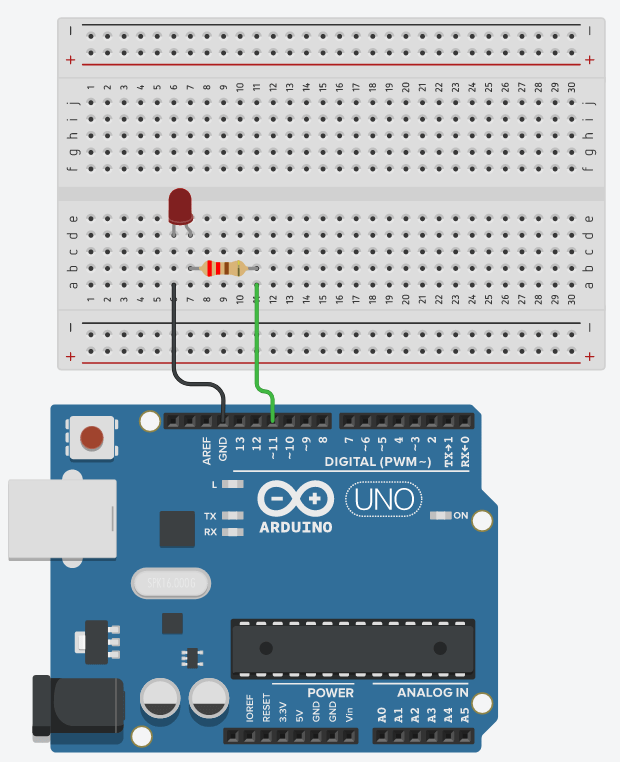
**Experiment No 1 : Blinking LED**

Blinking LED turns on an LED on for one second, then off for one second, repeatedly.

**Sketch:**

****#define LED 11 //Declare LED at pin 11

void setup() {

// put your setup code here, to run once:

pinMode(11, OUTPUT); //Sets pin 11 as output

}

void loop() {

// put your main code here, to run repeatedly:

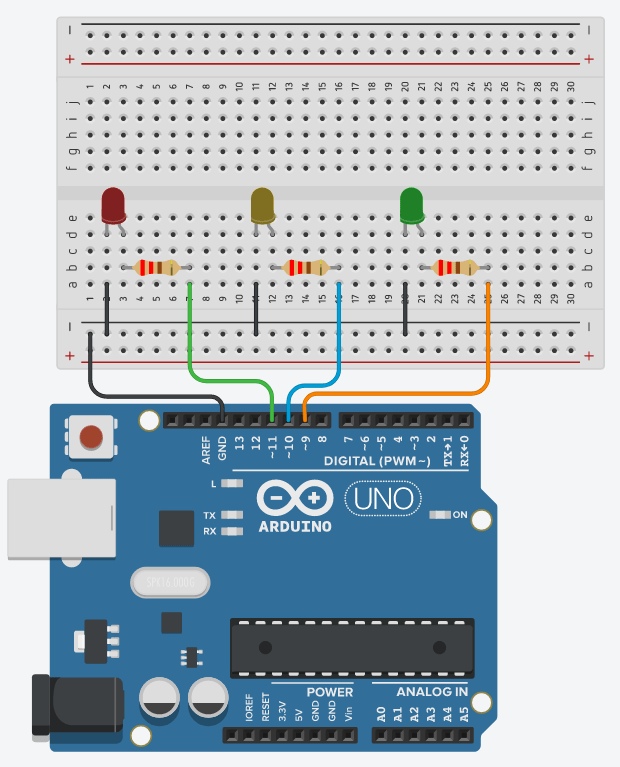
digitalWrite(11, HIGH); //Switch on LED

delay(1000); //pause for 1 second

digitalWrite(11, LOW); //Switch off LED

delay(1000); //pause for 1 second

}

**Experiment No 2 : Traffic Lights LED**

void setup() {

// initialize digital pin LED\_BUILTIN as an output.

pinMode(8, OUTPUT);

pinMode(9, OUTPUT);

pinMode(10, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(8, HIGH); //turn the LED on (HIGH is the voltage level)

delay(3000); // wait for 3 seconds

digitalWrite(8, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for 1 second

digitalWrite(9, HIGH);

delay(3000);

digitalWrite(9, LOW);

delay(1000);

digitalWrite(10, HIGH);

delay(3000);

digitalWrite(10, LOW);

delay(1000);

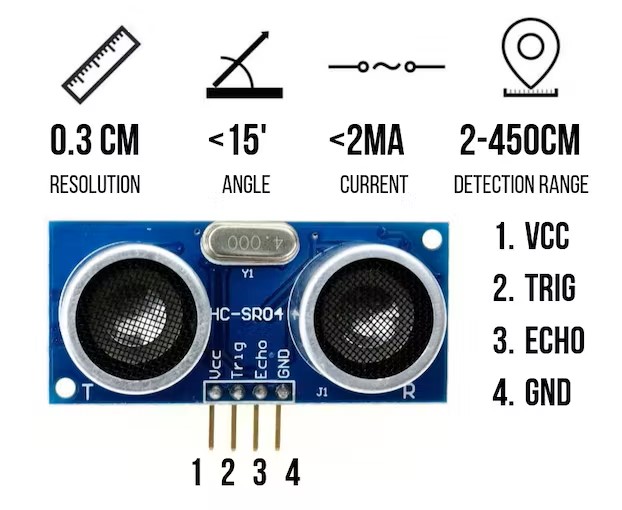
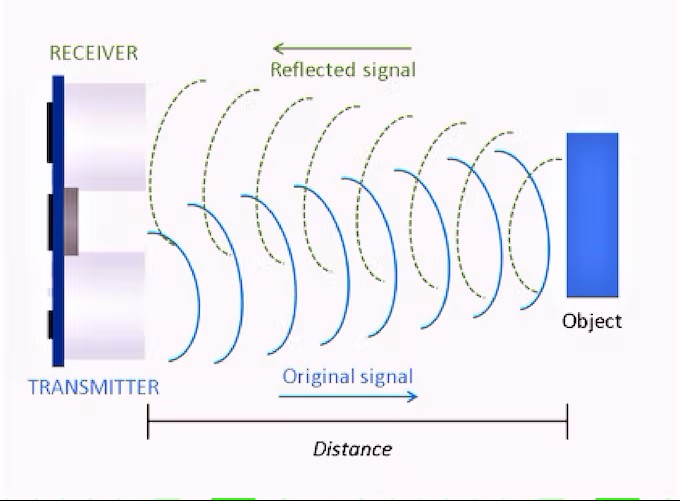
}

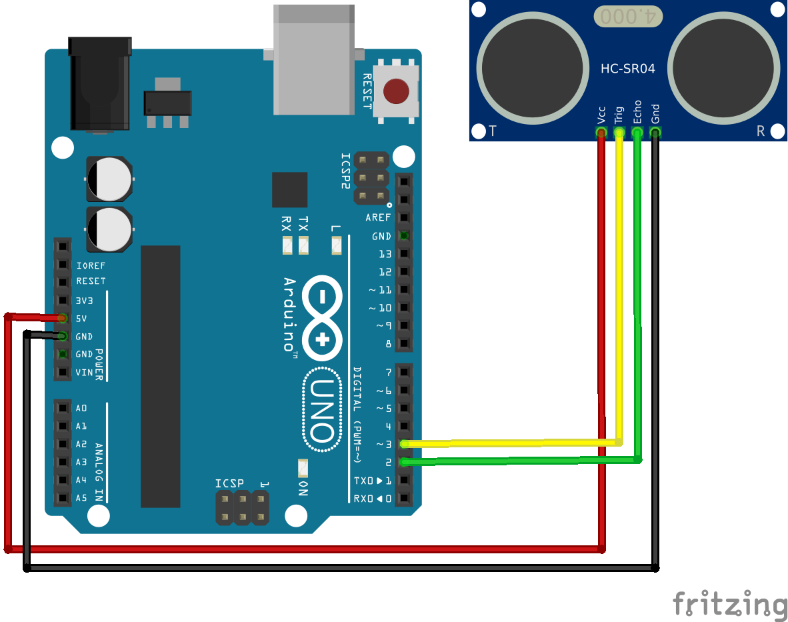
**Experiment No 3 : Ultrasonic Sensor**

Ultrasonic Sensor HC-SR04 is a sensor that can measure **distance**. It emits an **ultrasound**at

**40,000 Hz (40kHz)** which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

The configuration pin of HC-SR04 is VCC (1), TRIG (2), ECHO (3), and GND (4). The **supply voltage** of VCC is **+5V** and you can attach TRIG and ECHO pin to any Digital I/O in your Arduino Board



****

**Sketch:**

int trigger=7; // "trigger on the pin 7"

int echo=6; // "echo" on pin 6.

long time=0; // The value "time" will save the time between transmission and reception of soundwave.

long dist=0; // The value "dist" will save the calculated distance. It will start with 0.

// Instead of int we are using "long" for this value, to save bigger number.

// #define LED 11

void setup() {

/\*Starting the serial communication. It will send the data from the arduino board to the

computer to show it on the serial monitor.\*/

Serial.begin (9600);

pinMode(trigger, OUTPUT); //”trigger” (Pin 7) is an output.

pinMode(echo, INPUT); // ”echo” (Pin 6) is an input.

}

void loop() {

digitalWrite(trigger, LOW); //Low voltage on the trigger pin to produce a clear signal.

delay(5); //for 5 milliseconds.

digitalWrite(trigger, HIGH); //Creating the soundwave.

delay(10); //..for 10 milliseconds.

digitalWrite(trigger, LOW); //Stop creating the soundwave.

/\*With the command pulseIn (Capital “i” in the //front of the “n”) the arduino board measures

the time between sending and receiving the soundwave.\*/

time = pulseIn(echo, HIGH);

/\*This calculation transforms the measured time into the distance in centimeter.

(The sound needs 29.1 milli seconds for one centimeter. The time gets divided with two,

because we only want to get one distance and not the two ways that the soundwave has to take).\*/

dist = (time/2) / 29.1;

if (dist >= 500 || dist <= 0) {

//If the distance gets over 500cm OR under 0cm, the measurement is no longer accurate.

Serial.println("No measurement");

//So the serial monitor displays “No measurement”

}

else {

Serial.print(dist); //The calculated distance is shown on the serial monitor.

Serial.println("cm");

Serial.println(" ");

}

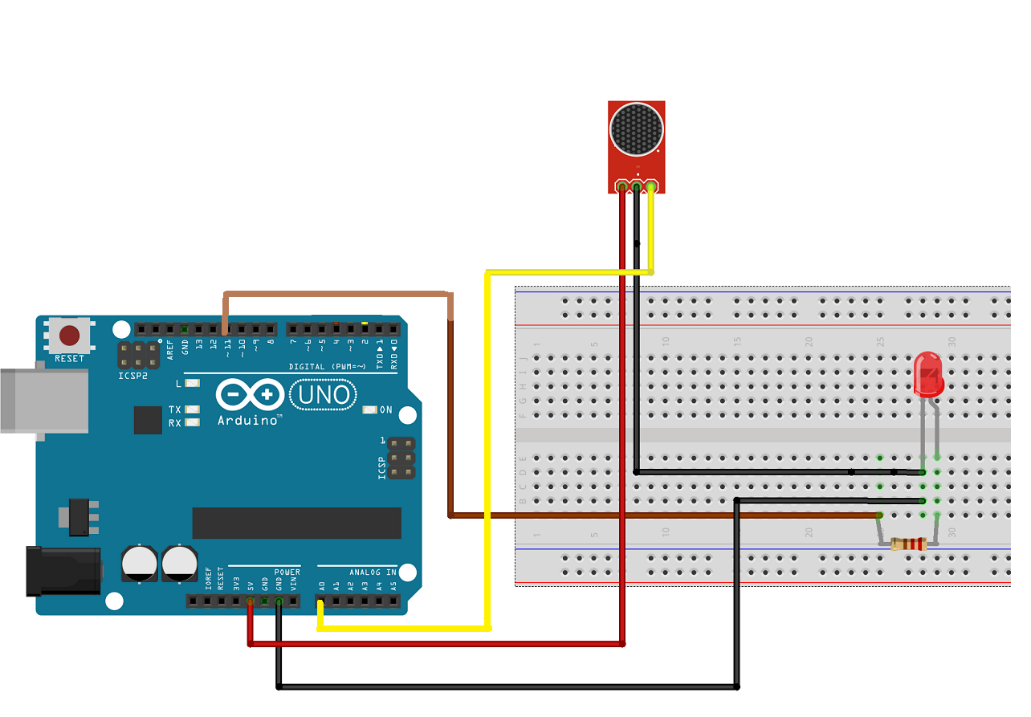
delay(1000); //This command causes a short break between the measurements.

}

**Experiment No 4 : Clap Switch**

The **sound sensor module makes it simple to detect sound and is commonly used to determine sound intensity**. For protection, switching and monitoring applications, this module can be used. It is easy to adjust its precision to ease of use. **It uses a microphone that provides an amplifier, high detector, and buffers for the signal.**

When a sound is detected, **the sensor generates an output signal voltage**, which is then sent to a micro-controller, which performs the required processing.  
The sound detector sensor module for Arduino determines whether or not sound has crossed a predefined threshold value. A microphone detects sound, which is then fed into an LM393 op-amp**. An onboard potentiometer is used to change the sound level set point. As the sound frequency reaches the threshold, an LED on the module illuminates and the output is reduced.**

****

**Sketch:**

const int LED=13;

const int sound=A2;

const int threshold=40;

void setup() {

Serial.begin(9600);

pinMode(LED,OUTPUT);

pinMode(sound,INPUT);

}

void loop() {

int soundsense=analogRead(sound);

if(soundsense>=threshold) {

digitalWrite(LED,HIGH);

delay(1000);

}

else

{

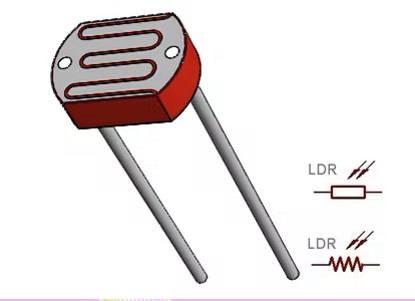
digitalWrite(LED,LOW);

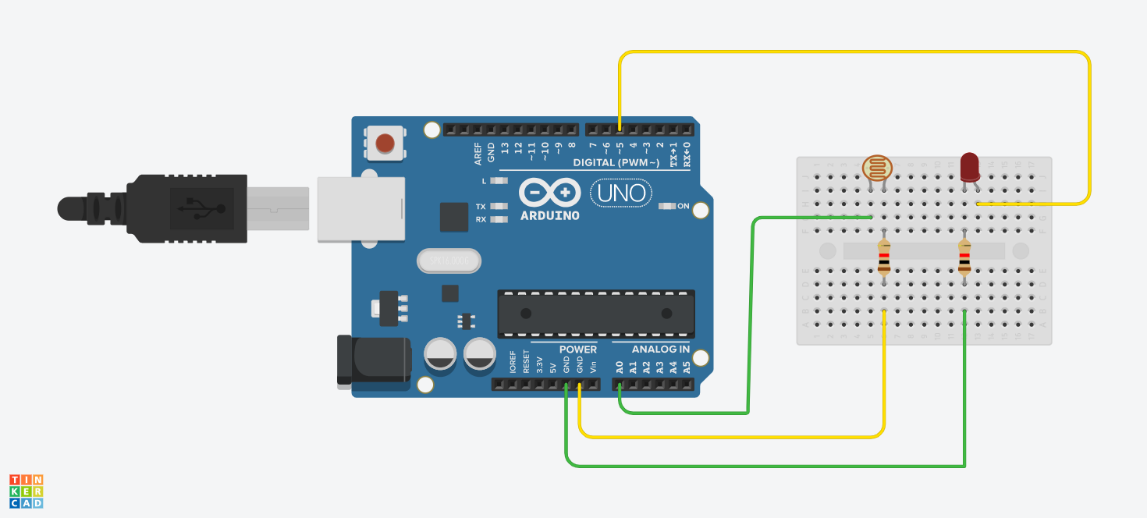
}

}

**Experiment No 5 : LED Toggle LDR**

LDR ( light dependent resistor ) also called photoresistors are responsive to light. Photoresistors are used to indicate the intensity or the presence or the absence of light. When there is darkness the resistance of photoresistor increases and when there is sufficient light it dramatically decreases. LDR which has two terminals. Terminal one is the signal pin which should be connected according to the code. Another terminal is considered as the ground pin which should be connected to the ground of the system.

****

****

Working of the circuit : Basically when there is darkness the led will glow and when there is sufficient light led will stop glowing. This a simple circuit for of interface Arduino uno with LDR sensor.

**m**

**Sketch :**

int ldr=A0;//Set A0(Analog Input) for LDR.

int value=0;

void setup() {

Serial.begin(9600);

pinMode(3,OUTPUT);

}

void loop() {

value=analogRead(ldr);//Reads the Value of LDR(light).

Serial.println("LDR value is :");//Prints the value of LDR to Serial Monitor.

Serial.println(value);

if(value<300)

{

digitalWrite(3,HIGH);//Makes the LED glow in Dark.

}

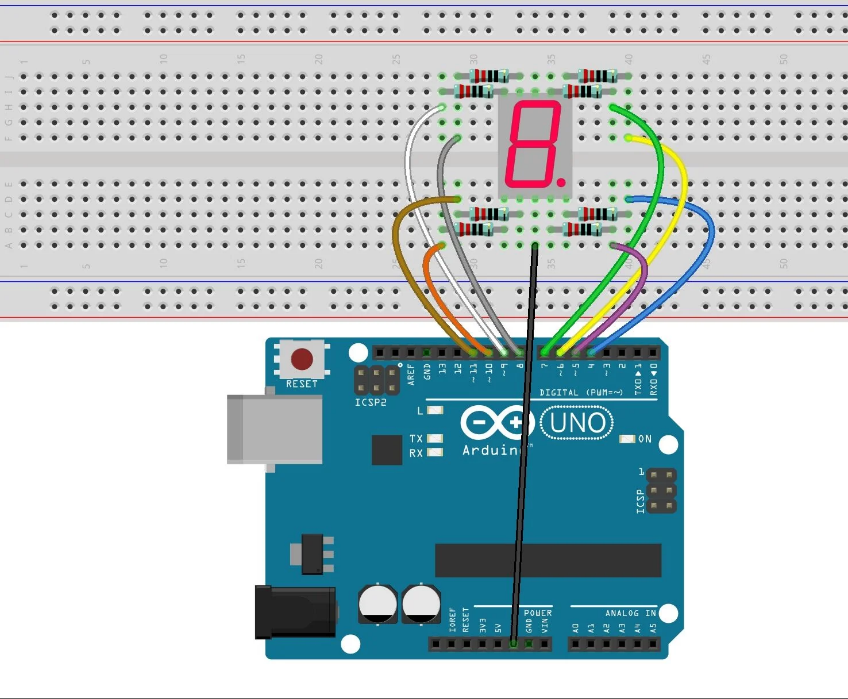
else

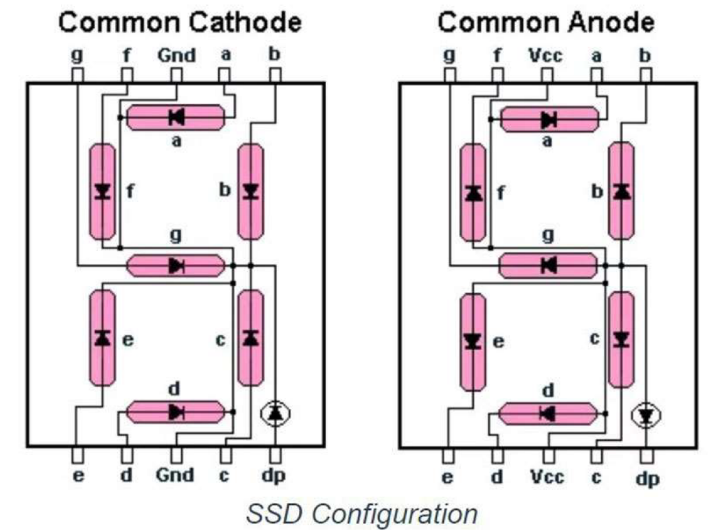
{

digitalWrite(3,LOW);//Turns the LED OFF in Light.

}

}

**Experiment No 6 : Seven Segment Display**

****

// Sketch to display numbers from '0' to '9' using a seven segment LED

#define segA 2 // Connect Segment A to pin number 2

#define segB 3 // Connect Segment B to pin number 3

#define segC 4 // Connect Segment B to pin number 4

#define segD 5 // Connect Segment B to pin number 5

#define segE 6 // Connect Segment B to pin number 6

#define segF 7 // Connect Segment B to pin number 7

#define segG 8 // Connect segment G to pin number 8

#define segDP 9 // Segment DP or Decimal Point is not connected since not used

void setup() {

pinMode(segA,OUTPUT); //Configure all 7 pins from 2 to 8 as output

pinMode(segB,OUTPUT);

pinMode(segC,OUTPUT);

pinMode(segD,OUTPUT);

pinMode(segE,OUTPUT);

pinMode(segF,OUTPUT);

pinMode(segG,OUTPUT);

pinMode(segDP,OUTPUT);

}

void loop() {

// Code to display '0'

digitalWrite(segA,HIGH);

digitalWrite(segB,HIGH);

digitalWrite(segC,HIGH);

digitalWrite(segD,HIGH);

digitalWrite(segE,HIGH);

digitalWrite(segF,HIGH);

digitalWrite(segG,LOW);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '1'

digitalWrite(segA,LOW);

digitalWrite(segB,HIGH);

digitalWrite(segC,HIGH);

digitalWrite(segD,LOW);

digitalWrite(segE,LOW);

digitalWrite(segF,LOW);

digitalWrite(segG,LOW);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '2'

digitalWrite(segA,HIGH);

digitalWrite(segB,HIGH);

digitalWrite(segC,LOW);

digitalWrite(segD,HIGH);

digitalWrite(segE,HIGH);

digitalWrite(segF,LOW);

digitalWrite(segG,HIGH);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '3'

digitalWrite(segA,HIGH);

digitalWrite(segB,HIGH);

digitalWrite(segC,HIGH);

digitalWrite(segD,HIGH);

digitalWrite(segE,LOW);

digitalWrite(segF,LOW);

digitalWrite(segG,HIGH);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '4'

digitalWrite(segA,LOW);

digitalWrite(segB,HIGH);

digitalWrite(segC,HIGH);

digitalWrite(segD,LOW);

digitalWrite(segE,LOW);

digitalWrite(segF,HIGH);

digitalWrite(segG,HIGH);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '5'

digitalWrite(segA,HIGH);

digitalWrite(segB,LOW);

digitalWrite(segC,HIGH);

digitalWrite(segD,HIGH);

digitalWrite(segE,LOW);

digitalWrite(segF,HIGH);

digitalWrite(segG,HIGH);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '6'

digitalWrite(segA,HIGH);

digitalWrite(segB,LOW);

digitalWrite(segC,HIGH);

digitalWrite(segD,HIGH);

digitalWrite(segE,HIGH);

digitalWrite(segF,HIGH);

digitalWrite(segG,HIGH);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '7'

digitalWrite(segA,HIGH);

digitalWrite(segB,HIGH);

digitalWrite(segC,HIGH);

digitalWrite(segD,LOW);

digitalWrite(segE,LOW);

digitalWrite(segF,LOW);

digitalWrite(segG,LOW);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '8'

digitalWrite(segA,HIGH);

digitalWrite(segB,HIGH);

digitalWrite(segC,HIGH);

digitalWrite(segD,HIGH);

digitalWrite(segE,HIGH);

digitalWrite(segF,HIGH);

digitalWrite(segG,HIGH);

digitalWrite(segDP,HIGH);

delay(1000);

// Code to display '9'

digitalWrite(segA,HIGH);

digitalWrite(segB,HIGH);

digitalWrite(segC,HIGH);

digitalWrite(segD,HIGH);

digitalWrite(segE,LOW);

digitalWrite(segF,HIGH);

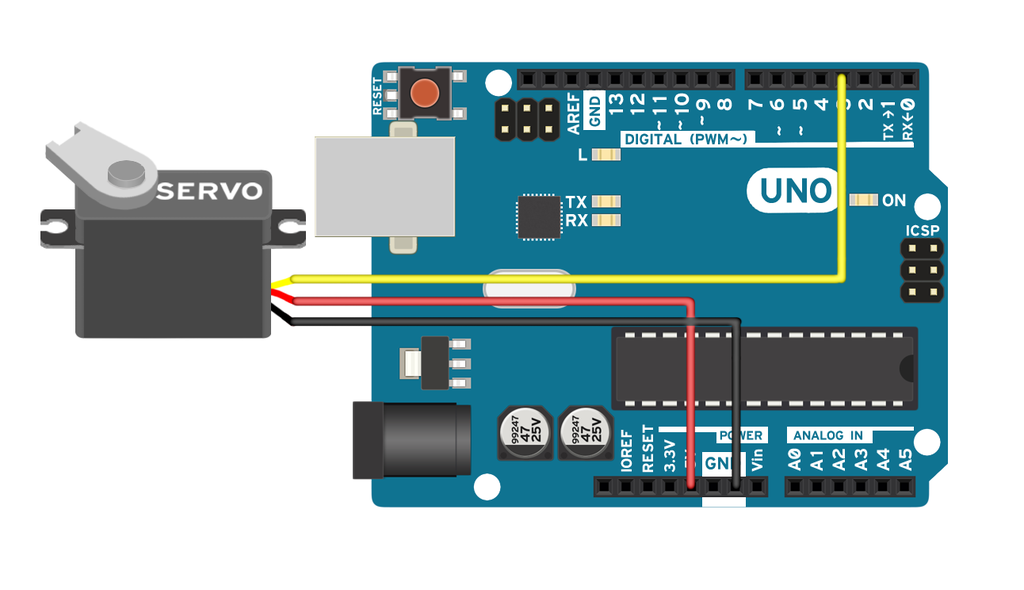
digitalWrite(segG,HIGH);

digitalWrite(segDP,HIGH);

delay(1000);

}

**Experiment No 7 : Servo Motor**



// Include the Servo library

#include<Servo.h>

// Declare the servo pin

int servoPin=3;

// Create a servo object

Servo Servo1;

void setup() {

// We need to attach te servo to the used pin number

Servo1.attach(servoPin);

}

void loop() {

// Make servo go to 0 degrees

Servo1.write(0);

delay(1000);

// Make servo got 90 degrees

Servo1.write(90);

delay(1000);

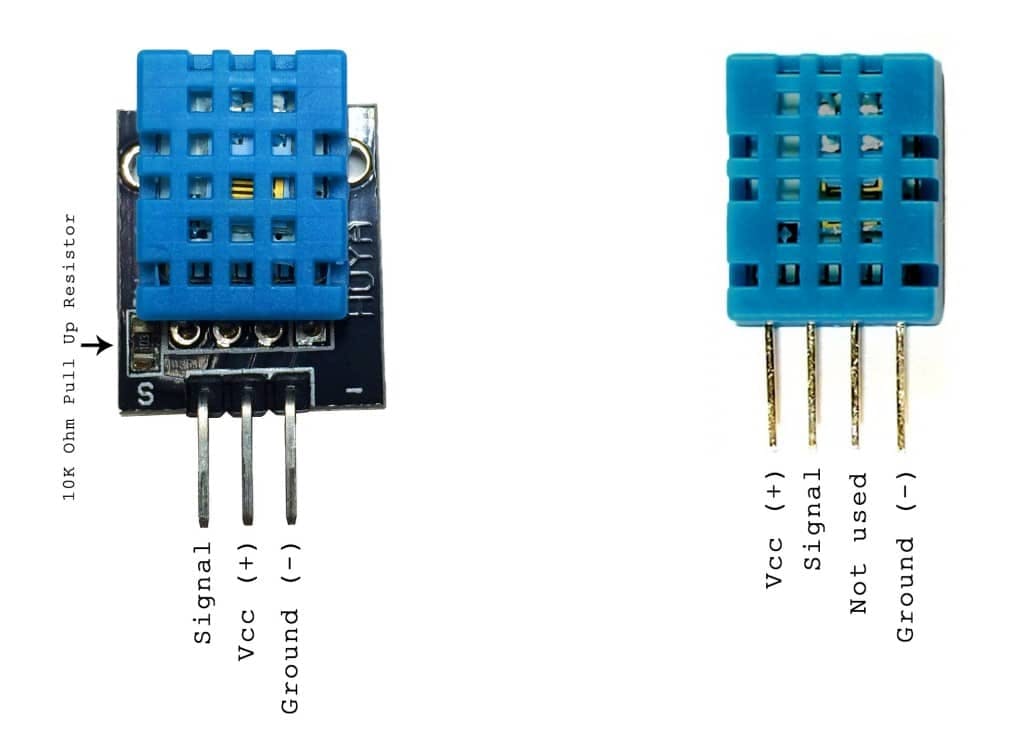
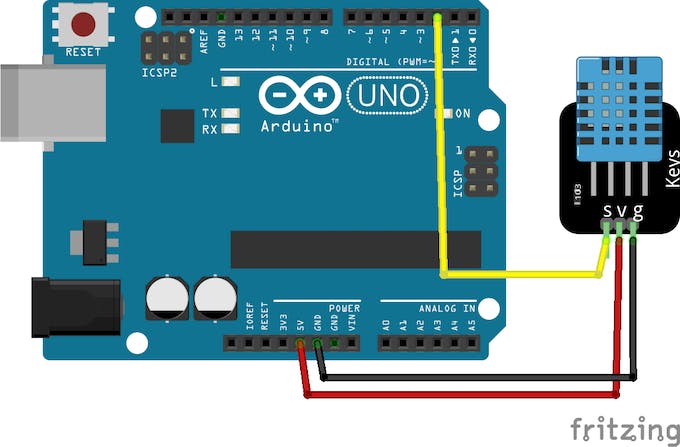
// Make servo got 180 degrees

Servo1.write(180);

delay(1000);

}

**Experiment No 8 : DHT Sensor**



#include "DHT.h"

#define DHTPIN 2 // Digital pin connected to the DHT sensor

#define DHTTYPE DHT11 // DHT 11

//#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(9600);

Serial.println(F("DHTxx test!"));

dht.begin();

}

void loop() {

// Wait a few seconds between measurements.

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) ) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

Serial.println(F(" Humidity: "));

Serial.print(h);

Serial.println(F("% Temperature: "));

Serial.print(t);

Serial.println(F("F"));

}