



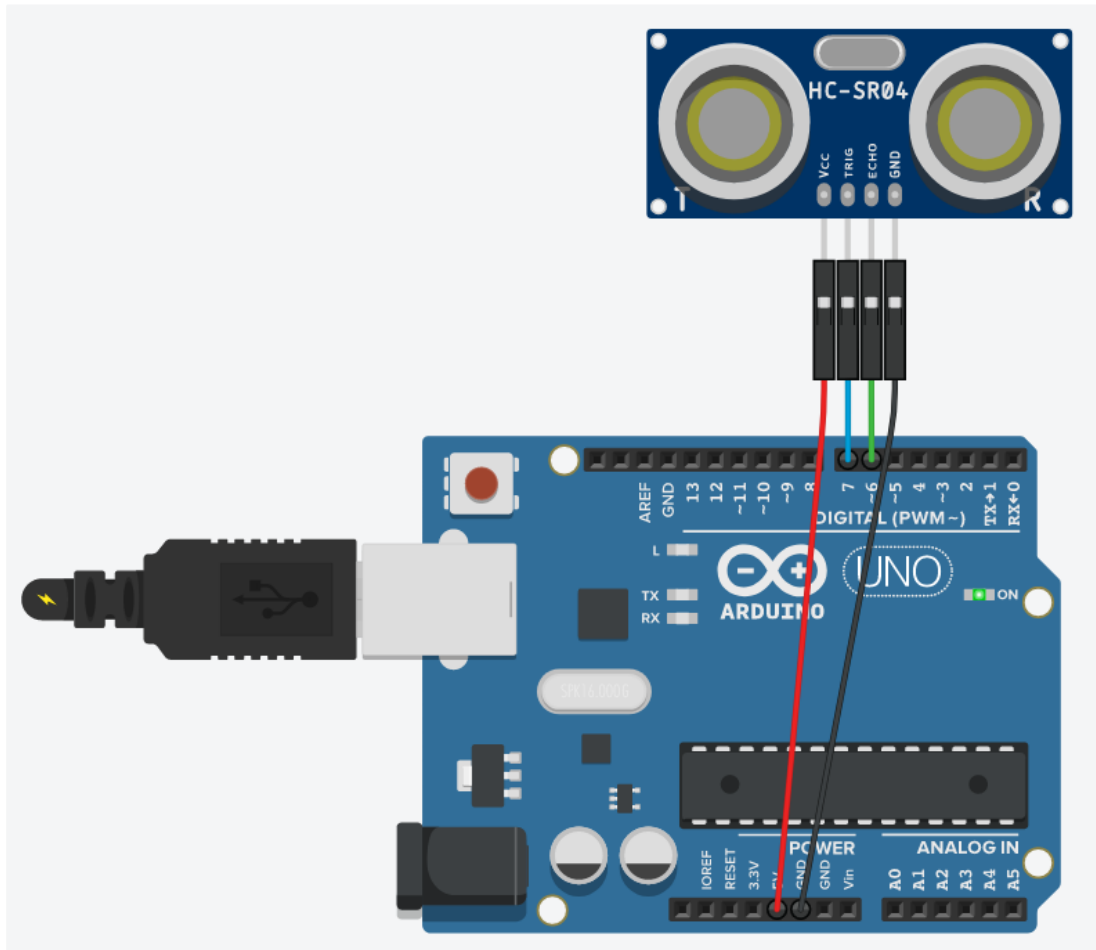
# Step 2 - Mission 1

👉 ↗ Step 2 - Mission 1 (5.10. - 6.16.)

## ▼ 2차시(가영)

- 거리 측정 센서

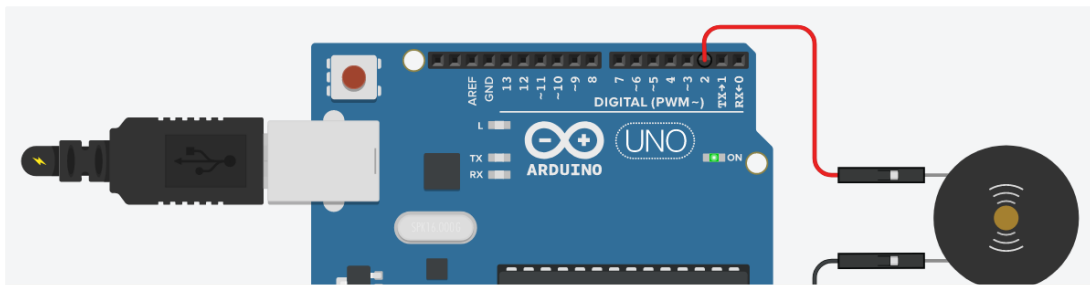
## 초음파센서



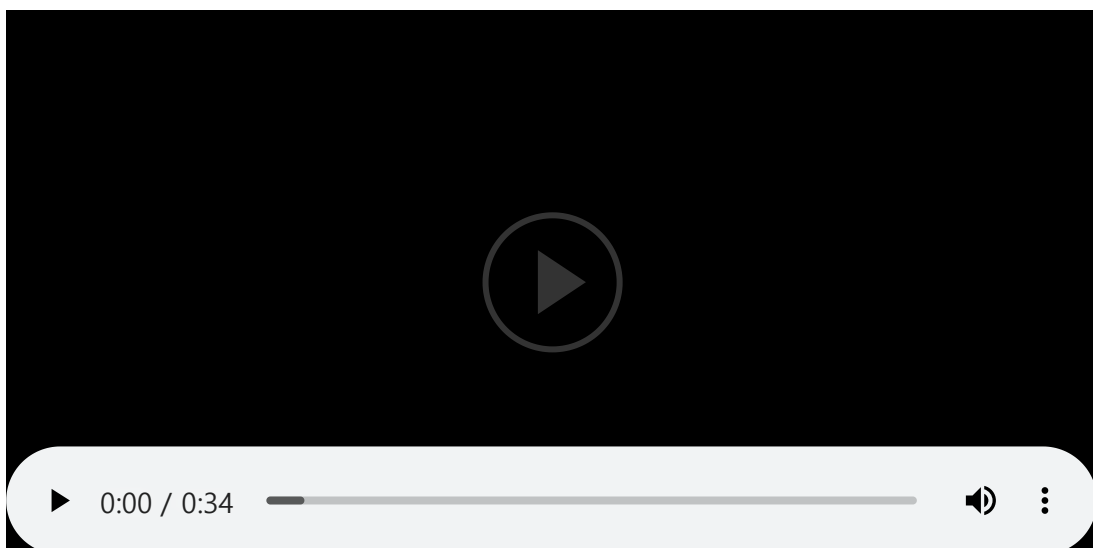
```
int TrigPin = 7; int EchoPin = 6; void setup() {
  Serial.begin(9600); pinMode(TrigPin, OUTPUT); pinMode(EchoPin,
  INPUT); } void loop() { digitalWrite(TrigPin, HIGH);
  delayMicroseconds(10); digitalWrite(TrigPin, LOW); float duration
  = pulseIn(EchoPin, HIGH); float distance = 340*duration / 10000 /
  2; Serial.print("distance : "); Serial.println(distance); }
```

- 소리 출력

## 피에조부저

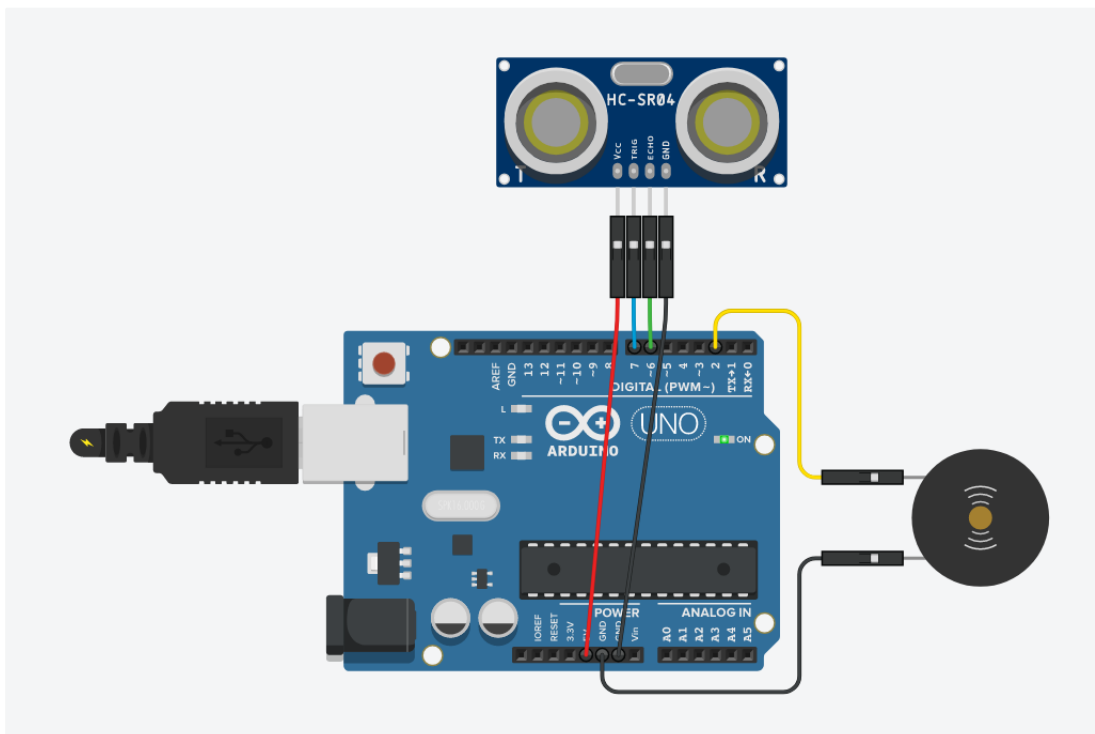


```
#define NOTE_AS4 466 #define NOTE_B4 494 #define NOTE_DS5 622
#define NOTE_E5 659 #define NOTE_FS5 740 #define NOTE_G5 784
#define NOTE_GS5 831 #define NOTE_A5 880 #define NOTE_AS5 932
#define NOTE_B5 988 #define NOTE_C6 1047 #define NOTE_CS6 1109
#define NOTE_D6 1175 int HarryPotter_melody[] = { NOTE_B4,
NOTE_E5, NOTE_G5, NOTE_FS5, NOTE_E5, NOTE_B5, NOTE_A5, NOTE_FS5,
NOTE_E5, NOTE_G5, NOTE_FS5, NOTE_DS5, NOTE_FS5, NOTE_B4, NOTE_E5,
NOTE_G5, NOTE_FS5, NOTE_E5, NOTE_B5, NOTE_D6, NOTE_CS6, NOTE_C6,
NOTE_GS5, NOTE_C6, NOTE_B5, NOTE_AS5, NOTE_AS4, NOTE_G5, NOTE_E5,
NOTE_G5, NOTE_B5, NOTE_G5, NOTE_B5, NOTE_G5, NOTE_C6, NOTE_B5,
NOTE_AS5, NOTE_FS5, NOTE_G5, NOTE_B5, NOTE_AS5, NOTE_AS4, NOTE_B4,
NOTE_B5, NOTE_G5, NOTE_B5, NOTE_G5, NOTE_B5, NOTE_G5, NOTE_D6,
NOTE_CS6, NOTE_C6, NOTE_GS5, NOTE_C6, NOTE_B5, NOTE_AS5, NOTE_AS4,
NOTE_G5, NOTE_E5 }; float HarryPotter_durations[]={ 4, 2.7, 8, 4,
2, 4, 1.3, 1.3, 2.7, 8, 4, 2, 4, 1.3, 2.7, 8, 4, 2, 4, 2, 4, 2, 4,
2, 4, 4, 2, 4, 1.3, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2.7, 8, 4, 2, 4,
1.3, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2.7, 8, 4, 2, 4, 1.3 }; int
BuzzerPin = 2; void setup() { pinMode(BuzzerPin, OUTPUT); } void
loop() { int notes =
sizeof(HarryPotter_durations)/sizeof(HarryPotter_durations[0]);
for (int i = 0; i < notes; i++){ int durations = 1000 /
HarryPotter_durations[i]; tone(BuzzerPin, HarryPotter_melody[i],
durations); delay(durations*1.5); } noTone(BuzzerPin); }
```



```
#define c4 262 #define d4 294 #define e4 330 #define f4 349
#define g4 392 #define a5 440 #define b5 494 #define c5 523 int
BuzzerPin = 2; void setup() { pinMode(BuzzerPin, OUTPUT); } void
loop() { tone(BuzzerPin, c4); delay(500); tone(BuzzerPin, d4);
delay(500); tone(BuzzerPin, e4); delay(500); tone(BuzzerPin, f4);
delay(500); tone(BuzzerPin, g4); delay(500); tone(BuzzerPin, a5);
delay(500); tone(BuzzerPin, b5); delay(500); tone(BuzzerPin, c5);
delay(500); noTone(BuzzerPin); }
```

- 최종 실습



```
int TrigPin = 7; int EchoPin = 6; int BuzzerPin = 2; void setup()
{ Serial.begin(9600); pinMode(TrigPin, OUTPUT); pinMode(EchoPin,
INPUT); pinMode(BuzzerPin, OUTPUT); } void loop() {
digitalWrite(TrigPin, HIGH); delayMicroseconds(10);
digitalWrite(TrigPin, LOW); float duration = pulseIn(EchoPin,
HIGH); float distance = 340*duration / 10000 / 2;
Serial.println(distance); if(distance >= 100 && distance < 200){
tone(BuzzerPin, 392, 300); delay(1200); }else if(distance >= 50 &&
distance < 100){ tone(BuzzerPin, 392, 300); delay(800); }else
if(distance < 50){ tone(BuzzerPin, 392, 300); delay(500); }else
noTone(BuzzerPin); }
```

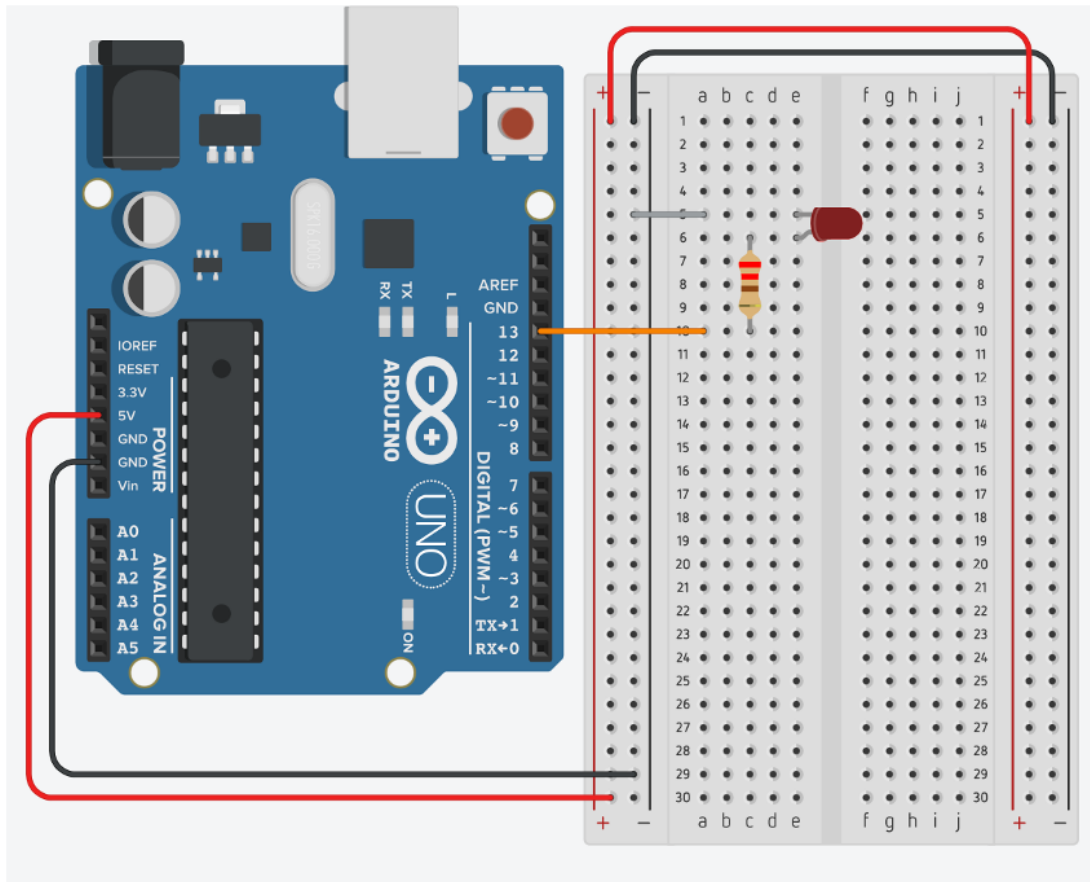


## ▼ 3차시(용환)

- LED 켜고 끄기
  - Built-in LED

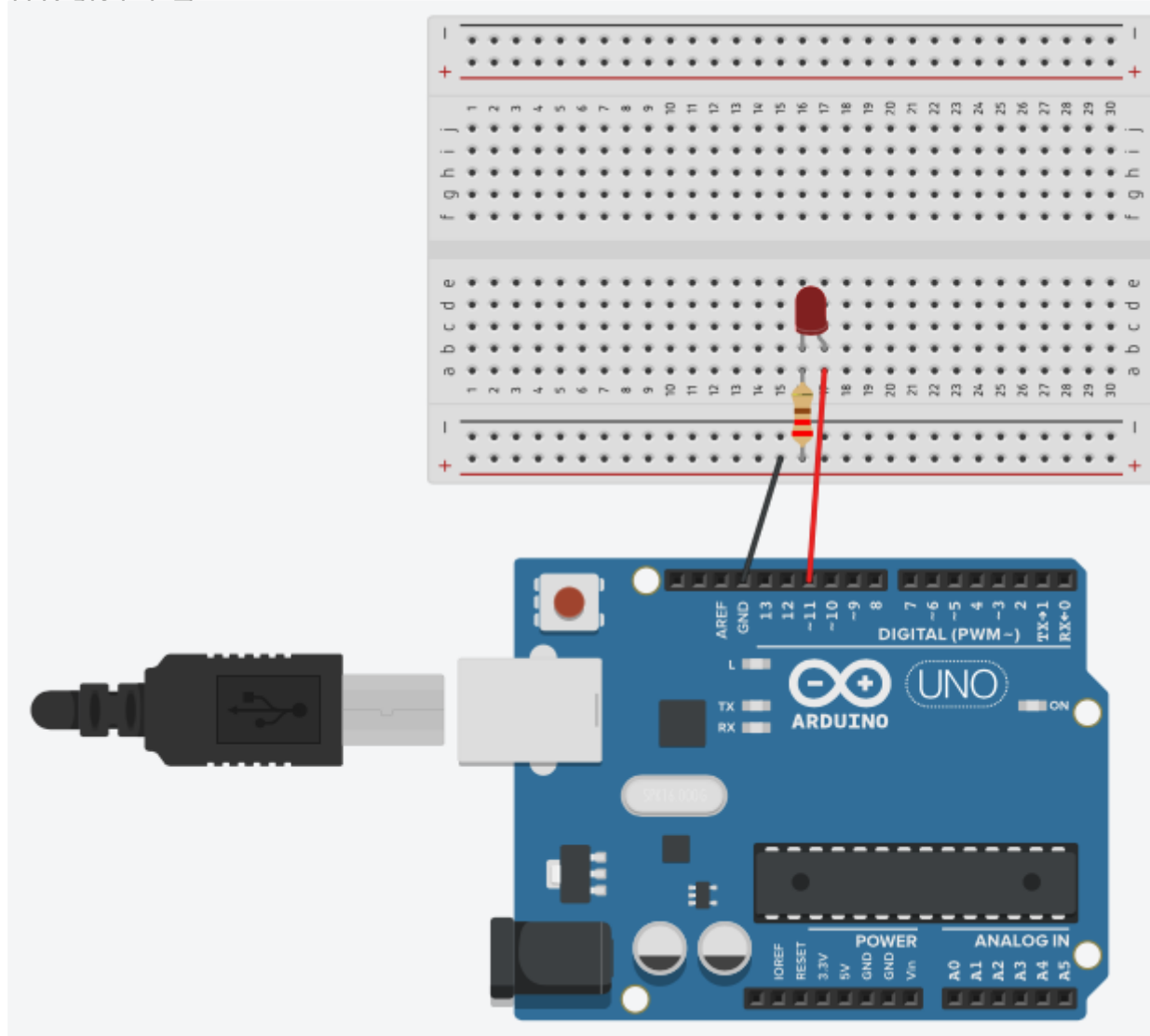
```
// the setup function runs once when you press reset or power the
board
void setup() { // initialize digital pin LED_BUILTIN as an
output.
pinMode(LED_BUILTIN, OUTPUT); } // the loop function runs
over and over again forever
void loop() {
digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH 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 }
```

- External LED (led connected to D13 pin)



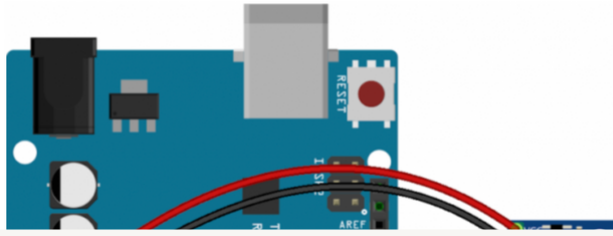
```
// Arduino Uno Blink code
void setup() { pinMode(13, OUTPUT); }
void loop() { digitalWrite(13, HIGH); delay(1000);
digitalWrite(13, LOW); delay(1000); }
```

- LED 밝기 조절



```
const int LED = 11; void setup() { } void loop() { for(int
t_high=0; t_high<=255; t_high++) { analogWrite(LED, t_high);
delay(4); } }
```

- 가속도 센서(accelerometer) - 상의 필요
  - Test : MPU-6050



```
// I2Cdev and MPU6050 must be installed as libraries, or else the
// .cpp/.h files // for both classes must be in the include path of
// your project #include "I2Cdev.h" #include "MPU6050.h" // Arduino
// Wire library is required if I2Cdev I2CDEV_ARDUINO_WIRE
// implementation // is used in I2Cdev.h #if I2CDEV_IMPLEMENTATION ==
// I2CDEV_ARDUINO_WIRE #include "Wire.h" #endif // class default I2C
// address is 0x68 // specific I2C addresses may be passed as a
// parameter here // AD0 low = 0x68 (default for InvenSense
// evaluation board) // AD0 high = 0x69 MPU6050 accelgyro; //MPU6050
// accelgyro(0x69); // <-- use for AD0 high int16_t ax, ay, az;
// int16_t gx, gy, gz; // uncomment "OUTPUT_READABLE_ACCELGYRO" if
// you want to see a tab-separated // list of the accel X/Y/Z and
// then gyro X/Y/Z values in decimal. Easy to read, // not so easy to
// parse, and slow(er) over UART. #define OUTPUT_READABLE_ACCELGYRO
// uncomment "OUTPUT_BINARY_ACCELGYRO" to send all 6 axes of data
// as 16-bit // binary, one right after the other. This is very fast
// (as fast as possible // without compression or data loss), and
// easy to parse, but impossible to read // for a human. // #define
// OUTPUT_BINARY_ACCELGYRO #define LED_PIN 13 bool blinkState =
// false; void setup() { // join I2C bus (I2Cdev library doesn't do
// this automatically) #if I2CDEV_IMPLEMENTATION ==
// I2CDEV_ARDUINO_WIRE Wire.begin(); #elif I2CDEV_IMPLEMENTATION ==
// I2CDEV_BUILTIN_FASTWIRE Fastwire::setup(400, true); #endif //
// initialize serial communication // (38400 chosen because it works
// as well at 8MHz as it does at 16MHz, but // it's really up to you
// depending on your project) Serial.begin(38400); // initialize
// device Serial.println("Initializing I2C devices...");
// accelgyro.initialize(); // verify connection
// Serial.println("Testing device connections...");
// Serial.println(accelgyro.testConnection() ? "MPU6050 connection
// successful" : "MPU6050 connection failed"); // configure Arduino
// LED pin for output pinMode(LED_PIN, OUTPUT); } void loop() { //
// read raw accel/gyro measurements from device
// accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz); // these
// methods (and a few others) are also available
// //accelgyro.getAcceleration(&ax, &ay, &az);
// //accelgyro.getRotation(&gx, &gy, &gz); #ifdef
// OUTPUT_READABLE_ACCELGYRO // display tab-separated accel/gyro
// x/y/z values Serial.print("a/g:\t"); Serial.print(ax);
// Serial.print("\t"); Serial.print(ay); Serial.print("\t");
```



```

Serial.print(az); Serial.print("\t"); Serial.print(gx);
Serial.print("\t"); Serial.print(gy); Serial.print("\t");
Serial.println(gz); #endif #ifdef OUTPUT_BINARY_ACCELGYRO
Serial.write((uint8_t)(ax >> 8)); Serial.write((uint8_t)(ax &
0xFF)); Serial.write((uint8_t)(ay >> 8)); Serial.write((uint8_t)
(ay & 0xFF)); Serial.write((uint8_t)(az >> 8));
Serial.write((uint8_t)(az & 0xFF)); Serial.write((uint8_t)(gx >>
8)); Serial.write((uint8_t)(gx & 0xFF)); Serial.write((uint8_t)(gy
>> 8)); Serial.write((uint8_t)(gy & 0xFF)); Serial.write((uint8_t)
(gz >> 8)); Serial.write((uint8_t)(gz & 0xFF)); #endif // blink
LED to indicate activity blinkState = !blinkState;
digitalWrite(LED_PIN, blinkState); }

```

- 최종 활동

가속도 센서를 흔들어, 값 변화량에 따라 LED 밝기 조절

```

// I2C device class (I2Cdev) demonstration Arduino sketch for
MPU6050 class // 10/7/2011 by Jeff Rowberg <jeff@rowberg.net> //
Updates should (hopefully) always be available at
https://github.com/jrowberg/i2cdevlib // // Changelog: // 2013-05-
08 - added multiple output formats // - added seamless Fastwire
support // 2011-10-07 - initial release /*
===== I2Cdev device library
code is placed under the MIT license Copyright (c) 2011 Jeff
Rowberg Permission is hereby granted, free of charge, to any
person obtaining a copy of this software and associated
documentation files (the "Software"), to deal in the Software
without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions: The above copyright notice and this permission notice
shall be included in all copies or substantial portions of the
Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF
ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE
AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
===== */ // I2Cdev and
MPU6050 must be installed as libraries, or else the .cpp/.h files
// for both classes must be in the include path of your project
#include "I2Cdev.h" #include "MPU6050.h" // Arduino Wire library

```

```

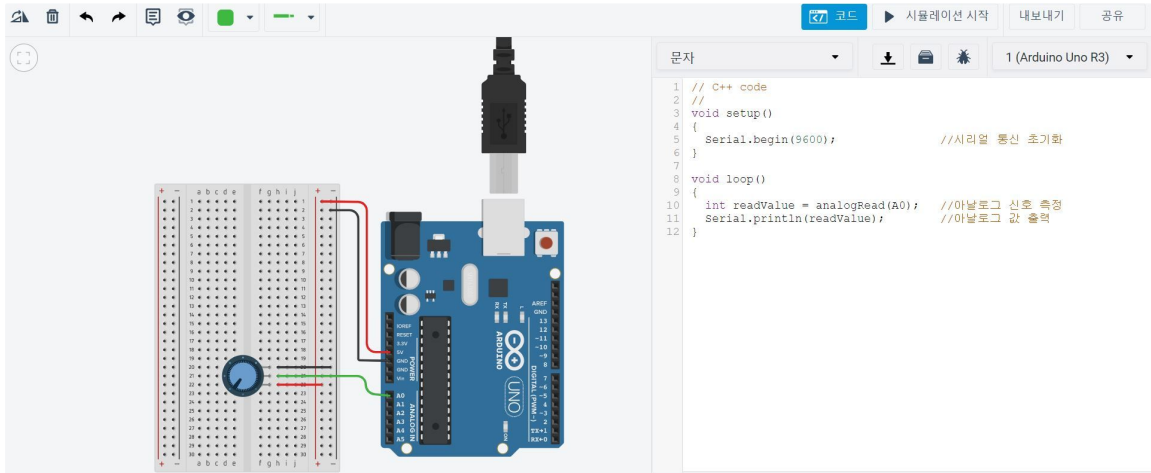
is required if I2Cdev I2CDEV_ARDUINO_WIRE implementation // is
used in I2Cdev.h #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h" #endif // class default I2C address is 0x68 //
specific I2C addresses may be passed as a parameter here // AD0
low = 0x68 (default for InvenSense evaluation board) // AD0 high =
0x69 MPU6050 accelgyro; //MPU6050 accelgyro(0x69); // <-- use for
AD0 high int16_t ax, ay, az; int16_t gx, gy, gz; // uncomment
"OUTPUT_READABLE_ACCELGYRO" if you want to see a tab-separated //
list of the accel X/Y/Z and then gyro X/Y/Z values in decimal.
Easy to read, // not so easy to parse, and slow(er) over UART.
#define OUTPUT_READABLE_ACCELGYRO // uncomment
"OUTPUT_BINARY_ACCELGYRO" to send all 6 axes of data as 16-bit //
binary, one right after the other. This is very fast (as fast as
possible // without compression or data loss), and easy to parse,
but impossible to read // for a human. // #define
OUTPUT_BINARY_ACCELGYRO #define LED_PIN 13 int LED_pin = 11; bool
blinkState = false; void setup() { // join I2C bus (I2Cdev library
doesn't do this automatically) #if I2CDEV_IMPLEMENTATION ==
I2CDEV_ARDUINO_WIRE Wire.begin(); #elif I2CDEV_IMPLEMENTATION ==
I2CDEV_BUILTIN_FASTWIRE Fastwire::setup(400, true); #endif //
initialize serial communication // (38400 chosen because it works
as well at 8MHz as it does at 16MHz, but // it's really up to you
depending on your project) Serial.begin(115200); // initialize
device Serial.println("Initializing I2C devices...");
accelgyro.initialize(); // verify connection
Serial.println("Testing device connections...");
Serial.println(accelgyro.testConnection() ? "MPU6050 connection
successful" : "MPU6050 connection failed"); // use the code below
to change accel/gyro offset values /* Serial.println("Updating
internal sensor offsets..."); // -76 -2359 1688 0 0 0
Serial.print(accelgyro.getXAccelOffset()); Serial.print("\t"); //
-76 Serial.print(accelgyro.getYAccelOffset()); Serial.print("\t");
// -2359 Serial.print(accelgyro.getZAccelOffset());
Serial.print("\t"); // 1688
Serial.print(accelgyro.getXGyroOffset()); Serial.print("\t"); // 0
Serial.print(accelgyro.getYGyroOffset()); Serial.print("\t"); // 0
Serial.print(accelgyro.getZGyroOffset()); Serial.print("\t"); // 0
Serial.print("\n"); accelgyro.setXGyroOffset(220);
accelgyro.setYGyroOffset(76); accelgyro.setZGyroOffset(-85);
Serial.print(accelgyro.getXAccelOffset()); Serial.print("\t"); //
-76 Serial.print(accelgyro.getYAccelOffset()); Serial.print("\t");
// -2359 Serial.print(accelgyro.getZAccelOffset());
Serial.print("\t"); // 1688
Serial.print(accelgyro.getXGyroOffset()); Serial.print("\t"); // 0
Serial.print(accelgyro.getYGyroOffset()); Serial.print("\t"); // 0
Serial.print(accelgyro.getZGyroOffset()); Serial.print("\t"); // 0
Serial.print("\n"); */ // configure Arduino LED pin for output
pinMode(LED_PIN, OUTPUT); } int previous_value; void loop() { //

```

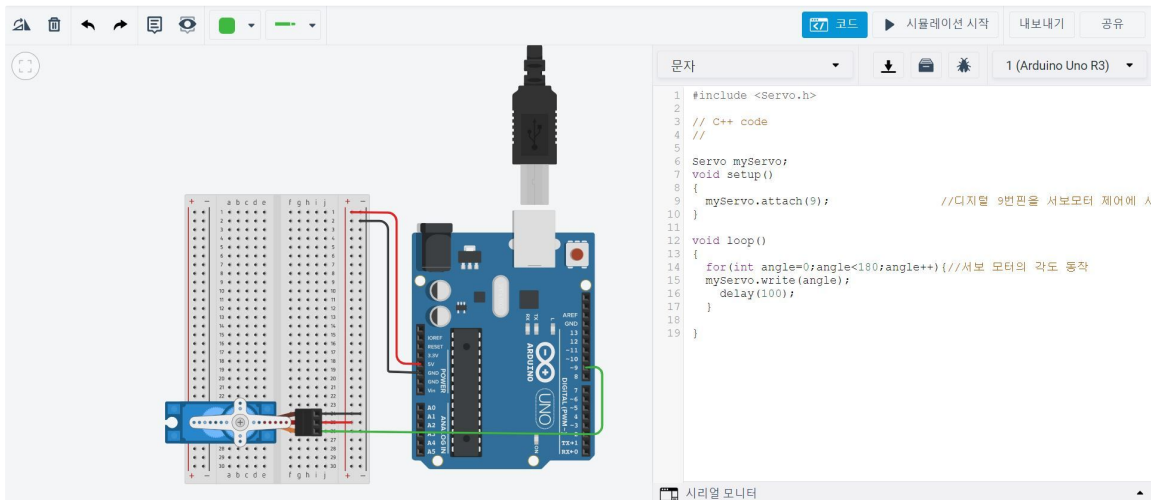
```
read raw accel/gyro measurements from device
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz); // these
methods (and a few others) are also available
//accelgyro.getAcceleration(&ax, &ay, &az);
//accelgyro.getRotation(&gx, &gy, &gz); #ifdef
OUTPUT_READABLE_ACCELGYRO // display tab-separated accel/gyro
x/y/z values Serial.print("a/g:\t"); Serial.print(ax);
Serial.print("\t"); Serial.print(ay); Serial.print("\t");
Serial.print(az); Serial.print("\t"); Serial.print(gx);
Serial.print("\t"); Serial.print(gy); Serial.print("\t");
Serial.println(gz); #endif #ifdef OUTPUT_BINARY_ACCELGYRO
Serial.write((uint8_t)(ax >> 8)); Serial.write((uint8_t)(ax &
0xFF)); Serial.write((uint8_t)(ay >> 8)); Serial.write((uint8_t)
(ay & 0xFF)); Serial.write((uint8_t)(az >> 8));
Serial.write((uint8_t)(az & 0xFF)); Serial.write((uint8_t)(gx >>
8)); Serial.write((uint8_t)(gx & 0xFF)); Serial.write((uint8_t)(gy
>> 8)); Serial.write((uint8_t)(gy & 0xFF)); Serial.write((uint8_t)
(gz >> 8)); Serial.write((uint8_t)(gz & 0xFF)); #endif if ( abs(ax
+ ay + az - previous_value) > 1000) { int brightness= map(0,
10000,0,255); analogWrite(LED_pin,brightness); } previous_value =
ax + ay + az; // blink LED to indicate activity blinkState =
!blinkState; digitalWrite(LED_PIN, blinkState); }
```

## ▼ 4차시(정현)

## • 가변 저항 기본

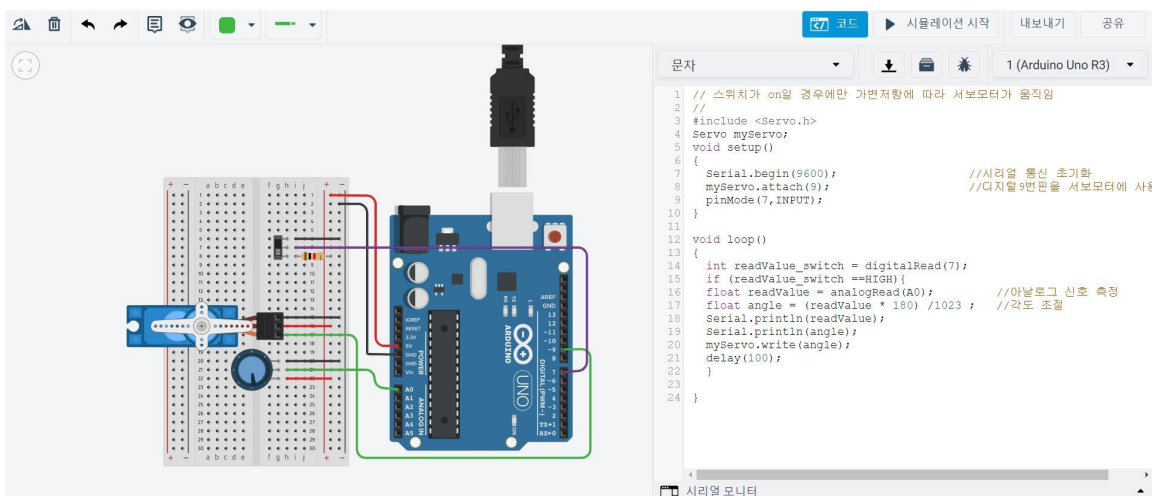


## • 서보모터 기본



## • 예시

가변저항 돌리는 만큼 서보모터가 돌아가는 예시 코드 입니다. 스위치 사용도 다뤄서 실습에 사용할 예정입니다.



- 실습

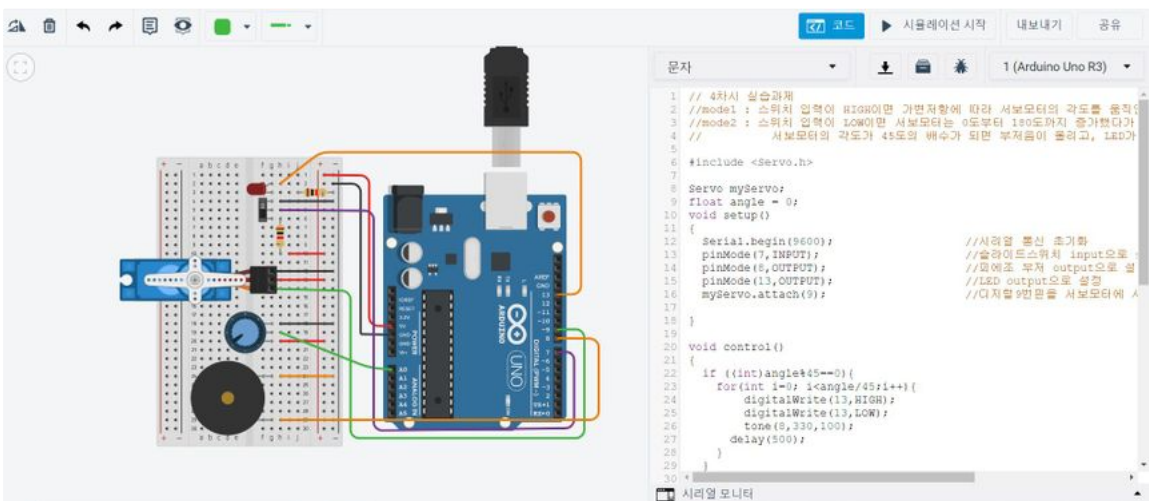
스위치로 mode1과 mode2로 구분

mode1 : 스위치 입력이 HIGH이면 가변저항에 따라 서보모터의 각도를 움직인다.-예시와 동일

mode2 : 스위치 입력이 LOW이면 서보모터는 0도부터 180도까지 증가했다가 0도로 다시 감소한다.

서보모터의 각도가 45도의 배수가 되면 부저음이 울리고, LED가 깜박인다. (45도는 1번, 90도는 2번...)

부저와 LED제어하는 함수를 만들어 사용할 것



// 4차시 실습과제 //mode1 : 스위치 입력이 HIGH이면 가변저항에 따라 서보모터의 각도를 움직인다. //mode2 : 스위치 입력이 LOW이면 서보모터는 0도부터 180도까지 증가했다가 0도로 다시 감소한다. // 서보모터의 각도가 45도의 배수가 되면 부저음이 울리고, LED가 깜박인다. (45도는 1번, 90도는 2번...)

```

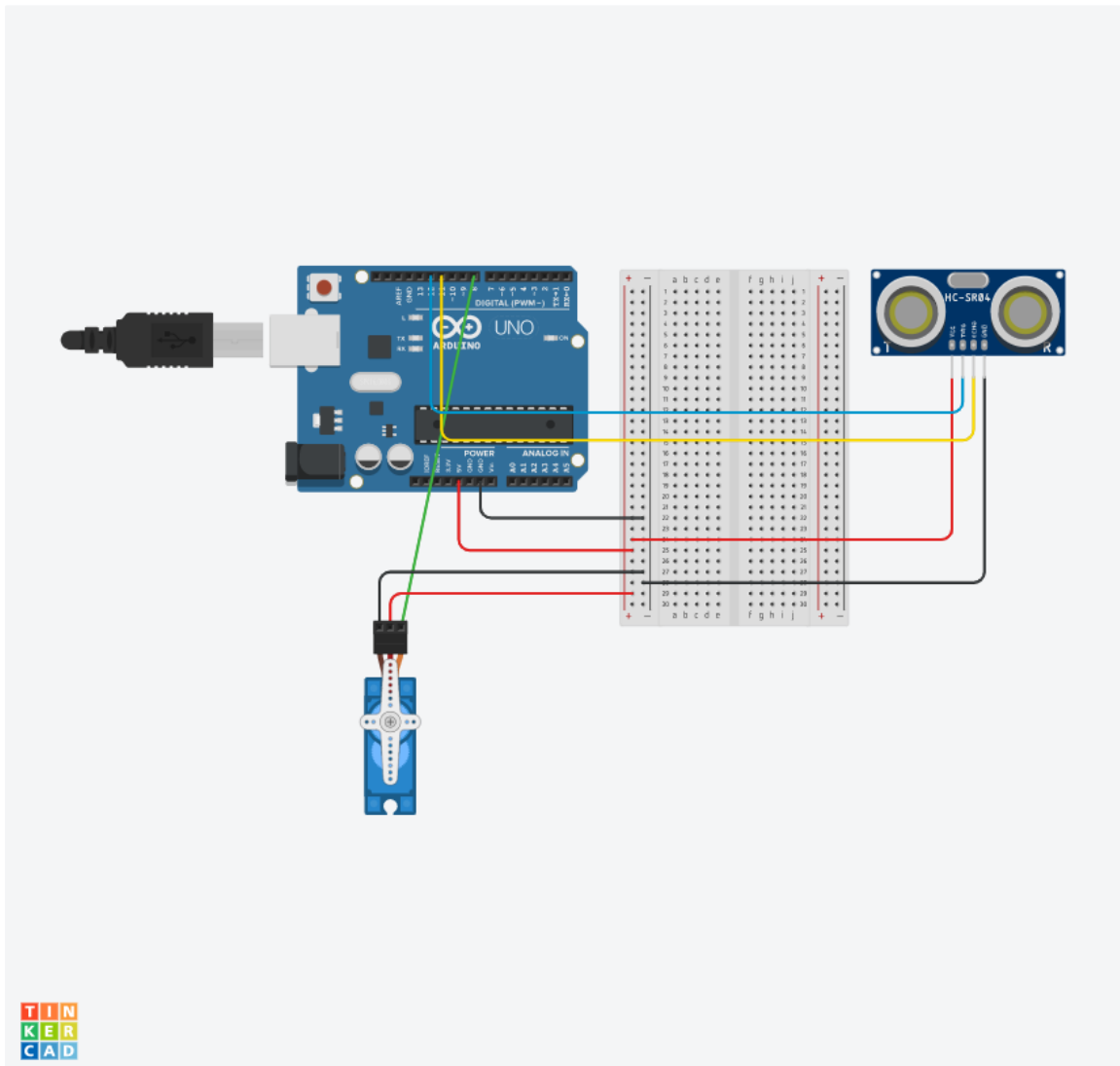
#include <Servo.h> Servo myServo; float angle = 0; void setup() {
  Serial.begin(9600); //시리얼 통신 초기화 pinMode(7,INPUT); //슬라이드스
  위치 input으로 설정 pinMode(8,OUTPUT); //피에조 부저 output으로 설정
  pinMode(13,OUTPUT); //LED output으로 설정 myServo.attach(9); //디지털9
  번핀을 서보모터에 사용 } void control() { if ((int)angle%45==0){
  for(int i=0; i<angle/45;i++){ digitalWrite(13,HIGH);
  digitalWrite(13,LOW); tone(8,330,100); delay(500); } } void loop() {
  float readValue = analogRead(A0); //아날로그 신호 측정 int
  readValue_switch = digitalRead(7); //스위치 신호 측정 if
  (readValue_switch == HIGH){ //mode1 angle = (readValue * 180) / 1023 ;
  //가변저항에 따라 각도 조절 Serial.println(angle);
  myServo.write(angle); delay(50); } else if (readValue_switch == LOW){
  //mode2 for(angle=0; angle<180; angle++){ //각도 증가
  myServo.write(angle); control(); delay(50); } for(angle = 180;

```

```
angle>0; angle--){//각도 감소 myServo.write(angle); control();  
delay(50); } } }
```

## ▼ 5차시(동원)

- 최종 산출물



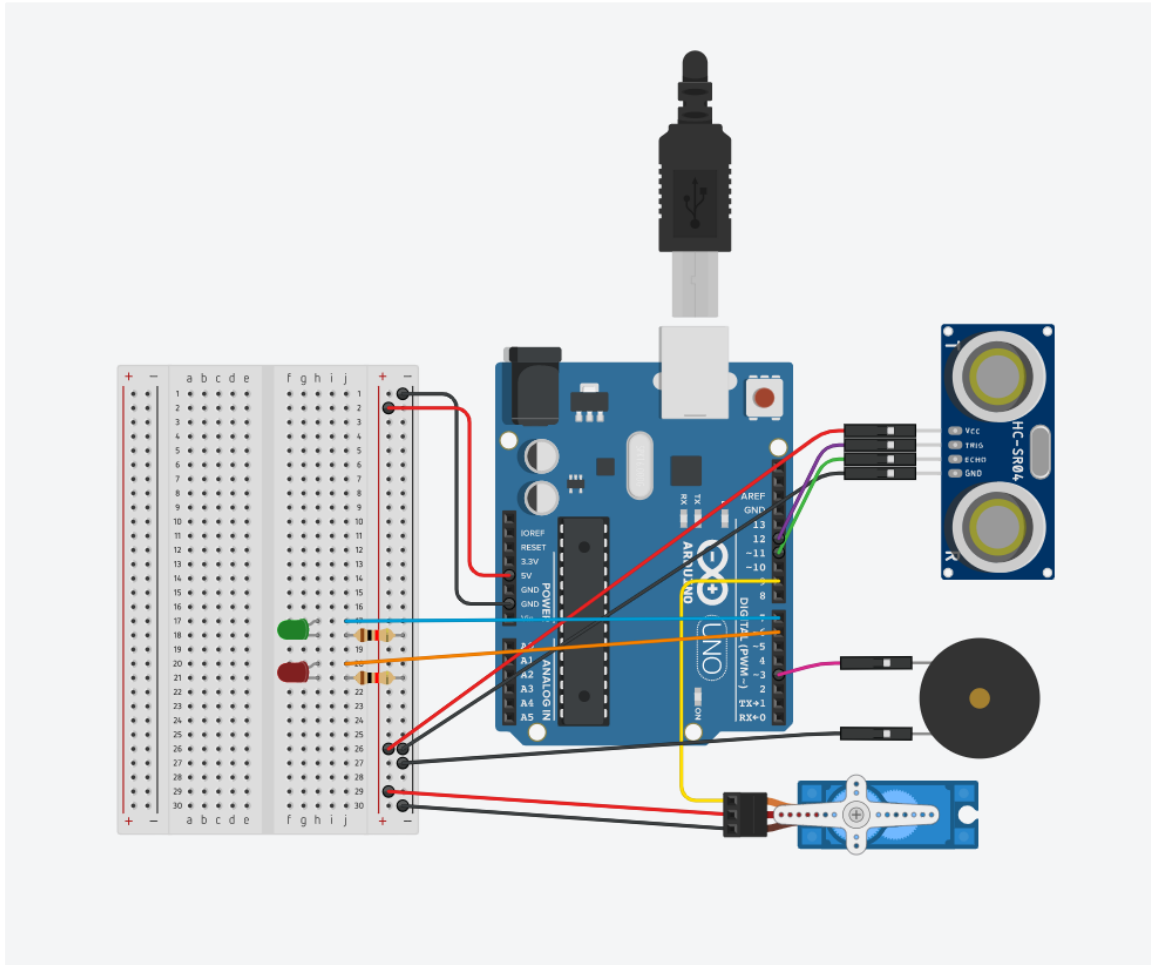
```
#include <Servo.h> #define echoPin 11 #define trigPin 12 #define
servoPin 9 Servo servo; void setup() { Serial.begin(9600);
pinMode(echoPin, INPUT); pinMode(trigPin, OUTPUT);
servo.attach(servoPin); } void loop() { if(get_distance() < 10){
for(int angle = 0 ; angle<90 ; angle++){ servo.write(angle);
delay(50); } for(int angle = 90 ; angle>0 ; angle--){
servo.write(angle); delay(50); } } } float get_distance(){
digitalWrite(trigPin, HIGH); delay(10); digitalWrite(trigPin, LOW);
float duration = pulseIn(echoPin, HIGH); float distance = ((float)(340
* duration) / 10000) / 2; return distance; }
```

- 실습과제 예시

이전에 배운 센서를 활용하여 기능 추가하기



ex) 손소독제 작동시에 LED 색 변화와 소리 출력 기능 추가



```
#include <Servo.h> #define echoPin 11 #define trigPin 12 #define
servoPin 9 #define ledPin1 6 #define ledPin2 7 #define BuzzerPin 3
Servo servo; void setup() { Serial.begin(9600); pinMode(echoPin,
INPUT); pinMode(trigPin, OUTPUT); pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT); pinMode(BuzzerPin, OUTPUT);
digitalWrite(ledPin1, HIGH); digitalWrite(ledPin2, LOW);
servo.attach(servoPin); } void loop() { if(get_distance() < 10){
digitalWrite(ledPin1, LOW); digitalWrite(ledPin2, HIGH); for(int angle
= 0 ; angle<90 ; angle++){ servo.write(angle); delay(50); } tone(8,
330, 100); for(int angle = 90 ; angle>0 ; angle--){
servo.write(angle); delay(50); } digitalWrite(ledPin1, HIGH);
digitalWrite(ledPin2, LOW); } } float get_distance(){
digitalWrite(trigPin, HIGH); delay(10); digitalWrite(trigPin, LOW);
float duration = pulseIn(echoPin, HIGH); float distance = ((float)(340
* duration) / 10000) / 2; return distance; }
```



