PIR MOTION SENSOR HC-SR501

Introduction:

The **HC-SR501 PIR Motion Sensor** is widely used for motion detection in security systems, home automation, and IoT applications. It detects infrared radiation changes caused by movement and outputs a signal accordingly. This document details its working principle, standalone testing with an oscilloscope, and integration with **ESP8266MOD (NodeMCU)**.

HC-SR501 PIR Motion Sensor Overview:

Specifications:

- **Operating Voltage:** 4.5V - 20V

- **Detection Angle:** Up to 120°
- **Detection Range:** 3m - 7m (Adjustable)
- **Delay Time:** 5s - 300s (Adjustable)
- **Trigger Modes:** Retriggerable and Non-Retriggerable

Working Principle:

The PIR sensor consists of two infrared-sensitive elements. When a moving object (human, animal) enters its detection range, the difference in infrared levels triggers a **HIGH** output signal. When no motion is detected, the output remains **LOW**.

Reference:

https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf

PIR Sensor Testing with MSO/DSO (Without ESP8266MOD):

Before integrating the PIR sensor with ESP8266, it was tested using an **MSO (Mixed Signal Oscilloscope) and DSO (Digital Storage Oscilloscope)** to analyze its output behavior.

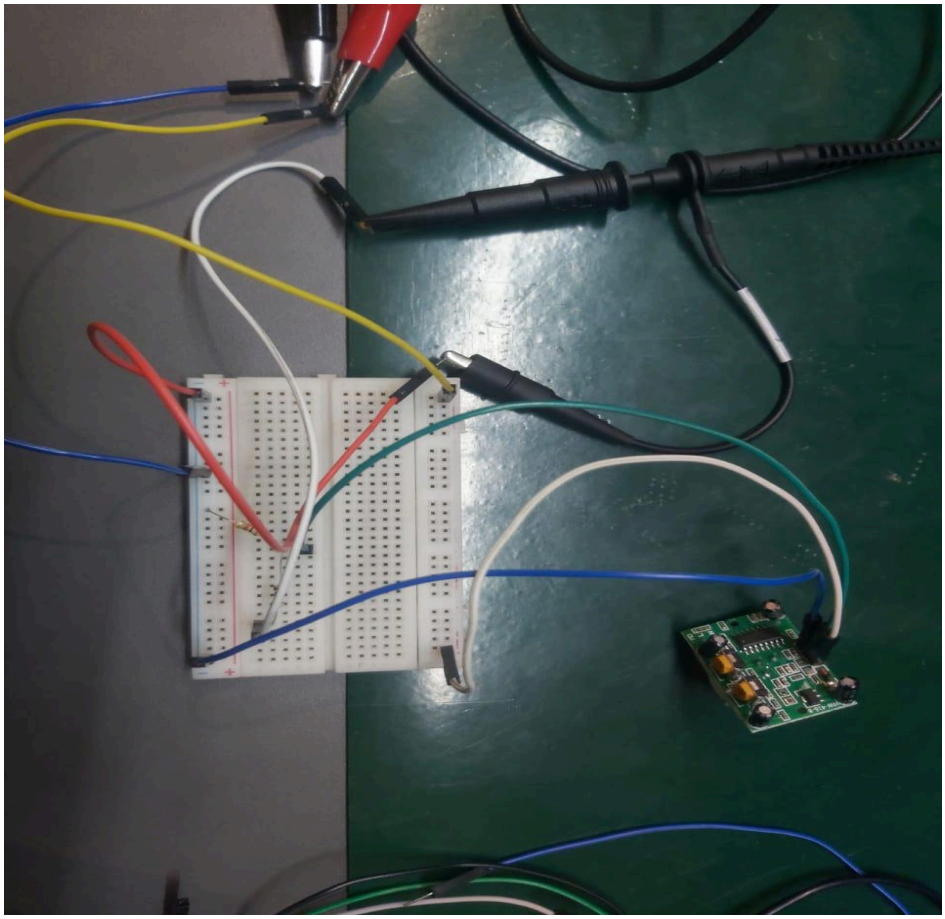
Required Components:

- HC-SR501 PIR Motion Sensor
- Oscilloscope (MSO/DSO)
- Power Supply (5V)
- Jumper Wires
- 1kΩ resistor (added for signal conditioning)

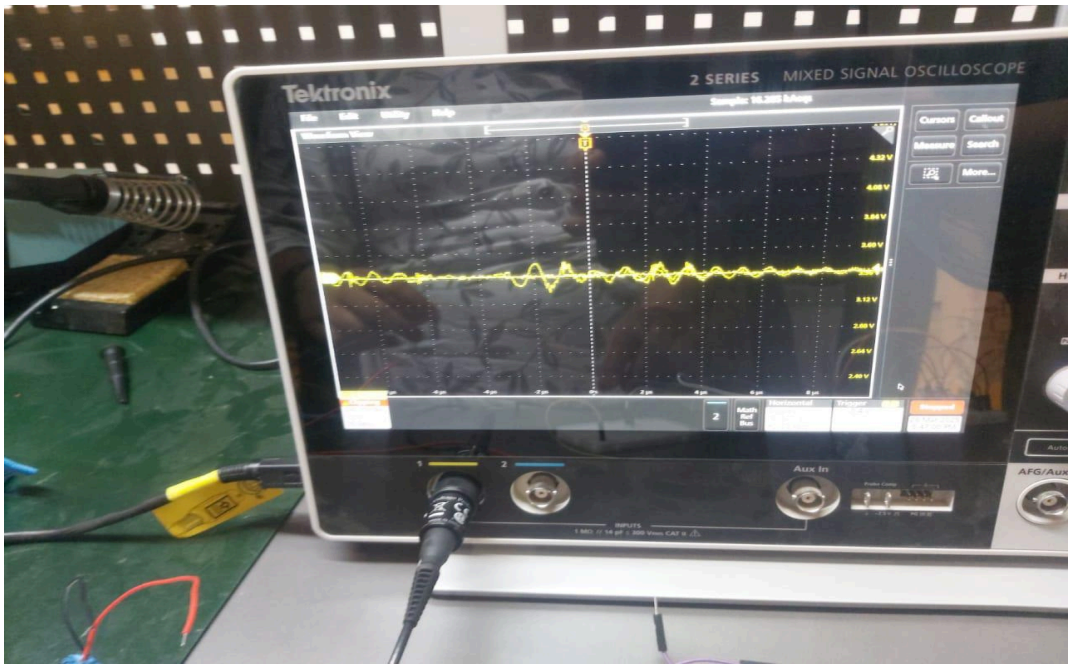
- 10kΩ pull-down resistor (added after issue occurred)

Connections:

- **VCC** → 5V power supply
- **GND** → Ground
- **OUT** → Oscilloscope probe
- **1kΩ resistor** was connected between **OUT and GND** initially for signal stabilization.
- **10kΩ pull-down resistor** was added later when an issue occurred, ensuring the output returned to **LOW** correctly.



MSO/DSO Output Analysis & Troubleshooting:



Issues & Solutions:

No Output Signal on MSO/DSO Initially:

- **Fix:** Added a **10kΩ pull-down resistor** between OUT and GND.
- **Reason:** The PIR sensor's output was floating, and the resistor ensured a proper LOW state when no motion was detected.

Arduino IDE Installation & Board Setup:

To program the **ESP8266MOD (NodeMCU)**, we use the **Arduino IDE** with additional board support.

Steps to Install Arduino IDE:

1. Download and install the **Arduino IDE** from the official website: https://www.arduino.cc/en/software.

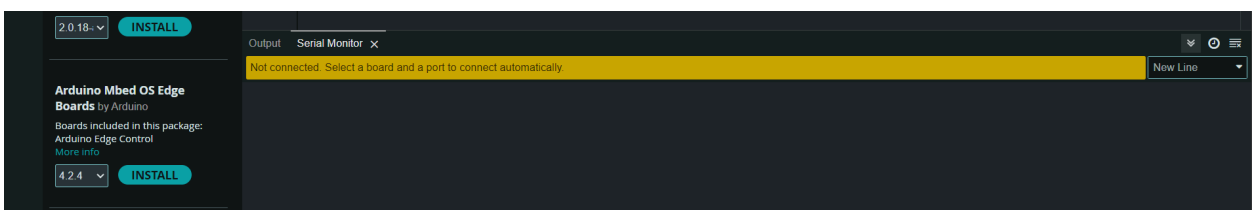
2. Open Arduino IDE and go to **File** → **Preferences**.
3. In **Additional Board Manager URLs**, add: http://arduino.esp8266.com/stable/package_esp8266com_index.json
4. Go to **Tools** → **Board** → **Boards Manager** and search for **ESP8266**.
5. Click **Install** and wait for the installation to complete.

Board & Port Selection:

1. Connect **NodeMCU** via USB.
2. In **Arduino IDE**, go to **Tools** → **Board** → **ESP8266** → **NodeMCU 1.0 (ESP-12E Module)**.
3. Select the correct **COM Port** under **Tools** → **Port**

Common Issues & Fixes:

1.Board Not Found in Boards Manager:



- Ensure the **Board Manager URL is correct** and internet connection is stable.

2. Error: "esptool.FatalError: Failed to connect to ESP8266"

- Hold the **Flash button** on NodeMCU while uploading code or go to device manager and and ports select ports and and uninstall the already existing library and reinstall it and restart the system.

PIR Sensor Integration with ESP8266MOD:

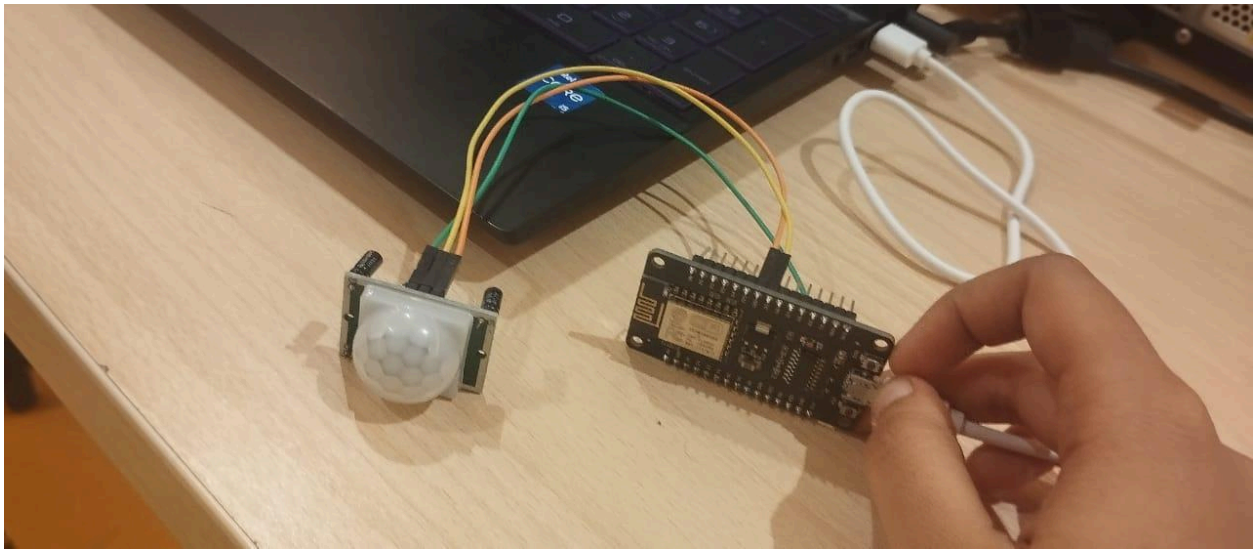
After confirming the PIR sensor's standalone functionality, it was interfaced with **ESP8266MOD (NodeMCU)** for IoT-based applications.

Required Components:

- ESP8266MOD (NodeMCU)
- HC-SR501 PIR Motion Sensor
- Jumper Wires

Connections:

- **VCC** → VIN (5V) on ESP8266
- **GND** → GND on ESP8266
- **OUT** → GPIO **D5** on ESP8266



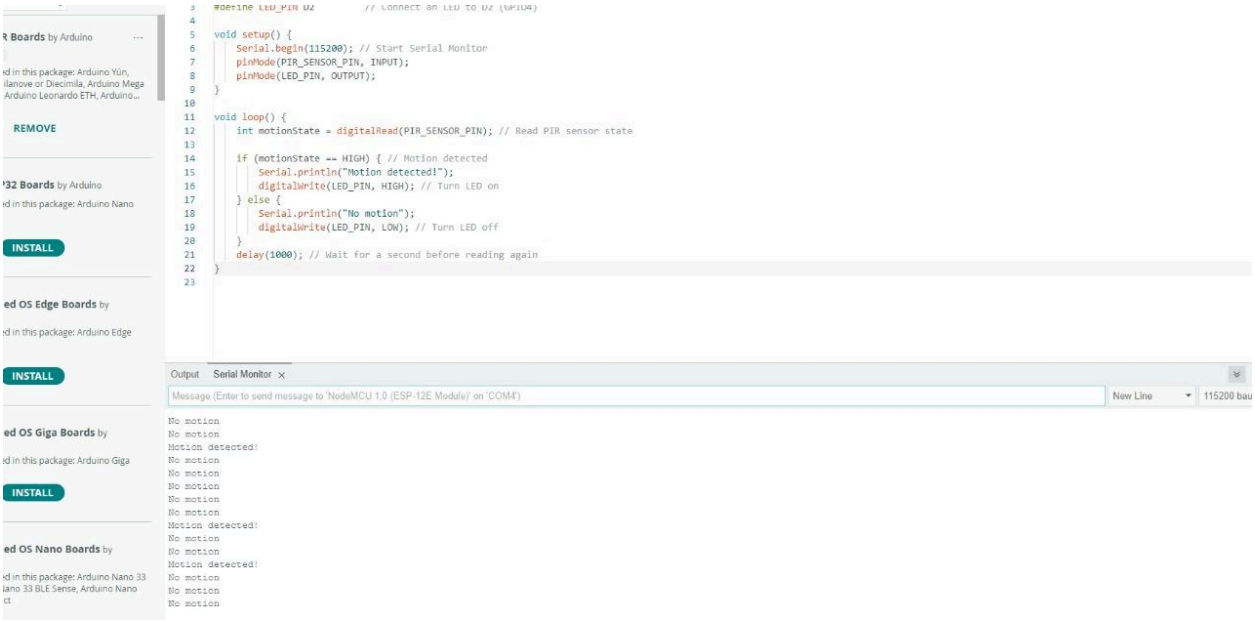
Code Implementation:

Code:

```
#define PIR_SENSOR_PIN D1
#define LED_PIN D2
void setup() {
   Serial.begin(115200); // Start Serial Monitor
```

```
    pinMode(PIR_SENSOR_PIN, INPUT);
    pinMode(LED_PIN, OUTPUT);
}

void loop()
 {
    int motionState = digitalRead(PIR_SENSOR_PIN);
    if (motionState == HIGH)
          {
           Serial.println("Motion detected!");
           digitalWrite(LED_PIN, HIGH);
          }
else
        {
        Serial.println("No motion detected");
        digitalWrite(LED_PIN, LOW);
       }
 delay(1000);
}
```

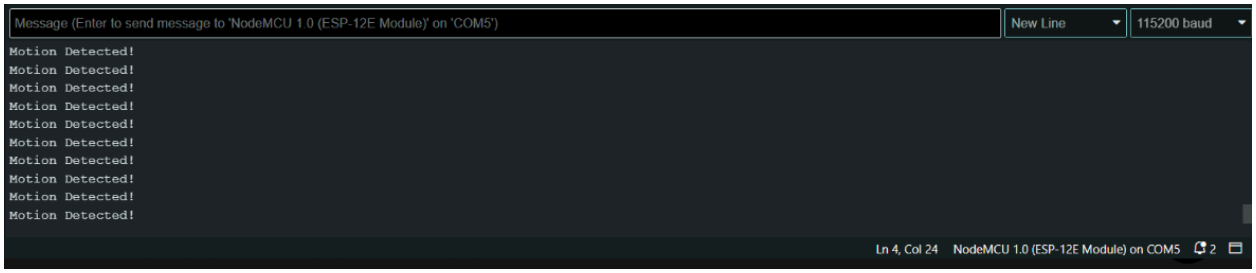


Output:

Output Analysis & Troubleshooting:

Common Issues & Fixes:

1.PIR Sensor Always Shows "Motion Detected!"



- Reduce PIR sensitivity.
- Add a **10kΩ pull-up resistor** between OUT and 3.3V for signal stabilization.

Conclusion:

This project successfully demonstrated motion detection using a PIR sensor, first by analyzing its behavior using an oscilloscope, and then integrating it with an ESP8266MOD for IoT applications. By following proper wiring, using stabilizing resistors, and troubleshooting common issues, the system reliably detects movement and can be used in security systems, automation, and smart home projects.