

# **Personal Assistant System Using Robot**

**A Project Submitted in Partial Fulfillment of the Requirements for the**

**Degree of**

**Bachelor of Science in Computer Science and Engineering**

**by**

**Md Ziaur Rahman**

**CSE 056 06849**

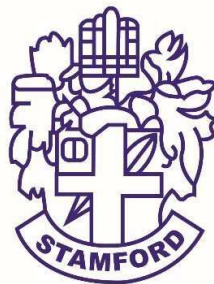
**&**

**Fokhrul Islam**

**CSE 056 06885**

**Supervised by: Asma-ull-Hosna**

**Assistant Professor**



**Department of Computer Science and Engineering**

**STAMFORD UNIVERSITY BANGLADESH**

**January 2019**

# **Abstract**

This paper presents a design and implemented of a prototype personal assistant robot (PAR). The goal for the development of PAR is to investigate how an intelligence machine can assist and interact with a person in a smart manner. The PAR prototype consists of a mobile platform with a 360 degree-of-freedom manipulator. Currently, implemented functions include human face recognition and tracking, and fuzzy behavior-based navigation. Experimental studies of these functions show that the robots demonstrate autonomous interactions as well as assistant functions in a natural indoor environment.

# Approval

The project report “A Prototype of Personal Assistant Robot System using Hardware Project” submitted by Md Ziaur Rahman ID: CSE 056 06849, Fokhrul Islam ID: CSE 056 06885, to the Department of Computer Science & Engineering, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc.) in Computer Science & Engineering and as to its style and contents.

Board of Examiner’s Name, Signature and Date:

.....	.....	.....
Md Shahedul Islam	Tamjid Rahman	Israt Jahan Mouri
<b>(Board Member 1)</b>	<b>(Board Member 2)</b>	<b>(Board Member 3)</b>
Date: 24-01-2019	Date: 24-01-2019	Date: 24-01-2019

Supervisor’s Signature and Date:

.....

**Supervisor Name**

**Asma-ull-hosna**

Date:24-01-2019

# Declaration

We, hereby, declare that the work presented in this Project is the outcome of the investigation performed by us under the supervision of Asma-ull-hosna Assistant Professor, Department of Computer Science & Engineering, Stamford University of Bangladesh. We also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Signature and Date:

.....

**Student Name:**

Md Ziaur Rahman

Date:24-01-2019

.....

**Student Name:**

Fokhrul Islam

Date:24-01-2019

**Dedicated to...**

**Beloved Parents & Teachers**

# Acknowledgements

First of all we are grateful to Allah, the Almighty, the Merciful without whose blessing, this project would not have been successful. Allah gave us confidence, courage and determination to overcome the obstacles we faced during this journey.

We would like to thank to Asma-ull-Hosna, Assistant Professor of the Department of Computer Science & Engineering, Stamford University Bangladesh, for her unconditional guidance, insights, immense patience, unlimited encouragement and inspirations throughout this whole project. She is the one who inspired us to take this project and supported us every step of the way in every possible way. Without her, this project would not have come to fruition. She provided the valuable technical advice and pointed us in the right directions which were much required for a project like this. We are truly grateful for having a teacher like her as our supervisor.

We specially thanks to all the teachers of the faculty of Computer Science & Engineering, Stamford University Bangladesh for their help, valuable suggestions and discussions. Moreover, we would like to thank all our friends who have supported and helped us throughout this project.

Last of all we would also like to thank our parents who have walked along with us in every step of our life and always encouraged and inspired us to do greater things in our life.

# Table of Contents.

<b>1</b>	<b>Introduction.....</b>	<b>1</b>
1.2	Scope.....	2
1.3	Challenges.....	2
1.4	Overview.....	2
<b>2</b>	<b>Literature Review.....</b>	<b>3</b>
2.1	Personal Assistant Robot.....	3
2.2	Definition.....	3
2.3	Characteristic of personal assistant robot .....	3
2.4	History of personal assistant robot.....	3
2.5	Generations of personal Assistant robot.....	3
<b>3</b>	<b>System Methodologies.....</b>	<b>4</b>
3.1	Voice commands .....	4
3.2	Speech Recognition.....	4
3.3	Statistical speech recognition .....	5
3.4	Test to speech .....	5
3.5	Desire to task complete .....	6
3.6	Query Processor .....	7
<b>4</b>	<b>Tools.....</b>	<b>8</b>
4.1	Hardware.....	8
4.1.1	Raspberry Pi 3 Model B.....	8
4.1.2	Raspberry Pi 3 Model B Required Accessories.....	8
4.1.2.1	Micro SD Card.....	8
4.1.2.2	Display and Connectivity Cable.....	9
4.1.2.3	Keyboard and Mouse.....	9
4.1.2.4	Power Supply.....	9
4.1.2.5	Ethernet (Network) Cable.....	9
4.2	Arduino Mega.....	9
4.2.1	Introduction.....	9
4.2.2	Arduino Mega Description.....	9

4.2.3	Power cable of Arduino.....	10
4.3	Web Cam.....	10
4.4	Havit M60 Mini Microphone.....	11
4.4.1	Key Features of Havit M60.....	11
4.4.2	Specifications of Havit M60 microphone.....	11
4.5	K12 mini Speaker.....	11
4.6	Servo Motor MG 996R.....	12
4.6.1	Specifications.....	13
4.7	Servo Power Supply.....	13
4.8	Gear motor.....	14
4.9	Lithium Polymer Battery.....	14
4.9.1	Lithium Polymer Battery Checker.....	15
4.10	Arduino Motor Shield.....	15
4.11	Quick start.....	16
4.12	Robot Wheels.....	16
4.13	fiberglass Chassis.....	17
4.14	Cooling Fan.....	17
4.15	Switch.....	17
4.15.1	Rocker Off-On SPST Switch.....	17
4.15.2	DPDT Switch.....	17
4.15.2.1	Features of DPDT Switch.....	18
4.16	USB Connector.....	18
4.16.1	USB connector characteristic.....	19
4.17	2-axis servo mount (Pan/Tilt) standard servo.....	19
4.17.1	Long u-shaped steering gear bracket.....	20
4.17.2	Features.....	20
4.17.3	Package included.....	20
<b>5</b>	<b>Software.....</b>	<b>21</b>
5.1	Raspberry pi stretch.....	21
5.2	Etcher.....	21
5.3	Arduino IDE.....	22



5.3.1	Arduino SDK.....	22
5.4	Wit.ai(Speech recognition software platform).....	22
5.5	eSpeak(Text to speech converting software).....	22
5.5.1	eSpeak is a available as.....	23
5.6	CMU-Lex tool(pronunciation dictionary genating tool).....	23
<b>6</b>	<b>System Features.....</b>	<b>24</b>
6.1	Weather Information.....	24
6.2	Time Schedule Information.....	24
6.3	Wikipedia information.....	25
6.4	Google Calendar.....	25
6.5	Allocating iPhone.....	26
6.6	Email Checking .....	26
<b>7</b>	<b>Network server and software.....</b>	<b>27</b>
7.1	Putty.....	27
7.1.1	Features.....	27
7.2	VNC.....	27
7.2.1	Features.....	27
7.3	Access Over Internet.....	28
<b>8</b>	<b>System Design.....</b>	<b>29</b>
8.1	design & developmen hardwaret.....	29
8.1.1	Raspberry pi3 installing.....	29
8.1.2	Burning the SD card.....	29
8.1.3	IP Scanning.....	29
8.1.4	Connecting through SSH.....	29
8.1.5	Monitor connect VNC to use laptop as pi.....	29
8.1.6	Configuring the Raspberry pi3.....	29
8.1.7	Installing the dependences.....	30
8.1.8	STT engine.....	30
8.1.9	Offline TTS engine.....	31
8.1.10	Module Adding.....	32
8.2	Create word dictionary.....	32
8.2.1	Profile creating.....	33

8.3	Device Interfacing.....	33
<b>9</b>	<b>Software Design &amp; Development.....</b>	<b>34</b>
9.1	Program Flow.....	34
<b>10</b>	<b>Implementation.....</b>	<b>34</b>
<b>11</b>	<b>Conclusion.....</b>	<b>36</b>
11.1	Limitation.....	36
11.2	Future plan.....	36
<b>12</b>	<b>References.....</b>	<b>37-40</b>
<b>13</b>	<b>Appendix.....</b>	<b>41-55</b>

# List of Figures

Figure 1.1	Block Diagram Personal Assistant system.....	3
Figure 1.2	Statistical Speech recognition.....	5
Figure 1.3	Text to speech.....	6
Figure 1.4	Raspberry pi model B.....	8
Figure 1.5	Arduino Mega.....	9
Figure 1.6	Power cable.....	10
Figure 1.7	USE Webcam.....	10
Figure 1.8	Havit M60 Mini Microphone.....	11
Figure 1.9	K12 mini Speaker.....	12
Figure 2.1	Servo Motor MG 996R.....	12
Figure 2.2	Servo Motor MG 996R Circuit.....	13
Figure 2.3	Servo Power Supply.....	14
Figure 2.4	DC Gear Motor.....	14
Figure 2.5	Lithium Polymer Battery.....	14
Figure 2.6	Arduino Motor shield.....	15
Figure 2.7	DF Robot 65mm Rubber Wheel Pair Wheel.....	16
Figure 2.8	Rocker Off-On SPST Switch.....	17
Figure 2.9	DPDT Switch.....	18
Figure 3.1	USB Connector.....	18
Figure 3.2	2-axis servo mount (Pan/Tilt) standard servo.....	19
Figure 3.3	Dataflow Diagram.....	20
Figure 3.4	Our Implemented Robot.....	34

# List of Table

4.6.1 USB connector characteristic.....	19
---	----

# 1 Introduction

With the development of internet, people's learning and working rely on the network more and more. However, according to our investigation, we are sometimes too busy with daily small things to balance well between learning and working because we are lack of effective plan and management. So it is necessary to design a personal assistant system to support our learning and working with its instantaneity, speed, effectiveness and low cost. Recently, as an interactive robot application and service platform, many personal robots have been developed by some labs or companies, such as Ver Bot, Xiao I robot, MSN robot, Chat robot etc. These robots are just like a real assistant, and can also query courses, schedules, maps, stocks, weather, television programs and other information. At present, personal assistant systems have got new application with the promotion of instant messaging technology. Instant messaging (IM) is a communication technology, which allows a user to find out who is online and available to receive messages. Nardi et al. pointed out that IM can allow collaboration, scheduling, impromptu meeting and contact with friends and family(B. A. Nardi, S. Whittaker, E. Bradner, 2000).[32] MSN robot is a friendly assistant robot with IM that provides novel user experience and practical network service for people basing on the research, development and application of artificial intelligence. For example, Xiao I Robot from Yingsi Software has integrated with the MSN, QQ, and Yahoo Messenger. It has become one of the most popular IM platforms. It has 10 million users and covers 100 countries. Now the company has developed other applications, such as Xiao I Chat, Xiao I Map, Xiao I Weather, Xiao I Q&A. In our research, we first studied the IM protocol, and used its landing principles and message sending principles to design and achieve an IM client.[33] The client can login a number of accounts at the same time in one computer, and start chatting with different friends. Then we studied the operation principle and designed the chat robot system based on the client. At the same time, in order to develop a convenient personal assistant tool for people's learning and working, we also studied the application of personal assistant software, instant messaging software and personal robots. At last, we made some investigation in our university and decided to develop a personal assistant system to help students manage their courses, homework, review plan, research subjects, campus activities and board wages. In addition, we also study the impact on interaction effect for students by providing chatting functions. For future work, we hope to develop a personal assistant system with more interactive and visual function. [12][31]

## 1.2 Scope:

It's here – a world in which helper robots live with us, get us through the day, and yes, become our trusted friend. Science fiction is becoming science fact. As sci-fi writer William Gibson has apparently noted: “The future is already here – it's just not very evenly distributed. The demand for helper robots is booming, and academia, industry and the military are working overtime to meet the need as a whole new industry gains momentum.

This project can play a vital role for make life easier. This project has been implemented with a low cost whereas anyone can use it, specially disabled person for make their everyday life easier.

## 1.3 Challenges

Personal assistant system robot's are become more popular tremendously. But Still there are some issues which makes a system less accurate which are discussed below:

**Mobility:** Most of personal assistant haven't mobility access.

**GPS ability:** Finding i-phone using assistant system.

**Weather information:** It was so necessary to have accurate information about weather by assistant system robot.

**Communication:** communications between social network and user's are not so much stablished properly.

**Voicecommands:** Voice command instruction's were so limited for existing robots. Less instructions replies with less informations.

**Hardware:** Only few robots have such a ability to recognize the location of it's sorroundings. The facts we mention in above are the main focus for our project. We tried our best to implement them in our personal assistant system robot.

## 1.4 Overview:

In this chapter, we explore the concept of personal assistant robot, its elaboration in detail, business opportunity and real world applications, Voice recognition and its prospects in daily life with its huge benefits such as safety for babies and physically disabled people.

In next chapter we discuss about the literature review of the personal assistant robot, its characteristics, applications etc as well as we discuss about some existing voice controlling. In 3<sup>rd</sup> chapter, we introduce about the system methodology, 4<sup>th</sup> we introduce about the hardware. 5<sup>th</sup> we introduce software tools that we use in this project. In 6<sup>th</sup> chapter, we elaborate the our system features. 7<sup>th</sup> chapter, introduce network service in this project. 8<sup>th</sup> we discussed system design in this project. we explore our project in detail.

And finally in 9<sup>th</sup> chapter, we conclude with some limitations of our system, future plans and possibities in our country.

## 2 Literature Review

In this chapter we will discuss about the home automation system. We will also compare with some existing project which is almost similar to our project.

### 2.1.1 Personal Assistant Robot

#### 2.1.2 Definition

A **personal robot** is one whose human interface and design make it useful for individuals. This is by contrast to industrial robots which are generally configured and operated by robotics specialists. A personal robot is one that enables an individual to automate the repetitive or menial part of voice controlling or work life making them more productive.[34][35]

#### 2.1.3 Characteristic of personal assistant robot

There are one main characteristics of personal assistant robot. These are as follows:

1. **Voice Command** : This system is control by voice command. Do the voice command then Robot response to the voice output. This system voice command implemented by a weather, time, Google search, robot movement, Google calendar, Email checking etc is working this system through the voice command.

#### 2.1.4 History of personal assistant robot

- iRobot Corp. introduced the Roomba in 2002
- The Institute for Personal Robots in Education introduced the concept to teach computing using personal robots in 2006.[14]
- Stanford University Personal Robotics Program introduced PR1 in 2007.<sup>[1]</sup> Stanford AI Robot (STAIR)
- Willow Garage introduced the PR2 robot in 2010[15]
- Robot Dynamics introduced Luna in 2011[16]
- Milagros Human Tech introduced India's 1st Robotic vacuum cleaner, the RedHawk in 2011 and then the World's 1st Body Massaging Robot in 2012[36]

#### 2.1.5 Generations of personal Assistant robot

Personal Assistant robotics generally divided into three generations:

- i. Remote Control (First generation)
- ii. Robot buddy (Second generation)
- iii. Artificial Intelligence(Third generation)

### 3 System Methodology

There are four main steps that we used for our personal assistant system is shown below. Firstly, we take some voice commands. Then those voice commands are sent to the speech recognition cloud site for conversion speech to text. After conversion, it passes the text value to the robot for matching, is that value is valid or not. If valid, then it complete the specific task through modules. At the last stage text to speech convert the text into particular voice command as reply.[37]

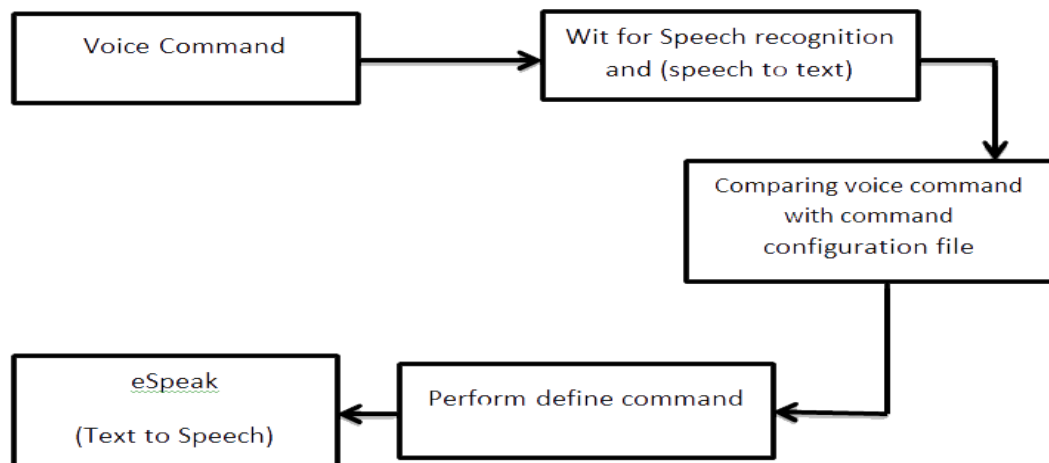


Figure 1.1: Block Diagram Personal Assistant system

#### 3.1 Voice Command

The system takes voice commands through usb mic or webcam. Voice frequency has been sent to the speech recognition engine for converting the speeches into texts.

#### 3.2 Speech recognition

##### Fundamental Equation of Statistical Speech Recognition:

If  $\mathbf{X}$  is the sequence of acoustic feature vectors (observations) and  $\mathbf{W}$  denotes a word sequence, the most likely word sequence  $\mathbf{W}^*$  is given by

$$\mathbf{W}^* = \arg \max_{\mathbf{W}} P(\mathbf{W} | \mathbf{X})$$

If  $\mathbf{X}$  is the sequence of acoustic feature vectors (observations) and  $\mathbf{W}$  denotes a word sequence, the most likely word sequence  $\mathbf{W}^*$  is given by

$$\mathbf{W}^* = \arg \max_{\mathbf{W}} P(\mathbf{W} | \mathbf{X})$$

Applying Bayes' Theorem:



$$\begin{aligned}
P(W | X) &= p(X | W)P(W) p(X) \\
&\propto p(X | W)P(W) \\
W^* &= \arg \max_w \underbrace{p(X | W)}_{\text{Acoustic model}} \underbrace{P(W)}_{\text{Language model}}
\end{aligned}$$

### 3.3 Statistical speech recognition

Statistical models offer a statistical “guarantee” see the licence conditions of the best known automatic dictation system, for example: Licensee understands that speech recognition is a statistical process and that recognition errors are inherent in the process. Licensee acknowledges that it is licensee’s responsibility to correct recognition errors before using the results of the recognition.[38]

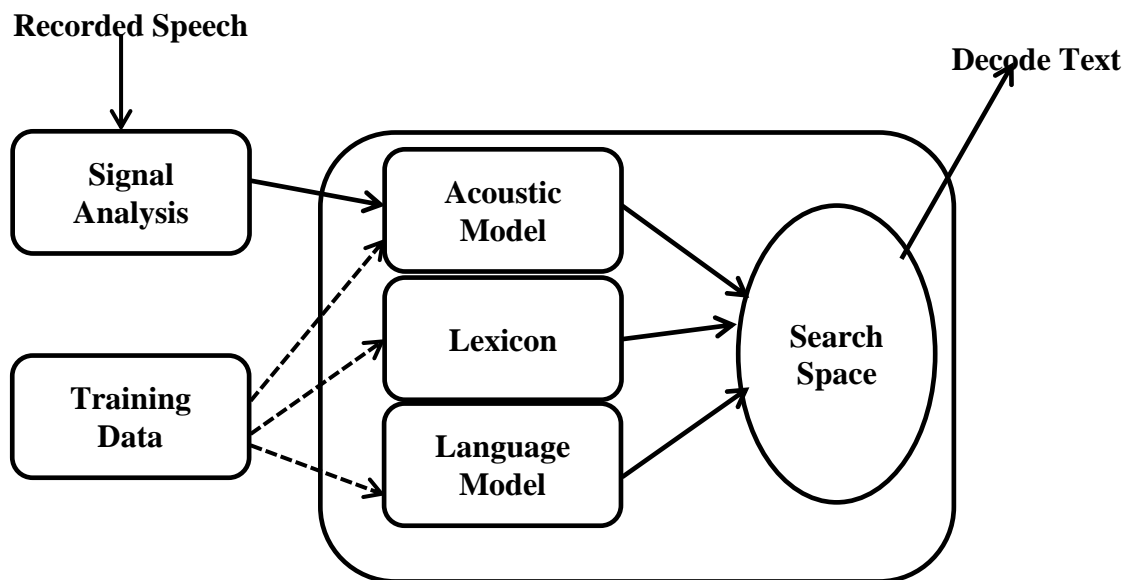


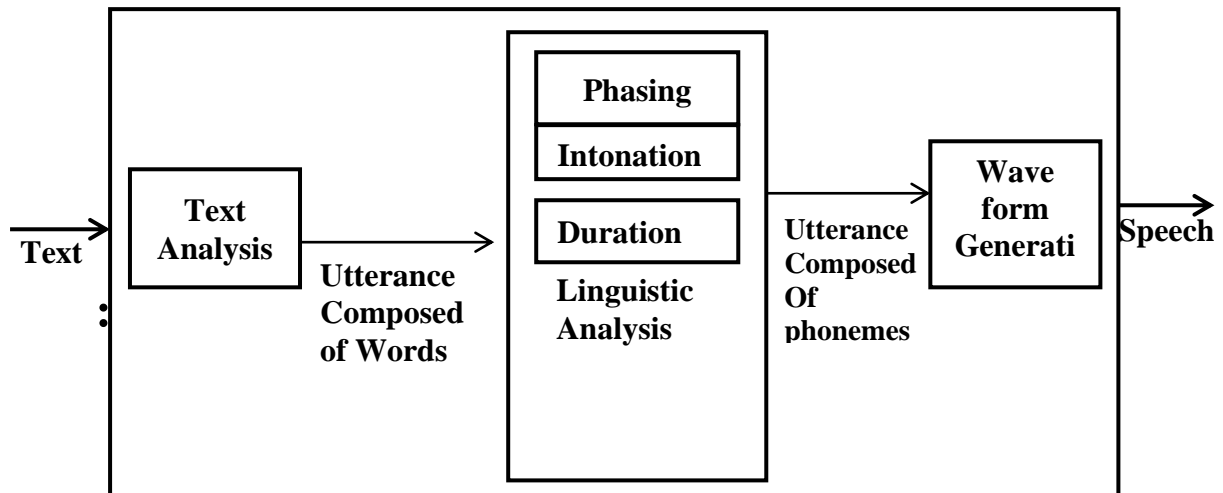
Figure 1.2 Statistical speech recognition

### 3.4 Text To Speech

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific

usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.[39]



**Figure 1.3 Text to Speech**

All speech-to-text systems rely on at least two models: an acoustic model and a language model. In addition large vocabulary systems use a pronunciation model. It is important to understand that there is no such thing as a universal speech recognizer. To get the best transcription quality, all of these models can be specialized for a given language, dialect, application domain, type of speech, and communication channel.[40]

Like any other pattern recognition technology, speech recognition cannot be error free. The speech transcript accuracy is highly dependent on the speaker, the style of speech and the environmental conditions. Speech recognition is a harder process than what people commonly think, even for a human being. Humans are used to understanding speech, not to transcribing it, and only speech that is well formulated can be transcribed without ambiguity[41].

From the user's point of view, a speech-to-text system can be categorized based in its use: command and control, dialog system, text dictation, audio document transcription, etc. Each use has specific requirements in terms of latency, memory constraints, vocabulary size, and adaptive features.

### 3.5 Desire task to complete

espeak is a more modern speech synthesis package than Festival. It sounds clear but does wail a little.

After matching the voice command with configuration file it performs the tasks using eSpeak through the voice.

### 3.6 Query Processor

The Voice Command System has a module for query processing which works in general like many query processors do. That means, taking the input from the users, searching for relevant outputs and then presenting the user with the appropriate output. In this system we are using the site wolfram alpha as the source for implementing query processing in the system. The queries that can be passed to this module include retrieving information about famous.

```
WORDS=["STATUS"]
```

```
def isValid(text):
```

```
    return bool(re.search(r'\b(status)\b', text, re.IGNORECASE))
```

## 4. Tools

**4.1 Hardware:** The hardware tools we have used for this project are as follows

### 4.1.1 Raspberry Pi 3 Model B

The Raspberry Pi is a very small and low-cost computer which has the size of a credit card and can be plugged into any computer monitor or TV. It uses a standard keyboard and mouse. It has all the capability of a desktop like browsing the Internet, playing high definition video, making spreadsheets, word processing and playing games. What is unique to Raspberry Pi is that it allows the user to interact with the outside world and is currently used in many digital projects. Users can learn to write programs by using languages like Scratch and Python. [9].



**Figure1.4 : Raspberry pi model B**

The Multipurpose Surveillance Robot has utilized the Raspberry Pi 3 Model B which is the second generation Raspberry Pi which replaced the original Raspberry Pi 3 Model B+. It has:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5 mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- Video Core IV 3D graphics core

### 4.1.2 Raspberry Pi 3 Model B Required Accessories

#### 4.1.2.1 Micro SD Card

An 8 GB class 4 micro SD card with Raspbian jessie pre-installed and another 8 GB class 4 micro SD card with Android Things pre-installed, is recommended. The minimum recommended capacity of an SD card is 8 GB. 4 GB is recommended for image installation. Even smaller cards can be used for some distributions like OpenElec8 and Arch. The SD card

class determines the write speed a card can sustain. A class 4 SD card can achieve 4 MB/s write speed whereas for a class 10 card 10 MB/s is attainable.

#### **4.1.2.2 Display and Connectivity Cable**

Any HDMI/DVI monitor and any TV works as display for pi. But one with an HDMI input is recommended.

#### **4.1.2.3 Keyboard and Mouse**

Raspberry pi works with any standard keyboard and mouse. Wireless keyboard and mouse will work if already paired with Pi. Keyboard layout can be configured through raspberry pi.

#### **4.1.2.4 Power Supply**

The Pi is powered by a USB micro power supply like most standard mobile phone chargers. A good-quality power supply is required that can supply at least 700mA at 5V. Power supplies with less than 700 mA current should work for basic usage but the Pi might reboot if too much power is drawn.

#### **4.1.2.5 Ethernet (Network) Cable**

An Ethernet cable is necessary to connect Pi to a local network and the internet. It is necessary when built-in Wi-Fi or Wi-Fi module is not available for connection.

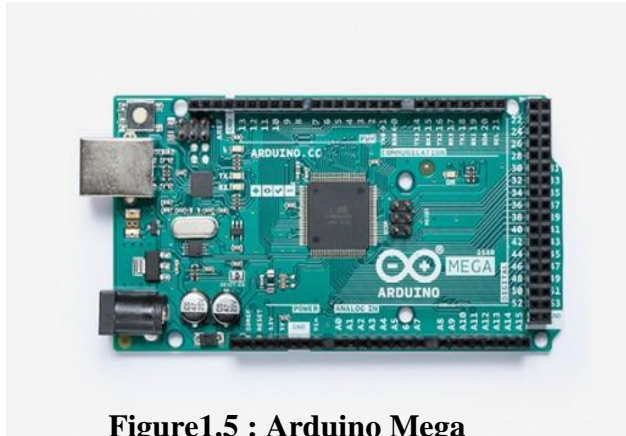
### **4.2 Arduino Mega**

#### **4.2.1 Introduction**

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Decimal.

#### **4.2.2 Arduino Mega Description**

The Arduino Mega is the addition to the Arduino family. This board is physically larger than all the other boards and offers significantly more digital and analog pins. The MEGA uses a different processor allowing greater program size and more. The Mega2560 differs from all preceding boards in that



**Figure1.5 : Arduino Mega**

it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 programmed as a USB-to-serial converter. The Mega has four hardware serial ports, which means maximum speed if you need a second or third (or fourth) port. The Arduino Mega works in the same way the Arduino Uno does but the difference is that it uses ATmega2560 microcontroller and has more number of digital pins, analog pins.[13]

#### **4.2.3 Power cable of Arduino**

The board can be supplied with power either from the DC power jack (7 - **12V**), the **USB** connector (**5V**), or the VIN pin of the board (7-**12V**). Supplying voltage via the **5V** or 3.3V pins bypasses the regulator, and can damage your Arduino.



**Figure1.6: Power cable**

### **4.3 Web Cam**

A **webcam** is a video camera that feeds or streams its image in real time to or through a computer to a computer network. When "captured" by the computer, the video stream may be saved, viewed or sent on to other networks travelling through systems such as the internet, and e-mailed as an attachment. When sent to a remote location, the video stream may be saved, viewed or on sent there. Unlike an IP camera(which connects using Ethernet or Wi-Fi), a webcam is generally connected by a USB cable, or similar cable, or built into computer hardware, such as laptops.



**Figure1.7 : USB Webcam**

The term "webcam" (a clipped compound) may also be used in its original sense of a video camera connected to the Web continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its web page over the Internet. Some of them, for example, those used as online traffic cameras, are expensive, rugged professional video cameras.

#### **4.4 Havit M60 Mini Microphone**

The most popular electronic brand “Havit” brings a high-performance microphone in cheap rate, whose name is “HavitM60”.If you want to buy Professional background noise cancellation technology uses a microphone. We recommend you first check the Havit M60 Microphone Full Specification and Features. Now this Havit M60 Mini Microphone.



**Figure1.8: Havit M60 Mini Microphone**

##### **4.4.1 Key Features of Havit M60**

- Professional background noise cancellation technology.
- The use of noise cancellation technology microphone, voice clarity.
- Portable, lightweight design, suitable for desktop and laptop dual-use.
- Condenser microphone High-quality MIC, perfect voice recorder Fit for all kinds of PC.
- Havit Noise cancellation microphone.

##### **4.4.2 Specifications of Havit M60 microphone**

- 5mm stereo-track plug

- Frequency response: 20Hz-16KHz
- Sensitivity: – 39dB+3dB
- Cable Length: – 2.5mmX1.5mm
- Plug: – 3.5mm
- Cable Length: Up to 1.5 meters or 5 feet.

#### 4.5 K12 mini Speaker:

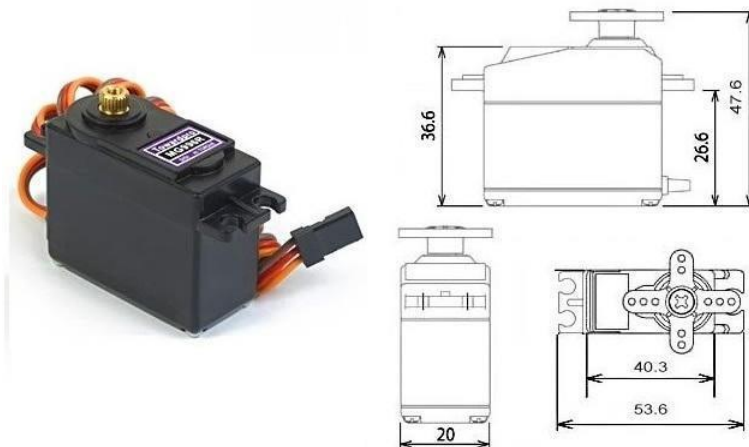
- Mini speaker for PC, MP3, MP4, and notebook
- with 3.5mm Audio Cable
- material: steel plastic
- Material: Plastic
- Power: USB5V



**Figure1.9 : K12 mini Speaker**

#### 4.6 Servo Motor MG 996R

This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo.



**Figure2.1:Servo Motor MG 996R**

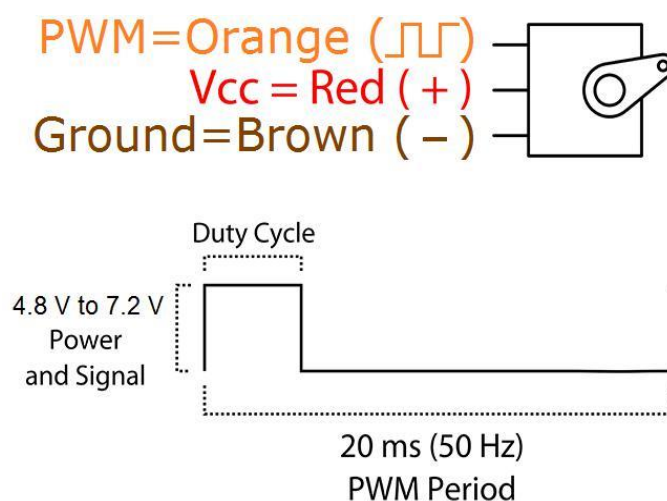


and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwidth and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spectrum .

This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast.[11]

#### 4.6.1 Specifications

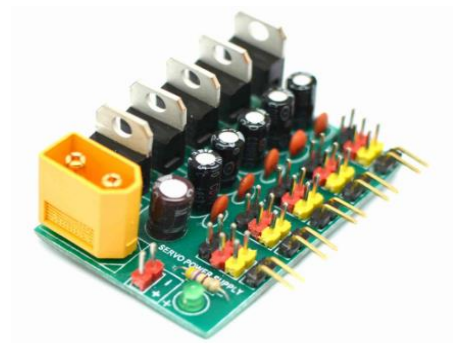
- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kg ·cm (4.8 V ), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5  $\mu$ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C



**Figure 2.2: Servo Motor MG 996R Circuit**

## 4.7 Servo Power Supply

While working on a project that requires a bunch of servo motors (e.g.: a robotic arm), the first concern is to provide sufficient power to all of the motors. This module is useful to supply clean 5V to all of the servo motors of your project easily. There are a total of 5 groups of servo connectors. Each of them are capable of supplying up to 1A peak current. A maximum of 10 micro sized servo or 5 medium sized servo motors can be powered up using this module.[30]



**Figure2.3: Servo Power Supply**

## 4.8 Gear motor

The gear motor wheel is perfect for your next robotics project. This gear motor is ideal for robotic car or line-tracing robot. With plastic construction and colored in bright yellow, the DC gear motor measures approx. 2.5 inch long, 0.85 inch wide and 0.7 inch thick. The wheel can be mounted on either side and the gear motor works well between 4V to 7V (recommended 6 Volts). At a ratio of 1:48 you can get some really good torque at 5 Volts.



**Figure 2.4 : DC Gear Motor**

## 4.9 Lithium Polymer Battery

Due to US post service (USPS) restriction, we do not ship the product internationally by USPS.

Quality Glacier Lipo batteries are made by a specialized factory that also manufactures packs for well-known premium brands. Glacier batteries are great for sport and 3D pilots, as well as competition drivers.



**Figure 2.5 :Lithium Polymer Battery**

The following characteristics make Glacier batteries outstand competitors that are sold at lower prices from oversea dealers, or even many name brands sold at higher prices domestically.

Conservative C rating. Glacier 30C and 35C batteries outperform many 40C and 45C batteries in the market.

Higher discharge platform, less voltage sag. You will feel more power during high discharge current situations, such as 3D flight.

Better temperature control technology. After high rate discharge, Glacier batteries are cooler than most competitors labeled the same

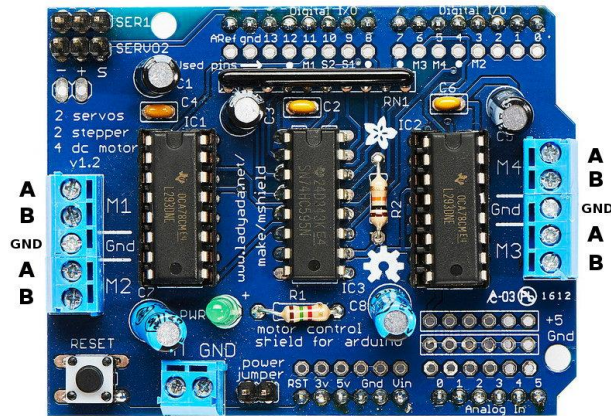
C rating. Longer cycle life. Better quality control. In additional to the manufacturer's strict QC and QA systems, we check the voltage of every single battery with capacity more than 800mAh before we send out your order.

#### **4.9.1 Lithium Polymer Battery Checker**

1. This battery checker is only for Lipo battery use.
2. This battery checker can check battery of Voltage range:  $3.5 < V < 16.5$
3. Battery checker has Voltage error:  $\pm 3\%$
4. This battery checker can check 2 to 4 cells maximum.

#### **4.10 Arduino Motor Shield**

Ad fruit designed a very useful shield (in 2008 ?), called "Ad fruit Motor Shield". It was (is) a useful shield with older components. The tutorial of Ad fruit explains the in 2014, Ad fruit has discontinued this shield and has a new and much better motor shield. This page is only here for the old shield, and for the clone boards.



**Figure 2.6: Arduino Motor shield**

The shield contains two L293D motor drivers and one 74HC595 shift register. The shift register expands 3 pins of the Arduino to 8 pins to control the direction for the motor drivers. The output enable of the L293D is directly connected to PWM outputs of the Arduino.

To increase the maximum current, the L293D allows extra chips with "piggyback". Piggyback is soldering one or two or three extra L293D drivers on top of the L293D drivers on the board to

increase the maximum current. The L293D allows parallel operation.

The Motor Shield is able to drive 2 servo motors, and has 8 half-bridge outputs for 2 stepper motors or 4 full H-bridge motor outputs or 8 half-bridge drivers, or a combination.

The servo motors use the +5V of the Arduino board. The voltage regulator on the Arduino board could get hot. To avoid this, the newer Motor Shields have connection points for a separate +5V for the servo motors.

#### **4.11 Quick start**

Supply the power either through the Arduino Board or the power header of the Motor Shield. Only if the Arduino Board and the Shield use a separate power supply, the jumper must be removed.

Load the sketch below in Arduino (the sketch is not for stepper motors) and follow the example in the sketch to drive your DC-motors or solenoids or lights.

#### **4.12 Robot Wheel**

- Hub is sold separately
- Tire material: Rubber
- Soft black rubber with light tread
- Tire width: 26mm
- Outside diameter: 65mm



**Figure 2.7 :- DF Robot 65mm Rubber Wheel Pair Wheel**

### **4.13 fiberglass Chassis**

**Fiberglass:** Fiberglass is an outstanding material for creating specific shapes that would be nearly impossible to make from metal or wood. It is also extremely strong and rigid once set, as well as waterproof. The process involves laying a fiberglass cloth and then applying a two-part resin on top of the cloth. It only takes about 1 hour to harden, but it does make some strong fumes.

Though wood has more advantages this post will cover the use of acrylic as the material for building chassis. Acrylic is available in all shapes and sizes with a wide range of thickness to choose from. One other compelling factor is that it is available in vibrant colors which gives a new visual appeal to your robot.

### **4.14 Cooling Fan**

10-15.6" Notebook/Laptop Adjustable Cooling Pad Stand Cooler Fan Blue LED USB Hub

100% brand new and high quality

Cooling pad for laptop with built-in powerful fan for easy thermal heat dissipation

USB Laptop Cooling Pad Quiet Fan

Extremely heavy-duty construction of the cooling pad to afford any weight or pressure put-on

No need for any installation

Your laptop can breathe easily with this notebook cooler

Extend the life and functionality of your laptop

Convenient, reliable and comfortable to carry

## 4.15 Switch

The switch is an electrical device that is used to break or make an electrical circuit manually or automatically. The types of switches depend on the connections of the circuit they make. Two essential components such as pole and through can confirm what types of connections a switch can make

### 4.15.1 Rocker Off-On SPST Switch

10A Rocker Switch 3-pin, SPDT.

This single-pole, double-throw (SPDT) rocker switch makes a great on/off power switch.

It is rated for 10 A at 125 VAC and 6 A at 250 VAC

- SPST on-off rocker switch
- 16A continuous at 125VAC
- Panel Mount, Snap-in
- 19.2mm x 12.9mm panel cutout



**Figure2.9 :Rocker Off-On SPST Switch**

### 4.15.2 DPDT Switch

A Double Pole Double Throw toggle switch acts exactly like two separate SPDT switches connected to the same switch bat. It has two separate common terminals and each of those is connected to one or the other of the other two terminals on the same side of the switch.



**Figure 3.1: DPDT Switch**

#### 4.15.2.1 Features of DPDT Switch

- Photopolymer.
- Good Process ability.
- Good Impact Resistance.
- Good Stiffness.
- High Current Flow.

#### 4.16 USB Connector

USB, short for Universal Serial Bus, is an industry standard that was developed to define cables, connectors and protocols for connection, communication, and power supply between personal computers and their peripheral devices.



**Figure 3.2: USB Connector**

USB was designed to standardize the connection of computer peripherals (including keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters) to personal computers, both to communicate and to supply electric power. It has largely replaced interfaces such as serial ports and parallel ports, and has become commonplace on a wide range of devices. USB connectors have replaced other types for battery chargers of portable devices.

##### 4.16.1 USB connector characteristic

Contact Resistance:	30m $\Omega$
Insulation Resistance:	100M $\Omega$ min
Withstand Voltage:	AC 500V/min
Temperature Range:	-55~85°C
Rated Load:	AC DC 30V 1A
Insertion Force	35N max

Extaction Force	7N min
Durability (Mating)	5000 times

#### **4.17 2-axis servo mount (Pan/Tilt) standard servo**

- Multifunction steering gear bracket
- Material: aluminum plate
- Thickness: 2mm
- Table: oxidation of fine sand silver and Black 2 color
- Size: 58X37X25.5mm
- Weight: 15.5G

##### **4.17.1 Long u-shaped steering gear bracket**

- Material: 2mm aluminum plate
- Table: abrasive oxide
- Color: black

##### **4.17.2 Features**

- Aluminum Matte Coat Pan and Tilt for horizontal surface, unassembled
- Using two servos (MG995 or HS322,HS422,Hitec,Parallax,Futaba, etc 40×20×36mm servo).
- Servo bracket used in the shoulders and knees or other joint of humanoid robots, biped robots etc.
- Can plant a camera or IR sensor for Robot



**Figure 3.3: 2-axis servo mount (Pan/Tilt) standard servo**



#### **4.17.3 Package included:**

- 1 x Aluminum Alloy multi-function servo bracket
- 1 x Short U-type servo bracket
- 2 x Metal Servo horn
- 1 x Ball bearing
- 1 x Screw set

## **5 Software**

### **5.1 Raspberry pi stretch**

Raspberry is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspberry provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi.[42]

The initial build of over 35,000 Raspberry packages, optimized for best performance on the Raspberry Pi, was completed in June of 2012. However, Raspberry is still under active development with an emphasis on improving the stability and performance of as many Debian packages as possible

Debian 9 (Stretch) was released on 17 June 2017, two years and two months after last release Debian 8 (Jessie), and contained more than 51,000 packages. and the latest minor updates, called point releases; is version 9.6, released on November 10, 2018; 61 days ago. Major upgrades include the Linux kernel going from version 3.16 to 4.9, GNOME desktop version going from 3.14 to 3.22, KDE Plasma 4 was upgraded to Plasma 5, LibreOffice 4.3 upgraded to 5.2 and Qt upgraded from 4.8 to 5.7. LXQt has been added as well.

### **5.2 Etcher**

Etcher is a graphical SD card writing tool that works on Mac OS, Linux and Windows, and is the easiest option for most users. Etcher also supports writing images directly from the zip file, without any unzipping required. To write your image with Etcher: Connect an SD card reader with the SD card inside.

Open Etcher and select from your hard drive the Raspberry Pi .img or .zip file you wish to write to the SD card. Select the SD card you wish to write your image to.

Review your selections and click 'Flash!' to begin writing data to the SD card.

### **5.3 Arduino IDE**

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, Mac OS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino board.<sup>[2]</sup>

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and

linked with a program stub `main()` into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware .

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them.

### **5.3.1 Arduino SDK**

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows ) 7previously XP) or later. As of March 2015, the SDK is not available on Android[43]

itself, but the software development is possible by using specialized Android applications. Until around the end of 2014, the officially supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plug-in, though IntelliJ IDEA IDE (all editions) fully supports Android out of the box and NetBeans IDE also supports Android development via a plug-in. As of 2015, Android Studio, made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others.[44]

Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g. triggering a reboot, installing software package(s) remotely). Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are download able components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing. Android applications are packaged in .apk format and stored under folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains doc files (compiled byte code files called David executable), resource files, etc.

## **5.4 Wit.ai(Speech recognition software platform)**

Wit.ai makes it easy for developers to build applications and devices that you can talk or text to. Our vision is to empower developers with an open and extensible natural language platform. Wit.ai learns human language from every interaction, and leverages the community: what's learned is shared across developerswe have used wit.ai STT(speech to text engine) to . [29].recognize the speech recognition

## **5.5 eSpeak(Text to speech converting software)**

eSpeak is a compact open source software speech synthesizer for English and other languages, for Linux and Windows.

eSpeak uses a "formant synthesis" method. This allows many languages to be provided in a small size. The speech is clear, and can be used at high speeds, but is not as natural or smooth as larger synthesizers which are based on human speech recordings.[6]

### **5.5.1 eSpeak is available as**

A command line program (Linux and Windows) to speak text from a file or from stdin.

A shared library version for use by other programs. (On Windows this is a DLL).A SAPI5 version for Windows, so it can be used with screen-readers and other programs that support the Windows SAPI5 interface.

eSpeak has been ported to other platforms, including Android, Mac OSX and Solaris, Features. Includes different Voices, whose characteristics can be altered. Can produce speech output as a WAV file.

SSML (Speech Synthesis Markup Language) is supported (not complete), and also HTML.

Compact size. The program and its data, including many languages, totals about 2 Mbytes.

Can be used as a front-end to MBROLA iphone voices, eSpeak converts text to phonemes with pitch and length information. Can translate text into phoneme codes, so it could be adapted as a front end for another speech synthesis engine.

Potential for other languages. Several are included in varying stages of progress. Help from native speakers for these or other languages is welcome. Development tools are available for producing and tuning phoneme data.[7]

## **5.6 CMU-Lex tool(pronunciation dictionary genating tool)**

This tool generates a pronunciation dictionary from a list of (English) words in a form suitable for use with a speech recognizer, such as CMUSphinx. The Lexicon Tool uses the CMUdict dictionary along with some simple normalization and inflection rules (as detailed below) to identify a word, and uses letter-to-sound rules when all else fails.

If you simply want to see if a word can be found in CMUdict, try this tool. CMUdict is a freely-available open-source pronunciation dictionary that was developed for use in speech recognition.[8]

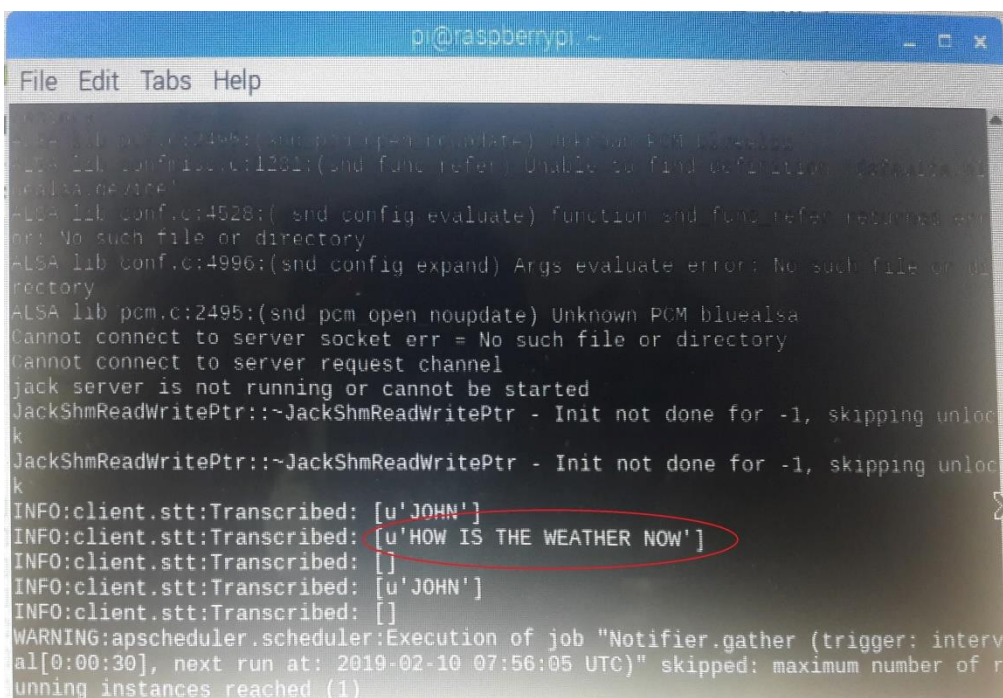
This tool is derived from the Logios package which allows you to input a Phoenix semantic grammar and receive a compiled grammar, an n-gram language model and a pronouncing dictionary.

This tool currently uses cmudict-0.7b and produces pronunciations using the (currently standard) 40 item phone inventory (see above). Please note that the dictionary is updated from time to time, so you may get slightly different results over time.[28]

## 6 System Features

### 6.1 Weather information

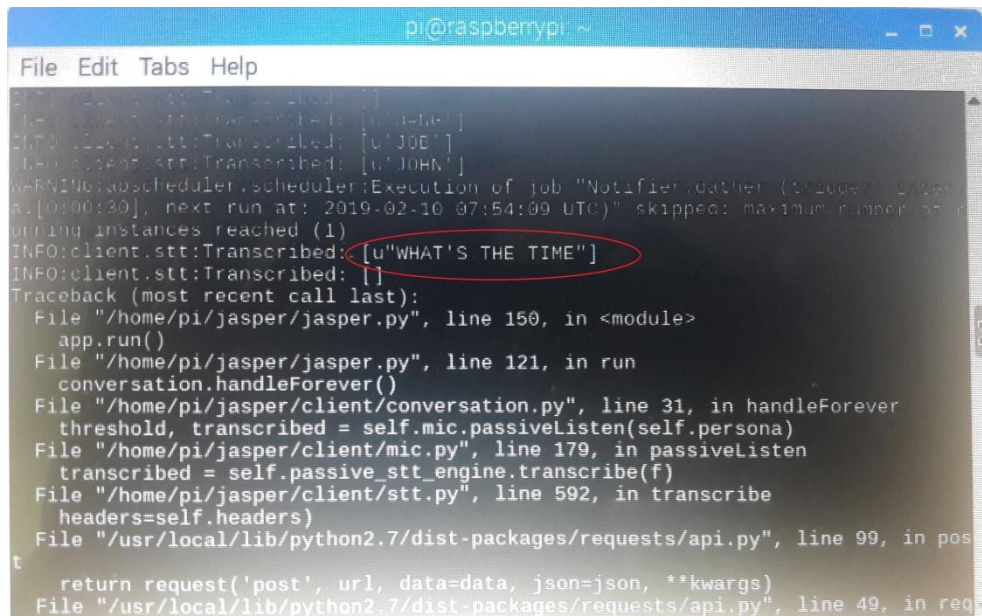
This module tells the user about the weather conditions of the location whose station identifier is specified in the profile of the user. This module can be executed by using the keyword “weather”. The weather information is taken from the weather underground service which includes the details of temperature, wind speed and direction etc. It generates an error message, if the information cannot be retrieved for the specified location.



```
pi@raspberrypi: ~
File Edit Tabs Help
ALSA lib pcm.c:4495:(snd_pcm_open_noupdate) Unknown PCM bluealsa
ALSA lib Conf.c:1281:(snd_func_refer) Unable to find definition 'alsa_devic
ALSA lib Conf.c:4528:(snd_config_evaluate) function snd_func_refer returned error: No such file or directory
ALSA lib Conf.c:4996:(snd_config_expand) Args evaluate error: No such file or directory
ALSA lib pcm.c:2495:(snd_pcm_open_noupdate) Unknown PCM bluealsa
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
INFO:client.stt:Transcribed: [u'JOHN']
INFO:client.stt:Transcribed: [u'HOW IS THE WEATHER NOW']
INFO:client.stt:Transcribed: []
INFO:client.stt:Transcribed: [u'JOHN']
INFO:client.stt:Transcribed: []
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:56:05 UTC)" skipped: maximum number of running instances reached (1)
```

### 6.2 Time Schedule reminder

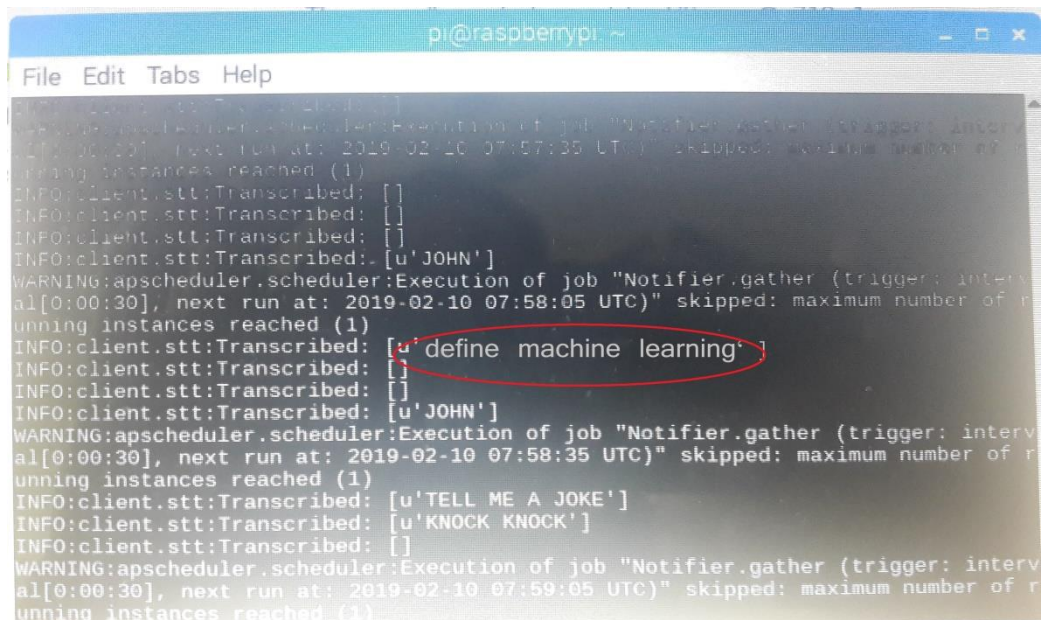
This module tells the user about the current Time of the Country whose station identifier is specified in the profile of the user. This module can be executed by using the keyword “time”. The time information is taken from the time underground service which includes the details of current time. It generates an error message, if the information cannot be retrieved for the specified country.

A terminal window titled 'pi@raspberrypi ~' with a menu bar 'File Edit Tabs Help'. The terminal shows the execution of the 'what's the time' command. The output includes several log messages: 'INFO:client.stt:Transcribed: [u'JOHN']', 'INFO:client.stt:Transcribed: [u'JOHN']', 'WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:54:09 UTC)" skipped: maximum number of running instances reached (1)', 'INFO:client.stt:Transcribed: [u"WHAT'S THE TIME"]', and 'INFO:client.stt:Transcribed: []'. A traceback follows, showing the call stack from 'app.run()' to 'requests/api.py'. The phrase 'u"WHAT'S THE TIME"' is circled in red.

```
pi@raspberrypi ~
File Edit Tabs Help
INFO:client.stt:Transcribed: [u'JOHN']
INFO:client.stt:Transcribed: [u'JOHN']
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:54:09 UTC)" skipped: maximum number of running instances reached (1)
INFO:client.stt:Transcribed: [u"WHAT'S THE TIME"]
INFO:client.stt:Transcribed: []
Traceback (most recent call last):
  File "/home/pi/jasper/jasper.py", line 150, in <module>
    app.run()
  File "/home/pi/jasper/jasper.py", line 121, in run
    conversation.handleForever()
  File "/home/pi/jasper/client/conversation.py", line 31, in handleForever
    threshold, transcribed = self.mic.passivelisten(self.persona)
  File "/home/pi/jasper/client/mic.py", line 179, in passivelisten
    transcribed = self.passive_stt_engine.transcribe(f)
  File "/home/pi/jasper/client/stt.py", line 592, in transcribe
    headers=self.headers)
  File "/usr/local/lib/python2.7/dist-packages/requests/api.py", line 99, in post
    return request('post', url, data=data, json=json, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/requests/api.py", line 49, in request
```

### 6.3 Wikipedia information

This module works on the keyword of “wiki”. The system asks for what you would like to learn about. Then the request is made to the Wikipedia API for the required query. It is made to the Wikipedia API for the required query. It generates the summary of the information regarding the query and the data is output through the microphone to the listener in audio form.

A terminal window titled 'pi@raspberrypi ~' with a menu bar 'File Edit Tabs Help'. The terminal shows the execution of the 'define machine learning' command. The output includes several log messages: 'INFO:client.stt:Transcribed: []', 'INFO:client.stt:Transcribed: []', 'INFO:client.stt:Transcribed: []', 'INFO:client.stt:Transcribed: [u'JOHN']', 'WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:58:05 UTC)" skipped: maximum number of running instances reached (1)', 'INFO:client.stt:Transcribed: [u'define machine learning']', 'INFO:client.stt:Transcribed: []', 'INFO:client.stt:Transcribed: []', 'INFO:client.stt:Transcribed: [u'JOHN']', 'WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:58:35 UTC)" skipped: maximum number of running instances reached (1)', 'INFO:client.stt:Transcribed: [u'TELL ME A JOKE']', 'INFO:client.stt:Transcribed: [u'KNOCK KNOCK']', 'INFO:client.stt:Transcribed: []', and 'WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:59:05 UTC)" skipped: maximum number of running instances reached (1)'. The phrase 'u'define machine learning'' is circled in red.

```
pi@raspberrypi ~
File Edit Tabs Help
INFO:client.stt:Transcribed: []
INFO:client.stt:Transcribed: []
INFO:client.stt:Transcribed: []
INFO:client.stt:Transcribed: [u'JOHN']
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:58:05 UTC)" skipped: maximum number of running instances reached (1)
INFO:client.stt:Transcribed: [u'define machine learning']
INFO:client.stt:Transcribed: []
INFO:client.stt:Transcribed: []
INFO:client.stt:Transcribed: [u'JOHN']
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:58:35 UTC)" skipped: maximum number of running instances reached (1)
INFO:client.stt:Transcribed: [u'TELL ME A JOKE']
INFO:client.stt:Transcribed: [u'KNOCK KNOCK']
INFO:client.stt:Transcribed: []
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:59:05 UTC)" skipped: maximum number of running instances reached (1)
```

### 6.4 Google calendar

Google Calendar module works based on current time and google API whose station identifier is specified in the profile of the user. This module can be executed by using the



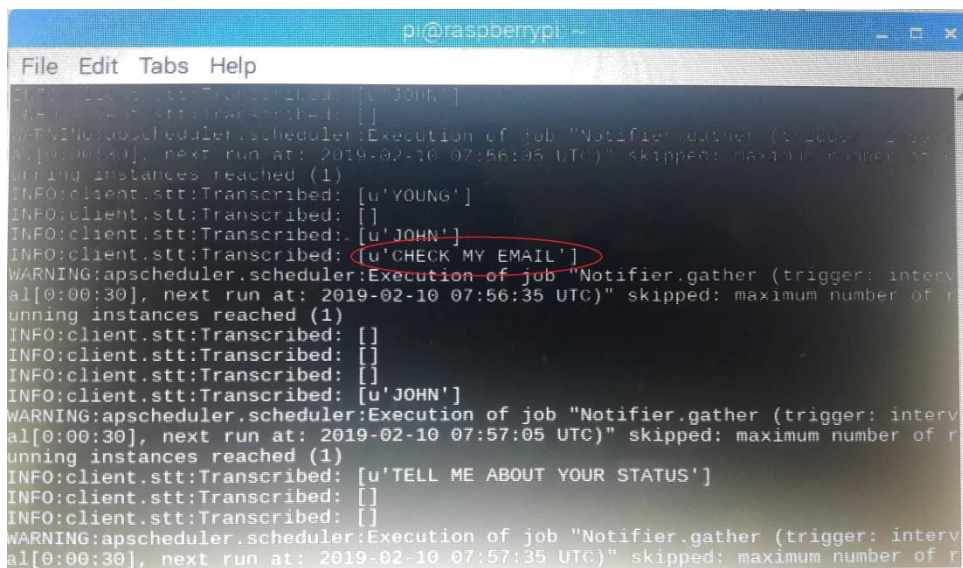
keyword "Event Day" Then Check your google calendar today or tomorrow for events, and it lets you add an event to your calendar. Then this system provide exact event schedule day and reminder of you.

## 6.5 Allocating iPhone

Find my iPhone module lets the users find their iPhone by their voice. This is done using the user's iCloud id and password to make the connection to the server. After the authentication of the password it checks the status of the iPhone. After checking, it connects to iPhone using the iCloud. After the connection, the command for ringing is sent to the iPhone to start a sound and notification on the phone for detection. If a failure occurs, the system generate an error that the iPhone is not found, An error is also generated when there are multiple iPhone associated with a single apple id.

## 6.6 Email checking

Check my email command will perform email information by checking the specific email account which has been set by the user. It can be modified by user choice.



```
pi@raspberrypi ~  
File Edit Tabs Help  
INFO:client.stt:Transcribed: [u'JOHN']  
INFO:client.stt:Transcribed: [u'JOHN']  
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:56:35 UTC)" skipped: maximum number of running instances reached (1)  
INFO:client.stt:Transcribed: [u'YOUNG']  
INFO:client.stt:Transcribed: []  
INFO:client.stt:Transcribed: [u'JOHN']  
INFO:client.stt:Transcribed: [u'CHECK MY EMAIL']  
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:56:35 UTC)" skipped: maximum number of running instances reached (1)  
INFO:client.stt:Transcribed: []  
INFO:client.stt:Transcribed: []  
INFO:client.stt:Transcribed: []  
INFO:client.stt:Transcribed: [u'JOHN']  
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:57:05 UTC)" skipped: maximum number of running instances reached (1)  
INFO:client.stt:Transcribed: [u'TELL ME ABOUT YOUR STATUS']  
INFO:client.stt:Transcribed: []  
INFO:client.stt:Transcribed: []  
WARNING:apscheduler.scheduler:Execution of job "Notifier.gather (trigger: interval[0:00:30], next run at: 2019-02-10 07:57:35 UTC)" skipped: maximum number of running instances reached (1)
```

## 7 Network server and software

### 7.1 Putty

**Putty** is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. The name "PuTTY" has no official meaning.[3]

Putty support many variations on the source remote terminal, and provides user control over the SSH encryption key and protocol version, alternate ciphers such as AES, 3DES, Arcfour, Blowfish, DES, and public-key authentication. Putty supports SSO through GSSAPI, including user provide GSSAPI DLLs. It also can emulate control sequences from xterm, VT220, VT102 or ECMA-48 terminal emulation, and allows local, remote, or dynamic port forwarding with SSH (including X11 forwarding). The network communication layer supports IPv6, and the SSH protocol supports the zlib@openssh.com delayed compression scheme. It can also be used with local serial port connections.

Putty was originally written for Microsoft Windows, but it has been ported to various other operating systems. Official ports are available for some Unix-like platforms, with work-in-progress ports to Classic Mac OS and mac OS, and unofficial ports have been contributed to platforms such as Symbian, Windows Mobile and Windows Phone.[4]

#### 7.1.1Features

### 7.2 VNC

VNC is a graphical desktop sharing system that allows someone to remotely control the desktop interface of one computer from another. It transmits the keyboard and mouse events form the controller, and receive updates to the screen over the network from the remote host. This enables a user to use the desktop of the Pi inside a window on their computer. They will be able to control it as though they were working on Raspberry Pi itself.

VNC is used widely across every industry sector by individuals and organizations for different use cases which include providing IT desktop support to colleagues and friends and also accessing system and services on the move.[5][26]

#### 7.2.1 Features

VNC Server Enterprise Edition builds on the same established VNC core as VNC Server Free Edition, adding improved authentication, session security and logging support.

On all Windows platforms, VNC sessions may be protected by 128-bit AES/EAX authenticated encryption. Secure VNC Servers identify themselves to clients using 2048-bit RSA keys for server-authentication purposes.



Under Windows NT based platforms, VNC sessions may be authenticated using either the classic VNC password mechanism, or against local or domain Windows user accounts, including Active Directory accounts. Under Windows NT based platforms, VNC session logon and logoff events are automatically logged to the Application Event Log of the relevant host computer or domain controller.

**Requirements:**

- System hardware meeting OS requirements
- Windows-compatible graphics card
- Windows-compatible network card
- TCP/IP v4 network stack.

### **7.3 Access Over Internet**

Raspberry Pi can be connected to another computer or a mobile device. One method is to set-up port forwarding. To set-up port forwarding one must change the configuration of their router to forward all inbound traffic from the Internet to a specific port to the local IP address of their 32 Raspberry Pi. One disadvantage of doing this is that it exposes a network port on a private LAN to the public network. This is security vulnerability and must be managed carefully. One secure alternative to port forwarding is the Weaved service. Weaved is software that needs to be install on the Raspberry Pi and it will allow Pi to connect to any device over the Internet.

# 8 System Design

## 8.1 hardware design and development

### 8.1.1 Installing Raspberry pi3

Official images for recommended operating systems are available to download from the Raspberry Pi website. We have used raspberry pi .Raspbian Stretch with desktop and recommended software.

Image with desktop and recommended software based on Debian Stretch

- Version: November 2018
- Release date:2018-11-13
- Kernel version:4.14

Alternative distributions are available from third-party vendors.

### 8.1.2 Burning the SD card

**Etcher** is a graphical SD card writing tool that works on Mac OS, Linux and Windows, and is the easiest option for most users. Etcher also supports writing images directly from the zip file, without any unzipping required.

### 8.1.3 IP Scanning

This system used advanced IP scanner for scan the raspberry pi3. IP which is connected with laptop through Ethernet cable. Advanced IP Scanner is fast and free software for network scanning. It will allow you to quickly detect all network computers and obtain access to them.

[2]

### 8.1.4 Connection through SSH

PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers. By putting the IP address of pi3 we connect the raspberry pi to our laptop through ssh connection.[27]

### 8.1.5 Connect VNC to use laptop as pi monitor

Virtual network computing (VNC) is a type of remote-control software that makes it possible to control another computer over a network connection. Keystrokes and mouse clicks are transmitted from one computer to another, allowing technical support staff to manage a desktop, server, or other networked device without being in the same physical location. This develop connection between raspberry pi to laptop for using the laptop monitor as a pi monitor.[22]

### 8.1.6 Configuring the Raspberry pi3

The Raspberry Pi 3's four built-in USB ports provide enough connectivity for a mouse, keyboard, or anything else that you feel the RPi needs, but if you want to add even more you can still use a USB hub. Keep in mind, it is recommended that you use a powered hub so as not to overtax the on-board voltage regulator. Powering the Raspberry Pi 3 is easy, just plug any USB power supply into the micro-USB port. There's no power button so the Pi will begin to boot as soon as power is applied, to turn it off simply remove power. The four built-in USB ports can even output up to 1.2A enabling you to connect more power hungry USB devices[23][24]

### 8.1.7 Installing Dependencies

To be able to understand what you say, We needs a Speech-to-Text (STT) engine. And we also needs a Text-to-Speech (TTS) engine to answer to your commands. This project's aims to be modular and thus gives you the choice which STT/TTS engine you want to use. Depending on your choice, it may be required to install additional software.[21]

### 8.1.8 STT Engine

We have used wit.ai will work through server alpha as a speech to text engine. Wit.ai makes it easy for developers to build applications and devices that you can talk or text to.

You need to choose which Speech-To-Text (STT) engine john should use. An STT engine is basically a piece of software that takes recorded speech and transforms it into written text. If you say "foo", but john understands "bar", it's either a problem with your microphone or a bad or misconfigured STT engine. So choosing the right STT engine is crucial to use john correctly. While most speech-recognition tools only rely on one single STT engine, john tries to be modular and thus offers a wide variety of STT engines. [10]

```
stt_engine:witai
```

```
witai-stt:
```

```
access_token:ERJKGE86SOMERANDOMTOKEN23471AB
```

### 8.1.9 Offline TTS engine

Google Text-to-Speech is a screen reader application developed by Google for its Android operating system. It powers applications to read aloud (speak) the text on the screen which support many languages. Text-to-Speech may be used by apps such as Google Play Books for reading books aloud, by Google Translate for reading aloud translations providing useful insight to the pronunciation of words, by Google Talkback and other spoken feedback accessibility-based applications, as well as by third-party apps. Users must install voice data for each language.[25]

```
espeak-tts:
```

```
voice:'default+m3'
```

pitch\_adjustment:40

words\_per\_minute:160

## 8.10 Adding Modules

friendly. Here is some module's We added some module that will make our robot more user :that we used for our assistant robot

- Google Calendar – The Google Calendar module lets you check your Google calendar today or tomorrow for events, and it lets you add an event to your calendar.
- Status – The Status module outputs system status information upon request.
- Define – The Define module is used to get definitions of words.
- Wiki – The Wiki module is used to get a Wikipedia summary of a word or phrase.
- Domoticz – The Domoticz home automation module is used to issue commands to a Domoticz server using the JSON API.
- Find My iPhone – The Find My iPhone module is used to force your phone to ring using the apple service “find my iphone”
- Yelp – The yelp module is used to find a good restaurant in a given area. The module can also answer various questions about the restaurant.
- Stocks – The stock module is used to get live stock quotes.
- OpenWeatherMap – Open Weather Map API to get weather forecast.
- Currency Exchange Rate Calculator – Get Xchange Rate for two Currencies.
- MQTT – An MQTT client which allows you to control other MQTT enabled devices in your home.
- Reboot – Reboot the Raspbery PI.
- Selfie – Take a webcam photo and upload it to a predefined email.
- Shutdown – Shutdown the Raspbery PI.
- Speedtest – Run a Speedtest of your network and report back the results.

so perfectly but few have some issues and we are still working to fixed some module works [20].them

## 8.2 Dictionary create word

The CMU Pronouncing Dictionary (also known as CMUdict) is an open-source pronouncing dictionary originally created by the Speech Group at Carnegie Mellon University (CMU) for use in speech recognition research.

CMUdict provides a mapping orthographic/phonetic for English words in their North American pronunciations. It is commonly used to generate representations for speech recognition (ASR), e.g. the CMU Sphinx system, and speech synthesis (TTS), e.g. the Festival system. CMUdict can be used as a training corpus for building statistical grapheme-to-phoneme (g2p) models<sup>[1]</sup> that will generate pronunciations for words not yet included in the dictionary.

lex tool-We created a Word dictionary file by using CMU

### **8.2.1 Creating Profile**

we need a configuration file that we call “profile”. In order for our robot to accurately report local weather conditions, send you text messages, and more, you first need to generate a user profile. Then create a folder and rename as profile.yml. Set your profile information that will be replies related to your model. e.g: weather information, local time-zone etc.

### **8.3 Device interfacing**

The Personal assistant System using robot consists of

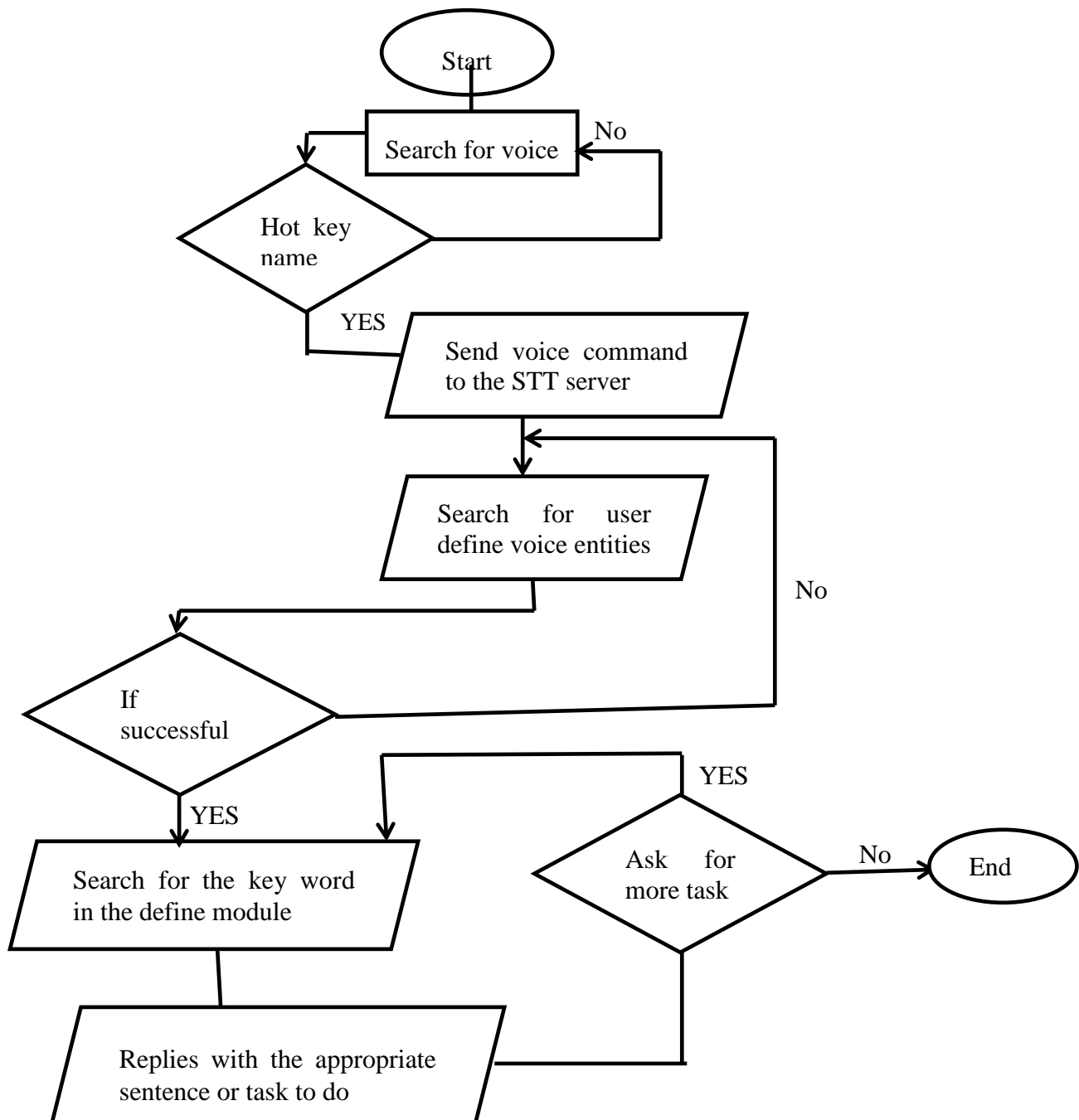
one Raspberry pi 3 model B

one Arduino mega, stereo speaker, one microphone, one Arduino motor shield, one servo motor driver, one voltage converter, two laptop cooling fans, four servo motors, four dc gear motors, two lipo batteries, four wheels, four switches, body chassis, jumper wires and cables, screw, nuts and glue.

## 9 Software Design & Development

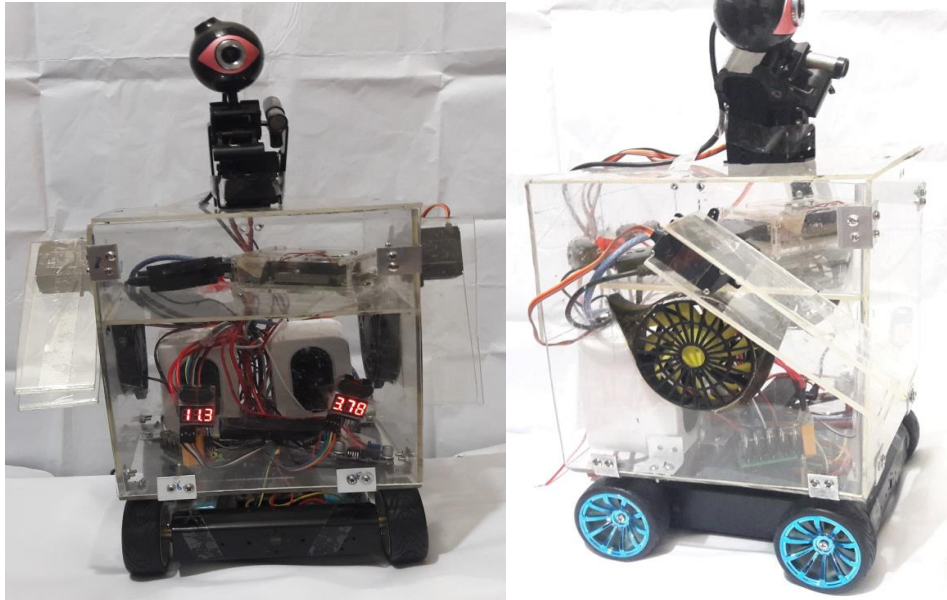
### 9.1 Program Flow

The system architecture is as shown in Figure 3. When the Battery is connected to Raspberry pi the user talk to voice command then robot control the voice input then output the voice other usage services such as restaurant location this system.



## 10 Implementation:

### Structures:



**Figure3.4 : Our Implemented Robot**

We design the structure in raw paper then added the parts into it. We always tried to keep the design lite-weight and easy to reconfigure transparent figure give it more sassy look .

# 11 Conclusion

Voice controlled personal assistant have made their way into our daily lives through software embedded in today's smartphones. Siri, Google voice, Cortana, Alexa are few recognizable names. This project (JHON) extends the benefits of voice recognition and speech processing to robotic. JHON follows a user's voice commands and can carry out per-programmed physicals actions. It can be a waiter, a warehouse assistant or a nurse. At home, it can get you coffee from the kitchen, get your morning newspaper from the front door or take a Selfie when you are partying with friends. The hardware and software can easily be adapted to any new environment. In the future we plan we plan to extent human following capabilities in JHON so that it can server as an assistant in an unknown environment. It can carry luggage on airports and supermarkets, ferry people in wheelchair and kids in prams, all without human-effort and while avoiding obstacles.

## 11.1 Limitations

- Sometimes can't gather the Gmail information because of authentication problem.
- Weather information varies with different time zone so that always we have to up to date the local time zone.
- Sometimes its breakout with the Arduino serial communication.
- It has to hands but can't carry any object because of less robotic arm facilities.
- It needs a strong internet facilities for communicate with online server sites.

## 11.2 Future plan

As for future work, the system will simulate the language environment to adapt to the practical pragmatic environment. In this way, its interactive behavior will be more real. Moreover, we will improve its function further according to the user's feedback. Some new interactive functions will be improved and added, too.

We are still working on including face recognition in it. With that facility our robot will have the ability to recognize people (using machine leaning) and have more comfortable interaction with peoples. Even, if someone wants to customize the robot they will have the full access to do that.



## 12 References

- [1] OSxdaily."How to Write an .img to SD Card on Mac the Easy Way with Etcher". --  
<http://osxdaily.com/2018/01/11/write-img-to-sd-card-mac-etcher/>
- [2] "Advanced IP Scanner.Reliable and free network scanner to analyse LAN."[available at]-  
<https://www.advanced-ip-scanner.com/>
- [3] Simon Tatham."PuTTY is a free implementation of SSH and Telnet for Windows and Unix platforms, along with an xterm terminal emulator." --  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/>
- [4] James,Justin(2009-08-06). "Review:PuTTYterminal application". TechRepublic.Retrieved 2015-09-18.
- [5] Richardson, T.; Stafford-Fraser, Q.; Wood, K. R.; Hopper, A. (1998)."Virtual network computing" (PDF).IEEE Internet Computing. 2: 33. doi:10.1109/4236.656066.--  
<https://ieeexplore.ieee.org/document/656066/>
- [6] Orlowski, Andrew. "Total recog: British AI makes universal speech breakthrough". The Register.Situation Publishing. Retrieved 17 May 2018.--  
[https://www.theregister.co.uk/2017/12/01/british\\_ai\\_makes\\_universal\\_speech\\_breakthrough/](https://www.theregister.co.uk/2017/12/01/british_ai_makes_universal_speech_breakthrough/)
- [7] Dennis H. Klatt (1979). "Software for a cascade/parallel formant synthesizer" (PDF). J. Acoustical Society of America, 67(3) March 1980.
- [8] M. Bisani and H. Ney."Joint-Sequence Models for Grapheme-to-Phoneme Conversion". Speech Communication, Volume 50, Issue 5, May 2008, Pages 434-451
- [9] <http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html>
- [10] "Speaker Independent Connected Speech Recognition- Fifth Generation Computer Corporation". Fifthgen.com. Archived from the original on 11 November 2013. Retrieved 15 June 2013.
- [11] Suk-Hwan Suh; SeongKyoong Kang; Dae-Hyuk Chung; Ian Stroud (22 August 2008). Theory and Design of CNC Systems.Springer Science & Business Media.pp. 11–. ISBN 978-1-84800-336-1. Archived from the original on 21 March 2017.
- [12] Cellan-Jones, Rory (5 May 2011). "A£15 computer to inspire young programmers". BBCNews. --  
[http://www.bbc.co.uk/blogs/thereporters/rorycellanjones/2011/05/a\\_15\\_computer\\_to\\_inspire\\_young.html](http://www.bbc.co.uk/blogs/thereporters/rorycellanjones/2011/05/a_15_computer_to_inspire_young.html)

- [13] David Kushner (2011-10-26). "The Making of Arduino". IEEE Spectrum.--  
<https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>
- [14] H. Seraji, "Motion Control of mobile manipulators," in Proc. IEEE International Conference on Intelligent Robots and Systems, pp. 2056-2063, 1993.
- [15] G. Foulon, J. Y. Fourquet, M. Renaud, "Planning point to point paths for nonholonomic mobile manipulators," in Proc. IEEE International Conference on Intelligent Robots and Systems, pp. 374-379, 1998.
- [16] J. Tan and N. Xi, "Unified model approach for planning and control of mobile manipulators," in Proc. IEEE International Conference on Robotics and Automation,
- [17] vol. 3, pp. 3145–3152, 2001. 7)Z. Bien, M. Chung, P. Chang, D. Kwon, D. Kim, J. Han, J. Kim, H. Park, S. Kang, K. Lee, S. L. Lim, "Integration of a rehabilitation robotic system (KARES II) with human-friendly man-machine interaction units" in Autonomous Robots, MA, Norwood:Kluwer, vol. 16, pp. 165-191, 2004.
- [18] H. Kwee, "Integrated control of MANUS and wheelchair", Proc. Int. Conf. Rehabilitation Robotics (ICORR'97), pp. 91-94, 1997.
- [19] A. Bichhi, G. Tonietti, "Dealing with the safety-performance tradeoff in robot arms design and control", IEEE Robot. Automat.Mag., vol. 11, no. 2, pp. 22-33, 2004.
- [20] M. G. Fujie, "How to commercialize service robots in the 21st century", Proc. 1st IARP/IEEE Workshop Technical Challenge for Dependable Robots in Human Environment, pp. 8-16, 2001.
- [21] Shi F., Cao Q., Leng C., Tan H. Based On Force Sensing-Controlled Human-Machine Interaction System for Walking Assistant Robot; Proceedings of the 8th World Congress on Intelligent Control and Automation; Jinan, China. 7–9 July 2010; pp. 6528–6533.
- [22] Clotet E., Martínez D., Moreno J., Tresanchez M., Palacín J. Collision Avoidance System with Deceleration Control Applied to an Assistant Personal Robot. Trends Pract. Appl. Agents Multi-Agent Syst. Sustain. 2015;372:227–228.
- [23] Stanford, Peter (3 December 2011). "Computing classes don't teach programming skills". The Daily Telegraph. London. Retrieved 27 February 2012
- [24] "Latest Raspberry Pi has double the RAM". BCS website. BCS. 16 October 2012. Retrieved 18 October 2012.
- [25] "Google, Hyundai show off new third-party Android Auto apps". CNET. CBS Interactive. Retrieved 17 January 2015.

- [26] "VNC® Wayland Developer Preview". 2014-07-08. Archived from the original on 2014-07-14. Retrieved 2014-07-10.
- [27] "SSH-CRC-32-Compensation-Attack-Detector Vulnerability". SecurityFocus. Archived from the original on 2008-07-25
- [28] [http://svn.code.sf.net/p/cmuspinx/code/trunk/cmudict/00README\\_FIRST.txt](http://svn.code.sf.net/p/cmuspinx/code/trunk/cmudict/00README_FIRST.txt)
- [29] Mauldin, Michael (1994), "ChatterBots, TinyMuds, and the Turing Test: Entering the Loebner Prize Competition", Proceedings of the Eleventh National Conference on Artificial Intelligence, AAAI Press, retrieved 2008-03-05 (abstract)
- [30] The Shocking Truth About Transformerless Power Supplies
- [31] COHEN, P., CHEYER, A., HOROVITZ, E., EL KALIOUBY, R. & WHITTAKER, S. A Future for Personal Assistants. ACM CHI 2016: Panel Session, San Jose, May 7-12, 2016. Available online at: <http://www.adam.cheyer.com/papers/chi16.pdf> (Retrieved 9th January, 2017).
- [32] MIT Technology Review Custom, PwC. AI Drives Better Business Decisions. MIT Technology Review [online]. 2016, Jun 20. Available online at:
- [33] HEMPEL, Jessi. Voice Is the Next Big Platform, and Alexa Will Own It. Backchannel [online]. 2016, Dec 19. Available online at: <https://backchannel.com/voice-is-the-next-big-platform-and-alexa-will-own-it-c2cf13fab911#.pcmsvex3r> (Retrieved 10th January, 2017).
- [34] SRI Inc. Siri. SRI [online]. Available online at: <https://www.sri.com/work/timeline-innovation/timeline.php?timeline=computing-digital#!&innovation=siri> (Retrieved 15th December)
- [35] Colton, Mark; Ricks, Daniel (June 2010). "Trends and Considerations in Robot-Assisted Autism Therapy" (PDF).
- [36] Darren Murph (2011-10-04). "iPhone 4S hands-on!". Engadget.com. Retrieved 2017-12-10.
- [37] "Amazon Lex, the technology behind Alexa, opens up to developers". TechCrunch. 2017-04-20. Retrieved 2017-12-10.
- [38] Goel, Vaibhava; Byrne, William J. (2000). "Minimum Bayes-risk automatic speech recognition". *ComputerSpeech&Language*. 14 (2):115–135. doi:10.1006/csla.2000.0138. Archived from the original on 25 July 2011. Retrieved 28 March 2011.
- [39] Klatt, D (1987). "Review of text-to-speech conversion for English". *Journal of the Acoustical Society of America*. 82 (3): 737–93. doi:10.1121/1.395275.

- [40] Szczepaniak, John (2014). The Untold History of Japanese Game Developers. 1. SMG Szczepaniak. pp. 544–615. ISBN 978-0992926007.
- [41] Muralishankar, R.; Ramakrishnan, A. G.; Prathibha, P. (February 2004). "Modification of pitch using DCT in the source domain". *Speech Communication*. 42 (2): 143–154
- [42] "Raspberry Pi FAQs – Frequently Asked Questions". Raspberry Pi. Retrieved 8 April 2017.
- [43] "Arduino's New CEO, Federico Musto, May Have Fabricated His Academic Record". WIRED. Retrieved 2017-12-22.
- [44] Williams, Elliot (28 March 2015). "Arduino SRL to Distributors: "We're the Real Arduino"". Hackaday.com. Hackaday.com. Retrieved 21 April 2015.

## 13 Appendix

We have used python language code in our personal assistant system. The codes are given below:

```
#!/usr/bin/env python2
```

```
# -*- coding: utf-8 -*-
```

```
import os
```

```
import sys
```

```
import shutil
```

```
import logging
```

```
import yaml
```

```
import argparse
```

```
from client import tts
```

```
from client import stt
```

```
from client import johnpath
```

```
from client import diagnose
```

```
from client.conversation import Conversation
```

```
# Add johnpath.LIB_PATH to sys.path
```

```
sys.path.append(johnpath.LIB_PATH)
```

```
parser = argparse.ArgumentParser(description='John Voice Control Center')
```

```
parser.add_argument('--local', action='store_true',
```

```
help='Use text input instead of a real microphone')
```

```
parser.add_argument('--no-network-check', action='store_true',
```

```
help='Disable the network connection check')
```

```
parser.add_argument('--diagnose', action='store_true',
```

```
help='Run diagnose and exit')
```

```

parser.add_argument('--debug', action='store_true', help='Show debug messages')
args = parser.parse_args()

if args.local:
    from client.local_mic import Mic
else:
    from client.mic import Mic

class John(object):
    def __init__(self):
        self._logger = logging.getLogger(__name__)

        # Create configdir if it does not exist yet
        if not os.path.exists(johnpath.CONFIG_PATH):
            try:
                os.makedirs(johnpath.CONFIG_PATH)
            except OSError:
                self._logger.error("Could not create configdir: '%s'",
                                   johnpath.CONFIG_PATH, exc_info=True)
                raise

        # Check if configdir is writable
        if not os.access(johnpath.CONFIG_PATH, os.W_OK):
            self._logger.critical("Configdir %s is not writable. John " +
                                   "won't work correctly.",
                                   johnpath.CONFIG_PATH)

        # FIXME: For backwards compatibility, move old config file to newly
        #         created configdir
        old_configfile = os.path.join(johnpath.LIB_PATH, 'profile.yml')
        new_configfile = johnpath.config('profile.yml')
        if os.path.exists(old_configfile):

```

```

if os.path.exists(new_configfile):
    self._logger.warning("Deprecated profile file found: '%s'. " +
                          "Please remove it.", old_configfile)
else:
    self._logger.warning("Deprecated profile file found: '%s'. " +
                          "Trying to copy it to new location '%s'.",
                          old_configfile, new_configfile)
try:
    shutil.copy2(old_configfile, new_configfile)
except shutil.Error:
    self._logger.error("Unable to copy config file. " +
                      "Please copy it manually.",
                      exc_info=True)
    raise

    # Read config
    self._logger.debug("Trying to read config file: '%s'", new_configfile)
try:
    with open(new_configfile, "r") as f:
        self.config = yaml.safe_load(f)
except OSError:
    self._logger.error("Can't open config file: '%s'", new_configfile)
    raise

try:
    stt_engine_slug = self.config['stt_engine']
except KeyError:
    stt_engine_slug = 'sphinx'
    logger.warning("stt_engine not specified in profile, defaulting " +
                  "to '%s'", stt_engine_slug)
stt_engine_class = stt.get_engine_by_slug(stt_engine_slug)

```

```

try:
    slug = self.config['stt_passive_engine']
    stt_passive_engine_class = stt.get_engine_by_slug(slug)
except KeyError:
    stt_passive_engine_class = stt_engine_class

try:
    tts_engine_slug = self.config['tts_engine']
except KeyError:
    tts_engine_slug = tts.get_default_engine_slug()
    logger.warning("tts_engine not specified in profile, defaulting " +
                   "to '%s'", tts_engine_slug)
    tts_engine_class = tts.get_engine_by_slug(tts_engine_slug)

    # Initialize Mic
    self.mic = Mic(tts_engine_class.get_instance(),
                   stt_passive_engine_class.get_passive_instance(),
                   stt_engine_class.get_active_instance())

def run(self):
    if 'first_name' in self.config:
        salutation = ("How can I be of service, %s?"
                      % self.config["first_name"])
    else:
        salutation = "How can I be of service?"
    self.mic.say(salutation)

    conversation = Conversation("JOHN", self.mic, self.config)
    conversation.handleForever()

if __name__ == "__main__":

```



```
print("*****")
print("      john - personal assistant robot  ")
```

```
print("*****")
```

```
logging.basicConfig()
logger = logging.getLogger()
logger.getChild("client.stt").setLevel(logging.INFO)
```

```
ifargs.debug:
logger.setLevel(logging.DEBUG)
```

```
if not args.no_network_check and not diagnose.check_network_connection():
logger.warning("Network not connected. This may prevent john from " +
               "running properly.")
```

```
ifargs.diagnose:
failed_checks = diagnose.run()
sys.exit(0 if not failed_checks else 1)
```

```
try:
app = john()
except Exception:
logger.error("Error occurred!", exc_info=True)
sys.exit(1)
```

```
app.run()
```

```
.....
.....
```

```
# -*- coding: utf-8 -*-
```

```
import re

def detectYears(input):
    YEAR_REGEX = re.compile(r'(\b)(\d\d)([1-9]\d)(\b)')
    return YEAR_REGEX.sub("\g<1>\g<2> \g<3>\g<4>", input)
```

```
def clean(input):
    """
    Manually adjust output text before it's translated into
    actual speech by the TTS system. This is to fix minor
    idiomatic issues, for example, that 1901 is pronounced
    "one thousand, ninehundred and one" rather than
    "nineteen oh one".
```

```
    Arguments:
    input -- original speech text to-be modified
    """
    return detectYears(input)
```

.....

.....

```
# -*- coding: utf-8 -*-

import smtplib
from email.MIMEText import MIMEText
import urllib2
import re
from pytz import timezone
```

```
defsendEmail(SUBJECT, BODY, TO, FROM, SENDER, PASSWORD,
SMTP_SERVER):
```

```
    """Sends an HTML email."""
```

```
    forbody_charset in 'US-ASCII', 'ISO-8859-1', 'UTF-8':
```

```
    try:
```

```
        BODY.encode(body_charset)
```

```
    exceptUnicodeError:
```

```
        Pass
```

```
    else:
```

```
        Break
```

```
    msg = MIMEText(BODY.encode(body_charset), 'html', body_charset)
```

```
    msg['From'] = SENDER
```

```
    msg['To'] = TO
```

```
    msg['Subject'] = SUBJECT
```

```
    SMTP_PORT = 587
```

```
    session = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
```

```
    session.starttls()
```

```
    session.login(FROM, PASSWORD)
```

```
    session.sendmail(SENDER, TO, msg.as_string())
```

```
    session.quit()
```

```
defemailUser(profile, SUBJECT="", BODY=""):
```

```
    """
```

```
    sends an email.
```

```
    Arguments:
```

```
    profile -- contains information related to the user (e.g., email
```

```

address)

    SUBJECT -- subject line of the email

    BODY -- body text of the email

    """

def generateSMSEmail(profile):
    """
    Generates an email from a user's phone number based on their carrier.

    """

    if profile['carrier'] is None or not profile['phone_number']:
        return None

    returnstr(profile['phone_number']) + "@" + profile['carrier']

    if profile['prefers_email'] and profile['gmail_address']:
        # add footer
    if BODY:
        BODY = profile['first_name'] + \
            ",<br><br>Here are your top headlines:" + BODY
        BODY += "<br>Sent from your john"

    recipient = profile['gmail_address']
    if profile['first_name'] and profile['last_name']:
        recipient = profile['first_name'] + " " + \
            profile['last_name'] + " <%s>" % recipient
    else:
        recipient = generateSMSEmail(profile)

    if not recipient:
        return False

```

```

try:
    if 'mailgun' in profile:
        user = profile['mailgun']['username']
        password = profile['mailgun']['password']
        server = 'smtp.mailgun.org'
    else:
        user = profile['gmail_address']
        password = profile['gmail_password']
        server = 'smtp.gmail.com'
    sendEmail(SUBJECT, BODY, recipient, user,
              "john <john>", password, server)

return True

except:
    return False

```

```
defgetTimezone(profile):
```

```
    """
```

Returns the pytztimezone for a given profile.

Arguments:

profile -- contains information related to the user (e.g., email address)

```
    """
```

```

try:
    returntimezone(profile['timezone'])
except:
    return None

```

```
defgenerateTinyURL(URL):
```

```
    """
```

Generates a compressed URL.

Arguments:

URL -- the original URL to-be compressed

```
    """
```

```
target = "http://tinyurl.com/api-create.php?url=" + URL
```

```
response = urllib2.urlopen(target)
```

```
returnresponse.read()
```

```
defisNegative(phrase):
```

```
    """
```

Returns True if the input phrase has a negative sentiment.

Arguments:

phrase -- the input phrase to-be evaluated

```
    """
```

```
returnbool(re.search(r'\b(no(t)?|don\'t|stop|end)\b', phrase,
```

```
re.IGNORECASE))
```

```
defisPositive(phrase):
```

```
    """
```

Returns True if the input phrase has a positive sentiment.

Arguments:

phrase -- the input phrase to-be evaluated

```
    """
```

```
returnbool(re.search(r'\b(sure|yes|yeah|go)\b', phrase, re.IGNORECASE))
```

```
.....
```

.....

```
# -*- coding: utf-8 -*-
```

```
import logging
```

```
import pkgutil
```

```
import johnpath
```

```
class Brain(object):
```

```
def __init__(self, mic, profile):
```

```
    """
```

Instantiates a new Brain object, which cross-references user input with a list of modules. Note that the order of brain.modules matters, as the Brain will cease execution on the first module that accepts a given input.

Arguments:

mic -- used to interact with the user (for both input and output)

profile -- contains information related to the user (e.g., phone number)

```
    """
```

```
self.mic = mic
```

```
self.profile = profile
```

```
self.modules = self.get_modules()
```

```
self._logger = logging.getLogger(__name__)
```

```
@classmethod
```

```
def get_modules(cls):
```

```
"""
```

Dynamically loads all the modules in the modules folder and sorts them by the `PRIORITY` key. If no `PRIORITY` is defined for a given module, a priority of 0 is assumed.

```
"""
```

```
logger = logging.getLogger(__name__)
locations = [johnpath.PLUGIN_PATH]
logger.debug("Looking for modules in: %s",
             ', '.join(["%s" % location for location in locations]))
modules = []
for finder, name, ispkg in pkgutil.walk_packages(locations):
    try:
        loader = finder.find_module(name)
        mod = loader.load_module(name)
    except:
        logger.warning("Skipped module '%s' due to an error.", name,
                       exc_info=True)
    else:
        ifhasattr(mod, 'WORDS'):
            logger.debug("Found module '%s' with words: %r", name,
                         mod.WORDS)
            modules.append(mod)
        else:
            logger.warning("Skipped module '%s' because it misses " +
                           "the WORDS constant.", name)
modules.sort(key=lambda mod: mod.PRIORITY if hasattr(mod, 'PRIORITY')
             else 0, reverse=True)
return modules
```



```

def query(self, texts):
    """
    Passes user input to the appropriate module, testing it against
    each candidate module's isValid function.

    Arguments:
    text -- user input, typically speech, to be parsed by a module
    """

    for module in self.modules:
        for text in texts:
            if module.isValid(text):
                self._logger.debug("%s' is a valid phrase for module " +
                                   "%s", text, module.__name__)
            try:
                module.handle(text, self.mic, self.profile)
            except Exception:
                self._logger.error('Failed to execute module',
                                   exc_info=True)
                self.mic.say("I'm sorry. I had some trouble with " +
                             "that operation. Please try again later.")
            else:
                self._logger.debug("Handling of phrase '%s' by " +
                                   "module '%s' completed", text,
                                   module.__name__)
            finally:
                return

            self._logger.debug("No module was able to handle any of these " +
                              "phrases: %r", texts)

```

.....

.....

```
# -*- coding: utf-8 -*-

import logging
from notifier import Notifier
from brain import Brain


class Conversation(object):

    def __init__(self, persona, mic, profile):
        self._logger = logging.getLogger(__name__)
        self.persona = persona
        self.mic = mic
        self.profile = profile
        self.brain = Brain(mic, profile)
        self.notifier = Notifier(profile)

    def handleForever(self):
        """
        Delegates user input to the handling function when activated.
        """
        self._logger.info("Starting to handle conversation with keyword '%s'.",
                           self.persona)
        while True:
            # Print notifications until empty
            notifications = self.notifier.getAllNotifications()
            for notif in notifications:
                self._logger.info("Received notification: '%s'", str(notif))
```

```

self._logger.debug("Started listening for keyword '%s'",
self.persona)

threshold, transcribed = self.mic.passiveListen(self.persona)

self._logger.debug("Stopped listening for keyword '%s'",
self.persona)

if not transcribed or not threshold:

self._logger.info("Nothing has been said or transcribed.")

continue

self._logger.info("Keyword '%s' has been said!", self.persona)

self._logger.debug("Started to listen actively with threshold: %r",
threshold)

input = self.mic.activeListenToAllOptions(threshold)

self._logger.debug("Stopped to listen actively with threshold: %r",
threshold)

if input:

self.brain.query(input)

else:

self.mic.say("Pardon?")

```

.....

.....