GOVERNMENT OF TAMILNADU

DIRECTORATE OF TECHNICAL EDUCATION, CHENNAI

NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED

STUDENTS DEVELOPMENT PROGRAMME

ON

IoT AND ITS APPLICATIONS

HOST INSTITUTION

xxxxx

COIMBATORE – 04

TRAINING PARTNER

ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD

DATE:

NAME	**ROLL NO**
NAME 1	1
NAME 2	2
NAME 3	3
NAME 4	4
NAME 5	5

TABLE OF CONTENTS

IOT BASED STREET LIGHT CONTROL

INTRODUCTION:

This street lighting is one of the largest energy expanses of a city. A street lighting system can cut municipal street lighting cost is 50% to 70%. The smart street lighting system is a system that adjusts light output based on the usage and occupancy, i.e., automatic classification of pedestrian versus cyclist, versus automotive. The project is mainly implemented to track the intensity of the light using sensors and it is done using the wireless system to control the energy consumption and uses reduction measures through power conditioning and control. The street light (ON/OFF Status) will be accessed from anytime, anywhere through internet based on the real time system. The street controller should be installed on the pole light which consists of ESP32. The data from the street light controller can be transfer to base station by using wireless technology to monitor the system. The operation of the system can be conducted using auto mode and manual mode the control system will switch on-off the lights are required timings and can also vary the intensity of the street light according to requirement.

ABSTRACT:

- The Smart Street Light Control system using ESP32 is an innovative project designed to enhance energy efficiency and public safety. The system utilizes an ESP32 microcontroller paired with a BH1750 light sensor to monitor ambient light levels. When the surrounding area becomes dark, the system automatically turns on the street lights; when it's bright, the lights are turned off.
- The ESP32 also connects to the ThingzMate cloud via Wi-Fi, enabling remote monitoring and control. Additionally, an LED indicator is included to provide a visual status of the lights. This automated system reduces energy consumption by ensuring street lights are only on when needed, contributing to smart city infrastructure. The project demonstrates the integration of IoT technology in urban lighting systems, paving the way for more sustainable and intelligent public services.

HARDWARE & SOFTWARE REQUIREMENTS:

SOFTWARE:

- Wokwi.
- Arduino Ide.
- Thingzmate.

HARDWARE:

- LDR sensor
- ESP32
- Relay
- Bulb
- Power supply

WOKWI:

Wokwi is an online platform that simulates electronic circuits and microcontrollers, such as Arduino and ESP32. Users design circuits by placing virtual components on a digital breadboard and writing code directly in the browser. Wokwi compiles and runs the code in real-time, allowing users to see how the circuit behaves without needing physical hardware. This enables easy testing, debugging, and iteration, making it a valuable tool for learning and prototyping electronics projects.

ARDUINO IDE:

The Arduino IDE (Integrated Development Environment) allows users to write, compile, and upload code to Arduino microcontrollers. It provides a user-friendly interface for coding in a simplified C/C++ language. Once code is written, the IDE compiles it into machine language and uploads it via USB to the connected Arduino board. The board then executes the code, controlling connected components like sensors and LEDs. This process enables easy prototyping and development of embedded systems and electronics projects.

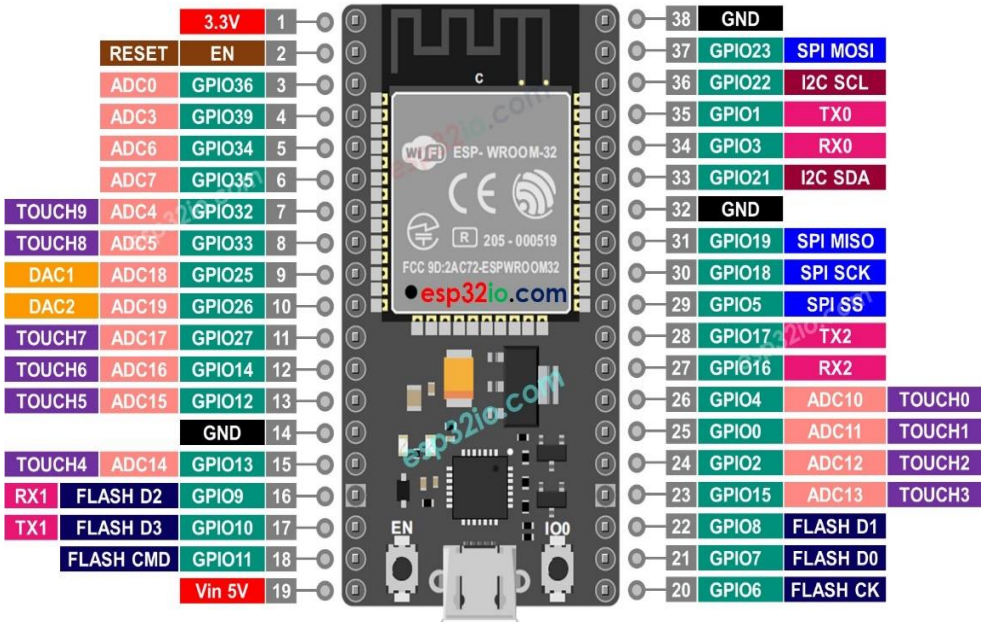
THINGZMATE:

ThingzMate is an IoT platform that enables the connection, monitoring, and management of IoT devices. It supports protocols like MQTT and HTTP, allowing devices to send data to the platform, where users can visualize it in real-time through custom dashboards. ThingzMate also offers remote device management, including firmware updates and configuration. Its cloud-based infrastructure ensures scalability, making it suitable for large-scale IoT deployments, helping developers quickly build and deploy smart, connected solutions.

ESP32 MICROCONTROLLER:

- The IoT-based street light control system using ESP32 automates street lighting by integrating sensors and cloud connectivity. An LDR (Light Dependent Resistor) detects ambient light levels, and a PIR (Passive Infrared) sensor detects motion. The ESP32 processes these inputs to control the street lights via a relay module.

- When ambient light is low or motion is detected, the ESP32 turns on the lights; otherwise, they remain off. Additionally, the ESP32 connects to a cloud platform through WiFi, enabling remote monitoring and control. This setup ensures energy-efficient operation and real-time management of street lighting.

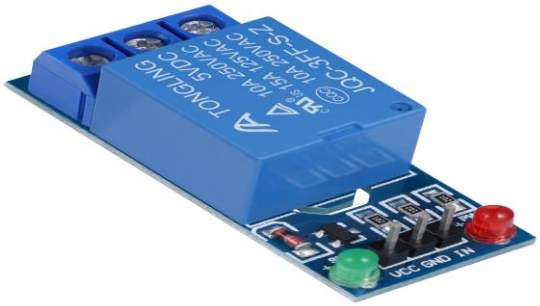


BH1750 SENSOR:

- In an IoT-based street light control system, the BH1750 sensor is used to measure ambient light intensity. This digital light sensor outputs lux values, which are read by the ESP32 microcontroller to determine whether the street lights should be on or off.
- When the light levels fall below a certain threshold, indicating low ambient light (e.g., during dusk or cloudy conditions), the ESP32 activates the street lights through a relay module. Conversely, when light levels rise above the threshold, the lights are turned off. This automatic control ensures energy-efficient operation of street lights based on real-time environmental conditions.

RELAY MODULE:

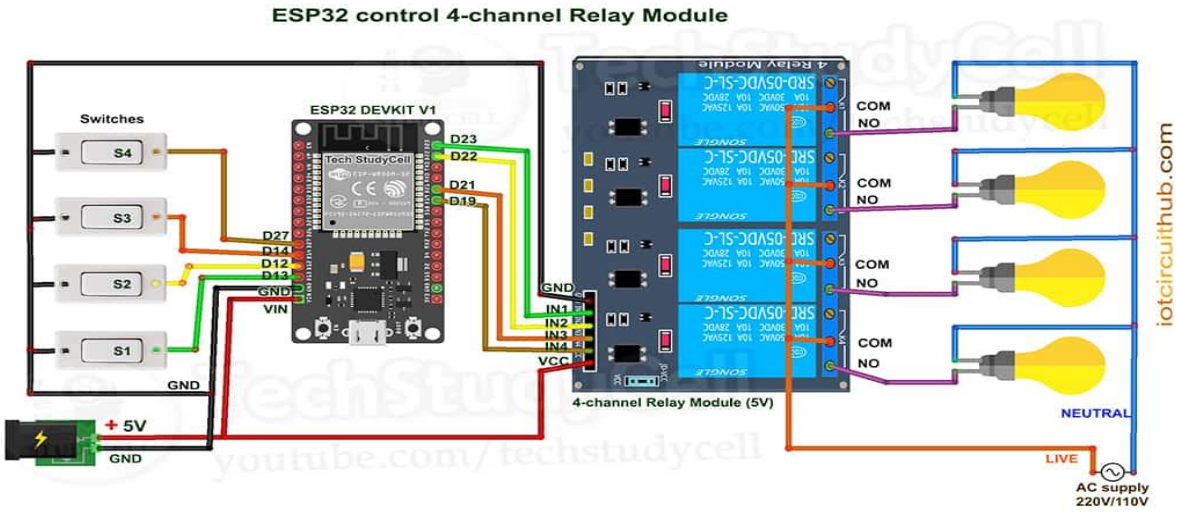
- In an IoT-based street light control system, a relay acts as an electrically operated switch that controls the power supply to the street lights. The relay is interfaced with the ESP32 microcontroller, which processes data from sensors like the LDR (Light Dependent Resistor) and PIR (Passive Infrared) sensor. When the ESP32 detects low ambient light or motion, it triggers the relay, closing the circuit and turning on the street lights. Conversely, when sufficient light is detected or no motion is present, the ESP32 deactivates the relay, opening the circuit and turning off the lights.
- The relay allows the ESP32, which operates at low voltage, to control high-voltage street lights safely. This setup ensures that the street lights are only activated when necessary, optimizing energy consumption. The relay's integration with the IoT system also enables remote control and automation of the street lighting, contributing to smarter and more efficient urban infrastructure.



POWER SUPPLY:

- In an IoT-based street light control system, the power supply provides the necessary electrical energy to operate the ESP32 microcontroller, sensors, and the relay module that controls the street lights. Typically, a 5V regulated power supply is used, which can be provided through a USB adapter or a dedicated power source.
- The power supply ensures that the ESP32 remains connected to WiFi and continuously monitors sensor inputs, such as ambient light and motion. When the sensors trigger a change, the ESP32 activates the relay, which switches the street lights on or off. A stable power supply is essential for the reliable operation of the entire system.

CIRCUIT DIAGRAM:



CODE:

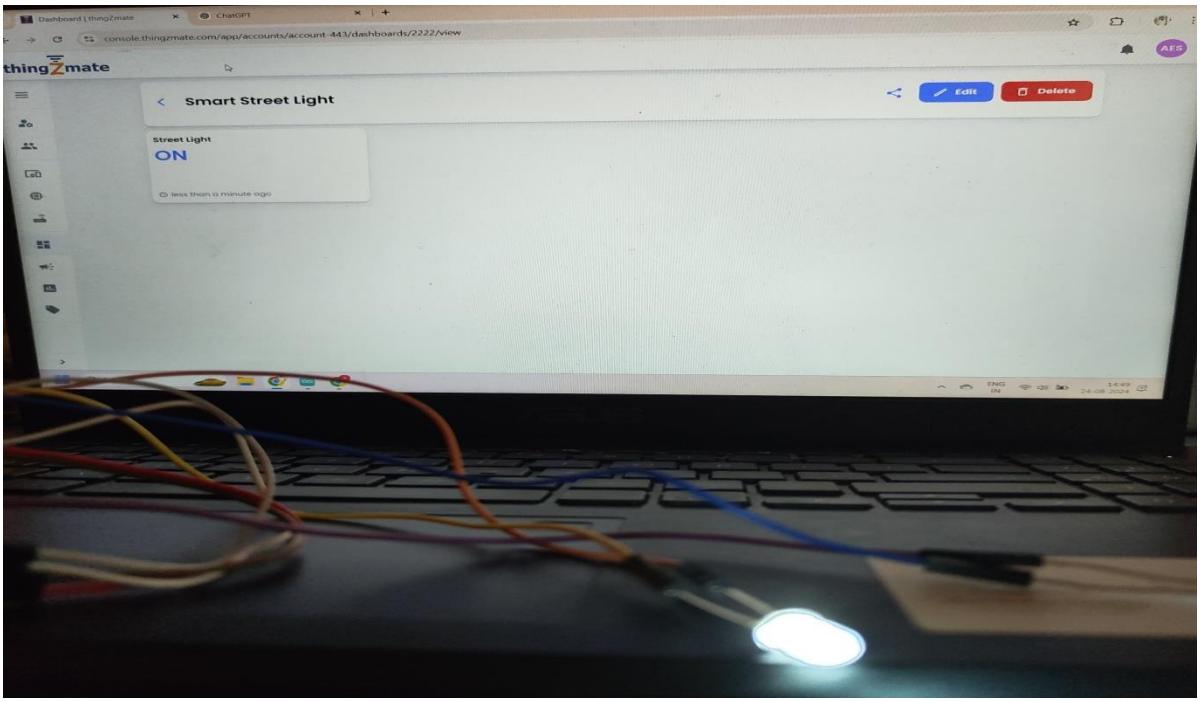
```
#include <Wire.h>

#include <BH1750.h>

#include <WiFi.h>

#include <HTTPClient.h>

const char* ssid = "tyrant";

const char* password = "speed123";

const char* serverName = "https://console.thingzmate.com/api/v1/device-
types/esp3211/devices/esp3298/uplink";

const char* apiKey = "f6a4c3ccdcd804fc4b532d36e00d4212";

#define LIGHT_THRESHOLD 100

BH1750 lightMeter;

int ledPin1 = 15; // Previously relayPin

int ledPin2 = 2;  // Additional LED pin

void setup() {

  Serial.begin(115200);

  Wire.begin();

  lightMeter.begin();

  pinMode(ledPin1, OUTPUT);

  pinMode(ledPin2, OUTPUT);

  digitalWrite(ledPin1, LOW);

  digitalWrite(ledPin2, LOW);
```

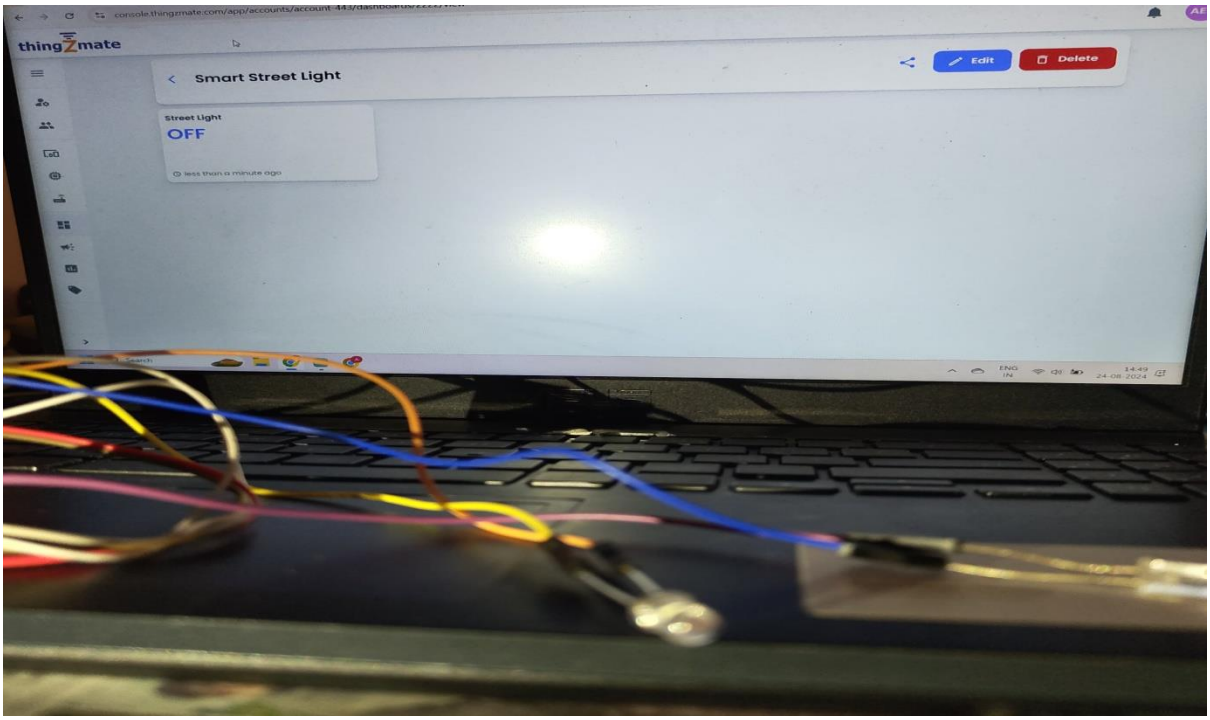
```
  // Connect to Wi-Fi
  connectToWiFi();
}
void loop() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("Wi-Fi not connected! Reconnecting...");
    connectToWiFi();
  }

  float lux = lightMeter.readLightLevel();
  Serial.print("Light: ");
  Serial.print(lux);
  Serial.println(" lx");

  if (lux < LIGHT_THRESHOLD) {
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, HIGH);
    sendDataToThingzMate("ON");
  } else {
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
   sendDataToThingzMate("OFF");
  }
```

```
 delay(2000);
}
void connectToWiFi() {
  WiFi.begin(ssid, password);
  Serial.print("Connecting to Wi-Fi");
  int max_attempts = 20; // Adjust the number of attempts if needed
  int attempts = 0;
    while (WiFi.status() != WL_CONNECTED && attempts < max_attempts) {
    delay(500);
    Serial.print(".");
    attempts++;
  }
    if (WiFi.status() == WL_CONNECTED) {
    Serial.println("Connected to Wi-Fi");
  } else {
    Serial.println("Failed to connect to Wi-Fi");
  }
}
void sendDataToThingzMate(String status) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
```

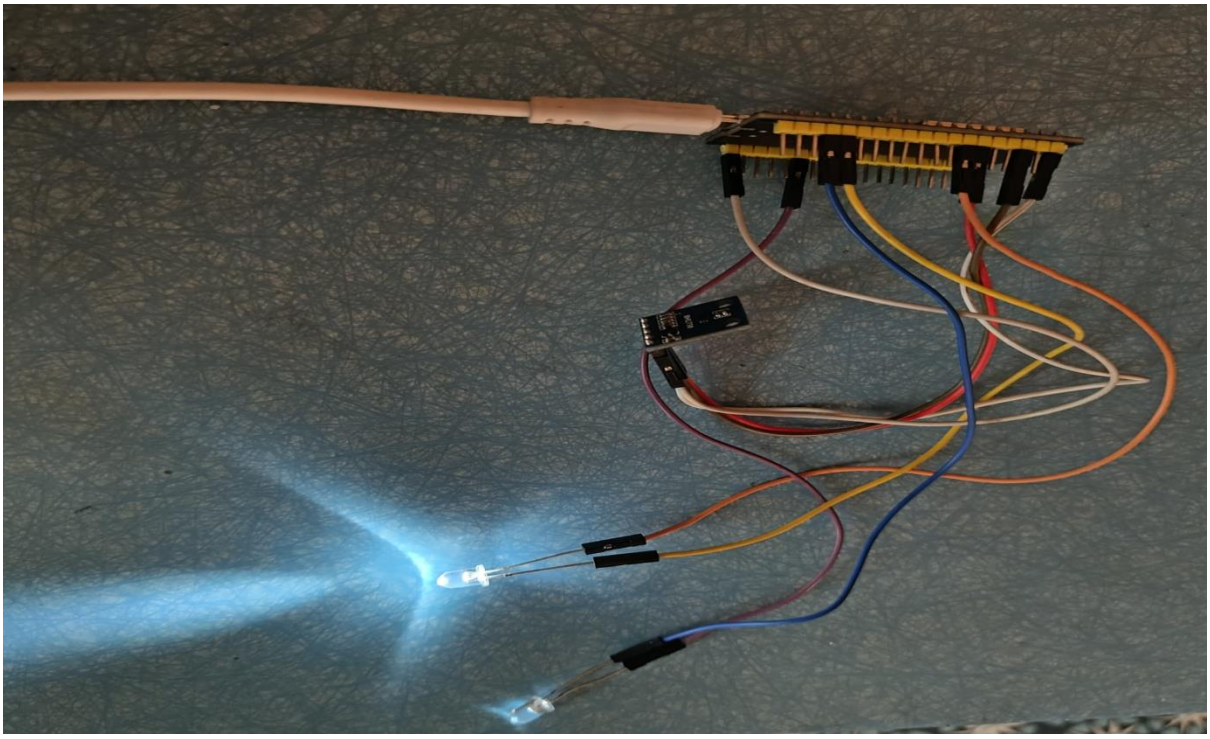
```
 // Set the request URL and headers
    http.begin(serverName);
    http.addHeader("Content-Type", "application/json");
    http.addHeader("Authorization", String("Bearer ") + apiKey);
    // Create JSON payload
    String payload = "{\"field1\":\"" + status + "\"}";
    // Send the POST request
    int httpResponseCode = http.POST(payload);
    // Check the response code
    if (httpResponseCode > 0) {
      Serial.print("HTTP Response code: ");
      Serial.println(httpResponseCode);
    } else {
      Serial.print("Error code: ");
      Serial.println(httpResponseCode);
    }
     http.end(); // Free resources
  } else {
    Serial.println("WiFi Disconnected");
  }
}
```

CLOUD OUTPUT:





OUTPUT RESULTS:



IoT Integration:

- **Cloud Integration:** Describe how data is sent to a cloud service for monitoring (e.g., using MQTT or HTTP).
- **Dashboard:** Discuss creating a user interface for monitoring and controlling the street lights remotely.

Testing and Results:

- **Test Cases:** Describe various scenarios tested, such as different lighting conditions and motion detection.
- **Results:** Provide screenshots or data showing the system's response to the test cases.

Conclusion:

The IoT-based street light control project successfully demonstrates the integration of smart technology into urban infrastructure, providing a practical solution to improve energy efficiency and operational management. By utilizing sensors and microcontrollers, the system automates street lighting based on real-time environmental conditions, ensuring lights are only active when needed. This approach not only reduces energy consumption but also lowers maintenance costs by minimizing manual intervention.

The system's remote monitoring and control capabilities allow for quick adjustments and real-time oversight, making it highly adaptable for larger-scale applications such as smart cities. Furthermore, the project highlights the potential for future enhancements, including the integration of renewable energy sources like solar power and the use of advanced analytics to optimize performance. Overall, this project represents a significant step toward more sustainable and efficient urban lighting solutions, showcasing the transformative power of IoT technology in public infrastructure.