\

A Major Project Synopsis on

**Home Automation Using IoT**

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

BACHELOR OF TECHNOLOGY

in

Electronics and Communication Engineering

2020-2021

By

Samrat Bhatnagar

Electronics and Communication Engineering

189202143

Under the Guidance of

Mr.Vidyadhar Jinnappa Aski

**ACKNOWLEDGEMENTS**

We would like to express our gratitude towards Mr. Vidhyadhar Jinnappa Aski (Assistant Professor, Computers & Communications Engineering Department) who is our project supervisor for this project titled “**Home Automation using IoT**”. We would also like to thank HOD of CCE Department **Dr. Vijaypal Singh Dhaka** for giving us this opportunity to work on this project.

**ABSTRACT**

Our project titled “Home Automation using IoT” holds relevance in today’s world which is largely driven by technology in many ways. Home Automation has always had one purpose and that is to control your home using either your voice or with a touch of a button, and since most of today’s technology is advanced and supports this idea, it only makes sense to utilise this idea of automation to the fullest if not in some way or the other.

To make this project a success, we used 4 things: ESP8266 Microcontroller, Amazon’s Alexa Voice Service, Kakopappa’s Sinric Library, 4 Channel Relay. Since this project is based on Internet of Things, We used ESP8266 as a thing, configured it using Arduino custom libraries, integrated it with Sinric, and used Alexa to voice/remotely control the devices we configured.

After running initial tests we found that we were right in our approach to not complicate things by using Amazon Web Services. To integrate this project with AWS we would have needed to create a AWS Lambda function, create our own Alexa Skill, and most importantly our own app to control our devices among other complicated steps. This method was not very feasible, since we were trying to create a small scale project in our homes during lockdown with added time constraints.

The important tools/software we used to build this project were an ESP8266 Wi-Fi Module, Amazon Alexa Mobile App, 4 Channel Relay, Breadboard, LEDs for demonstration purposes, and some Dupont wires for connecting this whole setup together.

**INTRODUCTION**

The IoT based Home Automation will enable the user to use a Home Automation System based on Internet of Things (IoT). Using IoT, you can enable any device to connect to the Internet and perform a desired action quickly, reliably, and easily. These devices can work alone or together with other devices or hubs for an integrated smart home experience. We are controlling lights, fans and switches in our project using mainly three things, which are AMAZON ALEXA, Sinric Pro and ESP8266 NODEMCU 12-E. The voice input is given by the user to Alexa. The input is then transmitted from Alexa. Here the input is saved as a signal and then passed over to the SINRIC cloud (Switched-mode power supply). It is then sent to Node MCU. Node MCU acts as a major role in our system since it helps to receive WiFi and support the entire system. The signal is then forwarded from Node MCU to Channel Relay, then to the socket extension box. From here the signal is transmitted to home appliances where the fan, light, and other electronic devices are connected.

**MOTIVATION**

* Upgrade from Mechanical switches that require timely replacement/Maintenance.
* Reduces chances of getting shocked through physical contact.
* Highly convenient (Turning on your thermostat at scheduled times, controlling lights and other appliances with just your voice, Aged people, people suffering from blindness are also relived and it makes their life easier)
* Avoids wastage of electricity.

**PROBLEM STATEMENT**

“Automate the electrical appliances in your home using Alexa.”

* Amazon Alexa, is a [virtual assistant](https://en.wikipedia.org/wiki/Virtual_assistant) [AI](https://en.wikipedia.org/wiki/AI) technology developed by [Amazon Lab126](https://en.wikipedia.org/wiki/Amazon_Lab126). It is capable of voice interaction, music playback, making to-do lists, [setting alarms](https://en.wikipedia.org/wiki/Alarm_clock), etc. Alexa can also control several [smart devices](https://en.wikipedia.org/wiki/Smart_device) using itself as a [home automation](https://en.wikipedia.org/wiki/Home_automation) system.
* Users are able to extend the Alexa capabilities by installing "skills" (additional functionality developed by third-party vendors, in other settings more commonly called [apps](https://en.wikipedia.org/wiki/Mobile_app) such as weather programs and audio features).
* There are generic Internet of Things (IoT) modules designed to connect devices to the internet and control them through cloud software.
* We are expected to develop economically suitable wireless IoT systems to solve this problem by integrating the amazon skills (can use third party software) to a Generic IoT Wi-Fi Module to control the various appliances such as lights, fans, switches, etc through Alexa voice assistance.

**FACILITIES USED**

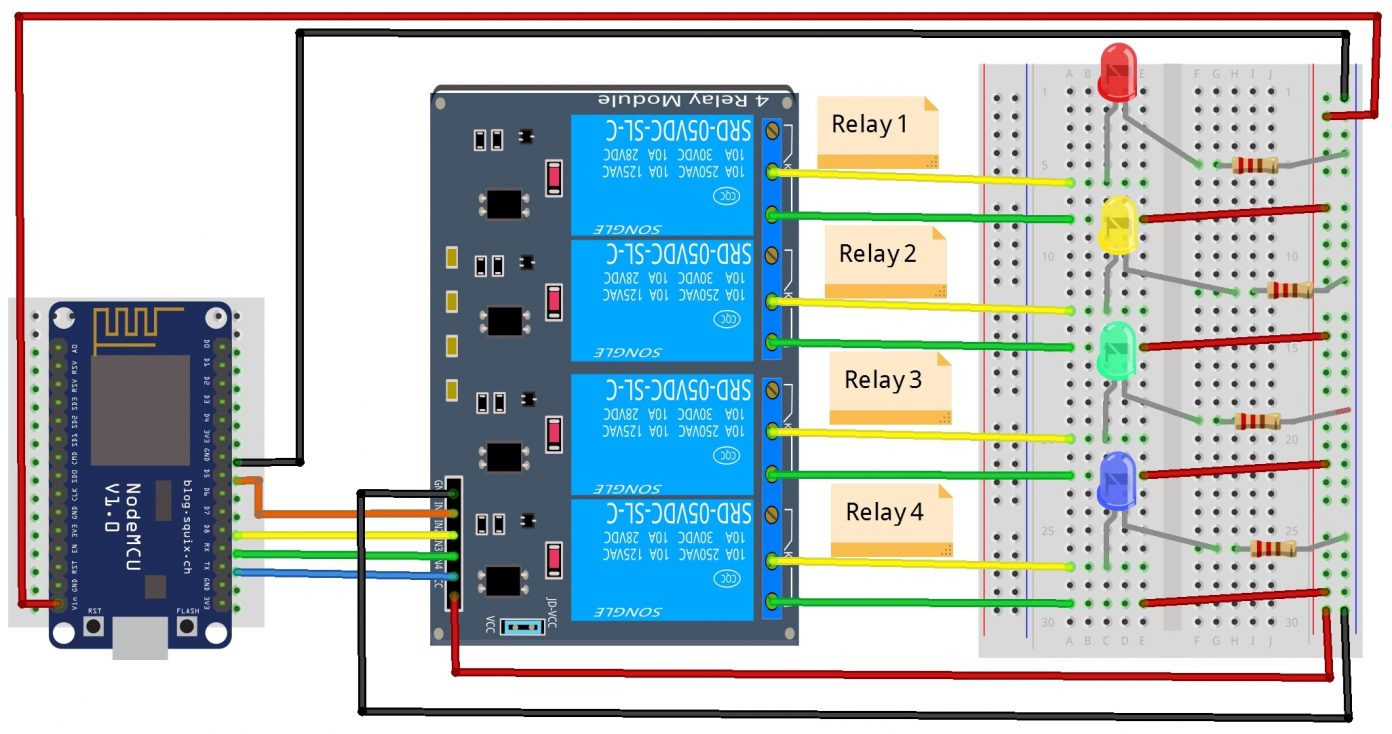
*SOFTWARE*:

* Sinric- With Sinric, you can connect your WeMos Mini D1, ESP8266, ESP32, Arduino development boards with Alexa without emulating as a Belkin switch or using native Amazon services like Lambda functions,or creating a Alexa Skill and most importantly the hassle of creating an app to control your devices (necessary front-end for our project.)
* Amazon Alexa- Alexa is a virtual assistant created by Amazon and released in November 2014. Ever since then it has made a niche for itself in Home Automation domain. Due to its easy to understand and friendly UI, cost-effective and availability, and global reach it is the perfect voice companion for our Home Automation project.
* Arduino IDE- The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.
* ESP8266 Library for Arduino IDE

*HARDWARE:*

* ESP8266 NodeMCU– It is a very versatile board from ESPRESSIF Systems, it has a WiFi module built in already which will prove to be very crucial during our project since all interfacing is done with the relay and Alexa service through this module only. It is already widely used in several home-automation projects.
* 4 Channel Relay- The 4 Channel Relay Module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontroller such as Arduino.
* Dupont Cables & LEDs- Dupont Cables are also called Jumper Wire cables. They are low cost and used to connect hardware such as sensors, Arduino boards and breadboards together.

**METHODOLOGY**



We followed the below mentioned steps for development stage of our project:

1. For programming the ESP8266 module using Arduino IDE we were required to download some additional modules, since Arduino IDE does not have native ESP Modules in its board manager:

* We Installed the current upstream Arduino IDE (1.8.9 level or above). The current version is on the [Arduino website](https://www.arduino.cc/en/main/software).
* Started the Arduino IDE and opened the Preferences window.
* Entered https://arduino.esp8266.com/stable/package\_esp8266com\_index.json into the Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.
* Opened Boards Manager from Tools > Board Menu and installed the esp8266 platform (and do not forget to select your ESP8266 board from Tools > Board menu after installation).

We then Downloaded the example Sinric code from GitHub by kakopappa (<https://github.com/kakopappa/sinric>)

1. Went to www.sinric.com
2. Created an account and logged in
3. Configured new devices. e.g.: light, fan, as switches
4. This generated unique API keys from the Sinric website & we copied these into the Source Code file. (It will be something like this db8d0309-84ce-485b-8957-xxxxxxxxxxxx)
5. Replaced our WIFI-SSID, our WIFI-Password and made other needful changes to the source code like if you wish to use 2-channel relay, or an 8-channel relay.
6. Installed/Enabled Sinric Smart Home Skill in Amazon’s Alexa App on your Android/iOS device
7. Asked Alexa to discover devices.
8. And finally, we could control our devices either manually or using Alexa Voice Service.

**IMPLEMENTATION**

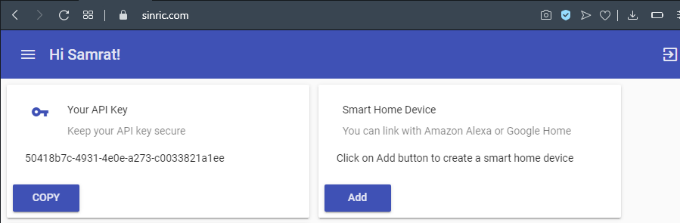
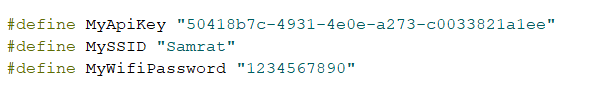
* We mainly used WebSockets Protocol for communication between Server and client which was readily available at GitHub by Kakopappa. Here Sinric providing the server & our ESP being the client.
* WebSockets is the best communications protocol used in IoT devices for real-time communications.
* It provides a 2-way low latency communication which is possible because websockets keep the connection open until deliberately closed by one of the two parties (i.e. the client or the server).
* websockets is a push-based strategy. This means, unlike polling or long-polling (like in HTTP) which involves asking the server recursively, the server in pub/sub itself sends a message to the interested client whenever any new data becomes available.

\*Why we didn’t use HTTP or MQTTP for Communications?

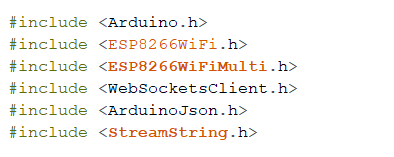
HTTP: is not considered to be a suitable protocol for real-time applications since it provides non-persistent connections, i.e. a new request-response cycle is established each time the client wishes to communicate with the server.

As for MQTTP: it is also a lightweight application-level protocol that allows you to set up a persistent connection between a server and a client, but integrating MQTT protocol with Alexa was getting difficult and somehow our ESP wasn’t able to establish a stable connection with AWS either. Which was also the main reason we scrubbed the AWS method and went on with Sinric.

# # #: The Sinric open-source web app provided us with the API key for authentication.



# # #: These are the libraries/modules we had to install externally for programming our ESP using Arduino.



# 1 #: This is the code we used to run the WebSocket Protocol for establishing connection between our ESP and the server.

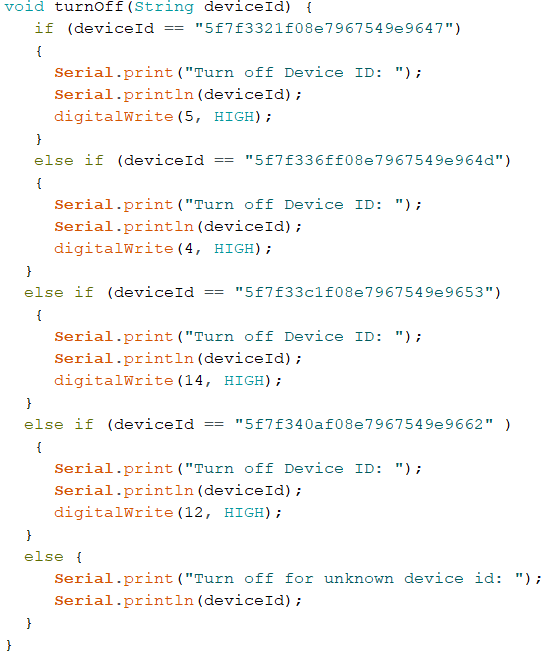


NOTE: ArduinoJSON is just another library to support coding in embedded C++ syntax since we are not using Arduino programming.

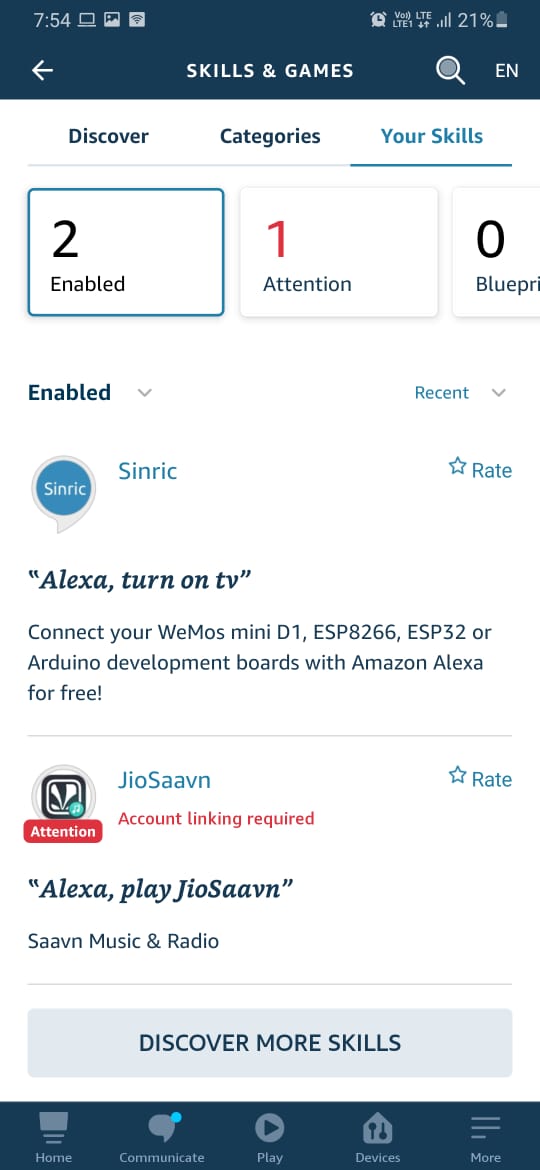
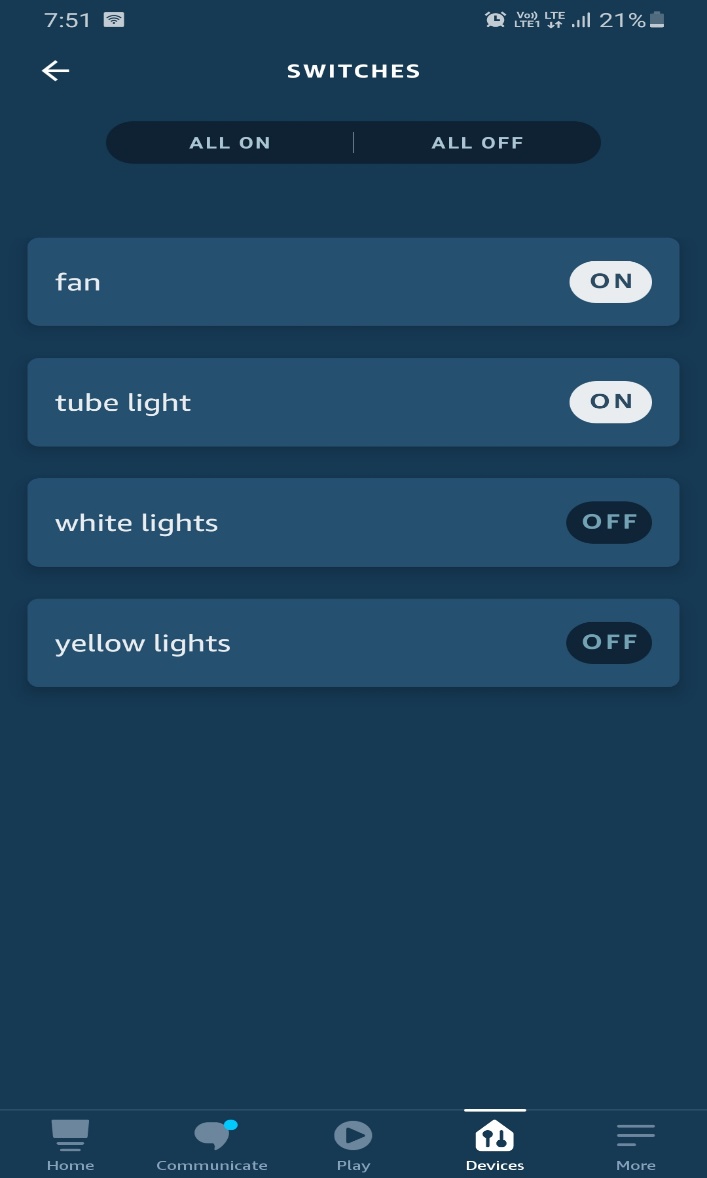
# 2 #: This is the code for setting up our ESP8266 for wi-fi connection and the pinouts for the relay:



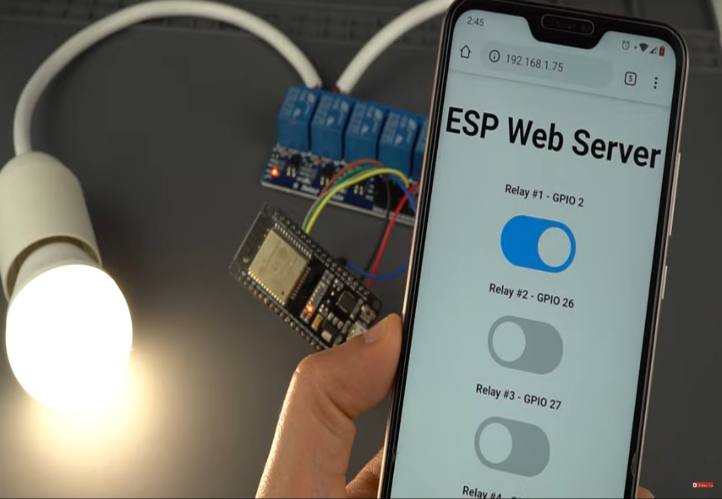
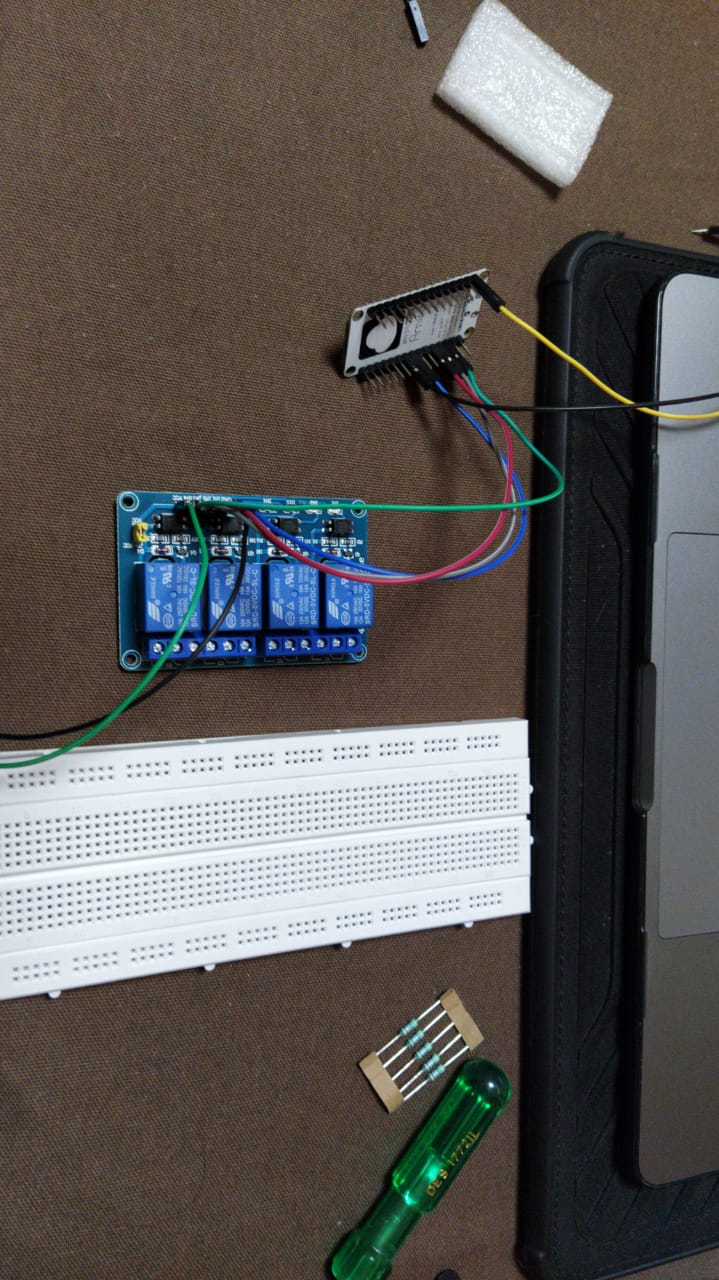
# 3 #: This is our loop to turn on or off the Devices that we configured on the Sinric Web server using unique Device IDs:



# 4 #: Finally, to interface Alexa with our device IDs from Sinric, we downloaded the Sinric Alexa Skill, which helped in establishing a successful connection between our devices and Alexa’s voice service. This method also eliminated the need to create an application as a front-end to control our devices since Amazon’s Alexa Mobile app serves this purpose.



PROTOTYPE STAGE:



* Initially, we used a custom webserver which was again readily available on GitHub for GPIO pin control.
* As a simple on-off switch to test our module for control with relay.
* We installed a few libraries for the custom webserver which included.

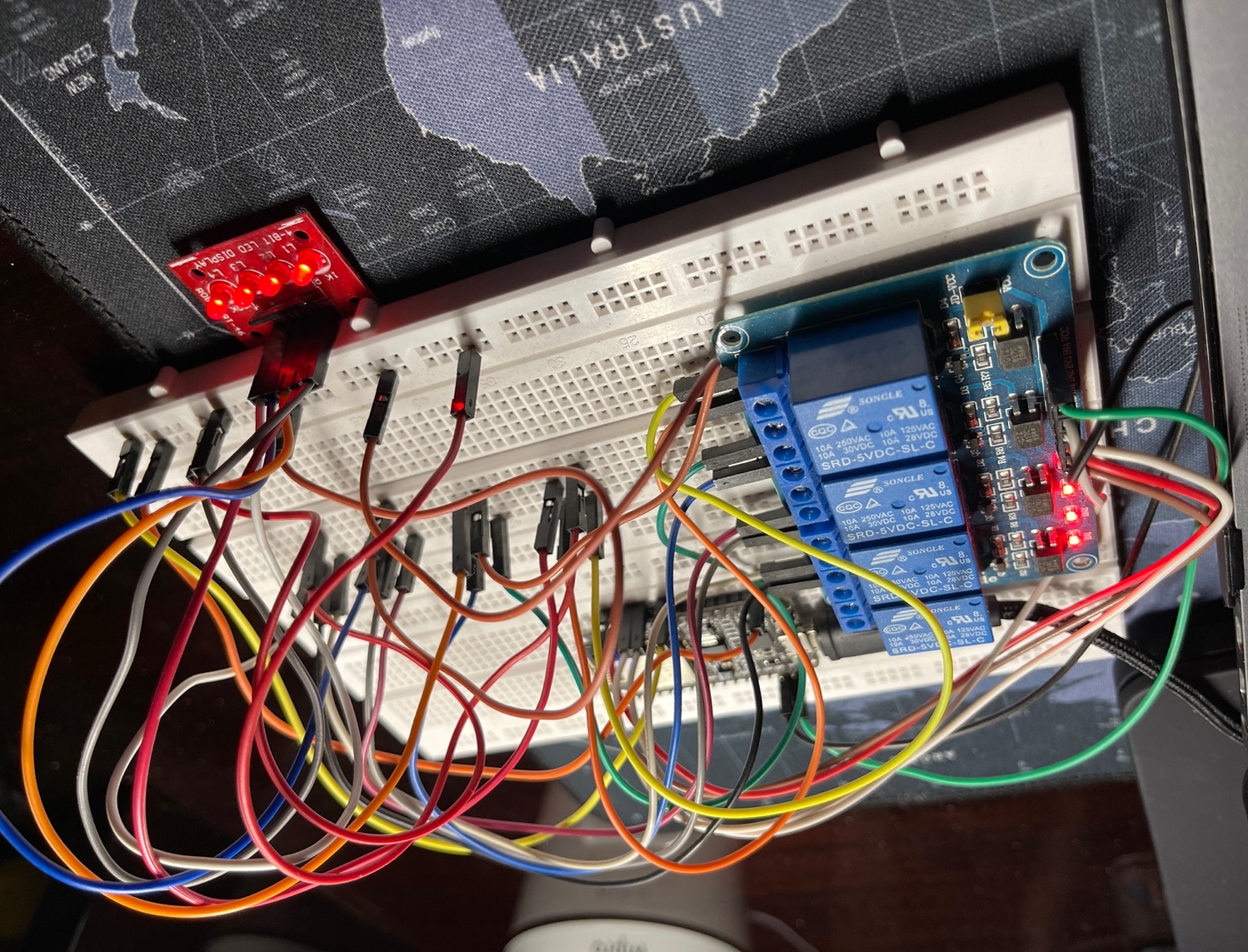
- ESPAsyncWebServer

-ESPAsyncTCP (ESP8266)

* This was a Transfer Control Protocol (TCP) which is not real-time just used for testing.
* Our Initial Tests were successful so we continued with our project using Sinric Web Services and Alexa for Home Automation.

**RESULTS AND ANALYSIS**

* After running initial tests we found that we were right in our approach to not complicate things by using Amazon Web Services or other Web Server based applications. To integrate this project with AWS/Web-Apps we would have needed to create a AWS Lambda function, create our own Alexa Skill, and most importantly our own app to control our devices among other complicated steps.
* Since our prototype was in proper working order, with no bugs in our code we proceeded ahead to add 4-Bit LED Display to the breadboard and tested for further additions to this setup, and we discovered that we can use bigger-channel relays (8,16 etc) and add even more devices.



Our Final Project Outcome

**CONCLUSION**

Since we were working on Home Automation, it only made sense to build this project while keeping in mind the cost effectiveness of the whole product. The hardware we used was readily available with us and it took very little time to set it up according to our Project topic and scope. As far as software was concerned, we used Arduino and custom libraries for ESP8266 like *fauxmo*, *WebSocket*, *Sinric (by Kakopappa)* and all of these were open source.

The code to integrate all parts of the project was also not very hard to find as most of the libraries we used came with their own code, we just had to change important parameters like WiFi SSID, Password, Device Names, etc. and hope and we had working hardware and didn’t get any bugs.

To view video demonstration of this project click [here](https://drive.google.com/file/d/16yQ25WbU5b2zLbNZ5npvUvEsImNtQ1uY/view?usp=sharing).

**FUTURE SCOPE OF THIS PROJECT:**

* Home Automation is already widely used in India and the rest of the world. Big players in this domain like Wipro, Philips Hue, Halonix, Oakter have already made their mark in India following similar approaches with their products (Tubelights, Smart Lights/Bulbs, Fans, Smart Switches, etc.)
* Automation is not just a word but a requirement of everyone in the future. Technology made it possible to control your home appliances with the help of mobile application or voice assistants.
* People in India are quickly adopting this technology but still, this technology is new for most people. You can set timers or run schedules on your appliances once you have made them smart, like turning on geyser at 7 AM automatically or turning on balcony lights at 8 PM when it is dark everyday .These devices will automatically run these commands according to the pre-defined schedules.
* This will be a revolution in the future to change simple homes into smart homes to make consumers more comfortable and add convenience to their life. Home automation will even help make your home secure as homeowners will be notified on their phone about any unusual activity in their smart home.
* Also, it will help homeowners to recognize who is ringing their home bell from the comfort of their phone. Everyone will opt for this technology happily because of the energy saving behaviour and more security accessibility features of smart homes. In short, home automation has a huge scope in the future.