Dr. B R Ambedkar National Institute of Technology

Jalandhar

Final Year Project Report

on

Smart Street Lighting System

Department of Electronics and Communication Engineering

Submitted By:

Arun Bagga (16106013)

Anmol Manik Singh (16102012)

Rajdeep Singh (16104057)

ABSTRACT

Conventional street lighting systems in areas with a low frequency of passersby are online most of the night without purpose. The consequence is that a large amount of power is wasted meaninglessly. With the broad availability of flexible-lighting technology like light-emitting diode lamps and everywhere available wireless internet connection, fast reacting, reliably operating, and power-conserving street lighting systems become reality.

The solution we propose here involves the system that considers the vehicles/pedestrians passing below the light and accordingly adjusts the glowing intensity of the neighboring light. This will lead to burning the minimum power when no object is under the light. According to the researches, it will reduce the total power consumption by 30-40% as compared to traditional static lighting system. The added advantage is that system itself will generate the notifications about the street light current health and will help in fault detection and maintenance.

ACKNOWLEDGEMENTS

The idea of working on this project came from various sources. Those are mentioned as below.

We respect and thank Mr. Pawan Kumar Verma, for providing us an opportunity to workon this project entitled ‘Smart Street Lighting System’ under his guidance. He encouraged us to explore the fields related to networking of devices and hence we arrived on this project.

I owe my deep gratitude to Dr. R.K. Sunkaria(Head, Department of Electronics and Communication Engineering) who gives us the great opportunity to workon real life problem related projects.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Department of Electronics & Communication Engineering who taught us well so that we feel motivated and capable for this project.

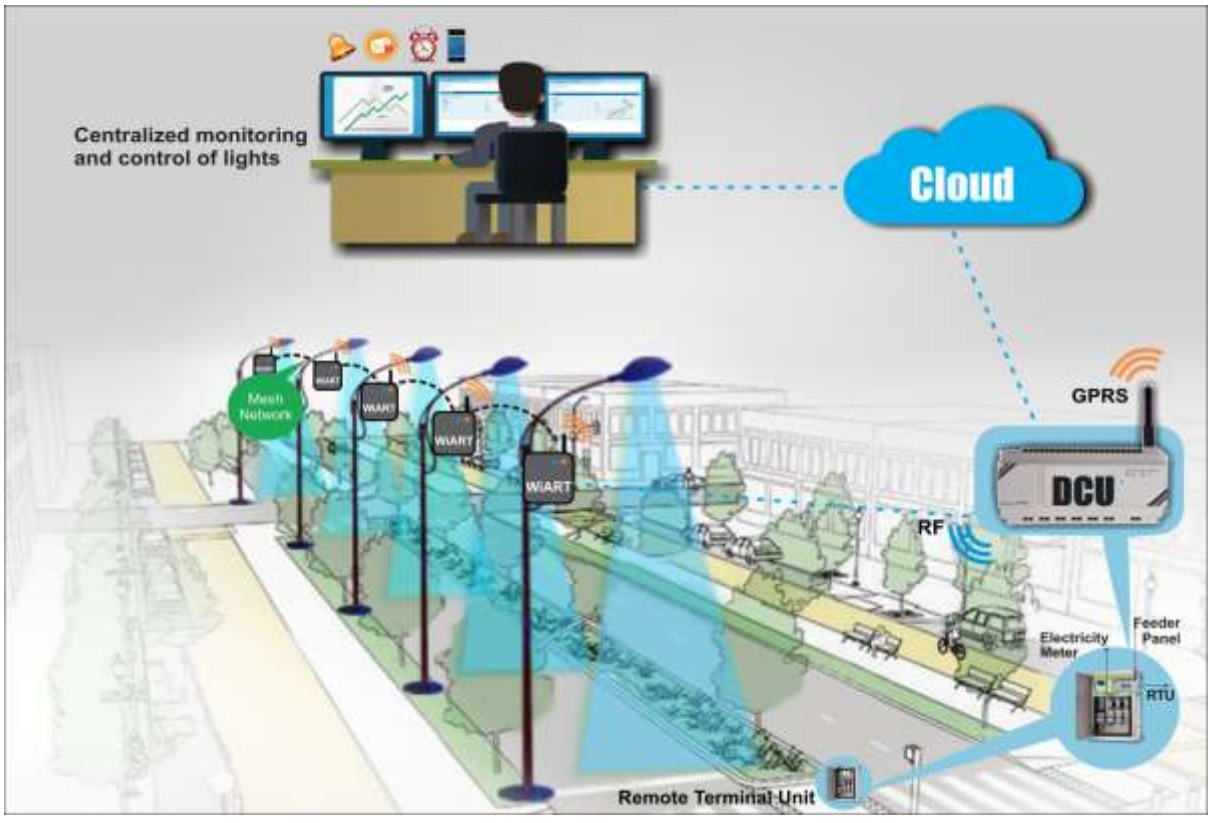
TABLE OF CONTENTS

1. INTRODUCTION

1.1 Overview

Lighting systems, particularly within the public sector, are still designed per the previous standards of reliability and that they don't usually take benefit of latest technological developments. Recently, however, the increasing pressure associated with the raw material prices and also the increasing social sensitivity to CO2 emissions are leading to develop new techniques and technologies which permit significant cost savings and larger respect for the environment.

Smart street lighting 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 illumination, or dimmable street lighting that dims at pre-determined times because of its dynamic movement dependent behavior.

Thus, it is expected to save a lot of electricity cost, maintenance cost and accidental cost if implemented properly.

1.1.1 Need of Smart Street Lighting System

Street lighting attributes nearly 19% of the worldwide use of electrical energy and entails 6% of global emissions of greenhouse gases. A decrease of 40% of energy spent for lighting purposes is made possible with the use of LED Lamps instead of any other light source. This is equivalent to eliminating a half of the emissions from the production of electricity and heat generation. This plays an important role, as it impacts for around 40% on the cities' energy budget.

A further decrease in consumption is expected by the successful implementation of the smart street lights which change their glowing intensities as per the movement on the roads. This kind of system can lead to 20-30% power saving and 5-year return on investment.

1.1.2 Features of Smart Street Lighting System

- Integral part of "Smart Cities Solutions".
- Huge Energy savings upto 30-40%.
- Excellent central operations and maintenance of street lights.
- Less Labor cost.
- Real time fault detection and Smart communication.

- Reduced Carbon Emission.
- Decrease Unnecessary Light Pollution.
- Long Lifetime of LEDs or Bulbs.

1.2 Internet of Things

1.2.1 What is IoT?

The **Internet of things** (**IoT**) is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

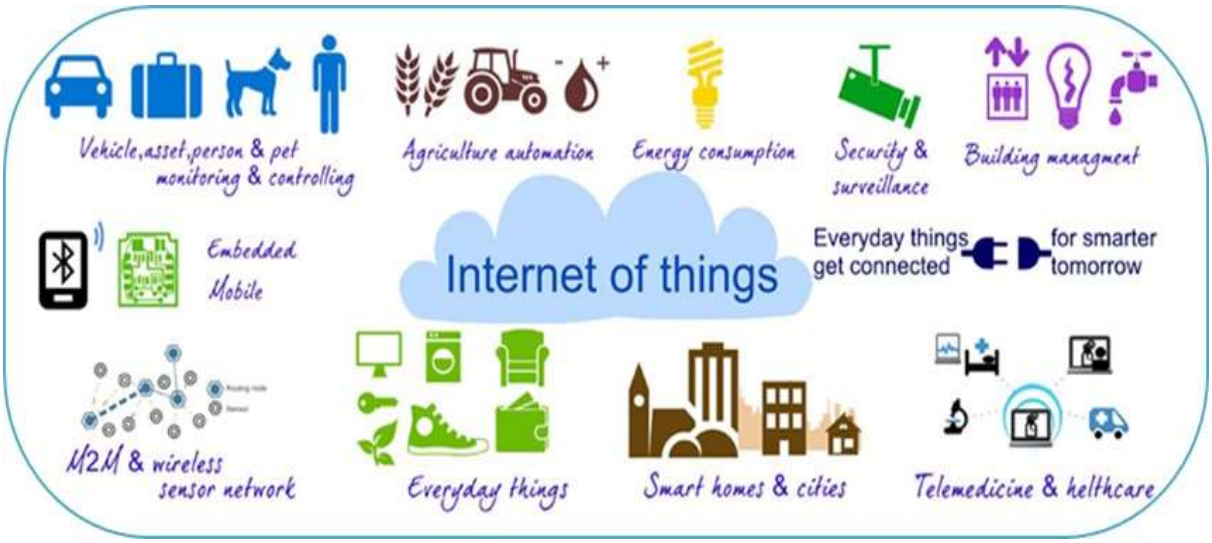
The definition of the Internet of things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smart phones and smart speakers.

The term Internet of Things is 16 years old. But the actual idea of connected devices had been around longer, at least since the 70s. Back then, the idea was often called “embedded internet” or “pervasive computing”. But the actual term

"Internet of Things" was coined by Kevin Ashton in 1999 during his work at Procter&Gamble. Ashton who was working in supply chain optimization, wanted to attract senior management's attention to a new exciting technology called RFID. Because the internet was the hottest new trend in 1999 and because it somehow made sense, he called his presentation "Internet of Things".

1.2.2 Applications of IoT

The extensive set of applications for IoT devices is often divided into consumer, commercial, industrial, and infrastructure spaces.



1. Consumer Applications

A growing portion of IoT devices are created for consumer use, including connected vehicles, home automation, wearable technology (as part of Internet of Wearable Things (IoWT),connected health, and appliances with remote monitoring capabilities.

2. Smart Homes

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems. Long-term benefits could include energy savings by automatically ensuring lights and electronics are turned off.

3. Medical and Healthcare

The Internet of Medical Things (also called the internet of health things) is an application of the IoT for medical and health related purposes, data collection and analysis for research, and monitoring. This 'Smart Healthcare' as it is also called, led to the creation of a digitized healthcare system, connecting available medical resources and healthcare services.

IoT devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fitbit electronic wristbands, or advanced hearing aids. Some hospitals have begun implementing "smart beds" that can detect when they are occupied and when a patient is attempting to get up. It can also adjust itself to ensure appropriate pressure and support is applied to the patient without the manual interaction of nurses.

4. Transportation

The IoT can assist in the integration of communications, control, and information processing across various transportation systems. Application of the IoT extends to all aspects of transportation systems (i.e. the vehicle, the infrastructure, and the driver or user). Dynamic interaction between these

components of a transport system enables inter- and intra-vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistics and fleet management, vehicle control, safety, and road assistance.

5. Building and Home Automation

IoT devices can be used to monitor and control the mechanical, electrical and electronic systems used in various types of buildings (e.g., public and private, industrial, institutions, or residential) in home automation and building automation systems. In this context, three main areas are being covered in literature:

- The integration of the Internet with building energy management systems in order to create energy efficient and IOT-driven "smart buildings".
- The possible means of real-time monitoring for reducing energy consumption and monitoring occupant behaviors.
- The integration of smart devices in the built environment and how they might to know how to be used in future applications.

6. Agriculture

There are numerous IoT applications in farming such as collecting data on temperature, rainfall, humidity, wind speed, pest infestation, and soil content. This data can be used to automate farming techniques, take informed decisions to improve quality and quantity, minimize risk and waste, and reduce effort required to manage crops. For example, farmers can now monitor soil temperature and moisture and even apply IoT-acquired data to precision fertilization programs.

7. Infrastructure Applications

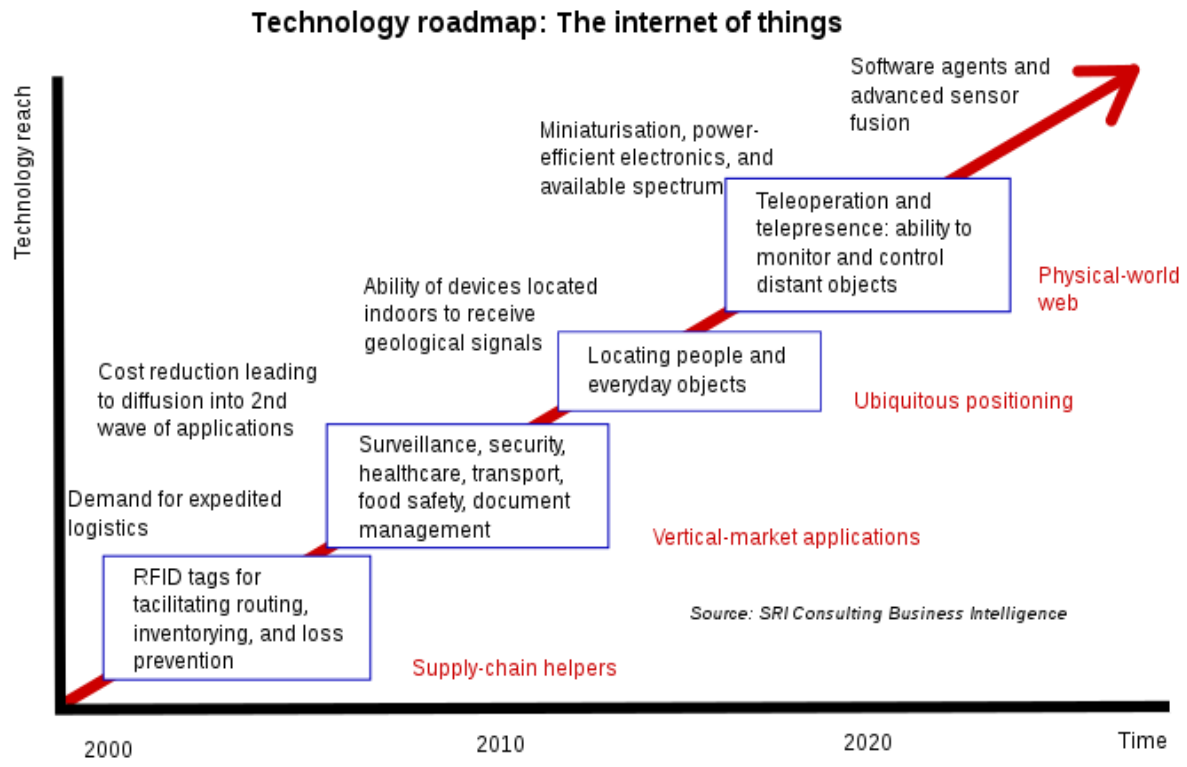
Monitoring and controlling operations of sustainable urban and rural infrastructures like bridges, railway tracks and on- and offshore wind-farms is a key application of the IoT. The IoT infrastructure can be used for monitoring any events or changes in structural conditions that can compromise safety and increase risk. The IoT can benefit the construction industry by cost saving, time reduction, better quality workday, paperless workflow and increase in productivity. It can help in taking faster decisions and save money with Real-Time Data Analytics. It can also be used for scheduling repair and maintenance activities in an efficient manner, by coordinating tasks between different service providers and users of these facilities. IoT devices can also be used to control critical infrastructure like bridges to provide access to ships. Usage of IoT devices for monitoring and operating infrastructure is likely to improve incident management and emergency response coordination, and quality of service, up-times and reduce costs of operation in all infrastructure related areas. Even areas such as waste management can benefit from automation and optimization that could be brought in by the IoT.

1.2.3 Trends and Characteristics

The IoT's major significant trend in recent years is the explosive growth of devices connected and controlled by the Internet. The wide range of applications for IoT technology mean that the specifics can be very different from one device to the next but there are basic characteristics shared by most.

The IoT creates opportunities for more direct integration of the physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions.

The number of IoT devices increased 31% year-over-year to 8.4 billion in the year 2017 and it is estimated that there will be 30 billion devices by 2020. The global market value of IoT is projected to reach $7.1 trillion by 2020.



1.2.4 Importance of IoT

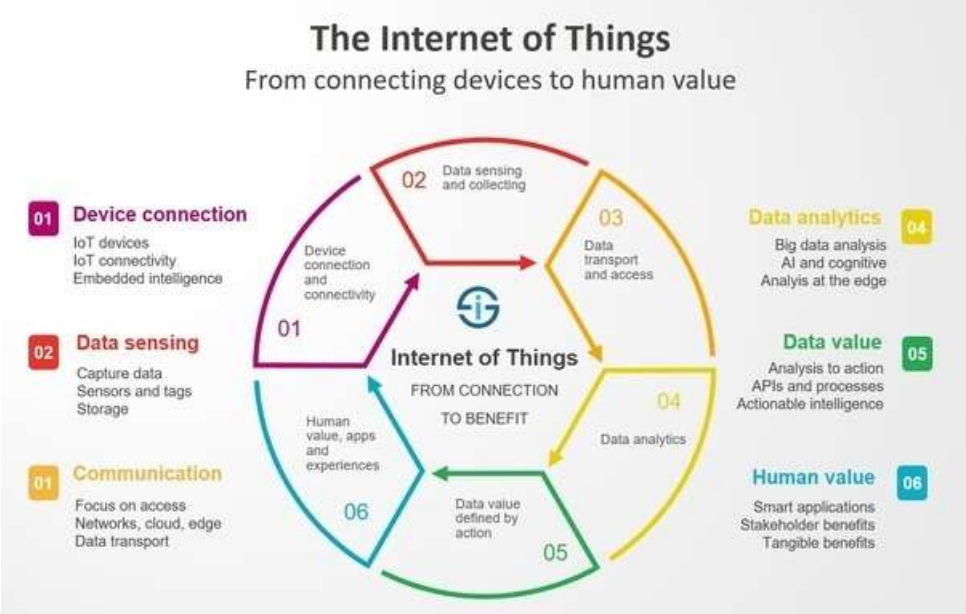
The **"Internet of Things"** (IoT) has the power to change our world. IoT will play an important role in the future and there is expected to be a significant amount of cash flowing through the market in the up-coming years. Over half of major new business processes and systems will incorporate IoT elements by 2020. The impact on consumers' lives and corporate business models is rapidly increasing as the cost of instrumenting physical things with sensors and connecting them to other things devices, systems and people continues to drop. Internet of Things offers some interesting applications to our lives easier like in Healthcare, Transportation, and Agriculture. However, various factors like security, privacy and data storage also need to be considered.

It is also worth noting that things have been connected to networks for ages without the guise of “Internet of Things”.

IoT is the connection of objects (things) to the Internet and these objects can collect and exchange data over the Internet and with each other.

The internet of things (IoT) has made the world more efficient, convenient and enjoyable, with the dramatic surge of internet-connected devices transforming how individuals, households, and businesses interact on a daily basis. A recent study released that 43% of all companies are using or planning to implement an IoT application in 2019.

Internet of Things applications is available in every industry for smart homes, construction, travel and transportation, health care and personal care, retail, agriculture, etc. And in industrial area Internet of Things revolves around automation and logistics and creating smarter solutions, programmatically adjusting to human behavior.



There are still some barriers to adoption is there, for example, battery life of devices and cost of devices. GSM and 4G networks are used more and more for IoT applications and new advancements recently made in the network software

and device stack will greatly improve on these aspects. IoT helps in finding the right ideas and business model, how to go to the market and secure ROI for the Internet of Thingsapplications are the most critical issues to solve for most companies in the coming years.From digital assistants to baby monitors to sensors monitoring traffic on major highways, a plethora of IoT devices connected to networks are helping drive many benefits for users.

1.2.5 Overcoming Threats to IoT

While the benefits of IoT devices is infinite, the internet of things is vulnerable to security threats.

Here is some technique to secure your IoT devices.

1. Secure Device Hardware Accessibility.

2. Secure Port Communication.

3. Use Encryptions.

4. Use Authentication.

5. Ensure Protection Against Phishing and Malware.

6. Employ Middleware Hardware Physical Security.

1.3 Problem Definition

The main objective of this project is to develop a **Smart Street Lighting System which** will overcome the difficulties and drawbacks associated with the traditional street lighting system.

Required Features:

- The system should be capable of turning lights on and off depending upon the sunlight.
- The system should be capable of increasing or reducing the brightness of LEDs depending upon the presence of vehicles/pedestrians on the road.
- The system should be capable of calculating power consumption on its own.
- The system should be capable of sending automatic alerts to the concerned authority as soon as a fault is detected by the system.
- The system should perform detailed analysis of the power consumed.
- The system should be value for money.

2. LITERATURE REVIEW

- Soyoung Hwang et al. [1], have discussed their work on remote monitoring and controlling system based on ZigBee networks. Real-time remote monitoring is implemented with JMF which is a multimedia extension API of Java.
- B. K. Subramanyam et al. [2] in 2013 worked on intelligent wireless street light control and monitoring system. Using solar panel at the lamp post and LDR it is possible to save some more power and energy, and also we can monitor and control the street lights using GUI application. But the implementation results were not focused.
- P. Nithya et al. [3], have focused on centralization of data sent by each node i.e. traffic light in their work 'Design of Wireless Framework for Energy Efficient Street Light Automation'. With the suggested system, maintenance can be easily and efficiently planned from the central station, allowing additional savings.
- R. Kavitha1 et al. [4], suggest that necessary controlling actions could be taken depending on the amount of traffic in a particular direction. Also emergency vehicles and VIP convoys can be passed efficiently. Moreover, attempts can be made to ensure that the complete system is self-sufficient.
- Anila Devi Y et al. [5], worked on GSM Based Remote Control System of High Efficiency Intelligent Street Lighting System using AZigbee Network of Devices and Sensor. New intelligent and smart street light system is designed with wireless technology for maintenance and network of sensors for controlling. They used high efficiency LED lamp which consumes less energy with high life time and is supplied with renewable energy of solar panels.
- Richu Sam Alex et al. [6], worked on designing a fully automated ZIGBEE control station that can analyze all the performance of the system. This

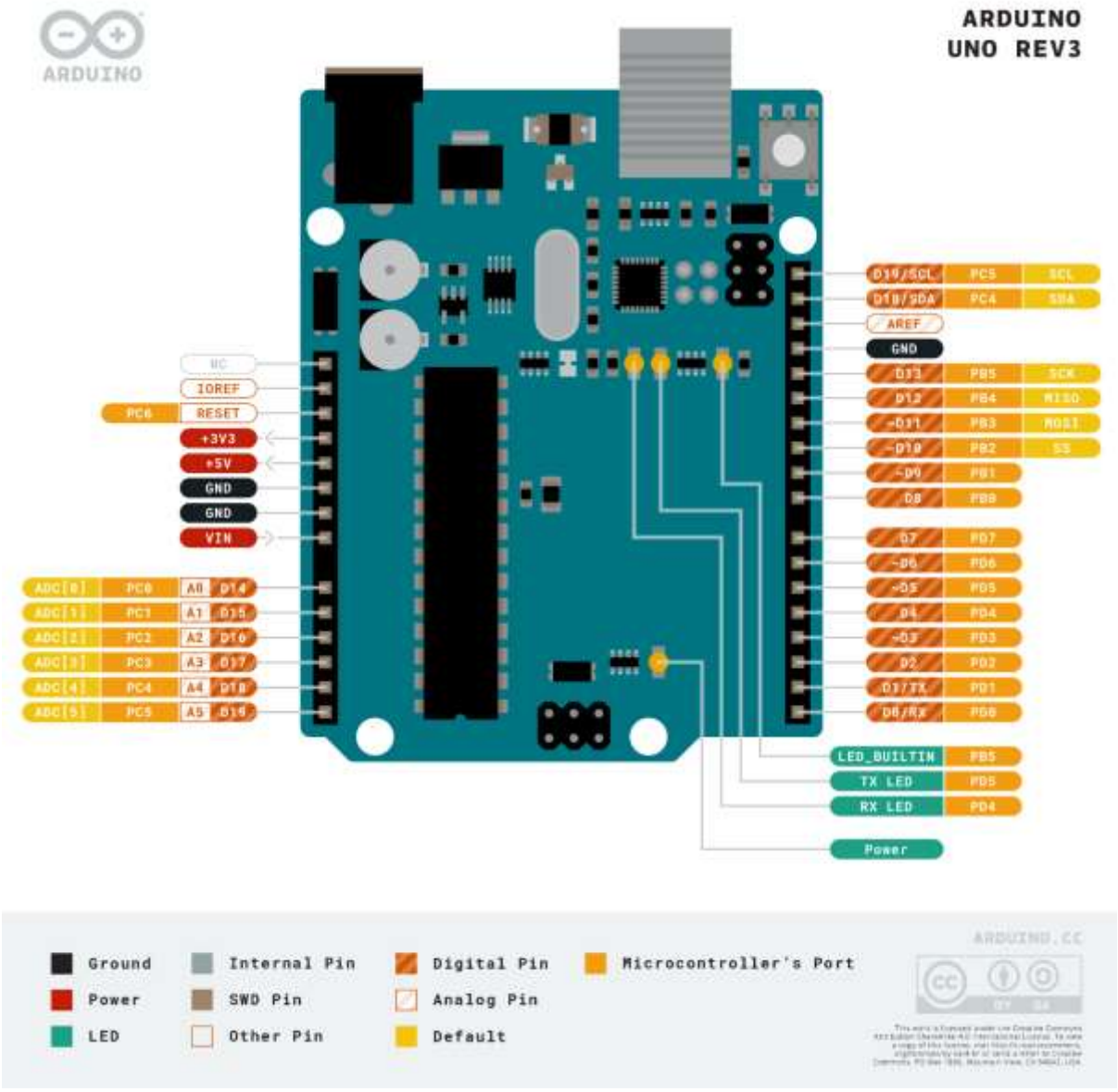
reduces power consumption of the street lighting system by about 20-35 % compared to conventional design.

- Srikanth M et al. [7], in their work on ZigBee Based Remote Control Automatic Street Light System mentioned about the streetlight control system that helps in energy savings, detection of faulty lights and maintenance time and increase in life span of system.

3. SYSTEM REQUIREMENTS

3.1 Hardware Requirements:

3.1.1 Arduino Uno-The Arduino Uno is an open-source micro-controller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

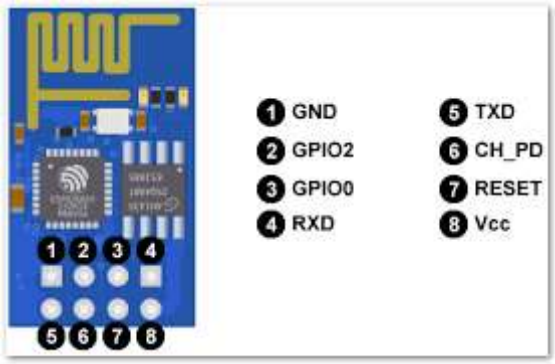


The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts.

Specifications:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13

3.1.2 ESP8266 WiFi Module-The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability, produced by Espressif Systems in Shanghai, China.

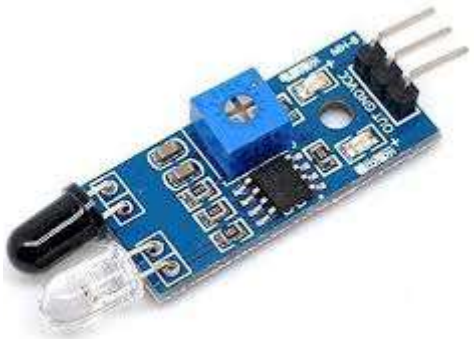


Specifications:

- 32-bit RISC CPU: TensilicaXtensa LX106 running at 80 MHz.
- 64 KiB of instruction RAM, 96 KiB of data RAM.
- External QSPI flash – 512 KiB to 4 MiB (up to 16 MiB is supported).
- IEEE 802.11 b/g/n Wi-Fi.
- Integrated TR switch, balun, LNA, power amplifier and matching network.
- WEP or WPA/WPA2 authentication, or open networks.
- 16 GPIO pins.
- SPI, I^2C.
- I^2S interfaces with DMA (sharing pins with GPIO).
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2.
- One 10-bit ADC.

3.1.3 IR Sensor Module-Infrared Sensor is an electronic device, that detects the infrared radiations to sense the aspects of its surroundings. It detects motions that measures the IR light from objects in the field of view. The IR

sensor receiver is a three terminal device used to decrease the size of circuit, which consists of three terminals, VCC, ground and output signal. When any obstacle detected, the sensor receives the signal and sends it to controller.



Specifications:

- 5VDC Operating voltage.
- I/O pins are 5V and 3.3V compliant.
- Range: Up to 20cm.
- Adjustable Sensing Range.
- Built-in Ambient Light Sensor.
- 20mA Supply Current.

3.1.4 LDR-A photoresistor or light-dependent resistor, or photo-conductive cell is an active component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface. The resistance of a photoresistor decreases with increase in incident light intensity; in other words, it exhibits photoconductivity.

3.1.5 5V Relay-A relay is an electrically operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal.

Specifications:

- Trigger Voltage (Voltage across coil) : 5V DC.
- Trigger Current (Nominal current) : 70mA.
- Maximum AC load current: 10A @ 250/125V AC.
- Maximum DC load current: 10A @ 30/28V DC.
- Compact 5-pin configuration with plastic moulding.
- Operating time: 10msec Release time: 5msec.
- Maximum switching: 300 operating/minute (mechanically).

3.1.6 LED- A light-emitting diode is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor.

3.1.7 Battery (9V)-For Powering Arduino UNO Board.

3.1.8 Jumper Wires-Jumper wires are simple wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires.

3.1.9 Computer System- A computer system of following specifications will be required at the control room for monitoring and maintenance purposes:

- Minimum 4GB of RAM .
- A Hard Disk Drive (HDD) of about 500GB.
- Intel Core i3 or higher processor.
- Windows XP or any other higher OS installed.

3.2 Software Requirements:

3.2.1 Arduino IDE-The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards.

3.2.2 Cloud Support- There are numerous cloud service providers available in the market such as Amazon Web Service (AWS), Microsoft Azure, Google Cloud Platform, IBM Cloud Services. We for our project work opted for ThingSpeak. It is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network. It has integrated support from the numerical computing software MATLAB from MathWorks, allowing its users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks.

3.2.3 Internet Connection-A working Wireless Access Point (AP) is required for Internet Connectivity. Internet is required for communication between micro-controller and cloud-based sensor log channel.

3.2.4 Python Development Environment-It will be required for running python programs. The python program will be responsible for sending the message and e-mail alert notifications.

3.2.4 Twilio- It is a popular third-party SMS functionality provider. Twilio allows software developers to programmatically make and send and receive text messages using its web service APIs.

3.2.5 Mailgun- is an Email automation service. It has a very powerful set of inbuilt API functions for sending emails. Developers can send emails programmatically with the help of Mailgun API.

4. DIFFIRENT PHASES OF THE PROJECT

Due to the sudden outbreak of COVID-19 in our country, we are left with no hardware required for implementing a model of the project. So, we had tried to simulate the circuit on an online simulation facility TinkerCAD. With the help of TinkerCAD, we were able to make the required circuit connections and simulate it. From this point onwards, the discussion of project work will only be related to simulation of the system.

Before we move to the simulation part, let us discuss the approach followed by us:

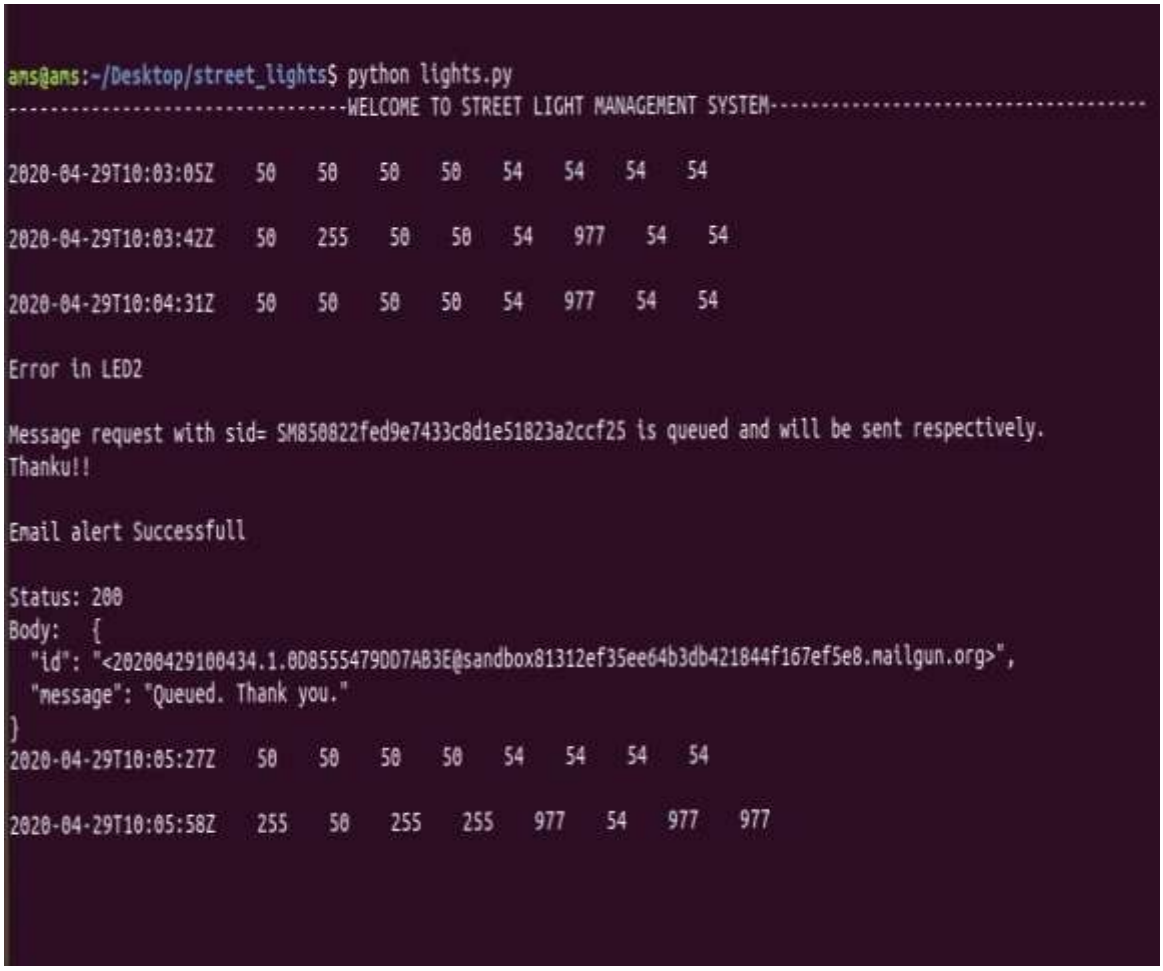
1. Power Up- The micro-controller will check for the need of lighting through the intensity of light incident on LDR. If the natural light incident on LDR is low, the micro-controller will light up the lamp. Thus, acting like an automatic switch. Initially, all the lamps will glow with some minimum brightness.

2. Detecting the Objects- The IR sensors will look for an object present in their view fields and correspondingly send the signal to micro-controller. The micro-controller will then increase the brightness of that particular lamp from which the signal was received.

3. The voltage levels of various lamps assigned by the micro-controller will be sent to the ThingSpeak cloud channel at regular intervals for further analysis.

4. Also the values from various LDRs, installed below the lamp will be send to cloud channel for detecting faults in the lamps if any.

5.The computer system at control unit will retrieve the value of voltage assigned to each lamp and LDR sensor values from the cloud channel and will compare them for the detection of faults. If there is a mismatch, an SMS and e-mail notification will be sent to the concerned authority along with the location of defected lamp. The location being retrieved from the stored database.

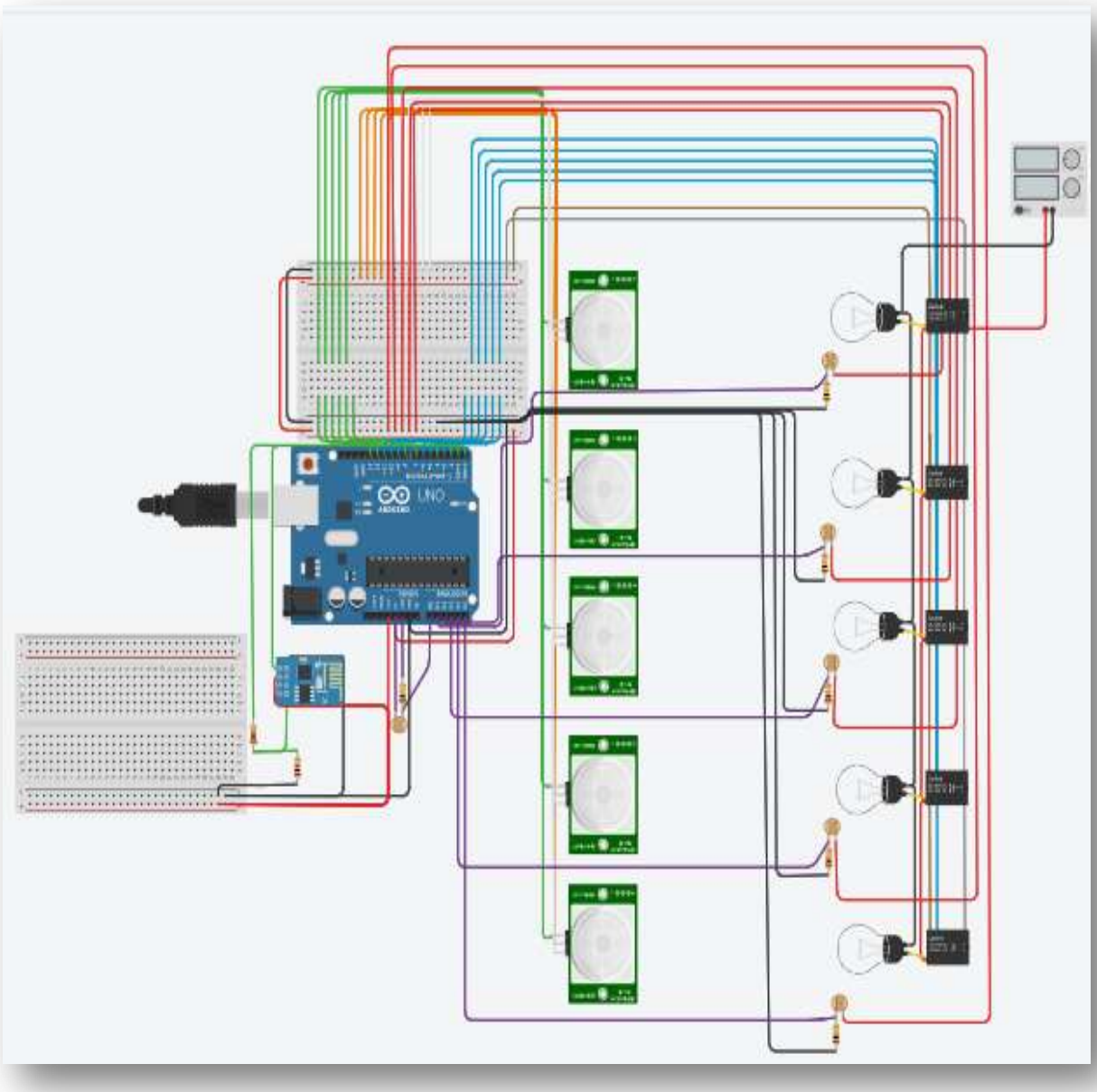


6.The power consumption plots will be provided using python program from which power savings can be calculated compared to traditional lighting system.

Now we will proceed with the **simulation part**.

4.1 Setting up the Circuit

The online simulation facility TinkerCAD provides its users with all the necessary components, sensors required for development of an IoT project. The following circuit was created onTinkerCAD.



4.2 Coding for the Required Operation of the System

The TinkerCAD provides online code editor where users can write and run their codes. The simulator provides following Arduino libraries that users can include in their codes:

EEPROM-Reading and writing to "permanent" storage.

IRremote-Library to decode IR sensors.

LiquidCrystal-Controlling liquid crystal displays (LCDs).

Keypad-Allows reading keypad button pushes.

NeoPixel-Controlling NeoPixel LEDs.

Servo-Controlling servo motors.

SoftwareSerial-Allow serial communication on other digital pins of the Arduino.

Wire-This library allows you to communicate with I2C / TWI devices.

SD-The SD library allows for reading from and writing to SD cards.

SPI-Communicating with devices using the Serial Peripheral Interface (SPI) Bus.

Stepper-Controlling stepper motors.

❖ We developed the following piece of **Arduino Code** as per our requirement:

```
#include<SoftwareSerial.h>
// WiFi Configuration
String ssid     = "Simulator Wifi";
String password = "";

// Server, File, and Port
const String hostname = "api.thingspeak.com";
const int port = 80;

void setup()
{
        Serial.begin(115200);
        Serial.flush();

         //Connecting to WiFi
        Serial.println("Connecting to WiFi");
        Serial.print("AT+CWJAP=\""+ssid+"\",\""+password+"\"\r\n");
        delay(50);

        pinMode(A0, INPUT);          pinMode(A1, INPUT);
        pinMode(A2, INPUT);          pinMode(A3, INPUT);
        pinMode(A4, INPUT);          pinMode(A5, INPUT);
        pinMode(4, INPUT);           pinMode(7, INPUT);
        pinMode(8, INPUT);           pinMode(12, INPUT);
        pinMode(13, INPUT);          pinMode(3, OUTPUT);
        pinMode(5, OUTPUT);          pinMode(6, OUTPUT);
        pinMode(9, OUTPUT);          pinMode(10, OUTPUT);
}

void loop()
{
         //Open TCP Connection
        Serial.println("AT+CIPSTART=\"TCP\",\"" + hostname + "\"," + port);
```

```
delay(50);

Serial.println(analogRead(A0));
  int led1,led2,led3,led4,led5;

  int ldr1,ldr2,ldr3,ldr4,ldr5;
delay(10);
 if(analogRead(A0)>300)
 {
        Serial.println("All Lights Switched OFF");
        analogWrite(3,0);
        analogWrite(5,0);
        analogWrite(6,0);
        analogWrite(9,0);
        analogWrite(10,0);
 }
 else
 {
          if(digitalRead(13)==1)
          {
              analogWrite( 3,255);
                  led1=255;
          }
          else
          {
              analogWrite( 3,50);
                  led1=50;
          }

          if(digitalRead(12)==1)
          {
              analogWrite( 5,255);
                  led2=255;
          }
          else
          {
```

```
        analogWrite( 5,50);
            led2=50;
    }

    if(digitalRead(8)==1)
    {
        analogWrite( 6,255);
            led3=255;
    }
    else
{
        analogWrite( 6,50);
        led3=50;
    }

    if(digitalRead(7)==1)
    {
        analogWrite( 9,255);
            led4=255;
    }
    else
    {
        analogWrite( 9,50);
           led4=50;
    }

    if(digitalRead(4)==1)
    {
        analogWrite( 10,255);
            led5=255;
    }
    else
    {
        analogWrite( 10,50);
           led5=50;
    }
```

```
}

// POST ***TEST***
String body = "api_key=OR07EPCMF3X9F6H2";
body+="&field1=";
body+=String(led1);
body+="&field2=";
body+=String(led2);
body+="&field3=";
body+=String(led3);
body+="&field4=";
body+=String(led4);

//LDR values to check leds working
ldr1=analogRead(A1);
ldr2=analogRead(A2);
ldr3=analogRead(A3);
ldr4=analogRead(A4);
ldr5=analogRead(A5);

body+="&field5=";
body+=String(ldr1);
body+="&field6=";
body+=String(ldr2);
body+="&field7=";
body+=String(ldr3);
body+="&field8=";
body+=String(ldr4);


Serial.println("LDR1 ");
Serial.println(ldr1);
Serial.println("LDR2 ");
Serial.println(ldr2);
Serial.println("\n");
```

```
 String req =       String("POST /update HTTP/1.1\r\n") +
                                "Host: " + hostname + "\r\n" +
                                "User-Agent: Arduino/1.0\r\n" +
                                "Content-Type: application/x-www-form-urlencoded;\r\n" +
                                "Content-Length: " + body.length() + "\r\n" +
                                "Connection: close\r\n" +
                                "\r\n" +
                                body + "\r\n";
 int len= req.length();

 // Send our request length
Serial.print("AT+CIPSEND=");
Serial.println(len);
delay(5);

 if ( Serial.find(">") == 0 )
 {
        Serial.println("Error");
 }

 // Send our http request
Serial.print(req);
 if (!Serial.find("SEND OK\r\n"))
{
        Serial.println("Error");
 }

 // Close TCP connection
Serial.println("AT+CIPCLOSE=0");
delay(5);

delay(1000);
}
```

❖ Backend **Python Script** which will perform the analysis part:

```
from urllib.request import urlopen

import json;

import time;

from twilio.rest import Client;

import requests;

import numpy as np;

import matplotlib.pyplot as plt;


#Twilio credentials

account_sid = 'your id';

auth_token = 'your token';

#Mailgun credentials

key =’your key’;

sandbox = 'your id’;

sender='your id’;

recipient = 'recipient’s id';


#Read API key of channel for thingspeak

READ_API_KEY='IM1PT4V63L9IYFG0';

CHANNEL_ID= '1045786';#channel id

prev_entry_id=0;


#Counter to count number of data reads

i=0;

#Power arrays for different LED

pow1=np.arange(5).astype(float);pow2=np.arange(5).astype(float);pow3=np.arange(5).astype(float);pow4=np.arange(5).astype(float);
```

```
high=100*.003;

low=60*0.003;

x = range(5);

print("----------------------------------WELCOME TO STREET LIGHT MANAGEMENT SYSTEM----------
----------\n")

while True:

   TS = urlopen("http://api.thingspeak.com/channels/%s/feeds/last.json?api_key=%s" \

                  % (CHANNEL_ID,READ_API_KEY));


   #Reading data from cloud

   response = TS.read();

   data=json.loads(response);


   #Checking entry_id of data

   entry_id=data['entry_id'];

   #Debugging purpose

   if i==5:

         break;

   if entry_id==prev_entry_id:

         continue;

   prev_entry_id=entry_id;


   #Picking up the required data from the recieved data

   create_time = data['created_at'];

   ir1 = data['field1'];

   ir2 = data['field2'];

   ir3 = data['field3'];

   ir4 = data['field4'];
```

```
ldr1 = data['field5'];

ldr2 = data['field6'];

ldr3 = data['field7'];

ldr4 = data['field8'];

print (create_time + "    " + ir1 + "    " + ir2 + "    " + ir3+ "    " + ir4 + "    " + ldr1 + "    " + ldr2+ "    " + ldr3+ "    " + ldr4);


#Checking working of led to send alert messages if required

#LED 1

if (ir1=="255" and ldr1<"900") or (ir1=="50" and ldr1>"900"):

        print("Error in LED1\n");

        client = Client(account_sid, auth_token);


        #Sending error message sms

        message = client.messages.create(body='LED1 is not working properly.\n Location: Street No:1, Type 4, RCF Colony, Kapurthala.',from_='+12056277345',to='+918195982962');


        #Sending error message sms

        if message.status=="queued":

                print("Message request with sid= "+message.sid+" is queued and will be sent respectively."+"\nThanku!!\n");

        else:

                print("Error in sending SMS\n");


        #Sending error message mail

        request_url = 'https://api.mailgun.net/v2/{0}/messages'.format(sandbox);

        request = requests.post(request_url, auth=('api', key), data={

                'from': sender,

                'to': recipient,
```

```
                'subject': 'Error',
                'text': 'LED1 is not working properly.\n Location: Street No:1, Type 4, RCF Colony, Kapurthala.'
                 });
        if request.status_code==200:
                print("Email alert Successfull\n");
        else:
                print("Error in sending Email.\n ");
        print('Status: {0}'.format(request.status_code));
        print('Body:   {0}'.format(request.text));
    elif ir1=="255":
        pow1[i]=high;
    elif ir1=="50":
        pow1[i]=low;
    #LED 2
    if (ir2=="255" and ldr2<"900") or (ir2=="50" and ldr2>"900"):
        print("Error in LED2\n");
        client = Client(account_sid, auth_token);

#Sending error message
        message = client.messages.create(
            body='LED2 is not working properly.\n Location: Street No:2, Type 4, RCF Colony, Kapurthala.',
            from_='+12056277345',
            to='+918195982962'
            );
        if message.status=="queued":
                print("Message request with sid= "+message.sid+" is queued and will be sent respectively."+"\nThanku!!\n");
```

```
        else:
                print("Error in sending SMS\n");
```

#Sending error message mail

```
        request_url = 'https://api.mailgun.net/v2/{0}/messages'.format(sandbox)
        request = requests.post(request_url, auth=('api', key), data={
                'from': sender,
                'to': recipient,
                'subject': 'Error',
                'text': 'LED2 is not working properly.\n Location: Street No:2, Type 4, RCF Colony, Kapurthala.'
                 })
        if request.status_code==200:
                print("Email alert Successfull\n");
        else:
                print("Error in sending Email.\n ");
        print('Status: {0}'.format(request.status_code));
        print('Body:   {0}'.format(request.text));

   elif ir2=="255":
        pow2[i]=high;
   elif ir2=="50":
        pow2[i]=low;
```

#LED 3

```
   if (ir3=="255" and ldr3<"900") or (ir3=="50" and ldr3>"900"):
        print("Error in LED3\n");
        client = Client(account_sid, auth_token);
```

```
        #Sending error message
        message = client.messages.create(
            body='LED3 is not working properly.\n Location: Street No:3, Type 4, RCF Colony, Kapurthala.',
            from_='+12056277345',
            to='+918195982962'
            );
        if message.status=="queued":
                print("Message request with sid= "+message.sid+" is queued and will be sent respectively."+"\nThanku!!\n");
        else:
                print("Error in sending SMS\n");

        #Sending error message mail
        request_url = 'https://api.mailgun.net/v2/{0}/messages'.format(sandbox);
        request = requests.post(request_url, auth=('api', key), data={
                'from': sender,
                'to': recipient,
                'subject': 'Error',
                'text': 'LED3 is not working properly.\n Location: Street No:3, Type 4, RCF Colony, Kapurthala.'
                 });
        if request.status_code==200:
                print("Email alert Successfull\n");
        else:
                print("Error in sending Email.\n ");
        print('Status: {0}'.format(request.status_code));
        print('Body:   {0}'.format(request.text));
    elif ir3=="255":
```

```
        pow3[i]=high;
    elif ir3=="50":
        pow3[i]=low;

    #LED 4
    if (ir4=="255" and ldr4<"900") or (ir4=="50" and ldr4>"900"):
        print("Error in LED4\n");
        client = Client(account_sid, auth_token);

#Sending error message
 message = client.messages.create(
 body='LED4 is not working properly.\n Location: Street No:4, Type 4, RCF Colony, Kapurthala.',
            from_='+12056277345',
            to='+918195982962'
            );
        if message.status=="queued":
            print("Message request with sid= "+message.sid+" is queued and will be sent respectively."+"\nThanku!!\n");
        else:
            print("Error in sending SMS\n");

        #Sending error message mail
        request_url = 'https://api.mailgun.net/v2/{0}/messages'.format(sandbox);
        request = requests.post(request_url, auth=('api', key), data={
            'from': sender,
            'to': recipient,
            'subject': 'Error',
```

```
                'text': 'LED4 is not working properly.\n Location: Street No:4, Type 4, RCF Colony, Kapurthala.'
                 });
        if request.status_code==200:
                print("Email alert Successfull\n");
        else:
                print("Error in sending Email.\n ");
        print('Status: {0}'.format(request.status_code));
        print('Body:   {0}'.format(request.text));
   elif ir4=="255":
        pow4[i]=high;
   elif ir4=="50":
        pow4[i]=low;
   i+=1;
   time.sleep(5);
   TS.close();
```

#Total power variables for each LED 10sec=.003hr

```
power1=0;power2=0;power3=0;power4=0;
```

#Total number of consumed units with old system for 5 data reads

```
total_units_with_old_method=100*.003*5;
for i in range (1,5):
        power1+=pow1[i];
for i in range (1,5):
        power2+=pow2[i];
print("\n");
for i in range (1,5):
```

```
        power3+=pow3[i];
print("\n");
for i in range (1,5):
        power4+=pow4[i];
```

#Graphical representation of power consumption

```
fig = plt.figure();
plt.subplot(2, 2, 1);
plt.plot(x, pow1);
plt.title('POWER CONSUMPTION BY LED1');
plt.ylabel('Power Units Consumed');
plt.subplot(2, 2, 2);
plt.plot(x, pow2);
plt.title('POWER CONSUMPTION BY LED2');
plt.ylabel('Power Units Consumed');
plt.subplot(2, 2, 3);
plt.plot(x, pow3);
plt.title('POWER CONSUMPTION BY LED3');
plt.ylabel('Power Units Consumed');
plt.subplot(2, 2, 4);
plt.plot(x, pow4);
plt.title('POWER CONSUMPTION BY LED4');
plt.ylabel('Power Units Consumed');
plt.show();
```

#Bar graph representation for comparison of power consumed

```
fig = plt.figure();
plt.subplot(2, 2, 1);
```

```
plt.bar([1],total_units_with_old_method,align='center',color='red',width=0.5,label="With Conventional Method",log=True);

plt.bar([3],power1,align='center',color='green',width=0.5,label="With Smart Street Lightning Implementation",log=True);

plt.title('POWER CONSUMPTION COMPARISON OF LED1');

plt.ylabel('Power Units Consumed');

plt.legend(prop={'size':10});

plt.subplot(2, 2, 2);

plt.bar([1],total_units_with_old_method,align='center',color='red',width=0.5,label="With Conventional Method",log=True);

plt.bar([3],power2,align='center',color='green',width=0.5,label="With Smart Street Lightning Implementation",log=True);

plt.title('POWER CONSUMPTION COMPARISON OF LED2');

plt.ylabel('Power Units Consumed');

plt.legend(prop={'size':10});

plt.subplot(2, 2, 3);

plt.bar([1],total_units_with_old_method,align='center',color='red',width=.5,label="With Conventional Method",log=True);

plt.bar([3],power3,align='center',color='green',width=0.5,label="With Smart Street Lightning Implementation",log=True);

plt.title('POWER CONSUMPTION COMPARISON OF LED3');

plt.ylabel('Power Units Consumed');

plt.legend(prop={'size':10});

plt.subplot(2, 2, 4);

plt.bar([1],total_units_with_old_method,align='center',color='red',width=.5,label="With Conventional Method",log=True);

plt.bar([3],power4,align='center',color='green',width=0.5,label="With Smart Street Lightning Implementation",log=True);

plt.title('POWER CONSUMPTION COMPARISON OF LED4');

plt.ylabel('Power Units Consumed');
```

```
plt.legend(prop={'size':10});

plt.show();
```

#Bar graph representation for comparison of total power consumed by whle system combined

```
sum_of_total_units_with_old_method=total_units_with_old_method*4;

totalsum=power1+power2+power3+power4;
```

#Percentage power saved

```
saved=(sum_of_total_units_with_old_method-totalsum)/sum_of_total_units_with_old_method;

saved=saved*100;

save_str=str(saved);

x_pos = 3;

y_pos = .1;

str_print="Percentage Decrease In Power Consumtion : "+save_str+"%.";

fig = plt.figure();

plt.bar([1],sum_of_total_units_with_old_method,align='center',color='red',width=0.5,label="With Conventional Method",log=True);

plt.bar([3],totalsum,align='center',color='green',width=0.5,label="With Smart Street Lightning Implementation\n"+ str_print,log=True);

plt.title('POWER CONSUMPTION COMPARISON OF THE WHOLE SYSTEM');

plt.ylabel('Power Units Consumed');

plt.legend(prop={'size':10});

plt.text(x_pos, y_pos,str_print);

plt.show();
```

4.3 Simulation of the System

The following things were observed after simulating our system:

- The sensor data was successfully logged to ThingSpeak cloud channel.
- The LEDs worked fine for some initial time but started misbehaving after a certain period of time.
- The reason behind misbehaving LEDs was found to be the increased response time of the system as the number of components in the circuit increases.
- The sensor data was successfully fetched from the cloud channel by python program.
- The python program was able to judge any defect and the corresponding alert notification was sent successfully on desired mobile number as well as email.

4.4 Results of the Simulation

❖ Cloud Data

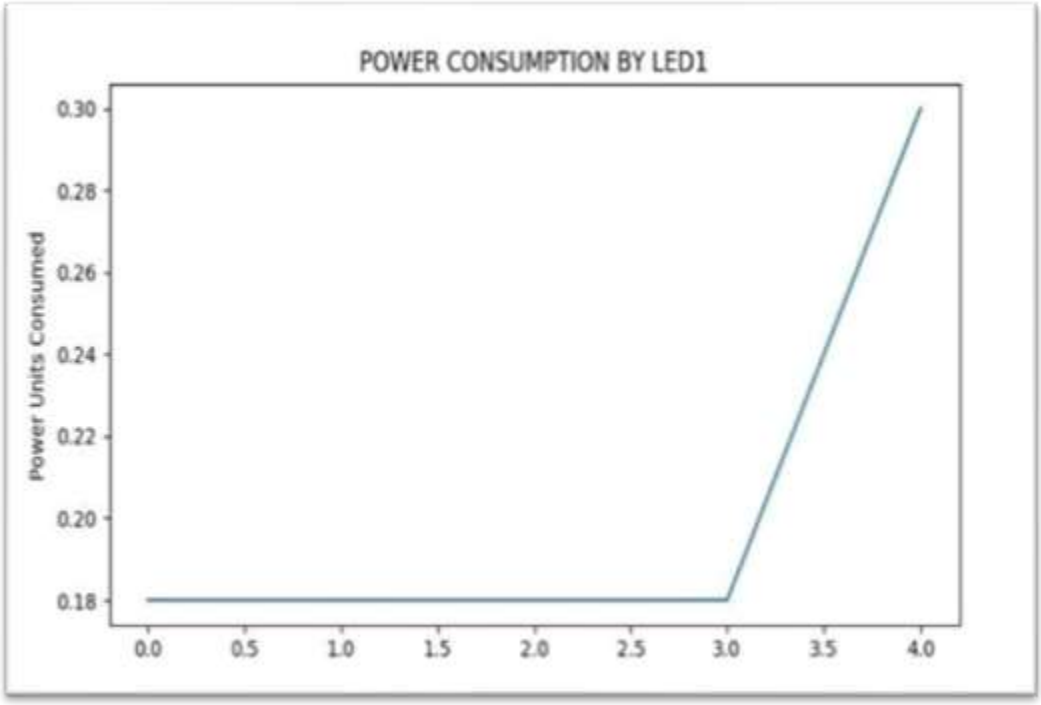
- **Screenshots showing the voltage value assigned to each LED depending upon the presence of vehicles by micro-controller. and the value from corresponding LDR.**

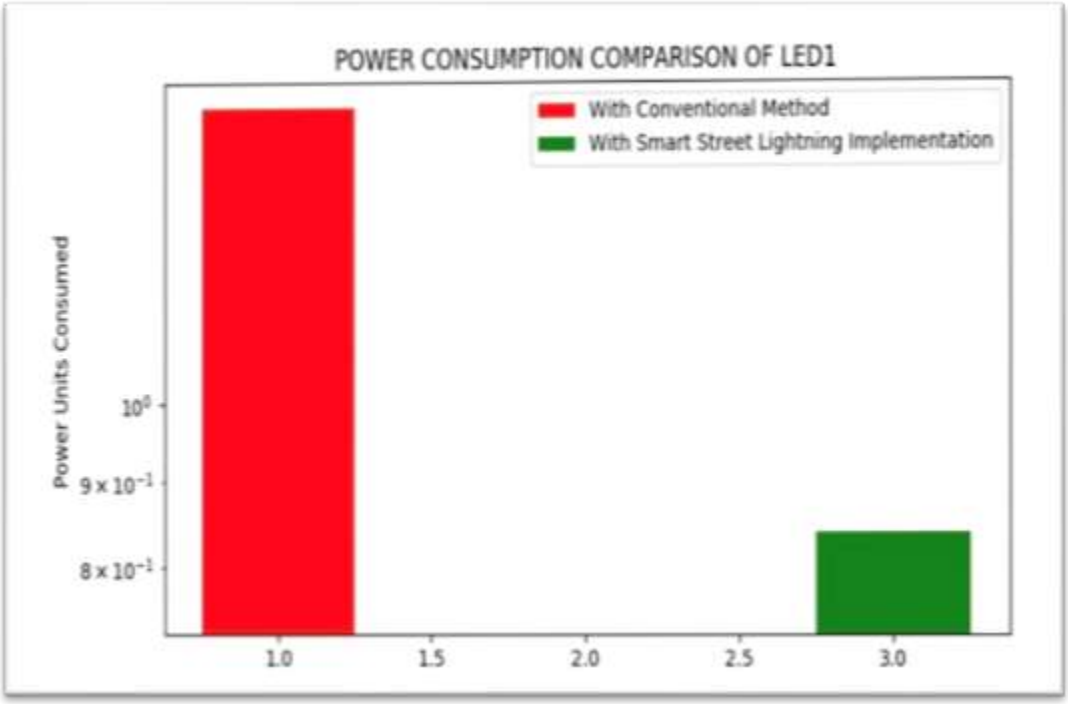
❖ Cloud Data

- **Screenshots showing the value from corresponding LDR.**

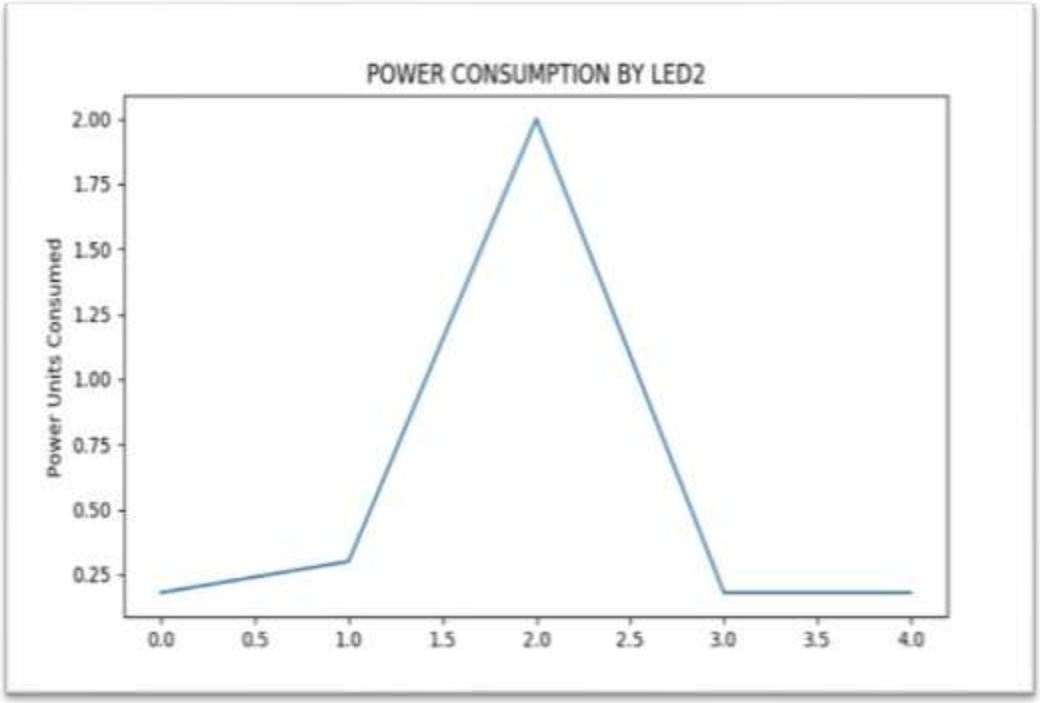
❖ Analysis Results

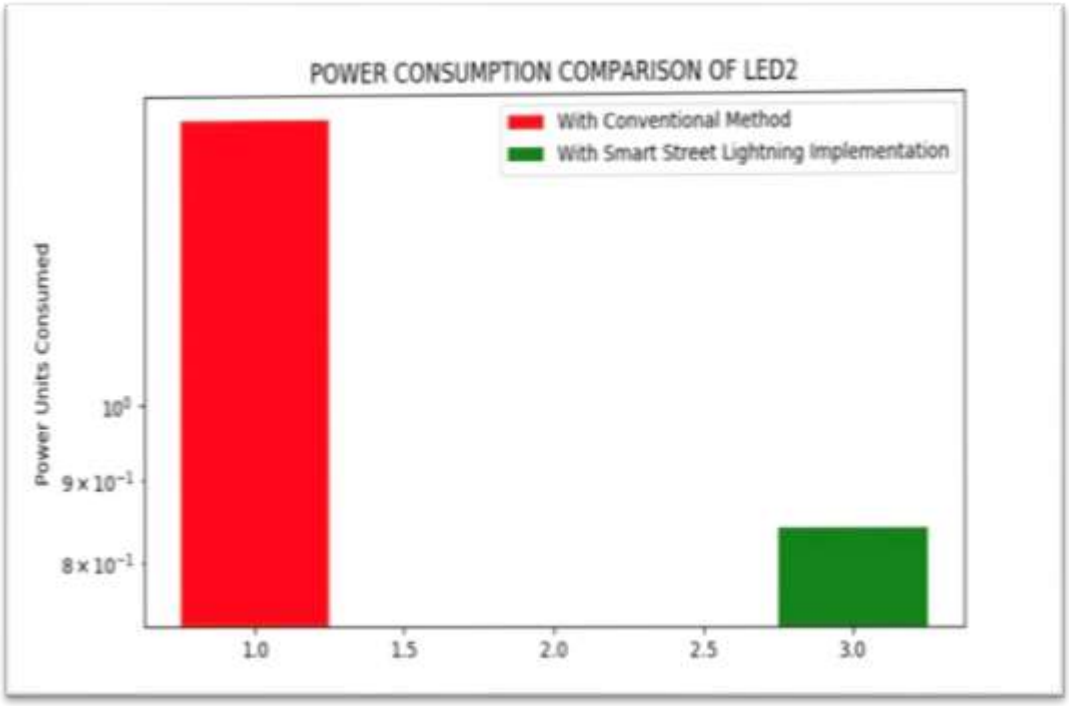
- **Screenshots showing power consumed by LED1 and corresponding consumption compared to the old system.**



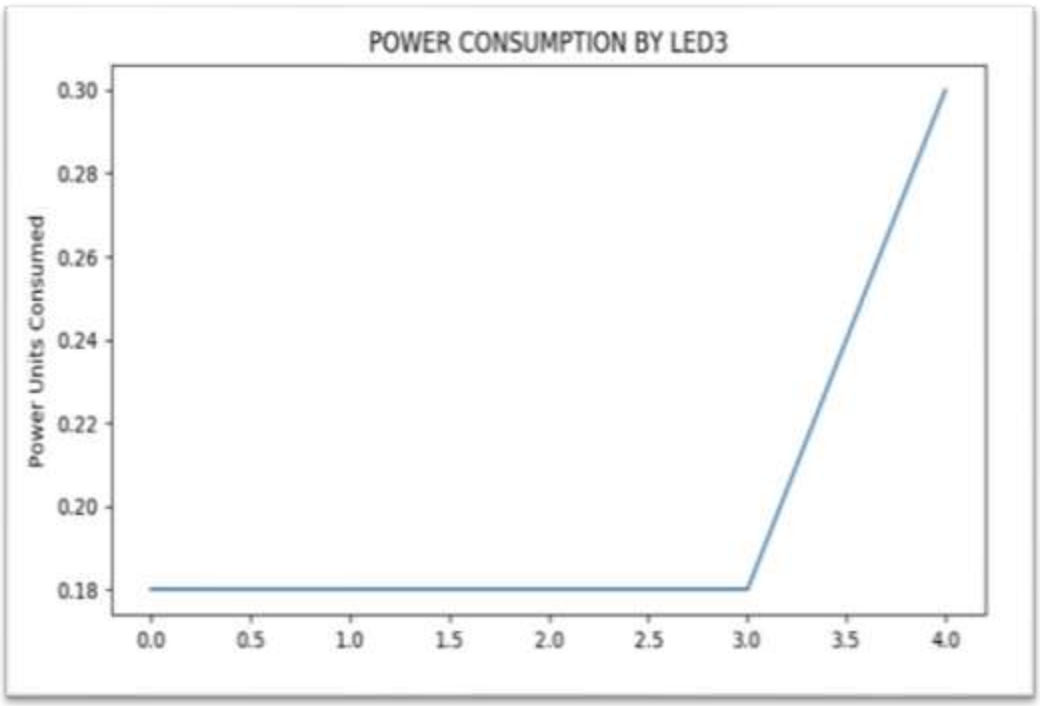


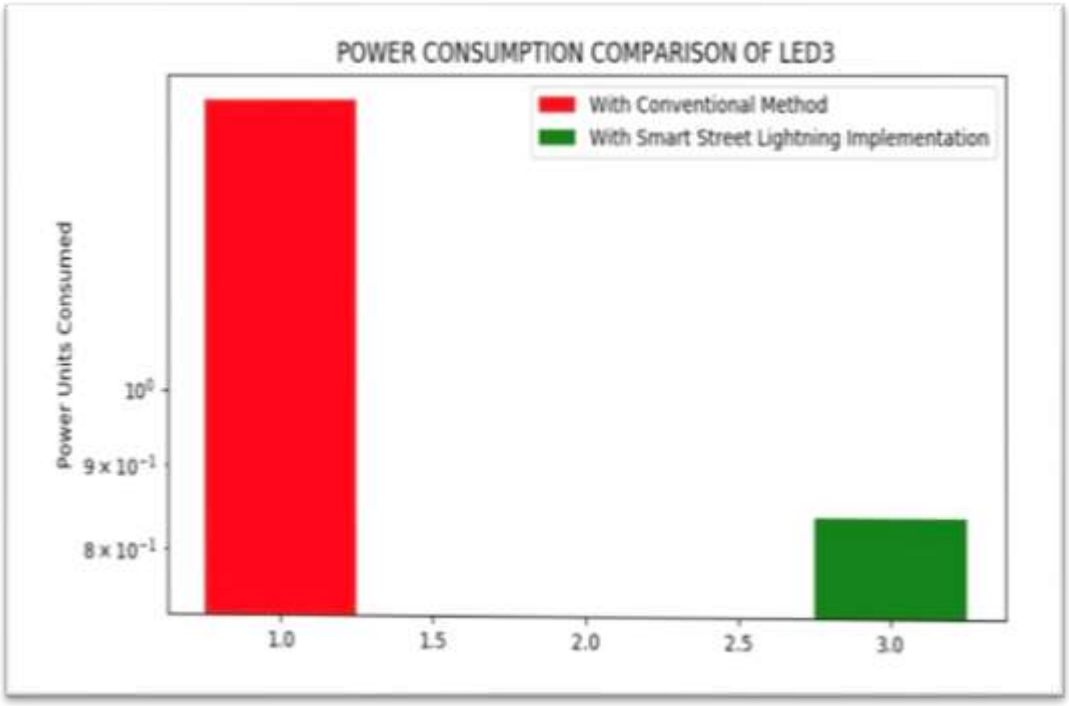
- **Screenshots showing power consumed by LED2 and corresponding consumption compared to the old system.**



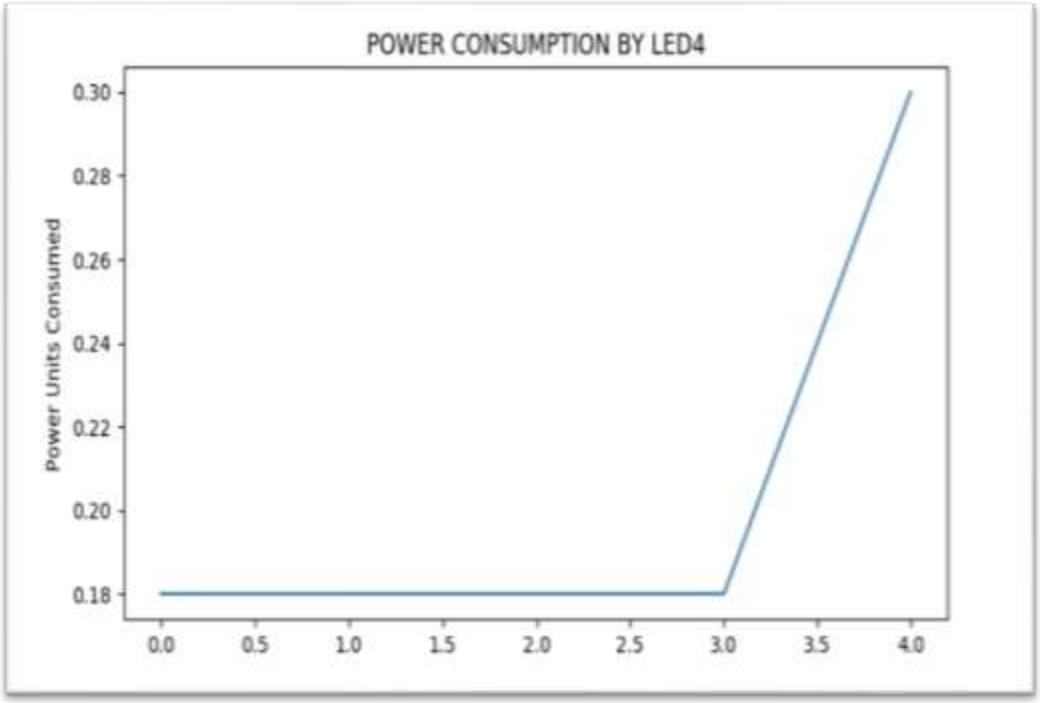


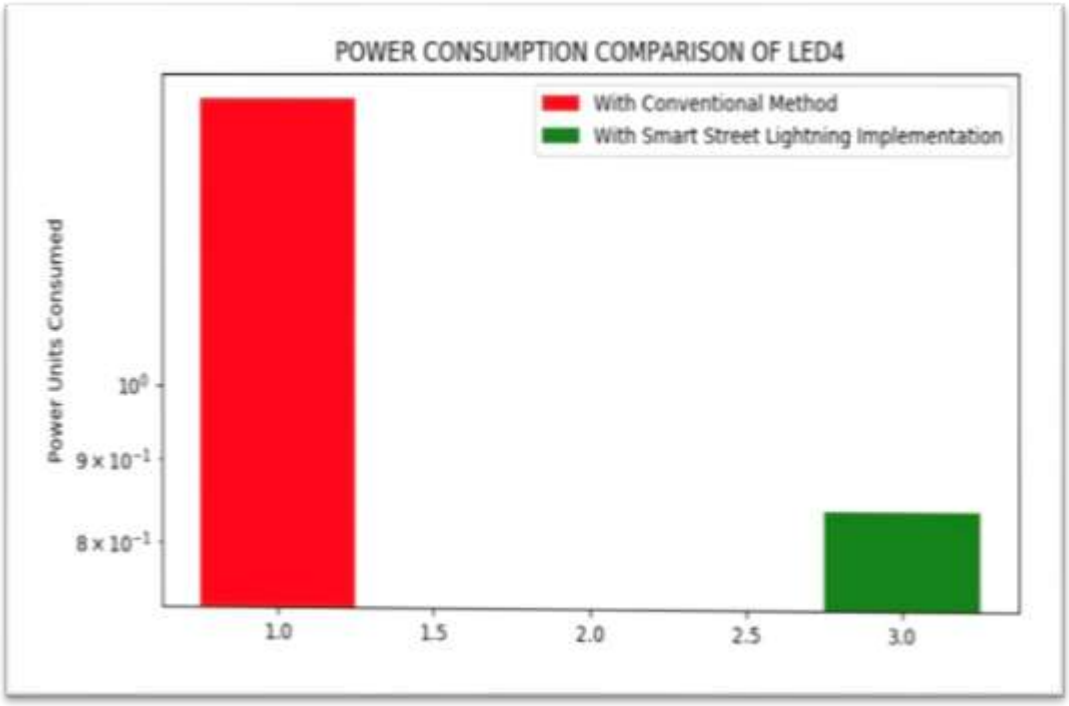
- **Screenshots showing power consumed by LED3 and corresponding consumption compared to the old system.**



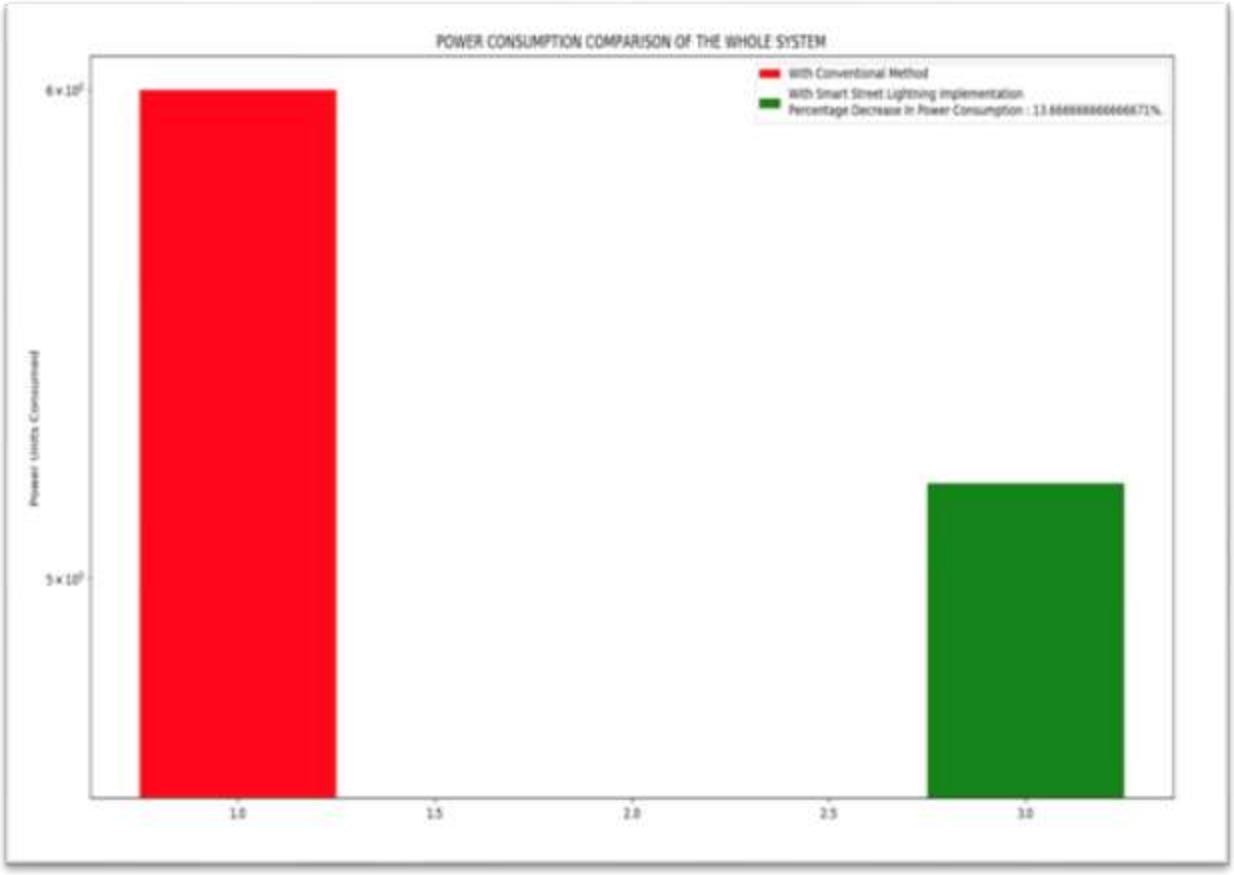


- **Screenshots showing power consumed by LED4 and corresponding consumption compared to the old system.**





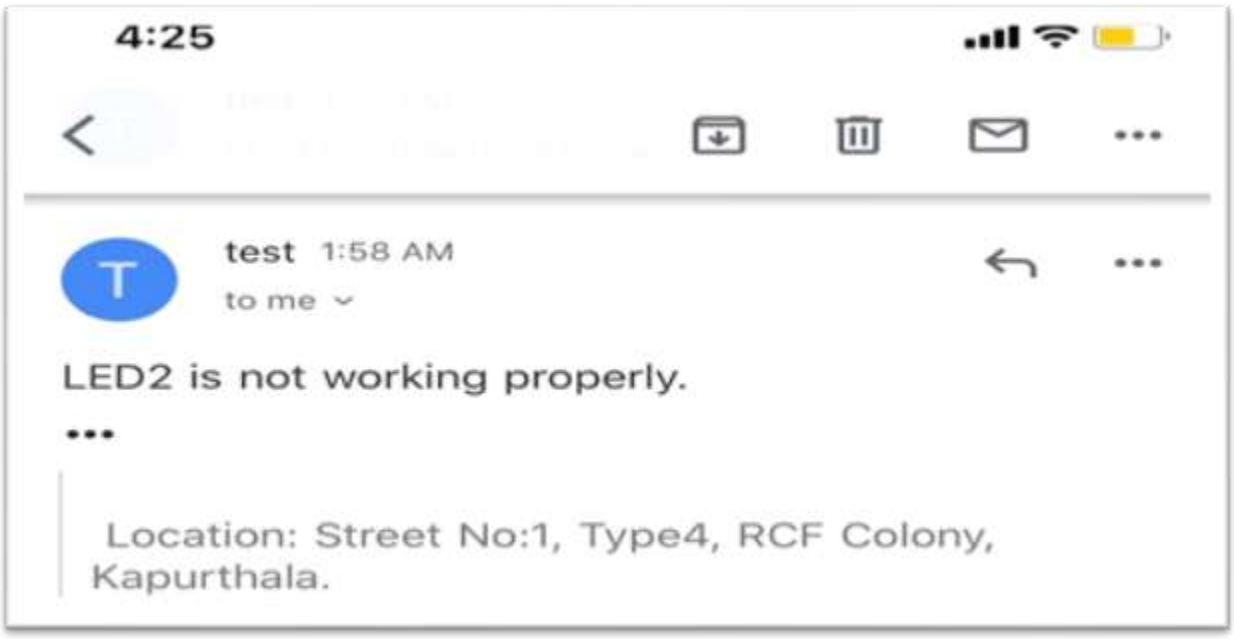
- **Screenshot showing the compared power consumption between old Conventional System and the Smart Street Lighting System.**



It shows that there is an approximate decrease in power consumption by **14%** for the particular simulation.

Alert Notifications

- **Screenshot showing the e-mail alert sent saying about the fault and location of the lamp.**



- **Screenshot showing the SMS alert sent saying about the fault and location of the lamp.**

5. CONCLUSION

1. So the required system was developed using the TinkerCad Simulation Platform and ThingSpeak cloud service.
2. The system worked as per the requirements and it successfully turned on/off the LEDs depending upon the sunlight.
3. The system successfully identified the movement of vehicles/pedestrians and successfully controlled the brightness of the the LEDs.
4. However the response time of the simulation increased with increasing circuit components.
5. The data was send to the cloud successfully.
6. The backend Python Script read data from the cloud simultaneously and performed the analysis successfully.
7. Successful plotting of Power Consumption ,Power Consumption Comparison and cloud data were done.
8. Fault detection was done accurately and the corresponding alerts were send successfully.
9. Number of simulations were done and it was found that on an average the power consumption reduced by **15%-25%.**
10. The best result found during a particular simulation was **45%** reduction in power consumption.

6. REFERENCES

[1] Soyoung Hwang and Donghui Yu, “Remote Monitoring and Controlling System Based on ZigBee Networks”International Journal of Software Engineering and Its Applications Vol. 6, No. 3, July, 2012.

[2] B.K. Subramanyam, K. Bhaskar Reddy, P. Ajay Kumar Reddy. “Design and development of intelligent wireless street light control and monitoring system along with GUI.” IJERA, Vol. 3, Issue4, 2013: pp. 2115-2119.

[3] P.Nithya, N.Kayalvizhi, “Design of Wireless Framework for Energy Efficient Street Light Automation”, International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014.

[4] R. Kavitha1, N. Thiyagarajan, “Distributed Intelligent Street Lamp Monitoring and Control System Based on Zigbee” International Journal of Science and Research (IJSR) P.P 2319-7064.

[5] Anila Devi Y, V. Jaya Prakash, “GSM Based Remote Control System of High Efficiency Intelligent Street Lighting System Using A Zigbee Network of Devices and Sensor” International Journal of Science and Research (IJSR) P.P 2319-7064.

[6] Richu Sam Alex, R NarcissStarbell “Energy Efficient Intelligent Street Lighting System Using ZIGBEE and Sensors”, International Journal of Engineering and Advanced Technology (IJEAT), P.P2249 – 8958, Volume-3, Issue-4, April 2014.

[7] Srikanth M1, Sudhakar K N, “ZigBee Based Remote Control Automatic Street Light System” P.P 2321 - 3361, IJESC June 2014.

[8]https://www.researchgate.net/publication/262352965_Smart_street_light_system_with_energy_saving_function_based_on_the_sensor_network

[9]https://www.itron.com/id/solutions/what-we-enable/smart-cities/intelligent-streetlights