Experiment No._06

Title: Interfacing ESP32 with PIR motion sensor

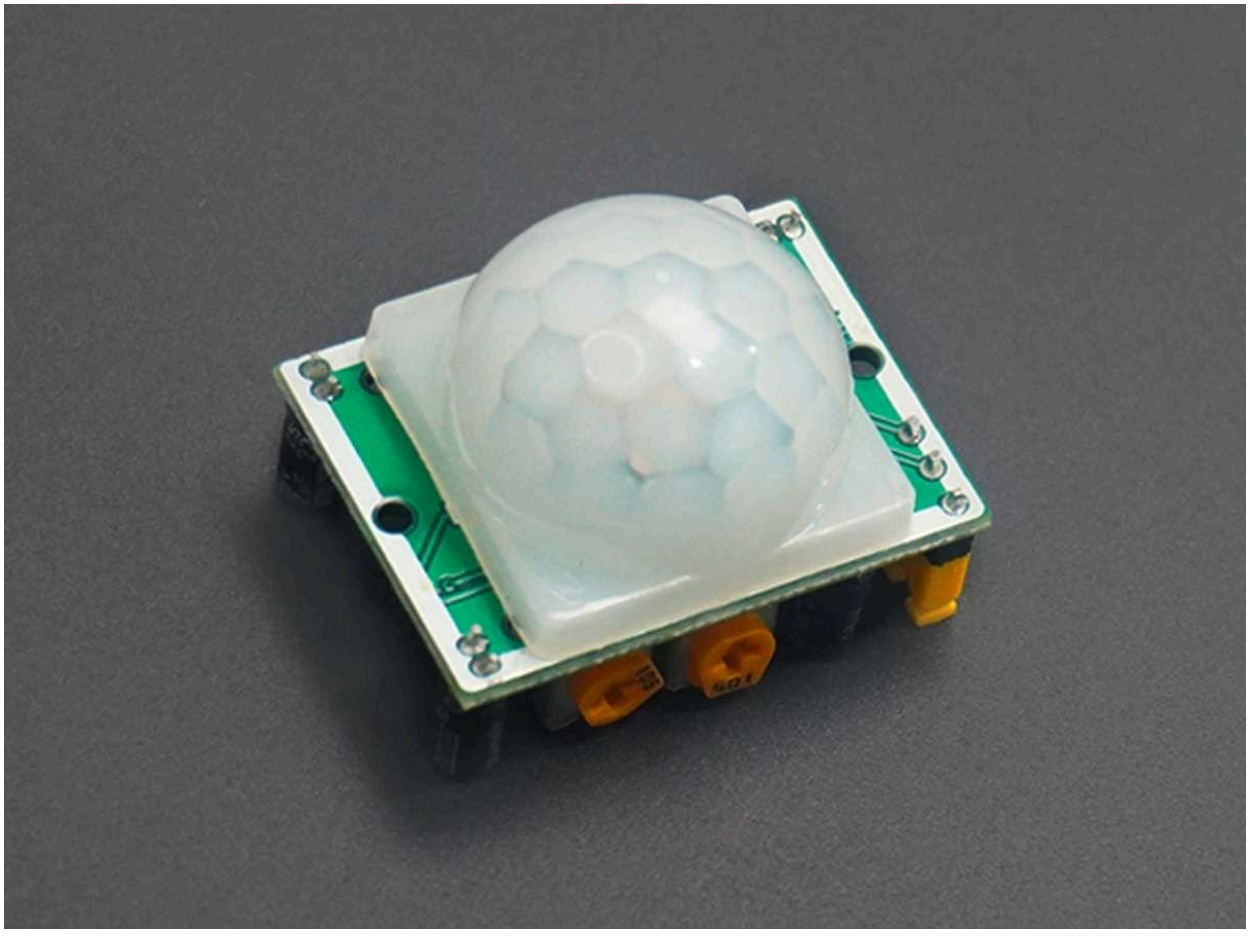
Batch: B–1 **Roll No.: 16010422234** **Experiment No.: 06**

Aim: Interfacing ESP32 with PIR motion sensor

Resources needed: ESP32, PIR sensor, and Jumper Wires.

Theory:

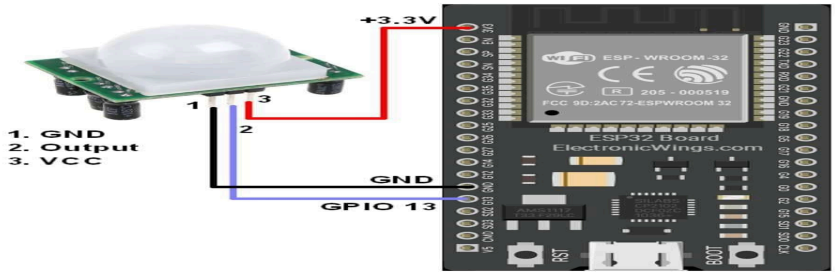
Overview of PIR Sensor



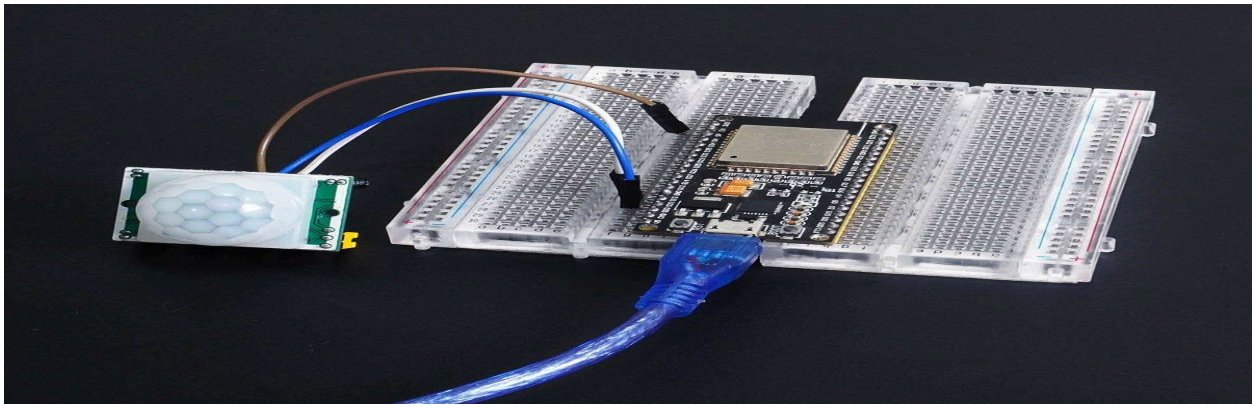
PIR Sensor

- PIR sensor is used for detecting infrared heat radiations. This makes them useful in applications involving detection of moving living objects that emit infrared heat radiations.
- The output (in terms of voltage) of PIR sensor is high when it senses motion; whereas it is low when there is no motion (stationary object or no object).
- For more information on PIR sensor and how to use it, refer the topic PIR Sensor in the sensors and modules section.

PIR Motion Sensor Hardware Connection with ESP32



ESP32 interfacing with PIR Motion Sensor



Note:

- PIR sensor: Never keep PIR Sensor close to the Wi-Fi antenna, ESP32, or NodeMCU.
- PIR (Passive Infrared) sensor close to a WiFi antenna impacts the sensor's performance.
- PIR sensors detect changes in infrared radiation for motion detection.
- WiFi signals emit electromagnetic radiation that can interfere with the PIR sensor. Which causes false detection.
- So always keep the PIR sensor and WiFi antenna as far apart as possible.
- Also, you can try to shield the PIR sensor from the WiFi signal. This can be done by using metal shields or Faraday cages around the PIR sensor.

Activity:

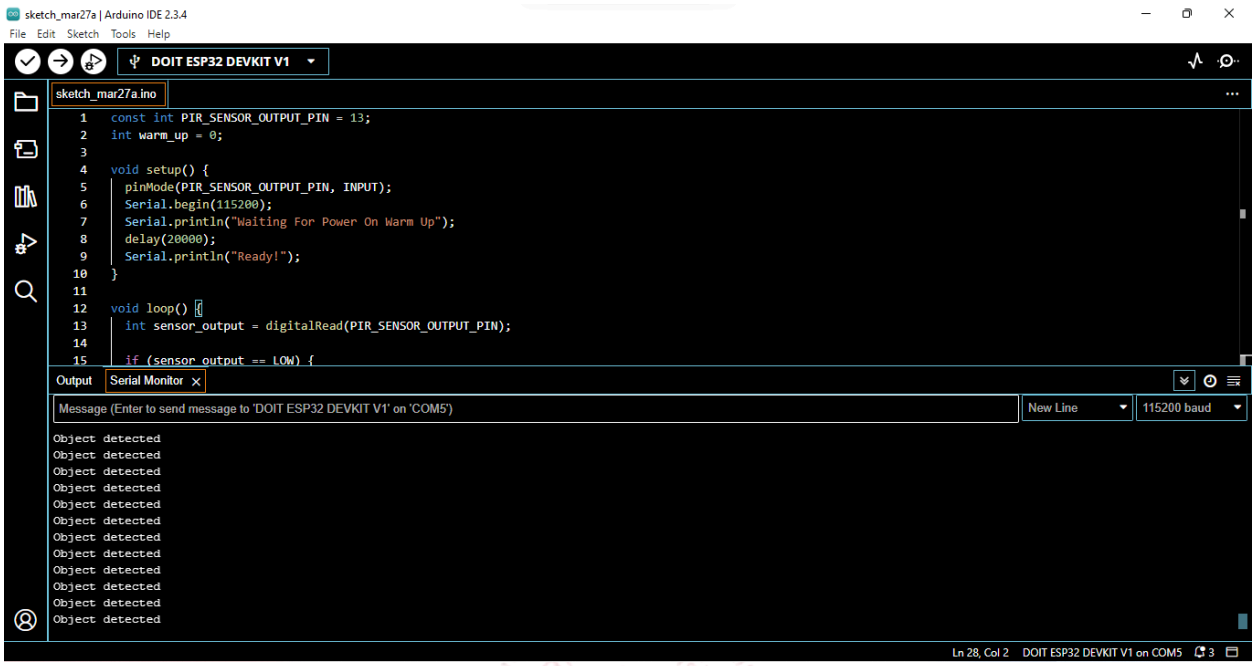
Detect PIR Motion using ESP32 and Arduino IDE

1. Motion detection of living objects using PIR sensor using Arduino IDE.

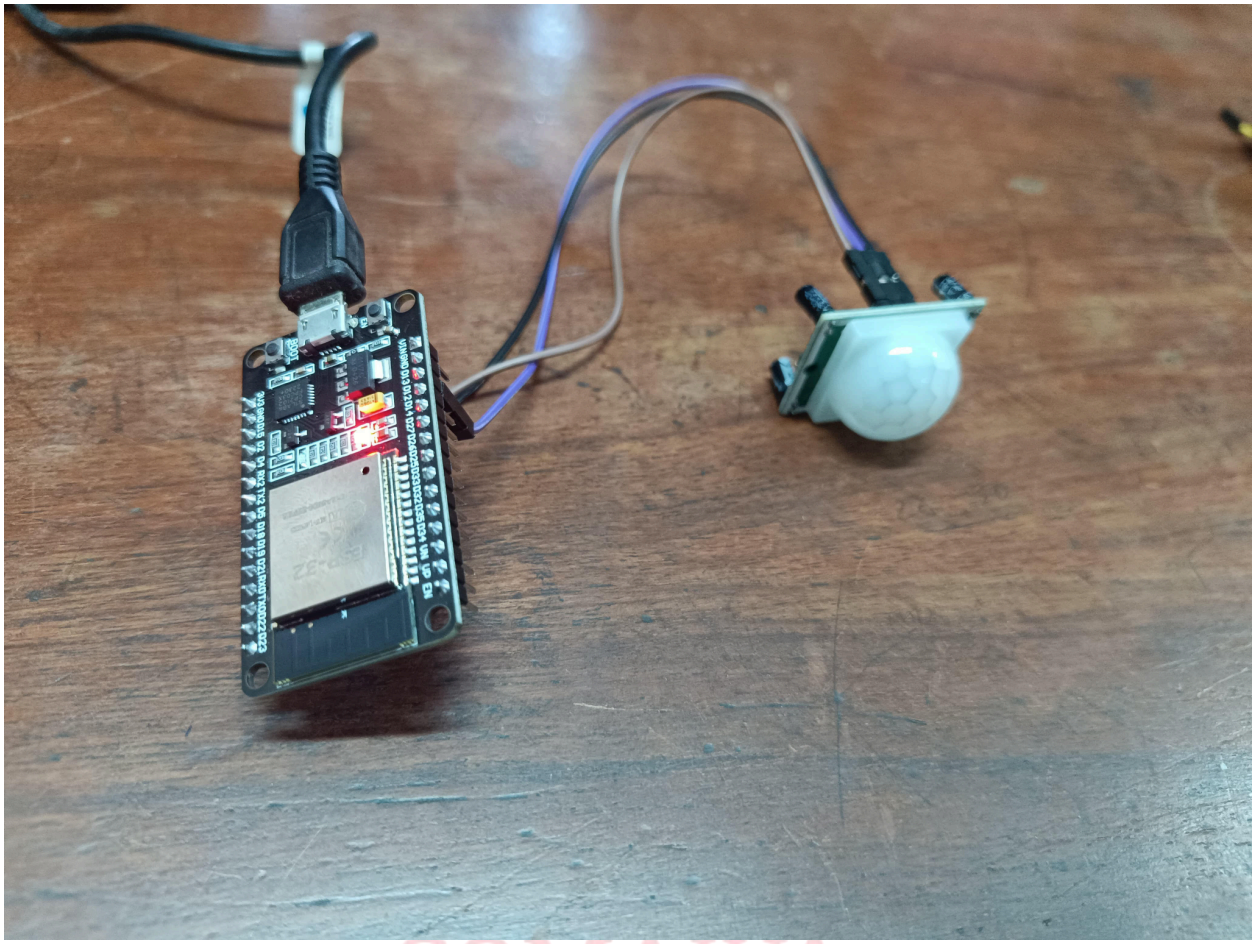
2. Upon detection of motion, "Object detected" is printed on the serial monitor of Arduino. When there is no motion, "No object in sight" is printed on the serial monitor of Arduino.

Results: (Program printout with output / Document printout as per the format)

```
const int PIR_SENSOR_OUTPUT_PIN = 13;
int warm_up = 0;

void setup() {
  pinMode(PIR_SENSOR_OUTPUT_PIN, INPUT);
  Serial.begin(115200);
  Serial.println("Waiting For Power On Warm Up");
  delay(20000);
  Serial.println("Ready!");
}

void loop() {
  int sensor_output = digitalRead(PIR_SENSOR_OUTPUT_PIN);

  if (sensor_output == LOW) {
    if (warm_up == 1) {
      Serial.println("Warming Up");
      warm_up = 0;
      delay(2000);
    }
    Serial.println("No object in sight");
  } else {
    Serial.println("Object detected");
    warm_up = 1;
  }

  delay(1000);
}
```


sketch_mar27a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
sketch_mar27a.ino
1 const int PIR_SENSOR_OUTPUT_PIN = 13;
2 int warm_up = 0;
3
4 void setup() {
5 pinMode(PIR_SENSOR_OUTPUT_PIN, INPUT);
6 Serial.begin(115200);
7 Serial.println("Waiting For Power On Warm Up");
8 delay(20000);
9 Serial.println("Ready!");
10 }
11
12 void loop() {
13 int sensor_output = digitalRead(PIR_SENSOR_OUTPUT_PIN);
14
15 if (sensor_output == LOW) {
Output Serial Monitor
Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM5')
New Line
115200 baud
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Ln 28, Col 2 DOIT ESP32 DEVKIT V1 on COM5


sketch_mar27a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
sketch_mar27a.ino
8 delay(20000); /* Power On Warm Up Delay */
Output Serial Monitor
Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on 'COM5')
New Line
115200 baud
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Object detected
Ln 28, Col 2 DOIT ESP32 DEVKIT V1 on COM5



Outcomes: CO2 — Comprehend IoT architecture, enabling technologies and protocols

Conclusion:

In this experiment, we successfully interfaced the ESP32 with a PIR motion sensor to detect motion using the Arduino IDE. The PIR sensor detected infrared radiation changes, allowing us to identify moving objects. The ESP32 processed the sensor's output and displayed appropriate messages on the serial monitor, indicating whether motion was detected or not. This experiment provided hands-on experience in IoT-based motion detection and reinforced the importance of sensor placement to minimize interference from WiFi signals.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of faculty in-charge with date

References:

1. https://en.wikipedia.org/wiki/Internet_of_things

2. Introduction - Blynk Documentation

Books:

1. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014.

2. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on-Approach)", 1stEdition, VPT, 2014.

3. Dr. Ovidiu Vermesan, Dr. Peter Friess, "Internet of Things - From Research and Innovation to Market Deployment", River Publisher, 2014
