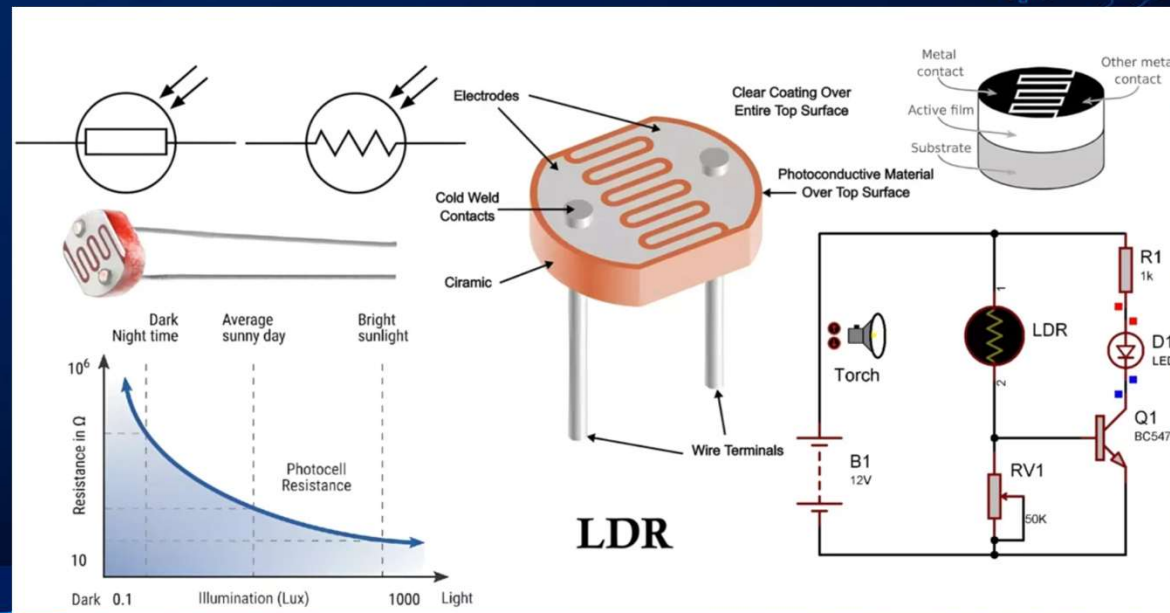


Light Dependent Resistor (LDR)

A Light Dependent Resistor (LDR) is a sensor that changes its resistance according to the intensity of light falling on it.

- More Light → Lower Resistance
- Less Light → Higher Resistance



```
const int ldrPin = 34;

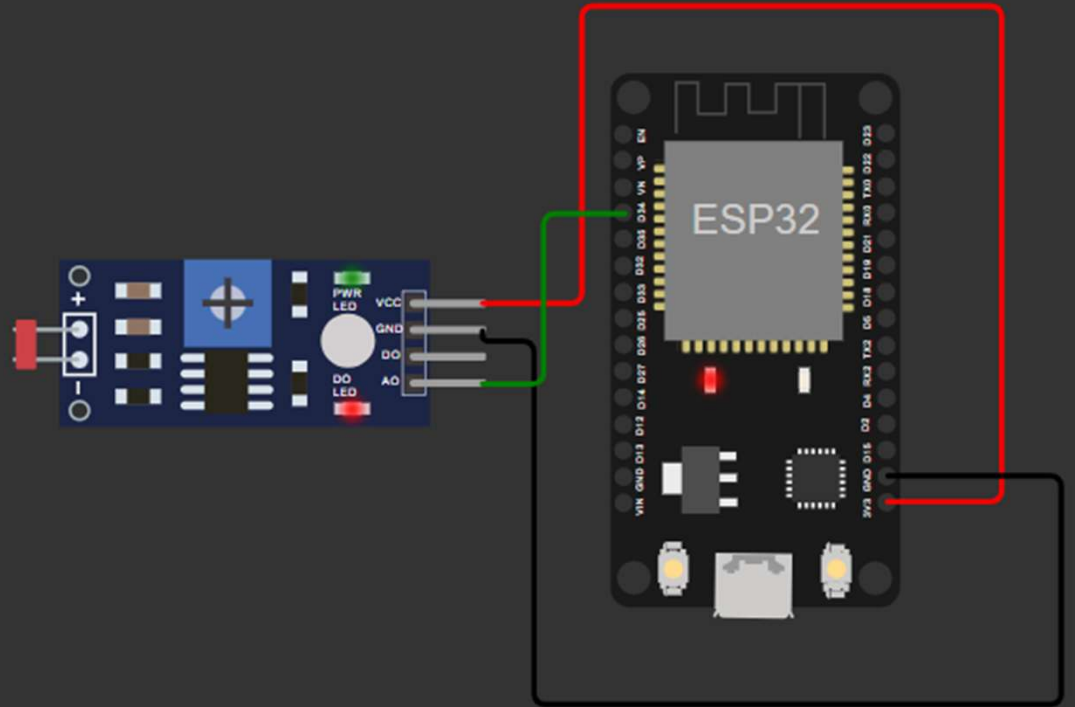
void setup() {
  Serial.begin(115200);
}

void loop() {

  int lightValue = analogRead(ldrPin);

  Serial.print("Light Value: ");
  Serial.println(lightValue);

  delay(500);
}
```





Bonus Challenge

Modify the program to include the following:

Add an **RGB LED**:

● Green → Day

● Red → Night

Add a **Buzzer**:

Beep once when it changes from Day to Night.

Display the light intensity as a percentage (0%–100%) in the Serial Monitor.



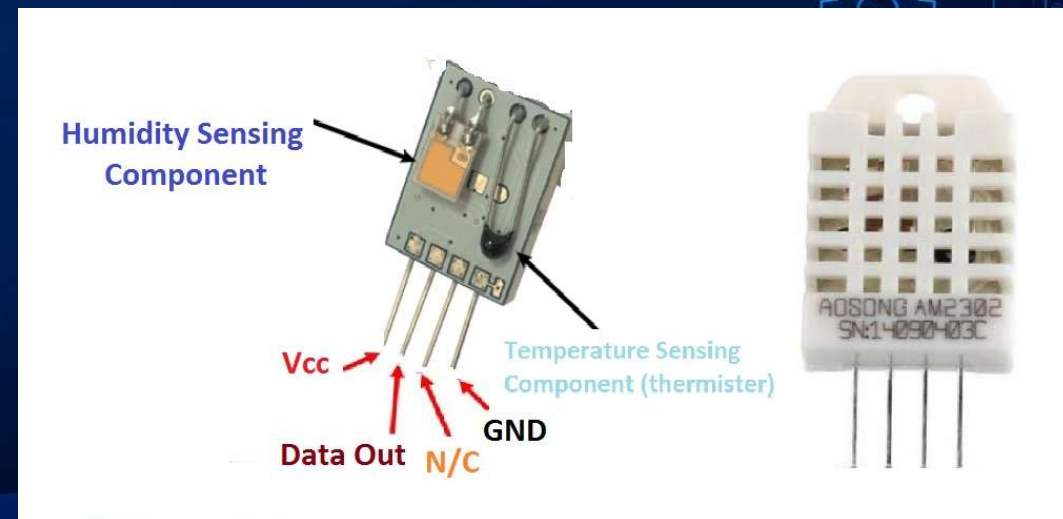


DHT22 Temperature & Humidity Sensor

The DHT22 is a digital sensor that measures:

- 🌡 Temperature (°C)
- 💧 Humidity (%RH)

Unlike analog sensors, it sends data digitally using a single data pin.





```
#include <DHT.h>

#define DHTPIN 15
#define DHTTYPE DHT22

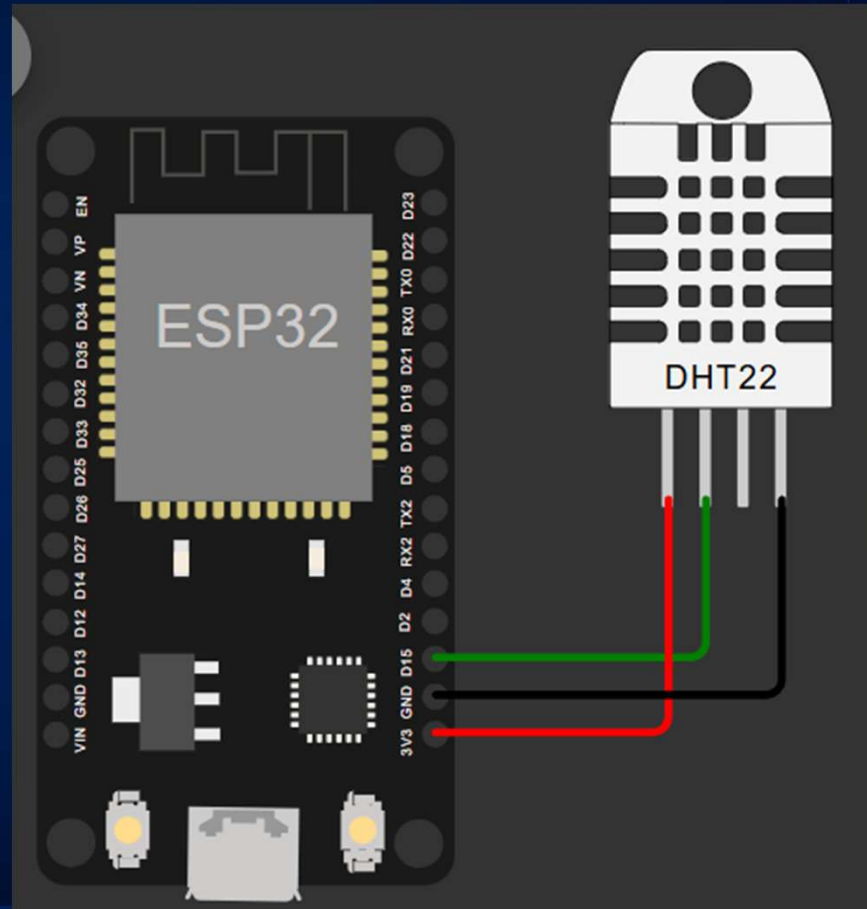
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  dht.begin();
}

void loop() {
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();

  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.print(" °C   Humidity: ");
  Serial.print(hum);
  Serial.println(" %");

  delay(2000);
}
```





Temperature Alert System

Create an alert system based on the temperature.

Requirements

- If Temperature $> 35^{\circ}\text{C}$
 - 🔴 Turn the LED ON
 - 🔔 Activate the buzzer
 - Display "High Temperature Alert!"
- Otherwise
 - 🟢 LED OFF
 - 🔔 Buzzer OFF
 - Display "Temperature Normal"





Ultrasonic Sensor (HC-SR04)

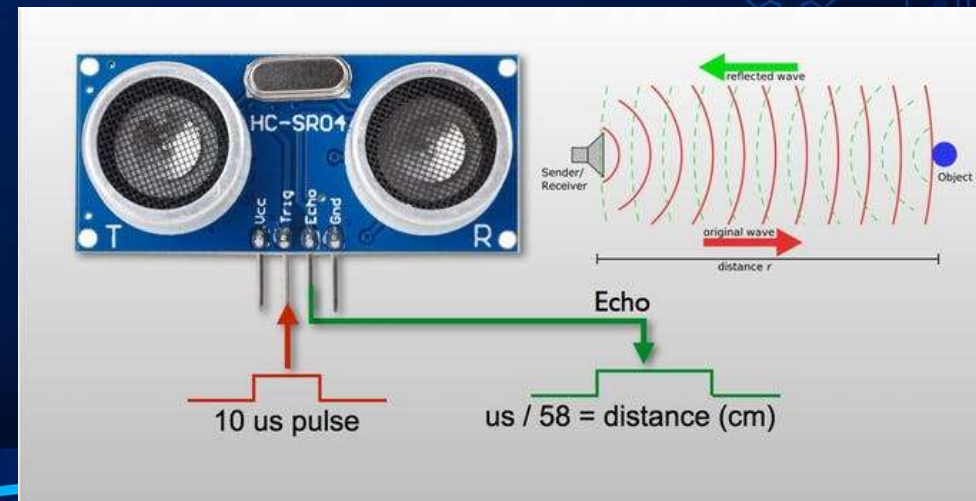
The HC-SR04 is a non-contact distance sensor that measures the distance between the sensor and an object using ultrasonic sound waves (40 kHz).





How It Works

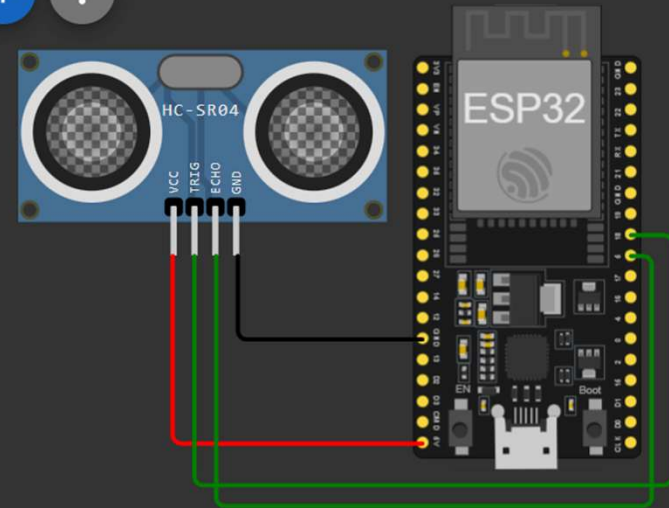
1. The **TRIG** pin sends an ultrasonic pulse.
2. The sound wave travels through the air.
3. It reflects back after hitting an object.
4. The **ECHO** pin receives the reflected signal.
5. The ESP32 calculates the distance based on the travel time.



sketch.ino • diagram.json • Library Manager ▾

```
1  #define TRIG 18
2  #define ECHO 5
3
4  void setup() {
5      Serial.begin(115200);
6      pinMode(TRIG, OUTPUT);
7      pinMode(ECHO, INPUT);
8  }
9
10 void loop() {
11     digitalWrite(TRIG, LOW);
12     delayMicroseconds(2);
13     digitalWrite(TRIG, HIGH);
14     delayMicroseconds(10);
15     digitalWrite(TRIG, LOW);
16
17     long duration = pulseIn(ECHO, HIGH);
18     int distance = duration / 58 - 4;
19
20     Serial.print("Distance: ");
21     Serial.print(distance);
22     Serial.println(" cm");
23     delay(500);
24 }
```

Simulation





Challenge : Distance Alert System

Objective

Turn an LED ON when an object comes close to the sensor.

Requirements

- Distance < 20 cm $\rightarrow$ LED ON
- Distance ≥ 20 cm $\rightarrow$ LED OFF



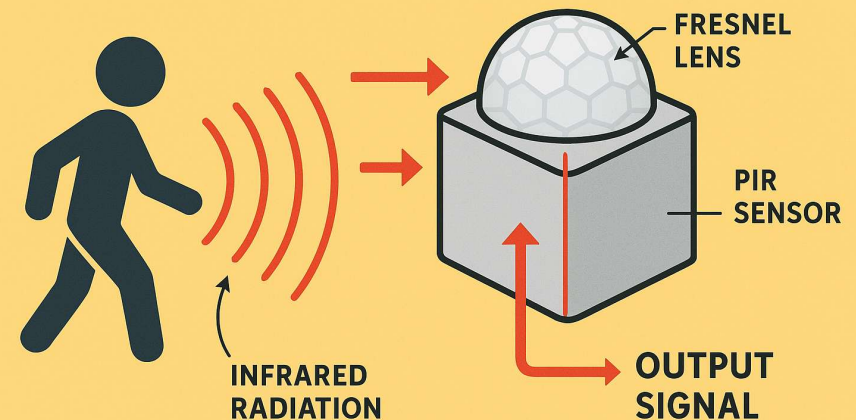
PIR Motion Sensor

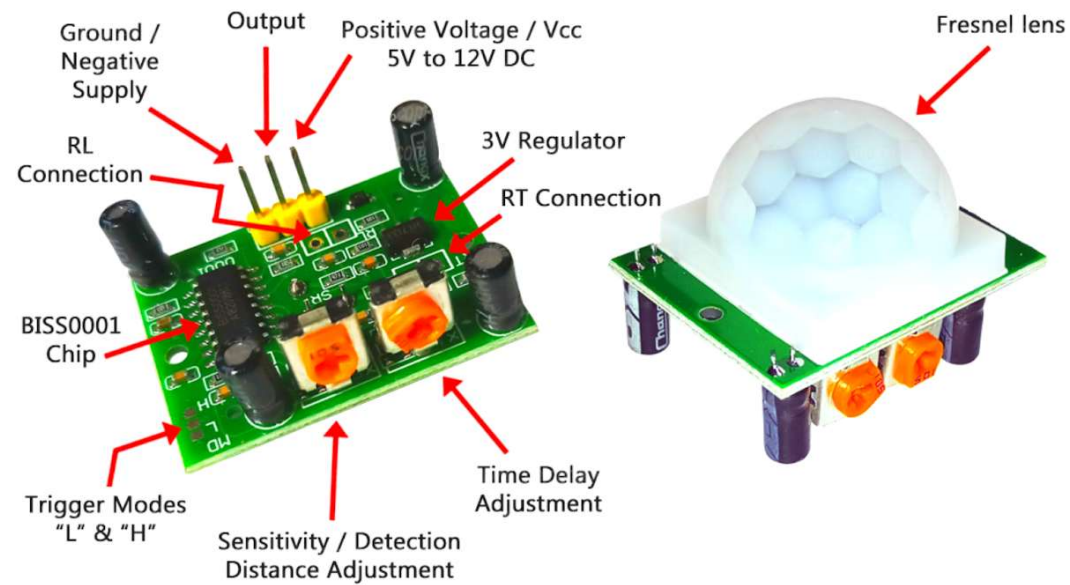
A Passive Infrared (PIR) sensor is an electronic device that measures infrared (IR) light radiating from objects in its field of view. Because warm bodies (like humans and animals) emit heat, the sensor detects movement by picking up changes in infrared energy. They are widely used in burglar alarms, automatic lighting, and smart home systems

Applications

- Home Security Systems
- Automatic Lighting
- Motion Detection
- Smart Door Systems
- Energy Saving

HOW THE PIR SENSOR WORKS



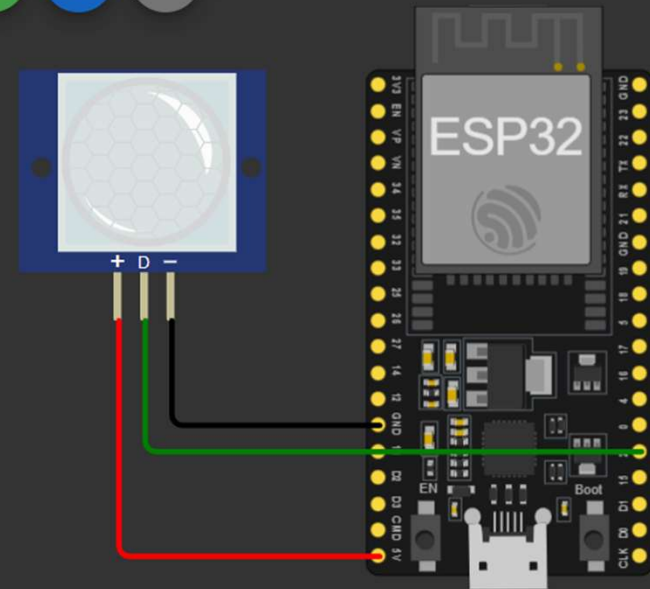




sketch.ino • diagram.json • Library Manager ▾

```
1  #define PIR 2
2
3  void setup() {
4      Serial.begin(115200);
5      pinMode(PIR, INPUT);
6  }
7
8  void loop() {
9
10     int motion = digitalRead(PIR);
11
12     if (motion == HIGH) {
13         Serial.println("Motion Detected");
14     }
15     else {
16         Serial.println("No Motion");
17     }
18
19     delay(500);
20 }
```

Simulation





Challenge : Smart Security Alarm

Requirements

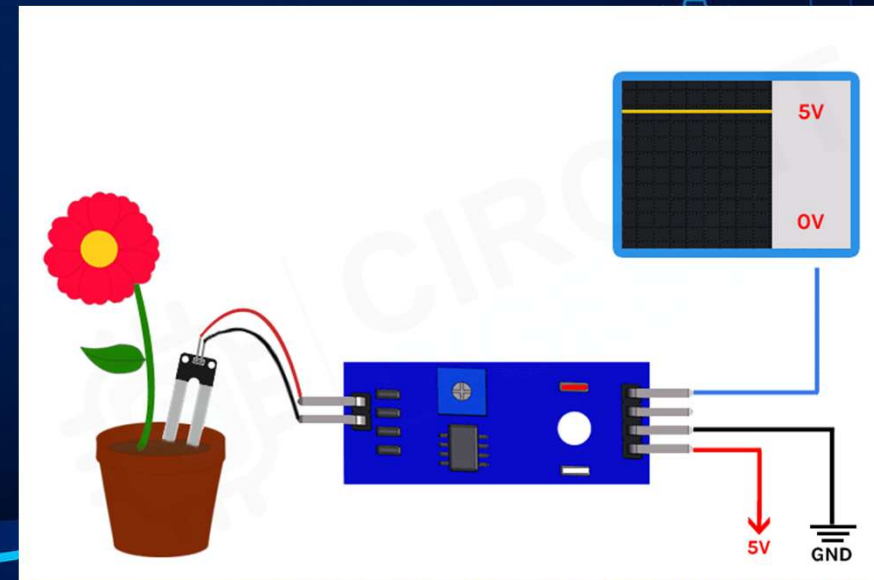
- Motion detected → Turn ON the LED.
- Motion detected → Activate the buzzer for 2 seconds.
- Display "**Intruder Detected!**" on the Serial Monitor.
- No motion → Turn OFF the LED and buzzer.

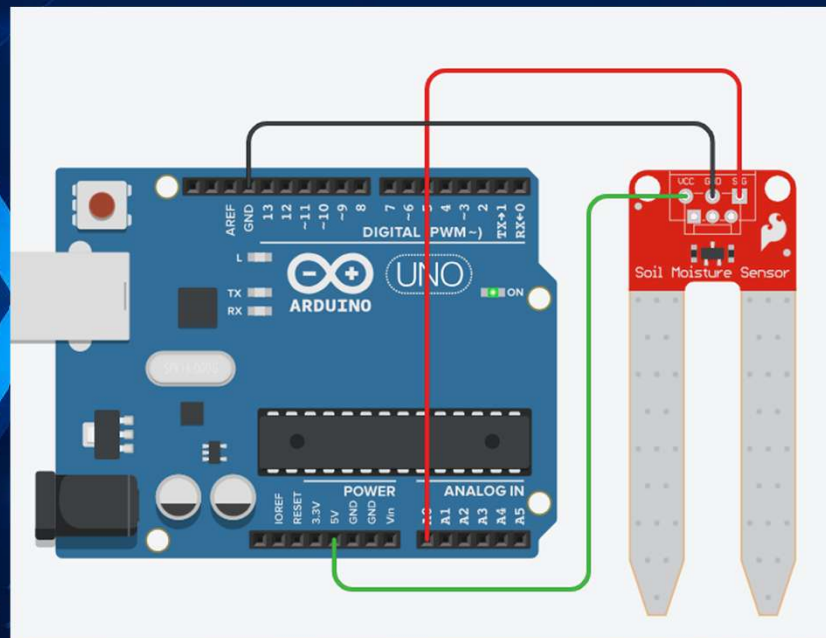




Soil Moisture Sensor

A soil moisture sensor measures the volumetric water content in soil. Typically used for smart farming, greenhouses, and DIY automated irrigation systems, they operate via electrical resistance or capacitance. Prices start around ₹90 for basic Arduino modules.





```
#define SOIL_PIN A0

void setup() {
  Serial.begin(9600);
}

void loop() {

  int moisture = analogRead(SOIL_PIN);

  Serial.print("Soil Moisture: ");
  Serial.println(moisture);

  delay(500);
}
```





Automatic Plant Watering System

Objective

Build a simple automatic irrigation system using the Soil Moisture Sensor.

Requirements

- Read the soil moisture value.
- If the soil is **dry**, turn the LED **ON**.
- If the soil is **wet**, turn the LED **OFF**.





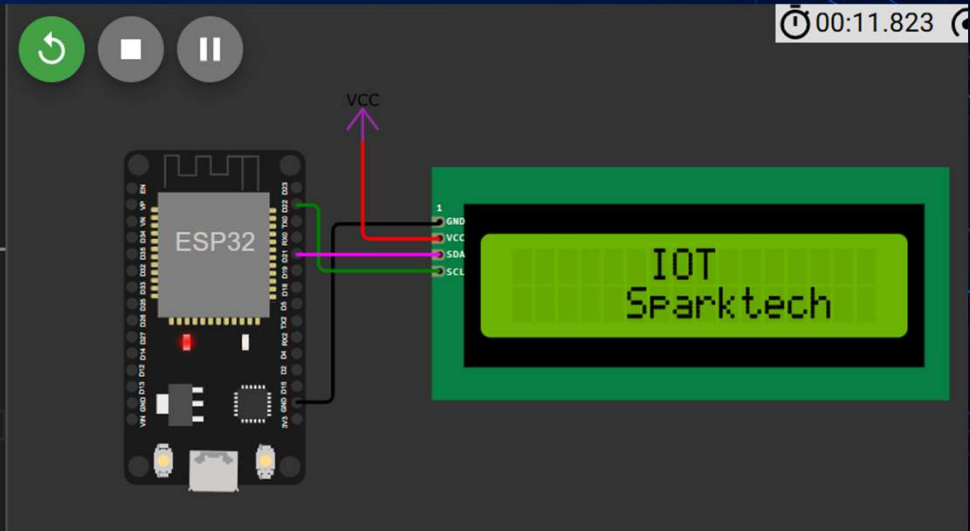
LCD 16×2 Display (I2C)

An **LCD 16×2 Display with an I2C interface** is a popular alphanumeric screen used in electronics to display 16 characters per row across 2 rows. By integrating a **PCF8574 I2C adapter backpack**, this setup drastically simplifies project wiring by reducing the required microcontroller pins from 16 down to just 4.





```
1 #include <LiquidCrystal_I2C.h>
2
3 LiquidCrystal_I2C lcd(0x27, 16, 2);
4 void setup() {
5     lcd.init(); // initialize the lcd
6     lcd.backlight();
7 }
8
9 void loop() {
10     lcd.clear();
11     lcd.setCursor(6, 0);
12     lcd.print("IOT");
13     lcd.setCursor(5, 1);
14     lcd.print("Sparktech");
15     delay(2000);
16 }
17
```





Objective

Create a live monitoring dashboard on the LCD.

Requirements

Integrate any Sensor and Show anything on display.

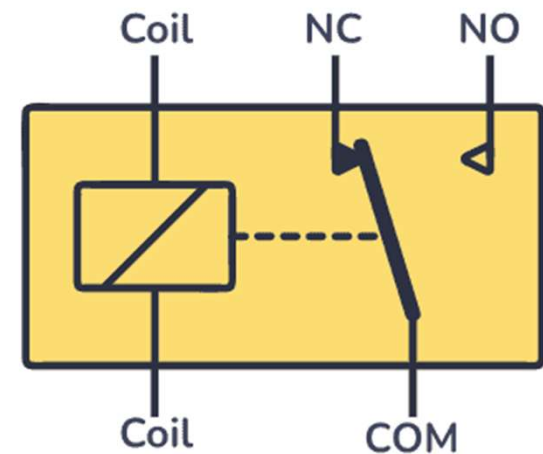
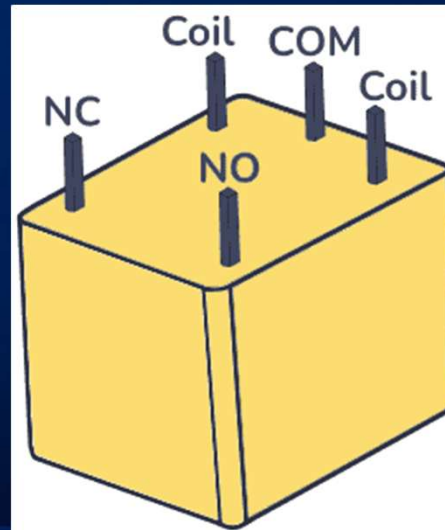


Relay Module

A relay is an electrically operated switch that allows a low-voltage microcontroller like the ESP32 to control high-voltage AC or DC devices safely.

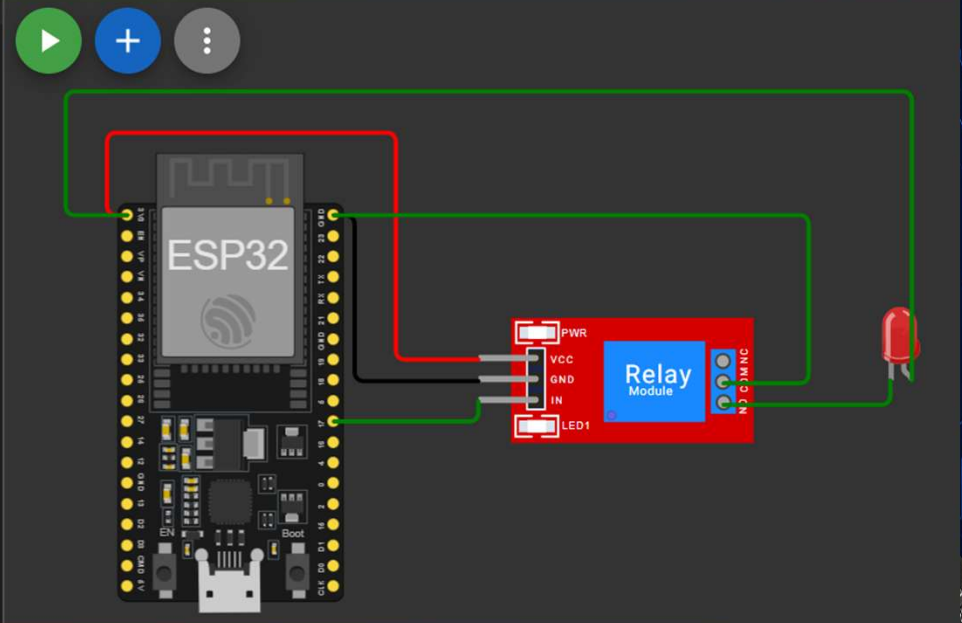
Applications

- Home Automation
- Smart Switches
- Motor Control
- Water Pump Control
- Industrial Automation
- Smart Irrigation





```
void setup() {  
  
  pinMode(17, OUTPUT);  
  // put your setup code here, to run once:  
  Serial.begin(115200);  
  Serial.println("Hello, ESP32!");  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  delay(1000); // this speeds up the simulation  
  digitalWrite(17, HIGH);  
  delay(1000);  
  digitalWrite(17, LOW);  
}
```





Smart Automatic Room Light Using Relay

Objective

Create an automatic room light controller using an **LDR Sensor** and a **Relay Module**.

Requirements

- Read the light intensity using the LDR.
- If it becomes **dark**, turn the relay **ON** (Light ON).
- If it becomes **bright**, turn the relay **OFF** (Light OFF).
- Display the light value and relay status on the Serial Monitor.



A top-down photograph of two open hands, palms facing up, holding a small, rectangular piece of white paper with deckled edges. The paper is centered between the fingers and contains the handwritten text "Thank you!" in a dark, cursive script. The hands are positioned against a dark, textured background that resembles slate or stone. The lighting is soft, highlighting the skin tones and the texture of the paper and background.

Thank you!