# ECONOMICAL HOME AUTOMATION
# WITH VOICE
# CONTROLLING SYSTEM

**Group members:**

- Graham Stephan Silva
- Udaka Tharindu
- Bawantha Lakshan
- Sachini Nimeshika

# TABLE OF CONTENT

## LIST OF FIGURES:

## LIST OF TABLES:

# 1. 0 INTRODUCTION

## 1.1 Project Title

Home Automation

## 1.2 Problem Overview

Now a days in our busy life we have to face many types of difficulties. Although many designs have been created to solve these problems, today they are limited to books and little in the practical world usage is minimal. We decided to innovate our project by using the knowledge gained from research papers and books. Compared to foreign countries, our country uses smart technology sparingly and it is only around in Colombo. (Especially the wealthy ones) Many people are reluctant to embrace this advanced technology because they do not understand it. And some parties are of the opinion that this is used only by wealthy people. They are also of the opinion that it will be costly. Surely it will consume a bit high amount if Google products or Amazon products are used. But with this concept any electronic device can be converted into a smart object and can be controlled via Google Assistant or Alexa. Our project focuses on people with disabilities and the elderly. Here, our project is designed to be accessible to the public.

## 1.3 Scope

In our busy life most, valuable thing is time. Therefore, we use to use modern technology to make our day-to-day work easy. So, we decided to create a smart room as our project by using modern technology. Under the smart room we decided to control room lights, fan, and door lock by giving voice controlling commands.

## 1.4 Aim

Aim of this project is to implement low cost, voice controlling home automation system. It can be used to switch on or off any devices in home control by using Google Assistant.

## 1.5 Objectives

This project is design to use the voice recognition technology to control home devices. It's specially focus on needs of disabled people. Google voice recognition facility will be devices to input voice.

Like ever product, there are pros and cons here. The pros are,

- We can use our product in hospitals because this is very useful for patients who are disable and elders.
- Cost of this product is very low.
- When we use this product, we can save energy and power.
- By using this product, we can connect high and low voltage devices.
- This is user-friendly.

- Cheaper than any Google Home or Amazon products.

The cons are,

- When there is a power failure, we cannot use our product. This is the main disadvantage we face.
- If there are some signal issues, then the product will not work properly.

# 2.0 LITRETURE SURVEY

## 2.1 RESEARCH PAPER 1

### SUMMARY

Home automation in India is already the buzz word, especially as the wave of second-generation homeowners grows. Modern homes are automated through the internet and the home appliances are controlled. These smart automation devices happen to have an interface with Internet of Things devices. System uses Google Assistant, IFTTT application, the Blynk application and the NodeMCU microcontroller. All these components are connected over the internet using Wi-Fi.

Human voice is used to give commands to the Google Assistant. System consists of two main categories: 1. Hardware and 2. Software. NodeMCU is an open source lua based firmware and development board. It runs on ESP8266 Wi-Fi SoC from Espressif Systems and is based on the ESP-12E module.

IFTTT's free tier no longer allows for unlimited applet creation; the maximum is set at three. Users that have created more than three will need to either pay for the Pro version or delete some applets. Arduino is an open-source electronics platform based on easy-to-use hardware and software. The Arduino modules can be assembled by hand and cost less than $50 each. It's easy-to-use and easy to understand for beginners, but flexible enough for advanced users as well.

After downloading the Blynk app, there is need to create a New Blyndk account. An account is needed to save the projects and have access to them from multiple devices. The Auth Token is a unique identifier needed to connect the hardware to the smartphone. Go to the Arduino wide website or windows store and install the ide. Next step is downloading the blynk library from GitHub.

Connect the Nodemcu to the breadboard then connect it to the wi-fi via Pc. When this is done, verify and upload the code to the board.

The aim of the proposed system was to implement a low cost, reliable and scalable voice-controlled home automation system. It can be used to switch on/off any home appliances as well as search the internet and control devices such as Google Home and other smart appliances. The purpose of this project is to make homes simpler, better, or more accessible. The future scope for google assistant-based home automation can be huge. There are many factors that make the project more reliable, intelligent, scalable, and to become better overall.

Research paper link.

https://ijtre.com/wp-content/uploads/2021/09/2021080504.pdf

## 2.2 RESEARCH PAPER 2

### SUMMARY

Home automation includes controlling aspects of home, remotely through electronic devices. In this paper, we are going to implement the home automation concept which is voice controlled. The entire communication between the devices is based on internet i.e., with Wi-Fi and appliances can be operated from anywhere. The idea is to control the different kind of home appliances by taking the voice command from the user via Google Assistant. We use a Smartphone for providing user commands using the Google assistant and NodeMCU, of course with Wi-Fi connectivity to access and control the devices.

The NodeMCU is an open source IoT hardware development board. It contains the Wi-Fi module (ESP8266) used to connect the devices with the network. The relay module can be used as series of switches which is used to turn ON or OFF the appliances. Google Assistant is a virtual assistant software based on Artificial Intelligence. The users can give the voice commands through Google Assistant to access smart devices and applications. IFTTT is a service where a user can program a response to event using simple conditional statements.

Arduino is an open-source software that uses Java as its programming language. The code written by user consists of two functions, setup and loop which are compiled, linked with main function. Arduino IDE is used to compile the program and to flash the code into Arduino board. With the help of Google Assistant, user gives voice commands and when these commands are received, then data is interpreted and analyzed whether the voice command is meant for IFTTT or Adafruit IO. Voice commands provided by users are decoded and given to NodeMCU (ESP8266). Code is uploaded onto NodeMCU board via Arduino via Arduino and connected to Wi-Fi.

Command 'turn light ON' is given through Google assistant to switch ON the light. The command 'turn light OFF' is to switch OFF the light.

Research paper link.

https://iaeme.com/MasterAdmin/Journal_uploads/IJARET/VOLUME_12_ISSUE_3/IJARET_12_03_014.pdf

## 2.3 COMPARISON

- Instead of ESP8266 module, ESP 32 has been used because of the following advantages

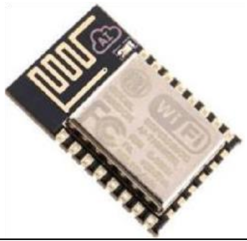
| | **ESP8266** | **ESP32** |
|---|---|---|
| |  | |
| **MCU** | Xtensa Single-core 32-bit L106 | Xtensa Dual-Core 32-bit LX6 with 600 DMIPS |
| **802.11 b/g/n Wi-Fi** | HT20 | HT40 |
| **Bluetooth** | **X** | Bluetooth 4.2 and BLE |
| **Typical Frequency** | 80 MHz | 160 MHz |
| **SRAM** | **X** | ✓ |
| **Flash** | **X** | ✓ |
| **GPIO** | 17 | 34 |
| **Hardware /Software PWM** | None / 8 channels | None / 16 channels |
| **SPI/I2C/I2S/UART** | 2/1/2/2 | 4/2/2/2 |
| **ADC** | 10-bit | 12-bit |
| **CAN** | **X** | ✓ |
| **Ethernet MAC Interface** | **X** | ✓ |
| **Touch Sensor** | **X** | ✓ |
| **Temperature Sensor** | **X** | ✓(old versions) |
| **Hall effect sensor** | **X** | ✓ |

*Table 1- Comparison of ESP8266 and ESP32*

Instead of Blynk as the third party application Sinric-Pro has been used because of the advanced options compared to Blynk app as well as the easiness of connecting with Google home.

# 3.0 THEORITICAL BACKGROUND

## 3.1 ESP 32

*Figure 1 - ESP 32*

The ESP32 is a low power system on a chip microcontroller which integrated WiFI and dual mode Bluetooth.

There are mainly two processors in the board, a CPU which consists of Xtensa dual-core 32-bit LX6 microprocessor, operating at 160 or 240 MHz and performing at up to 600 DMIPS and an Ultra-Low Power (ULP) co-Processor. It has a memory of 320KiB RAM, and 448 KiB ROM. ESP-32 use a WiFi 802.11 b/g/n and Bluetooth version 4.2 BR/EDR and BLE, the new versions of ESP-32 module have upgraded to Bluetooth version 5.0, which has higher stability and compatibility.

When it comes to interface it consists of 34 programmable GPIOs (General Purpose Input/Output – which can be used as both inputs and outputs),12-bit SAR (Successive Approximation) ADC (Analog to Digital Converter) up to 18 channels, 2× 8 bit DACs (Digital to Analog Converter), 10 touch sensors/ capacitive sensing GPIOs (a technology that can detect and measure anything conductive or anything that is dielectric from air), 4 SPI (Serial Peripheral Interface – a serial communication interface specifically used for short distance communication), 2×I$^2$S interfaces (serial bus interface specifically used for connecting digital audio), 2×I$^2$C interfaces (Interintegrated circuit), 3 UARTs (Universal Asynchronous Receiver Transmitter), SD/SDIO/CEATA/MMC/eMMC host controller, SDIO/SPI slave Controller, Controller Area Network (CAN) 2.0, Infrared remote controller (TX/RX to 8 channels), Motor PWM (), LED PWM (to 16 channels), Hall Effect Sensor, Ultra Low Power Analog pre-Amplifier.

When it comes to power management, it has internal low dropout regulator as well as individual power domain for RTC. It has a special feature of 5 µA deep sleep current and it can wake up from GPIO interrupt, timer, ADC measurements, capacitive touch sensor interrupt. It has 5v and 3v Outputs for the requirement.

## 3.2 Relay Module

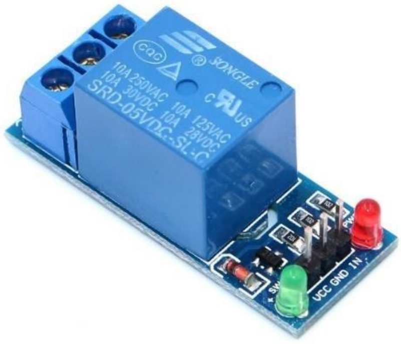


*Figure 2 - Relay module*

An electromagnet operates a power relay module, which is an electrical switch. A separate lowpower signal from a microcontroller activates the electromagnet. The electromagnet pulls to open or close an electrical circuit when energized.

A simple relay is made up of a wire coil wrapped around a soft iron core, known as a solenoid, an iron yoke that provides a low reluctance channel for magnetic flux, a moveable iron armature, and one or more sets of contacts. The moveable armature is connected to one or more sets of moving contacts and is hinged to the yoke. When the relay is de-energized, the armature, which is held in place by a spring, leaves a gap in the magnetic circuit. One of the two sets of contacts is closed in this configuration, while the other stays open.

When an electrical current passes through a coil, a magnetic field is created, which activates the armature. The movable contacts' movement establishes or breaks a connection with the fixed contact. When the relay is de-energized, the closed sets of contacts open and the connection is broken, and vice versa if the contacts are open. When the current to the coil is shut off, the armature is forced back to its relaxed position. A spring is normally used to create this force, but gravity can also be employed in some cases. The majority of power relays are designed to function quickly.

## 3.3 LCD Display 16×02

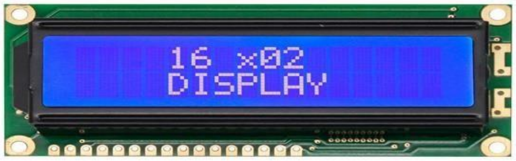


*Figure 3 - LCD display*

LCD 162 stands for Liquid Crystal Display, which is a flat panel display technology used in computer monitors and televisions, smartphones, tablets, and other mobile devices. Both LCD and CRT screens have the same appearance, yet they operate differently. A liquid crystal display has a backlight that provides light to each pixel in a rectangular network instead of electron diffraction at a glass display.

Each pixel has a blue, red, and green sub-pixel that may be turned on and off. It will appear black when all of these pixels are deactivated, and white when all of the sub-pixels are engaged. Different colour combinations can be achieved by varying the brightness of each light. This page provides an overview of the LCD 16X2 and how it interacts with apps.

LCD 162 is a type of electronic display device that shows data and messages. As the name implies, it has 16 columns and 2 rows, allowing it to display a total of 32 characters (162=32), each of which is made up of 588 (40) Pixel Dots. As a result, the total pixels in this LCD are 32 x 40, otherwise 1280 pixels.

Multi-segment LEDs are used extensively in 16 X2 screens. There are several types of displays available on the market with various combinations, such as 82, 81, 161, and 102, but the LCD 162 is widely used in devices, DIY circuits, and electronic projects due to its low cost, programmability, and ease of use.

## 3.4 Regulator (5v)

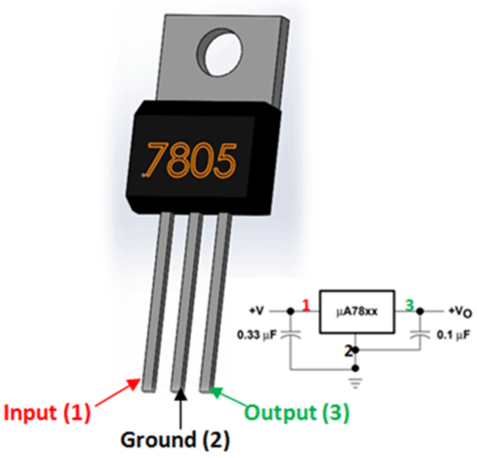


*Figure 4 - Regulator (5v)*

### What is voltage regulator?

Voltage regulator is a circuit that creates and maintains constant output voltage regardless of changes in input voltage or load conditions. Choosing the right voltage regulator is the one of the most important decisions in electronic circuit design. Regulator as the name suggests, voltage regulators take variable or unstable input voltages and then convert then to higher or lower constant output that matches the voltage and current requirements of an electronic circuit. Basic linear IC type regulators simply step down source to desired level and save as heat, while other switching types are most efficient. Voltage regulator is any electronic device or electrical device maintain the voltage of power source within acceptable limits. Voltage regulator is required to keep voltages within specified range that electrical equipment uses voltage uses withstand voltage.

### What is 5v regulator?

5v regulator is the basic L7805 voltage regulator, three-terminal positive regulator with 5V fixed output voltage. This fixed regulator provides local regulation, internal current limiting, thermal shut-down control, and safe area protection for the project.

### How it works?

This circuit is a TS7805CZ (5V 1A) regulator, which then steps the 12V voltage signal down to 5V, and pushes this on output. The 0.1$\mu$F capacitor then further cleans DC signal, leaves us with nice clean 5v source.

**Uses.**

Every product that operates on DC power uses voltage regulation. Electronic voltage regulators found in devices where stabilize DC voltage used by processor and other components. In automobile alternators and central plant generators, voltage regulators control the output of plant. For 7805 IC, it is +5V DC regulated power supply. This regulator IC also adds a provision for a heat sink. The input voltage to this voltage regulator can be up to 35V, and this IC can provide a constant 5V for any input value less than or equal to the threshold limit of 35V.

**How do you make**

To select voltage regulator for the system, start by assuming linear regulator used if input voltage is higher than output. If you want to higher output voltage than input voltage then use boost switching regulator.

To make 5V regulator we need only 3 components. They are, 7805 IC, two capacitors ($33\mu F$ and $0.1\mu$F).

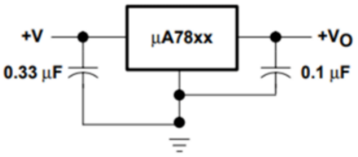


*Figure 5 - Regulator circuit diagram*

The input capacitor 0.33uF is a ceramic capacitor that deals with the input inductance problem and the output capacitor 0.1uF is a ceramic capacitor that adds to the stability of the circuit. These capacitors must be placed close to the terminals to work effectively. Ceramic capacitors should be of ceramic type as they are faster than electrolytic.

## 3.5 I2C module

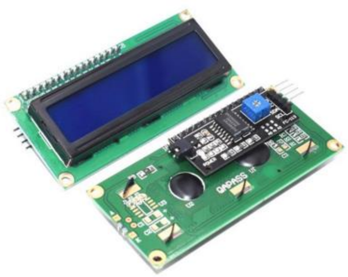


*Figure 6 - I2C module*

This 16x2 LCD display module has an I2C communication interface and is a premium 2 line, 16 character LCD module with on-board contrast control adjustment, backlight, and I2C interface. There is no longer a need for a difficult LCD driver circuit connection for Arduino beginners. The I2C Serial LCD Module's real benefits will be in terms of the Arduino board's I/O pins may be saved, the circuit connection is made simpler, and firmware programming is made easier with widely. Arduino libraries are available.

Any project involving the display of information must include Hitachi's HD44780-based character LCDs because they are reasonably priced, widely available, and indispensable. Desired data can be presented on the LCD using the LCD piggy-back board and the I2C bus. These backpacks are conceptually constructed around the PCF8574, a general-purpose bidirectional 8 bit I/O port expander that makes use of the I2C protocol. For most microcontroller families, the PCF8574 is a silicon CMOS circuit that offers general-purpose remote I/O extension across a two-line bidirectional bus. The majority of piggy-back modules, with a default slave address of 0x27, are centred around the PCF8574T. Your piggy-back board's default slave address will now be 0x3F if it has a PCF8574AT chip. In essence, the slave address 0x27 will be used if the piggy-back board is based on PCF8574T and the address connectors (A0-A1-A2) are not bridged with solder.

## 4.0 PROPOSED METHODOLOGY

As to fulfil the concept, several electronic models are to be used such as ESP 32, Relay modules, Lcd Display, I2C Interface, potentiometer, resistor and 5v voltage regulators. Apart from these main components a household bulb, cooling fan (as a replacement for a normal stand fan), LED light, A/C and D/C power sources and jumper wires are beeng used.

The main concept behind the project is to link all the household electronic items to a single device so that the electronics can be controlled from any place. The idea is to link the electronic devices through Wi-Fi instead of Bluetooth, because people can pair multiple devices with a satisfied range as well as it can be controlled with a mobile device even when the mobile device is not in the range of Wi-Fi which is connected to the electronic devices. ESP 32 will connect with a Wi-Fi network so the IP address will be shared, when a signal is sent to the IP address a task can be performed through the ESP 32 module, this is the basic concept of this project.

To connect ESP 32 to the Wi-Fi network, the module is programmed by Arduino IDE. In order to give commands through voice, Google Assistant is used. To connect Google Assistant a third-party application is linked with Google Assistant. Sinric-Pro is the application used to link the ESP 32 with Google Assistant.

Initially a room was created in sinric-pro and add up the devices to it, then device IDs were automatically be generated by the application. These IDs are controlled via ESP 32. A household Bulb (CFL or LED), a fan and a lamp are the devices that's used as the devices in the room. LCD display is used for the purpose to display whether the connection of ESP 32 with Wi-Fi is success or not, after a successful connection, the devices are given a task to perform on command by programming. In order to control the A/C components a relay module is used to pass the digital signal to the A/C devices and the relays are powered by 12v adapter with 5v regulator.

Afterwards the access to the third-party applications in Google Home was allowed by selecting Sinric-Pro. Then a routine was created individually to each device by login to the room that has been created and provided a specific function to be called.

Finally, when a voice command is given through Google assistant, the device will perform the specific task. Not only by voice it can be controlled through the Google Home app as well.

# 5.0 DESIGN OF HARDWARE AND SOFTWARE

## 5.1Hardware Design

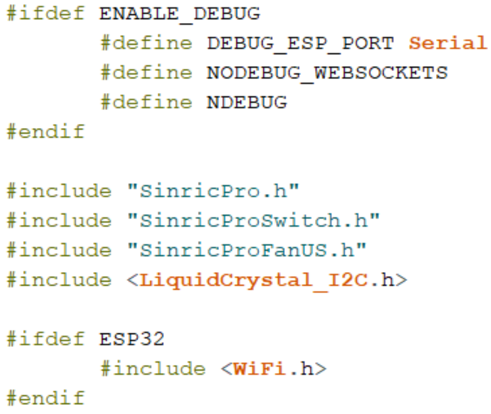
The cooling fan and the household bulb (either CFL or LED) is connected to two relays respectively. The input for the fan is given by the GPIO 5 of ESP-32 chip and the input to the relay which the bulb is connected is given by the pin 2 of ESP-32. Lamp is connected to the pin 4 of the ESP-32 chip with a resistor. All the relays are powered by a 12v Adapter and passed through a 5v regulator. Both the Bulb and the fan is connected to the NC (Normally Closed) pin and bulb is powered with 230v AC supply and the fan is powered by a 12v Adapter. Black and Red wires of the fan is connected to the 12v and to COM port in relay respectively.

16x02 LCD Display is used to present the current state of the project, for example if a voice command is given to switch on the fan, then the display will print the fan is ON. Since there are 16 pins in the display to reduce the complexity of the project an I2C module is used. Basically it converts the 16 pins into 4 main pins as Vcc,GND,SDA and SCL. Apart from that, two pins to control the backlight of the LCD Display is connected to a potentiometer to reduce and to increase the brightness of the backlight of the LCD display. Vcc is connected to 5v output and SDA and SCL are connected to the pin 21 and 22 respectively.

For designing the structure of the project, a prototype living room is made using cladding boards which is 12in x 12in and 5in width one end opened.

## 5.2 Software Design

### 5.2.1 Declaration of libraries



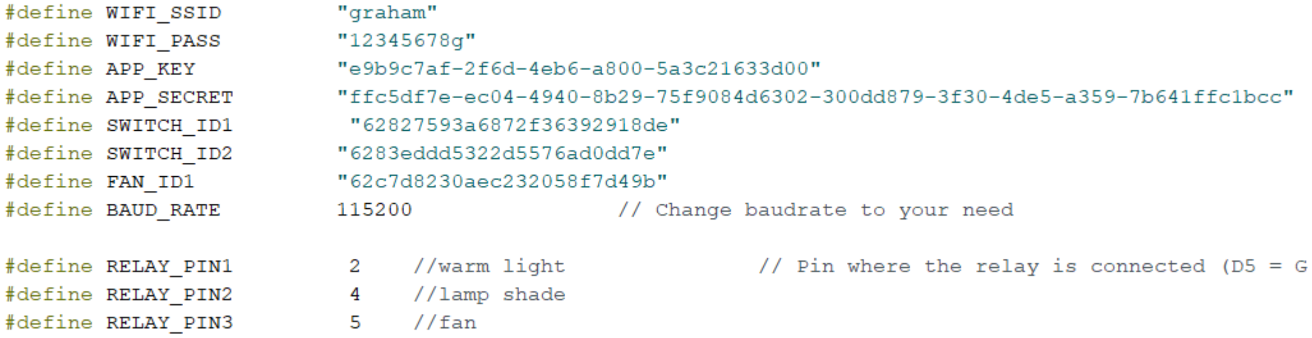
```
#ifdef ENABLE_DEBUG
       #define DEBUG_ESP_PORT Serial
       #define NODEBUG_WEBSOCKETS
       #define NDEBUG
#endif

#include "SinricPro.h"
#include "SinricProSwitch.h"
#include "SinricProFanUS.h"
#include <LiquidCrystal_I2C.h>

#ifdef ESP32
       #include <WiFi.h>
#endif
```

The above code is used to declare the sinric pro libraries to link the Arduino with sinric pro, the library of LCD display with I2C module and ESP32 with Wifi. It also enables the debugging.

### 5.2.2 Declaration of Wi-Fi and ESP32 pins

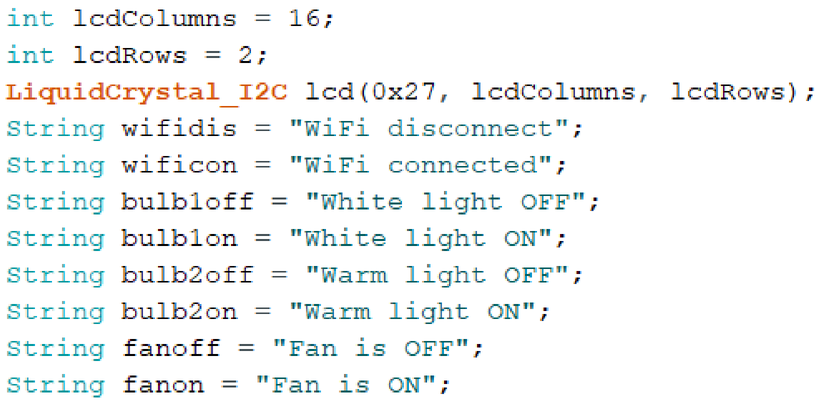


```
#define WIFI_SSID         "graham"
#define WIFI_PASS         "12345678g"
#define APP_KEY           "e9b9c7af-2f6d-4eb6-a800-5a3c21633d00"
#define APP_SECRET        "ffc5df7e-ec04-4940-8b29-75f9084d6302-300dd879-3f30-4de5-a359-7b641ffc1bcc"
#define SWITCH_ID1         "62827593a6872f36392918de"
#define SWITCH_ID2        "6283eddd5322d5576ad0dd7e"
#define FAN_ID1           "62c7d8230aec232058f7d49b"
#define BAUD_RATE         115200                // Change baudrate to your need

#define RELAY_PIN1         2    //warm light                // Pin where the relay is connected (D5 = G
#define RELAY_PIN2         4    //lamp shade
#define RELAY_PIN3         5    //fan
```

Here, the Wi-Fi SSID and password, Sinric-Pro keys and device Ids, baud rate and pin assignments are defined.

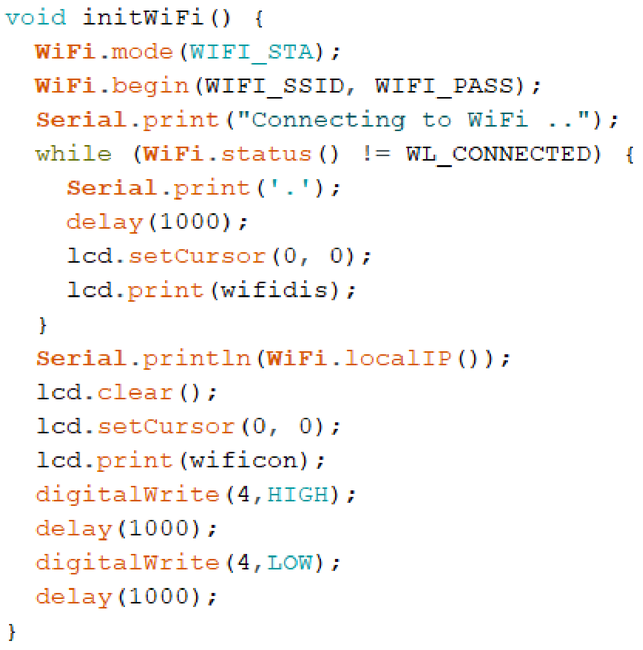
### 5.2.3 LCD Display declarations



```
int lcdColumns = 16;
int lcdRows = 2;
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
String wifidis = "WiFi disconnect";
String wificon = "WiFi connected";
String bulb1off = "White light OFF";
String bulb1on = "White light ON";
String bulb2off = "Warm light OFF";
String bulb2on = "Warm light ON";
String fanoff = "Fan is OFF";
String fanon = "Fan is ON";
```

Declaration of the number of columns and rows in the display and assigning it to an array to print the commands given is done by the above code.

### 5.2.4 Wi-Fi connectivity status



```
void initWiFi() {
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
    lcd.setCursor(0, 0);
    lcd.print(wifidis);
  }
  Serial.println(WiFi.localIP());
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(wificon);
  digitalWrite(4,HIGH);
  delay(1000);
  digitalWrite(4,LOW);
  delay(1000);
}
```

In the above code, command to connect to the wifi is given by Wifi.begin function by giving the SSID and the password of the relevant wifi. If the Wi-Fi Is not connected the serial monitor keeps adding dots(.) in front of the “Connecting to WIfi .. ” in a while loop until its getting connected and print “WiFi disconnect” on the LCD Display, else it will print “WiFi connected” on the LCD Display and it will print the local Ip address of the network in the serial port.

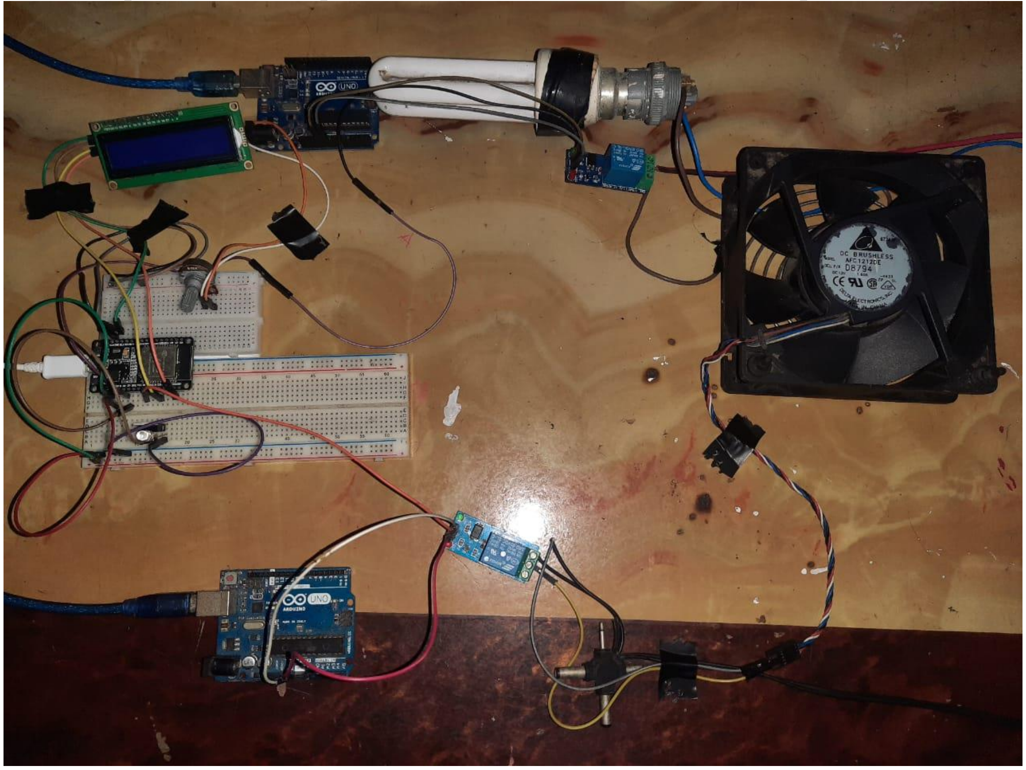


*Figure 7 - Prototype of the circuit digram*

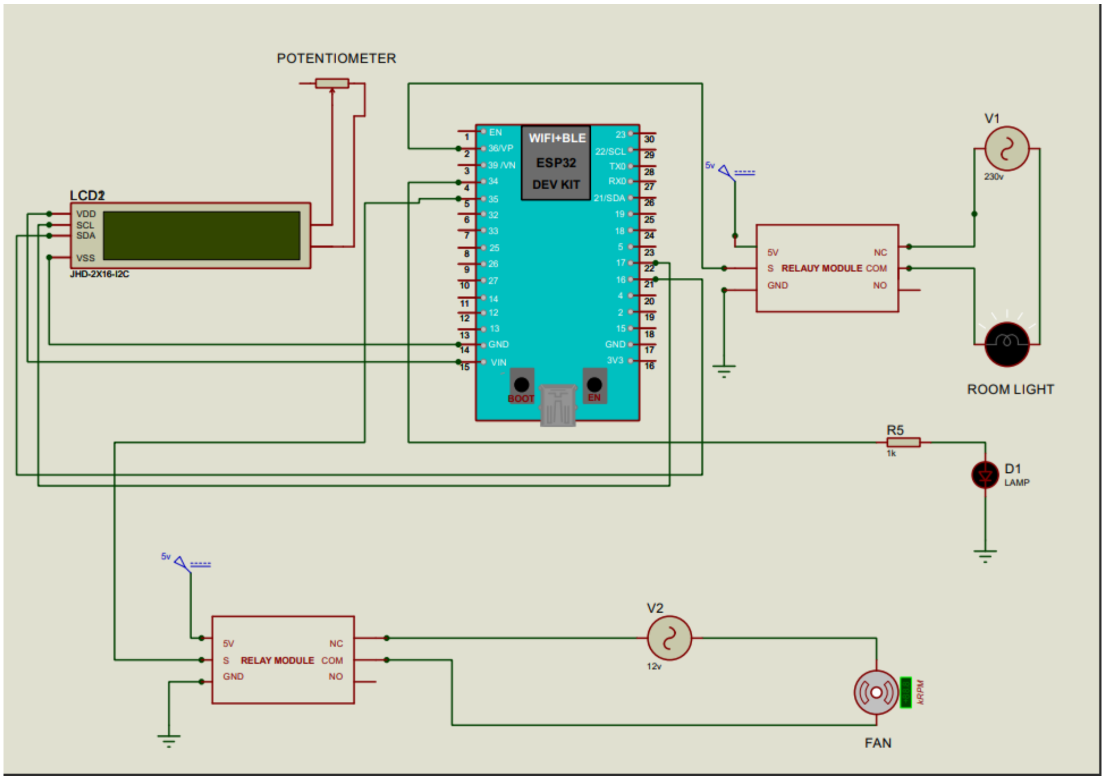
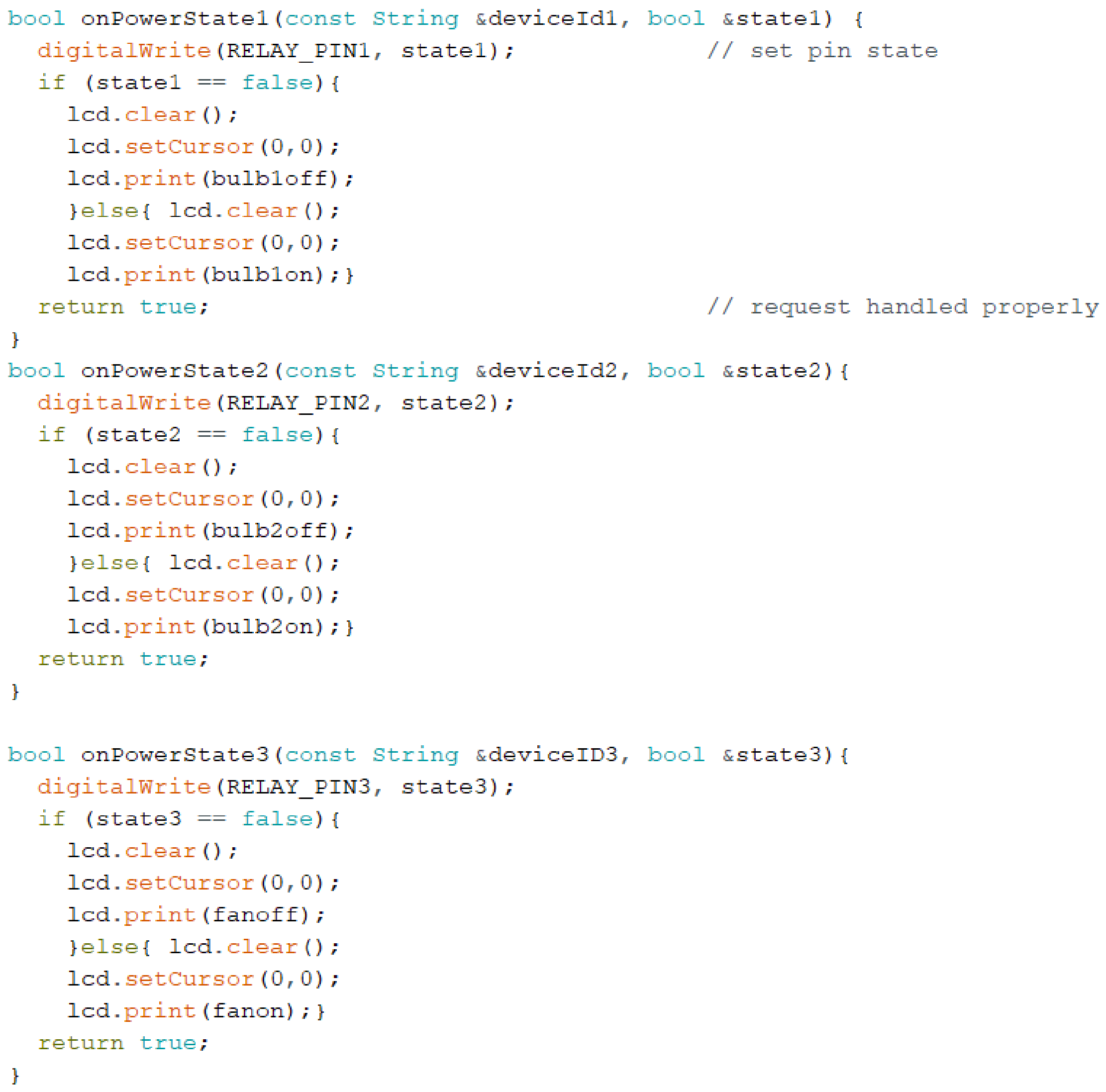


*Figure 8 - Circuit diagram*

### 5.2.5 Setting up the Functions

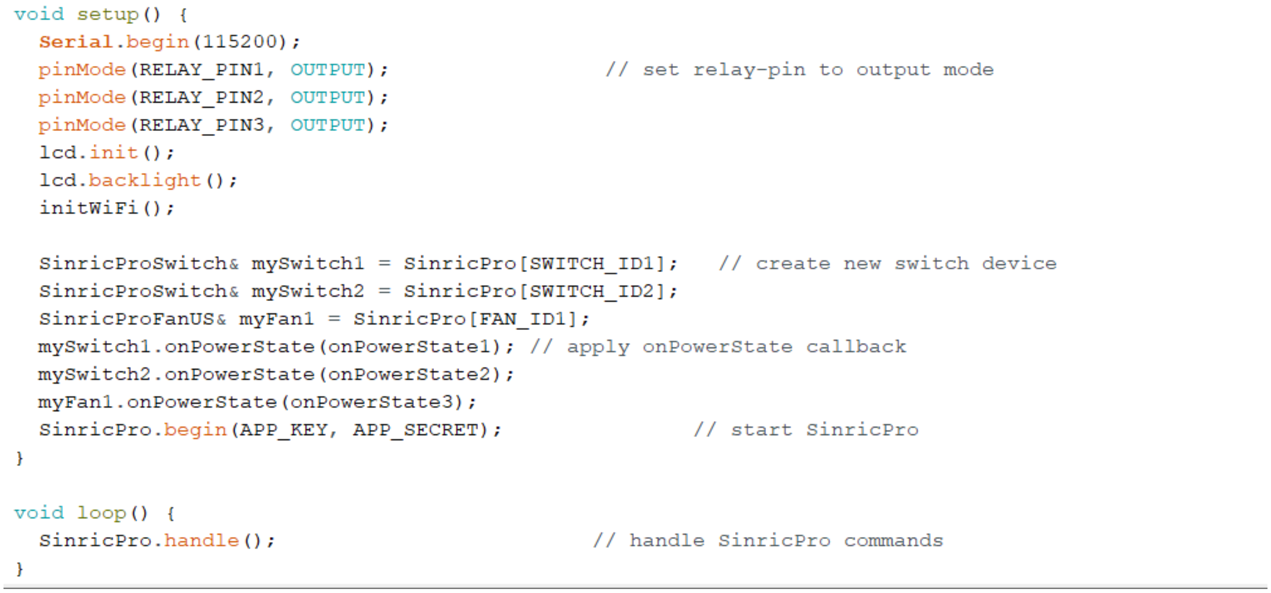


```
bool onPowerState1(const String &deviceId1, bool &state1) {
  digitalWrite(RELAY_PIN1, state1);             // set pin state
  if (state1 == false){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(bulb1off);
    }else{ lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(bulb1on);}
  return true;                                  // request handled properly
}
bool onPowerState2(const String &deviceId2, bool &state2){
  digitalWrite(RELAY_PIN2, state2);
  if (state2 == false){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(bulb2off);
    }else{ lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(bulb2on);}
  return true;
}

bool onPowerState3(const String &deviceID3, bool &state3){
  digitalWrite(RELAY_PIN3, state3);
  if (state3 == false){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(fanoff);
    }else{ lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(fanon);}
  return true;
}
```

In the above code, three functions are declared as onPowerState by giving the corresponding device Ids. If the function is false, the LCD display will print that the device been switched off and else the display will print it the device is turned on.

### 5.2.6 Setup and the loop function



```
void setup() {
  Serial.begin(115200);
  pinMode(RELAY_PIN1, OUTPUT);                 // set relay-pin to output mode
  pinMode(RELAY_PIN2, OUTPUT);
  pinMode(RELAY_PIN3, OUTPUT);
  lcd.init();
  lcd.backlight();
  initWiFi();

  SinricProSwitch& mySwitch1 = SinricPro[SWITCH_ID1];   // create new switch device
  SinricProSwitch& mySwitch2 = SinricPro[SWITCH_ID2];
  SinricProFanUS& myFan1 = SinricPro[FAN_ID1];
  mySwitch1.onPowerState(onPowerState1); // apply onPowerState callback
  mySwitch2.onPowerState(onPowerState2);
  myFan1.onPowerState(onPowerState3);
  SinricPro.begin(APP_KEY, APP_SECRET);               // start SinricPro
}

void loop() {
  SinricPro.handle();                            // handle SinricPro commands
}
```

In this code, the serial communication is begun by 115200 baud rate and the states/modes of the pins are clarified. Lcd and the lcd backlight functions are also called back. Afterwards for each device a switch is created in sinric-pro and apply the on Power State function declared previously to each corresponding device, then connect to the sinric-pro by providing the relevant keys.

Finally in the loop function, it allows sinric-pro to handle all the commands which is given accordingly.

# 6.0 Testing and Implementation

## 6.1 Wi-Fi Disconnect and Connect

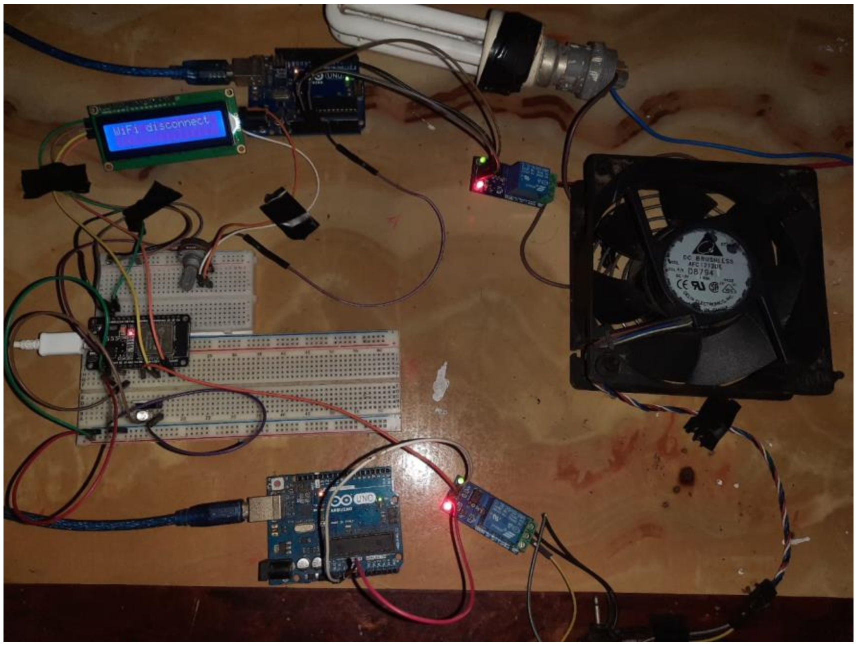


*Figure 9 -Testing Wi-Fi when not connected*

In the above figure it shows when the Wi-Fi is not connected to the ESP_32 chip, the LCD Display shows that the Wi-Fi is disconnect.

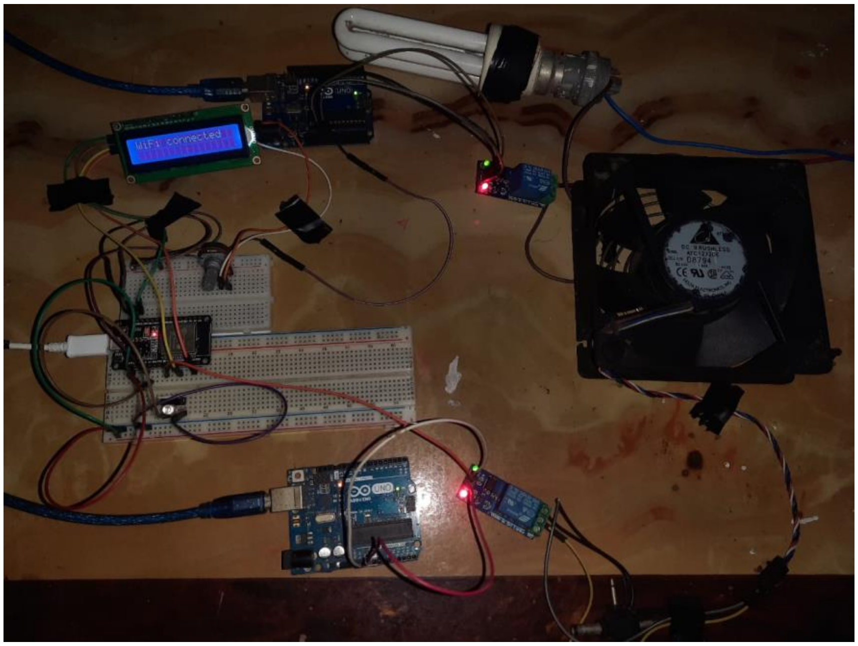
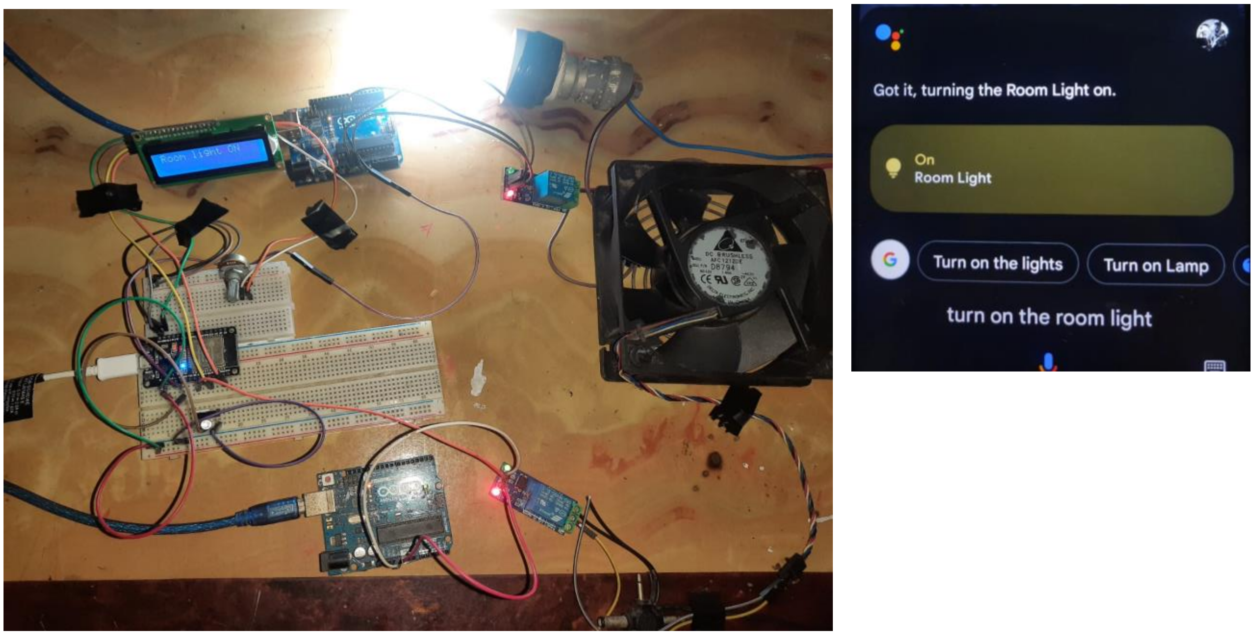


*Figure 10 - Testing Wi-Fi when connected*

In the above figure it shows when the Wi-Fi is connected to the ESP_32 chip, the LCD Display shows that the Wi-Fi is connected.

## 6.2 Bulb Switch ON and OFF



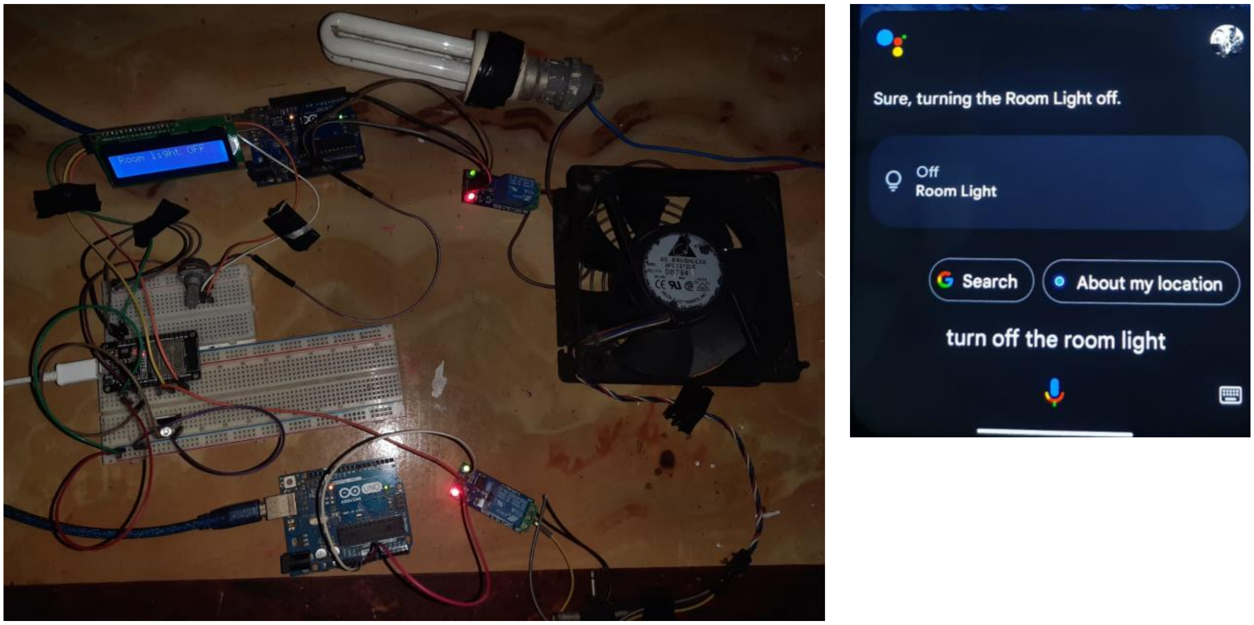
Got it, turning the Room Light on.

On
Room Light

turn on the room light

*Figure 11 - Bulb switch on when according to the voice command*

In the above figure it shows when a voice command is given to switch on the bulb, the blub will turn on and the LCD Display shows that the bulb is on.



Sure, turning the Room Light off.

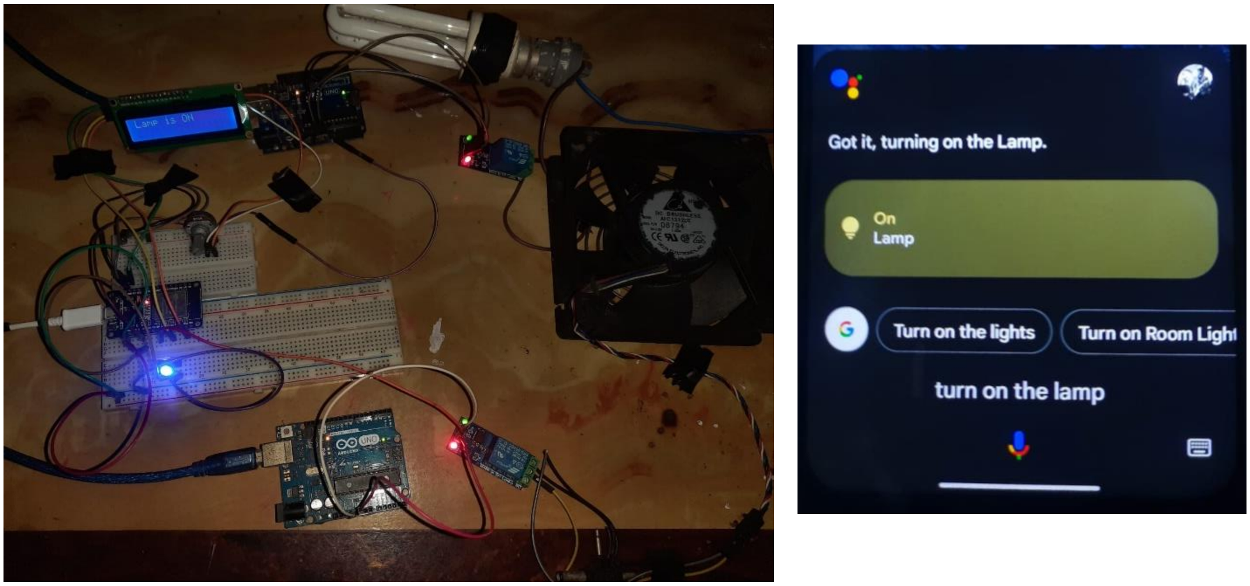
Off
Room Light

turn off the room light

*Figure 12 - Bulb switch off when according to the voice command*

In the above figure it shows when a voice command is given to switch off the bulb, the bulb will turn off and the LCD Display shows that the bulb is off.

### 6.3 Lamp Switch ON and OFF



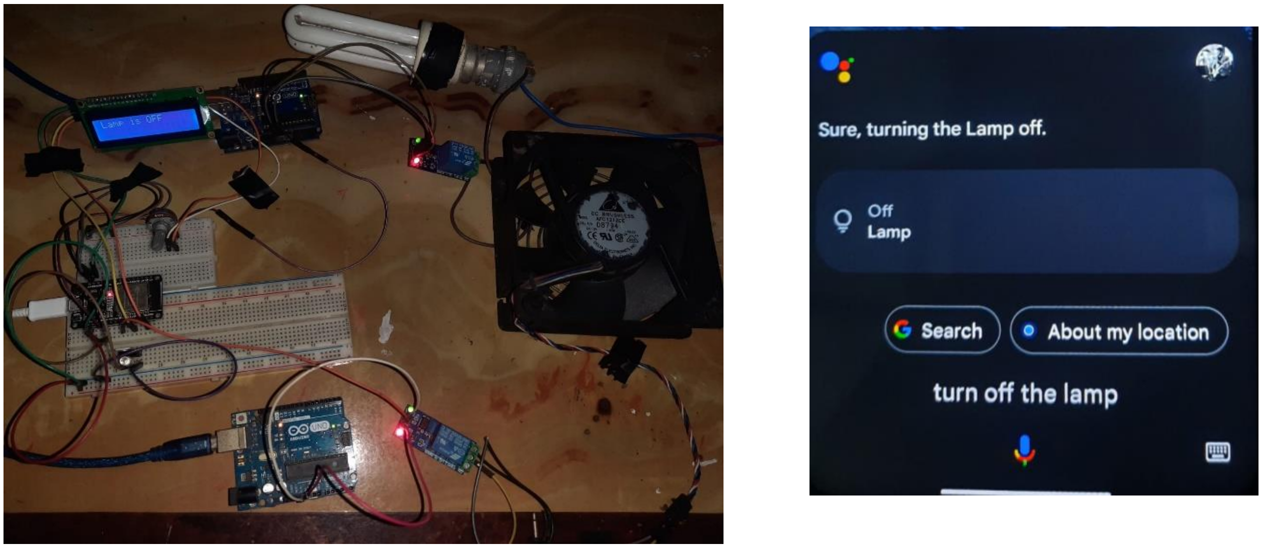
Got it, turning on the Lamp.

On
Lamp

turn on the lamp

*Figure 13 - Lamp is on according to the voice command*

In the above figure it shows when a voice command is given to switch on the lamp, the lamp will turn on and the LCD Display shows that the lamp is on.



Sure, turning the Lamp off.

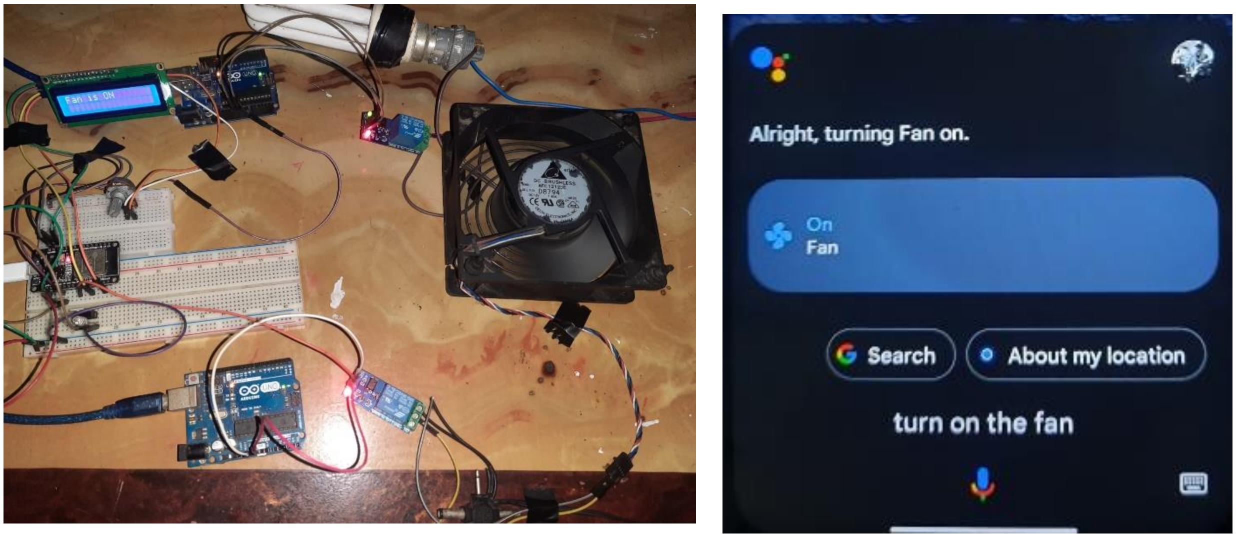
Off
Lamp

turn off the lamp

*Figure 14 - Lamp is on according to the voice command*

In the above figure it shows when a voice command is given to switch off the lamp, the lamp will turn off and the LCD Display shows that the lamp is off.

### 6.4 Fan Switch ON and OFF



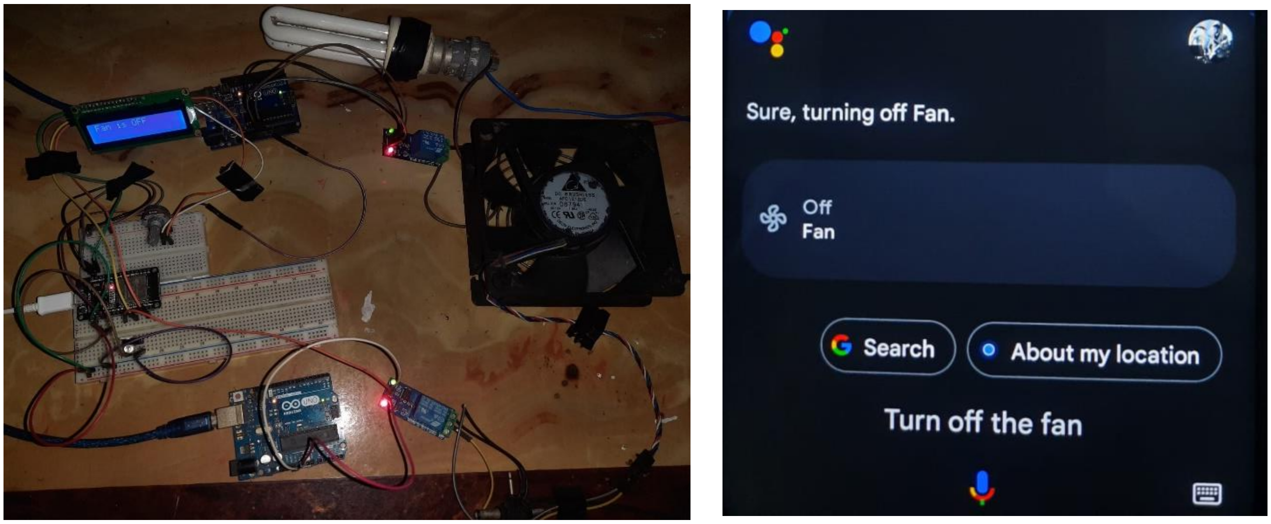
Alright, turning Fan on.

On
Fan

turn on the fan

*Figure 15 - Fan switch on according to the voice command*

In the above figure it shows when a voice command is given to switch on the fan, the fan will turn on and the LCD Display shows that the fan is on.



Sure, turning off Fan.

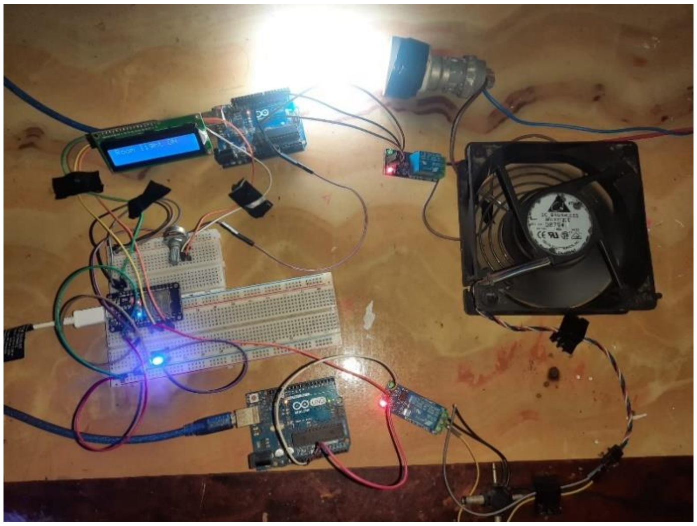
Off
Fan

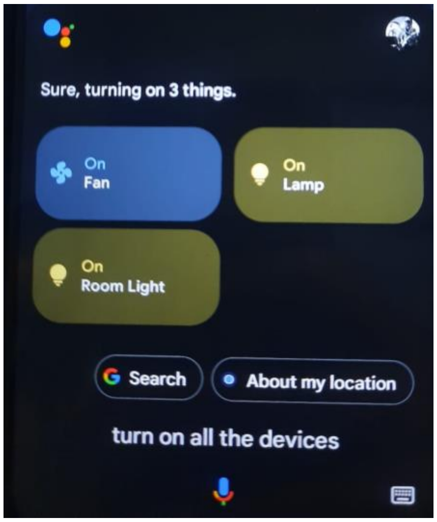
Turn off the fan

*Figure 16 - Fan switch off according to the voice command*

In the above figure it shows when a voice command is given to switch off the fan, the fan will turn off and the LCD Display shows that the fan is off.

## 6.5 Multiple Devices Switch ON and OFF



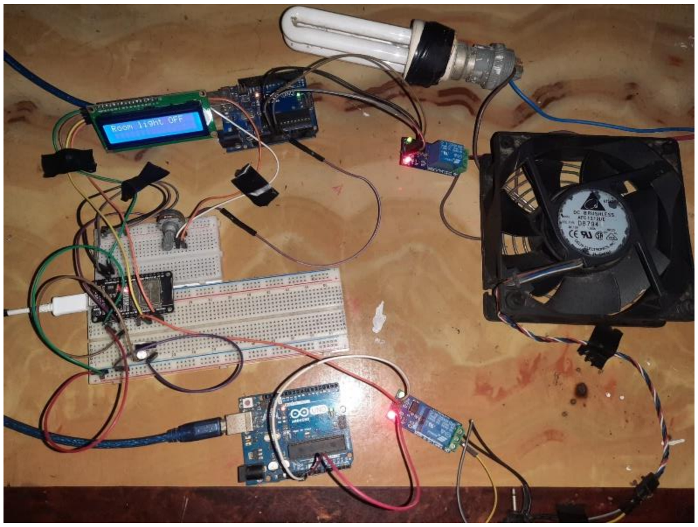


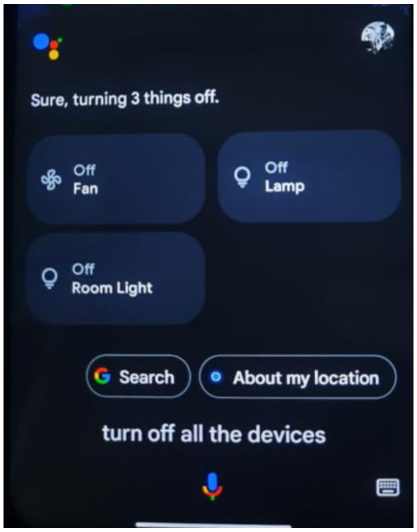
Sure, turning on 3 things.

turn on all the devices

*Figure 17 - Multiple devices switch on according to the voice command*

In the above figure it shows when a voice command is given to switch on all the devices, all the devices will turn on and the LCD Display shows that the bulb is off.





Sure, turning 3 things off.

turn off all the devices

*Figure 18 - Multiple switches off according to the voice command*

In the above figure it shows when a voice command is given to switch on the bulb, all the devices will turn off and the LCD Display shows that the bulb is off.

## 7.0 CONCLUSION

In this project, its mainly focused about the home automation system, which was using the voice controlling systems. It can be used to switch on/off any home appliances as well as search the intended control devices such as Google home and other smart appliances. The purpose of this project is to make homes simple, better, smarter and more accessible.

The main area that is focused on here is about the ESP 32 chip and how its working according to voice commands to operate the devices (lamp, lights, Fan). Mainly the whole task was going through the ESP32 over Wi-Fi.

Further it can be developed by reducing the above circuit into a small PCB by designing the schematics so that it can be connected to any device without an issue. So, the idea is to place a main device that will show the status of the devices and to manage the devices. Therefore, the above circuit breaks into two parts as main device and sub devices. Main device will consist of LCD Display and the ESP32 chip, and the sub devices will consist of the relays, regulators and relevant components need for the device, which every sub device is linked to the main device. Instead of LCD Display an OLED display can be used for a better view.

## 8.0 REFERENCES

- ❖ Master The Basics Of Arduino - Full Arduino Programming Course (2019). *Master The Basics Of Arduino - Full Arduino Programming Course*. YouTube. Available at: https://youtu.be/1R3fqSFCAjM.

- ❖ www.youtube.com. (n.d.). *Home Automation using Alexa | Setting up with Nodemcu and Sinric pro | With Arduino code*. [online] Available at: https://youtu.be/P-Fa-005bd4 [Accessed 27 April 2022].

- ❖ www.youtube.com. (n.d.). *Integrate ESP32 with Sinric Pro and Alexa*. [online] Available at: https://youtu.be/zXdBcfJYG5E [Accessed 05 May. 2022].

- ❖ www.academia.edu. (n.d.). *Home Automation System Research Papers –* □ *Academia.edu*. [online] Available at: https://www.academia.edu/Documents/in/Home_Automation_System.

- ❖ ResearchGate. (n.d.). *(PDF) Home Automation Systems - A Study*. [online] Available at:
- ❖ https://www.researchgate.net/publication/275338025_Home_Automation_Systems__A_Study.

- ❖ “7805 Voltage Regulator IC: Pinout, Diagrams, Equivalent & Datasheet,” *Components101*. https://components101.com/ics/7805-voltage-regulator-ic-pinout-datasheet.

- ❖ E. Electronics *et al.*, “Programming ESP32 Board with Arduino IDE,” *Circuit Digest*, Oct. 15, 2018. https://circuitdigest.com/microcontroller-projects/programming-esp32-with-arduino-ide

- ❖ “ESP32 - Code Structure,” *ESP32 Tutorial*. https://esp32io.com/tutorials/esp32-code-structure (accessed Aug.20 , 2022).

- ❖ “Getting Started with the ESP32 Development Board | Random Nerd Tutorials,” Aug. 27, 2018. https://randomnerdtutorials.com/getting-started-with-esp32/