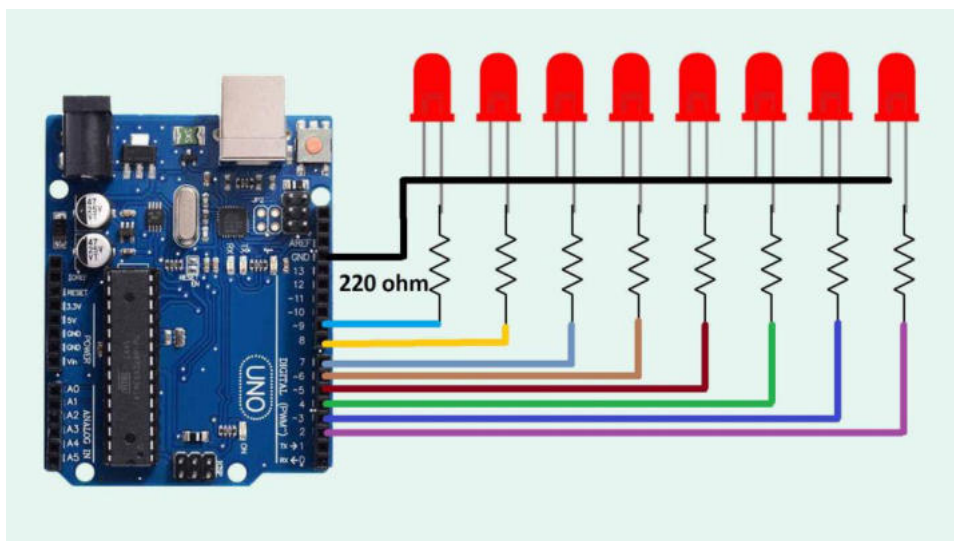
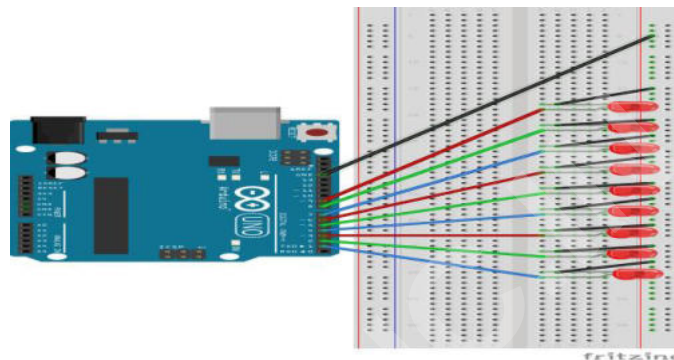
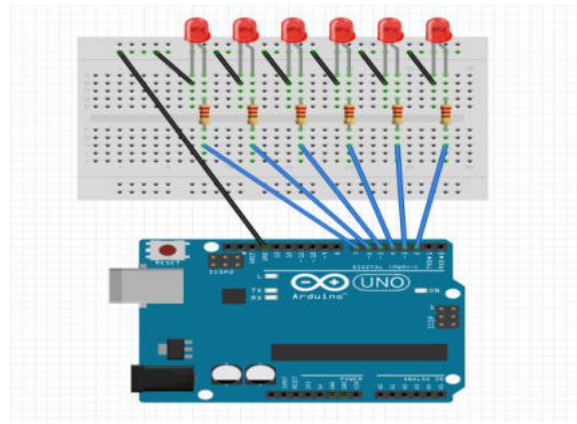


1. Develop a program to blink 5 LEDs back and forth

```
const int ledPins[] = {2, 3, 4, 5, 6}; // Pins for LEDs
const int numLeds = 5;
void setup()
{
  for (int i = 0; i < numLeds; i++)
  {
    pinMode(ledPins[i], OUTPUT);
  }
}
void loop()
{
  // Left to right
  for (int i = 0; i < numLeds; i++)
  {
    digitalWrite(ledPins[i], HIGH);
    delay(500);
    digitalWrite(ledPins[i], LOW);
  }
  // Right to left
  for (int i = numLeds - 1; i >= 0; i--)
  {
    digitalWrite(ledPins[i], HIGH);
    delay(500);
    digitalWrite(ledPins[i], LOW);
  }
}
```

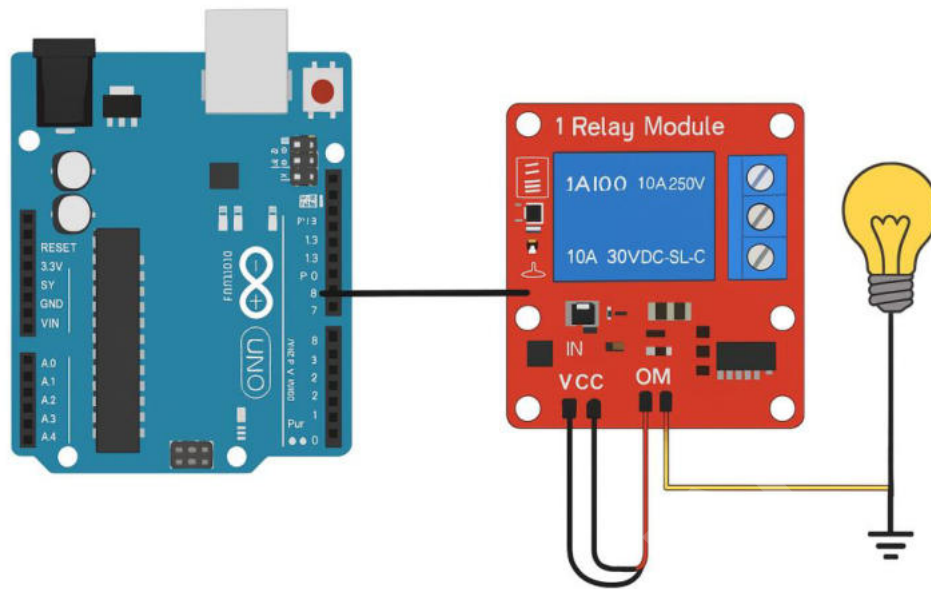




2. Develop a program to interface a relay with Arduino board

```
const int relayPin = 7;
void setup()
{
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW);
}

void loop()
{
  digitalWrite(relayPin, HIGH);
  delay(2000);
  digitalWrite(relayPin, LOW);
  delay(2000);
}
```



3. Develop a program to deploy an intrusion detection system using Ultrasonic and sound sensors

```
const int trigPin = 6;
const int echoPin = 7;
const int soundPin = 10;
const int buzzer = 4;
```

```
long duration;
float distance;
```

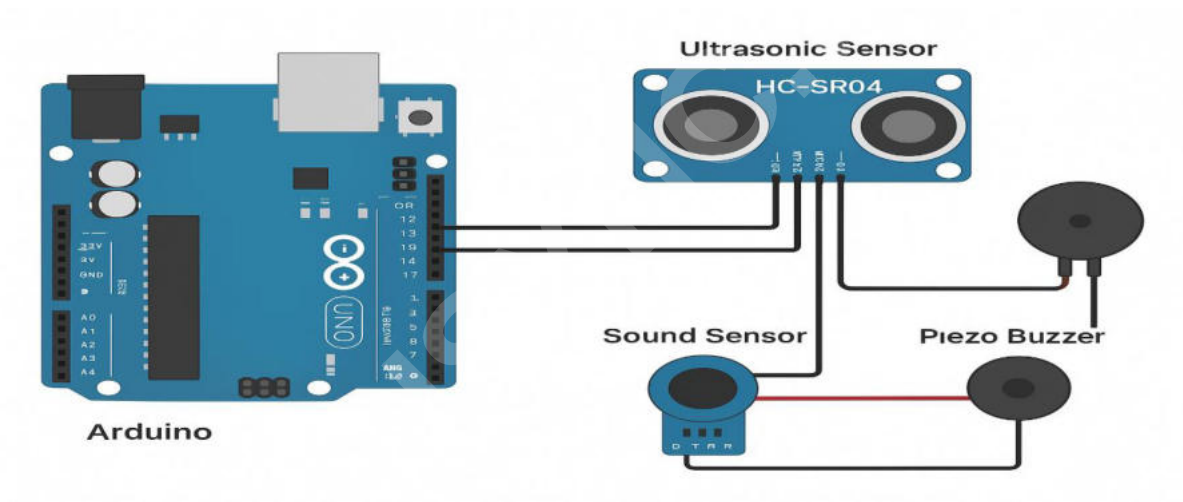
```
void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(soundPin, INPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  int soundVal = digitalRead(soundPin);
  Serial.print("Distance: ");
```

```

Serial.println(distance);
Serial.print("sound: ");
Serial.println(soundVal);
if ((distance > 0 && distance < 20) || soundVal == LOW)
{
    digitalWrite(buzzer, HIGH);
    Serial.println("Intrusion Detected!");
    delay(400);
} else {
    digitalWrite(buzzer, LOW);
    delay(200);
}
}
}

```



Output:

Distance: 45.32

sound: 1 (No Intrusion)

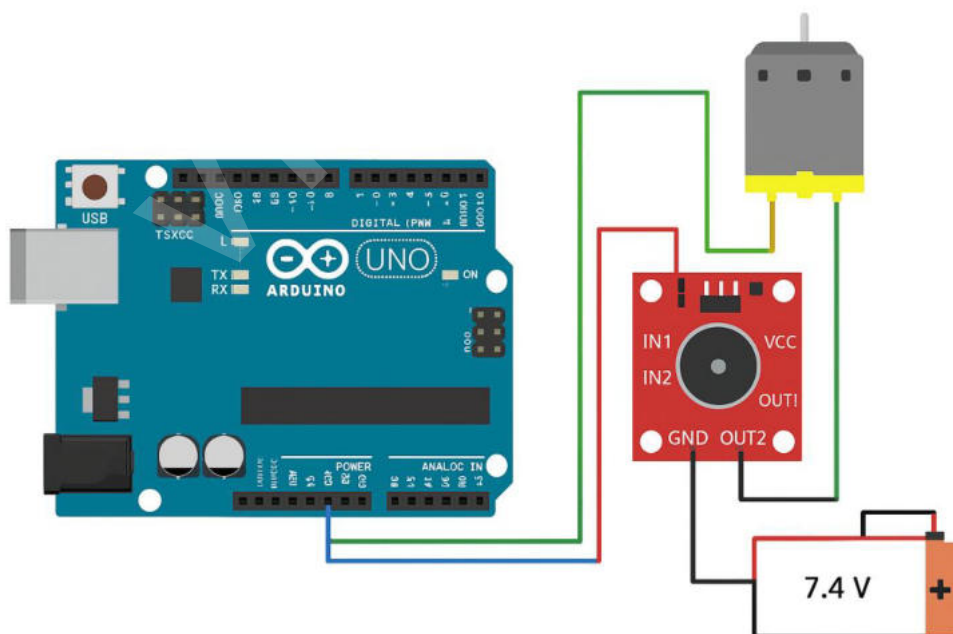
Distance: 12.48

sound: 0

Intrusion Detected! (Intrusion Detected (Distance < 20 cm OR sound = LOW)

4. Develop a program to control a DC motor with Arduino board

```
const int in1 = 8;  
const int in2 = 9;  
void setup()  
{  
  pinMode(in1, OUTPUT);  
  pinMode(in2, OUTPUT);  
}  
void loop()  
{  
  digitalWrite(in1, HIGH);  
  digitalWrite(in2, LOW);  
  delay(3000);  
  digitalWrite(in1, LOW);  
  digitalWrite(in2, HIGH);  
  delay(3000);  
}
```



5. Develop a program to deploy smart street light system using LDR sensor

```

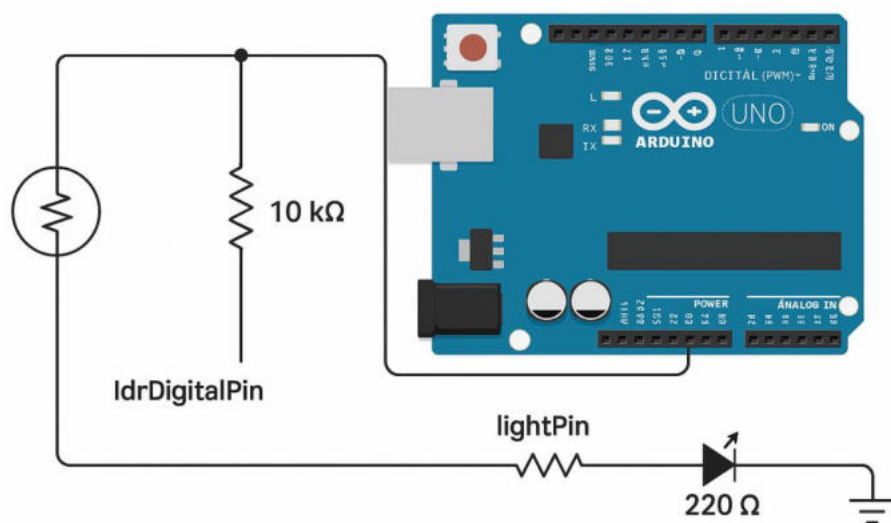
const int ldrDigitalPin = 2;
const int lightPin = 9;
void setup()
{
  pinMode(ldrDigitalPin, INPUT);
  pinMode(lightPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int ldrState = digitalRead(ldrDigitalPin);
  Serial.print("LDR State: ");
  Serial.println(ldrState);

  if (ldrState == HIGH)
  {
    digitalWrite(lightPin, HIGH);
  } else
  {
    digitalWrite(lightPin, LOW);
  }

  delay(500);
}

```



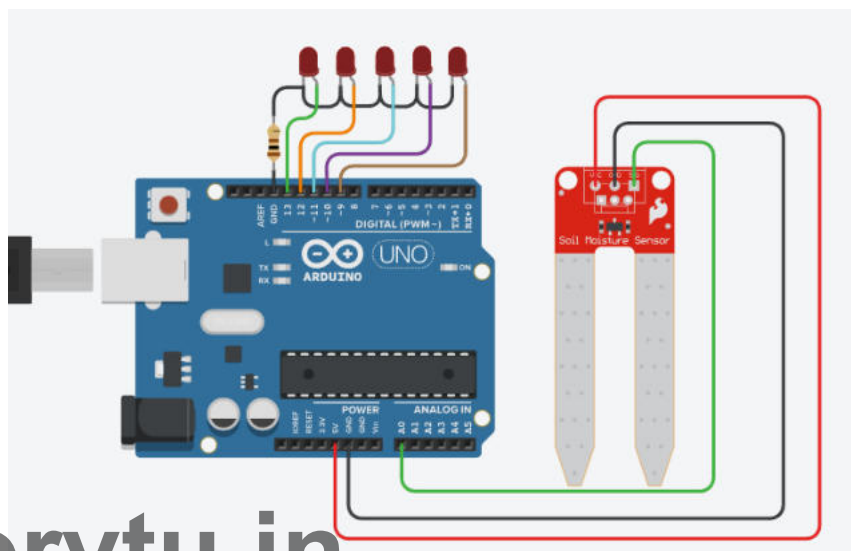
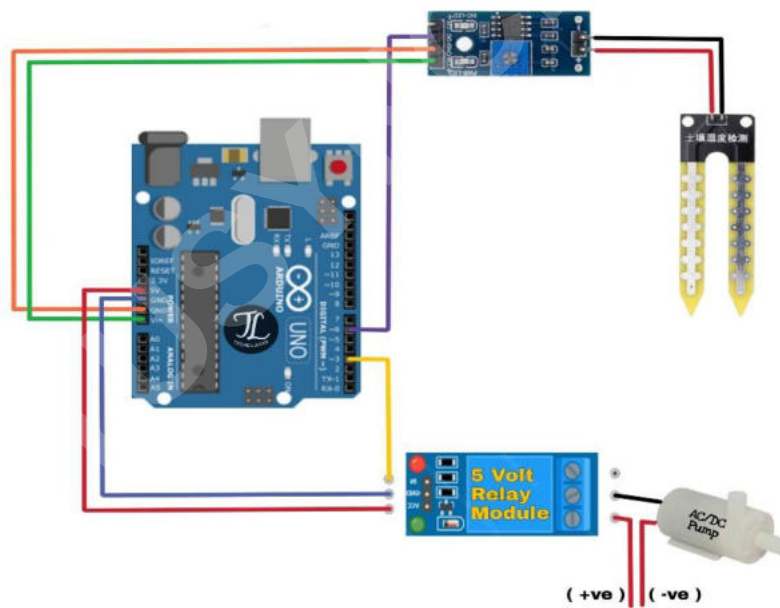
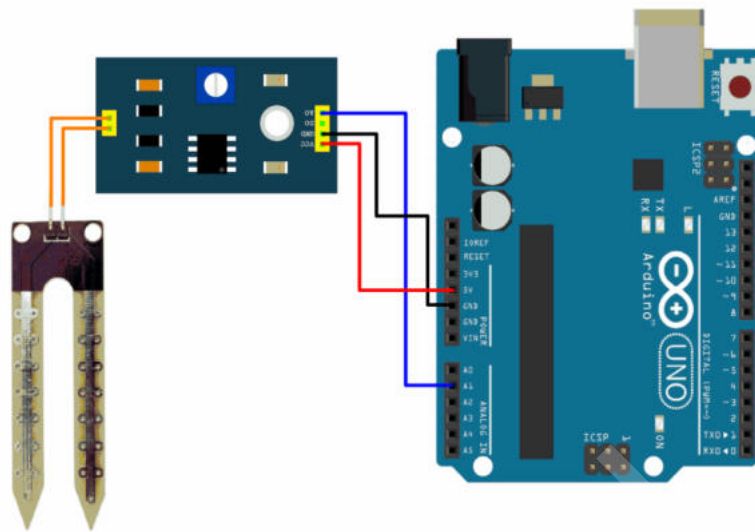
6. Develop a program to classify dry and wet waste with the Moisture sensor (DHT22)

```
const int moisturePin = A0;
    int moistureValue = 0;
    const int threshold = 400;
    void setup()
    {
        Serial.begin(9600);
        pinMode(moisturePin, INPUT);
    }
    void loop()
    {
        moistureValue = analogRead(moisturePin);
        Serial.print("Moisture Value: ");
        Serial.println(moistureValue);
        if (moistureValue > threshold)
        {
            Serial.println("Dry Waste Detected");
        }
        else
        {
            Serial.println("Wet Waste Detected");
        }
        delay(1000);
    }
```

OUTPUT:

Moisture Value: 650
Dry Waste Detected

Moisture Value: 250
Wet Waste Detected



7. Develop a program to read the pH value of a various substances like milk, lime and water

```
int sensorPin = A0;
const int samples = 10;
int readings[samples];
int temp;
unsigned long avgValue;

const float V_7 = 4.10;
const float slope = -5.0;
const float calibration_value = 7 - slope * V_7;
void setup()
{
    Serial.begin(9600);
}

void loop() {

    for (int i = 0; i < samples; i++)
    {
        readings[i] = analogRead(sensorPin);
        delay(30);
    }

    for (int i = 0; i < samples - 1; i++)
    {
        for (int j = i + 1; j < samples; j++)
        {
            if (readings[i] > readings[j]) {
                temp = readings[i];
                readings[i] = readings[j];
                readings[j] = temp;
            }
        }
    }

    avgValue = 0;
```

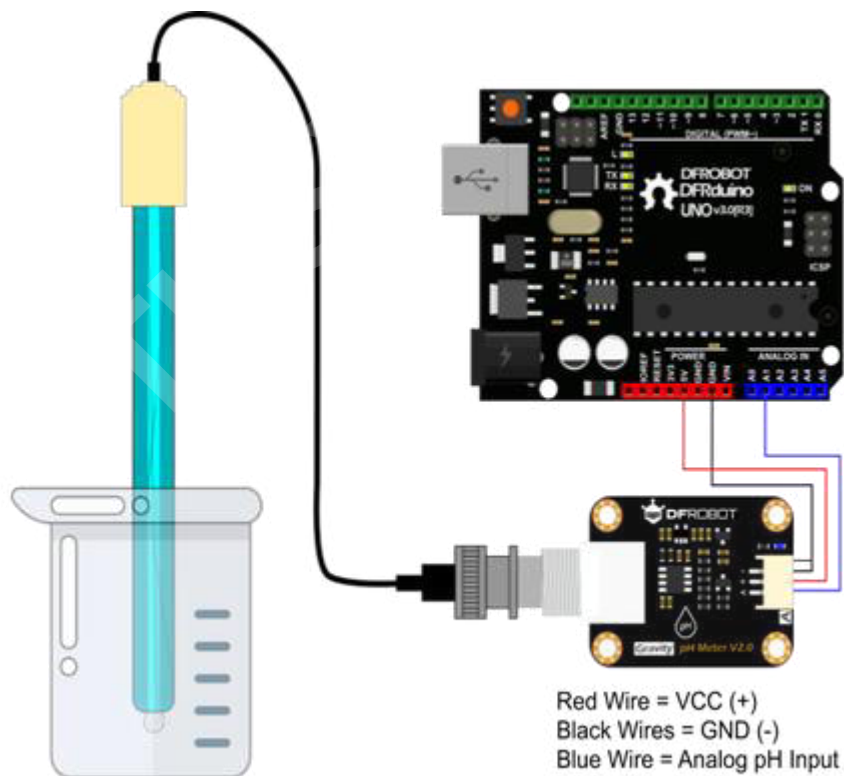
```

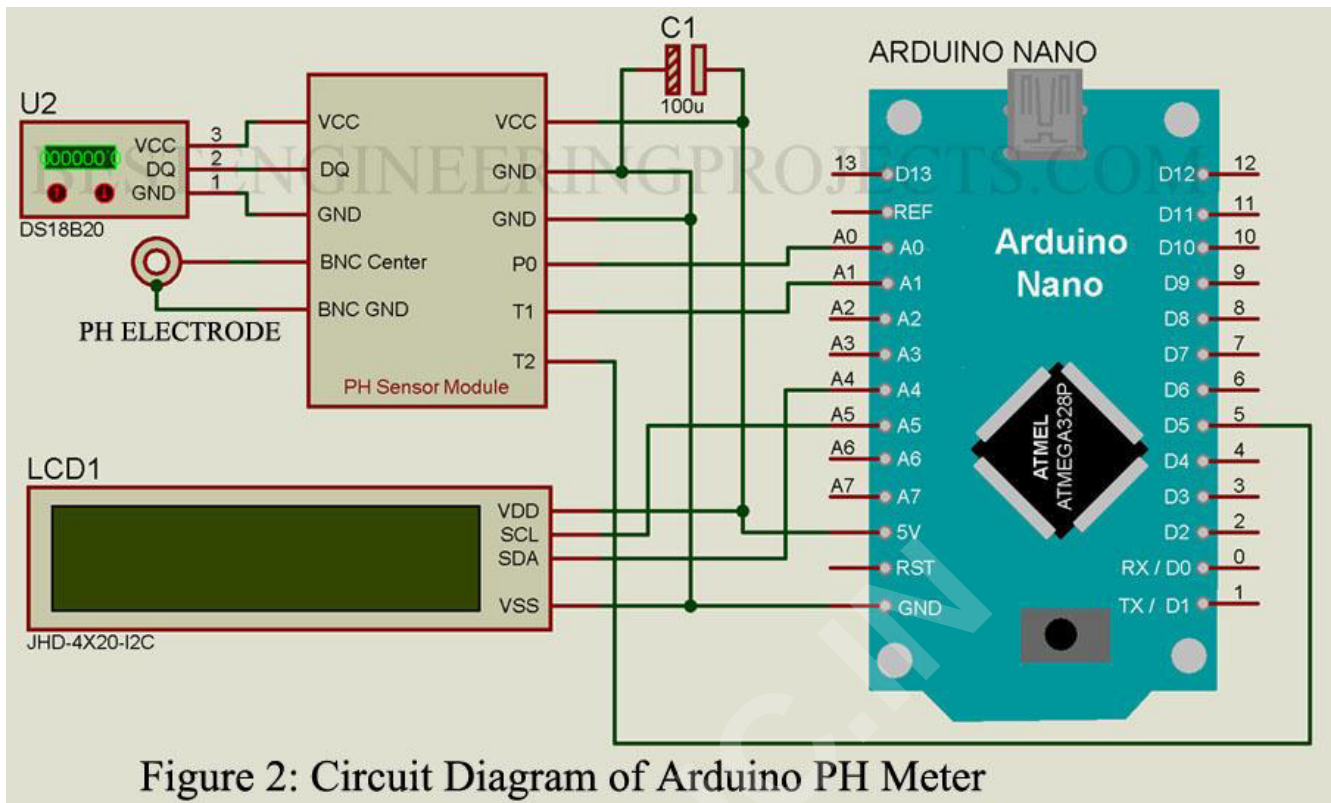
for (int i = 2; i < samples - 2; i++)
{
    avgValue += readings[i];
}
avgValue /= (samples - 4);

float voltage = avgValue * 5.0 / 1024.0;

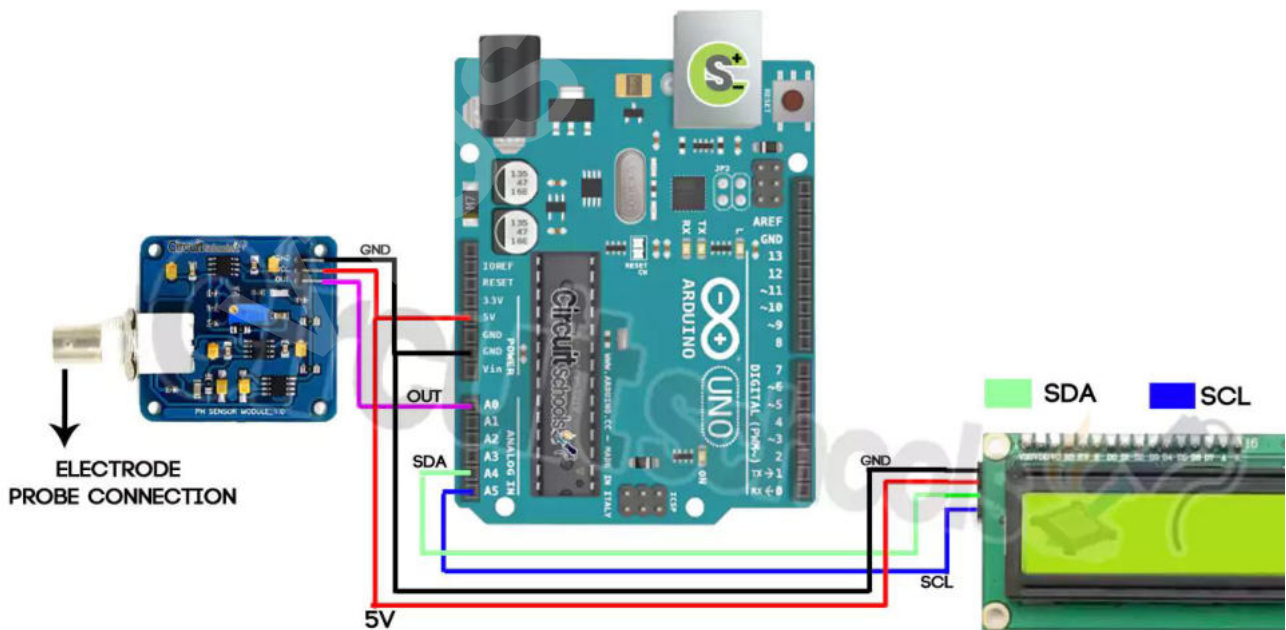
float pHValue = slope * voltage + calibration_value;
Serial.print("Analog: ");
Serial.print(avgValue);
Serial.print(" | Voltage: ");
Serial.print(voltage, 3);
Serial.print(" V | pH: ");
Serial.println(pHValue, 2);
delay(1000);
}

```





PH METER INTERFACING ARDUINO AND PH SENSOR CIRCUIT DIAGRAM



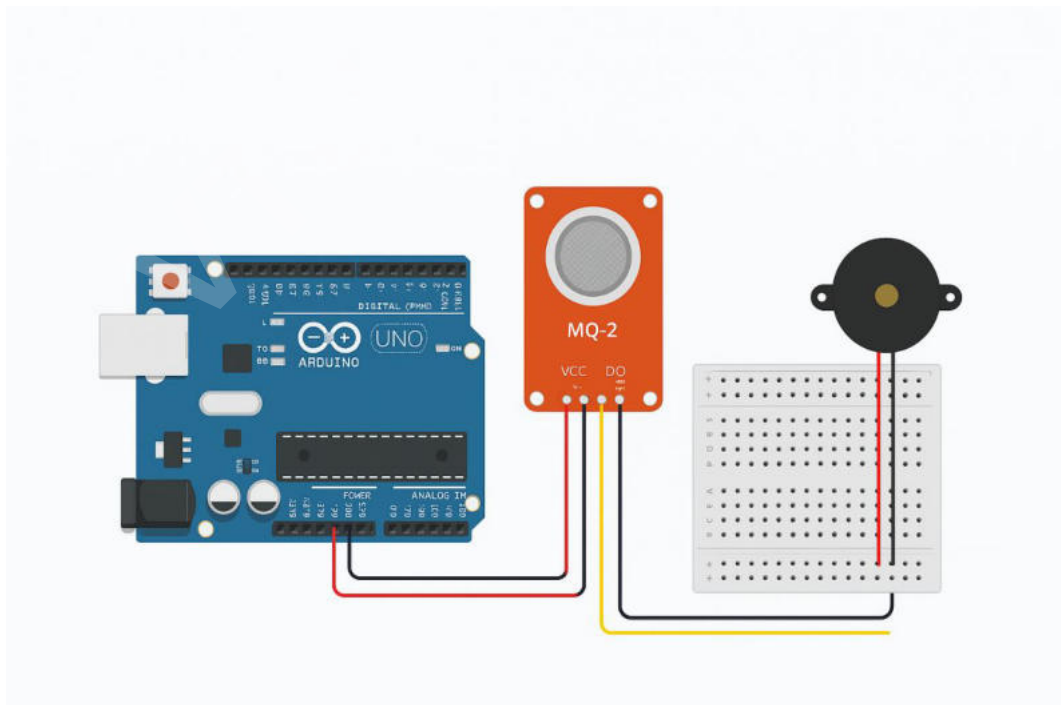
8. Develop a program to detect the gas leakage in the surrounding environment

```
const int gasSensorPin = A0;
const int buzzerPin = 8;
int gasLevel = 0;
int threshold = 400;
void setup()
```

```

{
  pinMode(gasSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  gasLevel = analogRead(gasSensorPin);
  Serial.print("Gas Level: ");
  Serial.println(gasLevel);
  if (gasLevel > threshold)
  {
    digitalWrite(buzzerPin, HIGH);
    Serial.println("Gas Leak Detected!");
  }
  else
  {
    digitalWrite(buzzerPin, LOW);
  }
  delay(1000);
}

```



OUTPUT:
 Gas Level: 512
 Gas Leak Detected!

9. Develop a program to demonstrate weather station readings using Arduino

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup()
{
  Serial.begin(9600);
  dht.begin();
  Serial.println("Weather Station Initialized");
}
void loop()
{
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  if (isnan(temp) || isnan(hum))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" °C");
  Serial.print("Humidity: ");
  Serial.print(hum);
  Serial.println(" %");

  Serial.println("-----");
  delay(2000);
}
```

OUTPUT:

Weather Station Initialized

Temperature: 27.00 °C

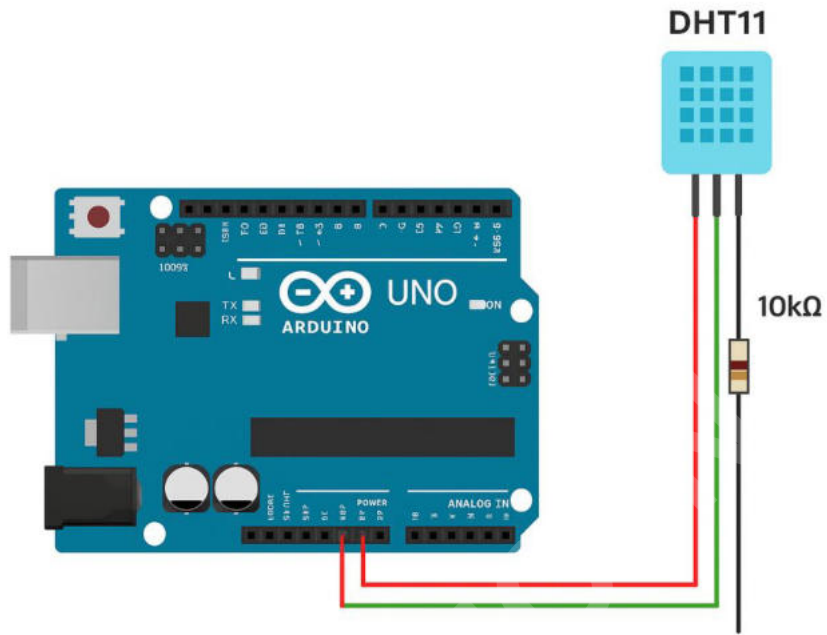
Humidity: 65.00 %

Temperature: 28.00 °C

Humidity: 63.00 %

Temperature: 27.50 °C

Humidity: 64.50 %



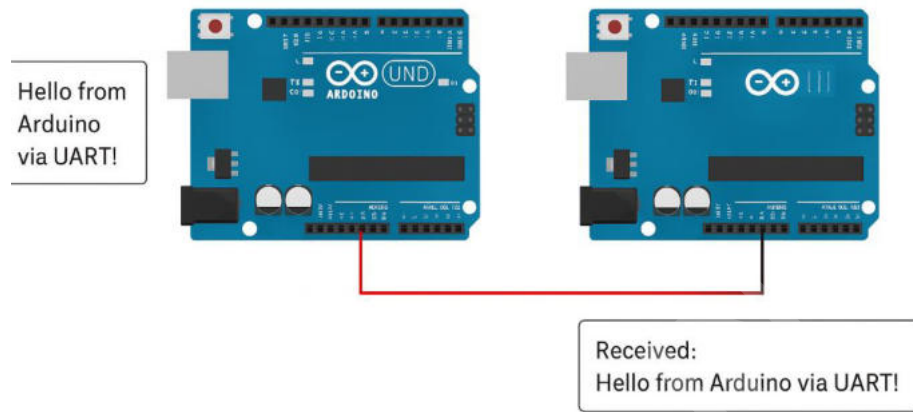
10. Develop a program to setup a UART protocol and pass a string through the protocol

```
void setup()
{
  Serial.begin(9600);
  delay(1000);
}

void loop()
{
  String message = "Hello from Arduino via UART!";
  Serial.println(message);
  delay(2000);
}
```

To Receive UART Data:

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  if (Serial.available())
  {
    String received = Serial.readString();
    Serial.print("Received: ");
    Serial.println(received);
  }
}
```



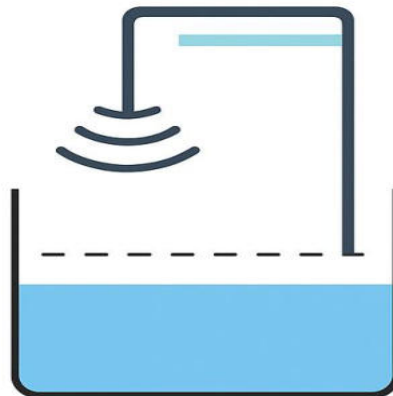
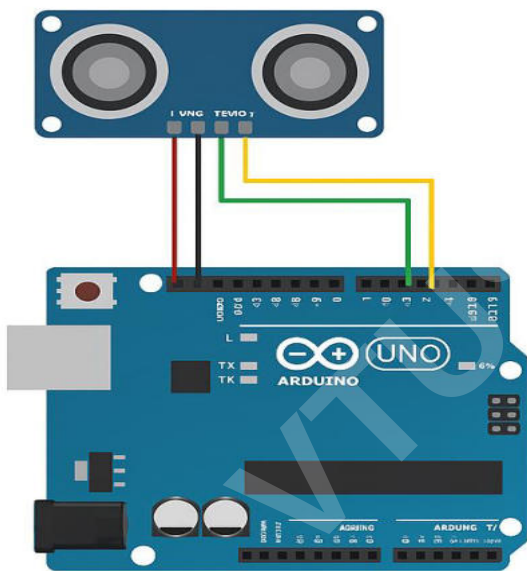
11. Develop a water level depth detection system using Ultrasonic sensor

```
const int trigPin = 9;
const int echoPin = 8;
const int MAX_DISTANCE_CM = 100;
long duration;
int distance;
int waterLevel;
void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
void loop()
{
  // Trigger ultrasonic pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  if (distance > MAX_DISTANCE_CM)
  {
    distance = MAX_DISTANCE_CM;
  }
}
```

```

waterLevel = MAX_DISTANCE_CM - distance;
Serial.print("Distance to water: ");
Serial.print(distance);
Serial.println(" cm");
Serial.print("Water Level: ");
Serial.print(waterLevel);
Serial.print(" cm ");
Serial.print(waterLevel * 100);
Serial.println("%");
Serial.println("----- ");
delay(2000);
}

```



Distance to water: 60 cm
Water Level: 40 cm 60 %

12. Develop a program to simulate interfacing with the keypad module to record the keystrokes

```
#include <Keypad.h>
const byte ROWS = 4;
const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {9, 8, 7, 6}; //R1=9,R2=8,R3=7,R4=6
byte colPins[COLS] = {5, 4, 3, 2}; //C1=5,C2=4,C3=3,C4=2

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
void setup()
{
  Serial.begin(9600);
  Serial.println("Hex Pad Ready");
}

void loop()
{
  char customKey = customKeypad.getKey();

  if (customKey)
  {
    Serial.print("Pressed: ");
    Serial.println(customKey);
  }
}
```

