

SENSOR DE AGUA

Un sensor de agua es un dispositivo electrónico es un dispositivo electrónico que se utiliza para detectar la calidad, presencia o el nivel de líquidos, como agua, en un entorno determinado. Es un componente clave en proyectos de automatización y monitoreo que involucran el control de fluidos.

MATERIALES

1. Sensor de agua
2. Arduino
3. Jumpers
4. Leds
5. Resistencias
6. Protoboard
7. Cable USB

CÓDIGO

```
int activador = A5;
int ledVerde = 8;
int ledAmarillo = 9;
int ledRojo = 10;

void setup() {
  pinMode(ledVerde, OUTPUT); //salida
  pinMode(ledAmarillo, OUTPUT); //salida
  pinMode(ledRojo, OUTPUT); //salida
  pinMode(activador, INPUT); //entrada
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(activador);

  Serial.println(sensorValue);
  delay(500);

  if (sensorValue > 450 && sensorValue < 550) {
    digitalWrite(ledVerde, HIGH);
    delay(1000);
    digitalWrite(ledVerde, LOW);
  }

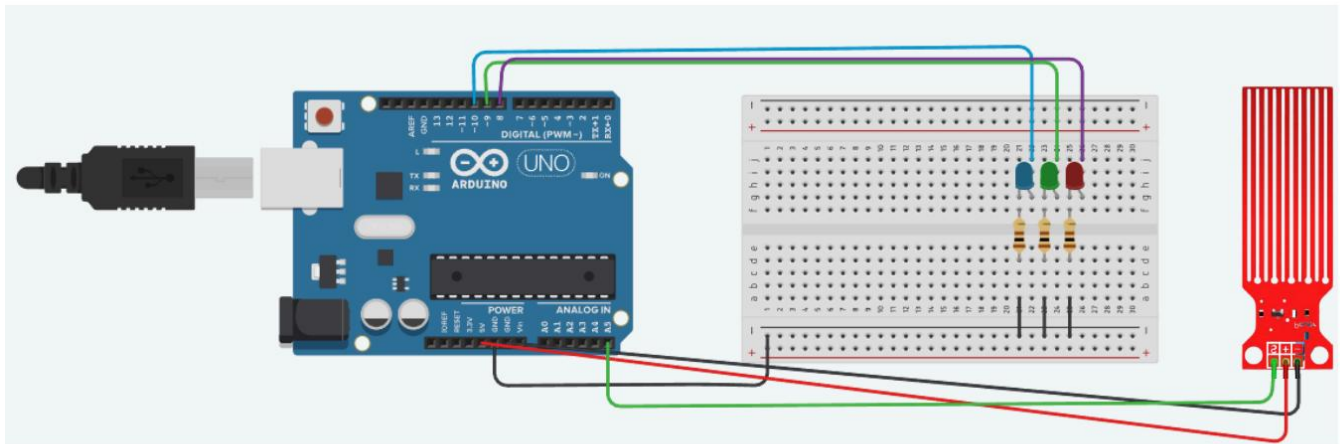
  if (sensorValue > 551 && sensorValue < 650) {
    digitalWrite(ledAmarillo, HIGH);
```

```

    delay(1000);
    digitalWrite(ledAmarillo, LOW);
  }
  if (sensorValue > 651 && sensorValue < 700) {
    digitalWrite(ledRojo, HIGH);
    delay(1000);
    digitalWrite(ledRojo, LOW);
  }
  else {
    digitalWrite(ledRojo, LOW);
    digitalWrite(ledVerde, LOW);
    digitalWrite(ledAmarillo, LOW);
  }
}
}

```

DIAGRAMA DE CONEXIÓN



JOYSTICK

Un joystick es un dispositivo de entrada que actúa como sensor, permitiendo detectar y transmitir movimientos en una o varias direcciones. Utiliza potenciómetros o sensores Hall para registrar las variaciones angulares del eje y convertirlas en señales eléctricas que un sistema puede procesar. Es común en aplicaciones como videojuegos, control de maquinaria o robótica, donde se requiere precisión en el control manual

MATERIALES

1. Arduino
2. Resistencias
3. Protoboard
4. Leds
5. Jumpers macho-hembra
6. Jumpers macho-macho
7. Joystick
8. Cable USB

CÓDIGO

```
int y;  
int x;  
  
void setup () {  
  Serial.begin(9600);  
  pinMode(9,OUTPUT);  
  pinMode(10,OUTPUT);  
  pinMode(11,OUTPUT);  
  pinMode(12,OUTPUT);  
}  
joistic  
  
void loop() {  
  y = analogRead(A1);  
  x = analogRead(A0);  
  Serial.print("y=");
```

```
Serial.print(y);
Serial.print("x=");
Serial.println(x);
delay(200);

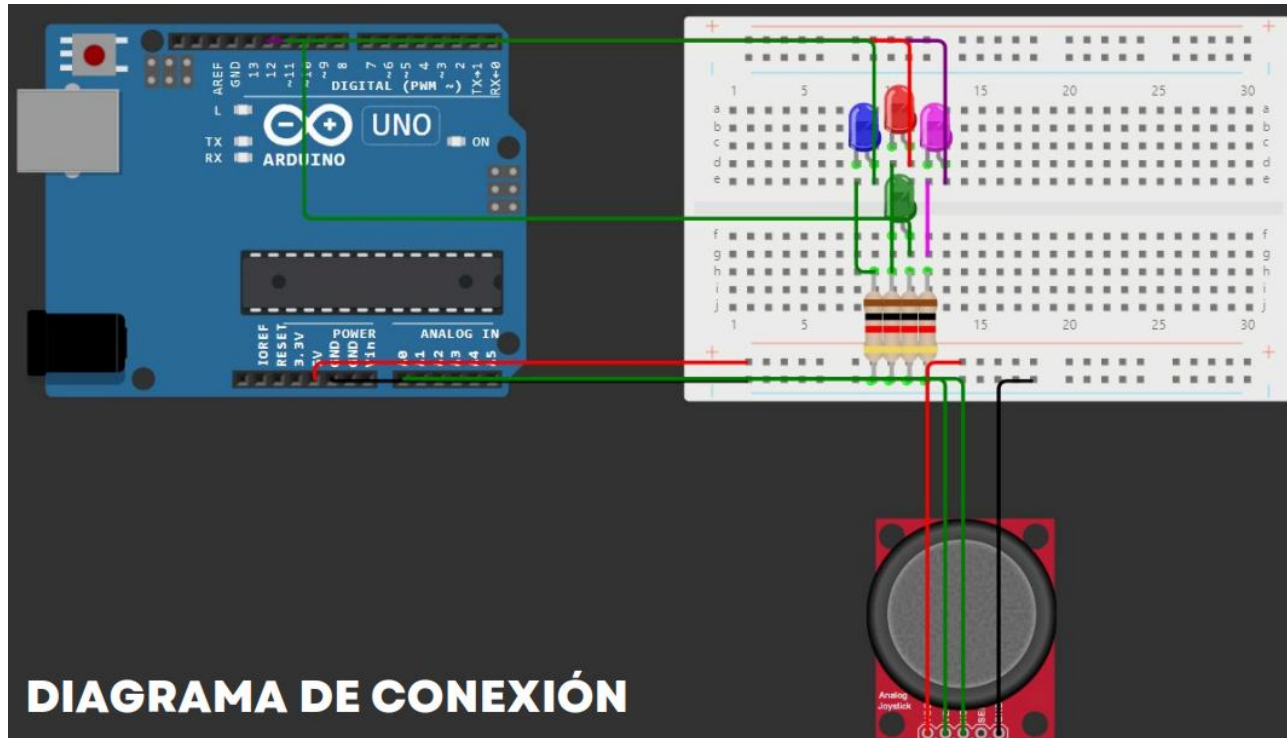
if (x>520){
  digitalWrite(9,HIGH);
}
else
digitalWrite(9,LOW);

if (x<500){
  digitalWrite(10,HIGH);
}
else
digitalWrite(10,LOW);

if (y>1020){
  digitalWrite(11,HIGH);
}
else
digitalWrite(11,LOW);

if (y<500){
  digitalWrite(12,HIGH);
}
else
digitalWrite(12,LOW);
}
```

DIAGRAMA DE CONEXIÓN



SENSOR TOUCH

Un sensor táctil capacitivo es un dispositivo que presenta un comportamiento similar a un pulsador, pero puede ser activado con poca o ninguna presión. Este tipo de sensor táctil basa su funcionamiento en la medición de la variación de la capacitancia.

MATERIALES

1. Leds
2. Resistencias
3. Jumpers macho-macho
4. Jumpers macho-hembra
5. Protoboard
6. Arduino
7. Cable USB

CÓDIGO

```
int led1 = 13;
int led2 = 12;
int led3 = 11;

int identificador = 0;

void setup() {
    // put your setup code here, to run once:
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
}

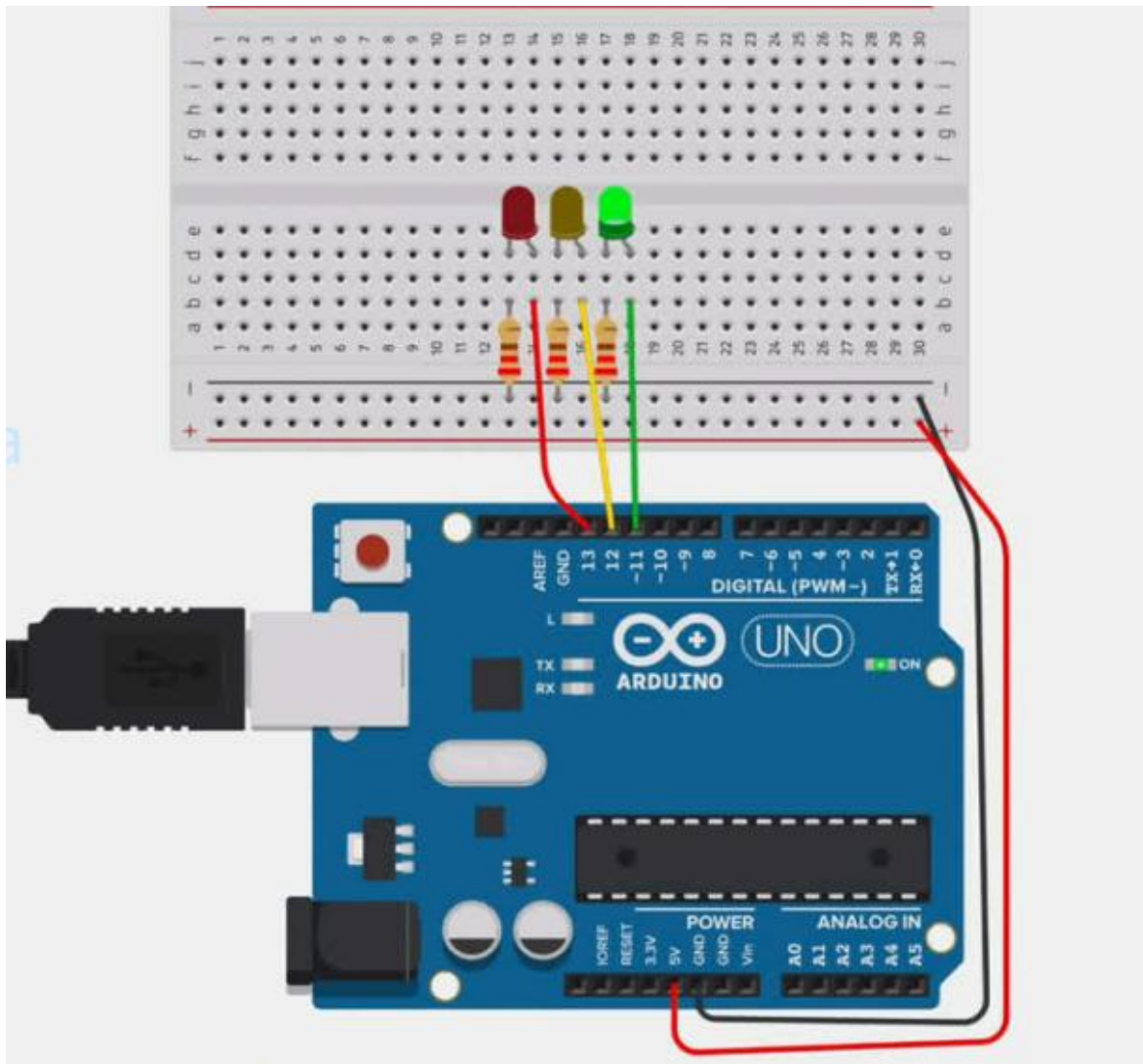
void loop() {
    // put your main code here, to run repeatedly:
    if(digitalRead(7) == HIGH){
        delay(200);
        if(identificador == 0){
            digitalWrite(led1, HIGH);
            delay(200);
            digitalWrite(led2, LOW);
            delay(200);
            digitalWrite(led3, LOW);
            identificador += 1;
        }
        else if(identificador == 1){
            digitalWrite(led1, LOW);
            digitalWrite(led2, HIGH);
            delay(200);
            digitalWrite(led3, LOW);
            identificador += 1;
        }
        else if(identificador == 2){
            digitalWrite(led1, LOW);
            digitalWrite(led2, LOW);
            digitalWrite(led3, HIGH);
            delay(200);
            identificador = 0;
        }
    }
}
```

```

else {
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
}
}

```

DIAGRAMA DE CONEXIÓN



SERVOMOTOR

Un servomotor es un dispositivo alimentado por corriente continua que puede controlar de modo muy exacto la posición (de 0° a 180°) o la velocidad (en revoluciones por minuto, rpm, en sentido horario o antihorario). Tienen 3 pines para su conexión: alimentación (5 V, normalmente), tierra (GND) y el pin de la señal.

MATERIALES

1. Arduino
2. Protoboard
3. Jumpers
4. Leds
5. Sensor ultrasónico
6. Servomotor
7. Cable USB

CÓDIGO

```
//se declaran las variables con los que se identificaran los diferentes
leds
int verde = 13;
int amarillo = 12;
int rojo = 8;
int verde2 = 7;
int rojo2 = 4;

void setup() {
  //se indica la salida de datos
  pinMode(verde, OUTPUT);
  pinMode(amarillo, OUTPUT);
  pinMode(rojo, OUTPUT);
  pinMode(verde2, OUTPUT);
  pinMode(rojo2, OUTPUT);
}

void loop() {
  //se indica que el led estará prendido durante medio segundo y
  despues se apagara durante medio segundo
  //sucesivamente pasando por todos los leds
```



```
digitalWrite(rojo2, HIGH);  
delay(500);  
digitalWrite(rojo2, LOW);  
delay(500);
```

```
digitalWrite(verde, HIGH);  
delay(500);  
digitalWrite(verde, LOW);  
delay(500);
```

```
digitalWrite(amarillo, HIGH);  
delay(500);  
digitalWrite(amarillo, LOW);  
delay(500);
```

```
digitalWrite(rojo, HIGH);  
delay(500);  
digitalWrite(rojo, LOW);  
delay(500);
```

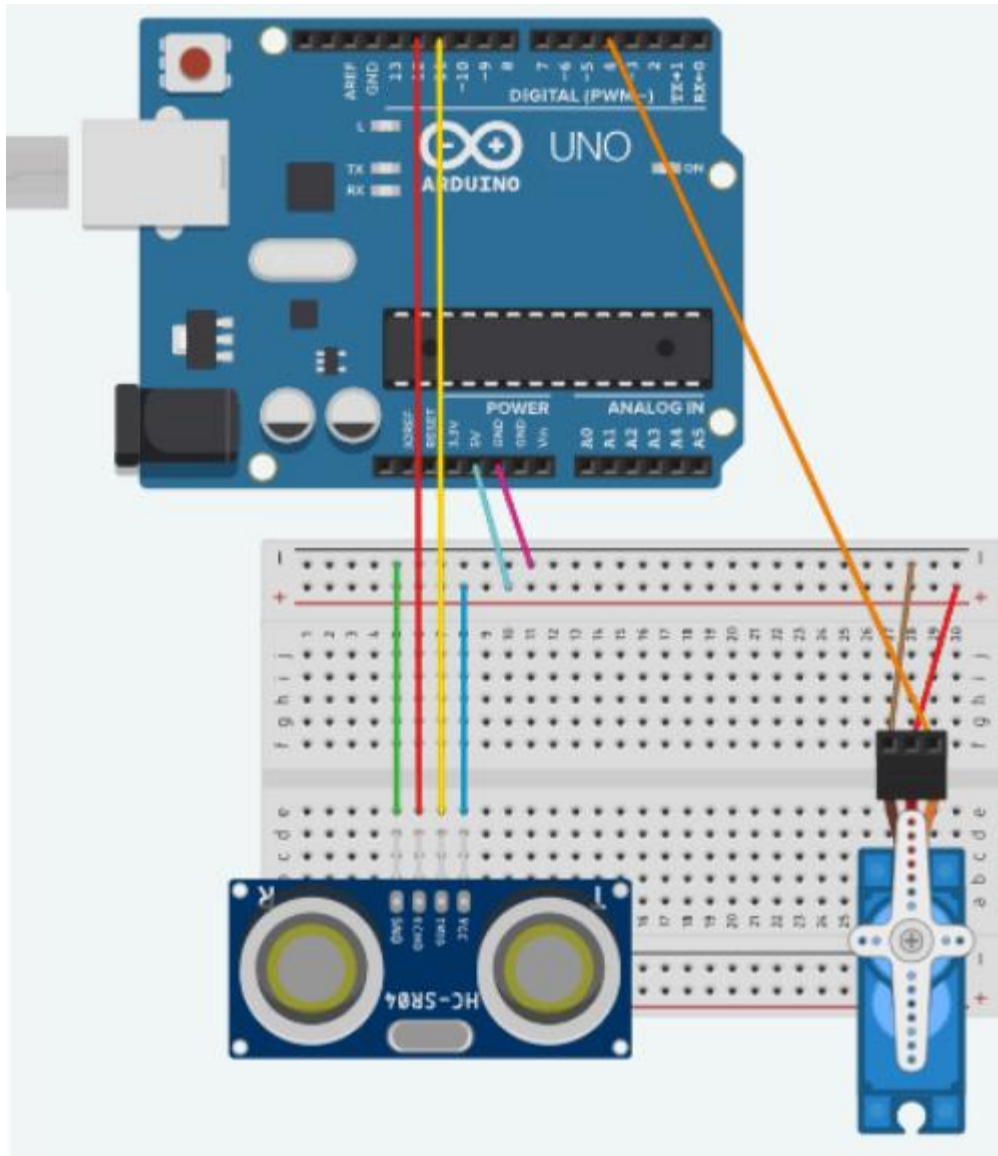
```
digitalWrite(verde2, HIGH);  
delay(500);  
digitalWrite(verde2, LOW);  
delay(500);
```

```
digitalWrite(rojo, HIGH);  
delay(500);  
digitalWrite(rojo, LOW);  
delay(500);
```

```
digitalWrite(amarillo, HIGH);  
delay(500);  
digitalWrite(amarillo, LOW);  
delay(500);
```

```
digitalWrite(verde, HIGH);  
delay(500);  
digitalWrite(verde, LOW);  
delay(500);  
}
```

DIAGRAMA DE CONEXIÓN



BUZZER

Un buzzer es un transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono (generalmente agudo). Sirve como mecanismo de señalización o aviso y se utiliza en múltiples sistemas, como en automóviles o en electrodomésticos, incluidos los despertadores.

MATERIALES

1. Arduino
2. Protoboard
3. Jumpers
4. Buzzer
5. Resistencia
6. Cable USB

CÓDIGO

```
int buzzer = 12;
const int sensorH1 = 9;
const int ledPin = 2;

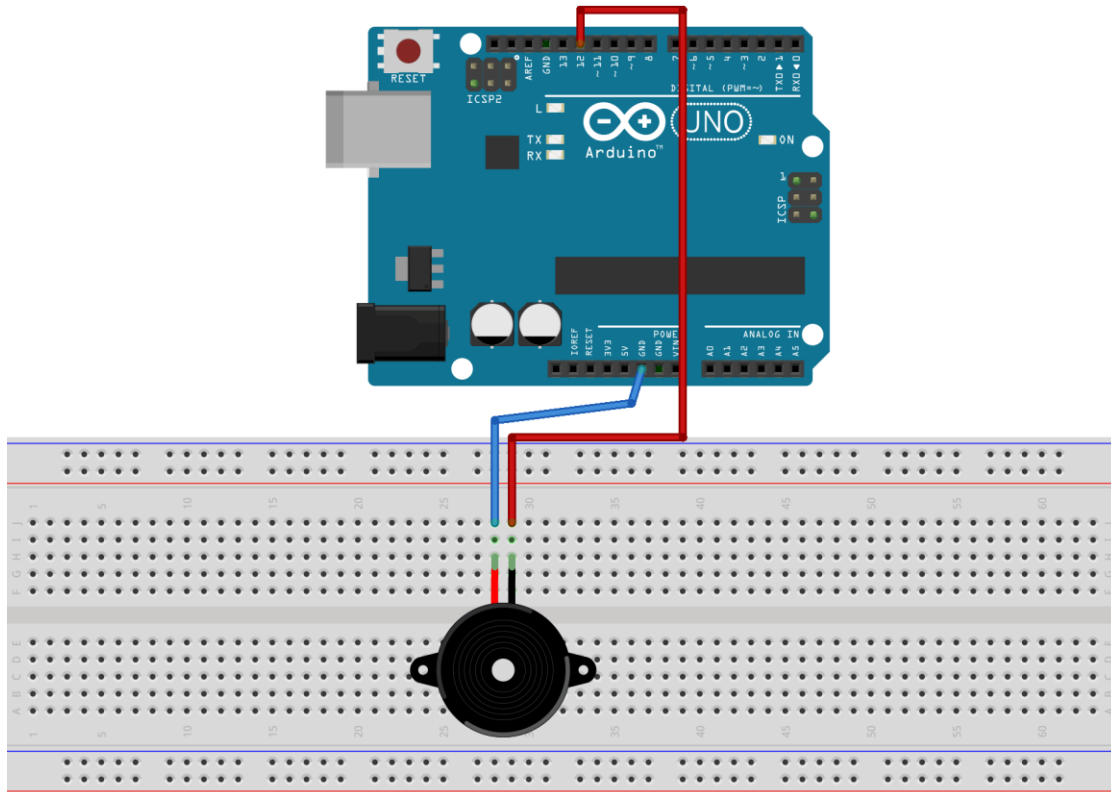
void setup() {
  // put your setup code here, to run once:
  pinMode(sensorH1, INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzer, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  int estadoP1 = digitalRead(sensorH1);

  if(estadoP1 == LOW){
    digitalWrite(ledPin, HIGH);
    noTone(buzzer);
  }else{
    digitalWrite(ledPin,LOW);
    tone(buzzer, 600);
  }
}
```

}

DIAGRAMA DE CONEXIÓN



SENSOR SONIDO

Un sensor de sonido es un sensor analógico de sonido utilizado para detectar sonido, utiliza un micrófono cilíndrico de alta sensibilidad. Posee una salida analógica (señal analógica de audio) y una salida digital (salida de comparador opamp, con umbral regulable por potenciómetro en placa).

MATERIALES

1. Arduino
2. Protoboard
3. Leds
4. Resistencias
5. Jumpers
6. Sensor de sonido
7. Cable USB

CÓDIGO

```
const int pinSensorSonido = A0;
const int numLEDs = 6;
int leds[numLEDs] = {2, 3, 4, 5, 6, 7};

void setup() {
  // Configura los pines de los LEDs como salida
  for (int i = 0; i < numLEDs; i++) {
    pinMode(leds[i], OUTPUT);
  }

  Serial.begin(9600); // Inicia la comunicación serial
}

void loop() {
  int nivelRuido = analogRead(pinSensorSonido); // Lee el valor
  analógico del sensor de sonido
  Serial.print("Valor crudo del sensor: ");
  Serial.println(nivelRuido); // Imprime el valor crudo para ver si
  está cambiando

  // Mapea el valor de 0-1023 (que es el rango del sensor analógico)
  a un rango de 0 a 28
  int ruidoMapeado = map(nivelRuido, 0, 1023, 0, 200);
  Serial.print("Nivel de ruido mapeado: ");
  Serial.println(ruidoMapeado); // Muestra el valor mapeado en el
  monitor serial

  // Decide cuántos LEDs encender según el nivel de ruido mapeado
```

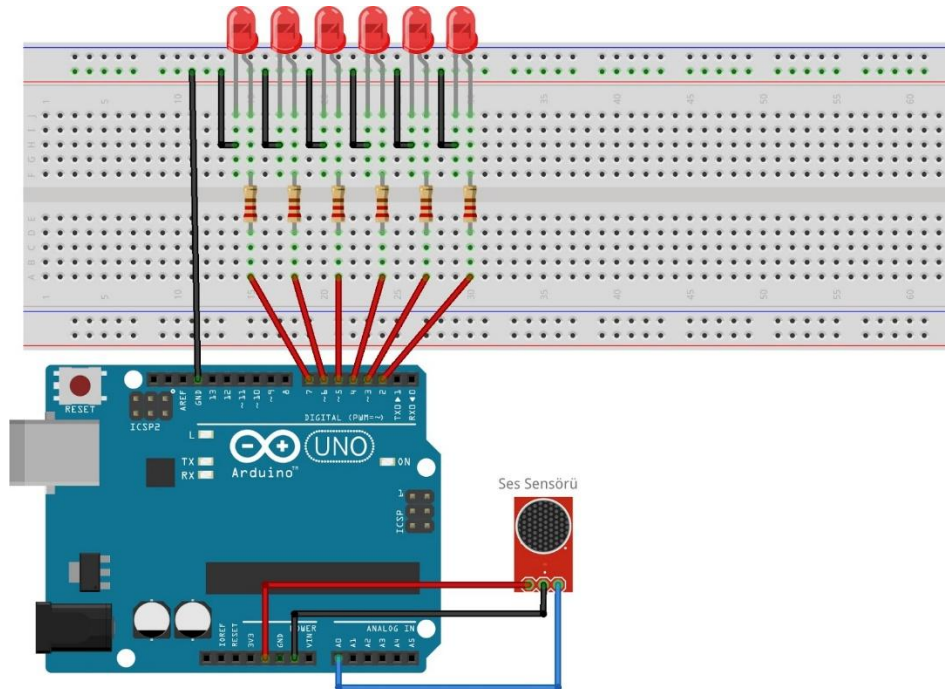
```

for (int i = 0; i < numLEDs; i++) {
  // Enciende el LED correspondiente si el valor de ruido mapeado
  es mayor que el umbral
  if (ruidoMapeado > (i * (38 / numLEDs))) {
    digitalWrite(leds[i], HIGH);
  } else {
    digitalWrite(leds[i], LOW);
  }
}

delay(100); // Retardo para estabilidad y lectura continua
}

```

DIAGRAMA DE CONEXIÓN



SENSOR DE HUMEDAD DE SUELO

El sensor de humedad permite medir de forma sencilla, por medio de 2 electrodos resistivos. El sensor se puede utilizar para checar la humedad de las plantas y así mismo puede estar al pendiente de cuando necesiten agua. Detecta la humedad en las plantas, básicamente su función es checar si la tierra que tienen las plantas esta seca o húmeda.

MATERIALES

1. Arduino
2. Protoboard
3. Jumpers
4. Resistencia
5. Led
6. Sensor de humedad de suelo
7. Cable USB

CÓDIGO

```
const int sensorH1 = 9;
const int ledPin = 2;
int buzzer = A1; //agregar

void setup() {
  // put your setup code here, to run once:
  pinMode(sensorH1, INPUT_PULLUP);
  pinMode(ledPin, OUTPUT);
  pinMode(buzzer, OUTPUT); //agregar
}

void loop() {
  // put your main code here, to run repeatedly:
  int estadoP1 = digitalRead(sensorH1);

  if(estadoP1 == LOW){
```

```

digitalWrite(ledPin, HIGH);
noTone(buzzer); //agregar
}else{
digitalWrite(ledPin, LOW);
tone(buzzer, 600); //agregar
}
}

```

DIAGRAMA DE CONEXIÓN

