Internet of Things

SMART STREET LIGHT MONITORING SYSTEM

Dhanush Gowdhaman-20Z214

Harsha Vardhan VM-20Z217

Kabilan KK-20Z223

Prednya Ramesh - 20Z239

Shri Shivathmika V - 20Z251

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE ENGINEERING

of Anna University

March 2023

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

Objective:

To design and develop a smart street LED monitoring system that can adjust the brightness of LEDs based on the intensity of light in the surrounding environment.

Problem Statement:

Traditional street lighting systems are not energy-efficient and are not adjustable to different lighting conditions. They are designed to provide maximum brightness during the night, even if the surrounding light levels are sufficient. This results in unnecessary energy consumption and light pollution. Therefore, there is a need for a smart street LED monitoring system that can adjust the brightness of LEDs based on the intensity of light in the surrounding environment.

Proposed Solution:

To address the above problem, we propose to design and develop a smart street LED monitoring system that uses an LDR sensor and an ultrasonic sensor to monitor the surrounding light and objects' distance. The system will adjust the brightness of LEDs based on the intensity of light in the surrounding environment.

Success Criteria for POC:

- The system should be able to adjust the brightness of LEDs based on the intensity of light in the surrounding environment.
- The system should be able to record the value of distance and intensity in Thingspeak cloud platform.
- The system should be able to operate continuously for 24 hours without any hardware or software failures.

Hardware and Software Requirements:

- NodeMCU
- LED
- LDR sensor
- 10K ohm resistor
- Ultrasonic sensor HCSR04
- Breadboard
- Jumper wires
- Micro USB cable
- Arduino IDE software
- ThingSpeak Cloud Platform

Implementation Plan:

1. Connect the NodeMCU to the computer using a USB cable.
2. Open the Arduino IDE software and install the required libraries for the NodeMCU board.
3. Connect the LDR sensor to the NodeMCU board and upload the code to the board.
4. Connect the ultrasonic sensor to the NodeMCU board and upload the code to the board.
5. Connect the LED to the NodeMCU board and upload the code to the board.
6. Test the system to ensure that it is working correctly.
7. Integrate the system with the Thingspeak cloud platform to record the value of distance and intensity.

Working:

The LDR (Light Dependent Resistor) sensor works by sensing the intensity of the light in the room. It is a passive electronic component that changes its resistance depending on the amount of light falling on it. In this scenario, the LDR sensor value is compared with a threshold value of 400 to determine if it is night time or not. If the LDR sensor value is less than 400, it indicates that it is night time.

Next, an ultrasonic sensor is used to detect the distance of an object from the sensor. If the distance is less than or equal to 5cm, it means that an object is very close to the sensor, and the LED needs to be at full brightness to provide sufficient illumination. If the distance is greater than 5cm, it indicates that no object is close to the sensor, and the LED can be dimmed to save power.

When the LDR sensor value is greater than 400, it means that it is not night time, and the LED should be turned off completely to save power.

To record the distance and intensity values, the ThingSpeak cloud platform can be used. It allows the user to collect and analyze data from sensors and other devices. The distance ,LDR sensor value and the LED brightness can be recorded on ThingSpeak, which can be used for further analysis and optimization of the street light system.

A webapp has been developed which takes the latest values of the distance,LDR sensor value and the LED brightness using the ThingSpeak API. The webapp is developed in such a way that it refreshes and always takes the latest values.

Code:

```
#include <ESP8266WiFi.h>

#include "ThingSpeak.h"



const int trigpin=D3;

const int echopin=D2;

const int ledpin=D7;

const int ldrpin=A0;
```

```
long duration;

int distance;

int safedistance;

int light;

int intensity=3;


const char* apiKey = "0IAORLIZBKLHXJ9W"; // write your "Write API key"

const char* ssid = "Habibi_come_to_JAIN"; // write your "wifi name"

const char* password = "24240802"; // write your "wifi password"

const char* server = "api.thingspeak.com";

WiFiClient client;


unsigned long myChannelNumber = 2083199;


void setup() {

  pinMode(trigpin,OUTPUT);

  pinMode(echopin,INPUT);

  pinMode(ledpin,OUTPUT);

  pinMode(ldrpin,INPUT);

  WiFi.begin(ssid, password);

  Serial.println();

  Serial.print("Connecting to ");

  Serial.println(ssid);
```

```
 while (WiFi.status() != WL_CONNECTED)

 {

   delay(500);

   Serial.print(".");

 }

 Serial.println("");

 Serial.println("WiFi connected");


 ThingSpeak.begin(client);  // Initialize ThingSpeak

}


void loop() {

 Serial.begin(9600);

 digitalWrite(trigpin,LOW);

 delayMicroseconds(2);

 digitalWrite(trigpin,HIGH);

 delayMicroseconds(10);

 digitalWrite(trigpin,LOW);

 duration=pulseIn(echopin,HIGH);

 distance=duration*0.034/2;

 safedistance=distance;

 light=analogRead(ldrpin);
```

```
if(light<=400){                //night

  if(safedistance<=5){

    analogWrite(ledpin, 255);  /* set initial 1000% duty cycle of 1023 */

    //digitalWrite(ledpin,HIGH);

    delay(30);

  }

  else{

    analogWrite(ledpin, intensity);                /* set intensity as 30% of 1023 */

    //digitalWrite(ledpin,LOW);

}}

 else      //day

 {

  digitalWrite(ledpin,LOW);

  delay(30);

 }

 delay(100);

 Serial.println("Dist: "+String(distance));

 Serial.println("LDR: "+String(light));

 // we have made changes in program to decrease the latency.

 // but it will take minimum 15 second to post data on Thingspeak channel.

 // set the fields with the values

 ThingSpeak.setField(1, distance);

 ThingSpeak.setField(2, light);
```

```
int x = ThingSpeak.writeFields(myChannelNumber, apiKey);

if(x == 200){
  Serial.println("Channel update successful.");
}
else{
  Serial.println("Problem updating channel. HTTP error code " + String(x));
}
delay(100);
}
```

Resources and Cost:

- NodeMCU: Rs. 420
- LED: Rs. 5
- LDR sensor: Rs. 75
- Ultrasonic sensor: Rs.120
- Breadboard: Rs. 90
- Jumper wires: Rs.150
- USB cable: Rs.30
- Arduino IDE software: Free
- ThingSpeak Cloud Platform: Free

Risk and Mitigation Strategies:

- The hardware may fail due to electrical or mechanical failures. To mitigate this risk, we can use high-quality components and perform regular maintenance.
- The software may have bugs or errors that could lead to system failure. To mitigate this risk, we can perform extensive testing and debugging before deploying the system.
- The system may face security issues due to unauthorized access or hacking. To mitigate this risk, we can use secure protocols and implement authentication and authorization mechanisms.

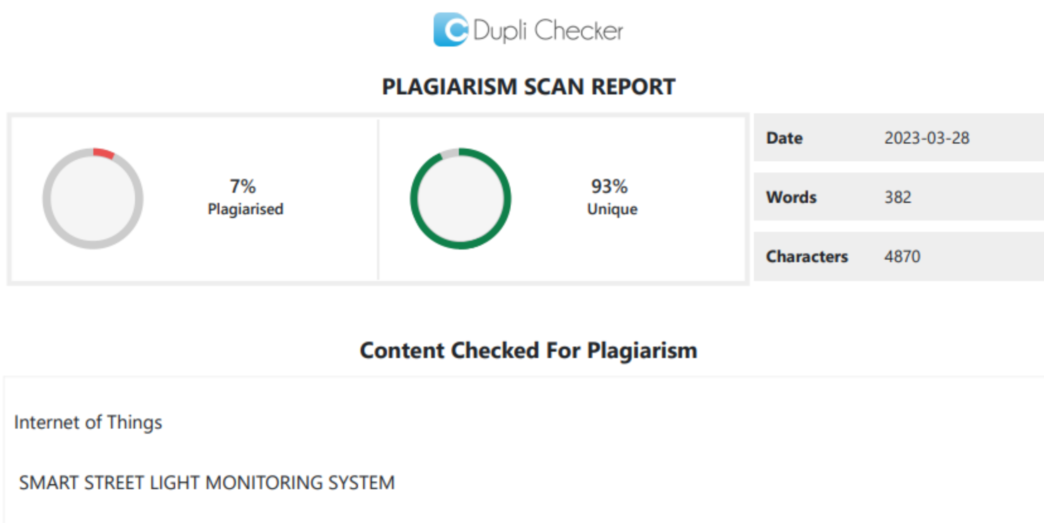
Advantages and Benefits:

- Energy-efficient: The smart street LED monitoring system will adjust the brightness of LEDs based on the intensity of light in the surrounding environment, resulting in reduced energy consumption.
- Cost-effective: The system is designed using low-cost components and open-source software, resulting in a cost-effective solution.
- Easy to maintain: The system is designed using simple components and can be easily maintained and repaired.
- Environmentally friendly: The smart street LED monitoring system will reduce light pollution and contribute to a cleaner environment.
- Reduced energy consumption
- Cost savings
- Improved street lighting
- Reduced light pollution
- Cleaner environment

Conclusion:

In conclusion, the proposed smart street LED monitoring system using IoT technology offers a practical and cost-effective solution for improving lighting control and reducing energy consumption. The system uses NodeMCU board, LDR and ultrasonic sensors, and LED lights to monitor the light intensity in a room and adjust the brightness of the lights accordingly. By leveraging ThingSpeak cloud platform to store and analyze data, users can monitor and optimize the system performance.

Plagiarism Report:


Dupli Checker

PLAGIARISM SCAN REPORT

7% Plagiarised

93% Unique

Date	2023-03-28
Words	382
Characters	4870

Content Checked For Plagiarism

Internet of Things

SMART STREET LIGHT MONITORING SYSTEM