

MPAINO SERIES

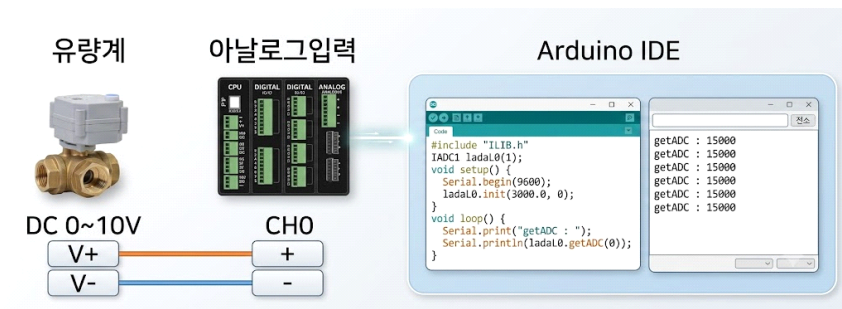
아날로그 입력 사용설명서

저희 (주)아이로직스 제품을 구입해 주셔서 감사합니다.

□ 개요

- 센서 등에서 출력되는 아날로그 전기 신호를 디지털 값으로 변환하여 Arduino IDE에서 프로그램 할 수 있습니다.

- 예를들어 0~30,000L/min 범위를 측정하는 유량센서가 유량값이 DC 0~10V로 출력된다면 이 출력을 입력받아 Arduino IDE에서 C언어 프로그램으로 제어할 수 있습니다. 아래 그림은 현재 15,009L/Min의 유량값이 측정되고 있습니다.



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

- 아날로그 입력 기능이외에 NTC 3950 써미스터 온도센서 입력기능을 지원합니다. 분해능은 0.1℃입니다.

□ 아날로그 입력 사양서

모델명	개 수	설 명
X	4 포인트	• 각 채널당 아래의 기능을 선택적 사용 - DC 0 ~ 5V / DC 1 ~ 5V - DC 0 ~ 10V - 0 ~ 20mA / 4 ~ 20mA - NTC 써미스터 온도센서
X2	8 포인트	
X3	12 포인트	
X4	16 포인트	
X5	20 포인트	• 해상도 : 15비트 (0 ~ 32,767)

- 아날로그 입력은 모델명에 X, X2, X3, X4, X5가 붙습니다.

- 아날로그 입력은 한 개의 모듈당 4포인트를 사용하실 수 있습니다.

- 아날로그 입력 모듈은 최대 5개의 모듈을 사용하실 수 있습니다. 따라서, 최대 20포인트까지 아날로그 입력을 사용하실 수 있습니다.

- MPAINO-8A8R**X** : 아날로그입력 4포인트,
- MPAINO-16A16R**X2** : 아날로그입력 8포인트
- MPAINO-16A32R**X3Y** : 아날로그입력 12포인트

□ 아날로그 입력 범위

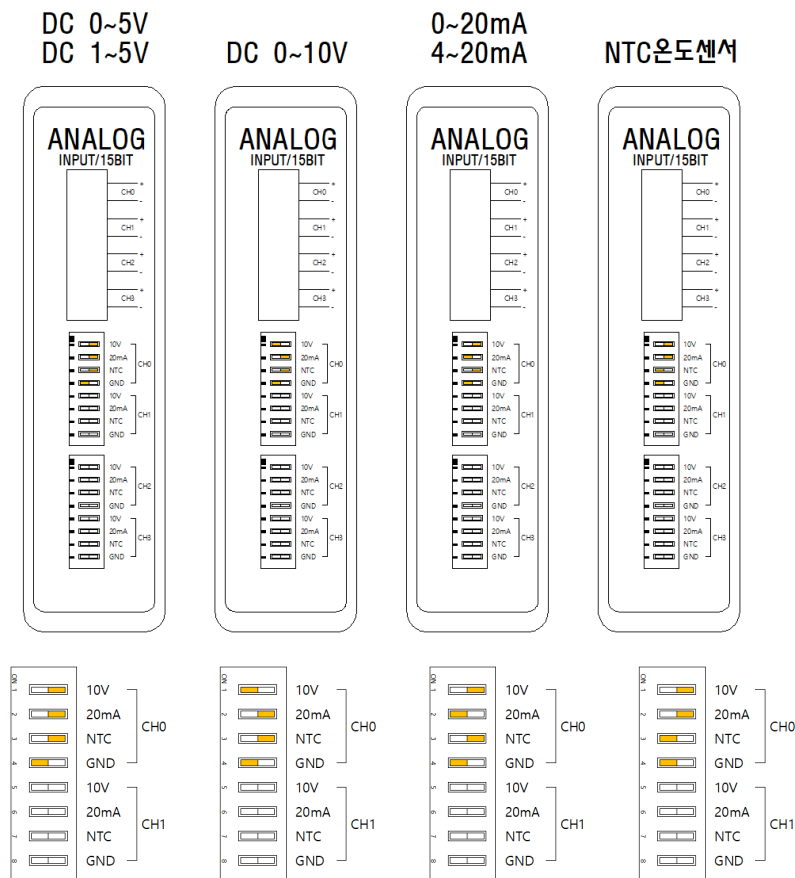
- 포인트 별로 DC 0~5V, DC 0~10V, DC 0~20mA, DC 4~20mA의 아날로그 신호를 입력받을 수 있습니다.

- 포인트 별로 NTC 3950 서미스터 온도센서를 연결하여 온도값을 측정할 수도 있습니다.

- 그 외, DC 0~1V가 출력되는 센서를 연결할 경우, DC 0~5V의 1/5이므로 아날로그 입력 분해능 0~32,767의 1/5인 0~6,553의 분해능으로 사용하실 수도 있습니다.

□ 아날로그 입력 설정

- 아날로그 입력범위는 아날로그 입력모듈 전면에 위치한 디스위치로 변경하실 수 있습니다.



- 디스위치 설정은 예시로 채널0만 표시하였습니다.

- GND 디스위치는 기본 ON상태로 출고되며, 항상 ON이어야 합니다.

□ 옵션모듈 선택

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

□ 아날로그 입력값 읽기 (DC 0~5V, 0~10V, 0~20mA)

- Board를 MPAINO 제품군으로 선택하셨다면, **analogRead(0~19)** 함수를 사용하여 아날로그 값을 15Bit(0~32,767)로 읽어올 수 있습니다.
- 아날로그 입력모듈에서 첫 번째 모듈은 analogRead(0~3)을 사용합니다.
- 아날로그 입력모듈에서 두 번째 모듈은 analogRead(4~7)을 사용합니다.

```
void setup() {
    Serial.begin(115200);
}

void loop() {
    Serial.print("RawValue: ");
    Serial.println(analogRead(0)); //0~32,767 값으로 표시합니다.
}
```

- 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 lscafe(float x, float in_min, float in_max, float out_min, float out_max)

```
void setup() {
    Serial.begin(115200);
}

void loop() {
    int16_t raw = analogRead(0);
    // raw값(0~32,767)을 0~100%로 변환하여 percent에 대입.
    int16_t percent = lscale(raw, 0, 32767, 0, 100);
    Serial.print("Percent: ");
    Serial.println(percent); //0~100 값으로 표시합니다.
}
```

- lanalogRead** 함수를 사용하여 아날로그 값을 Scale 처리하여 한번에 읽어올 수 있습니다.
- int32_t lanalogRead(uint8_t ch, int32_t min, int32_t max);
- float lanalogReadf(uint8_t ch, float min, float max);

```
void setup() {
    Serial.begin(115200);
}

void loop() {
    int32_t percent = lanalogRead(0, 0, 100);
    // raw값(0~32,767)을 0~100%로 변환하여 percent에 대입.
    Serial.print("Percent: ");
    Serial.println(percent); //0~100 값으로 표시합니다.
}
```

□ 아날로그 입력값 평균처리하여 읽기

- ☞ **analogReadAvg** 함수를 사용해서 이동 평균처리하여 값을 읽을 수 있습니다.
- ☞ analogReadAvg(채널, 샘플수)로 사용됩니다. ex) analogReadAvg(0, 10);
- ☞ int32_t analogReadAvg(uint8_t ch, uint8_t samples)
- ☞ float analogReadAvgf(uint8_t ch, uint8_t samples)
- ☞ 샘플수는 0 ~ 20을 입력할 수 있습니다. 0일 경우, 평균 처리 없이 값을 반환합니다.

```
void loop() {
  Serial.print("RawValue1: ");
  Serial.println(analogReadAvg(0, 10)); //10회 평균 처리하여 0~32,767 값으로 표시.
  Serial.print("RawValue2: ");
  Serial.println(analogReadAvgf(0, 10)); //10회 평균 처리하여 0.0~32,767.0 값으로 표시.
}
```

□ 아날로그 입력값 읽기 (4~20mA)

- ☞ **analogRead2** 함수를 사용해서 아날로그 값을 15bit(0~32,767)로 읽을 수 있습니다.
- ☞ 4~20mA 입력은 ADC 전체 측정 범위의 80%를 사용하며, 함수 반환값은 0~32,767 범위로 재조정됩니다.
- ☞ 3.6mA 미만으로 입력될 경우, 오류로 판정하여 65,535 값을 반환합니다.
- ☞ 3.6mA 이상 4mA 이하에서는 0을 반환합니다.
- ☞ 아날로그 입력모듈에서 첫 번째 모듈은 analogRead2(0~3)을 사용합니다.
- ☞ 아날로그 입력모듈에서 두 번째 모듈은 analogRead2(4~7)을 사용합니다.

```
void loop() {
  Serial.print("RawValue: ");
  Serial.println(analogRead2(0)); //0~32,767 값으로 표시합니다.
}
```

- ☞ **lanalogRead2** 함수를 사용하여 아날로그 값을 Scale 처리하여 한번에 읽어올 수 있습니다.
- ☞ int32_t lanalogRead2(uint8_t ch, int32_t min, int32_t max);
- ☞ float lanalogRead2f(uint8_t ch, float min, float max);

```
void setup() {
  Serial.begin(115200);
}

void loop() {
  int32_t percent = lanalogRead2(0, 0, 100);
  // raw값(0~32,767)을 0~100%로 변환하여 percent에 대입.
  Serial.print("Percent: ");
  Serial.println(percent); //0~100 값으로 표시합니다.
}
```

- ☞ **analogRead2Avg** 함수를 사용하여 아날로그 값을 평균처리하여 읽어올 수 있습니다.
- ☞ int32_t analogRead2Avg(uint8_t ch, uint8_t samples)
- ☞ float analogRead2Avgf(uint8_t ch, uint8_t samples)

```
void loop() {
  Serial.print("RawValue1: ");
  Serial.println(analogRead2Avg(0, 10)); //10회 평균 처리하여 0~32,767 값으로 표시.
  Serial.print("RawValue2: ");
  Serial.println(analogRead2Avgf(0, 10)); //10회 평균 처리하여 0.0~32,767.0 값으로 표시.
}
```

□ 아날로그 입력 변환속도 변경

- ☞ **analogReadInit** 함수를 사용해서 아날로그 입력 변환 속도를 변경할 수 있습니다.
- ☞ analogReadInit(0~475SPS)를 지정할 수 있습니다. 기본은 64 SPS입니다.
- ☞ SPS 단계는 8,16,32,64,128,250,475 SPS입니다. 만약, analogReadInit(7)을 사용할 경우 올림해서 analogReadInit(8)이 수행됩니다.
- ☞ 일반적으로 setup()에서 한 번 설정하며, loop()에서 반복 호출하지 마십시오.
- ☞ 변환 속도를 빠르게 하면 Ripple이 발생할 수 있는 단점이 있습니다.

```
void setup() {
  analogReadInit(128); // 128SPS로 변경합니다.
}
```

□ 온도값 읽기

- ☞ **ntcRead** 함수를 사용해서 NTC 온도센서의 온도값을 -40℃ ~ 120℃까지 읽어올 수 있습니다.
- ☞ int ntcRead(uint8_t ch)
- ☞ float ntcReadf(uint8_t ch)
- ☞ NTC 저항-온도표에서 인접한 두 지점 사이를 선형 보간하여 0.1℃ 단위의 온도를 반환합니다.
- ☞ ntcRead 함수는 -40℃는 -400 , 120℃는 1200으로 함수에서 반환됩니다.
- ☞ ntcReadf 함수는 -40℃는 -40.0 , 120℃는 120.0으로 함수에서 반환됩니다.
- ☞ 아날로그 입력모듈에서 첫 번째 모듈은 ntcRead(0~3)을 사용합니다.
- ☞ 아날로그 입력모듈에서 두 번째 모듈은 ntcRead(4~7)을 사용합니다.

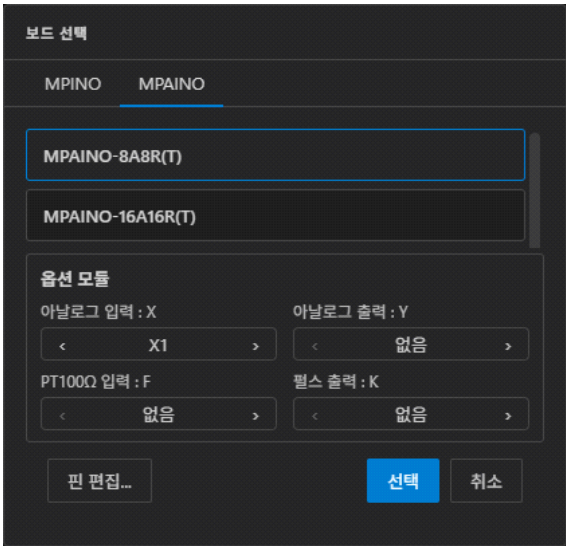
```
void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.print("RawValue2: ");
  Serial.println(ntcRead(0)); // 0채널 온도값을 -400~1200으로 표시합니다.
  Serial.println(ntcReadf(1)); // 1채널 온도값을 -40.0~120.0으로 표시합니다.
}
```

- ☞ 측정 범위를 벗어난 ADC 값은 -40.0℃ 또는 120.0℃로 제한됩니다.

□ MPINO STUDIO2 사용시

- ☞ 새파일 또는 보드변경으로 보드설정에서 보드를 선택하고 옵션모듈에서 아날로그 입력을 X1 ~ X5로 선택합니다.



- ☞ 핀 편집 버튼을 눌러서 아날로그 입력의 변수를 변경할 수 있습니다.

17	C0	Ain_0	AI0	IN/ANALOG	☒
18	C1	Ain_1	AI1	IN/ANALOG	☒
19	C2	Ain_2	AI2	IN/ANALOG	☒
20	C3	Ain_3	AI3	IN/ANALOG	☒

- ☞ 기본적으로 채널0~채널19는 C0~C19로 설정되어 있습니다.
- ☞ 메모리는 C메모리 또는 D메모리를 사용할 수 있습니다.
- ☞ 사용 컬럼의 체크박스를 해제하면 자동으로 변환되지 않습니다.

- ☞ 코드에 아래와 같이 프로그램 합니다.

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  ladderLoop();
  Serial.print("CH0 analog Input: ");
  Serial.println(C0);
}
```

- ☞ 아날로그 값은 raw값인 0~32,767의 값으로 저장됩니다.
 - ☞ DC 0~5V 아날로그 입력설정에서 DC 2.5V를 인가하면, 시리얼 모니터는 아래와 같이 표시됩니다.
- CH0 analog Input: 16383