



CNRLAB

CIM&ROBOTICS LABORATORY

로봇공학입문설계

2주차 아두이노 기본교육

로봇공학과

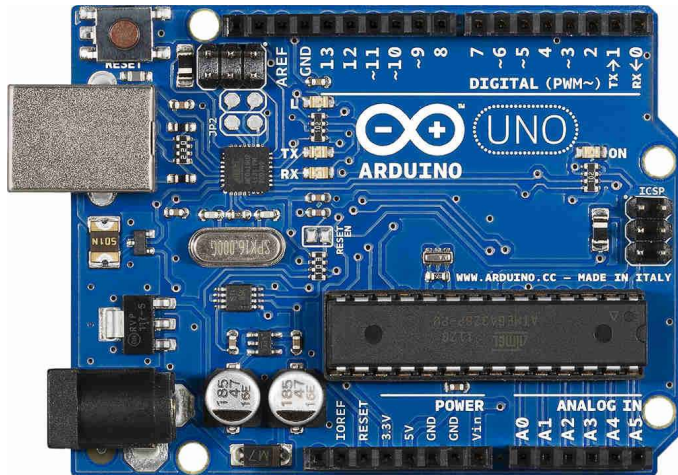
Contents

- 아두이노 시작하기
- Digital I/O
- Analog I/O
- 도전과제

아두이노 시작하기

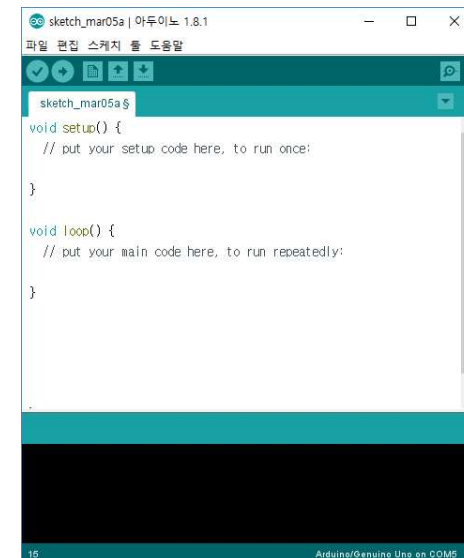
□ 구성

Hardware
(Board)



+

Software
(Arduino IDE)



IDE(Integrated Development Environment)

- 코딩, 디버그, 컴파일 등 모든 작업을 하나의 프로그램 안에서 처리할 수 있는 **프로그래밍 개발환경**.
- Arduino IDE에서 작성하는 프로그램을 '**Sketch**' 라고 부른다.

아두이노 시작하기

1. 아두이노 IDE 설치

<http://www.arduino.cc/en/Main/Donate>

2. USB 케이블 연결

3. 드라이버 설치

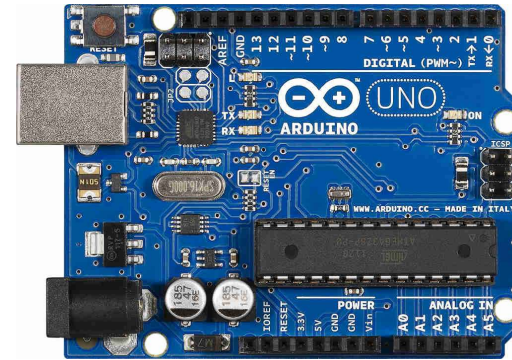
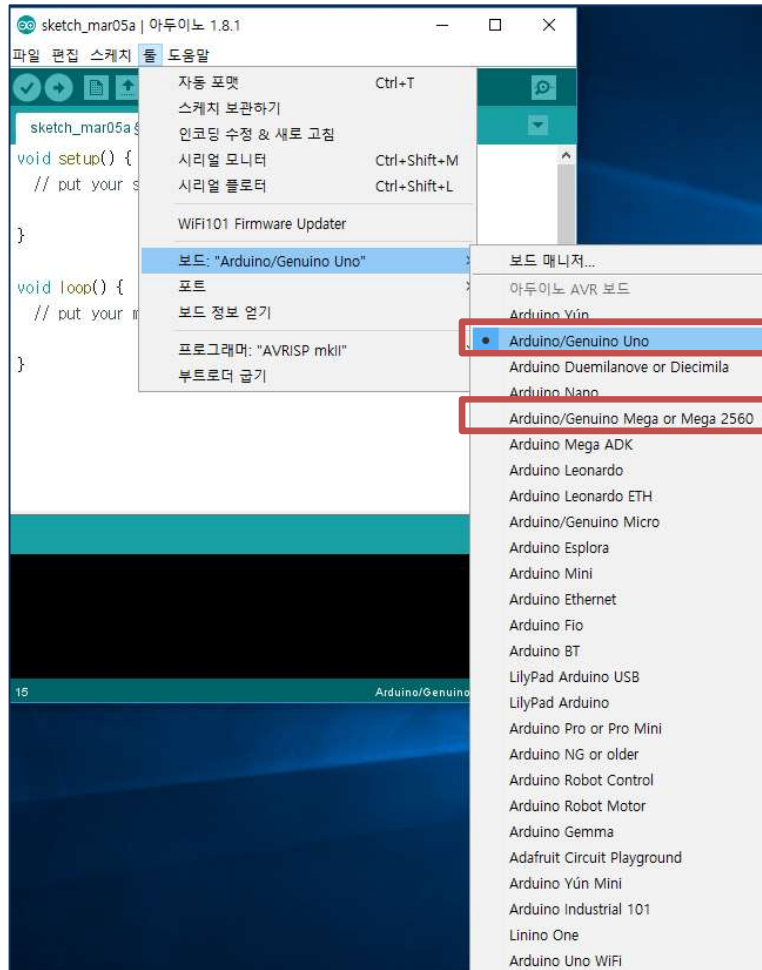
4. 아두이노 IDE 실행



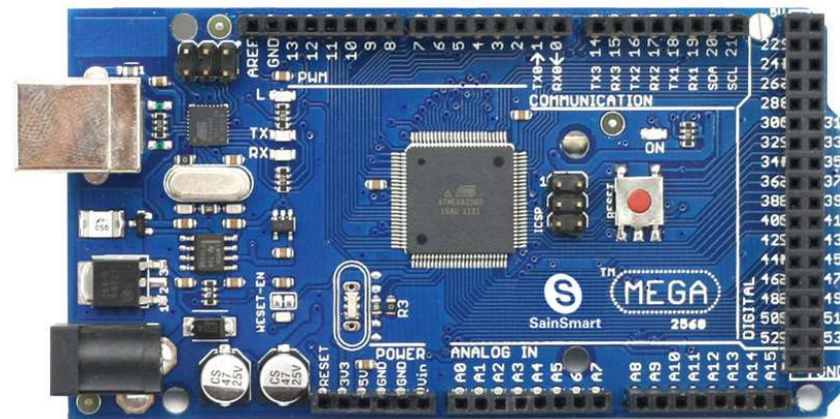
아두이노 IDE

아두이노 시작하기

5. 보드 종류 선택



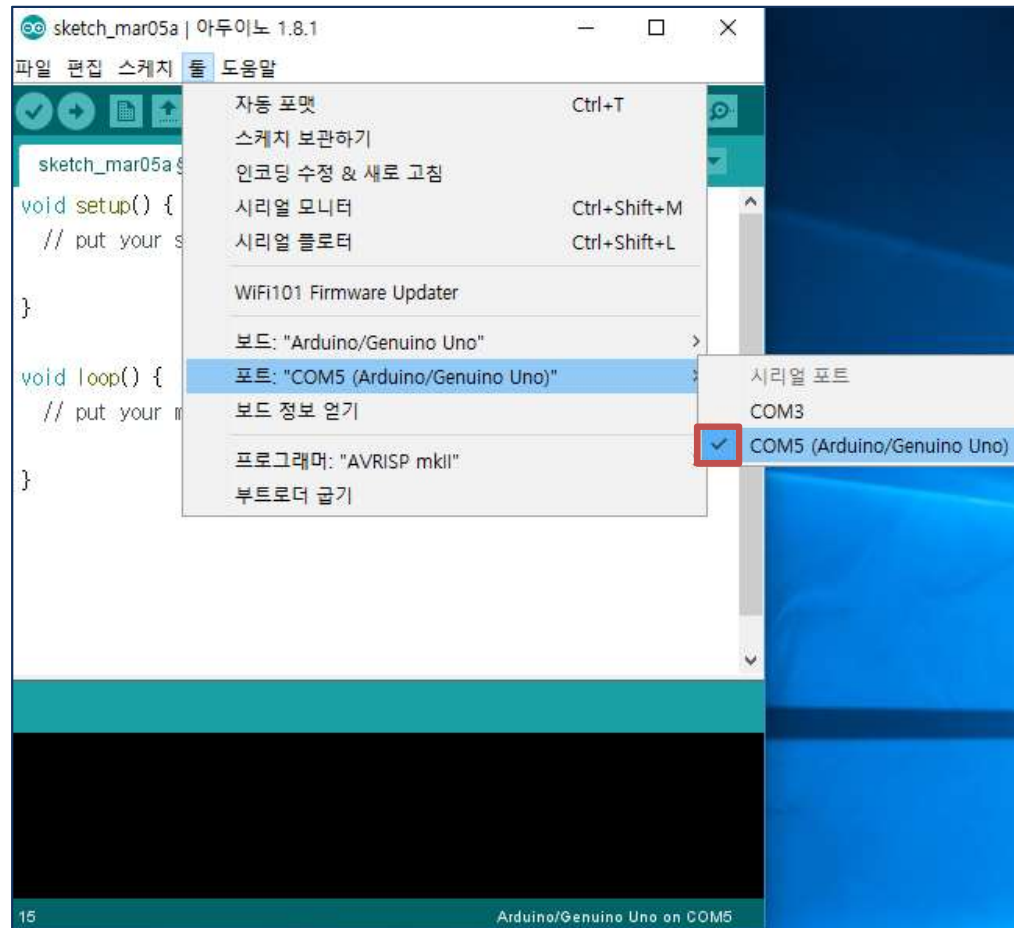
Arduino Uno



Arduino Mega 2560

아두이노 시작하기

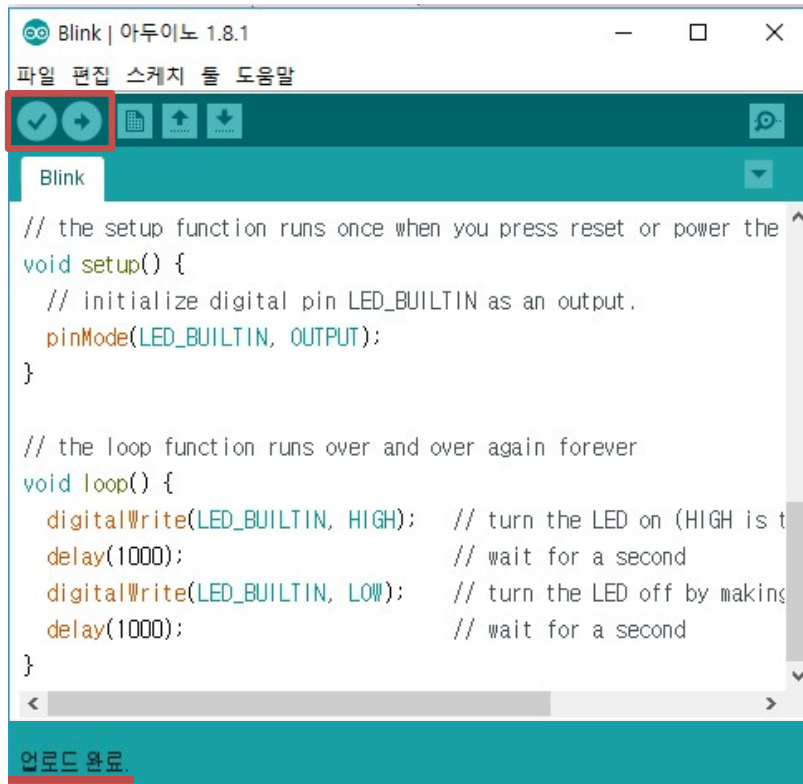
6. 시리얼 포트 선택



아두이노 시작하기

7. 프로그램 업로드

○ 파일->예제->01.Basics->Blink



확인

컴파일 하며 에러 체크



업로드

컴파일 후 아두이노 보드에 업로드

참고 : <http://arduino.cc/en/Guide/Environment>

아두이노 시작하기

□ sketch 구조



```
// Blink | 아두이노 1.8.1
파일 편집 스케치 툴 도움말

Blink

// the setup function runs once when you press reset or power the
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

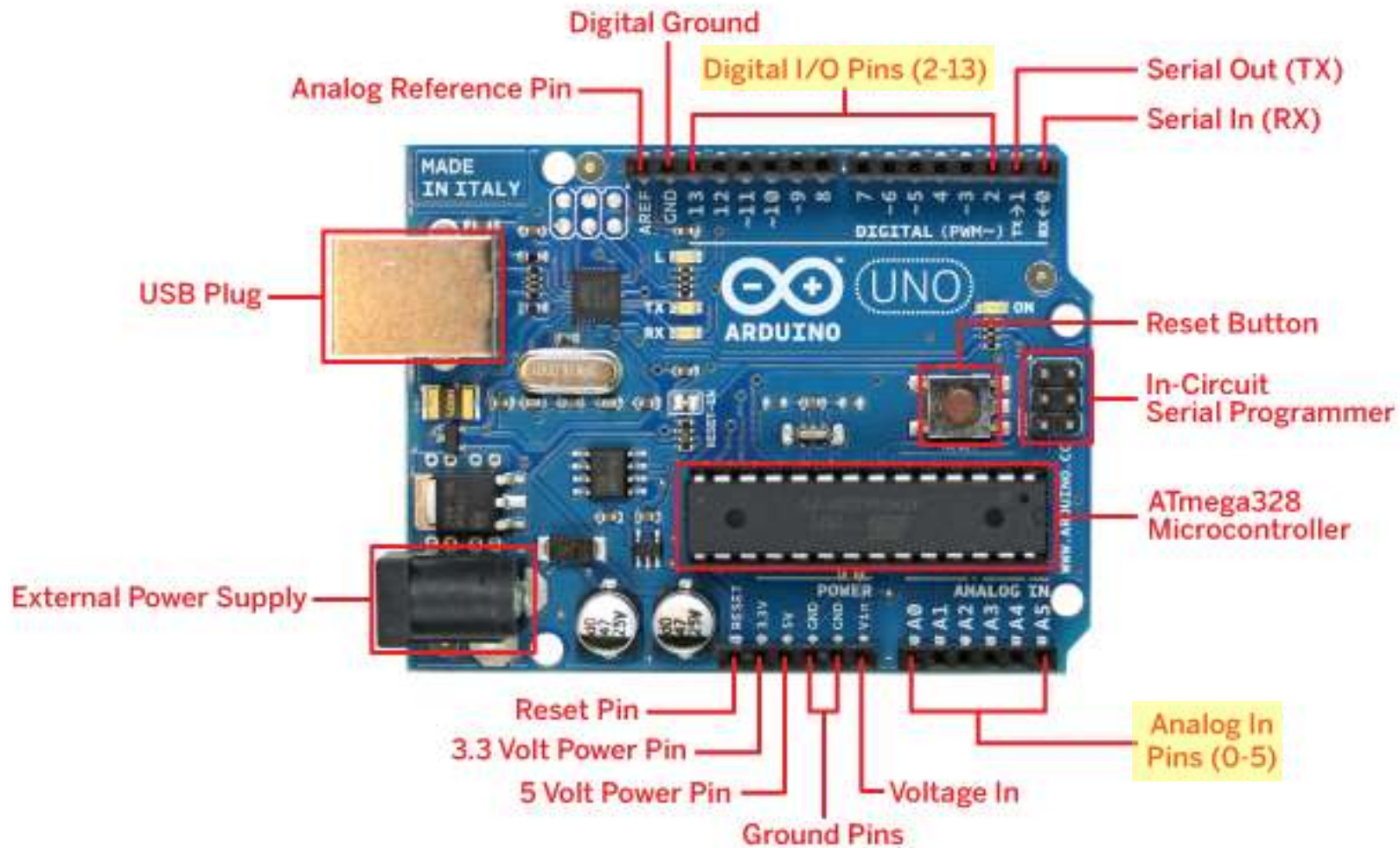
setup() : 처음 한번만 실행
→ 초기화 할 설정

loop() : 무한히 반복실행
→ 실행시킬 동작

Blink Sketch

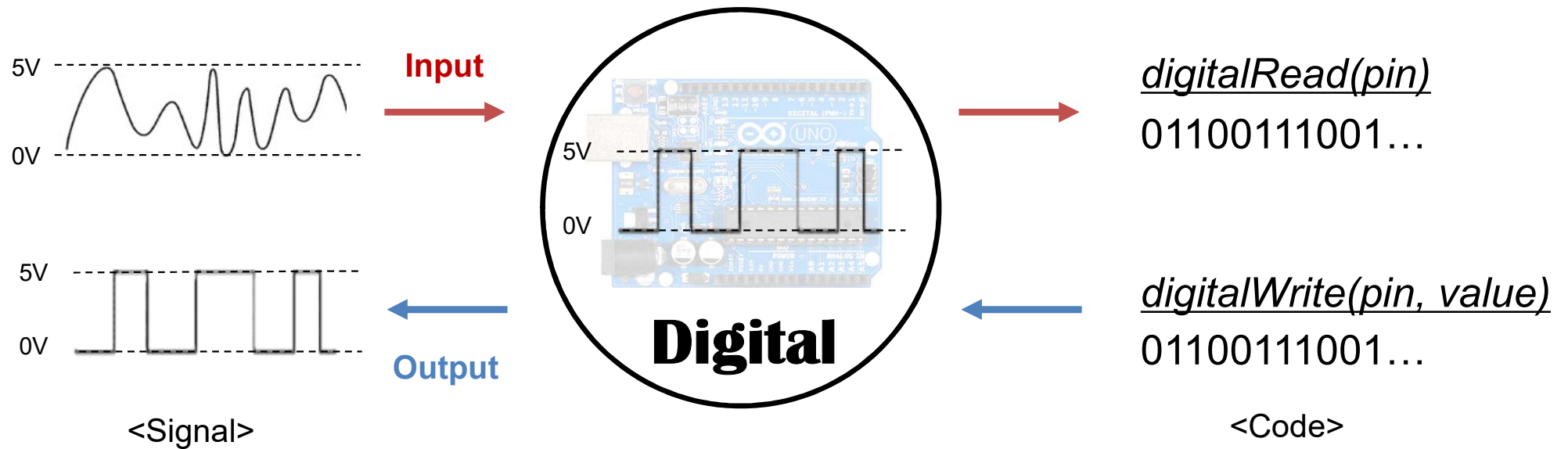
아두이노 시작하기

□ 아두이노 pinmap



Digital I/O

□ Digital I/O



- Digital pin
0번~13번

□ Digital I/O 관련 함수

pinMode(*pin*, *mode*)

pin : 할당한 핀 번호

mode : 할당한 핀에 대해 INPUT/OUTPUT 결정

pinMode는 setup함수에서 설정한다.

digitalRead(*pin*)

pin : 할당한 핀 번호(Digital pin들 중에서 설정)

pin의 값이 HIGH인지 LOW인지를 읽어 반환한다.

digitalWrite(*pin*, *value*)

pin : 할당한 핀 번호(Digital pin들 중에서 설정)

value : 할당한 핀에 대해 HIGH 또는 LOW값을 출력

□ 시간관련 함수

delay(*ms*)

입력한 시간[ms]동안 멈춘다.

millis()

아두이노가 현재 프로그램을 실행한 이후로부터의 시간[ms]을 반환한다.

delayMicroseconds(*us*)

입력한 시간[us]동안 멈춘다.

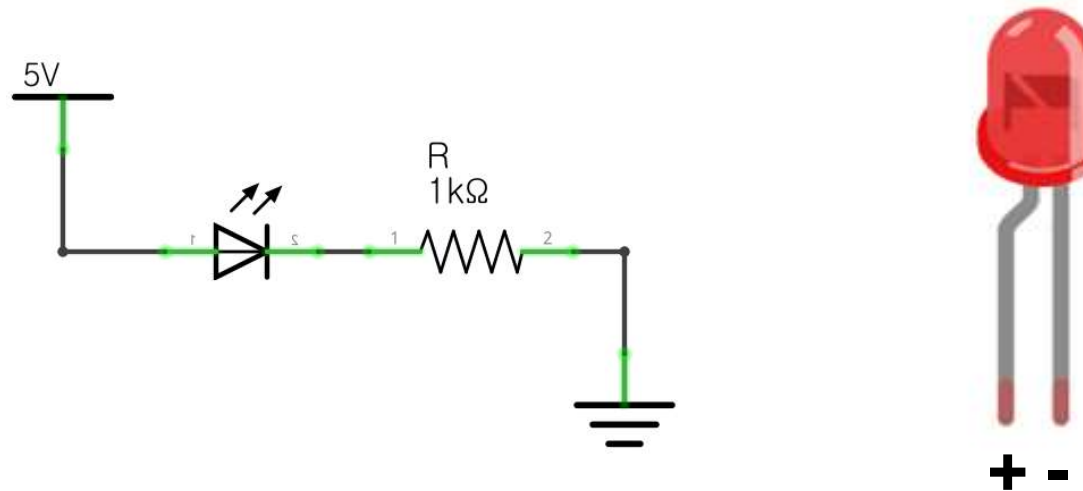
micros()

아두이노가 현재 프로그램을 실행한 이후로부터의 시간[us]을 반환한다.

[예제1] LED 켜기

□ LED

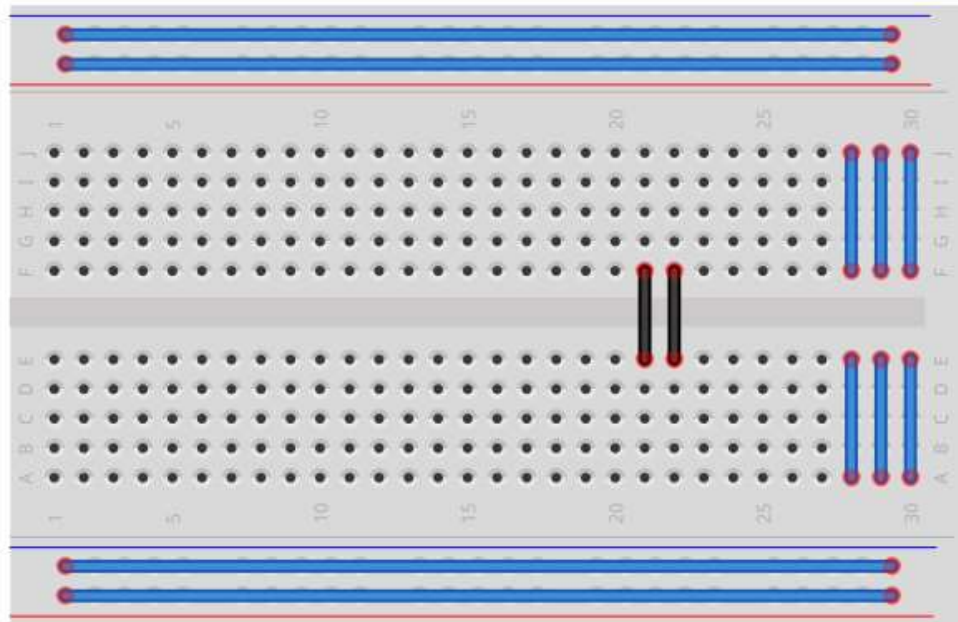
- 긴 쪽이 (+), 짧은 쪽이 (-)에 연결한다.
- (+)를 Anode, (-)를 Cathode라고도 부른다.
- 저항을 연결하여 LED의 양단에 걸리는 전압을 낮춘다.



아두이노의 13번 핀에는 1K의 내부저항이 있어 (+)와 (-)에 직접 연결하는 것이 가능하다.

[예제 1] LED 켜기

□ 브레드 보드

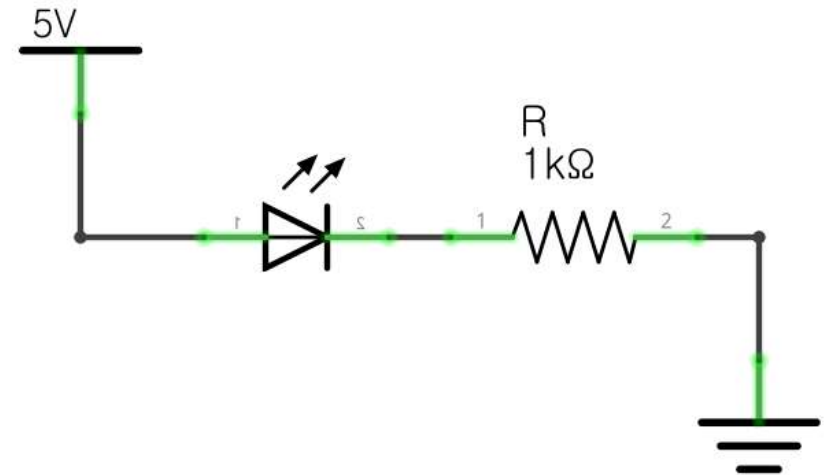
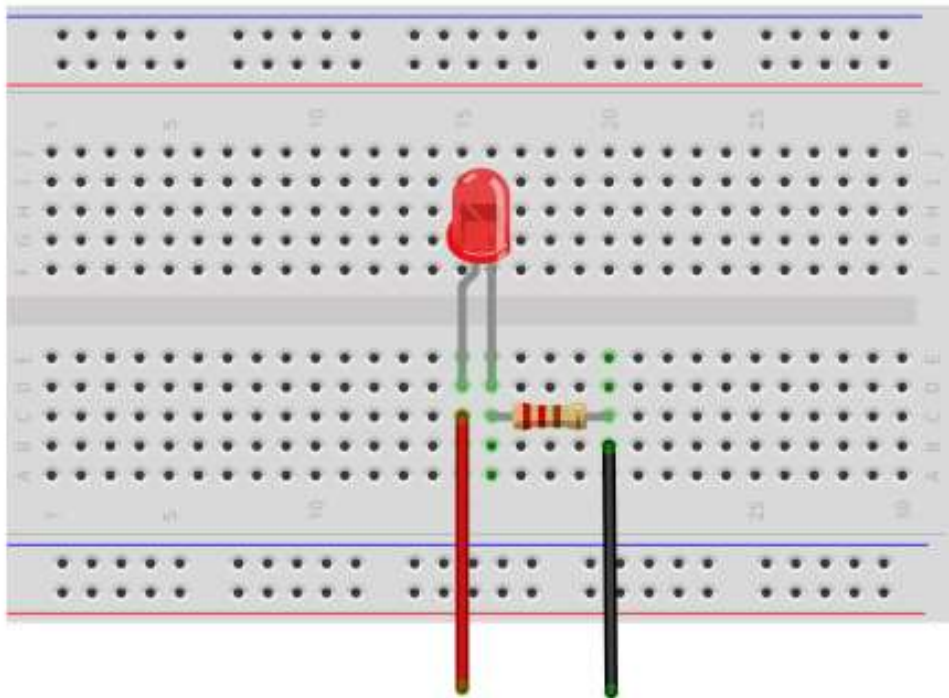


브레드 보드의 각 구멍들은 서로 연결되어 있다.

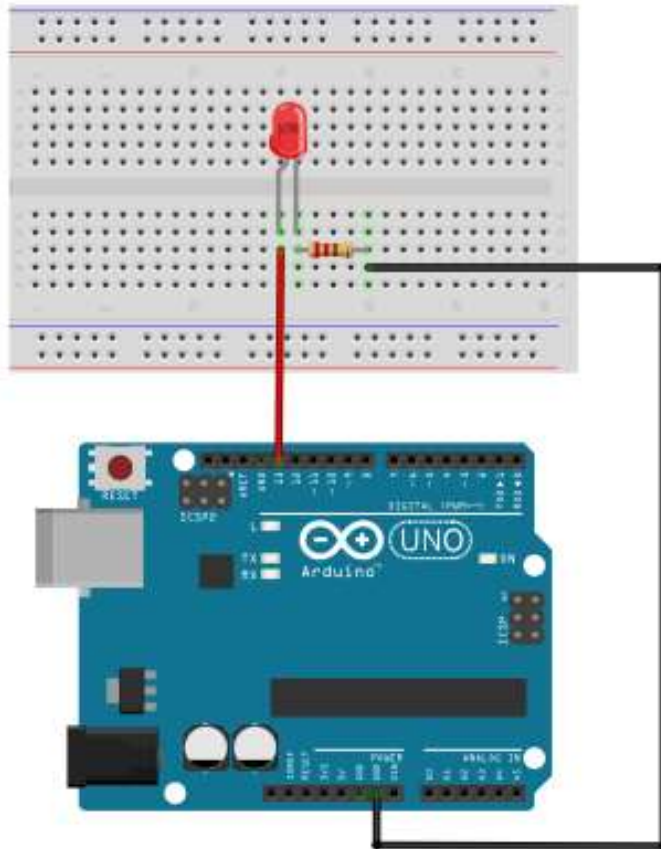
— 연결
— 연결x

[예제 1] LED 켜기

□ LED



[예제 1] LED 켜기



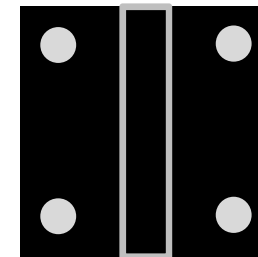
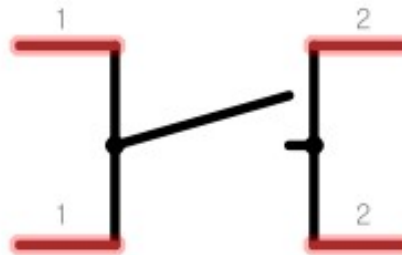
```
const int ledPin = 13;

void setup(){
    pinMode(ledPin, OUTPUT);
}

void loop(){
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}
```


[예제2] 스위치로 LED 켜기(1)

□ Tact 스위치

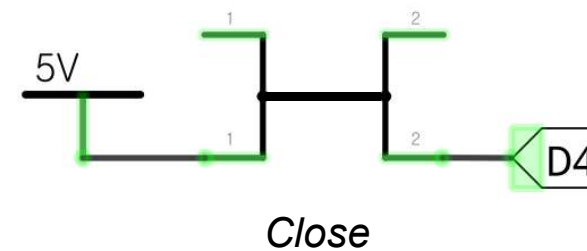
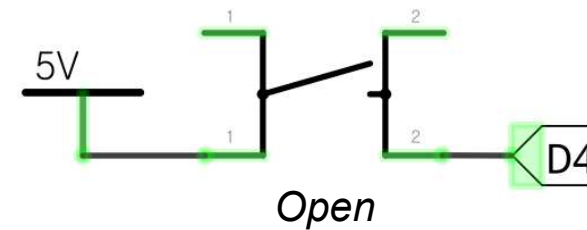
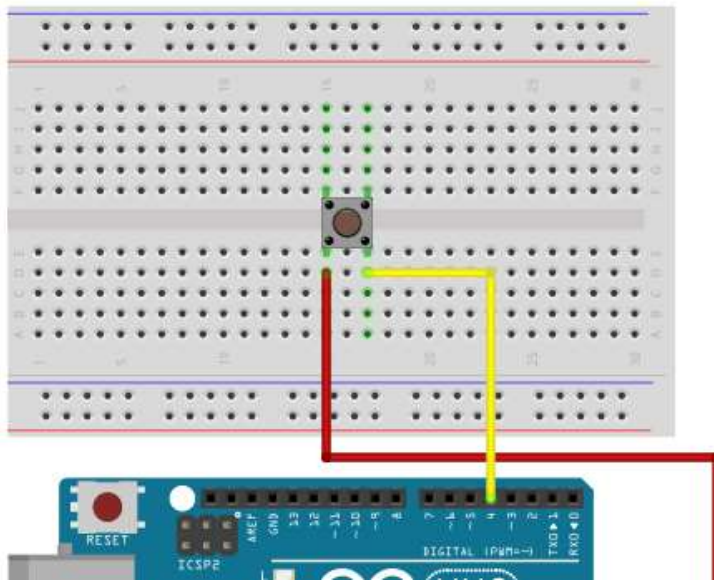


(NO, Normal Open)

평소에는 중간의 bar를 기준으로 단절되어 있다가 버튼을 누르면 연결된다.

[예제2] 스위치로 LED 켜기(1)

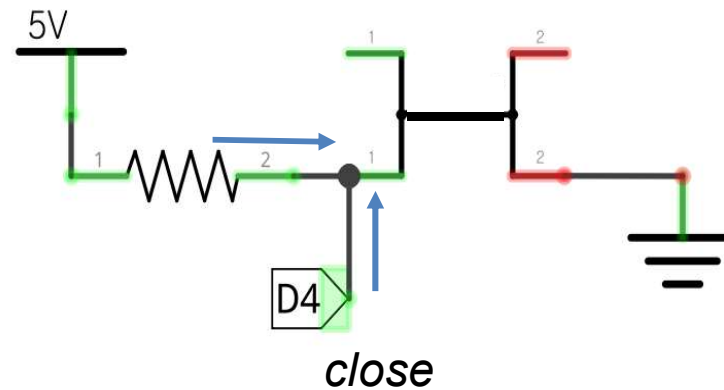
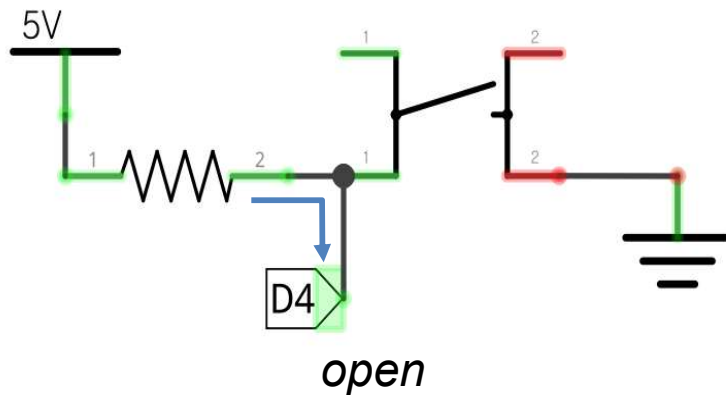
□ Tact 스위치



- 스위치가 열리면 연결된 전압이 입력으로 들어간다.
- 하지만, 스위치가 열리면 입력이 5V인지 0인지 알 수 없다.
- 이러한 상태를 **플로팅**이라고 한다.

[예제2] 스위치로 LED 켜기(1)

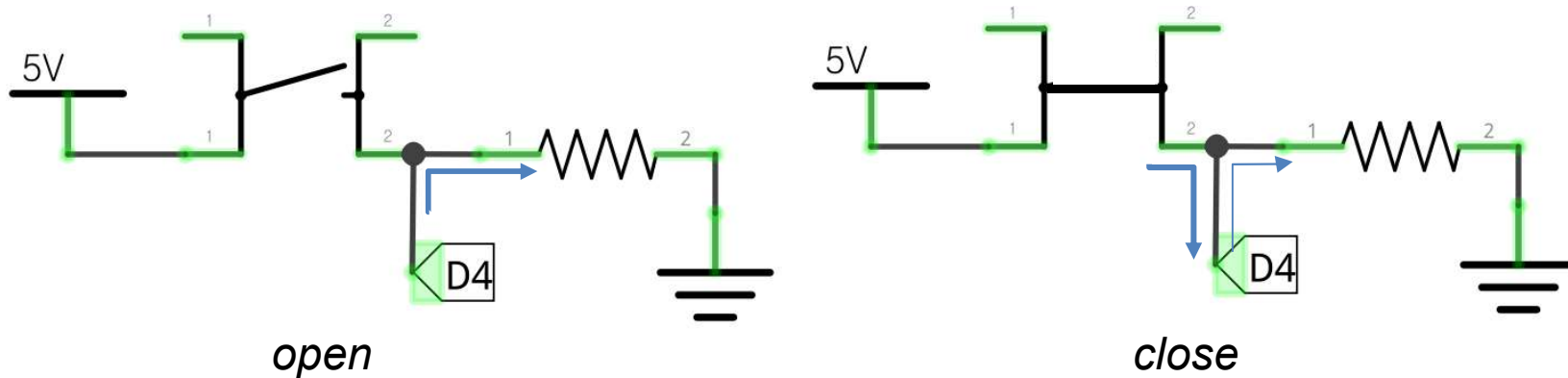
□ 풀업 저항



- 풀업(PULL-UP)은 높은 전압에 저항을 매달아 둔다는 뜻이다.
- 스위치가 열려있을 때는 5V의 전압이 가해져서 HIGH로 인식한다.
- 스위치가 닫히게 되면 GND쪽으로 전류가 흐르게 되어 LOW로 인식한다.

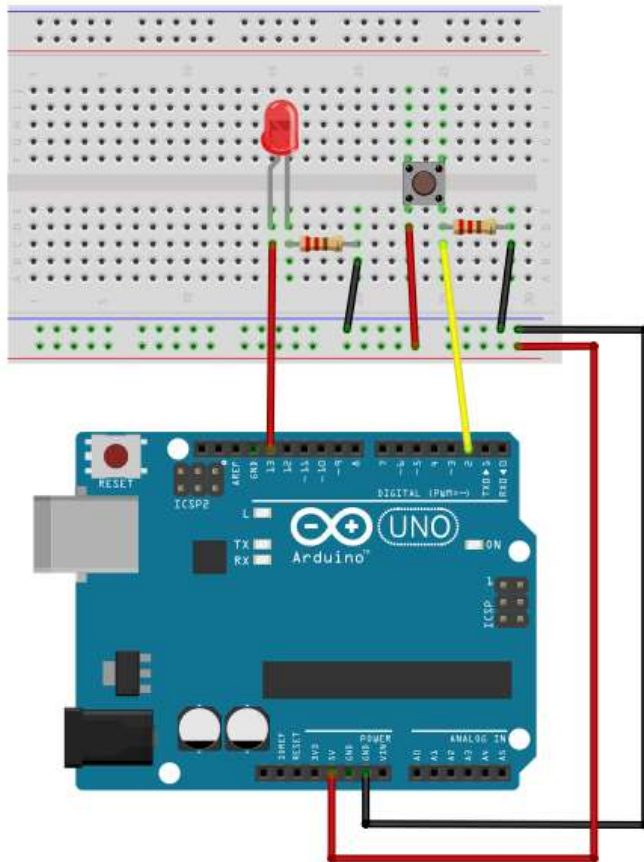
[예제2] 스위치로 LED 켜기(1)

□ 풀다운 저항



- 풀다운(PULL-DOWN)은 GND에 저항을 매달아 둔다는 뜻이다.
- 스위치가 열려있을 때는 GND와 연결되어 있으므로 LOW로 인식한다.
- 스위치가 닫히게 되면 소량의 전류가 GND쪽으로 흘러 나가지만 많은 양의 전류가 4번 핀으로 흐르게 되어 HIGH로 인식한다.

[예제2] 스위치로 LED 켜기(1)



```
const int ledPin = 13;
const int buttonPin = 2;

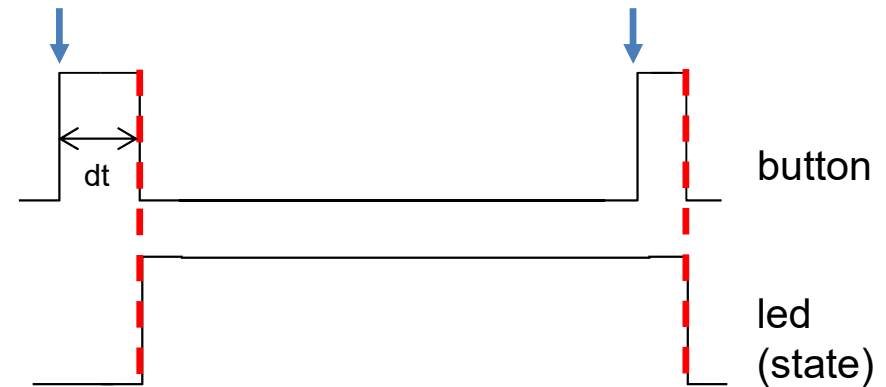
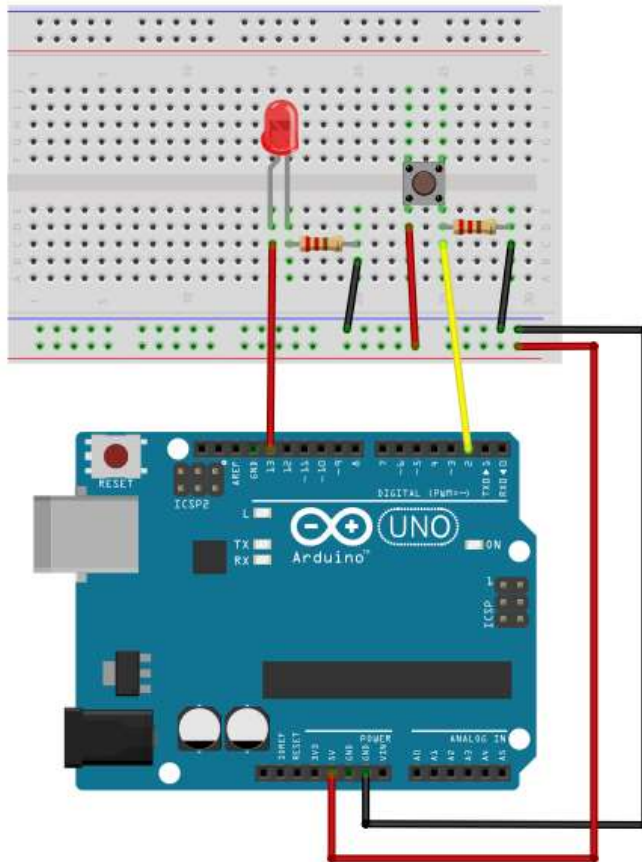
int buttonState = 0;

void setup(){
    pinMode(ledPin, OUTPUT);
    pinMode(buttonPin, INPUT);
}

void loop(){
    buttonState = digitalRead(buttonPin);

    if(buttonState == HIGH){
        digitalWrite(ledPin, HIGH);
    }
    else{
        digitalWrite(ledPin, LOW);
    }
}
```

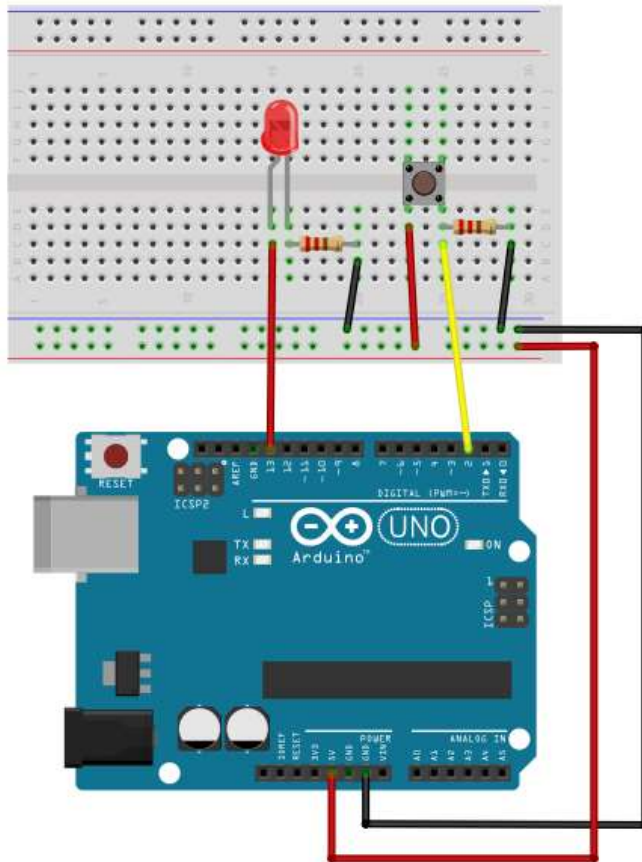
[예제3] 스위치로 LED 켜기(2)



button의 값

- 1) LOW \rightarrow HIGH : dt 의 기준 시각 측정
- 2) HIGH \rightarrow LOW : dt 시간이 충분하다면 state를 변화시킴

[예제3] 스위치로 LED 켜기(2)



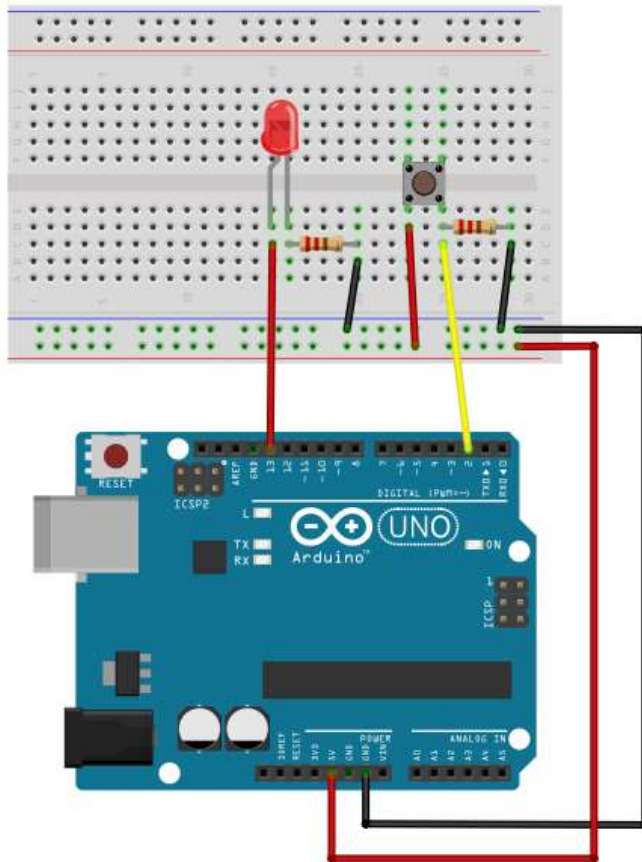
```
const int ledPin = 13;  
const int buttonPin = 2;
```

```
int state = LOW;  
int reading;  
int previous = LOW;
```

```
long start = 0;  
long dt = 0;  
long debounce = 10;
```

```
void setup(){  
  pinMode(ledPin, OUTPUT);  
  pinMode(buttonPin, INPUT);  
}
```

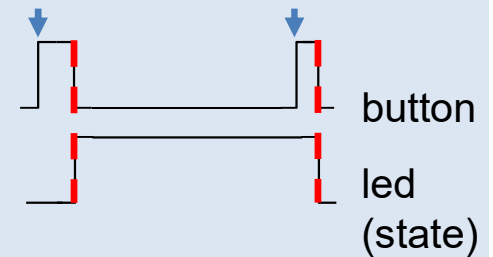
[예제3] 스위치로 LED 켜기(2)



```
void loop(){
  reading = digitalRead(buttonPin);

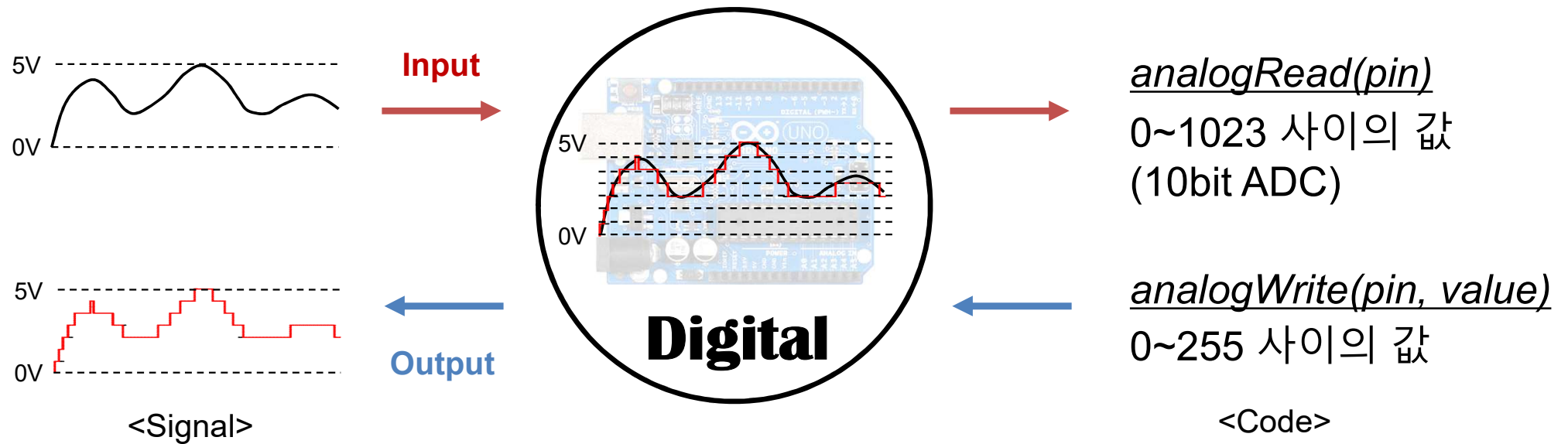
  if(previous == LOW && reading == HIGH){
    start = millis();
  }
  else if(previous == HIGH && reading == LOW){
    dt = millis() - start;
    if(dt>debounce){
      if(state == HIGH)      state = LOW;
      else                  state = HIGH;
    }
  }

  digitalWrite(ledPin, state);
  previous = reading;
}
```



Analog I/O

□ Analog I/O



- PWM pin
3, 5, 6, 9, 10, 11번



- AnalogIn pin
A0 ~ A5번

□ Analog I/O 관련 함수

pinMode(*pin*, *mode*)

pin : 할당한 핀 번호

mode : 할당한 핀에 대해 INPUT/OUTPUT 결정

pinMode는 setup함수에서 설정한다.

analogRead(*pin*)

pin : 할당한 핀 번호(AnalogIn pin들 중에서 설정)

pin의 값을 읽어 0~1023사이의 값으로 반환한다.

analogWrite(*pin*, *value*)

pin : 할당한 핀 번호(PWM pin들 중에서 설정)

value : 할당한 핀에 대해 0~255사이의 값을 출력

Analog I/O

[예제4] 가변저항 값 출력

Potential0 | 아두이노 1.8.1

파일 편집 스케치 툴 도움말

Potential0

```
const int potPin = A0;
int readValue = 0;

void setup() {
  Serial.begin(9600);
  pinMode(potPin, INPUT);
}

void loop() {
  readValue = analogRead(potPin);
  Serial.print("AnalogRead:");
  Serial.println(readValue);
}
```

저장 완료

15 Arduino/Genuino Uno on COM5

시리얼 모니터



COM5 (Arduino/Genuino Uno)

보낼 데이터

받은 데이터

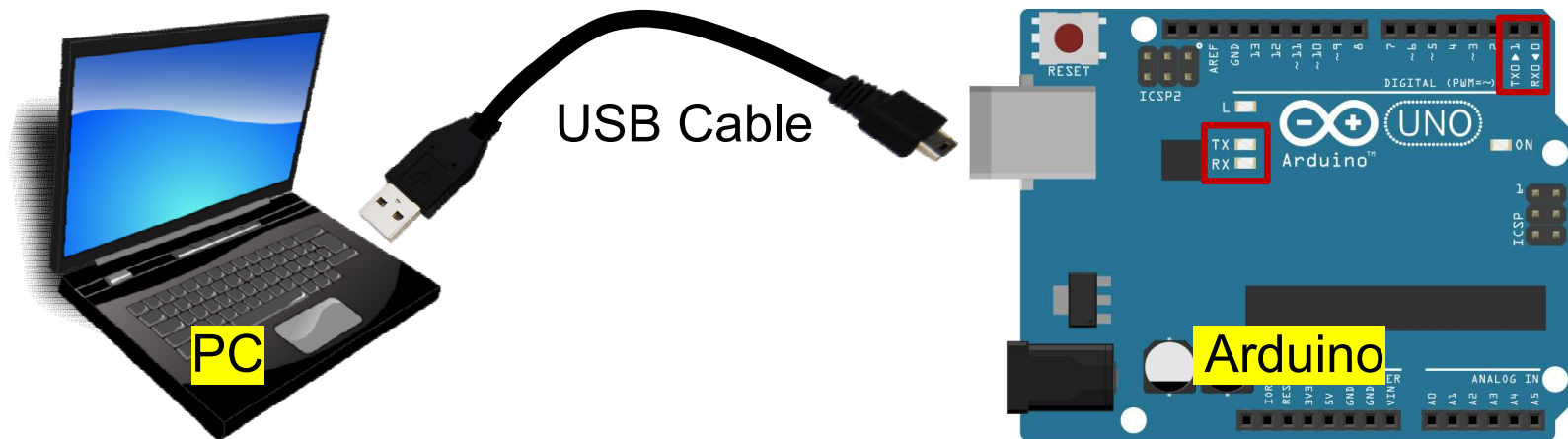
```
AnalogRead: 34
AnalogRead: 54
AnalogRead: 94
AnalogRead: 132
AnalogRead: 164
AnalogRead: 216
AnalogRead: 263
AnalogRead: 290
AnalogRead: 320
AnalogRead: 344
AnalogRead: 371
AnalogRead: 400
AnalogRead: 435
AnalogRead: 469
```

☐ 자동 스크롤 line ending 없음 9600 보드레이트

[예제4] 가변저항 값 출력

□ 시리얼 통신(Serial Communication)

- 송신(TX), 수신(RX)의 2개의 데이터 선을 이용한 PC와 Arduino간의 통신
- 1byte(=8bit)의 데이터를 비트 단위로 나누어 차례대로 전송하는 통신방식
- 시리얼 통신을 하기 위해서는 속도(1초당 전송되는 비트 수, baud rate)의 설정이 필요
- 아두이노 IDE에서는 시리얼 통신을 모니터 할 수 있는 툴을 제공



□ 시리얼 통신 관련 함수

Serial.begin(*baud*)

시리얼 통신을 사용하기 위해서는 **baud rate**를 setup함수에서 설정해준다.

Serial.available()

읽기가 가능한 바이트 수를 반환한다.

Serial.read()

시리얼 창에 입력한 값을 읽는다.

Serial.print()

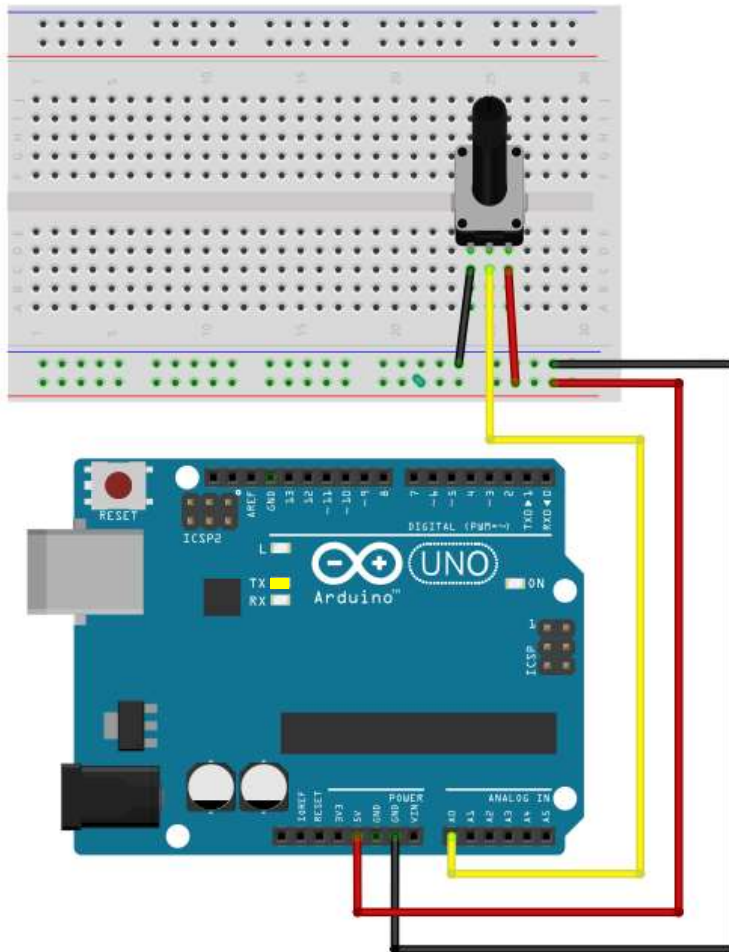
()안의 값을 시리얼 모니터에 출력한다.

Serial.println()

()안의 값을 시리얼 모니터에 출력한 후 개행 한다.

Analog I/O

[예제4] 가변저항 값 출력

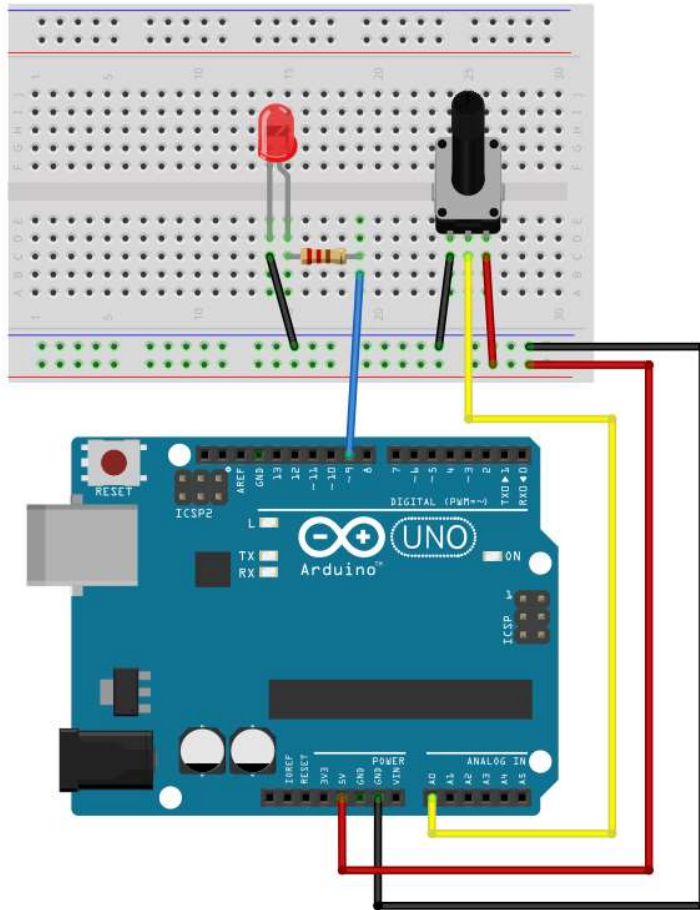


```
const int potPin = A0;
int readValue = 0;

void setup() {
  Serial.begin(9600);
  pinMode(potPin, INPUT);
}

void loop() {
  readValue = analogRead(potPin);
  Serial.print("AnalogRead:");
  Serial.println(readValue);
  delay(500);
}
```

[예제5] 가변저항을 이용하여 LED 밝기 조절



```
const int ledPin = 9;  
const int potPin = A0;  
int readValue = 0;  
int writeValue = 0;
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(potPin, INPUT);  
  pinMode(ledPin, OUTPUT);  
}
```

```
void loop() {  
  readValue = analogRead(potPin);  
  writeValue = (255.0/1023.0)*readValue;  
  analogWrite(ledPin, writeValue);  
  Serial.print("AnalogRead:");  
  Serial.print(readValue);  
  Serial.print(", AnalogWrite:");  
  Serial.println(writeValue);  
}
```

0~255

0~1023

[예제5] 가변저항을 이용하여 LED 밝기 조절

□ 소수점 연산

- (int) \odot (int) = (int)
- (int) \odot (float) = (float), (float) \odot (int) = (float)
- (float) \odot (float) = (float)

```
int val;  
Serial.print("#n127/1023=");  
Serial.print(127/1023);  
val = 127/1023;  
Serial.print(" -(int)-> ");  
Serial.println(val);  
  
Serial.print("#n127/1023.0=");  
Serial.print(127/1023.0);  
val = 127/1023.0;  
Serial.print(" -(int)-> ");  
Serial.println(val);  
  
Serial.print("#n127/1023.0*255=");  
Serial.print(127/1023.0*255);  
val = 127/1023.0*255;  
Serial.print(" -(int)-> ");  
Serial.println(val);
```



COM5 (Arduino/Genuino Mega or Mega 2560)

127/1023=0 -(int)-> 0

127/1023.0=0.12 -(int)-> 0

127/1023.0*255=31.66 -(int)-> 31

자동 스크롤 line ending 없음 9600 보드레이트

[예제5] 가변저항을 이용하여 LED 밝기 조절

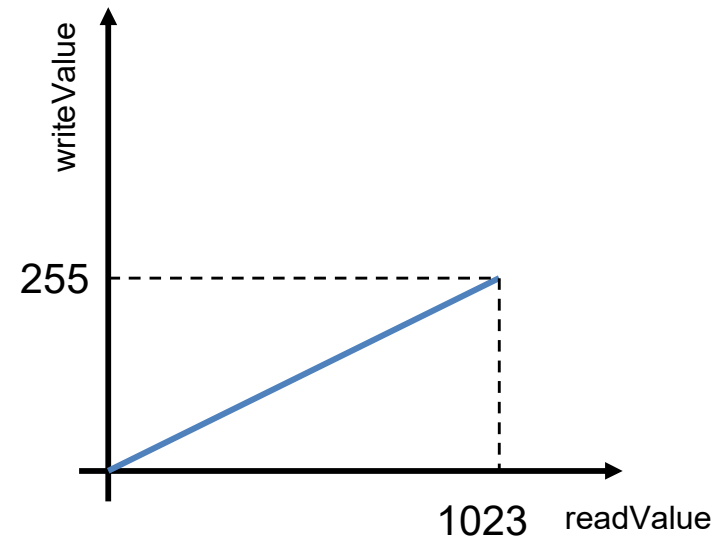
map(value, fromLow, fromHigh, toLow, toHIGH)

fromLow ~ fromHigh의 값을 가지는 value를 toLow ~ toHigh의 값으로 바꾸어 반환한다.

```
writeValue = (255.0/1023.0)*readValue;
```

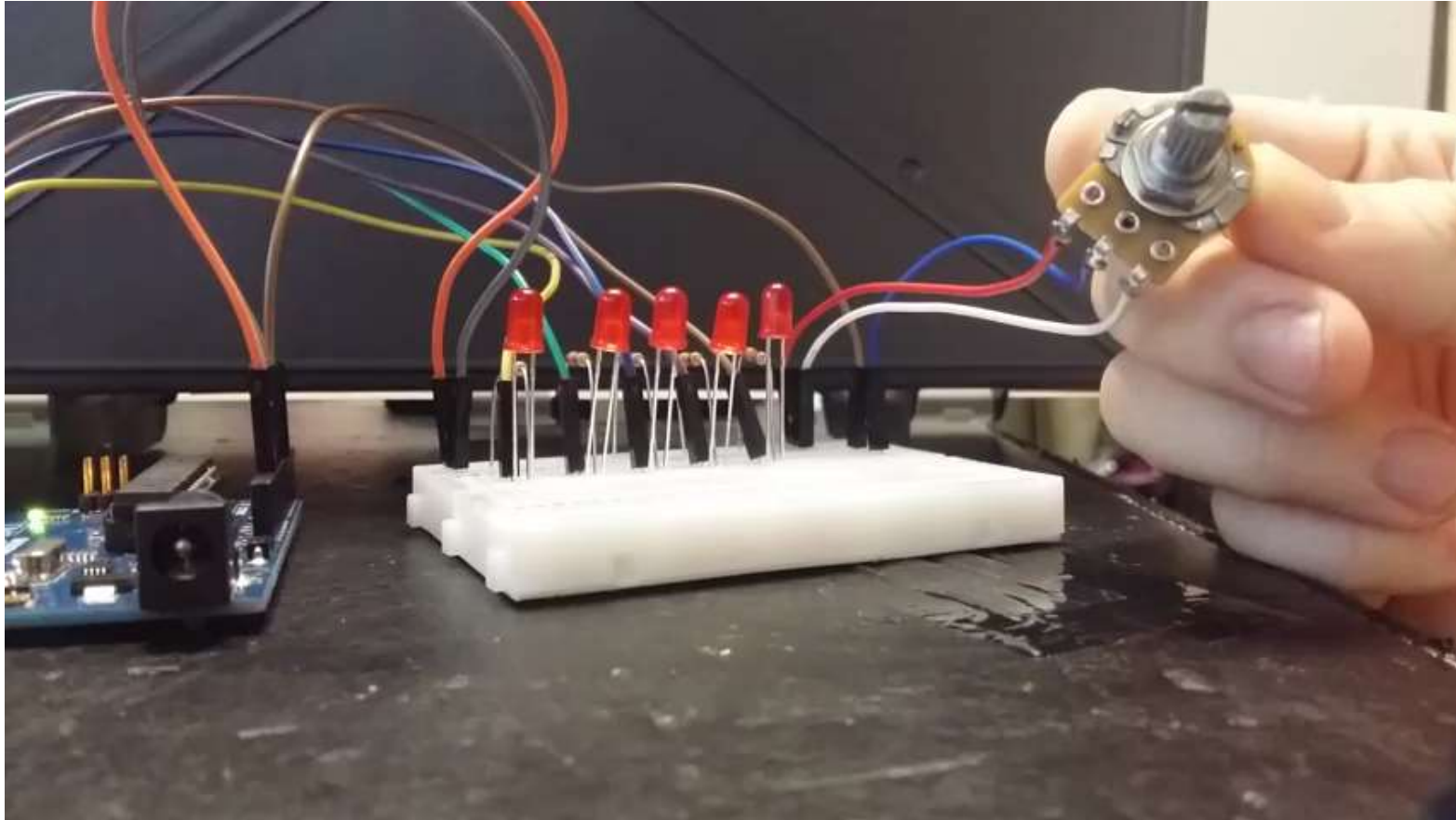


```
writeValue = map(readValue, 0, 1023, 0, 255);
```



도전과제

□ 도전과제1



□ 도전과제2

