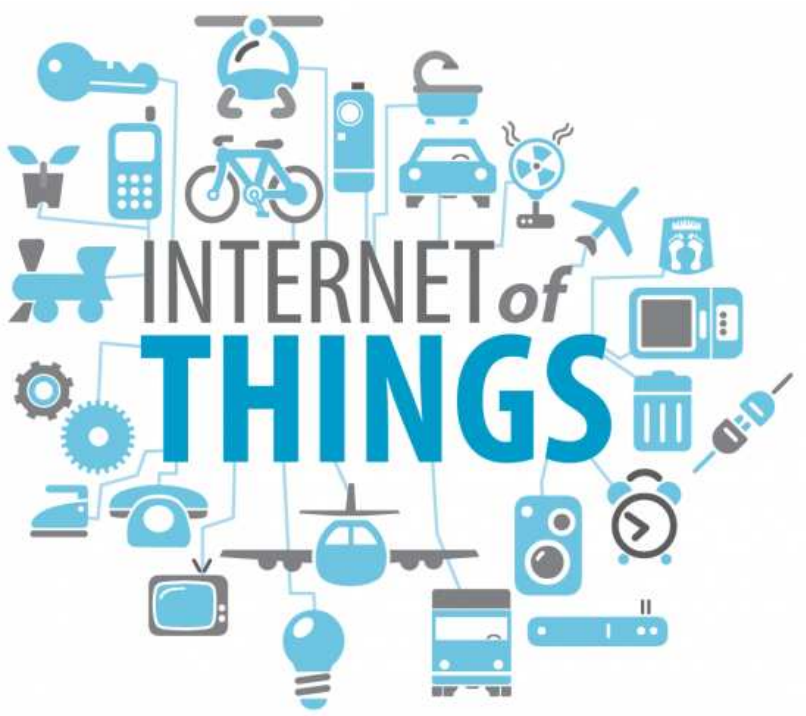
전자 회로 프로그래밍

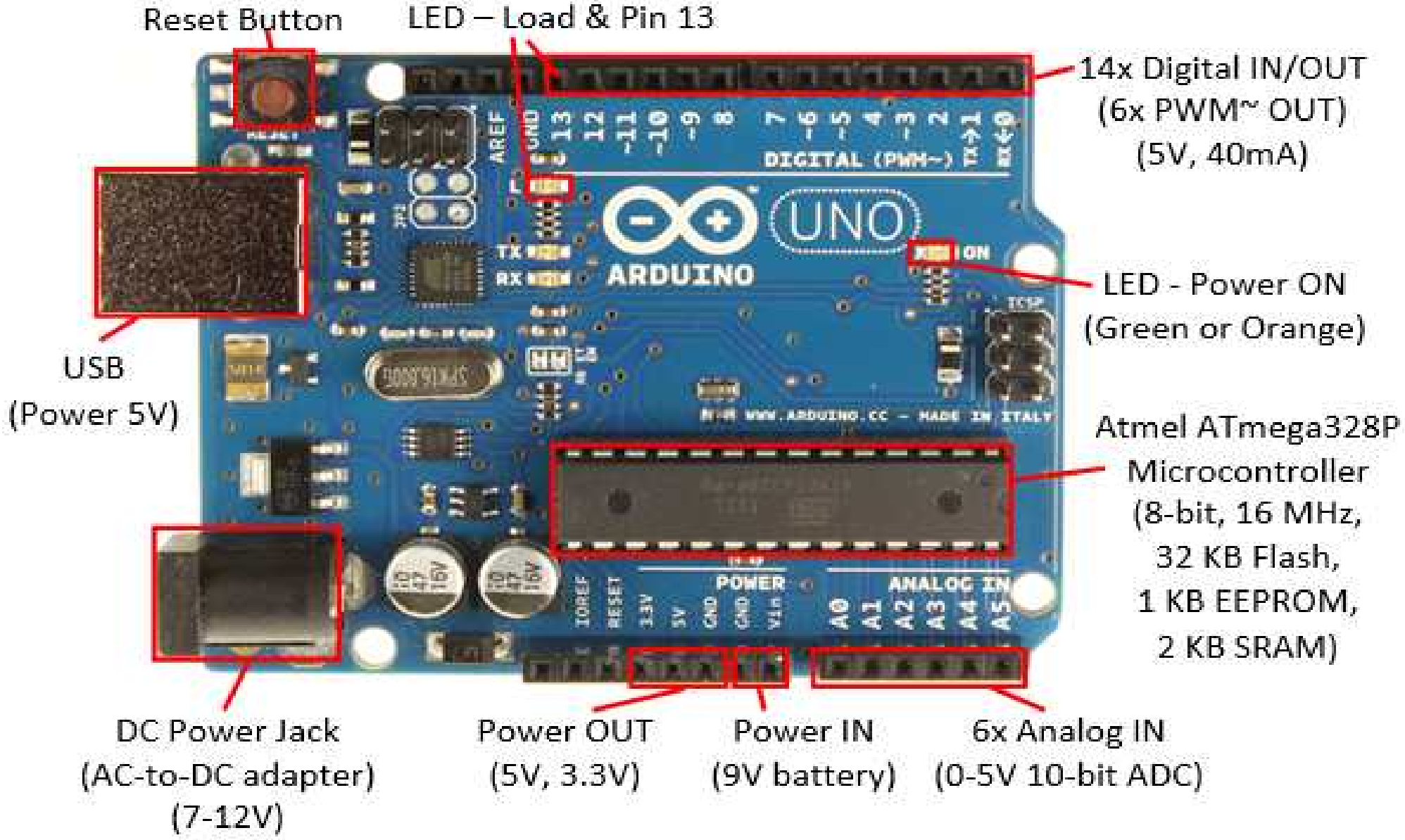
Arduino

❖ Internet of Things

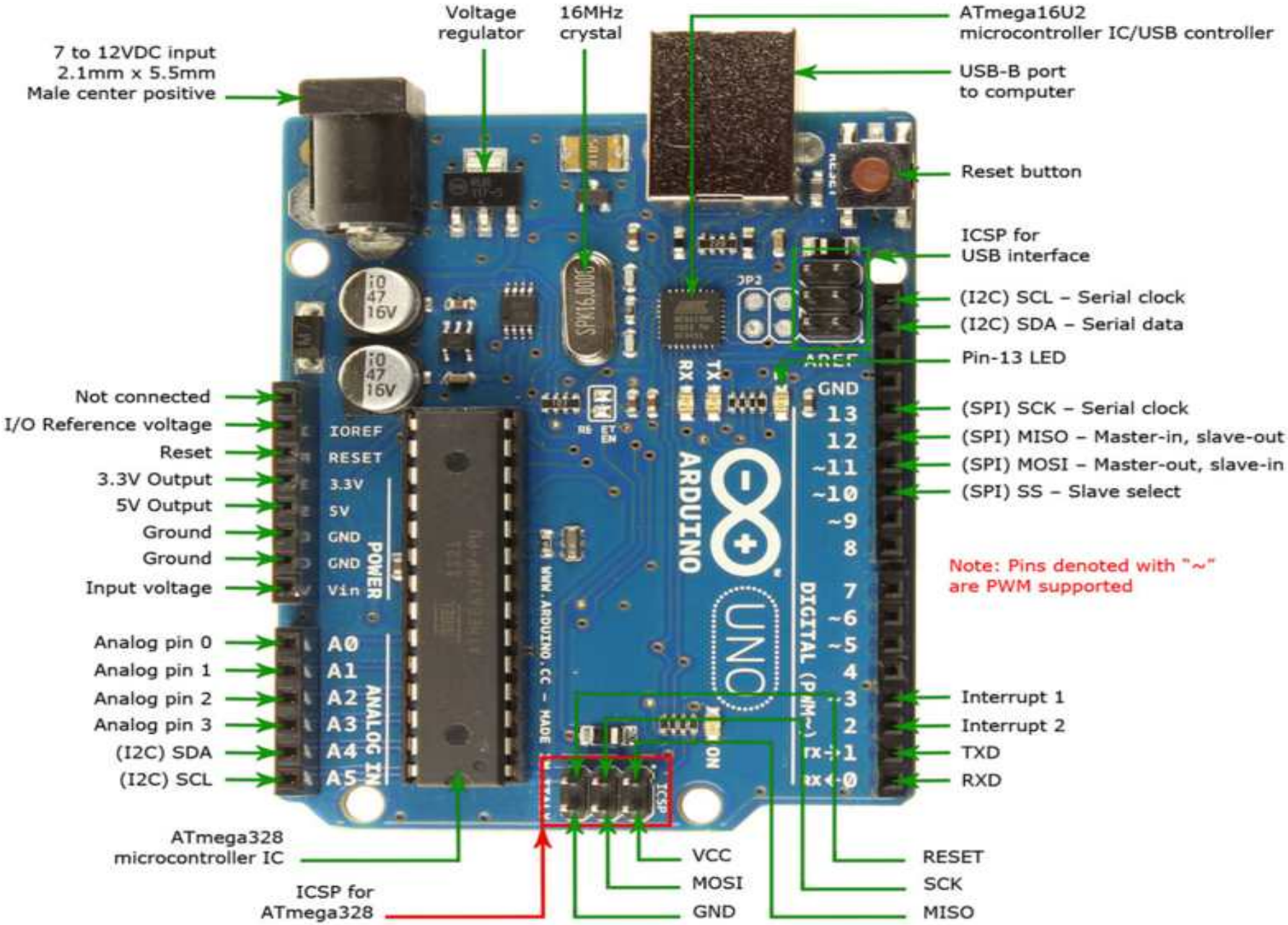
- 현실의 사물에 IP(외부 인터넷) 연결
- 그 사물들을 외부에서 제어할 수 있는 기술
- MCU + Sensor + Actuator + Internet + Database + AI + ETC



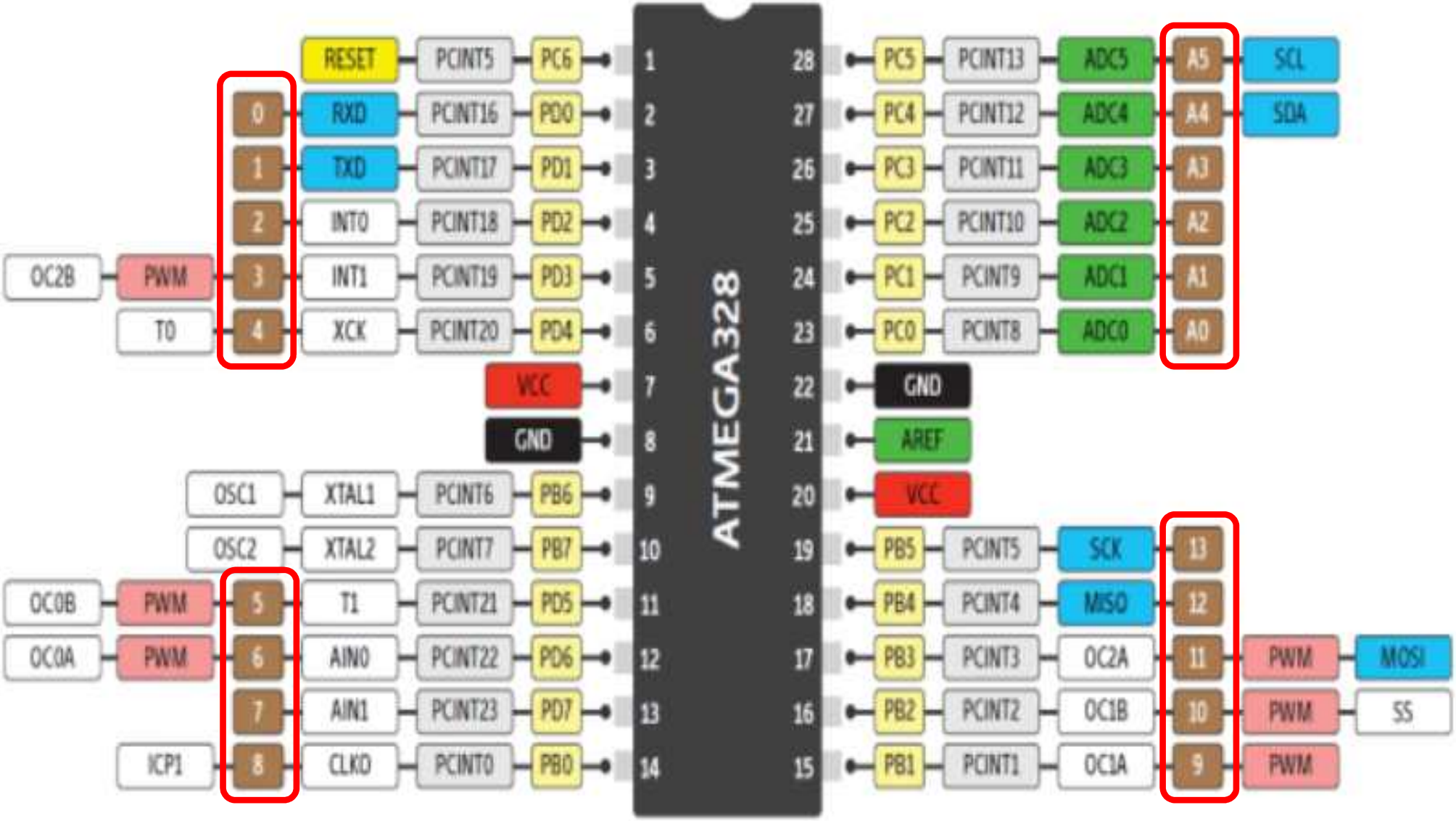
Atmega328


Reset Button
LED – Load & Pin 13
14x Digital IN/OUT
(6x PWM~ OUT)
(5V, 40mA)
LED - Power ON
(Green or Orange)
USB
(Power 5V)
Atmel ATmega328P
Microcontroller
(8-bit, 16 MHz,
32 KB Flash,
1 KB EEPROM,
2 KB SRAM)
DC Power Jack
(AC-to-DC adapter)
(7-12V)
Power OUT
(5V, 3.3V)
Power IN
(9V battery)
6x Analog IN
(0-5V 10-bit ADC)
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN

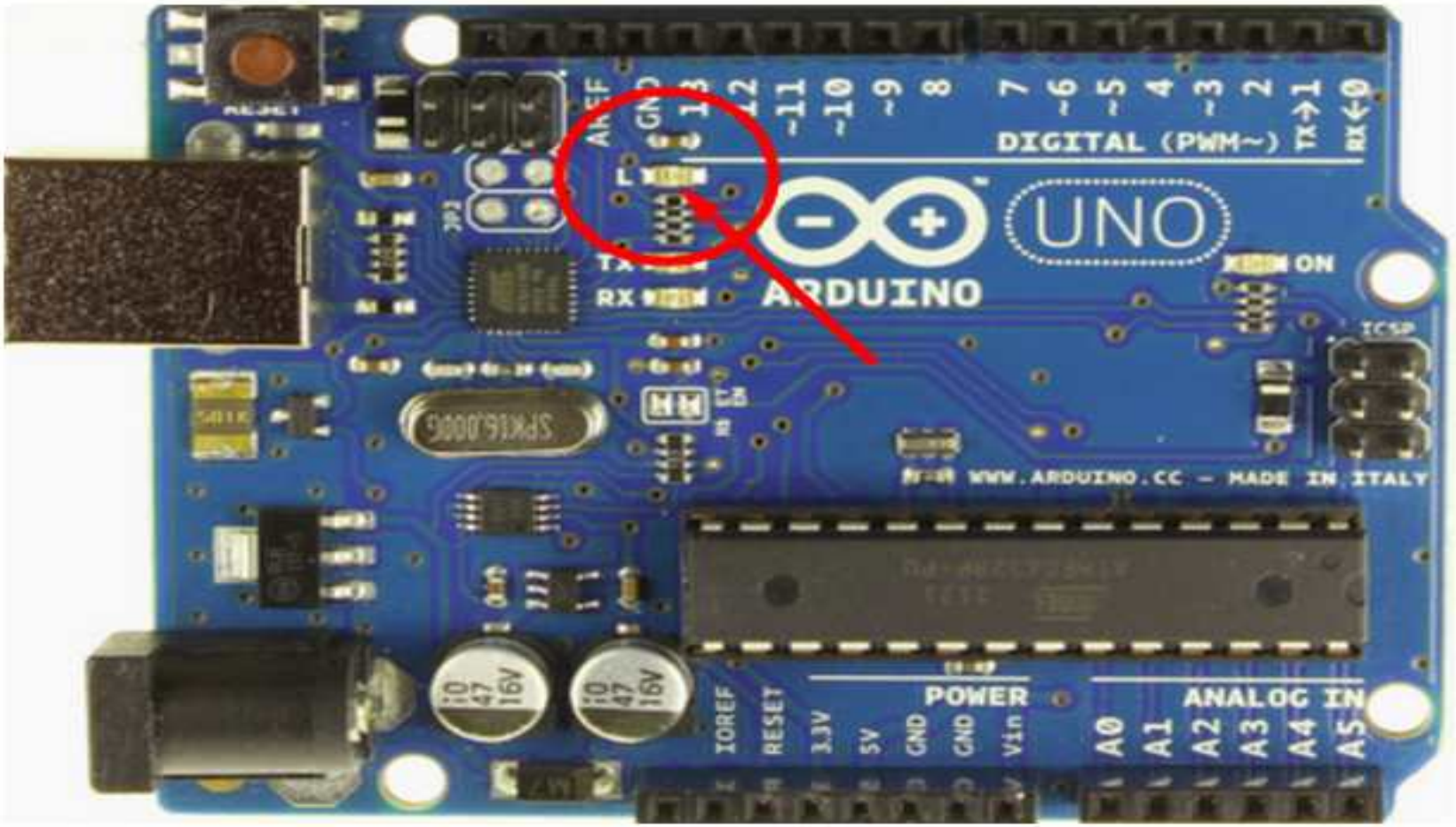
Atmega328


Voltage regulator
16MHz crystal
ATmega16U2 microcontroller IC/USB controller
7 to 12VDC input 2.1mm x 5.5mm Male center positive
USB-B port to computer
Reset button
ICSP for USB interface
(I2C) SCL – Serial clock
(I2C) SDA – Serial data
Pin-13 LED
Not connected
I/O Reference voltage
Reset
3.3V Output
5V Output
Ground
Ground
Input voltage
(SPI) SCK – Serial clock
(SPI) MISO – Master-in, slave-out
(SPI) MOSI – Master-out, slave-in
(SPI) SS – Slave select
Note: Pins denoted with "~" are PWM supported
Analog pin 0
Analog pin 1
Analog pin 2
Analog pin 3
(I2C) SDA
(I2C) SCL
Interrupt 1
Interrupt 2
TXD
RXD
ATmega328 microcontroller IC
ICSP for ATmega328
VCC
MOSI
GND
RESET
SCK
MISO

Atmel MCU ATmega328



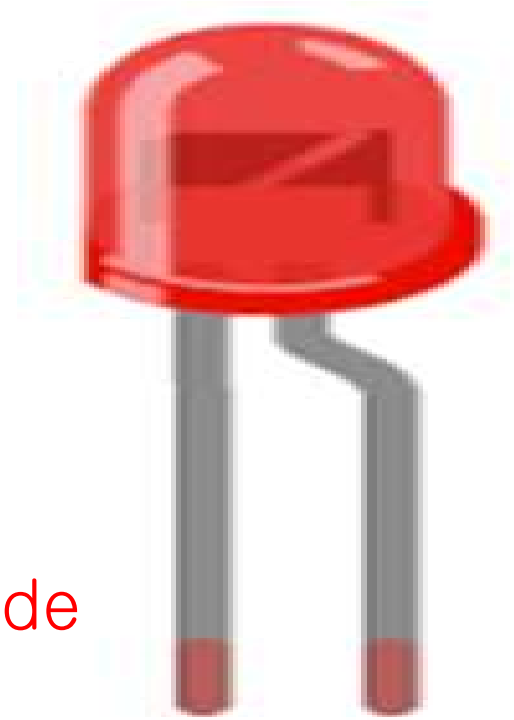
내장 LED 제어하기


DIGITAL (PWM~)
UNO
ARDUINO
ON
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

LED 소자 이해하기

❖ LED(Light Emitting Diode)

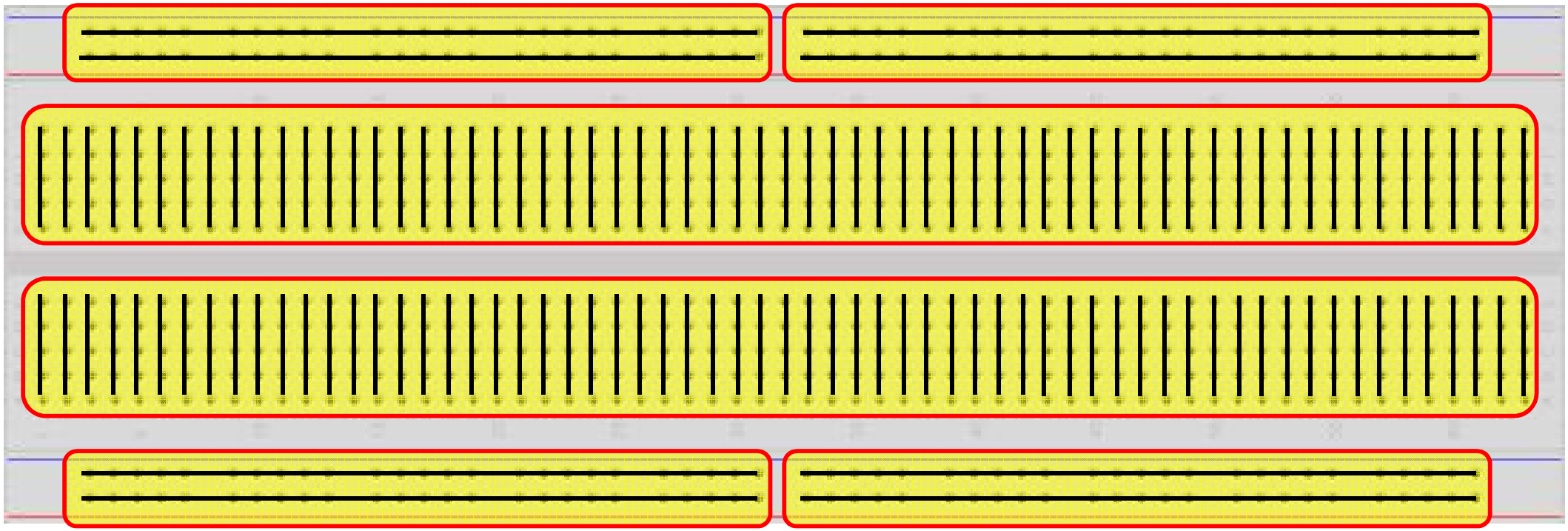
- ❖ 순방향으로 전압을 가했을 때 발광하는 반도체 소자
- ❖ Anode > + 극, Cathode > - 극 연결



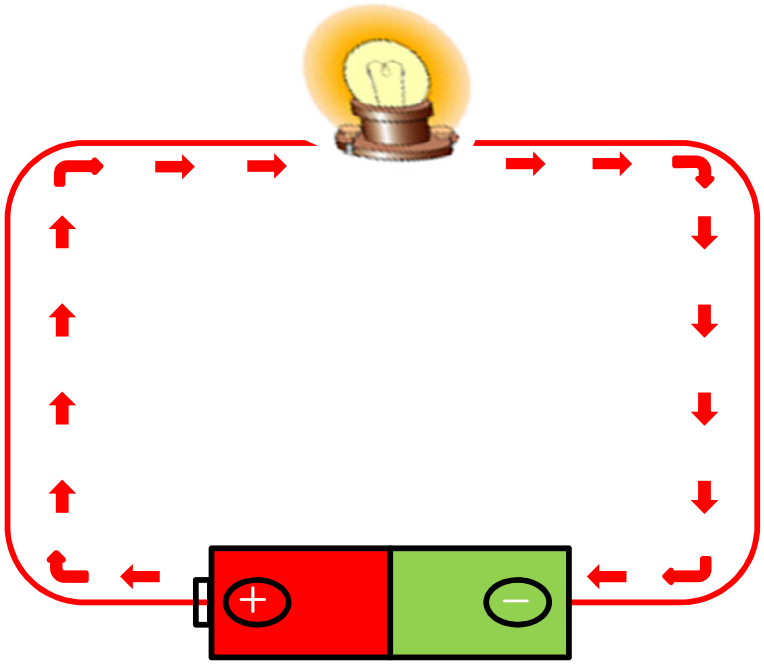
Cathode
(-)

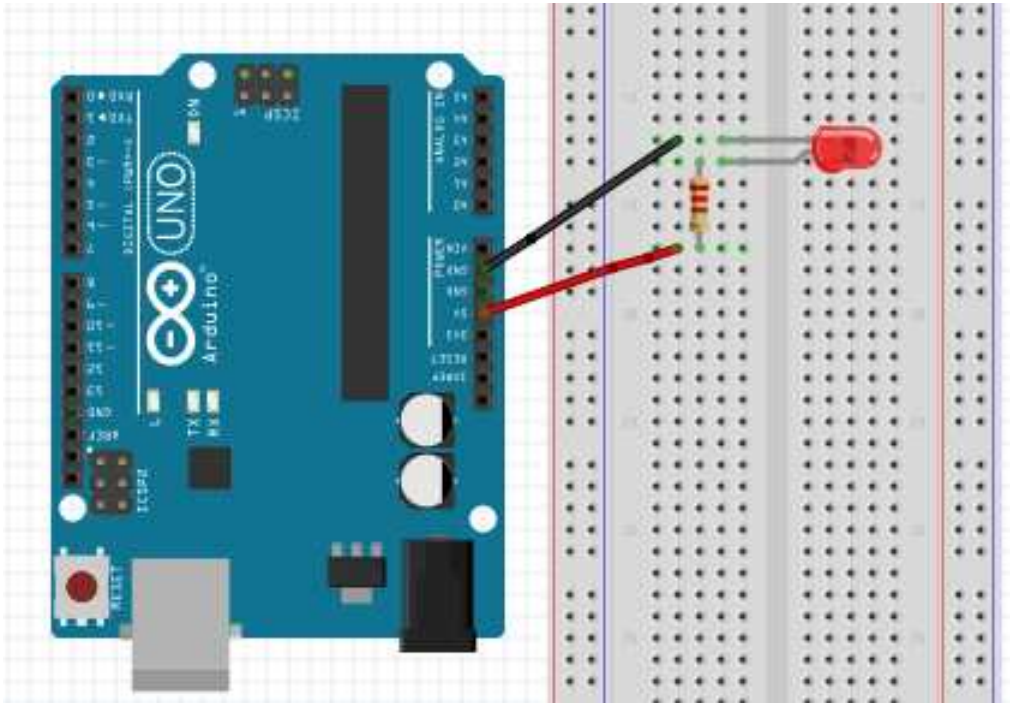
Anode
(-)

브레드보드 이해하기

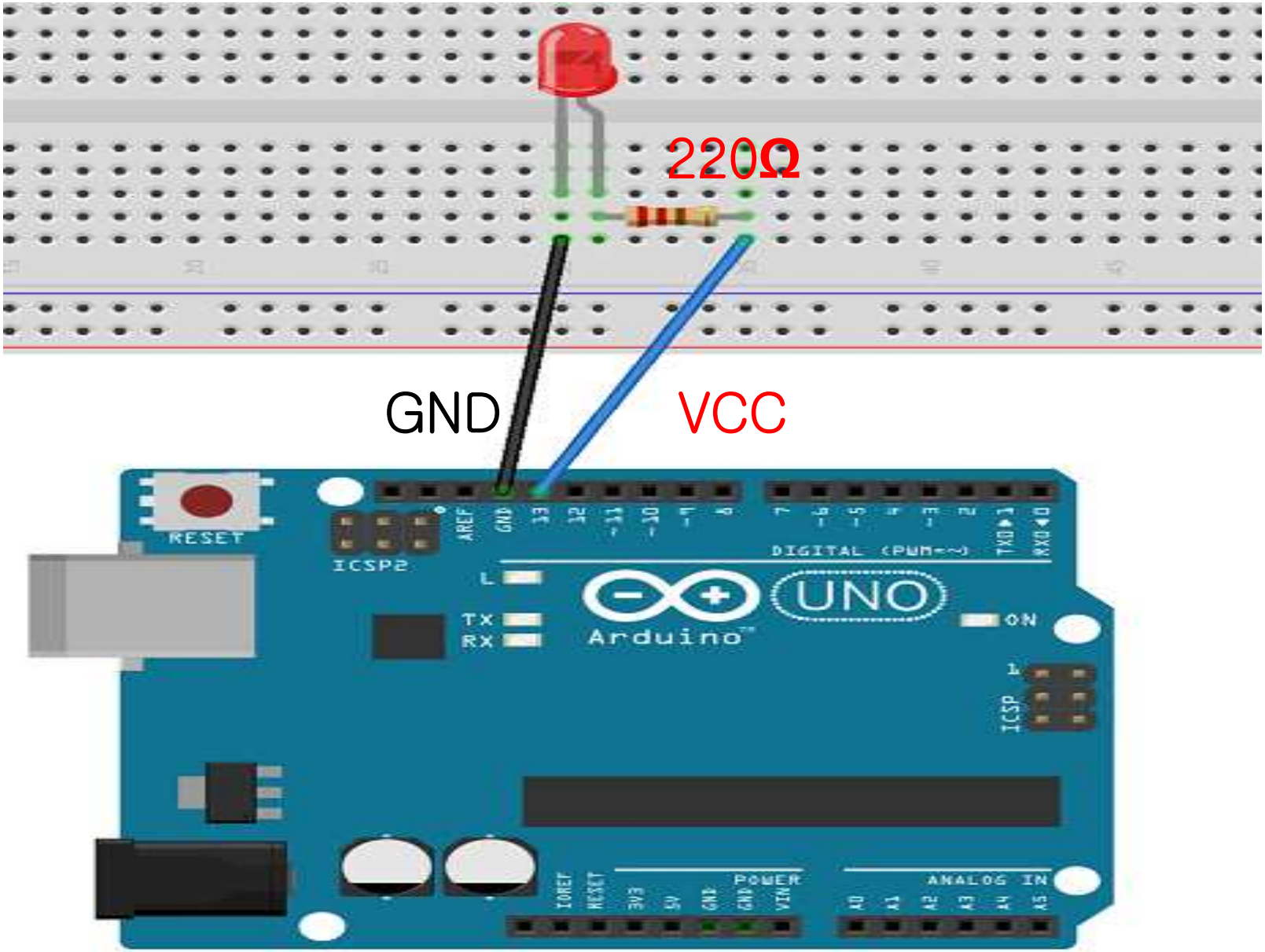


꼬마전구회로를 아두이노로 변환

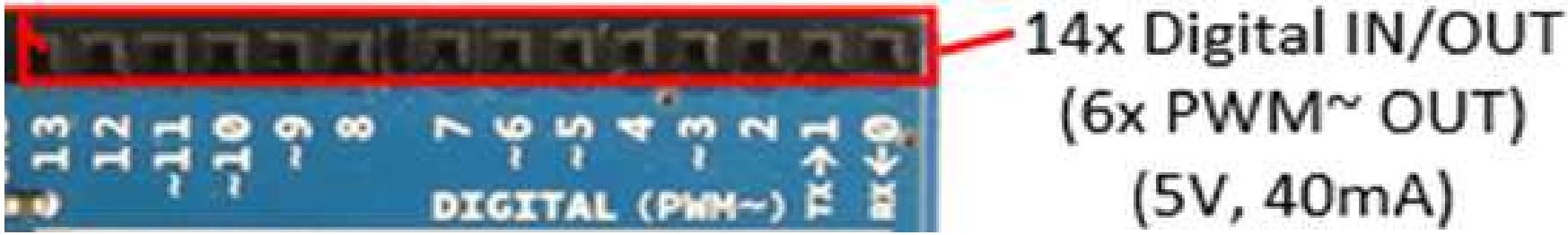




LED 연결하기


220Ω
GND
VCC
RESET
ICSP2
AREF
GND
DIGITAL (PWM~)
TX
RX
L
ON
UNO
Arduino
ICSP
POWER
ANALOG IN

디지털 출력(Digital Output)




```
void setup() // 사용핀 입출력 방향 결정
{
    pinMode(13, OUTPUT); //(핀번호, 방향)
}

void loop() // 반복적인 실행문
{
    digitwalWrite(13. HIGH); //13번 핀에서 5V 출력
    delay(1000)
    digitalWrite(13, LOW);    //13번 핀에서 0V 출력
    delay(1000)
}
```

삼색(RGB) LED 사용하기

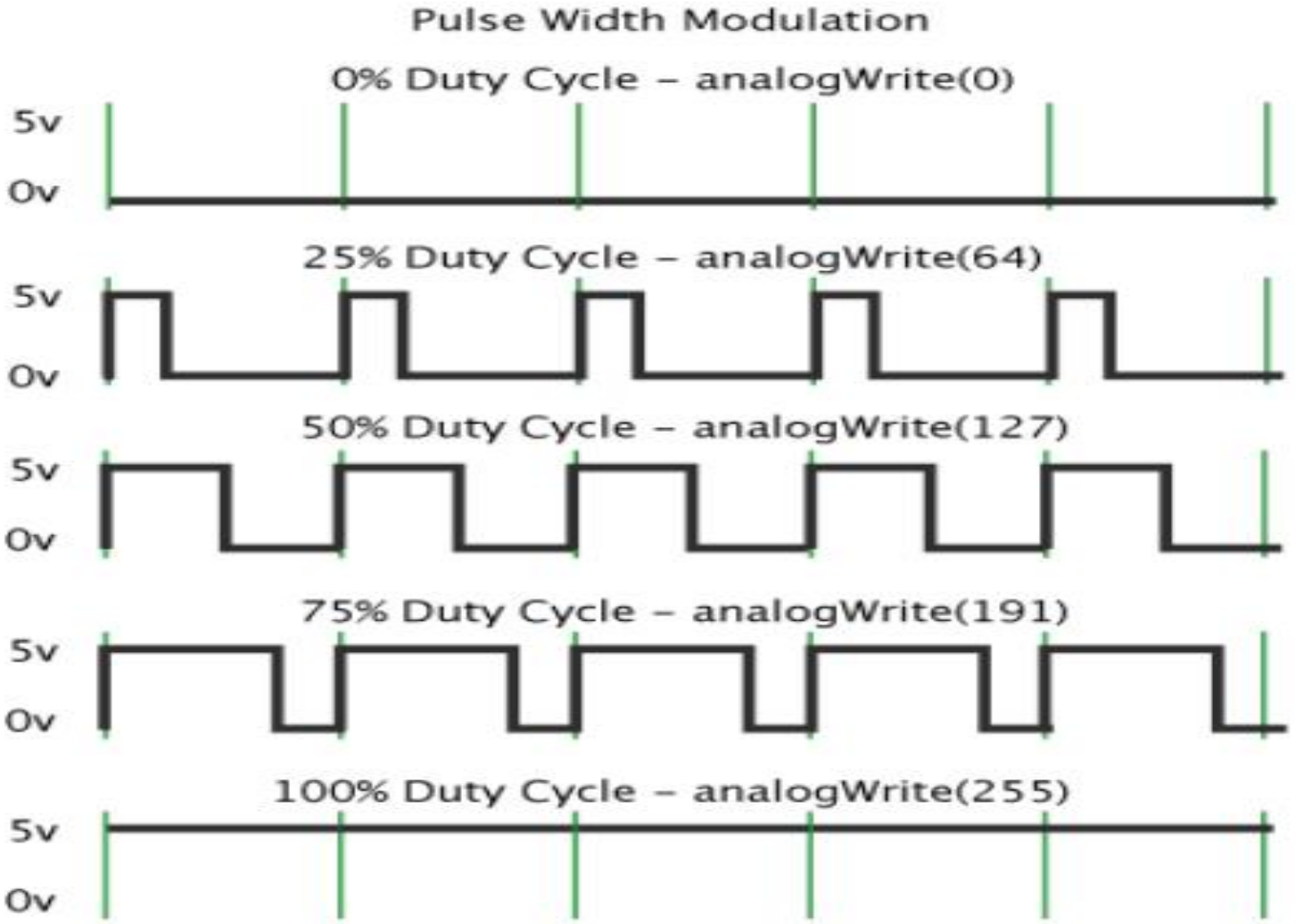
❖ 3색 LED 일반 LED 3개를 하나로 합친 LED

- ❖ R(RED), G(GREEN), B(BLUE), -(GND)
- ❖ 4개선으로 3색의 조합으로 다양한 색상 표현

아날로그 출력(Analog Output)

❖ LED 밝기 또는 모터 속도 제어

❖ analogWrite()는 0 ~ 255 크기

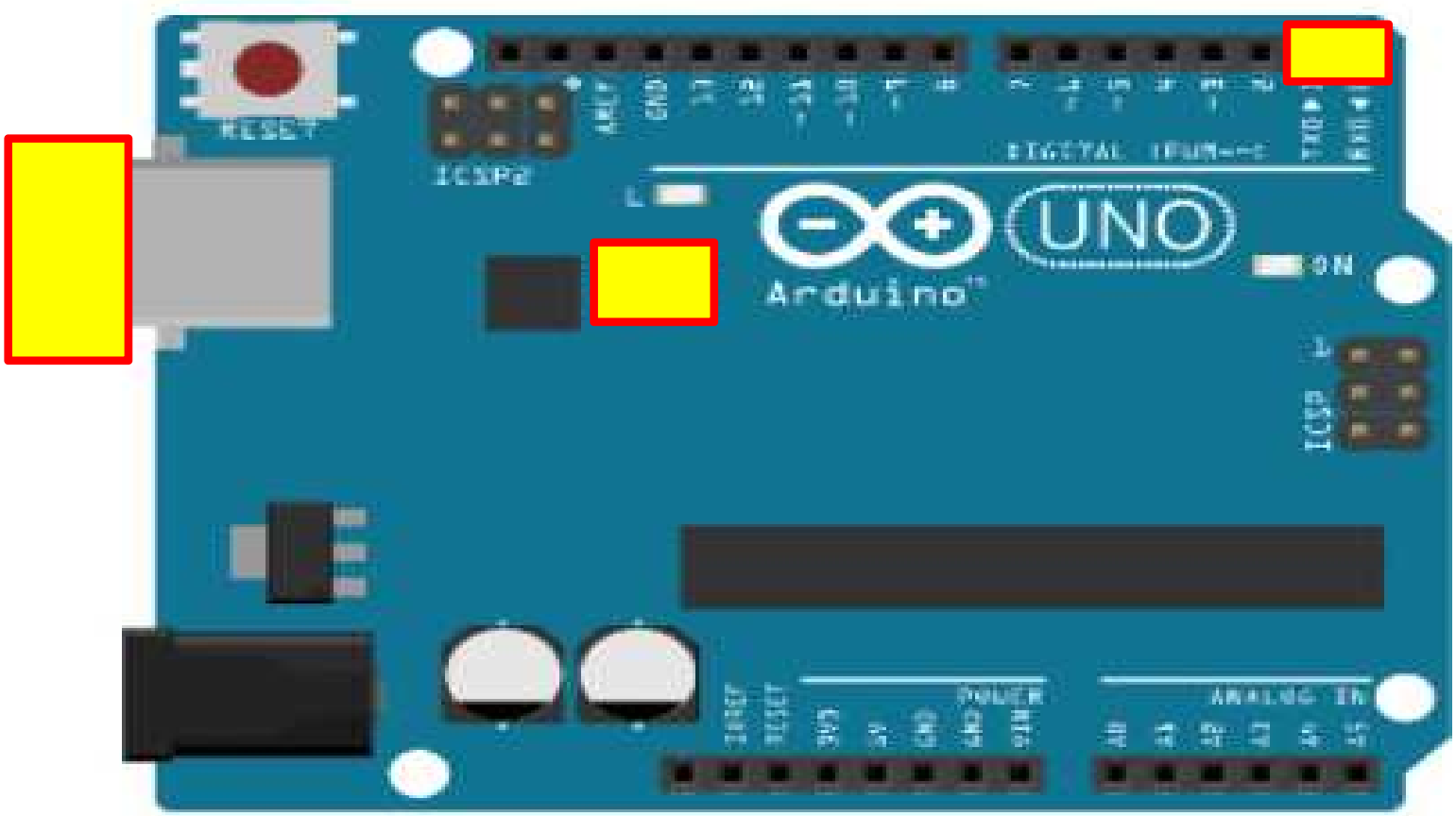


아날로그 출력(Analog Output)

```
void setup() // PWM 핀(3, 5, 6, 9, 10, 11)
{
    pinMode(3, OUTPUT); // PWM 핀결정
}

void loop() // 반복적인 실행문
{
    for(int i = 0; i < 255;  i++)
    {
        analogWrite(3,i);
        delay(100);
    }
}
```

UART(Universal Asynchronous Receiver/Transmitter)

- 시리얼(Serial) – 비동기 직렬통신
  - USB를 통해 원격지와 통신을 지원하는 방식
  - 통신하거나 데이터 송수신 가능



❖ 아두이노 시리얼 함수

- Serial.write(val) : 시리얼을 통해 1byte 또는 여러 바이트 데이터를 전송 함수
  - val : 여러 바이트 일 경우 문자열, 1byte 데이터일 경우 데이터의 값
- Serial.print(val, format) : 시리얼을 통해 전송할 데이터를 ASCII문자열로 전송함수
  - val : 전송할 문자열 또는 데이터 값
  - Format : 입력받은 문자열 또는 데이터를 출력할 형식을 설정하는 옵션

❖ 아두이노 시리얼 함수

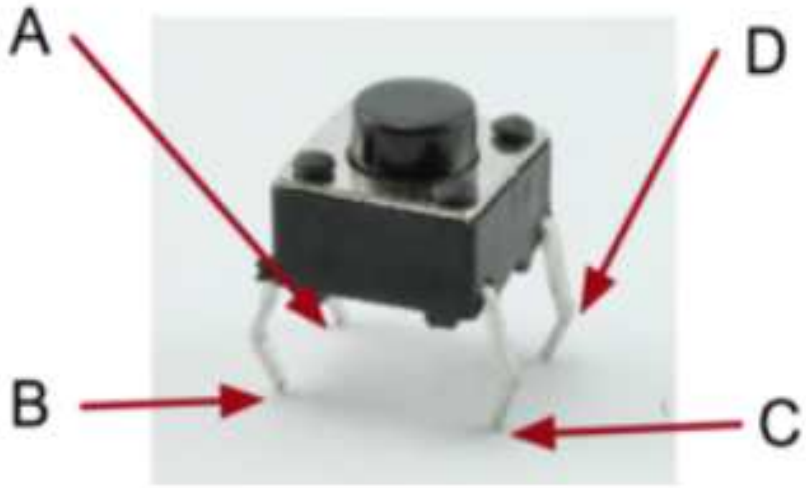
- ❖ Serial.begin(speed) : 사용할 UART와 통신 속도를 설정하는 함수
- ❖ Serial.available() : 수신된 데이터가 수신 버퍼에 몇 개 있는지 반환하는 함수
  - ❖ 문자 1개가 수신되면, '1' 반환
- ❖ Serial.read() : 수신 버퍼에서 1byte 데이터를 읽어오는 함수
  - ❖ 수신버퍼에서 데이터를 읽어온 후 데이터를 지움
- ❖ Serial.peek() : 수신 버퍼에서 1byte 데이터를 읽어오는 함수
  - ❖ 수신 버퍼에서 데이터를 읽어온 후 읽은 데이터를 지우지 않음

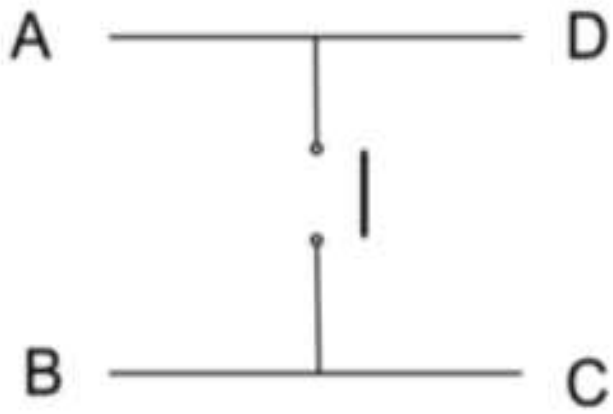
디지털 입력(Digital Iutput)

14x Digital IN/OUT
(6x PWM~ OUT)
(5V, 40mA)

```
void setup() // 사용핀 입출력 방향 결정
{
    pinMode(7, INPUT); //(핀번호, 방향)
    Serial.begin(9600);
}

void loop() // 반복적인 실행문
{
    int data = digitalRead(7); //13번 핀에서 입력
    Serial.println(data);
}
```

스위치 구조 이해하기



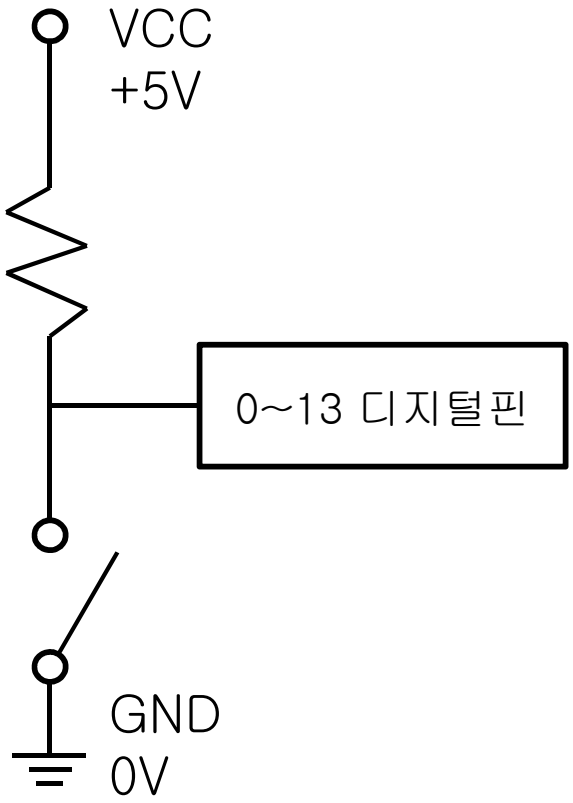


플로팅 현상

입력 값이 불안정하게 변하기 때문에 부정전압으로 오작동이 생성

❖ 풀업(Pull-up)저항

- ❖ +5V쪽에 저항을 연결해주는 방법
  - ❖ 기본 1(HIGH) 신호 상태

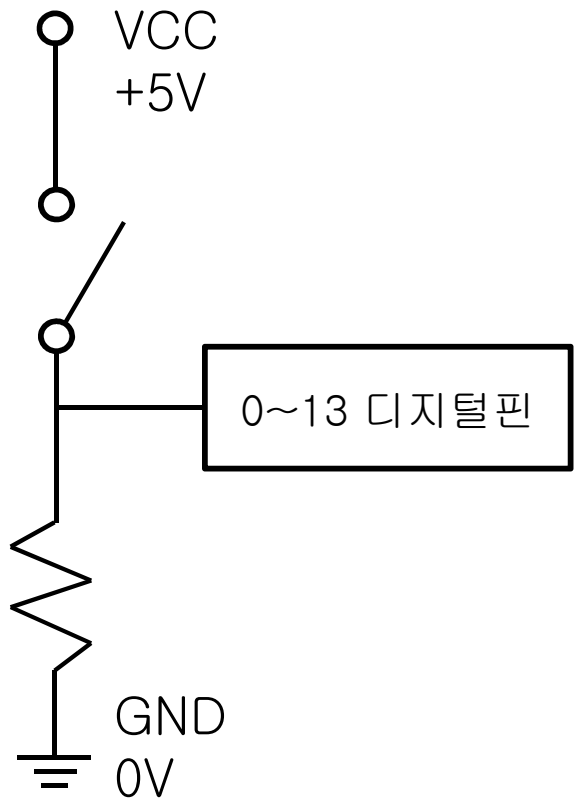


플로팅 현상

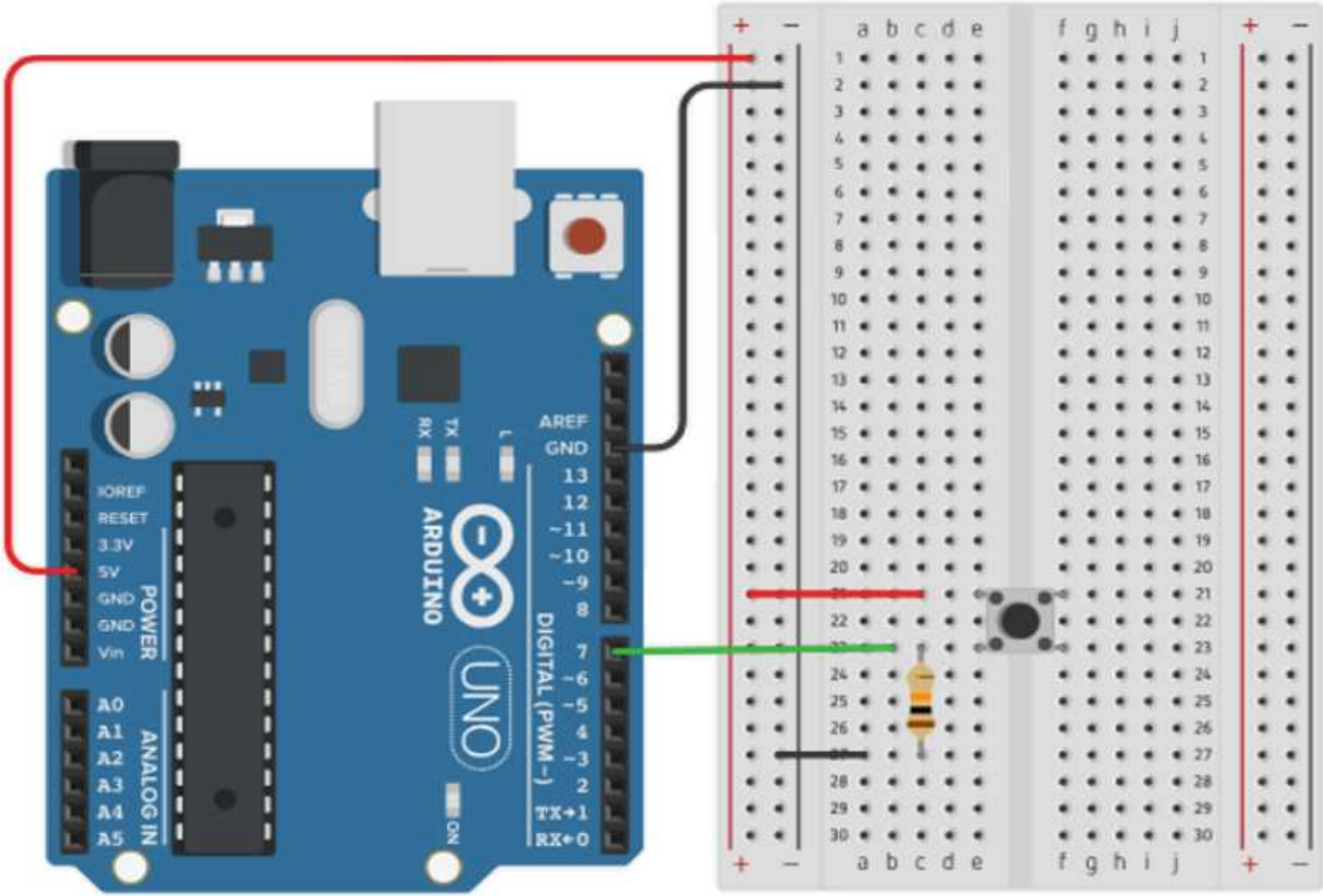
입력 값이 불안정하게 변하기 때문에 부정전압으로 오작동이 생성

❖ 풀다운(Pull-down)저항

- ❖ GND쪽에 저항을 연결해주는 방법
  - ❖ 기본 0(LOW) 신호 상태

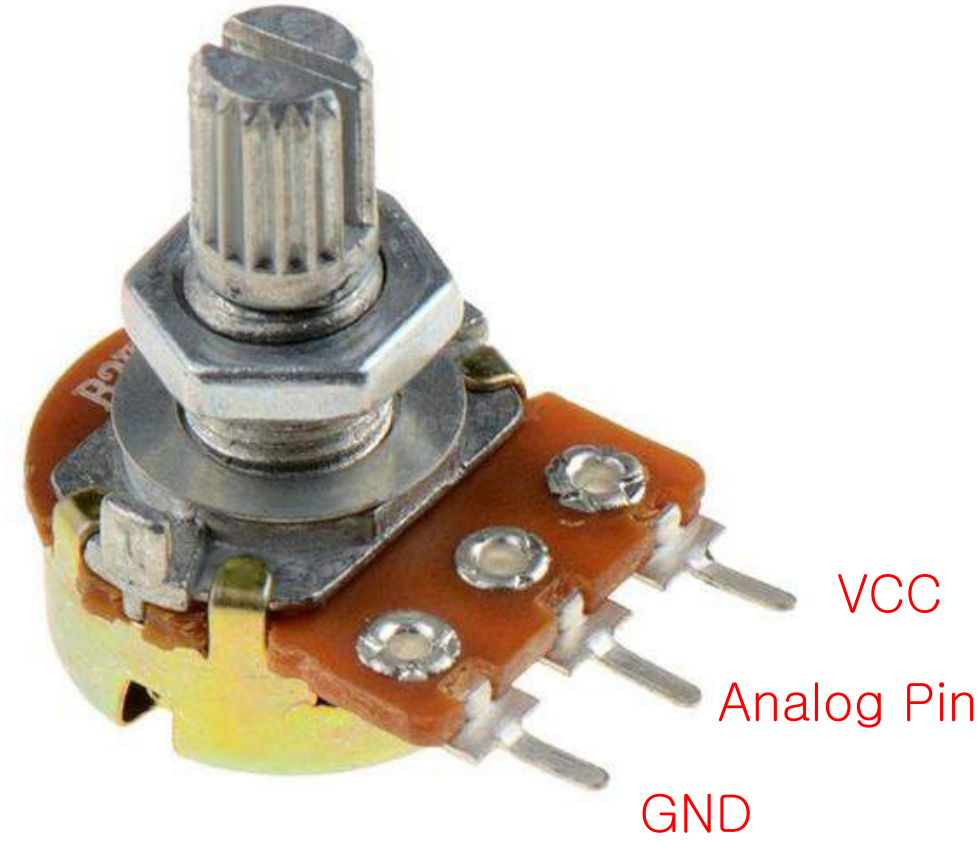


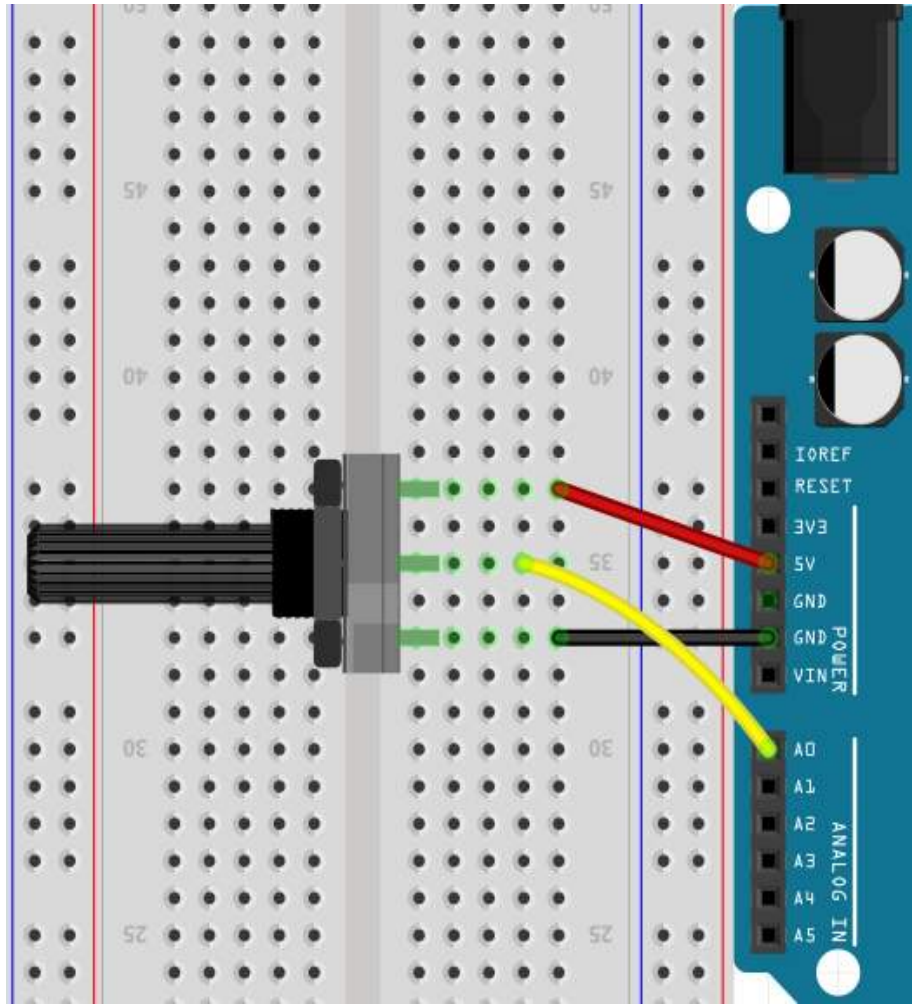
스위치 입력받기


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
ARDUINO
UNO
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
a b c d e
f g h i j

아날로그 입력

- 포텐셔미터 or 가변 저항
  - 회로에 가변 저항값을 입력하는데 사용



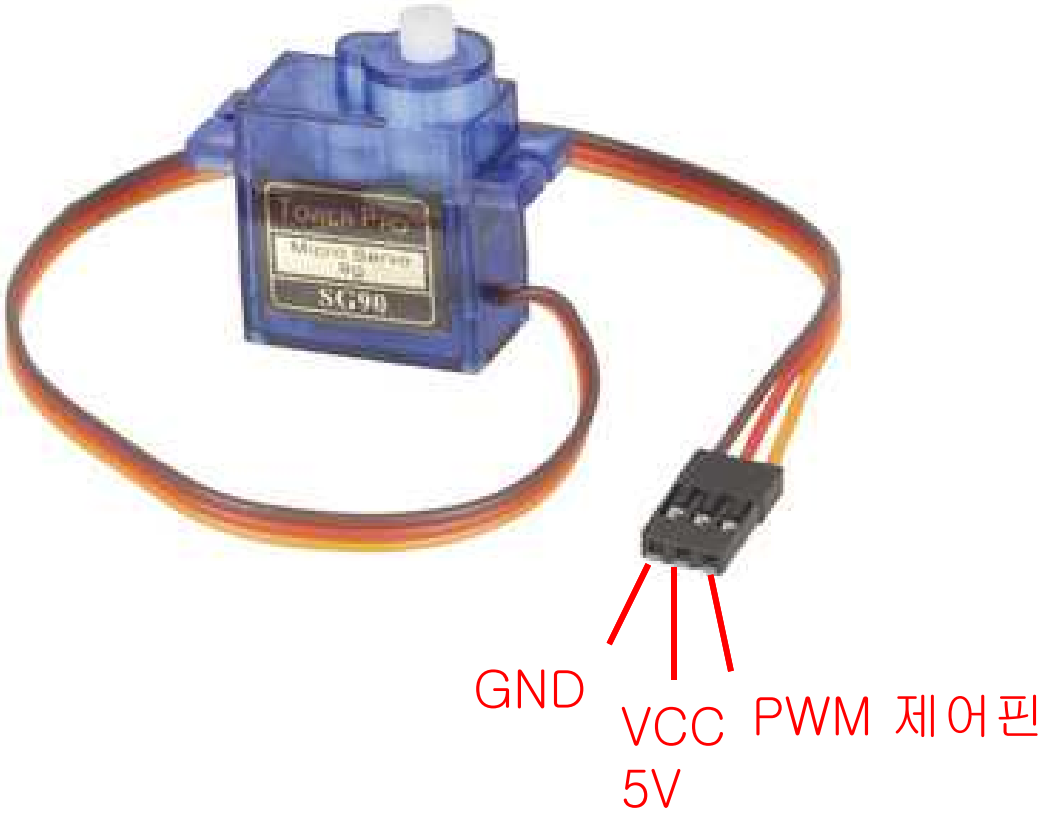


아날로그 입력(Analog Input)

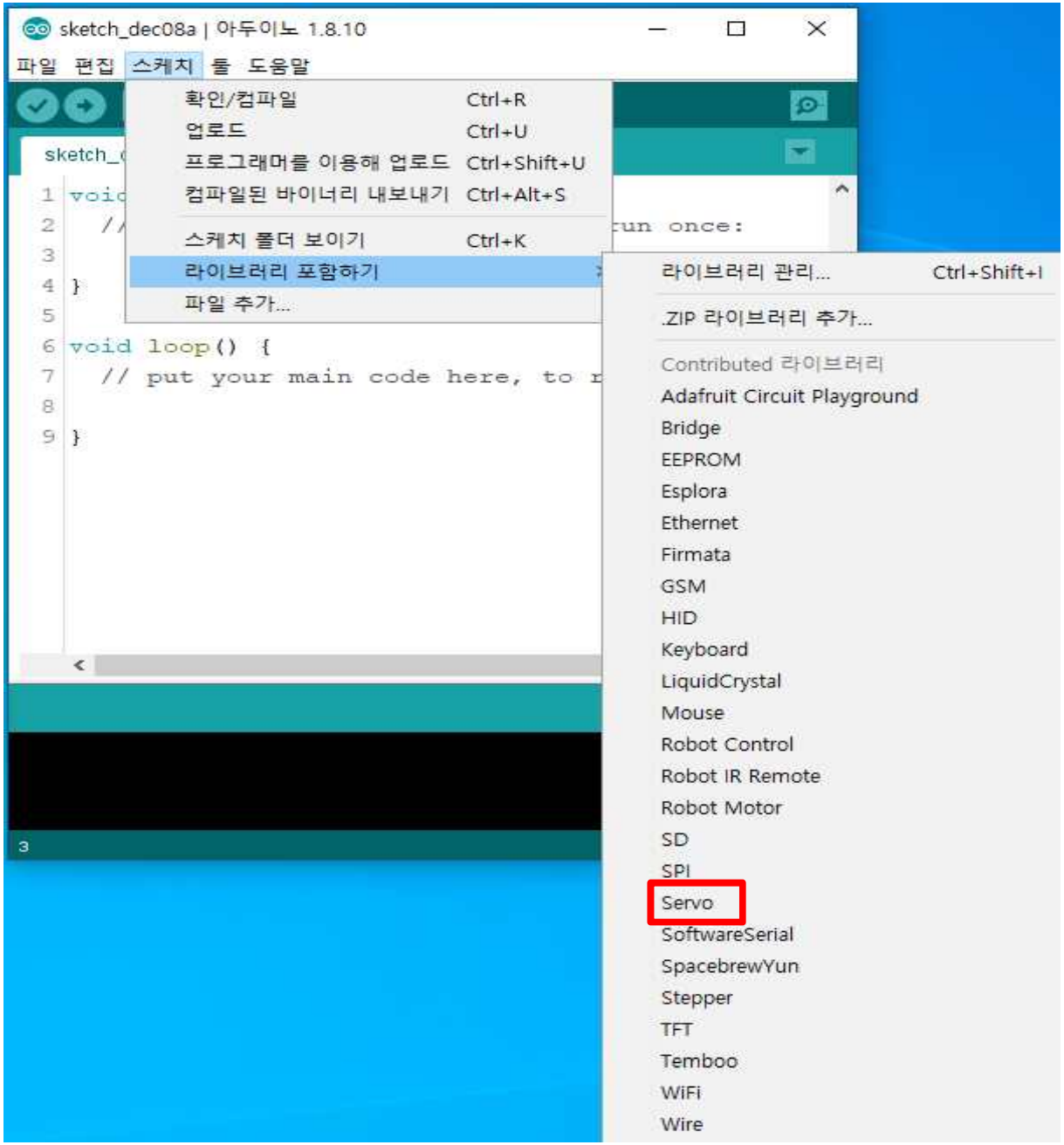

```
void setup() // 사용핀 입출력 방향 결정
{
    Serial.begin(9600);
}

void loop() // 반복적인 실행문
{
    int data = analogRead(A0); //13번 핀에서 입력
    Serial.println(data);
}
```

아날로그 출력(Analog Output)

- Servo 모터
  - 정해진 회전 반경(180)안에서 PWM 신호에 의해 제어



라이브러리 포함하기

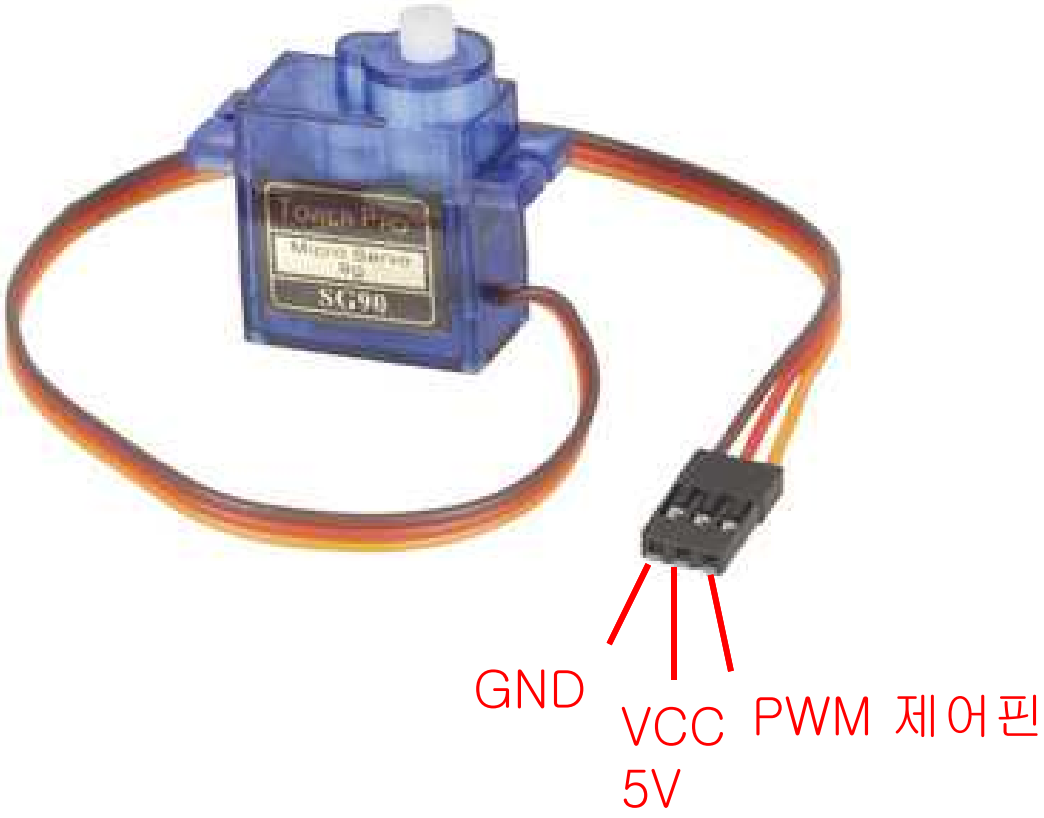

sketch_dec08a | 아두이노 1.8.10
파일 편집 스케치 툴 도움말
확인/컴파일 Ctrl+R
업로드 Ctrl+U
프로그래머를 이용해 업로드 Ctrl+Shift+U
컴파일된 바이너리 내보내기 Ctrl+Alt+S
스케치 폴더 보이기 Ctrl+K
라이브러리 포함하기
파일 추가...
라이브러리 관리... Ctrl+Shift+I
.ZIP 라이브러리 추가...
Contributed 라이브러리
Adafruit Circuit Playground
Bridge
EEPROM
Esplora
Ethernet
Firmata
GSM
HID
Keyboard
LiquidCrystal
Mouse
Robot Control
Robot IR Remote
Robot Motor
SD
SPI
Servo
SoftwareSerial
SpacebrewYun
Stepper
TFT
Temboo
WiFi
Wire
void loop() {
// put your main code here, to r
}

서보모터 제어하기

```
1  #include <Servo.h>
2  Servo servo;
3  int angle = 0;
4  void setup() {
5    servo.attach(9);
6  }
7  void loop() {
8    for(angle = 0; angle < 180; angle++)
9    {
10     servo.write(angle);
11     delay(20);
12   }
13   for(angle; angle > 0; angle--)
14   {
15     servo.write(angle);
16     delay(20);
17   }
18 }
```

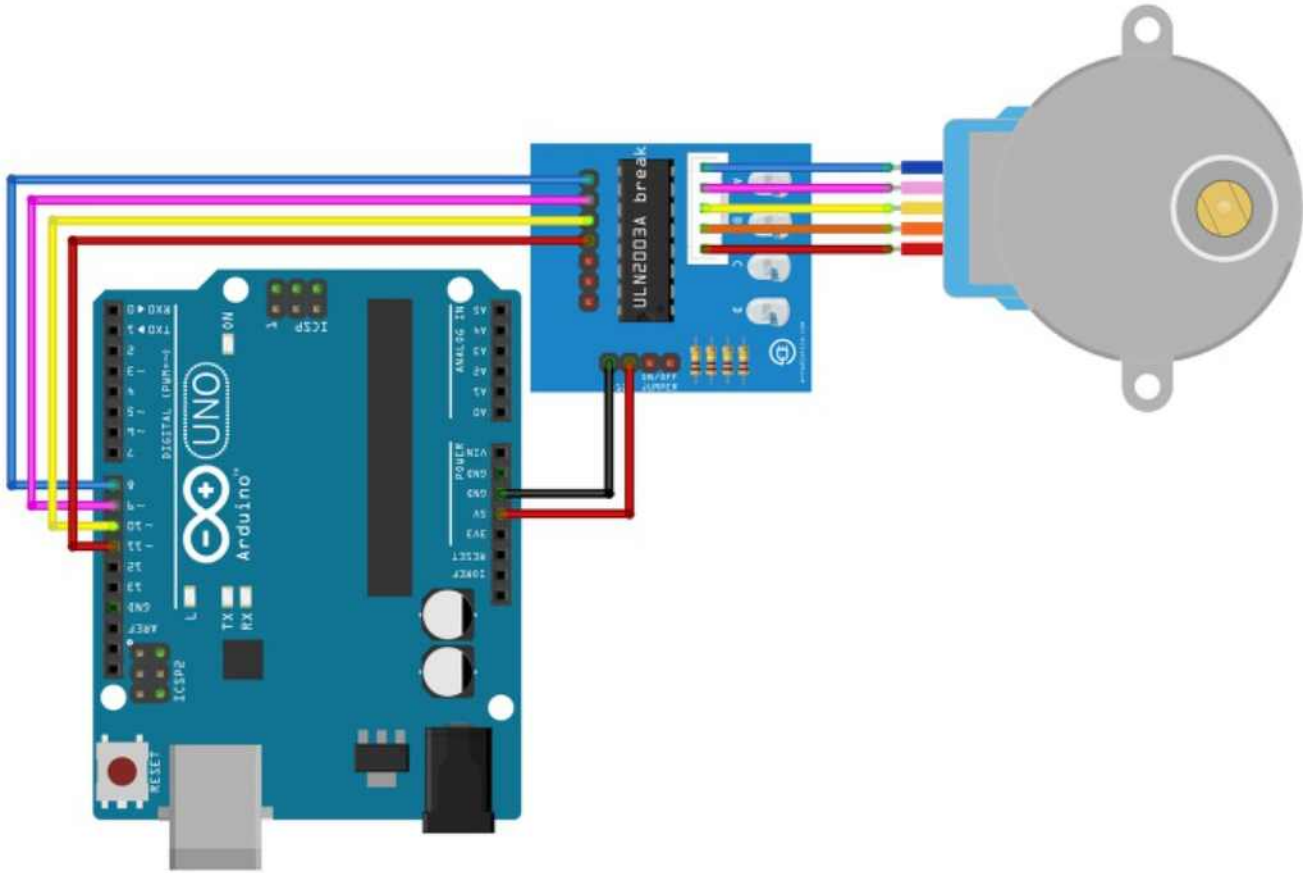
아날로그 출력(Analog Output)

- Servo 모터
  - 정해진 회전 반경(180)안에서 PWM 신호에 의해 제어



스텝모터 + 모터드라이브

- Stepping Moter(28BYJ-48) / Motor drive(ULN2003a)
  - 아두이노 최대 출력 전류 40mA
  - 모터구동 요구 출력 240mA

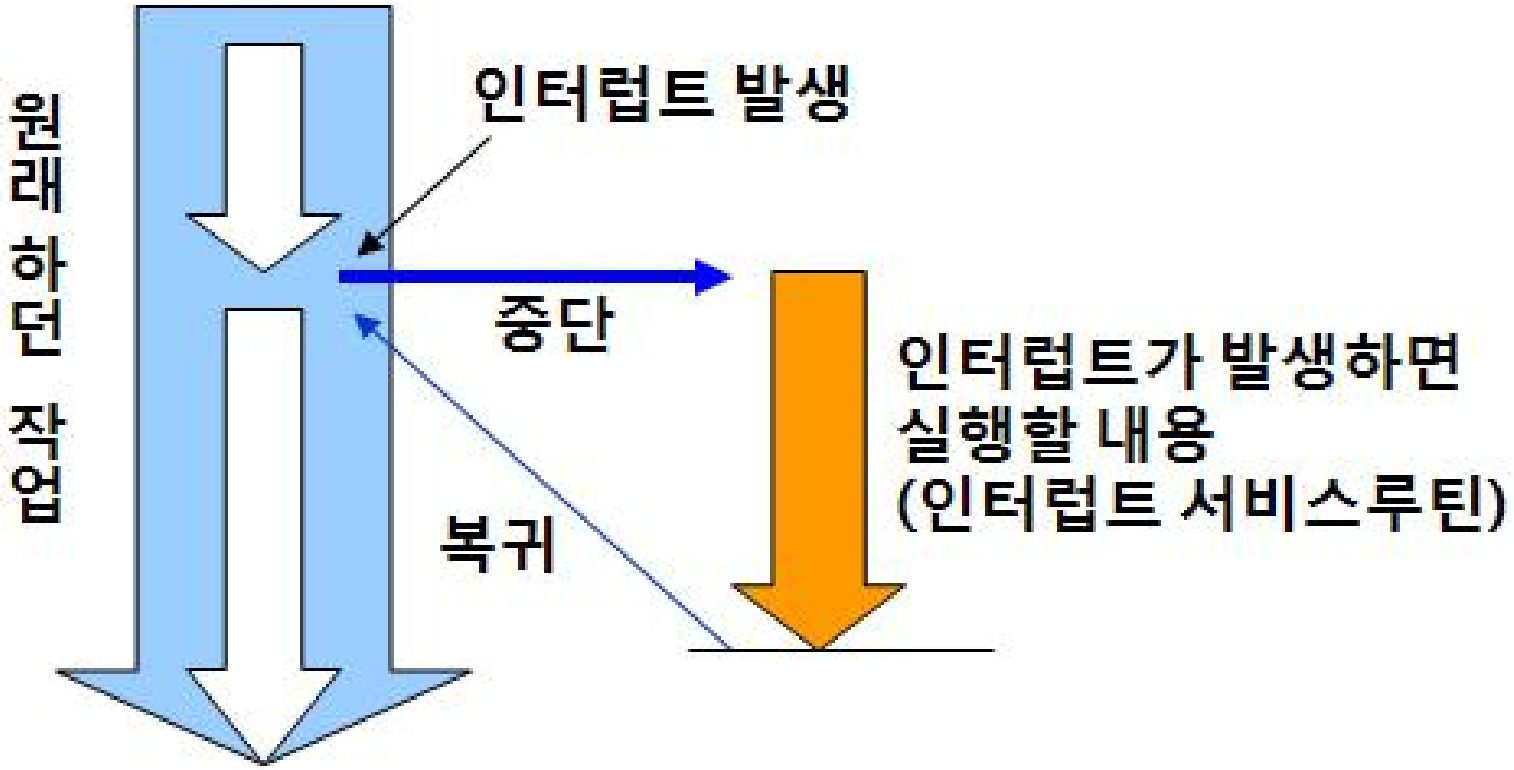


스텝모터 제어하기

```
#include <Stepper.h>
#define STEPS 2037    // 한바퀴를 이루는 스텝 갯수 입력
Stepper stepper(STEPS, 8, 10, 9, 11);  // 고정자 권선 순서 설정
void setup() {
  stepper.setSpeed(12); // 회전 속도 지정
}

void loop() {
  stepper.step(STEPS); // 정방향 회전
  delay(1000);
  stepper.step(-STEPS); // 역방향 회전
  delay(1000);
}
```

인터럽트의 개념

- 마이크로 컨트롤러가 어떤 작업을 진행하고 있다가 갑자기 다른 일이 발생하여 먼저 처리해야 하는 상황이 발생할 경우
- 현재 수행중인 일을 잠시 중단하고 급한 일을 처리한 후 원래의 일을 다시 이어서 수행



인터럽트의 종류

- 내부 인터럽트(소프트웨어 인터럽트) = 타이머 인터럽트
  - ⇨ 특정한 시간이 되면 인터럽트가 발생
- 외부 인터럽트
  - ⇨ 하드웨어 / GPIO 상태를 모니터링

타이머인터럽트 사용

- Mstimer2 라이브러리 사용
  - ⇨ MsTimer2::set(ms, ISR)
    - 타이머 인터럽트를 사용할 때 필요한 타이머와 인터럽트 처리 함수 설정
      - ms : 타이머 시간
      - ISR : 인터럽트 루틴으로 정해진 타이머 마다 실행될 함수
  - ⇨ MsTimer2::start()
    - 타이머 인터럽트 실행
  - ⇨ MsTimer2::stop()하드웨어 / GPIO 상태를 모니터링
    - 타이머 인터럽트 정지

타이머인터럽트 사용

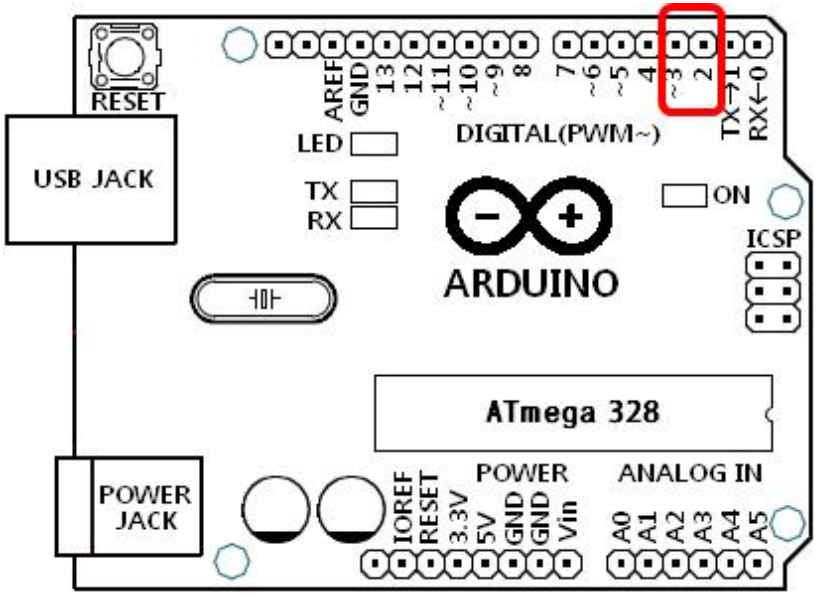
```
#include <MsTimer2.h>
void isr() {
  static boolean state = HIGH;
  digitalWrite(13, state);
  state = !state;
}
void setup() {
  pinMode(13, OUTPUT);
  MsTimer2::set(1000, isr);
  MsTimer2::start();
}
void loop() {
}
```

외부인터럽트 종류

외부입력 인식방법	동작 원리	처리방식
폴링 (polling)	사용자가 명령어를 통하여 입력 핀의 값을 계속 조사하여 변화를 알아내는 방식 (순차 실행)	소프트웨어적 방법
인터럽트 (interrupt)	MCU 자체가 하드웨어적으로 그 변화를 체크하여 변화가 발생하면 일정한 동작을 하는 방식	하드웨어적 방법

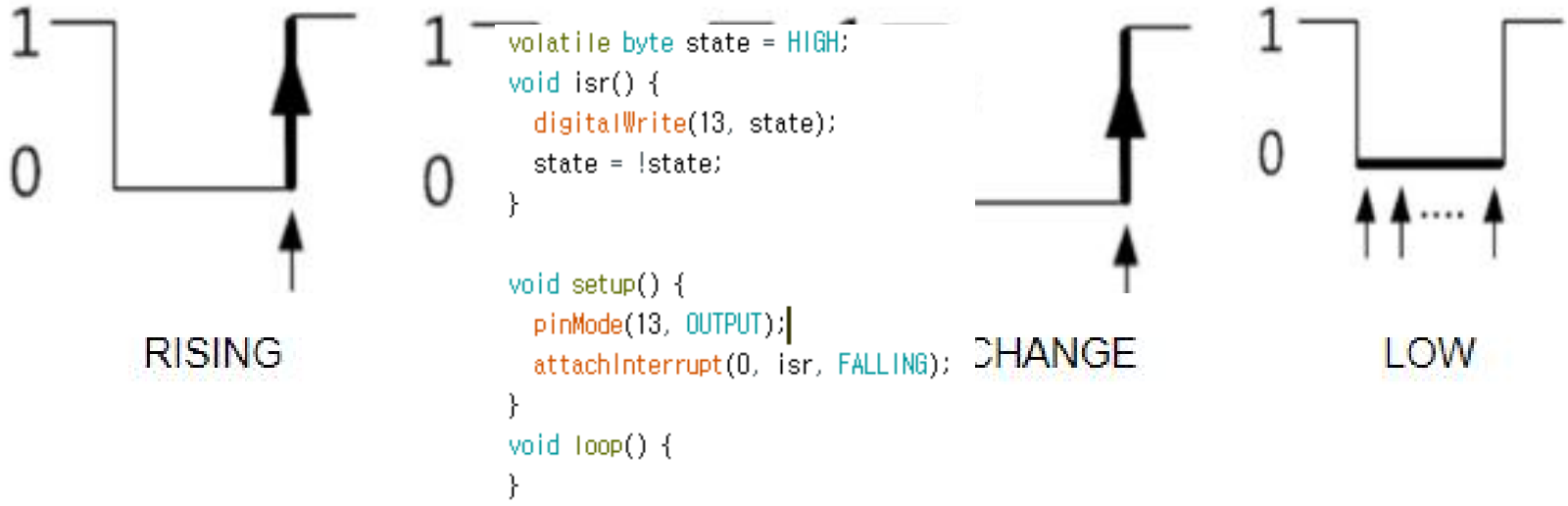
UNO 외부인터럽트

인터럽트 번호	0	1	2	3	4	5
UNO보드 핀 번호	2	3				
Mega보드 핀 번호	2	3	21	20	19	18



외부 인터럽트
UNO 보드 2개

- Mstimer2 라이브러리 사용
  - ⇨ attachInterrupt(interrupt, function, mode)
    - Interrupt : 인터럽트 번호(0, 1)
    - function : 인터럽트 서비스 루틴으로 정해진 모드에 따라 실행될 함수
    - Mode : RISING / FALLING / CHANGE / LOW



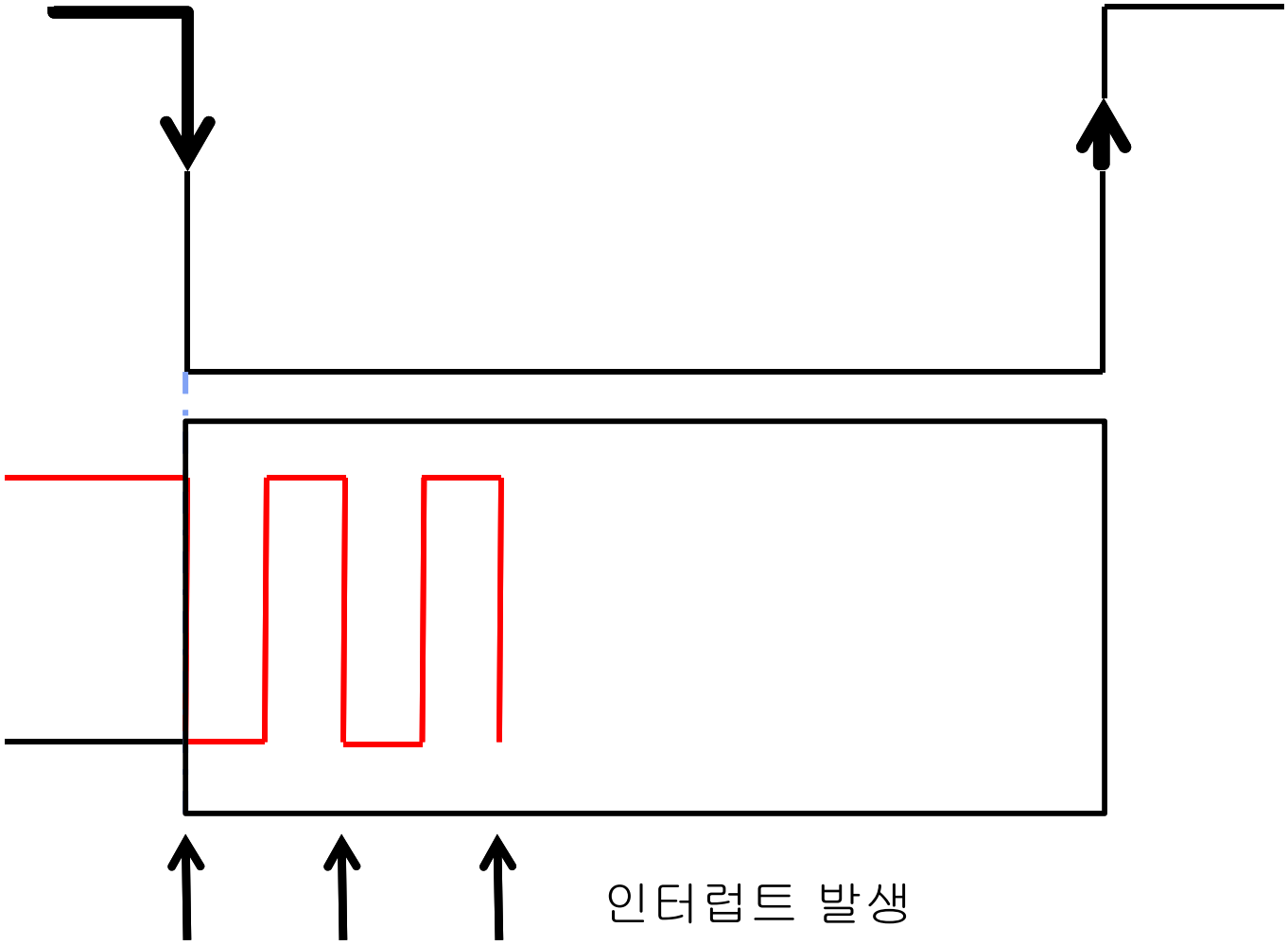

```
volatile byte state = HIGH;
void isr() {
  digitalWrite(13, state);
  state = !state;
}

void setup() {
  pinMode(13, OUTPUT);
  attachInterrupt(0, isr, FALLING);
}
void loop() {
}
```

외부인터럽트 사용

```
volatile byte state = HIGH;
void isr() {
  digitalWrite(13, state);
  state = !state;
}

void setup() {
  pinMode(13, OUTPUT);
  attachInterrupt(0, isr, FALLING);
}
void loop() {
}
```

외부인터럽트 사용(채터링 현상)

- 스위치의 상태가 변하는 순간 열림과 닫힘이 수회 반복되는 현상 또는 잡음



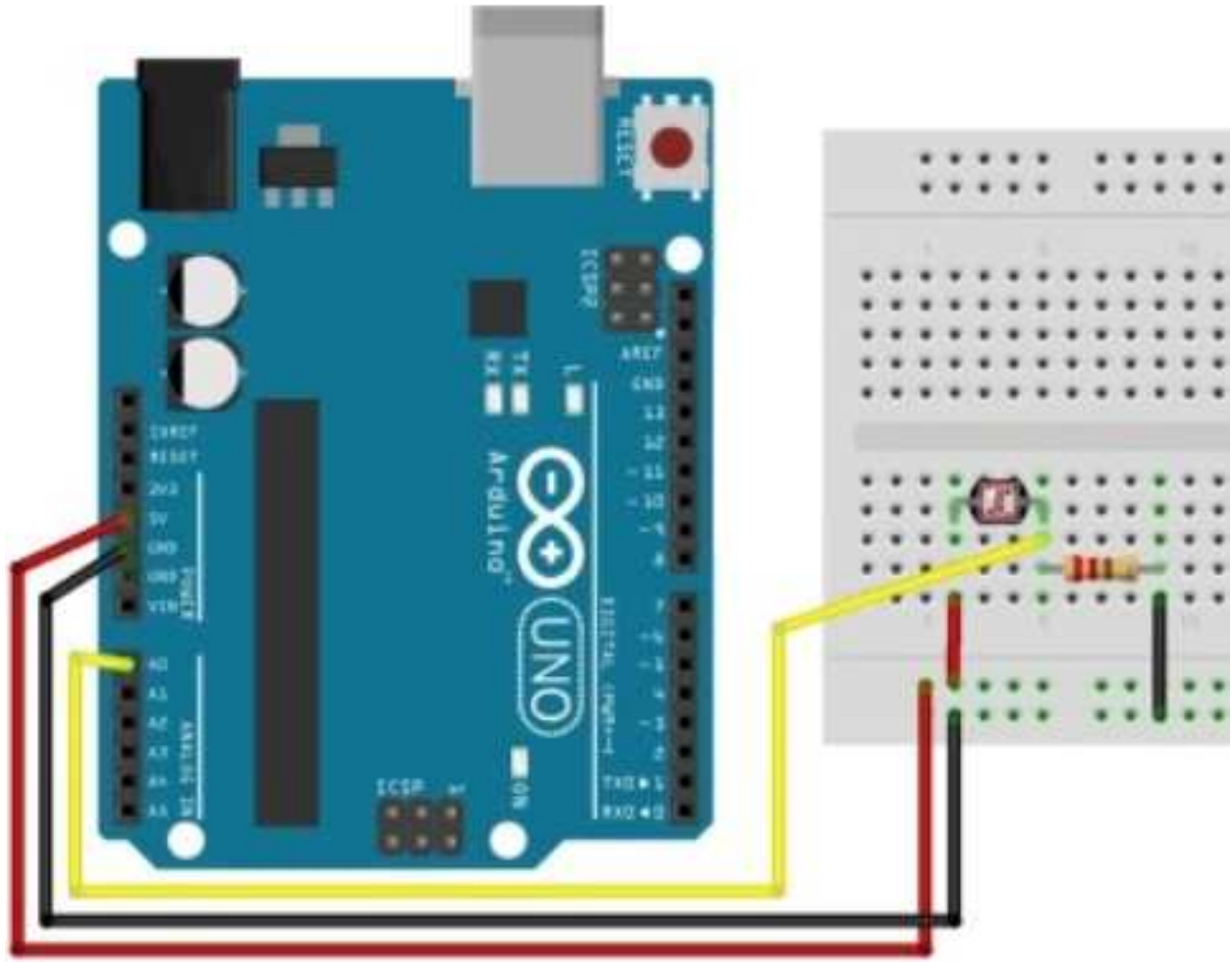
외부인터럽트 예제샘플

```
1  int duration = 500;
2  unsigned long pre_time = 0;
3  unsigned long cur_time = 0;
4
5  void isr()
6  {
7    cur_time = millis();
8    if(cur_time - pre_time >= duration)
9    {
10     Serial.println("push Button");
11     pre_time = cur_time;
12   }
13 }
14 void setup() {
15   Serial.begin(9600);
16   attachInterrupt(0, isr, FALLING);
17 }
18 void loop() {
19 }
```

CDS(조도센서)

❖ 측정된 빛의 세기에 따라 저항값이 변하는 센서

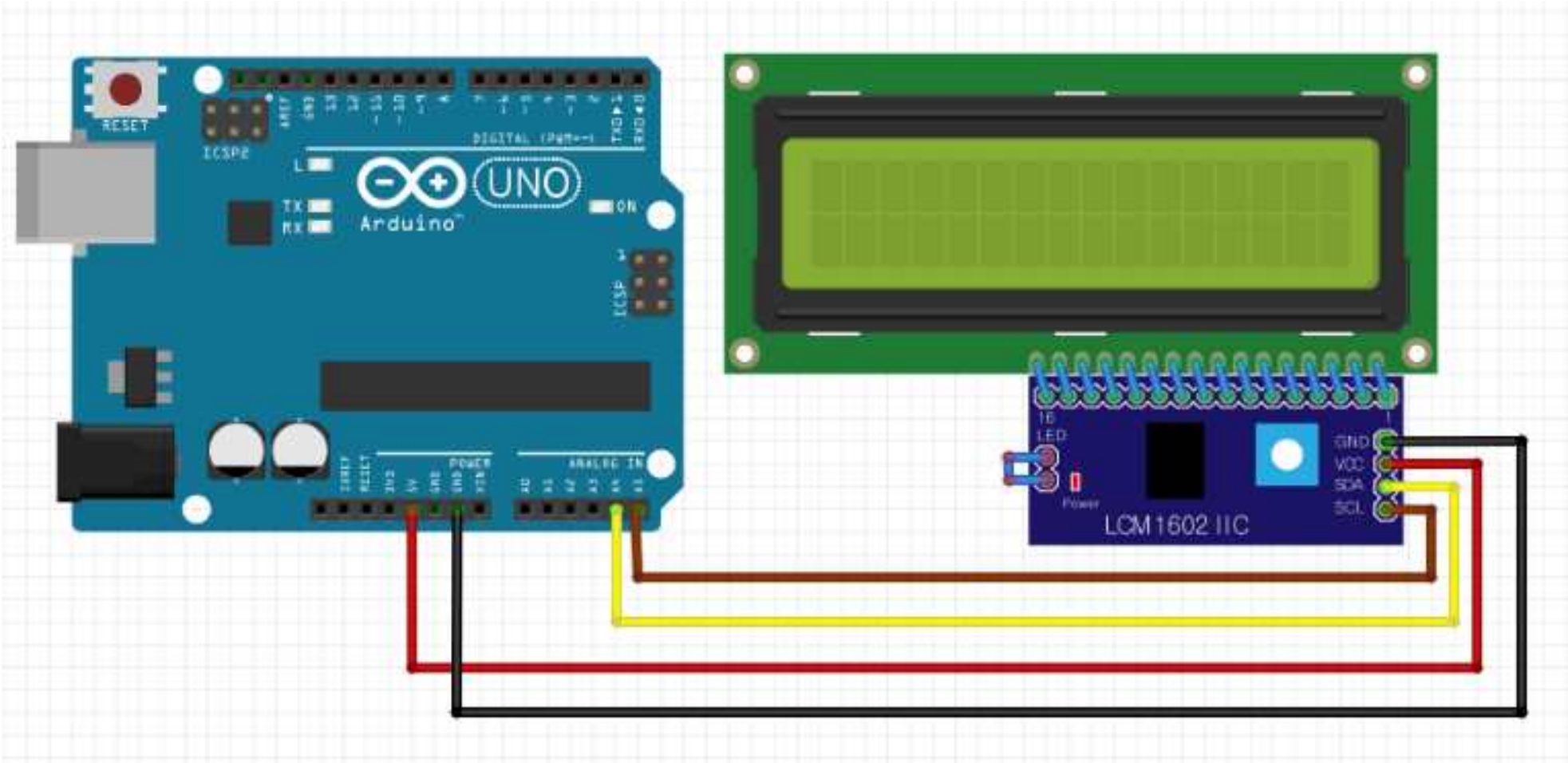
❖ 아날로그 입력



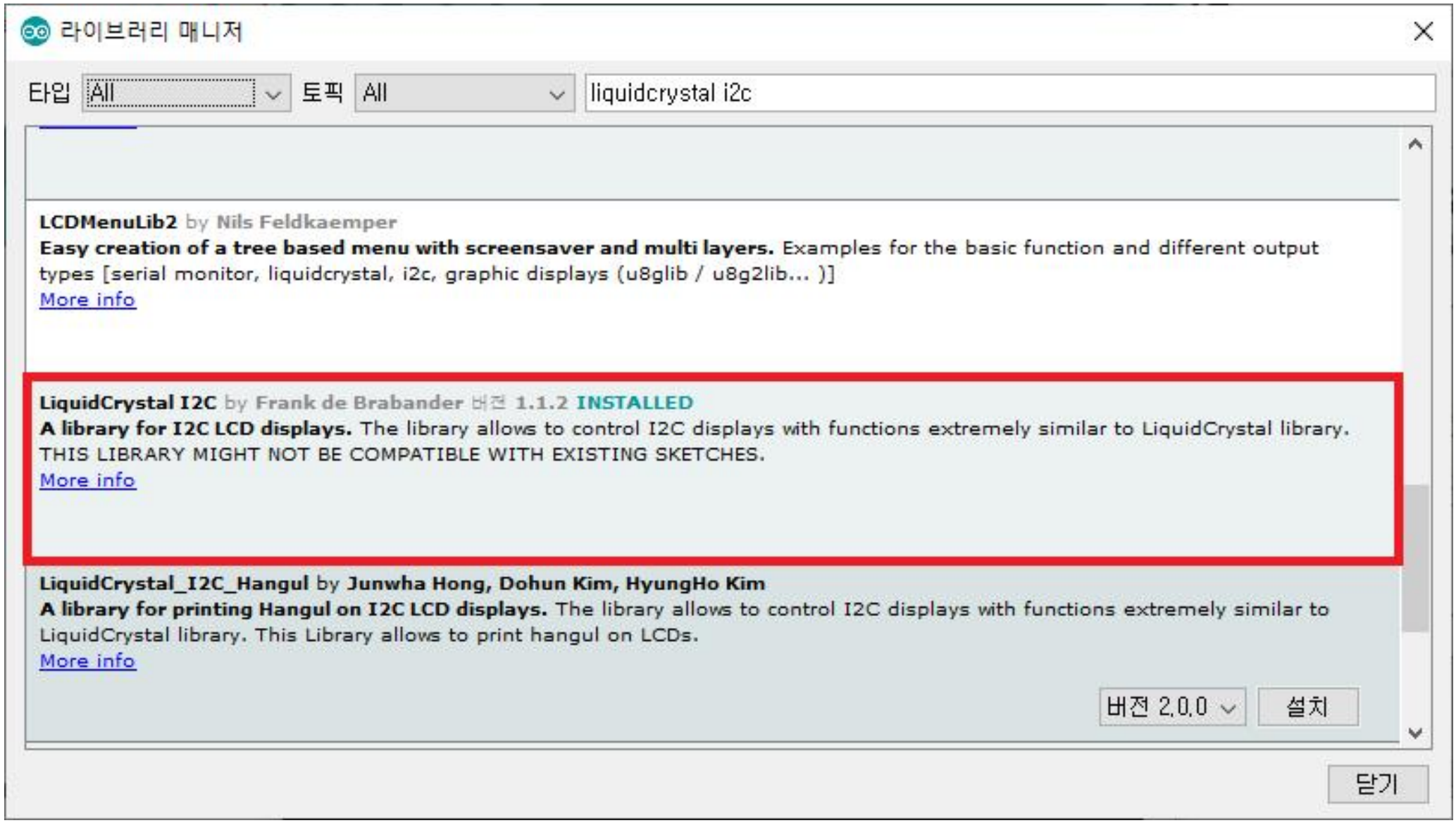
CDS(조도센서) 입력받기

```
1 void setup() {
2   Serial.begin(9600);
3 }
4 void loop() {
5   int val = analogRead(A0);
6   Serial.println(val);
7   delay(100);
8 }
```

Text LCD (연결)


RESET
ICSP2
UNO
Arduino
ON
TX
RX
ICSP
POWER
ANALOG IN
GND
VCC
SDA
SCL
LED
Power
LCM1602 IIC

Text LCD 라이브러리 추가



Text LCD 소스 코드

```
1  #include <LiquidCrystal_I2C.h>
2
3  LiquidCrystal_I2C lcd(0x27, 16, 2);
4
5  void setup() {
6    lcd.init();
7    lcd.backlight();
8    lcd.print("hello, world!");
9  }
10
11 void loop() {
12   lcd.setCursor(0,1);
13   lcd.print("2 line, INPUT");
14 }
```

Text LCD (슬레이브 주소 확인)

```
#include <Wire.h>

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

void loop() {
  byte error, address;
  int nDevices = 0;
  for(address = 1; address < 127; address++)
  {
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,BIN);
      Serial.println("  !");

      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found₩n");
  else
    Serial.println("done₩n");

  delay(5000);           // wait 5 seconds for next scan
}
```

피에조 부저

- 능동부저(Active Buzzer) – 극성이 있으며 스스로 소리를 냄
- 수동부저(Passive Buzzer) – 소리를 낼 수 없어 주파수를 만들어 사용

Active Buzzer

Passive Buzzer

Piezo Buzzer

(단위 : Hz)

옥타브 / 음계	1	2	3	4	5	6	7	8
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133

Piezo Buzzer

- 능동부저(Active Buzzer)
  - 제어명령 digitalWrite
- 수동부저(Passive Buzzer)
  - tone(pin, frequency) tone(pin, frequency, duration)
    - 핀번호 – 주파수 신호를 출력할 핀 번호를 설정 (PWM핀)
    - 주파수 – 31~65535 범위의 주파수(Hz)를 설정
    - 출력시간 – 밀리초단위(1000당 1초)입력, 얼마동안 주파수를 출력할지 설정
  - noTone(pin, frequency)
    - 주파수 신호 출력을 중지하기 위한 함수

Piezo Buzzer 제어하기

❖ 능동부저(Active Buzzer)

```
1 void setup() {
2   pinMode(5,OUTPUT);
3 }
4 void loop() {
5   digitalWrite(5, HIGH);
6   delay(1000);
7   digitalWrite(5, LOW);
8   delay(1000);
9 }
```

❖ 수동부저(Passive Buzzer)

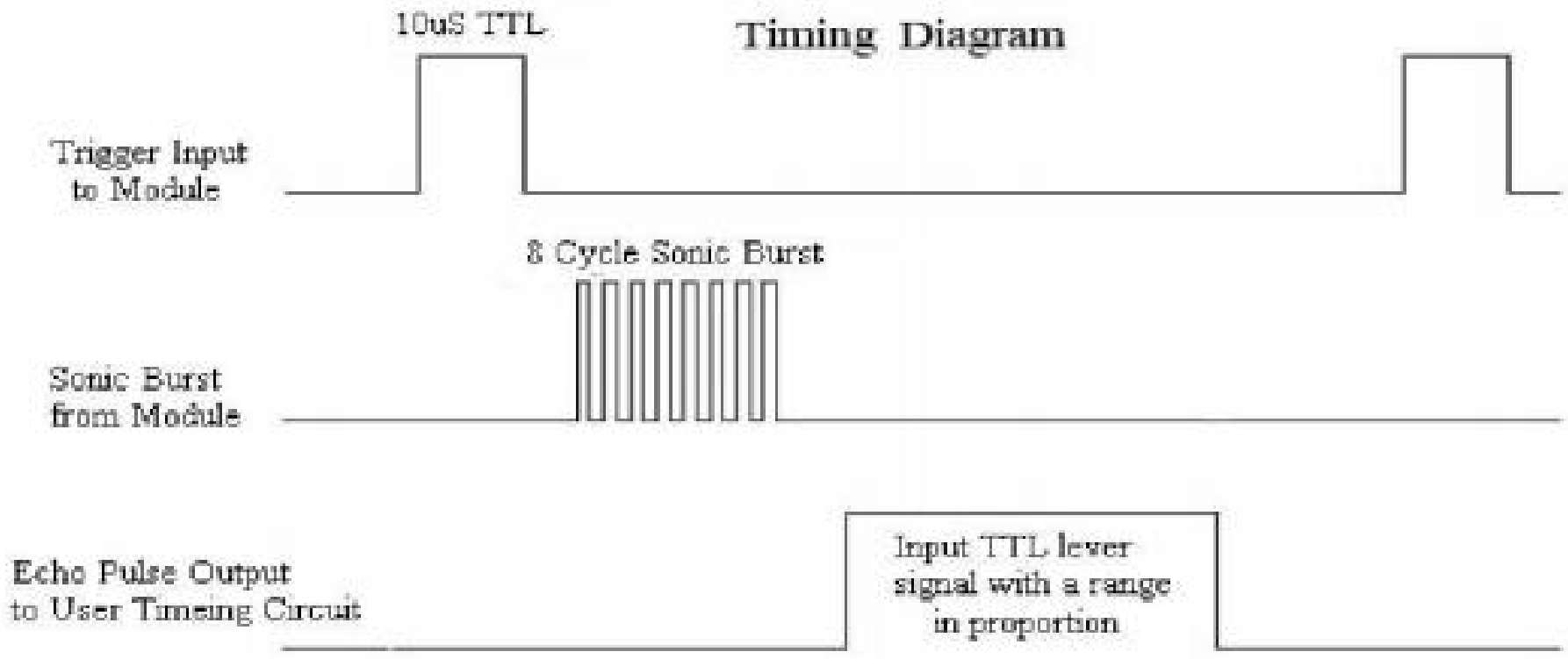
```
1  void setup() {
2    pinMode(5,OUTPUT);
3  }
4  void loop() {
5    tone(5,261);
6    delay(1000);
7    noTone(5);
8    delay(1000);
9
10   tone(5,294);
11   delay(1000);
12   noTone(5);
13   delay(1000);
14 }
```

초음파 센서(SR04)

- 20kHz 이상의 주파수를 가진 소리
  - 송신부와 송신부로 구성
  - 음파의 속도는 340m/s

초음파 센서(SR04)

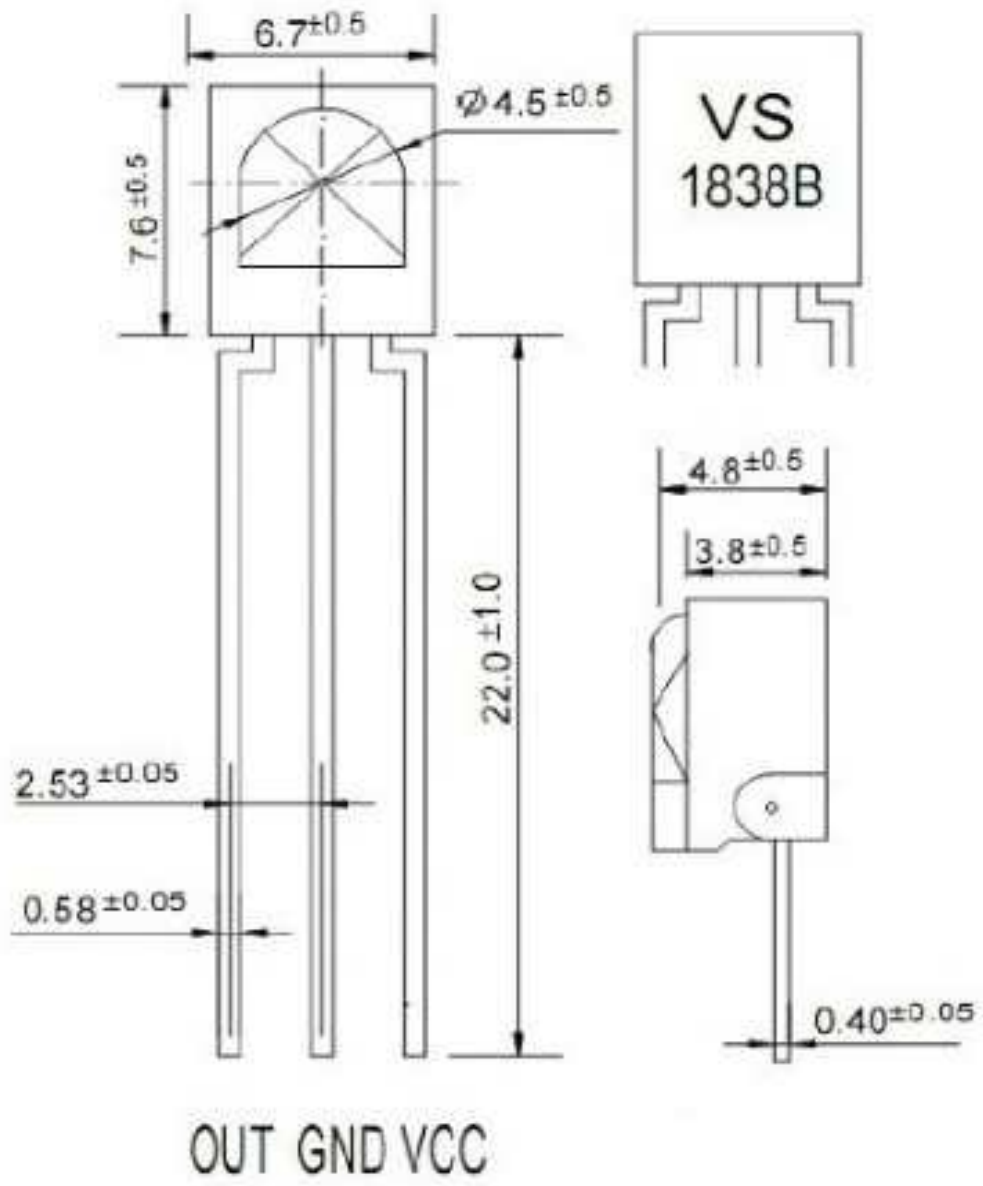
- 1cm를 이동하는데 걸리는 시간은 약 29us
  - (이동하는데 걸린 시간) / 29 / 2 = CM거리 값



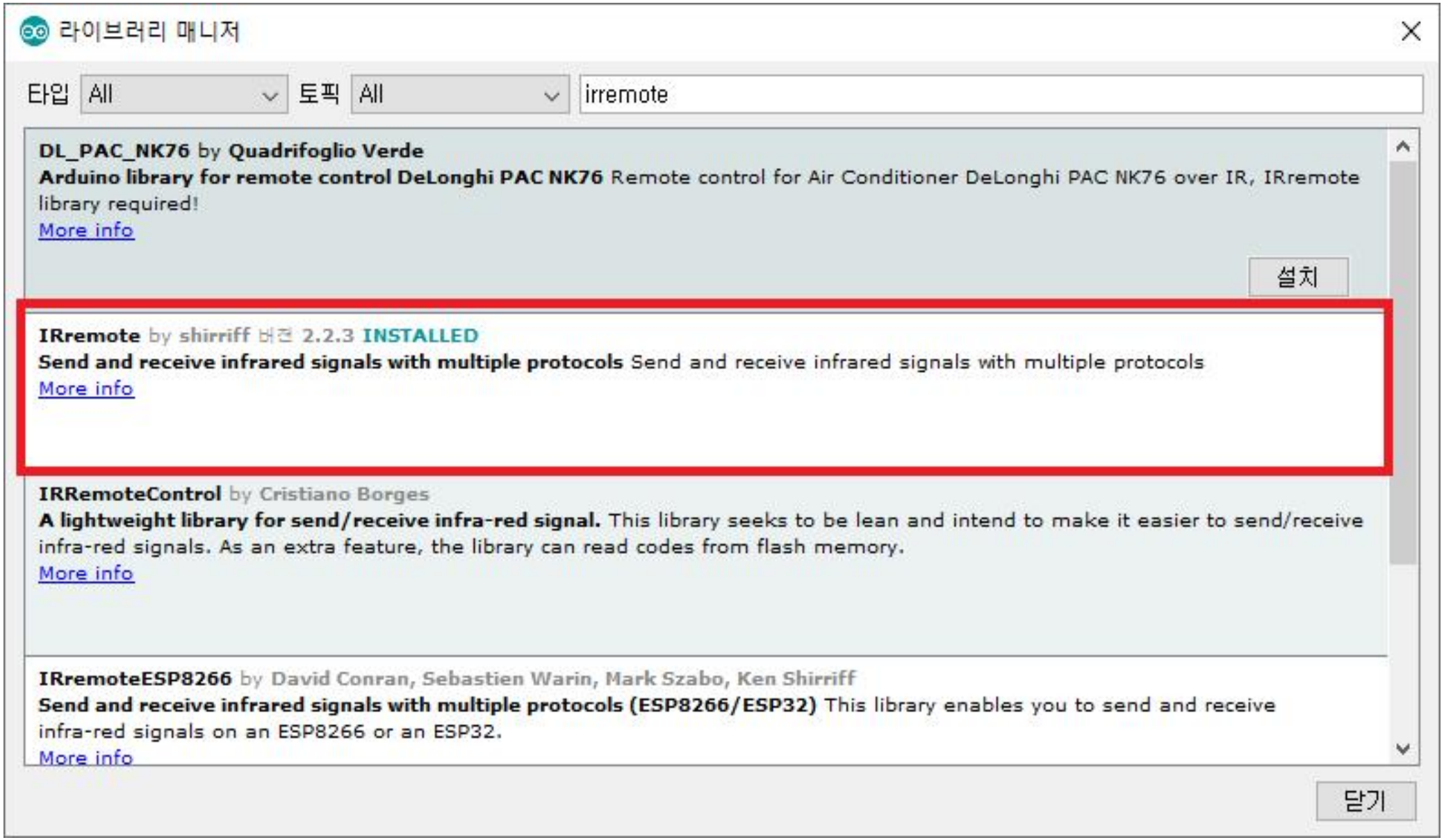
초음파센서 제어하기

```
1  int trig = 4;
2  int echo = 3;
3  void setup() {
4    pinMode(trig,OUTPUT);
5    pinMode(echo,INPUT);
6    Serial.begin(9600);
7  }
8  void loop() {
9    digitalWrite(trig, LOW);
10   delayMicroseconds(2);
11   digitalWrite(trig, HIGH);
12   delayMicroseconds(10);
13   digitalWrite(trig, LOW);
14   long duration = pulseIn(echo, HIGH);
15   long dis_cm = duration/29/2;
16   Serial.println(dis_cm);
17   delay(100);
18 }
```

적외선 센서


6.7±0.5
Ø4.5±0.5
7.6±0.5
VS
1838B
4.8±0.5
3.8±0.5
22.0±1.0
2.53±0.05
0.58±0.05
0.40±0.05
OUT GND VCC

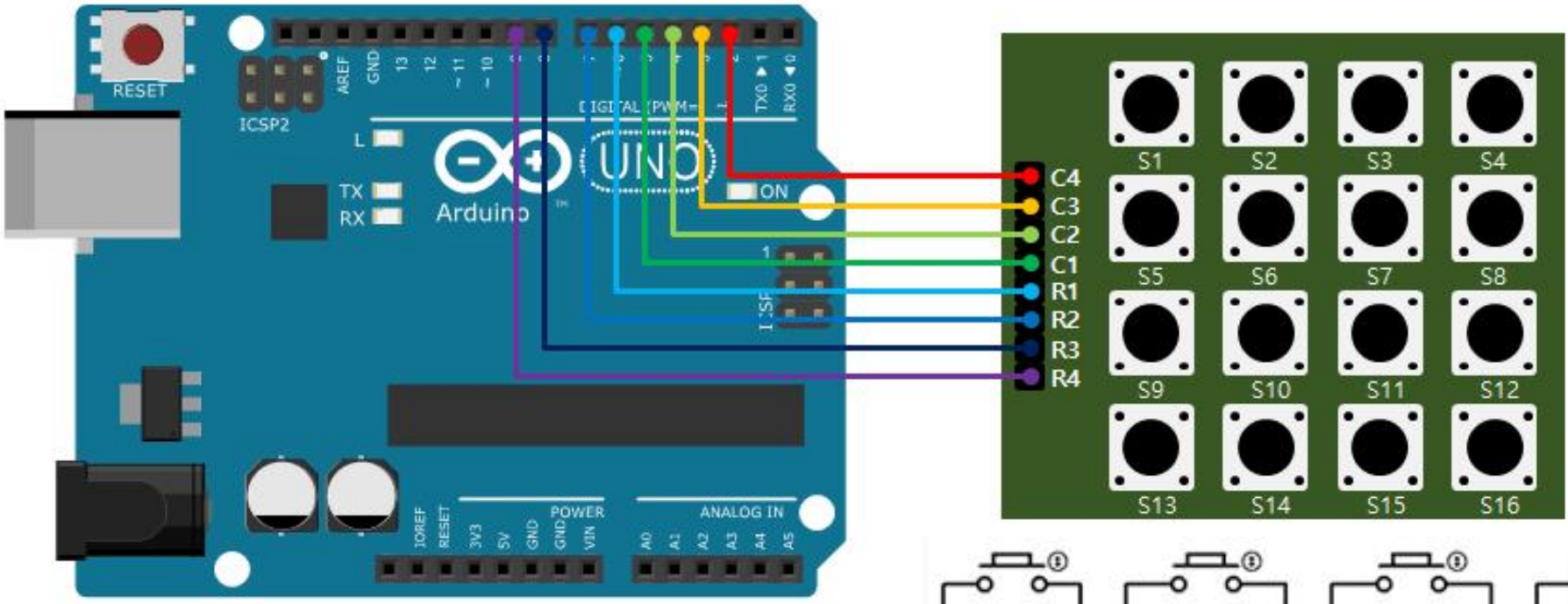
적외선 센서 라이브러리 추가

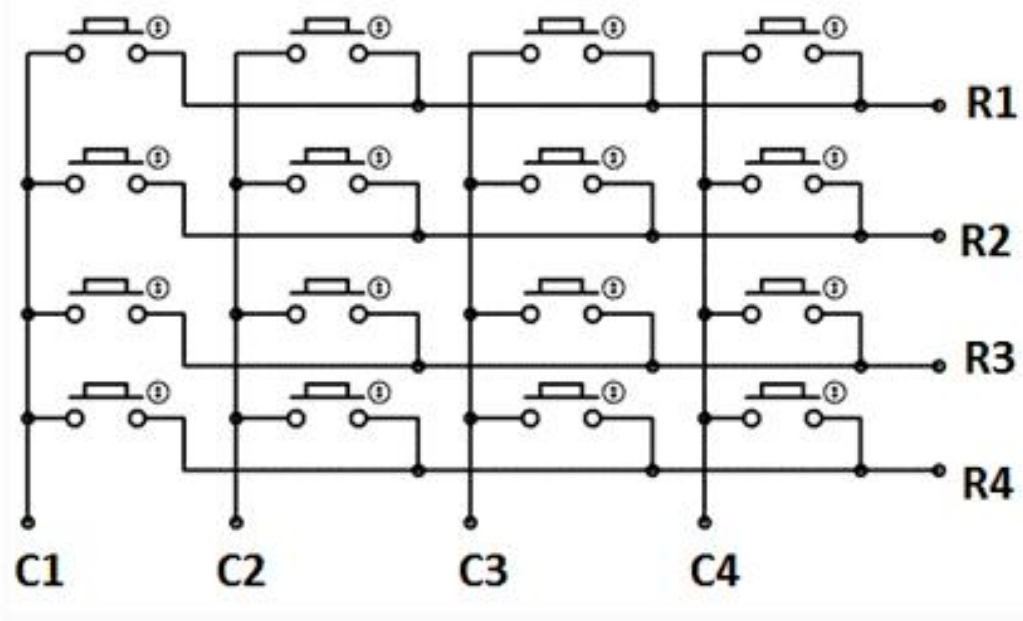


적외선 센서 코드

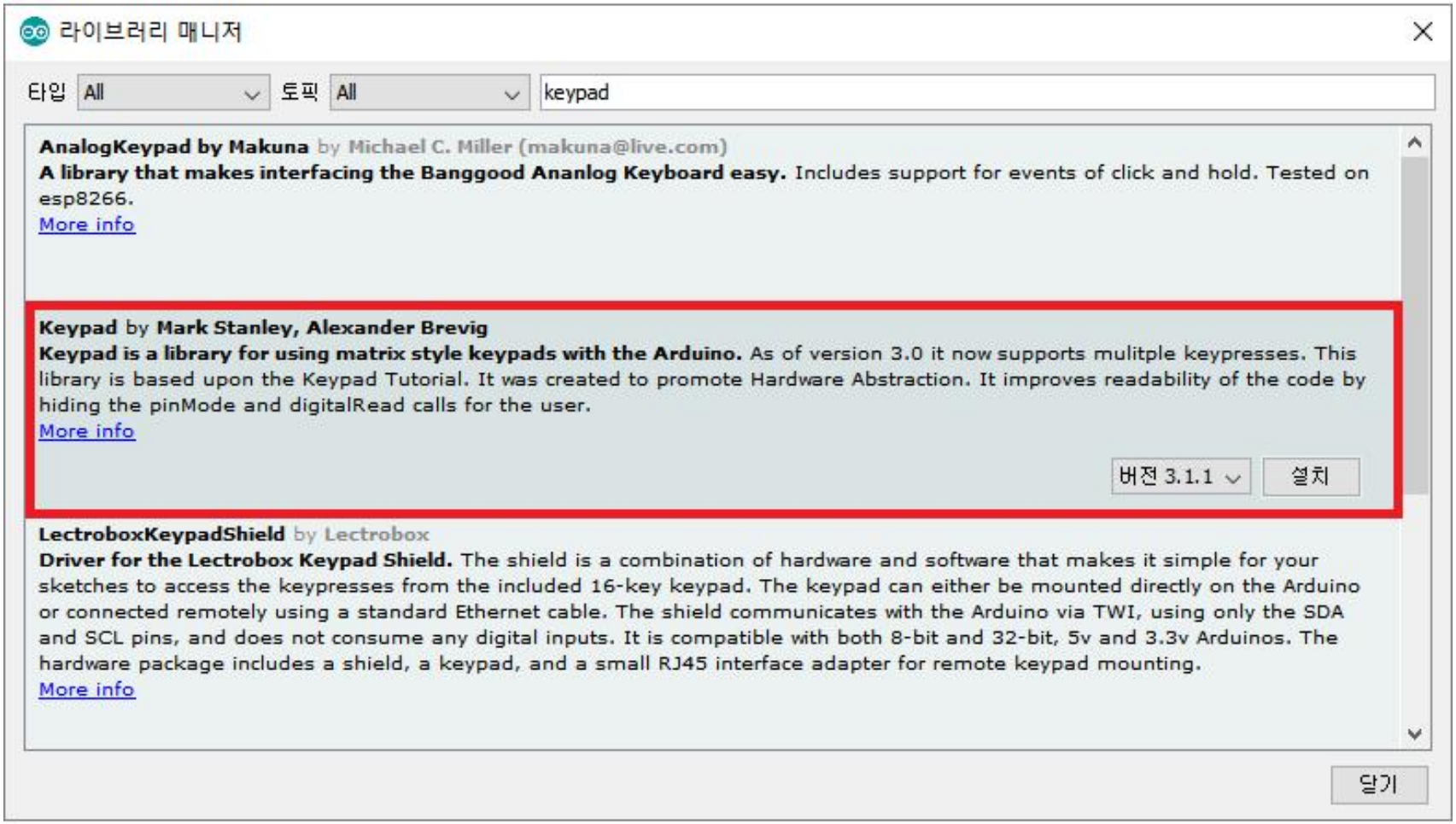
```
1  #include <IRremote.h>
2  int RECV_PIN=2;
3  IRrecv irrecv(RECV_PIN);
4  decode_results results;
5  void setup() {
6    Serial.begin(9600);
7    irrecv.enableIRIn();  //Start the receiver
8  }
9
10 void loop() {
11   if(irrecv.decode(&results)){
12     Serial.println(results.value, HEX);
13     irrecv.resume();     //Receive the next value
14   }
15   delay(100);
16 }
```

아두이노 4x4 KEYPAD





아두이노 4x4 KEYPAD



아두이노 4x4 코드

```
#include <Keypad.h>
const byte ROWS = 4;     // 행(rows) 개수
const byte COLS = 4;     // 열(columns) 개수
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {6, 7, 8, 9};    // R1, R2, R3, R4 단자가 연결된 아두이노 핀 번호
byte colPins[COLS] = {5, 4, 3, 2};    // C1, C2, C3, C4 단자가 연결된 아두이노 핀 번호
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
void setup() {
  Serial.begin(9600);
}
void loop() {
  char key = keypad.getKey();
  if (key) {
    Serial.println(key);
  }
}
```

아두이노 4x4 활용

```
#include <Servo.h>
#include <Keypad.h>

Servo myservo;
char* secretCode = "1234";  // 비밀번호를 설정(여기선 1234)
int position = 0;
int wrong = 0; // 입력 초기화
 const byte ROWS = 4;    // 행(rows) 개수
const byte COLS = 4;    // 열(columns) 개수

char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {6, 7, 8, 9};   // R1, R2, R3, R4 단자가 연결된 아두이노 핀 번호
byte colPins[COLS] = {5, 4, 3, 2};   // C1, C2, C3, C4 단자가 연결된 아두이노 핀 번호

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  myservo.attach(11);
}
```

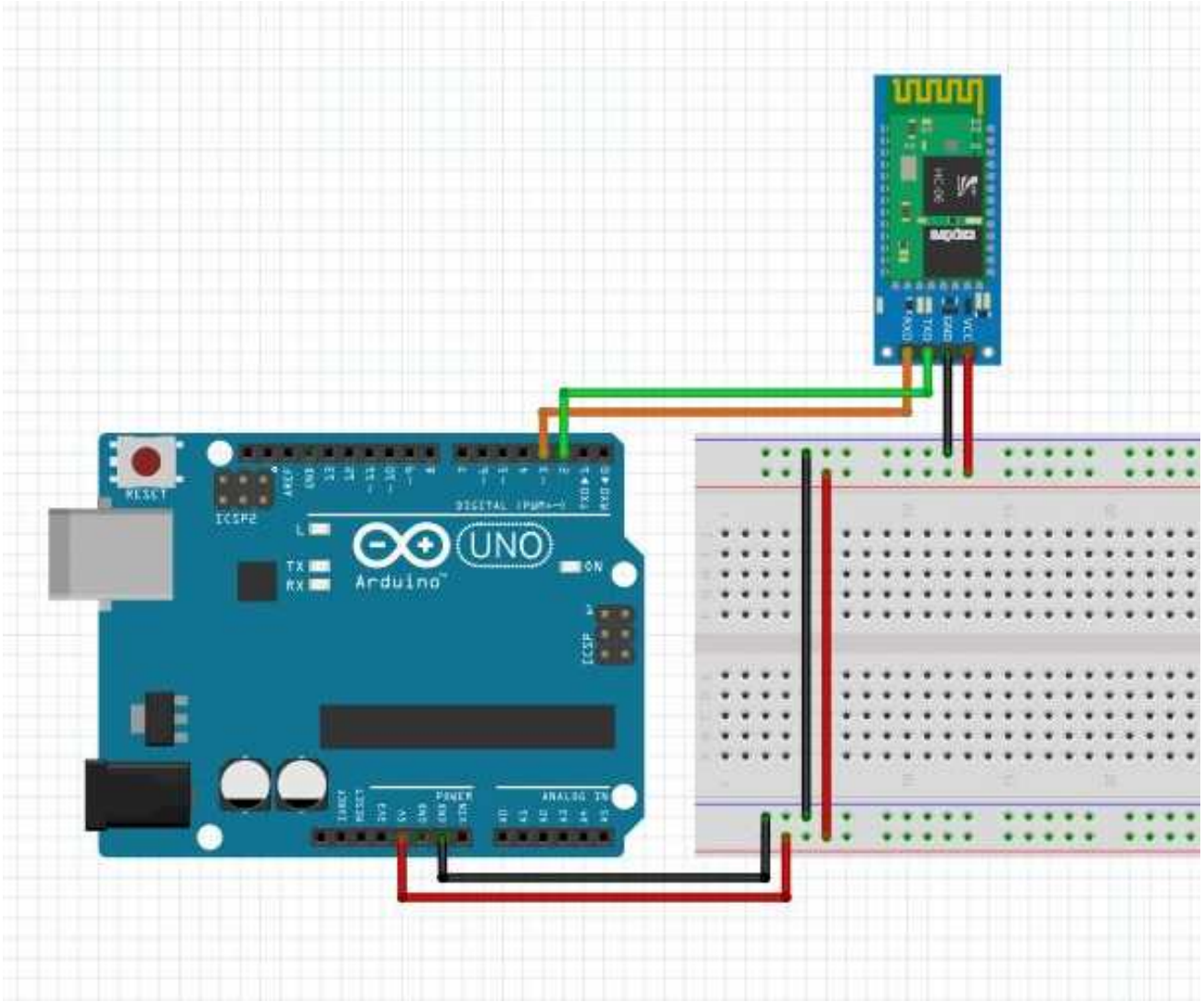
아두이노 4x4 활용

```
void loop() {
  char key = keypad.getKey();

   if((key >= '0' && key <= '9') || (key >= 'A' && key <='D')
    || (key == '*' || key == '#')){   // 키패드에서 입력된 값을 조사하여 맞게 입력된 값일 경우(키패드에
                                      있는 버튼이 맞을경우) 비교
      Serial.println(key);
      if(key == '*' || key == '#'){ // *, # 버튼을 눌렀을 경우
        position = 0;
        wrong = 0; // 입력 초기화
        setLocked(true); // 잠금상태로 세팅
        }
      else if(key == secretCode[position]){ // 해당 자리에 맞는 비밀번호가 입력됐을 경우
      position++; // 다음 자리로 넘어 감
      wrong = 0; // 비밀번호 오류 값을 0으로 만듬
      }

      else if(key != secretCode[position]){ // 해당 자리에 맞지 않는 비밀번호가 입력됐을 경우
      position = 0; // 비밀번호를 맞았을 경우를 0으로 만듬
      setLocked(true); // 잠금상태로 세팅
      wrong++; // 비밀번호 오류 값을 늘려준다
      }
       if(position == 4){ // 4자리 비밀번호가 모두 맞았을 경우
      setLocked(false); // 잠금상태를 해제 함
      Serial.println("FULL INPUT");
       position = 0;
       }
```

아두이노 4x4 활용

```
if(wrong == 4){ // 비밀번호 오류를 4번 했을 경우
      blink(); // LED를 깜빡여 준다.
      wrong = 0; // 비밀번호 오류 값을 0으로 만들어 준다.
    }
  }
  delay(100);
}
void setLocked(int locked){ // 잠금시와 해제시에 맞는 LED를 세팅해 주는 함수
  if(locked) { //잠금모드
     myservo.write(0);
     Serial.println("LOCK");
  }
  else{ // 잠금해제
     myservo.write(180);
     Serial.println("UNLOCK");
  }
}
void blink(){ // 비밀번호 4번 오류시 LED를 깜빡여 주는 함수.
  for(int i = 0; i < 5; i++){ // 딜레이 만큼 LED를 껐다 켰다 해준다.
    digitalWrite(13, LOW);
    delay(500);
    digitalWrite(13, HIGH);
    delay(500);
  }
}
```

블루투스


RESET
ICSP2
UNO
Arduino
TX
RX
ON
ICSP
POWER
ANALOG IN
DIGITAL (PWM~)
TXD
RXD
VCC
GND
STATE
EN

블루투스(AT 명령)

```
#include <SoftwareSerial.h>
int TX = 2;
int RX = 3;
SoftwareSerial my_blue(T, R);
void setup(){
  Serial.begin(9600);
  my_blue.begin(9600);
}
void loop(){
  if (my_blue.available()) {
    Serial.write(my_blue.read());
  }
  if (Serial.available()) {
    my_blue.write(Serial.read());
  }
```

1. 테스트
   명령어 : AT
   반환값 : OK

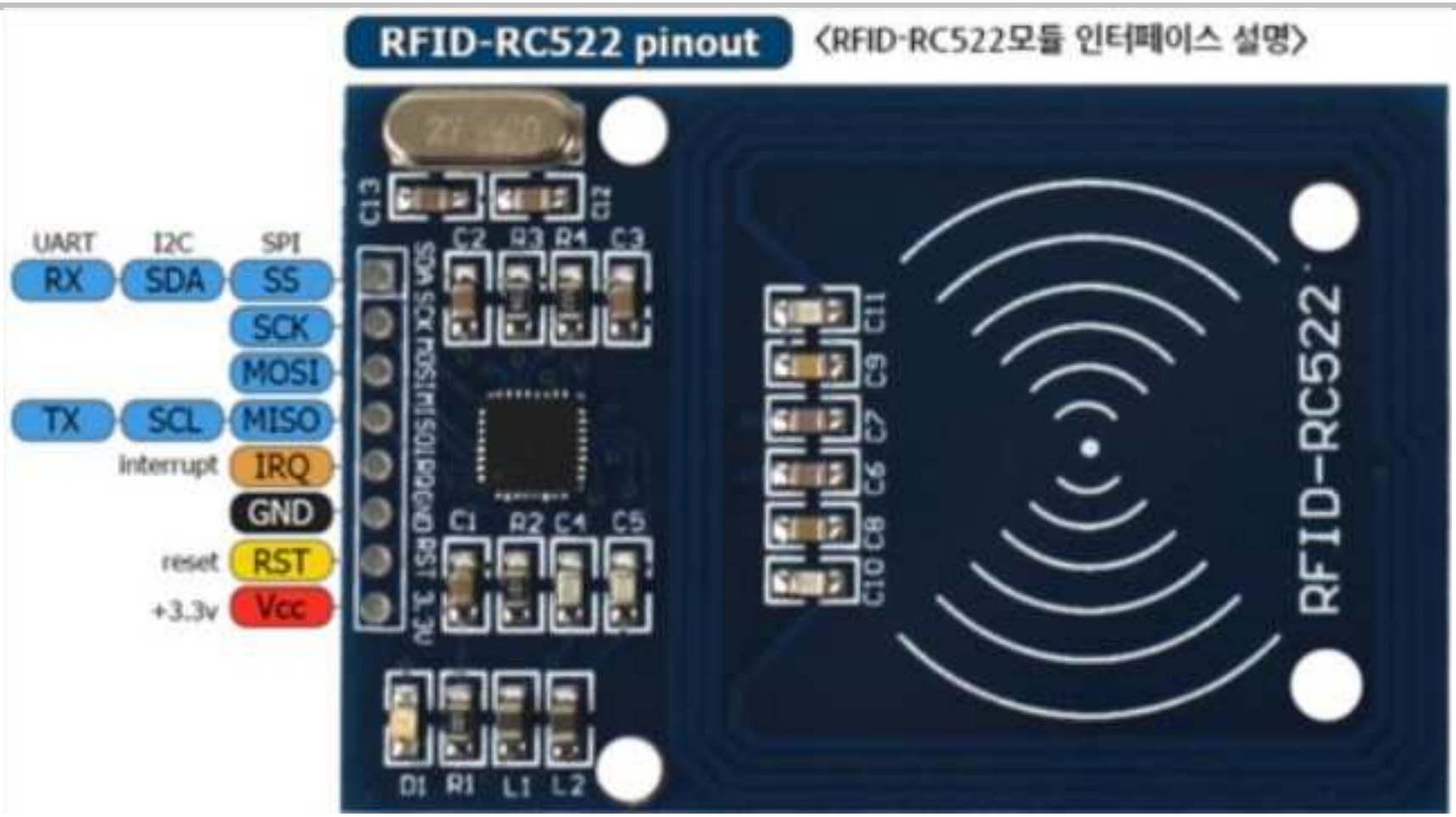
2. 버전 확인
   명령어 : AT+VERSION
   반환값 : OKlinvorV1.8

3. 이름 변경
   명령어 : AT+NAME이름
   반환값 : OKsetname

4. 보레이트 설정
   명령어 : AT+BAUD보레이트 메뉴값.
   보레이트 메뉴 :
   1 - 1200, 2 - 2400, 3 - 4800, 4 - 9600
   5 - 19200, 6 - 38400, 7 - 57600, 8 - 115200
   반환값 : OK보레이트 (예 : OK9600)

5. PIN설정
   명령어 : AT+PIN네 자리 숫자
   반환값 : OKsetPIN

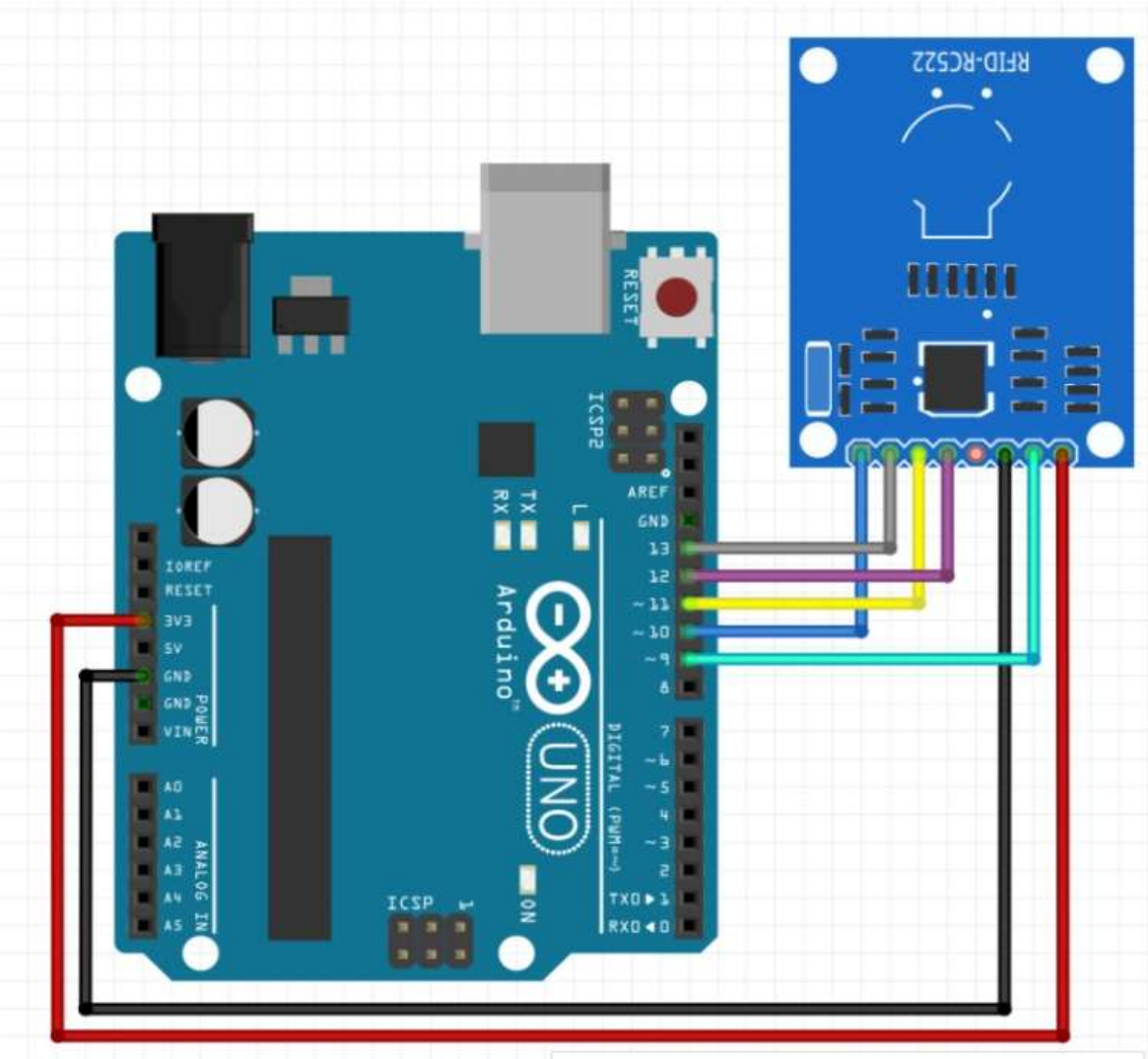
RFID-RC522


RFID-RC522 pinout
〈RFID-RC522모듈 인터페이스 설명〉
UART
I2C
SPI
RX
SDA
SS
SCK
MOSI
TX
SCL
MISO
interrupt
IRQ
GND
reset
RST
+3.3v
Vcc
RFID-RC522


라이브러리 매니저
타입 All
토픽 All
rc522
MFRC522 by GithubCommunity 버전 1.4.4 INSTALLED
Arduino RFID Library for MFRC522 (SPI) Read/Write a RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface.
More info
버전 선택
설치

RFID 연결도

1. 연결도



#1, SDA - 10~
#2, SCK - 13
#3, MOSI - 11
#4, MISO - 12
#5, IRQ - N.C(연결안함)
#6, GND - GND
#7, RST - 9
#8, 3.3V - 3.3V

RC522 카드 UID 확인

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN   9
#define SS_PIN    10

MFRC522 mfrc(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc.PCD_Init();
}

void loop() {

if(!mfrc.PICC_IsNewCardPresent())
        return;
 if(!mfrc.PICC_ReadCardSerial())
        return;
  Serial.print("Card UID:");
  for (byte i = 0; i < 4; i++) {
  {
    Serial.print(mfrc.uid.uidByte[i]);
    Serial.print(" ");
  }
  Serial.println();
}
```

RC522 카드 인식

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN   9
#define SS_PIN    10

MFRC522 mfrc(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc.PCD_Init();
}

void loop() {

if(!mfrc.PICC_IsNewCardPresent())
        return;
 if(!mfrc.PICC_ReadCardSerial())
        return;
if(mfrc.uid.uidByte[0] == 244 &&
   mfrc.uid.uidByte[1] == 181 &&
   mfrc.uid.uidByte[2] == 249 &&
   mfrc.uid.uidByte[3] == 233)
   {
     Serial.println("Authorized access");
     delay(500);
   }else {
    Serial.println("Access denied");
    delay(500);
  }
}
```