

MPAINO SERIES

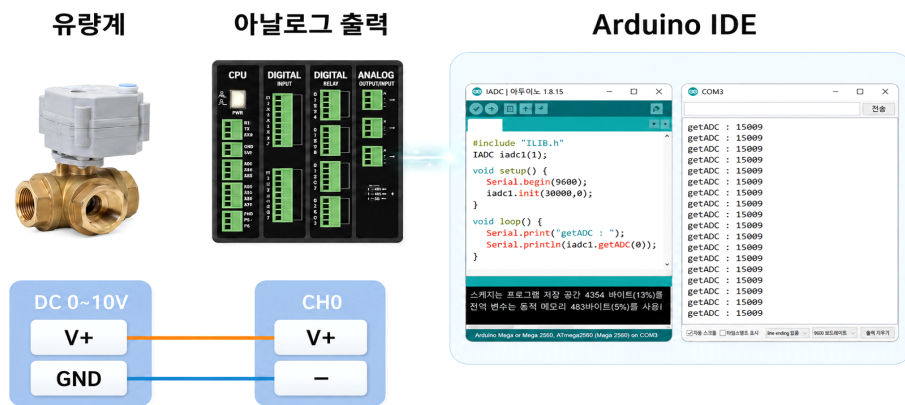
아날로그 출력 사용설명서



저희 (주)아이로직스 제품을 구입해 주셔서 감사합니다.

□ 개요

- 디지털 값을 아날로그 값으로 변환하여 아날로그 값이 필요한 구동부나 액추에이터를 동작시킬 수 있도록 Arduino IDE 또는 MPINO STUDIO에서 프로그램 할 수 있습니다.



- 유량계 이외에 DC 0~5V, DC 0~10V, DC 0~20mA, DC 4~20mA의 아날로그 신호를 입력받는 기기와 연결하여 사용할 수 있습니다.

- 전원 전압 없이 다운로드포트로만 전원 공급 시 정상작동이 되지 않으므로 반드시 12~24V 전원공급을 하여 사용합니다.

□ 아날로그 출력 사양서

모델명	개 수	모듈 수	채널번호	설 명
Y	3 포인트	1	0 ~ 2	• 각 채널당 아래의 기능을 선택적 사용 - DC 0 ~ 5V / DC 0 ~ 10mA - DC 0 ~ 10V / DC 0 ~ 20mA
Y2	6 포인트	2	0 ~ 5	• 정밀도 : ± 0.5% 이하 (전원전압 DC12V, Half Range 기준)
Y3	9 포인트	3	0 ~ 8	
Y4	12 포인트	4	0 ~ 11	분해능 : 16Bit

- 아날로그 출력은 모델명에 Y, Y2, Y3, Y4가 붙습니다.
- 아날로그 출력은 한 개의 모듈당 3포인트를 사용할 수 있습니다.
- 아날로그 출력 모듈은 최대 4개의 모듈을 사용할 수 있습니다.

- 아날로그 출력과 고속펄스 출력은 통합 12포인트를 초과할 수 없습니다. (고속펄스 출력 K(6포인트)를 사용할 경우 아날로그 출력은 Y2(6포인트)까지 사용하실 수 있습니다)

- MPAINO-8A8R^Y : 아날로그출력 3포인트,
- MPAINO-8A8R^{Y2} : 아날로그출력 6포인트
- MPAINO-16A16R^{Y3} : 아날로그출력 9포인트
- MPAINO-16A16R^{Y4} : 아날로그출력 12포인트

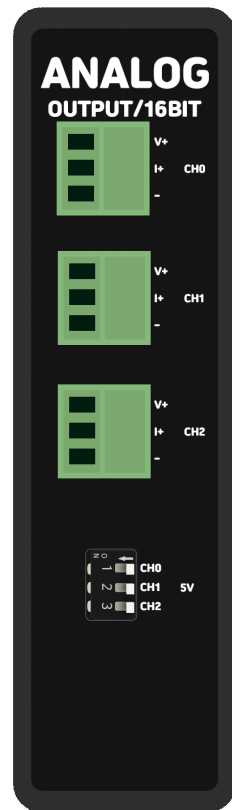
□ 아날로그 출력 범위

- 포인트 별로 DC 0~5V, DC 0~10V, DC 0~20mA, DC 4~20mA의 아날로그 신호를 출력 할 수 있습니다.

□ 아날로그 출력 범위 설정

- 아날로그 출력모듈 전면에 위치한 딥스위치로 설정할 수 있습니다.

- 전압은 V+단자에 전류는 I+단자에 전압과 전류가 동시에 출력됩니다.



딥스위치	전압출력	전류출력
ON	DC 0~5V	DC 0~10mA, DC 2~10mA
OFF	DC 0~10V	DC 0~20mA DC 4~20mA

□ 옵션모듈 선택

- Arduino IDE에서 도구(Tools) -> Analog Output Module에서 아날로그 출력 옵션 모듈을 반드시 선택해야 합니다.

□ 아날로그 출력값 쓰기 (DC 0~5V, 0~10V, 0~20mA)

- Board를 MPAINO 제품군으로 선택하였다면, analogWrite(0~11) 함수를 사용하여 아날로그 값을 16Bit(0~65,535)로 출력할 수 있습니다.
- 아날로그 출력모듈에서 첫 번째 모듈은 analogWrite(0~2)를 사용합니다.
- 아날로그 출력모듈에서 두 번째 모듈은 analogWrite(3~5)를 사용합니다.

```
void setup() {  
}  
void loop() {  
  if (digitalRead(0) == 1) analogWrite(0, 32767); // 채널0에 50%출력  
  else analogWrite(0, 0); // 채널0에 0%출력  
}
```

- lscale 함수를 사용해서 센서의 값으로 변경할 수 있습니다.
- int32_t lscale(int32_t x, int32_t in_min, int32_t in_max, int32_t out_min, int32_t out_max)
- float lscalef(float x, float in_min, float in_max, float out_min, float out_max)

```
void setup() {  
}  
void loop() {  
  int16_t percent = 50;  
  uint16_t outValue = lscale(percent, 0, 100, 0, 65535);  
  // 0~100%값을 0~65,535로 변환하여 outValue에 대입.  
  analogWrite(1, outValue); // 채널1에 50%출력  
}
```

- lanalogWrite 함수를 사용하여 scale 처리하여 아날로그 값을 출력할 수 있습니다.
- void lanalogWrite(uint8_t ch, int32_t min, int32_t max, int32_t value)
- void lanalogWritef(uint8_t channel, float min, float max, float value)

```
void setup() {  
}  
void loop() {  
  int16_t percent = 50;  
  lanalogWrite(0, 0, 100, percent); // 0~100% 값 범위에서 50%의 값만큼 출력.  
  lanalogWritef(0, 0.0, 100.0, 50.0); // 0.0~100.0% 값 범위에서 50.0%의 값만큼 출력.  
}
```

□ 아날로그 출력값 쓰기 (4~20mA)

- ☞ **analogWrite2(0~11)** 함수를 사용하여 아날로그 값을 16Bit(0~65,535)로 출력할 수 있습니다.
- ☞ void analogWrite2(uint8_t channel, uint16_t value)
- ☞ 아날로그 출력모듈에서 첫 번째 모듈은 analogWrite2(0~2)를 사용합니다.
- ☞ 아날로그 출력모듈에서 두 번째 모듈은 analogWrite2(3~5)를 사용합니다.

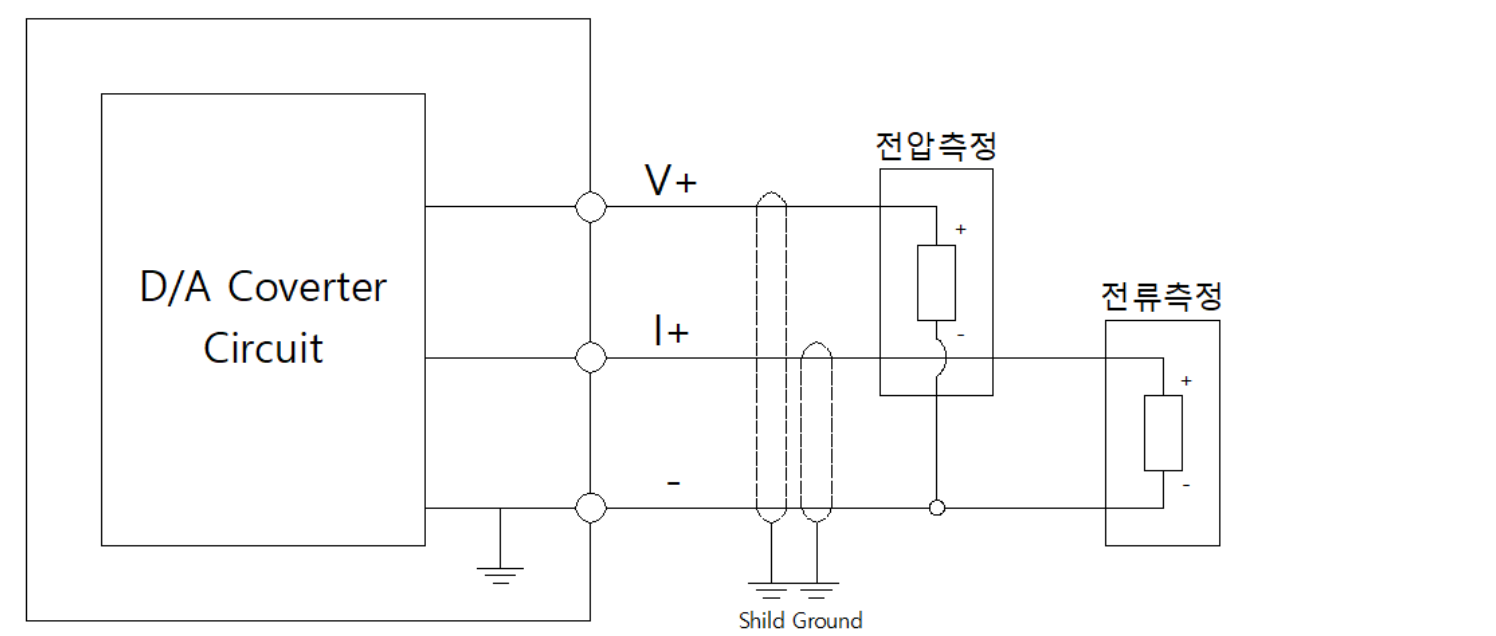
```
void setup() {  
}  
  
void loop() {  
    if (digitalRead(0) == 1) analogWrite2(0, 32767); // 채널0에 12mA(50%)출력  
    else analogWrite2(0, 0); // 채널0에 4mA(0%)출력  
}
```

- ☞ **lanalogWrite2** 함수를 사용하여 scale 처리하여 아날로그 값을 출력할 수 있습니다.
- ☞ void lanalogWrite2(uint8_t ch, int32_t min, int32_t max, int32_t value)
- ☞ void lanalogWrite2f(uint8_t channel, float min, float max, float value)

```
void setup() {  
}  
  
void loop() {  
    int16_t percent = 50;  
    lanalogWrite2(0, 0, 100, percent); // 0~100% 값 범위에서 50%의 값만큼 출력.  
    lanalogWrite2f(0, 0.0, 100.0, 50.0); // 0.0~100.0% 값 범위에서 50.0%의 값만큼  
    출력.  
}
```

□ 결선도

- ☞ 아날로그 출력의 (-) 단자는 제품 내부의 GND(Ground)와 연결되어 있습니다.
- ☞ 아날로그 신호는 노이즈에 취약하므로 정밀도가 요구되는 시스템에는 차폐(실드) 케이블을 사용을 권장드리며, 실드선은 GND(Ground)와 연결해 주시기 바랍니다.



□ 함수 리스트

사용 목적	권장 함수
16비트 원시 출력	analogWrite()
정수 사용자 범위 출력	lanalogWrite()
실수 사용자 범위 출력	lanalogWritef()
4~20mA 16비트 원시 출력	analogWrite2()
정수 사용자 범위의 4~20mA 출력	lanalogWrite2()
실수 사용자 범위의 4~20mA 출력	lanalogWrite2f()