

MPAINO SERIES

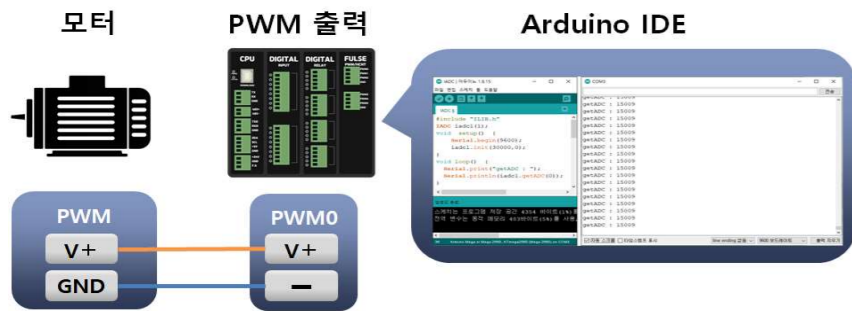
펄스 출력 사용설명서



저희 (주)아이로직스 제품을 구입해 주셔서 감사합니다.

□ 개요

- 디지털 값을 PWM 값으로 변환하여 PWM 값이 필요한 모터나 센서를 동작시킬 수 있도록 Arduino IDE에서 프로그램 할 수 있습니다.
- 예를 들어 주파수로 제어하는 모터가 있다면 Arduino IDE에서 C언어 프로그램으로 사용자가 입력한 값만큼 PWM 주파수 값을 조절하여 동작 할 수 있도록 제어할 수 있습니다.



모터 이외에 PWM 아날로그 신호 값을 입력받는 기기와 연결하여 사용할 수 있습니다.

□ PWM 출력 사양서

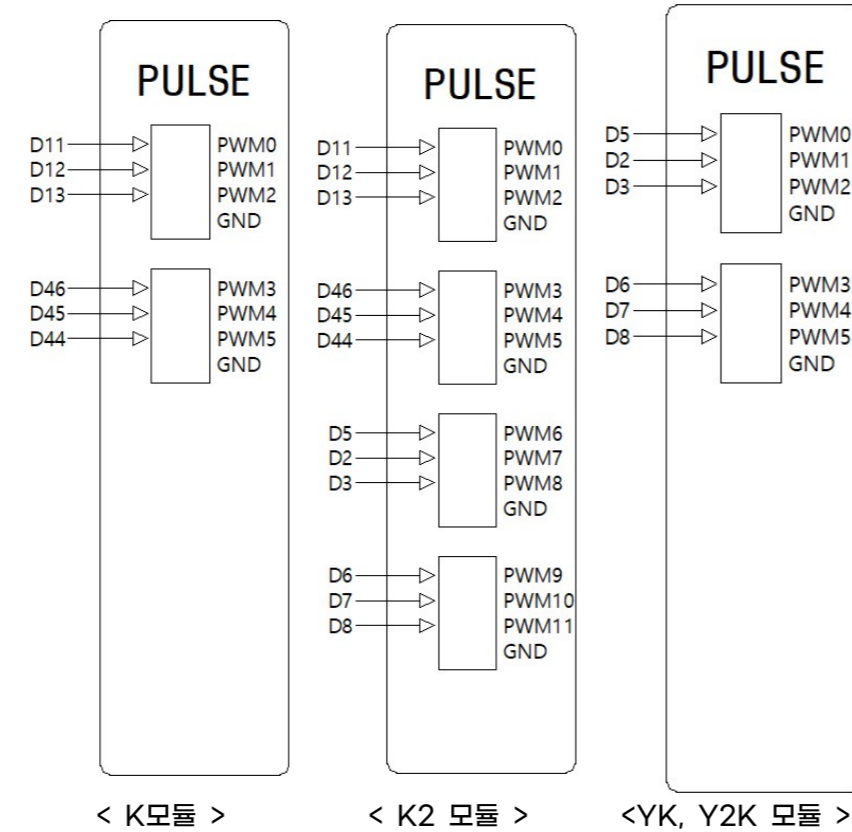
- DC 0(Low) ~ 5V(High) / 최대 30mA 출력
- 최대 1MHz 출력
- 분해능 : 16Bit
- 펄스출력은 모델명에 K 또는 K2가 붙습니다.
- 아날로그출력은 모델명에 Y가 붙습니다.

| 모델명 | 개수 | PWM 채널번호 | 설명 |
|-----------|--------|----------|---|
| K | 6 포인트 | 0 ~ 5 | • D11 ~ D13 / GND • D46 ~ D44 / GND |
| K2 | 12 포인트 | 0 ~ 5 | • D11 ~ D13 / GND • D46 ~ D44 / GND • D5, D2, D3 / GND • D6 ~ D8 / GND |
| YK Y2K | 6포인트 | 0 ~ 5 | • D5, D2, D3 / GND • D6 ~ D8 / GND |

- 펄스출력은 한 개의 모듈당 6포인트를 사용할 수 있습니다.
- 펄스출력 모듈은 최대 2개의 모듈을 사용할 수 있습니다. 따라서, 최대 12포인트까지 아날로그 입력을 사용할 수 있습니다.

- MPAINO-8A8R-K : 펄스출력 6포인트
- MPAINO-16A16R-K2 : 펄스출력 12포인트

□ K모듈 포트번호



□ PWM 주의사항

- 포인트 별로 최대 1MHz 주파수 펄스 출력을 사용할 수 있습니다.
- 동일한 타이머번호는 동일한 주파수를 출력해야 합니다.
Ex) PWM0~PWM2는 같은 타이머 자원을 사용하므로 PWM0이 1kHz로 펄스를 출력한다면 PWM1, PWM2도 1kHz로 출력해 합니다.
- 고속출력을 위하여 별도의 I.C를 거치지 않고 MCU의 GPIO를 그대로 사용하므로 PWM 핀에 과전압, 과전류 유입 및 과전류 출력시 MCU가 손상될 수 있습니다.
- K모듈을 사용할 경우 Y, Y2모듈을 추가 연결할 수 있습니다.
- K2모듈을 사용할 경우 Y모듈을 추가 연결할 수 없습니다.

□ 통합 표

ILIB 핀번호는 저희 아이로직스에서 제공하는 통합 라이브러리(ILIB)를 사용할 때의 번호입니다.

| 모델명 | 타이머 번호 | 채널명 | 아두이노 핀번호 | ILIB 핀번호 |
|---------|-----------|---------|----------|----------|
| K | Timer 1 | PWM0 | 11 | 0 |
| | | PWM1 | 12 | 1 |
| | | PWM2 | 13 | 2 |
| | Timer 5 | PWM3 | 46 | 3 |
| | | PWM4 | 45 | 4 |
| PWM5 | | 44 | 5 | |
| K2 | Timer 3 | PWM0 | 11 | 0 |
| | | PWM1 | 12 | 1 |
| | | PWM2 | 13 | 2 |
| | Timer 4 | PWM3 | 46 | 3 |
| | | PWM4 | 45 | 4 |
| | | PWM5 | 44 | 5 |
| | Timer 1 | PWM6 | 5 | 6 |
| | | PWM7 | 2 | 7 |
| | | PWM8 | 3 | 8 |
| | | PWM9 | 6 | 9 |
| | | PWM10 | 7 | 10 |
| Timer 5 | PWM11 | 8 | 11 | |
| | YK Y2K | Timer 3 | PWM0 | 5 |
| PWM1 | | | 2 | 7 |
| PWM2 | | | 3 | 8 |
| Timer 4 | | PWM3 | 6 | 9 |
| | | PWM4 | 7 | 10 |
| PWM5 | 8 | 11 | | |

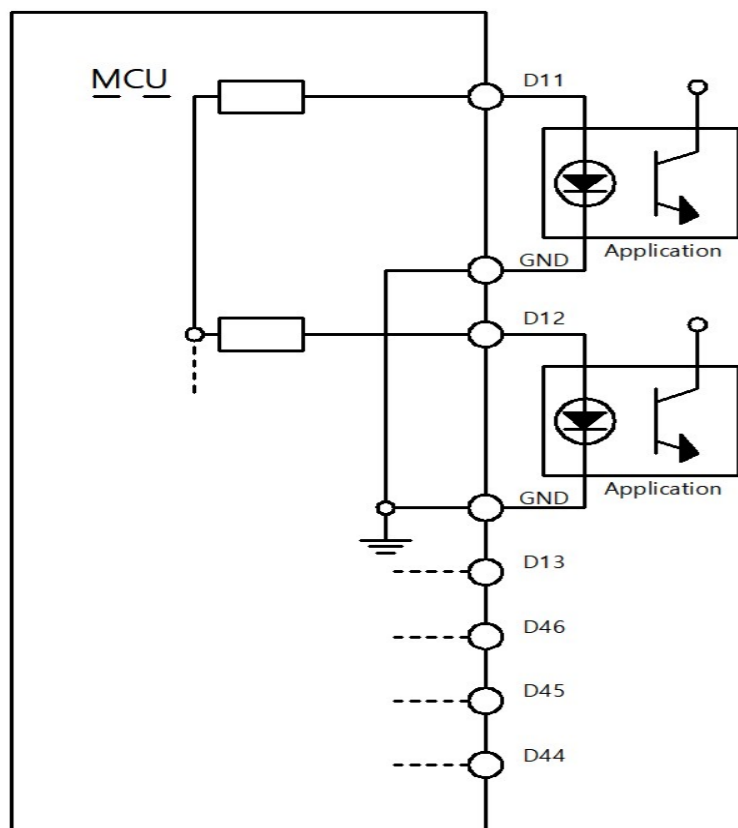
□ 디지털입력 포트 변경되는 사례

MPAINO 제품군에 Y2K, K2 모듈이 추가되면 입력단자가 D22 ~ D29입니다.

□ 펄스 출력 결선도

☞ 출력접점 D11 ~ D13, D44 ~ D46 핀을 0V, 5V로 최대 1MHz로 펄스 출력을 할 수 있습니다.

☞ 출력접점 D11 ~ D13, D44 ~ D46 핀은 GPIO핀으로 활용하실 수 있습니다.



□ 각 핀별 주파수 출력 예제

```
void setup() {
  pinMode(11, OUTPUT); // D11 핀을 출력모드로 설정합니다.
  pinMode(12, OUTPUT); // D12 핀을 출력모드로 설정합니다.
  pinMode(13, OUTPUT); // D13 핀을 출력모드로 설정합니다.
}

void loop() {
  //D11핀을 10/255값 만큼 PWM 파형을 출력합니다.
  analogWrite(11, 10);
  //D12핀을 127/255값 만큼 PWM 파형을 출력합니다.
  analogWrite(12, 127);
  //D13핀을 255/255값 만큼 PWM 파형을 출력합니다.
  analogWrite(13, 255);
}
```

☞ 자사 블로그에 업로드 되어 있는 통합라이브러리 ILIB.h를 사용하시면 다양한 PWM 파형을 출력하실 수 있습니다.

□ PWM 파형 출력

☞ analogWrite() 함수와 동일하지만 16비트 PWM을 출력합니다.

```
#include "ILIB.h"
void setup() {
}

void loop() {
  //PWM(uint8_t pin, int val, uint8_t onDutybit16);
  /*PWM(출력할 핀번호, 듀티비,
  듀티비 16비트로 변경시 HIGH(미변경시 false 및 생략);*/
  //11번핀을 40% 듀티비로 PWM출력
  PWM(0, 65535*0.4, 1);
}
```

□ 주파수 가변(FDPWM) 출력

☞ 주파수와 듀티비를 조절할 수 있는 16비트 PWM을 출력합니다.

```
#include "ILIB.h"
void setup() {
  PWM_RESET();
}

void loop() {
  //FDPWM(PIN, Hz, Duty);
  FDPWM(0, 4000, 40);
  FDPWM(1, 4000, 60);
  /*(D11, D12, D13), (D44, D45, D46), (D2, D3, D5), (D6, D7, D8)
  각 그룹별로 주파수를 동일하게 설정해야 정상동작합니다.*/
}
```

□ n개 펄스(NPWM) 출력

☞ 주파수 및 듀티비를 조절할 수 있는 n개의 16비트 펄스를 출력합니다.

```
#include "ILIB.h"
int SerialInput = 0;
void setup() {
  PWM_RESET();
}

void loop() {
  if (Serial.available() > 0) {
    SerialInput = Serial.read();
  }

  if (SerialInput == 49) { //시리얼 모니터에 1번 입력시
  //PWM D11번핀 200Hz, Duty비 30%, 4번 출력
  NPWM_BEGIN(11, 200, 30, 4);
  SerialInput = 0;
  }

  NPWM(11);
  /*(D11, D12, D13), (D44, D45, D46), (D2, D3, D5), (D6, D7, D8)
  각 그룹별로 주파수를 동일하게 설정해야 정상동작합니다.*/
}
```

□ GPIO 단자 사용

☞ K모듈의 포트는 GPIO로 사용이 가능합니다.

☞ 저렴한 센서등에 GPIO로 연결이 가능합니다.

☞ 예시로 DHT온습도 센서를 GPIO 단자선으로 연결하여 온습도 데이터를 받아오실 수 있습니다. (<https://blog.naver.com/ilogics/222451130930>)

□ SoftwareSerial 사용

☞ PWM(GPIO)핀으로 SoftwareSerial을 최대 3개 사용하실 수 있습니다.

☞ SoftwareSerial기능의 RX핀은 11, 12, 13핀만 가능합니다.

☞ TX는 PWM(GPIO)핀 중에 제한없이 사용이 가능합니다.

```
#include <SoftwareSerial.h>

#define rxPin 11
#define txPin 46

// Set up a new SoftwareSerial object
SoftwareSerial mySerial = SoftwareSerial(rxPin, txPin);
void setup() {
  // Define pin modes for TX and RX
  pinMode(rxPin, INPUT);
  pinMode(txPin, OUTPUT);

  // Set the baud rate for the SoftwareSerial object
  mySerial.begin(9600);
}

void loop() {
  // ...
}
```