

아두이노 우노 R3 호환보드

모터뱅크

모델명 : 아두이노 우노 R3 호환보드



제품설명

아두이노 우노 R3 는
아두이노 확장셴드와
각종 모터드라이브를 연결해서
제어용으로 사용할수 있는보드로
MCU 에 다양한 코드를 실습할수 있
습니다.그리고 MAS-001 셴드보드를
추가해서 쉽게 버튼과 속도조절까지
실험해 볼수 있는 세트입니다.

아두이노 우노 R3 상세구성

- 1) 아두이노 우노보드로 정품보드와 동일형식 구성보드.
- 2) 아두이노중 가장 많은 사용층이 우노보드이며 다양한 자료들이 많은게 장점.
- 3) MCU : Atmega328P (SMD or DIP).

제품의 전기적 특성 :

로직전압 : DC 5V
Vin 전압(Jack) : 7 ~ 12V
I/O 핀당 작동 전류 : 20mA
디지털 I/O 핀 : 14 개
PWM 핀 : 6 개(디지털 I/O 핀과 중복사용)
아나로그 입력핀 : 6 개

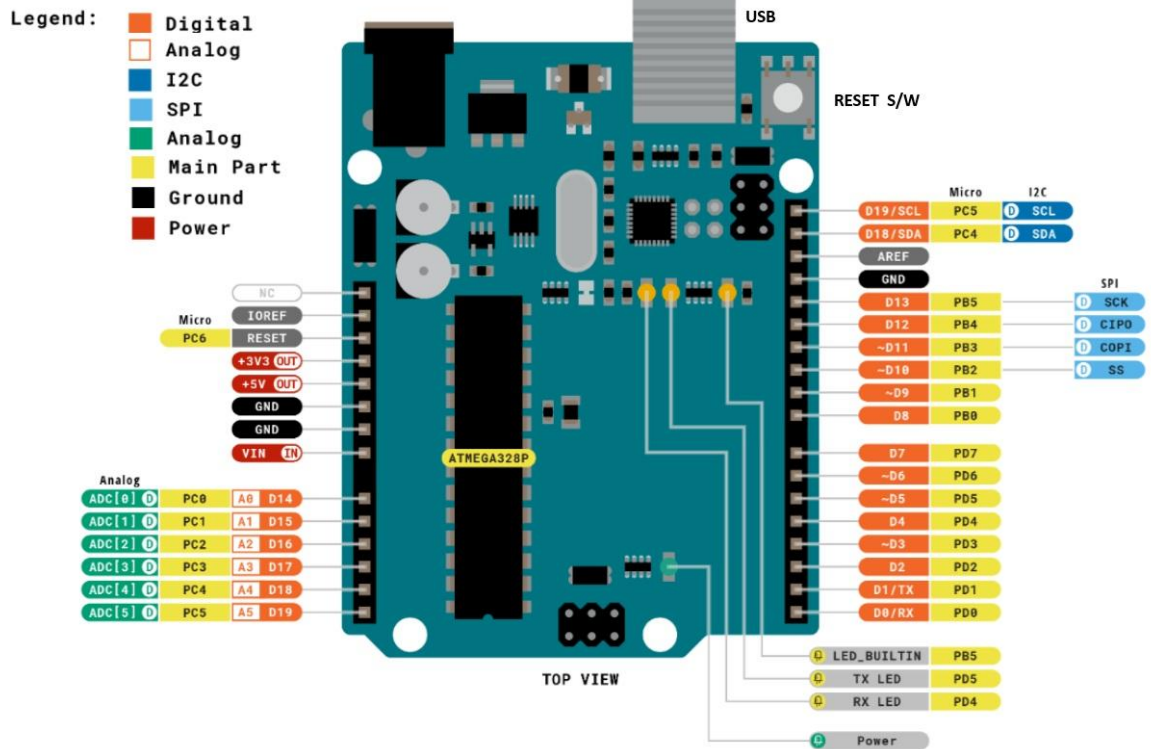
사양 :

MCU : Atmega328P (DIP 혹은 SMD)
16 MHZ, 2KB SRAM, 1KB EEPROM
32KB Flash(0.5KB bootrom)

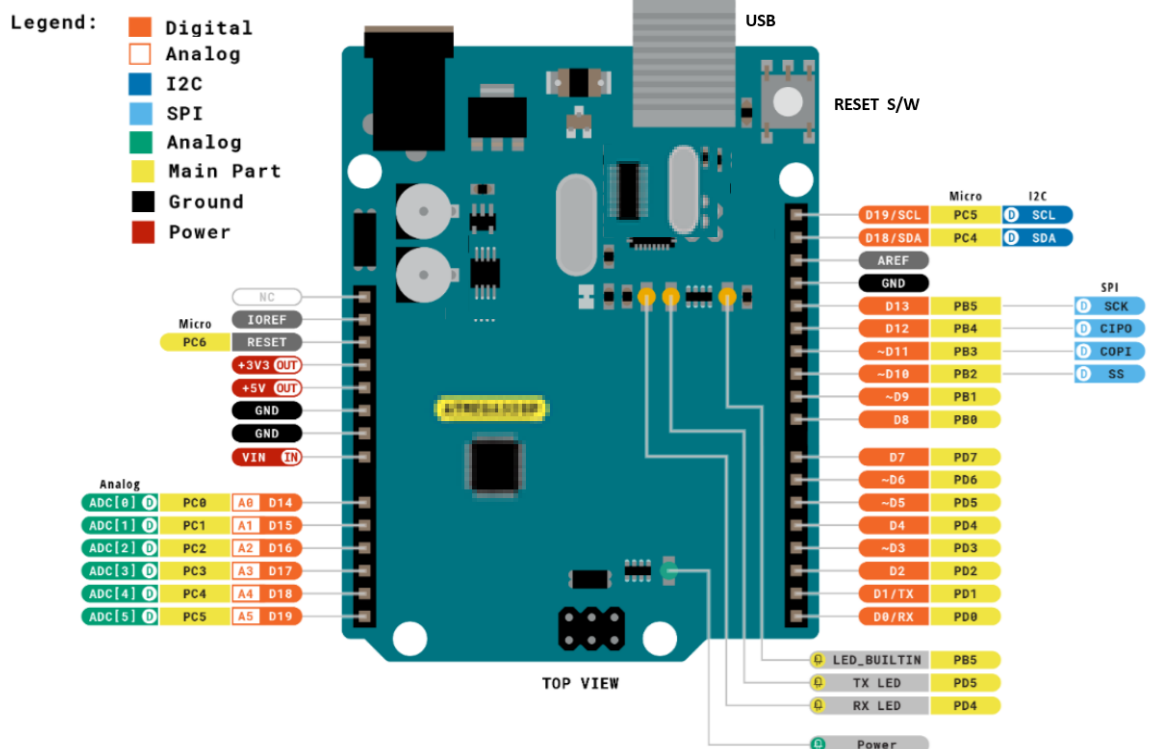
▶ Arduino Uno9(R3)

아두이노 Arduino Uno(R3)	
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

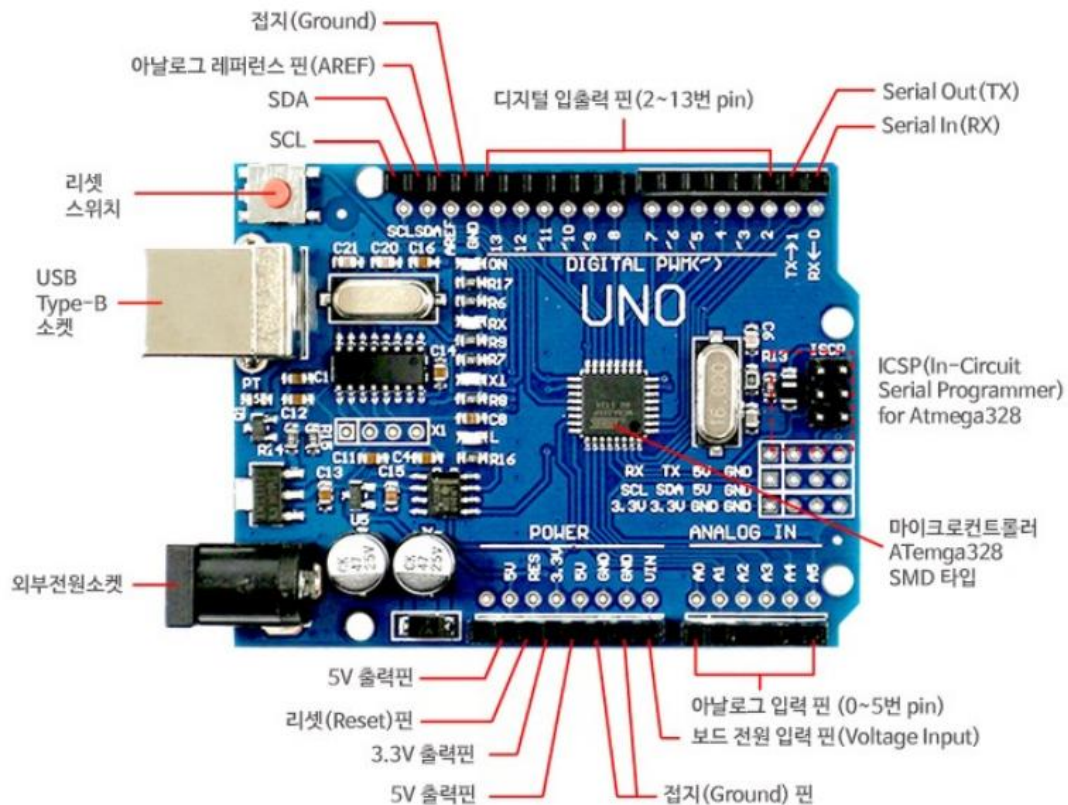
DIP type Board



SMD Type Board



두가지 클론보드가 경우에 따라 판매되는데
SMD 가 DIP 보다 대량생산에 유리하여 점차 SMD 로 바뀌고 있습니다.
DIP 은 H/W 디버깅과 고장시 수리가 용이한 점이 있지만 요즘은 장점이 크지가 않습니다.



제품치수(Dimension Drawing) :

68mm(L) x 52mm(W) Unit : mm

본 제품은 다양한 방법으로 활용이 됩니다.

개발환경은 PC 에서 아두이노 개발환경을 다운받아서 구축하면 되고

컴파일이 완료된 코드를 USB 로 간단하게 아두이노 우노보드로 다운로드 할수 있습니다.

모터뱅크에서

주로 사용하는 것은 모터(DC,BLDC,STEP)들을 돌리기 위한 방법으로 사용되고

모터드라이버에 연결하여 사용하면 다양한 제어기를 구성시킬수 있습니다.

가장 일반적이고 쉬운 형태는

아두이노 우노용 쉴드보드(MAS-100)를 추가하여 간단한 코딩으로 수동제어와 자동제어를 연습해보고 확장하는 방법입니다

참고로

여기서는 아두이노 우노의 개발환경구축은 생략합니다.

다양한 블로그나 유튜브에서 잘 소개하고 자료도 넘치므로

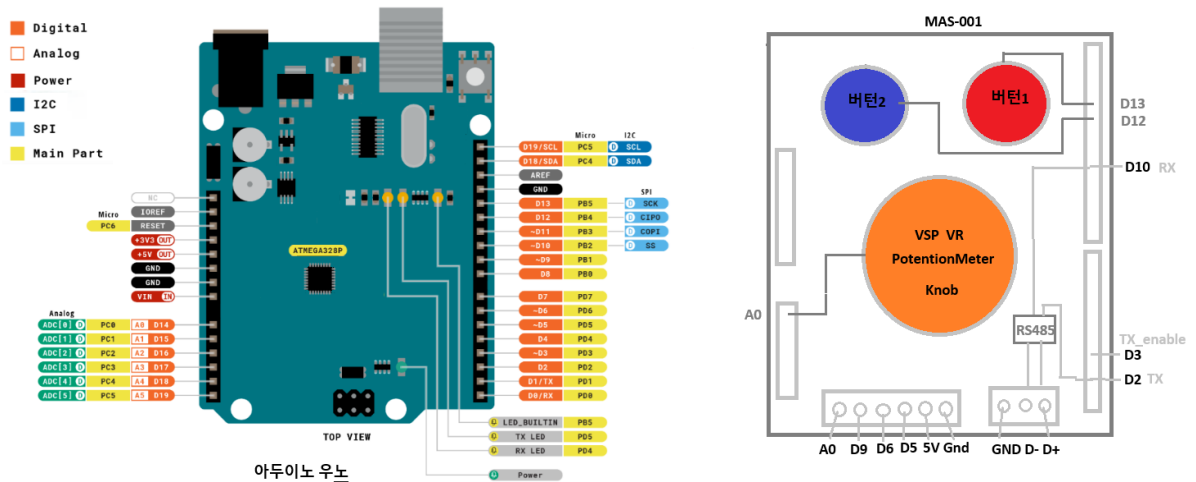
굳이 지면을 할애할 필요가 없다고 생략하니 양해바랍니다.

이후부터는

아두이노 우노에 쉴드보드(MAS-100)와 모터드라이브를 연결하는 예로

설명을 진행합니다.

간단한 아두이노 우노와 쉴드보드(MAS-001) 구성

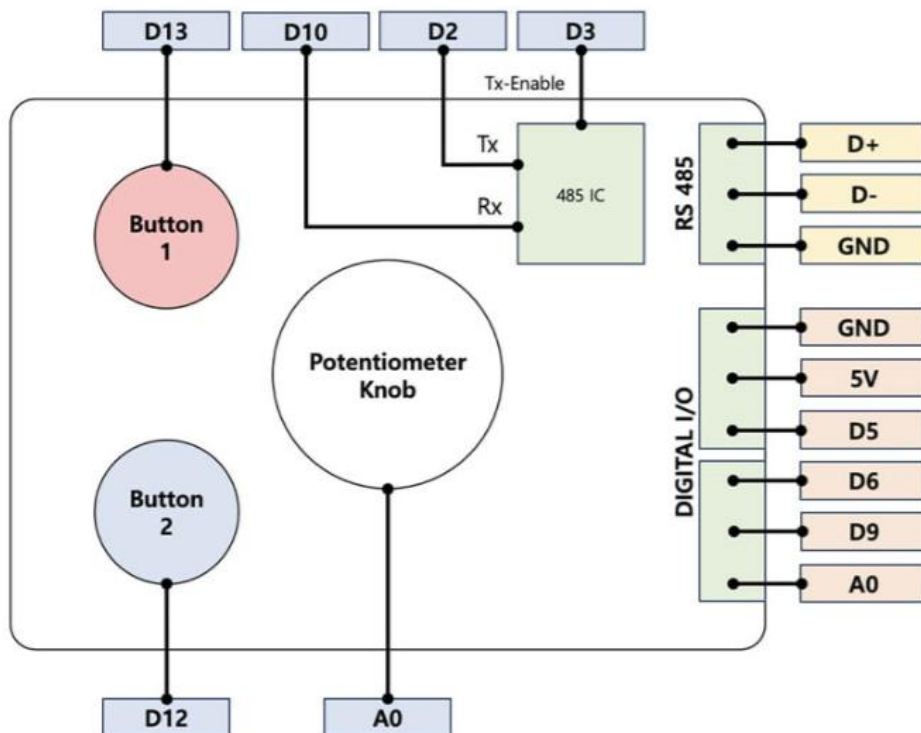


상기와 같이 쉴드보드(MAS-001)을 아두이노 우노보드에 꼽아서 사용하고
아두이노에서 Software 를 코딩하여 버튼과 출력포트(D5,D6,D9)를 매핑하고
가변저항값을 읽고 이것을 PWM 으로 변환하여 D5~D9 로 출력하거나 하면 되는식으로 제어.
특히 외부모터에서 FG 를 D5~D9 로 받아서 PWM 으로 속도제어하면 closeloop 제어도 가능.

결선구성 :

쉴드보드(MAS-001)를 아두이노 우노보드에 꼽으면 됩니다.
그리고 모터 드라이버에 VSP,PWM,DIR,FG,Enable 을 연결합니다.

제품의 연결구성도 :

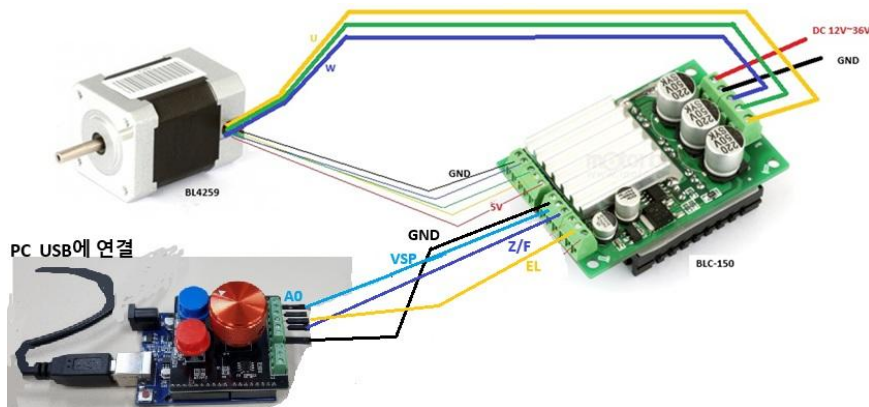


Mas-001 블록다이어 그램.

아두이노 코딩방법:

- 1) 버튼 1 이 디지털 I/O 의 D13 에 연결되어있어 이 포트를 읽고 D5 로 리매핑.
- 2) 버튼 2 가 디지털 I/O 의 D12 에 연결되어있어 이 포트를 읽고 D6 로 리매핑.
- 3) 가변저항을 A0 에서 아날로그로 읽어 D9 로 PWM 출력.
- 4) 비슷하게 RS485 나 FG 도 처리하면 됩니다

Mas-001 + 아두이노 우노 + BLC-150 + BL4259 구성예.



아두이노 우노 참조예:

아두이노 우노의 코딩소스 예;

참조: GITHUB 에 같은 소스가 upload 되어 있습니다.

```
#include "Arduino.h"

#define RS485_RX      10
#define RS485_TX      2
#define RS485_DE      3
#define MAS_BTN1      13
#define MAS_BTN2      12
#define MAS_POT        A0
#define EN_PIN  6
#define DIR_PIN 5

class MAS001{
public:
    MAS001();
    bool button1Clicked();
    bool button2Clicked();
    uint16_t getPot();
};

MAS001::MAS001(){
    pinMode(MAS_BTN1, INPUT);
    pinMode(MAS_BTN2, INPUT);
}

bool MAS001::button1Clicked(){
    if(digitalRead(MAS_BTN1) == LOW) return true;
    return false;
}

bool MAS001::button2Clicked(){
    if(digitalRead(MAS_BTN2) == LOW) return true;
    return false;
}

uint16_t MAS001::getPot(){
    return analogRead(MAS_POT);
}

class BLC150
{
public:
    BLC150();

    void enable(uint8_t value);
    void direction(uint8_t value);
};

BLC150::BLC150(){
    pinMode(EN_PIN, OUTPUT);
    pinMode(DIR_PIN, OUTPUT);
}
```

```

}
void BLC150::enable(uint8_t value){
    digitalWrite(EN_PIN, value);
}

void BLC150::direction(uint8_t value){
    digitalWrite(DIR_PIN, value);
}

////////////////////////////////////
MAS001 myShield;
BLC150 myDevice;

void setup() {

}

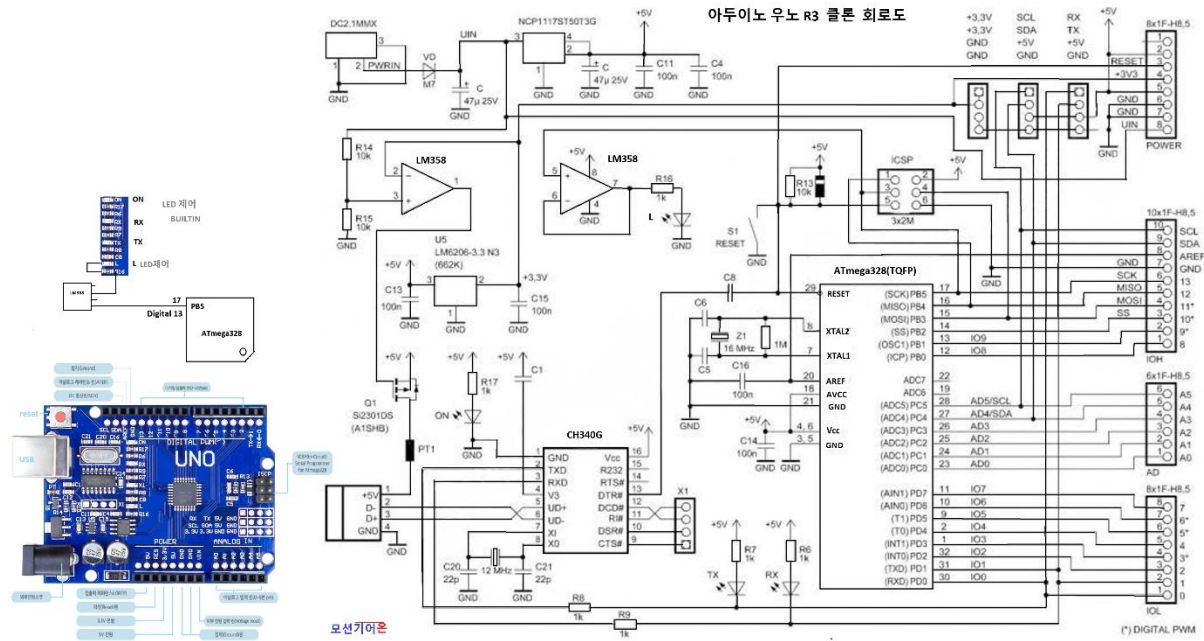
void loop() {
    if(myShield.button1Clicked()) myDevice.enable(LOW);
    else myDevice.enable(HIGH);

    if(myShield.button2Clicked()) myDevice.direction(LOW);
    else myDevice.direction(HIGH);

    delay(50);
}

```

아두이노 우노 R3 클론보드 회로도.



아두이노 우노 확장실드 회로도.

