LDR Module

LDR Sensor Module (Light Dependent Resistor – Analog Mode)

Pinout of LDR Module

LDR Module Pin	Description
VCC	Power Supply (3.3V–5V)
GND	Ground
A0	Analog Output (connect to ESP32 analog pin, e.g., GPIO 34)
D0	Digital Output (leave unconnected)

Formulas to be used:

Voltage at A_O pin is:

$$V_{OUT} = V_{CC} \times \frac{R_{LDR}}{R_{LDR} + R_{POT}}$$

V_{out} from LDR Module becomes V_{in} to ESP32:

$$ADC\ (Value) = 4095 \times \frac{V_{in}}{V_{CC}}$$

$\mathrm{ADC(Value)}: 12 - \mathrm{bit\ ADC} -> \ 0 - 4095$

How It Works (Analog Mode):

- The LDR changes resistance based on light intensity:
- In bright light → LDR resistance ↓ → divider output voltage at A0 ↓
- In darkness → LDR resistance ↑ → output voltage at A0 ↑
- A voltage divider converts the resistance change into an analog voltage.

- **This voltage is read by an ADC pin (like GPIO 35 on ESP32) to measure light level.**

Points to Remember:

- **Use analogRead() on ESP32 to get light values between 0 and 4095 (12-bit ADC).**
- **You can map or scale these values for controlling:**
  - **LEDs (dim with light)**
  - **Buzzer alerts in darkness**
  - **Screens, blinds, street lights, etc.**
- **Do not connect digital (D0) pin if using analog mode only.**
- **Light level range depends on LDR sensitivity and ambient conditions.**

Example Use Cases

- **Automatic street lighting (ON when it's dark)**
- **Day/night detection for smart blinds, fans**
- **Solar tracker systems (track sun position using 2 LDRs)**
- **Light-activated alarms**
- **Ambient brightness control for screens, displays**

CODE:

```
#define LDR_PIN 35 //Analog pin
void setup() {
  Serial.begin(115200);
  delay(1000);
}
void loop() {
  float ldrValue = analogRead(LDR_PIN);
  Serial.print("LDR Value: ");
  Serial.println(ldrValue);
  float percentage = (ldrValue / 4095.0) * 100.0;
  Serial.print("Light Intensity: ");
  Serial.print(percentage);
  Serial.println(" %");
  delay(2000);
}
```

IR Module (Active Low)

Pinout of IR Sensor Module (Typically):

IR Sensor Pin	Description
VCC	Power Supply (3.3V–5V)
GND	Ground
OUT	Digital Output (GPIO 4)

How It Works (Active Low):

- No obstacle → OUT = HIGH (1)
- Obstacle in front → OUT = LOW (0)

Points to remember:

- ✓ Reflective surfaces like white or glossy objects reflect IR better → longer detection range.
- ✓ Dark or matte surfaces absorb IR → shorter range.

Example Use Cases:

- Line-following robot (detect black/white line)
- Object counting
- Simple obstacle avoidance
- Edge detection for a table or staircase

CODE:

```
#define PIR 4
#define LED 2
void setup() {
  Serial.begin(115200);
  pinMode(PIR, INPUT);
  pinMode(LED, OUTPUT);
}
void loop() {
  int obstacle = digitalRead(PIR);
  if(obstacle == LOW){
    digitalWrite(LED, HIGH);
    Serial.println("Obstacle Detected");
  }
  else{
    digitalWrite(LED, LOW);
    Serial.println("No Obstacle");
  }
  delay(500);
}
```

PIR Module (HCSR501)

Pinout of HC-SR501 Module (Typical)

HC-SR501 Pin	Description
VCC	Power Supply (5 V recommended, 4.5 – 20 V tolerated)
OUT	Digital Output (goes HIGH on motion, ≈3.3 V logic level)
GND	Ground

Extra adjustments

- **Potentiometer_1 → *Sensitivity / Range* (≈3 m → 7 m)**
- **Potentiometer_2 → *Time-delay* after motion (≈0.3 s → 5 min)**

How It Works (Active High)

- **No motion in view → OUT = LOW (0 V)**
- **Human/animal motion detected → OUT = HIGH (≈3.3 V)**

The module contains a pyroelectric infrared (PIR) sensor with a Fresnel lens. When a warm body (≈ human skin temperature) moves across its field of view, the sensor sees a rapid change in IR energy, triggers the comparator, and drives OUT HIGH.

Points to Remember

- **Needs ~60 s warm-up after power-on for stable readings.**
- **Detects motion of warm objects—static people or objects are ignored.**
- **Strong sunlight, heaters, or moving air can cause false triggers; shield when outdoors.**

Example Use Cases

- **Automatic lighting (lights ON only when someone is present)**

- **Security alarms and motion-activated cameras**
- **Smart doorbell**

Connect VCC → 5 V, GND → GND, OUT → any digital GPIO (or GPIO 4). With the ESP32, read digitalRead(pin)—HIGH means motion detected

---

CODE:

```
#define PIR_PIN 4
#define LED_PIN 2
void setup() {
  Serial.begin(115200);
  pinMode(PIR_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
  Serial.println("PIR Sensor Test Initialized...");
}
void loop() {
  int motionDetected = digitalRead(PIR_PIN);
  if (motionDetected) {
    Serial.println("Motion Detected!");
    digitalWrite(LED_PIN, HIGH);
  } else {
    Serial.println("No Motion");
    digitalWrite(LED_PIN, LOW);
  }
  delay(200);
}
```