Indian Institute of Information Technology, Surat

IOT Project Report on

SMART STREET LIGHT MONITORING AND CONTROLLING SYSTEM

Submitted by

Aniket Maurya (UI21EC10)

Ankit Kumar (UI21EC12)

Ashwani Kumar (UI21EC18)

Course Faculty

Dr. Sudeep Sharma

Dr. Hemant Gokhlani

Department of Electronics and Communication

Engineering Indian Institute of Information Technology Surat Gujarat-394190, India - 2024

PROJECT OVERVIEW

IOT BASED SMART STREET LIGHT MONITORING AND CONTROLLING SYSTEM

ABSTRACT:

Most of the places have automatic street light which can sense the daytime and nighttime, and automatically turns on and off according the night and day. Here we are extending this project by adding one more constraint to turn on the light that is Street light will only glow if there is darkness and someone is passing through the street. The main objective of this project is to reduce the power consumption by glowing the Street light only when it is needed. In this project we are demonstrating the prototype of the Smart Street Light with 3 IR sensors, 1 LDR sensor and 3 LEDs - each representing one street light. We will also update the LDR sensor data to the ThingSpeak and control the LEDs (Street lights) over the internet from anywhere in the world.

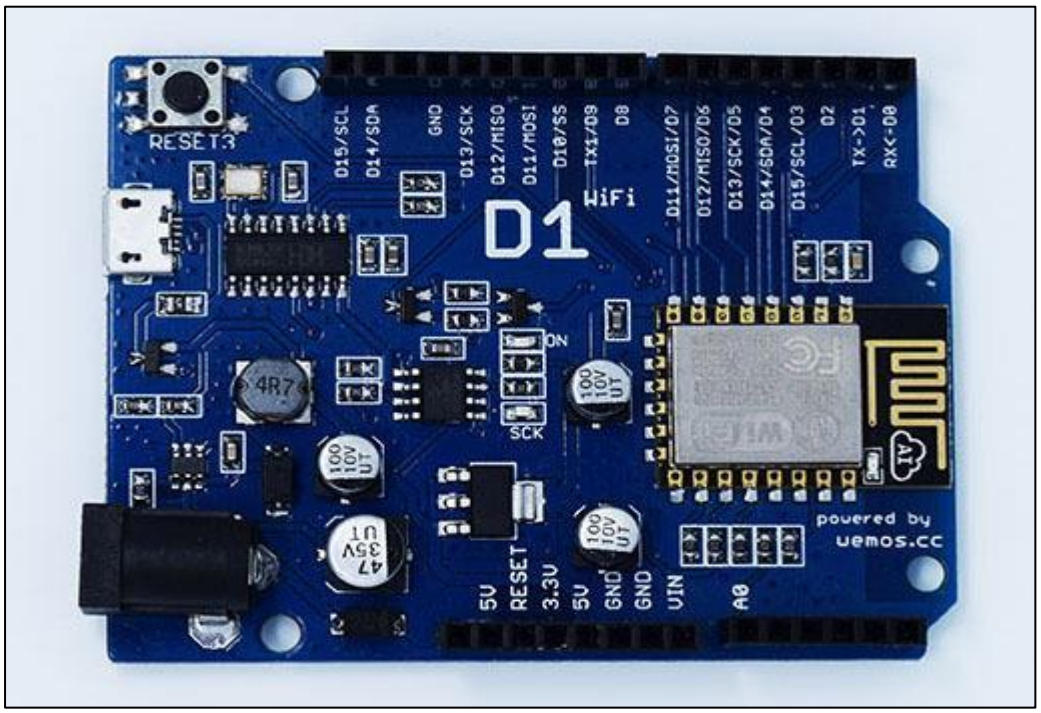
COMPONENTS REQUIRED

1. ESP8266 NodeMCU
2. Micro USB cable
3. LEDs
4. Jumper wires
5. IR sensors
6. LDR sensors
7. Jumper wires

Before moving forward, lets learn about the components used in the project.

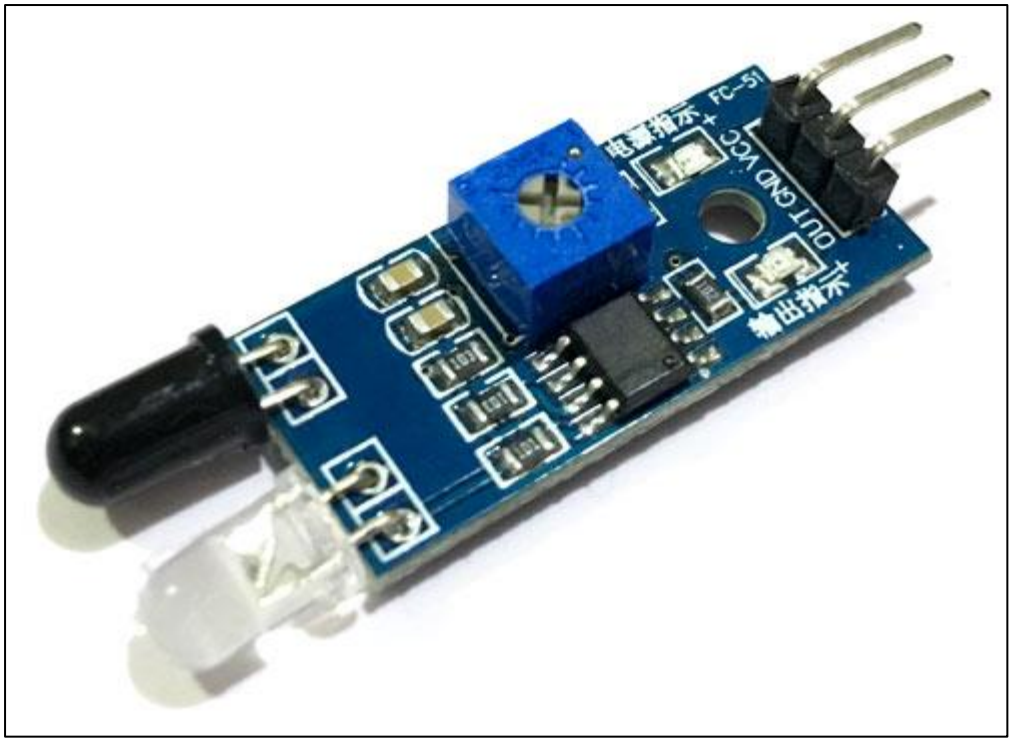
1. ESP8266 NODEMCU

It is an open source IoT platform. It includes firmware which runs on the low cost Wi-Fi enabled ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. It has GPIO, SPI, I2C, ADC, PWM AND UART pins for communication and controlling other peripherals attached to it. On board NodeMCU has CP2102 IC which provides USB to TTL functionality.



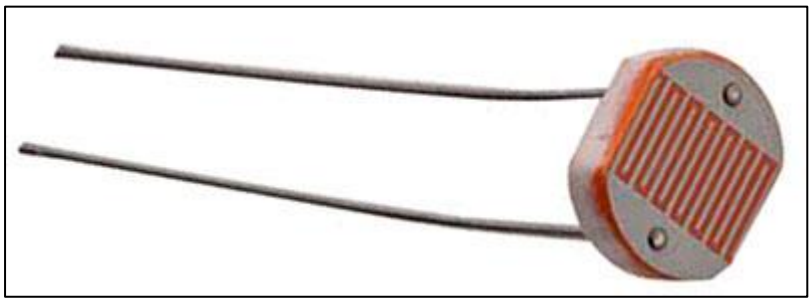
2. IR SENSOR

IR sensor is an electronic device which is used to detect objects by sensing infrared radiations reflected from the objects. It mainly consists of a transmitter **IR LED and a receiver photodiode**. It detects infrared radiations which have wavelength from 700nm to 1mm. When a specific positive voltage is applied across the transmitter LED it transmits the IR rays. If these rays fall on some object, then that object reflects back the IR rays which are received by the receiver photodiode. The receiver diode generates a voltage across its terminals which depend on the intensity of light reflected by the object. Generally, IR receiver LED is darker (black) whereas transmitter is transparent in colour.



3. LDR SENSOR

LDR stands for **Light Dependent Resistor** also known as photo-resistor. LDR is sensitive to light and its resistance changes according to the intensity of light falling on it. It is made up of high resistance semiconductor and its resistance increases in darkness and decreases in light. When light incident on the LDR exceeds some threshold, it absorbs the photons and allows electrons to jump into the conduction band. LDR generates a variable resistance which depends on the intensity of light falling on it. It is mainly used in electric circuits like street light, alarm clock, automatic brightness and contrast control etc.

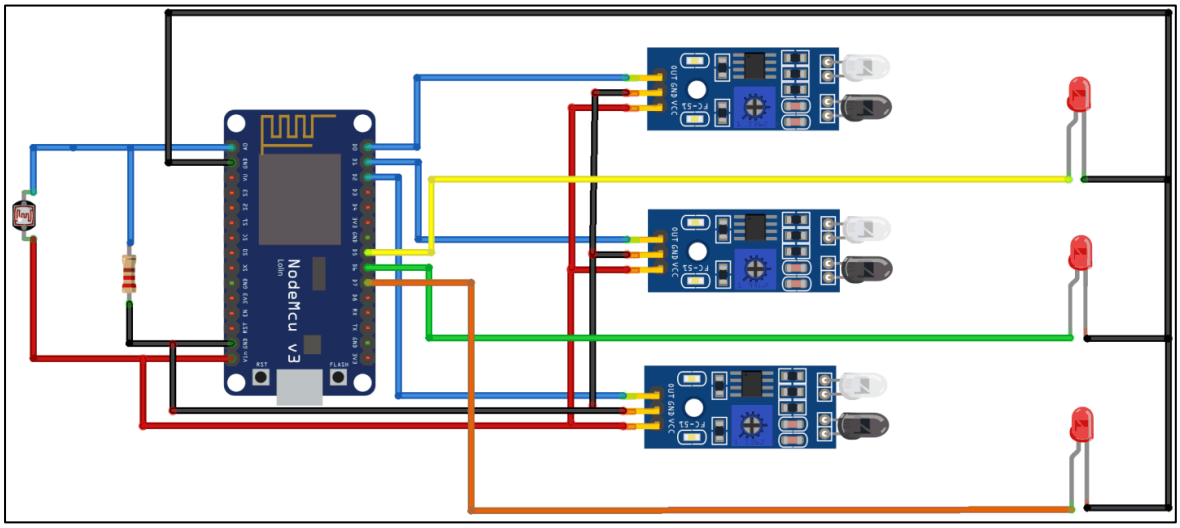


4. THINGSPEAK

ThingSpeak is a cloud based data platform which is used to send and receive the data in real time using HTTP protocol. It is used in Iot applicaiotn to store and monitor the data from anywhere in the world over internet.

BLOCK DIAGRAM/ CIRCUIT DIAGRAM

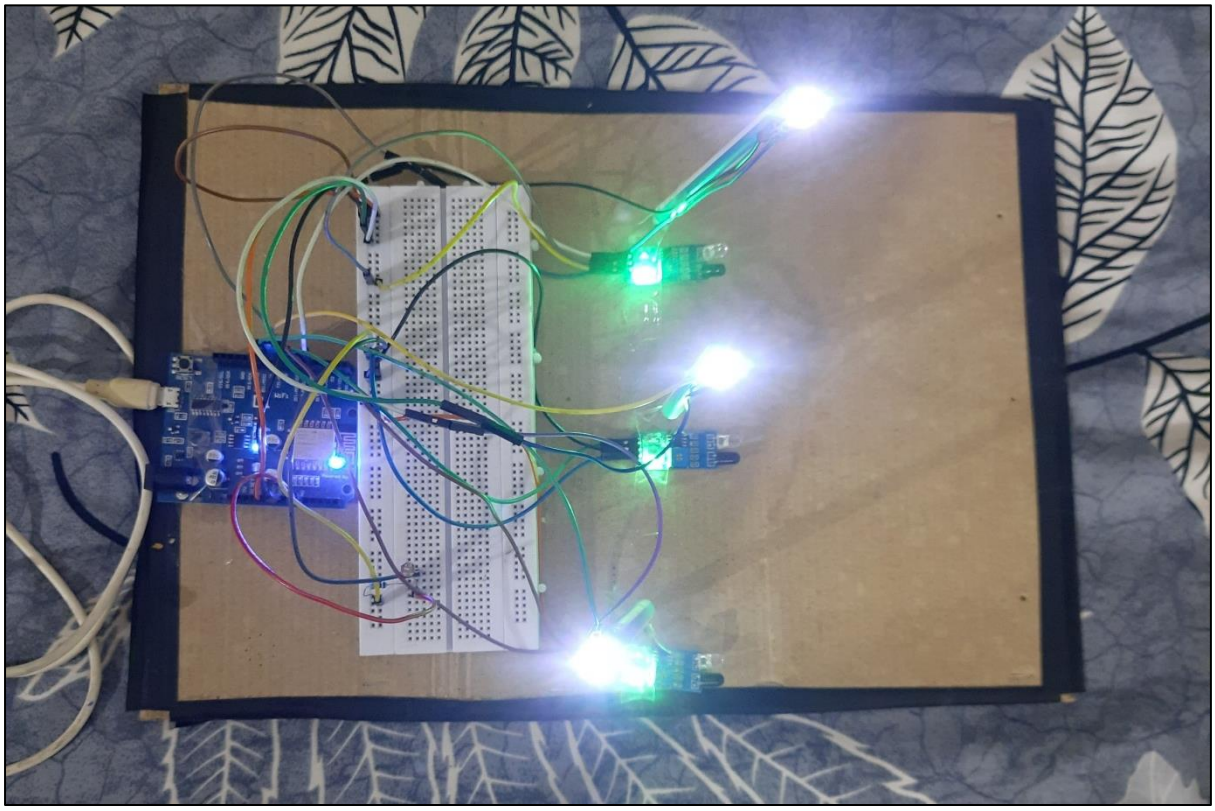
Circuit diagram for this IoT based Smart Street Light is as follows.

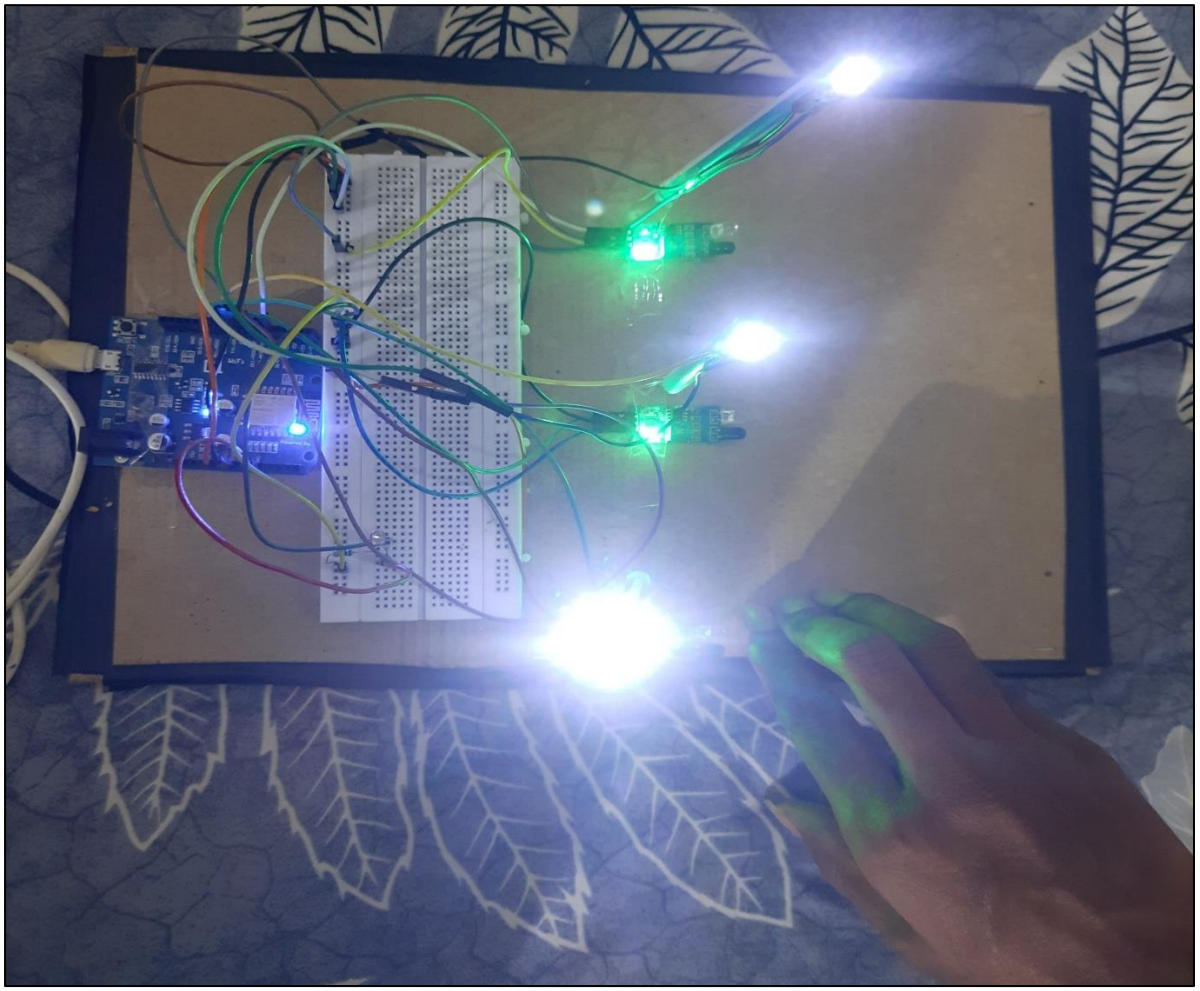


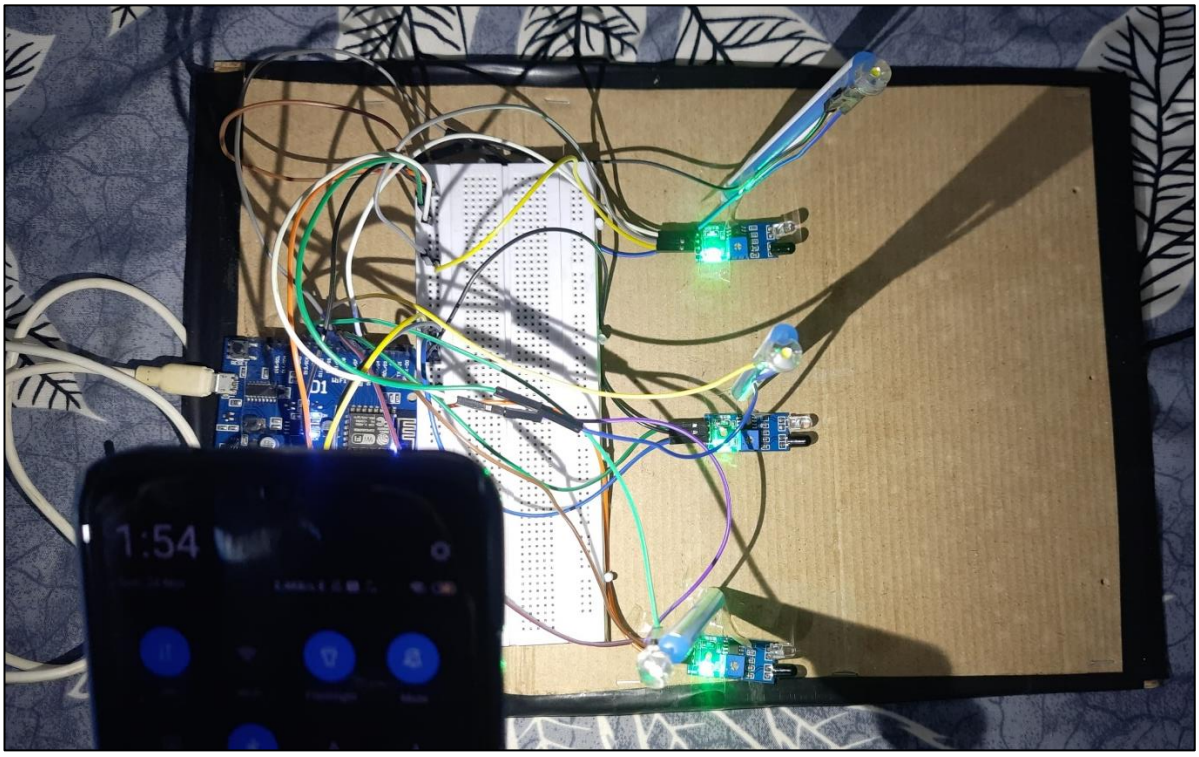
This circuit mainly consists ESP8266, LDR sensor, IR sensors and LEDs.

Here the **LDR sensor is used to detect whether it is daytime or night time**. Since LDR sensor generates variable resistance based on the amount of light falling on it, it has to be connected like a potentiometer. One end of the LDR sensor is connected to 5V and other end is connected to fixed resistance which is further connected to ground. NodeMCU has one **ADC pin** (A0) which is connected to point between fixed resistance and one end of the LDR sensor as shown in the circuit diagram. Since the LDR sensor gives variable resistance therefore variable voltage will be generated at A0 according to the amount of light falling on LDR.

IR sensors are used to detect if someone is crossing the street or not. It detects the obstacle or motion in the surrounding. The transmitter will transmit IR rays which will be reflected back if it falls on some object like person, animal, vehicles, etc. The reflected ray will be received by receiver diode and hence will confirm the presence of object and the corresponding LED will be glowed. This method will save significant amount of electricity as the street light will only turns on if there is someone present in the Street. IR sensor has 3 pins, two of which are VCC and ground and one is output pin. The output of IR sensor gets high if detects presence of some object. This pin is connected to GPIO pin of NodeMCU so whenever the IR sensor detects someone passing through the street it triggers the Street light. In our case one LED will be turned on.






1:54

CODE

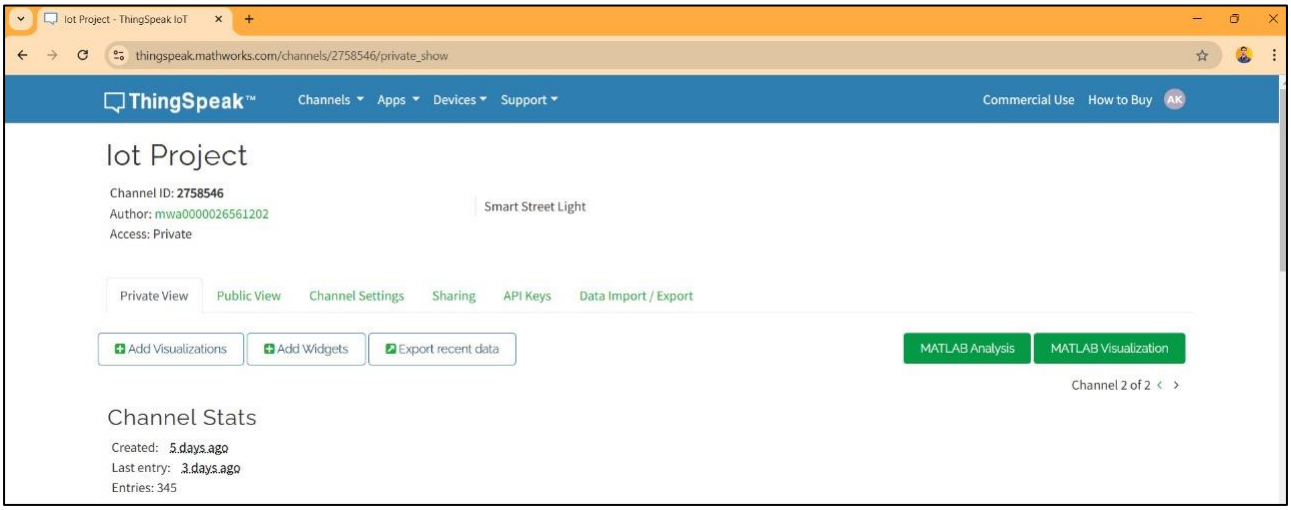
```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ThingSpeak.h>
#include <ESP8266WebServer.h>

// WiFi credentials
const char* ssid = "Bingo";
const char* password = "ankit1212";

// ThingSpeak settings
unsigned long channelID = 2758546;
const char* apiKey = "08T8NZBHR6FL8MIW";

// Pin definitions
const int irSensorPins[] = {D5, D6, D7}; // IR sensor pins (e.g., D5, D6, D7)
const int ledPins[] = {D3, D4, D8};      // LED pins (e.g., D3, D4, D8)
const int ldrPin = A0;                   // LDR pin (Analog input)

// WiFi and ThingSpeak client
WiFiClient client;
ESP8266WebServer server(80); // HTTP server on port 80

String serialData = ""; // String to store serial data

void handleRoot() {
  String webpage = "<html>\
  <head><title>ESP8266 Data Logger</title></head>\
  <body>\
  <h1>ESP8266 Data Logger</h1>\
  <pre>" + serialData + "</pre>\
  <script>\
  setInterval(() => { location.reload(); }, 5000);\
  </script>\
  </body>\
  </html>";
  server.send(200, "text/html", webpage);
}

void setup() {
  for (int i = 0; i < 3; i++) {
    pinMode(irSensorPins[i], INPUT); // Set IR sensor pins as input
    pinMode(ledPins[i], OUTPUT);     // Set LED pins as output
  }

  Serial.begin(115200);              // Initialize Serial Monitor
  Serial.println("IR and LDR Motion Detection Initialized");

  WiFi.begin(ssid, password);        // Connect to WiFi
```

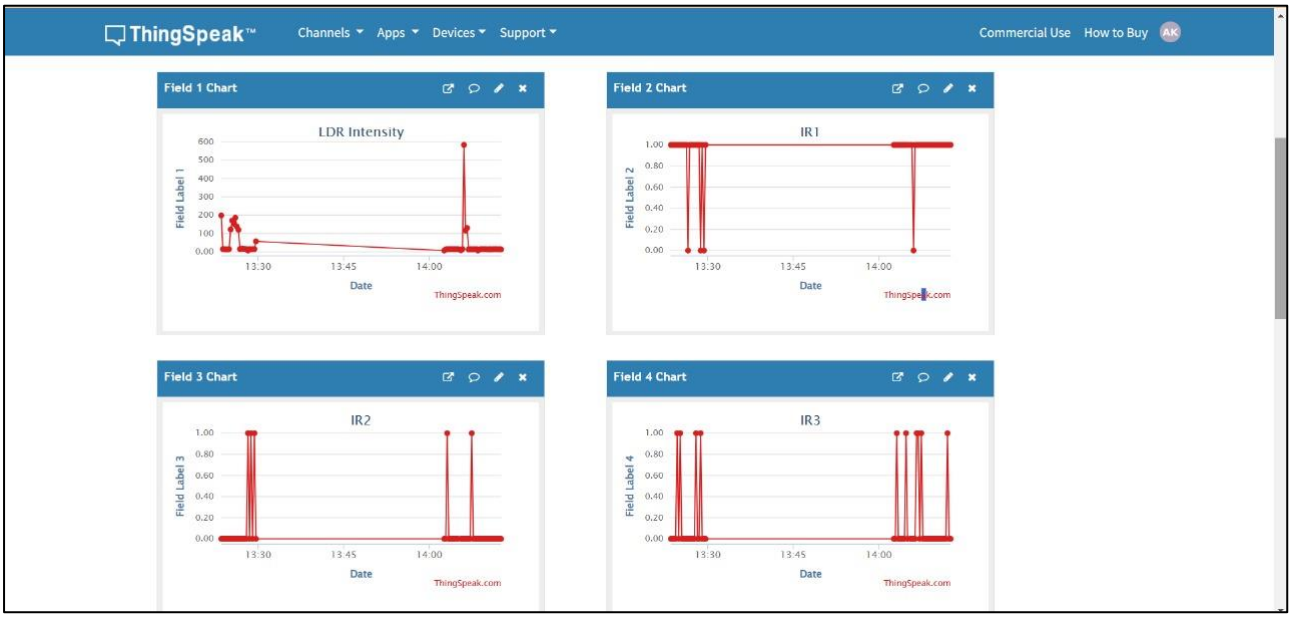
```
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("\nWiFi connected");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());

  ThingSpeak.begin(client);          // Initialize ThingSpeak

  server.on("/", handleRoot);        // Serve the root page
  server.begin();                    // Start the web server
}

void loop() {
  int ldrValue = analogRead(ldrPin); // Read LDR value (0-1023)
  int irStatus[3] = {0, 0, 0};       // Tracks motion detection for each IR sensor
  int ledStatus[3] = {0, 0, 0};      // Tracks LED intensity for each LED

  // Debugging the LDR value
  Serial.print("LDR Value: ");
  Serial.println(ldrValue);
  serialData += "LDR Value: " + String(ldrValue) + "\n";

  if (ldrValue < 300) { // Low light detected
    for (int i = 0; i < 3; i++) {
      analogWrite(ledPins[i], 50);  // Dim the LED at low intensity
      int irValue = digitalRead(irSensorPins[i]); // Read each IR sensor
      if (irValue == LOW) { // Motion detected by this IR sensor
        analogWrite(ledPins[i], 255); // LED at high intensity
        Serial.print("Motion detected on IR Sensor ");
        Serial.print(i + 1);
        Serial.println(": LED at high intensity (200)");
        serialData += "Motion detected on IR Sensor " + String(i + 1) + ": LED at high intensity
(200)\n";
        irStatus[i] = 1;    // Motion detected
        ledStatus[i] = 255; // LED intensity
      } else { // No motion for this IR sensor
        analogWrite(ledPins[i], 50); // LED at low intensity
        Serial.print("Low light, no motion on IR Sensor ");
        Serial.print(i + 1);
        Serial.println(": LED at low intensity (50)");
        serialData += "Low light, no motion on IR Sensor " + String(i + 1) + ": LED at low
intensity (50)\n";
        irStatus[i] = 0;     // No motion detected
        ledStatus[i] = 50;   // LED intensity
      }
    }
  } else { // Sufficient light
```

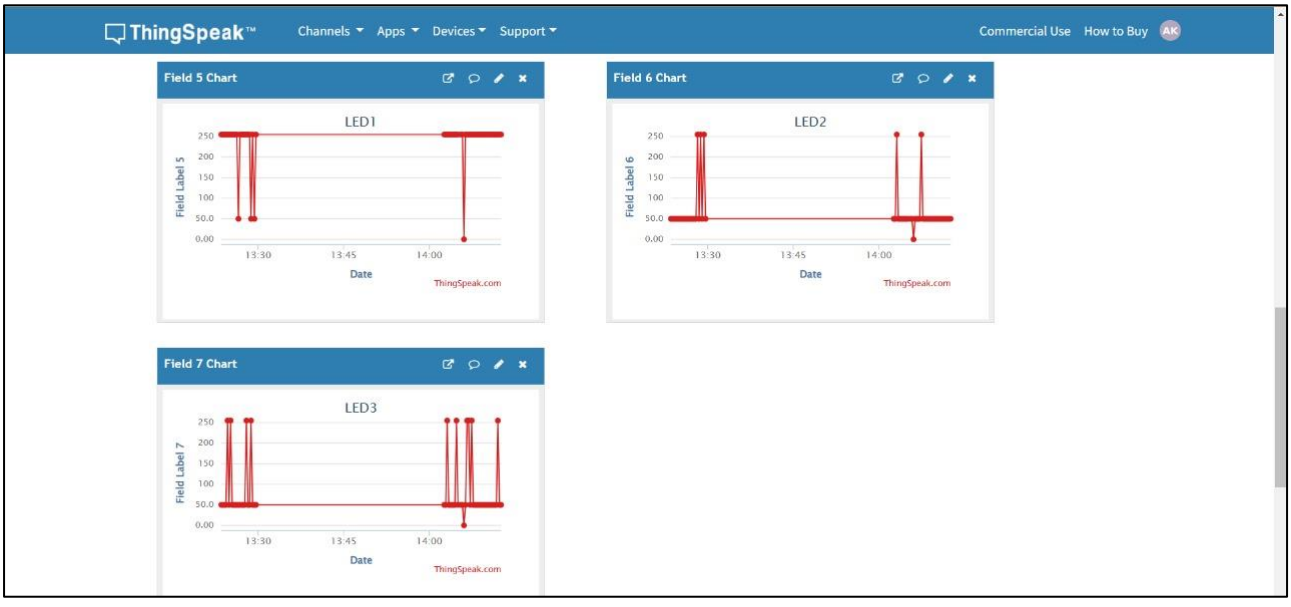
```
    for (int i = 0; i < 3; i++) {
      digitalWrite(ledPins[i], LOW); // Turn off all LEDs
      irStatus[i] = 0;          // No motion
      ledStatus[i] = 0;         // LED off
    }
    Serial.println("Sufficient light: All LEDs OFF");
    serialData += "Sufficient light: All LEDs OFF\n";
  }

  // Send data to ThingSpeak
  if (WiFi.status() == WL_CONNECTED) {
    ThingSpeak.setField(1, ldrValue);        // Field 1: LDR Value
    ThingSpeak.setField(2, irStatus[0]);    // Field 2: IR Sensor 1 Status
    ThingSpeak.setField(3, irStatus[1]);    // Field 3: IR Sensor 2 Status
    ThingSpeak.setField(4, irStatus[2]);    // Field 4: IR Sensor 3 Status
    ThingSpeak.setField(5, ledStatus[0]);   // Field 5: LED 1 Status
    ThingSpeak.setField(6, ledStatus[1]);   // Field 6: LED 2 Status
    ThingSpeak.setField(7, ledStatus[2]);   // Field 7: LED 3 Status

    int responseCode = ThingSpeak.writeFields(channelID, apiKey);

    if (responseCode == 200) {
      Serial.println("Data sent to ThingSpeak successfully");
    } else {
      Serial.print("Error sending data: ");
      Serial.println(responseCode);
    }
  } else {
    Serial.println("WiFi disconnected, retrying...");
    WiFi.begin(ssid, password);
  }

  if (serialData.length() > 1000) { // Prevent excessive memory usage
    serialData = serialData.substring(serialData.length() - 1000);
  }

  server.handleClient(); // Handle HTTP requests
  delay(15000); // Wait 15 seconds between updates (ThingSpeak limit)
}
```

RESULT:

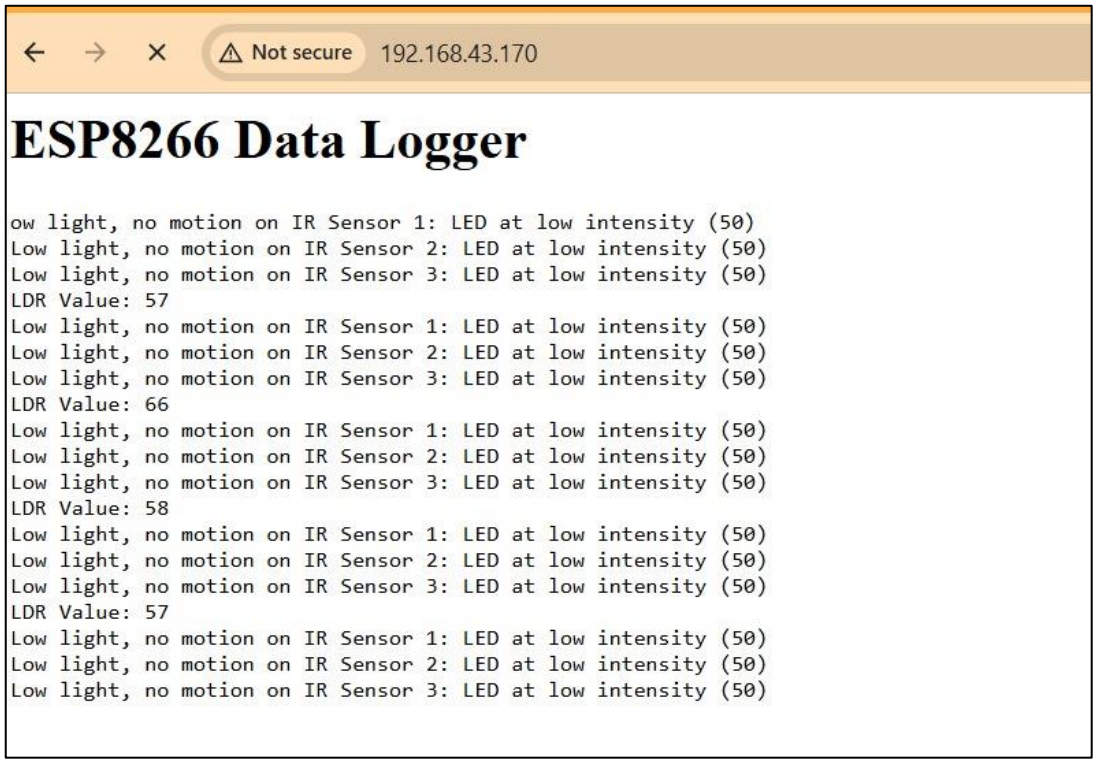
THINGSPEAK OUTPUT:

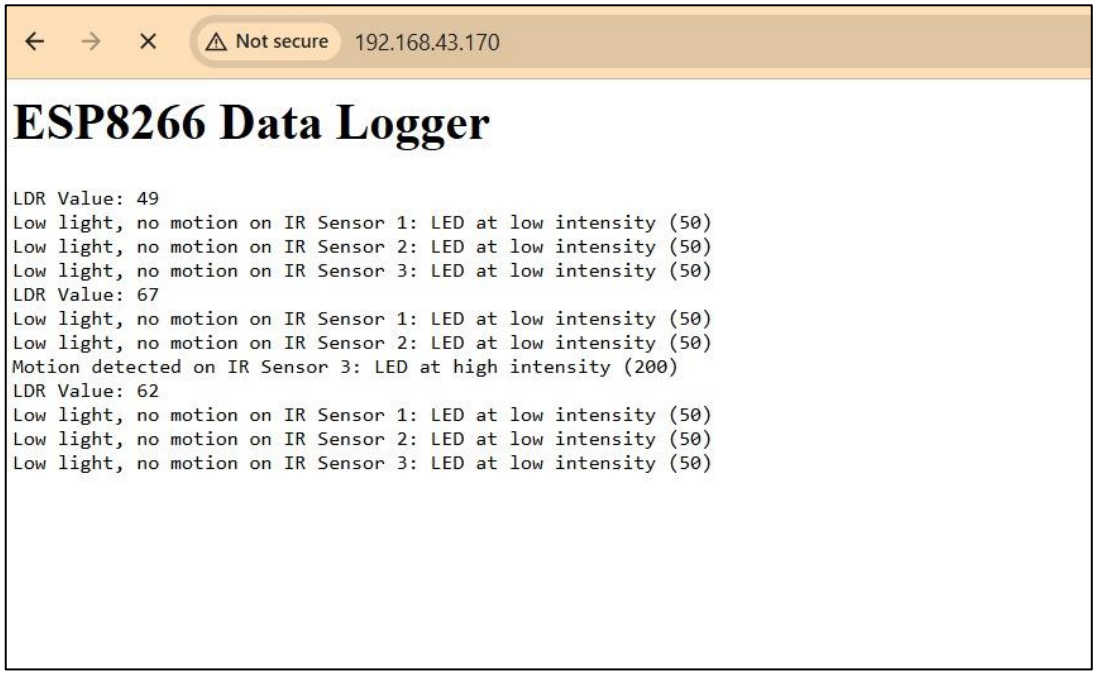






WEB PAGE OUTPUT:





CONCLUSION

The project successfully implements a Smart Street Light System, offering an energy-efficient, automated lighting solution. The system adapts to environmental and situational changes, ensuring optimized lighting for streets while saving electricity. Remote monitoring through ThingSpeak and a web interface makes it scalable for urban and rural deployments.

Advantages:

- Energy Efficiency: Reduces unnecessary energy consumption.
- Cost-Effective: Minimal hardware with IoT integration.
- Improved Safety: Ensures adequate lighting in response to motion.

Future Enhancements:

1. Integration with solar panels for sustainable power.
2. Use of advanced motion sensors for better accuracy.
3. Centralized control system for managing multiple units across a city.