

// IoT 5일차 (서보 모터, 온도계)

// 서보모터 9g



주황 - 9pin  
빨강 - 5v  
검정 - gnd

회전각도 0~180도,  
그 이상 서보 모터 손상  
사용전압 4.8~7.2V  
사용전류 0.2~0.7A  
무게 9g  
토크 1.8kg.cm  
크기 22.2 x 11.8 x 31 mm

<https://youtu.be/bLnAJ-mSEIE>

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

```
Servo myservo;
```

```
void setup(){
```

```
    Serial.begin(9600); // 시리얼 통신 9600 보드레이트,
```

```
    myservo.attach(9); // 9번핀 연결한 서보모터 활성화.
```

```
}
```

```
void loop(){
```

```
    if(Serial.available()){ // 시리얼 모니터로 받는 값이 있다면,
```

```
        int speed = Serial.parseInt(); // 그 값을 정수로 변환하여 speed로 저장.
```

```

if(speed >= 0 && speed <= 180){ // 값의 범위 0~180

    myservo.write(speed); // 서보 모터 동작.

}

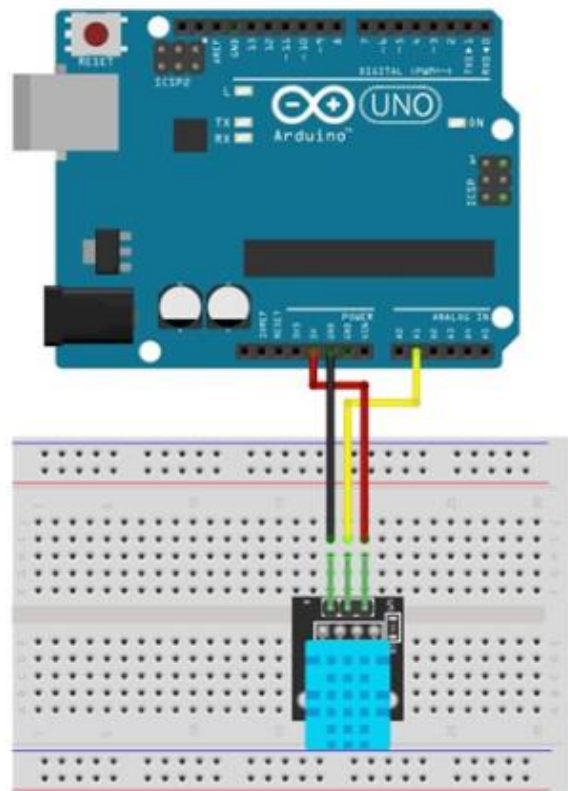
}

}

// 온습도 센서

```

우노 보드	온 · 습도 센서
5V	VCC
GND	GND
A1	DATA



<결선표 및 결선도>

<https://kocoafab.cc/tutorial/view/379>

```
#include "DHT.h"
```

```
// I2C LCD 를 쉽게 제어하기 위한 라이브러리를  
추가해줍니다.
```

```
#include <LiquidCrystal_I2C.h>
```

```
// 온습도 센서를 디지털 2 번 핀에 연결합니다.
#define DHTPIN A1
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

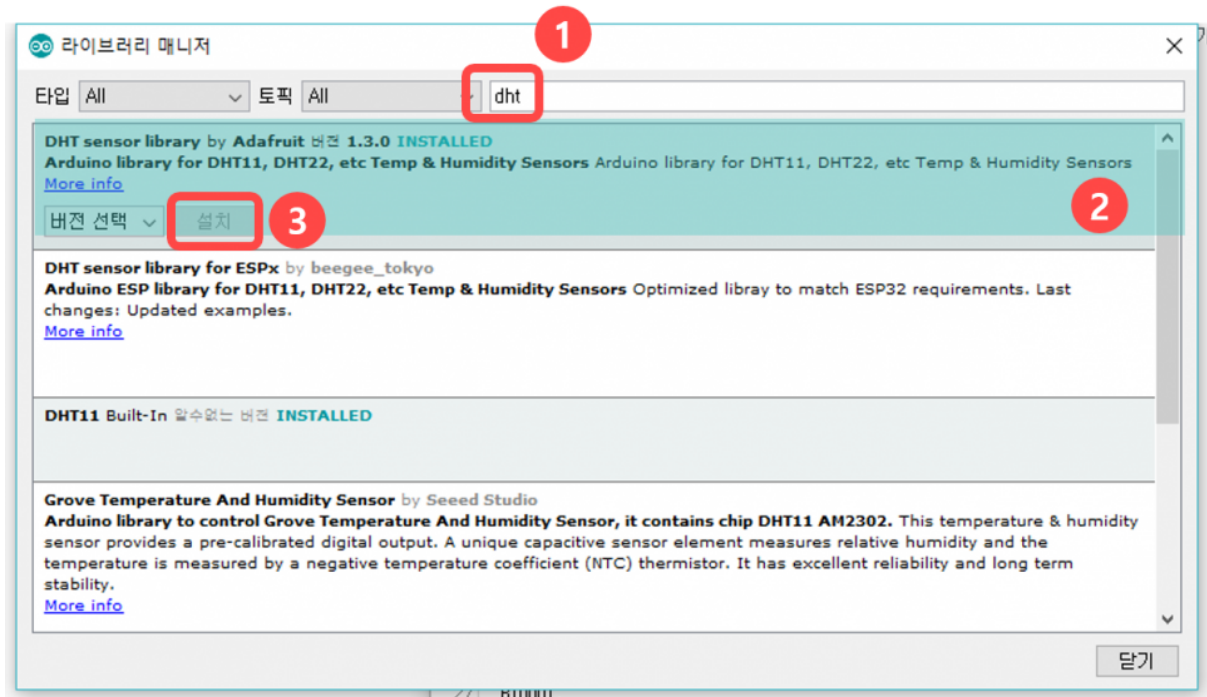
void setup() {
    Serial.begin(9600);
    dht.begin();
}

void loop() {
    // 습도와 온도값을 측정하고, 제대로 측정되었는지
    확인해줍니다.
    float humidity = dht.readHumidity();
    float temperature = dht.readTemperature();

    if (isnan(humidity) || isnan(temperature) ) {
        Serial.println("Failed to read from DHT
sensor!");
        return;
    }

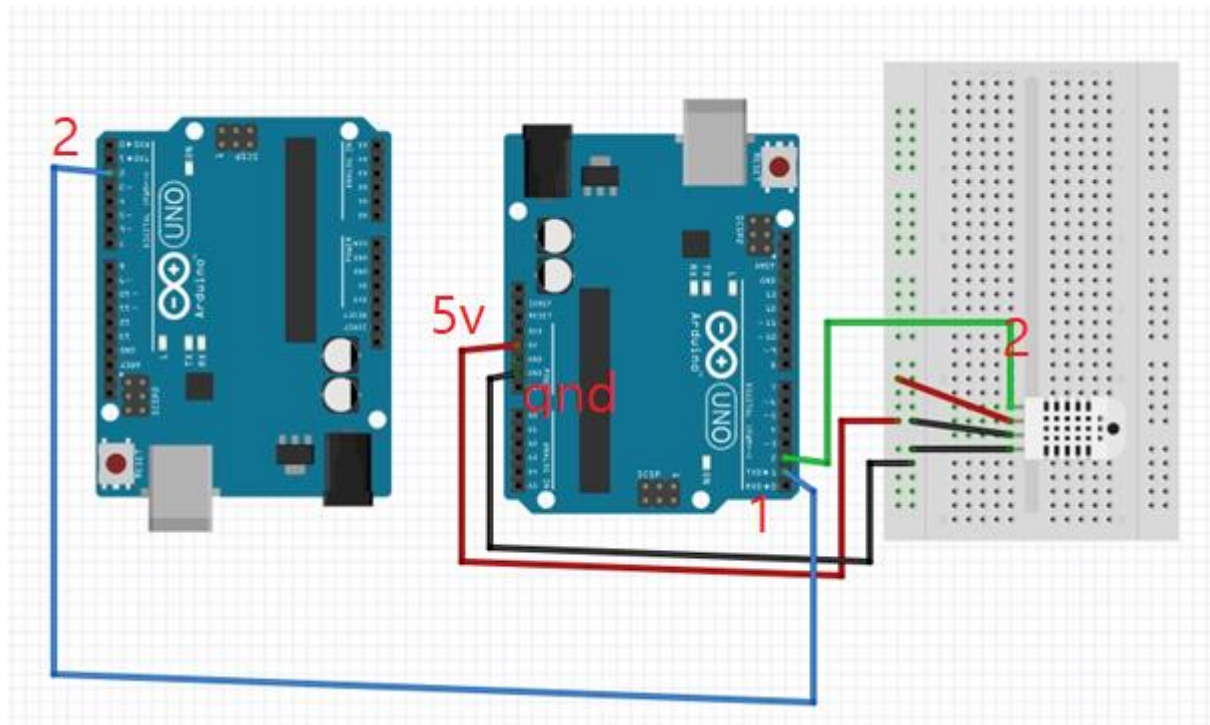
    // 온도와 습도값을 시리얼 모니터에 출력해 줍니다.
    Serial.print((int)temperature); Serial.print("
*C, ");
    Serial.print((int)humidity);
    Serial.println(" %");
}
```

```
delay(1500);  
}
```



install all 로 라이브러리 설치.

// 2개 아두이노 시리얼 통신.



// 압전(piezo) 스피커(buzzer) :



함수 이름	설명
tone(핀 번호, 주파수); 또는 tone(핀 번호, 주파수, 출력 시간);	사각파를 만들어 소리를 내게 한다. <ul style="list-style-type: none"> <li>• 핀 번호: 디지털 핀 번호</li> <li>• 주파수: 표 5-5에 나와 있는 값으로 도레미 음계를 나타냄(Hz)</li> <li>• 출력 시간: 소리를 내는 시간(밀리초)</li> </ul>
noTone(핀 번호);	tone 함수로 시작된 사각파 신호 발생을 멈춘다. <ul style="list-style-type: none"> <li>• 핀 번호: 디지털 핀 번호</li> </ul>

```
void setup(){
```

```
void loop(){
```

```
    tone(9,490); //piezo buzzer 피에조 부저
```

```
    delay(500);
```

```
    noTone(9);
```

```
}
```

```
// 징글벨
```

```
int speakerPin = 9;
```

```
int length = 26;
```

```
char notes[] = "eeeeeeegcde fffffeeeddedg";
```

```
// 부분부분 잘라서, 문자와 대응하는 숫자로 변환.
```

```
int beats[] = { 1, 1, 2, 1, 1, 2, 1, 1,
```

```
                1, 1, 4, 1, 1, 1, 1, 1,
```

```
                1, 1, 1, 1, 1, 1, 1, 1,
```

```
                2, 2
```

```
};
```

```
int tempo = 300;
```

```
void playTone(int tone, int duration) {
```

```
    for (long i = 0; i < duration * 1000L; i += tone * 2) {
```

```
        digitalWrite(speakerPin, HIGH);
```

```

    delayMicroseconds(tone);

    digitalWrite(speakerPin, LOW);

    delayMicroseconds(tone);
}

}

void playNote(char note, int duration) {

    char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' }; // 8음계 이름

    int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 }; // 8음계 높낮이


    // play the tone corresponding to the note name

    for (int i = 0; i < 8; i++) {

        if (names[i] == note) {

            playTone(tones[i], duration); // 높낮이와 재생시간.

        }

    }

}

void setup() {

    pinMode(speakerPin, OUTPUT); // 9번핀 출력.

}

void loop() {

    for (int i = 0; i < length; i++) { // 징글벨 길이.

        if (notes[i] == ' ') {

            delay(beats[i] * tempo); // 공백이 있으면 쉬었다가.

        } else {

```

```

    playNote(notes[i], beats[i] * tempo); // 소리 높낮이와 소리 간격.
}

// pause between notes

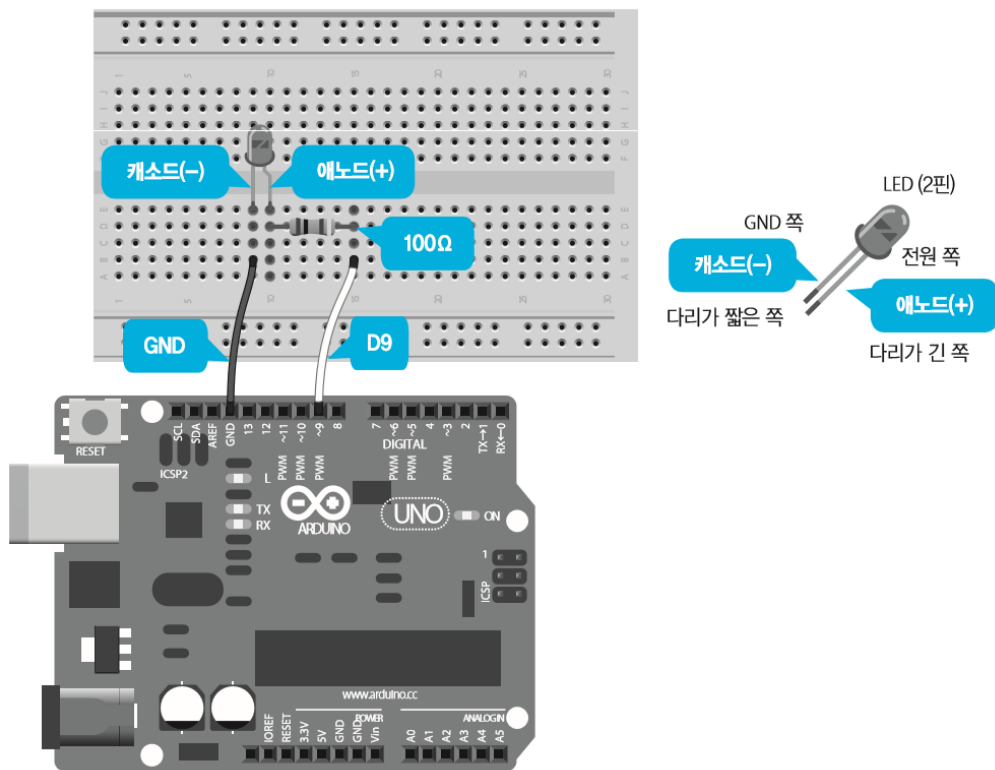
delay(tempo / 2);

}

}

// 엘이디 밝기 조절

```



```

void setup() {
}

```



```

void loop() {

    for (int x = 0; x < 255; x++) {

        analogWrite(9, x );

        delay(200);

    }

}

```

// 서서히 꺼지는 경우, pwm 100 이상의 경우, 차이가 육안으로 확인이 안되므로,  
100 에서 1씩 감소 처리하고, delay(10); 으로 빠르게 처리해야 구별 가능.

// 시간차 이용.

```

void setup() {

    pinMode(9, OUTPUT);

}

```

```

void loop() {

    for (int x = 0; x < 10; x++) {

        for (int i = 0; i < 10; i++) {

            digitalWrite(9, HIGH);        // LED를 켜
            delay(x);                      // 서서히 길게

            digitalWrite(9, LOW);         // LED를 끄
            delay(9 - x);                  // 서서히 짧게

        }

    }

}

```

```
}
```

```
// 차량용 후방 감지기 만들기. (초음파 센서 + 피에조 스피커)
```

```
#define TRIGPIN 8    // 트리거(송신쪽) 핀  
#define ECHOPIN 9   // 에코(수신쪽) 핀  
#define CTM 10      // HIGH 인 시간( $\mu$  초)  
#define spk 5  
#define beep 7902
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(TRIGPIN, OUTPUT); // 트리거 핀을 디지털 출력으로 설정  
  pinMode(ECHOPIN, INPUT);  // 에코 핀을 디지털 입력으로 설정  
}
```

```
int ms = 1000;
```

```
void loop() {  
  int dur;          // 시간 차( $\mu$  초)  
  float dis;        // 거리(cm)
```

```
  digitalWrite(TRIGPIN, HIGH);  
  delayMicroseconds(CTM);  
  digitalWrite(TRIGPIN, LOW);  
  dur = pulseIn(ECHOPIN, HIGH); // HIGH 가 되기까지 걸리는 시간을 측정  
  dis = (float) dur * 0.017;    // 음속을 사용해서 거리 계산
```

```
  if (dis > -106) { // 잘못된 값 제거.  
    Serial.print(dis);
```

```
    if (dis < 40) {  
      if (dis > 30) {  
        ms = 500;  
      }  
      else if (dis > 20) {  
        ms = 200;  
      }  
    }
```

```

else {
    ms = 50;
}
} else ms = 1000;

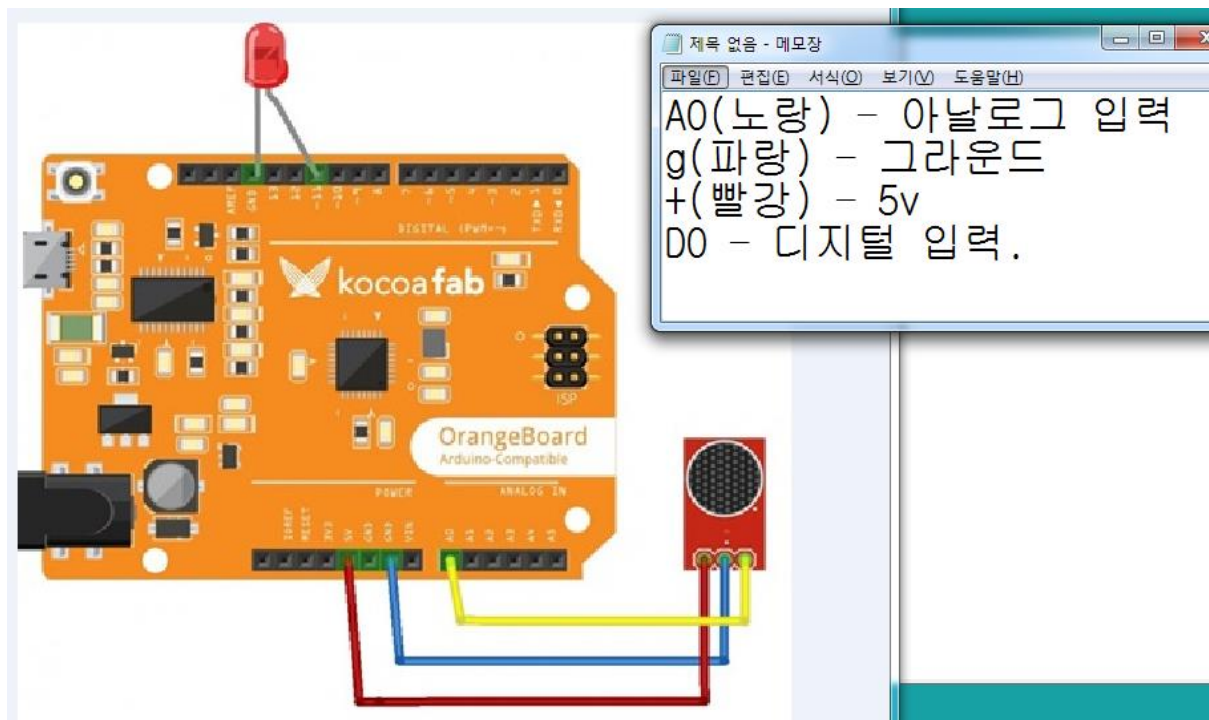
tone(5, beep, 500);

delay(ms);
noTone(5);

} else {
    Serial.print(0);
}
Serial.println(" cm");
Serial.print("ms : ");
Serial.println(ms);
delay(50);
}

```

// 소리 감지 센서 이용. (아래 그림에서 LED에 적절히 저항 달아 주세요.)



```
int sensorPin = A0;

// LED를 11번핀에 연결합니다.

int ledPin = 11;


void setup() {

    // ledPin을 출력으로 설정합니다.

    pinMode(ledPin, OUTPUT);

    //시리얼 통신을 시작합니다.

    Serial.begin(9600);

}


void loop() {

    // 사운드 센서로부터 MIC 센서값을 읽어 들입니다.

    int value = analogRead(sensorPin);

    // 읽어들인 MIC 센서의 값을 0~255로 변환합니다.

    int intensity = map(value, 20, 300, 0, 255);

    //ledPin에 변환된 값을 넣어줍니다.

    analogWrite(ledPin, intensity);

    //센서값을 시리얼 모니터로 출력하여 현재 센서값을 보여줍니다.

    Serial.println(value);

    //1000ms 동안 대기

    delay(100);

}


// 주말 잘 보내시고요. 월요일에 다시 만나요~
```