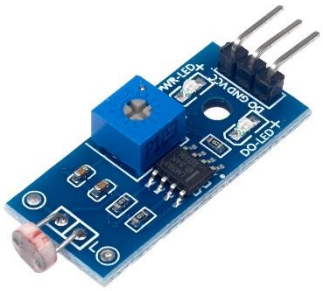
TEMA 5. SENSORES BÁSICO PARA ARDUINO

5.1.1. SENSOR DE LUZ LDR



Este módulo posee 1 salida digital. La salida digital del módulo posee solo 2 estados: activo/apagado (on/off), el cambio de un estado a otro depende del umbral que se fije con el potenciómetro del módulo. La salida digital puede utilizarse para controlar un relay y así realizar una acción dependiente de la intensidad de luz.

ESPECIFICACIONES TÉCNICAS

- Voltaje de Operación: 3.3V - 5V DC
- Conexión de 3 cables: VCC, GND, DO
- Salida digital(comparador)
- Opamp en modo comparador: LM393
- Potenciómetro para ajuste de comparador
- Led rojo de encendido y verde de salida digital
- Dimensiones:35x14mm
- Peso:4g

CONEXIÓN

- VCC: 3.3V - 5V
- GND: 0V, Tierra
- DO: Salida digital (TTL)

5.1.2. SENSOR DE TEMPERATURA LM35

Teoría:

El LM35 es un sensor de temperatura lineal que entrega una salida analógica proporcional a la temperatura en grados Celsius. Genera 10 mV por cada grado Celsius de temperatura detectada, lo que lo hace fácil de usar con microcontroladores.

Características Principales:

- Rango de operación: -55 °C a 150 °C.
- Precisión: ±0.5 °C a temperatura ambiente.
- Voltaje de alimentación: 4V a 30V.
- Señal de salida: Analógica (lineal con respecto a la temperatura).

Cálculo: La temperatura se obtiene directamente del voltaje de salida:

Temperatura (°C)=Vout×100

Donde VoutV_{out}Vout es el voltaje en volts.

ESPECIFICACIONES TÉCNICAS

- Modelo: LM35DZ/LFT1
- Encapsulado: TO-92 3-pin
- Fabricante: National Semiconductor (original)
- Voltaje de alimentación: 4V – 30V (5V recomendado)
- Rango de sensado temperatura: 0℃ hasta 100℃
- Precisión a temperatura ambiente: ±0.6°C
- Pendiente: 10mV/ºC
- Bajo consumo energético: 91.5uA

- Corriente de salida máxima: 10mA
- Pines: +VCC, V salida, GND
- Baja impedancia de salida

5.1.3.SENSOR ULTRASONICO

El sensor HC-SR04 es un sensor de distancia de bajo costo que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. El sensor HC-SR04 es el más utilizado dentro de los sensores de tipo ultrasonido, principalmente por la cantidad de información y proyectos disponibles en la web.

El sensor HC-SR04 posee dos transductores: un emisor y un receptor piezoeléctricos, además de la electrónica necesaria para su operación. El funcionamiento del sensor es el siguiente: el emisor piezoeléctrico emite 8 pulsos de ultrasonido(40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y asi se puede calcular la distancia al objeto. El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar).

La distancia se puede calcular utilizando la siguiente formula:

Distancia(m) = {(Tiempo del pulso ECO) * (Velocidad del sonido=340m/s)}/2

El sensor US-016 es similar al HC-SR04 pero con salida de tipo analógico, otro sensor ultrasonido es el sensor US-100 con salida de tipo uart/serial.

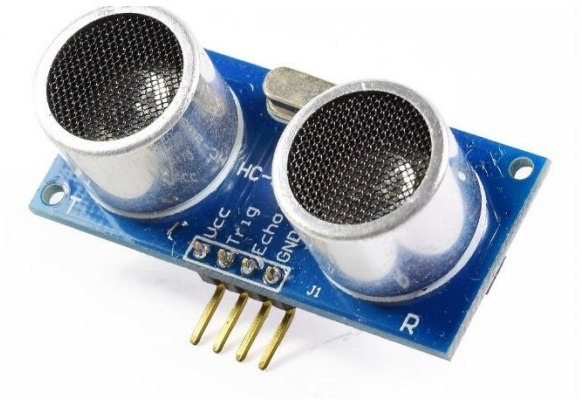
ESPECIFICACIONES TÉCNICAS

- Voltaje de Operación: 5V DC
- Corriente de reposo: < 2mA
- Corriente de trabajo: 15mA
- Rango de medición: 2cm a 450cm
- Precisión: +- 3mm
- Ángulo de apertura: 15°

- Frecuencia de ultrasonido: 40KHz
- Duración mínima del pulso de disparo TRIG (nivel TTL): 10 µS
- Duración del pulso ECO de salida (nivel TTL): 100-25000 µS
- Dimensiones: 45*20*15 mm
- Tiempo mínimo de espera entre una medida y el inicio de otra 20ms (recomendable 50ms)

CONEXIÓN

- VCC (+5V DC)
- TRIG (*Disparo del ultrasonido*)
- ECHO (*Recepción del ultrasonido*)
- GND (Tierra: 0V)



5.1.4.SENSOR DE GAS MQ135

DESCRIPCIÓN DEL PRODUCTO

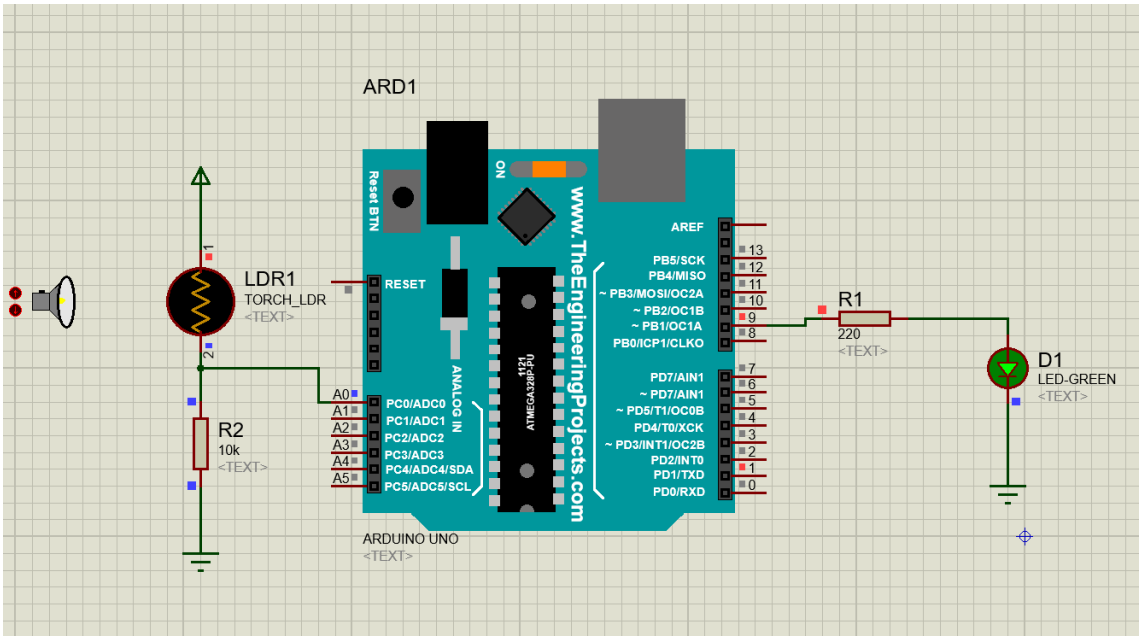
El MQ-135 es un sensor de gases peligrosos utilizado para el control de la calidad del aire y es adecuado para la detección de NH3 (amoníaco), NOx, alcohol, benceno, humo, CO2, etc.
Este sensor no proporciona valores absolutos, sino que simplemente proporciona una salida analógica que debe ser monitoreado y se comparada con los valores de umbral.

CARACTERÍSTICAS TÉCNICAS

- Chip principal: Sensor MQ-135 Air Quality Sensor.
- Doble salida: Salida analógica y salida de nivel TTL.
- Alta sensibilidad al Amoníaco (NH3), Óxidos de nitrógeno (NOx), Alcohol, Sulfuros, Benceno (C6H6), Monóxido de carbono (CO), humo y otros gases nocivos.
- Montado en módulo con pines de conexión.
- Sensibilidad ajustable con el potenciómetro.
- Rango de detección: 10-1000ppm.
- Tamaño: 32mm x 22mm x 24mm.
- Buena sensibilidad a los gases dañinos en amplia gama
- Alta sensibilidad a Amoníaco, Sulfuro y Benzeno
- Circuito de accionamiento simple
- Voltaje de trabajo: 5V.

PROGRAMACION

5.2. PROGRAMA SENSOR DE LUZ LDR



Encender un LED cuando la luz ambiental es baja usando un sensor LDR.

Materiales necesarios

1. Arduino Uno (o cualquier otra placa Arduino).
2. LED.
3. Resistencia de 220 ohmios (para el LED).
4. Sensor LDR (resistencia dependiente de la luz).
5. Resistencia de 10 kΩ (para formar un divisor de voltaje con el LDR).
6. Protoboard (opcional, para organizar mejor los componentes).
7. Cables de conexión.

Esquema de conexión

1. Circuito del LDR (Divisor de voltaje):
    - Conecta un extremo del LDR al pin de 5V de Arduino.
    - Conecta el otro extremo del LDR al pin A0 (entrada analógica) de Arduino.
    - Conecta una resistencia de 10 kΩ entre el pin A0 y GND.

Nota: Esto crea un divisor de voltaje donde el punto medio (conectado a A0) cambia de voltaje según la luz que incide en el LDR.

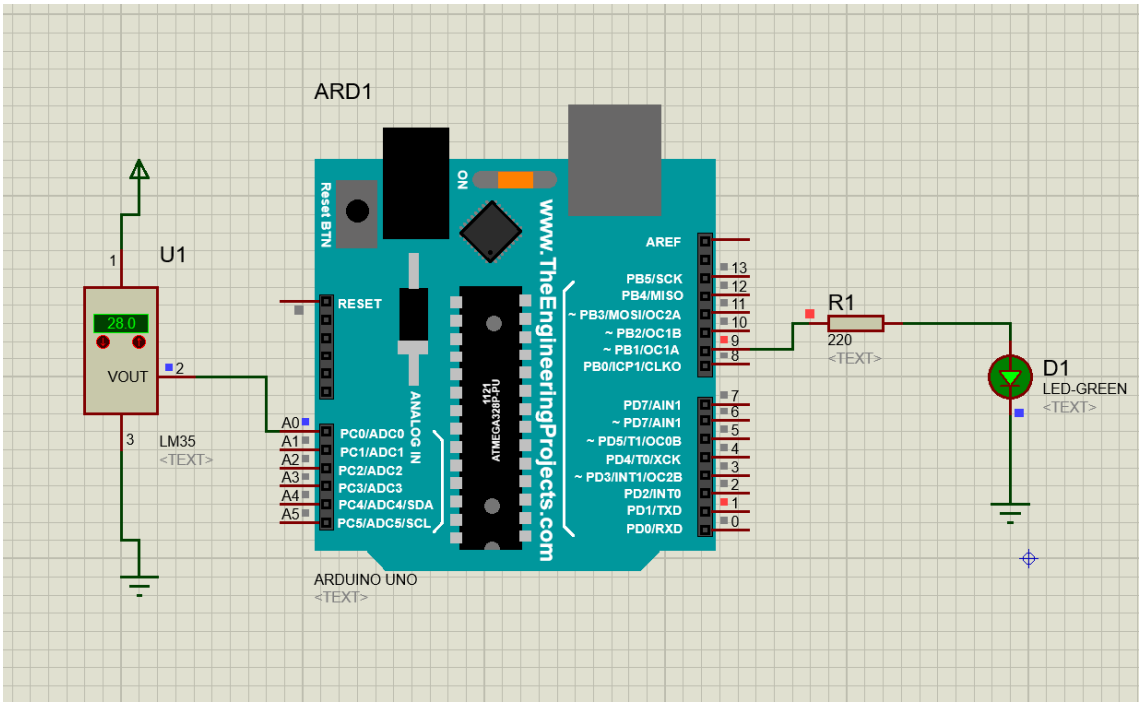
2. **Circuito del LED:**
   - Conecta el **ánodo** del LED (pata larga) al pin digital **9** de Arduino.
   - Conecta el **cátodo** del LED (pata corta) a una resistencia de **220 ohmios**.
   - Conecta el otro extremo de la resistencia a **GND**.

PROGRAMA EN ARDUINO

```
const int ldrPin = A0;   // Pin al que está conectado el LDR
const int ledPin = 9;    // Pin al que está conectado el LED
int ldrValue = 0;        // Variable para almacenar el valor leído del LDR
int threshold = 500;     // Umbral para determinar cuándo encender el LED

void setup() {
  pinMode(ledPin, OUTPUT); // Configurar el pin del LED como salida
  Serial.begin(9600);      // Inicializar comunicación serial para depuración
}

void loop() {
  // Leer el valor del LDR
  ldrValue = analogRead(ldrPin);

  // Mostrar el valor leído en el monitor serial
  Serial.print("Valor del LDR: ");
  Serial.println(ldrValue);

  // Comparar el valor con el umbral
  if (ldrValue < threshold) {
    // Si la luz es baja (valor menor al umbral), encender el LED
```

```
  digitalWrite(ledPin, HIGH);
 } else {
  // Si hay suficiente luz, apagar el LED
  digitalWrite(ledPin, LOW);
 }

 delay(500); // Esperar medio segundo antes de la siguiente lectura
}
```

5.3. SENSOR DE TEMPERATURA LM35



Materiales necesarios

1. Arduino Uno (o cualquier placa Arduino).
2. Sensor LM35.
3. Cables de conexión.
4. Protoboard (opcional).

Conexión del sensor LM35

El sensor LM35 tiene 3 pines:

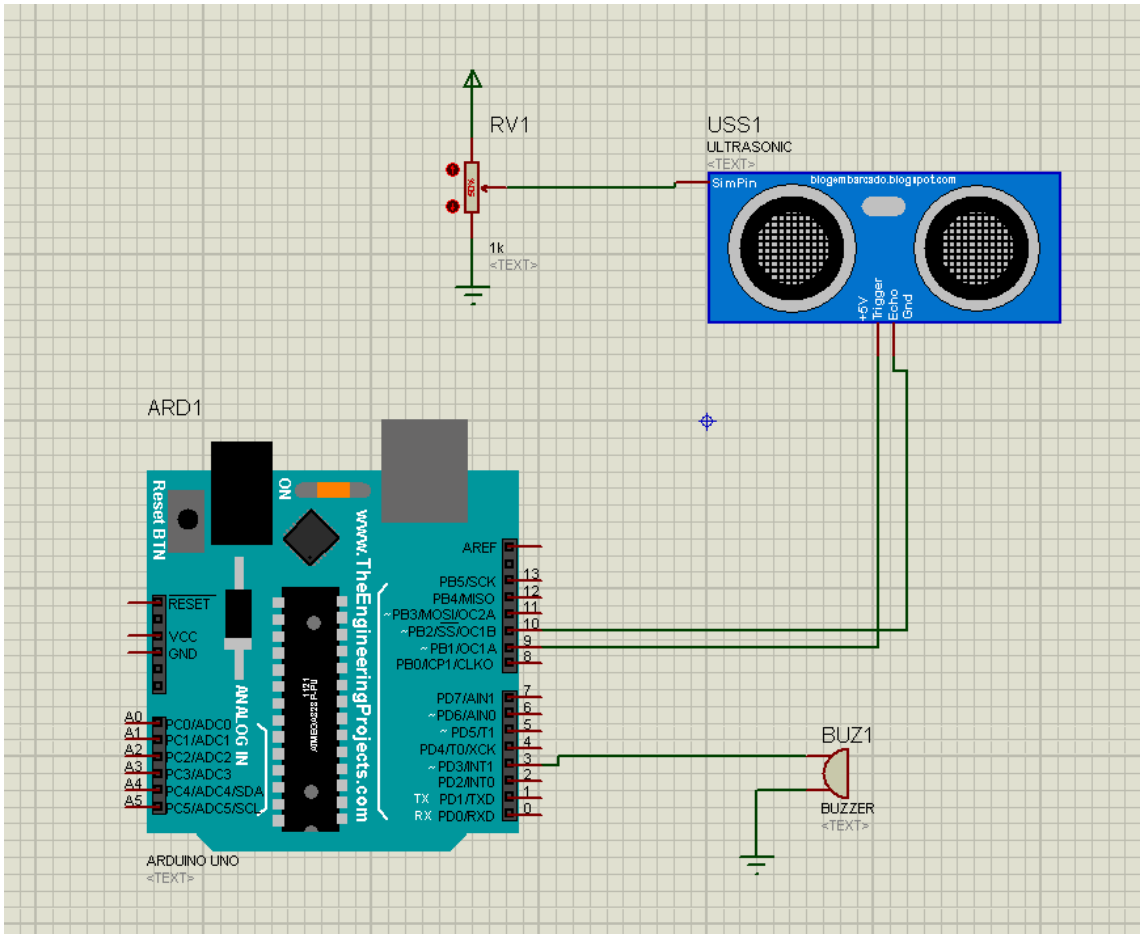
1. VCC (Pin 1): Conéctalo al pin de 5V de Arduino.
2. Salida (Pin 2): Conéctalo al pin analógico A0 de Arduino.
3. GND (Pin 3): Conéctalo al pin GND de Arduino.

PROGRAMA:

```
const int sensorPin = A0;  // Pin analógico al que está conectado el LM35
float temperatura;         // Variable para almacenar la temperatura

void setup() {
  pinMode(9, OUTPUT);
  Serial.begin(9600); // Inicializar comunicación serial para depuración
}

void loop() {
  // Leer el valor analógico del LM35 (0 a 1023)
  int lectura = analogRead(sensorPin);

  // Convertir la lectura analógica a voltaje (0 - 5V)
  float voltaje = lectura * (5.0 / 1023.0);

  // Convertir el voltaje a temperatura en grados Celsius
  temperatura = voltaje * 100.0;

  // Mostrar la temperatura en el monitor serial
  Serial.print("Temperatura: ");
  Serial.print(temperatura);
```

```
  Serial.println(" °C");
   // delay(1000); // Esperar 1 segundo antes de la próxima lectura
    if (temperatura > 23) {
      // Si la luz es baja (valor menor al umbral), encender el LED
      digitalWrite(9, HIGH);
    } else {
      // Si hay suficiente luz, apagar el LED
      digitalWrite(9, LOW);
    }
    delay(500); //
}
```

5.4.SENSOR ULTRASONICO



Materiales necesarios

1. **Arduino Uno** (o cualquier placa Arduino).
2. **Sensor ultrasónico HC-SR04**.
3. **Cables de conexión**.
4. **Protoboard** (opcional).

Especificaciones del HC-SR04

El HC-SR04 utiliza ultrasonido para medir la distancia. Sus principales pines son:

- **VCC:** Alimentación (+5V).
- **GND:** Tierra (0V).
- **TRIG:** Pin para enviar la señal ultrasónica.
- **ECHO:** Pin para recibir la señal reflejada.

Esquema de conexión

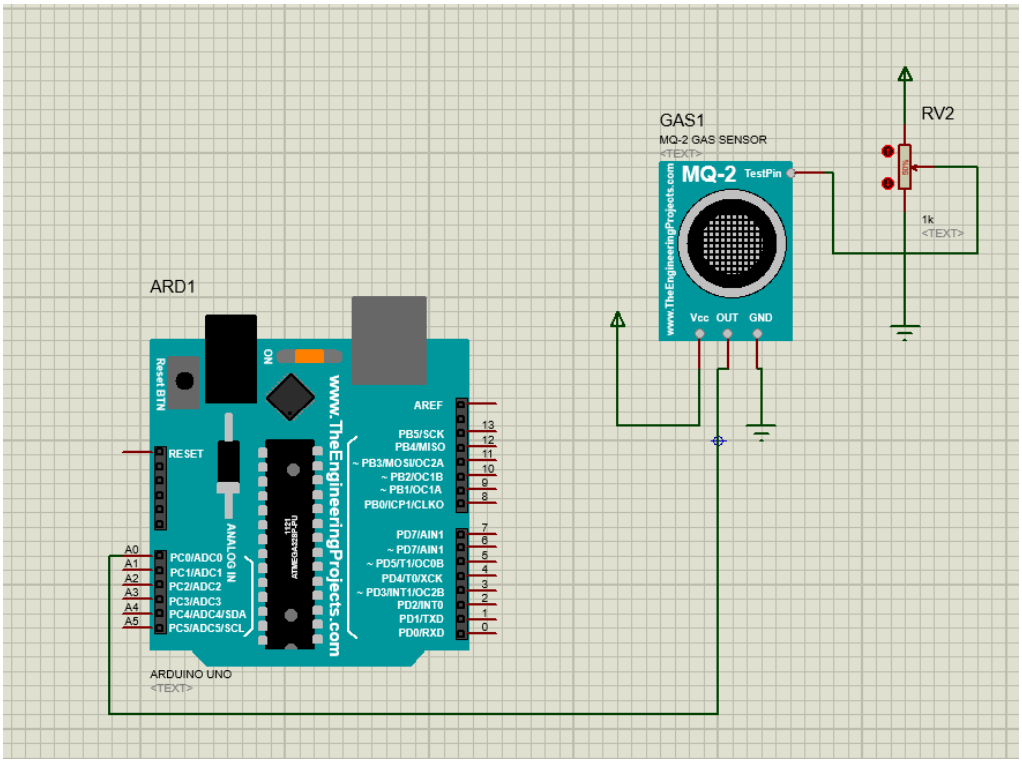
1. Conecta el pin **VCC** del sensor al pin de **5V** de Arduino.
2. Conecta el pin **GND** del sensor al pin **GND** de Arduino.
3. Conecta el pin **TRIG** al pin digital **9** de Arduino.
4. Conecta el pin **ECHO** al pin digital **8** de Arduino.

PROGRAMA:

```
// Pines del sensor ultrasónico
const int trigPin = 9;
const int echoPin = 10;
// Pin del buzzer
const int buzzerPin = 3;
// Umbral de distancia en centímetros
const int distanciaUmbral = 10;
void setup() {
  pinMode(trigPin, OUTPUT);
```

```
  pinMode(echoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  Serial.begin(9600); // Inicializar monitor serial
}
void loop() {
  // Generar un pulso en el pin Trigger
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Leer el tiempo que tarda en regresar el pulso al pin Echo
  long duracion = pulseIn(echoPin, HIGH);
  // Convertir el tiempo en distancia (en cm)
  long distancia = duracion * 0.034 / 2;
  // Mostrar la distancia en el monitor serial
  Serial.print("Distancia: ");
  Serial.print(distancia);
  Serial.println(" cm");
  // Activar el buzzer si la distancia es menor que el umbral
  if (distancia > 0 && distancia < distanciaUmbral) {
    digitalWrite(buzzerPin, HIGH); // Encender buzzer
  } else {
    digitalWrite(buzzerPin, LOW);  // Apagar buzzer
  }
  delay(200); // Pausa antes de la siguiente medición
}
```

5.5.SENSOR DE GAS MQ135



Materiales necesarios

1. Arduino Uno o similar.
2. Sensor MQ-135.
3. Protoboard.
4. Cables de conexión.
5. Resistor de 10 kΩ (opcional, si necesitas estabilizar la señal).
6. Fuente de alimentación externa (opcional, si necesitas más potencia).

Conexiones

El sensor MQ-135 tiene 4 pines:

1. **VCC**: Conecta al pin de 5V del Arduino.
2. **GND**: Conecta al pin GND del Arduino.
3. **AO**: Salida analógica, conecta al pin analógico A0 del Arduino.
4. **DO** (opcional): Salida digital, no se usará en este ejemplo básico.

PROGRAMA

```
void setup() {
  Serial.begin(9600); // Iniciar comunicación serial
  Serial.println("Iniciando sensor MQ-135...");
  delay(2000); // Esperar a que el sensor se estabilice
}
void loop() {
  int sensorValue = analogRead(A0); // Leer valor analógico del sensor
  float voltage = sensorValue * (5.0 / 1023.0); // Convertir a voltaje
  Serial.print("Valor bruto: ");
  Serial.print(sensorValue);
  Serial.print("  Voltaje: ");
  Serial.print(voltage);
  Serial.println(" V");
    delay(1000); // Esperar 1 segundo
}
```