

MPAINO SERIES MPAINO-8A8R

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.
- ⚠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 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, 물적 자원에 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
	2 포인트 < 절연 >	D2 ~ D3 / COM0	• 오퍼레이팅 입력 전압 : DC 0 ~ 40V • HIGH 인식 전압 : DC 5V 이상 • 입력가능 주파수 : 최대 8kHz • 디지털 입력 포인트를 사용
*인터럽트	2 포인트 < 비절연 >	0 SDA (D20), SCL (D21)	• 오퍼레이팅 입력 전압 : DC 0 ~ 5V • HIGH 인식 전압 : GND (LOW) • 입력가능 주파수 : 최대 50kHz • 4.7kΩ Pull-Up 내장 • 인터럽트 사용시 I2C 통신 사용불가
	8 포인트 < 절연 >	D30 ~ D33 /COM2 D34 ~ D37 /COM3	• 오퍼레이팅 연결 전압 - DC 0 ~ 30V , AC 0 ~ 250V • 4P / 1COM • 최대 출력 허용전류 - 5A / 1POINT - 8A / 1COM
릴레이 출력	1채널 < 비절연 >	I ² C	• I2C (WIRE 라이브러리 사용) • SDA: D20, SCL: D21
	3 채널 < 비절연 >	RS232 RS485, UART	• Serial1 : RS232 • Serial2 : RS485 • Serial3 : UART (TX: D14, RX: D15)

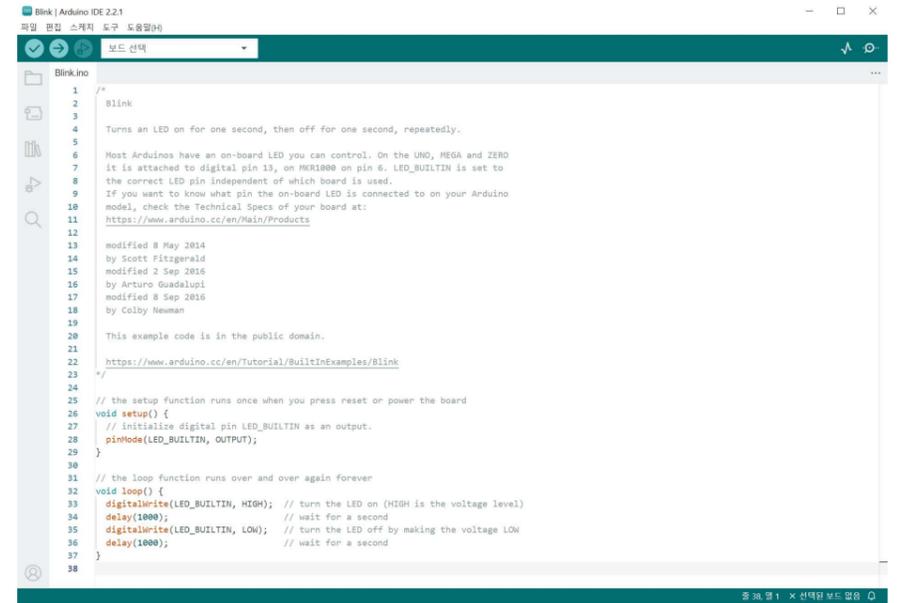
* 추가모듈 YK, Y2K, K2, Y3, Y4를 장착한 경우 디지털 입력 D2 ~ D9 단자 비교사항 는 D22 ~ D29로 변경됩니다. 또한, D2, D3 인터럽트를 사용할 수 없습니다.

메모리 사양서

- 256Kbyte Flash Memory
- 8Kbyte Data Memory
- 4Kbyte EEPROM Memory

프로그램 코딩 및 다운로드 소프트웨어

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



사용방법 [요약]

- 컴퓨터의 USB포트와 제품(MPAINO-8A8R)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-8A8R 제품이 USB로 연결되어 있어야 합니다)



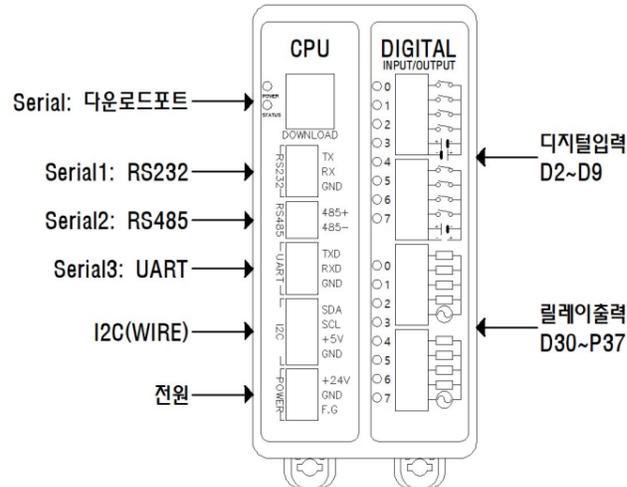
- Arduino IDE에서 위에서 확인한 COM포트를 선택합니다. (도구 -> 포트)
- Arduino IDE에서 Arduino Mega or Mega 2560을 선택합니다. (도구 -> 보드)
- Arduino IDE에서 프로그래머를 AVR ISP로 선택합니다. (도구 -> 프로그래머)
- 프로그래밍을 하고, 업로드를 합니다.

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://blog.naver.com/ilogics/223802536404>)

기능별 위치



전원

전원입력은 DC 12V ~ 24V를 사용할 수 있습니다. 입력된 전압은 Regulator를 통하여 DC 5V로 전환되어 사용되며, 전환된 DC 5V를 외 부에서 사용할 수 있도록 I2C 터미널단자의 +5V포트를 통하여 최대 1A까지 사용할 수 있습니다.

제품에 +24V 전원을 투입하지 않고 다운로드 포트에 USB 케이블을 컴퓨터와 연결하면, 컴퓨터의 5V 전원을 사용하여 제품이 동작됩니다.

정전유지

DC +5V 전원으로 모든 동작이 가능하도록 설계되어 있습니다. DC +5V 단자에 배터리를 연결하여, 정전 시에는 배터리의 전원으로 절체 되도록 하여 정전유지가 가능합니다.

비활성 메모리인 EEPROM을 이용하여 메모리를 보존할 수 있습니다. 단, EEPROM은 100,000번 이상 기록(Write)을 할 경우, 해당 섹션의 불량 발생 할 수 있으므로 빠른 속도로 변경되는 데이터를 기록하는 것은 올바르지 않습니다. 상시로 변하는 값을 EEPROM을 통하여 정전유지를 하고자 한다면 아래 블로그에서 방법을 설명하고 있으니 참고해주시기 바랍니다.

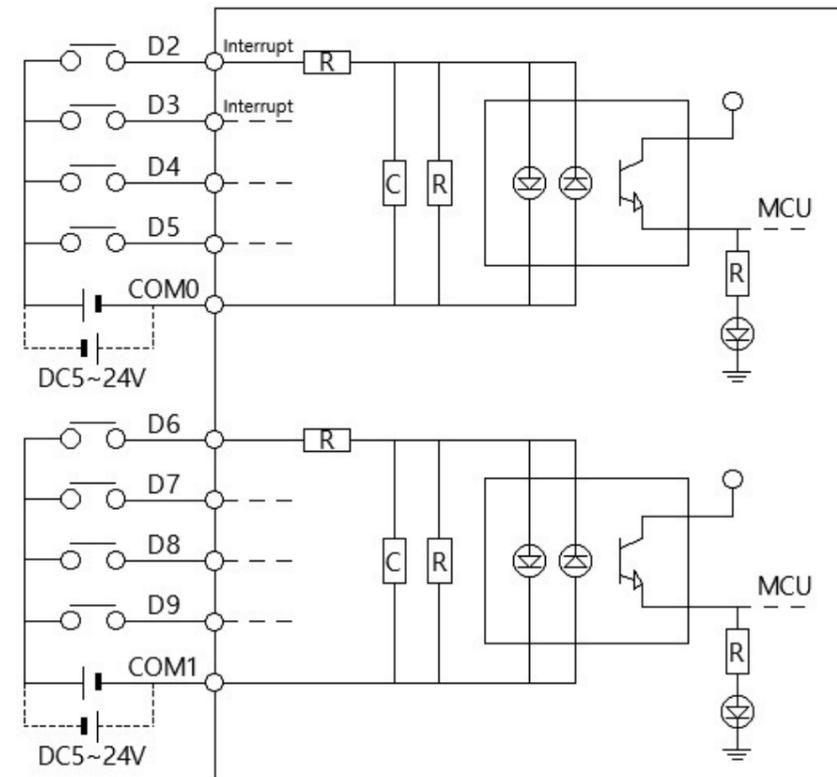
(<https://blog.naver.com/ilogics/223805063172>)

디지털 입력

디지털입력 포트는 D2~D9 핀 번호를 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, COM0은 GND를 연결해야 합니다. 반대로, D2~D9에 GND가 스위치, 센서 등에 의해 입력된다면, COM0은 DC 5V~24V를 연결해야 합니다.

아날로그 출력(Y모듈), 고속펄스 출력(K모듈)을 추가하여 YK, Y2K, K2, Y3, Y4 모듈을 장착한 경우 D2 ~ D9 포트는 D22 ~ D29포트로 변경됩니다.



관련 명령어

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

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

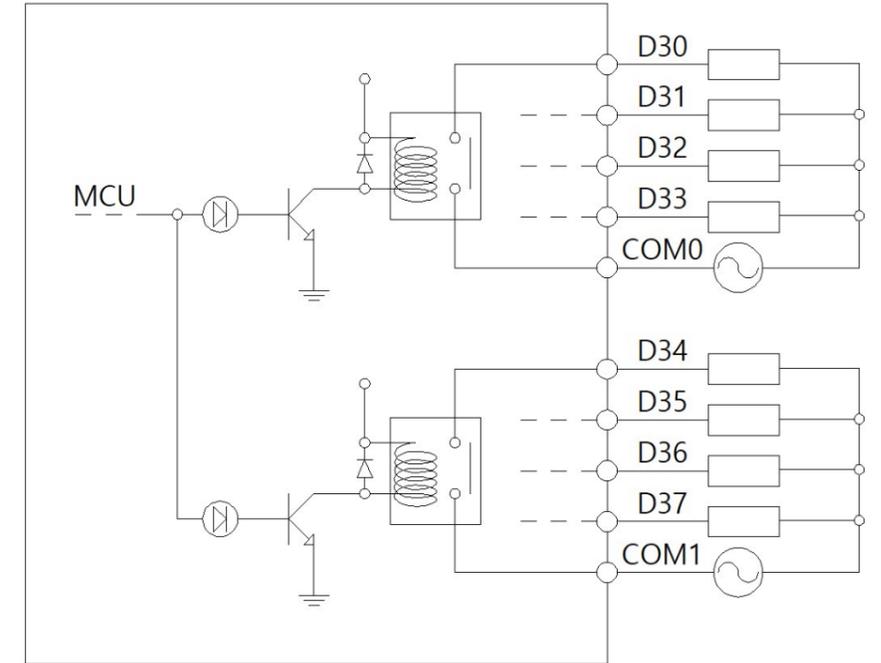
digitalWrite(pin, 0/1) pin포트의 출력상태를 LOW 또는 HIGH로 변환.

디지털 입/출력 예제

```
void setup() {
  pinMode(30, OUTPUT); // D30 릴레이를 출력모드로 설정합니다.
}
void loop() {
  // D2가 HIGH 이면, D30 릴레이를 ON 시킵니다.
  if (digitalRead(2) == 1) { digitalWrite(30, HIGH); }
  // D2가 HIGH가 아니면, 즉 LOW 이면, D30 릴레이를 OFF 시킵니다.
  else { digitalWrite(30, LOW); }
}
```

릴레이 출력

각각의 COM에 연결한 전원이 D30~D37로 출력됩니다. 릴레이는 스위치를 누르는 것과 같이 각각의 COM과 D30~D37을 물리적으로 연결하기 때문에 D.C 와 A.C를 모두 ON/OFF 시킬 수 있습니다.



1초마다 출력을 ON/OFF 시키는 예제

delay(ms) 명령어를 사용하여 시간지연을 사용할 수 있습니다.

```
void setup() {
  pinMode(31, OUTPUT); // D31 릴레이를 출력모드로 설정합니다.
}
void loop() {
  digitalWrite(31, HIGH); // D31 릴레이를 ON 시킵니다.
  delay(1000);           // 1000ms 동안 기다립니다.
  digitalWrite(31, LOW); // D31 릴레이를 OFF 시킵니다.
  delay(1000);          // 1000ms 동안 기다립니다.
}
```

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); }
}
```

□ 인터럽트 (attachInterrupt)

- ☞ 디지털신호의 입력을 받아 빠르게 처리해야 하는 사항이 있을 때 사용합니다.
- ☞ D2, D3, SDA(D20), SCL(D21) 포트로 총 4개의 인터럽트를 사용할 수 있습니다.
- ☞ 아날로그 출력(Y모듈), 고속펄스 출력(K모듈)을 추가하여 YK, Y2K, K2, Y3, Y4 모듈을 장착한 경우 D2, D3 인터럽트를 사용하지 않습니다.
- ☞ SDA(D20), SCL(D21) 인터럽트를 사용할 경우 I2C 통신은 사용할 수 없습니다.

☞ 관련 명령어

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

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

- ☞ D2 입력이 OFF이었다가 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");
}
```

□ 엔코더

- ☞ SDA(D20), SCL(D21)포트는 Open-Collector(오픈 콜렉터), Totem-Pole (토템폴), Push-Pull(푸시풀) 방식의 엔코더를 사용할 수 있습니다.
- ☞ D2, D3포트는 Totem-Pole(토템폴), Push-Pull(푸시풀) 방식의 엔코더를 사용할 수 있습니다.
- ☞ 아래 코드는 SDA(D20), SCL(D21) 포트를 사용하여 측정하는 예제입니다.

```
volatile long encoderValue = 0; // 엔코더 값 저장 변수
void setup() {
  Serial.begin(115200);
  attachInterrupt(digitalPinToInterrupt(SDA), updateEncoder, FALLING);
}
void loop() {
  Serial.print("Encoder Value: "); Serial.println(encoderValue);
  delay(500);
}
void updateEncoder() {
  if (digitalRead(SCL)) { encoderValue++; }
  else { encoderValue--; }
}
```

□ I2C 통신포트

- ☞ 1개의 I2C 통신포트를 제공합니다. 1:N 통신이 가능합니다.
- ☞ I2C 통신방법은 아래의 아두이노에서 제공하는 설명을 참고해 주시기 바랍니다. (<https://www.arduino.cc/reference/en/language/functions/communication/wire/>)
- ☞ I2C 통신으로 DS3231 RTC 모듈을 제어하는 예제를 참고해 주시기 바랍니다. (<https://blog.naver.com/ilogics/221806211175>)

□ 캐릭터 LCD 연결

- ☞ I2C 통신포트에 캐릭터 LCD를 연결하여 디스플레이를 구현할 수 있습니다.
- ☞ 아두이노 IDE에서 라이브러리 관리에서 "TM1637"로 검색하여 다양한 라이브러리를 사용할 수 있습니다.
- ☞ 저희가 제공하는 통합라이브러리인 "ILIB"를 사용할 수 있습니다.
- ☞ ILIB로 캐릭터 LCD를 제어하는 예제를 참고해 주시기 바랍니다. (<https://blog.naver.com/ilogics/222451135999>)

□ 7세그먼트 연결

- ☞ UART 포트로 1개의 7세그먼트 연결이 가능합니다.
- ☞ K(펄스출력 6포인트)를 추가하여 최대 4개의 7세그먼트 연결이 가능합니다.
- ☞ K2(펄스출력 12포인트)를 추가하여 최대 7개의 7세그먼트 연결이 가능합니다.
- ☞ 7세그먼트는 숫자, 영문, 특수문자 4자리수 표현이 가능합니다.
- ☞ K(펄스출력모듈)을 통하여 7세그먼트를 제어하는 예제를 참고해 주시기 바랍니다. UART 통신포트로 연결할 경우, D14와 D15로 사용해야 합니다. (<https://blog.naver.com/ilogics/222450896141>)



< 캐릭터 LCD 연결 >

< 7세그먼트 연결 >

□ 디버깅

- ☞ Debug는 Serial 함수를 이용해 주세요.

```
void setup() {
  Serial.begin(9600); // 업로드 포트를 보레이트가 9600인 시리얼포트로 정의
  for(int i = 30; i <=37; i++) {
    pinMode(i, OUTPUT);
  }
}
void loop() {
  // 디지털입력이 ON 되면 D30을 ON, OFF일 때 D30을 OFF
  for (int k = 0; k < 8; k++) {
    if (digitalRead(2 + k)==1) digitalWrite(30 + k, 1);
    else digitalWrite(30 + k, 0);
  }
  // 디지털입력 D2가 HIGH일 경우 디버깅 포트로 "D2 HIGH"를 전송
  // 디지털입력 D2가 LOW일 경우 디버깅 포트로 "D2 LOW"를 전송
  if (digitalRead(2) == 1) Serial.println("D2 HIGH");
  else if (digitalRead(2) == 0) Serial.println("D2 LOW");
  delay(500);
}
```

□ 시리얼 통신포트

통신 키워드	RS-232 Serial1	RS-485 Serial2	UART Serial3
--------	----------------	----------------	--------------

- ☞ 시리얼 통신방법은 아래의 아두이노에서 제공하는 설명을 참고해 주시기 바랍니다. ([아두이노 IDE의 Serial 함수 사용설명서](#))
- ☞ 산업현장에서는 산업 범용 프로토콜인 모드버스 통신 프로토콜을 주로 사용합니다.
- ☞ ILIB로 Modbus RTU Master를 사용하는 예제를 참고해 주시기 바랍니다. (<https://blog.naver.com/ilogics/222453991523>)
- ☞ ILIB로 Modbus RTU Slave를 사용하는 예제를 참고해 주시기 바랍니다. (<https://blog.naver.com/ilogics/222453993604>)

□ MPINO STUDIO

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

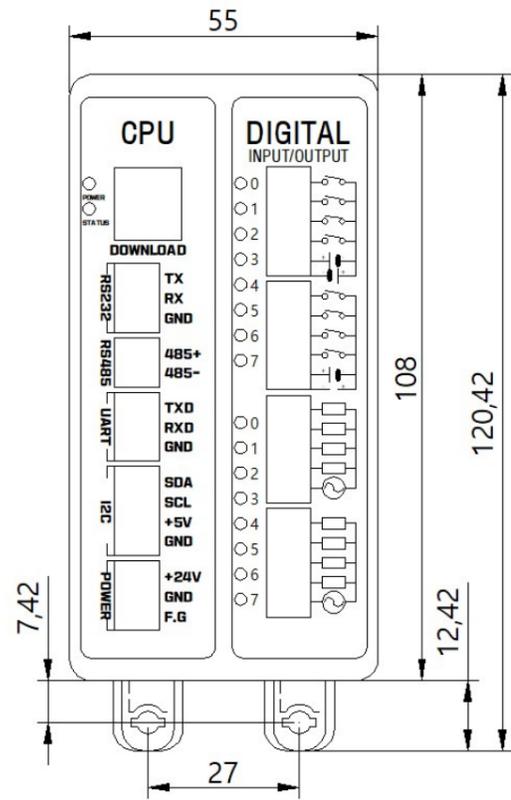
□ MP STUDIO

- ☞ 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.

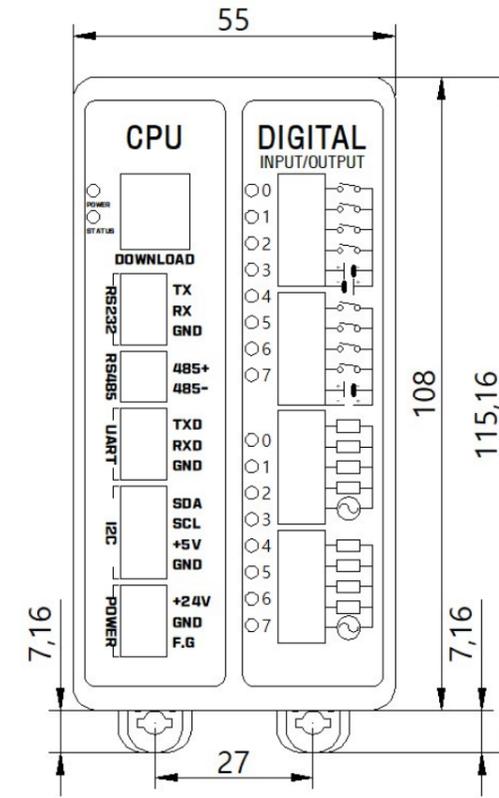
□ 감사드립니다.

- ☞ 저희 (주)아이로직스의 제품을 구매해주셔서 감사드립니다.
- ☞ 구매는 <https://www.ilogics.co.kr> 쇼핑몰에서 하실 수 있습니다.
- ☞ 구매/기술 상담은 0507-1362-5020으로 전화 주시기 바랍니다. (상담시간은 오전10시~오후5시 입니다)

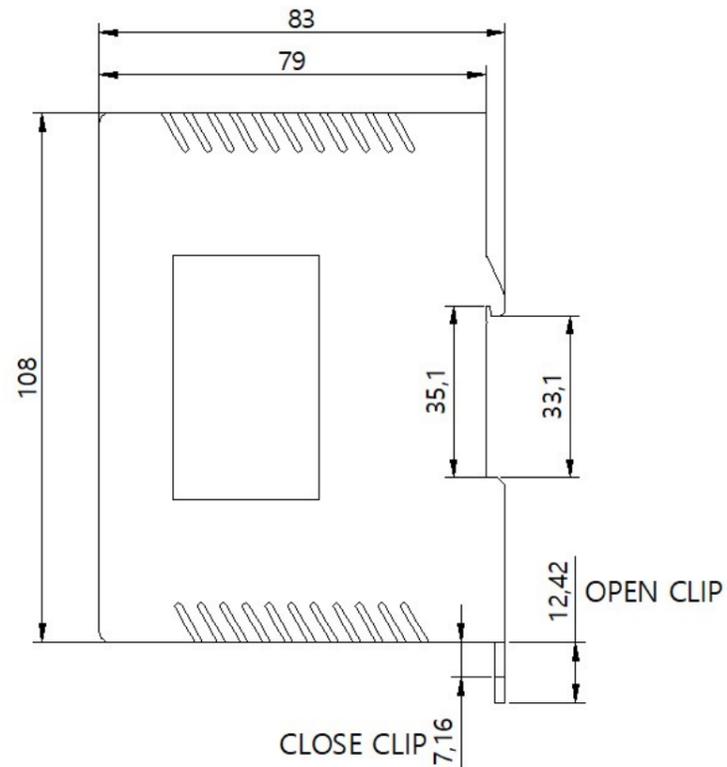
□ DIMENSION (클립 열었을 경우)



□ DIMENSION (클립 닫았을 경우)



□ DIMENSION (클립 열었을 때 / 딘 레일 체결 후)



□ DIMENSION (딘 레일 : 35mm)

