

Design and Implementation of Voice Controlled Home Automation System

BY

**Md. Nahid Hasan
ID: 151-19-1644**

**Md. Emdadul Haque
ID: 151-19-1687**

**Md. Golam Rabbi
ID: 151-19-1719**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Electronics and Telecommunication Engineering

Supervised By

Engr. Md. Zahirul Islam

Assistant Professor

Department of ICE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2019

APPROVAL

This Project titled “**Design and Implementataion of Voice Controlled Home Automation System**”, submitted by Md. Nahid Hasan and Md. Emdadul Haque And Md. Golam Rabbi to the Department of Information and Communication Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Electronics and Telecommunication Engineering and approved as to its style and contents. The presentation was held on January 2019.

BOARD OF EXAMINERS



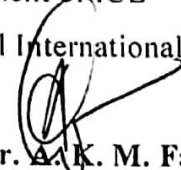
Md. Taslim Arefin

Chairman

Associate Professor and Head

Department of ICE

Daffodil International University



Prof. Dr. A. K. M. Fazlul Haque

Internal Examiner

Professor

Department of ICE

Daffodil International University



Dr. Engr. Quamruzzaman

Internal Examiner

Professor

Department of ICE

Daffodil International University



Dr. Subrata Kumar Aditya

External Examiner

Professor

Department of EEE

University of Dhaka

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Engr. Md. Zahirul Islam, Assistant Professor, Department of ICE**, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Engr. Md. Zahirul Islam

Assistant Professor

Department of ICE

Daffodil International University

Submitted by:

EMDADUL

(Md. Emdadul Haque)

ID: 151-19-1687

Department of ICE

Daffodil International University

Nahid

(Md. Nahid Hasan)

ID: 151-19-1644

Department of ICE

Daffodil International University

RABBI

(Md. Golam Rabbi)

ID: 151-19-1719

Department of ICE

Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete this project successfully.

We felt grateful to and wish our profound our indebtedness to Supervisor **Engr. Md. Zahirul Islam, Assistant Professor**, Department of ICE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of wireless network influenced us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Md. Taslim Arefin, Associate Professor, and Head, Department of ICE, for his kind help to finish our project and also to other faculty member and the staff of ICE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents

DEDICATION

THIS PROJECT IS DEDICATED

TO

OUR PARENTS

ABSTRACT

Voice controlled home automation framework is the destiny of private related advancement which is intended to convey and circulate number of administrations inside and outside the house by methods for sorted out contraptions in which all the diverse applications and the knowledge behind them are coordinated and interconnected. This framework is developed for the individuals who are impaired and unfit to move without the supporting of others. To solve this problem, a system is developed where the home appliances can be possible to control through voice command easily. In this project, Raspberry pi utilized as google assistant client to process the voice direction and connected to the home machines.

TABLE OF CONTENTS

Contents	Page No.
1.1 Introduction	1
1.2 IoT	2
1.3 Voice Controlled Home Automation System.....	3
1.4 Goals of The Project.....	4
1.5 Organization	4
Basic Hardware & Implementations.....	5
2.1 Raspberry Pi	5
2.1.1 Specifications	5
2.1.2 General Purpose Input Output (GPIO)	7
2.2 USB Sound Card	9
2.3 Microphone	9
2.4 Speaker	10
2.5 DHT11 Sensor.....	11
2.6 Breadboard & LEDs.....	11
Software's Embedding.....	13
3.1 Configuring Raspberry Pi.....	13
3.1.1 Required Item:	13
3.1.1.1 Installing Rasbian Debian Wheezy Operating System using windows:.....	14
3.1.1.2 Format the SD card:.....	15
3.1.1.3 Using Etcher:	16
3.1.1.4 Plugging into the Raspberry Pi:.....	18
3.1.1.5 Logging into the Raspberry Pi:.....	18

3.1.2 Installing Sound System	19
3.1.2.1 Installation	19
3.1.2.2 Testing the sound system:.....	21
3.2 Getting Ready for Google Assistant.....	23
3.2.1 Configuring Google’s Developer Project.....	23
3.2.2 Registering the Device Model	25
3.3 Access Using OAuth 2.0	28
3.4 Installing Google Assistant SDK	29
3.4.1 Configuring Python Virtual Environment:	29
3.4.2 Installing Python Library Packages:	29
3.4.3 Generating the Credentials:.....	30
3.4.4. Setting up the Hot-word:	31
Working Methodology.....	35
4.1 System Architecture	35
4.2 Overview of working procedure	36
Conclusion	38
Future Improvement.....	39
References.....	40

LIST OF FIGURES

Figure 1: Raspberry Pi [4].....	5
Figure 2: System on Chip Broadcom BCM2837 [3]	6
Figure 3: General Purpose Input Output pins [16].....	7
Figure 4: (a) GPIO.BOARD (b) GPIO.BCM Layouts [16].....	8
Figure 5: USB Sound Card	9
Figure 6: Microphone	10
Figure 7: DHT11 Sensor	11
Figure 8: Breadboard [10].....	12
Figure 9: LED [12].....	12
Figure 10: Raspberry Pi 3 B [4].....	13
Figure 11: Select the SD Card to Format.....	15
Figure 12: Formatting the SD Card	16
Figure 13: Flashing with Etcher.....	17
Figure 14: Done Flashing.....	17
Figure 15: Raspberry Pi Home page.....	19
Figure 16: Sound System Configuring	21
Figure 17: AlsaMixer	22
Figure 18: Google Developers	23
Figure 19: Googles Action Console.....	23
Figure 20: Selecting name for new project.....	24
Figure 21: Device Registration	24
Figure 22: Registering Device Model.....	25
Figure 23: Device Specifications	26
Figure 24: Downloading Credentials	26
Figure 25: Selecting Traits.....	27
Figure 26: Overview of Registered Model	27
Figure 27: Googles OAuth 2.0 Protocol	28
Figure 28: Sample Assistant Running.....	31
Figure 29: Code for Controlling GPIO	33
Figure 30: System Architecture	35

Figure 31: Flow Chart..... 36
Figure 32: Implimentation of voice control home automation 37

Chapter 1

Introduction

1.1 Introduction

Science is keeping pace with time. Changes are coming in time with people's thinking. Present world is working for making the human life easier and more comfortable. There have been several sectors where the present world is working for these purposes. We are trying to build a project that will make life easier and comfortable by reducing human work.

Our project is home automation system. Home automation system is one of the hot topics in research area in current scenario. There have been several types of home automation system such as Bluetooth controlled home automation system, Remote (IR remote) controlled home automation system, Internet controlled home automation system and RFID based home automation system. But we are building a home automation system that is based on the human voice command. Home appliances like our daily uses light and fan etc. can be controlled through our voice command. Such as when a human being command as “lights ON”, it will ON for him and when says “lights OFF”, it will also OFF. This project is also very helpful for those people those who are physically disabled or unable to move their body properly. They can easily full fill their requirement by voice command without any help of other person.

We are introducing a hardware that is called raspberry pi in our project. This is a special type of device which has significant features,

- (i) Built in Wi-Fi and Bluetooth
- (ii) SoC: Broadcom BCM2837
- (iii) CPU: 1.2 GHZ quad-core ARM Cortex A53
- (iv) GPU: Broadcom Video Core IV @ 400MHz
- (v) Memory: 1 GB LPDDR2-900 SDRAM
- (vi) USB ports: 4
- (vii) Network: 10/100 MBPS Ethernet, 802.11n Wireless LAN, Bluetooth 4.0

In this project we use google assistant SDK as a client. When the whole system is done and prepared to take the voice command and work through our voice it generates huge amounts of data. We need to sync these data with the databases. When the data is generated, it automatically uploads to our databases.

The purpose of these projects is to make human life more comfortable and at the same time relieve human pressure. Such as,

- (i) Control the home appliances over human voice command;
- (ii) People who are physically disabled are more benefited;
- (iii) Maintain the database based on our use;
- (iv) Can compare with our daily use of electricity;
- (v) Speech recognition can play an important role for the security.

1.2 IoT

The Internet of things (IoT) is a modern communication technology that envisions a near future, in which the objects of daily life will be equipped with microcontrollers, transceivers and appropriate protocol stacks that will establish a communication medium with other persons. The Internet of things or IoT makes people familiar with the internet, which will be helpful for the forthcoming world. The present world is discovering day by day, but people make it closer to ours by their latent talent. To make this vast world smaller, the internet plays an effective role.

The possibility of the Internet of Things was not authoritatively named before 1999. The starting point of this idea, the instances of an Internet of Things, is from the mid 1980s, and it was a Coca Cola machine which occurred at the Carnegie Mellon University. Neighborhood software engineers would interface by Internet to the refrigerated apparatus, and it would verify whether there was an accessibility to drink, and in the event that it was cool, before making the trek. From 2013, the Internet of Things had achieved a framework utilizing various innovations, changing from the Internet to the remote correspondence and from the small scale electromechanical frameworks (MEMS) to the implanted frameworks. The customary fields of

computerization remote sensor systems, GPS, control frameworks, and others, all help the IoT. As such, we expressed that, the Internet of Things are the arrangement of any gadget with an on/off change associated with the Internet. This incorporates nearly anything you can consider, extending from mobile phones to building support to the stream motor of a plane. Therapeutic gadgets, similar to a heart screen embed, can exchange information over a system and that are the individuals IoT. In the event that it has an off/on switch, it can, hypothetically, be a piece of the framework. The IoT framed of a substantial system of web associated with the gadgets. To Ring, a doorbell which is linkup to our advanced mobile phone be a shrewd case of an ongoing expansion to the Internet of Things. Ring signs to our phone when the doorbell is squeezed, and gives we a chance to can see who is squeezed the ringer and can chat with him [1].

While the Internet is a communication network connecting people to the information, the Internet of Things (IoT) is an interconnected ecosystem of uniquely addressable physical objects with varying degrees of sensing, processing, and actuation capabilities, sharing the ability to communicate and interoperate through the Internet as their common denominator [1].

Intelligent devices are the enough intelligence which makes the human task simple. They don't need to wait for the human decision, now they can take decision based on the situation and can implement it. IoT creates more opportunities for ours to make this world more livable and we should utilize it properly to be a part of this first world citizen. If we think more technically, we will find that how much benefited we are, if we implement it in a proper way. Almost all the thing in our life is possible to take closer to the touch of IoT. From morning to till night before going to the bed everything in our life is possible to make it to a part of the IoT which relief our pressure to look after a lot of thing and also think about it.

1.3 Voice Controlled Home Automation System

Automation performs a key role in our life. Home automation system allows the people to control the household electrical appliances like light, door, fan, AC etc. At the same time, it also provides the home security. Home automation system not only notices to reduce human efforts but also

energy efficiency and time saving. The main objective of home automation system is crippled and old aged people those who will be able to control the home appliances.

1.4 Goals of The Project

Lots of works and research have already been done with home automation system. Different people done their automation system in different ways. We are considering voice command to control our home. It enhances a new chapter in the technology. People can communicate their household appliances (like lights and fans etc.) through their voice command. Save the energy and time. The main objective of home automation system is crippled and old aged people those who will be able to control the home appliances.

1.5 Organization

In this project report the first chapter cover the basic concept of IoT and the voice-controlled home automation system. The second chapter contains the theory of raspberry pi, concept about USB Audio Hub Microphone and Speaker and the features of DHT11 Sensor. In third chapter we discussed about installing raspberry pi and google assistant SDK. Fourth chapter contains the Controlling Devices with Assistant SDK, chapter five discussed about setting up the sensors. Chapter six and seven cover the topic on future implementation and conclusion.

Chapter 2

Basic Hardware & Implementations

2.1 Raspberry Pi:

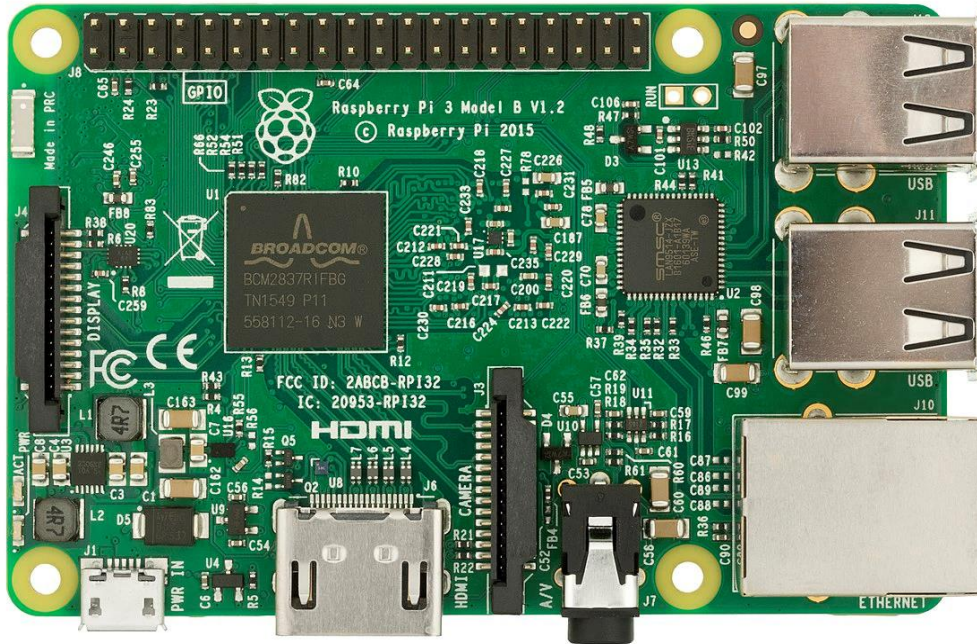


Figure 1: Raspberry Pi [4]

The Raspberry Pi which is one kind of small computer build with a single board as compared as our normal computer, usually it called pi device or RPI. Raspberry Pi Foundation developed it in the United Kingdom for prompting school students to teaching basic computer science [2]. Now-a-days it comes with GPIO port which makes it more expandable for science and development projects.

2.1.1 Specifications

In our project we used Raspberry Pi 3, according to Raspberry Pi Foundation promotion it has almost everything supportable.

System on Chip (SoC): This device uses Broadcom BCM2837 chip for high performance where four ARM Cortex-A53 Processor core working at 1.2 GHz with Level 1 32kB and Level 2 512kB cache memory [4].



Figure 2: System on Chip Broadcom BCM2837 [3]

The Pi 3 has 1GB LPDDR2 RAM which work at 900MHz it's also makes it faster than other older versions and Raspberry Pi zero W. Also, it has Broadcom Video-Core IV type GPU.

Networking: Raspberry Pi 3 has 2.4GHz 802.11n Wireless build in that's make this device more reliable than Pi 2 which had only Ethernet, as well as it has 10/100 Ethernet connectivity also. So that it doesn't required an external antenna for connecting wirelessly. One of the most useful wireless application Bluetooth 4.1 is also used in this board which consume low energy that's important for such device which is runs in 5V dc [3].

Ports: it has 3.5mm analogue audio-video jack HDMI, it doesn't require extra usb hub generally because it has 4× USB 2.0. Ethernet, Display Serial Interface (DSI) Camera Serial Interface (CSI), is another good feature of The Raspberry Pi 3 which shares the same SMSC LAN9514 chip [4].

2.1.2 General Purpose Input Output (GPIO)

The most useful features for developers the GPIO pins, these pins allow us to communicate with other circuitry such as extension boards sensors custom circuits and much more. All of these pins are operable with python code [4]. The GPIO's designed for giving output at high 3V or low 0V

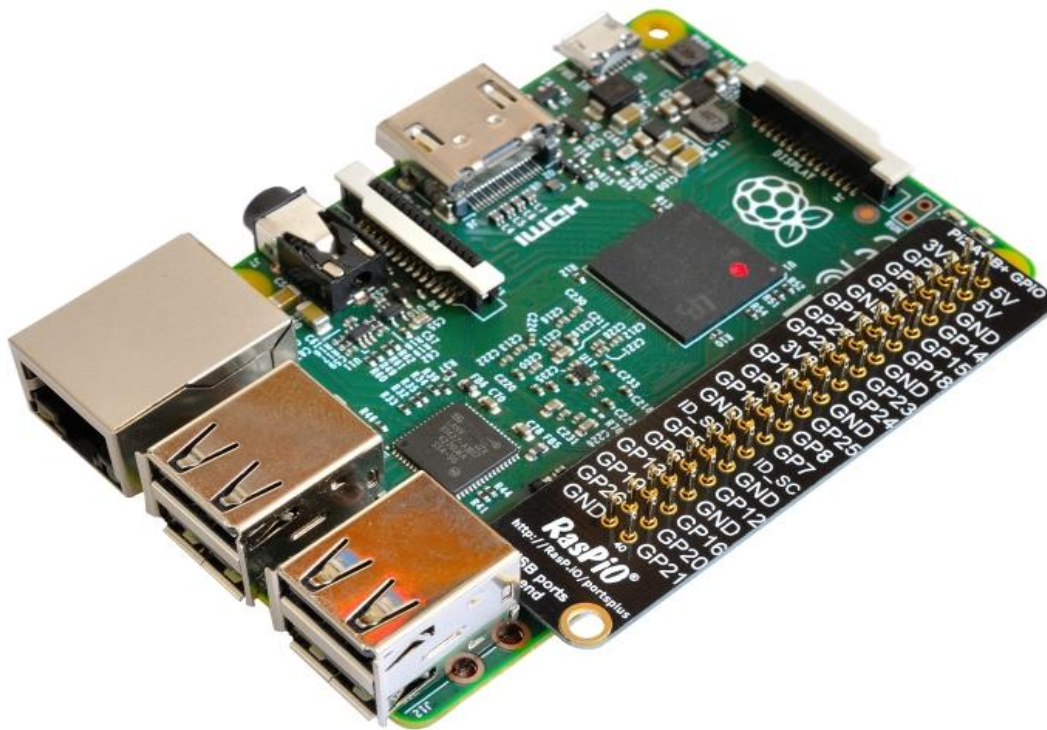


Figure 3: General Purpose Input Output pins [16]

and as like as input also. It has two 5V pins and 2 pins has 3V output capabilities. These pins are not only simply used as input output but also it can be used for some alternative functions:

Serial: Generally, GPIO14 & GPIO15 pins are used for TX pin & RX pin respectively.

Serial Peripheral Interface (SPI): GPIO10 is used for Master Output Slave Input (MOSI) or SPI0 GPIO9 for Midcontinent Independent System Operator (MISO), GPIO11 works as Serial Clock (SCLK), GPIO8 act as Chip Enable Zero (CE0) and GPIO7 for CE1.

Pulse Width Modulation (PWM): Although it expressed before all GPIO pins of raspberry pi devices are software PWM available and hardware PWM available on GPIO12, GPIO13, GPIO18 and GPIO19.

Serial Communication Protocol: Raspberry pi uses bi-directional two-wire (I2C) serial bus that provides a communication link between ICs [5]. In device GPIO2 act as data pin and clock GPIO3, EEPROM Clock GPIO1 and EEPROM Data GPIO0 pins [4].

The numbering system of GPIO pins is not in numerical order, there are two available numbering system one GPIO.BOARD and other's GPIO.BCM. The GPIO.BOARD numbering system specifies the pins as like *Figure: 4* and GPIO.BCM refer pins as Broadcom SOC channel

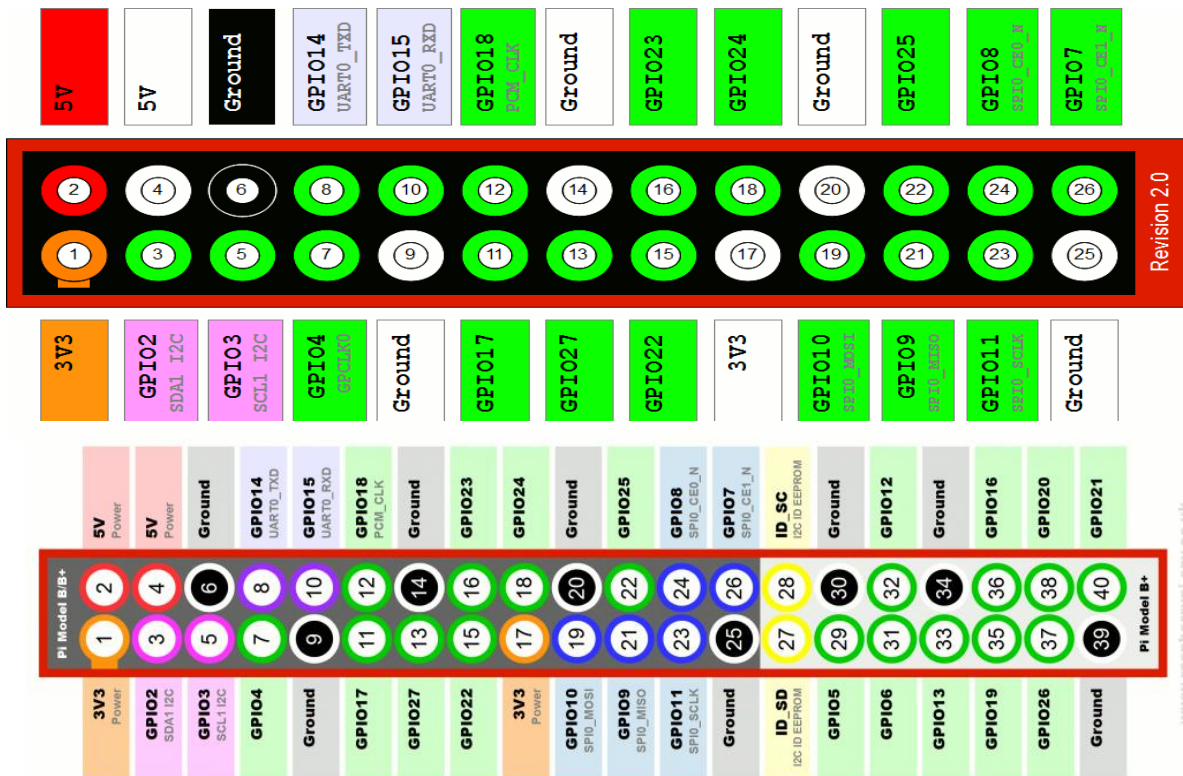


Figure 4: (a) GPIO.BOARD (b) GPIO.BCM Layouts [16]

numbering system as like as *Figure: 5*. The BOARD numbering system may vary with the model of raspberry pi devices [6].

2.2 USB Sound Card

The USB sound card is very simple and common device for all computer user, this device is one kind of hub which convert digital audio code to analog audio using DAC converter and send to 2.5mm audio output jack through speaker, on the other hand it convert analog audio signal to digital signal which gotten from microphone as user voice [7]. Todays sound cards are coming build in with motherboard, Raspberry Pi 3 has also build in sound card with 3.5mm audio output jack also with HDMI output capability. In this project we used Sound Card for input our voice through microphone so that the voice command shall executed.



Figure 5: USB Sound Card

2.3 Microphone

Microphone is a transceiver service that converts the sound waves into electrical signal. Different types of microphone have different techniques to convert the signal. Electrical microphone was invented in late 19th century and on that time there had not been any satisfactory way to send the audio signal from one place to another [8].



Figure 6: Microphone

The microphone generates the electrical current is very small. Signal measured in microphone is millivolts. When the diaphragms are vibrated, it causes the other component in microphones are vibrated. These vibrations are converted into electrical current which becomes the audio signal.

2.4 Speaker

To translate an electrical signal into an audible sound, speaker contains an electromagnet. Magnetic field is generated when electric current flows through a metal coil. The act of the coil is like a normal magnet.

The electromagnet is attached to a cone which is made of a flexible material such as paper or plastic which amplifies these vibrations, pumping the sound wave into the enclosure air and towards our ears.

Inside a speaker

- (i) Cone,
- (ii) Electromagnet,
- (iii) Permanent Magnet. [9]

2.5 DHT11 Sensor

DHT11 is a low cost, digital temperature and humidity sensor.

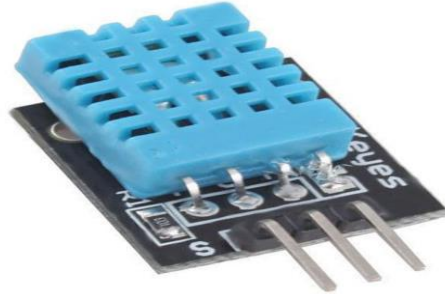


Figure 7: DHT11 Sensor

DHT111 sensor specification

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Humidity Range: 20% to 90%
- Temperature Range: 0°C to 50°C
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: $\pm 1^\circ\text{C}$ and $\pm 1\%$

2.6 Breadboard & LEDs

A breadboard is a simple device designed to let us create circuits without the need for soldering. They come in different sizes and the design may vary, but as a general rule the breadboard is look like this:

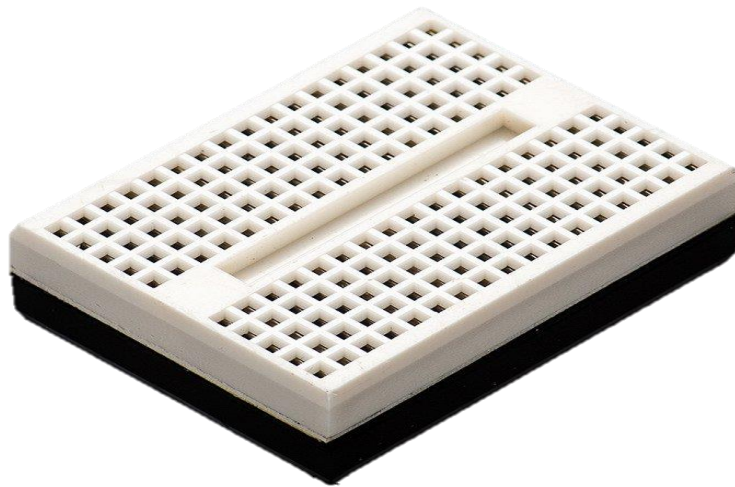


Figure 8: Breadboard [10]

Light Emitting Diode (LED) is the two led semiconductor light sources. LEDs are just like small light bulbs that sets up easily |an electrical circuit. Unlike ordinary luminous bulbs, they don't have a filament that will burn out, and it never get especially hot. They are illuminated only by the movement of electrons in a semiconductor material. The lifespan of an LED bulb by thousands of hours. It last just as long as a standard transistor. These tiny LEDs are already replacing the tubes that light up LCD/LED HDTVs to make the televisions thinner. [11]

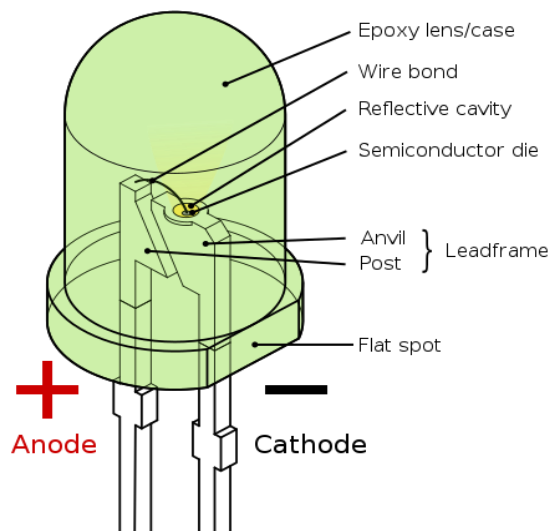


Figure 9: LED [12]

Chapter 3

Software's Embedding

3.1 Configuring Raspberry Pi

To configure the raspberry pi we divide it several sections.

3.1.1 Required Item:

- A raspberry pi 3 model B

To load the graphical user interface, we need to type startx and press Enter oor keyboard.

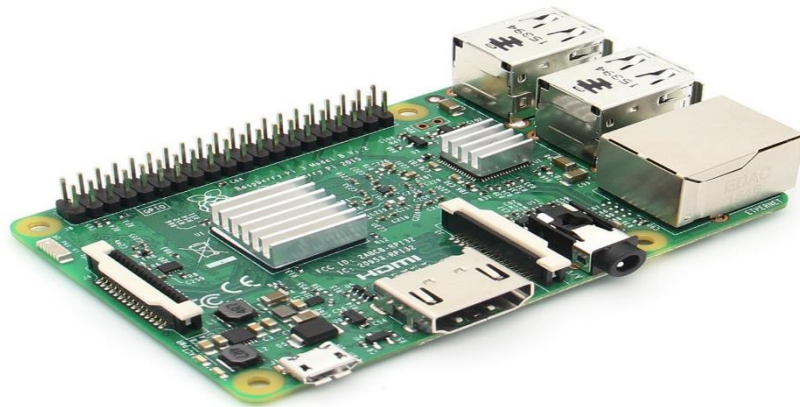


Figure 10: Raspberry Pi 3 B [4]

- SD Card:
 - 8GB Class 4 SD card
- Display and connecting cables:
 - HDMI/DVI monitor which we can use as display
 - For best results, use one with HDMI input, but other connections are available for older devices.

- Keyboard and Mouse:
 - USB keyboard and mouse will work with our Raspberry Pi.
- Power supply:
 - We use 5V micro USB power supply.
- Internet connection:
- For updating and downloading the required packages, we have to associate our Raspberry Pi to the web either by means of an Ethernet link or a Wi-Fi connector.
- Sound:
 - Headphones or speakers with a 3.5mm jack will work with our Raspberry Pi.

3.1.1.1 Installing Rasbian Debian Wheezy Operating System using windows:

- http://downloads.raspberrypi.org/raspbian_latest

Download the file “RASPBIAN Debian Wheezy.zip” file and extract the image file.

- Embed the SD card into our SD card peruser (arrange the SDcard) and check which drive letter was doled out. We can without much of a stretch see that the drive letter (for instance F:) by looking in the left section of Windows Explorer. We utilize the SD Card opening or a shabby SD connector in a USB port
- Download the Etcher-Portable-1.4.4-x64 utility from the Source forge Project page.
<https://www.techspot.com/downloads/6931-etcher.html>

Extract the executable from the compressed file and run the Win32DiskImager utility. We may need to run the utility as an administrator.

- Select the image file we extracted above.

- Select the drive letter of the SD card in the device box. We should be careful to select the correct drive. If we get the wrong, we may destroy our data on the computer's hard disk! When we are using an SD card slot in our computer and can't see the drive in the Win32DiskImager window, try using a cheap SD adaptor in a USB port.
- After clicking the write button, we have to wait to complete the write.
- Exit the imager and eject the SD card.

3.1.1.2 Format the SD card:

- Allocate the SD card drive, in browser, and secondary-click the mouse to bring up the context-sensitive menu. From the menu select “Format”. We have to ensure that the option FAT32 is selected and then click Start, to start the formatting.

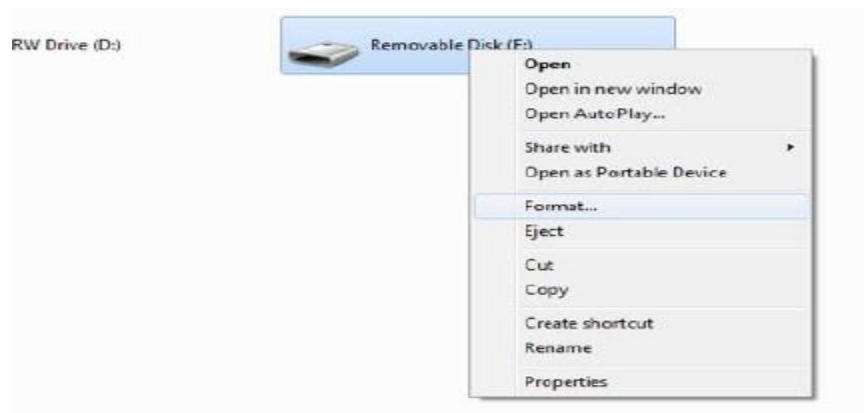


Figure 11: Select the SD Card to Format

- A few moments later you will see a confirmation that the format has been completed and you SD card is now ready for the next stage.

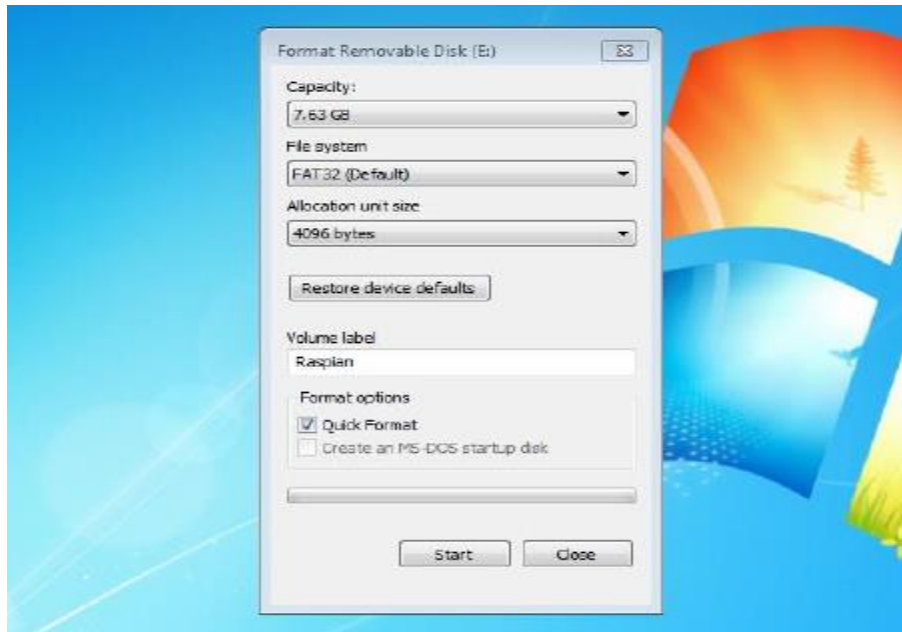


Figure 12: Formatting the SD Card

3.1.1.3 Using Etcher:

- Having plugged in our SD card, start or restart Win32Diskimager. Selecting a drive to copy the image to (in my case F:).

Choose the drive with SD card to write the OS image on

- Then click on the folder icon and choose the unzipped .img file from earlier that we want to put on the SD card. Then click Write, to write the Operating system on the card from the img file.

Write OS image from .img file to SD card

- We have to confirm to writing the OS.
- The progress bar will be showed that how much time it will take to install.

- When the installation is finished it will look like this.



Figure 13: Flashing with Etcher

- Now we can eject the card reader and remove the SD card. Then we can try it.



Figure 14: Done Flashing

3.1.1.4 Plugging into the Raspberry Pi:

1. Begin by slotting our SD card into the SD card slot on the Raspberry Pi, which will only fit one way.
2. Next, plug in our USB keyboard and mouse into the USB slots on the Raspberry Pi. We have to make sure that the monitor or TV is turned on, and that we have selected the right input (e.g. HDMI 1, DVI etc).
3. Then connect our HDMI cable from our Raspberry Pi to the monitor or TV.
4. If we intend to connect your Raspberry Pi to the internet, plug in an Ethernet cable into the Ethernet port next to the USB ports.
5. Finally, when we have plugged in all the cables and SD card required, plug in the micro USB power supply. This action will turn on and boot our Raspberry Pi.
6. If this is the first time to our Raspberry Pi SD card have been used, then we will have to select an operating system and configure it.

3.1.1.5 Logging into the Raspberry Pi:

- When the Raspberry Pi has completed the boot process, a login prompt will appear. The default login for Raspbian is username pi with the password raspberry. Note we will not see any writing appear when you type the password. This is a security feature in Linux.

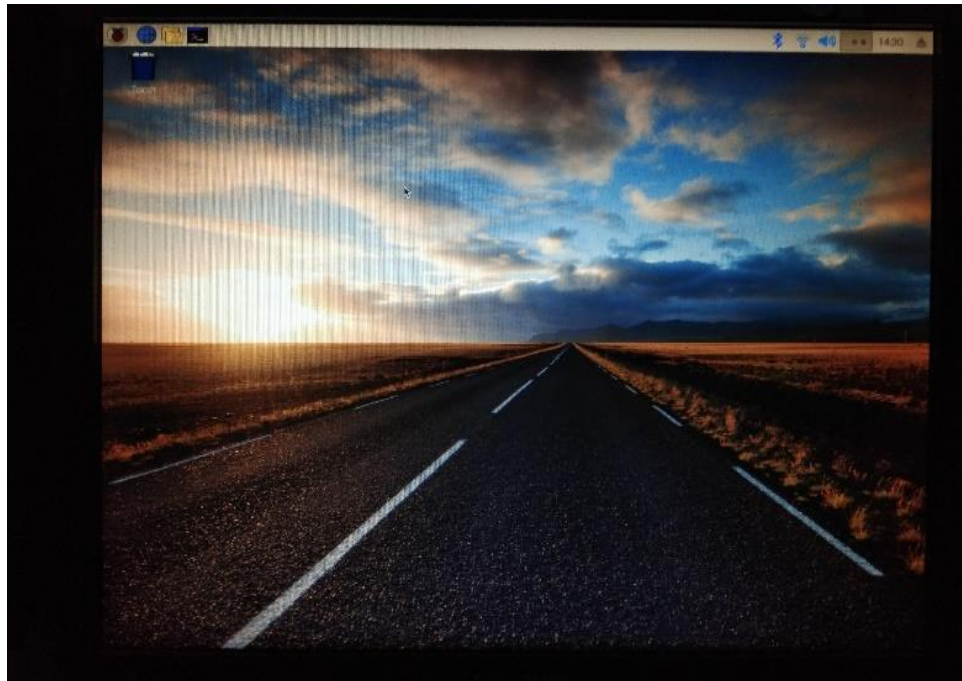


Figure 15: Raspberry Pi Home page

- After the successful logged in, we will see the command line prompt `pi@raspberrypi~$`.
- To get the graphical user interface (GUI), type `startx` and press Enter to our keyboard.

3.1.2 Installing Sound System

3.1.2.1 Installation

1. We have set up an account on the Google Actions Console. Now we have to configure the audio.

Started with setting up the audio on the Raspberry Pi, firstly we must get the card and device numbers for our various inputs and outputs.

1(a). Allocate the USB microphone by the command that are given below. Write down the card number and the device number for it.

```
arecord -l
```

1(b). Now we have to locate our speaker, we will be utilizing the following command. Again, writing the card number and device number.

Note: The Raspberry Pi's **3.5mm-jack** is typically labeled as **bcm2835 ALSA**, with the **HDMI output** being identified as **bcm2835 IEC958/HDMI**.

```
aplay -l
```

2. Now that we have taken our device and card numbers for both the microphone and our audio output, we need to create a file named as “**.asoundrc**” in the pi users home directory.

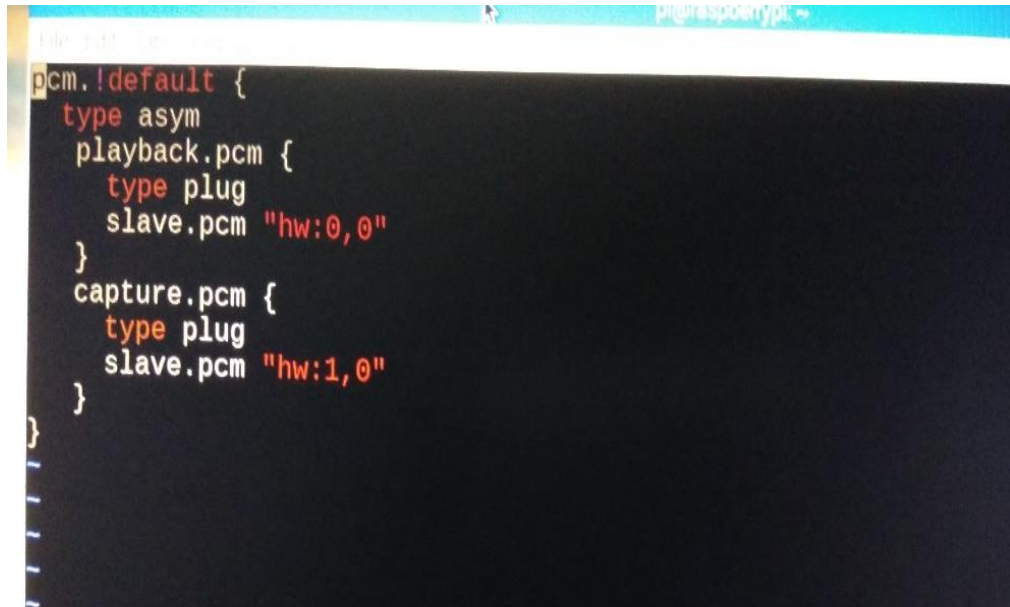
The file “**.asoundrc**” helps us to define which audio devices that the audio driver should be utilizing.

By running the following command to the terminal we can create the file:

```
nano /home/pi/.asoundrc
```

3. Within this file enter into the following lines.

We have to ensure that replacing <**card number**> and <**device number**> with their respective values that we regain during **Step 1**.

A screenshot of a terminal window on a Raspberry Pi. The terminal shows the configuration for the PCM (PulseAudio Core Module) in the ALSA configuration file. The configuration is as follows:

```
pcm.!default {
    type asym
    playback.pcm {
        type plug
        slave.pcm "hw:0,0"
    }
    capture.pcm {
        type plug
        slave.pcm "hw:1,0"
    }
}
```

The terminal window has a title bar that says "pi@raspberrypi ~".

Figure 16: Sound System Configuring

4. With the line, and change it. Press CTRL + X and press ENTER.

3.1.2.2 Testing the sound system:

- a) Setting up the Raspberry Pi Google Assistant and required API, we first test to make sure our audio is working.

At this moment we must have our USB microphone and speaker to attach with raspberry pi. By running the following command we test our speaker.

```
speaker-test -t wav
```

Now we can hear sound from the speakers. This sound is the person's voice.

- b) Now we are testing our microphone by making a recording. For recording we are running the following command on our raspberry pi.

```
arecord -format=S16_LE -duration=10 -rate=16000 -file-type=raw  
out.raw
```

This command will only be succeeded when it hear the microphone.

- c) As our recording is done now we can run the following command to read our raw output file and can play back it to the speaker.

Actually in this project we want that raspberry pi can't notice every noise but can notice the 'Ok Google'.

```
aplay -format=S16_LE -rate=16000 out.raw
```

- d) To solve the high or low recoding volume problem, we can run the following command for mixer.

```
alsamixer
```

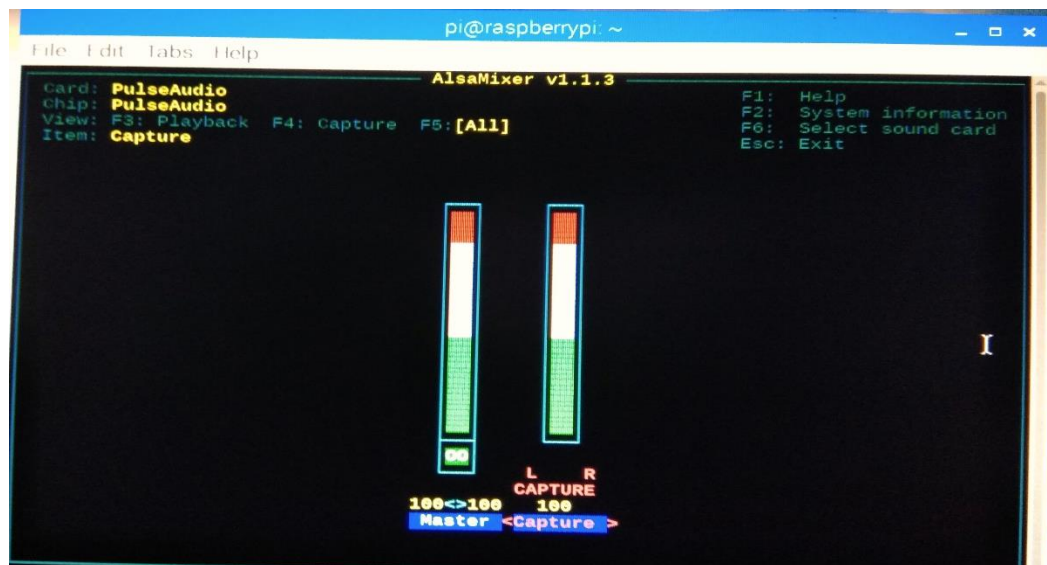


Figure 17: Alsamixer

3.2 Getting Ready for Google Assistant

3.2.1 Configuring Google's Developer Project

Google Developers site is stand for developing software, tools and application programming interfaces (APIs). In this site google provides dozens of technical resources as like Google Maps, Google Assistant SDK, Google Apps, YouTube etc.



Figure 18: Google Developers

As we explained previous section that, in our voice-controlled home automation project we used Google Assistant SDK for executing our voice command. For processing our command with google assistant we had to make a Google Developer Project on Google Developers website via following procedure.

- A. All of google cloud platform projects are managed by the googles action console. So that for enabling access of Google Assistant API we have to browse action console via following link. <https://console.actions.google.com> which opens a web page using a google account looks like [13].

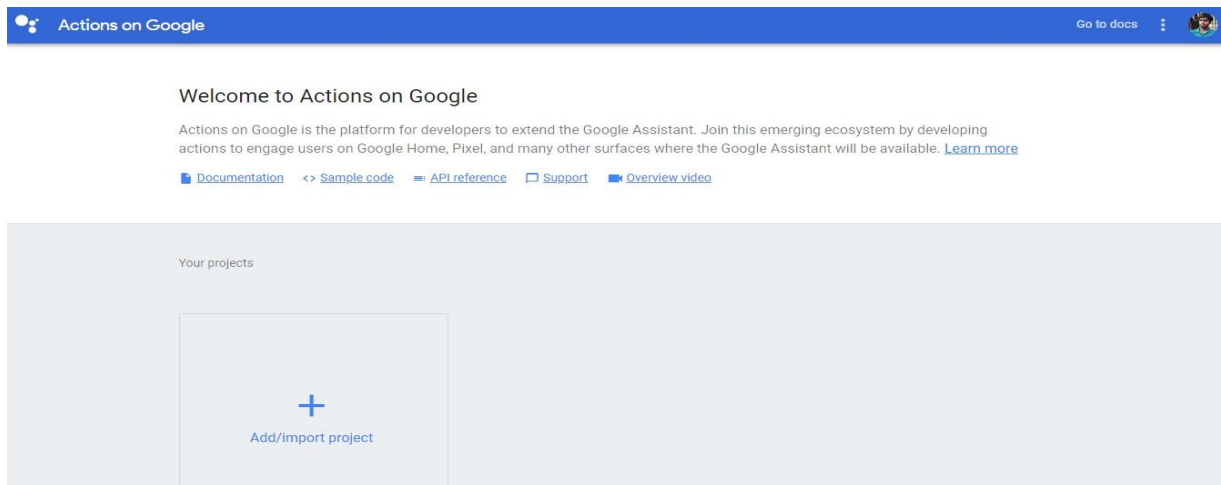


Figure 19: Googles Action Console

- B. Then we click on **Add/import project** for creating a new project which drive us another webpage where we have to accepted some googles term of conditions then the new project creating window shown up.

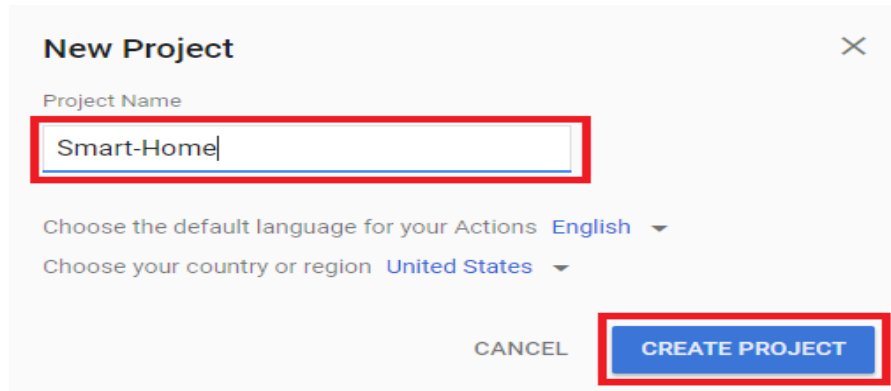


Figure 20: Selecting name for new project

Then we select the project name **Smart-Home** which is in red marked box and click create project button.

- C. After clicking create project button it opens another new window where we have to select the device.

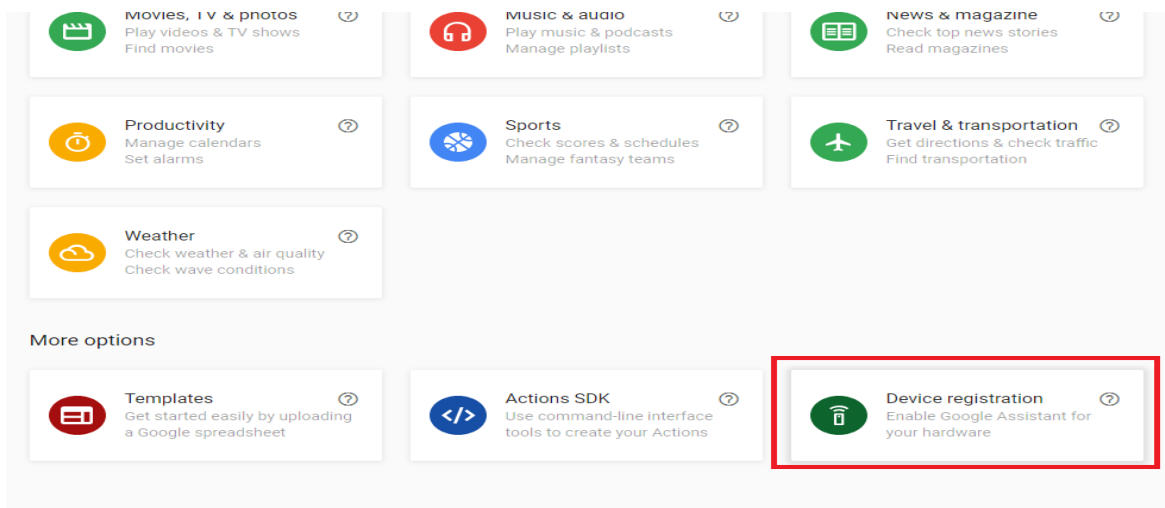


Figure 21: Device Registration

In this page, we select **Device registration** for enabling our Raspberry Pi device as a client of Google Assistant, and also by selecting this option it Drives us to the Device registration page and the by doing this the google developer project creating was done [13].

3.2.2 Registering the Device Model

To make Google Assistant responsive to our commands from pi device it needs information about our device. So that we have to register our device under the developer project which we created in previous section. This procedure makes google assistant accessible from our Raspberry Pi device.

A. As we done in last step of creating Googles Developer Project it opens the project UI mode which ask to register a model.

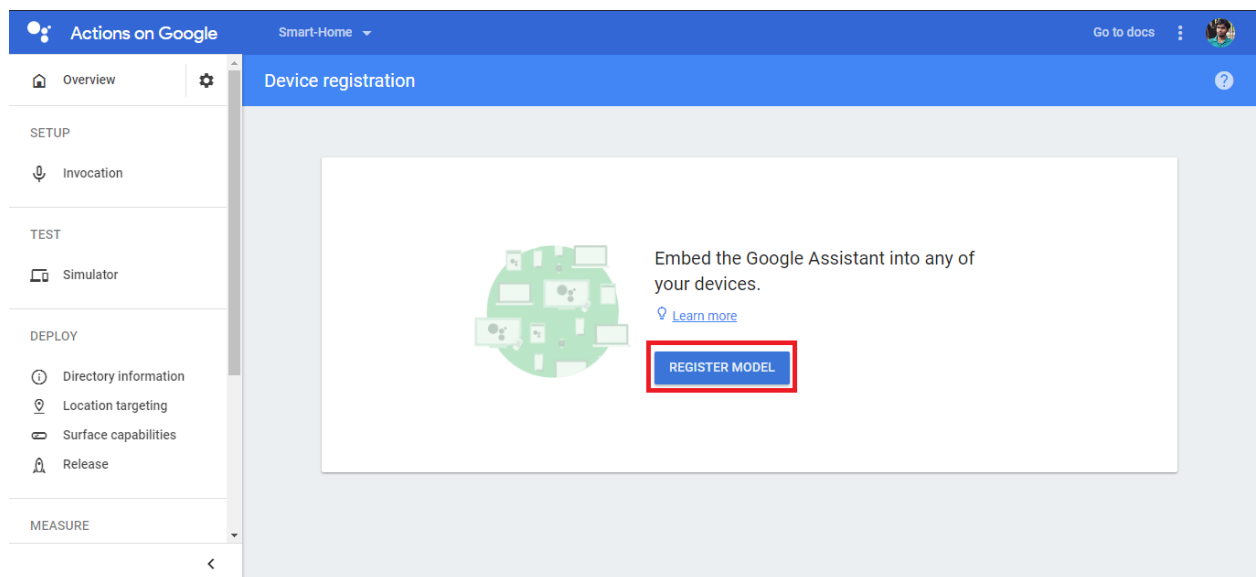


Figure 22: Registering Device Model

B. By clicking the **Register Model** button, it opens the register model window, in this window it asks some information about the device.

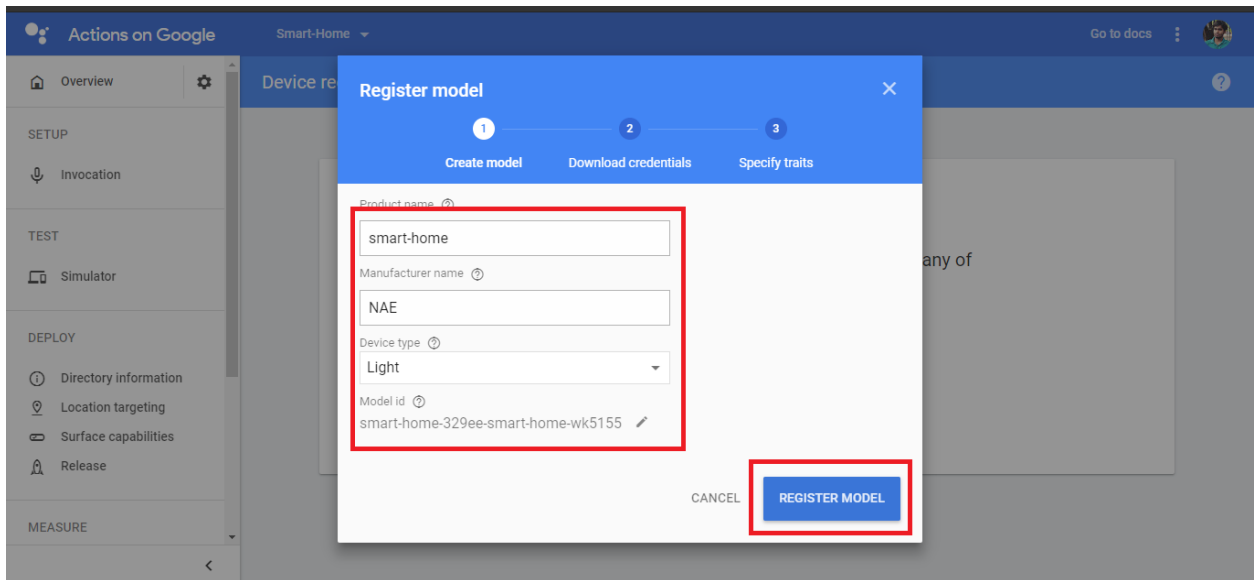


Figure 23: Device Specifications

After filling out with product name, manufacturer company name it provides a dropdown menu from which we select the category of product.

C. After taking basic information about our device it goes in next step.

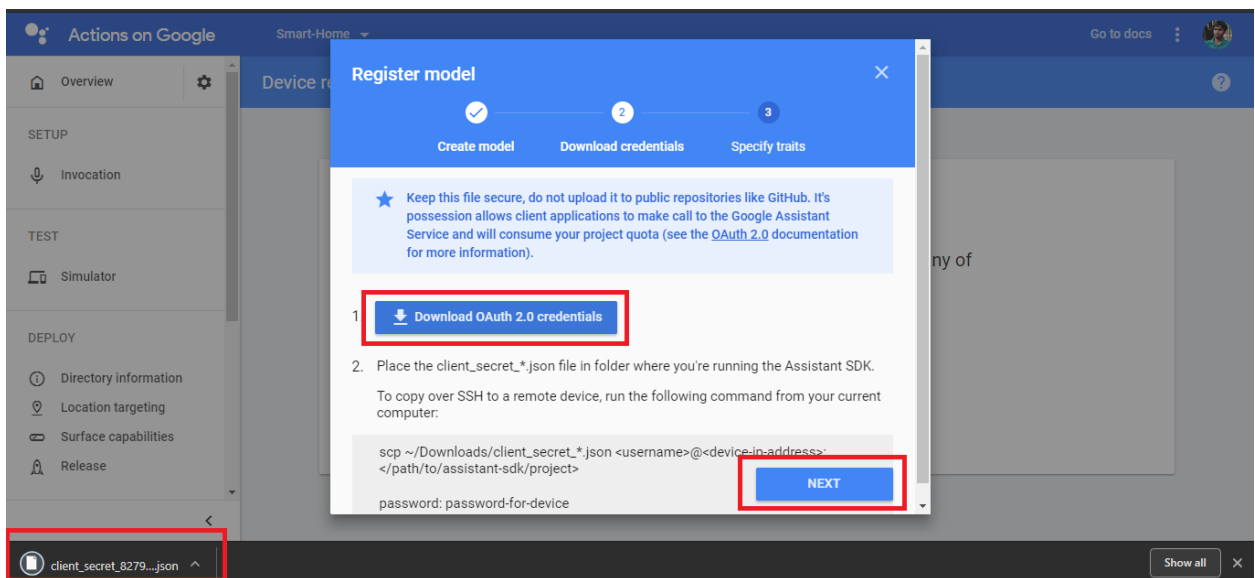


Figure 24: Downloading Credentials

Then it generates credential on basis of provided information which stored as .json file and appearing **Download OAuth 2.0 credentials** button. We download it and stored in our raspberry pi's *pi/home* directory.

D. After that it opens next step in which ask for specify traits.

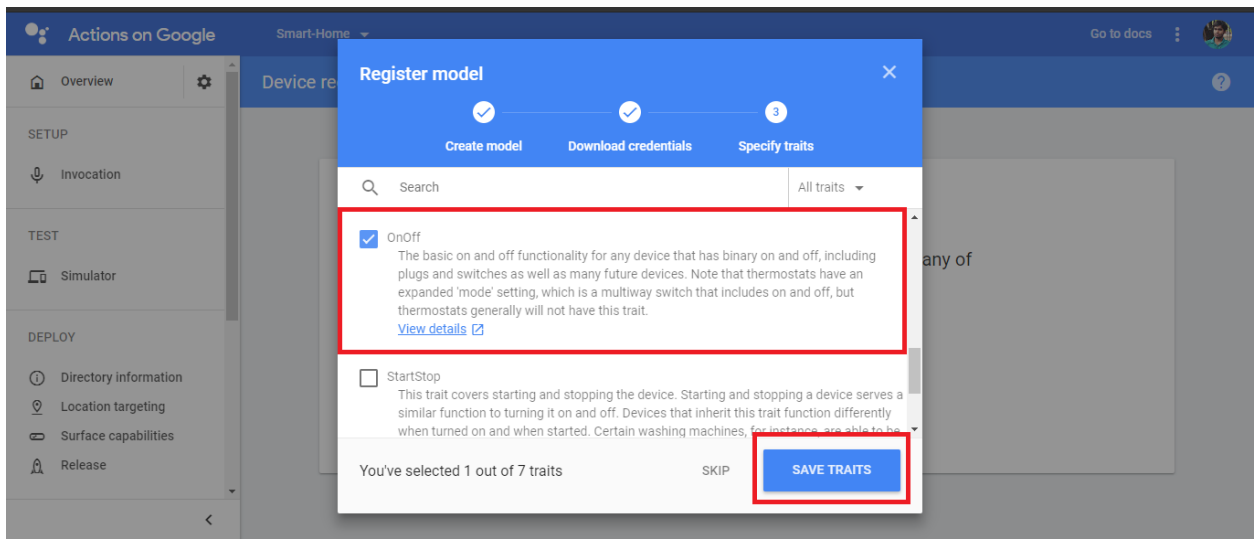


Figure 25: Selecting Traits

In this project we select only **OnOff** and **StartStop** trait, because we want to on off our devices like light, fans with this model.

Product name	Manufacturer name	Model ID	Device Type	Last updated time
smart-home	NAE	smart-home-329ee-smart-home-wk5155	action.devices.types.LIGHT	Dec 17,2018,09:41 PM

Figure 26: Overview of Registered Model

After clicking the **Save Traits** it done to creating the device for the project and its drive in this page which contain the device name and Model ID which we need for next step.

3.3 Access Using OAuth 2.0

In previous step we download a json file, which basically some sort of information that needs for authenticating our Pi device with Google Assistant SDK. Google API use OAuth 2.0 protocol for authentication.

To begin authentication with OAuth 2.0 protocol Google API console, obtain client credentials from json and when client application request for an access token from the Google Authorization Server it extracts a token from the response. An application first obtains an access token by granting access from API for accessing private data using Google API. The access token capable of varying degrees of access to multiple APIs [14].

Scenarios:

Googles OAuth 2.0 protocol supports several languages and frameworks such as PHP, Java, Python, Ruby, and ASP.NET as endpoint.

The procedure of authorization is beginning when our pi device redirects a link go Google URL including query parameters that indicates the type of access being requested. On the API side Google handles the user authentication and user consent as results it produces an authorization code from which the user and API exchange access token and refresh token [14]. Then the pi device uses the access token to access a Google API.

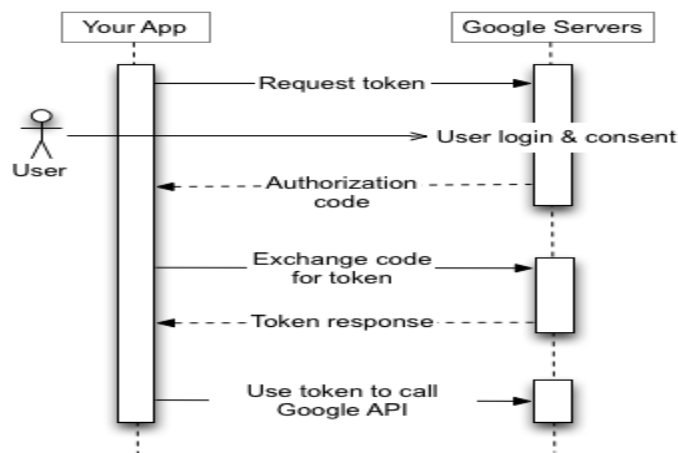


Figure 27: Googles OAuth 2.0 Protocol

After the operation the application stored the refresh token for future use the access token to access a Google API. And if the access token expires then application uses the refresh token to create a new token.

On the client side the google OAuth 2.0 authorization begins with redirecting a browser to a Google URL which includes query parameters that selects the type of access requested by end-user.

3.4 Installing Google Assistant SDK

3.4.1 Configuring Python Virtual Environment:

Use a Python virtual environment to separate the SDK and its dependencies from the system Python packages.

```
sudo apt-get update

sudo apt-get install python3-dev python3-venv # Use python3.4-venv
if the package cannot be found.

python3 -m venv env

env/bin/python -m pip install --upgrade pip setuptools wheel

source env/bin/activate
```

3.4.2 Installing Python Library Packages:

- (a) The Google Assistant SDK package contains all the code needed to get Google Support running on a device with sample code.

Installing the dependencies:

```
sudo apt-get install portaudio19-dev libffi-dev libssl-dev  
libmpg123-dev
```

By using the pip we can install the latest version of the Python packages in the virtual environment.

```
python -m pip install --upgrade google-assistant-library  
python -m pip install --upgrade google-assistant-sdk[samples]
```

3.4.3 Generating the Credentials:

a) Installing or updating the authorization tool.

```
python -m pip install --upgrade google-auth-oauthlib[tool]
```

b) Create credentials to be able to run sample codes and tools. Reference the JSON file that we downloaded in a previous step.

```
google-oauthlib-tool --scope  
https://www.googleapis.com/auth/assistant-sdk-prototype \  
-scope https://www.googleapis.com/auth/gcm --save --  
headless --client-secrets /path/to/client_secret_client-  
id.json
```

We have to look after the URL displayed in the terminal,

```
Please visit this URL to authorize this application: https://...
```


- c) Copy the URL and paste it into the browser. The page will ask to sign in to the Google account. Sign into the Google account that created the developer project in the previous step.
- d) After approving the permission request from the API, a code will appear in the browser. Copy and paste this code into the terminal:

```
Enter the authorization code:
```

- e) As the authorization was successful, we can see a response similar to the following:

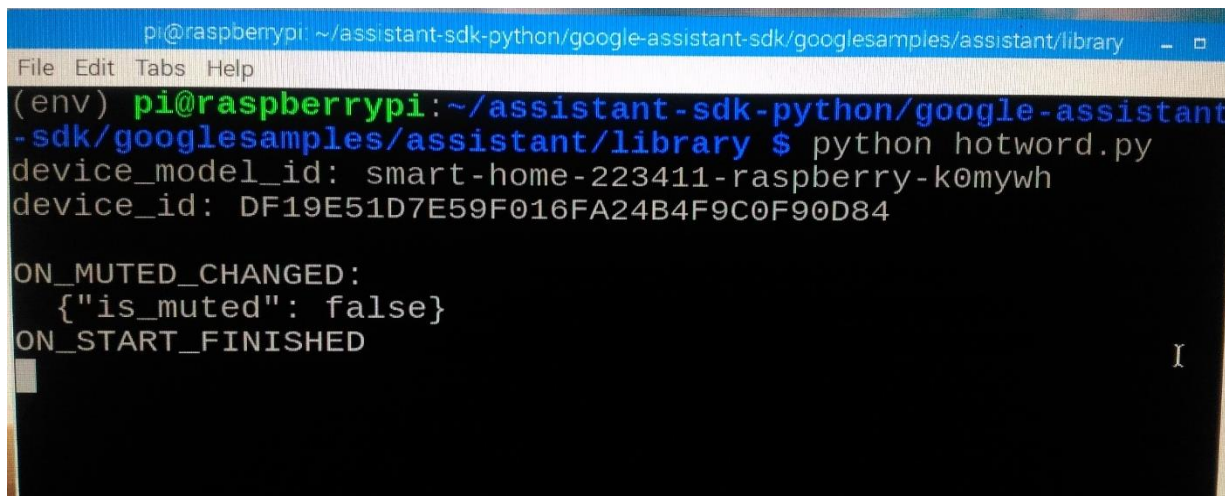
```
Credentials saved:/path/to/.config/google-oauthlib-  
tool/credentials.json
```

3.4.4. Setting up the Hot-word:

In this step its time for run assistant in our device. So that we run the command in terminal.

```
(env)$ googlesamples-assistant-hotword -project-id smart-home-  
223411 -user-model-id smart-home-223411-raspberry-k0mywh
```

This code runs the google assistant in pi device.



```
pi@raspberrypi: ~/assistant-sdk-python/google-assistant-sdk/googlesamples/assistant/library  
File Edit Tabs Help  
(env) pi@raspberrypi:~/assistant-sdk-python/google-assistant-  
-sdk/googlesamples/assistant/library $ python hotword.py  
device_model_id: smart-home-223411-raspberry-k0mywh  
device_id: DF19E51D7E59F016FA24B4F9C0F90D84  
  
ON_MUTED_CHANGED:  
  {"is_muted": false}  
ON_START_FINISHED
```

Figure 28: Sample Assistant Running

Then we can communicate with google assistant with the hot word as like as “hey google” or “ok google”. The voice is processing in this way:

```
INFO:root:Recording audio request.
INFO:root:End of audio request detected
INFO:root:Transcript of user request: "turn on".
INFO:root:Playing assistant response.
INFO:root:Turning device on
INFO:root:Waiting for device executions to complete.
INFO:root:Finished playing assistant response.
```

Using this assistant, we want to on off our light, fan & other devices so that we have to download some python library again for interact Raspberry Pi with Google API [15].

```
git clone https://github.com/googlesamples/assistant-sdk-python
```

This Github package contains the python code for `action.device.commands.OnOff` trait which can on off our led. After cloning the packages, we have to open the directory:

```
cd assistant-sdk-python/google-assistant-
sdk/googlesamples/assistant/library
sudo vim hotword.py
```

```

print(event)

if (event.type == EventType.ON_CONVERSATION_TURN_FINISHED and
    event.args and not event.args['with_follow_on_turn']):
    print()
if event.type == EventType.ON_DEVICE_ACTION:
    for command, params in event.actions:
        print('Do command', command, 'with params', str(params))
        if command == "action.devices.commands.OnOff":
            if params['on']:
                # if params['on'].get('name') == "bedroom":
                print('Turning the LED on.')
                GPIO.output(7,1)
            else:
                print('Turning the LED off.')
                GPIO.output(7,0)

# if command == "action.devices.commands.OnOff":
#     if params['on']:
#         print('Turning the Fan on.')
#         GPIO.output(5,1)
#     else:
#         print('Turning the Fan off.')
#         GPIO.output(5,0)
if command == "com.example.commands.BlinkLight":
    number = int( params['number'] )
    for i in range(int(number)):
        print('Device is blinking.')
        GPIO.output(5, 1)
        time.sleep(1)
        GPIO.output(5, 0)
        time.sleep(1)

```

Figure 29: Code for Controlling GPIO

In this code we have modified the import section for controlling GPIO port for on off command executing.

```
import RPi.GPIO as GPIO
```

Then we modified the code to set output pin initially to the low logic state. When the device got on command then set the pin to logic high state and when off command is received it set as low.

```

device_handler = device_helpers.DeviceRequestHandler(device_id)
GPIO.setmode(GPIO.BCM)
GPIO.setup(25, GPIO.OUT, initial=GPIO.LOW)

```

```
@device_handler.command('action.devices.commands.OnOff')
def onoff(on):
    if on:
        logging.info('Turning light on')
        GPIO.output(7, 1)
    else:
        logging.info('Turning light off')
        GPIO.output(7, 0)
```

After modifying the code, we run it in python virtual environment with this command:

```
python pushtotalk.py
```

Now when we use the command “Ok Google turn on the light” then it turns on the led which we are connected in breadboard for test purpose. In this procedure we add some more traits for on off another devices.

Chapter 4

Working Methodology

4.1 System Architecture

System Architecture

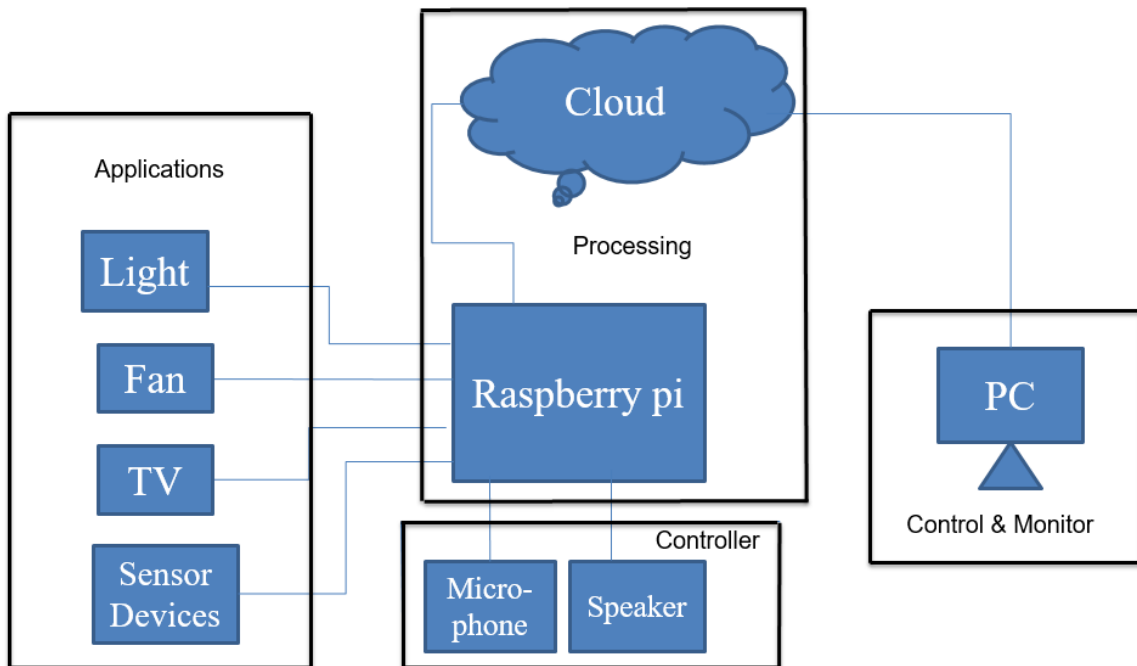


Figure 30: System Architecture

In this project we use raspberry pi 3 which works in processing section. Microphone and speaker works in controller and light, fan, tv and different types of sensor devices are work in application section. For controlling the total section we use a monitor.

In microphone, it detects our voice command and send it to the Raspberry pi. According to our voice command it prepare the total system for executing. When the microphone get the command lights on, it sends it to the raspberry pi and than execute it by turning on the light.

The brief description is given above.

Here Raspberry pi works as google assistant. When we give a voice command to the microphone for on-off the light, it sends it to the microphone and then microphone generate a token to get the access in cloud. After getting the access on cloud , it gives us two types of output.

First output for by turning on-off the light and another one is voice confirmation.

4.2 Overview of working procedure

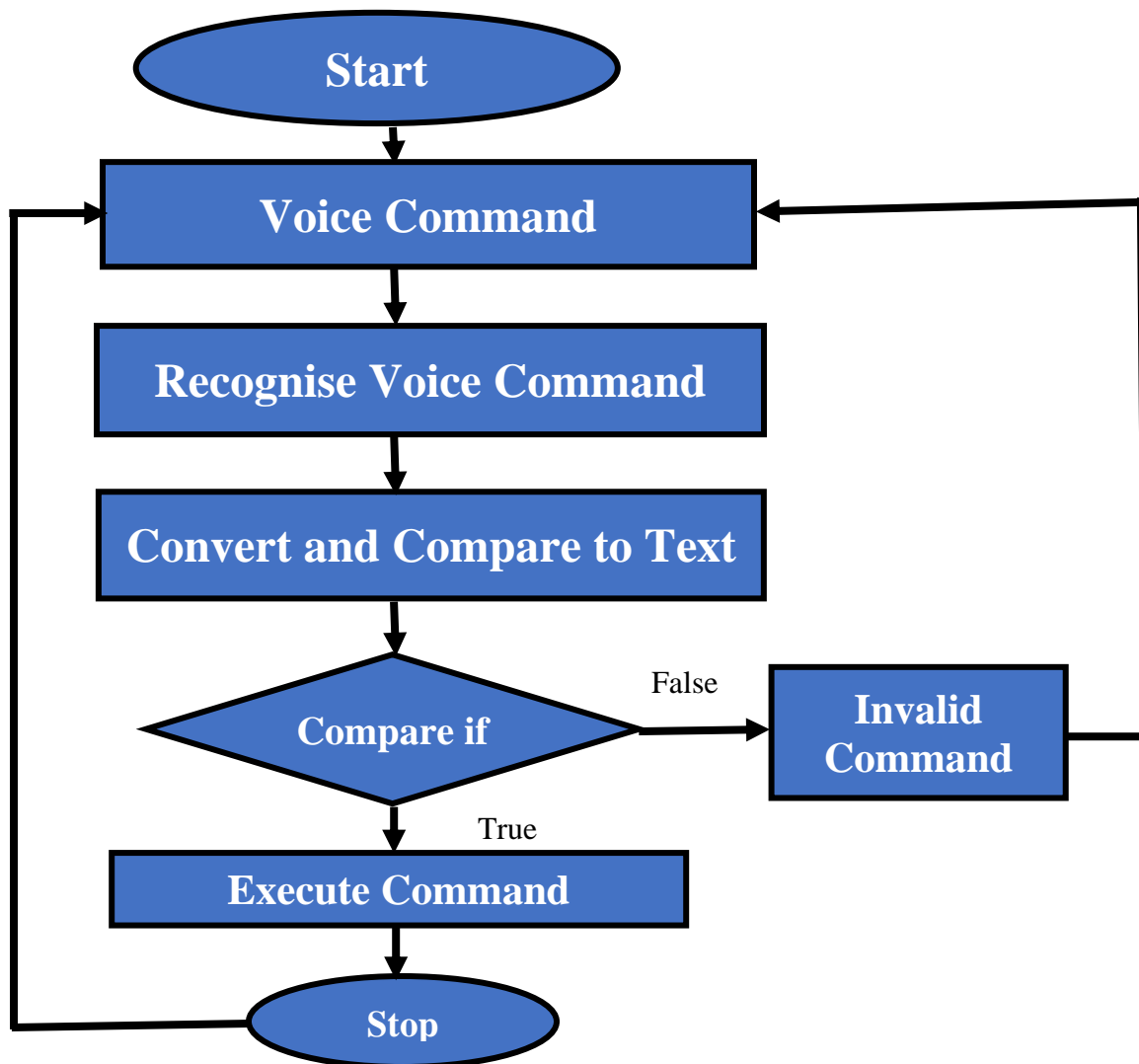


Figure 31: Flow Chart

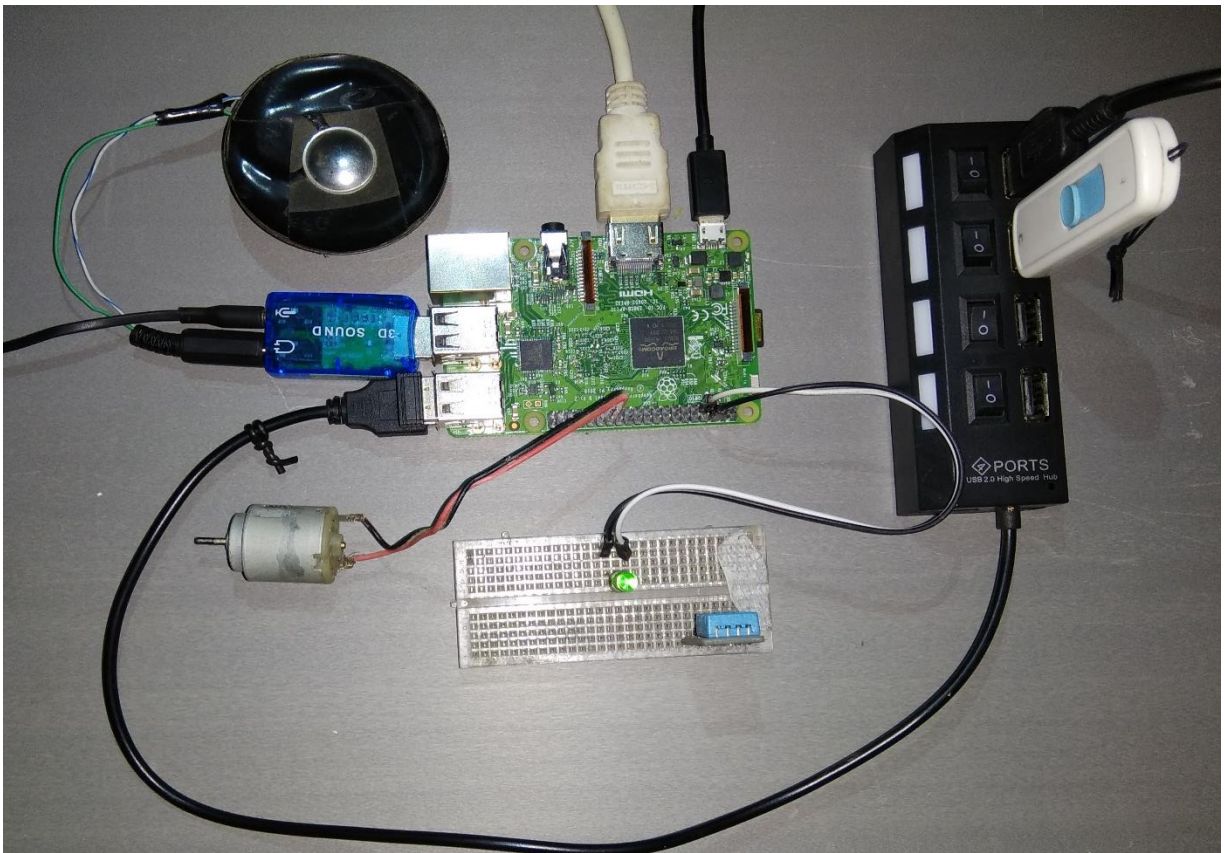


Figure 32: Implimentation of voice control home automation

Finally from the above figure we found that sound card is attached with raspberry pi to connect the speaker and microphone. Then run the Google assistant sample SDK and device is ready to communicate as a Google assistant client. After completing the running process we can communicate with Goole assistant through the syntax as 'Ok google' or 'Hey google'. Then the device is prepared to take the voice command. After that controlling the home appliances by running the 'hotword.pi' python file.

Example : When user says, 'Ok google turn on light' , assistant give a output by turning on the light.

Chapter 5

Conclusion

The intention is to control a portion of the significant family unit gadgets by voice. It isn't just gone for giving a sound what's the more, agreeable way of life to the clients, yet additionally at helping the debilitated or disabled and individuals living alone, with the goal that they can effectively handle every one of their undertakings at an accommodation. Making the plan sleeker and simpler to deal with, with a technique to control more machines at once are the future prerequisite. We have so far accomplished in controlling a similar apparatus, model, a light, in the meantime in two unique rooms, and we have possessed the capacity to control bigger burdens, for instance, a forced air system. The plan of the voice controlled home computerization framework is both convenient and prepared to be introduced in your primary family unit circuit. We have presented versatility in the whole framework, both the voice controlling remote and the application end.

Future Improvement

The future would be of cutting edge sensors and man-made brainpower.

Till date home mechanization has been viewed as something to just on/off your gadgets from cell phone, yet the situation be evolving quickly, customary computerization joined with keen remote system of sensors began to comprehend the human condition. Sensors like movements sensor, temperature sensors, light power sensors, smoke indicators, moistness sensors and so forth are being suited with home computerization and the information from these sensors would additionally be utilized to examined to comprehend use designs and thus with profound learning utilized for the knowledge of the gadget itself.

It will begin understanding its client and continue ensuring the client is at his most straightforwardness and security.

Smoke indicator, air quality sensor, LPG locator

Remote Motion sensor. Wireless Light power sensor

Your future home!!!!

References

- [1] E. Bogia, "The Internet of Things vision: key features, applications and open issues," *Computer Communication*, vol. 54, no. 2018, pp. 1-31, 2014.
- [2] R. Cellan-Jones, "A15\$ computer to inspire young programmers," *BBC News*, pp. 1-15, 5 May 2011.
- [3] Raspberry Pi Foundation, "RASPBERRY PI 3: SPECES, BENCHMARKS & TESTING," Raspberry pi Org, 2016. [Online]. Available: <https://www.raspberrypi.org/magpi/raspberrypi-3-specs-benchmarks/>.
- [4] RASPBERRY PI FOUNDATION, "Documentation Usage GPIO," RASPBERRY PI FOUNDATION UK, [Online]. Available: <https://www.raspberrypi.org/documentation/usage/gpio/>. [Accessed 12 December 2018].
- [5] F. L. J. Margaret Rouse, "I2C bus (Inter-IC bus)," September 2005. [Online]. Available: <https://whatis.techtarget.com/definition/I2C-bus-Inter-IC-bus>. [Accessed 12 December 2018].
- [6] StackExchange, "What is the difference between board and bcm for GPIO pin numbering ?," Raspberry Pi, 14 January 2014. [Online]. Available: <https://raspberrypi.stackexchange.com/questions/12966/what-is-the-difference-between-board-and-bcm-for-gpio-pin-numbering>. [Accessed 12 December 2018].
- [7] L. Tyler, "A BRIEF HISTORY OF SOUND CARDS," Digital Trends, 3 March 2018. [Online]. Available: <https://www.digitaltrends.com/computing/what-is-a-sound-card/>. [Accessed 15 Dec 2018].
- [8] C.Woodford, "explainthatstuff," 7 February 2018. [Online]. Available: <https://www.explainthatstuff.com/microphones.html>. [Accessed 16 December 2018].
- [9] "physics.org," [Online]. Available: <http://www.physics.org/article-questions.asp?id=54>. [Accessed 16 December 2018].
- [10] adafruit. [Online]. Available: <https://www.adafruit.com/product/65>. [Accessed 19 12 2018].
- [11] "Qutron," [Online]. Available: <http://www.qutron.dk/content/how-led-works>. [Accessed 16 December 2018].

- [12] zazzle, "LED Light Emitting Diode Schematic Poster," [Online]. Available: https://www.zazzle.com/led_light_emitting_diode_schematic_poster-228994509736180516. [Accessed 19 12 2018].
- [13] Google, "Google Assistant SDK for devices," Google , 2 October 2018. [Online]. Available: <https://developers.google.com/assistant/sdk/guides/library/python/embed/config-dev-project-and-account>. [Accessed 12 11 2018].
- [14] Google Inc., "Google identity platform," Google, 12 November 2018. [Online]. Available: <https://developers.google.com/identity/protocols/OAuth2>. [Accessed 17 December 2018].
- [15] Google inc, "Google Assistant SDK for devices," Google, 12 July 2018. [Online]. Available: <https://developers.google.com/assistant/sdk/guides/service/python/extend/handle-device-commands>. [Accessed 19 Dec 2018].
- [16] D.B.B.a.Bcm, "StackExchange,"[Online].Available: <https://raspberrypi.stackexchange.com/questions/12966/what-is-the-difference-between-board-and-bcm-for-gpio-pin-numbering>.
- [17] K. D. Foote, "A Brief History of the Internet of Things," DATAVAESITY, 16 August 2016. [Online]. Available: <http://www.dataversity.net/brief-history-internet-things>. [Accessed 12 December 2018].