

MPAINO SERIES

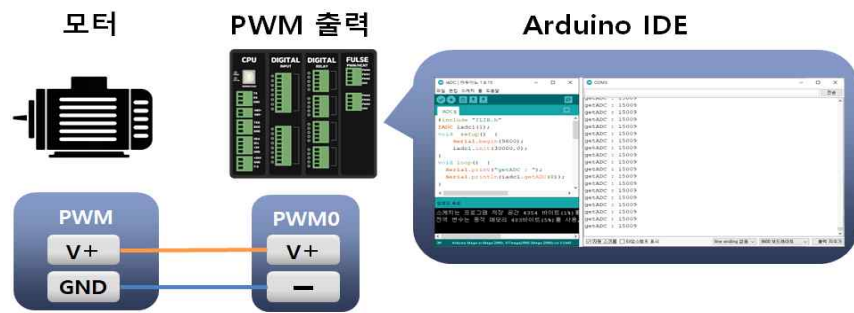
펄스 출력 사용설명서



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

모델명	개수	PWM 채널번호	설명
K	6 포인트	0 ~ 5	<ul style="list-style-type: none"> D11 ~ D13 / COM0 D44 ~ D46 / COM1
K2	12 포인트	0 ~ 5	<ul style="list-style-type: none"> D2, D3, D5 / COM0 D6 ~ D8 / COM1 <p>• 최대 1MHz 출력 분해능 : 16Bit</p>

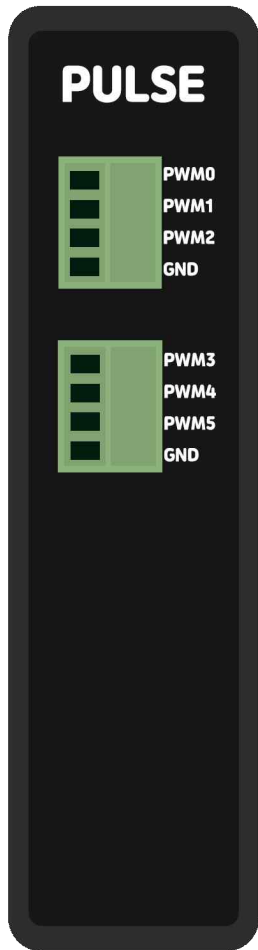
- 펄스출력은 모델명에 K, K2가 붙습니다.

- 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-8A8RK : 펄스출력 6포인트
- MPAINO-16A16RK2 : 펄스출력 12포인트

□ PWM 범위

- 포인트 별로 최대 1MHz 주파수 펄스 출력을 사용할 수 있습니다.
- 타이머번호가 같은 채널은 같은 주파수를 설정하여야 정상적으로 출력합니다. (타이머번호는 K 단자 번호 표에서 확인하실 수 있습니다.)



□ K 단자 번호

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

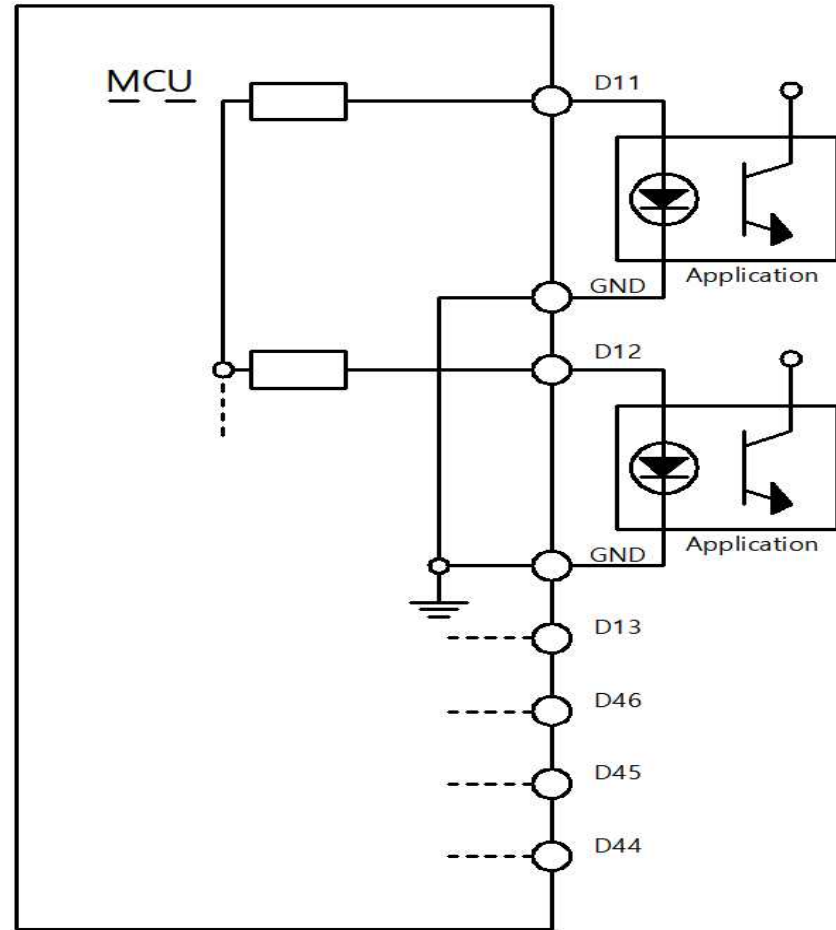
□ 입력 포트

- 펄스출력 모듈 개수에 따라 디지털 입력 핀 번호가 변경됩니다.
- MPAINO 제품군에 K 모듈이 추가되면 입력단자가 D2 ~ D9입니다.
- MPAINO 제품군에 Y2K, K2 모듈이 추가되면 입력단자가 D22 ~ D29입니다.
- K모듈을 1개 장착할 경우 Y, Y2모듈만 장착 할 수 있습니다.
- K모듈을 2개 장착할 경우 Y모듈을 장착 할 수 없습니다.

□ 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 사용하면 다양한 PWM 파형을 출력하실 수 있습니다.

□ PWM 파형 출력하기

```
#include "ILIB.h"
void setup() {
}
void loop() {
  //PWM(uint8_t pin, int val, uint8_t onDutybit16);
  /*PWM(출력할 핀번호, 듀티비, 듀티비 16비트로 변경시 HIGH(미변경 및 생략 시 false));*/
  //11번핀을 40% 듀티비로 PWM출력
  PWM(11, 65535*0.4, 1);
}
```

□ FDPWM 파형 출력하기

```
#include "ILIB.h"
void setup() {
  PWM_RESET();
}
void loop() {
  //FDPWM(PIN, Hz, Duty);
  FDPWM(11, 4000, 40);
  FDPWM(12, 4000, 60);
  /*(D11, D12, D13), (D44, D45, D46), (D2, D3, D5), (D6, D7, D8)
  각 그룹별로 주파수를 동일하게 설정해야 정상동작합니다.*/
}
```

□ NPWM 파형 출력하기

```
#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 D11핀 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.*/
}
```