

K L UNIVERSITY

Academic year: 2021-2022 (EVENSEM)

19TS4006

TECHNICAL PROFICIENCY & TRAINING-2

A Project based lab report on **“SMART HOME AUTOMATION USING ALEXA”**

Submitted by:

NAME:	ID.NO
M.Vaishnavi	190040295
G.HemaReddy	190040141
CH.Priya	190040382
V.Ramya	190040564

UNDER THE ESTEEMED GUIDANCE OF

Dr.China Satyanarayana



**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING KONERU
LAKSHMIAH EDUCATION FOUNDATION VADDESWAREM, -522502**

Guntur Dt., AP India



CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Project Report entitled, **“SMART HOME AUTOMATION USING ALEXA”** Submitted by **M.Vaishnavi, G.Hema Reddy, CH.Priya, V.Ramya** , bearing Reg Id. No **190040295,190040141,190040382, 190040564**, in partial fulfillment of the requirements for the award of the **Bachelor of Technology (B.Tech) in Electronics & Communication Engineering (ECE)** and submitted to the Department of Electronics & Communication Engineering of KLEF, Vaddeswaram, Guntur is an authentic record of my own work carried out during a period from August 2020 to November 2020 under the supervision of **Dr.China Satyanarayana, ECE Department.**

PROJECT SUPERVISOR

HEAD OF THE DEPARTMENT

Dr.China Satyanarayana

LIST OF CONTENTS

CHAPTER	TITLE	PAGE.NO
1.	ABSTRACT	4
2.	INTRODUCTION	5
3.	CIRCUIT DIAGRAM	6
4.	DESCRIPTION OF COMPONENTS & PROCEDURE	7-10
5.	CODE	11-22
6.	RESULT AND OUTPUT	23-24
7.	CONCLUSIONS	25

ABSTRACT

Now-a-days with the advancement in the technologies, IOT is boosting up continuously by deploying and innovating various smart devices which helps to modernize the society. This project Smart Home Automation using Alexa is already implemented with quiet expensive devices and used by many people but not everyone can afford that model to convert their home into smart home. Through this project the people who dreamt of living in a smart home with less cost and want to get hands on experience of automated device technologies will come true. In this the user will learn and use the recent trending technologies which is machine learning and also provide the information about and how to manage their usages and stay budget friendly. All this things will be done through a website that is created for this model users. In this way the number of people who own Smart Homes will increase and helps to accomplish the dream of Modern India .

INTRODUCTION

A few years ago, I explored how to use Alexa, an intelligent personal assistant made popular by the Amazon Echo and Echo-Dot, in Home Automation projects:

When IoT Meets AI: Home Automation With Alexa and NodeMCU and
Alexa — NodeMCU: WeMo Emulation Made Simple

Alexa is capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, and other real-time information. Alexa can also control several smart devices using itself as a home automation hub. We will use on this project, the “Echo Dot”, which allows users to activate the device using a wake-word, such as “Alexa” or “Computer”, as in “Star Trek!.

In the home automation space, Alexa can interact with several different devices like Philips Hue, Belkin Wemo, Sonoff, etc. We emulated WeMo devices on previous projects, using the fauxmoESP library, which seems outdated and not adequately maintained nowadays. Here we will use Espalexa, an easy to use Arduino library compatible with both ESP8266 and ESP32.

Circuit Diagram:

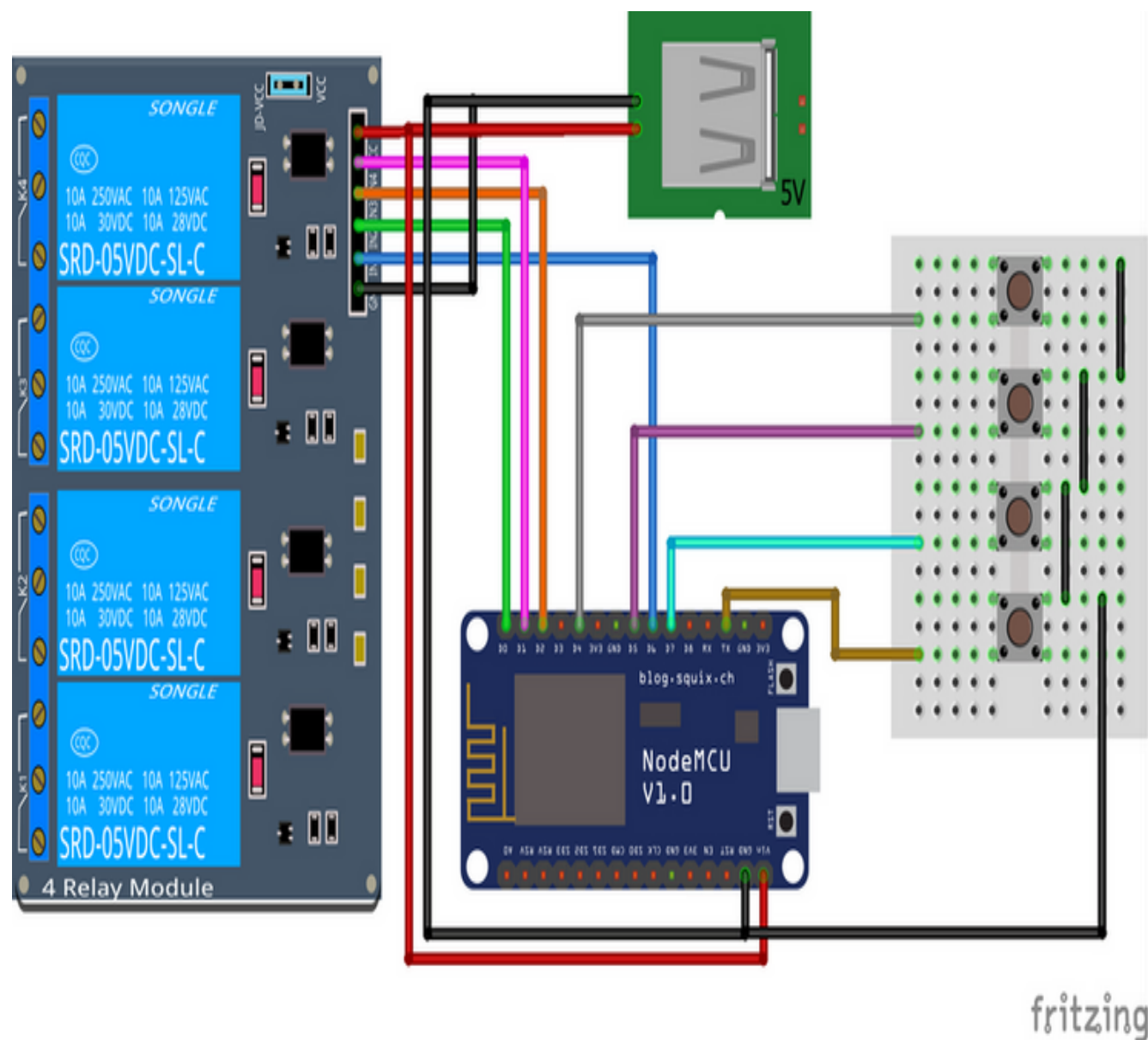


Fig.1.1

Components:

- Node MCU
- 4Relay Board
- Smart Bulbs
- Bulb Holders
- Mobile Phone
- Bread Board
- Connecting Wires
- Push Buttons/Switches
- 5v Battery

PROCEDURE:

Installation of ESP Boards:

At Arduino IDE Preferences → Additional boards Manager URLs enter with:

https://dl.espressif.com/dl/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

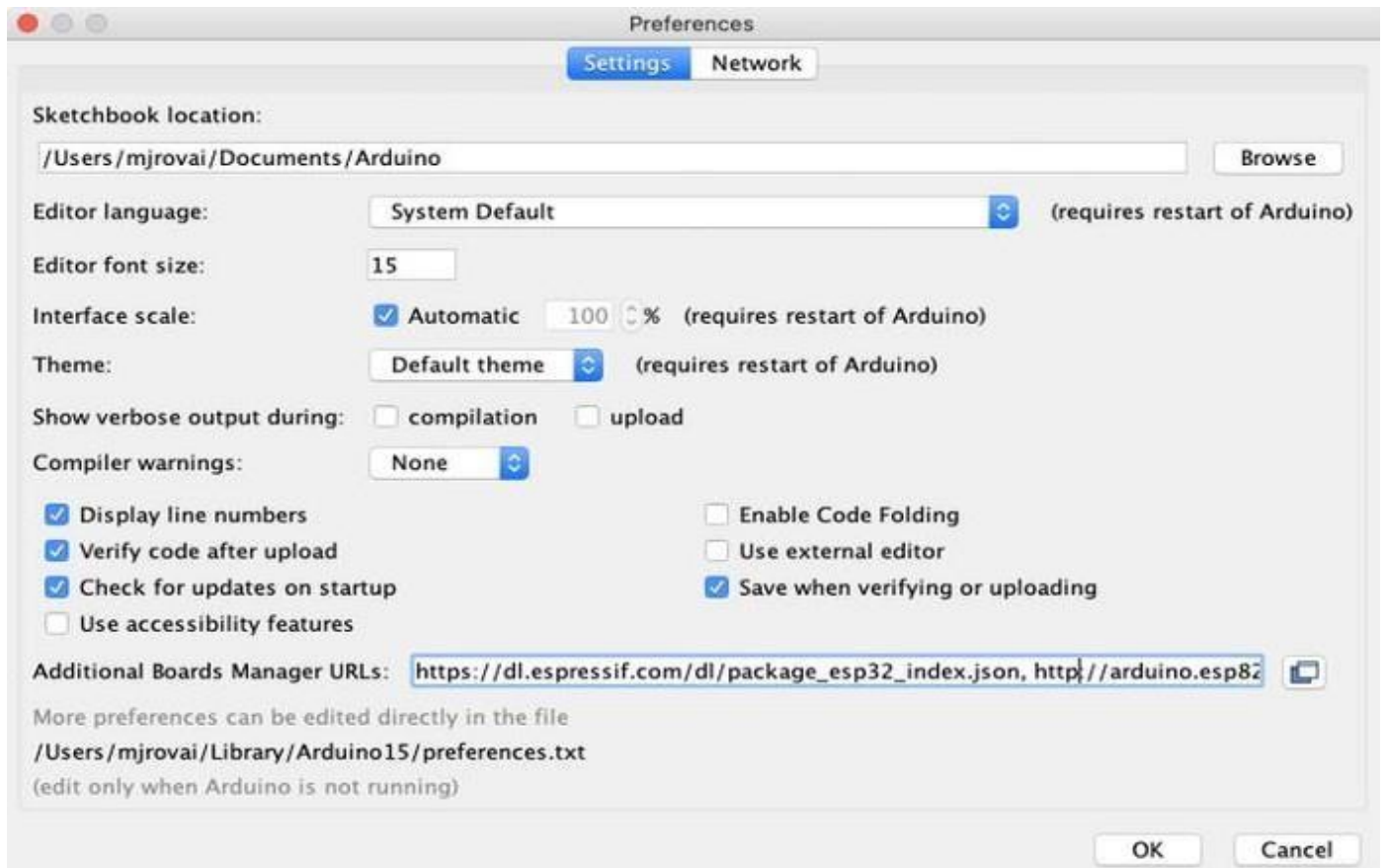


Fig.1.2

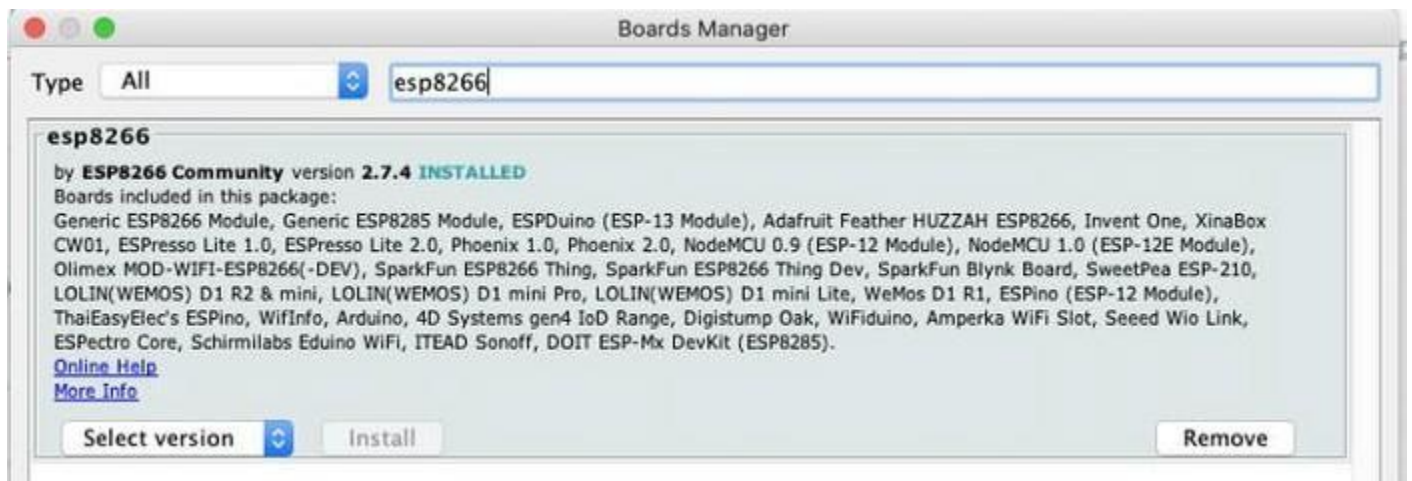


Fig.1.3

Downloading The Libraries:

- * ArduinoJson Library
- * arduinoWebSockets Library

Downloaded the libraries and added them to the Arduino Software by selecting tools ➤ Include Library ➤ Add Zip Library

Software Installation: Sinric Pro

Opened the Sinric Pro Website in the browser and logged in with credentials. In the devices added the four smart devices and got their respective Device Id's. After logging into sinric account got my API key. Now these device id's and api key is copied and pasted in the code.

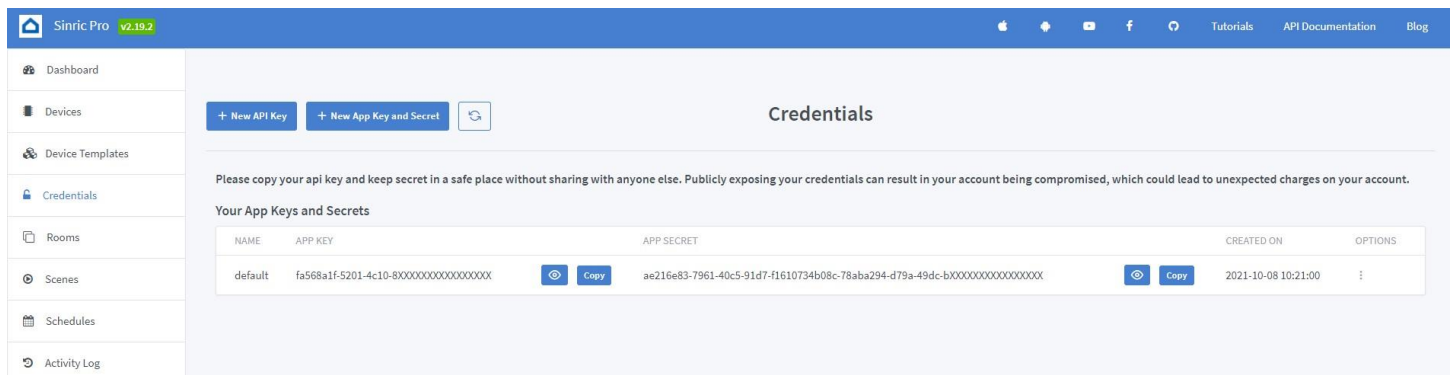


Fig.1.4

➤ Added my wifi credentials in the code and connected the circuit as per the circuit Diagram.

➤ Now installed the ALEXA APP in my mobile phone and logged in with my Amazon account

Enabled Sinric Pro Skill in Alexa app Settings ➤ Skills & Games ➤ Search ➤ Sinric Pro ➤

➤ Select Devices ➤ Add Devices ➤ Selected the four devices that I connected in the app
Run the code and upload it to the Node MCU.

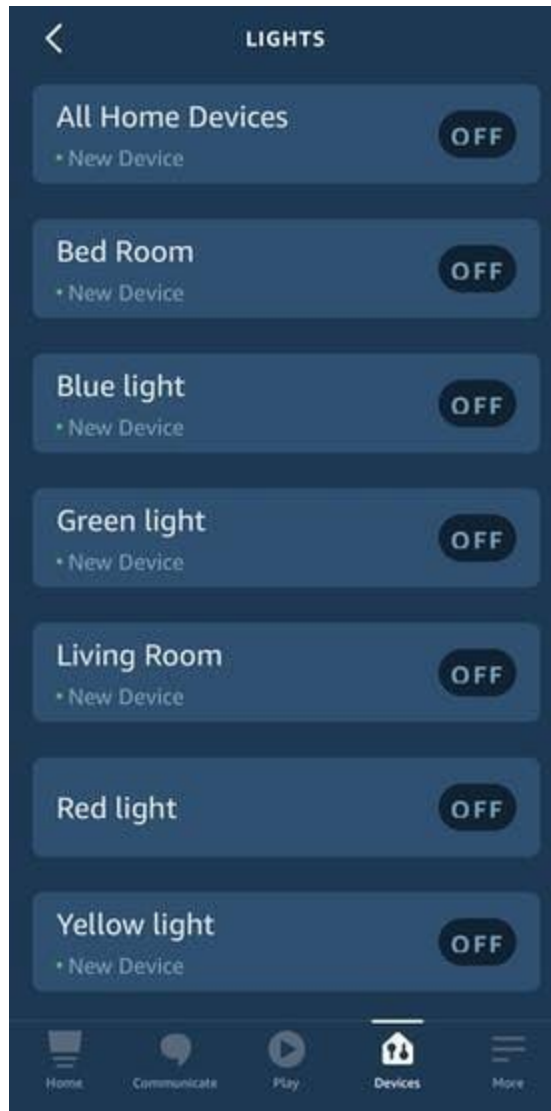


Fig.1.5

7 After completion of uploading the code the wifi will get connected and now we can control the devices.

I?_?9??

```
00:13:42.947 -> [Wifi]: Connecting..connected!
00:13:43.460 -> [WiFi]: IP-Address is 192.168.0.102
00:13:46.642 -> 616af1a870291a5128016434: off
00:14:58.779 -> 616af1a870291a5128016434: on
00:15:11.792 -> 616af1a870291a5128016434: off
00:15:23.728 -> 616af1a870291a5128016434: on
00:15:29.533 -> 616af1a870291a5128016434: off
```

Fig.1.6

CODE:

```
#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <ESP8266WiFiMulti.h>

#include <WebSocketsClient.h>

#include <ArduinoJson.h>

#include <StreamString.h>

ESP8266WiFiMulti WiFiMulti;

WebSocketsClient websocket;

WiFiClient client;

#define MyApiKey "-----" // TODO: Change to your sinric API Key. Your API
Key is displayed on sinric.com dashboard

#define MySSID "----" // TODO: Change to your Wifi network SSID

#define MyWifiPassword "-----" // TODO: Change to your Wifi network password
```

```
#define HEARTBEAT_INTERVAL 300000 // 5 Minutes
```

```
uint64_t heartbeatTimestamp = 0;
```

```
bool isConnected = false;
```

```
String device_ID_1 = " -----";
```

```
String device_ID_2 = " -----";
```

```
String device_ID_3 = " -----";
```

```
String device_ID_4 = " -----";
```

```
// Relays
```

```
#define RelayPin1 D1 //D1
```

```
#define RelayPin2 D2 //D2
```

```
#define RelayPin3 D5 //D5
```

```
#define RelayPin4 D6 //D6
```

```
// Switches
```

```
#define SwitchPin1 10 //SD3
```

```
#define SwitchPin2 D3 //D3
```

```
#define SwitchPin3 D7 //D7
```

```
#define SwitchPin4 3 //RX
```

```
//WiFi Status LED
```

```
#define wifiLed D0 //D0
```

```
int toggleState_1 = 1; //Define integer to remember the toggle state for relay 1
```

```
int toggleState_2 = 1; //Define integer to remember the toggle state for relay 2
int toggleState_3 = 1; //Define integer to remember the toggle state for relay 3
int toggleState_4 = 1; //Define integer to remember the toggle state for relay 4
```

```
void setPowerStateOnServer(String deviceId, String value);
```

// deviceId is the ID assigned to your smart-home-device in sinric.com dashboard. Copy it from dashboard and paste it here

```
void turnOn(String deviceId) {
  if (deviceId == device_ID_1) // Device ID of 1st device
  {
    Serial.print("Turn on device id: ");
    Serial.println(deviceId);
    digitalWrite(RelayPin1, LOW);
    toggleState_1 = 0;
  }
  if (deviceId == device_ID_2) // Device ID of 2nd device
  {
    Serial.print("Turn on device id: ");
    Serial.println(deviceId);
    digitalWrite(RelayPin2, LOW);
    toggleState_2 = 0;
  }
}
```

```
if (deviceId == device_ID_3) // Device ID of 3rd device

{
    Serial.print("Turn on device id: ");
    Serial.println(deviceId);
    digitalWrite(RelayPin3, LOW);
    toggleState_3 = 0;
}

if (deviceId == device_ID_4) // Device ID of 4th device
{
    Serial.print("Turn on device id: ");
    Serial.println(deviceId);
    digitalWrite(RelayPin4, LOW);
    toggleState_4 = 0;
}
}
```

```
void turnOff(String deviceId) {
    if (deviceId == device_ID_1) // Device ID of 1st device
    {
        Serial.print("Turn off Device ID: ");
        Serial.println(deviceId);
        digitalWrite(RelayPin1, HIGH);
        toggleState_1 = 1;
    }
}
```

```

}

if (deviceId == device_ID_2) // Device ID of 2nd device
{
    Serial.print("Turn off Device ID: ");

    Serial.println(deviceId);
    digitalWrite(RelayPin2, HIGH);
    toggleState_2 = 1;
}

if (deviceId == device_ID_3) // Device ID of 3rd device
{
    Serial.print("Turn off Device ID: ");
    Serial.println(deviceId);
    digitalWrite(RelayPin3, HIGH);
    toggleState_3 = 1;
}

if (deviceId == device_ID_4) // Device ID of 4th device
{
    Serial.print("Turn off Device ID: ");
    Serial.println(deviceId);
    digitalWrite(RelayPin4, HIGH);
    toggleState_4 = 1;
}
}

```

```

void websocketEvent(WStype_t type, uint8_t * payload, size_t length) {

switch (type) {

case WStype_DISCONNECTED:

    isConnected = false;

    WiFiMulti.addAP(MySSID, MyWifiPassword);

    Serial.printf("[WSc] Webservice disconnected from sinric.com!\n");


break;

case WStype_CONNECTED: {

    isConnected = true;

    Serial.printf("[WSc] Service connected to sinric.com at url: %s\n", payload);

    Serial.printf("Waiting for commands from sinric.com ...\n");

}

break;

case WStype_TEXT: {

    Serial.printf("[WSc] get text: %s\n", payload);


    #if ARDUINOJSON_VERSION_MAJOR == 5

        DynamicJsonBuffer jsonBuffer;

        JsonObject& json = jsonBuffer.parseObject((char*)payload);

    #endif


    #if ARDUINOJSON_VERSION_MAJOR == 6

        DynamicJsonDocument json(1024);

```



```

deserializeJson(json, (char*) payload);
    #endif

    String deviceId = json ["deviceId"];

String action = json ["action"];

    if (action == "setPowerState") { // Switch or Light
        String value = json ["value"];
        if (value == "ON") {
            turnOn(deviceId);
        } else {

            turnOff(deviceId);
        }
    }
    else if (action == "test") {
        Serial.println("[WSc] received test command from sinric.com");
    }
}

break;
case WStype_BIN:
    Serial.printf("[WSc] get binary length: %u\n", length);
    break;
}
}

```

```

void with_internet(){
  //Manual Switch Control
  if (digitalRead(SwitchPin1) == LOW)
  {
    delay(200);
    if(toggleState_1 == 1)
    {
      digitalWrite(RelayPin1, LOW); // turn on relay 1
      toggleState_1 = 0;
      setPowerStateOnServer(device_ID_1, "ON");
      Serial.println("Device1 ON");
    }
  }
  else
  {
    digitalWrite(RelayPin1, HIGH); // turn off relay 1

    toggleState_1 = 1;
    setPowerStateOnServer(device_ID_1, "OFF");
    Serial.println("Device1 OFF");
  }
}
else if (digitalRead(SwitchPin2) == LOW)
{
  delay(200);
  if(toggleState_2 == 1)

```

```

{
  digitalWrite(RelayPin2, LOW); // turn on relay 2
  toggleState_2 = 0;

  setPowerStateOnServer(device_ID_2, "ON");
  Serial.println("Device2 ON");
}
else
{
  digitalWrite(RelayPin2, HIGH); // turn off relay 2

  toggleState_2 = 1;
  setPowerStateOnServer(device_ID_2, "OFF");
  Serial.println("Device2 OFF");
}
}
else if (digitalRead(SwitchPin3) == LOW)
{
  delay(200);
  if(toggleState_3 == 1)
  {
    digitalWrite(RelayPin3, LOW); // turn on relay 3
    toggleState_3 = 0;
    setPowerStateOnServer(device_ID_3, "ON");
  }
}

```

```
Serial.println("Device3 ON");
}
else
{

digitalWrite(RelayPin3, HIGH); // turn off relay 3
toggleState_3 = 1;
setPowerStateOnServer(device_ID_3, "OFF");
Serial.println("Device3 OFF");
}
}
else if (digitalRead(SwitchPin4) == LOW)
{
delay(200);
if(toggleState_4 == 1)
{
digitalWrite(RelayPin4, LOW); // turn on relay 4

toggleState_4 = 0;
setPowerStateOnServer(device_ID_4, "ON");
Serial.println("Device4 ON");
}
else
{
digitalWrite(RelayPin4, HIGH); // turn off relay 4
```

```
toggleState_4 = 1;
setPowerStateOnServer(device_ID_4, "OFF");
Serial.println("Device4 OFF");
}
}
```

```
delay(100);
}
```

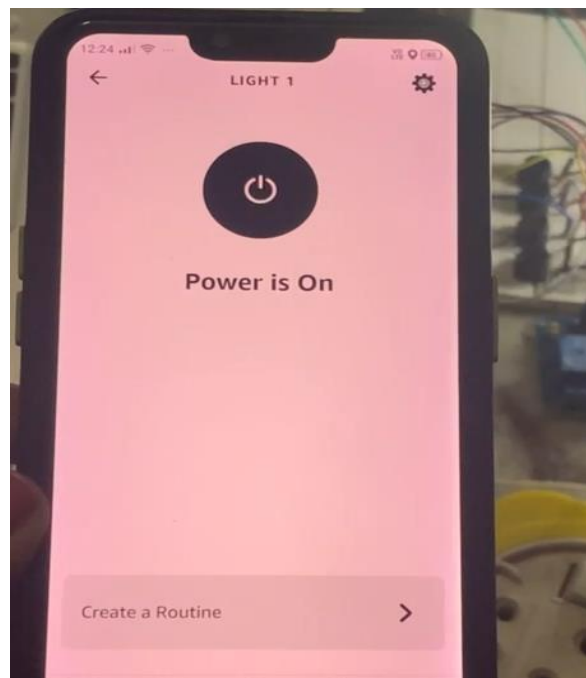
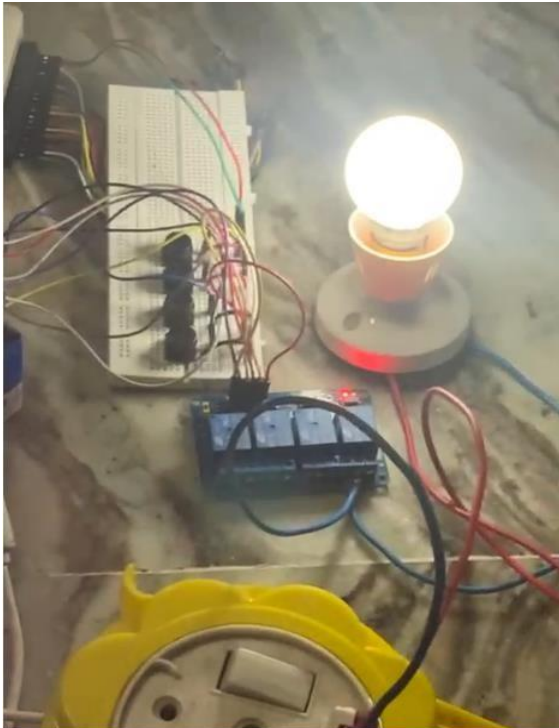
```
void without_internet()
{
//Manual Switch Control
if (digitalRead(SwitchPin1) == LOW)
{
delay(200);
if(toggleState_1 == 1)
{
digitalWrite(RelayPin1, LOW); // turn on relay 1
toggleState_1 = 0;
Serial.println("Device1 ON");
}
else
{
digitalWrite(RelayPin1, HIGH); // turn off relay 1
```

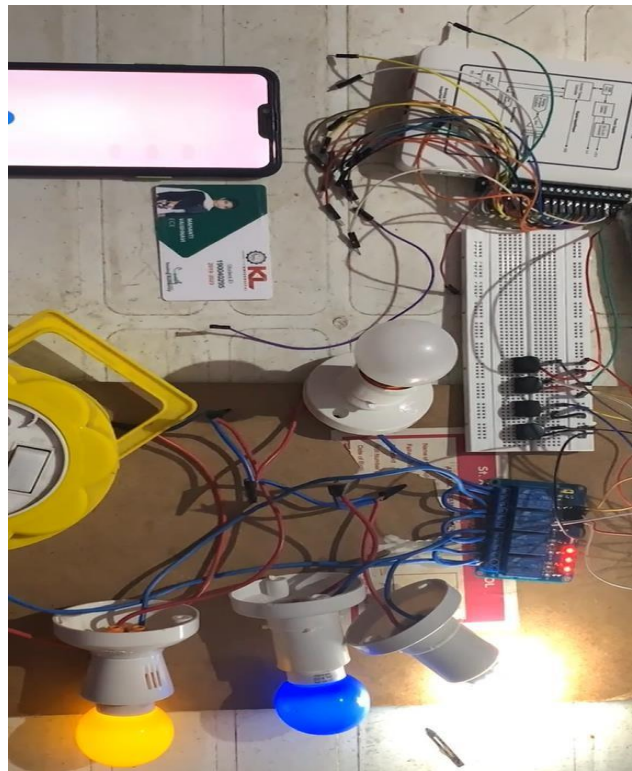
```
toggleState_1 = 1;
Serial.println("Device1 OFF");
}
}
else if (digitalRead(SwitchPin2) == LOW)
{

delay(200);
if(toggleState_2 == 1)
{
digitalWrite(RelayPin2, LOW); // turn on relay 2
toggleState_2 = 0;
Serial.println("Device2 ON");
}
else
{
digitalWrite(RelayPin2, HIGH); // turn off relay 2
toggleState_2 = 1;
Serial.println("Device2 OFF");
```

Result & Output:

The Smart Bulb is controlled by the android mobile Alexa app and by switches/push buttons as well.





CONCLUSION

The SMART HOME AUTOMATION USING ALEXA is successfully implemented using Node MCU. With this project we can control our home appliances like Smart lights, fan, water heater, Air conditioner etc.. easily by both manual and through mobile phone.

In this project we can control devices with very lowcost as we are using the Node MCU and relay module which are available at very cheaper cost unless like the Arduino and many other devices.