Smart Wheelchair for Handicapped People

WMDL Wijekoon

SP Wijayasekara

SA Weerasekara

GRGM Gamlath

EMR Tharanga

2018

Smart Wheelchair for Handicapped People

WMDL Wijekoon	ICT/2013/2014/080
SP Wijayasekara	ICT/2013/2014/078
SA Weerasekara	ICT/2013/2014/076
GRGM Gamlath	ICT/2013/2014/022
EMR Tharanga	ICT/2013/2014/071

Dissertation submitted in partial fulfillment of the requirements for the degree of Bachelor of the Science in Information and Communication Technology.

Department of Physical Sciences

Rajarata University of Sri Lanka.

2018

Declaration

The dissertation is my original work and has not been submitted previously for a degree at this or any other university/institute. To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student's name: W.M.D.L Wijekoon Date: 28.01.2018
Signature: ……..

Student's name: S.P Wijayasekara Date: 28.01.2018
Signature: ……..

Student's name: S.A Weerasekara Date: 28.01.2018
Signature: ……..

Student's name: G.R.G.M Gamlath Date: 28.01.2018
Signature: ……..

Student's name: E.M.R Tharanga Date: 28.01.2018
Signature: ……..

This is to certify that this dissertation is based on the work of W.M.D.L Wijekoon, S.P Wijayasekara, S.A Weerasekara, G.R.G.M Gamlath and E.M.R Tharanga under my supervision. The dissertation has been prepared according to the format stipulated and is of acceptable standard.

Certified by

Supervisor Name: Mrs. Thilini Irugalbandara Date: ……..
Signature: ……..

……..
Head/ Department of Physical Sciences,
Faculty of Applied Sciences,
Rajarata University of Sri Lanka,
Mihintale.
Date ……

ABSTRACT

The project focuses on developing a smart wheel chair. This product will be aimed at physically handicapped people. Input would be voice instructions and output would be the physical movement according to the voice instructions. Processing is done by a microcontroller which controls the movements. Disabled people constitute the largest minority in Sri Lanka. In February 2014, the Ministry of Health estimated that 10% of the entire population of Sri Lanka lives with some form of disability [1]. While the needs of many individuals with disabilities can be satisfied with power wheelchairs, some members of the disabled community find it is difficult or impossible to operate a standard power wheelchair. This project could be part of an assistive technology. This enables more independent, productive and enjoyable lifestyle for physically handicapped people.

DEDICATION

We would like to dedicate the final report of "Arduino based Smart Wheel Chair for handicapped people to develop the life style" to DR. Shantha Fernando, for all the ICT staff members at Rajarata University of Sri Lanka, Mrs. Thilini Irugalbandara , Mr. Buddhika Jayathilaka. And our parents who supported us to complete our project successfully.

Ultimately it is appreciated and special thanks to all colleagues of the group VITS in developing the project. Without team spirit, dedication and effort of all team members, we could not have successfully completed our innovation.

Table of content

1 CHAPTER 01 - INTRODUCTION AND OVERVIEW

1.1 Introduction

This chapter elaborates the overall knowledge about the project and defines the importance of the problem, as well as describes aim, objectives and the solution in brief.

There are many disabled people live in our society. Some of them cannot walk, some of them don't have both of hands and legs. Some are paralyzed. So they can't move anywhere. Then they need a support to move alone. People with arms and hand injury finds hard to use a normal wheelchair as they can't use their hands to control the normal wheelchair and cannot move it to any direction. Therefore, this wheelchair is manufactured to overcome the difficulties faced by such people and allow them to control the wheelchair. The wheelchair will be controlled using the voice instructions. The voice instructions to the wheelchair given by the one-sided mic placed in wheelchair arm. The voice recognition is done using programmable Raspberry minicomputer. The output from this received by Arduino. The programs in the Arduino helps to identify the signal of voice commands into considerable output and wheelchair will move consequently. By taking a wheelchair controller system persons will become more independent. By using the system, the user can operate the wheelchair by basically talking to the wheelchair's microphone. The rudimentary movement functions includes forward and back, left and right turns and stop. So the make of voice controlled wheelchair solve the problem about the movement of quadriplegic patient and make them independent of mobility.

In addition to the controlling of the wheel chair, there is a joystick. Then the person who can't only walk can operate from it by hand. The joystick is one of the easiest way to control the system. When the joystick moves the signal will be sent to Arduino to control wheelchair motors.

As well as the system can detect the obstacles around the wheelchair, because some patients may find it difficult to look around the wheelchair. So the LCD display will show the direction of the obstacle placed around the wheelchair. There are three different kinds of modes to detect the parameters within 5cm to 10cm, 10cm to 15cm, and 15cm to 20cm. It can be changed by using android application from the mobile phone.

In emergency situations the patient may need to send an emergency alert to someone who is responsible about the patient. Since the person who is responsible about the patient, cannot stay 24 hours near the patient. Then the patient has the facility of sending an SMS in case of an emergency to a responsible person.

This is the general overview of the project in brief.

1.2 Background and Motivation

Disabled persons constitute the major minority in Sri Lanka. In February 2014, the Ministry of Health predicted that 10% of the total population of Sri Lanka lives with some form of disability. Number of walking disabled people in Sri Lanka is 734,213 in 2012 (Population aged 5 years and over). The total rate per 1000 persons is 39. According to the World Health Organization, approximately 15% of the world's population is disabled [1].

Many people who need to move with help of some artificial means, whether over a sickness or an accident, is frequently growing. These means have to be progressively erudite, taking benefit of technical development, in order to rise the excellence of life for these persons and help their mixing into the working world. In this way a contribution may be made to facilitating movement and to making this increasingly simple and vigorous, so that it becomes similar to that of people who do not suffer shortages.

The smart wheelchair will help to overcome the patient's above mobility problem. The purpose of this study is to develop an interesting application using short English word identification system. The methodology is based on programming a microprocessor with a speech recognition development tools for sequestered word from a dependent speaker. The output design is used to move a wheelchair for a handicapped one based on the vocal commands. It includes the identification of isolated words from a narrow words.

1.3 Scope of the project

The **"Arduino based Smart Wheel Chair for handicapped people to develop the life style"** project is basically a hardware design. The hardware components included in the device are;

- Arduino board
- Raspberry pi 3 B
- Rechargeable battery
- USB ports
- Light emitting diodes
- Bluetooth adaptor
- GSM module
- Relays
- Switch
- Motors
- Ultra sonic sensors
- Wheelchair chasse
- Wheels
- Dot board
- Joystick

- Display
- Android mobile
- USB Mic

The scope of the project is to develop a device to facilitate the movement of the disabled people and elderly people who cannot move properly then enable them to lead better lives without any problem. The inputs of the system are voice instructions, which are predetermined and programmed in the raspberry pi. When detect and analyze the voice commands from raspberry pi, it send the signal to the Arduino.

Different kinds of processing is done inside the Arduino microcontroller which acts as the center for controlling all the components. After receiving the signal to Arduino, move the wheelchair. Depending on the voice command issued, the outputs will differ. They consists of driving the wheelchair forward, reverse, turning it left or right, turning a switch on/off and sending a pre-defined text message.

When ultra-sonic sensors detect the obstacle while moving the wheelchair, that signal send to Arduino. Then the Arduino guarantee to stop the wheelchair and send that signal to LCD display to determine which direction is the obstacle is placed. Then the user must give commands to avoid the obstacle and move.

As well as there is android application to change the mood of ultra-sonic sensors and move the wheelchair remotely. Before use the application, user must connect the mobile phone to Arduino. When android application send signal to Arduino, it control the ultra-sonic sensors and motors.

If user has emergency situation, user can press the button or give the instruction to send emergency message to someone who responsible. That signal primarily sent to Arduino Uno board, then it sent to GSM module. From GSM module that message sent to the responsible person's mobile phone.

1.4 Project Objectives

1. To control wheelchair by using voice commands.
2. To obstacles avoidance display.
3. To send emergency message to someone who is responsible.
4. Develop android application to control wheelchair and obstacle avoiding system.

1.5 Requirement Specification

1.5.1 Product Requirements

1.5.1.1 Usability Requirements

- The device shall be convenient to use by everyone and can be implemented with any third party device.
- All system users should be able to use the android application without any training.

1.5.1.2 Efficiency requirements

- As the main purpose of the device is to support handicapped people. So there shouldn't be any errors and it should be quick respond.

1.5.1.3 Performance Requirements

1.5.1.3.1 Space requirement

- The device should be compatible with any third party devices for future developments.
- System will support single user at one time.

1.5.1.3.2 Reliability requirement

- The system should provide 99% reliability. The users should be satisfied after using the device.
- Any connection shall not experience with any system major crashes or errors.

1.5.1.3.3 Portability requirement

- Since the device is implemented as an external device, it is portable, light weighted.

1.5.2 Organizational Requirements

1.5.2.1 Delivery requirement

- The anticipated device is to be developed by the end of December 2017. We hope to get the feedback from some users by giving them the chance to use the device we developed.
- The device will be developed and delivered as a single package.

1.5.2.2 Implementation Requirements

- We settle on with the agile methodology for the implementation of the purposed hardware.
- The hardware requirements should be accurate and should not have any defects.

1.5.2.3 Standard Requirements

- Documents: - Project proposal, System Design Specification and Design Documents.

- All the components of the device have to be integrated congruously.

1.5.3 External Requirements

1.5.3.1 Ethical requirement

- The device cannot be used for any unethical activities.

1.5.3.2 Legislative Requirements

1.5.3.2.1 Privacy Requirements

- By using the device no ethical and privacy requirements will be harmed.
- The authorized people can only use the device.

1.5.3.2.2 Safety Requirements

- Extended security feature is to be added.

1.6 Outline of the report

Chapter 01 – Introduction and Overview of the Project

1. Introduction
2. Background and Motivation
3. Scope of the Project
4. Project Objectives
5. Project Requirement Specification
6. Outline of the Report

Chapter 02 – Literature Exploration and Research

1. Literature Review
2. Other development research and work
3. Technologies Explored

Chapter 03 – Project Design

1. Project Concept
2. Novelty of the project
3. Project Development Methodology and Approach
4. Project Design
5. Algorithmic Design
6. Alternatives Considered
7. Detailed Project Design

Chapter 04 – Project Implementation

1. CASE Tools used
2. Activity diagrams
3. Sequence diagrams
4. Algorithm explanation

Chapter 05 – Testing and Results

Chapter 06 – Conclusion, Recommendation and Future Works

1. Conclusion
2. Recommendations
3. Future Works

2 CHAPTER 02 - LITERATURE EXPLORATION AND RESEARCH

2.1 Literature exploration

This chapter show a literature review of the products which are similar or seems to be like our product we are going to develop. It focuses on other prominent research and development works, technologies which we explored or adapted and models or concepts explored.

Our product is a smart wheelchair which provide multiple functions for user. When conducted the research using research papers, we could not find any similar product with all the features integrated as a whole unit. Our project, "Arduino based Smart Wheel Chair for handicapped people to develop the life style" is an innovation. We did a literature review on each feature separately and they are showed below.

2.2 Other development research and work

2.2.1 Control wheel chair according to the voice command

Raspberry pi recognize the voice command then process it. Finally produce specific values according to the voice command. Then Raspberry Pi send values to the Arduino Mega via serial communication. We connect Raspberry Pi and Arduino directly using USB cable. It helps to send signals faster. Arduino always waiting for the signals of RaspberryPi. When it captures a signal and identify the value that sent from Raspberry Pi. According to that value Arduino validate the method that should call. After call the method it send relevant signals to the Motor Shield of Wheel Chair. Motor shield control the whole moving process of wheel chair.

2.2.2 GPRS /GSM SIM900A modem with Arduino compatible

This is a very low cost and simple Arduino GSM and GPRS shield. We use the module SIMCom SIM900A. The Shield connects to Arduino to the internet using the GPRS wireless network. Just plug this module onto your Arduino board, plug in a SIM card from an operator offering GPRS coverage. This will help to easily send messages to the specific pre-defined mobile numbers easily.

It requires a SIM (Subscriber Identity Module) card just like mobile phones to activate communication with the network. Also they have IMEI (International Mobile Equipment Identity) number similar to mobile phones for their identification. Serial Communication library used for communicate between Arduino and GSM Module.

2.2.3 Voice recognition and filter voice from unwanted noise.

The Implementation process achieves speech to text by using Raspberry pi and Raspbian image installed on raspberry pi. This is the basic step to understand the commands of linux for converting speech to text conversion. 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.

Jasper is an open source platform for developing always-on, voice controlled applications Use your voice to ask for information, update social networks, control your home, and more. Jasper is always on, always listening for commands, and you can speak from meters away. Build it yourself with off-the-shelf hardware, and use our documentation to write your own modules. Jasper is designed specifically for the Raspberry Pi (Model B) and requires some additional hardware like a Wi-Fi adapter and USB microphone [2].

We used Ubuntu MATE image installed on Raspberry pi instead of Raspbian image, because Ubuntu MATE is easy to use and faster than other Operating Systems on Raspberry pi.

Here we used Google Speech Recognition API. Google has a great Speech Recognition API. This API converts spoken text (microphone) into written text (Python strings), briefly Speech to Text. You can simply speak in a microphone and Google API will translate this into written text. The API has excellent results for English language.
We have developed a Python code to recognize the voice through Google Speech Recognition API and filter voice from unwanted noise. A user will use pre-defined voice commands to control the wheelchair which are developed in the python code such as “forward”, “backward”, “left”, “right” and so on.

Instead of having additional Wi-Fi adaptor we used Raspberry pi 3 as it has a built-in Wi-Fi adaptor and we had to use a USB microphone.

2.2.4 Ultrasonic sensor system to detect and avoid obstacle and obstacle direction displaying system

Ultrasonic sensor provides an easy way in distance measurement. The sensor is perfect for distance measurements between moving or stationary objects. Ultrasonic Sensor measure the distance of the objects in air through non-contact technique. They measure distance without damage and are easy to use and reliable.

This application is based upon the reflection of sound waves. It can detect objects that are within a range of 2cm – 450cm and it works by sending an ultrasound pulse at around 40

KHz, it then waits and listens for the pulse to echo back, calculating the time taken in microseconds accurately.

We use ultrasonic sensors to control and indicate the position of objects and materials around wheel chair. We used 8 ultrasonic for the 8 corner of the wheel chair and it covers whole obstacles in all directions. ATmega2560 micro controller should read that 8 of echo values and process what are the distances to obstacles.

We should set a range to check the obstacles around wheelchair. That also given as a user prefer option. User can set the range of checking obstacles suitable for his environment. Android app used for select the pre-defined range mode.

Mode 1 is short range. It determines obstacles which are around 10cm to the wheel chair.

Mode 2 is mid-range. It determines obstacles which are around 20cm to the wheel chair.

Mode 3 is long range. It determines obstacles which are around 30cm to the wheel chair.

When select some mode it indicates in LED panel and user can notify which mode was selected.

Simultaneously it checks the obstacles according to the selected mode. And also it can identify the direction of obstacle using ultrasonic sensors. Then indicate the obstacle direction on LCD display.

Not only for indicate, if wheelchair hit on obstacle it will be harm to patient. Therefor we used cutoff distance of 5cm for every mode. If wheelchair more close to 5cm in any mode, then motors are stop.

2.2.4.1 Android Platform

Android is a mobile operating system developed by Google based on Linus Kernel and is mostly wield with touch screen smart phones .Android platform is reliable for building an android application, android is a common choice of people while they buy any cell phone. Android provides easy accessibility and understandability of different aspects for daily use. They are widely known as mobile computers and are expanding the sale of smartphones worldwide. They are widely accepted as they provide open architecture, platform independent and enormous capabilities. It is built from JAVA programming language and as android uses JAVA, android API provide easy access for hardware components. Android may use Bluetooth for connecting with the Smart wheel chair.

2.2.5 Develop Android application to change the moods of ultrasonic sensors and control wheelchair

Some development found when studying the research papers. In India university student made a robot using Android application and some IoT things. The compound robot will be able to move in four directions (left, right, forward, backward) and will detect the distance

of the obstacle from the robot on the android app. The main intent of that project is to design and bring about a robot prototype by using Arduino Uno, Motor Driver L293D, HC05- Bluetooth module and to procure the goal of this project, to gain knowledge about Ultrasonic sensor HCSR-04, reconcilable software and controlled motor circuit need to be determined. The robot will have several characteristics like continuous display of distance from the obstacle on the app, easy handling of a robot with the help of an app rather than any remote controller [3].

In our project Android studio is an Integrated Development Environment (IDE) for android app development. It can be downloaded at Windows, Mac and Linux smoothly, and replaced Eclipse Android Development Tools (ADT) which is Google's primary IDE for indigene application development. It has Grade based build support, Lint tools to catch performance and version compatibility. Android studio is a well-to-do layout editor which allows users to drag-and-drop UI components, and even provide user with option to preview layouts on multiple screens. There are several steps of the application. Before send data to the Arduino, should pair device to the mobile phone (HC-05 Arduino module).

After Android app first activity is connect device (HC-05 arduino module). Android application send char value to the Arduino. Then data receive from Arduino to HC-05 Bluetooth module. The advantages of this is bluetooth consumes less power than other devices and Android application is user-friendly. As well as wireless communication is enhanced.

Smart phones are becoming a rudimentary need in day to day lifecycle with huge storage capacities, fortified processors, full divertissement jobs and massive communicating methods. Bluetooth is primarily used for exchanging data among diverse devices be it two smart phones or be it a machine and a smart phone. In our project the signal from Android device will be sent through Bluetooth. This signal will be communicated with Arduino with the help of transmitter and receiver of both the devices. This signal will be represented by a single letter which change mode of obstacle detect distance.

2.3 Technologies Explored

By developing this product, we used Android, Raspberry, Arduino, Python as languages for coding and Bluetooth technology.

3 CHAPTER 03 - PROJECT DESIGN

3.1 Project Concept

Today in technological market there are large number of newly invented devices to perform multi-tasks. It was advised us to propose a new project concept which was not implemented before. So as the group VITS, it was our idea to propose a concept which would be very suitable for all kind of walking disabled persons target market with high end technologies. Most that technology devices are IoT devices. We founded most automated wheelchair not used Arduino technology. One of the new leading technology is Arduino today. We going to develop noise filtering system. Most of voice control wheelchair does not have it. It is benefit to user when user stay in noise area, he can use it easily. When using Raspberry Pi we can do noise reduction part aurally more than Arduino voice reorganization module.

Raspberry pi and Arduino incorporated together to send voice signals to motors. When studying research documents we failed to find a project similar to that. Some wheelchair use one ultrasonic to detect the obstacle in front of the wheelchair. Here we use 8 ultrasonic to sense obstacle in 8 direction. Its detection system (Example - Blink green LED bulb when obstacle detects in 20cm, Blink red LED bulb when obstacle detects in 15cm) also not found yet. So our main target is to provide multi-functional device in light weight and make it easy to use.

3.2 Novelty of the project

3.2.1 Control wheelchair according to the voice command and filter voice from unwanted noise.

In our living environment there are many kinds of high pitch low pitch noises. Vehicle sounds, animals sounds, machines sounds and etc. Sometimes human can't identify the noises of hearing to ears. Machines also the same. When machine couldn't identify user voice because of the noises it needs to filter it. We use it for our product. Unwanted voice filtering means, when user stay in loud area then speech recognizer cannot identify the user voice and background noise. We use Raspberry pi minicomputer to develop the system to filter the noise when user speak. The operating system of the Raspberry pi is Raspbian. The developed code is Python. When run the program get user voice and filter the noise and send the value to Arduino. Using that value Arduino do the motor controlling part.

Most of product use Arduino voice module to send voice command to Arduino. We need to program it do the function and must integrate to Arduino. As well as need to use pre-defined voice to identify the user voice. When use it, sometimes need to say the command as saved voice command. It's trouble in noise environment. Because there are no filtering

system to filter the user voice. Therefor we use filtering system with voice identification system using Raspberry pi.

3.2.2 Four side ultrasonic sensor system and detected obstacle direction displaying system

In many of Arduino projects use ultrasonic sensor to identify obstacle in front of the robot or machine. But they don't care about other sides. Sometimes we have to turn our wheelchair in a difficult environment. In that case we have to detect other sides obstacles.

That's the reason we include 8 ultrasonic for 4 sides in our wheelchair. We can detect all obstacles around our wheelchair. Most of products are fail in reading values from more ultrasonic sensors. Arduino board is become a slow because of more ultrasonic readings. That's happen all distance values are received at simultaneously. We change some conditions in our Arduino code and we won that part and we can handle eight ultrasonic without delays.

At same time obstacles directions are display on LCD display. We don't stop our wheelchair when detect some obstacle. It is become unhappy to user because of always it stops. Therefor we use technology for if we close to that obstacle it stops. We use pre define distance for closest range as 5cm. If any obstacle reach to that distance wheelchair is stop. That is very useful for patient protection. Therefor we use obstacle detection system for our wheelchair

3.3 Project Development Methodology and Approach

3.3.1 Agile Methodology

Requirements of a software/hardware project tend to change all the time. Software process activities may iteratively repeat to care for the changing requirements. The software / hardware is developed in short timeboxes called iterations and is released in mini-increments at the end of each iteration. The phases included in the Agile Software are;

Planning
Requirements analysis
Design
Coding
Testing and Documentation

This project is a hardware design, we choose to follow the Agile software methodology since the requirements and the aims of the project reformed time by time. While developing the prototype, Group recognized some proposed features such as "Voice Recognition by using C#" System is developed by using C# and developed by using Microsoft Visual Studio. It cannot be combined with the Arduino windows IOT due to

some minimization of system requirements. Then we moved for Linux kernel based operation system "Ubuntu MATE" and we did voice recognition with python speech recognition library and Google API.

3.3.2 Arduino Hardware Platform

In past years the employment of Arduino enhanced exponentially attributable to its readability and easiness. Arduino micro-controllers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip nonvolatile storage. Boards are loaded with program code via a serial association to a different pc. Additionally to victimization ancient complier tool chains, the Arduino provides associate degree Integrated Development surroundings (IDE) supported the process language. Once implementing our device we have a tendency to use Arduino technology to program the controls of our device with essential commands by causing to the module and instead of victimization different artificial language to try and do this dominant half operating with Arduino was build the implementation success

3.3.3 C Programming Language

This is a general-purpose, authoritative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type structure prevents many unintended processes. By design, C provides constructs that map efficiently to typical machine commands, and therefore it has found lasting use in applications that had formerly been coded in assembly language, including operating systems, as well as numerous application software for computers ranging from supercomputers to embedded systems.

The Arduino Platform uses C programming language and the main reasons why chose to program using C, are;

C is a structured programming language. It breaks and abstracts a program into small logical components which are responsible for performing a specific task. In our program we used functions to create libraries. It makes the program easier to understand and modify.

It is a powerful programming language which provides a wide variety of inbuilt data types and ability to create custom data types using structures.

It produces efficient programs hence C is a complied programming language which creates fast and efficient executable files. Also provides a set of library functions for common utilities.

3.3.4 Android Operating System

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touch screen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touchscreen devices, Google has additional developed Android TV for televisions, Android Auto for cars, and Android wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics.

We built out mobile applications using android as of the following reasons;

Andoid supports multiple languages but the feature phones only use English language in the mobile. But when using andoid it is easy to adopt to different languages. The JAVA supporting feature enables developers to enhance more features. Andoid performs multi-tasking. Users can do lots of tasks at once. Users can open several applications at once and manage them all together and also since Android has a great UI which makes easy for the users to perform their tasks. Performance stability and security is better than other mobile operating systems as it is based on Linux Kernel. Moreover, Android is very smooth and easy to operate and there are less chances of crashing down.

3.4 Project Design

"SMART WHEEL CHAIR FOR HANDICAPPED PEOPLE" is an innovative smart device. We have implemented a wheelchair to represent the innovation so we could get a use out of it in our day today lives. When using our device, the people can use it for handicapped people to control wheelchair by using voice recognition, obstacle avoidance and send emergency messages to responsible.

3.4.1 Design of components

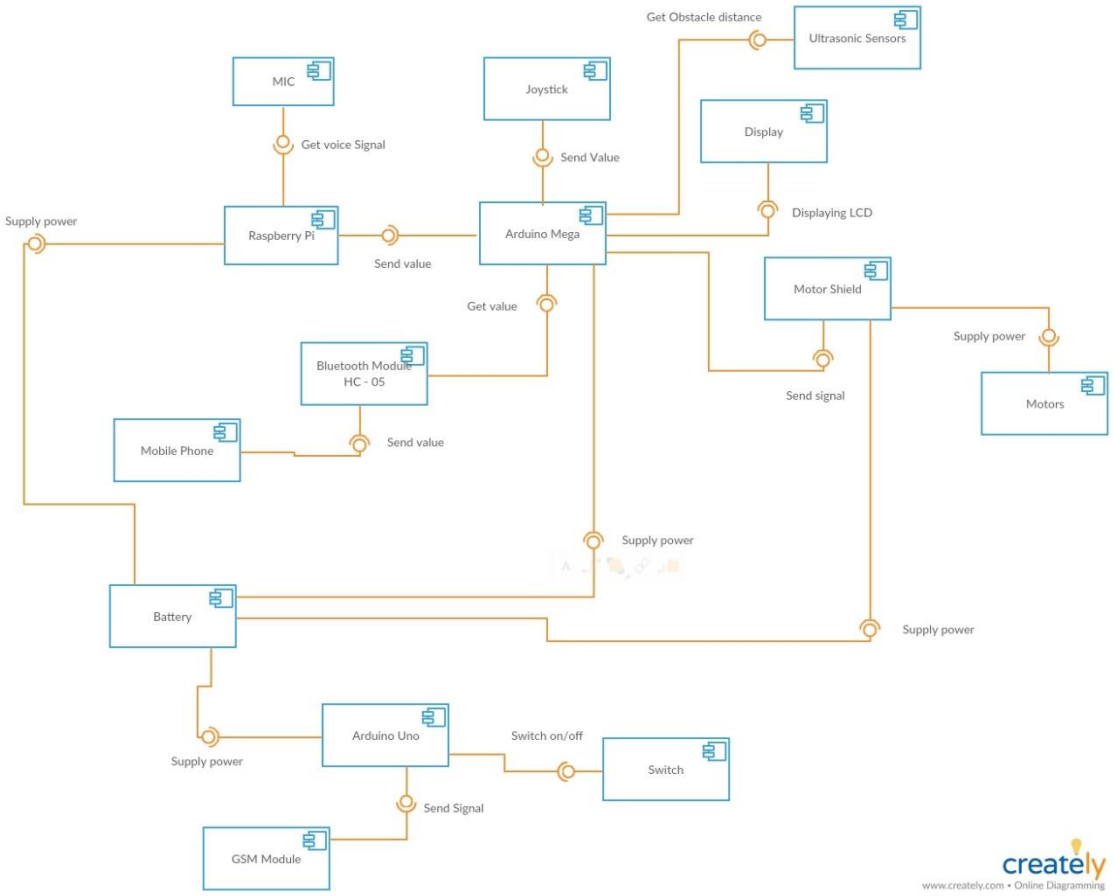

Figure 3.4.1.1

3.4.2 Component Description

3.4.2.1 Voice recognition unit

This is our main component in our project. This is for identify the voice commands. We used raspberry pi 3 for identify the voice commands. We used Ubuntu MATE image installed on Raspberry pi 3 because it light operating system than raspbian, windows IOT etc. Its include python powered code for recognize the voice. The Google Speech Recognition API used to get predefine voice commands. As well as developed python code for filter voice from unwanted noise. We use Raspberry pi minicomputer to develop the system to filter the noise when user speak. Then user can use it any noise area and clearly identify the voice commands. Using that value Arduino do the motor controlling part. Most of product use Arduino voice module to send voice command to Arduino. A user will use pre-defined voice commands to control the wheelchair which are developed in the python code such as “forward”, “backward”, “left”, “right” and so on.

When values pass to the Arduino board, it has method for identify the values and control the motors through motor shield. There are two viper motors locate in the backside in the wheelchair. Because the motors are locating in the back side of the wheelchair it easy to move smoothly. Likewise the voice recognition unit work with Arduino.

3.4.2.2 Obstacles detection and displaying system

This component use for detecting obstacle around the wheelchair. Therefor user could be identifying obstacles without an effort. Ultrasonic Sensor measure the distance of the objects in air through non-contact technique. They measure distance without damage and are easy to use and reliable. Its more safety because of user can identify the obstacles around the wheelchair.

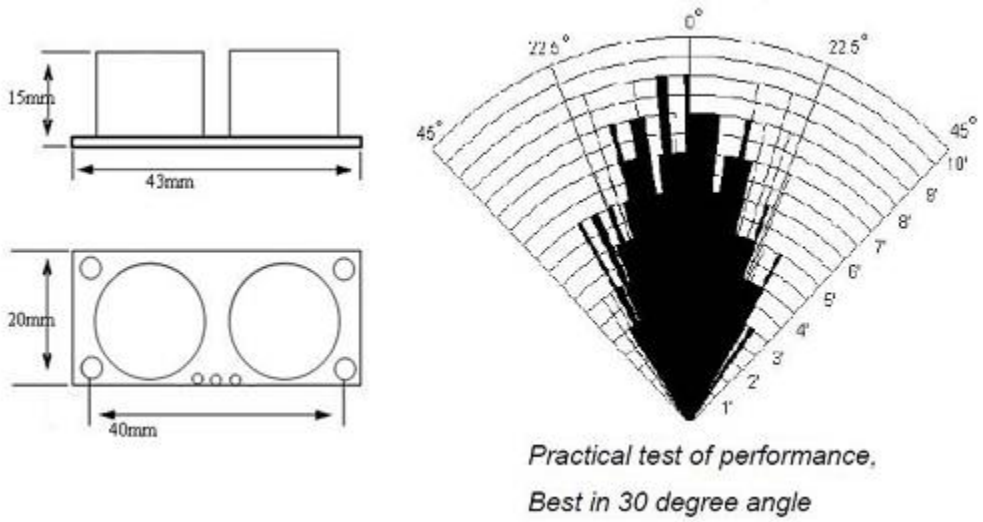

Figure 3.4.2.2.1

When see the figure 3.4.2.2.1 we can identify it has small range such as 30 degrees. So It cannot cover 180 degrees in a one side. That's why we cannot use four ultrasonic for four directions. Therefor used 8 ultrasonic in eight corner of the wheelchair. It can cover all the directions around the wheelchair.

There is an android app for change the range modes of obstacle detection unit. There are three range of modes.

Mode 1 is short range. (10 cm)

Mode 2 is mid-range. (20 cm)

Mode 3 is long range. (30 cm)

When select a mode using android application it indicates in LED panel locate in the wheelchair. This application and Arduino board is connected in Bluetooth module. Before use the android application user must pair the mobile phone with Bluetooth module.

When detecting obstacle, it direction is display in LCD display on left hand side armor. Then user can easily identify the direction of the obstacles. It will help to user for move the wheelchair easily.

Not only for indicate, we used cutoff distance of 5cm for every mode. When wheelchair more close to 5cm in any mode, then motors are stop.

3.4.2.3 Joystick control unit

This another way to move the wheelchair. Joystick also connect to the Arduino and ultrasonic system. It's for patients who can manually use the wheelchair without using voice commands. Some environments can be more loudness. In that kind of situation this system is more useful for easily move wheelchair.

When rotating joystick using button, it pass the values to Arduino board. It includes the methods and identify the direction to move. When moving the wheelchair, it check the obstacles and display the obstacle`s directions on the LCD display.

3.4.2.4 Emergency message sending unit

GPRS/GSM Module is an ultra-compact and reliable wireless module. It is a breakout board and minimum system of SIM900 Quad-band GSM/GPRS module. It can communicate with controllers via AT commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands). This module support software power on and reset.

The GPRS is configured and controlled via its UART using simple AT commands. Just connect on the Arduino / Raspberry Pi/AVR/PIC/ARM/FPGA board, you could easy to use AT command control it.

This module is create to send a message to the responsible party of the handicapped person in case of emergency. Initially the module establish a connection with the service provider, then the red light start to blink rapidly.

When connection is totally establish the blinking speed of the red light decrease.

This govern the interpretation whether the connection establish or not. For the detection of signals there is a separate antenna fixed in to the module. It is an essential facts that antenna and module has to attach enough. It only takes 10-30 minutes for the establishment of the connection.

3.4.2.5 Define Interface between Components

When designing the electrical system for the wheel chair, it was very important to keep a clear understanding of how all of the major components interface with one another. This system incorporates several different types of interfacing techniques.

The important stage of this project is automating it. This means that the wheel chair will be able to complete the task according to the voice commands. This part of the project is completely software and hardware based and will require an extensive amount of programming, and voice commands to successfully have wheel chair operate.

The wheel chair will also be fitted with sensors that will scan the area in front of the mower to detect any objects that may hinder the wheel chair's movement, cause harm to the mower, or cause harm to the object. The sensor will relay its information to the AtMega32 and it will analyze the data. If the program sees a foreign object, the route at which the wheel chair is will be altered to avoid the object.

The integration between the sensors integrated onto the wheel chair and AtMega32 and it will also have to be tested to ensure is constantly receiving their corresponding data. The longevity of the battery providing power to the AtMega32 and it will have to be tested.

Our project will require a good amount of wiring and connections between components. Wire, batteries, and other small parts will be required to properly connect, fasten and operate the components.

Possibly the most critical portion of the electrical system is the software that runs on the Arduino. The goal of this portion of the design was to create a single program written in the Arduino language that would fulfill several objectives. The software on the Arduino must also monitor the voltage level of the lead-acid batteries powering the system.

Micro Controller send messages according to the voice command to the pre-defined mobile numbers through the GSM module.

3.4.2.6 Deferent Process and Their Communications

In the wheel chair there are number of processes to operate it by internal component communication.

When user switches on the machine, it checks whether there is power to start the machine. When power on the machine all the interact components of the machine will started. (Battery, Micro controller, sensors etc.)

After machine started, microcontroller controls the automated machine by obstacles detecting and movements according to the voice commands.

Machine detects the path as programmed by microcontroller.

Micro Controller send messages according to the voice command to the pre-defined mobile numbers through the GSM module.

Machine detects the obstacles by using Ultrasonic sensors. While moving the machine, Ultrasonic sensors detect the obstacles and send signal to microcontroller. Then microcontroller handles the path of the machine to avoid obstacles by turning wheels and

move forward. Ultrasonic sensors detect obstacles 1m far and inform it to microcontroller to change the direction.

All the processes should communicate each other well to perform the functions of whole application.

3.4.3 Physical Arrangement of the device

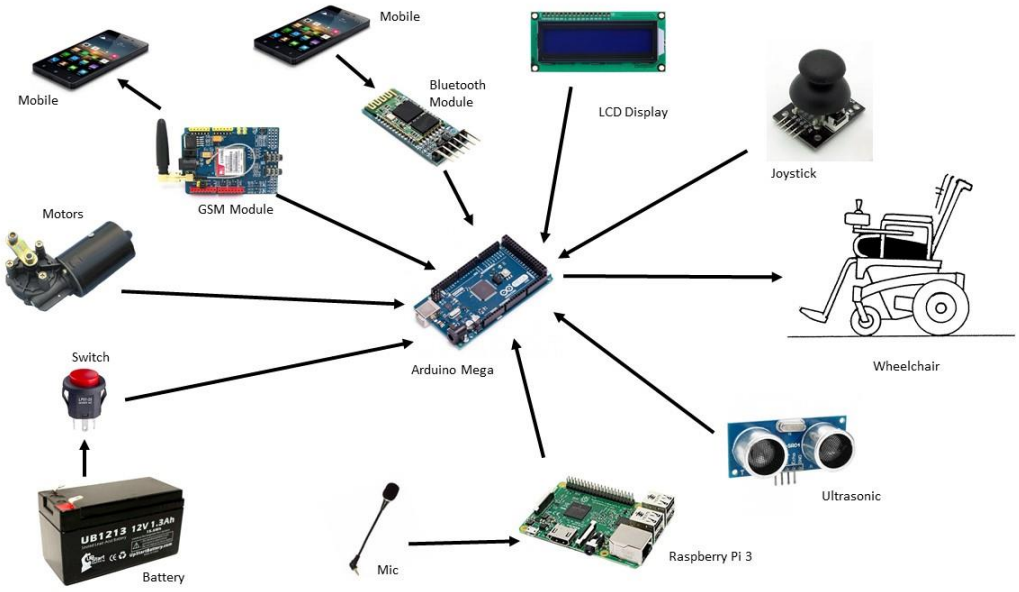

Figure 3.4.2.2.1

Above design shows that physical arrangement of devices and their communication.

In the beginning we have to enter coding to the Arduino through the computer. We do coding using C++ language. This coding part is the software part in our project. We do coding to detect voice commands, and obstacles. All coding build with Arduino – nightly software. Microcontroller, Ultrasonic sensors and other related components are fixing to the Arduino. 12V of battery and the battery provides its power to 2 motors. Two motors use to provide power to turn backward wheels. Then all these Arduino, motors, battery are set to the vehicle body. Finally we can get smart wheel chair. To bring our wheel chair to supposed level, there should be good interaction and communication among all these components.

3.4.4 Final product

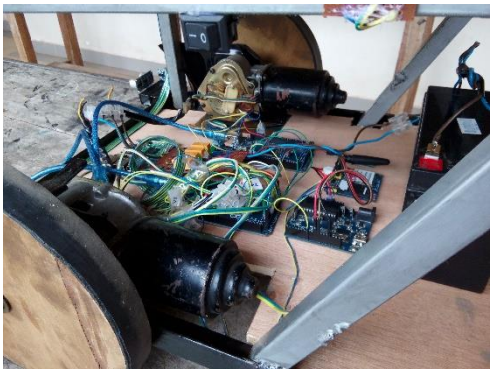

Figure 3.4.4.1 Circuits, motors, battery placement of wheelchair

Figure 3.4.4.2 Physical feature of wheelchair

Figure 3.4.4.1 Joystick position of right arm of the wheelchair

Figure 3.4.4.4 Ultrasonic mods selection indicating LEDs and emergency message sending switch

Figure 3.4.4.5 four direction ultrasonic sensor placement

Figure 3.4.4.6 Obstacle displaying system

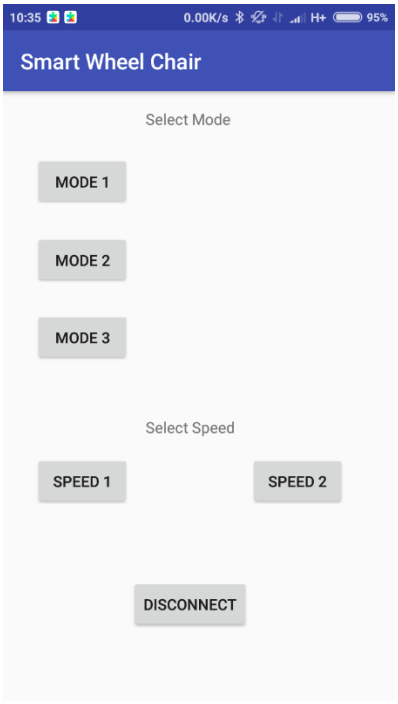

Figure 3.4.4.7 Android application

3.5 Algorithmic designs

Algorithm for "Switch on device"

1) Switch on device.

2) Check the LED bulbs on the board.

3) If green bulb is diffusing,
    a. The device is suitable to use.
    b. If not the device has to be charged.

Algorithm for "Voice Recognition"

1) Speak to the microphone using predefined voice command

2) If command is correct
    a. Wheelchair moves to the correct way

3) If not an LED bulb is blinked
    a. go to step 1

Algorithm for "Emergency message sender"

1) Check LED bulb whether the connection is established

2) If LED blinking rapidly
    a) Connection is not established yet

3) If LED blinking with a long delay
    a) Connection is established

Algorithm for "Obstacle Detection"

1) Select a distance mode using the application

2) LED is blinking according to the mode

3) If obstacle detected within the range
    a) Th0e side of the obstacle will be display on LED display

4) If obstacle is within the danger area
    a) Stop message will be display on LED display
    b) Wheelchair will be stopped

Algorithm for "Mobile Application"

1) Turn on Bluetooth

2) Create connection

3) Select distance mode

4) If bulb is blinking
    a) Reliable connection

5) If bulb is not blinking
    a) Unreliable connection

Algorithm for "Bluetooth disconnected"

1) Disconnect Bluetooth

2) Show paired device

3) If show paired device
    a. Turn off Bluetooth

4) Close application

3.6 Alternatives Considered

3.6.1 Using Arduino instead of using another microcontroller.

Using an Arduino make things easier the amount of hardware and software development needed to do to become a system running.

The Arduino hardware platform now has the power and reset circuitry setup as well as circuitry to program and connect with the microcontroller over USB. In addition, the I/O pins of the microcontroller are typically already fed out to sockets/headers for easy access.

On the software side, Arduino provides many libraries to make programming the microcontroller easier. The simplest of these are functions to control and read the I/O pins rather than having to fiddle with the bus/bit masks generally used to interface through the Atmega I/O. More useful are things such as being able to set I/O pins to PWM at a certain duty cycle using a single command or doing Serial communication. When working with the Arduino there are many examples that they have provided to test the codes.

3.6.2 Using Android instead of other operating systems.

Android is an open source operating system which is use by mobile phones, TV streaming boxes and tablets. It is developed by Google based on Linux kernel. Android is basically a piece of software which allows your hardware to function. The Android OS gives you access to apps, including many of Google's own creation. These allow you to look for information on the web, play music and videos, and check your location on a map, take photos using your device's camera and plenty more besides. Android is portable because native Android apps are developed using the Java programming language, and can easily be ported to other mobile operating systems like Blackberry, Symbian and Ubuntu. In addition, Android apps can also be ported easily to Chrome OS.

The Android has an IDE which can build mobile applications easily. Android Studio is an excellent IDE based on the equally excellent IntelliJ IDE. As the name suggests, Android Studio is an IDE designed and developed specifically for Android app development. It is blazingly fast and efficient, and you can setup a new Android project for different types of Android apps within seconds. When Android was launched, Android app development was done with Eclipse and the Android Developer Tools plugin.

3.7 Detailed Project Design

In the detailed project design, we have included the use case diagrams and class diagrams which describe the procedure of the device.

3.7.1 Use case Diagram

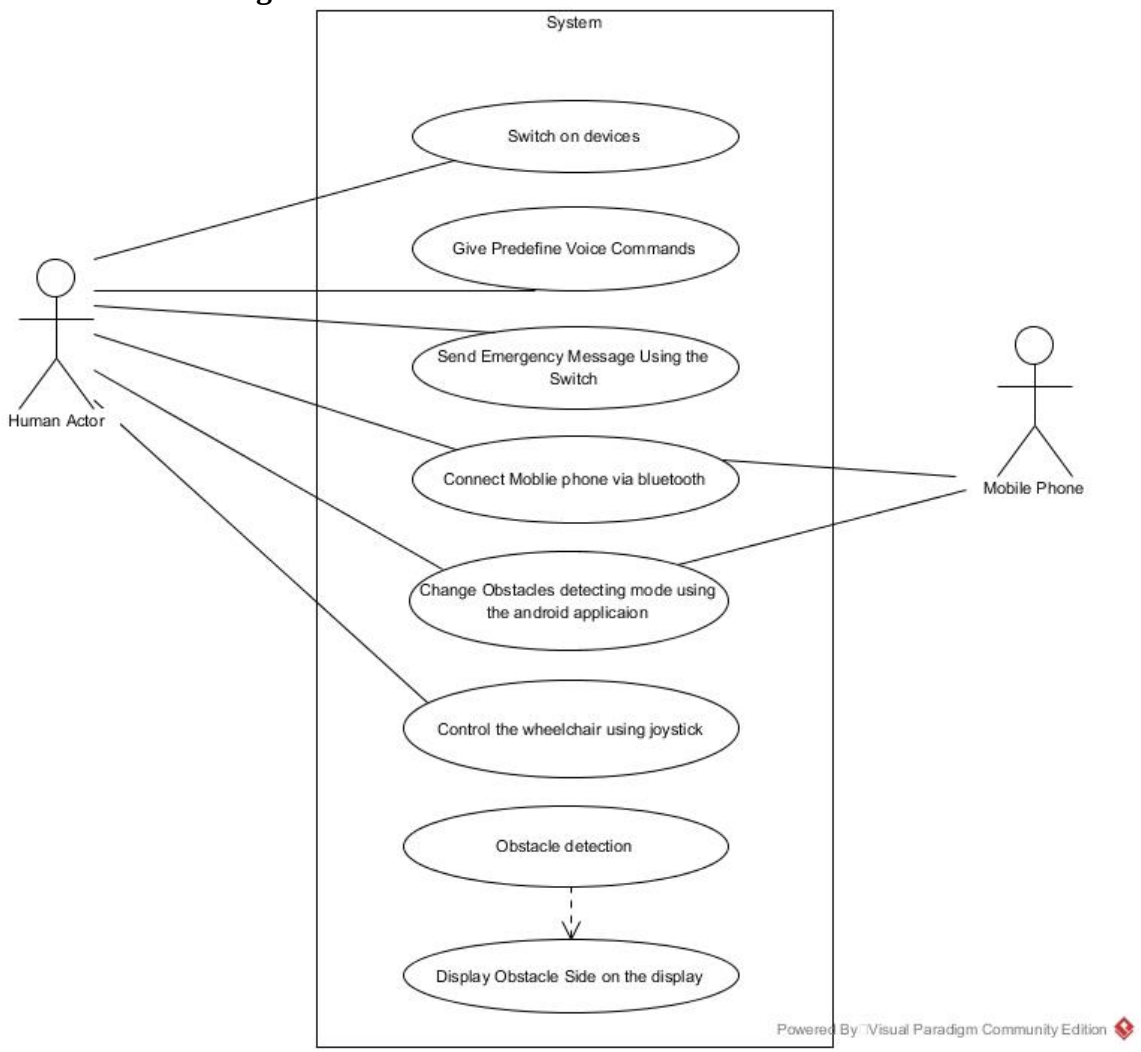

Figure 3.7.1.1

Description about actors and use cases

Use cases

- Switch on device
- Give predefined voice command**s**
- Send emergency message using the switch
- Change obstacle detection mood using the android application
- Connect mobile phone via Bluetooth
- Connect the wheelchair using joystick
- Obstacle detection
- Display obstacle detection side on the display

Use case scenarios and alternative scenarios

Use case 01

Use case name	**:**	Switch on devices
Overview	**:**	Switch on the device and give power
Actor	**:**	Human actor

Pre-condition

- Check the battery power

Flow of events

- Switch on the device using main switch
- Convert 12V DC power to 5V DC and given to the Arduino board and Raspberry pi.
- Flow 12V DC to the motor shield.

Post-condition

- Flow power to the Arduino board, Raspberry pi and motor shield.

Use case 02

Use case name	**:**	Give predefine Voice command
Overview	**:**	Give voice commands
Actor	**:**	Human actor, Command recognition component

Pre-condition

- Only the Pre-Defined voice commands should be given.

Flow of events

- Input the voice commands
- Recognize the voice command by Raspberry Pi's microphone and if those commands are pre-defined ones send signals to Arduino component

Post-condition

- ATmega32 accepts the voice commands and control the motors accordingly.

Use case 03

Use case name	**:**	Send Emergency Message
Overview	**:**	Send SMS to selected Mobile Numbers.
Actor	**:**	Human Actor and Command recognition component

Pre-condition

- The connection should be established between the device and GSM module.

Flow of events

- switch on/off
- Identify the signal
- Send predefined message to the pre-defined mobile numbers

Post-condition

- Respond message would be shown by blinking led

Use case 04

Use case name	**:**	Connect mobile phone via Bluetooth
Overview	**:**	Establish a connection between mobile app and Arduino board
Actor	**:**	Human Actor, Mobile App, Bluetooth module component

Pre-condition

- Pair Bluetooth module and android device.
-

Flow of events

- Select the device using mobile app.

Post-condition

- Establish a connection between mobile app and Arduino board.

Use case 05

Use case name	**:**	Change obstacles detecting mode using mobile app
Overview	**:**	Switch to 3 Different Distance from Android App
Actor	**:**	Human Actor and Distance Changing component

Pre-condition

- The connection should be established between the device and mobile phone via Bluetooth.

Flow of events

- Connect mobile phone and Arduino mega.
- Identify the signal.
- Change Arduino obstacle detection distance according to the selected mode.

Post-condition

- Change bulb blinking according to the mode.

Use case 06

Use case name	**:**	Control the wheelchair using joysticks.
Overview	**:**	Manually move the wheelchair without voice commands.
Actor	**:**	Human Actor, Joystick unit

Pre-condition

- Disable giving voice commands

Flow of events

- Identify the signal
- Avoid obstacles using given mode

Post-condition

- Move according to the joysticks

Use case 07

Use case name	**:**	Obstacle detection
Overview	**:**	Avoiding obstacles automatically by detecting obstacles
Actor	**:**	System Actor – Obstacle avoiding component

Pre-condition

- Check the emitting of led

Flow of events

- Identify the distance with Ultrasonic and control the movement of wheel chair

Post-condition

- If the light is not emitting obstacle avoiding system doesn't work

Use case 08

Use case name	**:**	Display obstacles side on the LCD display
Overview	**:**	Indicate the direction of the obstacles
Actor	**:**	Human Actor, LCD unit

Pre-condition

- Check the LCD light emitting

Flow of events

- Identify the signal

Post-condition

- Display according to the

3.7.2 Class Diagrams

Establish the connection with Arduino

Figure 3.7.2.1

Send value to Arduino

Figure 3.7.2.2

4 CHAPTER 04 - PROJECT IMPLEMENTATION

4.1 CASE tools used

Computer Aided-Software Engineering (CASE) is the application of a set of tools and methods to a software system with the desired end result of high quality, defect-free, and maintainable software products. It also refers to methods for the development of information systems together with automated tools that can be used in the software development process.

In our project we used Visual Paradigm to draw various diagrams such as class diagrams, activity diagrams, sequence diagrams and use case diagrams.

Visual paradigm

Visual paradigm is a software tool designed for software development teams to model business information system and manages development processes. Visual Paradigm supports key industry modeling languages and standards such as Unified Modeling Language (UML), SysML, SoaML, BPMN, XMI, etc.

Advantages;

- Persistence Made easy
- Model driven development
- Extensive database coverage
- IDE integration

4.2 Activity diagrams

Voice recognition when give voice command

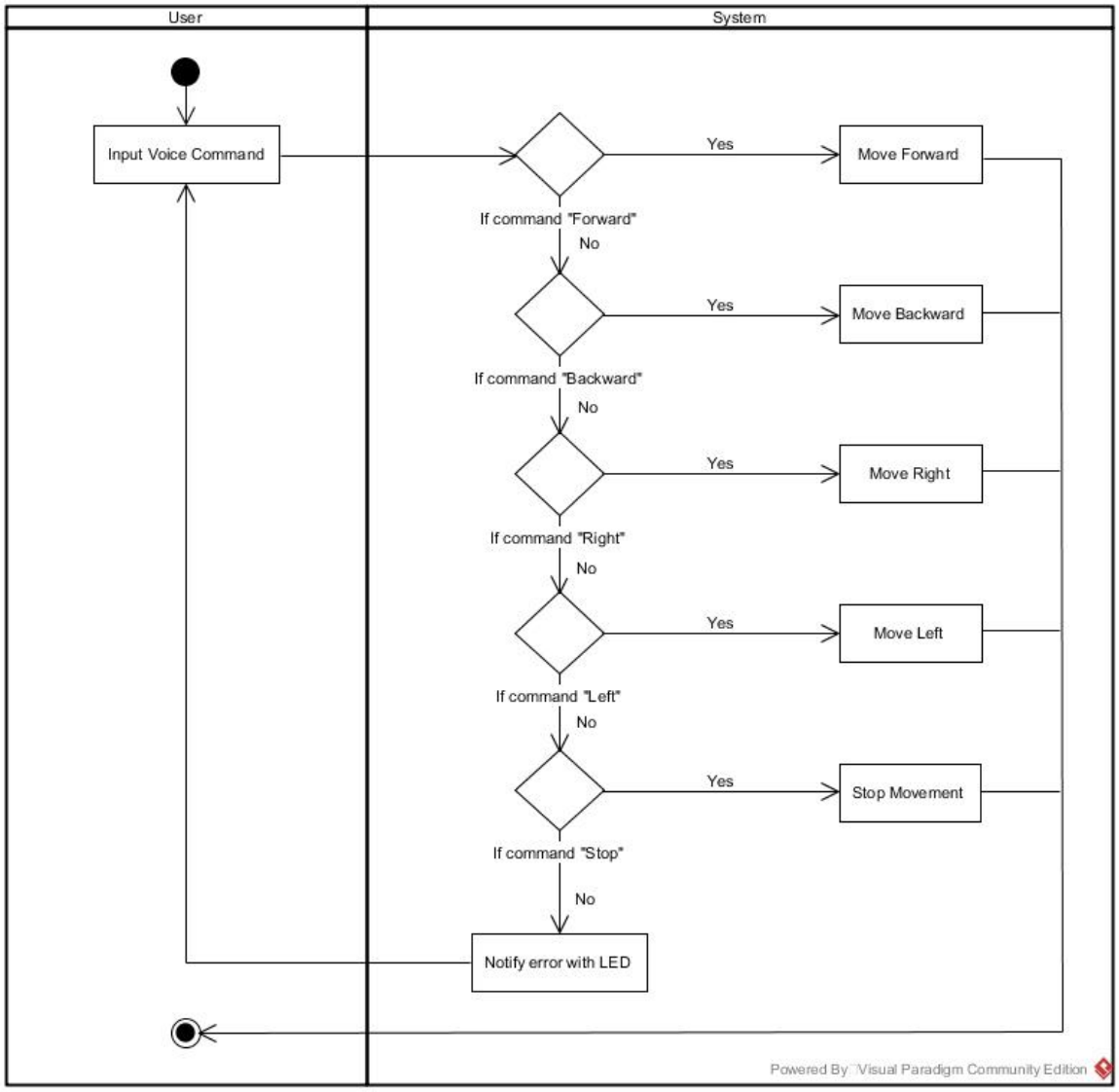

Figure 4.2.1

Obstacle detection

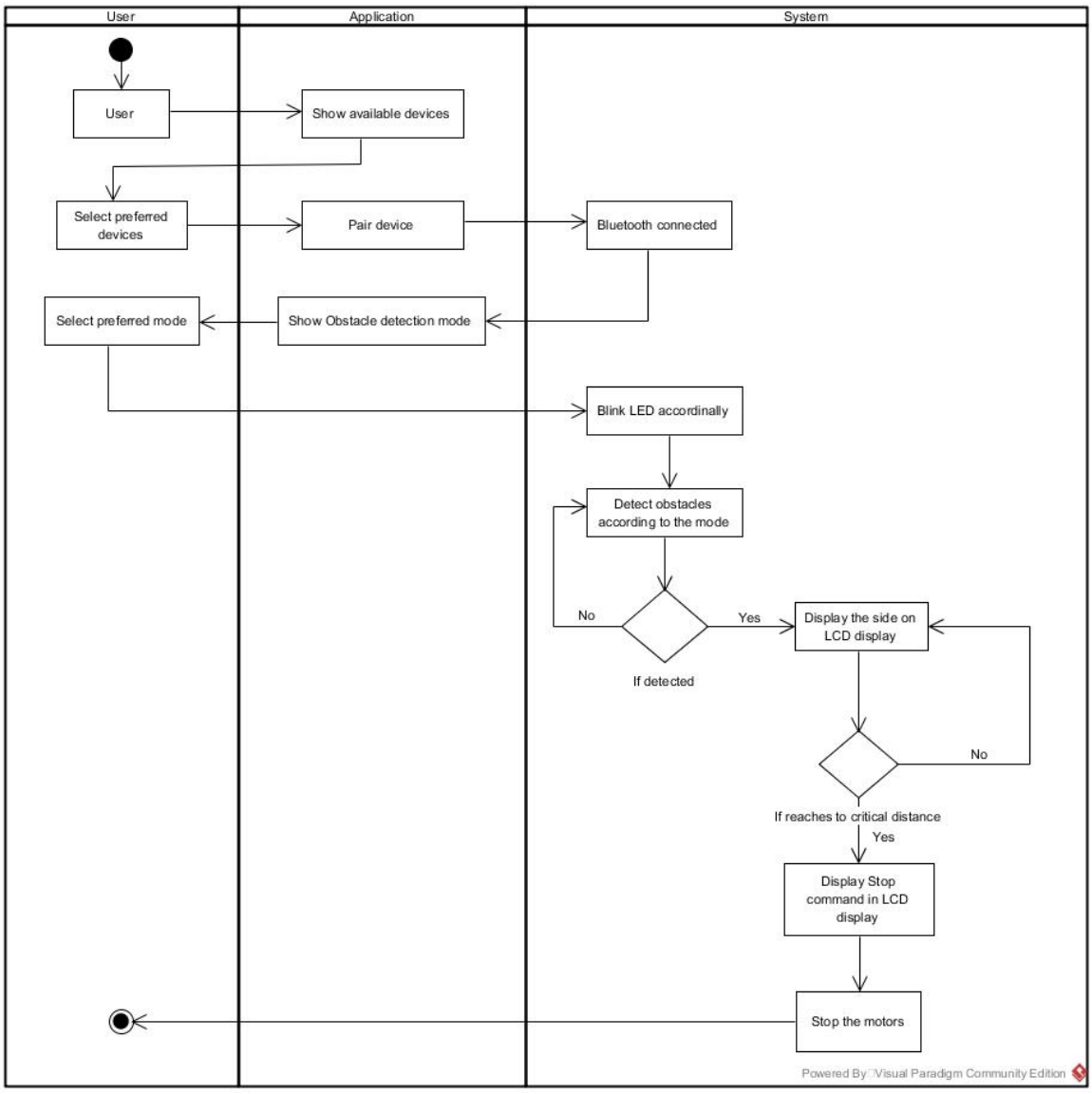

Figure 4.2.2

Joystick control to move wheelchair

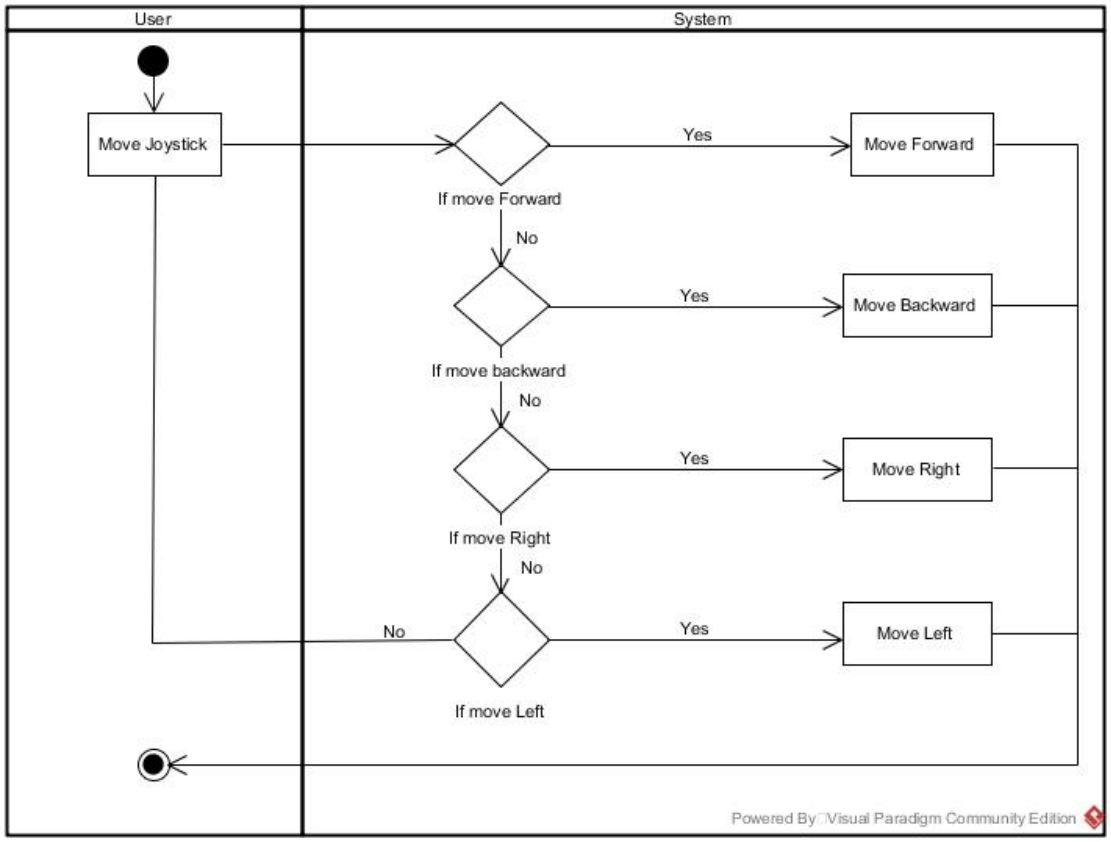

Figure 4.2.3

Emergency message sending

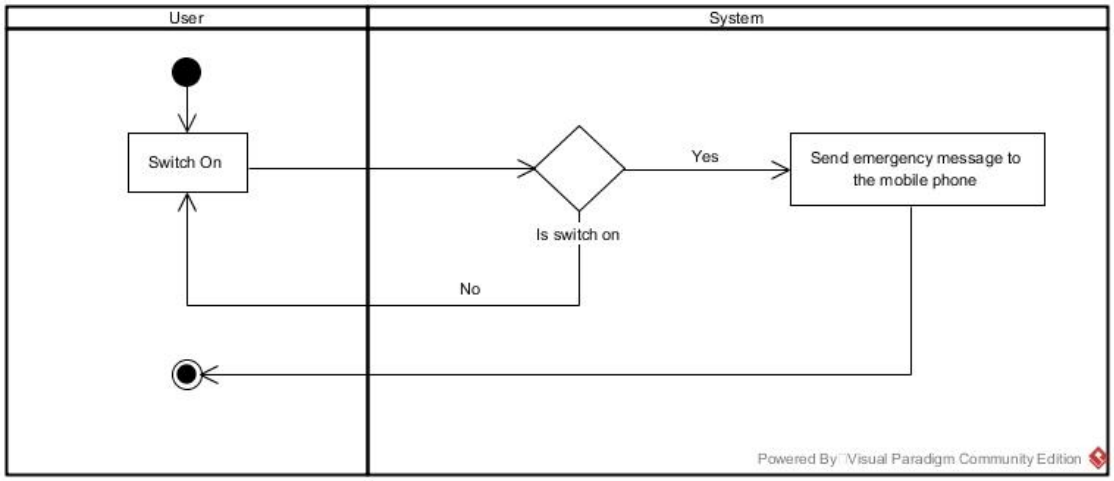

Figure 4.2.4

4.3 Sequence diagram

Voice recognition

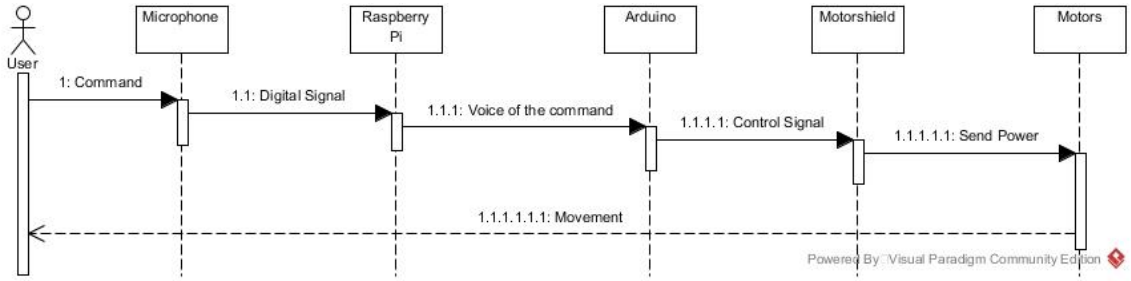

Figure 4.3.1

Joystick control

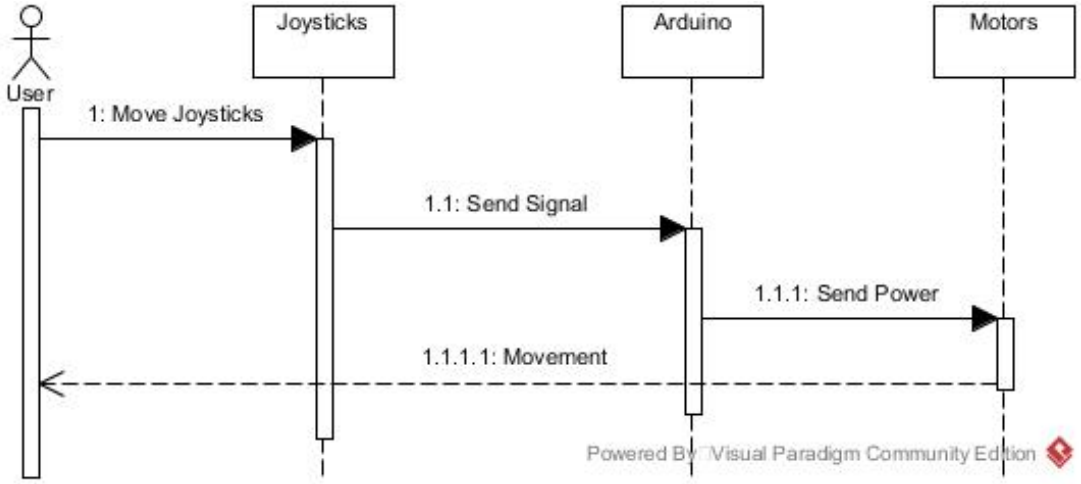

Figure 4.3.2

Emergency message sending

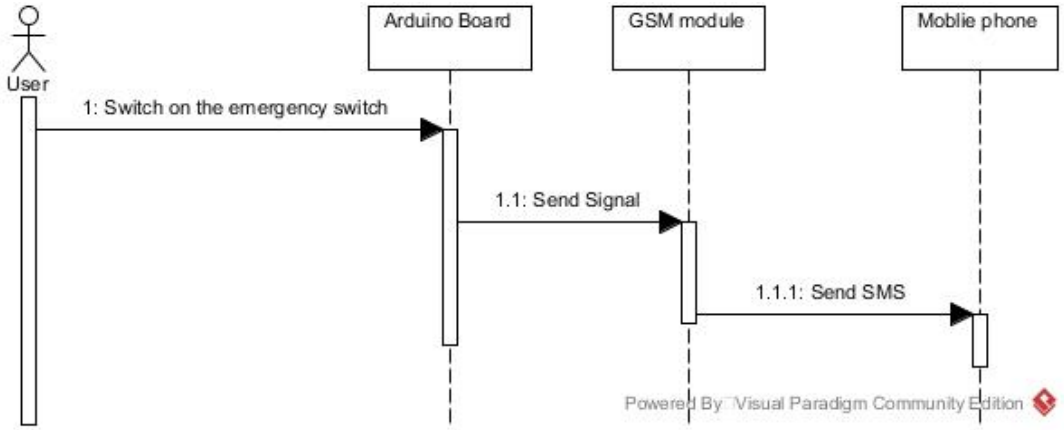

Figure 4.3.3

Establish Bluetooth connection

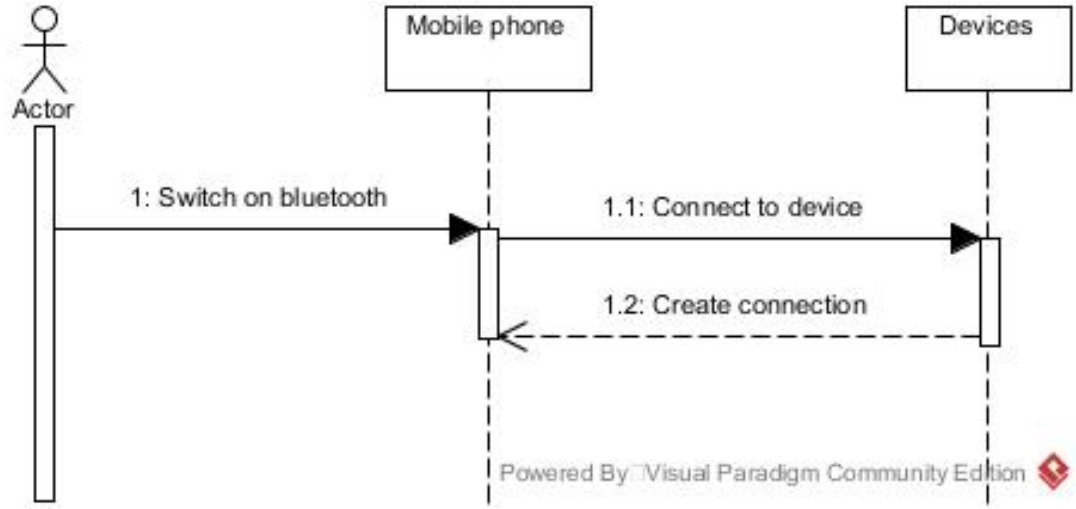

Figure 4.3.4

4.4 Algorithm explanation

Voice recognition

User inputs voice commands to raspberry pi 3 via microphone. In the python code it is identified by the Google Speech Recognition API and unwanted noise is removed. In the python code in the raspberry pi, it uses predefined voice commands such as forward, backward, left etc. If the given command is a predefined one, such as forward, backward, left, right or stop raspberry pi sends signal accordingly to the Arduino board. If the given command is not a pre-defined one an error message will be given to the user. Refer the voice recognition code in appendix (figure 8.1 and figure 8.2).

Obstacle detection and displaying

This system selects the range mode using received value from android application. It receives 'A' mode 1, 'B' mode 2 and 'C' mode 3. According to the selected mode LED bulb will indicate. Yellow LED for mode 1 short range, Orange LED for mode 2 mid-range, Red for LED mode 3 long range. After that it checks the obstacles according to the selected mode and it displays on LCD display. If any obstacle reaches to the 5cm wheelchair will stop. If obstacle is within the danger area, stop message will be display on LED display, wheelchair will be stopped. Refer the obstacle detection code in appendix (figure 8.3, figure 8.6 to figure 8.14)

Mobile application

Firstly, should pair the mobile application with Bluetooth device. Select Bluetooth device in Arduino and connect to it. Select a mode using application. If selected mode is 1, 'A' value pass to Arduino board through Bluetooth device. If selected mode is 2, 'B' value pass to Arduino board through Bluetooth device. If selected mode is 3, 'C' value pass to Arduino board through Bluetooth device. Refer the mobile application code in appendix (figure 8.20 to figure 8.26).

Joystick control

This is for manual use without voice instructions. When we press for forward, it passes x and y coordination to the Arduino board (x > 1021 && x < 1024 && y > 516 && y < 519) Then it will move forward. When we press for back, it passes x and y coordination to the Arduino board (x == 0 && y < 519 && y > 516) Then it will move backward. When we press for left, it passes x and y coordination to the Arduino board (x > 527 && x < 531 && y == 0) Then it will move Left. When we press for right, it passes x and y coordination to the Arduino board (x > 528 && x < 531 && y > 1021 && y < 1024) Then it will move

right. Without pressing anyone it`s in neutral, it passes x and y coordination to the Arduino board (x < 532 && x > 528 && y > 516 && y < 519) Then it not moves. Refer the Joystick control code in appendix (figure 8.4 and figure 8.5, figure 8.15 to figure 8.17).

Emergency message sending

Before press the emergency button should check the LED bulb is blinking rapidly or blinking long delay. If LED is blinking rapidly, the connection is not established. Then system cannot send message. If LED is blinking long delay, the connection is established. Then system can send message. Message body of the emergency message is “immediately contact your patient”. It is pre-defined one. Refer the Emergency message sending code in appendix (figure 8.18 and figure 8.29).

5 CHAPTER 05 - TESTING AND RESULTS

5.1 Test results based on voice commands

No	Speech	No. of words spoken	No. of times system correctly responds	Accuracy
1.	Noisy environment	50	41	82%
2.	Silent environment	50	48	96%

Table 5.1

5.2 Other tests and results

No	Test Case Name	Result	Status
1.	Check whether the obstacle detection mode is changed in the app correctly.	The three LEDs are blinked according to the selected mode.	PASS
2.	Check whether the joystick controller is working properly.	Movements of the wheelchair according to the controller.	PASS
3.	Check if the emergency message is sending to the mobile phone.	When press the emergency button a message will be sent to a responsible person's mobile phone.	PASS
4.	Check if the emergency message is sending to the mobile phone in low signal area.	The message will not sent to a responsible person's mobile phone.	FAIL
5.	Obstacle detection system's accuracy.	The side of the obstacle will be shown on the LCD display.	PASS
6.	Build connection between user's mobile phone and device through Bluetooth Technology using the mobile Android App.	Establish connection between user and device using Bluetooth connection (used Arduino code)	PASS

Table 5.2

6 CHAPTER 06 - CONCLUSION, RECOMMENDATIONS AND FUTURE WORKS

6.1 Conclusion

The project “Smart Wheelchair for Handicapped People”, was designed for the people in Sri Lanka and especially focused on Handicapped People. The idea was basically developed while we were in the university. We distinguished that Handicapped people use wheelchairs for day today activities but if there is person without arms he or she need another person to pull the wheelchair. We recognized this problem and decided to develop a solution.

At first, we definite how to gather the hardware components in order to make it a one device because of some inspiration from the project panel and we went on adding and removing features to make it to the standard required. First, we developed the wheelchair chasse according to the physics because it will help to integrate all other components. It was our preliminary step of our project.

The main feature of our project control wheelchair according to the voice commands. So we developed a voice recognition application using python programming language.

In here the voice command is recognized by the help of google voice recognition API. The recognized voice command is validated with Pre-defined commands and send to the Ardiuno Mega by giving a specific value.

Thereafter Ardiuno Mega accept the signals which come from Raspberry Pi, then control & move the wheels of the wheel chair based on relevant commands come through the motor shield.

The duty of the ultrasonic is to check whether there is an obstacle for the movement of wheelchair within the specific region. If an obstacle to be find in the critical range the wheel chair may stop.

The mobile app consists of 3 modes which are 30cm, 20cm, and 10cm enabling to fluctuate the detecting capability.

In addition, the wheel chair can be control by using a joystick also. It has the options for forward, backward, left and right controlling commands.

Display exhibit the direction of the obstacle which is detected.

In case of emergency, a GSM module is made use of to send a message to a responsible person via a predefined mobile number.

We Used Android Studio to develop Android application and C programming language for program Arduino. Arduino IDE helps for do this job get done easily.

We used our knowledge of the degree program to assemble all the circuits and also the electronic parts with the assistance of our lecturer. Embedded systems were a conjointly nice support in collection the hardware parts and that we perfect the air of mixing circuits, wires, regulators and etc. mobile application used our OOP knowledge we have a tendency to gained within the educational study.

We have effectively addressed our concept and accomplished our objectives and goals of the project. This could be more developed with research areas and the device could improve more features.

6.2 Recommendations

Our main Target is Sri Lankan Handicapped people. Therefore, handicapped people and their responsible persons are our main client and customers. Using of our smart wheel chair, user can easily control the wheel chair with voice command. It is very easy other than using effort of pulling wheelchair by using hand. We highly recommend our "Smart Wheelchair" project fir handicapped people. This will help people to save their energy and do their day to day activities easily.

Statistics in 2012 of Sri Lanka issued by UNESCAP notify that approximated 1.7 million persons live with a disability out of country's 21 million population. Approximately this is 10% of the population. The Sri Lankan Ministry of Health issued a statement in 2014 that the number of people with disabilities is likely to increase by 24.4% in 2040.

Therefore we have to consider about future as well as present. Our project will be surely fit for handicapped people on future as well as present. Human life style will be better than it was, with new technology.

This innovative smart wheel chair is more supportive not only for independent handicapped people but also for dependent once. Therefore we believe this is a revolutionized utmost patronage to standardize the daily life of the handicapped people from today to tomorrow.

This would be a very useful and innovative product which developed for the handicapped people recently

6.3 Future work

Mass Production

Our device is an innovation and this is not in the market for users to purchase in the present. So we decided to produce this in mass production with a company and place this in the market. Then it will be helpful to earn foreign exchange by exporting our wheelchair.

Wireless phone charger

We can implement mobile phone charging to “Smart Wheelchair”. It will help to person to charge their mobile phone and other USB equipment’s easily.

Water cooling seat system

We hope to implement our wheelchair seat with water cooling system to provide best combatable for patient. It is more comfortable for countries near the equator. Because of these countries are hotter than other countries.

Support for more Language

We decided to develop our wheelchair to support more languages other than English. Then many of world wild disability patient can help to move along. Currently voice commands are predefined with English.

Solar power battery charging system

Solar power battery charging system can charge the wheelchair battery by using solar power. It is useful when the wheelchair battery is low. As well as it is energy saving method to save home electricity.

7 CHAPTER 07 - LIST OF FIGURES AND TABLES

8 CHAPTER 08 - APPENDIX

Voice recognition source code

speech2text.py - C:\Users\SA Weerasekara\Desktop\Project WheelChair\untitled folder\speech2text.py (3.6.3)

File Edit Format Run Options Window Help

```
import speech_recognition as sr
import pyaudio
import serial

r = sr.Recognizer()

r.energy_threshold=3100
r.dynamic_energy_threshold= False


with sr.Microphone() as source:



      while True:
          print('Waiting :')
          ser=serial.Serial('/dev/ttyACM0', 9600)
          ser.write("8".encode())
          audio = r.listen(source)
          print('done')
          try:
              print('You Said :' + r.recognize_google(audio))
              if r.recognize_google(audio)==('forward'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
                    ser.write("7".encode())

              if r.recognize_google(audio)==('backward'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
                    ser.write("6".encode())

              if r.recognize_google(audio)==('left'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
                    ser.write("5".encode())

              if r.recognize_google(audio)==('right'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
```

Ln: 1 Col: 0

Figure 8.1

speech2text.py - C:\Users\SA Weerasekara\Desktop\Project WheelChair\untitled folder\speech2text.py (3.6.3)

File Edit Format Run Options Window Help

```
          ser=serial.Serial('/dev/ttyACM0', 9600)
          ser.write("8".encode())
          audio = r.listen(source)
          print('done')
          try:
              print('You Said :' + r.recognize_google(audio))
              if r.recognize_google(audio)==('forward'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
                    ser.write("7".encode())

              if r.recognize_google(audio)==('backward'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
                    ser.write("6".encode())

              if r.recognize_google(audio)==('left'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
                    ser.write("5".encode())

              if r.recognize_google(audio)==('right'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
                    ser.write("4".encode())

              if r.recognize_google(audio)==('stop'):
                    print('okay')
                    ser=serial.Serial('/dev/ttyACM0', 9600)
                    ser.write("3".encode())

              else:
                       ser=serial.Serial('/dev/ttyACM0', 9600)
                       ser.write("2".encode())


          except sr.UnknownValueError:
              print('Cant Identify the Voice')
          except sr.RequestError as e:
              print('could not request  results;  {0}'.format(e))
```

Ln: 1 Col: 0

Figure 8.2

Source code moving and obstacle detection

Figure 8.3

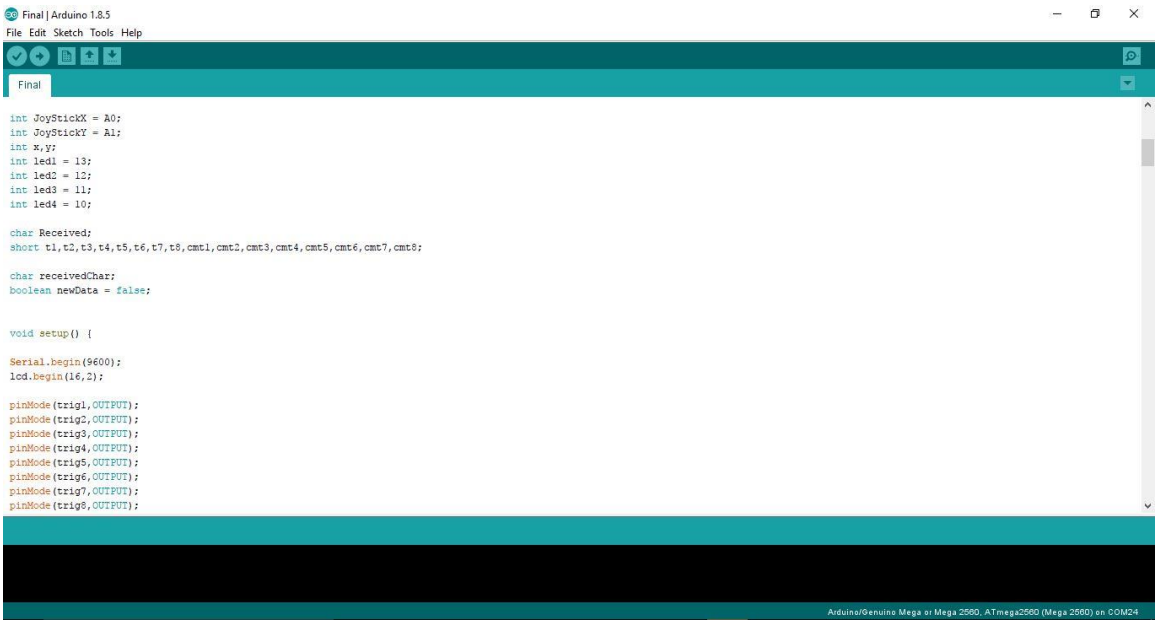

Figure 8.4

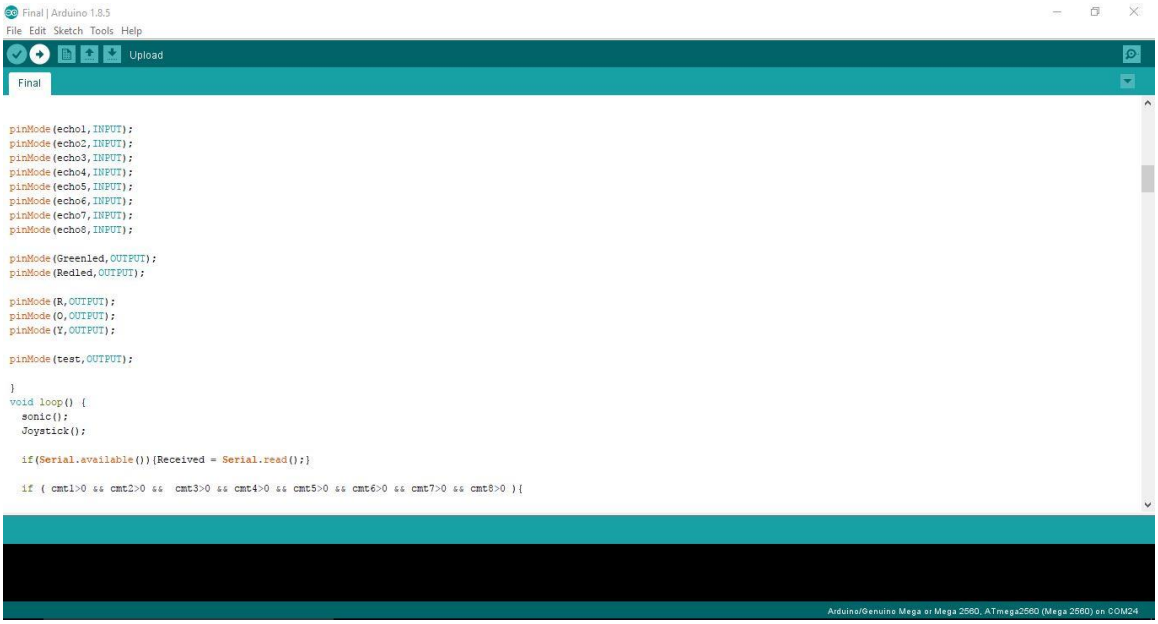

Figure 8.5

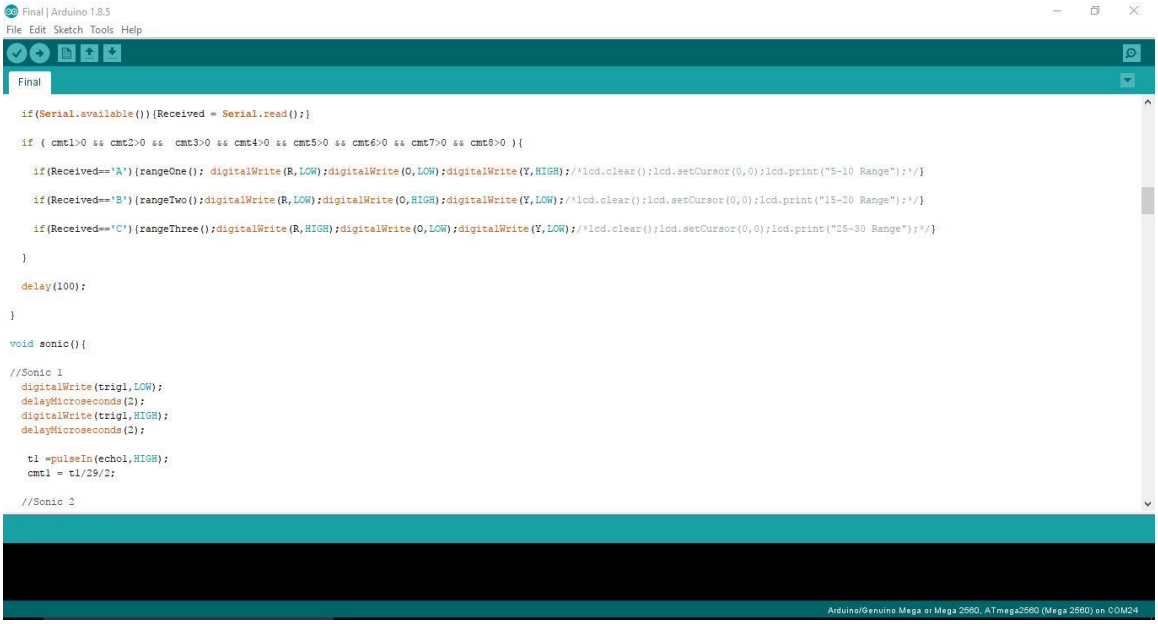

Figure 8.6

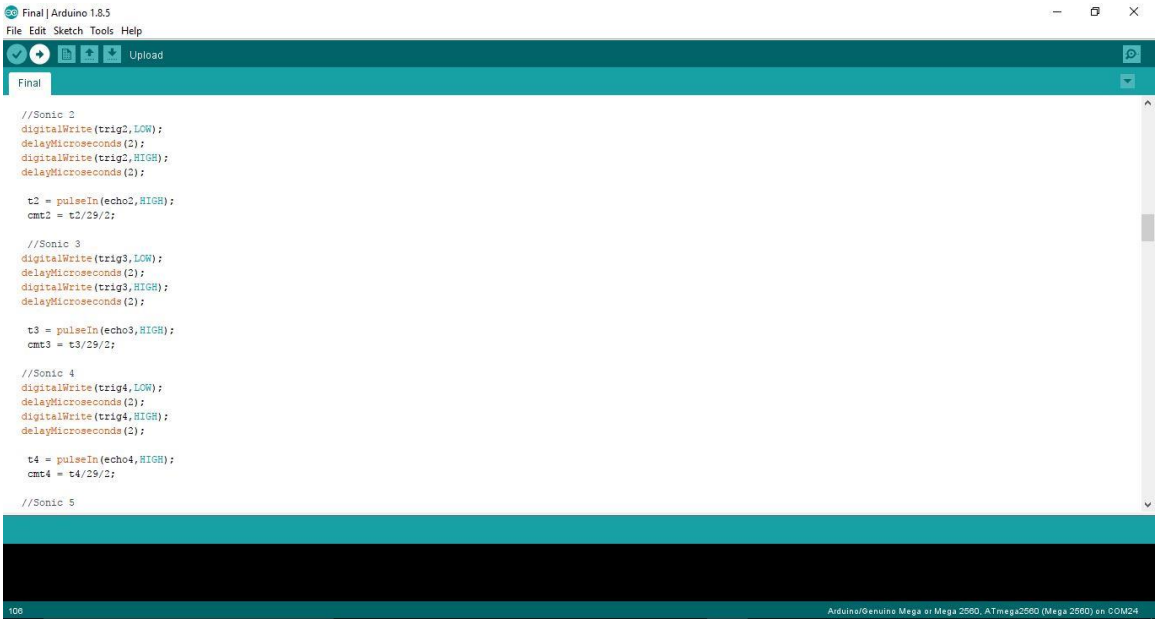

Figure 8.7

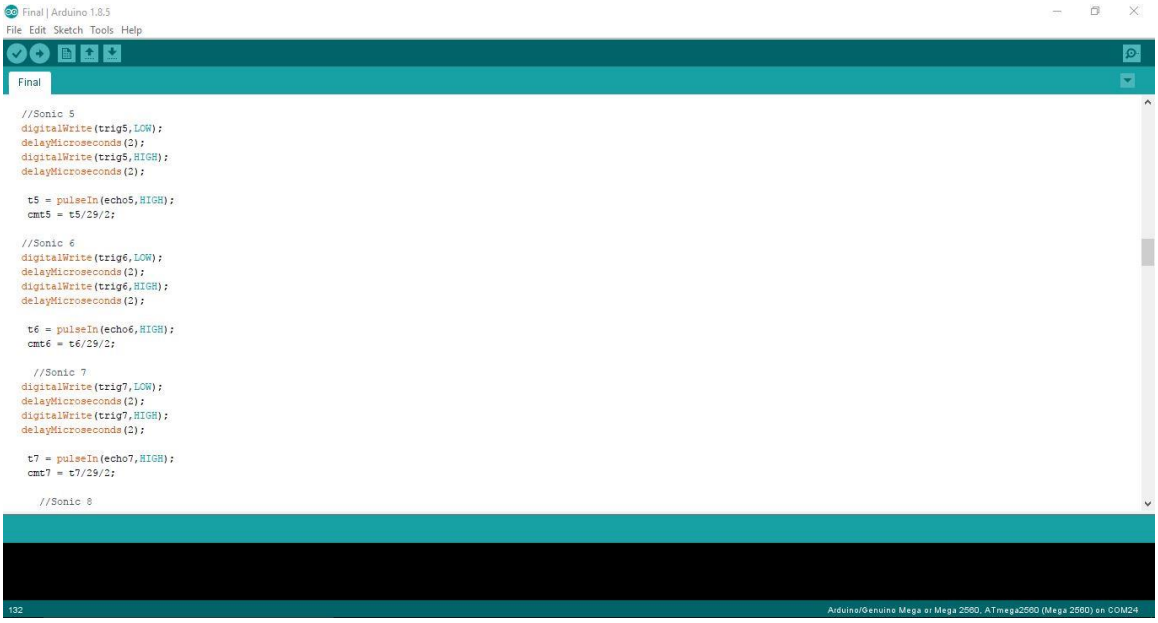

Figure 8.8

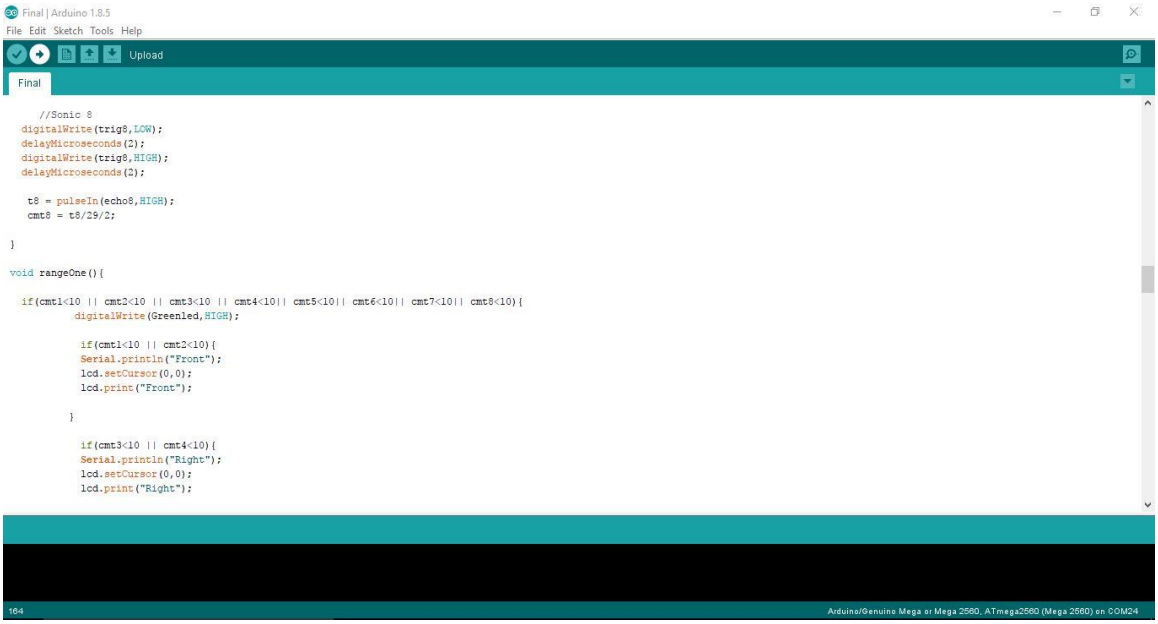

Figure 8.9

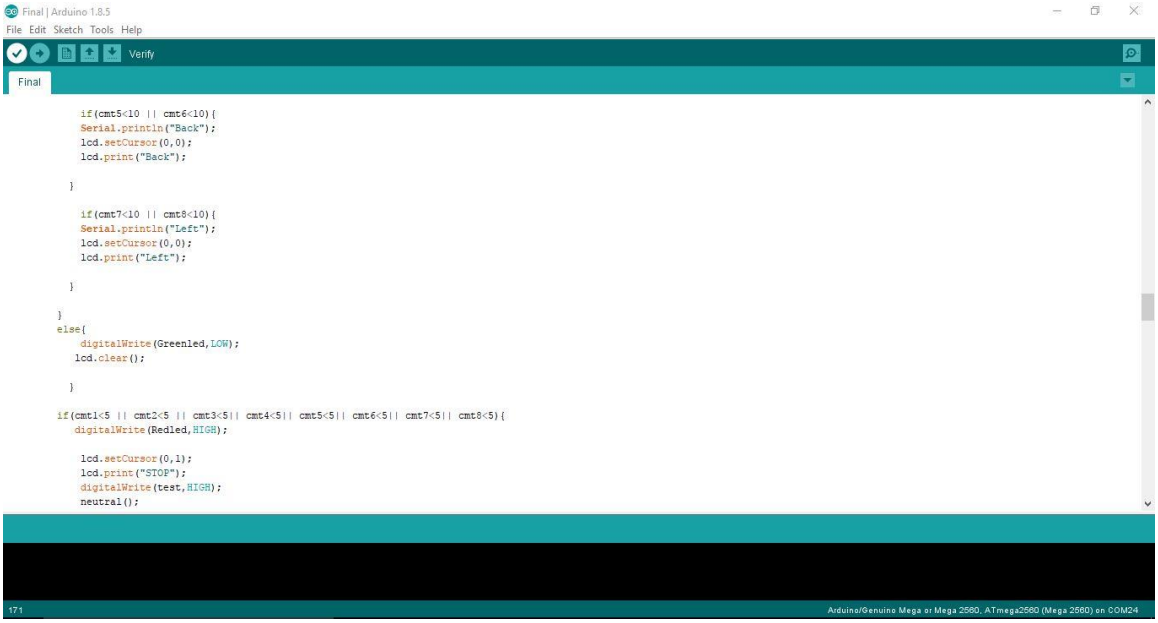

Figure 8.10

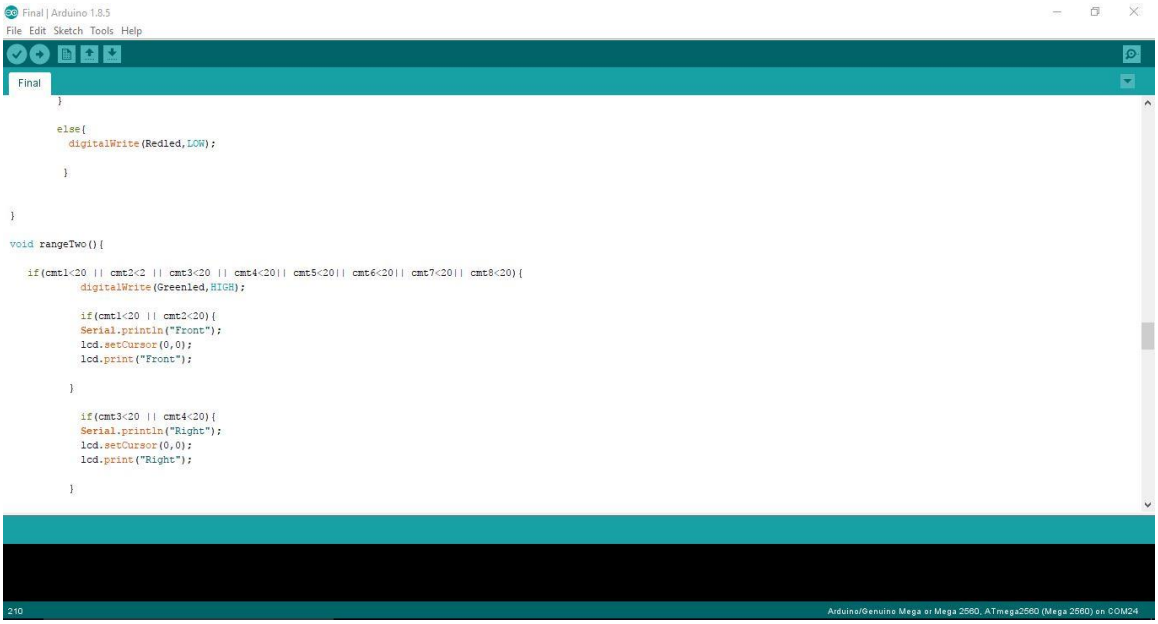

Figure 8.11

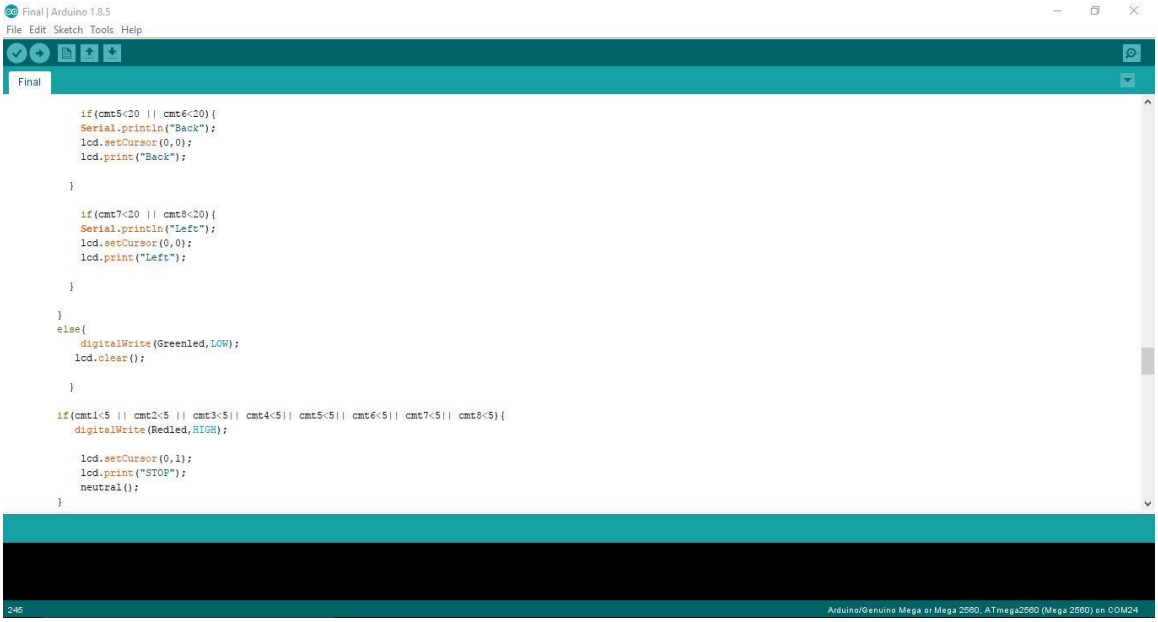

Figure 8.12

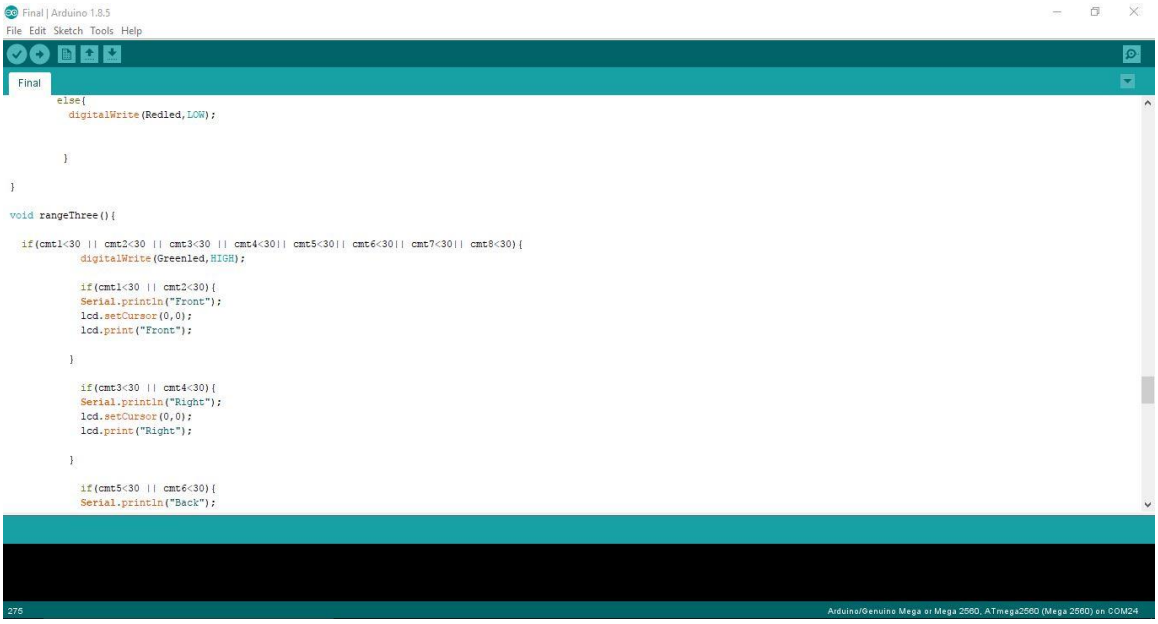

Figure 8.13

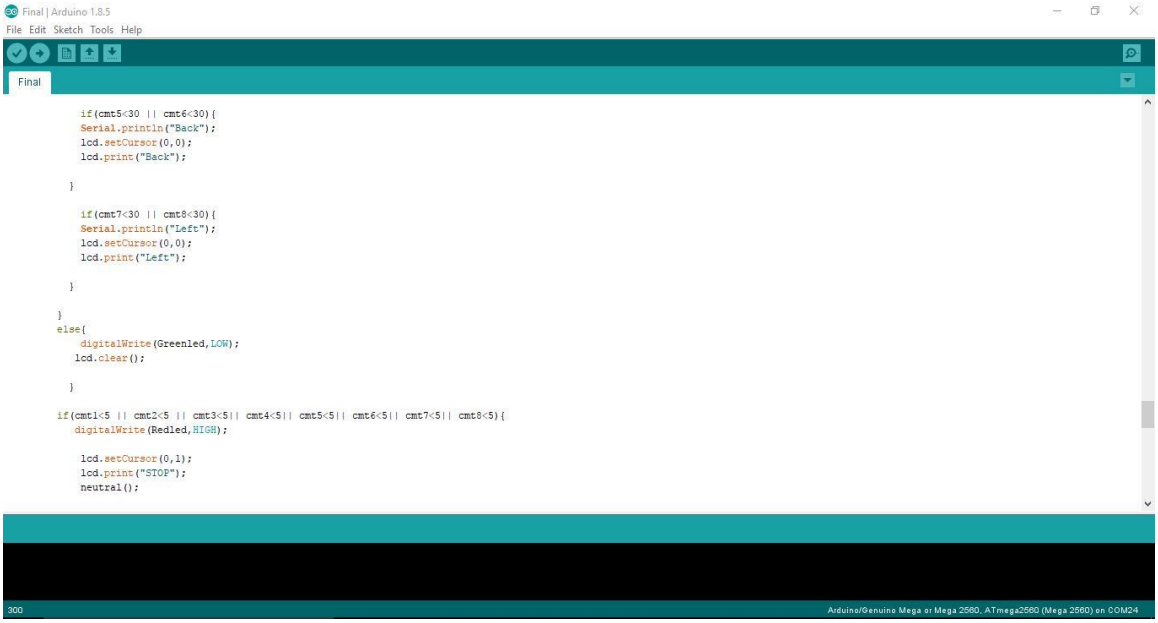

Figure 8.14

Figure 8.15

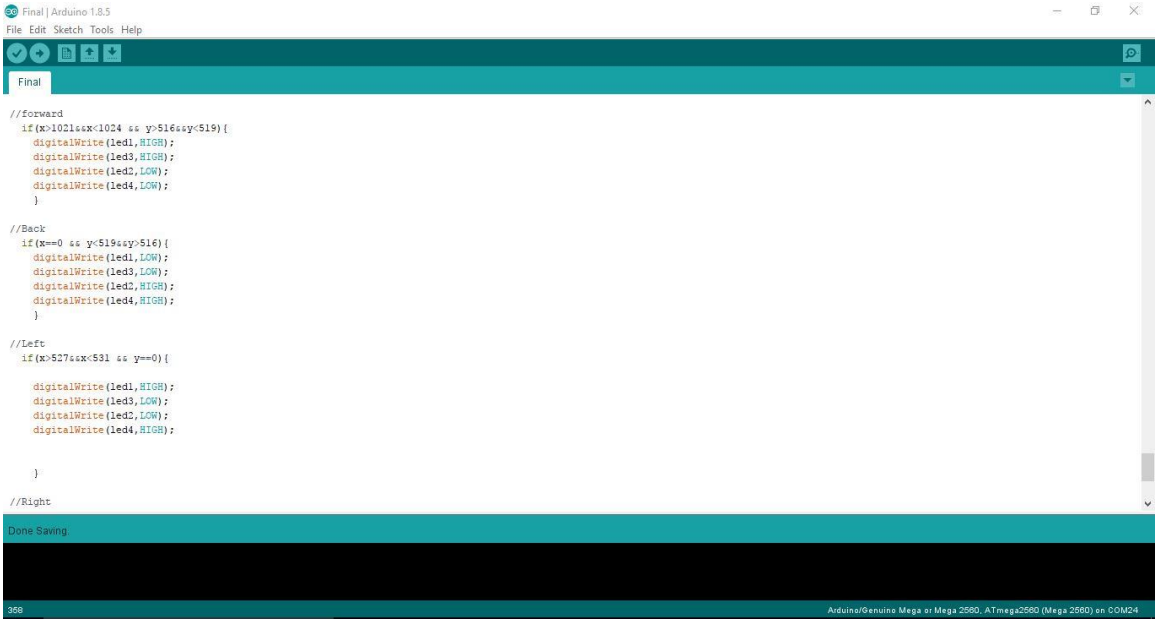

Figure 8.16

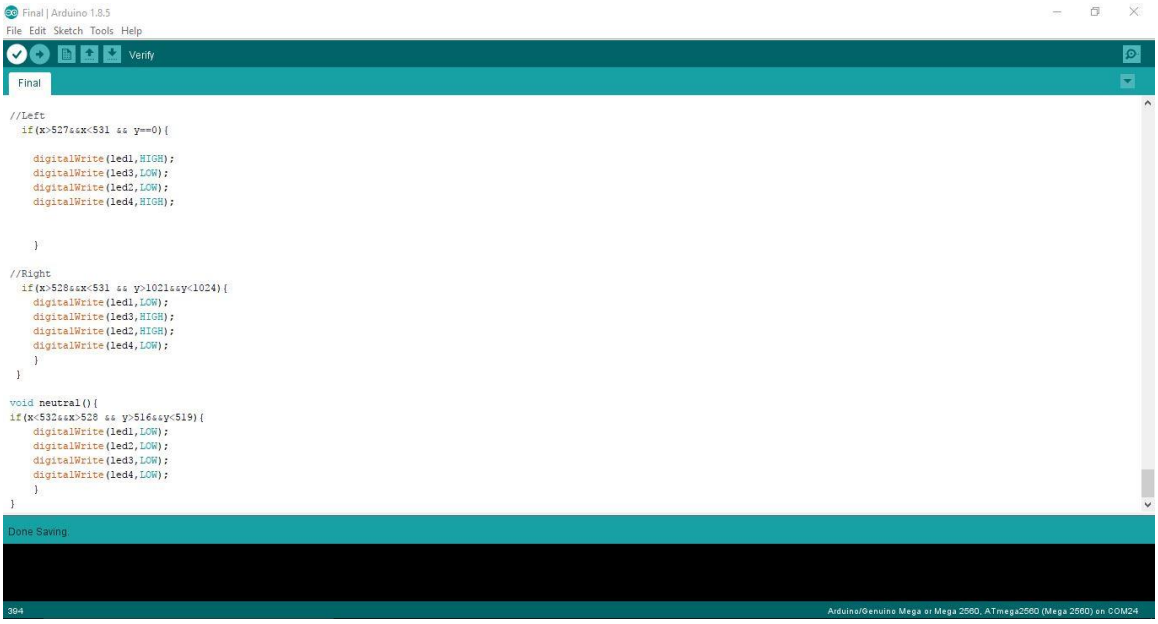

Figure 8.17

Source code for sending emergency message

Figure 8.18

Figure 8.19

Source code for android application

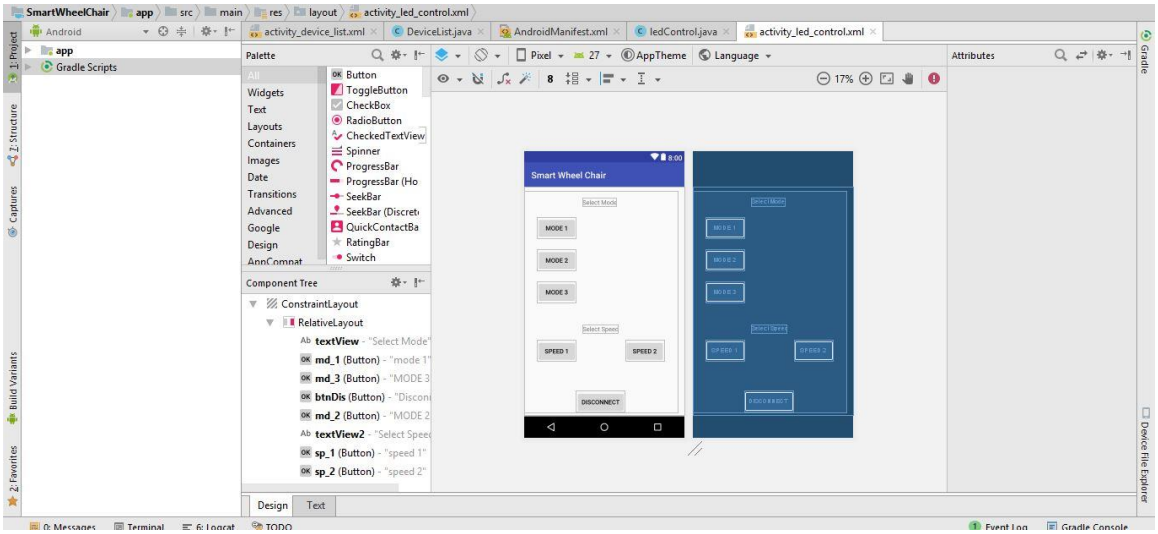

Figure 8.20

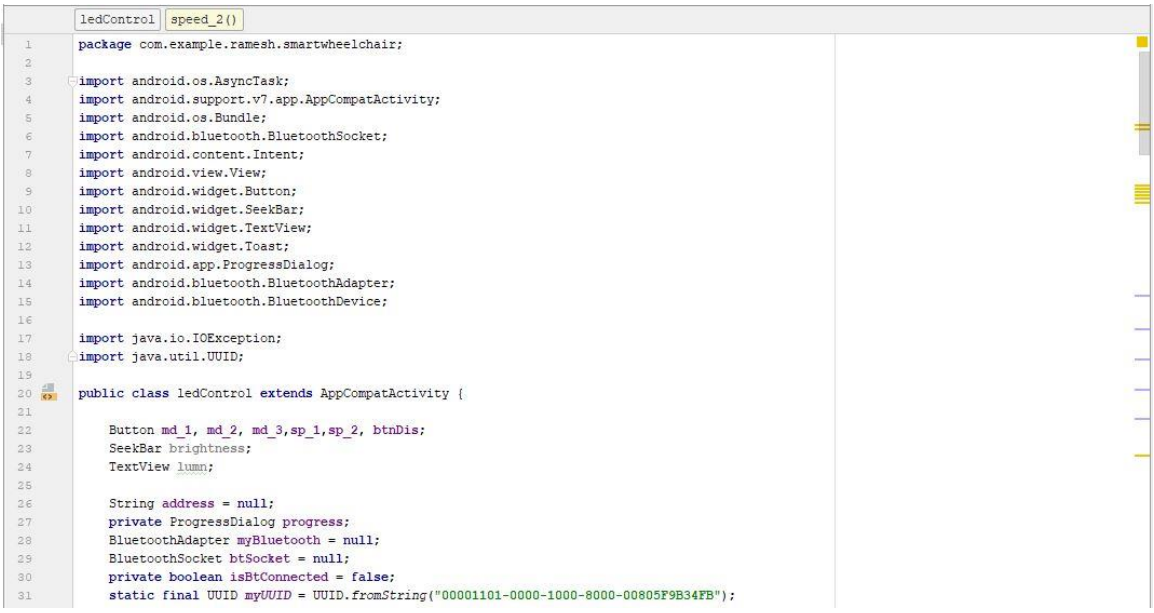

Figure 8.21

Figure 8.22

```
private void Disconnect() {
    if (btSocket != null) {
        try {
            btSocket.close();
        } catch (IOException e) {
            msg("Error");
        }
    }
    finish();

}

private void mode_1() {
    if (btSocket != null) {
        try {
            btSocket.getOutputStream().write("A".toString().getBytes());
        } catch (IOException e) {
            msg("Error");
        }
    }
}

private void mode_2() {
    if (btSocket != null) {
        try {
            btSocket.getOutputStream().write("B".toString().getBytes());
        } catch (IOException e) {
            msg("Error");
        }
    }
```

Figure 8.23

```
}
private void mode_3() {
    if (btSocket != null) {
        try {
            btSocket.getOutputStream().write("C".toString().getBytes());
        } catch (IOException e) {
            msg("Error");
        }
    }
}
private void speed_1() {
    if (btSocket != null) {
        try {
            btSocket.getOutputStream().write("X".toString().getBytes());
        } catch (IOException e) {
            msg("Error");
        }
    }
}
private void speed_2() {
    if (btSocket != null) {
        try {
            btSocket.getOutputStream().write("Y".toString().getBytes());
        } catch (IOException e) {
            msg("Error");
        }
    }
}

private void msg(String s) {
    Toast.makeText(getApplicationContext(), s, Toast.LENGTH_LONG).show();
```

Figure 8.24

```
    }

    private class ConnectBT extends AsyncTask<Void, Void, Void> {
        private boolean ConnectSuccess = true;

        @Override
        protected void onPreExecute() {
            progress = ProgressDialog.show( context: ledControl.this,  title: "Connecting...",  message: "Please wait!!!");

        }


        @Override
        protected Void doInBackground(Void... device) {
            try {
                if (btSocket == null || !isBtConnected) {
                    myBluetooth = BluetoothAdapter.getDefaultAdapter();
                    BluetoothDevice dispositivo = myBluetooth.getRemoteDevice(address);
                    btSocket = dispositivo.createInsecureRfcommSocketToServiceRecord(myUUID);
                    BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
                    btSocket.connect();
                }
            } catch (IOException e) {
                ConnectSuccess = false;
            }
            return null;
        }

        @Override
        protected void onPostExecute(Void result) {
```

Figure 8.25

```
        }

        @Override
        protected void onPostExecute(Void result) {
            super.onPostExecute(result);
            if (!ConnectSuccess) {
                msg("Connection Failed. Is it a SPP Bluetooth? Try again.");
                finish();
            } else {
                msg("Connected.");
                isBtConnected = true;
            }
            progress.dismiss();
        }
    }
}
```

Figure 8.26

9 References

[1] International Centre for Ethnic Studies. (2016, August 23). Equality for persons with disabilities [Online]. Available: http://ices.lk/wp-content/uploads/2016/08/ICES-Disability-Brief-29-Aug-16.pdf[

[2] B. Babu, V. Pillai, D. Shetty and S. Lohiya, Voice Based Enquiry System Using Raspberry Pi. (2016). International Journal of Innovative Research in Science, Engineering and Technology. [Online]. Available: https://www.ijirset.com/upload/2016/february/135_Voice.pdf

[3] Surbhi Verma. (2016, October 09). Android App Controlled Bluetooth Robot [Online]. Available: www.ijcaonline.org/archives/volume152/number9/verma-2016-ijca-911912.pdf

[4] Manuel Mazo Francisco, J. Rodríguez José, L. Lázaro Jesús UreñaJuan, C. García Enrique Santiso, Pedro Revenga, J. Jesús García (1995, September). Wheelchair for Physically Disabled People with Voice, Ultrasonic and Infrared Sensor Control. [Online]. Available: https://link.springer.com/article/10.1007/BF00710857

[5] Manoj V. Bramhe (2017 February, 2). Voice Controlled Wheelchair for Physically Disabled Person. [Online]. Available: https://www.ijareeie.com/upload/2017/february/55_Voice.pdf

[6] K.Sudheer (2012 November, 6). Voice and Gesture Based Electric - Powered Wheelchair Using ARM. [Online]. Available: www.ijrcct.org/index.php/ojs/article/viewFile/103/70

[7] Masato Nishimori, Takeshi Saitoh, Ryosuke Konishi (2007 September 20). Voice Controlled Intelligent Wheelchair. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/4421003

[8] Ali A. Abed (2015 December, 1). Design of Voice Controlled Smart Wheelchair. [Online]. Available: www.ijcaonline.org/research/volume131/number1/abed-2015-ijca-907235.pdf

[9] Hashimah Binti Ismail. (2006 April). Motion Control Using Voice for Wheelchair Application. [Online]. Available: www.eprints.utm.my/2320/1/HashimahIsmailMFKE2006.pdf

[10] Manuel Mazo,Frencisco. (1995 December). Wheelchair for Physically Disabled People with Voice, Ultrasonic and Infrared Sensor Control. [Online]. Available: https://www.robesafe.uah.es/index.php/en/publications?view=publication&task=show&id=869

[11] G Azam, M T Islam. (2006 December). Design and Fabrication of a Voice Controlled Wheelchair for Physically Disabled People. [Online]. Available: www.cuet.ac.bd/ICPSDT-15/files/Paper%2044

[12] Stefanie Tellex. (2005 May, 17). Relational Interface for a Voice Controlled Wheelchair. [Online]. Available: https://pdfs.semanticscholar.org/59f3/8192caedd5baabd57ee66539610ec84aefb6.pdf

[13] S D Suryawanshi, J S Chitode, S S Pethakar. (2013 May,5). Voice Operated Intelligent Wheelchair. [Online]. Available: www.srcpublications.com/admin/uploads/17416-050305.pdf