

06주차

아두이노 입력모듈 : 조도 및 초음파 센서

목차

1. 조도센서
2. 초음파센서

01 조도센서

저항에 의한 전압 분배 회로

■ 아날로그 센서

- 물리량 변화에 따라 저항 성질이 바뀌는 재료를 활용하여 센서 제작
- 가변저항과 같은 방식으로 전원을 가하여 저항값을 알아내고 물리량 파악
- 저항 성질이 바뀌는 재료를 사용하는 센서
 - 힘 센서, 구부림 센서 : 누르거나 구부릴 때 저항 성질이 변함
 - 조도 센서 : 빛의 양에 따라 저항 성질이 변함
 - 서미스터(thermistor) : 온도에 따라 저항 성질이 변함
 - 초음파 센서 : 반사물에 반사되어 돌아오는 시간을 측정해 거리를 측정

빛에 따라 저항이 변하는 이유

■ 조도센서 재질

- 조도센서는 반도체(대표적으로 CdS: 황화카드뮴)로 만들어져 있음
- 반도체는 전기가 잘 통할 수도 있고, 잘 안 통할 수도 있음

■ 어두울 때

- 전자가 대부분 제자리에 갇혀 있어서 움직이지 못함
- 전기가 잘 안 통해서 저항이 커짐.

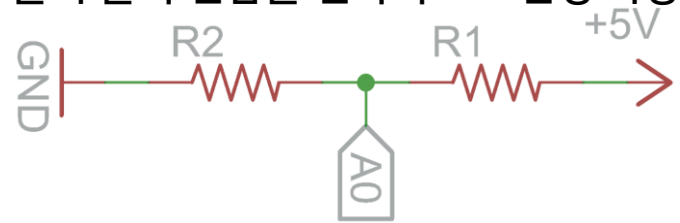
■ 밝을 때(빛을 받을 때)

- 빛 속에는 에너지를 가진 작은 입자(광자)가 있음.
- 이 빛이 반도체에 닿으면, 갇혀 있던 전자가 에너지를 받아서 풀려남
- 풀려난 전자가 움직이며 전류가 흐름 → 전기가 잘 통함 → 저항이 작아짐.

저항에 의한 전압 분배 회로

■ 전압 분배 회로

- 2개의 저항을 사용하여 사용한 저항 크기에 따라 입력 전압 일부만 출력
- 2개 저항 중 하나를 가변저항으로 사용하면 손잡이를 돌려 출력 전압을 연속적으로 변경 가능

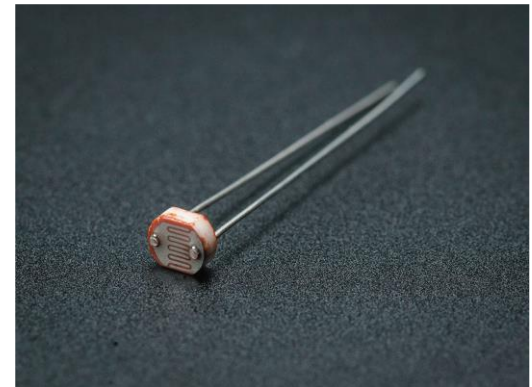


■ 전압 분배 식

- $V_{out} = V_{in} \left(\frac{R2}{R1+R2} \right)$
- R1, R2 저항 크기에 따라 출력 전압(V_{out}) 결정

■ 조도 센서

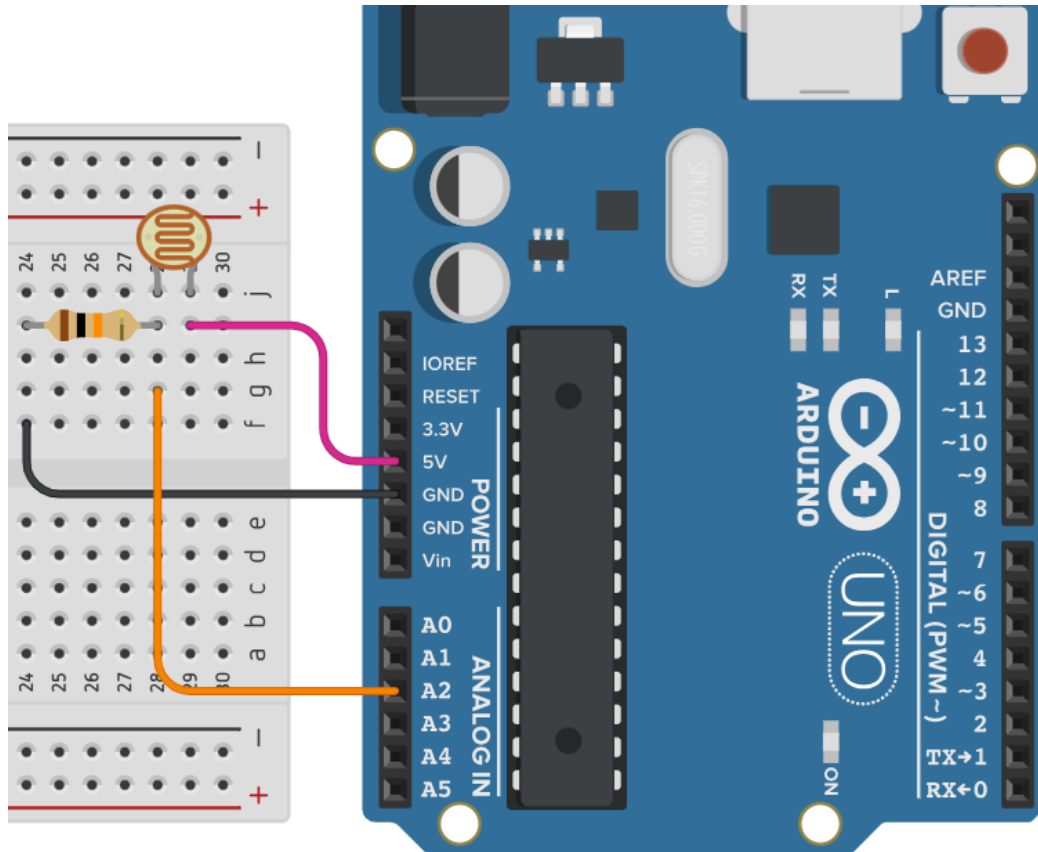
- 광량에 따라 저항값이 변함
- 완전히 어두울 때 약 200k Ω
빛이 가득 찬 상태일 때 약 5k Ω



조도 센서

■ 조도 센서 연결 회로

- **10k Ω 저항** 사용(밝기 변화 전체를 고르게 감지하기 좋은 절충값)
- 손으로 조도 센서를 가리거나 플래시로 빛 조절하며 시리얼 모니터 출력 확인



사용 함수

■ analogRead

- 물리량 변화에 따른 센서의 저항값을 디지털 값으로 변환하여 읽기
- 0~1023 사이 값

■ map

- `output = map(value, fromLow, fromHigh, toLow, toHigh)`
- 입력값(value)의 범위(fromLow~fromHigh)를 다른 범위(toLow~toHigh)로 변환 출력

■ constrain

- `output = constrain(value, min, max)`
- 입력값(value)이 최소 min 값, 최대 max 값을 가지도록 제한하여 출력

uno09_01.ino : 범위 내 광량 측정

```
1  const int INPUT_PIN = A2;
2  const int MIN_LIGHT = 100;
3  const int MAX_LIGHT = 700;
4  int val = -1;
5
6  void setup() {
7      Serial.begin(9600);
8      pinMode(INPUT_PIN, INPUT);
9  }
10
11 void loop() {
12
13     val = analogRead(INPUT_PIN);
14     val = map(val, MIN_LIGHT, MAX_LIGHT, 0, 255);
15     val = constrain(val, 0, 255);
16     Serial.println(val);
17     delay(500);
18
19 }
```

analogRead(INPUT_PIN)

- 조도센서에서 들어오는 전압(0~5V)을 10비트 값으로 읽음
- 즉, 0~1023 범위의 숫자가 들어옴
 - 0 → 0V
 - 1023 → 5V

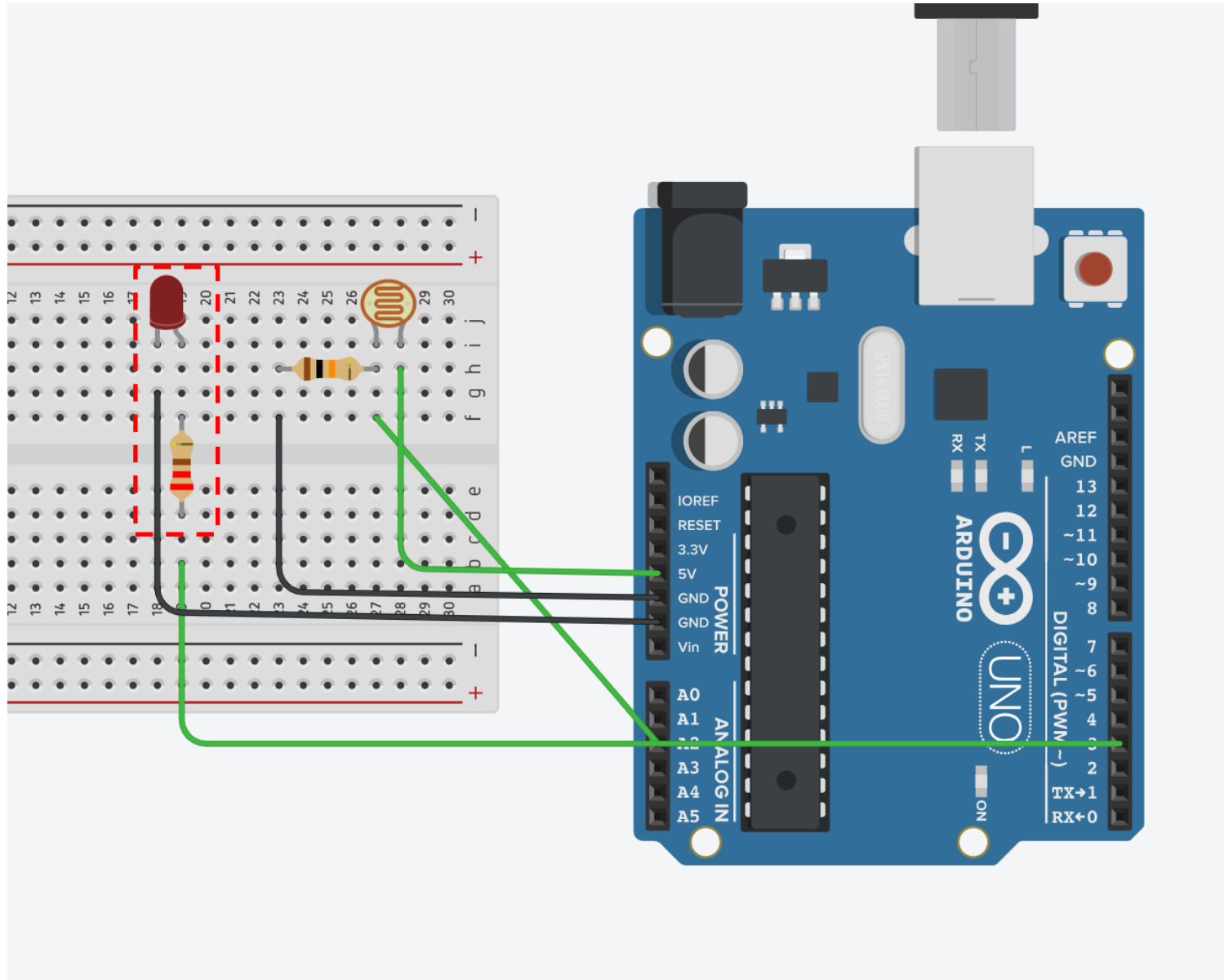
map(val, MIN_LIGHT, MAX_LIGHT, 0, 255)

- map() 함수는 입력값 범위를 다른 범위로 변환
- 여기서는 조도센서의 실제 동작 범위가 100~700 정도라서 그 구간을 **0~255**로 바꾸는 역할을 함
- 예를 들어,
 - val = 100 (어두움) → 0
 - val = 700 (밝음) → 255

constrain(val, 0, 255)

- 혹시 map 계산 과정에서 값이 0보다 작거나 255보다 클 경우 강제로 0~255 범위 안으로 조정함

광량을 측정하여 LED 켜기 & 끄기 - 회로





uno09_02.uno : 광량을 측정하여 LED 켜기 & 끄기

```
1  const int INPUT_PIN = A2;
2  const int OUTPUT_PIN = 3;
3
4  const int MIN_LIGHT = 100;
5  const int MAX_LIGHT = 800;
6  int val = -1;
7
8  void setup() {
9      Serial.begin(9600);
10     pinMode(INPUT_PIN, INPUT);
11     pinMode(OUTPUT_PIN, OUTPUT);
12
13 }
14
15 void loop() {
16
17     val = analogRead(INPUT_PIN);
18     val = map(val, MIN_LIGHT, MAX_LIGHT, 0, 255);
19     val = constrain(val, 0, 255);
20
21     Serial.println(val);
22
23     if(val<50) digitalWrite(OUTPUT_PIN, 1);
24     else digitalWrite(OUTPUT_PIN, 0);
25
26     delay(500);
27
28 }
```

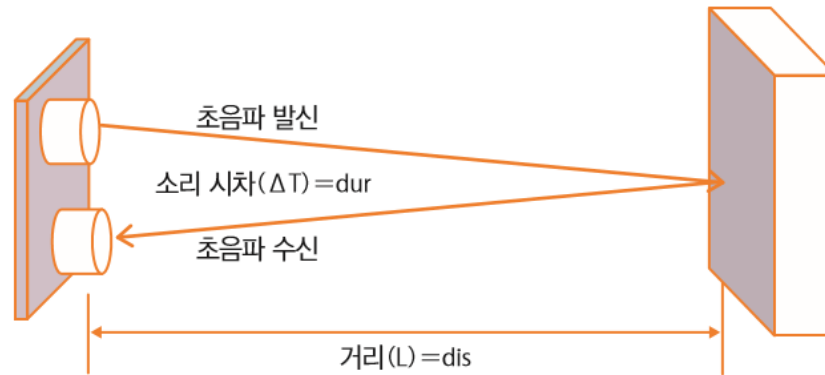
02 초음파 센서

초음파 센서

- 초음파 센서는 사람의 귀로는 들을 수 없는 고주파 소리를 출력한 후, 반사물에 반사되어 돌아오는 시간을 측정해 거리를 측정
- 사람, 동물, 물건 등이 다가오고 멀어지는 것을 감지하거나, 물이 불어나는 정도를 측정하거나, 전장까지의 높이를 측정하는 등 다양하게 활용
- 측정할 수 있는 거리의 범위는 수 센티미터에서 3~4m까지

제품 이름	HC-SR04	SEN136B5B
핀 수	4	3
핀의 의미 (정면 왼쪽부터)	Vcc: 5V Trig: 송신 쪽 핀 Echo: 수신 쪽 핀 Gnd: 접지	SIG: 송수신 핀 VCC: 5V GND: 접지
측정 거리 범위	2cm~4m	3cm~4m
제품		

초음파 센서 거리 측정



- 장애물까지의 거리(L), 장애물에 반사된 초음파가 돌아오는 데 걸리는 시간 (T), 음속(C) 사이의 관계를 다음과 같은 식으로 정리할 수 있음
- 초음파 거리 센서는 디지털 신호 HIGH와 LOW의 전환을 통해 송수신기(발신 쪽과 수신 쪽)에서 시간 차를 읽을 수 있음

초음파 거리 센서에 사용하는 식

$$L = C \times T / 2$$

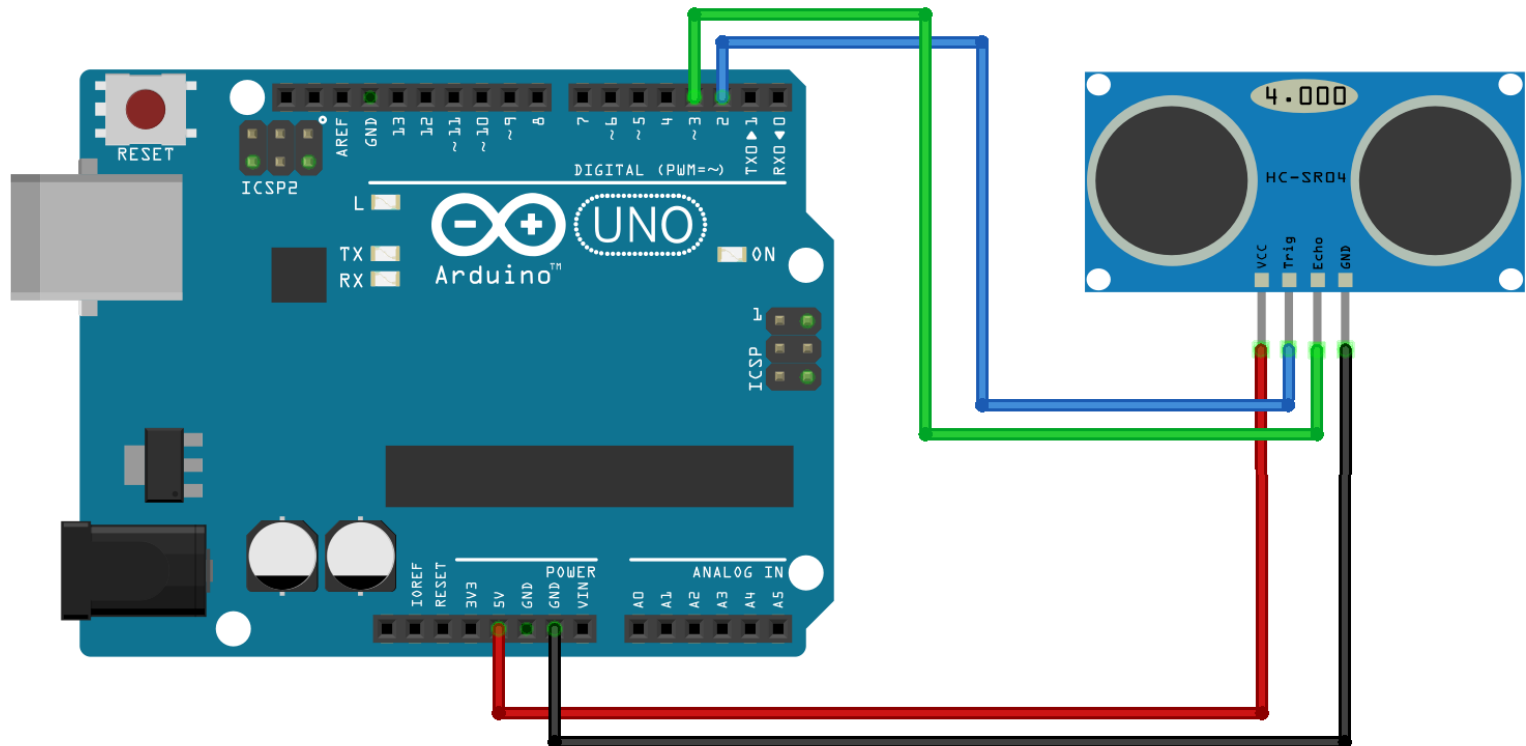
L은 장애물까지의 거리

C는 음속(간이식은 $331 + 0.6 \times t$: 단위 m/s)

T는 초음파를 발신한 직후부터 수신하기까지 걸리는 시간


초음파 센서 회로

- 5V → 5V (VCC)
- Trig → D2
- Echo → D3
- GND → GND



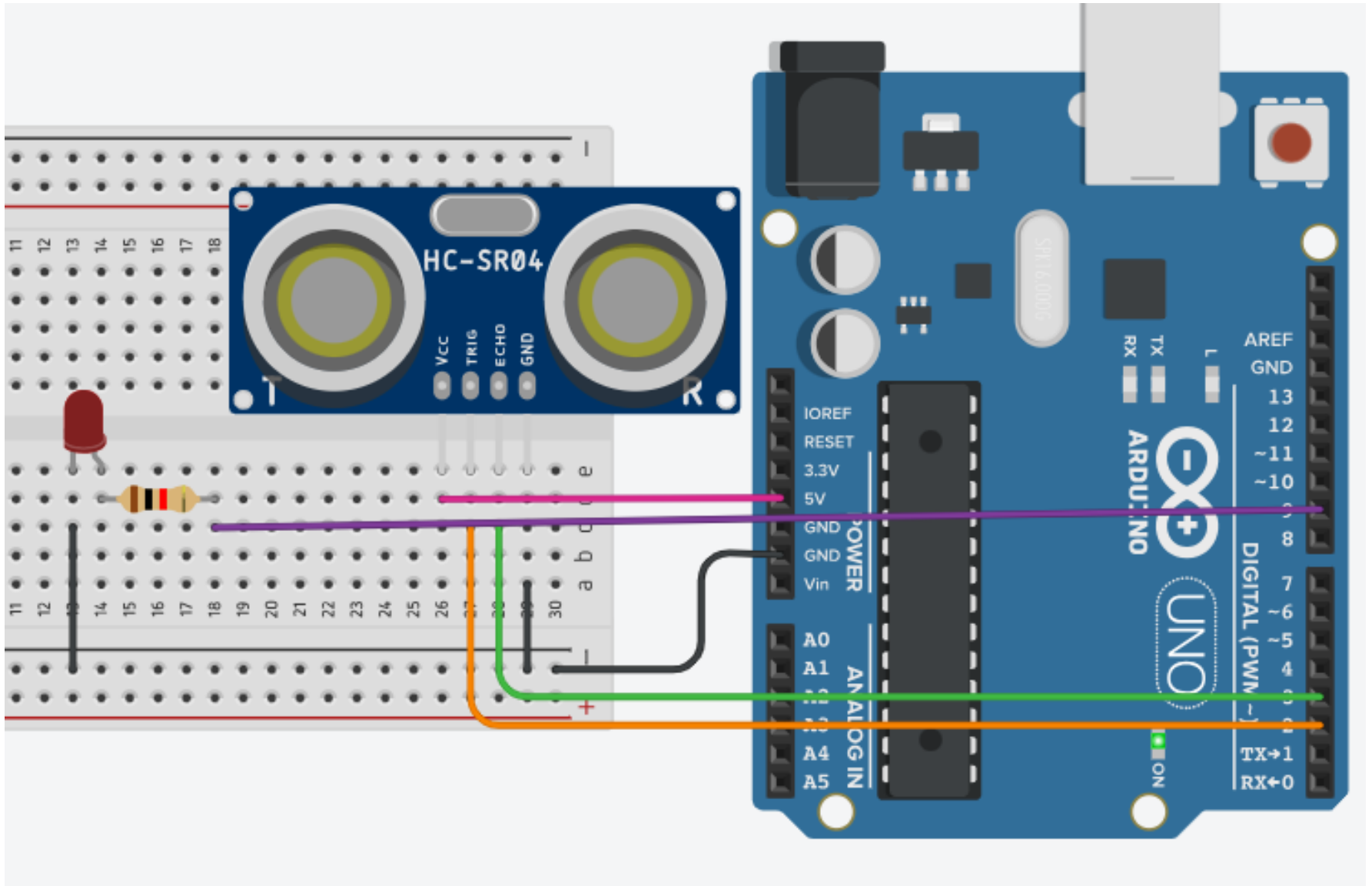
uno09_03.ino : 거리 측정

```
1  #define TRIGPIN 2      // 트리거(송신쪽) 핀      // 트리거 핀 (초음파 신호를 내보내는 쪽)
2  #define ECHOPIN 3     // 에코(수신쪽) 핀        // 에코 핀 (반사파를 받는 쪽)
3  #define CTM 10        // HIGH인 시간(μ 초)      // 트리거 신호를 HIGH로 유지할 시간 (μs)
4
5  int dur;               // 시간 차(μ초)
6  float dis;             // 거리(cm)
7
8  void setup() {
9      Serial.begin(9600);
10     pinMode(TRIGPIN, OUTPUT);    // 트리거 핀을 디지털 출력으로 설정
11     pinMode(ECHOPIN, INPUT);     // 에코 핀을 디지털 입력으로 설정
12 }
13 void loop() {
14
15
16     digitalWrite(TRIGPIN, HIGH); // 트리거 핀을 HIGH로 10μs 동안 유지 →
17     delayMicroseconds(CTM);       // 초음파를 '딱 한 번' 발사
18     digitalWrite(TRIGPIN, LOW);
19     dur = pulseIn(ECHOPIN, HIGH); // HIGH가 되기까지 걸리는 시간을 측정
20     dis = (float) dur * 0.017;    // 음속을 사용해서 거리 계산
21     Serial.print(dis);
22     Serial.println(" cm");
23     delay(500);
24
25 }
```



// 3. 거리 계산 (단위: cm)
// 음속 약 340m/s → 1μs 동안 약 0.034cm 이동
// 왕복 거리이므로 0.034 / 2 = 0.017

uno09_04.ino : 사람 인식하면 LED 켜기



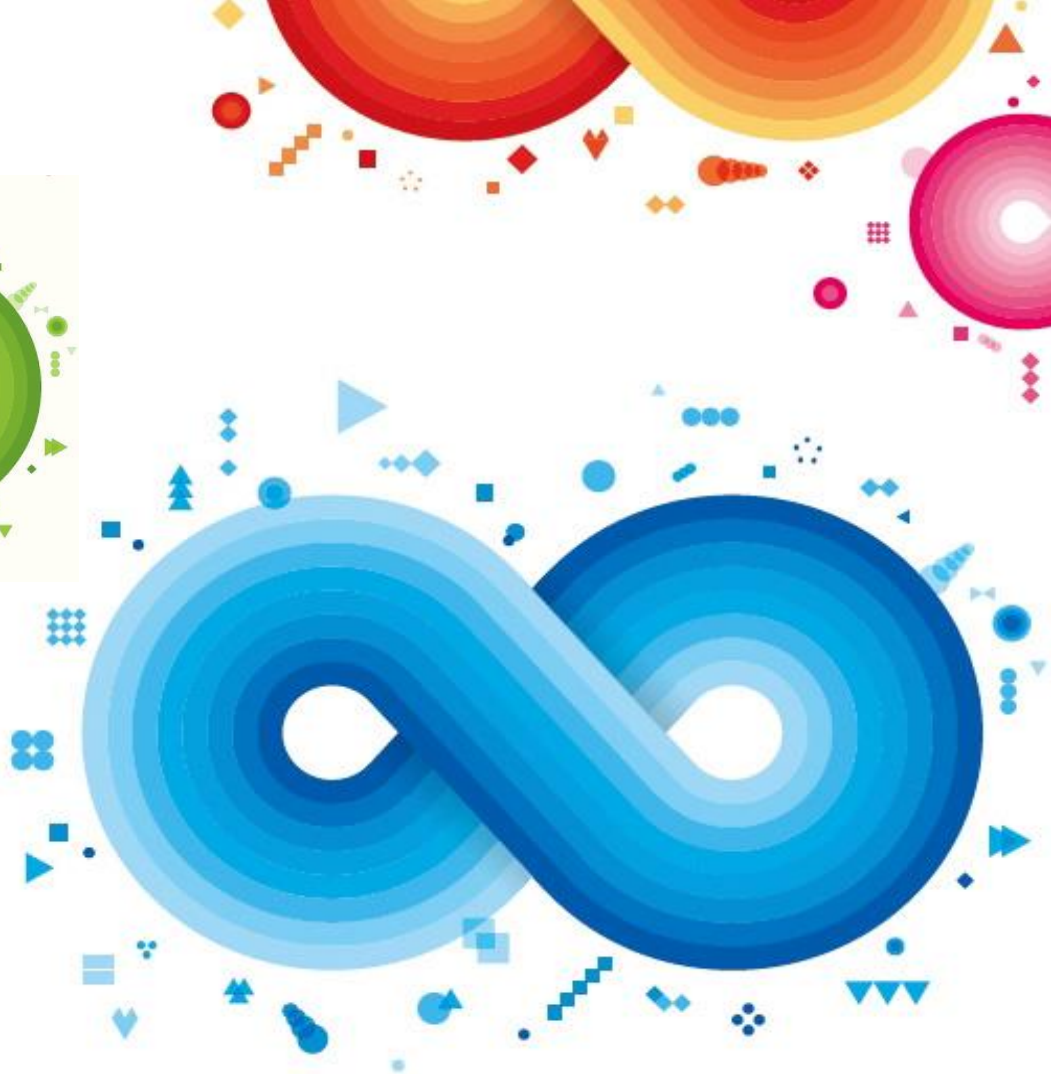
uno09_04.ino

: 사람 인식하면 LED 켜기

```
1  #define TRIGPIN 2      // 트리거(송신쪽) 핀
2  #define ECHOPIN 3     // 에코(수신쪽) 핀
3  #define LEDPIN 9      // LED 핀
4
5  #define CTM 10         // HIGH인 시간(μ 초)
6
7  int dur;               // 시간 차(μ초)
8  float dis;             // 거리(cm)
9
10 void setup() {
11     Serial.begin(9600);
12     pinMode(TRIGPIN, OUTPUT);    // 트리거 핀을 디지털 출력으로 설정
13     pinMode(ECHOPIN, INPUT);    // 에코 핀을 디지털 입력으로 설정
14     pinMode(LEDPIN, OUTPUT);    // LED 핀을 디지털 출력으로 설정
15 }
16
17 void loop() {
18
19
20     digitalWrite(TRIGPIN, HIGH);
21     delayMicroseconds(CTM);
22     digitalWrite(TRIGPIN, LOW);
23     dur = pulseIn(ECHOPIN, HIGH);    // HIGH가 되기까지 걸리는 시간을 측정
24     dis = (float) dur * 0.017;      // 음속을 사용해서 거리 계산
25     Serial.print(dis);
26     Serial.println(" cm");
27
28
29     if(dis<50) { //장애물이 가까이오면
30         digitalWrite(LEDPIN, HIGH); //LED 5초간 켜기
31         delay(5000);
32         digitalWrite(LEDPIN, LOW);
33     }
34
35     delay(500);
36
37 }
```

요약

- 조도 센서: 광량 조절하는 센서
 - 빛의 세기를 아날로그 값으로 받음
- 초음파 센서: 디지털 신호를 보내어 거리 측정 하는 센서
 - 디지털 신호를 Trig핀에서 보내고 Echo핀에서 신호를 받아 시간 차를 통해 거리 측정

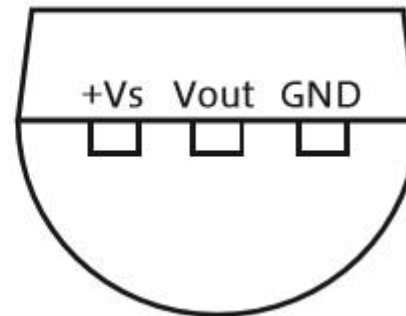


06주차

아두이노 입력모듈: 온도 센서

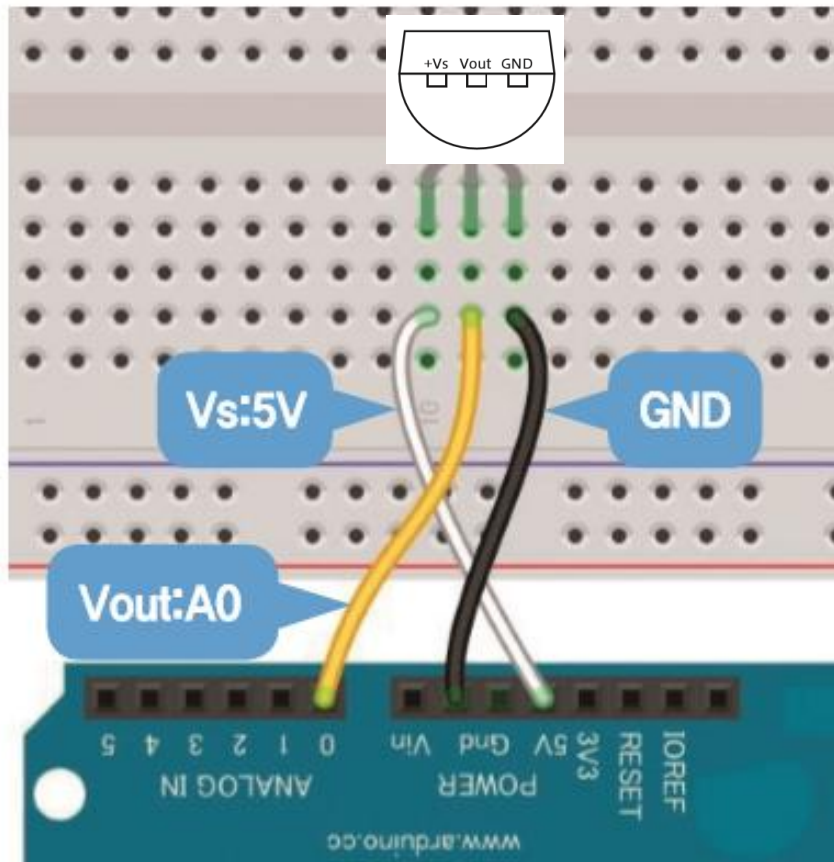
온도센서

- 온도 센서는 바깥 온도 변화에 따라 저항 값이 바뀌는 아날로그 센서
- 이러한 특성을 통해 섭씨온도(단위: $^{\circ}\text{C}$)나 화씨온도(단위: $^{\circ}\text{F}$)로 변환 계산을 함
- 핀이 3개 : 양쪽 끝 핀이 전원 전압($+V_s$)과 접지(GND,그라운드)이고, 가운데 핀이 출력 전압(V_{out})



회로

- 온도 값이 출력되는 Vout 핀은 아날로그 입력 포트 A0 핀에 연결
- 반드시 주의하여 연결! 쇼트남



출력전압을 통한 온도계산

•아두이노의 analogRead() 함수는 0~1023 사이의 정수 값을 반환함
•0 → 0V
•1023 → 기준 전압 Vs(보통 5V)
•즉, 아날로그 입력을 디지털로 변환(ADC) 한 값임.

- 스케치를 작성하기 전에 먼저 온도 센서의 출력 사양을 확인
- 사양을 보면 온도 센서의 출력 전압(Vout)의 관계식은 아래와 같음

$$V = V_{out} * V_s / 1024$$

- 이 식에서 Vs는 전원 전압
- 아두이노 우노의 전원은 5V이므로 다음과 같이 바꿔 쓸 수 있음

$$V = V_{out} * 5 / 1024$$

■ 출력 전압을 통한 온도계산(섭씨기준, C)

- 온도센서의 종류에 따라 다름

* 차이점은 온도가 0도일때 TMP36은 0.5V를 출력하지만 LM35의 경우에는 0V를 출력한다.

— TMP36

$$C = (V - 0.5) / 0.01;$$

— LM35

$$C = V / 0.01;$$

출력전압을 통한 온도계산

<TMP36 온도센서>

- TMP36센서는 출력 전압이 온도에 비례함

- 주요 특징:

- 0°C일 때 출력 전압: 0.5 V

- 온도 1°C 상승 → 출력 전압 10 mV 증가
(즉, 1°C → 0.01 V)

- 공식 의미

- $$C = \frac{V - 0.5}{0.01}$$

- V: 센서에서 측정한 전압 (단위: V)

- 0.5를 빼는 이유: 센서가 0°C일 때 0.5 V를 출력하므로, 기준점을 0°C로 맞추기 위해 0.5를 뺌

- 0.01로 나누는 이유: 전압 0.01 V(10 mV)가 온도 1°C에 해당하기 때문

uno10_1.uno

ADC는 **Analog-to-Digital Converter**의 약자임
즉, 아날로그 신호를 디지털
신호로 변환하는 장치를 의미
함

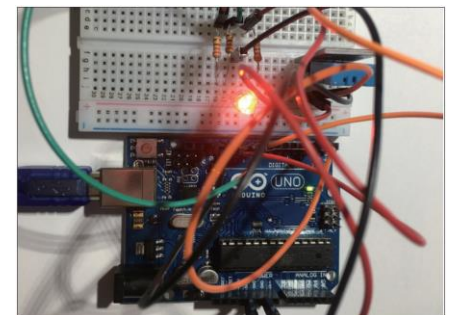
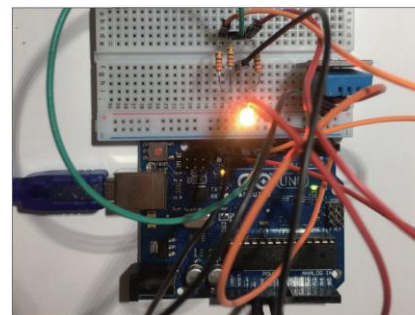
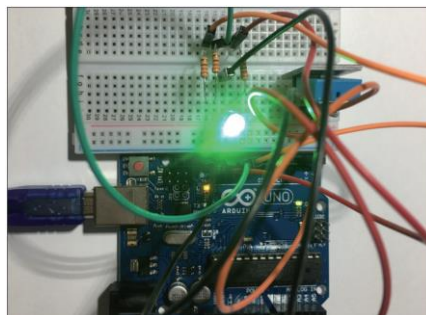
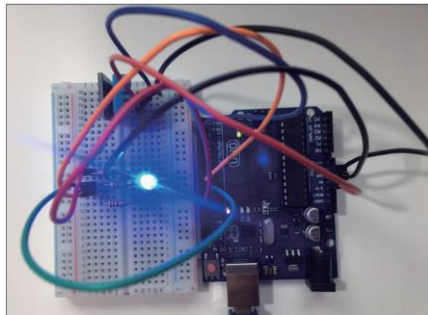
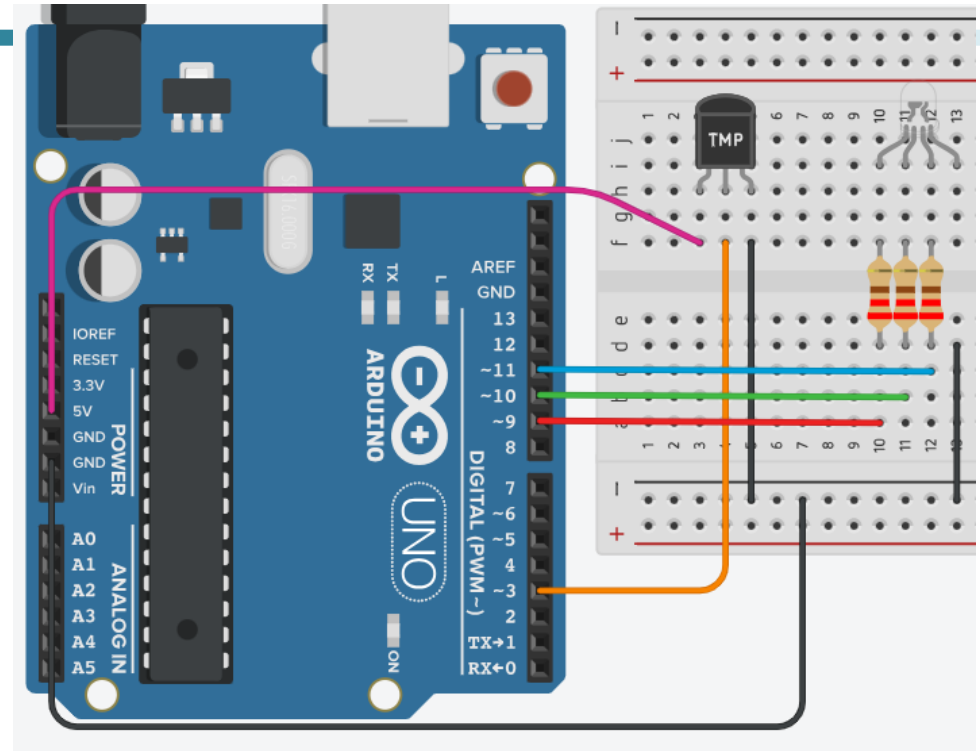
- 아날로그 신호: 연속적인 값,
예를 들어 온도 센서에서 나
오는 0~5V 전압
- 디지털 신호: 컴퓨터가 처리
할 수 있는 0과 1로 된 값, 예
를 들어 0~1023 범위의 정수

그래서 아두이노에서
analogRead()를 쓰면 ADC
를 통해 전압을 디지털 값으
로 변환해서 읽음

```
1  int tmp_sensor = A0;
2
3  int Vout;
4  float V;
5  float tempC,tempF;
6
7  void setup(){
8      Serial.begin(9600);
9      pinMode(tmp_sensor, INPUT);
10 }
11
12 void loop(){
13
14     Vout = analogRead(tmp_sensor);    // 1. 아날로그 값 읽기 (0~1023)
15
16     V = Vout * 5.0 / 1024.0;          // 2. ADC 값 → 전압 변환
17
18     tempC = (V - 0.5) / 0.01; //TMP36
19     //tempC = V / 0.01; //LM35
20
21     tempF = tempC *1.8 +32; //화씨
22
23     Serial.print("value: ");
24     Serial.print(Vout);
25
26     Serial.print("/voltage:");
27     Serial.print(V);
28
29     Serial.print("/Temperature:");
30     Serial.print(tempC);
31     Serial.print("C, ");
32     Serial.print(tempF);
33     Serial.println("F");
34     delay(1000);
35
36 }
```

온도에 따른 LED 변화

- $\sim 10^{\circ}\text{C}$ -> BLUE
 - $10 \sim 17^{\circ}\text{C}$ -> GREEN
 - $17 \sim 23^{\circ}\text{C}$ -> ORANGE
 - $23^{\circ}\text{C} \sim$ -> RED
-
- Pin: A0 -> 9, 10, 11



10°C

17°C

23°C

uno10_2.uno

```
1  int tmp_sensor = A0;
2
3  int Vout;
4  float V;
5
6  float tempC;
7
8  void setup(){
9
10     Serial.begin(9600);
11     pinMode(tmp_sensor, INPUT);
12     pinMode(9, OUTPUT);
13     pinMode(10, OUTPUT);
14     pinMode(11, OUTPUT);
15 }
16
17 void alloFF(){
18     analogWrite(9,0);
19     analogWrite(10,0);
20     analogWrite(11,0);
21 }
22
```

```
23 void loop(){
24
25     Vout = analogRead(tmp_sensor);
26     V = Vout * 5.0 / 1024.0;
27     tempC = (V - 0.5) / 0.01;//TMP36
28     //tempC = V / 0.01;//LM35
29     Serial.println(tempC);
30     if(tempC>=23) {//RED
31         alloFF();
32         analogWrite(9,255);
33         analogWrite(10,0);
34         analogWrite(11,0);
35     }
36     else if(tempC>=17) {//ORANGE
37         alloFF();
38         analogWrite(9,255);
39         analogWrite(10,128);
40         analogWrite(11,0);
41     }
42     else if(tempC>=10) {//GREEN
43         alloFF();
44         analogWrite(9,0);
45         analogWrite(10,255);
46         analogWrite(11,0);
47     }
48     else {//BLUE
49         alloFF();
50         analogWrite(9,0);
51         analogWrite(10,0);
52         analogWrite(11,255);
53     }
54 }
```

Thank You