1. ABSTRACT

The home automation system IoT project is a smart home system that integrates various devices and appliances into a centralized control system. The system is based on the Internet of Things (IoT) technology, which enables communication and data exchange among different devices. The project includes the development of a mobile application that enables users to control their home appliances remotely, as well as a central hub that serves as the main controller for the entire system. The system can be programmed to automate various tasks, such as turning on and off lights, adjusting thermostats, and controlling security systems. The project aims to provide a more convenient and efficient way of managing home appliances and systems, while also promoting energy conservation and cost savings. The aim is to create a home automation system that performs all basic functions of a virtual assistant like telling the time, date, temperature and also controlling the electrical appliances that it is connected to. The entire system is aimed to be voice operated so that there is no need to type anything at all. Apart from the voice operated commands, the system will also take the help of certain sensors to provide automation to certain appliances.

2. INTRODUCTION

2.1 Introduction to Arduino IDE

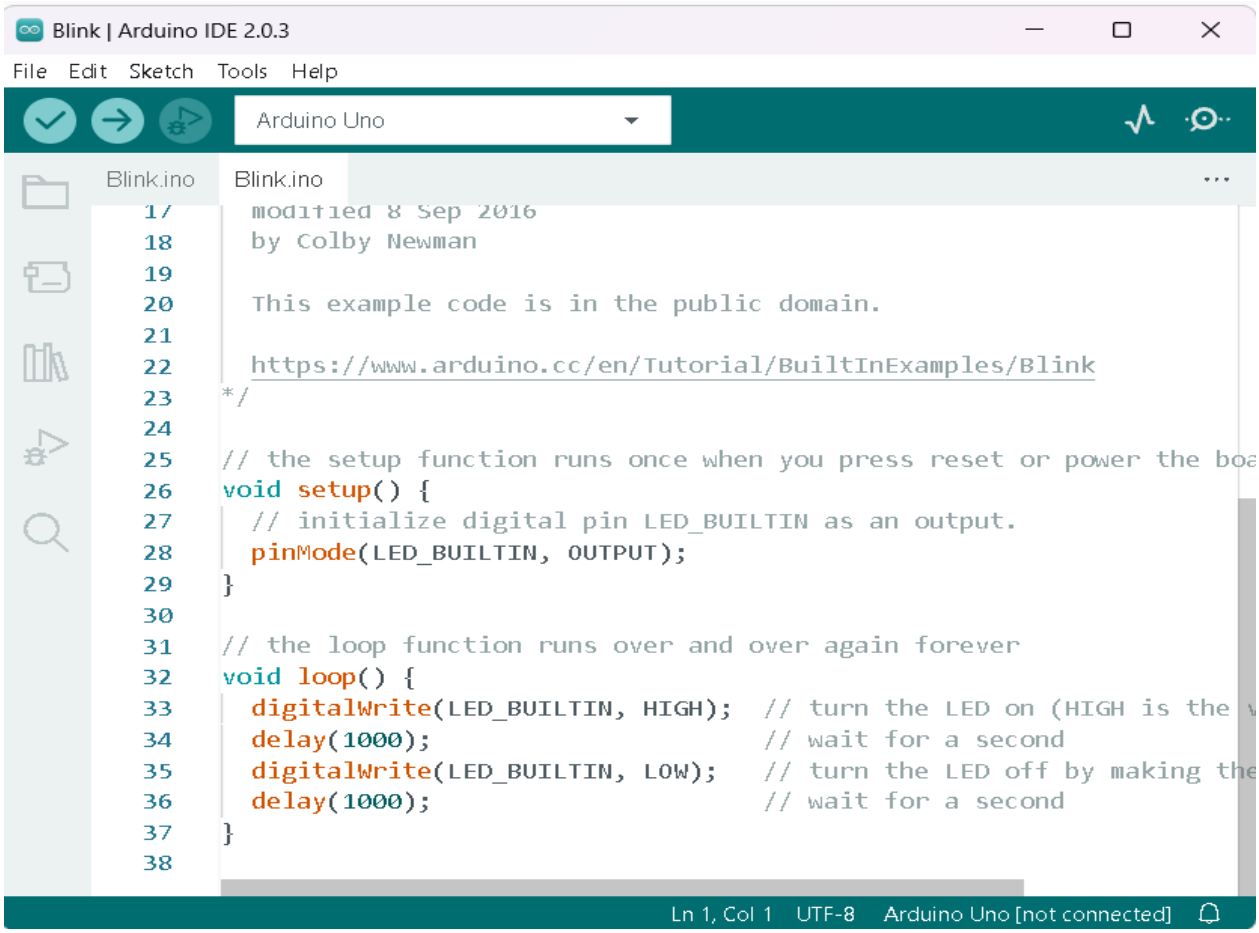

Figure 2.1: Arduino IDE with built-in Blink code

The Arduino Integrated Development Environment (IDE) is a software platform that provides a comprehensive environment for programming and developing projects using Arduino boards. Arduino is an open-source electronics platform that allows users to create interactive electronic projects by combining hardware and software components.

The Arduino IDE is designed to be user-friendly and accessible, even for individuals with little to no prior programming experience. It provides a simplified programming interface based on the C++ programming language, making it easier for beginners to get started with coding for Arduino.

2.2 Introduction to ESP-WROOM-32

Figure 2.2: ESP-WROOM-32 module with Wi-Fi inbuilt

The ESP-WROOM-32 is a highly versatile and powerful Wi-Fi and Bluetooth-enabled module developed by Espressif Systems. It is based on the ESP32 system-on-a-chip (SoC) and has gained significant popularity in the field of Internet of Things (IoT) development. This module offers a wide range of features and capabilities that make it an ideal choice for various IoT applications.

The ESP-WROOM-32 module integrates a dual-core Tensilica LX6 microcontroller, which operates at up to 240 MHz. It comes equipped with 4 MB of flash memory for program storage and data handling, as well as 520 KB of SRAM for efficient execution of code and data storage. Additionally, the module supports a wide range of peripherals, including SPI, I2C, UART, PWM, and ADC interfaces, enabling seamless integration with various sensors, actuators, and other devices.

2.3 Introduction to 4 Channel 5V Relay module

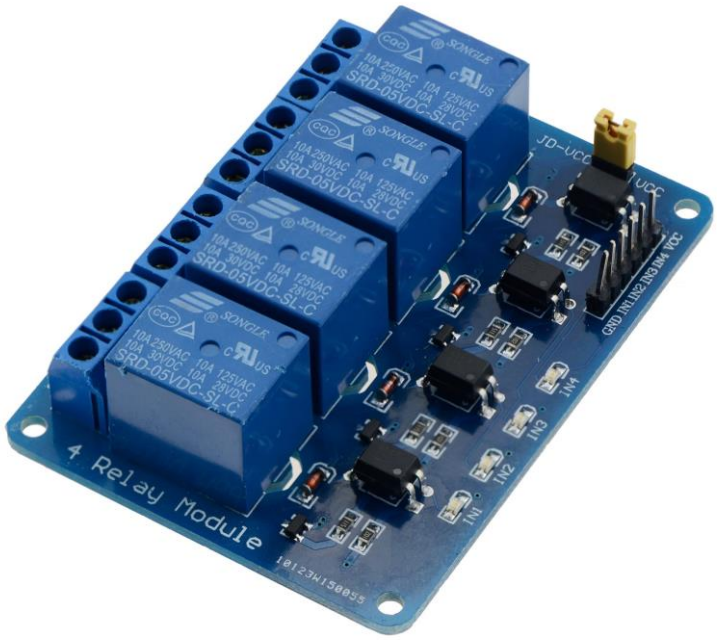

Figure 2.3: 4 Channel 5V Relay module

A 4 channel 5V relay module is an electronic device that is used to control high voltage or high current circuits using low voltage signals. It consists of a small printed circuit board with four independent relays mounted on it, each of which can be controlled independently by a separate signal. The module is designed to be triggered by a low voltage signal, typically in the range of 3-5V DC, which is usually provided by a microcontroller or other low voltage source. The relay contacts are capable of switching AC or DC loads and are typically rated for 10A/250VAC or 10A/30VDC. The 4 channel 5V relay module is commonly used in various projects such as home automation, robotics, industrial control, and other applications where low voltage control of high voltage or high current loads is required. It provides a safe and reliable way to switch on and off electrical circuits without the need for direct high voltage control signals.

3. REQUIREMENT SPECIFICATION

3.1 Hardware Requirements

- Smartphone with a hotspot capacity
- Laptop or PC to upload code
- ESP-WROOM-32 Wi-Fi module
- 4 Channel Relay module
- USB cable
- AC supply 220V
- Jumper Wires
- Bulb holder
- Bulb and wires
- Breadboard

3.2 Software Requirements

- Arduino IDE
- ESP-WROOM-32 board installed in IDE
- ArduinoJson Library
- SinricPro Library
- ArduinoWebSockets Library

4. DESIGN

4.1 System Architecture

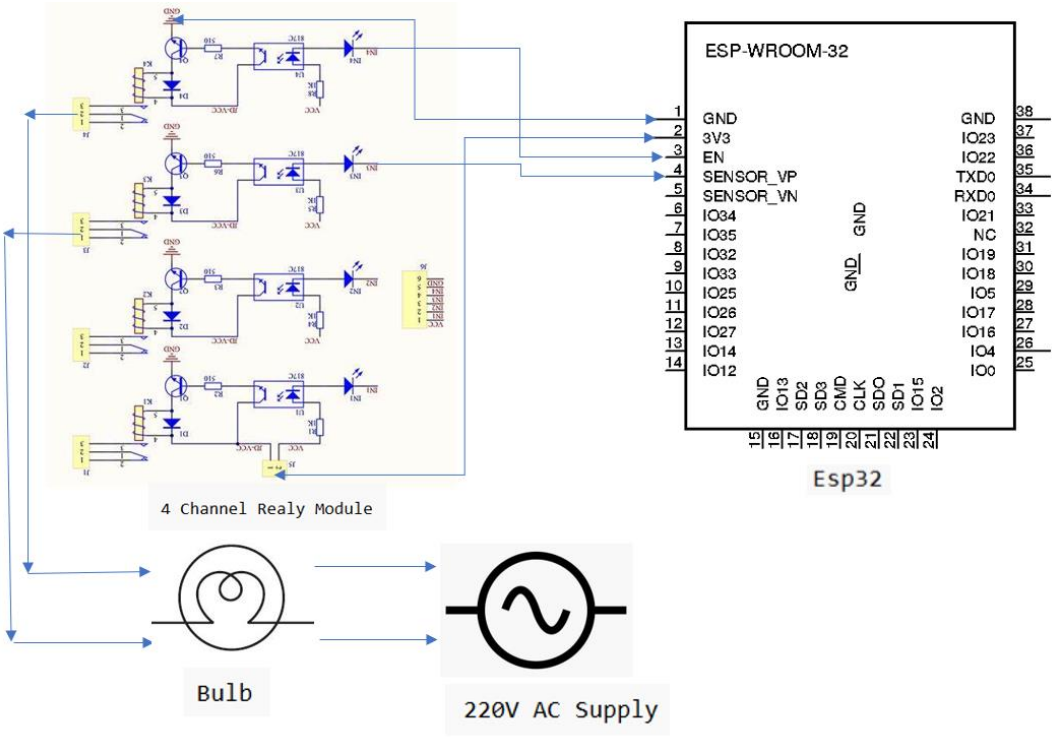

Figure 4.1: System Architecture for Home Automation system

The Home automation system consists of an ESP-WROOM-32 connected to a 4 Channel relay module. The ESP-WROOM-32, programmed using the Arduino IDE and equipped with Wi-Fi capabilities, acts as the brain of the system. The relay controls the bulb by turning it on and off. When the voice command is sent through Google assistant it reaches the sinricPro server Then esp32 will receive and process it then it is sent to rely which will turn off or on the bulb accordingly

4.2 Behavioral Model

After all the hardware setup, the first step is to upload the code. A laptop or PC can be used just to upload code through USB cable to ESP-WROOM-32 module. When the voice command is sent through Google assistant it reaches the sinricPro server Then esp32 will receive and process it then it is sent to rely which will turn off or on the bulb accordingly

.

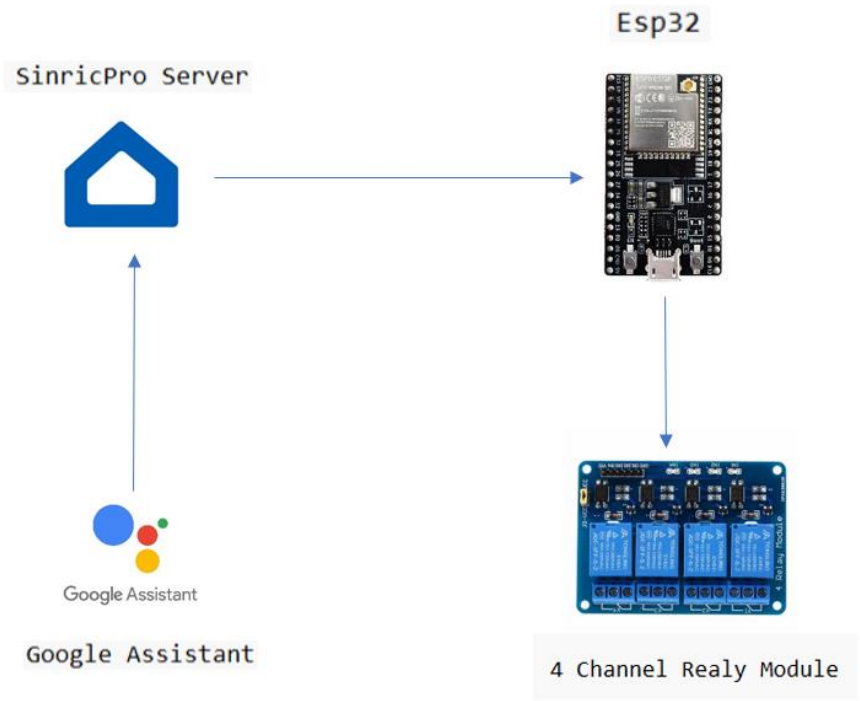

Figure 4.2: Behavioral Model

5. IMPLEMENTATION

5.1 Circuit

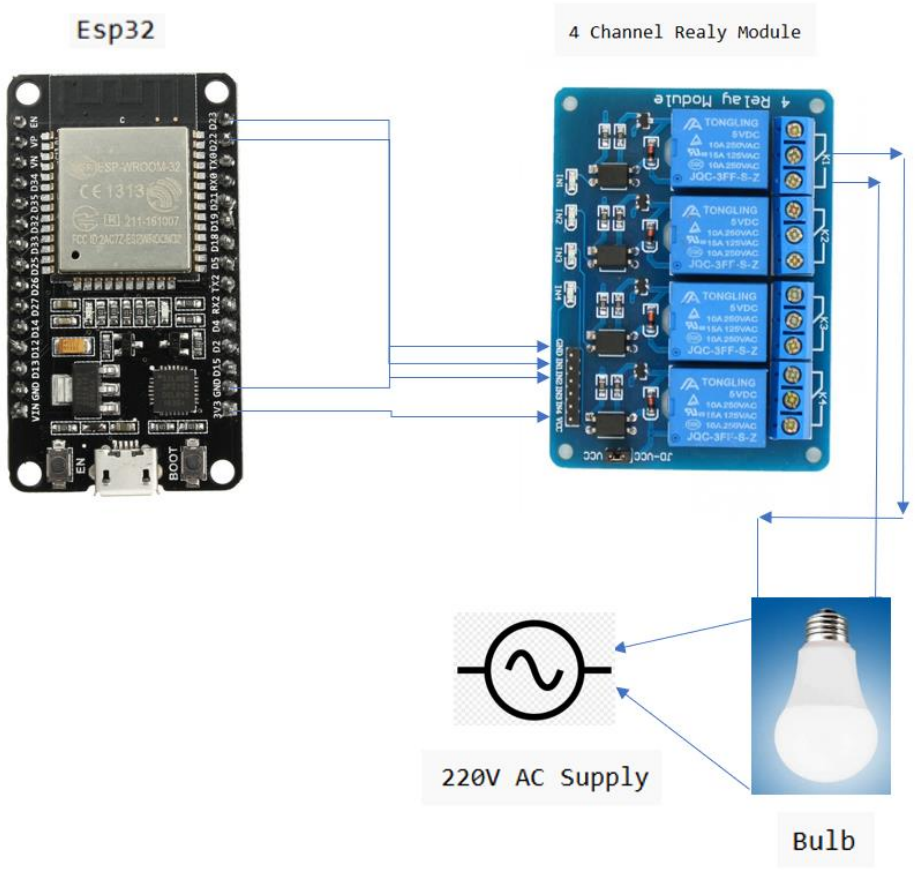

Figure 5.1: Complete circuit for Home automation system

(GND) – (GND)

(3V3) – (VCC)

(D23) – (IN1)

(D22) – (IN2)

5.2 Code

```
#include <Arduino.h>
#include <WiFi.h>
#include "SinricPro.h"
#include "SinricProSwitch.h"
#include <map>

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

#define WIFI_SSID         "rahullaptop"
#define WIFI_PASS         "988988988"
#define APP_KEY           "2a1dd46d-58c0-47f4-9966-8a59e283a1f9"
#define APP_SECRET        "2e9304df-c55a-4e9d-b156-b1e2e8a683b1-32712b51-e094-440f-a0fc-58071f237146"

#define device_ID_1   "645c7df4929949c1da63fbd9"
#define device_ID_2   "645c7da3743f912070160285"
#define RelayPin1 23
#define RelayPin2 22
#define SwitchPin1 13
#define SwitchPin2 12
#define wifiLed   2
#define BAUD_RATE   9600
#define DEBOUNCE_TIME 250

typedef struct {
  int relayPIN;
  int flipSwitchPIN;
} deviceConfig_t;


std::map<String, deviceConfig_t> devices = {

    {device_ID_1, {  RelayPin1, SwitchPin1 }},
    {device_ID_2, {  RelayPin2, SwitchPin2 }},

};

typedef struct {
  String deviceId;
  bool lastFlipSwitchState;
  unsigned long lastFlipSwitchChange;
} flipSwitchConfig_t;
```

```
std::map<int, flipSwitchConfig_t> flipSwitches;


void setupRelays() {
  for (auto &device : devices) {
    int relayPIN = device.second.relayPIN;
    pinMode(relayPIN, OUTPUT);
    digitalWrite(relayPIN, HIGH);
  }
}

void setupFlipSwitches() {
  for (auto &device : devices)  {
    flipSwitchConfig_t flipSwitchConfig;

    flipSwitchConfig.deviceId = device.first;
    flipSwitchConfig.lastFlipSwitchChange = 0;
    flipSwitchConfig.lastFlipSwitchState = false;
    int flipSwitchPIN = device.second.flipSwitchPIN;

    flipSwitches[flipSwitchPIN] = flipSwitchConfig;
    pinMode(flipSwitchPIN, INPUT_PULLUP);
}

bool onPowerState(String deviceId, bool &state)
{
  Serial.printf("%s: %s\r\n", deviceId.c_str(), state ? "on" : "off");
  int relayPIN = devices[deviceId].relayPIN;
  digitalWrite(relayPIN, !state);
  return true;
}

void handleFlipSwitches() {
  unsigned long actualMillis = millis();
  for (auto &flipSwitch : flipSwitches) {
    unsigned long lastFlipSwitchChange = flipSwitch.second.lastFlipSwitchChange;

    if (actualMillis - lastFlipSwitchChange > DEBOUNCE_TIME) {

      int flipSwitchPIN = flipSwitch.first;
      bool lastFlipSwitchState = flipSwitch.second.lastFlipSwitchState;
      bool flipSwitchState = digitalRead(flipSwitchPIN);
      if (flipSwitchState != lastFlipSwitchState) {
#ifdef TACTILE_BUTTON
        if (flipSwitchState) {
#endif
          flipSwitch.second.lastFlipSwitchChange = actualMillis;
          String deviceId = flipSwitch.second.deviceId;
          int relayPIN = devices[deviceId].relayPIN;
          bool newRelayState = !digitalRead(relayPIN);
          digitalWrite(relayPIN, newRelayState);
```

```
        SinricProSwitch &mySwitch = SinricPro[deviceId];
        mySwitch.sendPowerStateEvent(!newRelayState);
#ifdef TACTILE_BUTTON
       }
#endif
       flipSwitch.second.lastFlipSwitchState = flipSwitchState;
     }
   }
  }
}

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

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

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

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

void setup()
{
  Serial.begin(BAUD_RATE);
  pinMode(wifiLed, OUTPUT);
  digitalWrite(wifiLed, LOW);
  setupRelays();
  setupFlipSwitches();
  setupWiFi();
  setupSinricPro();
}
void loop()
{
  SinricPro.handle();
  handleFlipSwitches();
```

}

5.3 Steps

1. Firstly, we should create sinricPro account and add the devices to it

2. next we have to do the circuit as shown in the figure

3. then we should open Arduino and install esp32 , sinricpro library's.

4.once installed then we have to upload the the code.

5.once the code is uploaded without any error we need to connect to that network through a smart phone.

6. Now open your Google home app on your smartphone.connect the sinricPro account with Google home app . Give the voice command same as created .

7.When the voice command is sent through Google assistant it reaches the sinricPro server.

8.Then esp32 will receive and process it then it is sent to rely which will turn off or on the bulb accordingly.

6.RESULTS

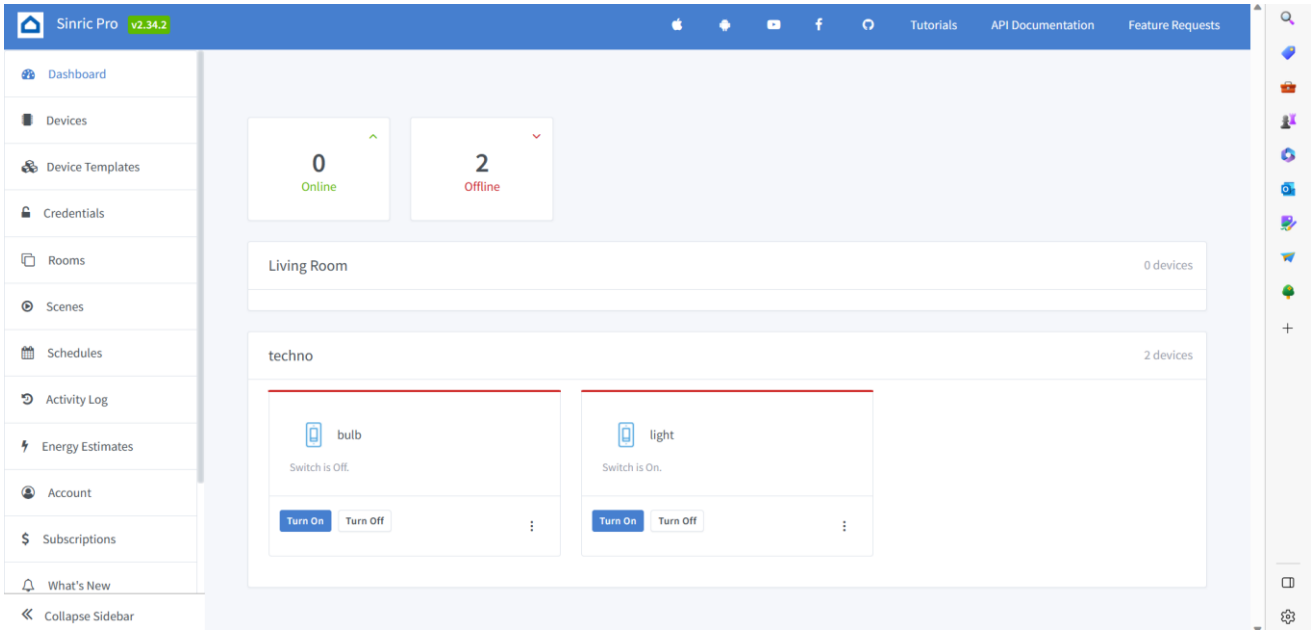

Figure 6.1: SinricPro Dashboard after creating the room ad adding devices.

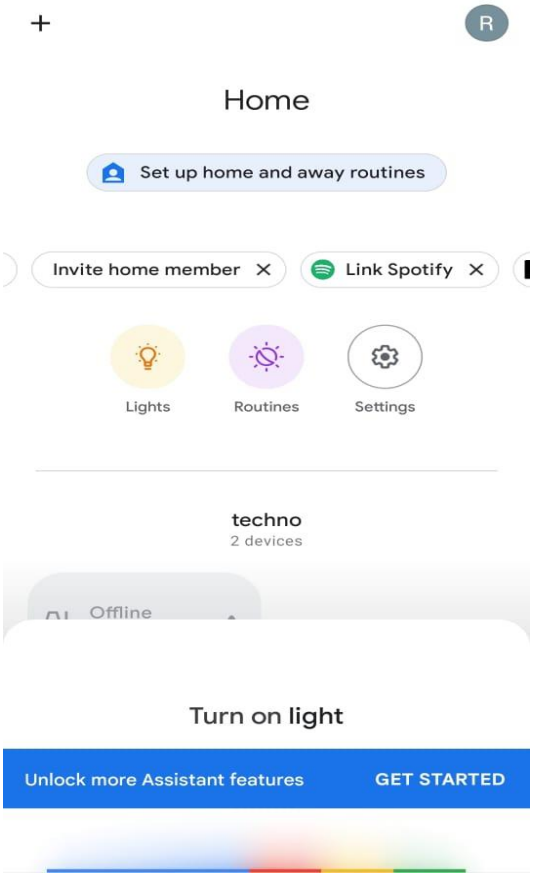

Figure 6.3: Google Home dashboard activating voice command

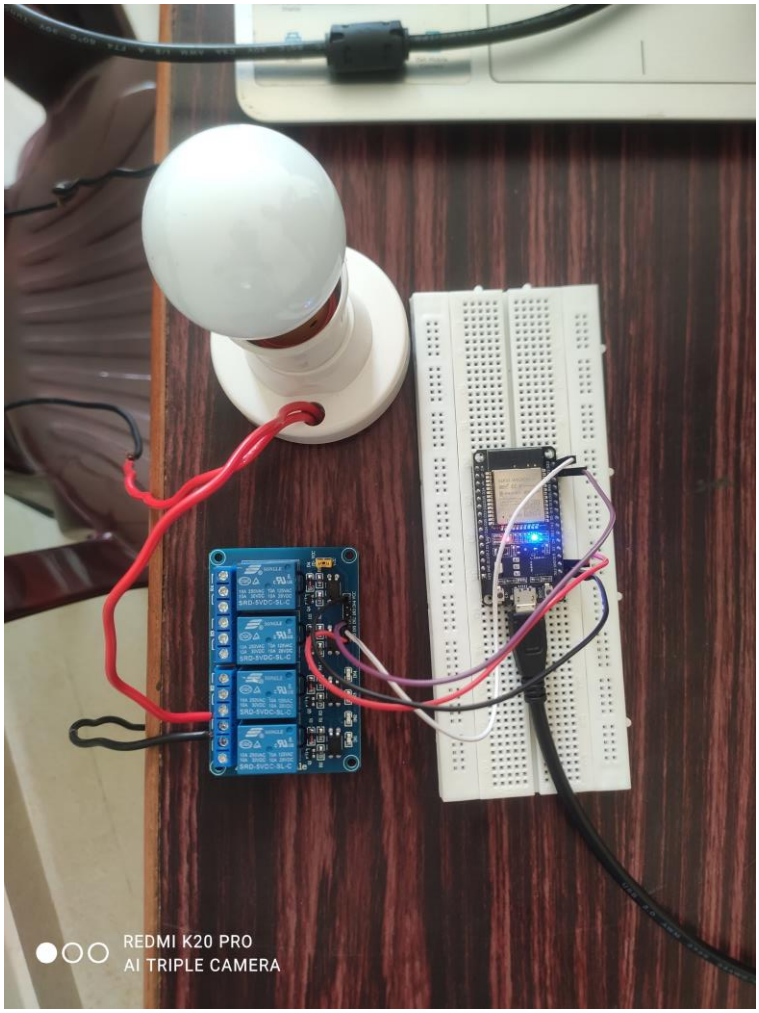

Figure 6.4: Home automation system setup before receiving voice command

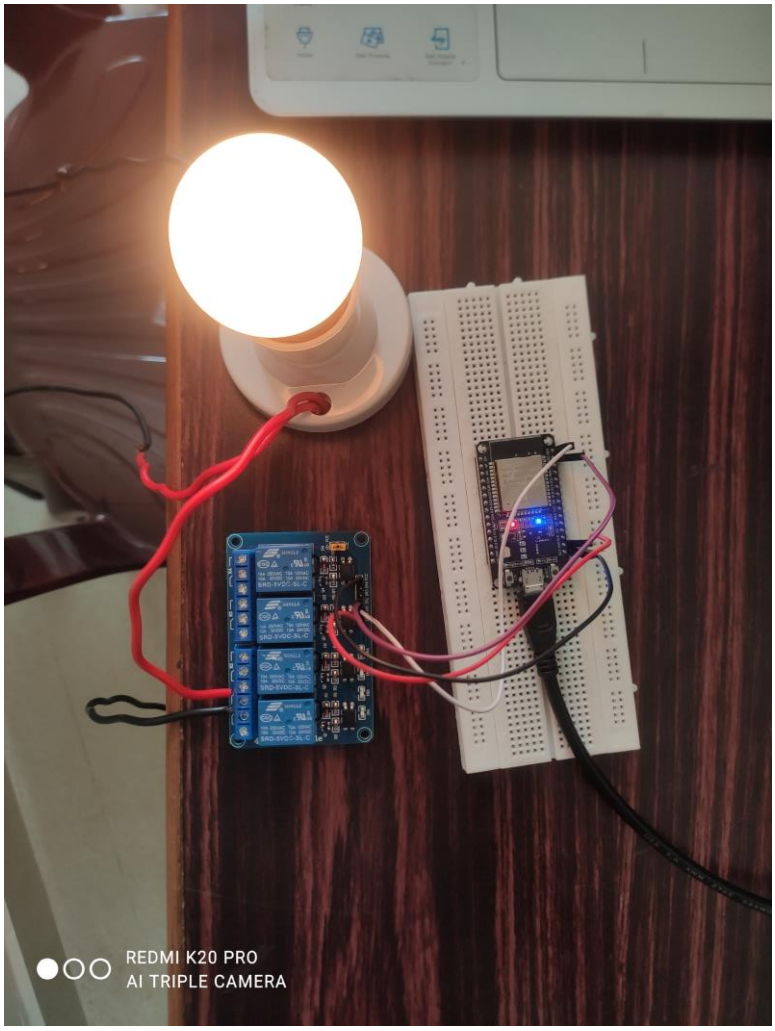

Figure 6.5: Home automation system setup after receiving voice command

7.CONCLUSION & FUTURE WORK

In conclusion, a home automation system IoT project can provide a convenient and efficient way of managing various home appliances and systems, while also promoting energy conservation and cost savings. By integrating different devices and appliances into a centralized control system, users can automate various tasks and control their home environment remotely using a mobile application or a web interface. The project can also improve home security and provide scalability and flexibility for future expansion. Overall, a home automation system IoT project can offer a more comfortable, convenient, and sustainable lifestyle for users.

Future Work :

The future of home automation systems using IoT (Internet of Things) technology is very promising, with many potential advancements and new applications in the pipeline.

Integration with voice assistants: One of the key trends in home automation is the integration of IoT devices with popular voice assistants such as Amazon Alexa and Google Assistant. This allows users to control their smart home devices using voice commands, making the home automation experience even more convenient and intuitive.

Increased security features: As IoT devices become more prevalent in our homes, security concerns have become more important. Future home automation systems are likely to incorporate stronger security features, such as advanced encryption and authentication protocols, to protect against hacking and other security threats.

8. REFERENCES

The other sources we referred while doing this project are as follows:

[1] https://youtu.be/DshR6Y9aTSs

[2] https://youtu.be/buMYCp9HxAM

[3] https://github.com/ahmadlogs/esp32/tree/main/blynk2-relay

[4] https://www.projectpro.io/article/top-20-iot-project-ideas-for-beginners-in-2021/428