STREET LIGHT AUTOMATION

Abstract :

This project presents a street light control system utilizing an ESP32 development module and a BH1750 ambient light sensor. The ESP32, a versatile microcontroller with built-in Wi-Fi and Bluetooth capabilities, serves as the central processing unit, while the BH1750 sensor provides accurate ambient light measurements. The system operates by continuously monitoring the ambient light levels using the BH1750 sensor. When the light intensity falls below a predefined threshold, indicating dusk or low-light conditions, the ESP32 activates the street lights. The ESP32 deactivates the street lights. This automated control mechanism ensures optimal energy usage by turning the lights on only when necessary.

Software Requirements :

- Arduino IDE :
    - ❖ Version : 1.8.19
    - ❖ OS : Windows 11 (32/64 bit)
- CH340 Driver
- ThingZmate

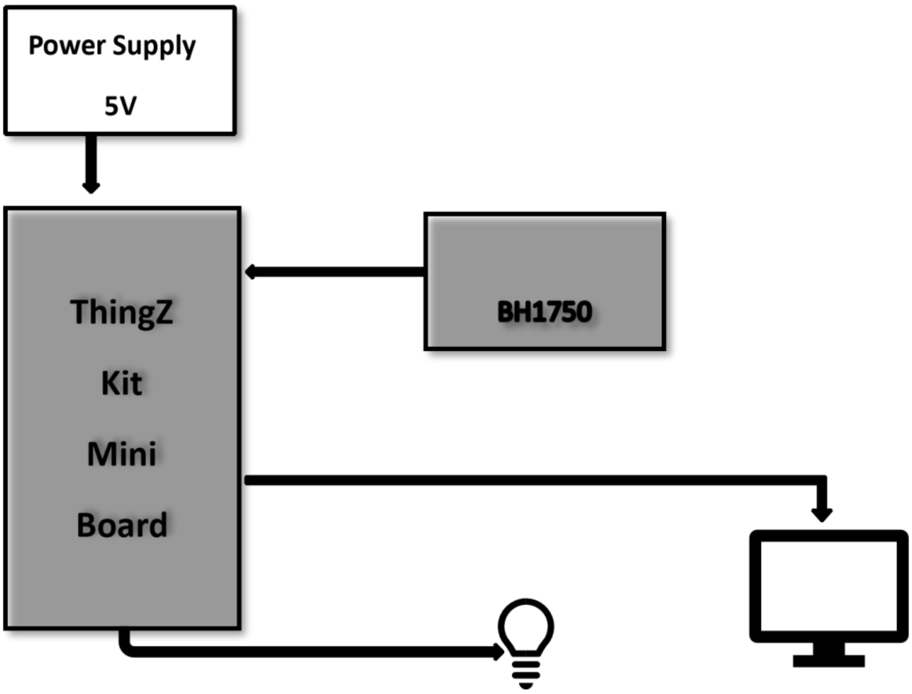
Hardware Requirements :

- ThingZkit Mini (ESP32-Controller)
- BH1750 Sensor

Component Required :

- LED
- USB cable
- Jumper wire

Block Diagram :



Connection Details :

ThingZkit mini Board	BH1750	LED
5v	VCC	
GND	GND	Cathode(-)
SDA-21	SDA	
IO 26		Anode(+)
SCL-22	SCL	

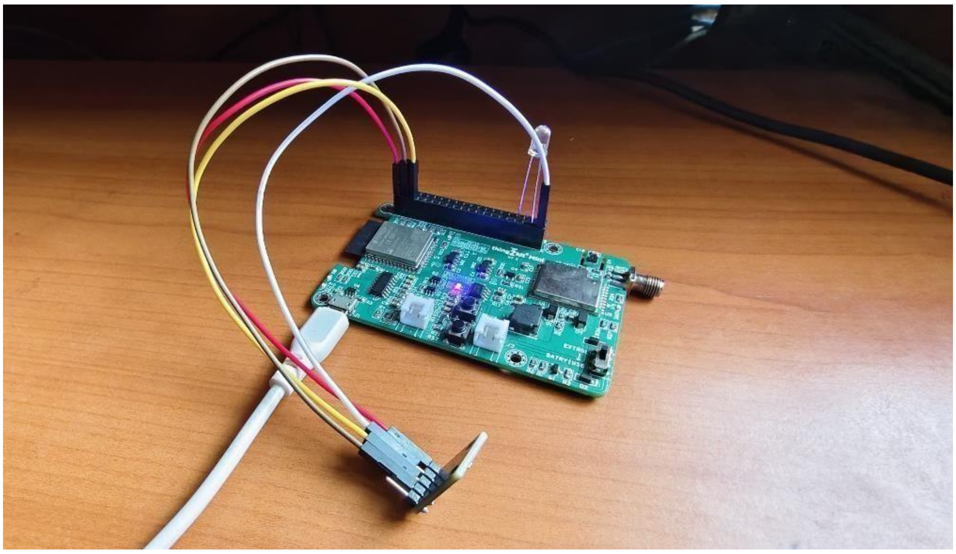
Source code :

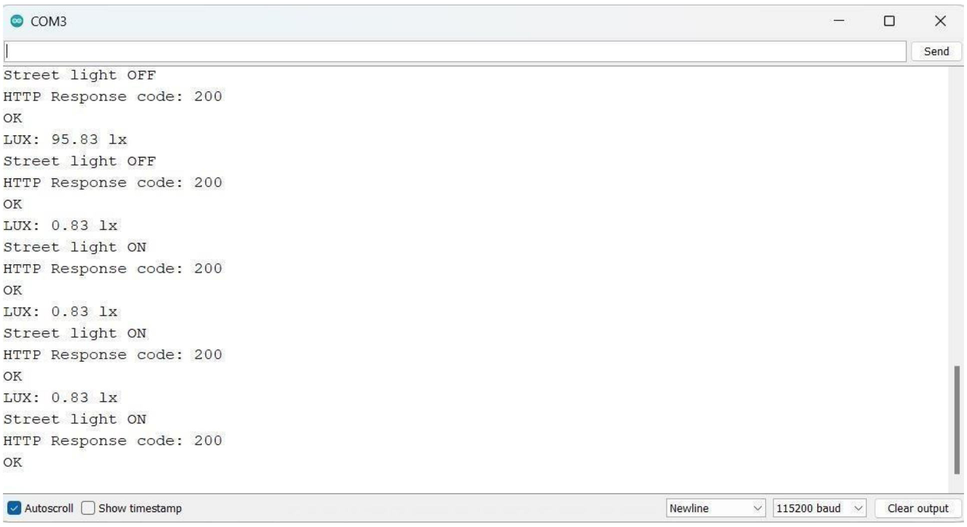
```
#include <WiFi.h>
#include <HTTPClient.h>
#include <BH1750.h>
#include <Wire.h>
#define WIFI_SSID "RA"
#define WIFI_PASSWORD "8086@000"
bool lightstatus; BH1750
lightMeter;
const char *serverUrl =
"http://console.thingzmate.com/api/v1/devicetypes/ra1/devices/smr/httpuplink";
// Replace with your server endpoint String AuthorizationToken = "Bearer
3e863807d0f719660c5c322b5a288ab1"; void setup() {
Serial.begin(115200);
 pinMode(26, OUTPUT);
digitalWrite(26, LOW);
Wire.begin();
lightMeter.begin();
  Serial.println(F("BH1750 Test begin"));
  // Connect to WiFi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi");
 while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
```

```
}

void loop() {   float lux = lightMeter.readLightLevel();
  Serial.print("LUX: ");
  Serial.print(lux);
  Serial.println(" lx");
  String light_status = (digitalRead(26) == HIGH) ? "ON" : "OFF";   if
(lux < 20) {     lightstatus=1;     digitalWrite(26, HIGH);
    Serial.println("Street light ON");
  } else {     lightstatus=0;
digitalWrite(26, LOW);
    Serial.println("Street light OFF");
  }
  if (WiFi.status()         ==        WL_CONNECTED)
{     HTTPClient          http;
            http.begin(serverUrl);      http.addHeader("Content-Type",
            "application/json");      http.addHeader("Authorization",
AuthorizationToken);
    // Create JSON payload
    String payload = "{\"LUX_VALUE\":" + String(lux) + ",\"light_status\":\"" + light_status
+ "\"}";
    // Send POST request     int httpResponseCode
= http.POST(payload);     if
(httpResponseCode > 0) {
      String response = http.getString();
      Serial.println("HTTP Response code: " + String(httpResponseCode));
      Serial.println(response);
    } else {
      Serial.print("Error code: ");
      Serial.println(httpResponseCode);
    }
http.end();
  }
delay(1000);
}
```

Hardware Connection Output:

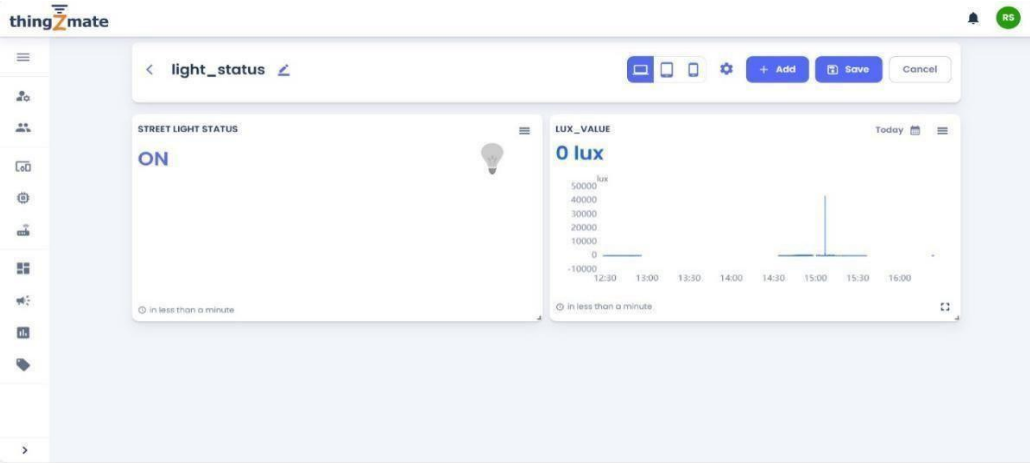




```
Street light OFF
HTTP Response code: 200
OK
LUX: 95.83 lx
Street light OFF
HTTP Response code: 200
OK
LUX: 0.83 lx
Street light ON
HTTP Response code: 200
OK
LUX: 0.83 lx
Street light ON
HTTP Response code: 200
OK
LUX: 0.83 lx
Street light ON
HTTP Response code: 200
OK
```

Cloud Output:

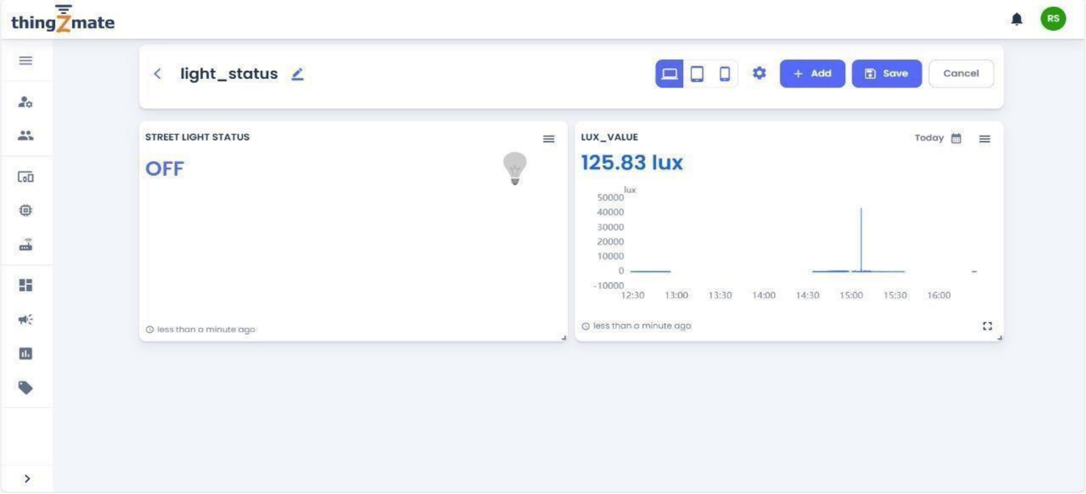
ON :



STREET LIGHT STATUS
ON

LUX_VALUE
0 lux

OFF :



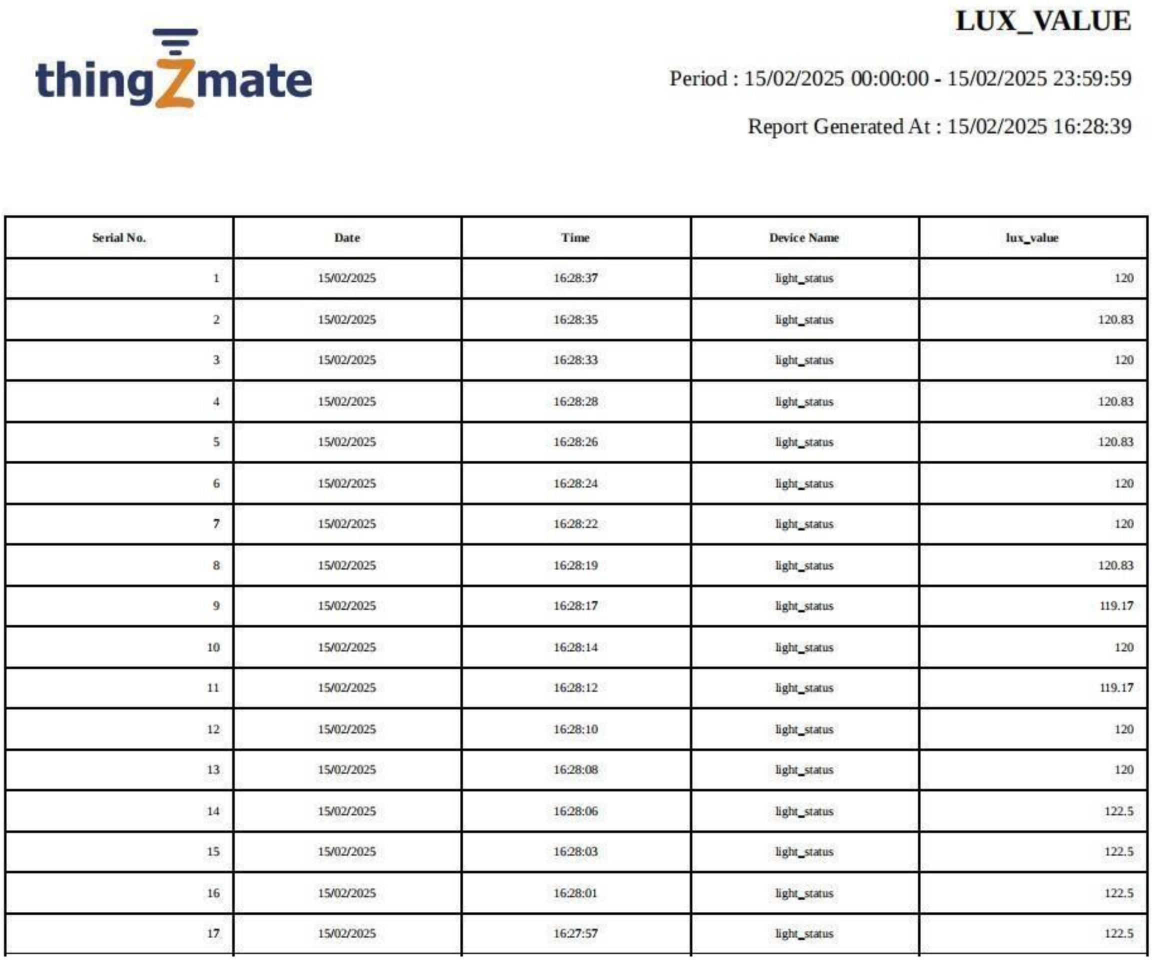
STREET LIGHT STATUS
OFF

LUX_VALUE
125.83 lux

Conclusion:

The proposed street light control system using BH1750 and Arduino with WiFi connectivity is an innovative solution for efficient and automated street lighting management. The system's ability to optimize energy consumption, ensure safety, and provide remote monitoring capabilities makes it an attractive solution for smart cities and urban development initiatives.

REPORT :



thingZmate

LUX_VALUE

Period : 15/02/2025 00:00:00 - 15/02/2025 23:59:59

Report Generated At : 15/02/2025 16:28:39

Serial No.	Date	Time	Device Name	lux_value
1	15/02/2025	16:28:37	light_status	120
2	15/02/2025	16:28:35	light_status	120.83
3	15/02/2025	16:28:33	light_status	120
4	15/02/2025	16:28:28	light_status	120.83
5	15/02/2025	16:28:26	light_status	120.83
6	15/02/2025	16:28:24	light_status	120
7	15/02/2025	16:28:22	light_status	120
8	15/02/2025	16:28:19	light_status	120.83
9	15/02/2025	16:28:17	light_status	119.17
10	15/02/2025	16:28:14	light_status	120
11	15/02/2025	16:28:12	light_status	119.17
12	15/02/2025	16:28:10	light_status	120
13	15/02/2025	16:28:08	light_status	120
14	15/02/2025	16:28:06	light_status	122.5
15	15/02/2025	16:28:03	light_status	122.5
16	15/02/2025	16:28:01	light_status	122.5
17	15/02/2025	16:27:57	light_status	122.5