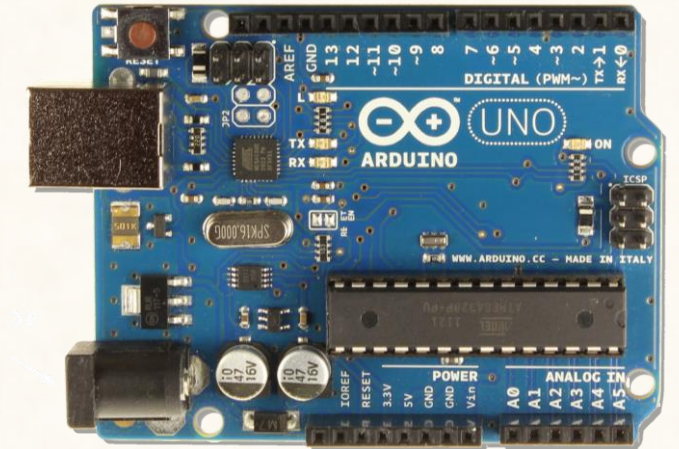


이것만 알면 나도 아두이노 할 줄 아는 척을?

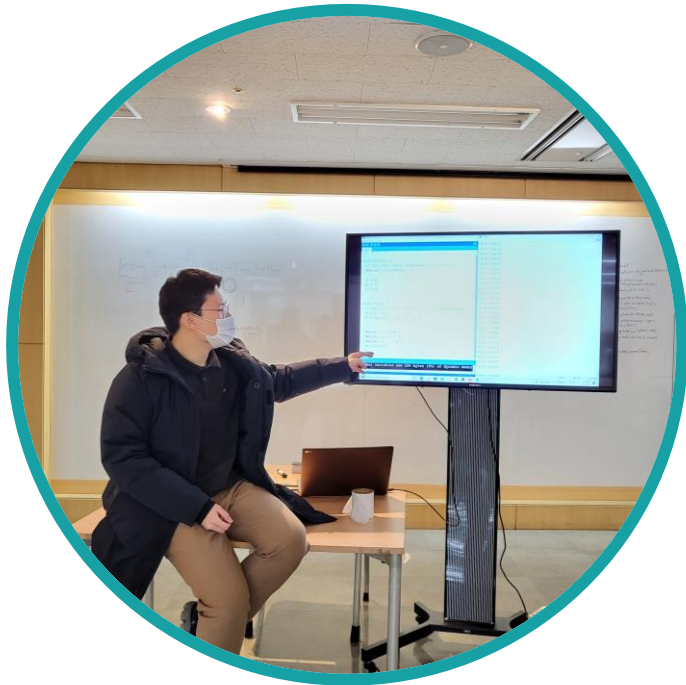
실전압축디자인용 아두이노사용법



이것만 알면 나도 아두이노 할 줄 아는 척을?

실전압축디자인용 아두이노사용법 2.0

튜터소개



■ 곽동준 *dongjoonkwak.com*

2018.09~2018.12 인터랙션 디자인 수강 (Red S Award)

2021.04~2021.05 생활디자인학과 신입생 튜터링 프로그램 아두이노 튜터

2021.12~2022.01 겨울 더작당 아두이노 스터디 튜터

2022.03~2022.11 국립어린이과학관 특별전시 “알쏭달쏭 머릿속” 개발참여

2022.11 JAKDANG X DESIGN STUDIO i 아두이노 특강

dongjoonkwak@gmail.com

010-8685-6168

자료 다운로드

dongjoonkwak.com/arduino

Contents

1. 기본개념 학습하기

- 아두이노 개요
- 아두이노 뜯어보기
- 입력과 출력
- 기초 코드의 구성

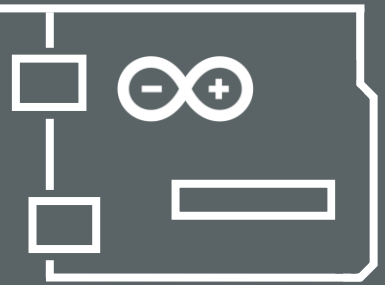
2. 주요장치 실습하기

- LED 사용해보기
- 버튼 사용해보기
- 가변저항 사용해보기
- 프로젝트 만들어보기

3. 추가 학습

- 서보모터 사용해보기
- 버튼 만들어보기
- delay/millis/count

기본개념 학습하기



아두이노란?

작은 컴퓨터이다.

왜 아두이노인가?

- 싸다.
- 조작하기 간단하다.
- 변형하기 쉽다.
- 커뮤니티가 크다.

아두이노로 할 수 있는 것

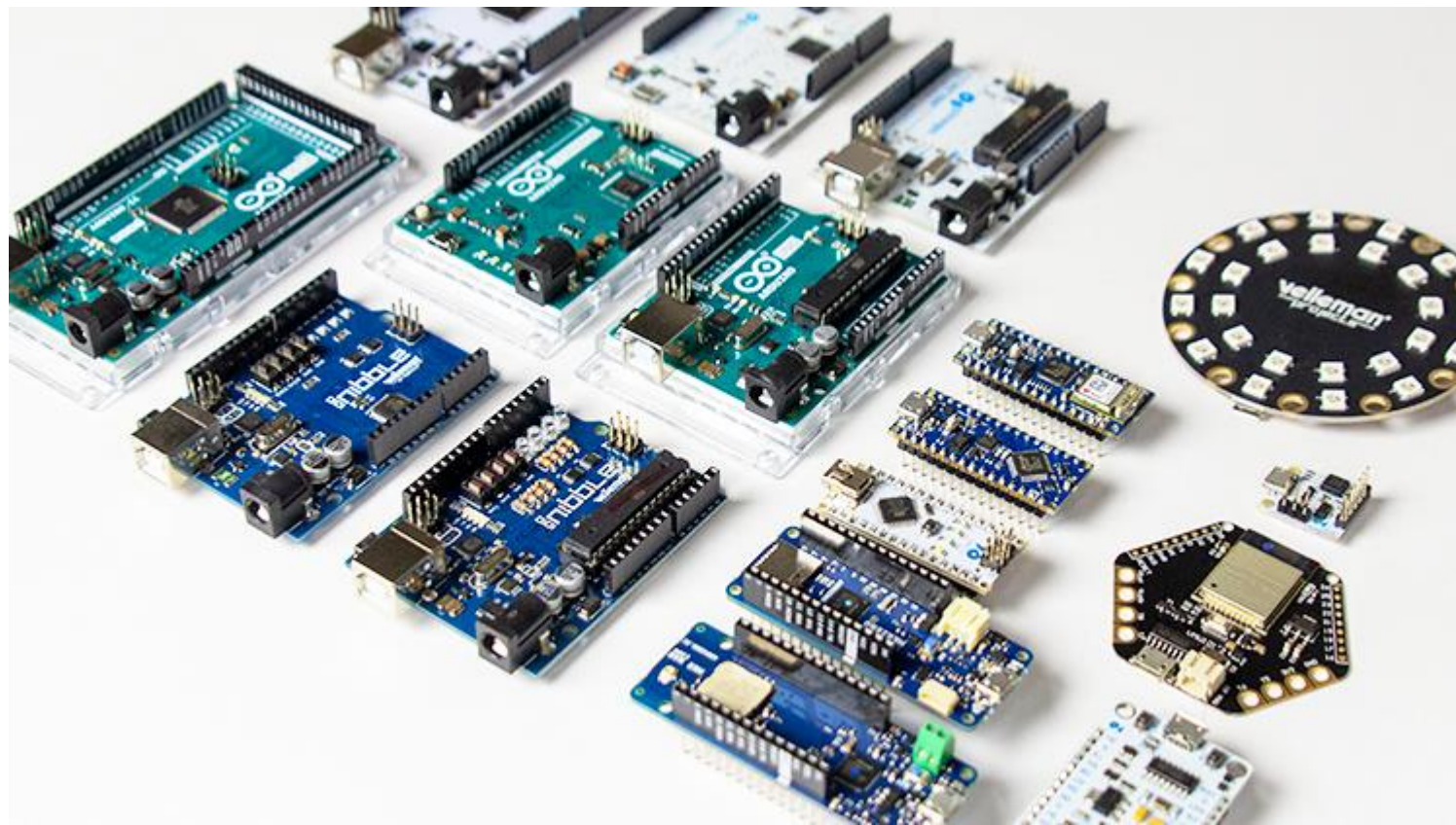
아두이노가 노트북이면 모터와 센서는 포토샵과 같은 프로그램이다.

모든 장치를 다 아는 것은 불가능하다. 중요한 것 몇가지와 구글링 하는 법을 익혀 두자.

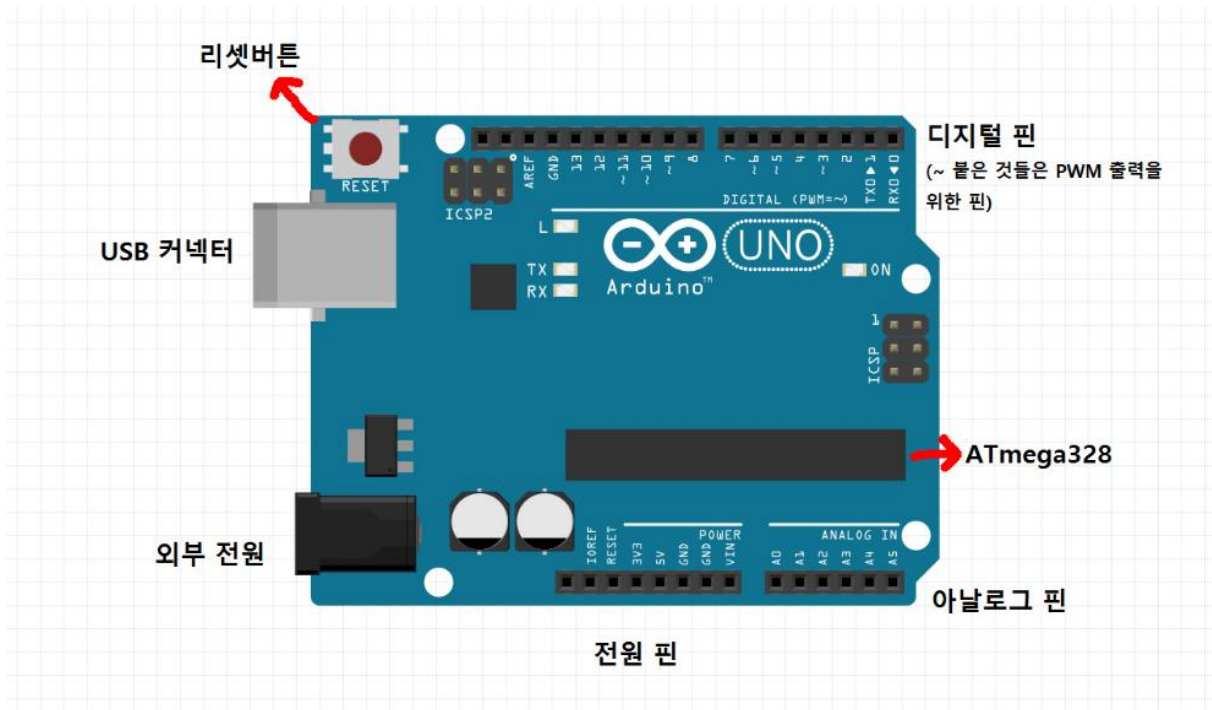
기본 입력 장치: 버튼, 가변저항, 조도센서...

기본 출력 장치: led, led 스트립, 모터, 스테퍼 모터...

아두이노의 종류



아두이노 뜯어보기(Arduino Uno)



USB 커넥터: 컴퓨터와 연결할 수 있는 포트이다.

외부 전원: 외부에서 전원을 받을 수 있는 포트이다.

전원 핀: 전원을 받고, 낼 수 있는 핀들이다.

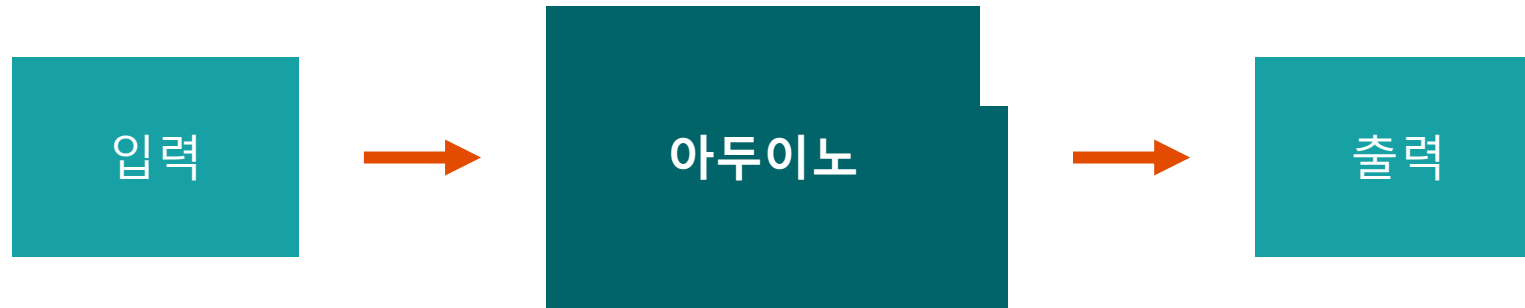
아날로그 핀: 아날로그 전기 입력을 받는 곳이다.

디지털 핀: 디지털 입력, 그리고 출력을 하는 곳이다.

리셋버튼: 아두이노를 잠시 껐다 켜다.

ATmega328: 아두이노의 실질적 뇌

입력과 출력



스위치



스위치:

전기의 흐름을 조절할 수 있는 도구이다. 푸시버튼, 토글, 로커 등 여러 종류가 있다.

가변저항



가변저항(potentiometer/variable resistor):
사용자가 저항 값을 조절 할 수 있는 저항이다.

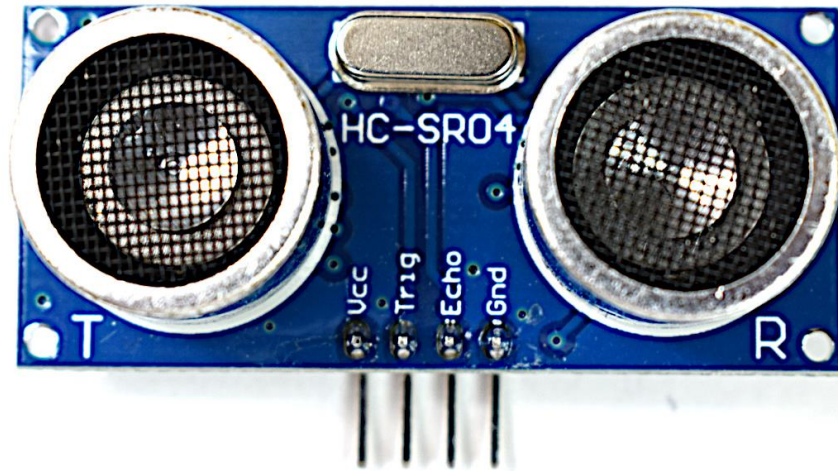
조도 센서



CDS 포토셀 센서:

황화카드뮴을 활용한 양방향 센서로, 들어오는 빛의 양에 따라 저항 값이 변한다.

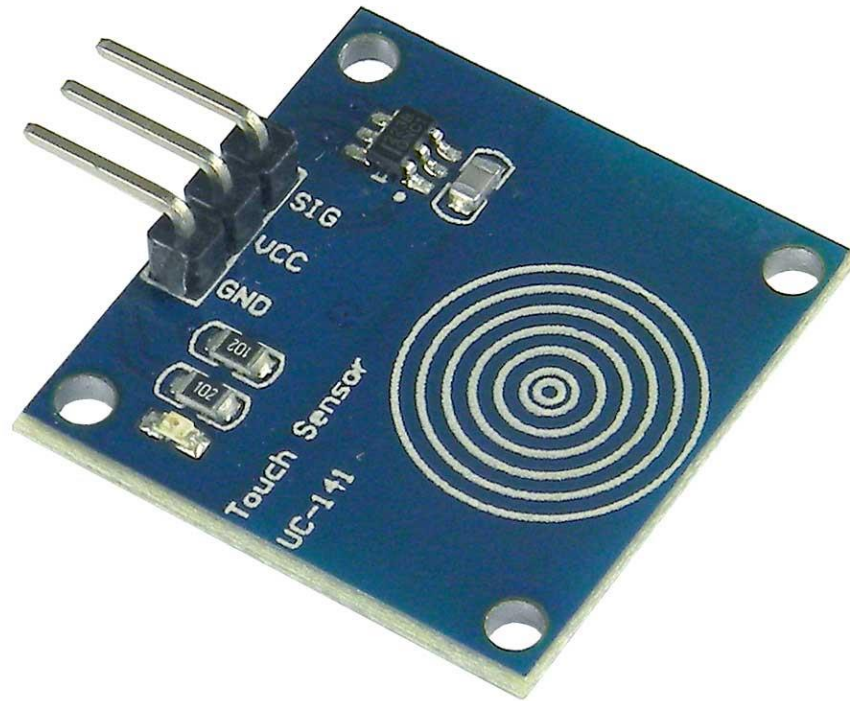
초음파 센서



초음파 거리 센서:

초음파를 발신하고 수신할 수 있다. 코딩을 통해 앞의 물체와의 거리를 구할 수 있다.

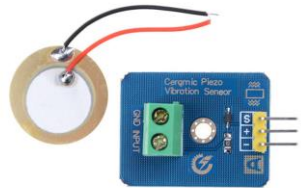
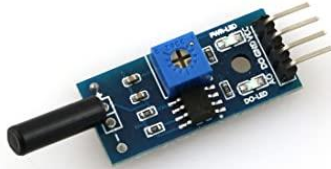
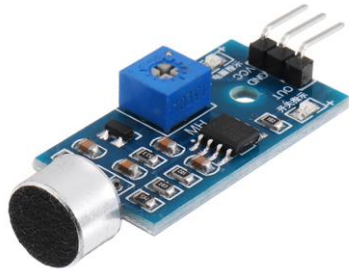
터치 센서



정전식 터치 센서(capacitive touch sensor):

축전기의 원리를 활용한 센서로, 전기 신호를 보내고 읽어 접지된 가까운 전도체와의 거리를 측정한다.

다른 센서들



PIR 센서: 움직임을 감지한다.

소리 센서: 소리를 측정한다.

틸트 센서: 센서가 흔들릴 경우, 스위치가 켜진다.

가속도 센서: 중력을 통해 센서의 방향을 감지한다.

피에조: 진동을 전기신호로 변환한다.

온습도 센서: 온도, 습도를 측정한다.

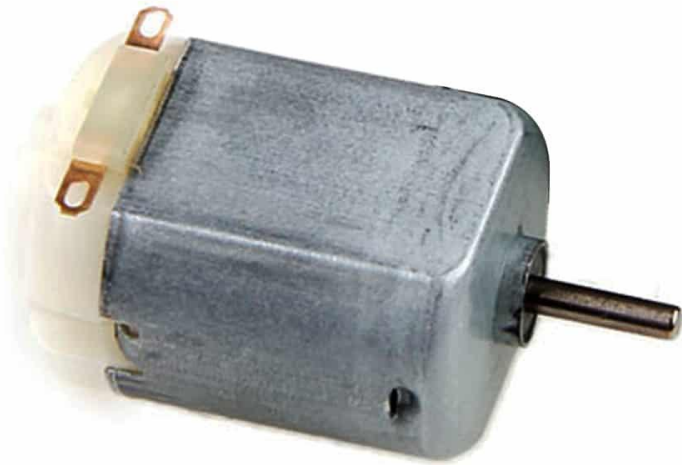
LED



발광 다이오드(Light Emitting Diode):

전기가 한방향으로 흐르게 하는 반도체로, 올바르게 전기가 통했을 때 빛이 난다.

DC모터



DC모터:
직류를 보냈을 때 축이 회전하는 장치이다.

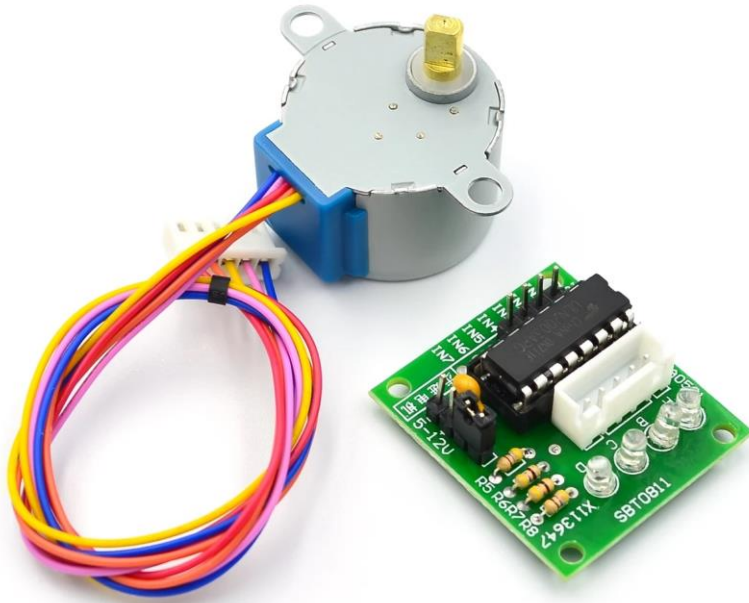
서보모터



서보모터:

기어박스과 모터가 결합된 형태로, 일정한 각도 내에서 원하는 만큼 축을 회전시킬 수 있다.

스텝퍼 모터



스텝퍼 모터:

모터 내의 전자석을 일정하게 켜고 끄면서 회전한다. 원하는 속도와 횟수만큼 축을 회전시킬 수 있다.

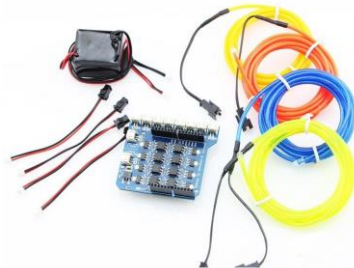
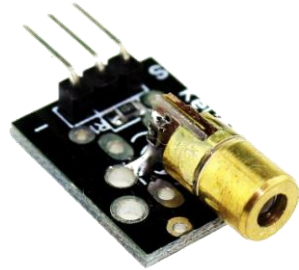
릴레이



릴레이:

아두이노 밖의 다른 전기 회로를 닫거나 열 수 있다.

다른 출력 장치들



LCD 화면: 화면을 통해 텍스트나 이미지를 나타낸다.

레이저 모듈: 레이저를 쏜다.

피에조 부저: 소리를 발생시킨다.

LED 스트립: 여러 LED를 조절 할 수 있게 한다.

LED 매트릭스: 여러 LED를 조절 할 수 있게 한다.

EL 와이어: 네온사인처럼 빛나는 끈이다.

디지털과 아날로그

*디지털, 아날로그 입력이란?

디지털 입력은 전기의 유무를 나타낸다. (ex: 버튼)

아날로그 입력은 전압 값을 통해 세기를 나타낸다. (ex: 조도센서, 압력센서)

준비하기

<https://www.arduino.cc/en/software>

(舊)아두이노 IDE 뜯어보기

컴파일 버튼

업로드 버튼

시리얼 모니터

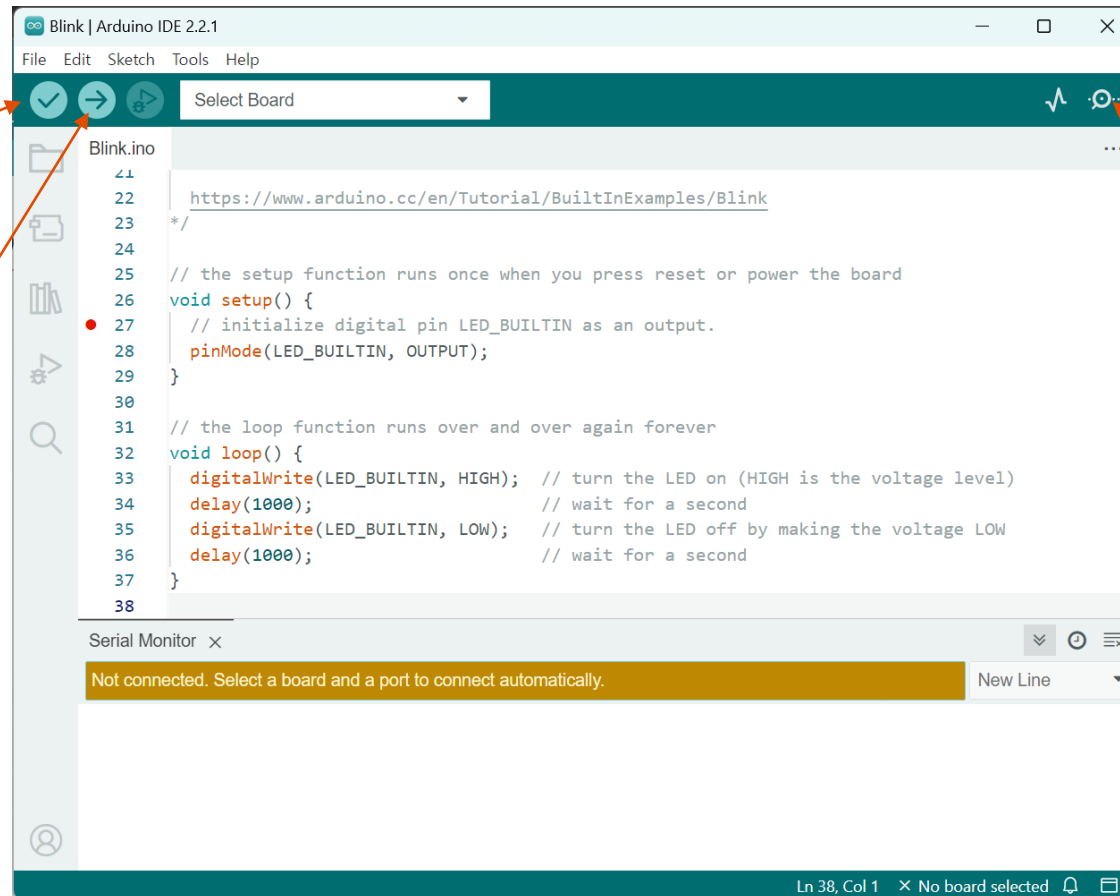


(新)아두이노 IDE 뜯어보기

컴파일 버튼

업로드 버튼

시리얼 모니터



아두이노와 코딩



```
Week4 §
/ 인풋과 아웃풋은 상수로 정의해 나중에 실수로 바뀌거나 섞이는 일 이 없도록 한다.
const int TRIG_PIN = 4;
const int ECHO_PIN = 5;

//이전상태를 저장할 변수를 만든다.
int prev = 3;

void setup()
{
  // 컴퓨터와 아두이노 간의 통신을 세팅해준다.
  Serial.begin(9600);

  // 초음파 센서를 세팅해준다.
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

void loop() {
  // 센서와 물체간의 거리를 계산한다.
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int distance = pulseIn(ECHO_PIN, HIGH) * 340 / 2 / 10000;

  // 거리 차이가 급변하면 전 값을 따라간다.
  if (distance-prev > 2) distance = prev;

  // 거리를 출력한다.
  Serial.print(distance);
  Serial.println("cm");

  // 거리를 주파수에 매핑해주고, 출력한다.
  tone(5, map(distance, 0, 20, 123.5, 261.9));

  //현재값을 이전값으로 저장한다.
  prev = distance;
}
```

40 Arduino Uno on COM5

아두이노 코드 뜯어보기

전처리

```
Week4 $  
/ 인풋과 아웃풋은 상수로 정의해 나중에 실수로 바뀌거나 섞이는 일 이 없도록 한다.  
const int TRIG_PIN = 4;  
const int ECHO_PIN = 5;  
  
//이전상태를 저장할 변수를 만든다.  
int prev = 3;
```

Setup

```
void setup()  
{  
  // 컴퓨터와 아두이노 간의 통신을 세팅해준다.  
  Serial.begin(9600);  
  
  // 초음파 센서를 세팅해준다.  
  pinMode(TRIG_PIN, OUTPUT);  
  pinMode(ECHO_PIN, INPUT);  
}
```

Loop

```
void loop(){  
  // 센서와 물체간의 거리를 계산한다.  
  digitalWrite(TRIG_PIN, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(TRIG_PIN, LOW);  
  int distance = pulseIn(ECHO_PIN, HIGH) * 340 / 2 / 10000;  
  
  // 거리 차이가 급변하면 전 값을 따라간다.  
  if (distance-prev > 2) distance = prev;  
  
  // 거리를 출력한다.  
  Serial.print(distance);  
  Serial.println("cm");  
  
  // 거리를 주파수에 매핑해주고, 출력한다.  
  tone(5, map(distance, 0, 20, 123.5, 261.9));  
  
  //현재값을 이전값으로 저장한다.  
  prev = distance;  
}
```

아두이노 코드 뜯어보기

전처리: 코드 전체에 영향을 미침

Setup: 초기화, 초기 세팅

Loop: 반복되는 실질적 코드

전처리

Setup

Loop

Loop

Loop

Loop

Loop

* 코드 전체에서 사용되는 변수 등은 전처리에서 선언

* 한 Loop 이나 Setup 내에서 선언된 변수는 해당 Loop 이나 Setup 내에서만 사용 가능

기초 코드의 구성

컴퓨터는 생각보다 머리가 안 좋다. = 시키는 거만 할 수 있다. = 우리가 잘 시켜야 한다.

컴퓨터와 대화하기 위한 기초 문법

//(주석): // 이후에 있는 모든 문장은 컴퓨터가 무시한다.

{}(중괄호): 코드 “블록”을 나눠준다.

;(세미콜론): 코드의 마침표 역할.



```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(9, OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite(9, HIGH);  
  delay(5);  
  digitalWrite(9, LOW);  
  delay(5);  
}
```

기초 아두이노 코드의 구성

아두이노가 컴퓨터와 이야기하기 위해서는 특수한 코드들이 필요하다.

`Serial.begin(9600)`: 컴퓨터와 통신을 시작.

`Serial.print(화면에 출력하고픈 것)`: 컴퓨터와 통신으로 원하는 것을 보냄.

`pinMode(핀 숫자, 입력/출력)`: 아두이노 핀이 입력인지 출력인지를 설정.

`digitalWrite(핀, 상태)`, `analogWrite(핀, 상태)`: 핀을 통해 출력한다.

`digitalRead(핀)`, `analogRead(핀)`: 핀에서 값을 입력받는다.

`delay(시간)`: 시간만큼 코드를 멈춘다.

기초 아두이노 코드의 구성

*시리얼 통신을 통해 출력된 값은 돋보기를 눌러 확인할 수 있다.

```
Week4
// 인풋과 아웃풋은 상수로 정의해 나중에 실수로 바뀌거나 섞이는 일이 없도록
const int TRIG_PIN = 2;
const int ECHO_PIN = 3;

//이전상태를 저장할 변수를 만든다.
int prev = 3;

void setup()
{
  // 컴퓨터와 아두이노 간의 통신을 세팅해준다.
  Serial.begin(9600);

  // 초음파 센서를 세팅해준다.
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

void loop() {
  // 센서와 물체간의 거리를 계산한다.
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int distance = pulseIn(ECHO_PIN, HIGH) * 340 / 2 / 10000;

  // 거리 차이가 급변하면 전 값을 따라간다.
  if (distance - prev > 2) distance = prev;

  // 거리를 출력한다.
  Serial.print(distance);
  Serial.println("cm");
}
```

Serial.begin(9600) → Serial.begin(9600);

pinMode(핀, 입력/출력) → pinMode(TRIG_PIN, OUTPUT);

digitalWrite(핀, 상태) → digitalWrite(TRIG_PIN, HIGH);

delay(시간) → delayMicroseconds(10);

Serial.print(화면에 출력하고픈 것) → Serial.print(distance);

34 Arduino Uno on COM10

변수, 연산자, 그리고 제어문

코드에서 사용되는 값들은 변수라는 공간에 보관된다.

자주 쓰이는 변수의 종류: int(정수), float(유리수), boolean(참/거짓)

변수 선언방법: *변수종류 변수이름 = 변수값;*

int a = 9;	a라는 정수변수에는 9의 값을 보관한다.
boolean k = true;	k라는 논리변수에는 참의 값을 보관한다
float c = 100.7;	c라는 유리수변수에는 1000.7이라는 값을 보관한다.

변수, 연산자, 그리고 제어문

코딩에는 오칙연산이 있다.

기본 연산자: +(더하기), -(빼기), *(곱하기), /(나누기), %(나머지)

$a = 9 + 8;$

a라는 변수에 9+8(17)의 값을 보관한다.

$a = a / 5;$

a라는 변수에 저장된 값을 5로 나눈 뒤 a에 보관한다. = a 를 5로 나눈다.

$a = 78 \% 4;$

a라는 변수에 78을 4로 나눈 나머지(2)의 값을 보관한다.

변수, 연산자, 그리고 제어문

비교 연산자를 통해 조건을 만들 수 있다.

기본 연산자: $>$, $>=$, $<$, $<=$, $==$, $!=$

`if( a > 5 )`

a에 저장된 값이 5보다 크다면

`if( a == b )`

a에 저장된 값이 b에 저장된 값과 같다면

`if( b <= 15 )`

b에 저장된 값이 15보다 작거나 같다면

* 비교의 경우, 같다는 하나의 `=`가 아닌 두개의 `==`를 사용한다.

* `!`는 not의 의미를 가진다.

변수, 연산자, 그리고 제어문

조건은 여러 개 걸 수 있다.

기본 연산자: &&(and), ||(or), !(not)

if(a > 5 && a == b) a에 저장된 값이 5보다 크고, a에 저장된 값이 b에 저장된 값과 같다면

if(a > 5 || a == b) a에 저장된 값이 5보다 크거나, a에 저장된 값이 b에 저장된 값과 같다면

if(!k) 논리변수 k에 저장된 값의 반대가 참이라면 = k가 거짓이라면

변수, 연산자, 그리고 제어문

If문은 최고의 친구이다.

기본 제어문: *if(조건) {본 코드;} else {본 코드;}* 조건이 참일 경우, 코드가 작동하고, 아니라면 else로 넘어간다.

<code>if(a > 5) {</code>	만약 a가 5보다 크다면 a의 값을 출력하라.
<code> Serial.print(a);</code>	
<code>} else if(a < 2) {</code>	그렇지 않고, a가 2보다 작으면 “작다” 라고 출력하라.
<code> Serial.print(“작다”);</code>	
<code>} else {</code>	
<code> a = a + 1;</code>	그렇지 않다면, a에 값에 1을 더해라.
<code>}</code>	

변수, 연산자, 그리고 제어문

코드를 반복하는 여러가지 방법

기본 제어문: *for (int i = 0; i < 반복 횟수; i++){본 코드;}* 혹은 *while(조건){본 코드;}*

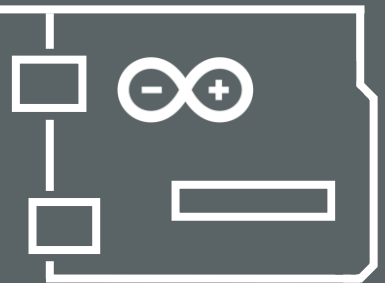
```
for (int i = 0; i < 10; i++){  
    Serial.print("반복");  
}
```

“반복”이라는 문구를 10번 출력해라.

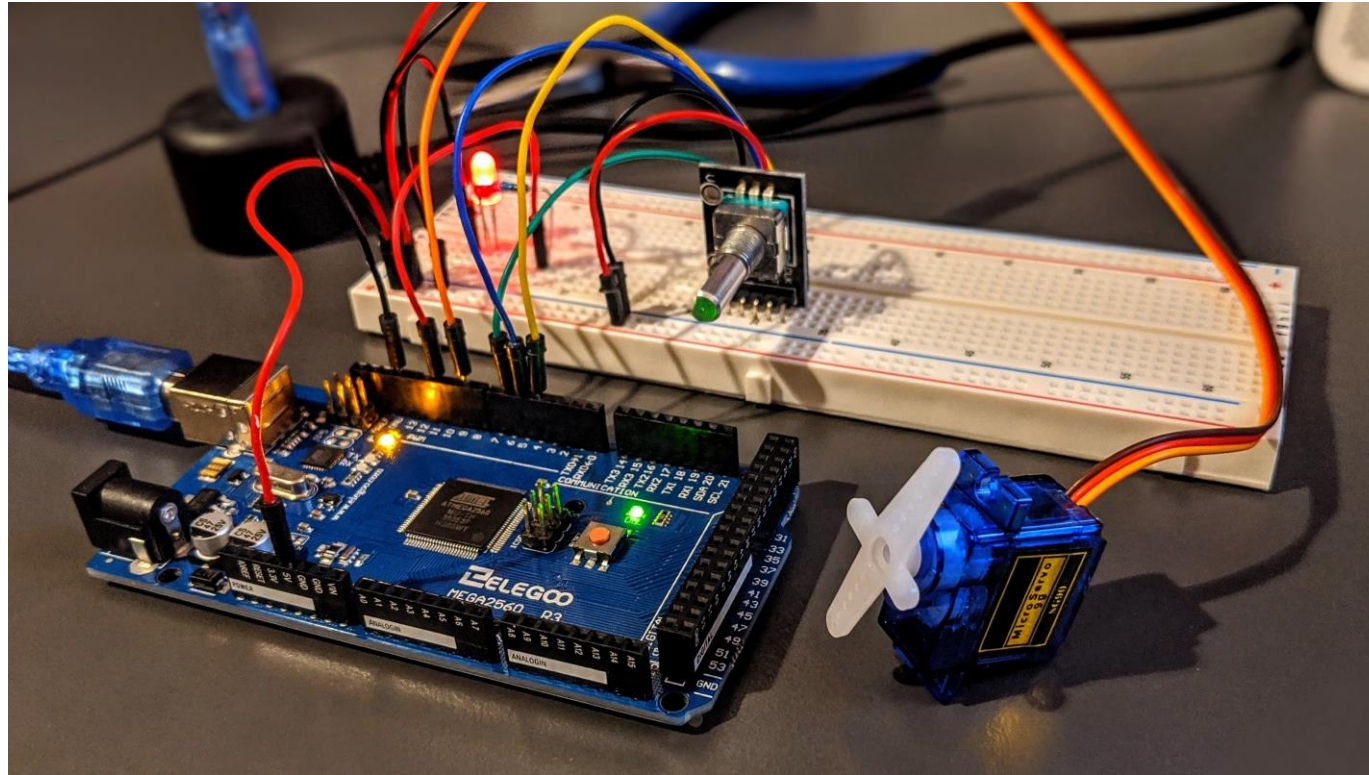
```
int i = 0  
while (i < 10){  
    Serial.print("반복");  
    i = i + 1;  
}
```

“반복”이라는 문구를 10번 출력해라

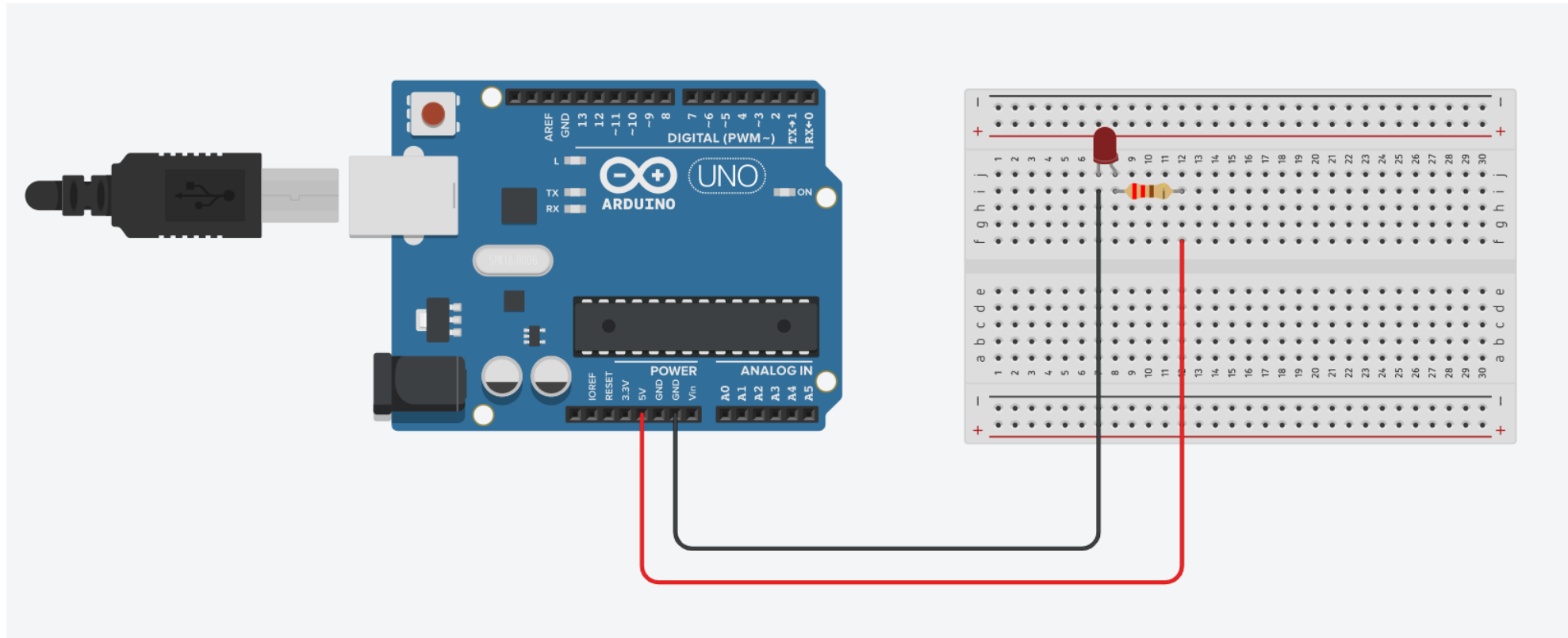
프로젝트 실습하기



아두이노와 전기

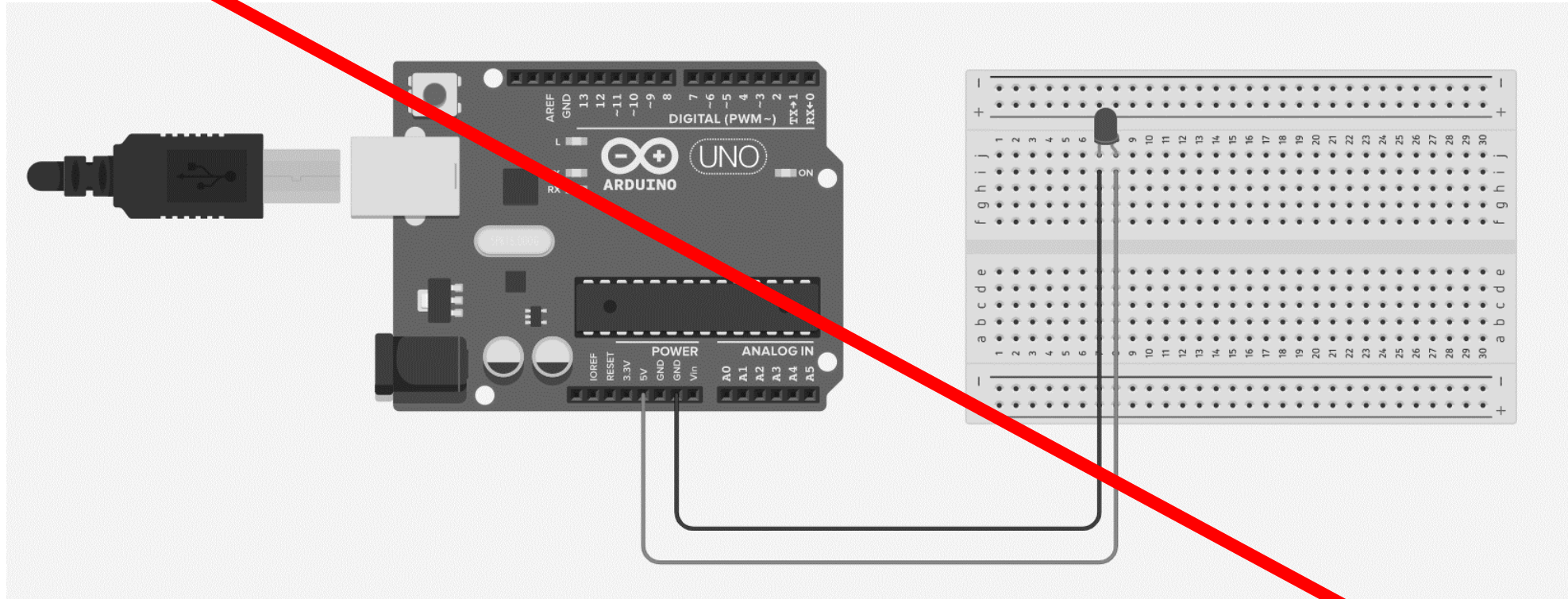


LED 사용해보기

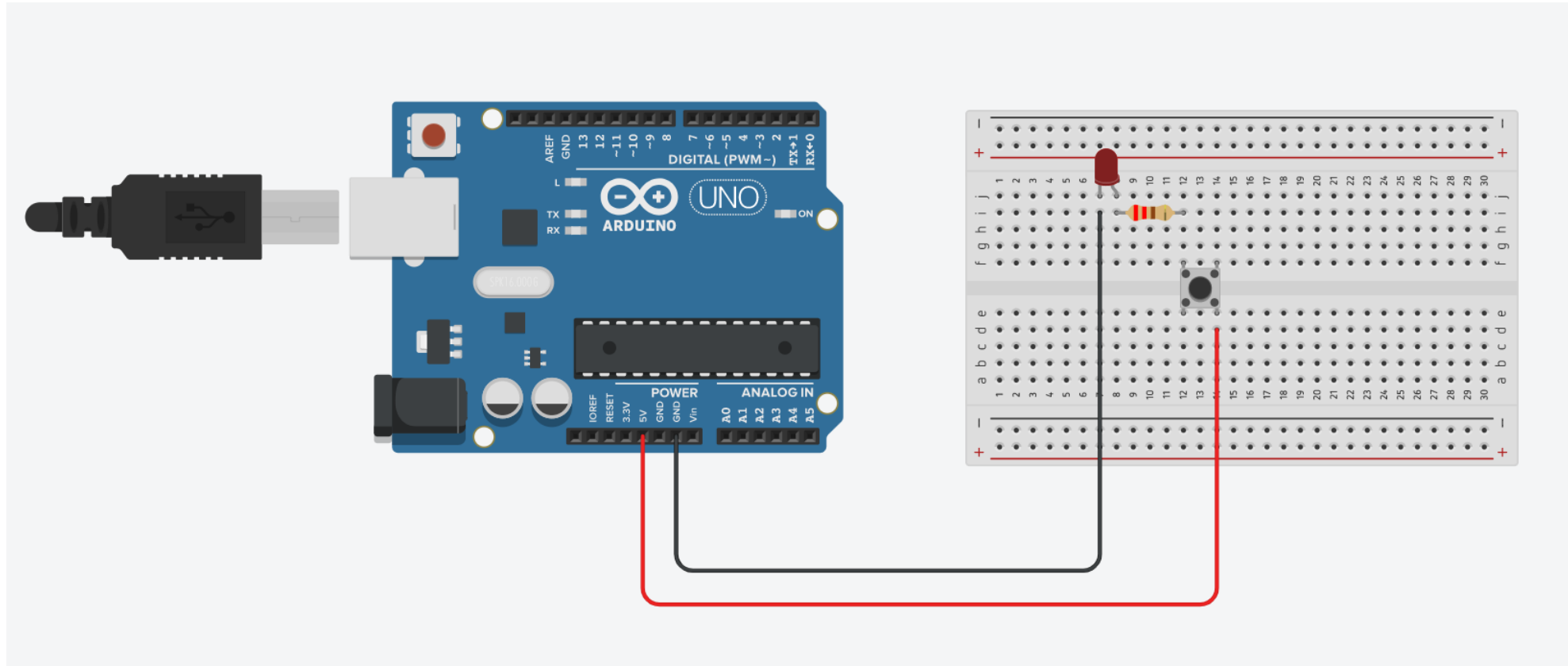


LED 사용해보기

저항을 연결하지 않은 LED는 회로를 태울 수 있으니 주의

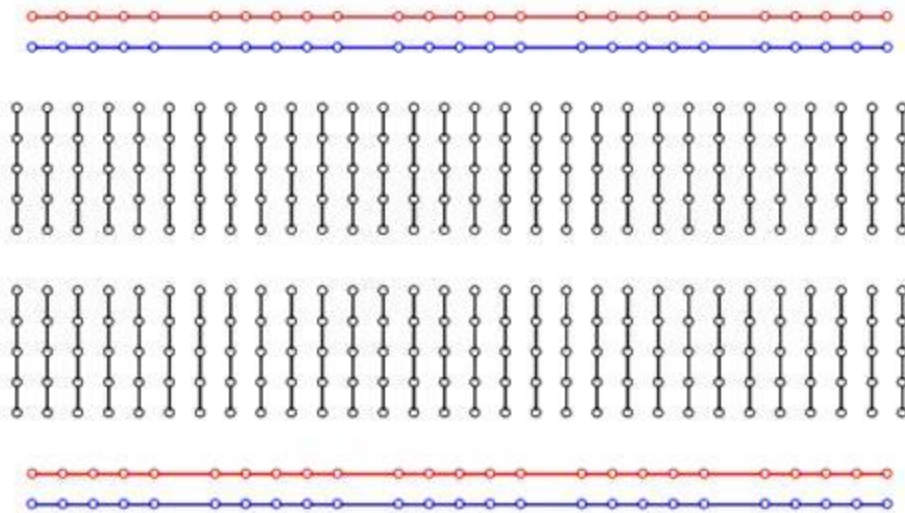


LED 사용해보기



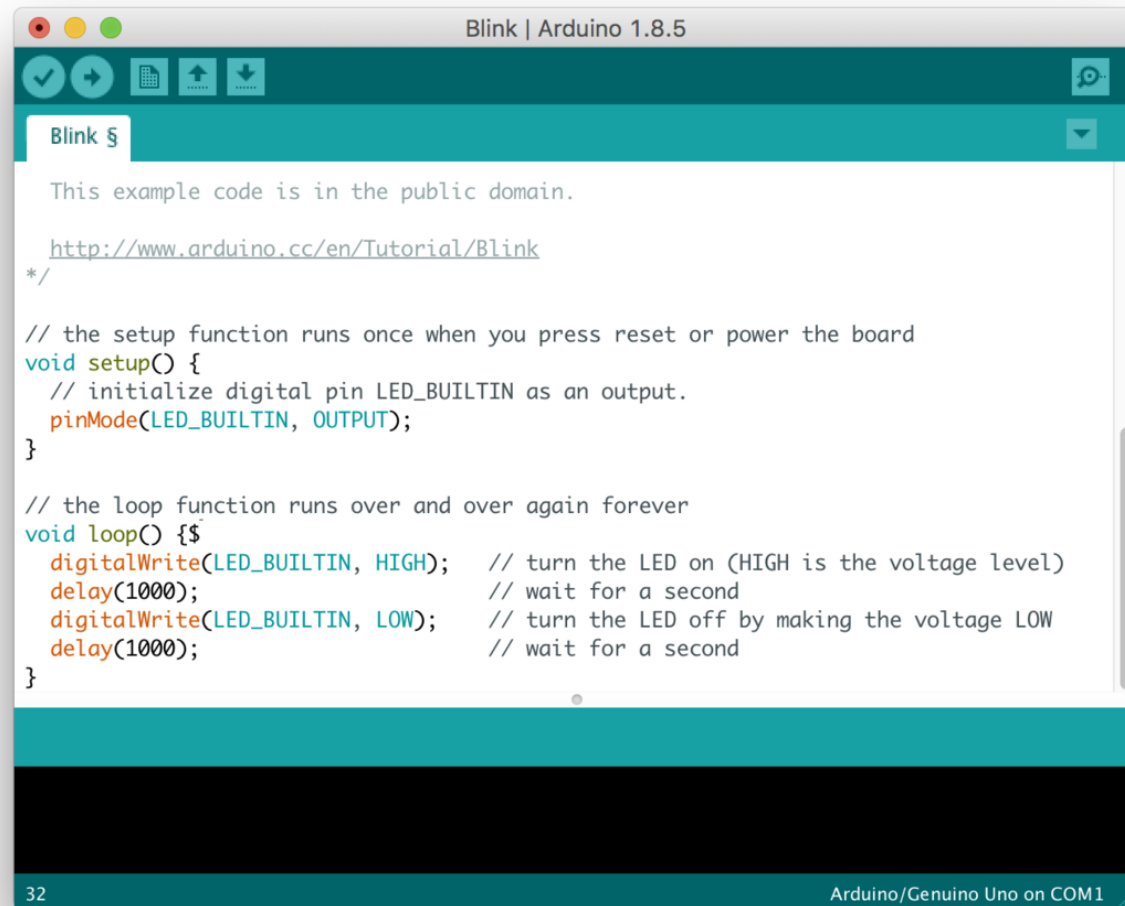
전기전자 핑크퐁, 브레드보드

브레드보드는 최종 프로젝트 만들기 전 편하게 테스트 할 수 있는 공간이다.



브레드보드는 가로전선과 세로전선으로 이어져 있다.

예제 코드 실행해보기



The screenshot shows the Arduino IDE interface with the 'Blink' example code loaded. The window title is 'Blink | Arduino 1.8.5'. The code is as follows:

```
This example code is in the public domain.

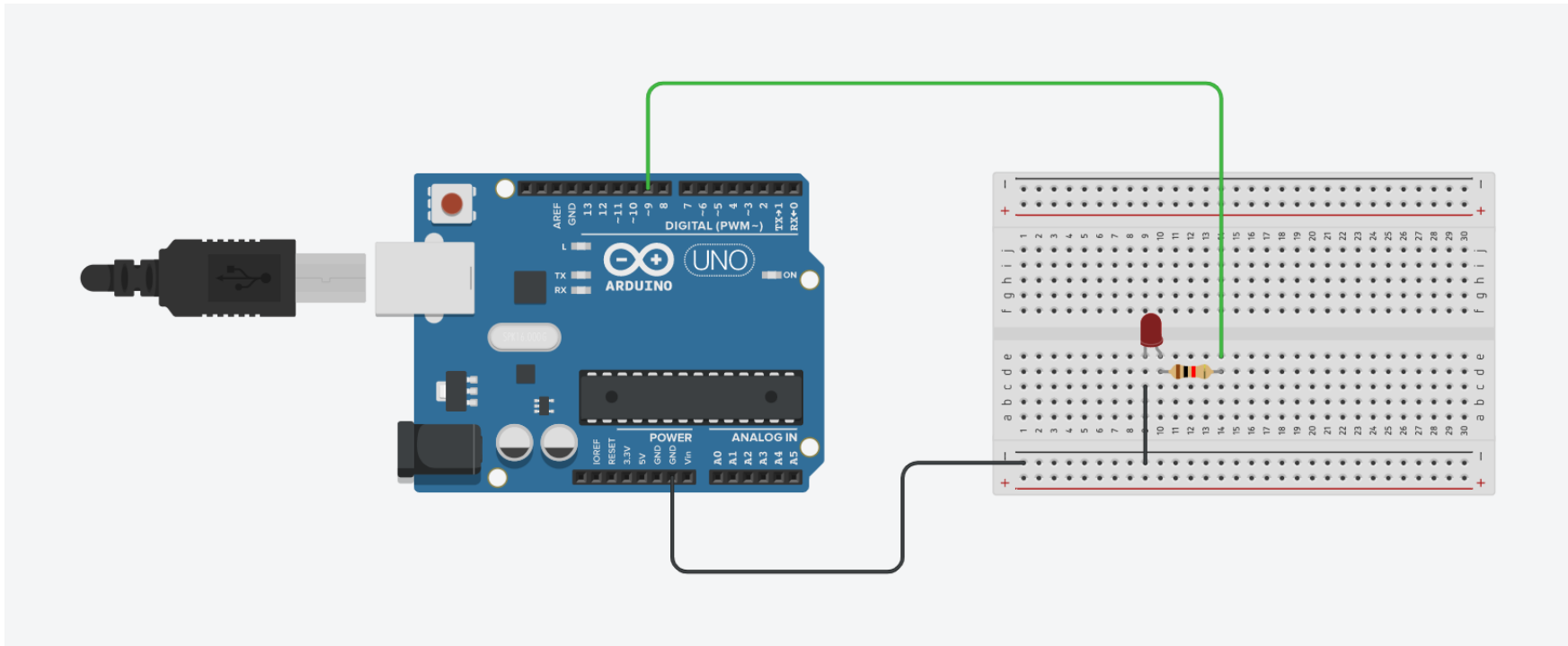
http://www.arduino.cc/en/Tutorial/Blink
*/

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

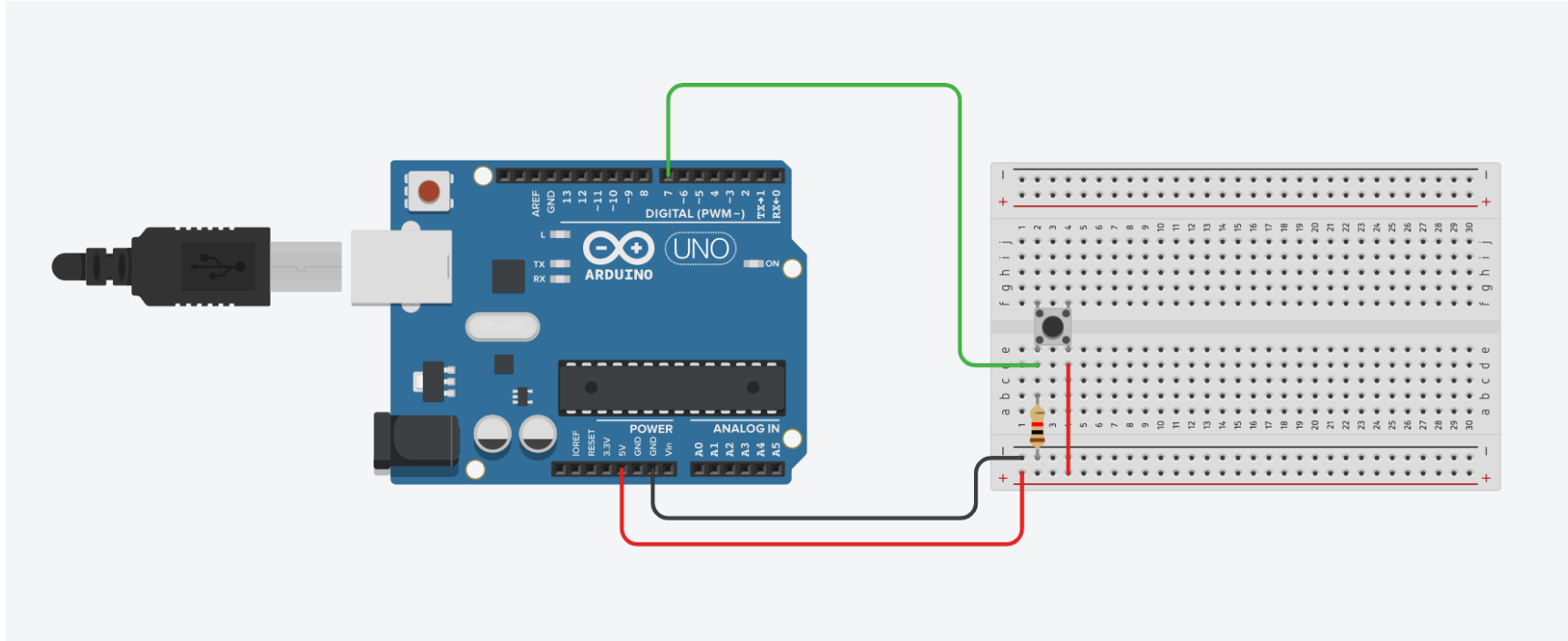
// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);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

The status bar at the bottom indicates '32' and 'Arduino/Genuino Uno on COM1'.

예제 코드 실행해보기



버튼 사용하기



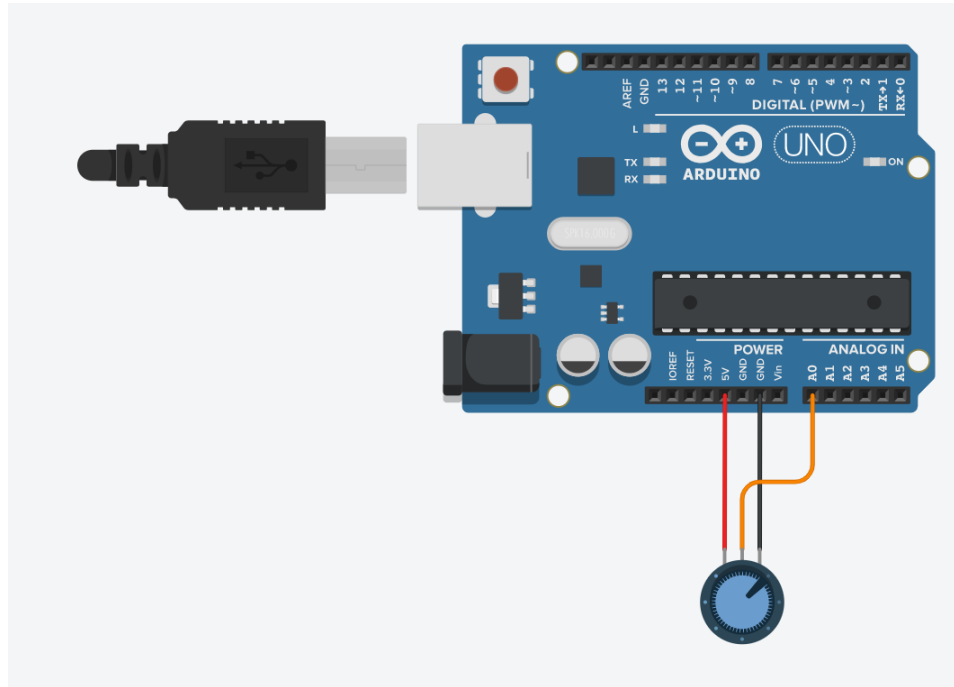
버튼 사용하기

```
int ledPin = 13;  // choose the pin for the LED
int inputPin = 7; // choose input pin 7 for the push button

void setup()
{
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare push button as input
}

void loop()
{
  int pushed = digitalRead(inputPin); // read input value
  if (pushed == HIGH){ // check if the input is HIGH
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else{
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```

가변저항 사용하기



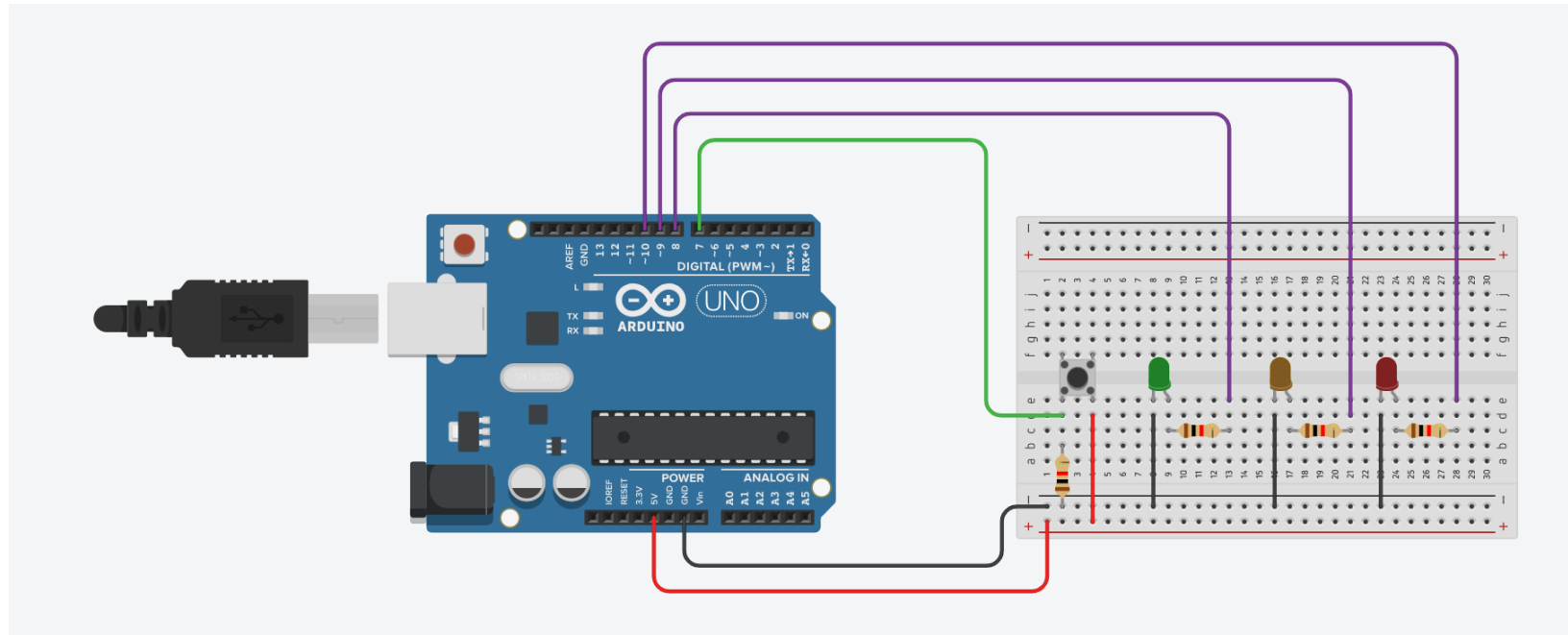
가변저항 사용하기

```
const int potpin = 0; // choose input pin A0 for potentiometer
int val;

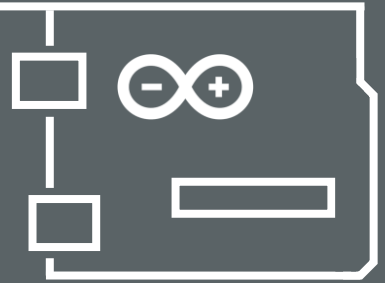
void setup() {
  Serial.begin(9600); // start communication with arduino
}

void loop() {
  val = analogRead(potpin); // read input
  Serial.println(val); // print input
}
```

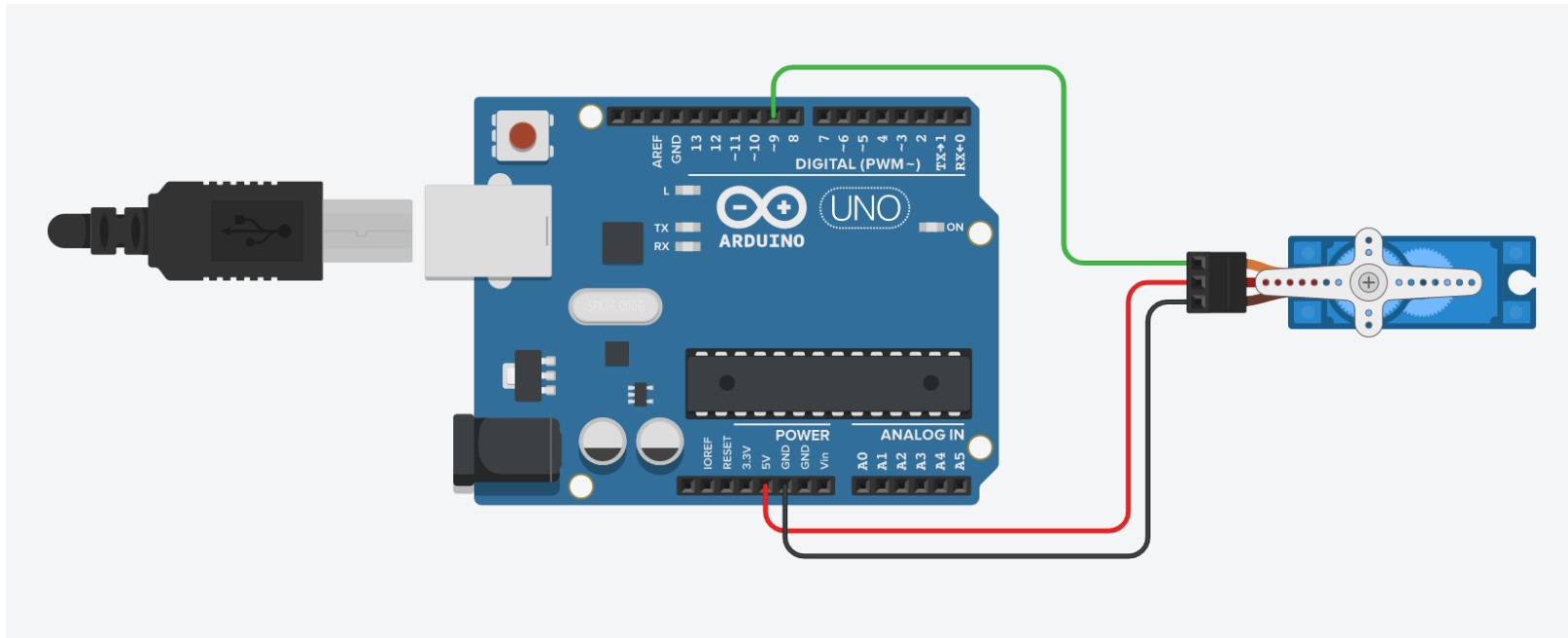
프로젝트 만들어보기: 신호등



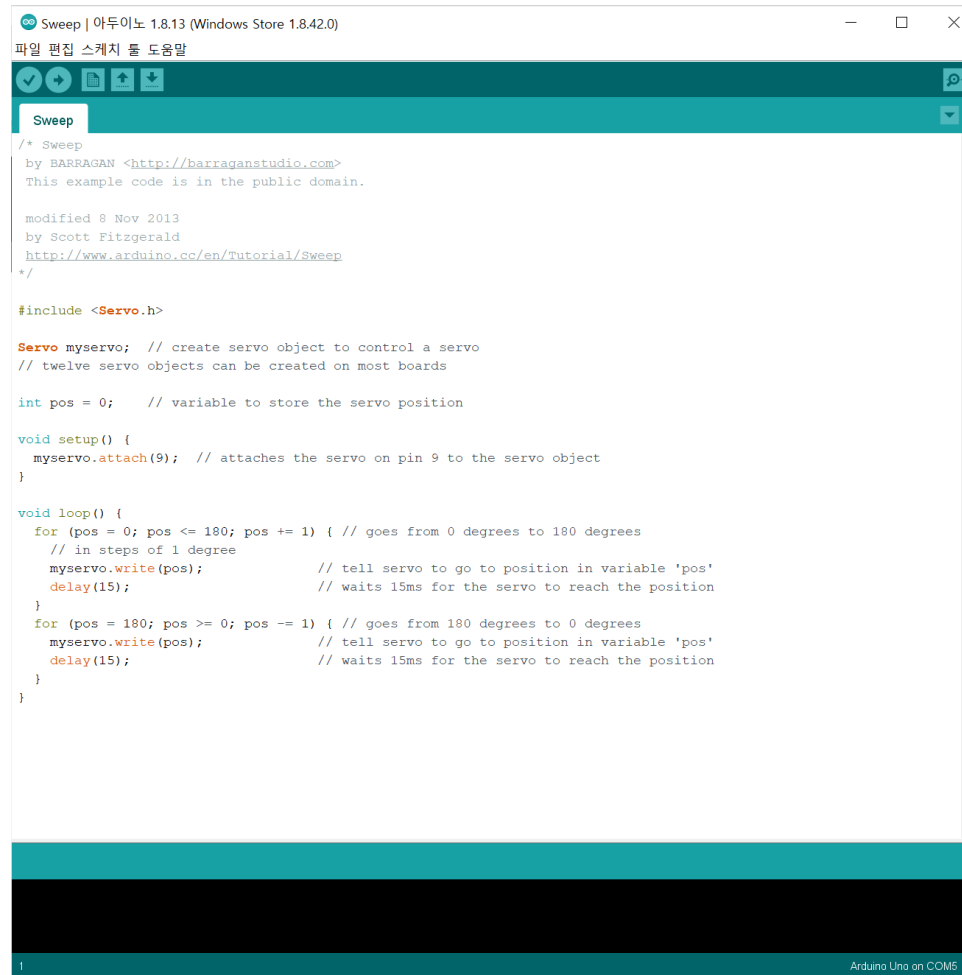
추가학습



서보모터 사용하기



서보모터 사용하기



```
/* Sweep
by BARRAGAN <http://barraganstudio.com>
This example code is in the public domain.

modified 8 Nov 2013
by Scott Fitzgerald
http://www.arduino.cc/en/Tutorial/Sweep
*/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

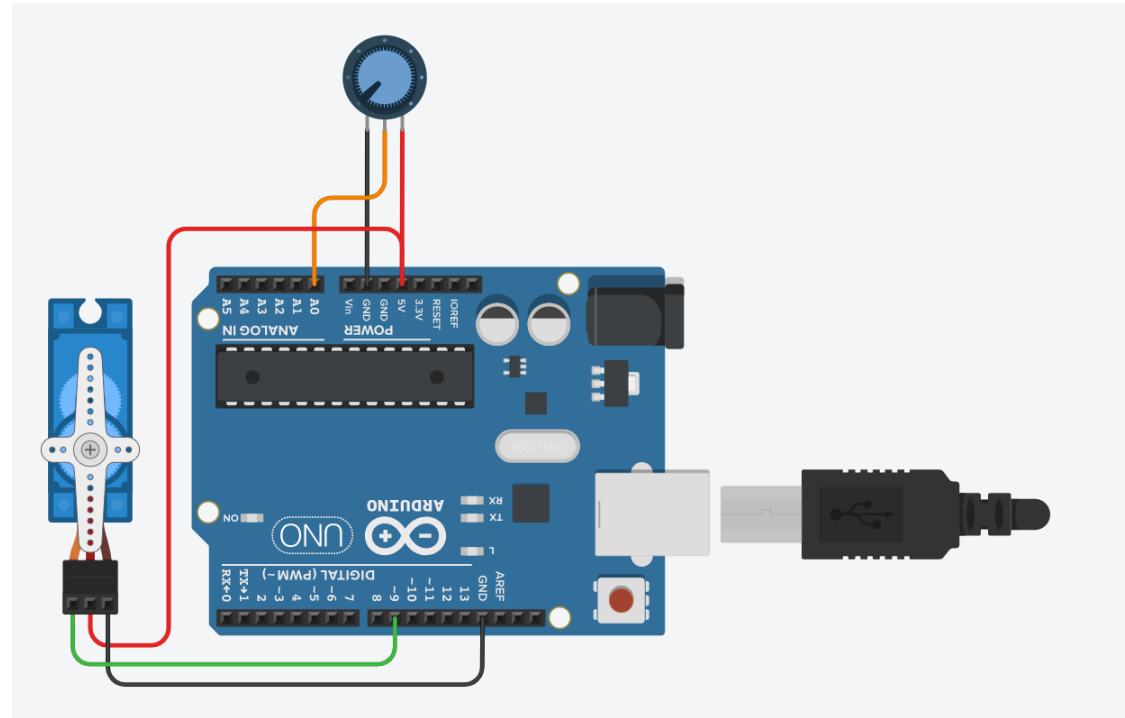
int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

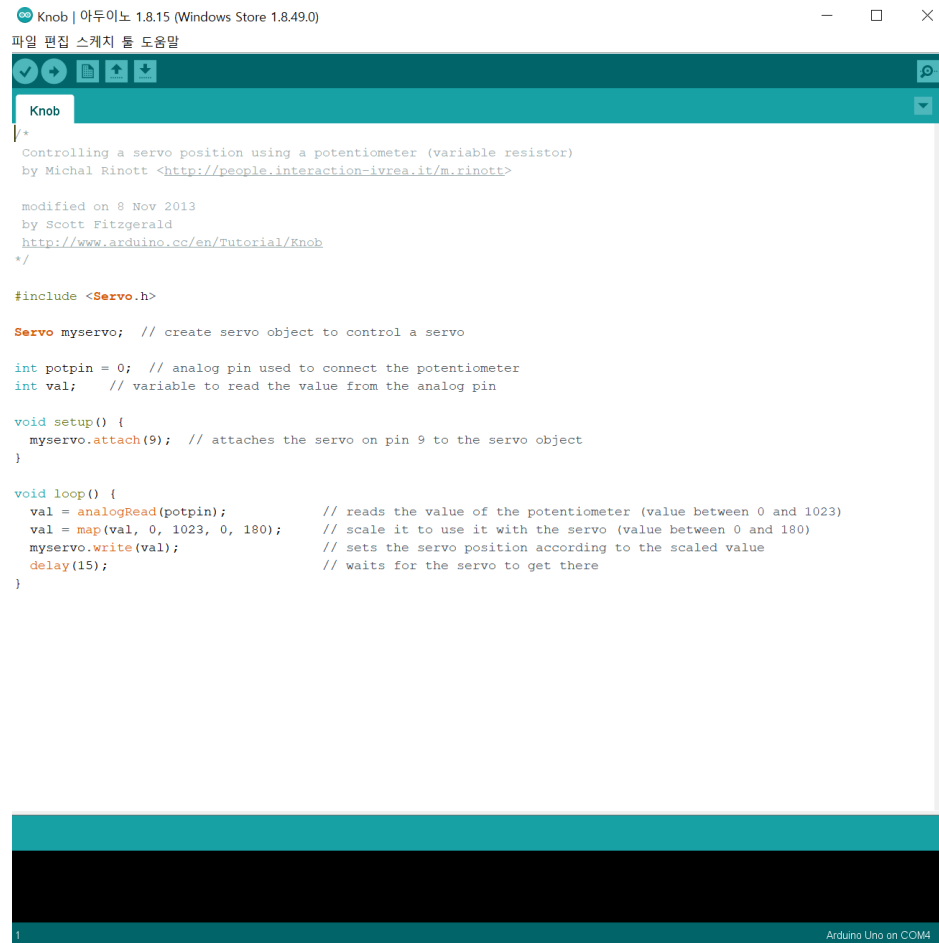
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

1 Arduino Uno on COM5

서보모터 사용하기



서보모터 사용하기



The screenshot shows the Arduino IDE interface. The title bar reads "Knob | 아두이노 1.8.15 (Windows Store 1.8.49.0)". The menu bar includes "파일", "편집", "스케치", "툴", and "도움말". The toolbar contains icons for opening, saving, and other functions. The "Knob" tab is active, displaying the following code:

```
/*
Controlling a servo position using a potentiometer (variable resistor)
by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

modified on 8 Nov 2013
by Scott Fitzgerald
http://www.arduino.cc/en/Tutorial/Knob
*/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo

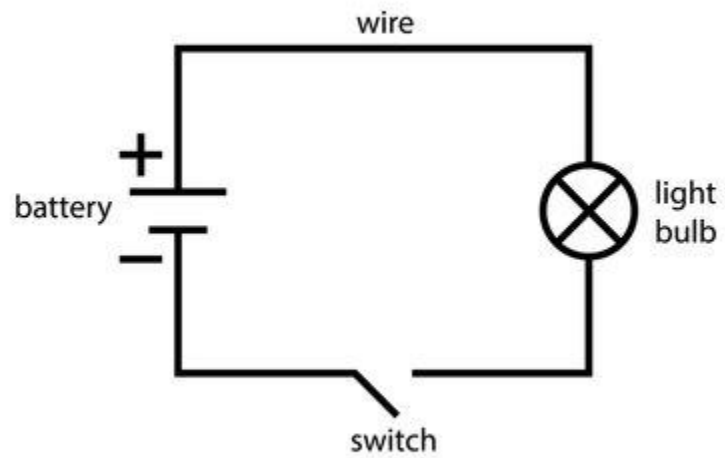
int potpin = 0;  // analog pin used to connect the potentiometer
int val;         // variable to read the value from the analog pin

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

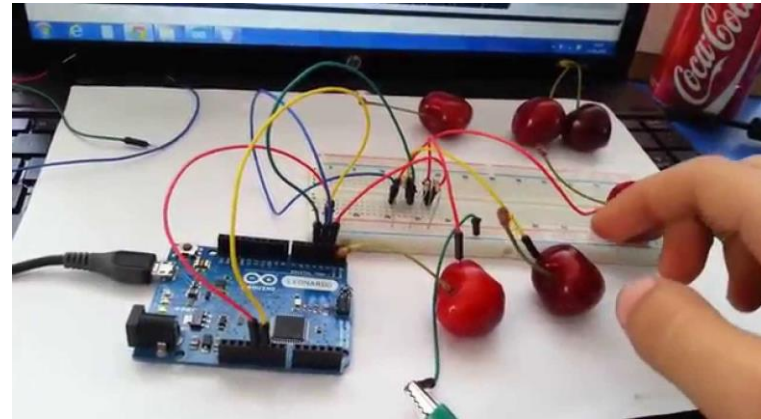
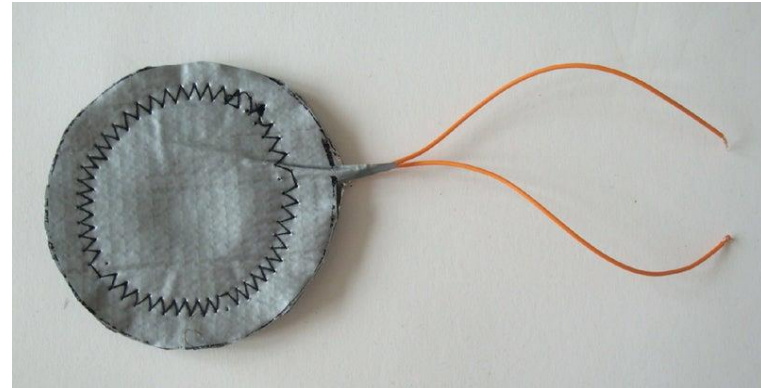
void loop() {
  val = analogRead(potpin);  // reads the value of the potentiometer (value between 0 and 1023)
  val = map(val, 0, 1023, 0, 180);  // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val);  // sets the servo position according to the scaled value
  delay(15);           // waits for the servo to get there
}
```

The status bar at the bottom indicates "1" and "Arduino Uno on COM4".

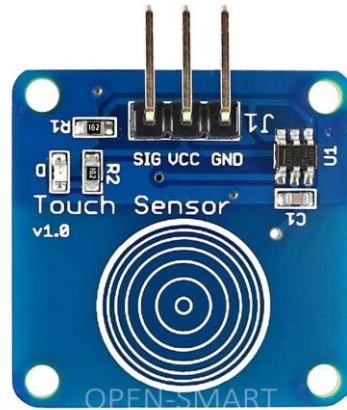
버튼 만들기



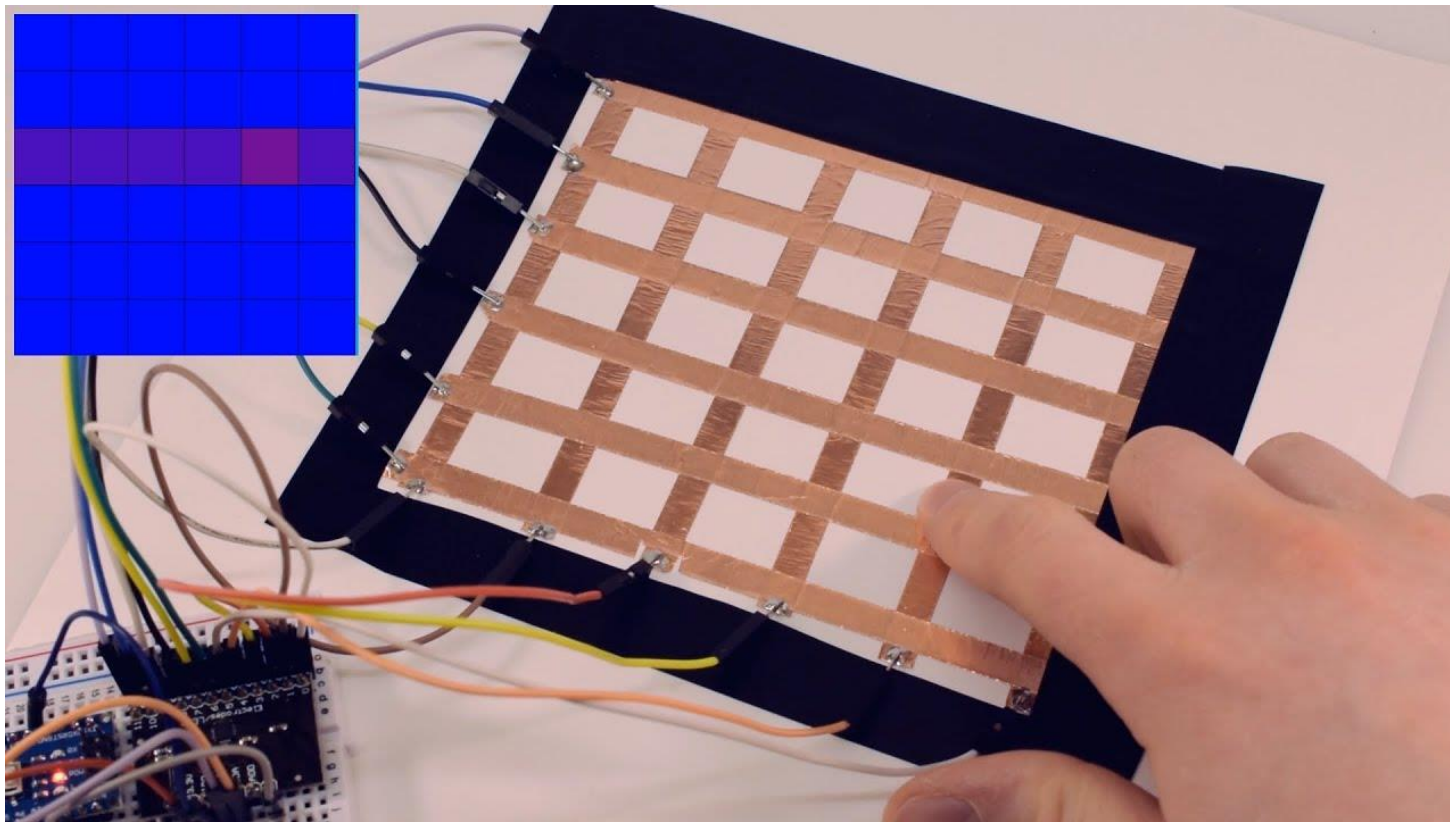
버튼 만들기



정전식 터치



정전식 터치



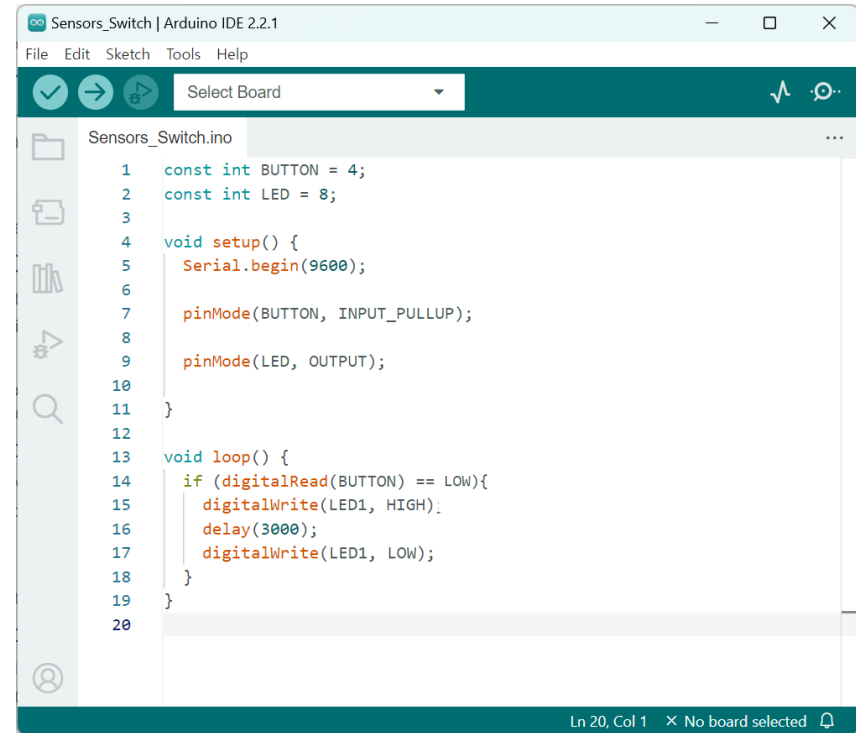
delay/millis/count

이름	형식	내용
delay()	함수	해당 시간동안 아두이노를 멈춘다 .
millis()	함수	현재 시각 을 받아 지난 시간을 계산한다.
counting	방법	loop마다 숫자를 세어 지난 loop 의 수를 계산 한다.

delay()

아두이노 전체를 중단시킨다.

(괄호 내 숫자 *1000) 초 만큼.



```
1  const int BUTTON = 4;
2  const int LED = 8;
3
4  void setup() {
5      Serial.begin(9600);
6
7      pinMode(BUTTON, INPUT_PULLUP);
8
9      pinMode(LED, OUTPUT);
10
11 }
12
13 void loop() {
14     if (digitalRead(BUTTON) == LOW){
15         digitalWrite(LED1, HIGH);
16         delay(3000);
17         digitalWrite(LED1, LOW);
18     }
19 }
20
```

Ln 20, Col 1 × No board selected

millis()

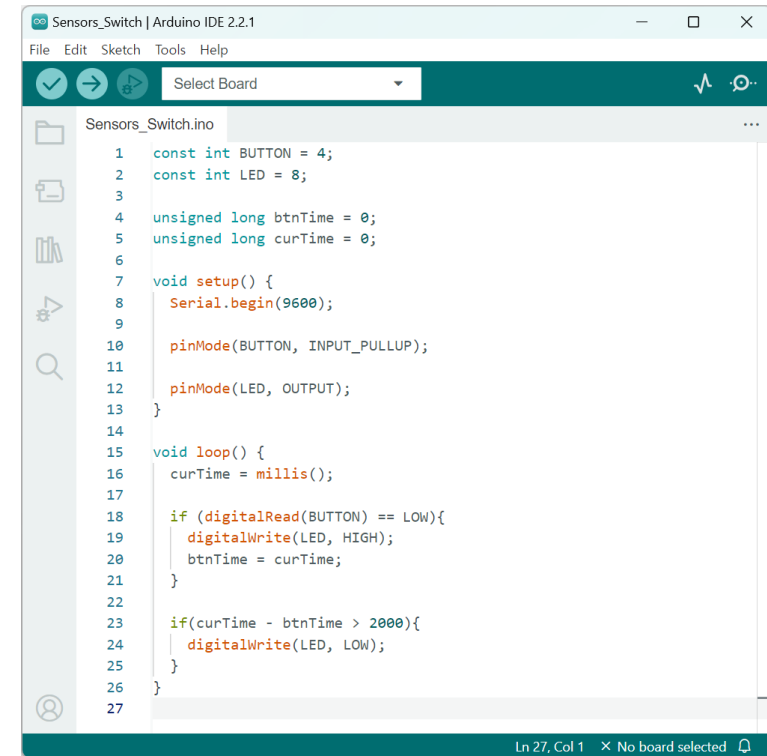
아두이노가 시작하고 지난 밀리 초 숫자를 반환한다.

기록용 변수를 지정하고, 트리거의 시각을 기록한다.
이후 해당 시각과 기록된 시각의 차이를 계산한다.

: 입력장치를 사용할 때 조심해야 한다.
(계속 입력 받는 중)

* 약 50 일 후에 오버플로우(O으로 돌아감)가 된다.

* 반환 값은 unsigned long 이므로 프로그래머가 int 와 같은 작은 자료형으로 산술을 수행하려고하면 논리 오류가 발생할 수 있다.



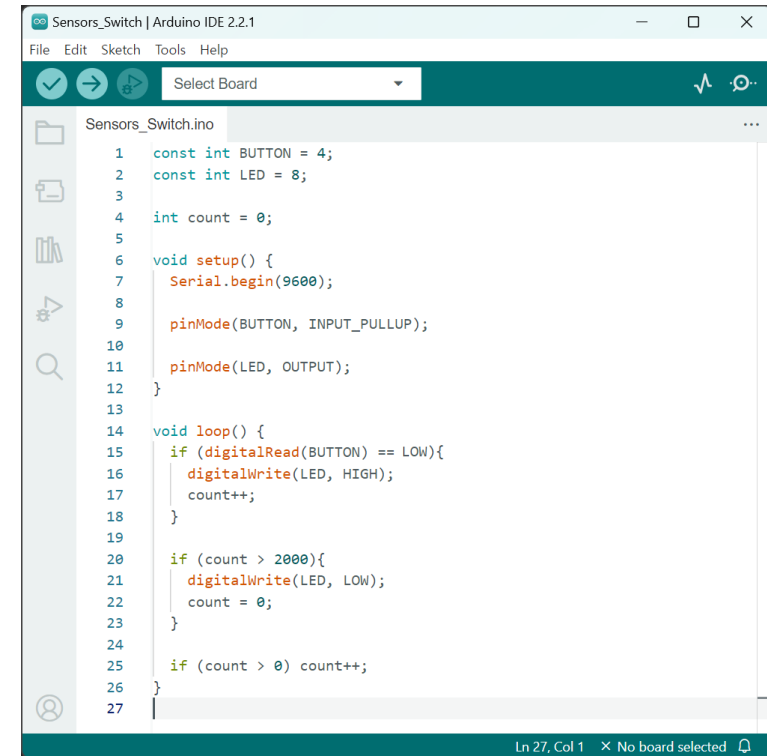
```
1  const int BUTTON = 4;
2  const int LED = 8;
3
4  unsigned long btnTime = 0;
5  unsigned long curTime = 0;
6
7  void setup() {
8      Serial.begin(9600);
9
10     pinMode(BUTTON, INPUT_PULLUP);
11     pinMode(LED, OUTPUT);
12 }
13
14
15 void loop() {
16     curTime = millis();
17
18     if (digitalRead(BUTTON) == LOW){
19         digitalWrite(LED, HIGH);
20         btnTime = curTime;
21     }
22
23     if(curTime - btnTime > 2000){
24         digitalWrite(LED, LOW);
25     }
26 }
27
```

counting

몇 loop이 지났는지 기록을 한다.

기록용 변수를 지정하고, 트리거에서 세기 시작한다.
원하는 횟수가 지난 후, 리셋시킨다.

: 단순한 시간보다는 조건을 확인하는데 적절
ex) 버튼 누른 횟수



```
1 const int BUTTON = 4;
2 const int LED = 8;
3
4 int count = 0;
5
6 void setup() {
7   Serial.begin(9600);
8
9   pinMode(BUTTON, INPUT_PULLUP);
10
11   pinMode(LED, OUTPUT);
12 }
13
14 void loop() {
15   if (digitalRead(BUTTON) == LOW){
16     digitalWrite(LED, HIGH);
17     count++;
18   }
19
20   if (count > 2000){
21     digitalWrite(LED, LOW);
22     count = 0;
23   }
24
25   if (count > 0) count++;
26 }
27
```

참고할 만한 사이트

<https://www.arduino.cc/reference/en/> - 아두이노 코딩언어 관련

<https://codingrun.com/> - 아두이노 기초 개념

<https://youtu.be/DYTYEcSIFKY> - 아두이노 기초 개념

<https://youtu.be/3e6GMI4Szwk> - 아두이노 기초 개념

<https://mechasolution.com/shop/main/index.php> - 부품구매 및 찾기

<https://www.adafruit.com/> - 부품구매 및 찾기

<https://www.google.com/>

<https://www.youtube.com/>

끝!