SMART STREET LIGHTNING SYSTEM

Submitted by

1. Anand Kumar Sahu (201020405)

2. Khelawan Singh (201020424)

3. Kshitij Kumar (201020426)

4.Tejendra Kanwar (201020458)

Data Science and Artificial Intelligence Engineering

under the guidance of

Dr. Venkenna U

Dr. Shyama Prasad Mukherjee

International Institute of Information Technology, Naya Raipur

Abstract

One of the major concerns faced by today’s cities is the efficient management of energy because, with the growing population, demand for electricity has to be met by the limited resources. A basic need for electricity is for lighting in public and private residential areas. Today, approximately 10% of the total energy distribution is consumed by public lighting. Undoubtedly, the efficient management and control of lighting systems in a smart city can be a preferred methodology to overcome this issue**.**This project proposes a model for modifying street light illumination by using sensors at minimum electrical energy consumption.Model mainly operates on two modes --Dim mode,when there is no vehicle in the road and Bright mode when the sensors detects presence of Vehicle on the road,This model shows real time power consumption and cost saving over conventional street lighting systems.

1.Introduction

At the beginning, streetlamps were controlled by manual control where a control switch is set in each of the street lamps which is called the first generation of the original street light. After that, another method that has been used was an optical control method done using a high pressure sodium lamp in their system. Nowadays, it is seen that the method is widely used in the country. The method operates by setting up an optical control circuit,changing the resistance by using a light sensitive device to control street lamps that light up automatically at 9 dusk and turn off automatically after dawn in the morning **.**

In India there are almost 35 million street lights and at night when no one is using that light is on and energy is wasted. In an hour about 5,250 million watt energy is consumed at an average by street light only. If atleast one hour of energy is saved per day by street-light then it will save 5,250 million watt of energy per day.

This project implements an IoT based automatic Smart Street Lightning System. It incorporates Intelligent light sensing which refers to public street lighting that adapts to movement by pedestrians, cyclists and cars. Intelligent street lighting, also referred to as adaptive street lighting, dims when no activity is detected, but brightens when movement is detected. This type of lighting is different from traditional, stationary and illumination, or dimmable street lighting that dims at predetermined times. The research work shows automatic control of street lights as a result of which power is saved to some extent.

- Smart Street Lighting System contributes to more livable leading to greater cost savings compared to simple LED luminaires.

- **Increased revenue opportunities -** Smart Street lights can provide new revenue opportunities with digital signage .

• **Reduced energy cost -** Smart street lights more efficiently manage electricity, leading to greater cost savings compared to simple LED luminaires.

2.OBJECTIVE

- The objective of Smart Street Lighting System is buila model which can obtain an autonomous and more efficient street lighting management system using IoT Technology.
- To reduce energy consumption of street light by avoiding wastage.

3. PROPOSED METHODOLOGY

The present system employs power delivery via a single phase line to the streetlight.
The proposed system involves five more components to regulate the power delivery.
A Microwave Radar Sensor at the base of the street light detects presence in a small area around the street light. The data from the sensor is sent to the Node MCU which forms the brain of the circuit. The Node MCU then commands an IRF520 MOSFET to switch between dim and bright modes depending upon the requirement and thus controls the brightness of the street light. A battery adapter, also powered by the single phase line, is used to supply 5V inputs to the sensors and Node MCU.

Components Required

- ESP8266 NodeMCU
- Micro USB cable
- LEDs
- Jumper wires
- Microwave RADAR sensors
- LDR sensors
- IRF520 MOSFET

Before moving forward, let's see about the major components used in the project.

<u>ESP8266 NodeMCU:</u> **ESP8266 NodeMCU** 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.

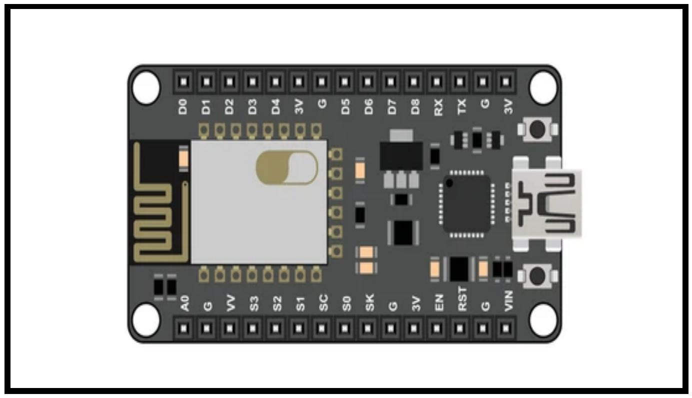


Fig. Node MCU ESP8266

<u>LDR SENSOR:</u> 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 lights, alarm clocks, automatic brightness and contrast control etc.

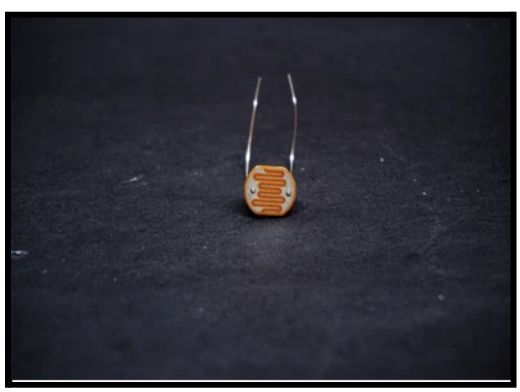


Fig .LDR Sensor

Microwave RADAR Sensor: Microwave sensors, also known as Radar, RF or Doppler sensors, detect walking, running or crawling human targets in an outdoor environment. Microwave sensors generate an electromagnetic (RF) field between transmitter and receiver, creating an invisible volumetric detection zone.

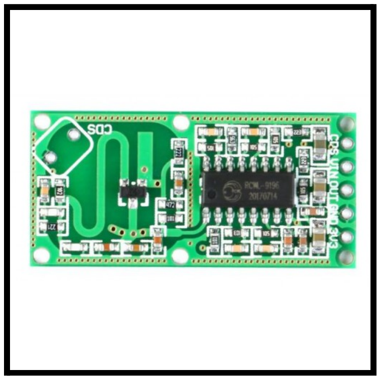
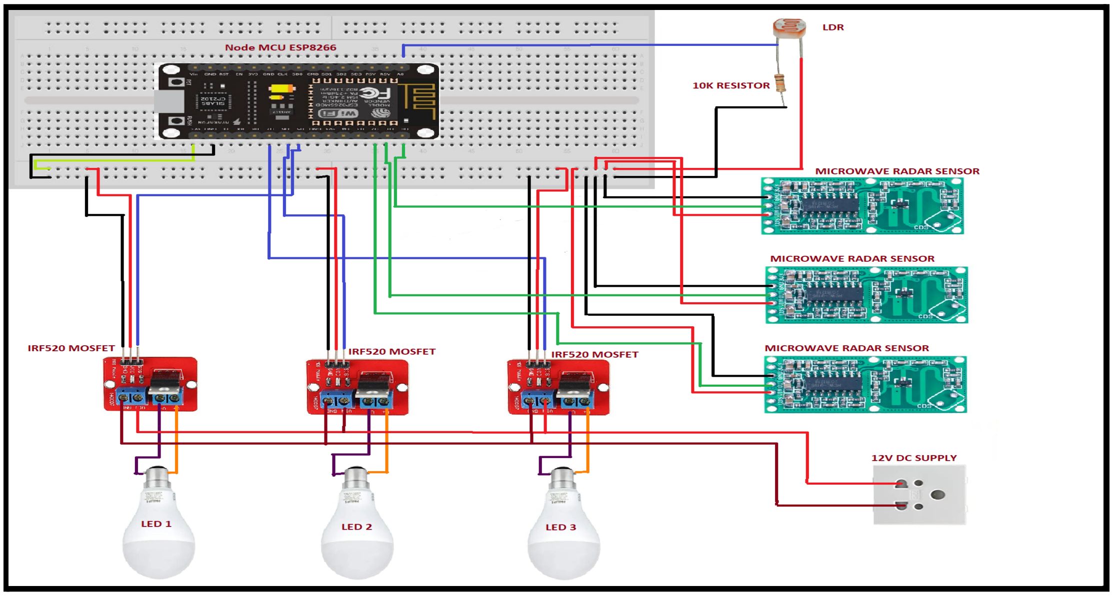


Fig. Microwave Radar sensor

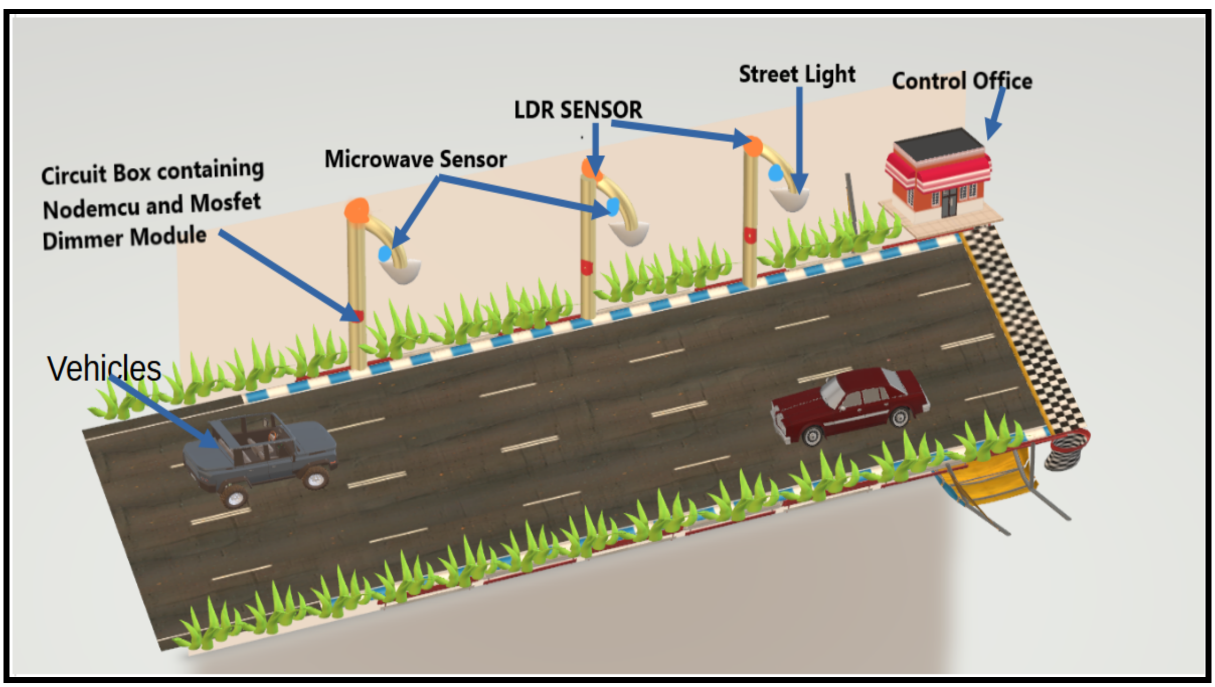
IRF520 MOSFET: It is a metal oxide semiconductor field-effect transistor used to switch or amplify voltages in circuits. The purpose of a MOSFET transistor is essentially to control voltage/current flow between the source and the drain.

Fig. IRF520 MOSFET

CIRCUIT DIAGRAM



CONCEPTUAL DIAGRAM:-

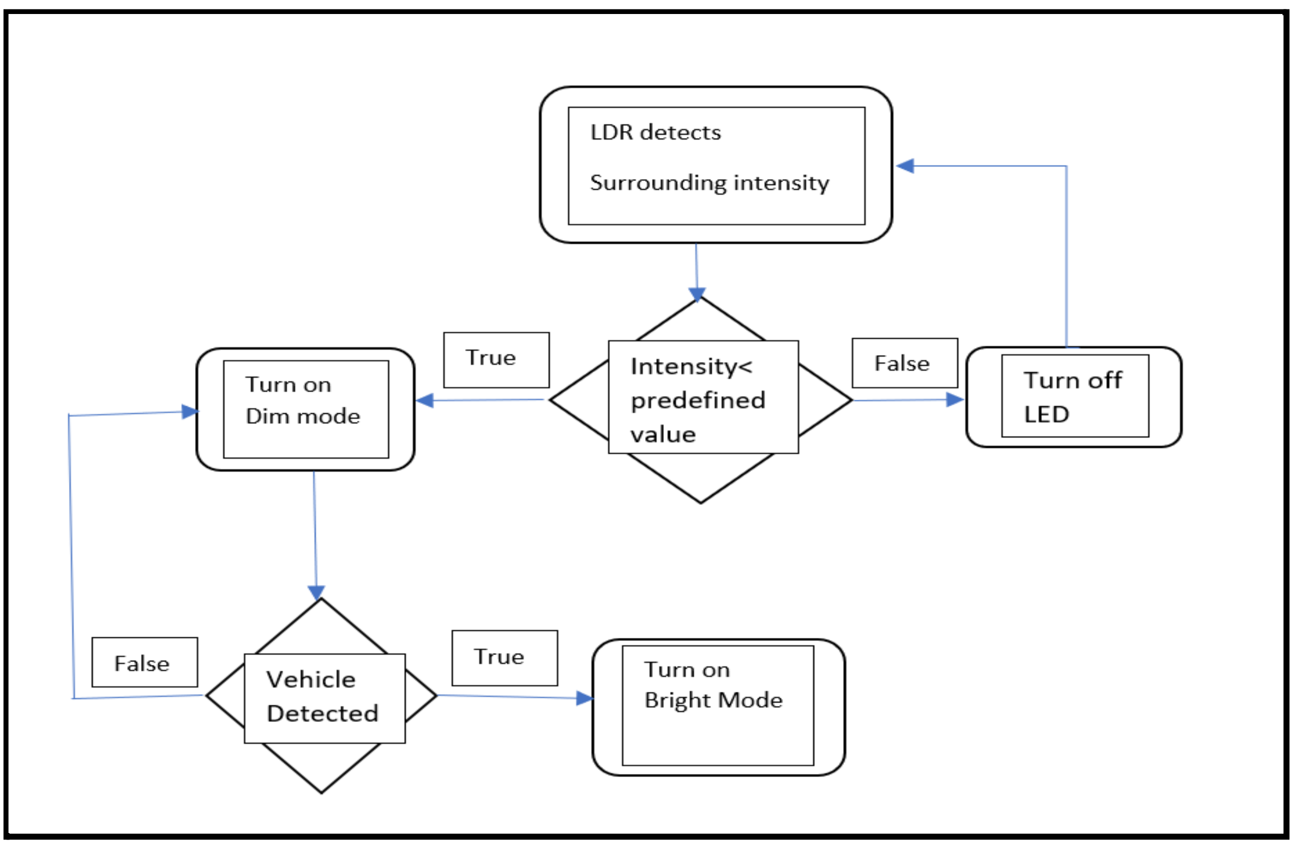


This figure shows the positioning of various sensors and controlling devices in the prototype.

LDR is mounted above the street light pole to detect the light intensity of the surrounding and send data to nodemcu. Microwave sensor is mounted parallel to the surface of the road to detect vehicles passing below the street light. A Light Dimmer(IRF520 MOSFET) is attached in the circuit box which controls the intensity of the LED Bulb according to the command given by NodeMCU.This is a conceptual diagram to better understand the model.

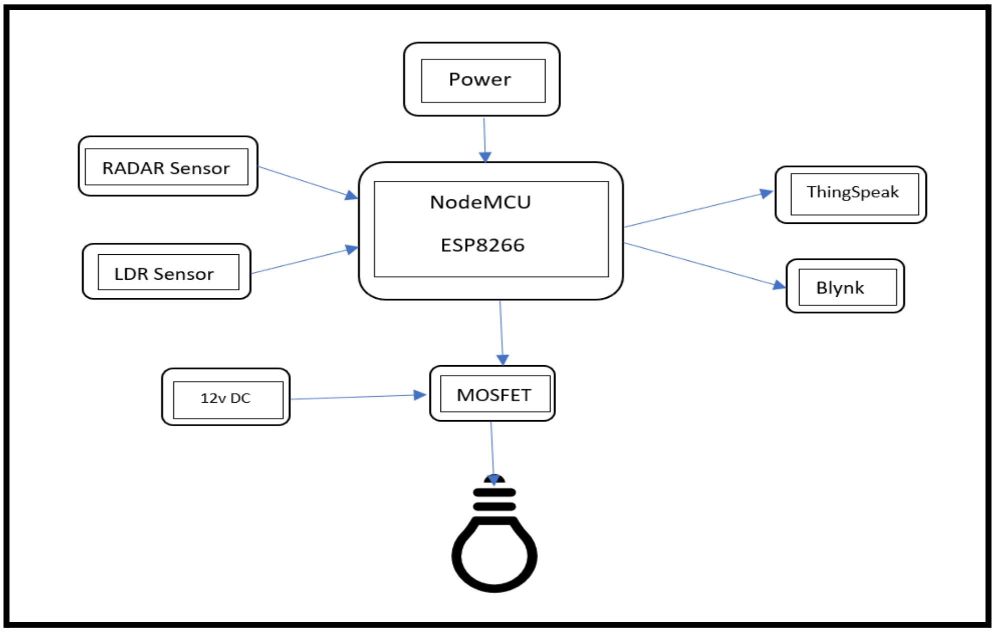
WORKING OF THE MODEL :

Firstly, the LDR sensor detects the light intensity of the surrounding area . Naturally during day time the intensity will be much higher than the preset value of intensity. There is no need to turn on the light. During Night, the observed intensity will go down the preset value which commands to dimmer turn on the lights at dim mode.
Lights will be dimmed until the Microwave sensor detects any movement. When Microwave sensor detects any vehicle nearby , the lights will be turned on into Bright Mode. And the Vehicle Passes by, lights will again come back to dim mode to save power as there is no need. In this way Light will only light up the road when there is need,otherwise it will be in energy saving mode.This method will save a significant amount of electricity as the street light will only turn on if there is someone present in the Street.



Flow Chart

BLOCK DIAGRAM



This diagram shows the principal parts and functions represented by blocks connected by lines that show the relationship of the block.

a)Input -LDR sensor and Microwave radar sensor collects data or parameters of the surrounding like ldr sensor detects environment intensity and microwave radar sensor detects any vehicle in the area under its range.

b)Processing -Nodemcu is the main control unit which collects data from the sensor, processes it and gives commands to actuators.

c)Output - According to the received command the light dimmer controls the intensity of the LED Bulb.

d)Communication - The embedded wifi module in the nodemcu helps to reach the internet and sends data to further analysis.

e)Storage -Data is stored over cloud(Thingspeak) and Blynk on which it is analysed further.

f)Interface - There are two user interfaces involved i.e Thingspeak Web Interface and Blynk Mobile Interface.

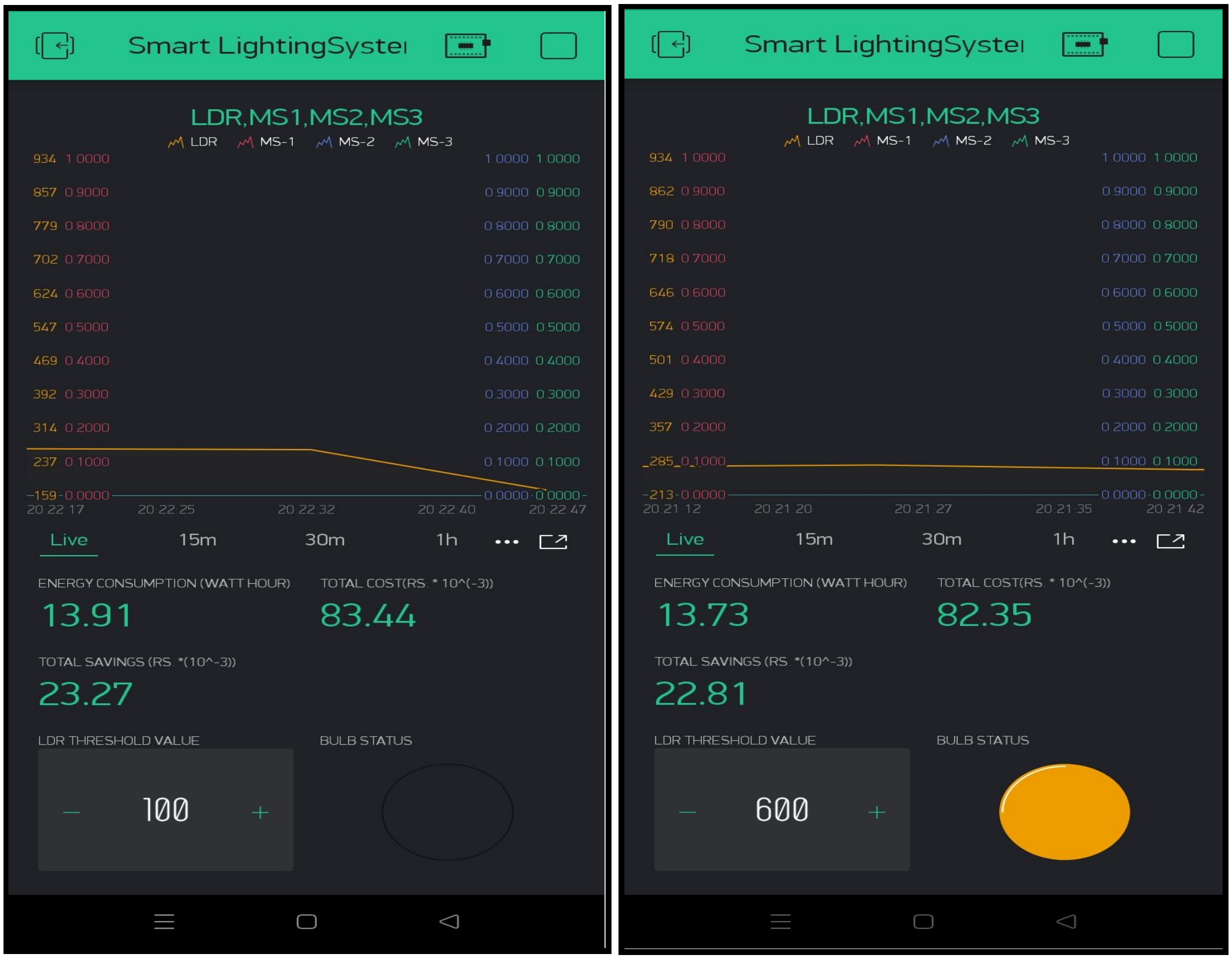
Prototype:

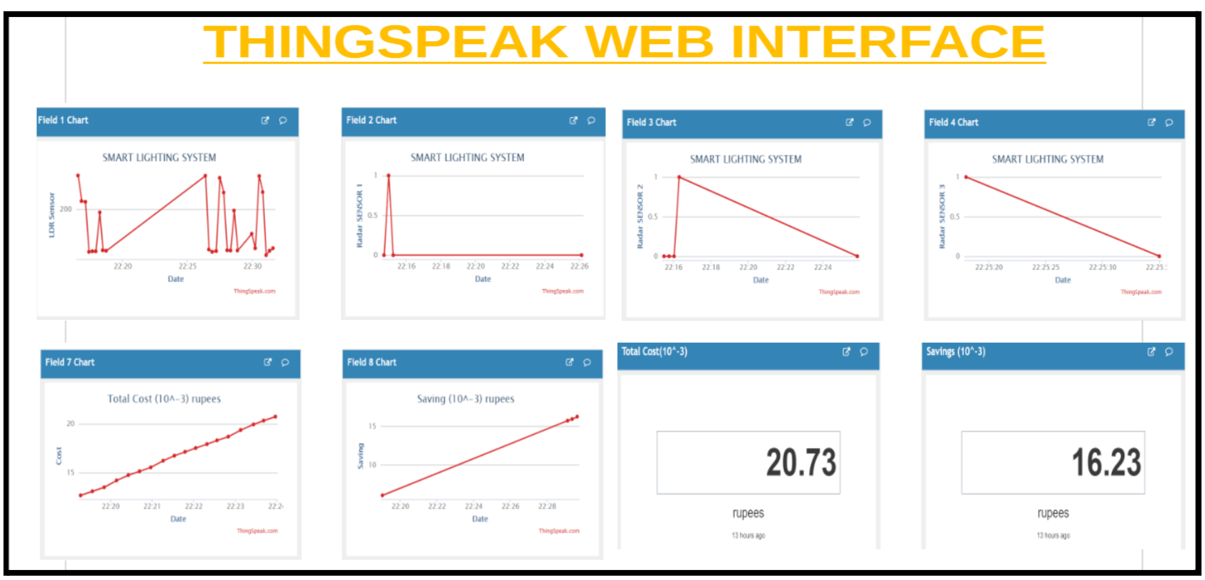
This is how a **Smart Street Light** works, it only glows if it is night time and someone is passing through the street. And it can also be controlled manually from anywhere in the world using the **Blynk** app interface.

4. _RESULTS:

The below figures show the amount of energy consumed and cost of saving if this model is used over existing solutions . We can observe the daily usage of the power consumed shown in the chart. As there is a consistency which implies that our system is working effectively. The above results obtained are best suitable to the real world and they further can be utilized for the energy preservation and the obtained, provides the necessity of using the module and how it can be utilized in order to preserve and store energy. This IOT based computerized streetlight framework is financially savvy. It can likewise wipe out the CO2 outflows and light contamination. The framework doesn't require labor and occasional check rather the framework status is consistently refreshed.

Blynk Interface





Once the LDR threshold value is send through Blynk the lights operates according to that value only and thus the bulb glows by comparing the threshold value to the present surrounding light intensity.We can see in the Blynk image shown above.We can also receive the total energy consumption,total cost that we get by setting the required cost per unit of that area and total savings by comparing to the cost used by energy consumption in the conventional lighting system.We can also see the bulb status of the LED bulb in the real time.Also we can change the brightness or change the state of the bulb as required in that area from sitting anywhere in the world.In case of any damage we can also get the alert about the damaged bulb by further extending the project.We can also get this all values in thingspeak web interface.

5 .COMPARISON OF PROPOSED SOLUTION WITH EXISTING SOLUTION

We have seen in the number of cities where the street lights are one of the huge energy expenses for a city. Currently we have a manual system where the lights will be switched ON in the evening before the sunset and they are switched OFF the next morning after there is sufficient light outside. Also sometimes the lights are switched ON all the time if there are no people available to switch it.So there is a lot of energy waste between ON and OFF timing.

Disadvantages of Existing System

- Manual Switching off/on of Street Lights
- More Energy Consumption.
- High expense.
- More manpower.

Advantages of the Proposed System

- Automatic Switching of Street lights.
- Maintenance Cost Reduction.
- Reduction in CO_2 emission.
- Reduction of light pollution.
- Wireless Communication.
- Energy Saving.
- Reduction of manpower.
- Frequently contacted if the street light is damaged or there is some fault in power supply.

6.CONCLUSION

This project **"IoT Based Smart Lighting System for Smart City "** is a cost effective , practical, eco-friendly and the safest way to save energy and this system's live status information can be accessed from anytime and anywhere. It clearly tackles the two problems the world is facing today, saving energy and also disposal of incandescent lamps, very efficiently. Initial cost and maintenance can be the drawbacks of this project.

.This Smart Street light project not only helps in urban areas but is also beneficial in rural areas too. As we are moving towards more advancement we require more power, so saving resources can be an advantage. With this project, we can even add smart parking of vehicles and it is even useful for driverless cars.

With the advances in technology and good resource planning the cost of the project can be cut down and also with the use of good equipment the maintenance can also be reduced in terms of periodic checks. The LEDs have a long life, emit cool light, do not have any toxic material and can be used for fast switching. For these reasons our project presents far more advantages which can overshadow the present limitations. Keeping in view the long term benefits and the initial cost would never be a problem as the investment return time is very less. The project has scope in various other applications like for providing lighting in industries, campuses and parking lots of huge shopping malls.

Conserving energy has been a huge task in our generation. By converting the manual process into automation we can save enormous amounts of energy. These also reduce manpower and prevent energy wastage. The efficiency of automated systems is more than the manual systems. We can reprogram these devices with respect to our needs. By using the API key, the generated data is stored in the Thingspeak database which we can use for future references. Only maintenance is the disadvantage of these systems.

7.LEARNING OUTCOMES

Currently we use manual systems to operate the street lights, this leads to the enormous energy waste all over the world and it should be changed. In this project we studied how IoT is used to develop street lights in a smart way for our modern era. It is an important fact to solve the energy crises and also to develop street lights for the entire world. In addition, with the study on smart street lighting systems we analyzed and described different sensors and components which are used in the IoT environment. All the components of this survey are frequently used and very modest but effective to make the unswerving intelligence systems.

Reference

[1] https://ieeexplore.ieee.org/document/8878770

[2] https://iotdesignpro.com/projects/iot-based-smart-street-light-using-esp8266-and-thingspeak

[3] https://www.youtube.com/watch?v=17vqLv508Uw

[4] https://www.youtube.com/watch?v=p06NNRq5NTU

[5] www.roboshala.com

Source Code

```
//Here we begin with the code

#include <ESP8266WiFi.h>;
#include<WiFiClient.h>;
#include<BlynkSimpleEsp8266.h>
#define BLYNK_PRINT Serial // Enables Serial Monitor
#include <ThingSpeak.h>;  //Importing thingspeak library

const char* ssid = "wifi name";   //wifi name
const char* pass = "password";          //wifi password

WiFiClient client;

unsigned long myChannelNumber = 1432792;         //Thingspeak channel number
const char * myWriteAPIKey = "VV604FV82A7TfrLV"; //Thingspeak API key

char auth[] = "1mKYOrvaY-C3cc0gSbsex_RIs5At_9eC"; //Blynk authorisation key

//Defining variables
int val;
int input_val=0;
int LDRpin = A0;
int led1 = 14;   //D5
int led2 = 12;  //D6
int led3 = 13;  //D7
int P_C = 0;
```

```
float P_sum =0.0;
float P1=0.0;
float P2=0.0;
float P3=0.0;
float sum = 0.0;
float last_time ;
float current_time;
float Wh = 0;
float price = 0;
float W=0;
float C_price = 0;
float saving =0;
int input_price=6;

#define BULB_PIN 14,12,13        // D5,D6,D7
WidgetLED BULB(V5);              //Bulb status in Blynk app

void setup() {
  Blynk.begin(auth, ssid, pass);
  Serial.begin(9600);

  pinMode(16, INPUT); //D0
  pinMode(5, INPUT); //D1
  pinMode(4, INPUT); //D2
  pinMode(LDRpin, INPUT);

  pinMode(led1, OUTPUT);  //D5
  pinMode(led2, OUTPUT);  //D6
  pinMode(led3, OUTPUT);  //D7
  delay(10);
  WiFi.begin(ssid, pass);

  ThingSpeak.begin(client);

}

BLYNK_WRITE(V6)      //For writing LDR threshold value from Blynk
{
  input_val = param.asInt();  //assigning incoming value from pin V6 to a variable
```

```
Serial.print("The Threshold Value is: ");
  Serial.print(input_val);
  Serial.println();
}

void loop() {

  Blynk.run();
  val = analogRead(LDRpin);
  Serial.println(val);
  Serial.println(".Microwave sensor 1:"+digitalRead(16));
  Serial.println(".Microwave sensor 2:"+digitalRead(5));
  Serial.println(".Microwave sensor 3:"+digitalRead(4));

  delay(100);
  //sending data to thingspeak in respective channel number and field
  ThingSpeak.writeField(myChannelNumber, 1, val, myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 2, digitalRead(16), myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 3, digitalRead(5), myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 4, digitalRead(4), myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 6, P_C, myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 5, P_sum, myWriteAPIKey);

  //Comparing threshold value
  if (val <= input_val)
  {
    P_C = 27;  //power consumption when all the bulbs are glown with its full brightness
    digitalWrite(led1, HIGH);
    analogWrite(led1, 255 / 20);

    digitalWrite(led2, HIGH);
    analogWrite(led2, 255 / 20);

    digitalWrite(led3, HIGH);
    analogWrite(led3, 255 / 20);
    P1=1;
    P2=1;
    P3=1;
```

```
if (digitalRead(16) > 0) {  //If sensor detects the vehicles
    digitalWrite(led1, HIGH);
    analogWrite(led1, 255);
    BULB.on();
    delay(1000);
    P1 = 9;

  }
  else {
    digitalWrite(led1, HIGH);
    analogWrite(led1, 255 / 20);
    P1 = 5;

    BULB.on();

  }//----------------------------------------------
  if (digitalRead(5) > 0) {
    digitalWrite(led2, HIGH);
    analogWrite(led2, 255);
    P2=9;
    BULB.on();
    delay(1000);

  }
  else {
    digitalWrite(led2, HIGH);
    analogWrite(led2, 255 / 20);
    P2=5;
    BULB.on();

  }//---------------------------------------------
  if (digitalRead(4) > 0) {
    digitalWrite(led3, HIGH);
    analogWrite(led3, 255);
    P3=9;
    BULB.on();
    delay(1000);
```

```
   }
 else {
     digitalWrite(led3, HIGH);
     analogWrite(led3, 255 / 20);
     P3 = 5;
     BULB.on();

   }//--------------------------------------------------------



  }
  else {
   digitalWrite(led1, LOW);
   digitalWrite(led2, LOW);
   digitalWrite(led3, LOW);
   P_C = 0;     //power consumption when no bulb glows
   P1=0;
   P2=0;
   P3=0;

   BULB.off();

  }
  //For electricity consumption and power cost calculation
  P_sum = P1+P2+P3;
  last_time = current_time;
  current_time = millis();
  Wh = Wh +P_sum*((current_time-last_time)/3600000.0);
  price = (Wh*input_price);
  W = W +P_C*((current_time-last_time)/3600000.0);
  C_price = (W*input_price);
  saving = saving +(C_price - price);

  ThingSpeak.writeField(myChannelNumber, 7, price, myWriteAPIKey);
  ThingSpeak.writeField(myChannelNumber, 8, saving, myWriteAPIKey);

//sending data to blynk in virtual key
  Blynk.virtualWrite(V1,val);
```

```
 Blynk.virtualWrite(V2,digitalRead(16));
 Blynk.virtualWrite(V3,digitalRead(5));
 Blynk.virtualWrite(V4,digitalRead(4));
 Blynk.virtualWrite(V7,Wh);
 Blynk.virtualWrite(V8,price);
 Blynk.virtualWrite(V9,saving);

 delay(1000);
}
```