



**설리번 프로젝트**  
**나만의 무드등 만들기**  
**오리엔테이션**

**설리번프로젝트**  
**권태훈 권보성 한효민 천예준**

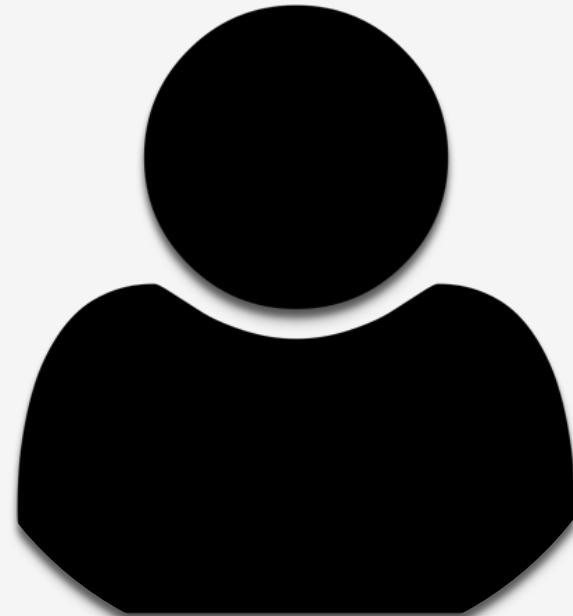
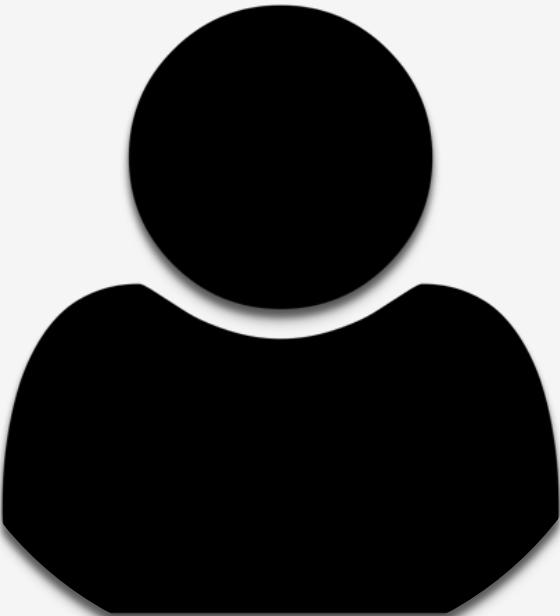
# 저희는 고등학생 입니다

편하게 대해주세요!

저희는 아이들과 함께  
무드등 만들기를 원할 뿐입니다.

저희와 함께 해주셔서 감사합니다.

# 함께할 선생님을 소개합니다



권보성

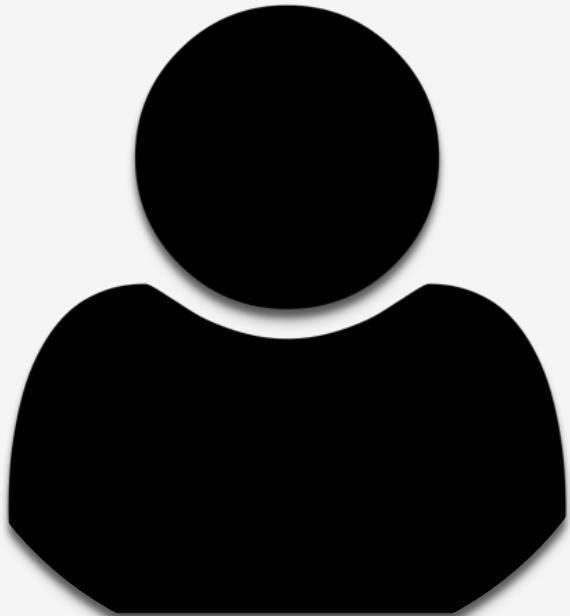
한국디지털미디어고등학교  
배드민턴 배드민턴 팡팡

권태훈

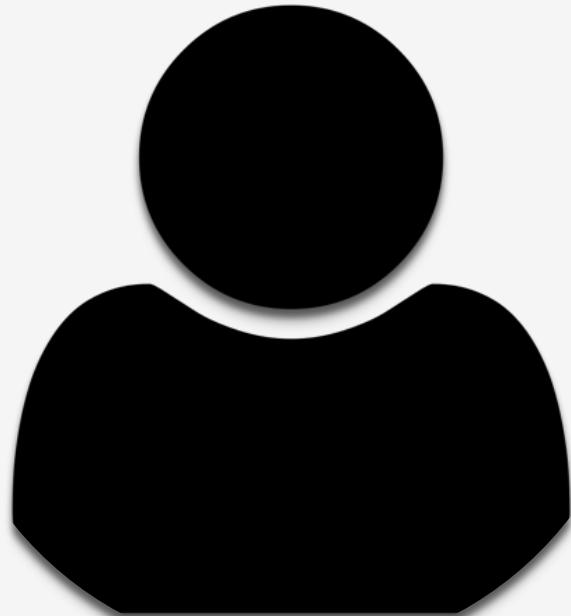
광동고등학교  
체교 가고 싶은 개발자

# 함께할 선생님을 소개합니다

03 ————— 04



한효민  
한국디지털미디어고등학교  
누구보다 친절하고 즐겁게 >\_<



천예준  
한국디지털미디어고등학교  
늘 새로운 것을 좋아하고

# 설리번 프로젝트는 무엇인가요

설리번 프로젝트는 코딩에 관심이 많고  
만들고 싶은 것도 많지만 도움을 받지 못하던  
지역의 학생들에게 코딩으로 만들고 싶은 것을  
만들 수 있도록 돕기 위해 시작한  
고등학생 주도의 코딩 교육봉사 [프로젝트](#)입니다.

# 강의는 어떻게 구성되나요

강의는 총 6차시로 1월 8일부터 1월 20일까지  
매주 월요일, 수요일, 금요일  
오후 3시 30분부터 6시까지  
한켠작업실에서  
3주동안 진행합니다.

원활한 진행을 위해 수업 시작 10분 전까지는  
이곳으로 모여주세요!

노트북을 꼭 지참해주세요!

# **연락수단은 뭐가 있을까요 우리 아이들을 보고싶어요**

카톡방을 운영하여  
매 수업마다 간략하게 공지를 할 예정입니다.

아이들이 수업에 참여하는 모습을  
틈틈히 찍어 카톡방에 공유할 예정입니다.

사진은 카톡방 공유과 설리번 프로젝트를 위해서만  
쓰이며, 다른 용도로는 쓰이지 않습니다.

# 그 외 자주 묻는 질문

Q. 학부모도 수업 참여가 가능한가요?

A. 죄송하지만, 학부모님께서는 수업에 참여하실 수 없습니다.

Q. 교육비가 필요한가요? 얼마인가요?

A. 교육은 무료로 진행합니다. 다만, 5~6차시 수업 진행을 위해 개인부담이 있을 수 있습니다. 이 점은 추후 자세히 공지하겠습니다.

**설리번 프로젝트**  
**나만의 무드등 만들기**  
**지금 시작합니다**

**설리번프로젝트**  
**권태훈 권보성 한효민 천예준**





아이스 브레이킹을 할거에요



모두 kahoot.it으로!



재미있으셨나요?



본격적으로 아두이노에 대해  
알아봅시다!

Q. 우리는 아두이노로 무엇을 만들 수 있을까요?



# ***TOP 10 Arduino Projects***

# 아두이노(arduino)

오픈소스를 기반으로 한 마이크로 컨트롤러(소형 컴퓨터)

이건 꼭!

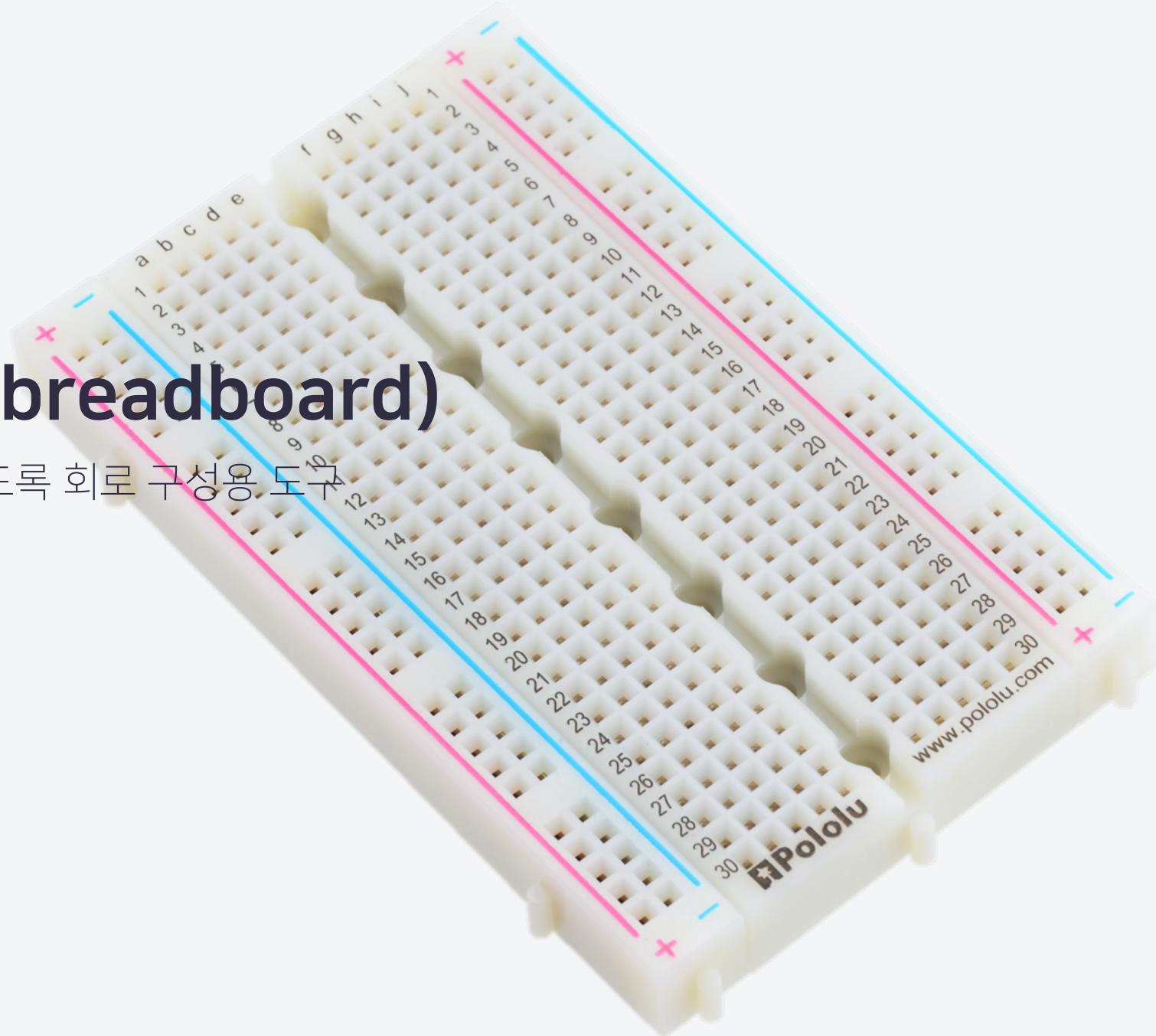
전압.

전류.

저항.

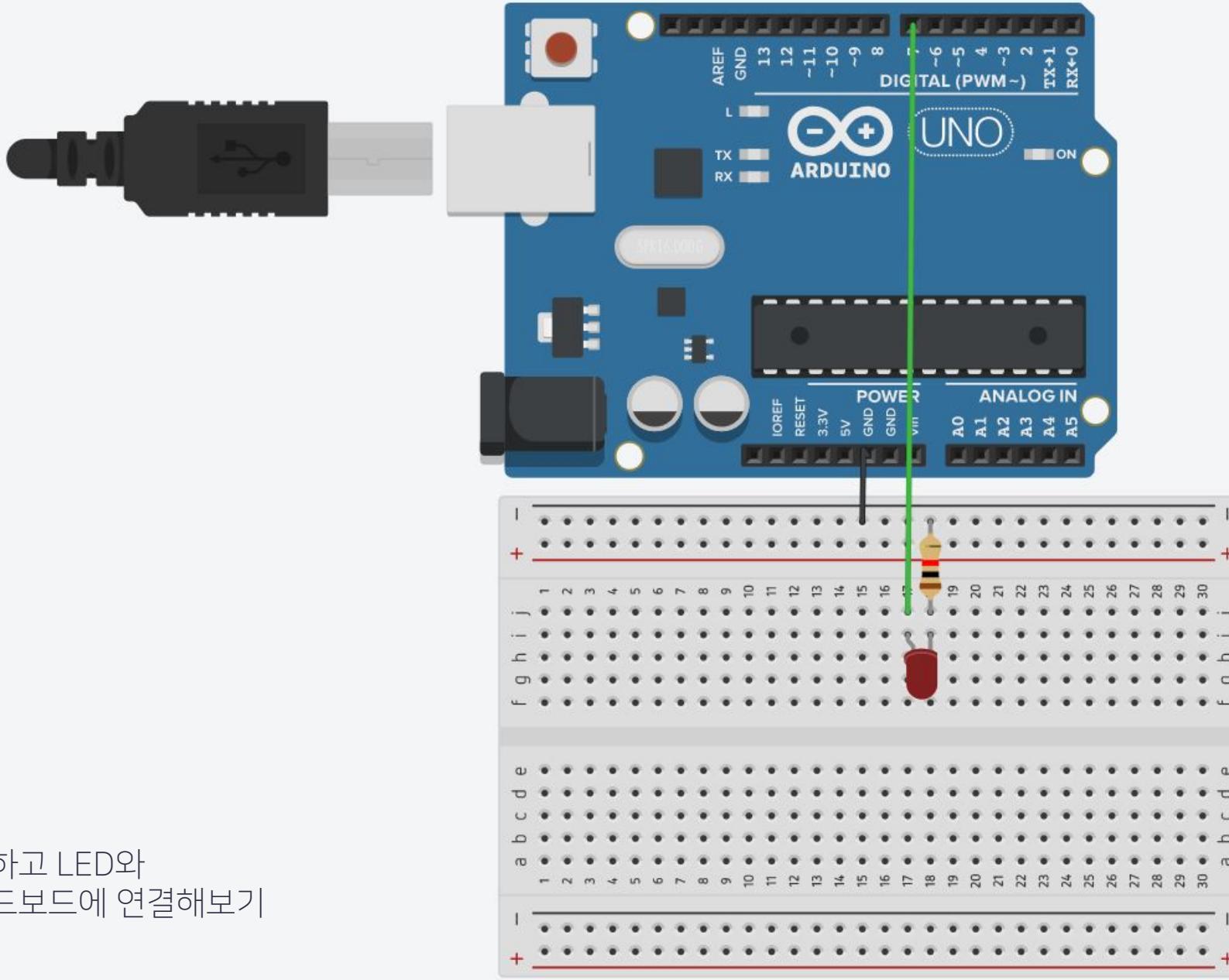
# 브레드보드(breadboard)

회로를 쉽게 구성할 수 있도록 회로 구성용 도구



# 급습

:전선을 연결하고 LED와  
저항을 브레드보드에 연결해보기





# Download the Arduino IDE



## ARDUINO 1.8.10

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer, for Windows XP and up

**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10



**Mac OS X** 10.8 Mountain Lion or newer

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32 bits

**Linux** ARM 64 bits

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

# <https://www.arduino.cc/en/Main/Software>

## HOURLY BUILDS

LAST UPDATE  
30 December 2019 18:13:24 GMT

Download a [preview of the incoming release](#) with the most updated features and bugfixes.

[Windows](#)

[Mac OS X](#) (Mac OSX Mountain Lion or later)

[Linux 32 bit](#), [Linux 64 bit](#), [Linux ARM](#), [Linux ARM64](#)

## BETA BUILDS

**BETA**

Download the **Beta Version** of the Arduino IDE with experimental features. This version should NOT be used in production.

[Windows](#)

[Mac OS X](#) (Mac OSX Mountain Lion or later)

[Linux 32 bit](#), [Linux 64 bit](#), [Linux ARM](#), [Linux ARM64](#)

# 실습

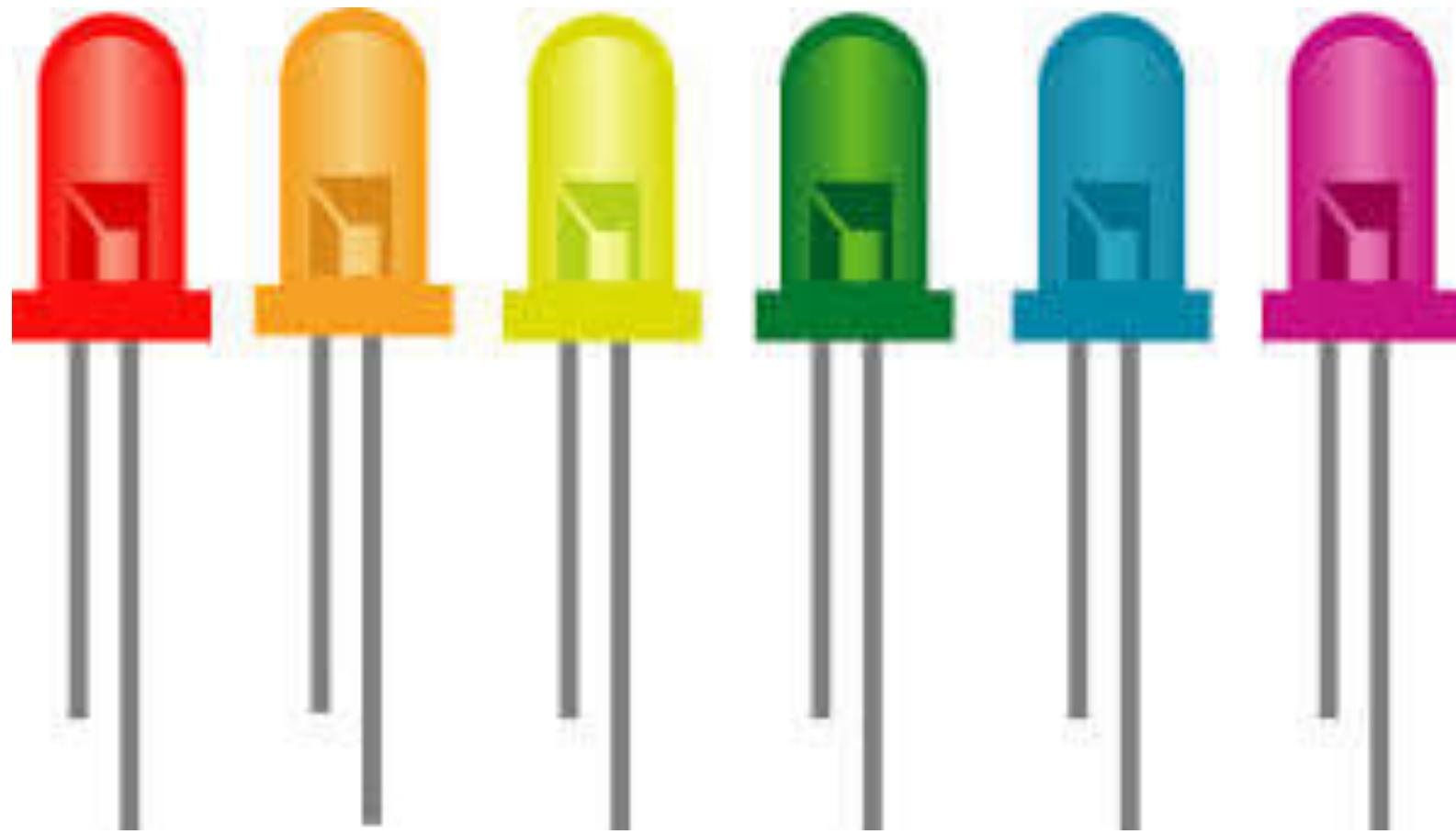
:아두이노 IDE에  
LED를 1초 간격으로 껐다가  
꺼다가를 반복하는 프로그램  
작성하고 업로드하기

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

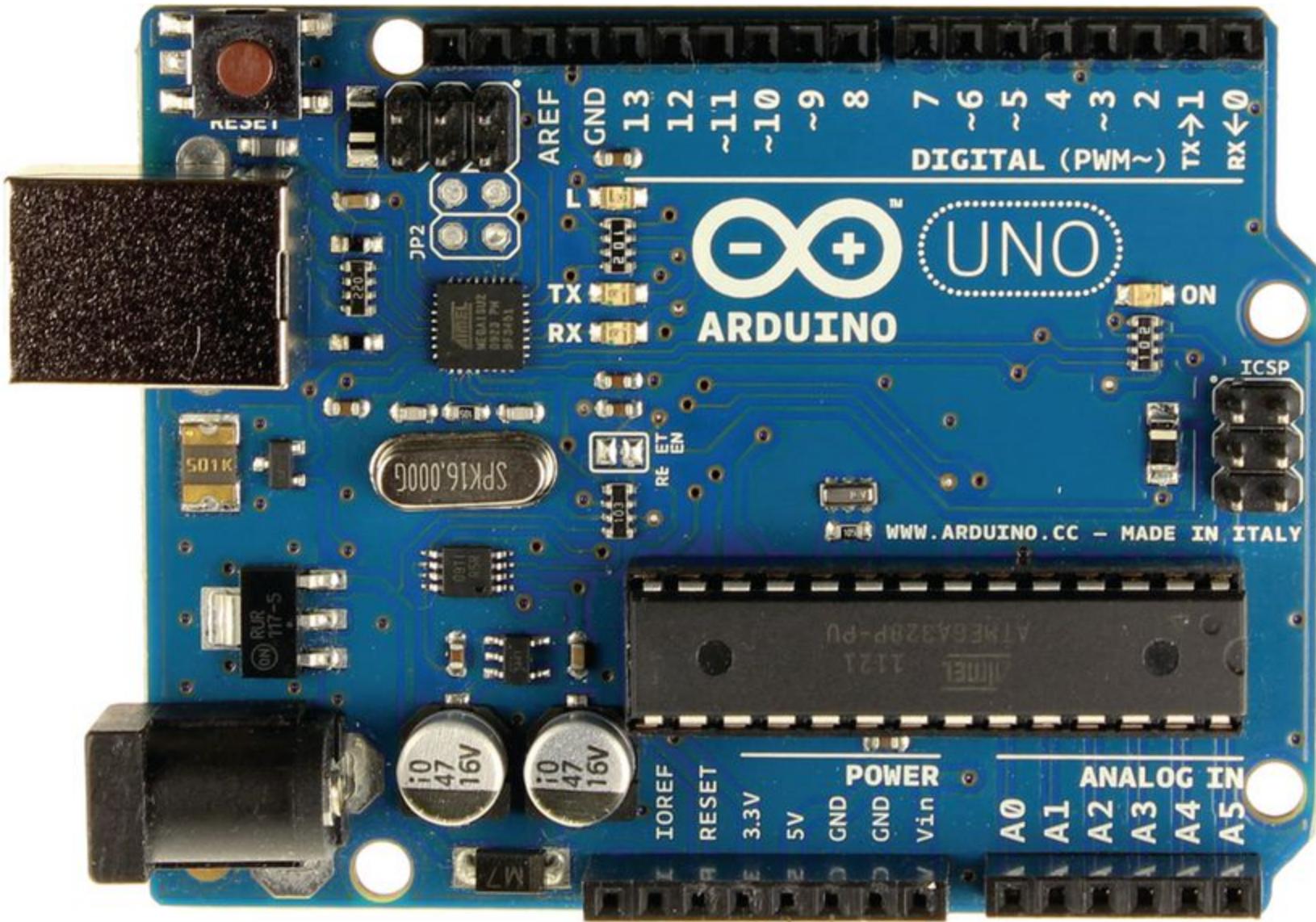


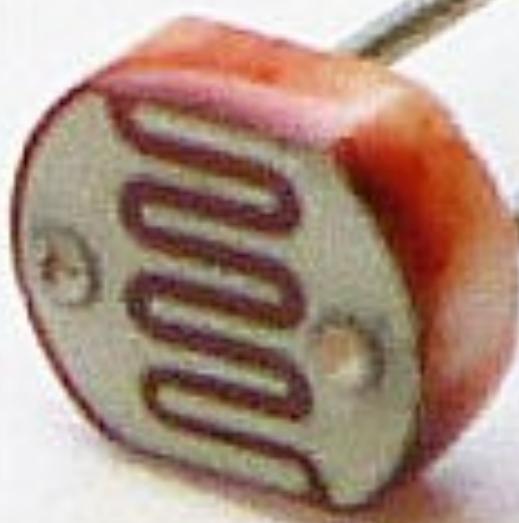
# 설리번 프로젝트

복습



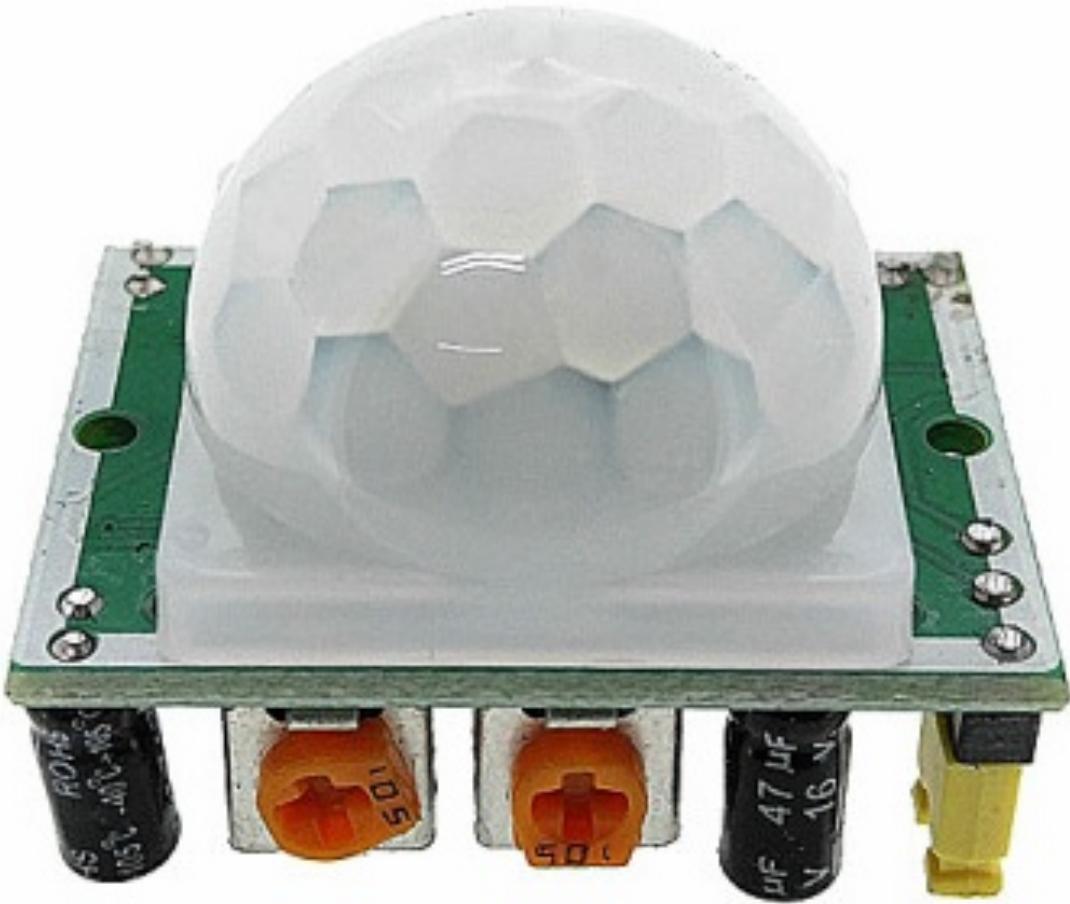
	a	b	c	d	e	f	g	h	i	j		a	b	c	d	e	f	g	h	i	j			
1	■					■					■	■				■						■	■	
2	■					■					■	■				■						■	■	
3	■					■					■	■				■						■	■	
4	■					■					■	■				■						■	■	
5	■					■					■	■				■						■	■	
6	■					■					■	■				■						■	■	
7	■					■					■	■				■						■	■	
8	■					■					■	■				■						■	■	
9	■					■					■	■				■						■	■	
10	■					■					■	■				■						■	■	
11	■					■					■	■				■						■	■	
12	■					■					■	■				■						■	■	
13	■					■					■	■				■						■	■	
14	■					■					■	■				■						■	■	
15	■					■					■	■				■						■	■	
16	■					■					■	■				■						■	■	
17	■					■					■	■				■						■	■	
18	■					■					■	■				■						■	■	
19	■					■					■	■				■						■	■	
20	■					■					■	■				■						■	■	
21	■					■					■	■				■						■	■	
22	■					■					■	■				■						■	■	
23	■					■					■	■				■						■	■	
24	■					■					■	■				■						■	■	
25	■					■					■	■				■						■	■	
26	■					■					■	■				■						■	■	
27	■					■					■	■				■						■	■	
28	■					■					■	■				■						■	■	
29	■					■					■	■				■						■	■	
30	■					■					■	■				■						■	■	



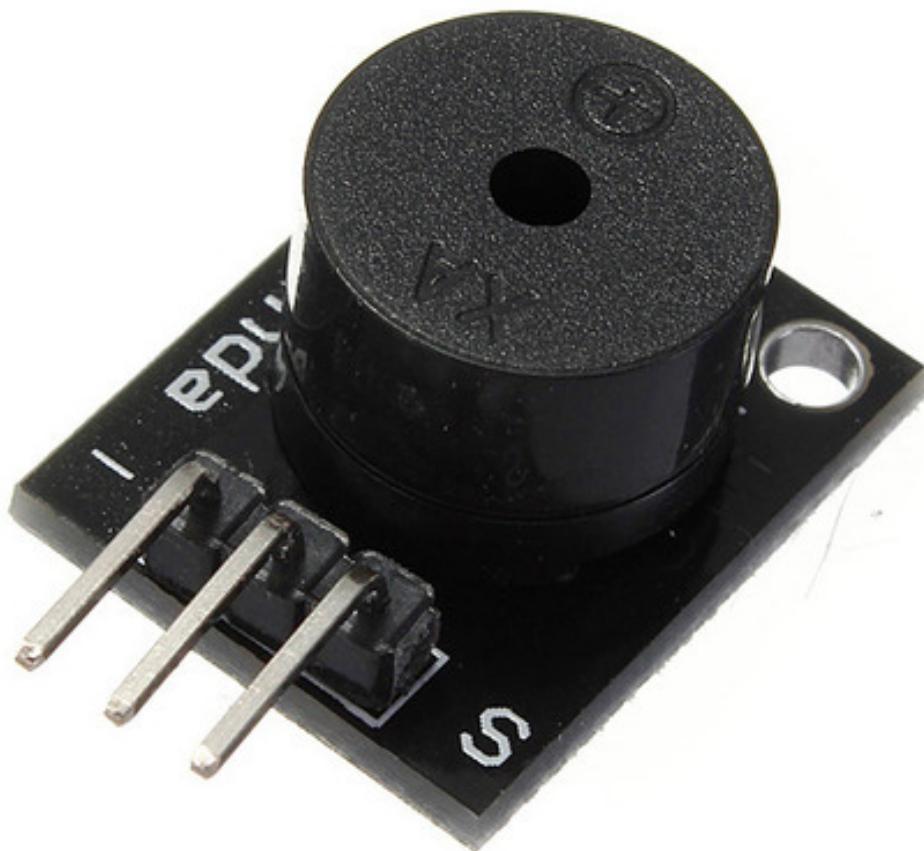


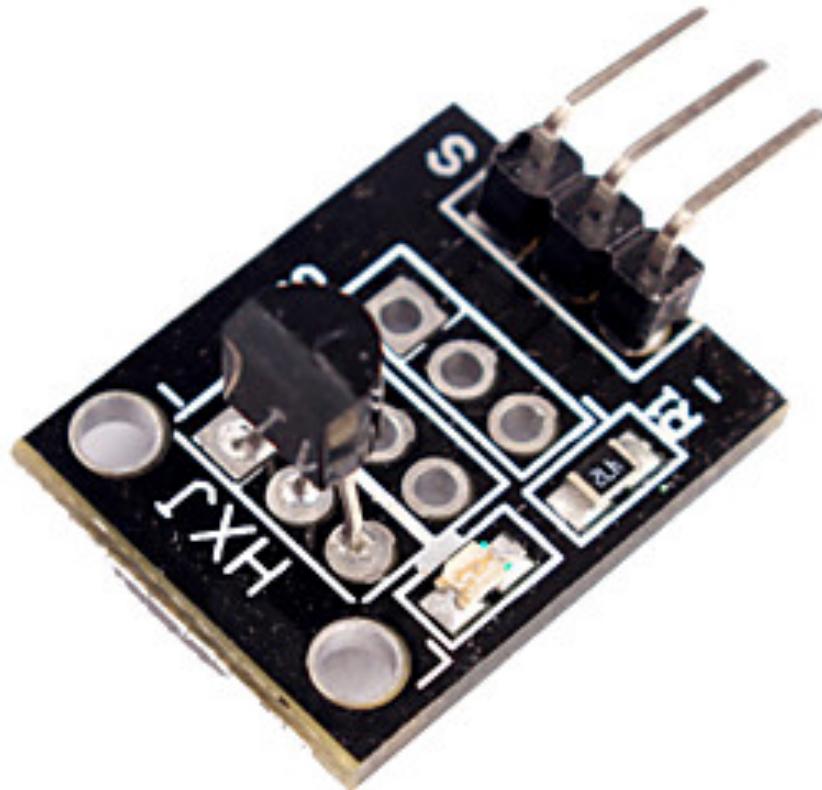




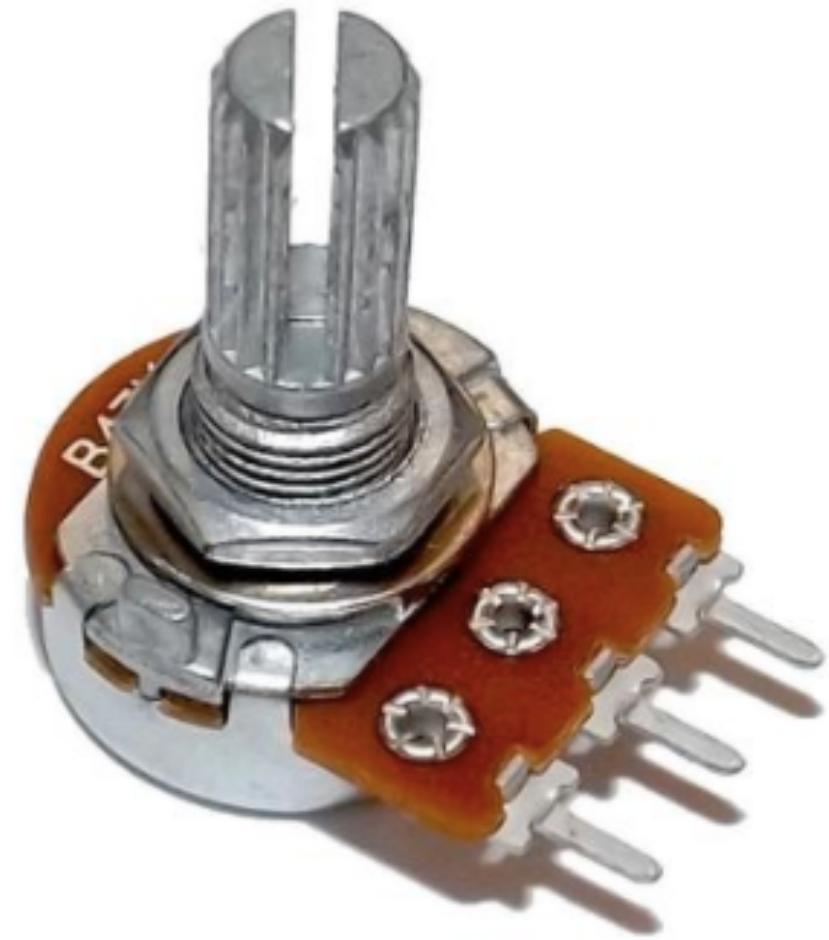














# Setup

처음 설정할때 사용

# Loop

여러번 반복 할 때

# Delay

대기

```
PinMode(13,OUTPUT);
```

# 아두이노 핀 설정

DigitalWrite(13,HIGH);

신호 보내기

오늘은?

LED를 켜보자

2인 1조

**실습을 해 볼시다!**

<미션>

빨강, 노랑, 초록 LED를  
사용해서 신호등을 만들어 봐요  
힌트 : digitalWrite, delay

스위치를 써볼까?

**실습을 해 볼시다!**

# Serial?

**실습을 해 볼시다!**

Digital? Analog?

**실습을 해 볼시다!**



설리번 프로젝트

# 나만의 **무드등** 만들기

3차시 : 통신과 제어

# 복습

```
void setup() {  
    pinMode(13, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

# 복습

void setup()

# 복습

```
void setup() {  
    pinMode(13, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

# 복습

```
pinMode(13, OUTPUT);
```

# 복습

```
void setup() {  
    pinMode(13, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

# 복습

void loop()

# 복습

```
void setup() {  
    pinMode(13, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

# 복습

```
digitalWrite(13, HIGH);
```

# 복습

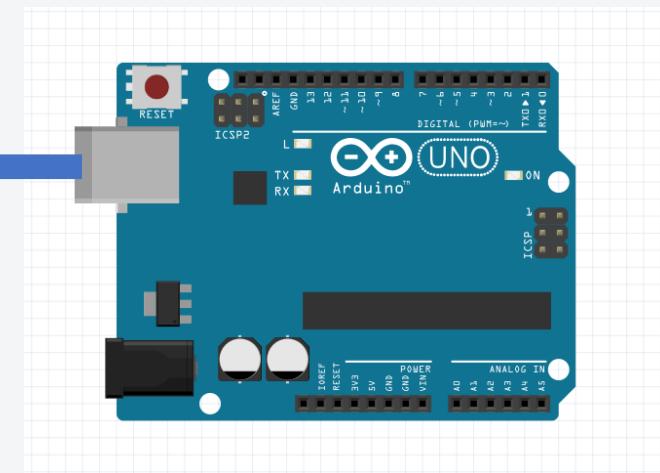
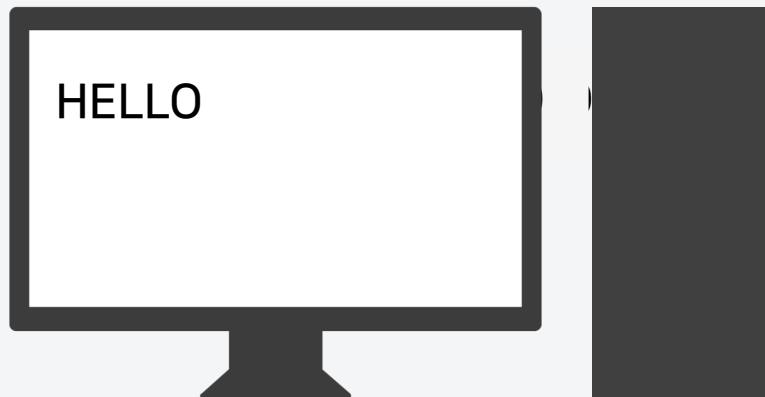
```
void setup() {  
    pinMode(13, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

# 복습

delay(1000);

# 1. 시리얼 통신



“아두이노와 컴퓨터가 소통하기 위한 방법!”

# 1. 시리얼 통신

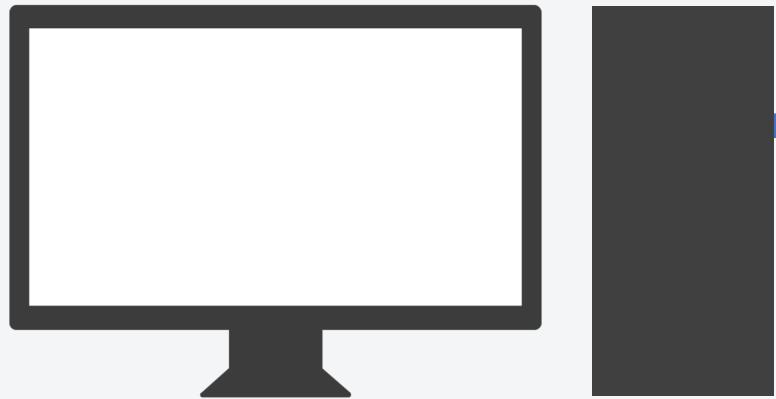
Serial.0000000

# 1. 시리얼 통신

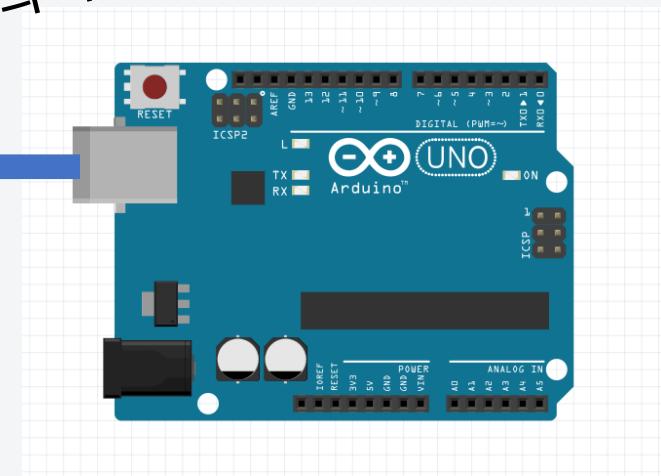
```
Serial.begin(9600);
```

시리얼 통신을 '9600'의 속도로 시작한다

# 1. 시리얼 통신



9600의 속도로 시작!

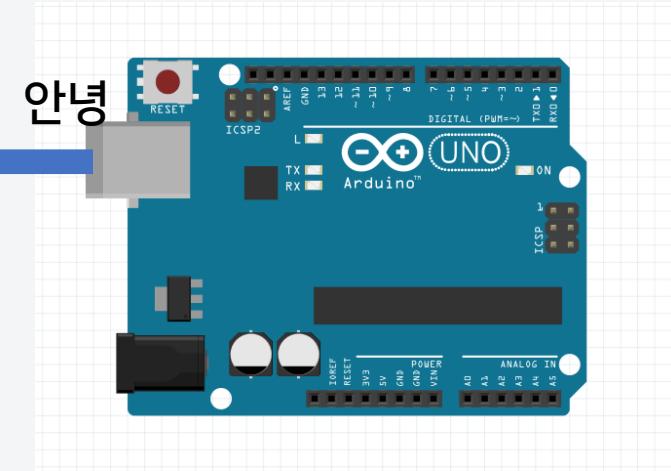


# 1. 시리얼 통신

```
Serial.print("안녕");
```

시리얼 통신으로 “안녕” 이라고 전송한다

# 1. 시리얼 통신

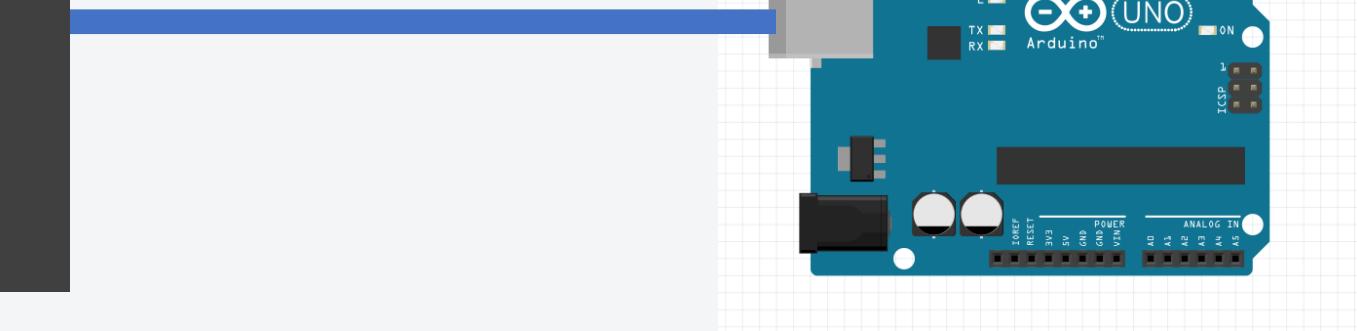


# 1. 시리얼 통신

Serial.Read();

시리얼 통신에서 받아온 값

# 1. 시리얼 통신



# 1. 시리얼 통신\_예제

아두이노에서 컴퓨터로 “안녕”이라고 보내보자!!

# 1. 시리얼 통신\_예제

아두이노에서 컴퓨터로 스위치의 상태를 보내보자!!

HINT : digitalRead로 읽고 Serial.print로 보내고~

# 1. 시리얼 통신\_예제

컴퓨터에서 보낸 시리얼 통신을 읽어보자  
그리고 만약 “1” 이라고 왔으면 led를 켜보자

HINT : Serial.Read로 컴퓨터 말을 읽어보자

# 엥?

읽는 거 까진 됐는데 어떻게 led를 그때만 켜지?

## 2. If문

특정한 상황에서만  
작동하는 코드를 쓰고 싶어요

## 2. If문

```
만약에 (배부르다면){  
    똥을 싸라!!!  
}
```

## 2. If문

```
if (조건) {  
    수행할 일;  
}
```

## 2. If문

```
if (컴퓨터가 말한게 10이다) {  
    led를 켜라;  
}
```

## 2. If문

```
if ( Serial.Read() ==1 ) {  
    digitalWrite(13,HIGH);  
}
```

## 2. If문\_예제

스위치가 켜졌을 때만 led를 켜보자

HINT : Serial.Read로 컴퓨터 말을 읽어보자

## 2. If문\_예제

내 무드등은  
“00할때 00한다”  
를 유사코드로 적어보자

HINT : 다양한 센서를 기억해보고 창의적으로 작성해보자

### 3. 반복문

같은 일을 반복해서 쓰기 귀찮아요..

### 3. 반복문

노가다

```
Serial.print("으악");  
Serial.print("으악");  
Serial.print("으악");  
Serial.print("으악");  
Serial.print("으악");  
Serial.print("으악");  
Serial.print("으악");  
Serial.print("으악");  
Serial.print("으악");  
Serial.print("으악");
```

반복문

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

### 3. 반복문

```
for (초기화식; 조건식 ; 변화식){  
    반복내용  
}
```

### 3. 반복문

```
for (int i=0; i< 100 ; i++){  
    반복내용  
}
```

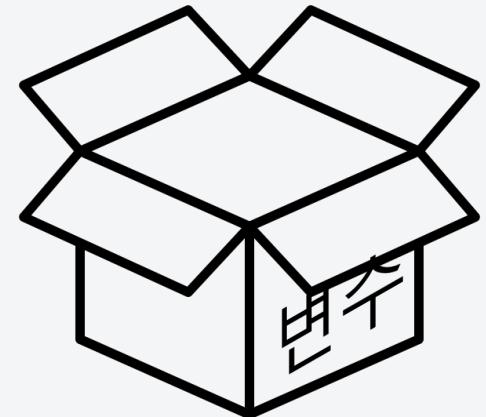
## 3.1 변수

변하는 수

# 3.1 변수

## 변수

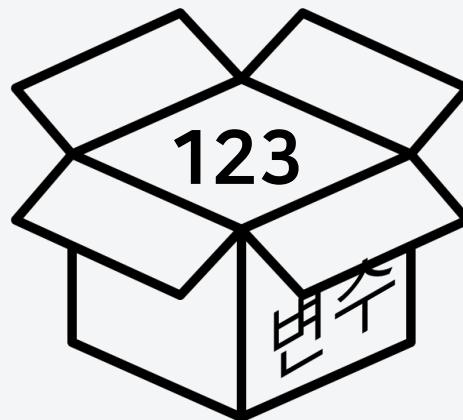
값을 저장해 놓을 수 있는 변하는 상자



# 3.1 변수



소수를 담는 상자



정수를 담는 상자



문자를 담는 상자

# 3.1 변수 선언

자료형 변수이름;

**int a;** a라는 정수 상자 생성

**float b;** b라는 소수 상자 생성

**char c** c라는 문자 상자 생성

## 3.1 변수 대입

변수이름 = 넣고싶은거;

a = 23;                      a상자에 23 넣기

b = 4.6;                      b상자에 4.6 넣기

c = 'd';                      c상자에 d넣기

## 3.1 변수 사용

변수이름

-> 상자안에 있는 값이 나옴

```
if (a == 23 ){  
    led켜라  
}
```

# 3.1 변수 test

```
int a;  
a=3;  
a=5;  
a=2;  
if(a==5){  
    Led켜기  
}
```

Led는 켜질까요?

## 3.2 연산

+	더하기	>	크다
-	빼기	<	작다
*	곱하기	>=	크거나 같다
/	나누기	<=	작거나 같다
%	나머지	==	같다
++	1더하기	=	대입

## 3.2 연산

+	더하기	>	크다
-	빼기	<	작다
*	곱하기	>=	크거나 같다
/	나누기	<=	작거나 같다
%	나머지	==	같다
++	1더하기	=	대입

Int a=12;

Int b= 4;

$$a + b == 16$$

$$a - b == 8$$

$$a / b == 3$$

$$a * b == 36$$

$$a++ == 13$$

$$a \% b == 1$$

## 3.2 연산

+	더하기	>	크다
-	빼기	<	작다
*	곱하기	>=	크거나 같다
/	나누기	<=	작거나 같다
%	나머지	==	같다
++	1더하기	=	대입

Int a=12;

Int b= 4;

a > b (

a < b (

a==b (

### 3. 반복문

변수 i 는 0으로 초기화

i 를 1증가

```
for (int i=0; i< 100 ; i++){  
    반복내용  
}
```

i 가 100보다 작을 때 까지 반복 수행

### 3. 반복문

반복문을 사용해 '으악'을 100번 출력해보자

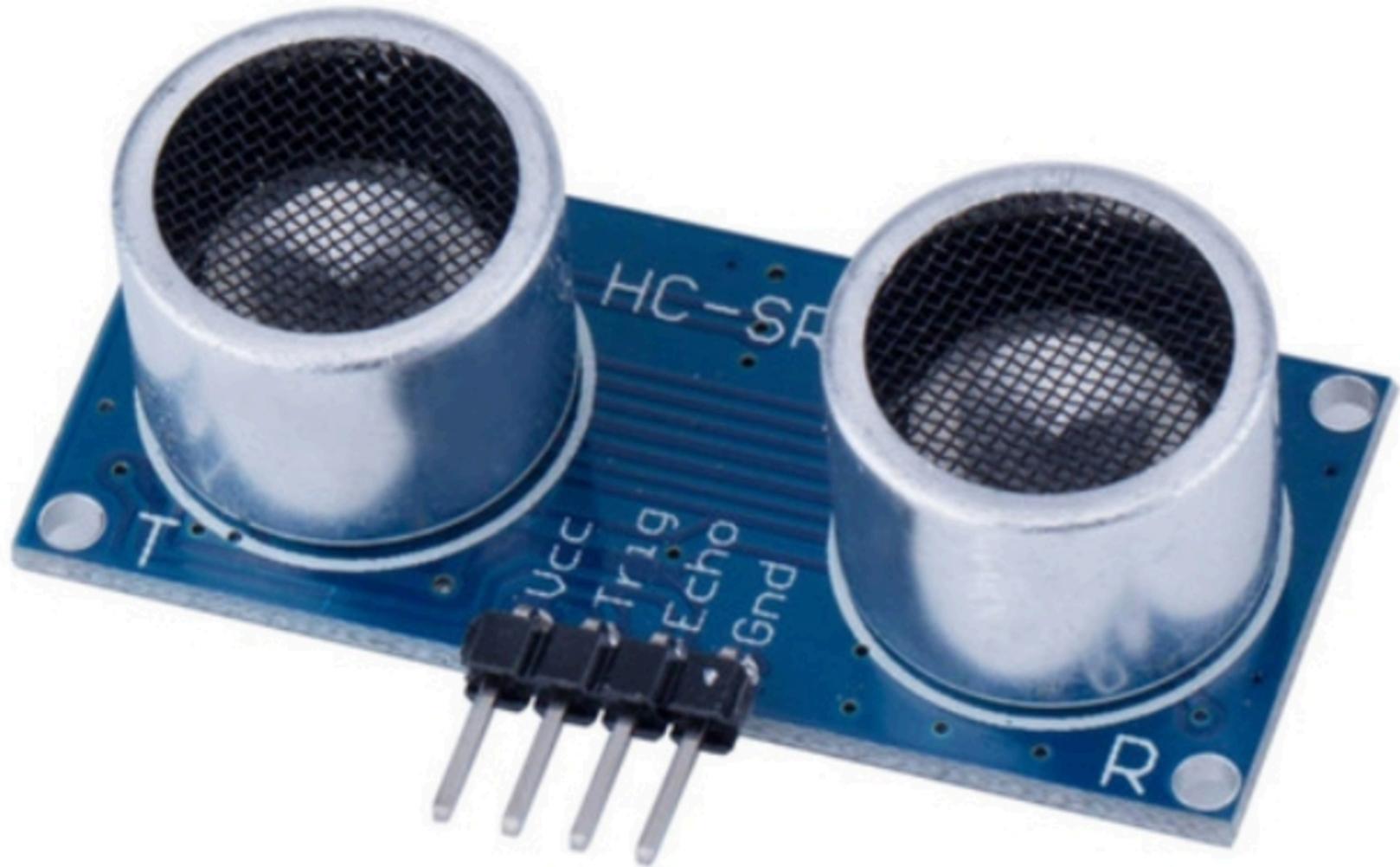
HINT : Serial.println("으악"); 을 하면 줄바꿈이 되네!

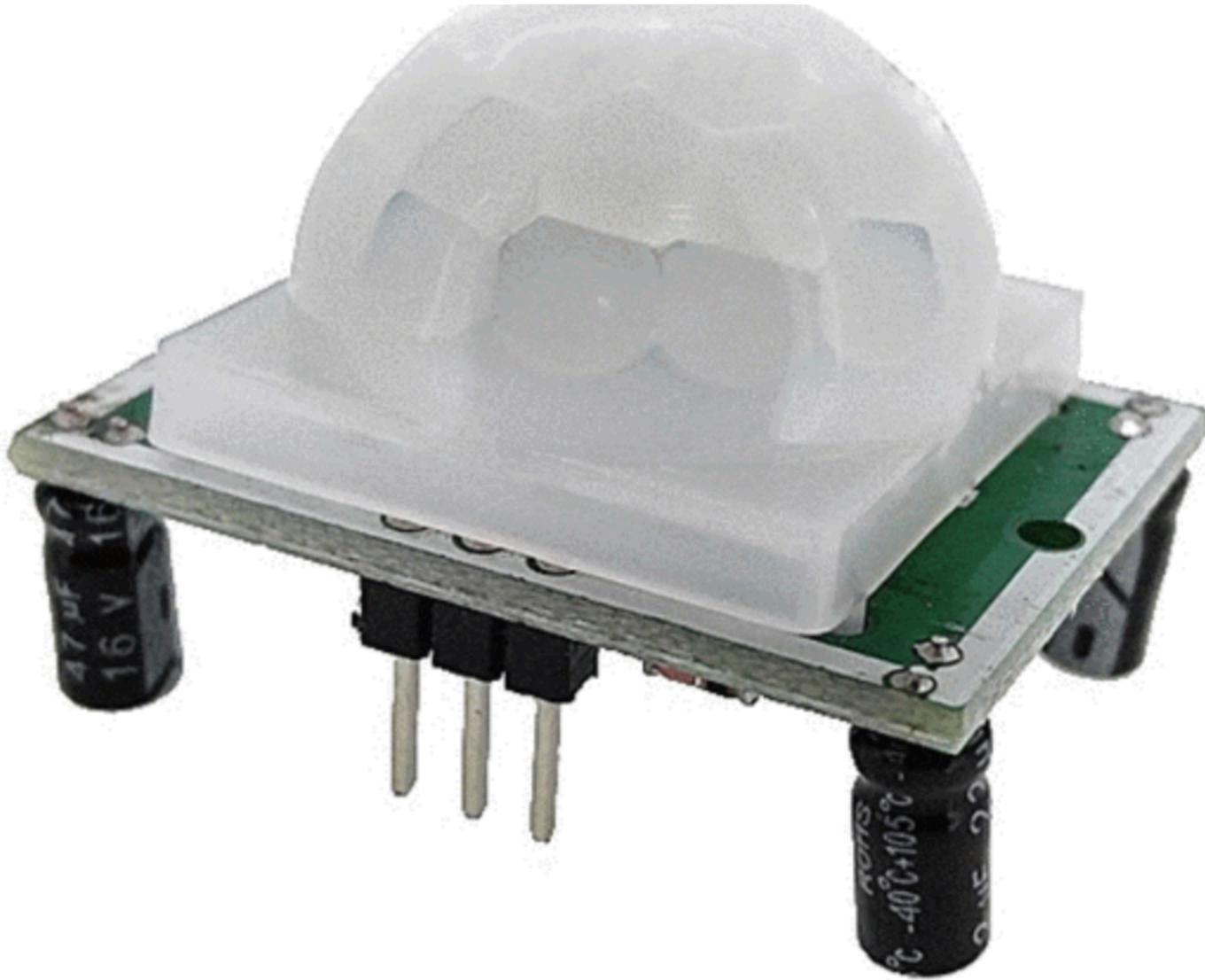
### 3. 반복문

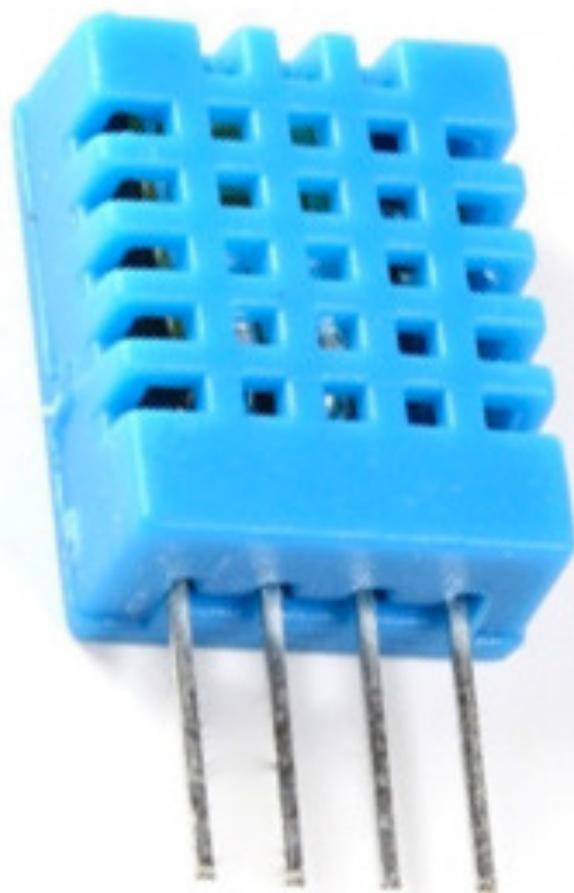
반복문을 사용해 1부터 100까지 출력해보자

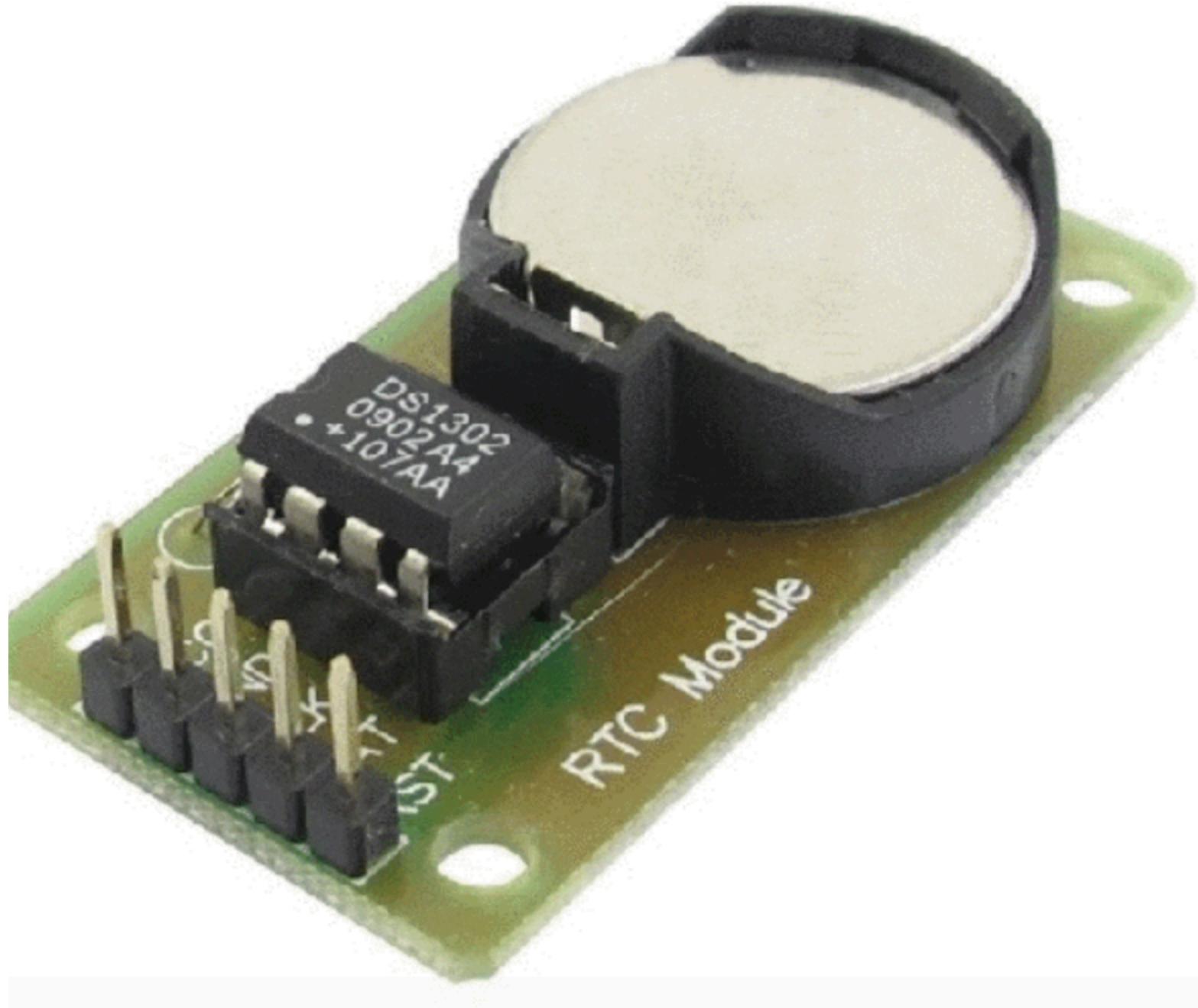
HINT : int i는 반복문이 몇 바퀴 돌았는지 알고있네!

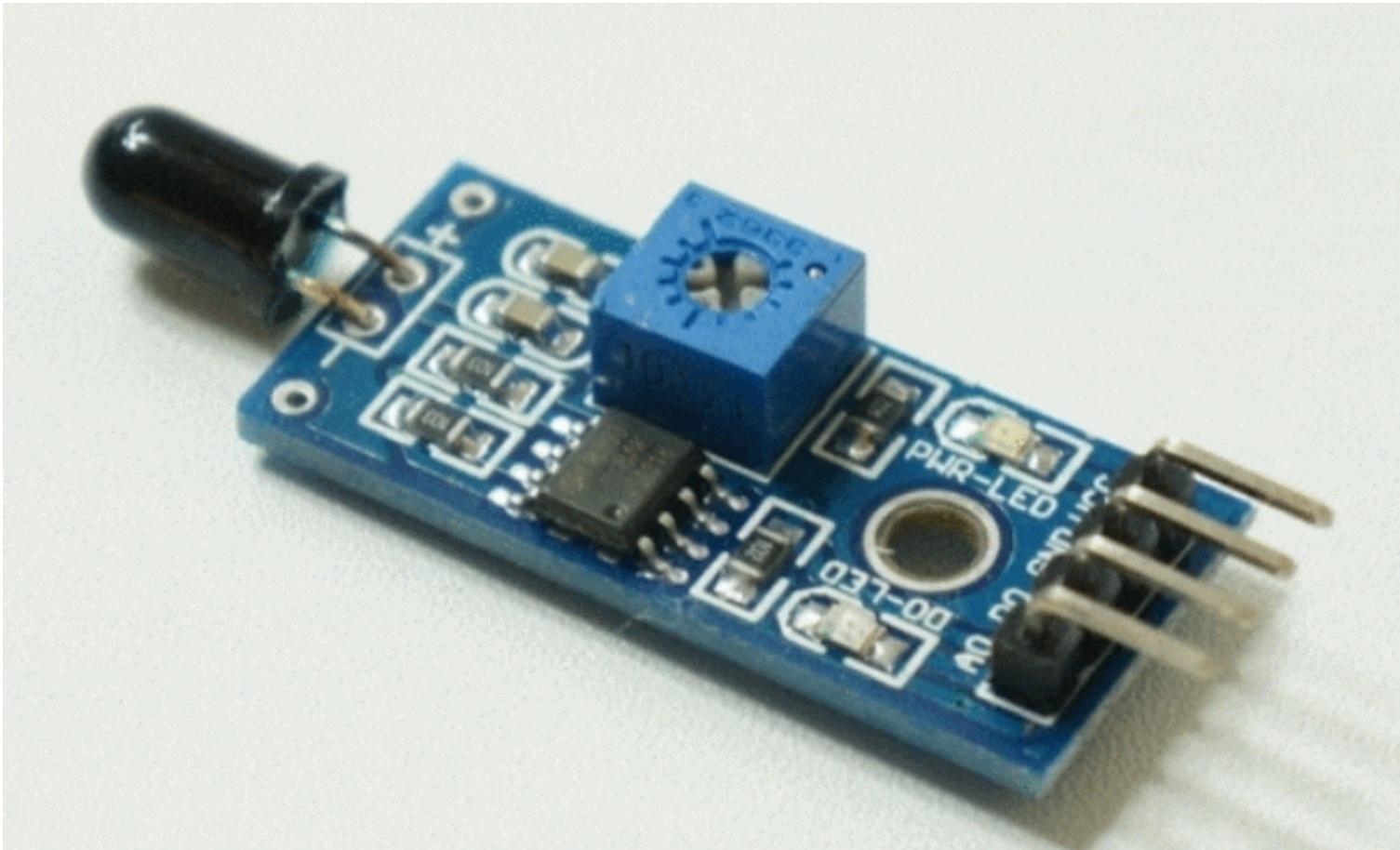


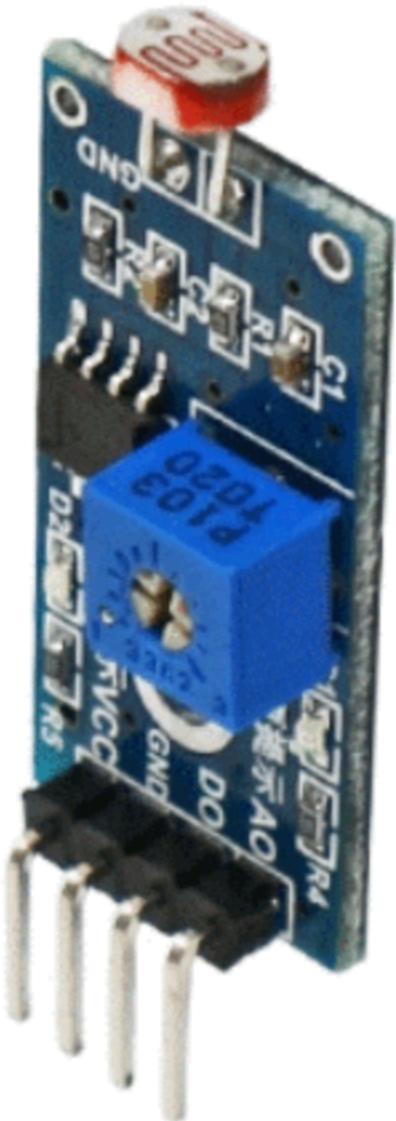


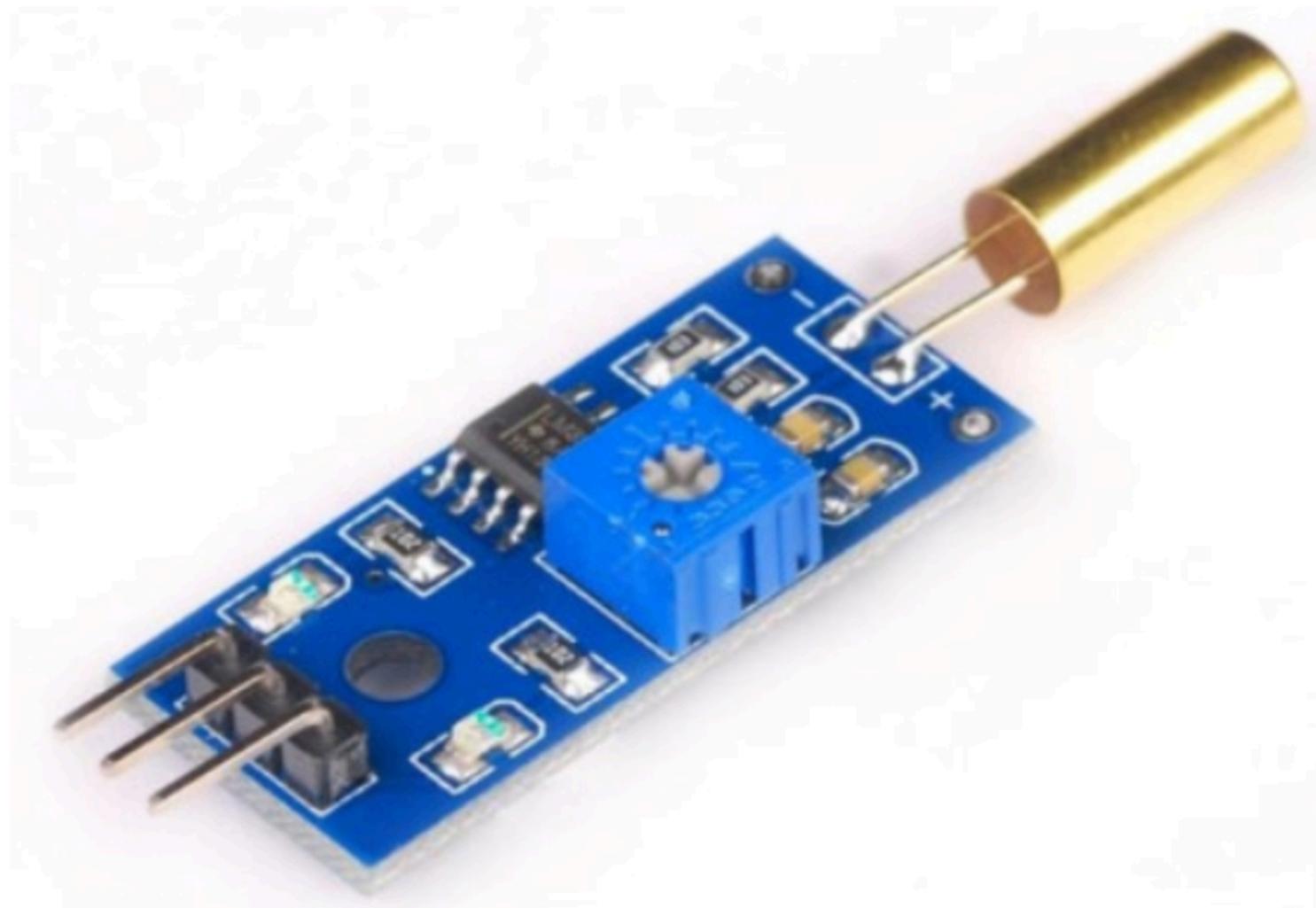


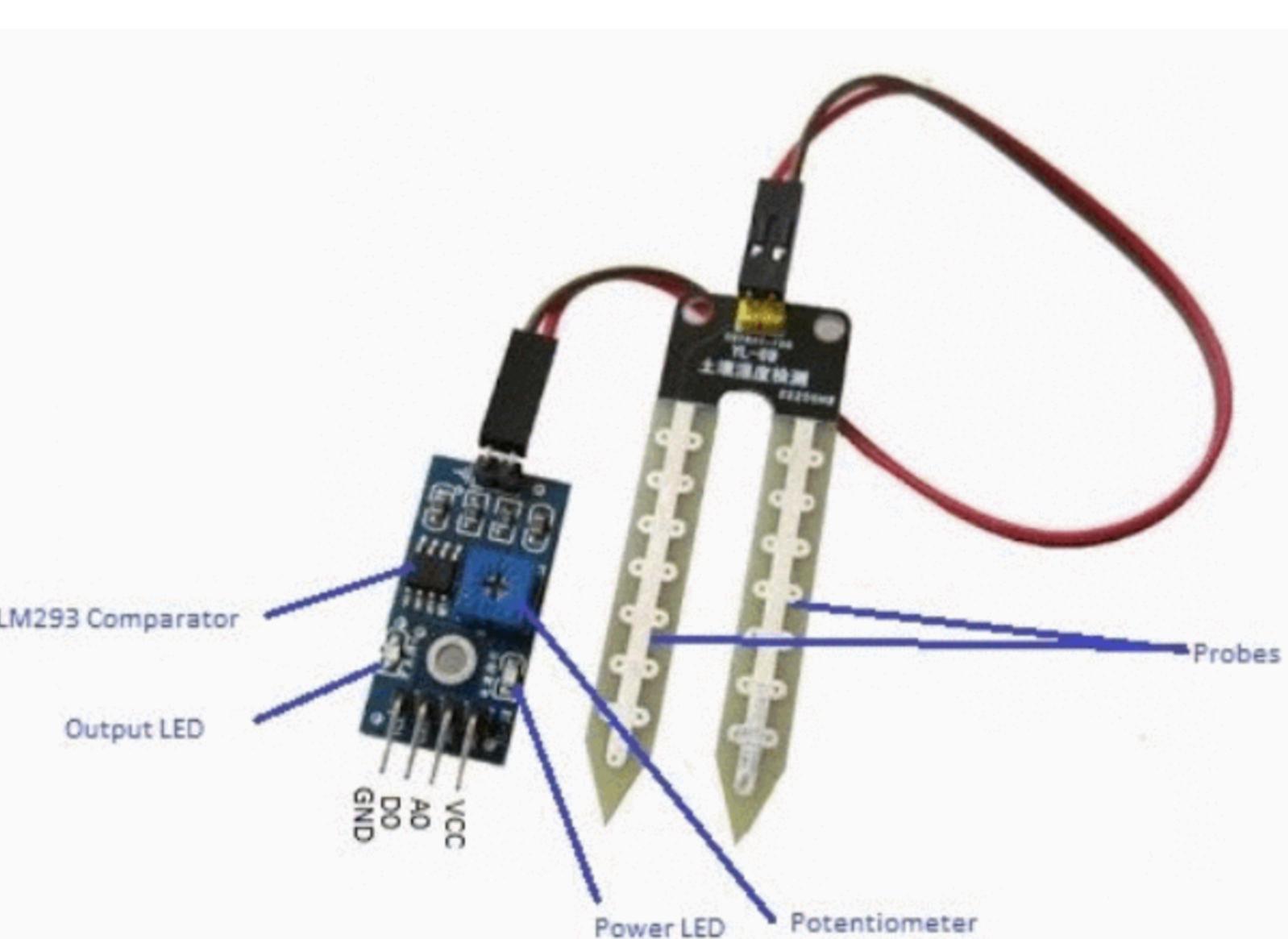


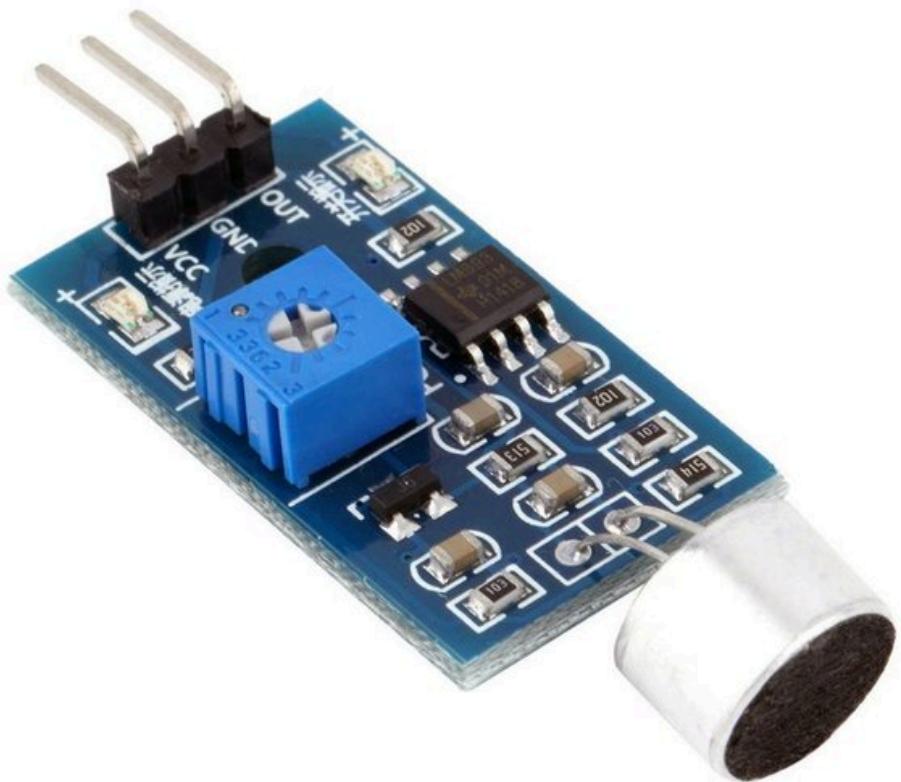






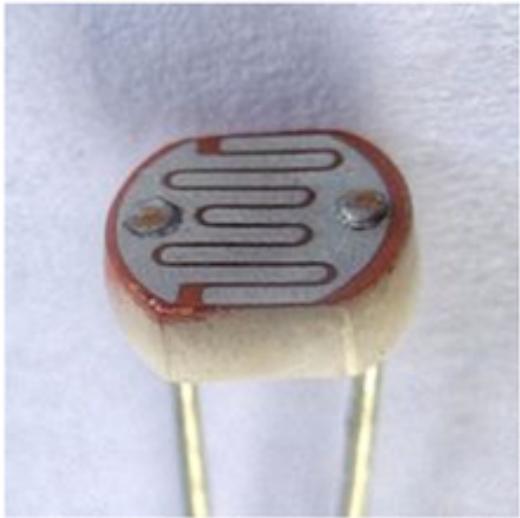




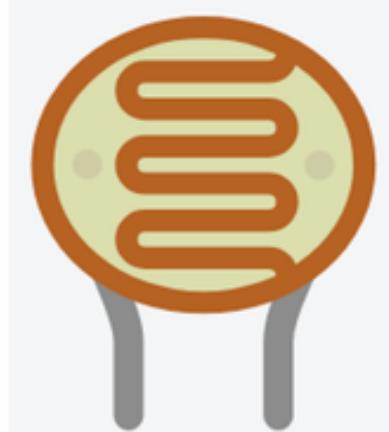


센서

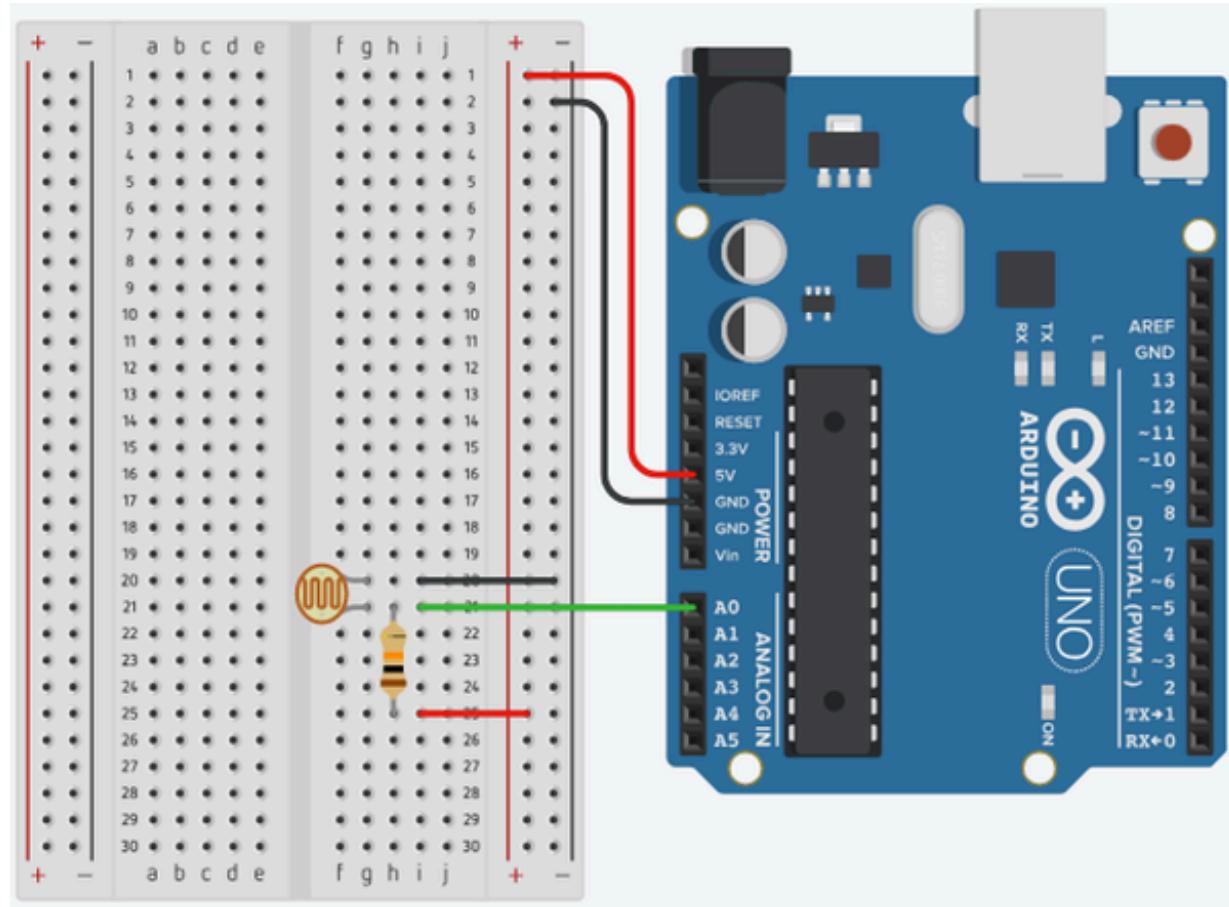




조도센서(Photoresistor)

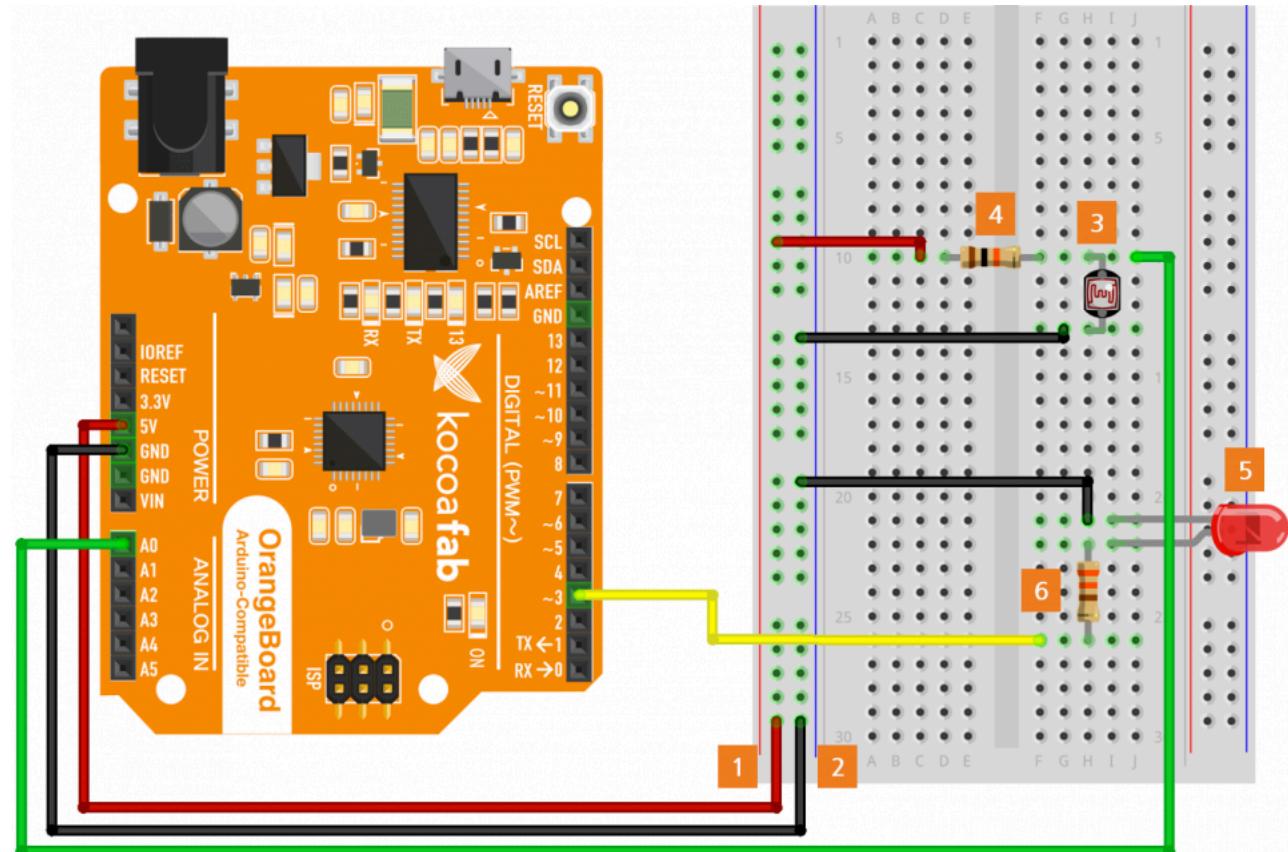


조도센서 구성요소



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

void loop()
{
    int readValue = analogRead(A0);
    Serial.println(readValue);
}
```

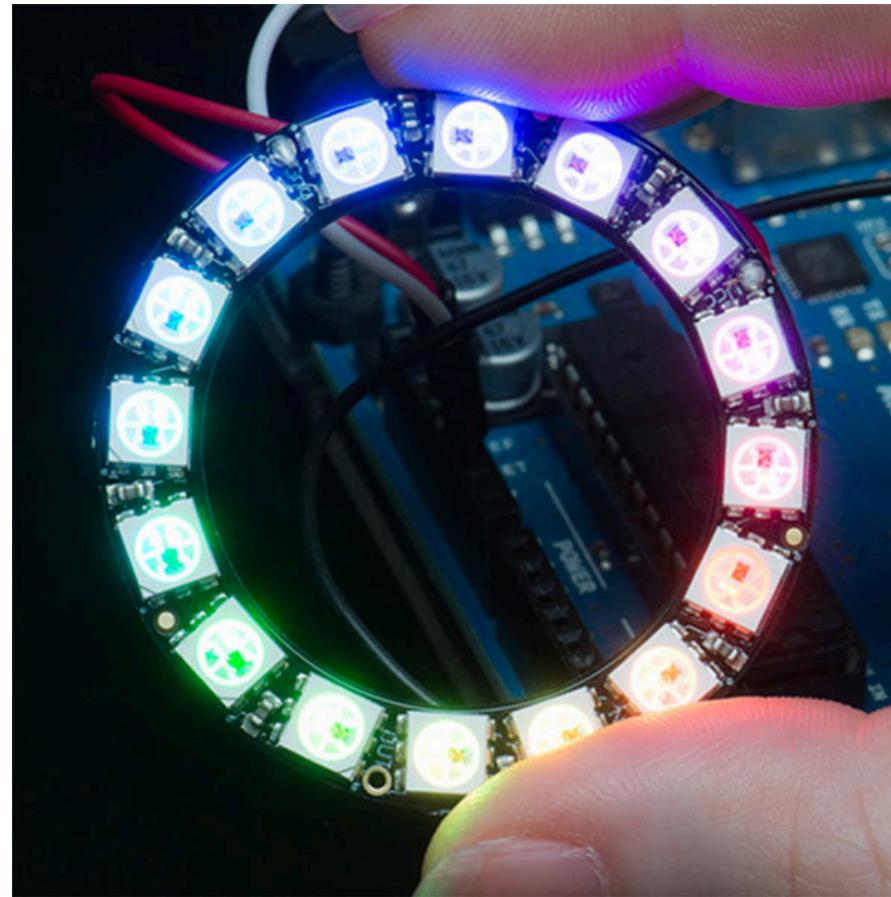


fritzing

```
int cds = A0;
int led = 3;
void setup() {
    Serial.begin(9600);
    pinMode(led, OUTPUT);
}
void loop() {
    int cdsValue = analogRead(cds);
    Serial.print("cds = ");
    Serial.println(cdsValue);

    if (cdsValue > 500) {
        digitalWrite(led, HIGH);
        Serial.println("LED ON (cds > 500)");
    }
    else {
        digitalWrite(led, LOW);
        Serial.println("LED OFF (cds < 500)");
    }
    delay(200);
}
```

네오픽셀



[https://github.com/adafruit/adafruit\\_neopixel](https://github.com/adafruit/adafruit_neopixel)

아두이노 – 스케치 – 라이브러리 불러오기

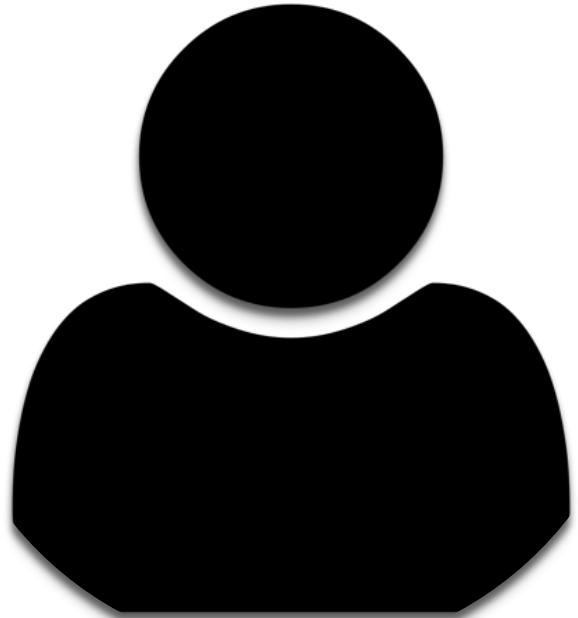
```
void loop() {  
    rainbow(20);  
    rainbowCycle(20);  
}
```



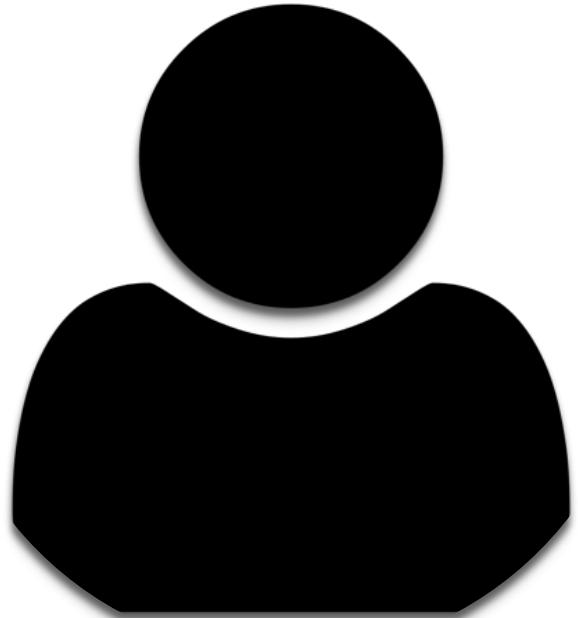


나만의 무드등을  
발표해보자!

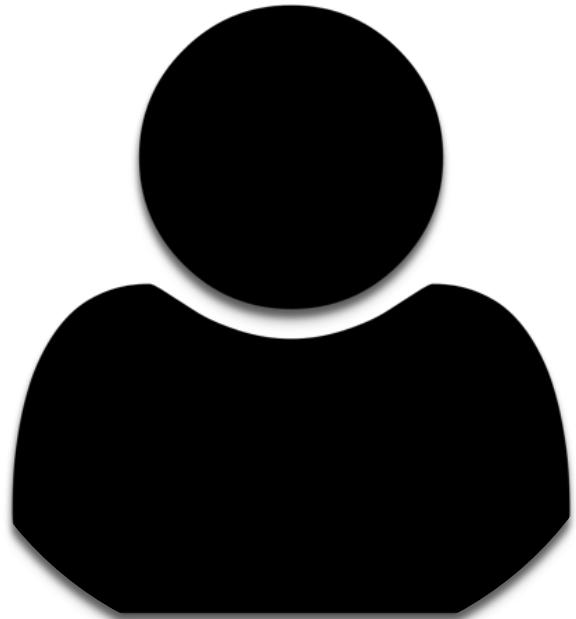
01\*호



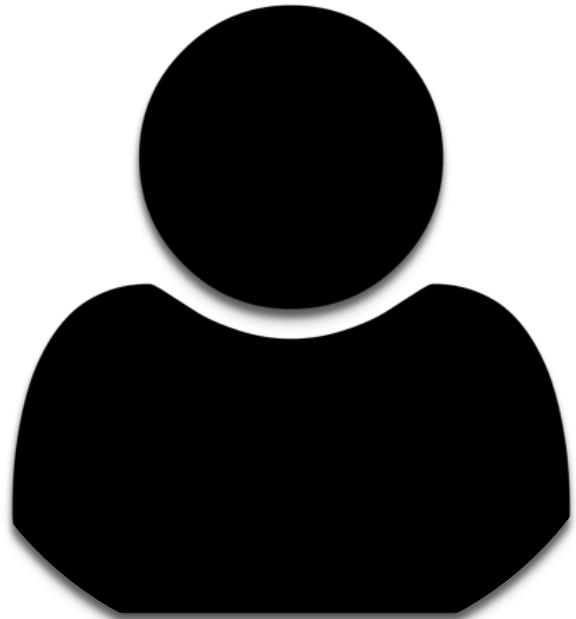
이\*우



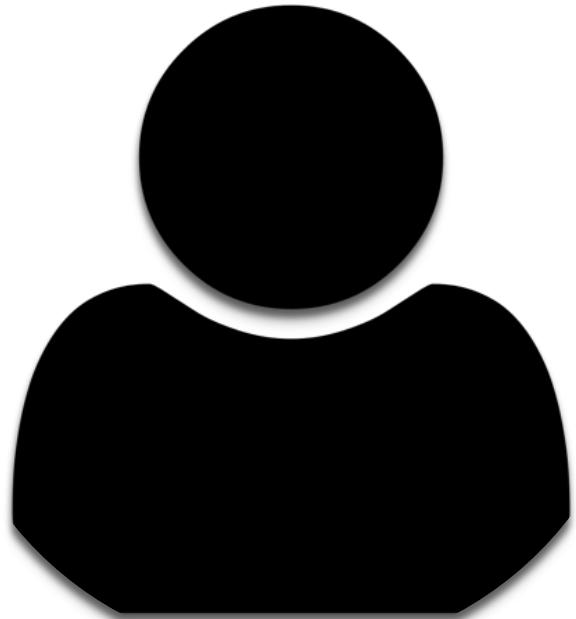
김\*현



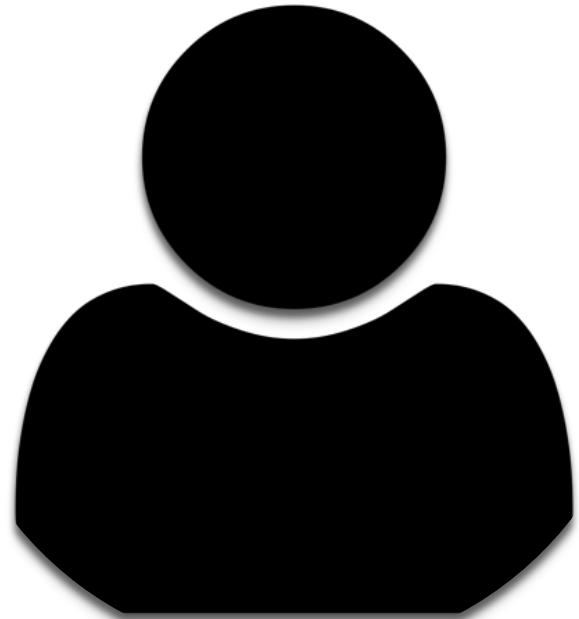
김\*율



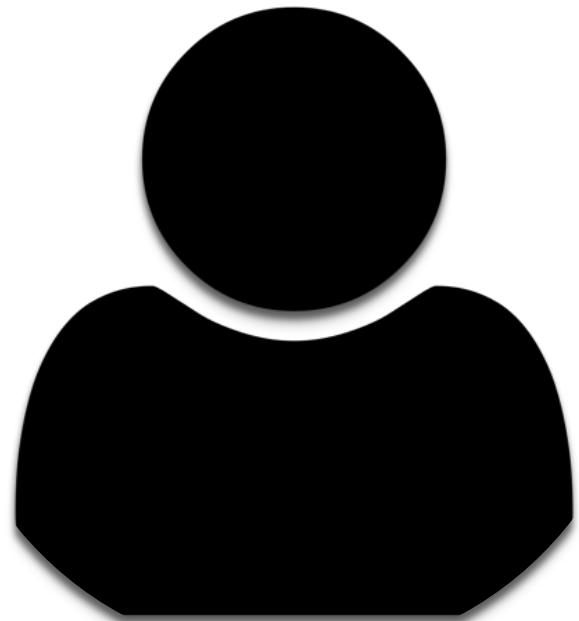
김\*우



박\*진



김\*준



**조\*재**

