Smart Mirror (A Interactive Voice Controlled Mirror That Maximize Time Efficiency And helping the incapable people)

BY

MD.Mostak Reza Chowdhury
ID: 142-19-1577
Mehedi Hasan
ID: 142-19-1560
And
Pranta Mojumder
ID: 142-19-1588

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

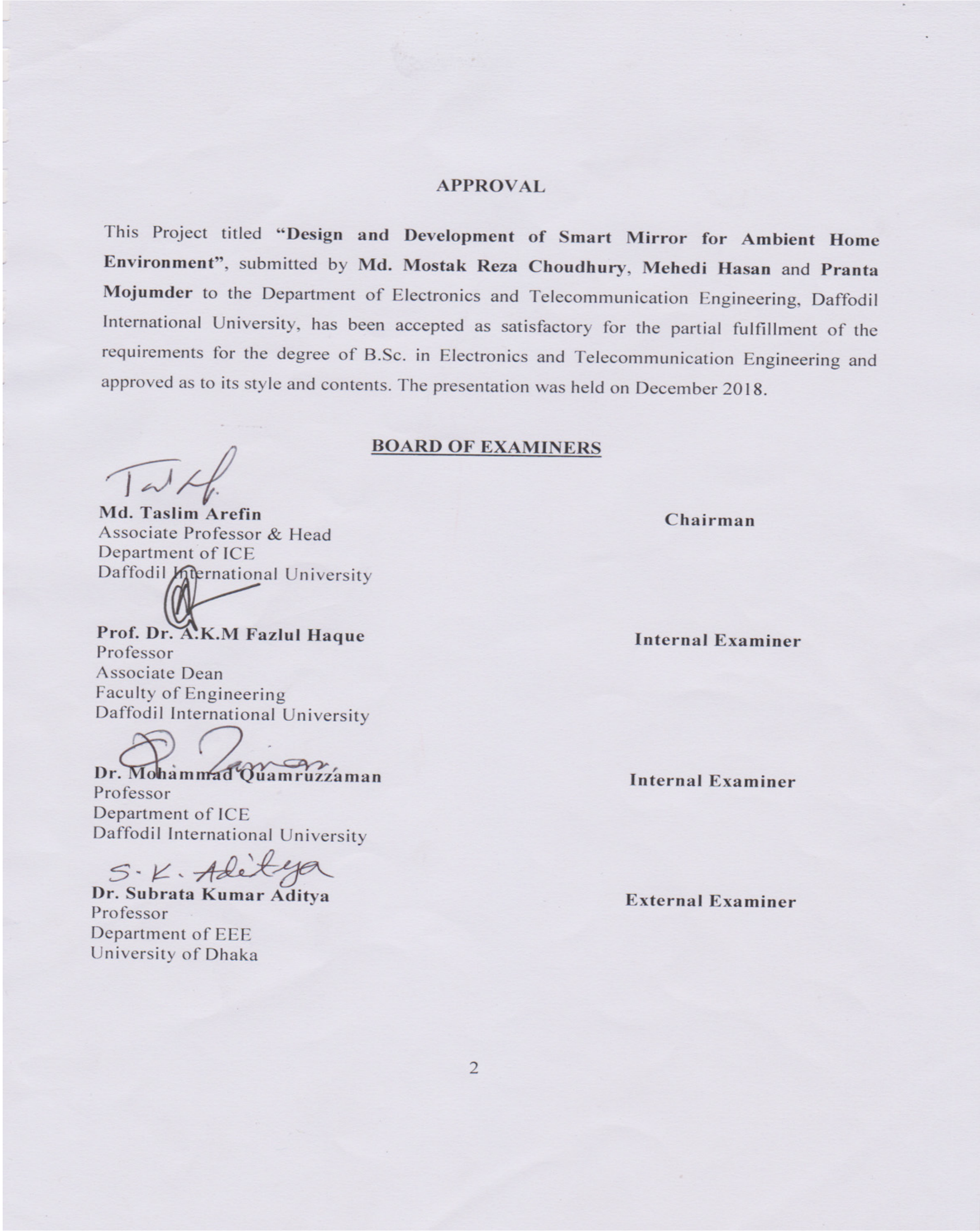
Supervised By

Md. Zahirul Islam
Lecturer
Department of ETE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DEC 2018



APPROVAL

This Project titled **"Design and Development of Smart Mirror for Ambient Home Environment"**, submitted by **Md. Mostak Reza Choudhury**, **Mehedi Hasan** and **Pranta Mojumder** to the Department of Electronics and Telecommunication 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 December 2018.

BOARD OF EXAMINERS

Md. Taslim Arefin Associate Professor & Head Department of ICE Daffodil International University	**Chairman**
Prof. Dr. A.K.M Fazlul Haque Professor Associate Dean Faculty of Engineering Daffodil International University	**Internal Examiner**
Dr. Mohammad Quamruzzaman Professor Department of ICE Daffodil International University	**Internal Examiner**
S. K. Aditya **Dr. Subrata Kumar Aditya** Professor Department of EEE University of Dhaka	**External Examiner**

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Md.Zahirul Islam, Lecturer, Department of ETE** 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:

____

Md. Zahirul Islam
Lecturer
Department of ETE
Daffodil International University

Submitted by:

____

(Md.Mostak Reza Chowdhury)
ID: 142-191-1577
Department of ETE
Daffodil International University

____

(**Pranta Mojumder)**
ID: 142-19-1588
Department of ETE
Daffodil International University

____

Mehedi Hasan
ID: 142-191-1560
Department of ETE
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 feel grateful to and wish our profound our indebtedness to **Supervisor Md. Zahirul Islam**, **Lecturer**, Department of ETE 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 draft 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 ETE**, for his kind help to finish our project and also to other faculty member and the staff of ETE 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

ABSTRACT

This project is on “**Smart Mirror**”. Digital assistant is a artificial intelligent that is a small handy gadget and works through web user interface with a wireless internet medium.Smart mirror can help the paralyzed and the blind people which cant able to handle the daily routine in their need. A incapable people can easily interact with this device through voice and can get information aroundrom the world.

It requires Raspberry pi hardware and digital sound card using speaker and microphone and a display. The voice is processed to a digital signal instead of conventional analog methods. Smart Mirror AI (Artificial Intelligence) is a gadget that is loaded with Weather forecast,News, Translator, Work Reminder, Personal Doctor, Instant Pleasant Music, Web Assistant, Social media notified and actually as an AI friend. Since Digital AI system is software and hardware based, you can make your life easier.

The aim of the smart mirror is a python software that integrates the best tools available for Raspberry pi based smart mirror, easy-to-use interface. However, this project the multi-model dialogue system are used which process combined input mode, such as image, voice video, touch and manual gesture in order to design a mirror. The model of virtual private assistant is increasing the interaction between human and machine by using different methodologies such as voice recognition, image recognition, sound recognition We concentrate on intellect of natural language and dialog management perspective as we believe that these parts of the technology pipeline require the biggest amount of augmentation .

Table of Contents

LIST OF FIGURES

FIGURES **PAGE**

Chapter 2

Chapter 1

Introduction

1.1 Introduction:

Intelligent personal assistant opened up a new world. Where we can communicate with machine via our natural voice. Human request to perform a task to this device for example" Hey,What is my reminders today" before our working day."What's my meeting time" .In the additional feature like traffic update system "What is the traffic update between Dhanmondi to Mohammadpur". It can create a good understanding between human and machine.

Smart Mirror usually based on spoken dialogue system.Spoken dialogue system is intelligent spy that helps user more capable via voice interaction .It can implement in various devices such as smart phone, smart TV,vehicle navigation etc. Smart Mirror has a wide range of application in business enterprises, Education, healthcare and Entertainment .Many company uses different technique to reduce the complexity of Smart Mirror. For example; Google use DNN Deep Neural Networks to improve the Google Mirror .It highlights the main component of voice recognition and the deep learning architecture.

A small single-hardware device called raspberry pi is introduced in this project. Python is a software implementation of a Raspberry pi .A third generation raspberry pi uses a Broadcom BCM2837B0 SoC with 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor. And a build in GPU OpenGL ES 2.0 WITH 1080p H.264/MPEG-4 AVC high-profile decoder and encoder. It has 1 GB memory shared with the GPU.

Raspberry pi operates with a Debian operating system named Raspbrian.Basically, raspbrian use PIXEL (Improved windows Environment, Lightweight) to create a desktop environment. It modified with composed LXDE (Lightweight X11 Desktop Environment) and open box windows manager.

In this project, we use Google SDK assistant in an appliance web server that integrates the best tools available for Debian based Raspberry pi easy-to-use interface. It also adds its own set of utilities and allows Google SDK with best software package available for open source communication with machine.

The target of the project is to minimize the complexity of human to do daily task.

Like,

- Saving time
- Virtualization support. You can run machines through the voice command.
- Really friendly Web user interface.
- Always get update notification of social media.
- Helping incapable person.

1.2 Goals of the Project:

- Smart Mirror device can implemented in low cost, and a single hardware device arranged all the facilities on it.

- Smart Mirror has made interacting a simpler job with the introduction of Raspberry pi based virtual assistant.

- A blind man can easily understand the machine sound to make his life easier.

- A paralysed personj can see the open world update through the smart mirror.

- User can get recent update throughout the internet using a single command.

- Delivers highest automation possible.

- Smart Mirror can keep your personal life clutter free since we don’t require wiring components for installation.

- Voice based input and output system may be the easier way to understand all users.

- Every user of all stages acquire basic education by using it, also the children’s.

- It advanced communication system for This Project, it is better to opt for Smart mirror rather than web browsing.

1.3 Organization:

This Project report has six chapters in total. The first chapter describes a brief idea about AI (Artificial Intelligence), virtual assistant, Raspberry pi, Raspbrian, python and Goal of the Project. The second chapter contains theory of Smart Mirror and its modules. In third chapter Installation of Smart Mirror. Fourth chapter explains the Configuration of Raspberry pi through python, chapter five explain Implementation and testing. Finally, chapter six gives the concluding remarks and suggestions for the future works.

Chapter 2

Implementation of Smart Mirror as a virtual assistant

2.1 Smart Mirror:

A smart mirror is a two way mirror with a display behind the glass. Viewer can see the different information on the display of the smart mirror, Like weather forecast, News, Calendars ,Personal assistant etc .This device can easily perform multitasking method as a same time.

Fig 2.1 User Interface of Smart mirror

We design this project to able to helping the incapable people like blind, paralyzed and basically the physically disabled people with their daily routine . Smart mirrors are a latest addition for a smart family that has been getting a lots of positive review in recent years by both commercial sector and lifestyle. It explores enhancing such mirrors with artificial intelligence. It targets is to develop a user voice recognition system that can differentiate between the incapable person and device by using a voice with a two-way mirror. The Raspberry Pi 3 B is used as the embedded computational device for the intelligent mirror.

2.2 Smart Mirror Architecture

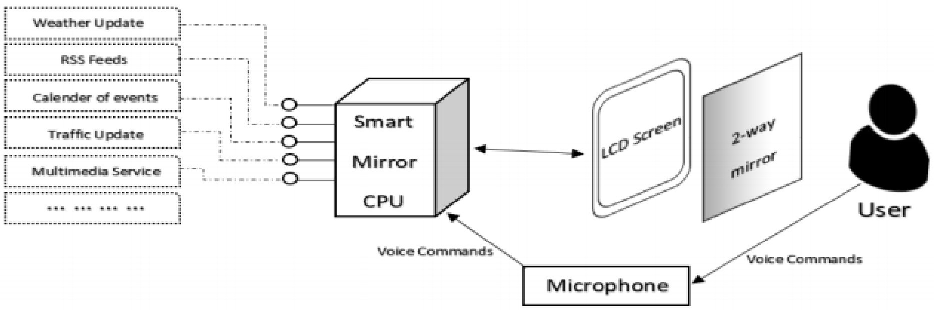


Fig : 2.2 Smart mirror Architecture

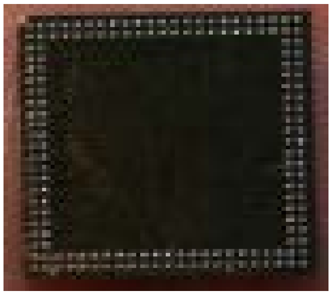
The interface of the mirror is designed with several widgets. A widget is a single window frame that contains an embedded browser. Widgets do not overlap and do not contain any complex interface elements. The smart mirror interface contains two types of widgets, one enables the remote device control (e.g. light turn on/off, and home automation control), and the other widzet enables access to use various information services (e.g. news, weather forecast, traffic etc.).

The Raspberry Pi is configured to using via voice command and do the response of multitask with the users. The Mirror is based on the Raspberry Pi and the output is seen on a monitor that's connected to the raspberry pi .This design of devices implements the Mirror and can be used in work, house, or public environments.

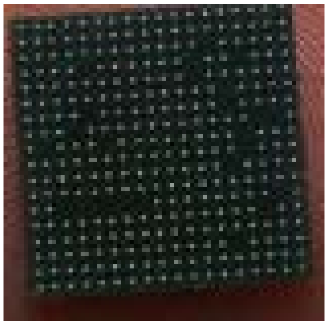
2.3 Raspberry pi:

A complex IC is integrated into a single chip, it can operates a programmable processor accelerating hardware functions. Both hardware and software are combined with the devices. Broadcom

BCM2835 SoC Multimedia processor used CPU model ARM 1176JZF-S (armv6k) 700MHz.



Samsung K4P2G324ED Mobile DRAM



BCM2835 SoC

A Broadcom Video IV is used in the raspberry pi with the memory of 1GB shared

2.2 Raspberry pi Architecture

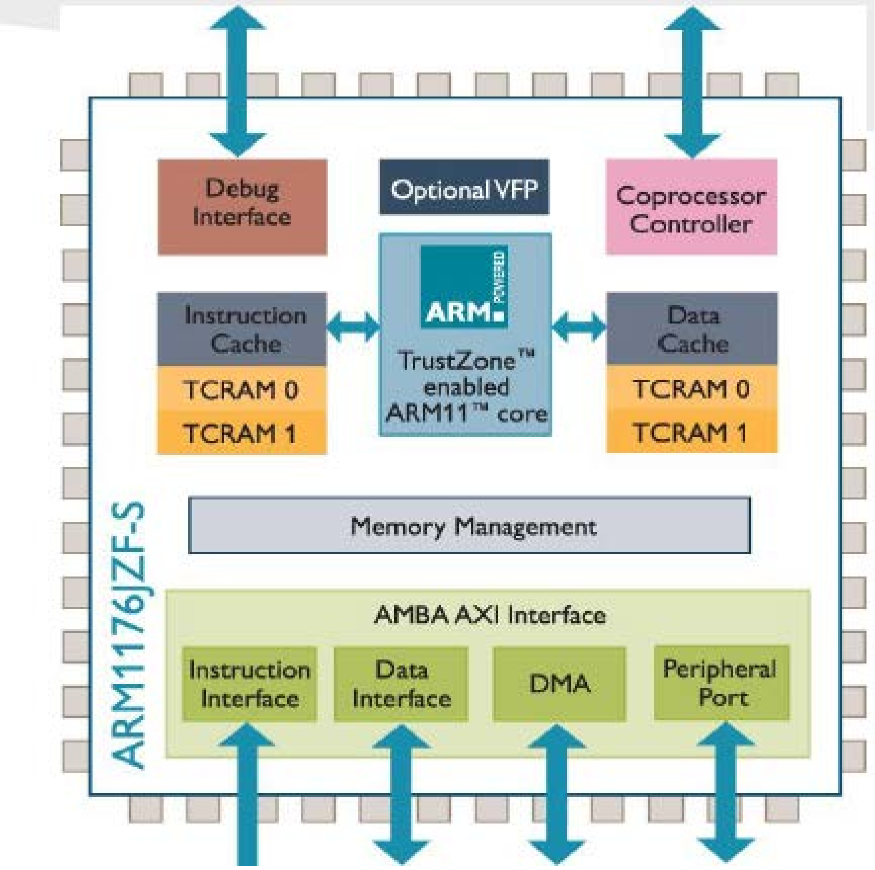


Figure 2.2 - Raspberry pi Architecture

The figure above shows the basic Raspberry pi Architecture concepts related to the architecture like interface and applications.

2.2.1BCM2835: CPU

BCM2835 is a 32 bit RISE (reduced instruction set computer) system. It has 700 MHZ clock rate and 8 pipeline stages. The CPU perform with ARM11J6JZF-S the family of ARM-11.

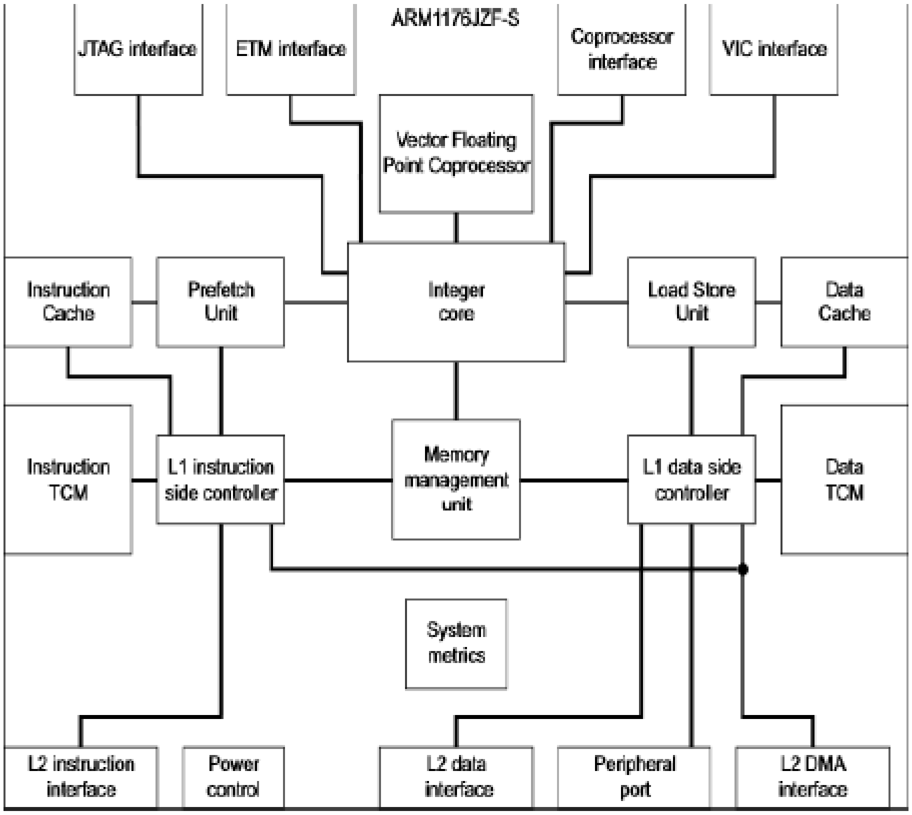


Fig.2.2.1 ARM1176JZF-S Technical Reference Manual

2.3 Google Assistant SDK:

Google assistant SDK is a key solution for anyone who want to integrate assistant into prototype device. TheGoogle assistant library is written in python and it supported a device with liunux-armv7 and linux-x86_64 architectures. The library reveal with high level, the API that is based on event is easily augmented. It provide the following features such as

- Hands-free activation - To activate the device, we need to say “Hey Google or Ok Google.
- Capture the audio and playback
- Conversation state management

The supported language for Google SDK is pythonbased. Google has introduced the API design on public.

Google API is a subroutine communication protocol and tools for building web software. Thegoal of making the API is simply to use and consistent API networks. Google use the REST (Representational state transfer) style for the architecture. It is worked well with HTTP/1.1.The REST refers to use individually addressable resources. The special method for REST API is *List, Get, Create, Update,* and *Delete.*

The design flow of the API is

- Determine, what types of resources API can provides.
- Find the relation between resources.
- Determine the name scheme of the resources with the relation of the resources.

Add the minimum set of methodology to the resources.

2.4 Speech-to-Text Conversion (STT)

- STT is the process of converting the voice sound to text format. It also called speechrecognition. All stages is required at least two model of STT
- Acoustic model
- language model

To use a large system model can use pronunciationmodel. To acquire a best transcription merit, all the model can be summarized with dialect, communicationchannel, type of speech.The other pattern of recognition technology, there generates an error.

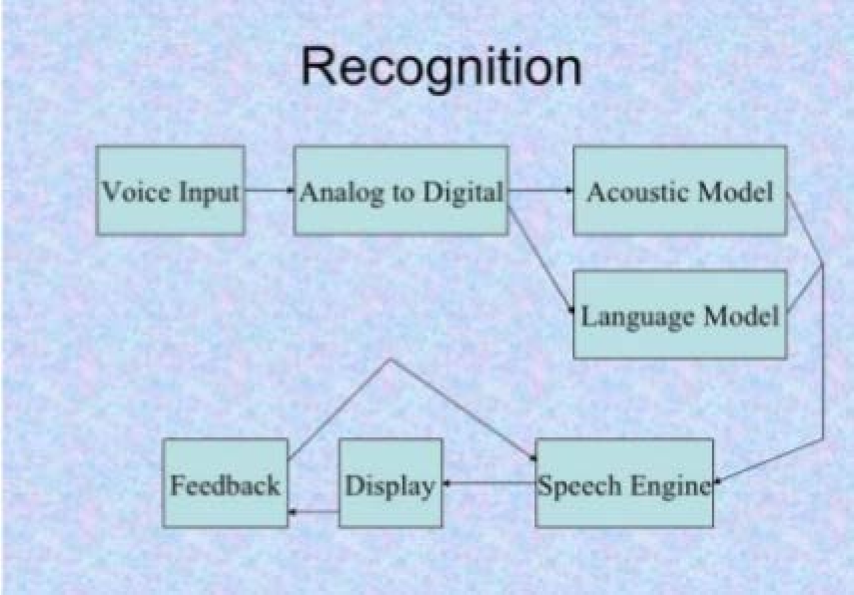


Fig.2.4 Block diagram of STT

2.4.1 Acoustic Model

Acoustic model invented to recording an audio file of speech and it is convert into a text transcription. It identify the software that make statistical representation of sound and make each word readable.

2.4.2 Language Model

Language model based on the file containing recognition probability of sequences of word. This model usually used to motivation the application where the grammar are used to desktop command.

2.5Text-to-SpeechConversion (TTS)

Raspberry pi multimedia capable to host a text to speech audio broadcastservice. Data stream broadcast the service as peer to peer application. It has a requester which sends a authentication request to broadcast the audio and then a broadcast application start into the pi.

Automatic speech recognition (ASR) transforms speech into text. The theory based on speech coding theory, simulatingrecent spectral analyses accomplished by ear. Voice recognition is the automatic source of ananalog voicewaveform. Text to Sound converter anticipate use of database of information, both text and speech. Previously resolvedvoice is stored in small units in the database, for focus in the proper discipline at runtime. Text to sound conversionsystem perform text processing, "letter-to-sound" conversion, generate the phonetic transcription, first. The output of the database containing words or sentences, which act with high quality performance. Besides, a synthesizer can also organize human voice quality to create a synthetic but any natural sound.

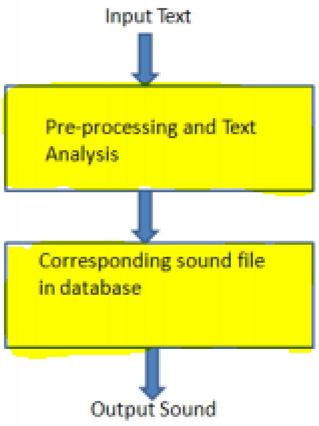
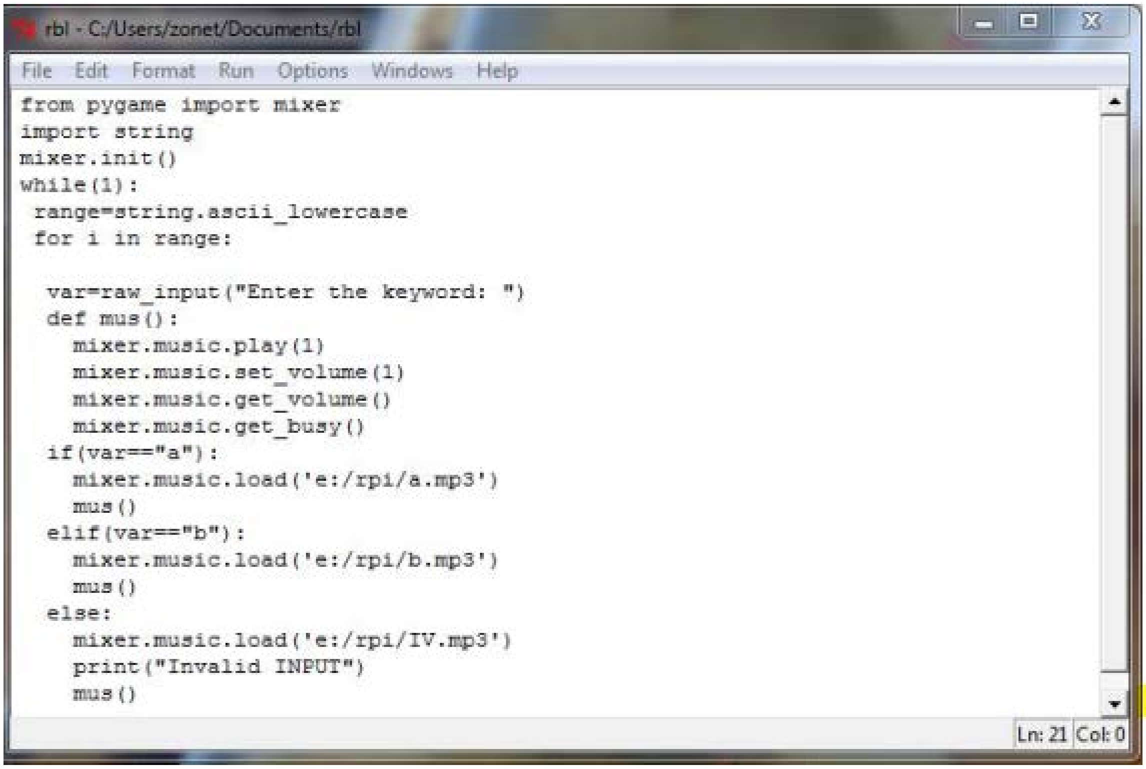

Input Text
Pre-processing and Text Analysis
Corresponding sound file in database
Output Sound

Fig.2.5 Flow chart of TTS



```
from pygame import mixer
import string
mixer.init()
while(1):
 range=string.ascii_lowercase
 for i in range:

  var=raw_input("Enter the keyword: ")
  def mus():
    mixer.music.play(1)
    mixer.music.set_volume(1)
    mixer.music.get_volume()
    mixer.music.get_busy()
  if(var=="a"):
    mixer.music.load('e:/rpi/a.mp3')
    mus()
  elif(var=="b"):
    mixer.music.load('e:/rpi/b.mp3')
    mus()
  else:
    mixer.music.load('e:/rpi/IV.mp3')
    print("Invalid INPUT")
    mus()
```

Fig.2.5b Pseudo code using python

The mentioned Text to speech (TTS) system schedule in order to generate a acoustic signal which is synchronized text provided in an input on Raspberry pi.The strategy of TTS is to involve with GPIO lines, audio amplification and the database connectivity.

2.6 ASR Model

ASR model works in real time of speech recognition with the microphone in cloud server to recognize the voice that a user speaks in microphone and convert it into text. Then it sends the data in the cloud server to analyzed the text and showing the result.

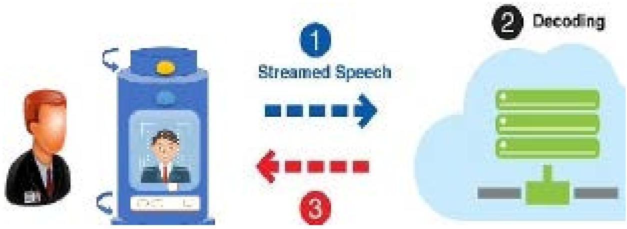


Fig.2.6 the ASR model

2.7 3.2. Improved Context Awareness

Moreover which is directly requested by the user, a spacious amount of information is available which may be sends by a system and in the end could server improve its context Awareness. For example, sensors studded in room, in the office or every day used appliances such as our smart-phones, are great data sources that could be used to increase human-machine interaction.

Internet is a potential data source that could be search in order to provide an exact knowledge of the “outside world”.

Chapter 3

Hardware Equipment’s

3.1 In this project we use-

- Raspberry Pi 3 Model B
- LCD Display
- USB Sound Card
- 5 V Audio Amplifier
- Microphone
- 3 w Speaker
- LED (generic)
- 1 k ohm Resistor
- 5 V Po
- Wood I

Story:

- **Raspbe**

  Raspbe from the
  fruit na 's cheapest
  and sma inux,
  Ubuntu oding etc.
  When w audio
  jack, m D card
  slot,40 al purpose.
  Raspbe sor.

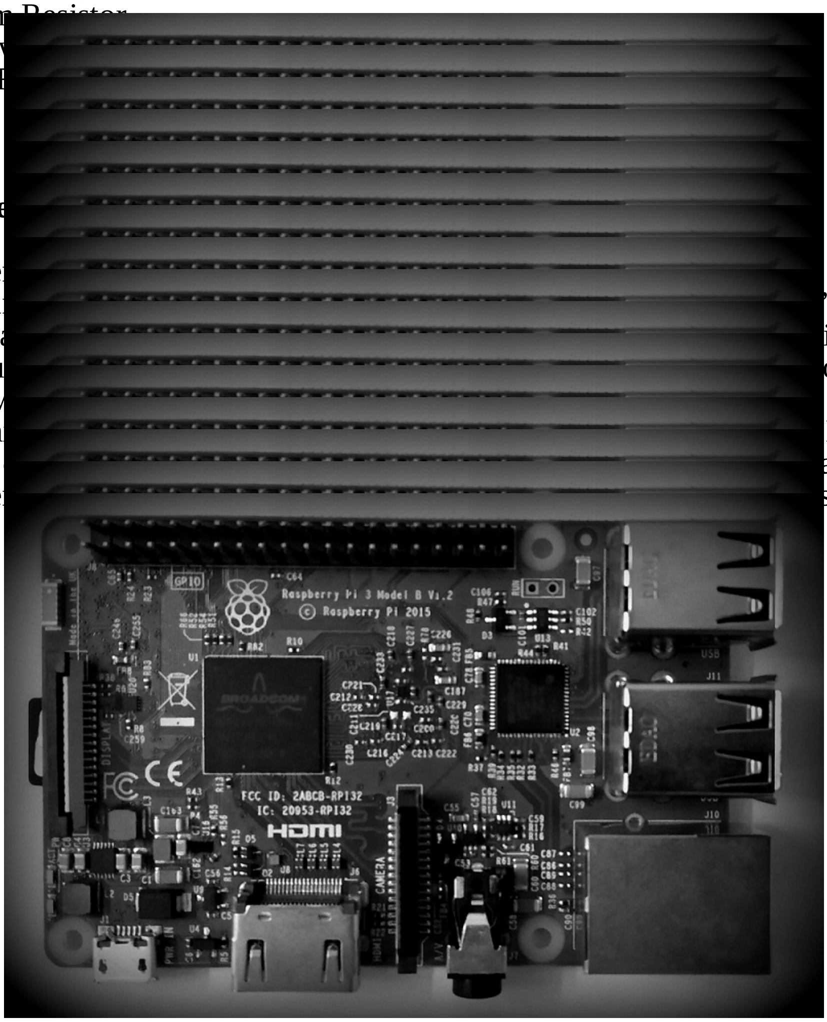


Figure: 3.1Raspberry Pi 3 Model B

- **USB Sound Card**

  We use an USB external sound card for the output audio and input microphone voice also. All USB sound card has a 2.5mm audio input jack and a 2.5 audio output jack. So we can easily solve the sound problem by using an USB sound card.

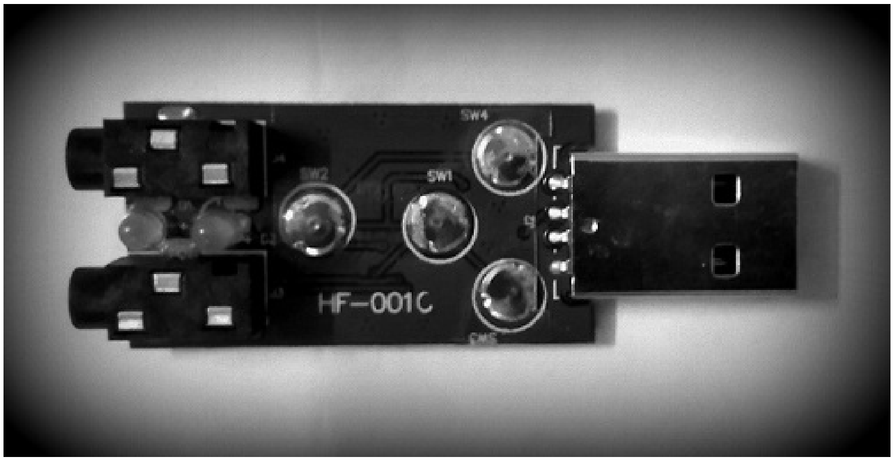


Figure3.1b: USB sound Card

- **5 V Audio Amplifier**

  We use a 5v audio amplifier for increasing our speaker sound. In this circuit we can see that there is a PAM8403 transistor for boost up our 3w speaker.

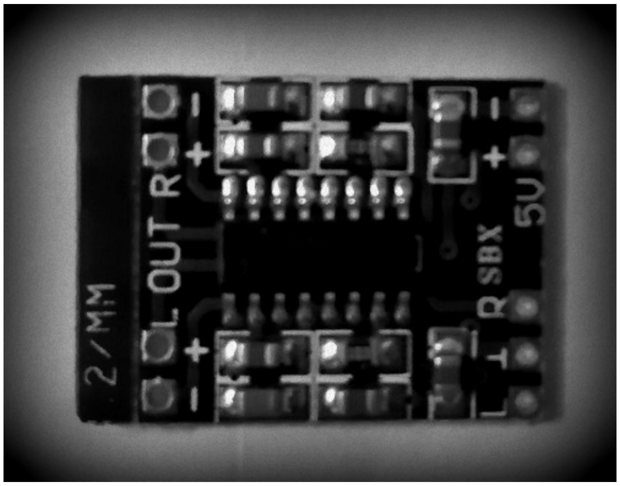


Figure3.1c: 5v Audio Amplifier

- **Microphone**

  We make a 2.5 mm microphone from the old headphone for input our voice into the circuit through USB sound card.

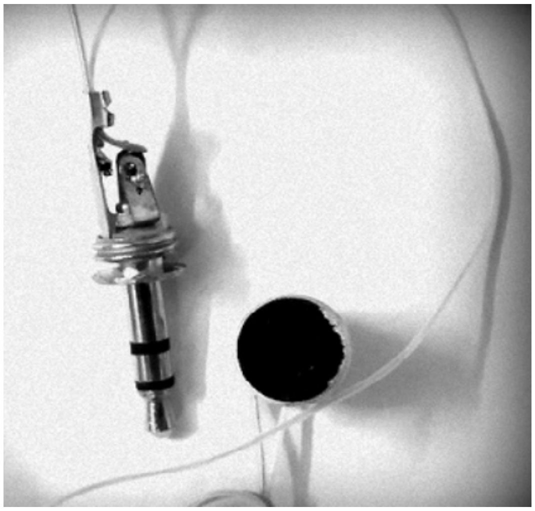


Figure3.1d: 2.5 mm Microphone

- **3 watt Speaker**

  In this project we use 2 pcs 3 watt speaker for output sound. We connect the speakers with the audio amplifier and also connect with the USB sound card.

Figure3.1 e: 3 Watt Speakers

- In the end of this project we use 5v adaptor for power supply and some LEDs for decoration.

Chapter4

Installation of Raspberry pi

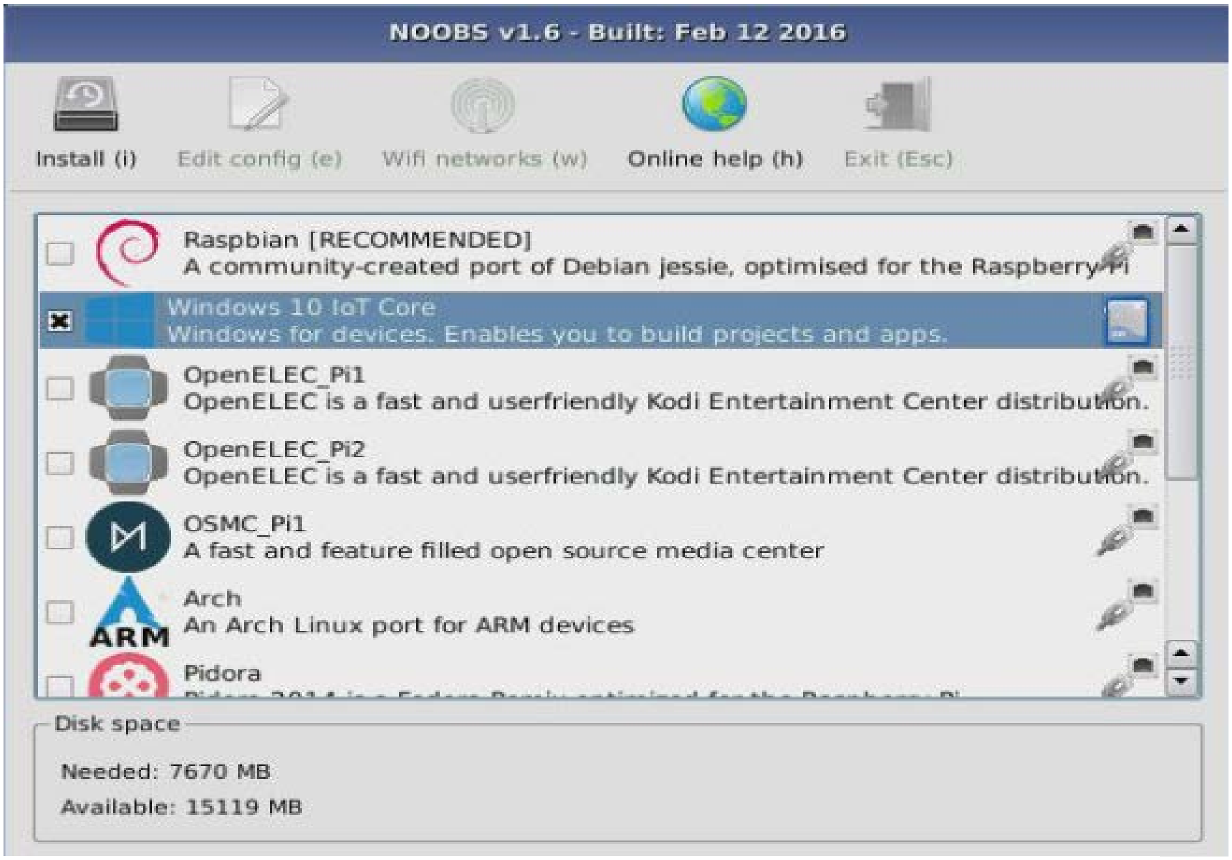
4.1Operating System

Raspberry pi usually supported withLinux based Debian operating system. After install the operating system raspberry pi works like a mini computer. Raspbrian operating system give the LXDE desktop environment to the user.

Basically, raspberry pi has no monitor build-in.For configure the circuit we should have a head device. A HDMI port given in the motherboard by birth. The HDMI port can directly connect to a monitor or a TV for configure the circuit.

4.2 Installation

For, installing the operating system a booted SD card is required to open it.A Ethernetcable, a mouse and a keyboard need to be adjust with the motherboard. A composed file is need to be booted in the SD card .When the monitor shows a windows ,that want to download the raspbrian can automatically download via the LAN networks. Raspberry pi 3 has a additional feature like wireless and Bluetooth connectivity. We can use the Wi-Fi connectivity alternative with Ethernet cable.



□ Raspbian [RECOMMENDED]
A community-created port of Debian jessie, optimised for the Raspberry Pi

☒ Windows 10 IoT Core
Windows for devices. Enables you to build projects and apps.

□ OpenELEC_Pi1
OpenELEC is a fast and userfriendly Kodi Entertainment Center distribution.

□ OpenELEC_Pi2
OpenELEC is a fast and userfriendly Kodi Entertainment Center distribution.

□ OSMC_Pi1
A fast and feature filled open source media center

□ Arch
An Arch Linux port for ARM devices

□ Pidora

Disk space

Needed: 7670 MB

Available: 15119 MB

Fig 4.2 Noobs windows in raspberry pi

Raspberian looks like the similar operating system that we used to. A highly customized operating system that can be tweaked enough to run in the pi.It works extremely well like as a normal computer.

To build a personal assistant we need a input and output devices. Raspberry pi do not understand its interfaces own.So the motivation to give the exact command to the board is necessary.

4.2.1 ALSA Sound system

Raspberry supports ALSA sound system. ALSA meant the device that is *nix based. ALSA is advanced Linux audio architecture. It provides the functionalities with the connect of Linux -kernel. ALSA replaced by the Original Open Sound System (OSSv3) providing drivers to a sound card.ALSA makes a user space storage for application developers, who want to maintain drivers with high level API.

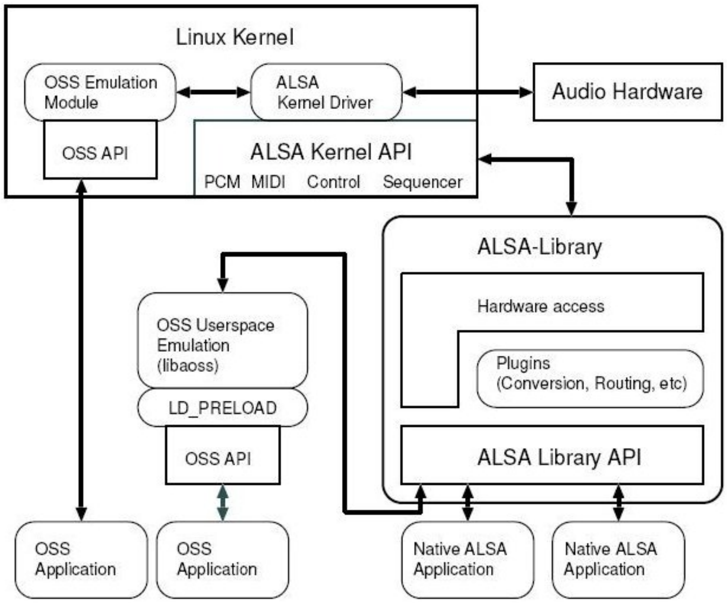
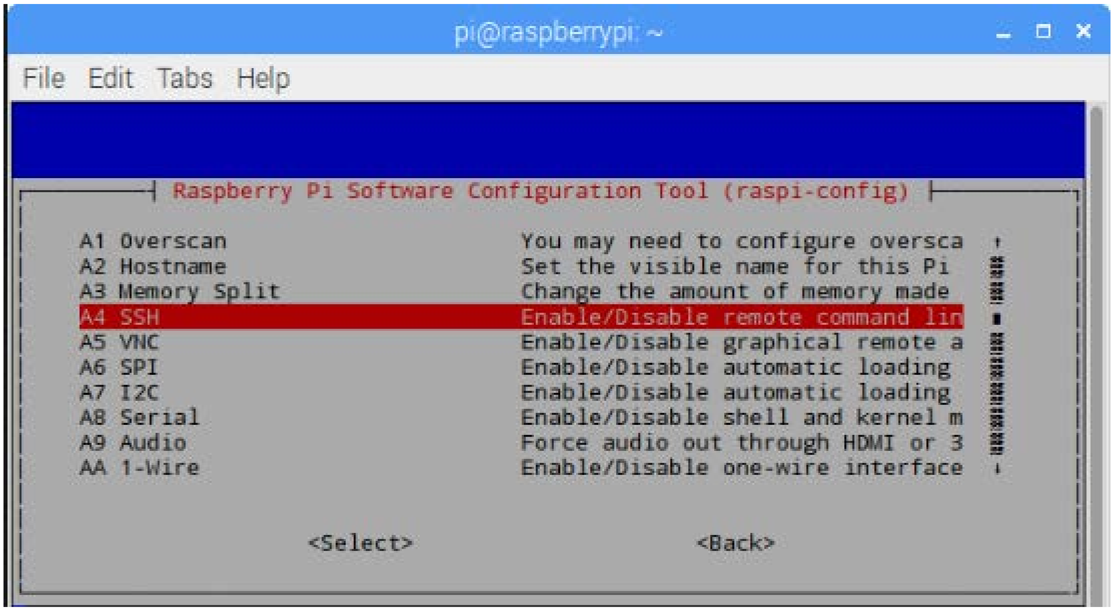


Fig.4.2.1 Flowchart of ALSA system

It needs to indicate the sound card on the selected interface using the command prompt.

4.2.2 Creating a VNC server

Virtual Network Computing is a linear desktop sharing system. If raspberry pi is headless we need to install VNC server .the SSH must be enable to gain access to the command line of the raspberry pi from another computer.



```
┌──────────┤ Raspberry Pi Software Configuration Tool (raspi-config) ├──────────┐
│                                                                               │
│    A1 Overscan                      You may need to configure oversca         │
│    A2 Hostname                      Set the visible name for this Pi          │
│    A3 Memory Split                  Change the amount of memory made          │
│    A4 SSH                           Enable/Disable remote command lin         │
│    A5 VNC                           Enable/Disable graphical remote a         │
│    A6 SPI                           Enable/Disable automatic loading          │
│    A7 I2C                           Enable/Disable automatic loading          │
│    A8 Serial                        Enable/Disable shell and kernel m         │
│    A9 Audio                         Force audio out through HDMI or 3         │
│    AA 1-Wire                        Enable/Disable one-wire interface         │
│                                                                               │
│                                                                               │
│                   <Select>                       <Back>                       │
│                                                                               │
└───────────────────────────────────────────────────────────────────────────────┘
```

Fig4.2.2 Enabling SSH

4.2.3 Audio setup and testing

Before running the configuration we need to test the input and output device on the raspberry pi .

The requirement to find the recording and the audio device is to input a sound card or microphone.

- For locating the microphone, it need to write down into the command prompt.

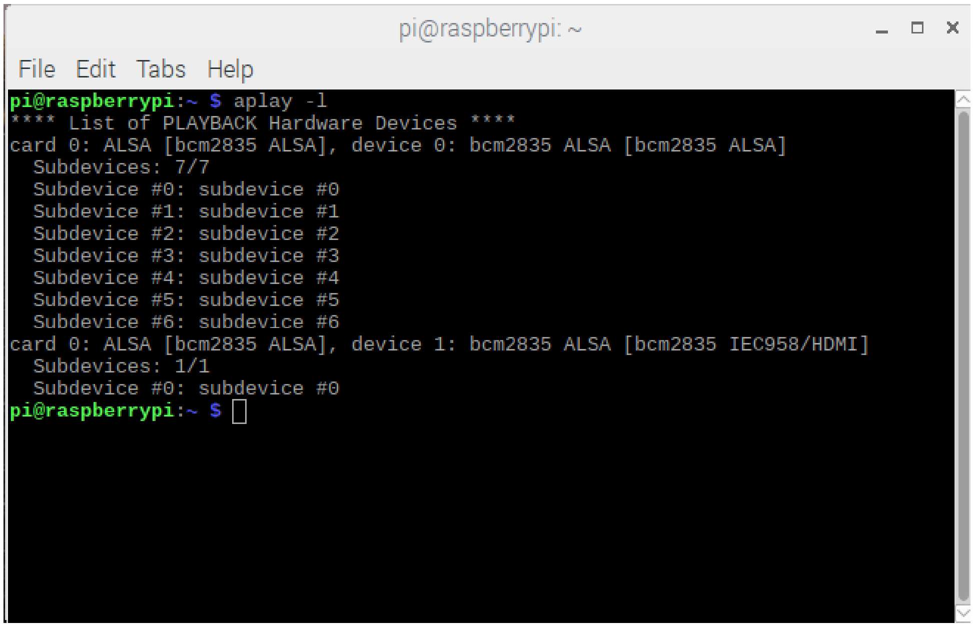
  aracord -l

  \
  Fig.4.2.3 Connected devices

  Here is my all the input device into raspberry pi.

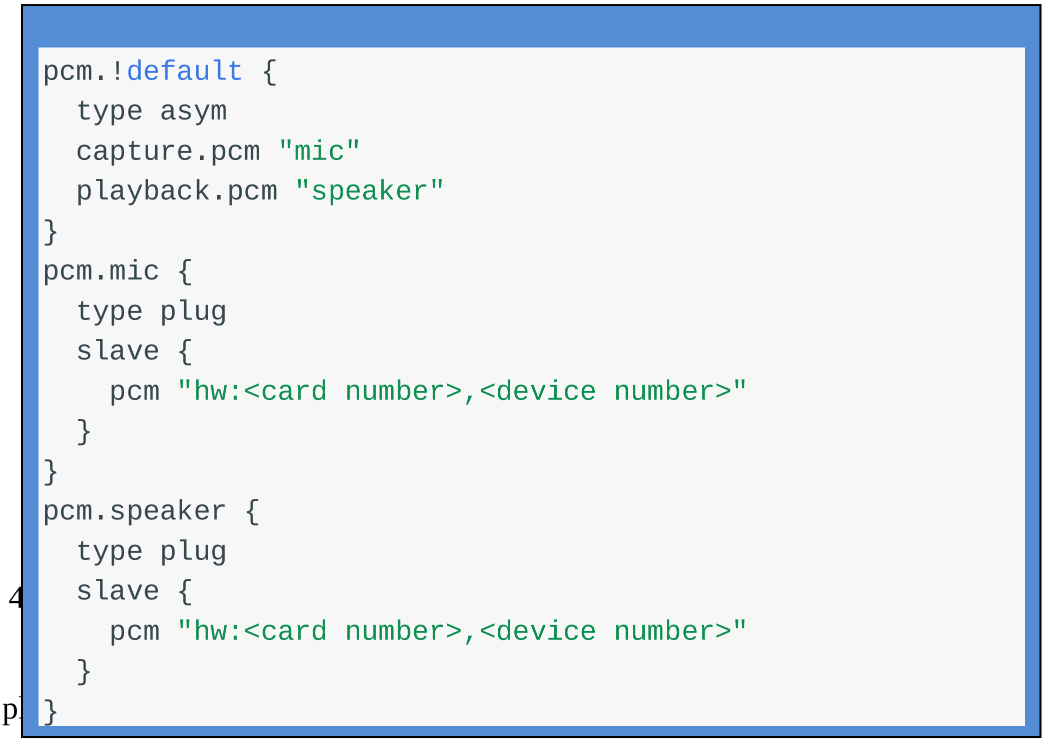
- For locating the speaker device must have the specific command

  aplay –l



```
pi@raspberrypi:~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 7/7
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
pi@raspberrypi:~ $ 
```

In the home directory of raspberry pi, we need to create a new file named *.asoundrc* .Now a identifiable speaker and microphone number should give in the following code.



```
pcm.!default {
  type asym
  capture.pcm "mic"
  playback.pcm "speaker"
}
pcm.mic {
  type plug
  slave {
    pcm "hw:<card number>,<device number>"
  }
}
pcm.speaker {
  type plug
  slave {
    pcm "hw:<card number>,<device number>"
  }
}
```

Fig 4

To p

To record an audio file.

```
arecord --format=S16_LE --duration=5 --rate=16000 --file-type=raw out.raw
```

For checking the sound

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

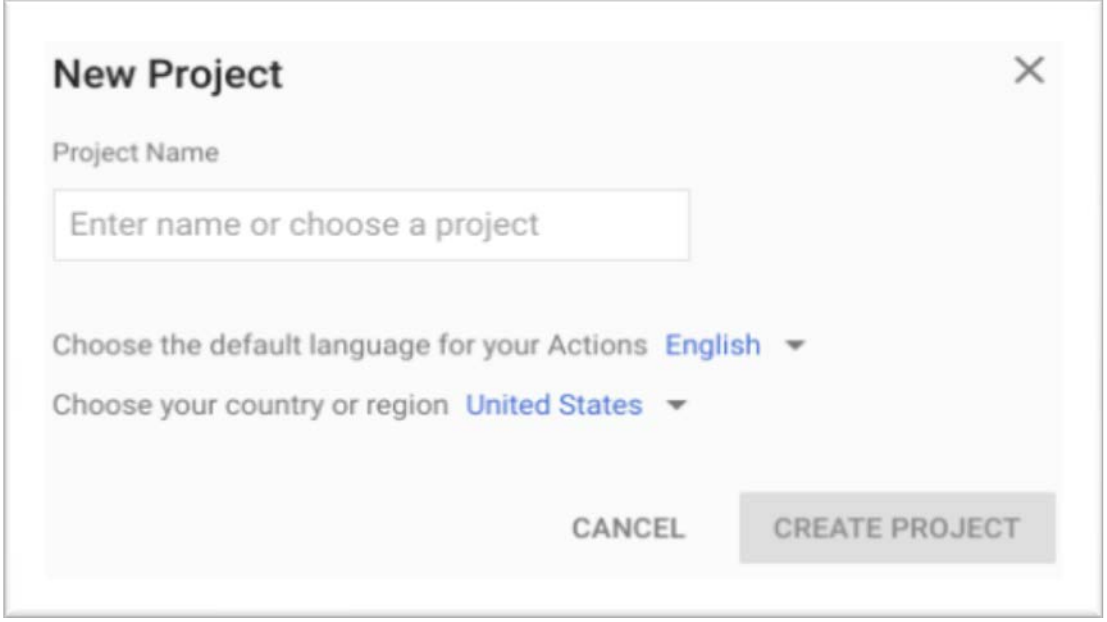
If the recording is play in the playback, then it is prepared for the overall configuration.

Chapter5

Configuration of Google assistant

5.1` Configure an Actions Console project

Google cloud platform introduced a console project for free. It gives our device to assess with the Digital Personal Assistant .For this project we can use the quota to develop a console in Google cloud. First we need to enable Google API and import the project .To Add a new project ,a project name must be write in the following box.



Choose the default language for your Actions English

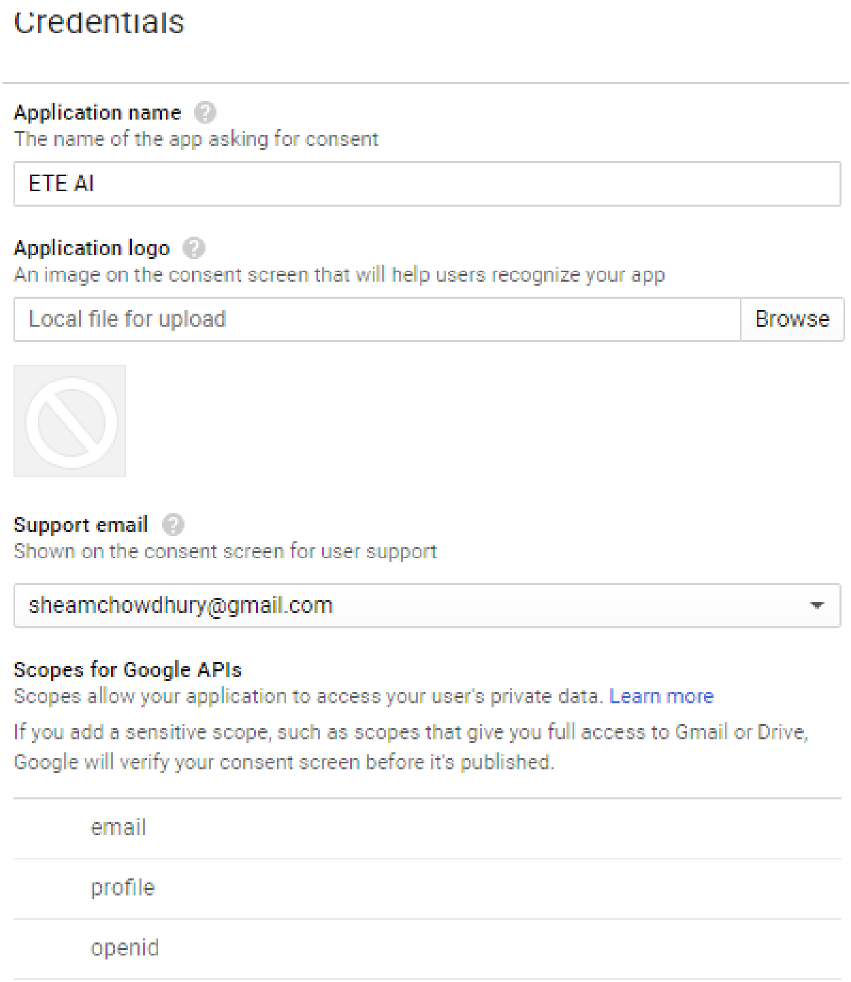
Choose your country or region United States

Fig.5.1 Create an action console project in cloud server

After create the project, registration of the project is required. To registries the device click the device registration button of the page under the advanced option.

5.1.2 Enable Google API

- To enabling the GoogleAPI, Google cloud platform console gives a vast number of API library.
- Enable the Google assistant API.
- We must configure the Oauth consent screen for project in the cloud platform.



Credentials

Application name
The name of the app asking for consent

ETE AI

Application logo
An image on the consent screen that will help users recognize your app

Support email
Shown on the consent screen for user support

sheamchowdhury@gmail.com

Scopes for Google APIs
Scopes allow your application to access your user's private data. Learn more

If you add a sensitive scope, such as scopes that give you full access to Gmail or Drive, Google will verify your consent screen before it's published.

email

profile

openid

Fig.5.1.2 Oauth concent screen

In order to make a Google assistant we must need to share the certain activity of data with Google. Google assistant function the data properly, this is not specific SDK.

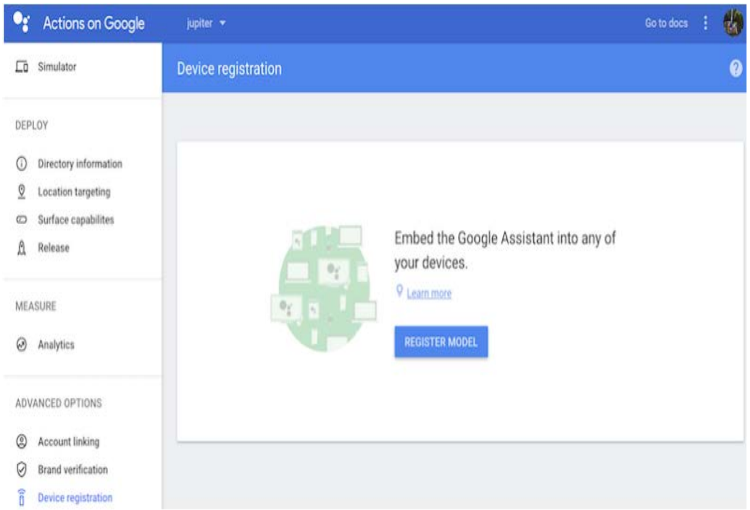
Open the activity of Google account that we need to use in the Google assistant.

Ensure the following activity are enable

- Web and app activity
- Device information activity
- Voice and information activity

To respond the following command what is given into Google assistant .Assistant device must be registrant to give information of our particular devices. The device type and the manufacturer need to include in the model. This model assemble the Google assistant associate with Action console project.

Use the registration UI to register the console.

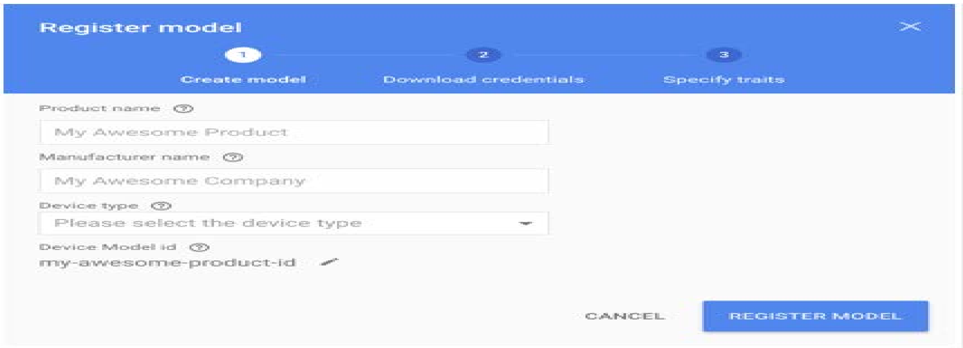


Embed the Google Assistant into any of your devices.

Fig.5.2 Register the model

5.1.3 Create Model

- Fill all the informationof ourdevice, and select any device type.
- We can find the device model in .json file



Product name

My Awesome Product

Manufacturer name

My Awesome Company

Device type

Please select the device type

Device Model id

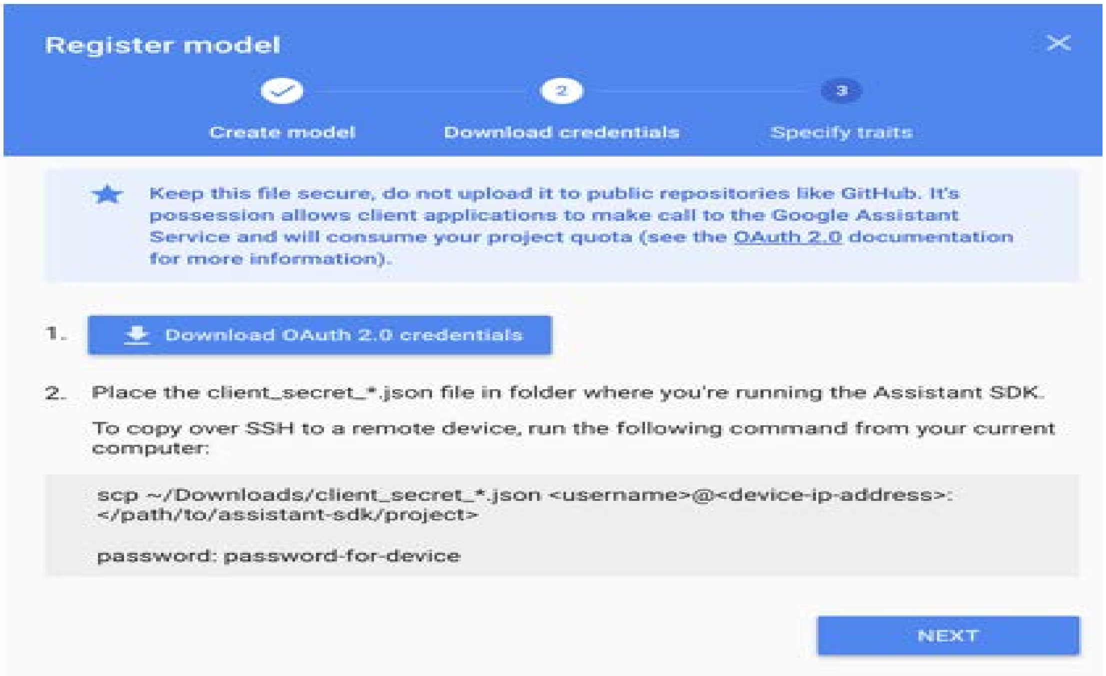
my-awesome-product-id

Fig 5.3 Creating the Model

5.1.4 Download credentials

- Download the Oauth 2.o credentials in your device.

- It must be located into the home/pi directory.
- The credentials has all the information of our model.
- To run the authorization tools Google need to authorized the model in the device.



Keep this file secure, do not upload it to public repositories like GitHub. It's possession allows client applications to make call to the Google Assistant Service and will consume your project quota (see the OAuth 2.0 documentation for more information).

1. Download OAuth 2.0 credentials

2. Place the client_secret_*.json file in folder where you're running the Assistant SDK.

   To copy over SSH to a remote device, run the following command from your current computer:

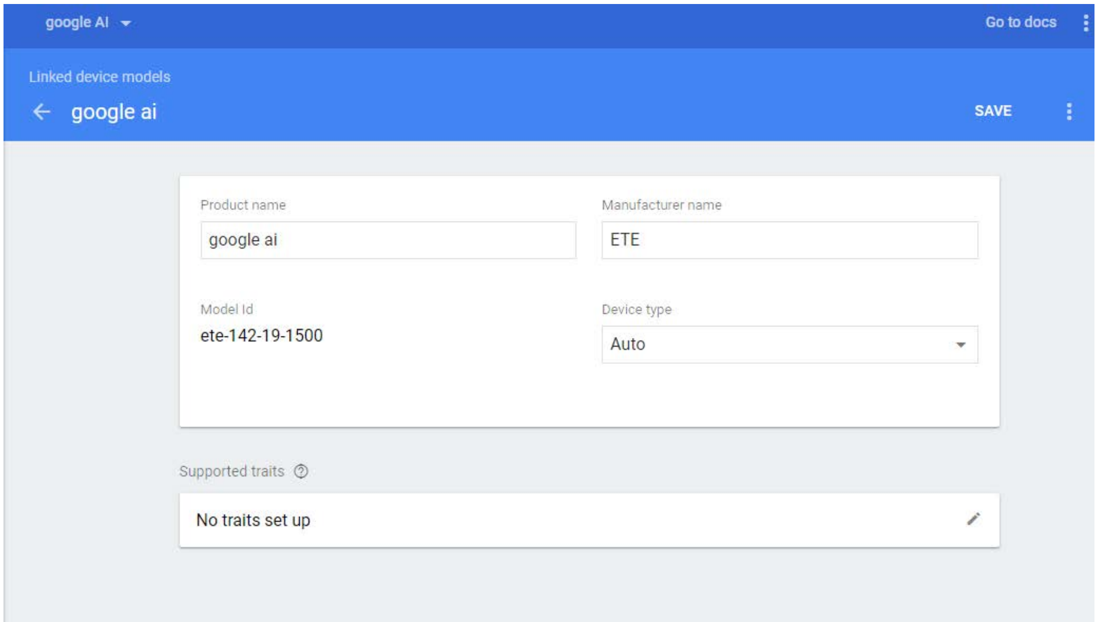
```
scp ~/Downloads/client_secret_*.json <username>@<device-ip-address>:
</path/to/assistant-sdk/project>

password: password-for-device
```

Fig 5.1.4 Download and locate the .json file

Open the new command terminal and run the following command to paste the .json file to home directory.

*scp ~/Downloads/client_secret_**client-id**.json pi@**raspberry-pi-ip-address**:/home/pi/*

Finally, it creates a device model to perform into the Google assistant.



Product name: google ai

Manufacturer name: ETE

Model Id: ete-142-19-1500

Device type: Auto

Supported traits: No traits set up

Fig.5.1.4 Final device model for Google Assistant

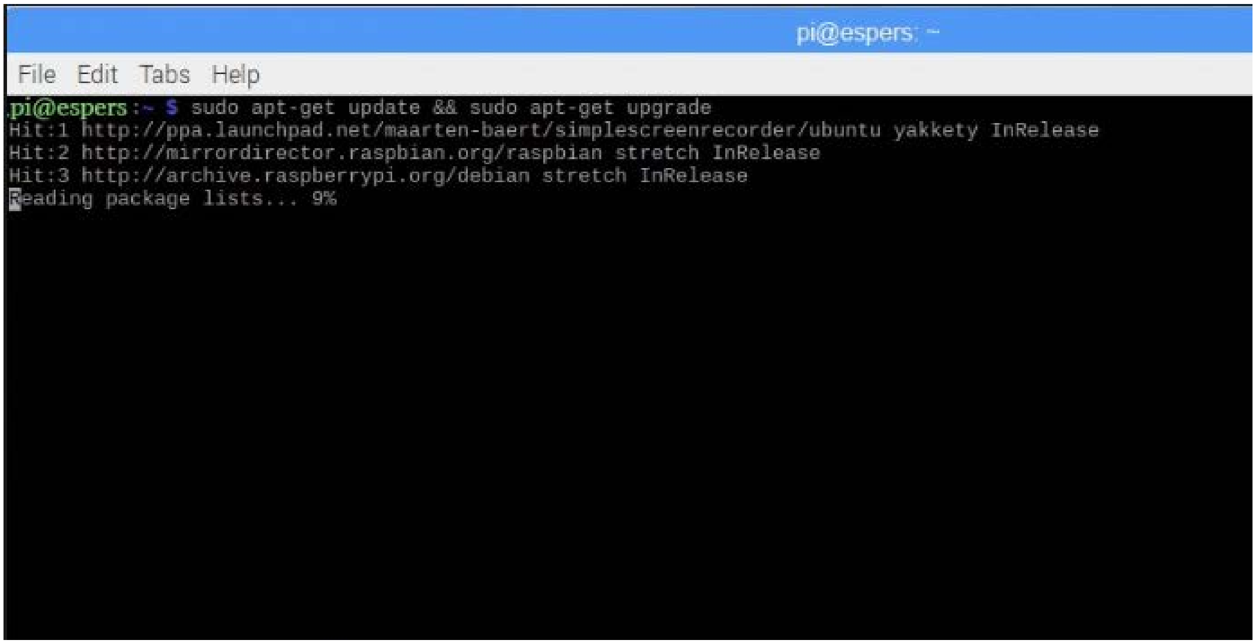
5.2 Raspberry pi configuration with Google assistant

5.2.1 Install Python 3

For install the python, we must check the latest update and upgrade of the raspberry pi.

To install the python 3 the code is

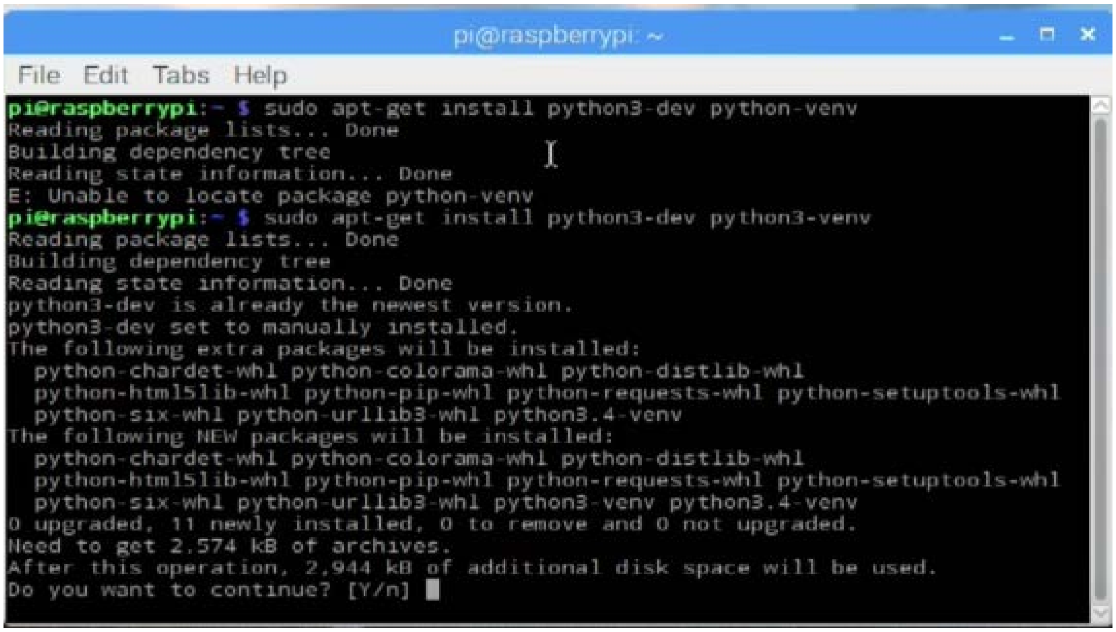
```
sudo apt-get update

sudo apt-get install python-dev python-virtualenv

virtualenv env --no-site-packages

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

source env/bin/activate
```



```
pi@espers:~ $ sudo apt-get update && sudo apt-get upgrade
Hit:1 http://ppa.launchpad.net/maarten-baert/simplescreenrecorder/ubuntu yakkety InRelease
Hit:2 http://mirrordirector.raspbian.org/raspbian stretch InRelease
Hit:3 http://archive.raspberrypi.org/debian stretch InRelease
Reading package lists... 9%
```

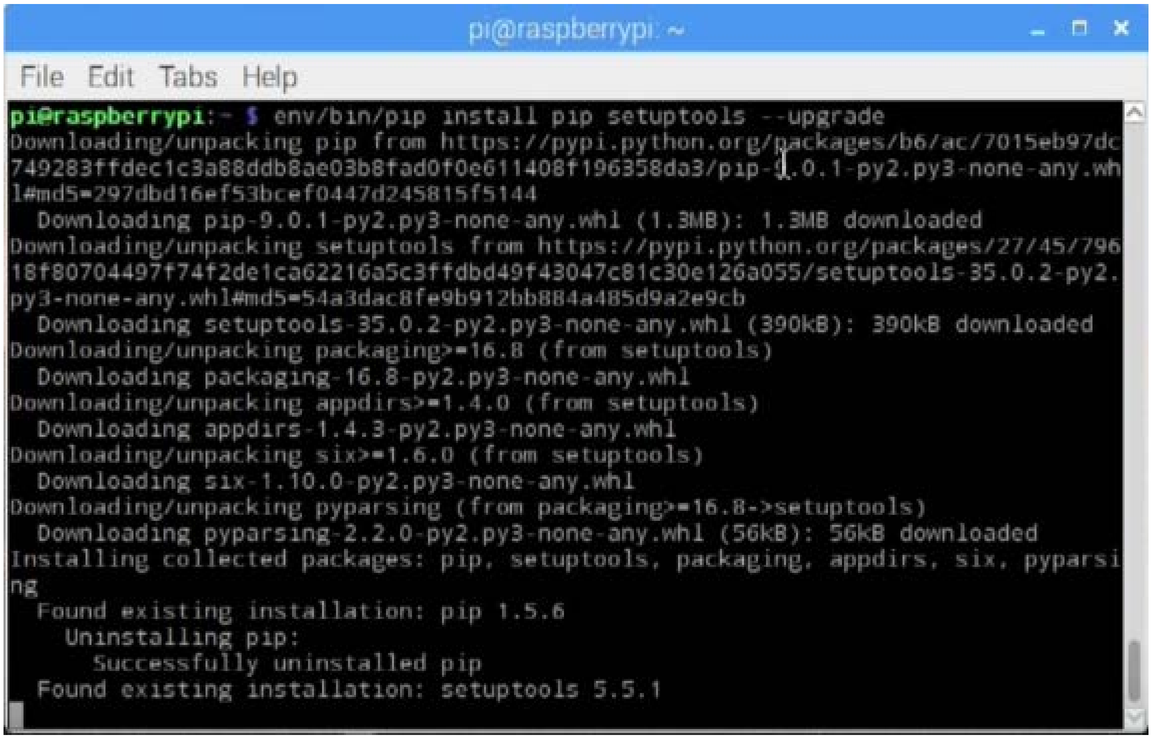
Fig.5.2.1 Update and Upgrade of Raspberry pi

Google personal assistant has to install the latest python version for isolate the SDK and its dependencies from the system python.



```
pi@raspberrypi:~ $ sudo apt-get install python3-dev python-venv
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package python-venv
pi@raspberrypi:~ $ sudo apt-get install python3-dev python3-venv
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-dev is already the newest version.
python3-dev set to manually installed.
The following extra packages will be installed:
  python-chardet-whl python-colorama-whl python-distlib-whl
  python-html5lib-whl python-pip-whl python-requests-whl python-setuptools-whl
  python-six-whl python-urllib3-whl python3.4-venv
The following NEW packages will be installed:
  python-chardet-whl python-colorama-whl python-distlib-whl
  python-html5lib-whl python-pip-whl python-requests-whl python-setuptools-whl
  python-six-whl python-urllib3-whl python3-venv python3.4-venv
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,574 kB of archives.
After this operation, 2,944 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Fig 5.2.1 Installing Python in raspberry pi



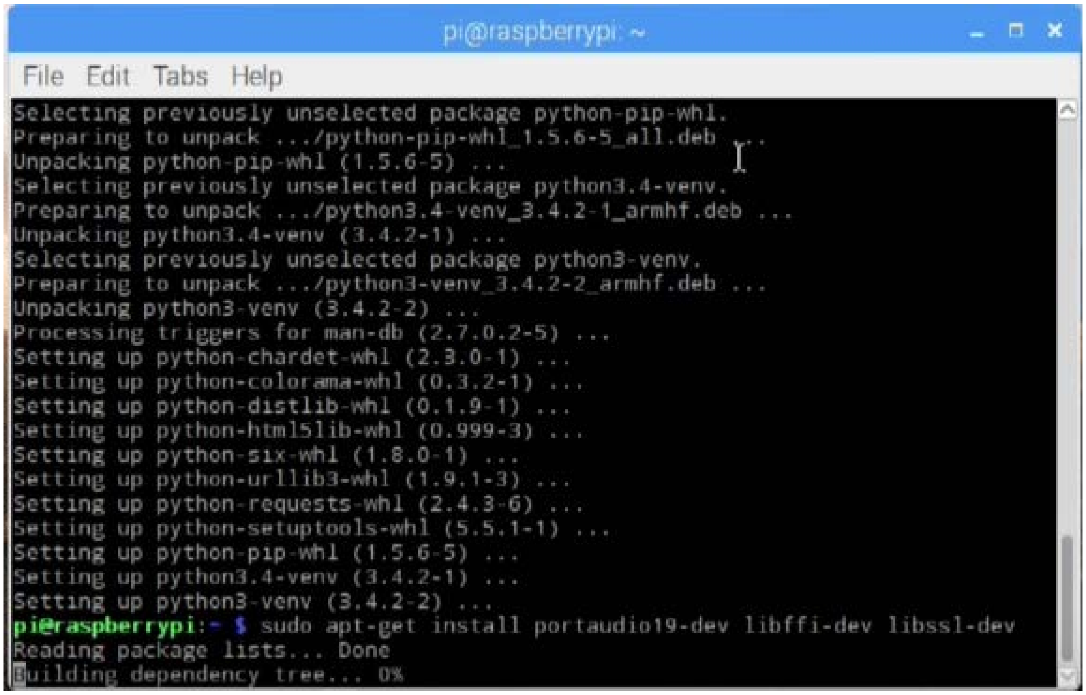
```
pi@raspberrypi:~ $ env/bin/pip install pip setuptools --upgrade
Downloading/unpacking pip from https://pypi.python.org/packages/b6/ac/7015eb97dc
749283ffdec1c3a88ddb8ae03b8fad0f0e611408f196358da3/pip-9.0.1-py2.py3-none-any.wh
l#md5=297dbd16ef53bcef0447d245815f5144
  Downloading pip-9.0.1-py2.py3-none-any.whl (1.3MB): 1.3MB downloaded
Downloading/unpacking setuptools from https://pypi.python.org/packages/27/45/796
18f80704497f74f2de1ca62216a5c3ffdbd49f43047c81c30e126a055/setuptools-35.0.2-py2.
py3-none-any.whl#md5=54a3dac8fe9b912bb884a485d9a2e9cb
  Downloading setuptools-35.0.2-py2.py3-none-any.whl (390kB): 390kB downloaded
Downloading/unpacking packaging>=16.8 (from setuptools)
  Downloading packaging-16.8-py2.py3-none-any.whl
Downloading/unpacking appdirs>=1.4.0 (from setuptools)
  Downloading appdirs-1.4.3-py2.py3-none-any.whl
Downloading/unpacking six>=1.6.0 (from setuptools)
  Downloading six-1.10.0-py2.py3-none-any.whl
Downloading/unpacking pyparsing (from packaging>=16.8->setuptools)
  Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB): 56kB downloaded
Installing collected packages: pip, setuptools, packaging, appdirs, six, pyparsi
ng
  Found existing installation: pip 1.5.6
    Uninstalling pip:
      Successfully uninstalled pip
  Found existing installation: setuptools 5.5.1
```

Fig5.2.1b Upgrade the Python pip

5.2.2 Install the Package system dependencies

Install the dependencies with following command

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

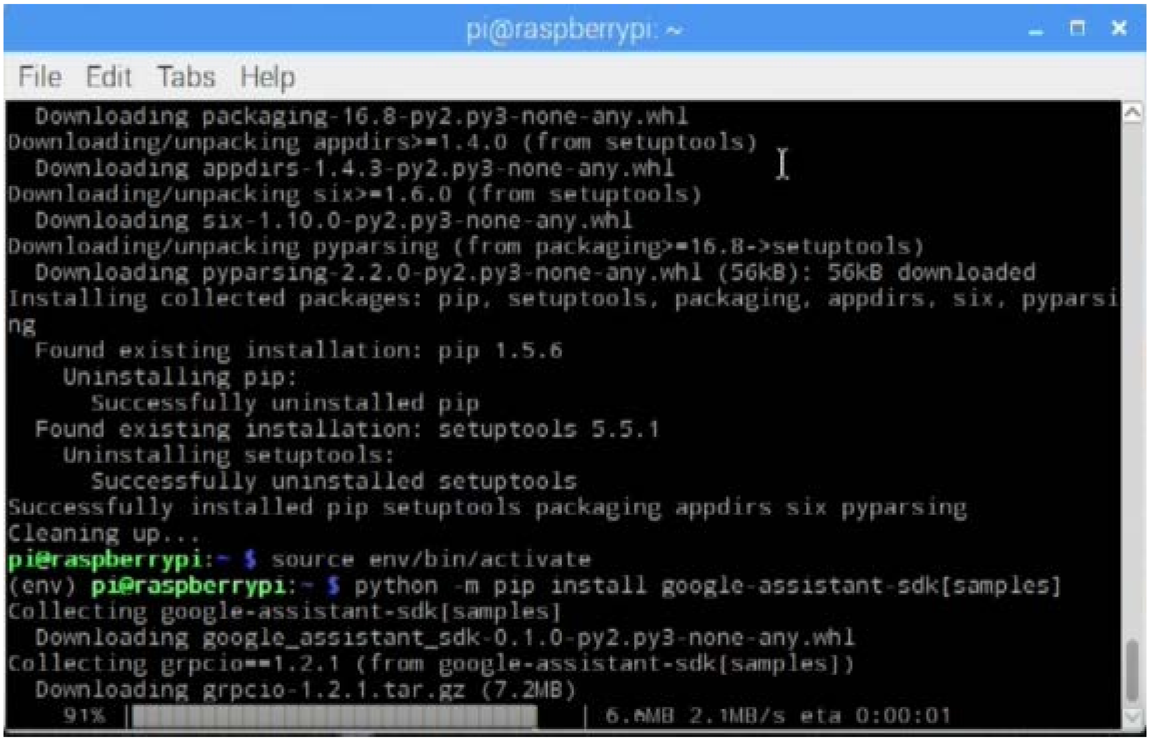


```
Selecting previously unselected package python-pip-whl.
Preparing to unpack .../python-pip-whl_1.5.6-5_all.deb ...
Unpacking python-pip-whl (1.5.6-5) ...
Selecting previously unselected package python3.4-venv.
Preparing to unpack .../python3.4-venv_3.4.2-1_armhf.deb ...
Unpacking python3.4-venv (3.4.2-1) ...
Selecting previously unselected package python3-venv.
Preparing to unpack .../python3-venv_3.4.2-2_armhf.deb ...
Unpacking python3-venv (3.4.2-2) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up python-chardet-whl (2.3.0-1) ...
Setting up python-colorama-whl (0.3.2-1) ...
Setting up python-distlib-whl (0.1.9-1) ...
Setting up python-html5lib-whl (0.999-3) ...
Setting up python-six-whl (1.8.0-1) ...
Setting up python-urllib3-whl (1.9.1-3) ...
Setting up python-requests-whl (2.4.3-6) ...
Setting up python-setuptools-whl (5.5.1-1) ...
Setting up python-pip-whl (1.5.6-5) ...
Setting up python3.4-venv (3.4.2-1) ...
Setting up python3-venv (3.4.2-2) ...
pi@raspberrypi:~ $ sudo apt-get install portaudio19-dev libffi-dev libssl-dev
Reading package lists... Done
Building dependency tree... 0%
```

Fig 5.2.2 .Dependencies of Python

Install the pip to create a virtual environment to the raspberry pi .It uses the following command.

```
python -m pip install --upgrade google-assistant-library
```



```
  Downloading packaging-16.8-py2.py3-none-any.whl
Downloading/unpacking appdirs>=1.4.0 (from setuptools)
  Downloading appdirs-1.4.3-py2.py3-none-any.whl
Downloading/unpacking six>=1.6.0 (from setuptools)
  Downloading six-1.10.0-py2.py3-none-any.whl
Downloading/unpacking pyparsing (from packaging>=16.8->setuptools)
  Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB): 56kB downloaded
Installing collected packages: pip, setuptools, packaging, appdirs, six, pyparsi
ng
  Found existing installation: pip 1.5.6
    Uninstalling pip:
      Successfully uninstalled pip
  Found existing installation: setuptools 5.5.1
    Uninstalling setuptools:
      Successfully uninstalled setuptools
Successfully installed pip setuptools packaging appdirs six pyparsing
Cleaning up...
pi@raspberrypi:~ $ source env/bin/activate
(env) pi@raspberrypi:~ $ python -m pip install google-assistant-sdk[samples]
Collecting google-assistant-sdk[samples]
  Downloading google_assistant_sdk-0.1.0-py2.py3-none-any.whl
Collecting grpcio==1.2.1 (from google-assistant-sdk[samples])
  Downloading grpcio-1.2.1.tar.gz (7.2MB)
    91% |█████████████████████████████     | 6.6MB 2.1MB/s eta 0:00:01
```

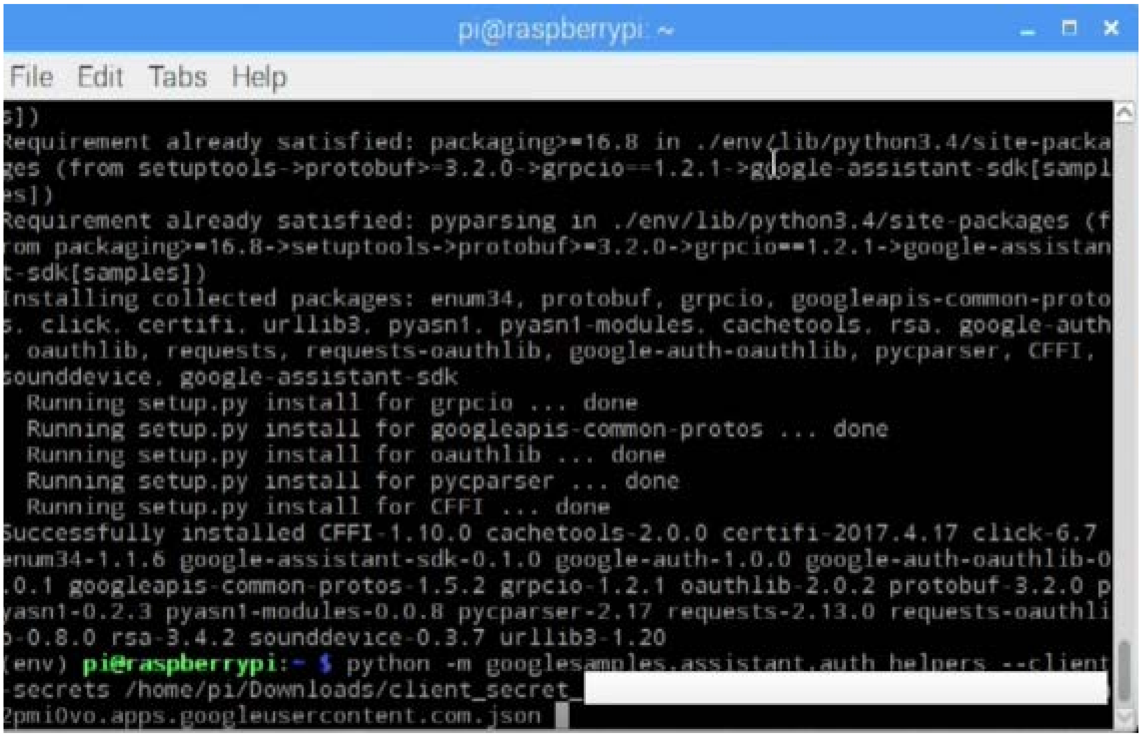
Fig.5.2.2b Creating e virtual environment

5.2.3 Generate credentials

We downloaded a .json file in previous in the home directory. Now we need to generate it with codes. Credentials able to generate and run the system tools. It might create the virtual environment to access through the internet with the Google action console cloud server and send an authentication code to the server for allow the whole system.
The following codes are

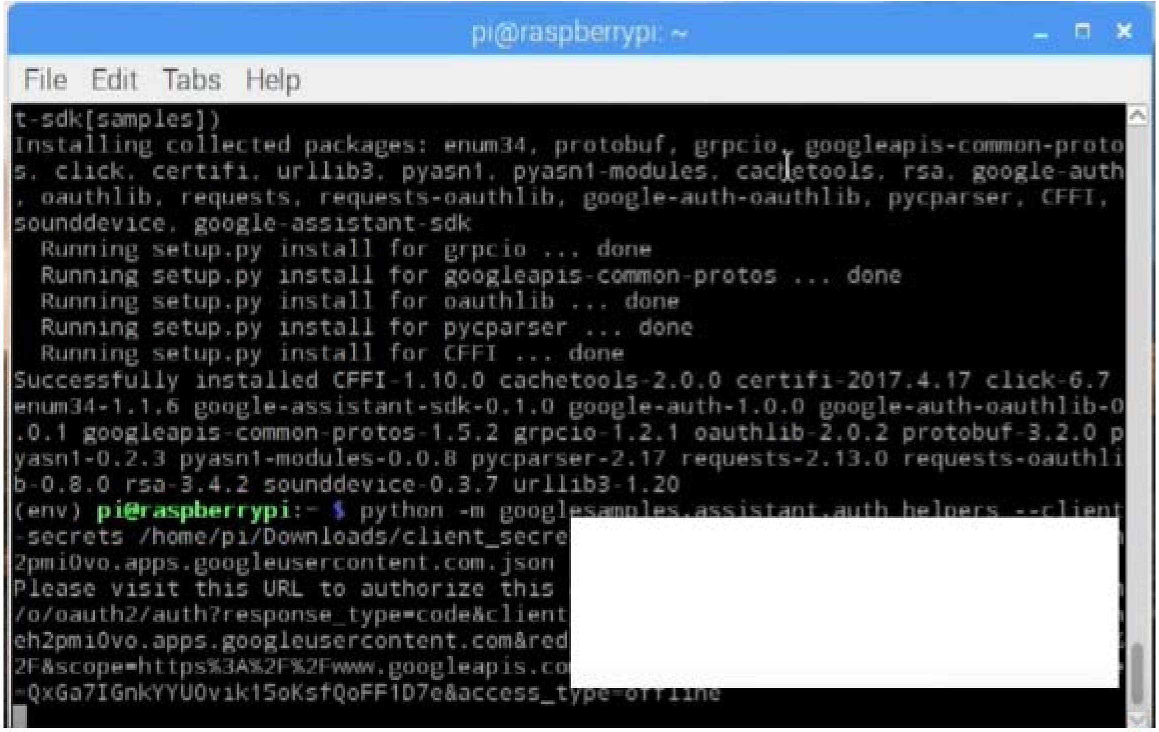
```
google-oauthlib-tool --scope https://www.googleapis.com/auth/assistant-sdk-prototype \

    --scope https://www.googleapis.com/auth/gcm \

    --save --headless --client-secrets /path/to/client_secret_client-id.json
```

We need to visit the given URL to get the authentication code.
The client secret .json file is the secret file that the device can easily understand to find the selected server. It authenticate the API into the raspberry pi.
Type the command into the command prompt.



```
s])
Requirement already satisfied: packaging>=16.8 in ./env/lib/python3.4/site-packa
ges (from setuptools->protobuf>=3.2.0->grpcio==1.2.1->google-assistant-sdk[sampl
es])
Requirement already satisfied: pyparsing in ./env/lib/python3.4/site-packages (f
rom packaging>=16.8->setuptools->protobuf>=3.2.0->grpcio==1.2.1->google-assistan
t-sdk[samples])
Installing collected packages: enum34, protobuf, grpcio, googleapis-common-proto
s, click, certifi, urllib3, pyasn1, pyasn1-modules, cachetools, rsa, google-auth
, oauthlib, requests, requests-oauthlib, google-auth-oauthlib, pycparser, CFFI,
sounddevice, google-assistant-sdk
  Running setup.py install for grpcio ... done
  Running setup.py install for googleapis-common-protos ... done
  Running setup.py install for oauthlib ... done
  Running setup.py install for pycparser ... done
  Running setup.py install for CFFI ... done
Successfully installed CFFI-1.10.0 cachetools-2.0.0 certifi-2017.4.17 click-6.7
enum34-1.1.6 google-assistant-sdk-0.1.0 google-auth-1.0.0 google-auth-oauthlib-0
.0.1 googleapis-common-protos-1.5.2 grpcio-1.2.1 oauthlib-2.0.2 protobuf-3.2.0 p
yasn1-0.2.3 pyasn1-modules-0.0.8 pycparser-2.17 requests-2.13.0 requests-oauthli
b-0.8.0 rsa-3.4.2 sounddevice-0.3.7 urllib3-1.20
(env) pi@raspberrypi:~ $ python -m googlesamples.assistant.auth helpers --client
-secrets /home/pi/Downloads/client_secret_[redacted]
2pmi0vo.apps.googleusercontent.com.json
```

Fig.5.2.3 Authentication request on API(client_secret file is hiding for security purposes)



```
t-sdk[samples])
Installing collected packages: enum34, protobuf, grpcio, googleapis-common-proto
s, click, certifi, urllib3, pyasn1, pyasn1-modules, cachetools, rsa, google-auth
, oauthlib, requests, requests-oauthlib, google-auth-oauthlib, pycparser, CFFI,
sounddevice, google-assistant-sdk
  Running setup.py install for grpcio ... done
  Running setup.py install for googleapis-common-protos ... done
  Running setup.py install for oauthlib ... done
  Running setup.py install for pycparser ... done
  Running setup.py install for CFFI ... done
Successfully installed CFFI-1.10.0 cachetools-2.0.0 certifi-2017.4.17 click-6.7
enum34-1.1.6 google-assistant-sdk-0.1.0 google-auth-1.0.0 google-auth-oauthlib-0
.0.1 googleapis-common-protos-1.5.2 grpcio-1.2.1 oauthlib-2.0.2 protobuf-3.2.0 p
yasn1-0.2.3 pyasn1-modules-0.0.8 pycparser-2.17 requests-2.13.0 requests-oauthli
b-0.8.0 rsa-3.4.2 sounddevice-0.3.7 urllib3-1.20
(env) pi@raspberrypi:~ $ python -m googlesamples.assistant.auth helpers --client
-secrets /home/pi/Downloads/client_secre[redacted]
2pmi0vo.apps.googleusercontent.com.json
Please visit this URL to authorize this
/o/oauth2/auth?response_type=code&client
eh2pmi0vo.apps.googleusercontent.com&red
2F&scope=https%3A%2F%2Fwww.googleapis.co
-QxGa7IGnkYYU0vik15oKsfQoFF1D7e&access_type=offline
```

Fig.5.2.3 authentication URL

Simply visit the URL given in the windows and allow access the permission to authenticate the API into the Raspberry pi.Copy the code and past to the windows. If authorization was successful, you we will see a response like this,

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

Now it is authenticate with Google assistant.

5.3 Run the sample code

: At this situation we are ready to make a query of the assistant. To run the samples write the following command to the windows

```
googlesamples-assistant-hotword --project-id my-dev-project --device-model-id my-model
```

Now it able to understand the human voice and we can make a conversation with the device.

Use command ”OK, GOOGLE” to listen up the voice that assistant conscious about.

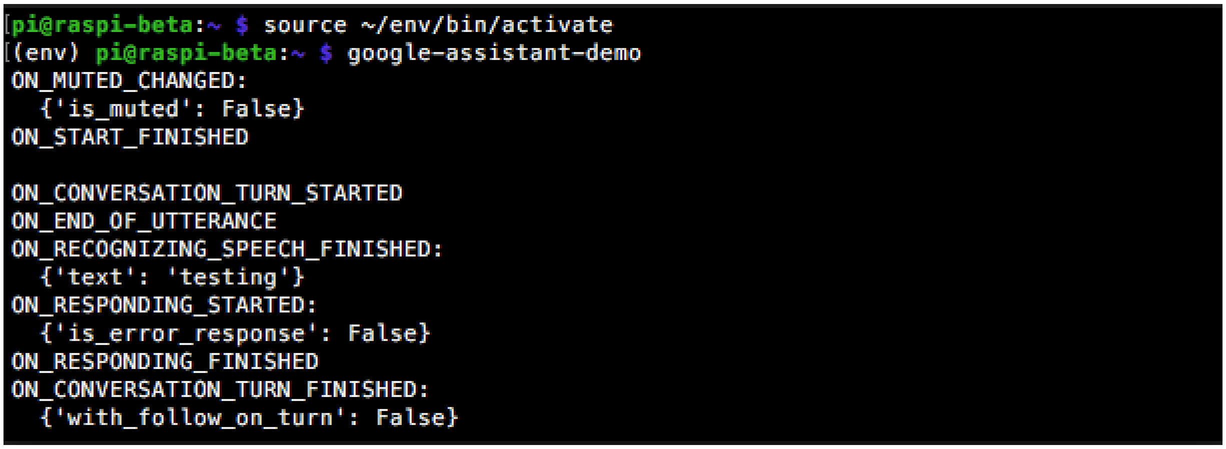
Like,

“Who is mark zugerberg?”

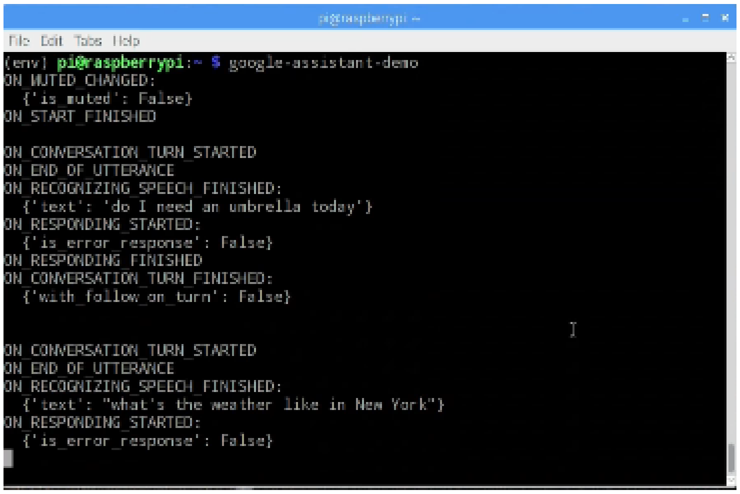
“Who are you?”

“Who am I”?

5.4 Results



```
[pi@raspi-beta:~ $ source ~/env/bin/activate
[(env) pi@raspi-beta:~ $ google-assistant-demo
 ON_MUTED_CHANGED:
   {'is_muted': False}
 ON_START_FINISHED

 ON_CONVERSATION_TURN_STARTED
 ON_END_OF_UTTERANCE
 ON_RECOGNIZING_SPEECH_FINISHED:
   {'text': 'testing'}
 ON_RESPONDING_STARTED:
   {'is_error_response': False}
 ON_RESPONDING_FINISHED
 ON_CONVERSATION_TURN_FINISHED:
   {'with_follow_on_turn': False}
```



```
(env) pi@raspberrypi:~ $ google-assistant-demo
ON_MUTED_CHANGED:
  {'is_muted': False}
ON_START_FINISHED

ON_CONVERSATION_TURN_STARTED
ON_END_OF_UTTERANCE
ON_RECOGNIZING_SPEECH_FINISHED:
  {'text': 'do I need an umbrella today'}
ON_RESPONDING_STARTED:
  {'is_error_response': False}
ON_RESPONDING_FINISHED
ON_CONVERSATION_TURN_FINISHED:
  {'with_follow_on_turn': False}


ON_CONVERSATION_TURN_STARTED
ON_END_OF_UTTERANCE
ON_RECOGNIZING_SPEECH_FINISHED:
  {'text': "what's the weather like in New York"}
ON_RESPONDING_STARTED:
  {'is_error_response': False}
```

Chapter 6

Smart Mirror

6.1 Setup Smart Mirror

Now we have to install the smart mirror interface on the raspberry pi .for the mirror interface we simply install a latest Node.js file with following command.

- curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
- sudo apt install -y node.js

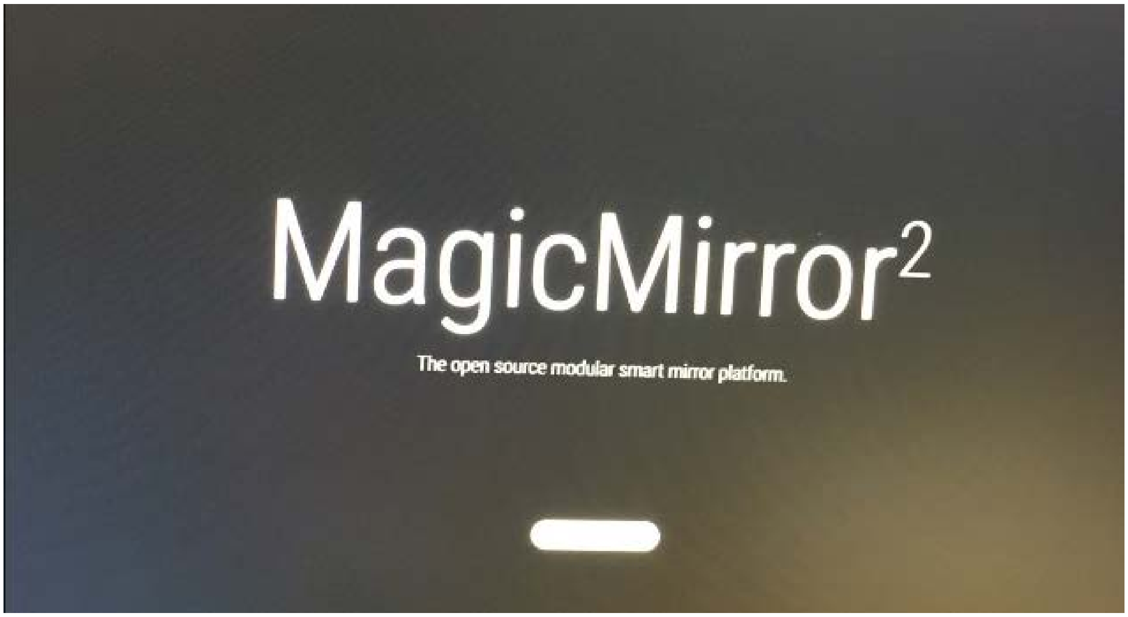
For the repository we are helped by the github developer MichMich .He create a repository for smart mirror.Simply we clone the repository using,

- git clone https://github.com/MichMich/MagicMirror

now Enter the repository and install and run the app

- cd/magicmirror
- npm install && npm start

we see a pop-up windows of smart mirror is shown in the display.like that,



In this part the mirror display is a blank display.Now we can add modules on this display as a user demand.

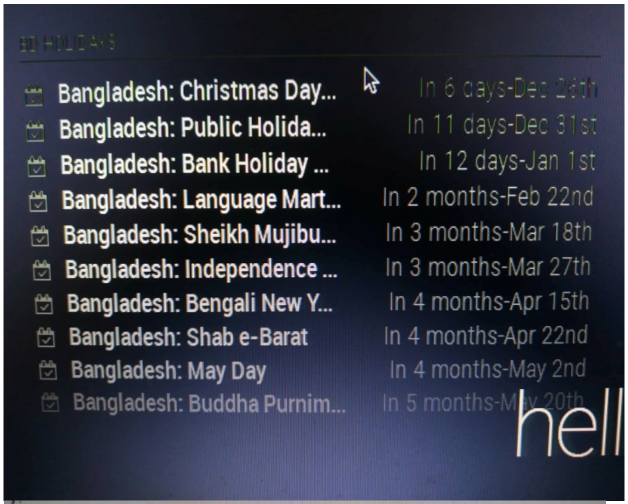
6.3 Smart Mirror Modules

6.2.1Clock

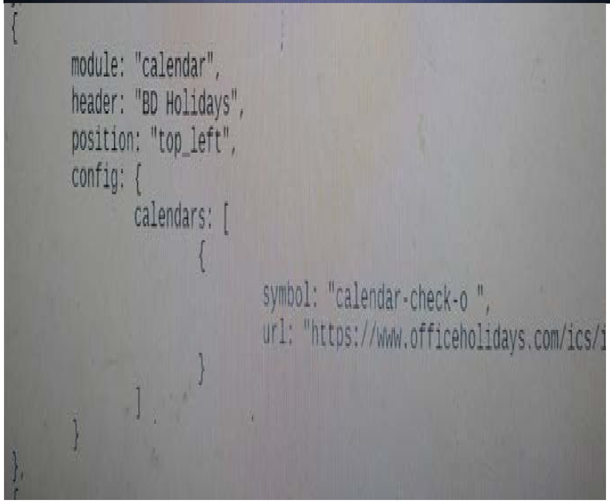
Clock are installed in the smart mirror by default. We can set the time zone to edit the config.js file into the magic mirror folder.

6.2.2 Calendar

We need to configure the calendar with Bangladesh holyday. Simply edit the config.js and add a calendar CIS in the following lines.



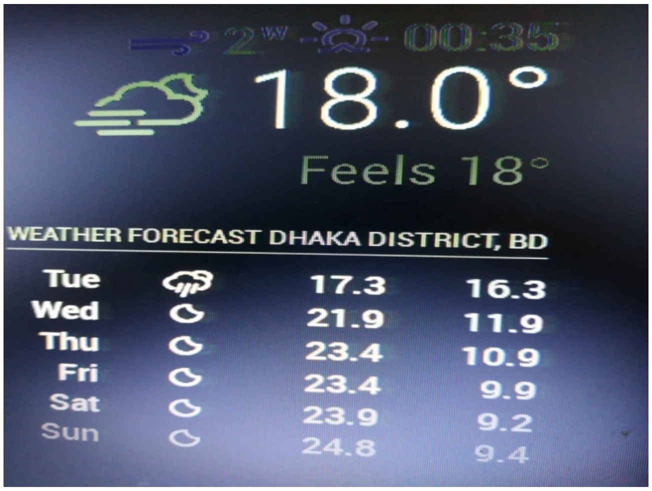
```
BD HOLIDAYS
Bangladesh: Christmas Day...      In 6 days-Dec 26th
Bangladesh: Public Holida...      In 11 days-Dec 31st
Bangladesh: Bank Holiday ...      In 12 days-Jan 1st
Bangladesh: Language Mart...      In 2 months-Feb 22nd
Bangladesh: Sheikh Mujibu...      In 3 months-Mar 18th
Bangladesh: Independence ...      In 3 months-Mar 27th
Bangladesh: Bengali New Y...      In 4 months-Apr 15th
Bangladesh: Shab e-Barat          In 4 months-Apr 22nd
Bangladesh: May Day               In 4 months-May 2nd
Bangladesh: Buddha Purnim...      In 5 months-May 20th
```



```
{
        module: "calendar",
        header: "BD Holidays",
        position: "top_left",
        config: {
                calendars: [
                        {
                                symbol: "calendar-check-o ",
                                url: "https://www.officeholidays.com/ics/i
                        }
                ]
        }
},
```

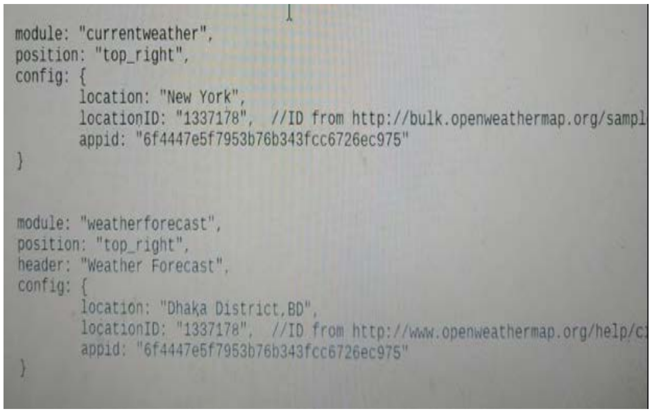
6.2.3 Weather Forecast

To add a weather forecast on magic mirror we need to create a Open weather API and simply edit the api key in the config.js file.



```
2 W   00:35
18.0°
Feels 18°
WEATHER FORECAST DHAKA DISTRICT, BD
Tue   17.3   16.3
Wed   21.9   11.9
Thu   23.4   10.9
Fri   23.4    9.9
Sat   23.9    9.2
Sun   24.8    9.4
```

Fig :6.2.3 Weather forecast Window

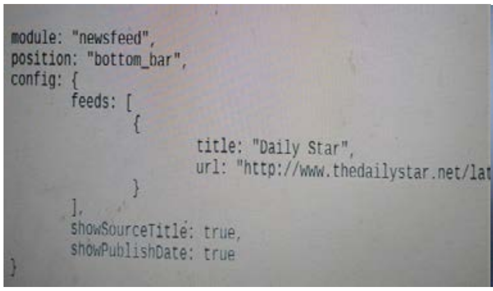


```
module: "currentweather",
position: "top_right",
config: {
        location: "New York",
        locationID: "1337178",  //ID from http://bulk.openweathermap.org/sampl
        appid: "6f4447e5f7953b76b343fcc6726ec975"
}


module: "weatherforecast",
position: "top_right",
header: "Weather Forecast",
config: {
        location: "Dhaka District,BD",
        locationID: "1337178",  //ID from http://www.openweathermap.org/help/c
        appid: "6f4447e5f7953b76b343fcc6726ec975"
}
```

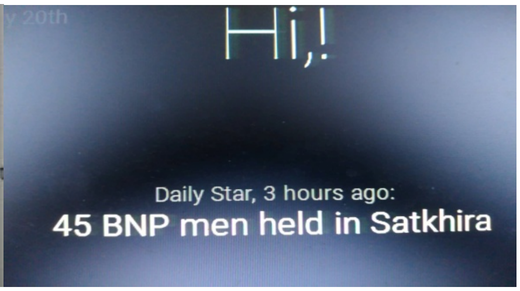
Fig 6.2.3 b Weather forecast setup

6.2.4 News Feed

Replace the RSS link from the bangle news portal into the news feed configuration.



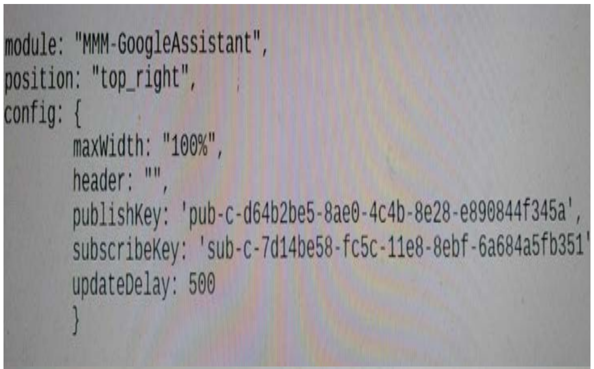
```
module: "newsfeed",
position: "bottom_bar",
config: {
        feeds: [
                {
                        title: "Daily Star",
                        url: "http://www.thedailystar.net/lat
                }
        ],
        showSourceTitle: true,
        showPublishDate: true
}
```



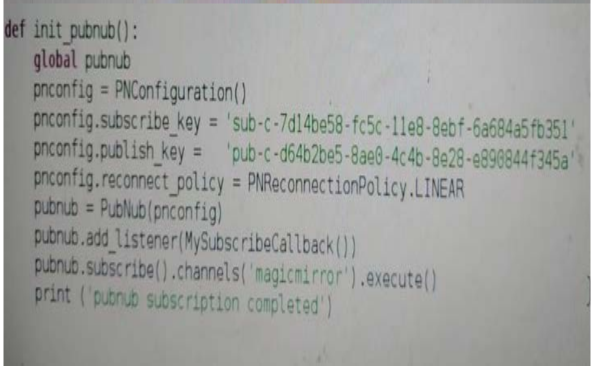
```
y 20th
Hi,!
Daily Star, 3 hours ago:
45 BNP men held in Satkhira
```

6.2.5 Google Assistant

Finally its time to add a google assistant module into the smart mirror .For this step we find a solution to use a online DSN (Data stream network) called PUBNUB.Just set the publish key and the subscriber key in the assistant.py .We need to install leatest pubnub dependencies into the env/bin directory.



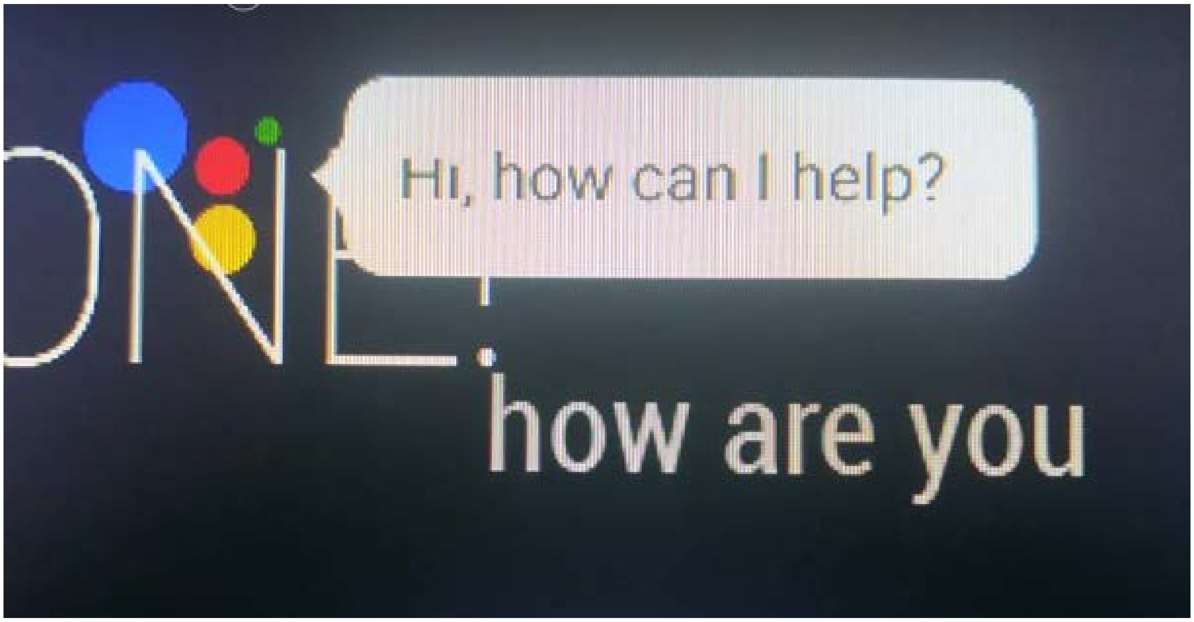
```
module: "MMM-GoogleAssistant",
position: "top_right",
config: {
      maxWidth: "100%",
      header: "",
      publishKey: 'pub-c-d64b2be5-8ae0-4c4b-8e28-e890844f345a',
      subscribeKey: 'sub-c-7d14be58-fc5c-11e8-8ebf-6a684a5fb351'
      updateDelay: 500
      }
```



```
def init_pubnub():
    global pubnub
    pnconfig = PNConfiguration()
    pnconfig.subscribe_key = 'sub-c-7d14be58-fc5c-11e8-8ebf-6a684a5fb351'
    pnconfig.publish_key =   'pub-c-d64b2be5-8ae0-4c4b-8e28-e890844f345a'
    pnconfig.reconnect_policy = PNReconnectionPolicy.LINEAR
    pubnub = PubNub(pnconfig)
    pubnub.add_listener(MySubscribeCallback())
    pubnub.subscribe().channels('magicmirror').execute()
    print ('pubnub subscription completed')
```

Fig :6.2.5 a config.js

Fig:6.2.5 b Assistant.py



Hi, how can I help?

how are you

Fig:6.2.5 c Assistant windows

Future Implementation

The idea of AI is first written about by ‘Alan Turing’. He posed a question that “Can machines think” in 1950. I think the answer is ‘Yes’. Because, Now AI is posed to have a profounder impact on our lives and its everywhere. The Future which be an exciting, fascinating new world completely from where it is now. Computer power doubles every 18 months. This is possible because Computers are rapidly catching up human intelligence. Now a day’s machine is more powerful and stronger than us. We have now improved of the quality of machine translation using AI. And of course, in the lab, in healthcare, in material science, agriculture, automated system, animation etc.Smart mirror can now automate everyday tasks, process and analyze huge amounts of data, help with management decisions, create algorithms to forecast growth, help doctors diagnose patients, and much more. In progress of the modern world is not thinkable without the development of science and technology and without the implementation of technological innovations. Now we discuss about some hottest smart mirror AI technology which are lead the future-

Natural Language Generation: It is basically introducing text from computer data. It is currently use in customers service and summarizing business intelligence.

Speech Recognition: It is transitive and transform of human speech. So it is useful for computer application. It is currently used in interactive voice response system and mobile applications.

Virtual Agents: It is the current dialing of the media. Now it is use by big forms like amazon, apple, Google, IBM, Microsoft etc.

In 2018 AI technology trends –

- **Deep Learning Theory**
- **Capsule Networks**
- **Probability Programming**
- **Digital Twin**
- **Lean and Augmented Data Learning**
- **Deep Reinforcement Learning**

In future AI help us to-

- **Rebuild Society**
- **Tactical Step**
- **Education**
- **Driverless Vehicles**

Chapter7

Conclusion

This project introduce the basic structure of a Smart Mirror with its Artificial intelligence.This is a Google based AI and designed to converse with human with a coherent structure for a incapable people. This smart mirror AI system has used speech to communicate with both input and output channel.Also the Smart Mirror Assistant system increase the interaction between the incapable patient or a blind man and computer by using specific feature like speech recognition, text recognition and the knowledge base. Moreover, this project enable the lengthy way to simplify the daily life to giving a simple command. This Google AI system can also be used in the healthcare, weatherforecast, education, business and entertainment. It has also ability to create a smart home, vehicle navigation, Robotics for security access.

Chapter8

Reference:

1. V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J.Hazen, and L. Hetherington, “JUPITER : A Telephone-Based Conversational Interface for Weather Information,”IEEE Transactions on Speech and Audio Processing,vol. 8, no. 1, pp. 85–96, 2000.

2. M. Schr¨oder and J. Trouvain, “The German text-tospeech synthesis system MARY: A tool for research, development and teaching,” International Journal of Speech Technology, 2003.

3.Amazon. Amazon Lex is a service for building conversational interfaces. https://aws.amazon.com.

4.Google action console SDK cloud server https://https://console.actions.google.com/

5. K. Wagner. Facebook's Virtual Assistant 'M' Is Super Smart. It's Also Probably a Human. https://www.recode.com.

6.R. Mead. 2017.Semio: Developing a Cloud-based Platform forMultimodal Conversational AI in Social Robotics. 2017 IEEE International Conference on Consumer Electronics (ICCE).

7. Apple. (n.d.). *Siri. Your wish is it's command.*.
Available:http://www.apple.com/uk/ios/siri/. Last accessed 15th April 2013.

8. Bednar, P M, Welch, C and Graziano, A (2007) ‘Learning Objects and their implications onLearning: a case of developing the foundation for a new Knowledge Infrastructure,’Chapter 6 in ‘*Learning Objects: Applications, Implications & Future Directions*‘. Harman,K and Koohang, A, eds., Informing Science Press, NY, pp 157-185