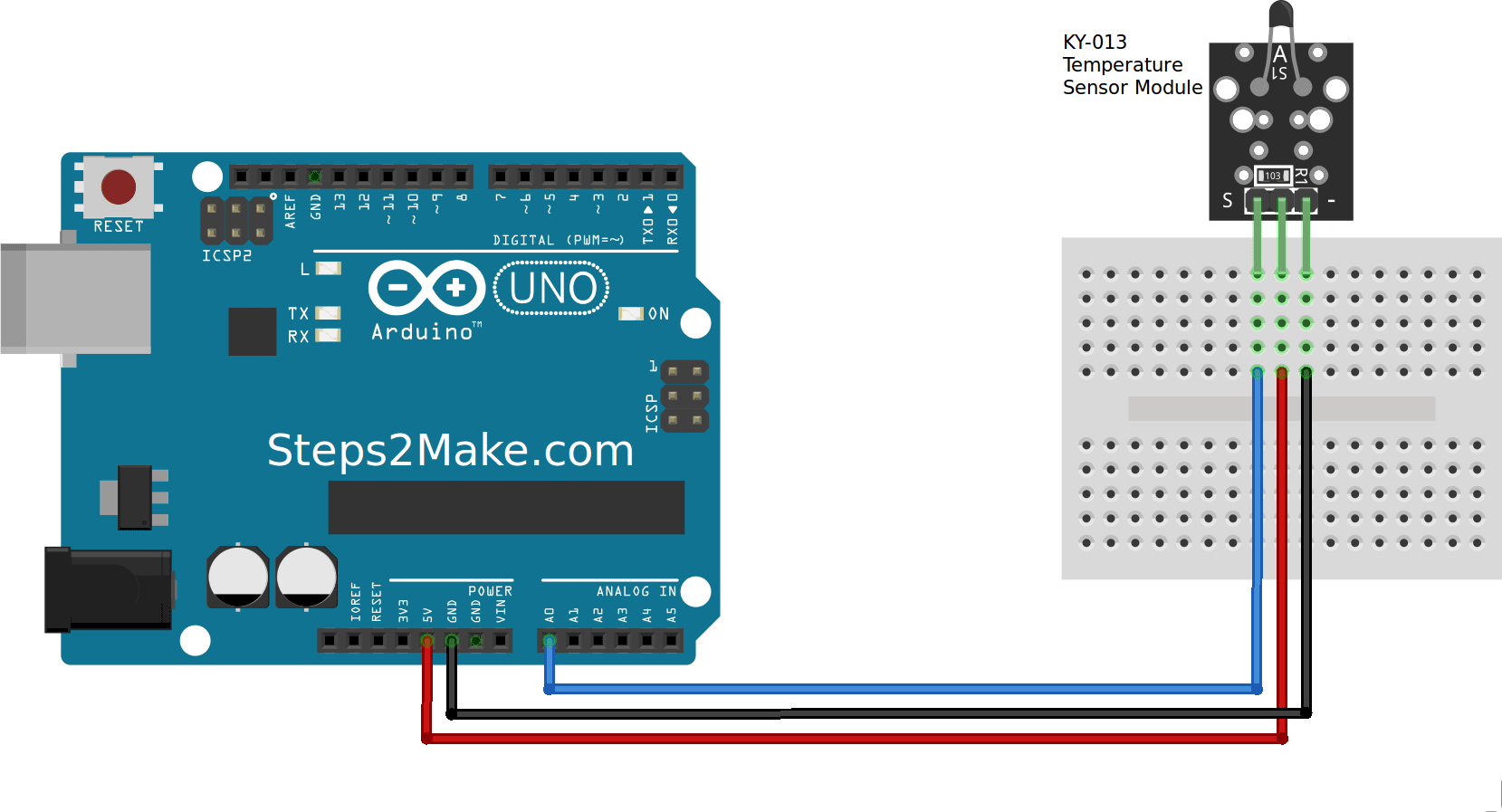
**1)TEMPERATURE SENSOR(LM35 Temperature Sensor)**



VCC (Pin 1): Connect to the 5V pin on the Arduino Mega.

GND (Pin 2): Connect to the GND pin on the Arduino Mega.

OUT (Pin 3): Connect to one of the analog input pins on the Arduino Mega (e.g., A0).

LED:

Anode (longer leg): Connect to a digital output pin on the Arduino Mega (e.g., D13).

Cathode (shorter leg): Connect to one end of a 220Ω resistor. Connect the other end of the resistor to the GND pin on the Arduino Mega.

const int tempPin = A0; // Analog pin where LM35 is connected

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

const float tempThreshold = 30.0; // Temperature threshold in Celsius

void setup() {

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

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

}

void loop() {

int sensorValue = analogRead(tempPin);

float voltage = sensorValue \* (5.0 / 1023.0); // Convert analog reading to voltage

float temperatureC = voltage \* 100.0; // Convert voltage to temperature in Celsius

Serial.print("Temperature: ");

Serial.print(temperatureC);

Serial.println(" C");

if (temperatureC > tempThreshold) {

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

} else {

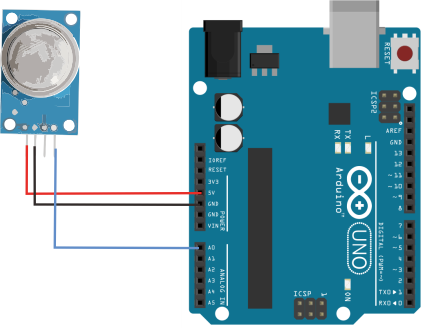
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

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

}

**2)** **MQ-6 Sensor Connections:(LPG)**



VCC (Pin 1): Connect to the 5V pin on the Arduino Mega.

GND (Pin 2): Connect to the GND pin on the Arduino Mega.

AO (Analog Output, Pin 3): Connect to one of the analog input pins on the Arduino Mega (e.g., A0).

DO (Digital Output, Pin 4): Optional, connect to a digital input pin on the Arduino Mega (e.g., D2).

LED (Optional):

Anode (longer leg): Connect to a digital output pin on the Arduino Mega (e.g., D13).

Cathode (shorter leg): Connect to one end of a 220Ω resistor. Connect the other end of the resistor to the GND pin on the Arduino Mega.

const int gasPin = A0; // Analog pin where MQ-6 AO is connected

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

const int gasThreshold = 400; // Threshold value for gas detection (adjust as needed)

void setup() {

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

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

}

void loop() {

int gasValue = analogRead(gasPin); // Read the analog value from MQ-6 sensor

Serial.print("Gas Value: ");

Serial.println(gasValue);

if (gasValue > gasThreshold) {

digitalWrite(ledPin, HIGH); // Turn LED on if gas concentration exceeds threshold

} else {

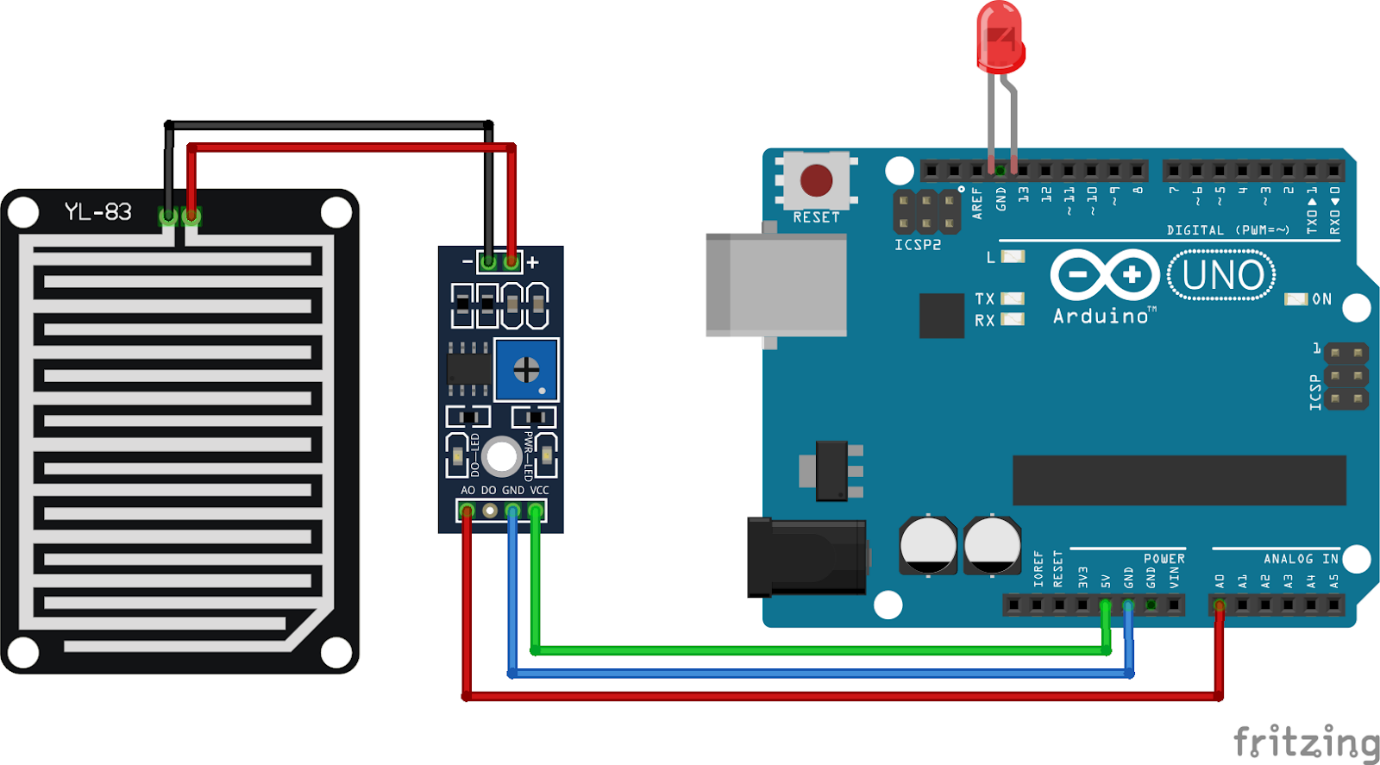
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

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

}

**3)RAINDROP SENSOR**



const int rainPin = A0; // Analog pin where raindrop sensor AO is connected

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

const int rainThreshold = 500; // Threshold value for rain detection (adjust as needed)

void setup() {

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

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

}

void loop() {

int rainValue = analogRead(rainPin); // Read the analog value from raindrop sensor

Serial.print("Rain Value: ");

Serial.println(rainValue);

if (rainValue < rainThreshold) { // Lower value indicates more rain

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

} else {

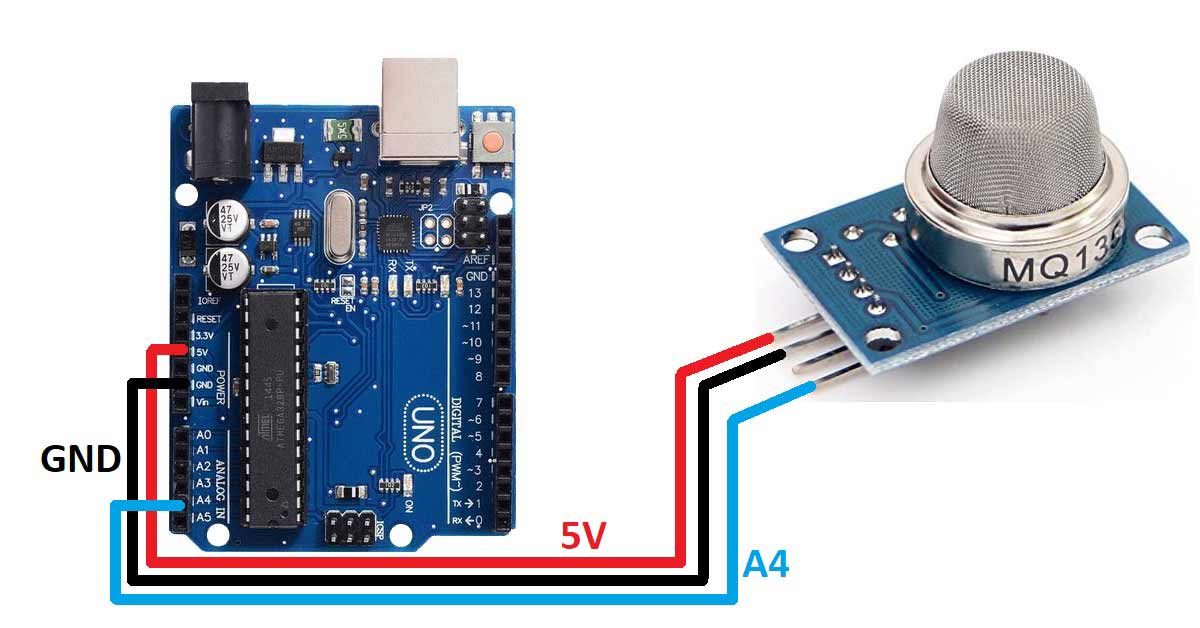
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

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

}

**4)AIRQUALITY SENSOR**



const int airQualityPin = A0; // Analog pin where MQ-135 AO is connected

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

const int airQualityThreshold = 400; // Adjust this threshold based on calibration

void setup() {

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

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

}

void loop() {

int Value = analogRead(airQualityPin); // Read the analog value from MQ-135 sensor

Serial.print("Air Quality Value: ");

Serial.println(Value);

if (Value > airQualityThreshold) {

digitalWrite(ledPin, HIGH); // Turn LED on if air quality is below threshold

Serial.println("POOR AIR QUALITY);

} else {

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

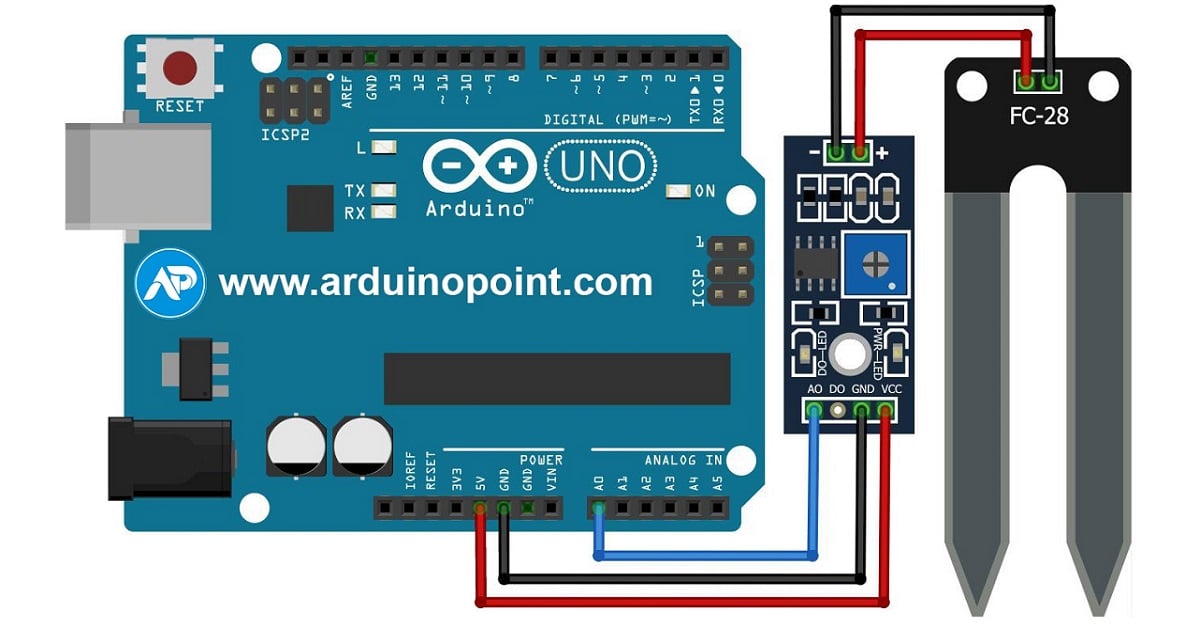
Serial.println("GOOD AIR QUALITY);

}

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

}

**5)SOIL MOISTURE SENSOR**



const int soilMoisturePin = A0; // Analog pin where soil moisture sensor AO is connected

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

const int moistureThreshold = 400; // Threshold value for soil moisture (adjust as needed)

void setup() {

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

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

}

void loop() {

int moistureValue = analogRead(soilMoisturePin); // Read the analog value from the soil moisture sensor

Serial.print("Soil Moisture Value: ");

Serial.println(moistureValue);

if (moistureValue < moistureThreshold) { // Lower value indicates higher moisture

digitalWrite(ledPin, HIGH); // Turn LED on if soil is dry

} else {

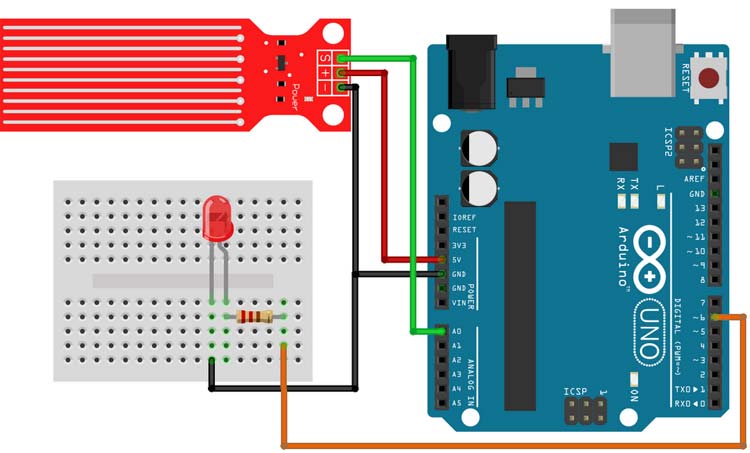
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

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

}

**6)WATER LEVEL SENSOR**



const int waterLevelPin = A0; // Analog pin where water level sensor AO is connected

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

const int waterLevelThreshold = 500; // Threshold value for water level (adjust as needed)

void setup() {

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

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

}

void loop() {

int waterLevelValue = analogRead(waterLevelPin); // Read the analog value from the water level sensor

Serial.print("Water Level Value: ");

Serial.println(waterLevelValue);

if (waterLevelValue > waterLevelThreshold) {

digitalWrite(ledPin, HIGH); // Turn LED on if water level is above threshold

} else {

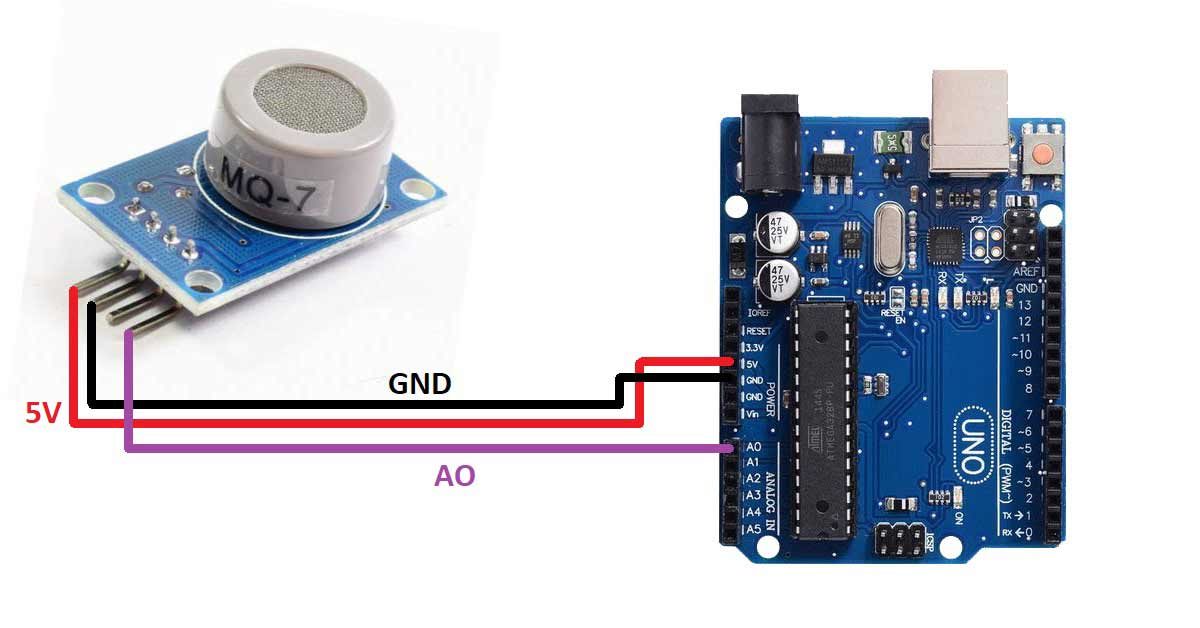
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

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

}

**7)CO SENSOR**



const int coSensorPin = A0; // Analog pin where MQ-7 AO is connected

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

const int coThreshold = 400; // Threshold value for CO detection (adjust as needed)

void setup() {

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

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

}

void loop() {

int coValue = analogRead(coSensorPin); // Read the analog value from MQ-7 sensor

Serial.print("CO Concentration Value: ");

Serial.println(coValue);

if (coValue > coThreshold) {

digitalWrite(ledPin, HIGH); // Turn LED on if CO concentration exceeds threshold

} else {

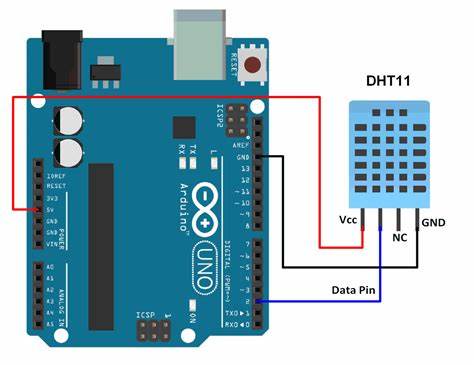
digitalWrite(ledPin, LOW); // Turn LED off otherwise

}

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

}

**8)HUMIDITY SENSOR DHT11**



#include <DHT.h>

#define DHTPIN 2 // Pin connected to the DATA pin of the DHT11

#define DHTTYPE DHT11 // Define the type of sensor (DHT11)

DHT dht(DHTPIN, DHTTYPE);

void setup() {

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

dht.begin(); // Initialize the DHT11 sensor

}

void loop() {

delay(2000);

float humidity = dht.readHumidity();

float temperature = dht.readTemperature();

if (isnan(humidity) || isnan(temperature))

{

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

return;

}

// Print the results to the Serial Monitor

Serial.print("Humidity: ");

Serial.print(humidity);

Serial.print(" %\t");

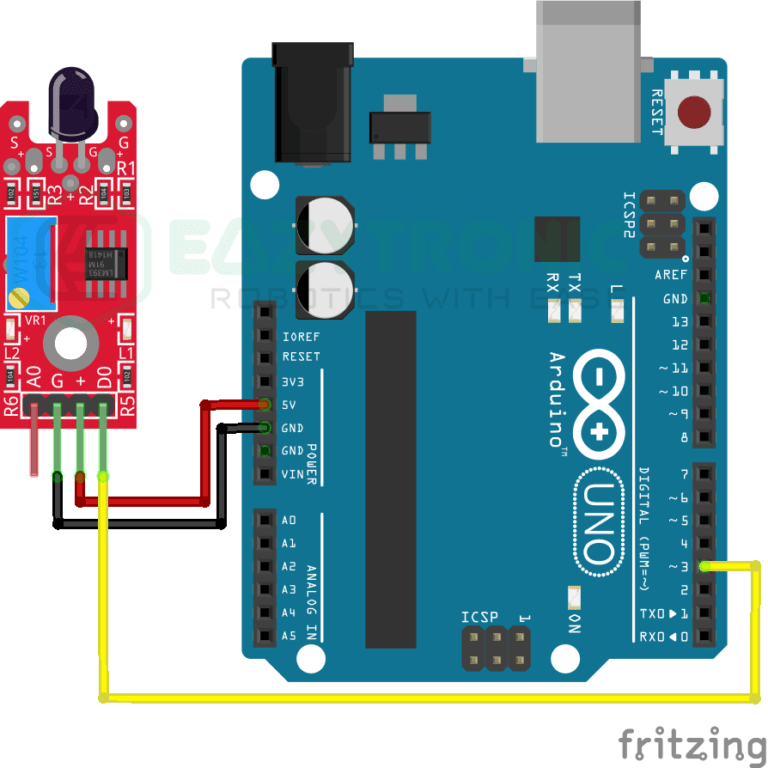
Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(" C");

}

**9)FLAME SENSOR**



const int flameSensorPin = 2; // Digital pin where flame sensor DO is connected

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

void setup() {

pinMode(flameSensorPin, INPUT); // Set flame sensor pin as input

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

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

}

void loop() {

int flameDetected = digitalRead(flameSensorPin); // Read the digital value from the flame sensor

if (flameDetected == HIGH) {

Serial.println("Flame detected!");

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

} else {

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

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

}

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

}