Dissertation Titled

Homoto: Personal Home Assistant

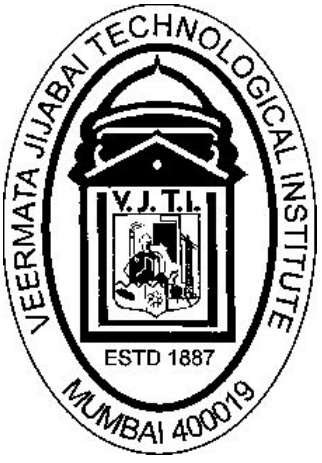
Submitted in partial fulfillment of the requirements of the degree of

B. Tech (Computer Engineering)
2017-18

By

Siddharth K. Nagda [141070023]
Akshit K. Gandhi [141070026]
Neel M. Rami [141070027]
Ankit P. Gala [141070063]

Under the guidance of
Dr. G. P. Bhole



Department of Computer Engineering & Information Technology
Veermata Jijabai Technological Institute
Mumbai - 400019
(An Autonomous Institute affiliated to University of Mumbai)

2017-18

Statement of Candidate

We wish to state that the work embodied in this report entitled "**Homoto: Personal Home Assistant**" is our group contribution to the work carried out under the guidance of Prof. G. P. Bhole at Veermata Jijabai Technological Institute. This work has not been submitted by any other degree/diploma of any University/Institute. Whenever references have been made to previous work of other, it has been clearly indicated.

Signature of Candidates:

Siddharth K. Nagda _____

Akshit K. Gandhi _____

Neel M. Rami _____

Ankit P. Gala _____

Approval Sheet

This Dissertation entitled "**Homoto: Personal Home Assistant**" by Siddharth K. Nagda [141070023], Akshit K. Gandhi [141070026], Neel M. Rami [141070027] and Ankit P. Gala [141070063], which is found to be satisfactory and approved for the Degree of Bachelors of Technology (Computer Engineering and Information Technology).

_____

Examiner

_____

Dr. G. P. Bhole

Date:

Place: VJTI, Mumbai.

Certificate

This is to certify that Siddharth K. Nagda [141070023], Akshit K. Gandhi [141070026], Neel M. Rami [141070027] and Ankit P. Gala [141070063] students of B. Tech Computer, Veermata Jijabai Technology Institute, Mumbai have successfully completed the Bachelors of Technology Dissertation on "**Homoto: Personal Home Assistant**" under the Guidance of Prof. G. P. Bhole.

Dr. G. P. Bhole
Project Guide

Examiner

Acknowledgements

We would like to thank our project guide **Dr. G. P. Bhole** for contributing his time and effort to help us during the course of this Project giving it the present shape. It would have been impossible to complete the project without his support, valuable suggestions, criticism, encouragement and guidance.

He has always been involved by discussing our topic at each phase to make sure that the experiment is designed and carried out in an appropriate manner and that our conclusions are appropriate, given their results. His constant support and interaction have been a driving force which has constantly motivated us to explore the different aspects of our project. We would also like to thank **Dr. V. B. Nikam**, Head of the Computer Engineering Information Technology Department for his guidance and motivation.

I am also grateful to all other teaching, non-teaching staff members and our colleagues of Computer Engineering & Information Technology department for directly or indirectly helping us for the completion of this project and the resources provided.

Abstract

Smart speakers are stand-alone hardware devices that provide voice-enabled digital guidance. They are powered by digital assistants which are an AI-powered software that responds to voice commands. Examples of these software include Alexa (Amazon), Assistant (Google), Siri (Apple) and Cortana (Microsoft). This project aims to build a voice assistant Homoto in a hardware device powered by RaspberryPi.

By using this device the user can perform various functionalities such as controlling home appliances such as fans, lights etc, getting news , getting weather information, playing songs, getting nearby places information, getting price estimates of cab rides, getting prices of products. All these functionalities are performed using users voice.

Most of the smart speakers work only with smart devices such as smart lights, smart fans, smart TVs etc. This project also aims to break away this barrier by working with traditional devices. Also this system works very well with Indian accent voices which is not present in other smart speakers.

Report Orientation

The report is comprised of seven chapters with different content and scenarios providing the complete details about the project. The report is completed in such a way that it first provides the background knowledge about the project and then gives the thorough details about it. The different chapters of the report are as follows :-

Chapter 1: Introduction

This chapter will provide introduction to the project. It describes the problem statement and the system requirements.

Chapter 2: Literature review

The chapter starts by discussing the available literature in general.

Chapter 3: System Design

This chapter contains System Architecture. It also contains a high level view of the software and hardware architecture.

Chapter 4: Implementation

This chapter gives details about the system was implemented from the hardware as well as software point of view.

Chapter 5: Implementation

This chapter starts by giving details about the steps required by the user to setup the device and then provides details about the functionalities offered by the device.

Chapter 6: Result And Discussions

This chapter gives the results obtained while working on project and gives details on the current problems being faced.

Chapter 7: Conclusion and Future Scope

This chapter gives the conclusion and future scope of the proposed System.

Contents

List of Figures

Abbreviations

ATM	**A**utomated **T**eller **M**achine
TV	**T**elevision
NLP	**N**atural **L**anguage **P**rocessing
TTS	**T**ext **T**o **S**peech
ML	**M**achine **L**earning **P**rocessing
AI	**A**rtificial **I**ntelligence
API	**A**pplication **P**rogramming **I**nterface
GPIO	**G**eneral **P**urpose **I**nput **O**utput
GPS	**G**lobal **P**ositioning **S**ystem
LED	**L**ight **E**mmiting **D**iode
SSH	**S**ecure **S**hell
IDE	**I**ntegrated **D**evelopment **E**nvironment
SPI	**S**erial **P**rogramming **I**nterface
I2C	**I**nter-**I**C **B**us
IOT	**I**nternet **O**f **T**hings
USB	**U**niversal **S**erial **B**us
MQTT	**M**essage **Q**ueue **T**elemetry **Transport**
IR	**I**nfra **R**ed
PHP	**P**ersonal **H**ome **P**age
CMU	**U**arnegie **M**ellon **U**niversity
RAM	**R**andom **A**ccess **M**emory
GB	**G**iga **B**ytes
GHz	**G**iga **H**ertz
HTML	**H**yper **T**ext **M**arkup **L**anguage
BJT	**B**ipolar **J**unction **T**ransitior

Chapter 1

Introduction and Problem Statement

1.1 Introduction

In today's tech world, one of the latest and the fastest growing trend is smart speakers. Almost half of the owners who own these smart speakers use them for playing music and listening to audio books. Around 20% of the theses users order products by giving voice commands to smart speakers. So it's clear that smart speakers have become one of the biggest trends of this decade.

A smart speaker is a type of wireless speaker with voice control built into them along with an integrated virtual assistant or artificial intelligence. These speakers are triggered by a hot word or several hot words. They can also act as smart devices by providing control to home automation devices by using wireless protocols such as Wi-fi, Bluetooth etc.

These speakers have some level of automation and can be programmed for some specific use. Most smart speakers have some local computing with intuitive user interface. Though configuration flexibility is limited, but they are fast and efficient in doing what they are expected to do.

A smart speaker can perform various operations like playing a song, answering a question which was asked by the user, turning on the lights in the room, providing weather report, booking an Uber, telling you a joke and many more things.

1.2 Problem Statement

A voice based smart home assistant (Homoto) which adds intelligence to traditional devices and also integrates third party services.

1.2.1 Elaboration

Homoto is a personal home assistant which is controlled via the user's voice. It provides an interface to control various traditional devices such as lights, air conditioners etc. This personal assistant is triggered by the hot word "Homoto". After the personal assistant is triggered, a user can perform various actions such as turning on the lights by saying "turn on the lights" or turn on the television by saying 'switch on the tv'.

Homoto also provides other functionalities by integrating with third party services. It provides the user with weather updates, get ride estimate for a cab, tell the price of an item from Amazon, give the latest news, play a song and provide information on nearby ATMs, restaurants, movie theatres etc.

This device also aims to work with traditional devices such as lights, fans rather than only with smart devices. So we can add intelligence to traditional devices. It recognizes Indian accent voices as well.

1.2.2 Significance

Smart speakers are becoming more and more important and are being considered as the next big thing after smart phones. They are being used more and more. They have various use cases which are as follows:-

- Disabled people can easily perform various actions only by using their voices. For example a blind person can easily control his fans, lights etc.
- People can easily get information by using their voice if they are engaged in some other tasks. For example a person while driving can easily get information about nearby petrol pumps
- Can control devices such as TV and ACs without using remote controller.
- Users can use voice and natural language which is more convenient than other interfaces

1.3 System Requirements

1. Wifi Connectivity.

2. A power source for charging the home assistant device.

Chapter 2

Review of Literature

Only a countable number of corporations have developed such smart speakers with voice-controlled intelligent personal assistant. This arises primarily due to the complex model of this device. The area of related work relevant to the project include:

- Methods and metrics for the evaluation of search systems.
- Modelling and classification of human speech.
- Inferring satisfactory behaviour from the inputs.[1]

Some of the major voice-controlled intelligent speakers in the market are:

2.1 Amazon Echo

Amazon Echo is a brand of smart speakers developed by Amazon. The devices connect to the voice-controlled intelligent personal assistant service Alexa, which responds to the hotword "Alexa".[2]

2.1.1 Operation

By default the device continuously listens to all speech, monitoring for the hotword "Alexa"(Customisable). Echo's microphones can be manually disabled by pressing a mute button to turn off the audio processing circuit. Internet connectivity is continuously required for the device to work. Echo's voice recognition functionalities is based on Amazons proprietary web services and some acquired voice platforms.User behavior modeling has been used extensively for evaluating search systems. [3]

2.1.2 Voice System

Alexa is a virtual assistant developed by Amazon, first used in the Amazon Echo and the Amazon Echo Dot smart speakers developed by Amazon Lab126. The devices have natural lifelike voices resulting from speech-unit technology. High speech accuracy is achieved through sophisticated natural language processing (NLP) algorithms built into the Echo's text-to-speech (TTS) engine. [4]

2.1.3 Applications

Amazon Echo is capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audiobooks, and providing weather, traffic, sports, and other real-time information, such as news. Alexa can also control several smart devices using itself as a home automation system. Users can extend the Alexa capabilities by installing "skills" (additional functionality developed by third-party vendors, in other settings more commonly called apps such as weather programs and audio features.[4]

Some of the commands are:

- Basic commands
  Ask for help: "Alexa, help." Have a conversation: "Alexa, let's chat."
- Show
  Ask for what the Echo Show can display: "Alexa, what can you show me?" Show your calendar: "Alexa, show my calendar."
- Media controls
  Play music: "Alexa, play some music." Play music on other Alexa devices: "Alexa, play in the living room" or "Alexa, play everywhere."
- Time and date
  Set an alarm: "Alexa, set an alarm for 7 a.m." or "Alexa, wake me up at 7 in the morning."
- Calls and messaging
  Call another Echo user: "Alexa, call [name]." Answer an incoming call: "Alexa, answer the call" or "Alexa, answer."
- News and weather
  Ask for your Flash Briefing: "Alexa, what's my Flash Briefing?" Add music news to your Flash Briefing: "Alexa, enable Today in Music."

- Math
  Convert units: "Alexa, how many [units] are in [units]?" Convert units: "Alexa, how many [units] are in 2 [units]?"

- Smart home
  Turn lights on or off: "Alexa, turn on the lights" or "Alexa, turn off the living room lights." Dim the lights: "Alexa, dim the lights to 50 percent."

- Search
  Get Wikipedia information: "Alexa, Wikipedia: [subject]." Tell Alexa to continue reading a Wikipedia entry: "Alexa, tell me more." [5]

2.1.4 Variants

There are multiple speakers avalable with changes according to the user needs :-

- Echo Dot
- Amazon Tap
- Echo Look
- Echo Show
- Echo Spot
- Echo Plus

2.2 Google Smart Speakers

Google Home is a brand of smart speakers developed by Google. Launched in 2016, Google Home speakers enable users to speak voice commands to interact with services through Google's intelligent personal assistant(A.I.) called Google Assistant. The original product has a cylindrical shape with colored status LEDs on the top for visual representation of its status, and the cover over the base is modular, with different color options offered through Google Store intended for the device to blend into the environment.

2.2.1 Operation

By default the device continuously listens to all speech, monitoring for the hotword "OK Google"(Customisable). The best and worst thing about Google Home is that it plugs into the Google brain and everything that company knows about the user : search queries, mapping and location, emails and web browsing too. [6]

2.2.2 Voice System

Google Assistant, an intelligent personal assistant, is included as the main and only assistant in Google Home. Unlike its cousin, Google Now, Assistant is able to engage in two-way conversations with users. Googles efforts in AI are aiding in these improvements. For example, a deep learning technique known as neural beamforming allowed the company to release Google Home with only two microphones, but achieving the same quality as having eight. On the system level, improvements include adapting the model using Google Home specific data. We present results on a variety of multichannel sets.[7]

2.2.3 Application

Various forms of both in-house and third-party services are integrated into Google Home, allowing users to speak voice commands to control interaction with them. Examples of supported services include Google Play Music, Spotify and iHeartRadio for audio, Netflix, YouTube and Google Photos for videos and photos, Google Calendar and Google Keep for tasks, and CNN, CNBC and The Wall Street Journal for news updates. New services are integrated on an ongoing basis.[8]
Some of the commands are:

- Basic commands
  Ask for help: "OK, Google, help." Control the volume: "OK, Google, turn it up," "OK, Google, Louder" or "OK, Google,
- Time
  Time: "OK, Google, what time is it?" Time in other locations: "OK, Google, what's the time in London?"
- Tools
  Recipes: "OK, Google, how do I make [dish]?" Uber: "OK, Google, order an Uber."

- Search
  Stocks: "OK, Google, how are Alphabet's stocks doing?" Words: "OK, Google, what does [word] mean?"

- Shopping
  Get voice shopping instructions: "OK, Google, how do I shop?" Order items from Google Express: "OK, Google, buy dish soap."

- Media
  Play music: "OK, Google, play some music" or "Play some [genre] music." Play ambient sounds: "OK, Google, help me relax" or "OK, Google, play white noise" or "OK, Google play forest sounds."

- Smart home
  Turn smart lights on/off: "OK, Google, turn on/off my lights." Dim smart lights: "OK, Google, dim my lights to fifty percent."

- Third-party actions
  21 Blackjack: "OK, Google, let me talk to 21 Blackjack." Best Dad Jokes: "OK, Google, talk to Best Dad Jokes." [9]

2.2.4 Variants

There are multiple speakers available with changes according to the user needs:-

- Home
- Home Mini
- Home Max

2.3 Apple HomePod

HomePod is a smart speaker developed by Apple Inc.It was announced on June 5, 2017, at the Apple Worldwide Developers Conference, and originally scheduled for release in December 2017. Only a single version of the speaker is out yet, with optional white or black speaker. Works with only iPhone 5s or later, iPad Pro, iPad (5th generation), iPad Air or later, iPad mini 2 or later, or iPod touch (6th generation) running iOS 11.2.5 or later.[10]

2.3.1 Audio Integration

- Apple Music1
- iTunes Music Purchases
- iCloud Music Library with an Apple Music or iTunes Match subscription
- Beats 1 Live Radio
- Apple Podcasts
- AirPlay other content to HomePod from iPhone, iPad, iPod touch, Apple TV, and Mac [11]

2.3.2 Voice System

The voice system in Home Pod is fully base on Apple's AI voice assistant Siri. Siri's speech recognition engine was provided by Nuance Communications, a speech technology company. The assistant uses voice queries and a natural-language user interface to answer questions, make recommendations, and perform actions by delegating requests to a set of Internet services. The software adapts to users' individual language usages, searches, and preferences, with continuing use. Returned results are individualized. [12]

2.3.3 Commands

- Phone and Text actions, such as "Call Sarah", "Read my new messages", "Set the timer for 10 minutes", and "Send email to mom"
- Check basic information, including "What's the weather like today?" and "How many dollars are in a Euro?"
- Schedule events and reminders, including "Schedule a meeting" and "Remind me to"
- Handle device settings, such as "Take a picture", "Turn on Wi-Fi", and "Increase the brightness"
- Search the Internet, including "Define...", "Find pictures of...", and "Search Twitter for..."
- Navigation, including "Take me home", "What's traffic like on the way home?", and "Find driving directions to..." [13]

Chapter 3

System Design

In this section we will elaborate on the system design. A basic architecture of the system will be shown as visualized by the end user, along with the hardware and the software architecture. A description of the various components will also be provided in the sections that follow.

3.1 System Architecture

In this section we will discuss the architecture of the system as perceived by the user. The architecture is depicted in the figure 3.1.

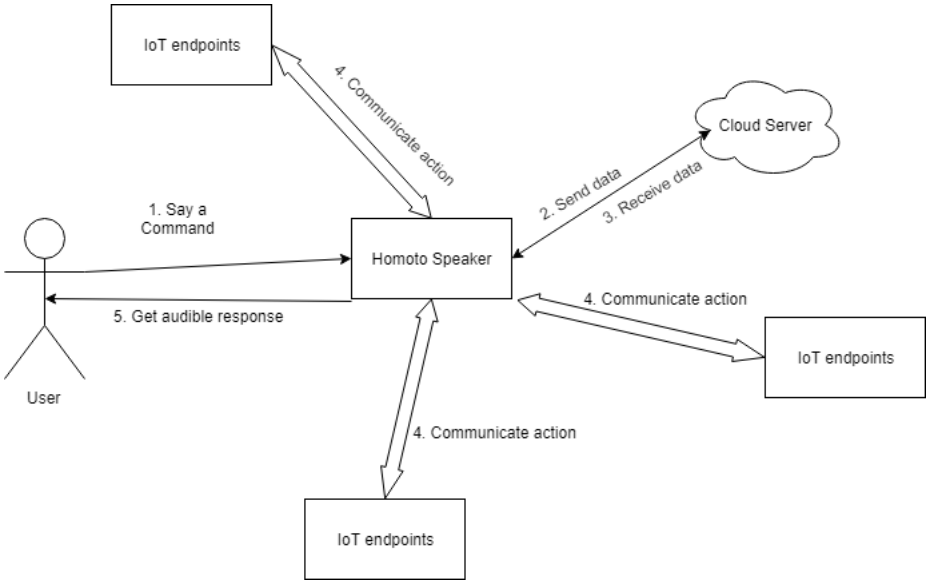


FIGURE 3.1: System Architecture

The entities in the architecture are:-

1. **User**: The user is the actor who will communicate with the system by giving commands to the speaker in the form of voice data.
2. **Homoto Speaker**: It is main hardware component that takes the user input in the form of voice, performs some prepossessing, sends the data to the cloud server and then performs the command received from the cloud server either by communicating with IoT endpoints or via third party API's.
3. **Cloud Server**: The server handles the requests received from the homoto speaker and sends the processed output back to the speaker.
4. **IoT endpoints**: The IoT endpoints are the embedded devices that will interact with the physical world objects like lights, fans, AC's & TV. They have an actuating capability.

3.2 Hardware Architecture

In this section we will discuss the high level hardware architecture of the homoto speaker. The homoto speaker has various components as shown in the figure 3.2.

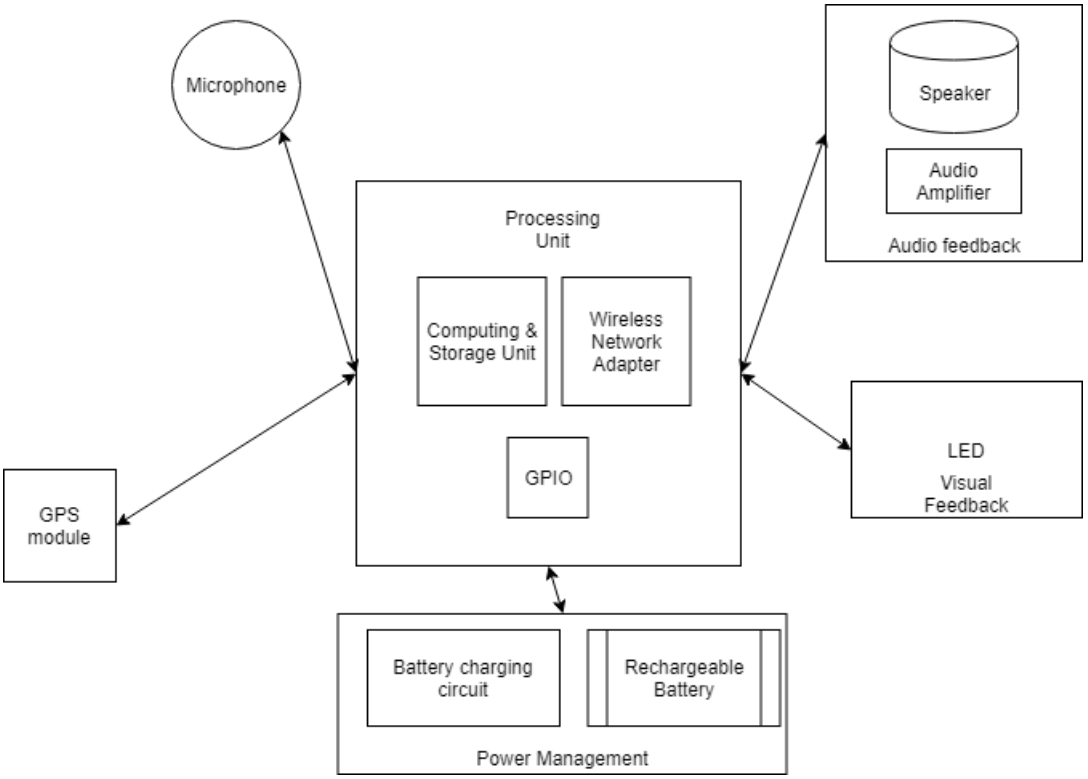


FIGURE 3.2: Homoto Speaker Architecture

The entities in this architecture are:-

1. **Processing Unit**: The heart of the system is the computer which has the capability to run an Operating System along with the following modules:-
    - **Computing & Storage Unit**: This module performs all the computation and storage of temporary and persistent data.
    - **Wireless Network Adapter**: This module provides wireless communication capabilities to the processing unit.
    - **GPIO**: This module provides Input Output capability to the processing unit. It allows sensor data to be taken as input and allows output of data to actuators and components.
2. **Microphone**: The microphone provides capability to capture audio data and send the data to the processing unit for further processing.
3. **Audio Feedback**: This module forms a part of the human interaction feedback mechanism by playing user understandable audio. This module consists of the speaker and the audio amplifier.
    - **Speaker**: This module allows the electrical signals from the processor to be converted into sound so that the user can get audible feedback.
    - **Audio Amplifier**: This module amplifies the electrical signal that is input to the speaker so that the audio can be heard clearly.
4. **GPS**: The Global Positioning System sensor provides the location of the user which will be used for various location based services.
5. **Power Management**: Since the speaker is a standalone system, it needs it's own power supply which will be provided by this module.
    - **Battery Charging Circuit**: This module helps in charging the onboard rechargeable battery.
    - **Rechargeable Battery**: The battery provides the power to the various onboard sensors and processors.

This was the high level architecture of the homoto speaker. Now, let's consider the architecture of the various IoT endpoints. It is depicted in figure 3.3

The entities in this architecture are:-

1. **Processing Unit**: The heart of the system is the computer which has the capability to run a program along with the following modules:-

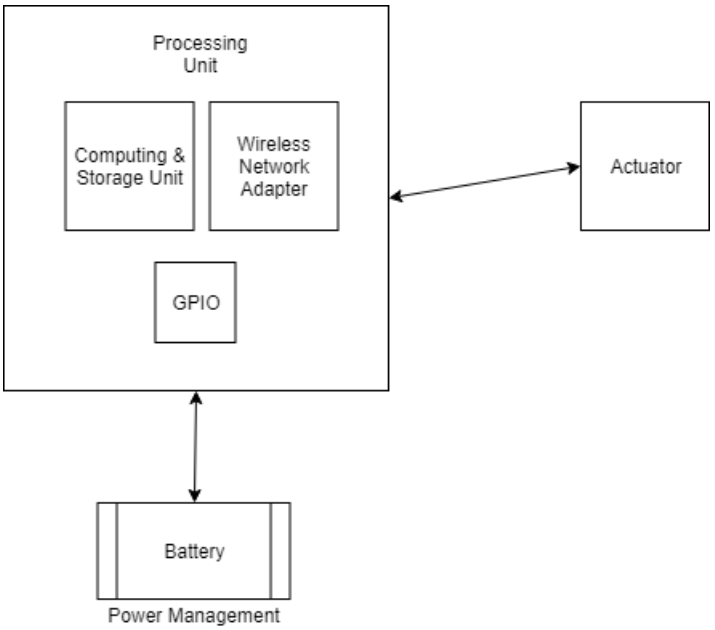


FIGURE 3.3: IoT endpoint Architecture

- **Computing & Storage Unit**: This module performs all the computation and storage of temporary and persistent data.
- **Wireless Network Adapter**: This module provides wireless communication capabilities to the processing unit.
- **GPIO**: This module provides Input Output capability to the processing unit. It allows sensor data to be taken as input and allows output of data to actuators and components.

2. **Power Management**: Since the IoT endpoint needs to be a standalone system, it needs it's own power supply which will be provided by this module.
   - **Battery Charging Circuit**: This module helps in charging the onboard rechargeable battery.
   - **Rechargeable Battery**: The battery provides the power to the various onboard sensors and processors.
3. **Actuator**: The actuator can either be a relay switch or an Infrared LED to change the state of the home appliance.

3.3 Software Architecture

In this section we will discuss the software architecture of the whole system. The architecture has various components as shown in figure 3.4

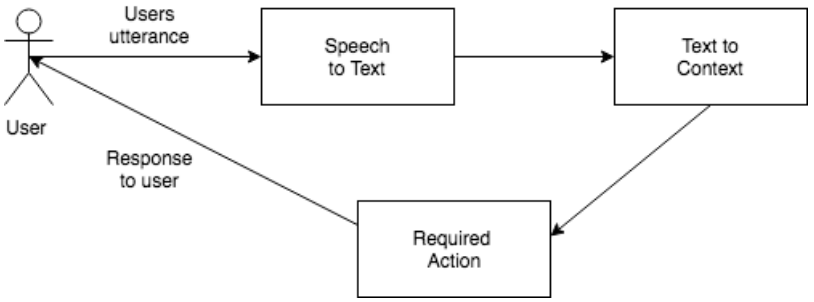


FIGURE 3.4: Software Architecture

The various components in this architecture are :-

1. **User**: The user is the actor of the system who wants some action to be performed.
2. **Speech to text**: The speech to text part of the system takes as input the audio file of the utterance and generates the text representation of the utterance. This text representation is passed as input to the next phase.
3. **Text to context**: This component generates appropriate context from the text. The context includes parameters and action.
4. **Required Action**: Based on the context generated from the previous phase, the required action is performed in this phase. Along with this appropriate response is given to the user.

Chapter 4

Implementation

In this section we will elaborate on the implementation aspect of the project.

4.1 Introduction

As the project consists of both hardware and software development, we distributed the tasks into hardware and software implementation. The hardware implementation involves the creation of the homoto speaker, 3D designing, 3D printing, component selection, integration of various components & embedded software development. The software implementation consists of various phases like data collection, converting speech to text, text to context & finally integrating 3rd party services.

4.2 Technical Requirements

In this section we will highlight the various techinical requirements required on the development side.

4.2.1 Hardware requirement

1. **Processor**: Minimum computing requirement is any processor with higher clock speed & frequency than intel core-i3.
2. **Main Memory**: A minimum of 4 GB RAM is required.
3. **Storage Memory**: A minimum of 500 GB Harddrive is required to develop.

4. **Networking requirement**: The system must have networking capability to connect to the cloud via internet and also connect to the deployment platform wirelessly, hence wireless networking capability is a must.

4.2.2 Software requirement

1. **Python 2.7 environment**: The embedded software for the Homoto Speaker and the cloud server are written in python.

2. **Text Editor**: Basic text editor to write code. In the implementation phase we have used Sublime Text editor.

3. **SSH client**: The client is required to access the RaspberryPi inside the Homoto Speaker via SSH. We have used PuTTy for this purpose.

4. **Arduino IDE**: The embedded software for the IoT endpoints is written, compiled and burned in the memory using Arduino IDE.

5. **Web Browser**: A thin client is required to access the cloud to manage and deploy the main server.

4.2.3 Hardware Implementation

In this section we will go into the depth of the hardware implementation for the Homoto speaker.

4.2.3.1 Component Selection

Selecting the various components based on the requirements obtained during the design phase we had to come up with some actual component that is either readily available in the market or can be made using off the shelf components. The process was challenging and it described in the sections that follow.

1. **Processing Unit - RaspberryPi Zero W**

   As mentioned in the design phase that we need some processing unit that has computing as well as the storage power along with wireless communication capability. The RaspberryPi is the best choice for the project because of it's capability to run an Operating System in a small form factor. There were tight design constraints with respect to the development of the speaker because we wanted it to be as

portable as possible and thus we selected the RaspberryPi Zero W which has the following specs [14] .

(a) **CPU**: 1 GHz ARM11 Broadcom CPU

(b) **Main Memory**: 512 MB

(c) **Storage Memory**: microSD card slot

(d) **Connectivity**: 1 x microUSB, mini-HDMI, 802.11n, Wireless LAN,Bluetooth 4.0 .

(e) **Connectors**: Unpopulated 40-pin GPIO header, SPI, I2C, Camera interface (CSI).

(f) **OS**: Various flavours of Linux like Raspbian, Ubuntu Mate, Windows 10 IoT Core.

2. **Microphone**

As the system is voice operated, we need to get input from the user in the form of voice signals. Thus, the logical choice is to use the microphone to capture the data. What we envisioned was a beamforming microphone that uses an array of mics to detect the sound and these combined omni-directional mics are able to capture a wider audio range [15] . But unfortunately there were no such mic arrays available in the market and implementing them was not possible for us because that would demand higher computing power. As the RaspberryPi doesnot have a dedicatded microphone input we had to use a USB microphone that is compatible with linux. Thus we chose the generic USB mic as shown above 4.1.

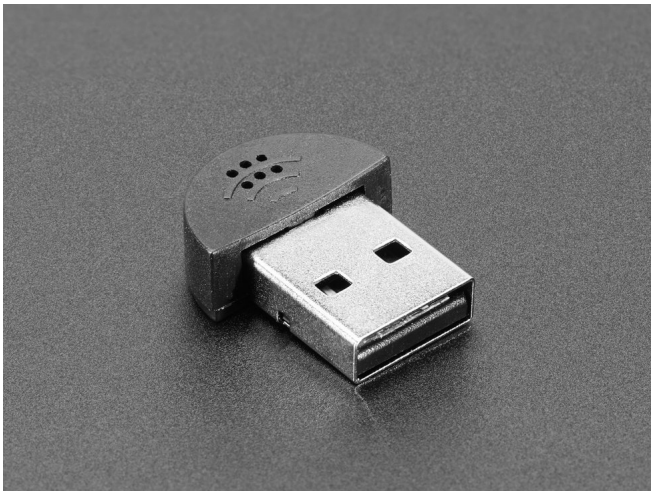


FIGURE 4.1: USB Microphone

3. **Audio Amplifier**

   The RaspberryPi Zero W doesnot have any audio jack to output the audio signals we need to find another way out to add audio output capability to the RaspberryPi. We came across Adafruit MAX98357 I2S Class-D Mono Amp that is compatible with 4-8 Ohm 3 Watt speaker [16]. The amplifier requires us to change some configuration in the Operating system so that the amplifier circuit receives audio signal via GPIO pins of the Rpi via I2S.

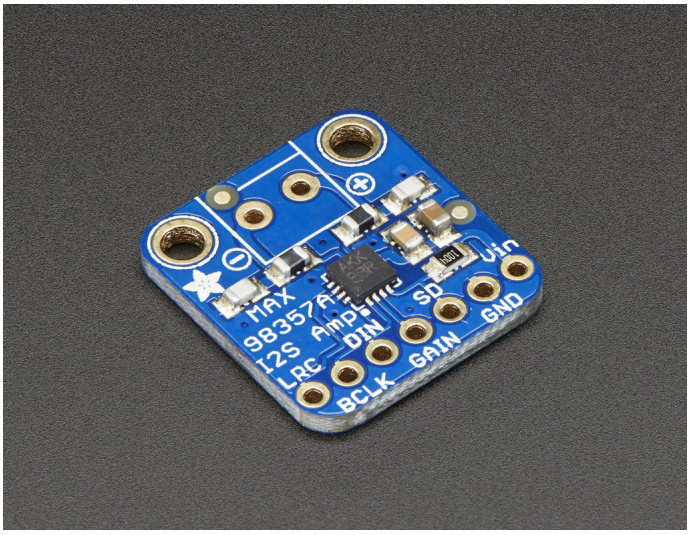


FIGURE 4.2: Adafruit MAX98357 I2S Class-D Mono Amp

4. **Speaker**

   In order to give the user an audio feedback as output, we need a speaker that is loud enough so that the user can hear. Hence we chose the below speaker that is compatible with the audio amplifier selected in the above section and it is a 4 Ohm 3 Watt speaker.

FIGURE 4.3: 4 Ohm 3 Watt Speaker

5. **GPS module**

   A GPS module is used to offer various location based services to the user. The GPS sensor will record the GPS co-ordinates of the user when he requests some location based service and send it to the server. The server will then process the co-ordinates along with the command. A lot of varieties of GPS sensors are available but we wanted something that has a small form factor and interfacable with the RaspberryPi Zero W. Hence we have used UBlox Neo 6M GPS Module.

FIGURE 4.4: UBlox Neo 6M GPS

6. **LED Lights**

   Whenever a user speaks with the system he needs to be given some feedback so that he understands that the system is listening to what the user is saying and processing it, or if any error occurs then the user should be notified both visually and through auditory channel. Because of this reason and to add more visual appeal to the speaker we have used a RGB WS2811 Led strip that operates on 5V output from raspberry pi and allows us the control of each and every led on the strip and also let's us change the colour of individual led rather than the whole strip.

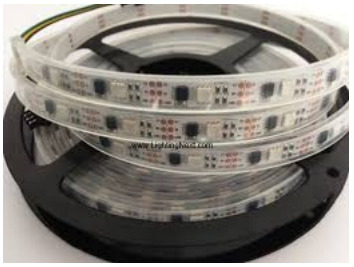


FIGURE 4.5: WS2811 LED Strip

7. **Rechargeable Battery**

   In order to make the speaker standalone and have it's own power supply then we need a rechargeable battery. The battery should have enough capacity to help the speaker last for atleast an hour without external power supply.Hence the battery which we chose was a 3.7 V 1200MaH battery.

FIGURE 4.6: 1200 MaH 3.7V LiPo Battery

8. **LiPo charging circuit**

   We are using a battery charging cicuit that would allow us to charge the battery and also boost the 3.7V of the battery to 5V suitable for the RaspberryPi. Hence we are using the WeMOS D1 mini charging circuit.

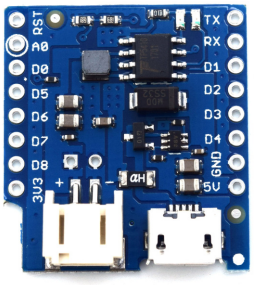


FIGURE 4.7: WeMOS D1 Mini Charging

9. **Node MCU ESP8266**

   Some applications require communicating points at specific location(Ex: IR controller for AC). We have used ESP8266 for the purpose because an inbuilt WiFi module will ease the communication using MQTT, further it has minimum required GPIO pins and small memory unit. It works well with the Arduino IDE, it just requires importing the board. All these advantages at a fraction of price in comparison to a Raspberry Pi or Arduino.

10. **IR transmitter**

    The Infrared light emitting diode (IR LED) is a special purpose LED emitting infrared rays ranging 700 nm to 1 mm wavelength.Our application requires the LED to be used for transmitting IR signals to varies electronic devices(Ex. TV, AC, etc).

11. **IR Receiver SM0038**

    The receiver is used to detect and de-code IR signal from a remote. Its a miniature receiver for Infrared remote control systems. PIN diode and pre-amplifier

FIGURE 4.8: ESP8266 NodeMCU

FIGURE 4.9: Infrared light emitting diode

are assembled on lead frame, the epoxy package is designed as IR filter. The demodulated IR signal can be processed directly by microprocessor.

FIGURE 4.10: SM0038 38KHz IR receiver

12. **2N2222 transistor**

The 2N2222 is a common NPN bipolar junction transistor - BJT used for general purpose low-power amplifying or switching applications. It is designed for low to medium current, low power, medium voltage, and can operate at moderately high speeds. The main function for transistor here is to increase the current flow

through the IR LED.

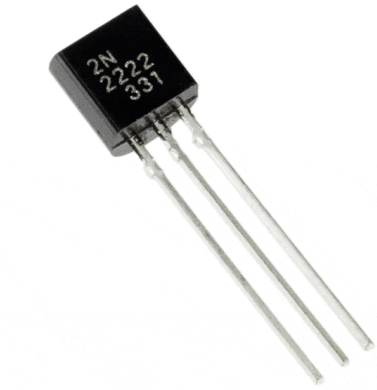


FIGURE 4.11: 2N2222 NPN Transistor

13. **Relay**

In order to switch on and off the various switch controlled devices we need a switch that can be controlled via a micro-controller or similar hence we have used a relay switch.

FIGURE 4.12: 5V 1 Channel Relay

14. **9V Battery**

To power the IoT enpoints we are using a standard 9V Battery.

FIGURE 4.13: 9V Battery

4.2.4 Integration

In this section we will discuss the integration of the components defined in the above section. Again we will divide the integration into 2 parts ie Homoto speaker and the other IoT endpoints.

4.2.4.1 Homoto Speaker

The below figure shows the CAD Design of the various parts of the main speaker shell 4.14 , the speaker grill 4.15 & the bottom cover 4.16 for the speaker that houses the AUX and MicroUSB ports for charging.

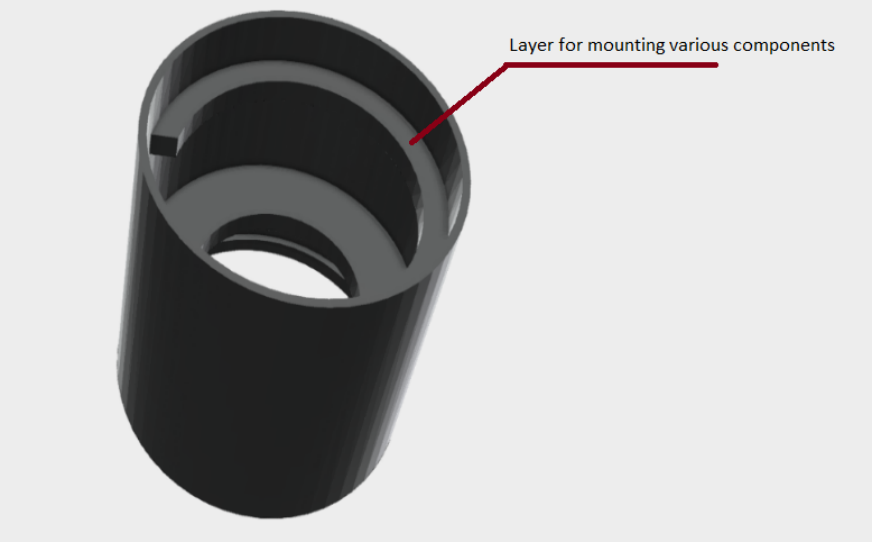


FIGURE 4.14: 3D Design of speaker body

Now we will see the actual final integrated speaker with the various components and where they are placed.

FIGURE 4.15: 3D Design of speaker grill

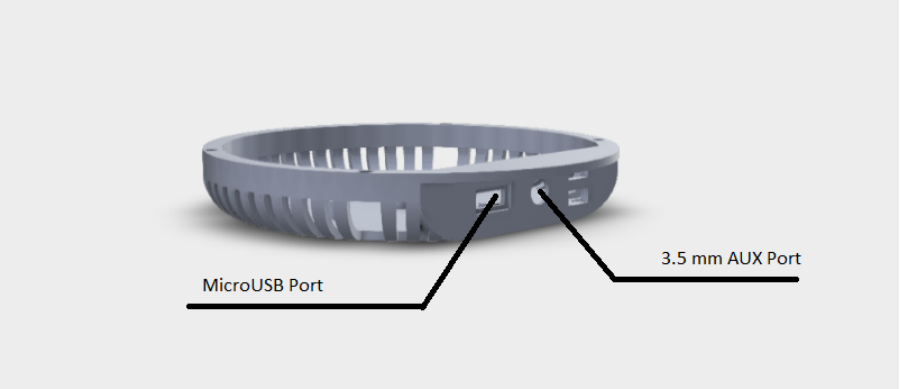


FIGURE 4.16: 3D Design of speaker base cover

FIGURE 4.17: Actual photo of the assembly

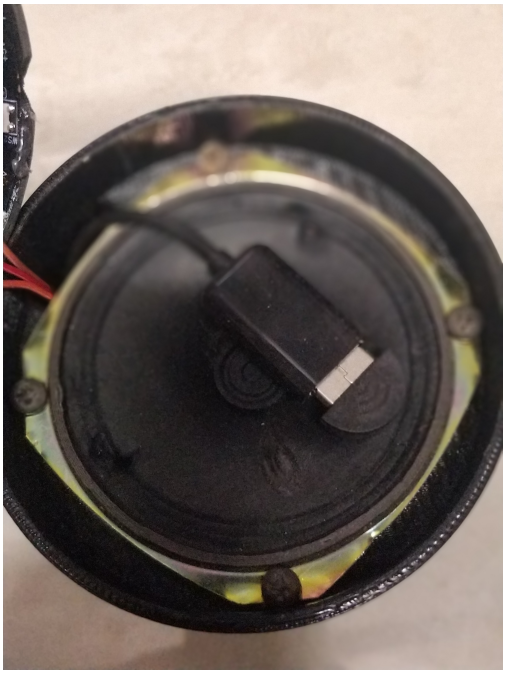


FIGURE 4.18: Actual photo of the mic and speaker

FIGURE 4.19: Actual photo of the speaker grill with led strip

FIGURE 4.20: Actual photo of the assembled RaspberryPi and GPS module

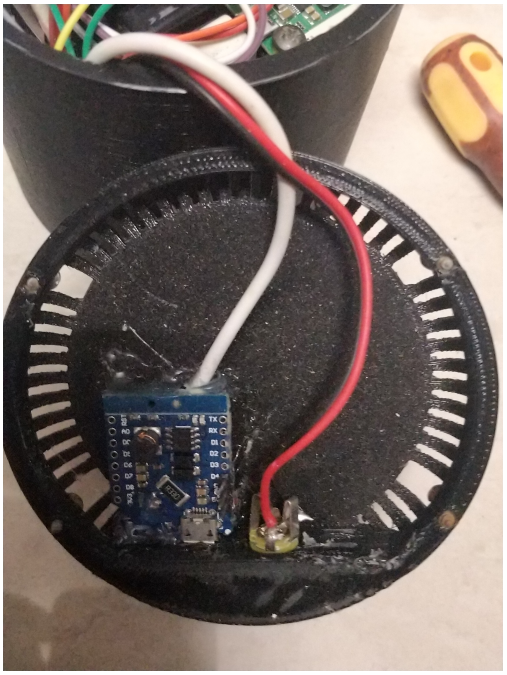


FIGURE 4.21: Actual photo of the assembled charging circuit and 3.5mm Jack

Below is the circuit diagram for the hardware component connection.

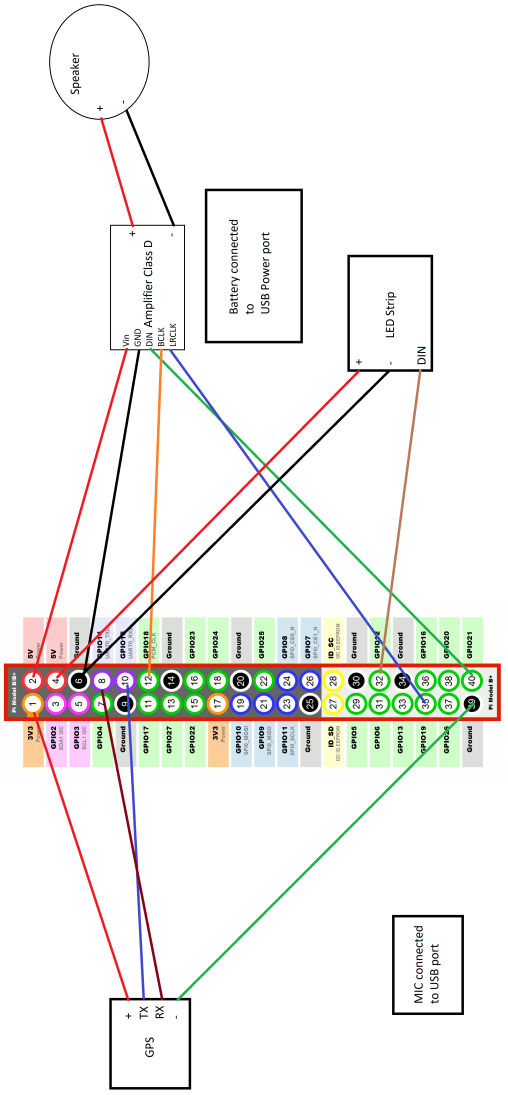


FIGURE 4.22: Circuit Diagram

4.2.4.2 IoT endpoint for IR Control

The below figure involves the circuit diagram of using the ESP8266 as a IR controller 4.23.

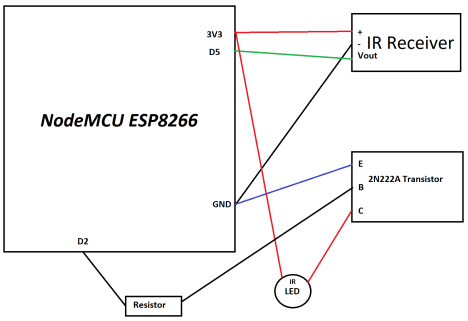


FIGURE 4.23: IR Circuit

4.2.4.3 IoT endpoint for switching device control

The below figure involves the circuit diagram of using the ESP8266 as a switch device controller 4.24.

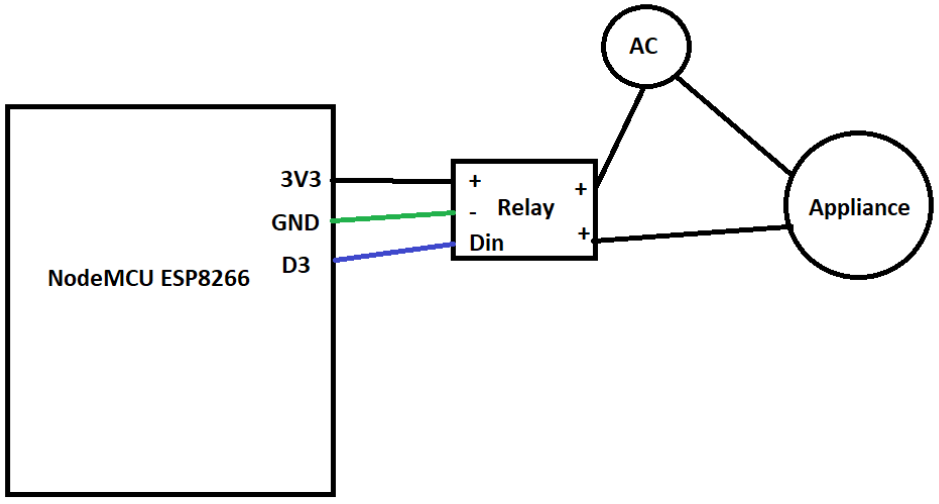


FIGURE 4.24: Relay Circuit

4.2.5 Embedded Software Development

In this section we will elaborate more on the embedded software development. The steps involved in the whole process is described below:-

1. **Installing Raspbian**: The most primitive step in the whole process is to get the pi up and running and thus we need the raspbian installed on the SD card and setting up the WiFi connection on the pi. We followed the steps given on this documentation [17].

2. **Integrating amplifier**: The amplifier uses i2s to get the audio data and thus we need to run some configuration to achieve that. The whole documentation can be found in [18]. The circuit connections are listed below:-

   (a) Amp Vin to Raspbery Pi 5V

   (b) Amp GND to Raspbery Pi GND

   (c) Amp DIN to Raspbery Pi #21

   (d) Amp BCLK to Raspbery Pi #18

   (e) Amp LRCLK to Raspbery Pi #19

3. **Implementing hotword detection& mic setup**: We are using Snowboy Hotword detection engine and the steps to install can be found in [19].

4. **Installing MQTT**: To communicate with the IoT endpoints we are using the MQTT protocol, the steps for installing it are in [20].

5. **Installing Apache and PHP**: In order to use a hotspot and host a local website to get user WiFi credentials we need apache and PHP so we have installed it.

6. **Integrating WS2811 LED**: The LED lights require some dependent packages to be installed in order to work in python environment. Ref [21]

7. **GPS setup**: The steps to integrate the GPS are referred from [22].

Once all the components and the basic dependencies have been setup we can move onto the software development phase wherein we will create the main speaker code that handles the user input and output along will controlling of speaker, LED & GPS units. The code is written in the form of systemd services which can be started automatically upon boot, reboot, poweroff, etc.

The reference for creating systemd files is mentioned [23]

After the code is running successfully we want to create a hotspot so that upon initial setup the user can connect to the hotspot created by the raspberrypi and provide WiFi SSID and Password to the pi so that it can connect to the WiFi and this will be a one time process. This process will be elaborated in much detail in the system usage chapter. The process to create hotspot was referred from [24].

The code for the IoT endpoints was written in Arduino IDE and the basic communication protocol was MQTT. The code is written in such a way that each endpoint has it's own unique identity which needs to be registered with the pi before usage, so each endpoint once connected to the local WiFi network sends a request to regsiter itself with the pi (Homoto Speaker) and this will be a one time process. After the registration each endpoint will begin it's own setup process. For example if you are using an IR control endpoint then the user needs to train the device with his/her IR remote controls, once the process is done the system is ready to use. The process of registering a device and the subsequent steps will be elaborated in the system usage chapter.

4.3 Software Implementation

4.3.1 Data Collection

Initially the plan was to train a machine learning model to perform the task of Speech to Text and for any machine learning task data is an integral component. So following were the methods which were used to collect raw voice samples of Indian accent.

4.3.1.1 CMU Arctic Database

The CMU Arctic databases were designed for the purpose of speech synthesis research at the Language Technologies Institute(LTI) at Carnegie Mellon University. These single speaker speech databases were carefully recorded under studio conditions and consist of nearly 1150 phonetically balanced English utterances. They are distributed as free software, without restriction on commercial or non-commercial use. These utterances were selected from out-of-copyright texts from Project Gutenberg (Project Gutenberg offers over 57,000 free eBooks). So one of these databases provides around 1132 utterances spoken by an Indian English male speaker of approximately 3-4 seconds each. The transcripts of voice samples are also provided.

4.3.1.2 Using TTS applications which provide Indian Voice as Output

A TTS or Text to Speech application is an application which creates a spoken sound version of given text. There are few TTS applications such as IndianTTS etc. which provide Indian accent voice data when provided text input.

4.3.1.3 Manual Collection of raw voice samples

Following were the few methods with which we had planned to collect more voice samples :-

1. Android App

   An Android application where the user would be asked to speak some predefined commands and then record their voice samples and store it on cloud.

2. Online Platforms

   There are various online platforms such as Coursera, YouTube, Khan Academy etc which provide educational videos for learning. These platforms have lecturers who deliver these videos and transcripts for these videos are also provided. So voice samples along with the captions or transcripts of these lectures which are given by Indian instructors can be recorded.

4.3.1.4 Conclusion

In the end, we decided not to use a machine learning model and instead used an API for the task of speech to text analysis because of the following reasons :-

1. The API(Liv.ai) gave us results with much better accuracy than both the pre-trained models of CMUSphinx and Mozilla DeepSpeech.

2. Also to build a custom machine learning model, a lot of data is required and there is no readily available open-source dataset of voice samples in Indian accent.

3. Further,we couldn't generate enough dataset of voice samples using the above mentioned data collection techniques.

4.3.2 Speech to Text

This part includes all the various methods that have been used to convert voice sample into text :-

1. CMU Sphinx
2. Mozilla DeepSpeech
3. Liv.ai

4.3.2.1 CMU Sphinx

Introduction

The CMU Sphinx toolkit is a leading speech recognition toolkit with various tools used to build speech applications. CMU Sphinx contains a number of packages for different tasks and applications.

- Pocketsphinx - lightweight recognizer library written in C.
- Sphinxbase - support library required by Pocketsphinx.
- Sphinx4 - adjustable, modifiable recognizer written in Java.
- Sphinxtrain - acoustic model training tool.

Characteristics of CMU Sphinx

1. Algorithm/Model : Hidden Markov Model
2. Open Source : Yes
3. Programming Language : Java
4. Supported Languages : English
5. Offline Support : Yes

The CMU Sphinx contains these three models :-

An acoustic model contains acoustic properties for each senone. There are context-independent models that contain properties (the most probable feature vectors for each phone) and context-dependent ones (built from senones with context).

A phonetic dictionary contains a mapping from words to phones. This mapping is not very effective. For example, only two to three pronunciation variants are noted in it. However, its practical enough most of the time. The dictionary is not the only method for mapping words to phones. You could also use some complex function learned with a machine learning algorithm.

A language model is used to restrict word search. It defines which word could follow previously recognized words (remember that matching is a sequential process) and helps to significantly restrict the matching process by stripping words that are not probable. The most common language models are n-gram language modelsthese contain statistics of word sequencesand finite state language modelsthese define speech sequences by finite state automation, sometimes with weights.

Implementation

A Java class was written where input was provided as a wav file. When an audio sample with audio "coffee" was provided as input, the output text generated was "kids kids". Also, the Java class was run on various other audio files were tested and the results were not at all satisfactory.

Input provided: **coffee**

Output given: **kids kids**

FIGURE 4.25: Output "kids kids" given by CMU Sphinx where input was "coffee"

4.3.2.2 Mozilla DeepSpeech

Project DeepSpeech is an open source Speech-To-Text engine. It uses a model trained by machine learning techniques, based on Baidu's Deep Speech research paper. Project DeepSpeech uses Google's TensorFlow project to make the implementation easier.

Characteristics of Mozilla DeepSpeech

1. Algorithm/Model : Deep Neural Networks
2. Open Source : Yes

3. Programming Language : Python
4. Supported Languages : English
5. Offline Support : No

There are 2 ways in which we can implement the this model

1. Making use of the pre-trained model
2. Training the model using our data

4.3.2.3 Liv.ai

Liv.ai is a speech API which enables users to convert speech-to-text by using Powerful Neural Network Models with exceptional accuracy and minimal latency. The API recognizes 9 major Indian Languages - English, Hindi, Bengali, Punjabi, Marathi, Gujarati, Kannada, Tamil and Telugu. It works with most accents and performs remarkably well in noisy environments.

Characteristics of Liv.ai

1. Algorithm/Model : Deep Neural Networks
2. Open Source : No
3. Programming Language : Not Known
4. Supported Languages : Indian languages such as English, Hindi, Tamil etc
5. Offline Support : No

Implementation

A python script was written where input was provided as a wav file and a request is made to Liv.ai API as per the documentation [25]. The API then sends the translated text along with its confidence score. The python script was run on various other audio files were tested and the output provided were really good.

Input provided: **get me the current weather in mumbai**

Output given: **get me the current weather in mumbai**

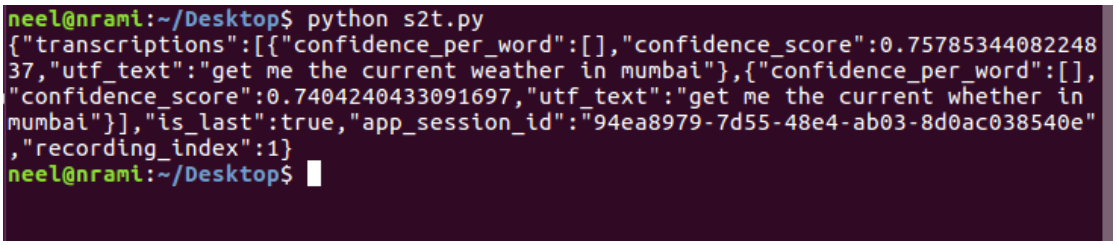


FIGURE 4.26: Output given by Liv.ai

4.3.2.4 Conclusion

Liv.ai was chosen by us for the task of Speech to Text analysis for the following reasons :-

1. CMUSphinx's pre-trained model wasn't even able to recognize basic words.
2. Mozilla DeepSpeech's pre-trained model gave better accuracy than CMUSphinx's pre-trained model but it would always fail to recognize a word or two in some sentences.
3. Also Mozilla DeepSpeech's pre-trained model would require more than 1GB of RAM to run.

4.3.3 Text to Context

This part includes all the methods that were considered to get meaningful context from text.

1. Rule-Based
2. Dialogflow

4.3.3.1 Rule-based

In rule based method commands and their variations are written in a grammar file of JSGF(Java Speech Grammar Format) format. For each command a separate grammar file will be written. Then after getting the command in text form from the previous phase by applying speech to text the obtained text will be be parsed through each of the grammar files. Depending on which grammar file parses the text, the corresponding command number will be obtained and appropriate action will be taken.

4.3.3.2 Dialogflow

Dialogflow is a chatbot building service provided by Google. It lets developers build chatbots for various services. We start off by building an agent. An agent is nothing but your projects name. Then we create intents for each type of commands. Lets consider an example:-

1. A user asks the your project's agent "What's the weather supposed to be like in San Francisco tomorrow?"
2. In Dialogflow, an intent houses elements and logic to parse information from the user and answer their requests. For the agent to understand the question, it needs examples of how the same question can be asked in different ways. Developers add these permutations to the Training Phrases section of the intent. The more variations added to the intent, the better the agent will comprehend the user.
3. The Dialogflow agent needs to know what information is useful for answering the user's request. These pieces of data are called entities. Entities like time, date, and numbers are covered by system entities. Other entities, like weather conditions or seasonal clothing, need to be defined by the developer so they can be recognized as an important part of the question.
4. Dialogflow sends this information to your webhook, which subsequently fetches the data needed (per your development). Your webhook parses that data, determines how it would like to respond, and sends it back to Dialogflow.

4.3.3.3 Conclusion

We have used Dialogflow in the final implementation because of the following reasons:-

- Rule based system was too restrictive and it cannot incorporate variations of utterances. Dialogflow has the capability to understand variations of utterances by using ML.
- Also it would be difficult to extract parameters in rule based system. Dialogflow makes this possible by using entities.

So what dialogflow essentially does is maps each user uttered text to an intent. So from this we know what action the user has asked to be performed. Also in each text dialogflow extracts entities and gives it to us. Entities are nothing but parameters of a

sentence. for example if user says 'What is the weather like in mumbai today?'. Here Mumbai are today are parameters. So in this way we can get the context from text

4.3.4 Third Party Services

We have used various third party APIs in our implementation for getting various services such as weather information, news, nearby places information etc. The APIs are as listed below.

4.3.4.1 News Api

We have used newsapi.org to get the latest news. News can be searched either by passing a topic or without topic. The documentation is given in [26]. When topic is not given by the user top headlines are returned to the user.
We have used two endpoints of this API:-

- https://newsapi.org/v2/top-headlines
  This endpoint is used when the user does not provide any topic. In this case headlines are given. We pass two parameters to this endpoint-language and country.
- https://newsapi.org/v2/everything
  This endpoint is used when the user provides the topic to search news on. In this case recent news on the given topic are given. We pass five parameters to this endpoint-from parameter, to parameter, language, sort by and topic parameter.

4.3.4.2 Cabs Api

We have used uber api to get the price estimate of a cab. The documentation is given in [**?**]. Users location is taken from the device's gps module.
We have used one endpoint of this API:-

- https://api.uber.com/v1.2/estimates/price
  We pass four parameters to this endpoint-start latitude, start longitude, end latitude, end longitude. The end latitude and longitude is taken from google maps api.

4.3.4.3 Places Api

We have used google places to get information on nearby places such as railway stations, atms, petrol stations, cinemas, restaurants, hospitals. The documentation is given in [27]. Users location is taken from the device's gps module.
We have used one endpoint of this API:-

- https://maps.googleapis.com/maps/api/place/nearbysearch/json
  We pass four parameters to this endpoint-location, rankby, type and key. The type is the place about which user wants to get information.

4.3.4.4 Products Api

We have used amazon's affiliate api to get prices of different products. The user will provide the product name for which prices are to be given.User is given the top three results. The documentation is given in [**?**].
We have used one endpoint of this API:-

- http://webservices.amazon.in/onca/xml
  We pass four parameters to this endpoint-timestamp, operation, searchindex and keywords. The keywords field is the product the user wants to search for. The operation field is specified as 'ItemSearch'

4.3.4.5 Weather Api

We have used openweathermap api to get various weather information. The user will provide what he wants to know. We have python wrapper pyowm for the api. The documentation of the wrapper is given in [28].
We have used eight functions of this wrapper:-

- get_temperature()
  This method is used to get the temperature. We pass celsius option to this method and we pass the location and time objects.

- get_humidity()
  This method is used to get the humidity. We pass the location and time objects to this method.

- get_wind()
  This method is used to get the wind speed. We pass the location and time objects to this method.

- will_have_fog()
  This method is used to get the fog prediction. We pass the location and time objects to this method.

- will_have_sun()
  This method is used to get the sun prediction. We pass the location and time objects to this method.

- will_have_clouds()
  This method is used to get the cloud prediction. We pass the location and time objects to this method.

- will_have_snow()
  This method is used to get the snow prediction. We pass the location and time objects to this method.

- will_have_rain()
  This method is used to get the rain prediction. We pass the location and time objects to this method.

4.3.4.6 Geocode Api

We have used google's geocode api to get location information such as latitude and longitude based on a address. The user will provide the address for which location information is needed. This api is used in the uber api to get latitude and longitude based on the address which user provides. The documentation is given in [29].
We have used one endpoint of this API:-

- https://maps.googleapis.com/maps/api/geocode/json
  We pass two parameters to this endpoint-address and key. The address is provided by the user.

4.3.4.7 HTML parser

We have used the Beautiful Soup python library to parse the HTML data in order to get the required links from youtube for playing songs. User passes the song he wants to hear. Then we get the web page using urllib2 by passing the song in search_query

parameter. After that the links are generated using Beautiful Soup parser. The documentation is given in [30]

Chapter 5

System usage

In this section we will discuss the system interaction with the user when he uses the product out of the box or in certain cases where the WiFi of the user has changed.

5.1 Setup of device

5.1.1 Setup of Homoto Speaker

1. **Case 1: First time setup**:

   (a) When the user powers up the device for the first time, the speaker will play a greeting message stating that "Homoto is not connected to the internet, please connect to the hotspot to configure it." along with a visual of red fading led.

   (b) After initializing the hotspot the sytem will play audio "Please search for Homoto in your WiFi settings and enter the password mentioned on me" .

   (c) The user will see Homoto named WiFi SSID in his smartphone/tablet/laptop WiFi settings as shown below. Also, he will be prompted to enter a password (the default pass is 'homoto@123') after which he has to connect open any web browser and enter the address as 10.0.0.2 .

   (d) He will be shown a WiFi configuration page where he has to enter the WiFi SSID and Password after which the system will reboot and try to connect to the entered WiFi.

   (e) Upon successful connection the user will be greeted with a green fading led and a message "Homoto is now listening for your commands" will be played.

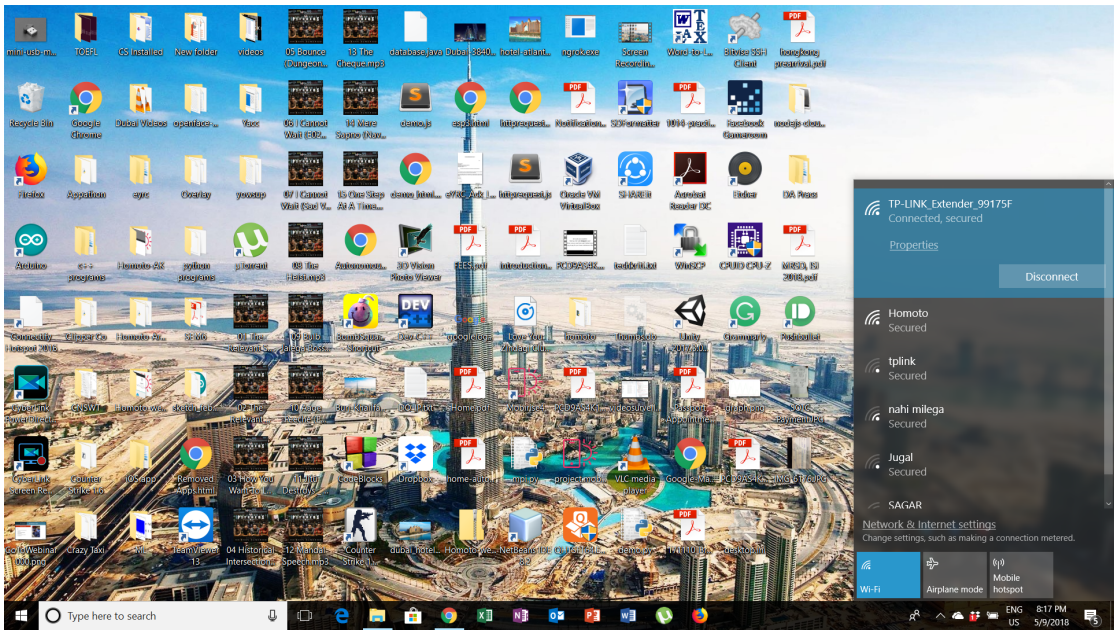


FIGURE 5.1: Homoto Hotspot visible in the WiFi settings

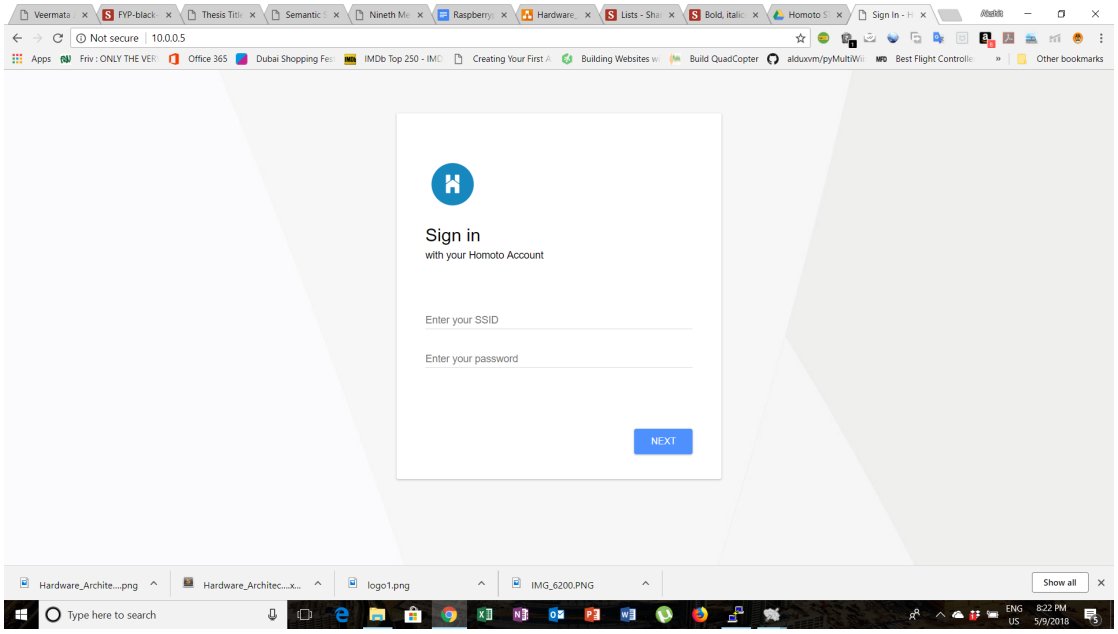


FIGURE 5.2: User enters his WiFi's SSID and Password

2. **Case 2: Setup of the WiFi in case of change in settings**: There maybe a case when the user has shifted to a new place where he has a different access point in that case he has to simply restart the speaker and then follow the steps as above.

5.1.2 Setup of IoT endpoint

1. **Case 1: First time setup**:

   (a) When the user powers the device for the first time, he has to connect to the hotspot created by the Homoto IoT endpoint. The process is shown in the screenshot below.

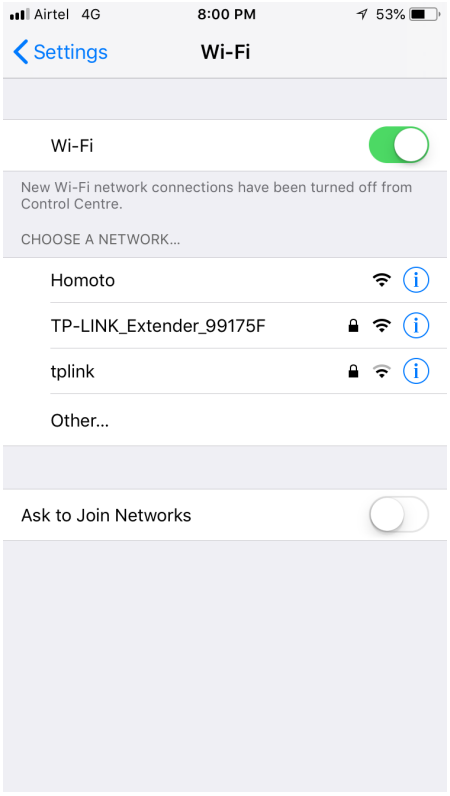


FIGURE 5.3: Homoto Hotspot visible in the WiFi settings

(b) The user will automatically be redirected to the WiFi captive portal as shown in screenshot 5.4 where he will enter his SSID and Password as shown in the screenshot 5.5

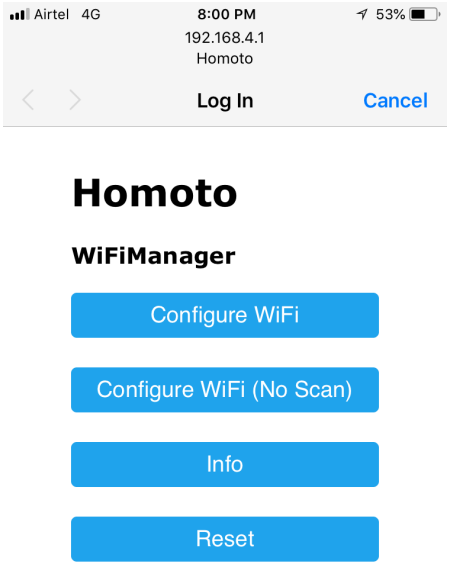


FIGURE 5.4: User shown the menu

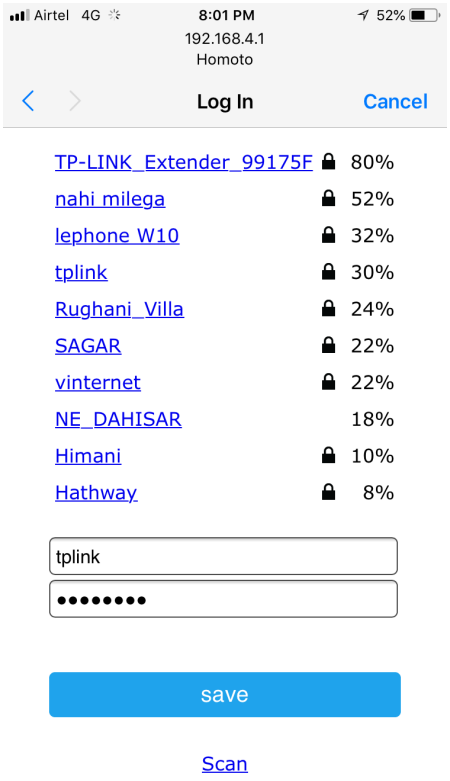


FIGURE 5.5: User selects his home WiFi network and enters the password

(c) Upon successful connection with the WiFi the system will reboot.

(d) After which he will be told by the speaker on the further steps of setting up the device.

(e) Upon successful setup the user will be notified by the speaker as shown in the screenshot 5.6

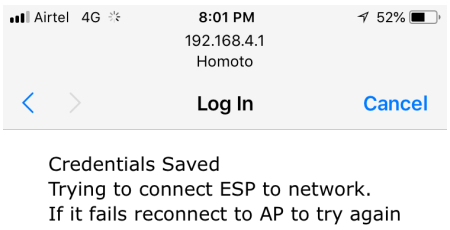


FIGURE 5.6: Acknowledgement for connection with WiFi

2. **Case 2: Setup of the WiFi in case of change in settings**:

   There maybe a case when the user has shifted to a new place where he has a different access point in that case he has to simply connect to the hotspot of the IoT endpoint and then follow the steps as above. Or else he can look on the led blink status on the device to know if the system is connected or not.

5.2 Usage of functionalities

This part includes all the functionalities that have been implemented in the device. Functionalities have been divided into two parts :-

1. IOT functionalities.
2. Third party functionalities.

5.2.1 IOT functionalities

It include functionalities which involve controlling IOT enabled traditional appliances such as fans, lights, television, air conditioners etc.We have implemented two IOT functionalities :-

1. Controlling switch based devices such as fans, lights, bulbs, etc.
2. Controlling IR enabled devices such as AC or TV.

5.2.1.1 Controlling switch based devices such as fans, lights

This command will be used to switch on or switch off the devices.
Example user inputs :-

- Can you please switch on the light.
- Switch on the light.
- Can you please switch off the light.
- Turn the fan on.

The following outputs will be given on successful execution :-

- The fan has been switched on .
- The light has been switched off.

On unsuccessful execution following output will be given :-

- Sorry can't respond at this time. Please try again later

5.2.1.2 Controlling IR enabled devices such as AC, TV

This command will be used to switch on or switch off Ir enabled devices such as AC,TV .
Example user inputs:-

- Can you please switch on the AC.

- Switch on the TV.
- Increase volume of TV.
- Increase the temperature.

The following outputs will be given on successful execution :-

- The AC has been switched on.
- The volume has been decreased.

On unsuccessful execution following output will be given :-

- Sorry can't respond at this time. Please try again later

5.2.2 Third party functionalities

Third party functionalities include functionalities which involve getting useful information from third party APIs and providing them to the user in a convenient form. We have implemented seven third party functionalities :-

1. Doing small talk with users.
2. Getting latest news.
3. Getting nearby places.
4. Getting prices of products.
5. Playing songs.
6. Getting weather information.
7. Getting ride estimates of a cab.

5.2.2.1 Doing small talk with users

Using this command, user can do basic conversations with the device. User can ask questions such as who are you, good morning etc.
An example command is as below :-
Are you busy?
To this command Homoto will respond as:-
Nah, tell me what can I do for you?
Some of the other example commands which can be executed as follows:-

- How old are you?
- Good evening!
- Nice to see you!
- What's up?
- Are you there?

5.2.2.2 Getting latest news

Using this command the user can get the latest news. News can be obtained in two ways-either by passing a topic or without using a topic. In case of no topic mentioned, the top headlines are given. In each case top three news is given to the user.
Some example commands as follows :-

- Get me news on Virat Kohli
- Get me the latest news
- Can you get me news on Bitcoin?
- What's happening around the World today?

Below is an example output for the command 'get me the latest news' :-
Here is the recent news on for today. Uttarakhand snowfall: Ex-CM Harish Rawat, other Congress leaders stranded in Kedarnath. Isha Ambani gets engaged to Anand Piramal: See pics. Karnataka Election Campaign LIVE Updates: PM Modi Addresses Election Rally In Vijayapura, Attacks Congress

5.2.2.3 Getting nearby places

Using this command the user can get the nearby places information of hospitals, restaurants, movie theaters, gas stations, atms, train stations. User will get the addresses of corresponding places.User will get the top three results in each case
Some example commands as follows :-

- where are the nearby railway stations
- get me nearby restaurants

- nearby movie theaters
- what are the nearby atms

Below is an example output for the command 'where are the nearby railway stations' :-
Here are the nearby atm. Vijaya Bank ATM at Dinesh Mahal, 211-E, 400019, Doctor Baba Saheb Ambedkar Road, Matunga, Mumbai. Vijaya Bank at 211-E, Dinesh Mahal, Dr Ambedkar Road, Matunga East, Mumbai. Bank Of Maharashtra ATM at 101, Road Number 32, Wadla Village, Wadala, Mumbai.

5.2.2.4 Getting prices of products

Using this command the user can get the prices of a product from amazon.in [31]. User will get the top three results in each case
Some example commands as follows :-

- tell me the price of dell Xps laptop
- get me price of iphone X from amazon
- can you get me the price of a pen drive from amazon?
- get me price of umbrella

Below is an example output for the command 'how much will it cost me to buy an alienware laptop?' :-
Here are the top three results from amazon. The price of Logitech G402 Hyperion Fury Ultra Fast FPS Gaming Mouse (Black) is rupees 2,479. The price of Alienware AW17R4-7003SLV-PUS 17" Gaming Laptop (7th Generation Intel Core i7, 8GB RAM, 256GB SSD + 1TB HDD, Silver) with NVIDIA GTX 1060 is rupees 164,820. The price of Dragonwar ELE-G9 Thor BlueTrack and Blue Sensor Gaming Mouse with Macro function is rupees 999.

5.2.2.5 Playing songs

Using this command the user can request for a song to be played. The song which user requested will be played from youtube.com [32] :-

- I would like to listen to despacito

- play sorry
- play me the shape of you

The requested song will be downloaded from youtube and then played.

5.2.2.6 Getting weather information

Using this command the user can get weather information such as getting temperature, humidity, wind speed, overall weather forecast. Also user can get to know whether it is going to rain, will it snow, is it sunny, is it foggy.Weather information for upto 5 days can be obtained.
Some example commands as follows :-

- Weather forecast
- What is the weather supposed to be
- What is the weather like?
- what is the current temperature
- will it rain on monday?

Below is an example output for the command 'what is the weather like today?' :-
The temperature in Mumbai is 33.01 degree celsius.The humidity in Mumbai is 62 percent.The wind speed in Mumbai is 4.1 metres per second. Weather information for upto 5 days can be obtained.

5.2.2.7 Getting ride estimates for a cab

Using this command the user can get price estimate of a cab for a given destination. Also the user can get to know how far away is the cab.In this case the users location is taken from the device's gps module. The cab estimates are given from uber.
Some example commands as follows :-

- how much will it cost for a cab to veermata jijabai technological institute
- price estimate for a cab to mahavir nagar
- get me a price estimate of a cab to dadar station

Below is an example output for the command 'what will be the cost for a cab to bandra kurla complex' :-

The ride estimate of an Uber Go to bandra kurla complex is Rs 176 to Rs 216. The cab is 3 minutes away.

Chapter 6

Results and Problems

6.1 Results

Following are the results we have obtained:-

- User can control switch based devices such as lights and fans
- User can get weather information
- User can get nearby places information
- User can get prices of products
- User can get price estimates for booking a cab
- User can listen to songs
- User can get recent news
- User can control ir based devices such as ac,tv etc So in this way traditional devices can be made smart using our system.Also this device supports Indian accent voices.

6.2 Current Problems

Here are some of the current problems that we are facing :-

1. Sometimes the device falsely identifies the hot word "Homoto". Even if the user doesn't say the hot word "Homoto", the user gets a visual feedback in the form of LED ring. (The glowing of the LED ring is an indication of the device recognizing the hot word "Homoto").
2. GPS doesn't work indoors because of low power antenna.
3. Raspberry Pi doesn't have the processing power to perform multiple tasks simultaneously.
4. Sometimes there is a delay between a user request such as "give me latest news" and the device response.
5. All third party APIs which have been used by us are on free basis. So these APIs provide limited calls. For ex. Liv.ai provides 1000 calls per day if a free plan is used.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

In this project, we have built a home assistant device which is being controlled by voice. We have developed this device to control household appliances such as lights and television. The device will help the user to turn on and turn off the lights etc. by giving voice commands like "turn on the lights". By giving the user a chance to control traditional devices by this home assistant, we are adding intelligence into these devices.

Further we have integrated third party services such as Uber, Amazon etc. to provide various functionalities like getting latest news, getting ride estimate, getting information on a product from Amazon, playing a song and getting information of nearby ATMs, restaurants and other places etc.

7.2 Future Scope

7.2.1 Conversations

Currently our device handles single-line queries. As the Artificial Intelligence(AI) systems driving speakers become more capable of understanding human conversation, its likely that these single-line queries will become more of a conversation, prompting users with follow-up questions, and encouraging more subjective, open-ended inquiries and responses.

7.2.2 Local networks

We can reduce latency with more local processing units. While this will only adds a few seconds to the average users interaction, that speed will become more relevant as users demand more from their devices. With more devices, local networks will help make the flow faster.

7.2.3 Monitoring and Analytics

In future we will add the ability to log commands, which will enable the device to track our lifestyle habits and make recommendations based on those habits, such as recommending new foods or suggesting changes to our routines to save time.

Bibliography

[1] Automatic online evaluation of intelligent assistants. . URL https://people.cs.umass.edu/~jpjiang/papers/www15_cortana_sat_final.pdf.

[2] Amazon echo introduction and help. . URL https://www.amazon.com/gp/help/customer/display.html?nodeId=201601770.

[3] Amazon echo features and invites. . URL http://androroot.com/2018/02/buy-amazon-echo-in-india.html.

[4] Alexa overview. . URL https://developer.amazon.com/docs/alexa-voice-service/api-overview.html.

[5] Amazon echo commands. . URL https://www.cnet.com/how-to/amazon-echo-the-complete-list-of-alexa-commands/.

[6] Google home a better virtual assistant. . URL https://www.ft.com/content/8ce022e8-ac20-11e6-ba7d-76378e4fef24.

[7] Googles speech recognition is now almost as accurate as humans. . URL https://9to5google.com/2017/06/01/google-speech-recognition-humans/.

[8] The complete list of services with google home support (updated). . URL https://www.androidauthority.com/google-home-services-749968/.

[9] The complete list of google home commands so far. . URL https://www.cnet.com/how-to/google-home-complete-list-of-commands/.

[10] Apple announces smart home audio speaker homepod. . URL http://variety.com/2017/digital/news/apple-introduces-homepod-a-smart-home-audio-speaker-1202454010/.

[11] Home pod - technical specifications - apple. . URL https://www.apple.com/homepod/specs/.

[12] What siri is? . URL http://www.cortanareachit.com/siri.html.

[13] The complete list of siri commands. . URL https://www.cnet.com/how-to/the-complete-list-of-siri-commands/.

[14] Raspberrypi zero w specs. . URL https://www.androidauthority.com/raspberry-pi-zero-w-review-756498/.

[15] Amazon beamforming mic. . URL https://developer.amazon.com/alexa-voice-service/dev-kits/amazon-7-mic/.

[16] Adafruit max98357 i2s class-d mono amplifier documentation. . URL https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/overview.

[17] Installing os image for raspberrypi. . URL https://www.raspberrypi.org/documentation/installation/installing-images/.

[18] Adafruit amplifier documentation. . URL https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp/raspberry-pi-usage.

[19] Snowboy documentation. . URL http://docs.kitt.ai/snowboy/.

[20] Installing mqtt on ubuntu documentation. . URL https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-the-mosquitto-mqtt-messaging-broker-on-ubuntu-16-04.

[21] Ws2811 led software integration. . URL https://learn.adafruit.com/neopixels-on-raspberry-pi/overview.

[22] Setup of gps. . URL https://github.com/SlashDevin/NeoGPS.

[23] Creating systemd files. . https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/ and https://bbs.archlinux.org/viewtopic.php?id=213751 and https://bbs.archlinux.org/viewtopic.php?id=207465 and https://www.reddit.com/r/raspberry_pi/comments/15m3cy/play_startup_sound_systemd/ and https://unix.stackexchange.com/questions/126009/cause-a-script-to-execute-after-networking-has-started and https://www.freedesktop.org/wiki/Software/systemd/NetworkTarget/.

[24] Uber website. . URL https://www.uber.com/en-IN/.

[25] Liv.ai api documentation. . URL https://liv.ai/api/speech/index.html.

[26] Newsapi.org documentation. . URL https://newsapi.org/docs.

[27] Google places api documentation. . URL https://developers.google.com/places/documentation.
)]Reference10 Amazon affilaite api documentation. . URL https://docs.aws.amazon.com/AWSECommerceService/latest/DG/CHAP_ApiReference.html.

[28] pyowm api documentation. . URL https://pyowm.readthedocs.io/en/latest/.

[29] Google geocode api documentation. URL https://developers.google.com/maps/documentation/geocoding/intro.

[30] Beautiful soup 4 documentation. URL https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[31] Amazon.in. URL https://amazon.in.

[32] youtube.com. URL https://youtube.com.