Brain Controlled Interface Based Automation Device

Submitted in partial fulfillment of

requirements For the degree of

Bachelor of Technology in Computer Engineering

by

Akshat Kacheria - 1511081

Vinit Todai - 1511116

Anil Guruprasad - 1512002

Guide

Prof. Mansi Kambli

Department of Computer Engineering

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Batch 2015 -2019

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **Brain Controlled Interface Based Automation Device** is bona fide record of the dissertation work done by

1. Akshat Kacheria
2. Vinit Todai
3. Anil Guruprasad

in the year 2018-19 under the guidance of **Prof. Mansi Kambli** of Department of Computer Engineering in partial fulfillment of requirement for the Bachelors in Technology degree in Computer Engineering of University of Mumbai.

Guide

Head of the Department

Principal

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **Brain Controlled Interface Based Automation Device** is bona fide record of project work done by

1. Akshat Kacheria
2. Vinit Todai
3. Anil Guruprasad

This project is approved for the award of Bachelors in Technology Degree in Computer Engineering of University of Mumbai.

Internal Examiner

External Examiner

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written thesis submission represents the work done based on my and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student **Roll No.**	**Signature of the Student** **Roll No.**
Signature of the Student **Roll No.**	**Signature of the Student** **Roll No.**

Date:

Place: Mumbai

Abstract

Various Motor Neuron Diseases such as Amyotrophic Lateral Sclerosis (ALS), brainstem stroke, cerebral palsy, muscular dystrophies, multiple sclerosis, and numerous other diseases impair the muscles. Many people around the world suffer through such diseases. Wheelchairs are mainly used by elderly or disabled people for their everyday movement. Such people find it extremely difficult to manoeuvre a mechanical wheelchair due to various physical constraints. Hence, there is a need to design a wheelchair to improve the quality of life and provide easy manoeuvrability for such people. Our body produces Biosignals which can be used as a source of information to trigger and control real-life applications. The proposed work is to develop a smart wheelchair that can assist the disabled people by providing them independence for basic mobility. The proposed system analyzes the brain wave signals, and uses Neurosky EEG headset. It will analyze the Eye Blinks, Attention Levels and Meditation Levels of the user. Every human has thousands of thoughts and emotions and is currently not possible to capture these thoughts. So we monitor waves from forehead frontal point (FP1). This Headset uses Electroencephalography (EEG) Technique to measure the Attention and Meditation Levels. It also uses the Electromyography (EMG) Technique to detect the muscle contractions while blinking the eyes. These waves are sensed by the EEG Headset and it will convert the raw data into packets and transmit through Bluetooth medium. Data Processing Module will receive the brain wave raw data and it will extract the relevant features and process the signal in the Android application.

Keywords:

People, Wheelchair, Headset, Disease, Electroencephalography, Level, Brain, Disable, Attention, Signals, Control, Raspberry Pi, Chassis, Threshold, Bluetooth, Movement, System, Module.

Contents

List of Figures

List of tables

Chapter 1

Introduction

This chapter sheds light on the introduction of the project. The brief idea about the project, the techniques and tools required to carry out the project. The scope, and the general outlook of the project is discussed here.

1.1 Background

In recent years, human to computer interfaces on speech and biometrics have been developed in order to initiate communication with computers. In most of the human to computer interfaces developed, the majority of them are aimed to provide people without disabilities with advanced techniques, meanwhile neglecting those individuals having severe disabilities. Various Motor Neuron Diseases such as Amyotrophic Lateral Sclerosis (ALS), brainstem stroke, cerebral palsy, muscular dystrophies, multiple sclerosis, and numerous other diseases impair the muscles. Many people around the world suffer through such diseases, they find it extremely difficult to manoeuvre a mechanical wheelchair due to various physical constraints. Hence, there is a need to design a wheelchair to improve the quality of life and provide easy manoeuvrability for such people. Our body produces Biosignals which can be used as a source of information to trigger and control real-life applications.

The emergence of the EEG Headset, which captures brain signals, has made this dream come close to reality. The brain waves are generated in response to external stimuli but the signal generated varies by a small microvolt from person to person. However, the only similarity is the section of the brain which responds to the stimuli.

The proposed work is to develop a smart brain controlled wheelchair that can assist the disabled people and provide them independence for basic mobility.

1.2 Motivation

Brain and spinal cord injury, Amyotrophic Lateral Sclerosis (ALS), brainstem stroke, cerebral palsy, muscular dystrophies, multiple sclerosis, and numerous other diseases impair the muscles. Approximately 1.9% of the total world population or roughly 5.6 million people encounter some difficulty or the other in physical movement. Some of those who have been severely affected lose complete muscle control.

There are thousands of people in India, who are facing disability in movement. They face discrimination on a regular basis which takes many forms. These people are considered more as a liability than an asset to the society. Since they encounter discrimination, they tend to alienate themselves from society as they feel unwanted and rejected. According to census 2001, in India, there were around 0.6% of the population had a disability in movements that is around 6 million people.

In addition, most of our public infrastructure, public transport and government buildings are not disabled-friendly. There has to be a concerted effort to ensure the disabled have access to these places and not feel hampered in any way.

Our mind-set has to change and accept these people and integrate them into our society. For this reason, we are building a special type of wheelchair which would be helpful to these people for being self-dependent and would be helpful to integrate them into our society.

A user-friendly system with a good user interface (UI) to address the issues of the severely disabled has not been developed yet. All of these brain waves are interdependent, as the transfer of the brain signals are carried by the neurons and every neuron sends the data to every other neuron in its neighbour and finally the shortest possible route to that part of the brain is decided and followed.

Continuing this research further, we are aiming to make commercially applicable Brain Control Devices such as a wheelchair for the disabled, by reducing the delay and adding an obstacle avoiding feature and Auto-Pilot feature to it..

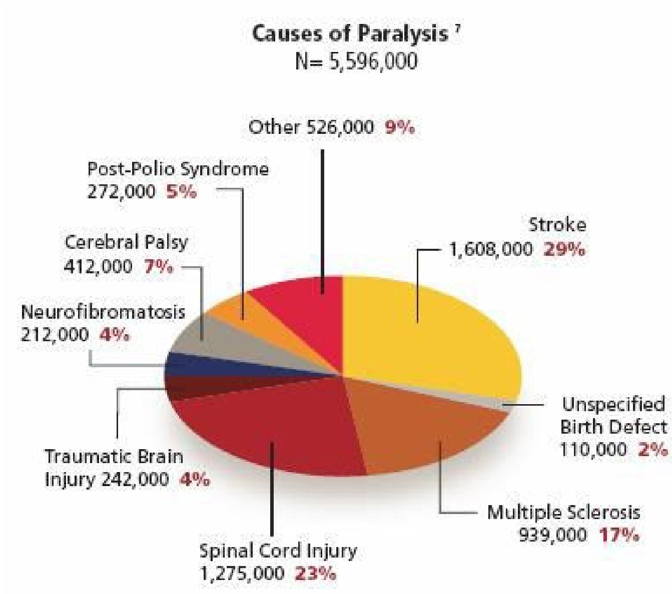


Figure 1.1

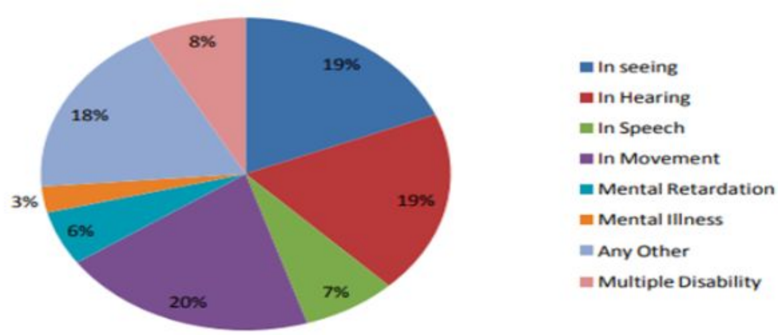


Figure 1.2

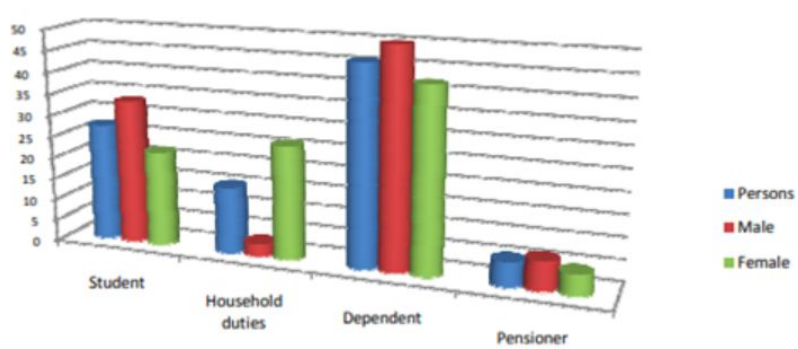


Figure 1.3

1.3 Problem Definition

Millions of people across the world suffer from various mobility impairments caused by the above mentioned diseases. A large portion of these people heavily rely upon wheelchairs to get on with their day to day activities. These people are dependent on others in order to move. A large number of patients are not prescribed wheelchairs at all, either because they find it physically difficult to control the manual wheelchair.

In our work with Brain Controlled Wheelchairs, we target a population who are or will become unable to use conventional interfaces, due to severe motor–disabilities. In order to assist this population with their mobility impairments and help them move in an autonomous way and also to lighten the duties of those who assist them, we propose to make a Brain Controlled Wheelchair.

1.4 Scope

The system will handle Incoming and outgoing data on serial Bluetooth port set up during simulation. The code will auto-detect communication ports. Data is pre-processed for noise internally from the headset. As fluctuation in EEG waves reduce we call it meditation or focus waves. The lower the amount of fluctuation, higher its focus value. Over multiple data collections we will set the meditation threshold. These waves are used to start a wheelchair. Once switched on, the subject must have a focus level greater than the threshold. When that is reached, the wheelchair stars. Every time we blink our eye, there is a drastic change in the wavelength. These particular moments are captured and sent as eye blinks. Which will be used for directional movement. An Android app will be created to display data values and feedback simultaneously also act as a connection interface for the system.

1.5 System Feasibility

Prior to carry out the whole project, a research regarding the project in terms of feasibility, in terms of operational and technical aspect of the project was created. The chances of an institution using our software for the purpose of examination, mainly comprises of two kinds of feasibility areas.

1.5.1 Operational Feasibility

The project would be a hardware based small scale prototype with a mobile application. With a detailed 3D design created for the actual model. The user will be able to perform the following:

- The user will be able to mind control small scale prototype after basic training.
- Physical efforts of the patient's caretaker will be significantly reduced.
- The project will be user friendly, intuitive and easy to operate.
- It is an operational feasible project in terms of software and hardware specification.
- The project is designed to be the cheapest in its segment.
- Sound and lights can also be controlled using attention levels

1.5.2 Technical Feasibility

- The system will be scalable to a full size model as demonstrated with a 3D design.
- The technology, which is required to carry out the project or any scope of improvement, is available.
- EEG technology is undergoing extensive research and our project will be a part of it.
- Microcontrollers and other open source softwares have been used to create project and its technical design

1.6 System Requirements

1.6.1 Hardware

(For Prototype)

1. Brainwave EEG headset.
2. Raspberry Pi 3B+
3. Android Phone
4. Ultrasonic Sensor (HC-SR04)
5. Rpi Camera
6. Jumper Wires

(For full scaled model)

7. Custom gear assembly
8. Wheelchair
9. Etonm ET-PGM42 Motor
10. 12-24V DC custom battery
11. Metal cover boxes for battery, chips and wires.

1.6.2 Software

1. Android Studio
2. Python
3. Solidworks CAD

1.7 Assumptions

The following assumptions were made during the development of our project:

- Project members will discuss the aspects and progress of the project on a regular basis.
- Project members will meet with the project mentor time to time.
- The programming languages required to carry out the project is known beforehand.
- Third party hardware used to develop the project is verified and error free.
- The time to completion of the project within the promised deadline.
- The progress report will be shown when needed.

Following are the constraints that can be encountered during the development of our project:

- The project must be implemented and fully deployed within the deadline of the project.
- The proposed features and specification of the project must be delivered.
- Software and prototype provide correct results.
- The project has been tested for desired features.
- All work done for the project is performed by the team members.

1.8 Deliverables

The project was considered complete when the following deliverables were achieved.

A. Literature survey-

Study of the existing system, pros and cons of the current system as well as analysis of how the new proposed system can overcome the challenges of existing system in terms of functionalities.

B. Documentation-

Software Project Management Plan, Software Requirement Specification Document, All applicable diagrams.

C. Implementation-

Final design and code for the working prototype of our project.

D. Prototype Models-

The prototype of a brain controlled automation device with an android app and the 3D design of the full scale model

E. Deployment and Maintenance

After project has been completed, the app is installed on the phone, connected with the hardware prototype and bluetooth headset. The prototype is then operational and maintained.

1.9 Stakeholders

Stakeholders for the project include:

- Decision-makers: People with a higher authority, who control the flow of the project, in terms of functionality, requirements. In this case, the Principal, Head of Department and Project Mentor and team members.
- Influencers: People who may influence the decision of the project, only by a certain margin. In this case, it would be the team members.
- End users: The final user of the software, who will use the project, the design has to be done accordingly. In this case a disabled patient.

1.10 Outcomes

The project will be considered to be a success if

1. Robotic prototype of a wheelchair can be controlled using attention values and eye blinks.
2. Smart emergency braking system with ultrasonic sensors is successfully used.
3. Feedback is correctly displayed.
4. Valid 3D design for full scale model is created.

1.11 Risks

1. Management of time - Managing the time for project was one of the major challenges faced.
2. Project integration - The project is an integration of various electronic, mechanical and software technologies which posed as a challenge that was overcome in the past one year.
3. Emerging technology - Brain controlled devices are still in research phase hence creating a challenge to use precedents or finding third party help.

Chapter 2

Literature Survey

This chapter discusses about the various papers, articles referenced in preparation for undertaking this project. These items serve as a benchmark to enable this project to be undertaken.

Brain computer interface (BCI) are systems that communicate between human brain and physical devices by translating different patterns of the brain activity into commands in real time [2]. The electrical activity of the brain is monitored in real time using an array of electrodes, which are placed on the scalp in a process known as electroencephalography (EEG) [1]. Traditional EEG sensors are expensive and their use is limited only to hospitals and laboratories. The electrodes of EEG sensors require conductive gel on skin in order to facilitate reading signals.[2] The advantage of using a portable EEG brainwave headset is that it uses a dry active sensor technology to read brain electric activity.

2.1 EEG brain signals:

The EEG signal is typically described in terms of rhythmic activity and transients. The rhythmic activity is divided waves into bands by frequency while the transient is referring to spike and sharp waves. To some degree of rhythmic activity, these frequency bands are a matter of classification but these designations occur because rhythmic activity within a certain frequency range was noted to have a certain distribution over the scalp or a certain biological significance.

There are five types mostly important.

2.2.1 Delta

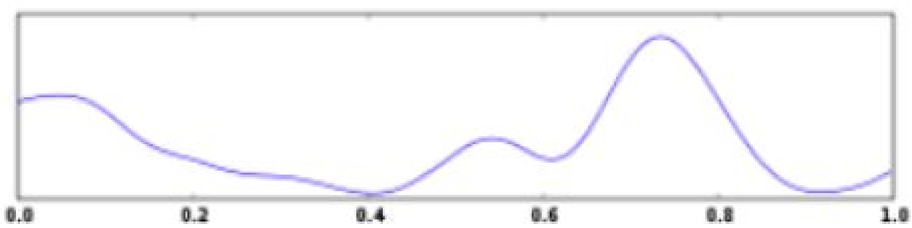


Figure 2.1

Delta waves lie within the range of 0.5 to 4 Hz, with variable amplitudes. It tends to be the highest in amplitude and the slowest waves. Delta waves are generally associated with slow wave sleep (during stages 3 and 4 of the stage of sleep). These brain wave is primarily associated with deep sleep, and in the waking state, were thought to indicate physical defects in the brain.

It is usually most prominent frontally in adults (e.g. FIRDA - Frontal Intermittent Rhythmic Delta) and posterior in children (e.g. OIRDA - Occipital Intermittent Rhythmic Delta).

2.2.2 Theta

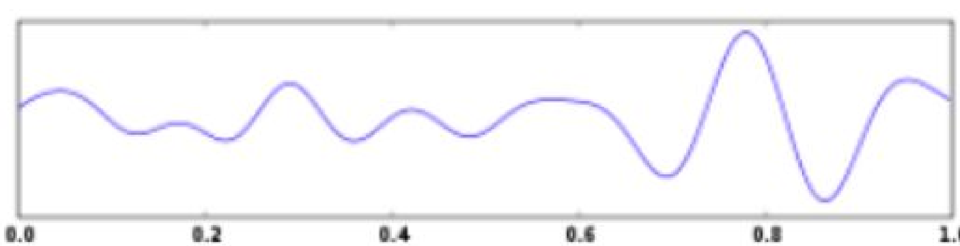


Figure 2.2

Theta waves lie within the range of 4 to 8 Hz, with an amplitude usually greater than 20μV. Theta arises from emotional stress, especially frustration or disappointment. Theta has been also associated with access to unconscious material, creative inspiration and deep meditation. The large dominant peak of the theta waves is around 7 Hz.

2.2.3 Alpha

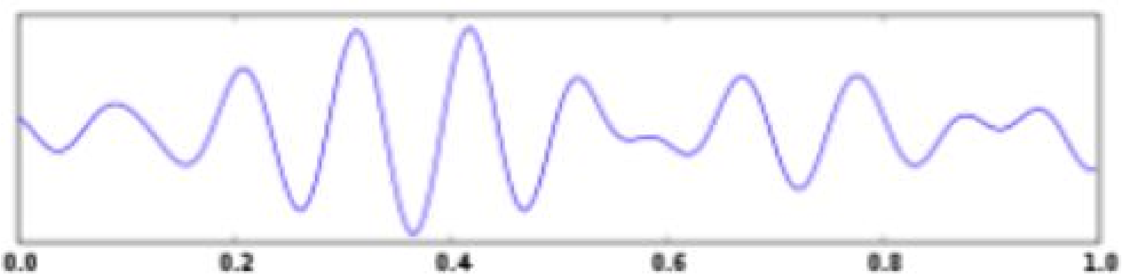


Figure 2.3

The rate of change lies between 8 and 13 Hz, with 30-50μV amplitude. Alpha waves have been thought to indicate both a relaxed awareness and also inattention. Alpha alone seems to indicate an empty mind rather than a relaxed one, a mindless state rather than a passive one, and can be reduced or eliminated by opening the eyes, by hearing unfamiliar sounds, or by anxiety or mental concentration. They are strongest over the occipital (back of the head) cortex and also over frontal cortex.

Alpha is the most prominent wave in the whole realm of brain activity and possibly covers a greater range than has been previously thought of.

It is frequent to see a peak in the beta range as high as 20 Hz, which has the characteristics of an alpha state rather than a beta, and the setting in which such a response appears also leads to the same conclusion.

2.2.4 Beta

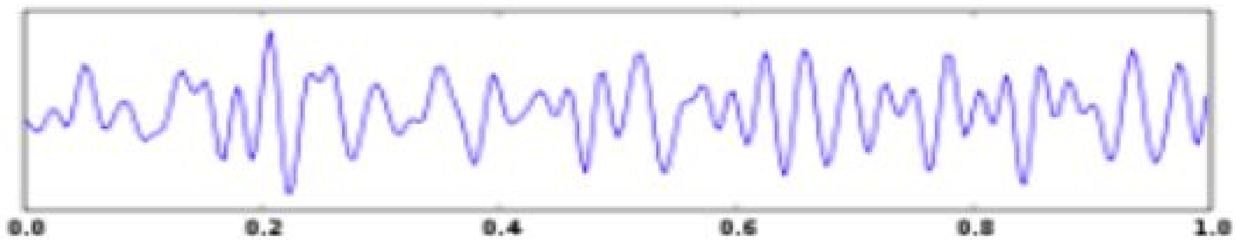


Figure 2.4

The rate of change lies between 13 and 30 Hz, and usually has a low voltage between 5-30µV. Beta activity is closely linked to motor behavior and is generally attenuated during active movements like active thinking, active attention, and focus on the outside world or solving concrete problems. It can reach frequencies near 50 Hz during intense mental activity. Rhythmic beta with a dominant set of frequencies is associated with various pathologies and drug effects, especially benzodiazepines.

It may be absent or reduced in areas of cortical damage. It is the dominant rhythm in patients who are alert or anxious or who have their eyes open.

2.2.5 Gamma

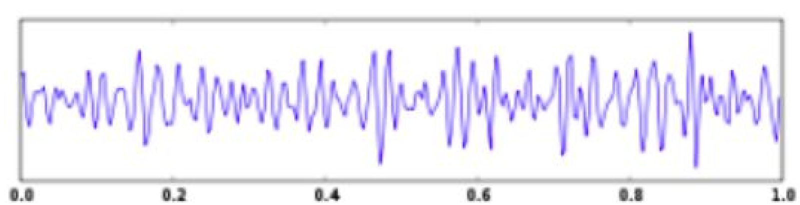


Figure 2.5

Gamma waves lie within the range of 35Hz and above. It is thought that this band reflects the mechanism of consciousness - represent the binding of different populations of neurons together into a network for the purpose of carrying out a certain cognitive or motor function. (Feeding back on themselves over time to create a sense of stream-of-consciousness)

The advantage of using a portable EEG brainwave headset is that it uses a dry electrode sensor technology to read brainwaves activity. Traditional gel based EEGs can take up to 20 minutes in order to start data acquisition while the Neurosky EEG headsets are ready instantly.
So, Neurosky Headset is cost-effective and easy to handle. The onboard Think Gear IC processes raw signals filters noise and makes digital signal [2-5]. The control system design for the wheelchair using various methods of BCI as well as speech and gesture recognition are discussed [2- 8], but the actual implementations are not shown.

The multipurpose manual wheelchair is designed to serve the various purpose of the patient as well as elderly people [9]. A wheelchair is designed which is controlled through Electrooculography. The wheelchair movement is restricted to a particular direction based on the eye's horizontal and vertical movements. But practically, the eye will also have some oblique movements for which the wheelchair is not satisfactorily responding for the movement in a particular direction [10].

The proposed work deals with the engineering an interface between the human brain and an electric wheelchair using a Low cost EEG headset and digital signal processing. The project eliminates the drawbacks of conventional EEG by using a dry sensor technology to pick up EEG signals instead of using a conductive gel and reducing the time it takes to set up.

This project aims at creating a cost efficient solution, later intended to be distributed as an add-on conversion unit for a normal wheelchair. Doing so would be of Nobel importance to 'brain-active-body-paralyzed' patients providing them the independence of mobility.

2.3 Research Gap:

The research gap of this project compared to the previous versions [1-10] are:

- The reduction in cost by making the design as a conversion kit for a regular wheelchair; the project doesn't reinvent the wheel, it instead builds on top of an existing framework and brings together the best of things. The mechanical modifications are narrowed down to a level where it can be reproduced easily and put on other manual wheelchairs.
- The project uses a mindwave headset instead of traditional EEG to acquire brain signals thereby reducing the setup time.
- Future work on the app allows the wheelchair's control to be either mind wave controlled or joystick controlled or combine both for efficiency and patient condition feedback; the two modes of operation makes it possible for this system to be useful for a wide range of disabilities in people.

Chapter 3

Software Project Management Plan

This document is the fully developed Software Project Management Plan for "Brain Controlled Interface Based Automation Device." This document highlights the deliverables, roles, task and schedule for the team/project.

3.1 Introduction

3.1.1 Purpose of Plan

The project plan for the project titled "Brain Controlled Smart Automation Device" will provide a definition of the project, its goals, objectives and will also provide other useful information related to the project.

The Project plan defines the following:-

- Project purpose
- Project's goals and objectives
- Scope and expectations
- Roles and Responsibilities
- Assumptions and constraints
- Project management approach
- Ground rules for the project
- Project budget
- Project timeline

3.1.2Project approach

Table 3.1

Project Plan 2018-2019	Akshat	Vinit	Anil
August - September	Literature Review, requirement analysis and scope development		
October	**IICDC, NES Innovation Awards, eYIC(IIT- B)** presentations**,** submissions and documentation		
November	Prototype Designing, Data acquisition using Matlab. Data extraction testing and feedback demonstration	System Flow, diagrams, system modeling, Android Code Development, Python code development. Shifting from matlab to android sdk	Hardware modeling, Webench completion, Electronic interfacing and chassis building
December	Simulating System and testing thresholds. Feature classification and algorithm development for eye blink patters with attention for the movement of the prototype.		
January – February	Final development with classification patterns and thresholds setting. Preparing cost analysis and bplan. Full scale prototype custom 3D modeling, power train calculation	Integrating Software with hardware using rfcomm protocols on python. Creating final android user interface. Integration of sounds and light control using eeg for android feedback	Setting up system on Raspberry pi. Establishing connections and electronic circuits for the prototype. Integration of ultrasonic based emergency braking system
March	**EYIC, IICDC, NES, Maker Mela, Ennovate** Semi Final/Finals preparation and demonstration		
April	Testing and monitoring for bugs before final project submission		

3.2 Project's goals and objectives

1. Allows users to control their prototype wheelchair with attention and eye blink.
2. Provide feedback mechanism.
3. Provide android application for feedback and connection.
4. Provide 3D design for full scale model

3.3 Scope of the project

- The system will handle Incoming and outgoing data on serial Bluetooth port set up during simulation. The code will auto-detect communication ports. Data is pre-processed for noise internally from the headset.
- As fluctuation in EEG waves reduce we call it meditation or focus waves. The lower the amount of fluctuation, higher its focus value. Over multiple data collections system will the meditation threshold. These waves are used to start a wheelchair. Once switched on, the subject must have a focus level greater than the threshold. When that is reached, the wheelchair starts.
- Every time we blink our eye, there is a drastic change in the wavelength. These particular moments are captured and sent as eye blinks. Which will be used for directional movement.
- An Android app will be created to display data values and feedback simultaneously also act as a connection interface for the system.

3.3.1 Projected budget

Table 3.2

Raw Material Costs	Variable Costs (per unit)
Wheelchair= ₹ 10,000	Transportation= ₹ 500
Headset= ₹ 8,000	Maintenance= ₹ 1,000
Microcontroller= ₹ 1,000	Labor Charge= ₹ 3,500
DC motors + Battery= ₹ 7,500	Packaging= ₹ 500
Motor Driver= ₹ 1,000	
Total= ₹ 27,500	Total= ₹ 5,500

3.4 Milestones

Table 3.3

Milestone	Estimated Date of Completion
Literature review, requirements and scope	01/08/2018
Rough first version of prototype	01/10/2018
Setting thresholds and feature classification	31/12/2018
3D design, documentation and final prototype version	31/02/2019
Testing and completion	01/04/2019

3.5 Project assumptions

- Project team members will adhere to the project plan.
- Project members will work collaboratively and encourage ideas from every member.
- The workplace will contain the necessary infrastructure required for the completion of this project.
- Any changes to the scope of the project will result in changes to the project plan.
- Any changes should the plan is agreed upon by all the users and any dispute shall be handled by the project manager.

3.6 Constraints

The following are the constraints on this specific project:-

- The budget of the project is limited.
- The available workforce is restricted to 3. (The number of project participants.)

3.7 Project roles and responsibilities

The following **table 3.4** shows the different roles and responsibilities performed by the team members for this project.

Akshat	Vinit	Anil
Literature Review, requirement analysis and scope development		
IICDC, NES Innovation Awards, eYIC(IIT- B) presentations, submissions and documentation		
Prototype Designing, Data acquisition using Matlab. Data extraction testing and feedback demonstration	System Flow, diagrams, system modeling, Android Code Development, Python code development. Shifting from matlab to android sdk	Hardware modeling, Webench completion, Electronic interfacing and chassis building
Simulating System and testing thresholds. Feature classification and algorithm development for eye blink patters with attention for the movement of the prototype.		
Final development with classification patterns and thresholds setting. Preparing cost analysis and bplan. Full scale prototype custom 3D modeling, power train calculation	Integrating Software with hardware using rfcomm protocols on python. Creating final android user interface. Integration of sounds and light control using eeg for android feedback	Setting up system on Raspberry pi. Establishing connections and electronic circuits for the prototype. Integration of ultrasonic based emergency braking system
EYIC, IICDC, NES, Maker Mela, Ennovate Semi Final/Finals preparation and demonstration		
Testing and monitoring for bugs before final project submission		

3.8 Issue Management

In case of any issues within the project, the issue needs to be resolved seamlessly after having considered every person's viewpoint. The following procedure would ensure the same.

1. An issue or a change is pointed out by a team member and brought to the notice of the other team members.
2. The entire team reviews the change/ issue and also conducts discussions regarding the same.
3. The impact and likelihood of the issue is discussed. The issue is then discussed with project manager.
4. If found to be a high impact issue, the process to overcome the issue is undertaken by the project manager.

Chapter 4

Software Requirement Specification Document

In this chapter, the different requirements of the system are discussed.

4.1 Introduction

The SRS document of the project consists of the various requirements of the project such as the external interface requirements, hardware requirements as well as the communication protocols that have been used in the project.

The target audience of this project includes the disabled patients, doctors and medical researchers.

4.2 Specific requirements

Following are the basic software and hardware requirements for the project:-

4.2.1 External interface

Requirements User Interface:-

- An interface that can be used to connect to the wheelchair and headset.
- An interface that is easy to use and keeps the user at the centre.
- An interface that allows the user to view feedback with attention and blink strength on the android application.
- Interface to show what is the current state of wheelchair movement.
- Interface to assist and inform the patient when attention is less than required.

Hardware Interface

- An android phone to install and run application
- Wheelchair prototype with castor wheel and motors
- Object detection using ultrasonic sensors
- Wheelchair must move by attention values of users and turn by eye blink strength

4.2.2 Communication Protocol

Protocols being used is the rfcomm protocol which uses bluetooth connections. It is used for communication between the Rpi 3B+ and the neurosky EEG headset with phone.

4.3 Software system attributes

Following are the software system attributes:-

1. **Reliability**

   The system is quite reliable in the sense that only a few undetected eye blinks were encountered during the testing phase. There is also obstacle detection for emergency braking.

2. **Availability**

   The system is available 24*7, irrespective of the user's network connection status. He can control the wheelchair even when he isn't on any external network and even without internet connection as long as batteries are charged.

3. **Security**

   The system is quite secure. It is disconnected from mobile networks. It has emergency braking which works with negligible response time.

4. **Portability**

   Portable to every place that has facilities for manual wheelchair and disabled friendly ramps.

5. **Performance**

   The system works with satisfying efficiency. There is very little lag and almost no communication delay between headset, wheelchair and phone.

Chapter 5

Software Design Development

This chapter will describe the architectural design from different perspectives with the use of various diagrammatic and graphical representations to develop a Brain Controlled Interface Based Automation Device.

5.1 <u>Introduction</u>

This section will provide a basic description of the system design which prepares the reader to understand the further sections easily.

5.1.1Design Overview

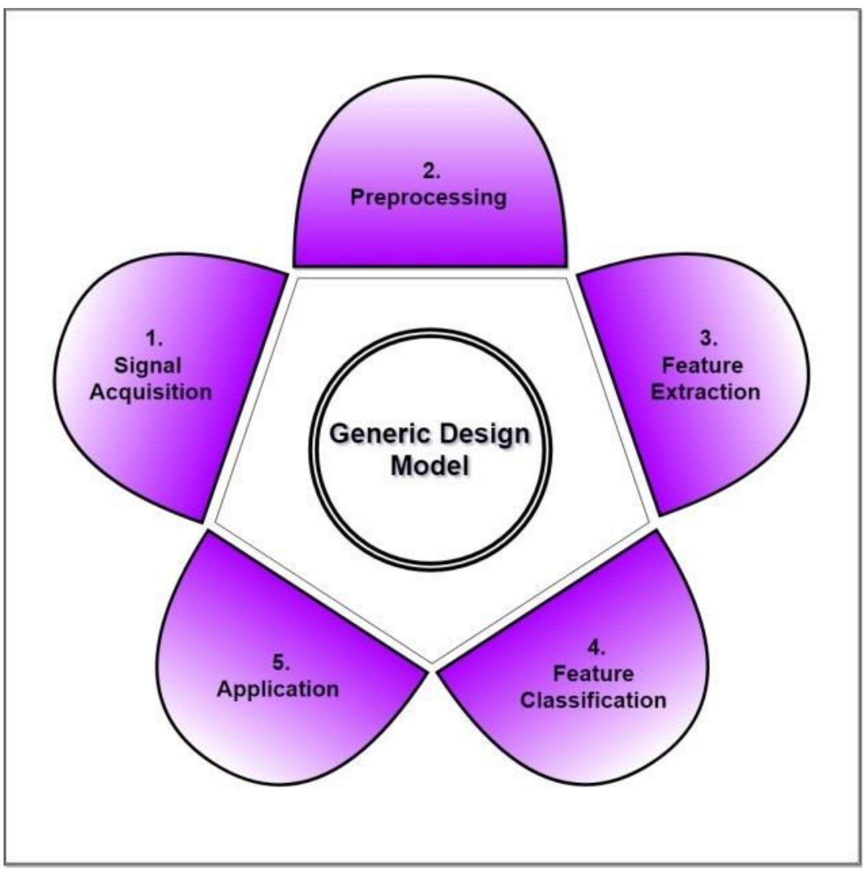


Figure 5.1

1) Signal Acquisition Module

The NeuroSky Mindwave EEG headset is used to capture the EEG signals.

2) Data Pre - Processing Module

The noise produced in the system can be removed by performing FFT analysis. The(Attention and blinking signal levels) are differentiated by using the Level analyzer Technique.

3) Feature Extraction Module

Here, the main features corresponding to the waves are extracted and then sent to the Classification module to classify it into 3 actions, i.e. Front, Turn, Stop

4) Feature Classification Module

The classification of waves can be done using Data Analytics and setting thresholds, the output signals are transmitted to the automation device using Internet Of Things.

5) Hardware Module

It consists of a microcontroller to control the movement of the wheelchair and an Obstacle Avoidance System

Example: Robotic Wheelchair Module/ Smart home lights

5.2 Detailed Description of Components

5.2.1User Interfaces

Our system provides an Android UI. The android UI provides a set of controls, covering different types of users and hence providing ease of usability

5.2.1.1 Android UI

The Android app for the system is developed. The Android app is used to set up the connection between the EEG headset and the raspberry pi controlled robotic prototype.The signal processing takes place in the Android app, the input EEG signal from the user is matched with the predefined threshold values in order to generate thought commands to control the robotic prototype.The Android application uses Bluetooth to connect to the EEG headset as well as the raspberry pi controlled wheelchair.

5.3.2RaspberryPi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. The Raspberry pi is used to control the movement of the robotic prototype in response of the thought commands received from the mobile application via bluetooth.

5.3.3Neurosky Mindwave EEG headset

1. The MindWave Mobile 2 safely measures and outputs the EEG power spectrums (alpha waves, beta waves, etc), NeuroSky eSense meters (attention and meditation) and eye blinks. The device consists of a headset, an ear-clip, and a sensor arm. The headset's reference and ground electrodes are on the ear clip and the EEG electrode is on the sensor arm, resting on the forehead above the eye (FP1 position). It uses a single AAA battery with 8 hours of battery life. It measure the following:-
2. 12 bit Raw-Brainwaves (3 - 100Hz) with Sampling rate at 512Hz
3. Outputs EEG power spectrums (Alpha, Beta, etc.)
4. Outputs NeuroSky proprietary eSense meter such as Attention Meditation, and other future meters
5. EEG/ECG signal quality analysis (can be used to detect poor contact and whether the device is off the head)

5.3.4 Motor Driver L293D

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

5.3.5 Ultrasonic Sensor

Ultrasonic sensors measure distance by using ultrasonic waves.
The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.An optical sensor has a transmitter and receiver, whereas an ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturization of the sensor head.
The distance can be calculated with the following formula:

$$\text{Distance } L = 1/2 \times T \times C$$

where L is the distance, T is the time between the emission and reception, and C is the sonic speed. (The value is multiplied by 1/2 because T is the time for go-and-return distance.)

5.4 Diagrams and figures

5.4.1Block Diagram

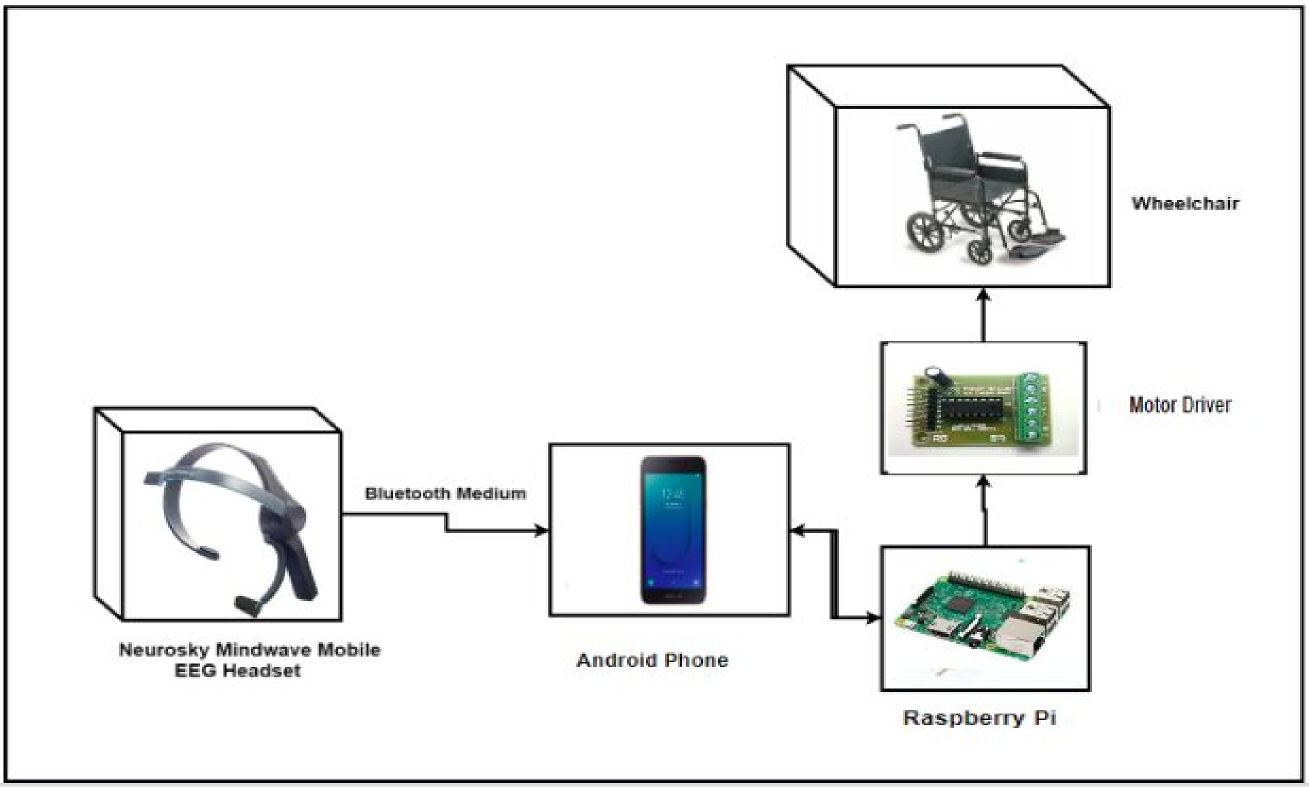


Figure 5.2

The EEG headset provides two types of signals i.e the Attention level values and the Eye blink Intensity values which are transmitted to the Android App via the bluetooth medium. The Android app performs the processing of the input signal and using the algorithms generates commands.The thought commands are sent to the raspberry pi through the bluetooth medium. The raspberry pi controls the movement of the wheels of the robotic prototype using the L296D motor driver circuit.

5.4.2 Flowcharts

5.4.2.1 Event flow and control flowchart.

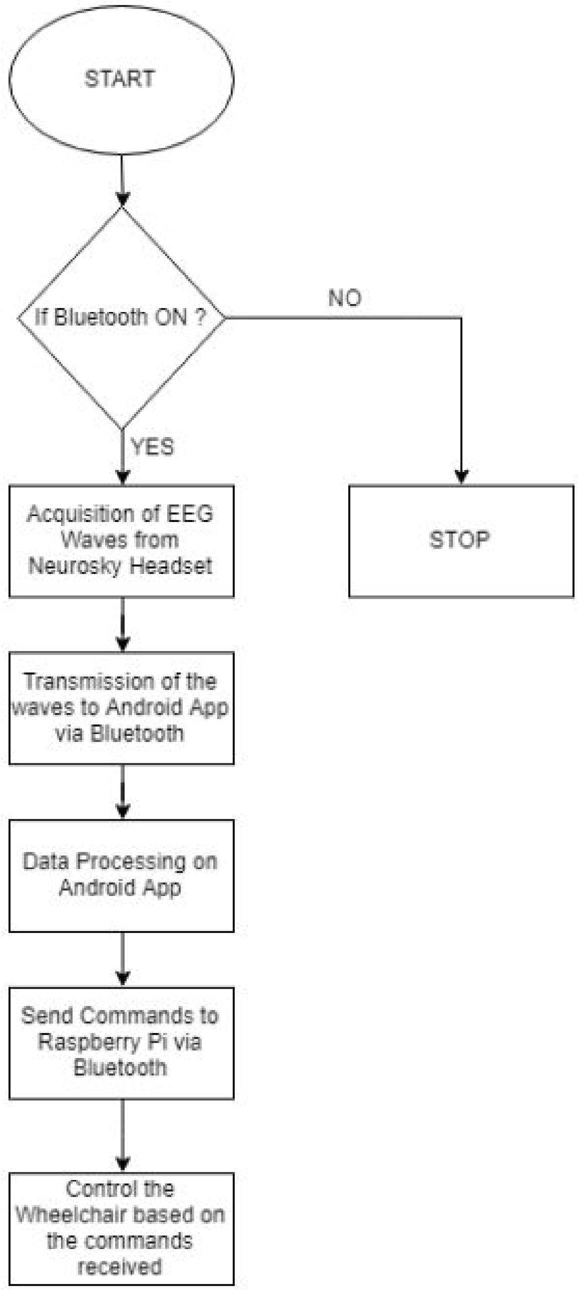


Figure 5.3

5.4.2.2 Working of the Automation device

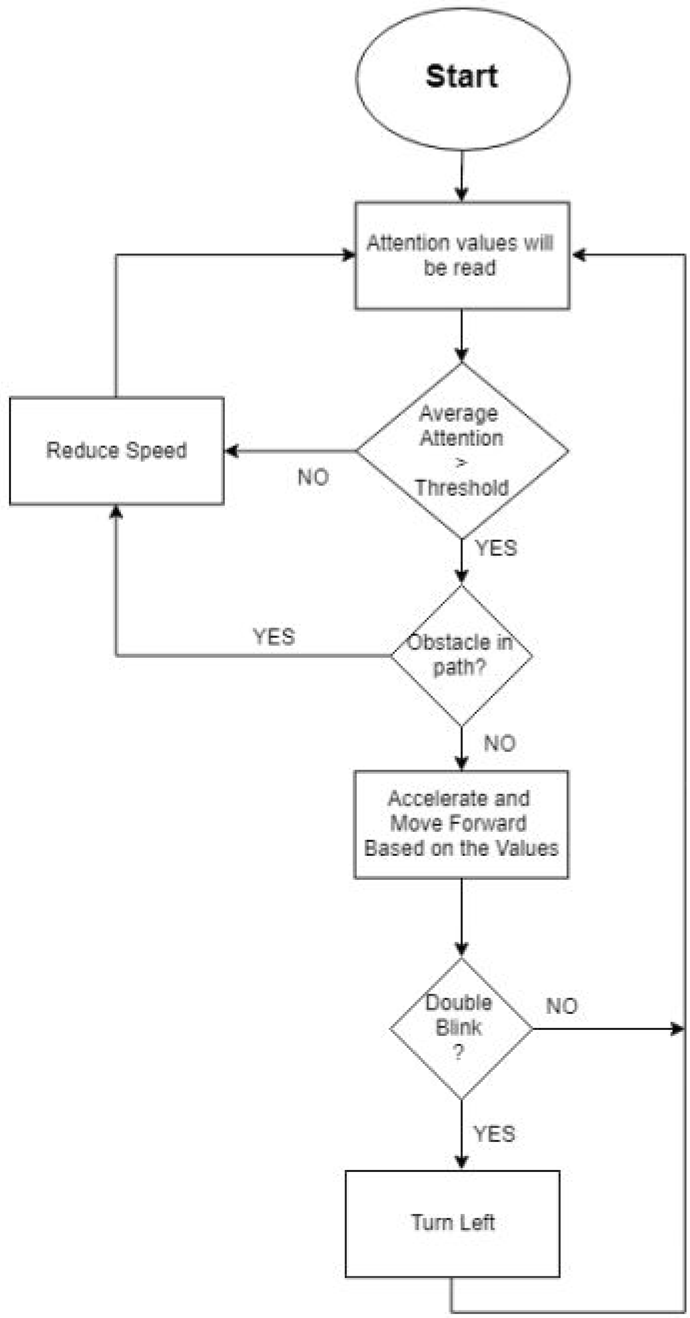


Figure 5.4

5.4.3 System Setup on robotic prototype

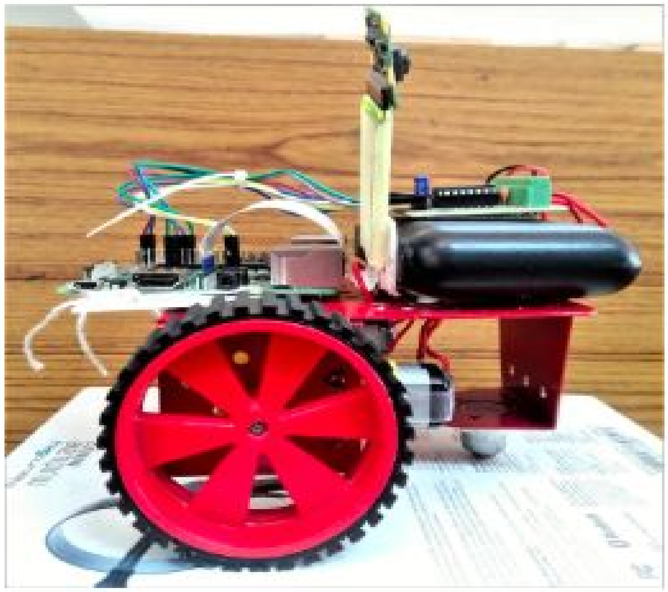


Figure 5.5

5.4.4 Android App screenshot

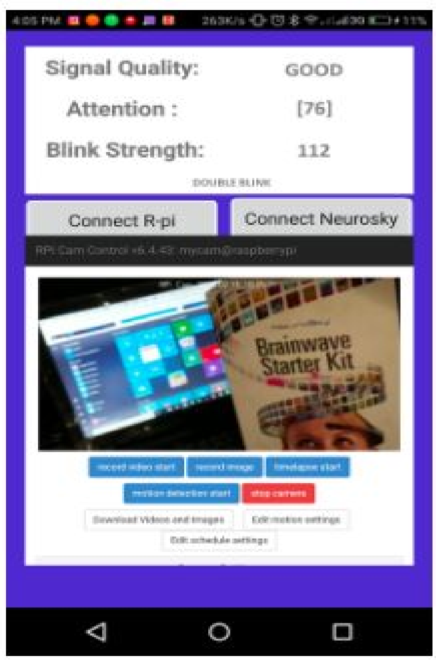


Figure 5.6

5.4.5 Wheelchair Design

Figure 5.7

Figure 5.8

Chapter 6

Implementation

This chapter discusses about the actual implementation of the project, the used technologies and the algorithms in the project development process.

6.1 Technologies

6.1.1 Raspbian OS

Raspbian is a Debian-based computer operating system for Raspberry Pi. Raspbian uses PIXEL - Pi Improved Xwindows Environment Lightweight as its main desktop environment. It is composed of a modified LXDE desktop environment and the OpenBox stacking window manager. The distribution is shipped with a copy of computer Algebra program ‘Mathematica’ , a version of ‘Minecraft’ called Minecraft Pi and a light-weight version of ‘Chromium’.

6.1.2 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.[26]

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

6.2 Algorithms

6.2.1 Python Script on RPi

```
from bluetooth import *
import time
import RPi.GPIO as GPIO

# creating RFCOMM Bluetooth socket which acts as bluetooth server and binds to a port
server_sock=BluetoothSocket( RFCOMM )
server_sock.bind(("",PORT_ANY))
server_sock.listen(1)

port = server_sock.getsockname()[1]

uuid = "94f39d29-7d6d-437d-973b-fba39e49d4ee"


#advertise the Bluetooth server to nearby devices
advertise_service( server_sock, "SampleServer",
                   service_id = uuid,
                   service_classes = [ uuid, SERIAL_PORT_CLASS ],
                   profiles = [ SERIAL_PORT_PROFILE ]
                  )

print "Waiting for connection on RFCOMM channel %d" % port

#Connect to the Android device after "Connect Rpi button" pressed on Android app
client_sock, client_info = server_sock.accept()
print "Accepted connection from ", client_info

#Motor Pins
Motor1A = 27        #Motor 1 pin 1
Motor1B = 17        #Motor 1 pin 2
Motor1E = 5         #Motor 1 enable pin

Motor2A = 23        #Motor 2 pin 1
Motor2B = 24        #Motor 2 pin 2
Motor2E = 6         #Motor 2 enable pin

GPIO.setmode(GPIO.BCM)          # GPIO Numbering
GPIO.setup(Motor1A,GPIO.OUT)      # All pins as Outputs
GPIO.setup(Motor1B,GPIO.OUT)
GPIO.setup(Motor1E,GPIO.OUT)
```

```
GPIO.setup(Motor2A,GPIO.OUT)      # All pins as Outputs
GPIO.setup(Motor2B,GPIO.OUT)
GPIO.setup(Motor2E,GPIO.OUT)

try:
   while True:
      client_info = client_sock.recv(1024)   #receive the data packets from connected android device
      print "received [%s]" % client_info

     if client_info=="Double Blink Detected":
       print "turning right ----------->>>>>>"

       GPIO.output(Motor1A,GPIO.LOW)      #Logic for turning Right
       GPIO.output(Motor1B,GPIO.HIGH)
       GPIO.output(Motor1E,GPIO.HIGH)

       GPIO.output(Motor2A,GPIO.HIGH)
       GPIO.output(Motor2B,GPIO.LOW)
       GPIO.output(Motor2E,GPIO.HIGH)

       sleep(2)

       GPIO.output(Motor1E,GPIO.LOW)     #stop turning after 2 seconds
       GPIO.output(Motor2E,GPIO.LOW)


      elif(client_info.isdigit()):        #if attentio value is received from Android app

       if(int(client_info)>60):          #the Attention threshold is set to 60 on a scale of 0-100
         print "moving front  ^^^^^^"
         GPIO.output(Motor1A,GPIO.HIGH)   #enable motor 1
         GPIO.output(Motor1B,GPIO.LOW)
         GPIO.output(Motor1E,GPIO.HIGH)

         GPIO.output(Motor2A,GPIO.HIGH)   #enable motor 2
         GPIO.output(Motor2B,GPIO.LOW)
         GPIO.output(Motor2E,GPIO.HIGH)

      else:
         print "invalid data received"


except KeyboardInterrupt:
   pass

except:
   print "Other error or exception occurred!"
```

```
#clean everything
finally:
    print "disconnected"
    io.cleanup()
    client_sock.close()
    server_sock.close()
    print "all done"
```

6.2.2 Android Script

```
package com.neurosky.algo_sdk_sample;
import android.media.MediaPlayer;
import android.net.Uri;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.content.res.Configuration;
import android.graphics.Color;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.TextView;
import android.widget.Button;
import android.widget.Toast;
import java.io.OutputStream;
import java.util.Set;
import java.io.IOException;
import java.io.InputStream;
import java.util.UUID;
import com.neurosky.AlgoSdk.NskAlgoDataType;
import com.neurosky.AlgoSdk.NskAlgoSdk;
import com.neurosky.AlgoSdk.NskAlgoSignalQuality;
import com.neurosky.AlgoSdk.NskAlgoType;
import com.neurosky.connection.ConnectionStates;
import com.neurosky.connection.TgStreamHandler;
import com.neurosky.connection.TgStreamReader;
import com.neurosky.connection.DataType.MindDataType;
import android.content.Context;
import android.media.AudioManager;
import android.widget.LinearLayout;
```

```
import java.util.Random;
/*

 * Project Name:       Brain Controlled Wheelchair
 * Author List:            Vinit Todai, Akshat Kacheria, Anil Guruprasad
 * Filename:                   MainActivity.java
 * Functions:                  onCreate(Bundle), BTinit(), BTconnect(), onBackPressed(),
 *                  onConfigurationChanged(Configuration), onStatesChanged(int),
 *                  onChecksumFail(byte[] , int, int), onRecordFail(int),
 *                  onDataReceived(int, int, Object), showToast(String, int)
 * Global Variables:    global_att
 */

public class MainActivity extends Activity
{

   final String TAG = "MainActivityTag";

   // COMM SDK handles
   private TgStreamReader tgStreamReader;
   private BluetoothAdapter mBluetoothAdapter;

   // canned data variables
   private short raw_data[] = {0};
   private int raw_data_index= 0;

   private Button neurosky_connect_btn;
   private TextView attValue;
   private TextView forced_blink_strength_text;
   private TextView sqText;
   private TextView test_textview;
   private NskAlgoSdk nskAlgoSdk;
   long previous_click_time;

   //Additional Bluetooth Connect Variables for Raspberry-pi
   private final String DEVICE_ADDRESS = "B8:27:EB:93:92:2D"; //MAC Address of Bluetooth
Module of Raspberry pi
   private final UUID PORT_UUID = UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");

   private BluetoothDevice device;
   private BluetoothSocket socket;

   private OutputStream outputStream;
   private InputStream inputStream;
   boolean connected = false;
   boolean isRunning = false;
   String command;
```

```java
Button bt_connect_btn;
Button soundbutton;

public int OldMax =100;
public int OldMin =0;
public int NewMax =15;
public int NewMin =0;
public int global_att;
public int OldRange = (OldMax - OldMin);
public int NewRange = (NewMax - NewMin);

private Context mContext;
private Activity mActivity;

private LinearLayout mRootLayout;
private LinearLayout BCKG_COLOR;

private Button mBtnSetMediaVolume;
private TextView mTVStats;

private AudioManager mAudioManager;
private Random mRandom = new Random();
public int att_global;

/*
 *
 * Function Name:    onCreate
 * Input:            None
 * Output:           The function called when the app is opened
 * Logic:            Sets up the Mobile Screen when the app is opened
 */

@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    bt_connect_btn = (Button) findViewById(R.id.bt_connect_btn);
    neurosky_connect_btn = (Button) findViewById(R.id.neurosky_connect_btn);
    WebView myWebView = (WebView) findViewById(R.id.webview);
    myWebView.setWebViewClient(new WebViewClient());
    myWebView.getSettings().setJavaScriptEnabled(true);
    myWebView.getSettings().setDomStorageEnabled(true);
    myWebView.getSettings().setUseWideViewPort(true);
    myWebView.setInitialScale(50);
    myWebView.loadUrl("https://192.168.137.53/");
```

```
        // Get the application context
        mContext = getApplicationContext();
        mActivity = MainActivity.this;

        // Get the widget reference from xml layout
        mRootLayout = findViewById(R.id.root_layout);
        BCKG_COLOR = findViewById(R.id.color_wala);

        soundbutton = findViewById(R.id.sound);

        final MediaPlayer mp = new MediaPlayer();
        mp.setAudioStreamType(AudioManager.STREAM_ALARM);
        try
        {
mp.setDataSource(this,Uri.parse("android.resource://"+mContext.getPackageName()+"/"+R.raw.water
));
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }

        try
        {
            mp.prepare();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
        mp.start();

        soundbutton.setOnClickListener(new View.OnClickListener(){

            public void onClick(View v) {
                mp.start();
            }
        });

        mBtnSetMediaVolume = findViewById(R.id.btn_media);
        mTVStats = findViewById(R.id.tv_stats);

        mAudioManager = (AudioManager) getSystemService(AUDIO_SERVICE);
```

```
    // Set click listener for media button
    mBtnSetMediaVolume.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            int media_current_volume =
mAudioManager.getStreamVolume(AudioManager.STREAM_MUSIC);

            int media_max_volume =
mAudioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC);

            int random_volume = att_global;
            int NewValue = (((random_volume - OldMin) * NewRange) / OldRange) + NewMin;
            random_volume = NewValue;


            // Set media volume level
            mAudioManager.setStreamVolume(
                    AudioManager.STREAM_MUSIC, // Stream type
                    random_volume, // Index
                    AudioManager.FLAG_SHOW_UI // Flags
            );
        }
    });


    nskAlgoSdk = new NskAlgoSdk();

    try
    {
        // (1) Make sure that the device supports Bluetooth and Bluetooth is on
        mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
        if (mBluetoothAdapter == null || !mBluetoothAdapter.isEnabled()) {
            Toast.makeText(
                    this,
                    "Please enable your Bluetooth and re-run this program !",
                    Toast.LENGTH_LONG).show();
        }
    }

    catch (Exception e)
    {
        e.printStackTrace();
        Log.i(TAG, "error:" + e.getMessage());
        return;
    }
```

```
attValue = (TextView)this.findViewById(R.id.attText);
forced_blink_strength_text = (TextView) this.findViewById(R.id.forced_blink_strength_text);
test_textview = (TextView) this.findViewById(R.id.test_textview);
sqText = (TextView)this.findViewById(R.id.sqText);
sqText.setText("AGAIN");

int algoTypes = 0;

algoTypes += NskAlgoType.NSK_ALGO_TYPE_MED.value;

algoTypes += NskAlgoType.NSK_ALGO_TYPE_ATT.value;

algoTypes += NskAlgoType.NSK_ALGO_TYPE_BLINK.value;

algoTypes += NskAlgoType.NSK_ALGO_TYPE_BP.value;


nskAlgoSdk.NskAlgoStart(false);


// Raspberry pi Bluetooth connect button logic


//onClick Listener for "Connect Rpi" Button, if BTinit returns true,
// i.e, bluetooth is available on device and turned on, then Raspberry pi will be connected.
// A Bluetooth server is deployed on Raspberry pi
bt_connect_btn.setOnClickListener(new View.OnClickListener(){
  @Override
  public void onClick(View v)
  {

    if(BTinit())
    {
      BTconnect();
    }

  }


});



//onClick Listener for "Connect Neurosky" Button,
neurosky_connect_btn.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
```

```
            raw_data = new short[512];
            raw_data_index = 0;


            tgStreamReader = new TgStreamReader(mBluetoothAdapter,callback);

            if(tgStreamReader != null && tgStreamReader.isBTConnected())
            {

                // Prepare for connecting Android app with EEG headset
                tgStreamReader.stop();
                tgStreamReader.close();
            }

            tgStreamReader.connect();
        }
    });



    //onClick Listener for Signal Quality of brainwaves from headset, if Signal is POOR or MEDIUM,
    // then UI will display it and corresponding data will be sent to raspberry pi

    nskAlgoSdk.setOnSignalQualityListener(new NskAlgoSdk.OnSignalQualityListener() {
        @Override
        public void onSignalQuality(final int level) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    String sqStr = NskAlgoSignalQuality.values()[level].toString();
                    sqText.setText(sqStr);

                    if(sqText.getText().toString().equals("NOT DETECTED") ||
sqText.getText().toString().equals("POOR") || sqText.getText().toString().equals("MEDIUM"))
                    {
                        command = "NOT_GOOD_SIGNAL";


                        // Write on outputstream, i.e. send data to Raspberry pi via bluetooth
                        try
                        {
                            outputStream.write(command.getBytes());
                        }
                        catch(IOException e)
                        {
                            e.printStackTrace();
```

```
                }
            }

            else
            {
                int cond=1;
            }

        }
    });
  }
});


//onClick Listener for Attention values from headset
    nskAlgoSdk.setOnAttAlgoIndexListener(new NskAlgoSdk.OnAttAlgoIndexListener()
    {
        @Override
        public void onAttAlgoIndex(int value)
        {
            global_att = value;
            String send_med_value = String.valueOf(value);

            //send the attention value directly to Raspberry pi
            try
            {
                outputStream.write(send_med_value.getBytes());
            }
            catch(IOException e)
            {
                e.printStackTrace();
            }

            Log.d(TAG, "NskAlgoAttAlgoIndexListener: Attention:" + value);
            String attStr = "[" + value + "]";
            final String finalAttStr = attStr;
            att_global=value;

            runOnUiThread(new Runnable()
            {
                @Override
                public void run()
                {
                    // change UI elements here
                    attValue.setText(finalAttStr);
                    mBtnSetMediaVolume.performClick();
```

```
                //The background color of Andrroid app consists of 7 colors which resemble the
rainbow colors, VIBGYOR
                //If Attention is less, then background is RED and when it is more, the background
turns violet and it
                // changes gradually in between depending on the Attention level

                //RED
                if(global_att > 0 && global_att < 15)
                {
                  BCKG_COLOR.setBackgroundColor(Color.parseColor("#FF0000"));
                }
                //ORANGE
                else if (global_att > 16 && global_att < 30)
                {
                  BCKG_COLOR.setBackgroundColor(Color.parseColor("#FF7F00"));
                }
                //YELLOW
                else if (global_att > 31 && global_att < 45)
                {
                  BCKG_COLOR.setBackgroundColor(Color.parseColor("#FFFF00"));
                }
                //GREEN
                else if (global_att > 46 && global_att < 60)
                {
                  BCKG_COLOR.setBackgroundColor(Color.parseColor("#00FF00"));
                }
                //BLUE
                else if (global_att > 61 && global_att < 75)
                {
                  BCKG_COLOR.setBackgroundColor(Color.parseColor("#0000FF"));
                }
                //INDIGO
                else if (global_att > 76 && global_att < 90)
                {
                  BCKG_COLOR.setBackgroundColor(Color.parseColor("#4B0082"));
                }
                //VIOLET
                else
                {
                  BCKG_COLOR.setBackgroundColor(Color.parseColor("#9400D3"));
                }

              }
            });
          }
        });
//onClick Listener for Eye Blinks
```

```
nskAlgoSdk.setOnEyeBlinkDetectionListener(new NskAlgoSdk.OnEyeBlinkDetectionListener() {
    @Override
    public void onEyeBlinkDetect(int strength) {
        Log.d(TAG, "NskAlgoEyeBlinkDetectionListener: Eye blink detected: " + strength);
        final int final_strength = strength;

        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                String text1 = "BLINK_DETECTED";
                soundbutton.performClick();

                    forced_blink_strength_text.setText(String.valueOf(final_strength));

                    // change UI elements here

                    try
                    {
                        outputStream.write(text1.getBytes());
                    }
                    catch (IOException e)
                    {
                        e.printStackTrace();
                    }

                    long temp = System.currentTimeMillis();

                    // logic to check for two eye blinks continuously in a span of 1 second threshold
                    if (previous_click_time != 0)
                    {
                        if (temp - previous_click_time < 1000)
                        {
                            test_textview.setText("DOUBLE_BLINK");
                            soundbutton.performClick();
                            command="DOUBLE_BLINK";

                            //send the string DOUBLE BLINK to raspberry pi via bluetooth
                            try
                            {
                                outputStream.write(command.getBytes());
                            }

                            catch (IOException e)
                            {
                                e.printStackTrace();
                            }
                        }
```

```
                    else
                    {
                        test_textview.setText("NO DOUBLE BLINK");
                    }
                }

                previous_click_time = temp;

            }
        });

    }
});


} // END OF ONCREATE



/*
 *
 * Function Name:      BTinit
 * Input:              None
 * Output:             Initializes the Bluetooth in Android device
 * Logic:              This function Checks if the device supports bluetooth and if yes, the is it
enabled.
 *                     If not, the program will ask permission from the user to enable it
 * Example Call:       BTinit()
 */
public boolean BTinit()
{
    boolean found = false;

    BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

    if(bluetoothAdapter == null) //Checks if the device supports bluetooth
    {
        Toast.makeText(getApplicationContext(), "Device doesn't support bluetooth",
Toast.LENGTH_SHORT).show();
    }

    if(!bluetoothAdapter.isEnabled()) //Checks if bluetooth is enabled. If not, the program will ask
permission from the user to enable it
    {
        Intent enableAdapter = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableAdapter,0);
```

```
        try
        {
            Thread.sleep(1000);
        }
        catch(InterruptedException e)
        {
            e.printStackTrace();
        }
    }

    Set<BluetoothDevice> bondedDevices = bluetoothAdapter.getBondedDevices();

    if(bondedDevices.isEmpty()) //Checks for paired bluetooth devices
    {
        Toast.makeText(getApplicationContext(), "Please pair the device first",
Toast.LENGTH_SHORT).show();
    }
    else
    {
        for(BluetoothDevice iterator : bondedDevices)
        {
            if(iterator.getAddress().equals(DEVICE_ADDRESS))
            {
                device = iterator;
                found = true;
                break;
            }
        }
    }

    return found;
  }



  /*
   *
   * Function Name:      BTconnect
   * Input:                  None
   * Output:                 Creates a socket to handle the outgoing bluetooth connection to
Raspberry pi
   * Logic:                  This function creates an RFCOMM Socket and mobile device acts as a
client and
   *                 the Raspberry pi acts as Bluetooth Server
   * Example Call:           BTconnect()
   */
```

```
    public boolean BTconnect()
    {
        try
        {
            socket = device.createRfcommSocketToServiceRecord(PORT_UUID); //Creates a socket to
handle the outgoing connection
            socket.connect();

            Toast.makeText(getApplicationContext(),
                    "Connection to Raspberry Pi successful", Toast.LENGTH_LONG).show();
            connected = true;
            neurosky_connect_btn.setEnabled(true);
        }
        catch(IOException e)
        {
            e.printStackTrace();
            connected = false;
        }

        if(connected)
        {
            try
            {
                outputStream = socket.getOutputStream(); //gets the output stream of the socket
            }
            catch(IOException e)
            {
                e.printStackTrace();
            }

            try
            {
                inputStream = socket.getInputStream(); //gets the input stream of the socket
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
        }

        return connected;
    }
```

```
/*
 *
 * Function Name:      onBackPressed
 * Input:                        None
 * Output:                       Closes the connection and quits app
 * Logic:                        Closes the app on Back pressed
 */

@Override
public void onBackPressed()
{
    nskAlgoSdk.NskAlgoUninit();
    finish();
}



private TgStreamHandler callback = new TgStreamHandler()
{


    /*
     *
     * Function Name:             onStatesChanged
     * Input:                     connectionStates
     * Output:                              Displays the different states the Headset signal is at present.
     *                    States can be : STATE_CONNECTING, STATE_CONNECTED,
STATE_GET_DATA_TIME_OUT etc as mentioned
     */

    @Override
    public void onStatesChanged(int connectionStates)
    {
        // TODO Auto-generated method stub
        Log.d(TAG, "connectionStates change to: " + connectionStates);

        switch (connectionStates)
        {
            case ConnectionStates.STATE_CONNECTING:
                // Do something when connecting
                break;
            case ConnectionStates.STATE_CONNECTED:
                // Do something when connected
                tgStreamReader.start();
                showToast("Connection to Neurosky Mindwave Mobile successful",
Toast.LENGTH_SHORT);
```

```
                break;
            case ConnectionStates.STATE_WORKING:
                break;
            case ConnectionStates.STATE_GET_DATA_TIME_OUT:
                showToast("Get data time out!", Toast.LENGTH_SHORT);
                break;
            case ConnectionStates.STATE_STOPPED:
                break;
            case ConnectionStates.STATE_DISCONNECTED:
                break;
            case ConnectionStates.STATE_ERROR:
                break;
            case ConnectionStates.STATE_FAILED:
                break;
        }
    }


    /*
     *
     * Function Name:        onChecksumFail
     * Input:                bytes, i, i1
     * Output:               Handles the output when the checksum received from the Headset
and the original checksum does not match
     */
    @Override
    public void onChecksumFail(byte[] bytes, int i, int i1)
    {
        // We can add the code to handle the output if checksum fails. We can make a toast and
display and quit the app

    }


    /*
     *
     * Function Name:        onRecordFail
     * Input:                flag
     * Output:               Makes a log in Android Logcat and display the error
     */
    @Override
    public void onRecordFail(int flag)
    {
        // You can handle the record error message here
        Log.e(TAG,"onRecordFail: " +flag);

    }
```

```
/*
 *
 * Function Name:          onDataReceived
 * Input:                  datatype, data, obj
 * Output:                 The important function that receives the data from EEG Headset
and classifies
 *                 it into different datatypes based on the Headset Documentation
 *                 eg : CODE_ATTENTION, CODE_MEDITATION, CODE_POOR_SIGNAL,
CODE_RAW
 * Logic:                  It receives the data packets from headset and classifies into datatypes as
described above
 */

@Override
public void onDataReceived(int datatype, int data, Object obj)
{
    // You can handle the received data here
    // You can feed the raw data to algo sdk here if necessary.
    Log.i(TAG,"onDataReceived");
    switch (datatype) {
        case MindDataType.CODE_ATTENTION:

            short attValue[] = {(short)data};


nskAlgoSdk.NskAlgoDataStream(NskAlgoDataType.NSK_ALGO_DATA_TYPE_ATT.value, attValue,
1);
            break;
        case MindDataType.CODE_MEDITATION:
            short medValue[] = {(short)data};

nskAlgoSdk.NskAlgoDataStream(NskAlgoDataType.NSK_ALGO_DATA_TYPE_MED.value,
medValue, 1);
            break;
        case MindDataType.CODE_POOR_SIGNAL:
            short pqValue[] = {(short)data};

nskAlgoSdk.NskAlgoDataStream(NskAlgoDataType.NSK_ALGO_DATA_TYPE_PQ.value, pqValue,
1);
            break;
        case MindDataType.CODE_RAW:
            raw_data[raw_data_index++] = (short)data;
            if (raw_data_index == 512) {

nskAlgoSdk.NskAlgoDataStream(NskAlgoDataType.NSK_ALGO_DATA_TYPE_EEG.value,
```

```
raw_data, raw_data_index);
                    raw_data_index = 0;
                }
                break;
            default:
                break;
        }
    }


    /*
     *
     * Function Name:          showToast
     * Input:                  msg --> The message to show in toast
     *                 timeStyle --> The time to show the message
     * Output:                         Displays a toast in android
     * Example Call:           showToast("Connection to Neurosky Mindwave Mobile successful",
Toast.LENGTH_SHORT);
     */

    public void showToast(final String msg, final int timeStyle) {
        MainActivity.this.runOnUiThread(new Runnable() {
            public void run() {
                Toast.makeText(getApplicationContext(), msg, timeStyle).show();
            }

        });
    }
  };}
```

Chapter 7

System Test Document

This chapter gives a brief idea regarding the testing strategy and approach used in the implementation of this project. The methodology of the tests performed and also the features of the project which were tested are listed in this chapter.

7.1 Introduction

7.1.1System Overview

The aim of this project is to create a Brain Controlled Smart Automaton Device, which can be used using a mobile application and a EEG headset. Using this application the user must be able to control a wheelchair prototype using attention and eye blink values. The applications communicates with the EEG headset and raspberry pie to control the prototype.

7.1.2Testing Approach

A test approach defines how the testing strategy would be executed. Testing would help us to find and rectify the faults (if any) within our system. A brief idea of the testing would be carried out is described below.

1. **Unit Testing**

Unit Testing will test each separate module or unit independently to ensure that all units are working as expected without any flaws. This allows us to identify errors within each module and rectify those faults easily.

2. System Testing

In System testing the whole system is tested after integrating all modules of the software. The complete software is tested to ensure that all the modules work perfectly in collaboration.

3. Performance Testing

Performance testing ensures that the software meets the performance expectations of the user and also meet the performance standards that are specified.

4. Documentation Testing

The documentation provided to the user is tested to make sure that it is accurate and that all the features are explained properly and can be understood easily.

7.2 Test Plan

7.2.1Features to be tested:

Module: Signal Acquisition Module

Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**
Electroencaphalograghy Waves Captured	Y	Y	N	N
Electromyograghy Waves Captured	Y	N	Y	N
Action				
Passes the prepared data to preprocessing module	Y	Y	Y	N

Module: Data Pre - Processing Module

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Attention strength calculated	Y	Y	N	N
Blink strength and count calculated	Y	N	Y	N
Action				
Passes prepared data to classification module	Y	N	N	N

Module: Feature Classification

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Attention Strength > 70	Y	Y	N	N
Blink strength > 80 && 2 blinks	Y	N	Y	N
Action				
Move forward	Y	Y	N	N
Rotate clockwise	Y	N	Y	N

Module: Obstacle avoidance

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Ultrasonic Sensor present	Y	Y	N	N
Object present	Y	N	Y	N
Action				
Wheelchair stops immediately	Y	N	N	N

7.2.3Testing tools and environment

Hardware Configuration

- PC (Intel Pentium 4 or above)
- Android Device(512MB RAM minimum)
- EEG Headset
- Raspberry Pi
- Robotic Prototype

Software Configuration

- Android Operating system(4.0 and above)
- Android SDK
- Windows OS (7 and above)
- Raspbian OS
- Python

7.3 <u>Test Cases</u>

Test Case TC01: Electroencephalography waves captured

Description: To check if eeg waves are correctly capture

Test steps

- Switch on eeg headset.
- Connect headset to android app
- Connect headset to raspberry pi
- Check wave strength

Expected Result: Attention values must be shown

Actual Result: Attention values shown. PASS

Test Case TC02: Electromyography waves captured
Description: To check if emg waves are correctly capture

Test steps

- After capturing eeg waves test if blink strength is shown
- Emg will cause spikes in waves

Expected Result: Spikes in waves with blinks

Actual Result: Spikes in waves with blinks shown. PASS

Test Case TC03: Attention strength calculated

Description: To check if correct attention values on scale of 1-100 are calculated

Test steps

- Once signals are captures. Waves must be passed through the algorithm to generate attention values

Expected Result: Integer based attention value must be shown

Actual Result: Attention value calculated. PASS

Test Case TC04: Blink strength and count calculated

Description: Calculate the blink strength on scale of 1-255 and collect count.

Test steps

- Once signals are captures. Blinks must be passed through the algorithm to generate blink strength values and blink count

Expected Result: Integer based blink strength on scale of 1-255 and count (single or double blink) must be captured

Actual Result: Blink strength and count calculated. PASS

Test Case TC05: Move forward if attention strength > 70

Description: Checking threshold values to start wheelchair

Test steps

- Set up and establish connection
- Focus till attention values > 70.

Expected Result: Wheelchair must start.

Actual Result: Wheelchair activated. PASS

Test Case TC06: Turn if blink strength > 80 and two blinks

Description: Check if double blink within 1 second and blink threshold exceeded for both as conditions to turn.

Test steps

- Set up and establish connection
- Double blink within 1 second and blink strength > 80

Expected Result: Wheelchair must turn clockwise.

Actual Result: Wheelchair turned. PASS

Test Case TC07: Ultrasonic sensor active

Description: Check if ultrasonic sensor correctly detects object.

Test steps

- Attach ultrasonic sensor
- Establish connection
- place object in the front of the wheelchair
- Check for feedback

Expected Result: Object must be detected.

Actual Result: Sensor active and object detected. PASS

Test Case TC08: Emergency Braking

Description: Wheelchair must brake under object detection by ultrasonic sensor.

Test steps

- Establish connection
- Place object in the front of the ultrasonic sensor

Expected Result: Motors must stop and wheelchair must brake.

Actual Result: Wheelchair stopped upon object detection. Emergency braking works. PASS

Chapter 8

Conclusion

This chapter gives the conclusion of the project, apart from this we also discuss the future scope of this project.

This project does not aim to reinvent the wheel. It is just making a user viable interface. Currently the value and unique selling point of our device will be in its low command delay, simple yet efficient command classification abilities, ease of use for layman and automatic avoidance for the robotic wheelchair. The device can be spread via doctors of that field, online,convention and charity drives.The current development of Brain controlled applications is really exciting as they can be used in a variety of applications such as the brain controlled wheelchair,mouse,drones etc and the world is only a few years away from perfecting this technology. Although the current robotic prototype only allows limited movements, we hope the future will increase the number of commands that can be given by the user during operation . We hope to be an integral part of this research that will change lives and revolutionize the extent of independency the physically challenged members of our society will achieve with this technology.

8.1 Future Scope

- Obstacle Avoidance with camera module -Automatically detect and avoid Obstacles.
- Control using user Attention - If Low attention, then it will stop.Accelerate based on the input values
- Emergency Braking System - If the user abruptly closes his eyes sensing some danger, then the Wheelchair will stop
- Staircase Climbing Mode - To assist the elderly to help climb the stairs

Bibliography

[1] Tom Carlson, Member IEEE, and Jos´e del R.Mill´an,Senior Member IEEE, "Brain controlled wheelchairs: A robotic architecture" IEEE Robotics and Automation Magazine, 20(1):65-73, March 2013

[2] Siliveru Ramesh, M Gopi Krishna, "Brain Computer Interface System for Mind Controlled Robot using Bluetooth", International Journal of Computer Applications (0975 – 8887) Volume 104 – No 15, October 2014

[3] Niha K., Dr. Aisha Banu W., "Brain Signal Processing: Technologies, Analysis and Application" IEEE International Conference on Computational Intelligence and Computing Research, 2016

[4] Imran Ali Mirza, Amiya Tripathy, "Mind-Controlled Wheelchair using an EEG Headset and Arduino Microcontroller", 2015 International Conference on Technologies for Sustainable Development (ICTSD-2015), Feb. 04 – 06, 2015, Mumbai, India, 2015

[5] Anupama H.S, N.K.Cauvery, Lingaraju G.M, "Brain controlled wheelchair for disabled", International journal of computer science engineering and information ISSUE technology", Vol.4, 2, April 2014, pp-156-166.

[6] Atanasios Vourvopoulos, Fotis Liarokapis, "Brain controlled NXT Robot: Teleoperating a robot through brain electrical activity", IEEE conference on Games and virtual worlds for serious applications, Athens, Greece, 2011.

[7] Naveen R.S, Anitha Julian, " Brain computing interface for wheelchair control", 4th ICCNT, 2013

[8] K. Sudheer, T.V.Jnarhana Rao, C.H.Sridevi, M.S.Madan Mohan, " Voice and gesture based electric powered wheelchair using ARM", International journal of research in computer and communication technology, vol.1, issue 6, November 2012

[9] Mohan Kumar R, Lohit H.S, Manas Ranjan Mishra, Md Basheer Ahamed, " Design of Multipurpose wheel chair for physically challenged elder people", SASTECH, Vol.11, Issue 1, April 2012. Pp;107-117.

EXTRA READS

[10] Jobby K.Chacko, Deepu OoMMEN, Kevin K.Mathew, Noble Sunny, N.Babu, "Microcontroller based EOG guided wheelchair", International journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering, vol.7, no.11, 2013.

[11] FRDM-KL25Z User's manual.

[12] Vijay Raghav Varada, Deepshikha Moolchadani, Asil Rohit, " Measuring and processing the brain's EEG signals with visual feedback for human machine interface", International Journal of Scientific and Engineering Research , Vol.4, Issue 1, Jan., 2013.

[13] Butt, A., Stanacevic, M.,"Implementation of Mind Control Robot" Systems, Applications and Technology Conference (LISAT), 2014 IEEE Long

Plagiarism Report

Turnitin Originality Report

Processed on: 10-May-2019 12:44 IST
ID: 1128213234
Word Count: 10786
Submitted: 1

C-45 By Akshat Kacheria

Similarity Index	Similarity by Source	
8%	Internet Sources:	7%
	Publications:	2%
	Student Papers:	9%

2% match (student papers from 05-Jun-2018)
Submitted to National Institute of Industrial Engineering on 2018-06-05

2% match (Internet from 30-May-2014)
http://liks.fav.zcu.cz/mediawiki/images/8/83/Baztarricadiplomaproject.pdf

2% match (student papers from 07-Nov-2015)
Submitted to Central Queensland University on 2015-11-07

1% match (student papers from 10-Apr-2019)
Submitted to Universiti Teknologi Petronas on 2019-04-10

1% match (Internet from 03-Nov-2010)
http://plumbot.com/EEG.html

Brain Controlled Interface Based Automation Device Submitted in partial fulfillment of requirements For the degree of Bachelor of Technology in Computer Engineering by Akshat Kacheria - 1511081 Vinit Todai - 1511116 Anil Guruprasad - 1512002 Guide Prof. Mansi Kambli Department of Computer Engineering K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai) Batch 2015 -2019 K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai) Certificate This is to certify that the dissertation report entitled Brain Controlled Interface Based Automation Device is bona fide record of the dissertation work done by 1. Akshat Kacheria 2. Vinit Todai 3. Anil Guruprasad in the year 2018-19 under the guidance of Prof. Mansi Kambli of Department of Computer Engineering in partial fulfillment of requirement for the Bachelors in Technology degree in Computer Engineering of University of Mumbai. Guide Head of the Department Principal Date: Place: Mumbai-77 K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai) Certificate of Approval of Examiners We certify that this dissertation report entitled Brain Controlled Interface Based Automation Device is bona fide record of project work done by 1. Akshat Kacheria 2. Vinit Todai 3. Anil

Participation and certificates

Following provides a summary as to how the project has represented the **college name at various platforms**.

ID	Competition Name	Organised by	Status
1	India Innovation Challenge Design Contest 2018 (IICDC 2018)	Department of Science and Technology (DST), Texas Instruments. Anchored by NSRCEL, Indian Institute of Management, Bangalore (IIMB) and supported by MyGov	Semi - Finalist (Top 60 out of 10,146 projects)
2	e-Yantra Ideas Competition	IIT Bombay and sponsored by Ministry of Human Resource Development	Regional Finalist
3	Natarajan Education Society (NES) Innovation Awards	NES Innovation Council (NESIC)	Finalist (Top 50 out of thousands of projects)
4	Best Project Award 2019	Dept. of Electronics Engg., KJSCE	Winner
5	Maker Mela 2019 - India's Biggest Maker Gathering	Riidl, Supported by Department of Science and Technology, Govt. of India	Top 100
6	Ennovate - Innovation and Business Plan Competition	BloomBox KJSCE, StrataHive	Runner - up
7	Prakalpa 2019 - State level technical paper presentation and project exhibition	KJSCE and the ISTE Students Chapter – KJSCE	Participant
8	IETE - Project Competition	IETE, KJSCE	Participant
9	Mumbai Times Cover page article on project	Maharashtra times	Published

ERTS Lab
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Powai, Mumbai-400 076

eyantra

Engineering a better tomorrow

Certificate of Participation

This is to certify that *Akshat*, a student from *K. J. Somaiya College of Engineering, Vidya Vihar* has participated in the Regional Finals of the e-Yantra Ideas Competition (eYIC-2019) held at *K. J. Somaiya College of Engineering, Vidya Vihar* on *21st February, 2019*.

He/She is a member of the team having following team members,

1. *Akshat*
2. *Vinit Todai*
3. *Anil Guruprasad*

Mentored By: *Prof. Manasi Kambli*

This team demonstrated the project titled " *Brain Controlled Wheelchair* ".

Prof. Kavi Arya
Principal Investigator, e-Yantra
Professor
Department of Computer Science and Engineering
Indian Institute of Technology Bombay

859befb17ee54dfa86eb1970625fbbba04ae1e7f

e-Yantra is a project sponsored by MHRD, Government of India, under the National Mission on Education through ICT (NMEICT).

Certificate of Merit: awarded to finalist teams
Certificate of Completion: awarded to teams for completing all the tasks of the competition
Certificate of Participation: awarded to teams for partial completion of tasks in the competition
Letter of Participation: awarded as an acknowledgement of participation on e-Yantra Letterhead

Department of Computer Science and Engineering
Indian Institute of Bombay, Powai Mumbai
e-Yantra Project

Respected Sir/Madam,

Greetings from e-Yantra!

We do hope participating in the Regional finals of eYIC-2019 held at ***K. J. Somaiya College of Engineering*** was an enriching experience.

As a token of appreciation for the team effort, there is a cash award for your team for participating and exhibiting the working prototype of your Project ***Brain Controlled Wheelchair.*** at the said Regional Finals.

Our heartiest congratulations to your team on being one of the Regional Finalist.

Prize Money:

The token of appreciation is a cash award of Rs. 5,000 which will be given to a team. This amounts to Rs. 1000 per team member (One teacher mentor and four students only). In case there are less than 5 members in your team the total amount of the cash award for the team will be accordingly less than Rs. 5,000.

The details of disbursement of the said amount will be communicated to you at a later date after the completion of all the regional finals.

For your e-Yantra Lab:

In addition to the cash award, it is indeed our pleasure to provide your embedded systems and robotics lab setup under e-Yantra Lab Setup Initiative (eLSI) with a drone kit in recognition of sustained use of lab and taking benefit of e-Yantra Initiatives.

If you have any query, please feel free to contact us on support@e-yantra.org or call us on 022 - 2576-4986.

Best Wishes,
e-Yantra Team

NES

NES INNOVATION AWARDS

Certificate of Participation

Presented to Mr. / Ms. **Anil Guruprasad**

In recognition of your project titled **Brain Controlled Smart Wheelchair**

from **K. J. Somaiya College Of Engineering**

during the year **2018 - 2019**

Dr. Ganesh Natarajan
Principal Trustee & Founder
Natarajan Education Society

BloomBox — The Entrepreneurship Cell, KJSCE StrataHive — Enhancing the Digital Experience

Somaiya TRUST

CERTIFICATE

OF EXCELLENCE

THIS CERTIFICATE IS AWARDED TO

AKSHAT KACHERIA

for securing third place in Ennovate 2019, conducted by BloomBox, E-Cell KJSCE on 5th April 2019.

PROF. AVANI SAKHAPARA	PROF. SUNAYANA JADHAV	SAGAR KORDE
EVENT COORDINATOR	EVENT COORDINATOR	FACULTY COORDINATOR
KJSCE	KJSCE	BLOOMBOX - ENTREPRENEURSHIP CELL, KJSCE

K.J. SOMAIYA COLLEGE OF ENGINEERING, MUMBAI-77
(An autonomous college affiliated to Mumbai University)

Presents

PRAKALPA'19

State Level Competition

Organised by

ISTE Students Chapter (MH-60)

CERTIFICATE OF PARTICIPATION

This is to certify that Bro/~~Sis~~

Akshat Kacheria

has participated in ~~Technical Paper Presentation~~/ Project Exhibition in the category Hardware in PRAKALPA held on 22nd-23rd March 2019.

SJ Gala		Shubha
BRO. SACHIN GALA	PROF. SHILA DHANDE/ PROF. ABHISHEK BHADURIA	DR. SHUBHA PANDIT
Chairperson	Faculty Advisor	Principal

K.J. SOMAIYA COLLEGE OF ENGINEERING, MUMBAI-77
(An autonomous college affiliated to Mumbai University)

Presents

PRAKALPA'19

State Level Competition

Organised by

ISTE Students Chapter (MH-60)

CERTIFICATE OF PARTICIPATION

This is to certify that Bro/~~Sis~~

Akshat. Kacheria

has participated in Technical Paper Presentation/ ~~Project Exhibition~~ in the category Medical - Health in PRAKALPA held on 22nd-23rd March 2019.

S J Gala		Shubha
BRO. SACHIN GALA	PROF. SHILA DHANDE/ PROF. ABHISHEK BHADURIA	DR. SHUBHA PANDIT
Chairperson	Faculty Advisor	Principal

K. J. Somaiya College of Engineering
(Autonomous College Affiliated to University of Mumbai)
Vidyavihar, Mumbai – 400077

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

I.E.T.E. STUDENTS' FORUM (KJSCE-ISF)

Certificate of Appreciation

This is to certify that

Bro.Akshat Kacheria

Bro.Vinit Todai

Bro.Anil Guruprasad

has participated in project competition organized by IETE-KJSCE-ISF.

Prof. Nitin S. Nagori **Faculty In-charge** **IETE- KJSCE-ISF**	**Dr. Ameya K. Naik** **HOD EXTC**	**Dr. Shubha Pandit** **PRINCIPAL**

K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)
Department of Electronics Engineering

Best Project Award -2019

certificate is hereby awarded in recognition of the work done by the team members

Anil Guruprasad, Akshat Kacheria, Vinit Todai

for the project title

Brain computer interface based automation device

guided by

Prof. Manasi Kambli & Prof. Sonia Joshi

(Dr. Shoba Gopalakrishnan)
Project In-charge

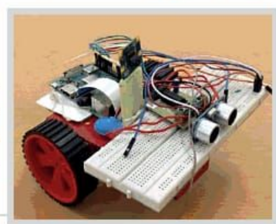
(Dr. J.H. Nirmal)
Head of the Department

पापणी हाले, व्हीलचेअर चाले

अनिकेत जाधव,
भारती विद्यापीठ कॉलेज ऑफ इंजिनीअरिंग
muntainbox@gmail.com

आयआयटीच्या ' इ यंत्र 'मधला आविष्कार

पापणी हलेल तशी चालणारी व्हीलचेअर, आग लागल्याची सूचना देणारा आणि ती विझवणारा पायरोबॉट अशा भन्नाट कल्पनांवरील प्रकल्प अलीकडेच पाहायला मिळाले आयआयटीच्या 'इ यंत्र' स्पर्धेमध्ये. या स्पर्धेची अंतिम फेरी येत्या ५ आणि ६ एप्रिलला आयआयटीमध्ये रंगणार असून, त्यात देशभरातल्या इंजिनीअरिंगच्या विद्यार्थ्यांचे प्रकल्प पाहायला मिळतील...



तरुणांच्या डोक्यामधल्या एकापेक्षा एक भन्नाट कल्पनांचा आविष्कार आयआयटीच्या 'इ यंत्र' स्पर्धेच्या प्रादेशिक फेरीमध्ये अलीकडेच पाहायला मिळाला. येत्या ५ आणि ६ एप्रिलला आयआयटीमध्ये त्याची अंतिम फेरी होतेय. एकूण १८ प्रकल्प यात सादर होणार आहेत. प्रादेशिक फेरीमध्ये ज्या दोन प्रकल्पांनी लक्ष वेधून घेतलं त्याविषयी...

आग विझवेल पायरोबॉट

आग विझवण्याचं जोखमीचं काम करताना अग्निशमन दलाच्या जवानांना अनेक अडचणींचा सामना करावा लागतो. हे लक्षात घेऊन नेरुळच्या रामराव आदिक इंजिनीअरिंग कॉलेजच्या विद्यार्थ्यांनी एक पायरोबॉट तयार केला आहे. महेश कुमा, किशन कुमा, सजीथ कुमार, सूरज डाकूआ अशी या विद्यार्थ्यांची नावं आहेत. स्वयंचलित असा हा रोबॉट आग विझवण्यासाठी उपयुक्त ठरू शकेल. ग्रीक भाषेमध्ये पायरो म्हणजे आग आणि यावरूनच या रोबोटला पायरोबॉट हे नाव दिलं आहे. विद्यार्थ्यांनी तयार केलेल्या या स्वयंचलित अग्निशमन यंत्रणेमध्ये कॅमेरा सिस्टिम आणि अग्निशमन करणाऱ्या बॉटचा समावेश आहे.

प्रकल्पाविषयी थोडं

कॅमेरा सिस्टिम इमारतीच्या फ्लॅट्समध्ये बसवता येईल. डीप लर्निंग या तंत्रज्ञानाच्या मदतीनं आग कुठे लागलीय ते यातून कळेल. आग कुठे आहे ते कळल्यावर इमारतीमध्ये ठेवलेल्या बॉटवर एक अॅलर्ट पाठवला जाईल. बॉट लाईन फॉलोअर तंत्राचा वापर करून, जिथे आग लागली आहे त्या खोलीत तो पोहोचेल. बॉटच्या आत ठेवलेल्या अग्निशमन यंत्राद्वारे आग विझवण्यात येईल. आग विझवण्याचं काम पूर्ण झाल्यावर बॉट परत जाईल. या पायरोबॉटमुळे आग विझवताना अग्निशमन दलाच्या जवानांची जीवावरची जोखीम कमी होईल. मोठमोठ्या गोदामात काही वेळा जवानांना जाणं शक्य नसतं. तिथे हा पायरोबॉट उपयुक्त ठरू शकेल. इलेक्ट्रॉनिक्स अँड टेलिकम्युनिकेशन इंजिनीअरिंगचे विभागप्रमुख मुकेश पाटील यांनी विद्यार्थ्यांना प्रकल्पासाठी कॉलेजकडून निधी उपलब्ध करून दिला. प्राध्यापक गजराज सिंह यांनी त्यांना मार्गदर्शन केलं.

ब्रेन कंट्रोल्ड व्हीलचेअर

पक्षाघात झालेल्या रुग्णाला हालचालींसाठी संपूर्णपणे दुसऱ्या व्यक्तीवर आणि व्हीलचेअरवर अवलंबून राहावं लागतं. विद्याविहारच्या के.जे. सोमय्या कॉलेज ऑफ इंजिनीअरिंगच्या विद्यार्थ्यांनी त्यांच्यासाठी एक व्हीलचेअर तयार केली आहे. विनित तोडाई, अनिल गुरूप्रसाद, अक्षत कचेरीया अशी या विद्यार्थ्यांची नावं आहेत. मेंदूमध्ये येणाऱ्या विचारांनुसार नियंत्रित करता येऊ शकणाऱ्या या व्हीलचेअरसाठी त्यांनी एक प्रोटोटाइप तयार केला आहे. त्याचा उपयोग करून तसं मॉडेल तयार करण्यासाठी हे विद्यार्थी प्रयत्नशील आहेत.

प्रकल्पाविषयी थोडं... या प्रकल्पात दोन प्रकारचे सिग्नल कॅप्चर करण्यासाठी यांनी इलेक्ट्रोएन्सेफोलोग्राफी हेडसेट वापरण्यात आले आहेत. ते अटेन्शन लेव्हल आणि आय ब्लिंक स्ट्रेंथ मोजण्याचं काम करतात. तसंच यात रासबेरी पाय हा मायक्रोप्रोसेसर वापरला आहे. विद्यार्थ्यांनी एक अॅप तयार केलं असून, ते ईईजी हेडसेट आणि रासबेरी पाय दरम्यान संपर्क स्थापित करण्यासाठी वापरलं जातं. रुग्ण एखाद्या गोष्टीवर आपलं लक्ष केंद्रित करेल तेव्हा एका विशिष्ट पातळीनंतर रुग्णाची व्हीलचेअर सरळ पुढे जाईल. जेव्हा लक्ष केंद्रित करण्याची पातळी कमी होईल ती व्हीलचेअर थांबेल. व्हीलचेअरची दिशा बदलण्यासाठी डोळ्याची पापणी लवण्याची तीव्रता मोजून ती वापरली जाईल. एक सेकंदात डोळ्याची पापणी दोनदा हलवत असेल तर व्हीलचेअर डावीकडे वळेल. भविष्यात हे विद्यार्थी त्यानुसार मॉडेल तयार करण्याचा प्रयत्न करताहेत. त्यात स्मार्ट इमर्जन्सी ब्रेकिंग सिस्टिम कशी वापरता येईल याचा अभ्यास ते करताहेत. जेव्हा गोंधळल्यामुळे रुग्ण त्याचे डोळे बंद करेल तेव्हा व्हीलचेअर स्वतःहून थांबेल. म्हणजे मेंदूतल्या विचारांद्वारे या व्हीलचेअरचं नियंत्रण केलं जाईल. या विद्यार्थ्यांना प्राध्यापक दीपक शर्मा आणि मानसी कांबळी यांनी मार्गदर्शन केलं.