



Multi-Language Voice Control Home Automation Using Raspberry Pi

¹Mr.K.Srikanth, ²Vemula Sruthi, ³Vallangi Laxmi Prasanna

⁴Putta Anaghalakshmi

¹Assistant Professor ^{2,3,4}UG Scholar

Balaji Institute of Technology & Science

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 enabled 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 aptget 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.

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.
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.

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

```

GNU nano 2.7.4
rphp
system("gpio -g mode 22 out");
system("gpio -g write 22 1");
?>
    
```

Figure 2- This is the code in PHP

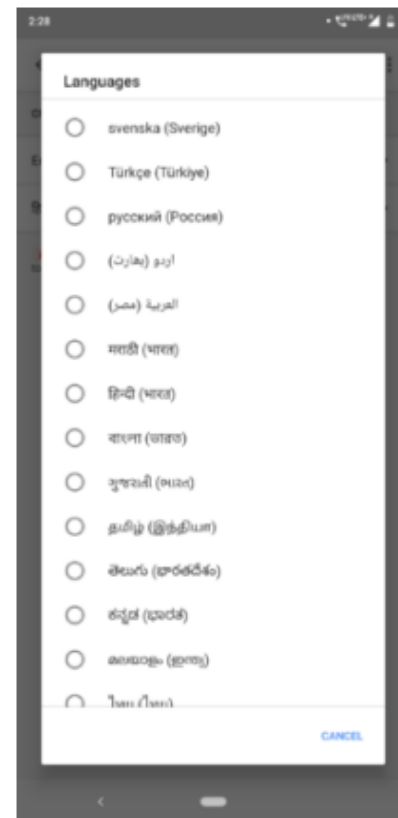


Figure 3- Setting up the language in google assistant

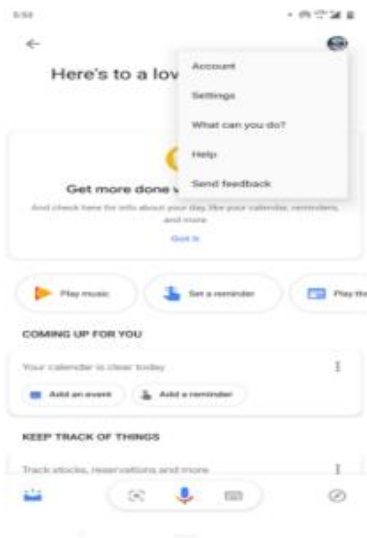


Figure 4- Finding the google assistant settings

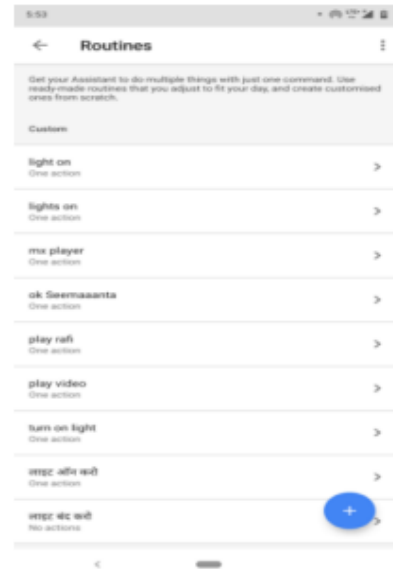


Figure 6- Google assistant routines settings

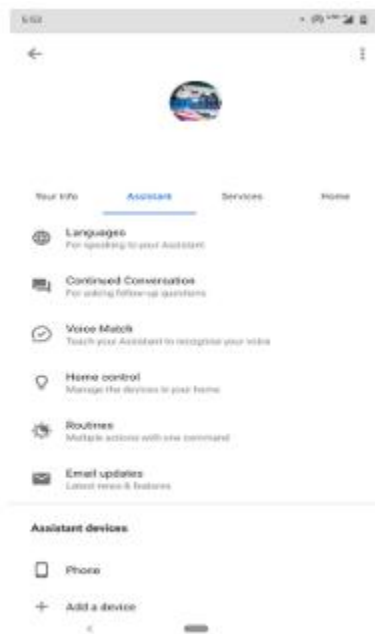


Figure 5- Google assistant setting menu



Figure 7- Google assistant command settings

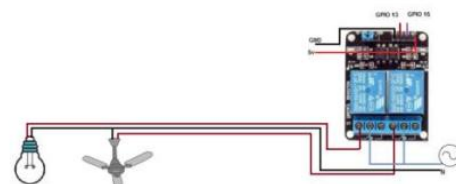


Figure 8- Shows the connection between the relay, light and raspberry pi



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

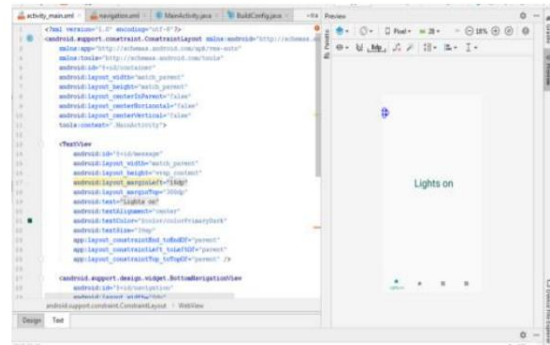


Figure 14- This is creating the navigation bar

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.smarthome.myapplication">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:exported="true">
            <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

```
package com.smarthome.myapplication;

import android.os.Bundle;
import android.support.design.widget.BottomNavigationView;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.design.widget.BottomNavigationView;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;

public class MainActivity extends AppCompatActivity {
    private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener
        = new BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            return false;
        }
    };

    private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener
        = new BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            return false;
        }
    };
}
```

Figure 11- the code shows how to create WebView

```
private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener
    = new BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        return false;
    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

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

REFERENCES

- [1]. Changsu Suh and Young-Bae Ko, "Design and Implementation of Intelligent Home Control Systems based on Active Sensor Networks", IEEE Transactions On Consumer Electronics, Vol 54, NO. 3, AUGUST 2008.
- [2]. Vikas Kumawat¹, Shubham Jain², Vikram Vashisth³, Neha Mittal⁴, Bhupendra Kumar Jangir⁵, "Design of Controlling Home Appliance Remotely Using Raspberry pi", 2017 2nd International Conference for Convergence in Technology.
- [3]. Souveer Gunpath, Anshu Prakash Murdan, Vishwamitra Oree, "Design and Implementation of a Low-Cost Arduino-Based Smart Home System", 9th IEEE International Conference on Communication Software and Networks, IEEE 2017.
- [4]. Mile Mrinal and Lakade Priyanka, Mashayak Saniya, Katkar Poonam and A.B. Gavali, "Smart Home – Automation and Security System Based on Sensing Mechanism", 2017 IEEE.
- [5]. H Bharathi¹, U Srivani¹, MD Azharudhin¹, M Srikanth¹, M Sukumarline¹, "Home Automation by Using Raspberry Pi And Android Application", International Conference on Electronics,



International Journal For Advanced Research In Science & Technology

A peer reviewed international journal

www.ijarst.in

IJARST

ISSN: 2457-0362

Communication and Aerospace Technology,
IEEE 2017.

[6]. Dhiraj Sunehra, SMIEEE, Vemula
Tejaswi, Implementation of Speech Based
Home Automation System using Bluetooth
and GSM.

[7].

[https://www.raspberrypi.org/downloads/rasp
bian/](https://www.raspberrypi.org/downloads/raspbian/)

[8].

[https://sourceforge.net/projects/win32diskim
ager/](https://sourceforge.net/projects/win32diskimager/)