A PROJECT

ON

VOICE CONTROLLED ROBOT USING AN ANDROID DEVICE

BY

1. SIDDHARTH GUPTA
2. SIDDHARTH BOOBNA

Under the guidance of

Internal Guide

Prof. Nilesh M. Patil

Department of Information Technology

MCT

MANJARA CHARITABLE TRUST

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53

University of Mumbai

April – 2013

MCT

MANJARA CHARITABLE TRUST

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

Juhu-Versova Link Road, Versova, Andheri(West) Mumbai -53

CERTIFICATE

Department of Information Technology

This is to certify that

1. SIDDHARTH GUPTA
2. SIDDHARTH BOOBNA

Have satisfactorily completed this project entitled

VOICE CONTROLLED ROBOT USING AN ANDROID DEVICE

Towards the fulfilment of the

BACHELOR OF ENGINEERING

IN

(INFORMATION TECHNOLOGY)

as laid by University of Mumbai.

Guide	H.O.D.
Prof. Nilesh M. Patil	**Prof. D. M. Dalgade**

Principal

Dr.Udhav Bhosle

Internal Examiner	External Examiner

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to Dr. U. V. Bhosle, Principal and Prof. D. M. Dalgade, H.O.D of Information Technology Department of RGIT for providing us an opportunity to complete our dissertation on “Voice Controlled Robot Using an Android Device".

This project bears on imprint of many people. We sincerely thank our project guide Prof. Nilesh M. Patil for his guidance and encouragement in successful completion of our project work.

We would also like to thank our staff members for their help in carrying out this project work.

Finally, we would like to thank our colleagues and friends who helped us in completing the project successfully.

1. SIDDHARTH GUPTA
2. SIDDHARTH BOOBNA

ABSTRACT

The project uses an Android device which has such widespread popularity to remotely control a robot. It consists of using an android application that recognizes the speech, and transmits it wirelessly over to a microcontroller based robot. The robot is equipped with a wireless camera to provide live feed of its surroundings. Its application is in surveillance, places where humans find difficult to reach. E.g. in a small pipeline, in fire-situations, and in highly toxic areas. It can be used to bring and place small objects. It also has potential in areas like home automation system, human identification system over voice and as a personal assistant to help the physically handicapped, the elderly or the children.

The android application is based on Android's 2.3.x version of "Gingerbread", programmed in Java language using Android SDK on Eclipse. The application takes an input as voice via the device's microphone and converts it into text format for the robot. This text is internally converted to a command from the database. These text commands are then sent using the device's native Bluetooth capability to the Bluetooth receiver installed on the robot. Bluetooth is a proprietary open wireless technology standard for exchanging data over short distances from fixed and mobile devices. Its data transfer rate is 3 Mbps with a frequency of 2.4 GHz and a range of 10 metres. The pre-programmed autonomous robot uses Atmel's ATmega32 8-bit Microcontroller with 32kb flash memory working at 16MIPS per 16 clock cycles. The microcontroller is programmed in 'C' language and the various functions are hard-coded on it with the appropriate signals. A stepper motor is used for the mobility of the robot. On receiving the signal, it recognizes the command and acts on it accordingly.

TABLE OF CONTENTS

LIST OF FIGURES

ACRONYMS

A	
ADT	Android Development Toolkit
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AVM	Android Virtual Manager
C	
CE	Conformité Européenne
E	
EDR	Enhanced Data Rate
EEPROM	Electrically Erasable Programmable Read-Only Memory
F	
FCC	*Federal Communications Commission*
G	
GCC	*GNU Compiler Collection*
GUI	*Graphical User Interface*
H	
HID	Human Interface Device
M	
MAC	Media Access Control
MCU	Multipoint Control Unit
MIPS	Million Instructions Per Second
N	
NCCI	NovaComm Control Interface
O	
OEM	*Original Equipment Manufacturer*
OPP	*Object Push Profile*
OS	Operating System

P	
PCB	Printed Circuit Board
PCM	*Pulse-Code Modulation*
PPI	Pixels Per Inch
R	
RAM	Random Access Memory
RFCOMM	Radio Frequency Communication
RFID	Radio Frequency Identification
RISC	*Reduced Instruction Set computing*
RoHS	Restriction of Hazardous Substances
S	
SDK	Software Development Kit
SMT	*Surface-Mount Technology*
SPP	Serial Port Profile
U	
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
UUID	Universally Unique Identifier
V	
VCR	Voice Controlled Robot
VR	Voice Recognition
W	
WBS	Work Breakdown Structure

INTRODUCTION & MOTIVATION

Speech-recognition software is nothing new. But put it on a Smartphone, and it comes to life. All of the frustrations of trying to control your robot by voice--fiddling for a microphone, repeating yourself again and again, running behind the robot--are eliminated when you use the same technology on your mobile phone. And it's becoming more popular all the time, thanks in large measure to the improved speech recognition capabilities of today's mobile phone platforms, such as Google's Android.

Visually impaired people mainly rely on voice commands, voice menu or voice feedbacks for any control operation. Here we integrate voice features into an android based device. Robot control is achieved by the use of voice and mobile in this project, voice-control is used, if in case the user is within the range of the robot where it has to be controlled. Nowadays robotics system is being applied in Medical Surgery, Factory automation, military, security systems, space research and many more. Now many vendors are interested in engaging robots in human’s daily life as service robot.

Voice Controlled Robot (VCR) is a mobile robot whose motions can be controlled by the user by giving specific voice commands.

In this unit we capture the speech signals coming from the microphone attached to the mobile device. The software running on the device processes the signals to recognize the voice commands ‘Forward’, ‘Stop’, ‘Left’, ’Right’ and ‘Back’. These commands are sent over wirelessly to the microcontroller based robot which will perform the tasks accordingly.

PROBLEM STATEMENT

The traditional way of performing day to day tasks is by exerting yourself. It can especially be tedious for the aged or physically handicapped people.

Though many popular speech recognition modules are available in the market, very few of them are compatible when we need to use them in a system for the physically challenged and also for the visually impaired people.

Normally the voice controlled robot which had voice recognition modules integrated on to the robot itself, had a few drawbacks. Firstly, we had to follow the robot to give the speech command to the VR module. So there was always a limitation on how far the person can be from the robot to command it. Secondly, the module may not be of the best quality and thus will have a poor noise cancellation rate. Thirdly, the module may not always function according to our needs.

In the military, during surveillance, there is a problem of surveying the opponent's territory with the help of just a camera or binoculars. They do not get a clear picture of the area since they are viewing it from a distance.

REQUIREMENT ANALYSIS

3.1 Functional Requirements

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. In this project, our primary requirement is to enable the user to command the robot using speech. It can be enlisted as follows:

Interface requirements:

Bluetooth: The Bluetooth module must be made discoverable prior to discovering for devices. A class 2 Bluetooth module is used in this project.

Input/Output requirements:

Activity 1: It displays all the paired devices and the newly discovered devices in a list. User has to select a device to connect to.

Activity 2: Opens only when user pairs with a Bluetooth device. It has five input buttons “Forward”, “Left”, “Right”, “Back”, “Press to Speak”. All the buttons perform different respective functions on clicking them.

The results of the speech-to-text conversion are shown in a list and the most probable result in a text box. This command is sent to the robot.

Regulatory/Compliance Requirements :

Bluetooth Pairing: The Bluetooth module is only capable of pairing with devices which do not require a passkey for pairing. Only Android devices running on Gingerbread 2.3.3 or previous versions are capable of connecting to the Bluetooth module.

Security Requirements :

The Bluetooth connection established is secure and will be able to prevent man-in-the-middle attacks or any unauthorised usage.

3.2 Non-Functional Requirements

Non-functional requirements impose constraints on the design or implementation (such as performance requirements, quality standards or design constraints). Users have implicit expectations about how well the software - hardware assembly will work. These characteristics include how easy the product is to use, how quickly it executes, how reliable it is, and how well it behaves when unexpected conditions arise.

1. Hardware Requirements:

- Android device running 2.3.x or higher.
- Minimum 128 MB RAM required.
- Minimum 256 MB memory space required.
- Android phone with microphone and Bluetooth facility.
- Processor speed 600 MHz or higher.
- Resolution 320 x 480 pixels, 200 ppi or higher.
- Android device with a 2G or a 3G data connection.

2. Software Requirements:

- OS: Android Gingerbread 2.3.3
- Android Programming: Android SDK
- Eclipse Juno or higher
- Android ADT
- Emulator: Android Virtual Manager(AVM)
- GUI: Java

- Programming language: Java

3. Performance requirements:

- Response time of the speech to text conversion taking place over the internet is as low as 1 second when using a 3G data connection and 2 seconds in case of 2G data connection.
- Performance may degrade in places where a stable internet connection is not possible.

4. Accuracy and Precision:

- The accuracy of the correct result was 65%. It's been enhanced to 90% by the detection of homophones which even on incorrect results enables the device to send the right command.
- Accuracy may degrade in places where lot of noise is present.

5. Physical environment:

- The environment must have as low noise levels as possible and device should be able to obtain a stable internet connection from the location.

PROJECT DESIGN

4.1 Use Case Diagram

Use cases represent typical sets of scenarios that help to structure, relate and understand the essential requirements. A Use Case Diagram that help system analyst to discover the requirements of the target from the user's perspective. Use Case Diagram can be used to describe the functionality of a system in horizontal way. That is, rather than merely representing the details of individual features of our system, it can be used to show all the available functionality of the system. The use case diagram for the project is depicted in figure 4.1.

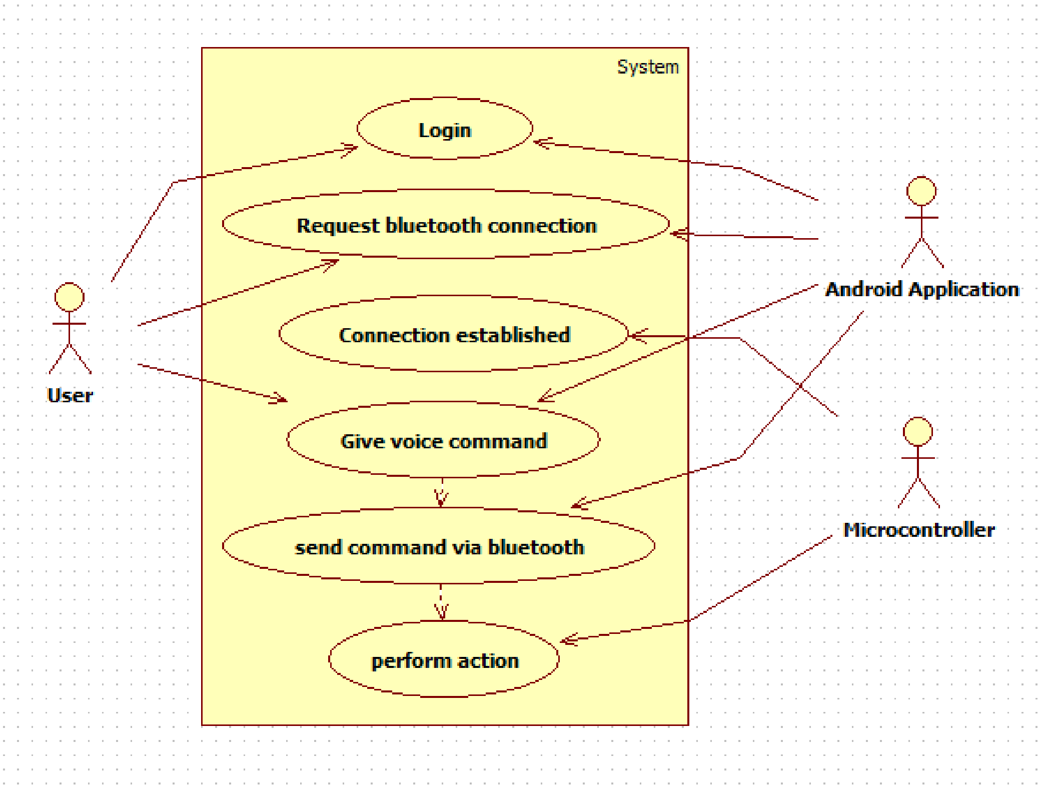


Figure 4.1: Use Case Diagram

4.2 Class Diagram

The first step in analyzing the requirements is to construct a class diagram. The class diagram shows the static data structure of the real world system and organizes it into workable pieces. The object model describes real world object classes and their relationships to each other. The class diagram for the project is depicted in figure 4.2.

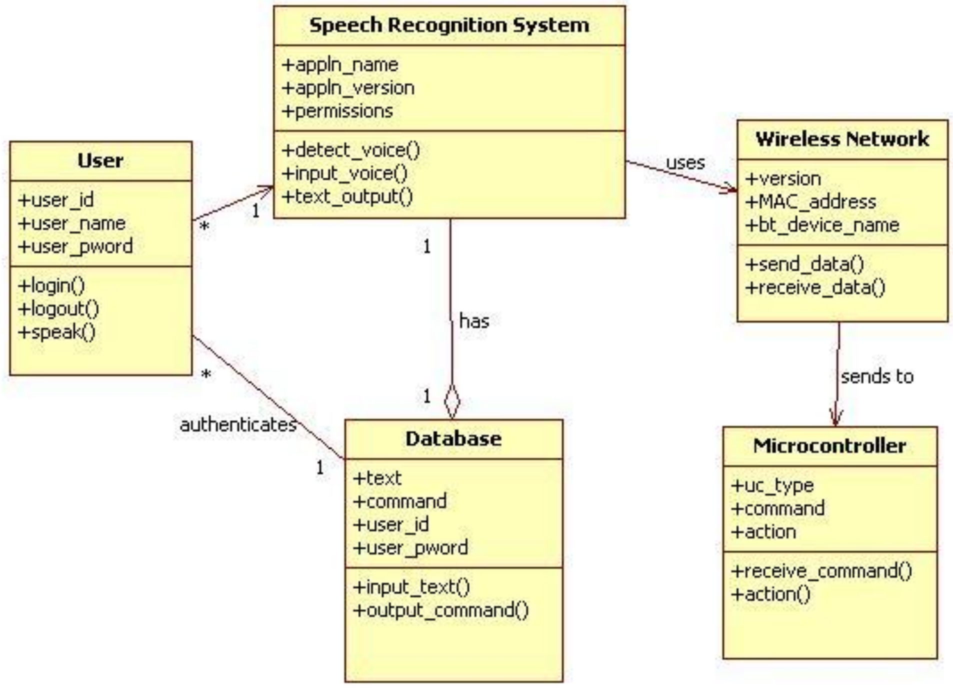


Figure 4.2: Class Diagram

4.3 Sequence Diagram

When an object passes a message to another object, the receiving object might in turn sends a message to another object, which might in turn sends message to yet another object. This stream of messages forms a sequence. The messages are ordered in sequence by time. Sequence diagrams model these sequences. The sequence diagram for the project is depicted in figure 4.3.

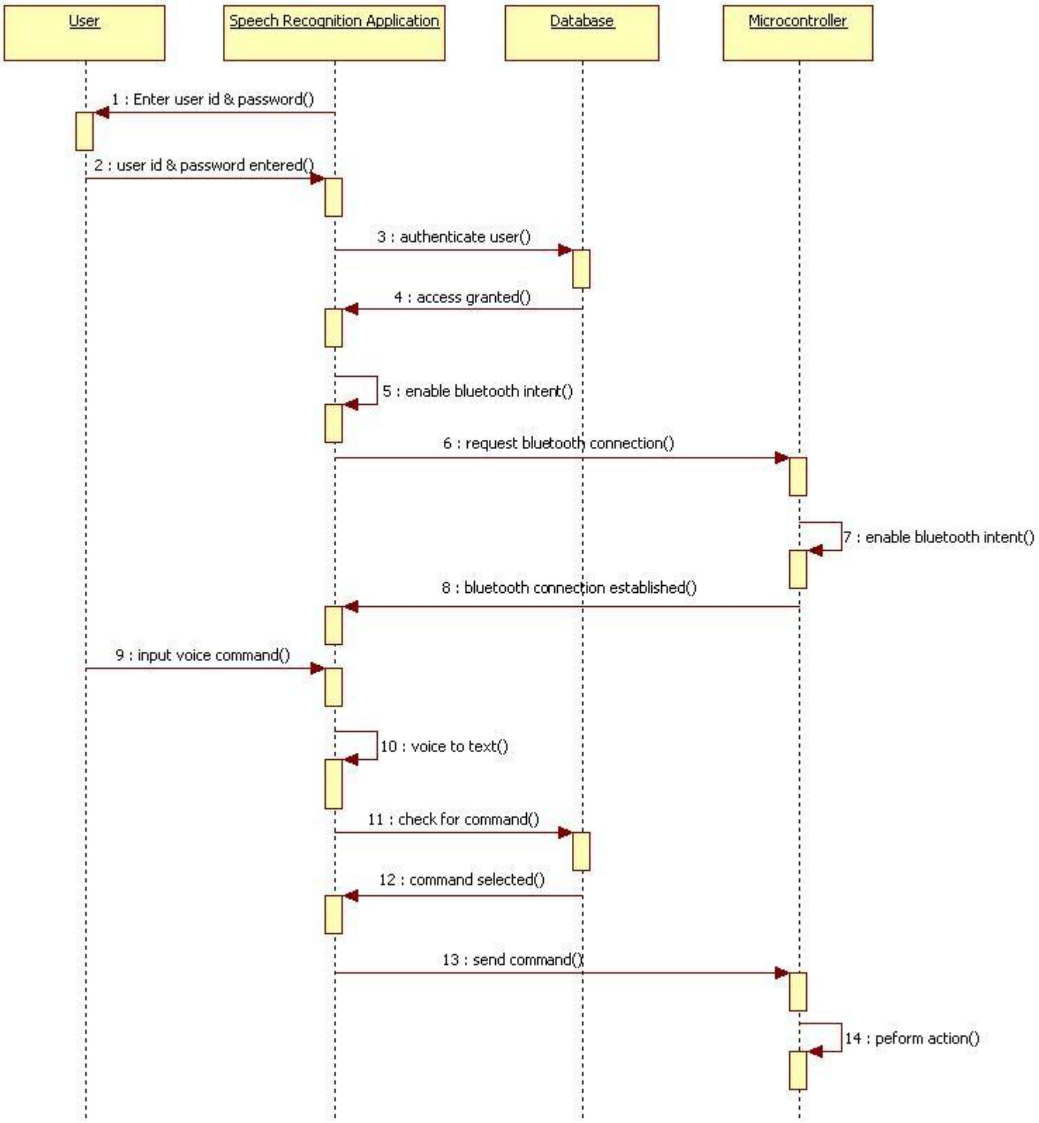

User
Speech Recognition Application
Database
Microcontroller
1 : Enter user id & password()
2 : user id & password entered()
3 : authenticate user()
4 : access granted()
5 : enable bluetooth intent()
6 : request bluetooth connection()
7 : enable bluetooth intent()
8 : bluetooth connection established()
9 : input voice command()
10 : voice to text()
11 : check for command()
12 : command selected()
13 : send command()
14 : peform action()

Figure 4.3: Sequence Diagram

4.4 State Diagram

Using a state machine, we can model the behavior of an individual object. A state machine is a behavior which specifies the sequences of states an object goes through during its lifetime in response to events, together with its response to those events. The state diagram for the project is depicted in figure 4.4.

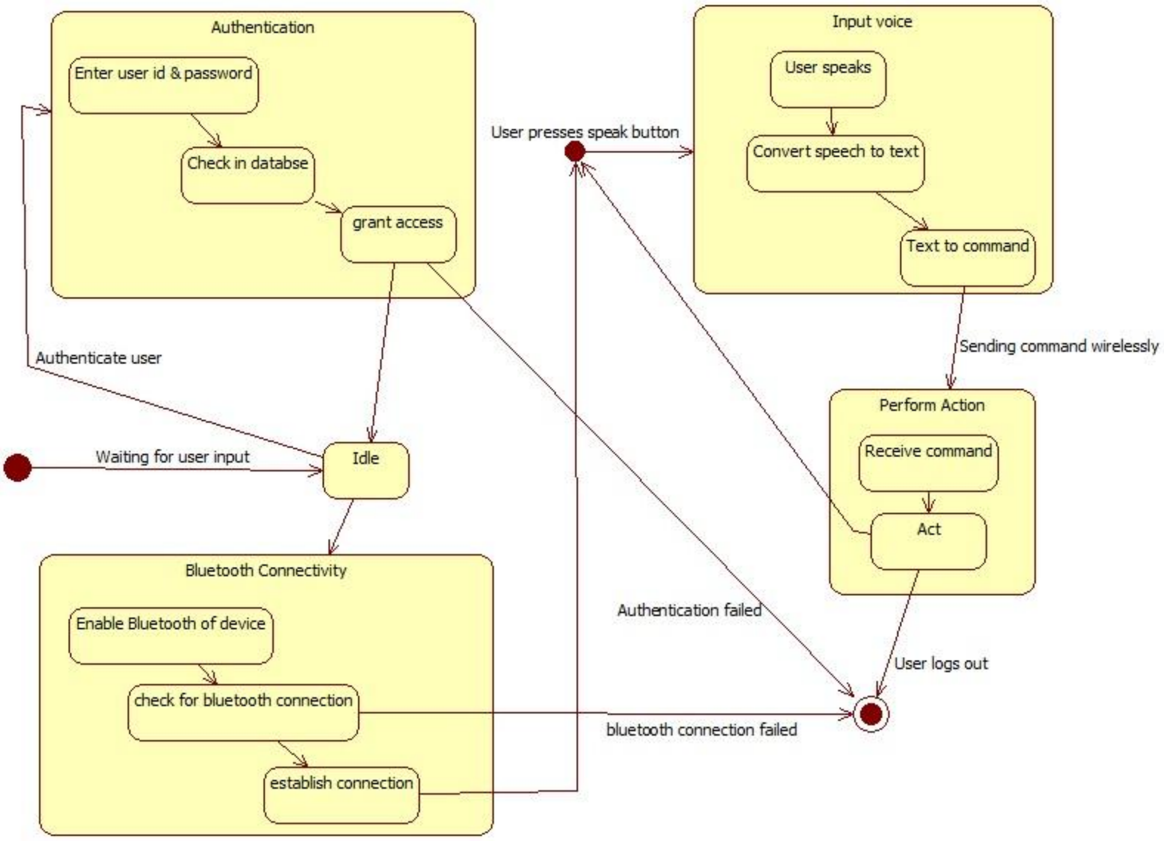


Figure 4.4: State Diagram

4.5 Collaboration Diagram

A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Graphically, a sequence diagram is a collection of vertices and arcs. A Collaboration diagram is very similar to a sequence diagram in the purpose it achieves; It describes a specific scenario. Numbered arrows show the movement of messages during the course of a scenario. Collaboration diagram express similar information as in sequence diagram, but shown in different way. The collaboration diagram for the project is depicted in figure 4.5.

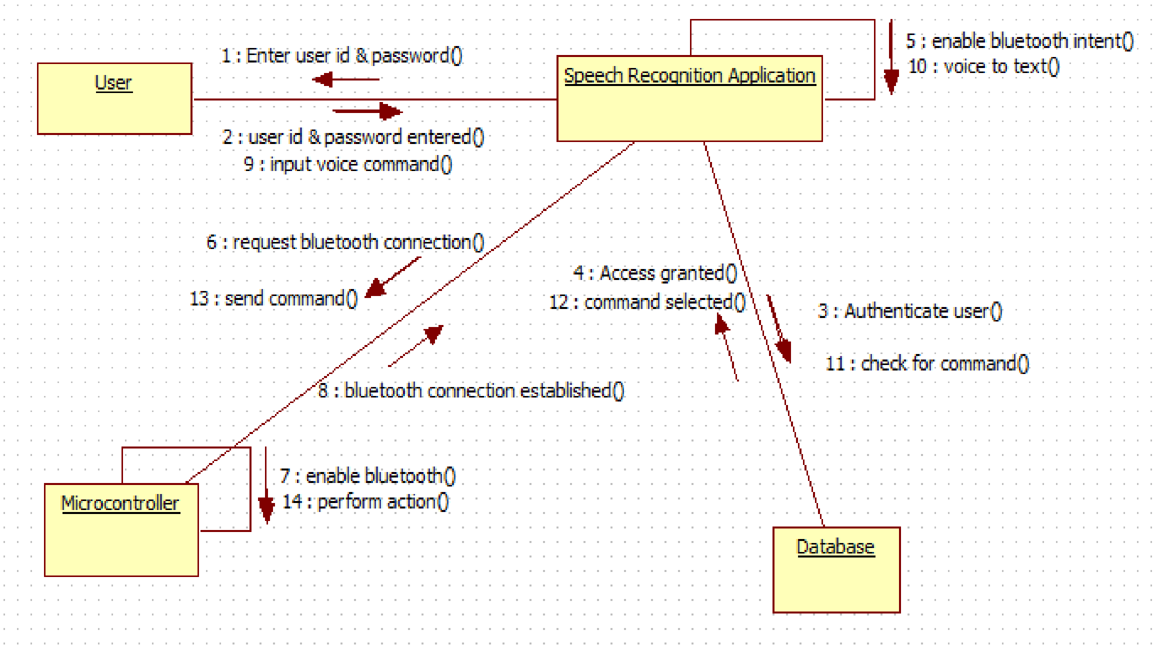

User
Speech Recognition Application
Microcontroller
Database
1 : Enter user id & password()
2 : user id & password entered()
9 : input voice command()
5 : enable bluetooth intent()
10 : voice to text()
6 : request bluetooth connection()
13 : send command()
4 : Access granted()
12 : command selected()
3 : Authenticate user()
11 : check for command()
8 : bluetooth connection established()
7 : enable bluetooth()
14 : perform action()

Figure 4.5: Collaboration Diagram

4.6 Component Diagram

Component diagrams are one of the two kinds of diagrams found in modeling the physical of object oriented systems. A component diagram shows the organization and inter-relationships among a set of components. Graphically, a component diagram is a collection of vertices and arcs. The different high level reusable parts of a system are represented in a component diagram. Collecting various executables, libraries, files, tables (physical things), we build the component diagram. The components can be reusable components or can be developed from scratch. The component diagram for the project is depicted in figure 4.6.

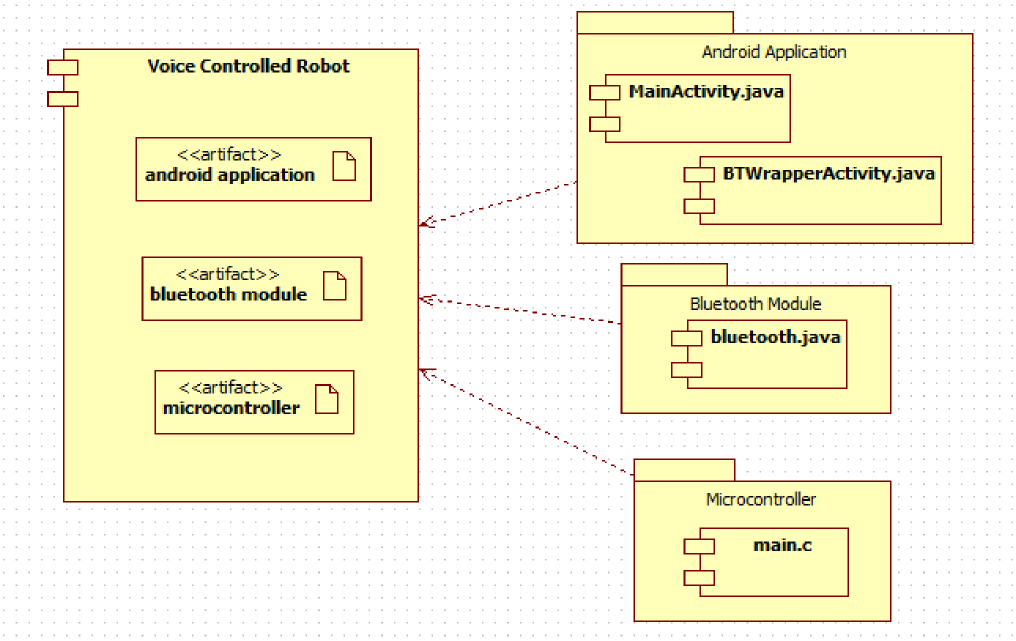


Figure 4.6: Component Diagram

4.7 Deployment Diagram

The deployment diagram provides a different perspective of the application. The deployment diagram captures the configuration of the run time elements of the application.

We use deployment diagrams to model the static and dynamic deployment view of a system. For the most part, this involves modeling the topology of the hardware on which our system executes. Deployment diagrams are essentially class diagrams that focus on a system's nodes. The deployment diagram for the project is depicted in figure 4.7.

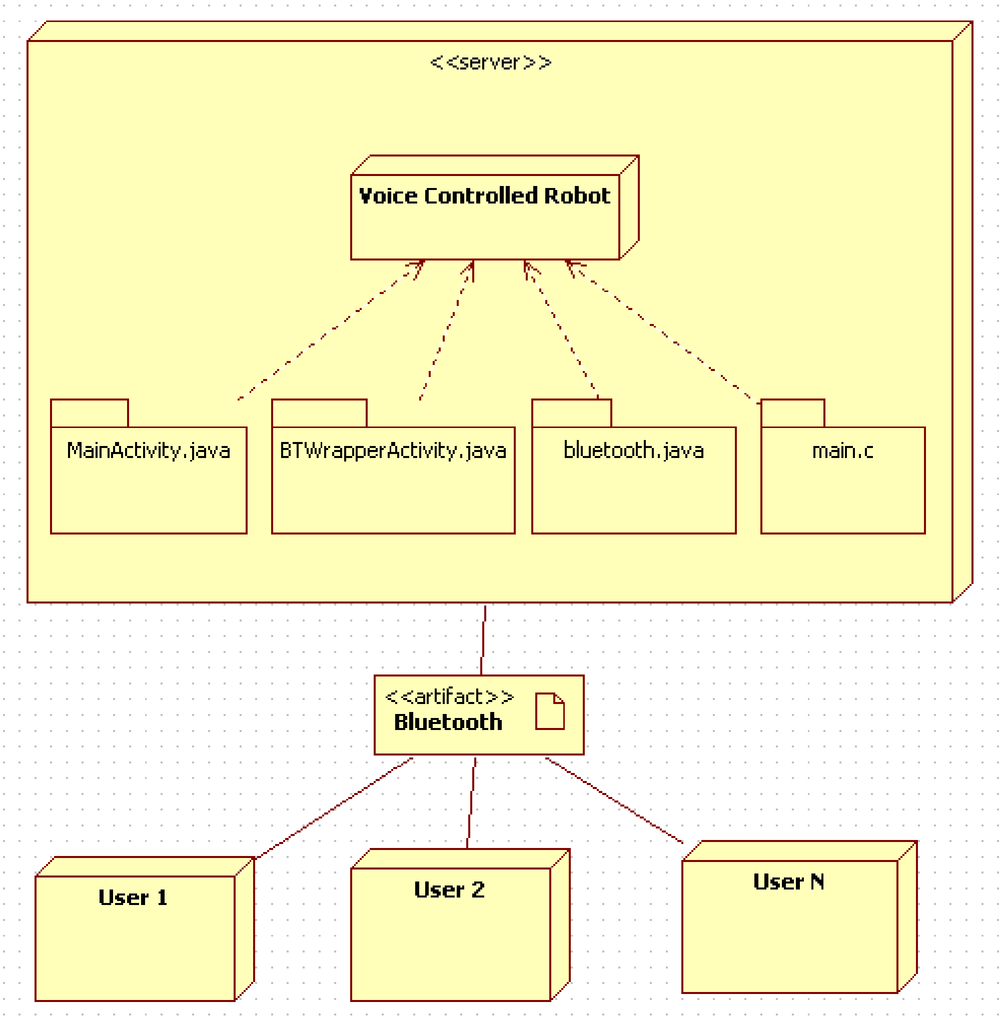


Figure 4.7: Deployment Diagram

IMPLEMENTATION DETAILS

The project is divided into 3 stages. The 1st stage consists of an android based speech recognition application. This application converts user’s voice input to text. The 2nd stage deals with sending the command over to the microcontroller using Bluetooth. Then the final stage comprises of the microcontroller based robot which receives the signal and acts according to the command.

5.1 Android Application

5.1.1 Speech Recognition

Android is an open platform, so applications can potentially make use of any speech recognition service on the device that's registered to receive a RecognizerIntent. Google's Voice Search application, which is pre-installed on many Android devices, responds to a RecognizerIntent by displaying the "Speak now" dialog and streaming audio to Google's servers -- the same servers used when a user taps the microphone button on the search widget or the voice-enabled keyboard.

You can make sure your users have the best experience possible by requesting the appropriate language model: free_form for dictation, or web_search for shorter, search-like phrases. We developed the "free form" model to improve dictation accuracy for the voice keyboard. We then match this text with a command from the database formed on the android phone. For example, we have command “1” defined for text “forward”.

5.1.2 Bluetooth

Now we use the Bluetooth of the phone to send the command to the Bluetooth module on the microcontroller. The Android platform includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The application framework provides access to the Bluetooth functionality through the Android Bluetooth APIs. These APIs let applications wirelessly connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features.

This setup is accomplished in two steps, using the **BluetoothAdapter**.

- Get the **BluetoothAdapter**

The **BluetoothAdapter** is required for any and all Bluetooth activity. To get the **BluetoothAdapter**, call the static **getDefaultAdapter()**method. This returns a **BluetoothAdapter** that represents the device's own Bluetooth adapter (the Bluetooth radio). There's one Bluetooth adapter for the entire system, and your application can interact with it using this object.

- Enable Bluetooth

Next, you need to ensure that Bluetooth is enabled. Call **isEnabled()** to check whether Bluetooth is currently enable. If this method returns false, then Bluetooth is disabled. To request that Bluetooth be enabled, call **startActivityForResult()** with the **ACTION_REQUEST_ENABLE** action Intent. This will issue a request to enable Bluetooth through the system settings (without stopping your application).

Using the BluetoothAdapter, you can find remote Bluetooth devices either through device discovery or by querying the list of paired (bonded) devices.

Device discovery is a scanning procedure that searches the local area for Bluetooth enabled devices and then requesting some information about each one. If a device is discoverable, it will respond to the discovery request by sharing some information, such as the device name, class, and its unique MAC address. Using this information, the device performing discovery can then choose to initiate a connection to the discovered device.

When a device is paired, the basic information about that device (such as the device name, class, and MAC address) is saved and can be read using the Bluetooth APIs. Using the known MAC address for a remote device, a connection can be initiated with it at any time without performing discovery (assuming the device is within range).

To start discovering devices, simply call startDiscovery().

The application registers a BroadcastReceiver for the ACTION_FOUND Intent in order to receive information about each device discovered. For each device, the system will broadcast the ACTION_FOUND Intent. This Intent carries the extra fields EXTRA_DEVICE and EXTRA_CLASS, containing a BluetoothDevice and a BluetoothClass, respectively.

In order to initiate a connection with a remote device (a device holding an open server socket), we must first obtain a BluetoothDevice object that represents the Bluetooth Module.

We then use the BluetoothDevice to acquire a BluetoothSocket and initiate the connection using createRfcommSocketToServiceRecord(UUID).

Now, we get the InputStream and OutputStream that handle transmissions through the socket, via getInputStream() and getOutputStream(), respectively.

We can now write data to the streams with write(byte[]).

5.2 Bluetooth for Atmega Microcontroller

It is possible to connect a Bluetooth module to an Atmega microcontroller via the UART serial interface and even further, discover it using a Bluetooth enabled computer or smartphone.

The 9600bps serial link established can then be used to exchange data.

We are directly interfacing the Android with the Atmega microcontroller via the serial Bluetooth link. So we can use this technology for other purposes than robots, since the Atmega has a large range of possible uses: control the light in your home using the Android phone, read various sensors and gather the data on the Android, and more.

We need to setup the Bluetooth module using its AT commands and make it discoverable.

The baud rate is set at 9600 with the Bluetooth Module which listens for commands coming via the Bluetooth serial socket.

5.3 Microcontroller based Robot

Coming on to the last phase of our project, we program the microcontroller Atmega32 in ‘C’ language. We directly define the actions that the robot is supposed to perform according to the command signal it receives over Bluetooth. The data received by Bluetooth is sent to the Microcontroller via UART interface. In the program, we initialize UART by using uart_init() method. The data received by UART is listened using uart_read(). The program, thus, checks for the data on the UART port and performs actions according to the command. We send the signal on the PORT that is connected to the motor to run it.

TECHNOLOGIES USED

6.1 Android 2.3.x

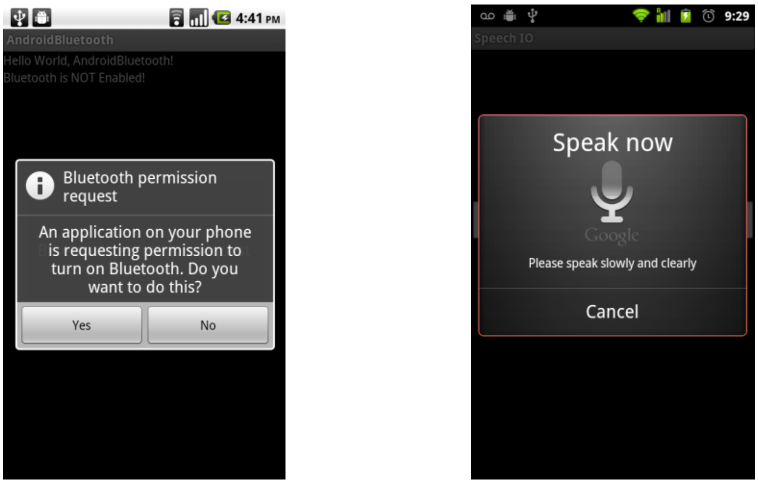
The Android Developer Tools (ADT) plugin for Eclipse provides a professional-grade development environment for building Android apps. It's a full Java IDE with advanced features to help you build, test, debug, and package your Android apps. It is free, open-source, and runs on most major OS platforms. The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS.

We have used the following Bluetooth and Speech Recognition APIs for the project that is available under Android SDK.

import android.speech.RecognizerIntent;

import android.bluetooth.*;



AndroidBluetooth

Hello World, AndroidBluetooth!

Bluetooth is NOT Enabled!

Bluetooth permission request

An application on your phone is requesting permission to turn on Bluetooth. Do you want to do this?

Speech IO

Speak now

Please speak slowly and clearly

Figure 6.1: Snapshots

6.2 AUBTM-20 Bluetooth Module

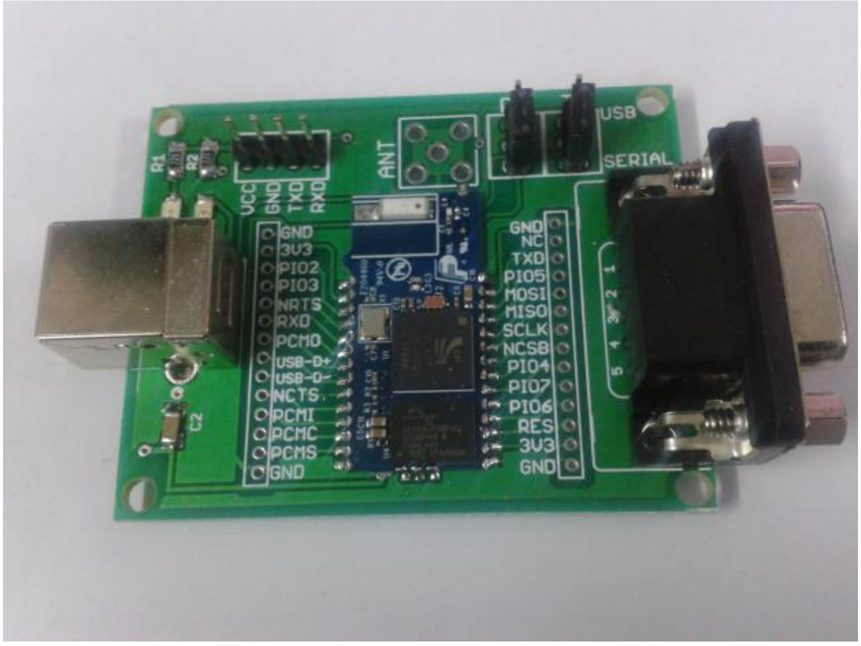


Figure 6.2: AUBTM-20 Module with MDCS41A Bluetooth chip

Bluetooth module is a small form factor and highly economic Bluetooth radio module (class 2) that allows OEM to add wireless capability to their products. The module supports multiple interfaces that make it simple to design into fully certified embedded Bluetooth solutions.

With module's BC#™ programming interfaces, designers can easily customize their applications to support different Bluetooth profiles, such as SPP, OPP, HID, and etc. Class 2 module supports 3Mbps data rate Transmission for distances up to 10 meters with its integrated chip antenna.

The module is an appropriate product for designers who want to add wireless capability to their products.

Features:

- Bluetooth v2.1+EDR
- Support different Bluetooth profiles (SPP， OPP， HID)
- UART and USB programming and data interfaces
- PCM digital audio interfaces

- 8MB on board flash
- SMT pads for easy and reliable PCB mounting
- Built-in ceramic antenna or select the external antenna on board
- BQB/FCC/CE Certified
- RoHS compliant

Applications

- Cable replacement
- Bar code and RFID scanners
- Measurement and monitoring systems
- Industrial sensors and controls
- Medical devices
- Industrial PCs and laptops

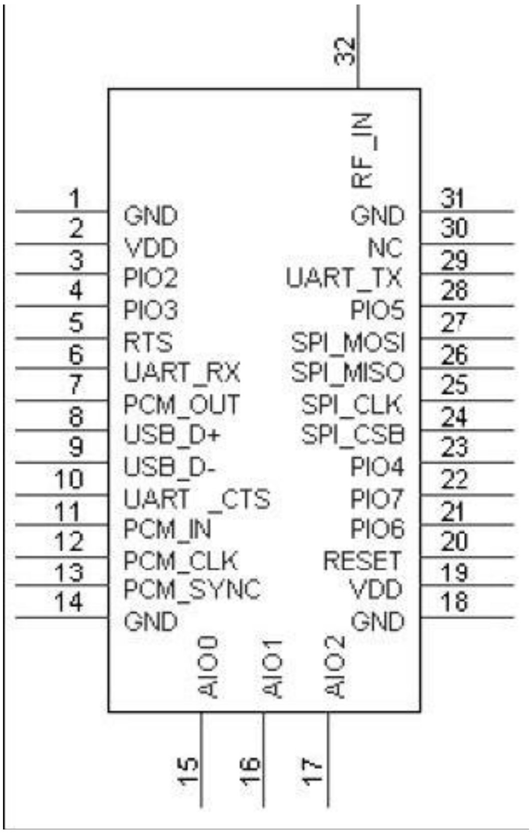


Figure 6.3: Pin Diagram of MDCS41A Bluetooth chip

6.2.1 UART

This is a standard Universal Asynchronous Receiver Transmitter (UART) interface for communicating with other serial devices. Four signals UART_TX, UART_RX, UART_CTS, and UART_RTS are used to implement the UART function, UART_CTS, UART_RTS can be used to implement hardware flow control. PIO2 and PIO3 can be configured as DTR and RTS.

6.2.2 NovaComm Control Interface

The NovaComm Control Interface (NCCI) is a set of ASCII commands and indicators with which the user can control the NovaComm's Bluetooth module via UART interface by a host (PC, MCU, etc.).

The commands are used to control the Bluetooth module sent by host. The indicators are output from the Bluetooth module to the host to indicate the status of the module.

The general syntax of NCCI command is shown as below:

BC:CMD[=Para1][,Para2][,RawData][,…]<CR><LF>

The general syntax of NCCI indicator is shown as below:

IDC[=Para1][,Para2][,RawData][,…]<CR><LF>

6.3 Atmega32 Microcontroller

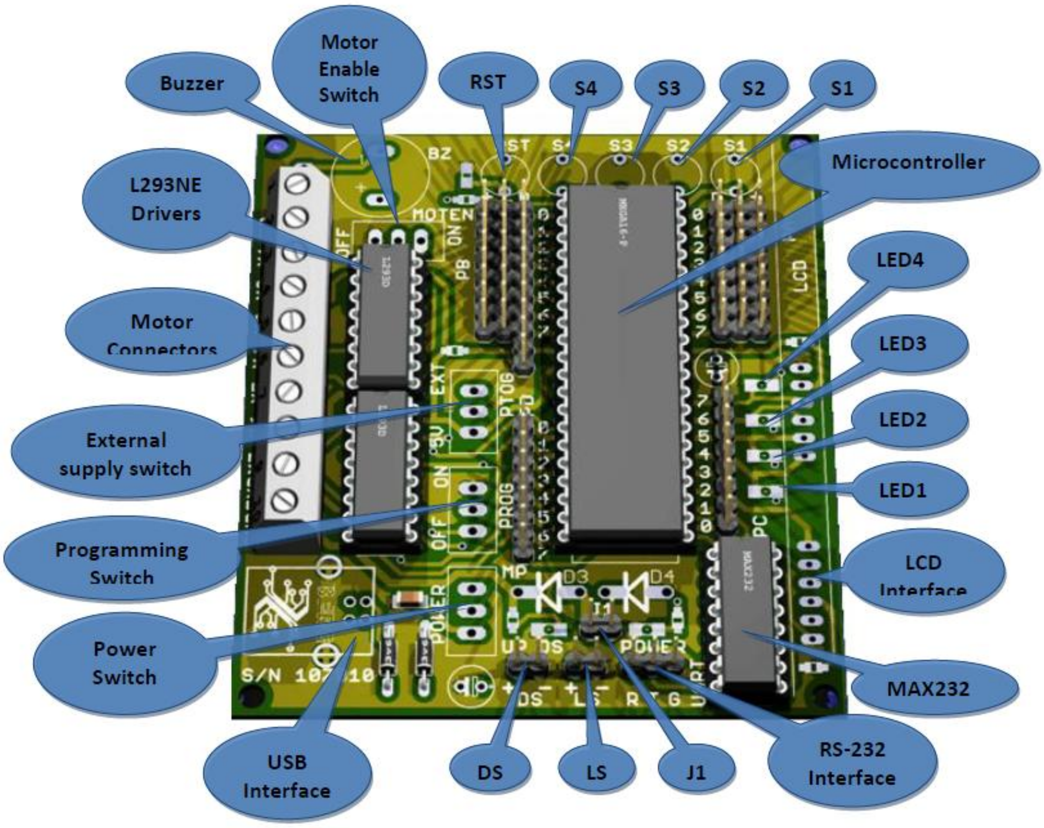


Figure 6.4: Atmega32 Motherboard

The Atmega32 microcontroller has the following features:-

- High performance, low power AVR 8-bit microcontroller.
- Upto 16 MIPS throughput at 16 MHz
- 32K bytes of In-System Self programmable flash endurance.
- 1024 bytes EEPROM.
- 32 programmable I/O lines.
- Operating voltage of 4.5 -5.5 V.

- Speed of 16 MHz.

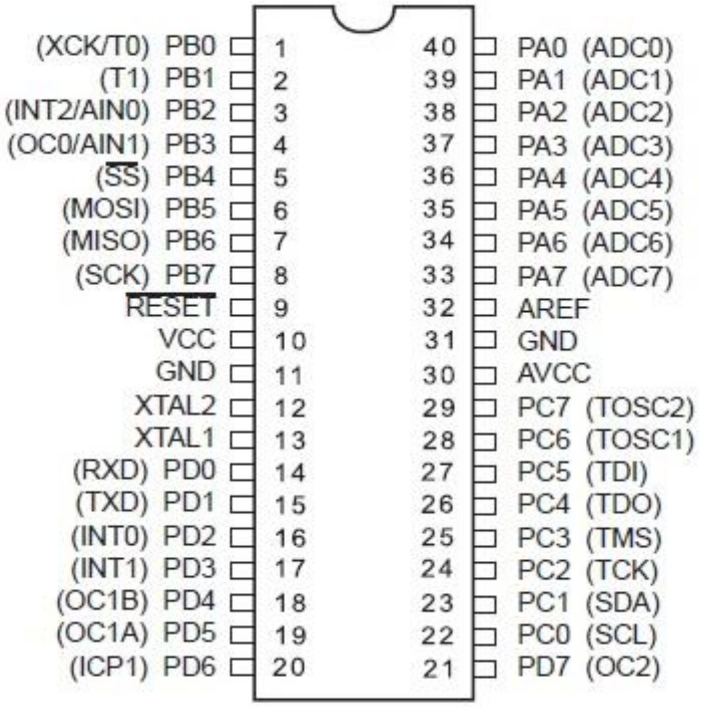


Figure 6.5: Pin Diagram of Atmega32

6.3.1 Configuring the pin

Each port pin consists of three register bits: DDxn, PORTxn and PINxn. The DDxn bits are accessed at the DDRx I/O address, the PORTxn at the PORTx I/O address and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of the pin. If DDxn is written logic one, Pxn is configured as an output pin. Id DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

6.3.2 Programming the chip

WinAVR is a suite of executable, open source software development tools for the Atmel's AVR series of RISC microcontrollers hosted on the Windows platform. It includes the GNU GCC compiler for C and C++. Steps for writing a code using WinAVR

- Open the Programmer's Notepad and write your code.
- Create a new folder and save your code in that folder with extension name ".c"
- Now open the make file and edit it as mentioned bellow:
- Make file→ main filename (give your file name here without extension)
- Make file→ MCU type→ ATmega→ (chose your UC)
- Make file→ Debug format→ AVR-ext-COFF
- Make file→ Programmer→ select your programmer (if your programmer is not in the list then follow the step3.d)
- Make file→ port→ (select the port where you have connected your programmer)
- Make file→ enable editing make file→ then in your make file edit the following things

- F_CPU = 16000000 (change it as for your crystal frequency) AVRDUDE_PROGRAMMER = stk500 (here write down you programmers name)
- Save the make file in your folder without changing its name.
- Now open the programmer's notepad.
- To compile your code and to generate hex file (Tools→ make all).
- To upload your code into your UC (Tools → program).

TEST CASES

Testing is the process of executing the program with the intent of finding errors. The system has been tested according to the following test mechanisms:

7.1 Unit Testing

In computer programming a unit test is a procedure used to validate that a particular module of source code is working properly. The procedure is to write test cases for all functions and methods so that whenever a change causes a regression, it can be quickly identified and fixed. Ideally, each test case is separate from the others; constructs such as mock objects can assist in separating unit test. This type of testing is mostly done by the developers and not by end-users.

Test Case Id: 01

Test Objective: To test the false detection of speech input.

No.	Test Operation	Speech as Input	Text as Output	Pass/ Fail
1	Convert speech to text	Night	Right	Pass
2		Bag	Back	Pass

Test Case Id: 02

Test Objective: To test the connection between the Bluetooth modules.

No.	Test Operation	Connection Request Input	Connection established Output	Pass/ Fail
1	Establish socket connection between the two Bluetooth devices	Socket Connection requested from Android to the Bluetooth module	Socket Connection established. Devices Paired.	Pass

7.2 Integration Testing

Integration testing is a phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing. Integration testing takes as its input module that have been checked out by unit testing, groups them in larger aggregates, applies tests define in an integration test plan to those aggregate, and delivers as its output the integrated system ready for system testing. The purpose of Integration testing is to verify functional, performance and reliability requirements placed on major design items.

Test Case Id: 03

Test Objective: To test the reception of data on Bluetooth module.

No.	Test Operation	Send data as Input	Receive data as Output	Pass/ Fail
1	Data sent from Device to the Bluetooth module	Integer sent from the device.	Data Received.	Pass
2	Data sent from Device to the Bluetooth module	Character sent from the device	Data not received	Fail

7.3 System Testing

System testing is the testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of Black box testing, and as such, should require no knowledge of the inner design of code or logic. Alpha testing and Beta testing are sub categories of System testing. As a rule, System testing takes, as its input, all of the 'integrated' software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s).

Test Case Id: 04

Test Objective: To test the robot performance after receiving data

No.	Test Operation	Command as Input	Direction of movement as Output	Pass/ Fail
1	The robot will move according to the command received	01	Forward	Pass
2		02	Left	Pass
3		05	No change	Pass

TASK DISTRIBUTION

9.1 Work Breakdown Structure

A work breakdown structure (WBS) is a deliverable oriented decomposition of a project into smaller components. It defines and groups a project's discrete work elements in a way that helps organize and define the total work scope of the project. The WBS for the project is depicted in figure 9.1.

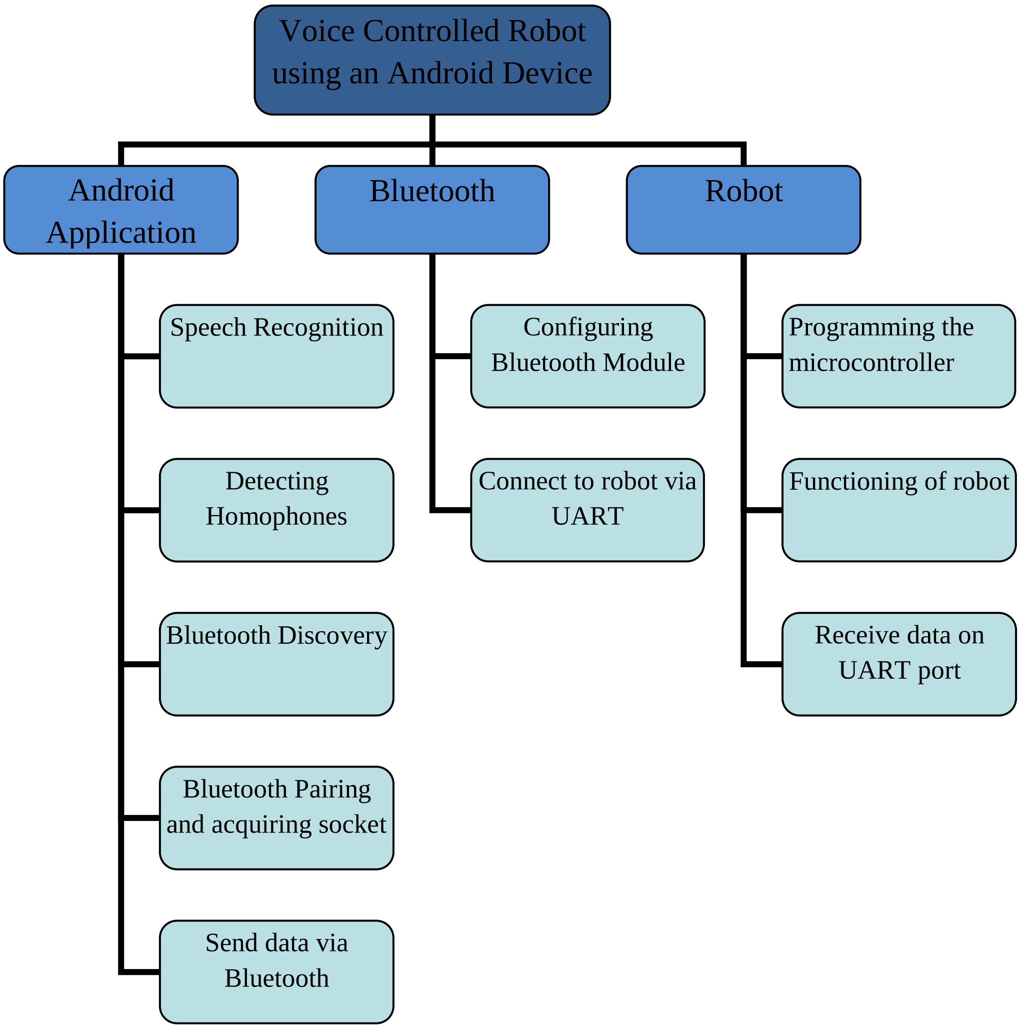

Voice Controlled Robot using an Android Device
Android Application
Bluetooth
Robot
Speech Recognition
Detecting Homophones
Bluetooth Discovery
Bluetooth Pairing and acquiring socket
Send data via Bluetooth
Configuring Bluetooth Module
Connect to robot via UART
Programming the microcontroller
Functioning of robot
Receive data on UART port

Figure 9.1: Work Breakdown Structure

CONCLUSION & FUTURE WORK

Thus the automation through voice control and also through mobile control has been designed and implemented. An Atmega32 microcontroller is used at the robot. A wireless connectivity between the android phone and the robot was developed using Bluetooth technology.

Assistive technologies have been given a considerable degree of importance since the past decade. With the advent of newer, faster and accurate microprocessors and microcontroller, development in this field got an urge to a completely new level. With our project, the handicapped can reduce its workload by using the robot as a personal assistant. For the blind, it is especially beneficial as it would be difficult for them to do work themselves. This technology can be used worldwide in various situations since a microcontroller is not just restricted to a robot.

With an additional camera feature on the robot, we can also get the live feed of the scene in front of the camera. This expands its applications like surveillance system used in military especially. We can thus send the robot to inhabited territories and navigate the robot from a distance.

Thus, to conclude, a voice controlled robot using android is a step closer to making this world an easier place to live in by automating many different processes.

REFERENCES

Technical report:

[1] Tanel Alumäe and Kaarel Kaljurand, “Open and Extendable Speech Recognition Application Architecture for Mobile Environments”, 2012.

Paper in a journal:

[2] Soon-Hyuk Hong and Jae Wook Jeon, “A Voice Command System for Autonomous Robots”, Transactions on Control, Automation and Systems Engineering Vol. 3, No. 1, March, 2001.

[3] V. Ramya and B. Palaniappan, “Embedded Home Automation for Visually Impaired”, International Journal of Computer Applications (0975 – 8887) Volume 41– No.18, March 2012.

[4] Mohri, M., "Edit-Distance of Weighted Automata: General Definitions and Algorithms", International Journal of Foundations of Computer Science, 2002.

Thesis report:

[5] Shafkat Kibria, “Speech Recognition for Robotic Control”, Master’s Thesis in Computing Science, December 18, 2005.

Website:

[6]http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1635148&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1635148

[7]http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4430710&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4430710

[8] http://developer.android.com