DEPARTMENT OF INFORMATION TECHNOLOGY FACULTY OF ENGINEERING & TECHNOLOGY

IOT PROJECT REPORT

SUBJECT TITLE: INTERNET OF THINGS

SUBJECT CODE: 15IT422E

SUBMITTED TO: Dr.Kayalvizhi Jayavel

Smart Street Lighting System

RAUNAK KANKARIA

RA1611008010138

Department of Information Technology

SRM
UNIVERSITY
(Under section 3 of UGC Act 1956)

SRM University, SRM Nagar, Kattankulathur-603203 Kanchipuram District, Tamil Nadu

LINKS TO GITHUB AND YOUTUBE:

YouTube:

https://youtu.be/WeiddhKdJVE

Github:

https://github.com/Raunak1998/smart-street-lighting

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to our IoT Professor Dr. Kayalvizhi Jayavel for being out there at every step of my course and guiding us all along the way to be capable of molding our ideas into smart projects and helping us enhance our skills.

I would also like to thank my parents, without whom I wouldn’t be able to anything in any regard.

ABSTRACT

Lighting is one the most power-hungry component anywhere across globe both in commercial as well as domestic use. It is the biggest contributor in energy bills to the state exchanger for operating & maintaining the street light across the city or state therefore it becomes the choice for implementing latest energy efficient technologies. City administrator are open and key to adopt new products & technologies which will help them save energy & reduce their power bills significantly.

The light industry is already introducing & implementing energy efficient LED technologies which reduce energy consumption significantly. This would further cut down by the use of cutting-edge Internet of Things concept. We have developed IoT based wireless controller & software platform to control & manage streetlights or commercial facilities lights. We use platform state of the art and wireless mesh technology to convert these lights locally and then connect to the cloud via some Wi-Fi internet gateway. The data on the cloud platform helps the user or concerned authority to monitor, control & manage these assets in real time and cut down energy bills by efficient use through smart scheme usage or traffic pattern.

HARDWARE REQUIRED:

- NodeMCU ESP8266 module (Link: https://www.amazon.in/Easy-Electronics-NodeMcu-Development-Board/dp/B06XYRS6KC/ref=sr_1_4?crid=9299GZMFXJKM&keywords=nodemcu+esp8266&qid=1555504849&s=gateway&sprefix=nodeMcu%2Caps%2C291&sr=8-4)(**Amount –** Rs.350)
- HC-SR04 Ultrasonic Sensor (Link: https://www.amazon.in/Electronics-HC-SR04-Ultrasonic-Distance-Sensor/dp/B078J3L8LD/ref=sr_1_2?keywords=ultrasonic+sensor&qid=1555504656&s=gateway&sr=8-2)(**Amount –** Rs.160)
- 1 Channel Relay
- Jumper Wires (Link: https://www.amazon.in/Jumper-Wires-Male-female-Pieces/dp/B00ZYFX6A2/ref=sr_1_10?crid=9299GZMFXJKM&keywords=n

odemcu+esp8266&qid=1555504849&s=gateway&sprefix=nodeMcu%2Caps%2C291&sr=8-10) (**Amount –** Rs.200)

- LDR (Any nearby electronic shop) (**Amount –** Rs.2)
- 10K ohm Resistor (Any nearby electronic shop) (**Amount –** Rs.1)
- Micro USB cable (**Amount –** Rs.100)
- Breadboard (**Amount –** Rs.100)

SOFTWARE REQUIRED:

- Arduino IDE.
- ESP8266 Library
- Adafruit MQTT Library
- Adafruit MQTT Client Library
- Firebase Arduino Library

TOTAL COST OF COMPONENTS: - Rs.700 - Rs.800

- ESP8266 - Rs.380
- Ultrasonic Sensor (HC-SR04) - Rs.160
- LDR -Rs.2
- Resistors – Rs.2
- Jumper wires- Rs.40
- Breadboard – Rs.100
- MicroUSB Cable – Rs.100

SYSTEM OVERVIEW

The main component of the setup is the NodeMCU ESP8266 module. All the other hardware components are connected to the NodeMCU. The microcontroller is programmed in Arduino IDE and uses the HC-SR04 (Ultrasonic Sensor), LDR and Resistors as hardware components. Adafruit MQTT Libraries and Arduino Firebase Libraries have been added to the Arduino IDE.

The NodeMCU is powered using a micro USB cable. The Ultrasonic sensor and the LDR are directly connected to the NodeMCU. For connecting the LDR to the NodeMCU we require 10k ohm resistors. The LDR detects the light intensity in the room and the ultrasonic sensor detects if someone is near the light. Whenever the light intensity falls below a threshold value and ultrasonic sensor detects someone near the light then a signal is sent and light is turned on. We have also used the Adafruit dashboard to create a manual override button, such that if any authority wants to check for faults or if they want to turn on all the lights manually, they can simply use the button to do so.

All the time-related values are also updated onto the firebase so that it can be later used for analysis that for how much time was the light switched on? Or From when to when was the light turned on?

Programming NodeMCU, setting up Adafruit Dashboard and sending values to Firebase:-

- Make all the necessary circuit connections.
- Import ESP8266WiFi.h, Adafruit MQTT, Arduino Firebase, Adafruit MQTT Client Libraries to the Arduino IDE.
- Refer the Adafruit MQTT example and copy the functions to connect to WiFi and Adafruit
- Upload LDR and Ultrasonic sensor code to your NodeMCU.
- Setup your Adafruit dashboard, add a new button and give it name.
- Run your code in Arduino ide.
- Once the code is running open COM3 port where all your readings are shown. Those readings are also updated to the firebase.
- These readings are reflected on Firebase.

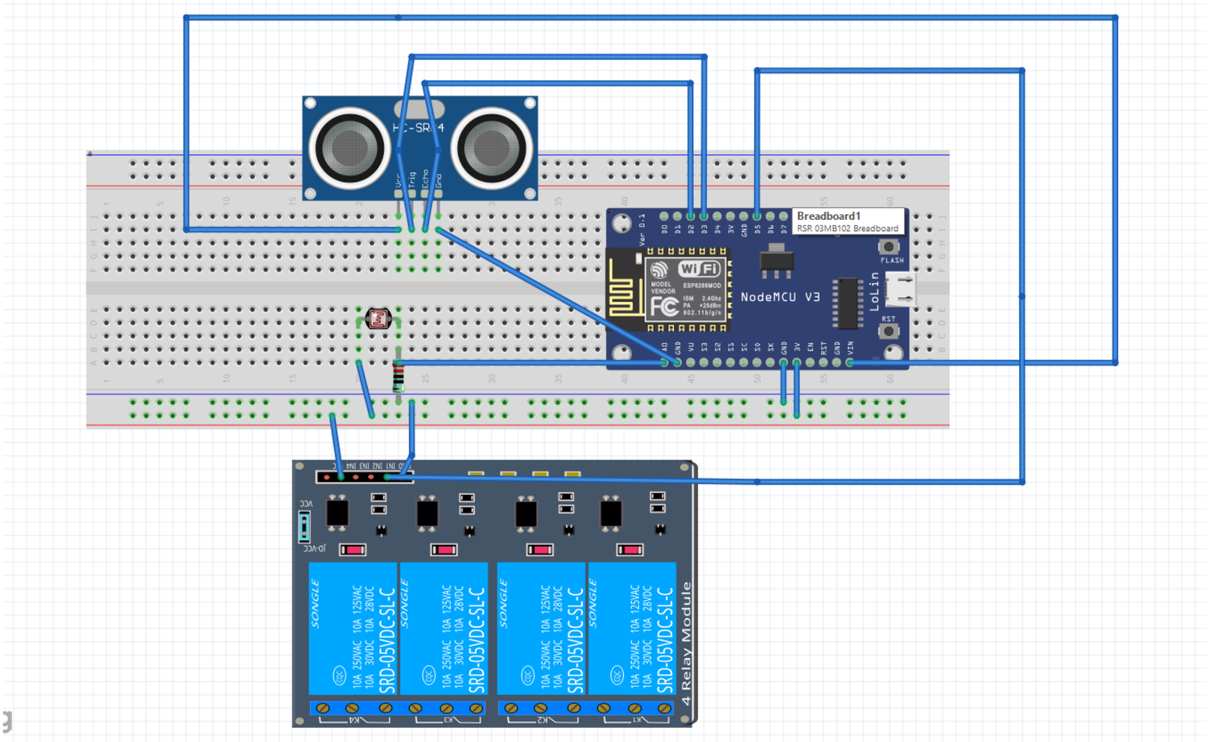
CODE

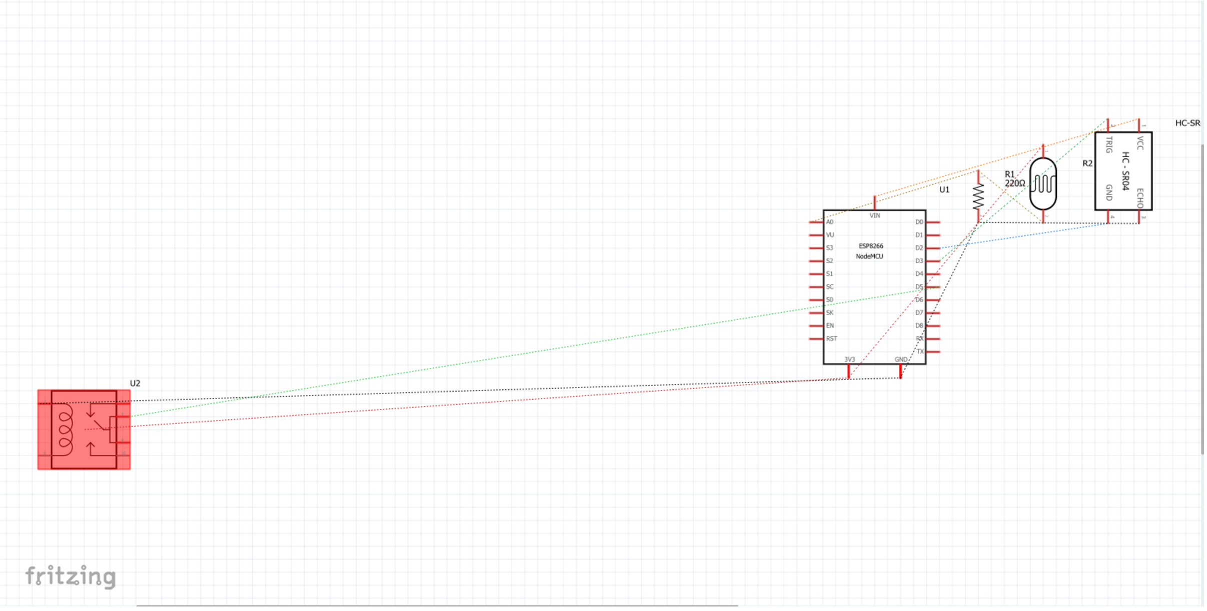
```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include<FirebaseArduino.h>

#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "raunak1998"
#define AIO_KEY         "ac661ef9a5af4106ad8d06f3025d3d50"

#define WLAN_SSID       "Raunak's 5T"
#define WLAN_PASS       "notforfree"

#define FIREBASE_HOST "street-lighting-86a4d.firebaseio.com"
#define FIREBASE_AUTH "EcmhNwfD1G4aYoOklYnhvCpiaS3dlI8HOGtHsagO"

WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT,
AIO_USERNAME, AIO_KEY);
Adafruit_MQTT_Subscribe onoffbutton = Adafruit_MQTT_Subscribe(&mqtt,
AIO_USERNAME "/feeds/street-light");
//Adafruit_MQTT_Publish photocell = Adafruit_MQTT_Publish(&mqtt,
AIO_USERNAME "/feeds/photocell");
const int trigPin = D3;
const int echoPin = D2;
const int relayPin = D5;
bool overrideOn = false;
bool turnedOn = false;

// defines variables
long duration;
int distance;
int startTime;

void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(relayPin, OUTPUT);
digitalWrite(relayPin, HIGH);
```

```
Serial.begin(9600);
Serial.print("Connecting to ");
Serial.println(WLAN_SSID);

WiFi.begin(WLAN_SSID, WLAN_PASS);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
mqtt.subscribe(&onoffbutton);
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Serial.println(Firebase.success());
}

void loop() {
    MQTT_connect();
    Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(1000))) {
    if (subscription == &onoffbutton) {
      Serial.print(F("Got: "));
      Serial.println((char *)onoffbutton.lastread);
      if(strcmp((char*)onoffbutton.lastread, "ON")==0)
        overrideOn = true;
      else
        overrideOn = false;
    }
  }
int sensorValue = analogRead(A0);

float voltage = sensorValue * (5.0 / 1023.0);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
```

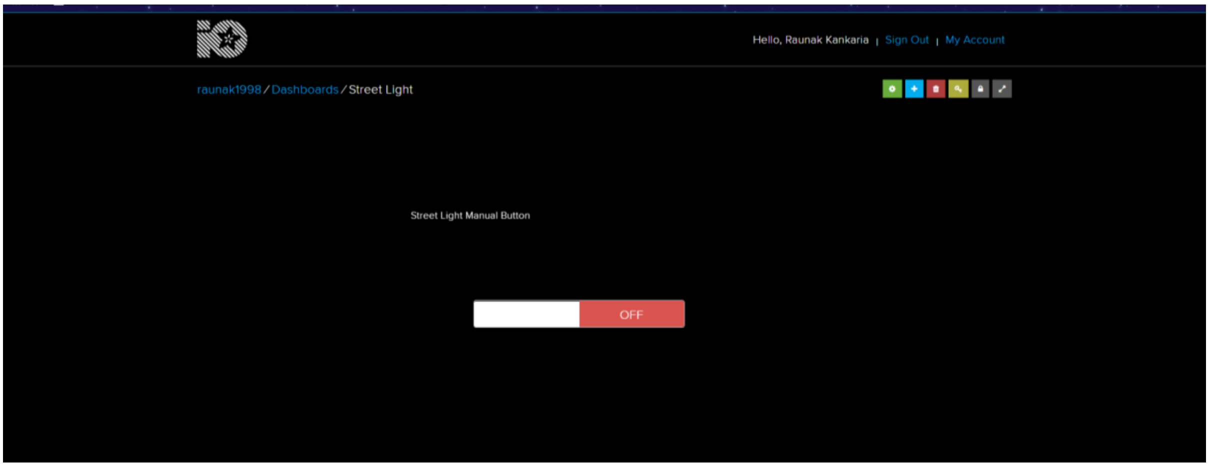
```
duration = pulseIn(echoPin, HIGH);

distance= duration*0.034/2;
Serial.print("Distance: ");
Serial.println(distance);
Serial.print("LDR: ");
Serial.println(voltage);
if(overrideOn || (voltage<2.5 && distance<50)){
    digitalWrite(relayPin, LOW);
    startTime = millis();
    turnedOn = true;
    }
    else if(turnedOn == true){
    digitalWrite(relayPin, HIGH);
    Firebase.pushFloat("/", (millis()-startTime)/(1000.0*60.0));
    turnedOn = false;
      }

   delay(1000);
}

void MQTT_connect() {
  int8_t ret;

  // Stop if already connected.
  if (mqtt.connected()) {
    return;
  }

  Serial.print("Connecting to MQTT... ");

  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
       Serial.println(mqtt.connectErrorString(ret));
       Serial.println("Retrying MQTT connection in 5 seconds...");
       mqtt.disconnect();
       delay(5000);  // wait 5 seconds
       retries--;
       if (retries == 0) {
        // basically die and wait for WDT to reset me
        while (1);
      }
  }
  Serial.println("MQTT Connected!");
}
```

Circuit Diagram: -

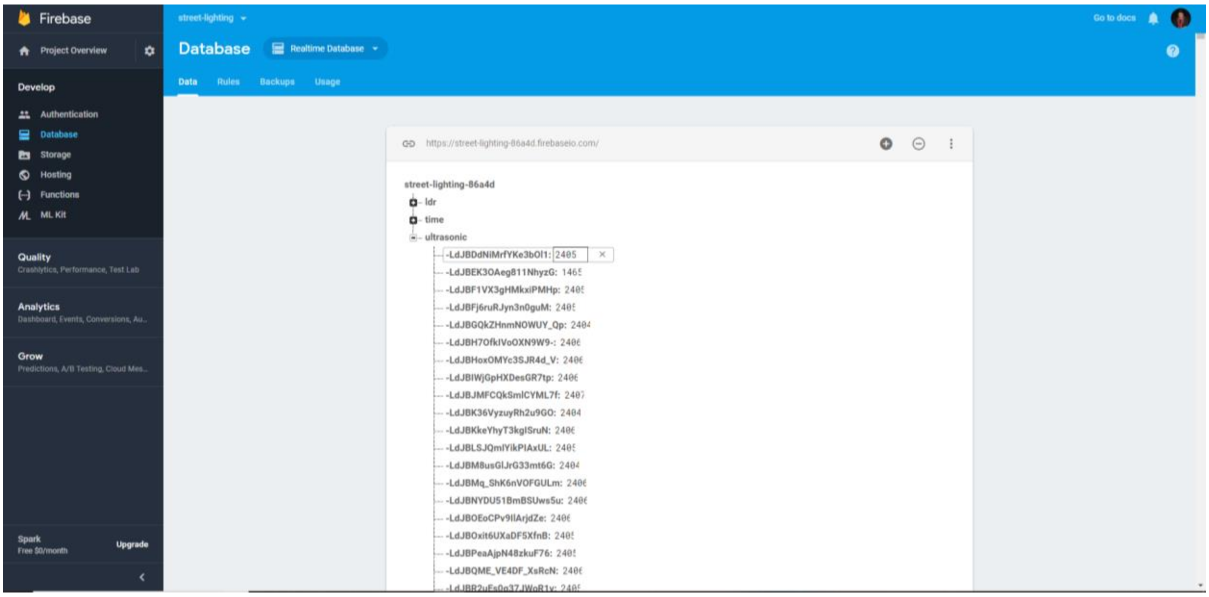




Adafruit Dashboard: -


Hello, Raunak Kankaria | Sign Out | My Account
raunak1998 / Dashboards / Street Light
Street Light Manual Button
OFF

Firebase Values: -



```
https://street-lighting-86a4d.firebaseio.com/

street-lighting-86a4d
 - ldr
 - time
 - ultrasonic
    -LdJBDdNiMrfYKe3bOl1: 2405
    -LdJBEK3OAeg811NhyzG: 1465
    -LdJBF1VX3gHMkxiPMHp: 2405
    -LdJBFj6ruRJyn3n0guM: 2405
    -LdJBGQkZHnmNOWUY_Qp: 2404
    -LdJBH7OfkIVoOXN9W9-: 2406
    -LdJBHoxOMYc3SJR4d_V: 2406
    -LdJBIWjGpHXDesGR7tp: 2406
    -LdJBJMFCQkSmlCYML7f: 2407
    -LdJBK36VyzuyRh2u9GO: 2404
    -LdJBKkeYhyT3kglSruN: 2406
    -LdJBLSJQmlYikPIAxUL: 2405
    -LdJBM8usGlJrG33mt6G: 2404
    -LdJBMq_ShK6nVOFGULm: 2406
    -LdJBNYDU51BmBSUws5u: 2406
    -LdJBOEoCPv9llArjdZe: 2406
    -LdJBOxit6UXaDF5XfnB: 2405
    -LdJBPeaAjpN48zkuF76: 2405
    -LdJBQME_VE4DF_XsRcN: 2406
    -LdJBR2uFs0g37JWoR1v: 2405
```

Result:

Smart Street Lighting System using ESP8266 NodeMCU, ultrasonic sensor and LDR has been successfully developed and implemented.