SMART HOME AUTOMATION

A report submitted in partial fulfillment of the requirements

for the award of

Bachelor of Technology

In

CSO (Computer Science and Engineering (IoT))

By

B. HARICHARAN REDDY
(20BQ1A4910)

R. MAHESH BABU
(20BQ1A4941)

S. SATYA KARTHIK
(20BQ1A4944)

VAIBHAV KAPIL .K
(20BQ1A4951)

SENSORS AND ACTUATORS FOR IoT

(Skill Oriented Course (Course Code: 20CO4C01))

DEPARTMENT OF CSO (Computer Science and Engineering (IoT))

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE and permanently affiliated to JNTUK, Accredited by NBA and NAAC)

NAMBUR (V), PEDAKAKANI (M), GUNTUR-522 508

JUNE 2022

DEPARTMENT OF CSO (Computer Science and Engineering (IoT))

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY: NAMBUR

(AUTONOMOUS)

CERTIFICATE

This is to certify that the Mini Project titled "**SMART HOME AUTOMATION**" is a bonafide record of work done by **B. Haricharan Reddy (20BQ1A4910), R. Mahesh Babu (20BQ1A4941), S. Satya Karthik (20BQ1A4944), Vaibhav Kapil .K (20BQ1A4951)** as part of the Skill Oriented Course **Sensors and Actuators for IoT (Course Code: 20CO4C01)** in partial fulfillment of the requirement of the degree for Bachelor of Technology in CSO (CSE (IoT)) during the academic year 2021–22.

Mr. S. KRISHNA PRASAD
Course Coordinator

DR. CH. V. SURESH
Head of Department

DECLARATION

We, **B. Haricharan Reddy (20BQ1A4910), R. Mahesh Babu (20BQ1A4941), S. Satya Karthik (20BQ1A4944), Vaibhav Kapil .K (20BQ1A4951),** hereby declare that the report entitled “Smart Home Automation” is done by us as part of the Skill Oriented Course **Sensors and Actuators for IoT (Course Code: 20CO4C01)** in partial fulfillment of the requirement of the degree for Bachelor of Technology in CSO (CSE (IoT)) during the academic year 2021–22.

DATE: 15-06-2022
PLACE:VVIT,NAMBUR

SIGNATURE OF THE CANDIDATES :

B. HARICHARAN REDDY (20BQ1A4910)

R. MAHESH BABU (20BQ1A4941)

S. SATYA KARTHIK (20BQ1A4944)

VAIBHAV KAPIL .K (20BQ1A4951)

ACKNOWLEDGEMENT

We thank Dr. Y. Mallikarjuna Reddy, Principal, Vasireddy Venkatadri Institute of Technology, Nambur, for providing us the resources for carrying out the project.

Our sincere thanks to Dr. CH. V. Suresh, Professor and Head, Department of CSO, for his co-operation and guidance which helped us to make our project successful and complete in all aspects.

We also express our sincere thanks and are grateful to our course coordinator Mr. S. Krishna Prasad, Associate Professor, Department of CSO, for motivating us to make our project successful and fully complete. We are grateful for his precious guidance and suggestions.

We also place our floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the project.

NAME OF THE CANDIDATES :

B. HARICHARAN REDDY (20BQ1A4910)

R. MAHESH BABU (20BQ1A4941)

S. SATYA KARTHIK (20BQ1A4944)

VAIBHAV KAPIL .K (20BQ1A4951)

TABLE OF CONTENTS

ABSTRACT

In this busy schedule, everyone needs a bit of comfort and a protected life. The development and transformation of standard homes into smart homes has increased dramatically in recent times. This is due to innovations such as the Internet of Things, Sensors, Mobile Phones, Sophisticated Gadgets, Distributed Computing and PDAs like Amazon Alexa, Google Home, Google Assistant, Apple Siri and Microsoft Cortana. Home automation Framework for remote control of household appliances to relieve stress. This paper represents the general structure of the Home Automation Framework with minimal remote control. This framework gives more control over the well-being of the switches with the low-voltage, Wi-Fi Module NodeMCU-ESP8266, smartphone and relay. The state of the switches is synchronized throughout the control structure, with each user interface constantly displaying the state of the current switches. The framework has been designed to control the family unit's electrical appliances with a general operating convenience plan, a user-friendly interface and easy installation. The remote and smart home environment performs important work in human life and extends its ubiquity due to its adaptability, compactness and little set-up costs. The smart system of home automation is useful in everyday life as it decreases the extraordinary tasks of the users and reduces the worries about the safety of the houses. The framework has been designed to control the family unit's electrical appliances with a general operating convenience plan, a user-friendly interface and easy installation.

1.INTRODUCTION

In this IoT project, I have explained how to make Smart Home with Google Assistant and Alexa using NodeMCU ESP8266 and Sinric Pro. With this NodeMCU ESP8266 project, you can control 3 home appliances with Google Assistant, Alexa, and manual switches. You can also control the relays from Google Home and Amazon Alexa App from anywhere in the world. You can control the relay module from the manual switches if there is no internet available.You don't need any Google Nest or Amazon Echo Dot devices for this voice control home automation project.

Fig 1.1: View of Home Automation System

With this home automation project, you can control & monitor the real-time feedback of the relays in the Google Home and Alexa App from anywhere in the world. If the WiFi is available, the NodeMCU will automatically connect with the Wi-Fi. For this project, I have used all the FREE tools. So if you follow all steps, you can easily make this Smart Home System with Google Home and Amazon Alexa to control the appliances with voice commands. The growth of the Internet of Things will reform a number of sectors, like healthcare, automation energy, transportation, etc. The cloud computing can be used in such case to implement the IoT infrastructure that augmented with sensors and actuators to monitor and control "things" from anywhere.

Even though the technology is developing in our day to day life, there is no help coming into existence for the people who are physically not good on the basis of technology. As the speech enabled, home automation system deploys the use of voice to control the devices. It mainly targets the physically disabled and elderly persons. The home automation will not work if the speech recognition is poor. The speech given by the user will be given as input to the Microphone. Microphone recognizes the speech given by the person and sends it to the recognizing module. It searches for the nearest word even if there are any disturbances in it. If the command (ON/OFF) is given, the action is done. Similarly, the line following robot functions with respect to the speech commands given to it. The line following robot moves forward and backward with the help of sensors and a motor driver board.

The major concept using in the Google assistant-controlled Home automation is the Internet of Things. The Internet of Things (IoT) can be connecting various types of objects like smart phones, personal computer and tablets to the internet, which brings new-fangled type of communication between things and things, and things and people. Even now when technology is handy enough only the well to do people of the society are blessed with their new smart home devices, as these devices costs are a bit high. However, not everyone is wealthy enough to be able to afford a human assistant, or some smart home kit.

Hence, the need for finding an inexpensive and smart assistant for normal families keeps growing. Any man-made objects that can be assigned an IP address and it has the ability to transfer data successfully over a network, the interaction through a network is called IoT. The internet helps us to bring immediate solutions for many problems and able to connect from any of the remote places. The Internets of Things technology is used to come in with innovative idea and large development space for smart homes to improve the living standards of life.

2.HARDWARE AND SOFTWARE REQUIREMENTS

Components Required :

S.No	Name of the Product	Quantity
1	NodeMCU ESP8266	1
2	4 Channel Relay	1
3	Switch Board	1
4	Zero watt Bulbs	3
5	Jumper Wires	As Required

Table 2.1: Components Required

The hardware architecture of this system consists of Node MCU and smartphone. The wireless communication between the smartphone and the Node MCU is done over the Internet. Android OS has a built-in voice recognizing feature named Google assistant which is used to develop a smartphone application which has ability to control the home appliances from user voice command. This application converts the user voice command into text, then it transmit that text message to Adafruit libraries which is connected with Node MCU through Siric website is a website used to create a simple chain of conditional statements called applets.

One advantage of voice-controlled home automation system is that user only pronounce the appliance name in smartphone microphone and telling it to switch ON or OFF the appliances, in this way the users can control home appliance easily without any effort. A voice recognition application provided a user-friendly interface to users and it has ability to add more home appliances into the system. This home automation system can be used in every building using electrical appliances and devices. The main drawback of system is that it is failed to work efficiently in a noisy environment. The main advantage is that its range can be extended as we are using Internet instead of Bluetooth as Bluetooth has the limited range but this solution will not be cost effective.

This complete Home Automation system has the following features:

- Control 3 relays with **Google Assistant, Alexa** , and **switches**.

- Create an account and add devices in **Sinric Pro**
- **Programming** the NodeMCU with Arduino IDE
- Connect Sinric Pro and add IoT devices with **Amazon Alexa App**.
- Connect Sinric Pro and add IoT devices with **Google Home App**.

Sinric Pro FREE account setup :

For this smart house project, we have used the Sinric Pro Free account. First, you have to add 3 devices to the Sinric Pro account. With Sinric Pro, you can easily connect the Google Home and Amazon Alexa App with ESP8266, NodeMCU or ESP32 microcontroller to control any appliance with Google Assistant and Alexa. So, you can easily make any IoT-based home automation project with Sinric Pro.

Create the Sinric Pro Account :

First visit sinric.pro/register

Then enter all the required details and click on Register.

Sinric Pro Login :

After the creating the account, please visit sinric.pro/login Then enter the email id and password, and click on login.

Create a Room in Sinric Pro :

Before adding the devices, first you have to create room in the Sinric Pro.

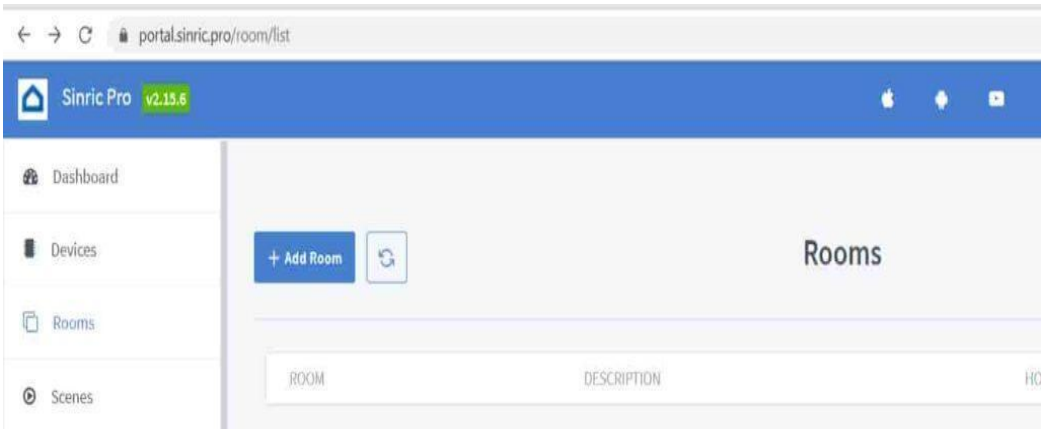

Fig 2.1: Creating a Room

Steps for creating rooms in Sinric Pro:

1. Goto **Rooms** in the left side menu.
2. Click on **Add Room** button.
3. Enter the **Room Name** and **Description**.
4. Click on **Save**.

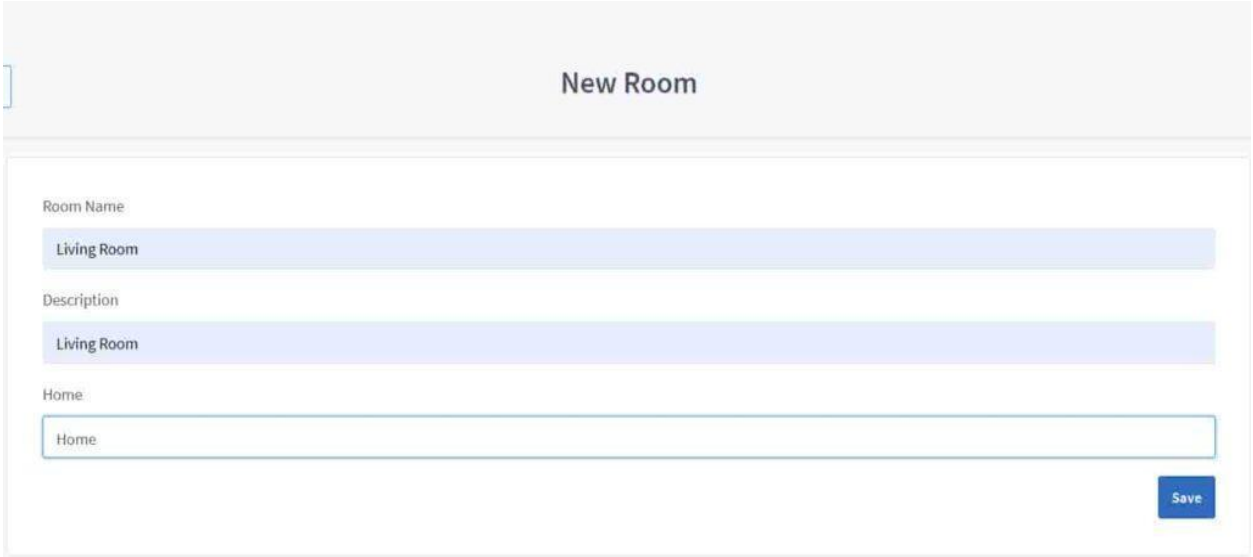

Fig 2.2: Adding a Room

In this way, you can create multiple rooms as per the requirement.

Add Devices in Sinric Pro :

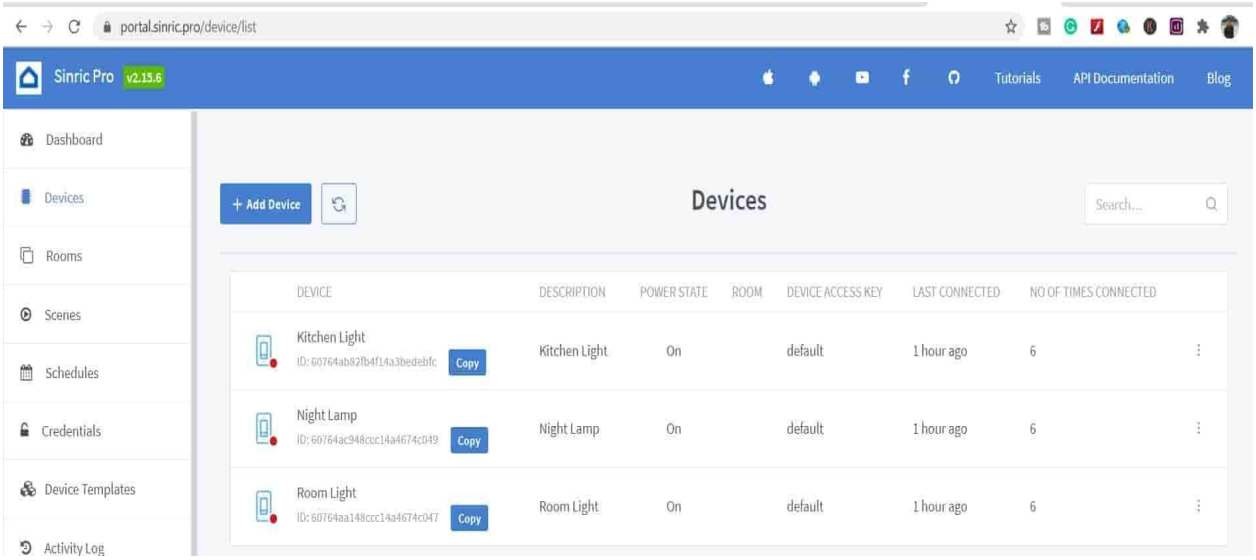

Fig 2.3: Adding Devices

Please follow the following steps to add devices to the Sinric Pro account.

Go to Devices from the left side menu.

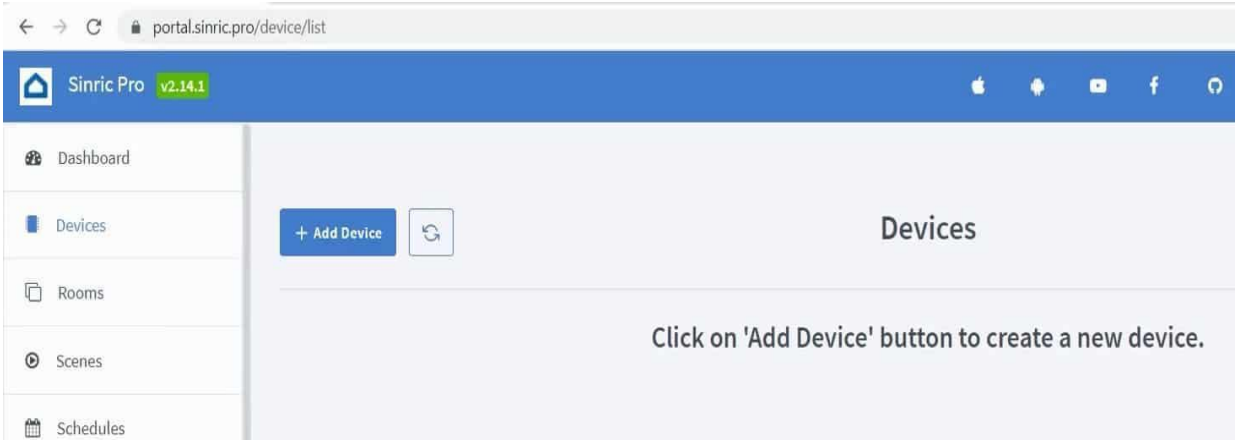

Fig 2.4: Selecting Devices

First select the **Devices** from left side menu, then click on **Add Device** button.

Enter the Device details:

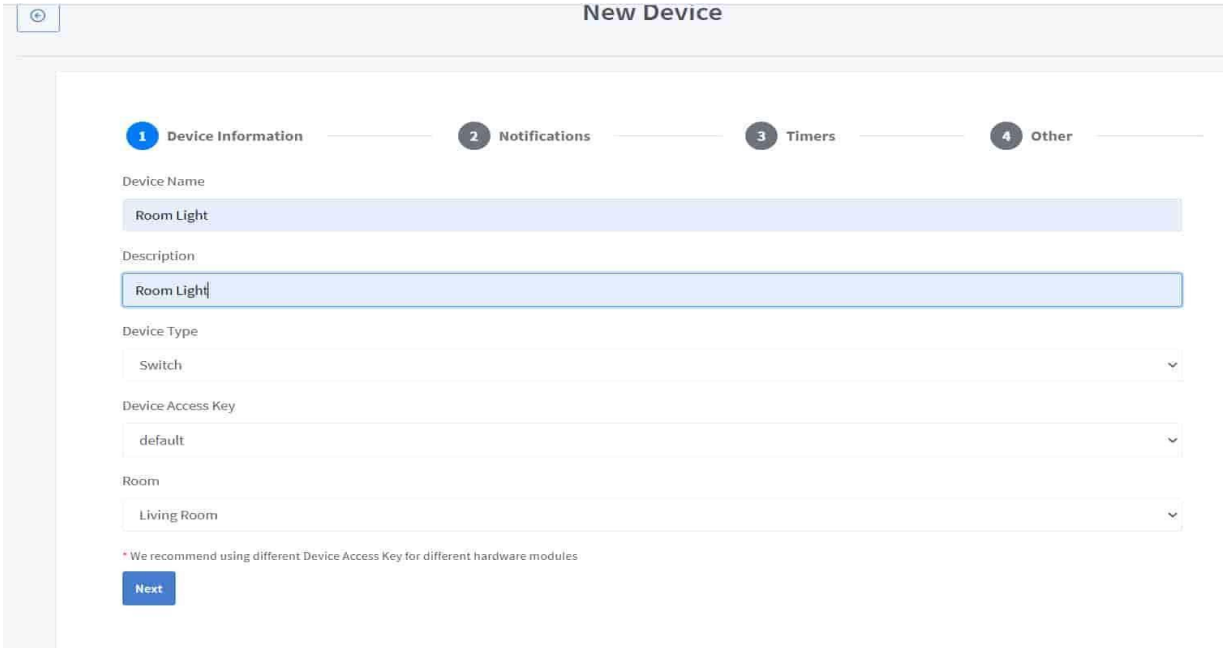

Fig 2.5: Give Device Details

Enter the **Device Name** and **Description**.

- Then select the **Device Type** as per the requirement. Here I have selected device type as Switch, as I will control the SPDT relay.

- Then select the **Room** for the device.

- After that click on **Next**.

Setup Push Notification to the Mobile:

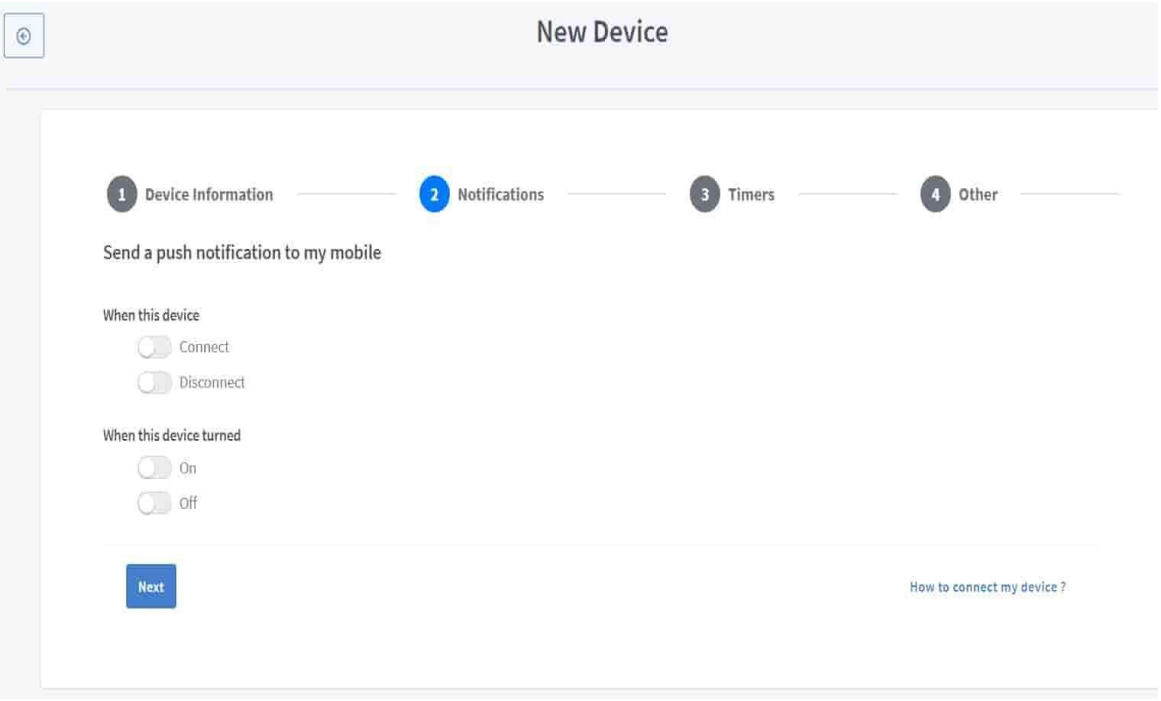

Fig 2.6 : To setup Push Notification to the Mobile

- If you want any **push notifications** related to this device, then you can turn on the notifications.
- This field is optional.
- Click on **Next**.

Sinric API KEY & API SECRET:

Before uploading any example sketch to ESP8266 or ESP32, you have to enter the Sinric API KEY and API SECRET

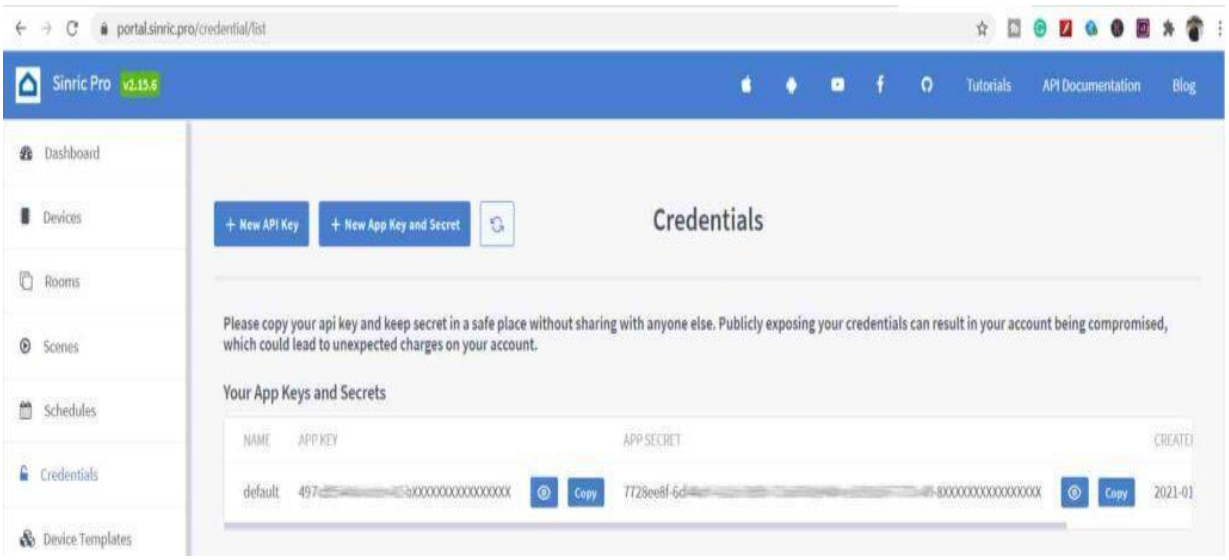

Fig 2.7: Sinric API KEY & API SECRET

To get the API KEY and API SECRET, you have to go to **Credentials** from left side menu.

```
#define WIFI_SSID         "YOUR-WIFI-NAME"
#define WIFI_PASS         "YOUR-WIFI-PASSWORD"
#define APP_KEY           "YOUR-APP-KEY"
#define APP_SECRET        "YOUR-APP-SECRET"
```

- After that, enter the **APP KEY** and **APP SECRET** with Wi-Fi name and Wi-Fi password in the code.
- Also enter the device id in the code. You will find the **Device ID** from Devices menu.
- When you create a device in Sinric Pro, a unique ID assigned to that device. If you create 3 devices, then there will be 3 unique device IDs

Program NodeMCU with Arduino IDE :

1. Update the **Preferences** –> Aditional boards Manager URLs:

   https://dl.espressif.com/dl/package_esp32_index.json,

   http://arduino.esp8266.com/stable/package_esp8266com_index.json

2. Then install the **ESP8266** board from the Board manager
3. Download the required libraries :

- **Sinric Pro** by Boris Jaeger (Download Sinric Pro examples for ESP8266 & ESP32)
- **WebSockets** by Markus Sattler (minimum Version 2.3.5)
- **ArduinoJson** by Benoit Blanchon (minimum Version 6.12.0)

Enter the **APP KEY** and **APP SECRET** with Wi-Fi name and Wi-Fi password in the code.

You can get the **APP KEY** and **APP SECRET** under the **Credentials** menu in Sinric Pro

```
#define WIFI_SSID         "YOUR-WIFI-NAME"
#define WIFI_PASS         "YOUR-WIFI-PASSWORD"
#define APP_KEY           "YOUR-APP-KEY"
#define APP_SECRET        "YOUR-APP-SECRET"
```

Also, enter the device id in the code. You will find the **Device ID** from the Devices menu.

```
//Enter the device IDs here
#define device_ID_1   "SWITCH_ID_NO_1_HERE"
#define device_ID_2   "SWITCH_ID_NO_2_HERE"
#define device_ID_3   "SWITCH_ID_NO_3_HERE"
#define device_ID_4   "SWITCH_ID_NO_4_HERE"
```

**When you add a device in Sinric Pro, a unique ID is assigned to that device. If you create 3 devices, then there will be 3 unique device IDs.

As I have used the free Sinric pro account, so I have entered the 3-device IDs. (Sinric Pro gives 3 devices for FREE)

Here we have explained how to connect Sinric Pro with the **Sinric Pro Alexa Home** Automation to control the appliances with the Alexa.

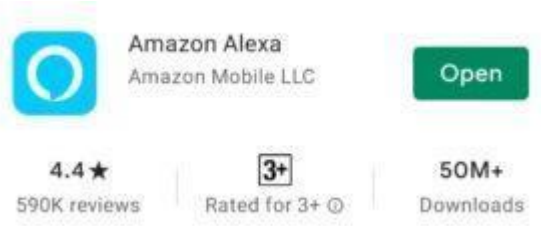

Fig 2.8 : Installation

First, download and install the [Amazon Alexa App] from Google Play Store or App Store.

Connect Sinric Pro with Alexa App :

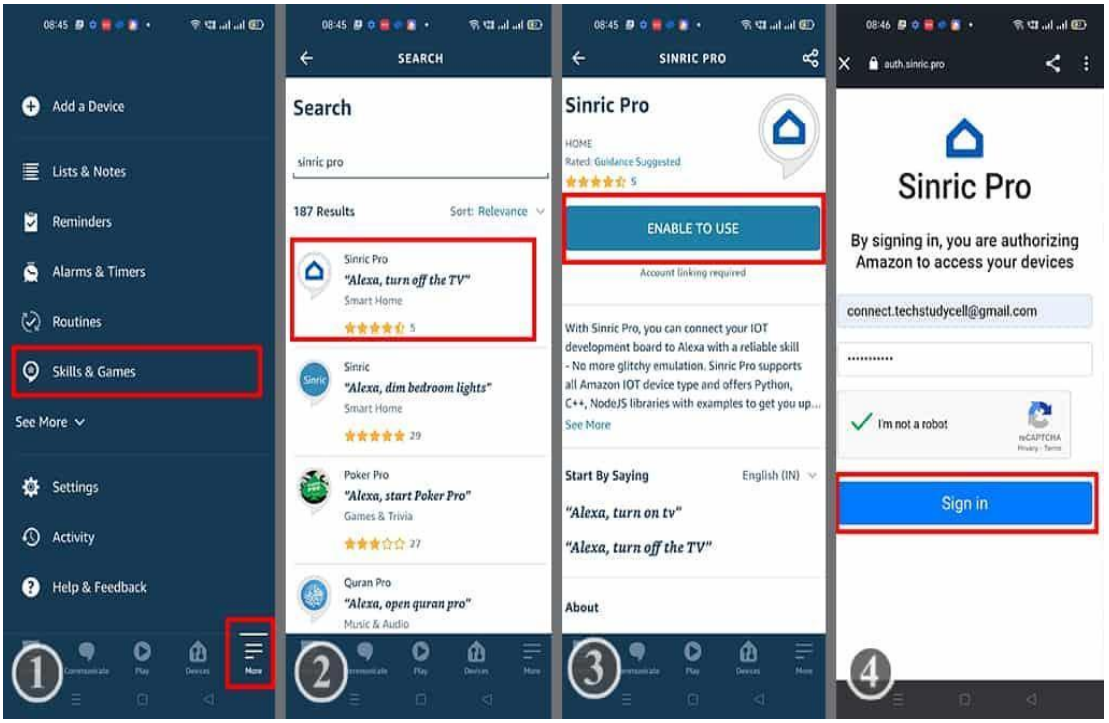

Fig 2.9 : To connect Sinric Pro with Alexa App

Steps to add Sinric Pro with Alexa App:

1. In the Alexa App tap on **More**, then select **Skills & Games**.
2. Search for Sinric Pro, then tap on **Sinric Pro**.
3. Tap on **ENABLE TO USE**.
4. Enter the email id and password used for the Sinric account, Then tap on **Sign in**.

Add Devices in Amazon Alexa App

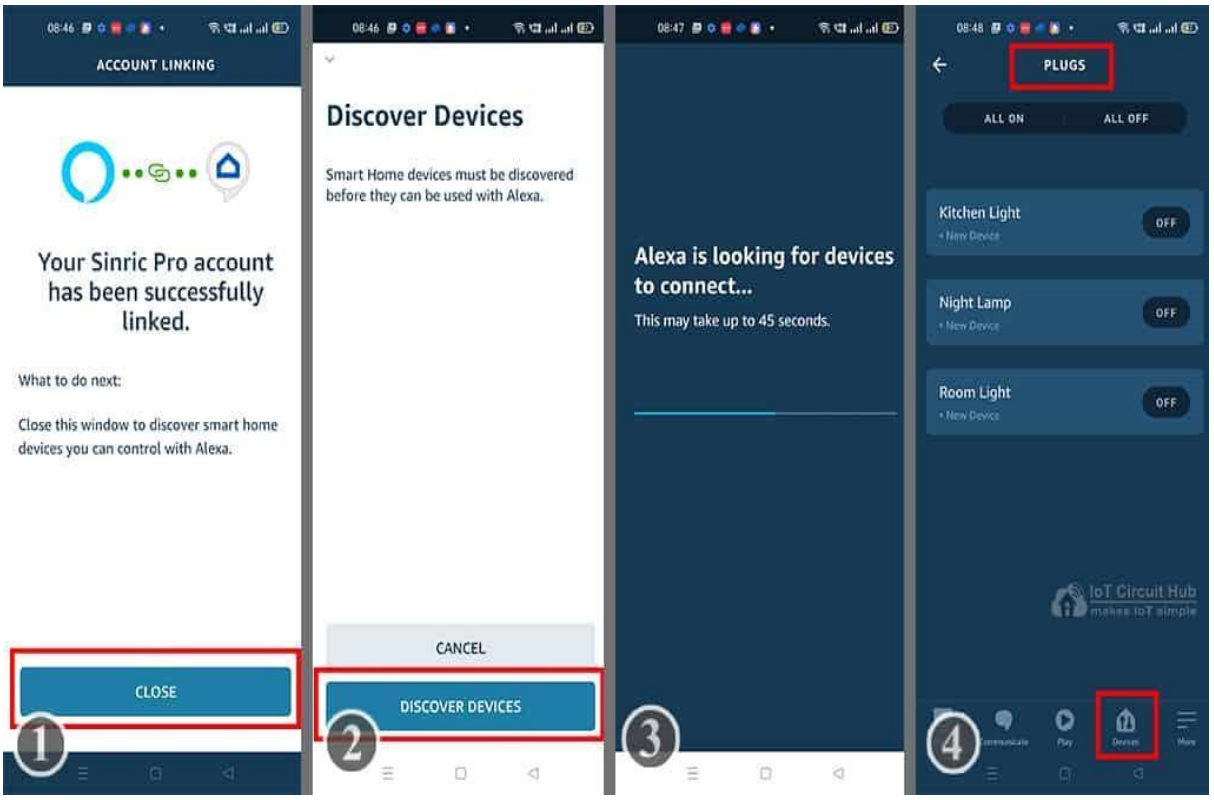

Fig 2.9 : To add devices in Alexa App

After connecting the Sinric Pro account, follow the following steps to add devices.

1. Tap on **CLOSE**.
2. Tap on **DISCOVER DEVICES**.
3. Now, Alexa will look for new devices. This may take some time.
4. After that, go to **Devices**, then select **Plug**. You will find all the connected devices.

Now, if the ESP32 or ESP8266 connected with the Wi-Fi, then you can control the appliances from **Alexa App**.

You can also say "**Alexa**, **Turn ON light**" to control the appliances with voice commands.

You don't need any Alexa devices, like ECHO DOT for this project, You can use Amazon Alexa App to control the appliances.

Here we have explained how to connect Sinric Pro with the Google Home app to control the appliances with the **Google Assistant**.

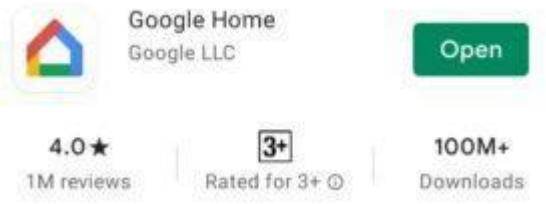

Fig 2.10 : Installation

First, download and install the **Google Home App** from Google Play Store.

Create a new Home in Google Home App :

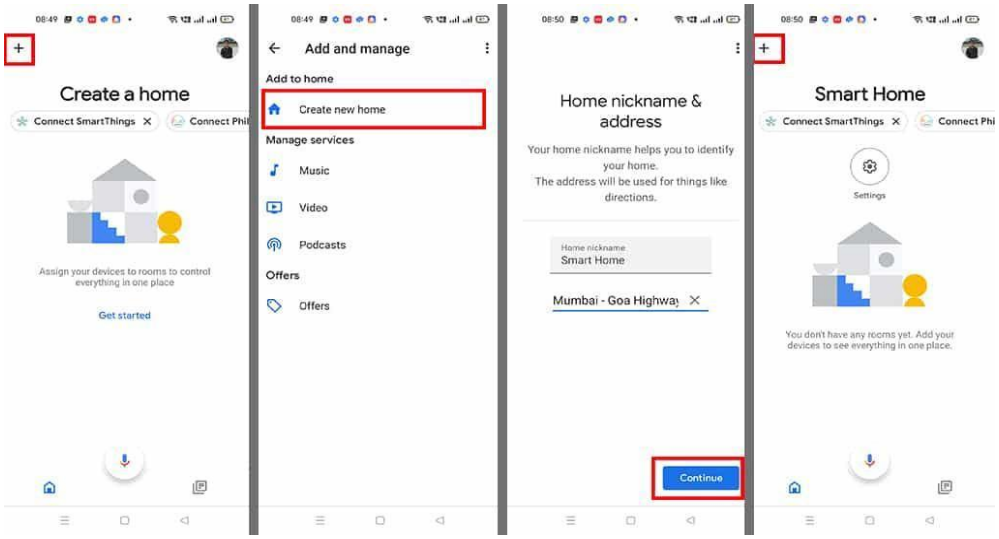

Fig 2.11: To connect Sinric Pro with Google Home App

In the Google Home App, follow the following steps to create a Home:

1. Click on the “+” icon (upper left corner).
2. Tap on **Create new home**.
3. Enter the Home nickname and address. Then tap on **Continue**.
4. The Home is created. Now again tap on the “+” icon to add devices.

After creating the Home in the Google Home app, you can connect the Sinric Pro with Google Home app.

Connect Sinric Pro with Google Home App :

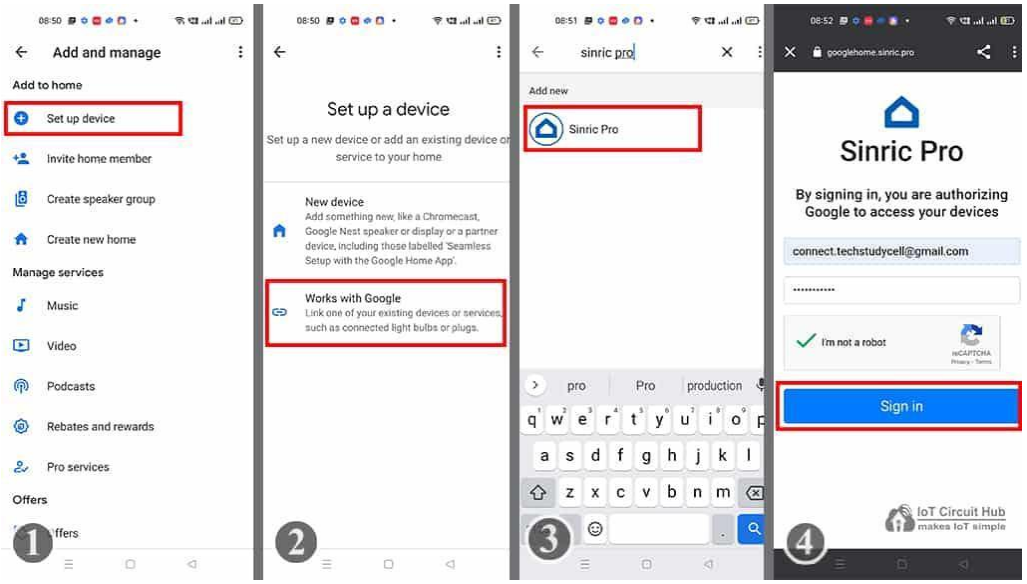

Fig 2.12: To add a new home in Google Home App

Steps to add Sinric Pro with Google Home:

1. Tap on the "+" icon, then select **Set up device**.
2. Tap on **Works with Google**
3. Search for Sinric Pro, then tap on **Sinric Pro**.
4. Enter the email id and password used for the Sinric account, Then tap on **Sign in**.

3. WORKING AND IMPLEMENTATION

Circuit of the NodeMCU Home Automation :

- The circuit is very simple, I have used **D1, D2, D5 & D6** GPIO to control the 4-channel relay module.
- And the GPIO **SD3, D3, D7** & **RX** are connected with manual switches to control the relay module manually.
- I have used the **INPUT_PULLUP** function in Arduino IDE instead of using the pull-up resistors with each switch.
- As per the source code, when the control pins of the relay module receive the **LOW** signal the respective relay will **turn on** and the relay will **turn off** for the **HIGH** signal in the control pin.
- I have used a 5V 2Amp mobile charger to supply the circuit.

- ****The Boot will fail if SD3 and D3 are grounded during the Boot process. So manual switch-S1 and switch-S2 must be OFF during NodeMCU Boot.**
- Now, if you want to use **pushbuttons** then just connect the pushbuttons across the GPIO pins and GND pin instead of switches.

Required Software:

1. Arduino IDE

2. Google Home App

3. Amazon Alexa App

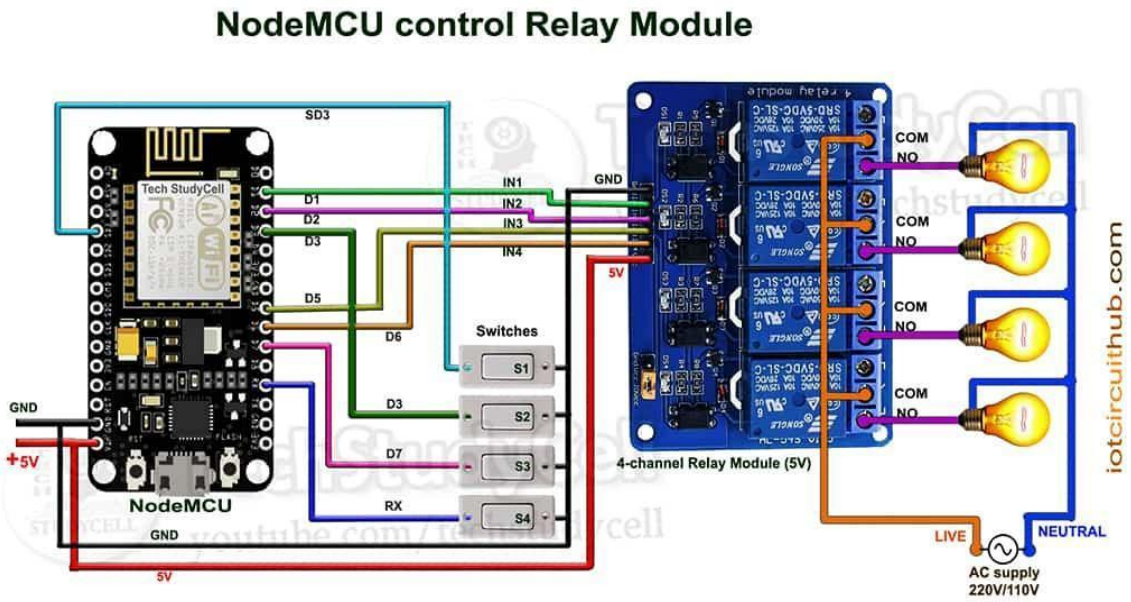

Fig 3.1:Circuit Diagram

NodeMCU control Relays with Alexa App:

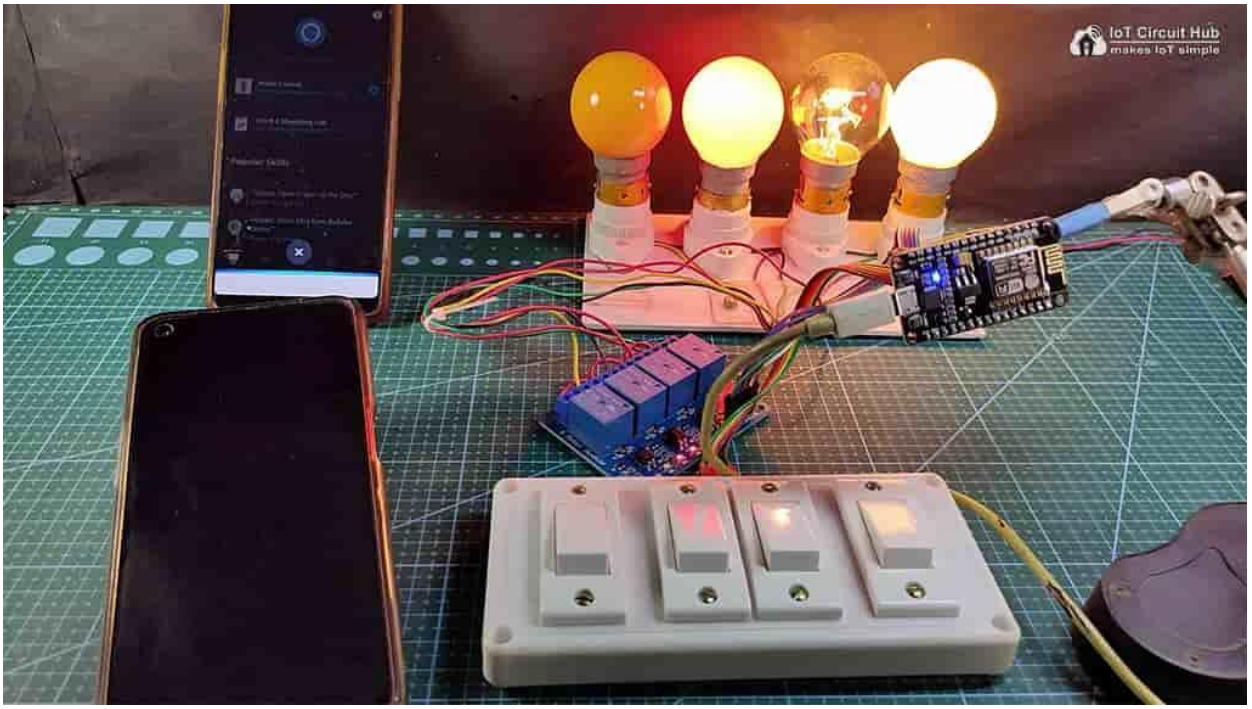

Fig 3.2: NodeMCU control Relays with Alexa App

If the NodeMCU is connected with WiFi, then you can ask Alexa, to turn on the light ["**Alexa, Turn ON Room Light**"]. Thus, you can control the appliances like light, fan, etc

with voice commands using Amazon Alexa App, and also monitor the current status of the switches from anywhere in the world from the Alexa App.

NodeMCU control Relays with Google Assistant :

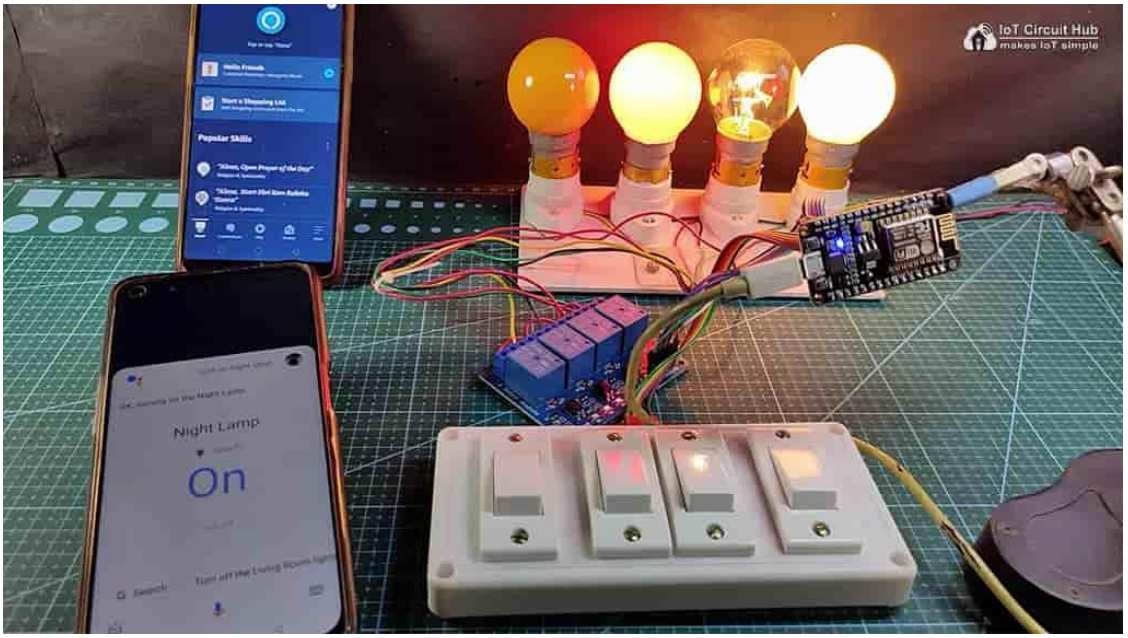

Fig 3.3: NodeMCU control Relays with Google Assistant

You can also ask Google Assistant, to turn on the light [“**Hey Google, Turn ON the Room Light**“]. Thus, you can control the appliances like light, fan, etc with voice commands using Google Assistant, and also monitor the current status of the switches from anywhere in the world from the Google Home App.

Control relays manually with Switches :

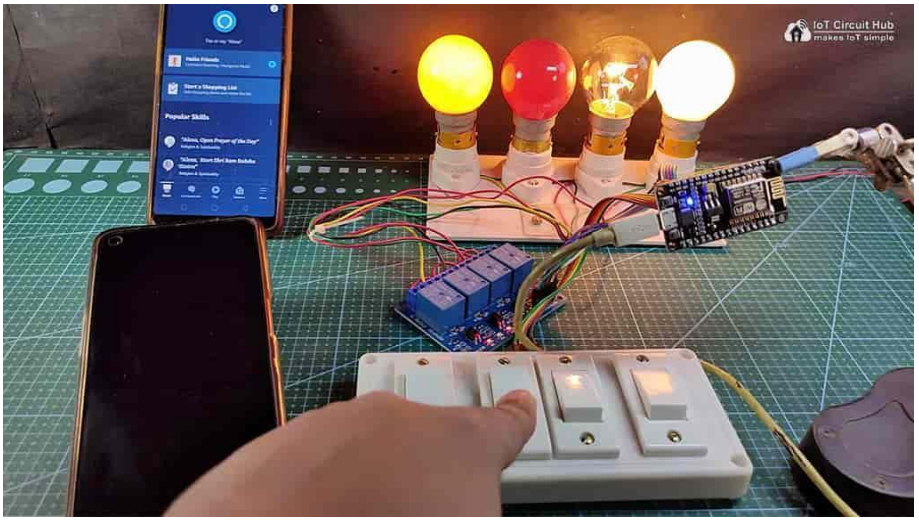

Fig 3.4: Control relays with Switches

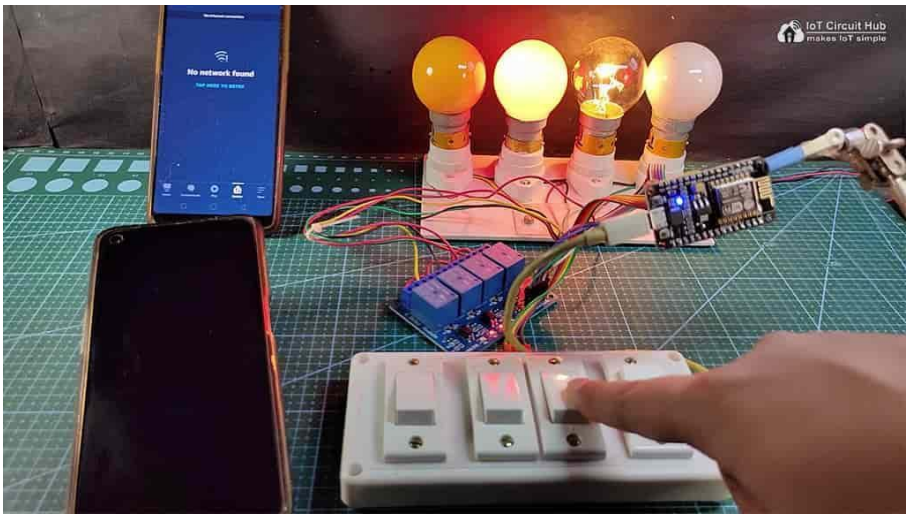

Fig 3.4: No Wi-Fi Control with Switches

- No Wi-Fi Control with Switches
- You can always control the appliances manually with switches or push buttons. and if the NodeMCU is connected with the Wi-Fi, then you can monitor the real-time status in Google Home and Alexa App

GOOGLE ASSISTANT:

The Google Assistant is an Artificial Intelligence based Virtual assistant software which allows its users to control all the apps in their device. It allows the users to control and command most of the apps in their devices using voice commands. This provides more convenience to the people as they only have to command the google assistant thorough voice command. Google Assistant is an artificial intelligence-powered virtual assistant developed by Google that is primarily available on mobile and smart home devices. Unlike the company's previous virtual assistant, Google Now, Google Assistant can engage in two-way conversations. Assistant initially debuted in May 2016 as part of Google's

messaging app Allo, and its voice activated speaker Google Home. After a period of exclusivity on the Pixel and Pixel XL smartphones, it began to be deployed on other Android devices in February 2017, including third-party smartphones and Android Wear (now Wear OS), and was released as a standalone app on the iOS operating system in May 2017.

Alongside the announcement of a software development kit in April 2017, the Assistant has been, and is being, further extended to support a large variety of devices, including cars and third-party smart home appliances. The functionality of the Assistant can also be enhanced by third-party developers. Users primarily interact with Google Assistant through natural voice, though keyboard input is also supported.

In the same nature and manner as Google Now, the Assistant is able to search the Internet, schedule events and alarms, adjust hardware settings on the user's device, and show information from the user's Google account. Google has also announced that the Assistant will be able to identify objects and gather visual information through the device's camera, and support purchasing products and sending money, as well as identifying songs

ARDUINO IDE:

The Arduino integrated development environment (IDE) is a crossplatform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring.

The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

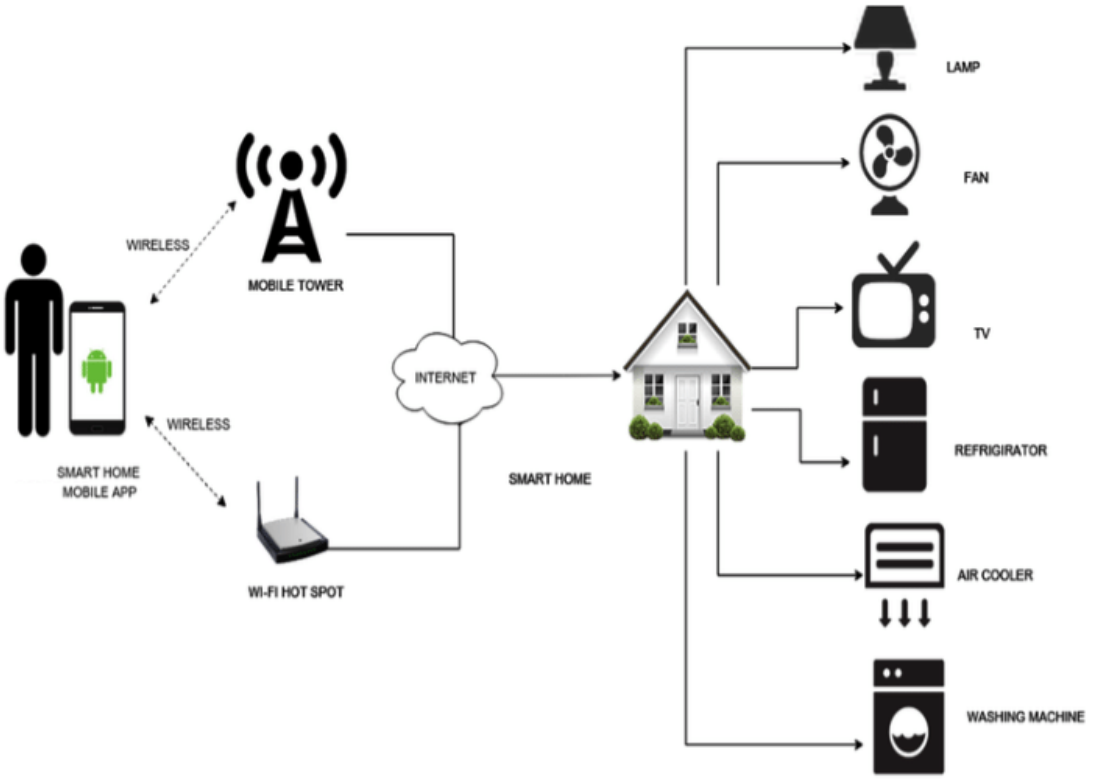

Fig 3.5: Smart Home Automation

4. RESULT AND DISCUSSION

The output for Alexa App controlled Home automation is shown below. Fig 4.1 shows the complete prototype implementation of the proposed system.

Fig 4.1: Smart Home Automation by Alexa

Thus, a software product was designed and created in work, namely an intelligent smart home system. Its architecture has been designed, the infrastructure and the interconnections between the services have been developed. All the services that make up the system and their communication system were created and tested. During the development and testing of the system, the necessary hardware and software were identified, the basic requirements were determined [1-2]. After the result was completed, a control example was analysed, which showed that the system is fully functional and performs all the tasks set before it. Comparing their relevance now and three years ago, we can conclude that such techniques will gain popularity every year, especially regarding the security and safety of their own homes.

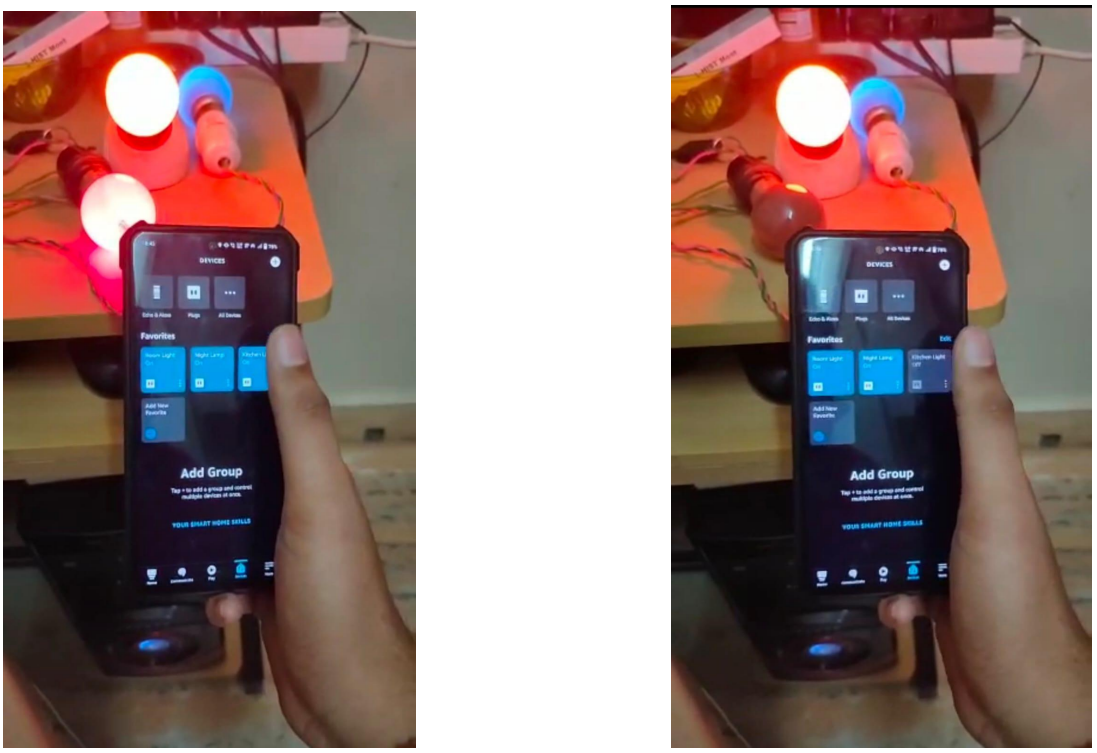

Fig 4.2(a),(b): To ON/OFF with Alexa app

In this project, voice commands are given to the Google assistant. The voice commands for Google assistant have been added through Sinric website and the Sinric account is also linked to it. In this home automation, user have given commands to the Google assistant. Home appliances like Bulb, Fan and Motor etc., are controlled according to the given commands. The commands given through the Google assistant are decoded and then sent to the microcontroller and it control the relays. The device connected to the respective relay turned On or OFF as per the users request to the Google Assistant. The microcontroller used is NodeMCU (ESP8266) and the communication between the microcontroller and the application is established via Wi-Fi (Internet).

5. SUMMARY AND FUTURE SCOPE

There has been tremendous growth in the home automation sector, and many reputed companies utilizing their opportunity to work with Sinric to deliver an elegant way to connect families to their homes. Consumers are looking to secure their home environment in today's unpredictable world, and the new Home automation service gives them the peace of mind that they need to protect their family's well-being. This project is about wireless home automation using Android mobile helps us to implement such a fantastic system in our home at a very reasonable price using cost-effective devices.

Thus, it overcomes many problems like costs, inflexibility, security etc. In addition, will provide greater advantages like it decrease our energy costs, it improves home security. In addition, it is very convenient to use and will improve the comfort of our home. The project has proposed the idea of smart homes that can support a lot of home automation systems. programming language and Node microcontroller have been used to connect the sensors circuit to the home.

Also, in home and building automation systems, the use of wireless technologies gives several advantages which cannot be achieved by using a wired network.

1) Reduced installation costs.

2) Easy deployment, installation, and coverage.

3) System scalability and easy extension.

4) Aesthetical benefits.

5) Integration of mobile devices.

For all these reasons, wireless technology is not only an attractive choice in renovation and refurbishment, but also for new installations.

APPLICATIONS :

• Lighting control system

• Appliance control with a smart grid

• Indoor positioning systems

• Home automation for elderly and disabled people

FUTURE SCOPE :

There are a variety of enhancements that could be made to this system to achieve greater accuracy in sensing and detection.

a) There are a lot of other sensors that can be used to increase the security and control of the home like pressure sensor that can be put outside the home to detect that someone will enter the home.

b) Changing the way of the automated notifications by using the GSM module to make this system more professional.

c) A smart garage that can measure the length of the car and choose which block to put the car into it and it will navigate the car through the garage to make the parking easy for the homeowner in his garage.

In this project, voice commands are given to the Google assistant. The voice commands for Google assistant have been added through Sinric website and the Sinric account is also linked to it. In this home automation, user have given commands to the Google assistant. Home appliances like Bulb, Fan and Motor etc., are controlled according to the given commands. The commands given through the Google assistant are decoded and then sent

to the microcontroller and it control the relays. The device connected to the respective relay turned On or OFF as per the users request to the Google Assistant.

The microcontroller used is NodeMCU (ESP8266) and the communication between the microcontroller and the application is established via Wi-Fi (Internet). There has been tremendous growth in the home automation sector, and many reputed companies utilizing their opportunity to work with Sinric to deliver an elegant way to connect families to their homes. Consumers are looking to secure their home environment in today's unpredictable world.

The new Home automation service gives them the peace of mind that they need to protect their family's well-being. This project is about wireless home automation using Android mobile helps us to implement such a fantastic system in our home at a very reasonable price using cost-effective devices. Thus, it overcomes many problems like costs, inflexibility, security etc. In addition, will provide greater advantages like it decrease our energy costs, it improves home security. In addition, it is very convenient to use and will improve the comfort of our home. The project has proposed the idea of smart homes that can support a lot of home automation systems. For all these reasons, wireless technology is not only an attractive choice in renovation and refurbishment, but also for new installations.

REFERENCES

[1]. IoT Application Development: Home Security System, Raj G Anvekar School of Electronics and Communication Engineering, KLE Technological University Hubli, India, 2017 IEEE International Conference on Technological Innovations in ICT For Agriculture and Rural Development (TIAR 2017) 978 [2]. Method and Device to Communicate via SMS after a Security Intrusion ,Karri V and Daniel Lim J S 1st International Conf. on Sensing Technology Palmerston North New Zealand2005 21-23 [3]. Design and Implementation of Security for Smart Home based on GSM technology, Jayashri B and Arvind S 2013 International Journal of Smart Home 7 201-08 [4]. Design and Development of Activation and Controlling Of Home Automation System VIA SMS through Microcontroller, Ch.Naga Koti Kumar,www.ijera.com, vol.2, issue2,Marc-apr 2012, pp.1349-1352. [5]. Home Automation and Security System UsingAndroid ADK, Deepali Javale, Mohd. Mohsin, Shreerang Nandanwar, International Journal of Electronics Communication and Computer Technology (IJECCT) Volume 3 Issue 2 (March 2013). [6]. Vaishnavi S Gunge and Prathibha S Yalagi, Smart Home Automation, IJCA-0975-8887, National seminar on Recent Trends in Dtamining RTDM 2016. [7]. Swati Tiwari and Rahul Gedam, Review paper on Home Automation System based on IoT, International Research Journel of Engineering Technology, eISSN:2395-0056, p-ISS:2395-0012, volume:03. Issue:05/May 2016. [8]. Prof. B.P Kulkarni, Aniket V Joshi, Vaibhav V Jadhar and Akshay kumar T Dhamange, IoT Based Home Automation using RPI, International Journal of Innovative Studies in Sciences and Engineering Technology, ISSN:2455-4863, volume:3, Issue:4, April 2017. [9]. D. Norris, The Internet of Things: Do-ItYourself at Home Projectsfor Arduino, Raspberry Pi and BeagleBone Black. Tab Electronics, 2015. [10]. D. Giusto, A. lera, G. Morabito, and L. Atzori, The Internet of Things. New York, NY:Springer New York, 2010. [11]. A. Z. Alkar and U. Buhur , An Internet based wireless home automation system for multifunctional devices, IEEE Trans. Consum. Electron., vol.51, no.4, pp. 1169-1174, Nov.2005. [12]. R. Piyare. M. Tazil, Bluetooth Based Home Automation and Security System using Cellphone, 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), volume.02. Issue:04, August-2011. [13]. S.Anusha1, M.Madhavi, R.Hemalatha,, „Home Automation Using ATmega328 Microcontroller and Android Application‟‟,International Research Journal of Engineering and Technology (IRJET),Volume: 02,Issue: 06,Sep-2015

APPENDIX

Source CODE:

```
// Uncomment the following line to enable serial debug output
//#define ENABLE_DEBUG

#ifdef ENABLE_DEBUG
      #define DEBUG_ESP_PORT Serial
      #define NODEBUG_WEBSOCKETS
      #define NDEBUG
#endif

#include <Arduino.h>
#include <ESP8266WiFi.h>
#include "SinricPro.h"
#include "SinricProSwitch.h"

#include <map>

#define WIFI_SSID         "YOUR-WIFI-NAME"
#define WIFI_PASS         "YOUR-WIFI-PASSWORD"
#define APP_KEY           "YOUR-APP-KEY"      // Should look like
"de0bxxxx-1x3x-4x3x-ax2x-5dabxxxxxxxx"
#define APP_SECRET        "YOUR-APP-SECRET"   // Should look like
"5f36xxxx-x3x7-4x3x-xexe-e86724a9xxxx-4c4axxxx-3x3x-x5xe-x9x3-333d65xxxxxx"

//Enter the device IDs here
#define device_ID_1   "SWITCH_ID_NO_1_HERE"
#define device_ID_2   "SWITCH_ID_NO_2_HERE"
#define device_ID_3   "SWITCH_ID_NO_3_HERE"
#define device_ID_4   "SWITCH_ID_NO_4_HERE"

// define the GPIO connected with Relays and switches
#define RelayPin1 5  //D1
#define RelayPin2 4  //D2
#define RelayPin3 14 //D5
#define RelayPin4 12 //D6

#define SwitchPin1 10  //SD3
#define SwitchPin2 0   //D3
#define SwitchPin3 13  //D7
#define SwitchPin4 3   //RX

#define wifiLed   16   //D0

// comment the following line if you use a toggle switches instead of tactile buttons
//#define TACTILE_BUTTON 1

#define BAUD_RATE   9600

#define DEBOUNCE_TIME 250

typedef struct {      // struct for the std::map below
```

```
  int relayPIN;
  int flipSwitchPIN;
} deviceConfig_t;


// this is the main configuration
// please put in your deviceId, the PIN for Relay and PIN for flipSwitch
// this can be up to N devices...depending on how much pin's available on your device ;)
// right now we have 4 devicesIds going to 4 relays and 4 flip switches to switch the relay manually
std::map<String, deviceConfig_t> devices = {
    //{deviceId, {relayPIN,  flipSwitchPIN}}
    {device_ID_1, {  RelayPin1, SwitchPin1 }},
    {device_ID_2, {  RelayPin2, SwitchPin2 }},
    {device_ID_3, {  RelayPin3, SwitchPin3 }},
    {device_ID_4, {  RelayPin4, SwitchPin4 }}
};

typedef struct {      // struct for the std::map below
  String deviceId;
  bool lastFlipSwitchState;
  unsigned long lastFlipSwitchChange;
} flipSwitchConfig_t;

std::map<int, flipSwitchConfig_t> flipSwitches;    // this map is used to map flipSwitch PINs to
deviceId and handling debounce and last flipSwitch state checks
                                    // it will be setup in "setupFlipSwitches" function, using informations
from devices map

void setupRelays() {
  for (auto &device : devices) {          // for each device (relay, flipSwitch combination)
    int relayPIN = device.second.relayPIN; // get the relay pin
    pinMode(relayPIN, OUTPUT);            // set relay pin to OUTPUT
    digitalWrite(relayPIN, HIGH);
  }
}

void setupFlipSwitches() {
  for (auto &device : devices)  {                  // for each device (relay / flipSwitch combination)
    flipSwitchConfig_t flipSwitchConfig;             // create a new flipSwitch configuration

    flipSwitchConfig.deviceId = device.first;        // set the deviceId
    flipSwitchConfig.lastFlipSwitchChange = 0;       // set debounce time
    flipSwitchConfig.lastFlipSwitchState = true;     // set lastFlipSwitchState to false (LOW)--

    int flipSwitchPIN = device.second.flipSwitchPIN;  // get the flipSwitchPIN

    flipSwitches[flipSwitchPIN] = flipSwitchConfig;   // save the flipSwitch config to flipSwitches map
    pinMode(flipSwitchPIN, INPUT_PULLUP);                  // set the flipSwitch pin to INPUT
  }
}

bool onPowerState(String deviceId, bool &state)
{
  Serial.printf("%s: %s\r\n", deviceId.c_str(), state ? "on" : "off");
  int relayPIN = devices[deviceId].relayPIN; // get the relay pin for corresponding device
  digitalWrite(relayPIN, !state);            // set the new relay state
  return true;
}

void handleFlipSwitches() {
```

```
  unsigned long actualMillis = millis();                                  // get actual millis
  for (auto &flipSwitch : flipSwitches) {                                 // for each flipSwitch in flipSwitches
map
    unsigned long lastFlipSwitchChange = flipSwitch.second.lastFlipSwitchChange;  // get the
timestamp when flipSwitch was pressed last time (used to debounce / limit events)

    if (actualMillis - lastFlipSwitchChange > DEBOUNCE_TIME) {            // if time is > debounce
time...

      int flipSwitchPIN = flipSwitch.first;                               // get the flipSwitch pin from
configuration
      bool lastFlipSwitchState = flipSwitch.second.lastFlipSwitchState;   // get the
lastFlipSwitchState
      bool flipSwitchState = digitalRead(flipSwitchPIN);                  // read the current flipSwitch
state
      if (flipSwitchState != lastFlipSwitchState) {                       // if the flipSwitchState has
changed...
#ifdef TACTILE_BUTTON
        if (flipSwitchState) {                                            // if the tactile button is pressed
#endif
          flipSwitch.second.lastFlipSwitchChange = actualMillis;          // update
lastFlipSwitchChange time
          String deviceId = flipSwitch.second.deviceId;                   // get the deviceId from config
          int relayPIN = devices[deviceId].relayPIN;                      // get the relayPIN from config
          bool newRelayState = !digitalRead(relayPIN);                    // set the new relay State
          digitalWrite(relayPIN, newRelayState);                          // set the trelay to the new state

          SinricProSwitch &mySwitch = SinricPro[deviceId];                // get Switch device from
SinricPro
          mySwitch.sendPowerStateEvent(!newRelayState);                   // send the event
#ifdef TACTILE_BUTTON
        }
#endif
        flipSwitch.second.lastFlipSwitchState = flipSwitchState;          // update
lastFlipSwitchState
      }
    }
  }
}

void setupWiFi()
{
  Serial.printf("\r\n[Wifi]: Connecting");
  WiFi.begin(WIFI_SSID, WIFI_PASS);

  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.printf(".");
    delay(250);
  }
  digitalWrite(wifiLed, LOW);
  Serial.printf("connected!\r\n[WiFi]: IP-Address is %s\r\n", WiFi.localIP().toString().c_str());
}

void setupSinricPro()
{
  for (auto &device : devices)
  {
    const char *deviceId = device.first.c_str();
    SinricProSwitch &mySwitch = SinricPro[deviceId];
```

```
    mySwitch.onPowerState(onPowerState);
  }

  SinricPro.begin(APP_KEY, APP_SECRET);
  SinricPro.restoreDeviceStates(true);
}

void setup()
{
  Serial.begin(BAUD_RATE);

  pinMode(wifiLed, OUTPUT);
  digitalWrite(wifiLed, HIGH);

  setupRelays();
  setupFlipSwitches();
  setupWiFi();
  setupSinricPro();
}

void loop()
{
  SinricPro.handle();
  handleFlipSwitches();
}
```