Soham Thakur D10B/68 Arjun Prabhu D10B/54

IOT based application using IR sensors + actuator(LED)

Aim

To design and implement an IoT-based obstacle detection system using an infrared obstacle sensor and an ESP8266 microcontroller. The system will turn on an LED when an obstacle is detected and turn it off when no obstacle is present. Additionally, the obstacle detection status (0 or 1) will be displayed on the Serial Monitor.

Apparatus Required

1. ESP8266 (NodeMCU) microcontroller
2. Infrared Obstacle Sensor
3. LED
4. Jumper wires
5. Breadboard
6. Micro USB cable
7. Laptop with Arduino IDE installed

Theory

Infrared (IR) obstacle sensors are commonly used in automation and robotics for detecting objects. These sensors emit infrared light, which reflects off nearby objects and is detected by the sensor’s receiver. When an obstacle is present, the sensor outputs a low signal (0), and when no obstacle is detected, it outputs a high signal (1).

The ESP8266 microcontroller processes the IR sensor’s output and controls an LED. If an obstacle is detected, the LED turns on; otherwise, it remains off. The status of obstacle detection is also displayed on the Serial Monitor.

Working

1. The IR obstacle sensor continuously monitors for obstacles.
2. When an obstacle is detected, it sends a low signal (0) to the ESP8266.
3. The ESP8266 processes this input and turns on the LED.
4. The obstacle detection status is displayed on the Serial Monitor.
5. If no obstacle is detected, the sensor sends a high signal (1), turning off the LED and updating the Serial Monitor accordingly.

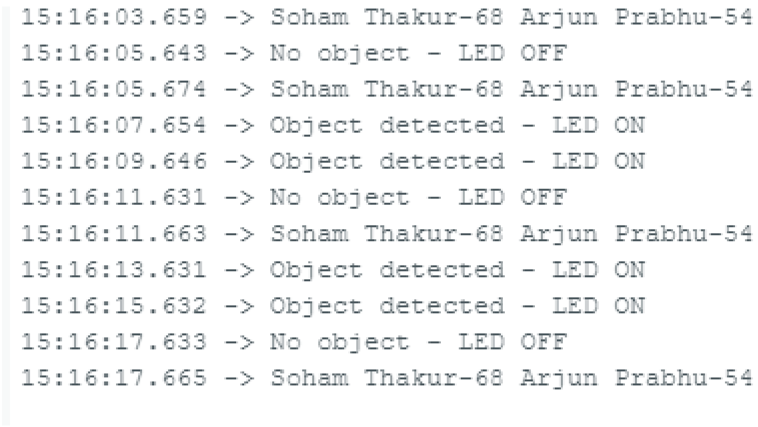
Procedure

1. **Hardware Connections:**
    - Connect the **VCC** of the IR obstacle sensor to the **3.3V** pin of the ESP8266.
    - Connect the **GND** of the IR obstacle sensor to the **GND** of the ESP8266.
    - Connect the **OUT** pin of the IR obstacle sensor to **D5** (GPIO14) of the ESP8266.
    - Connect the **anode** of the LED to **D6** (GPIO12) of the ESP8266.
    - Connect the **cathode** of the LED to **GND**.
2. **Software Setup:**
    - Install the **Arduino IDE** and add the ESP8266 board package.
    - Write the code to read sensor data, control the LED, and display the output on the Serial Monitor.
3. **Upload and Test:**
    - Connect the ESP8266 to a computer via a USB cable.
    - Upload the program to the ESP8266 using the Arduino IDE.
    - Open the Serial Monitor to check the obstacle detection status in real-time.

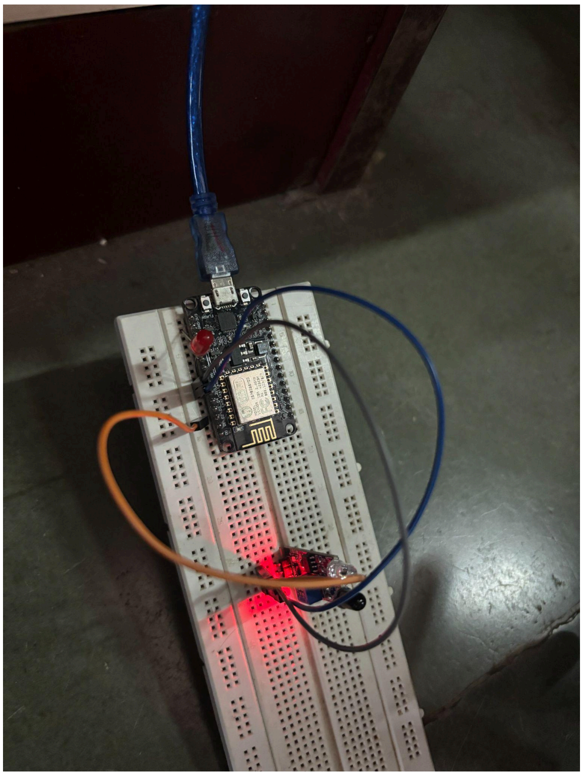
Code

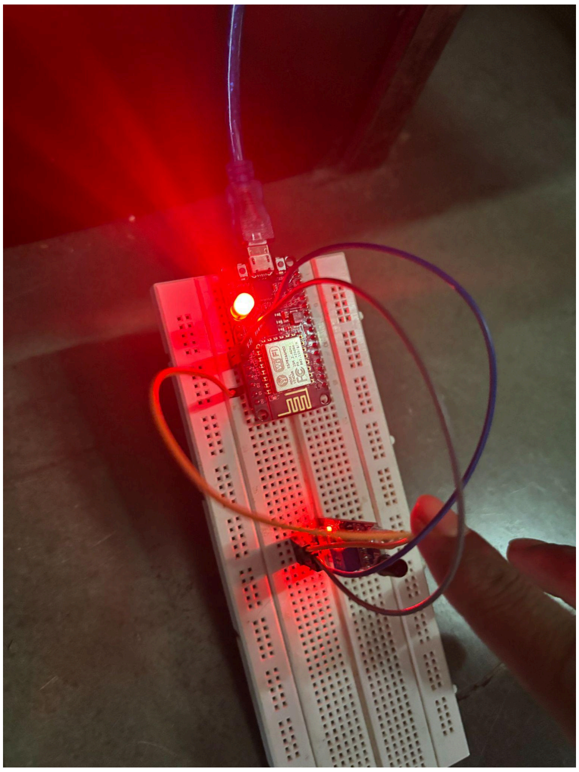
```
#define IR_SENSOR_PIN D2
#define LED_PIN D5
void setup() {
   pinMode(IR_SENSOR_PIN, INPUT);
   pinMode(LED_PIN, OUTPUT);
   digitalWrite(LED_PIN, LOW);
Serial.begin(9600);
}
void loop() {
   int sensorValue = digitalRead(IR_SENSOR_PIN);
   if (sensorValue == LOW) {
        digitalWrite(LED_PIN, LOW);
      Serial.println("Object detected - LED OFF”);
   } else {
      digitalWrite(LED_PIN, HIGH);
      Serial.println("No object - LED ON”);
      Serial.println("Soham Thakur-68 Arjun Prabhu-54”);
   }
   delay(2000);
}
```

Output



```
15:16:03.659 -> Soham Thakur-68 Arjun Prabhu-54
15:16:05.643 -> No object - LED OFF
15:16:05.674 -> Soham Thakur-68 Arjun Prabhu-54
15:16:07.654 -> Object detected - LED ON
15:16:09.646 -> Object detected - LED ON
15:16:11.631 -> No object - LED OFF
15:16:11.663 -> Soham Thakur-68 Arjun Prabhu-54
15:16:13.631 -> Object detected - LED ON
15:16:15.632 -> Object detected - LED ON
15:16:17.633 -> No object - LED OFF
15:16:17.665 -> Soham Thakur-68 Arjun Prabhu-54
```





Conclusion

In this experiment, we successfully implemented an IoT-based obstacle detection system using an infrared obstacle sensor and ESP8266. The LED responded accurately to obstacle detection, and the real-time status was displayed on the Serial Monitor. This project demonstrates the use of IoT in automation and obstacle detection applications.