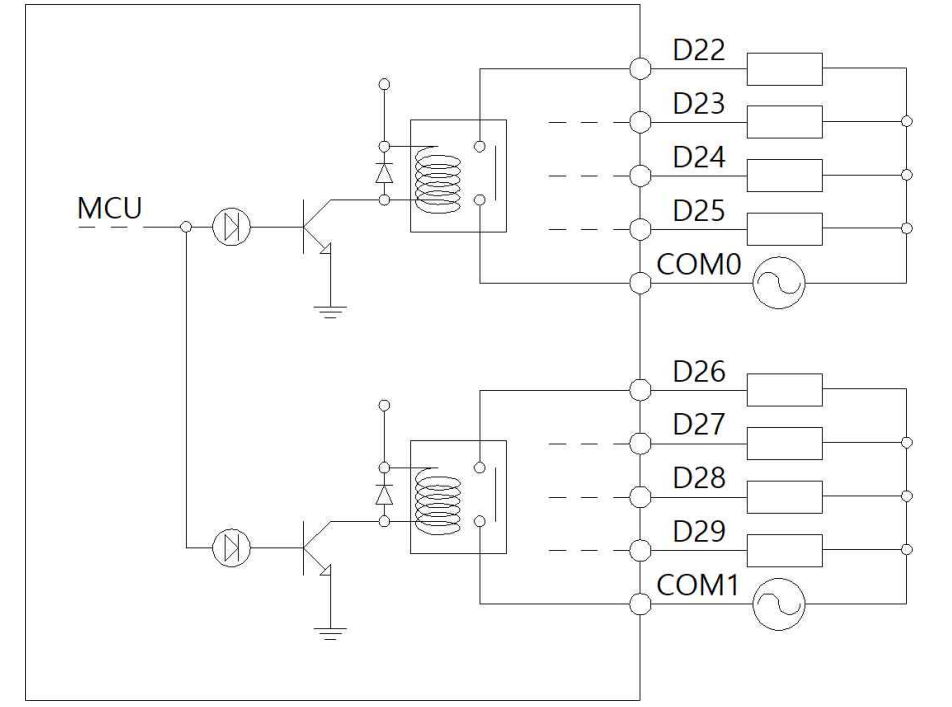
- 관련 명령어
pinMode(Pin, INPUT/OUTPUT) Pin포트를 INPUT 또는 OUTPUT으로 설정.
digitalRead(Pin) Pin포트의 입력상태를 "0" 또는 "1"로 반환.
digitalWrite(pin, 0/1) pin포트의 출력상태를 LOW 또는 HIGH로 변환.

디지털 입/출력 예제

```
void setup() {
  pinMode(22, OUTPUT); // D22 릴레이를 출력모드로 설정합니다.
}
void loop() {
  // D2가 HIGH 이면, D22 릴레이를 ON 시킵니다.
  if (digitalRead(2) == 1) { digitalWrite(22, HIGH); }
  // D2가 HIGH가 아니면, 즉 LOW 이면, D22 릴레이를 OFF 시킵니다.
  else { digitalWrite(22, LOW); }
}
```

릴레이 출력

각각의 COM에 연결한 전원이 D22~D29로 출력됩니다. 릴레이는 스위치를 누르는 것과 같이 각각의 COM과 D22~D29를 물리적으로 연결하기 때문에 D.C 와 A.C를 모두 ON/OFF 시킬 수 있습니다.



상태 LED

D48핀으로 STATUS LED를 ON/OFF 시킬 수 있습니다.

```
void setup() {
  pinMode(48, OUTPUT); //STATUS LED를 출력모드로 설정
}
void loop() {
  // D2가 HIGH 이면, STATUS LED를 ON 시킵니다.
  if (digitalRead(2) == 1) { digitalWrite(48, HIGH); }
  // D2가 HIGH가 아니면, 즉 LOW 이면, STATUS LED를 OFF 시킵니다.
  else { digitalWrite(48, LOW); }
}
```

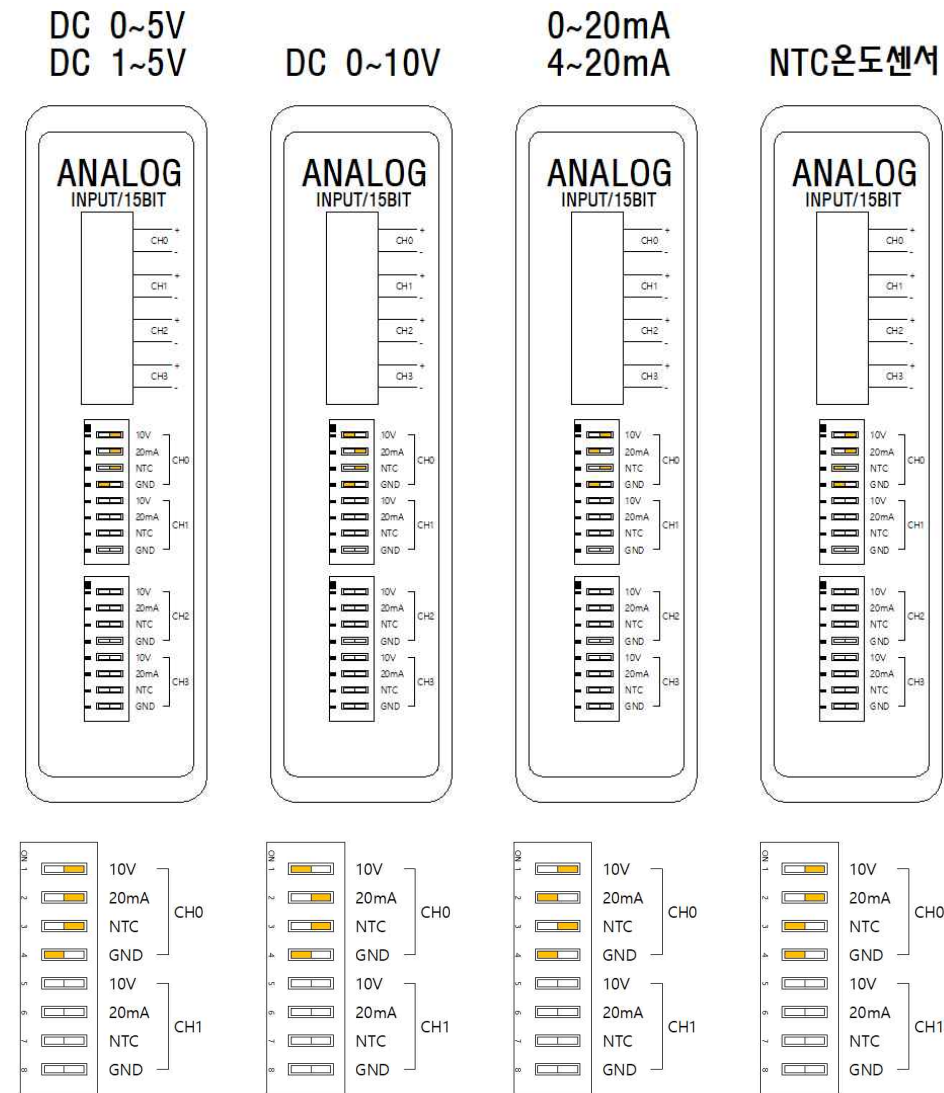
1초마다 출력을 ON/OFF 시키는 예제

delay(ms) 명령어를 사용하여 시간지연을 사용할 수 있습니다.

```
void setup() {
  pinMode(23, OUTPUT); // D23 릴레이를 출력모드로 설정합니다.
}
void loop() {
  digitalWrite(23, HIGH); // D23 릴레이를 ON 시킵니다.
  delay(1000); // 1000ms 동안 기다립니다.
  digitalWrite(23, LOW); // D23 릴레이를 OFF 시킵니다.
  delay(1000); // 1000ms 동안 기다립니다.
}
```

□ 아날로그 입력

- 아날로그 입력 모듈에는 16비트의 ADS1118 I.C가 삽입되어 있습니다.
- 16비트는 Full Scale(-32768 ~ +32767)의 범위 이지만 본 제품에서는 음수를 사용하지 않아 15비트(0~32767)입니다.
- ADS1118은 SPI 통신으로 측정한 아날로그 값을 읽어올 수 있습니다.
- 아두이노 IDE에서 메뉴(툴->라이브러리 관리)에서 ADS1118을 검색하여 라이브러리를 다운로드 받거나 저희가 제공하는 라이브러리를 아두이노 IDE에 참조하여 사용해야 합니다.
- 라이브러리는 C:\Users\윈도우 계정이름\Documents\Arduino\libraries\ 폴더에 ADS1118_library 폴더 채로 있어야 합니다.
- 아날로그 입력범위는 DC 0~5V, DC 0~10V, 0~20mA, NTC(10k Ω β -3950) 온도센서중 한 개를 채널별로 선택하여 사용할 수 있습니다.
- DC 1~5V 및 4~20mA의 아날로그 입력을 사용하고 싶을 경우, 아날로그 측정값에서 DC 1V 또는 4mA의 아날로그 값인 6553(32767/5)을 빼서 사용해야 합니다.
- 아날로그 입력범위는 아래와 같이 DIP스위치로 변경할 수 있습니다.



- DIP스위치 설정은 예시로 채널0 설정만 표시하였습니다.
- GND 스위치는 아날로그출력 제품의 전원이 아두이노 PLC와 공통이면 OFF를 권장합니다.
(ON으로 설정하여도 동작에 문제는 없습니다.)

- 아날로그 입력의 양단 전압은 OPAMP를 통하여 차동증폭을 하기 때문에 공통 노이즈를 감소하고 과전압으로부터 보호 됩니다. OPAMP를 사용하기 때문에 DC 0~10V의 아날로그 입력모드를 사용할 경우, 제품의 전원전압이 DC 12V 이상이어야 정상동작합니다.

□ 아날로그 입력 예제

- 아날로그 입력은 ADS1118 라이브러리를 참조 및 명령어를 사용하여 전기적인 신호를 0~32,767의 디지털값으로 환산하여 사용할 수 있습니다.

```
#include <ADS1118.h>
#include <SPI.h>

int ADC0, ADC1, ADC2, ADC3;

ADS1118 ads1118(53); //칩 선택 핀을 53으로 클래스 선언

void setup() {
  Serial.begin(9600); //다운로드 포트의 보레이트를 9600으로 시작

  ads1118.begin(); //SPI 통신을 시작.
  ads1118.disablePullup(); //통신 풀업 저항을 OFF,
  ads1118.setSingleShotMode(); //싱글(One Shot) Conversion 모드로 설정
  ads1118.setFullScaleRange(ads1118.FSR_2048); //기준전압을 2.048V로 설정
  ads1118.getMilliVolts(); //초반 설정값을 전달하기 위해 통신동작
}

void loop() {
  ADC0 = ads1118.getADCValue(ads1118.AIN_0);
  Serial.print("CH0 : "); Serial.println(ADC0);

  ADC1 = ads1118.getADCValue(ads1118.AIN_1);
  Serial.print("CH1 : "); Serial.println(ADC1);

  ADC2 = ads1118.getADCValue(ads1118.AIN_2);
  Serial.print("CH2 : "); Serial.println(ADC2);

  ADC3 = ads1118.getADCValue(ads1118.AIN_3);
  Serial.print("CH3 : "); Serial.println(ADC3);

  Serial.println("=====");
}
```

□ NTC 온도입력 예제

- NTC 온도센서 입력은 ADS1118 라이브러리를 참조 및 명령어를 사용하여 NTC 온도센서의 온도에 의한 저항변화를 디지털 수치로 변환하여 NTC의 고유 온도-저항 변환 테이블을 응용 또는 Log그래프 특성을 사용하여 온도값으로 환산합니다. 아래는 Log그래프 특성을 사용한 예제입니다.

```
#include <ADS1118.h>
#include <SPI.h>

float TEMPO;

ADS1118 ads1118(53); //칩 선택 핀을 53으로 클래스 선언

void setup() {
  Serial.begin(9600); //다운로드 포트의 보레이트를 9600으로 시작

  ads1118.begin(); //SPI 통신을 시작.
  ads1118.disablePullup(); //통신 풀업 저항을 OFF,
  //싱글(One Shot) Conversion 모드로 설정
  ads1118.setSingleShotMode();
  //기준전압을 2.048V로 설정
  ads1118.setFullScaleRange(ads1118.FSR_2048);
  ads1118.getMilliVolts(); //초반 설정값을 전달하기 위해 통신동작
}

void loop() {
  TEMPO = nTemp(ads1118.getADCValue(ads1118.AIN_0));
  Serial.print("CH0 : "); Serial.println(TEMPO);
}

float nTemp(unsigned int RawADC)
{
  float v;
  v = (15999.5F / (float)RawADC) - 1.0F;
  v = 10000.0F / v;
  float steinhart;
  steinhart = v / 10000.0F;
  steinhart = log(steinhart);
  steinhart /= 3950.0F;
  steinhart += 1.0F / (25.0F + 273.15F);
  steinhart = 1.0F / steinhart;
  steinhart -= 273.15F;
  return steinhart;
}
```

- NTC 서미스터는 온도에 의한 저항변화가 리니어하지 않기 때문에 온도-저항 변환테이블을 이용하여 1도단위의 기울기를 이용하여 온도값을 구하는 것이 보다 정확합니다. 이는 저희 (주)아이로직스 블로그에서 설명하고 있으니 참고 바랍니다.

□ 인터럽트 (attachInterrupt)

- ☞ 디지털신호의 입력을 받아 빠르게 처리해야 하는 사항이 있을 때 사용합니다.
- ☞ 디지털입력 터미널블럭의 D2, D3을 이용하여 총 2개의 인터럽트 핀을 사용할 수 있습니다.

☞ 관련 명령어

attachInterrupt(digitalPinToInterrupt(pin)), ISR, mode);

- pin : 2 또는 3
- ISR : 호출되는 함수명
- mode : LOW, CHANGE, RISING, FALLING
 - LOW : 하강검출 (입력상태가 ON에서 OFF로 될 때)
 - CHANGE : 변경검출 (입력상태가 변경될 때)
 - HIGH : 상승검출 (입력상태가OFF에서 ON으로 될 때)

- ☞ 입력이 OFF였다가 ON될 때, _INT2() 함수를 호출하는 예제입니다.

```
void setup() {
  // 다운로드포트를 9600보레이트로 오픈
  Serial.begin(9600);
  // D2핀에 상승엿지 입력이 검출되면, _INT2 함수 호출실행
  attachInterrupt(digitalPinToInterrupt(2),_INT2, RISING);
}

void loop() {
}

// D2핀에 상승엿지 입력이 검출되면, 실행되는 함수.
void _INT2(){
  // 다운로드포트로 "ok"를 송신
  Serial.println("ok");
}
```

□ 디버깅

- ☞ Debug는 Serial 함수를 이용해 주세요.

```
void setup() {
  Serial.begin(9600); // 업로드 포트를 보레이트가 9600인 시리얼포트로 정의
}

void loop() {
  // 디지털입력이 ON 되면 릴레이출력을 ON, OFF일때 릴레이출력을 OFF
  for (int k = 0; k < 8; k++) {
    if (digitalRead(2 + k)==1) digitalWrite(22 + k, 1);
    else digitalWrite(22 + k, 0);
  }
  // 디지털입력 D2가 HIGH일 경우 디버깅 포트에 "D2 HIGH"를 전송
  // 디지털입력 D2가 LOW일 경우 디버깅 포트에 "D2 LOW"를 전송
  if (digitalRead(2) == 1) Serial.println("D2 HIGH");
  else if (digitalRead(2) == 0) Serial.println("D2 LOW");
  delay(500);
}
```

□ I2C(WIRE) 통신포트

- ☞ 1개의 I2C(WIRE) 통신포트를 지원합니다.
- ☞ 명령어는 링크를 참조해 주시기 바랍니다. <https://www.arduino.cc/en/Reference/Wire>

□ 시리얼 통신포트

- ☞ RS232는 Serial1, RS485는 Serial2, UART는 Serial3 함수를 이용해 주세요.
- ☞ 모드버스 통신은 Arduino IDE에서 툴 -> 라이브러리 관리..에서 ModbusRTU_Slave by geabong의 라이브러리를 다운로드 받아서 라이브러리를 등록하여 사용할 수 있습니다.

□ MPINO STUDIO

- ☞ 저희 (주)아이로직스에서는 산업에서 사용하기 쉽도록 Arduino 와 Ladder Logic을 모두 사용하여 프로그램 할 수 있는 MPINO STUDIO를 무료로 제공하고 있습니다.

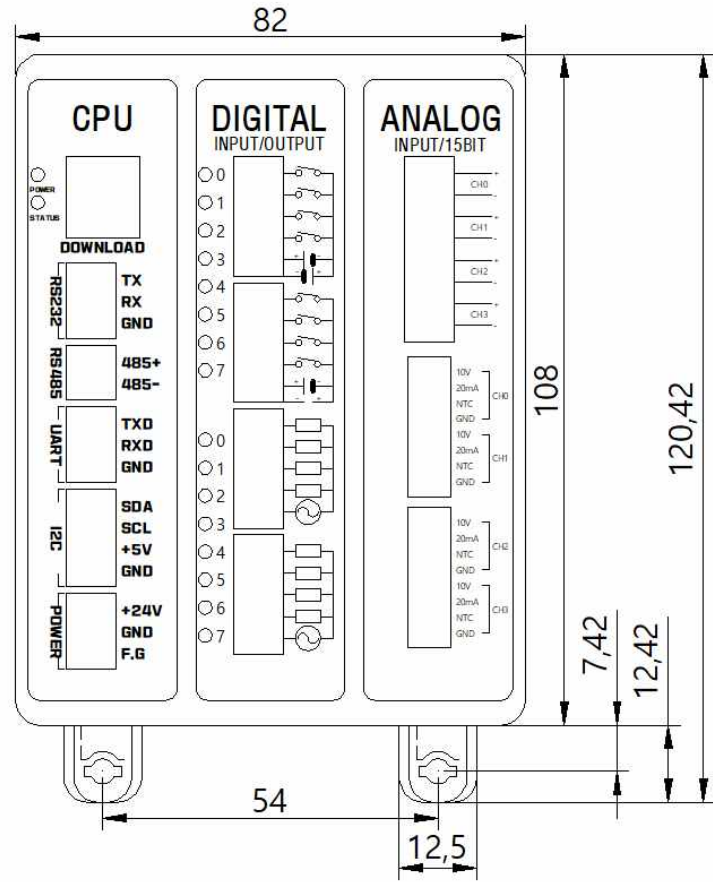
□ MP STUDIO

- ☞ 저희 (주)아이로직스에서는 Ladder Logic만을 사용하여 프로그램 할 수 있는 MP STUDIO를 무료로 제공하고 있습니다. MP STUDIO는 MPS 및 MPA 시리즈 제품군에 사용할 수 있습니다.

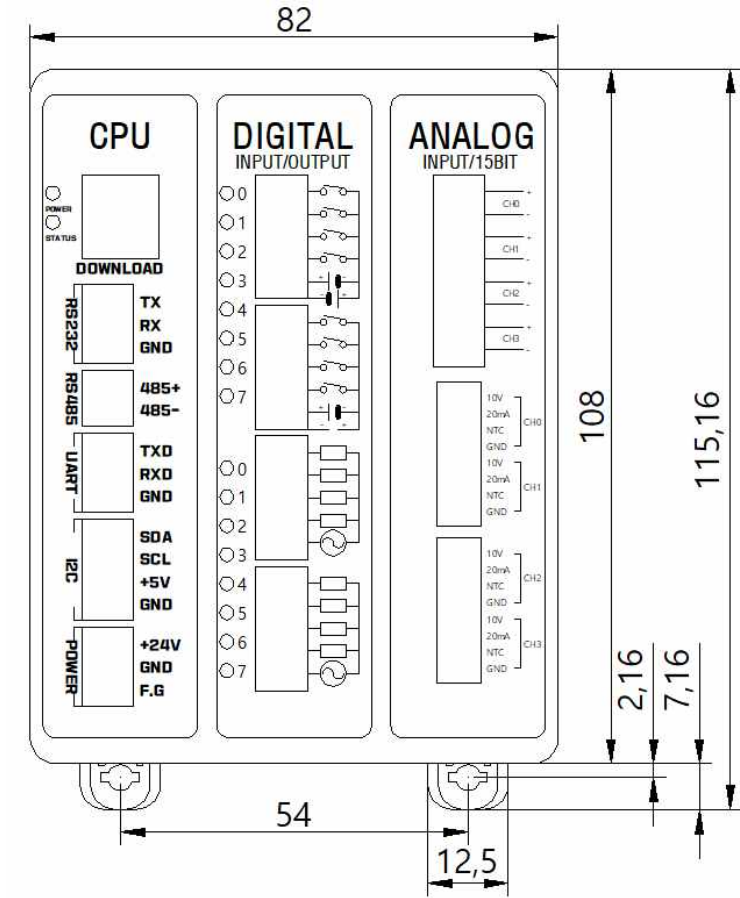
□ 감사드립니다.

- ☞ 저희 (주)아이로직스의 제품을 구매해주셔서 감사드립니다.
- ☞ 구매는 [Http://www.ilogics.co.kr](http://www.ilogics.co.kr) 쇼핑몰에서 하실 수 있습니다.
- ☞ 구매/기술 상담은 031-505-5020 또는 010-8485-5020으로 전화 주시기 바랍니다. (상담시간은 오전10시~오후5시 입니다)

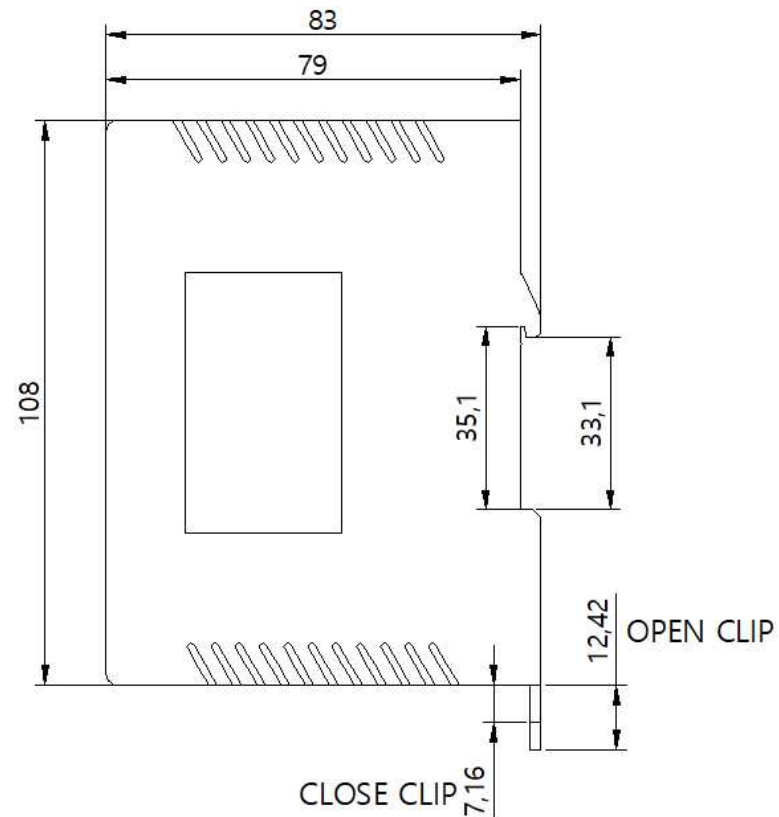
□ DIMENSION (클립 열었을 경우)



□ DIMENSION (클립 닫았을 경우)



□ DIMENSION (클립 열었을 때 / 단 레일 체결 후)



□ DIMENSION (단 레일 : 35mm)

