ESP-32 BASED GOOGLE HOME ASSISTANT

ABSTRACT

Home Assistant is a device which integrates various components that are used to automate things around us. IoT is Internet of Things which denotes the intercommunication among devices by using an embedded system. The backbone of Home Automation is the Home Assistant which means controlling all the appliances without going close to the switches to control them. Automation has become cost-effective using embedded systems. An embedded system provides user comfort, security and flexibility along with maintaining the living standard. This Automation comprises of controllers, here a cheap micro-controller. The automatic control system created in this project uses ESP32. ESP32 is a microcontroller equipped with 2.4 GHz Wi-Fi and Bluetooth technology; the ESP32 development kit can be easily found on the market at a low price. Using this board and google assistant we will make a voice-controlled system to automate the appliances.

REQUIREMENTS

Hardware Requirements

1. ESP-32 module
2. Relays
3. Wires
4. Laptop
5. Smartphone

Software Requirements

1. Arduino-IDE
2. Google Assistant
3. Blynk
4. IFTTT

ESP-32

ESP32 is the name of the chip that was developed by Espressif Systems. This provides Wi-Fi and Bluetooth connectivity to embedded devices. The ESP32 is most commonly engineered for mobile devices, wearable tech, and IoT applications – such as Home Assistant. The ESP32 has gained a reputation as the ultimate chip or module for hobbyists and IoT developers. While this reputation is not unmerited, the low-cost device can also be used in a number of different production systems, and its capabilities and resources have grown impressively over the past few years.

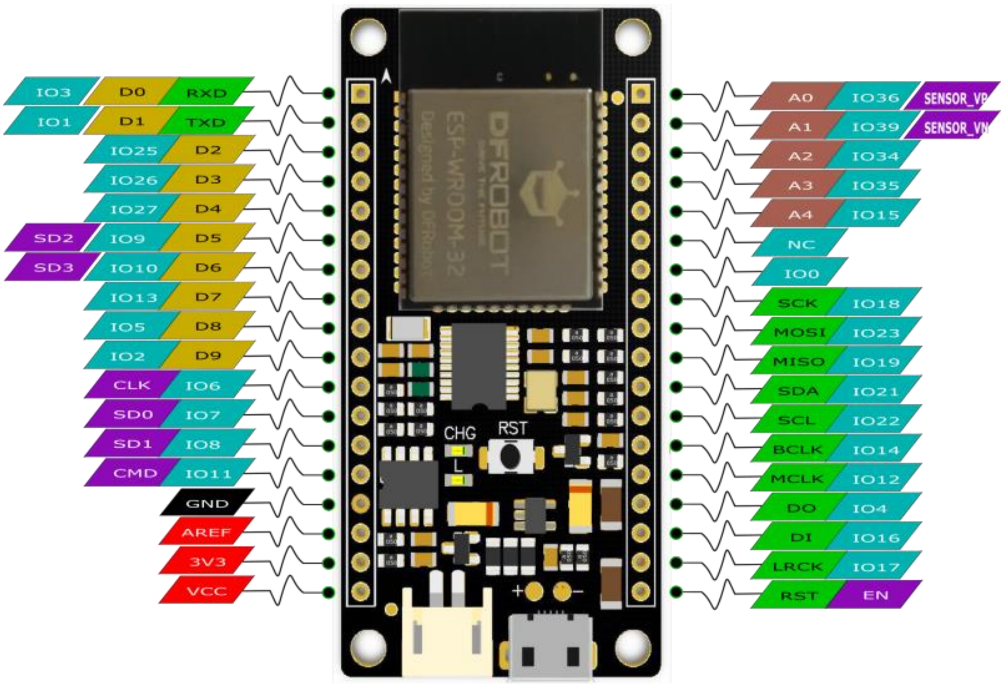


Fig: ESP-32 Module

ESP32 module is the core of this project. With the help of several onboard GPIO pins, multiple devices can be controlled wirelessly. It can be programmed easily using Arduino IDE with the help of several built-in libraries.

GOOGLE ASSISTANT

Google Assistant is an artificial intelligence powered virtual assistant developed by Google that is primarily available on mobile and smart home devices. It is Google's virtual voice assistant. Nowadays it comes pre-installed on pretty much every android device or it can be installed from Play Store. It is very easy to set-up and use. It offers voice commands, voice searching, and voice-activated device control, that lets control a number of tasks after saying the "OK Google" or "Hey Google" wake words. It is trained to give realistic conversational interactions. Google Assistant can also control devices and most importantly smart home.

BLYNK

Blynk is the most popular Internet of Things platform for connecting any hardware to the cloud, designing apps to control them, and managing our deployed products at scale. With Blynk Library we can connect over 400 hardware models (including ESP32, Node-MCU, all Arduinos, Raspberry Pi, etc.) to the Blynk Cloud. With Blynk apps for iOS and Android apps we can easily drag-n-drop graphic interfaces for any DIY or commercial project. Hardware can connect to Blynk Cloud (open-source server) over the Internet using hardware connectivity available on your board (like ESP32), or with the use of various shields (Ethernet, Wi-Fi, GSM, LTE, etc.). Blynk Cloud is available for every user of Blynk for free. Direct connection over Bluetooth is also possible.

IFTTT

If This Then That is a private commercial company that runs services that allow a user to program a response to events in the world. IFTTT has partnerships with different service providers that supply event notifications to IFTTT and execute commands that implement the responses.

HOW IT WORKS?

IOT CLOUD Platform BLYNK is used for implementing this. BLYNK is the most popular IoT platform to connect your devices to the cloud. BLYNK is a digital dashboard where we can build a graphic interface for our project by simply dragging and dropping widgets. After signing-up on BLYNK app using Gmail id, we need to create a new project selecting ESP32 device. An authentication token will be provided which will give address to our project. After programming ESP32 module using the authentication token we will be able to manually control ESP32 devices using BLYNK app.

Now here comes the interesting part, instead of controlling manually we can use voice commands to operate devices. For this we use the GOOGLE ASSISTANT on mobile phone. Google assistant cannot directly communicate with BLYNK. For making a chain link between Google assistant and Blynk we use IFTTT. If This Then That, also known as IFTTT, is a free web-based service to create chains of simple conditional statements, called applets. An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, etc. In this project we make use of WEBHOOKS on IFTTT to handle a web service, which is BLYNK.

Google assistant will send user voice command to IFTTT, there it will use WEBHOOKS to contact BLYNK CLOUD SERVER. Now cloud server will send command to ESP32 via home Wi-Fi router. ESP32 will then turn the appliance on or off according to the received command.

PROBLEM STATEMENT

IoT is Internet of Things which denotes the intercommunication among devices by using an embedded system. The backbone of Smart Homes is Home Automation which means controlling all the appliances without going close to the switches to control them. Home Automation has become cost-effective using embedded systems. An embedded system provides user comfort, security and flexibility along with maintaining the living standard.

1. Blynk App:

    a. Making an account on Blynk App to get token for the project

    b. Creating new project and selecting the microcontroller device

    c. Adding buttons to act as switches for electrical appliances

2. Google Voice Assistant

    a. Acts as a speech to text generator to send commands to the webserver.

    b. Includes a database of various voice commands

    c. Enhance the security by verifying user Voice Match

3. IFTTT (If This Then That)

    a. Runs services that allow a user to program a response to events in the world.

    b. Webhooks provides event notifications and execute commands that implement the responses.

4. Wi-Fi Router

    a. Provides wireless internet connectivity to the microcontroller.

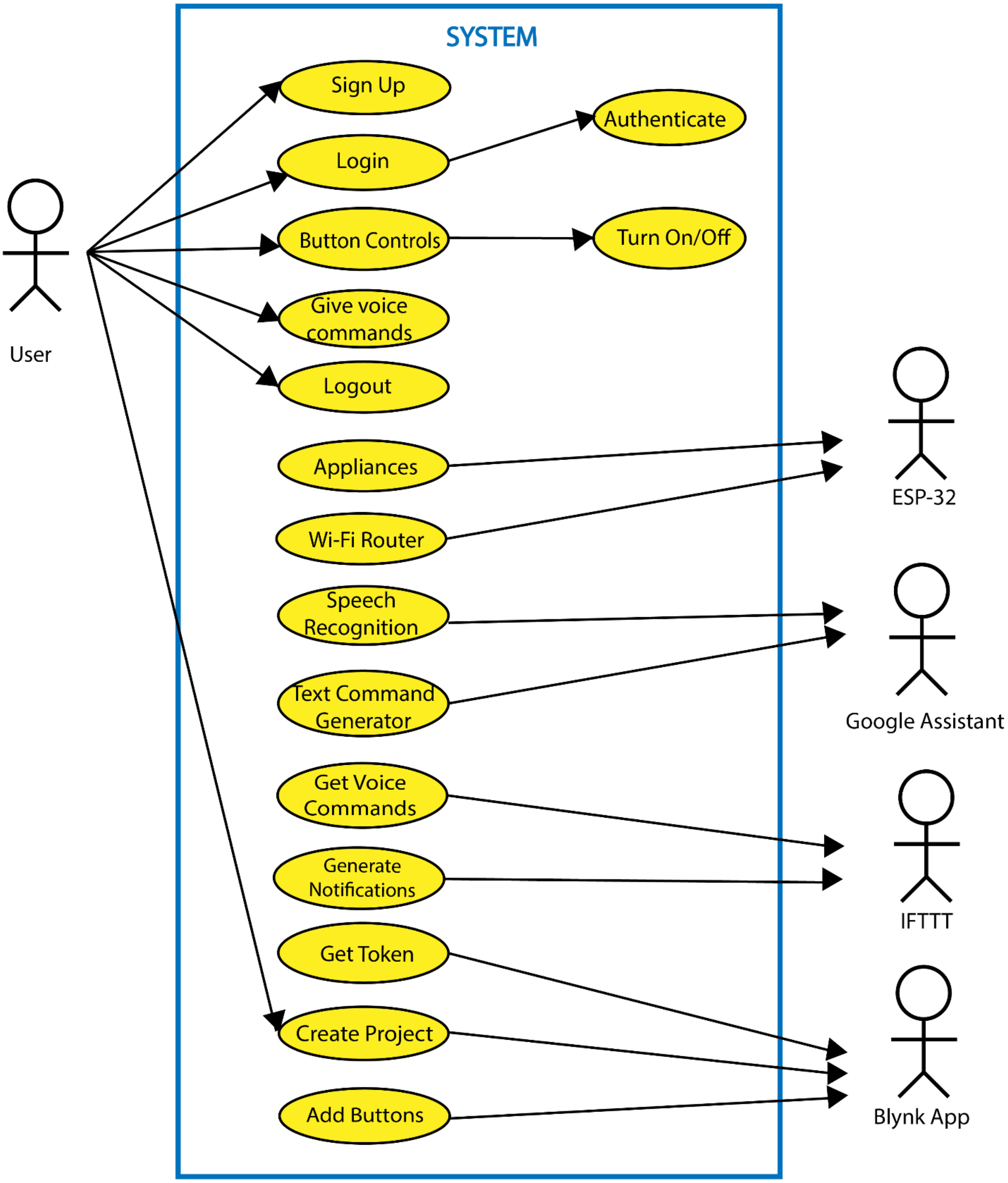
    b. Secure encryption with WPA-2-PSK security.

5. ESP-32

    a. A wireless microcontroller with inbuilt Wi-Fi.

    b. Using Arduino IDE to program the code in it.

USE CASE DIAGRAM



USE CASE DESCRIPTION

1. **Sign Up**

Introduction

- This use case describes about creating a Blynk account using google account.

Actors

- User

Pre-Conditions

- This functionality can be used by the app user.

Post Conditions

- If the use case is successful, the account is created and token is provided.

Basic flow

- This use case starts when the user wishes to create Blynk account.
- Registration must be done using google account.

Alternate flow

- While setting up a new profile, if we enter a username that already exists, the system will display an error message

Special requirement

- None

Use case relationship

- None

2. **Appliances Control**

Introduction

- This use case controls the light and other appliances.

Actors

- User
- ESP-32

Pre-Conditions

- The appliances must be connected to ESP module and power source.

Post Conditions

- When the ESP-32 detects a command, it turns on/off the lights.

Basic flow

- The appliances will initially remain off, they will work only after microcontroller commands.

Alternate flow

- Using manual switch, the user can turn on/off the appliances.

Special requirements

- None

Use case relationship

- The appliances can be controlled by both microcontroller and manual switch.

3. **Speech Recognition**

Introduction - This use case describes how the google assistant receives the commands and generates further response.

Actors

- Google Assistant
- User

Pre-Conditions

- The assistant must have the permission to access the microphone.

Post Conditions

- The assistant starts listening to voice commands given to it.

Basic flow

- The assistant recognizes the voice and converts speech to text.
- The text command is then sent to IFTTT for further operation.

Alternate flow

- If the recognized text is not a command, the assistant will not send command to IFTTT.

Special requirements

- Microphone Access, Internet Connectivity

Use case relationship

- None

4. **Generate Notifications**

Introduction

- This use case describes the response of the command in the form of notification from the IFTTT app.

Actors

- IFTTT

Pre-Conditions

- IFTTT must have permission to access notifications.

Post Conditions

- When the state of appliance is changed successfully using voice command then IFTTT sends a notification.

Basic flow

- This use case starts when a command completes its execution.

Alternate flow

- None

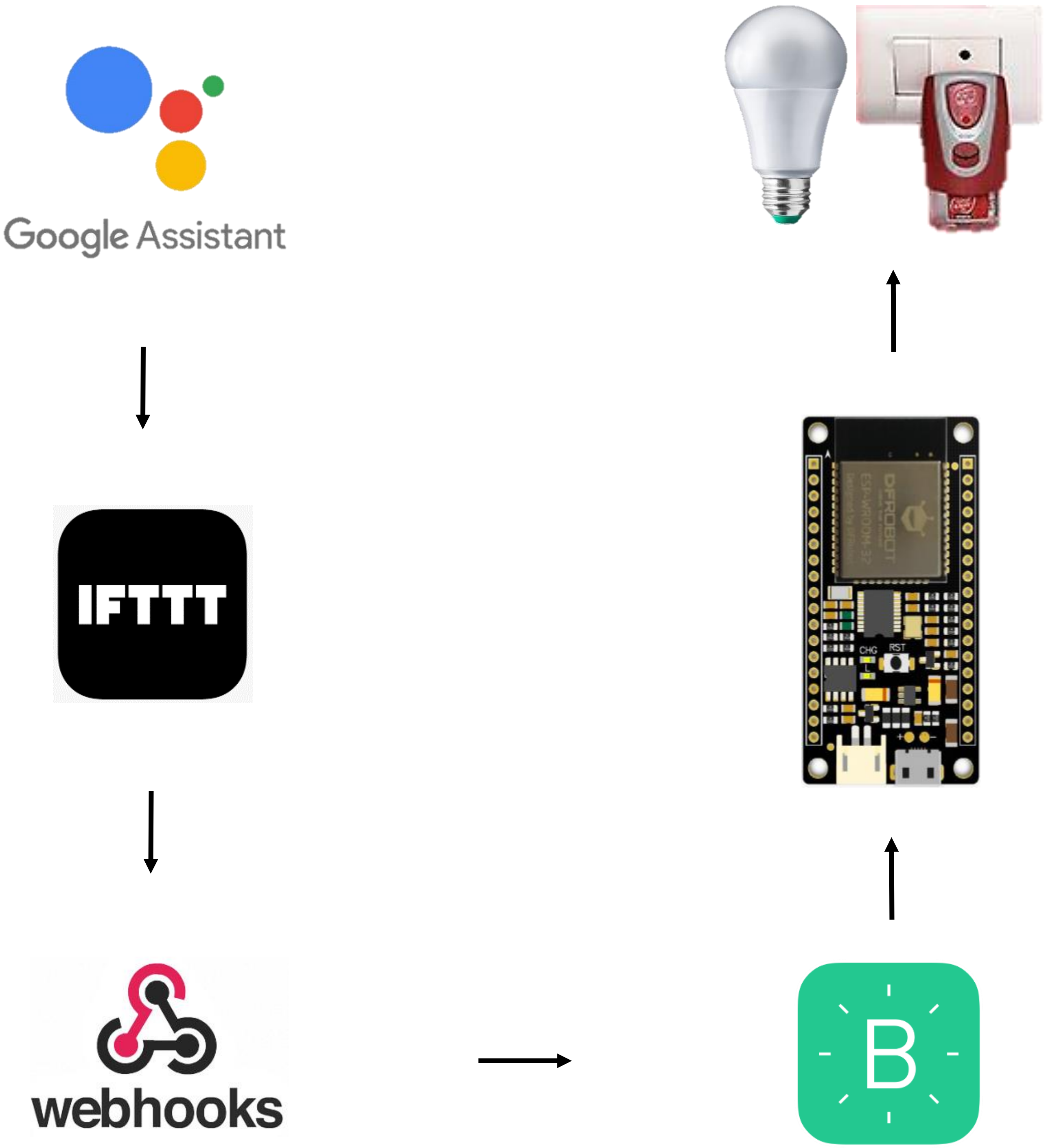
Special requirements

- Notification Access

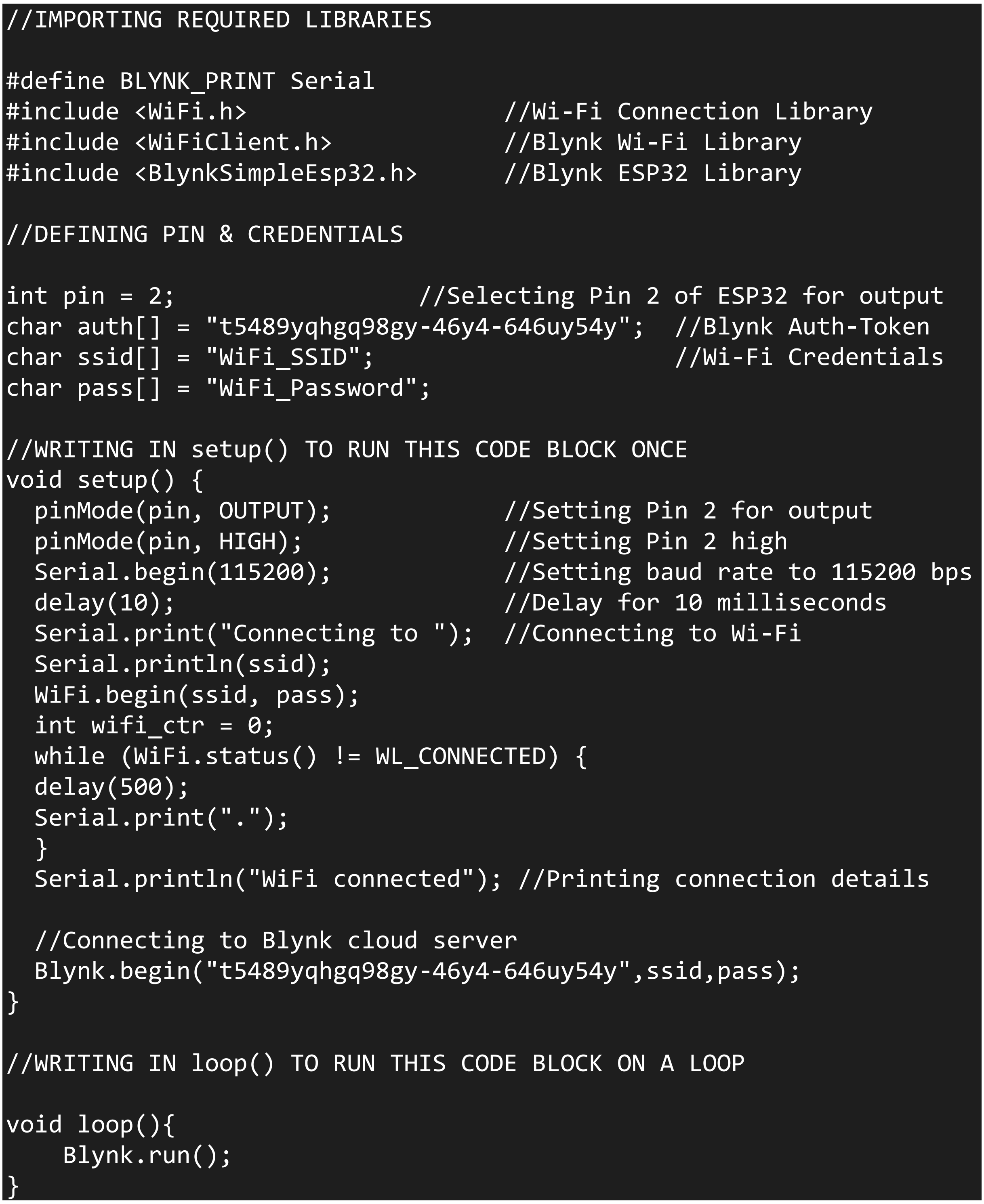
Use case relationship

- None

PROCESS DIAGRAM

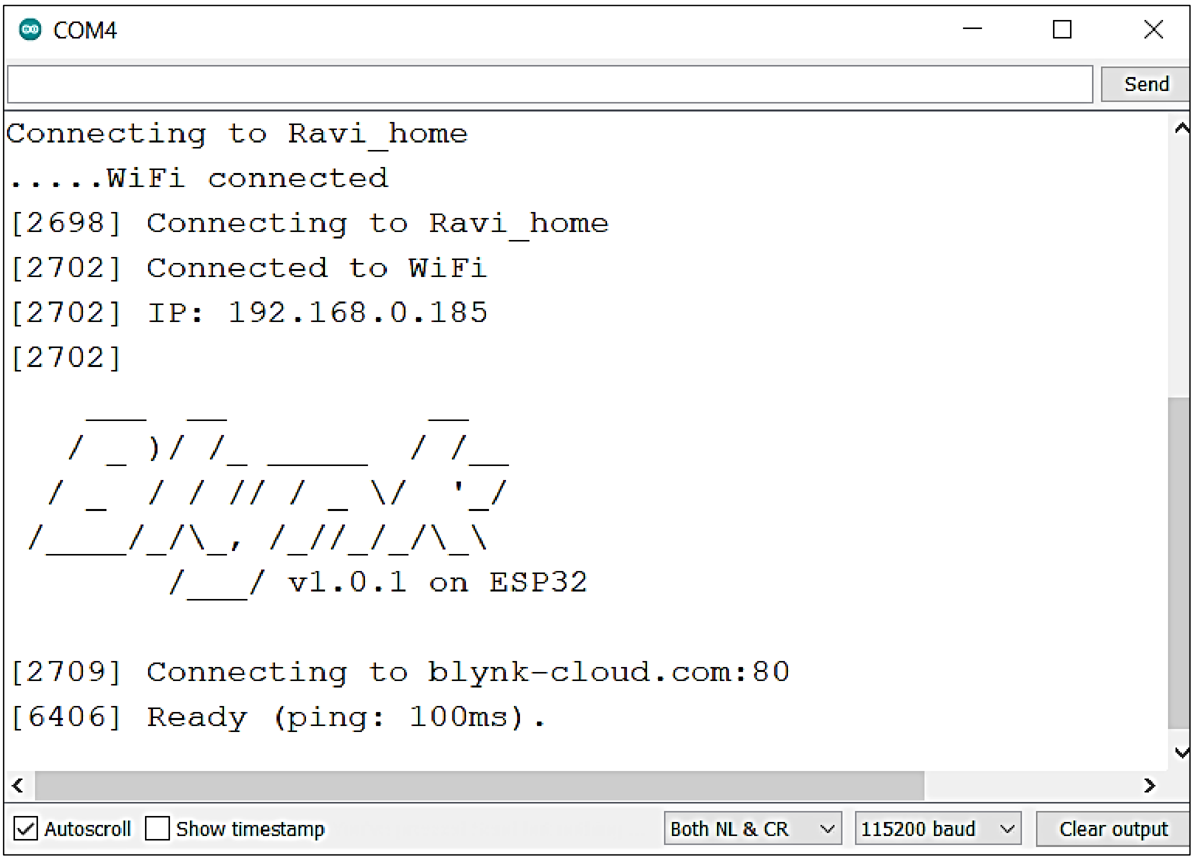


SOURCE CODE



```
//IMPORTING REQUIRED LIBRARIES

#define BLYNK_PRINT Serial
#include <WiFi.h>                  //Wi-Fi Connection Library
#include <WiFiClient.h>            //Blynk Wi-Fi Library
#include <BlynkSimpleEsp32.h>      //Blynk ESP32 Library

//DEFINING PIN & CREDENTIALS

int pin = 2;                 //Selecting Pin 2 of ESP32 for output
char auth[] = "t5489yqhgq98gy-46y4-646uy54y";  //Blynk Auth-Token
char ssid[] = "WiFi_SSID";                     //Wi-Fi Credentials
char pass[] = "WiFi_Password";

//WRITING IN setup() TO RUN THIS CODE BLOCK ONCE
void setup() {
  pinMode(pin, OUTPUT);            //Setting Pin 2 for output
  pinMode(pin, HIGH);              //Setting Pin 2 high
  Serial.begin(115200);            //Setting baud rate to 115200 bps
  delay(10);                       //Delay for 10 milliseconds
  Serial.print("Connecting to ");  //Connecting to Wi-Fi
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  int wifi_ctr = 0;
  while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
  }
  Serial.println("WiFi connected"); //Printing connection details

  //Connecting to Blynk cloud server
  Blynk.begin("t5489yqhgq98gy-46y4-646uy54y",ssid,pass);
}

//WRITING IN loop() TO RUN THIS CODE BLOCK ON A LOOP

void loop(){
    Blynk.run();
}
```

OUTPUT



```
Connecting to Ravi_home
.....WiFi connected
[2698] Connecting to Ravi_home
[2702] Connected to WiFi
[2702] IP: 192.168.0.185
[2702] 
    ___  __          __
   / _ )/ /_ _____  / /__
  / _  / / // / _ \/  '_/
 /____/_/\_, /_//_/_/\_\
        /___/ v1.0.1 on ESP32

[2709] Connecting to blynk-cloud.com:80
[6406] Ready (ping: 100ms).
```

☑ Autoscroll □ Show timestamp Both NL & CR 115200 baud

CONCLUSION

In this project, voice commands are given to the Google assistant. The voice commands for Google assistant have been added through IFTTT website and the Adafruit account is also linked to it. In this home automation, user have given commands to the Google assistant. Home appliances like Bulb, Fan and Motor etc., are controlled according to the given commands. The commands given through the Google assistant are decoded and then sent to the microcontroller and it control the relays. The device connected to the respective relay turned On or OFF as per the users request to the Google Assistant.

There has been tremendous growth in the home automation sector, and many reputed companies utilizing their opportunity to work with IFTTT to deliver an elegant way to connect families to their homes. Consumers are looking to secure their home environment in today's unpredictable world, and the new home automation service gives them the peace of mind that they need to protect their family's well-being. This project is about wireless home automation using Android mobile helps us to implement such a fantastic system in our home at a very reasonable price using cost-effective devices. Thus, it overcomes many problems like costs, inflexibility, security etc. In addition, will provide greater advantages like it decrease our energy costs, it improves home security. In addition, it is very convenient to use and will improve the comfort of our home. The project has proposed the idea of smart homes that can support a lot of home automation systems.