INTELLIGENT SYSTEM BASED ON SPEECH RECOGNITION WITH CAPABILITY OF SELF-LEARNING

Ms. Sneha K. Upadhyay[1] (M.E. Student), Mr. Vijay N. Chavda[2] (Assistant Professor)
[1]VLSI and Embedded System Design, GTU PG School, Ahmedabad, India.
[2]Government Engineering College, Sector-28, Gandhinagar, India.

Abstract: In this recent era of innovations and developments in various technologies, our life has become more sophisticated and very easy. New and smart devices are being introduced every day. as we know there are many applications that can be used as text to speech translator but it can be more smart if we also convert speech data into text as well as. Moreover, if we are going to search something on search engine and as an output we can get all the possible way of search in text generated way then it can be more difficult for specially blind people to recognize the output. The specific application is designed to recognize the speech spoken by a person and represent optimized data that best shows the meaning of that speech form. The same data can be also converted into text by means of GUI application and display on desktop/laptop or any other display device connected to Raspberry Pi Board. Smart arrangements of solenoid plates can be useful for easily recognize the output for blind people. Objective is to design a device which makes life easier for blind people to search out anything. Design of speech browser with using raspberry pi module and on Raspbian platform will be used for voice based Google search and the output of that search will be given to solenoid plates and these plates are useful to recognize that search output by hand touch of blind people.

Keywords: Speech Recognition, Raspberry Pi Module, Raspbian Platform, Solenoid plates.

I. INTRODUCTION

The principle behind this prototype is based on speech technology. There are two types of speech technology. Speech synthesis is computer-generated simulation of human speech. It is used to translate written information into aural information where it is more convenient, especially for mobile applications such as voice-enabled e-mail and unified messaging. Synthesized speech can be created by concatenating pieces of recorded speech to create complete sentence. Speech Recognition is a technique to analyze the speech spoken by a person that how it is differ to other personas with any approach to voice recognition, the first step is for the user to speak a word or phrase into a microphone. The electrical signal from the microphone is digitized by an "analog-to-digital (A/D) converter", and is stored in memory.

To determine the "meaning" of this voice input, the computer attempts to match the input with a digitized voice sample, or template that has a known meaning. This allows the user to control any device by voice, rather than having to use the mouse and keyboard. The embedded system which is converting the speech data into text and by using Google speech API the voice search will be done. The output of voice search will be stored into database and that searched data will be interface with solenoid plates by means of GPIO pins of raspberry pi. For this purpose, the main task is to make GUI which is based on voice search. To accomplish this purpose, speech browser is made by using Embedded/QT creator tool which is available in raspberry pi tool chain. This speech browser contains record button, by pressing the button any person can input a speech and the result based on voice search will be shown in the window same as like Google search. To satisfy this purpose Google speech API is used. And the result of search will be sent to GPIO pins. And that pins are connected to various solenoid plates.

II. SYSTEM ALGORITHM

The need to design this application is specifically for physically impaired people. Objective is to design an intelligent system based on speech recognition which will work according to voice search. Solenoid plates are combination of six dots and braille code is also a combination of six dots. Different output on plates shows the different characters according to braille script so that blind people can easily console with this application. Solenoid plates which are going to be used are of Lee series 120 solenoid valves. It's a UK based company which produces solenoid valves. Its world's smallest solenoid valve. The Lee Series 120 Solenoid Valve sets a new standard in reducing space, weight and power consumption. The valve requires only a momentary 1 ms pulse to switch the state (+5 VDC to open,-5 VDC to close).

These plates are representing output of web search into braille. This browser is developed or designed by software which is based on linux. In linux environment any application is able to run at a quite faster rate.Linux based raspberry pi module will be very efficient to overcome the requirement. There are many linux based OS and among them Ubuntu is sufficient for my research work and has some advantage to use this. Ubuntu is user-friendly. For a beginner it is an absolute test. It is both official and user contributed. It has wealth of documentation and also has many more features. The software is used to design a speech browser is QT creator which is used to make efficient GUI application.

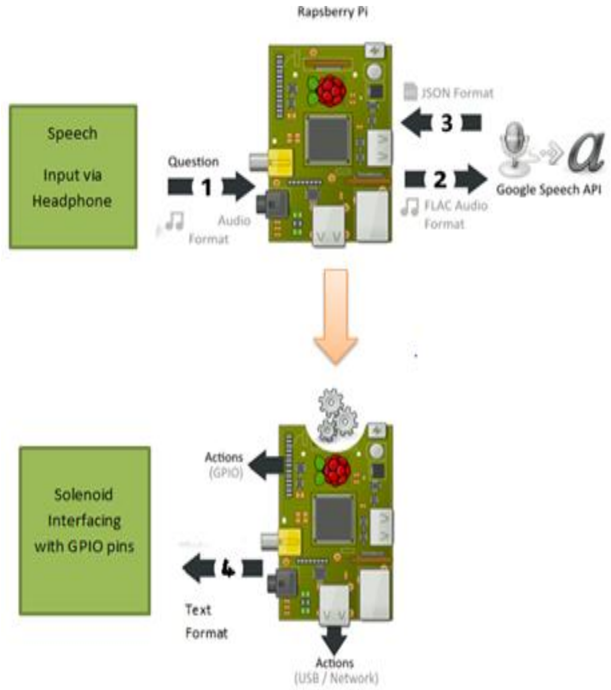


Fig1. Algorithm analysis of Application using raspberry pi

Qt Creator is a good example of an application that mixes different user interface technologies. In fact, it uses all of the three different approaches described below. Qt Creator uses the traditional Qt Widgets such as menus and dialogs as a basis of the user interface, Qt Quick amongst others for the welcome screen, and Qt WebKit for presenting the Qt reference documentation. Qt Creator includes a project manager that uses a cross platform project file format (.pro). A project file can contain information such as what files are included into the project, custom build steps and settings for running the applications. Qt Creator includes a code editor and integrates Qt Designer for designing and building graphical user interfaces (GUIs) from Qt widgets. The code editor can parse code in C++ and QML languages... It is possible to compose and customize the widgets or dialogs and test those using different styles and resolutions directly in the editor. Widgets and forms created with Qt Designer are integrated with programmed code, using the Qt signals and slots mechanism. The audio is collected from the headphone, and then passed via an HTTPS POST to a Google web service, which responds with a JSON object with the results. Looking through their audio encoder code, the audio can be either FLAC or Wav file. JSON (JavaScript Object Notation) is a lightweight data interchange format. . It can represent integer, real number, string, an ordered sequence of value, and a collection of name/value pairs. QJson is a Qt-based library that maps JSON data to Qvariant object. UTF-8 (UCS Transformation Format—8-bit) is a variable-width encoding that can represent every character in the Unicode character set. UTF-8 has become the dominant character encoding for the World Wide Web, accounting for more than half of all Web pages. The output from the code will be into text and that will directly call through Google webpage and which will go for further processing via GPIO pins and these solenoid plates are controlled through GPIO pins. Output from the Google search results are represented by matrix of solenoid plates.

III. IMPLEMENTATION

A. Implementation of speech recognition on Raspberry pi Terminal

The Implementation process for converting speech to text by using Raspberry pi and on the terminal of Raspbian image installed on raspberry pi. This is the basic step to understand that how can we convert any voice data into text and this will be helpful to understand the commands of linux. The Raspberry Pi is an ultra-low-cost, deck-of-cards sized Linux computer. It is controlled by a modified version of Debian Linux optimized for the ARM architecture. It has two models model A and model B. The Model B has 512 MB RAM, BCM2385 ARM11, 700 MHz System on chip processor. It has 2 USB ports, HDMI out, audio output jack and Ethernet port for internet access and the USB sound card is needed to interface with pi model because it has only audio output jack and so by using sound card we can input voice data into pi.

- Requirements for this implementation: The latest version of Raspbian installed
- An internet connection
- The correct sound card drivers for your headset. So the required inputs connections are shown below in the Figure 2.

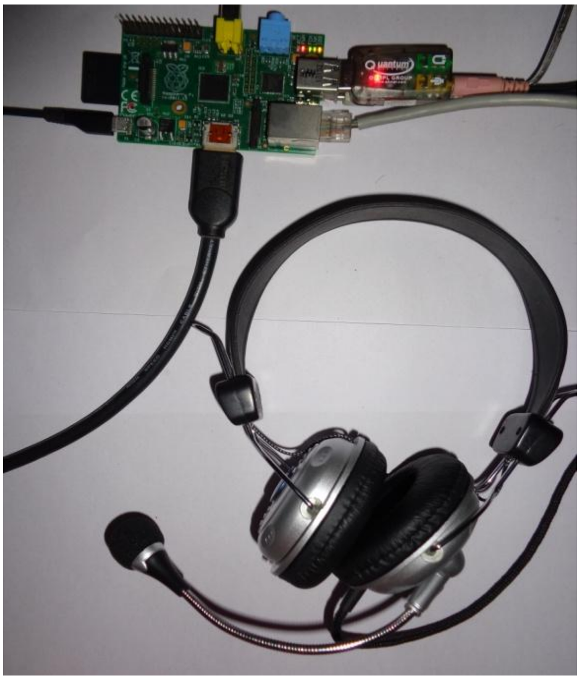
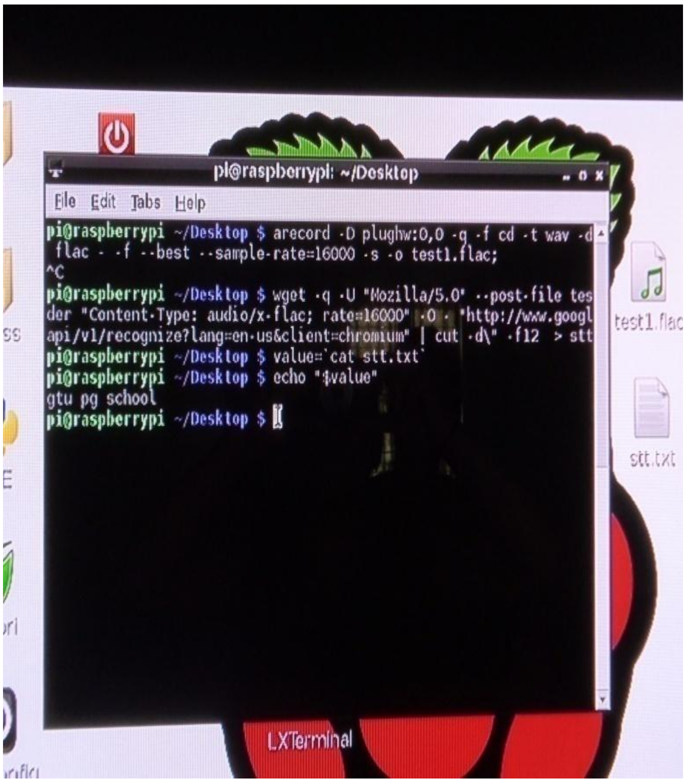


Fig. 2.Hardware connections with Raspberry Pi

Plug in your USB Headset and run the following commands to test the headset.

```
cat /proc/asound/cards
cat /proc/asound/module
```

Raspberry Pi does not have an audio input device configured. To solve this, I need to update ALSA (Advanced Linux Sound Architecture) to use our Headset as default for audio input and output. This can be done by a quick change to the ALSA config file located in /etc/modprobe.d/alsa-base.conf. After recording the audio file, apply a set of commands and the output of speech to text conversion will store into .txt file. Here, the recorded data is gtu pg school.

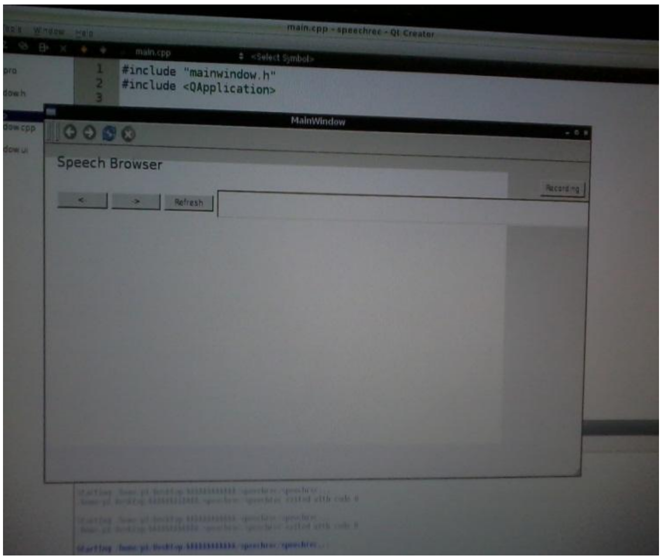


```
pi@raspberrypi ~/Desktop $ arecord -D plughw:0,0 -q -f cd -t wav -d
 flac - -f --best --sample-rate=16000 -s -o test1.flac;
^C
pi@raspberrypi ~/Desktop $ wget -q -U "Mozilla/5.0" --post-file tes
der "Content-Type: audio/x-flac; rate=16000" -O - "http://www.googl
api/v1/recognize?lang=en-us&client=chromium" | cut -d\" -f12  > stt
pi@raspberrypi ~/Desktop $ value=`cat stt.txt`
pi@raspberrypi ~/Desktop $ echo "$value"
gtu pg school
pi@raspberrypi ~/Desktop $ 
```

Fig. 3. Output of Google speech to text on Raspbian

B. Implementation of Speech Browser using software analysis

QT Creator is the software which is used to design GUI application of speech browser. By only after button press anyone can record a speech and after a time period of 3seconds the speech input will automatically saved. At the result by using Google speech API the browser will shows the search data and furthermore, if we want to search out other data then we have to give voice input "forward" by only after button press to the headset connected with raspberry pi. Similarly if we want to go back to the previous search then simply after button press speak "back" and we can see previous searched output. So this application is used for normal people and as well as for blind people also. Because the searched data will be sent to GPIO pins by using QT creator software and that general purpose input output pins are connected with solenoid plates which will moves up if it gets "1" on pin and settled down if it gets "0" on GPIO. The browser is as shown below.

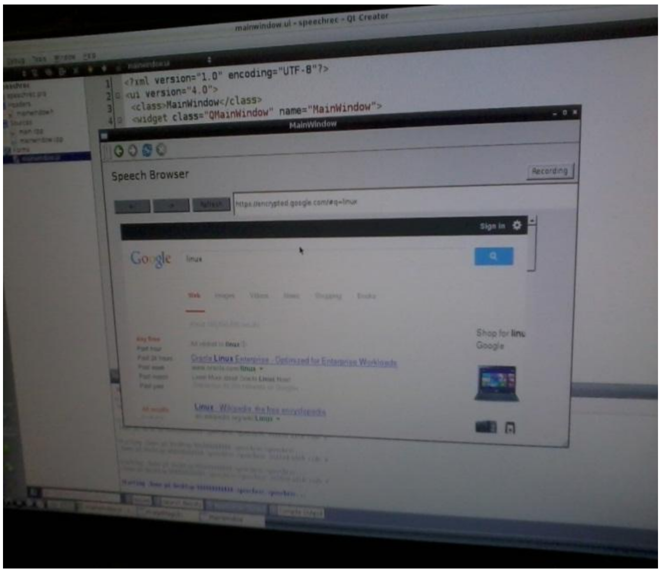


```
#include "mainwindow.h"
#include <QApplication>
```

Fig. 4. Speech browser on Qt creator tool

IV. RESULT

As discussed in the implementation part, Speech browser will only get the data by voice input. After pressing button 3seconds will be counted and whatever input is to spoken by a user will be search out using Google speech API. Suppose here "linux" is the spoken data and the output is as shown below. Speech Browser will show the result of id if it will not able to detect anything via headset or after 3seconds, the user hasn't spoken anything. As shown in Figure5 the first search output is Oracle Linux Enterprise. So that will be sent to the GPIO pins of Pi. And the GPIO pins are getting correct data of the results one by one. The data is the combination of six signals because braille code is combination of six dots. It is dependent on which character is displayed on the search.


mainwindow.ui - speechrec - Qt Creator
MainWindow
Speech Browser
Recording
https://encrypted.google.com/#q=linux
Sign in
Google
linux

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>MainWindow</class>
 <widget class="QMainWindow" name="MainWindow">
```

Fig. 5. Google search result on Speech Browser

V. CONCLUSION

Physically invisible people experience difficulty and inconvenience using computers through a keyboard and mouse. The purpose of this system is to provide a way the blind people population can easily control many functions of a computer via speech. When blind people speak, the audio voice input is sent to the speech Browser .solenoid plated are very useful to convert this web search into braille. Many applications running on this purpose but not all the applications able to fulfil over it and this project has better aspects in future for normal people as well as blind people. This application is firstly embedded on raspberry pi and Qt creator is the software which is being useful to interface this GUI with the hardware connected to Pi.

VI. ACKNOWLEDGEMENT

I hereby would like to acknowledge the invaluable support of my project guide Prof. Vijay N. Chavda. He has helped me through his invaluable suggestions and support throughout the duration of my thesis and have been an unending source of inspiration for me. I take this opportunity to express my gratitude towards him for his useful comments, remarks and engagement through the learning process of this master thesis.

REFRENCES

[1] S. Kumar, M. A. Qadeer and A. Aupta, "Location Based Service using Android", Internet Multimedia Service Architecture and Applications, IEEE international Conference, 2009.

[2] P. Tsiakoulis, A. Chalamandaris, S. Karabetsos and S. Raptis, "A statistical method for database reduction for embedded unit selection speech synthesis", IEEE ICASSP 2008, pp. 4601-4604, 2008.

[3] S. Karabetsos, P. Tsiakoulis, A. Chalamandaris, and S. Raptis, "Embedded unit selection text-to-speech synthesis for mobile devices," IEEE Trans. on Consumer Electronics, vol. 55, no. 2, pp. 613-621, 2009.

[4] Anuraj Khare, "ARM Processor Based Multiple Languages E-Dictionary for Blind People", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 8, August – 2013.

[5] Jae Sung Cha, Dong Kyun Lim and Yong-Nyuo Shin, "Design and Implementation of a Voice Based Navigation for Visually Impaired Persons", International Journal of Bio-Science and Bio-Technology Vol. 5, No. 3, June, 2013.

[6] Jasmin Blanchette and Mark Summer field I, C++ GUI Programming with Qt4 (2nd Edition). New Jersey: Prentice Hall, February 2008, pp. 279-280.

[7] N. Markus, S. Malik, Z. Juhasz and A. Arato, "Accessibility for the Blind on an Open-Source Mobile Platform: Mobile Slate Talker (MOST) for Android", Lecture Notes in Computer Science, vol. 7383, 2012.

[8] G. E. Lee and J. W. Lee, "Google Android phone Personal open market", Korean Multimedia Society, Fall Conference, 2009, pp. 346-349.

[9] Robert Batusek and Ivan Kopecek, "User Interfaces for the Visually Impaired people", Masaryk University, 2000.

[10] V.D. Kanna and S. Aswin Amirtharaj, P.V. Deepak Govind Prasad, N. Sriram Prabhu "Design of a FPGA based Virtual Eye for the Blind", 2nd International Conference on Environmental Science and Technology IPCBEE vol.6, 2011.