

Low cost speech recognition system running on Raspberry pi to support automation applications

Santhoshkumar G¹, MNVLM Krishna²

¹M. Tech student, ²Assistant Professor,

Embedded systems & Department of E I E & GITAM University, Visakhapatnam, Andhra Pradesh, INDIA

Abstract—This paper deals with the implementation of portable voice-based authentication system by using Bluetooth with the help of android application. Existing methods are expensive and also speech recognition is available with different techniques but here we are using our own android app for speech recognition technology and Raspberry Pi for controlling Purpose. Human voice converted into text by using Android app and wirelessly transferred to Bluetooth module which is connected to Raspberry Pi. According to text the corresponding application will works.

Keywords—Raspberry Pi, HC-05 Bluetooth module, Android App

I. INTRODUCTION

The most important element of this project is that we can run on a low cost embedded device such as a Raspberry pi, and for voice recognition better solution here we used is our own Android app and the communication made possible by using Bluetooth module. At first the Smartphone recognizes our voice with the help of Android app and converts into text, and the text transferred through Bluetooth of phone to external HC-05 Bluetooth module which was connected to Raspberry pi board then the Bluetooth transfers the message to our target board serially with half duplex communication and the Raspberry Pi searches the command given by us. If it found then the corresponding will works. For this work we have to implement the code by using python/Linux commands or Embedded C. Based on the particular code our voice command works if it has the command. It is very helpful to implement a group of projects on single board and that to works on voice, main purpose of the project is to implement a small laboratory prototype of electronic projects to the students. Not only lab projects it might be helpful for automation applications.

II. SYSTEM OVERVIEW

The main elements of this project are that Raspberry Board, Bluetooth module HC-05 and smart phone with android app. Here Raspberry works for controlling purpose and Bluetooth is for communication unit. Android app is for receiving the input voice signals.

A. RASPBERRY PI

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. Raspberry pi has ability to interact with outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. In this Project we are using Raspberry Pi B board and those specs are as follows

- **Memory** : 512 Mb RAM
- **Connections**: 2 USB ports
 - Ethernet port(10/100 BaseT Ethernet socket
 - 3.5mm jack for audio out
 - HDMI
 - composite video
- **Processor**: SoC is a Broadcom BCM2835
 - This contains an ARM1176JZF5 (ARM11 using an ARMv6-architecture core) with floating point, running at 700Mhz, and a Video core 4 GPU.
- **OS**: Boots from SD card, running a version of the Linux(Rasbian) operating system
- **Power**: Micro USB socket 5V, 2A
- **26** GPIO pins

As the operating system Raspbmc distribution was used (<http://www.raspbmc.com/>) (aunque tests with Raspbian distribution were also made without any problem). In order to run the operating system, programs and storing multimedia content, an SD card 8 GB Class 10 Regarding the RAM used by each module was used, tests were conducted and it was determined that the media center used to 70 MB while a movie (<http://www.bigbuckbunny.org/>) format 1080p H.264 hardware decoding and reproduced, while the authentication system with all modules and arrays came to use up to 6 MB,

and server using 1 MB. This confirmed that the 512 MB available were sufficient, so the final system memory was partitioned so that the GPU could use 256 MB and leave the rest to the CPU.

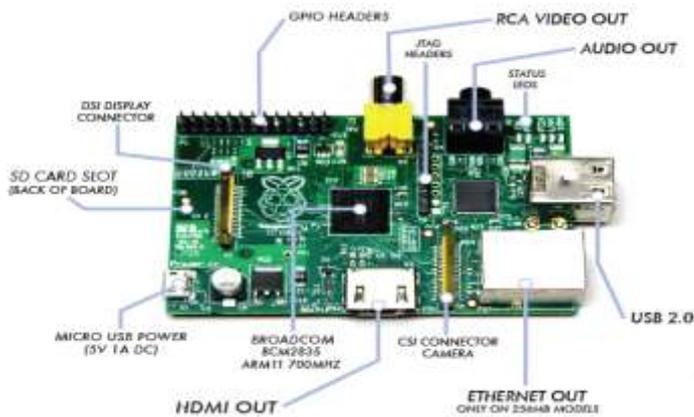


Fig .1. Raspberry Pi B board

B. BLUETOOTH MODULE

Bluetooth Smart technology is a wireless communications system intended to replace the cables connecting many types of devices, from mobile phones and headsets to hear monitors and medical equipment. In my project I am using another module for communication between voices from smart phone to raspberry pi by using Bluetooth module. For this purpose **HC-05** module being used and the specifications of Bluetooth module as follows.

- **Bluetooth protocol:** Bluetooth Specification v2.0+EDR
- **Frequency:** 2.4GHz ISM band
- **Modulation:** GFSK(Gaussian Frequency Shift Keying)
- **Emission power:** ≤4dBm, Class 2
- **Sensitivity:** ≤-84dBm at 0.1% BER
- **Speed:** Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- **Security:** Authentication and encryption
- **Profiles:** Bluetooth serial port
- **Power supply:** +3.3VDC 50mA

Working temperature: -20 ~ +75Centigrade

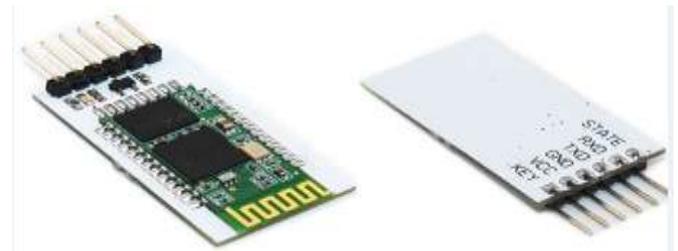


Fig 2 :HC-05 Bluetooth Module

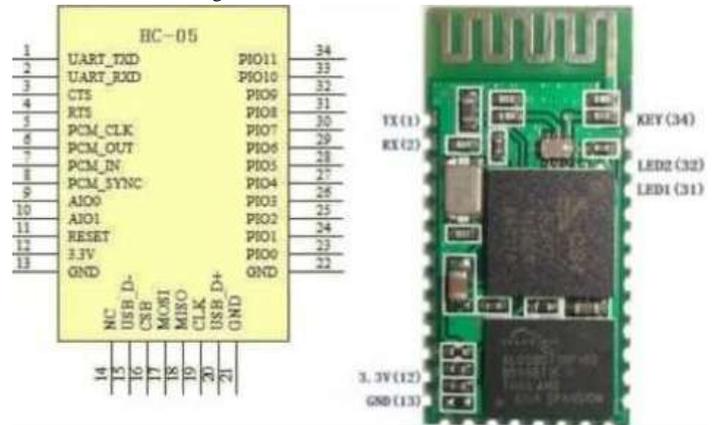


Fig.3. Bluetooth Pin Out Diagram

We have to connect this Bluetooth module to Raspberry pi to control our requirements based on our input given by us through the app. Here Communication applied based on half duplex communication. And the key element in this project is voice so we have to discuss about the Android app for transmission of voice signals.

C. ANDROID APPLICATION

Many apps are existed in Google play store but we can't modify the background process so here we are using our own android application which is prepared using MIT app inventor 2. By using this we can modify background operation based on our requirements so corresponding person have complete idea about what are the inputs. So this is the major advantage of this project.

Steps to create Application:

- Go to www.google.com
- Search MIT app inventor2
- Click on MIT app inventor 2 Login with Gmail
- Take survey later
- Continue
- Create new project
- And we can build our app with help of options provided in left side.

III. PROJECT DESIGN METHODOLOGY

A. WIRELESS TRANSMISSION OF TEXT

In this project first and main thing is that converting voice signal to text that can be made possible by our own Android app i.e. as follows

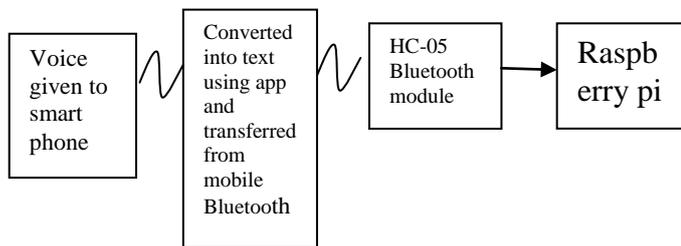


Fig .4. Wireless Transmission of Text

The Voice which was taken by ours can be converted into text by using an android app designed in MIT app inventor2 and it will transferred from our mobile Bluetooth to HC-05 module which is connected to Raspberry pi Board. The text transferred from Smartphone Bluetooth to Hc-05 Bluetooth serially in a half duplex manner.

B. CONNECTING FROM BLUETOOTH OF SMARTPHONE

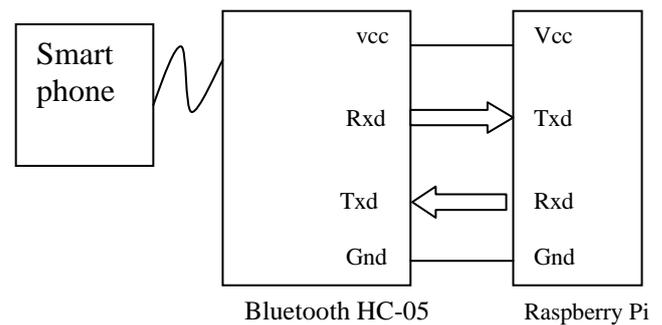
The voice can be converted as text by using my app and before converting into text we have pair with HC-05 and scan for devices and connect to that Bluetooth then the LED in the Bluetooth module blinks rapidly that is the sign that the module is ready and waiting for another device to pair up. Then the devices are paired, if we give voice that can be converted into text and transferred to HC-05 which is connected to Raspberry pi and searches the command whether it was written in the code or not, if result is positive then it works based on that particular command.

C. BLUETOOTH INTERFACING

The voice signals from can be converted into text and transferred to HC-05 Bluetooth from Bluetooth in the smart phone. The HC-05 module connected to Raspberry pi as follows

Bluetooth module(HC-05)	Raspberry pi GPIO
Vcc	Vcc(pin #2)
Rxd	Txd(pin #10)
Txd	Rxd(pin #8)
GND	GND(pin #6)

Pin connection from Bluetooth to Rasperry pi



D. RASPBERRY PI CONFIGIRATION

Here Raspberry Pi used for controlling purpose and it is almost like a small computer so to start with this we have to configure the board. The programming language here going to use is python. We are going to running a recent release of Raspbian on my Raspberry Pi. Commands for required installations as follows

1. sudo apt-get install python-dev for python installation
2. sudo apt-get install python-rpi-gpio for accessing GPIO pins of Raspberry pi

Both the commands can be done by in online manner we can done it in offline also. By default the Raspberry Pi is configured to write boot time messages to the serial port, and also to start a login console on it. Unfortunately, the default baud rate that the RPi uses for its serial port is 115200 bps, while the Bluetooth module comes preconfigured from factory to 9600 bps. It is easier to configure the RPi to use 9600 bps so we will try that first. There are two config files that need to be updated.

File /boot/cmdline.txt contains the kernel options that are used to boot the system. In my Raspbian based system this file contains the following options:

```

dwc otg.lpm enable=0console=ttyAMA0,115200
kgdboc=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline
rootwait
    
```

The interesting options are console and kgdboc, because these configure the serial port device

/dev/ttyAMA0 to 115200 bps.

We need to change these two configurations to 9600 bps.

After you make these changes the file should read

dwc otg.lpm enable=0 console=ttvAMA0,9600

kgdboc=ttvAMA0,9600 console=ttv1

root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline

rootwait

The second configuration file is /etc/inittab. Inside this file you have to locate the following line:

T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100

This tells the system to start a terminal on the serial port, and again it uses 115200 bps to configure the port. You have to change this line to use 9600 bps:

T0:23:respawn:/sbin/getty -L ttyAMA0 9600 vt100.

Remember that these are configuration files so they are not writable to the default pi user, to edit these files you have to use sudo. As far as text editors, Raspbian provides two, vi and pico. For example, to edit cmdline.txt with pico you would run the following command:

\$ sudo pico /boot/cmdline.txt

The system will ask for us password and after that we will be able to make changes to this file. I recommend that we save original copies of these config files in case we make a mistake.

E ANDROID APP MAKING

Here we used MIT app inventor2 to prepare own application based on our requirement .Steps to prepare an app as follows

1. Open the browser and open www.google.com
2. search for MIT app inventor2 and click on app inventor 2
3. Login with Gmail account
4. Take survey later
5. Create new project with project name

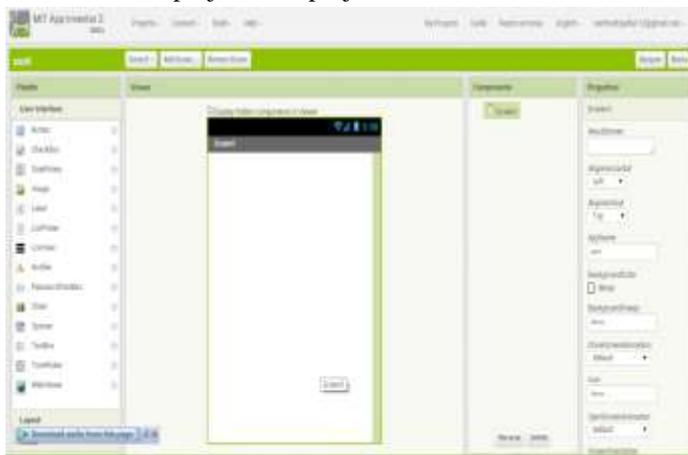


Fig .5 New project window

Left side we have buttons, list pickers, labels and many predefined blocks for the purpose of different applications. The application is for Bluetooth transmission so we need button, list picker and these blocks are available in user interface.

1. Button has ability to detect clicks. Many aspects of its appearance can be changed as well as whether it is clickable, can be changed, as well as whether it is clickable can be changed in the designer or in the blocks editor.

2. A label can display a piece of text, which is specified through the Text property. Other properties, all of which can be set in the Designer or blocks Editor, control the appearance and placement of text.

3. A button that, when clicked on, displays a list of texts for the user to choose among. The texts can be specified through the Designer or Blocks Editor by setting the Elements From String property to their string-separated concatenation (for example, *choice 1, choice 2, choice 3*) or by setting the Elements property to a List in the Blocks editor.

And for speech recognition we need speech recognizer which helps to convert the speech to text and Bluetooth client is also necessary for transmission and we have to pair up the Bluetooth module with smart phone Bluetooth. The text from smart phone transferred to Bluetooth module and for the transmission the properties are as follows.

Addresses and Names

The addresses and names of paired Bluetooth devices

Available

Whether Bluetooth is available on the device

Character Encoding

DelimiterByte

Enabled

Whether Bluetooth is enabled

HighByteFirst

IsConnected

Secure whether to invoke SSP (Simple Secure Pairing), which is supported on devices with Bluetooth v2.1 or higher. When working with embedded Bluetooth devices, this property may need to be set to False. For Android 2.0-2.2, this property setting will be ignored. We need a clock i.e. Non-visible component that provides the instant in time using the internal clock on the phone. It can fire a timer at regularly set intervals and perform time calculations, manipulations, and conversions. Methods to format the date and time are also available. All these elements are used in Designer section and the final designer for our application is as follows

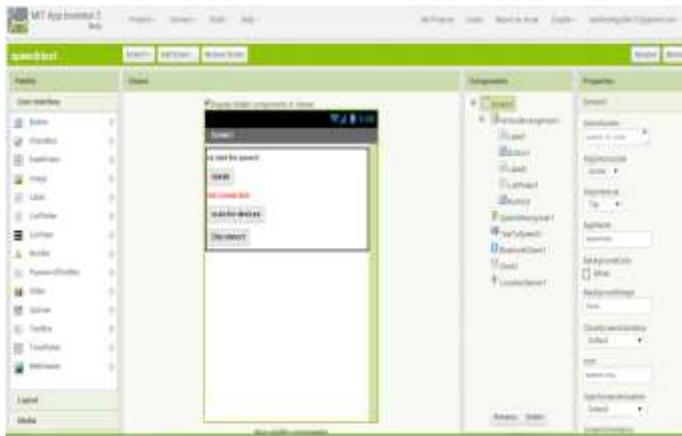


Fig .6 APP Designer

To perform the operations with that button, label, list pickers, Bluetooth client we need to write the code and that will be in blocks section .In this blocks section we have different inbuilt components with the help of that blocks we have to prepare the code.

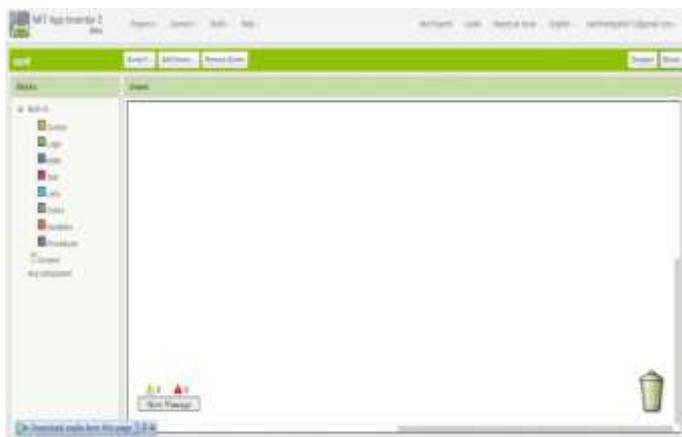


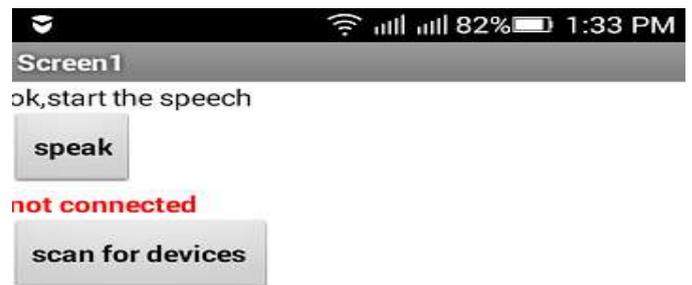
Fig .7 Blocks window

With the help of blocks we can prepare code for corresponding application and for this paper the final code for App is



Fig .8 parts of App code

After the creation of app we have to save the project. And we have a build option at the top to create .apk file to install the app into our phone and the screen shot of this app is given below



What we mentioned in the above pages is all about what are elements we used and how it works and the final block diagram is

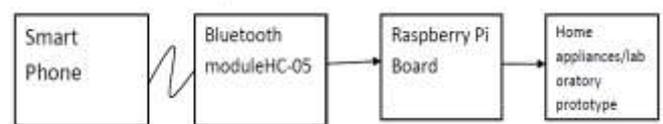


Fig .9 Block Diagram

The main program for this system written by using python.

IV. PROPOSED SYSTEM

Finally our proposed is consists of Smartphone with app, Bluetooth module HC-05, Raspberry pi B board and target board.



Fig .10 Prototype of proposed system

This is the basic arrangement to implement our idea it consists a smart phone acts as voice input and that can be send to Bluetooth module which was connected to Raspberry pi and

searches the text what we written in the code section of Raspberry pi .If the text matches then the corresponding section will executes.

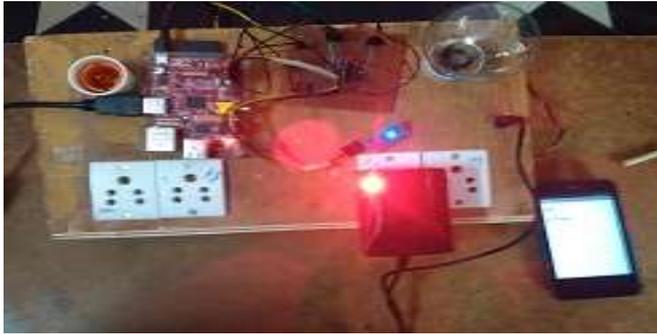


Fig .11 small Arrangements of home appliances

In the power section a simple 9-12V battery is connected with the transmitting end as well as receiving end or we may use 12V adapter to drive AC elements. Since the microcontroller which is the central processing unit works on regulated +5V power supply so this can be achieved by connecting a +5 volt regulator at the output of battery. This is shown in figure 12.

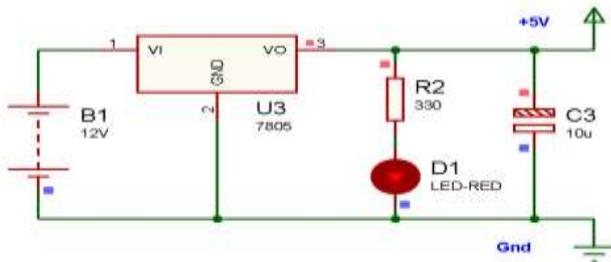
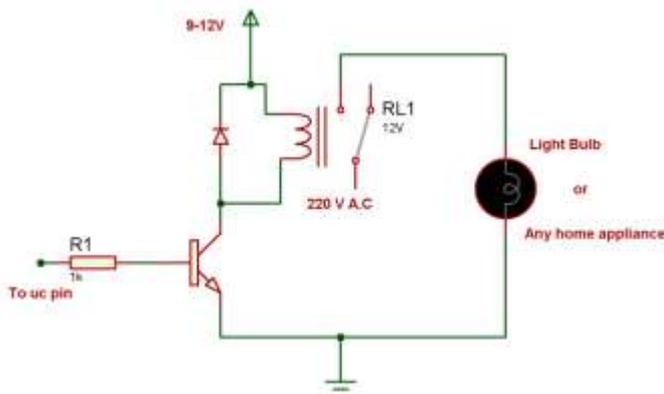


Fig .12: Power supply unit

A single relay switch connection to microcontroller is shown below. Any home appliance operating under A.C voltage supply could be connected directly with this relay interface circuit. A buzzer can also be connected by applying +12V at the common terminal of relay. Remaining relays of the interface circuit board could also be configured with the microcontroller in the similar fashion



Similarly different sensors could be interface with the microcontroller by using analog to digital convertor or comparator circuit. So any sensors can be connected with the remote station according to the system.

V. CONCLUSION AND FUTURE SCOPE

Implementation of portable voice-based authentication system using Bluetooth and our own Android app is very in expensive and it is very useful project for the adults and physically disabled persons, With the Voice Recognition along with Bluetooth network we can eliminate the complication of wiring in case of wired automation and also it prevent to get up and down again and again to on/off appliances. It very in expensive it can be help in improves small laboratory works also.

REFERENCES

1. L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.
2. www.imagesco.com/articles/hm2007/SpeechRecognitionTutorial01.html
3. V. Ramya, B. Palaniappan, Bobby George, Embedded System for Automatic Irrigation of Cardamom Field using Xbee-PRO Technology, *International Journal of Computer Applications* (0975 – 8887) Volume 53– No.14, September 2012..
4. F. L. Alegre, "Application of ANN and HMM to Automatic Speaker Verification", *IEEE LATIN AMERICA TRANSACTIONS*, Vol. 5, No. 5, pp. 329-337, Sept. 2007
5. <http://appinventor.mit.edu/explore/>