

Smart Home with Google Assistant & Alexa using NodeMCU ESP8266, Arduino UNO & Sinric Pro

Make a Smart Home with Google Assistant & Alexa using NodeMCU ESP8266, Arduino Uno & Sinric Pro. Explained with Circuit and CODE & used FREE tools.

This Project Make In

GOVERNMENT POLYTECHNIC BANKA

NAME	POSITION	MOBILE NO.
Prof. BIRBAL KUMAR RAJAK	HEAD OF DEPARTMENT IN ELECTRONICS BRANCH	9097257911
Prof. MUNESWAR KUMAR	MENTOR	9907599797
PRASHANT KUMAR	CO-MENTOR	9631258587

TEAM PARTICIPANTS NAME

NAME	ROLL NO	SESSION	BRANCH
SUSHANT SAGAR	511402120020	2020-23	ELECTRONICS
ADITYA BHARTI	511402120301	2020-23	ELECTRONICS
GOURAV KUMAR	511402120602	2020-23	ELECTRONICS
ANKIT KUMAR	511402120302	2020-23	ELECTRONICS
SUJIT KUMAR	511402120028	2020-23	ELECTRONICS
SIDDHANT KUMAR	511402120016	2020-23	ELECTRONICS

Table of Contents

1. Introduction	03
2. SPDT Switch Uses	05
3. Circuit of the NodeMCU Home Automation	06
4. Required Component for the NodeMCU Project	07
5. Sinric Pro FREE Account Setup	08
6. Program NodeMCU With Arduino IDE	14
7. Connect Sinric Pro with Alexa App	16
8. Add Devices In Amazon Alexa App	17
9. Create a new Home in Google Home App	18
10.Connect Sinric Pro with Google Home App	19
11.Control Devices With Goggle Assistant or Google Home	20
12.Get The IR Codes (HEX Code) From Remote	21
13.Program Arduino UNO With Arduino IDE	22
14.NodeMCU Control Realys With Alexa	23
15.NodeMCU Control Realys With Goggle Assistant	24
16.Control Relays Manually With Switches	25
17.Finally !! the Sinric Pro Home Automation system Is Ready	25

INTRODUCTION

- In this IoT project, I have explained how to make Smart Home with Google Assistant and Alexa using NodeMCU ESP8266 , Arduino 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.
- And you don't need any Google Nest or Amazon Echo Dot devices for this voice control home automation project.



- 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.
- This Smart Home Automation system has the following features:
 1. Control appliances with Sinric Dashboard.
 2. Control appliances with Alexa.
 3. Control appliances with Alexa App.
 4. Control appliances with Goggle Assiant.
 5. Control appliances with Google Home App.
 6. Used Arduino EEPROM to remember previous states.
 7. Control appliances with IR remote.
 8. Control appliances manually with switches.
 9. Control home appliances manually without internet.
 10. All resources used for this project are FREE.





SPDT Switch Uses

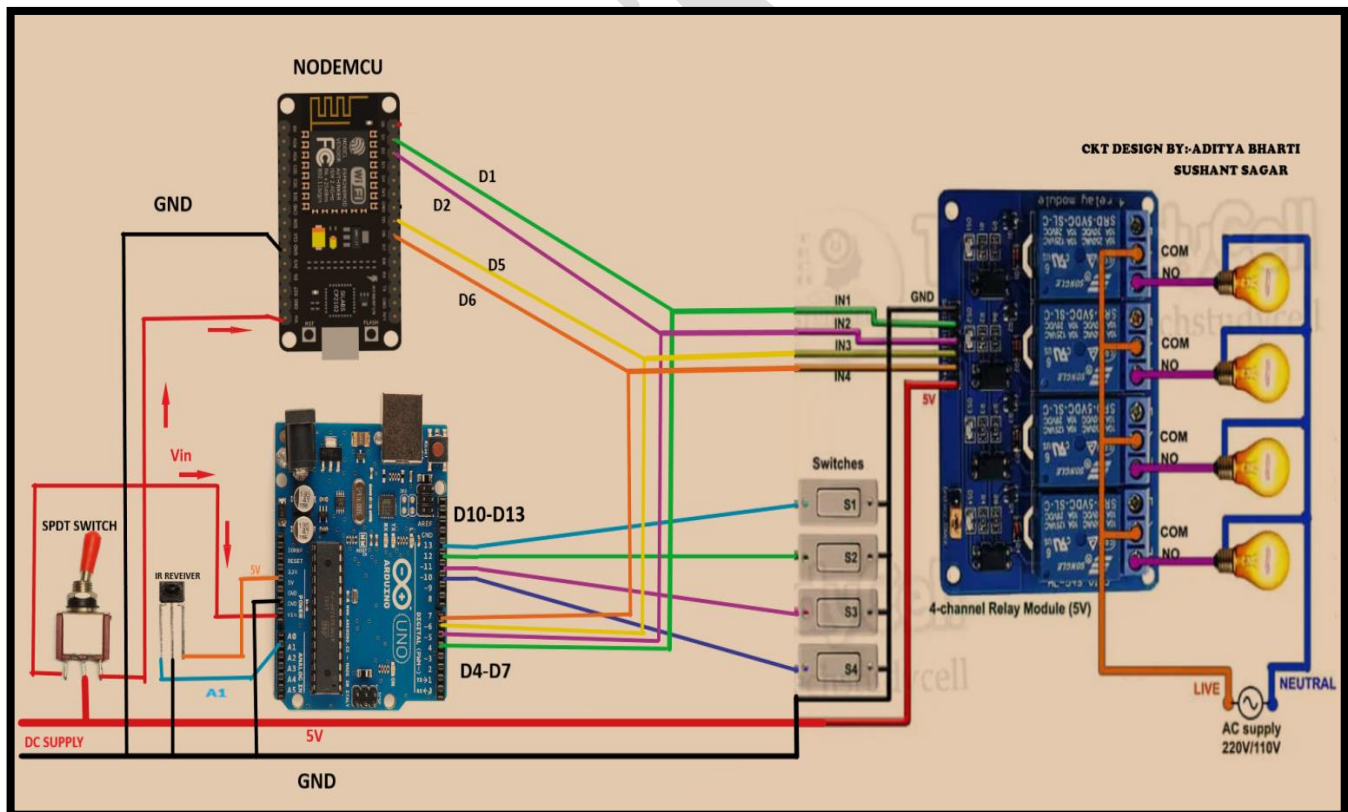
An SPDT switch consists of three terminals and connects the source terminal and one of two output terminals. An SPDT switch allows for an "ON/ON" configuration, which means the switch's input terminal is always completing one of the two possible circuits that the switch controls.

In this Project Use SPDT Switch as a Control Power NodeMCU & Arduino UNO Boards.

At a Time One Board Power Is ON.

- If NodeMCU Board Power Is ON, then
 1. We can Control The Appliance With Alexa
 2. We can Control The Appliance With Alexa App
 3. We can Control The Appliance With Google Assistant
 4. We can Control The Appliance With Goggle Home
- If Arduino UNO Board Power Is ON, then
 1. We can Control The Appliance With IR Remote
 2. We can Control The Appliance With Manually

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.

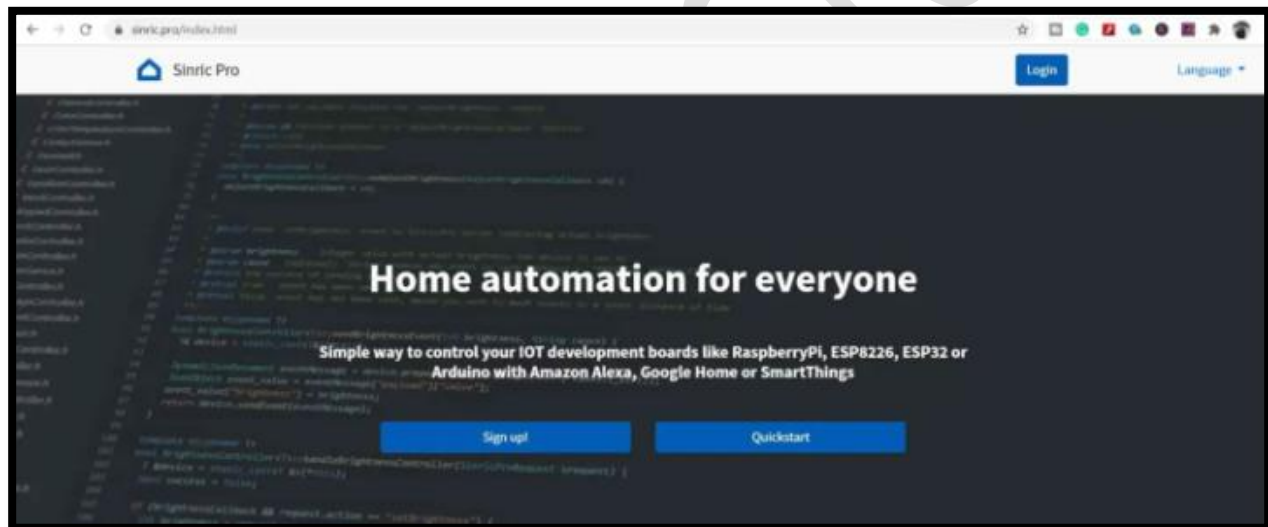
Required Components for the NodeMCU projects

1. NodeMCU
2. 4-channel 5V SPDT Relay Module
3. Manual Switches
4. SPDT Switch
5. Arduino UNO Board
6. IR Receiver Sensor
7. 5V Charger or DC Power Source
8. Connecting Wires

9. 220V Bulb
10. 5 Pin Charging Socket
11. Electricity Board
12. Amazon Echo Dot (optional)
13. Google Nest Mini (optional)

Sinric Pro FREE Account Setup

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. As per their current plan (13th, March 2023) you will get 3 devices for free.

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.

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**.



New Room

Room Name
Living Room

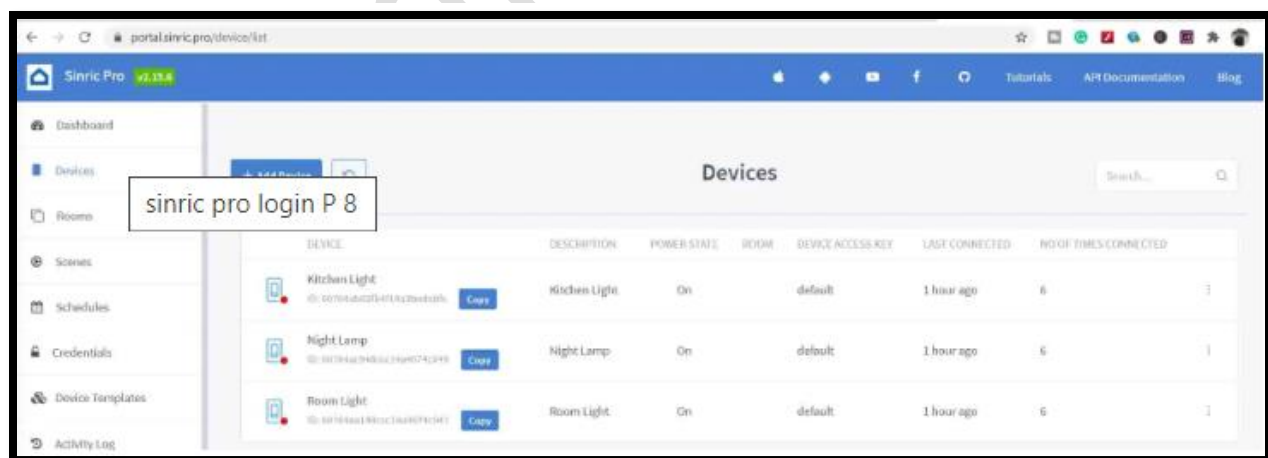
Description
Living Room

Home
Home

Save

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

Add Devices in Sinric Pro

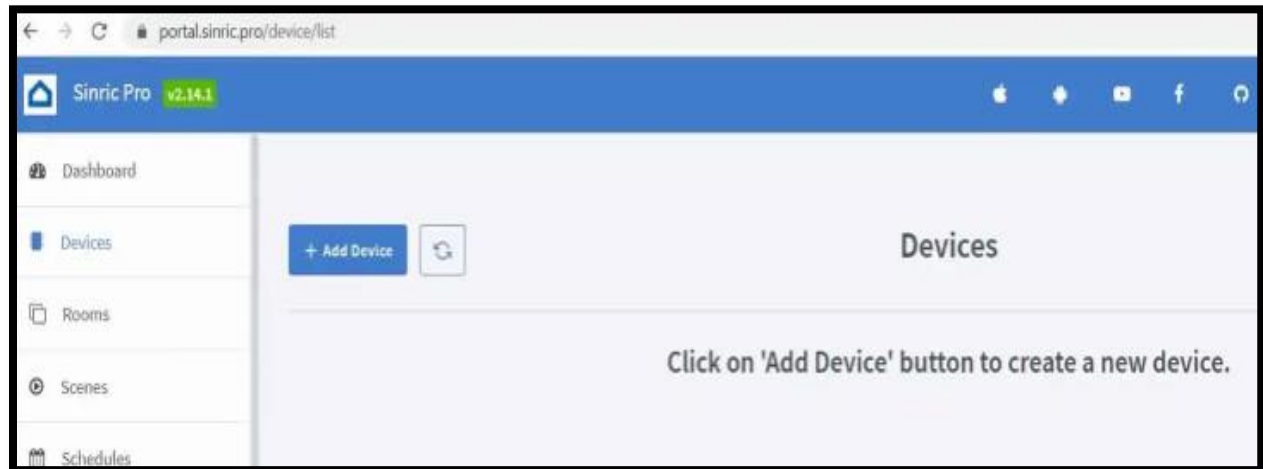


Please follow the following steps to add devices to the Sinric Pro account.

Go to Devices from the left side menu.

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

Enter the Device details:



1 Device Information 2 Notifications 3 Timers 4 Other

Device Name
Room Light

Description
Room Light

Device Type
Switch sinric pro loginl P 12

Device Access Key
default

Room
Living Room

* We recommend using different Device Access Key for different hardware modules

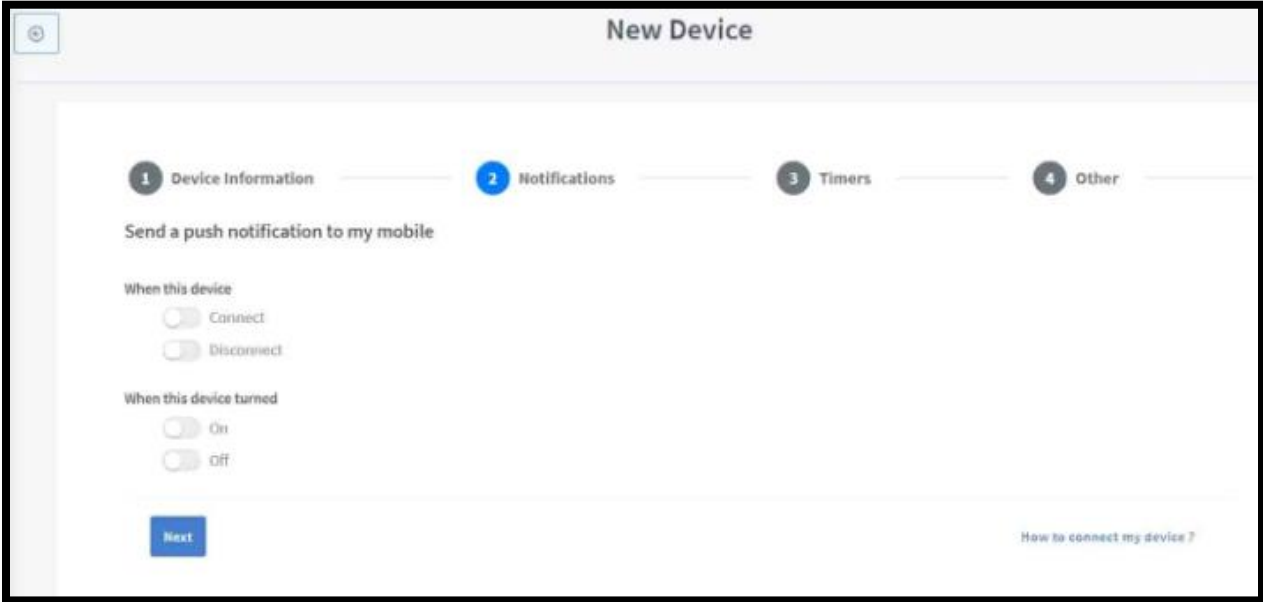
Next

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



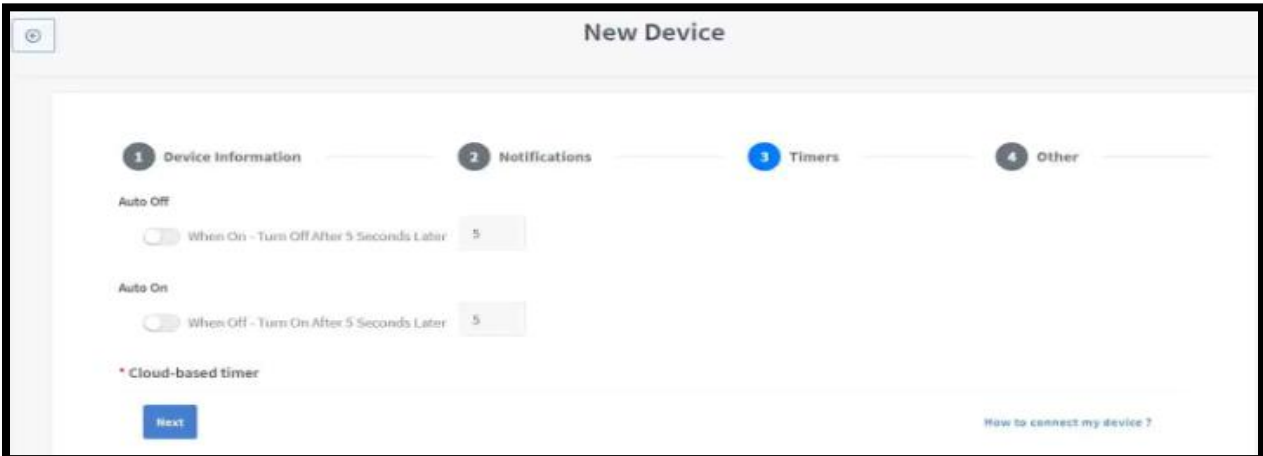
The screenshot shows the 'New Device' setup interface. At the top, there's a progress bar with four steps: 1. Device Information, 2. Notifications (highlighted in blue), 3. Timers, and 4. Other. Below the progress bar, the text 'Send a push notification to my mobile' is displayed. Underneath, there are two sections of toggle switches. The first section, 'When this device', has two toggles: 'Connect' and 'Disconnect'. The second section, 'When this device turned', has two toggles: 'On' and 'Off'. At the bottom left, there is a blue 'Next' button. At the bottom right, there is a link that says 'How to connect my device?'. The entire interface is enclosed in a light gray border.

If you want any **push notifications** related to this device, then you can turn on the notifications.

This field is optional.

Click on **Next**.


Set up Timers



The screenshot shows the 'New Device' setup interface, now at Step 3: Timers. The progress bar at the top shows steps 1. Device Information, 2. Notifications, 3. Timers (highlighted in blue), and 4. Other. Below the progress bar, there are two sections of toggle switches. The first section, 'Auto Off', has a toggle switch labeled 'When On - Turn Off After 5 Seconds Later' with a value of '5' in a text box. The second section, 'Auto On', has a toggle switch labeled 'When Off - Turn On After 5 Seconds Later' with a value of '5' in a text box. Below these, there is a section labeled '* Cloud-based timer' with a text input field. At the bottom left, there is a blue 'Next' button. At the bottom right, there is a link that says 'How to connect my device?'. The entire interface is enclosed in a light gray border.

If you want to set any timer to Auto On or Auto Off the device, after the predefined time, then you the setup here. Again this field is optional. Click on Next.

Set Up Energy Usage (Estimation)



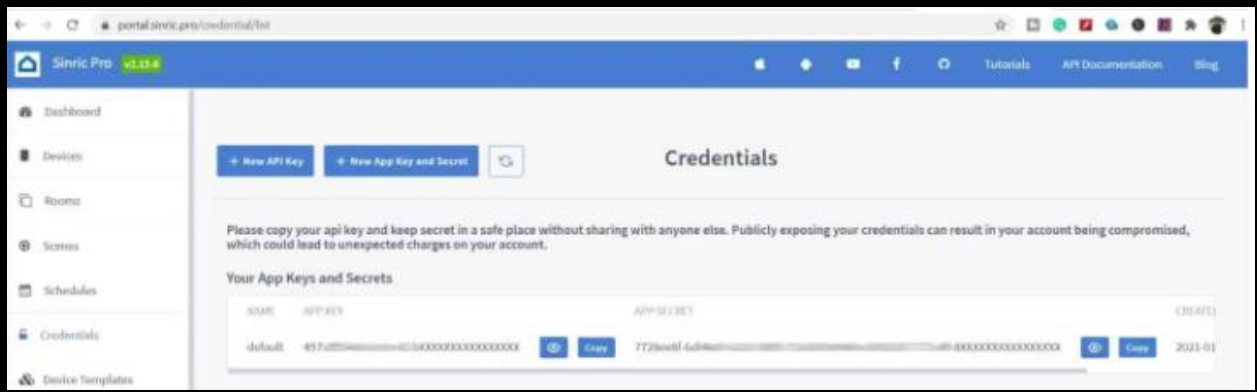
The screenshot shows the 'New Device' setup interface, specifically the 'Other' step (4). The 'Energy Usage (Estimation)' section is highlighted. It includes a toggle switch for enabling energy estimation. Below the toggle are two input fields: 'Stand by Wattage (W)' and 'On Wattage (W)'. A 'Save' button is located at the bottom left of the form. The top navigation bar shows steps 1 (Device Information), 2 (Notifications), 3 (Timers), and 4 (Other).

This is another optional field. You can define the wattage rating of the connected appliance to get the energy consumption estimation. Now Click on **Save**. the device will be created.

If you use Sinric Pro free account, then you can add 3 devices for free.

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



The screenshot shows the Sinric Pro dashboard, specifically the 'Credentials' section. It features a 'New API Key' button and a table listing app keys and secrets. The table has columns for NAME, APP KEY, and APP SECRET. A 'Copy' button is next to each entry. The dashboard also includes a sidebar with navigation links like Dashboard, Devices, Rooms, Scenes, Schedules, Credentials, and Device Templates.

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

I have explained all the steps to program the NodeMCU using Arduino IDE.

1. Update the **Preferences** -> Additional 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 or [Click Here](#) to download the **ESP8266** board.
3. Download the required libraries from the following links:

- [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)

**Please download the latest version of the libraries from the given links.

Then install the libraries at Arduino IDE – **Sketch – Include Library – Add Zip Library**.

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"
```

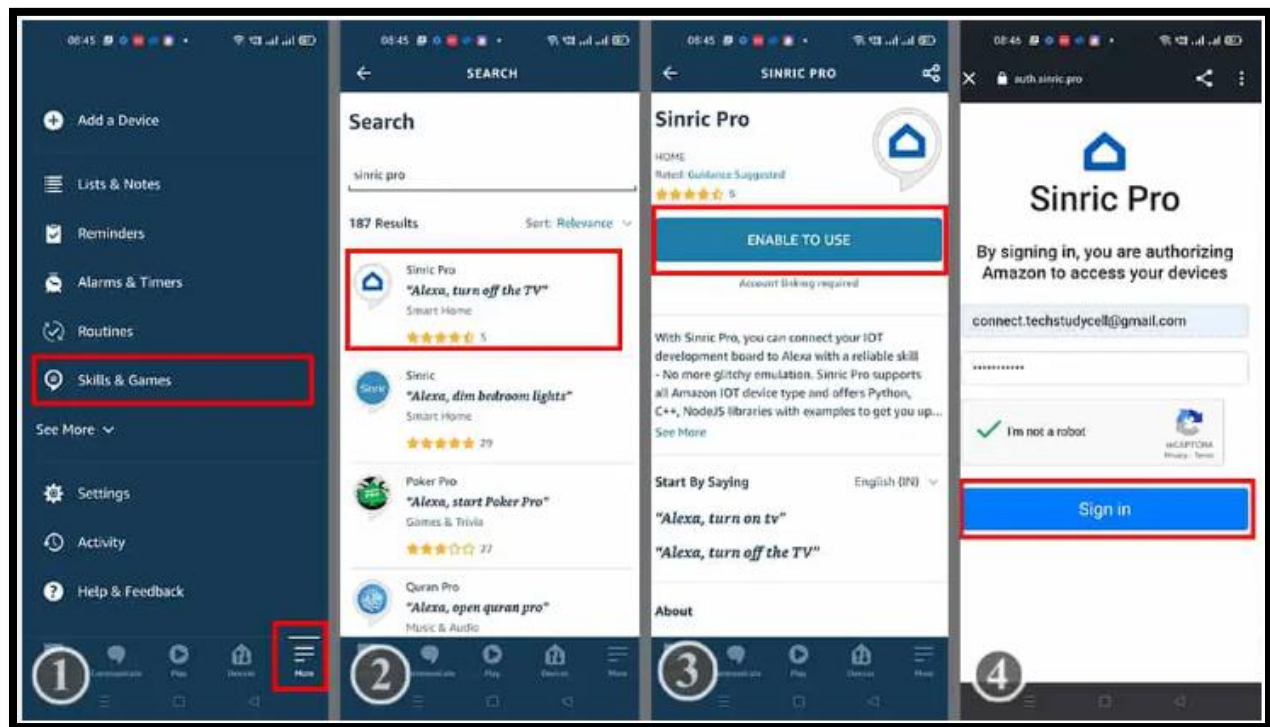
**When you add a device in Sinric Pro, a unique ID is assigned to that device. If you create 3 devices, then there will be 3 unique device IDs.

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

Uncomment the following line if you use **pushbuttons** instead of toggle switches.

Connect Sinric Pro with Alexa App

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



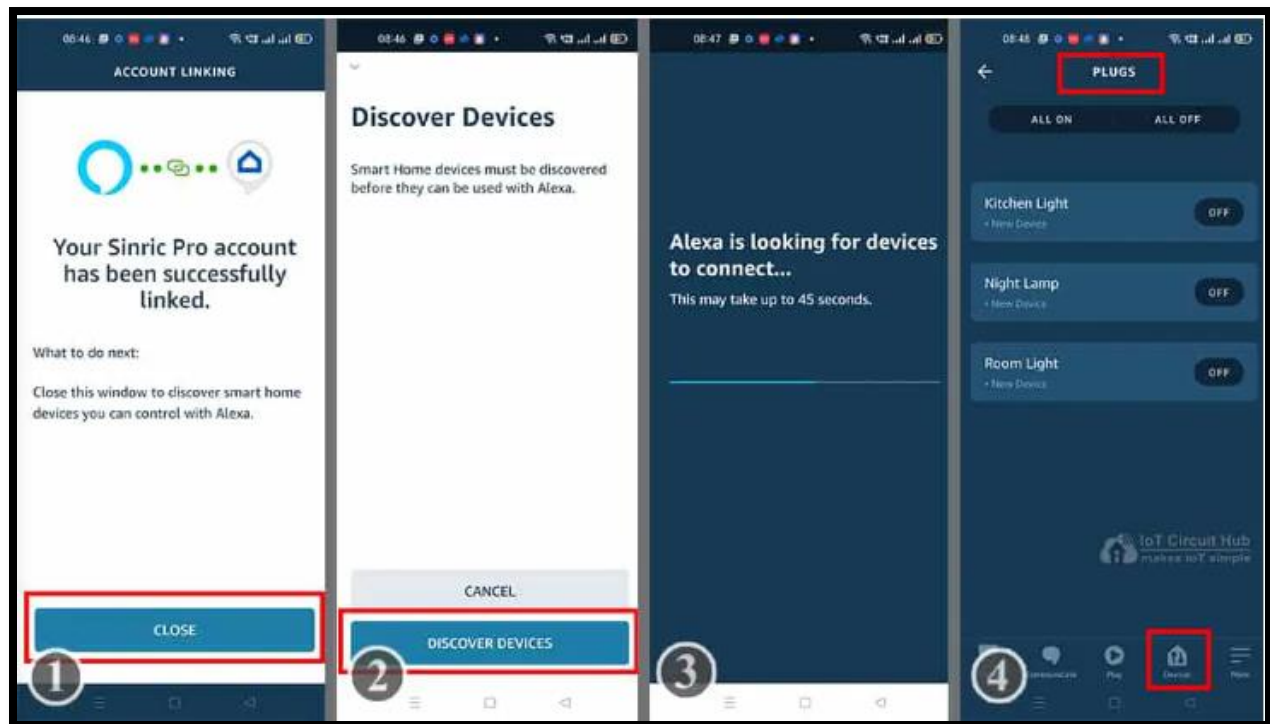
Before connecting the Alexa, you have to add devices in the Sinric Pro.

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



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.

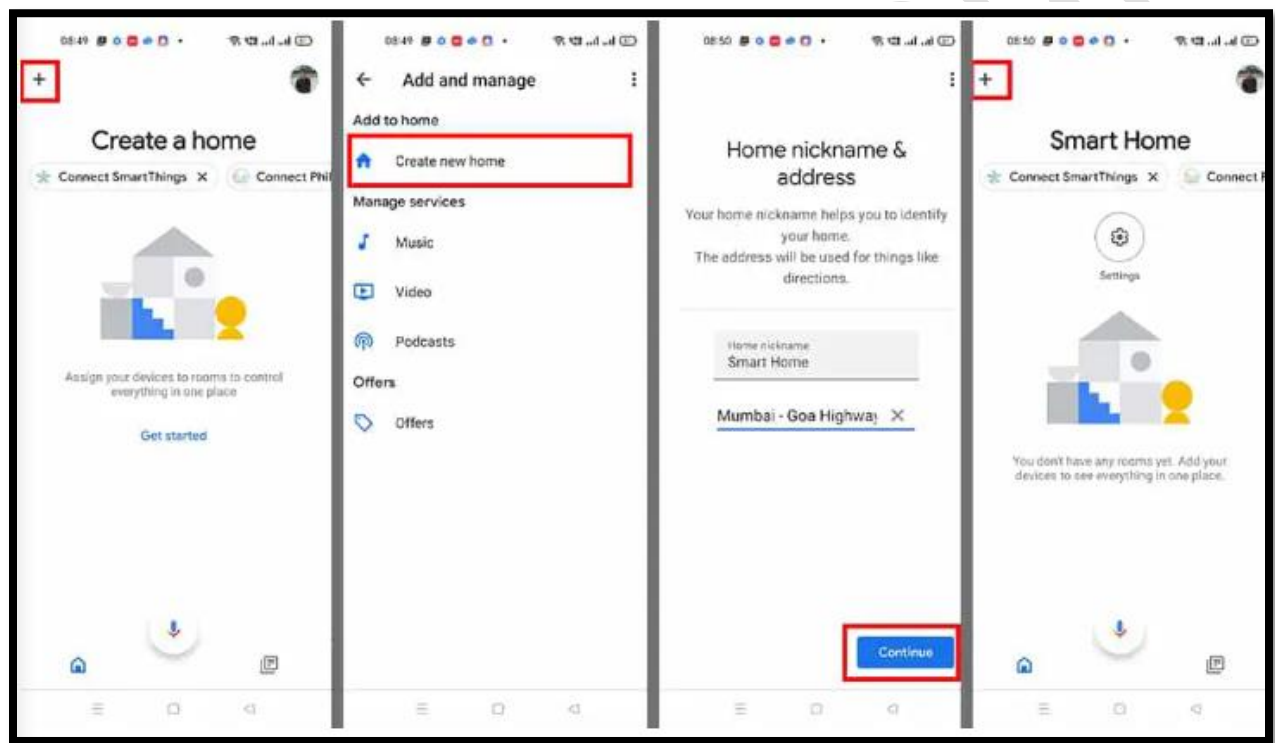
Now, if the ESP32 or ESP8266 is connected to the Wi-Fi, then you can control the appliances from **Alexa App**.

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.

Create a new Home in Google Home App

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



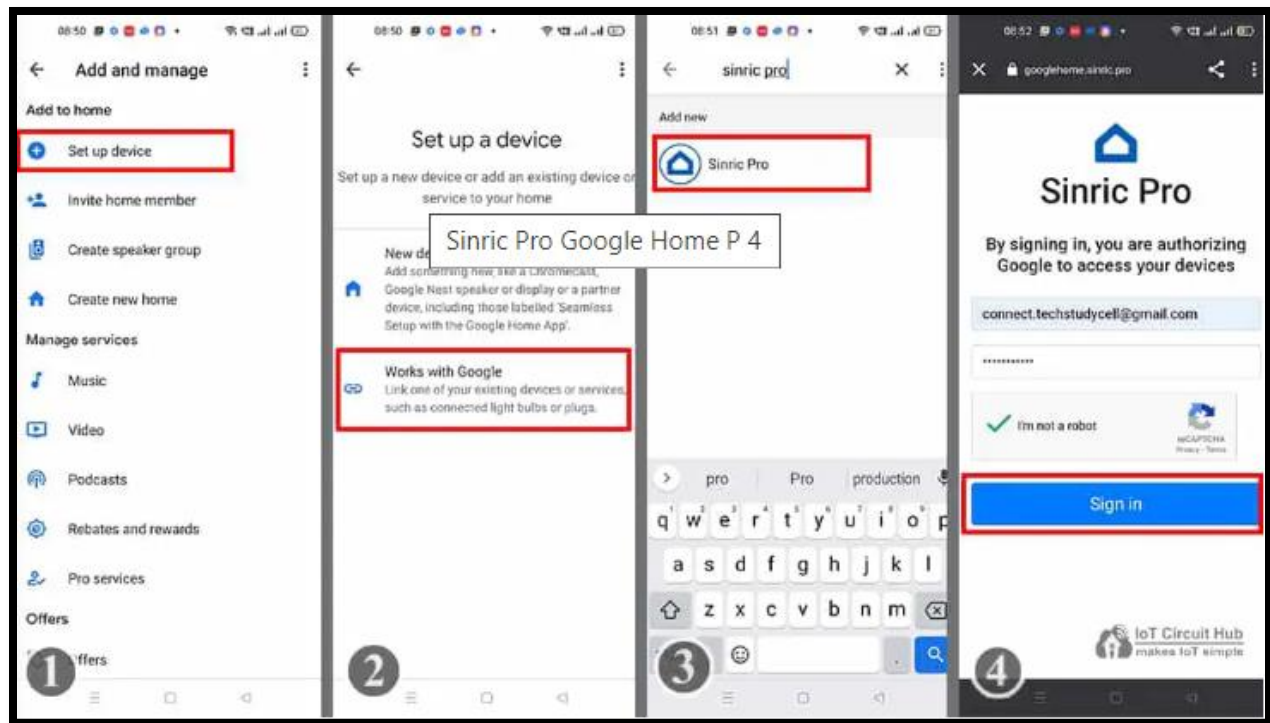
In the Google Home App, follow the following steps to create a Home:

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.

Before connecting the Google Home, you have to add devices in Sinric Pro.

Connect Sinric Pro with 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**.

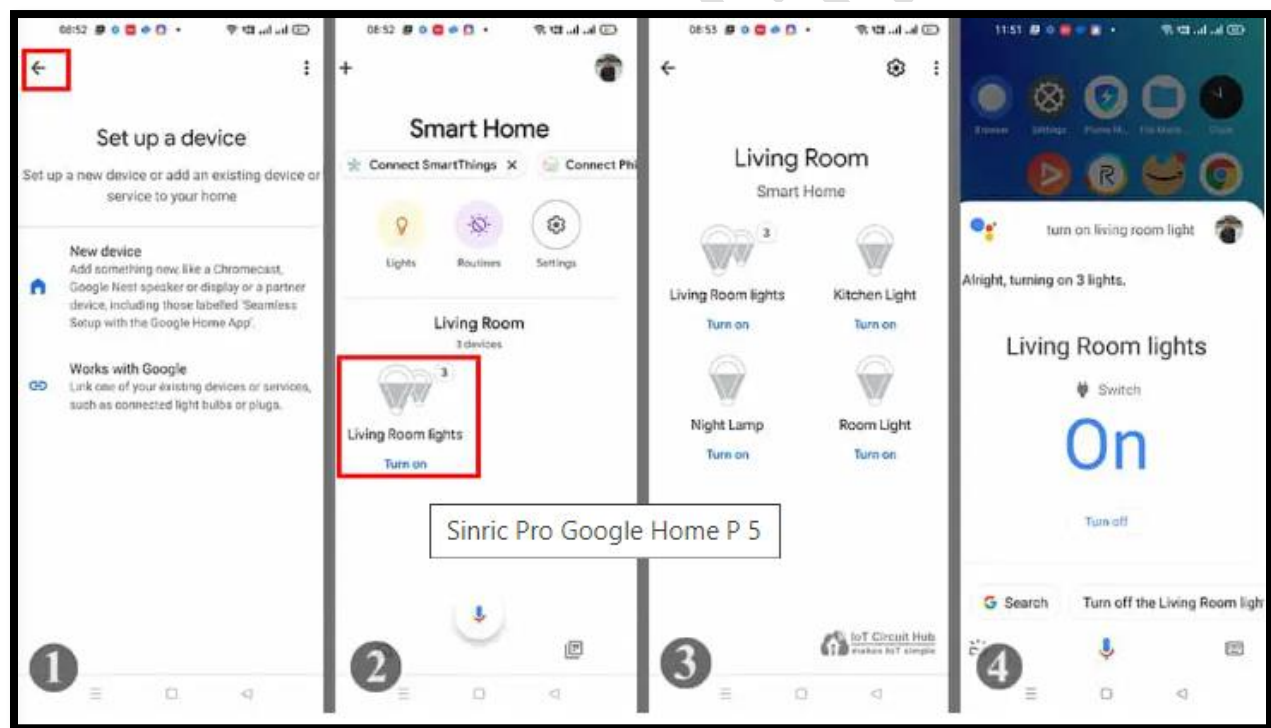
After that back to the home screen of Google Home App. Now you can see all the devices in your Google Home app.

Control devices with Google Home or Google Assistant

After connecting the Sinric Pro, go to the home page of the Google Home app. You can see the connected devices from Sinric Pro.

Now, if the ESP32 or ESP8266 is connected to the Wi-Fi, you can control the appliances from the **Google Home app**.

You can also ask **Google Assistant**, "Hey Google, Turn ON light" to control the appliances.



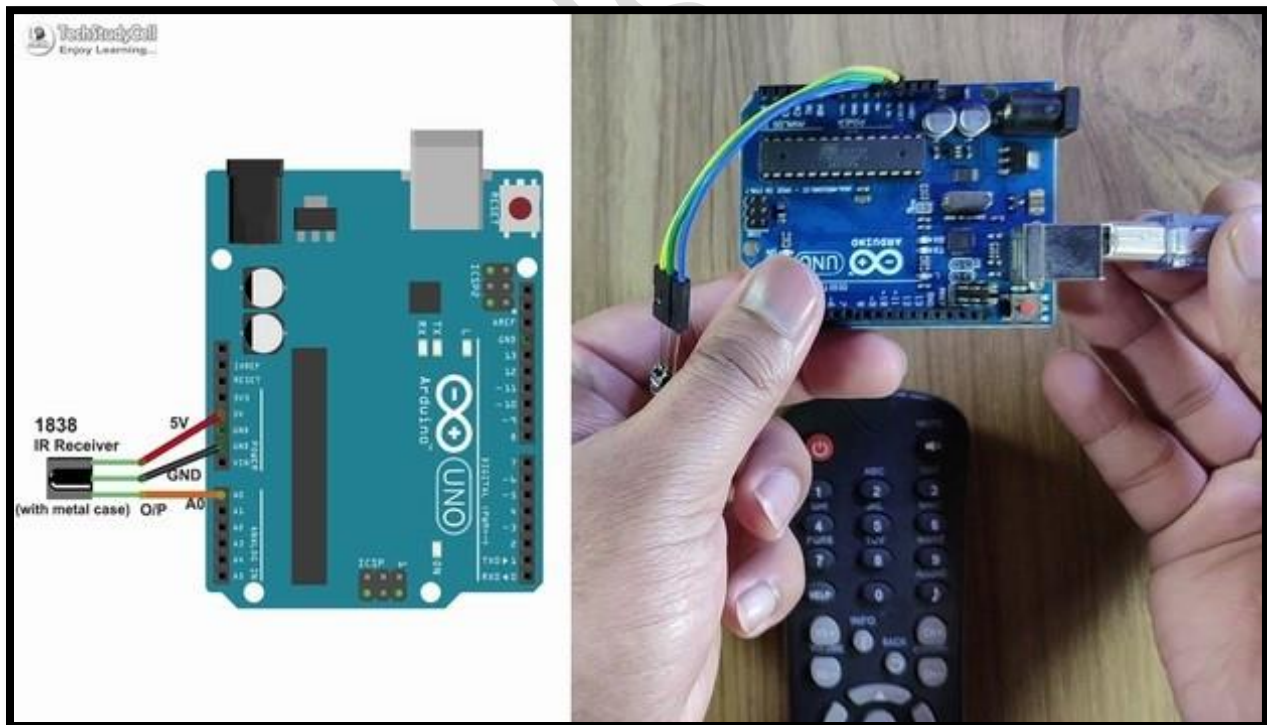
Get the IR Codes (HEX Code) From Remote

Now, to get the HEX codes from the remote, rst, we have to connect the IR receiver output pin with the **A0 pin** of Arduino UNO.

And give the 5V across the VCC and GND. The IR receiver **must have a metallic casing**, otherwise, you may face issues.

Then follow the following steps to get the HEX codes:

1. Install the **IRremote library(3.6.1)** in Arduino IDE.
2. **Download the attached code**, and upload it to Arduino UNO.
3. **Open Serial Monitor** with Baud rate **9600**.
4. Now, **press the IR remote button**.
5. The respective HEX code will populate in the serial monitor.
6. **Save all the HEX codes** in a text le. You have to update these HEX codes in the main sketch.





Program Arduino UNO With Arduino IDE

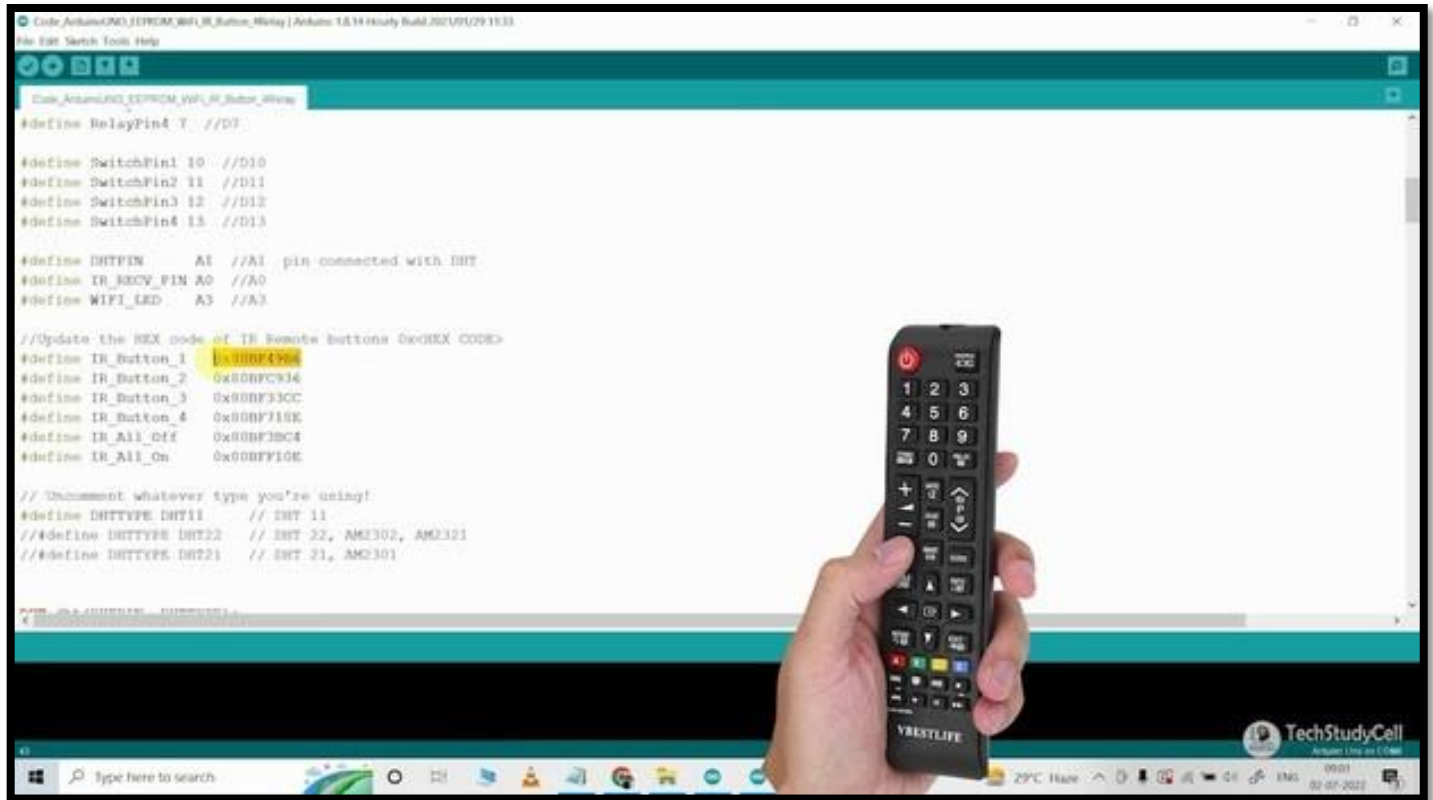
In the PCB I have used an ATmega328P microcontroller, so I have used USB to TTL board to program the microcontroller .

But if you don't use PCB then you can directly connect the Arduino UNO with the laptop.

For the main sketch, you need to install the following libraries

- IIRemote 3.6.1 Library
- AceButton 1.9.1 Library
- Arduino-timer 2.3.1 Library

Then you have to update the HEX code in the sketch for IR remote control.



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

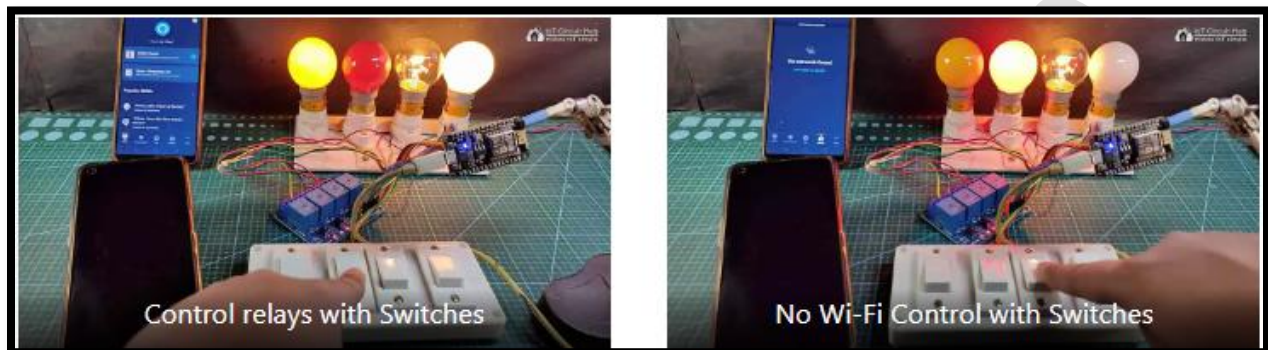
You can also ask Google Assistant, to turn on the light [**"Hey Google, Turn ON the Room Light"**]. Thus, you can control the appliances like light, fan, etc with voice commands using Google Assistant, and also monitor the current status of the switches from anywhere in the world from the Google Home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



NodeMCU Alexa Google Home Control P 21

Control relays manually with Switches

You can always control the appliances manually with switches or push buttons. and if the Arduino Board ON, then you can monitor the real-time status in Google Home and Alexa App.



Finally!! the Sinric Pro Smart Home System Is Ready

Now you can control your home appliances in a smart way.

I hope you have liked this Sinric Pro, Google Assistant and Alexa control home automation project. I have shared all the required information for this project.

I will really appreciate it if you share your valuable feedback. Also if you have any queries please write in the comment section.