DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING & TECHNOLOGY

IOT PROJECT REPORT

SUBJECT TITLE : INTERNET OF THINGS

SUBJECT CODE: 15IT422E

SUBMITTED TO: Prof Kayalvizhi Jayavel

HOME AUTOMATION SYSTEM

HARIHARAN S

RA1611008010115

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/k1RowOjPz24

Github:

https://github.com/09hari

ACKNOWLEDGEMENT

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

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

ABSTRACT

Home automation system achieved great popularity in the last decades and it increases the comfort and quality of life. In this paper an overview of current and emerging home automation systems is discussed. Nowadays most home automation systems consist of a smartphone and microcontroller. A smart phone application is used to control and monitor the home appliances using different type of communication techniques. In this paper the working principle of different type of wireless communication techniques such as ZigBee, Wi-Fi, Bluetooth, EnOcean and GSM are studied and their features are compared with each other so the users can choose their own choice of technology to build home automation system. Moreover in this research work the survey of different home automation systems is discussed and their advantages and drawbacks are also highlighted.

HARDWARE REQUIRED:

- NodeMCU ESP8266
- Bread Board
- Female-Female Jumper wires
- 2-Channel Relay

SOFTWARE REQUIRED :

- Arduino IDE.
- ESP8266 library.
- Adafruit MQTT Library.

MISCELLANEOUS :

- Wifi Internet Connection
- Soldering tool
- Soldering wires

TOTAL COST OF COMPONENTS :- RS.1800 - RS.2000.

SYSTEM OVERVIEW

The main component of the setup is the Nodemcu ESP8266 module. All the other hardware components are connected to the Nodemcu. The board is programmed in Arduino IDE and uses the ESP8266, Adafruit MQTT libraries. These libraries have been added to the Arduino IDE. The MPU6050 module with gyroscope and accelerometer is directly connected to the Nodemcu using a micro USB cable.

Programming NodeMCU and setting up Adafruit Dashboard :

- Import Adafruit Library to the Arduino IDE
- Fill your Adafruit token, ssid, and wifi password to the code.
- Upload home automation code to your NodeMCU
- Setup your Adafruit dashboard, add a custom widget, choose two state widget and give it name.

CODE

```
#include <ESP8266WiFi.h>


#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

/********* WiFi Access Point ***********/
```

```
#define WLAN_SSID       "Incubation"
#define WLAN_PASS       "SRMIPcenter"

/********* Adafruit.io Setup ***********/

#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883
for SSL
#define AIO_USERNAME    "Dymatize"
#define AIO_KEY
"9debf6a3fa834e84bf3aa573eb5948eb"

/**** Global State (you don't need to change this!)
******/

// Create an ESP8266 WiFiClient class to connect to
the MQTT server.
WiFiClient client;
// or... use WiFiFlientSecure for SSL
//WiFiClientSecure client;

// Setup the MQTT client class by passing in the WiFi
client and MQTT server and login details.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER,
AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

/********** Feeds *************/

// Setup feeds for publishing.
// Notice MQTT paths for AIO follow the form:
<username>/feeds/<feedname>
// Setup feeds subscribing to changes.
```

```
Adafruit_MQTT_Subscribe LED =
Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/light");

/********* Sketch Code ************/

// Bug workaround for Arduino 1.6.6, it seems to need
a function declaration
// for some reason (only affects ESP8266, likely an
arduino-builder bug).
void MQTT_connect();

void setup() {
  Serial.begin(115200);
  delay(10);
pinMode(D3,OUTPUT);
  // Connect to WiFi access point.
  Serial.println(); Serial.println();
  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());

  // Setup MQTT subscription
  mqtt.subscribe(&LED);


}

uint32_t x=0;

void loop() {

  MQTT_connect();

  Adafruit_MQTT_Subscribe *subscription;
  while ((subscription =
mqtt.readSubscription(5000))) {
    if (subscription == &LED) {
      Serial.println((char *)LED.lastread);
      int state=atoi((char*)LED.lastread);
       digitalWrite(D3,state);
    }
  }

 // Now we can publish stuff!

  // ping the server to keep the mqtt connection alive
  // NOT required if you are publishing once every
KEEPALIVE seconds
  /*
  if(! mqtt.ping()) {
    mqtt.disconnect();
  }
  */
}
```

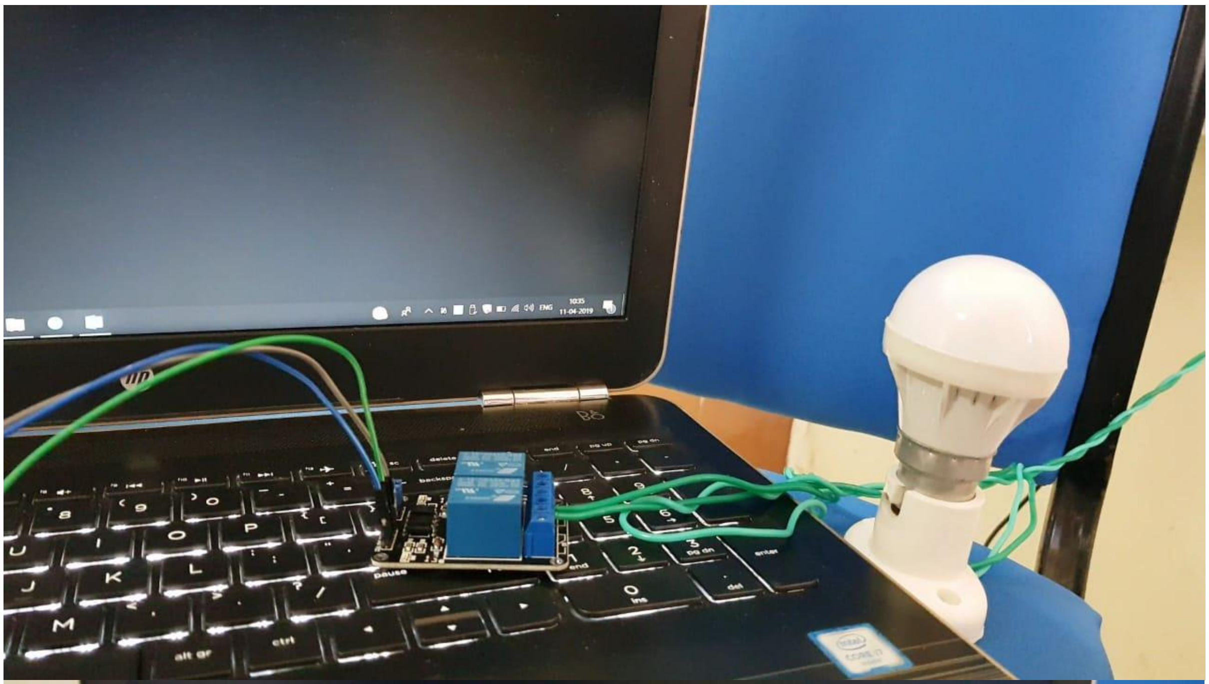
```
// Function to connect and reconnect as necessary to
the MQTT server.
// Should be called in the loop function and it will take
care if connecting.
void MQTT_connect() {
  int8_t ret;

  // 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!");
}
```

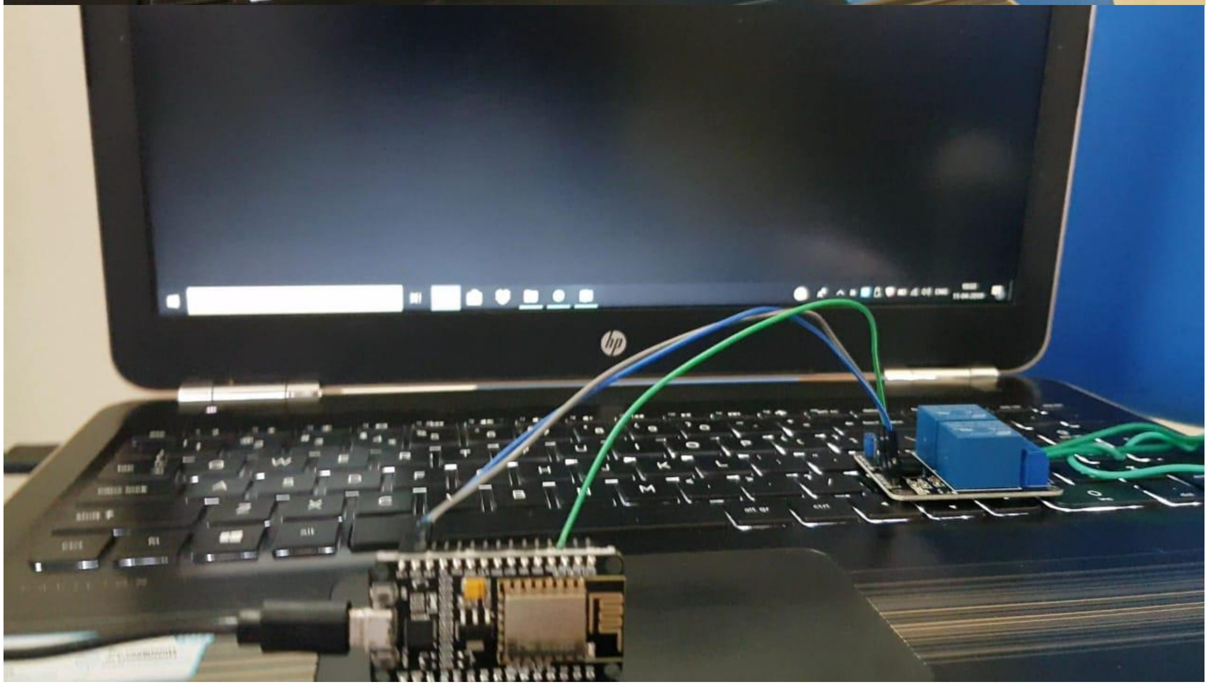
Adafruit Dashboard, Widgets and Triggers :

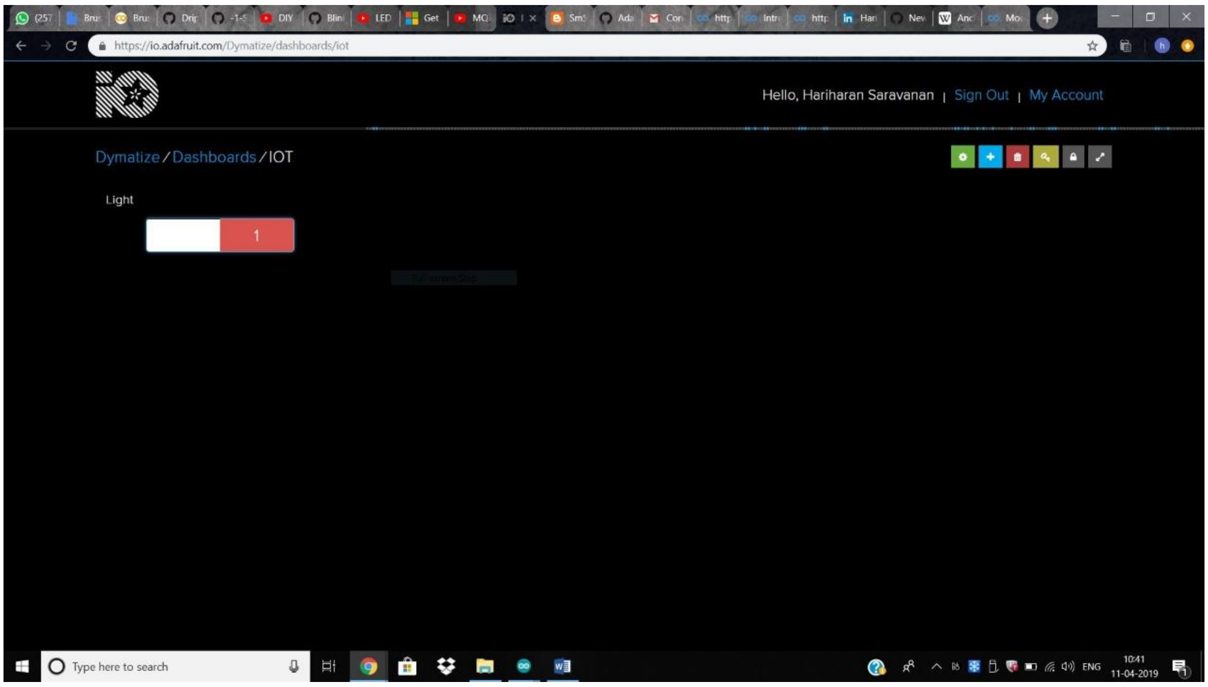
We give our WiFi SSID and password inside the code before compiling and uploading it to the Nodemcu. Also in order to connect to the Adafruit database, we give the Adafruit username, Adafruit password and Adafruit client ID so that it can connect via MQTT.

Result :

Home Automation using ESP8266 NodeMCU has been successfully developed and implemented.






https://io.adafruit.com/Dymatize/dashboards/iot
Hello, Hariharan Saravanan | Sign Out | My Account
Dymatize / Dashboards / IOT
Light
1
Type here to search
10:41
11-04-2019