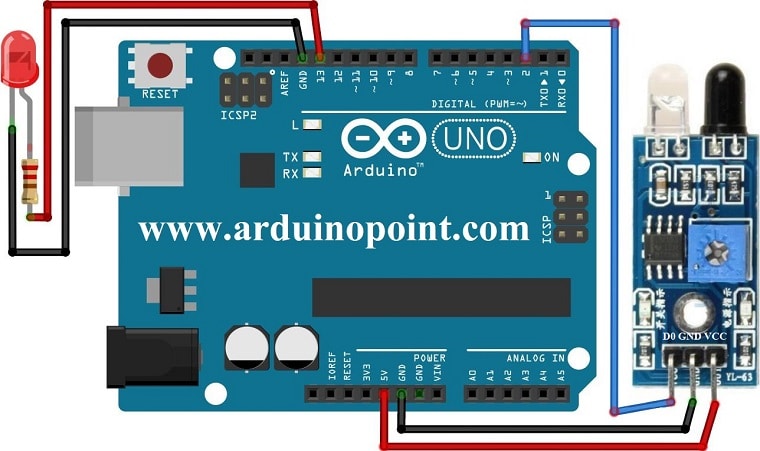
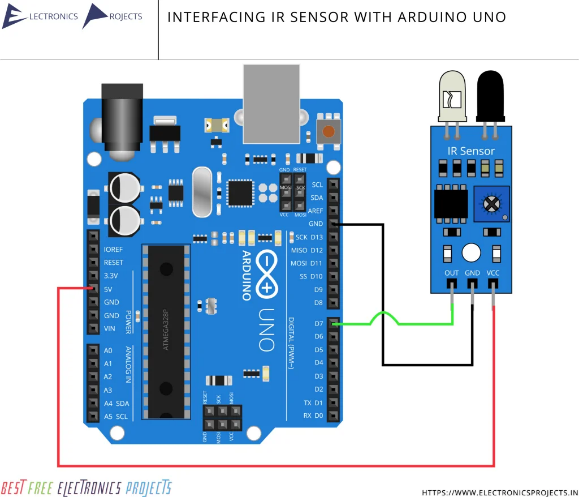
**1)IR SENSOR**

**DIGITAL**

const int irSensorPin = 2; // Digital pin where the IR sensor DO is connected

const int ledPin = 13; // Digital pin where the LED is connected

void setup()

{

pinMode(irSensorPin, INPUT); // Set IR sensor pin as input

pinMode(ledPin, OUTPUT); // Set LED pin as output (optional)

Serial.begin(9600); // Start serial communication

}

void loop() {

int obstacleDetected = digitalRead(irSensorPin);

if (obstacleDetected == HIGH) {

Serial.println("Obstacle detected!");

digitalWrite(ledPin, HIGH); // Turn LED on if an obstacle is detected

} else {

Serial.println("No obstacle detected.");

digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

delay(1000); // Wait 1 second before the next reading

}

ANALOG

const int irSensorPin = A0; // Analog pin where the IR sensor AO is connected

const int ledPin = 13; // Digital pin where LED is connected (optional)

const int distanceThreshold = 200; // Example threshold for distance measurement (adjust as needed)

void setup() {

pinMode(irSensorPin, INPUT); // Set IR sensor pin as input

pinMode(ledPin, OUTPUT); // Set LED pin as output (optional)

Serial.begin(9600); // Start serial communication

}

void loop() {

int sensorValue = analogRead(irSensorPin);

Serial.print("Distance Sensor Value: ");

Serial.println(sensorValue);

if (sensorValue < distanceThreshold) {

Serial.println("Object detected within threshold!");

digitalWrite(ledPin, HIGH); // Turn LED on if the object is within threshold distance

} else {

Serial.println("Object detected beyond threshold.");

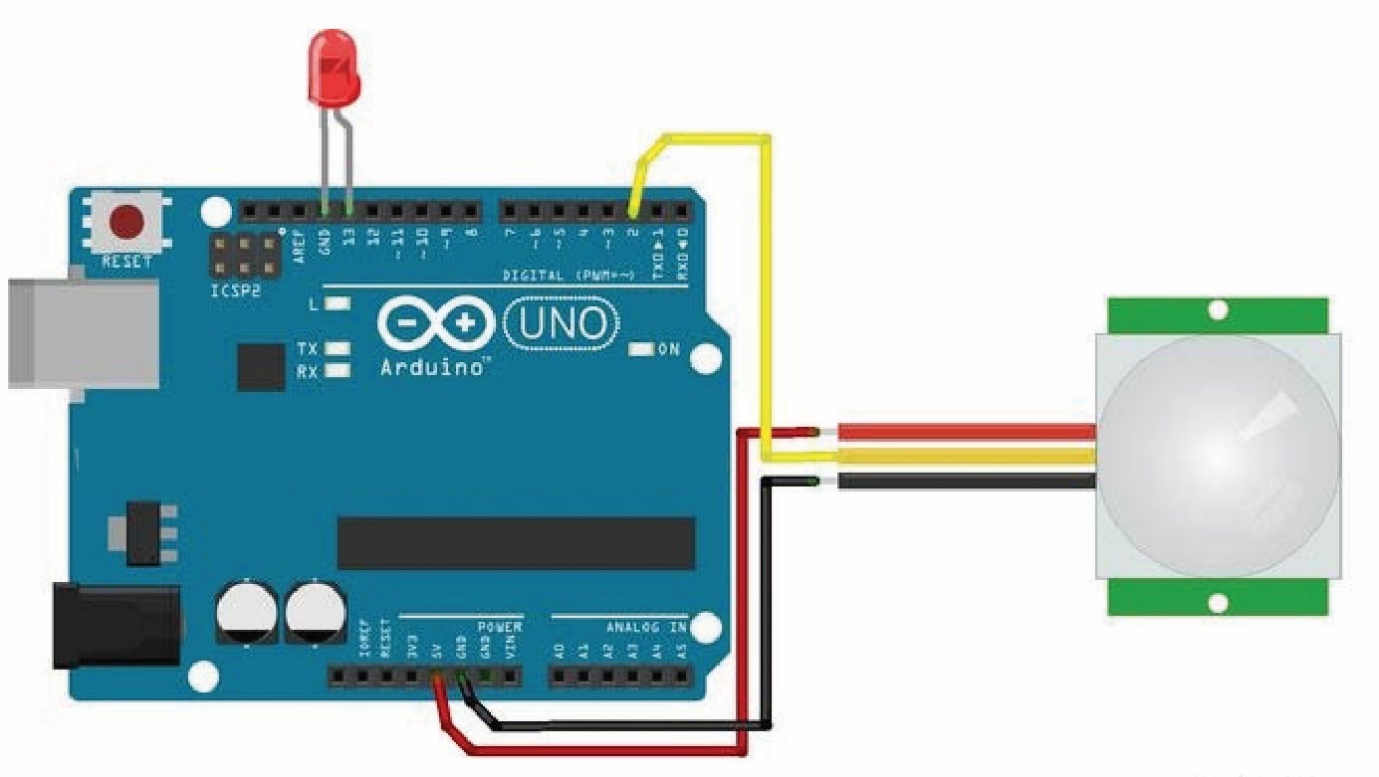
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

delay(1000); // Wait 1 second before the next reading

}

**2)PIR SENSOR**



const int pirSensorPin = 2; // Digital pin where the PIR sensor OUT is connected

const int ledPin = 13; // Digital pin where the LED is connected (optional)

void setup() {

pinMode(pirSensorPin, INPUT); // Set PIR sensor pin as input

pinMode(ledPin, OUTPUT); // Set LED pin as output (optional)

Serial.begin(9600); // Start serial communication

}

void loop() {

int motionDetected = digitalRead(pirSensorPin); // Read the digital value from the PIR sensor

if (motionDetected == HIGH) {

Serial.println("Motion detected!");

digitalWrite(ledPin, HIGH); // Turn LED on if motion is detected

} else {

Serial.println("No motion detected.");

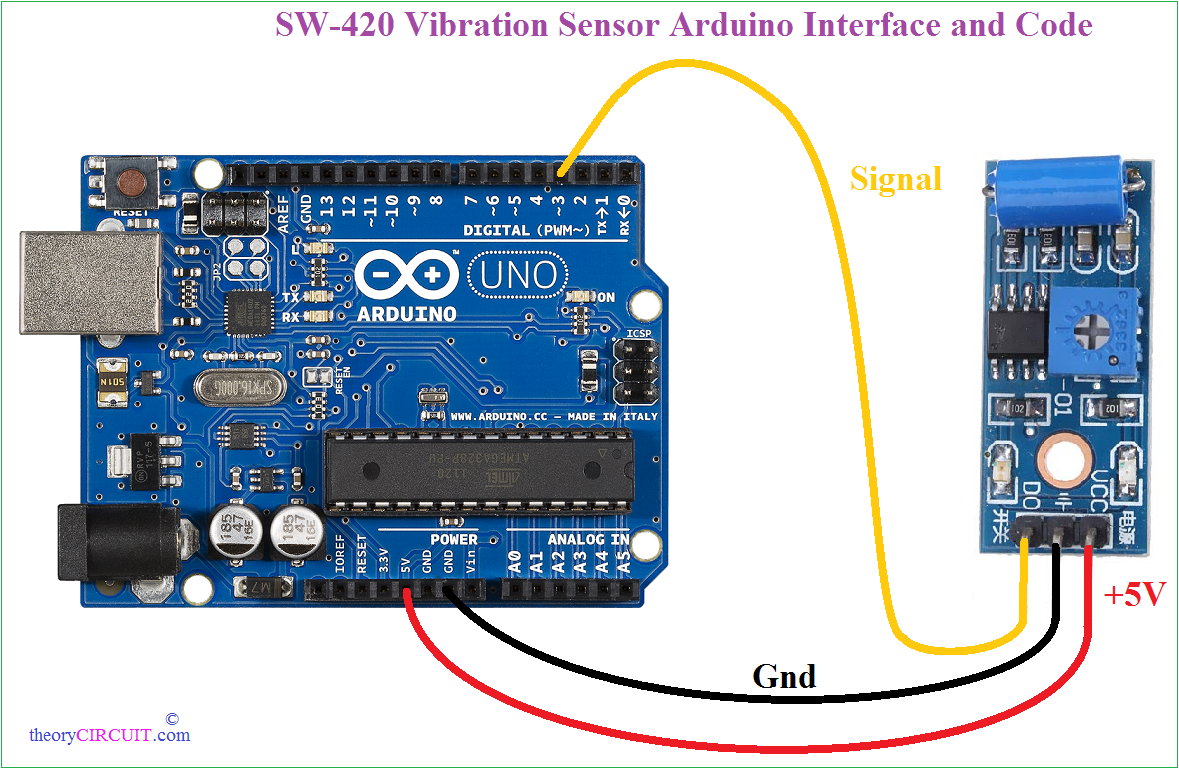
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

delay(1000); // Wait 1 second before the next reading

}

**3)VIBRATION SENSOR**



const int vibrationSensorPin = 2; // Digital pin where the SW-420 DO is connected

const int ledPin = 13; // Digital pin where the LED is connected (optional)

void setup() {

pinMode(vibrationSensorPin, INPUT); // Set vibration sensor pin as input

pinMode(ledPin, OUTPUT); // Set LED pin as output (optional)

Serial.begin(9600); // Start serial communication

}

void loop() {

int vibrationDetected = digitalRead(vibrationSensorPin); // Read the digital value from the vibration sensor

if (vibrationDetected == HIGH) {

Serial.println("Vibration detected!");

digitalWrite(ledPin, HIGH); // Turn LED on if vibration is detected

} else {

Serial.println("No vibration detected.");

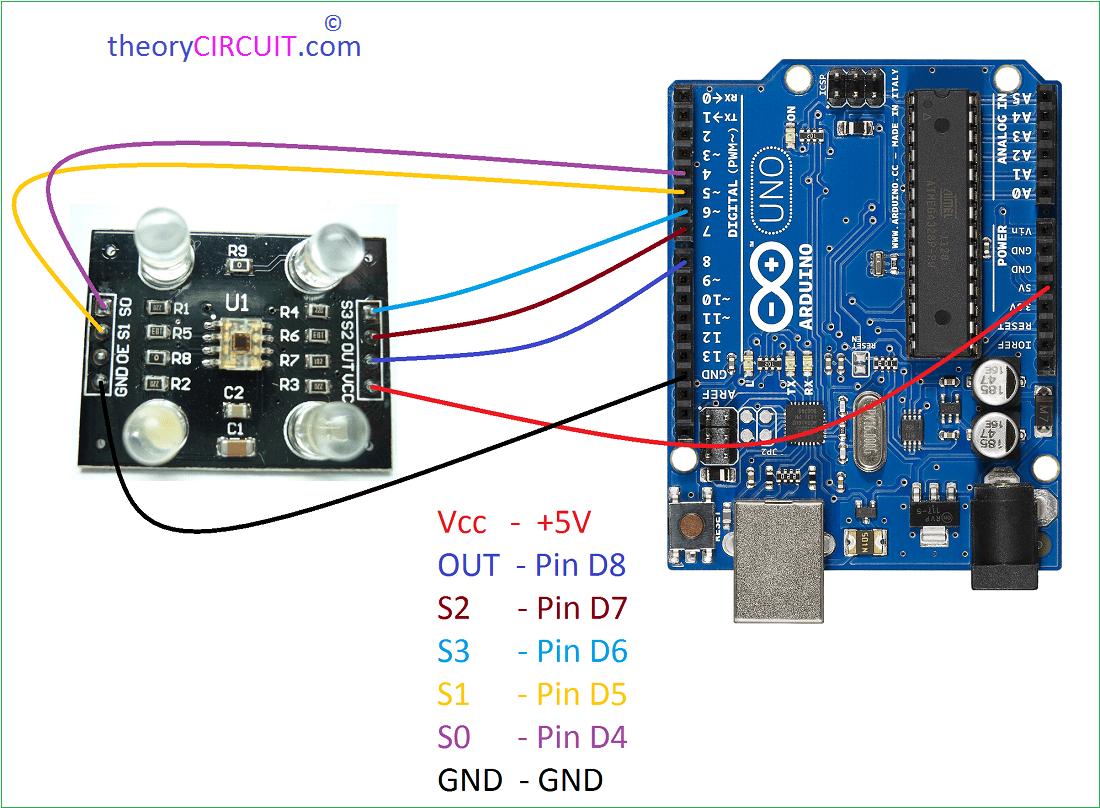
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

delay(1000); // Wait 1 second before the next reading

}

**4)COLOUR SENSOR**



// Pin definitions

const int S0 = 4; // Frequency scaling control pin S0

const int S1 = 5; // Frequency scaling control pin S1

const int S2 = 6; // Color filter control pin S2

const int S3 = 7; // Color filter control pin S3

const int sensorOut = 8; // Sensor output pin

// Variables to store the color frequency

int redFrequency = 0;

int greenFrequency = 0;

int blueFrequency = 0;

void setup() {

// Initialize pins as input or output

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

pinMode(sensorOut, INPUT);

// Initialize serial communication

Serial.begin(9600);

// Set frequency scaling to 20% (S0 HIGH, S1 LOW)

digitalWrite(S0, HIGH);

digitalWrite(S1, LOW);

}

void loop() {

// Read Red

digitalWrite(S2, LOW);

digitalWrite(S3, LOW);

redFrequency = pulseIn(sensorOut, LOW); // Measure duration of the LOW pulse

Serial.print("Red: ");

Serial.print(redFrequency);

Serial.print(" ");

// Read Green

digitalWrite(S2, HIGH);

digitalWrite(S3, HIGH);

greenFrequency = pulseIn(sensorOut, LOW); // Measure duration of the LOW pulse

Serial.print("Green: ");

Serial.print(greenFrequency);

Serial.print(" ");

// Read Blue

digitalWrite(S2, LOW);

digitalWrite(S3, HIGH);

blueFrequency = pulseIn(sensorOut, LOW); // Measure duration of the LOW pulse

Serial.print("Blue: ");

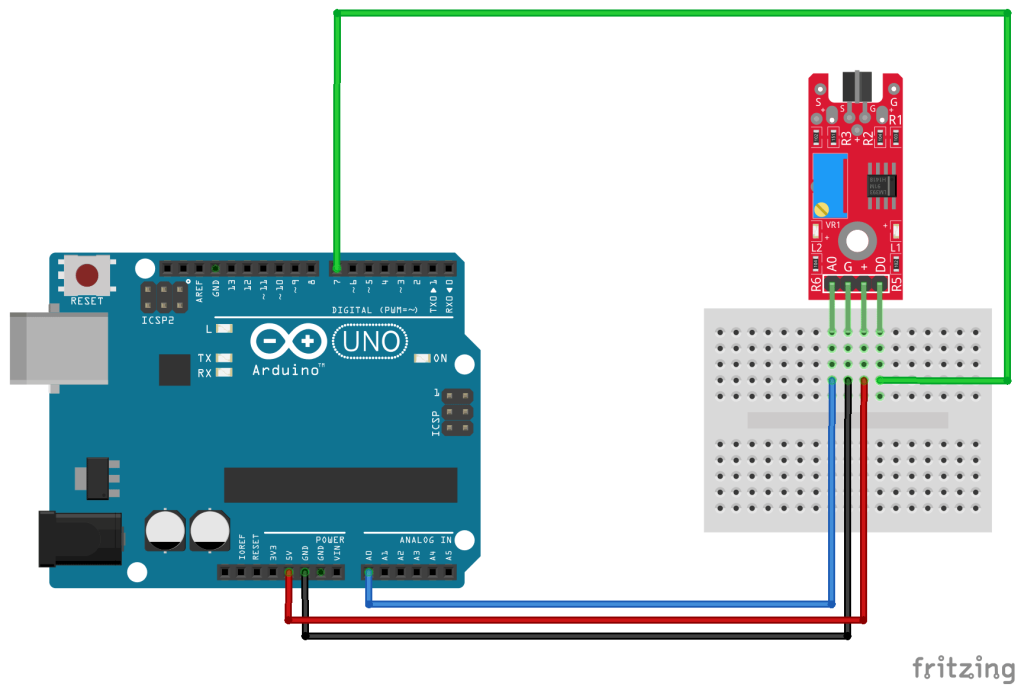
Serial.println(blueFrequency);

// Add a delay for readability

delay(500);

}

**5)TOUCH SENSOR**



const int touchSensorPin = 2; // Digital pin where the touch sensor OUT is connected

const int ledPin = 13; // Digital pin where the LED is connected (optional)

void setup() {

pinMode(touchSensorPin, INPUT); // Set touch sensor pin as input

pinMode(ledPin, OUTPUT); // Set LED pin as output (optional)

Serial.begin(9600); // Start serial communication

}

void loop() {

int touchState = digitalRead(touchSensorPin); // Read the digital value from the touch sensor

if (touchState == HIGH) {

Serial.println("Touched!");

digitalWrite(ledPin, HIGH); // Turn LED on if the sensor is touched

} else {

Serial.println("Not touched.");

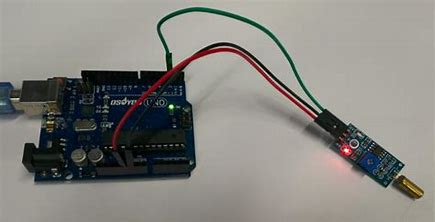
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

delay(100); // Short delay to debounce the sensor

}

**6)TILT SENSOR**



const int tiltSensorPin = A0; // Analog pin where the tilt sensor OUT is connected

const int ledPin = 13; // Digital pin where the LED is connected (optional)

void setup() {

pinMode(tiltSensorPin, INPUT); // Set tilt sensor pin as input

pinMode(ledPin, OUTPUT); // Set LED pin as output (optional)

Serial.begin(9600); // Start serial communication

}

void loop() {

int sensorValue = analogRead(tiltSensorPin); // Read the analog value from the tilt sensor

// Convert the analog value to a tilt angle or use it directly for threshold comparisons

int tiltAngle = map(sensorValue, 0, 1023, 0, 180); // Adjust mapping based on sensor characteristics

Serial.print("Tilt Angle: "); Serial.println(tiltAngle);

// Example threshold for turning LED on

if (tiltAngle > 90) {

Serial.println("Tilted!");

digitalWrite(ledPin, HIGH); // Turn LED on if tilt angle exceeds threshold

} else {

Serial.println("Not tilted.");

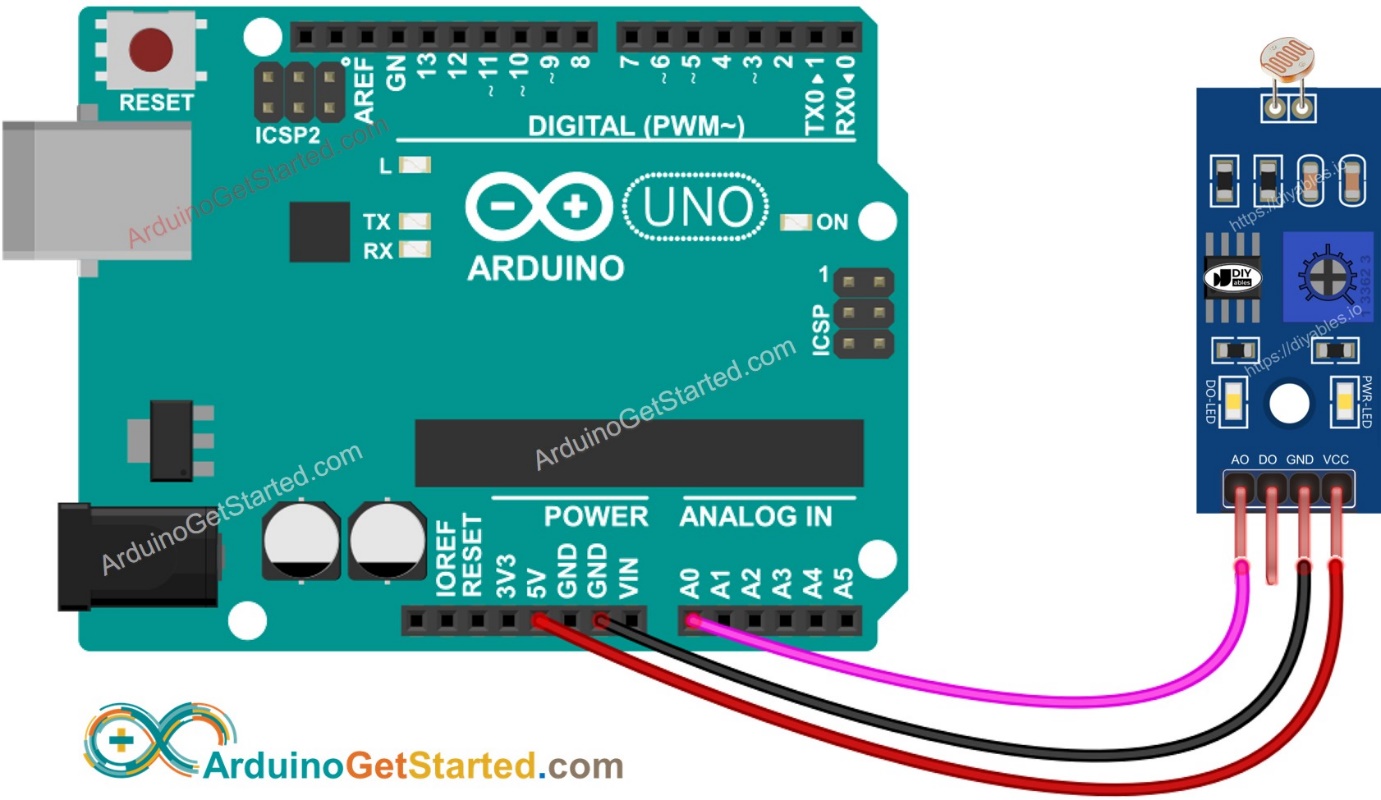
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

delay(100); // Short delay

}

**7)LDR SENSOR**



const int ldrPin = A0; // Analog pin where the LDR output is connected

const int ledPin = 13; // Digital pin where the LED is connected (optional)

void setup() {

pinMode(ldrPin, INPUT); // Set LDR pin as input

pinMode(ledPin, OUTPUT); // Set LED pin as output (optional)

Serial.begin(9600); // Start serial communication

}

void loop() {

int sensorValue = analogRead(ldrPin); // Read the analog value from the LDR

Serial.print("LDR Value: ");

Serial.println(sensorValue); // Print the LDR value to the Serial Monitor

// Example threshold for turning LED on

if (sensorValue < 500) { // Adjust the threshold as needed

Serial.println("Low light detected!");

digitalWrite(ledPin, HIGH); // Turn LED on if light level is below threshold

} else {

Serial.println("Sufficient light detected.");

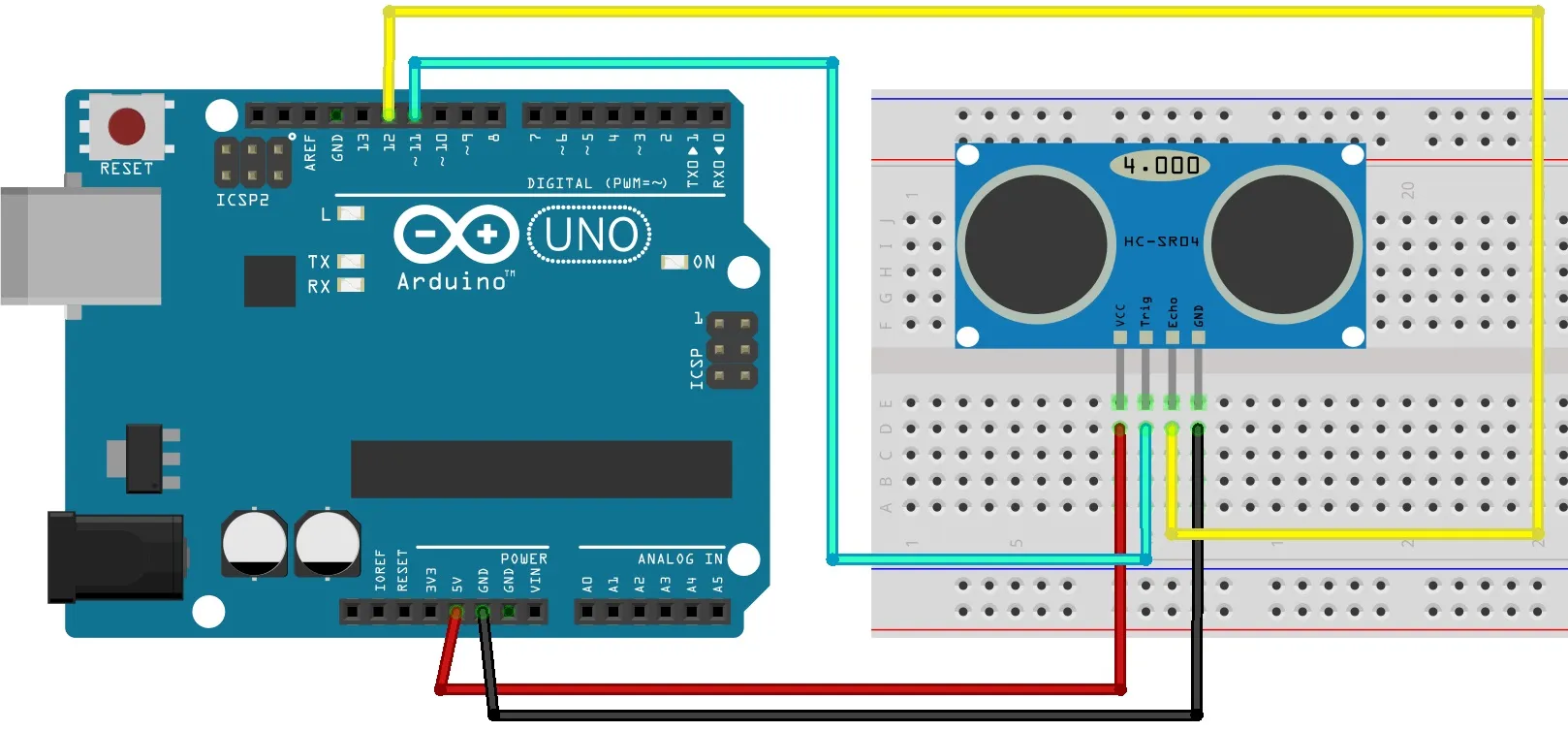
digitalWrite(ledPin, LOW); // Turn LED off if light level is above threshold

}

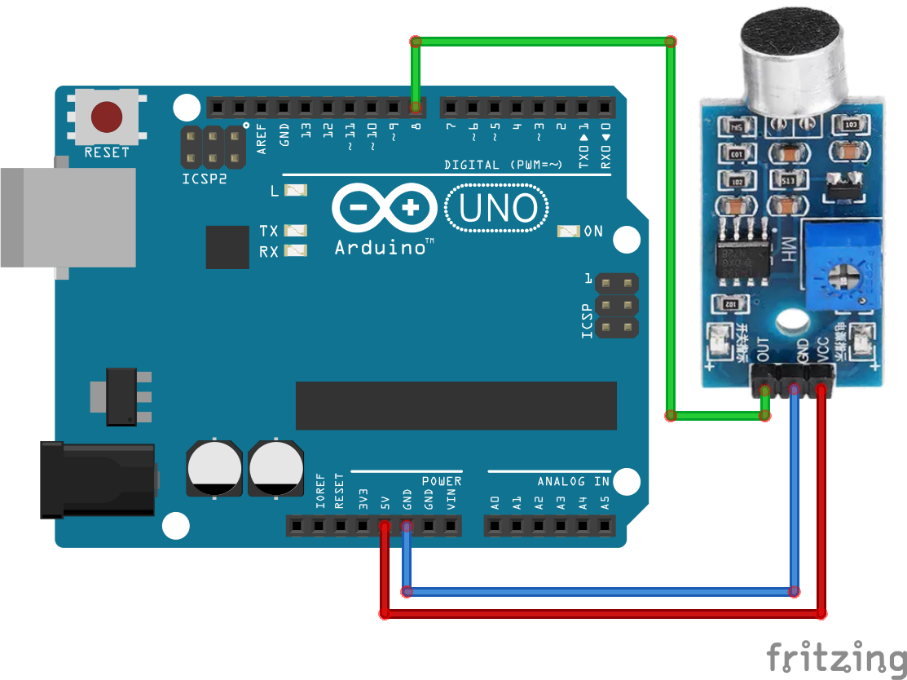
delay(1000); // Wait 1 second before the next reading

}

**8)US SENSOR**



**9)SOUND SENSOR**



const int soundSensorPin = 2; // Digital pin where the sound sensor OUT is connected

const int ledPin = 13; // Digital pin where the LED is connected (optional)

void setup() {

pinMode(soundSensorPin, INPUT); // Set sound sensor pin as input

pinMode(ledPin, OUTPUT); // Set LED pin as output (optional)

Serial.begin(9600); // Start serial communication

}

void loop() {

int soundState = digitalRead(soundSensorPin); // Read the digital value from the sound sensor

if (soundState == HIGH) {

Serial.println("Sound detected!");

digitalWrite(ledPin, HIGH); // Turn LED on if sound is detected

} else {

Serial.println("No sound.");

digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

delay(100); // Short delay before the next reading

}