Title: *Multi-Language Voice Control IoT Home Automation Using Google Assistant and Raspberry Pi*

I. Summary

The purpose of this project is to build a multi-language voice control IoT home automation using Google Assistant and Raspberry Pi. This project will enable the user to control a home despite the language being used. This will be enables by using IoT devices, various components and the Raspberry Pi Smart devices and home automation is growing with the advancement of technology, specially IoT devices. It is important for upcoming engineering students to have basic knowledge about IoT devices, so the student can thrive in a job dealing with the design of smart cities, smart homes, smart devices and more.

II. Objectives

The purpose of this project is to build a multi-language voice control IOT home automation using Google Assistant and Raspberry Pi. The project will teach the student about image processing and to program the Raspberry Pi device.

III. Industry-Based Applications

This project is related to industry by many companies are implanting the internet of things into many products. These products are seen in everyday lives products, such as, televisions, refrigerators, smart speakers and more. Therefore, it is important for student to become familiar with IoT devices and the construction of them. The IoT devices will

be the future of many engineering design projects. This project will develop the students programming skills and circuit design for real world application of a product.

In another paragraph (or same previous paragraph), you should tell where the importance of your project can be used in industry and use reference number ([xx]). References should be all at the end of the paper.

IV. Project Methodology

a. Parts

i. Raspberry Pi Zero W

ii. Raspberry Pi Camera

iii. Raspberry Pi Case

iv. 5V Relay

v. Servo motor and wires

b. Procedure

i. Setting Up Raspberry Pi

1. Set up an Apache server in Raspberry Pi.
2. Open the terminal window and run the command “sudo apt-get install apache2 -y”
3. Check it by typing the I.P address of your Raspberry Pi in any web browser
4. If the server is working fine, the Apache page in the web browser.
5. Use the Apache server to process PHP Files, for this the latest version of PHP module for Apache. To get this module run the command, “sudo apt-get install php libapache2-mod-php –y”

6. Create a PHP file to control the GPIO of Raspberry Pi. To do so, open terminal in Raspberry Pi and go to its html directory by this command as you can see in Figure 1.
7. Create a PHP file by using the command “sudo nano lightsoff.php” and write the code:
    a. <?php
    b. System (“gpio -g mode 27 out”);
    c. System (“gpio -g write 27 0”);
    d. ?>
8. Press Ctrl+X to save and exit the editor.
9. Create a lightson.php file for turning the lights on and paste the code into the files:
    a. <?php
    b. System (“gpio -g mode 27 out”);
    c. System (“gpio -g write 27 1”);
    d. ?>
10. Repeat the same process for controlling other GPIO of Raspberry Pi.

ii. Setting Up Google Assistant

1. Open the language settings of google assistant and select the language of your choice.
2. Open the google assistant settings and go to “Routines” option.
3. Click on the plus floating button in the menu of Routines to set the voice command.

iii. Connection

1. Use the illustration below to connect relay board and Raspberry Pi

Raspberry Pi	Relay Board
GPIO 13	Relay IN 1
GPIO 15	Relay IN 2
5V	Relay 5V
GND	GND

iv. App Making

1. Open Android studio to create a new project with button navigation bar and do coding as illustrated in Figures 10 through 14.
2. Set the Permission in app to access the internet to use Wi-Fi to control GPIO of Raspberry Pi.
3. Go to Mainactivity.java to create a WebView and String variables like in Figure 11.
4. Create the navigation Bar.

v. Testing

1. Power the Raspberry PI with 5V DC.
2. Connect Raspberry PI and phone to a Wi-Fi network or hotspot.
3. Next, say “hey google” followed by the voice command that is set in the preferred language

4. It will turn on the lights
5. The app can be controlled using the app as well.
6. Open the app and press the icons in navigation to turn the lights on/off

V. Photos and References

a. https://electronicsforu.com/electronics-projects/multi-language-voice-control-iot-home-automation-system-using-google-assistant-and-raspberry-pi

b. https://youtu.be/gbJ4yC-ICvQ

c. **Code:** https://electronicsforu.com/wp-contents/uploads/2019/05/CODE.zip

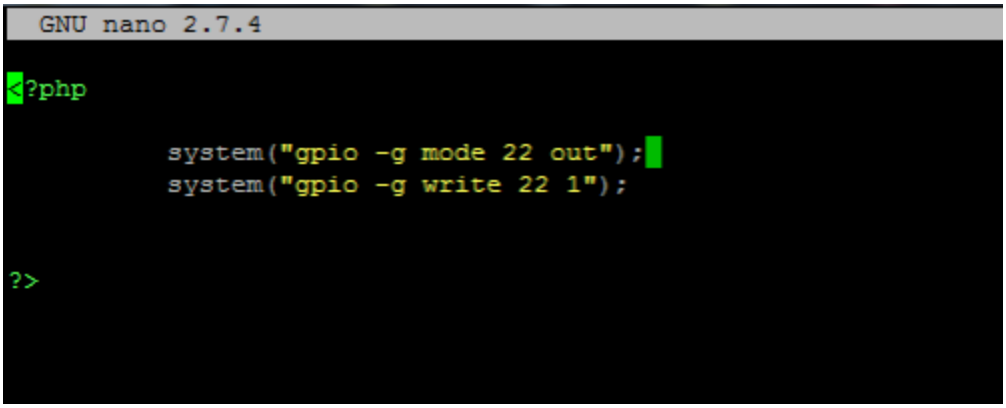

Figure 2- This is the code in PHP

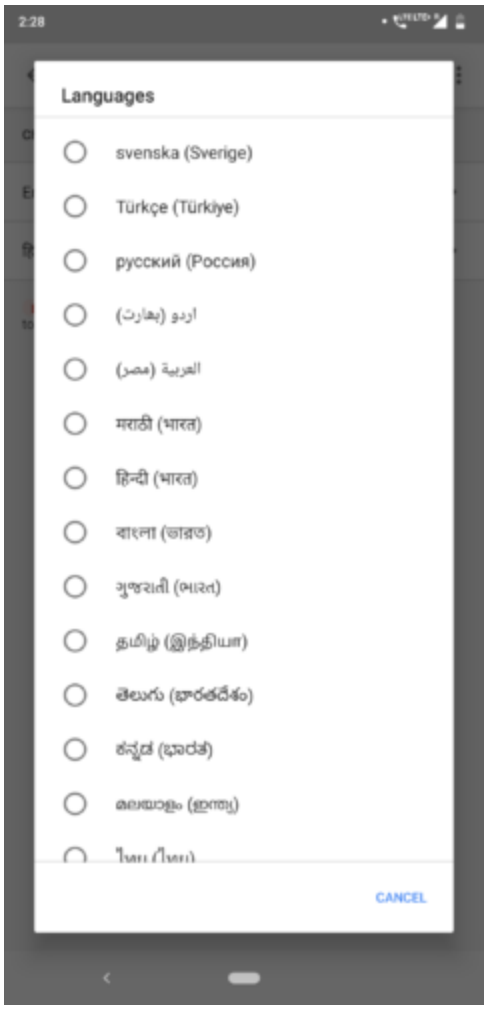

Figure 3- Setting up the language in google assistant

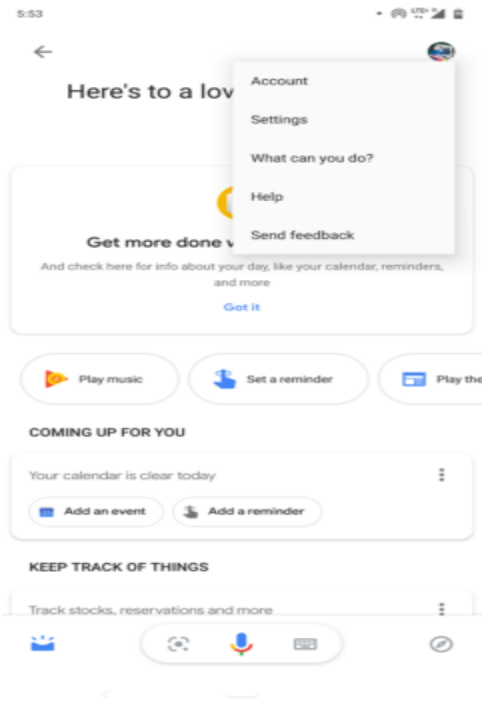

Figure 4- Finding the google assistant settings

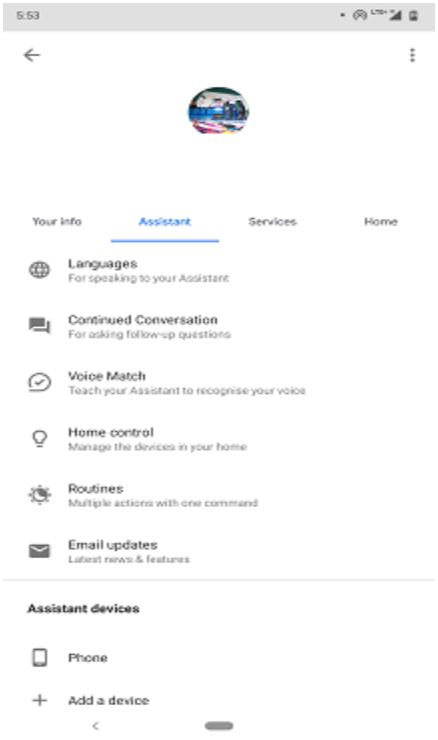

Figure 5- Google assistant setting menu

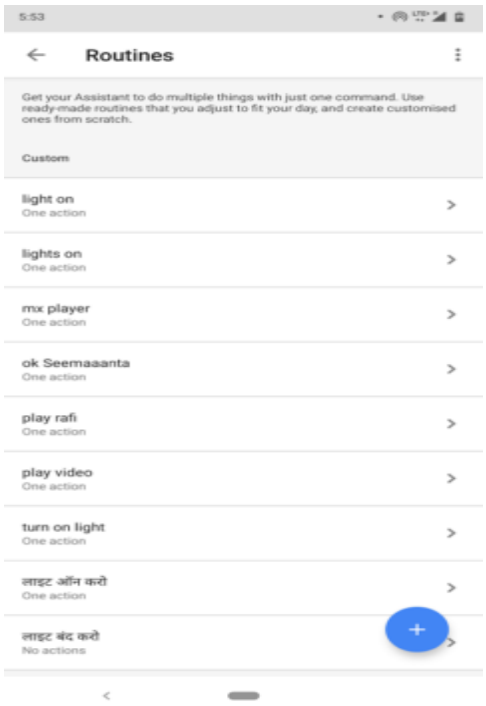

Figure 6- Google assistant routines settings

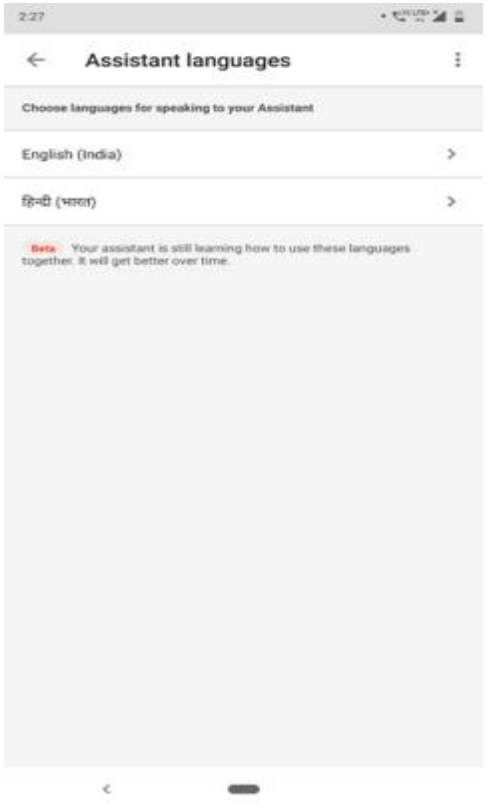

Figure 7- Google assistant command settings

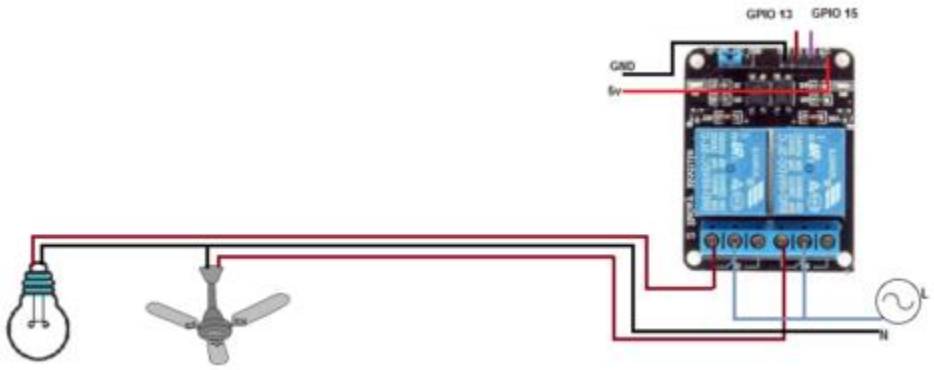

Figure 8- Shows the connection between the relay, light and raspberry pi

Figure 9- The picture above shows the physical connection enclosed into a box

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.esmart.myapplication">

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="EFY Home Automation"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="EFY Home Automation">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Figure 10- The code shows how to set the permission in manifest

Figure 11- the code shows how to create WebView

Figure 12- This figure shows how to create the code to load URLs of RPi server

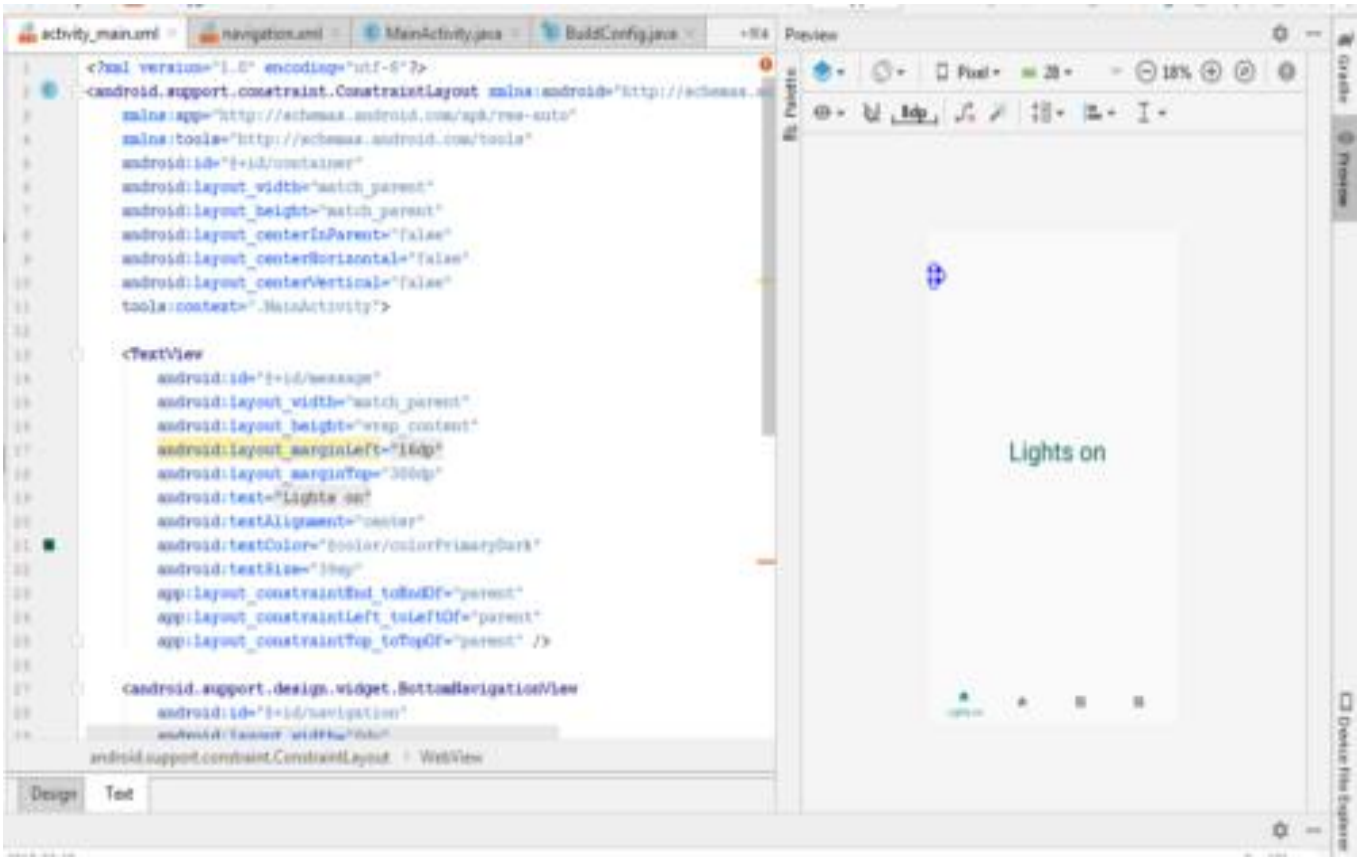

Figure 14- This is creating the navigation bar