Voice and Manual controlled Smart Switch Board with feedback

In this racing world, we have no time to do every work in our home. We all need a personal assistant to do our time taking works such as washing clothes in washing machine, making a cup of coffee, controlling appliances such as T.V., lights, fan, air conditioners, etc. which are far away from us. The solution for all these problems is this Smart Switch Board. We can control it with voice commands through Alexa , Alexa app in our Mobile phone and PCs and also in our traditional way controlling with manual switches. We can also get real time feedback in Alexa app while operating with manual switches. Moreover, it is portable and compact device.

# Objective

The main objective of this project is to control any Electrical appliances with both voice commands, Alexa app and with manual switches from anywhere in the world.

\*The project aim is to make our works done sitting at a place with a single voice command.

\*We can also check the state of appliances whether it is turned on or off in our mobile at anytime and anywhere in the world.

\*You can also control the appliances with manual switches when the internet connection is not alive.

# Components required

## Hardware

### 1.Nodemcu board:

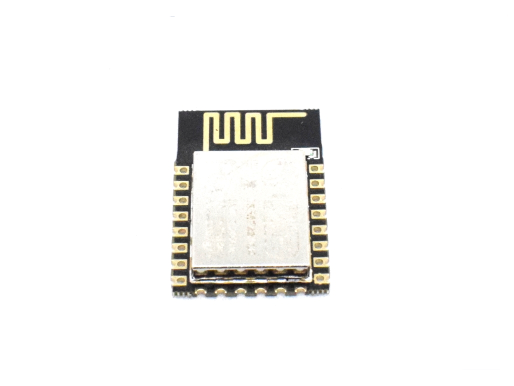
We will upload our code into this development board. It is more compatible than any other development boards available.

   
Buy it from here

<https://www.hnhcart.com/products/node-mcu-cp-210x?_pos=1&_sid=16b795a75&_ss=r>

If you want to make your project cheaper you can use ESP-12e board. You can check how to program it here

<https://youtu.be/UxIBCyP0V18>



Buy it from here

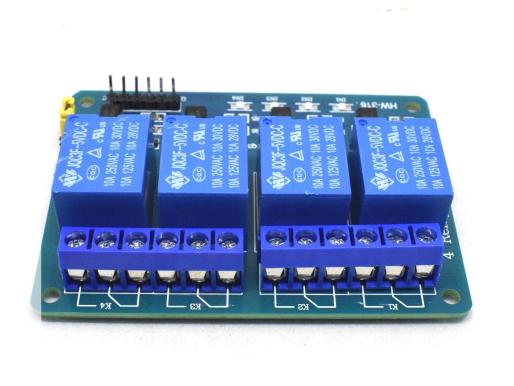
<https://www.hnhcart.com/products/esp-12e-wifi-module?_pos=1&_sid=ec9ba047b&_ss=r>

Don’t forgot to add 3.3v voltage regulator ( AMS 1117 3.3v regulator is recommended)

Buy it from here

<https://www.hnhcart.com/products/reg1117-ic?_pos=2&_sid=39426c2de&_ss=r>

### 2. 4 channel relay board:

It acts as a switch. It can be controlled with little dc voltage to control high voltage appliances. 



<https://www.hnhcart.com/products/4-channel-5v-relay-module?_pos=11&_sid=959d29598&_ss=r>

### 3. Jumper wires:

As its name, it is used to connect pins of nodemcu and relay board as well as switches.



### 4. Switches, Sockets and 8-way Gang box:

There is nothing to talk about it. These are familiar to everyone. These are available in every electrical shop.

### 5. Any Alexa device:

Alexa is a voice assistant made by Amazon.

Some Alexa devices are mentioned here. You can buy any of these for this project:

1. Alexa Echo Dot:

<https://www.amazon.in/Echo-Dot-3rd-Gen/dp/B07PFFMP9P/ref=sr_1_1?dchild=1&keywords=alexa+devices&qid=1610024316&sr=8-1>

<https://www.amazon.in/Echo-Dot-4th-Gen-Blue/dp/B084KSRFXJ/ref=sr_1_2?dchild=1&keywords=alexa+devices&qid=1610024411&sr=8-2>

1. Alexa Echo Portable:

<https://www.amazon.in/Echo-Input-Portable-Smart-Speaker/dp/B07YP9WYFN/ref=sr_1_10?dchild=1&keywords=alexa+devices&qid=1610024411&sr=8-10>

## Software

**Arduino Software (IDE):** It is an open-source *Software* (*IDE*) which makes it easy for us to write code and upload it to the Arduino boards. It runs on Windows, Linux etc.

### **Sinric Account** : Sinric acts as bridge between our switch board and Alexa server. You have to go to Sinric.com and create your own account. Do not share its API keys and device ids with anyone.

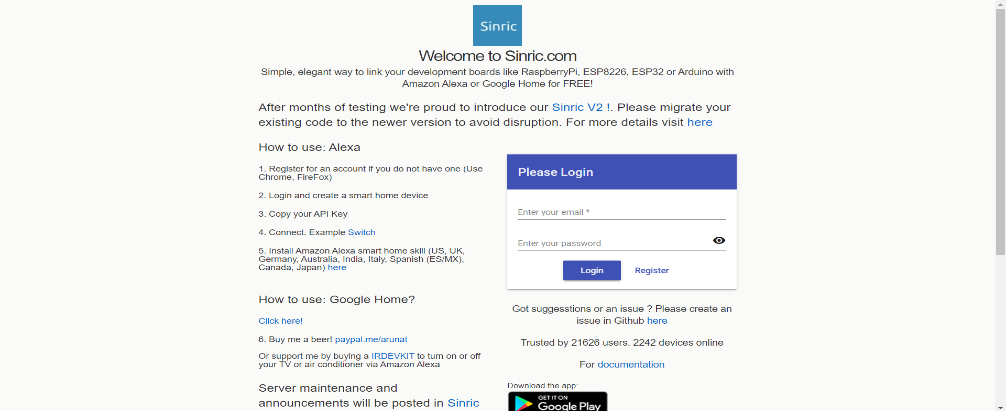
# Code

To upload the program, follow the below steps:

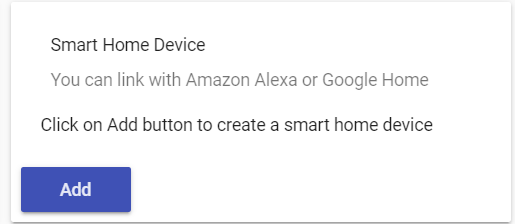
1. You should add Esp8266 to your Arduino IDE. To do this please watch this video

<https://youtu.be/OC9wYhv6juM>

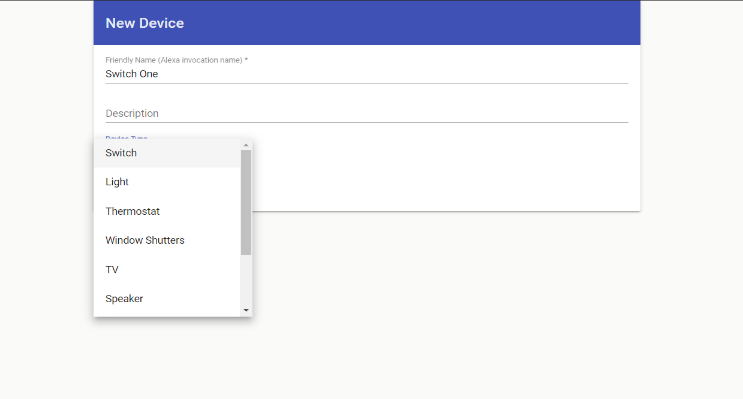
2.Go to Sinric.com and login with your account.



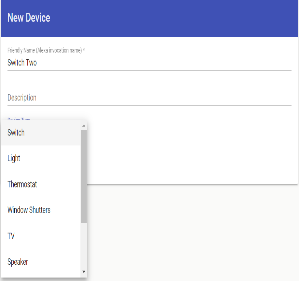
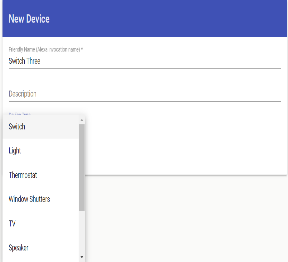
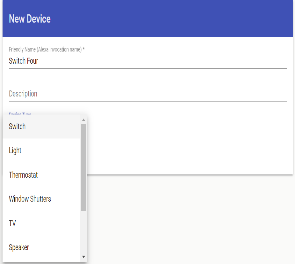
3.Now click on “Add” button to create a smart home device.



4.Write “Switch One” at friendly name and at device type select “Switch” and save.



5. Repeat this process up to three more times. But at Friendly Name you should type now type Switch Two, Switch Three, Switch Four respectively in each Add.

6.After all these steps copy the code given below and go to Arduino IDE and paste it. Add the necessary Libraries in the Software which are provided in the code.

\*code starts from here\*

//This code is written by MADUGULA SAI VENKATA SARAN

int device1 = 0;

int device2 = 2;

int device3 = 12;

int device4 = 13;

// Switch

const int BUTTON1\_PIN = 5;

const int BUTTON2\_PIN = 4;

const int BUTTON3\_PIN = 14;

const int BUTTON4\_PIN = 10;

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <ESP8266WiFiMulti.h>

#include <WebSocketsClient.h> // get it from https://github.com/Links2004/arduinoWebSockets/releases

#include <ArduinoJson.h> // get it from https://arduinojson.org/ or install via Arduino library manager

#include <StreamString.h>

#include <AceButton.h> // https://github.com/bxparks/AceButton

using namespace ace\_button;

ESP8266WiFiMulti WiFiMulti;

WebSocketsClient webSocket;

WiFiClient client;

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

#define MySSID "Your WiFi Network SSID" // Change to your Wifi network SSID

#define MyWifiPassword "Your WiFi Password" // Change to your Wifi network password

#define HEARTBEAT\_INTERVAL 300000 // 5 Minutes

uint64\_t heartbeatTimestamp = 0;

bool isConnected = false;

ButtonConfig config1;

AceButton button1(&config1);

ButtonConfig config2;

AceButton button2(&config2);

ButtonConfig config3;

AceButton button3(&config3);

ButtonConfig config4;

AceButton button4(&config4);

void handleEvent1(AceButton\*, uint8\_t, uint8\_t);

void handleEvent2(AceButton\*, uint8\_t, uint8\_t);

void handleEvent3(AceButton\*, uint8\_t, uint8\_t);

void handleEvent4(AceButton\*, uint8\_t, uint8\_t);

String device\_ID\_1 = "Paste your Switch One deviceID Here ";

String device\_ID\_2 = "Paste your Switch Two deviceID Here";

String device\_ID\_3 = "Paste your Switch Three deviceID Here";

String device\_ID\_4 = "Paste your Switch Four deviceID Here";

//This code is written by MADUGULA SAI VENKATA SARAN

// If you are going to use a push button to on/off the switch manually, use this function to update the status on the server

// so it will reflect on Alexa app.

// eg: setPowerStateOnServer("deviceid", "ON")

void setPowerStateOnServer(String deviceId, String value);

void setTargetTemperatureOnServer(String deviceId, String value, String scale);

void button1Handler(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

Serial.println("EVENT1");

switch (eventType) {

case AceButton::kEventPressed:

Serial.println("kEventPressed");

setPowerStateOnServer(device\_ID\_1, "OFF");

digitalWrite(device1, LOW);

break;

case AceButton::kEventReleased:

Serial.println("kEventReleased");

setPowerStateOnServer(device\_ID\_1, "ON");

digitalWrite(device1, HIGH);

break;

}

}

void button2Handler(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

Serial.println("EVENT2");

switch (eventType) {

case AceButton::kEventPressed:

Serial.println("kEventPressed");

setPowerStateOnServer(device\_ID\_2, "OFF");

digitalWrite(device2, LOW);

break;

case AceButton::kEventReleased:

Serial.println("kEventReleased");

setPowerStateOnServer(device\_ID\_2, "ON");

digitalWrite(device2, HIGH);

break;

}

}

void button3Handler(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

Serial.println("EVENT3");

switch (eventType) {

case AceButton::kEventPressed:

Serial.println("kEventPressed");

setPowerStateOnServer(device\_ID\_3, "OFF");

digitalWrite(device3, LOW);

break;

case AceButton::kEventReleased:

Serial.println("kEventReleased");

setPowerStateOnServer(device\_ID\_3, "ON");

digitalWrite(device3, HIGH);

break;

}

}

void button4Handler(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

Serial.println("EVENT4");

switch (eventType) {

case AceButton::kEventPressed:

Serial.println("kEventPressed");

setPowerStateOnServer(device\_ID\_4, "OFF");

digitalWrite(device4, LOW);

break;

case AceButton::kEventReleased:

Serial.println("kEventReleased");

setPowerStateOnServer(device\_ID\_4, "ON");

digitalWrite(device4, HIGH);

break;

}

}

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

//This code is written by MADUGULA SAI VENKATA SARAN

void turnOn(String deviceId) {

if (deviceId == "Paste your Switch One deviceID Here") // Device ID of device 1

{

Serial.print("Turn on device id: ");

Serial.println(deviceId);

digitalWrite(device1, HIGH);

}

else if (deviceId == "Paste your Switch Two deviceID Here") // Device ID of device 2

{

Serial.print("Turn on device id: ");

Serial.println(deviceId);

digitalWrite(device2, HIGH);

}

else if (deviceId == "Paste your Switch Three deviceID Here") // Device ID of device 3

{

Serial.print("Turn on device id: ");

Serial.println(deviceId);

digitalWrite(device3, HIGH);

}

else if (deviceId == "Paste your Switch Four deviceID Here") // Device ID of device 4

{

Serial.print("Turn on device id: ");

Serial.println(deviceId);

digitalWrite(device4, HIGH);

}

}

void turnOff(String deviceId) {

if (deviceId == "Paste your Switch One deviceID Here") // Device ID of device 1

{

Serial.print("Turn off Device ID: ");

Serial.println(deviceId);

digitalWrite(device1, LOW);

}

else if (deviceId == "Paste your Switch Two deviceID Here") // Device ID of device 2

{

Serial.print("Turn off Device ID: ");

Serial.println(deviceId);

digitalWrite(device2, LOW);

}

else if (deviceId == "Paste your Switch Three deviceID Here") // Device ID of device 3

{

Serial.print("Turn off Device ID: ");

Serial.println(deviceId);

digitalWrite(device3, LOW);

}

else if (deviceId == "Paste your Switch Four deviceID Here") // Device ID of device 4

{

Serial.print("Turn off Device ID: ");

Serial.println(deviceId);

digitalWrite(device4, LOW);

}

}

void webSocketEvent(WStype\_t type, uint8\_t \* payload, size\_t length) {

switch(type) {

case WStype\_DISCONNECTED:

isConnected = false;

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);

// Example payloads

// For Switch or Light device types

// {"deviceId": xxxx, "action": "setPowerState", value: "ON"} // https://developer.amazon.com/docs/device-apis/alexa-powercontroller.html

//This code is written by MADUGULA SAI VENKATA SARAN

// For Light device type

// Look at the light example in github

DynamicJsonBuffer jsonBuffer;

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

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 == "SetTargetTemperature") {

String deviceId = json ["deviceId"];

String action = json ["action"];

String value = json ["value"];

}

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 setup()

{

pinMode (device1, OUTPUT);

pinMode (device2, OUTPUT);

pinMode (device3, OUTPUT);

pinMode (device4, OUTPUT);

pinMode (LED\_BUILTIN, OUTPUT);

// Setup button

// Button uses the built-in pull up register.

pinMode(BUTTON1\_PIN, INPUT\_PULLUP); // INPUT\_PULLUP so no need a 10K resistor

pinMode(BUTTON2\_PIN, INPUT\_PULLUP); // INPUT\_PULLUP so no need a 10K resistor

pinMode(BUTTON3\_PIN, INPUT\_PULLUP); // INPUT\_PULLUP so no need a 10K resistor

pinMode(BUTTON4\_PIN, INPUT\_PULLUP); // INPUT\_PULLUP so no need a 10K resistor

Serial.begin(115200);

WiFiMulti.addAP(MySSID, MyWifiPassword);

Serial.println();

Serial.print("Connecting to Wifi: ");

Serial.println(MySSID);

// Waiting for Wifi connect

while(WiFiMulti.run() != WL\_CONNECTED) {

Serial.print(".");

digitalWrite(LED\_BUILTIN ,LOW);

delay(500);

}

if(WiFiMulti.run() == WL\_CONNECTED) {

Serial.println("");

Serial.print("WiFi connected. ");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

}

// server address, port and URL

webSocket.begin("iot.sinric.com", 80, "/");

// event handler

webSocket.onEvent(webSocketEvent);

webSocket.setAuthorization("apikey", MyApiKey);

// try again every 5000ms if connection has failed

//This code is written by MADUGULA SAI VENKATA SARAN

webSocket.setReconnectInterval(5000); // If you see 'class WebSocketsClient' has no member named 'setReconnectInterval' error update arduinoWebSockets

config1.setEventHandler(button1Handler);

config2.setEventHandler(button2Handler);

config3.setEventHandler(button3Handler);

config4.setEventHandler(button4Handler);

button1.init(BUTTON1\_PIN);

button2.init(BUTTON2\_PIN);

button3.init(BUTTON3\_PIN);

button4.init(BUTTON4\_PIN);

}

void loop() {

button1.check();

button2.check();

button3.check();

button4.check();

webSocket.loop();

//This code is written by MADUGULA SAI VENKATA SARAN

if(isConnected) {

uint64\_t now = millis();

// Send heartbeat in order to avoid disconnections during ISP resetting IPs over night.

if((now - heartbeatTimestamp) > HEARTBEAT\_INTERVAL) {

heartbeatTimestamp = now;

webSocket.sendTXT("H");

}

}

}

void setPowerStateOnServer(String deviceId, String value) {

DynamicJsonBuffer jsonBuffer;

JsonObject& root = jsonBuffer.createObject();

root["deviceId"] = deviceId;

root["action"] = "setPowerState";

root["value"] = value;

StreamString databuf;

root.printTo(databuf);

webSocket.sendTXT(databuf);

}

//eg: setPowerStateOnServer("deviceid", "CELSIUS", "25.0")

void setTargetTemperatureOnServer(String deviceId, String value, String scale) {

DynamicJsonBuffer jsonBuffer;

JsonObject& root = jsonBuffer.createObject();

root["action"] = "SetTargetTemperature";

root["deviceId"] = deviceId;

JsonObject& valueObj = root.createNestedObject("value");

JsonObject& targetSetpoint = valueObj.createNestedObject("targetSetpoint");

targetSetpoint["value"] = value;

targetSetpoint["scale"] = scale;

StreamString databuf;

root.printTo(databuf);

webSocket.sendTXT(databuf);

}

//This code is written by MADUGULA SAI VENKATA SARAN

\*code ends here\*

7.Now again go to your Sinric account and copy the API Key, Device Ids and paste them at the places where it is asked in the code respectively at their appropriate positions.

8. Select the appropriate port and board.

9. Check everything once again and upload the code.

10. Now install the Alexa app in your Mobile phone or Laptop whatever it is. Login with your Amazon Account (it must be same with your Alexa device account). Accept all the asked terms and conditions. Now go to “More”. Tap on “Skills & Games”. Search for “Sinric” and tap “Enable”. Add your Sinric Account there if it is asked.

11. You can get a notification in your Alexa App that “4 devices are available”. Tap on it and add devices.

# Circuit Diagram

D3 of NodeMCU - IN 1 of Relay board

D4 of NodeMCU - IN 2 of Relay board

D6 of NodeMCU - IN 3 of Relay board

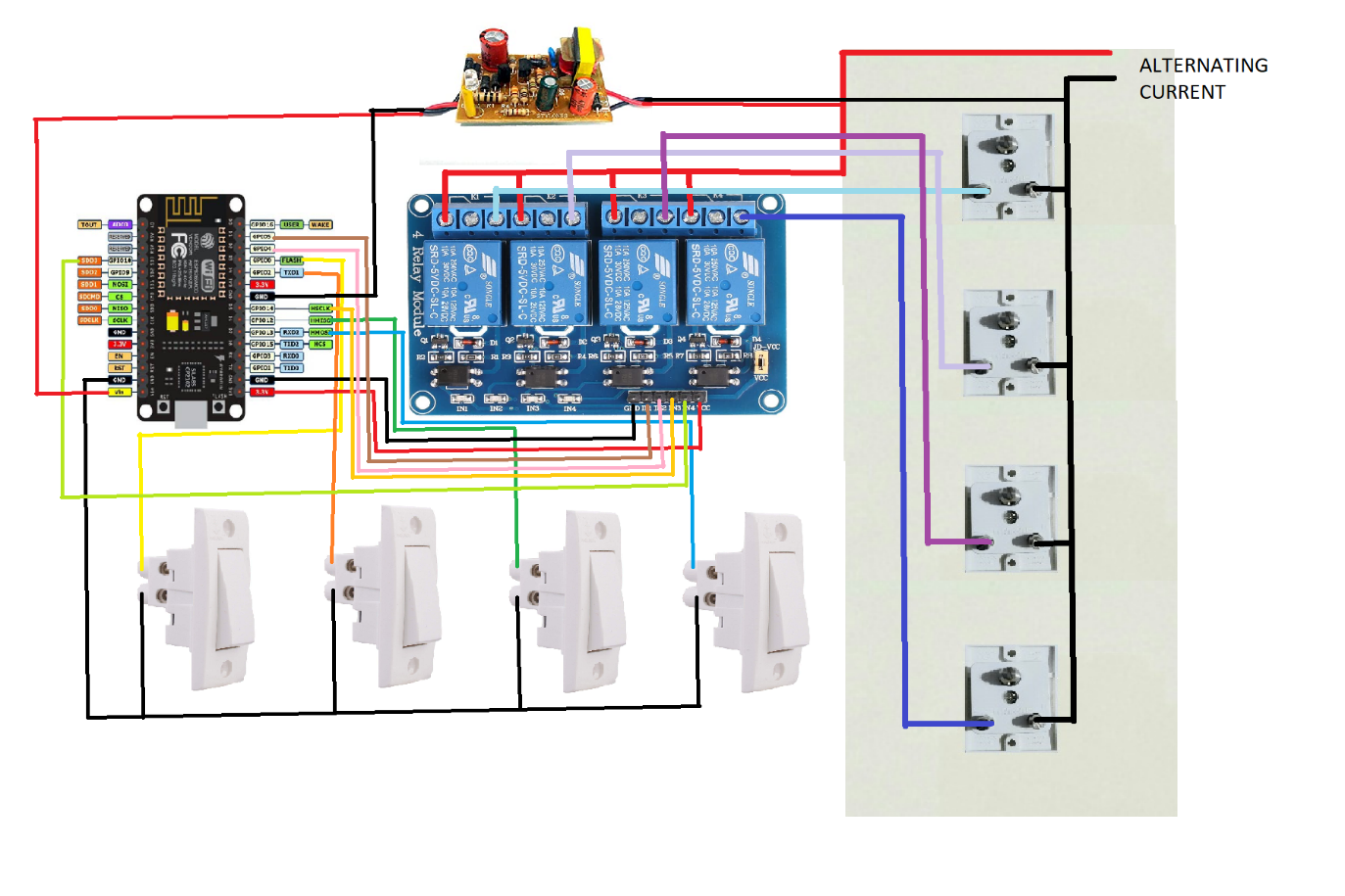
D7 of NodeMCU - IN 4 of Relay board

D1 of NodeMCU - Switch 1

D2 of NodeMCU - Switch 2

D5 of NodeMCU - Switch 3

SD3 of NodeMCU - Switch 4



# Conclusion

Now all set. Your Smart Switch Board is ready. It is portable, can be used at any place, easy to carry anywhere. It is dead cheaper than any other Smart Switch Boards available in the market.

You can control it in three ways

1. With your voice commands, just like “turn on Switch One” to turn on first switch.
2. From your Alexa App in Devices section from any where in the world, if internet connection is present.
3. Whether the internet connection is alive or not, it doesn’t matter. You can control the Appliances with switches on the Smart Switch Board at any time.

## By

## Madugula Sai Venkata Saran